Resource Ordering and Status System (ROSS) Program

Professional Development Services

System and Software Design Document (SSDD)

Revision 1

February 19, 2001

LOCKHEED MARTIN

Contract: GS-35F-4863G
Delivery Order Number: GA81
Lockheed Martin Reference Number: GA8120010219R1 System and Software Design Document

Revision Chart

Version	Description of Version	Date Completed
Final	Final deliverable.	2/5/01
Revision 1	Incorporates client hardware requirements and updates to the client software requirements.	2/19/01

Table of Contents

Tables

Figures

Appendices

1. Introduction

1.1 Purpose of Document

The **purpose** of the System/Software Design Document (SSDD) is to provide an overall description of the Resource Ordering and Status System (ROSS) design and to specify the hardware and software system detail design.

The SSDD covers the following subject areas relative to the ROSS application:

- Primary functional users
- System architecture
- System software design
- System hardware design
- Software Application Design
- Database Design

1.2 Scope of Document

The **scope** of the System/Software Design Document (SSDD) is constrained by the ROSS requirements stated in the ROSS Software Requirements Specification (SRS) / Requirements Traceability Matrix (RTM) as derived from the document sources outlined below:

- Statement of Work (SOW) for Task Order 43-82X9-9-5140, September 27, 1999
- ROSS System Design Document, September 20, 1999
- ROSS Infrastructure Requirements, May 1, 1999

1.3 Intended Audience

The intended audience for the SSDD consists of those persons involved in the deployment, maintenance, support, operations, and training activities associated with ROSS. The SSDD provides sufficient information along with the System Administration Manuals and User Manuals for these teams to accomplish their respective tasks without reliance on the authors of the SSDD.

1.4 Document Identification

The System/Software Design Document (SSDD) satisfies the following deliverable requirements identified in the statement of work for Task Order 43-82X9-9-5140 under Contract GS-35F-4868G.

- Task #2: ROSS Design and Documentation Updates
  - Documented revisions to the system architectural design and Infrastructure Requirments Plan
  - Updates to the ROSS Physical Data Model that occur as a result of design refinement during the build phase
  - Validate and provide updates to any plans or documentation created durint the exectuion of Task Order 43-82X9-9-5042

1.5 References

The following documents of the exact issue shown form a part of this document to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this document, the contents of this document shall be considered a superseding design.

- *ROSS System Design Document (SDD)*, September 20, 1999

- *Resource Ordering and Status System (ROSS) Task Order Statement of Work*, January 24, 1999 (including all appendices)
- *ROSS System Requirements Document/Requirements Traceability Matrix (RTM)*, October, 2000
- *The Windows Interface Guidelines for Software Design*, © 1998 Microsoft Corporation http://msdn.microsoft.com/isapi/msdnlib.idc?theURL=/library/books/winguide/platfrm2/d5/s115b5.htm
- *Analysis and Discussion Related to the Use of the Multi-Agency Incident Resource Processing System (MIRPS) on a Nationwide Basis*, Working Copy May 12, 1999

2. System Overview

Each year, wildland incident dispatch centers mobilize massive amounts of resources to combat wildfires and other hazards that threaten public lands throughout North America. In some instances, resources are allocated to assist wildland incident teams acting abroad, as well. Resources include aircraft, crews, equipment, overhead (personnel) and supplies. The primary functional process to be supported by ROSS is the resource order and status process of the national interagency wildland incident dispatch organizations associated with the National Wildfire Coordination Group (NWCG). Among the agencies that participate in the NWCG are the National Fire Protection Agency, State Foresters, United States Fire Administration, USDA Forest Service, USDI Bureau of Indian Affairs, USDI Fish and Wildlife Service, USDI National Park Service and USDI Bureau of Land Management. Personnel from any of these organizations may work together at an interagency dispatch unit.

Currently, manual processes are used to accomplish the resource order and status process. Resource orders are prepared at each dispatch unit and relayed in hardcopy or electronic form (Figure 2-1) to other units, where the data is manually re-entered. This process continues until a resource is found to fill the order. Status information is gathered and documented by local dispatch units, then transmitted daily to Geographic Area Coordination Centers (GACCs) and to the National Interagency Coordination Center (NICC), located in Boise, Idaho. Because status data is normally 12-18 hours old when it is transmitted, it cannot reliably be used to assess resource availability when resource orders are taken.

The ROSS design fulfills the documented functional, data and performance requirements to automate the current manual resource ordering and status business process of the NWCG. Using accurate and timely information delivered by ROSS, wildland fire managers can provide faster incident response and increase the cost-effectiveness of resource mobilization and demobilization.

INCIDENT/PROJECT ORDER NUMBER

RESOURCE ORDER **EQUIPMENT**	Initial Date/Time	2. INCIDENT/PRJECT NAME				3. INCIDENT/PROJECT ORDER NUMBER			4. OFFICE REFERENCE NUMBER
5. DESCRIPTIVE LOCATION/RESPONSE AREA		6. SEC	TWN	RNG	Base MDM	8. INCIDENT BASE/PHONE NUMBER			9. JURISDICTION/AGENCY
		7. MAP REFERENCE							10. ORDERING OFFICE
11. AIRCRAFT INFORMATION		LAT				LONG			

BEARING	DISTANCE	BASE OR OMNI	AIR CONTACT	FREQUENCY	Ground Contact	FREQUENCY	RELOAD BASE	OTHER AIRCRAFT/HAZARDS

12. Request Number	Ordered Date/ Time	From / To	Q T Y	RESOURCE REQUESTED	Needed Date/Time	Deliver To	From / To	Time	Agency ID	RESOURCE ASSIGNED	ETD / ETA	RELEASED Date	RELEASED To	Time / ETA

13. ORDER RELAYED Req No.	Date	Time	To/From	Action Taken	ORDER RELAYED Req No.	Date	Time	To/From	Action Taken
.					.				
.					.				
.					.				

Figure 2-1 Resource Order Form for Manual Process

2.1 Primary Functional Users

The primary functional users of the ROSS application will be the dispatchers and support personnel engaged in the resource order and status process. The potential user base is the NICC, eleven GACCs, and approximately 400 3rd and 4th tier dispatch units.

2.2 Functional Context

Figure 2-2 depicts the overall context of ROSS with respect to the resource order and status process. ROSS and its associated ROSS Users are the mechanisms by which the list of inputs is transformed into the list of outputs. Resource Availability and Resource Assignments identify the specialized inputs that constrain the transformation of the inputs into outputs. Figure 2-3 provides a first level functional decomposition of the resource order and status process. The mechanisms in this diagram identify the ROSS developmental software modules that will be described in Section 5.

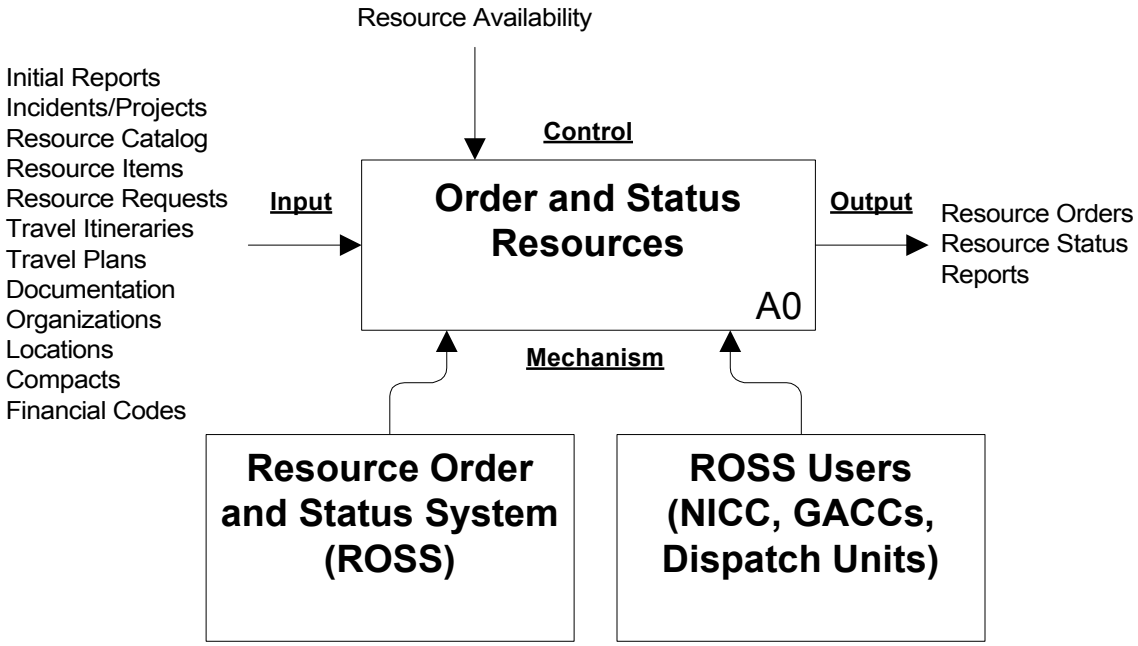

Figure 2-2 Functional Context Diagram

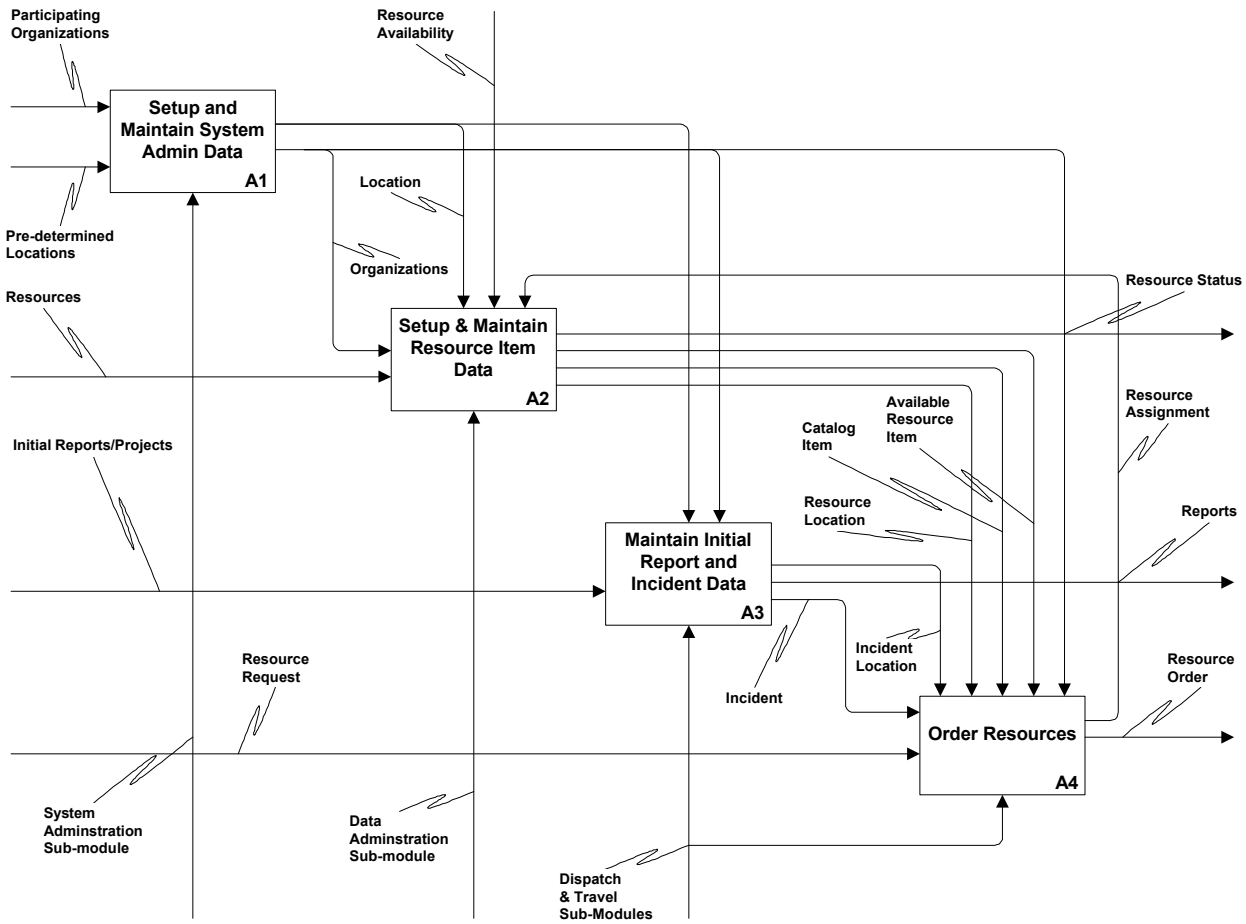

Figure 2-3 First Level Functional Decomposition

2.3 Architecture

The ROSS application is executed within a 3-tier architecture (Figure 2-4). The Client Tier consists of a web browser and Java applets running in a web browser window or Java applications. The Java applets and applications are downloaded from the Application Tier and stored on the Client Tier. The Client Tier supports external interfaces executing as HTML web pages or Java applets through a web browser or as Java applications. The Application Tier contains proxy servers, web servers, servlet engines, locater services, application servers, and the geographic information system (GIS) servers executing under the control of Java applications. The Database Tier contains the data sources. The ROSS architecture allows for a heterogeneous database environment, supporting integration of multiple and disparate data sources.

The next sections specify the system hardware, the non-developmental software components that execute the developmental software, and the developmental software consisting of HTML and Java applets / applications.

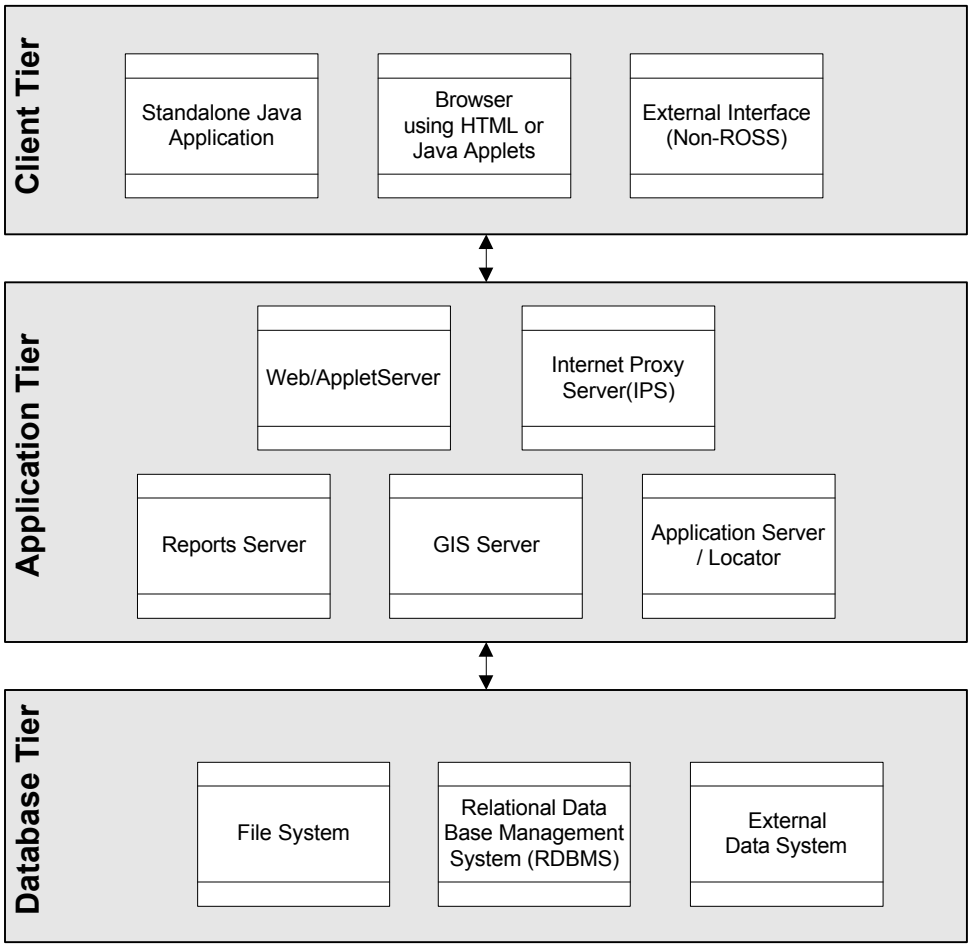

Figure 2-4 ROSS Software Architecture

3. System Hardware

The ROSS hardware consists of a production system and a test system. Each system consists of an AIX operating system component and a Windows NT operating system component. The systems run both developmental and non-developmental software. The non-developmental software for the AIX operating system hosts the Apache HTTP Server as the web server, Versata Logic Server (VLS) as the application server, and Oracle as the database server. The ROSS developmental software application is served through the web server using the VLS to access the Oracle database. The Windows NT operating system hosts Mapinfo MapXtreme GIS (Geographical Information System). The Mapinfo application accesses data from the Oracle database on the RS6000 computers.

3.1 Production System

The AIX hardware component of the production system (Figure 3-1) consists of two identical IBM RS6000 servers sharing a disk array. High availability is achieved through the use of IBM HACMP software to constantly monitor the health of the primary server. In the event the primary server stops functioning, HACMP starts a secondary server as a replacement. HACMP moves all hostname/IP information from the primary to the secondary server and then proceeds to startup the ROSS non-developmental software (i.e., Oracle, Versata, Apache). The HACMP software relies on the shared drive configuration as Oracle, Apache and Versata all exist on the shared drives. Other required software and libraries may exist on the internal disks of the server which will require that both the primary and secondary servers have identical configurations with respect to installed software/applications and

overall configuration. For more information on HACMP consult with the system administration personnel at the host facility responsible for installation and configuration of HACMP.

The Windows NT hardware component of the production system (Figure 3-1) consists of two identical IBM Netfinity Servers sharing a disk array. High availability is achieved through the use of Legato Co-Standby HA software in the same manner as that described above for the HACMP software. For more information on Legato Co-Standby HA consult with the system administration personnel at the host facility responsible for installation and configuration of the Legato Co-Standby HA software. Table 3-1 provides both the AIX and Windows NT production system hardware configurations.

Table 3-1 Production System Hardware Configuration

Item	Model	Qty
AIX Hardware Component		
IBM RS6000 F50 - 4 PowerPC 604e SMP 332 MHz Processors - 3 GB Physical Memory - 2 wide SCSI I/O Controllers - 2 GXT130P Graphics Adapter - 2 IBM SSA 160 Serial RAID Adapter - 2 Wide/Fast SCSI I/O Controllers - 8 Internal 16bit SCSI Disk Drives (9100 MB) - 1 20/40GB SCSI-2 Fast/Wide 8mm Internal Tape Drive - 1 SCSI Multimedia CD-ROM Drive (650 MB) - 2 IBM 10/100 Mbps Ethernet PCI adapter - 1 ASCII Console with Keyboard	F50	2
SSA Disk Enclosure		1
SSA160 Disk Drives (18200 MB)	SSA160	8
Windows NT Hardware Component		
IBM Netfinity 6000R Server - 4.5 GB Physical Memory (ECC SDRAM RDIMM) - Redundant Power Supplies - 1 Integrated 10/100 Ethernet Adapter - 2 PCI 10/100 Ethernet Adapter - 4 700 MHz P3 Xeon Processors - 1 Netfinity EXP300 Storage Expansion Unit - 2 Integrated Ultra 160 SCSI Controllers - 2 Netfinity ServeRaid-3L Ultra2 SCSI Adapter II - 2 Netfinity 9.1GB 10,000RPM Disk Drives	6000R	2

Item	Model	Qty
Ultra 160 How Swap Disk Drive Enclosure	Ultra160	2
Netfinity 18.2GB 10,000RMH Disk Drives		16

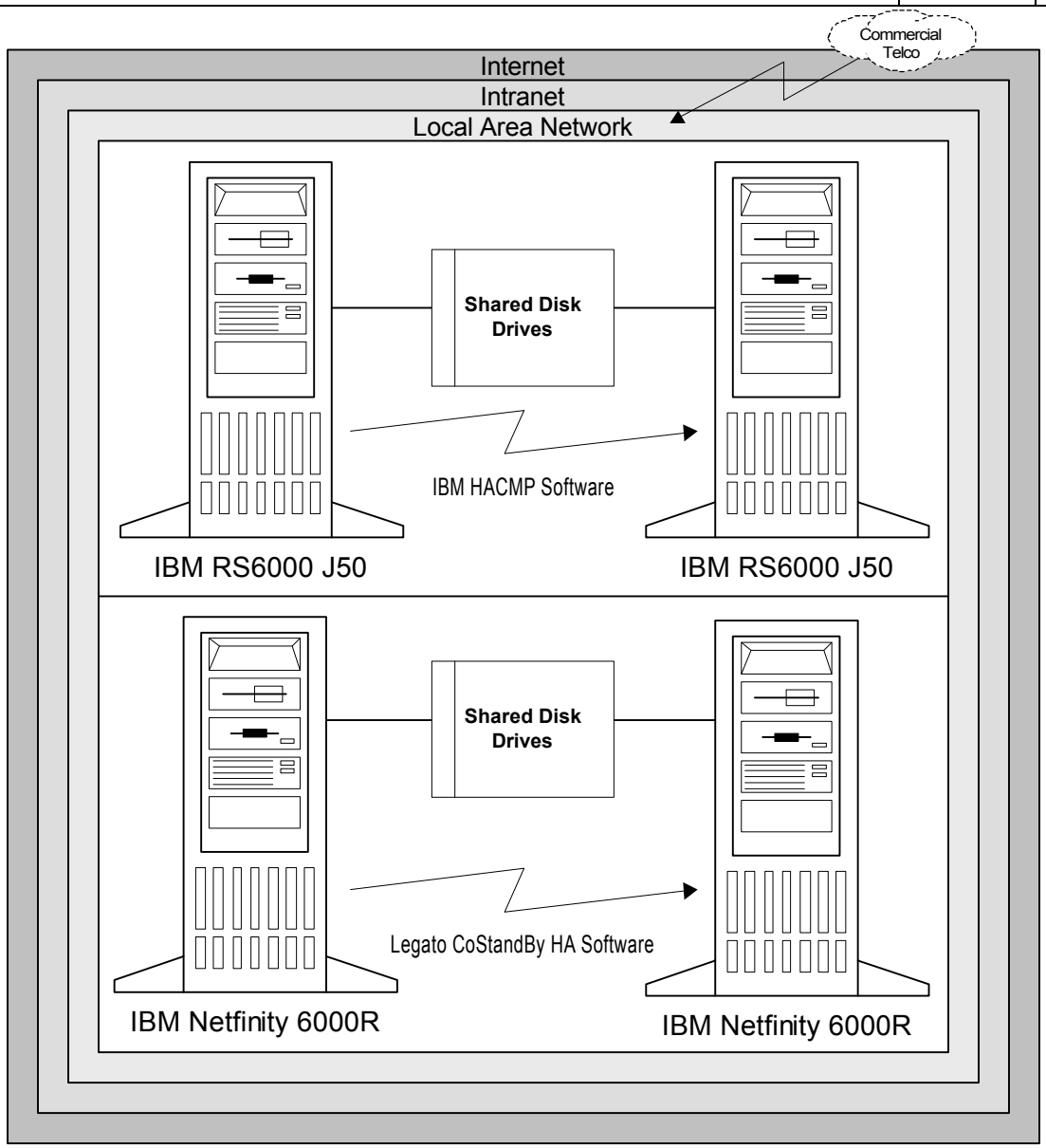

Figure 3-1 Production System Hardware

3.2 Test System

The test system (Figure 3-2) AIX hardware component matches the production system except for the model of the hardware. The Windows NT hardware component differs from the production system by only having one server platform and no high availability configuration. Table 3-2 provides both the AIX and Windows NT test system hardware configurations.

Table 3-2 Test System Hardware Configurations

Item	Model	Qty
AIX Hardware Component		
IBM RS6000 F50 - 6 PowerPC 604e SMP 200 MHz Processor w/2MB L2 Cache - 2 Wide/Fast SCSU I/O Controllers - 2 Internal SCSI Disk Drives (9100 MB) - 1 Internal SCSI Disk Drives (4500 MB) - 1 SCSI Multimedia CD-ROM Drive (650 MB) - 1 10/100 Ethernet MC Adapter / SMP - 2 GB Physical Memory - 1 33.6kbps External Modem - 1 ASCII Console w/ Keyboard	F50	2
7137 SCSI Disk Enclosure	7137	2
SCSI Disk Drives (4.19MB)		8
Windows NT Hardware Component		
Gateway Pentuim III - 1 PIII 900MHz Processor - 1 IDE Internal Disk Drive 36GB - 1 42X CDROM - 384MB Physical RAM - 1 10/100 Ethernet NIC		1

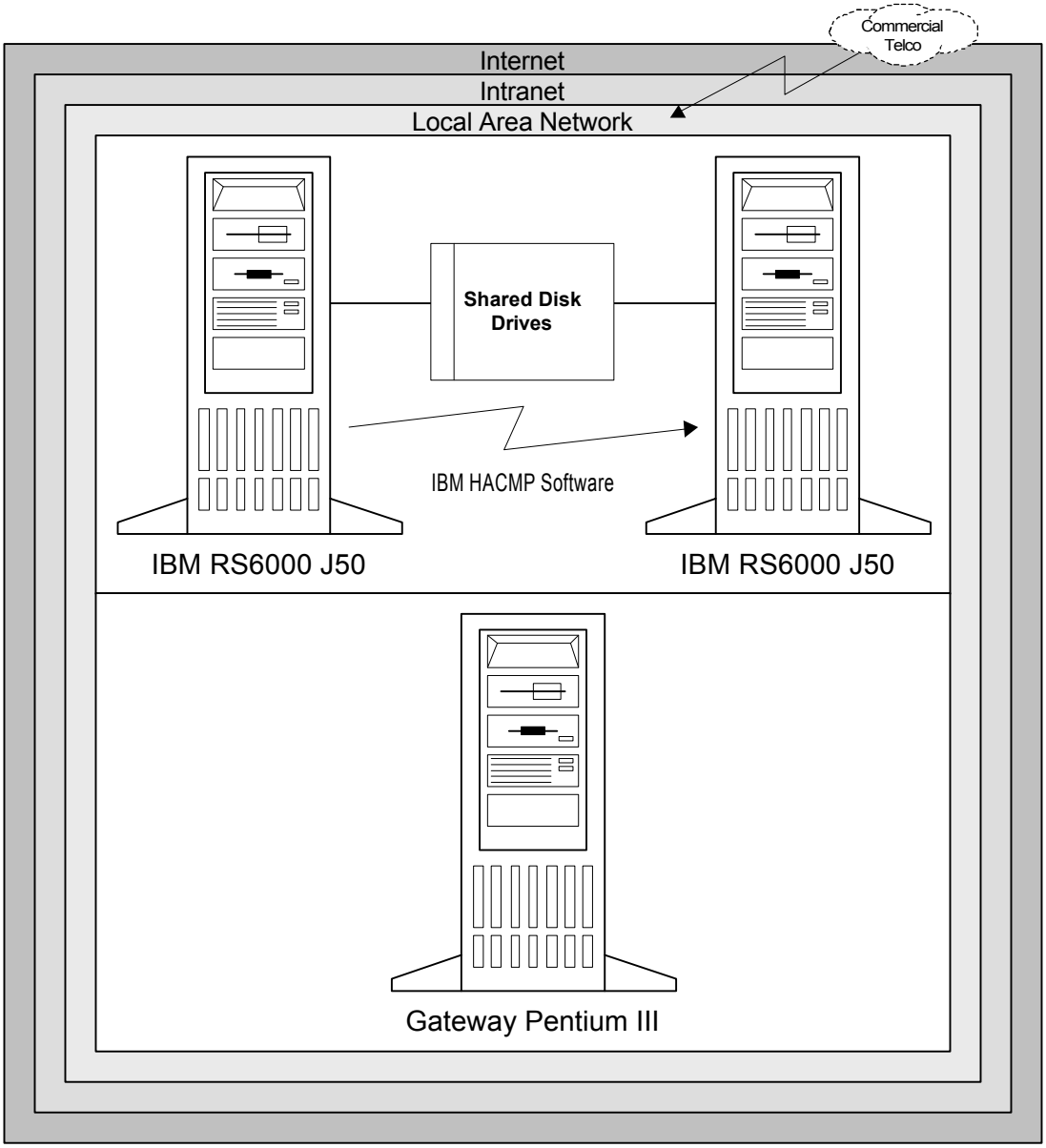

Figure 3-2 Test System Hardware

3.3 Client

The minimum recommended client hardware requirements are as follows:

- Operating System - Microsoft Windows 95, Windows 98, or Microsoft Windows NT 4.0
- Video - Monitor and video card with 1024 x 768 video resolution
- Memory - 64 megabytes of random access memory (RAM)
- Disk Space - 30 megabytes for the ROSS application - recommended
- Central Processing Unit (CPU) - Pentium running at 200 MHz or higher recommended
- Network - Access to the world-wide web (WWW) via a modem connection through an Internet service provider (ISP) or wide area network connectivity

3.4 Development System

The development system is located at the contractor facility in Lakewood, Colorado. This system is used by the contractor to develop the ROSS application and provide initial testing. The development environment does not have any high availability capability. Table 3-3 provides a listing of the development system hardware. Table 3-4 contains a list of the software tools for development provided as GFE by the government. Not all tools were used in the final development of the ROSS application.

Table 3-3 Development System Hardware

Item	Model	Qty
IBM RS6000 Model 43P 150 -1 wide SCSI I/O Controller -1 Graphics Adapter -2 Internal 16 bit Disk Drives (18200 MB) -1 SCSI Multimedia CD-ROM Drive (650 MB) -1 IBM 10/100 Mbps Ethernet PCI adapter -1 GB Physical Memory -1 Processor	150	1
Compaq DeskPro PC -384 MB Physical Memory -IDE Disk Drive (9500 MB) -10/100 Ethernet Card	PII-450	1

Table 3-4 Inventory of Software Packages (GFE)

Copies	Description of Software Package
1	Oracle 8I
2	RoboHELP 2000
2	ER-WIN
11	Niko Mak Comp Win Zip
8	Oracle SQL*Plus
2	Sun Micro Java Web Server
7	Tek-Tools Kawa
1	Vision Jade Business Logic Server
7	Vision Jade Developer Studio
1	Silverrun BPM, ERX, RDM
12	Tower Razor

Copies	Description of Software Package
1	Issue Weaver
1	Tower Razor NT Server
1	Knowledge Based Systems, Inc ProSim

4. Non-Developmental Software

Figure 4-1 provides an overview of the ROSS system (commercial off-the-shelf) software components and the communication protocol used between each set of components without regard to the hardware specification for the location of each component.

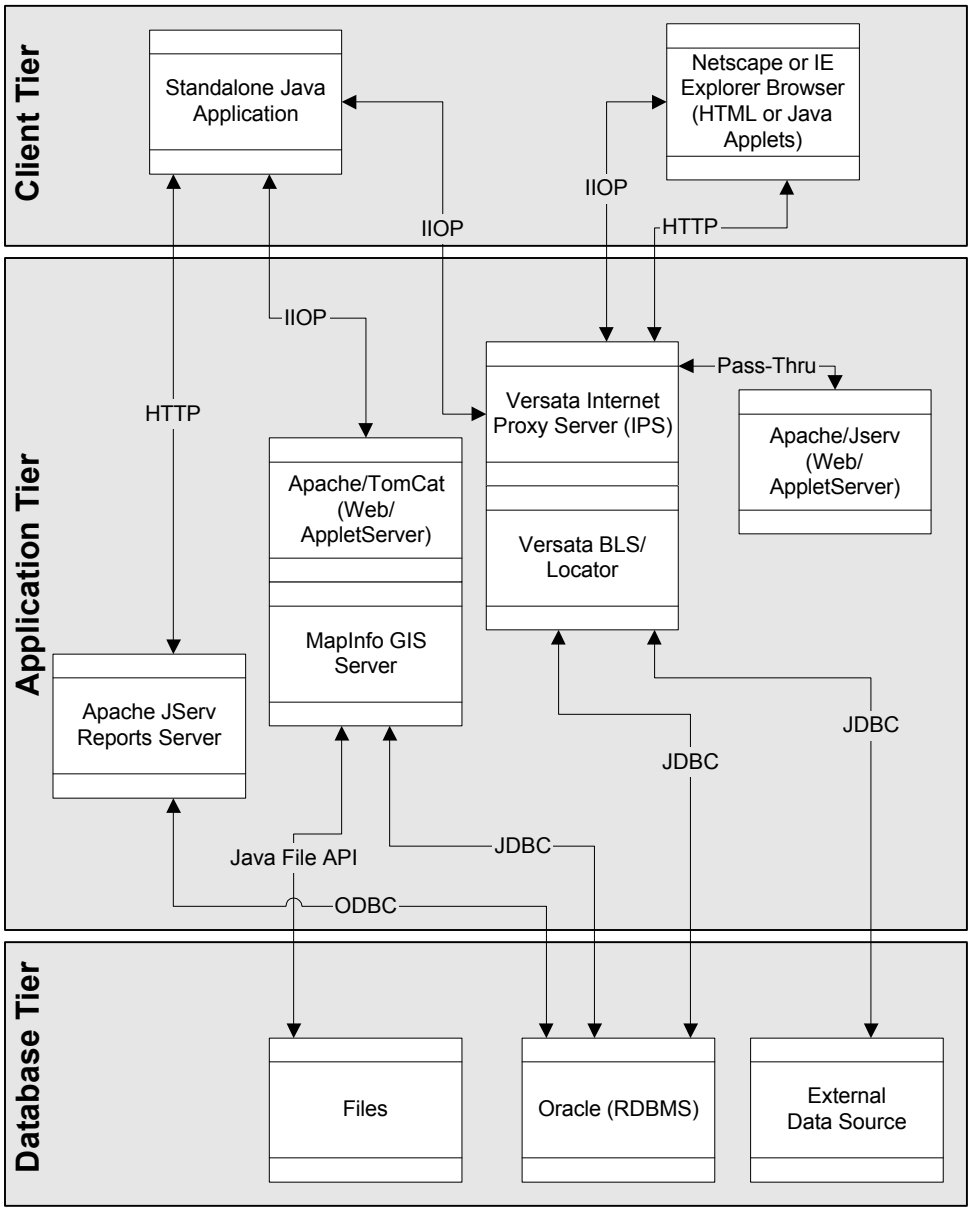

Figure 4-1 ROSS System Software Design

The following sections specify the identified system software components by tier and classification.

4.1 Client Tier Components

4.1.1 Web Browser

Category	Specification
Product	Netscape Navigator 4.5, 4.6, or 4.7 Microsoft Internet Explorer 5.x
Definition	A program that uses the hypertext transfer protocol (HTTP) and secure HTTP (HTTPS) to make requests of web servers throughout the Internet on behalf of the user. The web browser

Category	Specification
	transmits requests from the client to the server; translates hypertext markup language (HTML) from the server into readable pages; and receives and executes Java applets (Java applications that execute within a web browser).
Constraints	The web browser must be capable of displaying HTML 3.2 or higher and running a Java Virtual Machine (JVM) compliant with Sun's JDK 1.3.0.C or higher. The connection that the web browser uses to access ROSS must not block TCP/IP port 80. This component is external to ROSS, it is assumed that the user has an acceptable browser installed and correctly configured.

4.1.2 Java Runtime Environment

Category	Specification
Product	Java Runtime Environment (JRE) - Version 1.3.0
Definition	Java Plug-in software enables enterprise customers to direct applets or JavaBeansTM on their intranet web pages to run using Sun's Java 2 Runtime Environment, Standard Edition (JRE) instead of the web browser's default virtual machine.
Constraints	Support for Microsoft Windows and Sun Solaris-based browsers

4.1.3 Java ™ Certificate

Category	Specification
Product	Java ™ Certificate
Definition	A signed applet file is a mechanism for ensuring an applet can be trusted on the client. For a signed applet to be trusted, the client must trust the key that was used to sign the applet file on the server providing the applet. The Java ™ Certificate provides the mechanism for the client to authenticate the source of the applet.
Constraints	Support for Microsoft Windows and Sun Solaris-based browsers.

4.2 Application Tier Components

4.2.1 Internet Proxy Server (IPS)

Category	Specification
Product	Versata 5.0 Internet Proxy Server
Definition	A proxy server acts as an intermediary between a client and the Internet to ensure security, allow administrative control, provide caching services, and separate an organization's intranet or internal network from the Internet or external network. A proxy server receives a request for an Internet service (such as a Web page request) from a user. If it passes filtering and security requirements, the proxy server, acting as a client on behalf of the user, uses one of its own IP addresses to request the page from the web server. When the page is returned, the proxy server relates it to the original request and forwards it on to the user.
Constraints	To support high availability, the proxy server must have at least one back up.

4.2.2 Web Server

Category	Specification
Product	Apache HTTP Server - Version 1.3.12 (AIX and NT) Note: For future implementations of ROSS, Secure Sockets Layer (SSL) has been included in the installation of Apache HTTP Server. SSL is a commonly-used protocol for managing the security of a message transmission on the Internet. SSL uses a program layer located between the Internet's Hypertext Transfer Protocol (HTTP) and Transport Control Protocol (TCP) layers. SSL is included as part of both the Microsoft and Netscape browsers and most Web server products. SSL was developed by Netscape and is now supported by Microsoft and other Internet client/server developers as well. The "sockets" part of the term refers to the sockets method of passing data back and forth between a client and a server program in a network or between program layers in the same computer. SSL uses the public-and-private key encryption system from RSA, which also includes the use of a digital certificate. RSA is an Internet encryption and authentication system that uses an algorithm developed in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman (RSA). The RSA algorithm is the most commonly used encryption and authentication algorithm and is included as part of the Web browser from Netscape and Microsoft. The encryption system is owned by RSA Security. The company licenses the algorithm technologies and also sells development kits. The technologies are part of existing or proposed Web, Internet, and computing standards. SSL is an integral part of most Web browsers (clients) and Web servers. If a Web site is on a server that supports SSL, SSL can be enabled and specific Web pages can be identified as requiring SSL access. SSL is an alternative to another commonly-used security protocol, HTTPS. Newer browsers support both SSL and HTTPS. Installation includes: mod_ssl - Verion 2.6.6-1.3.12 (Apache interface to openssl), openssl - Version 0.9.5 (opensource toolkit for library which implements ssl), and Rsaref20 (deprecated RSA library for US citizens).
Definition	A web server sends content over the Internet using the Hyper Text Markup Language (HTML). The Web server accepts requests from browsers like Netscape and Internet Explorer and then returns the appropriate HTML documents. Server-side technologies, such as Java applets, are used to increase the power of the server beyond its ability to deliver standard HTML pages.
Constraints	High availability for 24x7 operation

4.2.3 Servlet Engine

Category	Specification
Product	Apache JServ - Version 1.1.2 for AIX TomCat Jakarta - Version 3.1 for NT
Definition	The servlet engine provides a server-side Java Virtual Machine (JVM) in which server-side Java applications (servlets) may be executed.
Constraints	The servlet engine will be capable of 24x7 operation with fail-over. Requires JDK - Version 1.2 with Java Runtime Environment - Version 1.3 or higher and the Java Servlet Development Kit (JSDK) - Version 2.0 Apache Web Server must be available in both AIX and NT environments.

4.2.4 Application Server

Category	Specification
Product	Versata Logic Server - Version 5.0.44
Definition	A server program running in a computer in a distributed network that provides the business logic for one or more software applications.
Constraints	Provides fail-over, fault tolerance, and load balancing Maintains data integrity across the entire application Enforces the ROSS business rules on all transactions

4.2.5 GIS Server

Category	Specification
Product	MapInfo MapXtreme 3.0
Definition	Provides users with the ability to display map information in a graphic format.
Constraints	N/A

4.3 Data Tier Components

4.3.1 Relational Data Base Management System (RDBMS)

Category	Specification
Component	Oracle 8i - Version 8.1.6
Definition	The database server processes requests to create, update, and delete data stored in relational tables on magnetic storage media. It is also used to stage data for migration into ROSS, archiving data, and supporting data requirements of the geographical information system (GIS).
Constraints	Data must be available for use by report writing applications. The model shall be enhanced with keys, indexes, data types, data length, and physical storage design. Any required stored procedures, triggers, functions, and packages shall also be identified. No vendor specific extensions or enhancements to the ANSI standard will be used, in order to preserve vendor independence.

5. Developmental Software

The software for ROSS as been developed using several commercial-off-the-shelf (COTS) tools: Versata Studio, ERWin, and Oracle. Section 5.1 describes the application and database specifications for the ROSS application, Section 5.2 describes the use of these tools to specify and develop the ROSS application, and Section 5.3 provides a functional description of the ROSS application by module and sub-module.

5.1 Application and Database Design Specifications

Versata Studio is used to develop the ROSS application software in a rapid application development (RAD) environment. With the exception of archive and some import and report functions, Versata Studio has been the primary mechanism for specifying and building the ROSS application. Versata Studio is therefore the best vehicle for viewing the application specification representations. Representations include, but are not limited to, hierarchical decompositions of both the application and the data objects, designer screens to view the specifications for each application and data object, a business rules report for the data dictionary, and an application report. The representations satisfy the requirement for an application architecture that depicts the dialog map diagrams to show interactions, access pathways, and navigation possible between displays (screens); program hierarchy diagrams; screen layouts; interface data models; interface specifications, including, but not limited to, editing rules for all data elements, screens or presentation logic, error messages, and navigation options; and software component specifications. Appendix A contains listings of the software development files produced by Studio and the generated code. An overview of Versata Studio is provided in the next section.

The physical data model for ROSS is represented using the ERWin data modeling tool by LogicWorks. ERWin produces various output products including reports and plots of the data model. A plot of the data model is included in Appendix B. While ERWin is used to maintain the data model graphical representation, Versata Studio maintains the data dictionary and deploys the data dictionary to the Oracle DBMS. An extract program is used to pull the database metadata into ERWin and generate the graphical representation.

5.2 Development Environment

The Versata Studio is an object-oriented integrated development environment (IDE) for designing, building, and deploying Web-enabled applications. The Versata Studio includes designers, managers, wizards, and other utilities that automate many of the tasks for developing Java™ and HTML applications, including code generation. Some of the capabilities provided in the Versata Studio are as follows:

- Model database objects and their relationships to each other
- Reengineer and deploy data model information from and to supported database servers
- Input business logic as declarative business rules, using the Versata's SQL-like rules language
- Build business logic into CORBA objects and deploy them to Versata's Logic Server (VLS)
- Design application user interfaces including the forms for each application, the data to be displayed on each form, and the navigation between forms
- Customize with Java™ code
- Build, compile, and test applications.

The ROSS application has been developed in Java and deploys to the Common Object Request Broker Architecture (CORBA) version of the VLS. Studio stores it's repository business data objects in eXtensible Markup Language (XML) files. These files may be read using the latest version of Internet Explorer or any other XML compatible browser.

Figure 5-1 provides a view of the Versata Studio main display. The main display is divided into three separate areas. The left side of the display is devoted the object browser displaying a hierarchical presentation of the repository objects. The two principle types of objects are business objects (data) and applications. The developer uses the object browser to select objects for display in the right side of the display, the work environment for the repository objects. The left side of the main display is also used to provide a file view of the repository objects.

The bottom of the display is used for compilation messages, including error messages, when the developer has chosen to 'build' an application. Once an application has been built and deployed, it is run by the VLS and automatically deployed to a user's client workstation.

The Application Designer, shown in the right side of the main display in Figure 5-1, specifies the hierarchical representation of the components of the CatalogResource application listed in the left window of the display. The

lines between the components specify the transitions between the components. A transition maybe from one component to another on the same screen, to a dialog box, or to an entirely different screen.

Selecting a component in the Application Designer opens the Forms Designer (Figure 5-2) in the right side of the main display. The Forms Designer is the primary tool for specifying the graphical user interface for the application. Prototype screen component designs are automatically generated from the business data objects. These design components may be modified and other components placed on the screen and manipulated using standard 'drag and drop' Windows functions.

Business data objects are the foundation of the Versata IDE. The repository browser display provides an alphabetical listing of all the business data objects and their attributes (Figure 5-3). Selecting a data object opens the Business Rules Designer in the right hand display (Figure 5-4). The Business Rules Designer is used to specify attributes, relationships, constraints, actions and properties for the data object. Data objects may also be reverse engineered from existing data sources. The business data objects must be specified prior to specification of the screen design components.

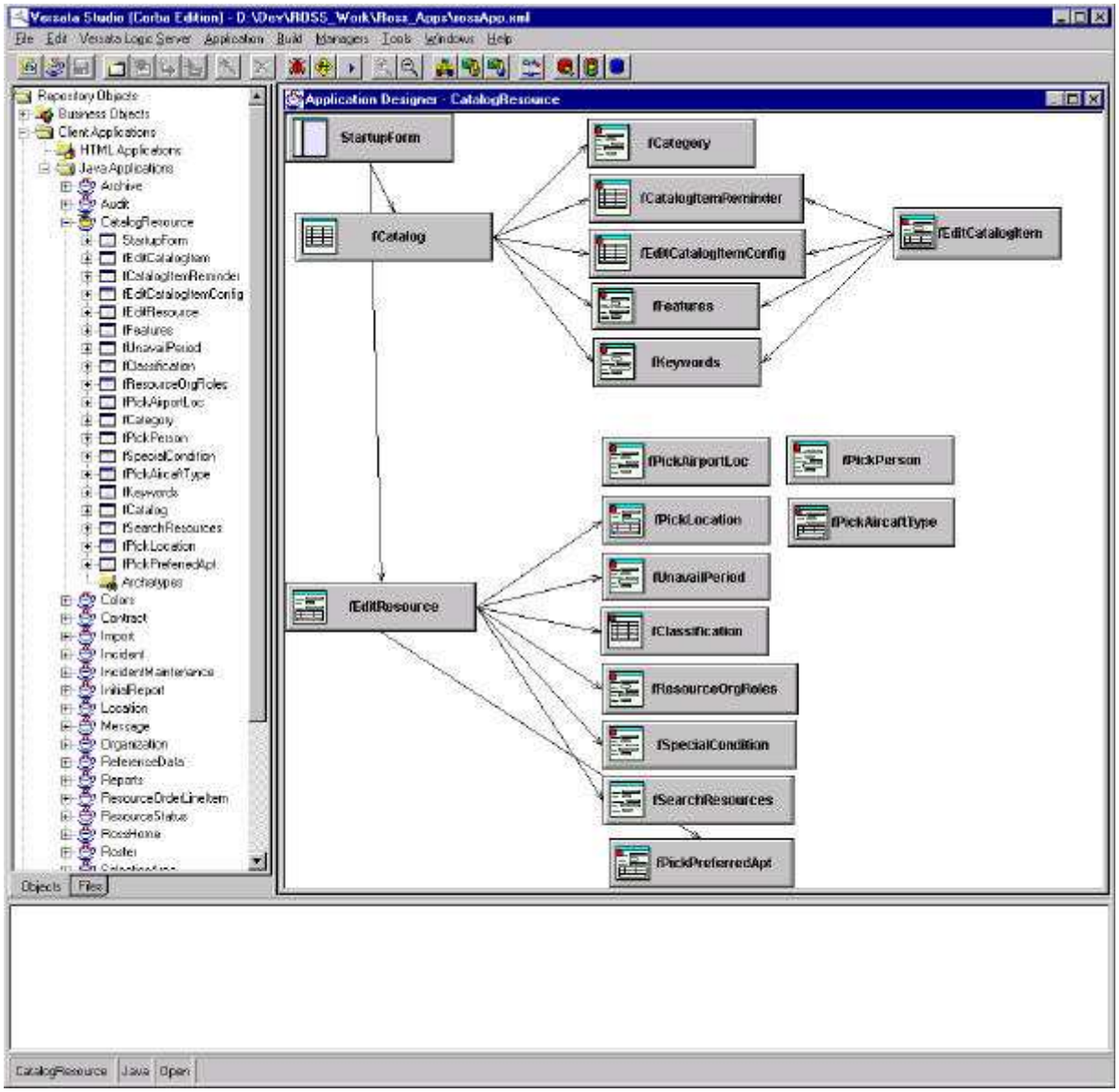

Figure 5-1 Versata Studio Main Display

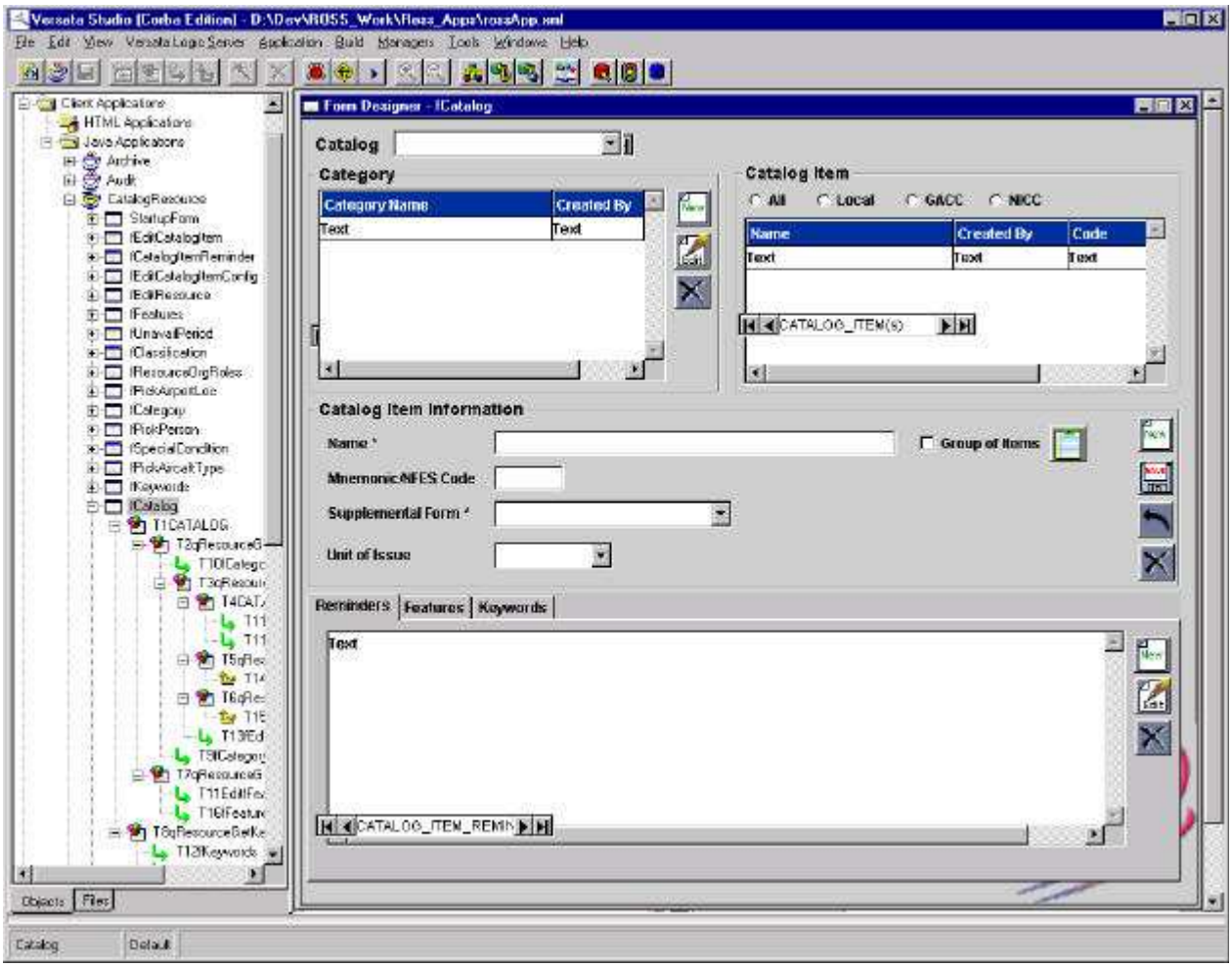

Figure 5-2 Versata Form Designer

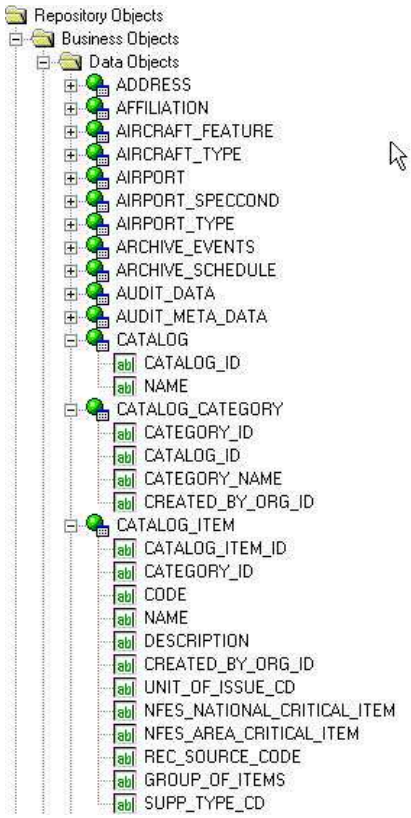

Figure 5-3 Versata Business Data Object Browser View

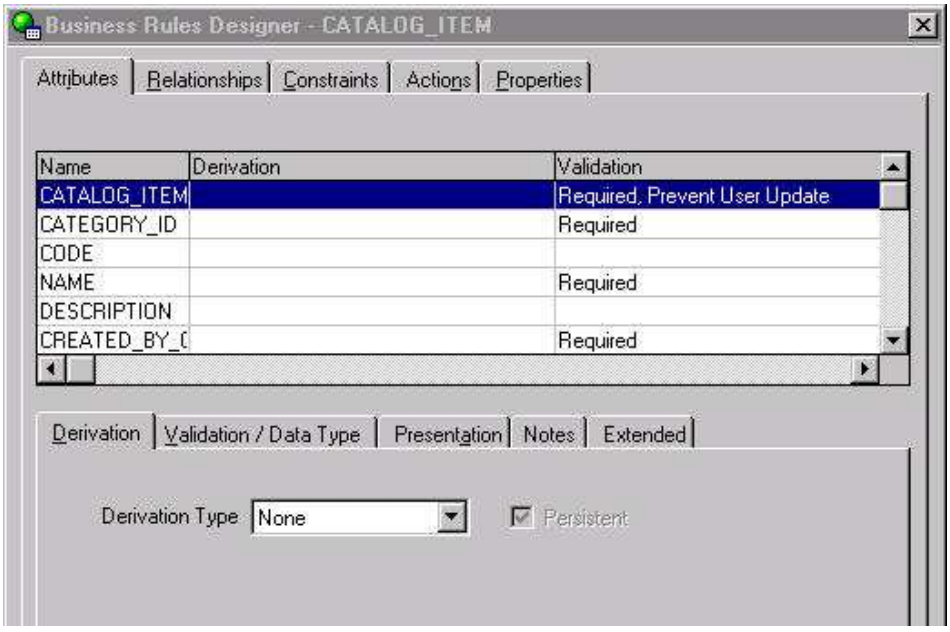

Figure 5-4 Versata Business Rules Designer

5.3 Module / Sub Module Descriptions

The ROSS application is comprised for four major functional modules: System Administration, Data Administration, Dispatch and Travel. Table 5-1 identifies the ROSS application modules their respective sub-modules. A functional description is provided for each sub-module. Each of the sub-modules is specified in Versata Studio.

Table 5-1 Application Modules and Sub-modules

System Administration	Data Administration	Dispatch	Travel
• Airport	• Catalog	• Initial Report	• Travel Plan
• Audit	• Resource Item	• Incident	• Travel Itinerary
• Compact	• Resource Status	• Resource Request	
• Hazard	• Roster	• Merged Incidents	
• Import	• Contract	• Complex Incidents	
• Organization		• New Request	
• Location		• Quick Fill	
• Person		• Place/Fill/UTF Request	
• Political Unit		• Place/Fill/UTF Supply	
• Reference Data		• Release/Reassign Request	
• Screens & Roles		• Complete/Cancel Request	
• Selection Area		• Multi-Edit Request	
• User Accounts		• Notifier	
		• Notifier Settings	
		• Reports	

5.3.1 System Administration Functions

The System Administration module maintains the application's physical resources. This module integrates key organization and user data tables necessary for the fundamental operation of the system, user access and permissions, and system interfaces.

5.3.1.1 Screens and Roles

Screens and Roles provides the mechanism for establishing application access security within the ROSS application. Screens are assigned to one or more Roles. One or more Roles are assigned to an Organization Security Type (National Dispatch, Geographic Area Dispatch, Local Dispatch, Agency, Cache, Other Government, Vendor, etc.). Users have one or more User Accounts with an Organization with a specific Organization Security Type.

5.3.1.2 User Accounts

User Accounts allows the ROSS User to create, edit and delete **Assigned Roles**, **User name** and **Password** information.

5.3.1.3 Organization

Organization creates records of interest to the dispatch community regarding federal and state cooperators, vendors, and jurisdictional organizations. The Organization screen provides the capability to build a hierarchy of organizations and geopolitical units, as well as an interface to attach persons to an organization. Additionally, it provides the means to establish the Benefiting, Dispatching and Billing Organizations for an Incident. Usually the dispatch office entering the record is the Host Organization; however, there are instances where the Benefiting Organization is private, foreign, or non-fire cooperator organization. In rare cases, dispatch responsibilities are passed to another dispatch office and sometimes an Incident is sponsored by a non-cooperator organization and the host organization wants the bills sent directly to the sponsor.

5.3.1.4 Person

The **Person** screen can be arrived at from the Organization or directly from the menu. Information regarding Persons are stored in the ROSS database with their associated Organization. This information is maintained even when individuals move from one location to another by utilizing Link and Unlink.

5.3.1.5 Airport

Airport data is imported from the Federal Aviation Administration (FAA) Airport database. The **FAA** data can not be edited nor deleted. **Airport** is used by ROSS Users only to provide additional information about an airport or for airports that are created locally.

5.3.1.6 Aviation Hazard

Aviation Hazard documents any air hazards or conditions that may affect the delivery or the performance of resources. The descriptions of hazards are reusable; however, if the hazard no longer exists in the environment it should be removed.

5.3.1.7 Location

Location contains three kinds of records. This includes; United States Geological Survey (USGS), Incident locations, and geographic locations as added by ROSS Users. ROSS users will not be able to delete the USGS records or locations entered by other offices. Only locations that do not have any associations can be deleted. Additionally, location provides a method for adding and updating location information. The information stored for a location includes name, declination, navigation instructions, state, geopolitical unit, record source, record status, and organization. The geographical coordinates can be entered as latitude/longitude, township/section/range, or UTM. A User can search for a location by name, type or location coordinates with a return up to 20 of the closest locations.

5.3.1.8 Selection Area

The purpose of **Selection Area** is to maintain a list of dispatch offices that comprise a cooperative within which resources may be exchanged directly via cooperative agreements and agency jurisdiction. These Selection Areas are used to designate the placement authority of a dispatch office, coordination center, or NICC, which defines the boundaries within which the office may operate directly with providers, without enlisting external dispatch channels. The Selection Area screen allows Data Administrators and System Administrators at the GACC and NICC level to edit Selection Areas for Dispatch Units.

5.3.1.9 Compact

A **Compact** is a formal working agreement among agencies to obtain mutual aid. Only Data and System Administrators at the GACC and NICC level may use a Compact.

5.3.1.10 Audit

Audit logs are security-relevant actions performed by logged-on users for later review by System Administrators. Actions such as logging on and off, opening and closing forms, and changes in data are recorded. System Administrators specify which User ID to track as well as a beginning and end date for the audit. System Administrators can search by any or all data items recorded.

5.3.1.11 Reference Data

Reference Data contains tables that are used by various screens throughout the ROSS application. Reference Data tables can not be deleted, however, the options or records within each table can be changed or deleted by the Data Administrator or through imported data files.

5.3.1.12 Political Unit

Political Unit is used to associate an organization with its political unit (e.g., city, state, nation, and province).

5.3.2 Data Administration Functions

Data Administration maintains the application's data resources. This module provides the key functionality to describe a resource catalog item; create, update, and delete a resource; update the availability of a resource and provide the status of a resource.

5.3.2.1 Catalog

The **Catalog** is used to organize the categories and catalog items for each of the catalogs: Aircraft, Crew, Equipment, Overhead and Supply. It is also used to create configurations (i.e., crews, teams, etc.) for a specific catalog item.

5.3.2.2 Contract

A **Contract** is an agreement between a Vendor and an Organization. An Vendor may be associated with multiple Contracts, but a Contract may only be with one Organization. Contracts can also include resources and / or services or both. Contract provides the capability to track and maintain an Organization's contracts, the resources offered and any features associated with the resources.

5.3.2.3 Resource Item

Resource Items are inventory records that must be tracked individually. The intention of Resource Item is to record the number of resources that a particular provider has available. A provider may provide more than one type of resource. Inventory items are categorized under Aircraft, Crew, Equipment, Overhead and Supply.

5.3.2.4 Roster

A **Roster** is an identified group that is permanently assigned to a resource. This group may be a crew roster, or a support group for aircraft or equipment.

5.3.2.5 Resource Status

Multiple requests can be edited together using **Multi Edit Request**. All requests to be edited must be selected from the table. Text boxes below the table left blank are **not** updated. Only information pertaining to the current incident appears.

5.3.3 Dispatch Functions

Dispatch performs the essential functions that locate and allocate equipment, supplies and personnel to support fire-fighting incidents. This module tracks each incident and its associated requests.

5.3.3.1 Initial Report

The **Initial Report** initiates a process to informally document emergency information that has not been validated as an event. These screens may be the initial entry point for information about an incident or possible incident. Enter as much information as is deemed necessary to document the dialog between the reporting party and the receiving dispatcher. Only initial reports that do not have associated resource requests may be deleted.

5.3.3.2 Incident

The **Incident** screen allows the dispatcher to manage an incident. Incidents can be created direct or they can be promoted from initial reports. The Incident screen provides information about the incident such as the location, associated radio frequencies, hazards, resource order line items, detail requests, related incidents, VORs, etc.

5.3.3.3 Incident Complex

Incident Complex is the process of associating multiple Incidents with a single Incident responsible for managing the associated Incidents. Each Incident maintains their resources and requests.

5.3.3.4 Incident Merge

Incident Merge is used to merge multiple incidents into a single incident. Resources may be automatically reassigned from the closing incident to the surviving incident. Open requests for the incident to be closed will be canceled or reordered. All incidents must already exist within ROSS prior to their selection for the merge operation

5.3.3.5 New Request

New Request is used to create a request. Once the request has been created other request screens are used to maintain the request information.

5.3.3.6 Quick Fill

Quick Fill is used to directly fill resources and to quickly request the most common resources. The only resources that can be used to Quick Fill are Aircraft, Crew, and Equipment. To verify that a resource can be filled utilizing Quick Fill, see Resource Item.

5.3.3.7 Place / Fill / UTF Request

The purpose of **Place / Fill / UTF Request** is to record the selection of a resource to fill a resource request (exception is Supply) for the current organization, another organization, or to advise return the Placed Request with notification that the Request could not be filled. The resource is attached to the request; however, actual commitment of the resource does not occur until the resource is mobilized.

The More Information menu leads the user to additional details provided for the incident that does not fit on this screen. The subsequent dialog boxes contain a variety of Incident related records on associated organizations, financial, delivery locations, special needs, related and supported requests, and documentation.

5.3.3.8 Place / Fill / UTF Supply Request

The purpose of **Place / Fill / UTF Supply Request** is to record the selection of supply resources to fill a resource requests for the current organization, another organization, or to advise return the Placed Request with notification that the Request could not be filled. Supplies are automatically assigned without an interim selected status.

The More Information menu leads the user to additional details provided for the incident that does not fit on this screen. The subsequent dialog boxes contain a variety of Incident related records on associated organizations, financial, delivery locations, special needs, related and supported requests, and documentation.

5.3.3.9 Release / Reassign Request

The **Release / Reassign Request** screen handles the release of resources currently located at an Incident. Additionally, if another Incident has existing open or pending requests, the released resources may be reassigned singly or in multiple quantities.

5.3.3.10 Request Status

Request Status is used for the expedient editing and display of the request statuses of owned by the currently logged-on organization.

5.3.3.11 Complete / Cancel Request

Complete Request is used either complete or cancel a request, once the resources have been demobilized and returned to their home locations or been assigned to other Incidents. Canceled Requests may only be canceled by the original requesting organization. Documentation regarding the nature of the cancellation is required.

5.3.3.12 Multi-Edit Request

Multiple requests can be edited at one time using **Multi Edit Request**. All requests to be edited must be selected from the table. Text boxes below the table left blank are not updated. Only information pertaining to the current incident appears.

5.3.3.13 Supplemental

Supplemental supports requests for three unique services: Food Service, Infrared Aircraft Scanner and Temporary Flight Restrictions. These requests record the information regarding the details to support the request.

5.3.3.14 Notifier

The **Notifier** provides automatic notification and messages to a user as specified by the user's Notifier Settings. The Notifier Settings determine which request or incident actions send a message, a message and a notification, or no message at all.

5.3.3.15 Notifier Settings

Notifier Settings is where the user configures the notifier messages and actions. The User's preference includes whether a record is active or passive, and whether the User wants to receive messages for Aircraft, Crew, Equipment, Overhead and Supplies entries.

5.3.4 Travel Functions

Travel Itinerary deploys personnel assigned to fire-fighting incidents. This module coordinates travel plans with travel requirements and funding codes for resources allocated to specific incidents. It is also used to develop the vehicle transport route making up the travel plan.

5.3.4.1 Travel Plan

Travel Plan sets up the travel plan for transport vehicles (e.g., as airplanes, buses, etc.) traveling to and from assignments. Transport vehicles can also take on Request passengers or Administrative passengers.

5.3.4.2 Travel Itinerary

Travel Itinerary is for personnel resources and is limited to a single mode of travel; therefore, if a resource is traveling via airplane and ground vehicle, two travel itineraries are required. Many resources assigned to different Incidents may be attached to the same travel leg. If the traveling resource makes multiple stops en route to its destination, a travel leg record must be created for each stop. This provides information regarding the delivery of resources to all specified destinations and is not intended to provide a mechanism for tracking or following the transport mission.

5.4 External Interfaces

The system's interfaces(s) to the outside world are illustrated below in Figure 5-5. Appendix C provides draft interface control documents (ICDs) to describe the interface to the external data systems.

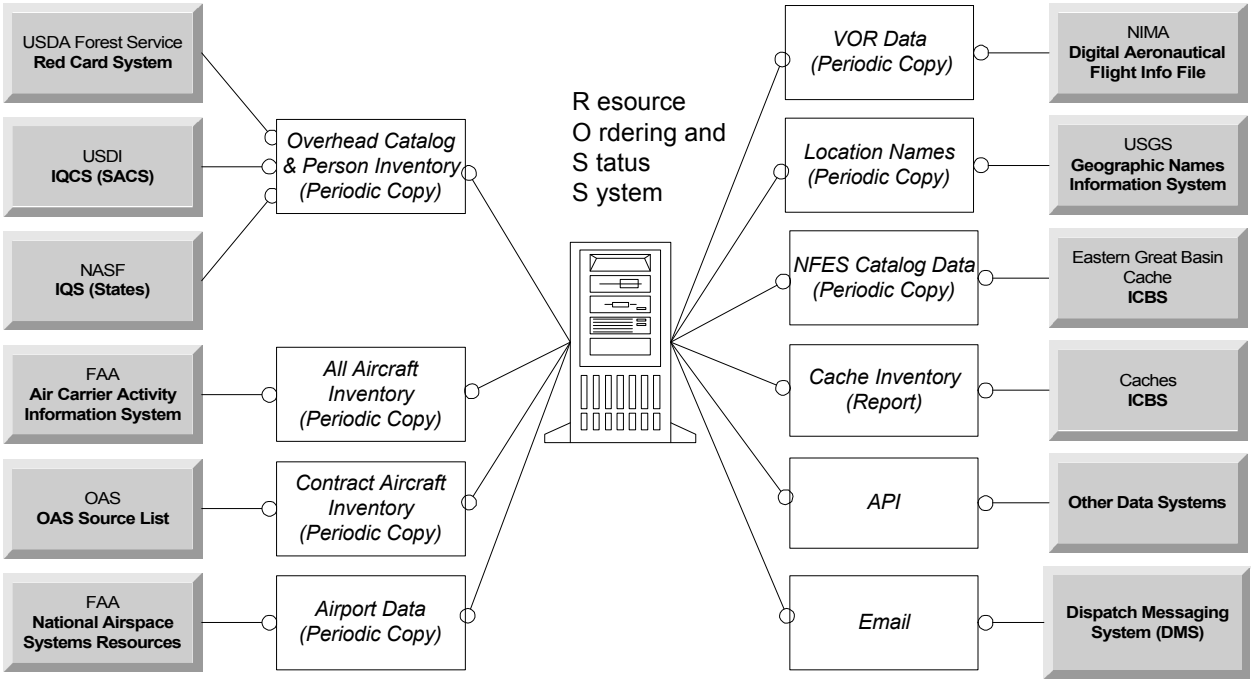

Figure 5-5 System Interface Diagram

6. Security and Access Methodology

The critical nature of the ROSS operational environment, and the need to protect Agency sensitive and personnel sensitive information from unauthorized access, requires that the ROSS application and underlying technical infrastructure support appropriately high assurance levels with respect to both operational availability and information integrity. This section describes the access methodology and security employed for the ROSS Wide Area Network (WAN) connectivity, ROSS server Local Area network (LAN) connectivity and ROSS application in support of the ROSS application.

6.1 Summary of Approach

The ROSS access methodology relies upon three levels of access control:

- Network level IP address filtering;
- Application/proxy (i.e., firewall) filtering; and,
- ROSS application level access controls, consisting of:
  - logi.crypto Java Package, version 1.1.1 encryption of userID/password logon sessions;
  - UserID/password authentication with strictly enforced password criteria; and,
  - Role based access control based upon authenticated UserID to:

    - ROSS system level controls;
- ROSS application components;
- ROSS forms;
- Query/method objects; and,
- Data fields.

The ROSS application will be hosted on servers supported by the U.S. Department of Agriculture's, National Information Technology Center (NITC) in Kansas City, Missouri. These servers, addressed further in Section 3.0, are the same hardware platforms currently used to support the NWCG Dispatch Messaging System (DMS), an email system used for the transmission of critical information between dispatch offices. The NITC will have the principal responsibility for maintaining the ROSS infrastructure environment, including all network routers, firewalls, ROSS servers and RAID non-volatile disk arrays. The need for IP address filtering and the manner in which it will be implemented is an issue for coordination between the ROSS contracting officer and the NITC. Implementation of a firewall level application/proxy and the degree of filtering/monitoring to be implemented is also an issue open to negotiation between the ROSS contracting officer and the NITC. The ROSS application design and the COTS software components networking requirements will need to be taken into account when implementing a firewall/network security strategy.

ROSS application level access controls, however, are entirely implemented within the ROSS application and supporting COTS software infrastructure. These controls are addressed by this document in detail in Section 6.4.

6.2 ROSS WAN Connectivity

As indicated in Figure 6-1, the NITC facility is part of a U.S. Department of Agriculture Intranet comprised of three additional data centers located in Albany, California, Fort Collins, Missouri, and Washington, D.C. Each of these three remote USDA data centers provides T1 or greater access to the commercial Internet. The data center in Washington, D.C., provides access to the Forest Service's own Intranet.

The ROSS is a web-enabled application which relies upon commercial internet connectivity as the primary means of connectivity between the over 400 federal, state, municipal, and local agency wildland incident dispatch offices with the ROSS application servers. As indicated in Figure 6-1, ROSS users will have three access paths, via the Internet, to the ROSS application servers.

Pending final decision by the NITC and the ROSS Contracting officer, any site which loses Internet connectivity may be provided the option of obtaining emergency connectivity to the ROSS by modem via a ROSS Remote Access Server (RAS). Also pending final decision by the NITC and the ROSS Contracting officer, is whether the ROSS will implement a ROSS specific application proxy firewall. Please note that the actual location of the ROSS firewall and the ROSS RAS server (if provided) will be determined in accordance with NITC approved policy, and may not be configured as indicated in Figure 6-1, below.

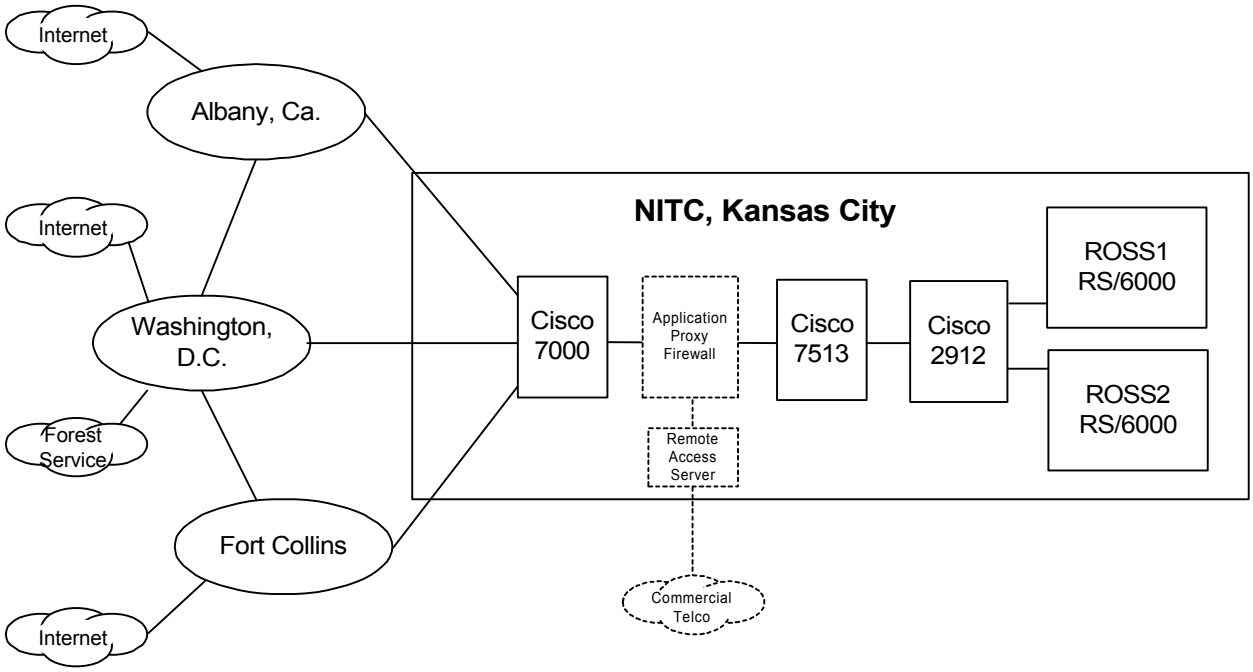

Figure 6-1 ROSS Wide Area Network (WAN) Connectivity

6.3 ROSS LAN Connectivity

As discussed in Section 6.1, the ROSS application will be hosted upon server platforms currently used to support the NWCG Dispatch Messaging Service (DMS). The USDA has selected two RISC System/6000 Model F50 servers (Figure 3-1) as their cluster nodes. The two servers are identical in hardware and software configuration. The nodes are running AIX 4.3.3 and HACMP v4.3 (with all fixes). The servers consist of three internal SCSI disks, two 9.5 GB and one 4.5 GB, containing operating system and paging space. Both servers have two 10/100 Mb Ethernet cards for connectivity to the local LAN. The servers use a third serial port on each server to provide a serial network connection used by HACMP. The two servers are jointly connected to a SSA160 RAID Array disk subsystem through an SSA RAID Adapter. The disk array is configured as RAID 5.

6.4 ROSS Application Level Access Controls

The ROSS system provides access control security over the following entities:

- Application
- Screens
- Data

UserIDs and Passwords restrict access to the application.

A user with the ROSS user role of Account Manager provides user's with their userIDs and passwords. Each time a user attempts to access the ROSS application, the ROSS application checks to see if the user is authorized to access the application.

A ROSS user performs a login to the ROSS application by entering a User ID and Password into a web browser application on the workstation that encrypts the User ID and Password. Once the web browser application connects to the ROSS application on the server at NITC, the encrypted User ID and Password is compared with the User ID and encrypted password contained in the ROSS database.

If the login is successful, the user is able to change the User ID and Password through the User Settings screen. When the application is deployed, the applications to set and change User IDs and Passwords will be programmed to accept entries in accordance with policy for the ROSS application.

Encryption of the userID/Password logon is completed via logi.crypto, an encryption scheme that is utilized within the java application. Logi.crypto is a non-certified 100% pure java library for using strong encryption in Java 1.1 programs. It includes tools for encryption and authentication, and a framework for general cryptographic protocols.

Screen-Role permissions restrict access to each screen and to the data. Screen-Role permissions are base on a combination of the role (e.g., dispatcher, dispatch manager, system administrator, data administrator, etc.,) and the organization level, such as National Interagency Coordination Center (NICC), geographical area coordination center (GACC), local dispatch, national cache, etc.

To meet the requirement for ease of use with the dispatch environment, the ROSS application allows switching between users once a session has been established between the web browser and the ROSS server. The user session is controlled by Screen-Roles assigned to the current UserID for the session. Screen-Roles not assigned to the newly selected user are automatically closed by the application when the change is users occurs.

7. Glossary of Terms, Acronyms & Abbreviations

Applet – **A** small program that can be sent along with a Web page to a user. Java applets can perform interactive animations, immediate calculations, or other simple tasks without having to send a user request back to the server.

Client – The requesting program or user in a distributed architecture. For example, the user of a Web browser is effectively making client requests for pages from servers all over the Web. The browser itself is a client in its relationship with the computer that is getting and returning the requested HTML file. The computer handling the request and sending back the HTML file is a server.

RAD – Rapid Application Development. Term to denote tools that allow applications to be rapidly prototyped and developed using a graphical interface with predefined behaviors and abilities.

Server – In the client/server programming model, a server is a program that awaits and fulfills requests from client programs in the same or other computers. A given application in a computer may function as a *client* with requests for services from other programs and a *server* of requests from other programs.

Thin Client – A simple client program or hardware device which relies on most of the function of the system being in the server. With thin clients, changes can be made to the client and then broadcast to the remote sites, eliminating the need for onsite administrators.

AIX	IBM's version of UNIX, normally found on the RS/6000
ANSI	American National Standards Institute
API	Application Programming Interface
ATB	Air Tanker Base, and airport facility used to reload and launch fire fighting tanker aircraft in CANWin
BPM	Business Process Model
CANWin	COMPUTER-AIDED NAVIGATION (CAN) FOR WINDOWS
CD	Compact Disk
CDM	Conceptual Data Model, in the ROSS environment, the Silverrun ERX model
CIO	Chief Information Officer
COOP	Continuity of Operations
CRUD Matrix	Data Objects Cross Reference (CRUD Matrix) where CRUD stands for Create, Read, Update, Delete
CS	Computer Services, an organization which provides computer support
CSM	Component Security Manager
CSS	Cascading Style Sheet language
DAA	Designated Approving Authority
DEC Alpha	COMPAQ/Digital Equipment Corporation's very high end work station
Digital UNIX	COMPAQ/Digital Equipment Corporation's version of UNIX, (previously called OSF-1)
DLL	Dynamic Link Library
DLT	Digital Linear Tape
DMS	Dispatch Messaging System
DMZ	De-Militarized Zone
DR	Disaster Recovery
DTD’s	Document Type Definitions
FAA	Federal Aviation Administration
GACC	Geographical Area Coordination Center, one of 11 second tier command centers
GB	Gigabyte, term of measure for data storage reffering to 1,073,741,824
GIS	Geographic Information System
GUI	Graphical User Interface
HAZ	Hazard, referring to the location of a known hazard (CANWin)
HELI,	Helicopter, indicative of a helicopter base in CANWin
HP	Hewlett Packard, a major vendor in the computing indusry
HP-UX	Hewlett Packard's version of UNIX
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol

HTTPS	Hypertext Transfer Protocol Secure
IBM	International Business Macines,a major vendor in the computing industry
INFOSEC	Information Security
IRIX	Silicon Graphics version of UNIX
IIOP	Internet Inter-Orb Protocol
ISO	The International Standards Organization
ISSO	Information Systems Security Officer
JDBC	Java Database Connectivity
Jserv	Servlet Engine for Java
JVM	Java Virtual Machine
K	K, same as KB = Kilobyte term of measure for data storage reffering to 1024
Lat/Lon	An abbreviation for Latitude / Longitude, representing a standard Mercator location
Legal	The official term for Township, Section, and Range geographic location
MB	MegaByte+C311,048,576
MIRPS	Multi-Agency Incident Resource Processing System
NICC	National Incident Command Center
NITC	National Information Technology Center
NWCG	The National Wildfire Coordination Group
OCL	Object Class Library
OS/2.	IBM's Operating System for Intel based PC's
OSM	Office Security Manager
PDF	Portable Document Format (Adobe)
PDM	Physical Data Model
PKI	Public Key Infrastructure
proxy server	A server that sits between a client application, such as a Web browser, and a real server.
PS	Postscript, A file format widely used by various vendors of printer
RAID	Redundant Array of Independent Disk
RAM	Random Access Memory
RAS	Remote Access Server
ROSS	The Resource Ordering and Status System
RS/6000	IBM's RISC based mid-tier computing solution, normally runs under AIX
RTM	Cross reference matrix of requirements to software components and data objects [requirements traceability matrix(RTM)]
SBU	Sensitive But Unclassified, generally refers to privacy related data
SCSI	Small Computer System Interface

SDD	System Design Document
Servlets	An applet that runs on a server
SGML	ISO's Standard Generalized Markup Language
Solaris	Sun's commercially available variant of UNIX
SPARC	Sun's RISC architecture for mid to high end work stations
SSL	Secure Socket Layer
Sun OS	Sun's proprietary variant of UNIX
TASO	Terminal Area Security Officer
TSEC	Trusted Computer System Evaluation Criteria
UNIX	A popular multi-user, multitasking operating system developed at Bell Labs in the early 1970s
USGS	United States Geological Survey
UTF	Unable to fill
UTM	Universal Transverse Mercator, a method of geographic location that accounts for the planet's shape as an oblate spheroid
VGA	A standard for video rendition in the computing industry, Video Graphics Adapter, is named for the PC component required to render graphics at the level 800 x 600 pixels.
VLS	Versata Logic Server
VOR	aViation Omni-Radio, a radio direction finding beacon, used to locate aircraft in US airspace.
VOR Ref	VOR Reference, a method of location depending on a known point (the aViation Omni Radio beacon), a direction from the point and a distance from the point.
W3C	World Wide Web Consortium
Windows 95	A Microsoft Operating System based on the Window paradigm
Windows NT	A Microsoft Operating System based on the Window paradigm, NT = New Technology, a larger business oriented version
WYSIWYG	Acronym for phrase, "What You See Is What You Get"
XML	eXtensible Mark-up Language