Leveraging Serial Digital Interfaces Standardization

The RASTA Reference Architecture Facility at ESTEC

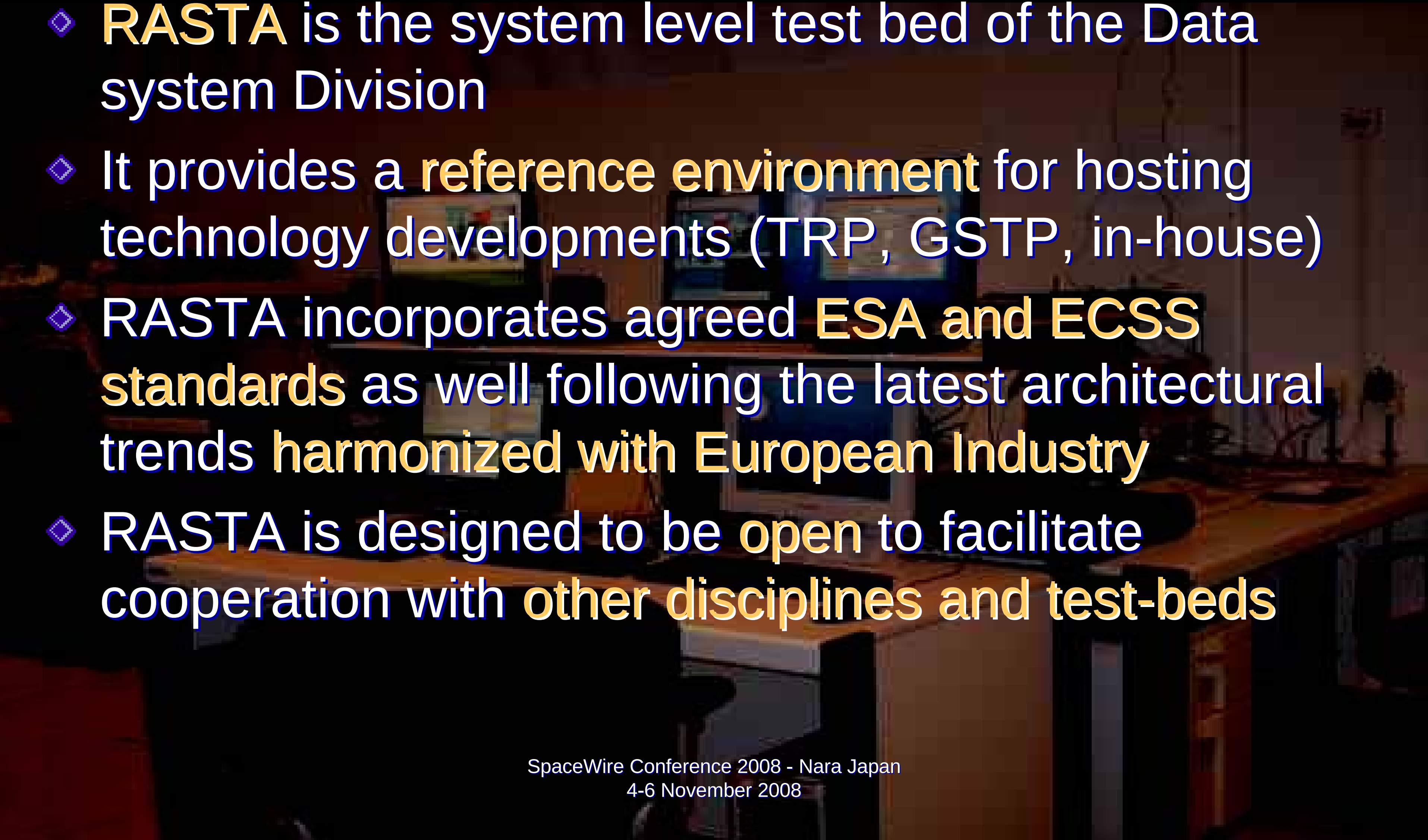
Farid Guettache, Aitor Viana Sanchez, Massimiliano Ciccone, Claudio Monteleone, Chris Taylor ESA/ESTEC
Manual Prieto, Ignacio Garcia Tejedor University of Alcala Madrid Spain

SpaceWire Conference 2008 - Nara Japan
4-6 November 2008

Content

- What is RASTA
- Objectives
- Reference Architectures
- Data systems Testbed
- Current Hardware configuration
- Current Software configuration
- Current Developments
- Conclusion

RASTA



- RASTA is the system level test bed of the Data system Division
- It provides a reference environment for hosting technology developments (TRP, GSTP, in-house)
- RASTA incorporates agreed ESA and ECSS standards as well following the latest architectural trends harmonized with European Industry
- RASTA is designed to be open to facilitate cooperation with other disciplines and test-beds

Objectives -1

- Early & Rapid prototyping of new avionic architectures (HW,SW and HW+SW)
- Test & Demonstration of developed technology in representative environment
- Testing & Demonstration in areas of standardization (CAN, SOIS, Mil-1553)
- Project support: benchmarking, independent analysis

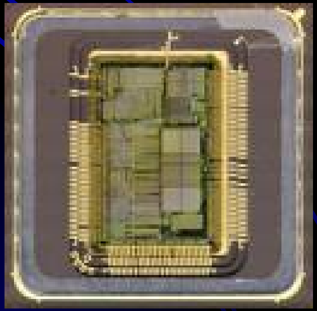
Objectives -2

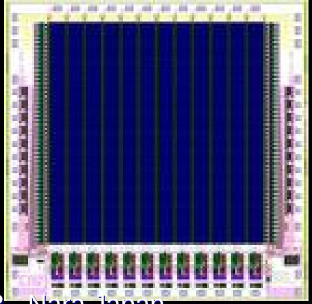
- Assessment of performances (with HW in the loop)
- Validate new trends before implementation (CAN, protocols, sensor bus, …)

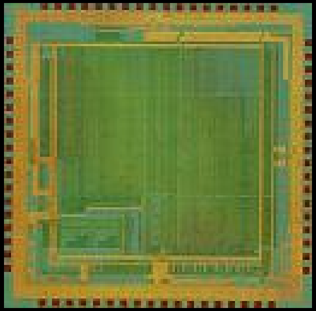
Reference Architecture

Integrated Architecture

- Core functions combined in a few large ASICS, providing enhanced reliability and small footprint
- Essentially a ‘System On Chip’
- The core DHS function (S/C controller) are mission independent

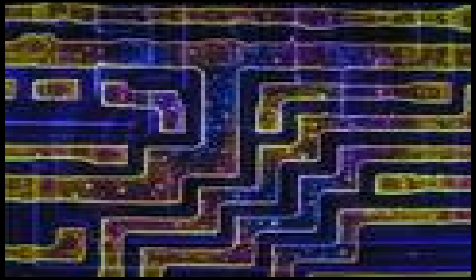




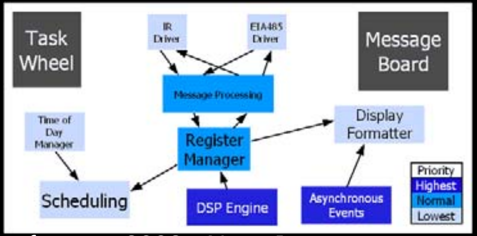


Decentralised Architecture

- Data System implemented as a set of processing nodes communicating through a (high performance) network



- High speed microprocessors and fast inter-processor communications supported by Leon, and SpaceWire



Integrated or Decentralised?

- There is no single architecture that fits all needs, Rather, the on-board DHS architecture can be a compromise between two models:
  - Highly integrated and Decentralised

- Rasta must provide capabilities to support both models

The Test bed for Data Systems -1

To setup the testbed we have to consider

A- **Repository of Building blocks**

- The individual components developed for space usage (Processors, IP cores)
- The functional blocks that comprise the onboard data system
- The way in which the functional blocks are interconnected and assembled to form an onboard architecture

The Test bed for Data Systems -2

B- Deployment and configuration of selected components and architecture

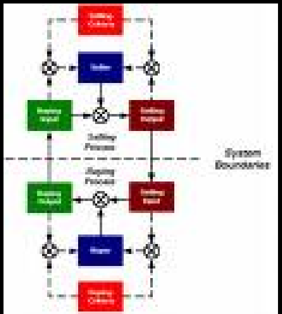
- Boards and racks
- Software layers

C- Development

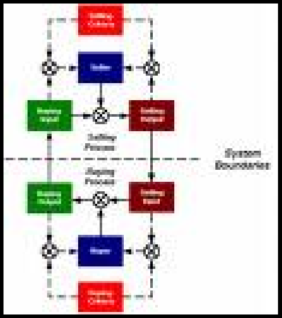
- Application layer support component like CFDP and SOIS
- New Functional block like the Mass Memory

Data Systems Components

- Processor: ERC32, LEON, COTS
- Interfaces components: Milbus, CAN, SpaceWire, …….
- Memory: Flash, SDRAM,…..
- IP cores & dedicated ASICs: TM, TC, RTC, …..



Data Systems Functional Blocks

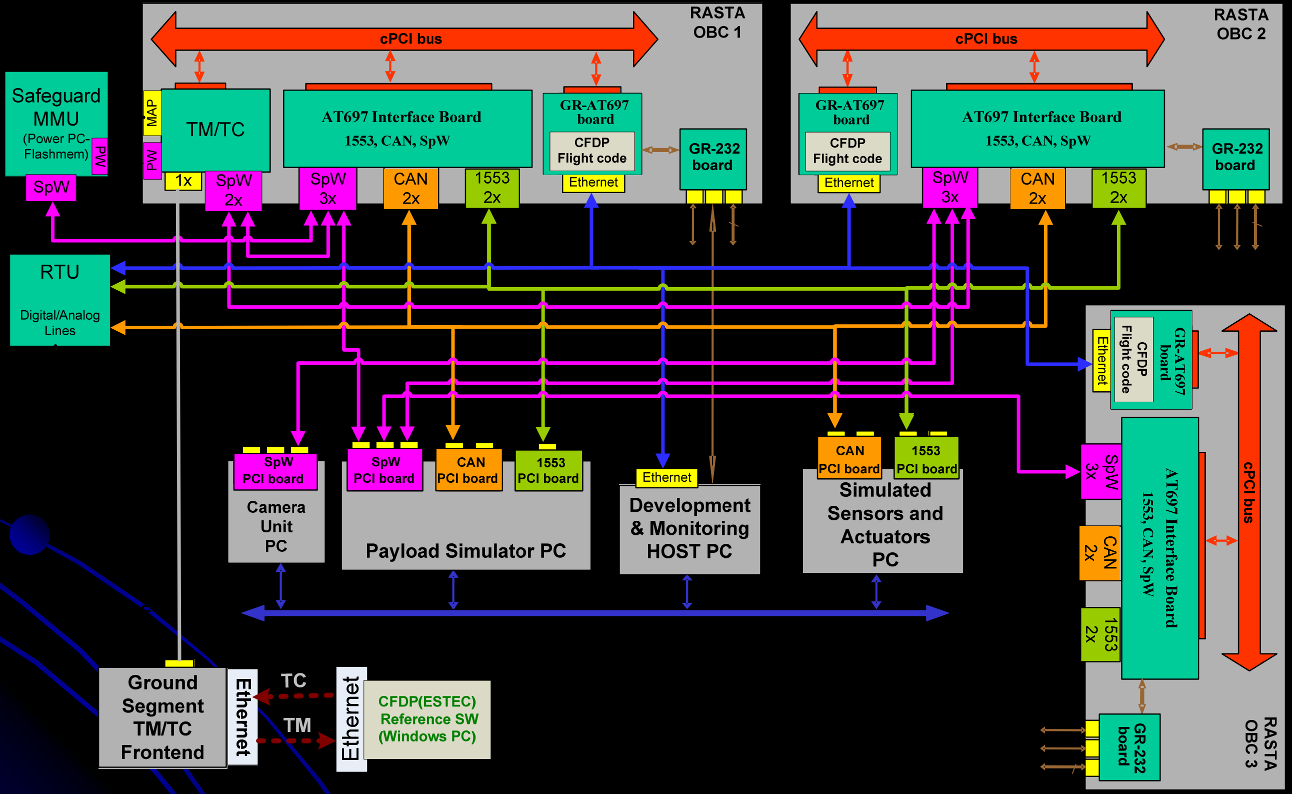


- Processor Module
- I/O Modules
- Data Storage: Mass Memory
- TMTC system
- Communication Protocols: space link & onboard
- Reconfiguration Module , safeguard memory

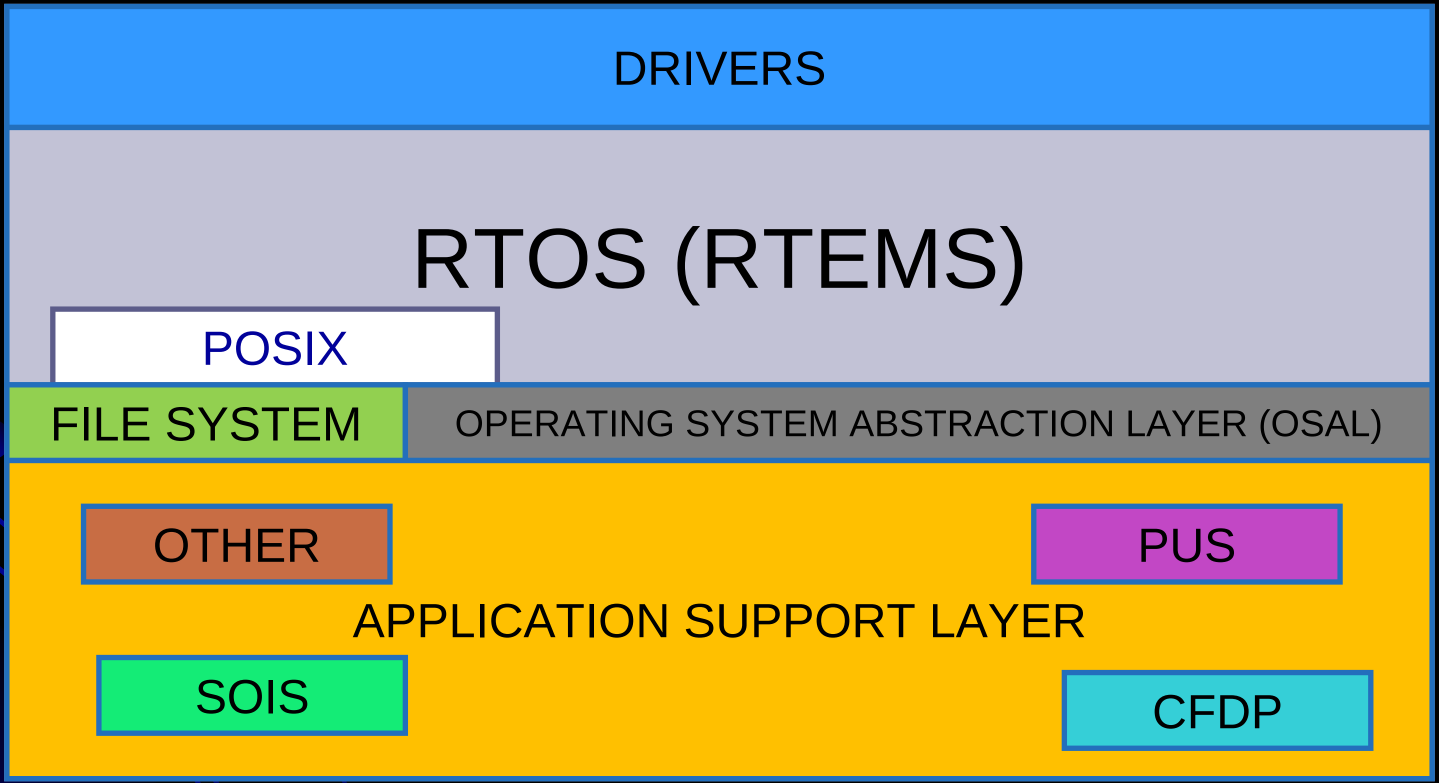
Data Systems Connectivity

- I/O System architecture: Bus, network, point to point
- Redundancy Architecture, Primary Chain/Secondary Chain,
- Cross-coupling/Cross-Trapping & Back plane
- TMTC input output

Current Hardware Configuration



Current Software Configuration



DRIVERS

RTOS (RTEMS)

POSIX

FILE SYSTEM

OPERATING SYSTEM ABSTRACTION LAYER (OSAL)

OTHER

PUS

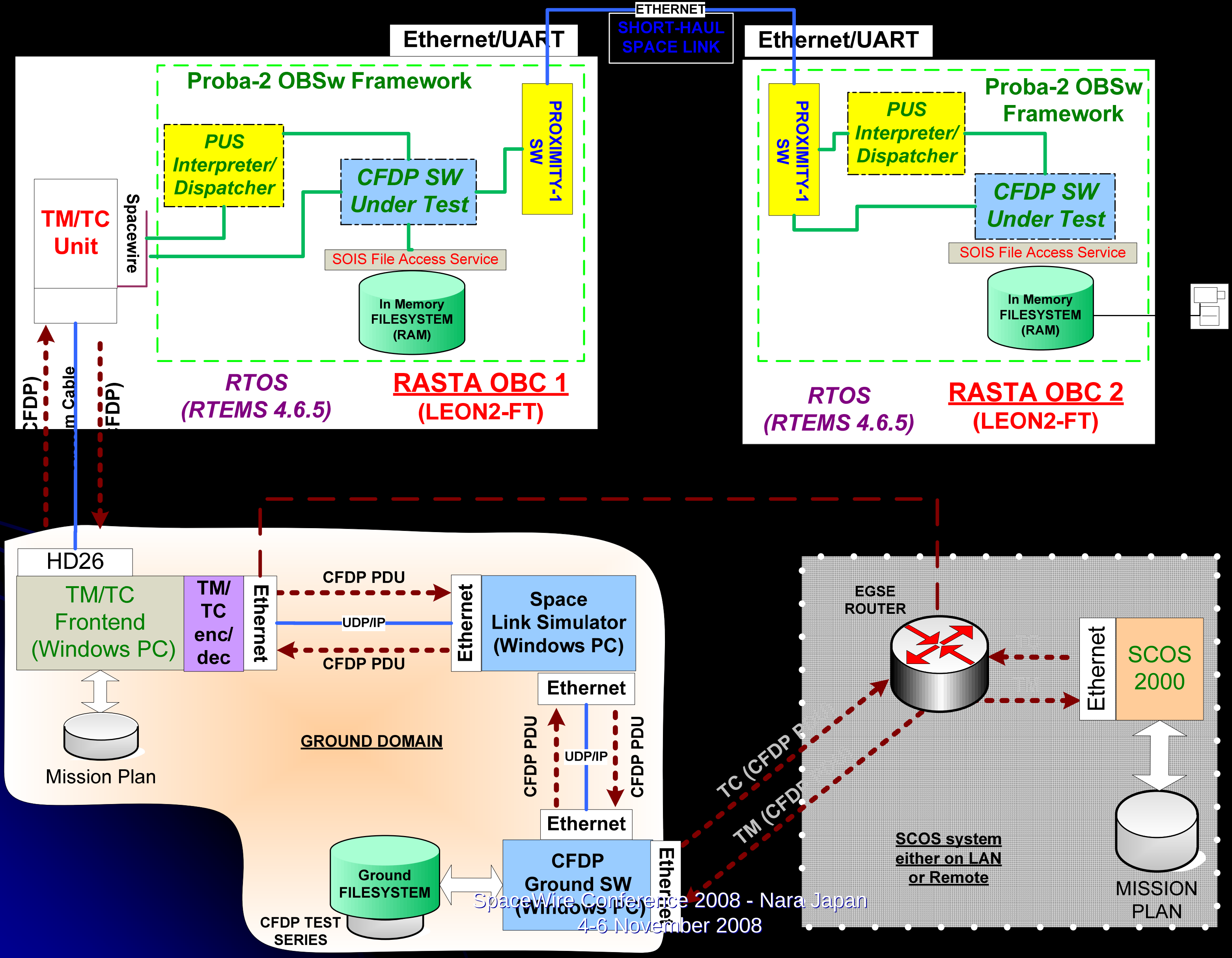
APPLICATION SUPPORT LAYER

SOIS

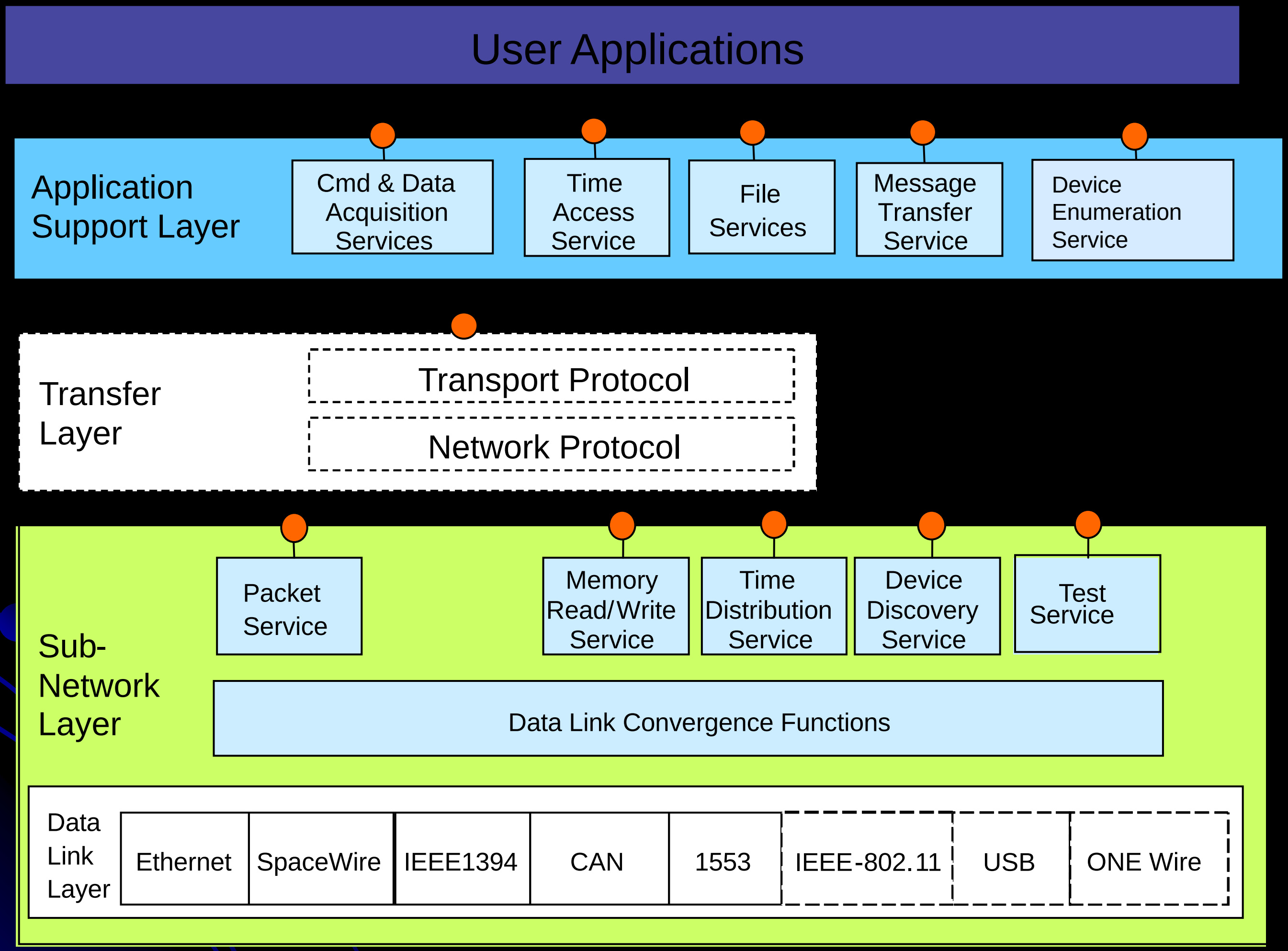
CFDP

Current Development

CFDP end to end application test



SOIS Implementation



Mass Memory

- There are several issues related to the use of Mass Memory that require evaluation:
  - Access – file system, packet store, raw
  - Dumb or intelligent
  - Connectivity – direct downlink, via DHS
  - Architectural - dedicated to payload, partitioned
- Ultimately we probably need a combination of the above

Modular Advanced Mass Memory Architecture

- Independent from memory device technology (SDRAM, Flash ..)
- Connectivity based on Spacewire
- Implementing advanced concept

Conclusion -1

- Previous/Ongoing activities
  - Support to CCSDS/ECSS standardisation process
  - Demonstrating and validating CFDP implementations in end to end configuration
  - SOIS Implementation
  - Mass Memory
- Future activities

Conclusion -2

- One wire sensor bus implementation
- Incorporation of Wireless interface
- Evaluation of ExoMars CFDP configuration
- SpW components and Protocols in backplane, sensors, distributed architecture