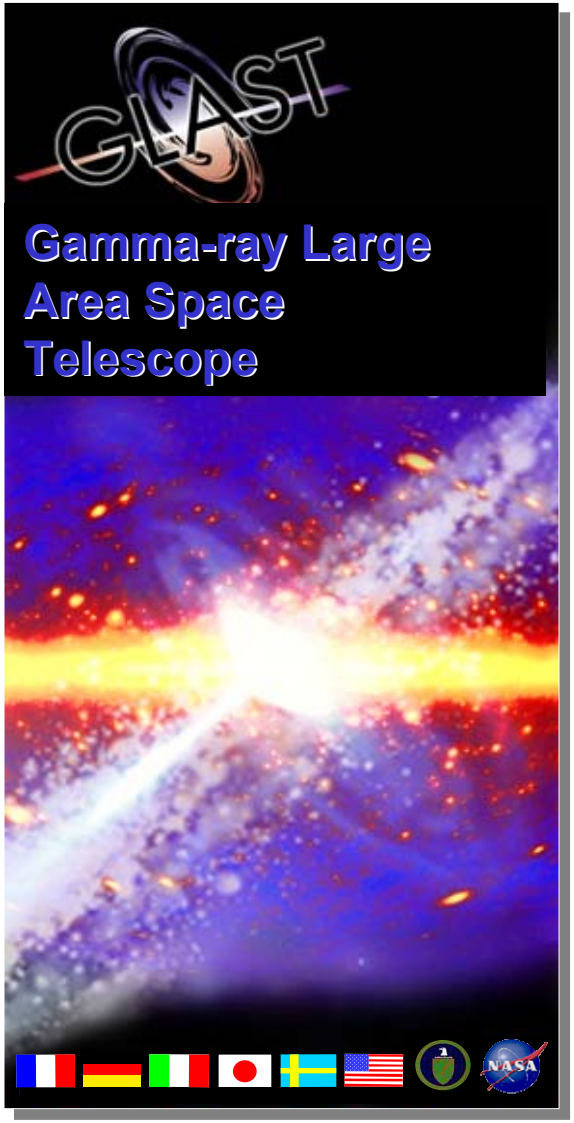



# GLAST Large Area Telescope

## Instrument Flight Software
## NASA EM2 Review
## 26 February 2004

## Testing

**Eric Hansen**
**Stanford Linear Accelerator Center**

**hansene@slac.stanford.edu**
**(650) 926-4057**

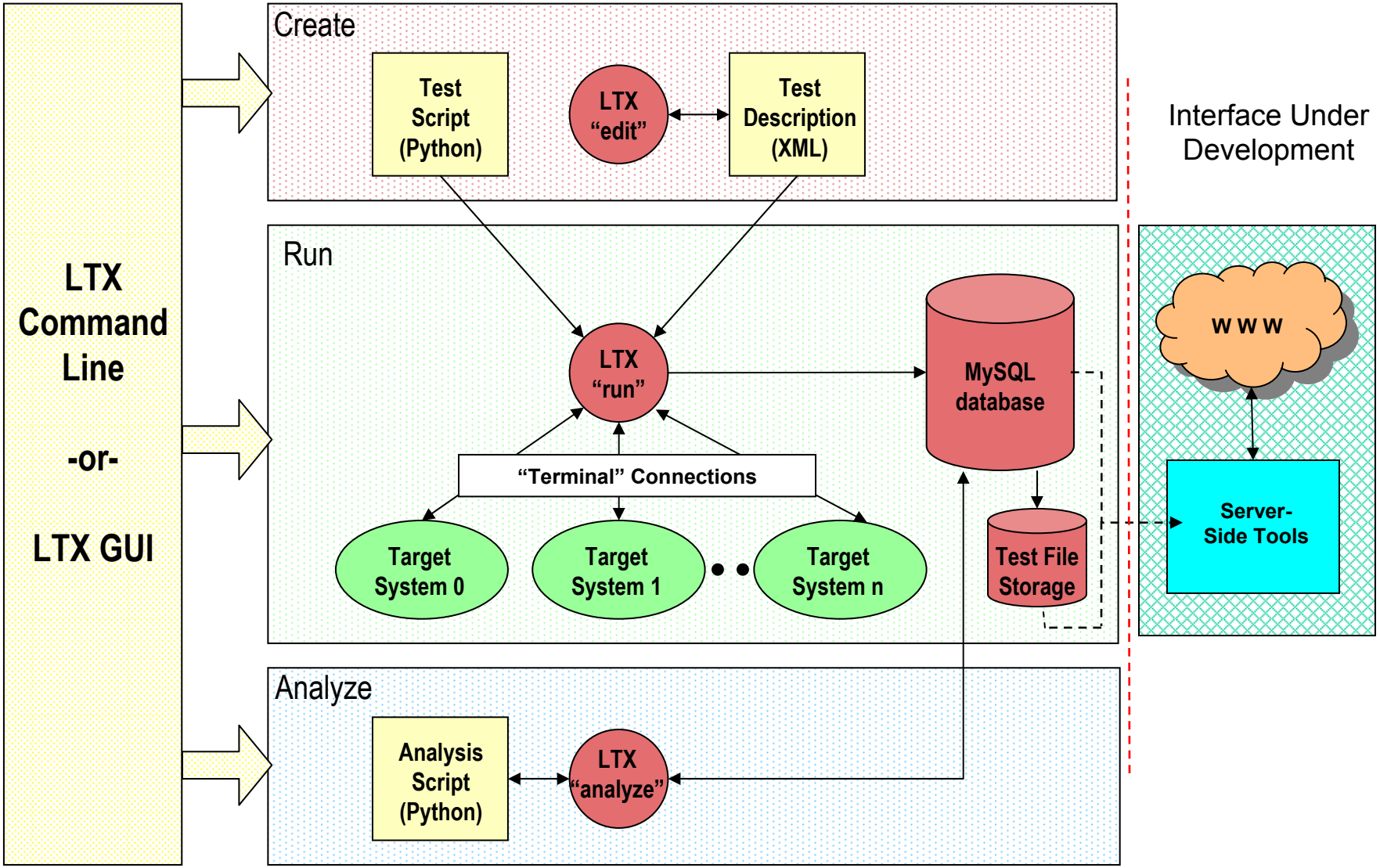
# Test Plan Status

- **Test Plan Document: LAT-TD-00786**
  - **Most still accurate**
    - **Identifies functionality expected at EM1, EM2, FU**
    - **Traces tests to requirements**
  - **Needs some revising**
    - **Reflect improved knowledge of final system**
    - **Add specificity to formal tests to be executed for FU**
- **ISIS Test Plan Document**
  - **Follow-on to ongoing requirement definition phase**
  - **Test Procedure document to follow**

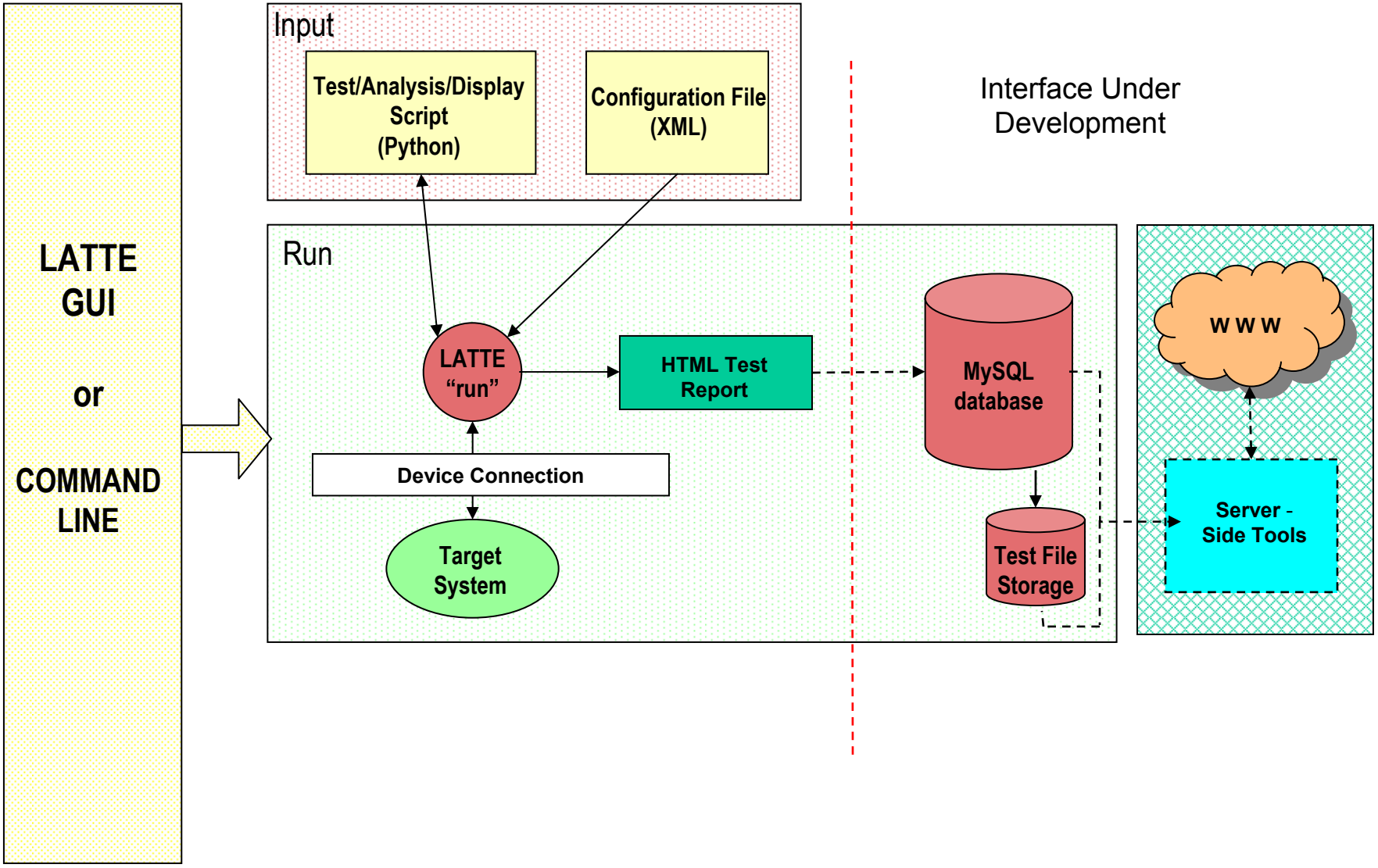
# Test Executives: LTX, LATTE, AstroRT

- LTX (low level interface)
  - Flight software product
  - Sees LAT processors through a VxWorks shell task
    - Can load and run test-only code in VxWorks context
    - Can do anything allowed at the shell prompt (memory dumps, …)
  - Well suited to flight software internals testing
  - Primarily "white-box" testing
- LATTE (high level interface)
  - I&T product
  - Sees LAT through spacecraft-like interface (CTDB)
    - Can only access what flight software provides
      - FSW currently providing register level access to hardware
  - Well suited to interface level testing
  - Primarily "black-box" testing
- AstroRT (very high-level interface)
  - Runs on SIIS
  - Used for system-level testing
- Test group chooses
  - LTX: Low level utilities, performance testing, …
  - LATTE: Higher-level internal testing (eases transfer to I&T)
  - AstroRT: System-level testing, V&V

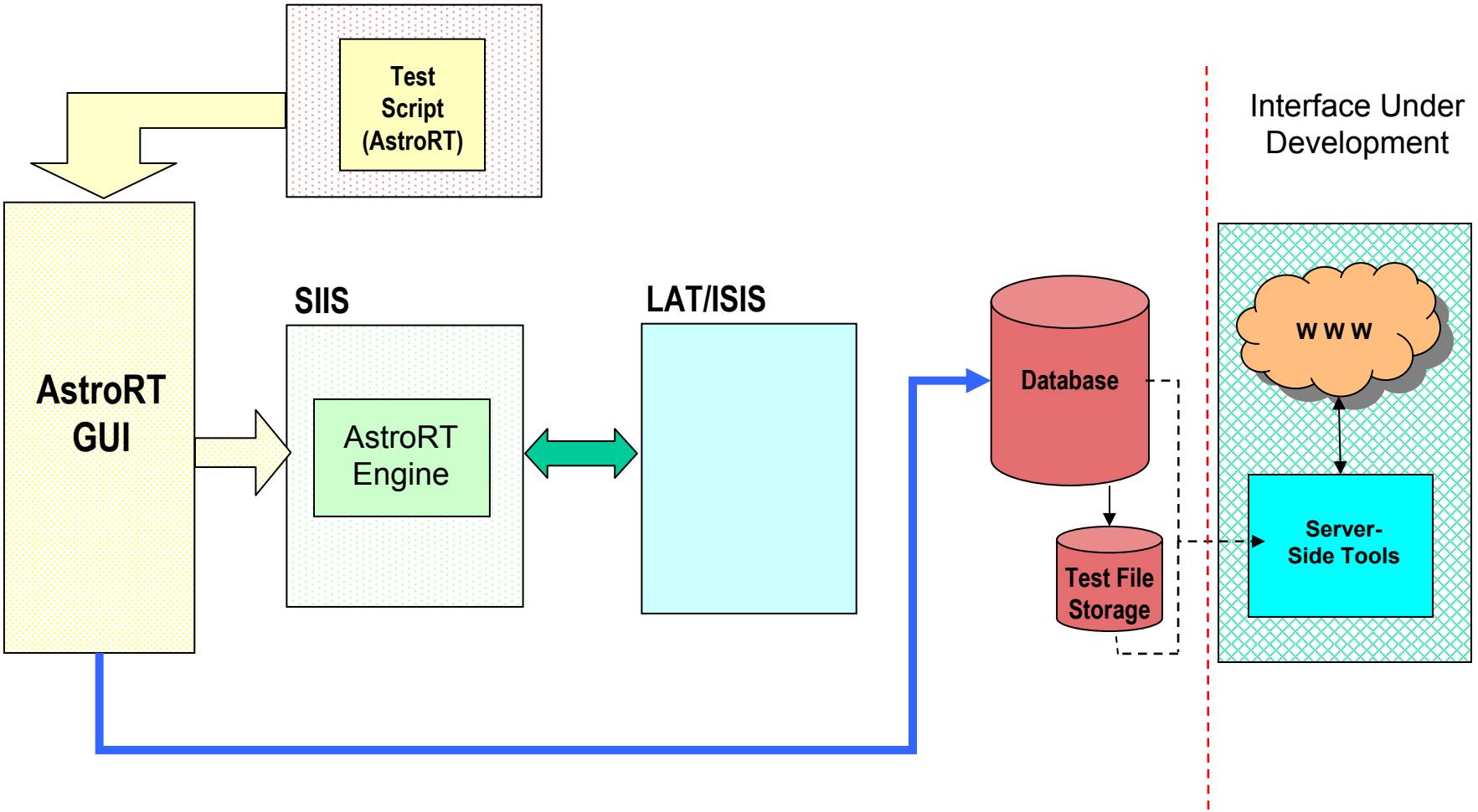
# LTX


LTX Command Line
-or-
LTX GUI
Create
Test Script (Python)
LTX "edit"
Test Description (XML)
Run
LTX "run"
MySQL database
"Terminal" Connections
Target System 0
Target System 1
Target System n
Test File Storage
Analyze
Analysis Script (Python)
LTX "analyze"
Interface Under Development
W W W
Server-Side Tools

# LATTE


LATTE GUI or COMMAND LINE
Input
Test/Analysis/Display Script (Python)
Configuration File (XML)
Interface Under Development
Run
LATTE "run"
HTML Test Report
MySQL database
W W W
Device Connection
Target System
Test File Storage
Server - Side Tools

# AstroRT


Test
Script
(AstroRT)
AstroRT
GUI
SIIS
AstroRT
Engine
LAT/ISIS
Database
Test File
Storage
Interface Under
Development
W W W
Server-
Side Tools

# Overall Test Coverage

- **Best is to go through process & plan for a specific release, e.g, the ISIS**

- **Generic Process:**
  - **1) Requirements**
  - **2) Identify hardware required to meet requirements**
  - **3) Identify software required to meet requirements (functionality)**
  - **4) Identify/create scripts required to test system**
  - **5) Acquire hardware and collect software (identify packages)**
  - **6) Unit test (LTX) packages, archive scripts/results in MySQL**
  - **7) Function test release per requirements on target (LTX/LATTE)**
  - **8) Run LATTE scripts, with MySQL interface for test reporting similar to LTX interface (keep current-format HTML filename in table)**
  - **9) Run AstroRT scripts with SIIS, results into database**
  - **10) Publish test results using FSW group's web pages**

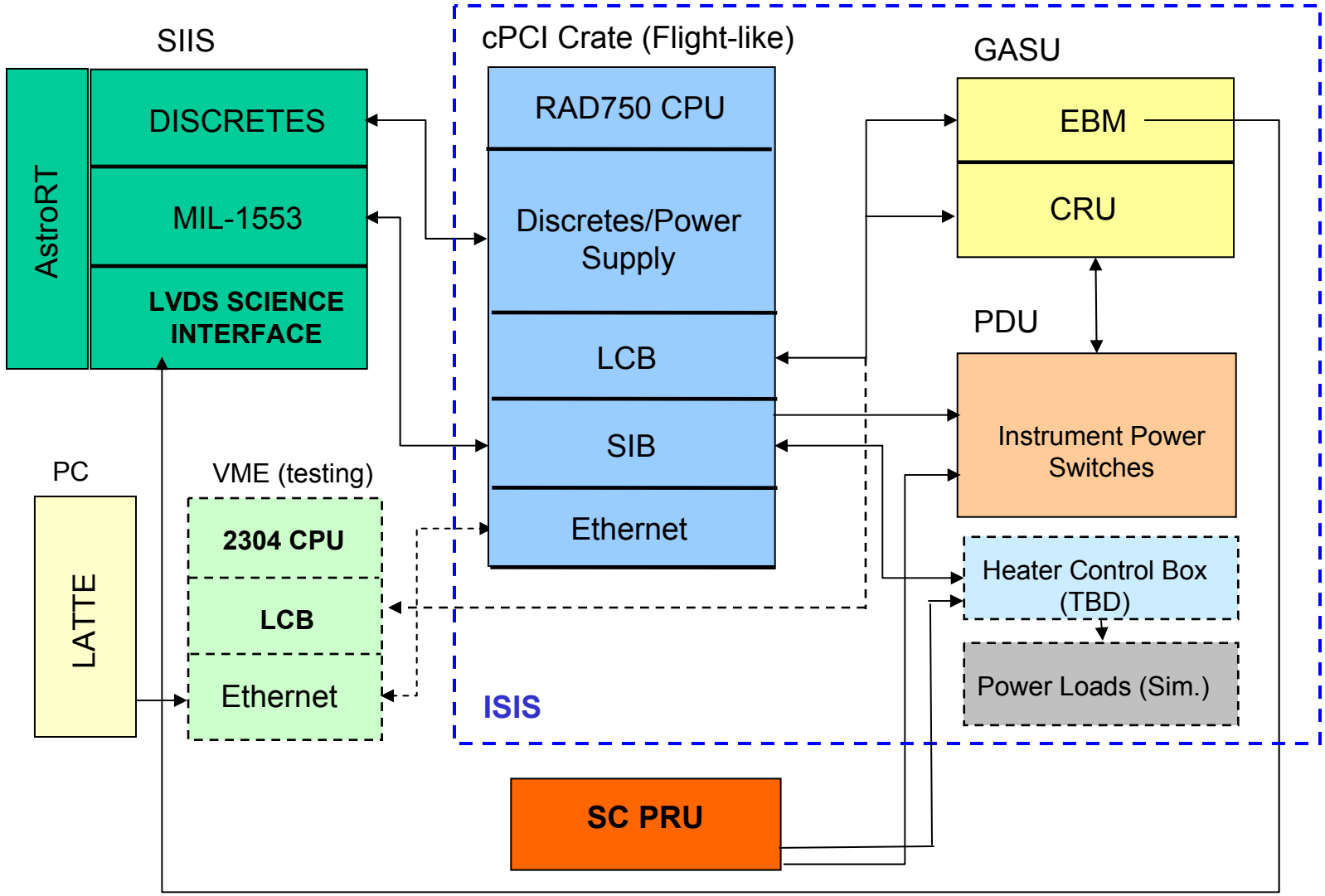
# ISIS Requirements

- Requirements
  - High-level and lower-level derived requirements defined
    - Currently under review
  - ISIS requirements finalized by 3/19/04
- Requirements Review Personnel:
    - E. Hansen (SLAC) LAT ISIS lead working with
    - Mission project E. Andrews (GSFC)
    - Spectrum Astro
      - Tim Morse
      - Ron Zitek
    - M. DeKlotz (GSFC Systems Engineering)
- Requirements Status:
  - Software requirements review proceeding
  - Hardware requirements in work

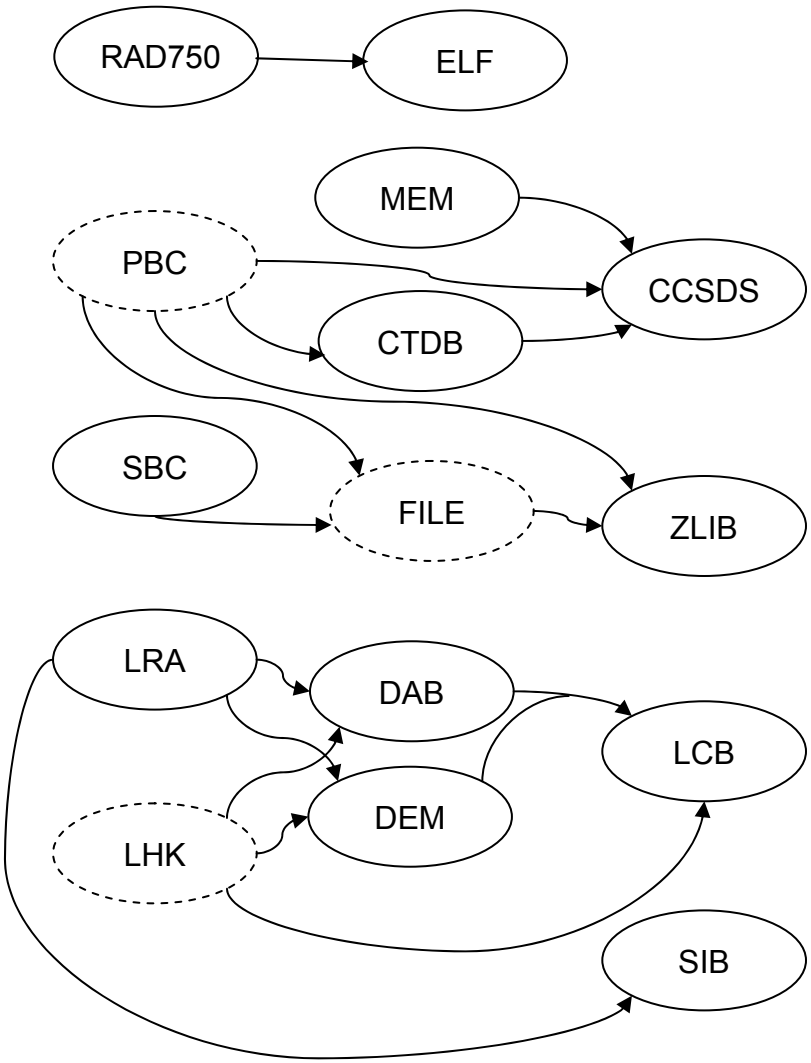
# ISIS Requirement Map

| | Test Stands | | ISIS | I&T | FSW Test-bed | FSW SRS Deliverable Requirement |
|---|---|---|---|---|---|---|
| | TKR,CAL | ACD | | | | |
| Configure TKR and CAL front end electronics | Y | | | Y | Y | 5.3.4.6 |
| Configure ACD front end electronics | | Y | | | Y | 5.3.4.6 |
| Configure GASU (CRU, GEM, EBM, AEM) | | Y | Y | Y | Y | 5.3.4.6 |
| Configure PDU | | | Y | Y | Y | 5.3.4.6 |
| Configure XBRD | Y | Y | | Y | | N/A |
| Configure by compressed file | | | Y | Y | Y | 5.3.4.6 |
| Real event delivery (instrument to CPU) | Y | Y | | Y | Y | 5.2.2.3 |
| Housekeeping data stream | Y | Y | Y | Y | Y | 5.3.4.8 |
| RAD750 boot and crate initialization | | | Y | Y | Y | 5.2.2.1, 5.3.4.1 |
| 1553 bus communications | | | Y | Y | Y | 5.3.1.1 |
| Telecommand/telemetry database and services | | | Y | Y | Y | 5.3.4.2, 5.3.4.4 |
| Emulated event delivery (to science data interface) | | | Y | | | 5.3.1.4 |
| CPU internal communications/task frameworks | | | Y | Y | Y | Derived |
| Software watchdog | | | Y | Y | Y | 5.2.1.2, 5.3.2.1 |
| Wall clock time services (GPS) | | | Y | Y | Y | 5.3.4.3.4-5 |

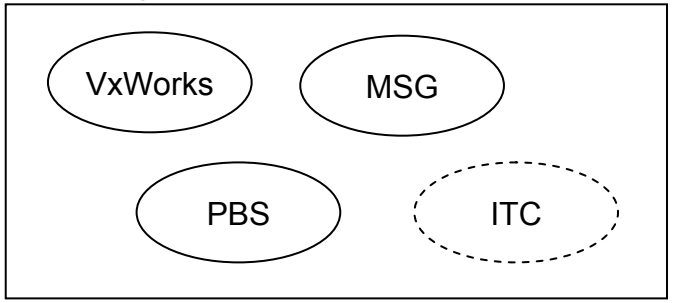
# ISIS Test-Stand/Hardware



# ISIS Software Packages




Packages Common to all

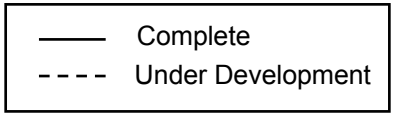



Still needed (not developed):

- Test pattern generation code (small effort)
- Cmd + Tlm Database (ISIS specific)

Note: For packages that are complete, functional and unit-level test scripts already exist

Key



# ISIS Schedule

- External Schedule Assumptions
  - SIIS delivered end of April
    - Error free operation
  - Full delivery of connectors/panels
  - 3 month lead on BAE RAD750
    - Order date =~ 3/15/04
    - RFP scheduled for return by 3/5/04
      - BAE “re-evaluating” pricing strategy
    - Will use available CPUs until then

# ISIS Schedule (2)

| Task | Length | Start | Complete |
|---|---|---|---|
| **Instrument-Spacecraft I/F Simulator** | 135 days? | 1/26/2004 | 7/30/2004 |
| ISIS Plan/Def'n/Schedule | 15 days? | 2/23/2004 | 3/12/2004 |
| ISIS Integration Area Available | 10 days? | 3/29/2004 | 4/9/2004 |
| ISIS Test Ready Review | 1 day? | 6/21/2004 | 6/21/2004 |
| ISIS Formal Testing | 10 days? | 6/22/2004 | 7/5/2004 |
| ISIS Pack/Ship Week | 5 days? | 7/19/2004 | 7/23/2004 |
| ISIS Install/Train/RegressionTest | 5 days? | 7/26/2004 | 7/30/2004 |
| **ISIS Documentation** | 103 days? | 2/2/2004 | 6/23/2004 |
| FSW Requirements Def'n/Mapping | 35 days? | 2/2/2004 | 3/19/2004 |
| ISIS Test Plan / Procedures | 50 days? | 3/22/2004 | 5/28/2004 |
| ISIS Test Reports | 25 days? | 5/10/2004 | 6/11/2004 |
| ISIS C&T DB Dev't / Delivery | 80 days? | 3/4/2004 | 6/23/2004 |
| ISIS FSW User Man/ Vers Desc Doc | 60 days? | 3/29/2004 | 6/18/2004 |
| **ISIS Hardware Available** | 102 days? | 1/26/2004 | 6/15/2004 |
| **SIU** | 102 days? | 1/26/2004 | 6/15/2004 |
| SIB | 1 day? | 1/26/2004 | 1/26/2004 |
| CPU | 67 days? | 3/15/2004 | 6/15/2004 |
| **LCB** | 50 days? | 1/26/2004 | 4/2/2004 |
| PMC Version | 5 days? | 1/26/2004 | 1/30/2004 |
| cPCI Version | 2 days? | 4/1/2004 | 4/2/2004 |
| Enet I/F | 10 days? | 3/22/2004 | 4/2/2004 |
| **GASU** | 12 days? | 5/17/2004 | 6/1/2004 |
| AEM module | 12 days? | 5/17/2004 | 6/1/2004 |
| GEM module | 12 days? | 5/17/2004 | 6/1/2004 |
| EBM module | 12 days? | 5/17/2004 | 6/1/2004 |
| CRU module | 12 days? | 5/17/2004 | 6/1/2004 |
| PDU | 1 day? | 4/1/2004 | 4/1/2004 |
| Temp / Volt Sensor PCB | 1 day? | 4/15/2004 | 4/15/2004 |
| Dummy Load PCB | 1 day? | 4/15/2004 | 4/15/2004 |

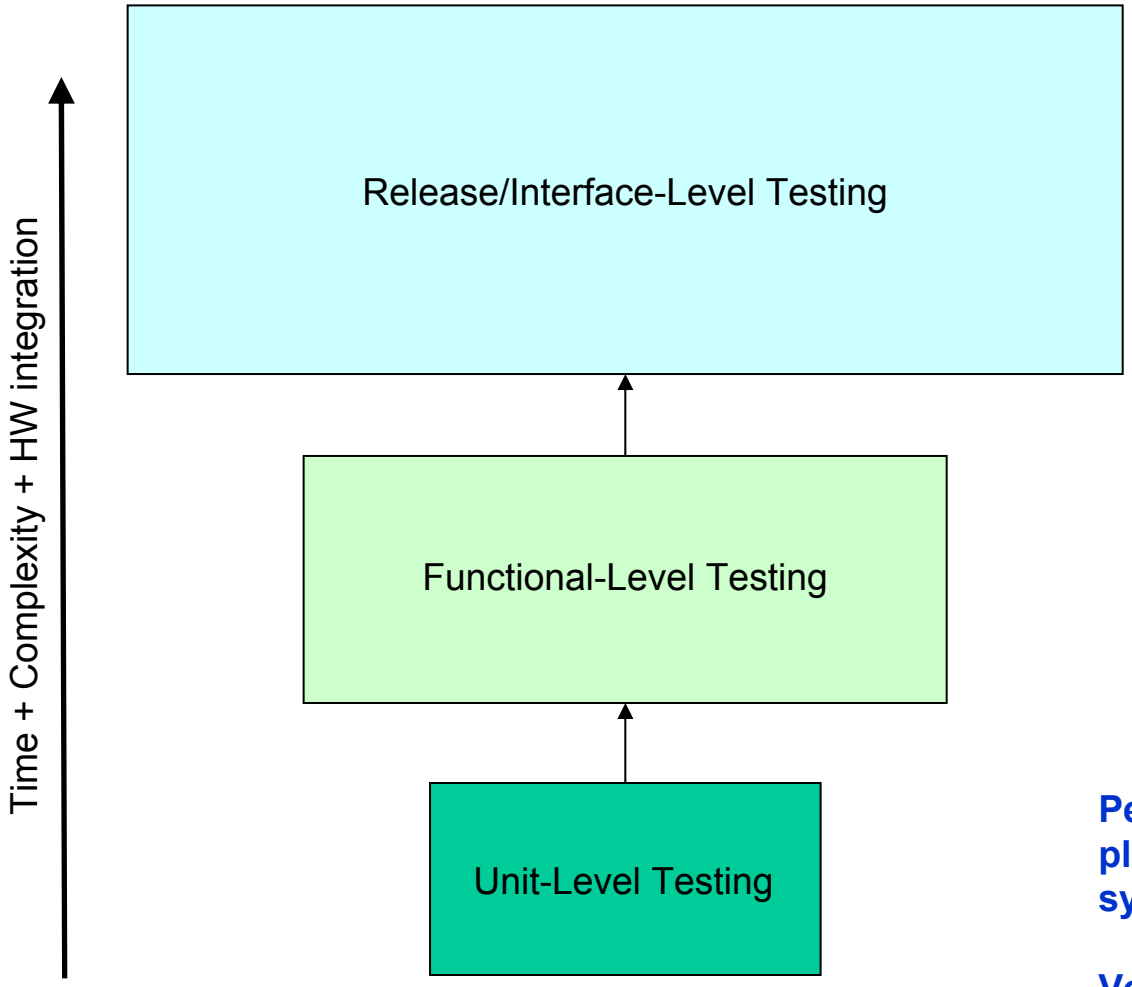
# ISIS Schedule (3)

| Task | Length | Start | Complete |
|---|---|---|---|
| **Rack/Enclosure** | 12 days? | 4/1/2004 | 4/16/2004 |
| LAT I/F Panel | 1 day? | 4/1/2004 | 4/1/2004 |
| Rack/Enclosure | 1 day? | 4/16/2004 | 4/16/2004 |
| **Cabling** | 1 day? | 4/15/2004 | 4/15/2004 |
| SIU - GASU | 1 day? | 4/15/2004 | 4/15/2004 |
| SIU - PDU | 1 day? | 4/15/2004 | 4/15/2004 |
| GASU - I/F Panel | 1 day? | 4/15/2004 | 4/15/2004 |
| SIU - I/F Panel | 1 day? | 4/15/2004 | 4/15/2004 |
| PDU - I/F Panel | 1 day? | 4/15/2004 | 4/15/2004 |
| Temp/Volt PCB - I/F Panel | 1 day? | 4/15/2004 | 4/15/2004 |
| Dummy Load PCB - I/F Panel | 1 day? | 4/15/2004 | 4/15/2004 |
| ISIS Assembly & Integration | 34 days? | 5/10/2004 | 6/24/2004 |
| **ISIS Ground Support Equip** | 50 days? | 3/1/2004 | 5/7/2004 |
| PC / LATTE WS | 13 days? | 3/1/2004 | 3/17/2004 |
| PC / LTX WS | 13 days? | 3/1/2004 | 3/17/2004 |
| SIIS | 10 days? | 4/26/2004 | 5/7/2004 |
| **ISIS FSW** | 125 days? | 2/2/2004 | 7/23/2004 |
| FSW Requirements Identified | 35 days? | 2/2/2004 | 3/19/2004 |
| **FSW Integration Testing (w/ Test Procs)** | 44 days? | 4/23/2004 | 6/23/2004 |
| Test 1a: 1553 Commands | 15 days? | 6/1/2004 | 6/21/2004 |
| Test 1b: Attitude, Ancillary, TimeTone(?) | 5 days? | 6/15/2004 | 6/21/2004 |
| Test 2: HKP Telemetry | 25 days? | 5/20/2004 | 6/23/2004 |
| Test 3: Alert Telemetry | 25 days? | 4/26/2004 | 5/28/2004 |
| Test 4: Diag Telemetry | 15 days? | 5/17/2004 | 6/4/2004 |
| Test 5a: Test Cmds for SciDataIF | 26 days? | 5/17/2004 | 6/21/2004 |
| Test 5b: Test Cmds for Discretes | 26 days? | 5/17/2004 | 6/21/2004 |
| Test 5c: Test Cmds for ARR TC | 10 days? | 5/17/2004 | 5/28/2004 |
| Test 6: Boot Cmds/Tlm | 42 days? | 4/23/2004 | 6/21/2004 |
| FSW Formal Testing (w/ Procs) | 15 days? | 6/28/2004 | 7/16/2004 |
| **Contingency** | 5 days? | 7/19/2004 | 7/23/2004 |

# Testing (ISIS + FSW)

- What will be tested and by who
  - Package/Unit level: LTX test scripts
    - Developer of packages
  - Release/function/performance test: LTX test scripts
    - FSW test team
  - Release/interface test + necessary regression: LATTE/LTX/AstroRT
    - Test group
      - Lead: E. Hansen (acting)
      - Scripts by
        - » J & G Thayer
        - » A. Perazzo
        - » E. DeLange
        - » RA's (3-4 as needed)
    - Test script generation
      - Test script(s) to test all MIL1553 commands
      - Test script(s) to test science interface
      - Etc.

# Software Testing Approach




**Highest level tests – verify & validate against requirements**

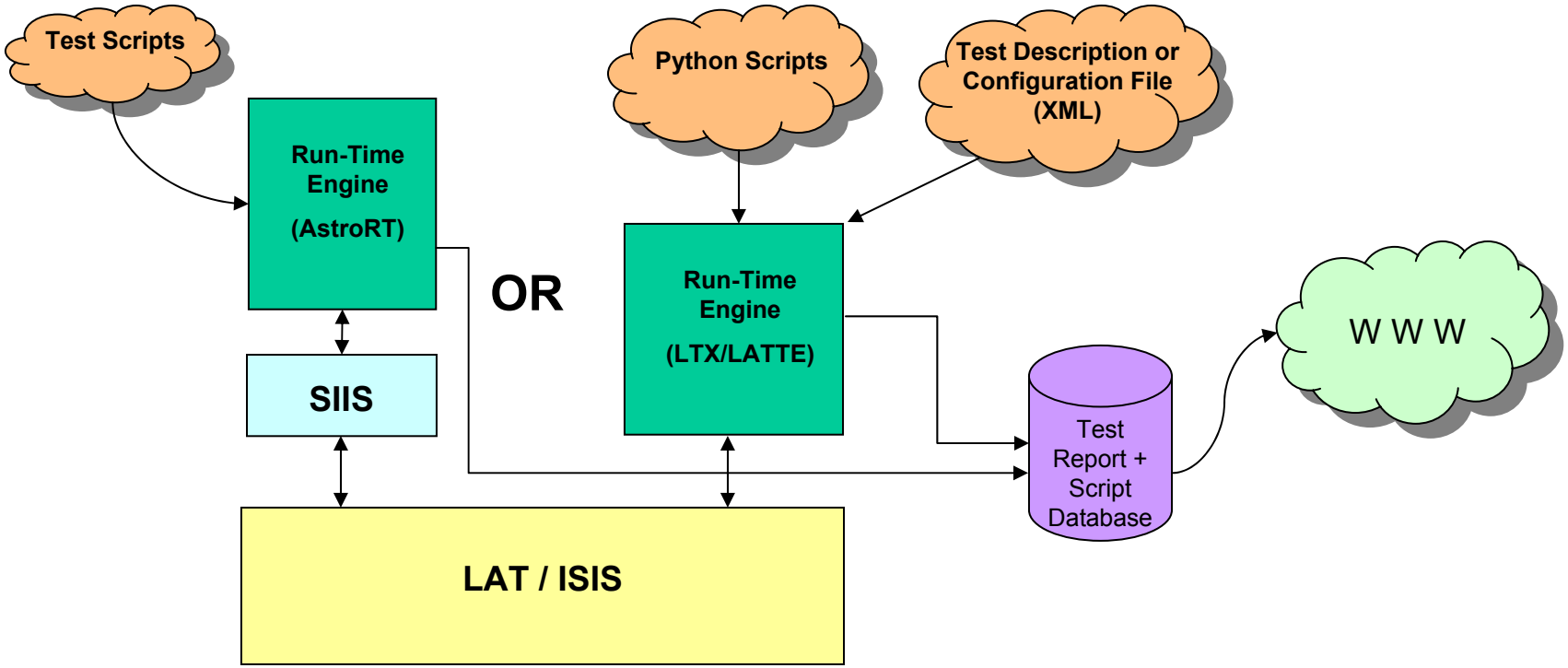
**Higher-level functionality tests – combines many packages**

**Verify functionality**

**Performed on developer's platform or captive embedded system**

**Verify algorithm development, debug software logic**

# Test Environment




- Test Scripts/Description generated by developer/test group
- Depending on test phase, either LTX/LATTE or AstroRT chosen
- Run-Time Engine executes selected scripts, performs analysis
- Test results + all scripts output to database for repeatability
- FSW Web able to access test reports for any test performed throughout the development/ V&V process
  - Allows program management to assess development progress

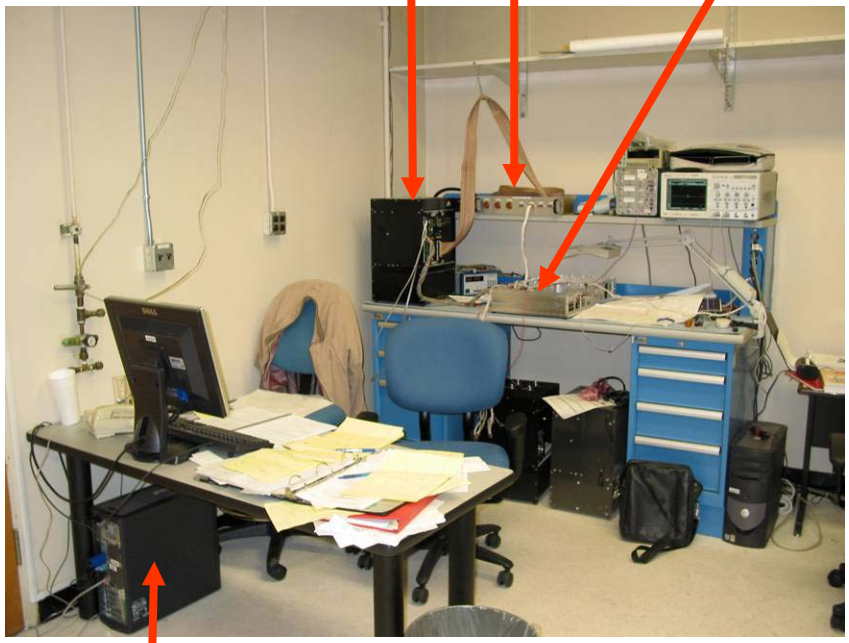
# GASU Development Test Stand (GASU is main EM1 to EM2 increment)

Trigger Signal Breakout Box (for Global Trigger Module Testing)

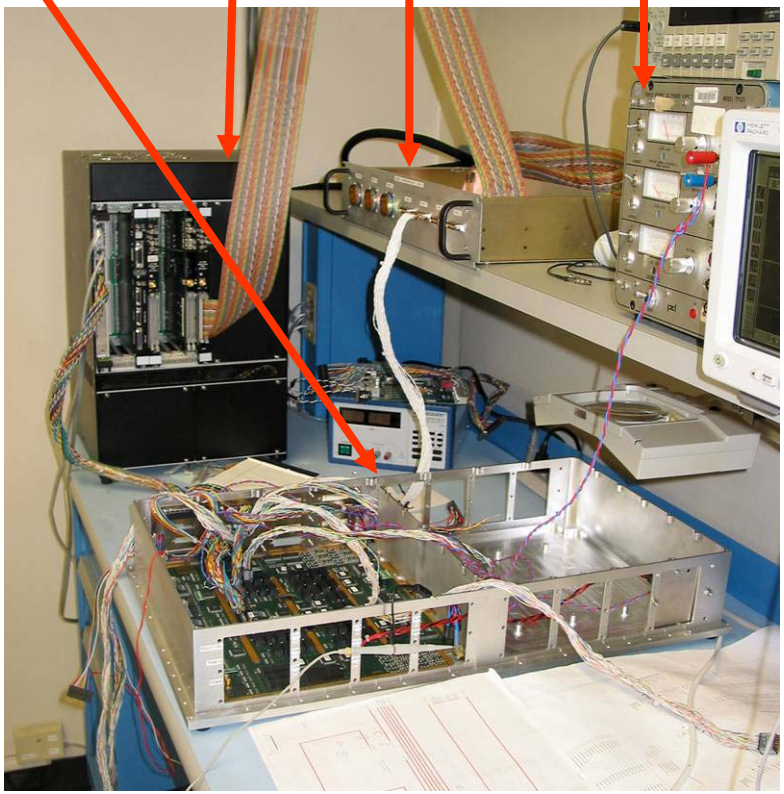
VME Test Stand

28V Supply

GASU



PC running LATTE



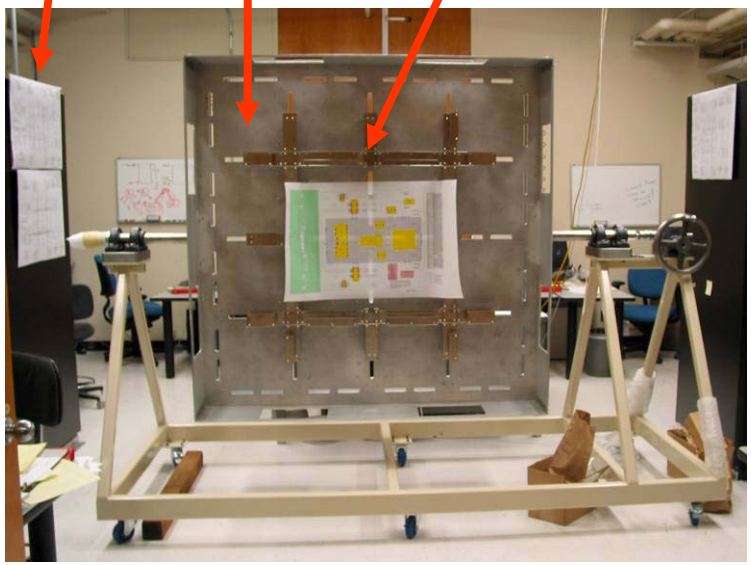
# Test-Stand (2)

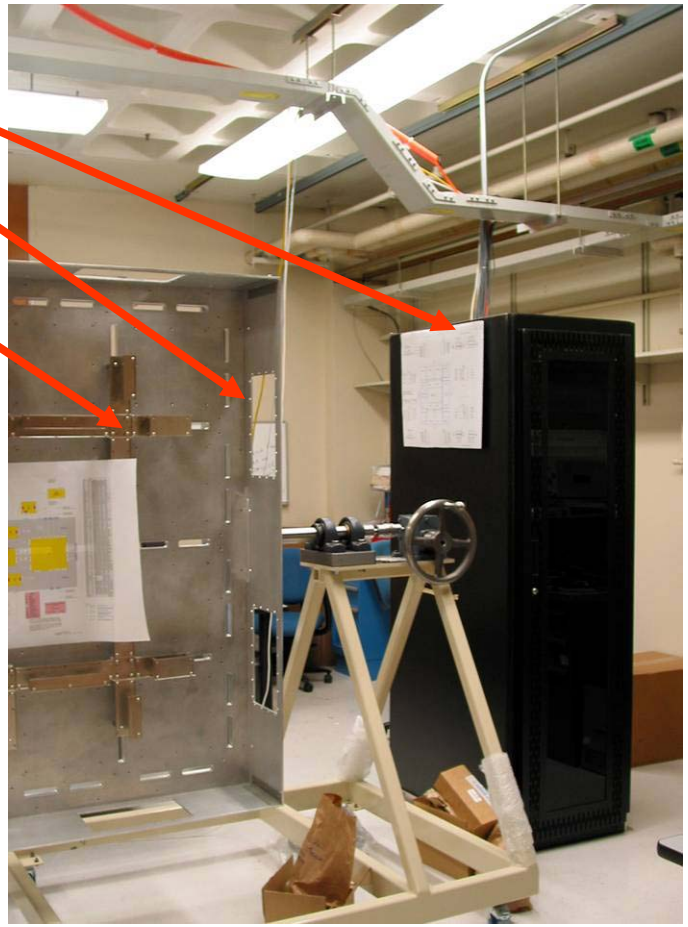
Racks for FES “feeder” CPUs

4x4 grid for electronics mounting

Cable feedthroughs (as flight)

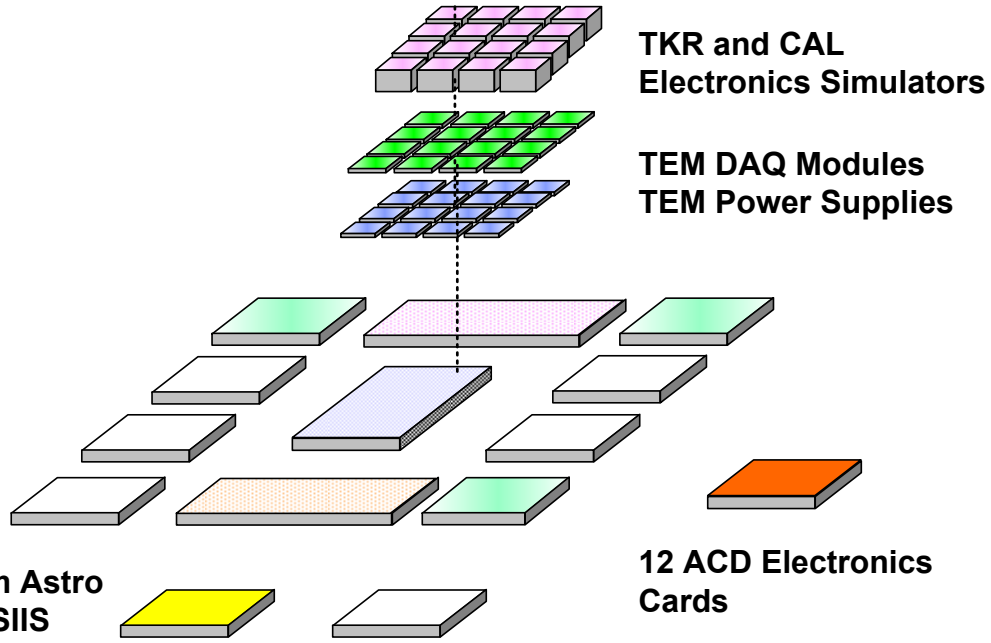
Cable management (as flight)





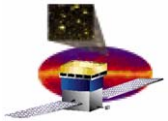
# Software Test-Bed for EM2/FU Tests

- Provides a full DAQ system with EM2 hardware (interfaces and functionality identical to flight) before flight hardware is available
  - 16 TEMs and 16 TEM power supplies
  - Full TKR and CAL front-end electronics for one tower
  - Front end simulators
  - Full ACD EM2 electronics
  - Full GASU and PDU
  - Full complement of CPUs
  - Spectrum Astro SIIS
- Excellent software test-bed



# Backup

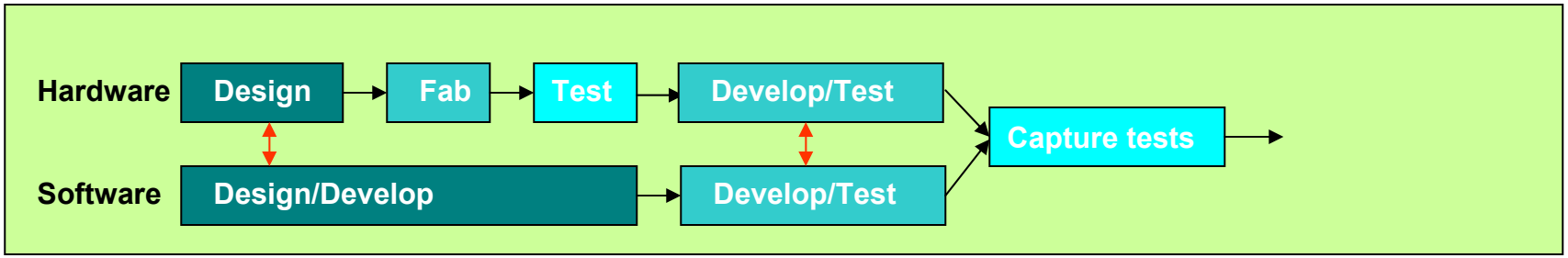
**Review material from former test presentations**



# Testing

- Unit Testing
  - Software only testing
  - Software testing on target hardware
    - Example - LCB package (LAT Communications Board)
  - Uses test executive (LTX)
- Build testing
  - Is tested on target hardware
  - Uses LTX/LATTE
- Formal Testing for ISIS & FU Releases
  - Includes
    - Formal requirement sign-off
    - Released test procedures
    - Released test report
  - If any test fails in formal testing:
    - Analyze failure
    - Review proposed fix
    - Implement fix
    - Perform regression testing

# Test & Verification




- Hardware and software development closely integrated
  - Design of hardware versus software complexity optimized continuously
  - Software runs at LAT engineering model electronics stage
  - Continuous hardware versus software verification
- Hardware and software development extends throughout the organization
  - Frequent hardware/software exchanges between collaborating institutions
    - For over a year, the following exchanges have been routine
      - ACD hardware, TKR hardware, CAL hardware: from institutions to SLAC
      - DAQ hardware, Flight software, I&T software: from SLAC to institutions
      - ACD Scripts, TKR scripts, CAL scripts, DAQ scripts: all to all
    - Provides independent verification process
  - Components integrated early and often
    - No surprises late in the game
- Code management already in operation at this stage
  - FSW code distributions are CVS tagged and annotated in a "builds" web page/document
  - FSW code has built-in self-identification (automatically generated by the build engine)
    - Can ask a running system what code/version is being run