4. Release B Architecture Overview

4.1 Introduction

Sections 4 and 5 provide a hierarchical overview of the Release B architecture and design. Section 4 discusses the Release B context within the EOSDIS Ground System and its interfaces to all external systems as well as cross DAAC Interfaces. The section also discusses the partitioning of Release B into segment and subsystems. Section 5 continues this partitioning to software and hardware configuration items and local area network (LAN) design. Much of this material is abstracted from the subsequent detailed design documents (305-CD-021-002 through 305-CD-039-002) and is presented here only as an overview. These documents continue the system partitioning through the use of object models and associated material. The reader is referred to those documents for a more in-depth understanding of the Release B design.

4.2 Release B Context Description

Release B provides capabilities to support the Release B Objectives described in Section 3. The Release B external interfaces to ECS and between ECS components (including the SMC and the DAACs) that support Release B mission operations capabilities are presented in Appendix A.

Release B continues the operational support provided in Release A for the TRMM spacecraft including interfaces to SDPF and TSDIS and full operational capability for the ingest, product generation and archival of TRMM data for CERES and LIS instruments, and receipt, archiving and distribution of TRMM data products received from TSDIS. In addition Release B operational support provides access to all V0 data including data still held in V0 data servers through V0 interoperability, and V0 data migrated to ECS data servers. Release B also provides V0 users access to all ECS held data, thus providing two way V0 interoperability.

This section provides a context for understanding the Release B architecture and design components and their mission at each EOSDIS Facility. The remaining sections of this document provide an overview of the Release B design and as such do not deal specifically with the configuration of components at each EOSDIS Facility. For more information on the site unique configurations refer to the DAAC Implementation subdocuments (305-CD-030-002 through 305-CD-037-002) and the SMC Design Specification (305-CD-038-002). Additional details on the Release B external interfaces can be found in the Release B External Interface Control Documents listed in Section 2.

4.3 Release B Architecture

The ECS design has been partitioned into three segments, namely:

- A Science Data Processing Segment (SDPS) which is responsible for ECS applications which provide
  - data management and archiving functions,
  - a processing environment for the execution of science software,

- external interfaces for the acquisition of data needed for processing or intended for archiving, and
- functions which support the search and retrieval of ECS managed data by science and other users., and augmented by subsetting and resampling capabilities.

- A Communications and System Management Segment (CSMS) which is responsible for all communications, networking, and enterprise management functions, including
  - a distributed applications and operating system infrastructure,
  - various communications services such as electronic mail and file transfer,
  - billing and accounting,
  - security,
  - monitoring and management of networking, system, and application resources,
  - access control, security, time synchronization, and reliable communications among applications within a distributed environment,
  - local area network services and external network connectivity,
  - Mode Management, to allow multiple operating environments,
  - fault and performance monitoring, and
  - user accounting
- A Flight Operations Segment (FOS) which is responsible for space craft and instrument command and control functions.

This design overview document addresses the Release B capabilities of Science Data Processing, and the Communications and System Management capabilities they require. It does not include FOS and its supporting CSMS functions, which are the subject of separate documents.

The following sub-sections provide an overview of the SDPS and CSMS subsystems. Section 5, which provides the formal breakdown of the segments below the level of subsystems, into software configuration components (CSCI) and hardware configuration components (HWCI), including the functions allocated to each and pointers to the subsystem detailed design subdocuments of DID 305 which describe the design of each of these components.

4.3.1 SDPS Segment Architecture

SDPS is composed of seven (7) subsystems. They provide the hardware and software resources needed to implement the SDPS functionality. The subsystems are defined in the SDS and have been presented at SDR and IDR. This section provides a brief review (see Figure 4.3-1).

The SDPS supports the services required to ingest, process, archive, access and manage science data and related information from the entire EOSDIS. More specifically the SDPS will provide hardware, software and operations to:

- ingest, process, archive and manage all data from EOS instruments, NASA Flight of Opportunity (FOO) missions, other selected remotely sensed data, and their associated data products;

- receive, process, archive and manage ancillary data required by the EOSDIS science software;
- receive, archive and manage *in situ* correlative data;
- provide the Earth science community with access to all EOS data and other Earth Science data held by the ECS and the data products resulting from research using these data;

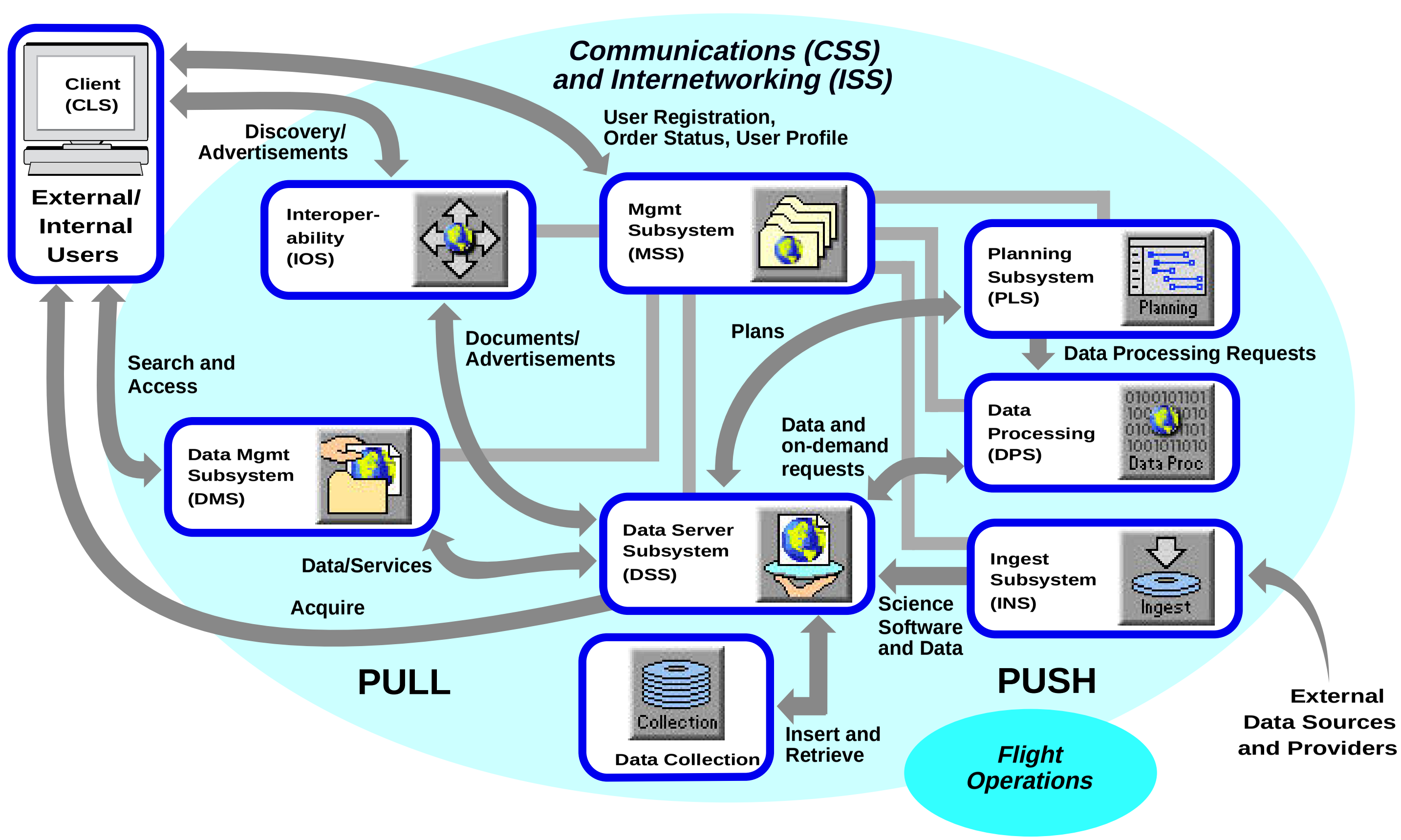

Figure 4.3-1. ECS Context Diagram

Further details on these objectives are provided in ECS Science Requirements Summary White Paper (194-WP-902-002).

The subsystems can be grouped into the following four categories:

- Data storage and management is provided by the Data Server Subsystem (DSS), with the functions needed to archive science data, search for and retrieve archived data, manage the archives, and stage data resources needed as input to science software or resulting as output from their execution.

  A key driver in the architecture of this subsystem was that the costs for data storage and processing capacities are in rapid decline, hand-in-hand with improvements in throughput and response times. In the long term, therefore, ECS will be best served with an approach which affords the full capabilities of database management to science data, even though this approach cannot be fully implemented in the near future. Therefore, the Data Server Subsystem provides access to earth science data in an integrated fashion through an Application Programming Interface that is common to all layers.

- Information search and data retrieval is provided by the science user interface functions in the Client Subsystem (CLS), by information search support functions in the Data Management Subsystem (DMS), and by capabilities in the Interoperability Subsystem (IOS) which assist users in locating services and data of interest to them and their projects.

  The following were key drivers in the architecture of these subsystems. First, these capabilities should be capable of evolving to meet the needs of the wider GCDIS or UserDIS, as described in Section 3. Second, ECS should provide the basic underpinning for an earth science workbench, the assumption being that some of the science community will be in a much better position to develop the data viewing, search, and analysis tools they need which should fit into this workbench. Finally, as the Data Server Subsystem evolves to provide DBMS functionality for earth science data, the data access subsystems should be able to undergo a similar evolution to support this concept for the end user.

- Data processing is provided by the Data Processing Subsystem (DPS) for the science software; and by capabilities for long and short term planning of science data processing, as well as by management of the production environment provided by the Planning Subsystem (PLS).

  Routine data processing and re-processing will occur in accordance with the established production plans. In addition ECS will provide "on-demand processing", where higher level products are produced only when there is explicit demand for their creation. The ability to evolve to a mix of routine production and on-demand processing as users ask for data not currently stored by the Data Server Subsystem is a key driver in the subsystems' software architecture. The need to provide highly scalable and configurable processing resources is the main driver in their hardware architecture.

- Data ingest is provided by the Ingest Subsystem (INS), which interfaces with external applications and provides data staging capabilities and storage for an approximately 1-year buffer of Level 0 data (so that reprocessing can be serviced from local storage).

  The number of external interfaces which ECS will have is potentially very large, and the interfaces can serve very diverse functions, such as high-volume ingest of level 0 data and low-volume ingest of data from field campaigns. Thus, the ability to dynamically configure all or a large part of an external interface from standardized components is a key driver of the subsystem's software architecture. Resiliency against failure and loss of ingested data is a key driver in its hardware configuration.

The following sub-sections provide brief overviews for each of these subsystems. More detailed discussions of their design breakdown can be found in Section 5 and in the corresponding 305 design subdocuments.

Context diagrams summarizing the data flows for each subsystem with other subsystems and with external-to-ECS entities are presented in Appendix B, Section B-1.

4.3.1.1 Client Subsystem (CLS)

The SDPS client subsystem has three main objectives:

- provide earth science users with an interface via which they can access ECS services and data
- offer an environment into which science users can integrate their own tools
- give science programs access to the ECS services, as well as direct access to ECS data

The client subsystem software, therefore, consists of graphic user interface (GUI) application, tools for displaying the various kinds of ECS data (e.g., images, documents, tables), and libraries representing the client APIs to ECS services. Modern user interfaces are based on an object paradigm. The SDPS client subsystem is no exception; the graphic user interface programs will follow an object oriented design. The design will be built around a core set of 'root' objects from which all other software will inherit its behavior. This will lead to a consistent look and feel and reduce the amount of software that needs to be developed. This core set is called the desktop. The remainder of the software is collectively called the workbench.

For Release A, the client subsystem consists of the desktop, a user interface which allows users to search and browse a database describing the data and services available within ECS (called the advertising database), and a data visualization tool (EOSView). The remainder of the Release A user interface will be provided by an enhanced version of the V0 System Client (also referred to as the Release A Client). It provides data search and access for ECS science data, and a browsing interface for Guide documents and other types of ECS document data. In Release B the V0 System Client will be replaced, but the existing V0 client will be supported through the V0 interoperability gateway. A new set of client tools will be provided to provide search and access of ECS science data sets. The data search tool will be separated from the product request tool in order to allow users more flexibility in the way they acquire data. The users will no longer be required to submit a search for data before ordering. If the user knows the location of data, an order can be constructed without the search phase. These tools along with other new tools are collectively called the "workbench".

For the purposes of the SDPS design, it has been decided to group interfaces required by operational staff with the respective subsystem, and within each subsystem, with the respective CSCI. This ensures that operational interface requirements are properly addressed and presented in context.All custom GUI development follows the guidelines developed by a team of Human Factors Engineering (HFE) experts. The guidelines for Release B user interfaces, including those found in the client subsystem, are documented in the updated version of the ECS User Interface Style Guide, Version 6. The client subsystem components will be deployed on the hardware components which support operational staff, and provision for the necessary user interface hardware exists in the respective HWCI. The client subsystem is downloaded by science users to their local workstations. More detail on CLS can be found in 305-CD-021-002, Release B SDPS Client Subsystem Design Specification.

4.3.1.2 Interoperability Subsystem (IOS)

The SDPS interoperability subsystem provides an "advertising service.” It maintains a database of information about the services and data offered by ECS, and provides interfaces for searching this database and for browsing through related information items. The Client Subsystem provides a user interface which enables scientists to locate services and data that may be of interest to them. This software is being developed in an incremental fashion, and early versions have been made available in Evaluation Packages (EP) to provide science user feedback to the developers.

The full functionality of the advertising service will be implemented in Release A. The only enhancements that will be made to the advertising service in Release B are to incorporate the Earth Science Query Language(ESQL) interface and the standard query protocol.

4.3.1.3 Data Management Subsystem (DMS)

The Data Management subsystem provides three main functions:

- Provide end-users with a consolidated logical view of a distributed set of data repositories
- Allow end-users to obtain descriptions for the data offered by these repositories. This also includes descriptions of attributes about the data and the valid values for those attributes.
- Provide data search and access gateways between ECS and external information systems.

The subsystem includes distributed search and retrieval functions and corresponding site interfaces. Release A uses the capabilities of the Version 0 IMS to provide both the user search interface, and the intersite search functions. In Release B, the Data Management subsystem provides the distributed search and access capabilities across wide area and local area networks using ECS query languages and protocols. There will be a common language and protocol across this subsystem, Interoperability subsystem, and the Data Server subsystem.

In addition, the subsystem includes an interface between ECS and the Version 0. Release A implements the Version 0 to ECS gateway, in support of the Version 0 IMS user interface. The gateway accepts Version 0 requests (i.e., directory search, inventory search, browse, order and guide access), and translates them to allow access to the ECS Data Server Subsystem. The gateway also reformats the results of search and data access requests into the formats used by the Version 0 IMS. In Release B, the reverse direction of the gateway will be developed. The ECS clients and client API will be able to issue queries using the ECS query language and protocol to the Version 0 gateway and it will translate the requests into Version 0 requests. ECS gateways will follow a general gateway architecture to allow reuse of design components. This architecture is presented in Section 6.

4.3.1.4 Data Server Subsystem (DSS)

The Data Server subsystem provides the management, cataloging, access, physical storage, distribution functions for the ECS earth science data repositories, consisting of science data and their documentation. The Data Server provides interfaces for other ECS subsystems which require access to data server services. The Data Server Subsystem consists of the following principal design components:

- Database Management System - The Data Server subsystem will use database technology to manage its catalog of earth science data, and for the persistence of its system administrative and operational data.
- Document Management System - Web server and database technology are used to implement a document management system to provide storage and information retrieval for guide documents, science software documentation, and ECS earth science related documents.
- Data Type Libraries - The Data Server will use custom dynamic linked libraries (DLLs) to provide an extensible means of implementing the variety of ECS earth science data types and services, and will provide a consistent interface for use by other ECS subsystems requiring access to those services and data.
- File Storage Management System - Archival and staging storage for large volumes of data will be provided through the use of several hardware/software configurations which are either off-the-shelf or a mixture of off-the-shelf and developed software.
- Distribution System - The Data Server provides the capabilities needed to distribute bulk data via electronic file transfer or physical media.

Additional components support subscription service support, and system administration and configuration.

4.3.1.5 Ingest Subsystem (INS)

This subsystem deals with the initial reception of all data received at an EOSDIS facility and triggers subsequent archiving and processing of the data.

Given the variety of possible data formats and structures, each external interface, and each ad-hoc ingest task may have unique aspects. Therefore, the ingest subsystem is organized into a collection of software components (e.g., ingest management software, translation tools, media handling software) from which those required in a specific situation can be readily configured. The resultant configuration is called an ingest client. Ingest clients can operate on a continuous basis to serve a routine external interface; or they may exist only for the duration of a specific ad-hoc ingest task.

The ingest subsystem also standardizes on a number of possible application protocols for negotiating an ingest operation, either in response to an external notification, or by polling known data locations for requests and data. The subsystem will use the components of the general ECS external interface architecture which are presented in Section 6.

Special considerations exist with respect to the handling and archiving Level 0 data. This is a major and mission critical part of the SDPS interfaces and design (see SDPS Ingest Subsystem Design Specification [305-CD-025-002]).

4.3.1.6 Data Processing Subsystem (DPS)

The main components of the data processing subsystem - the science algorithms - will be provided by the science teams. The data processing subsystem will provide the necessary hardware resources, as well as a software environment for queuing, dispatching and managing the execution of these algorithms. The processing environment will be highly distributed and will consist of heterogeneous computing platforms. ECS has selected a product (AutoSys) which is designed to manage production in a distributed UNIX environment. AutoSys is being integrated with custom software designed to manage distributed, heterogeneous computing resources.

The DPS also interacts with the DSS to cause the staging and de-staging of data resources in synchronization with processing requirements. In addition, Release B is implementing "predictive staging" to identify the appropriate time to stage PGE input data. More details are in the SDPS Data Processing Subsystem Design Specification [305-CD-027-002].

4.3.1.7 Planning Subsystem (PLS)

The Planning Subsystem provides the functions needed to plan routine data processing, schedule on-demand processing, and dispatch and manage processing requests. The subsystem provides access to the data production schedules at each site, and provides management functions for handling deviations from the schedule to operations and science users. The ECS processing environment poses a number of challenges:

- a processing environment which eventually will be highly distributed and consist of heterogeneous computing platforms
- existence of inter-site and external data dependencies
- dynamic nature of the data and processing requirements of science algorithms
- need for high availability
- providing a resource scheduling function which can accommodate hardware technology upgrades
- support for on-demand processing (as an alternative to predominantly routine processing)
- ability to provide longer-term (e.g., monthly) processing predictions as well as short term (e.g., daily) planning and scheduling

To support the complexity of and demands on the processing environment, it was decided to separate long term planning as a custom software implementation from short term planning and scheduling as provided by commercial off the shelf software (AutoSys). Production planners will be able to study the impact of changes in resource allocation or processing plans (e.g., the introduction of substantial reprocessing) on overall schedules. At the same time, operations personnel can make decisions regarding the daily workload in a timely fashion, interacting with a highly responsive database, and using a robust commercial product specifically designed for that purpose. More detail on PLS can be found in 305-CD-026-002, Release B Planning Subsystem Design Specification.

4.3.2 CSMS Segment Architecture

Release B has been designed as a fully distributed, heterogeneous system. To implement this, SDPS will make use of the services which are provided by Communications and System

Management Segment (CSMS), most notably, its Communications Subsystem (CSS) capabilities. The following are a few examples:

- SDPS internal interfaces typically use the CSMS Server Request Framework and the Distributed Object Framework.
- A number of COTS software components of SDPS are currently using UNIX sockets (e.g., Sybase), and will rely on that CSMS service.
- Many external interfaces as well as the bulk data transfers within SDPS will use CSMS file transfer services.
- SDPS will utilize the Process Framework for fault and event reporting, and to provide and obtain security management information.
- Some SDPS components will generate electronic mail messages (e.g., Science Data Server notices, and subscription notices sent via electronic mail).

In addition, a bulletin board service, electronic mail, file transfer and distributed file services, telnet, and other CSMS services which can be invoked directly by a user via a CSMS (or operating system) provided user interface will be used by science users and operational staff to perform various SDPS related functions (e.g., operations might notify a scientist of the arrival of a software delivery vie e-mail).

CSMS supplies security management and authentication services to SDPS. SDPS will add security services for internally managed objects whose structure is transparent to CSMS (e.g., there will be additional security capabilities within the Data Server CSCIs).

The CSMS accomplishes the interconnection of users and service providers, transfer of information between ECS (and many EOSDIS) components, and enterprise management of all ECS components. It supports and interacts with the Science Data Processing Segment (SDPS) and the Flight Operations Segment (FOS).

The services provided by CSMS at the System Monitoring and Coordination Center, (SMC) located at Goddard Space Flight Center (GSFC), are collectively referred to as Enterprise Monitoring and Coordination (EMC) throughout this document. In the same context, services provided by CSMS at Distributed Active Archive Centers (DAACs) and the EOC (sites) are collectively referred to as Local System Management (LSM).

At its highest design level, CSMS consists of three parts:

- System Management Subsystem (MSS)

  MSS is a collection of applications which manage all ECS resources, including all SDPS, FOS, ISS, and CSS components. MSS directly uses CSS services.

- Communications Subsystem (CSS)

  CSS is a collection of services providing flexible interoperability and information transfer between clients and servers. CSS services correspond loosely to layers 5-7 of the Open Systems Interconnection Reference Model (OSI-RM).

- Internetworking Subsystem (ISS)

  ISS is a layered stack of communications services corresponding to layers 1-4 of the OSI-RM. CSS services reside over, and employ, ISS services.

The following sub-sections briefly describe the CSMS subsystems and characterize their relationships with one another, SDPS and FOS, and external entities discussed above.

4.3.2.1 Communications Subsystem (CSS)

CSS plays a key role in the interoperation of the SDPS subsystems. SDPS applications follow an object-oriented design. That is, their lowest level software components are "software objects". SDPS also implements a distributed design, e.g., its software objects are distributed across many platforms. For the software objects to communicate with each other requires "distributed object" communications environment. This environment is provided by CSS, using off-the-shelf technology augmented with some custom software. The environment allows software objects to communicate with each other reliably, synchronously as well as asynchronously, via interfaces that make the location of a software object and the specifics of the communications mechanisms transparent to the application.

In addition, CSS provides the infrastructural services for the distributed object environment. They are based on the Distributed Computing Environment (DCE) from the Open Software Foundation (OSF). DCE includes a number of basic services needed to develop distributed applications, such as remote procedure calls (rpc), distributed file services (DFS), directory and naming services, security services, and time services.

Finally, CSS provides a set of common facilities, which include legacy communications services required within the ECS infrastructure and at the external interfaces for file transfer, electronic mail, bulletin board and remote terminal support. The Object Services support all ECS applications with interprocess communication and specialized infrastructural services such as security, directory, time, asynchronous message passing, event logging, lifecycle service, transaction processing and World Wide Web (WWW) service. More information on CSS can be found in 305-CD-028-002, Release B CSMS Communications Subsystem Design Specification.

4.3.2.2 Management Subsystem (MSS)

The Management Subsystem (MSS) provides enterprise management (network and system management) for all ECS resources: commercial hardware (including computers, peripherals, and network routing devices), commercial software, and custom applications. With few exceptions, the management services will be fully decentralized, such that no single point of failure exists which would preclude the system from continuing to operate or system operations and management to come to a halt.

MSS provides two levels of an ECS management view: the local (site/DAAC specific) view, provided by Local System Management (LSM), and the enterprise view, provided by the Enterprise Monitoring and Coordination (EMC) at the SMC.

Enterprise management relies on the collection of information about the managed resources, and the ability to send notifications to those resources. For network devices, computing platforms, and some commercial of the shelf software, MSS relies on software called "agents" which are usually located on the same device/platform and interact with the device's or platform's control and application software, or the commercial software product.

However, a large portion of the ECS applications software is custom developed, and some of this software - the science software - is externally supplied. For these components, MSS provides a set of interfaces via which these components can provide information to MSS (e.g., about events which are of interest to system management such as the receipt of a user request or the detection of a software failure). These interfaces also allow applications to accept commands from MSS, provided to MSS from M&O consoles (e.g., an instruction to shut down a particular component).

Applications which do not interact with MSS directly will be monitored by software which acts as their "proxies". For example, the Data Processing Subsystem (DPS) acts as the proxy for the science software it executes. It notifies MSS of events such as the dispatching or completion of a PGE, or its abnormal termination.

ECS selected HP OpenView as the centerpiece of its system management solution, and is augmenting it with other commercially available "agents", as well as custom developed software (e.g., the applications interfaces mentioned above). The information collected via the MSS interfaces from the various ECS resources is consolidated into an event history database, some on a near real-time basis, some on a regular polling basis (every 15-to 30 minutes) as well as on demand, when necessitated by an operator inquiry. The database is managed by Sybase, and Sybase query and report writing capabilities will be used to extract regular and ad-hoc reports from it. Extracts and summaries of this information will be further consolidated on a system wide basis by forwarding it to the SMC (also on a regular basis). These reporting capabilities and strategies are further discussed in Section 6.

MSS provides fault and performance management and other general system management functions such as security management (providing administration of identifications, passwords, and profiles); configuration management for ECS software, hardware, and documents; Billing and Accounting; report generation; trending; request tracking; and mode management (operational, test, simulation, etc.).

4.3.2.3 Internetworking Subsystem (ISS)

The ISS is a layered stack of communications services corresponding to layers 1-4 of the Open Systems Interconnect Reference Model (OSI-RM). The ISS provides local area networking (LAN) services at ECS installations to interconnect and transport data among ECS resources. The ISS includes all components associated with LAN services including routing, switching, and cabling as well as network interface units and communications protocols within ECS resources.

The ISS also provides access services to link the ECS LAN services to Government-furnished wide-area networks (WANs), point-to-point links and institutional network services. Examples include the NASA Science Internet (NSI), Program Support Communications Network (PSCN), and various campus networks "adjoining" ECS installations. More detail of ISS is provided in Section 5 of this document.

4.3.3 Operator Positions

The operations staff is an intricate part of the overall design architecture. The following charts defines the operator positions defined for ECS operations. Three charts are presented for operator position definition: SMC Operator Positions, DAAC Operator Positions, and EOC Operator Positions. A more detailed discussion of these operator positions can be found in 607-CD-001-001, Maintenance and Operations Manual for the ECS Project.

Table 4.3.3-1 SMC Operator Positions

Position	Responsibilities
SMC Operations Supervisor	Scheduling and personnel manager, Responsible for operations (policies, installation, readiness, reporting, training, etc.)
SMC Resource Controller	Responsible for SMC Hardware, Software, LAN, DCE Cell Configuration
SMC Fault Manager	Performs real-time fault analysis,
SMC Security Controller	Supports development of security policies, reviews implementation, identifies security training requirements, coordinates with other security organizations. Monitors, audits, and reports on security of system.
SMC Computer Operator	Initializes, configures and operates SMC hardware, software, and EOC office staff workstations.
SMC Network Configuration Manager	Coordinates with external network operations organizations. Manages ECS-wide network. Maintains inter-site inventory database. Maintains inter-site logical network topologies.

Table 4.3.3-2 DAAC Operator Positions

Position	Responsibilities
DAAC Operations Supervisor	Responsible for the performance of all "on-line" operations personnel and resources.
DAAC Production Planner	Populates and maintains the Production Planning Database. Develops, coordinates and maintains Data Availability Schedules.
DAAC Production Monitor	Monitors Data Processing Request validation. Monitors/manages processing queues to optimize utilization. Provides real-time science product QA support.
DAAC Resource Manager	Responsible for site hardware, software, LAN and local DCE cell configuration.
DAAC Archive Manager	Responsible for site science data ingest, storage, and distribution operations performance. Supervises Data Ingest-Distribution Technicians.
DAAC Ingest-Distribution Technician	Monitors and reports performance of data request, arrival, and delivery schedules
DAAC Computer Operator	Initializes, configures and operates site hardware and software.
DAAC Data Specialist	Provides on-site expertise on use and development of data and metadata, subsets of data, numerical methods and tools, etc.
DAAC User Assistant	Assists users in locating and accessing EOSDIS-related data. Assists users with problems in ECS environment.

Table 4.3.3-3 EOC Operator Positions

Position	Responsibilities
Operations Controller/Shift Supervisor	Lead EOC position on shift. Responsible for shift briefings/debriefings. Monitors EOC activity timelines, lead real-time communications anomaly resolution.
Command Activity Controller	Reviews timeline for upcoming TDRSS support, controls ground script execution, verifies command load contents, transmits reconfiguration commands, verifies command and load uplink.
Spacecraft Evaluator	Reviews timelines for upcoming TDRSS events, verifies graphical strip chart configuration, monitors ground script, monitors spacecraft subsystem health and safety.
Instrument Evaluator	Reviews timelines for upcoming TDRSS events, verifies graphical strip chart configuration, monitors health and safety of instruments, reviews alarms and limits.

4.4 Requirements for Incremental Development

Two development tracks are being used on the ECS project: formal track development and incremental track development.

Formal development is characterized by relatively long development cycles (18-24 months) with formal reviews, documentation, and testing. This development approach is typically used for mission critical areas of the system. Therefore, the CIs that comprise the Data Server, Ingest, Planning, Internetworking, System Management and Data Processing subsystems are being developed using the formal development approach. The Communications Subsystem, that was on the incremental track in Release A, has moved to the formal track for Release B

Incremental development is characterized by a sequence of short development cycles (6-9 months each), with each increment building upon the previous one. It is used for areas of the system where it is desirable to obtain early user feedback and to minimize the turnaround time required to incorporate this feedback into the system.

The CIs that comprise the Client, Interoperability, and Data Management Subsystems are being developed on the incremental track. It is anticipated that user feedback will impact the Level 4 requirements; therefore, the Level 4 requirements for the Client, Interoperability, and Data Management subsystems are draft requirements. Final as-built requirements will be available after the increments are complete.

More detailed information on the ECS development approach can be found in the Systems Engineering Plan for the ECS Project (194-201-SEI-001).

This page intentionally left blank.