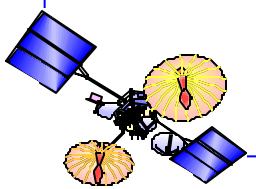
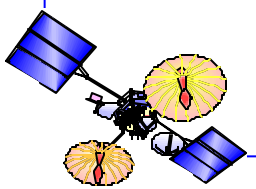
Space Network (SN) Web Services Interface (SWSI) Requirements/Design Review

October 19, 2000



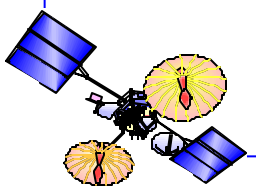
Introduction

Tom Sardella
Code 583/450



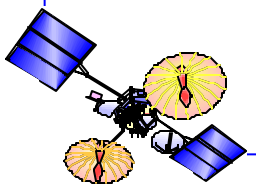
Agenda

Introduction	Tom Sardella
Requirements	Tom Sardella
Design Overview	Harshna Sampat
Hardware	Tom Sardella
Security	Joe Stevens
Client Design	Geri Klitsch
Break	
Application Server Design	Geri Klitsch
Isolator Design	Maurice Assaraf
SNIF Design	Tom Sardella
Database Design	Harshna Sampat
Summary	Tom Sardella



SWSI Overview

- **Java-based platform independent customer interface to NCCDS for performing TDRSS scheduling and real-time service monitoring and control**
- **Secure Socket Layer (SSL) encrypted interface from Closed IONET, Open IONET, and Internet**
- **TDRSS Unscheduled Time (TUT) access from Open IONET and Internet**
  - **Current access to NCCDS TUT is from Closed IONET only**
- **Customer workstation requirements**
  - **Web browser**
  - **Java Virtual Machine 1.2**
    - **Currently supported on Win 95/98/NT, Unix**
    - **Future support on MacX, HP-UX**



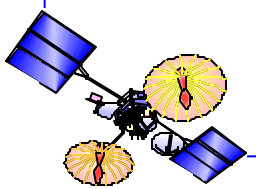
SWSI Overview (Cont'd)

SWSI Customers

- Long Duration Balloon (LDB)
- Ultra Long Duration Balloon (ULDB)
- Gravity Probe B (GP-B) (May 02 Launch)
- Seismic Star
- Swift

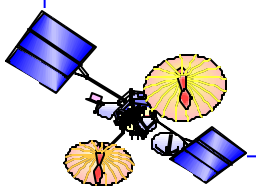
Demand Access System (DAS) customer interface

- Requirements analysis and design still in progress
- Will be addressed in Delta DR



SWSI Prototype

- **In-house prototyping effort initiated in February 1997 to explore feasibility of performing web-based SN customer scheduling and service monitoring and control**
  - **HTML-based user interface enhanced with Java and Javascript**
  - **CGI forms allow user to enter and view NCCDS messages**
- **SSL encrypted interface**
- **2-tier proxy architecture allows access from Closed IONET, Open IONET, and Internet**
- **Currently supporting short duration (2 week) Long Duration Balloon (LDB) missions**
- **Limitations**
  - **User interface is NCCDS/MOC ICD message-based**
  - **No automated tracking of "Active Schedule"**
  - **HTML not well suited to building custom GUI**



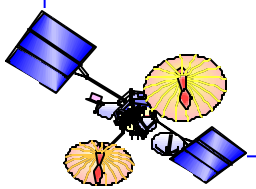
Purpose of Review

Review SWSI requirements & design

Email comments to:

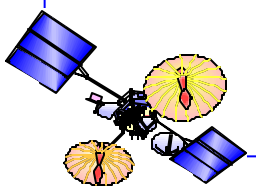
swsirdr@msp.gsfc.nasa.gov

Please submit comments by COB 11/3/00



Schedule

ID	Task Name	Start	Finish
1	**Project Start**	**Sat 4/15/00**	**Sat 4/15/00**
2			
3	NCCDS I/F Detailed Design	Mon 4/17/00	Fri 10/13/00
4	**Requirements/Design Review**	Fri 10/13/00	Fri 10/13/00
5			
6	DAS I/F Detailed Design	Mon 9/25/00	Thu 2/15/01
7	**Delta Design Review**	Thu 2/15/01	Thu 2/15/01
8			
9	Software Development	Mon 10/16/00	Fri 11/2/01
10	**NCCDS Test Readiness Review**	Fri 11/2/01	Fri 11/2/01
11			
12	DAS I/F Acceptance Test	Mon 11/5/01	Mon 12/31/01
13			
14	NCCDS I/F Acceptance Test	Mon 11/5/01	Mon 12/31/01
15			
16	**Operational Readiness Review**	Tue 1/15/02	Tue 1/15/02
17	**Transition to Operations**	Thu 1/31/02	Thu 1/31/02



Documentation

Awaiting approval

SWSI Product Management Plan, 453-PMP-SWSI

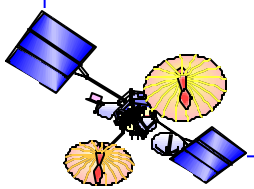
Draft for review

SWSI System Design Specification, 453-SDS-SWSI

SWSI System Requirements, 453-SRD-SWSI

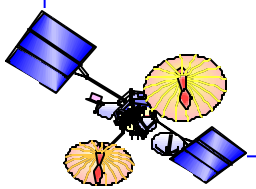
SWSI web page with online documents is under development

http://msp.gsfc.nasa.gov/swsi

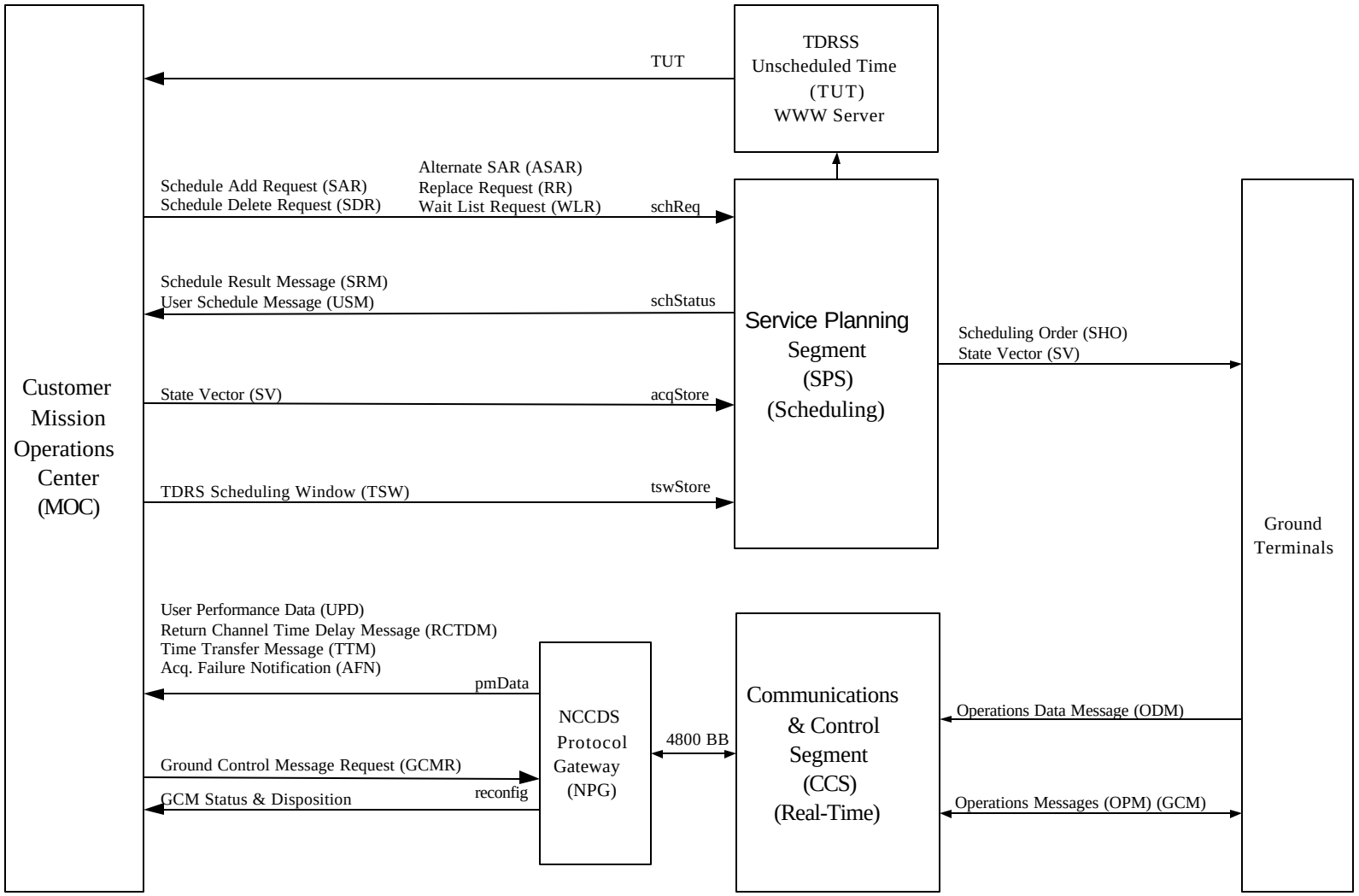


Requirements

Tom Sardella
Code 583/450

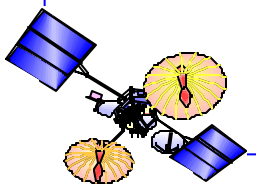


NCCDS-MOC Interface



Requirements

- **Customer Web interface to perform all NCCDS customer functions (full support only)**
  - **SWSI implements NCCDS/MOC message interface on behalf of customers**
  - **All full support messages supported, including newer flexible scheduling messages**
  - **Standards-based cross-platform**
  - **Ease of use in manual mission operations**
  - **Multi-mission support**
  - **"Normal" services only (no Shuttle-unique services)**
- **Access to both Operational NCCDS and Auxiliary NCC (ANCC) for Engineering Interface (EIF) testing**
- **Secure interface from Closed IONET and open networks (Open IONET and Internet)**



Requirements (Cont'd)

Adhere to existing NCCDS RMA requirements

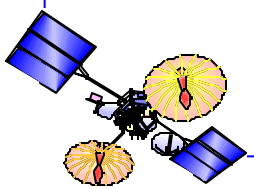
- 2500 hours mean time between failures (MTBF)
- 30 minutes mean time to repair (MTTR)
- 0.9998 availability
- Redundant servers in High Availability configuration

Customer Scheduling

- Customer submission of schedule requests (SAR, Alt SAR, etc.)
- Database of customer Service Specification Codes (SSCs) with default parameter settings copied from NCCDS database
- Schedule Requests display shows previously submitted requests
- Maintain Active Schedule based on SRMs, USMs returned from NCCDS
- Active Schedule display

Vector Storage

- IIRVs from customer-generated file transmitted to NCCDS
- Conversion of latitude, longitude, & altitude to Type 8 (Stationary) vector



Requirements (Cont'd)

TSW Storage

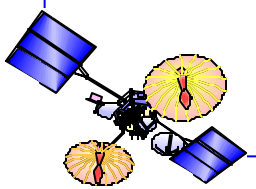
- TSWs from customer-generated file transmitted to NCCDS

Performance Data

- UPDs formatted into user-friendly displays
- Default displays for each service with facility for customer to generate own custom displays
- Limit checking of selected parameters for Warning & Out of Tolerance severity levels
- RCTD and TTM returned to customer as binary files
- Acquisition Failure Notification sent to customer as an alert

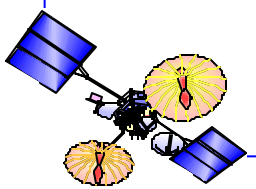
Ground Control

- Customer submission of GCMRs
- USMs, GCMRs, and GCM Status used to track current parameter settings
- GCM Status & Disposition sent to customer as alerts



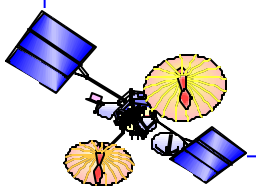
Requirements (Cont'd)

- **Access to TUT from Open IONET (current access is from Closed IONET only)**
- **Message and event logging**
  - **Log all messages exchanged with NCCDS**
  - **Provide delogging capability**

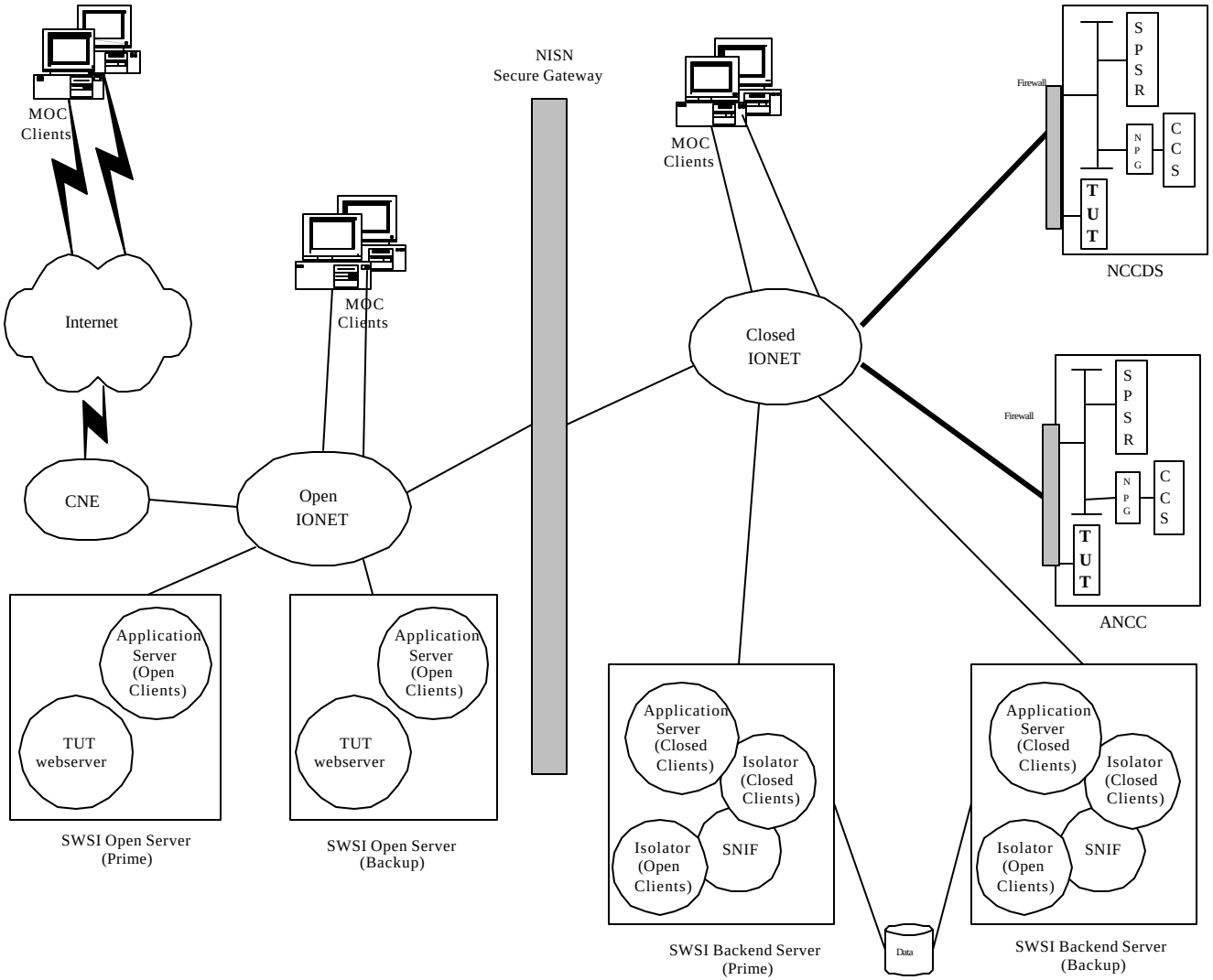


Design Overview

Harshna Sampat
CSC



SWSI Architecture



SWSI Architecture

Adapted from Java-Based Spacecraft Web Interface to Telemetry & Command Handling (Jswitch) architecture

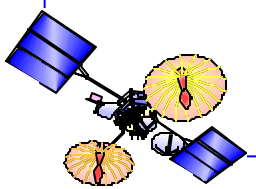
- Jswitch is used by MOC (XTE) to securely monitor spacecraft health and safety from a remote location

SWSI components

- Client
- Open Server
- Backend Server

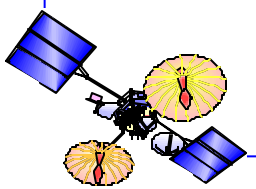
SWSI subsystems

- Client
- Application Server
- Isolator
- SNIF
- RDBMS Database
- Open TUT Server



SWSI Architecture (Cont'd)

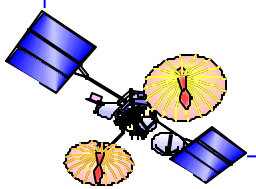
- Redundant Servers for high server availability
- Shared RAID Array for high data reliability
- Scalable Architecture to easily expand user base and SICs supported
- Client and Server Digital Certificates for strong authentication
- Data Encryption with SSL3 protocol for data confidentiality/ privacy and data integrity
- Security Tools: E.g. TcpWrapper, PortSentry, SecureShell
- Webserver for TUT Server and SWSI documents



SWSI Subsystems

Client

- Thin Java application
- User Interface
  - Schedule SN services
  - Reconfigure scheduled services
  - Monitor scheduled services
  - Monitor alerts
  - Monitor user performance data
  - Monitor connection status
- Logs Alerts
- Stores Return Channel Time Delay (RCTDM) and Time Transfer Messages(TTM) to the Client Disk
- Reads TDRS Scheduling Window (TSW) and State Vectors (SV) from the Client Disk



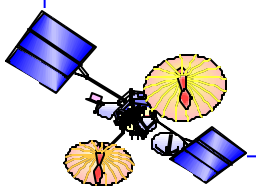
SWSI Subsystems (Cont'd)

Application Server

- Mid-tier Java application
- Proxy server
- Manages user requests
- One secure socket connection (SSL) with clients
- Three secure socket connections (SSL) with Isolator via NISN Secure Gateway

Isolator

- Back end Java application
- SWSI data manager (Oracle 8i database)
- Translates Java objects to UDP messages and vice versa
- Translates Java objects to SQL directives and vice versa
- Three secure socket connections (SSL) with Application Server
- Uses User Data-gram Protocol (UDP) to communicate with SNIF subsystem
- Stores TSWs and SVs files on the the Backend Server Disk
- Reads TCTDMs and TTMs files from the Backend Server Disk
- Logs System Error Messages



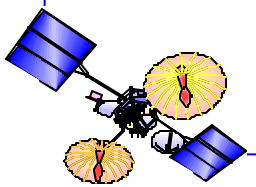
SWSI Subsystems (Cont'd)

SWSI-NCCDS Interface (SNIF)

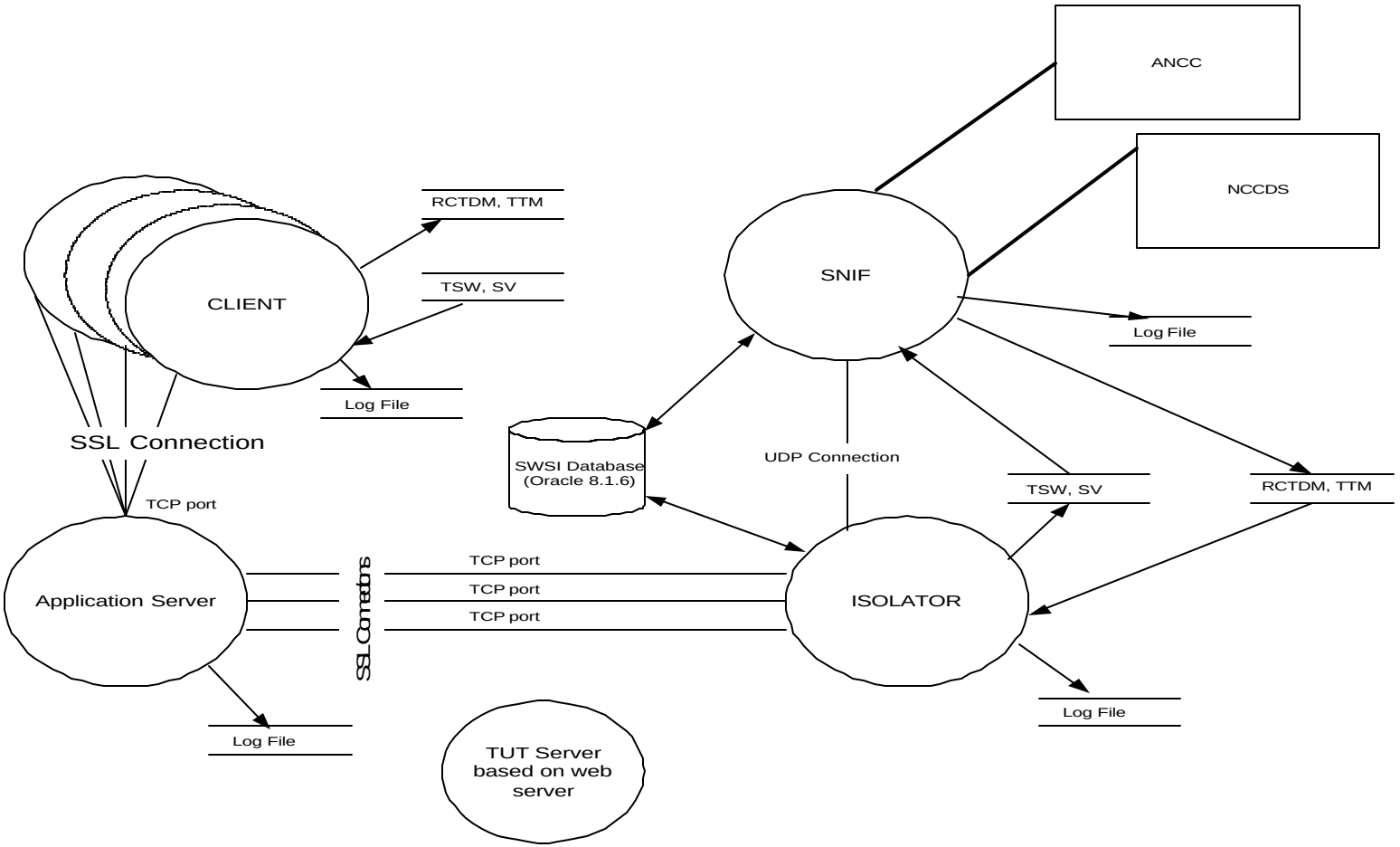
- Back end "C" application
- Interface to NCCDS and ANCC
- Establishes and maintains appropriate TCP connections with NCCDS/ANCC for each SWSI customer
- Implements message interface as defined in NCCDS/MOC ICD
- Maintains Active Schedule based on SRM & USM responses from NCCDS
- Stores RCTDMs and TTMs as files on the Backend Server Disk upon receipt from NCCDS /ANCC
- Reads TSWs and SVs files from the Backend Server Disk for transmission to NCCDS/ANCC

Open TUT Server

- Provides TDRS Unscheduled Time information to the users on the Open IONet and Internet

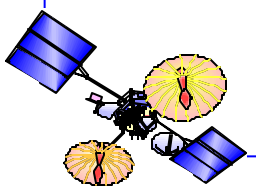


SWSI High Level Data Flow



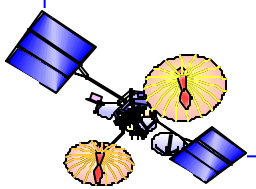
COTS/GOTS

- Solaris 7 Operating System
- Sun Professional Developer Suite
- GNU Development Tools
  - GCC 2.95.2
  - GDB 4.18
  - Data Display Debugger (DDD) 3.1.3
- Jbuilder Professional 3.5
- Oracle 8i Enterprise Edition Server 8.1.6
- Oracle Pro*C 8.1.6
- Java 2 Standard Edition 1.2.2 (free)



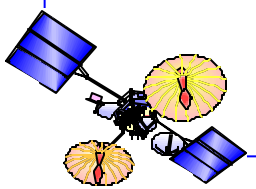
COTS/GOTS (Cont'd)

- HotSpot 1.0.1 (free)
- InfoBus 1.2 (free)
- Phaos SSLava™ Toolkit 1.11 for initial Builds
- Phaos J/CA Toolkit for digital certificate generation for initial Builds
- Oracle supplied JDBC Thin Driver
- Webserver (Apache or Netscape Enterprise)
- Security Tools
  - TcpWrapper 7.6, PortSentry 1.0, SecureShell 1.0
- GOTS
  - High Availability (HA) Tool, NPG DeLogger



Hardware

Tom Sardella
Code 583/450

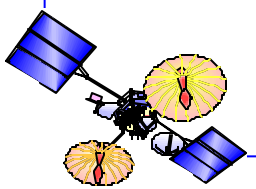


Server Hardware

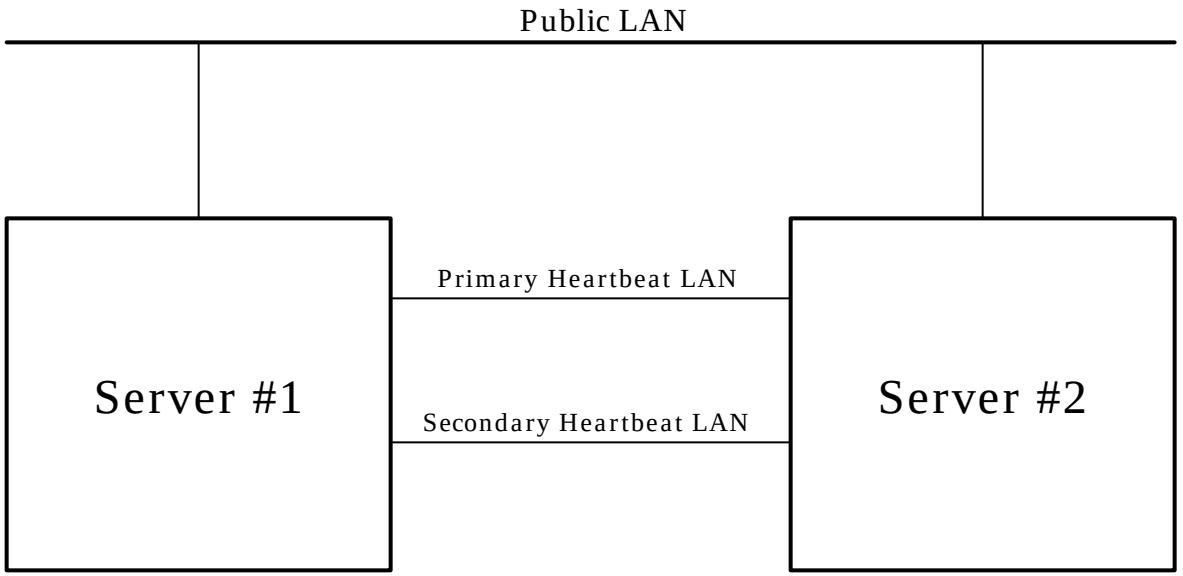
Sun Ultra 2 Desktop Workstations

- 300 MHz UltraSPARC-II CPU
- 9 GB SCSI Disk
- 4mm DDS-3 Tape Backup
- RAID disk for backend server database storage
- Quad Fast Ethernet provides additional interfaces for High Availability heartbeat

Final hardware configuration may change, depending on additional capability for DAS

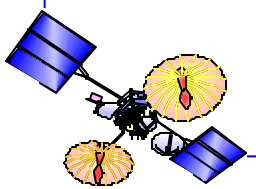


High Availability Configuration



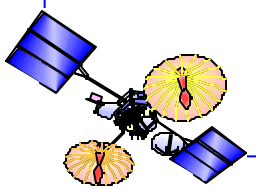
High Availability/Redundancy

- Custom HA application developed for NCC98 Sun-based subsystems
  - NCCDS Protocol Gateway (NPG)
  - Firewall
  - FTP/TUT Servers
- Automatic switchover to backup system upon failure of primary
- Shared IP address(es) allow server to appear as single address to client application
- Separate heartbeat interface(s) between systems coordinate transitions to ensure only one primary



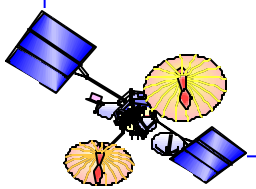
High Availability (Cont'd)

- System failover occurs on application or system failure
  - Application failure only after pre-configured number of application restarts
- TCP connections would have to be re-established after a switchover
- Database on shared RAID disk



Security

Joe Stevens
Code 566/450



Security Requirements

NASA Procedures and Guidelines (NPG) 2810.1, Security of Information Technology

Security Plan for the Network Control Center, NCC 98, 451-SP-NCC/1998

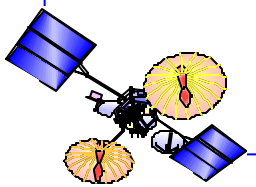
IP Operational Network (IONet) Security Plan, 290-003, September 1999

***Security Plan for Space Network Web Services Interface*, 452-SP-SWSI, May 10, 2000**

The Space Network (SN) is considered a Mission (MSN) critical resource

SWSI is an extension of the existing SN services

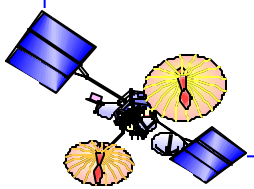
SWSI is considered a MSN resource



Security Model

The SWSI COTS SSL implementation provides:

- **Protocol**
  - SSL v.3 over TCP provides strong data integrity.
  - Between Client and Application Server; Application Server and Isolator
- **Authentication**
  - Signed Digital Certificates from Certificate Authority (CA)
  - 2-Way authentication for Client, Application Server, and Isolator
  - Isolator validates ALL Client requests.
- **Level of Security**
  - Secure session key exchange
  - Prevents session high-jacking and replay attacks
- **Implementation**
  - Initial Build(s) use(s) Phaos Toolkit -- SWSI Project generate certificates
  - Final Release uses Entrust Toolkit -- NASA to generate certificates
  - Supports NASA's Public Key Infrastructure (PKI) Initiative



Security Features

Flexibility

- Security is implemented at application level
- Supports numerous cryptographic algorithms (Triple-DES)

Portability

- Written in Java
- Client and Application Server can reside on different platforms

Enforces Client password attributes

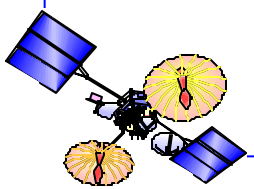
- Length, Content, Aging

Restricted number of failed attempts

Management of password via table in database

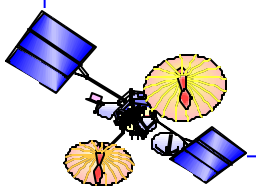
Audit Files

- Logins
- Host accesses
- Network activities



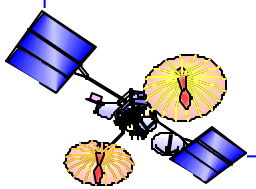
System Level Security

- All unused network services disabled
- Latest OS Security Patches installed
- Periodic vulnerability scans performed
- Periodic monitoring of all user accounts
- Local/Remote access monitored
- Periodic monitoring of audit logs
- IP Filtering of remote accesses
- Weekly Incremental Backup/Monthly Backups



Client Design

Geri Klitsch
CSC



Client Functionality

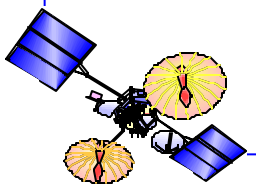
Establish SSL connection and login to Application Server

Generate and Send:

- **Schedule Add Requests (SARs)**
- **Alternate Schedule Add Requests (ASARs)**
- **Schedule Delete Requests (SDRs)**
- **Wait List Requests (WLRs)**
- **Replace Requests (RRs)**
- **Ground Control Message Requests (GCMRs)**
- **State Vectors (SVs) Type 8 vectors only**

Monitor:

- **Alerts**
- **Schedule Requests and details**
- **Active Schedule and details**
- **User Performance Data (UPDs)**



Client Functionality (Cont'd)

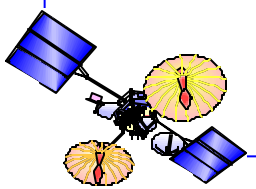
Send and Receive Data from and to files

- Send (User selects file to send from file chooser):
  - TDRS Scheduling Windows (TSWs) from file
  - State Vectors from file
- Receive and Store (done automatically by client):
  - Return Channel Time Delay Messages (RCTDMs) to file
  - Time Transfer Messages (TTMs) to file

Provide Online Help

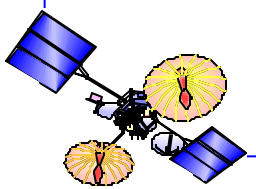
Panels mimic the NCCDS operator interface

- Driven by the same functionality
- Panels shown are prototypes (data values shown are samples and may not be valid)

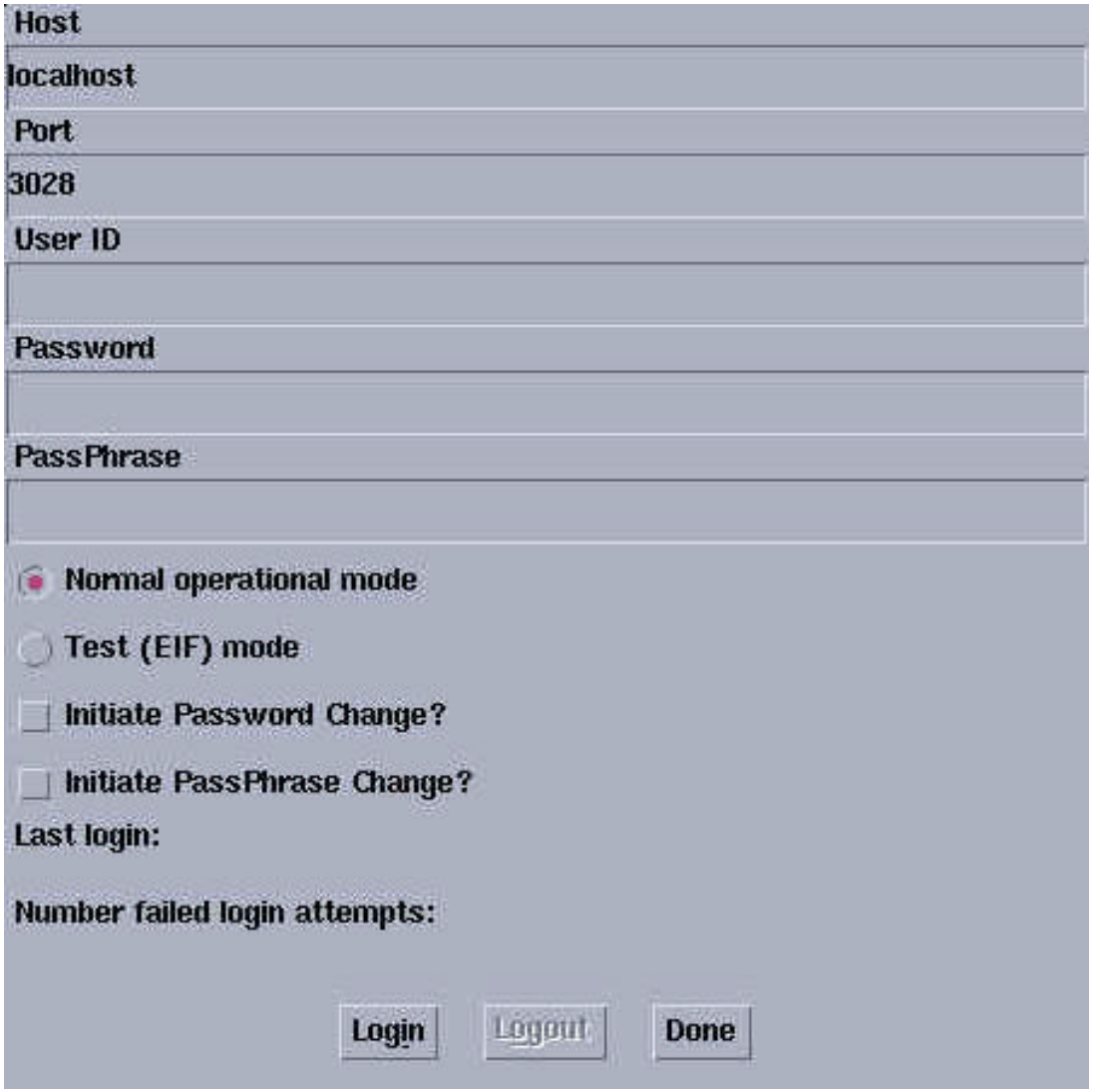


Main Control Panel

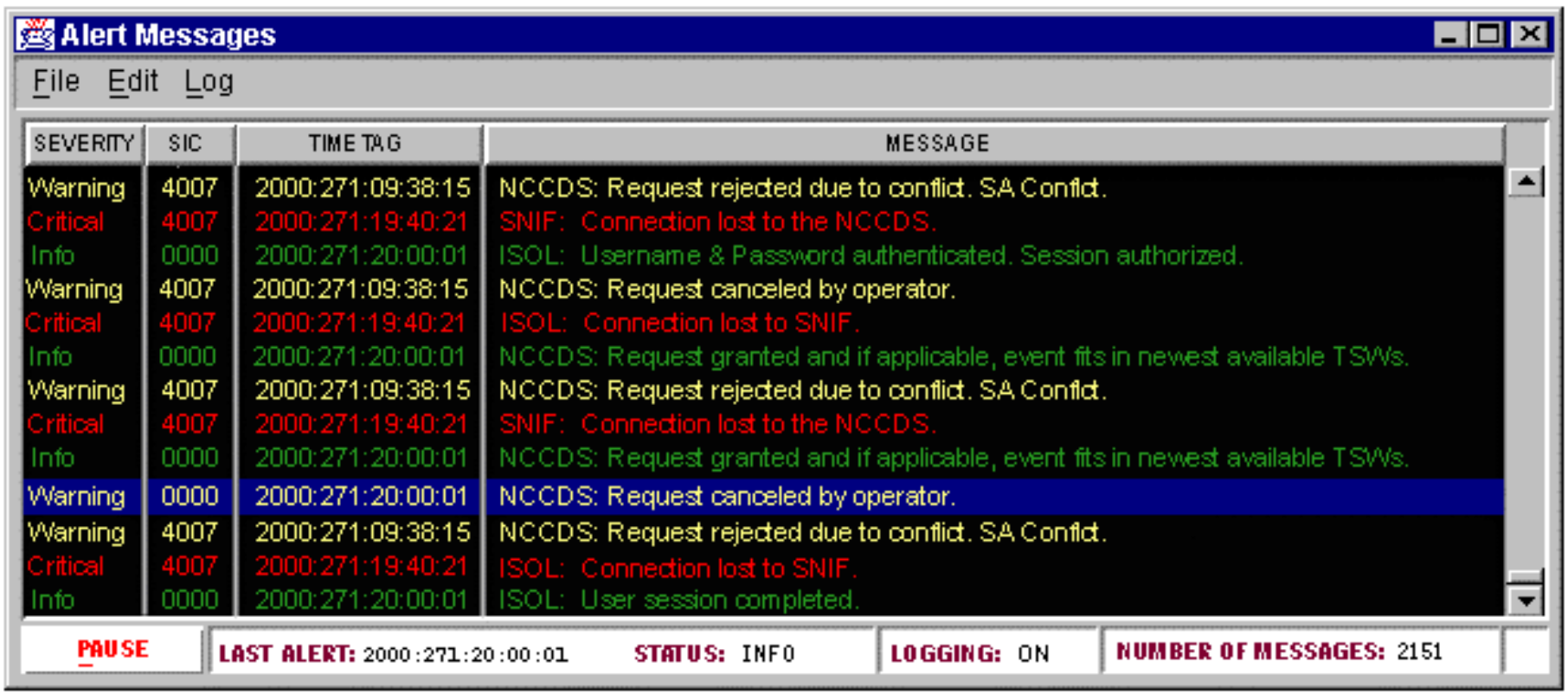
SWSI			
Control	Panels	Current GMT	Help
Log-in	Alerts	GMTClock	User's Guide
Log-out	Active Schedules		About SWSI
Exit	Schedule Requests		
Metal Look and Feel	GCMRs		
Motif Look and Feel	UPDs		
Windows Look and Feel	State Vectors		
Mac Look and Feel	TSWs		
	Create SAR		



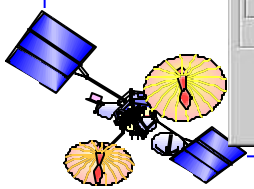
Login Panel


Host
localhost
Port
3028
User ID
Password
PassPhrase
Normal operational mode
Test (EIF) mode
Initiate Password Change?
Initiate PassPhrase Change?
Last login:
Number failed login attempts:
Login
Logout
Done

Alert Panel


Alert Messages
File Edit Log
SEVERITY SIC TIME TAG MESSAGE
Warning 4007 2000:271:09:38:15 NCCDS: Request rejected due to conflict. SA Conflct.
Critical 4007 2000:271:19:40:21 SNIF: Connection lost to the NCCDS.
Info 0000 2000:271:20:00:01 ISOL: Username & Password authenticated. Session authorized.
Warning 4007 2000:271:09:38:15 NCCDS: Request canceled by operator.
Critical 4007 2000:271:19:40:21 ISOL: Connection lost to SNIF.
Info 0000 2000:271:20:00:01 NCCDS: Request granted and if applicable, event fits in newest available TSWs.
Warning 4007 2000:271:09:38:15 NCCDS: Request rejected due to conflict. SA Conflct.
Critical 4007 2000:271:19:40:21 SNIF: Connection lost to the NCCDS.
Info 0000 2000:271:20:00:01 NCCDS: Request granted and if applicable, event fits in newest available TSWs.
Warning 0000 2000:271:20:00:01 NCCDS: Request canceled by operator.
Warning 4007 2000:271:09:38:15 NCCDS: Request rejected due to conflict. SA Conflct.
Critical 4007 2000:271:19:40:21 ISOL: Connection lost to SNIF.
Info 0000 2000:271:20:00:01 ISOL: User session completed.
PAUSE
LAST ALERT: 2000:271:20:00:01 STATUS: INFO
LOGGING: ON
NUMBER OF MESSAGES: 2151

Schedule Add Request Panel



Create SAR

Opened: <2000/233 20:46:07>

Message Class: SAR Request ID: 0000000 Explanation

SUPIDEN: A0338CS TDRS: PBK

ReferencedRequestID: None Priority: 1

Prototype Events / SSC Add

Name	Type
A01	MAF Normal
B01	MAR Normal
H01	SSAF Normal
H02	SSAF Normal
H03	SSAF Normal
H04	SSAF Normal
H05	SSAF Normal
H06	SSAF Normal

Nominal Event Start Time: 2000 233 20 49 41 Freeze Interval: 028 00 00 00

Plus Tolerance: 00 00 00 Wait List if unscheduled

Minus Tolerance: 00 00 00 Use TSWs to constrain scheduling

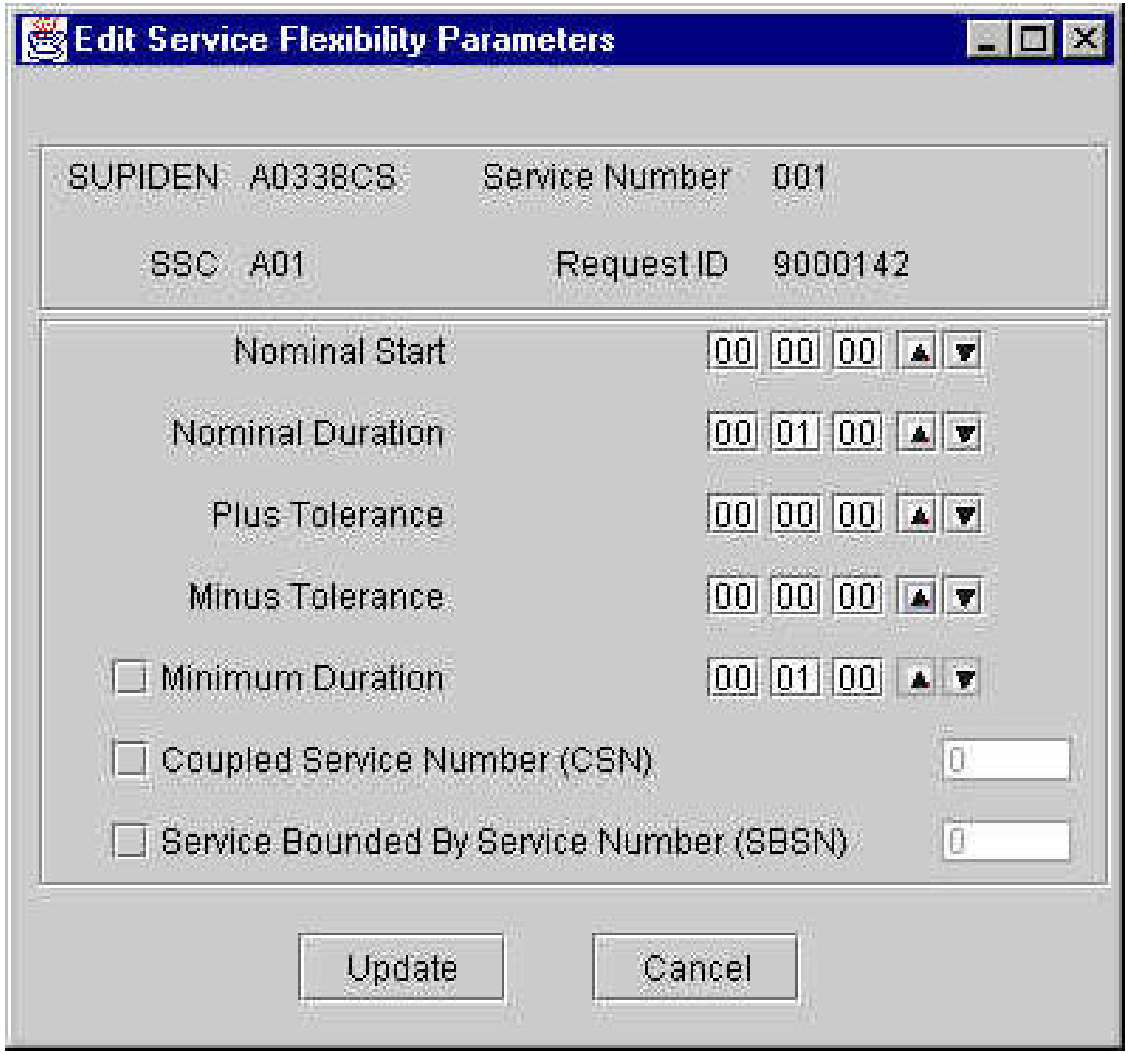
Service Request

Number	SSC	Service Type	Nominal Start	Nominal Duration	CSN	SBSN	(+)Tolerance	(-)Tolerance	Minimum Duration
001	A01	MAF Normal	00:00:00	00:01:00	000	000	00:00:00	00:00:00	00:01:00
002	B01	MAR Normal	00:00:00	00:01:00	000	000	00:00:00	00:00:00	00:01:00
003	T01	Tracking Normal	00:00:00	00:01:00	000	000	00:00:00	00:00:00	00:01:00

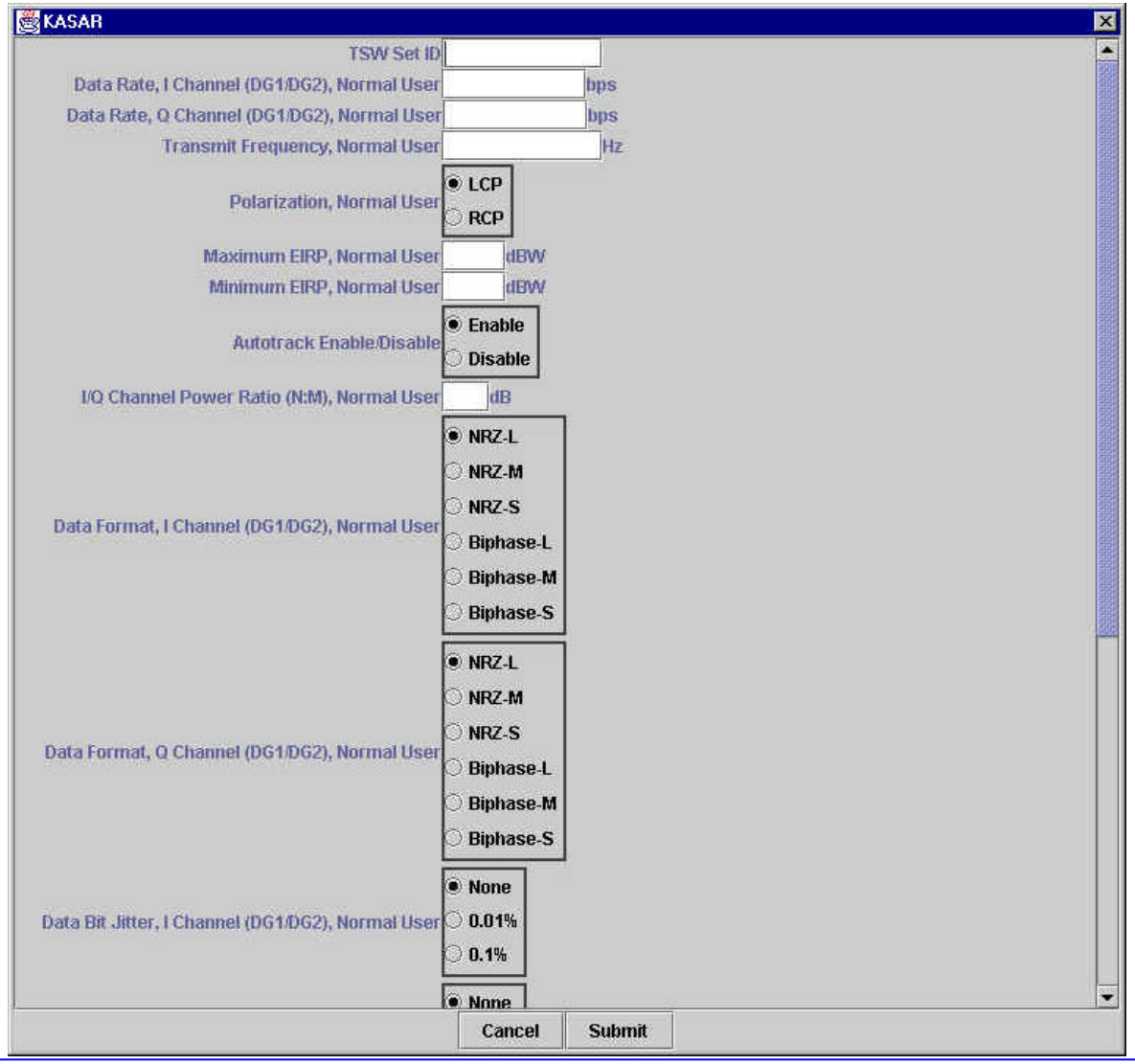
Remove Move Up Move Down Remove All Modify Service... Parameters...

Submit Cancel

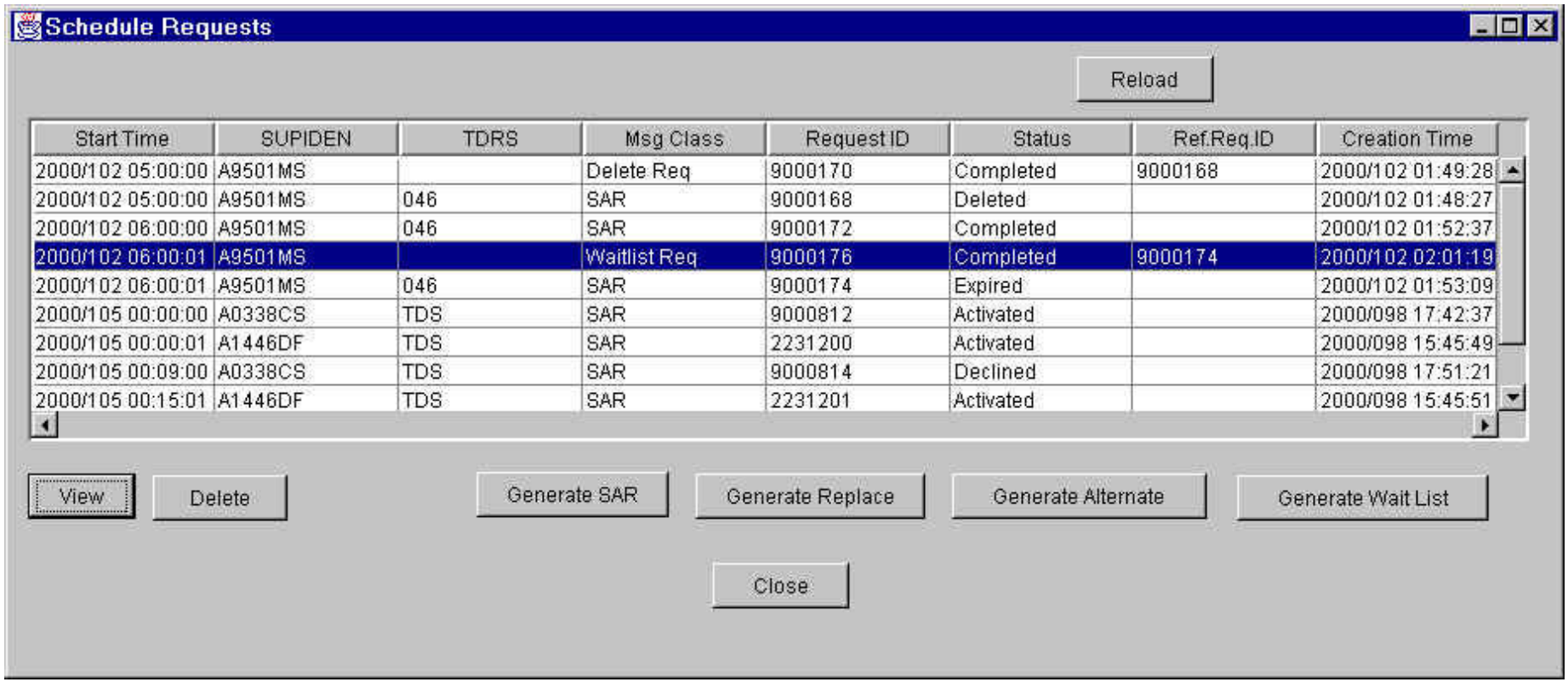
Flexibility Parameters Panel



Respecifiable Parameters



Schedule Requests Panel



Referencing Requests

SDR - select request and press 'Delete'

WLR - select request and press 'Generate Wait List'

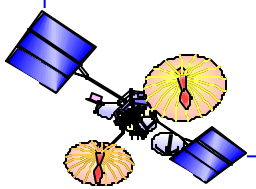
- **panel prompting for expiration date will appear**

ASAR - select request and press 'Generate Alternate'

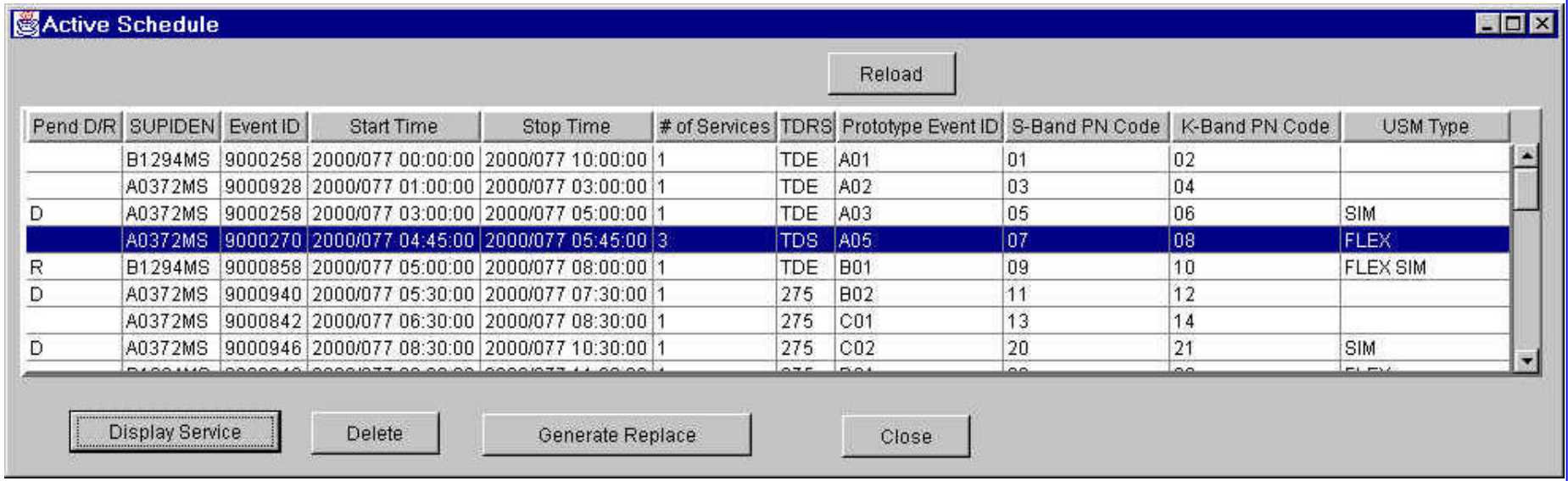
- **panel similar to SAR appears**
- **SUPIDEN, priority, and wait list flag fields are 'grayed-out'**

RR - select request and press 'Generate Replace'

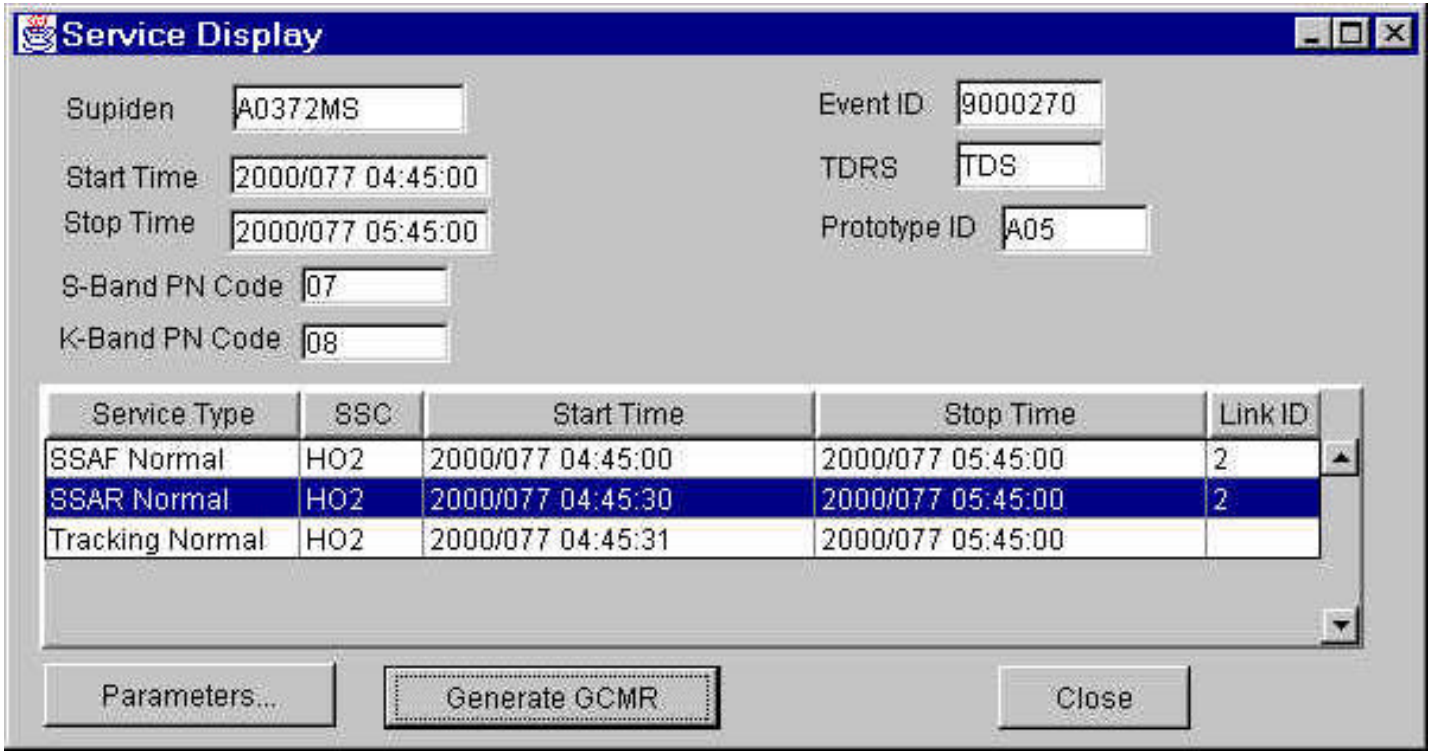
- **panel similar to SAR appears**
- **SUPIDEN and priority fields are 'grayed-out'**



Active Schedule Panel

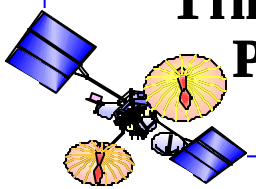


Service Display Panel



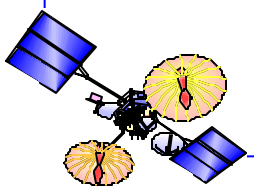
UPD Summary Panel

- User can monitor User Performance Data (UPDs) for all SICs associated with that User ID via the UPD Summary Panel
- Summary Panel contains, for each active service:
  - UPD type
  - TDRS ID
  - SUPIDEN
  - antenna or link number
  - service status (button)
  - Submit GCMR (button)
- Service Status button shows overall UPD status (text and color)
- Selecting Service Status button generates UPD Detail Panel
- Selecting Submit GCMR button triggers GCMR Menu Panel
  - pre-filled with SUPIDEN, TDRS ID, and service type
- Time-outs flag and remove ended service from Summary Panel



UPD Detail Panel

- **Standard layout dynamically created based on client login setup**
  - **Standard layout includes all UPD parameters**
- **Layout is user customizable (drag and drop)**
  - **User can select parameters to display and/or reorder them**
- **Customized layouts can be saved by UPD type**
- **Users can specify a default layout per UPD type**
- **Users can switch between default and other layouts**
- **Data displayed as:**
  - **Text types**
  - **Numerical types**
  - **Enumeration types with severity given in color**



GCMR Generation

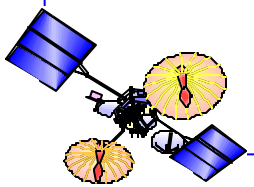
GCMR Menu Panel available from:

- Main Panel (user must know which service type/SUPIDEN/TDRS ID is active)
- Service Display Panel (service type/SUPIDEN/TDRS ID get pre-filled)
- UPD Summary Panel (service type/SUPIDEN/TDRS ID get pre-filled)

User selects GCM Type:

- Service Reconfiguration
- User Reacquisition Request
- Forward Link Sweep Request
- Forward Link EIRP Reconfiguration - Normal Power
- Forward Link EIRP Reconfiguration - High Power
- Expanded User Frequency Uncertainty Request
- Doppler Compensation Inhibit Request - none SSA Shuttle

For 'Service Reconfiguration', a reconfigurable parameters panel is generated (similar to Respecifiable Parameters panel)



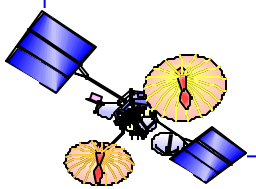
Client Input and Output Files

Client Input Files:

Name	Description
ssl.prop	Java property file containing SSL specific setup properties
client.prop	Java property file containing client specific setup properties
SWSI-ca-cert.der	Certificate authority certificate
SWSI-client-cert.der	Client Certificate
enc-SWSI-client-key.der	Encrypted Client key
any.tsw	User supplied TDRS Scheduling Windows in ICD format*
any.iirv	User supplied IIRV data in ICD format*
window_name.win	User defined UPD window layouts

Client Output Files:

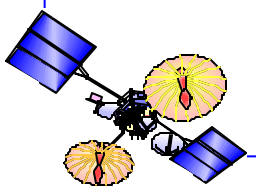
Name	Description
User_ID_#.log	Log file of alerts
(EIF or norm)_rctd_(date/time)_#.dat	return channel time delay data
(EIF or norm)_ttm_(date/time)_#.dat	time transfer message data
window_name.win	User defined UPD window layouts



***** *Interface Control Document Between the Network Control Center Data System and Mission Operations Center*, 530-ICD-NCCDS/MOC

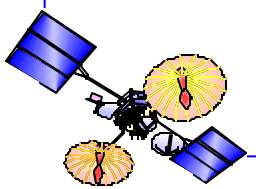
Application Server Design

Geri Klitsch
CSC

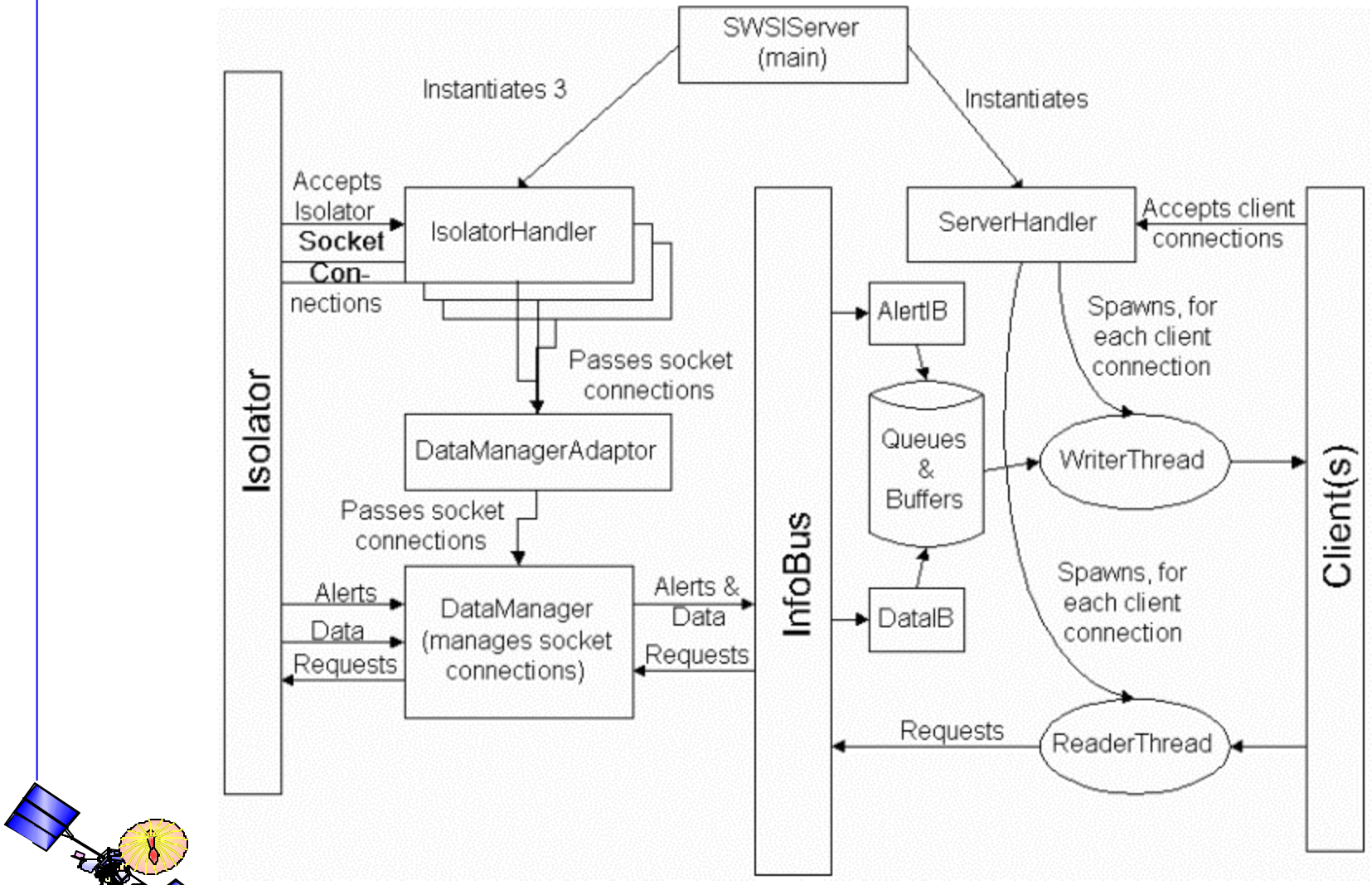


Application Server Functionality

- **Accept Isolator and Client SSL connections**
  - Authentication using digital certificates
  - Tag Client logins (user ID and password) with IP address and forward to Isolator for validation
- **Accept client schedule and data requests**
  - Tag with user ID and forward to Isolator
- **Accept client file transfers (SVs, TSWs)**
  - Tag with user ID and forward to Isolator
- **Route Isolator responses to clients**
- **Route alerts to clients**
- **Route Real-Time data (UPDs, RCTDs, & TTMs) to clients**
- **Inform Isolator of client disconnects**
- **Accept Isolator reconnects**



Application Server Design



Application Server Main Threads

Application Server Main Task - main thread

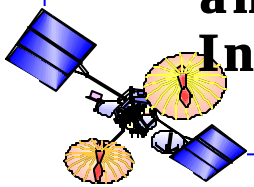
IsolatorHandler - creates server sockets and waits for Isolator connections. Passes connections through DataManagerAdaptor to DataManager

DataManager - creates 3 threads (one for each socket) that read/write the socket and get/pass data from/to the InfoBus

InfoBus - A class library distributed by Sun Microsystems and developed by Lotus Development Corp. that interconnects beans or classes and supports the exchange of data items

ServerHandler - creates a server socket and accepts client connections. For each client connection, the ServerHandler clones itself and spawns 2 threads - WriterThread and ReaderThread

The ServerHandler clone creates an instance of the AlertIB and DataIB classes which receive alerts and data from the InfoBus to send to the client through the WriterThread



Application Server Logging

Activity Log

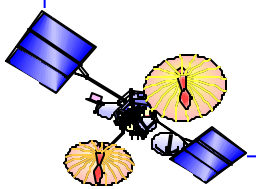
- Records successful user logins
- Records user Ids and IP addresses
- Records user requests (activity) with time and user ID

Failed Login Log

- Records unsuccessful logins
- Records address of failed connection attempt
- Records reason for failed connection

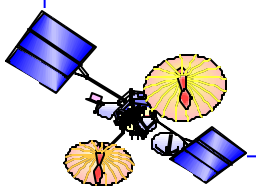
Debug Logging

- Allows levels of debug to be set

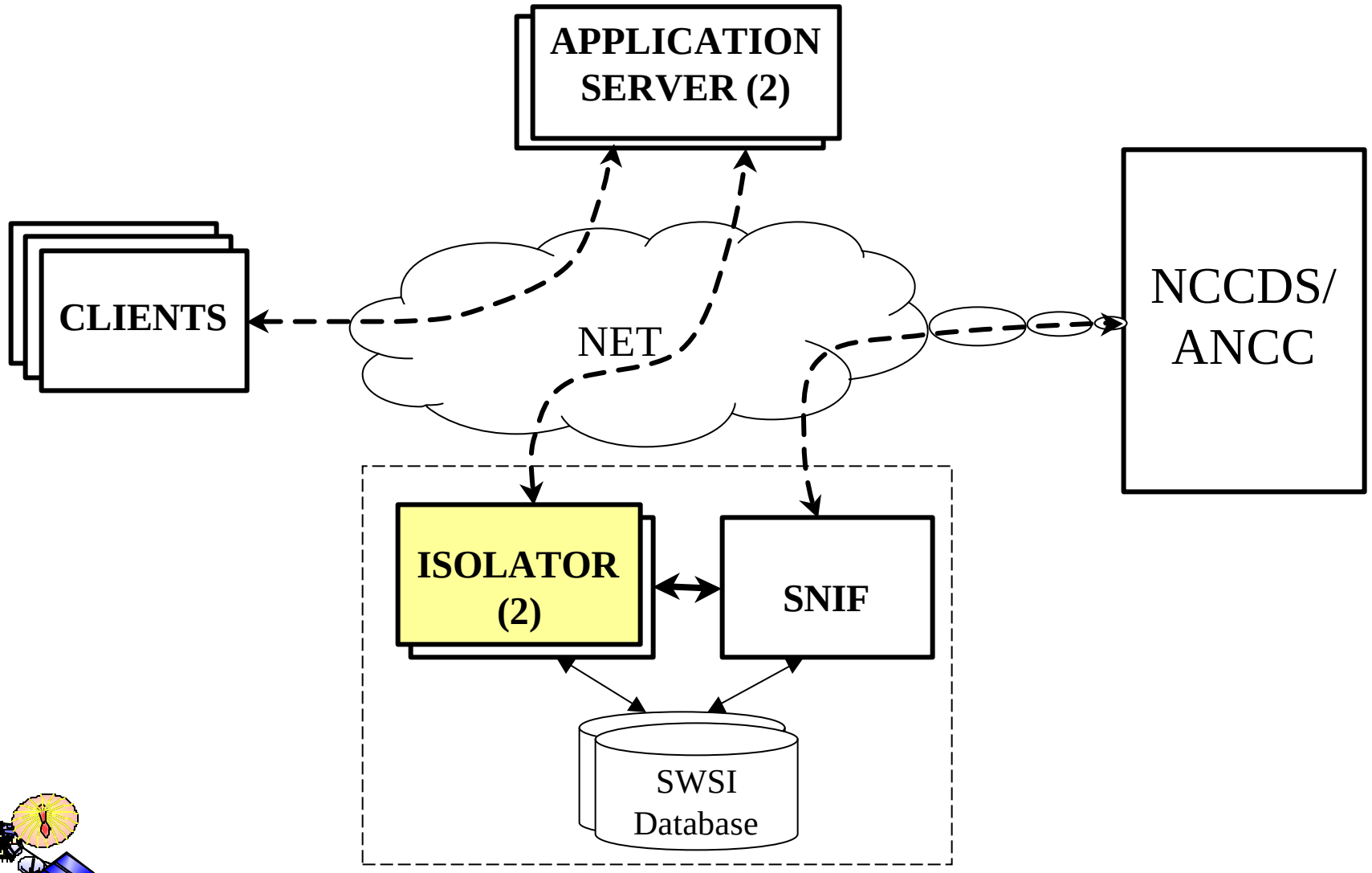


Isolator Design

Maurice Assaraf
CSC



SWSI High Level Diagram



SWSI High Level Diagram (Cont'd)

2 Application Servers

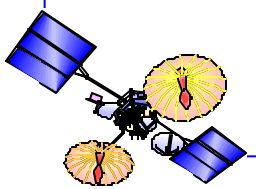
- One runs on Open Server
- One runs on Backend Server

2 Databases

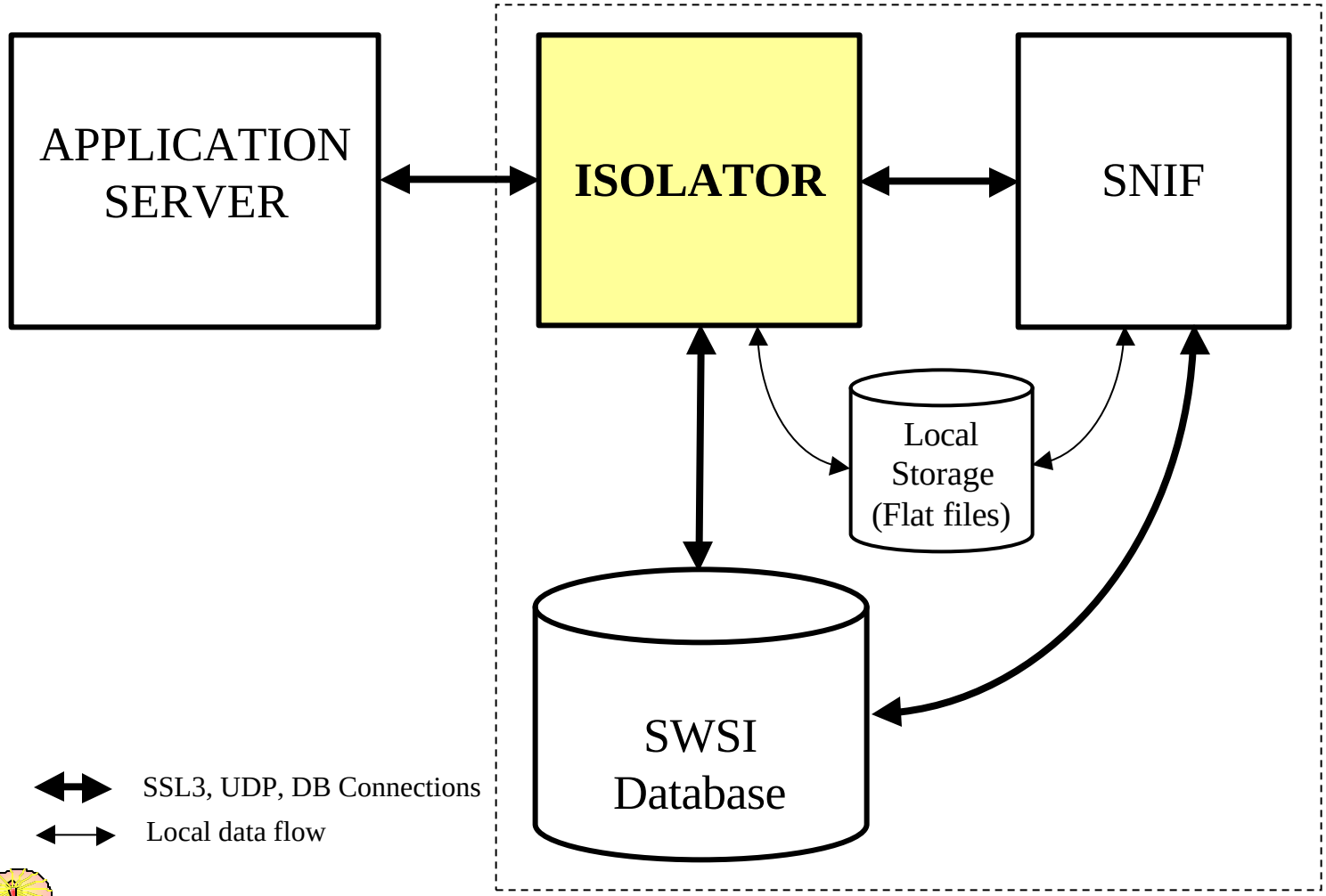
- NCCDS Database
- ANCC Database

2 Isolators

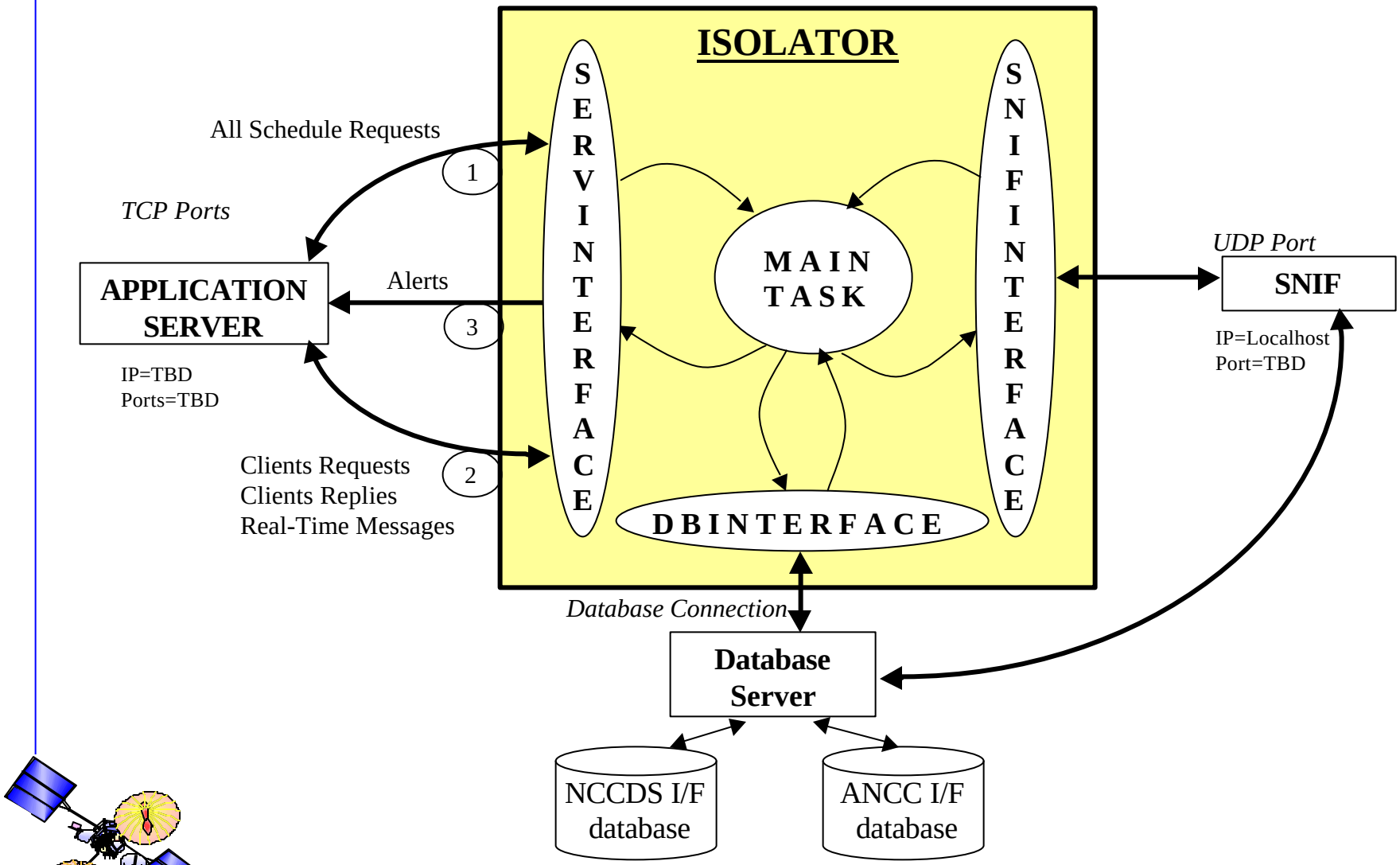
- Both run on Backend Server
  - Communicates with Application Server on Open side
  - Communicates with Application Server on Closed side
  - Connects with both Databases (NCCDS I/F & ANCC I/F)



Isolator Context Diagram



Isolator Main Threads



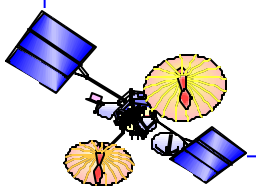
Isolator Main Threads (Cont'd)

Isolator Main Task (MainTask)

Application Server Interface (ServInterface)

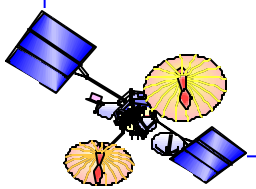
Database Interface (DbInterface)

SNIF Interface (SnifInterface)



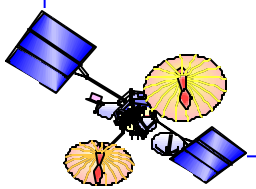
MainTask Thread

- Reads profile/configuration data for initial parameters
- Initiates all the Isolator main threads
- Manages and routes all processing and I/O requests to the appropriate Isolator threads
- Monitors the status and events of all the Isolator threads and queues (Executive task)
- Logs System's error messages

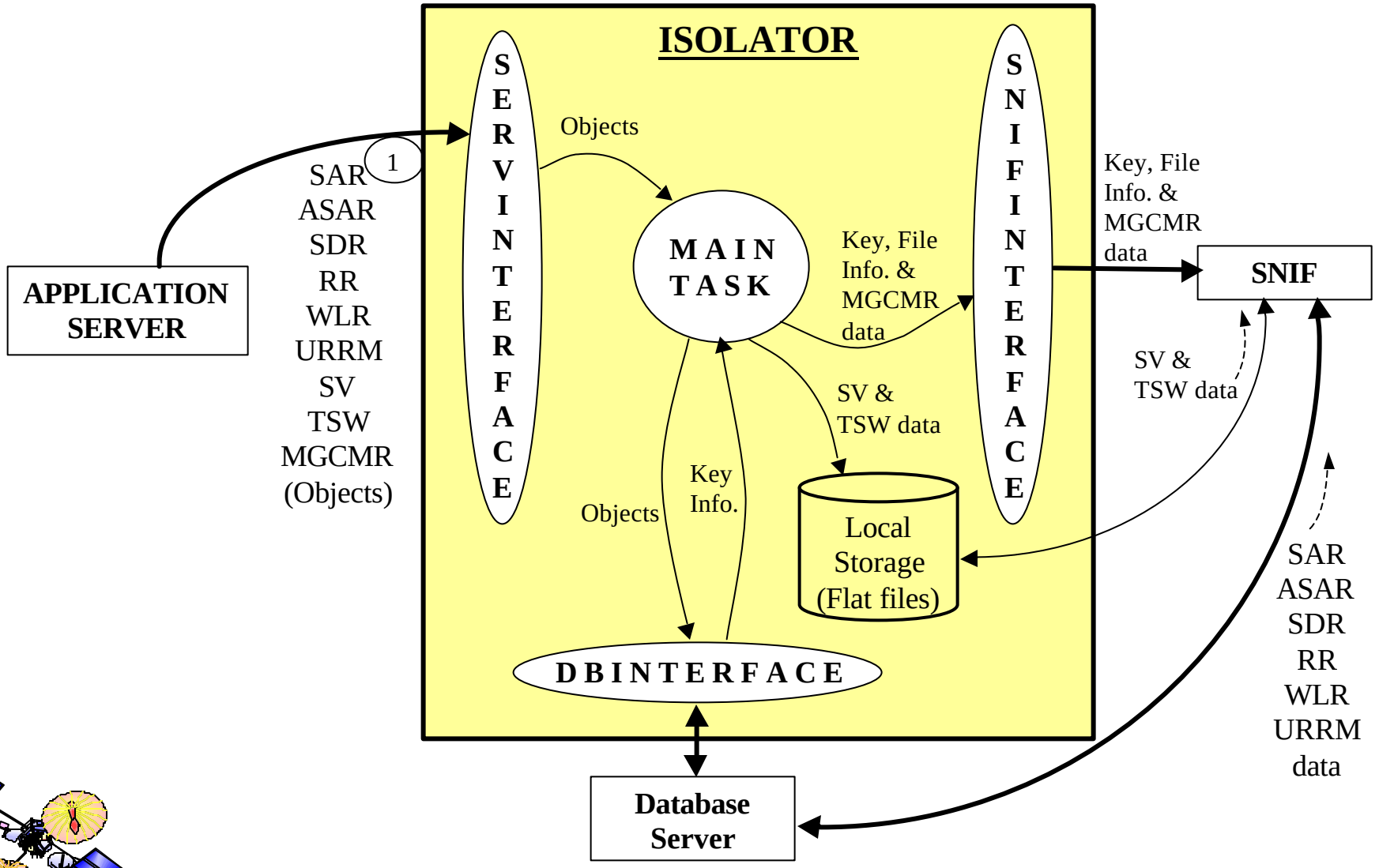


ServInterface Thread

- Handles all the communications between the Application Server and the Isolator
- Uses 3 TCP/IP (SSL3) ports
- Initiates 3 sub-threads TP1, TP2, and TP3
  - TP1 accepts all the Schedule Requests
  - TP2 accepts all users requests and sends back users replies and Real-time Messages
  - TP3 sends all the alert messages



TP1 Messages



TP1 Messages (Cont'd)

Stored in Database

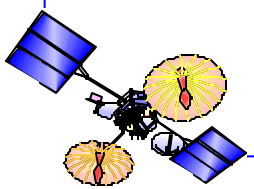
- Schedule Add Request (SAR)
- Alternate SAR (ASAR)
- Schedule Delete Request (SDR)
- Replace Request (RR)
- Wait List Request (WLR)
- User Reconfiguration Request (URRM)

Stored in Flat Files

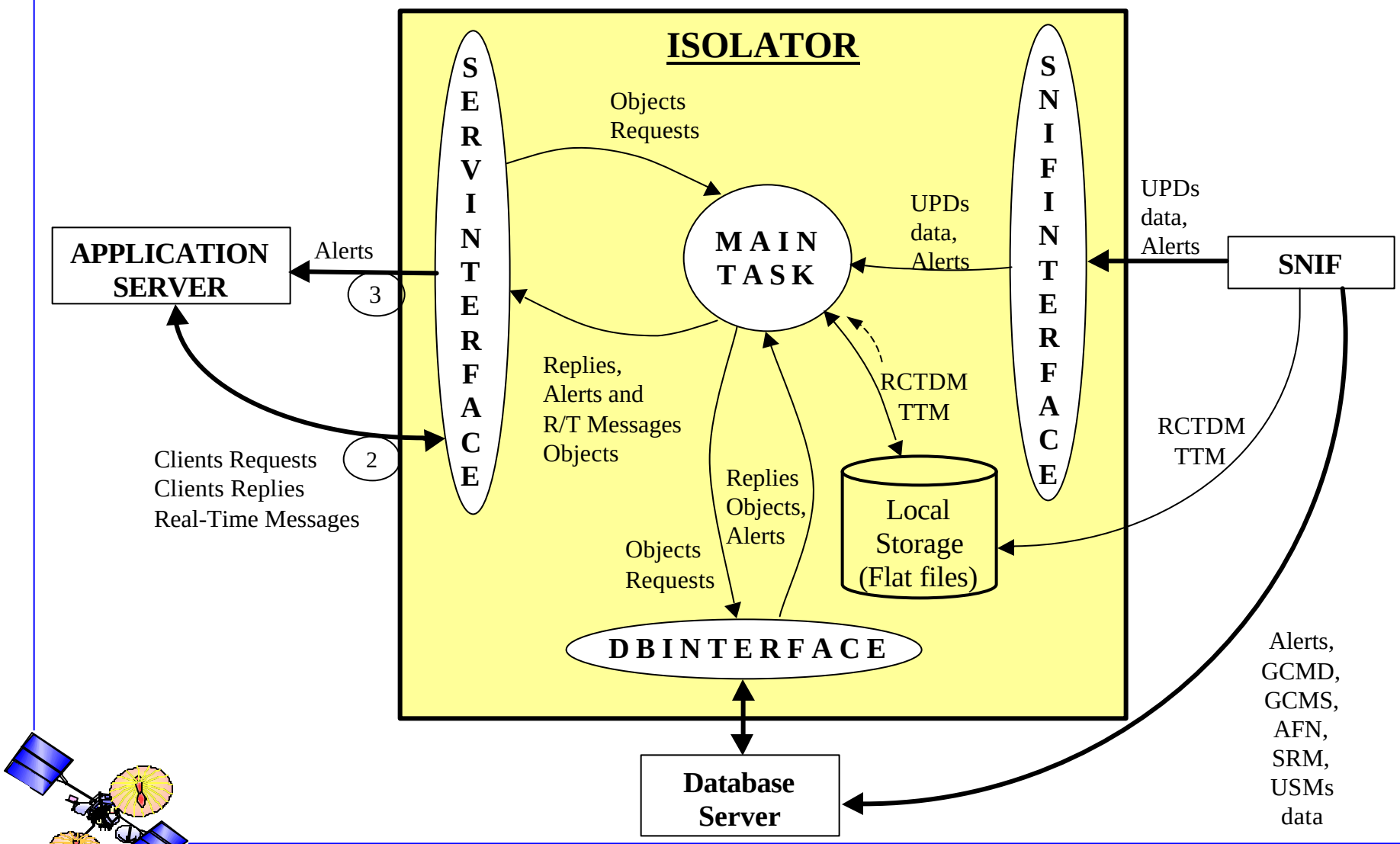
- State Vector (SV)
- TDRS Scheduling Window (TSW)

Forwarded directly

- Multiple Ground Control Message (MGCMR)
  - User Reacquisition Request (URR)
  - Forward Link Sweep Request (FLSR)
  - Forward Link EIRP Reconfiguration (FLER)
  - Expanded User Frequency Uncertainty Request (EUFUR)
  - Doppler Compensation Inhibit Request (DCIR)



TP2/TP3 Messages



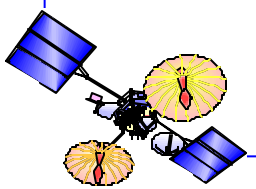
TP2/TP3 Messages (Cont'd)

TP2 Messages:

- User Login
- User requests (data from database/files)
- User replies (data from database/files)
- Return Channel Time Delay Measurement (RCTDM) & Time Transfer Message (TTM)
- User Performance Data (UPD)

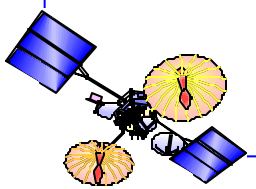
TP3 Messages:

- Alerts



DbInterface Thread

- Connects to the SWSI database server
- Gets objects from MainTask
- Sends SQL directives to database server to:
  - Store pertinent objects data in DB tables
  - Retrieve requested data from DB tables
  - Polls for start/stop of events to generate alerts
- Formats the retrieved data into objects
- Passes objects and Key Info. to MainTask



Major Database Tables accessed by Isolator

SCHEDULE_REQUEST, SR_SERVICE, SC_PARAM (Schedule Requests)

USR_GCMR, GCMR_PARAM (User Ground Control Message Requests)

ACTIVE_SCH_SERVICE (Read parameter values for GCMR)

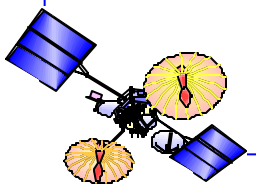
SWSI_USER (user authentication)

SSC, SSC_PARAMS (client initialization)

PROTOTYPE_EVENT_CODE (client initialization)

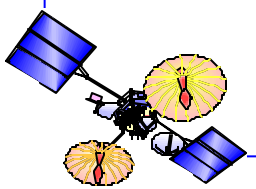
UPD and its associated tables (client initialization)

SIC, SUPIDEN, TDRS_NAME (client initialization)



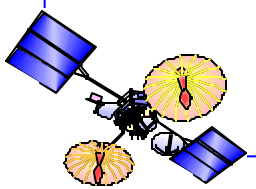
SnifInterface Thread

- **Handles all the communications between the Isolator and SNIF**
- **Uses 1 UDP port for data exchange**
- **Accepts from SNIF**
  - **Formatted User Performance Data**
  - **Alerts**
  - **File Info. messages stored in files**
- **Sends to SNIF**
  - **Key Info. of data stored in database**
  - **File Info. of messages stored in files**
  - **MGCMR messages**



SNIF Design

Tom Sardella
Code 583/450



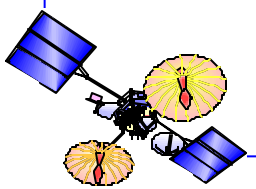
SNIF Overview

Functionality

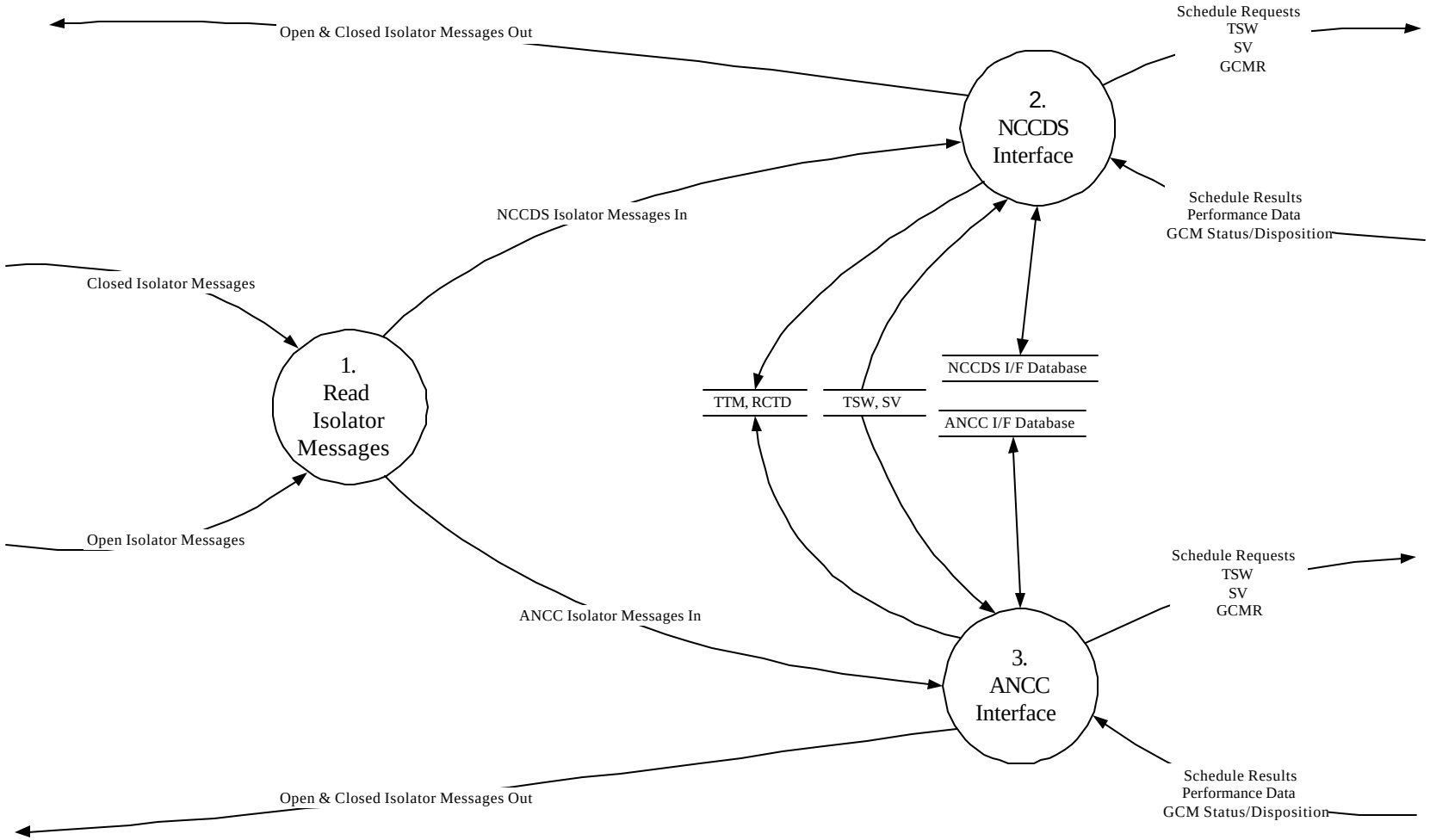
- Interface to NCCDS and ANCC
- Establishes and maintains appropriate TCP connections with NCCDS for each mission group
- Implements message interface as defined in NCCDS/MOC ICD
- Maintains Active Schedule based on SRM & USM responses from NCCDS

Environment

- Back end “C” application
- POSIX threads (pthreads) for concurrency within single process
- Separate thread assigned for each blocking socket I/O connection
- Custom queue routines for inter-thread communications based on pthread mutex



SNIF Level 0 Diagram



SNIF High-Level Processes

Read Isolator Messages Thread

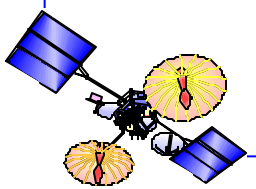
- Single thread reads all Isolator UDP messages
- Routes incoming Isolator messages to appropriate NCC interface process

NCCDS Interface Process

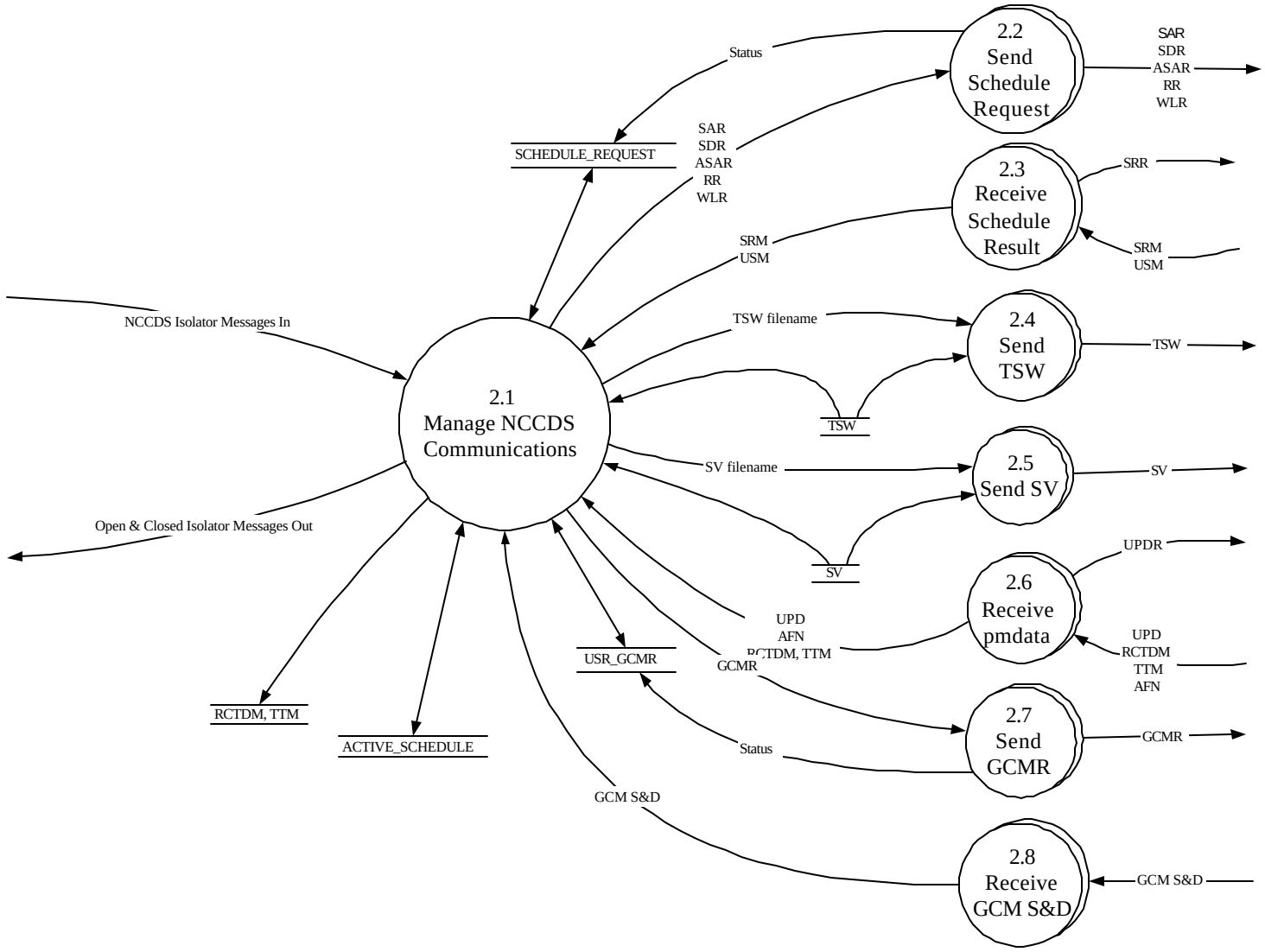
- Controls all communication with NCCDS
- Access NCCDS I/F Database

ANCC Interface Process

- Controls all communication with ANCC
- Access ANCC I/F Database



NCCDS I/F Process



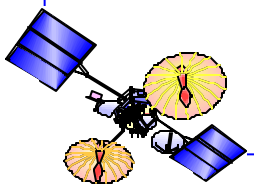
NCCDS I/F Process

Manage NCCDS Communications Thread

- Primary thread for processing NCCDS messages
- Request messages constructed from Isolator messages and database entries
- Uses SRMs & USMs from NCCDS to update SCHEDULE_REQUEST and ACTIVE_SCHEDULE Database tables
- Reformats UPDs as name-value pairs
- Generates Client alerts for SRM, USM, GCM Status & Disposition, Acquisition Failure Notification, etc.
- Stores RCTDM, TTM in files

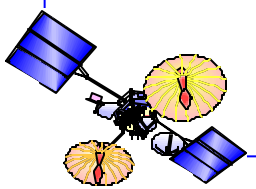
Connection Control Threads

- Separate thread for each NCCDS service (schReq, schStatus, etc)
- Separate instance of each thread for each connection
- Permanent connections for receive data (schStatus, pmData, reconfig)
- Temporary connections for transmit data (schReq, tswStore, acqStore)
  - Time out after configurable period of inactivity
- Connection configuration defined by Database. System restart required after configuration change



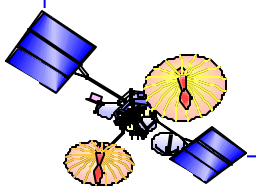
SNIF Logging

- **Messages logged in NCCDS Centralized Delogger (NCD) format**
  - **All formatted messages exchanged with NCCDS & ANCC**
  - **Significant events and errors (e.g., connection establishment and loss)**
- **NCCDS Protocol Gateway (NPG) Delogger used to delog and display previously logged data**
- **Debug-level logging to text file to troubleshoot application and system problems**



Database Design

Harshna Sampat
CSC



Database Overview

RDBMS (Oracle 8i)

Static Tables

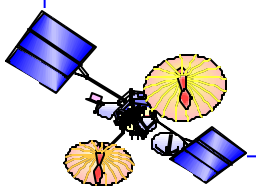
- **Synchronized with NCCDS & ANCC database via SQL scripts**
- **Manual updates of some static data by DBA**

Dynamic Tables

- **Updated by SWSI software (Isolator and SNIF)**

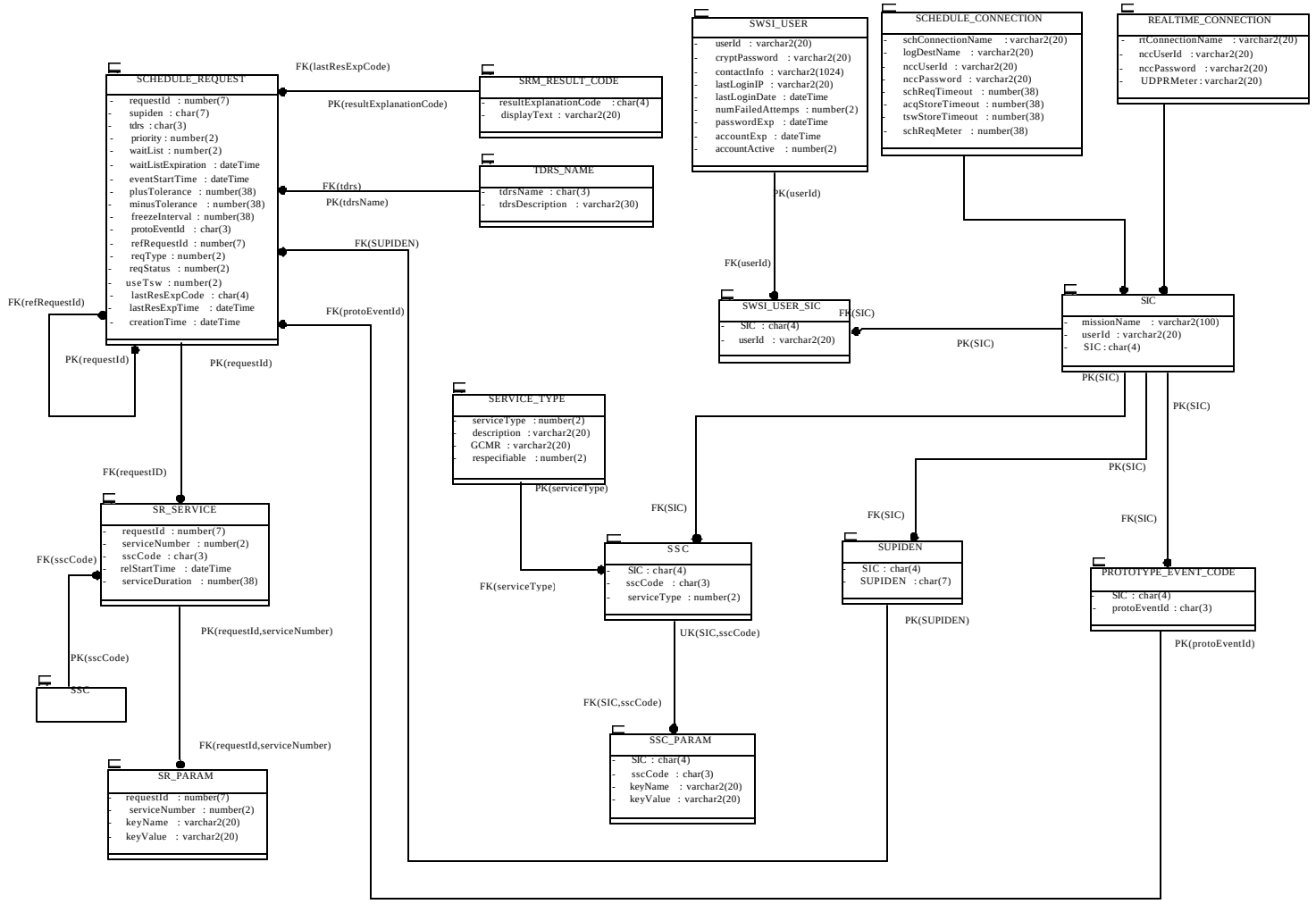
Data Purging of old data by a 'cron' script

Backup/Recovery using Oracle Enterprise Manager



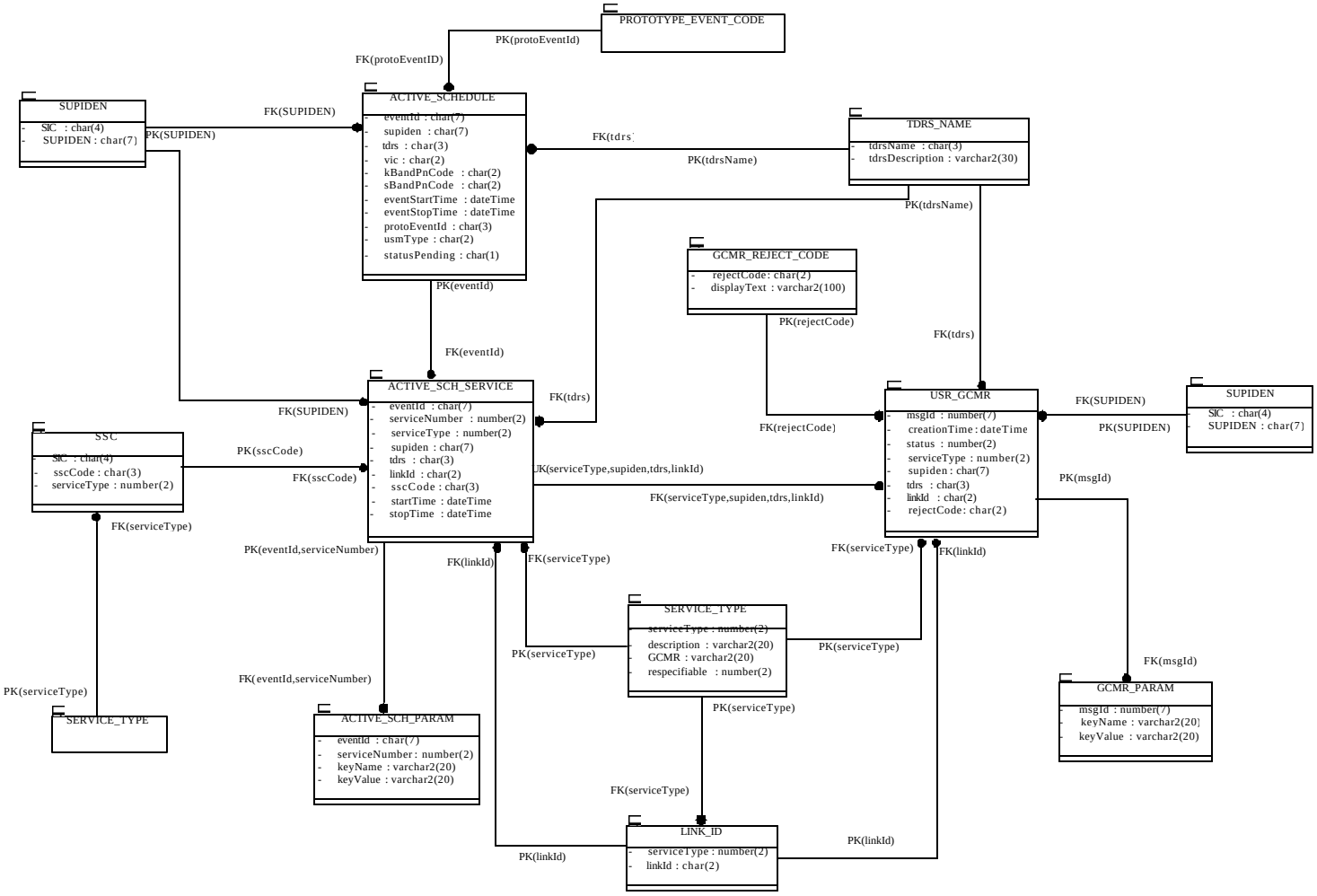
Database Schema (1 of 3)

swsi-01-database on 10/3/2000



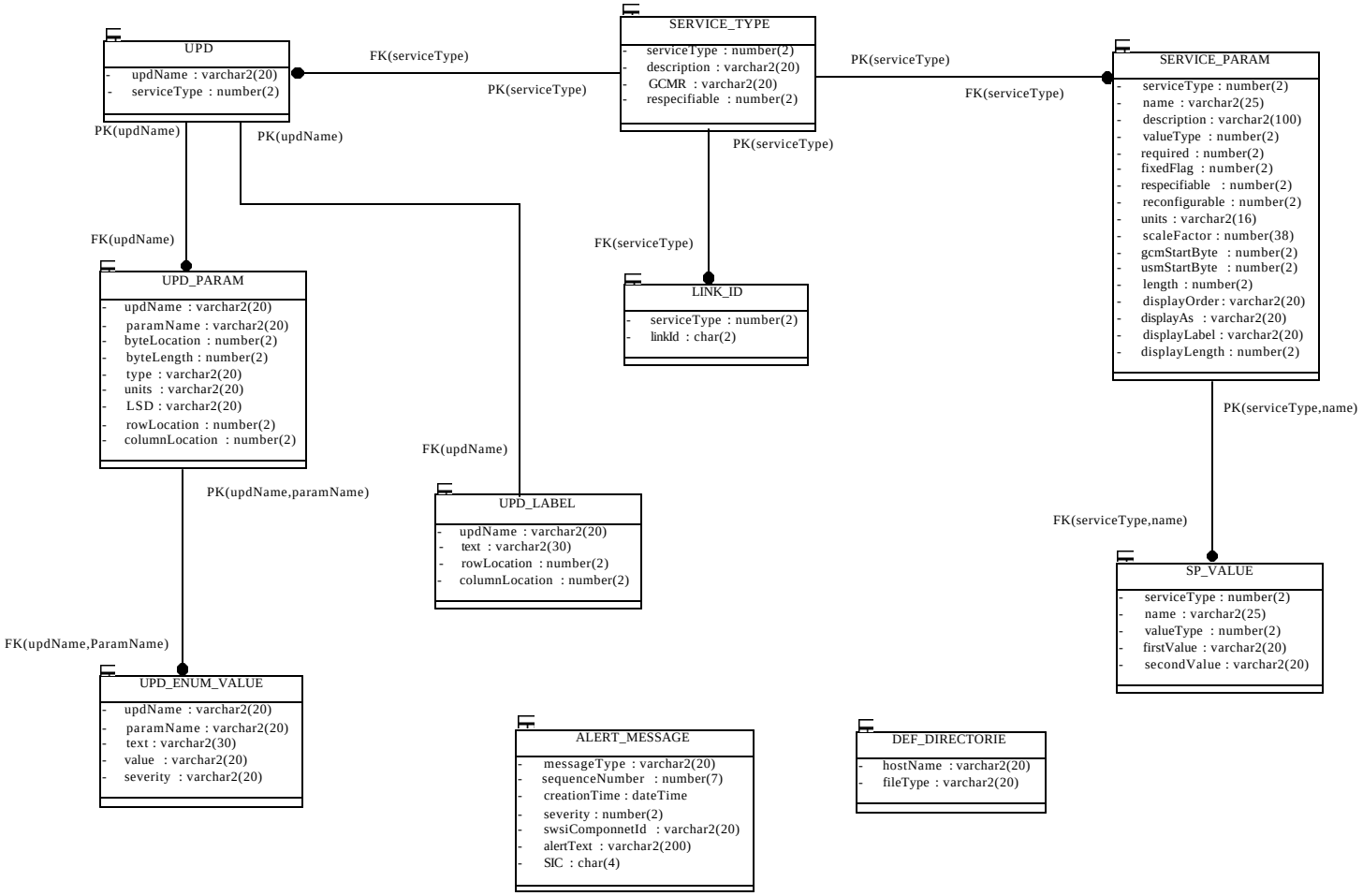
Database Schema (2 of 3)

swsi-02-database on 10/3/2000



Database Schema (3 of 3)

swsi-03-database on 10/3/2000



Major Static Tables

SSC

- Every SIC can have multiple Service Specification Codes
- Contains parameter values pre-set for each SSC Code

PROTOTYPE_EVENT_CODE

- Contains valid Prototype Event Codes for each SIC

SERVICE_PARAM

- Contains information for Client to construct service parameter displays

SWSI_USER

- Contains Valid SWSI user ids and related information

SCHEDULE_CONNECTION, REALTIME_CONNECTON

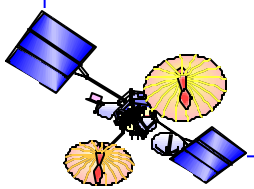
- Contains information to make socket connections with NCCDS/ANCC

SERVICE_TYPE

- Contains all valid service types supported by SWSI

UPD

- Contains information for Client to construct UPD displays



Major Dynamic Tables

SCHEDULE_REQUEST

Stores all the Schedule Requests performed by SWSI users

USER_GCMR

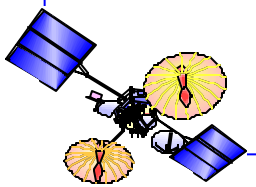
Stores all the User Reconfiguration Requests (98/04) made by SWSI users

ACTIVE_SCHEDULE

Stores information from USMs received from NCCDS/ANCC

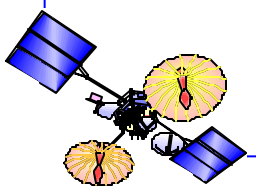
ALERT_MESSAGE

Stores all the alerts produced by SWSI



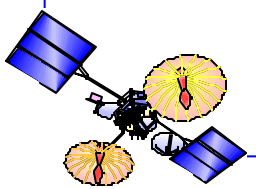
Summary

Tom Sardella
Code 583/450



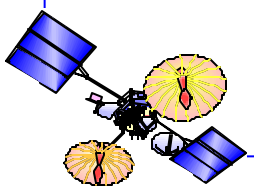
Code Estimates

Subsystem	Language	Delivered Source Instructions (DSI)	Reuse from Jswitch
Client	Java	15000	10%
Application Server	Java	6000	80%
Isolator	Java	5000	0%
SNIF	C	5000	N/A
Total		**31000**	



Issues/Concerns

- **Availability of NCCDS Vector Storage (acqStore) service**
- **Mission Support requirements**
  - **LDB 2 week missions will continue to be supported by SWSI prototype until new system operational**
  - **ULDB extended flight 12/01**
    - **Trust, prototype SWSI systems still available**
    - **UPS access is currently being set up**
  - **GP-B launch 5/02**
    - **Main concern is testing and training starting early Fall 01**
- **Build schedule still being developed to incorporate DAS requirement**
- **Performance impact of DAS interface is unknown**
  - **New hardware may be required**
  - **Evaluate performance late in development cycle and possibly upgrade/replace current hardware**



Issues/Concerns (Cont'd)

- NASA PKI Initiative required to be in place for transition to operations
- Hardware installation (GSFC Building 13 vs. WSC/DSMC) is still being worked
- SWSI/SWIS Integration