



A Distributed Network Architecture for PC-Based Telemetry Systems

by

**Paul A. Thoreson, Lockheed Martin
Telemetry & Instrumentation**

Presented to

Ground Systems Architectures Workshop

February 26, 1997

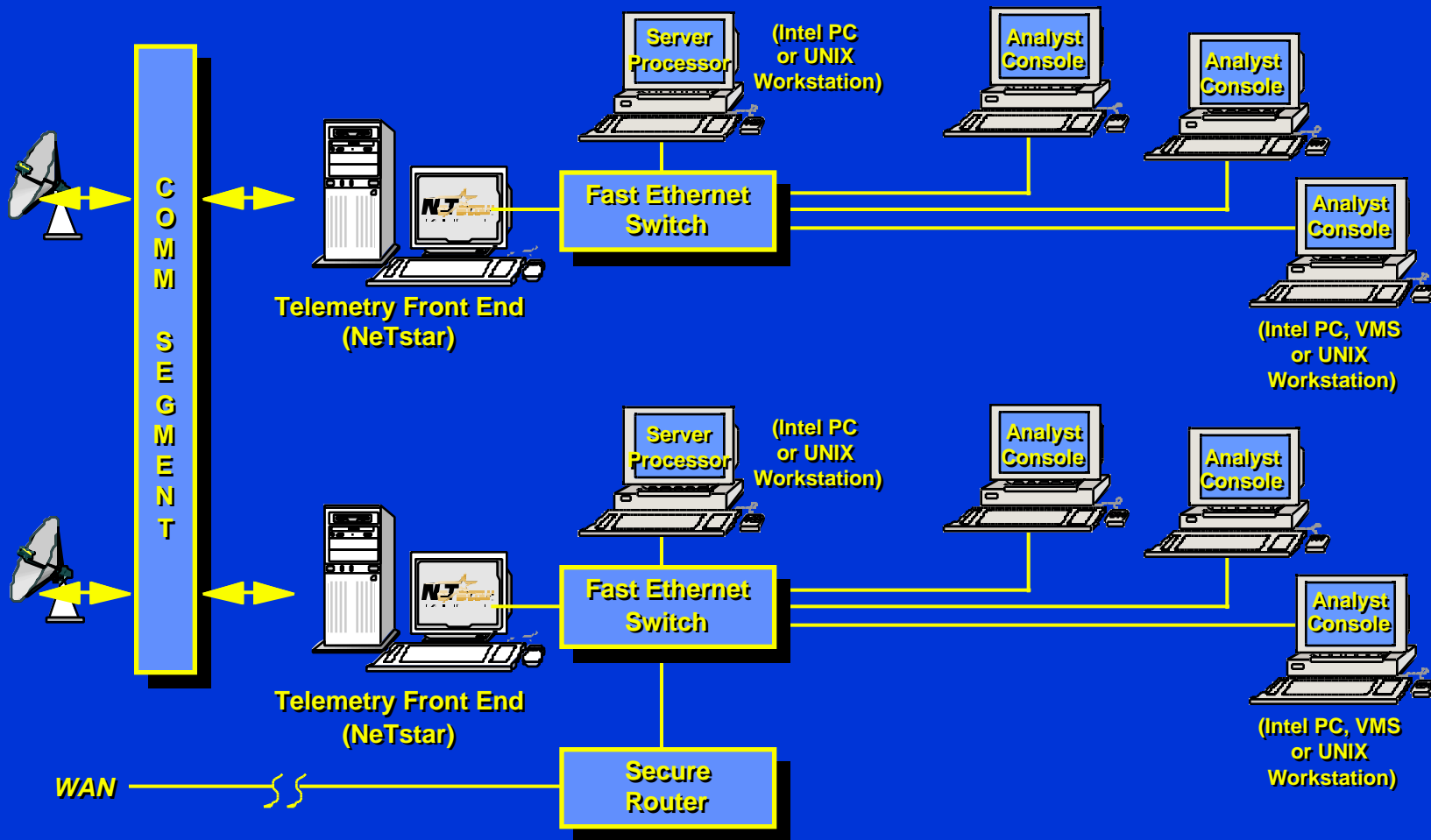


Objective:

To describe a candidate software architecture for a satellite ground station with emphasis on telemetry processing, software component interfacing, and platform interoperability

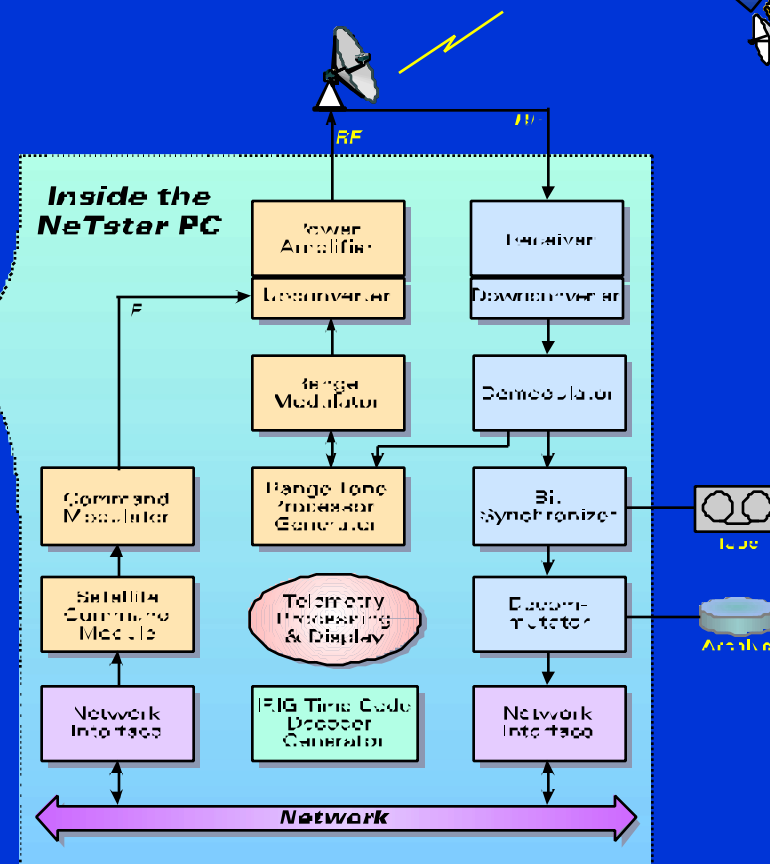
The described architecture is comprised of commercial off-the-shelf hardware and software components requiring a system integration to provide minor “glueware” or “wrapper” applications and software component upgrade synchronization

Hardware Architecture Overview



NeTstar Block Diagram

- Pentium Processors
- Windows NT
- Client / Server Setup
- LAN / WAN / Ethernet
- Remote and Local Control
- RF and 70 MHz Receivers / Demodulators
- Telemetry Processing, Display, and Analysis
- Data Archiving and Playback
- Uplink Modulation / Commanding / Ranging
- 1553 Bus Testing
- Command Software
- Ranging and Receiver Displays
- CCSDS Processing & Display
- Third-Party Hardware / Software-Compatible



Lockheed Martin Telemetry & Instrumentation

Hardware Components



Telemetry Front End:

- **NeTstar – LMTI's Windows NT-based PC TFE**
- **Alternative TFE – Any networked frame synchronized or decommutated data source**

Server Processor:

- **Intel CPU-based personal computer running Windows NT**
- **Alternative Server – DEC, SUN or SGI UNIX workstations**

Hardware Components (Cont.)



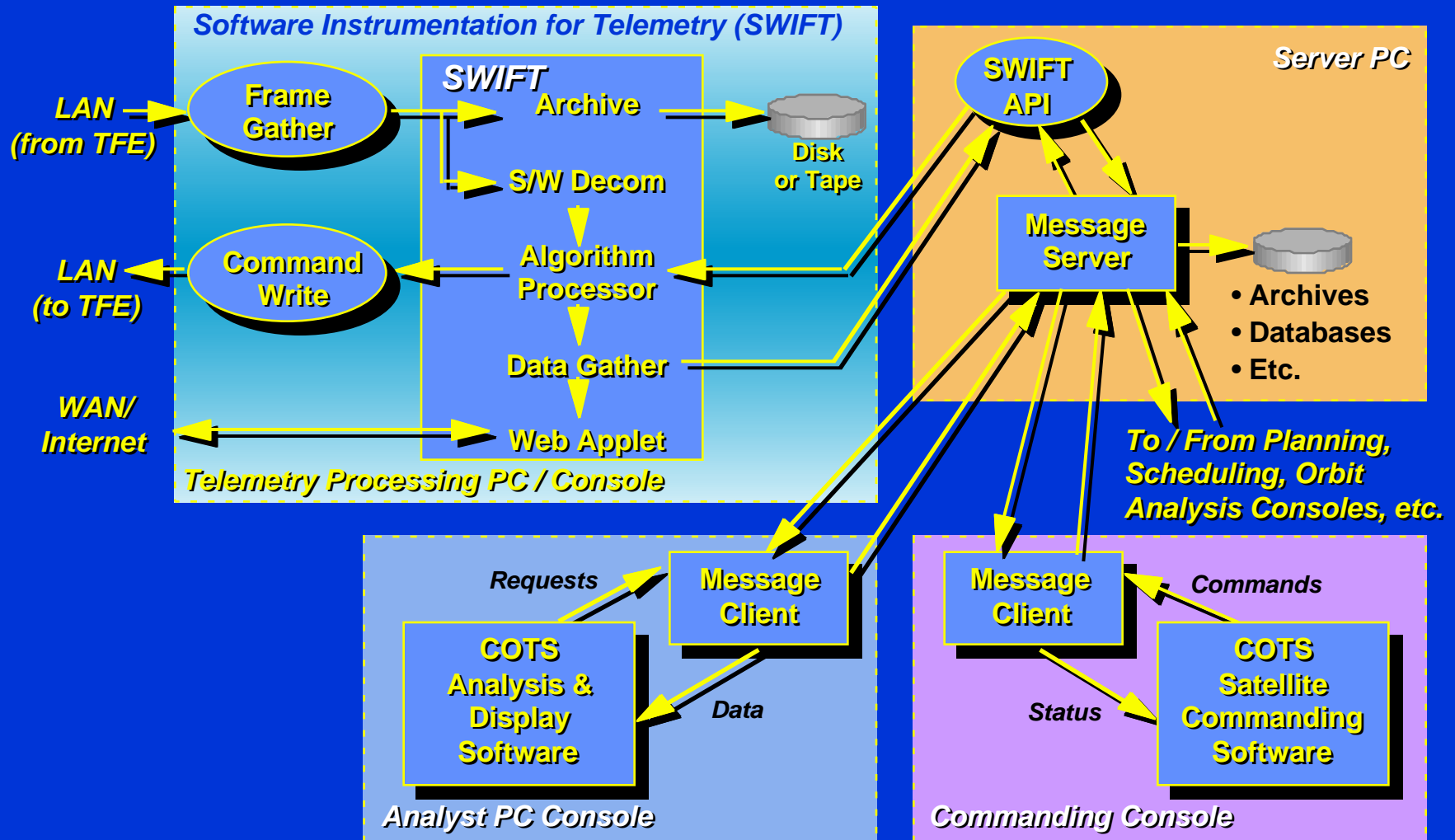
Analyst Station:

- Intel CPU-based personal computer running Windows NT
- Alternative Analyst Stations – DEC, SUN, SGI, IBM and HP UNIX workstations and VMS workstations

Local Area Network (Switched As Necessary)

- Fast Ethernet (100baseT)
- Alternative Networks – Standard Ethernet (10BaseT), FDDI, or ATM

Software Architecture





Software Architecture

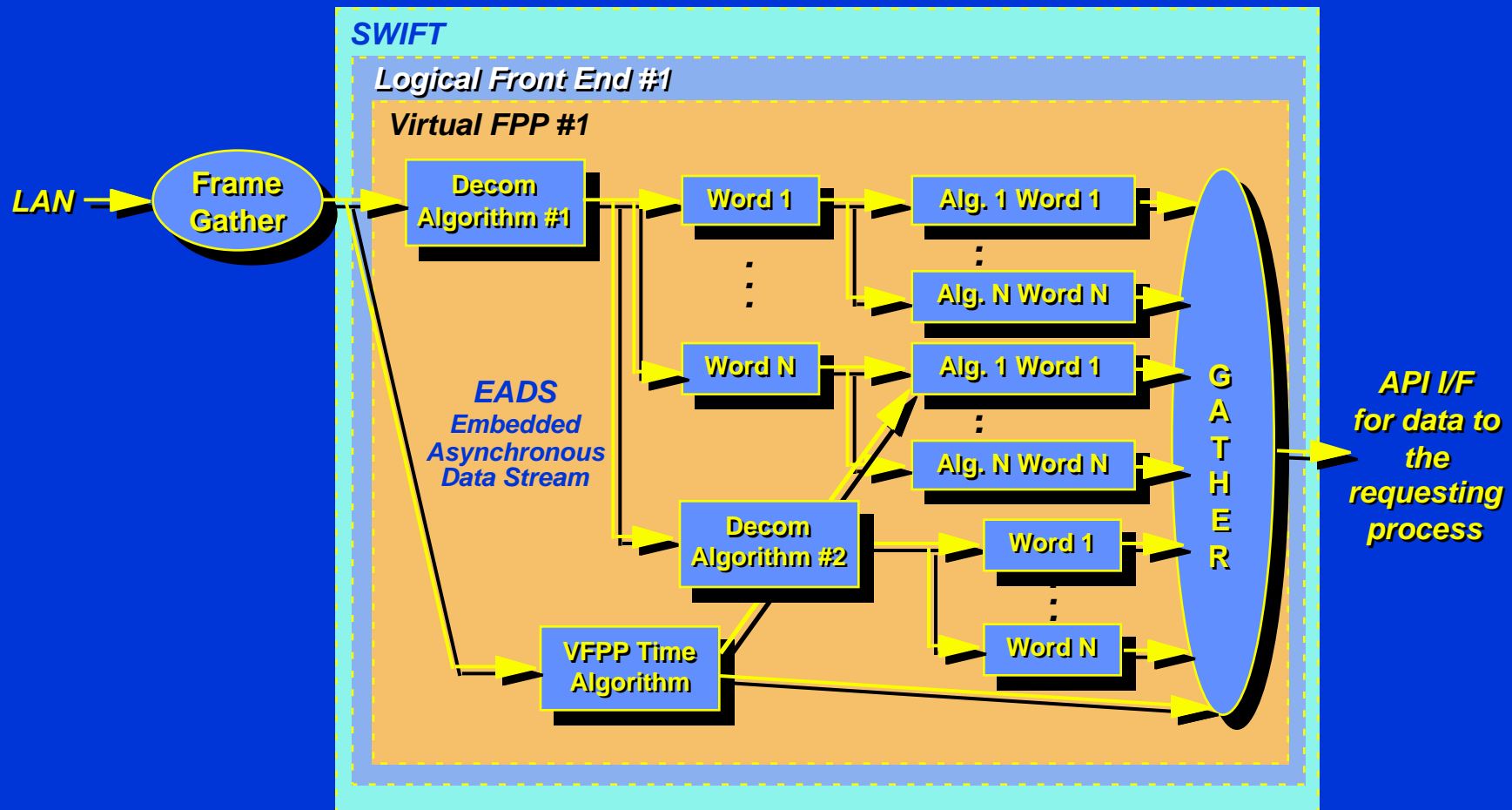
- **Core of telemetry processing and TFE interface is LMTI's Software Instrumentation for Telemetry (SWIFT)**
- **SWIFT derived from System 500 application software and entails telemetry decommutation, data archival and retrieval, database functions, algorithmic processing of TLM data and data distribution**
- **“Frame Gather” & “Command Write” needed to format data from/to TFE (not required for LMTI TFEs)**



Software Architecture (Cont.)

- **SWIFT's Application Programmer Interface (API) used to connect SWIFT's data services to software bus**
- **Architecture utilizes COTS/GOTS software packages for software bus, data visualization, mission planning, database management, scheduling, orbit analysis, etc.**
- **WAN/Internet access of telemetry data available if desired/allowed**

Telemetry Processing





Telemetry Processing

- **“Frame Gather” glueware provides SWIFT with time-tagged raw (frame-synched) or decommutated data**
- **Major SWIFT element called Virtual Field Programmable Processor (VFPP) provides decommutation (if needed), format switching algorithmic processing (i.e., limit checking, EU conversion, derived measurements, etc.), data compression and formatting of data prior to distribution to S/W bus**

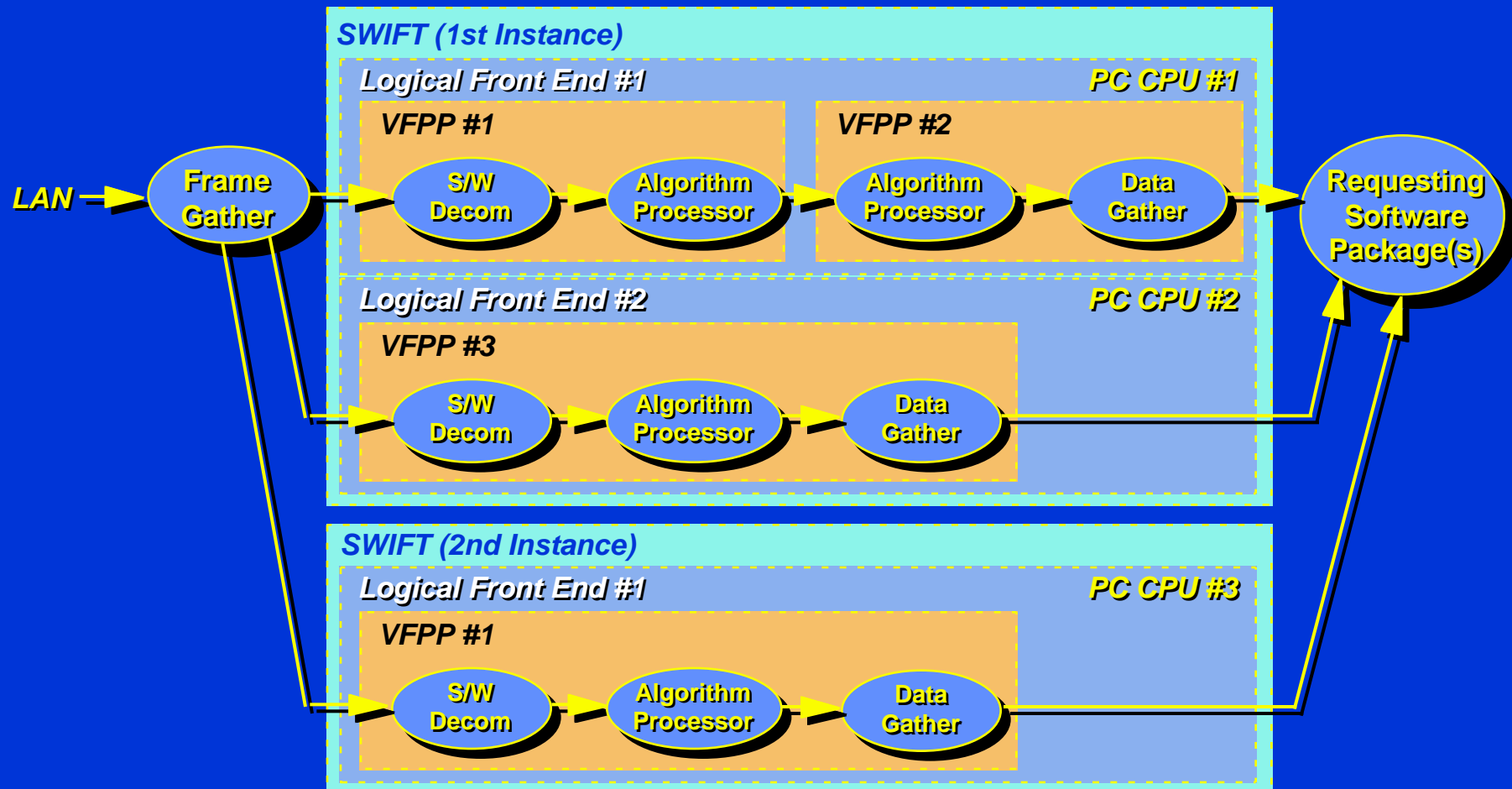


Telemetry Processing (Cont.)

- **Logical Front-End (LFE) encapsulates functionality (database setup) of one or more VFPPs**
- **Application Programmer I/F (API) package allows bi-directional data transfers to/from VFPP**
- **Each instance of SWIFT accommodates 64K uniquely-tagged measurands (raw telemetry, EU-converted data, etc.)**
- **Multiple instances of SWIFT can be active on single workstation providing unlimited tag space**



Telemetry Processing, (cont'd.)



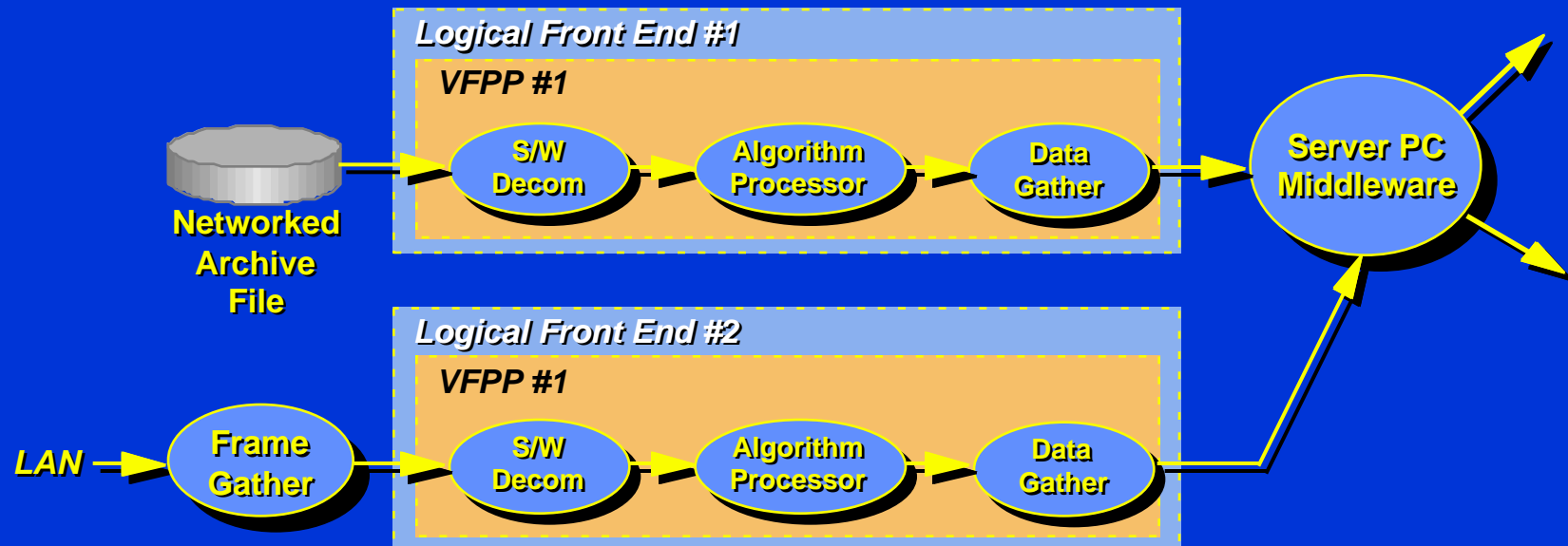


Telemetry Processing (Cont.)

- **Multiple instances of Logical Front Ends (LFEs) in an instance of SWIFT allow independent satellite contact supports (both real-time & playback)**
- **Multiple Virtual FPPs in an instance of an LFE allows logical separation of embedded asynchronous data stream processing or multi-level packet processing tasks**
- **LFEs or VFPPs can be assigned to specific workstation CPUs to provide for CPU load balancing**

Telemetry Processing

Playback & Real Time



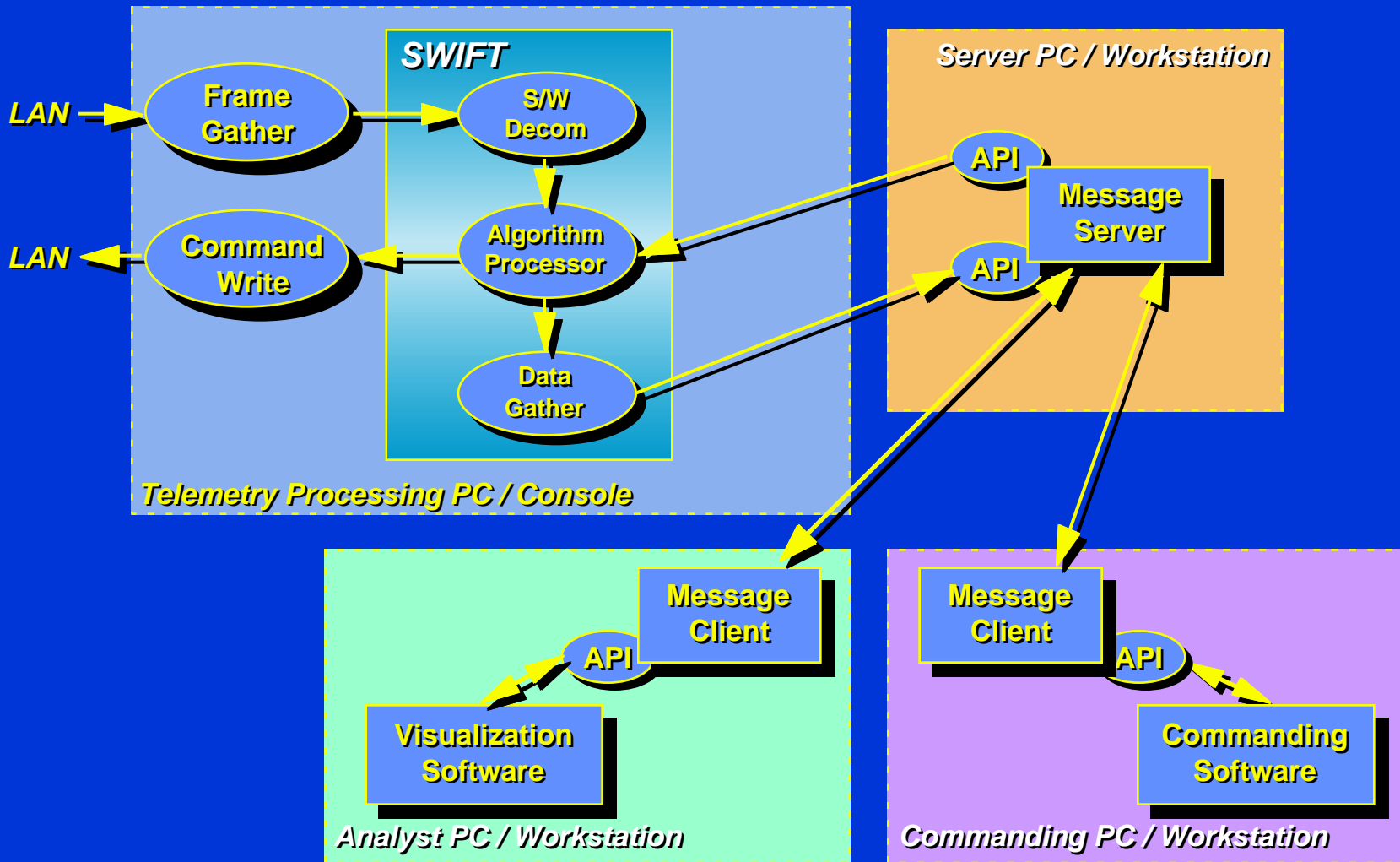
Telemetry Processing

Playback & Real Time



- **Multiple Logical Front Ends (LFEs) allow simultaneous processing of playback and real-time data**
- **Enables trend analysis and measurement comparisons concurrent with real-time operations**
- **Post-analysis tools augmented with VFPP algorithm library**
- **VFPP algorithm library can be enhanced by system integrator or system operator using SWIFT's Algorithm Development Software**

Software Bus Interface

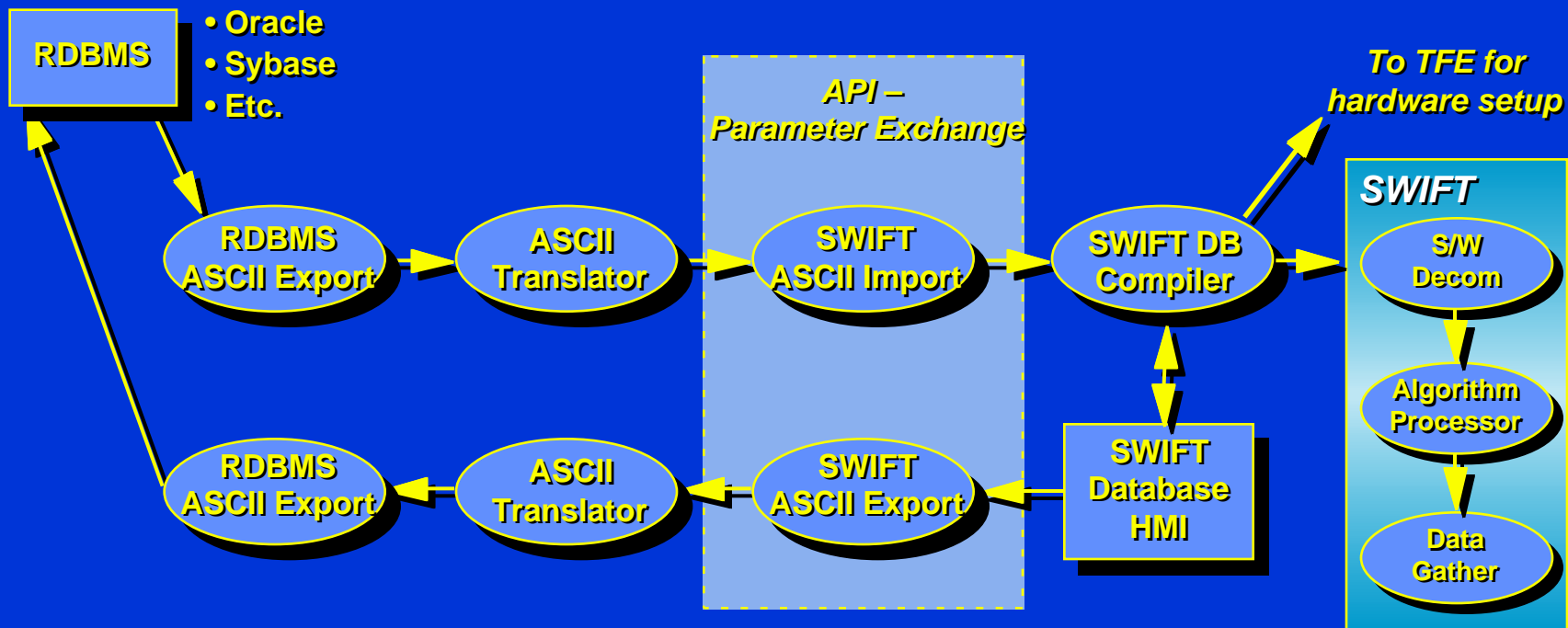




Software Bus Interface

- **SWIFT's Applications Programmer Interface (API) supplies data from/to telemetry processor to/from external S/W processes**
- **Software bus is selected by system integrator to fit the mission's unique cost/performance requirements**
- **SWIFT's API can be used as a s/w bus for real-time ground station applications (i.e., telemetry, commanding, & data visualization)**

Relational Database Interface

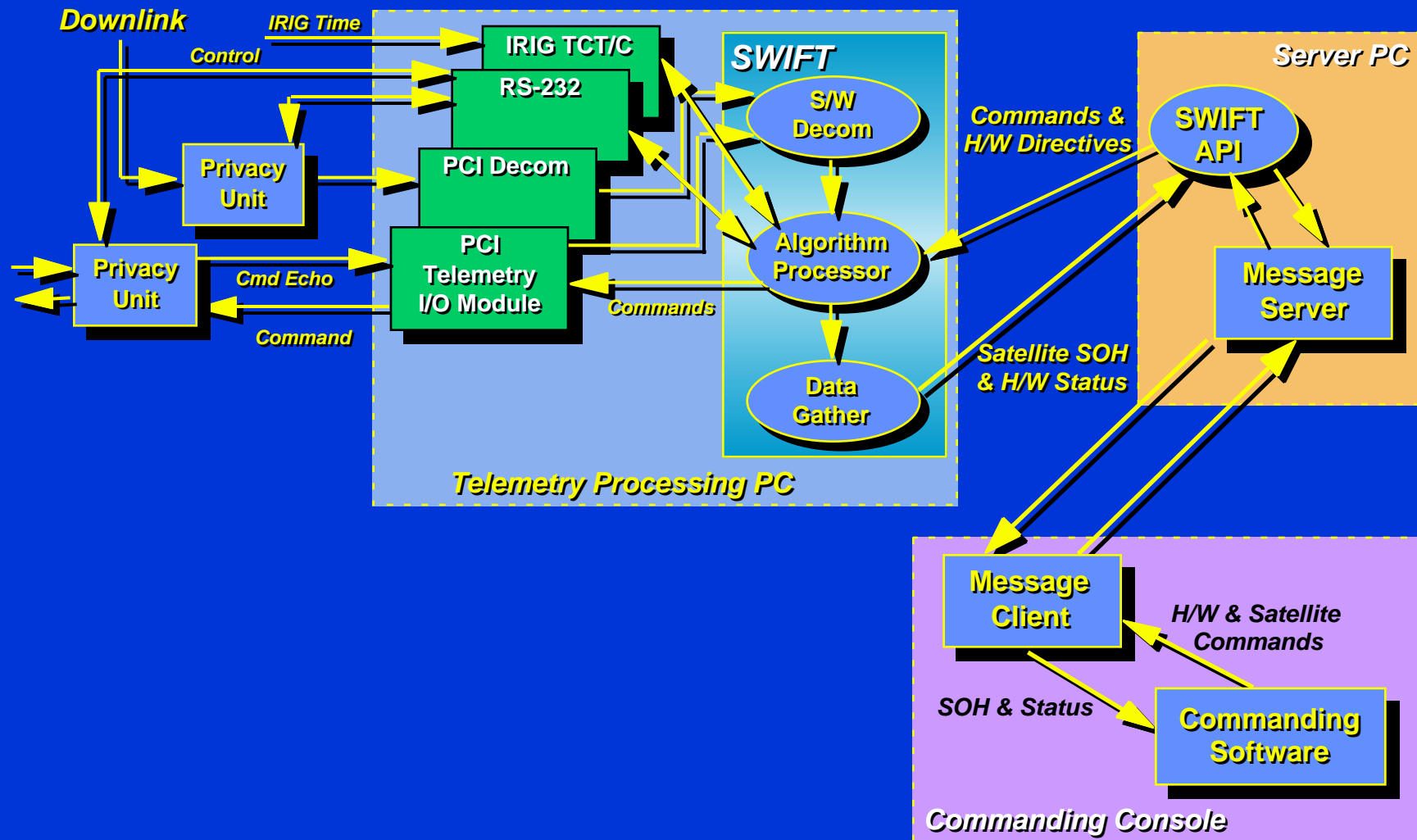


Relational Database Interface



- **SWIFT's API includes "Parm Exchange" utility**
- **Parm Exchange imports/exports ASCII flat files compatible with SWIFT database compiler**
- **Database management can occur exclusively in RDBMS or using a combination of RDBMS plus SWIFT's database HMI**
- **ASCII hardware setups included in SWIFT's API if using LMTI Front-End**

Satellite Commanding

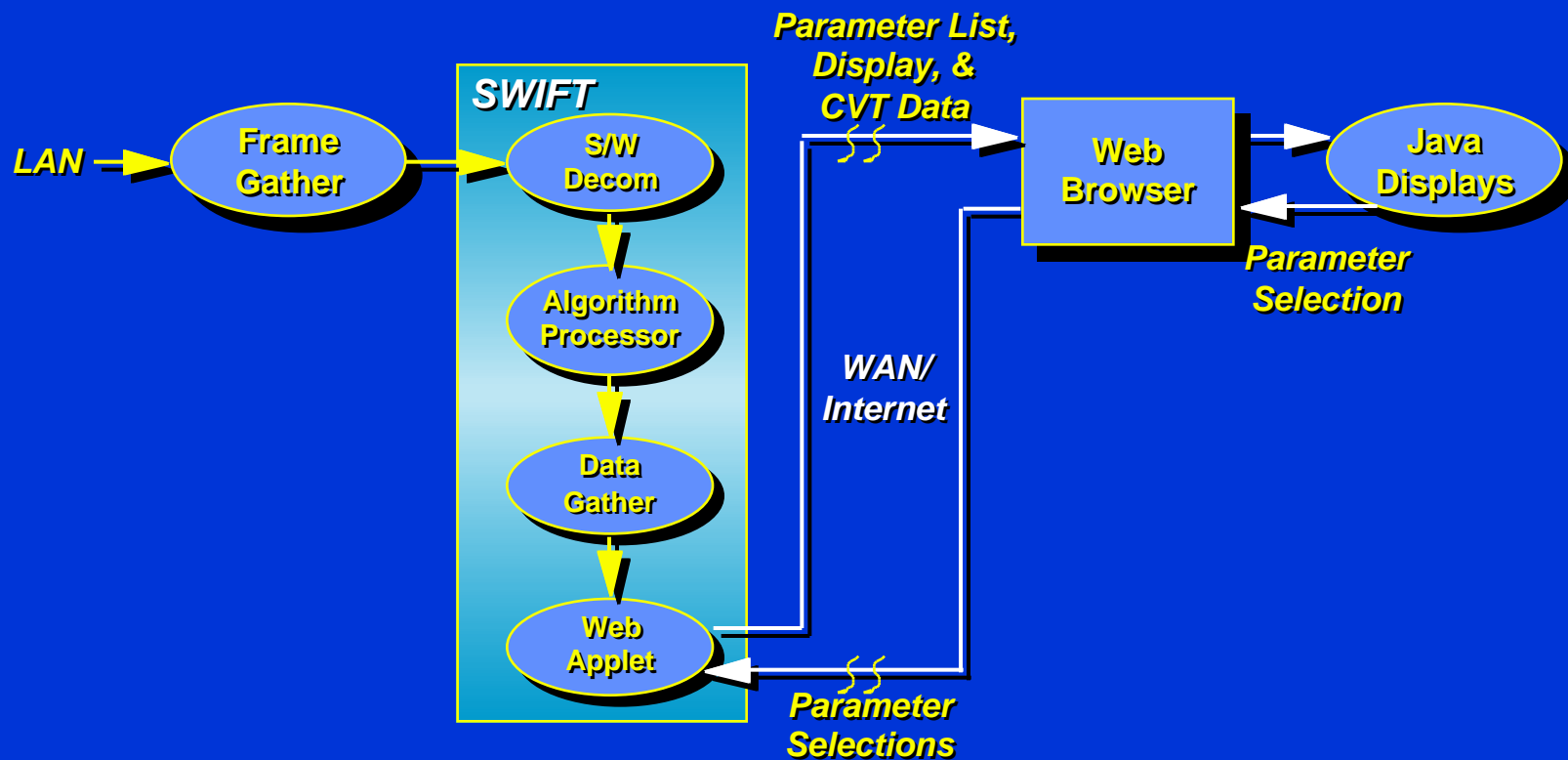




Satellite Commanding

- **NeTstar or similar TFE for commanding hardware interfaces**
- **SWIFT performs satellite SOH downlink processing, command echo processing, command formatting (if needed), and time-critical command releases**
- **Any COTS, GOTS, or “in-house” commanding software package can be tailored for use**

Internet Data Service (Monitor Only)





WAN/Internet Data Service

- **Remote users have access to telemetry data via SWIFT's Web applet**
- **Compatible with Netscape's or Microsoft's Internet Explorer browser**
- **Web applet exports parameter list and Java displays**
- **Security issues may preclude use in all cases**



Summary

- **Software/hardware architecture capable of exploiting 3rd-party TFEs, PC, or UNIX workstations, COTS software buses, COTS/GOTS commanding packages, COTS data visualization packages, etc.**
- **Provides flexible telemetry processing capabilities including frame-synched or decommutated data, multiple simultaneous contacts and concurrent post-analysis**
- **Convenient ground station configuration management with ASCII database interface**



Summary (Cont.)

- **Time-critical commanding capability retained in combined SWIFT and NeTstar (or similar) TFE hardware**
- **Internet access of decommutated/processed telemetry data supported**
- **Architecture populated with low-cost, “small” COTS software and hardware products as selected by system integrator (SI)**



Summary (Cont.)

- **A true COTS product architecture allows SIs to receive product improvements funded by multiple sources and to quickly tailor products to meet specific applications without the direct and expensive efforts of the component vendors themselves**

Questions/Comments



*If you have any questions or comments,
please e-mail me at paul@ti.lmco.com*

Thank you for your time.