Deos

Real-Time Operating System for Mission/Safety-Critical Software Systems

– Technical Overview –

DDC-I

Safety Critical Software Solutions for Mission Critical Systems

Agenda

- Introductions
- Company Background
- Deos competitive differentiators
- Technical Overview
  - Avionics Pedigree
  - Scheduling Partitioning, RMA, 653, POSIX
  - Certified Software Reuse
  - Data Distribution Service
  - Multicore and FAA CAST-32A
  - FACE, POSIX and Additional Components
  - Development Tooling

DDC-I

Leading provider of mission/safety-critical software solutions for 30 years.

- Headquarters in Phoenix, AZ
  - World-wide presence
- Primary market: Certifiable avionics software

Core Competencies

- Certifiable, safety-critical RTOS products
  - Deos (ARINC-653, RMA, POXIS or hybrid)

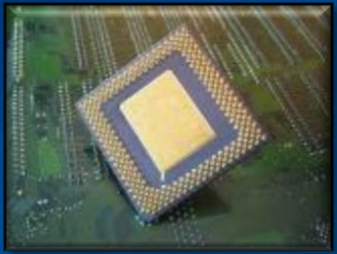
- Integrated Development Environment (IDE)
  - Development, testing & analysis tools

- DO-178/ED-12 certification expertise
  - First DO-178 DAL-A (Ada) product released in 1992
  - We perform our own certification work
  - We defend our certification artifacts during all audits
  - We do not reverse engineer certification artifacts

Corporate Milestones

- Founded 1985 – Ada compiler for x86, 860
- 1988 – DACS selected by Honeywell, RC, etc.
- 1991 – DO-178 DAL-A Ada runtime
- 1998 – Deos certified to DAL-A (by Honeywell)
- 2008 – Licensed Deos and HeartOS
- 2012 – ARINC-653 offered on Deos
- 2016 – Deos SafeMC released, ARM offered
- 2019 – FACE 3.0 conformance, Engineering doubled in size to keep up with rapid growth

Supported CPUs



- ARM A
  - A9 - Xilinx Zynq 7000, NXP i.MX6, Altera Cyclone V
  - A53 - Xilinx Ultrascale+, NXP i.MX8, S32V234 , QorIQ Layerscape LS1043A, LS1048A
  - A72, A15 – Contact DDC-I
- Intel (and compatibles)
  - x86 (Pentium, Atom, Celeron, Core (duo, i7, …), AMD GSeries)
- PowerPC
  - e6500/e5500/e500/e500mc (QorIQ), e200 micro-controllers (51xx, 52xx, 56xx, 57xx), G2/603e (82xx), G3 (7xx), e300 (83xx), G4/e600 (74xx, 86xx), 405x, 440x, 465x
- MIPS

… support for more processor families & processor cores than any other partitioned certifiable COTS RTOS…

The Deos RTOS

Differentiators

Avionics Pedigree

Scheduling Partitioning, RMA & 653

Certified Software Reuse

Data Distribution Service

Multicore and FAA CAST-32A

Development Tooling

Top Level Differentiators

- One RTOS product line - no product fragmentation for 653, multicore, or processor type (no CPU dependencies)
- Designed for Reuse – DAL A linking loader built in, RTOS built of independent relocatable libraries.
- Technical - Almost 20 patents on Deos
  - Originally : Slack scheduling, RMA scheduling, etc.
  - Recently : Cache partitioning, Safe Scheduling, etc.
    - *Controlling interference patterns on multicore systems*
    - *Key multicore design criteria – cannot be "bolted on as an afterthought"*
- Very strong support team and support models
- Company focus on real-time safety critical

Differentiators - Portability

- Consistent certifiable code base line across processors
  - PowerPC to ARM to x86 is a recompile (ask our partners)
  - Only one Deos baseline (not 6 or 7 versions which is costly for customers)
  - Deos designed modularly 20 years ago

  Customer Benefit :

  1. ***Modularity and reuse yields significant cost savings***
  2. ***Deos is easily offered on new processors***
  3. ***Tooling is consistent, 3rd party products migrate***
- Consistent set of artifacts, processes and test suite
  - DDC-I has always done our artifacts in house (no Verocel or 3rd party)
  - Modularity enables us to leverage prior certifications
  - Rarely break backward compatibility

Differentiators - Technical

- Flexible scheduling
  - Can use RMA scheduling to compliment ARINC 653 scheduling
  - Slack scheduling maximizes CPU performance
  - Safe Scheduling for multicore (patented) – can use all cores for DAL-A
- Performance
  - Extremely quick boot up times
  - Often 30% faster than competition on context switch times
- Toolset
  - Based on Eclipse and GNU (but can run other)
  - DO-330 qualified where applicable
  - Timing & resources tracked in Deos kernel and presented via tooling
  - Critical time kernel gathers performance statistics
  - Included coverage tool (ABC) provides MC/DC at binary level

Differentiators – Technical (cont)

- DDS - Deos has a DAL-A DDS (IOI) built in
  - Provides data abstraction – needed for binary reuse.
  - Highly efficient – customers track thousands of data points
  - User Space library, configurable via XML package
- Optional components:
  - CFFS (certifiable fast file system) - Optional ARINC 653 P2 APIS
  - AFDX libraries (ARINC 664)
  - TTE libraries (time triggered Ethernet)
  - POSIX libraries
  - ARINC 615A Target Data Loader libraries
- Watermarking
  - Deos kernel tracks and can report performance statistics
  - Can leverage CPU performance monitors, useful in multicore

The Deos RTOS

Overview

✦ **Avionics Pedigree** ✦

Scheduling Partitioning, RMA & 653

Certified Software Reuse

Data Distribution Service

Multicore and FAA CAST-32A

Security Building Blocks

Development Tooling

Background

- Designed *from the ground up* for:
  - Mission-/safety-critical applications & DAL A
  - Real-time performance
  - Integrated Modular Avionics (IMA)
- 1998: Initial certification baseline (DAL A)
- 2017: Latest certification baseline (DAL A)
- 2020: Next planned certification baseline (DAL A)

… a mature, trusted product, widely deployed for over 15 years.

Pedigree

Deos Deployments

Aircraft

- Agusta AB-139
- Airbus A320, A330, A340, A380, A400M
- Bell-Boeing V-22 Osprey
- Boeing 757, 777, 787, F-18,
- Bombardier CSeries, Global Express
- Cessna Citation V, Sovereign
- Comac ARJ21, C919
- Dassault F7X, F900, F2000
- Embraer ERJ-170, ERJ-175, ERJ-190, ERJ-195
- Gulfstream GIV-X, GV, G150, G200, G350, G450, G500, G550, G650
- Hafei Y-12
- Hawker Horizon, 450
- Hindustan Aeronautics Limited, LUH, ALH
- Lockheed C-5, C-130J, C-141
- Pilatus PC-12NG, PC-12-NGX
- Spectrum S-40
- Viking Twin Otter

Avionics Functions

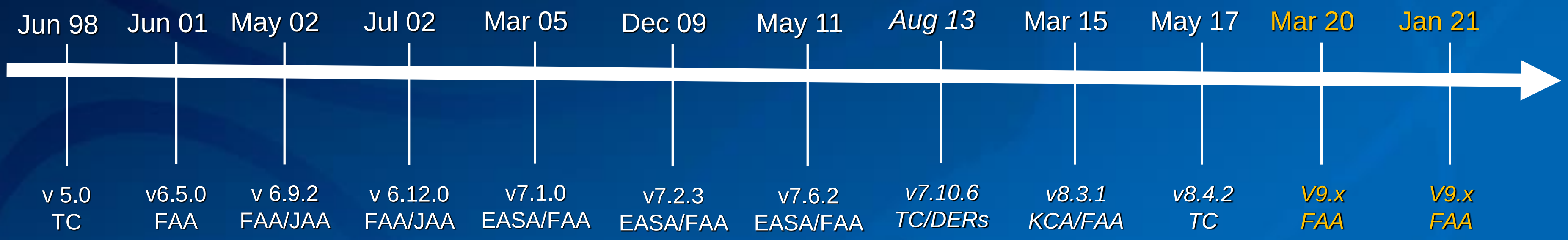
- Air Data Computer
- Air Data Inertial Reference Unit
- Cockpit Video
- Communications & Radios
- Data Recorders
- De-Icing
- Digital Engine Controller
- Displays
- Electronic Flight Bag
- Enhanced Ground Proximity Warning
- Flight Controls
- Flight Instrumentation
- Flight Management
- Health Management
- Maintenance
- Power Distribution
- Traffic Collision Avoidance System
- Weather Radar

… best-in-class service history on a broad range of aircraft & functions.

Certification Baselines (DAL A)

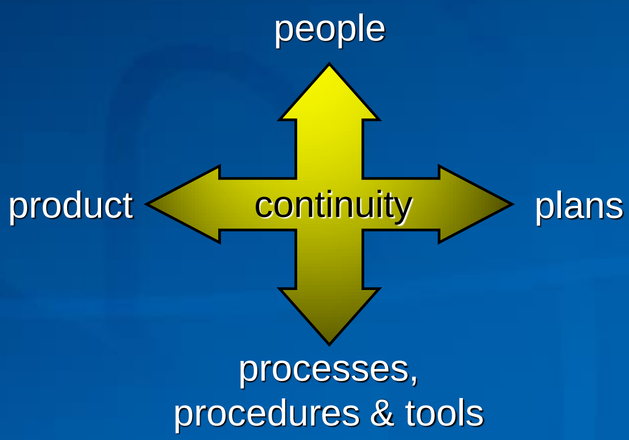
Experience
Experience
observing, encour
practical knowledge, skill
resulted in understanding
of events participated in

A single line of on-going DAL-A development



Jun 98	Jun 01	May 02	Jul 02	Mar 05	Dec 09	May 11	*Aug 13*	Mar 15	May 17	Mar 20	Jan 21
v 5.0 TC	v6.5.0 FAA	v 6.9.2 FAA/JAA	v 6.12.0 FAA/JAA	v7.1.0 EASA/FAA	v7.2.3 EASA/FAA	v7.6.2 EASA/FAA	*v7.10.6* *TC/DERs*	*v8.3.1* *KCA/FAA*	*v8.4.2* *TC*	*V9.x* *FAA*	*V9.x* *FAA*

Planned



Some auditors we've worked with

- **Jozef van Baal – JAA**
- **Connie Beane – FAA**
- **Marc Bouvelle – EADS**
- **Jorge Castillo – FAA**
- **Mike DeWalt – FAA**
- **Ricky Jones – FAA**
- **Sylvain Le-Borgne – EADS**
- **Ian Mac Laren – EASA**
- **Pippa Moore – JAA**
- **Don-Jacques Ould-Ferhat – EADS**
- **John Philbin – FAA**
- **Gerald Pilj – FAA**
- **Leanna Rierson – FAA**
- **Cecile Saint-Marcoux – EADS**
- **Cédric Salson – EASA**
- **Will Struck – FAA**
- **Giuliana Tamburro – ENAC**
- **Dennis Wallace – FAA**
- **DERs – Agusta, Airbus, Boeing, Bombardier, Cessna, Dassault, Embraer, Goodrich, Gulfstream, Honeywell & Raytheon**

… best-in-class certification record, well-known & respected in the industry.

The Deos RTOS

Overview

Avionics Pedigree

Scheduling Partitioning, RMA & 653

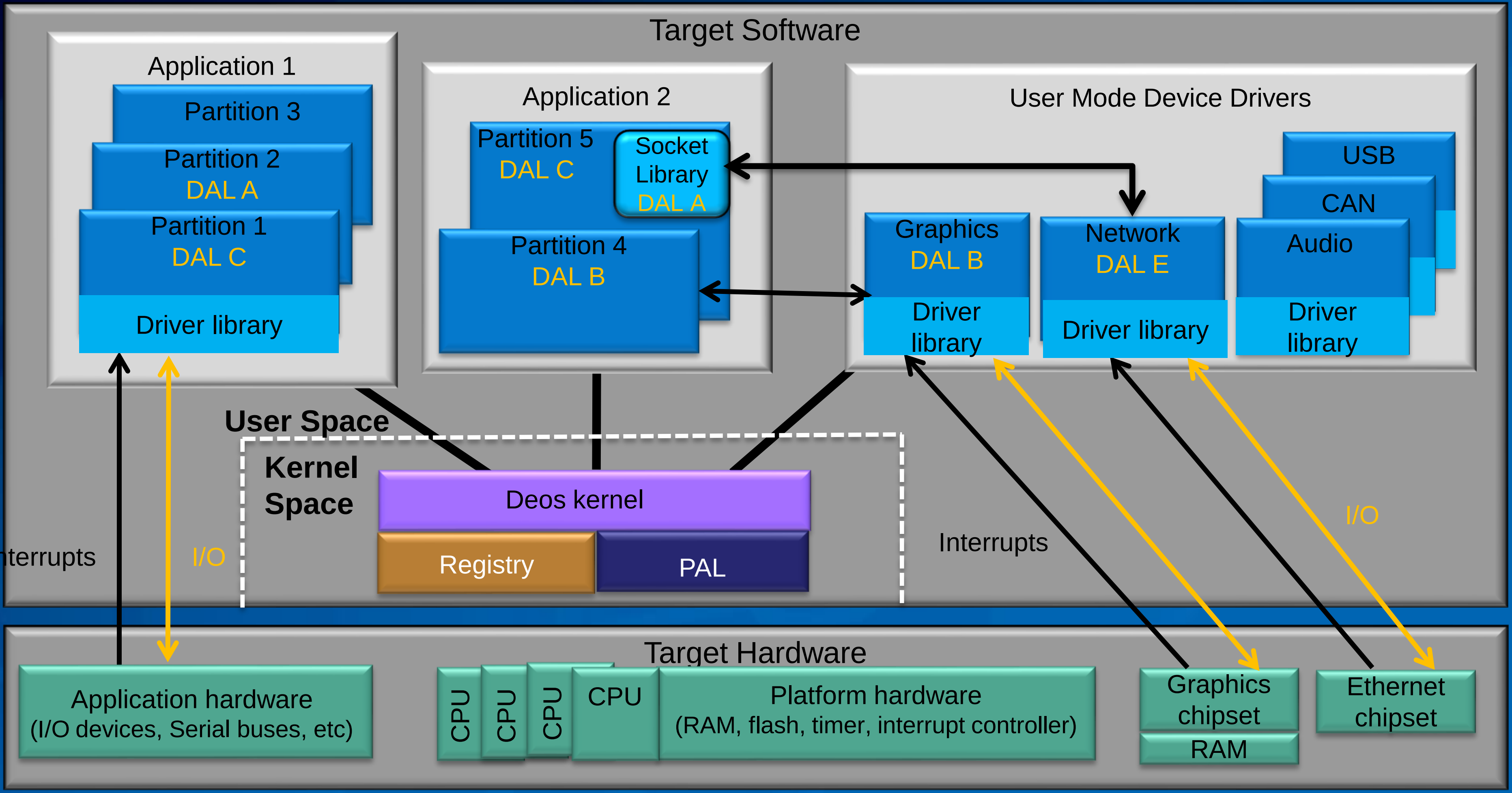
Certified Software Reuse

Data Distribution Service

Multicore and FAA CAST-32A

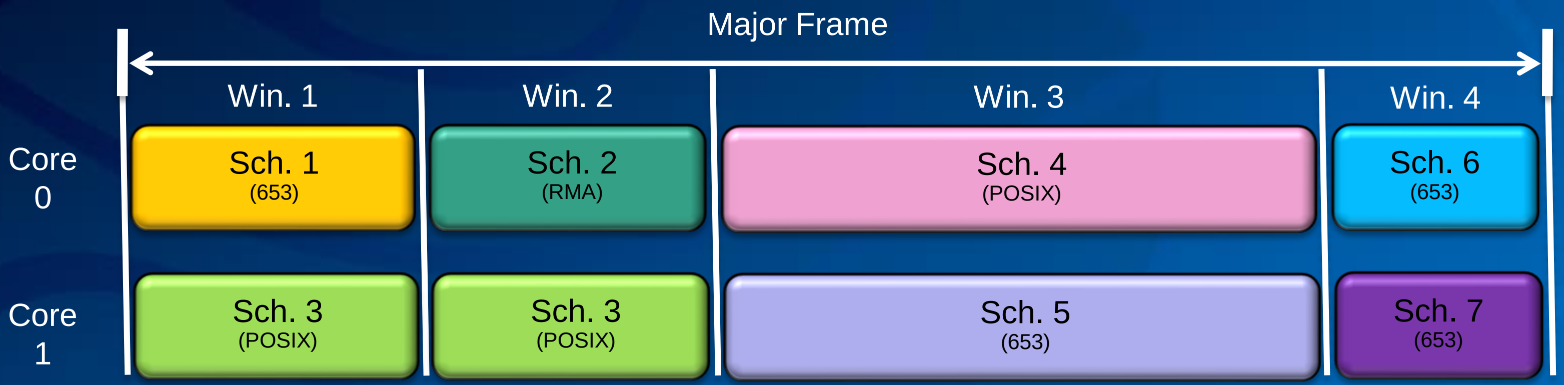
Development Tooling

Deos High-Level Architecture



… loosely-coupled, modular application software partitions.

Deos Safe Scheduling for Multicore



Major Frame

	Win. 1	Win. 2	Win. 3	Win. 4
Core 0	Sch. 1 (653)	Sch. 2 (RMA)	Sch. 4 (POSIX)	Sch. 6 (653)
Core 1	Sch. 3 (POSIX)	Sch. 3 (POSIX)	Sch. 5 (653)	Sch. 7 (653)

- Bounds, controls & minimizes cross-core contention
  - Major frame partitioned into “windows”
  - Window boundaries align across cores
- Multiple scheduler/API types available
- Allows for a mix of safety apps, or safety & non-safety apps

… optimizes application ACET/WCET behaviors and bounds WCET behavior.

Deos: Four Scheduling Options

1. Rate Monotonic – Highest throughput & fidelity
2. ARINC 653 – Industry standard API
3. Hybrid (RMA + ARINC 653) – Best of both
4. POSIX – real-time priority preemptive scheduler.

   Supports Safety Base Profile (Face Technical Standard)

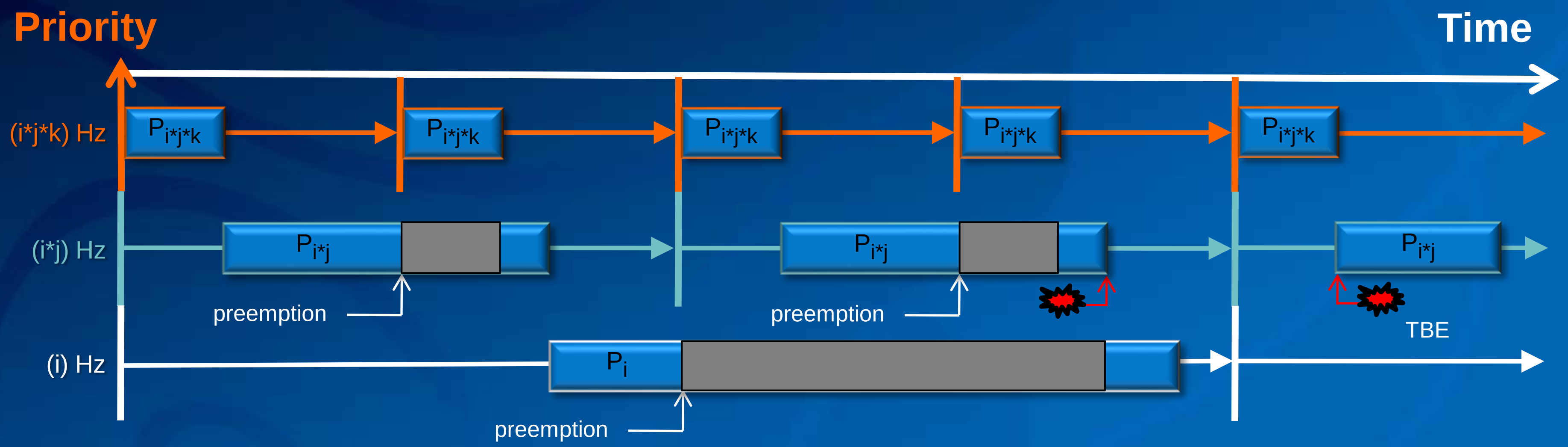
Each are highly deterministic

All but POSIX are DO-178 DAL-A certifiable

All share a large common code base, tooling, etc.

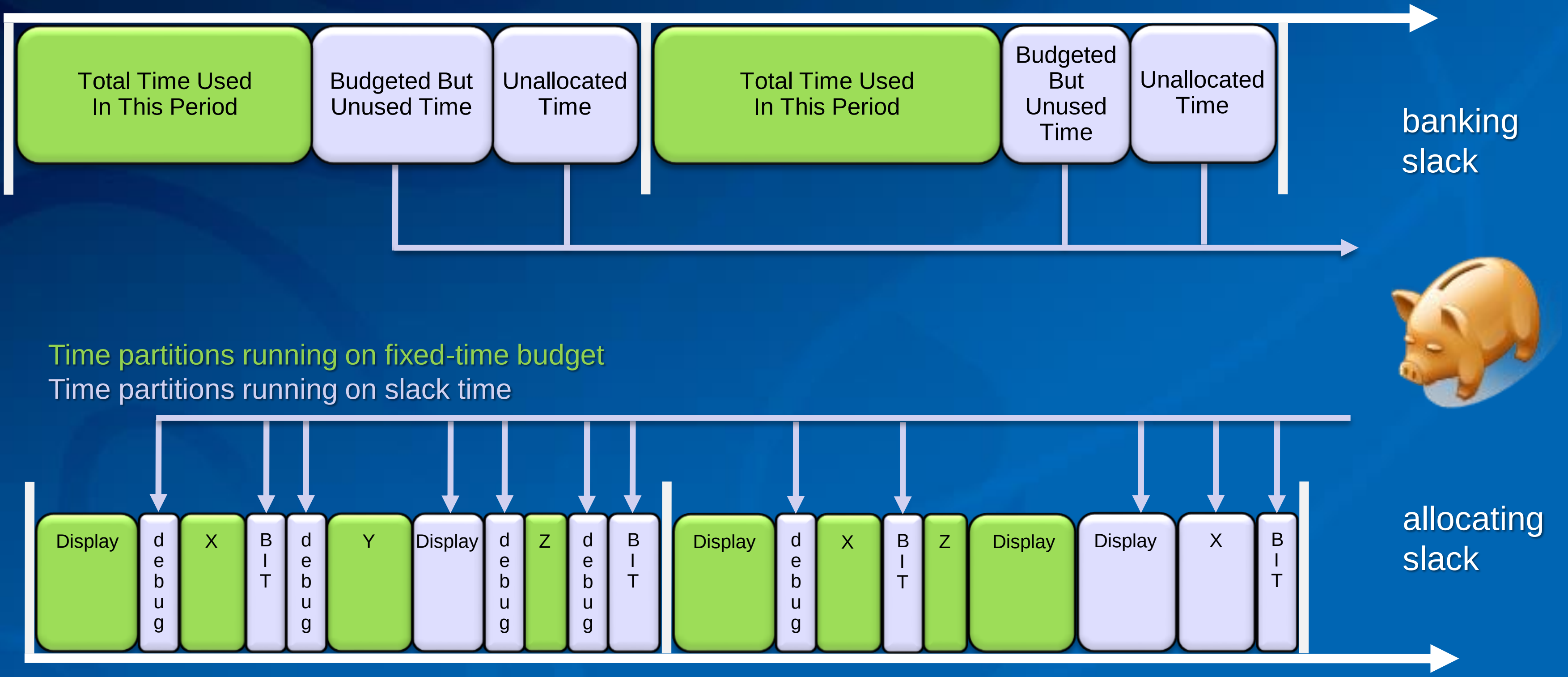
RMA Scheduling & Time Partitioning

Configurable Harmonic rates



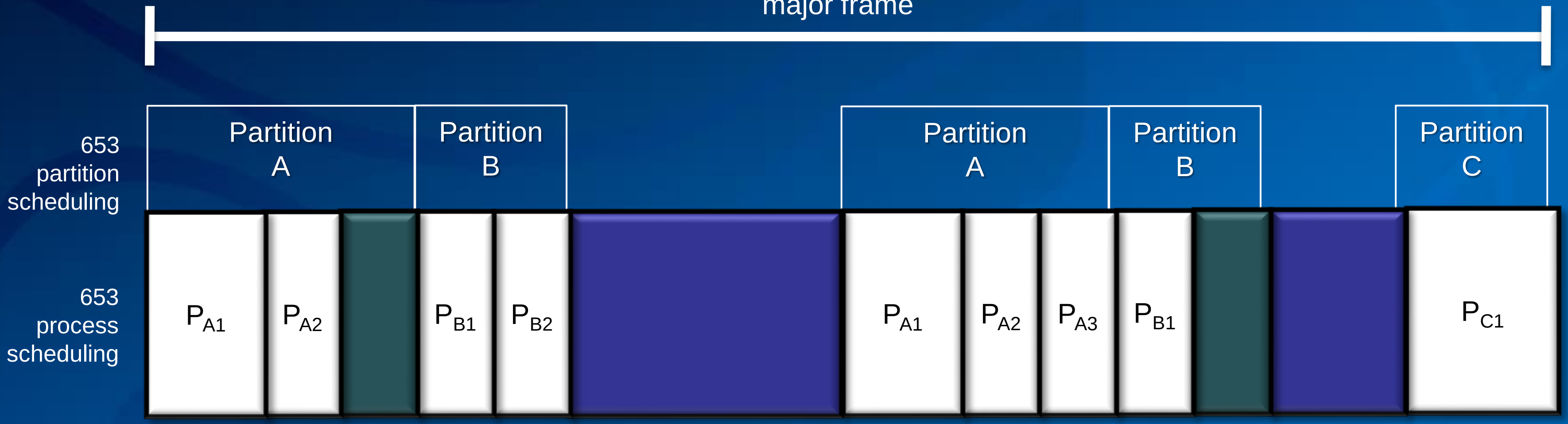
- Rate monotonic scheduling of periodic time partitions
- Threads have execution rates & time budgets
- Scheduler enforces time budgets
- Aperiodic rates & interrupts supported

RMA & Slack Scheduling



- Pure slack applications & enhanced quality of service
- Increased processor utilization

653 Scheduling & Time Partitioning



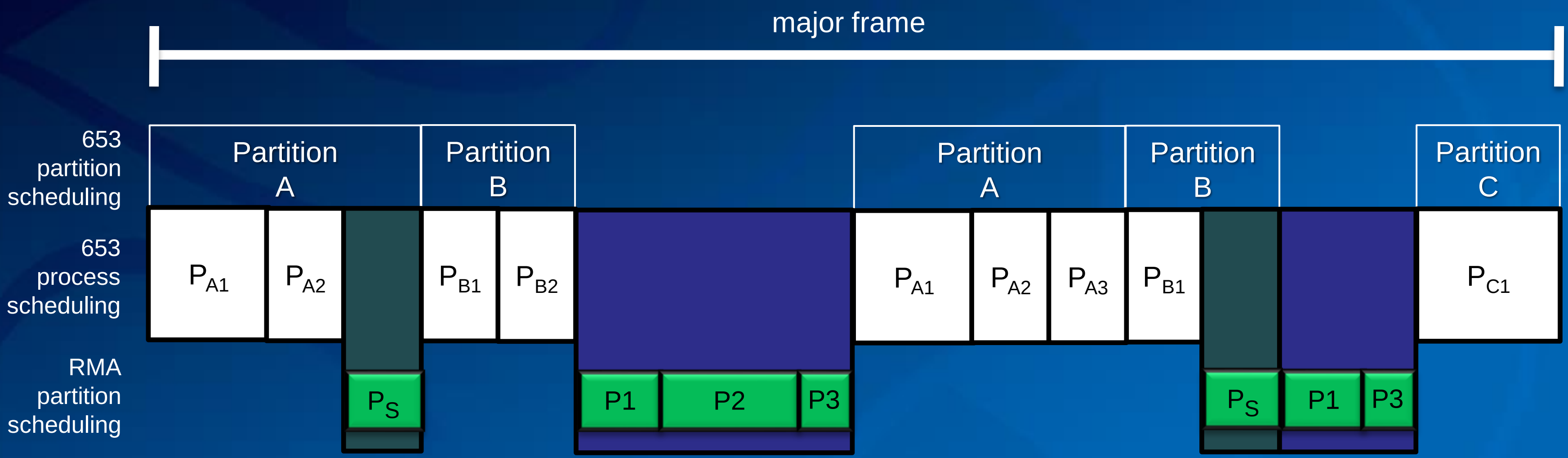
□ Scheduling of 653 partitions/processes

▨ Budgeted but unused 653 timeline

■ Unallocated 653 timeline

- Fixed-cyclic scheduling of 653 partitions
- Processes scheduled within partitions
- 653 partitions have time budgets
- Scheduler enforces time budgets
- Interrupts not supported during partition

Hybrid RMA/653 Scheduling



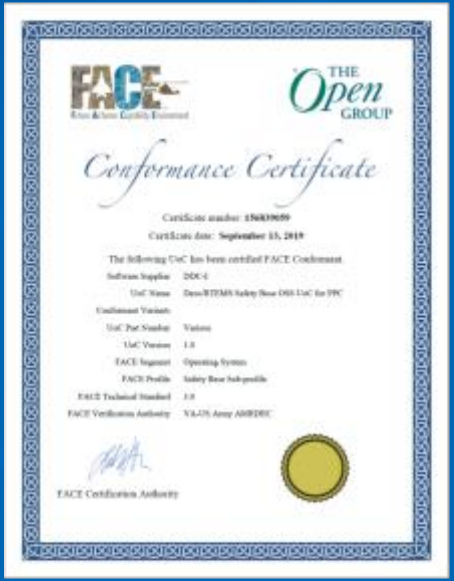
- Enables RMA (with all its features) for more optimal performance *and* ARINC-653 for 3rd party applications
- May be used with Deos 653 Slack Windows & interrupts

□ Scheduling of 653 partitions/processes
⬛ Budgeted but unused 653 timeline; becomes RMA slack
🟪 Unallocated 653 timeline
🟩 Scheduling of RMA partitions

POSIX Support via RTEMS

- RTEMS provides the POSIX interfaces and 27 years of RTOS experience in the space and military domains
- POSIX 1003.1b a.k.a. Open Group Single Unix Specification
- Within limits of single process environment
- High performance with deterministic behavior
- Low overhead and predictable

- FACE 3.0 conformance
- PPC (Deos SafeMC)
- ARM and Intel underway (Deos SafeMC)
- Safety Base conformance (Security is a subset)


FACE
Future Airborne Capability Environment
THE Open GROUP

Conformance Certificate

Certificate number: 156839059

Certificate date: September 13, 2019

The following UoC has been certified FACE Conformant

Software Supplier	DDC-I
UoC Name	Deos/RTEMS Safety Base OSS UoC for PPC
Conformant Variants	
UoC Part Number	Various
UoC Version	1.0
FACE Segment	Operating System
FACE Profile	Safety Base Sub-profile
FACE Technical Standard	3.0
FACE Verification Authority	VA-US Army AMRDEC

FACE Certification Authority

Performance

- Designed *from the ground up* to meet the most demanding real-time requirements
- Very low overhead:
  - System tick handling
  - Context switch – 2.6µs
  - Interrupt latency – 5.8µs (kernel-mode) to 11.8µs (user-mode)
  - Cold/warm-start – as low as 100ms
    - Customer Boot requirements
    - From end of Boot to first line of application code 7ms
- Scales-up gracefully

for what is thought
to be best in any rela
point of view.
Optimize
make effect
maximum eff
capacity, time, cos
what is thoug

… best-in-class performance.

Context Switch Timing Multicore

Xilinx Ultrascale A53, 4 cores.

Partition Switch Times (usec)

Partitions	Min	Max	Avg
4	3.3	4.72	3.83

Process Switch Times (usec)

Partitions	Min	Max	Avg
4	1.29	1.47	1.36

NXP PPC T1042, 4 cores.

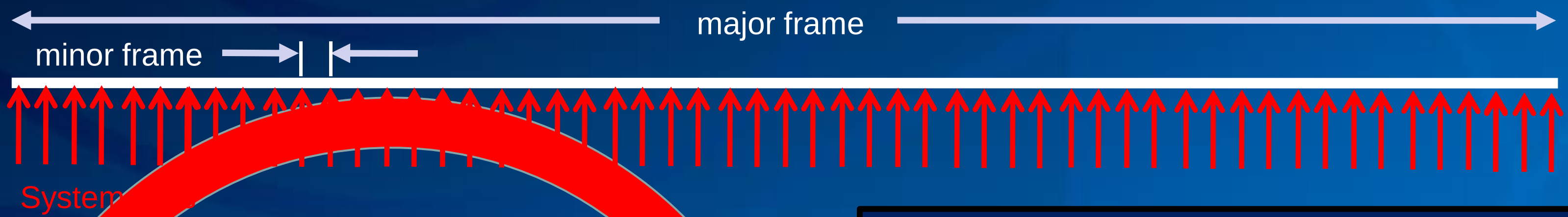
Partition Switch Times (usec)

Partitions	Min	Max	Avg
4	6.56	10.32	7.7

Process Switch Times (usec)

Partitions	Min	Max	Avg
4	1.12	1.68	1.24

Common ARINC-653 RTOS Implementation Issues



- System ticks = Minor fra[illegible]
  - WCET system tick can be se[illegible] or worse (vendor dependent), cache impact, etc.
  - Example impact: If smallest partition is 0.5ms, and system tick overhead is 50μs, 10% of the processing bandwidth is lost (e.g., a 20 ms partition will have:

    40 ticks x 50μs = 2ms system tick overhead)

Not with Deos! System Tick can be at major frame rate, slower, or faster

Key Scheduling Differentiators

- MMU (not MPU) Partioning
- Scheduler Options:
  1. ARINC 653 Part 1
  2. RMA architecture facilitates higher performance
  3. Hybrid RMA/653 capability enables best of “both worlds”
  4. POSIX
- Slack scheduling
  - Utilize “budgeted-but-unused” time, maximizing CPU utilization
  - Enable sophisticated “background” tasks
  - Enable higher quality-of-service behaviors (e.g., Ethernet transfers, display refreshes, etc.)
  - Enables tooling to run without imposing on timelines

The Deos RTOS

Overview

Avionics Pedigree

Scheduling Partitioning, RMA & 653

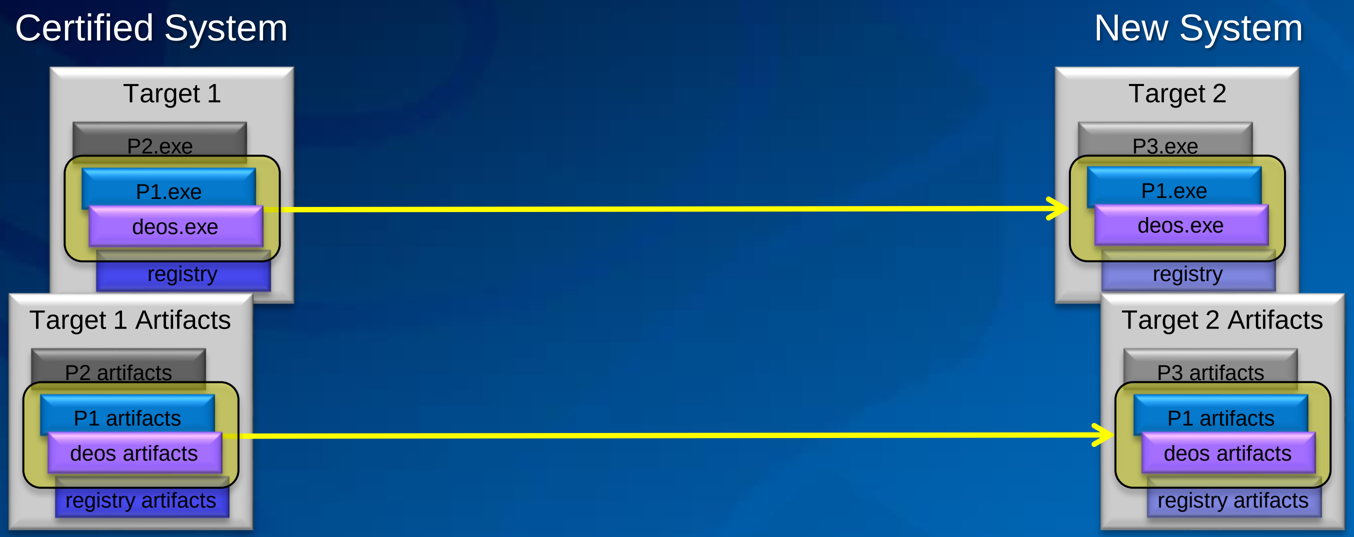
Certified Software Reuse

Data Distribution Service

Multicore and FAA CAST-32A

Development Tooling

Binary Modularity



- Enabling Deos Technology: DAL-A loader & (dynamic) linker
- Application partitions & libraries (.exe & .so)
  - Each has a DAL with a full certification package
  - Identical binaries & artifacts used program-to-program
  - Minimal rework (verf/cert activities & artifacts) & integration testing required
- Same strategy Deos & its libraries have used for 20 years

… modularity enables reuse of software & certification artifacts.

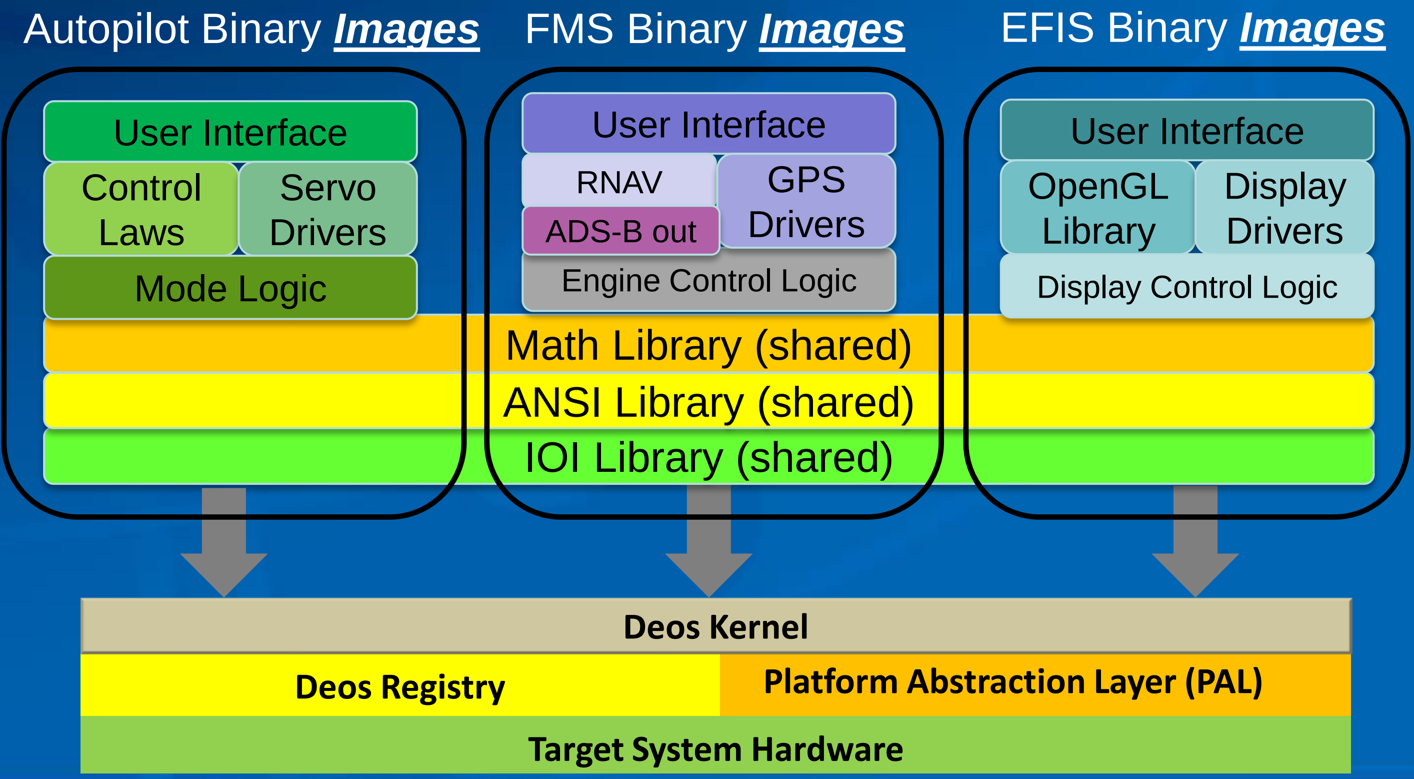
Reusable By Design

- Deos is designed to enable application software component portability, where binary (not just source code) reuse is the ultimate form of portability.
- Deos key technologies:
  - DO-178C-Level A, Time and space partitioning
  - DO-178C-Level A, dynamic loader/linker which is a critical element in enabling binary modularity and the reuse of DO-178 certification artifacts. Released decades ago.
  - Well controlled software component interfaces
  - Component based documentation and XML configuration files
  - No artificial constraints on resources
    - No magic numbers in code
    - API parameters in XML configuration files, not in the code

Binary Reusable Software Components

- The Solution: Binary partitioning isolates change impact & maximizes reuse
- Binary Reusable Software Components is the software equivalent of a black box.

- The APIs are the connections
- Cyclic Redundancy Check (CRC) is the protective casing



© 2019
Confidential and Proprietary

DDC-I
Safety Critical Software Solutions for Mission Critical Systems

Certifiability – Binary Component

- Each Deos Binary Component Is Separately Verified, Having its own:
  - Configuration (XML)
  - Verification (requirements/design review) results & reports
  - Source Code (design/review, code reviews) reports
  - Documentation, including integration & guidance
  - Tests & test adequacy reports (structural coverage analysis)
  - Binary executable with CRC

Binary
Configuration
Verification
Integration
Test ……Etc.
Binary Component
XML

… Simplifies certification & reduces change impact costs

The Deos RTOS

Overview

Avionics Pedigree

Scheduling Partitioning, RMA & 653

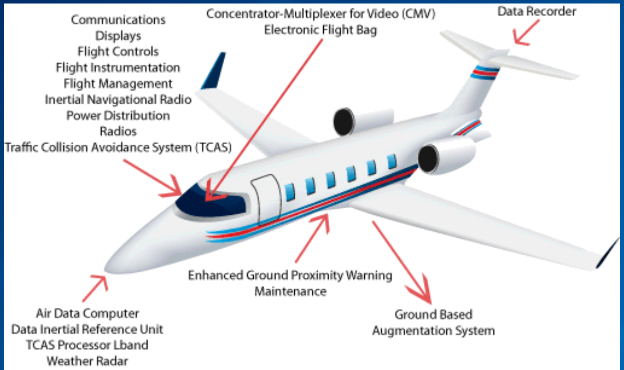
Certified Software Reuse

✦ **Data Distribution Service** ✦

Multicore and FAA CAST-32A

Development Tooling

The Reality of I/O

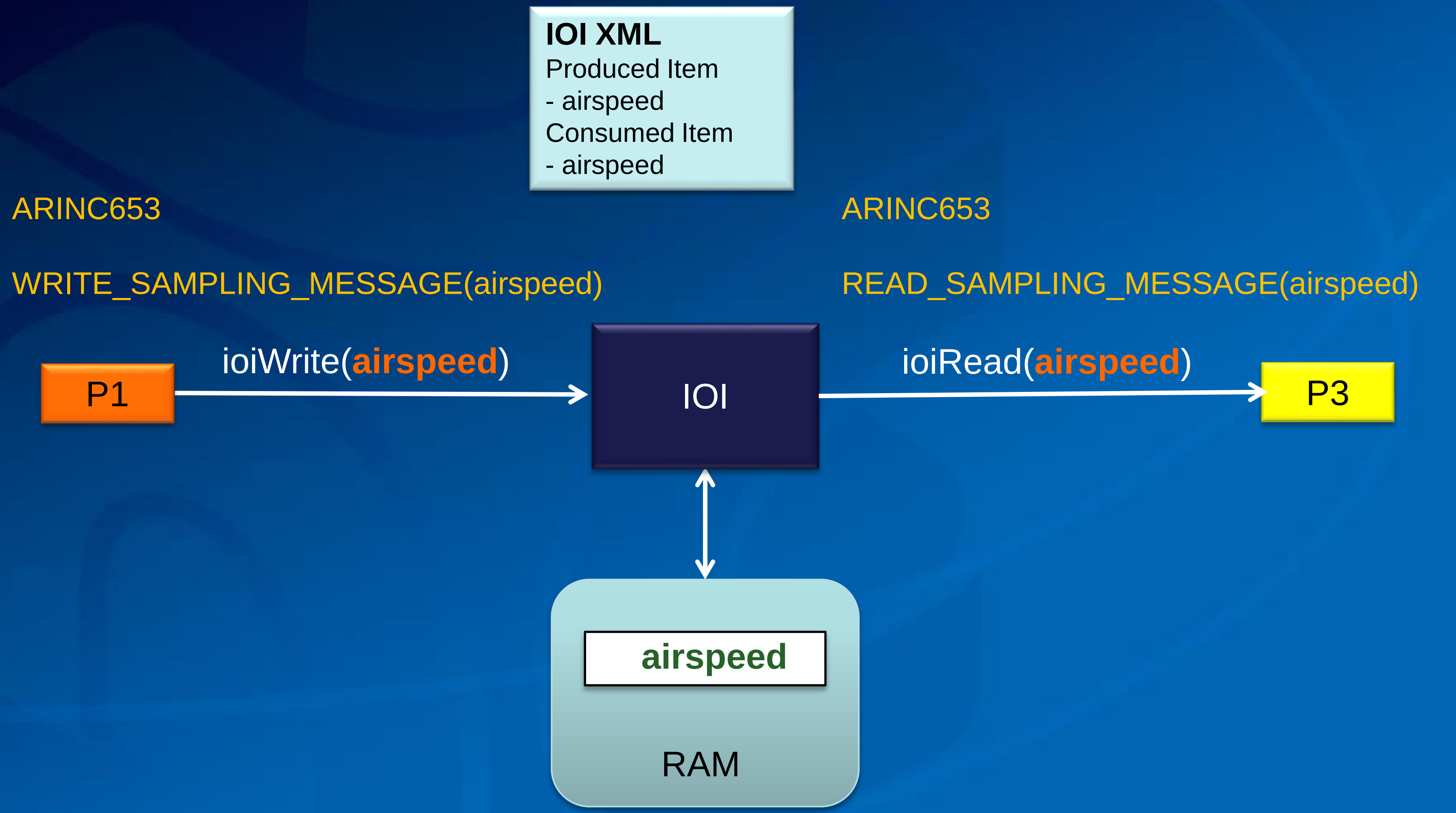


- Aircraft to aircraft, platform to platform, I/O tends to be volatile
  - data sources & destinations
  - engineering units, data types, data rates
- I/O is often a major component of any avionics development
  - large numbers of data items
  - significant change impact driving significant change
- How to isolate your software from this volatility?
  - reduce risk, shorten schedule & minimize cost
  - maximize reuse

Data Distribution (IOI)

- DAL-A User Space Publish/Subscribe Mechanism
- Data handling
  - Periodic and aperiodic
  - Mixed data rates & buffering
  - Freshness checking
  - ARINC 653 (Sampling / queuing ports)
  - Data Queuing (FIFO, Blackboard, LastProduced)
- User-defined formatting functions
  - Data type conversion, unit conversion, etc.
- XML-based configuration
  - Qualified tool

Inter Process Communication



IOI XML
Produced Item
- airspeed
Consumed Item
- airspeed

ARINC653

WRITE_SAMPLING_MESSAGE(airspeed)

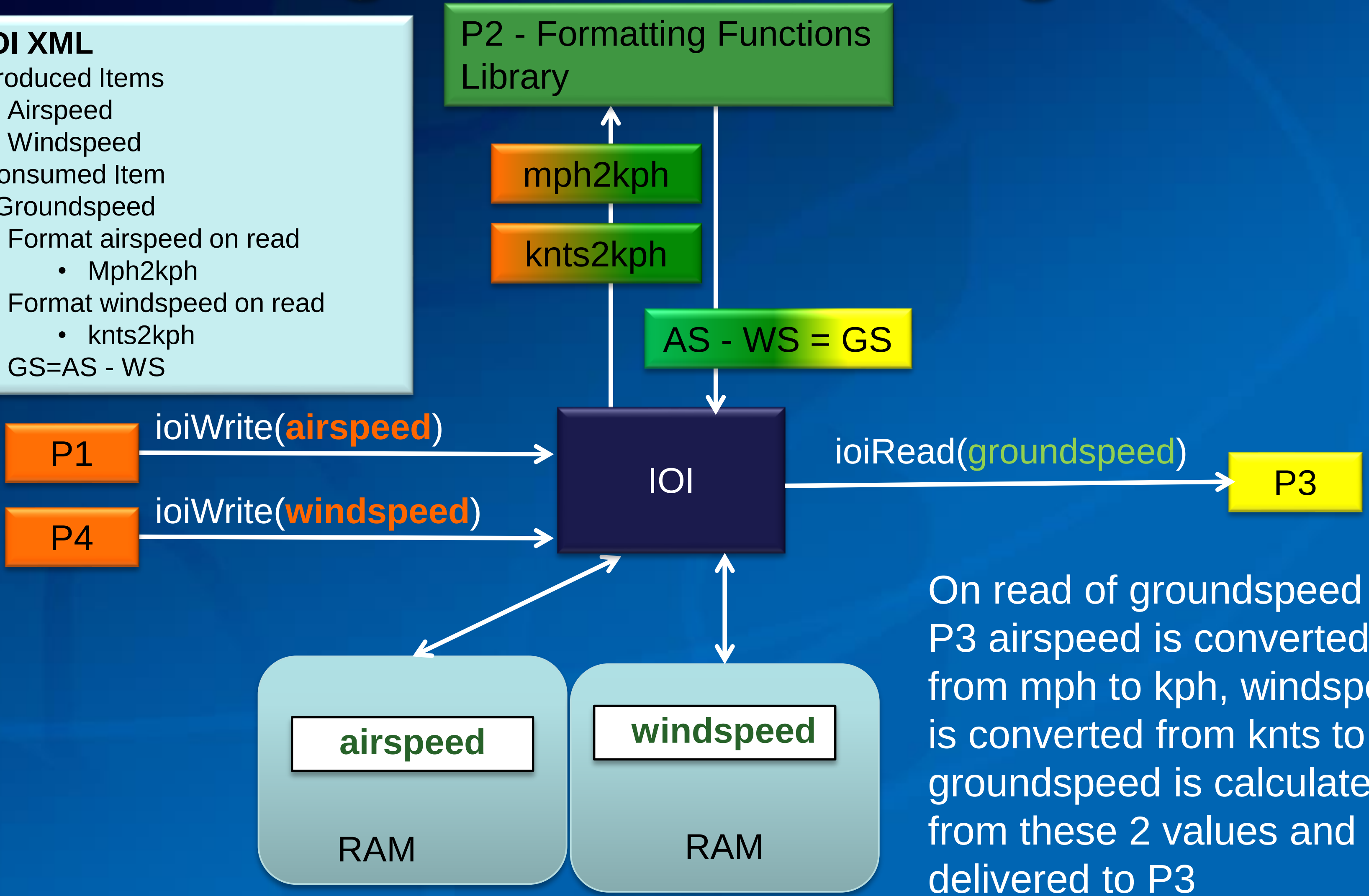
ARINC653

READ_SAMPLING_MESSAGE(airspeed)

Chaining and Formatting

IOI XML
Produced Items
- Airspeed
- Windspeed

Consumed Item
- Groundspeed
- Format airspeed on read
  - Mph2kph
- Format windspeed on read
  - knts2kph
- GS=AS - WS



On read of groundspeed by P3 airspeed is converted from mph to kph, windspeed is converted from knts to kph, groundspeed is calculated from these 2 values and delivered to P3

Binary Modularity & IOI Differentiators

- Think components, interfaces and black box
- Concurrent component development possible
- Minimize/bound change impact
  - Adapt to late-in-program changes
  - Isolate I/O volatility
  - Supports product line development methodology
- Enables independent development & testing
- Allows for-credit, off-target functional testing

Maximize reuse of software & certification artifacts

The Deos RTOS

Overview

Avionics Pedigree

Scheduling Partitioning, RMA & 653

Certified Software Reuse

Data Distribution Service

Multicore and FAA CAST-32A

Development Tooling

Multicore Guidance CAST-32A

- Software Planning
  - How many processors, what OS architectures and how they manage the cores.
- Planning and configuration of MCP
  - Document MCP settings to satisfy requirements
  - Document MCP settings contingency plans
  - Document resource partitioning and how you plan to mitigate contention issues.
- Interference Channels and Resource Usage
  - Identified the interference channels and chosen means of mitigation of the interference.
- Software Verification
  - Verify all the hosted software components function correctly and have sufficient time to complete their execution in the final configuration.
  - Verify that the data and control couplings between all the individual software components hosted on the same core or on different cores.
- Error Detection and Handling, and Safety Nets
- Reporting of Compliance with the Objectives of this Document

https://www.faa.gov/aircraft/air_cert/design_approvals/air_software/cast/cast_papers/media/cast-32A.pdf

Multicore Processor Objectives

OBJECTIVES	DALs	DESCRIPTION	DDC-I COMMENT
MCP_Planning_1:	A, B, & C		
MCP_Resource_Usage_1:	A, B, & C		
MCP_Resource_Usage_2:	A & B		
MCP_Planning_2	A, B, & C		
MCP_Resource_Usage_3:	A & B		
MCP_Resource_Usage_4:	A & B		
MCP_Software_1:	A, B, & C		
MCP_Software_2:	A, B, & C		
MCP_Error_Handling_1:	A & B		
MCP_Accomplishment_Summary_1:	A, B, & C		

MCP Objectives Sample for Deos

OBJECTIVES	DALs	DESCRIPTION	DDC-I COMMENT
MCP_Planning_1:	A, B, & C	The applicant's software plans or other deliverable documents:	
		1) Identify the specific MCP processor, including the unique identifier from the manufacturer,	Deos supports many different multicore processors (MCP), this is product specific to be addressed by target system developer.
		2) Identify the number of active cores,	Deos supports the ability to select the number of which cores of an MCP to use, this determination is product specific to be addressed by target system developer.
		3) Identify the MCP software architecture to be used and all the software components that will be hosted on the MCP	The Deos software architecture is bounded multiprocessing (BMP). The component features of Deos to be used as well as the software applications are product specific to be addressed by target system developer.
		4) Identify any dynamic features provided in software hosted on the MCP that will be activated and provide a high-level description of how they will be used,	Deos allows target system developers to bound dynamic features in both time and space. Consequently, for a given configuration, there is a deterministic bound on execution time and required memory. Additionally, Deos uses a BMP scheduler as part of its bounded execution time guarantee.
		5) Identify whether or not the MCP device will be used in an IMA platform to host software applications from more than one system,	Deos provides support for the development of IMA systems with multiple levels of safety. The desire to take advantage of these features is product specific to be addressed by target system developer
		6) Identify whether or not the MCP Platform will provide Robust Resource and / or Time Partitioning as defined in this document,	The MCP Platform will provide Robust Resource and Time Partitioning as defined in CAST-32A. The Deos product line provides Robust Resource Partitioning and Robust Time Partitioning by giving the target system developer interference channel solutions that range from elimination of the interference channel to a definitive bound on the interference channel utilizing features like Safe Scheduling, Cache Partitioning, and bounding memory transactions.
MCP_Resource_Usage_1:	A, B, & C	The applicant has determined and documented the MCP configuration settings that will enable the hardware and the software hosted on the MCP to satisfy the functional, performance and timing requirements of the system.	DDC-I provides a detailed users guides for the functionality and configuration of Deos. The target system developer is responsible for using these Users Guides to ensure correct configuration as well as CBIT check of configuration, if applicable

Safety Critical Multi-Core

Safety Critical Multicore Concerns:

1. Bound & control interference patterns
   - A. Minimize contention for shared resources (e.g., cache, memory, & devices)
   - B. Coordinate behaviors amongst cores
2. Getting good value from adding secondary cores

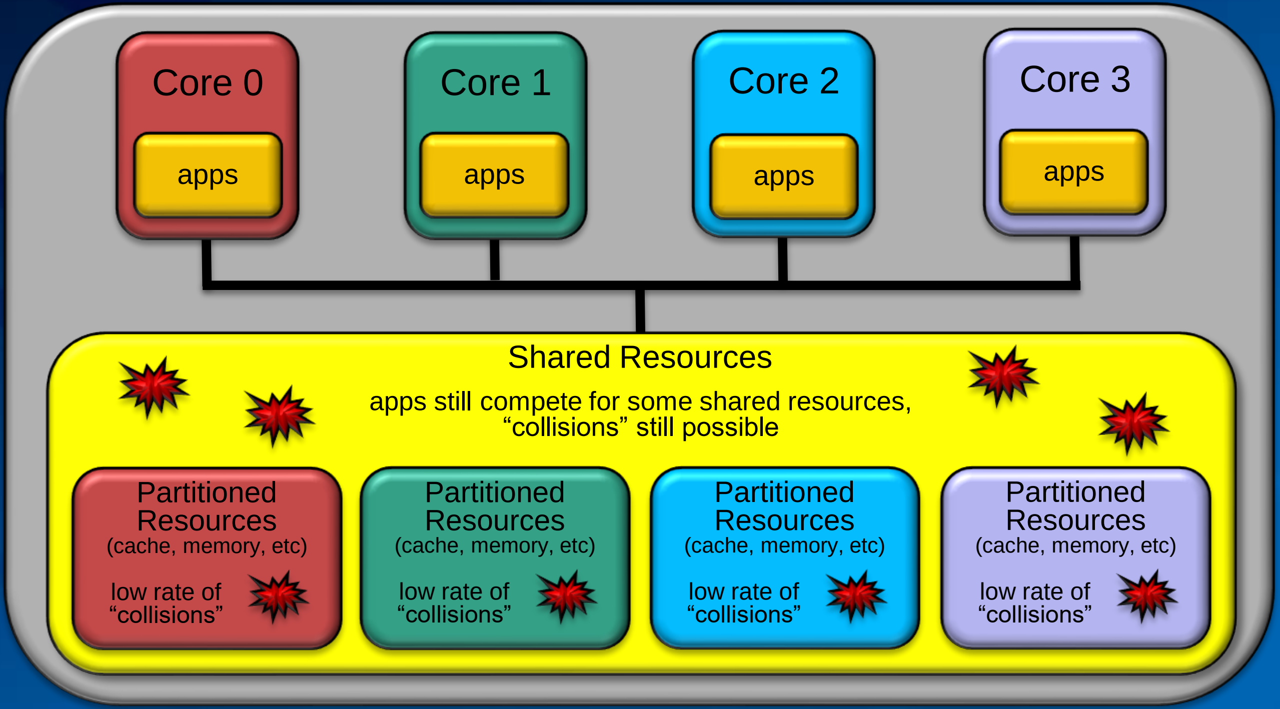
   Example concern: WCE will increase due to multicore interference patterns

Deos Multicore Solutions: SafeMC™

1. Reduce interference patterns and reduce WCETs
   - A. Memory pooling & cache partitioning[1]
   - B. Safe-scheduling
2. Performance enhancing features
   - A. Concurrent execution as a performance advantage
   - B. Slack scheduling[1], including Window Activation[1] for multicore
     - Recovers and applies additional slack resulting from higher WCETs
   - C. Enable deterministic interrupting devices

[1] Patented

Multicore Scheduling with Deos



- Critical and non-critical apps can run on any core
- Each core has dedicated (partitioned) resources

- All cores turned on
- Resource contention is low
- Resource utilization is maximized

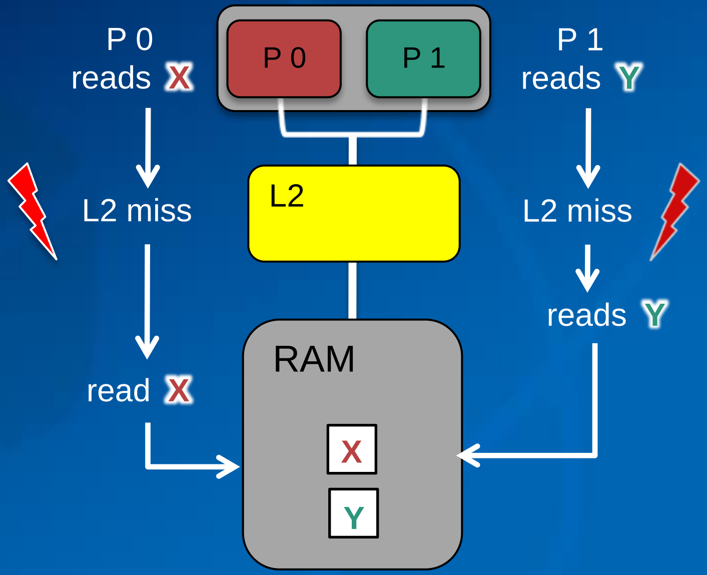
… highly-utilizes processing resources & maximizes SW design flexibility.

Cache In Deterministic Systems

- The greatest performance factor for modern processors
- Growing in size and number of levels (e.g., L1, L2, and L3)
- Left uncontrolled, cache will cause performance variability (e.g., cache thrashing which increases the gap between best and worst case execution time (WCET))

Studies show that cache variability must be resolved in deterministic multicore systems

Cache Contention



Processes/Partitions 0 & 1 access RAM:

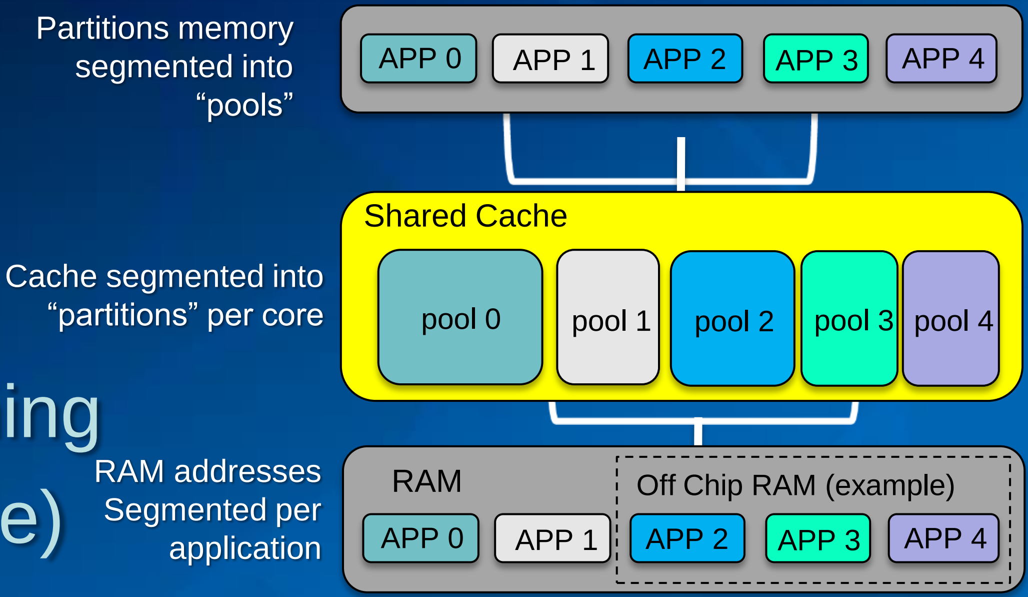
- Each overwrite portions of cache
- Resulting cache ‘thrashing’ makes WCE difficult to determine
- In the end, P0 & P1 pay the “cost” of cache misses

Cache Performance Variability Solutions

Cache variability is a significant issue for deterministic systems, that must be solved. Fixes include:

1. Cache flushing (e.g., flush cache between applications)
   - Good: Reduces performance variability
   - Bad: Forces cache flush overheads at an application context switch
2. Disabling of cache
   - Good: Eliminates cache performance variability
   - Bad: Huge performance penalty – forces the processor to a low level of performance. Also impractical for multicore processors.
3. Cache Partitioning – Several option with various results
   - Deos cache partitioning (patented)
   - Cache locking

Deos Memory Pools & Cache Partitioning

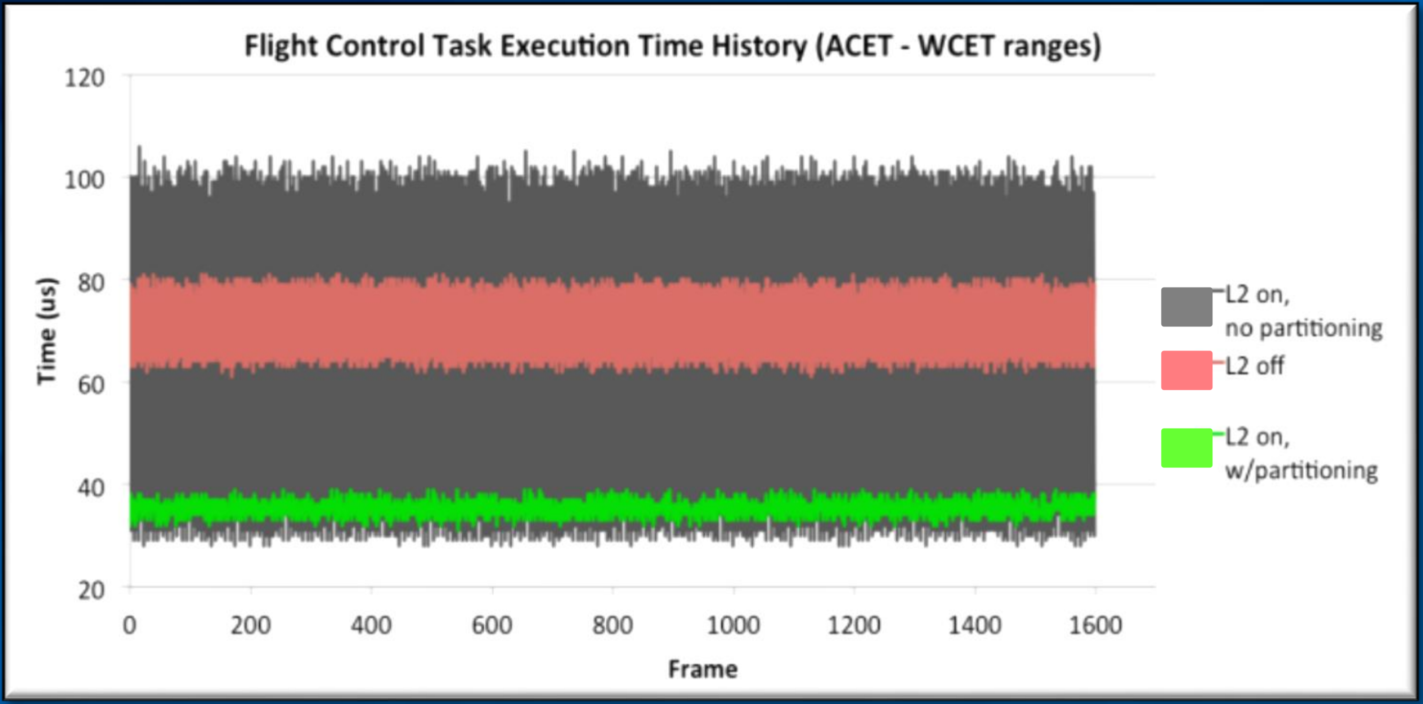


Partitions memory segmented into "pools"
APP 0
APP 1
APP 2
APP 3
APP 4
Shared Cache
Cache segmented into "partitions" per core
pool 0
pool 1
pool 2
pool 3
pool 4
RAM addresses Segmented per application
RAM
Off Chip RAM (example)
APP 0
APP 1
APP 2
APP 3
APP 4

- Reduces cache thrashing
- All in software (portable)
- Memory Pooling - Enables physical memory segmentation (a key advantage for microcontrollers
- Partition per application per core.
- No application specific code
- No cache locking instructions used

… minimizes cache contention, maximizes cache hits, and improves WCETs.

Cache Partitioning – Bounding WCETs



- Bounds & controls cache interference patterns
- Can dramatically improve WCET performance

Fine Grain Locking

- In the kernel all locking is done in a single core space only, therefore, no cross core blocking is possible.
  - No cross core locks (No resources used for all cores)
  - No single lock for scheduling (each core has a scheduler)
  - No single lock for all kernel interface objects (each object created has its own lock)
- Cross core blocking is only possible if a developer designs it to happen
  - Threads on different cores share a kernel interface object (semaphore, event, mailbox, etc.)
  - Thread creates another thread and schedules it on a different core
  - Threads of different cores share a memory pool
  - In these cases affects limited to the cores in question and not the others.

Safety Nets

- Built-in Performance Counters
  - Certain processors have hardware capability to count processor level events.
  - Setup CPU to send interrupt when a particular performance counter threshold is reached.
  - Last level cache miss is a selectable event to be counted.
  - When threshold is hit for a particular partition, a decision can be made to on how to deal with the offending partition.

Addressing CAST-32A

- Safe Scheduling
  - Addressing aligning applications to minimize contention for shared resources
- Memory Pooling and Cache Partitioning
  - Addressing reducing the activity on the memory controller/memory bus by reducing or elimination the need for cache flush and refills.
- Safety Net
  - Utilizing the hardware available capability to monitor activity and enable the throttling of the offending application.

Key Performance Differentiators

- Multicore
  - Addresses the objectives in CAST-32A
  - Multicore features implemented in software
  - Significantly reduces porting effort, expands processor coverage, etc.
- Cache partitioning & memory pools
  - Minimize WCET & maximize CPU utilization
  - Reduce interference in single & multi-core systems
  - Simplify verification & certification processes

The Deos RTOS

Overview

Avionics Pedigree

Scheduling Partitioning, RMA & 653

Certified Software Reuse

Data Distribution Service

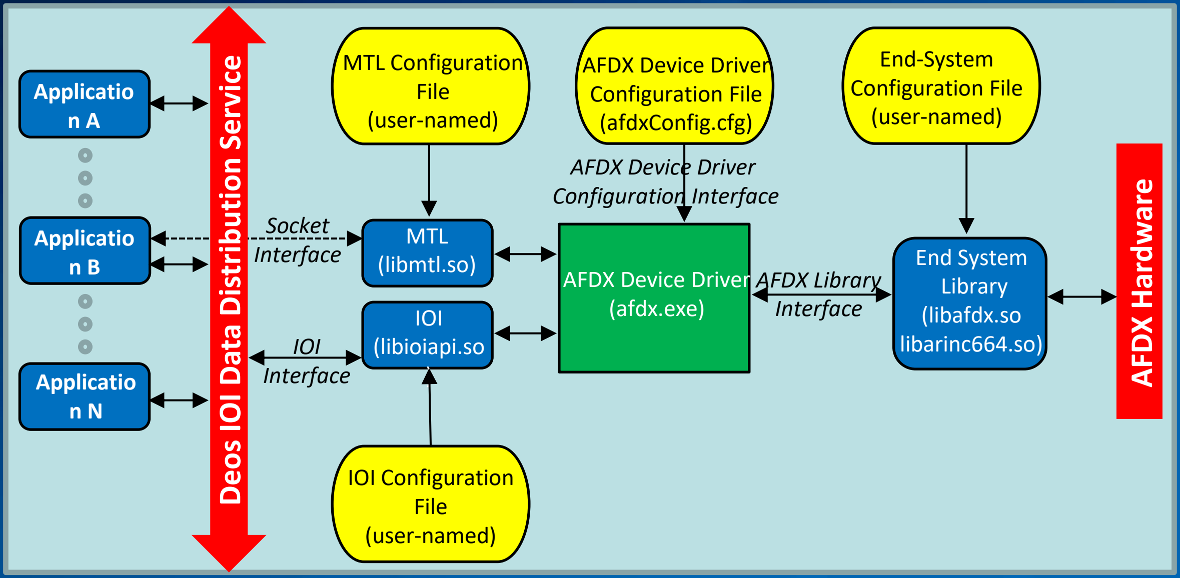
Multicore and FAA CAST-32A

Security Building Blocks

✦ **Additional Features** ✦

Development Tooling

Optional: Deterministic & Robust Network

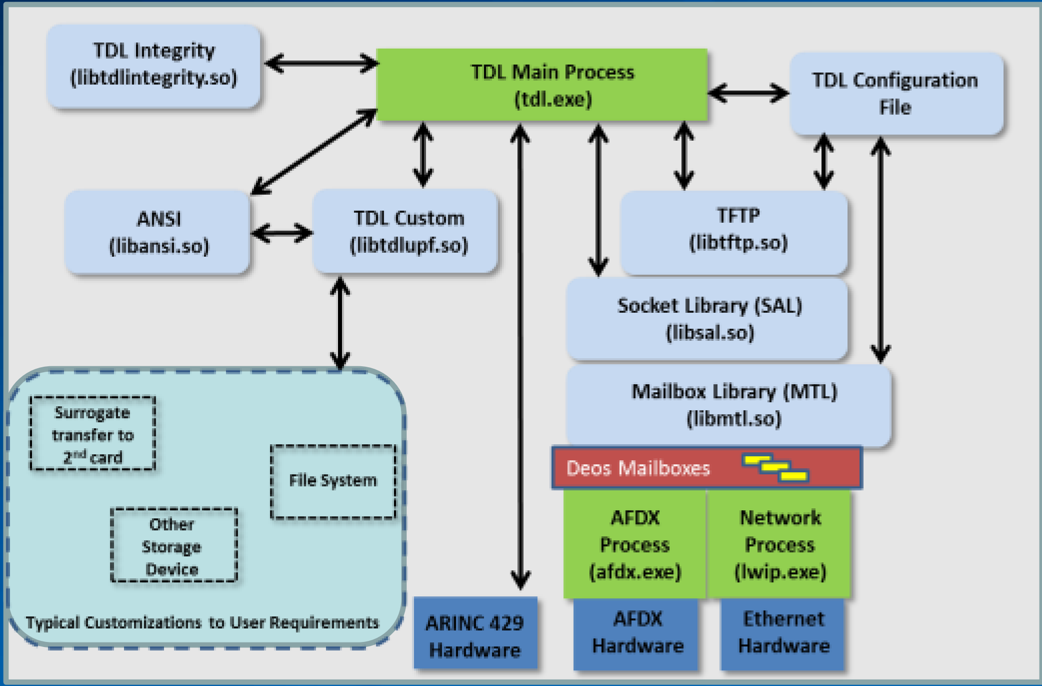

Application A
Application B
Application N
Deos IOI Data Distribution Service
Socket Interface
IOI Interface
MTL Configuration File (user-named)
MTL (libmtl.so)
IOI (libioiapi.so
IOI Configuration File (user-named)
AFDX Device Driver Configuration File (afdxConfig.cfg)
AFDX Device Driver Configuration Interface
AFDX Device Driver (afdx.exe)
AFDX Library Interface
End-System Configuration File (user-named)
End System Library (libafdx.so libarinc664.so)
AFDX Hardware

- Deos DAL-A
- Single/Multi-core
- Arm, x86, PowerPC
- User Space Libraries
- Collins V5/V6

  (TTTech upcoming)
- Data Decoupling and Reconfiguration
  - Optional connection to Deos IOI data distribution service
  - ARINC-664/AFDX packet data can change without recompiling
  - XML configurable, with qualifiable tooling

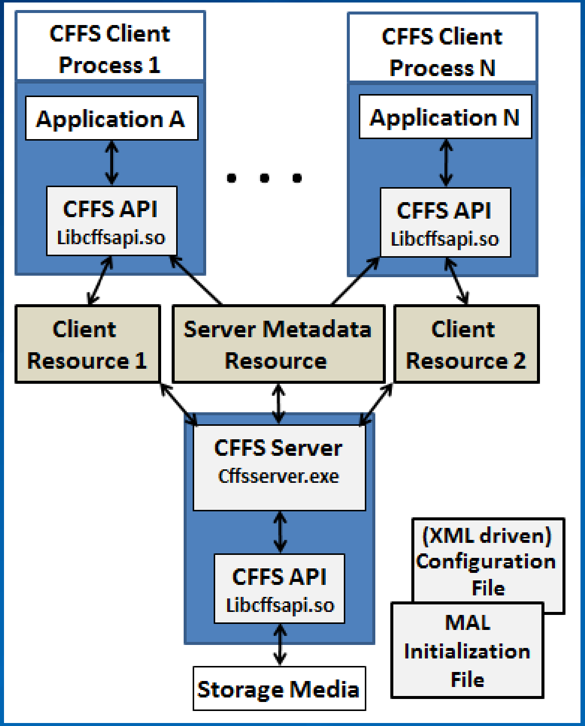
Optional: ARINC-615 Target Data Loader

- Supports Ethernet, ARINC 664/AFDX, ARINC 429
- DAL A Integrity Library
- Customizable to Specific User Requirements
- Portability via XML Configuration File
- Supports ARM, PowerPC and x86

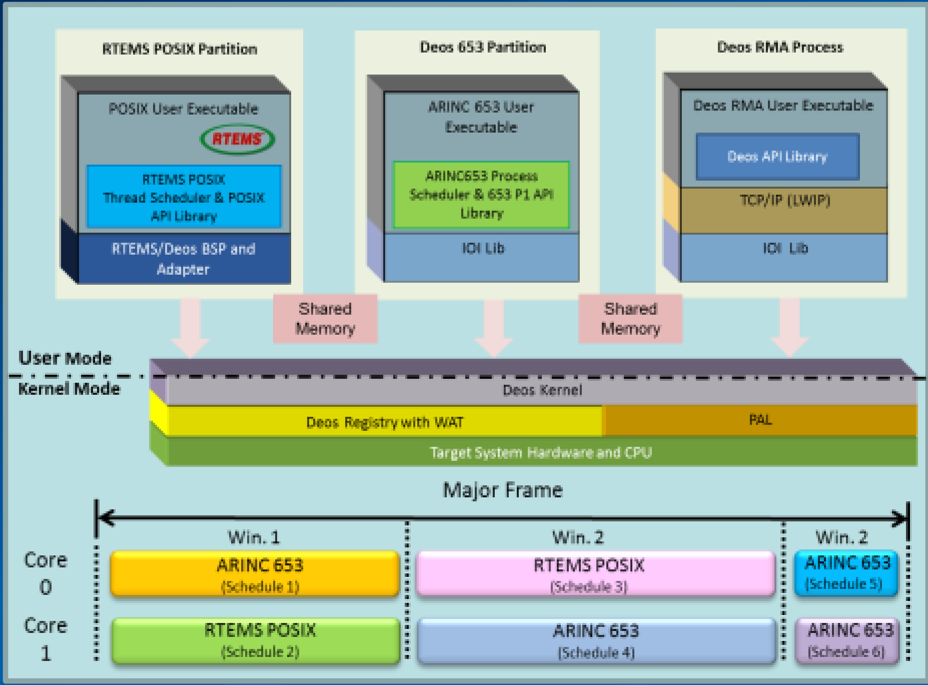

TDL Integrity (libtdlintegrity.so)
TDL Main Process (tdl.exe)
TDL Configuration File
ANSI (libansi.so)
TDL Custom (libtdlupf.so)
TFTP (libtftp.so)
Socket Library (SAL) (libsal.so)
Mailbox Library (MTL) (libmtl.so)
Surrogate transfer to 2nd card
File System
Other Storage Device
Typical Customizations to User Requirements
Deos Mailboxes
AFDX Process (afdx.exe)
Network Process (lwip.exe)
ARINC 429 Hardware
AFDX Hardware
Ethernet Hardware

Optional: Certifiable Fast File System (CFFS)

- DAL-A User Space Module on ARM, PowerPC, or x86
- Single and multicore operation
- Power-fail-safe (journaled)
- Optional ARINC-653 Part 2 API
- Supports mixed DAL applications
- Block oriented high speed transfers
- Portable, with binary & certification reuse (XML configurable)
- Multiple Storage options (e.g., SATA, CFlash, NAND, NOR)
- Protected against non-authorized writes



Optional: POSIX (RTEMS) Library



- Reusable Deos library
- FACE Safety Base POSIX Profile
- Single or multicore operation
- ARM, PowerPC, & x86
- Applications configured via an XML based file, with a qualified checker
- POSIX RTEMS paravirtualized in a Deos time and space partition
  - Priority pre-emptive RTOS
  - RTEMS - Mature, stable, and open source RTOS supporting POSIX since 1990
  - Employed on numerous safety/mission critical military and space applications
- Development environment integrated seamlessly with Deos tool chain
  - Compiler/linker/debugger integrated with DDC-I Open Arbor Eclipse framework
  - Deos profiling tools exhibit POSIX application resource and timing usage

The Deos RTOS

Overview
Avionics Pedigree
Scheduling Partitioning, RMA & 653
Certified Software Reuse
Data Distribution Service
Multicore and FAA CAST-32A
Security Building Blocks
Development Tooling

OpenArbor IDE & Tools

- OpenArbor IDE
  - DDC-I’s eclipse-based IDE
  - support for C/C++
- Development Tools
  - compiling & debugging
    - gcc 7.3
  - development targets
    - target simulation
    - reference boards
    - ARM, MIPS, PPC, x86
- Verification Tools
  - status monitor
  - time map
  - IOI for simulation/test
  - structural coverage
  - qualified verification tools
- Configuration Tools
  - XML-based integration
    - Deos kernel, IOI, etc
  - timing analysis

… industry standard eclipse IDE and full toolset for developing, verifying & configuring mission/safety-critical applications.

OpenArbor IDE

- OpenArbor
  - DDC-I's Eclipse-based IDE
  - Connects to command line tools

    (e.g., can be integrated with customer Perl test scripts)
- Target Interface
  - Standard: Ethernet & FTP
    - No need to download large apps – quick debug/compile/debug cycles
  - Optional: JTAG
- Floating FlexNet controlled license
  - No project restrictions
  - Can be 'borrowed' from servers

Compiling & Debugging

- C/C++*
  - GCC – the defacto industry standard, www.gcc.gnu.org
  - Not bound to a proprietary compiler, or specific (older) version.
  - User selectable optimization settings
- Ada 95 (optional)
- Debugger
  - Full-featured mixed language Eclipse-based debugging
  - Via Ethernet or JTAG

Development Targets

- Virtual target – QEMU
  - Open source processor emulator
  - Supports ARM, MIPS, PPC & x86 instruction sets
  - www.qemu.org

- Reference targets
  - ARM,
  - PowerPC
  - x86
  - MIPS

Third Party Partners

Communications

Collins Aerospace

rti Your systems. Working as one.

TTTech

TWINOAKS COMPUTING INC PRACTICAL MIDDLEWARE EXPERTISE

COTS Processor Boards

AXIOM MANUFACTURING

DATA PATTERNS

X-ES

abaco SYSTEMS

Aitech

ARROW

CURTISS-WRIGHT

EMBEDDED PLANET Embedded Design

fit-PC Quality fanless computers

kontron AN S&T COMPANY

minnowboard.org

mercury systems.

North Atlantic Industries

Portwell

Processors

Contact DDC-I for respective design reference card BSP availability

ALTERA®

AMD Embedded Solutions

ARM

intel Intelligent Systems Alliance

NXP

TELEDYNE e2v Everywhereyoulook™

XILINX®

Security

MOCANA

STAR LAB

wolfSSL

Synergistic Application Software

COREAVI

ENSCO®

HCC embedded

LOGICIRCUIT

DAR AN INTREPID COMPANY

Tools

LAUTERBACH DEVELOPMENT TOOLS

ANSYS

LDRA

Performance Seeing Things Differently

MathWorks®

RAPITA

VECTOR software

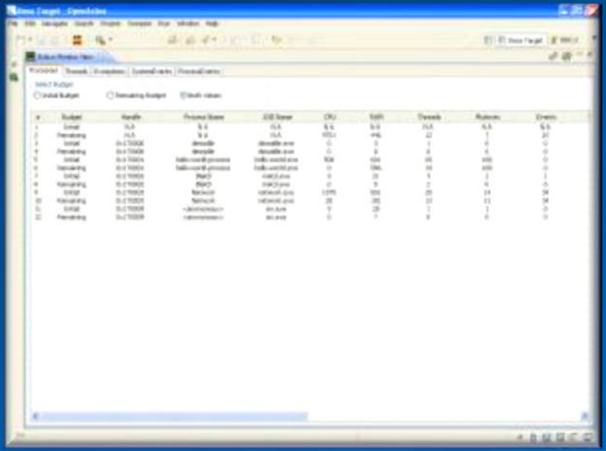
Profiler Tools

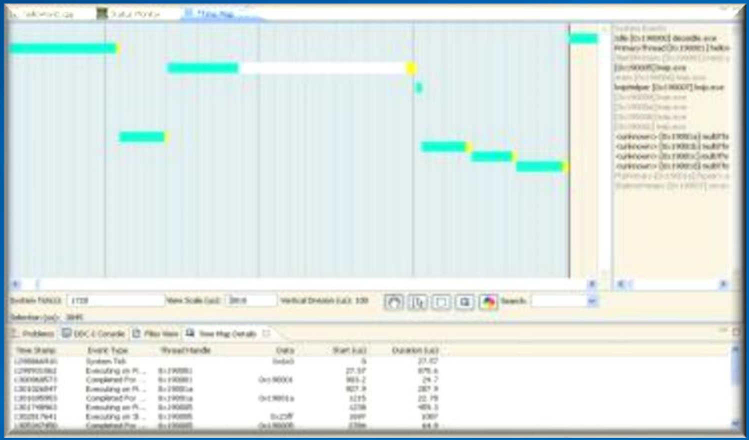
Status Monitor

- Logs events & exceptions
- Time budgets
- Stack size
- RAM quotas
- Semaphores
- Mutexes
- Mailboxes
- … etc

Time Map

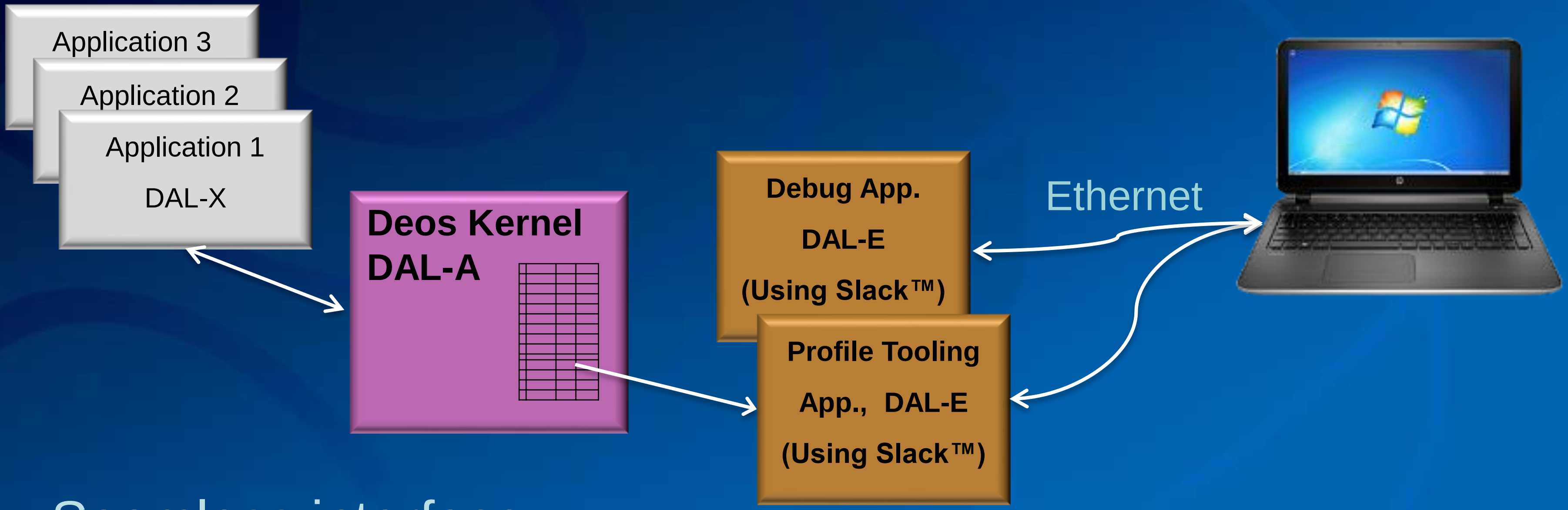
- Graphical timeline display
- Monitors thread execution, interrupts, events & exceptions





- Useful throughout V&V
  - No application instrumentation or system timeline intrusion
- Exportable (e.g., to .xls)

Deos V&V Tooling Advantage

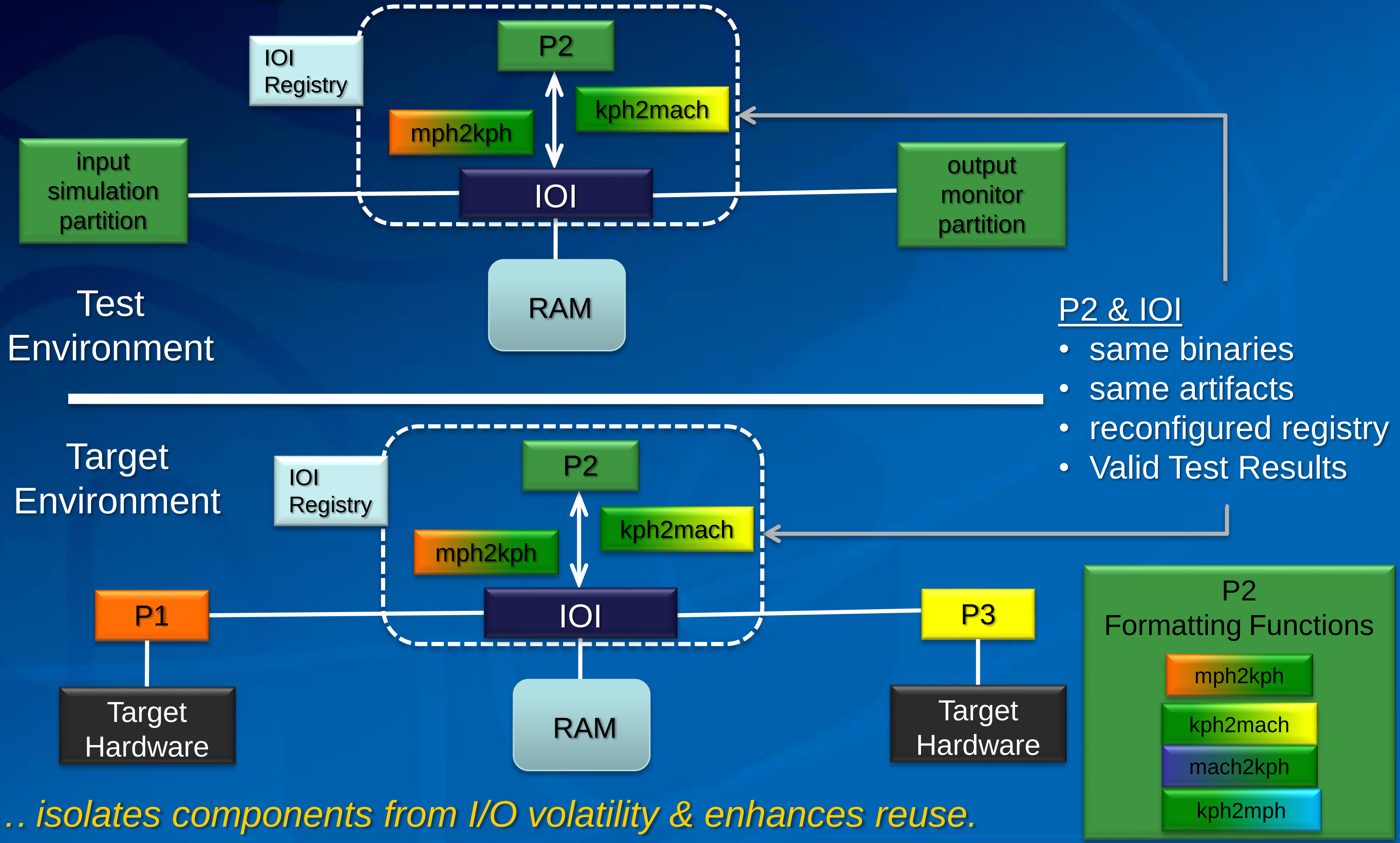


- Seamless interface
- Profile information gathered from DAL-A kernel

  Information may also be used for app. monitoring, watermarking, etc.
- Profile & debug tools don't impact application timelines

No instrumented app + no timeline impacts = use in V&V

Configurable I/O & Verification



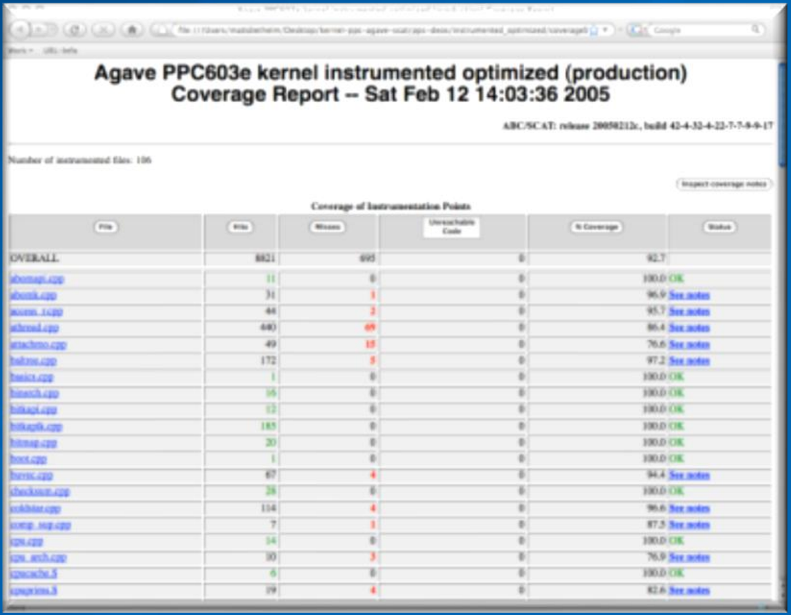
P2 & IOI

- same binaries
- same artifacts
- reconfigured registry
- Valid Test Results

… isolates components from I/O volatility & enhances reuse.

Structural Coverage Analysis

- Based on Assembly Branch Coverage
  - Performed at object-code level
  - Demonstrated as equivalent to MC/DC at source-code level
  - No source-object traceability required for DAL A
- No compiler qualification required
- Qualified verification tool



Agave PPC603e kernel instrumented optimized (production)
Coverage Report -- Sat Feb 12 14:03:36 2005

ABC/SCAT: release 20050212c, build 42-4-32-4-22-7-7-9-9-17

Number of instrumented files: 106

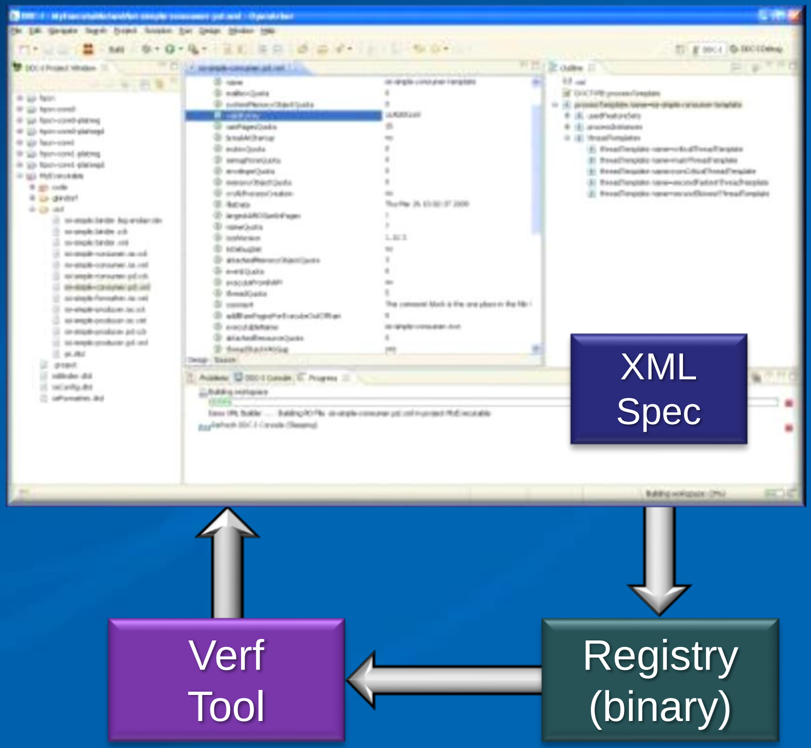
Inspect coverage notes

Coverage of Instrumentation Points

File	Hits	Misses	Unreachable Code	% Coverage	Status
OVERALL	8821	695	0	92.7	
abomapi.cpp	11	0	0	100.0	OK
abortk.cpp	31	1	0	96.9	See notes
access_1.cpp	44	2	0	95.7	See notes
athread.cpp	440	69	0	86.4	See notes
attachmo.cpp	49	15	0	76.6	See notes
baltree.cpp	172	5	0	97.2	See notes
basics.cpp	1	0	0	100.0	OK
binsrch.cpp	16	0	0	100.0	OK
bitkapi.cpp	12	0	0	100.0	OK
bitkaptk.cpp	185	0	0	100.0	OK
bitmap.cpp	20	0	0	100.0	OK
boot.cpp	1	0	0	100.0	OK
bsysc.cpp	67	4	0	94.4	See notes
checksum.cpp	28	0	0	100.0	OK
cokhlstar.cpp	114	4	0	96.6	See notes
comp_sup.cpp	7	1	0	87.5	See notes
cpu.cpp	14	0	0	100.0	OK
cpu_arch.cpp	10	3	0	76.9	See notes
cpucache.S	6	0	0	100.0	OK
cpuprims.S	19	4	0	82.6	See notes

Integration Tool

- For configurable components (e.g., apps, IOI, AFDX, etc.)
- XML-based specification
  - Time budgets
  - Execution rates
  - RAM quotas
  - Resource “ownership”
  - … etc
- Generates binary registry
- Qualified verification tool
  - Checks syntax correctness
  - Enables binary registry
    Correctness to XML specification



WCET Timing Analysis

- Cache Trasher Technology
  1. Cache Trasher puts software under test in the worst-case cache condition (i.e., cache write back)
  2. Use a test set that drives the worst-case paths
  3. Measure the WCET (e.g., with Deos profiler tools), and budget accordingly (e.g., in Deos Integration Registry)

- Critical Time Kernel
  - Measure worst-case target-specific timing
    - Context switch, kernel critical path execution, etc

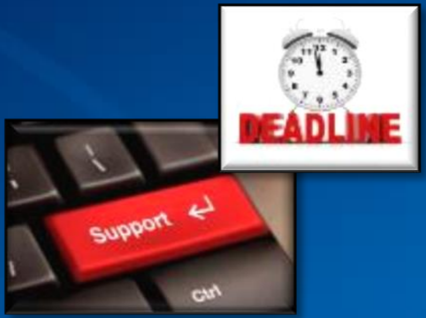
    Values then inserted in the Deos Integration Registry

Training

- Deos training
  - Deos API
  - BSP & driver development
  - Integration & certification
  - Held at customer site
  - Taught by experienced avionics engineers

… best-in-class training from experienced engineering staff.

Support & Services



- Main channel: support@ddci.com
- Direct access to engineering team
  - 25% of each engineer’s time dedicated to customer support
- Optional on-site support (ideal for start-up & integration phases)
- Defense of certification artifacts
- BSP & driver development
- Consulting engineering

… best-in-class support from experienced engineering staff.

Key Tooling, Test & Support Differentiators

- Floating Dev. License includes all tools
- Tools can be used through V&V (and flight, as applicable)
- Independent of compiler (current version, user sets optimizations)
- Tooling for WCE determination (apps and target)
- Processor emulator enables immediate development
- Identical binaries tested on reference & target HW
- Qualified source & object code coverage tool
- Direct support by DDC-I Engineering
- Training & support
  - Get what you need, when you need it, from people with best-in-class experience – all from a single company focused on avionics.

Summary

- Pedigree
  - Low risk
- RTOS solutions
  - Reuse, features, & performance = low cost of ownership
- Tools
  - Shorter V&V process
- Support, training & services
  - DDC-I's team augments yours

Thank you!

Contact Information

Laurent Meilleur lmeilleur@ddci.com

Theresa Rickman trickman@ddci.com

Gary Gilliland ggilliland@ddci.com

www.ddci.com