
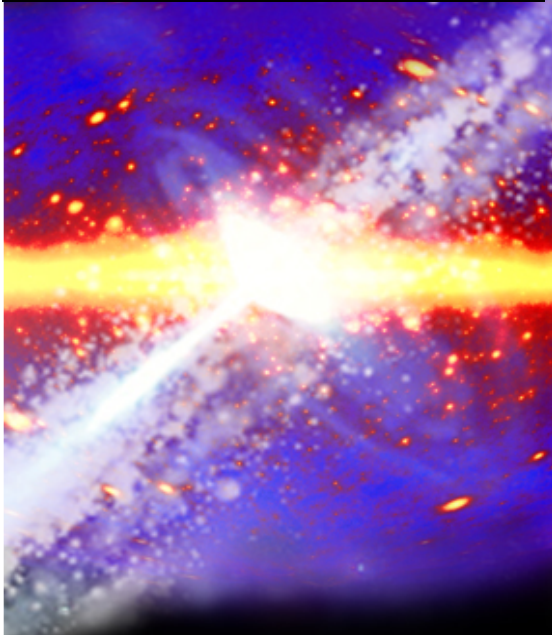


**Gamma-ray Large
Area Space
Telescope**



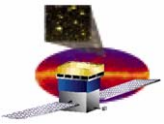
GLAST Large Area Telescope

**Instrument Flight Software
EM2 Design Review
29 January 2004**

Software Architecture

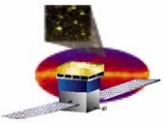
**A.P.Waite
Stanford Linear Accelerator Center**

**apw@slac.stanford.edu
(650) 926-2075**

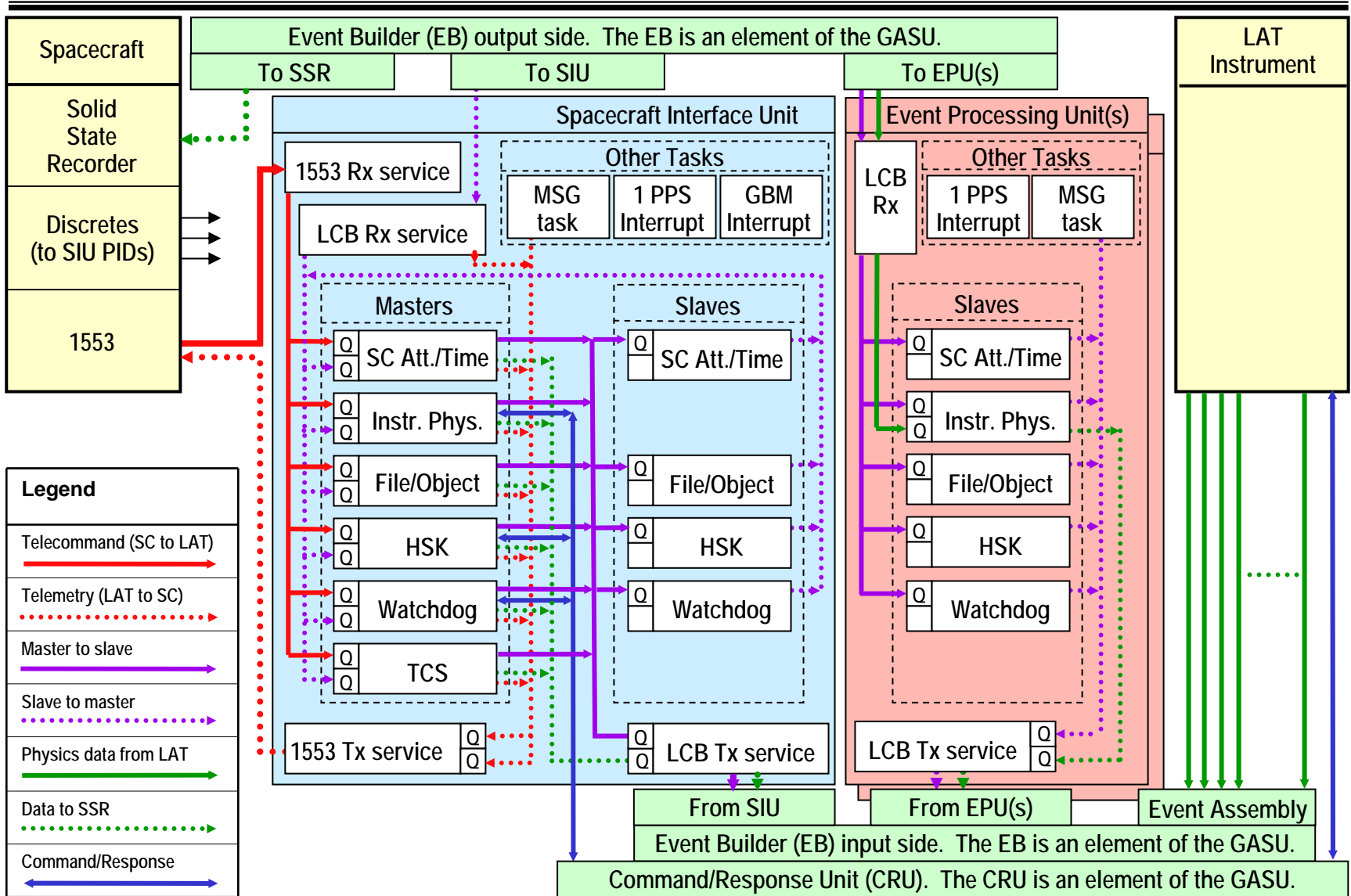


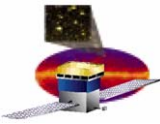
Overview

- Review final system software architecture
- Identify elements of the final architecture needed in EM2
 - Identify “consumers”
 - Identify consumer hardware architectures
 - Identify consumer required software functionality
 - Map functions to consumers and to the final system
- Describe the relationship between functions and packages
- Finding more information
 - The LAT flight software web pages have seen radical improvement in the last few months
 - Start at the flight software [home page](#) and find links to
 - Flight software specific [LATDocs](#)
 - Flight software [package documentation](#) of Application Programming Interface (API) (look under “Doxygen”)
 - Introductory tutorials and navigation aids
 - Telecommand and telemetry packet definitions
 - ...



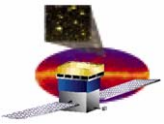
LAT FSW Architecture





Master/Slave Tasks

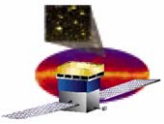
- **Spacecraft Attitude/Time Master/Slave**
 - Master receives seven messages per second from the spacecraft
 - 5 attitude
 - 1 time-tone
 - 1 ancillary (containing orbit information and status info)
 - Master redistributes the messages to slaves on all CPUs
 - Slaves use messages to build time and attitude tables for interrogation by other tasks/functions
 - Attitude table can be interpolated to give spacecraft attitude at requested time
 - Time tone message combined with time hack (also distributed to all CPUs) provides wall clock time throughout system to ~200 nanoseconds
- **Instrument Physics Master/Slave**
 - Master manages and controls the instrument as a physics acquisition device
 - Instrument configuration
 - Instrument activity
 - Physics acquisition
 - Charge injection calibration
 - Subsystem diagnostics (for diagnostics based on the “event path”)
 - Slaves provide the CPU horsepower for compute intensive tasks
 - Physics data filtering
 - Charge injection calibration
- **File/Object Master/Slave**
 - Master receives requests for operations on the RAM and EEPROM based file systems
 - File upload, dump, copy, delete, ...
 - File system interrogation (directory dumps, file checksum checks, ...)
 - Master redistributes the requests to the slave on the targeted CPU
 - Slave implements the operation



Master/Slave Tasks


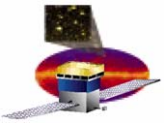
- **Housekeeping Master/Slave**
 - Master is responsible for interrogating the instrument housekeeping systems (currents, voltages, counters, ...)
 - Master assembles housekeeping packets to go into 1553 data stream
 - Slaves provide packets of CPU metrics for the master to put into housekeeping stream
- **Watchdog Master/Slave**
 - Master receives and redistributes to targeted slave, requests for software watchdog change-of-state requests
 - Slave implements software watchdog function (interrogate registered tasks for progress)
 - In passive mode: perform watchdog function but always refresh the hardware watchdog
 - In active mode: perform watchdog function but only refresh the hardware watchdog if all subsystems report progress, otherwise perform warm boot
- **Thermal Control System (Master only)**
 - Interrogate temperature sensors
 - Apply Lockheed provided algorithms to determine heater switch positions for next cycle

 - NOTE: It is the responsibility of LAT FSW to maintain the LAT instrument temperature within a narrow range to benefit the physics analysis (mechanical stability of the instrument and electrical stability of the read-out electronics). It is the responsibility of the *spacecraft* to maintain the LAT instrument temperature within (much wider) survival limits.


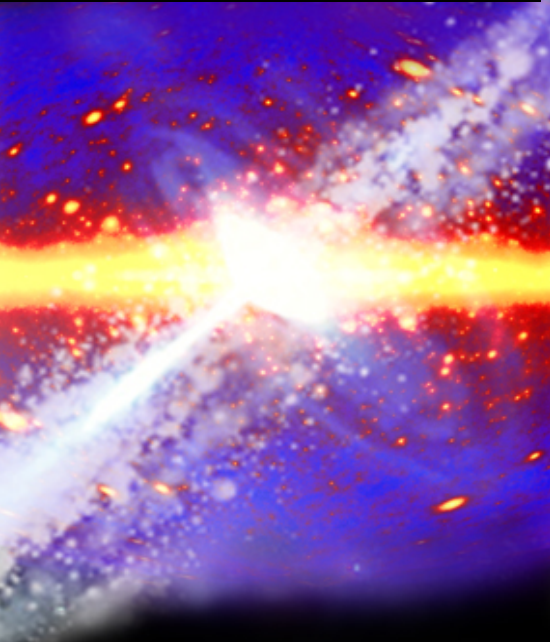


Other Tasks

- **1553 Rx**
 - A service level task/driver to receive 1553 telecommands from the spacecraft
 - Received telecommands are dispatched by APID to master tasks
- **1553 Tx**
 - A task to service requests to send CCSDS/1553 telemetry to the spacecraft
 - Implements two priority encoded queues
- **LCB Rx**
 - A service level task/driver to receive LAT internal protocol packets from the Event Builder Module (part of the GASU)
 - Packets may originate from another CPU or from the instrument
 - Packets are dispatched according to LAT internal protocol
- **LCB Tx**
 - A task to service requests to send packets to one of:
 - Another CPU in the system (LAT internal message protocol)
 - The Solid State Recorder (SSR) on the spacecraft (CCSDS packets preceded by framing word)
- **MSG**
 - General purpose message handling facility
 - Any task can use MSG to format and queue a message for forwarding to the spacecraft
 - The message may be solicited (“command foo completed successfully”)
 - The message may be unsolicited (“temperature bar exceeded yellow high limit”)
- **1 PPS (Pulse Per Second)**
 - Task is woken by the reception of the spacecraft 1 PPS time hack
 - Task works cooperatively with the Spacecraft Attitude/Time slave task (shared memory structure) to build time interpolation tables
- **GBM (Gamma-ray Burst Monitor)**
 - Task is woken by an interrupt from the GBM (the other instrument on GLAST)
 - Task responds by sending a message to the Instrument Physics master

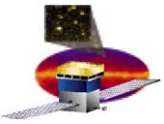


**Gamma-ray Large
Area Space
Telescope**



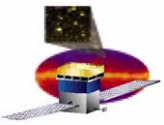
The image is a vertical banner for the GLAST mission. At the top is the stylized 'GLAST' logo. Below it, the text 'Gamma-ray Large Area Space Telescope' is written in blue. The main part of the banner is a colorful image of the cosmic microwave background, showing a bright yellow and orange band across the center. At the bottom, there is a row of flags representing the participating countries: France, Germany, Italy, Japan, Sweden, and the United States, followed by the European Space Agency (ESA) logo and the NASA logo.

EM2 Consumers

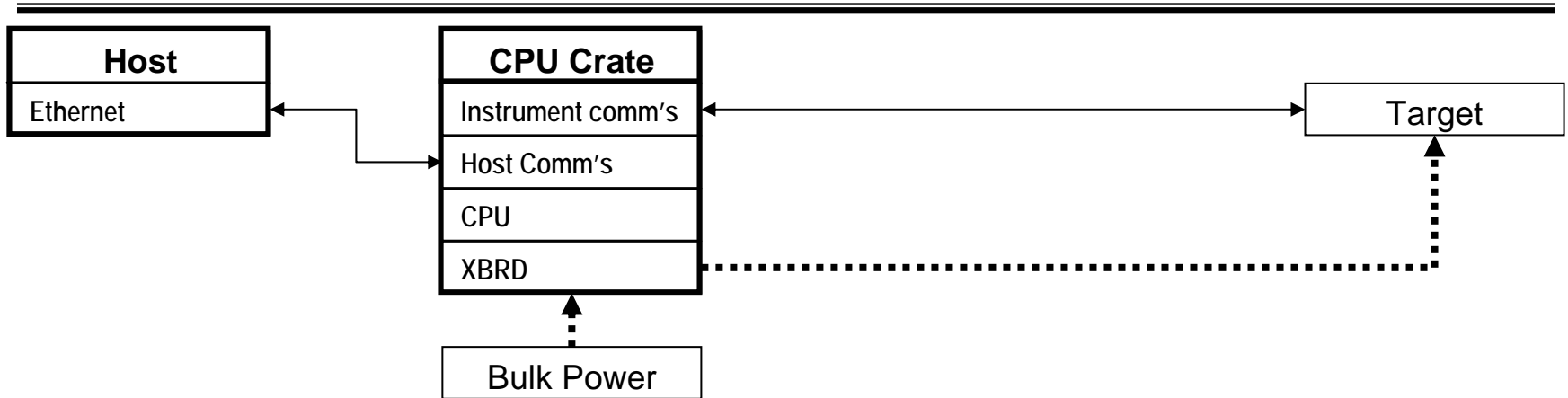


EM2 Consumers

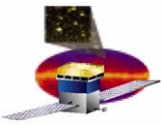
- **Field test stands for ACD, CAL, TKR & DAQ**
 - **Functionality to test subsystems**
 - **Configuration**
 - **Event read-out**
 - **ACD has special hardware needs, being dependent on**
 - **All elements of the GASU**
 - **CRU, AEM, GEM, EBM**
 - **Through CRU/AEM, access to**
 - **ACD front end electronics (GAFE, GARC)**
 - **Field maintenance is the responsibility of LAT I&T**
 - **FSW responsible for providing embedded system software only**
- **Spectrum Astro**
 - **Instrument Spacecraft Interface Simulator (ISIS)**
- **I&T**
 - **Building up the instrument**
- **Instrument Flight Software (ourselves)**
 - **Development of instrument flight software final deliverable**
 - **Development done on the test-bed**
 - **A full flight-like DAQ system (except sensors)**
 - **Sensors emulated by Front End Simulators (FES)**



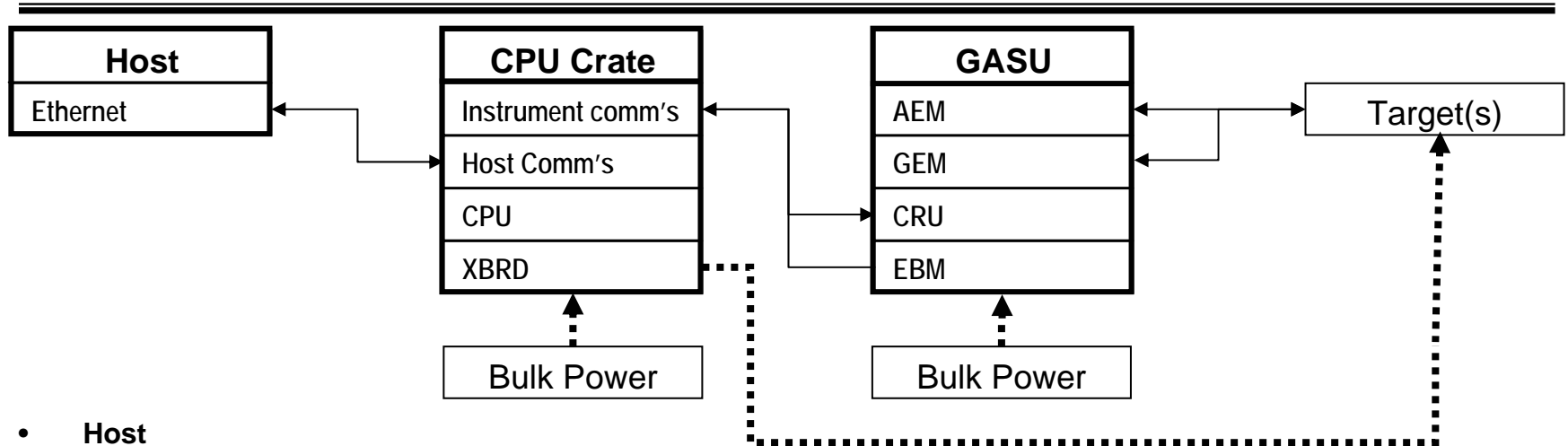
TKR or CAL Test Stand Hardware Configuration



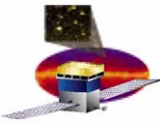
- **Host**
 - Wintel PC running I&T LATTE software
 - Ethernet communications (both command/response and event acquisition)
- **CPU Crate**
 - VME chassis
 - Instrument communications: LAT communications board (LCB) (PMC form factor)
 - Host communications: Ethernet (built in to COTS CPU)
 - CPU: Motorola MV2304 COTS processor
 - XBRD: Minimal emulations of the trigger function and the power distribution function
- **GASU**
 - None
- **Power Distribution**
 - Transition board (XBRD)
- **Target(s)**
 - Sensors under test, read by a Tower Electronics Module (TEM)



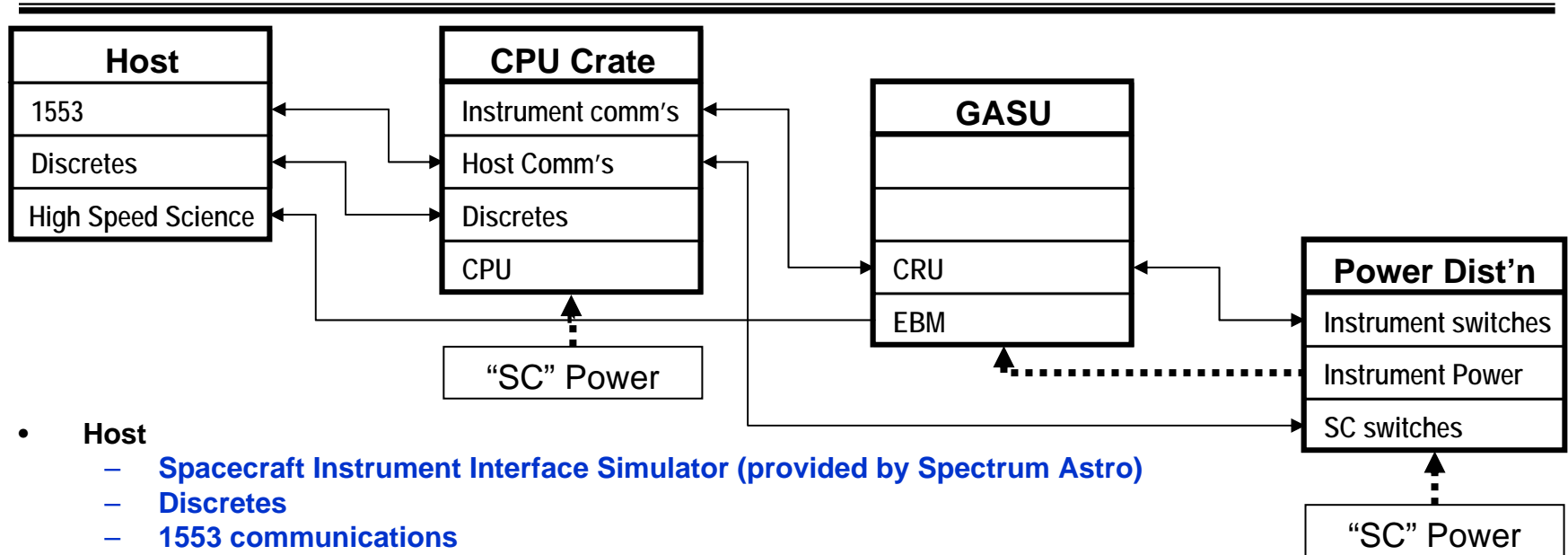
ACD Test Stand Hardware Configuration



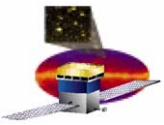
- **Host**
 - Wintel PC running I&T LATTE software
 - Ethernet communications (both command/response and event acquisition)
- **CPU Crate**
 - VME chassis
 - Instrument communications: LAT communications board (LCB) (PMC form factor)
 - Host communications: Ethernet (built in to COTS CPU)
 - CPU: Motorola MV2304 COTS processor
 - XBRD: Emulation of the power distribution function
- **GASU**
 - Full GASU implementation
 - AEM, GEM, CRU, EBM
 - Primary and redundant sides
- **Power Distribution**
 - Transition board (XBRD) to target hardware
 - Simple bulk 28V supply for GASU
- **Target(s)**
 - Sensors under test, triggered by GEM and read back by FREE board(s) via the GASU AEM unit



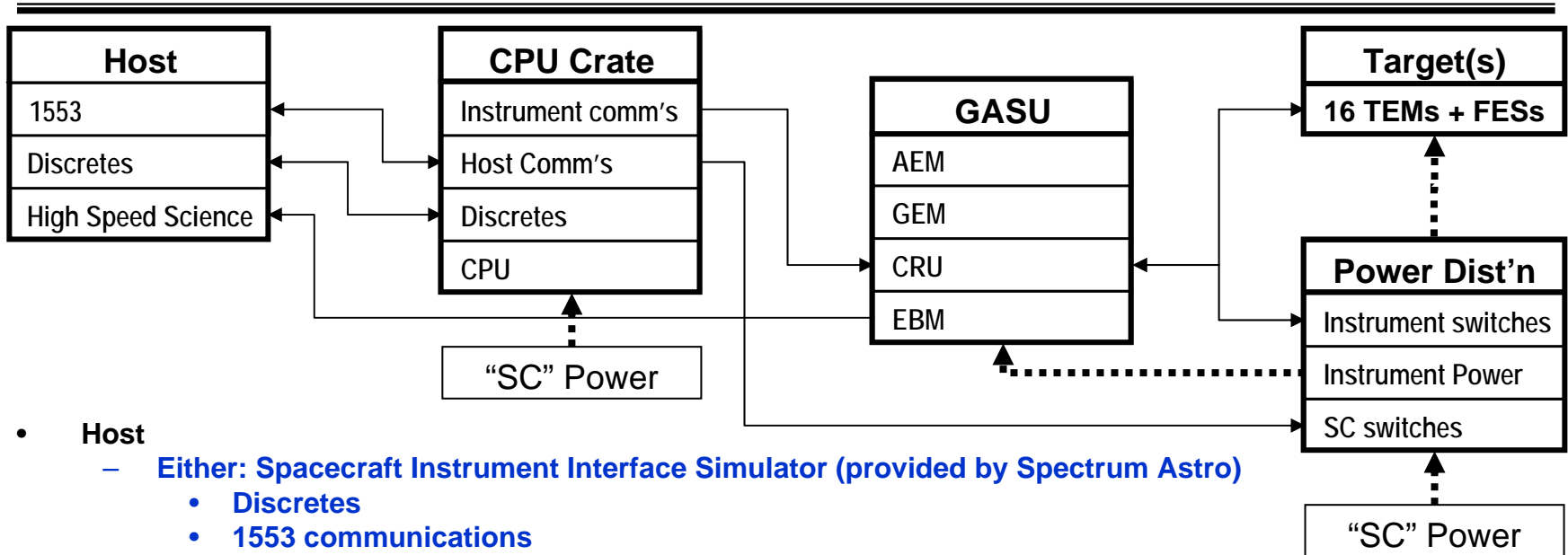
ISIS Hardware Configuration



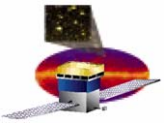
- **Host**
 - Spacecraft Instrument Interface Simulator (provided by Spectrum Astro)
 - Discretes
 - 1553 communications
 - High speed science interface
- **CPU Crate**
 - cPCI flight equivalent chassis
 - Instrument communications: LAT communications board (LCB) (cPCI form factor)
 - Host communications: Storage and Interface Board (SIB) (also does power bootstrap)
 - CPU: RAD750 processor
- **GASU (GASU-lite) (primary side only)**
 - Command/response unit (CRU)
 - Event Builder Module (EBM)
- **Power Distribution**
 - Full PDU implementation (primary and redundant) but with dummy loads for most instrument targets
- **Target(s)**
 - None



Test-Bed Hardware Configuration

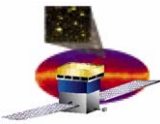


- **Host**
 - Either: Spacecraft Instrument Interface Simulator (provided by Spectrum Astro)
 - Discretes
 - 1553 communications
 - High speed science interface
 - Or: Ethernet simulation
- **CPU Crate**
 - cPCI flight equivalent chassis or commercial cPCI chassis
 - Instrument communications: LAT communications board (LCB) (cPCI form factor)
 - Host communications: Storage and Interface Board (SIB) (also does power bootstrap)
 - CPU: RAD750 processor or Motorola MCP750 COTS processor
- **GASU**
 - Full GASU implementation (including primary and redundant sides)
- **Power Distribution**
 - Full PDU implementation (including both primary and redundant sides)
- **Target(s)**
 - 16 Tower Electronics Modules (backed by Front End Simulators)



Functions to Support Consumers


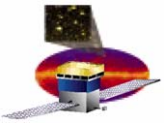
- **Functions evident in previous hardware diagrams**
 - **Configure TKR and CAL front end electronics**
 - **Configure ACD front end electronics**
 - **Configure GASU (CRU, GEM, EBM, AEM)**
 - **Configure PDU**
 - **Configure XBRD**
 - **Configuration by compressed file**
 - **Real event delivery to CPU**
 - **Housekeeping data stream**
 - **RAD750 boot and crate initialization**
 - **1553 bus communications**
 - **Command/Telemetry database and services**
 - **Emulated event delivery across science data interface**
- **Derived functions internal to instrument flight software**
 - **CPU internal communications/task frameworks**
 - **Software watchdog**
 - **Wall clock time services (GPS)**
 -
 -




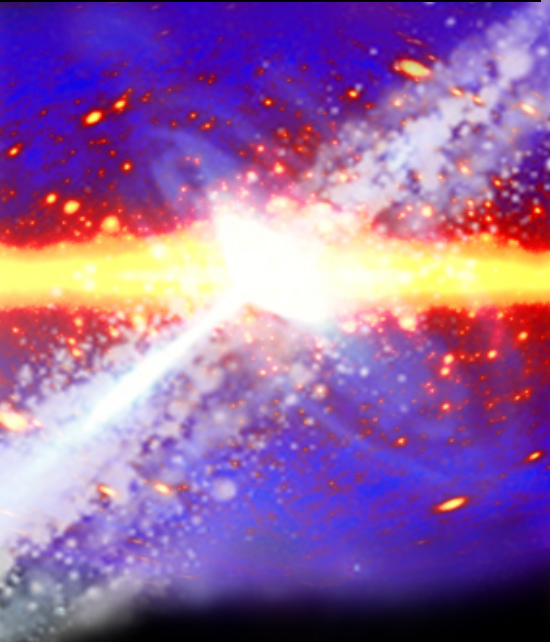
Function/Consumer Mapping

	Test Stands		Spectrum Astro	I&T	FSW Test-bed	FSW Deliverable
	TKR,CAL	ACD				
Configure TKR and CAL front end electronics	Y			Y	Y	Y
Configure ACD front end electronics		Y			Y	Y
Configure GASU (CRU, GEM, EBM, AEM)		Y	Y	Y	Y	Y
Configure PDU			Y	Y	Y	Y
Configure XBRD	Y	Y		Y		
Configure by compressed file			Y	Y	Y	Y
Real event delivery (instrument to CPU)	Y	Y		Y	Y	Y
Housekeeping data stream	Y	Y	Y	Y	Y	Y
RAD750 boot and crate initialization			Y	Y	Y	Y
1553 bus communications			Y	Y	Y	Y
Telecommand/telemetry database and services			Y	Y	Y	Y
Emulated event delivery (to science data interface)			Y			
CPU internal communications/task frameworks			Y	Y	Y	Y
Software watchdog			Y	Y	Y	Y
Wall clock time services (GPS)			Y	Y	Y	Y

- **Only two specialized functions (not in final code base)**
 - **Configure XBRD (XBRD is a pure EGSE board)**
 - **Emulate event delivery (Spectrum Astro interested in rate/protocol/integrity, not physics data)**

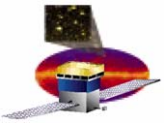


**Gamma-ray Large
Area Space
Telescope**



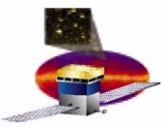
The image features the GLAST logo at the top, followed by the text "Gamma-ray Large Area Space Telescope" in blue. Below this is a large, vibrant image of a gamma-ray background, showing a bright yellow and orange band across the center, surrounded by a field of blue and red points. At the bottom, there is a row of flags representing the international partners: France, Germany, Italy, Japan, Sweden, and the United States, followed by the ESA and NASA logos.

Functions & Packages



Functions & Packages

- Functions are implemented with a collection of software “packages”
- Packages contain the source code and make-like build instructions
- Packages obey a hierarchy called the “use tree”
 - Low nodes in the hierarchy provide low level utility functionality
 - Higher nodes provide increasingly sophisticated functionality
 - At the top of the hierarchy are packages that provide the “applications”:
 - Implemented as threads of execution (tasks in the VxWorks lexicon)
 - Queue driven and operating asynchronously
 - Lower level packages export their call interface to higher level packages
 - In CMT-speak, higher level packages “use” lower level packages
- To illustrate, take an example from EM1: configuring TKR, CAL electronics
 - Commands are received (via ethernet) by a task implemented in package OCS (Online Command Server)
 - OCS calls entry points in package DEM to talk to the instrument
 - DEM itself calls entry points in:
 - GNAT (to construct bit streams for, and to drive, COMM I/O boards)
 - VXW (VxWorks operating system)
 - VXW not shown in diagram on the next slide
 - Omitted for clarity ... nearly all packages call the operating system
 - The current package “use tree” (regenerated automatically every night) is at:
 - http://www.slac.stanford.edu/exp/glast/flight/web/FSW_use_tree.shtml
 - Describes public packages only, not packages in initial development



Package Use Tree

