SMC-IT Panel Discussions:

Space IT Systems Case Studies

Chair: Dr. Philip Varghese, JPL

ABSTRACT

Space Information Systems have evolved over the past several decades from simple toolsets in the early ages of space exploration into very complex systems encompassing an enormous breath of technologies. On the way, we have learned valuable lessons in designing, building, integrating and deploying these systems. With increasingly ambitious space missions covering greater distances from Earth and often operating in hostile environments, managing mission risks through increased but prudent application of intelligent space information systems has been gaining increased attention in the space exploration programs around the world. The lessons learned over the years are crucial in implementing space information systems of the future.

This panel will present case studies of actual Space Information Systems, outlining key properties of the systems, experiences gained from their use in the real operational environment, and the resulting feedback for future system development. The panel will include IT experts who participated in the implementation of space information systems at the NASA Centers as well as at international partner facilities for space exploration.

PANELISTS:

Anne Elson	-	Deep Impact (DI) Flight Mission
Earl Maize	-	JPL, Cassini Flight Mission
Karen Moe	-	GSFC, Intelligent data Understanding
David Nichols	-	JPL, IT & Software Systems Division
Keyur Patel	-	JPL, Space Infrared Telescope Facility (SIRTF)
Kanna Rajan	-	AMES, Autonomous Reasoning group
Glenn Reeves	-	JPL, MER Flight Mission
Jay Trimble	-	AMES

FYI - covered in another Doc review submission

Deep Impact Project
A Space IT Case Study
In Progress

July 14, 2003

Anne Elson
DI Project System Engineering Manager

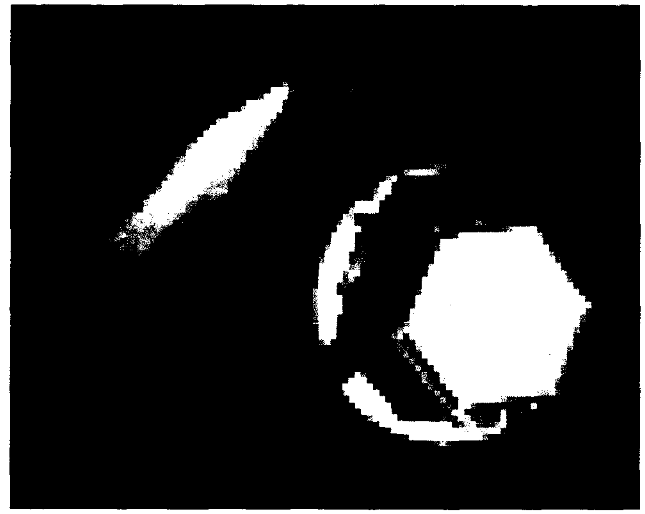
Agenda

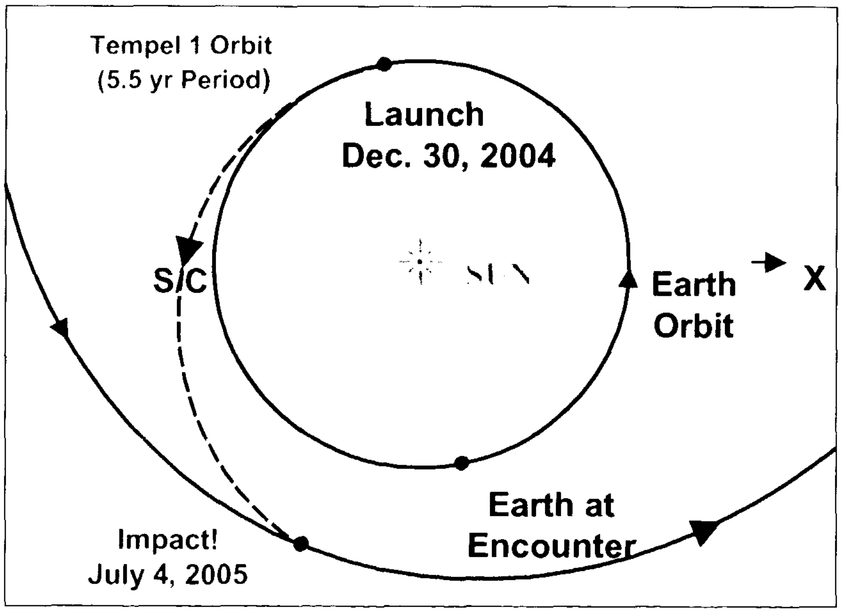
- Deep Impact Mission Overview
- Mission Objectives
- Programmatic Approach
- Flight Software Development Process
- Flight Software Architecture showing integrated BATC/JPL FSW
- CFDP
- Autonomous Navigation with Biased Scene Analysis
- System Fault Protection

Deep Impact Mission Overview

Salient Features

- Deliver a 370 kg impactor at 10 km/s to open the interior of a comet nucleus. Target is Comet P/Tempel 1.
- Impactor produces crater dependent on comet porosity and strength.
- Flyby spacecraft observes impact, crater development, ejecta and final crater with visible and IR multi-spectral instruments.
- On-board autonomous optical navigation used for precise targeting and control of impactor and fly-by spacecraft.
- 7 month mission duration. Launch: Dec. 30, 2004 Encounter:July 4, 2005





Science

- To determine the differences between the interior of a cometary nucleus and its surface.
- Determine basic cometary properties by observing how the crater forms after impact.
- To identify materials in the pristine comet interior by measuring the composition of the ejecta from the comet crater.
- Determine the changes in natural outgassing of the comet produced by the impact.
- To help discover whether comets lose their ice, or seal it in over time (evidence for dormancy vs. extinction).
- Address terrestrial hazard from cometary impacts

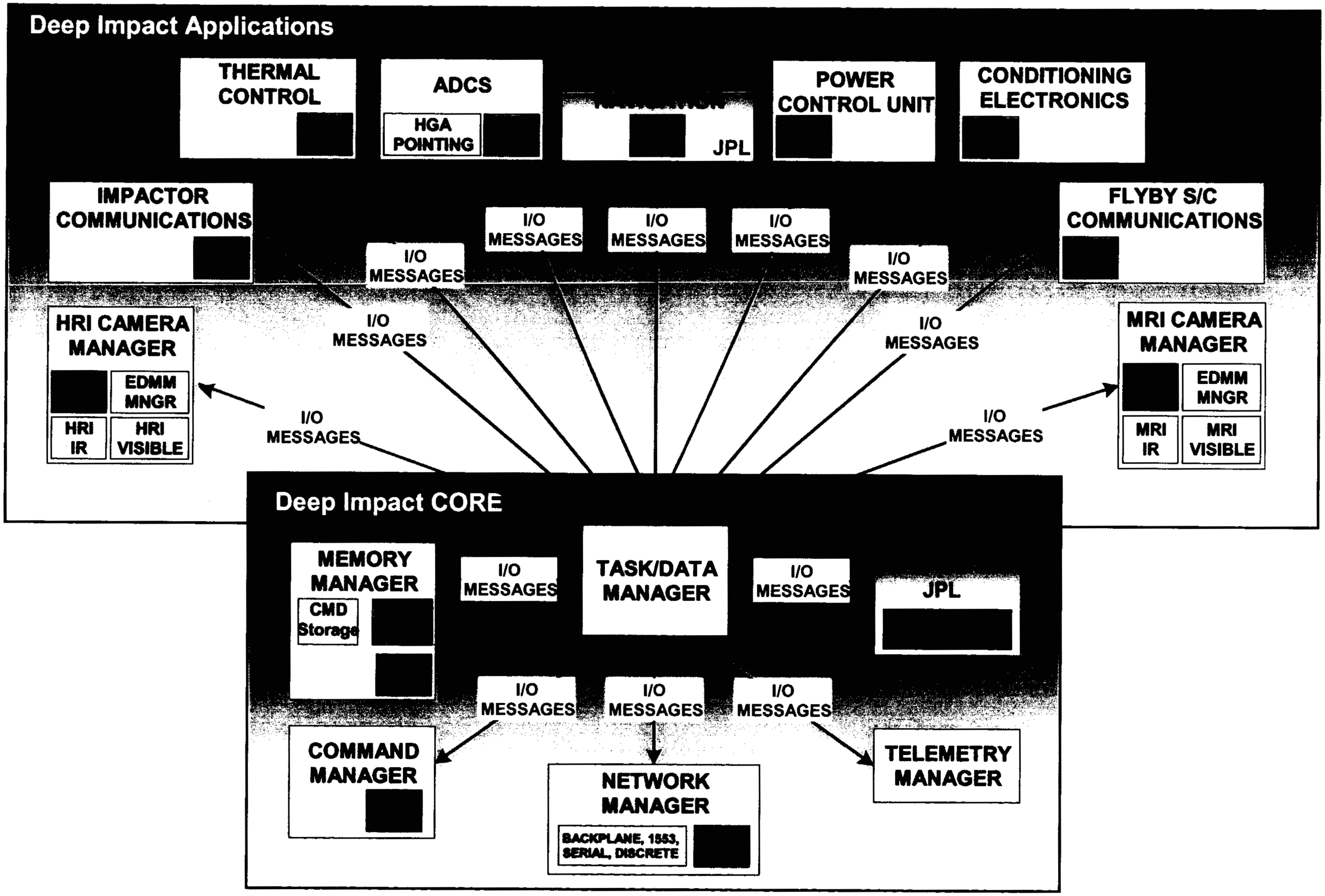
Deep Impact Programmatic Approach

- Deep Impact is 8th Discovery mission
- Project is a partnership between U of Maryland (UMD), NASA/Jet Propulsion Lab (JPL) and Ball Aerospace and Technology Corporation (BATC).
- Implementation mode:
  - System contract with most flight hardware and flight S/W developed at BATC
    - JPL GFE's computers, memory, X-band transponders, TWTA's, S-band transceivers and antennae
    - JPL provides autonomous navigation, fault protection, CCSDS File Delivery Protocol architecture and software
    - Contract for science instrument development at BATC through UMD
  - Software integration at BATC (JPL deliverables integrated at S/S level at JPL), validation at BATC and JPL on high fidelity testbeds
  - System integration and test at BATC
  - Mission operations at JPL, supported by BATC personnel at JPL for critical events
  - Science team, and outreach managed by U of Maryland

Flight Software Development Process

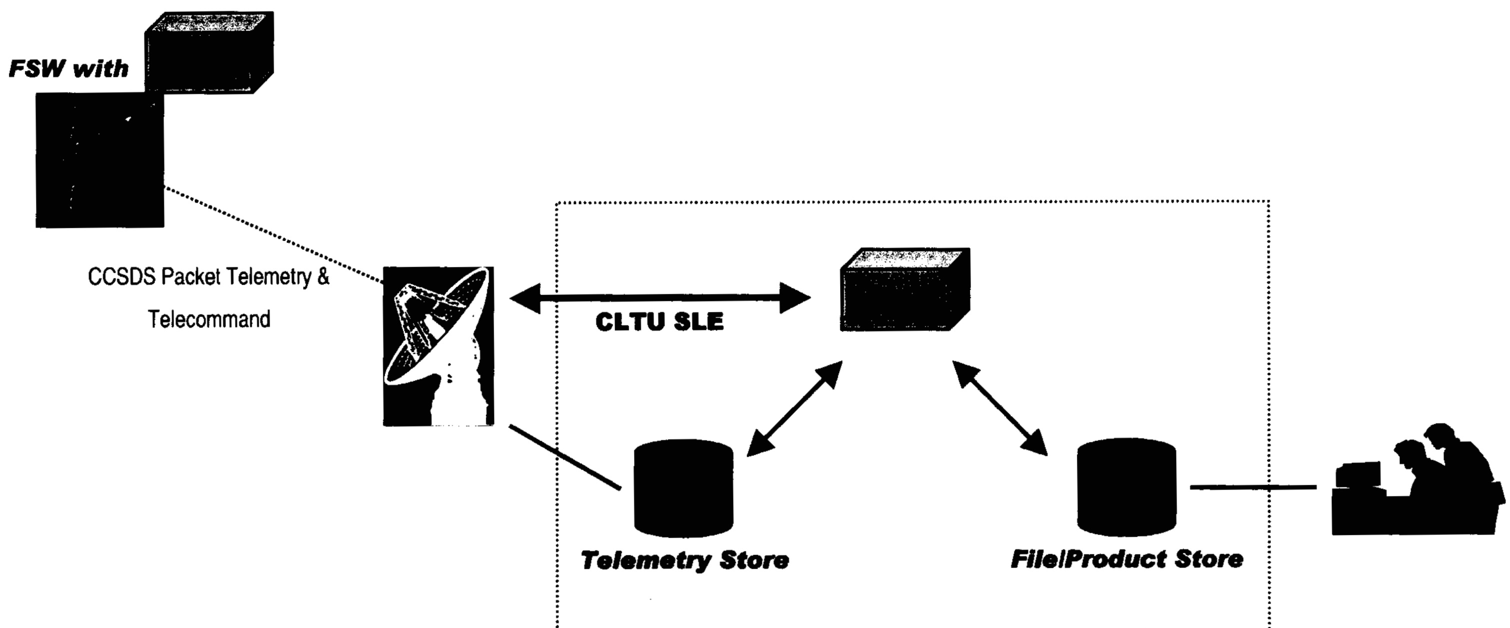
- Coordinated JPL/BATC Software Management Plans
  - Compliant with NASA/JPL SW Process and Product Requirements
    - NASA Directives for SW development (NPD 2820.1, NPD/NPG 8730)
    - D-15378, JPL Software Development Process Description
    - D-17868 Appendix A, S/W Development Principles for Flight Systems
- Shared SW Development Processes including requirements, design, implementation approach common to both teams
- Team Testing Approach
  - Unit testing, integration testing, S/S functional testing local to team
  - Final software subsystem testing at BATC (part of SW FQT)
- Coordinated Software Configuration Management and Change Tracking
  - Software Configuration Management tool: CVS
  - SW Problem Report and SW Change Tracking Tools
- Coordinated SW Risk Tracking and Management
- Verification and Validation process well defined
  - Internal V&V (iV&V) testing of software system at JPL and BATC
  - Extensive JPL and BATC Software QA effort
  - Independent V&V (IV&V) analysis of software system by NASA IV&V
- S/W approach fully reviewed and approved by internal and outside boards

JPL BATC DI Integrated FSW

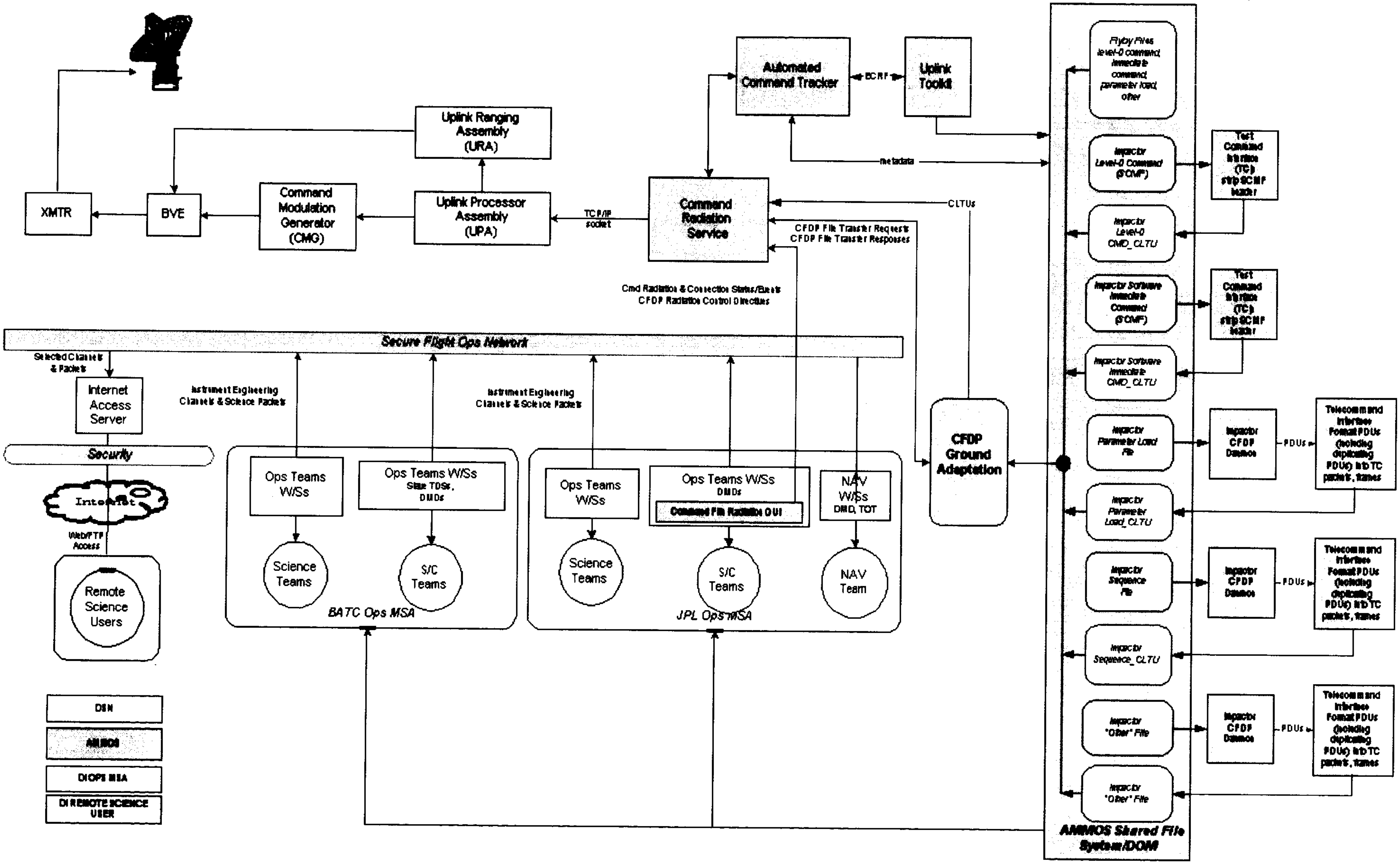


CFDP Scope and Content: An Overview

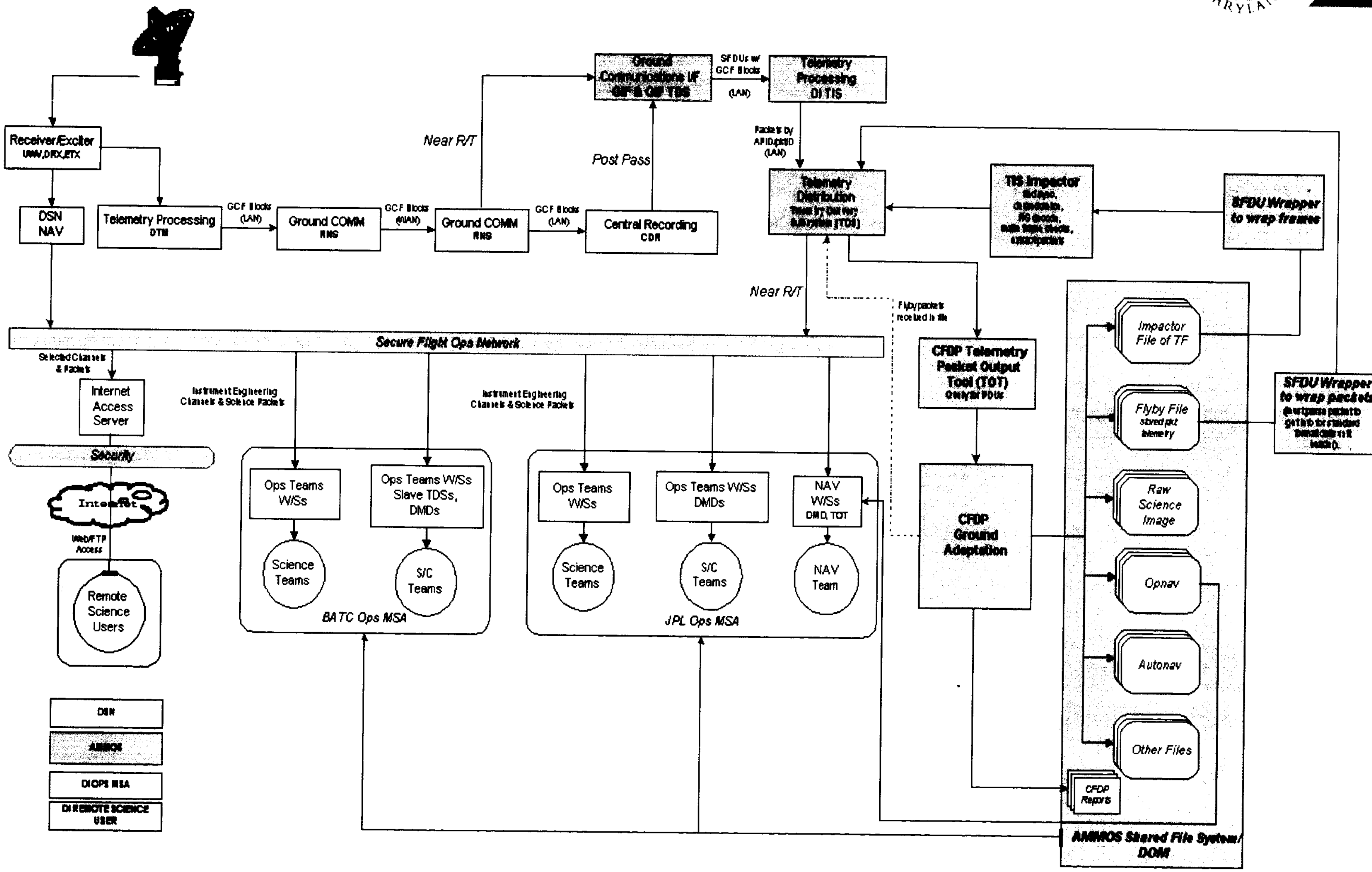
- CCSDS File Delivery Protocol (CFDP)
  - Reliable & unreliable file transfer from spacecraft to mission database (and vice versa)
  - CFDP Core Software will be integrated into both the DI Flight Software and DSMS Ground Systems
- CFDP Core Software controls:
  - File transfer, file assembly, local notifications, and submission of files to the DSMS Data Management System (DOM)
- The CFDP Protocol Data Unit (PDU) is central to this protocol
  - File and its metadata are segmented into PDUs and PDUs are transferred over link layer



Uplink Dataflow



Downlink Dataflow



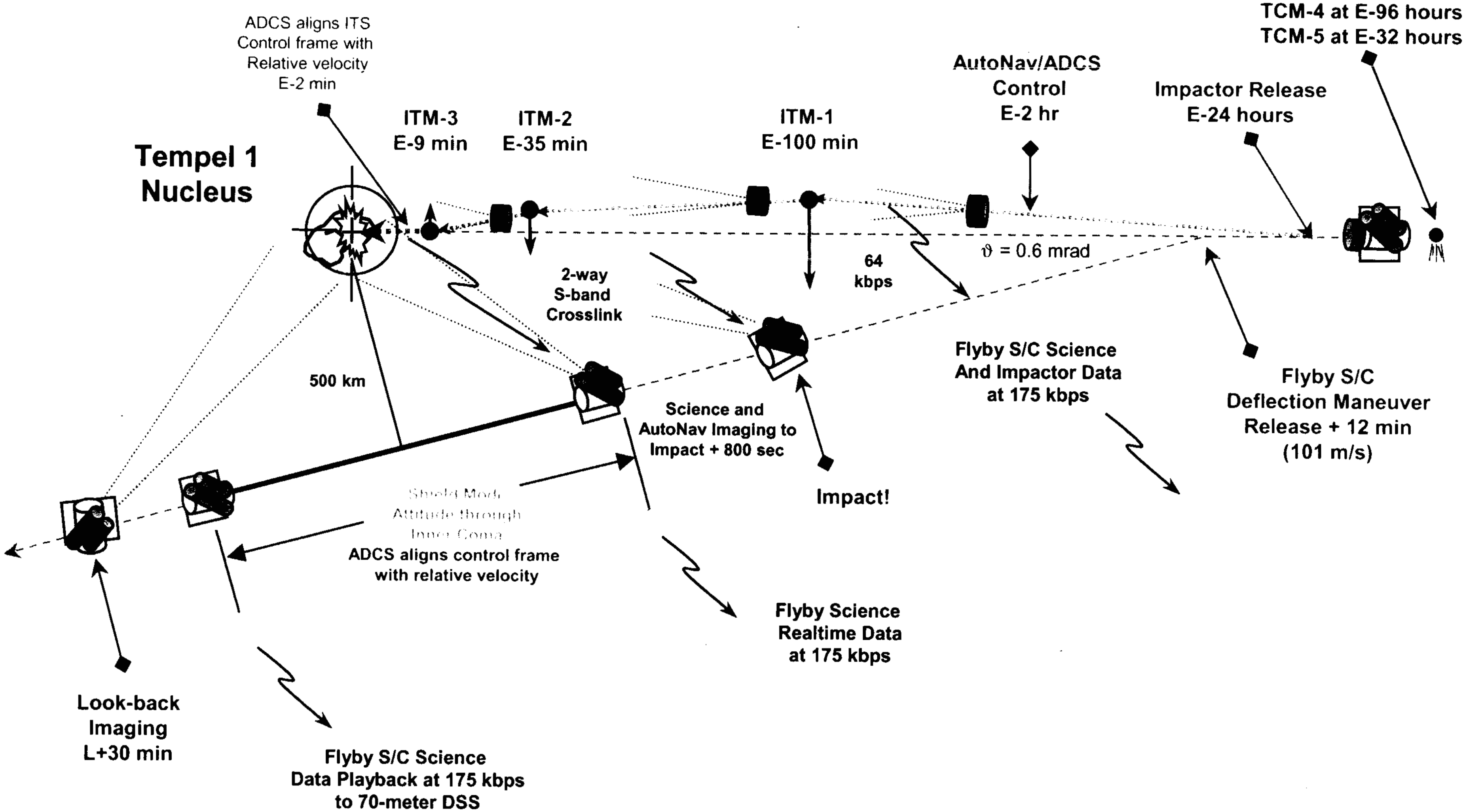
AutoNav at Encounter (E-5 days to E+1 day)

- Events: TCM-4 (E-4 days), TCM-5 (E-32 hrs), impactor release (E-24 hrs), flyby deflection maneuver (separation + 12 min), 3 impactor targeting maneuvers (ITMs), science and opnav imaging, impact event (4 Jul 05), transition to shield mode for coma crossing (E+800 sec), data playback and lookback imaging

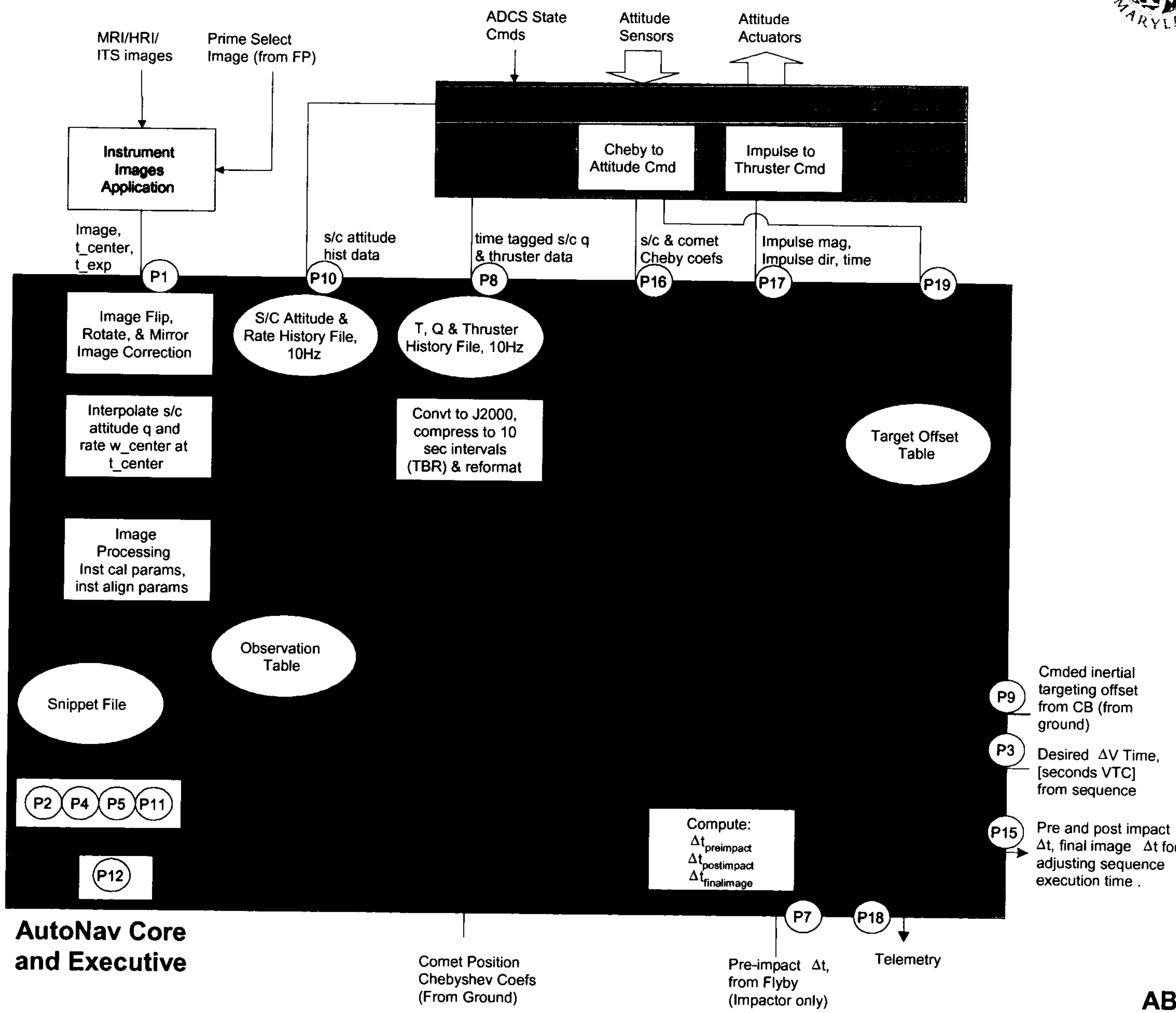
- Attitude: continuous comet pointing (HRI/MRI boresight on comet, min. +Y to Sun) for flyby spacecraft; ITS boresight on comet for impactor

- DSN Coverage: continuous 34m coverage to E-2 d; continuous 70m coverage from E-2 d to E+1 d; 34m arrayed to a 70m equivalent at impact

Encounter Schematic



ADCS AutoNav SW Interfaces



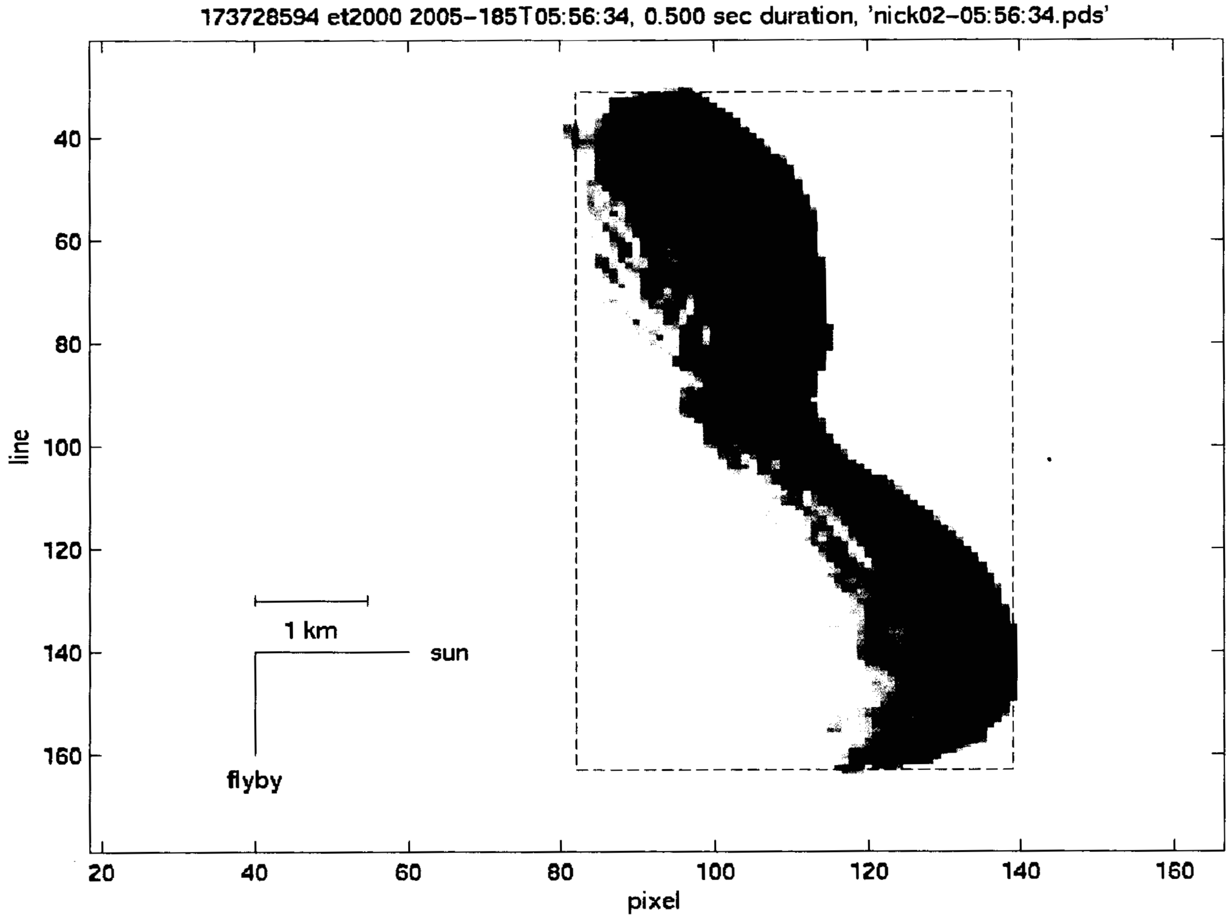
AutoNav Biased Scene Analysis

- Image is raster-scanned to find best location
  - Selection depends on predefined circular area around each pixel
  - Size of circular area is parameter driven
  - Area of lit pixels, shadowed pixels (on-nucleus), and dark pixels (off-nucleus) is computed
  - Integrated brightness of pixels within circular area is computed
- Site selection determines whether current location is better than "best-so-far"

Decreasing Priority ↓

  - Greater lit area (Highest priority)
  - Smaller dark area (Second highest priority)
  - Closer to Flyby's closest approach point ⇐ Biased Scene Analysis
  - Lower integrated lit pixel brightness
  - Greater integrated shadow brightness
- Result is a 2-D offset in pixel and line from the center of brightness
  - Pixel,line offset converted to 3-vector in inertial coordinates with units of kilometers (km)
  - Offset sent to ADCS for pointing correction and used by AutoNav core, on the Impactor, for maneuver computation

Trace Progression of Biased Scene Analysis Algorithm



C-language (flight code) version of Biased Scene Analysis:

– Yellow marker near center is center-of-brightness

– Green trace is best-so-far locations as Biased Scene Analysis scans image

– Red marker near bottom is biased SA's selected target location

Fault Manager CSC
High Level Functionality & Purpose

- Supports both spacecrafts, Flyby and Impactor.
- Implements system level autonomous fault recovery allocated to software.
- Maintains configurable mapping between symptoms (monitor output) and associated faults and between faults and associated responses.
- Provides centralized fault protection engine that coordinates and tracks fault response execution based on monitor output.
  - Design inherited from Deep Space 1
- Provides commands and telemetry to support spacecraft fault management control and status.
- Initialization:
  - Executes at 1 Hz.
  - APR initialization varies based on boot type (first vs. cold)
    - Mission Context Flag – allows mission mode dependent response behaviors
    - Fault enable status

Fault Manager CSC Software Diagram