IV&V on Orion's ARINC 653 Flight Software Architecture

Adelbert Lagoy and Todd A. Gauer

Agenda

- ARINC 653/DO 178 background
- Advantages to a 653 FSW architecture
  - Development
  - V&V
  - Maintenance
- Available COTS 653 systems
- The impact of the selection of a ARINC 653 approach on baseline documentation
  - Alignment and misalignment of ARINC 653 to NASA documentation
- IV&V impacts from application of 653 approach
  - Architecture verification impacts
  - Requirements validation impacts
  - FSW code verification impacts
  - Potential recertification impacts
- Other impacts (adverse impacts)
  - “overhead” from 653
  - Static partition definitions
  - COTS OS opacity
- Orion IV&V Experience

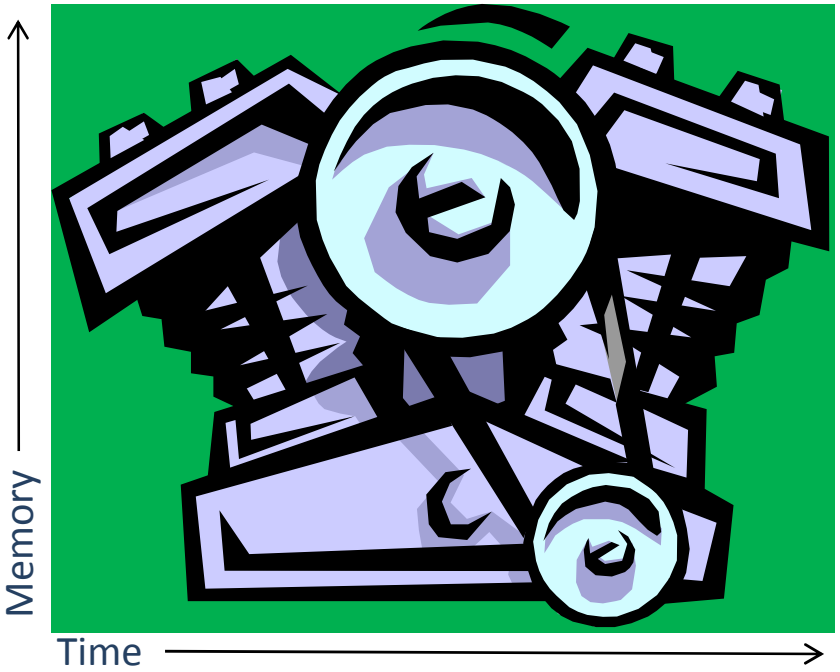
ARINC 653/DO 178 background

ARINC 653

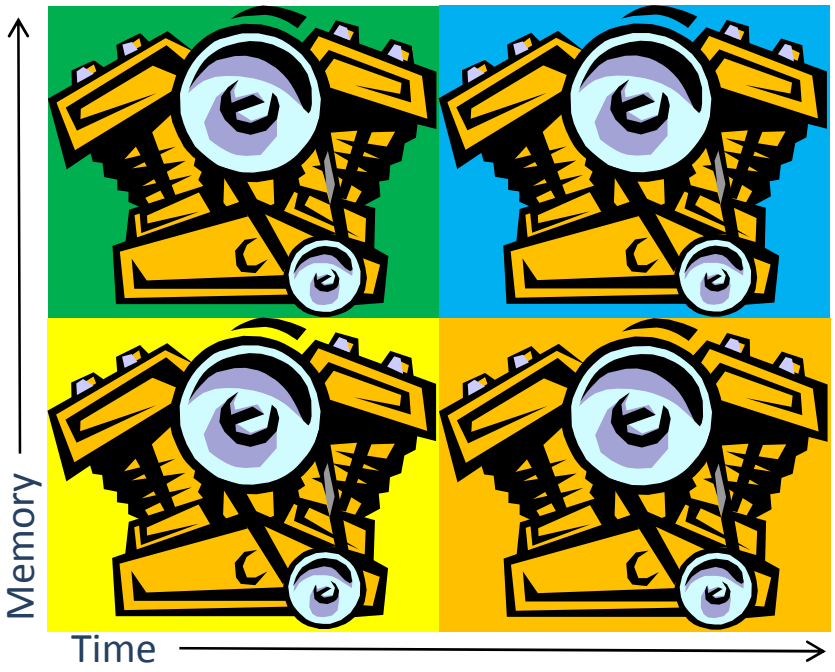
- The Aeronautical Radio, Incorporated (ARINC) specification ARINC 653 is a software time and space partitioning standard for Real Time Operating Systems (RTOSs).
- The ARINC 653 standard supports Integrated Modular Avionics (IMA) architecture allowing appropriate integration of avionics software of differing levels within a single hardware device.
- ARINC 653 provides a level of fault protected operation.
  - Faults within a partition should not stop other partitions from executing.
- *Metaphor – ARINC 653 compliant system's partitions are like "virtual flight computers" within the flight computer.*

Metaphor – ARINC 653

Conventional Monolithic System



ARINC 653 System with 4 Partitions



- ARINC 653 splits the available processor time and space into partitions (partitions do not need to be the same size).
- When we talk “partition” in this discussion you can substitute “virtual flight computer” if that is helpful.

ARINC 653/DO 178 background

ARINC 653

- Each partition is assured of its allocated processing time and memory
  - Lockup, freeze, endless loop, overrun, etc within one partition will not prevent another partition from operating.
  - “Data starvation” of running partitions from faulted partitions ceasing to deliver necessary data is possible.
- ARINC partitioning allows:
  - Safer mixing of multiple software criticality levels in a single flight computer.
  - Highly structured interface controls
  - Assured access to processor time and memory by each partition.

Advantages to an ARINC 653 FSW architecture

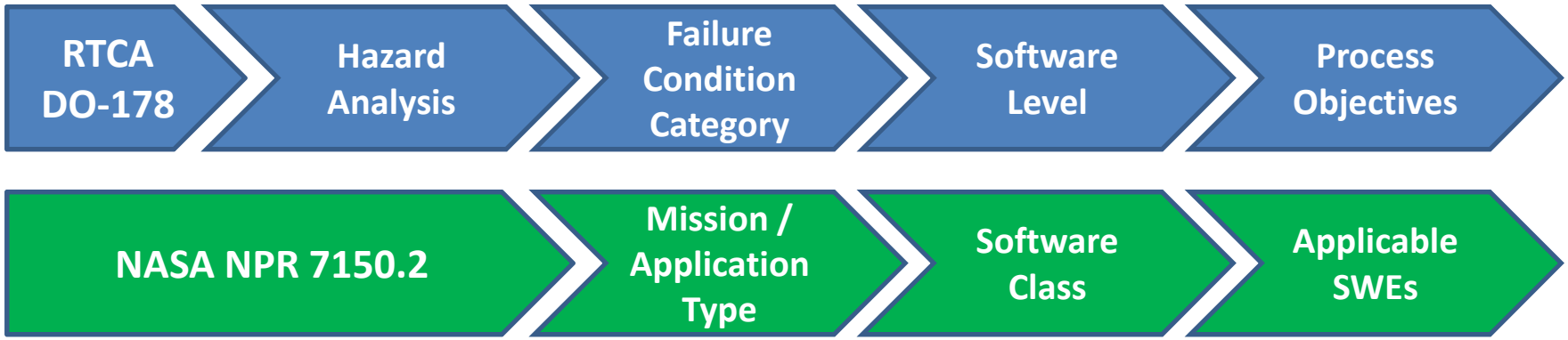
- Development
  - Formalized partitions should ease development of highly complex systems
  - Abstracted from HW, should support increased processor portability
  - COTS OS SW presents reduced cost/schedule risk
- V&V
  - Verification of Partitions should be less complex as greater verification credit can be taken from unit level testing (less integrated testing should needed)
  - Less regression testing for bug fixes if properly architected
- Maintenance
  - Less regression testing for bug fixes

NASA and ARINC

- NPR 7150.2A does not incorporate ARINC 653 partition thinking into its practices.
  - *"For a given system or subsystem, software is expected to be uniquely defined within a single class."*

DO 178 background

- DO 178B *Software Considerations in Airborne Systems and Equipment Certification* from RTCA
- DO 178B processes are an accepted path to FAA certification
- DO-178 Software levels set by HA results
  - Software Levels establish process objectives
  - Similar to NPR 7150.2 Classes



Comparison of DO 178 and NPR 7150.2A

Class	Class Descriptions	SWEs
A	**Human Rated Space Software Systems -** needed to perform a primary mission objective of human space flight and directly interacts with human space flight systems	132
B	**Non-Human Space Rated Software Systems or Large Scale Aeronautics Vehicles –** software must perform reliably to accomplish primary mission objectives, or major function(s)	132
C	**Mission Support Software or Aeronautic Vehicles, or Major Engineering/Research Facility Software -** software necessary for the science return from a (non-primary) instrument	118
D	**Basic Science/Engineering Design and Research and Technology Software** Ground software that performs secondary science data analysis,	74
E	**Small Light Weight Design Concept and Research and Technology Software** Software developed to explore a design concept/hypothesis, not used to make decisions for a Class A, B, or C system	34

Level	Software Level	Objectives	Objectives with Independence
A	Software whose anomalous behavior could cause/contribute to a catastrophic failure condition for the aircraft	66	25
B	Software whose anomalous behavior could cause/contribute to a hazardous/severe-major failure condition for the aircraft	65	14
C	Software whose anomalous behavior could cause/contribute to a major failure condition for the aircraft	57	2
D	Software whose anomalous behavior could cause/contribute to a minor failure condition for the aircraft	28	2
E	No Impact to safety, aircraft operation or crew workload	0	0

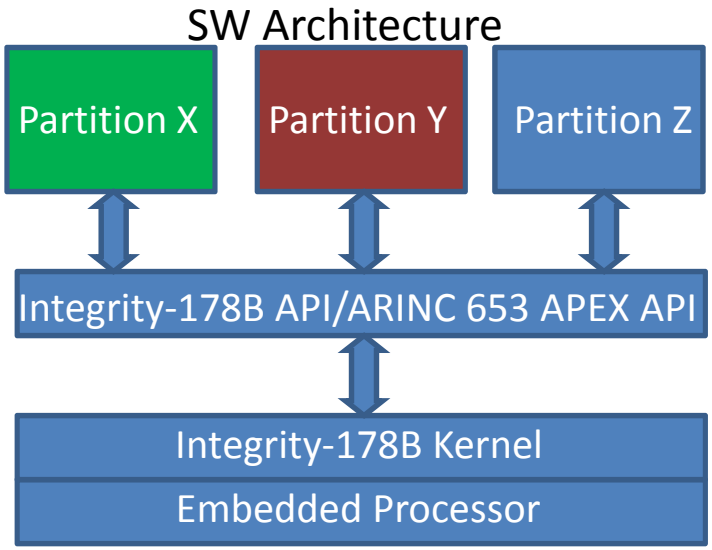
ARINC 653/DO 178 background (cont.)

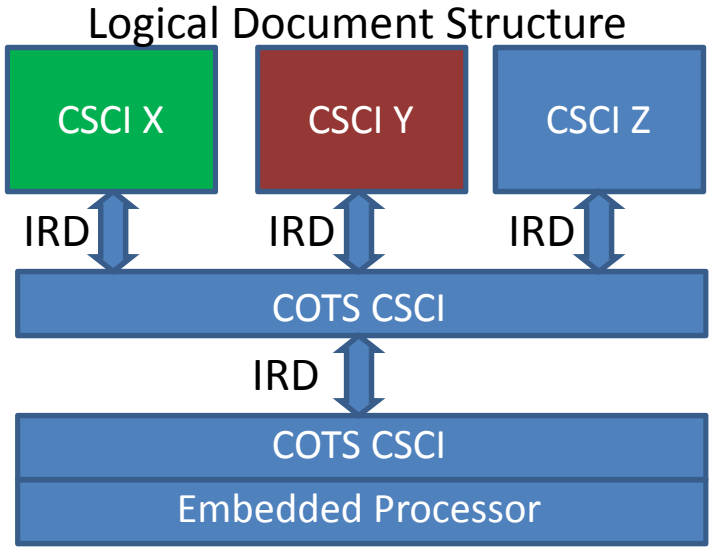
- DO 178B is a process specification, as such it is not inspectable but depends on application in development (similar to some NASA practices)
- DO 178 verification data packages (showing conformance to the necessary processes) are sold separately
  - Orion has elected to not purchase the DO 178 verification data package

Available COTS ARINC 653 systems

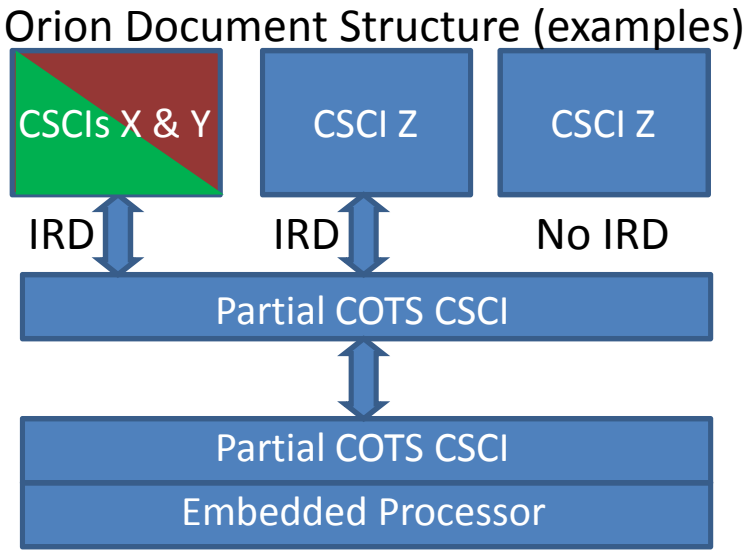
- Green Hills Integrity 178B
  - Used by Orion
- Wind River VX Works 653 Platform
  - Flight and command computers in the Ares I
- LynuxWorksOS-178
  - Commercial avionics
  - UAV avionics

Document Structure

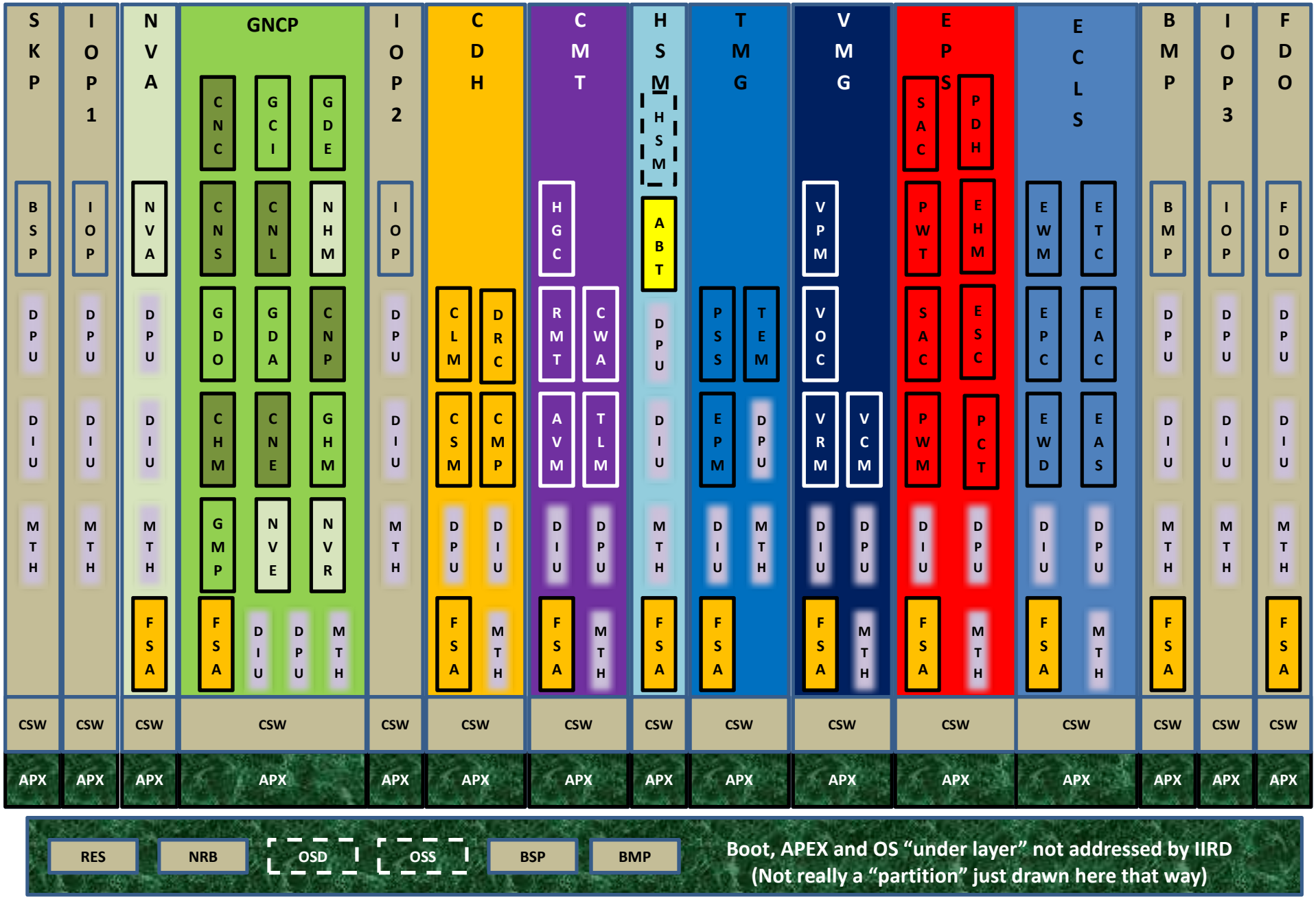




- ARINC 653 partitions provide a natural division for CSCI definitions
- Alignment of partitions to CSCIs supports effective documentation development, regression testing, V&V, structured integration, and eases development
- Orion does not follow the logical approach resulting in undocumented CSCI to CSCI interfaces and more complex regression analysis

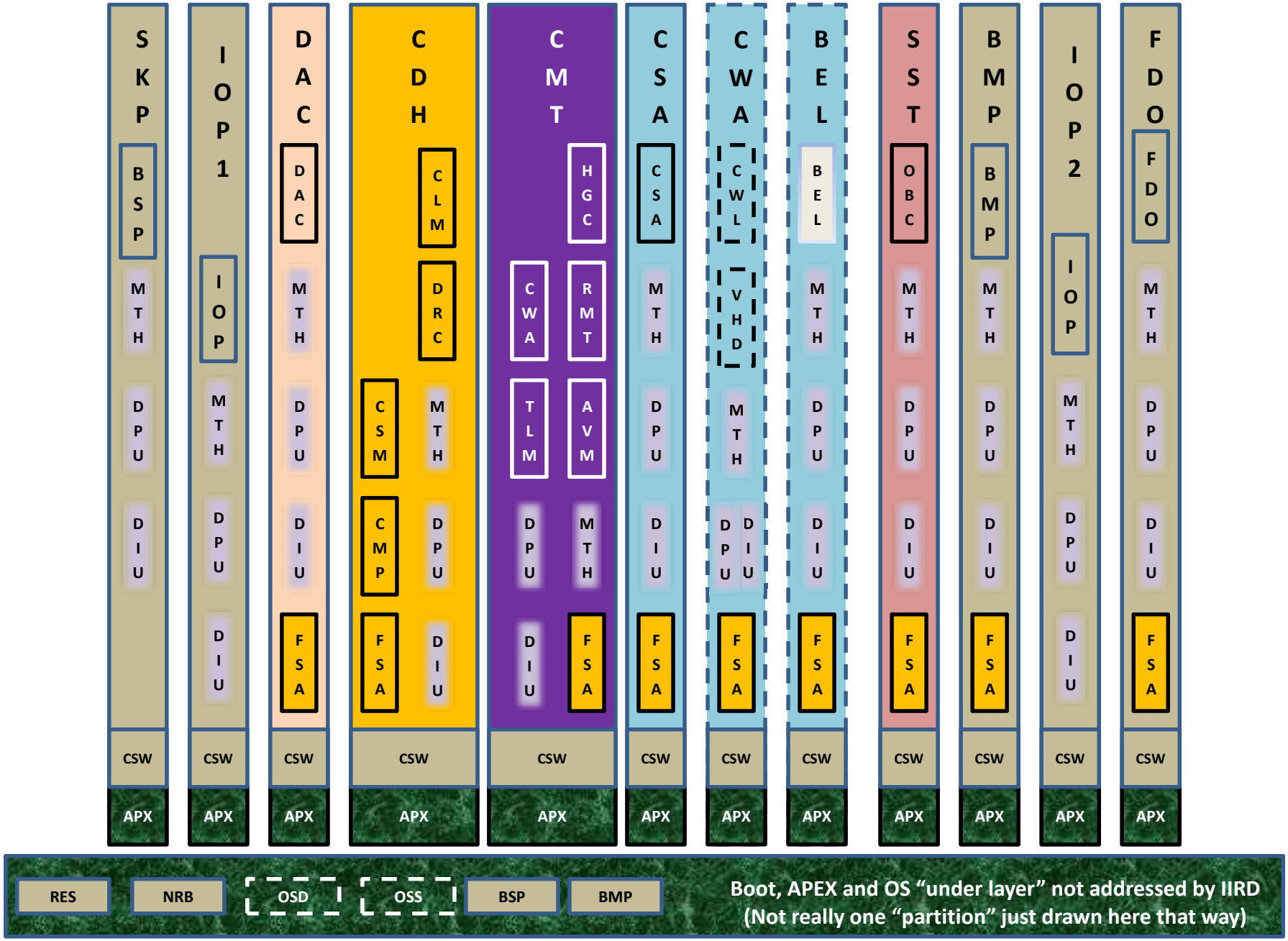


FCM FSW Partitions and Domains



DCM FSW Partitions

CWA & B EL inconsistently represented



Software Rqmnts Spec. Shall Include - SWE-109

	Document Element		Document Element
😐	System overview.	😐	Software quality characteristics.
🙂	CSCI requirements:	😐	Design and implementation constraints.
🙂	Functional requirements.	😐	Personnel-related requirements.
😐	Required states and modes.	😐	Training-related requirements.
🙂	External interface requirements.	😐	Logistics-related requirements.
😐	Internal interface requirements.	😐	Packaging requirements.
😐	Internal data requirements.	😐	Precedence and criticality of requirements.
🙂	Adaptation requirements	😐	Qualification provisions
😐	Safety requirements.	😐	Bidirectional requirements traceability.
😐	Performance and timing requirements.	🙂	Requirements partitioning for phased delivery.
🙂	Security and privacy requirements.	🙂	Testing requirements that drive software design decisions
🙂	Environment requirements	😐	Supporting requirements rationale
🙂 ☹	Computer resource Rqmnts (🙂 processor Rqmnts , ☹ timing Rqmnts)		

Software Requirements Specification Shall Include - SWE-109 (Continued)

The application of ARINC 653 architecture to FSW development provides:

- potential SRS improvements in some areas,
- and potential CPU utilization issues increasing:
  - the complexity of specifying computer resource requirements and
  - increasing overall demands on the computer's resources

Software Design Description shall include: [SWE-111]

	Document Element		Document Element
😐	CSCI-wide design decisions/trade decisions.	☺	Concept of execution, including data flow, control flow, and timing.
☺	CSCI architectural design.	😐	Requirements, design and code traceability.
☺	CSCI decomposition and interrelationship between components:	☹	CSCI's planned utilization of computer hardware resources.
😐	Description of how the software item satisfies the software requirements, including algorithms, data structures, and functional decomposition.	😐	Rationale for software item design decisions/trade decisions including assumptions, limitations, safety and reliability related items/concerns or constraints in design documentation.
☺	Software item I/O description.	☺	Interface design.
☺	Static/architectural relationship of the software units.		

The application of ARINC 653 architecture to FSW development provides potential SDD improvements, and potential CPU utilization issues increasing the complexity of specifying CPU utilization of computer resources and increasing overall demands on the computer's resources.

Interface Design Description shall include: [SWE-112]

	Document Element		Document Element
😐	Priority assigned to the interface by the interfacing entity(ies).	😐	Specification of protocols the interfacing entity(ies) will use for the interface.
☺	Type of interface to be implemented.	😐	Other specifications, such as physical compatibility of the interfacing entity(ies).
😐	Specification of individual data elements that the interfacing entity(ies) will provide, store, send, access, and receive.	😐	Traceability from each interfacing entity to the system or CSCI requirements addressed by the entity's interface design, and traceability from each system or CSCI requirement to the interfacing entities that address it.
😐	Specification of individual data element assemblies that the interfacing entity(ies) will provide, store, send, access, and receive.	😐	Interface compatibility
☺	Specification of communication methods that the interfacing entity(ies) will use for the interface.	😐	Safety-related interface specifications and design features.

The application of ARINC 653 architecture to FSW development provides potential IDD improvements

Regression analysis and testing. (IEEE STD 1012)

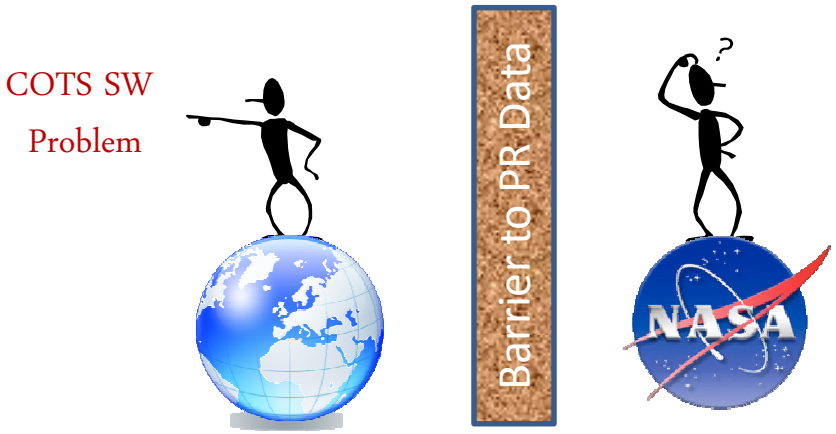
"Determine the extent of V&V analyses and tests that must be repeated when changes are made to any previously examined software products. Assess the nature of the change to determine potential ripple or side effects and impacts on other aspects of the system. Rerun test cases based on changes, error corrections, and impact assessment, to detect errors spawned by software modifications."

ARINC 653 partitions

- provide natural structure to scope regression testing,
- limit special analyses for establishing regression testing,
- increase relative certainty of regression scope
- limit the potential unintended consequences (simplify environment in which changes occur).

Software Problem Report

- NASA may represent a small fraction of users for a COTS SW package
- SW problems may be identified by any user
- NASA access to only NASA's PRs may be inadequate
  - Problem report data has development, verification and operation impacts
  - User data (PRs) would be sent to COTS developer
  - GIDEP program does not address software
- NASA access to necessary RTOS SW PR data for Orion has not been established
  - A known Project concern



Software Change Request/Problem Report shall contain: [SWE-113]

a. Identification of the software item.
b. Description of the problem or change to enable problem resolution or justification for and the nature of the change, including: assumptions/ constraints and change to correct software error.
c. Originator of Software Change Request/Problem Report and originator's assessment of priority/severity.
d. Description of the corrective action taken to resolve the reported problem or analysis and evaluation of the change or problem, changed software configuration item, schedules, cost, products, or test.
e. Life-cycle phase in which problem was discovered or in which change was requested.
f. Approval or disapproval of Software Change Request/Problem Report.
g. Verification of the implementation and release of modified system.
h. Date problem discovered.
i. Status of problem.
j. Identify any safety-related aspects/considerations/ impacts associated with the proposed change and/or identified problem.
k. Configuration of system and software when problem is identified (e.g., system/software configuration identifier or list of components and their versions).
l. Any workaround to the problem that can be used while a change is being developed or tested.

Architecture verification impacts

- Application of the partitioned approach to FSW should improve architecture verification at the integrated architecture level
  - Limited communication mechanisms (including specification of sampling ports, minor frame structure, major frame structure) between partitions simplifies aspects of architecture verification
    - Control flow and data flow analysis in support of requirements validation and architecture verification
- RTOS CSCI architecture artifacts may suffer from COTS opacity
  - COTS RTOS architecture often masked by special data restrictions or delivered executable file formats
    - Trade secret,
    - NDA,
    - Etc.
  - Orion is not seeking verification data package
    - Problems in APEX/OS may be found during integrated software testing

Requirements validation impacts

- Development of valid requirements for COTS RTOS is a significant challenge
  - "Testable" and "Complete" requirements for COTS RTOS CSCI are particularly difficult
  - Orion project reluctant to spend time/money on COTS RTOS SW requirements development
  - NASA V&V of COTS CSCI only required by NPR 7150.2A to be to "the same level of confidence" as developed FSW
    - No single unique meaning to "the same level of confidence"

Requirements validation impacts (Cont)

- Developed FSW CSCIs likely to have identical requirements in multiple partitions which will drive separate V&V
  - Overarching requirements: command handling/validation, partition initialization and health & status/fault reporting
- HW abstraction level will alter or limit computer resource requirements under SWE 109
- FSW performance and timing requirements under SWE 109 are likely to be more complicated from monolithic systems due to need to define (a) major frame, (b) minor frame and (C) CSCI rate groups for behaviors or requirements

FSW code verification impacts

- Significant code delivered COTS
  - Reduced test failure risk
  - COTS opacity makes failures more inscrutable
  - Non-test verification elements more difficult
  - SWE 135 “impossible” to apply to opaque executable COTS code
    - *“The project shall ensure that results from static analysis tool(s) are used in verifying and validating software code.”*
- Static code analysis should be easier for developed FSW CSCIs
- Unit testing of developed FSW CSCIs should be more accurate

Potential recertification impacts

- Should ease regression testing of Dev'd CSCIs
  - Limits potential scope to defined sampling ports/shared memory and effected partition
- Regression testing of COTS RTOS CSCI should be limited
  - Stable COTS code can be evaluated prior to purchase decision
- HW impacts should be reduced by abstraction layer
- CSCI Unit testing should be more accurate over unpartitioned (monolithic) system
- Documentation changes should be reduced

Other impacts (adverse impacts)

- DO 178, IEEE STD 1012 and NASA NPR 7150.2A all assume similar V&V requirements for both COTS products and developed SW
  - Achieving suitable V&V for COTS products is non-trivial
  - Opaque COTS product V&V presents additional issues

- Within IEEE STD 1012 - COTS SW is a special form of software reuse
  - IEEE STD 1012 Annex D (“*V&V of Reuse Software”*) and Appendix G *(“Optional V&V Tasks - Reusability analysis”) Reusability analysis. Verify that the artifacts (products) of the domain engineering process conform to project-defined purpose, format, and content (e.g., IEEE Std 1517-1999 [B11]). Verify that the domain models and domain architecture are correct, consistent, complete, accurate, and conform to the domain engineering plan. Analyze the asset (software item intended for reuse) to verify that the asset is consistent with the domain model and domain architecture.*

Other impacts (adverse impacts)

- NPR 7150.2A requires for COTS SW *"The software component is verified and validated to the same level of confidence as would be required of the developed software component."*
- DO-178 states that *"COTS software included in airborne systems or equipment should satisfy the objectives of this document"*
- Opacity impacts DO-178, NPR 7150.2 and IEEE 1012 V&V requirement conformance
- Vendors can provide DO 178 data. Not purchased for Orion.

Other impacts (adverse impacts)

- “Overhead” from 653
  - Significant overhead to perform essential ARINC 653 behaviors
    - IOPs
    - Sampling ports
  - Partitions sized for “worst case” needs results in unused time in each frame
    - Partitions must reach end in major frame or error flag generated
  - Static partition definitions within mission
    - Reallocation of frame resources (memory/time) not planned for Orion
  - Partitions can increase the effective SLOC
    - Identical processes in multiple partitions will need to be delivered as multiple copies of code/executable

Other impacts (adverse impacts)

- COTS OS opacity
  - IEEE STD 1012 and NASA NPR 7150.2A both assume similar V&V requirements for COTS products and developed SW
  - Achieving suitable V&V for COTS products is non-trivial
    - IEEE STD 1012 Annex D (“*V&V of Reuse Software”*) and Appendix G *(“Optional V&V Tasks - Reusability analysis”)*
      - *Reusability analysis.* Verify that the artifacts (products) of the domain engineering process conform to project-defined purpose, format, and content (e.g., IEEE Std 1517-1999 [B11]). Verify that the domain models and domain architecture are correct, consistent, complete, accurate, and conform to the domain engineering plan. Analyze the asset (software item intended for reuse) to verify that the asset is consistent with the domain model and domain architecture.
  - Need to establish a mechanism to receive SW Problem Report’s for COTS product (from non-NASA users)

Orion IV&V Experience

- Orion IV&V has identified issues with the representation of the COTS product within the NASA CSCI documentation
  - CX_ORION- TIM – 3096 accepted by Project
    - Project represented a piece of the COTS RTOS as meeting all RTOS requirements
  - CX_ORION- TIM – 3097 accurate but rejected by Project
    - Project represented COTS RTOS as compliant with standard. Project/NASA tailoring of COTS product results in non compliant installation (unused standard RTOS features are not supported by this installation).
    - RTOS CSCI is not the same thing as the RTOS COTS package.

Orion IV&V Experience (cont)

- The definition of the RTOS CSCI SRS is incomplete
  - CX_ORION- TIM – 3100 – Anticipated system specific functions not identified in CSCI SRS.
  - CX_ORION- TIM – 3613 No requirements to terminate to a safe known state.
  - CX_ORION- TIM – 3612 No requirements to initialize to a safe known state
  - Additional necessary requirements are known to be omitted from the APEX OS CSCI and PITS issues are being developed.
    - E.g. *SWE 134 subpart k – software provides error handling of safety-critical functions.*

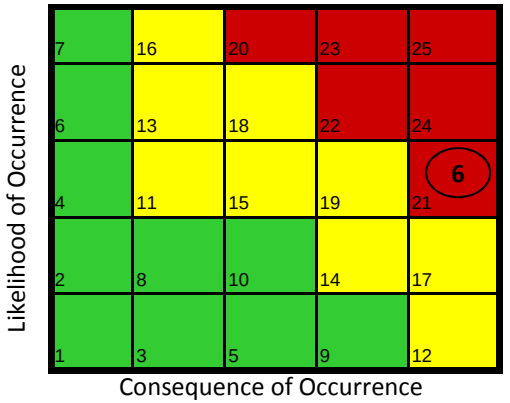
Orion IV&V Experience (cont)

- IV&V Observation – Orion's # of partitions cannot be justified by FDIR/Risk.
  - Developmental reasons for a large # of partitions exist.
  - # of partitions has an impact of processor utilization
  - Orion processor utilization @ PDR > MMP target
- IV&V Observation – Orion partitions cannot be justified by performance and timing rqmts.
  - Orion's CSCIs have NO performance & timing rqmnts.
  - Restructuring architecture for new timing rqmnts remains possible. Cost/schedule impact.

Risk 6 – Project

Update

Risk Title	COTS and Legacy Code In FSW
Risk Statement	***Given that*** the Orion flight software (FSW) incorporates a significant degree of Commercial Off The Shelf (COTS) and reused/legacy code ***there is*** a significant potential for this code to receive insufficient, incomplete or inappropriate review.
Context	Honeywell has brought a significant amount of legacy code to Orion. Orion FSW includes significant COTS elements directly or by incorporation (e.g. APEX). Legacy and COTS code was not developed using NASA-specific processes with associated artifacts. **Update: Orion has rejected external analysis of Green Hills APEX OSS OSD.** Impacts of COTS and legacy code on IV&V process assumptions need to be considered by management. Special NDA agreements have been required to support reviews or discussions of Honeywell elements. Current practices stifle sharing, openness, and teamwork . Potential impact to the identification and resolution of FSW issues and concerns. COTS and Legacy code decrease code access and process transparency, while increasing the potential for inappropriate code. The reuse of COTS or legacy code appears to increase the risk of flying dead code. See separate discussion of dead code. COTS and legacy code decrease costs and limit development risk
Closure Criteria	Establishing a plan for V&V of the COTS and legacy SW that conforms to NPR 7150.2 and IV&V expectations.



Updated
Board Approval Required.

"Context" updated.
Consequence unchanged.
Likelihood increased to "3"

Consequence	Rationale	Likelihood	Rationale
5	A significant FSW defect could result in LOC if V&V errors are also present and allow the defect to remain undetected. Detection of the FSW defect may be masked by the incomplete development of the associated requirements.	3	Orion project has rejected external analysis of APEX OSS and OSD. Controls exist with some uncertainties. $\approx 10^{-3}$ Controls include existing user base's experience with the products, anticipated incorporation of APEX OSS OSD into verification activities. Uncertainties include absence of Validation and no identified means for receiving information about performance errors identified by other users.

Conclusion

Commercially available ARINC 653 systems provide a time and space partitioned real time operating system with:

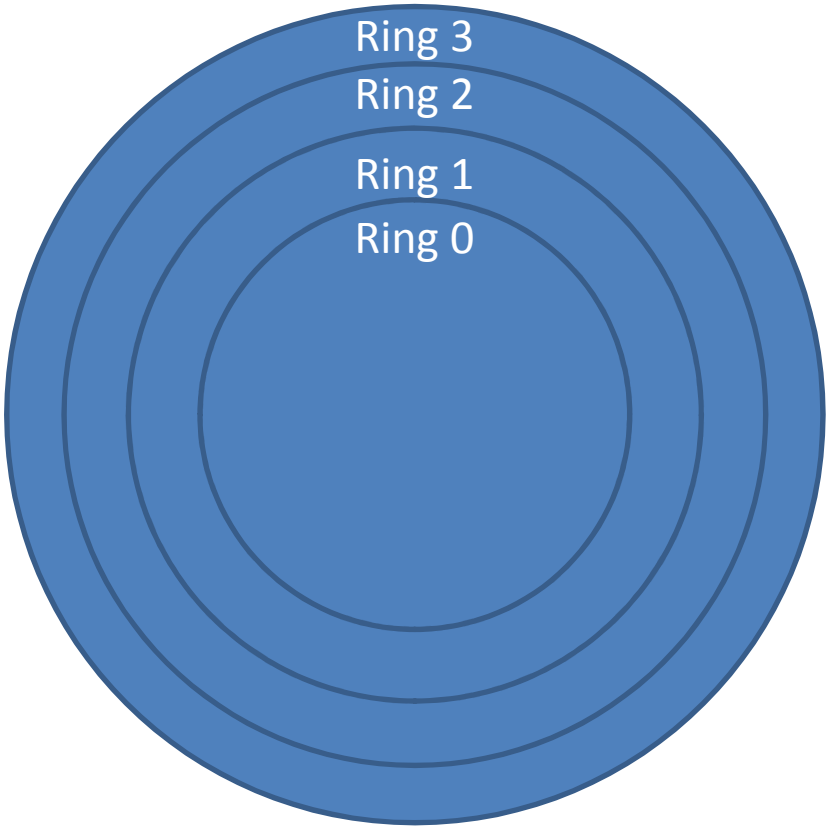
- Support for applications with differing levels of SW criticality.
- Low/No development risk (cost and schedule)
- Potential DO 178 compliant SW development
- Prior acceptance in hazardous applications
  - non-zero customer base,
  - not all ARINC 653 APEX OS are DO 178B Class A
  - unlikely to be developed to NASA stds

Commercially available ARINC 653 systems:

- Represent unique validation and verification challenges
  - Challenges that must be addressed in V&V planning
- Can adversely impact CPU utilization
- May have DO-178 artifacts for separate purchase.

Just In Case / Backup

Metaphor – ARINC 653 is similar to x86 “Protected Mode”



- ARINC 653 is similar to early x86 Protected Mode.
  - “Protected Mode” allowed for multiple levels of trust for an application within a multitasking environment.
  - ARINC 653 allows for multiple levels of SW criticality within a single system.

DO-178 Failure Condition Categorization

Category	Definition	Objectives	Obj with Indepen
Catastrophic	Failure Conditions which would prevent continued safe flight and landing	66	25
Hazardous/Severe-Major	Failure conditions which would reduce the capability of the aircraft or the ability of the crew to cope with adverse operating conditions to the extent that there would be: • Large reductions in safety margins or functional capabilities • Physical distress or higher work load (impacting crew operations reliability) • Adverse effects on occupants including serious or fatal injuries to a small number	65	14
Major	Failure conditions which would reduce the capability of the aircraft or the ability of the crew to cope with adverse operating conditions to the extent that there would be: • Significant reductions in safety margins or functional capabilities • Significant increase in crew workload or conditions impairing crew efficiency • Discomfort to occupants possibly including injuries	57	2
Minor	Failure conditions which would not significantly reduce aircraft safety, and which would involve crew actions that are well within their capabilities.	28	2
No Effect	Failure conditions which do not affect the operational capability of the aircraft or increased crew workload.	0	0

Partition Sketch Key

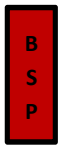
Partition – Long thin rectangle

CSCI Color Code

ECLS
EPS
SST
CMT
VMG
DAC
Core
ADL
SMG
C&DH
UTL
TMG
Control
APEX ,OS & Boot
Guide
Navi

FSA Domain – implied by FSM IIRD

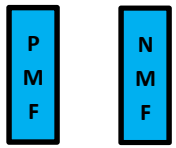
FSA Domain – possible with soft border



Domain – known with hard border

Domain – possible with soft border

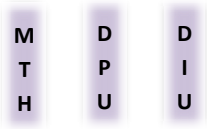
Unmapped NMF Domain



Unmapped FSA Domain

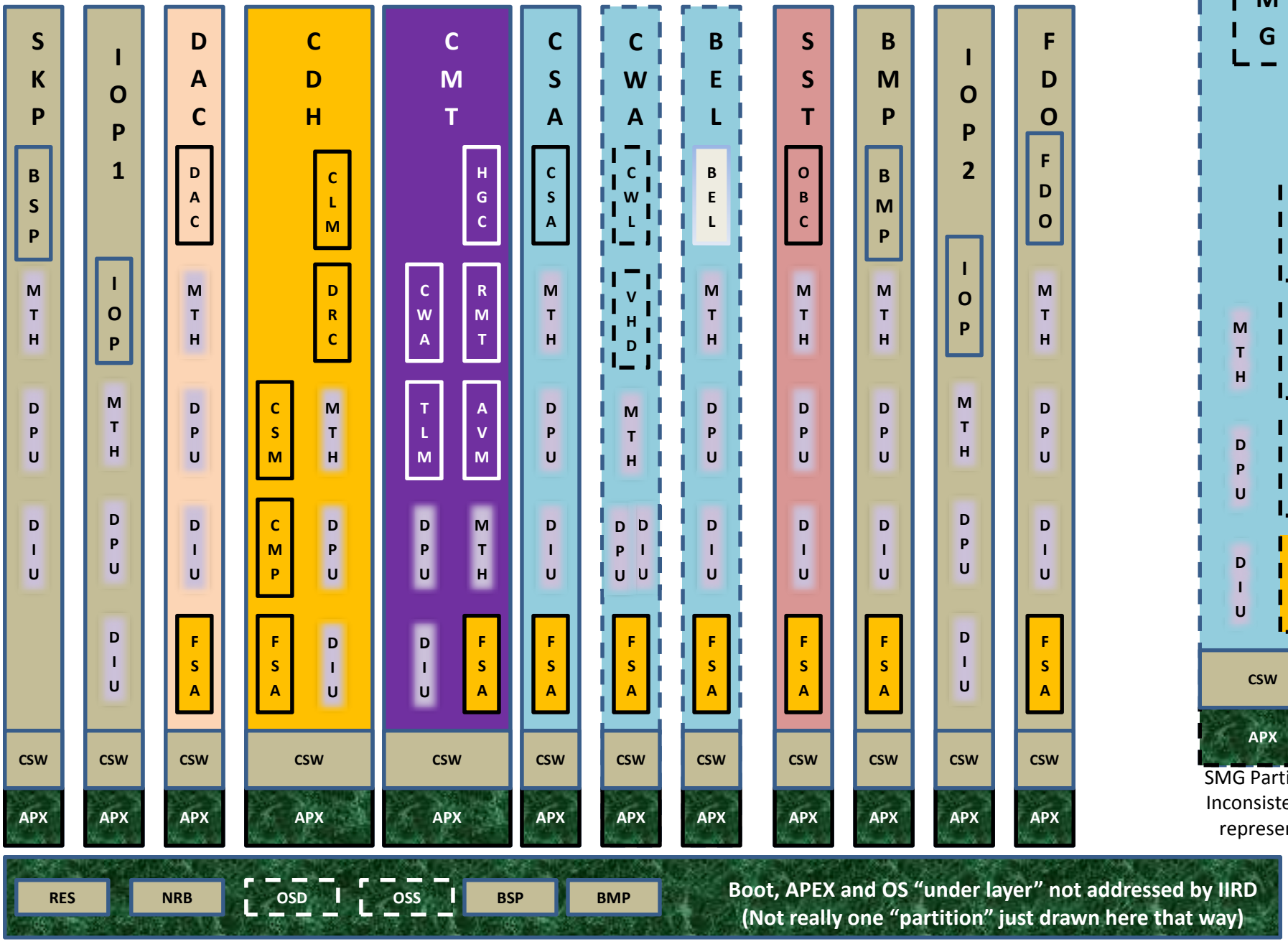
CCM File System

Unmapped UTL Domains. The application of UTL CSCI Domains is not tracked



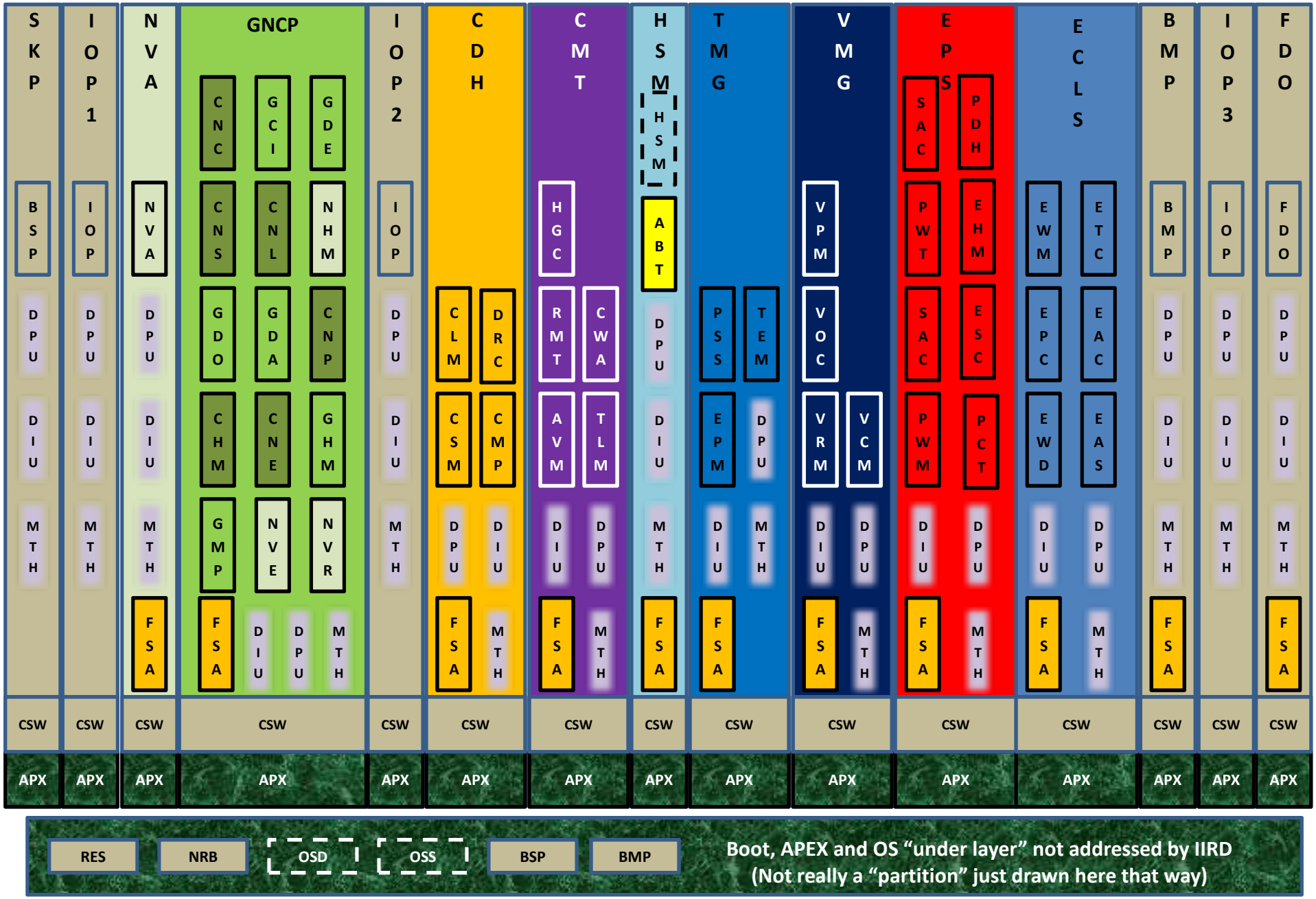
DCM FSW Partitions

CWA & B EL inconsistently represented

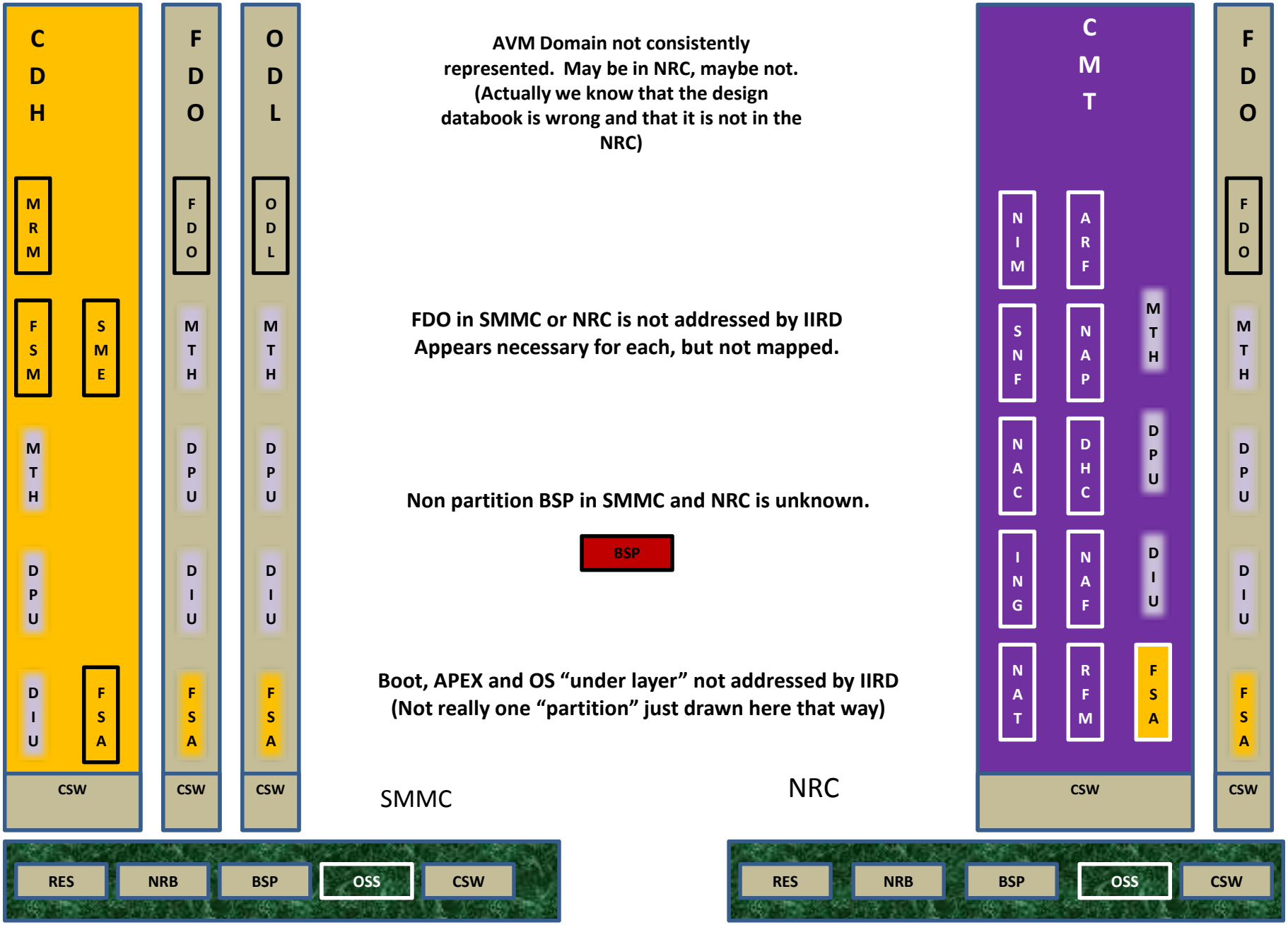


SMG Partition Inconsistently represented

FCM FSW Partitions and Domains



CCM FSW Non - Partitions



Comparison of Doc. Structure to SW Structure

FSW Code

29 Major Partitions

- 18 Unique
  - SKP, IOP (multiple versions), NVA, GNCP, CDH, CMT, HSM, TMG, VMG, EPS, ECLSS, BMP, FDO, DAC, CSA, CWA, BEL, and SST. SMG inconsistent.

BEL Partition UNDOCUMENTED

15 FCM Partitions
SKP, IOP1, NVA, GNCP, IOP2, CDH, CMT, HSM, TMG, VMG, EPS, ECLS, BMP, IOP3, FDO

12 DCM Partitions
SKP, IOP1, DAC, CDH, CMT, CSA, CWA, BEL, SST, BMP, IOP2, and FDO. SMG inconsistent.

2 CCM "Non-Partitions"
- SMMC and NRC

8 Under layer Domains

- RES, NRB, APX, OSD, OSS, CSW, BSP , BMP

9 "Unmapped" Domains

HSM, VHD and CWL of SMG
BEL?
NMF of BMP does not appear in IIRD but appears in Core SDD
FSA of CDH does not appear in IIRD but appears in SRS
MTH, DPU, DIU of Comm Util

BSP and BMP Appear As Under layer and Partition Domain.

Document Structure

16 CSCI Documents

ADL, APX, Core, CDH, CMT, CNT, DAC, ECLS, EPS, GID, NAV, SMG, SST, TMG, UTL and VMG

BEL Domains UNDOCUMENTED

11 FCM CSCI
- ADL, Core, CDH, CMT, CNT, ECLS, EPS, GID, NAV, TMG, VMG

6 DCM CSCI
- Core, CDH, CMT, DAC, SMG, SST

5 CCM CSCIs
- CMT, CDH, ODL, FDO, BSP(?)

2 Under layer CSCI

- APX and Core

2 "Unmapped" Domains

FSA of CDH
NMF of BMP

1 "Unmapped" CSCI

UTL CSCI

VPU omitted for clarity

Additional DO 178 Information

- Planning Output
  - Plan for Software Aspects of Certification
  - Software Development Plan
  - Software Verification Plan
  - Software Configuration Management Plan
  - Software Quality Assurance Plan
  - System Requirements
  - Software Requirements Standards*
  - Software Design Standards*
  - Software Code Standards*

Not required for DO 178 Level “D”

Additional DO 178 Information

- Development Output
  - Software Requirements Data
  - Software Design Description
  - Source Code
  - Executable Object Code
  - Trace from system requirements to source code typically required

Not required for DO 178 Level “D”

Additional DO 178 Information

- Verification Output
  - Software Verification Cases and Procedures
  - Software verification Results
    - Review of all requirements, design and code
    - Code coverage analysis
  - Trace of requirements to tests to results

Not required for DO 178 Level “D”

Additional DO 178 Information

- Configuration Management
  - Software Configuration Index
  - Software Lifecycle Environment Configuration Index

Not required for DO 178 Level “D”

Additional DO 178 Information

- Quality Assurance
  - Software Quality Assurance Records
  - Software Conformity Review
  - Software Accomplishment Summary

Not required for DO 178 Level “D”

Additional DO 178 Information

- Other
  - Tools generating embedded code must be qualified as development tools (with the same constraints as the embedded code).
  - Tools used to verify code must be qualified as verification tools (primarily through testing)

Not required for DO 178 Level “D”

DO 178 Information

- Application
  - DO-178 is instructional.
    - No “shalls”
  - NPR 7150.2 is mandatory.
    - SWEs and “shalls” are the central focus
- Development
  - DO-178 is consensus based
  - NPR 7150.2 is directed by management with defined management level waiver processes.
- Author
  - DO-178 “association of aeronautical organizations”
  - NPR 7150.2 NASA Office of the Chief Engineer

Not required for DO 178 Level “D”

Comparison of DO-178 to NASA std's

- Approach to COTS
  - DO-178.
    - *COTS software included in airborne systems or equipment should satisfy the objectives of this document*
  - NPR 7150.2A
    - *When software components use COTS applications (e.g., spreadsheet programs, database programs) within a NASA system/subsystem application, the software components need to be assessed and classified as part of the software subsystem in which they reside.*

ARINC 653 Selection - Impact to Baseline Documentation

- ARINC 653 partitions provide one natural division for CSCI definitions
  - Partitions maybe defined late in architecture development or adjusted after initial development (potential to limit the value of the 653 architecture)
- Alignment of partitions to CSCIs supports effective documentation development, regression testing, V&V, structured integration, and eases development.
  - The time/space partitioned CSCI maps effectively into required documents
- Orion does not follow the logical approach resulting in undocumented CSCI to CSCI interfaces and more complex regression analysis

Software Requirements Specification Shall Include - SWE-109

a. System overview.

b. CSCI requirements:

1) Functional requirements.
2) Required states and modes.
3) External interface requirements.
4) Internal interface requirements.
5) Internal data requirements.
6) Adaptation requirements (data used to adapt a program to a given installation site or to given conditions in its operational environment).
7) Safety requirements.
8) Performance and timing requirements.
9) Security and privacy requirements.
10) Environment requirements.
11) Computer resource requirements:
    a) Computer hardware resource requirements, including utilization requirements.
    b) Computer software requirements.
    c) Computer communications requirements.
12) Software quality characteristics.
13) Design and implementation constraints.
14) Personnel-related requirements.
15) Training-related requirements.
16) Logistics-related requirements.
17) Packaging requirements.
18) Precedence and criticality of requirements.

c. Qualification provisions (e.g., demonstration, test, analysis, inspection).

d. Bidirectional requirements traceability.

e. Requirements partitioning for phased delivery.

f. Testing requirements that drive software design decisions (e.g., special system level timing requirements/checkpoint restart).

g. Supporting requirements rationale.

Software Design Description shall include: [SWE-111]

a. CSCI-wide design decisions/trade decisions.
b. CSCI architectural design.
c. CSCI decomposition and interrelationship between components:
    1) CSCI components:
        a) Description of how the software item satisfies the software requirements, including algorithms, data structures, and functional decomposition.
        b) Software item I/O description.
        c) Static/architectural relationship of the software units.
        d) Concept of execution, including data flow, control flow, and timing.
        e) Requirements, design and code traceability.
        f) CSCI's planned utilization of computer hardware resources.
    2) Rationale for software item design decisions/trade decisions including assumptions, limitations, safety and reliability related items/concerns or constraints in design documentation.
    3) Interface design.

Interface Design Description shall include: [SWE-112]

a. Priority assigned to the interface by the interfacing entity(ies).
b. Type of interface (e.g., real-time data transfer, storage-and-retrieval of data) to be implemented.
c. Specification of individual data elements (e.g., format and data content, including bit-level descriptions of data interface) that the interfacing entity(ies) will provide, store, send, access, and receive.
d. Specification of individual data element assemblies (e.g., records, arrays, files, reports) that the interfacing entity(ies) will provide, store, send, access, and receive.
e. Specification of communication methods that the interfacing entity(ies) will use for the interface.
f. Specification of protocols the interfacing entity(ies) will use for the interface.
g. Other specifications, such as physical compatibility of the interfacing entity(ies).
h. Traceability from each interfacing entity to the system or CSCI requirements addressed by the entity's interface design, and traceability from each system or CSCI requirement to the interfacing entities that address it.
i. Interface compatibility.
j. Safety-related interface specifications and design features.

Software Data Dictionary shall include: [SWE-110]

a. Channelization data (e.g., bus mapping, vehicle wiring mapping, hardware channelization).
b. Input/Output (I/O) variables.
c. Rate group data.
d. Raw and calibrated sensor data.
e. Telemetry format/layout and data.
f. Data recorder format/layout and data.
g. Command definition (e.g., onboard, ground, test specific).
h. Effecter command information.
i. Operational limits (e.g., maximum/minimum values, launch commit criteria information).

IEEE STD 1012 SRS, IDD and IRS

- **Interface Design Document (IDD):** Documentation that describes the architecture and design interfaces between system and components. These descriptions include control algorithms, protocols, data contents and formats, and performance.
- **Interface Requirements Specification (IRS):** Documentation that specifies requirements for interfaces between systems and components. These requirements include constraints on formats and timing.
- **Software Requirements Specification (SRS):** Documentation of the essential requirements (functions, performance, design constraints, and attributes) of the software and its external interfaces.

The Software Test Plan shall include: [SWE-104]

a. Test levels (separate test effort that has its own documentation and resources, e.g., component, integration, and system testing).
b. Test types:
    1) Unit testing.
    2) Software integration testing.
    3) Systems integration testing.
    4) End-to-end testing.
    5) Acceptance testing.
    6) Regression testing.
c. Test classes (designated grouping of test cases).
d. General test conditions.
e. Test progression.
f. Data recording, reduction, and analysis.
g. Test coverage (breadth and depth) or other methods for ensuring sufficiency of testing.
h. Planned tests, including items and their identifiers.
i. Test schedules.
j. Requirements traceability (or verification matrix).
k. Qualification testing environment, site, personnel, and participating organizations.