4.5 Order Manager Subsystem Overview

The Order Manager subsystem (OMS) manages all orders arriving via the Data Management subsystem's V0 Gateway (V0 GTWAY), (i.e., submitted by EDG and ECHO users), data orders submitted by the Spatial Subscription Server, orders from Machine to Machine Gateway (MTMGW), orders from the EPD Server (i.e., external subsetter request and S4PM) and orders from the Data Pool Web GUI (including HEG orders). The capability includes a server (the Order Manager Server) to which the data distribution orders are submitted. With release 7.20, all distribution orders are processed by the OMS. Once a request comes into the OMS subsystem, the server validates the request. Upon successful validation, the server stages the order in Data Pool storage area. For Ftp Pull requests, links are created from the staged files to the FtpPull directory in the Data Pool storage while for Ftp Push requests, the OMS Ftp Push driver directly distributes the data. Physical media requests are created on the physical media by the Production Modules. Upon successful shipment, OMS sends a Distribution Notice to the end user. An order is considered shipped (i.e., as soon as the request status is updated to "Shipped" in the MSS Database; for FtpPull order, the request status is updated to "Shipped" after the order is staged and file links are made in the Data Pool storage; for FtpPush order, request status is "Shipped" after Order Manager Server finishes pushing all the data associated to the order to its destination; For physical media order, the order is shipped when the Operator updates the request status to "Shipped" through the OMS GUI).

Special orders such as HEG and External Subsetter orders require further processing by the HEG Server or the External Subsetter. For HEG orders,, the Order Manager creates HEG requests per granule based on the processing instructions in the original HEG order. The HEG requests are submitted to the HEG Server through the HEG API. HEG server processes the HEG requests and returns the final output to the Order Manager Server which then distributes the final output to the end user. For External Subsetter Orders, the External Subsetter creates output granules which are then associated with the Order by the EPD Server. These output granules are later distributed by the Order Manager Server. The Order Manager Subsystem also includes a database that stores all order information persistently as soon as an order is received by EMD and before its receipt is acknowledged. This allows operators to resubmit an order if it encounters errors downstream, and allows the Order Management Service to perform some up front checks on the order and alert the operators if their intervention is needed.

Order Manager Subsystem Context

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

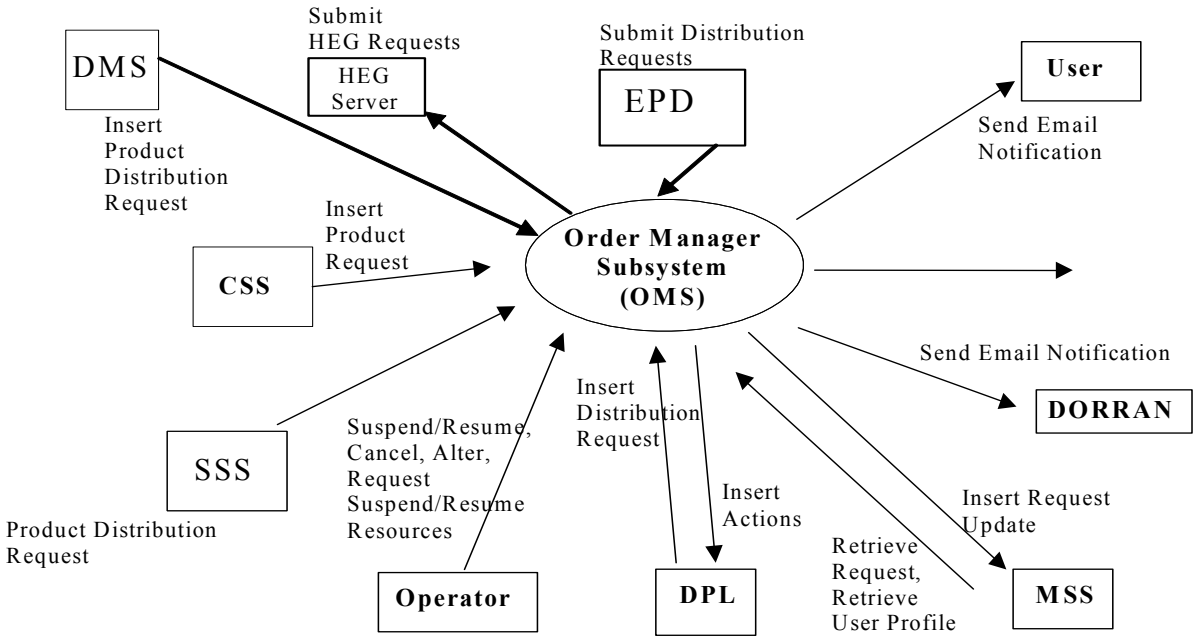


Figure 4.5-1. Order Manager Subsystem Context Diagram

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

Event	Interface Event Description
Submit Media Distribution Requests	The OMS processes all physical media distribution requests.
Submit Electronic Distribution Requests	The OMS processes all electronic distribution requests. Examples of electronic distribution requests are Ftp Push, Ftp Pull, and Secure Distribution.
Insert Product Distribution Requests	The OMS receives Product Distribution Requests from the **Data Management Subsystem (DMS)**, the **Spatial Subscription Server** (SSS) and the EPD Server (External Product Dispatcher).
Insert Media Distribution Requests	The OMS receives Media Distribution Requests from the **DPL Subsystem** (Web GUI).
Insert Actions	The OMS submits DPL insert actions to the **DPL Subsystem**.
Suspend/Resume, Cancel, Alter and Retrieve Requests	The **Operator** suspends resumes, cancels, alters and retrieves requests from the OMS (OMS DB).
Suspend/Resume Resources	The Operator suspends, resumes dispatching to all or selected resources.

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

Event	Interface Event Description
Send Email Notification	The OMS sends email notification to **users** and **DORRAN** when a request is altered, canceled or shipped or when a request is intervened by operators, or when an alert or intervention is generated. The OMS sends email notification to users from the **DPL** when OMS receives requests.
Insert Request	The OMS inserts requests into the MSS (MSS DB).
Update Request	The OMS updates request information in the MSS (MSS DB).
Retrieve Request	The OMS retrieves request information from the MSS (MSS DB).
Retrieve User Profile	The OMS retrieves user profile information from the MSS (MSS DB).
Submit Distribution Request	The OMS receives Distribution Requests from the EPD Server (External Product Dispatcher)
Submit Heg Request	The OMS submits HEG requests to the HEG Server.

Order Manager Subsystem Structure

- The Order Manager Subsystem consists of three CSCIs: the OMSRV, the OM GUI (described in the 609 document), and the Production Module. The Order Manager Server (EcOmOrderManager) is a software configuration item. The Order Manager Server receives product distribution requests and submits them to the appropriate EMD component based upon the media type specified for the request. The OMS server stages granules associated with a request in the DPL storage area and then distributes the data via electronic media (i.e Ftp Push, Ftp Pull, SCP) or physical media. . For special orders such as HEG Orders, the Order Manager creates HEG requests based on processing instructions and submits the HEG requests to the HEG Server. The output of the HEG requests are later distributed to the end user. Similar to HEG Orders, output granules associated with an External Subsetter requests are distributed to the end user. Order Manager Subsystem information is stored persistently in a relational Database Management System (DBMS). The Order Manager GUI (OMGUI) is used to monitor and control the operations of the Order Manager Server. In addition, the OMGUI is used to respond to Operator Intervention Requests generated by the Order Manager Server. Production module is responsible for creating the physical medial associated to hard media requests.

Use of COTS in the Order Manager Subsystem

- RogueWave's Tools.h++

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

- Sybase Open Client / CT_LIB

  The Sybase Open Client provides access between OMS custom code and the Sybase SQL Server DBMS.

- Sybase Server

  The Sybase SQL server provides access for OMS to insert, update and delete Product Distribution Requests, OMS configurations, and Operator Interventions. The Sybase SQL Server must be running during operations for the OMS to process Product Distribution Requests.

4.5.1 Order Manager Subsystem Software Description

4.5.1.1 Order Manager Server CSCI Functional Overview

The Order Manager Server (OMSRV) CSCI executes as a process and interacts with the following CSCIs: Order Manager Database, the Science Data Server (SDSRV), and the Data Pool System (DPL). The V0 Gateway, the Spatial Subscription Server (SSS), Machine to Machine Gateway (MTMGW), HEG Server, EPD Server and the Data Pool Web GUI submit product distribution requests to the OMS. These requests are all validated. . Upon successful validation, the server stages the granules in a request in the DPL storage area. Hard media requests staged in DPL storage area are distributed through the production module while electronic Ftp push media requests are directly pushed to the end user. Note that Order Manager Server dispatches HEG requests to the HEG Server for processing before being distributed to the end user. For invalid request, an Operator Intervention is generated. DAAC OPS personnel can use the Order Manager GUI to correct and resubmit the request. In response to an intervention, the Operator can also generate an email message, which is sent to the user by the Order Manager Server. The Order Manager Server also generates an alert and sends an email to a pre-configured email address when it detects internal or external resource failure. While a resource is suspended, the OMS Server halts dispatching of the requests that are utilizing the suspended resource.

4.5.1.2 Order Manager Server CSCI Context

Figure 4.5-2 is the Order Manager Server CSCI context diagrams. The diagrams show the events sent to the Order Manager Server CSCI and the events the Order Manager Server CSCI sends to other CSCIs. Table 4.5-2 provides descriptions of the interface events shown in the Order Manager Server CSCI context diagrams.

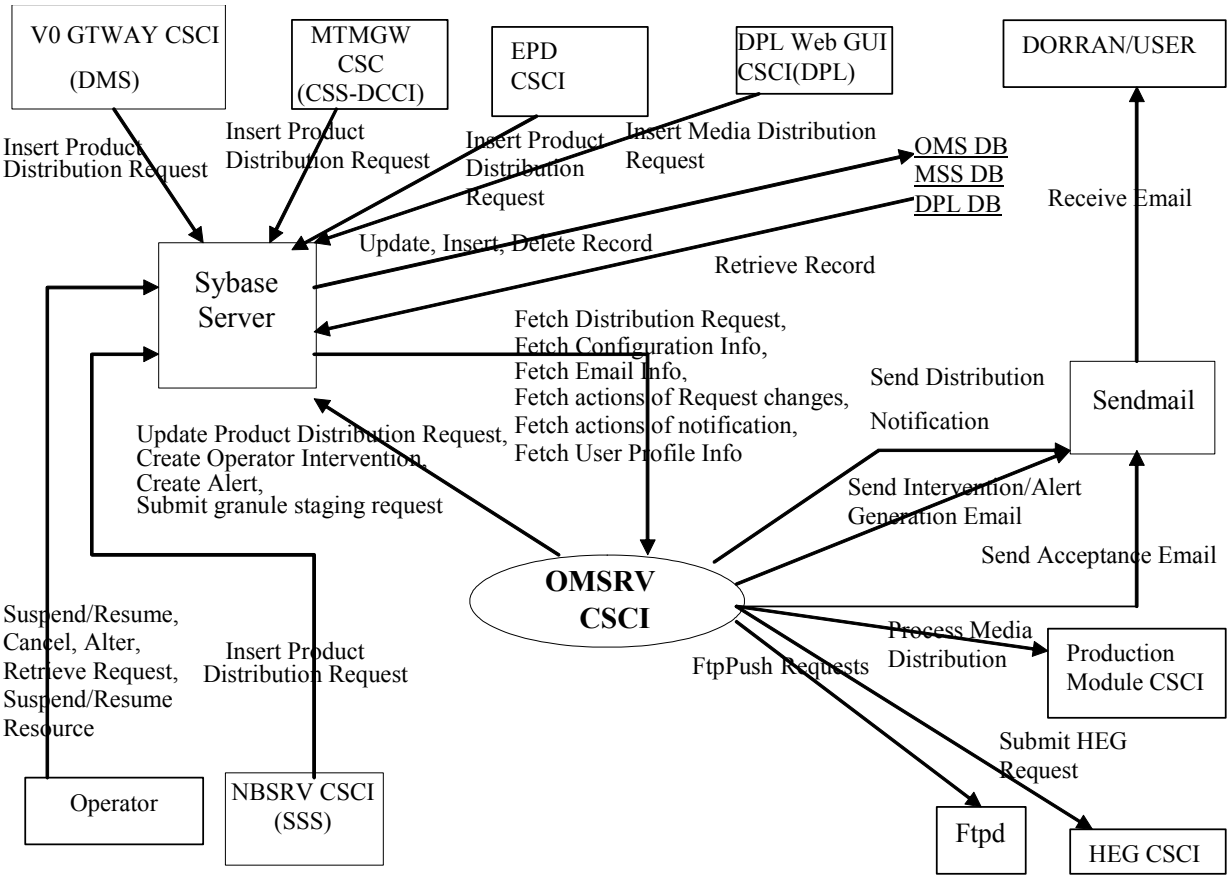


Figure 4.5-2. Order Manager Server CSCI Context Diagram

Table 4.5-2. Order Manager Server CSCI Interface Events (1 of 2)

Event	Interface Event Description
Submit Media Distribution Request	The OMSRV CSCI processes physical media requests using the Production Module CSCI.
Submit Electronic Distribution Request	All electronic distribution requests are processed by the OMS CSCI.
Send Distribution Notification	The OMSRV CSCI sends distribution notifications to the **end-users**.
Send Intervention/Alert Generation Email	The OMSRV CSCI sends intervention/alert generation email to the **end-users**.
Send Acceptance Email	The OMSRV CSCI sends a request acceptance email to the **DPL Web GUI** users upon receiving the request in the OMS DB.
Receive Email	The **DORRAN** billing and accounting system and **User** receive a status email sent by the OMSRV CSCI when request is interventioned, shipped or failed

Table 4.5-2. Order Manager Server CSCI Interface Events (2 of 2)

Event	Interface Event Description
Insert Product Distribution Request	The V0 GTWAY CSCI, NBSRV CSCI, SCLI CSCI and Machine–to-Machine Gateway (MTMGW) CSC insert product distribution requests into the **Sybase Server** (**OMS DB**) to be queued for processing by the OMSRV CSCI.
Insert Media Distribution Request	The DPL Web GUI inserts media distribution request into the **Sybase Server** (**OMS DB**).
Update Product Distribution Request	The OMSRV CSCI updates product distribution requests in the **Sybase Server** (**OMS DB** as well as the **MSS DB**).
Create Operator Intervention	The OMSRV CSCI creates new Operator Intervention for request failures in the **Sybase Server (OMS database)**.
Create Alert	The OMSRV CSCI creates an alert for resource failures and stores the alert in the **Sybase Server (OMS database)**.
Submit granule staging request	The OMSRV CSCI submits a request to stage a granule to the **Sybase Server** (DPL storage in the OMS DB), which in turn calls DPL stored procedures to insert an action into the DPL DB.
Fetch Distribution Request	The OMSRV CSCI retrieves information associated with a product distribution request from the **Sybase Server** (OM Database).
Fetch Configuration Info	The OMSRV CSCI retrieves the OMSRV Configuration information from the **Sybase Server**.
Fetch Email Info	The OMSRV CSCI retrieves information related to an operator intervention required to generate an email notification from the **Sybase Server**.
Fetch actions of Request changes	The OMSRV CSCI retrieves actions regarding request changes, such as, request priority change, cancel request, suspend request, and update request ftppush parameters from the **Sybase Server**.
Fetch actions of notification	The OMSRV CSCI retrieves actions regarding granule staged and DPL file system modified notification from the **Sybase Server**.
Fetch User Profile Info	The OMSRV CSCI retrieves user profile information from the Sybase Server (OMS DB), which in turn calls an MSS stored procedure to retrieve user profile information from the MSS DB.
FtpPush Request	The OMSRV CSCI Ftp Pushes a request to the **end-user**.
Process Media Distribution	The OMSRV CSCI submits physical media request with Synergy IV processing mode to the Production Module.
Submit HEG Request	The OMSRV CSCI submits HEG requests to the HEG Services for processing.
Suspend/Resume, Cancel, Alter and Retrieve Request	The Operator suspends, resumes, cancels, alters and retrieves requests from the **Sybase Server** (**OMS DB**).
Suspend/Resume Resources	The OMSRV CSCI suspends or resumes dispatching to all or selected resources in the **Sybase Server**.
Update, Insert, Delete Record	The **Sybase Server** performs update, insert, and delete database operations to the OMS DB, MSS DB and DPL DB.
Retrieve Record	The **Sybase Server** performs retrieval database operations to/from the OMS DB, MSS DB and DPL DB.

4.5.1.3 Order Manager Server CSCI Architecture

Figure 4.5-3 is the Order Manager Server (OMSRV) CSCI architecture diagram. The diagram shows the events sent to the OMSRV CSCI processes and the events the OMSRV CSCI processes send to other processes.

The OM Server CSCI consists of one process. This process is the EcOmOrderManager process.

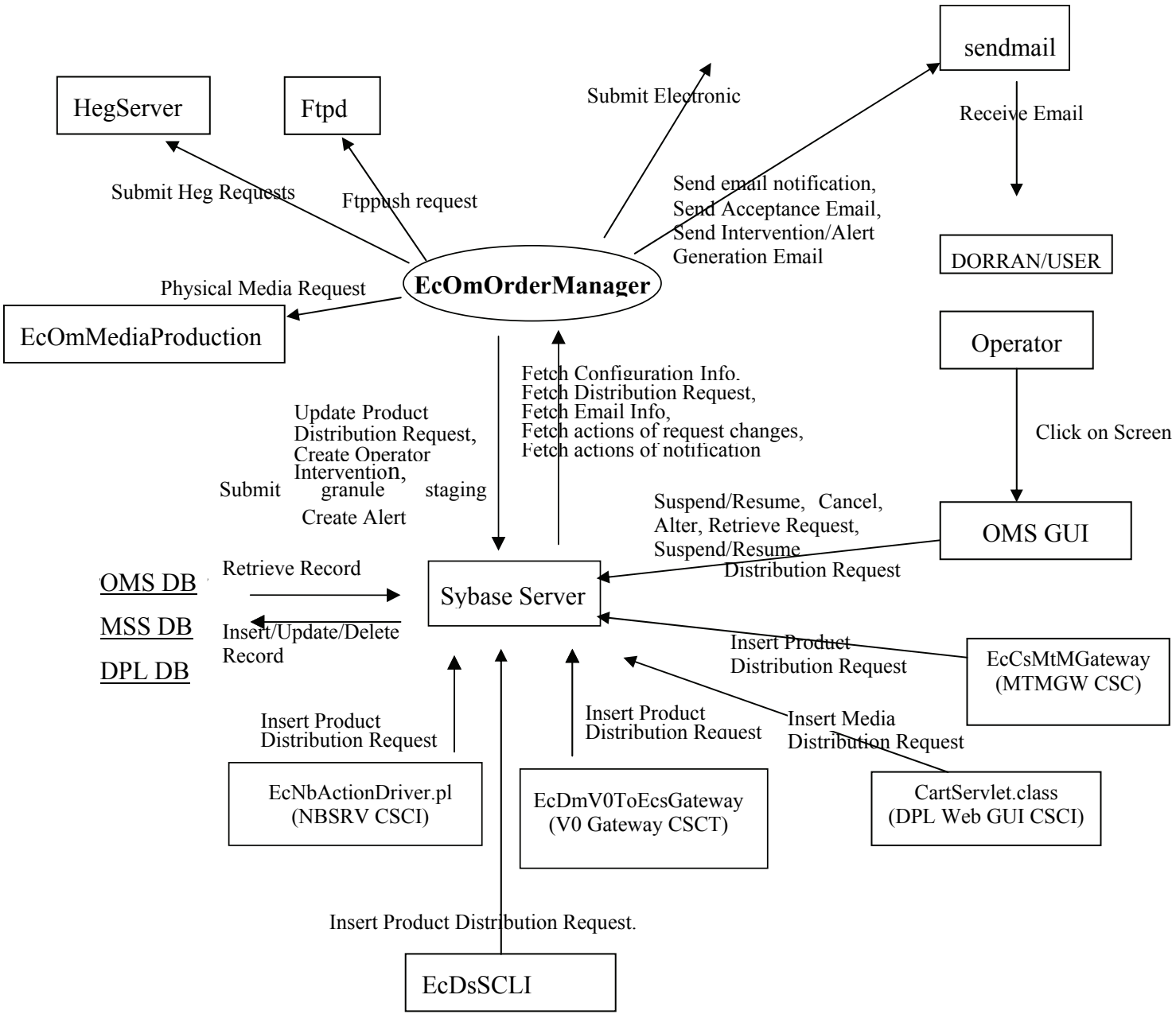


Figure 4.5-3. Order Manager Server CSCI Architecture Diagram

4.5.1.4 Order Manager Server CSCI Process Description

Table 4.5-3 provides descriptions of the processes shown in the OMSRV CSCI architecture diagram.

Table 4.5-3. OMSRV CSCI Process

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcOmOrderManager	Server	OMSHW	Developed	The Order Manager Server stages the request data into the DPL storage. Hard media request data staged in DPL storage are distributed through the Production Module while Ftp media requests are directly pushed to the end user. Note that HEG requests are first dispatched to the HEG Server for processing. The processed output is then distributed to the end user. Order Manager Server sends a Distribution Notification to the end-user on completing an order.

EMD Baseline Information System (EBIS) Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.5.1.5 Order Manager Server CSCI Interface Description

Table 4.5-4 provides descriptions of the interface events shown in the Order Manager Server (OMSRV) CSCI architecture diagram.

Table 4.5-4. Order Manager Server CSCI Process Interface Events (1 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
Physical Media Requests	One or more per service requests	*Process:* EcOmPdMediaProduction	*Process:* EcOmOrderManager *Class:* OmSrPrepareMediaAction OmSrCreateMediaAction	The Order Manager Server uses the production module to create the physical media.
Insert Product Distribution Request	One per service request	*Process:* Sybase Server (COTS)	*Process:* EcDmV0ToEcsGateway EcCsMtMGateway *Script:* EcNbActionDriver.pl	The EcDmV0ToEcsGateway, EcCsMtMGateway and EcNbActionDriver.pl insert product distribution request into OMS DB.
Submit Heg Request	One or more per Service Request	*Process:* *HgServer*	*Process:* EcOmOrderManager *Class:* OmSrHegProcessingAction	The EcOmOrderManager submits HEG requests to the Heg Server.

Table 4.5-4. Order Manager Server CSCI Process Interface Events (2 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
Insert Media Distribution Request	One per service request	*Process:* Sybase Server (COTS)	*Process:* CartServlet.class	The CartServlet.class (DPL Web GUI CSCI) inserts media distribution request. Media distribution request could be HEG request.
Send acceptance email	One per email notification request	*Process:* Sendmail (COTS)	*Process:* EcOmOrderManager *Class:*	The EcOmOrderManager sends email notification to the DPL Web GUI end-users upon receipt of the request.
Send Intervention/ Alert Generation email	One per email notification request	*Process:* Sendmail (COTS)	*Process:* EcOmOrderManager *Class:*	The EcOmOrderManager sends intervention/ alert generation email to a configured user email account.
Receive Email	One per email notification	*End User:* DORRAN Configured User	*Process:* Sendmail (COTS)	The DORRAN billing and accounting system and User receive email sent by the EcOmOrderManager.
Send email notification	One per email notification request	*Process:* Sendmail (COTS)	*Process:* EcOmOrderManager *Class:* OmSrEmailRequest	The EcOmOrderManager sends email notifications to the end-users.
Fetch configuration info	One per startup/ One per configurable interval	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Class:* OmSrDbInterface	The EcOmOrderManager retrieves configuration information from the Sybase Server (OMS database).
Fetch Distribution Request	One per configurable interval	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Classes:* OmSrDbInterface, OmSrDistributionRequest	The EcOmOrderManager retrieves information associated with a product distribution request from the database.

Table 4.5-4. Order Manager Server CSCI Process Interface Events (3 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
Fetch actions of request changes	One per configurable interval	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Classes:* OmSrDbInterface, OmSrDistributionRequest	The EcOmOrderManager retrieves actions regarding request changes, such as, request priority change, cancel request, suspend request, and update request Ftp Push parameters.
Fetch actions of notification	One per configurable interval	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Classes:* OmSrDbInterface, OmSrDistributionRequest	The EcOmOrderManager retrieves actions regarding granule staged, and DPL file system modified notification.
Fetch email info	One per configurable interval	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Classes:* OmSrDbInterface, OmSrDistributionRequest, OmSrEmailRequest	The EcOmOrderManager retrieves information related to an operator intervention required to generate an email notification from the Sybase Server (OM DB).
Update Product Distribution Request	One per request	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Classes:* OmSrDbInterface, OmSrDistributionRequest	The EcOmOrderManager updates existing product distribution requests in the Sybase Server (OMS DB and MSS DB).
Create Operator Intervention	One per request	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Class:* OmSrDbInterface	The EcOmOrderManager creates a new Operator Intervention request in the Sybase Server (OM DB).

Table 4.5-4. Order Manager Server CSCI Process Interface Events (4 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
Create Alert	One per resource	*Process:* Sybase Server (COTS)	*Process:* EcOmOrderManager *Library:* Sybase Ct-library *Class:* OmSrDbInterface	The EcOmOrderManager creates an alert related to a resource failure such as Ftp Push Destination, Archive, and DPL File System failure to store in the Sybase Server.
Insert Product Distribution Request	One per service request	*Process:* Sybase Server (COTS)	*Process:* EcDmV0ToEcsGateway EcCsMtMGateway EcDsSCLI *Script:* EcNbActionDriver.pl *Library:* OmClientlib *Classes:* OmSrDbInterface, OmSrV0InputIf	The EcDmV0ToEcsGateway and the EcNbActionDriver, EcCsMtMGateway, the EcOmSCLI, DPL Web GUI insert product distribution requests into the Sybase Server (Order Manager DB).
Insert/Update/ Delete/Retrieve Record	One per request	*Database:* OMS DB, MSS DB, DPL DB (COTS)	*Process:* Sybase Server (COTS)	The Sybase server performs database operations (inserts, updates, deletions and retrievals) to the OMS DB, DPL DB and MSS DB.
Ftppush request	One per file	*Process:* Ftpd (COTS)	*Process:* EcOmOrderManager	The EcOmOrderManager Ftp Pushes request data to the end-user.
Click on Screen	One per click	*Scripts:* OMS GUI scripts.	Operator	The Operator clicks on the screen to select the action.
Suspend/ Resume, Cancel, Alter and Retrieve Requests	One per click	*Process:* Sybase Server (COTS)	*Script:* OMS GUI script	The OMS GUI scripts send suspend/resume, cancel, alter and retrieve request commands to the Sybase Server.
Suspend/ Resume Resource	One per click	*Process:* Sybase Server (COTS)	*Script:* OMS GUI script	The OMS GUI scripts send suspend/resume resource commands to the Sybase Server.

4.5.1.6 Data Stores

There are data stores associated with the Order Manager Server. They are the OMS DB, DPL DB and MSS DB. Table 4.5-5 provides a description of these data stores.

Table 4.5-5. CSCI Data Stores

Data Store	Type	Description
OMS DB	Sybase	OMS Database is designed to store the persistent information of user request, processing mode configuration, media configuration, staging policy configuration, Ftp Push policy configuration, request aging configuration, and request cleanup configuration.
DPL DB	Sybase	The Data Pool (DPL) database implements the large majority of the persistent data requirements for the DPL subsystem which supports Large online cache of important EMD data at each DAAC and avoids tape access to EMD archive.
MSS DB (Order Tracking DB)	Database	The Order Tracking DB contains product orders and user requests with the associated current processing status.

4.5.1.7 Production Module CSCI Functional Overview

The Production Module is the interface between Order Manager Server and the various tape and disc hardware. Order Manager Server places an input file for the Production Module into a SAN file system visible from the platform to which the physical media devices are attached. The Production Module references this file and stages the indicated data under a single directory so that it can be processed to tape or disc. The Production Module processes one volume of a request at a time. For tapes the Production Module writes to media through a system call to pax. For discs, it creates an image file and then call makes a system call to the Luminex press program. The Production Module returns overall job status to Order Manager Server through QuickServer. If there are errors for particular granules, the Production Module places a list of the problem granules in a file, which can be referenced by Order Manager across the same SAN mount in the event of failure.

Verification is handled by a Perl script, OmPdQcMain.pl. The script interacts with the tape and disc devices to do a listing of the media and compares the filenames and file sizes listed with those created in a summary file during production. Volume and Granule level verification status is returned to the Order Manager in the same manner as in production.

4.5.1.8 Production Module CSCI Context

Figure 4.5-4 is the Production Module CSCI Context Diagram. The diagram indicates the interaction the Production Module has with the Order Manager CSCI and the Data Pool File System.

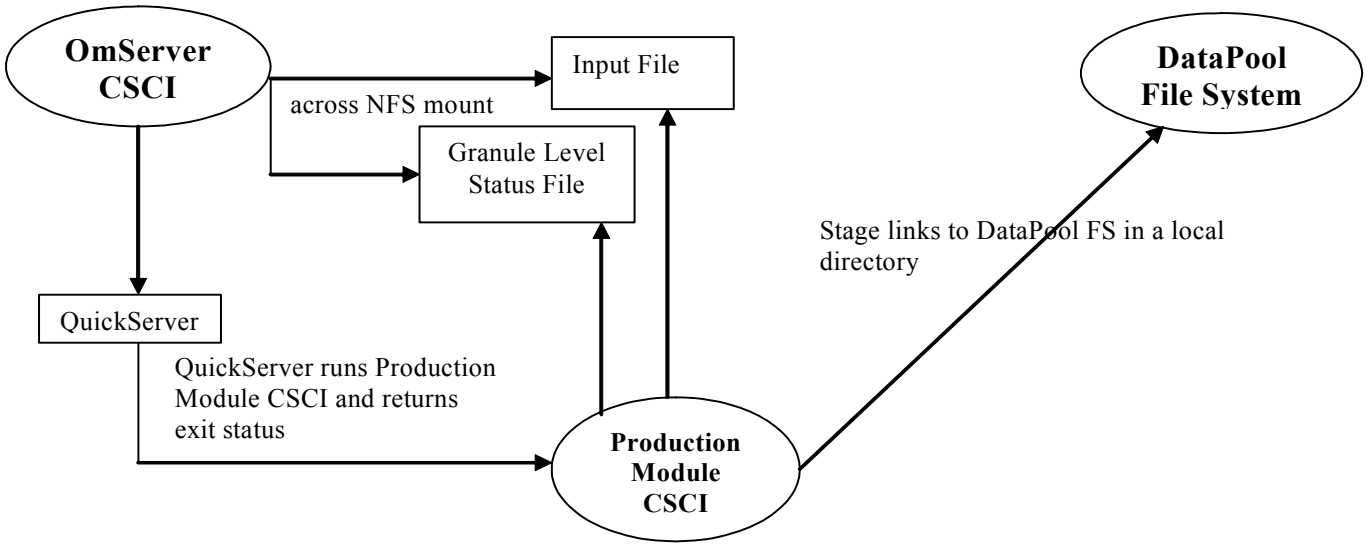


Figure 4.5-4. Production Module CSCI Context Diagram

Table 4.5-6 describes the Production Module CSCI Interface Events.

Table 4.5-6. Production Module CSCI Interface Events (1 of 2)

Event	Interface Event Description
Tape Media Preparation	The Production Module CSCI stages soft links to the indicated Data Pool files under a single directory. It creates a summary file and the file used for the tape label. It returns status to the Order Manager CSCI through QuickServer.
Tape Media Creation	The Production Module CSCI tars the data to tape, prints the tape label and returns status to the Order Manager CSCI through QuickServer.
Disc Media Preparation	The Production Module CSCI stages soft links to the indicated Data Pool files under a single directory. It calls the COTS product, mkisofs, to create an ISO image file. It creates the files for the jewel case insert, the summary file and returns status to the Order Manager CSCI through QuickServer.
Disc Media Creation	The Production Module CSCI makes a system call to the Luminex press command. The Production Module polls the press log until the disc burning process has reached a terminal state.
Granule Level Error	The Production Module CSCI writes the granule numbers and error code to a file which can be seen by Order Manager and returns a failure status code
Tape Verification	The Production Module CSCI QC script does a listing of the tape in the specified drive. It compares the filenames and sizes in a listing with the summary file created during the production process. Status is returned to Order Manager Server though QuickServer.

Table 4.5-6. Production Module CSCI Interface Events (2 of 2)

Event	Interface Event Description
Disc Verification on PC	A Production Module CSCI script runs continually on a PC. It polls a mapped Unix drive for the appearance of a signal file. Using QuickServer, Order Manager starts the Production Module CSCI QC script, which places the signal file. The PC script does a listing of the CD/DVD drive on the PC and places it into a file on the Unix drive. The QC script then parses this file and the summary file and compares the filenames and sizes. Status is returned to Order Manager Server though QuickServer.
Disc Verification on UNIX	The Production Module CSCI QC script does a listing of a defined disc drive mount point and compares the filenames and sizes with those in the summary file. Status is returned to Order Manager Server though QuickServer.
Order Cleanup	The Production Module CSCI cleanup script is executed by Order Manager through QuickServer and deletes temporary files not needed for later event tracing.
Archive/Cleanup	The Production Module CSCI Archive/Cleanup GUI creates a cleanup.sh file and writes an entry in the crontab to execute this file. Remaining artifacts and logs can be assigned for archiving or removal

4.5.1.9 Production Module CSCI Architecture

Figure 4.5-5 is the Production Module CSCI architecture diagram. The diagram shows the interaction with the media hardware as directed by the Order Manager CSCI remotely through QuickServer.

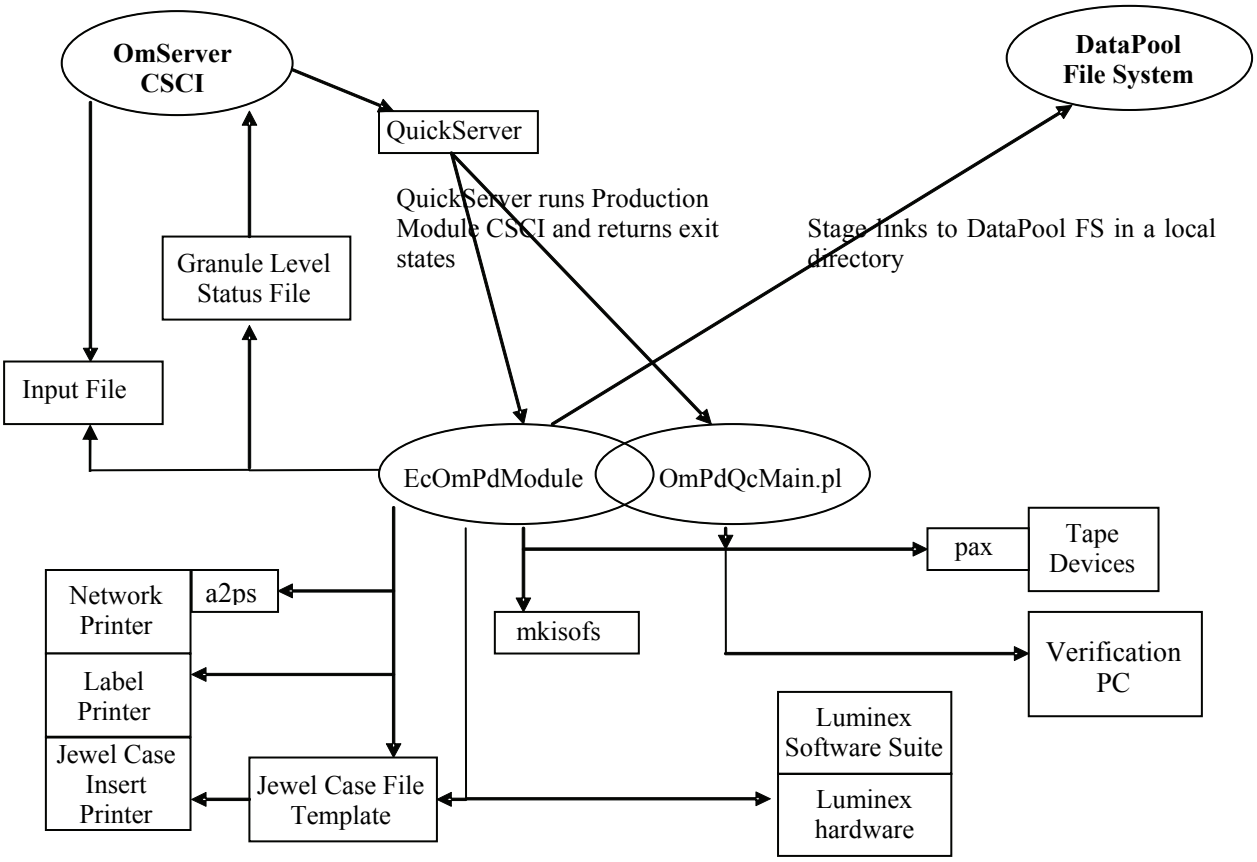


Figure 4.5-5. Production Module CSCI Architecture Diagram

4.5.1.10 Production Module CSCI Process Description

Table 4.5-7 provides descriptions of the processes shown in the Production Module Architecture Diagram.

Table 4.5-7. Production Module CSCI Process

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcOmPdModule	driver	DIPHW	Developed	Receives instructions for physical media processing from Order Manager through a .PPF file. Also it prints the tape label and the jewel case insert.
OmPdQcMain.pl	Perl script	DIPHW	Developed	This script interfaces with the tape devices, the Linux disc drive mount point and a script running on the QC PC to verify filenames and sizes written to physical media.
OmPdNtQC.pl	Perl script	DIPHW	Developed	Runs on the QC PC and polls the Unix drive for the arrival of a signal file. It then does a listing of the PC's CD/DVD drive and puts the result in a file on the Linux box where it can be seen by OmPdQcMain.pl.
OmPdCleanup.pl	Perl script	DIPHW	Developed	A script run by Order Manager through QuickServer which deletes leftover request files not needed for logging or later reference.
OmPdCleanupGUI	tcl GUI	DIPHW	Developed	Directs periodic removal or archive of remaining request and log files through an entry in the crontab.
EcOmPdPrintJCIFile.pl	Perl script	DIPHW	Developed	Inserts granule and request data into the Jewel Case Insert template and sends the new file to the Jewel Case printer.
mkisofs	process	DIPHW	COTS	Creates an ISO image file for burning to a CD/DVD that can be read by PC or UNIX operating systems.
a2ps	printer driver	DIPHW	COTS	Facilitates the formatted printing of the Packing List and the QC reports.
Jewel Case Insert template	file	DIPHW	COTS	Postscript format file containing images and place holders for granule and request data.

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.5.1.11 Production Module CSCI Interface Description

Table 4.5-8 provides descriptions of the interface events shown in the Order Manager Server CSCI architecture diagram.

Table 4.5-8. Production Module CSCI Interface Events

Event	Event Frequency	Interface	Intiated By	Event Description
Media Preparation	Once per request volume	QuickServer	Order Manager	Order Manager submits a RequestId_Vol.PPF file across SAN mount containing all non-configuration data needed for physical media processing.
Media Creation	Once per request volume	QuickServer	Order Manager	Order Manager submits a RequestId_Vol.PPF file across SAN mount containing all non-configuration data needed for physical media processing.
Verification	Once per request volume	QuickServer	Order Manager	Order Manager submits a command containing the requestId and the volume number.
Cleanup	Once per order	QuickServer	Order Manager	Order Manager submits a command containing the requestId.

4.5.1.12 Data Stores

The Production Module CSCI receives all needed data from a configuration file and an input file from Order Manager. It does not interface with a relational database.

4.5.1.13 Production Module Hardware

The Production Module Hardware is not shared with any other subsystem.

4.6 Planning Subsystem Overview (REMOVED)

4.7 Data Processing Subsystem Overview (REMOVED)

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) Maintenance and Development Project (EMD)
- Transfer information between the EMD 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
- Retrieve attribute-value pairs from the Configuration Registry

Communications Subsystem Context Diagram

Figure 4.8-1 is the Communications Subsystem (CSS) context diagrams and Table 4.8-1 provides descriptions of the interface events shown in the CSS context diagrams. **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 CSMS subsystems.

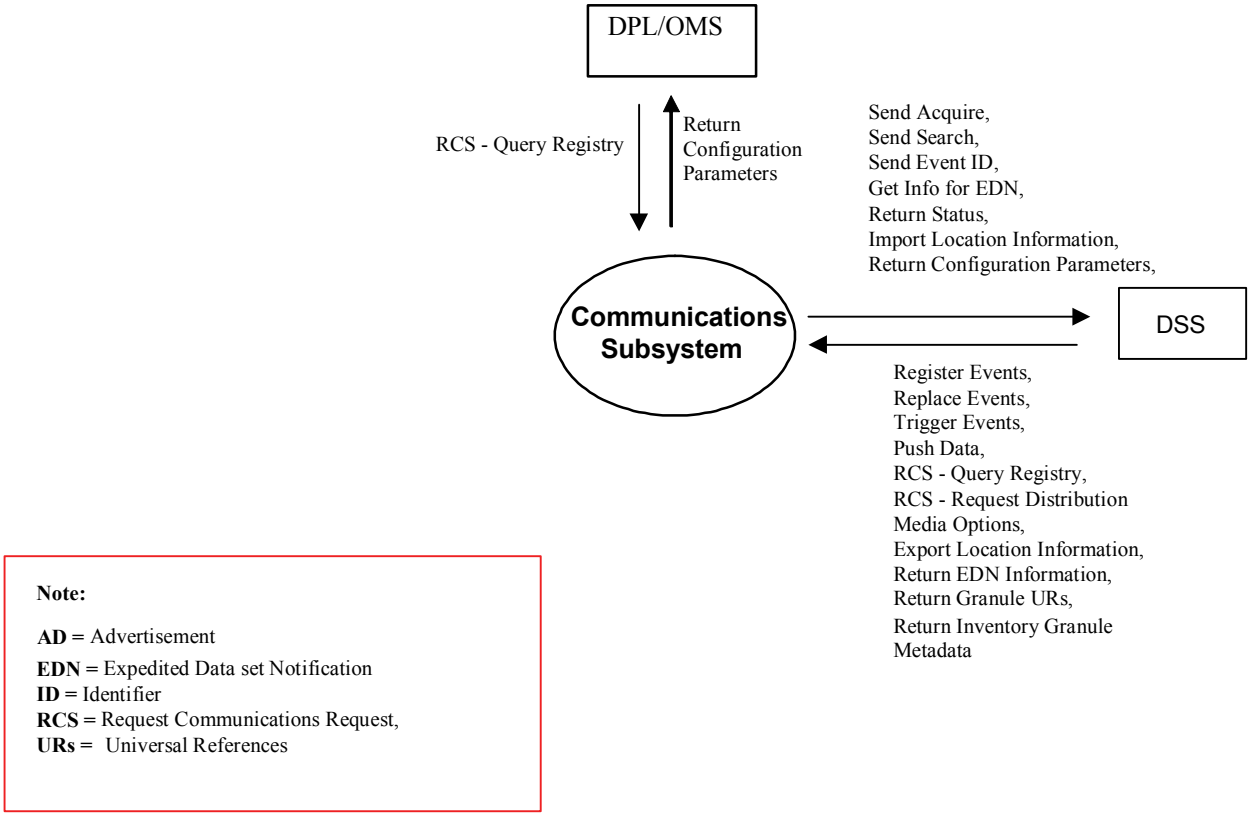


Figure 4.8-1. Communications Subsystem (CSS) Context Diagram

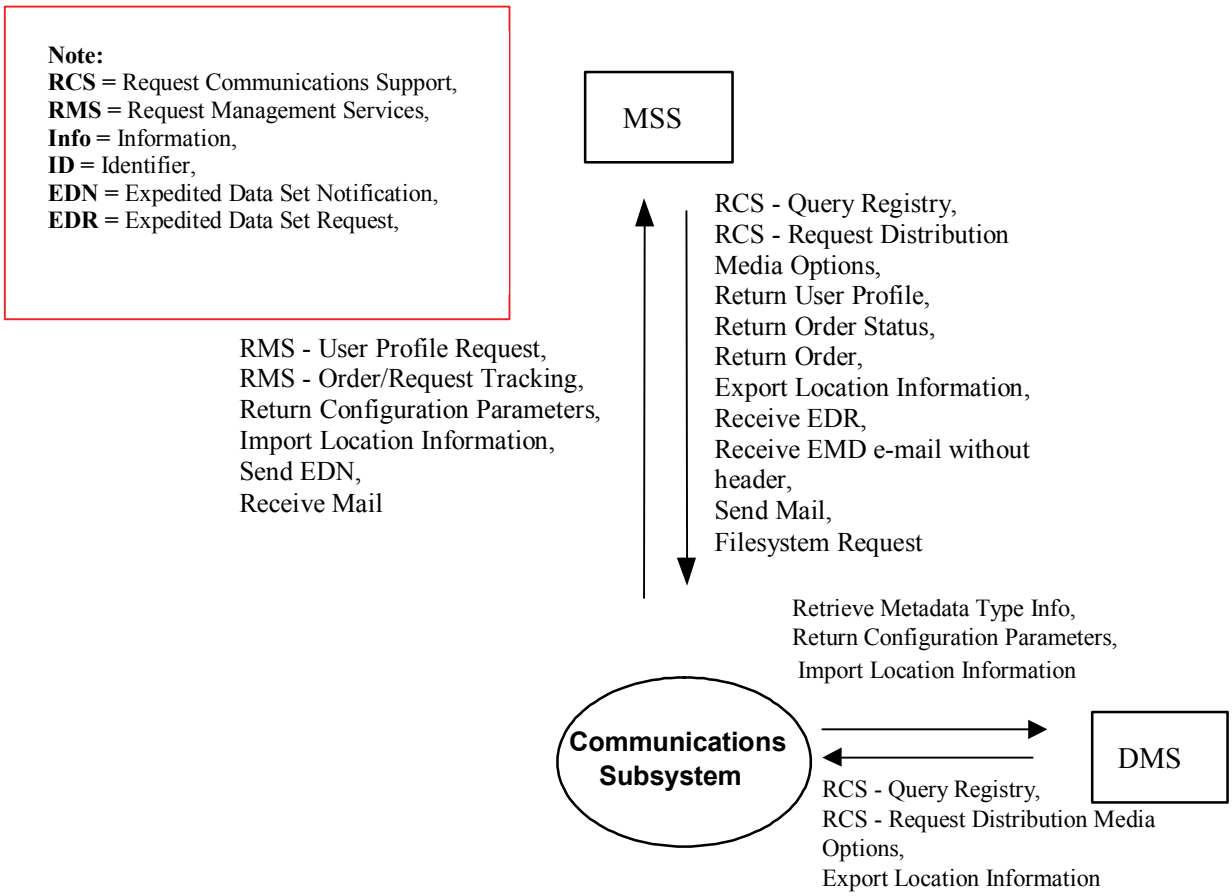


Figure 4.8-1. Communications Subsystem (CSS) Context Diagram (cont.)

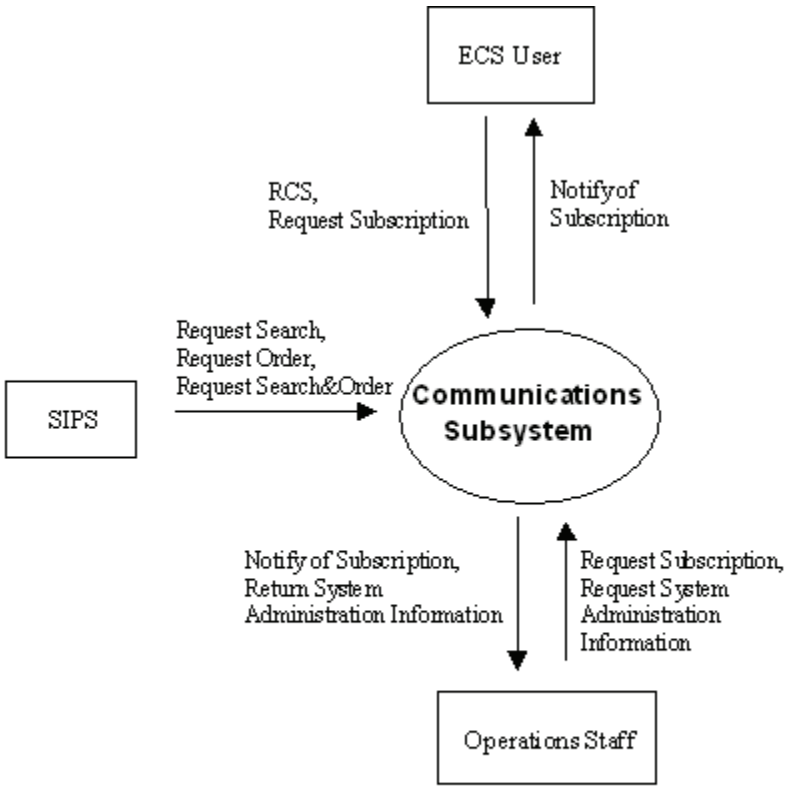


Figure 4.8-1. Communications Subsystem (CSS) Context Diagram (cont.)

Table 4.8-1. Communications Subsystem (CSS) Interface Events (1 of 3)

Event	Interface Event Description
Send Acquire	An “acquire” (instruction to obtain data) is created by the CSS and sent to the **DSS** via CCS Middleware calls. This is similar to the “Request Product” interface event, except it applies to EDOS expedited data. Also, the Subscription Server sends an "acquire" command to the DSS when an "acquire" action is specified in a subscription.
Send Search	The CSS sends search requests received via the SIPS interface to the **DSS** on behalf of an external ECS user.
Send Event ID	The CSS sends Event IDs to the **DSS** when ESDTs are installed or when ESDTs are updated by adding additional events.
Get info for EDN	Expedited Data Set Notification (EDN) information is obtained from the **DSS**, by request, and used by the CSS to send messages.

Table 4.8-1. Communications Subsystem (CSS) Interface Events (2 of 3)

Event	Interface Event Description
Return Status	Status returned by the CSS to the **DSS** to simply indicate that the request was received, not that the action succeeded.
Import Location Information	The **DSS** retrieves physical and logical server location information from the CSS.
Return Configuration Parameters	The **DPL INGEST, DSS, MSS,** and **DMS** receive the configuration parameters and associated values from the Registry Server within the CSS.
Return Dist. Media Options	The **DSS** receives the requested distribution media options from the CSS.
Register Events	The **DSS** sends the subscription events for an Earth Science Data Type to the CSS Subscription Server when an ESDT is installed into the system or when an ESDT is updated by adding additional events.
Replace Events	The **DSS** sends the updated subscription events with modified qualifiers for an Earth Science Data Type (ESDT) to the CSS Subscription Server when an ESDT is updated. This event replaces the original event in the DSS.
Trigger Events	The **DSS** notifies the CSS (via an event trigger) when a subscription event occurs on an ESDT Service.
Push Data	The **DSS** assembles instructions to send data to other external users via the CSS. The DSS pushes data, via the FTP service and followed by a signal file, to the destination specified in an acquire instruction (by particular ESDTs that function this way).
Request Communications Support (RCS)	The CSS provides a library of services available to **each CSMS** subsystem. The subsystem services required to perform specific assignments are requested from the CSS. These services include: • CCS Middleware Support • Database Connection Services • File Transfer Services • Network & Distributed File Services • Bulk Data Transfer Services • File Copying Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Message Passing • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry • Request Distribution Media Options from the Configuration Registry
Export Location Information	The **DSS**, **DMS**, and **MSS** send physical and logical server location information to the CSS for data location.
Return EDN Information	The CSS receives and uses the Expedited Data Set Notification (EDN) information from the **DSS** to send messages to users.

Table 4.8-1. Communications Subsystem (CSS) Interface Events (3 of 3)

Event	Interface Event Description
Return Granule URs	The CSS receives Earth Science Data Type (ESDT) Universal References (URs) for the requested granules from the **DSS**.
Return Inventory Granule Metadata	The CSS (MTMGW Server) receives the inventory granule metadata identifying the scene within the granule from the **DSS** based on an inventory search request.
Return User Profile	The CSS receives user profile information from the **MSS** (Accountability Management Service) to authenticate a user.
Filesystem Request	The CSS (NFS client) receives requests for ECS files and directories via an established mount point from the **MSS**. The CSS (NFS Server) makes the storage device(s) and its data accessible for use by the clients.
Retrieve Metadata Type Info	The CSS retrieves type information for qualifying metadata specified in a SIPS search request from the **DMS**.
Return Dist. Media Options	The CSS returns distribution media options to the **DMS** and **MSS** for distribution requests.
Import Location Information	The **DMS** and **MSS** request server location information from the CSS for data searching purposes.
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: • **System startup and shutdown** – Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625). • **User Profile Request** – The **MSS** provides requesting subsystems with User Profile parameters such as e-mail address and shipping address to support their processing activities. • **Order/Request Tracking** – The **MSS** provides an order tracking service to requesting subsystems to create and track user product orders by request.
Request System Administration Information	The **Operations Staff** requests information on system administration including application administration, fault metrics, performance metrics and system alarms.
Return System Administration Information	The **Operations Staff** receives information on system administration including application administration, fault metrics, performance metrics and system alarms.
Request Search	Search requests are sent to the CSS via the **SIPS** interface.
Request Order	Order requests are sent to the CSS via the **SIPS** interface.
Request Search&Order	Integrated search and order requests are sent to the CSS via the **SIPS** interface.
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: • **Order/Request Tracking** – The **MSS** provides an order tracking service to requesting subsystems to create and track user product orders by request.

Communications Subsystem Structure

Note: The CSS logical names used in this document do not exactly match the physical names in the directory structure where the software is maintained. Therefore, after the logical name of each Computer Software Component (CSC) in parentheses, there is a physical directory structure name where the software is found. For example, the DCCI CSCI software can be found under the directory structure Distributed Object Framework (DOF) and the Server Request Framework software can be found under the directory structure /ecs/formal/common/CSCI_SRF.

The CSS is composed of one CSCI, the Distributed Computing Configuration Item (DCCI, the software is found in directory DOF) and one HWCI. The CSS software is used to provide communication functions, processing capability, and storage.

Use of COTS in the Communications Subsystem

Note: The following RogueWave Libraries are currently delivered with custom code as static libraries. A separate installation of dynamic libraries is no longer required.

- RogueWave's Tools.h++

  The Tools.h++ class libraries provide basic functions and objects such as strings and collections.

- RogueWave's DBTools.h++

  The DBTools.h++ C++ class libraries provide interaction, in an object-oriented manner, to the Sybase ASE database SQL server. The DBTools provide a buffer between the CSS processes and the relational database used.

- RogueWave's Net.h++

  The Net.h++ C++ class libraries, which provide functions and templates that facilitate writing applications, which communicate with other applications.

Other COTS products include:

- 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 Adaptive Server Enterprise (ASE)

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

In addition, the Configuration Registry stores configuration values for ECS applications in the Sybase database. The Configuration Registry Server retrieves the values from the database via a Sybase ASE.

- CCS Middleware

  CCS Middleware provides a common NameServer Mechanism, which packages the common portions of the communication mechanisms into global objects to be used by all subsystems. It provides a set of standard CCS Proxy/Server classes, which encapsulates all of the common code for middleware communications (e.g., Portals, Couplers, RWCollectables, etc.). It also provides a code generator, which produces the application specific proxy & server code. This allows the software engineer to concentrate on the application specific code without worrying about the infrastructure.

- UNIX Network Services

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

Error Handling and processing

EcUtStatus is a class used throughout the ECS custom code for general error reporting. It is almost always used as a return value for functions and allows detailed error codes to be passed back up function stacks.

When an error occurs, the error is logged into the applications log (ALOG). The Communications Subsystem (CSS) and System Management Subsystem (MSS) have two main mechanisms to handle the error:

1. Return an error status
2. Throw an exception.

The CSS uses the following classes for error handling and processing:

The EcUtStatus class is used to describe the operational status of many functions. The values most often reported are "failed" and "ok." But depending upon the application, detailed values could be set and sent. Please refer to the definition of this class (located in /ecs/formal/COMMON/CSCI_Util/src/Logging/EcUtStatus.h) for all possible values.

The EcPoError class defines the basic error types and handling functions for using the EcPoConnectionsRW class (based upon RWDBTool++). The CSS Subscription Server, IOS Server and MSS Order tracking Server use the EcPoConnectionsRW class.

The RWCString is used to store some status value returned by applications.

Integer is used to return some error status by applications. This is used specifically between client and server communications.

Many types of exceptions can be sent and handled by the CSS. These include exceptions sent by Commercial Off The Shelf (COTS) products (such as DCE, RWDBTools++, RWTools++), systems and exceptions defined by individual applications.

4.8.1 The Distributed Computing Configuration Item Software Description

The DCCI CSCI (the software is found in directory DOF) consists mainly of COTS software and hardware providing servers, gateways, and software library services to other CSMS CSCIs. The CSCI is composed of 17 computer software components (CSCs) briefly described here followed by a description of the HWCI.

The CSCI is composed of 17 computer software components (CSCs) briefly described here as processes followed by a description of the HWCI.

1. The Configuration Registry Server (the software is found in directory /ecs/formal/CSS/DOF/src/REGISTRY) provides a single interface to retrieve configuration attribute-value pairs for ECS Applications from the Configuration Registry Database. Configurable run-time parameters for Process Framework-based ECS Applications (including clients and servers) are stored in the Configuration Registry Database. Upon startup, the Process Framework retrieves this information from the Configuration Registry Server for the application.
2. The Machine-to-Machine Gateway (the software is found in directory /ecs/formal/CSS/DOF/src/RELB_GATEWAY/MTMGW) The Machine-to-Machine Gateway (MTMGW) Server provides an automated ordering capability to allow the Science Investigator-Led Processing Systems (SIPS) to reprocess data externally from the ECS. In order to safeguard communications between the MTMGW Server and the SIPS, a SSH (Secure Shell protocol) is employed to secure the line.
3. The CCS Name service group is a COTS software set of Name and Time Services.
    - The ECS Naming Service (the software is found in directory /ecs/formal/CSS/DOF/src/NS/naming) provides a link between clients and the ECS servers they need to communicate with to obtain ECS data and services. Servers register their location information in the ECS Naming Service, independent of physical location. The clients use the ECS Naming Service to find servers based on an operating mode. This is the primary way clients locate servers.
    - The Time Service (the software is found in directory /ecs/formal/CSS/DOF/src/TIME/time) keeps the ECS computer network system clocks synchronized by monitoring and adjusting the operating system clock for each individual host machine in the network. The Time service provides an API to obtain time in various formats. Some applications need to simulate the current time by applying a delta to the current time. The Time Service retrieves time deltas and applies them to the system time.

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.

4. FTP (the software is found in directory /ecs/formal/CSS/DOF/src/FTP) is an Internet standard application for file transfers. It is a client/server model in which the FTP is a client program started by the user while the FTP daemon is the server running on the

target host. 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. The FTP application is used to ingest data into the ECS from remote locations and to distribute data to remote servers for users. The DPL INGEST CSCI uses the FTP application to ingest data from external data providers.

5. FTP Notification (the software is found in directory /ecs/formal/CSS/DOF/src/FTP) provides successful completion notifications for FTP (get) data pulls and (put) data pushes. The DPL INGEST CSCI provides notifications to external data providers of data ingest into the ECS and the OMS CSCIs provide notifications for data distribution from the ECS.

6. The NFS (no physical directory) provides a distributed file sharing system among computers. NFS consists of a number of components, including a mounting protocol and server, a file locking protocol and server, and daemons that coordinate basic file service. A server exports (or shares) a filesystem when it makes the filesystem available for use by other machines in the network. An NFS client must explicitly mount a filesystem before using it.

7. The Filecopy utility (the software is found in directory /ecs/formal/common/CSCI_Util/src/CopyProg) copies files from a specified source location to a specified destination location with options available for data compression. The DPL CSCI uses the Filecopy utility to transfer large files.

8. The mail support group provides electronic mail service.

    - E-mail (the software is found in directory/ecs/formal/CSS/DOF/src/EMAIL/email) 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. E-mail messages are sent among ECS users in the United States and between the ECS.

14. Virtual Terminal (no physical directory) provides the capability for the Operations staff on an ECS platform to remotely log onto another ECS machine.

15. Cryptographic Management Interface (CMI, the software is found in directory /ecs/formal/CSS/DOF/src/AUTHN) provides processes a means for obtaining random passwords and gaining access to Sybase.

16. The Domain Name Service (DNS, the software is found in directory ecs/formal/CSS/NameServer/src) 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. DNS is distributed among a set of servers (name servers); each of which implements DNS by running a daemon called in.named. On the client side, the service is provided through the resolver, which is not a daemon. The resolver resolves user queries by needing the address of at least one name server (provided in a configuration file parameter). Each domain must have at least two kinds of DNS servers (a primary and secondary server) maintaining the data corresponding to the domain. The primary server obtains the master

copy of the data from disk when it starts up the in.named. The primary server delegates authority to other servers in or outside of the domain. The secondary server maintains a copy of the data for the domain. When the secondary server starts in.named, the server requests all data for the given domain from the primary server. The secondary server checks periodically with the primary server for updates. DNS namespace has a hierarchical organization consisting of nested domains like directories. The DNS namespace consists of a tree of domains. See Figure 4.8-33 for an illustration of the domain tree hierarchy.

17. The Infrastructure Library provides a set of services including the following.

    - Process Framework (PF) (the software is located in directory /ecs/formal/CSS/DOF/src/PF/pf**)**: The PF is a software library of services, which provides a flexible mechanism (encapsulation) for the ECS Client and Server applications to transparently include specific ECS infrastructure features from the 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 (CCS Middleware naming service), 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 EMD project from the client/server perspective. Then the required capabilities are allocated for each respective process level and type. PF-based ECS applications use Process Framework to read in their configuration information at startup. PF-based servers use Process Framework to initialize themselves as a CCS Middleware server and put it in a listen state to begin to accept requests from appropriate clients.

    - Universal References (the software is found in directory /ecs/formal/ COMMON/CSCI_UR/src/UR/framework): Universal References (URs) provide applications and users a system wide mechanism for referencing ECS data and service objects. Manipulating logical entities represented at run time as C++ objects in virtual memory performs ECS functions. Users and applications require references to the logical entities beyond the effective computational time to keep the objects in memory. Therefore, applications and users are given URs to these objects. Once an UR is made for an object, the object can be disposed of and later reconstituted from the UR. URs take up a small fraction of the space to keep in memory and can be externalized into an ASCII string, which an end user can manage. URs have the capability of re-accessing and/or reconstituting the object into memory as needed.

Therefore, the object does not have to remain in memory, and can if appropriate, be written to a secondary storage system, like a database. While the UR mechanism guarantees reliable data externalization and internalization, the content of each type of UR is application specific. Only the object (this is referred to as the "UR Provider") that initially provides the UR is allowed to access and understand its content. URs are strongly typed to enforce appropriate access control to internal data both at compile time and during run time. Since URs are typed and have object specific data in them, separate UR object classes exist for each UR Provider class referred to. All of these UR classes use the mechanisms provided by the UR framework.

- Event Logging (the software is found in directoryLOGGING): Event logging is the capability of recording events into files and provides a convenient way to generate and report detailed events. All ECS CSCIs use event and error logging as an audit trail for all transactions that occur during the ECS data processing and distributing.

- Server Locator (the software is found in directory /ecs/formal/CSS/DOF/src/NS/service_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. The Server Locator is used in ECS in any client-server CCS Middleware-based communication.

- Failure Recovery Framework (the software is found in directory /ecs/formal/CSS/DOF/src/FH): The Failure Recovery Framework provides a general-purpose fault recovery routine enabling client applications to reconnect with servers after the initial connection is lost. This is accomplished through the CCS Naming Service, through which the Failure Recovery Framework can determine whether a server is listening. The Failure Recovery Framework provides a default and configurable amount of retries and duration between retries. This fault recovery takes effect for each attempt by the client to communicate with the server for all applications that employ the Failure Recovery Framework.

- EcPo Connections (the software is found in directory /ecs/formal/ COMMON/CSCI_DBWrapper): A suite of classes providing a basic set of database connection management methods and an error handling mechanism for database users, which is found in the DBWrapper directory of the Infrastructure Library Group.

- Time Service (the software is found in directory /ecs/formal/CSS/DOF/src/TIME/time): the class providing the structured time information and get RogueWave type of time information.

- 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.

4.8.1.1 Configuration Registry Server Software Description

4.8.1.1.1 Configuration Registry Server Functional Overview

The Configuration Registry Server provides an interface to retrieve configuration attribute-value pairs and another interface to retrieve distribution options for ECS Servers from the Configuration Registry Database, via a Sybase ASE. The Configuration Registry Server maintains an internal representation of the tree in which configuration attribute-value pairs and distribution options are stored. General configuration parameters used by many servers are stored in higher nodes in the tree. Parameters specific to a single ECS Server are contained in the leaf nodes of the tree.

The Configuration Registry Server not only accepts queries to the Configuration Registry Database with a configuration path and returns a list of attribute-value pairs, but also accepts queries of distribution options to the Configuration Registry Database with an ESDT short name and version and returns a hierarchical list of attributes. A wild-card character may be specified as the last element in the path to retrieve all attributes in the sub-tree specified. Each Configuration Registry Server is MODE specific, with multiple Registry Servers running in a mode to provide redundancy.

4.8.1.1.2 Configuration Registry Server Context

Figure 4.8-2 is the Configuration Registry Server context diagrams. Table 4.8-2 provides descriptions of the interface events in the Configuration Registry Server context diagrams.

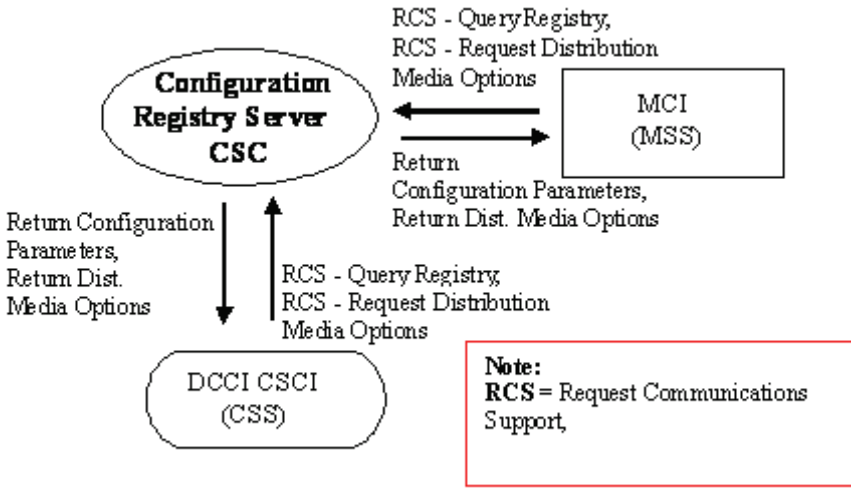


Figure 4.8-2. Configuration Registry Server Context Diagram

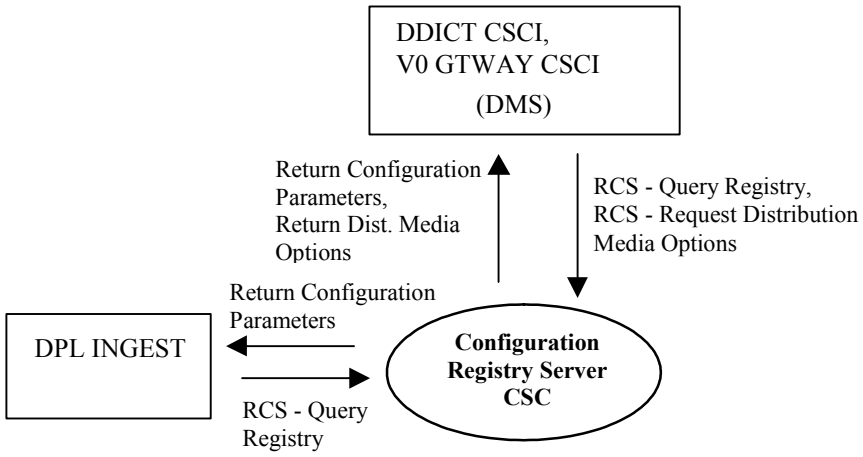


Figure 4.8-2. Configuration Registry Server Context Diagram (cont.)

Table 4.8-2. Configuration Registry Server Interface Events (1 of 2)

Event	Interface Event Description
RCS - Request Distribution Media Options	The **MCI**, **SDSRV** and other **DCCI CSCI CSCs** query the Configuration Registry Server for configuration parameters. The ECS Servers pass in an ESDT short name and version. The Registry Server uses this information as a starting point in the tree and returns all distribution options associated with it.
RCS - Query Registry	Upon startup, the **MCI**, **SDSRV** other **DCCI CSCI CSCs** query the Configuration Registry Server for configuration parameters and their respective value(s). The ECS Servers pass in a path that corresponds to a sub-tree in the MODE configuration value tree maintained by the server. The Registry Server uses this path as a starting point in the tree and returns all parameters and their associated values in the sub-tree below it.
Return Configuration Parameters	The Configuration Registry CSC returns the requested configuration parameters to the **SDSRV, MCI and DCCI,** CSCIs.
Return Dist. Media Options	The Configuration Registry CSC returns the requested distribution media options to the **SDSRV, MCI and DCCI** CSCIs.

Table 4.8-2. Configuration Registry Server Interface Events (2 of 2)

Event	Interface Event Description
RCS - Request Distribution Media Options	The **DDICT and V0 GTWAY CSCIs** query the Configuration Registry Server for configuration parameters. The ECS Servers pass in an ESDT short name and version. The Registry Server uses this information as a starting point in the tree and returns all distribution options associated with it.
RCS - Query Registry	Upon startup, the **DDICT, V0 GTWAY, and DPL INGEST CSCIs** query the Configuration Registry Server for configuration parameters and their respective value(s). The ECS Servers pass in a path that corresponds to a sub-tree in the MODE configuration value tree maintained by the server. The Registry Server uses this path as a starting point in the tree and returns all parameters and their associated values in the sub-tree below it.
Return Configuration Parameters	The Configuration Registry CSC returns the requested configuration parameters to the **DDICT, V0 GTWAY, and DPL INGEST** CSCIs.
Return Dist. Media Options	The Configuration Registry CSC returns the requested distribution media options to the **DDICT and V0 GTWAY** CSCIs.

4.8.1.1.3 Configuration Registry Server Architecture

Figure 4.8-3 is the Configuration Registry Server architecture diagram. The diagram shows the events sent to the Configuration Registry Server process and the events the Configuration Registry Server process sends to other processes.

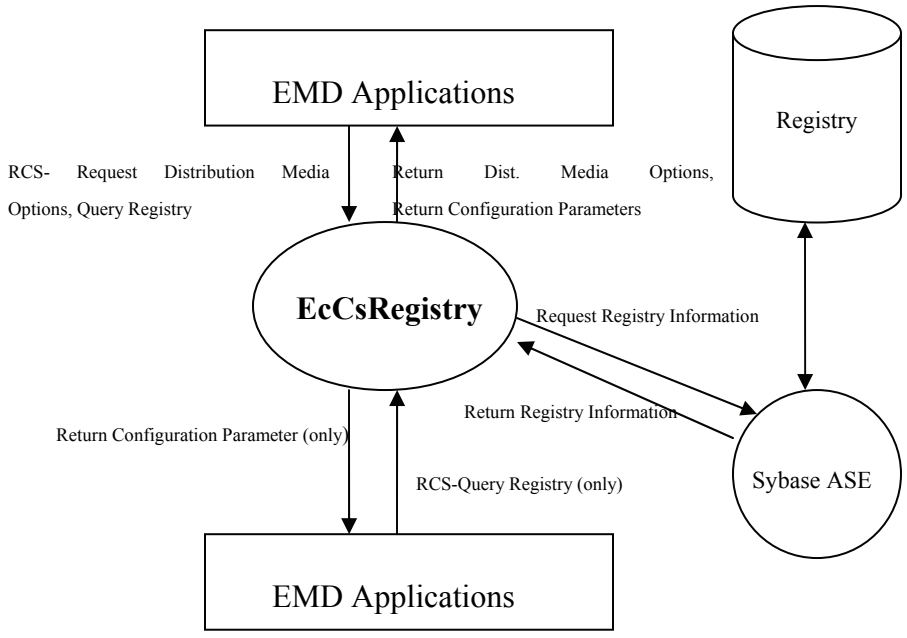


Figure 4.8-3. Configuration Registry Server Architecture Diagram

4.8.1.1.4 Configuration Registry Server Process Descriptions

Table 4.8-3 provides a description of the processes in the Configuration Registry Server architecture diagram.

Table 4.8-3. Configuration Registry Server Processes

Process	Type	Hardware CI	COTS / Developed	Functionality
EcCsRegistry	Server	OMSHW	Developed	The Configuration Registry Server provides an interface to retrieve configuration attribute-value pairs (the returned attribute-value pairs are stored in cache memory on the PF client side). The Configuration Registry Server provides an interface to retrieve distribution options for ECS Servers from the Configuration Registry Database, via a Sybase ASE. The Configuration Registry Server not only accepts queries to the Configuration Registry Database with a configuration path and returns a list of attribute-value pairs, but also accepts queries of distribution options to the Configuration Registry Database with an ESDT short name and version and returns a list of attributes. A wild-card character may be specified as the last element in the path to retrieve all attributes in the sub-tree specified. The Configuration Registry Server provides another interface to retrieve external data subsetters for Synergy.

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.8.1.1.5 Configuration Registry Server Process Interface Descriptions

Table 4.8-4 provides descriptions of the interface events shown in the Configuration Registry Server architecture diagram.

Table 4.8-4. Configuration Registry Server Process Interface Events (1 of 2)

Event	Event Frequency	Interface	Initiated by	Event Description
RCS - Query Registry	One per client request	*Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	*Processes:* EcDsScienceDataServer, EcDsHdfEosServer, EcDmV0ToEcsGateway, EcCsMtMGateway	An **ECS application** (process) sends a query request to the Configuration Server to retrieve a list of attribute-value pairs (configuration parameters) needed by the application.
Return Configuration Parameters	One set per request	*Processes:* EcDsScienceDataServer, EcDsHdfEosServer, EcDmV0ToEcsGateway, EcCsMtMGateway	*Process:* EcCsRegistry *Library:* EcCsRegistry Class: EcRgRegistryServer_C	The EcCsRegistry returns the attribute-value pairs (configuration parameters) to the various **ECS applications** (processes) upon request.
Retrieve Registry Information	Per client request	Registry Database	Sybase ASE (COTS)	The **Sybase ASE** receives the request and retrieves the necessary attribute-value pairs and returns them to the EcCsRegistry.
Request Registry Information	Per client request	Sybase ASE (COTS)	*Process:* EcCsRegistry *Class:* EcRgRegistryServer_S	The Configuration Server sends the request to the **Sybase ASE** to retrieve the attribute-value pairs.
Return Registry Information	Per client request	*Process:* EcCsRegistry Class: EcRgRegistryServer_S	Sybase ASE (COTS)	The Configuration Server receives the registry information (attribute-value pairs) from the **Sybase ASE**.

Table 4.8-4. Configuration Registry Server Process Interface Events (2 of 2)

Event	Event Frequency	Interface	Initiated by	Event Description
RCS - Query Registry (only)	One per client request	*Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	*Processes:* EcDmDictServer, EcCsEmailParser, EcMsAcRegUserSrvr, EcMsAcOrderSrvr	An **ECS application** (process) sends a query request to the Configuration Server to retrieve a list of attribute-value pairs (configuration parameters) needed by the application.
Return Configuration Parameters (only)	One set per request	*Processes:* EcDmDictServer, EcCsEmailParser, EcMsAcRegUserSrvr, EcMsAcOrderSrvr	*Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	The EcCsRegistry returns the attribute-value pairs (configuration parameters) to the various **ECS applications** (processes) upon request.
RCS - Request Distribution Media Options	One per client request	*Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	*Processes:* EcDsScienceDataServer, EcDsHdfEosServer, EcDmV0ToEcsGateway, EcCsMtMGateway	An **ECS application** (process) sends a query of distribution options to the Configuration Server to retrieve a list of distribution options.
Return Dist. Media Options	One set per request	*Processes:* EcDsScienceDataServer, EcDsHdfEosServer, EcDmV0ToEcsGateway, EcCsMtMGateway	*Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	The EcCsRegistry returns the distribution media options (tape) to the various **ECS applications** (processes) upon request.

4.8.1.1.6 Configuration Registry Server Data Stores

The Configuration Registry Server uses a Sybase ASE database for its persistent storage. The following is a brief description of the types of data contained in the database:

- **Mode**: This data store contains the list of modes and a description of the purpose of the mode.

- **Node**: This data store contains information that describes each node in the tree.
- **NodeContact**: This data store contains the information of the person who is responsible for the information contained in each node of the tree.
- **Attribute tree**: This data store contains a list of tree names and a description of each tree.
- **Attribute**: This data store contains a description of each attribute whose value is assigned to a particular node.
- **AttributeValidEnum**: This data store contains enumerated string values for attributes of enumerated types.
- **AccessControlList**: This data store contains the access control information for each node.
- **ConfiguredValue**: This data store contains the value for the parameter stored in a node, and associated information.
- **ConfigurationManagementContact**: This data store contains a list of configuration management contacts for information stored in the Configuration Registry.

Table 4.8-5 provides descriptions of the data found in the separate Sybase ASE data stores used by the Configuration Registry Server. More detailed information on these data stores can be found in the Configuration Registry Database Design and Schema Specifications for the EMD Project.

Table 4.8-5. Configuration Registry Server Data Stores (1 of 2)

Data Store	Type	Functionality
Mode	Sybase	This data store contains the list of modes and a description of the purpose of the mode. It also contains a mapping of the mode to the tree name.
AttributeTree	Sybase	This data store contains a list of tree names and a description of each tree.
Node	Sybase	This data store contains information that describes each node in the tree. This information includes a NodeID, the tree name to which it belongs, the node name, its parent NodeID, the node type, and a node description.
NodeContact	Sybase	This data store contains the information of the person who is responsible for the information contained in each node of the tree. It includes the NodeID, and the FirstName, LastName, Org (organization), and Email address of the person responsible.
AccessControlList	Sybase	This data store contains the access control information for each node. It includes the NodeID, an AclSequenceNumber, AclType, AclUser, AclGroup, and Create, Read, Update, and Delete flags.
Attribute	Sybase	This data store contains a description of each attribute whose value is assigned to a particular node. It lists the attribute type, minimum and maximum values, and the NodeID.

Table 4.8-5. Configuration Registry Server Data Stores (2 of 2)

Data Store	Type	Functionality
AttributeValidEnum	Sybase	This data store contains enumerated string values for attributes of enumerated type. It includes a string name for each enumerated value, a description of the value, and a NodeID.
ConfiguredValue	Sybase	This data store contains the value for the parameter stored in a node, and associated information. It includes the NodeID, DataType, and TimeStamp of last change, Comment for the change, Float, Integer, or String value, ValueVersion number, and userid of the user who made the change.
ConfigurationManagementContact	Sybase	This data store contains a list of configuration management contacts for information stored in the Configuration Registry.

4.8.1.2 Machine-to-Machine Gateway Server Software Description

4.8.1.2.1 Machine-to-Machine Gateway Server Functional Overview

The Machine-to-Machine Gateway (MTMGW) Server provides an automated ordering capability to allow the Science Investigator-Led Processing Systems (SIPS) to reprocess data externally from the ECS. In order to safeguard communications between the MTMGW Server and the SIPS, a SSH (Secure Shell protocol) is employed to secure the line. A SIPS user account is set up manually with SSH encryption keys on both the local SIPS host and a DAAC host. The SIPS Operations Staff initiates the key exchange using the SSH via a script.

The MTMGW Server is capable of receiving inventory search requests from the SIPS, submitting search requests to the DSS (SDSRV CSCI) for the selected metadata whose type information is obtained from the DMS (DDICT CSCI), and returning search results back to the SIPS.

The MTMGW Server is capable of accepting order requests based on UR or GranuleID from the SIPS, forwarding the corresponding acquire request to the Order Management Service (OMS) when configured to submit orders to the OMS or DSS (SDSRV CSCI) when configured to submit orders to SDSRV, and returning a response message to the SIPS indicating the status of order.

The MTMGW Server is capable of dealing with integrated search and order requests from the SIPS. In this case, no search result is returned to the SIPS. Instead, the search results are staged inside the MTMGW Server for further order processing. A response message is returned to the SIPS indicating the status of orders.

In the last two cases above, the ECS order tracking service keeps track of the MTMGW Server orders based on the user profile ID provided by the SIPS.

The MTMGW Server supports multiple concurrent servers; each server is independently configured by the DAAC. Each MTMGW Server supports multiple concurrent requests. DAAC operations can configure the maximum number of concurrent requests for each server.

4.8.1.2.2 Machine-to-Machine Gateway Server Context

Figure 4.8-4 is the Machine-to-Machine Gateway Server context diagram. Table 4.8-6 shows descriptions of the interface events in the Machine-to-Machine Gateway Server context diagram.

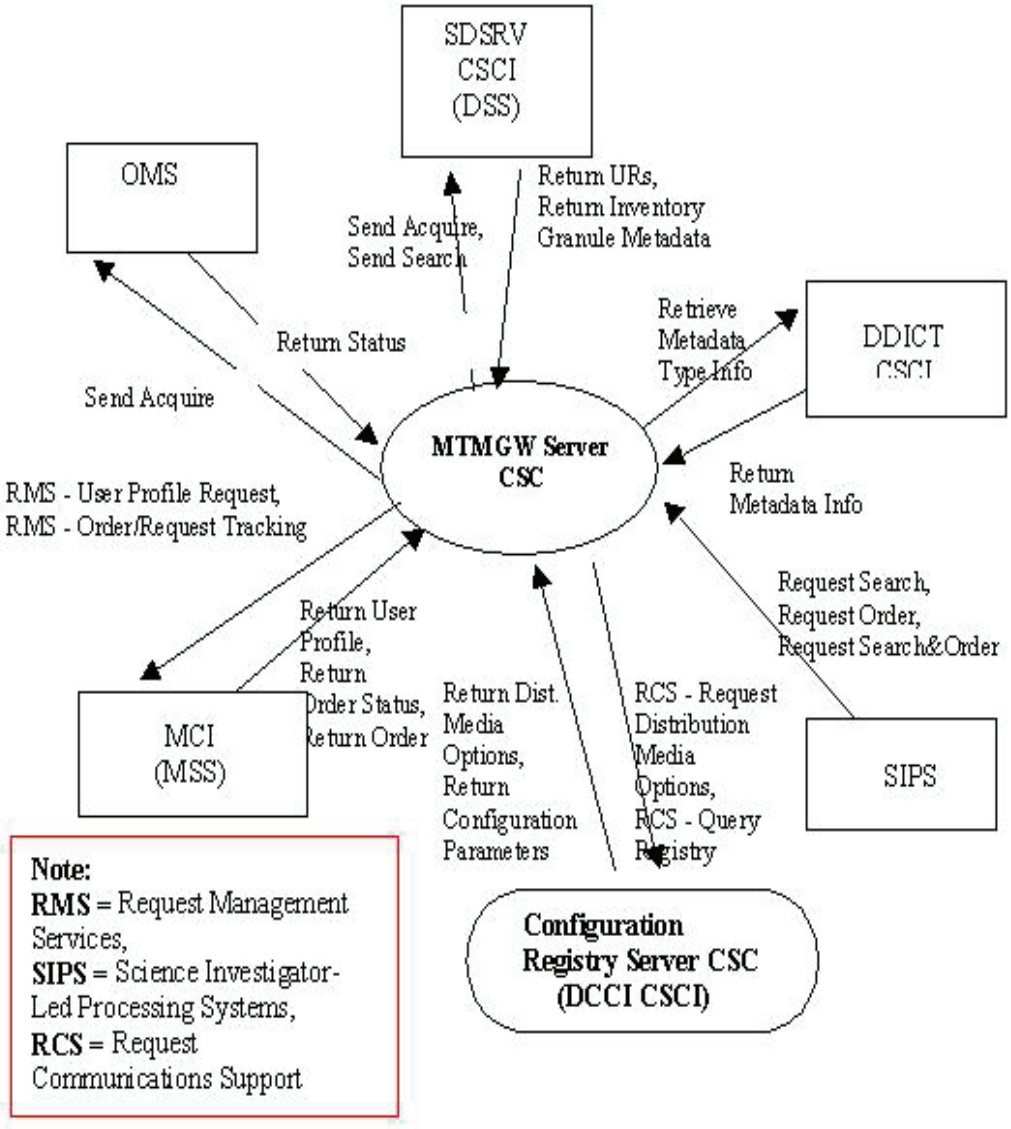


Figure 4.8-4. Machine-to-Machine Gateway Server Context Diagram

Table 4.8-6. Machine-to-Machine Gateway Server Interface Event (1 of 2)

Event	Interface Event Description
Send Acquire	The MTMGW supports a configuration parameter that specifies the manner of order submission to be either to the Order Management Service (OMS) or the ECS Science Data Server (SDSRV). When the submission parameter is configured to OMS, the MTMGW Server CSC submits acquire requests to the **OMS CI** for order or search & order requests from the SIPS, When the submission parameter is configured to SDSRV, the MTMGW Server CSC submits acquire requests to the **SDSRV CSCI** for order or search & order requests from the SIPS.
Send Search	The MTMGW Server CSC sends search requests to the **SDSRV CSCI** on behalf of the SIPS.
Retrieve Metadata Type Info	The MTMGW Server CSC retrieves metadata type information from the **DDICT CSCI** pertaining to search or search & order requests from the SIPS.
Return Metadata Info	The MTMGW Server CSC receives metadata information from the **DDICT CSCI**.
Request Search	The MTMGW Server CSC receives search requests from the **SIPS for ECS products**.
Request Order	The MTMGW Server CSC receives order requests from the **SIPS for ECS products**.
Request Search&Order	The MTMGW Server CSC receives integrated search and order requests from the **SIPS for ECS products**.
Request Communications Support (RCS)	The **DCCI CSCI** provides a library of services available to each CSMS CSCI/CSC. The CSCI services required to perform specific assignments are requested from the DCCI CSCI/CSC. These services include: • CCS Middleware Support • Database Connection Services • File Transfer Services • Network & Distributed File Services • Bulk Data Transfer Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested attribute-value pairs from the Configuration Registry • Request Distribution Media Options from the Configuration Registry
Return Dist. Media Options	The MTMGW Server CSC receives the distribution media options from the **Configuration Registry Server** CSC.
Return Configuration Parameters	The MTMGW Server CSC receives the requested configuration parameters from the **DCCI CSCI (Configuration Registry Server)**.

Table 4.8-6. Machine-to-Machine Gateway Server Interface Event (2 of 2)

Event	Interface Event Description
Request Management Services	The **MCI** provides a basic management library of services to the CSCIs, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes: • **System startup and shutdown -** Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625). • **User Profile Request** – The **MCI** provides requesting CSCIs with User Profile parameters such as distribution priority and shipping address to support their processing activities. • **Order/Request Tracking** – The **MCI** provides an order tracking service to requesting CSCIs for creating and tracking order or search and order requests from the SIPS.
Return User Profile	The MTMGW Server CSC receives user profile information from the **MCI** to authenticate a user.
Return Order Status	The MTMGW Server CSC receives the status of an order for the requested ECS product from the **MCI**.
Return Order	The MTMGW Server CSC receives the product order object from the **MCI**.
Return URs	The MTMGW Server CSC receives Earth Science Data Type (ESDT) Universal References (URs) for the granules from the OMS or **SDSRV CSCI**.
Return Inventory Granule Metadata	The MTMGW Server CSC receives the inventory granule metadata identifying the scene within the granule from the **SDSRV CSCI** based on an inventory search request.

4.8.1.2.3 Machine-to-Machine Gateway Server Architecture

Figure 4.8-5 is the Machine-to-Machine Gateway Server architecture diagrams. The diagrams show the events sent to the Machine-to-Machine Gateway Server process and the events the Machine-to-Machine Gateway process sends to other processes.

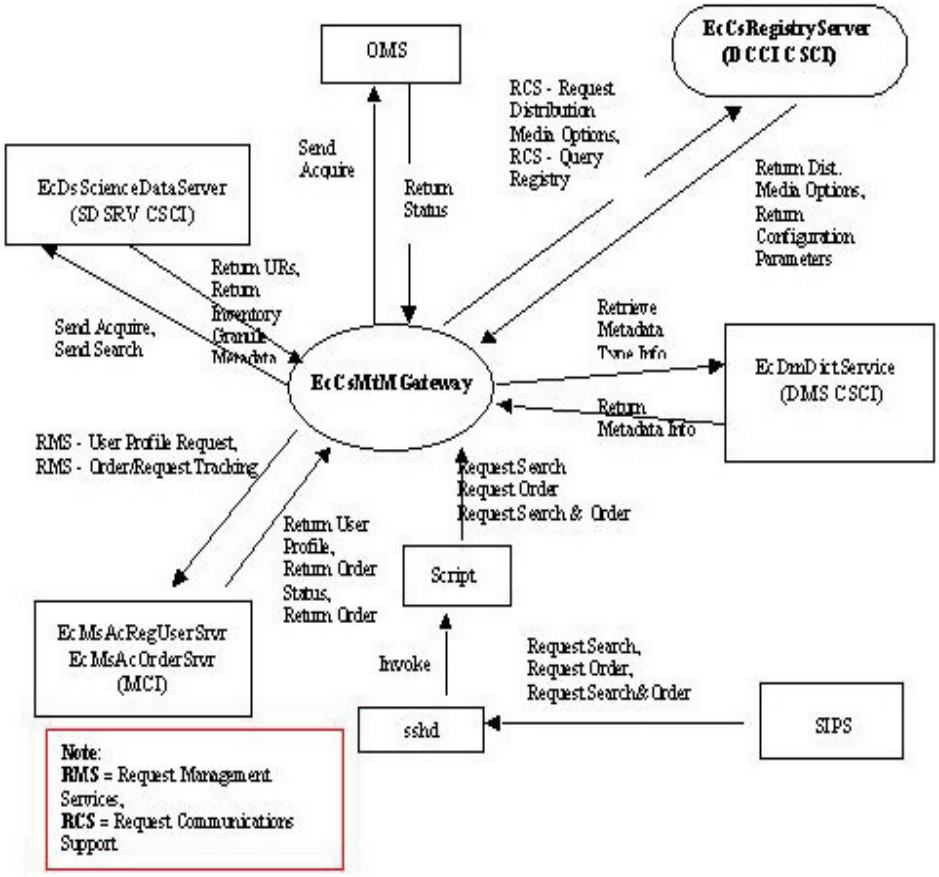


Figure 4.8-5. Machine-to-Machine Gateway Server Architecture Diagram

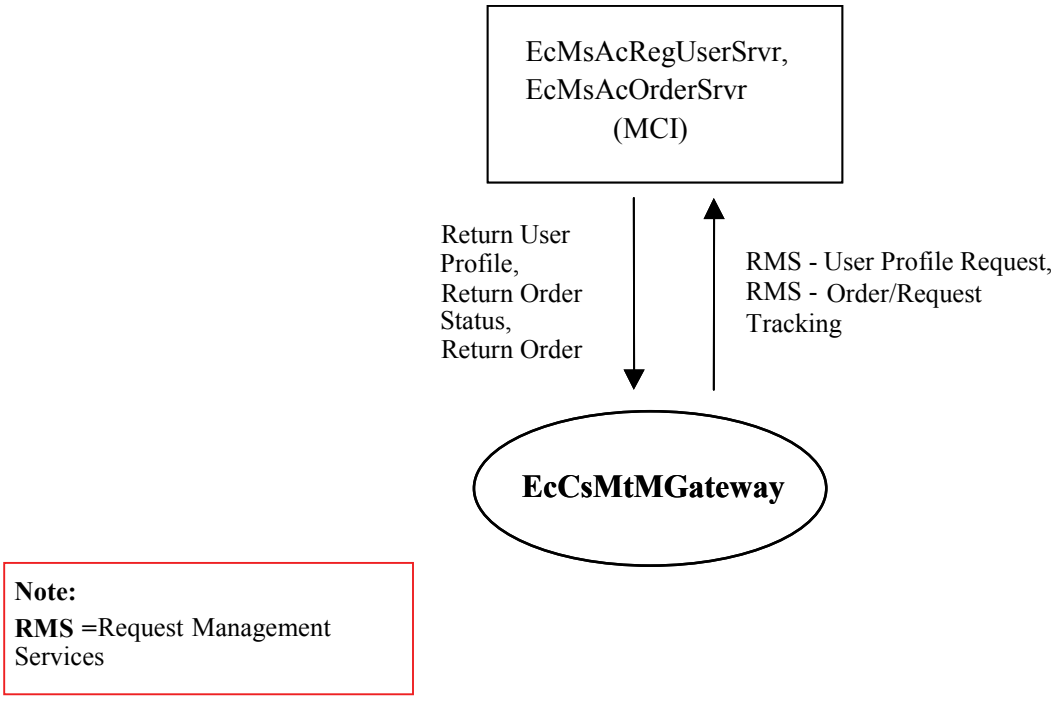


Figure 4.8-5. Machine-to-Machine Gateway Server Architecture Diagram (cont.)

4.8.1.2.4 Machine-to-Machine Gateway Server Process Descriptions

Table 4.8-7 provides a description of the processes in the Machine-to-Machine Gateway Server architecture diagrams.

Table 4.8-7. Machine-to-Machine Gateway Server Processes

Process	Type	Hardware CI	COTS / Developed	Functionality
EcCsMtMGateway	Server	INTHW	Developed	The Machine-to-Machine Gateway Server receives search, order, and integrated search & order requests from the SIPS via a sshd daemon. To fulfill search requests, the Machine-to-Machine Gateway Server uses the metadata type information retrieved from the EcDmDictServer process as the search request input to the EcDsScienceDataServer process and returns the search results back to the SIPS via the sshd. To process orders, the Machine-to-Machine Gateway Server process sends user profile requests to the EcMsAcRegUserSrvr process (using a user profile ID sent as part of the order request) to get distribution priority and shipping information from the user profile. The Machine-to-Machine Gateway Server process then sends order tracking requests to the EcMsAcOrderSrvr process to create an order, and submits acquire requests to the EcOmOrderManager when configured to submit orders to OMS or EcDsScienceDataServer when configured to submit orders to SDSRV. Finally, the Machine-to-Machine Gateway Server returns a response message back to the SIPS via the sshd2. For search & order requests, the Machine-to-Machine Gateway Server process integrates the above two cases into one by staging the search results of the first step without returning to the SIPS and making order requests immediately.

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.8.1.2.5 Machine-to-Machine Gateway Server Process Interface Descriptions

Table 4.8-8 provides descriptions of the interface events shown in the Machine-to-Machine Gateway Server architecture diagram.

Table 4.8-8. Machine-to-Machine Gateway Server Process Interface Events (1 of 6)

Event	Event Frequency	Interface	Initiated by	Event Description
Request Communications Support	One service per request	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy *Library (Common):* EcUr *Class:* EcUrServerUR *Library:* event *Class:* EcLgErrorMsg *Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	***Process:*** EcCsMtMGateway *Library:* EcCsMtMGateway *Class:* EcCsMtMManagedSrv	The **DCCI CSCI** provides a library of services available to each CSMS CSCI/CSC. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • File Transfer Services • Network & Distributed File Services • Bulk Data Transfer Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry • Request Distribution Media Options from the Configuration Registry

Table 4.8-8. Machine-to-Machine Gateway Server Process Interface Events (2 of 6)

Event	Event Frequency	Interface	Initiated by	Event Description
Retrieve metadata type info	One or more client request	*Processes:* EcCsMtMGateway, EcDmDictServer *Libraries:* DmAsGwCommon, Common, DmGwV0Util, EcDmDClient, DmDdMsg *Class:* DmDdClRequestServer	*Process:* EcCsMtMGateway *Class:* EcCsMtMAttributeDict	The EcCsMtMGateway retrieves metadata type information from the **EcDmDictServer** based upon the qualifying metadata contained in the requests sent by the SIPS via a CCS Middleware call.
Return Metadata Info	Per search or order request	*Process:* EcCsMtMGateway *Class:* EcCsMtMAttributeDict	*Processes:* EcCsMtMGateway, EcDmDictServer *Libraries:* DmAsGwCommon, Common, DmGwV0Util, EcDmDClient, DmDdMsg *Class:* DmDdClRequestServer	The EcCsMtMGateway receives metadata information from the **EcDmDictServer**.
Request Search	One per client	*Process:* EcCsMtMGateway sshd *Script:* mtmsearch	*Process:* sshd2 (COTS)	The SIPS send search requests to the sshd. The **sshd2** decrypts the request and forwards it to the EcCsMtMGateway Server.
Request Order	One per client	*Process:* EcCsMtMGateway sshd *Script:* mtmorder	*Process:* sshd2 (COTS)	The SIPS send order requests to the sshd. The **sshd2** decrypts the request and forwards the request to the EcCsMtMGateway Server.
Request Search & Order	One per client	*Process:* EcCsMtMGateway sshd *Script:* mtmsearchorder	*Process:* sshd2 (COTS)	The SIPS send search & order requests to the sshd. The **sshd2** decrypts the request and forwards the request to the EcCsMtMGateway Server.

Table 4.8-8. Machine-to-Machine Gateway Server Process Interface Events (3 of 6)

Event	Event Frequency	Interface	Initiated by	Event Description
Invoke Command	One or more per client	*Process:* sshd2 *Scripts:* mtmsearch, mtmorder, mtmsearchorder	Process: sshd2 (COTS)	The sshd2 daemon process invokes the sshd2 command on the SIPS side to send an invoke command to start the scripts on the ECS side of the interface to accept and process requests.
Send Acquire	One or more client request	*Process:* *EcOmOrderManager or*EcDsScienceDataServer *Libraries:* DsCl, DsCn, DsSh, DsGe, DsSr, DsDe2, Gl *Class:* DsClESDTReferenceCollector	*Process:* EcCsMtMGateway *Classes:* EcCsMtMDataServerMgr, EcCsMtMECSOrderProxy, EcCsMtMOrderImp, EcCsMtMECSSearchOrderProxy, EcCsMtMSearchOrderImp	The EcCsMtMGateway Server sends acquire requests based upon the created order constructed with the qualifying metadata contained in requests received from the SIPS to the EcOmOrderManager when configured the order submission to OMS or **EcDsScienceDataServer** via CCS Middleware calls when configured the order submission to SDSRV.
Send Search	One or more client request	*Process:* EcDsScienceDataServer *Libraries:* DsCl, DsCn, DsSh, DsGe, DsSr, DsDe2, Gl *Class:* DsClESDTReferenceCollector	*Process:* EcCsMtMGateway *Class:* EcCsMtMECSSearchProxy, EcCsMtMSearchImp, EcCsMtMSdsrvMgr, EcCsMtMDataServerMgr	The EcCsMtMGateway sends search requests, constructed based upon the qualifying metadata information received from the SIPS requests, to the **EcDsScienceDataServer** via CCS Middleware calls for inventory searches.

Table 4.8-8. Machine-to-Machine Gateway Server Process Interface Events (4 of 6)

Event	Event Frequency	Interface	Initiated by	Event Description
Return URs	One or more client request	*Process:* EcCsMtMGateway *Classes:* EcCsMtMDataServerMgr, EcCsMtMECSOrderProxy, EcCsMtMOrderImp, EcCsMtMECSSearchOrderProxy, EcCsMtMSearchOrderImp	*Process:* EcDsScienceDataServer *Libraries:* DsCl, DsSh *Classes:* DsClRequest, DsClCommand, DsClESDTReferenceCollector	The EcCsMtMGateway receives Earth Science Data Type (ESDT) Universal References (URs) for the granules from the **EcDsScienceDataServer**.
Return Inventory Granule Metadata	One or more client request	*Process:* EcCsMtMGateway *Class:* EcCsMtMECSSearchProxy, EcCsMtMSearchImp, EcCsMtMSdsrvMgr, EcCsMtMDataServerMgr	*Process:* EcDsScienceDataServer *Libraries:* DsCl, DsSh *Classes:* DsClESDTReference, DsClESDTReferenceCollector	The EcCsMtMGateway receives the inventory granule metadata identifying the scene within the granule based on an inventory search request sent to the **EcDsScienceDataServer**.
Return Dist. Media Options	One set of options per request	*Process:* EcCsMtMGateway *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	*Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	The EcCsMtMGateway receives the distribution media options from the **EcCsRegistry**.
Return Configuration Parameters	One per configuration registry query	Process: EcCsMtMGateway Library: EcCsRegistry Class: EcRgRegistryServer_C	Process: EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	The **EcCsRegistry** returns the requested configuration parameters to the EcCsMtMGateway.

Table 4.8-8. Machine-to-Machine Gateway Server Process Interface Events (5 of 6)

Event	Event Frequency	Interface	Initiated by	Event Description
Request Management Services (RMS)	One per service request	N/A	N/A	The **EcMsAcRegUserSrvr** and **EcMsAcOrderSrvr** provide a basic management library of services to the processes, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes the items below.
RMS (cont.)	At system startup or shutdown and for restarts	*Process:* EcCsMtMGateway	DAAC unique startup scripts	**System startup and shutdown -** Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625).
RMS (cont.)	One or more per client	*Processes:* EcCsMtMGateway, EcMsAcRegUserSrvr *Libraries:* MsAcClnt, MsAcComm	*Process:* EcCsMtMGateway *Class:* EcCsMtMAcctSrvMgr	**User Profile Request** - The **EcMsAcRegUserSrvr** provides requesting processes with User Profile parameters such as e-mail address and shipping address to support their processing activities.
RMS (cont.)	One or more per client	*Processes:* EcCsMtMGateway, EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	*Process:* EcCsMtMGateway *Class:* EcCsMtMAcctSrvMgr	**Order/Request Tracking** - The MCI provides an order tracking service (via the **EcMsAcOrderSrvr**) to requesting CSCIs for creating or tracking order or search and order requests from the SIPS.

Table 4.8-8. Machine-to-Machine Gateway Server Process Interface Events (6 of 6)

Event	Event Frequency	Interface	Initiated by	Event Description
Return User Profile	One per user request	*Process:* EcCsMtMGateway *Class:* EcCsMtMAcctSrvMgr	*Processes:* EcCsMtMGateway, EcMsAcRegUserSrvr *Libraries:* MsAcClnt, MsAcComm	The EcCsMtMGateway receives user profile information from the **EcMsAcRegUserSrvr**.
Return Order Status	One per user request	*Process:* EcCsMtMGateway *Class:* EcCsMtMAcctSrvMgr	*Processes:* EcCsMtMGateway, EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	The EcCsMtMGateway receives the status of an order from the **EcMsAcOrderSrvr**.
Return Order	One per user request	*Process:* EcCsMtMGateway *Class:* EcCsMtMAcctSrvMgr	*Processes:* EcCsMtMGateway, EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	The EcCsMtMGateway receives the product order object from the **EcMsAcOrderSrvr**.

4.8.1.2.6 Machine-to-Machine Gateway Server Data stores

Data stores are not applicable for the Machine-to-Machine Gateway Server.

4.8.1.3 CCS Middleware Support Group Description

The CCS Middleware support group consists of the CCS Name Server.

4.8.1.3.1 CCS Middleware Functional Overview

The CCS Name Server of the CSS enables clients to locate and communicate with the various ECS servers. The ECS servers register their location information into the CCS Name Server (EcCsIdNameServer) independent of the server's physical location. Servers registering in the EcCsIdNameServer are available to be accessed by other application clients. Clients use the remote service name and the ECS operating mode to find the server of interest.

4.8.1.3.2 CCS Middleware Context

Figure 4.8-6 is the CCS Middleware context diagram. Table 4.8-9 provides descriptions of the interface events shown in the CCS Middleware context diagram.

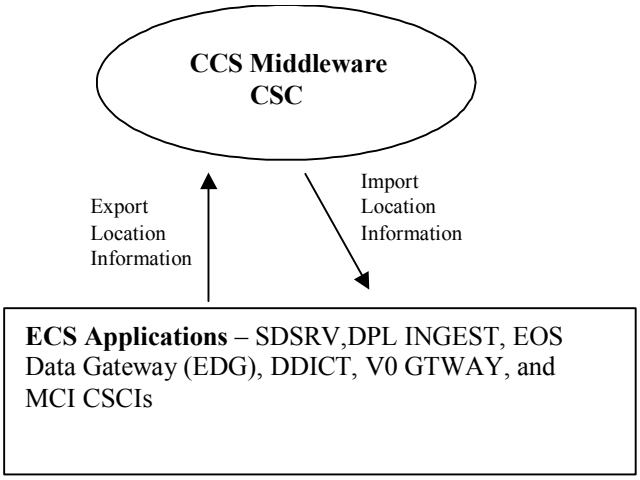


Figure 4.8-6. CCS Middleware Context Diagram

Table 4.8-9. CCS Middleware Interface Events

Event	Interface Event Description
Import location information (binding information)	An **ECS application** requests server location information from the CCS Name Server and saves the information in its local cache via the CCS Name Server client proxy component.
Export location information (binding information)	The CCS Middleware CSC stores physical and logical location information received from **ECS Applications** in the CCS Name Server via the CCS Name Server client proxy component.

4.8.1.3.3 CCS Middleware Architecture

Figure 4.8-7 is the CCS Middleware support group architecture diagram. The diagram shows the events sent to the CCS Middleware process and the events the CCS Middleware process send to other processes.

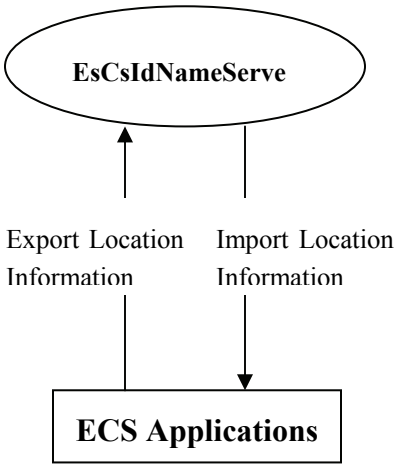


Figure 4.8-7. CCS Middleware Architecture Diagram

4.8.1.3.4 CCS Middleware Process Descriptions

Table 4.8-10 provides descriptions of the processes shown in the CCS Middleware architecture diagram. ECS applications store server location information on the CCS Name Server (EcCsIdNameServer).

Table 4.8-10. CCS Middleware Processes

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcCsIdNameServer	Internal	OMSHW	COTS	Stores server location information and provides interfaces for storing and retrieving the location information.

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.8.1.3.5 CCS Middleware Process Interface Descriptions

Table 4.8-11 provides descriptions of the interface events shown in the CCS Middleware architecture diagram.

Table 4.8-11. CCS Middleware Process Interface Events

Event	Event Frequency	Interface	Initiated By	Event Description
Import location information	One per server	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	*Processes:* ECS Applications (All servers identified in the architecture diagram.) *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	An **ECS application** retrieves server location information from the EcCsIdNameServer via the CCS Name Server client component in the ECS application.
Export location information	One per server	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	*Processes:* ECS Applications (All processes identified in the architecture diagram.) *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	The **ECS application** places physical and logical location information in the EcCsIdNameServer via the CCS Name Server client component in the ECS application.

4.8.1.3.6 CCS Middleware Data Stores

Table 4.8-12 provides a description of the data store shown in the CCS Middleware architecture diagram.

Table 4.8-12. CCS Middleware Data Stores

Data Store	Type	Functionality
StringId	Sybase	This data store contains the ECS Mode information.
Service	Sybase	This data store contains the service name an ECS application server listens on.
Host	Sybase	This data store contains the host name an ECS application server runs on.
ProcessId	Sybase	This data store contains the process id an ECS application server runs with.
ClassId	Sybase	This data store contains the binding information of an ECS application server, which includes process id and port number.
ServerUR.map	Other	A flat file for the Server Locator classes to map short, logical service names to CCS Name Server entry names.

4.8.1.4 Mail Support Group Description

4.8.1.4.1 E-mail Server Functional Overview

The E-mail server provides an interactive and a development interface for managing the electronic mail 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.

4.8.1.4.2 E-mail Server Context

Figure 4.8-8 is the E-mail Server context diagram. Table 4.8-13 provides descriptions of the interface events shown in the E-mail Server context diagram.

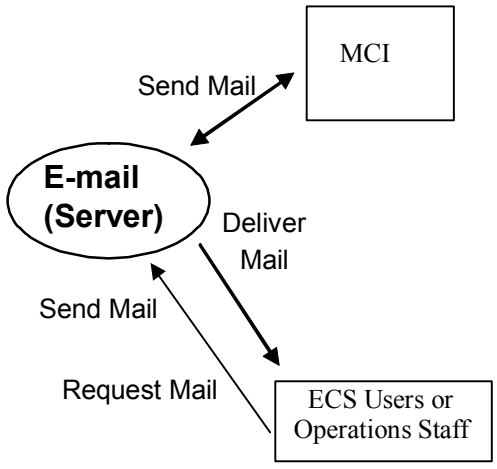


Figure 4.8-8. E-mail Server Context Diagram

Table 4.8-13. E-mail Server Interface Events

Event	Interface Event Description
Send Mail	The **MCI** uses the e-mail client software to send mail and the API spawns a *sendmail* process to deliver the message. Interactive users, use the COTS software product sendmail, which delivers the mail message.
Deliver Mail	The mail server delivers the mail to the addressed **ECS user** or **Operations Staff**.
Request Mail	**ECS users** or **Operations Staff** sends requests for e-mail messages to the E-mail Server.

4.8.1.4.3 E-mail Server Architecture

The E-mail server is a COTS software product. Figure 4.8-9 is the E-mail server architecture diagram.

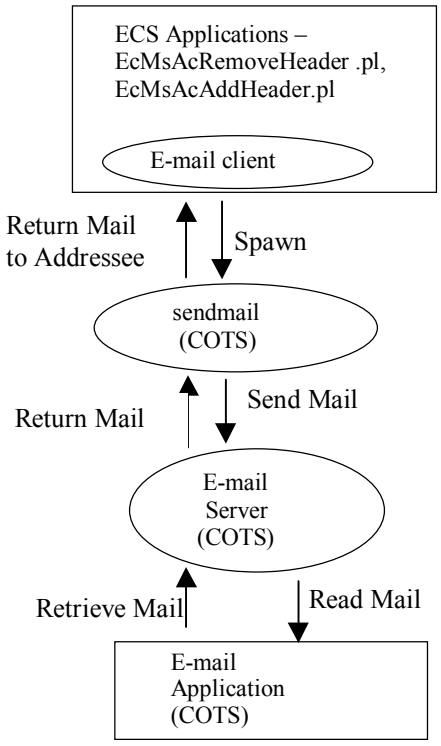


Figure 4.8-9. E-mail Server Architecture Diagram

4.8.1.4.4 E-mail Server Process Descriptions

Table 4.8-14 provides descriptions of the processes shown in the E-mail Server architecture diagram.

Table 4.8-14. E-mail Server Processes

Process	Type	Hardware CI	COTS/ Developed	Functionality
E-mail Client	Other	CSSHW	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	CSSHW	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	CSSHW	COTS	A COTS software E-mail server product.
E-mail Application	Other	CSSHW	COTS	A COTS software product for sending, receiving, and reading E-mail.

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.8.1.4.5 E-mail Server Process Interface Descriptions

Table 4.8-15 provides descriptions of the interface events shown in the E-mail server architecture diagram.

Table 4.8-15. E-mail Server Process Interface Events (1 of 2)

Event	Event Frequency	Interface	Initiated by	Event Description
Spawn	One per process spawned	*Library:* libc API: System ( )	*Scripts:* EcMsAcRemoveHeader.pl, EcMsAcAddHeader.pl *Library:* CsEmMailRelA *Class:* CsEmMailRelA	The **EcMsAcRemoveHeader.pl** and **EcMsAcAddHeader.pl** scripts invoke a process to send E-mail via an API.
Send mail	One per E-mail send	*Process:* E-mail Server (COTS)	*Process:* Sendmail (COTS)	A command from the sendmail process to send electronic mail routed via the E-mail server.

Table 4.8-15. E-mail Server Process Interface Events (2 of 2)

Event	Event Frequency	Interface	Initiated by	Event Description
Read mail	One per E-mail read	*Process:* E-mail Application (COTS)	*Process:* E-mail Server (COTS)	E-mail is received from the COTS application (sendmail) and routed to another user via the E-mail server.
Retrieve mail	One per E-mail send	*Process:* E-mail Server (COTS)	*Process:* E-mail Application (COTS)	E-mail is received from an ECS application, a sendmail process is spawned to forward the mail, and the E-mail is sent via the E-mail server to another user.
Return Mail	One per E-mail read	*Process:* Sendmail (COTS)	*Process:* E-mail Server (COTS)	E-mail is sent from the Sendmail package to a user via the E-mail Server.
Return Mail to Addressee	One per E-mail read	*Process:* E-mail Client	*Process:* Sendmail (COTS)	E-mail is retrieved by a user from the Sendmail package via the E-mail Server.

4.8.1.4.6 E-mail Server Data Stores

Data stores are not applicable for the E-mail Server.

4.8.1.15 Virtual Terminal Description

4.8.1.5.1 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 secure shell server (sshd2) on available systems and common capability support for the ECS remote terminal service.

4.8.1.5.2 Virtual Terminal Context

The CSS provides the secure shell (sshd2) remote access to the ECS systems. SSH is distributed as a third party remote server access service. The SSH service provides users with access to the ECS character-based user interface (CHUI) search and order tool. Figure 4.8-10 is the Virtual Terminal context diagram and Table 4.8-16 provides the descriptions of the interface events shown in the Virtual Terminal context diagram.

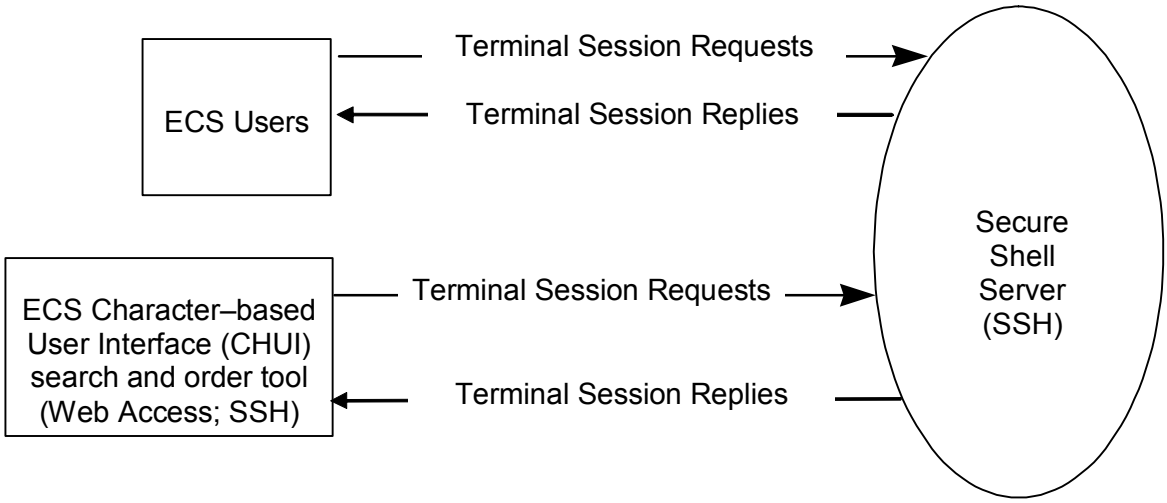


Figure 4.8-10. Virtual Terminal Context Diagram

Table 4.8-16. Virtual Terminal Interface Events

Event	Interface Event Description
Terminal Session Requests (Web access)	**ECS users** request a connection to a specified host via SSH.
Terminal Session Requests (ECS Users)	**ECS users** request a telnet session with a specified ECS host.
Terminal Session Replies (from SSH to ECS or other remote users)	The SSH Server residing on the ECS host responds to the terminal session requests and interacts via the successful connection.

4.8.1.5.3 Virtual Terminal Architecture

Figure 4.8-11 is the Virtual Terminal architecture diagram. The diagram shows the event traffic between the Remote Terminal Session with ECS Users and SSH with remote users.

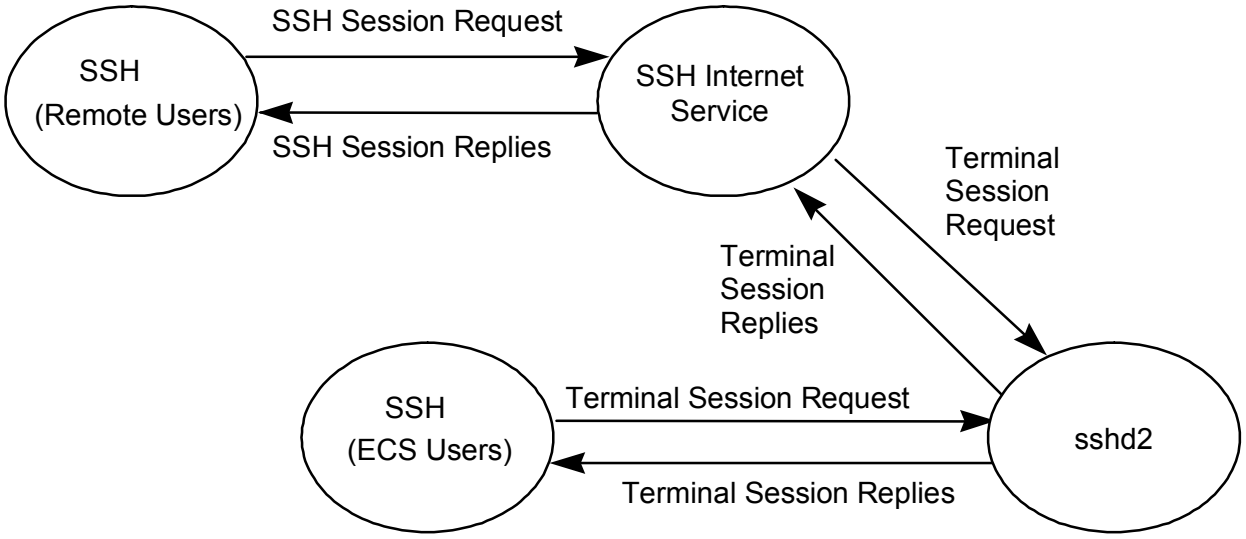


Figure 4.8-11. Virtual Terminal Architecture Diagram

4.8.1.5.4 Virtual Terminal Process Descriptions

Table 4.8-17 provides the descriptions of the processes shown in the Virtual terminal architecture diagram.

Table 4.8-17. Virtual Terminal Processes

Process	Type	Hardware CI	COTS/ Developed	Functionality
SSH (Remote Users)	Client	CSSHW	COTS	Provides the dial-up terminal session as requested on the client-side via remote service.
SSH (ECS Users)	Client	CSSHW	COTS	Provides the user interface to a remote system using the SSH protocol.
(Internet Service)	Server/Client	CSSHW	COTS	Enables users to interact with the host through a remote service.
sshd2	Server	CSSHW	COTS	Function provides servers supporting SSH with virtual terminal protocol.

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.8.1.5.5 Virtual Terminal Process Interface Descriptions

Table 4.8-18 provides the descriptions of the interface events shown in the Virtual Terminal architecture diagram.

Table 4.8-18. Virtual Terminal Process Interface Events

Event	Event Frequency	Interface	Initiated by	Event Description
SSHSession Request	One per connection request	*Process:* sshd2 (dial-up service)	*Process:* SSH (COTS – remote users)	Any **ECS user** requiring a logon to another machine from the current machine. Users request to establish connection to a specified host via SSH.
SSH Session Replies	One per session reply	*Process:* SSH (remote service)	*Process:* sshd2 (remote service)	The **SSH Server** service provides **users** a remote session to request a terminal session to the secure shell client.
Terminal Session Request (SSH)	One per request to establish a session	*Process:* sshd2	*Process:* sshd2 (remote service)	Either the user or the client application service requests to establish a session with the specified host.
Terminal Session Replies (SSH)	One per connection request	*Process:* SSH (ECS users)	*Process:* sshd2	The Host Virtual Terminal Process, sshd2, responds to the connection requests and establishes or maintains the sessions.

4.8.1.5.6 Virtual Terminal Data Stores

Data stores are not applicable for the Virtual Terminal.

4.8.1.6 Cryptographic Management Interface Software Description

4.8.1.6.1 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 security required services (i.e., Sybase ASE). These passwords (and optionally login names) are generated dynamically based on a psuedo-random number used as the seed for the password.

4.8.1.6.2 Cryptographic Management Interface Context

Figure 4.8-12 is the Cryptographic Management Interface context diagram. Servers (PF or non-PF) use the CMI with a need for access to security required services. Table 4.8-19 provides descriptions of the interface events shown in the Cryptographic Management Interface context diagram.

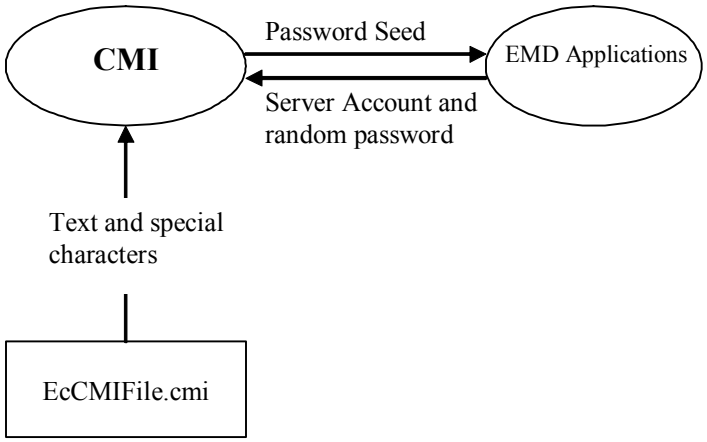


Figure 4.8-12. Cryptographic Management Interface Context Diagram

Table 4.8-19. Cryptographic Management Interface Events

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

4.8.1.6.3 Cryptographic Management Interface Architecture

Figure 4.8-13 is the Cryptographic Management Interface (CMI) architecture diagram. The diagram shows the event traffic between the CMI process and the ECS applications that interact with CMI for database connections.

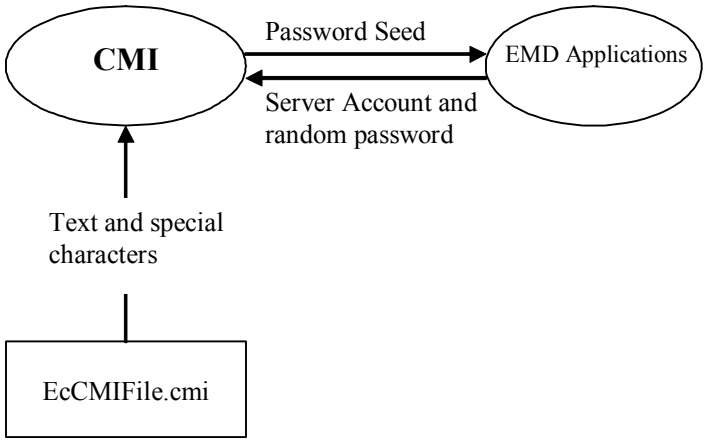


Figure 4.8-13. Cryptographic Management Interface Architecture Diagram

4.8.1.6.4 Cryptographic Management Interface Process Descriptions

Table 4.8-20 provides descriptions of the processes shown in the Cryptographic Management Interface context diagram.

Table 4.8-20. Cryptographic Management Interface Processes

Process	Type	Hardware CI	COTS/ Developed	Functionality
ECS Process Names	Server	CSSHW	Developed	Requests account with random password for access to security required services.
CMI	Other	CSSHW	Developed	A server account and randomly generated password are returned to the requesting server.

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.8.1.6.5 Cryptographic Management Interface Process Interface Descriptions

Table 4.8-21 provides the descriptions of the interface events shown in the Cryptographic Management Interface architecture diagram.

Table 4.8-21. Cryptographic Management Interface Process Interface Events

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

4.8.1.6.6 Cryptographic Management Interface Data Stores

Table 4.8-22 provides descriptions of the data store shown in the Cryptographic Management Interface architecture diagram.

Table 4.8-22. Cryptographic Management Interface Data Stores

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

4.8.1.7 Domain Name Server Software Description

4.8.1.7.1 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 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 EMD project for nasa.gov.

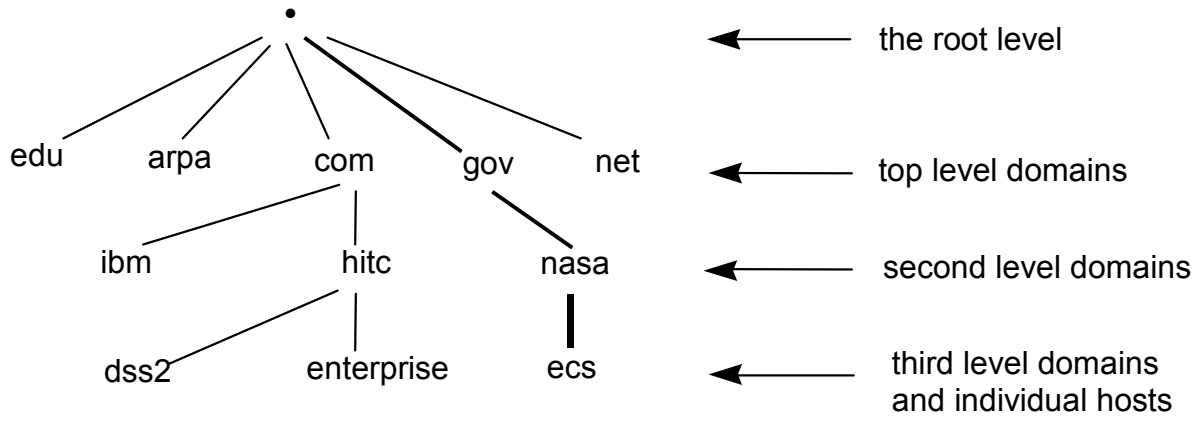


Figure 4.8-14. Domains Hierarchy Diagram

The fourth level domains in the EMD project include domains of DAACs: gsfcb, gsfcmo, and etc. Figure 4.8-15 is the hierarchy diagram of the fourth level domains in the EMD 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 production network and the latter are for the GSFC M&O network.

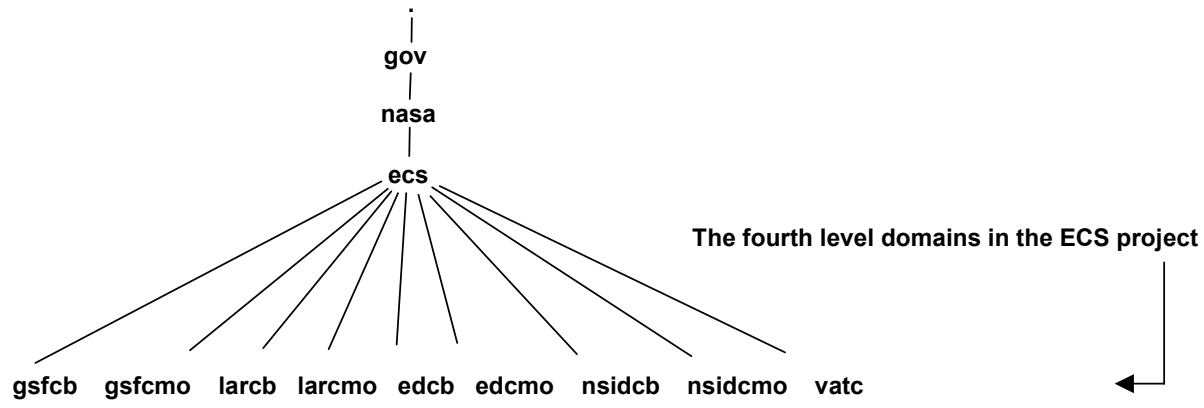


Figure 4.8-15. DNS Domains of the EMD Project Diagram

Figure 4.8-16 is the ECS topology domain diagram.

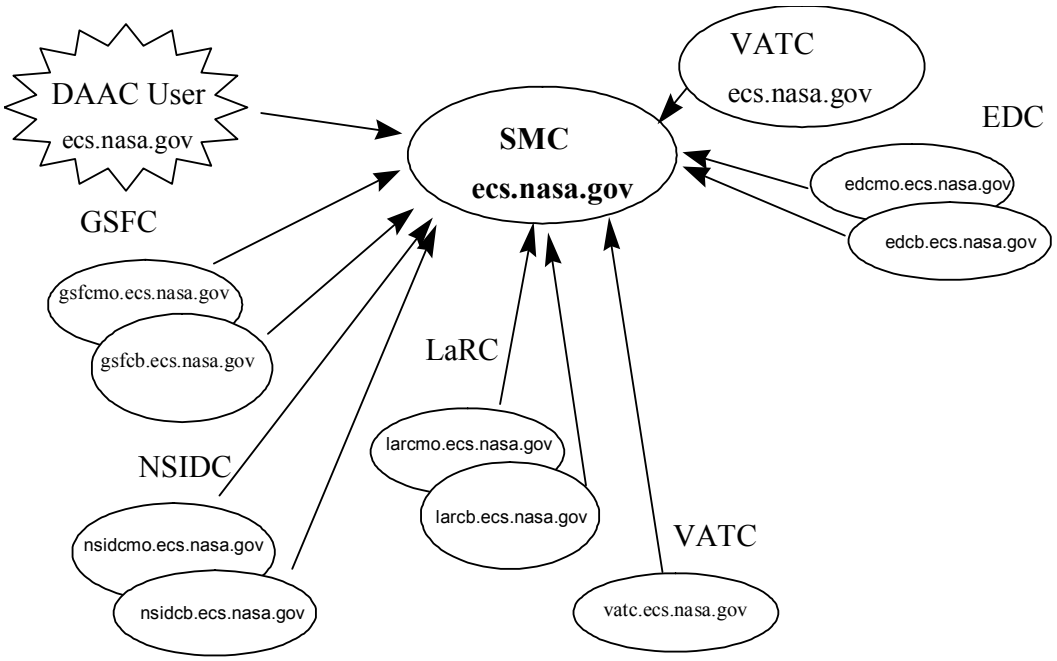


Figure 4.8-16. ECS Topology Domains Diagram

4.8.1.7.2 Domain Name Server Context

Figure 4.8-17 is the Domain Name Server context diagram.

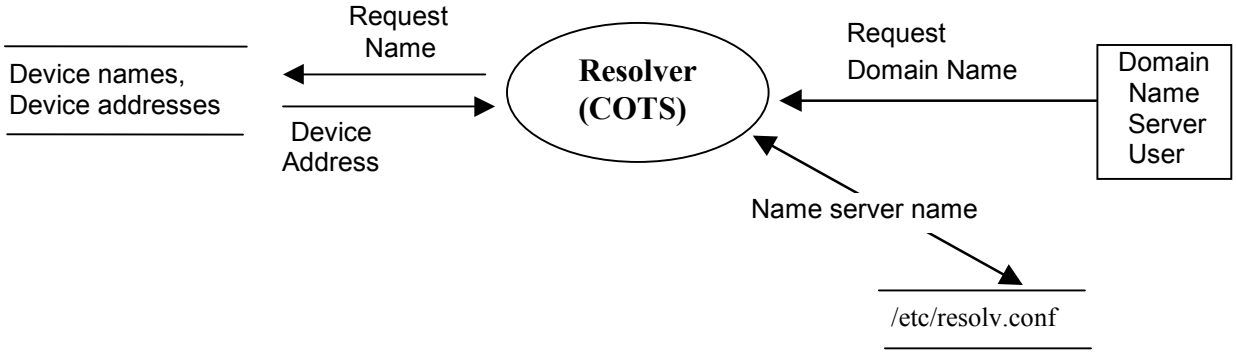


Figure 4.8-17. Domain Name Server Context Diagram

4.8.1.7.3 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.

4.8.1.7.4 Domain Name Server Process Descriptions

Table 4.8-23 provides descriptions of the Domain Name Server processes shown in the Domain Name Server context diagram.

Table 4.8-23. Domain Name Server Process

Process	Type	Hardware CI	COTS/ Developed	Functionality
resolver	Client	CSSHW	COTS	Searches data store of device names and device 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.

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.8.1.7.5 Domain Name Server Process Interface Descriptions

Table 4.8-24 provides descriptions of the interface events shown in the Domain Name Server architecture diagram.

Table 4.8-24. Domain Name Server Process Interface Events

Event	Event Frequency	Interface	Initiated by	Event Description
Request Domain Name	One per user request	*COTS Software:* resolver	User	A DNS **user** requests data.
Name server name	One per search directory change	Data Store	*COTS Software:* resolver	The resolver retrieves the pathname for the directory to search for the user requested data from the /etc/resolv.conf database table. New file names are added to the list in the order they are stored.
Device Address	One per resolved address	*COTS Software:* resolver	*COTS Software:* name server	Returns the resolved address to the domain name requester via the Resolver.
Request Name	One per domain name request	*COTS Software:* name server	*COTS Software:* resolver	The resolver retrieves the domain name (device name and address) for the name server from an internal file used by the COTS software.

4.8.1.7.6 Domain Name Server Data Stores

Table 4.8-25 provides descriptions of the data store shown in the Domain Name Server architecture diagram.

Table 4.8-25. Domain Name Server Data Stores

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

4.8.1.8 Infrastructure Libraries Group Description

4.8.1.8.1 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 4.8-26 provides descriptions of the infrastructures in the ILG.

Table 4.8-26. Infrastructure Libraries (1 of 2)

Library	Description
Process Framework (PF)	The PF is a software library of services, which provides a flexible mechanism (encapsulation) for the ECS client and server applications to transparently include specific ECS infrastructure features from the library of services, such as mode management, error and event logging, life-cycle services, and the CCS Middleware Naming Service.
Server Request Framework (SRF)	The SRF infrastructure provides the standard for ECS synchronous and asynchronous communications between ECS applications. SRF is used to provide the client-server communications between the DPL INGEST Request Manager and Granule Server. SRF provides enhanced CCS Middleware call message passing and persistent storage as a CSS support capability with the described features available by subsystem request.
Universal References (UR)	A Universal Reference provider object from C++ objects generates UR 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.
Error/Event Logging	Event/Error logging is the capability of recording events into files and provides a convenient way to generate and report detailed events. All ECS CSCIs use event and error logging as an audit trail for all transactions (requests for data or services) that occur during the ECS data processing and distributing.
Message Passing (MP)	Message Passing provides peer-to-peer asynchronous communications service, which notifies clients of specific event triggers. This service is provided upon subsystem request by the CSS. It is an alternative means of communication.

Table 4.8-26. Infrastructure Libraries (2 of 2)

Library	Description
ServerUR	Provides unique identification (universal reference) for data and service objects in the ECS. 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. The Server Locator is used in ECS in any client-server CCS Middleware-based communication.
Fault Handling (FH)	The Failure Recovery Framework provides a general-purpose fault recovery routine enabling client applications to reconnect with servers after the initial connection is lost. This is accomplished through the CCS Naming Service, through which the Failure Recovery Framework can determine whether a server is listening. The Failure Recovery Framework provides a default and configurable amount of retries and duration between retries. This fault recovery takes effect for each attempt by the client to communicate with the server for all applications that employ the Failure Recovery Framework.
DBWrapper directory	The DBWrapper directory is the DBMS Interface Infrastructure Library used by ECS applications to connect to the Sybase ASEs. Sybase ASEs operate by ECS defined guidelines for mode management, thread safety, error handling, error recovery, security, configuration management, and performance of database connections.

4.8.1.8.2 Infrastructure Libraries Group Context

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

4.8.1.8.3 Infrastructure Libraries Group Architecture

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

4.8.1.8.4 Infrastructure Libraries Group Process Descriptions

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

4.8.1.8.5 Infrastructure Libraries Group Interface Descriptions

Table 4.8-27 provides descriptions of the interfaces the Infrastructure Libraries Group.

Table 4.8-27. Infrastructure Libraries Group Interfaces (1 of 3)

Library	Interface	Initiated by	Library Description
Process Framework (PF)	*Library:* EcPf *Classes:* EcPfManagedServer, EcPfClient	EcDsScienceDataServer, EcDsHdfEosServer, EcClDtUserProfileGateway, EcDmDictServer, EcDmV0ToEcsGateway, EcCsRegistry, EcCsMtMGateway, EcMsAcRegUserSrvr, EcMsAcOrderSrvr	The PF is a software library of services, which provides a flexible mechanism (encapsulation) for the ECS client and server applications to transparently include specific ECS infrastructure features from the library of services. Features and services include: • Mode management • Error and event logging • Life-cycle services • CCS Naming Service
Server Request Framework (SRF)	*Library (Common):* srf *Classes:* EcSrRequestServer_C, EcSrAsynchRequest_C	EcDmDictServer, EcMsAcRegUserSrvr, EcDsScienceDataServer	The SRF infrastructure provides the standard for ECS synchronous and asynchronous communications between ECS applications. SRF is used to provide the client-server communications between the DPL INGEST Request Manager and Granule Server. SRF provides enhanced CCS Middleware calls, message passing and persistent storage as a CSS support capability with the described features available by subsystem request.
Universal References (UR)	*Library (Common)*: EcUr	Object Origination	A Universal Reference provider object from C++ objects generates UR 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.

Table 4.8-27. Infrastructure Libraries Group Interfaces (2 of 3)

Library	Interface	Initiated by	Library Description
Error/Event Logging	*Library:* event *Class:* EcLgErrorMsg	EcDmDictServer, EcMsAcRegUserSrvr, EcMsAcOrderSrvr, EcDsScienceDataServer, EcDsHdfEosServer, EcDmV0ToEcsGateway, and EcCsMtMGateway	Event/Error logging is the capability of recording events into files and provides a convenient way to generate and report detailed events. All ECS CSCIs use event and error logging as an audit trail for all transactions (requests for data or services) that occur during the ECS data processing and distributing.
ServerUR	*Library (Common):* EcUr *Class:* EcUrServerUR	*Processes:* EcDsScienceDataServer *Classes:* EcNsServiceLoc *DSS Libraries:* DsBt, DsDe1, DsGe	Provides unique identification (universal reference) for data and service objects in the ECS. 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. The Server Locator is used in ECS in any client-server CCS Middleware-based communication.
Fault Handling (FH)	*Library:* EcFh *Class:* EcFhExecutor	EcDmDictServer, EcMsAcRegUserSrvr, EcMsAcOrderSrvr, EcDsScienceDataServer, EcDsHdfEosServer	The Failure Recovery Framework provides a general-purpose fault recovery routine enabling client applications to reconnect with servers after the initial connection is lost. This is accomplished through the CCS Naming Service, through which the Failure Recovery Framework can determine whether a server is listening. The Failure Recovery Framework provides a default and configurable amount of retries and duration between retries. This fault recovery takes effect for each attempt by the client to communicate with the server for all applications that employ the Failure Recovery Framework.
DBWrapper directory	*Libraries:* EcPoDbRW, EcPoDb *Class:* EcPoConnectionsRW	*Processes:* EcDmDictServer	This is the DBMS Interface Infrastructure Library. Sybase ASEs implement ECS defined guidelines for mode management, thread safety, error handling, error recovery, security, configuration, and performance of database connections.

Table 4.8-27. Infrastructure Libraries Group Interfaces (3 of 3)

Library	Interface	Initiated by	Library Description
Time Service	*Libraries:* EcTiTimeService *Class:* EcTiTimeService	*n/a*	This class provides the structured current time information and RogueWave time information.

4.8.1.8.6 Infrastructure Library Group Data Stores

Data Stores are not applicable for the Infrastructure Library Group.

4.8.2 The Distributed Computing Configuration Item Context

Figure 4.8-18 is the Distributed Computing Configuration Item (DCCI) CSCI context diagrams. The diagrams show the events sent to the DCCI CSCI and the events the DCCI CSCI sends to other CSCIs. Table 4.8-28 provides descriptions of the interface events shown in the DCCI CSCI context diagrams.

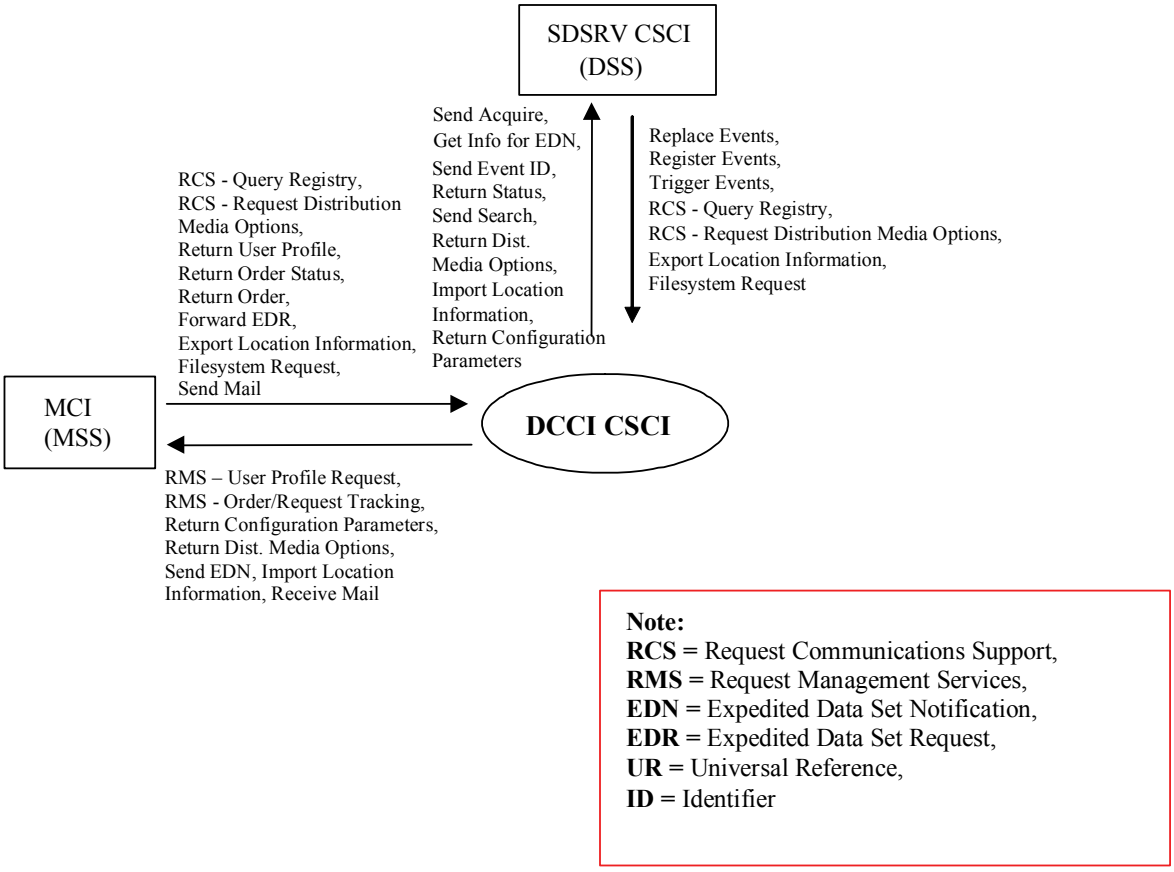


Figure 4.8-18. Distributed Computing Configuration Item (DCCI) CSCI Context Diagram

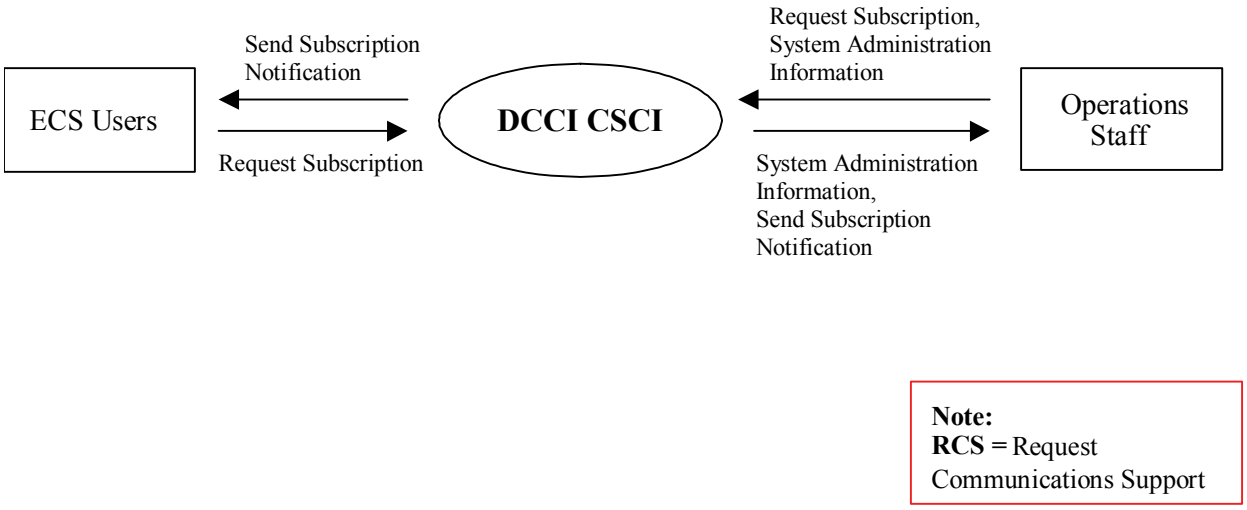


Figure 4.8-18. Distributed Computing Configuration Item (DCCI) CSCI Context Diagram (cont.)

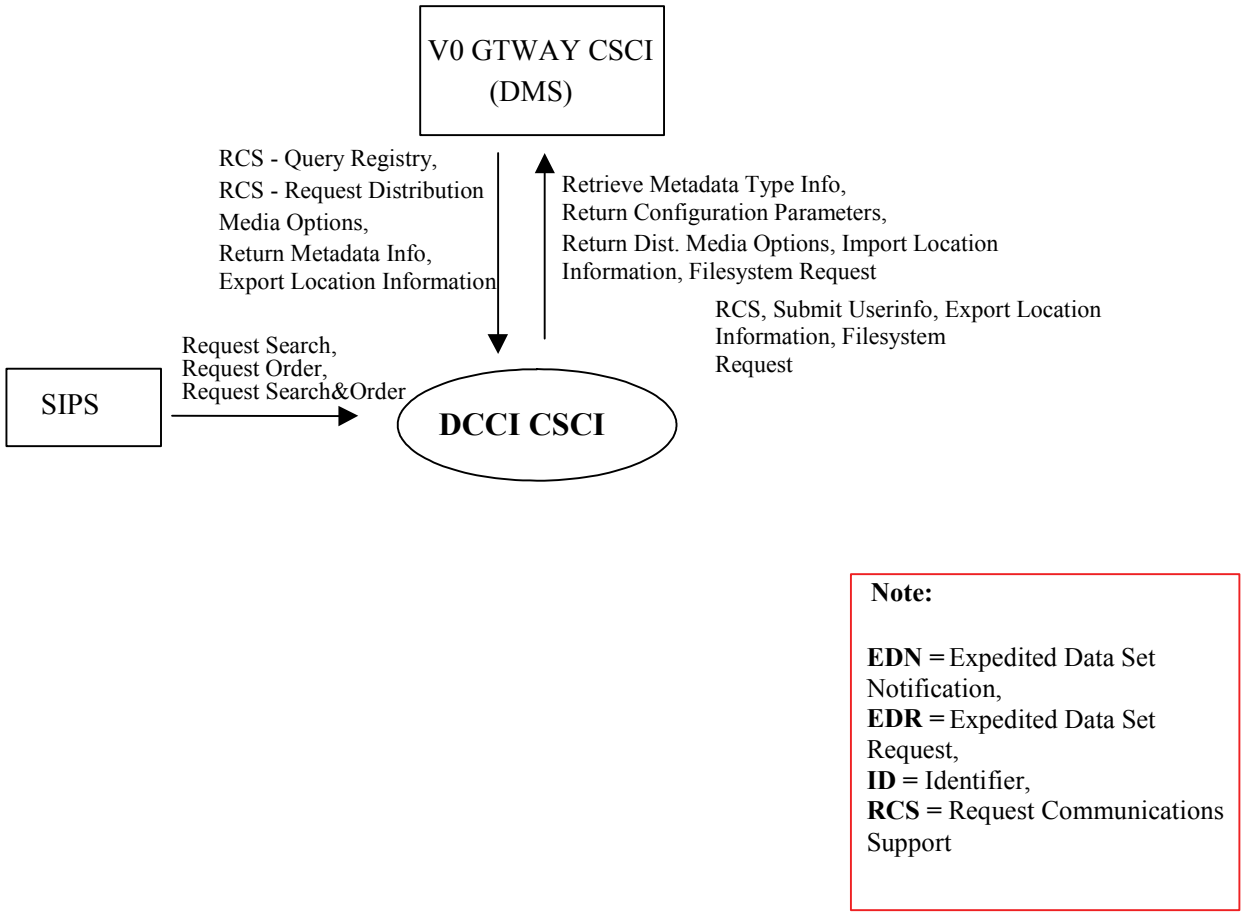


Figure 4.8-18. Distributed Computing Configuration Item (DCCI) CSCI Context Diagram (cont.)

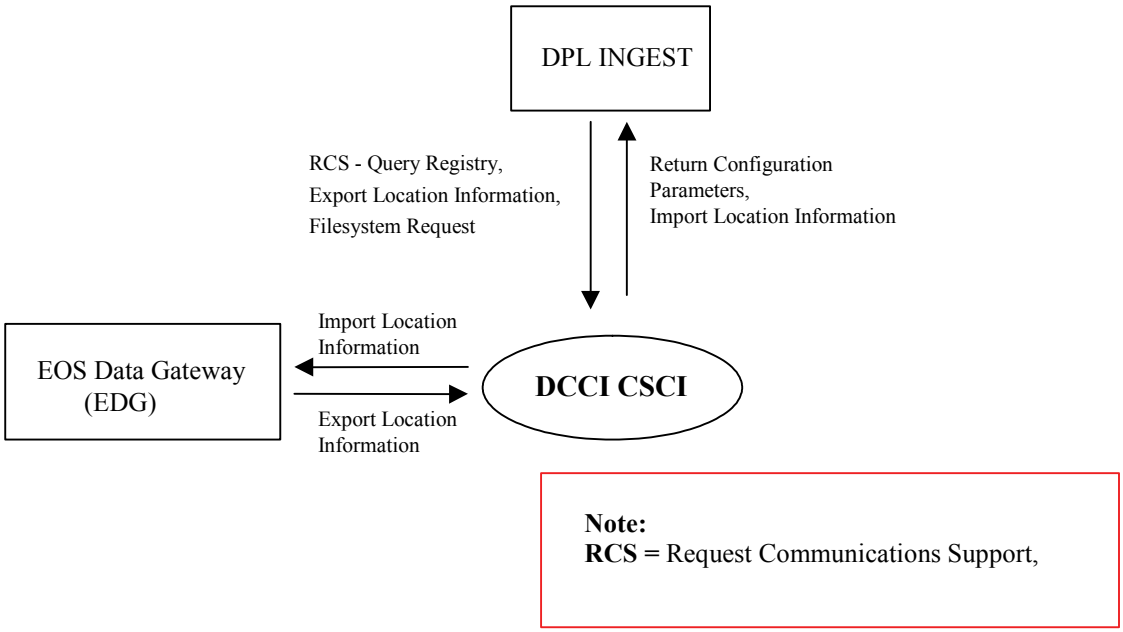


Figure 4.8-18. Distributed Computing Configuration Item (DCCI) CSCI Context Diagram (cont.)

Table 4.8-28. Distributed Computing Configuration Item (DCCI) CSCI Interface Events (1 of 6)

Event	Interface Event Description
Send Acquire	An “acquire” (instruction to obtain data) is created by the DCCI CSCI and sent to the **SDSRV CSCI** via CCS Middleware calls. This is similar to the “Request Product” interface event, except it applies to EDOS expedited data.
Get Info for EDN	Expedited Data Set Notification (EDN) information is obtained from the **SDSRV CSCI**, by request, and used by the DCCI CSCI to send messages to users.
Send Event ID	The DCCI CSCI sends Event IDs to the **SDSRV CSCI** when ESDTs are installed or when ESDTs are updated by adding additional events.
Return Status	Status returned by the DCCI CSCI to the **SDSRV CSCI** to simply indicate that the request was received, not that the action succeeded.
Send Search	The DCCI CSCI (Machine-to-Machine Gateway Server CSC) sends search requests received via the SIPS interface to the **SDSRV CSCI** on behalf of an external ECS user.
Return Dist. Media Options	The Configuration Registry CSC returns the requested distribution media options to the **SDSRV** and **MCI CSCIs.**
Import Location Information	The **SDSRV CSCI** request server location information from the CCS Name Server.
Return Configuration Parameters	The DCCI CSCI returns the requested configuration parameters to the **SDSRV, MCI CSCI and MTMGW CSC**.
Replace Events	The **SDSRV CSCI** sends the updated subscription events with modified qualifiers for an Earth Science Data Type (ESDT) to the DCCI CSCI (Subscription Server) when an ESDT is updated. This event replaces the original event in the DCCI CSCI.
Register Events	The **SDSRV CSCI** sends the subscription events for an Earth Science Data Type to the DCCI CSCI (Subscription Server) when an ESDT is installed into the system or when an ESDT is updated by adding additional events.
Trigger Events	The **SDSRV CSCI** notifies the DCCI CSCI (via an event trigger) when a subscription event occurs on an Earth Science Data Type Service.

Table 4.8-28. Distributed Computing Configuration Item (DCCI) CSCI Interface Events (2 of 6)

Event	Interface Event Description
Request Communications Support	The DCCI CSCI provides a library of services available to the **SDSRV** and **MCI CSCIs**. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • File Transfer Services • Network & Distributed File Services • Bulk Data Transfer Services • File Copying Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Message Passing • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry • Request Distribution Media Options from the Configuration Registry
Export Location Information	The DCCI CSCI stores physical and logical location information received from the **MCI CSCI** in the CCS Name Server.
Filesystem Request	The NFS clients request ECS 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.
Submit Subscription	The **DPL CSCI** submits a subscription request to the DCCI CSCI using the advertisement subscribing to an insert event for an ESDT.
Password Seed	The **DPL CSCI** requests an account and provides a password seed to the CMI.
Notify of Subscription	The DCCI CSCI sends notification (via message passing) to the **DPL CSCI** when the subscribed event occurs.
Server Account and random password	An account with random passwords, created for the server, is passed back to the server in the **DPL** CSCI from the DCCI CSCI.

Table 4.8-28. Distributed Computing Configuration Item (DCCI) CSCI Interface Events (3 of 6)

Event	Interface Event Description
Request Management Services	The **MCI** provides a basic management library of services to the CSCIs, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes: • **System startup and shutdown -** Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625). • **User Profile Request** – The **MCI** provides requesting CSCIs with User Profile parameters such as e-mail address and shipping address to support their processing activities. • **Order/Request Tracking** – The **MCI** provides an order tracking service to requesting subsystems to create and track user product orders by request.
Send EDN	The DCCI CSCI (E-mail Parser Gateway Server CSC) stores the EDN messages with URs, time range, etc., and sends the EDN to the **MCI.**
Receive Mail	The **MCI** requests electronic mail sent by users to the DCCI CSCI mail server.
Return User Profile	The **MCI** returns user profiles to the DCCI CSCI to authenticate users for use of ECS data and services.
Return Order Status	The **MCI** provides order ids and order status information for products requested by users.
Return Order	The MCI returns the product order object to the requester via the DCCI CSCI.
Send Mail	The **MCI** sends electronic mail to the DCCI CSCI mail server to be stored for other users.
Request Subscription	An **ECS user** or **Operations Staff** member submits a request for a subscription to the DCCI CSCI. The subscription notifies the user or Operations Staff member whenever the desired event occurs in the system.
System Administration Information	The **Operations Staff** requests and receives information on system administration including application administration, fault metrics, performance metrics and system alarms from the DCCI CSCI.
Send Subscription Notification	An **ECS user** or **Operations Staff** member receives notification the subscription event has occurred.

Table 4.8-28. Distributed Computing Configuration Item (DCCI) CSCI Interface Events (4 of 6)

Event	Interface Event Description
Request Communications Support	The DCCI CSCI provides a library of services available to the **DPL CSCIs**. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • File Transfer Services • Network & Distributed File Services • Bulk Data Transfer Services • File Copying Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Message Passing • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry • Request Distribution Media Options from the Configuration Registry
Filesystem Request	The NFS clients request ECS 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.
Password Seed	The **DPL CSCI** requests an account and provides a password seed to the CMI.
Import Location Information	The **DPL CSCI** request server location information from the CCS Name Server.
Server Account and random password	An account with random passwords, created for the server, is passed back to the server in the **DPL CSCI** from the DCCI CSCI.
Retrieve Metadata Type Info	The DCCI CSCI retrieves type information for qualifying metadata specified in a SIPS search request from the **DDICT CSCI**.
Return Configuration Parameters	The DCCI CSCI returns the requested configuration parameters to the **DDICT and V0 GTWAY CSCIs**.
Return Dist. Media Options	The Configuration Registry CSC returns the requested distribution media options to the **DDICT and V0 GTWAY CSCIs**.
Import Location Information	The **DDICT, V0 GTWAY CSCI** request server location information from the CCS Name Server.
Filesystem Request	The NFS clients (via **DDICT and V0 GTWAY)** request ECS 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.
Export Location Information	The DCCI CSCI stores physical and logical location information, received from the **DDICT and V0 GTWAY,** in the CCS Name Server.

Table 4.8-28. Distributed Computing Configuration Item (DCCI) CSCI Interface Events (5 of 6)

Event	Interface Event Description
Request Search	Search requests are sent to the DCCI CSCI (Machine-to-Machine Gateway Server CSC) via the **SIPS** interface.
Request Order	Order requests are sent to the DCCI CSCI (Machine-to-Machine Gateway Server CSC) via the **SIPS** interface.
Request Search&Order	Integrated search and order requests are sent to the DCCI CSCI (Machine-to-Machine Gateway Server CSC) via the **SIPS** interface.
Return Metadata Info	The DCCI CSCI receives metadata type information from the **DDICT CSCI**.
Return Configuration Parameters	The DCCI CSCI returns the requested configuration parameters to the **DPL INGEST CSCI**.
Import Location Information	The **EOS Data Gateway (EDG)** and **DPL INGEST CSCI** request server location information from the CCS Name Server.
Push Data	The **SDSRV CSCI** pushes data (i.e., EDS), using the FTP service, to the DCCI CSCI for data distribution per user request. A signal file is also sent to indicate the completion of the file transfer for some ESDTs.
Export Location Information	The DCCI CSCI stores physical and logical location information, received from the **EOS Data Gateway (EDG)** and **DPL INGEST** and **SDSRV CSCIs,** in the CCS Name Server.
Request Communications Support	The DCCI CSCI provides a library of services available to the **SDSRV and DPL INGEST CSCIs**. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • File Transfer Services • Network & Distributed File Services • Bulk Data Transfer Services • File Copying Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Message Passing • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry • Request Distribution Media Options from the Configuration Registry

Table 4.8-28. Distributed Computing Configuration Item (DCCI) CSCI Interface Events (6 of 6)

Event	Interface Event Description
Filesystem Request	The NFS clients (via **DDICT and V0 GTWAY)** request ECS 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.
FTPFile	The **SDSRV CSCI** sends requests to the FTP Daemon to transfer the files to the Pull cache or to an external user.
Copy File	The **SDSRV CSCI** inserts data into the archives sending a request for a Unix file copy into the AMASS cache by buffered read/write software using the Filecopy utility.

4.8.3 Distributed Computing Configuration Item Architecture

An architecture diagram is not applicable for the DCCI CSCI. However, Table 4.8-29 shows the mapping between CSMS CSCIs and CSS CSCs.

Table 4.8-29. CSMS CSCI to CSS CSC Mappings (1 of 2)

CSMS CSCI	CSS CSC	Process Used	CSS Libraries Used
SDSRV	– CCS Middleware – E-mail Parser Gateway Server – Configuration Registry Server	– EcCsIdNameServer – EcCsEmailParser – EcCsRegistry	– Process Framework (PF) – ServerUR, EcSbCl – Error Logging – Event Logging – Universal Reference (UR) – EcCsRegistry – CCS Middleware
DPL INGEST	– CCS Middleware – E-Mail Parser Gateway Server – FTP – NFS – Configuration Registry Server	– EcCsIdNameServer – EcCsEmailParser – ftp_popen – NFS Client – EcCsRegistry	– PF – ServerUR – Error Logging – Event Logging – UR – Fault Handling Services – Server Request Framework (SRF) – CCS Middleware

Table 4.8-29. CSMS CSCI to CSS CSC Mappings (2 of 2)

CSMS CSCI	CSS CSC	Process Used	CSS Libraries Used
EOS Data Gateway	– CCS Middleware	– EcCsIdNameServer	– PF – ServerUR – UR – Error Logging – Event Logging – CCS Middleware
Desktop	N/A	N/A	– PF – ServerUR – Error Logging – Event Logging – UR – CCS Middleware
DDICT	– CCS Middleware – Configuration Registry Server	– EcCsIdNameServer – EcCsRegistry	– PF – ServerUR – Error Logging – Event Logging – UR – CCS Middleware
V0 Gateway	– CCS Middleware – Configuration Registry Server	– EcCsIdNameServer – EcCsRegistry	– PF – ServerUR – Error Logging – Event Logging – UR – CCS Middleware
MCI (CSMS)	– E-Mail Parser Gateway Server	– EcCsEmailParser	– PF – ServerUR – Error Logging – Event Logging – UR – CCS Middleware

4.8.4 Distributed Computing Configuration Item Process Descriptions

Process descriptions are not applicable for the DCCI CSCI.

4.8.5 Distributed Computing Configuration Item Process Interface Descriptions

Process interface descriptions are not applicable for the DCCI CSCI.

4.8.6 Distributed Computing Configuration Item Data Stores

Data stores are not applicable for the DCCI CSCI.

4.8.7 Communications Subsystem Hardware CI Description

Document 920-TDx-001 (HW Design Diagram) provides descriptions of the Distributed Computing Configuration HWCI and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

Three DCCI software programs run on this host including the Domain Name Server (DNS), Network Information Services (NIS), and Mail Server. DNS enables host names to be distinguished based on their host name and IP address relationship. NIS is a service that stores information that users, workstations, and applications must have to communicate across the network.. This information includes machine addresses, user names, passwords, and network access permissions. The Mail Server provides standard electronic mail capability.

The CSS Server is a stand-alone host and intrinsically does not have fail-over capability. DNS and Distributed Time Service (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.

4.9 System Management Subsystem Overview

The System Management Subsystem (MSS) provides a complement of tools and services to manage EMD operations. The management services provided cover five major areas including fault, configuration, accountability, performance, and security (FCAPS). The MSS is implemented using COTS products customized to meet EMD requirements, wherever possible. The MSS maintains policy neutrality in implementing EMD management support.

The MSS software is installed locally at each DAAC to manage production operations. The MSS software is also installed at the System Monitoring Center (SMC) at GSFC to perform designated common support functions for all sites, such as maintaining the EMD User Profile database and replicating it to the DAACs for their local use.

System Management Subsystem Context

Figure 4.9-1 is the System Management Subsystem context diagrams. The external systems referred to in the context diagrams are EDOS, ASTER, NASA Internet Science Network (NISN), Version 0 (V0) Information Management System (IMS), ESDIS, and Science Users. Table 4.9-1 provides descriptions of the interface events shown in the MSS context diagrams.

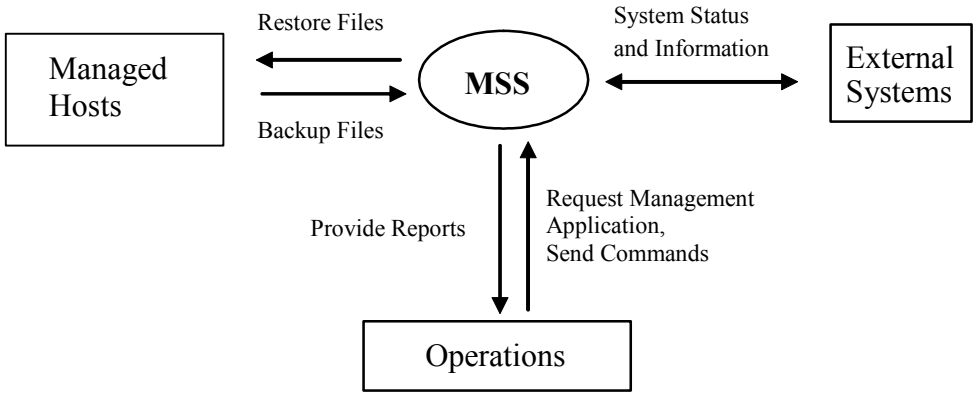


Figure 4.9-1. System Management Subsystem Context Diagram

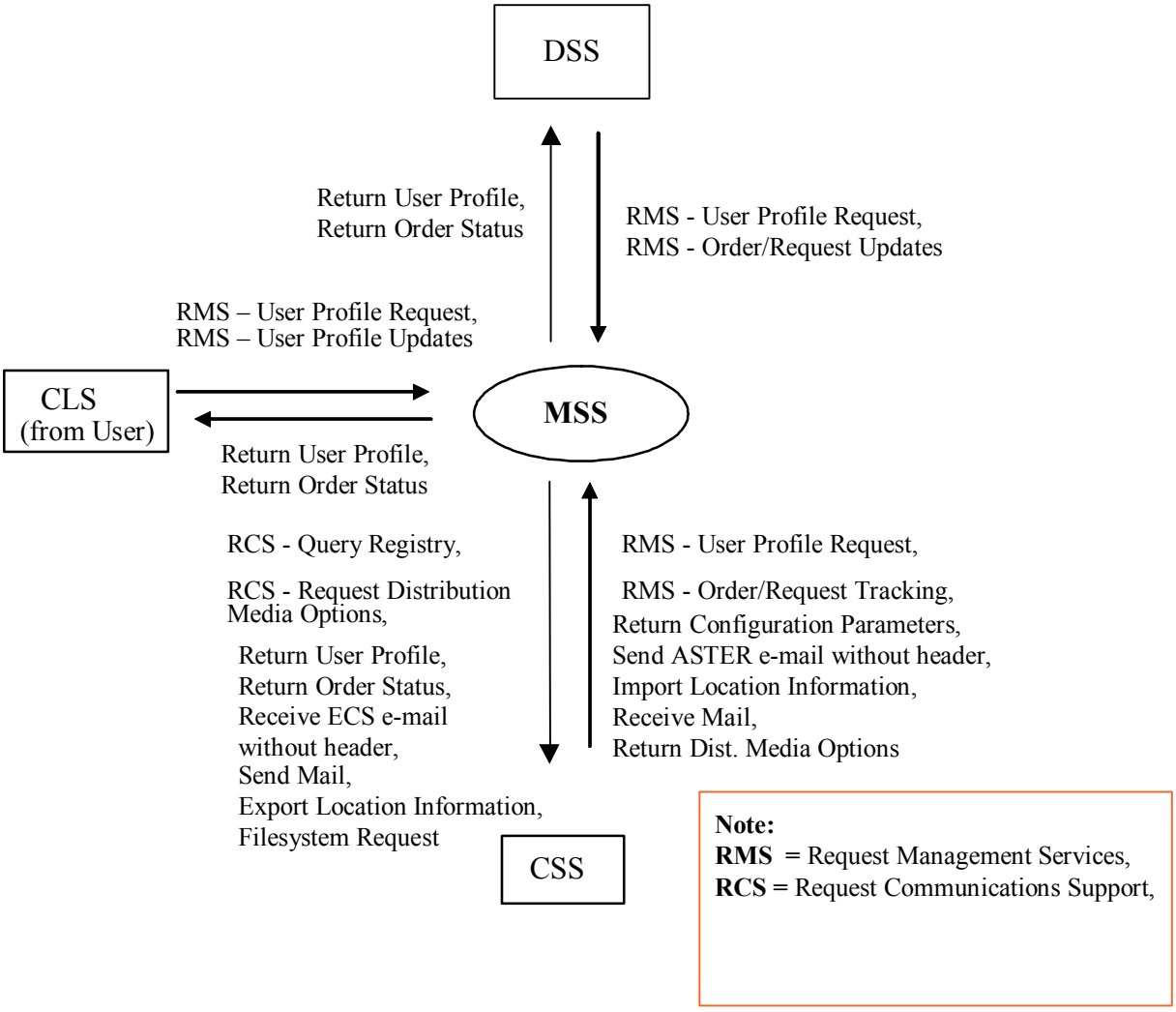


Figure 4.9-1. System Management Subsystem Context Diagram (cont.)

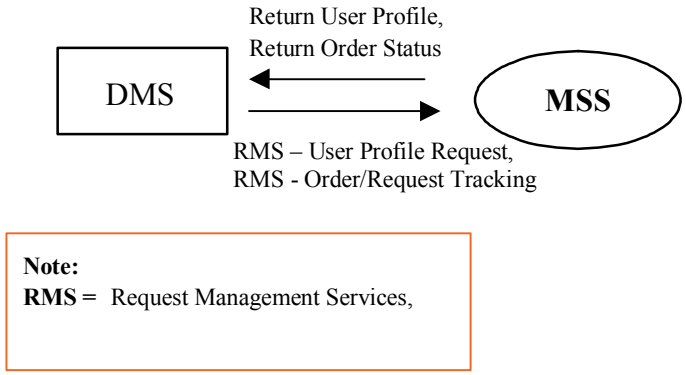


Figure 4.9-1. System Management Subsystem Context Diagram (cont.)

Table 4.9-1. System Management Subsystem Interface Events (1 of 4)

Event	Interface Event Description
System Status and Information	The MSS exchanges system status, trouble reports, and management report information with **external systems** such as the NASA Integrated Services Network (NISN), and V0 Information Management System (IMS) via NISN.
Request Management Application	The **Maintenance and Operations** (**M&O**) **staff** interacts with the MSS management application service tools in the fault, configuration, accountability, performance and security management areas. These tools enable the Operations staff to collect information/metrics, schedule resources for maintenance, monitor and analyze trends, maintain the baseline and schedules, and maintain user profiles.
Send Commands	The **M&O Staff** issue commands to the MSS (Security Service CSC utilities) to exercise system security setup.
Provide Reports	The MSS (Security Service CSC utilities) perform their functions and report results to the **M&O Staff**.
Backup Files	Data is passed from the **Managed Hosts** (the client) to the MSS (the server) and archived to tape.
Restore Files	Data is passed to **Managed Hosts** (the client) to restore lost data from the tape backups from the MSS (the server).

Table 4.9-1. System Management Subsystem Interface Events (2 of 4)

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 includes: • **System startup and shutdown** – Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625). The MSS also interfaces with other subsystems to perform the following: • **User Profile Request** – The MSS provides requesting subsystems (**DSS, CSS, CLS and DMS**) with User Profile information such as e-mail address and shipping address to support their processing activities. • **Order/Request Tracking** – The **CSS** (MTMGW CSC) and **DMS** (V0 GTWAY) can also create a user product order and retrieve it from the MSS. • **User Profile Updates** – The MSS receives user profile parameter updates from a user (via **CLS**) and makes the updates in the user profile database.
Return Configuration Parameters	The **CSS** returns the requested configuration parameters to the MSS.
Import Location InformationSend ASTER e-mail without header	The **CSS** returns physical and logical server location information to the MSS.The **CSS** sends an e-mail message to a predefined ASTER e-mail alias within the EMD (in MSS), without a header (e.g., Expedited Data Set Notification or EDN).
Receive MailImport Location Information	The **CSS** returns mail to the MSS to be distributed to addressees.The **CSS** returns physical and logical server location information to the MSS.
Return Dist. Media OptionsReceive Mail	The **CSS** returns distribution media options to the MSS.The **CSS** returns mail to the MSS to be distributed to addressees.
Return Dist. Media Options	The **CSS** returns distribution media options to the MSS.

Table 4.9-1. System Management Subsystem Interface Events (3 of 4)

Event	Interface Event Description
Request Communications Support	The **CSS** provides a library of services available to each SDPS and CSMS subsystems. The subsystem services required to perform specific assignments are requested from the CSS. These services include: • CCS Middleware Support • Database Connection Services • File Transfer Services • Network & Distributed File Services • Bulk Data Transfer Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry – Retrieving the requested configuration attribute-value pairs from the Configuration Registry • Request Distribution Media Options from the Configuration Registry
Return User Profile	The MSS returns user profile information requested by users to the **DSS, CSS** and **CLS**.
Return Order Status	The MSS provides order ids and order status information to the **CLS** and **CSS** for products requested by users.
Receive EMD e-mail without header	The header is removed from the inbound message, logged, and forwarded to the predefined EMD recipient of the e-mail alias by the **CSS**.
Send Mail	The MSS uses the CsEmMailRelA interface to send mail to the **CSS**.
Export Location Information	The MSS stores physical and logical server location information in the **CSS**.
Filesystem Request	The **CSS** (NFS client) receives requests for EMD files and directories via an established mount point from the MSS. The CSS (NFS Server) makes the storage device(s) and its data accessible for use by the clients.

Table 4.9-1. System Management Subsystem Interface Events (4 of 4)

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 includes: • **System startup and shutdown** – Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625). • **User Profile Request** – The MSS provides the **DMS** with User Profile information such as e-mail address and shipping address to support their processing activities. • **Order/Request Tracking** – The **DMS** interfaces with the Order/Request Tracking service to create and track a user product order.
Return User Profile	The MSS returns user profile information requested by users to the **DMS**.
Return Order Status	The MSS provides order ids and order status information to the **DMS** for products requested by users.
Return Order	An internal description of the order requested by the user (an object).

System Management Subsystem Structure

The MSS has two CSCIs and one hardware CI. The two CSCIs are the Management Software CSCI (MCI) and the Management Logistics CSCI (MLCI). The MCI is mainly COTS software and provides distributed system management support capabilities in the fault, configuration, accountability, performance, and security service areas. The MCI custom software mainly consists of accountability software and custom extensions to COTS applications. The MLCI is largely COTS software, some of which has been customized by the vendor for the EMD, which supports managing the configuration of the EMD.

The MSS hardware CI consists of a single hardware configuration item, the MSSHW, provided at the SMC, the Earth Observing System Operations Center (EOC), and each DAAC. The MSSHW includes an MSS management server, an MSS backup Server, an MSS application server, management workstations, and printers. The MSSHW provides processing and storage support for the execution of the management applications within the MCI and MLCI.

Use of COTS in the System Management Subsystem

The MSS design uses COTS software to implement and provide management services as described below. Detailed explanations of the COTS software are provided in the CSC descriptions.

Note: The RogueWave libraries mentioned below are currently delivered with EMD custom code as static libraries. A separate installation of the libraries is no longer required.

- RogueWave's Tools.h++

  The Tools.h++ class libraries provide strings and collections.

- RogueWave's DBTools.h++

  The DBTools.h++ class libraries interact with the Sybase database Structured Query Language (SQL) server. The use of DBTools buffers the MSS processes from the relational database used.

- ICS' Builder Xcessory

  The Builder Xcessory GUI builder tool modifies the displays of MSS GUIs. The Builder Xcessory tool also generates the C++ code that produces MSS GUIs at run time. No operational part of this tool is needed at run-time.

- Sybase Adaptive Server Enterprise (ASE)

  Sybase's ASE provides access for MSS to insert, update, and delete MSS database information. The Sybase ASE must be running during operations for the User Profile Server and Order Tracking Server to operate.

- Crack

  Crack is a security management program that identifies user passwords that can be easily guessed. Crack enables systems administrators to force users to create passwords that are more difficult for a potential intruder to exploit.

- Anlpassword

  Anlpassword is a security management program that enables system administrators to set certain rules for password creation (e.g., must be at least 8 characters long and contain a number or symbol). The Anlpassword program makes it more difficult for passwords to be guessed and exploited by potential intruders.

- TCP Wrappers

  TCP Wrappers is a security management program that monitors and controls user applications that connect to various network services, such as TFTP, EXEC, FTP, RSH, TELNET, RLOGIN, FINGER, and SYSTAT. The actions performed by the TCP Wrappers program are configurable, but consist of logging the remote host name and performing basic checks on the request origin.

- Tripwire

  Tripwire is security management program, which is an integrity monitor. Tripwire uses several checksum/signature routines to detect file changes and monitors selected items of system-maintained information. Tripwire monitors permission, link, file size, and directory changes. It also detects file additions or deletions based on selected directories that are watched.

- ClearCase

  ClearCase is a UNIX software change management application used to maintain algorithms at each DAAC.

- ClearCase Baseline Management (ClearCase BLM)

  ClearCase BLM custom tool used at Landover to manage the EMD Baseline Management System (EBIS). It provides both general and site specific baselines for COTS and Custom software, O/S patches for each EMD host, and UNIX kernel parameters for each EMD host.

- Networker

  Networker is an application, which provides capabilities to backup and restore files or directories for all EMD hosts. Networker provides an interface for the system administrator to identify the files or directories for back up or restoring and performs the backup or restore according to specifications.

- DDTS

  DDTS is a UNIX based configuration management tool to handle configuration change requests (CCRs) in the EMD system. DDTS provides the user the capability to generate, monitor, and report on EMD CCRs.

- Remedy's Action Request System (ARS)

  The Remedy ARS (usually referred to as "Remedy") is used to support inventory, logistics, and maintenance (ILM) activities for ECS. The Remedy ILM application provides the capability to track EMD hardware inventory, logistics, software licenses, and maintenance transactions.

- TestTrack Pro (TTPro)

  TestTrack Pro provides the capability to electronically compose, submit, update, and report on trouble tickets via Linux, Unix, Solaris and Windows clients as well as Web browsers. It also allows trouble tickets to be escalated to the ECS Development Facility for assistance or resolution.

- Sun Java System Web Server, Enterprise Edition

  The Sun Java System Web Server provides world-wide web services for EMD applications such as Trouble Ticketing. For example, it implements a web interface for TestTrack Pro, enabling EMD users to submit, update, and track the progress of trouble tickets via a web browser.

- Perl

  The Perl language is used for various system utilities..

- FLEXlm

  FLEXlm is a license manager for controlling the use of software products. It manages license checkout and checkin processing, logs licensing events and errors, removes user licenses for specific features, displays the status of installed licenses and network licensing activities, and reports hostids of a system.

- Tcl/Tk

  The Tool Command Language is a scripting language, which runs on multiple platforms. As an interpreted language each statement is read in, parsed and executed in runtime. Tcl provides most of the handy utility functions shell scripts do to go through directories and sort the file names, execute commands and so on. An associate add-on toolkit allows a user to quickly create graphical applications without delving into packages like Win32, Motif and the X toolkit. With a surprisingly small amount of code, a user can quickly develop graphical applications.

- CCS Middleware Client

  CCS Middleware Client provides MSS with communications between other subsystems. CCS Middleware can reside on one or both sides of the interface. An instance must be installed on the platform where MSS resides. Although the CCS Middleware Client is part of CSS, this COTS item must be installed for MSS to run in the SDPS operational and test environment.

Error Handling and processing

When an error occurs, the error is logged into the applications log (ALOG). The Communications Subsystem (CSS) and System Management Subsystem (MSS) have two main mechanisms to handle the error:

1. Return an error status
2. Throw an exception

The MSS uses the following classes for error handling and processing:

The EcUtStatus class is used to describe the operational status of many functions. The values most often reported are "failed" and "ok." But depending upon the application, detailed values could be set and sent. Please refer to the definition of this class (located in /ecs/formal/COMMON/CSCI_Util/src/Logging/EcUtStatus.h) for all possible values.

The EcPoError class defines the basic error types and handling functions for using the EcPoConnectionsRW class (based upon RWDBTool++). The Subscription Server and MSS Order tracking Server use the EcPoConnectionsRW class.

The MsAcDatabaseError class defines the error types and access Application Program Interfaces (APIs) for MSS database access.

The RWCString is used to store some status value returned by applications.

Integer is used to return some error status by applications. This is used specifically between client and server communications.

4.9.1 Management Software Computer Software Configuration Item Description

4.9.1.1 Management Software Functional Description

The Management Software CSCI (MCI) provides distributed system management support capabilities in the fault, configuration, accountability, performance, and security service areas. Its Computer Software Components (CSCs) include:

- Network and Enterprise Management Framework: This CSC enables M&O to monitor and control communications devices, hosts, and applications in the distributed system. It also provides the framework for integrating a range of other management service applications.
- Security: The security service is implemented using a variety of free-ware or public domain packages which monitor and evaluate the various aspects of the security setup at each site and reports status.
- Accountability Management: The accountability management support is provided by custom developed software for user registration and user profile attribute updates. The accountability management CSC also provides a tracking mechanism for user product orders.
- Trouble Ticketing: The Trouble Ticketing CSC manages system problem reports submitted by users and by external systems. The trouble ticket CSC also records problem assignees, tracks investigation progress, and provides users with problem resolution status.
- Network Backup/Restore: The Network backup and restore CSC enables the Operations staff to perform system backups and restores from a central administration position (at each site).

4.9.1.1.1 MCI – Network and Enterprise Management Computer Software Component Description

The Management Software CSCI (MCI) is COTS and custom software enabling the Operations staff to monitor and coordinate the EMD services. The MCI has the following CSCs:

1. Network and Enterprise Management Framework
2. Security Service
3. Accountability Management
4. Network Backup/Restore
5. ASTER E-mail Header Handler

4.9.1.1.1.1 Network and Enterprise Management Framework Functional Overview

The network and enterprise management framework is a capability to monitor network devices and services and notify Operations staff when problems are detected. It consists of the Big Brother monitoring and notification COTS product. Big Brother can discover and map a site's network devices automatically and track their status using its integrated capabilities.

4.9.1.1.1.2 Network and Enterprise Management Framework Context

Figure 4.9-2 is the Network and Enterprise Management Framework context diagram. The diagram shows the events sent to the Network and Enterprise Management Framework CSC and the events the Network and Enterprise Framework CSC sends to other CSCIs or CSCs. Table 4.9-2 provides descriptions of the interface events shown in the Network and Enterprise Management Framework context diagram.

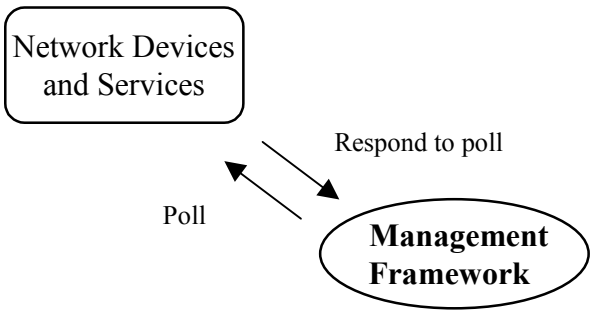


Figure 4.9-2. Network and Enterprise Management Framework Context Diagram

Table 4.9-2. Network and Enterprise Management Framework Interface Events

Event	Interface Event Description
Poll	Network and enterprise management framework polls devices and services on the devices to check on their status.
Respond to poll	Network devices and services acknowledge polling requests if they can.

4.9.1.1.1.3 Network and Enterprise Management Framework Process Architecture

Figure 4.9-3 is the Network and Enterprise Management Framework architecture diagram. The diagram shows the events sent to the Network and Enterprise Framework CSC processes and the events the Network and Enterprise Framework processes send to other processes.

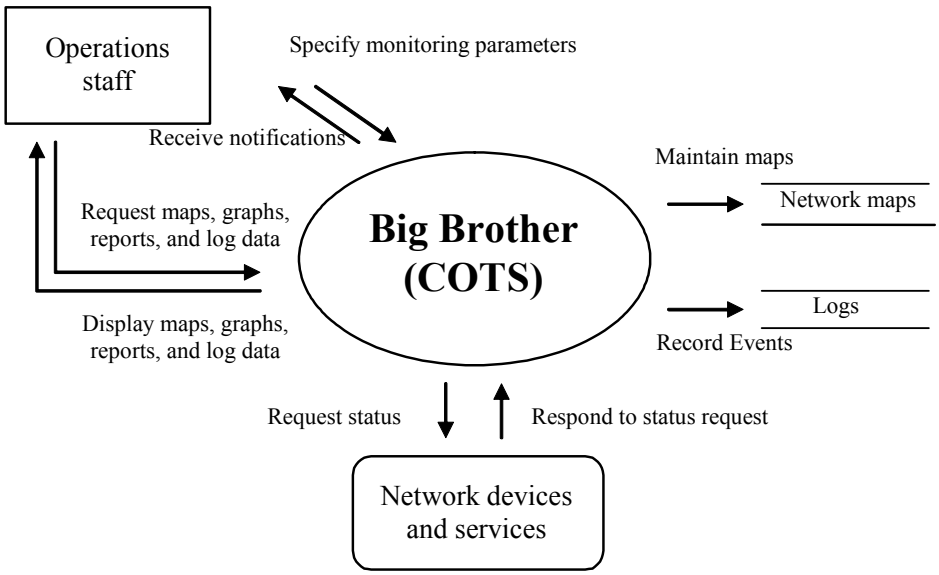


Figure 4.9-3. Network and Enterprise Management Framework Architecture Diagram

4.9.1.1.1.4 Network and Enterprise Management Framework Process Descriptions

Table 4.9-3 describes the processes in the Network and Enterprise Management Framework architecture diagram.

Table 4.9-3. Network and Enterprise Management Framework Processes

Process	Type	Hardware CI	COTS/ Developed	Functionality
Fault and Performance Monitor	Server	MSSHW	COTS	The Fault and Performance Manager discovers network devices and services; determines their status, recording polling requests and responses as events; compiles event statistics, issues alarms when problems are detected; and produces graphs and reports in response to Operations staff requests.

EMD Baseline Information System (EBIS) Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.9.1.1.1.5 Network and Enterprise Management Framework Process Interface Descriptions

Table 4.9-4 provides descriptions of the Network and Enterprise Management Framework interface events shown in the Network and Enterprise Management Framework architecture diagram.

Table 4.9-4. Network and Enterprise Management Framework Process Interface Events (1 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
Specify monitoring parameters	One per operator request	*Process:* Fault and Performance Monitor (COTS)	Operations staff	Big Brother permits operators to specify: • Devices and services to monitor • Polling methods and intervals • Subnets • Map colors and views • Alerts to be issued for each device and the triggering criteria for each • Notification types and content • Events to ignore

Table 4.9-4. Network and Enterprise Management Framework Process Interface Events (2 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
Request status	One per device per operator-specified time period	*Process:* Network devices and services	*Process:* Fault and Performance Monitor (COTS)	Big Brother polls a device by sending packets (echo requests) to the device using a monitoring method, polling frequency, time period, and timeout value specified by the operator. It monitors a service by communicating with the default port that the service runs on.
Respond to status requests	One per polling request	*Process:* Fault and Performance Monitor (COTS)	*Process:* Network devices and services	Network devices and services acknowledge a Big Brother status request.
Maintain maps	One per polling or update request	File	*Process:* Fault and Performance Monitor (COTS)	Big Brother creates and maintains network maps using auto discovery or parameters specified by the operator, and it uses polling results to continually update properties of the icons the maps contain.
Record events	One per polling request	Log	Fault and Performance Monitor (COTS)	Big Brother provides notification of device and service problems on devices.
Receive notifications	One per problem detected	Operations staff	*Process:* Fault and Performance Monitor (COTS)	When configured by an operator, Big Brother provides notification of device and service problems on devices to users.

Table 4.9-4. Network and Enterprise Management Framework Process Interface Events (3 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
Request graphs, reports, and log data	One per operator request	*Process:* Fault and Performance Monitor (COTS)	Operations staff	Big Brother permits operators to request: • Current status information about a device • Performance graphs by map, type (comprehensive, daily, monthly, day of the week, daily text/average, availability, hourly), device, graph format (bar or area chart), date range, data source, and output file type. Data in performance graphs can be sorted in ascending or descending device name order, and a report's view size can be adjusted • Event reports by map, type (summary, detail, or raw data) and date range. Data in summary and detail reports can be sorted in ascending, descending, or "worst first" order • Statistics reports by map, type (detail (by device), raw data, or day of the week) and date range. Data in detail reports can be sorted in ascending or descending device name order • Displays of Big Brothers's logs

Table 4.9-4. Network and Enterprise Management Framework Process Interface Events (4 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
Display requested maps, graphs, reports, and logs	One per operator request	Operations staff	*Process:* Fault and Performance Monitor (COTS)	In response to operator requests, Big Brother provides operators with: • Maps of the devices in a network • Up-to-the-minute status of a device, including such information as its up/down state, number of times it was polled, round trip time of the last polling packet sent and received, poll history, and poll success rate • Performance graphs that show devices by best or worst performance based on aggregated polling statistics. The graphs can show summaries of device and service availability and response times • Event reports that show device up and down events, service up and down events • Statistics reports that show round trip times and percentage of missed polls based on the accumulated polling statistics for each device • A GUI with which to browse Big Brother's log files

4.9.1.1.1.6 Network and Enterprise Management Framework Data Stores

Table 4.9-5 provides descriptions of the data stores used in the Network and Enterprise Management Framework architecture diagram.

Table 4.9-5. Network and Enterprise Management Framework Data Stores

Data Store	Type	Functionality
Syslog	File	Holds standard UDP messages sent from routers, switches, UNIX hosts, etc. Name is SL-*yyyy-mm-dd*.tab.
Event Log	File	Holds records that describe changes to network status, such as a device going down or coming back up. Name is EV-*yyyy-mm-dd*.tab.
Statistics Log	File	Holds polling statistic records – the accumulated round trip times of polls sent to a device – used to measure the availability and performance of a device. Name is ST-*yyyy-mm-dd*.tab.
SNMP Trap Log	File	Holds records of all SNMP traps that have been received. Name is SP-*yyyy-mm-dd*.tab.
Network Maps	File	Contains a network map's definition. Name is <*map_name*>.wup.

4.9.1.1.2 MCI – Security Service Computer Software Component Description

4.9.1.1.2.1 Security Service Functional Overview

Security Service monitoring in the EMD is accomplished through several commercial and public domain programs. The programs vary from aiding in administration of CCS Middleware, assisting the user in choosing a password difficult to break, monitoring key system files for signs of tampering and probing hosts for well known security violations.

4.9.1.1.2.2 Security Service Context

Figure 4.9-4 is the Security Service context diagram. The diagram shows the events sent to the Security Service from the host operating system, communications devices, and the M & O staff and the events the Security Service sends to the M & O staff. Table 4.9-6 provides descriptions of the interface events shown in the Security Service context diagram.

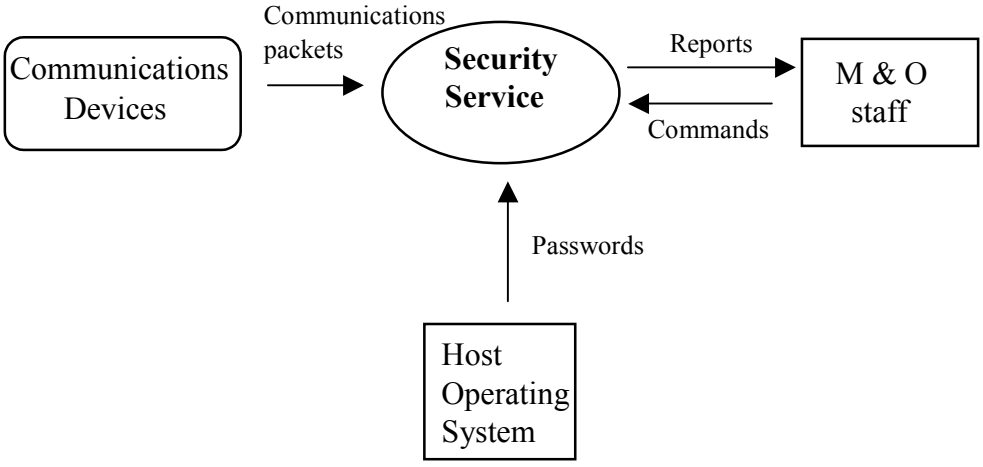


Figure 4.9-4. Security Service Context Diagram

Table 4.9-6. Security Service Interface Events

Event	Interface Event Description
Reports	The Security Service CSC utilities perform their functions and report results to the **M&O Staff**.
Commands	The **M&O Staff** issue commands to the Security Service CSC utilities to exercise system security setup.
Passwords	A password list is obtained from the Network Information Service (NIS) master (in the **host operating system**) by issuing a ypcat password command. This list is analyzed to see if decryption of a password is possible.
Communications packets	A packet reaches the EMD host from either an external source or from a host within the same site (**communications devices**). The Security Service CSC analyzes packets for authorized sending sources.

4.9.1.1.2.3 Security Service Architecture

Figure 4.9-5 is the Security Service architecture diagram. The diagram shows the events sent to the Security Service processes and the events the Security Service processes send to other processes.

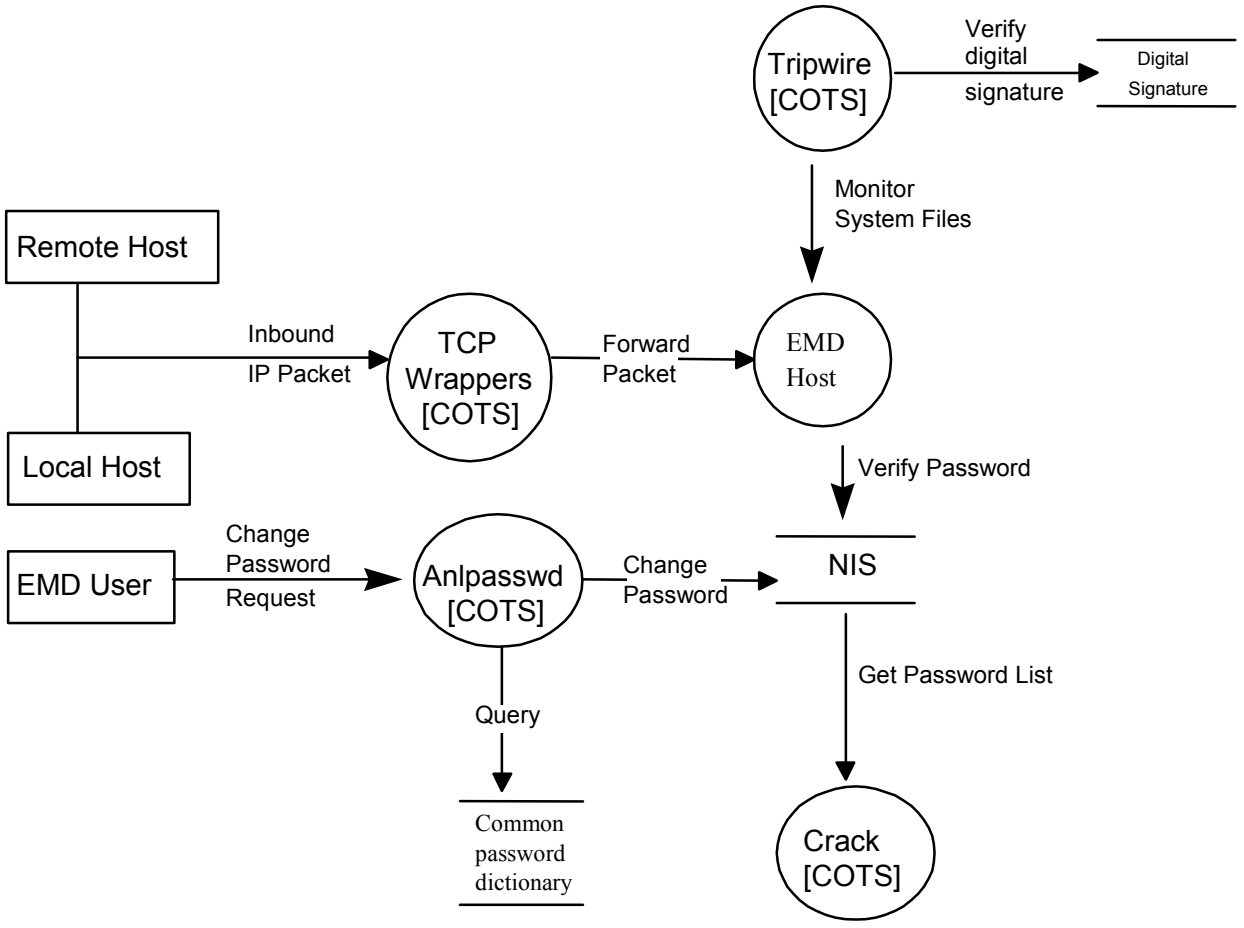


Figure 4.9-5. Security Service Architecture Diagram

4.9.1.1.2.4 Security Service Process Descriptions

Table 4.9-7 provides descriptions of the processes shown in the Security Service architecture diagram.

Table 4.9-7. Security Service Processes

Process	Type	Hardware CI	COTS / Developed	Functionality
Anlpasswd	Other	ACMHW, DIPHW, DRPHW, SPRHW, MSSHW, AITHW, CSSHW, DMGHW, DPSHW, INTHW, DPLHW, OMSHW	COTS	Anlpasswd is a replacement for the standard UNIX passwd and yppasswd programs. Anlpasswd provides functionality by checking the selected user password to determine if the password is common or trivial and easy to break.
TCP Wrappers	Other	ACMHW, DIPHW, DRPHW, SPRHW, AITHW, CSSHW, DMGHW, MSSHW, DPSHW, INTHW, OMSHW, DPLHW	COTS	TCP Wrappers verifies the origin of incoming IP packets from an authorized host for services TCP Wrappers can filter. TCP Wrappers runs on each EMD host (UNIX) at a specific site.
Tripwire	Other	ACMHW, DIPHW, DRPHW, SPRHW, MSSHW, AITHW, CSSHW, DMGHW, DPSHW, INTHW, DPLHW, OMSHW	COTS	Tripwire periodically verifies that system files have not been altered. Tripwire is able to catch modifications by verifying the current and stored digital signatures of the command.
Crack	Other	CSSHW	COTS	An M&O Staff member runs Crack periodically to search for passwords that can be broken and were not caught by Anlpasswd.

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.9.1.1.2.5 Security Service Process Interface Descriptions

Table 4.9-8 provides descriptions of the interface events shown in the Security Service architecture diagram.

Table 4.9-8. Security Service Process Interface Events

Event	Event Frequency	Interface	Initiated By	Event Description
Verify digital signature	Once per verify digital signature	*Data Store:* Digital Signature	*Process:* Tripwire (COTS)	The newly computed digital signature of a file is verified against the stored historical copy of the same file via a Tripwire internal call.
Monitor system files	One per monitor system files	EMD Host	*Process:* Tripwire (COTS)	Critical system files are watched periodically for changes in their digital signature that could signal a maliciously altered system file or service on an **EMD Host**.
Forward Packet	One per forward packet	Inetd – UNIX daemon on an EMD Host	TCP Wrappers (COTS)	If the IP header indicates the packet originates from a host that has not been blocked by TCP Wrappers, the packet is forwarded via the appropriate internet service to an **EMD Host**.
Verify password	Once per verify password	*Data Store:* NIS	EMD Host	A request is sent from the **EMD host**, via a NIS system call, to the NIS database to verify that a login password is valid.
Get password list	One per get password list	*Data Store:* NIS	*Process:* Crack (COTS)	A password list is obtained from the **NIS** master by issuing a ypcat passwd command (NIS system call). This list is run through crack to see if crack is able to decrypt any user's password.
Change password	One per change password	*Data Store:* NIS	*Process:* Anlpasswd (COTS)	After the new password passes the Anlpasswd validation process, a request is sent to the **NIS** master, via a NIS system call, to modify the user's password.
Query	One per query	*Data Store:* Common Password Dictionary	*Process:* Anlpasswd (COTS)	The Anlpasswd makes a NIS system call to check the **common password dictionary** to ensure the attempted new password is not in this list.
Change password request	One per change password request	*Process:* Anlpasswd	User/Command line	An **EMD user** attempts to change their password and the Anlpasswd verifies the request that the new password does not contain any trivial or easy to guess password.
Inbound IP packet	One per inbound IP packet	*Process:* TCP Wrappers (COTS)	Remote or Local Host	A packet reaches the TCP Wrappers process from either an external source (**remote host**) or from a **host** within the same site via TCP/IP protocols.

4.9.1.1.2.6 Security Service Data Stores

Table 4.9-9 provides descriptions of the data stores shown in the Security Service architecture diagram.

Table 4.9-9. Security Service Data Stores

Data Store	Type	Functionality
NIS	Database	This UNIX service enables a common login on a number of machines and mapping for a user's Network File System (NFS) mounted home directory. The passwd map stores a user's login id, group id, and password in the NIS database.
Common password dictionary	Database	This sorted text file contains common words used by a user as a password. Anlpasswd verifies that the new password change does not include a word listed in the Anlpasswd file.
Digital Signature	Database	This proprietary database is used by Tripwire to record the digital signature for each system file it monitors.

4.9.1.1.3 MCI - Accountability Management Service Computer Software Component Description

4.9.1.1.3.1 Accountability Management Service Functional Overview

The Accountability Management Service supports User Registration and Order Tracking.

User Registration

EMD provides for two generic classes of users: guest users and registered users. Guest users are not formally registered. Registered users have submitted requests for a registered user account and have accounts, based on an approval process. Registered users can access services and products beyond those available to guest users.

The user registration server supports the creation, modification and maintenance of profiles for each registered user. The user profile is maintained at the SMC and replicated at each DAAC. Each DAAC is capable of browsing foreign user profiles locally, but only capable of modifying user profiles at the SMC for which it is the designated home DAAC.

The user registration GUI enables the DAAC Operations staff to view registered user profiles (at the DAAC and the SMC). The user profile information includes the user's name, identification code, primary DAAC, organizational affiliation, investigating group (such as an instrument team) affiliation (if any), assigned project, mailing address, shipping address for data or product order distribution media preferences for product orders, telephone number and electronic mail address (if any).

The Accountability Management Service enables the various subsystems to request user profile information such as the user's electronic mail address and the shipping address for product order or data distribution.

4.9.1.1.3.2 Accountability Management Service Context

Figure 4.9-6 is the Accountability Management context diagram. The diagram shows the events sent to the Accountability Management and the events the Accountability Management sends to

other CSCIs or CSCs. Table 4.9-10 provides descriptions of the interface events shown in the Accountability Management context diagram.

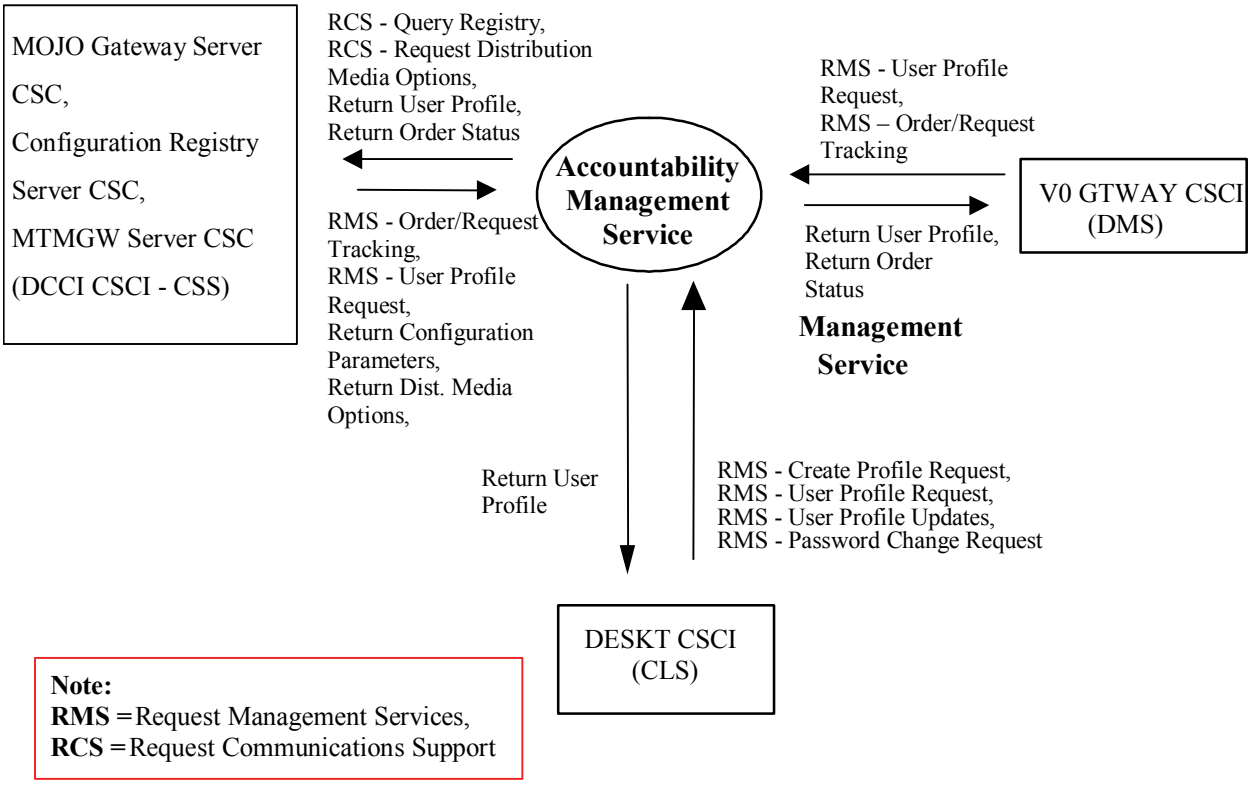


Figure 4.9-6. Accountability Management Service Context Diagram

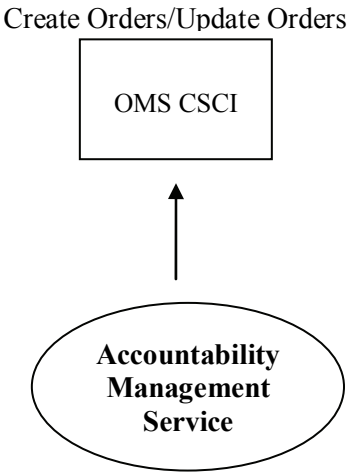


Figure 4.9-6. Accountability Management Service Context Diagram (cont.)

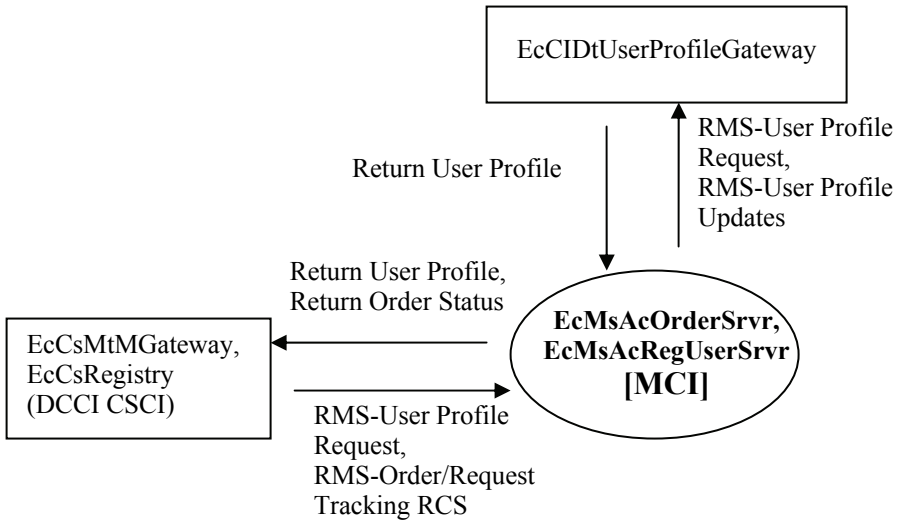


Figure 4.9-6. Accountability Management Service Context Diagram (cont.)

Table 4.9-10. Accountability Management Service Interface Events (1 of 3)

Event	Interface Event Description
Request Management Services	The MCI provides a basic management library of services to the CSCIs/CSCs, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services include: The MCI interfaces with other CSCIs/CSCs to perform the following: • **Create Profile Request** - The MCI receives user information for becoming a registered user of the EMD from the DESKT CSCI. • **Order/Request Tracking** – The **MTMGW CSC** can create and track order or search and order requests received from the SIPS interface via the MCI Order Tracking Server. The **V0 GTWAY CSCI** interfaces with Accountability Management Service Order/Request Tracking service to create a user product order. • **User Profile Request** – The Accountability Management Service provides the **V0 GTWAY**, **DESKT** CSCI and the **MTMGW Server** CSC with User Profile information such as e-mail address and shipping address to support their processing activities. • **User Profile Updates** – The MCI receives user profile parameter updates from a user (via the **DESKT CSCI**) and makes the updates in the user profile database. This capability is available only at the SMC. • **Password Change Request** – The **DESKT CSCI** sends requests on behalf of EMD users to the MCI to change or reset users' authenticators in the MSS database.
Return User Profile	The **MTMGW Server CSC** and the **DESKT CSCI** receive user profile information from the Accountability Management Service to authenticate a user.
Return Order Status	The **MTMGW Server CSC, DESKT CSCI** and **V0 GTWAY CSCI** receive an order id and status for the requested EMD product from the Accountability Management Service.
Return Configuration Parameters	The **Configuration Registry CSC** returns the requested configuration parameters to the Accountability Management Service.
Return Dist. Media Options	The Accountability Management Service receives the distribution media options from the **Configuration Registry Server CSC**.

Table 4.9-10. Accountability Management Service Interface Events (2 of 3)

Event	Interface Event Description
Request Communications Support	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS CSCI/CSC. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • Network & Distributed File Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry • Request Distribution Media Options from the Configuration Registry
Request Management Services	The **MCI** provide a basic management library of services to the CSCIs/CSCs, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services include: The MCI interfaces with other CSCIs/CSCs to perform the following: • **User Profile Request** - The Accountability Management Service provides **the CGI Scripts** with User Profile information such as e-mail address and shipping address to support their processing activities.
Return User Profile	The Accountability Management Service CSC returns the requested user profile to a **CGI script**.
Return Order Status	The Accountability Management Service CSC returns the order status for the requested product to a **CGI script**.

Table 4.9-10. Accountability Management Service Interface Events (3 of 3)

Event	Interface Event Description
Send Profile Request	The Sun Java System Web Server sends the request information for a user profile to a CGI script.
Send Validated ODP Request	The Sun Java Web Server sends a request to a CGI script to access the MSS database. The EcClOdRequest process accesses the MSS database via CGI scripts and sends the user back the authentication.
Send Profile	The CGI scripts provide the user profile to the Sun Java System Web Server.
Send Order Status	The CGI scripts provide an order id and status for the requested EMD product to the Sun Java System Web Server.

4.9.1.1.3.3 Accountability Management Service Architecture

Figure 4.9-7 is the Accountability Management Service architecture diagram. The diagram shows the events sent to the Accountability Management Service processes and the events the Accountability Management Service processes send to other processes.

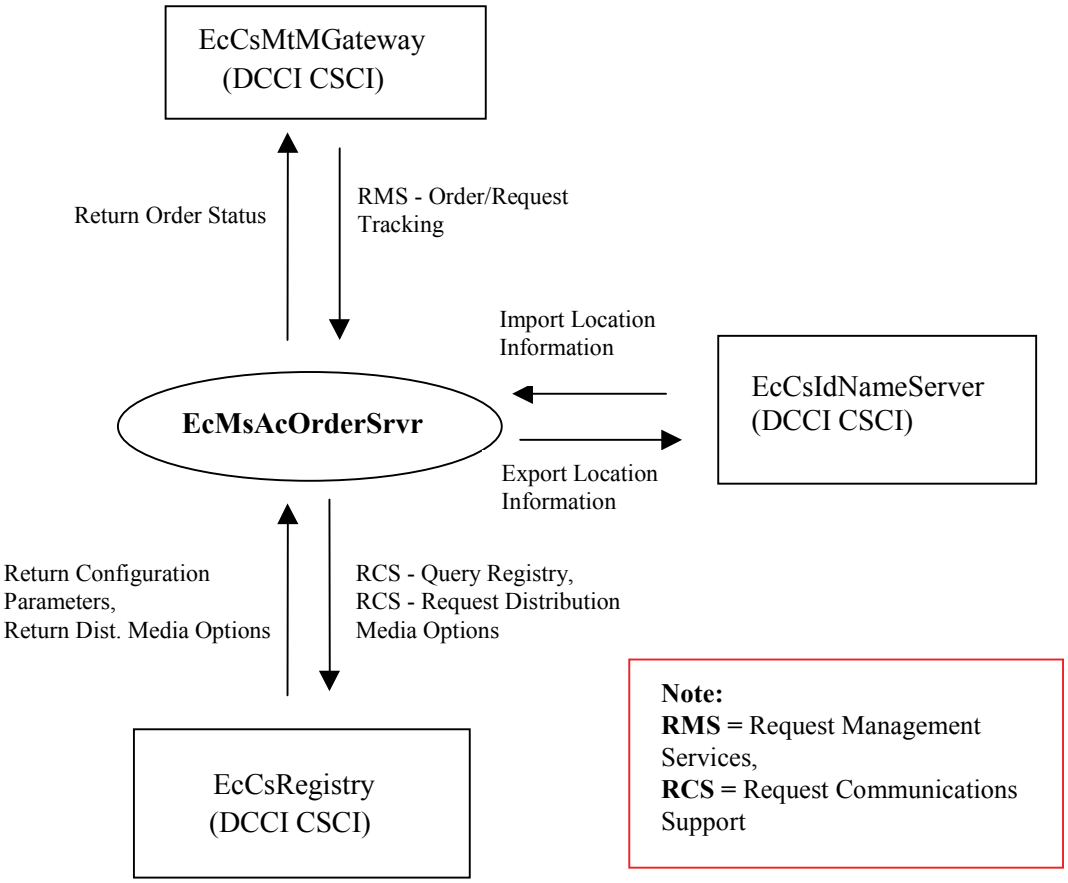


Figure 4.9-7. Accountability Management Service Architecture Diagram

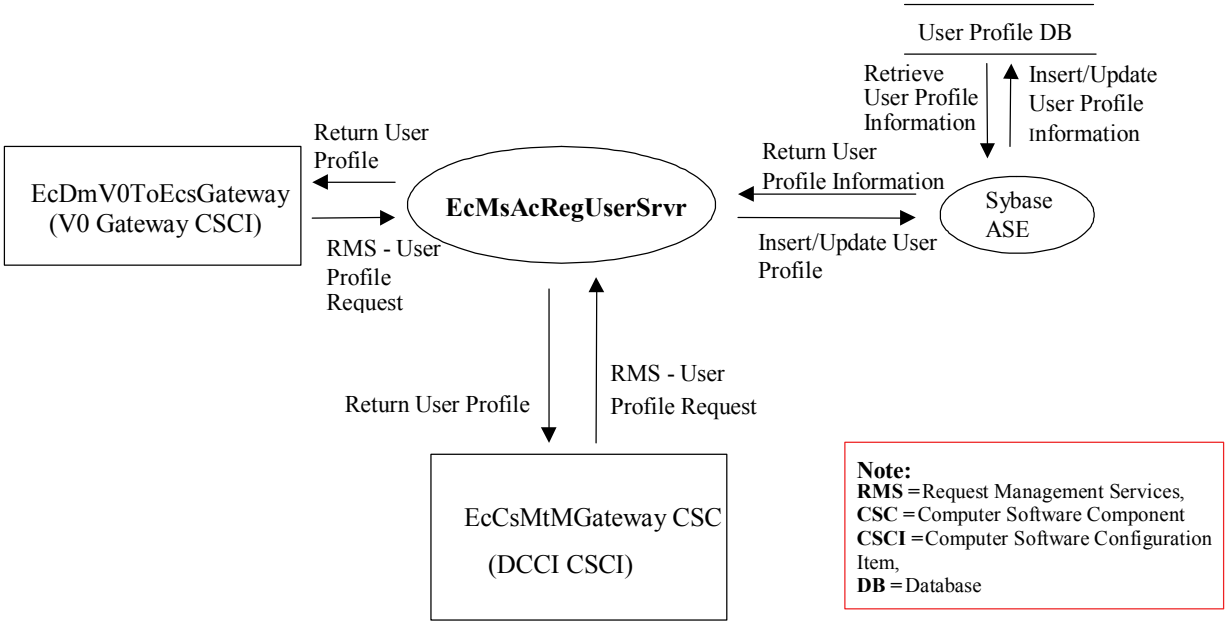


Figure 4.9-7. Accountability Management Service Architecture Diagram (cont.)

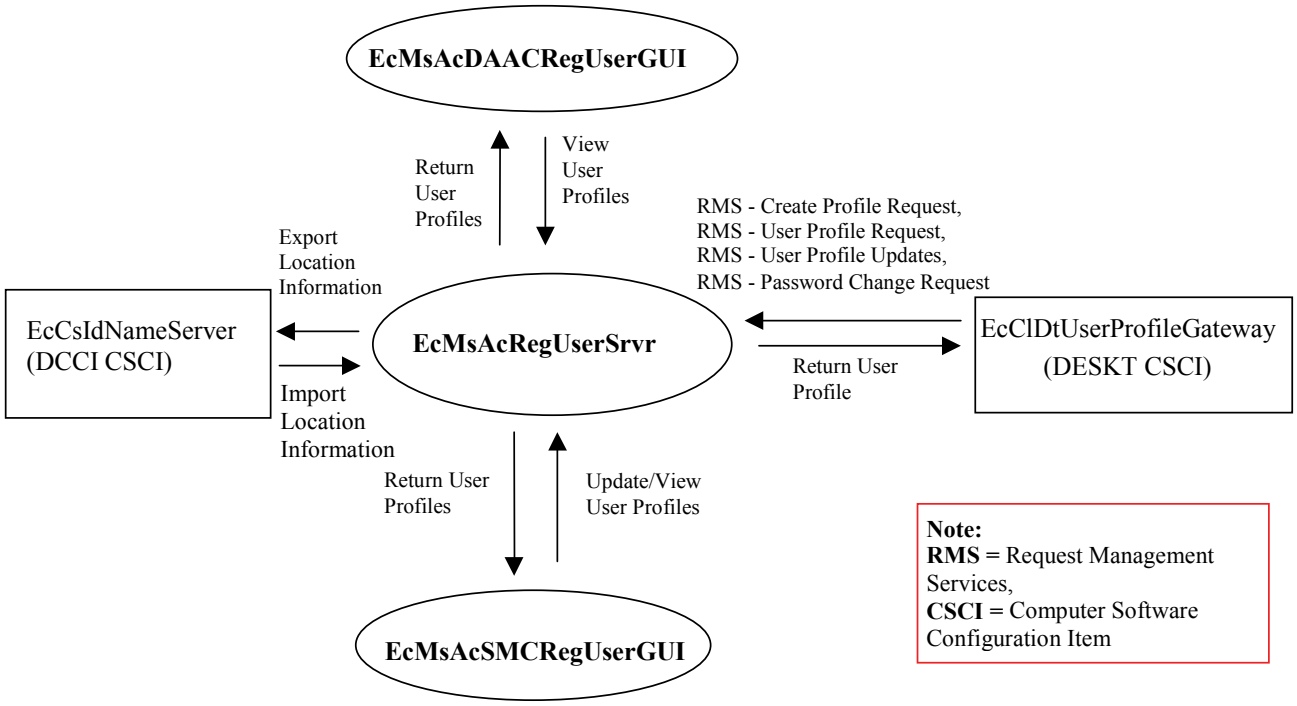


Figure 4.9-7. Accountability Management Service Architecture Diagram (cont.)

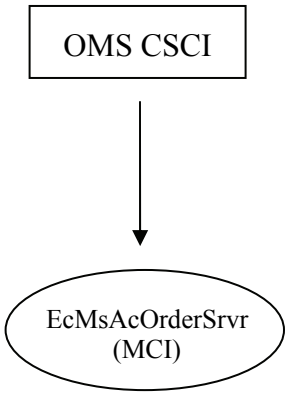


Figure 4.9-7. Accountability Management Service Architecture Diagram (cont.)

4.9.1.1.3.4 Accountability Management Service Processes

Table 4.9-11 provides the descriptions of the processes shown in the Accountability Management Service architecture diagram.

Table 4.9-11. Accountability Management Service Processes

Process	Type	Hardware CI	COTS / Developed	Functionality
EcMsAcRegUserSrvr	Server	OMSHW	Developed	The User Registration Server provides an internal interface to the User Registration GUI and an external interface to other CSCIs/CSCs. The functions are: 1. Insert, delete, update, retrieve registered user profile 2. Retrieve a list of registered user profiles 3. Change V0 gateway password The EcMsAcRegUserSrvr supports: • Single requests at a time • Multiple concurrent requests • Asynchronous request processing • Multiple threads within a single request
EcMsAcOrderSrvr	Server	OMSHW	Developed	The Order Tracking Server provides an external interface to other CSCIs/CSCs. The functions are: 1. Insert, delete, update, retrieve order 2. Insert, delete, update, retrieve request 3. Retrieve a list of orders 4. Retrieve a list of requests 5. Update order status 6. Update request status The EcMsAcOrderSrvr supports: • Single requests at a time • Multiple concurrent requests • Asynchronous request processing • Multiple threads within a single request
Sybase ASE	Server	ACMHW	COTS	The Sybase ASE supports access to the Sybase ASE DBMS. The interface between processes and the databases for storage and retrieval of data or information.

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.9.1.1.3.5 Accountability Management Service Process Interface Descriptions

Table 4.9-12 provides descriptions of the interface events shown in the Accountability Management Service architecture diagram.

Table 4.9-12. Accountability Management Service Process Interface Events (1 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Display Order Status	One per order request	*Process:* EcMsAcOrderGUI *Libraries:* MsAcClnt, MsAcComm	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	The EcMsAcOrderSrvr provides the order status information to the EcMsAcOrderGUI for display by the Operations Staff.
Display Order Information	One per order request	*Process:* EcMsAcOrderGUI *Libraries:* MsAcClnt, MsAcComm	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	The EcMsAcOrderSrvr provides the order information to the EcMsAcOrderGUI for display by the Operations Staff.
Insert Product Order Information	One per order request	*Data Store:* Order Tracking DB	*Process:* Sybase ASE (COTS)	The **Sybase ASE** inserts product order information in the Order Tracking DB.
Update Product Order Information	One per order request	*Data Store:* Order Tracking DB	*Process:* Sybase ASE (COTS)	The **Sybase ASE** updates product order information in the Order Tracking DB.
Retrieve Product Order Information	One per order request	*Data Store:* Order Tracking DB	*Process:* Sybase ASE (COTS)	The **Sybase ASE** retrieves product order information in the Order Tracking DB.
Insert Order Request	One per insert order request	Sybase ASE (COTS)	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	The EcMsAcOrderSrvr submits a request to the **Sybase ASE** to insert a product order request into the Order tracking database (DB).
Retrieve Order Status	One per product order	Sybase ASE (COTS)	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	The EcMsAcOrderSrvr sends a request to the **Sybase ASE** to retrieve the status of an order placed by a user.

Table 4.9-12. Accountability Management Service Process Interface Events (2 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Update Order Information	One per update of order information	Sybase ASE (COTS)	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	The EcMsAcOrderSrvr submits a request to the **Sybase ASE** to update order information in the Order tracking database (DB).
Retrieve Order Information	One per update of order information	Sybase ASE (COTS)	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	The EcMsAcOrderSrvr submits a request to the **Sybase ASE** to retrieve order information from the Order tracking database (DB).
Return Order Information	One per return of order information	*Process:* EcMsAcOrder Srvr *Libraries:* MsAcClnt, MsAcComm	*Process:* Sybase ASE (COTS)	The **Sybase ASE** returns product order information per operations request to the EcMsAcOrderSrvr.
Return Order Status Information	One per return order status	*Process:* EcMsAcOrder Srvr *Libraries:* MsAcClnt, MsAcComm	*Process:* Sybase ASE (COTS)	The **Sybase ASE** returns product order status per operations request to the EcMsAcOrderSrvr.
Request Management Services (RMS)	One per service request	N/A	N/A	The EcMsAcOrderSrvr provides a basic management library of services to the processes, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes the items below.

Table 4.9-12. Accountability Management Service Process Interface Events (3 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
(RMS – cont.)	At system startup or shutdown and for restarts	*Processes:* EcMsAcOrderSrvr	DAAC unique startup scripts	**System startup and shutdown -** Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625).
(RMS – cont.)	One order per request	*Process:* EcMsAcOrderSrvr *Library:* MsAcClnt *Class:* EcAcOrderCMgr	*DMS Process:* EcDmV0ToEcsGateway *Class:* DmGwRequestReceiver	**Order/Request Tracking** - The **EcDmV0ToEcsGateway** interfaces with the MCI Order/Request Tracking service to create a user product order and track the order.
Return Order Status	One per user request	*DMS Process:* EcDmV0ToEcsGateway *Class:* DmGwRequestReceiver	*Process:* EcMsAcOrderSrvr *Library:* MsAcClnt *Class:* EcAcOrderCMgr	The **EcDmV0ToEcsGateway** processes receive the order id and status from the EcMsAcOrderSrvr.
Request Order Status	One per order request	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	*Process:* EcMsAcOrderGUI *Libraries:* MsAcClnt, MsAcComm	The EcMsAcOrderGUI sends requests for order status to the EcMsAcOrderSrvr.
Request Order Information	One per order request	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	*Process:* EcMsAcOrderGUI *Libraries:* MsAcClnt, MsAcComm	The EcMsAcOrderGUI sends requests for order information to the EcMsAcOrderSrvr.

Table 4.9-12. Accountability Management Service Process Interface Events (4 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Order Information Update	One per order request	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	*Process:* EcMsAcOrderGUI *Libraries:* MsAcClnt, MsAcComm	The EcMsAcOrderGUI sends requests for order information updates to the EcMsAcOrderSrvr.
Request Management Services (RMS)	One per service request	N/A	N/A	The EcMsAcOrderSrvr provides a basic management library of services to the processes, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes the items below.
(RMS – cont.)	One order per SIPS user request	*Process:* EcMsAcOrderSrvr *Library:* MsAcClnt *Class:* EcAcOrderCMgr	*Processes:* EcCsMtMGateway, EcMsAcOrderSrvr *Library:* MsAcClnt, MsAcComm	**Order/Request Tracking** - The **MTMGW CSC** interfaces with the MCI Order/Request Tracking service to create a user product order and track the progress of the order.

Table 4.9-12. Accountability Management Service Process Interface Events (5 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Import Location Information	As required for processing	*Process:* EcCsIdName Server *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManaged Server, CCSMdwNam eServer, FoNsNameSe rverProxy, CCSMdwRwN etProxy	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	The EcMsAcOrderSrvr requests physical and logical server location information from the **EcCsIdNameServer**.
Export Location Information	Once at system startup and after each restart	*Process:* EcCsIdName Server *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManaged Server, CCSMdwNam eServer, FoNsNameSe rverProxy, CCSMdwRwN etProxy	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	The EcMsAcOrderSrvr stores physical and logical server location information in the **EcCsIdnameServer**.

Table 4.9-12. Accountability Management Service Process Interface Events (6 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Communications Support	Request service(s) as required	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy *Library (Common):* EcUr *Class:* EcUrServerUR *Library:* event *Class:* EcLgErrorMsg *Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS CSCI/CSC. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry - Retrieving the requested configuration attribute-value pairs from the Configuration Registry • Request Distribution Media Options from the Configuration Registry

Table 4.9-12. Accountability Management Service Process Interface Events (7 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Return Configuration Parameters	One set per request	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	*Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	The **EcCsRegistry** returns the attribute-value pairs (configuration parameters) to the EcMsAcOrderSrvr upon request.
Return Dist. Media Options	One set per request	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	*Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	The **EcCsRegistry** returns the requested distribution media options to the EcDmV0ToEcsGateway.
Return Order Status	One per user request	*Process:* EcCsMtMGateway *Class:* EcCsMtMAcctSrvMgr	*Processes:* EcCsMtMGateway, EcMsAcOrderSrvr *Library:* MsAcClnt, MsAcComm	The **EcCsMtMGateway** receives the order id and status from the EcMsAcOrderSrvr.
Request Management Services (RMS)	One per service request	N/A	N/A	The EcMsAcOrderSrvr and EcMsAcRegUserSrvr provide a basic management library of services to the processes, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes the items below.
(RMS – cont.)	At system startup or shutdown and for restarts	*Processes:* EcMsAcRegUserSrvr	DAAC unique startup scripts	**System startup and shutdown -** Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625).

Table 4.9-12. Accountability Management Service Process Interface Events (8 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
(RMS – cont.)	One profile per request per session	*Process:* EcMsAcRegUserSrvr, *Libraries:* MsAcClnt, MsAcComm *Class:* EcAcProfileMgr	*DMS Process:* EcDmV0ToEcsGateway *Class:* DmGwRequestReceiver *Library:* DsDdSSh *Class:* DsDdMedia	**User Profile Request** - The EcMsAcRegUserSrvr provides requesting processes with user profile information such as e-mail address and shipping address to support their processing activities.
Insert/Update User Profile Information	One insert/update per request	*Data Store:* User Profile DB	*Process:* Sybase ASE (COTS)	The **Sybase ASE** stores or updates user profile information by request.
Retrieve User Profile Information	One per profile request	Sybase ASE (COTS)	*Data Store:* User Profile DB	The Sybase ASE obtains the user profile information requested by the M&O Staff from the User Profile DB.
Return User Profile Information	One per profile request	*Process:* EcMsAcRegUserSrvr *Libraries:* MsAcClnt, MsAcComm	Sybase ASE (COTS)	The **Sybase ASE** returns the user profile information to the EcMsAcRegUserSrvr to send back to the user.
Insert/Update User Profile Information	One per user insert/ update profile	Sybase ASE (COTS)	*Process:* EcMsAcRegUserSrvr *Libraries:* MsAcClnt, MsAcComm	The EcMsAcRegUserSrvr sends requests to the **Sybase ASE** to add or modify user profile data in the User Profile database (DB).

Table 4.9-12. Accountability Management Service Process Interface Events (9 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Return User Profile	One profile per request	*DMS Process:* EcDmV0ToEcsGateway *Class:* DmGwRequestReceiver *Class:* EcMjRetrieveProfileProxy, *CSS Process:* EcCsMtMGateway *Class:* EcCsMtMAcctSrvMgr	*Process:* EcMsAcRegUserSrvr, *Libraries:* MsAcClnt, MsAcComm *Class:* EcAcProfileMgr	The EcMsAcRegUserSrvr returns user profile information provided from the Sybase ASE to the requester.
View User Profiles	One per view user profile request	*Process:* EcMsAcRegUserSrvr *Libraries:* MsAcClnt, MsAcComm	M&O staff *Process:* EcMsAcDAACRegUserGUI *Libraries:* MsAcClnt, MsAcComm	The **M&O staff** request, via the EcMsAcDAACRegUserGUI, to retrieve a user profile from the EcMsAcRegUserSrvr.
Request Management Services (RMS)	One per service request	N/A	N/A	The EcMsAcRegUserSrvr provides a basic management library of services to the processes, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes the items below.
(RMS - cont.)	Per user request	*Process:* EcMsAcRegUserSrvr, *Libraries:* MsAcClnt, MsAcComm *Class:* EcAcProfileMgr	*CLS Process:* EcClDtUserProfileGateway *Class:* ClDtProfileServer	**Create Profile Request -** The **EcClDtUserProfileGateway** sends user information for becoming a registered user of the EMD to the EcMsAcRegUserSrvr. The EcMsAcRegUserSrvr sends a response to the user when the request is received.

Table 4.9-12. Accountability Management Service Process Interface Events (10 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
(RMS – cont.)	One profile per request per session	*Process:* EcMsAcRegUserSrvr, *Libraries:* MsAcClnt, MsAcComm *Class:* EcAcProfileMgr	***CLS Process:*** EcClDtUserProfileGateway *Class:* ClDtProfileServer	**User Profile Request** - The EcMsAcRegUserSrvr provides the requesting process (**EcClDtUserProfileGateway**) with user profile information such as e-mail address and shipping address to support their processing activities.
Table 4.9-12. Accountability Management Service Process Interface Events (RMS – cont.)	One profile per request per session One password change per configured period	*Process:* EcMsAcRegUserSrvr, *Libraries:* MsAcClnt, MsAcComm *Class:* EcAcProfileMgr	***CLS Process:*** EcClDtUserProfileGateway *Class:* ClDtProfileServer	**User Profile Updates** - The EcMsAcRegUserSrvr provides the requesting process (**EcClDtUserProfileGateway**) with updates to user profile parameters such as e-mail address and shipping address to support their processing activities. **Password Change Request -** The **EcClDtUserProfileGateway** sends requests on behalf of EMD users to the EcMsAcRegUserSrvr to change or reset users' authenticators in the MSS database.
Return User Profile	One per request	Process: EcClDtUserProfileGateway *Class:* ClDtProfileServer	*Process:* EcMsAcRegUserSrvr, *Libraries:* MsAcClnt, MsAcComm *Class:* EcAcProfileMgr	The EcMsAcRegUserSrvr returns user profile information provided from the Sybase ASE to the requester (**EcClDtUserProfileGateway**) .
Update/View User Profiles	One per update/view user profile	*Process:* EcMsAcRegUserSrvr *Libraries:* MsAcClnt, MsAcComm	M&O staff *Process:* EcMsAcSMCRegUserGUI *Libraries:* MsAcClnt, MsAcComm *Class:* MsAcRegUsrUtl	The **M&O staff** sends requests to modify or view user profile information via the EcMsAcSMCRegUserGUI.

Table 4.9-12. Accountability Management Service Process Interface Events (11 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Return User Profiles	One per return of user profile	*Processes:* EcMsAcSMCRegUserGUI, EcMsAcDAACRegUserGUI *Libraries:* MsAcClnt, MsAcComm	*Process:* EcMsAcRegUserSrvr *Libraries:* MsAcClnt, MsAcComm	The EcMsAcRegUserSrvr returns user profile information provided from the **Sybase ASE** to the requester at the EcMsAcSMCRegUserGUI or EcMsAcDAACRegUserGUI.
Import Location Information	As required for processing	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy	*Process:* EcMsAcRegUserSrvr *Libraries:* MsAcClnt, MsAcComm	The EcMsAcRegUserSrvr requests physical and logical server location information from the **EcCsIdNameServer**.

Table 4.9-12. Accountability Management Service Process Interface Events (12 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Export Location Information	Once at system startup and after each restart	*Process:* EcCsIdName Server *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManaged Server, CCSMdwNam eServer, FoNsNameSe rverProxy, CCSMdwRwN etProxy	*Process:* EcMsAcRegUserSrvr *Libraries:* MsAcClnt, MsAcComm	The EcMsAcRegUserSrvr stores physical and logical server location information in the **EcCsIdnameServer**.
Request Management Services (RMS	One per user request	*Process:* EcMsAcRegU serSrvr, *Libraries:* MsAcClnt, MsAcComm *Class:* EcAcProfileMg r	*Processes:* CGI scripts	**User Profile Request** - The EcMsAcRegUserSrvr receives requests from **CGI scripts** for user profile information such as e-mail address and shipping address to support their processing activities.
Update Order Status	One per product request	*Process:* Executing stored procedure: ProcUpdRequ estStatus	*Process:* Order Manager Server	By executing OMS Stored procedures the MSS stored procedures will be called.
Send Profile Request	One request per user	*Processes:* CGI scripts	*Process:* Sun Java System Web Server (COTS)	The Sun Java Web Server sends a request for a user profile to the CGI scripts on behalf of a user.

Table 4.9-12. Accountability Management Service Process Interface Events (13 of 13)

Event	Event Frequency	Interface	Initiated By	Event Description
Send Profile	One per user	*Process:* Sun Java Web Server (COTS)	*Processes:* CGI scripts	The CGI scripts provide the user profile to the Sun Java Web Server.
Send Order Status	One per request	*Process:* Sun Java Web Server (COTS)	*Processes:* CGI scripts	The CGI scripts provide an order id and status for the requested EMD product to the Sun Java Web Server.
Return User Profile	One per user	*Processes:* CGI scripts	*Process:* EcMsAcRegUserSrvr, *Libraries:* MsAcClnt, MsAcComm *Class:* EcAcProfileMgr	The EcMsAcRegUserSrvr returns a user profile to the CGI scripts for a user to access or use EMD products or services.

4.9.1.1.3.6 Accountability Management Service Data Stores

Table 4.9-13 provides descriptions of the data stores shown in the Accountability Management Service architecture diagram.

Table 4.9-13. Accountability Management Service Data Stores

Data Store	Type	Description
User Profile DB	Database	The User Profile DB contains the profile information including mailing addresses, e-mail address, and project affiliations of approved registered users.
Order Tracking DB	Database	The Order Tracking DB contains product orders and user requests with the associated current processing status.

4.9.1.1.4 MCI - Trouble Ticketing Computer Software Component Description

4.9.1.1.4.1 Trouble Ticketing Functional Overview

TestTrack Pro, commonly referred to as TTPro, implements the Trouble Ticketing service for ECS. GUIs provided with TTPro enable users and operators to enter and track trouble tickets affecting both local and system-wide resources. The delivered configuration also includes a trouble ticket ticket escalation mechanism, customized user and operator notifications, and status reports to aid in the problem resolution process.

4.9.1.1.4.2 Trouble Ticketing Context

Figure 4.9-8 is the Trouble Ticket (TT) context diagram. TTPro receives new trouble tickets from users and operators, as well as from legacy ECS trouble ticket systems, and records them in the project database configured for the site. Notifications are automatically sent via e-mail to people designated to receive them when trouble tickets are created, updated, and closed. System-level TTs are escalated in NCR format to the ECS Development Facility (EDF) for resolution by the EMD staff.

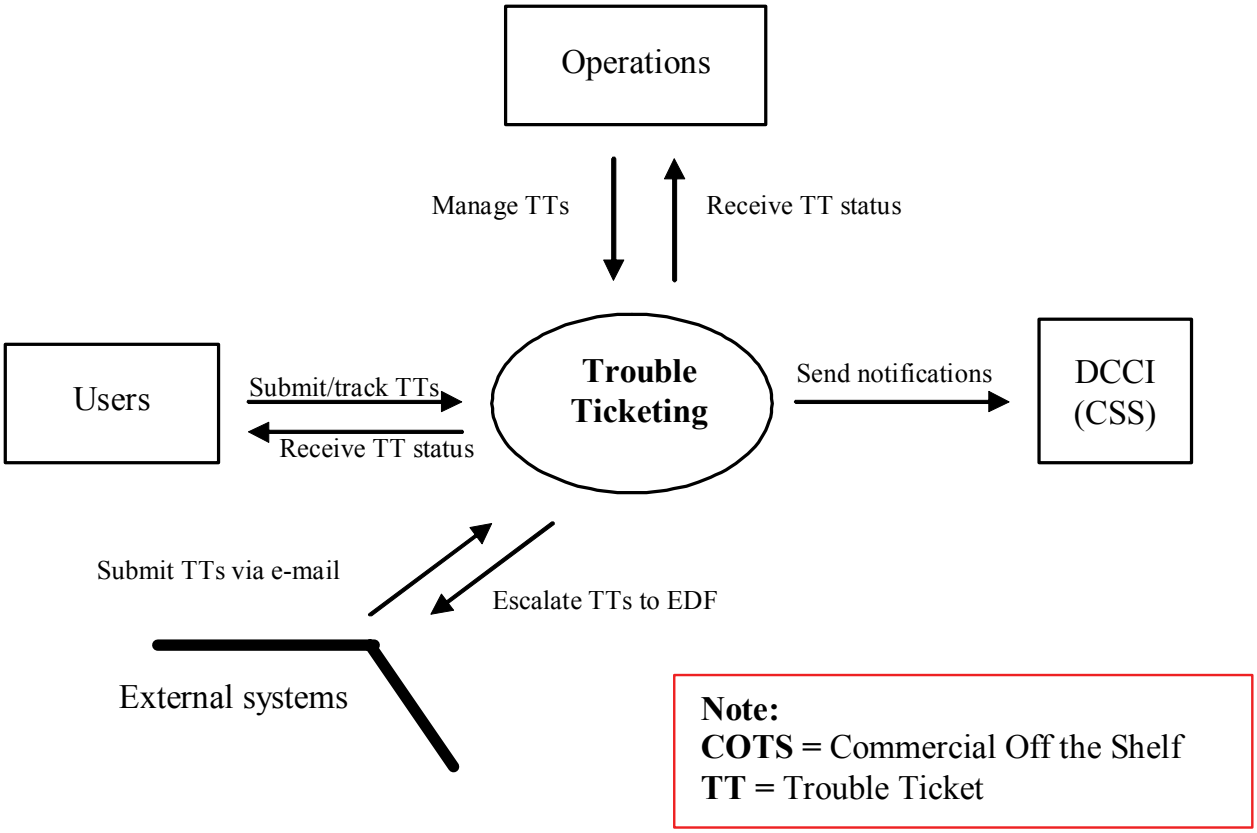


Figure 4.9-8. Trouble Ticketing Context Diagram

Table 4.9-14 provides descriptions of the interface events shown in the Trouble Ticket context diagram.

Table 4.9-14. Trouble Ticketing Interface Events

Event	Interface Event Description
Submit/track TTs	Users (**Users**) log trouble tickets and monitor their status as reported issues are worked and resolved.. *Windows-, Linux-, or Solaris-based clients connect directly to the Trouble Ticketing server. A CGI client provides a functionally equivalent interface through a Web server.*
Manage TTs	Operations staff (**Operations**) update TTs to document analyses, assignments, proposed solutions, and resolutions and advance the TTs through their lifecycle states.
Receive TT status	Users **(Users)** and operators (**Operations**) are presented the displays and reports of TT data needed for submitting, tracking, and managing TTs.
Send notifications	Trouble Ticketing sends configurable, system notifications to the system mail server for delivery to designated users and operations staff whenever a TT is created, changed, assigned, closed or escalated,to the EDF.
Escalate TTs to EDF	Trouble tickets whose resolution requires a change to ECS' design or implementation are escalated to the EDF in non-conformance report (NCR) format.
Submit TTs via e-mail	Trouble tickets are generated from e-mail messages received from other applications such as trouble ticketing systems at legacy ECS sites.

4.9.1.1.4.3 Trouble Ticketing Architecture

Figure 4.9-9 is the Trouble Ticket architecture diagram. The diagram shows the events sent to the TTPro COTS process and the events the TTPro COTS process sends to other processes (TTPro GUIs, and custom scripts).

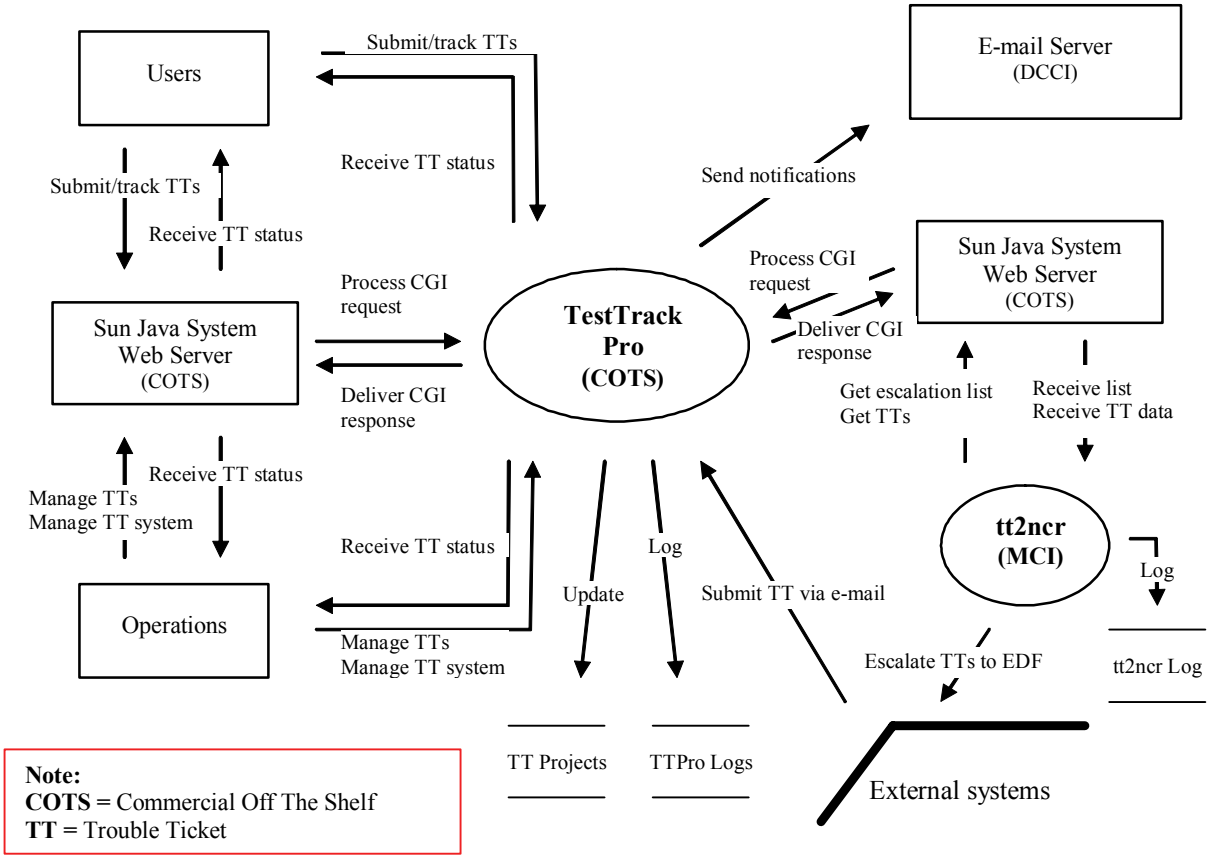


Figure 4.9-9. Trouble Ticketing Architecture Diagram

4.9.1.1.4.4 Trouble Ticketing Process Descriptions

Table 4.9-15 provides descriptions of the processes shown in the Trouble Ticketing architecture diagram.

Table 4.9-15. Trouble Ticketing Processes

Process	Type	Hardware CI	COTS / Developed	Functionality
TestTrack Pro	Program	n/a (PC)	COTS	Provides the capability to compose, submit, update, and report about ECS trouble tickets. It consists of: • the TTPro server that manages TT data; services user, operator, and application program requests; and issues system notifications; • the Seapine License server that controls access to TTPro; • Windows-, Linux-, Solaris-, and Web-based clients.that enable users or operators to: o report ECS issues by submitting TTs o update TTs as progress is made towards resolving reported issues o browse TTs and generate reports to monitor and track status o maintain system health; assign user and operator privileges; tailor workflows, data definitions, and client GUIs; and create and control system notifications. • the TTPro SOAP SDK, an API for writing a client to communicate with the TTPro server via TTPro's SOAP CGI • the SOAP CGI that accepts requests from SOAP clients and authors appropriate responses
tt2ncr	Program	n/a (PC)	Developed	Forwards escalated trouble tickets to the EDF in non-conformance report (NCR) format.

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

Trouble Ticketing Process Interface Descriptions

Table 4.9-16 provides descriptions of the interface events shown in the Trouble Ticketing architecture diagram.

Table 4.9-16. Trouble Ticketing Process Interface Events (1 of 3)

Event	Event Frequency	Interface	Initiated By	Event Description
Submit/track TTs	One per problem report.	*Program:* TestTrack Pro	Users	Users (**Users**) use TTPro's GUIs to log trouble tickets and monitor their status.
Manage TTs	One per new TT submit via browser	*Process:* TestTrack Pro (COTS)	Operations	Operations staff (**Operations**) use TestTrack Pro client software to document analyses, assignments, proposed solutions, and resolutions as they advance TTs through their lifecycle states.
Manage TT system	One per trouble ticketing system maintenance event	*Process:* TestTrack Pro (COTS)	Operations	TT system administrators (**Operations**) modify system notifications, fields, access privileges, and screen layouts. Modifying fields is not encouraged as it can produce TTs that are incompatible with other elements of the system such as TT escalation mechanisms.
Receive TT status	One per submit/track TT request	*Users*	*Process:* TestTrack Pro	TTPro provides lists, details, or reports about one or more TTs in response to ad hoc requests from users (**Users**) and operations staff (**Operators**).
Process CGI request	One per web-based submit/track TT, maintain TT, and maintain TT system event request One per get escalation list and get TT data request	*Process:* TestTrack Pro (COTS)	*Process:* Sun Java System Web Server (COTS)	TTPro's Web client, a CGI program, interacts with the TTPro server to respond to Web-based trouble ticketing service requests from users (**Users**), operations staff (**Operations)**, and application programs.

Table 4.9-16. Trouble Ticketing Process Interface Events (2 of 3)

Event	Event Frequency	Interface	Initiated By	Event Description
Deliver CGI response	One per web-based submit/track TT, maintain TT, and maintain TT system event request One per get escalation list and get TT data request	*Process:* Sun Java System Web Server (COTS)	*Process:* TestTrack Pro (COTS)	The Web Server accepts from TTPro the HTML-formatted response to a CGI request and ships it to the requestor.
Send notifications	One per qualifying TT event	*Process:* *Mail Server (COTS)*	*Process:* TestTrack Pro (COTS)	TTPro sends configurable, system notifications to the system mail server for delivery to designated users whenever a TT is created, changed, assigned, closed or escalated,to the EDF.
Get escalation list	One per TT project per tt2ncr program execution	*Process:* Sun Java System Web Server (COTS)	*Process:* tt2ncr	The ECS program "**tt2ncr**" uses TTPro's SOAP API to request, via the web server, a list of TTs that have been designated by operators for escalation to the EDF. These are TTs in the "Forwarded" state without their Forwarded flags set.
Receive list	One per Get escalation list request	*Process:* tt2ncr (COTS)	*Process:* Sun Java System Web Server (COTS)	**tt2ncr** accepts from the web server the list of TTs supplied by TTPro in response to the get escalation list request.
Get TTs	One per item in escalation list	*Process:* Sun Java System Web Server (COTS)	*Process:* tt2ncr	The ECS program "**tt2ncr**" uses TTPro's SOAP API to request TT records to be forwarded to the EDF.
Receive TT data	One per Get TTs request	*Process:* tt2ncr (COTS)	Sybase ASE (COTS)	**tt2ncr** accepts from the web server the data for one or more TTs supplied by TTPro in response to the get TTs request

Table 4.9-16. Trouble Ticketing Process Interface Events (3 of 3)

Event	Event Frequency	Interface	Initiated By	Event Description
Escalate TT to EDF	One per escalation request	*External systems*	*Process:* TestTrack Pro (COTS)	The ECS program "**tt2ncr**" converts a TT to NCR format and submits it in NCR format to the EDF for resolution by developers and engineers.
Submit TTs via e-mail	One per problem being reported by e-mail	*Process:* TestTrack Pro (COTS)	External systems	Applications such as legacy trouble ticketing systems can send an e-mail message to the TTPro server's mailbox. The server monitors the mailbox and creates a TT using the message's subject, body, and sender as the TT summary, description, and submitter, respectively. The server can be configured to accept attachments as well.
Update	One per submit, modify, and delete TT request	*Data Store:* TT Projects	*Process:* TestTrack Pro (COTS)	
Log	One per reportable event	*DataStore:* TTPro Logs tt2ncr Log	*Process:* TestTrack Pro (COTS) tt2ncr	An entry is added when errors, security-related events, and unusual activities are detected.

4.9.1.1.4.5 Trouble Ticketing Data Stores

Table 4.9-17 provides descriptions of the data stores shown in the Trouble Ticket architecture diagram. Also, descriptions are provided for the configuration files used by the Trouble Ticketing CSC.

Table 4.9-17. Trouble Ticketing Data Stores

Data Store	Type	Functionality
TT Databases	Database	Trouble ticket information is stored in a separate "project" (database) for each ECS site (e.g., LPDAAC_TTs). A testbed project also exists for each site (e.g., Testbed_LPDAAC). Projects also contain system notification rules, workflow definitions, field and record definitions, ECS projects use TTPro's native database format, Information needed by the TTPro server to communicate with clients, the mail system, and the license server is stored in a TTPro server database. Access control information is stored in the license server database.
Logs	Text	TTPro writes errors, warnings, and notices to log files. A separate log file exists for each project, the TTPro server, the Seapine License Server and the tt2ncr program. TTPro and tt2ncr logs are human readable. The TTPro logs can also be viewed using TTPro's client programs.
TTPro configuration files (not shown)	Text	Various files contains TTPro end-user, server configuration, and server connection preferences. (See vendor manuals for details.)
tt2ncr configuration file (not shown)	Text	Defines configurable parameters needed to poll projects for TTs pending escalation and to retrieve those records from the database,

4.9.1.1.5 MCI - Networker Backup/Restore Computer Software Component Description

4.9.1.1.5.1 Networker Backup/Restore Functional Overview

The Legato vendor's Networker package provides a suite of integrated tools for backup and recovery, archival and retrieval, and hierarchical storage management. The product supports multi-platform networks, contains a motif-based GUI with on-line help, and supports concurrent device support for parallel backup and recovery using up to 16 storage devices. Authorized users can perform scheduled and ad-hoc backups, recoveries, and other data management services. Networker software consists of two parts: a client portion, which runs on the systems to be backed up, and a server portion, which is the system to which the backup devices are connected. The client portion sends the data to be backed up to the server portion, which writes the data out to disk.

4.9.1.1.5.2 Networker Backup/Restore Context

A context diagram is not applicable to the Network Backup/Restore CSC.

4.9.1.1.5.3 Networker Backup/Restore Architecture

Figure 4.9-10 is the Networker Backup/Restore architecture diagram. The diagram shows the events sent to the Network Server of the Networker Backup/Restore CSC and the events the Network Server of the Networker Backup/Restore CSC sends to other processes (network clients).

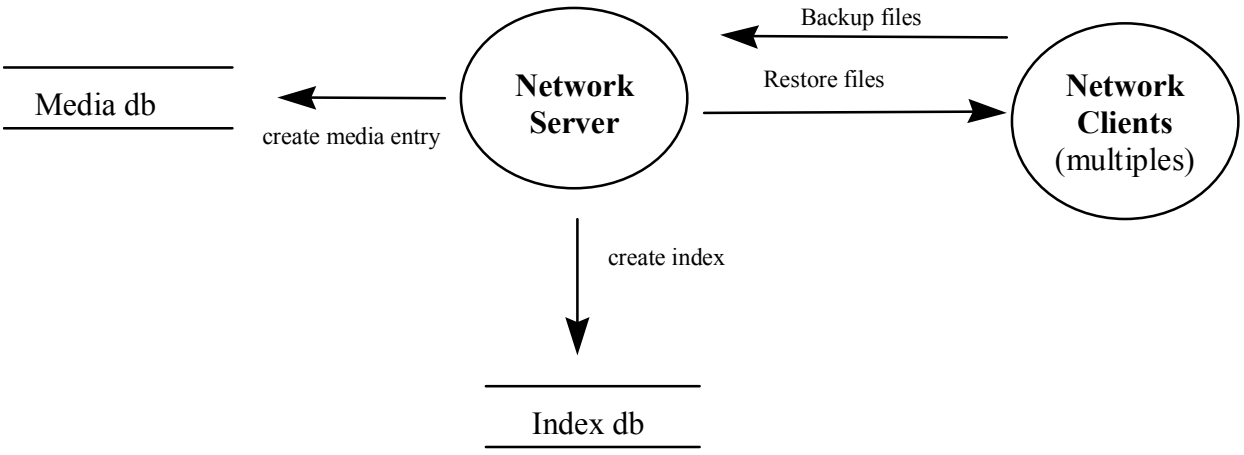


Figure 4.9-10. Networker Backup/Restore Architecture Diagram

4.9.1.1.5.4 Networker Backup/Restore Process Descriptions

Table 4.9-18 provides descriptions of the processes shown in the Network Backup/Restore architecture diagram.

Table 4.9-18. Networker Backup/Restore Processes (1 of 2)

Process	Type	Hardware CI	COTS / Developed	Functionality
Network Server	Server	MSSHW	COTS	The server can support multiple requester backups simultaneously. An index file is created to enable the backup operator to quickly find the proper tape from which to restore files or file systems.

Table 4.9-18. Networker Backup/Restore Processes (2 of 2)

Process	Type	Hardware CI	COTS / Developed	Functionality
Network Clients	Client	INTHW, AITHW, CSSHW, SPRHW, DMGHW, DRPHW, ACMHW, MSSHW	COTS	On each host that is backed up by Network, a client portion is installed. The client portion can compress data before sending it to the server; however, doing so increases CPU usage on the client machine.

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.9.1.1.5.5 Networker Backup/Restore Process Interface Descriptions

Table 4.9-19 provides descriptions of the interface events shown in the Networker Backup/Restore architecture diagram.

Table 4.9-19. Networker Backup/Restore Process Interface Events

Event	Event Frequency	Interface	Initiated By	Event Description
Backup Files	One per backup	*Process:* Network Server (COTS)	*Process:* Network Clients	Data is passed from the client to the server and archived to tape.
Restore Files	One per restore	*Process:* Network Clients	*Process:* Network Server (COTS)	Data is passed to the client to restore lost data from the tape backups.
Create Index	One per create index	COTS DB (Media db)	*Process:* Network Server (COTS)	While saving data to tape, an index is created that gives the tape identification for any version of a file that needs to be restored.
Create media entry	One per create media entry	COTS DB (Index db)	*Process:* Network Server	After saving data files (save sets) to tape, the Networker Server makes an entry in the media db identifying what save sets are on the tape.

4.9.1.1.5.6 Networker Backup/Restore Data Stores

Table 4.9-20 provides descriptions of the data stores shown in the Networker Backup/Restore architecture diagram.

Table 4.9-20. Networker Backup/Restore Data Stores

Data Store	Type	Functionality
Index db	Other	This proprietary index enables the backup operator to determine the location of the file(s) needing to be restored without searching all the tapes in the stacker. This index includes version number information where appropriate.
Media db	Other	This media db tracks what file systems (save sets) are on each tape.

4.9.1.1.6 MCI – ASTER E-mail Header Handler Computer Software Component Description

4.9.1.2 Management Software Context

Figure 4.9-11 is the Management Software CSCI (MCI) context diagram. The diagram shows the events sent to the MCI and the events the MCI sends to the other CSCIs. Table 4.9-21 provides descriptions of the interface events shown in the MCI context diagram.

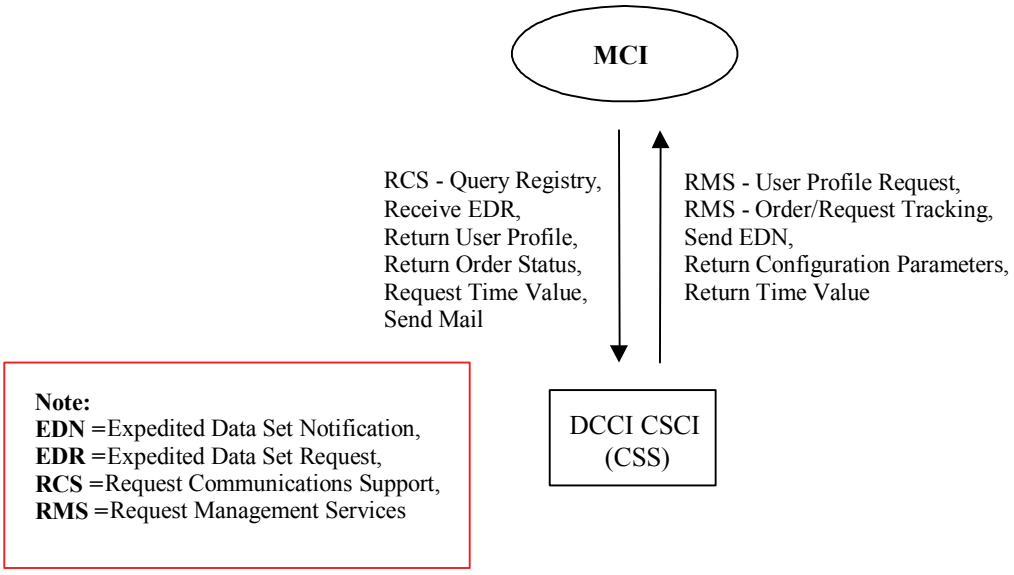


Figure 4.9-11. Management Software CSCI (MCI) Context Diagram

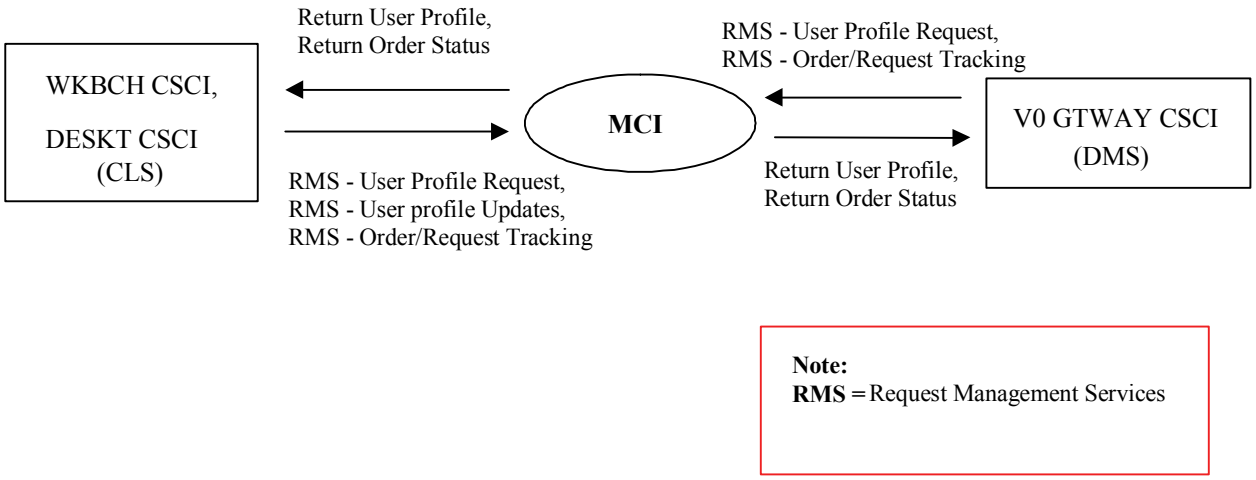


Figure 4.9-11. Management Software CSCI (MCI) Context Diagram (cont.)

Table 4.9-21. Management Software CSCI (MCI) Interface Events (1 of 2)

Event	Interface Event Description
Request Management Services	The MCI provides a basic management library of services to the CSCIs, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes: The MCI interfaces with other subsystems to perform the following: • **User Profile Request** – The MCI provides requesting CSCIs with User Profile information such as e-mail address and shipping address to support their processing activities.
Send EDN	The **DCCI CSCI** stores the EDN messages with Urs, time range, etc., and sends the EDN to the MCI.
Return Configuration Parameters	The **DCCI CSCI** returns the requested configuration parameters to the MCI.
Return Time Value	The **DCCI CSCI** returns a time value to the MCI.

Table 4.9-21. Management Software CSCI (MCI) Interface Events (2 of 2)

Event	Interface Event Description
Request Communications Support (RCS)	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS CSCI/CSC. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • File Transfer Services • Network & Distributed File Services • Bulk Data Transfer Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry – Retrieving the requested configuration attribute-value pairs from the Configuration Registry • Request Distribution Media Options from the Configuration Registry
Receive EDR	The MCI strips the EDR message header and the message is sent to the **DCCI CSCI**.
Return User Profile	The MCI returns user profile information to the **DCCI** CSCI.
Return Order Status	The MCI returns the status of an order (upon request) to the **DCCI** CSCI.
Request Time Value	The MCI submits time requests to the **DCCI CSCI**.
Send Mail	The MCI uses the CsEmMailRelA interface to send mail to the **DCCI CSCI**.
Request Management Services	The MCI provides a basic management library of services to the CSCIs, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes: The MCI interfaces with other subsystems to perform the following: • **Order/Request Tracking** – The MTMGW CSC (within the **DCCI CSCI**) can also create a user product order and retrieve it from the MCI. The **V0 GTWAY CSCI** interfaces with the MCI Order/Request Tracking service to create a user product order record. • **User Profile Request** – The MCI provides requesting CSCIs (**V0 GTWAY, DESKT** and **WKBCH**) with User Profile information such as e-mail address and shipping address to support their processing activities. • **User Profile Updates** – The **MCI** receives user profile parameter updates from a user (via the **DESKT CSCI**) and makes the updates in the user profile database.
Return User Profile	The MCI returns user profile information to the **V0 GTWAY** and **DESKT** CSCIs.
Return Order Status	The MCI returns the status of an order (upon request) to the **V0 GTWAY** CSCI.

4.9.1.3 Management Software Architecture

Figure 4.9-12 is the Management Software CSCI (MCI) architecture diagram. The architecture diagram shows the events sent to the MCI processes and the events sent by the MCI processes to other processes or COTS software.

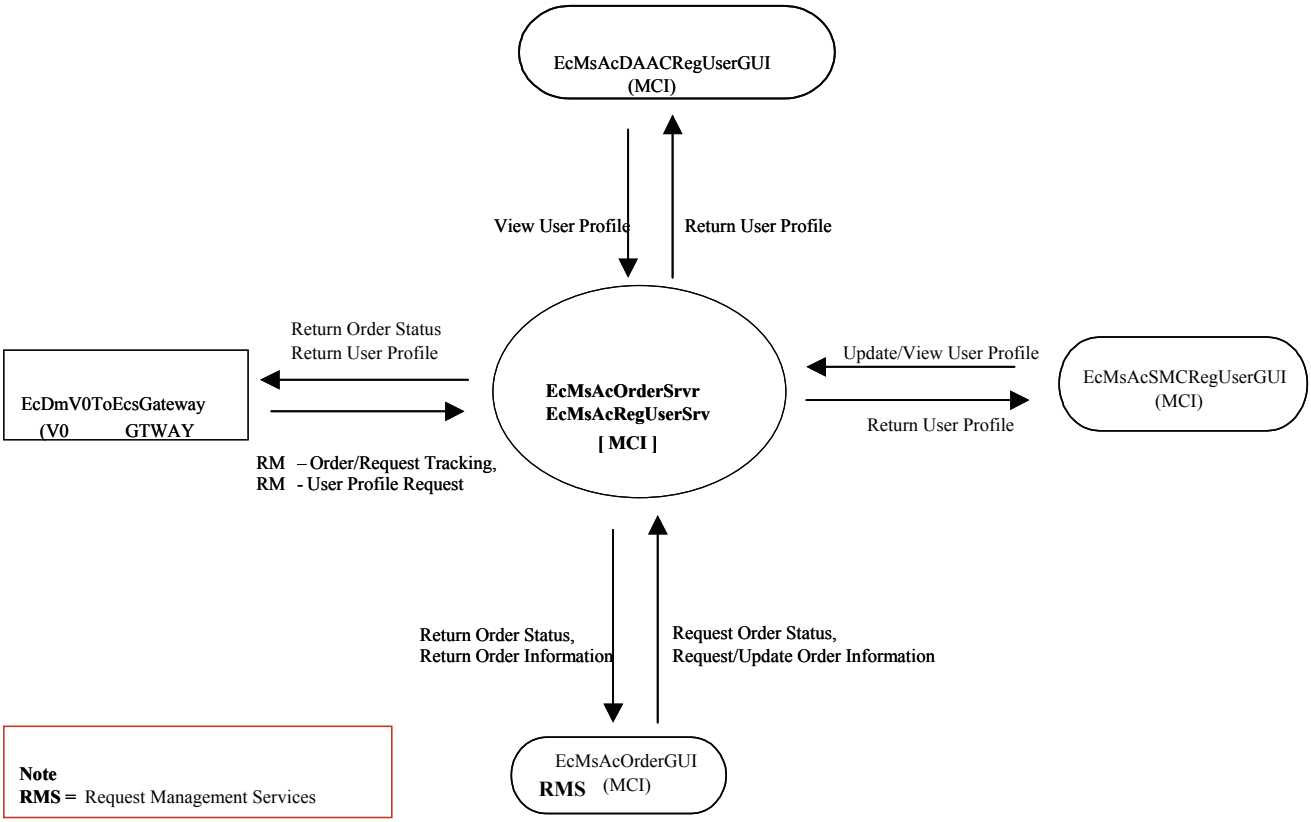
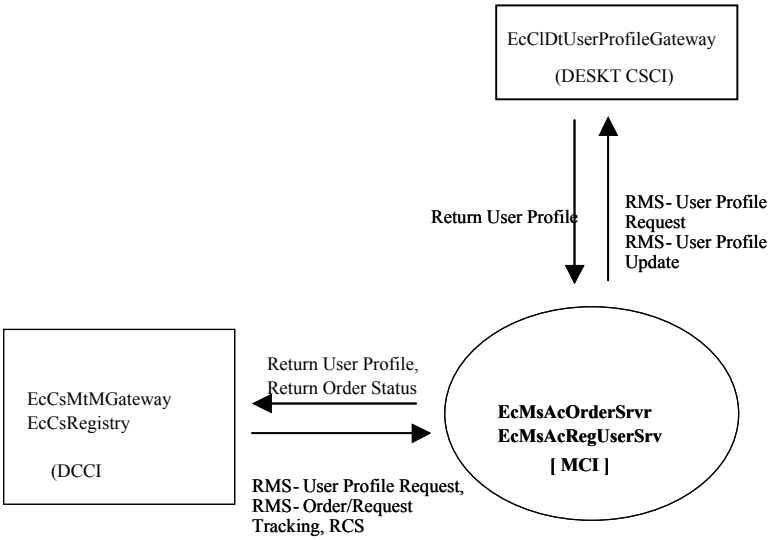
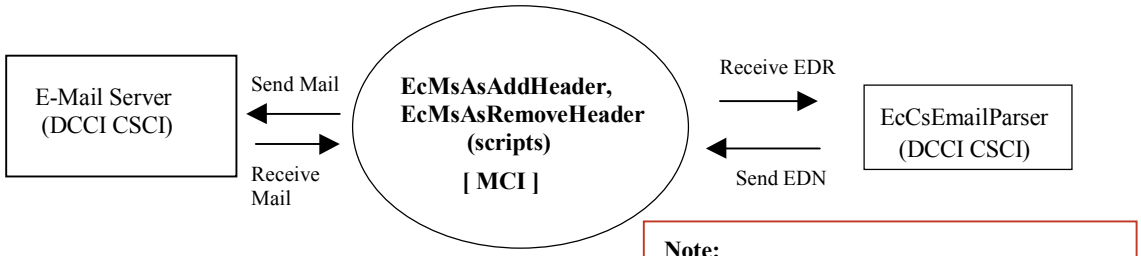


Figure 4.9-12. Management Software CSCI (MCI) Architecture Diagram



Note:
RMS Request Management Services,
RCS Request Communications Support

Figure 4.9-12. Management Software CSCI (MCI) Architecture Diagram (cont.)



Note:
EDN = Expedited Data Set Notification,
EDR = Expedited Data Set Request

Figure 4.9-12. Management Software CSCI (MCI) Architecture Diagram (cont.)

4.9.1.4 Management Software Process Descriptions

Table 4.9-22 provides descriptions of the Management Software CSCI (MCI) processes shown in the MCI architecture diagram.

Table 4.9-22. Management Software CSCI (MCI) Processes

Process	Type	Hardware CI	COTS/ Developed	Functionality
EcMsAcRegUserSrvr	Server	OMSHW	Developed	The User Registration Server provides an internal interface to the User Registration GUI and an external interface to other CSCIs/CSCs. The functions are: 1. Insert, delete, update, retrieve user registration request 2. Insert, delete, update, retrieve registered user profile 3. Retrieve a list of user registration requests 4. Retrieve a list of registered user profiles 5. Change V0 gateway password The EcMsAcRegUserSrvr supports: • Single requests at a time • Multiple concurrent requests • Asynchronous request processing • Multiple threads within a single request
EcMsAcOrderSrvr	Server	OMSHW	Developed	The Order Tracking Server provides an external interface to other CSCIs/CSCs. The functions are: 1. Insert, delete, update, retrieve order 2. Insert, delete, update, retrieve request 3. Retrieve a list of orders 4. Retrieve a list of requests 5. Update order status 6. Update request status The EcMsAcOrderSrvr supports: • Single requests at a time • Multiple concurrent requests • Asynchronous request processing • Multiple threads within a single request

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.9.1.5 Management Software Process Interface Descriptions

Table 4.9-23 provides descriptions of the interface events shown in the Management Software CSCI (MCI) architecture diagram.

Table 4.9-23. Management Software CSCI (MCI) Process Interface Events (1 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
Request Management Services (RMS)	One per service request	N/A	N/A	The EcMsAcOrderSrvr and EcMsAcRegUserSrvr provide a basic management library of services to the processes, implemented as client or server applications, using the DCCI CSCI Process Framework. The basic management library of services includes the items below.
RMS – (cont.)	At system startup or shutdown and for restarts	*Processes:* EcMsAcOrderSrvr, EcMsAcRegUserSrvr	DAAC unique startup scripts	**System startup and shutdown –** Please refer to the release-related, current version of the Mission Operations Procedures for the EMD Project document (611) and the current EMD Project Training Material document (625).
(RMS – cont.)	One per request	*Process:* EcMsAcOrderSrvr *Library:* MsAcCInt *Class:* EcAcOrderCMgr	*Process:* EcDmV0ToEcsGateway *Class:* DmGwRequestReceiver CGI Interface, CGI Scripts	The EcMsAcOrderSrvr interfaces with other processes to perform the following: **Order/Request Tracking** - The **EcDmV0ToEcsGateway** interfaces with the EcMsAcOrderSrvr (Order/Request Tracking service) to create a user product order record. In addition, the **MTMGW CSC** can also create a user product order and retrieve it from the EcMsAcOrderSrvr.

Table 4.9-23. Management Software CSCI (MCI) Process Interface Events (2 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
(RMS – cont.)	One profile per request	*Process:* EcMsAcRegUserSrvr, *Libraries:* MsAcClnt, MsAcComm *Class:* EcAcProfileMgr *Process:* EcMsAcOrderSrvr *Library:* MsAcClnt *Class:* EcAcOrderCMgr *Process:* EcCsMtMGateway *Libraries:* MsAcClnt, MsAcComm	*DMS Process:* EcDmV0ToEcsGateway *Class:* DmGwRequestReceiver *CLS Process:* EcClDtUserProfileGateway *Class:* ClDtProfileServer *CSS Process:* EcCsMtMGateway *Class:* EcCsMtMAcctSrvMgr	**User Profile Request** – The EcMsAcRegUserSrvr provides requesting processes with user profile information such as e-mail address and shipping address to support their processing activities.
View User Profile	One per view user profile request	*Process:* EcMsAcRegUserSrvr *Libraries:* MsAcClnt, MsAcComm	M&O staff *Process:* EcMsAcDAACRegUserGUI *Libraries:* MsAcClnt, MsAcComm	The M&O staff request, via the **EcMsAcDAACRegUserGUI**, to retrieve a user profile from the EcMsAcRegUserSrvr.

Table 4.9-23. Management Software CSCI (MCI) Process Interface Events (3 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
Return User Profile	One per return of user profile	*Processes:* EcMsAcSMCRegUserGUI, EcMsAcDAACRegUserGUI, EcDmV0ToEcsGateway, EcClDtUserProfileGateway, EcCsMtMGateway, *Libraries:* MsAcClnt, MsAcComm	*Process:* EcMsAcRegUserSrvr *Libraries:* MsAcClnt, MsAcComm	The EcMsAcRegUserSrvr returns user profile information provided from the Sybase ASE to the requester at the **EcMsAcSMCRegUserGUI** or the **EcMsAcDAACRegUserGUI** and to the **EcDmV0ToEcsGateway**, **EcClDtUserProfileGateway,** and the **EcCsMtMGateway**.
Update/View User Profile	One per create/ update user profile	*Process:* EcMsAcRegUserSrvr *Libraries:* MsAcClnt, MsAcComm	M&O staff *Process:* EcMsAcSMCRegUserGUI *Libraries:* MsAcClnt, MsAcComm *Class:* MsAcRegUsrUtl	The M&O staff sends requests to modify or review user profile information via the **EcMsAcSMCRegUserGUI**.
Request Order Status	One per request order status	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	M&O Staff *Process:* EcMsAcOrderGUI *Libraries:* MsAcClnt, MsAcComm	The EcMsAcOrderGUI is used (by the M&O Staff) to retrieve the current order status. The **EcMsAcOrderGUI** submits a request for current order status to the EcMsAcOrderSrvr.
Request/Update Order Information	One per request/update order information	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	M&O Staff *Process:* EcMsAcOrderGUI *Libraries:* MsAcClnt, MsAcComm	The **EcMsAcOrderGUI** is used (by the M&O Staff) to retrieve order information by submitting requests to the EcMsAcOrderSrvr.
Return Order Status	One per return order status	*Processes:* EcMsAcOrderGUI, EcDmV0ToEcsGateway, EcCsMtMGateway *Libraries:* MsAcClnt, MsAcComm	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	The EcMsAcOrderSrvr receives the product order status from the Sybase ASE and returns the status to the **EcMsAcOrderGUI**, the **EcDmV0ToEcsGateway**, and the **EcCsMtMGateway**.

Table 4.9-23. Management Software CSCI (MCI) Process Interface Events (4 of 4)

Event	Event Frequency	Interface	Initiated By	Event Description
Return Order Information	One per return of order information	*Process:* EcMsAcOrderGUI *Libraries:* MsAcClnt, MsAcComm	*Process:* EcMsAcOrderSrvr *Libraries:* MsAcClnt, MsAcComm	The EcMsAcOrderSrvr receives the product order information from the Sybase ASE and returns the order information to the **EcMsAcOrderGUI**.
Request Communications Support	Request service(s) as required	*Process:* EcCsIdNameServer *Libraries:* EcPf, Middleware, FoNs, FoIp, oodce *Classes:* EcPfManagedServer, CCSMdwNameServer, FoNsNameServerProxy, CCSMdwRwNetProxy *Library (Common):* EcUr *Class:* EcUrServerUR *Library:* event *Class:* EcLgErrorMsg *Process:* EcCsRegistry *Library:* EcCsRegistry *Class:* EcRgRegistryServer_C	*Processes:* EcMsAcOrderSrvr, EcMsAcRegUserSrvr *Libraries:* MsAcClnt, MsAcComm	The **DCCI CSCI** provides a library of services available to each SDPS and CSMS CSCI/CSC. The CSCI services required to perform specific assignments are requested from the DCCI CSCI. These services include: • CCS Middleware Support • Database Connection Services • Name/Address Services • Password Services • Server Request Framework (SRF) • Universal Reference (UR) • Error/Event Logging • Fault Handling Services • Mode Information • Query Registry – Retrieving the requested configuration attribute-value pairs from the Configuration Registry

4.9.1.6 Management Software Data Stores

Data stores are not applicable for the Management Software CSCI.

4.9.2 Management Logistics Computer Software Configuration Item Description

4.9.2.1 Management Logistics Functional Overview

The Management Logistics CSCI (MLCI) supports the configuration management of the EMD. The MLCI is the following three CSCs:

- Inventory, Logistics, Maintenance (ILM) Manager: The ILM CSC maintains property records about contract purchased items, storing information such as vendor, date of receipt, installation, and warranty expiration. The ILM CSC also maintains maintenance records about contract purchased items as well as license management records about COTS software.
- Software Change Manager: The Software Change Manager CSC supports maintenance and change control of the science software configuration at each DAAC.
- Software License Manager: The Software License Manager CSC monitors and controls licensing of COTS products installed in the EMD.

4.9.2.1.1 MLCI - Inventory/Logistics/Maintenance Manager Computer Software Component Description

4.9.2.1.1.1 Inventory/Logistics/Maintenance Manager Functional Overview

The Inventory/Logistics/Maintenance (ILM) Manager is the principal tool used for EMD integrated logistics support (ILS). It tracks and maintains the key data pertaining to EMD contract purchased items including hardware, COTS software and associated licenses, consumable items, and Government Furnished Equipment (GFE). Information tracked for each item includes dates (e.g., receipt, installation, and warranty expiration), location, manufacturer, vendor, purchase order, Original Equipment Manufacturer (OEM) part number, model/version, and description. The ILM Manager also stores and maintains detailed maintenance data about hardware -- including corrective maintenance.

The ILM Manager gives authorized users access to property, license and maintenance data via a graphical user interface (GUI). The user can query for and display an individual record or a list of a set of records that meets specified search criteria. Reporting capabilities are available to the user by accessing the report menu and entering valid values. The user can also obtain ad-hoc reports of ILM data and transmit the reports to the screen, a file, or a printer.

- The functionality for the Inventory/Logistics/Maintenance Manager is implemented via the Remedy Action Request System and ILM related custom scripts.

4.9.2.1.1.2 Inventory/Logistics/Maintenance Manager Context

The ILM Manager does not have an interface with any other subsystem, CSCIs or CSCs.

4.9.2.1.1.3 Inventory/Logistics/Maintenance Manager Architecture

The ILM Manager is implemented as a Remedy ARS application. Remedy ARS is a client/server COTS application that facilitates the ILM function through a set of inventory, logistics, license and maintenance transactions supporting forms and processing capabilities. Remedy ILM's data is stored in a Sybase relational database. Figure 4.9-13 is the ILM Manager architecture diagram.

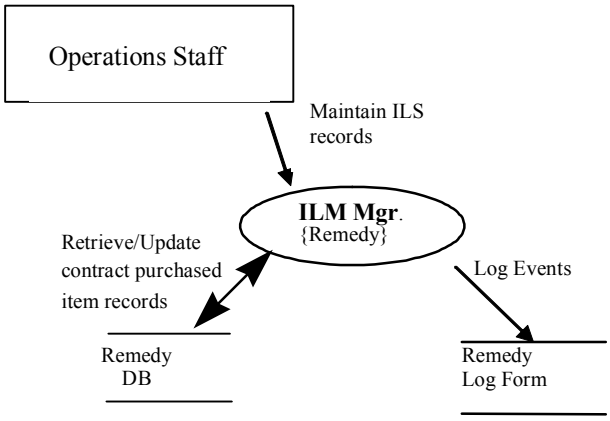


Figure 4.9-13. Inventory/Logistics/Maintenance (ILM) Manager Architecture Diagram

4.9.2.1.1.4 Inventory/Logistics/Maintenance Manager Process Descriptions

The Remedy ILM application is initiated by the Remedy PC User tool. Once logged into the user tool, the user has access to a variety of forms/capabilities that he or she is authorized to use. These forms/capabilities enable the user to perform the functions summarized in Table 4.9-24.

Table 4.9-24. Inventory/Logistics/Maintenance (ILM) Manager Processes

Process	Type	Hardware CI	COTS / Developed	Functionality
ILM Mgr. (Remedy ARS)	Other	INTHW MSSHW	COTS	The ILM performs the following tasks: • Captures and maintains all pertinent data for project hardware and COTS software licenses • Manages, distributes and reports on consumables • Maintains historical log of maintenance actions against individual items within ILM • Tracks movement and archive actions for project property and reports on same • Maintains all other pertinent property management data required for the efficient use of the ILM tool such as vendor, manufacturer, and internal usage codes

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.9.2.1.1.5 Inventory/Logistics/Maintenance Manager Data Stores

ILM Manager's principal data stores are the Remedy ILM database. The data store descriptions are provided in Table 4.9-25.

Table 4.9-25. Inventory/Logistics/Maintenance (ILM) Manager Data Stores

Data Store	Type	Functionality
Remedy ILM DB	Database	A non-replicated collection of inventory and maintenance-related data that exists at the EDF. For ILM, it contains records identifying and describing: • Inventory items • Corrective maintenance • Location data • License management data • Maintenance support information
Remedy ILM Transaction Log Form	Form	• A collection of transaction records that describe the receipt, installation, relocation, and archiving of inventory items.

4.9.2.1.2 MLCI - Software Change Manager Computer Software Component Description

4.9.2.1.2.1 Software Change Manager Functional Overview

The Software Change Manager aids the DAACs, EOC, and SMC staffs in organizing and partitioning software, controlling software changes and versions, and in assembling sets of software for release purposes. The Software Change Manager consists of a COTS application called ClearCase.

4.9.2.1.2.2 Software Change Manager Context Diagram

The Software Change Manager does not interact with any CSCIs or CSCs.

4.9.2.1.2.3 Software Change Manager Architecture

The Software Change Manager (ClearCase) does not interface with any external processes.

4.9.2.1.2.4 Software Change Manager Process Descriptions

The Software Change Manager's primary process is the COTS package, ClearCase. ClearCase has both a command line and a graphical user interface to execute its programs. Table 4.9-26 provides a summary of its functions.

Table 4.9-26. Software Change Manager Processes

Process	Type	Hardware CI	COTS / Developed	Functionality
SW Change Mgr. (ClearCase)	Program	n/a	COTS with custom developed scripts	• Organizes and stores software in a software library • Manages multiple versions of software files • Regulates access to software file versions • Controls and logs changes to software file versions • Manages software file version's progress through the development cycle • Performs builds of software according to user defined version specifications • Maintains records of a build's content (files, compiler, and other resources used)

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.9.2.1.2.5 Software Change Manager Process Interface Descriptions

Software Change Manager Process Interface Descriptions are not applicable to the MLCI.

4.9.2.1.2.6 Software Change Manager Data Stores

The Software Change Manager's COTS package, ClearCase's data stores consist of a database and log files. Table 4.9-27 provides descriptions of the data stores shown in the Software Change Manager architecture diagram.

Table 4.9-27. Software Change Manager Data Stores

Data Store	Type	Description
ClearCase Database	Database	ClearCase uses a proprietary database management/database scheme that consists of versioned object base(s) (VOB) and views. A VOB is a data structure mounted as a multi-version file system and is created through use of the ClearCase "make vob" command. A VOB contains versions of directories and files, user-defined metadata, build records, event records, and configuration records. A view is also a data structure that's used as short-term storage for data created during the development process. View stores checked-out versions of file elements, a user's private files, and newly built derived objects.
ClearCase Log Files	File	ClearCase log files record error and status information from various ClearCase server programs and user programs. These log files are ASCII files and are described in the ClearCase Reference Manual.

4.9.2.1.3 MLCI – Software License Manager Computer Software Component Description

4.9.2.1.3.1 Software License Manager Functional Overview

The Software License Manager manages network license activities associated with using COTS products. The Software License Manager maintains information about license provisions, meters use of installed licenses, and reports on licensing events and statistics for vendor software having embedded FLEXlm licensing technology.

Software License Manager functionality is implemented by the COTS product FLEXlm and its associated data files.

The Software License Manager contains no custom scripts or files.

4.9.2.1.3.2 Software License Manager Context

Software License Manager runs at every EMD site, providing local network licensing services to requesting COTS applications and EMD operators. The Software License Manager does not interact with any other CSCIs or CSCs.

4.9.2.1.3.3 Software License Manager Architecture

Figure 4.9-14 is the Software License Manager architecture diagram and consists of FLEXlm and its related data files. It has a client/server architecture with license servers responding to requests from client processes embedded in managed COTS applications or license manager utilities and multiple license servers can run concurrently. It provides a command line interface for the Operations Staff.

FLEXlm consists primarily of license manager daemons, vendor daemons, license files, and client applications code embedded in licensed application. Each FLEXlm server must have its own license file, and each server logs errors and licensing events to its own "debug file". Options files are used to specify operating parameters for handling individual vendors' products. Redundant FLEXlm servers can be configured to insulate against server failure; however, this requires three license server hosts. A redundant server implemenation is currently used at all sites for Linux hosts.

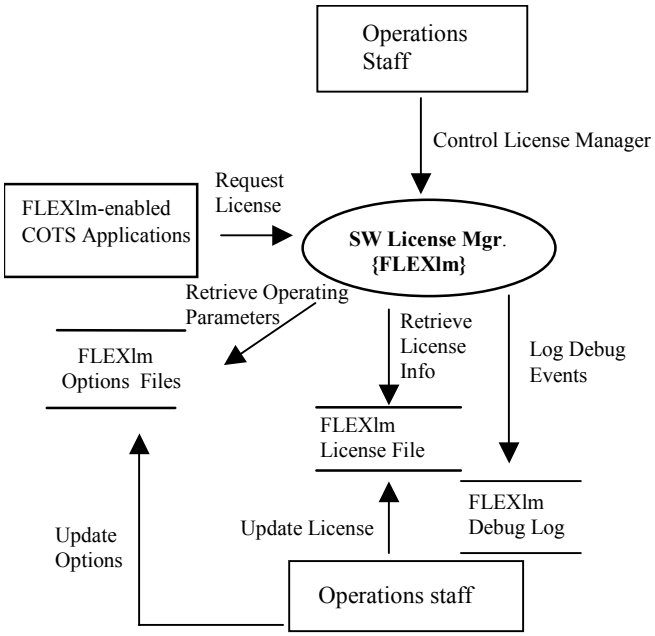


Figure 4.9-14. Software License Manager Architecture Diagram

4.9.2.1.3.4 Software License Manager Process Descriptions

License management services are performed at local sites by a set of COTS daemons and utilities identified collectively in Table 4.9-28 as the Software License Mgr. The principal persistent process is the server daemon, lmgrd. It controls interactions between client processes and vendor daemons. The latter grants or denies an application's request to use a license. Table 4.9-33 provides a description of the process involved in software licenses management for a local site.

Table 4.9-28. Software License Manager Processes

Process	Type	Hardware CI	COTS / Developed	Functionality
Software License Mgr. (FLEXlm)	Program	CSSHW, MSSHW, DPSHW, ACMHW, SPRHW, DRPHW, DIPHW	COTS	The FLEXlm server daemon (lmgrd) with its associated command line utilities: • Shuts down and restarts license daemons on a license server node and makes license data available to the servers • Manages license checkout and checkin processing for FLEXlm-enabled COTS products • Logs licensing events and errors to files on the local network • Removes a user's license for a specified feature • Displays the status of installed licenses and of network licensing activities; this includes listing licensed software features and their associated product versions, vendors, hosts, and expiration dates • Reports the hostid of a system (needed to obtain license key from vendors)
FLEXlm-enabled COTS Application	Program	CSSHW, MSSHW, DPSHW, ACMHW, SPRHW, DRPHW, DIPHW	COTS	Client software within vendor products communicates with FLEXlm's license server and vendor daemons to request licenses for product users to run.

EBIS Document 920-TDx-001 (Hardware Design Diagram) provides descriptions of the HWCI, and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

4.9.2.1.3.5 Software License Manager Process Interface Descriptions

Most of the Software License Manager's process interfaces exist between the license servers and their client applications embedded within COTS applications to handle license requests. They are not described here in detail, as they are internal to the COTS software.

Table 4.9-29 provides descriptions of the interface events shown in the Software License Manager architecture diagram.

Table 4.9-29. Software License Manager Process Interface Events

Event	Event Frequency	Interface	Initiated By	Event Description
Control License Manager	One per event	Process: SW License Mgr. (COTS) (FLEXlm)	Operations Staff	The **Operations Staff** start, stop and monitor license servers, as well as update license manager databases, and request reports about installed licenses and selectable license events.
Log Debug Events	One per debug event	*Data Store:* FLEXlm Debug Log	*Process:* SW License Mgr. (FLEXlm) (COTS)	The FLEXlm server logs all errors and license events (e.g., start/stop date, and status of installed license, product version, host(s) to which servers are connected) in the **FLEXlm Debug Log**.
Update License	One per update request	*Data Store:* *FLEXlm* License File	Operations Staff	The **Operations Staff** adds and removes the managed FLEXlm software licenses.
Retrieve License Info	One per request	*Data Store:* FLEXlm License File	*Process:* SW License Mgr. (FLEXlm) (COTS)	Reads license provisions for one or more FLEXlm-enabled COTS products from the **FLEXlm License File**.
Update Options	One per parameter change	*Data Store:* FLEXlm Options Files	Operations Staff *Process:* Unix file editor	Store parameters for regulating license usage and event logging in the **FLEXlm Options Files**.
Retrieve Operating Parameters	One per license request	SW License Mgr.(COTS) (FLEXlm)	*Process:* SW License Mgr. (FLEXlm) (COTS)	Retrieve parameters for regulating license usage and event logging from the **FLEXlm Options Files**.
Request License	One per license request	*Process:* SW License Mgr. (COTS) (FLEXlm)	*Programs:* Flexlm - enabled COTS applications	Communication among license servers and clients to establish connections and checkout, check-in, and monitor activity of licenses are obtained from **FLEXlm-enabled COTS applications**.

4.9.2.1.3.6 Software License Manager Data Stores

License Manager's principal data stores are the FLEXlm license, debug, and option files. FLEXlm files are described in the FLEXlm End User Manual. Table 4.9-30 provides descriptions of the data stores shown in the Software License Manager architecture diagram.

Table 4.9-30. Software License Manager Data Stores

Data Store	Type	Functionality
FLEXlm license file	Text file	A collection of records containing license provisions and passwords for one or more FLEXlm-enabled COTS products. They identify: • Servers - name, hostid, and port number of the license manager daemon • Daemons - name and path of vendor daemons that track licenses checked out and to whom. An options file can be named for each vendor daemon • Features - description of the license to use a product Each license server uses one license file, and operators combine license files received from vendors, as possible, to reduce the number of servers in the network. License file content and format is described in the FLEXlm End User Manual.
FLEXlm debug log	Text file	A collection of records describing licensing errors and events that have occurred. Records contain a timestamp, an informational message, and the name of the daemon generating the message. Each license server (except redundant servers) writes to its own log file.
FLEXlm options files	Text file	Collections of records that specify optional operating parameters for managing specific vendors' products. Options files are named in license files. There can be one options file for each vendor each license file specifies.

4.9.2.2 Management Logistics Context

The Management Logistics CSCI (MLCI) does not interact with any other CSCIs or CSCs.

4.9.2.3 Management Logistics Architecture

An architecture diagram does not apply to the MLCI because it consists of standalone tools.

4.9.2.4 Management Logistics Process Description

Process descriptions are not applicable for the MLCI.

4.9.2.5 Management Logistics Process Interface Descriptions

Process Interface Descriptions are not applicable for the MLCI.

4.9.2.6 Management Logistics Data Stores

Data Stores are not applicable for the MLCI.

4.9.3 ECS Assistant Script Library

4.9.3.1 ECS Assistant Script Library Functional Overview

ECS Assistant Script Library supports the ECS Assistant GUI and almost all-internal EMD shell scripts for system installation, configuration, monitoring and shutting down servers and many other functions. These functions are described in the Process Interface Descriptions sub-section.

4.9.3.2 ECS Assistant Script Library Context

The context diagram is not applicable to the ECS Assistant Script Library.

4.9.3.3 ECS Assistant Script Library Architecture

The architecture diagram is not applicable to the ECS Assistant Script Library.

4.9.3.4 ECS Assistant Script Library Processes

The Process Table is not applicable to the ECS Assistant Script Library.

4.9.3.5 ECS Assistant Script Library Process Interface Descriptions

The ECS Assistant Script Library provides functions available to calling scripts. Table 4.9-31 describes the interface.

Usage: /ecs/formal/COMMON/scripts/EcCoScriptlib command [args]

Table 4.9-31. ECS Assistant Script Functions (1 of 3)

Function	Functionality
add_cds_entry	Add CDS entry.
add_client	Add CCS Middleware client.
add_size	Record the package size in the .install_stats file.
archive_if_needed	Save a copy of the destination file if different from the original file.
cache_subsys_components	Cache the subsystem-component information.
cfgpatch	Patch parameter files based on the .cfgpatch file.
check_dirs	Check for the existence of specified directories.
clean_logs_dir	Remove logs in the log directory older than a specified date.
cleanup	Cleanup after servers in a subsystem and component.
components_health	Show health of component installation.
convert_shell	Convert a file to a format to be included by the specified shell.
date_to_daynum	Compute the date from the day number in the year.
daynum_to_date	Compute the day number in the year from the date.
days_ago	Compute the date num_days ago.

Table 4.9-31. ECS Assistant Script Functions (2 of 3)

Function	Functionality
del_cds_object	Delete server object from CDS name space.
Extract_desc_value	Extract a value from an ESDT descriptor file, based on its object name.
generate_servers_file	Generate servers file for a given mode, subsystem and component.
get_architecture	Display the ARCH variable that identifies the machine.
get_colorpair	Get color pairs from a predetermined palette of color pairs.
get_health	Display installation health indicators for mode, subsystem and component.
get_hostlist	Get the host list for the current site.
get_hostname	Display the HOSTNAME variable that identifies the machine.
get_included_files	Get names of included files from a package.
get_revlevel	Display the revision level of a parameter file.
get_site	Get the site we are running on.
get_site_acronym	Get the standard three-letter acronym of the site we are running on.
get_taglist	Get a list of tags from a file.
getuname	Print user name.
getview	Print Clearcase view or NONE.
install	Install to a mode.
install_type	Determine the installation type of an installation.
is_leap_year	Determine whether the year is a leap year.
kill	Kill servers in a subsystem and component.
kill_procs	Kill the named processes.
list_apps	List applications in a mode, subsystem and component.
list_apps_by_ius	List applications by which IUs are installed.
list_archpkgs	List packages for a particular architecture.
list_components	List components of a subsystem.
list_componentsdirs	List components' directories of subsystem.
list_execs	List executables of a given type in a mode, subsystem and component.
list_execs_by_app	List executables, which belong to the specified application.
list_execs_by_ius	List executables of a given type in a mode by which IUs are installed.
list_installtypes	List file types, which can be installed.
list_ius	List installed units in a mode, subsystem and component.
list_logfiles	List the log files for the servers in a subsystem and component.
list_mkcfgs	List the Mkcfg scripts, which are available for execution.
list_servers	List servers in a subsystem and component.
list_servers_from_file	List servers in a .servers file.
list_subsystems	List subsystems.
look_for_proc	Look for a process.
lookup_component	Lookup details on a component.
lookup_installtype	Look up a file type, which can be installed.

Table 4.9-31. ECS Assistant Script Functions (3 of 3)

Function	Functionality
lookup_subsystem	Lookup details on a subsystem.
mk_cds_entry	Make CDS entries for a subsystem and component.
mkcfg	Make config files for servers in a subsystem and component.
mkmodedirs	Make all the directories required for a mode.
mode_health	Display health indicators for a mode.
modify_parms	Modify a dbparms, extparms or cfgparms file.
monitor	Monitor servers in a subsystem and component - continuously.
monitor_once	Monitor listed servers - a one-time snapshot.
package_health	Show health of a package installation.
package_size	Return the size of the contents of the package in kilobytes.
record_proc	Record a process.
refresh_common_scripts	Refresh common EMD scripts.
refresh_control_files	Refresh EMD control files.
revert_parms	Recover the most recently time stamped parameters file.
set_cfgparms	Set configurable parameters in the environment.
set_dbparms	Set database parameters in the environment.
set_environment	Write the appropriate commands to set up the standard environment.
set_fileperms	Set file permissions.
set_symbolic_hosts	Set up the array of symbolic host names for this site.
Start	Start servers in a subsystem and component.
start_client	Start an EMD client program.
start_ecs	Start the whole EMD system.
start_gui	Start an EMD GUI.
start_server	Start a single EMD server with standard checks.
start_server_group	Start a group of servers with standard checks.
strip_comments	Strip comments and blank lines from a file.
subsys_health	Show health of subsystem installation.
Uninstall	Uninstall from a mode.
Viewlog	View logs.

4.9.3.6 ECS Assistant Script Library Data Stores

ECS Assistant data files have some common properties. All files support comments beginning with a "#" sign and terminating at the end of the line. Also, blank lines are permitted. All blank lines and comments are ignored in the processing of the file.

Applications File (.applications)

The .applications file is delivered from /ecs/formal/COMMON/.applications in Clearcase to $ECS_HOME/$MODE/.applications in the mode. It is always delivered whenever ECS Assistant is used to install.

The format of this file is shown in Table 4.9-32. The file maps application names to application ids.

Table 4.9-32. Applications File Format

Header	
NAME	**Application ID**
EcEcEcsApp (System – The Boss Application)	9999999

Cfgparms File (.cfgparms)

The .cfgparms files have a header, several tags denoted by a word ending in a colon ":", and one or many parameter value pairs associated with each tag. There is one .cfgparms file associated with each subsystem – component pair. For each Mkcfg script, which exists for a component configurable parameters for a particular subsystem and component. It is always delivered whenever ECS Assistant is used to install that particular subsystem and component. Table 4.9-33 (whether it is delivered or not), there is a tag and the parameter value pairs, which describe the is the .cfgparms file format.

Table 4.9-33. Cfgparms File Format

Header	
EcDsScienceDataServerAppMOscfg:	
NumOfHDFServer	3
EcDsHdfEosServerMkcfg:	
AppLogSize	1000000
AppLogLevel	0
DebugLevel	0
DBLibrary	SYBASE_CT
DBServer	OTIS_SERVER

Cfgpatch File (.cfgpatch)

The .cfgpatch file specifies all the changes between one drop of code and another, between releases. The parameter REVISION_LEVEL needs to be defined in the file so the ECS Assistant software can check whether a .cfgparms file needs to be patched or not. The syntax for having the REVISION_LEVEL parameter expanded before execution of the patching is %${REVISION_LEVEL}. The first few lines in this cfgpatch file are standardized, and they add the parameter RevisionLevel to the .cfgparms file with the value specified in the REVISION_LEVEL definition:

```
DEFINE  REVISION_LEVEL  6B.05.EPSILON.01
```

Subsequent lines in the file specify parameters and tags to add and delete in cfgparms and dbparms files. The possible *actions* are: *DEFINE, ADD, UPD, DEL, ADD_TAG, DEL_TAG*. The *DEFINE* action takes two additional arguments, *Parameter* and *Value*. The *ADD* and *UPD* actions need all 10 fields specified. The *DEL* action does not need *valuelist* specified. The *DEL_TAG* action does not need *Parameter* or *ValueList* specified. *Site* can be any of the supported sites listed in the sitemap or ALL. *Host* is any host name at the specified site or ALL. *Mode* is any supported mode or ALL. *Subsys* is any subsystem label as specified in the subsystems file. *Component* is one of the components specified in the components file for that subsystem. *Filename* can be either .cfgparms, .dbparms or .extparms. *Tag* can be any tag in the .cfgparms file, or could be a new tag to add. *Parameter* can be any string without spaces, and starting with an alphanumeric character. *ValueList* can be any string, even with spaces in it.

A sample file is shown in table 4.9-34.

Table 4.9-34. Cfgpatch file format

Define a macro									
Action	**Parameter**		**Value**						
DEFINE	REVISION_LEVEL		6B.05.EPSILON.01						
ALL files									
Action	**Site**	**Host**	**Mode**	**Subsys**	**Comp**	**Filename**	**Tag**	**Parameter**	**ValueList**
ADD	ALL	ALL	ALL	ALL	ALL	.dbparms	Rev Level	Revision Level	%${REVISION_LEVEL}
ADD	ALL	ALL	ALL	ALL	ALL	.cfgparms	Rev Level	Revision Level	%${REVISION_LEVEL}
GSFC .cfgparms changes									
Action	**Site**	**Host**	**Mode**	**Subsys**	**Comp**	**Filename**	**Tag**	**Paramet**	**ValueList**
ADD	GSFC	ALL	ALL	CLS	EcCl	.cfgparms	EcClWbJdtMkcfg	SUBSCRIPTION_ESDT_SN	AST_L1BT
ADD	GSFC	ALL	ALL	CLS	EcCl	.cfgparms	EcClWbJdtMkcfg	SUBSCRIPTION_ESDT_VID	001

Components Files (.components)

The components file specifies which logical components exist under a subsystem. For example, the DSS subsystem has the following components: EcDsSr, EcDsSt, EcDsDd, and EcDsDo. They are specified in the components file for that subsystem. Each subsystem will have a components file, even if there is only one component. For example, Ingest has only one component, EcIn. The components file specifies the component names and the directory under the main subsystem directory where the ECS Assistant files for that component can be found. Table 4.9-35 is a sample of the file format.

Table 4.9-35. Components file format

Mnemonic	Directory	CSCI
EcDsSr	SDSRV	Science Data Server

Dbparms File (.dbparms)

The dbparms file has the exact same format as the .cfgparms file, except that the tags refer to database scripts, and the parameter value pairs apply to values used in the database operations. There is one dbparms file delivered for each subsystem component pair, whether there is a database or not. If there is no database, the file will have only a header and no tags or parameter value pairs.

Installable Unit (IU) and Files

An installable unit file contains several types of entries. There is usually a header consisting of comments. The file may have one or more tags denoted by a word ending in a colon. Those tags come from the installtypes file, see below. There are also some pseudo tags. The tags preinstall, postinstall are used to invoke an action before or after the main installation. ANY UNIX COMMAND can be put there. There is also the *ssunique* tag, which allows for installation of objects not referenced by the installtypes file. Associated with the ssunique tag, you can have commands of the form *copy*, *link*, *cpln*, and *mkdir*. The *copy* command takes two arguments, *orig* specifying the file to be copied, and *dest* specifying the location to copy the file to. Macros specified in the EA macros file are used when specifying these paths, so that hard coding of paths can be eliminated. The *link* command can be used to make a symbolic link from one file or directory to another. The *cpln* command will make a symbolic link in a development installation and a copy in a non-development installation. The *mkdir* command will make the specified directories if they do not exist, and will not complain if they exist already. The usage of the four commands is as follows:

copy [-r] permissions orig dest

link orig dest

cpln permissions orig dest

mkdir dir dir …

An IU file may also include another file using the %include command. It takes one file name as an argument. It is generally only used in PKG files, and not in IU files.

Eamacros File (.eamacros)

This file lists the standard ECS Assistant macros available to be used in IU and PKG files.

Envvars File (.envvars)

This file occurs in the /ecs/formal/COMMON and in every subsystem. It specifies in an architecture specific way the environment to be used when running servers. The file is converted to a shell script file, which is then read in at run time to create the standard environment. The other entries are in the form of parameter value pairs. The resulting file consists of entries of the form

export parameter=”${value}”

These commands are executed at run time and read into the environment in a Korn shell script. This same format applies to the common and subsystem-component levels.

Executables File (.executables)

The executables file specifies all the executable servers, GUIs, cgi-bin programs, clients and utilities. The file has a header and five fields for each un-commented line as follows:

Program is the executable name

IU Name is the IU file, which delivers the program

Type, can be S for server, C for client, G for GUI, T for test program, U for utility, or W for CGI Web interface

Program Id is a number assigned to that program by the Architect’s Office

Application name is a name associated with the application the executable belongs to, and which must be in the applications file.

Table 4.9-36 is a sample of the file format.

Table 4.9-36. Executables file format

Program	IU Name	Type	ProgID	Application Name
EcClOdProductRequest	EcClOdAsterOnDemand.iu	C	1000005	EcClOdProductRequestApp
EcClDtDesktopDaacUser	EcClDtDesktopDaacUser.iu	G	1000100	EcClDtDesktopApp
EcClDtDesktopSciUser	EcClDtDesktopSciUser.iu	G	NONE	EcClDtDesktopApp
NOTES:				
Program Types				
S- Server	C – Client	G – Gui	T – Test Program	U – Utility
W – CGI Web Interface				

Hostmap File (.hostmap)

The hostmap file has a header and uncommented entries of the form *site : host.* This file maps the hosts at each site to the site name. Table 4.9-37 is a sample of the file format.

Table 4.9-37. Hostmap file format

Site : Host
LP DAAC:e4iil01

Installtypes Files (.installtypes)

The installtypes file format is described in Table 4.9-38. It consists of a file type tag, the permissions to be used for the installation of that type, three location fields describing where the information comes from, and one describing where it goes in the mode. The fields use the standardized macros in the eamacros file.

Table 4.9-38. Installtypes file format

Installtype	Description
[File Type]	Type of file to install in square brackets
Permissions	Permissions to be set on installation
Orig MODE staging Copy	Location to copy from for MODE staging
Orig CM_BUILD Copy	Location to copy from for CM_BUILD
Orig NORMAL or STAGE Copy	Location to copy from for Normal or STAGE
Dest. Copy	Location in mode to copy to
Binary Files [BIN] 755	
${MODE_STAGING_DIR}/CUSTOM/bin/${SUBSYS_DIR_MODE}	
${CMTOP/${SUBSYS_DIR_CM}/install/CUSTOM/bin/${SUBSYS_DIR_MODE}	
${CMTOP}/${STAGE}/${SUBSYS_DIR_CM}/bin/${ARCH}	
${BINDIR}	

Package (PKG) Files

A package file supports exactly the same format as an IU file, but is generally used only to include IU files with the %include command. This allows the system designers to install a set of IU files as one package.

Packages File (.packages)

The packages file specifies which packages are delivered for a particular architecture. The package name, subsystem, and component identify the package. The platforms are specified with linux tags separated by pipe symbols "|". The EcCoMkDeliver script uses the packages file to determine which packages need to be delivered when making the tar files for a drop on a particular architecture. Table 4.9-39 is a sample of the file format.

Table 4.9-39. Packages file format

Package	Subsys	Comp	Platforms	Comment
.EcClEOSView.pkg	CLS	EcCl	Linux	
.EcClINTFCSVR.pkg	CLS	EcCl	Linux	
.EcClINTFCSVR_EDC.pkg	CLS	EcCl	Linux	
.EcClJdt.pkg	CLS	EcCl	Linux	
EcCsCommon.pkg	CSS	EcCs	Linux	
EcCsINTFCSVR.pkg	CSS	EcCs	Linux	
EcCsINTFCSVR_EDC.pkg	CSS	EcCs	Linux	
EcCsINTFCSVR_GSFC.pkg	CSS	EcCs	Linux	
EcCsRegistryGUI.pkg	CSS	EcCs	Linux	

Sitehostmap File (.sitehostmap)

The sitehostmap file specifies the mapping of symbolic hosts to actual hosts at the various DAACs. The format is to use one column per site with the first column indicating the symbolic host name. The format is getting to be unwieldy as we add new DAACs. Putting the whole thing into a database solves this. The sample format is shown in table 4.9-40.

Table 4.9-40. Sitehostmap file format

Symbolic Name	LPDAAC	GSFC	LaRC	NSIDC	PVC
FtpServer02					p4ftl01

Sitemap File (.sitemap)

The sitemap file specifies the packages that can be installed on particular hosts at particular sites by ECS Assistant. The entries consist of a 4-tuple consisting of site: host : subsystem : component followed by a list of packages, one to a line. The format is shown in Table 4.9-41.

Table 4.9-41. Sitemap file format

Entry	Package(s)
EDC :e4mdl01 :MSS :EcMs # Media Distribution Server 1	EcMsWzCommon.pkg

Subsystems File (.subsystems)

The subsystems file consists of four fields. The four fields are:

- A *two letter acronym* (obsolete)
- A *project acronym,* which is used as the principal identifier of a subsystem in all ECS Assistant code
- A *clearcase location,* which specifies the location of the ECS Assistant files in clearcase relative to /ecs/formal
- A mode location, which specifies the name of the directory used in the modes relative to the COMMON/bin, COMMON/lib, COMMON/www, directories)

There are other directories, which have subsystem sub-directories. The format is shown in Table 4.9-42.

Table 4.9-42. Subsystems file format

Two Letter Acronym	Project Acronym	Clearcase Location	Mode Location	Description
cl	CLS	CLS	CLS	Client
cs	CSS	CSS	CSS	Communications
dm	DM	DM	DMS	Data Management
ds	DSS	DSS	DSS	Data Server
es	ESDT	ESDT	ESDT	ESDT
om	OMS	OMS	OMS	Order Manager
ms	MSS	MSS	MSS	Management
v0	VOC	V0_Client	V0_Client	V0 Client

ECS Assistant GUI

The ECS Assistant GUI calls the ECS Assistant Script Library for what it needs to interface with the underlying installation system. It calls the ECS Assistant Script Library for the functions shown in Table 4.9-43.

Table 4.9-43. Functions

Function	Description
Cfgpatch	Patch parameter files based on the .cfgpatch file.
Check_dirs	Check for the existence of specified directories.
Clean_logs_dir	Remove logs in the log directory older than a specified date.
Get_health	Display installation health indicators for mode, subsystem and component.
Get_included_files	Get names of included files from a package.
Get_revlevel	Display the revision level of a parameter file.
Install	Install to a mode.
List_apps	List applications in a mode, subsystem and component.
List_components	List components of a subsystem.
List_execs_by_app	List executables, which belong to the specified application.
List_mkcfgs	List the Mkcfg scripts, which are available for execution.
List_servers	List servers in a subsystem and component.
List_servers_from_file	List servers in a .servers file.
Lookup_component	Lookup details on a component.
Mk_cds_entry	Make CDS entries for a subsystem and component.
Revert_parms	Recover the most recently time stamped parameters file.

ECS Start Scripts

The ECS Start scripts call the ECS Assistant Script Library for a variety of services, which are shown in Table 4.9-44.

Table 4.9-44. Start Scripts (1 of 2)

Start Script	Functionality
check_dirs	Check for the existence of specified directories.
Get_architecture	Display the ARCH variable that identifies the machine.
Getuname	Print user name.
getview	Print Clearcase view or NONE.
Install_type	Determine the installation type of an installation.
Look_for_proc	Look for a process.
Record_proc	Record a process.

Table 4.9-44. Start Scripts (2 of 2)

Start Script	Functionality
set_cfgparms	Set configurable parameters in the environment.
Set_environment	Write the appropriate commands to set up the standard environment.
Start_client	Start an EMD client program.
Start_ecs	Start the whole EMD system.
Start_gui	Start an EMD GUI.
Start_server	Start a single EMD server with standard checks.
Start_server_group	Start a group of servers with standard checks.
Strip_comments	Strip comments and blank lines from a file.

ECS Mkcfg Scripts

The ECS Mkcfg scripts call the ECS Assistant Script Library for a variety of services, which are shown in the Table 4.9-45.

Table 4.9-45. Mkcfg Scripts

Mkcfg Script	Functionality
check_dirs	Check for the existence of specified directories.
Get_architecture	Display the ARCH variable that identifies the machine.
Get_hostname	Display the HOSTNAME variable that identifies the machine.
Get_site_acronym	Get the standard three-letter acronym of the site we are running on.
Getuname	Print user name.
set_cfgparms	Set configurable parameters in the environment.
Set_environment	Write the appropriate commands to set up the standard environment.
Set_fileperms	Set file permissions.
Strip_comments	Strip comments and blank lines from a file.

4.9.4 Systems Management Subsystem Hardware Components

4.9.4.1 Systems Management Hardware (MSSHW) Description

The MSSHW includes the following: Application Servers, Blade Management Servers, Configuration Management (CM) servers, MSS File Servers, one DNS/NIS Servers, Tape Backup Servers, and multiple management PCs.

Document 920-TDx-001 (HW Design Diagram) provides descriptions of the System Management HWCI and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping

The SMC environment has two MSS Application Servers, a DNS/NIS Primary Server, the Knowledge Base Server and the Remedy PC. This environment is unique and will not change with evolution.

The Blade Management Server will be installed at the PVC and at each of the evolution DAACs. The DNS/NIS servers will be hosted on evolution servers at each of these DAACs.

The CM Server will remain at LP DAAC only. The key MLCI function is the Remedy software used for trouble ticket generation and tracking.

The MSS File Server has been migrated to a commodity server. Each of the DAACs have an HP DL380 attached to an EMC CX500 RAID hosting the home directories, the automounted COTS and the custom code distribution.

The Tape Backup Server is a either an IBM Blade HS20 or an HP DL360 running the Legato software to perform daily backups to the Quantum Pathlight Virtual Tape Library. This is configured as an appliance with an EMC CX-300 disk array and a Scalar 100 tape library. The LTO3 tape drives are used to move from virtual tape to physical tape for longer retention. The key MCI functions are the network backup and restore components (EMC NetWorker).

Multiple Pentium PCs are used at each DAAC site in support of office automation requirements. These are standalone hosts, which enable operators to perform policy and procedure management. One MSS software component runs on each PC.

In the past, custom code and applications are loaded on the internal disks of all hosts. This was to prevent dependencies on specific hosts or any peripherals. For cost efficiency, the evolution servers in the IBM Blade Center will boot from the SAN. This will allow for quick recovery in the event of hardware failures.

4.10 Internetworking Subsystem (ISS) Overview

The Internetworking Subsystem (ISS) contains one hardware configuration item (HWCI), the Internetworking HWCI. INCI provides internetworking services based on protocols and standards corresponding to the lower four layers of the OSI reference model as described below.

Transport Protocols

EMD provides IP-based connection-oriented and connectionless transport services. The connection-oriented service is implemented using TCP, while User Datagram Protocol (UDP) is used for connectionless transport. Higher layer applications use one or the other based on such requirements as performance and reliability.

Transmission Control Protocol (TCP), specified in RFC 793, is a connection-oriented, end-to-end reliable protocol designed to fit into a layered hierarchy of protocols to support multi-network applications. It provides for reliable inter-process communication between pairs of processes in host computers attached to networks within and outside EMD. Because TCP assumes it may obtain potentially unreliable datagram service from the lower level protocols, it involves additional overhead due to the implementation of re-transmission and acknowledgment processes.

The UDP, specified in RFC 768, provides a procedure for application programs to send messages to other programs with minimal overhead. The protocol is transaction oriented and delivery of data is not guaranteed, since there is no acknowledgment process or re-transmission mechanism. Therefore, applications requiring ordered and reliable delivery of data would use TCP.

Network Layer Protocols

The network layer provides the functional and procedural means to transparently exchange network data units between transport entities over network connections, both for connection-mode and connectionless-mode communications. It relieves the transport layer from concern of all routing and relay operations associated with network connections.

The Internet protocol (IP) Version 4, specified in RFC 791, is the EMD supported network protocol, based on its dominance in industry usage and wide community support. As part of IP support, ICMP and ARP are also supported.

Physical/Datalink Protocols

Physical and data-link protocols describe the procedural and functional means of accessing a particular network topology. For the DAAC and SMC networks, the data-link/physical protocol is 10/100/1000 Mbps Ethernet.

Internetworking Hardware HWCI (INCI)

This HWCI provides the networking hardware for internal and external DAAC and SMC connectivity. The HWCI includes Ethernet switches and cabling; routers and cabling; and network test equipment. Each network hardware device is discussed in detail in Section 4.10.2.

4.10.1 Internetworking Subsystem Description

4.10.1.1 DAAC LAN Architecture

This section provides an overview of the DAAC network architecture. Information on DAAC specific implementation level detailed designs can be found in Section 4.10.1.5.

The generic architecture for DAAC Local Area Networks (LANs) is illustrated in Figure 4.10-1. The topology consists of a Production Network, and a SAN LAN Network. A Portus Firewall protects the Production network. Each of the networks is discussed in more detail below.

Note that not all sites have the complete complement of hardware and subsystems shown in Figure 4.10-1. For instance NSIDC's EMD router also has a direct connection to NASA Integrated Services Network (NISN), EDC does not have an EMD router, and LaRC does not have an EMD router or Portus firewall.

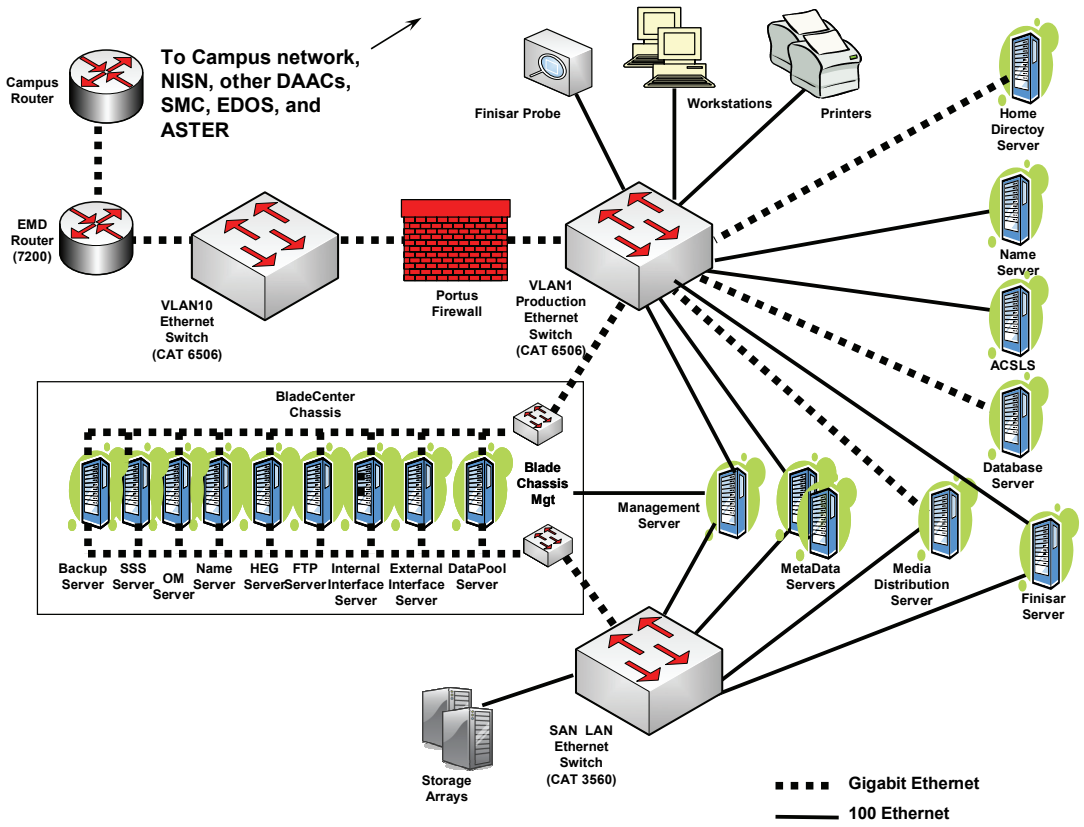


Figure 4.10-1. DAAC Networks: Generic Architecture Diagram

The Production Network consists of a Catalyst 6506 multi-port Ethernet Switch. All servers, workstations, printers, and the BladeCenter chassis are connected to individual switch ports.

The SAN LAN Network consists of a Catalyst 3560 multi-port Ethernet Switch. This network is used for the StorNext file system MetaData and to manage the storage arrays. All servers which use the StorNext file system and storage arrays are connected to this network.

All servers in the BladeCenter chassis connect via the two internal Ethernet switches to both the Production and SAN LAN Ethernet switches.

4.10.1.2 SMC Network Architecture

The SMC network architecture, as illustrated in Figure 4.10-2, consists of a Catalyst 2924 Ethernet Switch. Each server, workstation, printer, and x-term is connected to individual switch ports. A Portus Firewall protects the SMC network.

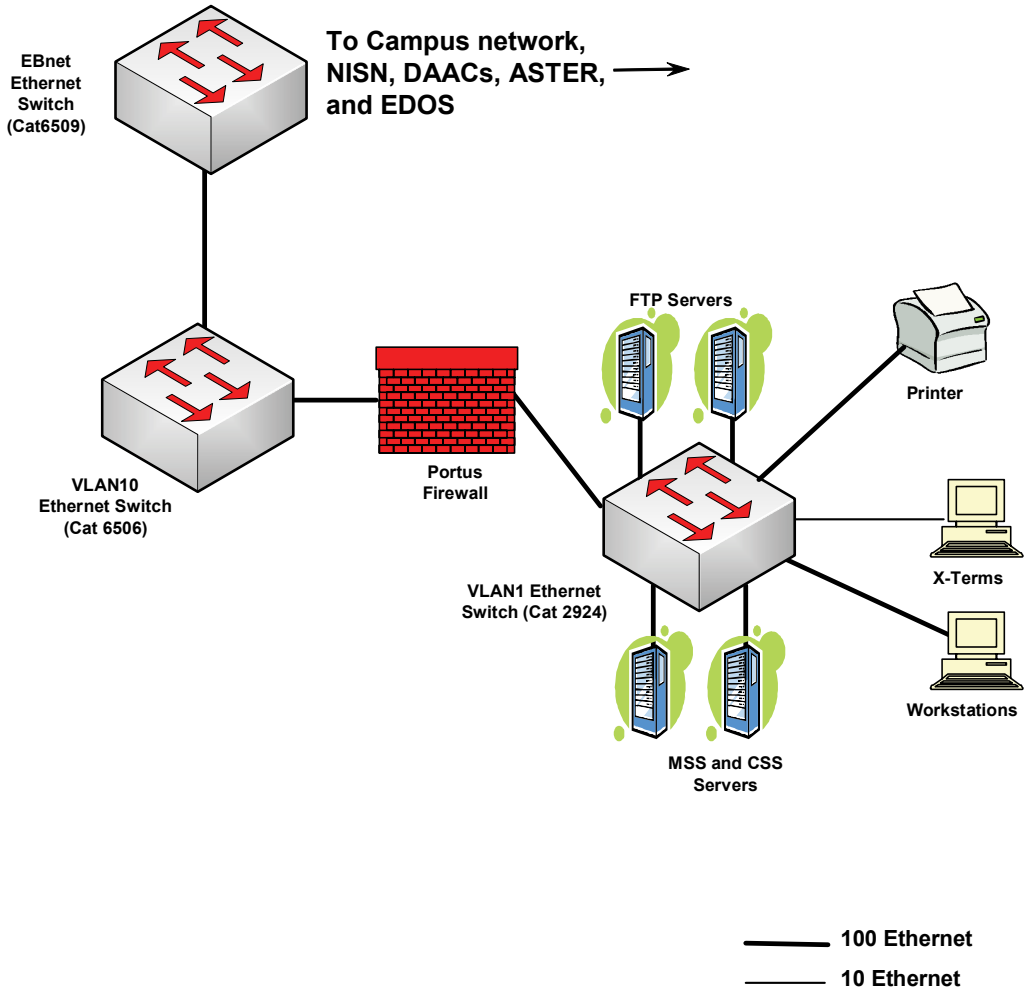


Figure 4.10-2. SMC Network Architecture Diagram

4.10.1.3 DAAC Addressing and Routing Architecture

All devices connected to the Production Network are assigned Class C address space. All devices connected to the SAN LAN are assigned private addresses as specified in RFC 1918 (as of 02/96). Documents that list IP address assignments to all hosts and network attached devices are listed in Table 4.10-1. All EMD address space (except for addresses used on the SAN LAN Ethernet networks) is provided from address blocks designated by NISN.

The use of static routes is the main protocol used to route IP packets within EMD. Routing Information Protocol (RIP) is used to route IP packets from the PVC and VATC Production networks. EMD Production Networks are advertised to all EMD via NISN.

4.10.1.4 Network-based Security Architecture

The network architecture provides a strong level of security by implementation a Proxy Firewall (Portus). This firewall blocks incoming network traffic unless there is a rule specifically allowing the traffic to pass into the DAACs and SMC. Note that in addition to network-based security; EMD has implemented other security measures, such as secure shell (SSH) and host access lists (ACLs), which are discussed in the CSS sections of this document.

4.10.1.5 Internetworking Subsystem Detailed Design

The ISS implementation level detailed design is documented in the documents listed in Table 4.10-1. Document 920-TDx-001 (HW Design Diagram) provides descriptions of the ISS HWCI and document 920-TDx-002 (Hardware-Software Map) provides site-specific hardware/software mapping.

All of the documents are under configuration control and can be obtained from EMD Configuration Management. . The documents are not on line for security reasons. Therefore special authorization is needed for their release.

Table 4.10-1. Internetworking Subsystem Baseline Documentation List

Document Name	EDC	LaRC	NSIDC	SMC
Hardware/Network Diagram	921-TDE-002	921-TDL-002	921-TDN-002	921-TDS-002
Host IP Address Assignment Table	921-TDE-003	921-TDL-003	921-TDN-003	921-TDS-003
Network Hardware IP Address Assignment	921-TDE-004	921-TDL-004	921-TDN-004	921-TDS-004

4.10.2 Network COTS Hardware

The DAAC and SMC LANs contain three types of COTS hardware: Firewall, Ethernet switches, and Routers. All hosts in the DAACs and SMC are attached to Ethernet switches. The Routers

are used to provide access to external networks (NISN, Abilene, and Campus nets). Table 4.10-2 provides a list of networking hardware used in EMD networks.

The following descriptions of Network Hardware devices are provided as illustrative detail. All details of the hardware configuration should be verified with the appropriate Hardware/Network documents listed in Table 4.10-1.

Table 4.10-2. Networking Hardware for EMD Networks

Networking Hardware	Vendor
Firewall	Portus IBM PowerPC Server
Router (EMD Router)	Cisco 7200
Ethernet Switch	Catalyst 3560G
Ethernet Switch	Catalyst 6506
Ethernet Switch	Catalyst 2924
Ethernet Cables	10baseT, 100baseT, or 1000baseT connection to servers, workstations, printers, PCs, and x-terms
Fiber Cable	1000baseSX connection to host

4.10.2.1 EMD Ethernet Switch

The EMD Ethernet switch is the Cisco Catalyst 6506 with multiple 10/100/1000 Mbps ports and powerful packet engines. The switch has a switching fabric of 32Mbps. It forms the core of the EMD Production network by interconnecting all servers, workstations, printers, PCs, and x-terms. The switch has redundant power supply and fan units. It also has redundant packet engines. All modules are hot swappable.

4.10.2.2 EMD Router

The EMD Router is a Cisco 7200 series router running Cisco's Internetwork Operating System (IOS). The router has three 1000 Mbps Ethernet ports. The EMD Router is only used at NSIDC and it provides connectivity to EMD sites and the Internet via its interfaces with NISN and the local campus network.

The ECS Router has redundant power supply and fan units.

For support purposes, the PVC and VATC in Landover also have 7200 routers which interface with EBnet at GSFC.

4.10.2.3 SAN LAN Ethernet Switch

The EMD SAN LAN Ethernet switch is a Cisco Catalyst 3560 switch capable of supporting up to 48 10/100/1000 Mbps ports. .

4.10.2.4 Firewall

The EMD Firewall is an IBM PowerPC Server. It is a Proxy type firewall, which is capable of supporting several 100/1000 Mbps Ethernet interfaces. 1000 Mbps interfaces are used for the

Production network. All Production networks are connected to the firewall. In addition, the SMC is connected to the firewall.

Note: All M&O networks are connected to their local Campus network.

At LP DAAC, the firewall interfaces directly with the Campus routers which provide all external network connectivity.