



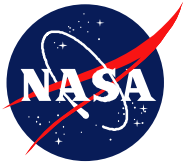
Security, Monitoring, and Control of the Re-engineered Hubble Space Telescope Control Center System

Caleb Principe, NASA Goddard Space Flight Center

Larry Barrett, Orbital Sciences Corporation

Thomas Buchanan, QSS Group Inc.

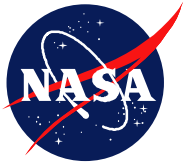
Jay Lockwood, Lockwood Software



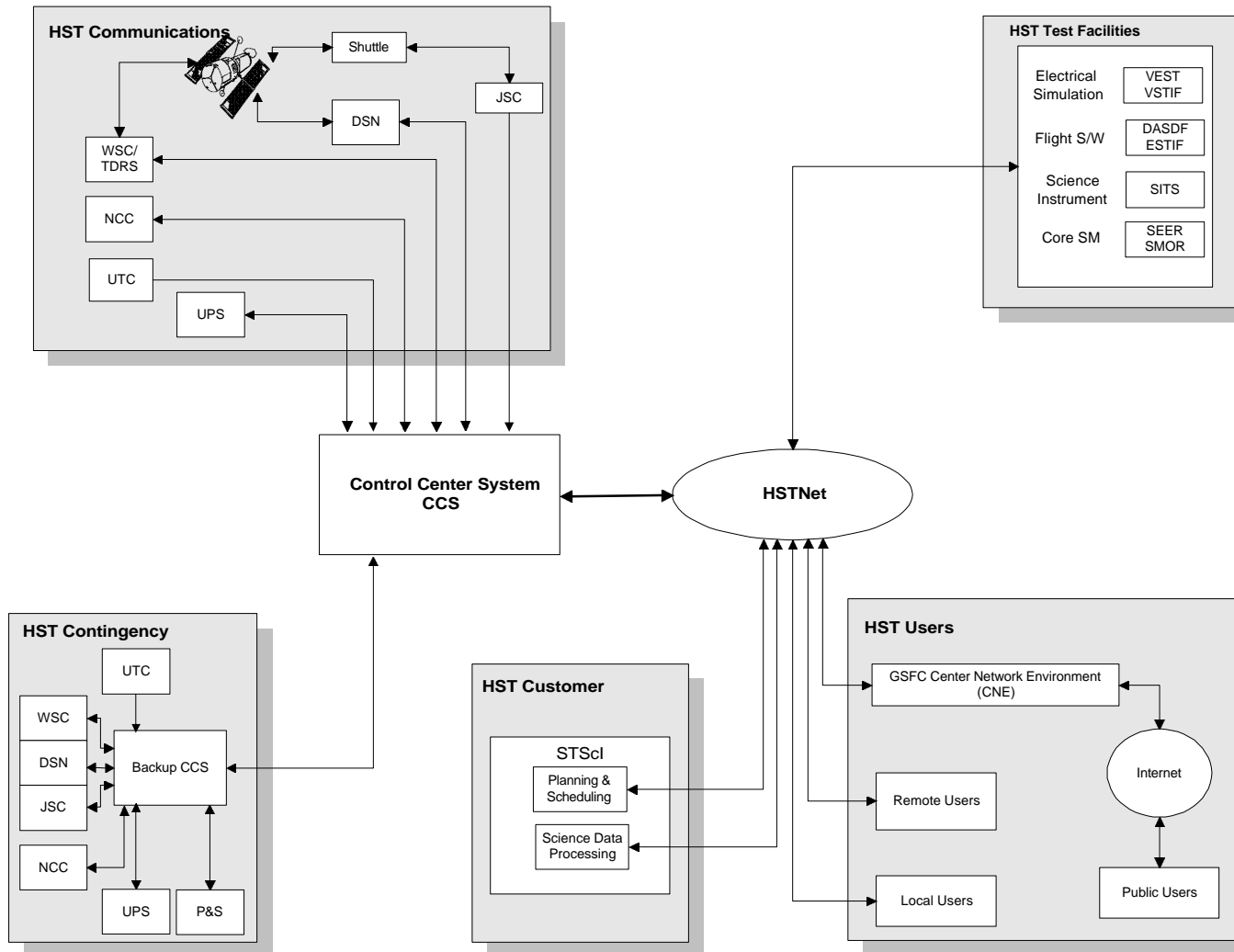
CCS Overview



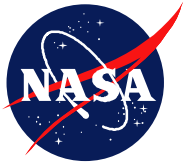
- ◆ Hubble Space Telescope (HST) Control Center System (CCS) is one component of a larger, more complex spacecraft management system
- ◆ CCS provides the following functions:
 - Spacecraft Communications Coordination
 - Spacecraft Commanding
 - Spacecraft Health and Safety Analysis
- ◆ CCS does not perform:
 - Scheduling of Spacecraft Observations or Resources
 - Processing or Distribution of Downlinked Science Data



CCS Overview



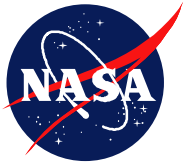
February 25, 1998



Goals of CCS Re-engineering



- ◆ Significantly Reduce Cost of Operations by:
 - streamlining business processes for normal operations
 - automating routine and repetitive operational procedures
 - providing secure, remote access to system resources
 - maximizing utilization of spacecraft resources
- ◆ Reduce Maintenance Costs by:
 - utilizing state-of-the-practice technologies and methods
 - adhering to government and industry standards in development
 - cost-effective use of off-the-shelf (OTS) components
 - building fault-tolerance into the system architecture



System Concept Drivers



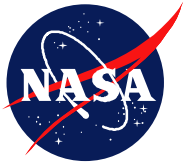
- ◆ Architectural
 - modular and extensible to facilitate maintenance and reuse
 - scalable to allow deployment of functional subsets
- ◆ Operational
 - automate ground-system operations, provide manual override
 - engineering expertise captured in on-line knowledge-bases
- ◆ Developmental
 - integrated development environment established to maximize productivity
 - integrated product team (IPT) based organization instituted to minimize implementation errors



Target Environments



- ◆ Operational Environments
 - Highly distributed, server-class processors used for:
 - ◆ Spacecraft control and monitoring
 - ◆ CCS system maintenance
- ◆ Test Facilities
 - Small number of co-resident processors used for:
 - ◆ Flight software development
 - ◆ Spacecraft anomaly isolation and resolution
- ◆ Stand-alone Configurations
 - Single processor configuration used for:
 - ◆ Science instrument development and check-out



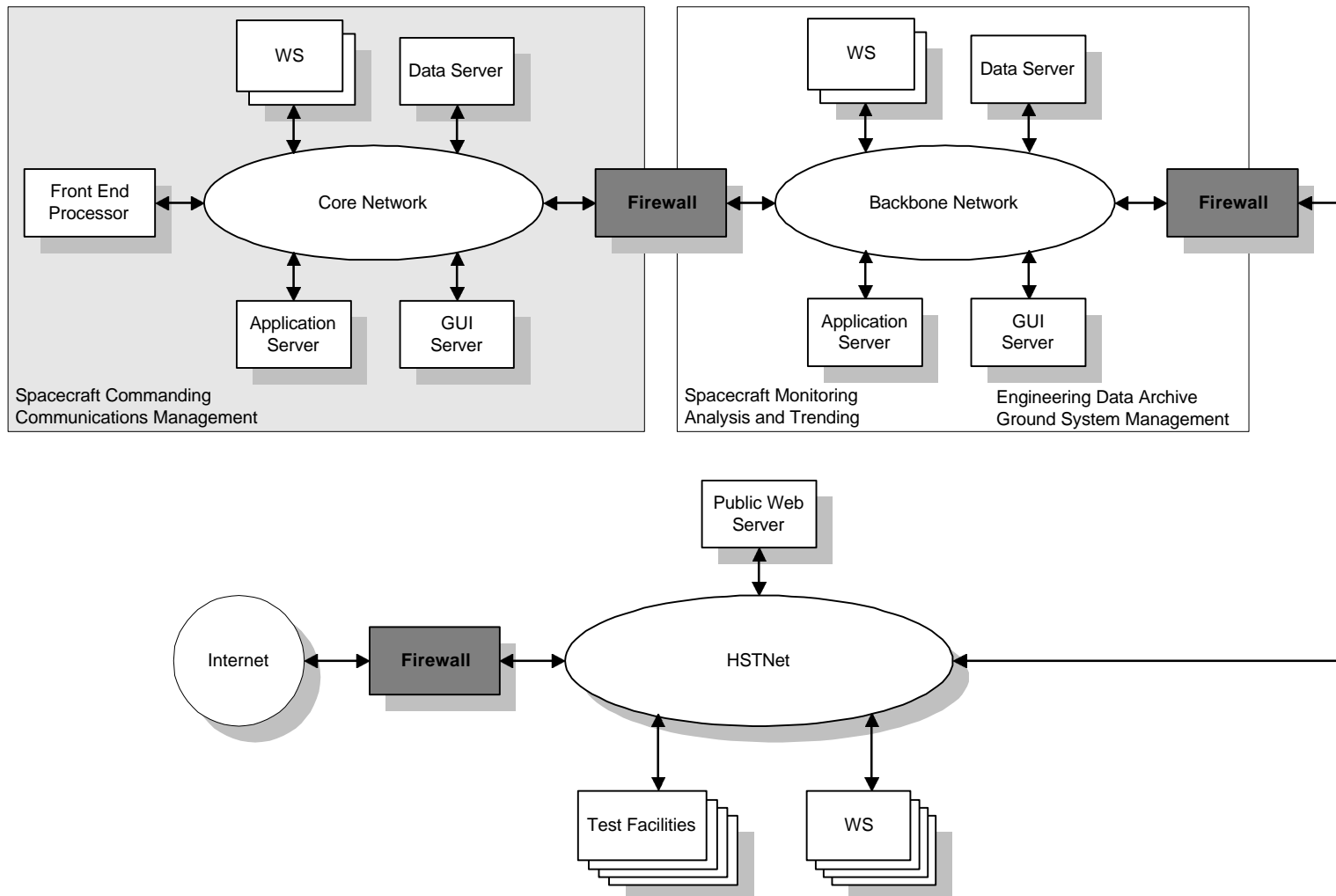
CCS System Architecture

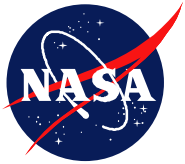


- ◆ System partitioned into three functional segments:
 - Command and Communications
 - Engineering Data Processing
 - User Workstations
- ◆ Logical Processor concept used to enable scalability
 - highly cohesive set of functions
 - decoupled through use of middleware
 - independent of physical nodes
- ◆ Data Driven architecture supports tailoring and reuse
 - configuration database drives most system functionality



CCS System Architecture

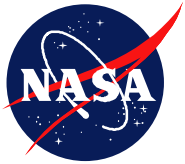




Development Methodology



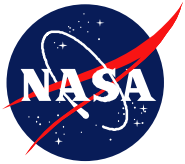
- ◆ Hybrid methodology established using best of:
 - Business Process Re-engineering
 - Top-Down Functional Decomposition with Data-Flow Analysis
 - Thread-based Dynamic Behavior Models
 - Object-Oriented Analysis/Design
 - Entity-Relationship Modeling
- ◆ Methods adopted with elements from waterfall, incremental, and spiral approaches
 - Applied method that best fit the development of the target product
 - Development Environment/Tools tailored to support project specific needs



Technological Enablers



- ◆ **Middleware**
 - Encapsulates interprocess communication methods
 - Nameserver provides directory of software applications
- ◆ **Security Firewalls**
 - Applications unaware of Firewall in communication path
 - Unnecessary for reduced configurations
- ◆ **Web Servers and Browsers**
 - Common user interface across multiple workstation platforms
- ◆ **Automated System Monitors**
 - Distributed resource monitoring and failover support



OTS Component Integration



- ◆ Functional prototyping used to assess candidate products
 - Provided method of identifying best-of-breed
- ◆ Primary selection criteria included:
 - adherence to appropriate standards
 - scalability
 - maintainability
 - compatibility with other products
- ◆ Encapsulation used to insulate applications from OTS product features
 - Rogue Wave libraries and custom software used



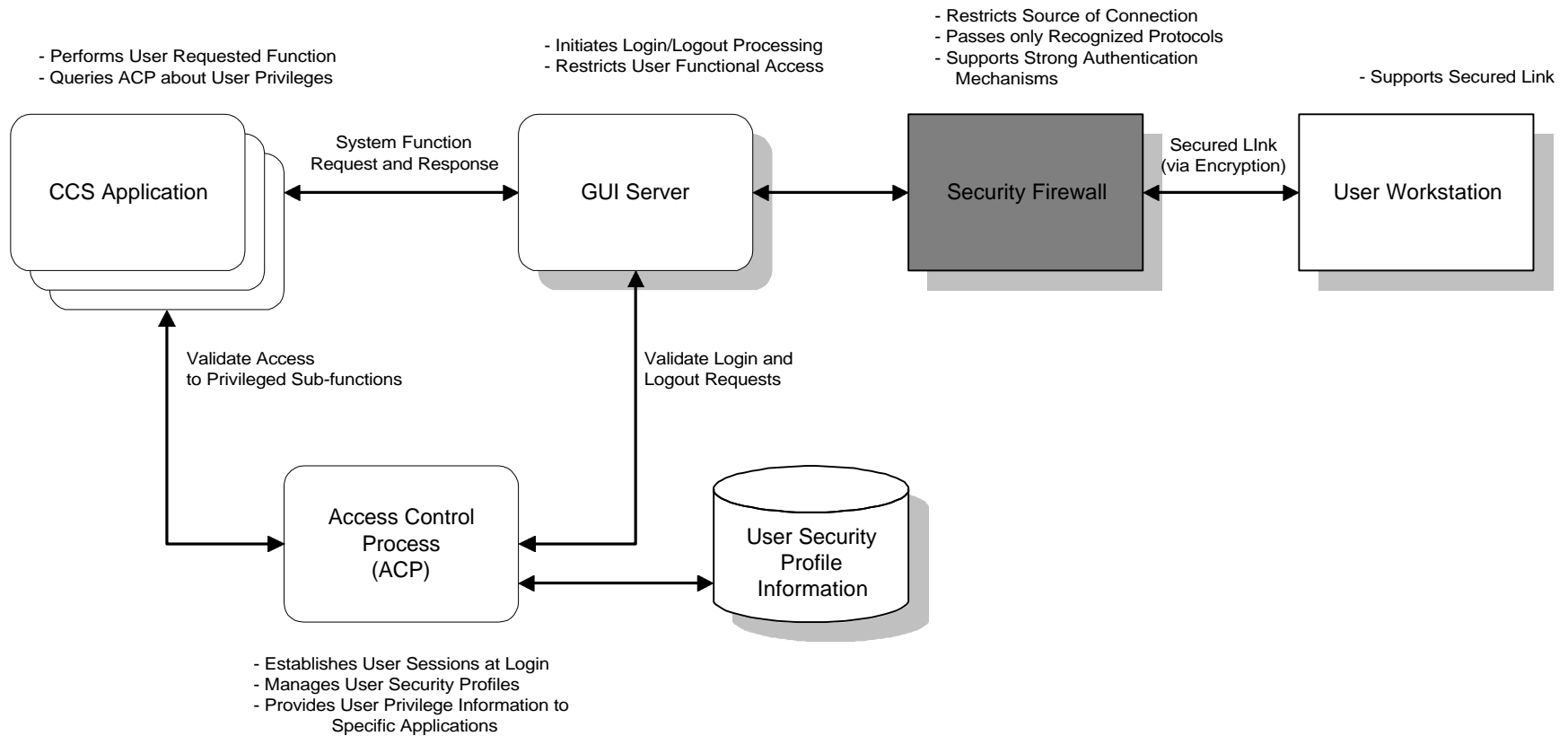
Security Considerations



- ◆ Security built-in from the beginning
 - Guideline: *“Prohibit what is not explicitly allowed”*
 - Drove network topology and functional allocation
 - Eliminated some OTS products from consideration
- ◆ Implementing security concurrently with applications simplified system integration process
 - Stateful-inspection firewall technology supported scalability
 - Security concerns detected and corrected immediately
- ◆ Functional access controls implemented at application level



Functional Security Architecture

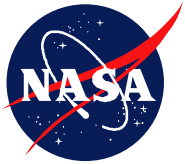




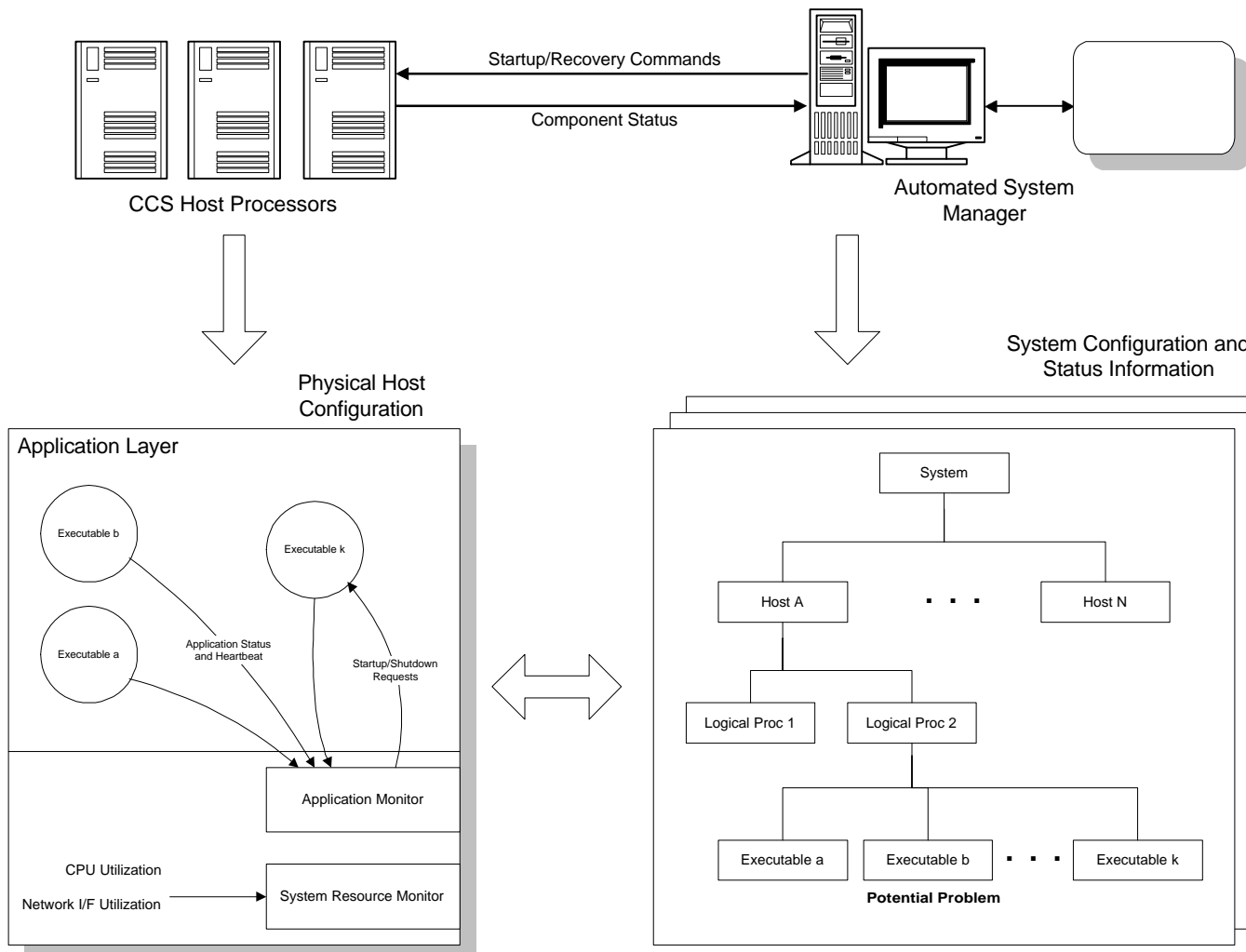
System Management Approach

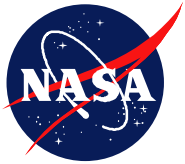


- ◆ Application dependency information used to automate startup and failover sequences
 - allows system to operate in multiple valid configurations
- ◆ Local monitoring of each resource performed
 - COTS products used to monitor system resources and applications
- ◆ Centralized analysis engine used to identify and recover from suspected failure conditions
 - Knowledge-base integrates system-wide monitoring information to determine probable source of detected error conditions
 - Recovery measures range from simple application restart to full processor failover



System Management Architecture





Current Status



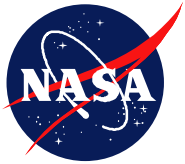
- ◆ Preparing for sixth incremental Release in 18 months
- ◆ CCS has been successfully delivered in operational, test facility, and single processor configurations
- ◆ Preparing for switchover to CCS for Servicing Mission 3
 - from ‘shadow’ mode operation to primary spacecraft control system
- ◆ Independent Assessment of Security Architecture pending
- ◆ System Management architecture is finalized
 - system dependency analysis complete
 - knowledge-base population and validation in progress



Conclusions



- ◆ A configurable, extensible ground system architecture can be developed
 - Use of OTS products reduces development time and costs
 - Design based on Logical Processors can facilitate scalability
 - Early emphasis on security drives other architectural decisions
 - Browser-based user interface provides flexibility
 - Automated System Management functions are reaching maturity
 - Tailored methodology and tools are critical to meeting development goals



Contact Information



- ◆ URL <http://ccs.hst.nasa.gov>
- ◆ e-mail
 - caleb.principe@gsfc.nasa.gov
 - larry.barrett@gsfc.nasa.gov