



**MULTI-CORE PROCESSORS (MCP)
AIRWORTHINESS PLAYBOOK
(MAP)**

**CRADA No. 0630
SRD-SWAW-AWP-MCP_MAP-001**

**U.S. Army Combat Capabilities Development Command
Aviation & Missile Center (DEVCOM AvMC)**

And

Rockwell Collins Inc. a part of Collins Aerospace

**DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited.
PR2025237**



Revision History

Version	Date	Author	Remarks
-	22 August 2025		Initial Release
A	4 December 2025		DEVCOM AvMC Public Release

MULTI-CORE PROCESSORS (MCP) AIRWORTHINESS PLAYBOOK (MAP)

8/22/2025

CRADA No. 0630

Originating Entity Names:

U.S. Army Combat Capabilities Development Command Aviation & Missile Center (DEVCOM AvMC)

and

Rockwell Collins Inc. a part of Collins Aerospace

Originating Entity Addresses:

U.S. Army Combat Capabilities Development Command Aviation & Missile Center (DEVCOM AvMC)

5400 Fowler Rd

Redstone Arsenal, AL 35898

and

Rockwell Collins Inc.

400 Collins Rd NE

Cedar Rapids, IA 52498

Table of Contents

1	Introduction.....	9
1.1	Purpose	9
1.2	Open System Approach Considerations	11
1.3	Document Structure	12
1.3.1	Development Phases	12
1.4	References	12
2	Definitions.....	13
2.1	Terms	13
2.2	Lifecycle Data Item	14
2.3	Acronyms	16
3	Roles & Responsibilities.....	18
4	Developing the Computing System.....	19
4.1	Plan the MCP Platform	20
4.1.1	Preliminary MCP Platform Architecture.....	21
4.1.2	Preliminary Safety Activities.....	21
4.1.3	MCP Platform Plan.....	26
4.1.4	Microbenchmarks	28
4.2	Design the MCP Platform	28
4.2.1	MCP Platform Requirements.....	29
4.2.2	MCP Platform Design Data.....	30
4.2.3	Preliminary Multi-Core Interface Analysis [Design]	30
4.2.4	MCP Platform Requirement and Design Validation	30
4.3	Build the MCP Platform	31
4.3.1	Prototype MCP Platform.....	31
4.3.2	Test Software.....	31
4.4	Verify the MCP Platform	33
4.4.1	Verification Procedures and Results	33
4.4.2	Final Safety, Partitioning, and Interference Analyses	33
4.5	Output	34
4.5.1	Integrator Mitigation Methodology.....	35
4.5.2	Integrator ‘Configuration Definition’ Methodology	37
4.5.3	Accomplishment Summary	38

- 4.5.4 MCP Platform User Guide 39
- 5 Hosted Application System 40
 - 5.1 Plan the Hosted Application 41
 - 5.1.1 Aircraft/Integrated System Needs for the Hosted Application..... 41
 - 5.1.2 Preliminary Hosted Application Design Data..... 41
 - 5.2 Design the Hosted Application 41
 - 5.2.1 Mature Hosted Application Design Data..... 42
 - 5.3 Build the Hosted Application 42
 - 5.3.1 Implement the Hosted Application Software 42
 - 5.3.2 Integrate the Hosted Application Software..... 43
 - 5.4 Verify the Hosted Application 43
 - 5.4.1 Hosted Application Verification..... 43
 - 5.4.2 HA System Verification 43
 - 5.4.3 Hosted Application WCET 43
 - 5.4.4 Considerations for the Allocation of Shared Resources..... 44
 - 5.5 Output 44
 - 5.5.1 MCP Platform User Guide 44
 - 5.5.2 Software Accomplishment Summary..... 44
- 6 Integrated System Development..... 45
 - 6.1 Plan the Integrated System 46
 - 6.1.1 Preliminary Integrated System Architecture..... 46
 - 6.2 Design the Integrated System 47
 - 6.2.1 Integrated System Requirements & Design Data..... 47
 - 6.2.2 Preliminary Integrated System Safety Assessment 47
 - 6.3 Plan the Configuration Data Item 47
 - 6.3.1 Defining the Configuration Data Item Build Process 48
 - 6.3.2 Defining the Configuration Data Item Development Process 48
 - 6.3.3 Plan for Integrated MCP Aspects of Certification (PIMAC)..... 48
 - 6.4 Design the Configuration Data Item 49
 - 6.4.1 Interference Analysis and Integrator Mitigation Methodology..... 50
 - 6.4.2 Configuration Data Item Requirement and Design Data..... 50
 - 6.5 Build the Configuration Data Item 50
 - 6.5.1 Executing Integrator Mitigation Methodology..... 50

- 6.5.2 Executing the Configuration Data Item Build Process 50
- 6.5.3 Configuration Data Item Build Example..... 51
- 6.6 Verify the Configuration Data Item 52
 - 6.6.1 Verifying the Configuration Data Item..... 52
- 6.7 Build the Integrated System 52
 - 6.7.1 Integrating the Computing System and Hosted Application Systems..... 53
- 6.8 Verify the Integrated System 53
 - 6.8.1 Requirement and Robustness Verification 53
 - 6.8.2 Integrated System Safety Assessment 54
 - 6.8.3 Integrated System Determinism Analysis..... 54
- 6.9 Output 55
 - 6.9.1 Accomplishment Summary 56
- 7 Aircraft Development..... 57
- 8 MCP Platform – Progress Oversight 58
 - 8.1 Detailed Design Reviews 58
 - 8.2 Verification Readiness Reviews 59
 - 8.3 Completion Reviews 60
 - 8.4 Quality and Process Assurance 61
- 9 MCP Tools 62
 - 9.1 Types of Tools 62
 - 9.2 Procurement or Development of Tools 62
 - 9.3 Tool Qualification 62
 - 9.4 Tool Configuration Management & Maintenance 62
- 10 Multi-Core and IMA Considerations..... 64
- 11 Advisory Circular 20-193 Mapping 66
- Appendix A Integrated System Development Example 83
 - A.1 Integrated System Design and Use Cases..... 83
 - A.2 MCP Platform Interference Analysis and User Guide 83
 - A.3 Integrated System Configuration Data Item..... 85
 - A.4 Integrated System Change - Add new Hosted Application 89
 - A.5 Integrated System Change - Modified Integrated System..... 91

Table of Tables

Table 1 – MCP Guidance Term Definitions	13
Table 2 – MCP Guidance Lifecycle Data Item Definitions.....	14
Table 3 – Acronyms	16
Table 4 – Roles and Responsibility	18
Table 5 – MCP Guidance Support for DO-297 Tasks.....	64
Table 6 – Advisory Circular 20-193 Mapping	67

Table of Figures

Figure 1 – Development V	10
Figure 2 – User Guide Flow	11
Figure 3 – Five Main Development Phases	12
Figure 4 – Boundary of Computing Equipment and MCP Platform.....	19
Figure 5 – Example of Computing System.....	20
Figure 6 – Safety Process Flow.....	22
Figure 7 – Example Interference Channel Diagram from FAA Tech Report	23
Figure 8 – Example Interference Channel Diagram of L3 Cache.....	25
Figure 9 – Example Test Software Configuration.....	32
Figure 10 – Generic Integrator Mitigation Methodology.....	35
Figure 11 – Generic Configuration Definition Methodology.....	38
Figure 12 – Example #1 Hosted Application System	40
Figure 13 – Example #2 Hosted Application System	40
Figure 14 – Relationship of Integration	45
Figure 15 – Example Integrated System	46
Figure 16 – Generic CDI Build Process.....	48
Figure 17 – Generic CDI Development Process.....	48
Figure 18 – Example of Multiple CDI Build Scenarios.....	52
Figure 19 – Integrated Systems Example: System Design, Preliminary.....	83
Figure 20 – Integrated Systems Example: Refinement of the Allocation	86
Figure 21 – Integrated Systems Example: Hosted App Execution Design, Preliminary.....	87
Figure 22 – Integrated Systems Example: System Design, Final	87
Figure 23 – Integrated Systems Example: Hosted App Execution Re-Design.....	88
Figure 24 – Integrated Systems Example: Hosted App Execution Re-Design 2.....	89
Figure 25 – Integrated Systems Example: System Design, New Hosted App	89
Figure 26 – Integrated Systems Example: Hosted App Execution with Hosted App 6	91
Figure 27 – Integrated Systems Example: System Design, New Function	91
Figure 28 – Integrated Systems Example: Modified Hosted Applications	92

1 Introduction

The aerospace industry and other real-time systems builders face a challenging transition from Single Core Processor (SCP) chips to Multi-Core Processor (MCP) chips. More than for SCP chips, Commercial, Off-The-Shelf MCP chips now exhibit significant potential for problematic on-chip hardware contention. Sharing on-chip hardware resources between cores can cause nondeterministic slowdowns to software execution times on MCP chips. This problematic potential raises questions about real-time determinism that can be addressed by utilizing this guidance. Specifically, traditional methods for calculating software Worst Case Execution Time (WCET), e.g., from software source code, assumed SCP chips and are not sufficient to reveal an MCP chip's nondeterministic slowdown potential. In the face of this transition from SCP to MCP, the aerospace industry needs to adapt how it certifies real-time software deployed to MCP chips.

This new challenge in deploying real-time software to MCPs is due to their Hardware Interference Channels (HIC). A HIC can be triggered when more than one software application accesses a shared hardware resource (either simultaneously or sequentially). The collection of HIC types and their impacts can be expected to vary in each chip, circuit board and system design. HIC effects on software slowdown can vary greatly.

FAA Advisory Circular (AC) 20-193 [1] and EASA Acceptable Mean of Compliance (AMC) 20-193 [2] have been recently released to provide an acceptable means of compliance (MOC) for aircraft systems utilizing MCP Platforms.

1.1 Purpose

The objective of this document is to provide guidance for what is needed to complete the certification of a System or Systems which implements an MCP Platform and reduce the risk of unintended contention discoveries late in the development. This guidance document is intended to supplement A(M)C 20-193 by providing additional details on activities and lifecycle data to support the objectives defined in the acceptable MOC and focuses on MCP Platforms with the intent of satisfying Robust Time and/ or Resource Partitioning as described in A(M)C 20-193.

This guidance document recommends utilizing the following existing industry practices as a baseline: ARP 4754 [3], ARP 4761 [4], DO-297 [5], DO-254 [6], and DO-178 [7]. The goal is to leverage existing development processes sourced from these types of industry standards and ensure MCP Platform aspects are thoroughly considered. Therefore, this guidance document does not duplicate guidance or objectives from these existing standards mentioned above. It is understood that a development could deviate from these standards or utilize other similar standards. The details in this guidance should be tailored to support those deviations.

As depicted in Figure 1, these good engineering practices define objectives to create and validate requirements, to design, implement, and integrate per requirements, and to verify the requirements were satisfied. Suppliers usually have these process constructs already in place for the development of Systems, Equipment, Hardware, and Software. In Figure 1, the green 'MCP' blocks indicate there are MCP Platform considerations detailed in the corresponding sections. Using the traditional development "V" construct, this guidance document identifies MCP Platform considerations, for each role and for each phase, to ensure the lifecycle data contains the required data to satisfy the required objectives (Sections 4 – 7). The A(M)C 20-193 objectives are mapped to each lifecycle data item (Section 11).

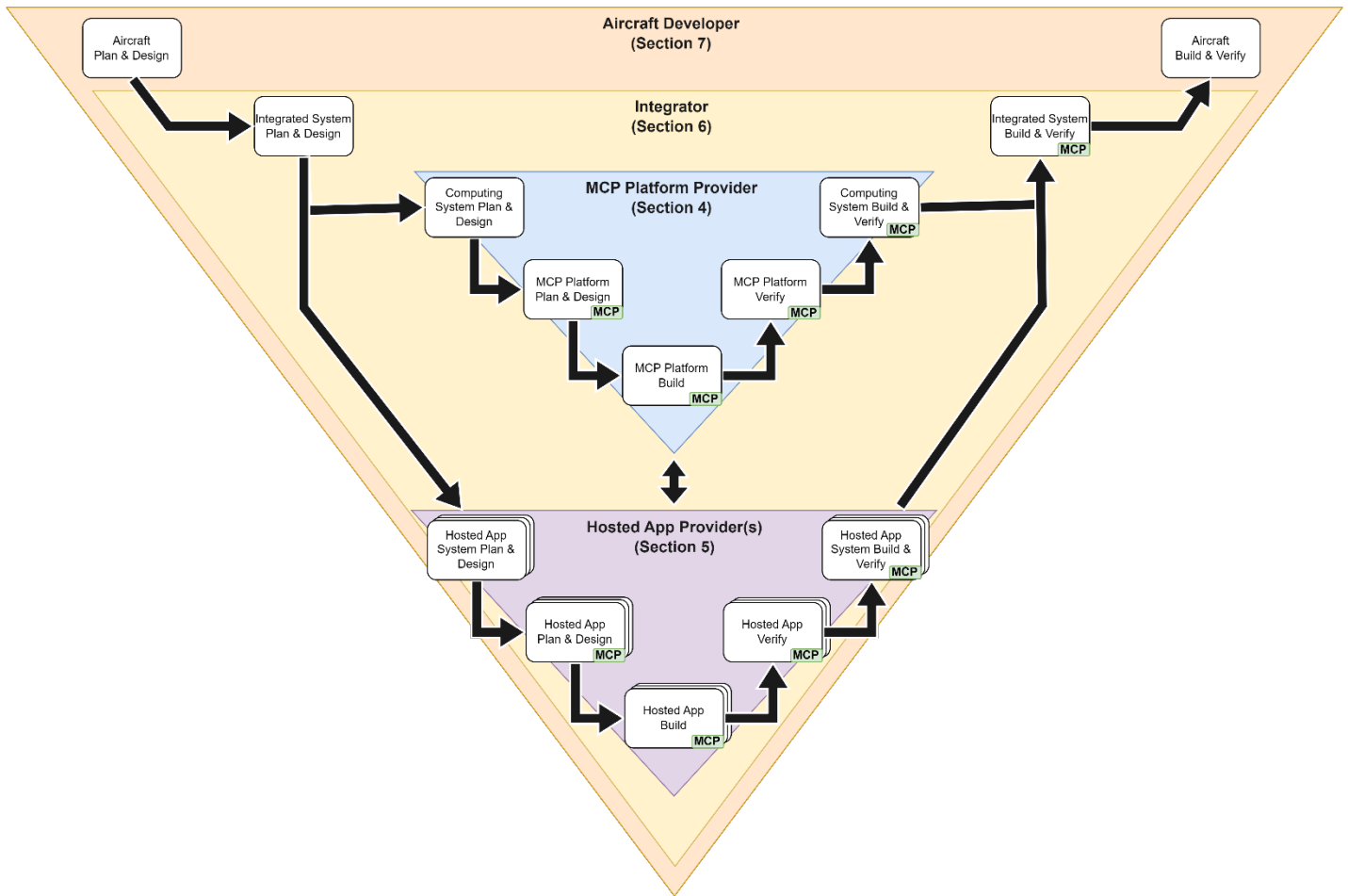


Figure 1 – Development V

This guidance document also identifies important oversight methods to reduce risk of late discoveries within the MCP development (Section 8), with the focus on early data reviews in the beginning of the MCP development ensuring the design and requirements are maturing.

Key considerations:

- 1) The traditional development V process is an iterative process flowing updates, errors, defects, and omissions back to the appropriate point in the process for correction. This requires a change management process in development using Problem Reports to track changes to requirements, design, implementation, or verification methods. As an example, a preliminary Interference Analysis may be executed multiple times as the requirements and design of the MCP Platform are modified during the development.
- 2) User guides are vehicles to provide critical data to the user of the component or equipment. This critical data must be consumed by the receiving role and appropriate requirements levied on the product. Example, MCP configuration constraints identified in the MCP Platform User Guide must be consumed and implemented correctly by the Integrator and Aircraft Developer. Figure 2 depicts a generic flow of a “User” (building the product) consuming the user’s guide of a component provided by the “Producer”.

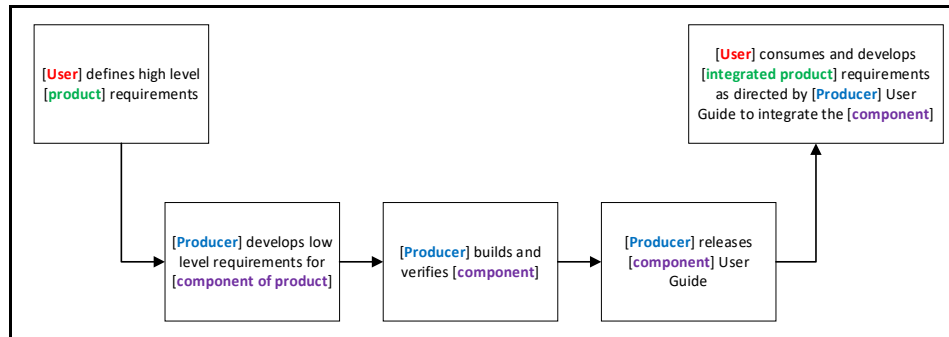


Figure 2 – User Guide Flow

- 3) Generic titles are used in this guidance document for the different lifecycle data items. It is not mandatory to use these titles as written and data could be packaged differently to meet the same intent. Planning documents should map the lifecycle data items and the industry standard objectives being met.
- 4) This guidance document defines the flow of design, to implementation, and to verification but is not prescribing a development methodology (e.g., waterfall vs agile). It is not implied that the MCP development must wait for Aircraft design and requirements. Example, the MCP Platform may be fully developed and is reused on a new Integrated System. In this example, the Integrated System requirements must still flow to MCP Platform Provider with a validation exercise to ensure the existing MCP Platform requirements are adequate.
- 5) While not depicted in Figure 1, it is understood that feedback loops between the roles (MCP Platform Provider, Hosted App Provider, Integrator, Aircraft Developer) are needed during the phases of development to ensure quality of the product, constrain aspects of the design or implementation, validate assumptions, and support completion of the activities.
- 6) This guidance document recommends early prototyping and analysis activities of the proposed MCP Platform to understand the performance characteristics and to determine if the MCP Platform is viable for System Use Cases.

1.2 Open System Approach Considerations

Systems that employ modular and open principles under an Open System Approach (OSA), such as the Modular Open System Approach (MOSA), likely incorporate MCP Platforms. This guidance document describes processes and lifecycle data that can be used to complete an MCP development and certification. This guidance document does not require application of OSA but is in alignment with the OSA principles where applicable. Modular aspects are covered in Section 10.

This guidance document is focused on a singular MCP Platform with multiple Hosted Applications for an Integrated System. However, in the OSA environment, it is anticipated that there are several MCP Platform Providers developing MCP Platform solutions for the same Integrated System. The same could be true for Hosted Applications as well. The Integrator and/or Aircraft Developer consumes all the data from the MCP Platform Provider (e.g., Interference Analyses, User Guides) and from the Hosted App Provider and maintain the integration and configuration to ensure the Integrated System functions as expected.

1.3 Document Structure

The following sections provide the reader with important details on the structure of the document.

1.3.1 Development Phases

Within Sections 4-6, the document is separated into five main phases (Figure 3) that define inputs, outputs, and activities to complete the development of the Multi-Core Processor and provide the needed lifecycle data to support Hosted App Providers and Integrators.



Figure 3 – Five Main Development Phases

Each phase defines the Inputs, Outputs, and the Activities.

- Inputs → Data need to start phase activities
- Inputs from → Role identifier to provide the Input data
- Activity → Development actions for the Role to perform
- Outputs → Data produced and under configuration control from executing the Activities
- Outputs to → Role identifier to receive the Output data

1.4 References

Latest revision unless noted below.

- [1] Advisory Circular 20-193, Use of Multi-Core Processors, dated 1/8/2024
- [2] Acceptable Mean of Compliance 20-193, Use of multi-core processors, dated 1/26/2022
- [3] ARP 4754, Guidelines for Development of Civil Aircraft and Systems
- [4] ARP 4761, Guidelines for Conducting the Safety Assessment Process on Civil Aircraft, Systems, and Equipment
- [5] DO-297, Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations
- [6] DO-254, Design Assurance Guidance for Airborne Electronic Hardware (AEH)
- [7] DO-178, Software Considerations in Airborne Systems and Equipment Certification
- [8] FAA Tech Report TC-16/51, Assurance of Multicore Processors in Airborne Systems, dated July 2017
- [9] TSO-C153a, Integrated Modular Avionics (IMA) Platform and Modules, dated 5/29/2019
- [10] TSO-214, Functional TSO Equipment using a TSO-C153a Authorized IMA Platform(s) and/or Modules, dated 7/29/2021
- [11] DO-330, Software Tool Qualification Considerations

2 Definitions

2.1 Terms

Table 1 – MCP Guidance Term Definitions

Term	Definition
Aircraft	Machine or vehicle capable of flight.
Aircraft Function	A capability of the aircraft that is provided by the hardware and software of the systems on the aircraft. (DO-297) Example: Display of aircraft heading to crew
Computing System	A system with the primary purpose of providing a computing function for hosted software applications. A computing system on its own does not provide an aircraft function.
Core Software	The operating system and support software that manage resources to provide an environment in which applications can execute. (DO-297) Core software is a necessary component of a platform, and it typically consists of one or more modules. A Real-time Operating System (RTOS) is considered Core Software.
Critical Configuration Settings	Those configuration settings that the applicant has determined to be necessary for the deterministic execution of the software or any settings that, if inadvertently altered, could change the behavior of the processor so as to cause the hosted software to no longer comply with its requirements. (AMC 20-193)
Hardware Interference Channel (HIC)	A shared hardware property within the platform that may cause interference between software applications.
Hosted Application	The software hosted on one or more cores of the MCP Platform.
Hosted Application (HA) System	A system that utilizes at least one Hosted Application executing on the shared resources of a Computing System along with other applicable equipment to perform one or more aircraft functions.
IMA System	Consists of the IMA platform(s) and the hosted applications.
Integrated Modular Avionics (IMA) Platform	A module or group of modules, including core software, that manages resources in a manner sufficient to support at least one application. (DO-297)
Interference Channel	A platform property that may cause interference between software applications or tasks (AC 20-193)
Integrated System	The overall system consisting of the Computing System (shared resources), the Hosted Application System(s), and the allocation or configuration data to allow the systems to operate as intended.
Multi-Core Processor (MCP)	An AEH device that contains two or more processing cores. A core in an MCP is defined as a device that executes software. This includes virtual cores (e.g., in a simultaneous multithreading microarchitecture). An MCP is typically implemented in a device that may also include resources such as memory or peripheral controllers, internal memory, peripherals, and internal interconnects. (AC 20-193) The MCP is considered a shared resource.

Term	Definition
MCP Platform	<p>Consists of the MCP itself and, in many cases, the platform software, such as an operating system and/or software hypervisor, which provides the interface between the software applications and the MCP. (AC 20-193)</p> <p>The MCP Platform is the processing subsystem which includes the MCP device and support components (such as memory, peripheral devices, etc.) and core software.</p>
Robust partitioning	Both robust resource partitioning and robust time partitioning.
Robust resource partitioning	<p>Robust resource partitioning is achieved when:</p> <ul style="list-style-type: none"> • Software partitions cannot contaminate the storage areas for the code, IO, or data of other partitions. • Software partitions cannot consume more than their allocations of shared resources; and • Failures of hardware unique to a software partition cannot cause adverse effects on other software partitions. <p>(AC 20-193)</p>
Robust time partitioning	Robust time partitioning (on an MCP) is achieved when, as a result of mitigating the time interference between partitions hosted on different cores, no software partition consumes more than its allocation of execution time on the core(s) on which it executes, irrespective of whether partitions are executing on none of the other active cores or on all of the other active cores. (AC 20-193)
Shared Resource	Any shared object (processor, memory, software data, etc.) or component provided by the MCP Platform and used by a Hosted Application

2.2 Lifecycle Data Item

Table 2 – MCP Guidance Lifecycle Data Item Definitions

Lifecycle Data Item	Definition
Accomplishment Summary*	Summary lifecycle data item that shows and concludes the [domain] was implemented per the requirements and successfully verified while following a structured process. Data item also concludes compliance to industry guidance objectives as applicable.
Architecture Design Data*	Data describing the design of the [domain] used to support development of requirements.
Configuration Index*	Configuration details for the [domain] defining all required equipment (hardware and software) part identifiers and loadable software identifiers.
Development Plan*	This planning document defines the processes used to develop the [domain] which includes artifact development for requirements, for validation and verification data, and for configuration management data. This planning document defines the level of integration and the integration activities with other systems, equipment, hardware, and software aspects.
Interference Channel (IC) Diagrams	A model of the MCP Platform consisting of diagrams showing the chip architecture, HIC types, HIC causes, and HIC effects
Integrated System Determinism Analysis	Lifecycle data item that analyzes the integrated configuration of the Hosted Applications executing on the MCP Platform. This analysis justifies all Hosted Applications execute as intended per the allocated resources of the MCP Platform.

Lifecycle Data Item	Definition
Microbenchmark	Verification software to generate stress by triggering HIC on the MCP Platform. This software is also referred to as: “aberrant apps”, “red apps”, or “adversarial apps”.
Interference Analysis	Lifecycle data item that analyzes the MCP design and the interference channels per the methods defined in the MCP Platform Plan. This analysis justifies the mitigations or safety nets required in the MCP Platform, Computing System, and/or Integrated System to eliminate or reduce interference.
Multi-Core Processor (MCP) Platform Plan	<p>Planning document to describe the MCP Platform, define the methods to develop and verify MCP Platform in support of the Integrated System, and define the specific MCP artifacts satisfying the AC 20-193 objectives.</p> <p>This planning document can be a separate plan or included in another Development Plan.</p>
Requirements Specification*	Configuration controlled requirement data for the [domain] created per the Development Plan.
Verification Data*	Test, Analyses, Simulation, and Inspection data created per the Development Plan to show the [domain] complies with the defined requirements.

*These documents are applicable to each *domain* (Systems, Hardware, or Software). These documents are usually separate by domain (i.e. System Development Plan, Hardware Development Plan, Software Development Plan). These can be broken down further into the following at the discretion of the development team.

Systems:

- Computing System
- Hosted Application System(s)
- Integrated System

Hardware:

- MCP Platform

Software:

- Core Software
- Hosted Application

2.3 Acronyms

Table 3 – Acronyms

Acronyms	Definition
AC	Advisory Circular
AEH	Airborne Electronic Hardware
AMACC	Army Military Airworthiness Certification Criteria
AMC	Acceptable Mean of Compliance
ARP	Aerospace Recommended Practice
ASA	Aircraft Safety Assessment
CDI	Configuration Data Item
COTS	Commercial-Off-The-Shelf
DDR	Double Data Rate
EASA	European Union Aviation Safety Agency
FAA	Federal Aviation Administration
FHA	Functional Hazard Assessment
FMEA	Failure Modes and Effects Analysis
HA	Hosted Application
HIC	Hardware Interference Channel
HW	Hardware
IC	Interference Channel
IMA	Integrated Modular Avionics
IO	Input/Output
MCP	Multi-Core Processor
MMU	Memory Management Unit
MOC	Means of Compliance
MOSA	Modular Open System Approach
OoO	Out of Order
OSA	Open System Approach
PHAC	Plan for Hardware Aspects of Certification
PCI	Peripheral Component Interconnect
PIMAC	Plan for Integrated MCP Aspects of Certification
PR	Problem Report
PSAC	Plan for Software Aspects of Certification
PSSA	Preliminary System Safety Assessment
QoS	Quality of Service
RAM	Random Access Memory
RTOS	Real-time Operating System
SCP	Single Core Processor
SoC	system-on-chip
SPI	Serial Peripheral Interface
SSA	System Safety Assessment

Acronyms	Definition
SW	Software
TBD	To Be Determined
TDMA	Time Division Multiple Access
TSO	Technical Standard Orders
V&V	Validation and Verification
WCET	Worst Case Execution Time

3 Roles & Responsibilities

The following roles separate the different areas of development and integration. It is possible that multiple teams/suppliers are used for the different roles defined below or the same team/supplier covers multiple roles.

Table 4 – Roles and Responsibility

Role	Primary Responsibility
Aircraft Developer	The team or group responsible for the architecture of the Aircraft and the required functions of the Aircraft. They complete the required activities to demonstrate the Integrated System(s) on the Aircraft.
Hosted App Provider	The team or group responsible for architecture of the Hosted Application Software. They define requirements and verify that the software, as executing on the MCP Platform, functions as intended.
Integrator	<p>The team(s) or group(s) responsible for the architecture of the Integrated System. They complete the required activities to integrate the Computing System with the Hosted Function System(s), which includes ensuring lifecycle data handoffs among the required roles. They define and verify the configuration and allocation of the shared resources.</p> <p>Multiple groups may be needed to completely satisfy the Integrator role. For example: The supplier may fulfill aspects of the Integrator role to facilitate development of the Integrated System. As the Integrated System is integrated with other systems on the Aircraft, the Integrator role may be needed at the Aircraft level.</p>
MCP Platform Provider	The team or group responsible for architecture of the Computing System and the MCP Platform. They define requirements and verify that the integration of all MCP-related equipment, hardware, and software functions as intended.

As a secondary responsibility, each role provides support to the other roles for the completion of the development. As an example, the Integrator should support the Aircraft Developer with the demonstration activities.

4 Developing the Computing System

The MCP Platform Provider, responsible for the shared resources of the platform, consumes the Integrated System requirements and design data from the Integrator. The MCP Platform Provider defines requirements and design objectives for the Computing System and MCP Platform that support the parent requirements. The MCP Platform Provider produces the required lifecycle data per the agreed-to industry standards and Development Plan(s). These sections identify additional considerations to ensure MCP Platform aspects are adequately captured.

Figure 4 provides a generic diagram to delineate the boundary of the MCP Platform (light blue) and the Computing Equipment (dark blue).

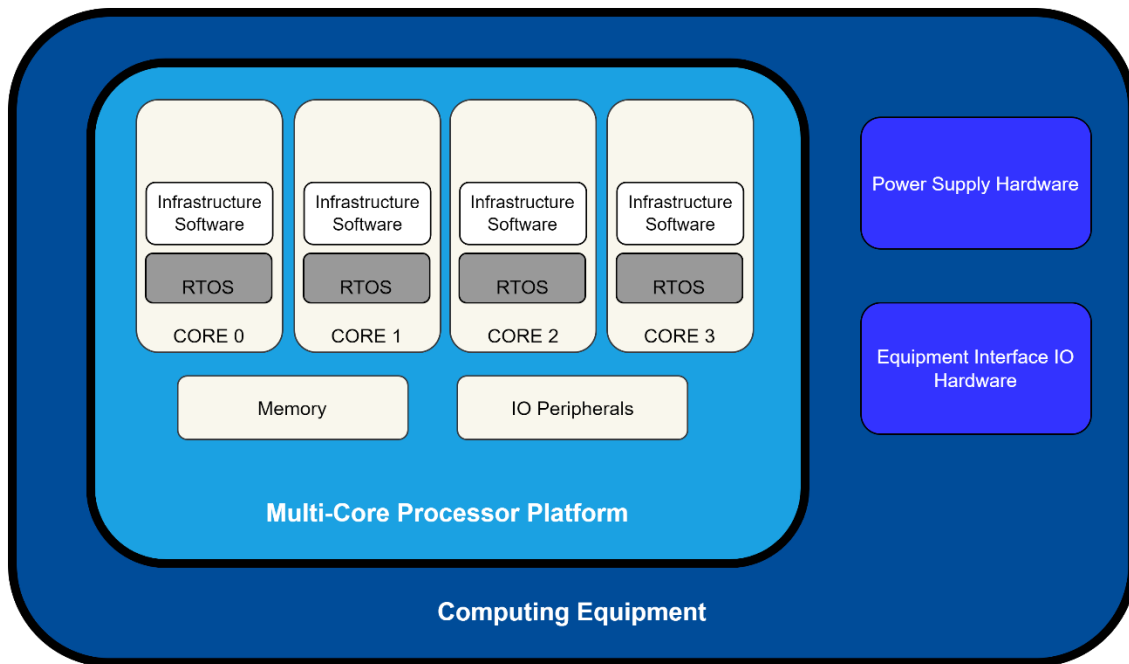


Figure 4 – Boundary of Computing Equipment and MCP Platform

The Computing System (Figure 5) is comprised of the Computing Equipment and other elements and components, such as aircraft sensors. The following subsections focus on the MCP Platform development and less on the development of the Computing System.

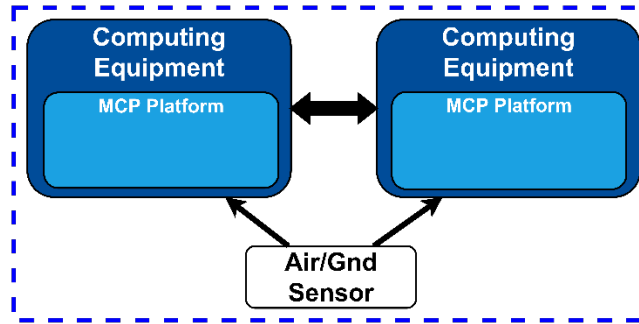


Figure 5 – Example of Computing System

The role of the Integrator requires coordination with the MCP Platform Provider and Hosted App Provider and therefore the outputs of data within Section 4 are provided to the Integrator. The Integrator provides the necessary MCP Platform lifecycle data to the Hosted App Provider. It is recommended that the Hosted App Provider and MCP Platform Provider coordinate as needed with the Integrator.

4.1 Plan the MCP Platform

Early in the development, the MCP Platform Provider receives stakeholder needs for a computing system. With these needs, Use Cases are developed on how this system utilization is envisioned, particularly for the shared computing resource. After developing this planning data, the MCP Platform Provider proposes a preliminary MCP Platform architecture.

An important first step is documenting the plan to develop this Computing System down to the hardware and software. The MCP Platform Provider utilizes development plans for requirement definition and validation, assessing safety features, and verifying the equipment. The planning documents follow the guidance of the agreed-to industry standards.

Inputs: Aircraft/Integrated System needs and use cases

Inputs from: Integrator/Aircraft Developer

The MCP Platform Provider completes the following activities:

1. Consume Aircraft/Integrated System needs for a Computing System with shared resources.
2. Develop Use Cases describing how the Computing System could be utilized.
3. Develop a proposed architecture identifying preliminary MCP Platform details (Section 4.1.1)
4. Complete the preliminary safety analyses which includes the Preliminary Interference Analysis (Section 4.1.2)
 - Identify Critical Configuration Settings
 - Develop Interference Channel (IC) Diagrams
 - Develop Partitioning Analysis
 - Develop Mitigation Strategies
5. Document the MCP Platform Plan (Section 4.1.3) using the data from the Preliminary Interference Analysis.
6. Develop Prototype Microbenchmarks (Section 4.1.4)

Outputs: Preliminary Interference Analysis, MCP Platform Plan, Microbenchmarks

Outputs to: MCP Platform Provider, Integrator

4.1.1 Preliminary MCP Platform Architecture

There are several aspects of an MCP Platform that need attention when planning the architecture of the processing subsystem, and they include both hardware and software.

First, as driven by the Computing System use cases mentioned above, a multicore chip, likely a system-on-chip (SoC) package, is selected as the heart of the MCP Platform. Whether internal to the SoC or as external devices, the surrounding processing subsystem components, such as memory controllers, bus controllers, power regulators, and memory devices have relevance to the MCP behavior and certification. The number and type of peripheral devices (input/output data, video, graphics, data storage, etc.) interfacing with the MCP Platform is also an area of consideration in defining the architecture. The hardware planning data should capture the multicore chip, processing subsystem components, and peripheral interfaces planned for use in the system and any relevant determinism or certification concerns with their use.

From a software perspective, the overall system intent for hosting application software should drive the selection process for the core software. This may include the use of a software hypervisor, a real-time operating system (RTOS), a general-purpose operating system (such as Linux or Windows), or a combination of these software types. Special attention should be given to software providing a safety critical environment, real-time operation, or any sort of partitioning of system functionality. These areas may see increased complexity when used in a multi-core processing scenario. Where hypervisor and/or operating system software is used, clear expectations must be set for that software's involvement in any MCP determinism concerns or airworthiness certification objective compliance needs. The software planning data captures the core software being used, how it is configured for the system, what responsibilities it has regarding deterministic operation and certification compliance evidence, and what vendor-supplied data or activities are applicable to the system certification.

4.1.2 Preliminary Safety Activities

The MCP Platform provider should develop a Preliminary Safety Assessment per the agreed-to industry standards and Development Plan. This analysis supports development of the MCP Platform design and validation of the safety requirements, traced and derived. The industry standard safety process is still executed for overall safety of the Computing System. Additional analyses support the safety objectives. Safety requirements for the MCP Platform can be derived from all analyses: Fault Tree Analysis, Common Mode Analysis, Partitioning Analysis, Interference Analysis, etc.

System Function Hazard Assessments and other safety lifecycle data (e.g., SSA or FMEA) should analyze determinism failures related to the MCP Platform and define the appropriate safety nets. The FAA Tech Report [8] provides guidance on these safety considerations.

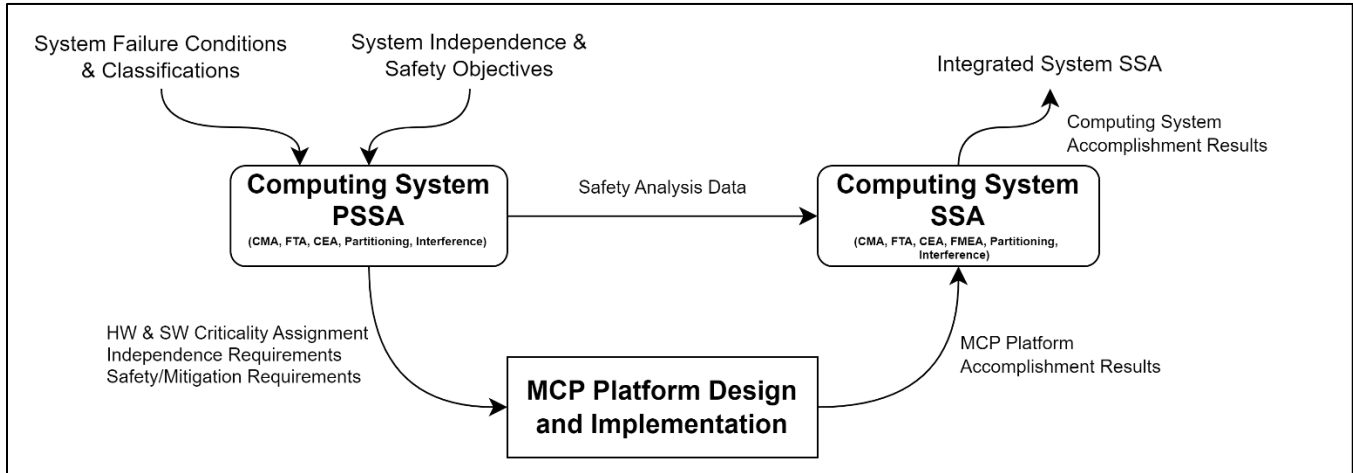


Figure 6 – Safety Process Flow

The development of a standalone Interference Analysis similar to the analysis described in the FAA Tech Report [8] is suggested. This interference analysis characterizes the performance of the MCP Platform such that the appropriate safety net(s) are designed into the MCP Platform, and an Integrator has the tools and analysis data to correctly allocate the shared resources to support the Hosted Applications. Similar to the objectives for a Preliminary System Safety Analysis, a Preliminary Interference Analysis should be generated early in the Plan phase to support the Requirements and Design Phase.

To effectively analyze the platform, the MCP Platform Provider needs to understand the intended configurations of the platform by the Integrator and Aircraft Developer. If the intended usage is unknown or not fully understood, the MCP Platform Provider should generate system use cases – assumptions on utilization of the MCP Platform by the Integrator and Aircraft Developer. Aspects to consider in the use cases:

- Number of and relative execution bandwidth required by the Hosted Applications on each core
- Allocation architecture of Hosted Applications executing on each core considering safety critical applications intermixed with non-safety critical applications
- Characterization of the Hosted Applications:
 - Access to peripherals
 - Routine tasks vs event driven tasks
 - Timing/Latency

When system use cases are utilized, the MCP Platform Provider communicates to the Integrator safety assumptions and MCP configuration constraints to ensure the analysis remains valid. These details are included in the User Guide (Section 4.5.4).

4.1.2.1 Preliminary Interference Analysis [Plan]

The Interference Analysis is a critical lifecycle data artifact to assess the MCP Platform and its ability to support the larger system functions. This preliminary activity should be completed early in the development process to establish the MCP Platform performance capabilities for a given set of system use cases and design. On SCP implementations, this analysis focuses on the partitioning analyses and software to software data coupling. For MCP Platforms, this activity needs to include the assessment and characterization of the Hardware Interference Channels. This Preliminary Interference Analysis includes the Interference Channel Diagrams to document the interference and microbenchmarks as a method to trigger interference to document the impact.

4.1.2.1.1 Critical Configuration Settings

Critical Configuration Settings are those MCP Platform settings that have the potential to impact deterministic operation of a Hosted Application executing on the MCP Platform.

A systematic method is needed to identify all relevant configuration registers for the selected MCP Platform and to tag the “Critical” registers. A critical register is defined as a register that impacts deterministic execution of the software or any settings that, if inadvertently altered, could change the behavior of the MCP Platform and cause unintended behavior of the Hosted Application. This list of configuration registers and those identified as critical are documented in the Preliminary Interference Analysis.

Examples of critical configuration registers:

- Cache Partitioning Registers
- DDR Configuration Registers
- IO Device Registers
- Clock Configuration Registers

The Preliminary Interference Analysis also documents the proposed mitigation to prevent or status inadvertent modification to the Critical Configuration Settings.

4.1.2.1.2 Interference Channel Diagrams

The Interference Channel (IC) Diagrams are a tool for communicating the intended boundaries for potential sources of interference between the Hosted Applications executing on the MCP Platform. The diagrams are useful in describing the interactions between the different interference channels in a consistent and visual manner and in quantifying pertinent interference channels thereby removing superfluous details of MCP Platform design. These IC Diagrams are included in the Preliminary Interference Analysis. The FAA Tech Report [8] provides an example Interference Channel Diagram as shown in Figure 7.

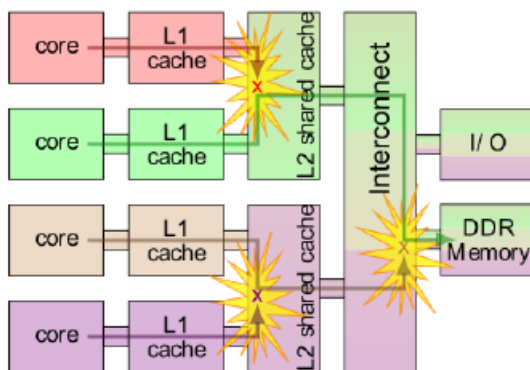


Figure 7 – Example Interference Channel Diagram from FAA Tech Report

Evaluation of interference channels should begin from the perspective of the functional applications utilizing the MCP Platform. The impact of interference varies based upon how applications are able to access MCP Platform resources. IC Diagrams detail how the execution of an individual application is impacted by the shared resource usage of other functional entities utilizing the MCP. This causal relationship of application usage and impact to the shared resource for each identified Interference Channel is included in the Preliminary Interference Analysis.

The modeled interference encompasses all shared memory and non-memory resources of the MCP Platform. It is probable the modeled interference shows impactful interference on memory accesses, whether for instructions or data. It is for this reason that memory-bound or peripheral-intensive applications are more susceptible to the effects of MCP interference than applications that are logic/execution bound. When evaluating a single application, interference channels can be identified by tracing memory accesses throughout the MCP memory system. Each step of the memory hierarchy reveals additional memories, bus masters, peripherals, or other shared resources that have the potential to induce interference.

While the focus is on memory accesses from the perspective of the processing core, this should not be taken to imply that all memory accesses are the same. Memory accesses may be associated with shared IO resources, networks (PCI, Ethernet, SPI, etc.), or other peripheral interfaces on the MCP device. These accesses should not be viewed as equivalent despite them being accessed through the MCP memory map.

An Interference Channel Diagram should include the following details:

- Shared resource of interest (target)
- Transaction initiators
- Initiator-Target Relationship

Shared resources for which there are initiators owned by multiple applications are potential sources of MCP interference. Assuming the above attributes are addressed, the format of the IC Diagram modeling is at the discretion of the MCP Platform Provider. It is recommended that all stakeholders agree to the formatting, including the customer and relevant regulatory authorities.

An example Interference Channel Diagram is shown in Figure 8 targeting the interference with L3 cache. In this example, each processing core has a private L1 cache and an L2 and L3 cache that is shared amongst core pairs. Each element in the diagram is either a target, initiator, or both. The arrows in the diagram describe the initiator-target relationships. Each target element in the diagram has a transaction response time that is a function of the initiators. As illustrated below, the MCP interference model quickly becomes a complex system of equations.

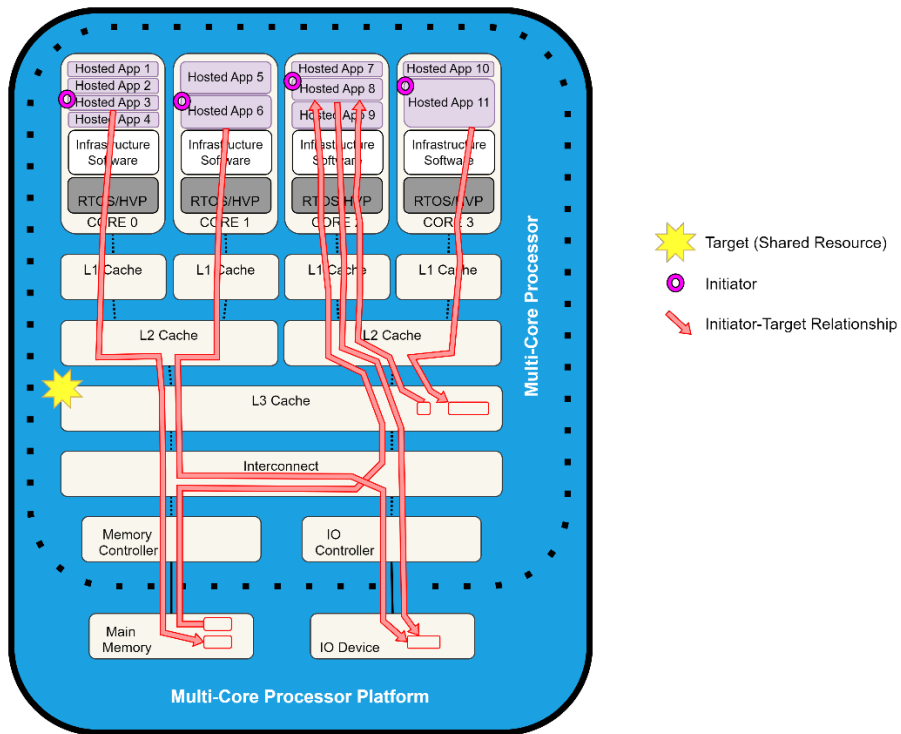


Figure 8 – Example Interference Channel Diagram of L3 Cache

The information necessary to construct IC Diagrams is generally sourced from technical reference manuals or other design documentation available for the MCP. It should not be assumed that publicly available material is sufficient for this activity unless supported by the testing activities conducted against the IC Diagrams. When utilizing an MCP for which design data may be limited, it is extremely difficult to ensure that all interference channels have been identified and accurately portrayed through the diagrams. As a result, testing must be relied upon to validate the relationships described by the IC Diagrams.

4.1.2.1.2.1 Interference Channel Impact Measurements

For each interference channel identified in the IC Diagrams, impact measurements of the interference channel are assessed (may be qualitative or quantitative).

The MCP Platform Provider has the burden of validating the causal relationships described by their IC Diagrams. This may be accomplished through testing, simulation, or a combination of data from both. Frequently, the MCP Platform Provider also seeks to quantify the relative performance impact of interference on each of the interference channels to ease the later integration process of applications onto the MCP Platform. The fidelity of the performance impact measurements differs between programs and MCP platform design approaches. Measurement approaches that are unable to validate the causal relationships of the interference channels, though, may result in wasteful amounts of margin in assigned application execution times or onerous testing during platform integration or risk to the certification acceptance.

4.1.2.1.3 Partitioning Analysis

The Interference Analysis should include a Preliminary Partitioning Analysis that assesses hardware and software partitioning vulnerabilities and required mitigations for unacceptable vulnerabilities. Mitigation requirements are derived in the Design phase to resolve the vulnerabilities.

The assessment of partitioning should consider vulnerabilities related to possible hardware interference. Note that vulnerabilities may be inherent in the hardware and always present, or they may be exposed, amplified, or constrained by the Core Software design.

4.1.2.1.4 Mitigation Strategies

The Preliminary Interference Analysis defines mitigation for the resource interference identified in the IC Diagrams and for partitioning vulnerabilities. These mitigations are commonly implemented through software hosted on the MCP Platform but may rely on performance characterization measured by the MCP hardware.

Definition of mitigation strategies should occur prior to MCP Equipment HW and SW development. The primary goal of the early definition of mitigation strategies is to allow for the evaluation of the MCP interference in the target configuration before significant investments are made in MCP Platform development. Depending on the severity of performance impact of each interference channel, it may be necessary for an MCP Platform Provider to prototype proposed mitigations to evaluate their effectiveness. Additionally, the negative impact of mitigations on MCP Platform performance should be considered when evaluating each mitigation.

For interference channels that are mitigated, the IC Diagram identifies the mitigation and describes its relative effectiveness, utilizing the relevant impact measurement. As the analysis is completed, it may be determined that mitigation is not needed for a given IC Diagram because the HIC impact is negligible.

The partitioning analysis identifies the mitigation used to resolve the partitioning vulnerability. Similarly, it may be determined that no mitigation is required because the impact is negligible.

Alternatively, the MCP Platform Provider may define mitigations that result in design constraints passed to the Integrator. These mitigation strategies are discussed in Section 4.5.

4.1.2.1.5 Iteration

The early planning phase can be iterative as more data characterizing the MCP Platform is realized and refined. Any changes to the design are reanalyzed to determine impacts to Critical Configuration Settings, IC Diagrams, Partitioning Analysis, and Mitigations. The Preliminary Interference Analysis is continually updated as required for these impacts.

It is recommended that the MCP Platform Provider define the point to transition from the planning phase to the design phase.

4.1.3 MCP Platform Plan

Like a Plan for Hardware Aspects of Certification (PHAC) or Plan for Software Aspects of Certification (PSAC), the MCP Platform Plan communicates the justification and means proposed to meet the airworthiness objectives (e.g., AC 20-193). The MCP Platform Plan includes the following details:

- Provide a high-level description of how MCP shared resources are used/allocated and the safety mechanisms to detect and handle MCP failures. This description should include the provided or assumed system use cases that validated the architecture.
- Identify the specific MCP processor, including the unique identifier from the manufacturer.
- Identify the number of active cores.
- Identify the Core Software architecture.
- Identify any dynamic features provided in software hosted on the MCP that are activated and provide a high-level description of their usage.

- Identify whether or not the MCP device is used in an IMA platform to host software applications (potentially from multiple suppliers and developed independently).
- Identify whether or not the MCP Platform provides Robust Resource and / or Time Partitioning.
- Identify any Hardware dynamic features of the MCP device that are active and their usage. Examples:
 - Caching
 - Prefetching and branch prediction
 - Register renaming
 - Out of Order (OoO) execution
 - Interconnect QoS
- Identify any MCP-specific methods and tools to develop and verify the MCP design.
 - Define the iterative process to execute the Interference Analysis and develop requirements from the analysis.
 - Define/identify the safety process.
 - Define process to allocate and verify the use of shared resources. (Section 4.1.3.1)
 - Define tools required to support development and verification of the MCP. This includes performance tools to measure performance of the allocated MCP Architecture.
 - Define tools required to configure the MCP Architecture and allocate Hosted Software Applications.
 - Define verification approach, including the use of data flow diagrams, control flow diagrams, data dictionaries, calling trees, traceability tagging of data flow and control flow elements, and the use of software tools that support the requirements-based test coverage and structural coverage analyses of the data and control coupling analyses.
- Identify the Integrator Mitigation Strategies (Section 4.5) to support WCET for Hosted App Providers and Integrators
- Scope of MCP Platform User Guide.
- Proposed means to satisfy airworthiness objectives.

Several of the details identify capabilities or configurability of the MCP Platform. For example, the number of active cores may be configurable by the Integrator. Therefore, the MCP Platform Plan specifies the capability of the number of active cores. The Integrator specifies the number of active cores in the final configuration.

The MCP Platform Plan should be provided to the Integrator to support creation of the Plan for Integrated MCP Aspects Certification (Section 6.3).

Note: The MCP Platform Plan details can be included in other Development Assurance planning document(s). The MCP Platform Plan can reference lower level PHAC or PSAC documents if covered.

MCP Platform Planning Checklist can be utilized to complete the details needed for the MCP Platform Plan. (Section Appendix A)

4.1.3.1 Shared Resource Verification Plan

The Shared Resource Verification Plan defines the method to ensure the MCP Platform satisfies the requirements as well as show the shared resources are properly partitioned for aspects like HIC impacts. Microbenchmarks are test software elements intended to trigger HIC. While use of Microbenchmarks is not mandated, they are strongly recommended as a means by which an MCP platform may be examined.

The MCP Platform Plan defines the process for microbenchmark development providing coverage for all critical interference channels identified in the IC Diagrams. The plan defines the types of measurements needed to support characterization of the MCP Platform.

Note: Alternate verification methods should be documented within the MCP Platform Plan and gain the appropriate concurrence.

4.1.4 Microbenchmarks

Microbenchmarks can be utilized through the entirety of the MCP Platform development process. It is recommended that preliminary microbenchmarks be developed during this early planning phase to understand MCP Platform characteristics.

Microbenchmarks can be used to characterize the impact of various interference channels in the MCP and validate the associated IC Diagrams. The initial characterization informs Core Software definition, as well as the definition of the application environment, to better mitigate or tolerate the effects of interference.

Microbenchmarks are developed and configured to align with the layout of the shared resources for a specific MCP platform and use case of interest. This may include aspects of both hardware and software. Using the defined IC Diagrams, the MCP platform must be analyzed for interference channels within its shared resources. Microbenchmarks are then utilized to exercise the interference channels in a well understood pattern. As an example, a microbenchmark may be written to access and evict cache lines in a predetermined pattern to reveal the specific cost of each of the operations. These low-level interactions with the hardware are not evident if only running more generic, hardware agnostic, benchmarks.

When utilized to support the development and integration objectives of an MCP Platform, microbenchmarks must be carefully defined so that their functionality is well understood by the MCP Platform Provider. The following attributes should be defined for each microbenchmark:

- Interference Channel(s), and associated measurements, that the microbenchmark is targeting
- Interference Channel(s), and associated measurements, that are incidentally impacted by the microbenchmark
- Instruction-level description of program execution within the hardware
- Definition of all microbenchmark configuration parameters

These details noted above should be documented in the Preliminary Interference Analysis.

Results from the microbenchmarks can also show negligible impacts of a specific interference channel. This data is valuable to justify the impact of the defined IC Diagram and mitigations for that Interference Channel are not required. This justification should be documented in the Preliminary Interference Analysis.

4.2 Design the MCP Platform

The MCP Platform Provider develops Requirement Specifications and Design Data for the Computing System and MCP Platform. This specification defines the functional, performance, and integrity requirements. Additional Architecture Design data is defined to assess the high-level design concept, providing further clarity on implementation of the requirements. Preliminary safety analyses are conducted to validate the requirement set has the appropriate design assurance level and safety requirements.

Inputs: Preliminary Interference Analysis, Integrator requirements

Inputs from: MCP Platform Provider, Integrator

The MCP Platform Provider completes the following activities:

1. Define MCP Platform requirements (Section 4.2.1)
 - a. Consume Integrator requirements
 - b. Consume Preliminary Interference Analysis and define mitigation requirements
 - c. Allocate and decompose Hardware and/or Core Software requirements
2. Mature the detailed design data (Section 4.2.2)
 - a. MCP Platform Architecture
 - b. Hardware
 - c. Core Software
3. Complete impact analysis on Preliminary Interference Analysis
 - a. Update analysis for changes to Critical Configuration Settings, IC Diagrams, and Mitigations
 - b. Trace defined requirements to Critical Configuration Settings and Mitigations
4. Validate requirement set (Section 4.2.4)

Output: Validated requirements, detailed design data (including complete MCP architecture design), Traceability data

Output to: MCP Platform Provider, Integrator

4.2.1 MCP Platform Requirements

MCP Platform Requirements are decomposed from the Computing System requirements which define aspects like IO capabilities, Processing capabilities, System safety features, etc.

The following categories of requirements are necessary to achieve a robustly designed MCP platform. The fidelity of these requirements vary based on the design assurance level of the specific MCP Platform. The Preliminary Interference Analysis (Section 4.1.2) is a crucial input for requirement development as it identifies vulnerabilities in the platform.

- Develop requirements that define mitigation of hardware interference channels
- Develop requirements related to the MCP configuration settings that enable/disable functionality in HW or SW and configuration of the shared resources (e.g., cache). These requirements establish the application environments that is supported by the MCP Platform either configured by the MCP Platform HW/SW or configured by the Integrator at the time of CDI.
- Develop requirements related to the MCP configuration settings that prevent or mitigate unacceptable modification to or corruption of critical configuration settings. These monitors are relied upon for safety and partitioning analyses of the MCP Platform.
- Develop requirements describing the qualitative probability of hosted application safety impacts due to MCP interference. These types of requirements allow for the MCP interference considerations in fault tree analyses, if desired to be included in the safety analysis. They additionally force the rigor of the MCP Platform design to be commensurate with the criticality of the functions it is intended to support and the failure modes that may result in a functional hazard.
- Develop requirements related to robust partitioning. This includes partitioning mechanisms to maintain time and resource partitioning of the MCP Platform.

MCP Platform Requirements are baselined and controlled per Configuration Management process(es). When a baseline is snapped, these MCP Platform requirements are allocated to Hardware, Core Software, or both. The requirements are

further decomposed and derived at lower Hardware and Core Software (e.g., RTOS) levels. As new baselines are created, Configuration Management processes determine the impact and required action at the lower requirement level.

It may be necessary for the MCP Platform Provider to define “requirements” for the Integrator to comply with to ensure robust partitioning is maintained. For example, when an Integrator is allocating the Hosted Applications to the cores, all hosted applications that utilize a particular serial data bus on the MCP Platform must be allocated to the same core. These “requirements” should be included in the MCP Platform User Guide (Section 4.5.4).

4.2.2 MCP Platform Design Data

Design data is matured to show that the MCP Platform design fulfills the defined requirements. This design data includes the following details:

- Complete and detailed architecture description of the MCP Platform
- Integration details of the hardware and software which includes the inter-core interface boundaries
- System assumptions
- Low level Hardware and Core Software design data (or references to the specific Hardware/Software Design Documents)
- Constraints/Limitation

The necessary MCP design details should be included in the MCP Platform User Guide (Section 4.5.4) to support the Integrator activities.

4.2.3 Preliminary Multi-Core Interface Analysis [Design]

As the requirements and design mature, there could be changes to the MCP Platform Design, Critical Configuration Settings, or Mitigations that could impact the Preliminary Interference Analysis. The Configuration Management process defines the method to reanalyze and update the analysis for each baseline.

4.2.3.1 Critical Configuration Settings

All critical configuration registers that are used within the MCP Platform and the mitigations for inadvertent modification must show traceability to requirements establishing their use within the platform as well as the settings they must contain.

4.2.3.2 Interference Channel Diagrams

Mature MCP Platform design data is reviewed with the existing IC Diagrams and changes are incorporated into the Preliminary Interference Analysis, if required.

4.2.3.3 Partitioning Analysis

Mature MCP Platform design data is reviewed, and the partitioning analysis is modified as needed.

4.2.3.4 Mitigation Strategies

The Preliminary Interference Analysis must provide traceability to requirements establishing the mitigation for interference and partitioning. If changes were introduced during the requirement and design capture phase, the analysis is updated for the defined mitigation(s).

4.2.4 MCP Platform Requirement and Design Validation

When a baseline is snapped, the set of MCP Platform requirements are validated following the guidance of the agreed-to industry standards. The validation activity ensures correct tracing of the requirements to the Preliminary Interference Analysis.

4.3 Build the MCP Platform

The MCP Platform Provider continues the development of the hardware and Core Software (i.e. RTOS) per the agreed-to plans, creating the Design data and Implementation data. When these elements are determined mature, the MCP Platform Provider implements and integrates the Hardware and Core Software which assembles the MCP Platform and manages the configuration.

Inputs: Valid requirements, Microbenchmarks

Inputs from: MCP Platform Provider

The MCP Platform Provider completes the following activities:

1. Consume requirements and implement Hardware and Core Software (Section 4.3.1)
 - a. Prototype MCP Platform and test with Microbenchmark and other test software
 - b. Document undesired behavior and redo Design Phase as needed
2. Implement Test Software to support verification (Section 4.3.2)
 - a. Test Applications
 - b. Microbenchmarks
 - c. MCP-specific Tools

Output: MCP Platform, Microbenchmarks, Test Application, Tool(s)

Output to: MCP Platform Provider, Integrator

4.3.1 Prototype MCP Platform

Due to the complexity of MCP Platforms, prototypes should be used to confirm the design approach of the MCP Platform, as well as its performance under contention. The prototype is valuable in confirming the accuracy and feasibility of the Interference Analysis before making substantial investment into the building and verification of the MCP Platform. When applicable, microbenchmarks should be used to evaluate the performance and interference of the MCP Platform, as described in section 4.3.2.2.

Development of an MCP Platform should be viewed as an iterative process. In scenarios where the prototype performance does not match expectations, the MCP Platform moves back to the design phase. This type of iteration should be allowed for in program scheduling, as the detailed behavior and performance of MCP Platforms is difficult to fully understand prior to implementation.

The MCP Platform prototypes may additionally be useful for hosted application development. In many cases, the MCP Platform design shapes the development of Hosted Applications. Early access to representative hardware allows application developers to examine and tune application performance as necessary. This also allows the opportunity for application developers to provide feedback to the MCP Platform Provider, if desired.

The configuration of the specific MCP Platform (hardware and software) should be maintained to clearly trace the verification results to the uniquely configured MCP Platform.

4.3.2 Test Software

Test Software is the combination of two types of software:

- Test Applications: Generic software to mimic hosted application software utilizing the MCP Platform shared resources in a typical environment.

- Microbenchmarks: Software hosted on the MCP Platform to trigger interference on a specific interference channel(s).

To prepare for verification of the MCP Platform, test software is needed to verify requirements as well as finishing the characterization of the MCP Platform for Worst Case Execution Test (WCET) models. These test software builds should be under configuration control as defined in the Configuration Management plan. Figure 9 shows an example how Test Applications and Microbenchmarks could be allocated to support verification and characterization activities.

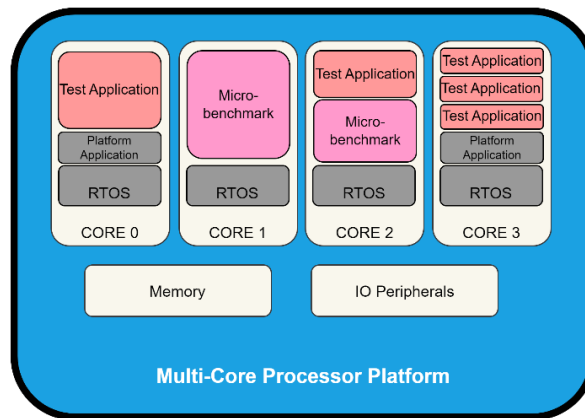


Figure 9 – Example Test Software Configuration

The MCP Platform Verification plan needs to detail the verification strategy, and this strategy determines the quantity of Test Applications and Microbenchmarks that are required to verify the MCP Platform.

4.3.2.1 Test Applications

Test applications should be created that simulate the expected use cases for the Computing System. If the Integrator use cases are unknown, the MCP Platform Provider generates applications that align to the assumed use cases and applications that support identifying constraints for the Integrator. When employed, the assumed use cases should clearly document the constrained usability and should be described in the MCP Platform User Guide (section 4.5.4).

4.3.2.2 Microbenchmarks

As discussed in section 4.1.4, microbenchmarks are unique test software to trigger interference. Once the MCP Platform is implemented, the verification plan along with the IC Diagrams are reviewed to develop the microbenchmarks by the MCP Platform Provider. The MCP Platform Provider should ensure the microbenchmarks are verified to perform their intended function (triggering the expected interference from the IC Diagram(s)). If preliminary microbenchmarks were created during the planning phase, the MCP Platform Provider can determine if the reuse of these microbenchmarks is acceptable. It may be required to modify preliminary microbenchmarks or provide additional configuration controls to allow for verification. The details on the microbenchmarks should be documented in Final Interference Analysis or similar verification lifecycle data.

4.3.2.3 MCP-Specific Tools

Development and verification tools are utilized to support test environments and creation of these test software applications. Tools are covered in Section 9.

4.4 Verify the MCP Platform

Per the agreed-to industry standards and Development Plan(s), the MCP Platform Provider verifies the implementation of Computing System and MCP Platform. This verification data can be contained in System/Equipment Test Reports, System Analyses, System Inspection Records, or a combination of these data item.

Inputs: MCP Platform, Microbenchmarks, Test Application, Tool(s), Preliminary Interference Analysis

Inputs from: MCP Platform Provider

The MCP Platform Provider completes the following activities:

1. Define Verification Procedures (Section 4.4.1)
 - a. Trace Procedures to requirement
2. Execute Verification Procedures and document results (Section 4.4.1)
3. Complete the final safety analyses which includes the Final Interference Analysis (Section 4.4.2.1)
 - a. Configuration Analysis
 - b. IC Analysis
 - c. Mitigation V&V
 - d. MCP Platform Timing Characteristics

Output: MCP Platform Verification Data, Final Interference Analysis

Output to: MCP Platform Provider, Integrator

4.4.1 Verification Procedures and Results

The MCP Platform Provider verifies the requirements captured from the activities in Section 4.2. Verification procedures are developed and traced to requirements. The verification procedures are executed on the MCP Platform along with the required Test Applications and Microbenchmarks to ensure the robust partitioning complies with requirements and design of the MCP platform. Execution of verification procedures also provide observable impacts of a particular resource interference on executing applications. If any inter-core data or control coupling exists in the Core Software design, the verification of this coupling is required and should consider the impacts of the Interference Channels.

The verification results are captured and controlled per the Configuration Management Plan. Verification results that were unsuccessful are reviewed and determined if acceptable. If results are unacceptable, the appropriate method for reporting errors is used to drive change to the design/requirements, implementation, or verification procedure.

4.4.2 Final Safety, Partitioning, and Interference Analyses

The MCP Platform provider should develop a Final Computing System Safety Analysis per the agreed-to industry standards and Development Plan. This analysis verifies safety requirements levied on the Computing System & MCP Platform and supports the Integrator and Aircraft Developer in the System and Aircraft Safety Analyses.

4.4.2.1 Final Interference Analysis

The Preliminary Interference Analysis transitions to the Final Interference Analysis to ensure all partitioning properties have been satisfied and verified, and all identified vulnerabilities have justification for mitigation. It is possible that an identified vulnerability has negligible system level impacts and therefore is justified to have no required mitigation. This justification is provided in the analysis. All associated verification data is referenced from the analysis. The Final Analysis is additionally responsible for summarizing any MCP interference mitigations that must be implemented and verified by the Integrator.

4.4.2.1.1 Configuration Analysis

Analysis provides the final critical configuration settings, traces the requirements defining the settings, and traces the requirements defining mitigations for inadvertent behavior.

4.4.2.1.2 IC Analysis

Analysis provides all finalized IC Diagrams with the assessment of interference channel causes and effects. The analysis provides detailed justification on the IC Diagrams deemed negligible.

4.4.2.1.3 Partitioning Analysis

The Final Partitioning Analysis verifies the mitigation requirements were successfully implemented to resolve vulnerabilities. If the vulnerability cannot be fully mitigated by the MCP Platform, the vulnerability details are included in the MCP Platform User Guide to ensure the Integrator mitigates appropriately. For example, the MCP Platform may be designed to allow multiple Hosted Applications to access to the same memory region. However, unauthorized Hosted Application access is a partitioning violation. Therefore, the MCP Platform Provider includes in the user guide instructions on correctly implementing the configuration of the MCP Platform and on verifying the allocation was correctly defined.

4.4.2.1.4 Mitigation Analysis

Analysis summarizes the final mitigation requirements derived from the IC Diagrams and traces to the requirements and to the verification results showing the final MCP Platform configuration complies. Unmitigated vulnerabilities are identified with justification explaining final configuration is acceptable and with constraints or instruction for the Integrator. These constraints and instructions for the Integrator are included in the User Guide.

4.4.2.1.5 MCP Platform Timing Characterization

This analysis shows the timing characteristics by compiling the data recorded from the Microbenchmark testing. Statistical analysis is likely utilized to obtain minimum, maximum, and average timing.

Analysis details include:

- Definition of Microbenchmarks
- Definition of Test Applications / Use Cases
- Justification for any assumptions
- Reference to raw data, if excluded from the analysis
- Final Time Characterization data

If the MCP Platform includes infrastructure software (e.g., Health Monitoring application), the WCET analysis is completed to define the appropriate allocation for these software applications. This allocation definition is provided to the Integrator to ensure the final allocation of the MCP Platform is acceptable.

4.5 Output

The MCP Platform Provider transitions to the Output phase when the verification is successfully completed. Per the agreed-to industry standards and Development Plan(s), the MCP Platform Provider completes the remaining lifecycle data.

Inputs: MCP Platform Verification Data, Final Interference Analysis

Inputs from: MCP Platform Provider

The MCP Platform Provider completes the following activities:

1. Develop the Integrator Mitigation Methodology

2. Develop the Integrator Configuration Definition Methodology
3. Complete the MCP Platform Accomplishment Summary
 - a. Defer open problem report, if acceptable
4. Complete the MCP Platform User Guide

Output: MCP Platform Accomplishment Summary, MCP Platform User Guide

Output to: Integrator

4.5.1 Integrator Mitigation Methodology

While the MCP Platform Provider mitigates the impacts of interference, it is likely that some interference considerations must be mitigated by the design of the Hosted Application and its allocation within the Integrated System. Using the data in the Interference Analysis (Section 4.4.2.1), the MCP Platform Provider should define the Integrator Mitigation Methodology to support the Integrator’s ability to successfully allocate the shared resources considering the impacts of interference. The Integrator executes this method as part of Integrator’s CDI process (Section 6.5).

As depicted in Figure 10, the Integrator Mitigation Methodology is a set of inputs provided by the Hosted App Provider or Integrator and has an output of the allocation of the MCP Platform shared resources. A feedback loop, defined as Refinement of the Allocation, is used to balance performance and impact of future change within the deterministic time and space elements.

The methodology should, at a minimum, define the relationship between the following (each is discussed in further detail below):

- Hosted Application Design Data
- Hosted Application Allocation
- Refinement of the Allocation

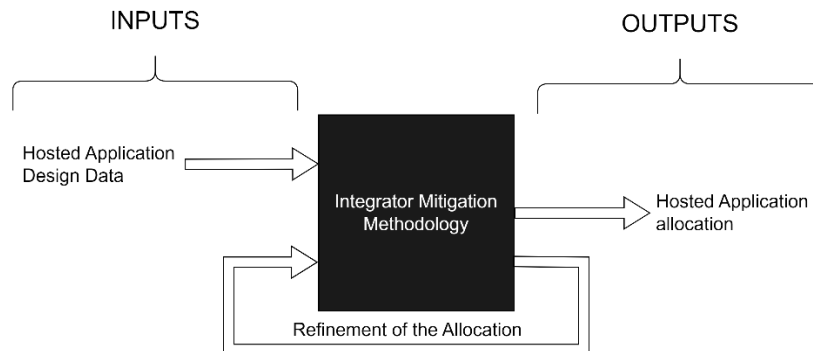


Figure 10 – Generic Integrator Mitigation Methodology

4.5.1.1 Hosted Application Design Data

The characterization data of the Hosted Applications executing on this MCP Platform is required to support accurate allocation. The MCP Platform Provider must define the types of inputs (required and optional) and instructions on what Hosted Application data should be measured, collected, or analyzed. The required design data is unique to the MCP Platform.

The following list provides critical considerations. These are considerations for the MCP Platform Provider to define for the given MCP Platform design and to provide to the Integrator (and Hosted App Provider) to support Hosted Application allocation.

- Instructions for setting up the HA WCET within a specified environment and mode of operation to measure HA data. For each HA measurement:
 - It is critical that the measurement environment is well defined. For example, a measurement may be taken in an uncontested environment (it is the sole user of shared resources), in an integrated environment (representative hosted applications accessing shared resources, in a heavily contested environment (in which microbenchmarks are inducing worst case resource contention), or some other specific environment. Regardless, each environment must be well defined for Hosted App Providers to generate accurate results.
 - The HA mode of operation must be considered at the time of measurement. As an example, it is common that the worst-case path through a critical deadline process must be characterized for correct worst case execution time analysis of the application in the integrated environment. Therefore, the MCP Platform Provider, with their detailed design knowledge of the platform interference analysis and mitigations, must define the expectations for the Hosted Application scenarios for a given measurement.
- Instructions on recording the measurements and data to provide to the Integrator.
 - To maintain a consistent and accurate set of results, the MCP Platform Provider defines the instructions and setup to record the required data.
- Required types of Hosted Application Design details. Examples include:
 - Quantified usage of the shared resources
 - Estimated execution time
 - Safety criticality level
 - Inter Hosted Application dependencies
- Hosted Application design best practices – Design efficiencies to minimize interference
- Integrated System design best practices – Design efficiencies to minimize interference

4.5.1.2 *Hosted Application Allocation*

The Hosted Application Allocation is the output of the Integrator Mitigation Methodology and is used to support the Integrator building the configuration of the Integrated System. Allocation reports and tools can be used to help the Integrator (and Hosted App Provider) to ensure the allocation supports the Hosted Application.

4.5.1.3 *Refinement of the Allocation*

Refinement of the allocation could be required to ensure a hosted application can successfully perform its function in an integrated MCP Platform environment due to the presence of interference. The Integrator Mitigation Methodology should establish a method for the Integrator to analyze the output (Hosted Application Allocation) and to apply allocation refinements. The following are examples of refinement types:

- Modifications to Allocation Margin
- Modifications to QoS
- Modifications of Core Allocation
- Modifications of Memory Allocation

4.5.1.4 *Special Considerations*

4.5.1.4.1 *Justification of the Hosted Application Allocation*

Justification of sufficient allocation is ultimately the responsibility of the Integrator. An engineering assessment of the Hosted Application Allocation should consider the functional hazards identified in the Integrated System Safety Analysis and determine the need for refinement. Assessments should also consider mission critical functions. For example, a non-critical application may not require refinement given no adverse impact to mission or safety. However, an intermix of critical and non-critical applications on the MCP Platform should necessitate adequate justification leading to potential iterative refinement of the allocation.

When analyzing the Hosted Application Allocation, several variables could influence the Integrator justification of sufficient allocation and the amount of refinement that is required. The following are examples of variables. The MCP Platform Provider should define all potential variables that could impact the Integrator's assessment.

- Hosted Application Design Data – Lack of knowledge or details on the Hosted Application, particularly on usages of the shared resources may require special justification or additional mitigation, depending on application criticality.
- Design of the MCP Platform – Simplified/Reduced Interference Channel approaches, limited microbenchmarks verification testing by an MCP Platform Provider (Section 4.3.2.2) or statistical outliers in collected data (Section 4.4.2.1.5) .
- Hosted Application System –Safety or mission critical requirements of the Hosted Application function(s).

4.5.1.4.2 *Distribution of Mitigation Methodology to Integrator*

The Integrator must understand the details of the mitigation methodology to ensure a compliant Integrated System can be configured that supports the requirements of the Hosted Applications. The MCP Platform Provider must publish the details of the mitigation methodology and provide this data to the Integrator. This data must include assumptions on use cases of the MCP Platform and any constraints. This data can be packaged at the discretion of the MCP Platform Provider however the MCP Platform User Guide (Section 4.5.4) is the recommended lifecycle data item. The mitigation methodology could be implemented into a tool suite to allow for efficiency in scheduling the integrated MCP Platform. Tool user guides and manuals are published if applicable.

4.5.1.4.3 *Validation of the Mitigation Methodology*

The Integrator Mitigation Methodology may vary in complexity based on the fidelity of the interference analysis or the integration processes of a specific program. Integrators may have additional needs when allocating their Hosted Applications and could drive changes to the methodology. For example, the Integrator may require no priority schemes for the Hosted Application thus limiting the refinement methods. Another example is a methodology that constructs an allocation supporting Hosted Application independence and allows for certain modifications to the Integrated System while not impacting the Hosted Application. The MCP Platform Provider should coordinate with the Integrator to ensure the mitigation strategies provided in the MCP Platform User Guide meet any program constraints or requirements.

4.5.2 *Integrator 'Configuration Definition' Methodology*

As described above, requirements define the configurable aspects of the MCP Platform. Concurrently with the development of the MCP Platform, the MCP Platform Provider finalizes the method for the Integrator to create the Configuration Data Item (CDI). This method can be accomplished within Tools (Section 9) or by other means to generate the configuration file which is typically a loadable software part.

The Configuration method should integrate with the methods and tools used for the Integrator Mitigation Methodology to support allocation of the Hosted Applications. Figure 11 provides a generic example.

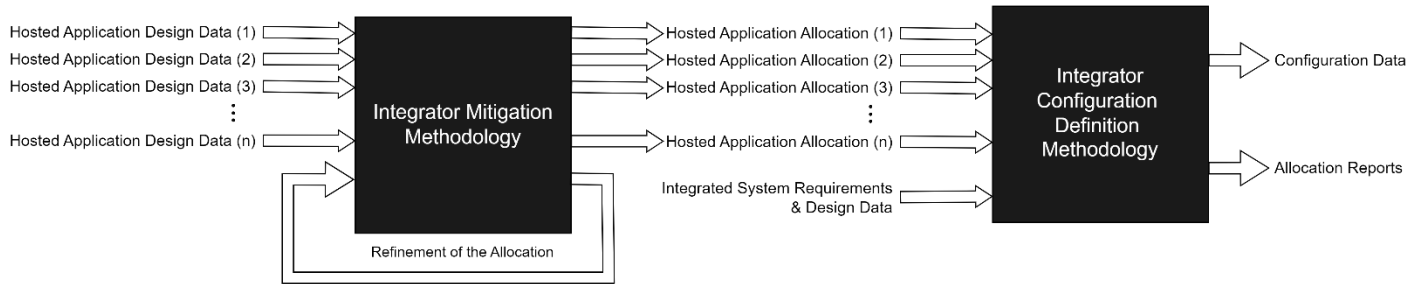


Figure 11 – Generic Configuration Definition Methodology

4.5.3 Accomplishment Summary

The MCP Platform Provider summarizes in the MCP Platform Accomplishment Summary successful completion of the development. This Accomplishment Summary document should also summarize the data produced by the MCP Platform Provider showing the required industry and airworthiness objectives were satisfied. Additionally, it should summarize the low-level Hardware and Core Software aspects or cross-reference the applicable Hardware Accomplishment Summary and Software Accomplishment Summary.

Completed lifecycle data items:

- MCP Platform Plan
- Interference Analysis
 - Configuration Analysis
 - Interference Channel Analysis
 - Partitioning Analysis
 - Mitigation Analysis
 - Time Characterization
- Integrator Mitigation Methodology
- Requirements (may include more than MCP Platform)
- Validation and Verification data (may include more than MCP Platform)
- Traceability data (may include more than MCP Platform)
- MCP Platform Configuration Index
- MCP Platform Tool documentation

The open problem reports against the MCP Platform, Hardware, and Core Software must be reviewed and justified to remain open. The list of all justified open problem reports along with the expected system level effects are included in the MCP Platform Accomplishment Summary. If an open problem report cannot be justified for deferral, the appropriate design change must be implemented to resolve the problem, and a change impact assessment is completed to determine what lifecycle artifacts are impacted by the change.

The MCP Platform User Guide should be referenced in the MCP Platform Accomplishment Summary.

4.5.4 MCP Platform User Guide

The MCP Platform Provider additionally develops the MCP Platform User Guide (which includes design details on the low-level Hardware and Software) to provide installation and integration instructions and constraints for the Integrator (Hosted App Provider) and Aircraft Developer.

Typical items to include in the MCP Platform User Guide are as follows:

- Description of the MCP Platform design
 - Available shared resources, configuration details
 - Assumptions on use cases supported by the design
 - Characterization of Hardware Interference and design constraints on the Hosted Application and Integrated System
 - RTOS specification and design aspects
- List of MCP Platform Lifecycle artifacts
- MCP Platform Interface and Installation specifications/instructions
- MCP Platform Tool(s) (Section 9)
 - Tool content and executable needs.
 - Tool instructions to correctly allocate shared resources for the Configuration Data Item (Section 6.5.2)
- Hosted Application design best practices – Design efficiencies to minimize interference
- Integrated System design best practices – Design efficiencies to minimize interference
- Compatibility Matrix of MCP Platform HW and SW
- System level impacts for MCP Platform Open Problem Reports
- Integrator Mitigation Strategies details
 - Required Hosted Application Design Data to collect/measure
 - Available Refinement of Allocation methods
 - Instructions to obtain final Hosted Application allocation of shared resources
- Define integration expectations:
 - Integrate Computing System (MCP Platform) with Hosted Applications
 - Operate/allocate the MCP Platform with the defined constraints on configurability and useability
 - Mitigate defined Partitioning and Interference Vulnerabilities at Integrator level
 - Complete verification activities at the Hosted Application and Integrator level
 - Complete the Determinism Analyses (Partitioning, Interference, WCET) for the Integrated System
 - Complete the System Safety Assessment
 - Validating key MCP Platform safety assumptions
 - Validating the FHA failure modes and affects (e.g., determinism failures)

5 Hosted Application System

Similar to the MCP Platform Provider, the Hosted App Provider consumes the Integrated System requirements and design data and defines requirements and design objectives for the Hosted Application System (HA System). Per the agreed-to industry standards and Development Plan(s), the Hosted App Provider produces the required lifecycle data. These sections identify additional Hosted Application considerations when hosted on an MCP Platform.

Figure 12 provides a generic diagram to delineate the boundary of the HA System. In this example, the Computing System (dashed blue box) is two interfacing Computing Equipment Units (dark blue) each including an MCP Platform (light blue). The HA System (dotted purple box) is comprised of the Computing System, the Hosted Applications hosted on the MCP Platform, and the additional hardware and software to support the function (Display Units, Control Panel).

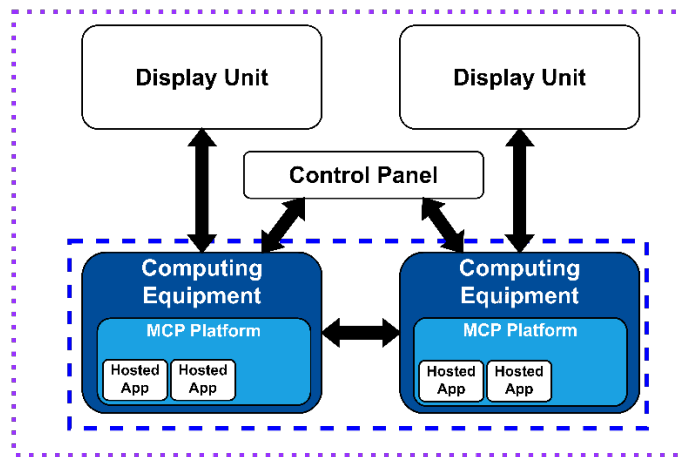


Figure 12 – Example #1 Hosted Application System

Depending on the Integrated System, there could be more than one Hosted App Provider. The following subsections are based on the assumption that the activities and lifecycle data are completed for each Hosted Application regardless of Hosted App Provider. Figure 13 provides a generic diagram of a second independent HA System also utilizing the same Computing System. This second HA System could be a different Hosted App Provider.

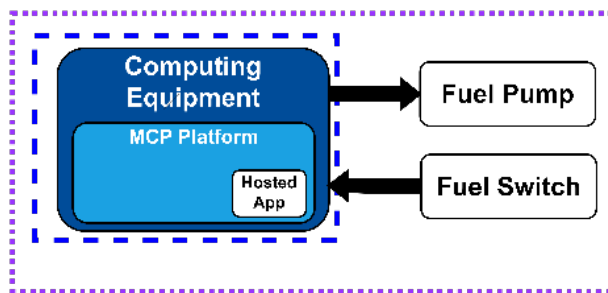


Figure 13 – Example #2 Hosted Application System

The Integrator provides the necessary MCP Platform lifecycle data to the Hosted App Provider. It is recommended that the Hosted App Provider and MCP Platform Provider coordinate as needed with the Integrator. The Hosted App Provider needs the details on the Computing System and MCP Platform to effectively plan, design, build and verify the Hosted Application(s). The MCP Platform User Guide is the preferred method to share the pertinent details (Section 4.5.4). If this specific lifecycle data is not available because the MCP Platform development and Hosted Application development are concurrent, the Integrator should still provide the required MCP Platform data by an alternative means until the user

guide is available. Whenever “MCP Platform User Guide” is listed as an input in the following subsections, it may be replaced with the required data from an alternate means until the user guide is available.

5.1 Plan the Hosted Application

The Hosted Application planning phase also includes other planning activities (e.g., PSAC development). This section only covers those aspects related to MCP Platform development.

Inputs: Aircraft/Integrated System needs and use cases, MCP Platform User Guide

Inputs from: Integrator, Aircraft Developer

The Hosted App Provider completes the following activities:

1. Consume Aircraft/Integrated System needs for the Hosted Application understanding the design aspects of the Computing System. (Section 5.1.1)
2. Develop a proposed architecture identifying preliminary Hosted Application details (Section 5.1.2)

Outputs: Preliminary Hosted Application Design data

Outputs to: Hosted App Provider

5.1.1 Aircraft/Integrated System Needs for the Hosted Application

The Integrator identifies the Aircraft/Integrated System needs and use cases to satisfy the Aircraft Function needs and objectives. The Hosted App Providers consume these needs and start planning the Hosted Applications. This planning includes preliminary design details on the Hosted Application(s) hosted on the MCP Platform. The Integrator also needs to facilitate the communication of appropriate MCP Platform design details to the Hosted App Provider.

5.1.2 Preliminary Hosted Application Design Data

The Hosted Application design needs to consider the aspects of the MCP Platform (i.e. RTOS, MCP, peripherals) and validate an architecture that satisfies the function(s) of the HA System. This preliminary architecture includes initial design of the Hosted Application(s) hosted on the MCP Platform. Given this is preliminary design activity, it may be too early to solidify all details. Here are some key design elements to consider:

- Hosted Application performance characteristics
- Hosted Application utilization in System Use Cases, identifying the MCP Platform shared resources
- Description of each Hosted Application, including details on mode of operation(s) and on safety considerations

5.2 Design the Hosted Application

The Hosted App Provider develops Requirement Specifications and Design Data for the HA System. This specification defines the functional, performance, and integrity requirements. Additional Architecture Design data is defined to assess the high-level design concept, providing further clarity on implementation of the requirements. These system requirements and design data decompose into software requirements and design data for the Hosted Application.

Inputs: Preliminary Hosted Application Design data, Integrator Requirements, and Interface data, MCP Platform User Guide

Inputs from: Hosted App Provider, Integrator

The Hosted App Provider completes the following activities:

1. Define software requirements for the Hosted Application(s)

2. Mature the detailed design data including the Computing System and MCP Platform aspects (Section 5.2.1)
 - a. Define the initial allocation time slice(s) and memory usage needs for the Hosted Application

Output: Hosted Application requirement and design data

Output to: Hosted App Provider, Integrator

5.2.1 Mature Hosted Application Design Data

The Hosted App Provider matures the preliminary design data while capturing requirements. This Hosted Application design data should incorporate aspects of the Computing System and MCP Platform to ensure complete set of requirements are captured. The key elements identified in Section 5.1.2 are completed as well as:

- Separation of Hosted Applications among Cores
- Hosted Application Execution details, IO rates, memory usage
- Safety criticality considerations

To ensure the Hosted Application properly functions on the MCP Platform, the Hosted App Provider must consume the MCP Platform User Guide and develop the Hosted Application requirements and design per the instructions, constraints, and limitations defined in the user guide. Some of these constraints could be requirements the Integrator allocates to the Hosted App Provider.

At this point in the development, the Hosted App Provider defines the initial characterization of the shared resources. The initial allocation of execution time and memory usage are important characteristics to define. Determining these initial allocations could be iterative process as the Hosted App Provider integrates the Hosted Applications. It is recommended that the Hosted App Provider document this initial allocation and the Hosted Application Resource Needs as this detail may be needed by the Integrator when integrated with other Hosted Applications from other providers.

5.3 Build the Hosted Application

Inputs: Valid requirements, MCP Platform (Computing System), MCP Platform Tools and Development Environments, MCP Platform User Guide

Inputs from: Hosted App Provider, Integrator

The Hosted App Provider completes the following activities:

1. Consume requirements and implement Hosted Application software (Section 5.3.1)
2. Integrate Hosted Applications with the Computing System (Section 5.3.2)

Output: Hosted Application Software, Hosted Application Resource Needs

Output to: Hosted App Provider, Integrator

5.3.1 Implement the Hosted Application Software

Per the agreed-to plans, the Hosted App Provider implements each Hosted Application software based on the captured software requirements. The software is developed using developmental environment(s) specified by the MCP Platform Provider. The MCP Platform User Guide provides details on the tool set and instructions on software development (i.e. compilers, errata details).

5.3.2 Integrate the Hosted Application Software

When these software elements are determined mature, the Hosted App Provider implements and integrates the Hosted Application(s) with the Computing System and with other HA System components (e.g., Figure 12). This integration requires the allocation of the MCP Platform shared resources.

5.4 Verify the Hosted Application

Inputs: Hosted Application System (including Computing System), Hosted Application, CDI Software, Allocation Reports, MCP Platform User Guide

Inputs from: Hosted App Provider, Integrator

The Hosted App Provider completes the following activities:

1. Verify the Hosted Application(s) (Section 5.4.1)
2. Verify the HA System (Section 5.4.2)
3. Execute Hosted Application WCET (Section 5.4.3) and worst-case memory usage (RAM / nonvolatile memory / file system allocation)

Output: Hosted Application Verification Data, Hosted Application WCET Data

Output to: Hosted App Provider, Integrator

5.4.1 Hosted Application Verification

Per the agreed-to industry standards and Software Development Plan(s), the Hosted App Provider verifies the implementation of Hosted Applications. This verification data shows the software requirements were successfully implemented. A developmental MCP Platform may be used to support requirement verification as well as preliminary verification on target hardware (e.g., MCP Platform).

5.4.2 HA System Verification

Per the agreed-to industry standards and System Development Plan(s), the Hosted App Provider verifies the Hosted Application within the HA System environment. This verification data can be contained in System/Equipment Test Reports, System Analyses, System Inspection Records, or a combination of these data items.

When testing the system, the verification environment should be representative of the HA System, which includes the Computing System, and could simulate any external IO as required. If any inter-core data or control coupling exists in the Hosted Application design, the verification of this coupling of the Hosted Application System is required and should consider the impacts of the Interference Channels.

5.4.3 Hosted Application WCET

Software industry standards define the need to perform WCET analysis and testing. Traditional WCET activity first determines the set of tasks within the Hosted Application that consume the most execution time of the allotted time slice. However, it is expected that additional timing related performance requirements (e.g., latency requirements) may need analysis to ensure coverage for contention in an MCP Platform environment. The activity also shows that the Hosted Application, under the specific configuration and mode of operation, can complete the necessary functions within the allocated processor time. Many factors can influence the WCET, and it is recommended to detail assumptions and configuration details.

The Interference Channels identified in the MCP Platform could negatively impact the WCET. The MCP Platform increases the complexity of factors that influence the WCET. The Hosted App Provider should consume the MCP Platform User

Guide to understand the Interference Channels impacts and to obtain instructions on how to execute WCET impact testing. The Hosted Application Provider provides the WCET Data to the Integrator to support allocation of the Integrated System.

5.4.4 Considerations for the Allocation of Shared Resources

Executing verification tests on the target MCP Platform and Computing System requires the allocation of the shared resources. The Hosted App Provider should coordinate with the Integrator on test environment(s) and determine the method of allocation. The Hosted App Provider could utilize MCP Platform Provider Tools to allocate the Hosted Application(s) and access to shared resources. This likely constrains the verification test to configuration that does not completely reflect the integrated system. Conversely the Integrator can provide CDI software (test only, preliminary, and/or final) to the Hosted App Provider to support Hosted Application verification. (Section 6.5).

The configuration details of the Hosted Application(s) are likely to change over the development of the Hosted Application, MCP Platform, and/or Integrated System. Additional verification activities may be required when changes are made to the CDI Software. CDI Allocation Reports can provide useful data to help in the determination of what additional verification is required.

5.5 Output

The Hosted App Provider transitions to the Output phase when the verification is successfully completed. Per the agreed-to industry standards and Development Plan(s), the Hosted App Provider completes the remaining lifecycle data.

Inputs: Hosted Application Verification data, MCP Platform User Guide

Inputs from: Hosted App Provider, Integrator

The Hosted App Provider completes the following activities:

1. Consume the MCP Platform User Guide (Section 5.5.1)
2. Complete the Software Accomplishment Summary and other required data for the Integrator (Section 5.5.2)

Output: Hosted Application Software Accomplishment Summary

Output to: Integrator

5.5.1 MCP Platform User Guide

The Hosted App Provider consumes the published MCP Platform User Guide when available from the MCP Platform Provider via the Integrator. If the Hosted Application and MCP Platform are developed concurrently, the published user guide may not be available until the MCP Platform development is complete. It should be shown that the Hosted Application complies with limitations and constraints defined in the user guide and any open problem reports against the MCP Platform are acceptable to the Hosted Application. If needed, changes should be made to the Hosted Application to comply with the user guide.

5.5.2 Software Accomplishment Summary

The Hosted App Provider completes the Software Accomplishment Summary including evidence to show compliance to the MCP Platform User Guide. The Hosted App Provider should also complete any lifecycle data needed for the Integrator which includes the required design and allocation data.

6 Integrated System Development

Per the agreed-to industry standards and Development Plan(s), the Integrator integrates the Computing System and HA System(s) into an Integrated System and produces the required lifecycle data. These sections identify additional considerations to ensure MCP Platform aspects are adequately captured.

The Integrator role provides alignment and coordination among the MCP Platform Provider and the Hosted App Providers. This is important particularly when the development involves multiple providers. The Integrator ensures data among the providers is made available and ensures that the data is consumed and understood. Figure 14 shows the breakdown of scope. Section 6 covers the aspects noted in Figure 14 which includes the aspects related to the Configuration Data Item (CDI). The CDI is the software used to configure and allocate the shared resources of the Computing System and MCP Platform.

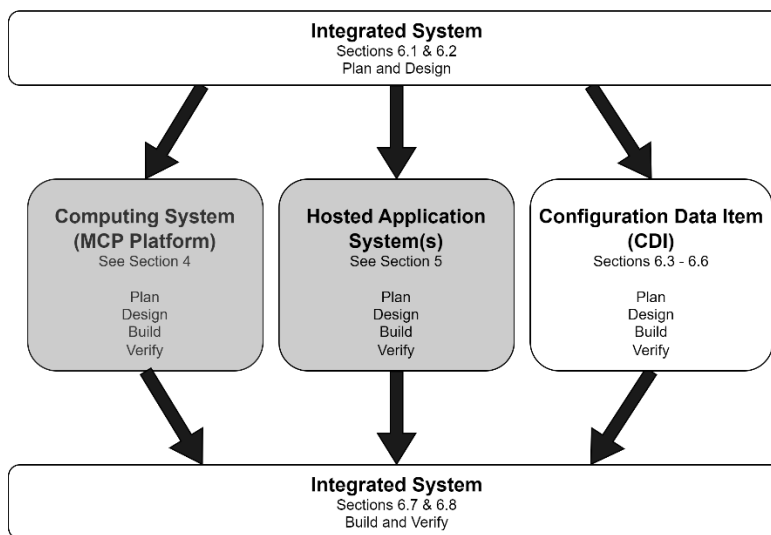


Figure 14 – Relationship of Integration

Figure 15 provides a generic diagram to define the Integrated System (yellow rectangle). In this example, the two HA Systems are combined with an allocated Computing System.

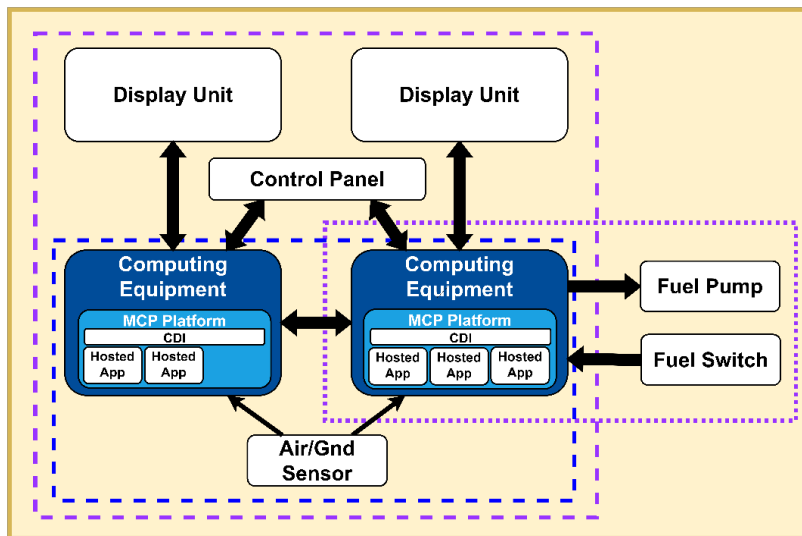


Figure 15 – Example Integrated System

The Integrator needs the details on the Computing System (MCP Platform) and the Hosted Application System to effectively plan, design, build and verify the Integrated System. The MCP Platform User Guide is the preferred method to share the pertinent details (Section 4.5.4). If this specific lifecycle data is not available because the MCP Platform development is still ongoing, the MCP Platform Provider should still provide the required data by an alternative means until the user guide is available. When “MCP Platform User Guide” is defined in following subsection, this can be replaced with an alternate means until the user guide is available.

6.1 Plan the Integrated System

Inputs: Aircraft needs and use cases

Inputs from: Aircraft Developer

The Integrator completes the following activities:

1. Consume data from Aircraft Developer and further decompose system needs and use cases (Section 6.1.1)
2. Define the Preliminary Integrated System (Section 6.1.1)

Outputs: Preliminary Integrated System Design data, Integrated System needs and use cases

Outputs to: MCP Platform Provider, Hosted App Provider

6.1.1 Preliminary Integrated System Architecture

Using inputs from the Aircraft Developer, the preliminary architecture of the Integrated System is developed. Example of inputs:

- Aircraft Functions satisfied by Integrated System
- Aircraft Use Cases and Environmental Characteristics
- Preliminary Aircraft Safety Analysis

The preliminary architecture of the Integrated System likely does not include specific details on an MCP Platform but rather functional and performance needs of a Computing System with the ability to host several applications. If the Computing Equipment or MCP Platform implementation already exist, the Integrator can include these design data and assumptions into their preliminary plans.

6.2 Design the Integrated System

Inputs: Preliminary Integrated System Architecture, Aircraft Requirements

Inputs from: Integrator, Aircraft Developer

The Integrator completes the following activities:

1. Define the Integrated System Requirements and mature the Design Data (Section 6.2.1)
 - Selection of the MCP Platform Provider
2. Complete the Preliminary System Safety Assessment

Outputs: Integrated System Requirements, Design Data, Preliminary System Safety Assessment

Outputs to: MCP Platform Provider, Hosted App Provider

6.2.1 Integrated System Requirements & Design Data

Requirement development should be at a high level of abstraction and typically does not include requirements specific to the MCP Platform. These requirements focus on the functionality and performance of the Computing System and the needs for shared resources. These requirements flow down to the MCP Platform Provider and the Hosted App Provider.

Selection of the MCP Platform Provider focuses on the provider's ability and experience with developing MCP Platforms that satisfy the requirements and objectives of the Integrated System. Considerations should include efficiency of configuring the MCP Platform, methods and/or tools to analyze allocation modifications and the impacts, and capability of characterizing interference channels. This also includes the assessment of the impacts on the integration activity due to mitigation methods available to the Integrator.

6.2.2 Preliminary Integrated System Safety Assessment

The Preliminary System Safety Assessment (PSSA) should also be at a high level of abstraction. However, the MCP Platform Provider uses the PSSA to support design decisions at the MCP Platform level. This PSSA provides important data on the aircraft functions supported by the Integrated System, the associated aircraft hazards, and the safety criticality of these functions. It also provides the expected contribution of the Computing System. It is recommended that the Integrator provide these details within the PSSA to the MCP Platform Provider.

6.3 Plan the Configuration Data Item

Inputs: MCP Platform User Guide, MCP Platform Plan

Inputs from: MCP Platform Provider

The Integrator completes the following activities:

1. Consume the MCP Platform User Guide Tool instructions and generate CDI Build Process (Section 6.3.1)
2. Develop plan and process details to support creation of CDI (Section 6.3.2)

- Document the Plan for Integrated MCP Aspects of Certification (PIMAC) using the data from the MCP Platform Plan and User Guide as well as Hosted Application Design data.

Outputs: CDI Planning document(s), PIMAC

Outputs to: Integrator

6.3.1 Defining the Configuration Data Item Build Process

As defined in Section 4.5.2, the MCP Platform Provider defines the ‘Configuration Definition’ Methodology and needed tool suite. Typically, this information is included in the MCP Platform User Guide. The Integrator uses this methodology and tool suite to define the CDI Build Process which generates the Configuration Data Item software. This includes defining the set of inputs for each Hosted Application allocation and for the Integrated System Requirements and Design Data. The output of the Build Process is the CDI data used to generate the software. Additional outputs, like allocation reports, could be needed as well. Allocation reports include analysis of successful allocation of resources, timing, margin, etc.

The Integrator should determine if additional tools are required to support their build process.

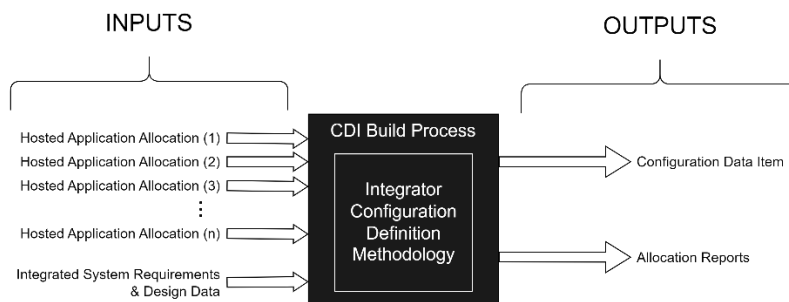


Figure 16 – Generic CDI Build Process

6.3.2 Defining the Configuration Data Item Development Process

The Integrator defines the development plan for the CDI allocating the shared resources (e.g., memory allocation, core execution time, bandwidth, or quality of service settings, etc.). The Integrator should define the process to consume the Hosted Application(s) inputs, to consume the MCP Platform inputs, to define allocation requirements and design data, to execute the Build Process, and to verify the CDI.

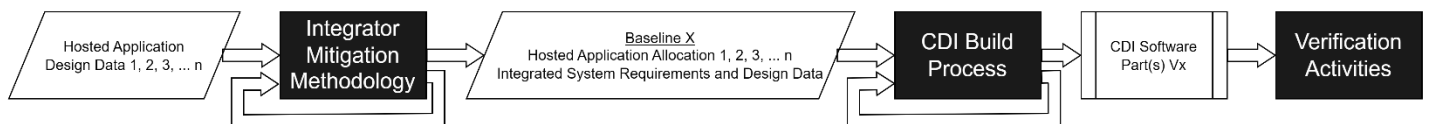


Figure 17 – Generic CDI Development Process

6.3.3 Plan for Integrated MCP Aspects of Certification (PIMAC)

The Integrator defines within the PIMAC the justification and means proposed to meet the airworthiness objectives (e.g., AC 20-193). The PIMAC delineates the different providers and proposed associated lifecycle artifacts to support the airworthiness objectives, similar to Section 11.

The MCP Platform Plan (Section 4.1.3) provides details on the MCP Platform. Due to possible configurability by the Integrator, the PIMAC must supplement the MCP Platform Plan to account for the proposed final configuration. For the example, the MCP Platform Plan may identify four active cores can be configured but the PIMAC could identify that only two cores are active in the final configuration. The following details should be included in the PIMAC or cross reference the MCP Platform Plan:

- Provide a high-level description of how MCP shared resources are used/allocated and the safety mechanisms to detect and handle MCP failures.
- Identify the specific MCP processor, including the unique identifier from the manufacturer.
- Identify the number of active cores.
- Identify the Core Software architecture.
- Identify any active dynamic features provided in software hosted on the MCP and provide a high-level description of how they are utilized.
- Identify whether or not the MCP device is used in an IMA platform to host software applications (potentially from multiple suppliers and developed independently).
- Identify whether or not the MCP Platform provides Robust Resource and / or Time Partitioning.
- Identify any active Hardware dynamic features of the MCP device and how they are utilized.
- Identify any MCP-specific methods and tools to develop and verify the MCP design.
 - Includes inter-core data coupling and control coupling among the hosted applications
 - Includes the use of data flow diagrams, control flow diagrams, data dictionaries, calling trees, traceability tagging of data flow and control flow elements, and the use of software tools that support the requirements-based test coverage, structural coverage analyses, static and dynamic analyses of the data and control coupling, as appropriate.
 - Includes the identification of test cases, scenarios, or microbenchmarks that support the evidence of the thoroughness of the requirements-based hardware/software integration testing and software integration testing for demonstrating the data and control coupling between partitions/cores.
 - Includes the identification of manual or tool-based data and control coupling measures that will be presented for the configuration data item(s).

The Integrator should define within the PIMAC the expected methods to assess change to the Integrated System (e.g., Hosted Application changes) which determine the impact to existing requirements, design, and verification data. The PIMAC should detail if Hosted Application independence is required, meaning a design that shows certain Hosted Application changes do not impact other Hosted Applications.

6.4 Design the Configuration Data Item

Inputs: PIMAC, MCP Platform Interference Analysis, MCP Platform User Guide, MCP Platform Design data and Resource Needs, Hosted Application Design data and Resource Needs, WCET data

Inputs from: Integrator, MCP Platform Provider, Hosted App Provider

The Integrator completes the following activities:

1. Review the Interference Analysis and validate the Integrator Mitigation Methodology (Section 6.4.1)
2. Develop CDI Requirements and Design data (Section 6.4.2)

Outputs: CDI Requirements and Design data

Outputs to: Integrator

6.4.1 Interference Analysis and Integrator Mitigation Methodology

To ensure the CDI design satisfies the PIMAC objectives, the Integrator should review the MCP Platform Interference Analysis (Section 4.4.2.1) showing all identified ICs and the impacts of those ICs not fully mitigated by the MCP Platform Provider. Specific mitigation details are provided by the Integrator Mitigation Methodology (Section 4.5.1). The Integrator reviews the Integrator Mitigation Methodology to ensure the mitigations are acceptable and support the deployment of the Integrated System. Modifications to the Integrator Mitigation Methodology may be required following this review.

6.4.2 Configuration Data Item Requirement and Design Data

Per the agreed-to industry standards and Development Plan(s), the allocation requirements and design data are captured by the Integrator. The Integrator receives from the MCP Platform Provider details to properly configure the Computing System. The Integrator receives Hosted App Provider details defining the Hosted Application requirements for shared resources. MCP Platform and Hosted Application configuration setting and allocation needs determine the requirements and design data developed for the CDI. Example of data types:

- Allocation of Hosted Application to each core
- Core execution time
- Fault response
- Shared IO definition and usage
- Shared Memory access needs
- Timing, latency parameters
- MCP Platform Settings (e.g., deactivation of unused function)

6.5 Build the Configuration Data Item

Inputs: CDI Requirements and Design Data, CDI Build Process, MCP Platform User Guide

Inputs from: Integrator, MCP Platform Provider

The Integrator completes the following activities:

1. Execute the Integrator Mitigations for all Hosted Applications (Section 6.5.1)
2. Execute the CDI Build Process (Section 6.5.2)

Outputs: CDI Software, Allocation Reports

Outputs to: Integrator, Hosted App Provider

6.5.1 Executing Integrator Mitigation Methodology

As detailed in the MCP Platform User Guide, the Integrator Mitigation Methodology (Section 4.5) are applied to allow for refinement of allocation for the given Hosted Application usage of the MCP Platform shared resources. The results may impact the CDI Design Data and should be updated if needed.

The Integrator should review the MCP Platform Provider open problem reports associated with the Integrator Mitigation Methodology and determine corrective action as necessary. For example, if a tool is used to define the allocation and make any refinements, there may be tool problem reports limiting the functionality of the tool.

6.5.2 Executing the Configuration Data Item Build Process

Using the Build Process defined in Section 6.3.1, the Integrator creates the CDI integrating all Hosted Application on the MCP Platform.

Tools could be utilized to input and maintain design data, to develop the time and memory allocations, to validate the allocation was successfully constructed, and to produce the physical CDI Software. See MCP Tool Section 9 for more details.

Several CDI builds may be required to mature the integration of the Computing System and HA System. Configuration control of the requirements to implementation is important to maintain. It is recommended that the Integrator, with support from MCP Platform Provider and Hosted App Provider, develop baseline test(s) to execute following a new CDI build to verify generic Integrated System behavior prior to fully performing all verification activities in Section 6.6.

6.5.3 Configuration Data Item Build Example

The Integrator likely needs to execute the CDI build process several times during the development of the Integrated System. Figure 18 shows an example of multiple scenarios that could drive new CDI Software parts. Some scenarios:

- Changes to the MCP Platform (changes in HIC mitigations)
- Changes to the Hosted Application Design Details (e.g., Additional time execution)
- Changes to the Integrated System Design (e.g., reallocation Hosted Application to different core)
- Adding new Hosted Application
- Refinement of the allocation based on results of Integrated WCET

When changes are introduced by the MCP Platform Provider or by the Hosted App Provider, the Integrator must reassess the design data, particularly the Hosted Application resource needs, and determine what activities from the CDI Build process must be executed.

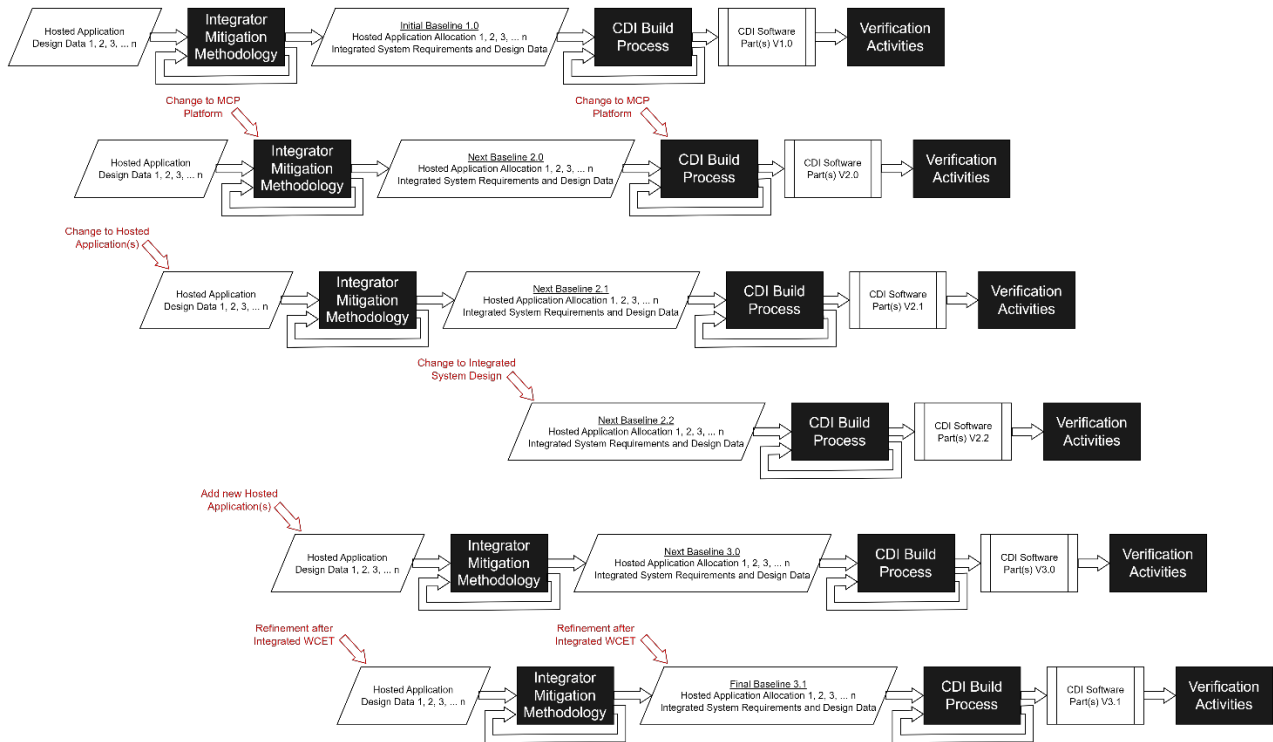


Figure 18 – Example of Multiple CDI Build Scenarios

As depicted in Figure 18, configuration control is maintained on each formal build as verification activities by the Hosted App Provider, by the Integrator, or both are completed. It is recommended that change impact assessments are completed to determine the scope of verification required on the new build.

6.6 Verify the Configuration Data Item

Inputs: CDI Software, Allocation Reports

Inputs from: Integrator

The Integrator completes the following activities:

1. Completion the verification of the CDI Software (Section 6.6.1)

Outputs: CDI Software Verification data

Outputs to: Integrator

6.6.1 Verifying the Configuration Data Item

The Integrator verifies the CDI ensuring that allocation of shared resources satisfies the defined requirements (Section 6.4.2). The Allocation Reports could provide data verifying the CDI was allocated per the requirements. Depending on the Integrator verification process, the CDI may be verified during the integrated system verification (Section 6.8).

6.7 Build the Integrated System

Inputs: CDI Software, Computing System (MCP Platform), HA System(s) (Hosted Applications)

Inputs from: Integrator, MCP Platform Provider, Hosted App Provider

The Integrator completes the following activities:

1. Integrate the Computing System and HA System(s) and define the mature configuration (Section 6.7.1)

Outputs: Integrated System, Configuration Lifecycle data

Outputs to: Integrator

6.7.1 Integrating the Computing System and Hosted Application Systems

The MCP Platform Provider and Hosted App Provider supply the Integrator implementations of the Computing System (MCP Platform) and the Hosted Application System. The Integrator integrates these components with the CDI. Several iterations may be needed to mature the Integrated System as changes are applied to the Computing System, HA System(s), and CDI. Depending on the Integrated System architecture, multiple CDI Software parts could be required to configure multiple Computing Systems/MCP Platforms.

Once the Integrator has determined the Integrated System is mature, the configuration of the system is baselined. Maturity of the Integrated System is highly dependent on the maturity of the Computing System and HA System(s).

6.8 Verify the Integrated System

Inputs: Integrated System, Final Interference Analysis, Preliminary System Safety Analysis, MCP Platform User Guide

Inputs from: Integrator, MCP Platform Provider

The Integrator completes the following activities:

1. Execute Verification Procedures and document results (Section 6.8.1)
2. Complete the final System Safety Analysis (Section 6.8.2)
3. Complete Integrated System Determinism Analysis
 - Review MCP Platform Provider data and validate documented vulnerabilities for partitioning and interference channels have been mitigated (Section 6.8.3.1)
 - Execute WCET on Integrated System and analyze performance (Section 6.8.3.2)

Outputs: Integrated System Verification data, Integrated System Safety Analysis, Integrated System Determinism Analysis, Integrated System Configuration Index

Outputs to: Integrator

6.8.1 Requirement and Robustness Verification

The Integrator develops verification test procedures and a test environment that represents the Integrated System. The test procedure(s) should be based on the system requirements and focus on verifying intended interaction of the Hosted Application with other Hosted Application and the Computing System. Using the mature implementation of the Integrated System, the Integrator executes the test procedures and obtains the verification results. The verification of inter-core data coupling and control coupling of all software applications utilizing the MCP Platform is performed on the Integrated System and considers the impacts of the Interference Channels.

Robustness testing should also be executed to ensure the Integrated System performs as expected under abnormal modes of operation (e.g., anomalous inputs, induced failures, maximum exercise of systems' function, etc.). Robustness testing should focus on heavy utilization of the shared resources. The amount of robustness testing depends on the Integrated System and the aircraft functions it supports. For example, safety critical functions should have adequate robustness testing to show behavior during the abnormal scenarios.

Unsuccessful verification results are analyzed by the Integrator and may require coordination with the MCP Platform Provider or Hosted App Provider. Design changes should be initiated if the verification result is unacceptable for airworthiness.

The Integrated System Configuration Index documents the final configuration.

6.8.2 Integrated System Safety Assessment

The Integrator ensures the safety of the Integrated System using the System Safety Assessment (SSA) to show that safety objectives are met. The Integrator utilizes safety lifecycle artifacts from the MCP Platform Provider to complete the SSA. For example, a common mode analysis needs to assess the Computing System and specifically the MCP Platform to ensure adequate independence in the system design. Additionally, the MCP Platform Provider could include system assumptions within their assessment of the MCP Platform. Typical MCP Platform Safety Assessment assumption categories:

- Aircraft/System Usage durations
- Escalation of fault reports
- Additional independent system monitors
- Hosted Application usage of shared resources

The Integrator should focus on determinism related failures defined within the MCP Platform safety lifecycle data (e.g., FHA, FMEA) ensuring the unmitigated vulnerabilities are understood. Integrator should implement the required safety nets, as needed.

6.8.3 Integrated System Determinism Analysis

The Integrator creates the Integrated System Determinism Analysis to analyze the final hardware and software configuration for determinism. This analysis utilizes the Final Interference Analysis as the foundation and continue the analysis by focusing on partitioning and interference channels of the specific configuration of the Integrated System. It also assesses the Hosted Application performance of the specific configuration of integrated Hosted Applications.

The Integrated System Determinism Analysis is considered complete when the final configuration is analyzed. If changes are made to the Integrated System, Computing System or Hosted Application System, the Integrator must assess the impact to the Integrated System Determinism Analysis and complete delta analyses as needed. It is recommended that the Integrator develop a process to assess change to the Integrated System which includes CDI changes. Complex Integrated Systems with several Hosted Applications on several cores could require additional analysis or WCET activities for changes to MCP Platform, Hosted Applications, or CDI.

6.8.3.1 *Verify Mitigations for Design Vulnerabilities, Assumptions, and Constraints*

The Final Interference Analysis (Section 4.4.2.1) concludes the design vulnerabilities related to interference channels and the mitigations intended to eliminate or reduce these vulnerabilities. The analysis defines the vulnerabilities that the MCP Platform Provider could not mitigate and must be analyzed by the Integrator. The Integrator must determine if additional design mitigations are required to satisfy airworthiness standards. Similarly, the Partitioning Analysis must review and determine if additional design mitigations are needed for partitioning vulnerabilities.

The MCP Platform User Guide must be reviewed to ensure the correct allocation of shared resources. The Integrator needs to understand the assumption and constraints levied by the user guide and implement mitigations if needed.

The System Safety Analysis and aircraft functional hazards are important to consider when unmitigated vulnerabilities are assessed by the Integrator. As an example, a non-safety critical function could have no mitigation for the vulnerability if it

doesn't impact airworthiness. Additionally, inadvertent CDI modifications to the allocation of shared resources are considered vulnerabilities and need the appropriate safety net mitigations to maintain a deterministic MCP Platform environment.

The Integrated System Determinism Analysis should contain the following:

- Identification of all unmitigated vulnerabilities and any assumptions from the Final Interference Analysis and Partitioning Analysis
- Validation of the assumptions concluding the analysis provided by the MCP Platform Provider is acceptable to use.
- Analysis of the system level impacts of the unmitigated vulnerabilities and results showing the mitigations provided by the Integrator to resolve those vulnerabilities; or justification is provided for unmitigated or partially mitigated vulnerabilities.

6.8.3.2 Final Configuration WCET

When the Integrator determines the final configuration of the system, the integrated system must complete the WCET testing and analysis to ensure that all Hosted Applications function as intended under the worst-case environment. The Integrator coordinates with the MCP Platform Provider and Hosted App Provider as needed to ensure the WCET is executed correctly. The MCP Platform User Guide contains the details and instructions related to executing the WCET testing. Clearly defined pass/fail criteria is needed, noting that non-essential Hosted Applications could have different criteria from the safety critical Hosted Applications.

Methods could differ based on the MCP Platform and providers. One method may have the Integrator execute the Final Configuration WCET for each Hosted Application in the integrated environment. This method likely requires Hosted App Provider involvement to validate that the test scenario is valid for the worst-case and confirming success. Another method may have the Integrator requiring the Hosted App Provider to complete the WCET. This requires the Integrator providing the test environment including the final CDI and conducting a post analysis of all Hosted Application WCET data results. Other methods could be developed to show the overall integrated system is built correctly.

The microbenchmarks created by the MCP Platform Provider (Section 4.3.2.2) may be utilized for this WCET activity to stress the Integrated System further verifying a Hosted Application's sensitivity to the interference channels.

The Integrated System Determinism Analysis should contain the following:

- Define the Final Configuration (or reference Configuration Index lifecycle data)
 - Computing System
 - HA System(s)
 - CDI
- Description of test environment, scenario(s), pass/fail criteria
- Reference to test procedures
- Summary of test results and measurements
- Summary of WCET data recorded by the Hosted App Provider

6.9 Output

Inputs: Integrated System Verification data, CDI Software Verification data, Integrated System Safety Analysis, Integrated System Determinism Analysis, MCP Platform Accomplishment Summary, Hosted Application Software Accomplishment Summaries

Inputs from: Integrator, MCP Platform Provider, Hosted App Provider

The Integrator complete the following activities:

1. Complete the Integrated System Accomplishment Summary (Section 6.9.1)
 - a. Defer open problem report, if acceptable

Output: Integrated System Accomplishment Summary

Output to: Aircraft Developer

6.9.1 Accomplishment Summary

The Integrator summarizes in the Integrated System Accomplishment Summary successful completion of the development. This Accomplishment Summary should also summarize the data produced by the Integrator showing the required industry and airworthiness objectives were satisfied. It is likely that the Integrator utilizes MCP Platform Provider or Hosted App Provider Lifecycle Artifacts to support compliance to industry and airworthiness objectives. When this is the case, the Integrated System Accomplishment Summary should provide a detailed mapping of the Computing System or HA System Lifecycle Artifacts and a summary of the data.

The open problem reports against the Integrated System must be reviewed and justified to remain open. The list of all justified open problem reports along the with expected system level effect are included in the Integrated System Accomplishment Summary. If an open problem report cannot be justified for deferral, the appropriate design change must be implemented to resolve the problem, and a change impact assessment is completed to determine what lifecycle artifacts are impacted by the change.

7 Aircraft Development

The Aircraft Developer plans and designs the aircraft by defining the aircraft performance and environmental attributes, installation constraints, as well as the aircraft functions. In addition to these design aspects, the Aircraft Developer coordinates with the airworthiness regulator to define the airworthiness or certification basis and the method of compliance (e.g., Advisory Circular 20-193 for MCP Platform). These requirements are provided to the Integrator for development of the Integrated System.

At the Aircraft development level there is less MCP Platform specific activities. The Aircraft Developer is cognizant of the Computing System and its usage of an MCP Platform and the shared resources as demonstration of aircraft functions rely on this knowledge. The Aircraft Developer relies on lifecycle data generated by the Integrator, MCP Platform Provider, and Host App Provider(s) to complete demonstration of compliance.

When the aircraft is available for ground/flight testing and demonstration, the Aircraft Developer should complete the following:

- Verify the Integrated System installed in the aircraft by demonstrating safe operation and intended function of the Integrated System. The Integrator provides the specific configuration of the Integrated System that completed verification (Section 6.8).
- Complete the appropriate Aircraft Safety Assessment (ASA) per the agreed-to industry standard. This ASA relies on data and analyses from the Integrated SSA which includes contributions from the Computing System and MCP Platform.

Problem reports should be created against the Integrated System for anomalies observed during ground and flight testing and for anomalies detailed from the ASA. If the problem report can remain unresolved, it must be reviewed and justified by the Aircraft Developer. The Aircraft Developer should engage the Integrator, MCP Platform Provider, and Hosted App Provider as needed. If an open problem report cannot be justified for deferral, the appropriate design change must be implemented to resolve the problem, and a change impact assessment is completed to determine what lifecycle artifacts are impacted by the change.

The Aircraft Developer generates the required aircraft lifecycle data to capture the verification and demonstration activities. The aircraft lifecycle data should reference and trace airworthiness objectives to all applicable data completing the method of compliance including data generated by Integrator, MCP Platform Provider, and Host App Provider(s).

8 MCP Platform – Progress Oversight

It is recommended to plan periodic activities to ensure the development is progressing to meet the required objectives. The following oversight activities are typical for product development, and specific MCP topics/criteria are provided to help evaluate that the MCP design and implementation are on target to meet MCP objectives. It is acceptable to combine these reviews with other technical reviews.

8.1 Detailed Design Reviews

As requirements and design are baselined, detailed architecture reviews are recommended prior to implementation for each domain: Integrated System, Computing System, Hosted Application System.

Each review should include the Airworthiness Regulator, Integrator, MCP Platform Provider, and Hosted App Provider. The focus of the design review ensures progress of the system, hardware, and software development to meet the agreed-to objectives and to coordinate with other teams. The list below defines the required lifecycle data for the review. If data is not available, it is up to the team's discretion on holding the review and what delta reviews are required when the data is available.

Important aspects to cover:

- Defined system architecture is understood with validated requirements and with any defined assumptions
- Parent requirements and other inputs are available or alternative plans are understood
- Integration details and dependencies
- MCP Platform Topics:
 - MCP Platform data is available (defined in lists below)
 - Critical Configuration Registers are identified, and settings are defined in requirements
 - Interference Channels with potential impactful effects are identified
 - Design of safety nets and mitigations are proposed and potential constraints to levy on Integrator
 - Method(s) for MCP Platform development are defined
 - Assumptions on MCP Platform usages at Integrated System level
 - Shared resources and allocation needs
 - Assessment of derived requirements related to the MCP architecture

Lifecycle Data for Computing System/MCP Platform Review

Unless noted otherwise, all data should be under configuration control per the agreed-to industry standards and Development Plan(s).

- Computing System:
 - System, Hardware, and Software Development Plans
 - MCP Platform Plan
 - Requirements
 - Computing System
 - MCP Platform
 - Core Hardware
 - Core Software (RTOS)
 - Detailed Design Data
 - Preliminary Interference Analysis
 - Preliminary MCP Platform Safety and Partitioning Analyses

- Other data items:
 - Parent Requirements (For Reference)
 - Integrated System Design Data and Hosted Application Use Cases (For Reference)
 - Preliminary Integrated System Safety Assessment (For Reference)

Lifecycle Data for Hosted Application System Review

- Hosted Application System:
 - System, Hardware, and Software Development Plans
 - Requirements
 - Hosted Application System
 - Hardware (if applicable)
 - Hosted Application Software
 - Detailed Design Data
 - Preliminary System Safety Assessment
- Other data items:
 - Parent Requirements (For Reference)
 - Integrated System Design Data (For Reference)
 - Preliminary Integrated System Safety Assessment (For Reference)
 - MCP Platform Design Data, User Guide (For Reference)

Integrated System

- Integrated System:
 - System and CDI Development Plans
 - Requirements
 - Integrated System
 - CDI
 - Detailed Design Data
 - Preliminary Integrated System Safety Assessment
 - Hosted Application Use Cases
- Other data items:
 - Parent Requirements (For Reference)
 - MCP Platform Design Data, User Guide (For Reference)
 - Preliminary Interference Analysis (For Reference)
 - Preliminary MCP Platform Safety and Partitioning Analyses (For Reference)

8.2 Verification Readiness Reviews

When implementation is mature for the Computing System and MCP Platform (Section 4.3), it is recommended that the MCP Platform Provider hold reviews on the readiness of starting verification.

The review should include the Airworthiness Regulator, Integrator and MCP Platform Provider. The focus of the verification review ensures progress of the system, hardware, and software development to meet the agreed-to objectives and to coordinate with other teams. The list below defines the required lifecycle data for the review. If data is not available, it is up the team's discretion on holding the review and what delta reviews are required when the data is available.

Important aspects to cover:

- Defined implementation is mature based on prototype build(s)
- Parent requirements are stable
- MCP Platform data is available (defined in lists below)
- Open Problem Reports related to Interference Channels and justification for proceeding to verification

Lifecycle Data for Computing System/MCP Platform Review

Unless noted otherwise, all data should under configuration control per the agreed-to industry standards and Development Plan(s).

- Computing System:
 - System, Hardware, and Software Development Plans
 - MCP Platform Plan
 - Requirements and Traceability
 - Detailed Design Data
 - Preliminary Interference Analysis
 - Preliminary MCP Platform Safety and Partitioning Analyses
 - Verification Plans and Procedures
 - MCP Platform Configuration Index
 - Verification Environment Drawings
- Other data items:
 - Parent Requirements (For Reference)
 - Integrated System Design Data and Hosted Application Use Cases (For Reference)
 - Preliminary Integrated System Safety Assessment (For Reference)

8.3 Completion Reviews

When the Computing System and MCP Platform has been successfully verified (Section 4.4), it is recommended that the MCP Platform Provider hold reviews to confirm completeness of the development.

The review should include the Airworthiness Regulator, Integrator and MCP Platform Provider. The focus of the completion review ensures the system, hardware, and software development meet the agreed-to objectives by reviewing the data included in the MCP Platform Accomplishment Summary. The list below defines the required lifecycle data for the review.

Important aspects to cover:

- Defined implementation is mature
- Parent and Computing System requirements are stable
- MCP Platform data is available (defined in lists below)
- Open Problem Reports and justification for deferral

Lifecycle Data for Computing System/MCP Platform Review

Unless noted otherwise, all data should under configuration control per the agreed-to industry standards and Development Plan(s).

- Computing System:

- Verification data
- Final Interference Analysis
- Final MCP Platform Safety and Partitioning Analyses
- MCP Platform Configuration Index
- Draft MCP Platform Accomplishment Summary
- Draft MCP Platform User Guide

8.4 Quality and Process Assurance

Per the agreed-to Development Plans and Configuration Management Plans, there should be process(es) defining quality and process assurance activities. These activities can be critical when multiple handoffs of lifecycle data and integration of products. Early and frequent reviews should be considered particularly in the MCP development and integration.

9 MCP Tools

It is expected that systems, hardware, and software development tools typically used for airborne systems are also employed while developing systems that utilize multi-core processors. Existing guidance for use of these tools applies to MCP-based systems. However, it is also likely that new or different types of tools may be useful when complying with this guidance document. This section provides areas for consideration relating to tools explicitly used for MCP-based systems.

9.1 Types of Tools

There are several different types of tools that may be beneficial in performing analyses on MCPs. Tools should be selected based on use cases and should have a clearly defined purpose. Tools planned for use should be identified in the project planning documentation that aligns with where the tool is used (e.g. an interference generator tool should be listed in the MCP Platform Plan). Actual tool use during project execution should be captured in the associated Accomplishment Summary data.

The following list provides some example tool types but is not an exhaustive list.

- Interference generators
- Performance monitors
- Test automation
- Data analyzers
- Shared Resource allocation

9.2 Procurement or Development of Tools

A key consideration for the selection of tools is whether COTS tools is used or custom tools is created either from scratch or by modifying a COTS tool. The use of COTS tools is acceptable, provided their capabilities align with the defined purpose and use cases identified for the project, and there is adequate verification evidence that they meet their intended function. If tool development or customization is chosen, it is recommended that requirements for the tool be documented to ensure alignment with the purpose of the tool.

Regardless of procurement style, tools used should be validated against the stated purpose and use case for the tool. Each tool should also be verified to meet its intended function, preferably via requirements-based verification. Tool verification evidence should be identified in the relevant Accomplishment Summary.

9.3 Tool Qualification

When tools are used to replace an activity that is used for certification objective compliance, a tool assessment should be completed and determine the appropriate level of tool qualification (e.g. DO-330 [11]) depending on the agreed-to industry standards and Development Plan(s). Regardless of any tool qualification process requirements, all tools used to ensure safe use of an MCP should be verified to meet their intended function as employed for multi-core analysis.

9.4 Tool Configuration Management & Maintenance

Tools used for critical MCP systems development and verification should be uniquely identifiable and version controlled. Industry standard hardware or software configuration management practices are expected for MCP-related tools. This applies to both COTS tools as well as custom tools developed for a given MCP-related use case.

Throughout the life cycle of the MCP development and usage, tools should be monitored for expected operation. Any observed anomalies should be documented in a problem report (PR). The project team should periodically review any open PRs and properly disposition them. Any PRs that result in a change to the tool should also cause a version change with a new unique identifier. If a COTS tool is used, the vendor should be contacted periodically to obtain a current Open

PR listing. If a vendor declines to resolve a PR to the satisfaction of the project team, an alternative tool or other mitigation may be necessary.

The version of all tools used in final development or verification process steps should be recorded with the associated results or applicable Accomplishment Summary.

10 Multi-Core and IMA Considerations

Integrated Modular Avionics (IMA) provides a partitioned processing environment which allows applications performing aircraft functions to share hardware resources. Three documents primarily drive IMA solutions: DO-297 [5], TSO-C153a [9], and TSO-C214 [10]

DO-297 defines six tasks with objectives to support the incremental acceptance of IMA building blocks in the certification process as well as defining the roles and responsibilities for each task. Table 5 describes these six tasks and correlates the tasks to the sections of this document. As shown, there is alignment between an MCP Platform development and an IMA development.

Table 5 – MCP Guidance Support for DO-297 Tasks

DO-297 Task	Guidance Support	How
Task 1: Module acceptance	Section 4	Task 1 objectives include IMA planning, specification, and verification, etc.
Task 2: Application software or hardware acceptance	Section 5	Task 2 objectives include verifying the application uses the platform resources in accordance with the IMA platform’s users guide, etc.
Task 3: IMA system acceptance	Section 6	Task 3 objectives include verifying proper interaction, demonstrating the configuration is correct, and performing integration activities on the IMA system, etc.
Task 4: Aircraft integration of IMA system – including Validation and Verification (V&V)	Section 7	Task 4 objectives include the evaluation of specific anomaly repercussions, address failure modes, perform V&V, etc.
Task 5: Change of modules or applications	Section 4-7	Task 5 includes integrating the changed component into the IMA system, perform configuration control, etc.
Task 6: Reuse of modules or applications	Section 4-7	Task 6 includes analyzing suitability of the reused module or application, evaluate open problem reports, integrate the module or application, etc.

TSO-C153a provides a certification basis for IMA platforms and/or modules associated with Task 1. The TSO provides a set of requirements for classes of intended functions. An IMA platform and/or module realizes one or more intended function class(es) which then forms the set of requirements applicable to that IMA solution. Both hardware and software aspects are addressed within each of the intended function classes. There are seven intended function classes: Rack Housing (RH), Processing (PR), Graphical Processing (GP), Data Storage (DS), Interface (IF), Power Supply (PS), Display Head (DH). Note, communication busses, such as PCIe, I2C, etc. require the IF class per the TSO standard.

In support of Task 2, TSO-C214, Functional TSO Equipment using a TSO-C153a platform or module, is the minimum performance standards equipment must meet when the aircraft function is implemented on one or more TSO-C153a authorized modules. Another TSO standard defines the minimum operations performance specifications for the aircraft function.

When there is an intent to certify an MCP Platform as an IMA resource, the Interference Analysis needs to consider the impact of interference channels on deterministic IMA partitioning. Deterministic IMA partitioning applies the foundational elements of TDMA scheduling and hardware MMU to isolate applications' execution environment. Other processing resources may also require partitioning mechanisms to support aircraft installation of the IMA platform.

Partitioning, as referenced within Section 4, implies a range of determinism and its effectiveness may or may not support what's needed for an IMA platform. There may be assumptions or limitations in Hosted Application deployment implied by the robustness of the partitioning mechanisms regardless of the MCP Platform's item design assurance level. Fairness algorithms that limit access to MCP Platform resources should be carefully reviewed for exactly how and when they can impact a Hosted Application's execution environment. The ability to characterize and analyze the MCP Platform is key to enabling all cores to host applications, enabling multiple applications on one core, and/or enabling hosted apps of different item design assurance levels to execute on the same shared environment.

The MCP Platform Provider and/or Integrator should ensure the use of an MCP Platform as the underlying technology for an IMA Platform limits the impacts it has on Hosted Applications or adapt the system design to operate within the limitations of an MCP Platform solution.

11 Advisory Circular 20-193 Mapping

Table 6 maps the AC 20-193 objectives [1] to the applicable sections of this document and the Lifecycle Artifacts used to satisfy the objective. As mentioned in the Introduction, it is not mandatory to use these titles as written and data could be packaged differently to meet the same intent. The planning document(s) and accomplishment summary document(s) should map the applicable Lifecycle Artifacts to the objectives.

Table 6 also provides applicability of each role with the following definitions:

- Prime: Responsible to develop lifecycle artifacts to satisfy objective. May receive data from support role.
- Joint: Two or more roles are responsible to develop lifecycle artifacts to full satisfy objective.
- Support: Provides required data to Prime or Joint roles.
- Overseer: Reviews data as needed to support showing of compliance to objectives.

Table 6 – Advisory Circular 20-193 Mapping

AC 20-193 Objective	Objective Text	Applicable Section	Lifecycle Artifact	MCP Platform Provider	Hosted App Provider	Integrator	Aircraft Developer
------------------------	----------------	--------------------	--------------------	-----------------------------	------------------------	------------	-----------------------

<p>MCP_Planning_1</p>	<p>The applicant's plans or other deliverable documents:</p> <ol style="list-style-type: none"> 1. Identify the specific MCP processor, including the unique identifier from the manufacturer. 2. Identify the number of active cores. 3. Identify the MCP software architecture to be used and all the software components that will be hosted on the MCP. 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. 5. Identify whether the MCP will be used to host software 	<p>4.1.3</p> <p>6.3.3</p>	<p>MCP Platform Plan</p> <p>PIMAC</p>	<p>Joint</p>	<p>Support</p>	<p>Joint</p>	<p>Overseer</p>
------------------------------	--	---------------------------	---------------------------------------	--------------	----------------	--------------	-----------------

	<p>applications from more than one system, and whether it will be used in an integrated modular avionics (IMA) platform.</p> <p>6. Identify whether the MCP platform will provide robust resource partitioning and / or robust time partitioning as defined in this document.</p> <p>7. Identify the methods and tools to be used to develop and verify all the individual software components hosted on the MCP so as to meet the objectives of this document and the applicable software guidance, including any methods or</p>						
--	---	--	--	--	--	--	--

AC 20-193 Objective	Objective Text	Applicable Section	Lifecycle Artifact	MCP Platform Provider	Hosted App Provider	Integrator	Aircraft Developer
	tools needed due to the use of an MCP or the selected MCP architecture.						

AC 20-193 Objective	Objective Text	Applicable Section	Lifecycle Artifact	MCP Platform Provider	Hosted App Provider	Integrator	Aircraft Developer
	<p>be used to support the objectives in this AC.</p> <p>3. Identify any hardware dynamic features of the MCP that will be active and provide a high-level description of how they will be used.</p> <p>4. Identify the aspects of the use of the MCP that may require a safety net or other mechanisms to detect and handle failures in the MCP.</p>						

AC 20-193 Objective	Objective Text	Applicable Section	Lifecycle Artifact	MCP Platform Provider	Hosted App Provider	Integrator	Aircraft Developer
MCP_Resource_Usage_1	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.	4.1.2.1.1 4.2.3.1 4.4.2.1.1	Interference Analysis	Prime	Support	Overseer	Overseer
MCP_Resource_Usage_2	Reserved. Covered by AC 20-152A objective COTS-8 Objective COTS-8: "If the complex COTS device contributes to DAL A or B functions, the applicant should develop and verify a means that ensures an appropriate mitigation is specified in the event of any inadvertent alteration of the "critical configuration settings" of the COTS device."	4.1.2.1.1 4.2.3.1 4.4.2.1.1	Interference Analysis				
	The applicant:						

AC 20-193 Objective	Objective Text	Applicable Section	Lifecycle Artifact	MCP Platform Provider	Hosted App Provider	Integrator	Aircraft Developer
MCP_Resource_Usage_3	has identified the interference channels that could permit interference to affect the software applications hosted on the MCP cores	4.1.2.1.2 4.2.3.2 4.4.2.1.2	Interference Analysis	Prime	Support	Overseer	Overseer
	and has verified the applicant's chosen means of mitigation of the interference.	4.1.2.1.4 4.2.3.4 4.4.2.1.3					
MCP_Resource_Usage_4	The applicant:						
	has identified the available resources of the MCP and of its interconnect in the intended final configuration,	6.3.3	PIMAC	Support	Support	Prime	Overseer
	has allocated the resources of the MCP to the software applications hosted on the MCP,	6.4.2	CDI Requirements and Design data				
		6.5.2	CDI Software, Allocation Reports				
	6.6.1	Verification data					

AC 20-193 Objective	Objective Text	Applicable Section	Lifecycle Artifact	MCP Platform Provider	Hosted App Provider	Integrator	Aircraft Developer
	and has verified that the demands for the resources of the MCP and of the interconnect do not exceed the available resources when all the hosted software is executing on the target processor.	6.8.1 6.8.3	Verification Data Integrated System Determinism Analysis				

<p>MCP_Software_1</p>	<p>The applicant has verified that all the software components hosted by the MCP meet the objectives of the applicable software guidance. In particular, the applicant has verified that all the hosted software components function correctly and have sufficient time to complete their execution when all the hosted software and hardware of the MCP is executing in the intended final configuration.</p> <p>The way in which the applicant should satisfy this objective depends on the type of the MCP platform:</p> <ul style="list-style-type: none"> ▪ <i>MCP platforms with robust partitioning:</i> Applicants who have verified that their MCP platform provides both robust resource partitioning and robust time partitioning (as defined in this document) may verify software 	<p>4.4</p> <p>5.4</p> <p>6.8</p>	<p>Verification Data, Core Software WCET Data</p> <p>Verification Data, Hosted Application WCET Data</p> <p>Verification data, Integrated System Safety Analysis, Integrated System Determinism Analysis</p>	<p>Support</p>	<p>Joint</p>	<p>Joint</p>	<p>Overseer</p>
------------------------------	--	----------------------------------	--	----------------	--------------	--------------	-----------------

AC 20-193 Objective	Objective Text	Applicable Section	Lifecycle Artifact	MCP Platform Provider	Hosted App Provider	Integrator	Aircraft Developer
	applications separately on the MCP and determine their WCETs separately.						

	<ul style="list-style-type: none">▪ All other MCP platforms: Applicants may verify separately on the MCP any software component or set of requirements for which the interference identified in the interference analysis is mitigated or is precluded by design. Software components or sets of software requirements for which interference is not avoided or mitigated should be tested on the target MCP with all software components executing in the intended final configuration, including robustness testing of the interfaces of the MCP. The WCET of a software component may be determined separately on	N/A – This guidance document supports robust partitioning of MCP Platforms.	N/A	N/A	N/A	N/A	N/A
--	---	---	-----	-----	-----	-----	-----

AC 20-193 Objective	Objective Text	Applicable Section	Lifecycle Artifact	MCP Platform Provider	Hosted App Provider	Integrator	Aircraft Developer
	<p>the MCP if the applicant shows that time interference is mitigated for that software component; otherwise, the WCET should be determined by analysis and confirmed by test on the target MCP with all the software components executing in the intended final configuration.</p>						

AC 20-193 Objective	Objective Text	Applicable Section	Lifecycle Artifact	MCP Platform Provider	Hosted App Provider	Integrator	Aircraft Developer
MCP_Software_2	The applicant has verified that the data and control coupling between all the individual software components hosted on the same core or on different cores of the MCP has been exercised during software requirement-based testing, including exercising any interfaces between the software components via shared memory and any mechanisms to control the access to shared memory, and that the data and control coupling is correct.	4.4.1 5.4.1 6.8.1	Core Software Verification Data Hosted Application Verification data Integrated System Verification data	Support	Joint	Joint	Overseer
MCP_Error_Handling_1	The applicant: has identified the effects of failures that may occur within the MCP	4.1.2	Safety Analysis, Partitioning Analysis, Interference Analysis	Prime	Support	Overseer	Overseer

AC 20-193 Objective	Objective Text	Applicable Section	Lifecycle Artifact	MCP Platform Provider	Hosted App Provider	Integrator	Aircraft Developer
	and has designed, implemented, and verified means commensurate with the safety objectives, by which to detect and handle those failures in a fail-safe manner that contains the effects of any failures within the equipment in which the MCP is installed. These means may include a "safety net" independent from the MCP.	4.2.1 4.4 6.8.2	Validated requirements Safety Analysis, Interference Analysis Integrated System Safety Analysis	Joint	Support	Joint	Overseer
MCP_Accomplishment_Summary_1	In addition to providing the information requested by the applicable software and AEH guidance, the applicant has provided documentation that summarizes how they have met each of the objectives of this document.	4.5.3 5.5.2 6.9.1	MCP Platform Accomplishment Summary Software Accomplishment Summary Integrated System Accomplishment Summary	Joint	Joint	Joint	Overseer

Appendix A Integrated System Development Example

The following example is intended to further expand on the provided guidance. This example focuses on the Systems design and implementation activities to support multiple applications hosted on an MCP Platform.

A.1 Integrated System Design and Use Cases

The Integrator designed an Integrated System as shown in Figure 19 by developing the following system use cases. Requirements were captured and provided to the MCP Platform Provider and the Hosted App Providers. Hosted Applications 1, 2, and 3 are developed by one Hosted App Provider while Hosted Application 4 is developed by a different provider.

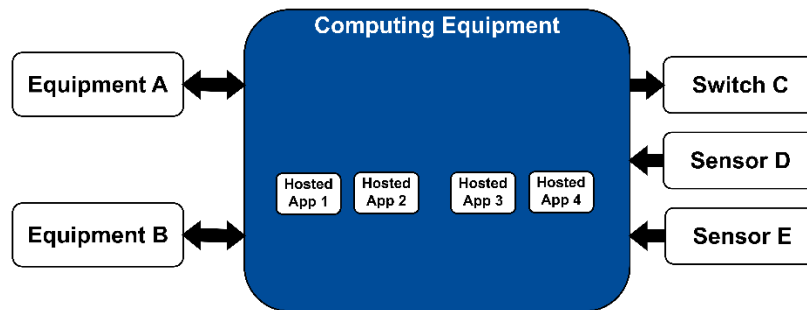


Figure 19 – Integrated Systems Example: System Design, Preliminary

Integrator's System Use Cases

- Equipment A send commands to change the position of Switch C.
- Sensor D and Sensor E monitor the environment controlled by Switch C.
- Equipment B receives the monitored data to confirm switch C position matches the commanded position.
- The Computing Equipment interfaces with Equipment A, Equipment B, Switch C, Sensor D, and Sensor E and Hosted Applications perform the required functions to drive and monitor Switch C position.
 - Hosted App 1 is a safety critical application that receives data from Equipment A and uses logic to change the position of Switch C with a system latency of 1 second.
 - Hosted App 2 receives the analog inputs from Sensor D and E, converts the data to engineering units, and stores the data in memory at 40 Hz.
 - Hosted App 3 monitors the converted Sensor D and Sensor E data, monitors Equipment A data, and uses logic to transmit to Equipment B to inform that Switch C is in the correct commanded position. The data is monitored at 1 Hz and transmitted to Equipment B every 5 seconds.
 - Hosted App 4 receives fault data from Equipment A and Equipment B and stores the data. Hosted App 4 also records fault data of the Computing Equipment. Fault data is recorded at 1 Hz.

A.2 MCP Platform Interference Analysis and User Guide

The MCP Platform Provider developed an MCP Platform which contains four cores, partitioned real-time operating system (e.g., an ARINC 653 RTOS), and the required peripherals to support the Integrator's requirements. The MCP Platform is designed to satisfy robust partitioning (space and time) allowing the hosted applications to be different criticality.

The MCP Platform Provider completed the Interference Analysis.

- Critical Configuration Settings are identified, and mitigations are captured in requirements and verified.
- IC Diagrams are constructed for all interference channels with safety nets implemented as directed by the MCP Platform Safety process. The analysis shows several hardware interference channels which the MCP Platform Provider mitigates. There are 3 HICs identified as CONTENTION_A, CONTENTION_B, and CONTENTION_C that require Integrator Mitigation.
- The Partitioning analysis is complete for the shared resources and mitigates the identified partitioning vulnerabilities.
- Microbenchmarks and test applications are utilized to understand the possible impacts and effects of shared resource contention.
- The analysis identifies the MCP Platform mitigation requirements tracing to the verification data.

The MCP Platform Provider develops the following Integrator Mitigation Methodology to address the three HICs. The methodology has the Integrator or Hosted App Provider execute WCET to determine the required time allocation. This data is then used to add margin to the execution allocation to account for the three contention channels. The allocation design of the integrated system is reviewed and refinements are made based on the engineering assessment. The MCP Platform Provider provides a refinement option to modify the allocation margin. The MCP Platform Provider indicates that additional mitigations for CONTENTION_C include limiting the number of Hosted Applications that utilize the shared resource. These mitigations are included in the MCP Platform User Guide along with instruction on MCP Platform Provider tools to utilize for the allocation process.

Integrator Mitigation Methodology:

Required inputs:

1. HOSTED-APPx_BASELINE_WCET – The recorded WCET of Hosted Application X in an integrated environment (no microbenchmarks)
2. HOSTED-APPx_USAGE – Hosted Application Design detail on whether the Hosted Application X utilizes the specific shared resource on the MCP Platform.
3. ADD_MARGIN – The amount of additional margin based on engineering assessment of the allocation design.

Output:

1. TOTAL_HAx_ALLOCATION – Defined maximum time allocation for Hosted Application X used in the integrated MCP Platform.

Hosted Application Allocation Equations:

The first equation calculates the total applicable measured contention from the MCP Platform. The Hosted Application Design Data determines if CONTENTION_C is used in the calculation.

$$\text{APPLICABLE_CONTENTION} == [\text{CONTENTION_A}] + [\text{CONTENTION_B}] + [\text{CONTENTION_C, if HOSTED-APPx_USAGE = "yes"}]$$

The second equation calculates the Total Hosted Application X Allocation by adding the Total Contention Margin with the Hosted Application X Baseline WCET data.

$$\text{TOTAL_HAX_ALLOCATION} == [\text{HOSTED-APPx_BASELINE_WCET}] + [\text{APPLICABLE_CONTENTION}]$$

The TOTAL_HAX_ALLOCATION is analyzed to determine acceptability of allocated resources. If acceptable, no further action is taken. If unacceptable, margin is added at the discretion of the Integrator. The Integrator can also modify the Hosted Application design to limit impacts of CONNECTION_C.

1. Adding execution margin:

Determine the amount of margin to add and calculate the new Total Hosted Application X Allocation.

$$\text{TOTAL_HAX_ALLOCATION} == [\text{HOSTED-APPx_BASELINE_WCET}] + [\text{APPLICABLE_CONTENTION}] + [\text{ADD_MARGIN}]$$

2. Limiting impacts of CONNECTION_C:

Determine if the Hosted Application requires the shared resource associated with CONNECTION_C and determine if alternate designs can limit its usage.

A.3 Integrated System Configuration Data Item

The Integrator develops the CDI Development Process using the User Guide and any additional coordination with MCP Platform Provider. The process collects the Hosted Application WCET data from each Hosted App Provider, execute the Allocation Equations from the MCP Platform User Guide, review the Allocation Report to determine successfully allocation, and make changes as needed.

The Integrator executes their process to complete the “Integrator Mitigation Methodology”. The Integrator tasks each Hosted App Provider with executing the WCET for each Hosted Application. Per instructions from MCP Platform Provider, this WCET does not include Microbenchmarks.

- Hosted App 1:
 - HOSTED-APPx_BASELINE_WCET – 20ms
 - HOSTED-APPx_USAGE – Shared Resource A, B, & C
- Hosted App 2:
 - HOSTED-APPx_BASELINE_WCET – 11ms
 - HOSTED-APPx_USAGE – Shared Resource A, B, & C
- Hosted App 3:
 - HOSTED-APPx_BASELINE_WCET – 25ms
 - HOSTED-APPx_USAGE – Shared Resource A, B, & C
- Hosted App 4:
 - HOSTED-APPx_BASELINE_WCET – 8ms
 - HOSTED-APPx_USAGE – Shared Resource A & B

From User Guide, MCP Platform Provider defined the contention as:

- Contention A – 2.5 ms

- Contention B – 1.5 ms
- Contention C – 3 ms

Integrator executes the equations for each Hosted Application.

- Hosted App 1 = 27ms (20ms + (2.5ms + 1.5ms + 3ms))
- Hosted App 2 = 18ms (11ms + (2.5ms + 1.5ms + 3ms))
- Hosted App 3 = 32ms (25ms + (2.5ms + 1.5ms + 3ms))
- Hosted App 4 = 12ms (8ms + (2.5ms + 1.5ms))

Integrator performs design reviews of the allocation design. Given Hosted App 1 is safety critical, the Integrator decides to increase the allocation which reduces future processing margin of the Core but provides better performance for Hosted App 1 functionality. The additional margin supports future growth or change to Hosted App 1. Therefore, Hosted App 1 is recalculated to add 1 ms. The other applications are determined to be ‘no change’.

- Hosted App 1 = 28ms (20ms + (2.5ms + 1.5ms + 3ms) + 1ms)

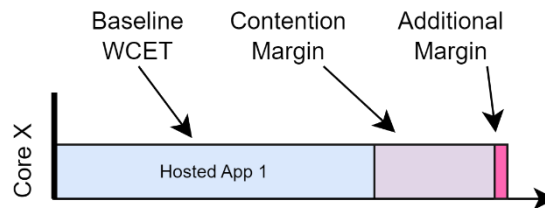


Figure 20 – Integrated Systems Example: Refinement of the Allocation

When the “Integrator Mitigation Methodology” process is complete, the Integrator uses this data to start the CDI Build process. Integrator reviews the required allocation details and defines design of the Integrated System and specifically the design of the Hosted Application. For this example, the Integrator has defined the following:

- To reduce the possibility of hardware interference, only Core0 and Core1 are utilized. System requirements defined to disable Core2 and Core3 and allocated to the CDI Software.
- Hosted App 1 and Hosted App 3 require separate cores for execution to ensure safety objectives are met. System requirement defined to separate the allocation of Hosted App 1 and 3 and allocated to the CDI Software.

Integrator executes the CDI Build process from the inputs noted above (Hosted App Allocation data and Integrated System Requirements). The following CDI design is created:

- Hosted App 1 and 2 are allocated to Core0 and Hosted App 3 and 4 are allocated to Core1 because Hosted App 1 and 3 must be hosted on separate cores.
 - Core0 allocated 46ms (28ms + 18ms) of the 50ms major frame.
 - Core1 allocated 44ms (32ms+ 12ms) of the 50ms major frame.

Using the MCP Platform provided tools, the overall allocation (execution time, memory, etc.) is defined. The allocation report depicts the Hosted Application execution design as shown in Figure 21.

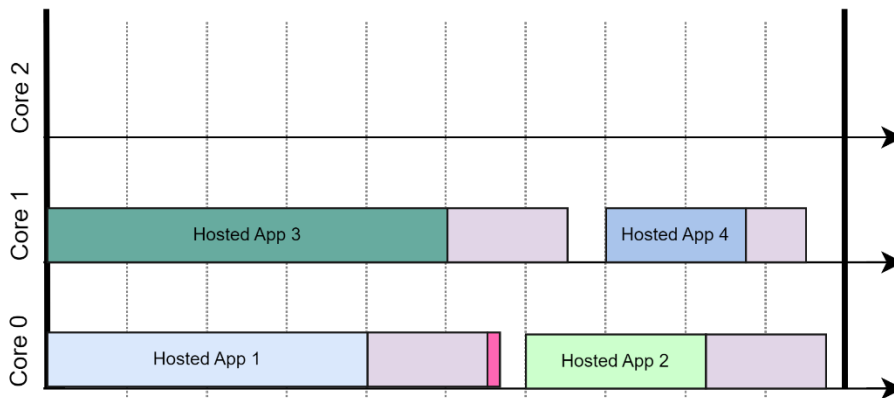


Figure 21 – Integrated Systems Example: Hosted App Execution Design, Preliminary

Integrator performs an engineering assessment of the allocation report following the CDI build. The report shows potential contention during certain modes of operations of Hosted App 1 and 3, raising concerns of missing a deadline. Integrator discusses the results with the MCP Platform Provider, and the following refinements are implemented:

- The MCP Platform has synced cores and therefore Hosted App 4 is scheduled prior to Hosted App 3 to stagger when Hosted App 1 and 3 execute.
- To mitigate CONTENTION_C, Hosted App 5 is created to manage the shared resource for all Hosted Applications. Hosted App 1, 2, and 3 interface with Hosted App 5 to access the shared resource. Hosted App 5 executes on Core 2
- Modify allocation of Core 1 to better support Hosted App 3:
 - To provide better performance of Hosted App 3 functionality, 1 ms is added to the allocation margin. The additional margin supports future growth or change to Hosted App 3.
 - Given Hosted App 4 is non-critical, Hosted App 4 allocation is reduced allowing for reserve future processing margin. Therefore, Hosted App 4 allocation is recalculated to remove 2 ms.

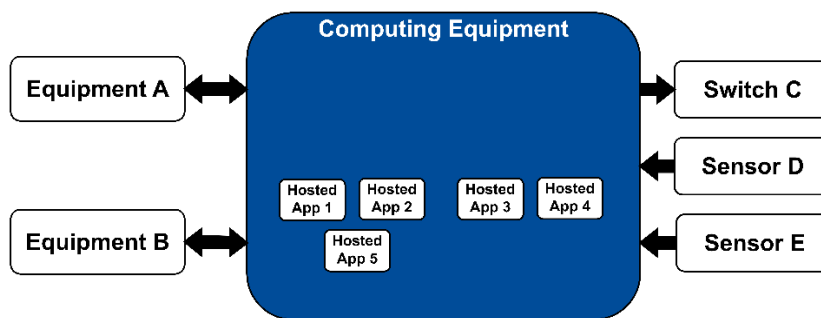


Figure 22 – Integrated Systems Example: System Design, Final

The Integrator updates the System Requirements.

- New requirement to schedule Hosted App 1 and 3 to stagger the execution times.
- New requirements to define Hosted App 5 allocation on Core 2.

Integrator executes the equations for each Hosted Application removing CONTENTION_C.

- Hosted App 1 = 25ms (20ms + (2.5ms + 1.5ms)) + 1ms
- Hosted App 2 = 15ms (11ms + (2.5ms + 1.5ms))
- Hosted App 3 = 30ms (25ms + (2.5ms + 1.5ms)) +1 ms
- Hosted App 4 = 10ms (8ms + (2.5ms + 1.5ms)) -2ms
- Hosted App 5 = 15ms (11ms + (2.5ms + 1.5ms))

Integrator executes the CDI Build process again using the new the inputs.

- New Allocations are as follows:
 - Core0 allocated 40ms (25ms + 15ms) of the 50ms major frame.
 - Core1 allocated 40ms (30ms+ 10ms) of the 50ms major frame.
 - Core2 allocated 15ms of the 50ms major frame

The allocation report depicts the new allocation design.

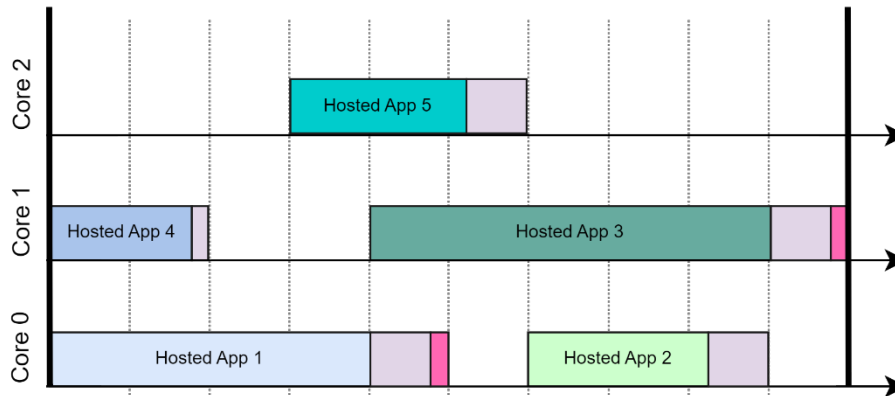


Figure 23 – Integrated Systems Example: Hosted App Execution Re-Design

Using the new CDI software, Integrator performs the WCET using the provided microbenchmarks. Given the data coupling with Hosted App 5 and the other Hosted Application, the Integrator completes specific verification activities on the inter-core dependencies. This verification shows unacceptable results and modifications are needed to ensure Hosted App 5 has additional execution time.

The following refinements are made:

- To mitigate data coupling issues, Hosted App 5 execution time is expanded in a second allocation.

The Integrator updates the System Requirements.

- New requirement to schedule two instances for Hosted App 5.

Integrator executes the CDI Build process again using the new the inputs. The allocation report depicts the new allocation design.

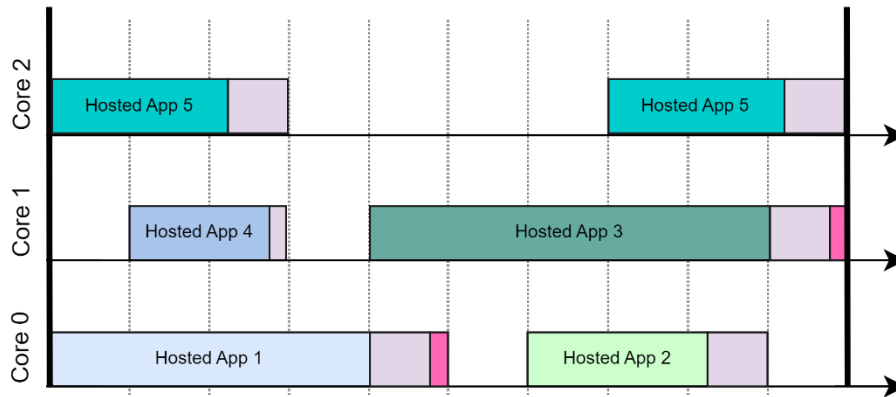


Figure 24 – Integrated Systems Example: Hosted App Execution Re-Design 2

With this final configuration, the Integrator performs the WCET using the provided microbenchmarks. The Integrator also completes data and control coupling verification activities. The results are determined to be successful and included in the accomplishment summary as well as the required data to complete the airworthiness objectives.

A.4 Integrated System Change - Add new Hosted Application

After the initial development, the Integrator determines a self-test is required to ensure reliability. To accomplish this new function the Integrator decides to add Hosted App 6. The Integrator develops the design and requirements for the following System Use case:

- Hosted App 6 executes a self-test by toggling the Switch C position several times and calculate the response time as recorded by Hosted App 3. Equipment B sends the command to trigger this test only under special maintenance mode of operations. This application does not execute under normal modes of operation. Hosted App 6 sends pass/fail results to Hosted App 4.
- Hosted App 4 is modified to consume the test results.

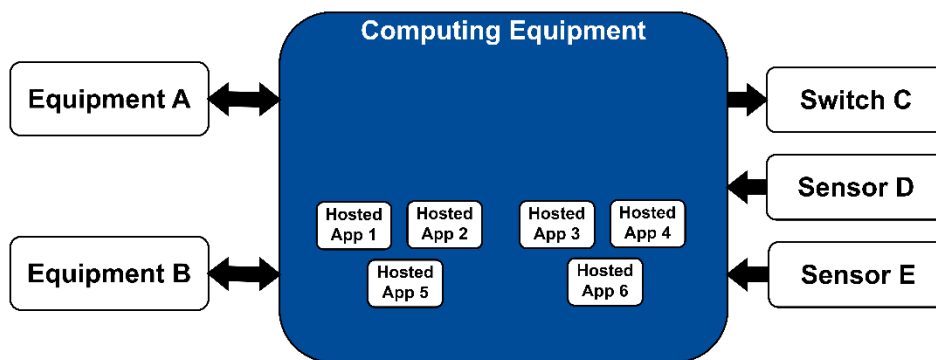


Figure 25 – Integrated Systems Example: System Design, New Hosted App

The Integrator tasks the Hosted App Provider with executing the WCET for Hosted App 6. Per instructions from MCP Platform Provider, this WCET does not include Microbenchmarks.

- Hosted App 1 (no change)
- Hosted App 2 (no change)

- Hosted App 3 (no change, modification for Hosted App 6 interface does not impact WCET or shared resource usage)
- Hosted App 4 (no change, modification for Hosted App 6 interface does not impact WCET or shared resource usage)
- Hosted App 5 (no change)
- Hosted App 6:
 - HOSTED-APPx_BASELINE_WCET – 5ms
 - HOSTED-APPx_USAGE – Shared Resource A & B

Integrator executes the equations for Hosted App 6.

- Hosted App 6 = 9ms (5ms + (2.5ms + 1.5ms))

Hosted App 6 is not critical for safety and only performs the test function in a specific mode of operation. The Integrator has no concerns with performance and utilizes the 9ms allocation.

Integrator reviews the required allocation details and defines design of the Integrated System and specifically the design of all five Hosted Applications.

- Hosted App 6 is allocated to Core 2.

The Integrator assesses the existing System Requirements and determines the requirements changes for the addition of Hosted App 6.

- No change: Requirement defined to disable Core3 only.
- No change: Requirement defined to separate the allocation of Hosted App 1 and 3.
- No change: Requirement defined to stagger schedule Hosted App 1 and 3 to minimize the overlap of execution time.
- No change: Requirement defined to include two instances of Hosted App 5 execution time slots on the same core.

Integrator executes the CDI Build process from the inputs noted above. The following CDI design is created:

- Hosted App 1 and 2 are allocated to Core0, Hosted App 3 and 4 are allocated to Core1, and Hosted App 5 and 6 is allocated to Core2.
 - Core0 allocated 40ms (25ms + 15ms) of the 50ms major frame.
 - Core1 allocated 40ms (30ms+ 10ms) of the 50ms major frame.
 - Core2 allocated 39ms (15ms+ 10ms+ 15ms) of the 50ms major frame.

The allocation report depicts the new allocation design.

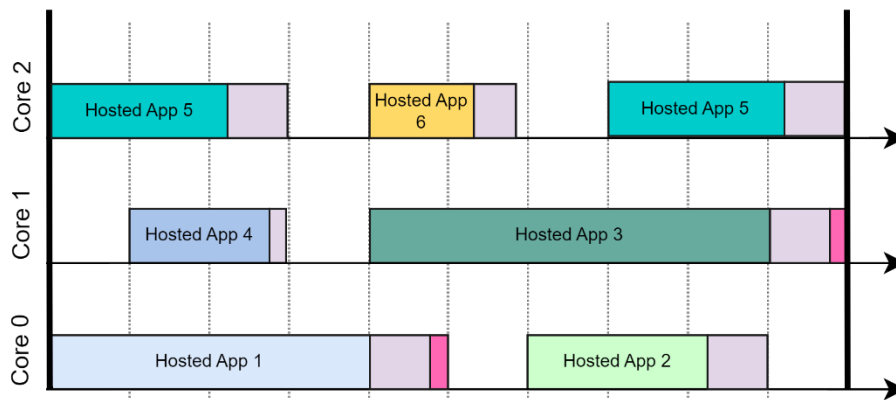


Figure 26 – Integrated Systems Example: Hosted App Execution with Hosted App 6

The Integrator performs a change impact analysis to determine if the existing verification data is acceptable. Because there were no changes to the allocation of Hosted App 1, 2, 3, 4, 5 and Hosted App 6 executes in maintenance mode, the existing requirements verification and WCET verification results for normal operation conditions is determined to be acceptable. The verification tests associated with maintenance mode of operation are repeated or new verification cases are created and executed.

A.5 Integrated System Change - Modified Integrated System

The Integrator develops a new need to transmit Sensor F data to Equipment G. The Integrator determines that the Computing Equipment has provisioned IO connections that support Sensor F and Equipment G. Hosted App 2 is modified to additionally receives the analog inputs from Sensor F, convert the data to engineering units, and store the data in memory. Hosted App 3 is modified to additionally retrieve the stored data and transmit to Equipment G.

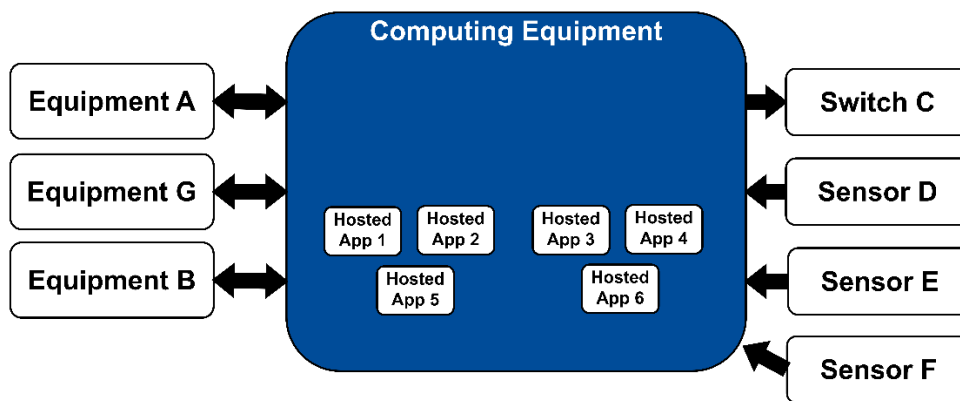


Figure 27 – Integrated Systems Example: System Design, New Function

The MCP Platform Provider completes the change impact assessment and determines no impact as the interfaces were previously provisioned in the MCP Platform.

The Hosted App Providers complete the change impact assessments and provide the following impacts:

- Hosted App 1: No Impact

- Hosted App 2: Impact to software for new function. New Hosted Application Allocation design required.
- Hosted App 3: Impact to software for new function. Existing allocation supports modified function.
- Hosted App 4: No Impact
- Hosted App 5: Impact to software for new function. Existing allocation supports modified function.
- Hosted App 6: No Impact

Integrator executes the equations for each Hosted Application based on the change impact assessment and design details from the Hosted App Providers.

- Hosted App 1 = 25ms (no change)
- Hosted App 2 = 20ms (16ms + (2.5ms + 1.5ms))
- Hosted App 3 = 30ms (no change)
- Hosted App 4 = 10ms (no change)
- Hosted App 5 = 15ms (no change)
- Hosted App 6 = 9ms (no change)

Integrator reviews the required allocation details and defines design of the Integrated System and specifically the design of all five Hosted Applications.

- No Change to Core allocations
- Additional 5ms needed for Hosted App 2

The Integrator assesses the existing System Requirements and determines no modifications to the requirements.

Integrator executes the CDI Build process from the inputs noted above. The allocation report depicts the new allocation design.

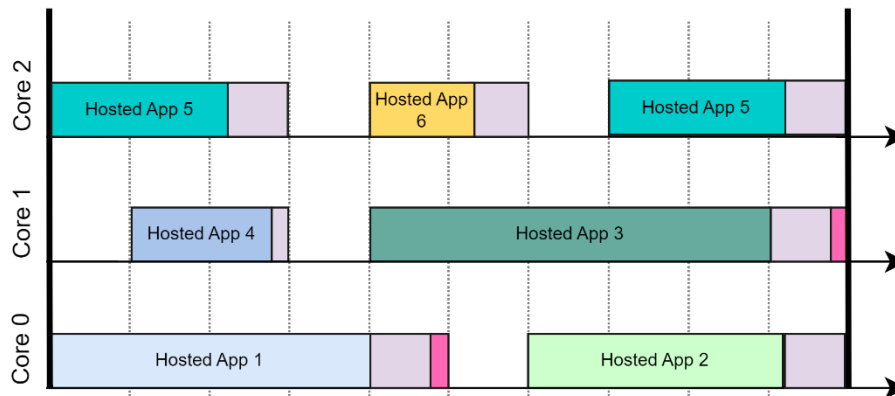


Figure 28 – Integrated Systems Example: Modified Hosted Applications

The Integrator performs a change impact analysis to determine if the existing verification data is acceptable. The changes to the Integrated System have no impacts on Hosted App 4 and 6. Given the change to Hosted App 2 and inter-core interfaces to Hosted App 3 and 5, verification activities must be re-executed along with modified or new activities for the new function (Sensor F data transmitted to Equipment G). This verification activity includes WCET focusing on the data and control coupling given the new interfaces.

As an alternate solution, the Integrator could define all new functionality into Hosted App 2 and not impact Hosted App 3. If Hosted App 3 remained unchanged, the change impact assessment could determine less re-verification activities both for Hosted App 3 and the Integrated System. The data and control coupling verification is likely still required unless the MCP Platform Provider's allocation report or tools can show Hosted App 1 and 3 are not impacted by the changes to Hosted App 2.