Safety-critical Partitioned Software Architecture

Seung Chung, Dan Dvorak, Dave Hecox, Greg Horvath

Jet Propulsion Laboratory
California Institute of Technology
2009-11-05

Acronyms

A653 - ARINC 653

ApEx - Application Executive

FSW - Flight Software

HM - Health Monitor

IMA - Integrated Modular Avionics

OS - Operating System

RTOS - Real-Time Operating System

SPSA - Safety-critical Partitioned Software Architecture

Overview

- Introduction: A departure from "Business as Usual"
- Background: ARINC 653
- A New Approach: Partitioned FSW Systems
- Issues, Challenges, and other Fun Things

Introduction

Business As Usual

- For the past 10+ years, most JPL flight software has followed the 'monolithic image' paradigm
  - All modules linked together into a single executable image
- These systems tend to be:
  - Message-based
  - Highly event-driven
  - Heavily reliant on interrupts

Business As Usual

- For the past 10+ years, most JPL flight software has followed the 'monolithic image' paradigm
  - All modules linked together into a single executable image

Business As Usual

- For the past 10+ years, most JPL flight software has followed the 'monolithic image' paradigm
  - All modules linked together into a single executable image

✓ Makes efficient design easier to achieve

✓ Facilitates development of reactive behaviors

Business As Usual

- For the past 10+ years, most JPL flight software has followed the 'monolithic image' paradigm
  - All modules linked together into a single executable image

✓ Makes efficient design easier to achieve

✓ Facilitates development of reactive behaviors

- Makes testing and analysis difficult

- Priority scheduling and lack of memory protection can have unintended and (potentially) deleterious effects

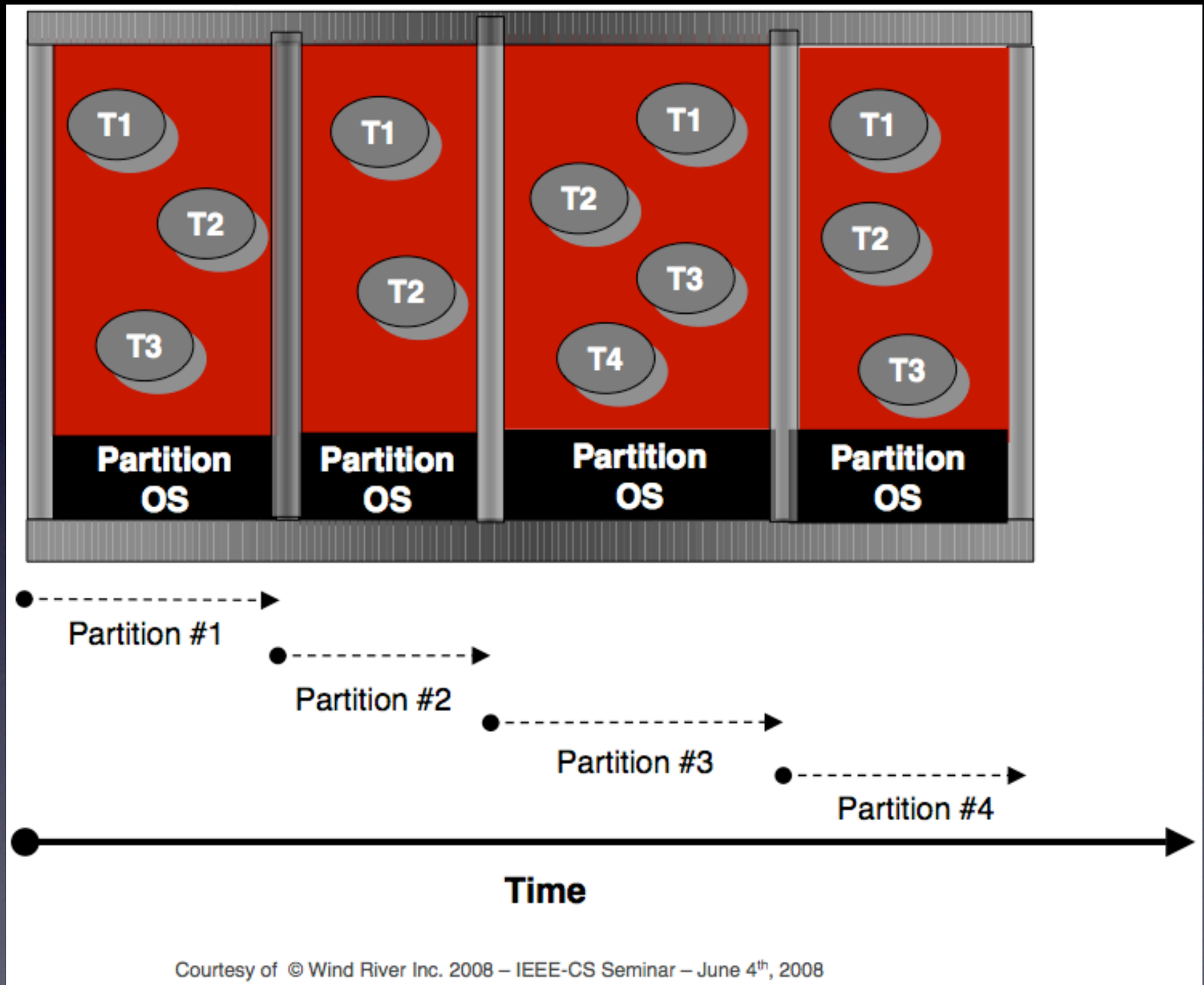
Partitioned RTOS Basics

- Recent interest in Partitioned RTOS Platforms provides a strong starting point for a new approach
    - Space Partitioning: Applications runs in separate 'brick-wall' memory partitions, and communicate using statically-defined ports and channels
    - Time Partitioning: Each partition is allocated processor time according to a fixed schedule; each partition application runs according to its own schedule.

Partitioned RTOS Basics

- The 'partition' concept establishes a computational unit suitable for analysis of memory usage, scheduling, fault containment, and testing
  - **Fault Containment:** An errant process in one partition cannot steal processor cycles or memory from another partition
  - **Mixed-Criticality Software:** Can mix safety-critical and lower-criticality software on the same processing unit
  - **Testing Benefits:** Partitioning imposes structure that can simplify testing and debugging procedures

Time and Space Partitioning



Courtesy of © Wind River Inc. 2008 – IEEE-CS Seminar – June 4th, 2008

The Pitch

- While partitioned RTOS platforms have been widely applied in the civil aviation sector, their application to space exploration is a relatively new endeavor
  - Several Constellation elements baselined on ARINC 653 platforms
- ESA interested in developing a partitioned flight software platform (RTEMS-based?)
- Academia investigating use of partitioned OS platforms for IVHM solutions (Karsai/Vanderbilt ISIS Lab)

The Pitch

- However, most documented existing applications of partitioned RTOS platforms indicate that the partitioning aspect is being used primarily to facilitate integration from multiple sources
- We propose a **partitioned flight software architecture for safety-critical applications** which best leverages the benefits of a partitioned RTOS platform, and addresses the concerns and difficulties of working with such a platform

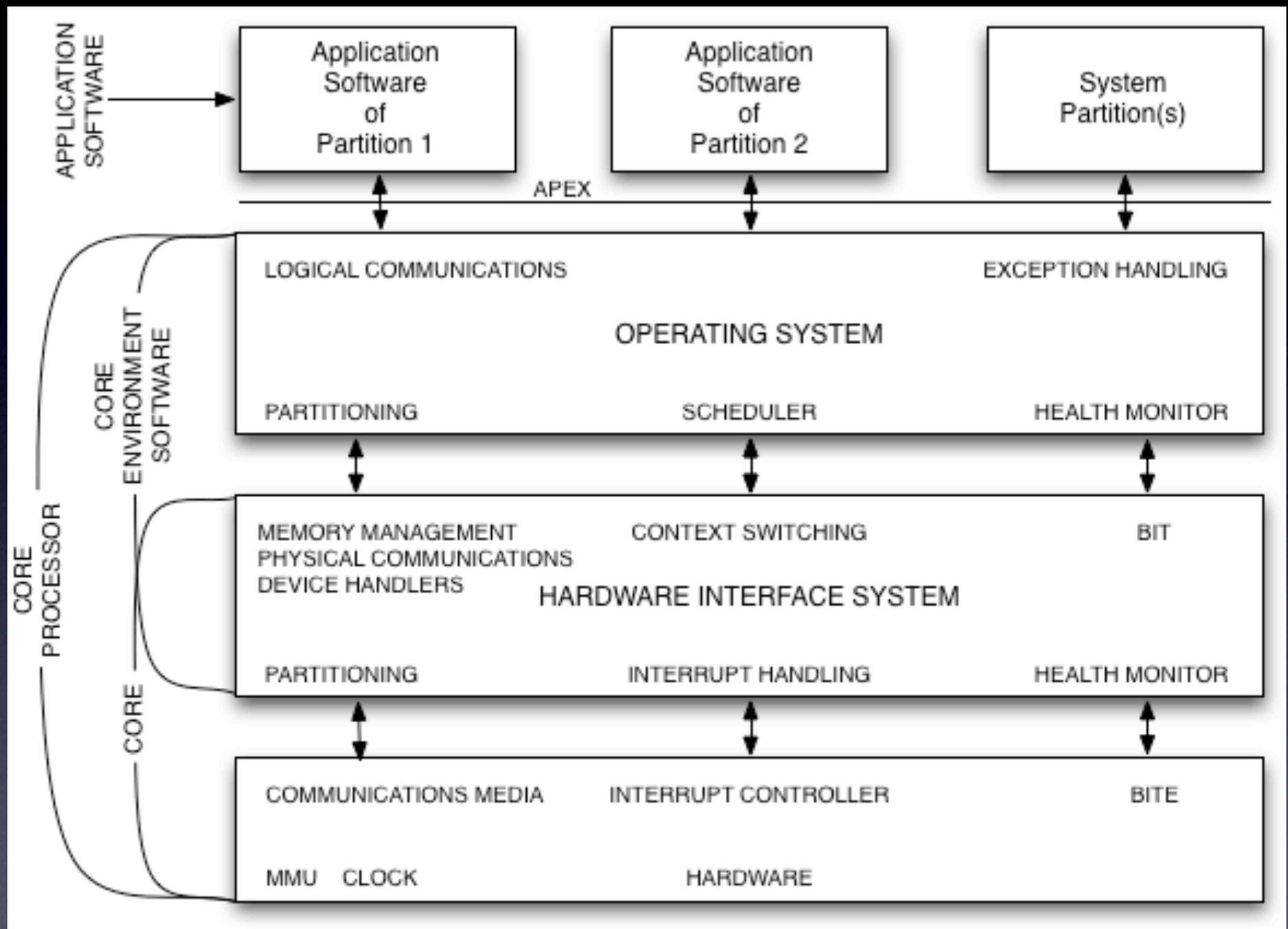
Background:

ARINC 653

ARINC 653 Standard

- ARINC 653 is a language-independent standard published by Aeronautical Radio, Inc (ARINC) that defines an operational environment for Integrated Modular Avionics (IMA)
- The standard additionally specifies an Application Executive (ApEx) which serves as the interface between the platform OS and the application software
- Defines services for
  - process and partition scheduling
  - inter- and intra-partition communication
  - status reporting

Canonical ARINC 653 System Architecture



BIT
- Built-In Test

BITE
- Built-In Test Executive

Figure reprinted from ARINC 653P1-2 Specification Document.

ARINC 653 Software View

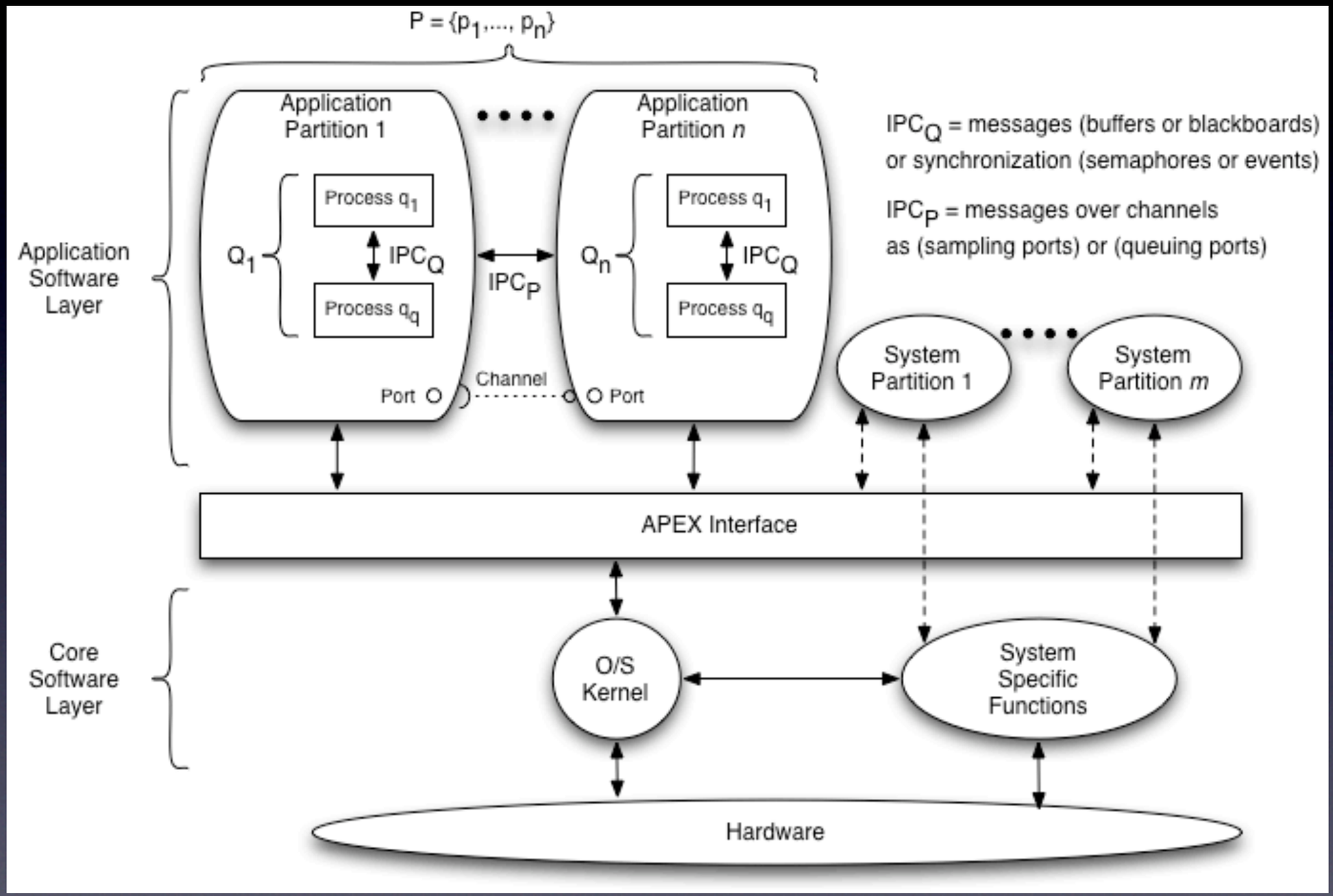


Figure reprinted from ARINC 653P1-2 Specification Document.

- Application Partitions contain application code and is limited to use of the ApEx services.

- System Partitions contain application code that requires services beyond those provided by the ApEx -- i.e. hardware communication, custom fault management schemes -- but that still benefit from time and space partitioning.

ARINC 653 System Configuration Model

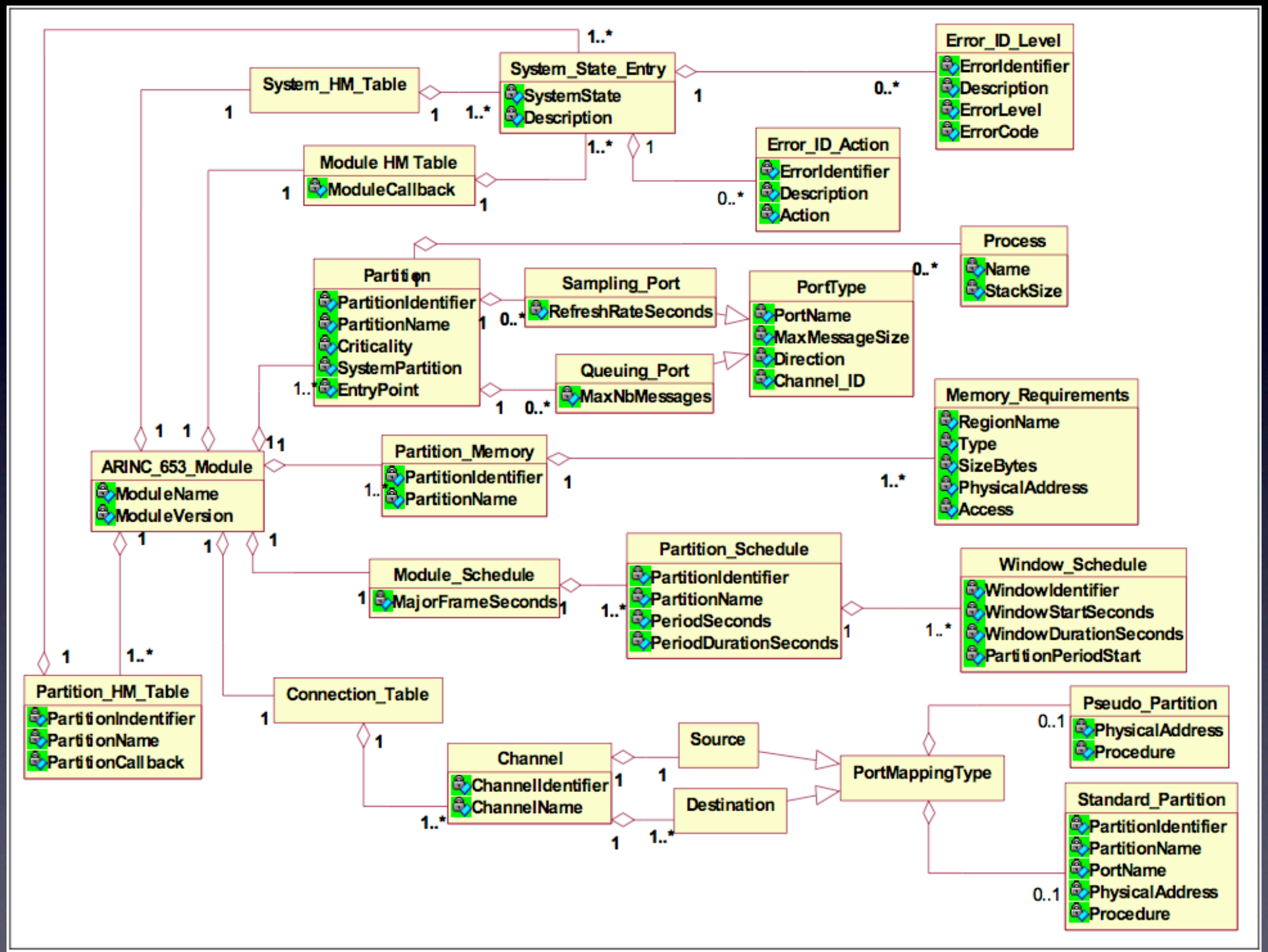


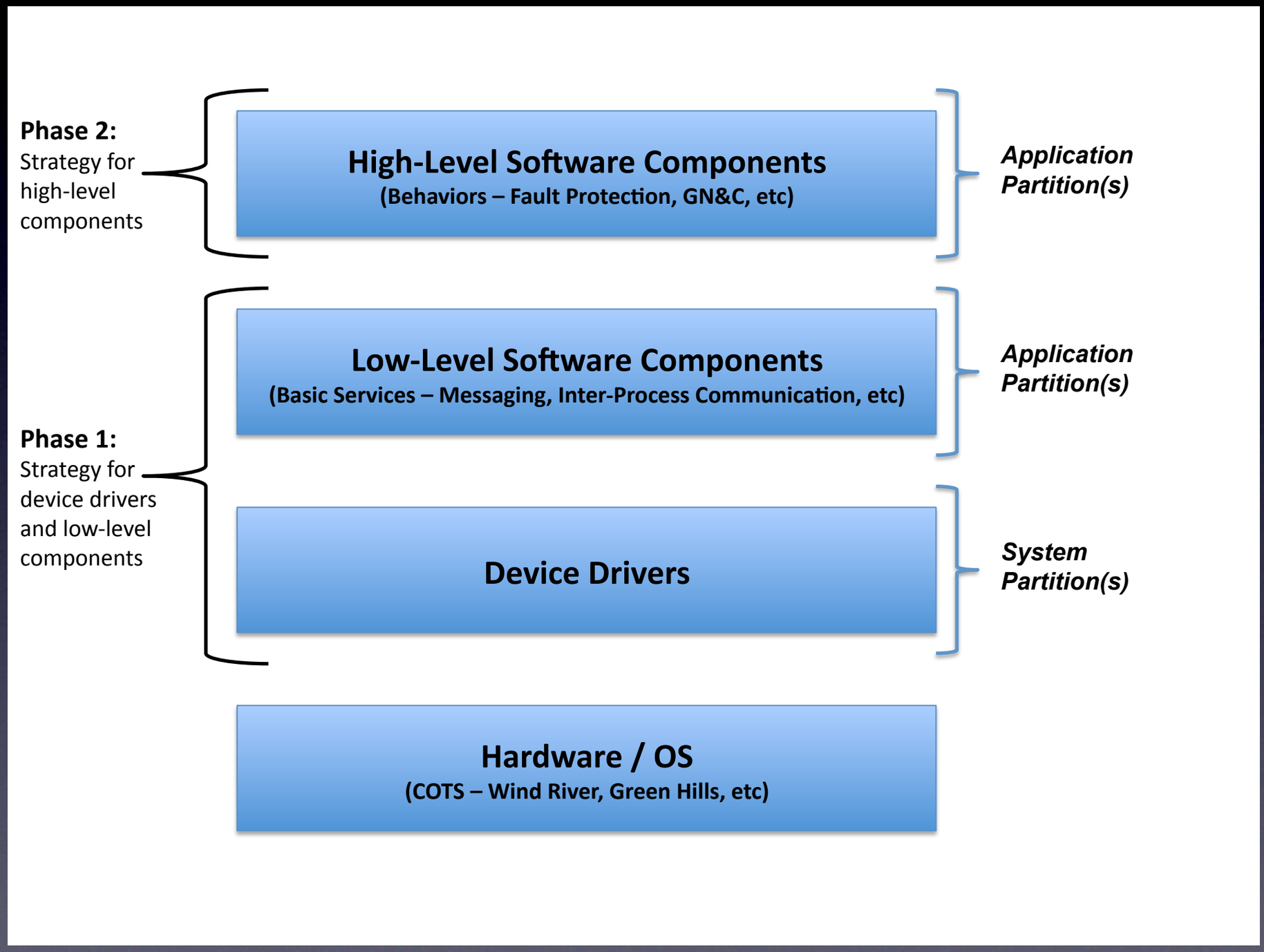
Figure reprinted from ARINC 653P1-2 Specification Document.

ARINC 653 Health Monitor

- ARINC 653 also defines a Health Monitoring (HM) capability
  - Table-driven mapping of signal → response action
  - Configurable at multiple levels (module, partition, process)
- Can be extended with user-defined signals and monitoring/response logic

A New Approach: Partitioned Flight Software Systems

Task Overview



A New Approach...

- **The Task**: Develop a software architecture for safety-critical applications, designed to take full advantage of a partitioned RTOS platform
    - Intelligently allocate FSW functionality across partitions
    - Ensure timeliness requirements are met
    - Increase software fault containment
- Minimize the use of interrupts to the extent possible
    - Enable more accurate timing analyses not possible in interrupt-driven systems

...is not without Challenges

- **The Challenges**: Develop a software architecture for safety-critical applications, designed to take full advantage of a partitioned RTOS platform
    - The competing objectives of 'ensuring timeliness' and 'effective partitioning' are often at odds, if not orthogonal
    - Meeting the throughput requirements of software modules which rely on high frequency data from underlying hardware *without interrupts* will demand strict attention to detail in system configuration
    - Efficient estimation of memory use and allocation of memory to partitions is essential, as unused memory in one partition is not available to other partitions

Three-Pronged Approach

- Three main types of task deliverables have been scoped, addressing the salient features of the work
  - **Architecture**: For each phase, produce an architecture description which documents the essential features, guiding principles, guarantees, and quality attributes of the architecture.
    - Provide a qualitative assessment of the architecture against the identified quality attributes
    - Provide a comparative assessment against existing FSW architectures
  - **Application**: For each phase, produce a realistic demonstration which will serve as a detailed example application of the architecture.
  - **Adaptation**: For each phase, produce a design handbook which provides guidelines for practitioners wishing to apply the architecture to a new system.

Issues, Challenges, and Other Fun Things

Expected Challenges

- **Technical**
  - Timeliness
  - Fault containment
  - Communication with underlying hardware
  - Efficient use of memory
- **Conceptual / Programmatic**
  - Suggesting such a drastic shift in the prevailing software development paradigm is likely to generate much debate/discussion/resistance

Wrap-Up

Presentation Goals

- Existing processes and systems have served us well
  - Many Successes! (MPF, MER, DS-1, Stardust, MRO...)
- As missions become larger and more complex, existing systems and processes start to become unwieldy
- Partitioned RTOS platforms provide some attractive features which could potentially improve the quality of our fielded software
- The move to a new platform -- and a new set of processes and methodologies -- will only give rise to new problems if done in a hurried and haphazard manner
- SPSA task aims to establish a set of principles and guidelines to allow future missions to make an informed decision regarding the choice of FSW architecture and platform

Questions?

- Expect a publication of Phase 1 results in Q3, 2010
- Contact

Greg Horvath

818/393.6234

ghorvath@jpl.nasa.gov