GLAST Large Area Telescope

Instrument Flight Software
EM2 Review
26 February 2004

Software Architecture

J. J. Russell
Stanford Linear Accelerator Center

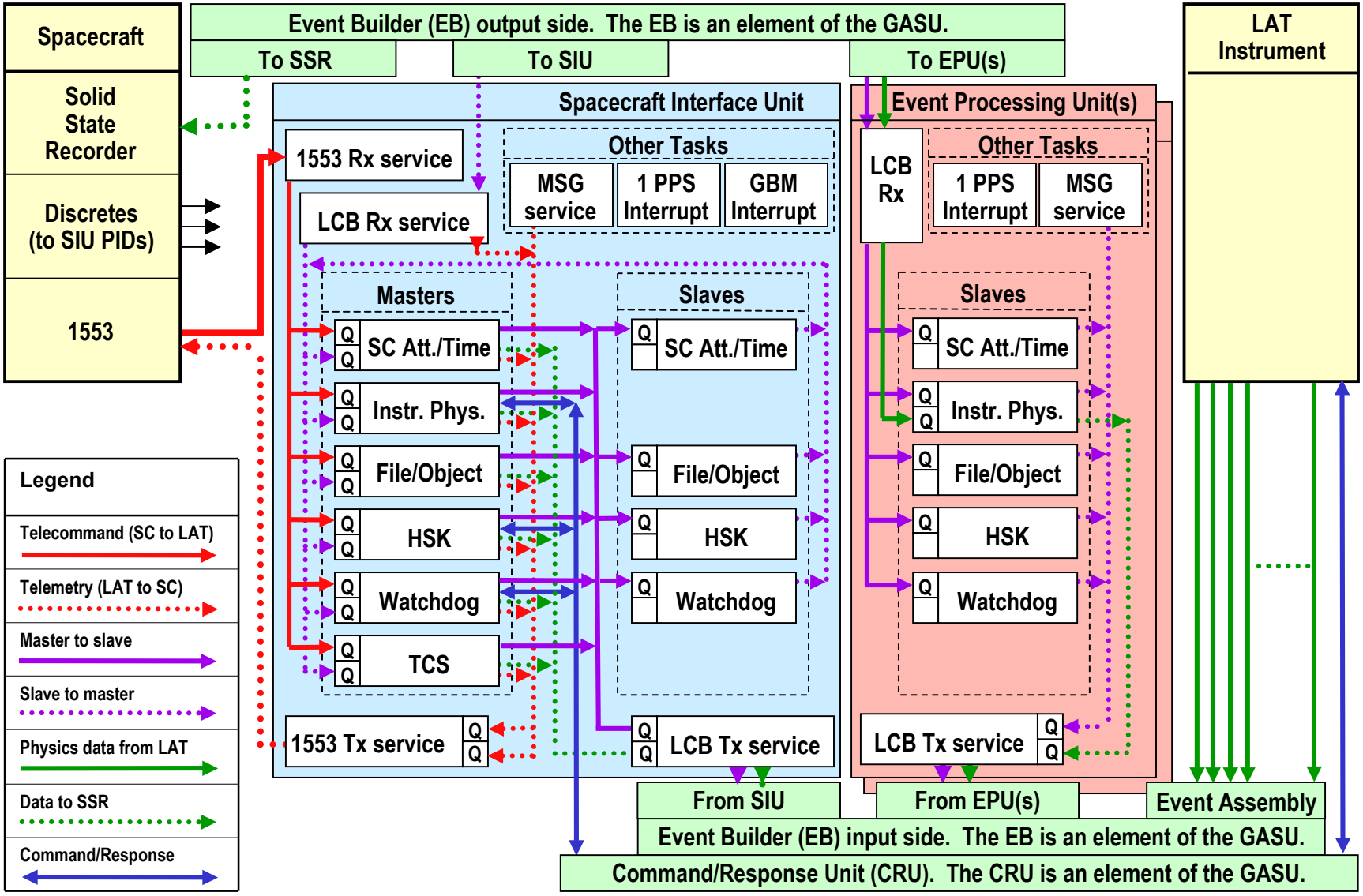
russell@slac.stanford.edu
(650) 926-2583

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

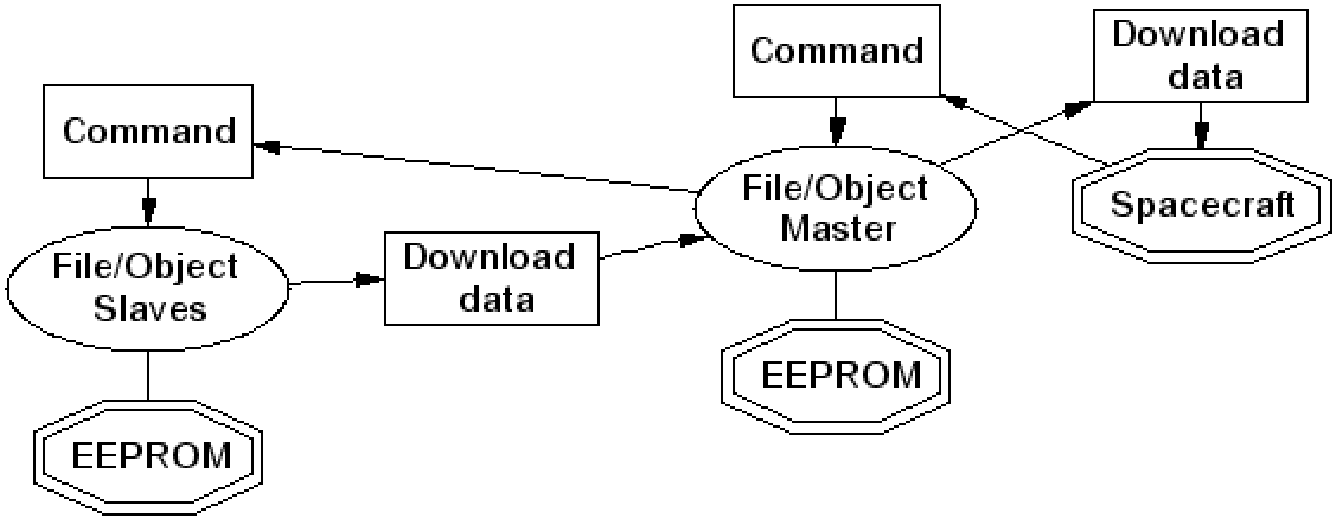


Overview of Tasks

- The flight software runs as a set of VxWorks "tasks"
  - Tasks are analogous to Unix processes, but they share a common address space.
  - In practice, however, FSW tasks communicate with each other via (possibly inter-machine) message queues.
- Flight software tasks can be grouped into two rough categories: Services and Special-Purpose Tasks
  - Services, such as the 1553 Receive Service, are used by many other tasks throughout the system
    - Services act as interfaces to hardware
    - Services are not concerned with the types of data they process; instead, they are defined by the source or destination of the data they process
  - Special-purpose tasks, such as the File/Object task, perform specialized functions.
    - Special-purpose tasks are defined by the type of data they process
- Some flight software functions are divided up between "master" and "slave" tasks.
  - The master tasks (located on the SIU, because of its privileged position) receive, perform higher level verification, and queue commands
  - The slave tasks actually do the work

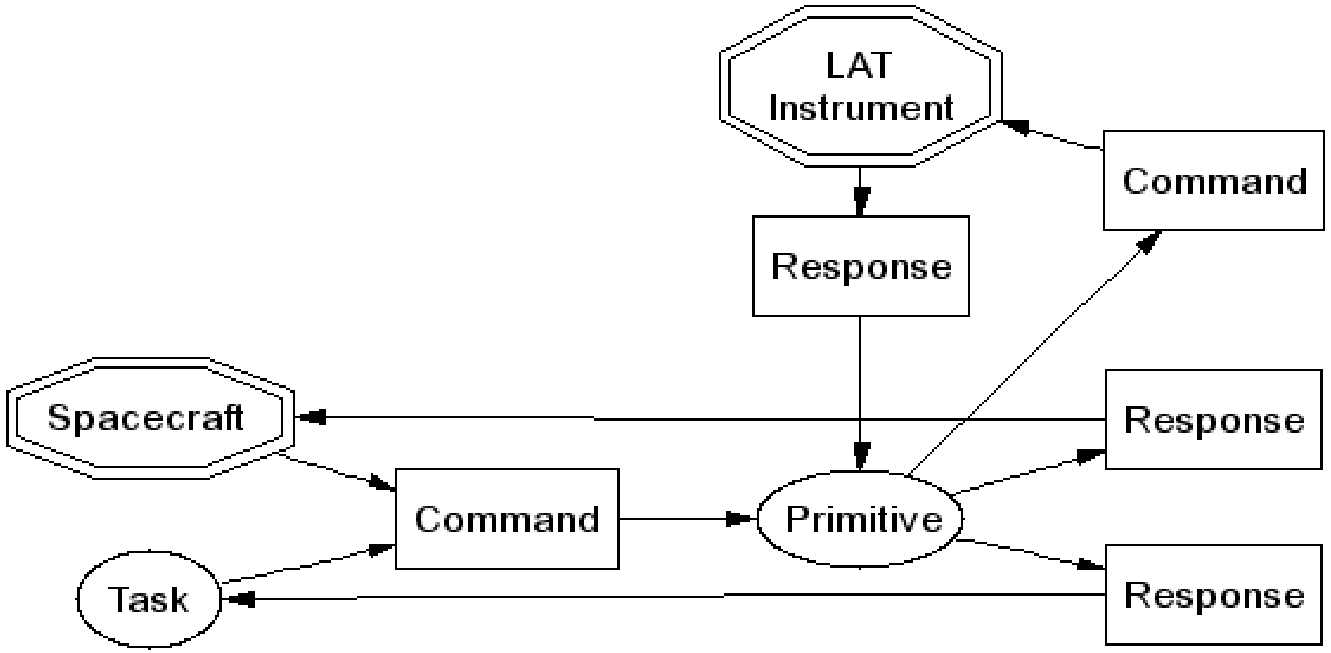
File/Object Master/Slave Tasks

- Master receives requests for operations on the RAM and EEPROM based file systems
  - 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
- Covered in more detail later by Dan Wood



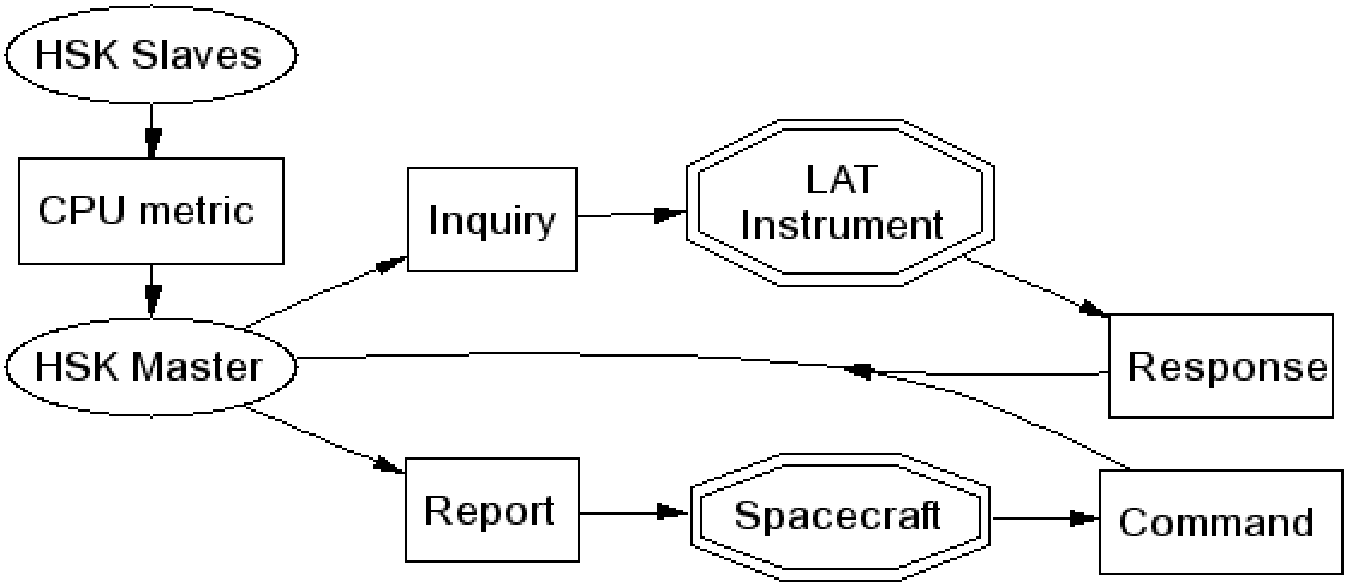
Primitive Master Task

- This task handles "primitive" messages from the spacecraft, providing emergency access to low-level hardware interfaces in the LAT instrument.
- This will covered in more detail later by Curt Brune.
- See LAT-TD-1380 for more information.



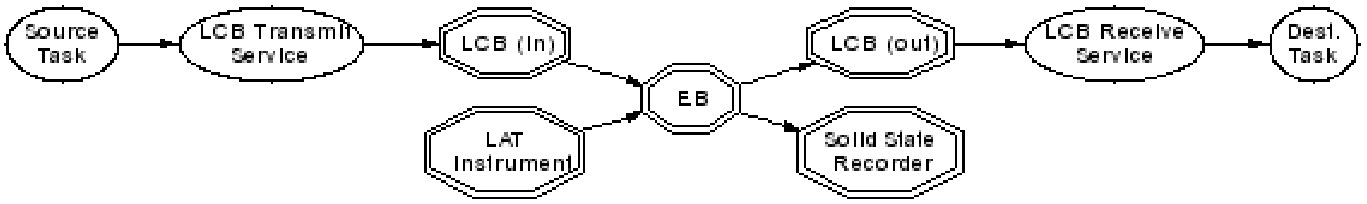
Housekeeping Master/Slave Tasks

- 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
- Covered in more detail later by Sergio Maldonado



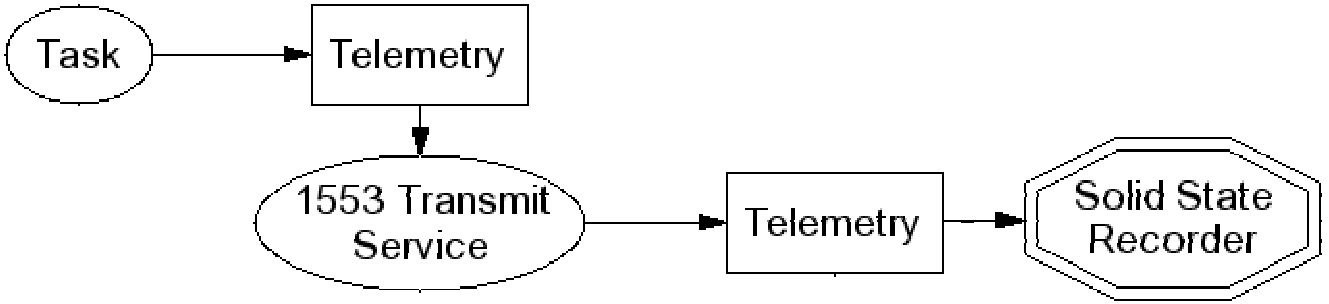
LCB Transmit and Receive Services

- **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)**
- **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**
- **This will be covered later in more detail by J. J. Russell**
- **See LAT-TD-1380**

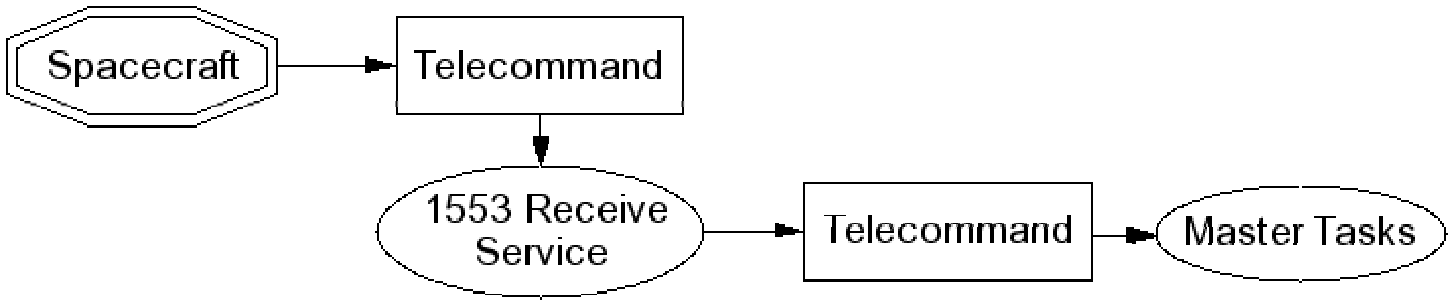


1553 Transmit and Receive Services

- **1553 Tx**
  - **A task to service requests to send CCSDS/1553 telemetry to the spacecraft**

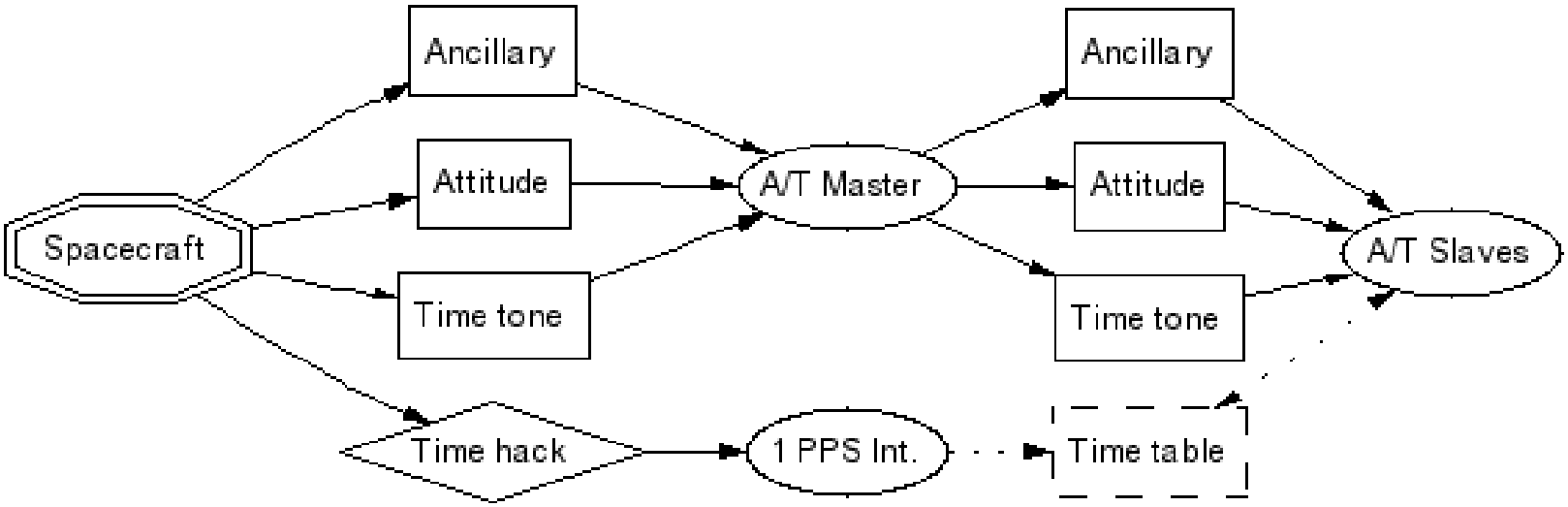


- **1553 Rx**
  - **A task to service requests to receive CCSDS/1553 telemetry to the spacecraft**
- **These wil be covered in more detail by Dan Wood.**
- **Also see CTDB Traveler document on the FSW Web page.**



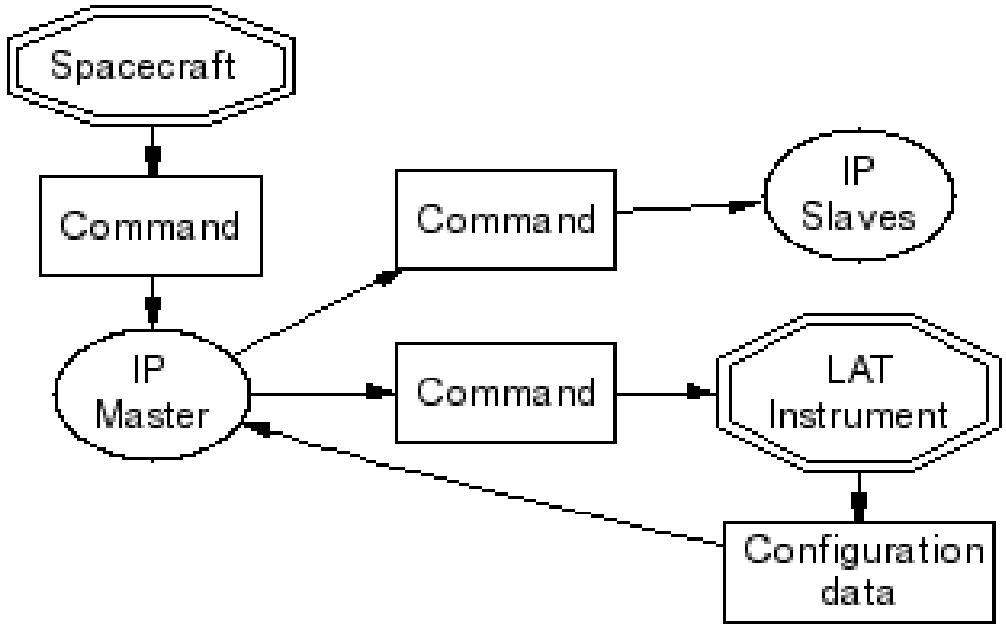
Spacecraft Attitude/Time Master/Slave Tasks

- 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
- Not part of EM2

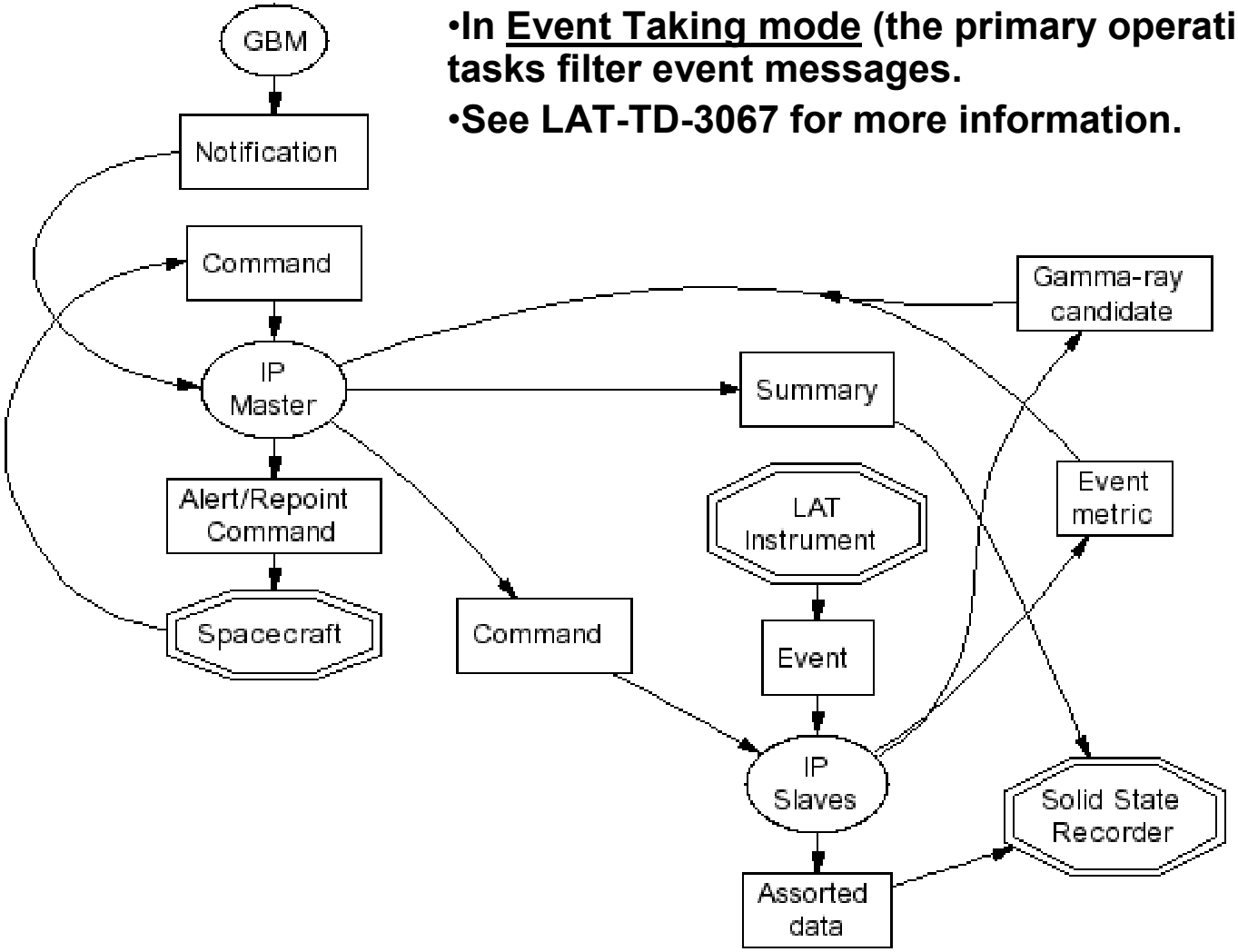


Instrument Physics Master/Slave Tasks

- The master task manages and controls the instrument as a physics-acquisition device.
  - The master controls instrument activity, including physics acquisition, charge-injection calibration, and subsystem diagnostics (for diagnostics based on the "event path").
- The slave tasks provide the CPU horsepower for compute-intensive tasks, including physics data filtering and charge-injection calibration.
- These tasks can operate in any of several modes, including Event Taking, Calibration, Diagnostic, and Idle (quiescent). Before any of these modes can begin, however, the tasks must perform the Configuration phase. The processing path for this phase is illustrated below.
- This will be covered in more detail by Curt Brune and James Swain.



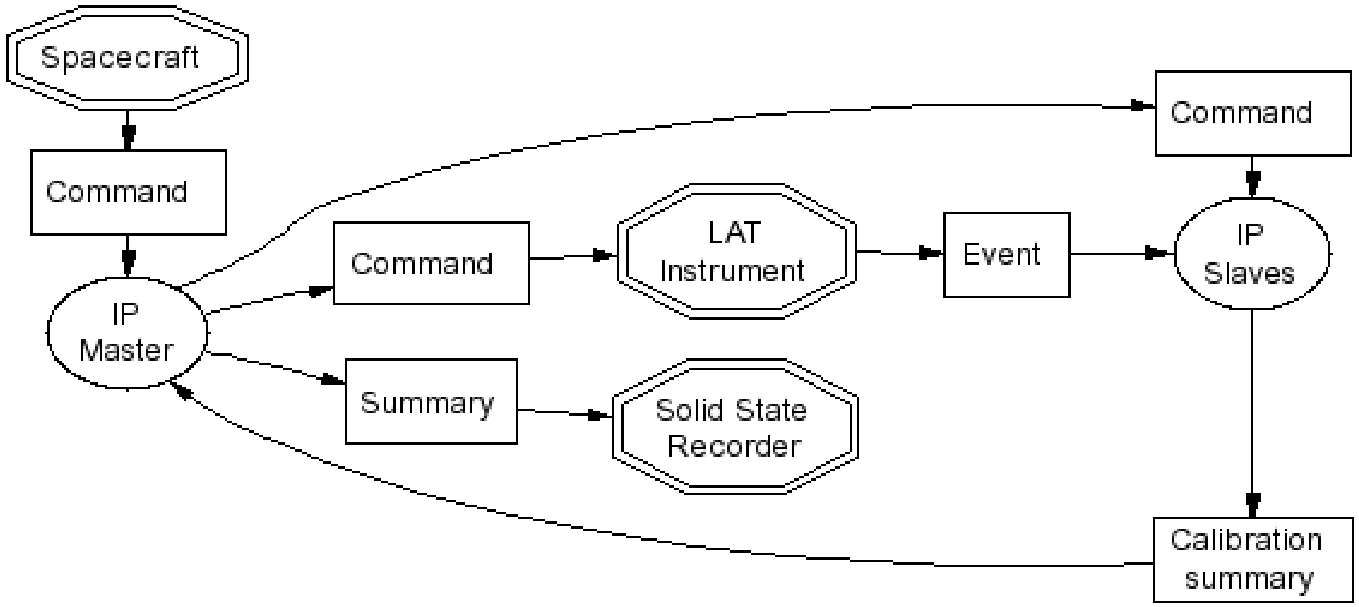
Instrument Physics Task: Event Taking Mode



- In Event Taking mode (the primary operation mode), the tasks filter event messages.
- See LAT-TD-3067 for more information.

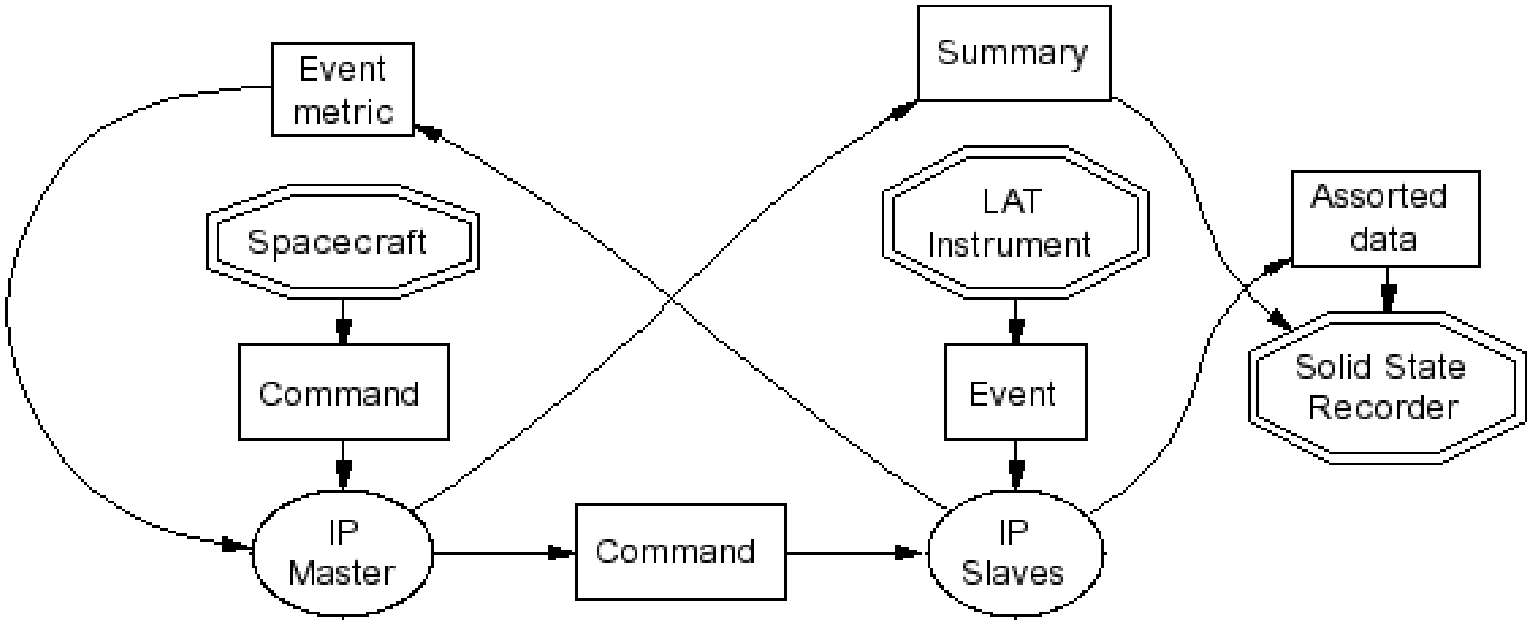
Instrument Physics Task: Calibration Mode

- **Calibration mode is used to collect quantitative data on the LAT instrument (e.g., systematic biases). In this mode, the tasks provide known signals to portions of the instrument, collecting the resulting data.**
- **Not applicable to EM2.**



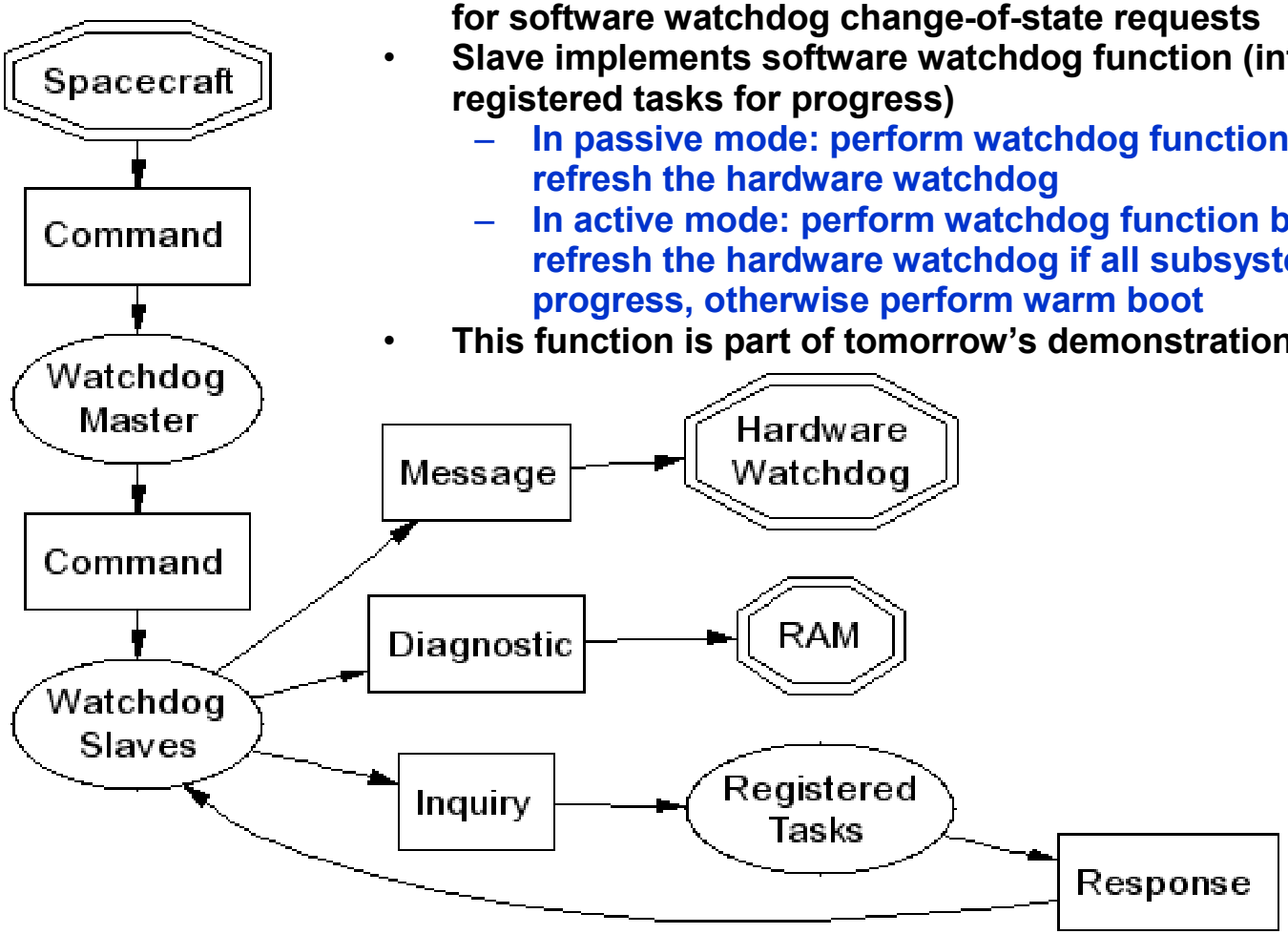
Instrument Physics Task: Diagnostic Mode

- **Diagnostic mode is used to collect qualitative data on the LAT instrument (e.g., component functionality). In this mode, the tasks operate subsets of the instrument, looking for expected behavior.**
- **Not applicable to EM2.**



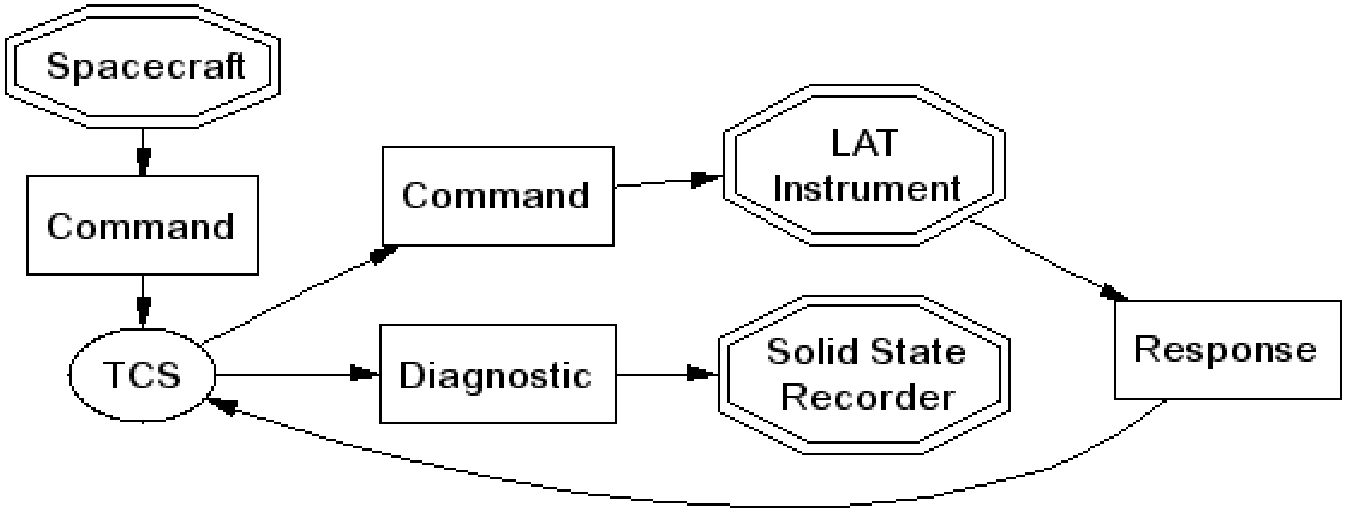
Watchdog Master/Slave Tasks

- 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
- This function is part of tomorrow's demonstration.



Thermal Control System Master Task

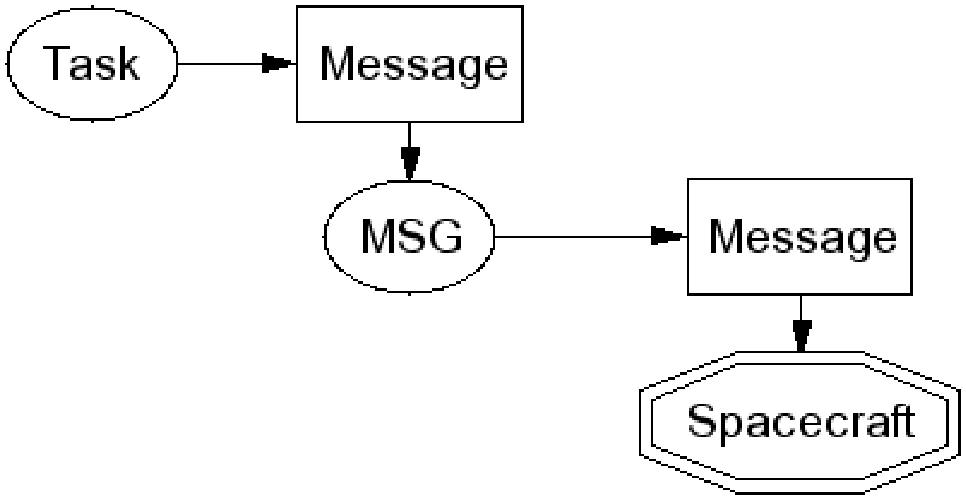
- **Thermal Control System (Master only)**
  - **Interrogate temperature sensors**
  - **Apply Lockheed provided algorithms to determine heater switch positions for next cycle**
  - **See LAT-SS-02896**
  - **Not part of EM2**



–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.

MSG (Messaging) Task

- 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")
  - Consult Traveler document for MSG package.



EM2 Consumers

EM2 Consumers: Test Stands

- Field test stands for ACD, CAL, TKR & DAQ
  - Functionality to test subsystems
    - Configuration
    - Event read-out
    - Housekeeping stream
  - 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

EM2 Customers: I&T

- All the needs of the Test Stands
  - This is where they will be checking out new software to be delivered to the field
- Plus the ability to “Build up the instrument”
  - In practice, this means
    - Supporting Multi-towers
    - A clean interface to the GASU triggering system

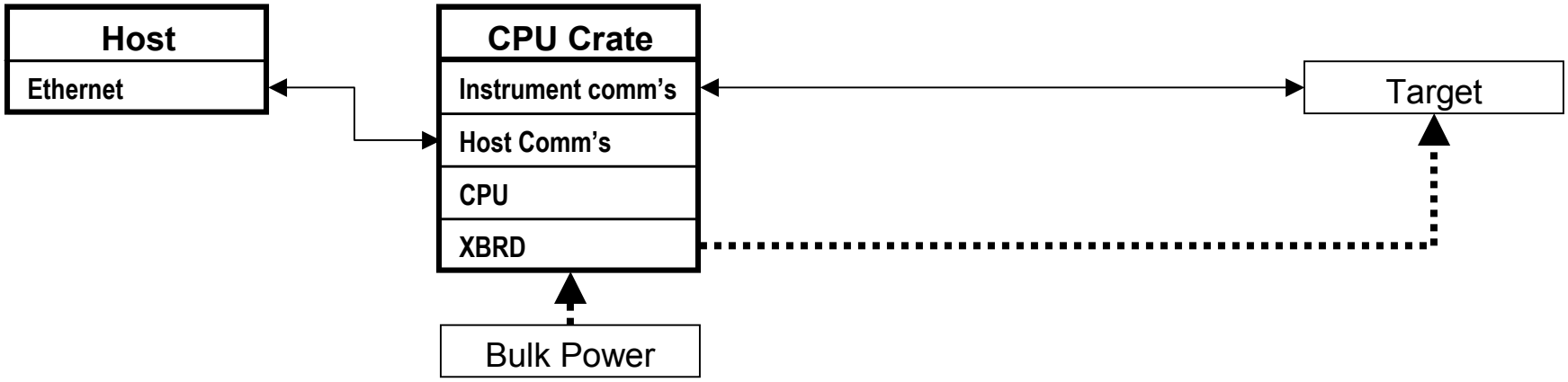
EM2 Customers: ISIS

- SIU Boot Sequence
- PID Interface Testing
  - Both inputs and outputs
- 1553 Interface Testing
  - Exchange of commands and telemetry
  - Keep track of statistics, i.e. count messages
  - Housekeeping stream
- Science Interfaces
  - Soliciting of known test patterns

EM2 Consumers: Flight Software Test-Bed

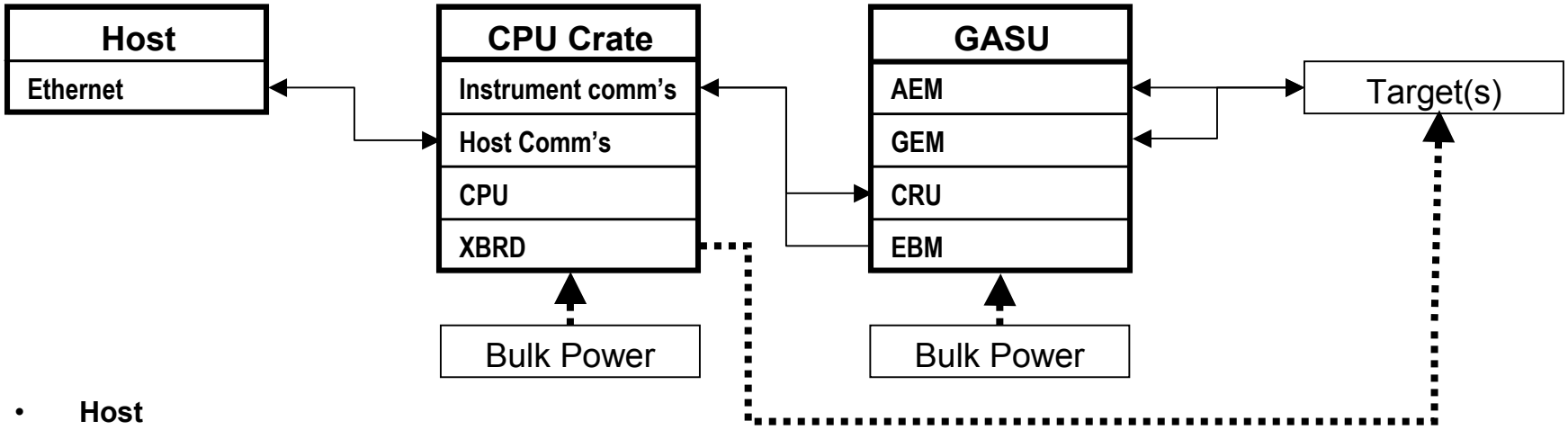
- **Instrument Flight Software (ourselves)**
  - **Development of instrument flight software final deliverable**
  - **Development done on the test-bed**
    - **A full flight-like DAQ system (except sensors)**
    - **Sensors emulated by Front End Simulators (FES)**

EM2 TKR or CAL Test Stand Hardware Configuration



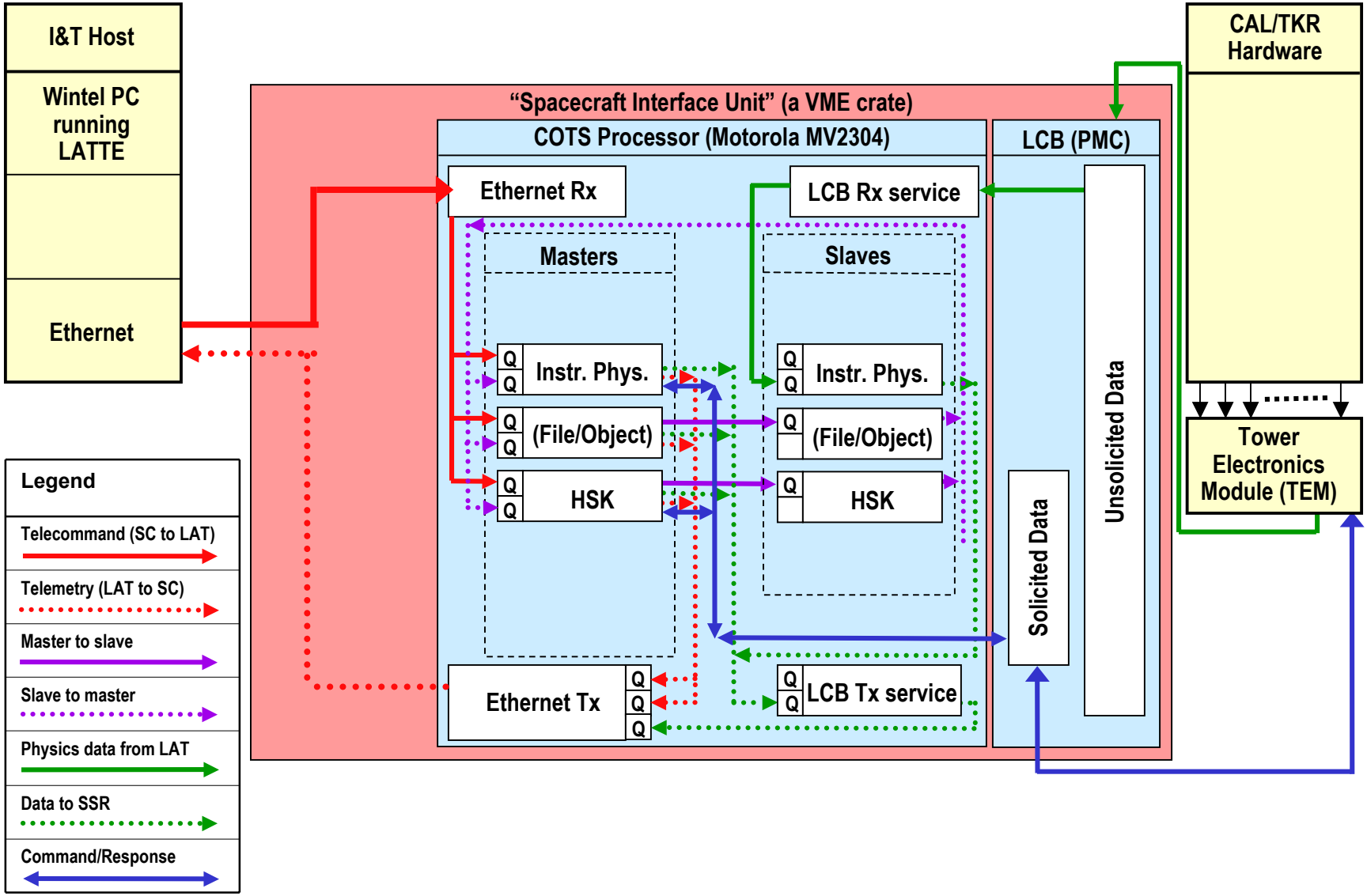
- Host
  - Wintel PC running I&T LATTE software
  - Ethernet communications (both command/response and event acquisition)
- CPU Crate
  - VME chassis
  - Instrument communications: LAT communications board (LCB) (PMC form factor)
  - Host communications: Ethernet (built in to COTS CPU)
  - CPU: Motorola 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)

EM2 ACD Test Stand Hardware Configuration

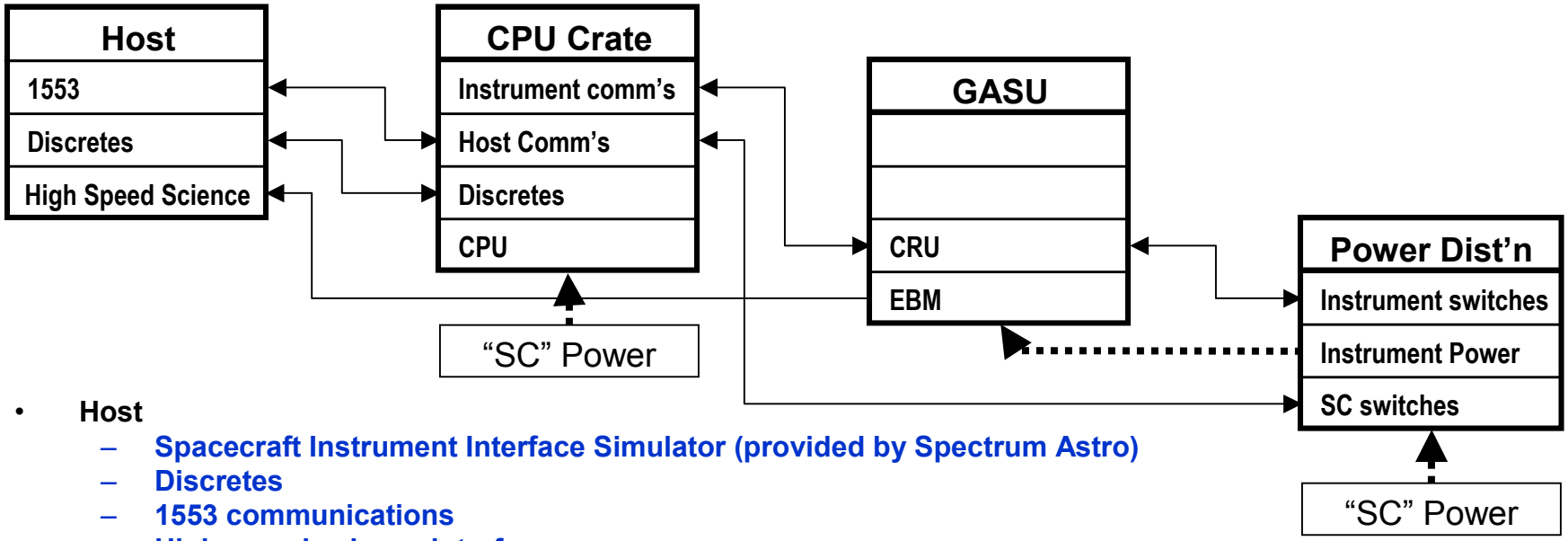


- Host
  - Wintel PC running I&T LATTE software
  - Ethernet communications (both command/response and event acquisition)
- CPU Crate
  - VME chassis
  - Instrument communications: LAT communications board (LCB) (PMC form factor)
  - Host communications: Ethernet (built in to COTS CPU)
  - CPU: Motorola 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

EM2 TKR, CAL, ACD Test Stand Software Configuration

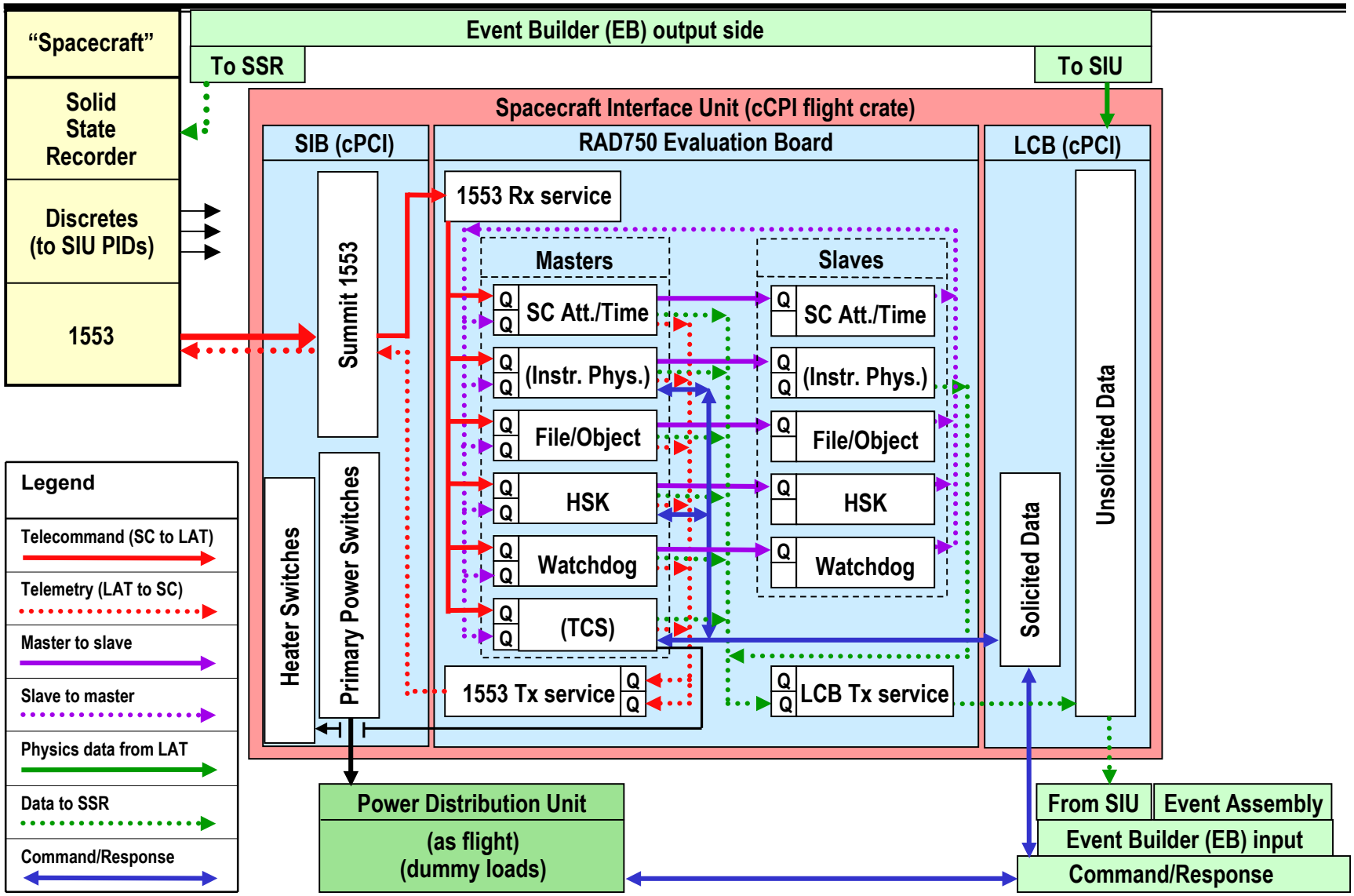


ISIS Hardware Configuration

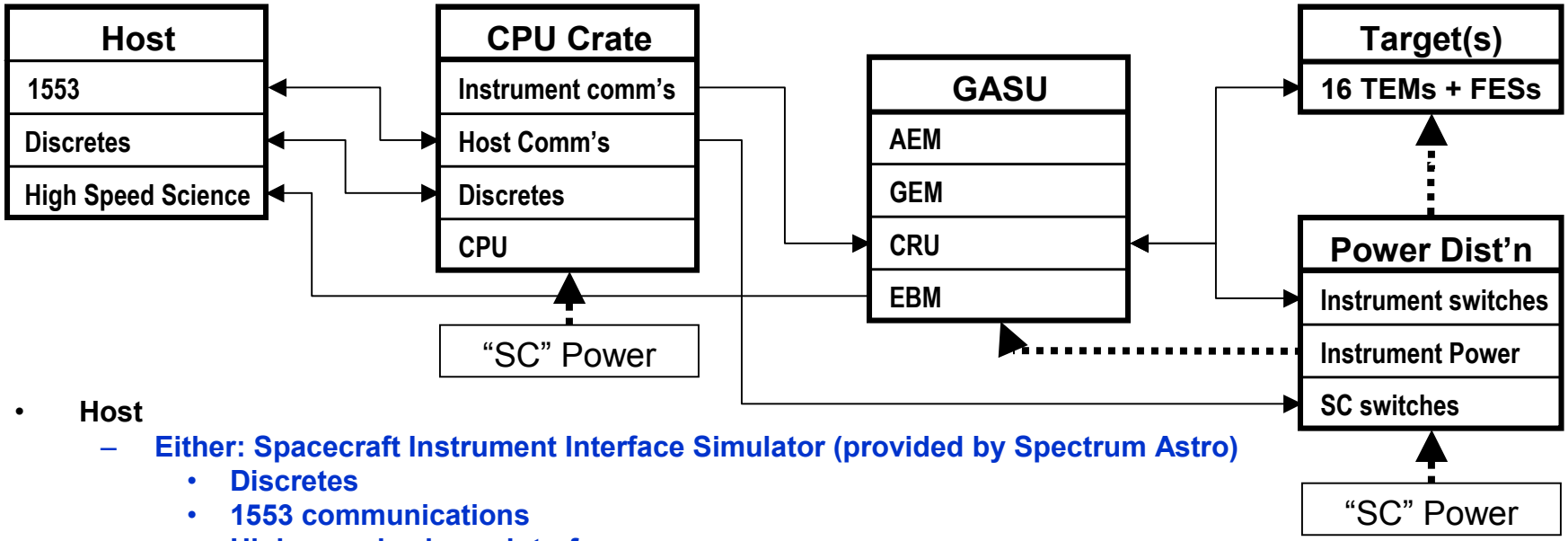


- Host
  - Spacecraft Instrument Interface Simulator (provided by Spectrum Astro)
  - Discretes
  - 1553 communications
  - High speed science interface
- CPU Crate
  - cPCI flight equivalent chassis
  - Instrument communications: LAT communications board (LCB) (cPCI form factor)
  - Host communications: 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

ISIS Software Configuration

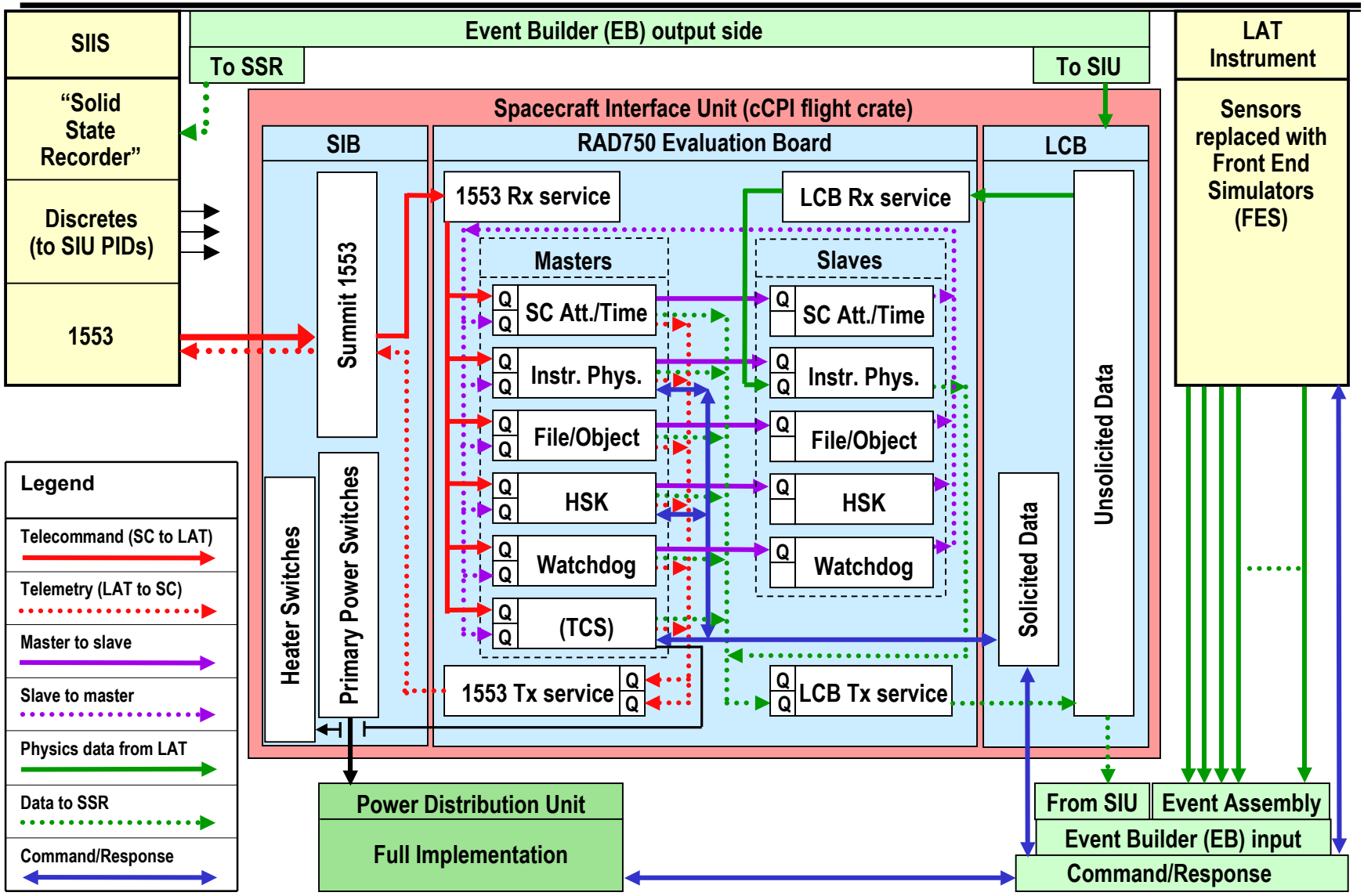


EM2 Test-Bed Hardware Configuration



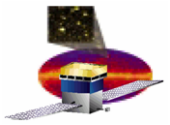
- Host
  - Either: Spacecraft Instrument Interface Simulator (provided by Spectrum Astro)
    - Discretes
    - 1553 communications
    - High speed science interface
  - Or: Ethernet simulation
- CPU Crate
  - cPCI flight equivalent chassis or commercial cPCI chassis
  - Instrument communications: LAT 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)

EM2 Test-Bed Software Configuration



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



Function/Consumer Mapping

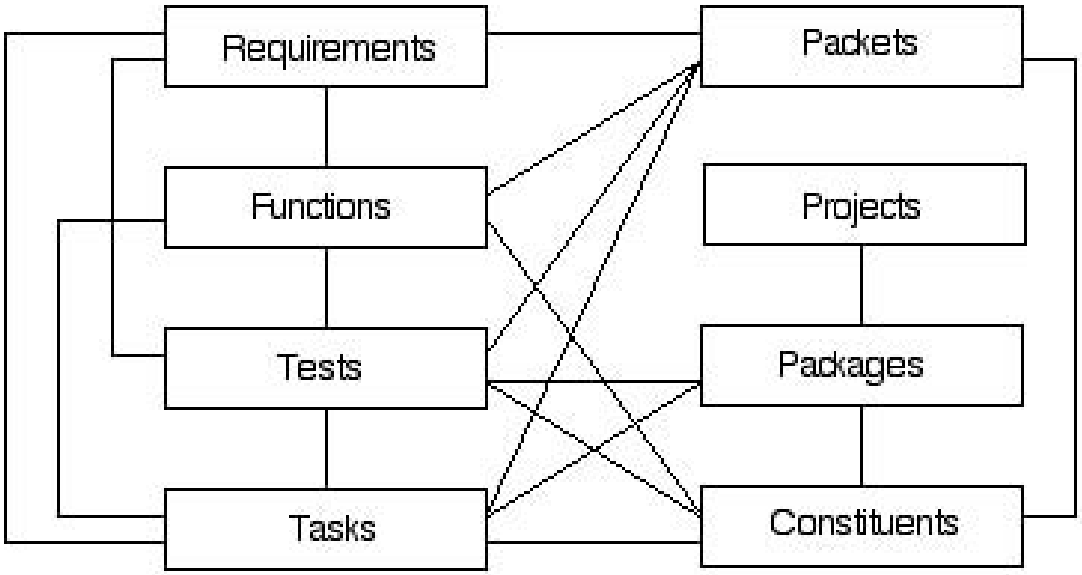
	Test Stands		ISIS	I&T	FSW Test-bed	FSW Deliverable
	TKR,CAL	ACD				
Configure TKR and CAL front end electronics	Y			Y	Y	Y
Configure ACD front end electronics		Y		Y	Y	Y
Configure GASU (CRU, GEM, EBM, AEM)		Y	Y	Y	Y	Y
Configure PDU			Y	Y	Y	Y
Configure XBRD	Y	Y		Y		
Configure by compressed file			Y	Y	Y	Y
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)**

Functions & Packages

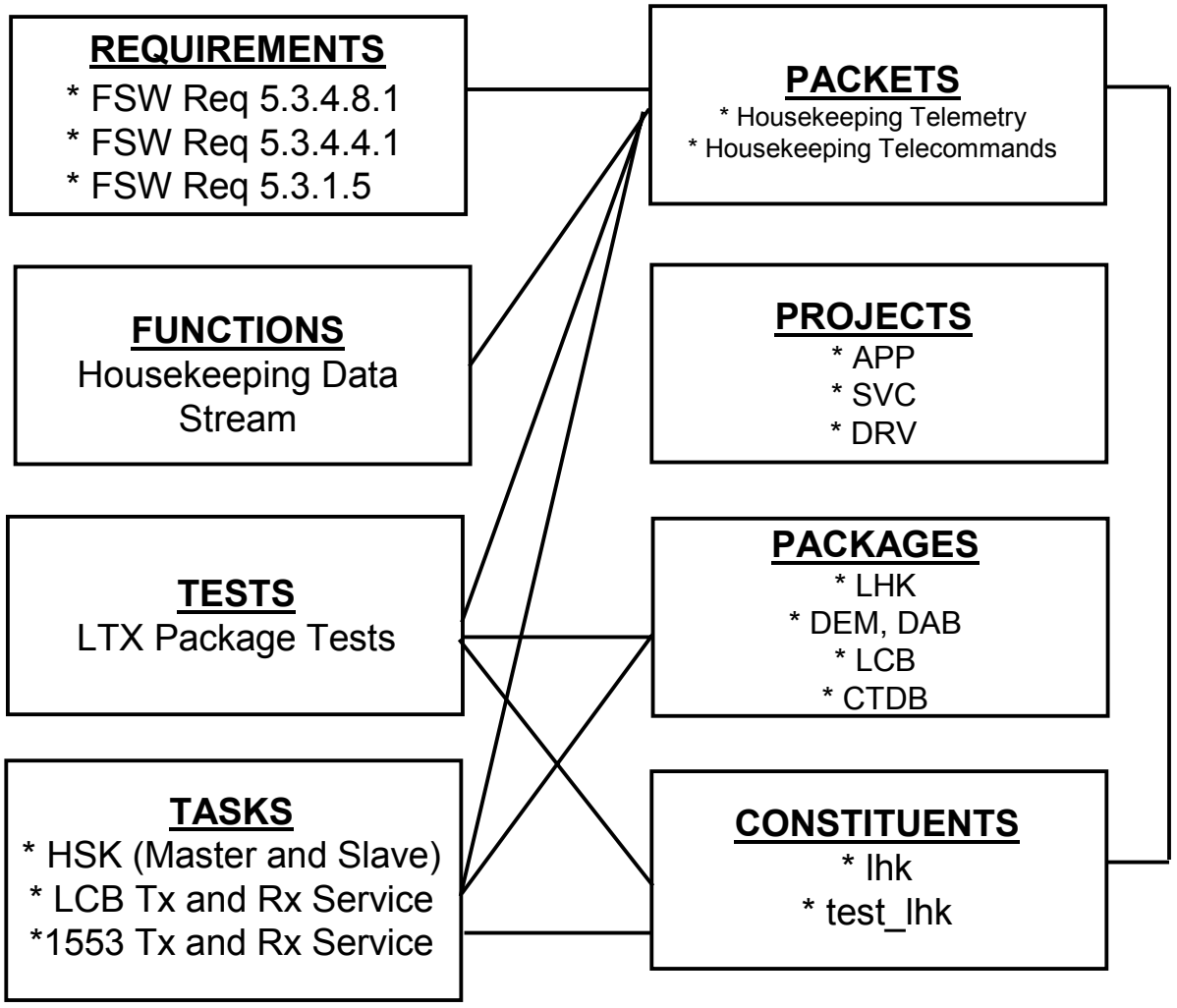
Functions & Packages

- **The diagram below illustrates the major entities which FSW must track.**
  - **The top-level requirements are dictated by the goals of the scientific mission and the engineering and physical realities of spaceflight. Lower-level requirements are developed to add necessary detail. Most requirements relate to particular functions (e.g., Thermal Control) that must be performed.**
  - **The FSW group is developing tests to verify that its software meets the relevant requirements. Along with verifying the underlying hardware, these tests verify that the operational functions of the flight software are being performed correctly by the FSW tasks.**
  - **Tasks are assembled, at run time, from libraries of dynamically-loadable constituents. For code-maintenance purposes, constituents are collected into packages and projects.**
  - **Most FSW communication (e.g., between tasks, with the spacecraft) is performed by means of packets. The code to generate, send, accept, and process these is defined in the relevant constituents.**



Functions & Packages (cont'd)

Let's take the Housekeeping function as an example. The entities associated with Housekeeping are shown at right.

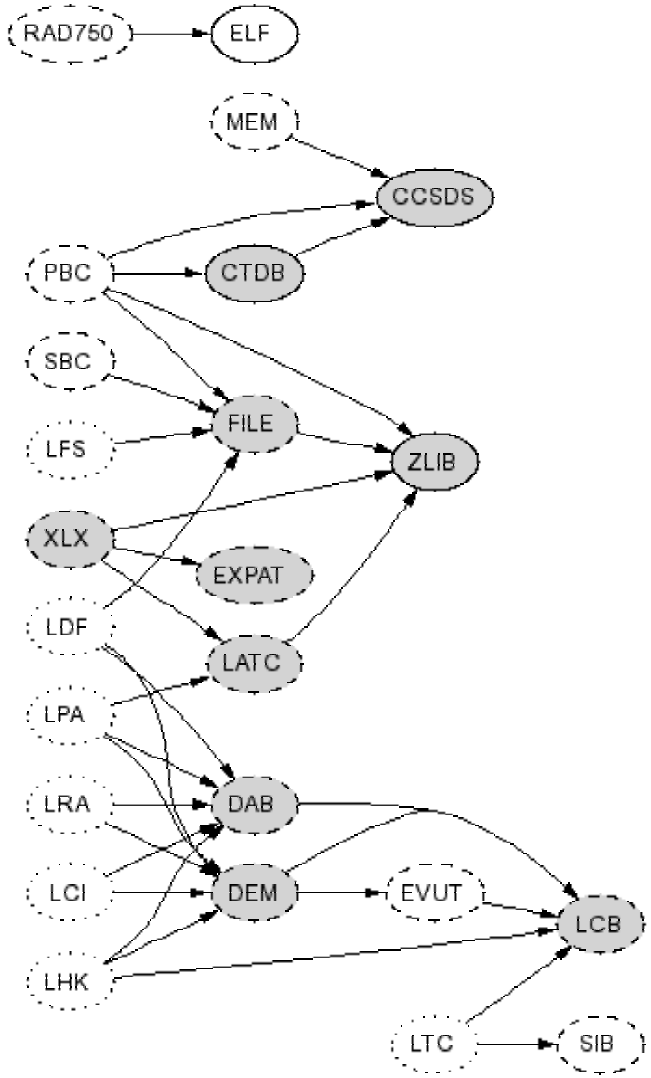


Function/Package Mapping

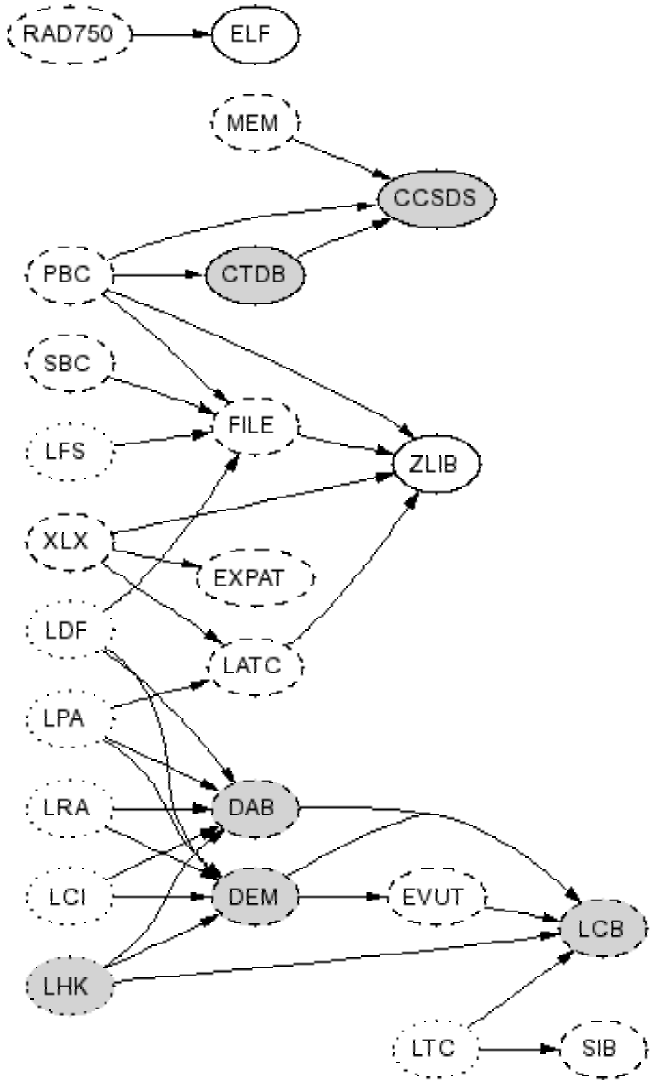
		CCSDS	CLI	CMX	CTDB	DAB	DEM	EXPAT	FILE	GGLT	GNAT	GRL	LATC	LCB	MSG	OCS	OES	PBC	PBI	PBS	VXW	XLX	ZLIB
Function	Configure TKR and CAL front end electronics			Y			Y							Y	Y				Y	Y	Y		
	Configure ACD front end electronics			Y			Y							Y	Y				Y	Y	Y		
	Configure GASU (CRU, GEM, EBM, AEM)			Y		Y								Y	Y				Y	Y	Y		
	Configure PDU			Y		Y								Y	Y				Y	Y	Y		
	Configure XBRD			Y						Y	Y			Y	Y				Y	Y	Y		
	Configure by compressed file			Y		Y	Y	Y					Y		Y				Y	Y	Y	Y	Y
	Real event delivery (instrument to CPU)			Y		Y	Y			Y				Y	Y	Y	Y		Y	Y	Y		
	Housekeeping data stream		Y	Y		Y	Y			Y		Y		Y	Y				Y	Y	Y		
	RAD750 boot and crate initialization	Y	Y	Y	Y				Y			Y			Y			Y	Y	Y	Y		Y
	1553 bus communications	Y	Y	Y								Y			Y				Y	Y	Y		
	Telecommand/telemetry database and services			Y																			
	Emulated event delivery (to science data interface)			Y																			
	CPU internal communications/task frameworks			Y																			
	Software watchdog			Y																			
	Wall clock time services (GPS)			Y																			

- [Access to package API documentation (main index)]

Function/Package Diagram: Configuration Function



Function/Package Diagram: Housekeeping Function



Function/Package Diagram: Event Delivery Function