Georgia Tech Research Institute

Problem. Solved.

An Application in Technology Insertion for a Legacy System

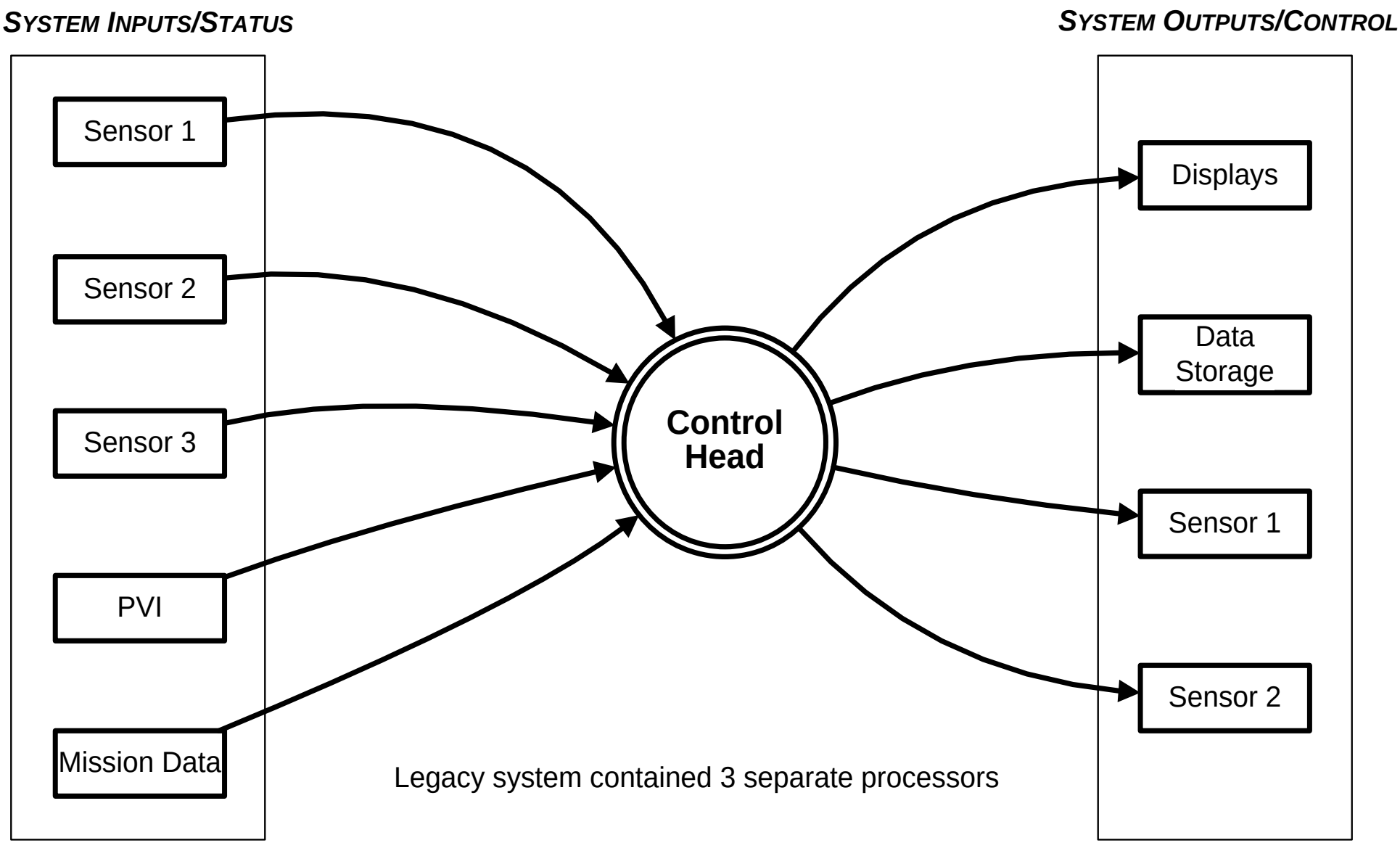
Nathan Adams - nathan.adams@gtri.gatech.edu
Jimmy Simmons - jimmy.simmons@gtri.gatech.edu

NDIA 14th Annual Systems Engineering Conference

Topics

- Legacy System Overview
- Legacy System Limitations
- Software Port
- Adding New Interface
- Support Tools

Legacy System Overview



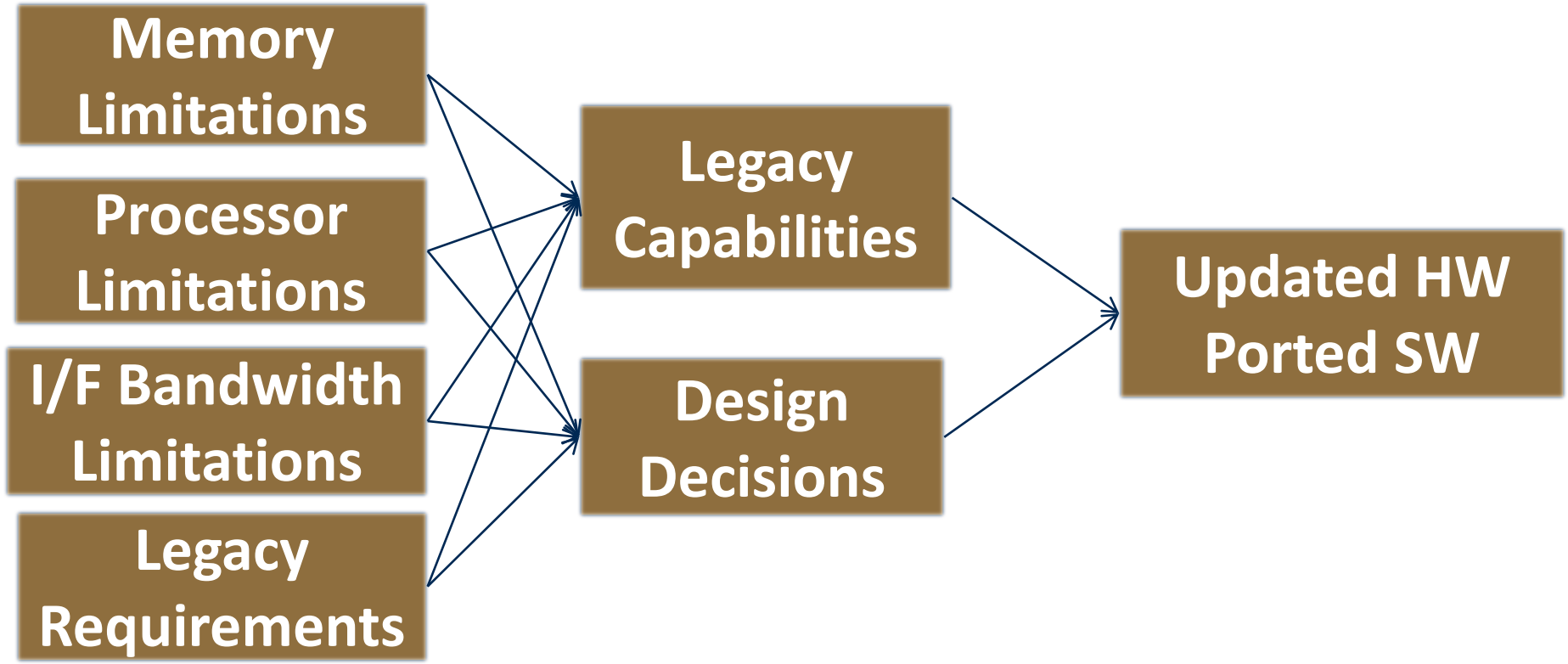
Legacy system contained 3 separate processors

Legacy System Limitations

- Single software build could not longer support multiple aircraft creating substantial sustainment cost increase
  - Processor memory/throughput could not sustain single OFP, resulted in split development paths
- Reached Processor limitations
- Diminishing Manufacturing Sources (DMS)
- External I/O bandwidth restrictions
- Engineering limitations and cost led to the desire to upgrade the legacy system

Legacy System Limitations

Design Constraints



Legacy System & Design Constraints

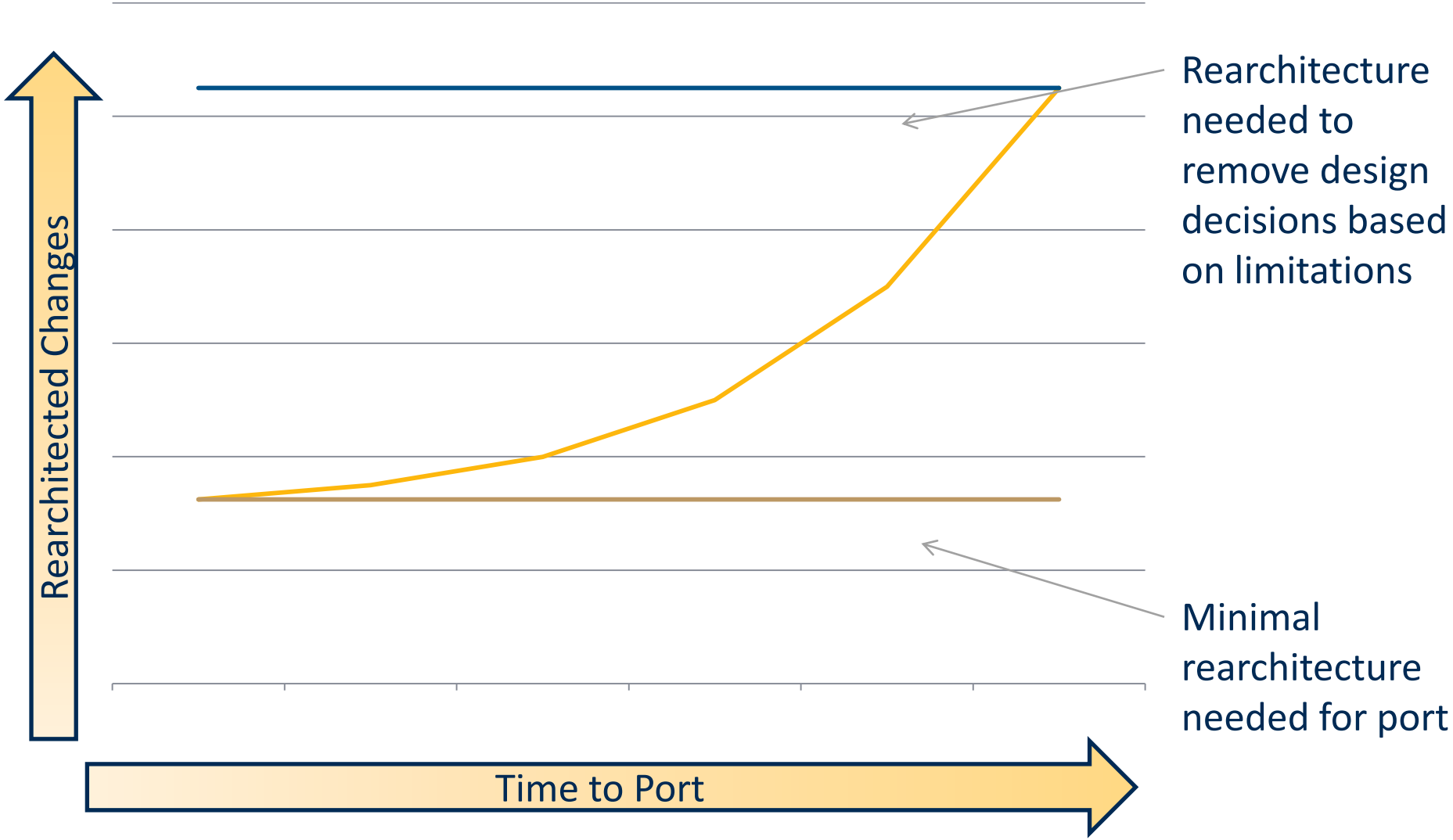
- Hardware updates included:
  - Replaced (3) legacy Processors with single more powerful Processor
  - Additional Memory
  - FPGA for Data I/O
  - Ethernet Interface for Growth
- New Processor decisions limited by legacy heat and power constraints
- Most software design decisions focused on minimal effort to port the legacy functionality to new hardware due to cost and schedule constraints

Software Port

- With schedule constraints, temptation to focus on fastest solution with minimal effort
- Care must be taken to evaluate potential future requirements and how to incorporate that into design
- For this system, forward looking design decisions included:
  - Detecting the required version at runtime, this allows for fewer development paths and lower sustainment cost
  - Massaging data from the external interfaces to appropriate software modules
  - Passing data through consistent internal interfaces

Software Port

Notional Chart: Rearchitected Changes versus Time to Port



Georgia Tech Research Institute

Selection of real-time operating system (RTOS)

- VxWorks 653
  - Certified space and time partitioning
  - Setup more complex than 6.x
  - Used by engineers in previous project
- VxWorks 6.x
  - User managed time partitioning
  - Setup less complex
- Green Hills Integrity
  - Fewer engineers had experience with this RTOS
- Hardware Vendor Proprietary RTOS
- VxWorks 653 was chosen based on the partitioning aspect and engineering experience from previous projects

Software Port

Georgia Tech Research Institute

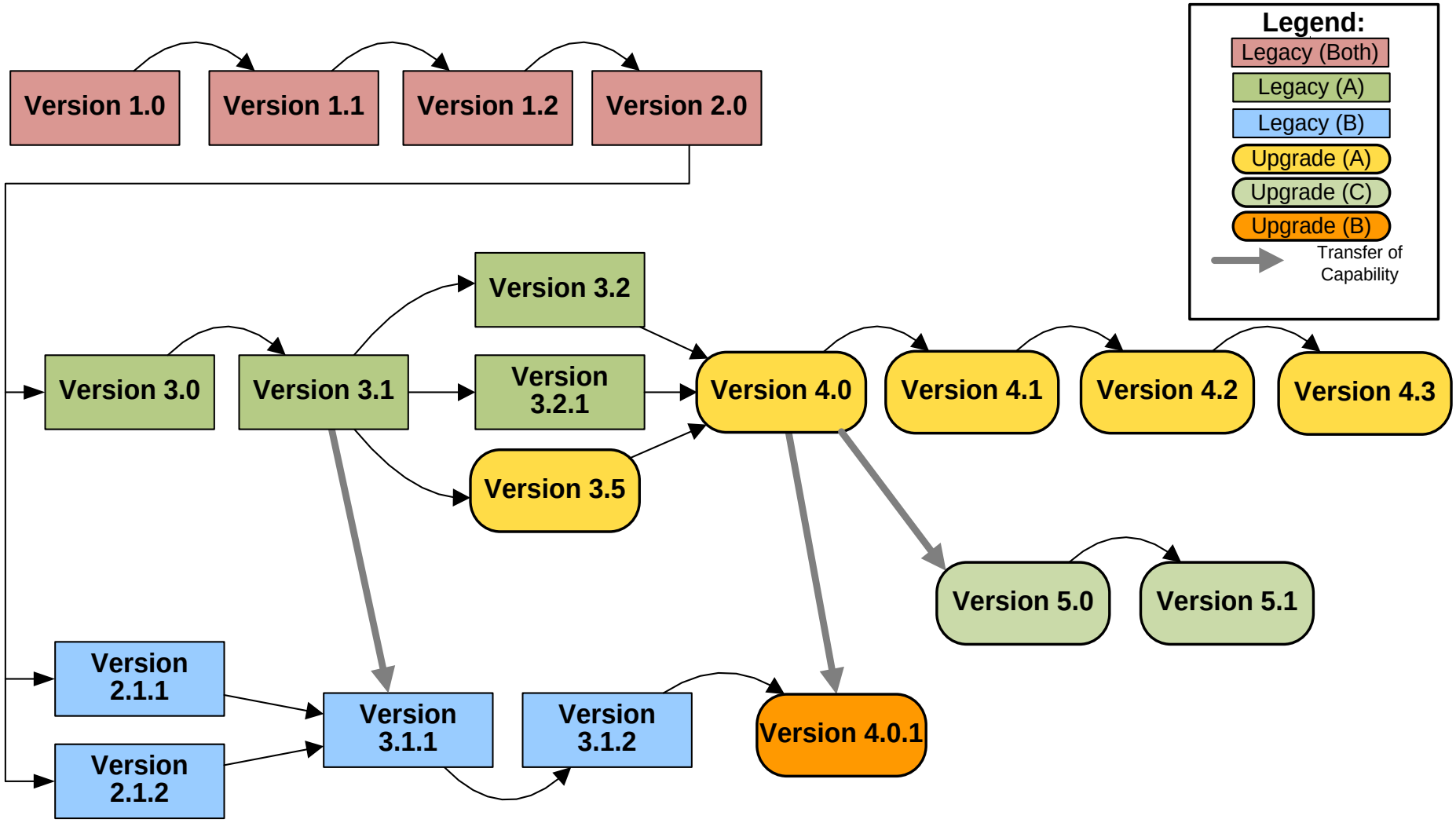
SW Port based on Tight Schedule Constraints While Designing for Future Growth:

- One legacy processor ported directly to one of the new RTOS partitions
  - Later split into (2) partitions based on functionality
- Second legacy processor was split in to three partitions based on different functionalities as well
  - Another option was to use a single partition, but for future growth splitting the functionality was preferred
- Third legacy processor was ported to a soft-core processor on an FPGA, no software changes were required!
  - Another option was to port the functionality to another partition, but scheduling constraints were the limiting factor

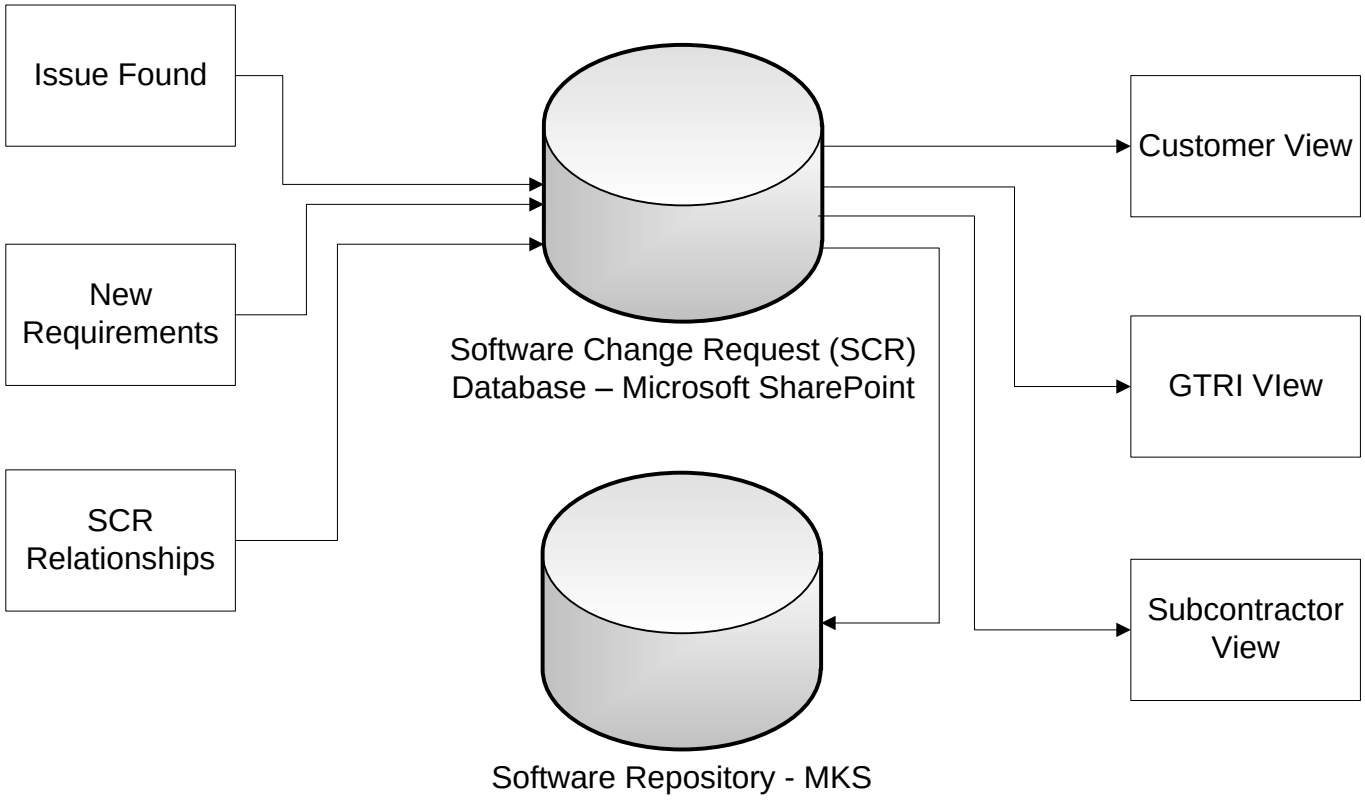
Software Port

- Decisions Going Forward
  - Parallel Support for Upgraded and Legacy System
  - Effects of Adding New Interface
  - Legacy Support Tools

Parallel Support



Parallel Support Solution

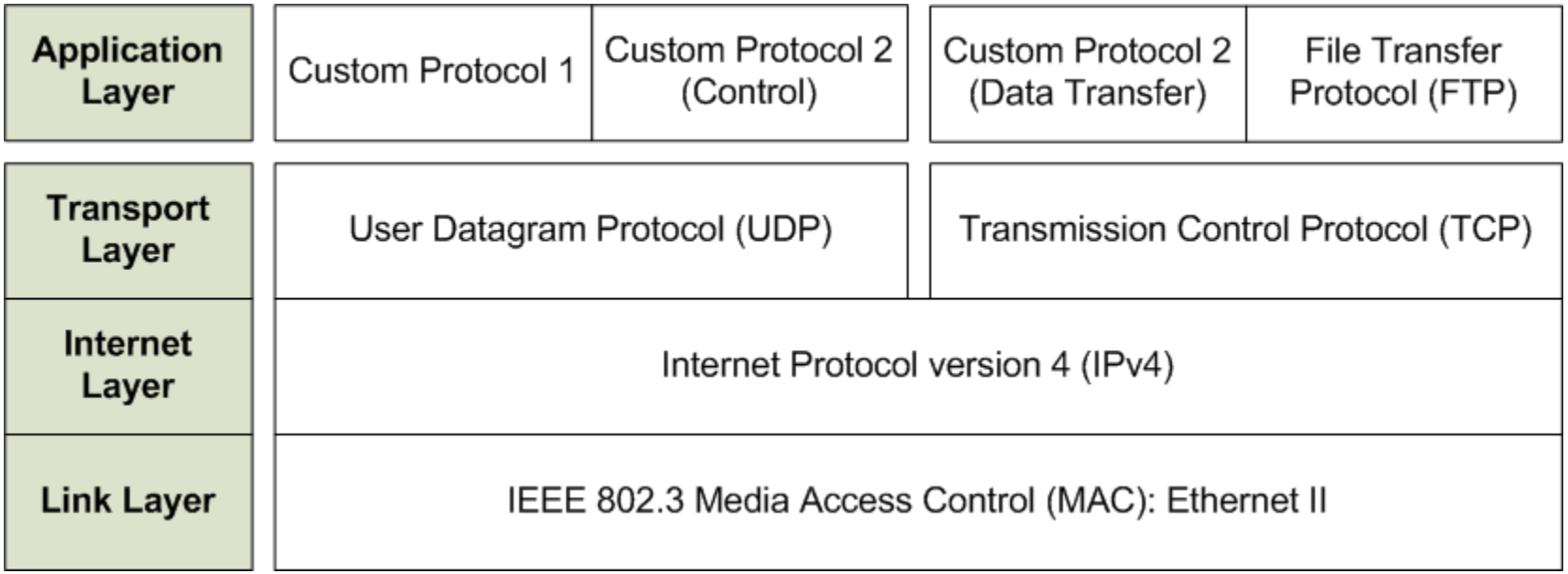


Adding New Interface

- Initial interfaces were designed for loosely coupled systems, i.e. not much information exchange
- As the systems became less federated in the associated system of systems, interface bandwidth became a limiting factor for requirements
- MIL-STD-1553B primary data bus for legacy system
  - Small message size, 64 bytes
  - Slower data rate, 1 Mbit/s
  - Requires specialized and increasingly obsolete hardware & software tools
- 10/100 Ethernet interface added
  - Max 65 Kbyte per message
  - 100 megabits per second data rate
  - Ubiquitous hardware and plethora of software tools

Adding New Interface

- TCP and UDP chosen as transport layer protocols
- Application layer protocols
  - Tradeoffs exist between proprietary protocols and older vetted protocols
  - Ultimately File Transfer Protocol (FTP) and two custom / application specific protocols were chosen



Support Tools

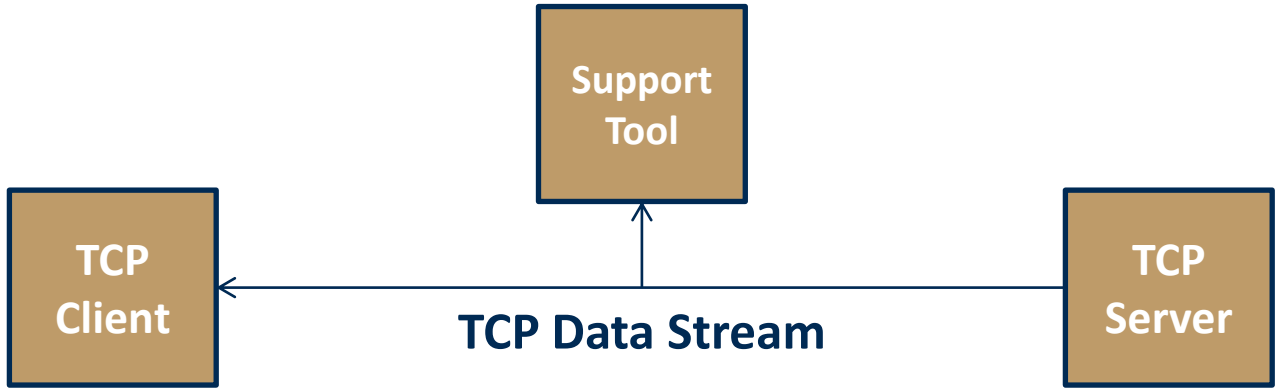
- Legacy tools based on MIL-STD-1553B messaging
  - Commercial Off the Shelf (COTS) bus analyzers
  - Custom in-house applications
    - Graphical displays for monitoring message traffic
    - Simulation application for messages to Control Head
    - Support for playback of recorded flight-test data files
- Design decision had to be made to add new interface support to old tool or design a new tool
  - Using old tool meant both simulation and analysis with a single tool, but subject to design decisions made for legacy interface
  - Creating new tool means multiple tools have to be supported, but allows for fresh start for design

Support Tools

- Updated and new tools for Ethernet based messaging
  - Opensource TCP/IP sniffing application
    - Wireshark: http://www.wireshark.org/
  - Custom in-house applications
    - Desire to see MIL-STD-1553B and Ethernet messages co-mingled for system level analysis
    - Next generation monitoring application that supports arbitrary number of messaging protocols
    - Update simulation application to support Ethernet messaging
    - Support for variety of file formats
      - IRIG 106 Chapter 10 (MIL-STD-1553B and Ethernet)
      - Firefly (MIL-STD-1553B only)
      - PCAP (Ethernet only)

Support Tools

- **Challenges with monitoring Link Layer (Ethernet) traffic**
  - **Must first decode Link, IP, and Transport Layer in order to decode Application Layer data**
  - **Must handle reassembly of fragmented packets at IP Layer**
  - **Must handle segmentation of TCP streams**
    - **No inherent way for 3rd party observer (e.g. support tool) to know where you are in the data stream!**
      - **Unless you start recording before the TCP connection is *made (not likely or guaranteed in real world)***



Support Tools

- Decision to re-work TCP-based application layer protocol to better accommodate test and evaluation tools
  - Lesson Learned: Consider test & evaluation up-front during interface design
- Legacy simulation application had many MIL-STD-1553 specific assumption embedded in design and implementation
  - Re-architected to better support multiple messaging protocols
  - Lesson Learned:
    - Software reuse is not free
    - Cost & schedule trade for new capabilities

Summary

- Legacy system sustainment is a challenge
- Technology insertion alleviated the system exhaustion but came with various hurdles
- Hardware replacement and SW port is only part of the solution, support tools also must be considered
- By the time new system fields, not unusual for system exhaustion to be at the forefront again!

Questions?

Georgia Tech Research Institute
Problem. Solved.