



The European Organisation for Civil Aviation Equipment
L'Organisation Européenne pour l'Équipement de l'Aviation Civile

REQUIREMENTS SPECIFICATION FOR AN AVIONICS COMPUTER RESOURCE (ACR)

This document is the exclusive intellectual and commercial property of EUROCAE.

It is presently commercialised by EUROCAE.

This electronic copy is delivered to your company/organisation for internal use exclusively.

In no case it may be re-sold, or hired, lent or exchanged outside your company.

ED-96

July 2000

REQUIREMENTS SPECIFICATION FOR AN AVIONICS COMPUTER RESOURCE (ACR)

This document is the exclusive intellectual and commercial property of EUROCAE.

It is presently commercialised by EUROCAE.

This electronic copy is delivered to your company/organisation for internal use exclusively.

In no case it may be re-sold, or hired, lent or exchanged outside your company.

ED-96

July 2000

TABLE OF CONTENTS

FOREWORD	iii
CHAPTER 1	INTRODUCTION 1
1.1	Purpose and Scope 1
1.2	Technical Concept 2
1.2.1	Application of the Open System Concept.....2
1.2.2	System Overview.....3
1.3	Operational Goals and Objectives 4
1.4	Specification Language.....4
1.5	Referenced Documents 5
1.6	Organization of the Document 5
CHAPTER 2	REQUIREMENTS AND QUALIFICATION 6
2.1	Function and Operation 6
2.1.1	Mandatory Services 7
2.1.2	Extensions to API Services 10
2.1.3	Interrupt Management 10
2.1.4	Fault Tolerance.....11
2.2	Performance Parameters 11
2.2.1	Processing Performance 12
2.2.2	Memory.....13
2.2.3	Communication 14
2.2.4	Input / Output.....14
2.2.5	Logical Connections 15
2.3	Power-On Initialization and Power Interruptions 16
2.4	Environmental Qualification 16
2.5	ACR Function and Operation Test Procedure 17
2.5.1	Mandatory services 17
2.5.2	Extension to API Services 19
2.5.3	Interrupt Management 19
2.5.4	Fault Tolerance.....19
2.6	Performance Parameters Test Requirements 19
2.6.1	Processing Performances 19
2.6.2	Memory.....20
2.6.3	Communication 20
2.6.4	Input / Output.....20
2.6.5	Logical Connections 21
2.7	Power-On Initialization and Power Interruption Test Requirements 21

CHAPTER 3	INTEGRATION, INSTALLATION AND CERTIFICATION	22
3.1	Accessibility of equipment and controls	22
3.2	Aircraft Environment	23
3.3	Display Visibility	23
3.4	Dynamic Response	23
3.5	Failure Information	23
3.6	Electromagnetic Interference	23
3.7	Inadvertent Turnoff	23
3.8	Electrical load analysis	23
3.9	Software Applications	24
3.10	ACR Utilization	24
3.11	Configuration and partition assignment	24
3.12	ACR and Application Criticality	25
3.13	Installation Functional Testing	25
3.13.1	Power Input	25
3.13.2	Associated Equipment or Systems	25
3.13.3	Adjustment of Equipment	25
CHAPTER 4	CERTIFICATION OF MODIFIED SYSTEMS	26
4.1	Installation of New or Modified Applications	26
4.2	Application-to-Application Compatibility	26
ANNEX A	DATA SHEET FOR ACR ATTRIBUTES	27
ANNEX B	APPLICATION INTERFACE SPECIFICATION EXAMPLE	39
ANNEX C	GLOSSARY OF TERMS	116
ANNEX D	LIST OF ABBREVIATIONS AND ACRONYMS	119
ANNEX E	EXPECTED RE-USE BENEFITS ASSOCIATED WITH THE ACR (COMMENTARY)	120
ANNEX F	MEMBERSHIP EUROCAE WORKING GROUP 48 / RTCA SPECIAL COMMITTEE 182	132

FOREWORD

1. This document, prepared by EUROCAE Working Group 48 "Avionics Computer Resource", jointly with RTCA SC-182, was accepted by the Council of EUROCAE on February 2000.
2. EUROCAE is an international non-profit making organisation. Membership is open to European users and manufacturers of equipment for aeronautics, trade associations, national civil aviation administrations and, under certain conditions, non-European organisations. Its work programme is principally directed to the preparation of performance specifications and guidance documents for civil aviation equipment, for adoption and use at European and world-wide levels.
3. The findings of EUROCAE are resolved after discussion among its members and in co-operation with RTCA Inc., Washington DC, USA and/or the Society of Automotive Engineers (SAE), Warrendale PA, USA through their appropriate committees.
4. This document is identical with RTCA DO-255, except for few minor differences in text which have no technical significance.
5. EUROCAE performance specifications are recommendations only. EUROCAE is not an official body of the European Governments; its recommendations are valid as statements of official policy only when adopted by a particular government or conference of governments.
6. Copies of this document may be obtained from:

EUROCAE
102, rue Etienne Dolet
92240 MALAKOFF
France

Tel.: 33 1 40 92 79 30
Fax: 33 1 46 55 62 65
E-mail: eurocae@eurocae.net
Internet: www.eurocae.net

CHAPTER 1

INTRODUCTION

As the aviation industry moves to embrace implementation techniques developed for the computer industry, relationships between avionics equipment and operational function are changing. Software has become a principal medium for implementing the operational function. The airplane appliance has become a computer platform which executes the software program. Technology turnover in computer and electronic industry is short compared with the lifetime of an aircraft. Open system concepts allow the application software and the computer platform to be developed and evolve independently. During aircraft installation the software and computer platform components are integrated into a system which provides the desired operational capability.

These requirements specifications for the Avionics Computer Resource (ACR) are intended to facilitate certification efficiency and economy of scale of the computer platform. The functionality is that of a general purpose computing resource. As a resource for hosting application software, these parameters are a consistent set of computation services, the performance of those services, a robust partition environment, quantity and capacity of resources.

Certain attributes, valuable to a high integrity computing platform suitable for aviation application, are prescribed. However, specific values for performance, quantity, and capacity that would quickly become obsolete are left for the ACR manufacturer to specify. This specification requires the preparation of a data sheet that enumerates the relevant aspects of the resource being provided. This provides for scalability and allows this specification to be stable over time.

To comply with this Requirements Specification, the ACR manufacturer is obligated to prepare a data sheet which conveys performance, capability and capacity in a consistent manner. Qualification of the ACR design assures that the necessary attributes are implemented and that the performance declared in the data sheet over a foreseeable range of environmental conditions is achieved. Compliance with this Requirements Specification demonstrates equipment operation and performance prior to installation on an aircraft.

1.1 PURPOSE AND SCOPE

This document contains the requirements specification for an Avionics Computer Resource (ACR). The ACR provides shared computation resources, core software and signal conditioning necessary to interface with a variety of aircraft systems. The ACR provides a highly adaptable computer environment for a variety of software applications. By complying with the requirements specified by this specification the ACR and the software applications may be independently developed. Together the ACR and installed software applications perform one or more aircraft functions.

The aircraft functions can be hosted on multiple ACR platforms that can be provided by multiple vendors. Compliance with this requirements specification ensures a complete specification of the Application Programming Interface (API) provided by the ACR, its capacity and capability, and its performance. Based on this ACR specification, application software may be independently developed.

The document is intended to assist manufacturers, software application developers and installers in achieving an ACR and installed software applications which perform their intended functions satisfactorily under all conditions normally encountered in routine aeronautical operations.

This document stipulates that the ACR manufacturer provides a suite of tests and analyses to assure the operation, performance and capacity of the ACR. Although specific test procedures are not included, assurance objectives are established for each ACR attribute.

In developing an ACR as specified by the requirements in this requirement specification, the ACR manufacturer is required to specify the development environment required only to develop the ACR. Whether the ACR manufacturer specifies, recommends, or supplies a development environment for application software developers is a decision between the ACR manufacturer and prospective software application developers, and should be documented as part of the software application approval process rather than under this specification. The ACR manufacturer is also obligated to specify the integration and configuration tools and environment necessary such that requirements for the addition or deletion of applications to the ACR will be adhered to at the time they are integrated.

1.2 TECHNICAL CONCEPT

The concept of “open system” architecture is adapted from the general computer industry to the aviation electronics industry by this requirements specification. The open system architecture features a defined API which:

- Allows independent development of the hardware computing platform and the software application;
- Promotes re-use of the software application on multiple aircraft models;
- Reduces the certification effort of follow-on approvals;
- Enhances “economy of scale” of both the computer platform and software application programs;
- Facilitates technology upgrades.

1.2.1 Application of the Open System Concept

The open system concept is expected to mitigate follow-on certification costs such that an operator can economically achieve a common operational capability across a fleet of mixed aircraft types, and frequently upgraded functional capability.

The open system concept facilitates evolutionary growth in operational capability after initial installation with an ACR platform. To have the ACR recognized as such, the ACR manufacturer is obligated to provide detailed characteristics, captured in the ACR data sheet that promotes an open architecture and ensures portability. Robust partitioning features of the ACR ensure isolation of the applications from one another and from the ACR platform.

Various configurations of ACR may be generated to accommodate fundamental airframe differences such as interconnect technology. In keeping with the open system concept, each ACR platform is expected to provide an interoperable API.

1.2.2 System Overview

This requirements specification defines a computer platform suitable for hosting multiple, independent software applications. An approved Avionics Computer Resource (ACR) is seen as an enabling step towards standardized, re-usable avionics software applications. The ACR API is defined such that it enables the hardware and software to be developed and evolve independently.

The avionics adaptation of open system architecture is composed of three parts:

- The ACR platform;
- One or more software applications;
- Aircraft installation of the platform and a specific configuration.

An approval process is associated with the computer platform, the software applications, and integration of the ACR, the software applications and the aircraft installation.

1.2.2.1 Avionics Computer Resource

An ACR that complies with this specification provides:

- A set of system services corresponding to a defined API;
- Allocation of resources to each application;
- Means to segregate applications;
- Means to detect, prevent and annunciate any attempted partition violations;
- Health monitoring of the computer platform and the associated system services;
- Means to communicate with other aircraft systems.

Approval of the ACR ensures that the defined API services, partitioning and health monitoring perform correctly over the full range of environmental conditions encountered in aeronautical operations.

1.2.2.2 Application Software

Aircraft operational functionality is provided by software applications that can be developed independently from the ACR platform. Application software is interoperable between ACR platforms that have the same API and sufficient available resources. Approval of the software application would ensure compliance with the API and associated service requests, confirmation of the quantity and capacity demands and evidence of adequate software engineering discipline during development. In addition approval of the software contributes to assurance of operation and performance of the function.

1.2.2.3 Aircraft Installation

The approval of the ACR and each application, prior to integration, contribute to the assurance of airworthiness. The first certification of the ACR and its installed applications is described in chapter 3. It includes verification of consistency between the aircraft environment and the environmental test categories of the platform, the architecture of the installation, and consistency between the software level and the associated failure condition classifications. During aircraft installation the ACR aircraft interfaces, interoperability of the combined set of software applications, and compatibility with other aircraft systems are confirmed via appropriate analyses, ground and flight tests.

1.3 OPERATIONAL GOALS AND OBJECTIVES

The operational goal of the ACR is to provide an open system environment capable of hosting multiple, independently developed software applications. In achieving these goals the ACR:

- Complies with a defined API;
- Provides computer resources for each installed application;
- Provides isolation and protection between applications;
- Is field loadable;
- Includes health monitoring.

The open system environment facilitates efficient re-use of software. Such re-use leads to:

- Uniform operational capability across mixed fleets;
- Timely response to changes in the national airspace operating environment;
- Minimized follow-on certification effort.

1.4 SPECIFICATION LANGUAGE

This document contains requirements that are mandatory and others that are recommended.

The use of the word “shall” indicates a mandated criterion; i.e., compliance with the particular procedure or specification is mandatory and no alternative may be applied.

The use of the word “should” (and phrases such as “it is recommended that...”, etc.) indicate that though the procedure or criterion is regarded as the preferred option, alternative procedures, specifications or criteria may be applied, provided that the manufacturer, installer or tester can provide information or data to substantiate and justify the alternative.

1.5 REFERENCED DOCUMENTS

1. EUROCAE ED-12B / RTCA DO-178B: "Software Considerations in Airborne Systems and Equipment Certification" dated December 1992 and amendment 1 dated October 1999.
2. EUROCAE ED-14D / RTCA DO-160D: "Environmental Conditions and Test Procedures for Airborne Equipment" dated February 1998.

1.6 ORGANIZATION OF THE DOCUMENT

Chapter 1 provides an introduction to the concepts embodied in the ACR.

Chapter 2 contains the requirements for the ACR. Also included are test requirements necessary to demonstrate equipment compliance with the stated specification.

Chapter 3 describes the requirements of the ACR and software applications when installed on an aircraft.

Chapter 4 describes the procedures after a change is made to a previously approved system and defines conditions that will assure the equipment user that operations can be conducted safely and reliably in the (changed) operational environment.

Annex A contains the templates for the ACR data sheet.

Annex B contains the API reference material.

Annex C contains the glossary of terms specific to this document.

Annex D lists the abbreviations and acronyms.

Annex E provides commentary regarding the potential re-use of life cycle data.

CHAPTER 2

REQUIREMENTS AND QUALIFICATION

This chapter contains the requirements for the ACR. Sections 2.1 through 2.4 provide the requirements for specifying function and operation, performance and capacity, power initialization, and operational environment of the ACR. Sections 2.5 through 2.7 provide the requirements for specifying tests and/or analyses to verify compliance with the specifications from sections 2.1 through 2.4.

The ACR manufacturer shall provide a data sheet that describes the ACR attributes and their performance. A template data sheet is provided in Annex A.

2.1 FUNCTION AND OPERATION

The following sections of this specification define the minimum system services that are provided by an ACR platform. A general description of each system service and its operation is provided. The ACR platform manufacturer shall provide a detailed description of the Application Programming Interface and describe the operation of each system service implemented. The ACR platform manufacturer may provide this description directly as a "user's guide" or by reference to a published, public standard.

The ACR shall include internal hardware and software management methods as necessary to ensure that time, space, and I/O allocations are deterministic and static. "Deterministic and static" means, in this context, that time, space and I/O allocations are determined at compilation, assembly, or link time, remain identical at each and every initialization of a program or process, and are not dynamically altered during runtime. If there is a mechanism in the ACR to reallocate resources at the time an application is hosted or unhosted, the reallocation shall be fixed at the time the allocation is made.

To ensure the partitioning function, an ACR shall manage all hardware resources residing within the ACR and shall monitor access to all hardware resources connected to the ACR. The ACR core software shall execute on the ACR hardware with control over all hardware and software resources to ensure partitions are non-interfering. The core software performs an essential access mediation function, conceptualized in FIGURE 1, that is independent of other services provided by the ACR. Access mediation shall be complete, tamper-proof, and assured, where these attributes are defined as follows:

Complete. All resources shall be under control of the core software. There shall be no way for software running in any partition to bypass the core software and access resources not allocated to that partition.

Tamper-proof. There shall be no way for software running in any partition to tamper with the core software or its data so as to subvert the core software control of system resources.

Assured. The core software shall comply with the assurance requirements of ED-12B / DO-178B (ref. 1). The ACR manufacturer shall include the assurance level of the ACR in the data sheet. Compatibility between the ACR assurance level and the failure condition classification of the functions performed by the applications hosted should be confirmed via a system safety assessment.

Collectively these attributes ensure that an ACR has the minimum structural properties needed to achieve high-integrity partitioning. An ACR shall possess these attributes regardless of which core software services are offered to applications running within the ACR's partitions.

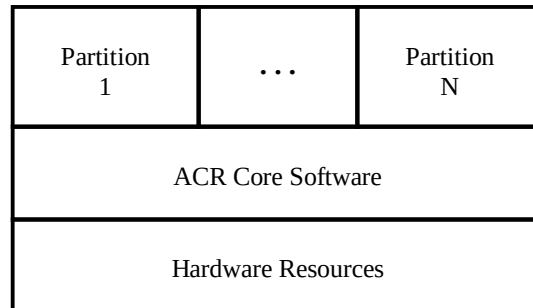


FIGURE 1: ACR ACCESS MEDIATION

The ACR manufacturer shall specify (and may supply) the development, integration, and configuration environment necessary so that requirements for the addition or deletion of applications to the ACR will be adhered to at the time of integration and configuration.

2.1.1 Mandatory Services

The ACR requirements specification defines the minimum core applications services. The ACR manufacturer shall provide the interface specification for the following application service groups:

- Partition Management
- Process Management
- Time Management
- Memory Management
- Communications
- Health Monitoring
- Field Loadability of software

The ACR manufacturer shall provide the interface specification for each service implemented and data structures supported using the data sheet template provided in Annex A. Annex B is an example of an interface specification.

2.1.1.1 Partition Management

The ACR shall provide a defined set of partition management services. These services shall allow one or more robust software partitions to be structured onto a processing resource. The ACR Core software shall prevent any adverse behaviour of software contained in a partition from affecting the operation of any other partition. The partitions shall be scheduled in a deterministic manner, ensuring that the allocation of processing time leads to predictable real time behaviour with demonstrable properties. Each partition shall exist in pre-defined areas of memory. The ACR manufacturer shall describe the mechanism for managing partitions in the ACR data sheet.

The ACR manufacturer shall document the method of time allocation to include:

- The maximum number of partitions supported by the allocation scheme
- The partition scheduling (e.g. fixed at manufacture, set at integration, set at load, set at power-up)
- Partition oriented parameters (e.g. period, duration, jitter)
- The scheme for external I/O (e.g. continuous, at begin or end of period, between context switch)

The ACR shall include internal hardware and software mechanisms necessary to ensure that software contained in a partition is prevented from violating the time and space requirements of other partitions. Any such attempt at partition violation shall be detected, blocked, and reported in a manner such that the ACR is provided with a means of uniquely identifying and handling the error. The handling of attempted violations shall not affect software contained in other partitions running on the ACR.

2.1.1.2 Process Management

The ACR shall include a defined set of process management services that provide the means of handling processes within partitions. A partition may contain one or more processes. The services shall provide prioritization of processes. The process management services shall establish and maintain deterministic behavior of the ACR. The ACR manufacturer shall describe the mechanism for process management in the ACR data sheet.

2.1.1.3 Time Management

The ACR shall provide a defined set of time management services. The ACR shall include internal hardware and software mechanisms to manage execution time of each partition and process. The time management services shall establish and maintain deterministic behavior of the ACR. The services shall provide the mechanisms necessary for partition and process scheduling. Time services (e.g. acquire system time) shall be available to support the applications in the partition. The alteration of timing and scheduling parameters of one partition shall not affect timing and scheduling parameters of another partition except as part of an aircraft level integration and configuration activity. The ACR manufacturer shall describe the mechanism for time management in the ACR data sheet.

2.1.1.4 Memory Management

The ACR shall provide a defined set of memory management services that ensure robust memory segregation of the partitions. The memory resource for each partition shall be defined during the installation of a specific application configuration. The software contained in a partition shall not be able to obtain memory resource exceeding its allocation defined during configuration. The ACR manufacturer shall describe the mechanism for memory management in the ACR data sheet.

2.1.1.5 Communications

The ACR shall provide a defined set of communication services to support all communication requirements. The communication services shall support information flow between partitions and within a partition. Direct communication by an application outside of the communication services shall not be permitted. The communication services shall support streams of information and event information, the types of services shall be derived from the functional requirement of the partition. Applications should only depend on the communication characteristics as specified in the ACR data sheet. The operation of communications services shall not violate partition boundaries. All communication characteristics shall be specified in the ACR data sheet.

2.1.1.6 Health Monitoring

The ACR shall provide a defined set of health monitoring services that provide the health status of the ACR to an external system. The health monitoring services shall provide information on the health status of partitions on the ACR. The ACR shall provide the capability to receive information from applications. The health monitor shall synthesize an ACR status report composed of:

- Errors raised by applications;
- Status reported by BIT run at power-up or on request;
- Maximum resource (time, space and I/O) utilization per partition;
- Any attempted violation of a partition boundary;
- Other data defined by the ACR manufacturer.

The health monitor shall provide the capability to store the health report into a non-volatile memory and shall provide the capability to report to an external system. The actions to be performed if a fault is detected within a partition or within the whole ACR itself shall be described. This description shall be part of the ACR data sheet.

COMMENTARY: Different functions and systems may require different approaches for handling errors, faults and failures. Whereas one application may tolerate re-initialization another may require a mode reversion. Other applications may be required to be shut down for the remainder of the operating period.

The ACR shall provide a mechanism for the detection of platform detected run-time errors and notify the application when an exception occurs. All error handling conditions shall be specified in the ACR data sheet. Each time an application calls the core software through a service routine, error information shall be returned to the calling application. Interrupts, error traps, and other detection and handling means for other error and fault types shall be identified and specified. Application programmable means of error handling, such as programmable vectoring, shall be satisfied. Application attempts to breach partitioning shall be detected and blocked, and ACR handling of the event shall be specified. Any error handling that will affect the performance of software applications shall be specified and such specification shall be provided in the ACR data sheet.

2.1.1.7 Field Loadable Software

The ACR shall provide a mechanism for the application software and the core software to be field loadable on the aircraft. The field load capability shall:

- Ensure that data loading is performed only when scheduled and while the aircraft is on the ground;
- Provide confirmation of successful data transfer or annunciation of a failure to transfer;
- Provide version identification of all field loadable items.

The field loading communication interface, protocol specification and means of ensuring correct data transfer shall be included in the ACR data sheet.

2.1.2 Extensions to API Services

If an ACR provides extended services, they shall be specified to the same level of detail as provided for mandatory services. This information shall be provided as part of the data sheet provided by the ACR manufacturer.

2.1.3 Interrupt Management

The interrupt management shall include the facilities and methods available to control all supported interrupts. Recognition of interrupts and distribution of related events shall be accomplished by the ACR core software. The interrupt mechanism shall comply with the ACR partition requirements.

The following information (as a minimum) shall be provided as part of the ACR data sheet:

- Minimum and maximum allowed interrupt rates
- Interrupt types supported (e.g. edge or level triggered)
- Interrupt Latency
- Masking (identify non-maskable interrupts)

- Hardware and software interrupts (identify interrupt ranges and relative priorities)
- Pre-assigned Interrupts (identify interrupts allocated by the ACR)

2.1.4 Fault Tolerance

If fault tolerance is provided by the ACR, the data identifying the specific type of fault tolerance, including information regarding effects on hosted application, shall be specified in the ACR data sheet.

2.2 PERFORMANCE PARAMETERS

The ACR performance shall be specified using measurable parameters defined in this section. This requirement specification does not specify absolute values, but parameters that may be measured and compared for relative merit. Performance characteristics shall be stated in terms of representative or actual performance of the ACR, not theoretical maxims.

COMMENTARY: This information is required for system integration activities to enable determination of the capabilities of the ACR and its ability to host a specific configuration of applications. This information is essential reference material for application developers, ACR system integrators and the certification authorities.

The ACR performance parameters shall be provided using a standard data sheet format as shown in Annex A.

2.2.1 Processing Performance

The ACR manufacturer shall specify, as part of the ACR data sheet, the supported processing performance parameters. The information provided by the manufacturer shall include, as a minimum, the data shown in Table 1 for each component.

Attribute	Description
Hardware CPU bandwidth and performance	CPU family and clock speed.
Cache Memory	Cache sizes for all caches (data, instruction for example).
Configuration of Cache	Cache type, such as burst cache, size and whether it may be disabled.
Instruction mix benchmark	Benchmark used to measure the ACR performance.
Maximum Partitions Supported	Value for the maximum number of partitions that are supported by the ACR
Minimum Partition Duration	Value for the shortest period available to an application
Maximum Partition Duration	Value for the maximum time available to an application
Maximum Partition Jitter	Value for the maximum variation in activation of a partition
Partition context switch	Value for the time taken to switch context of a partition.
Process context switch	Value for the time taken to switch context of a process.
Instruction per clock cycle	Number of instructions specified and permitted per clock cycle and whether number of instructions may be configured.
Instruction Set Architecture	Architecture attributes such as RISC and CISC, instructions, addressing modes, and byte ordering and storage units.
On Chip Power Management	Configuration and ability to set parameters such as disable.

TABLE 1: PROCESSING PERFORMANCE ATTRIBUTES

2.2.2

Memory

The ACR manufacturer shall specify, as part of the ACR data sheet, information concerning the available memory provided by the ACR. The ACR system integrator is responsible for the allocation and usage of memory for installed applications. The information provided shall include, as a minimum, the data shown in Table 2 for each memory type.

Attribute		Description
Technology		Type of installed technologies (e.g., EPROM, NVRAM, ROM, RAM).
Addressing Capacity		Maximum address range (logical and physical)
Minimum Addressing Unit		Number of data bits between two successive memory addresses
Access Time		Read/write cycle times of memory devices for each technology used
Memory Allocation		
	In ACR	Installed capacity of total memory
	Used by the core software (includes executive services)	Total size and type of memory allocated to core software functions
	Cache	If cache is used, the maximum access time
	Allocable Size for Partitions	Amount of allocable memory that can be assigned during development/system integration for partitions Maximum and minimum memory size allocable to a partition
Memory Usage (Available to Partitions)		
	Data Storage	Memory space available to each partition for data storage and retrieval
	Program Storage	Memory space available to each partition for program storage

TABLE 2: MEMORY ATTRIBUTES

2.2.3 Communication

The ACR manufacturer shall specify, as part of the ACR data sheet, the supported communication mechanisms. The information provided by the manufacturer shall include, as a minimum, the data shown in Table 3 for each component of the communication mechanisms.

Attribute	Description
Type	Type of communications protocols supported.
Quantity	Number of communications ports available.
Capacity	Maximum (sustained and burst) data throughput supported by the ports
Delay	Delay (minimum and maximum) due to overhead from the protocol and the ACR scheduler
Rate	Expected data rates supported by the protocol and port. This information includes: minimum, maximum, and default data rates supported
Boundary Conditions	Communication operation at partition boundaries

TABLE 3: COMMUNICATION ATTRIBUTES

2.2.4 Input / Output

The ACR manufacturer shall provide, as part of the ACR data sheet, the description of the characteristics of input/output of the ACR (the physical interfaces). The information provided by the manufacturer shall include, as a minimum, the data shown in Table 4 for each I/O component.

Attribute	Description
Identification	Physical addressing
Physical layer supported	
Propagation delay from application to physical layer	Delay imposed by the device in transferring data from the application level to the physical
Direction	Simplex, duplex
Device driver queuing properties	Discipline enforced (such as FIFO, truncation on queue full, overwrite on queue full)
Number of devices supported	Quantity of each type
Latency properties	Factors that influence I/O latency (polled I/O cycle time, Interrupt latency for interrupt driven I/O, etc.)
Buffer size	Size of the I/O buffers used to support each device
Data buses	Quantity of each type
Data throughput properties	
Capacity	Minimum and maximum capacity - number of messages and sizes.
Rate	Rate at which I/O can be handled.
Delay	Expected time delays.

TABLE 4: INPUT / OUTPUT ATTRIBUTES

2.2.5 Logical Connections

For each type of input and output device, the ACR manufacturer shall specify, as part of the ACR data sheet, the functionality and performance characteristics of each device driver.

COMMENTARY: The software application can only access physical resources through logical connections as shown in FIGURE 2.

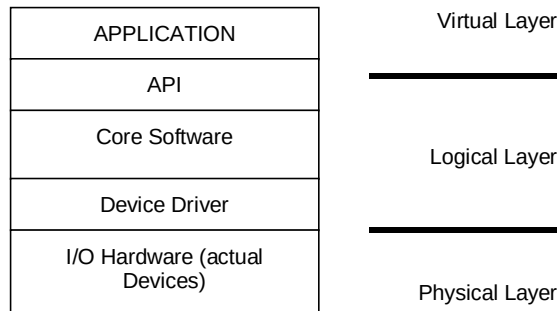


FIGURE 2: LOGICAL CONNECTIONS

The information provided by the manufacturer shall include, as a minimum, the data shown in Table 5 for each device.

Attribute		Description
I/O Device		
	Validation	Internal integrity checking of information
	Buffering	Temporary storage mechanism until information is needed
	Storage	Physical storage device and location, and attributes such as whether or not memory is retained during power interrupts
	Unbundling	Mechanism for removing information from packaging
	Transmission	Mechanism for output of information e.g. interrupt driven
	Reception	Mechanism for input of information
Capacity		Amount of information that the device is capable of moving
Rate		Rate at which the information is flowing
Delays		Any intentional time introduced by the I/O device to meet timing requirements
Boundary Conditions		Logical connection operation at partition boundaries
Queuing		Mechanism for prioritizing and scheduling information
Latencies		Time overhead incurred to send information from the source to the destination
Protocol		Format and interface specification (API)

TABLE 5: ATTRIBUTES FOR LOGICAL CONNECTIONS

2.3

POWER-ON INITIALIZATION AND POWER INTERRUPTIONS

The power on initialization and power interrupt operational characteristics of the ACR shall be described as part of the ACR data sheet. The description shall address the activities performed by the ACR from the time power is applied until control is passed to applications, and response to interruption of power such as length of any hold up capability and notification to applications.

2.4

ENVIRONMENTAL QUALIFICATION

The ACR shall be qualified to the environmental conditions contained in ED-14D / DO-160D (ref. 2). The ACR manufacturer shall include in the ACR data sheet the environmental categories to which the ACR has been tested.

COMMENTARY: The specific environmental qualification requirements are dependent on the location within the aircraft where the ACR is installed, the operational environment of the aircraft, and the aircraft functions performed by the applications being hosted.

Acceptable ACR operation and performance under environmental tests shall show that:

- System services operate correctly;
- I/O Interfaces operate correctly;
- Partition integrity is confirmed;
- The allocation of system resources to partitions is confirmed.

2.5 ACR FUNCTION AND OPERATION TEST PROCEDURE

The criteria specified in this section provide assurance that functional and operational characteristics of the ACR described in the previous sections, and expressed in the ACR data sheet, have been properly implemented. The means used to demonstrate these characteristics shall be proposed by the applicant and may include test and/or analysis.

Tests and/or analyses shall be conducted which demonstrate the proper operation of the ACR to manage time, space and I/O allocations to software partitions.

Specific test cases shall be included to show proper operation of:

- Allocation of resources to multiple partitions;
- Scheduling and control of each partition;
- Prevention of one partition to interfere with another;
- Prevention of a partition from interfering with the platform core software.

2.5.1 Mandatory services

Tests and/or analyses shall be conducted that demonstrate the proper operation of the ACR for all API services offered by the platform.

The evaluation of the mandatory services may require instrumentation of the ACR. If any such instrumentation is permanently resident in the ACR, means shall be provided to disable it and to ensure it remains disabled during operational use. The effect of the instrumentation on performance or capability of the ACR shall be accounted for in the test results.

2.5.1.1 Partition Management

The ACR manufacturer shall specify the tests and/or analyses that demonstrate the proper operation of the ACR to:

- Associate specific areas of memory with each partition;
- Schedule multiple partitions with predictable repetition, duration, and jitter;
- Manage data flow between applications and the physical I/O;

- Detect and prevent attempted violations of a partition by its application software;
- Report the attempted violation to the offending application and health monitor function;
- Provide system service to the active partition.

2.5.1.2 Process Management

The ACR manufacturer shall demonstrate via test and/or analysis the proper operation of the ACR to:

- Manage multiple processes within a partition;
- Provide system service to the active process.

2.5.1.3 Time Management

The ACR manufacturer shall specify the tests and/or analyses that demonstrate the ACR functional and operational characteristics for time management.

2.5.1.4 Memory Management

The ACR manufacturer shall specify the tests and/or analyses that demonstrate the ACR functional and operational characteristics for memory management.

2.5.1.5 Communication Management

The ACR manufacturer shall specify the tests and/or analyses that demonstrate the proper operation of the ACR to:

- Provide communication services within a partition and between partitions;
- Provide communication services between partitions and the physical I/O;
- Coordinate the communication requests of multiple partitions.

2.5.1.6 Health Monitoring

The ACR manufacturer shall specify the tests and/or analyses that demonstrate the proper operation of the ACR to:

- Detect failures of the platform resources;
- Detect and prevent attempted violations of a partition by its application software;
- Report health status to application;
- Record health status in non-volatile memory;
- Report health status to an external entity.

2.5.1.7 Field Loadable Software

The ACR manufacturer shall specify the tests and/or analyses that demonstrate the proper operation of the ACR to:

- Perform software load using a suitable procedure and load equipment;
- Verify a successful data load;
- Detect and inhibit operation of an unsuccessful data load.

2.5.2 Extension to API Services

The ACR manufacturer shall specify the tests and/or analyses which demonstrate the proper operation of the ACR for all extensions to API services offered by the platform.

2.5.3 Interrupt Management

The ACR manufacturer shall specify the tests and/or analyses that demonstrate the proper operation of the ACR to:

- Respond to all supported interrupts;
- Mask and prioritize interrupts.

2.5.4 Fault Tolerance

If the ACR design includes fault tolerant characteristics, the ACR manufacturer shall specify the tests and/or analyses which demonstrate the proper operation as implemented. Where applicable, the following shall be tested:

- Fault detection monitors;
- Redundant elements;
- Reconfiguration or masking;
- Reporting of fault tolerant status to the health monitor function.

2.6 PERFORMANCE PARAMETERS TEST REQUIREMENTS

The test criteria specified in this section provide assurance that performance parameters provided by the ACR data sheet are valid. The means used to demonstrate these characteristics shall be proposed by the applicant.

2.6.1 Processing Performances

Each performance attribute of the data sheet shall be evaluated using a method representative of maximum loading of the platform. The results shall confirm that the values provided by the data sheet are achieved for all foreseeable operating conditions.

The ACR manufacturer shall specify the tests and/or analyses that demonstrate the capability of the ACR:

- To achieve the stated instruction bench mark;
- Partition context switch delay;
- Process context switch delay.

2.6.2 Memory

The ACR manufacturer shall specify the tests and/or analyses that demonstrate the performance of the ACR:

- Transaction time of each type of memory;
- Transaction time of each type of cache;
- Operations of all data addressing techniques supported.

2.6.3 Communication

The ACR manufacturer shall specify the tests and/or analyses that demonstrate the proper operation of each communication facility provided. Including:

- Throughput of the channel;
- Latency time from when the message is created in an application until it is delivered;
- Maximum and minimum message size.

2.6.4 Input / Output

The ACR manufacturer shall specify the tests and/or analyses that demonstrate the proper operation of each I/O facility provided. Including:

- Range, resolution and accuracy of the physical layer;
- Buffer capacity, and queue properties;
- Throughput of the channel;
- Latency time from when the message is created in an application until it is delivered;
- Latency time from when a message arrives at the ACR until it is delivered to an application.

2.6.5 Logical Connections

The ACR manufacturer shall specify the tests and/or analyses that demonstrate the proper operation of each type of logical channel. Including:

- Connection between the application and the physical layer;
- Buffer capacity, and queue properties;
- Throughput of the channel;
- Latency time from when the message is created in an application until it is delivered;
- Latency time from when a message arrives at the ACR until it is delivered to an application.
- Latency time from when a message arrives at the ACR until it is delivered to an application.
- Integrity checking and connection to the health monitor function.

2.7 POWER-ON INITIALIZATION AND POWER INTERRUPTION TEST REQUIREMENTS

The test criteria specified in this section provide assurance that the ACR functions properly during power turn on and power transients. The means used to demonstrate these characteristics shall be proposed by the applicant.

The ACR manufacturer shall specify the tests and/or analyses that demonstrate the performance of the ACR:

- To initialize properly, allocated resources to partitions and begin scheduling;
- In response to power interrupts, that data sheet values for hold up, and/or warm start are achieved for all foreseeable operating conditions.

CHAPTER 3

INTEGRATION, INSTALLATION AND CERTIFICATION

This section states the requirements for the ACR and hosted applications when installed in the aircraft. For the most part, installed performance requirements are the same as those contained in chapter 2, which were verified through bench and environmental tests. However, certain requirements may be affected by the physical installation and can only be verified after installation. The installed performance limits stated below take into consideration these situations.

The ACR, its applications, and their installation shall comply with all rules applicable for type certification or operation of the aircraft. A system safety assessment shall be performed which assesses the installation of the ACR and its applications, and the subsequent effect on the aircraft and its other installed systems and equipment.

FIGURE 3 shows consumers and producers of data necessary for an ACR based certification process. The contents of the data sheet are explained in Annex A.

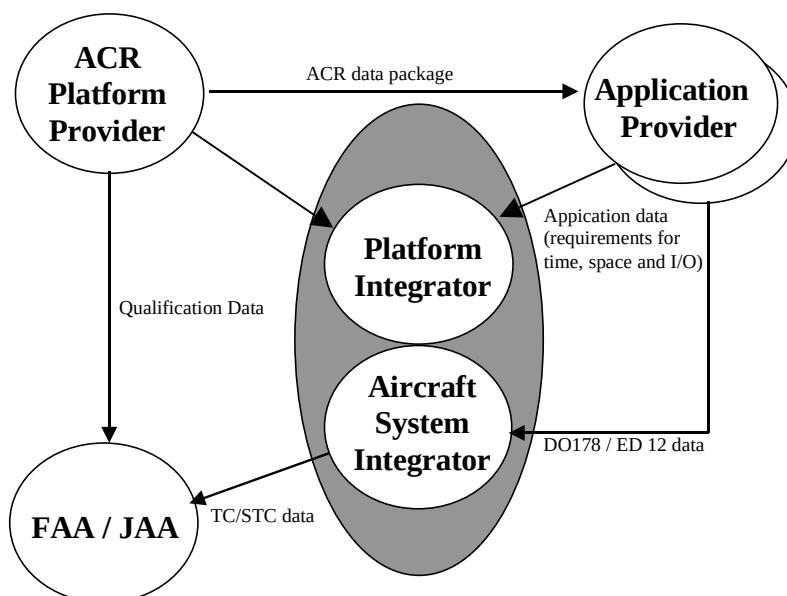


FIGURE 3: FLOW OF CERTIFICATION DATA

As specific compliance criteria are affected by the nature and kind of the applications presented for installation approval, the requirements of this chapter should not be assumed to be complete. The applicant should seek early agreement with the certification authority as to the specific compliance demonstrations that will be required for a particular certification program.

3.1 ACCESSIBILITY OF EQUIPMENT AND CONTROLS

The applicant shall demonstrate that other equipment and controls remain accessible after ACR installation. In addition, the applicant shall demonstrate that the ACR and its controls are accessible as necessary for its intended use, maintenance requirements, and other requirements of the ACR and its installed applications.

3.2 AIRCRAFT ENVIRONMENT

The ACR shall be shown to be compatible with the environmental conditions present in the specific location in the aircraft where the equipment is installed and with the operational environment of the aircraft. When adding applications to a previously installed ACR, it shall be considered whether the ACR will continue to operate within its qualified limits with the additional applications operating.

3.3 DISPLAY VISIBILITY

It shall be shown that visibility and readability of displays affected or added by the installation of the ACR or its installed applications are acceptable.

3.4 DYNAMIC RESPONSE

Aircraft maneuvering or changes in attitude encountered in normal flight operations shall not adversely affect operation of the equipment.

3.5 FAILURE INFORMATION

Where information to the crew must be provided in the event of a failure or unsafe operating condition of the ACR or its applications, it shall be shown that such information is provided in conformance with the applicable rules.

3.6 ELECTROMAGNETIC INTERFERENCE

It shall be shown that the ACR as installed is not a source of conducted or radiated interference nor is adversely affected by conducted or radiated emissions from other equipment or systems installed in the aircraft.

3.7 INADVERTENT TURNOFF

Appropriate protection shall be provided to prevent the inadvertent turnoff of the equipment.

3.8 ELECTRICAL LOAD ANALYSIS

An electrical load analysis shall be performed which considers the effects of the installation of the ACR and its installed applications. The analysis should consider, for example, loads on power generation systems, wiring capacity, and circuit protective devices.

3.9 SOFTWARE APPLICATIONS

The ACR and application software shall be developed according to the guidance provided by ED-12B / DO-178B (ref. 1) or equivalent. The ACR integrator shall determine if the application mix for a given ACR configuration will satisfy the execution, performance and safety requirements for each resident application. Architectural considerations for a given mix of applications on the same ACR platform shall include system level safety requirements such as single point failure and limitations of physical location.

The application developer shall provide documentation of execution environment resources necessary for the proper operation of the application. The ACR integrator shall determine if the mix of applications for a specific ACR will be schedulable.

3.10 ACR UTILIZATION

Ground or flight test is not sufficient for establishing that resource utilization remains acceptable after the installation of applications to the ACR platform. Sufficient number of input conditions may not be examined within a reasonable demonstration period. Therefore, when installing application software on the ACR, computer resource utilization shall be analyzed to show that:

- The number of partitions on the ACR does not exceed the total number allowed as specified by the ACR data sheet.
- Time, space and I/O resources are sufficient for hosting all applications installed.
- The necessary pathways for input signals and output signals for an application exist in the ACR.
- The ACR-host-configuration is not changed in any physical way in order to accommodate the applications.
- Any other resource utilization is sufficient for the specific set of installed applications.

The depth of analysis should be appropriate to the worst case failure effect associated with the functions performed. If tools are used to provide assurance and their output is not verified, then those tools shall be qualified.

3.11 CONFIGURATION AND PARTITION ASSIGNMENT

To demonstrate configuration and partition assignment, it shall be shown that:

- Applications are executed without violations of the boundaries of their resources
- Mechanisms used to prevent potential conflicts or violations operate properly
- Mechanisms used to measure overall performance operate properly
- A configuration management process that controls the ACR and each combination of application hosted is used.

3.12 ACR AND APPLICATION CRITICALITY

It shall be shown that the ACR and its applications comply with the safety requirements determined from a system safety assessment as required by the rules applicable for type certification or operation of the aircraft. Criticality levels and assurance levels of the ACR and the applications shall be appropriate to the results of the system safety assessment.

3.13 INSTALLATION FUNCTIONAL TESTING

The applicant shall conduct and provide evidence of laboratory tests, ground tests, flight tests, or a combination to demonstrate the proper functioning of the ACR with its installed applications. On-board verification methods shall be used to ensure the installed applications are resident on the proper ACR and that the operational hardware configuration supports the installed applications.

3.13.1 Power Input

Tests conducted on the aircraft to show proper functioning of the equipment as installed shall be conducted with the equipment powered by the aircraft's electrical power generating system.

3.13.2 Associated Equipment or Systems

Unless otherwise specified, all aircraft electrically operated equipment and systems shall be turned on when conducting electromagnetic compatibility tests.

3.13.3 Adjustment of Equipment

Circuits of the equipment under test shall be properly aligned and otherwise adjusted in accordance with the manufacturer's recommended practices prior to application of the specified tests.

CHAPTER 4

CERTIFICATION OF MODIFIED SYSTEMS

The certification of modified systems should apply analysis, flight tests, ground tests, cockpit evaluation and safety assessment consistent with existing regulations and processes.

The qualification of the ACR platform assures that the addition of a new application or modification of an existing one does not affect the robustness of the partitioning mechanism.

By the re-use of previously approved data, the applicant may be able to reduce the effort of certifying a change of applications or change of installation for a previously qualified and installed ACR or application(s). An analysis should be conducted to show how much of the existing data, hardware, and software approved during the previous certification or by qualification programs will remain unchanged. Parts of hardware and software which are not modified and which are shown to be unaffected by the new installation or application changes may be accepted using previous certification data.

COMMENTARY: Annex E illustrates three re-use scenarios and provides an indication of the expected reuse benefits with respect to the objectives of ED-12B / DO-178B (ref. 1). The life cycle data associated with meeting these objectives would be re-usable to the extent that the product described by the data did not change. Each applicant must confirm with the approving authority the applicability of re-used data.

4.1 INSTALLATION OF NEW OR MODIFIED APPLICATIONS

When installing a new or modified application on the ACR, computer resource utilization shall be reviewed. The following topics shall be demonstrated. The means used to demonstrate these characteristics shall be proposed by the applicant and may include test and/or analysis.

- The number of partitions on the ACR does not exceed the total number allowed as specified by the ACR data sheet.
- Time, space and I/O resources are sufficient for hosting the new configuration of new or modified applications.
- Time, space and I/O resources of existing applications are unaffected by the introduction of the new or modified application.
- The necessary pathways for input signals and output signals for a new or modified application exist in the ACR.
- That the ACR-host-configuration is not changed in any physical way in order to accommodate the new or modified application.
- Any other resource utilization is sufficient for the specific set of installed applications.

4.2 APPLICATION-TO-APPLICATION COMPATIBILITY

The applicant shall demonstrate the proper operation of peer to peer communication between applications in the modified configurations. The applicant shall verify that data dependencies or resource usage connections are acceptable for the new or modified configuration of applications.

ANNEX A

DATA SHEET FOR ACR ATTRIBUTES

The applicant for ACR approval must provide a data sheet describing the product to be approved. This annex provides an ACR data sheet template. The template is composed of four tables corresponding to the requirement sections of chapter 2:

- Function and Operation (Table A-1)
- Performance parameters (Table A-2)
- Power-On Initialization and Power Interruptions (Table A-3)
- Environmental Qualification (Table A-4)

The template contains the implementation details of all ACR attributes along with means of verification.

The first column of each table contains the grouping according to the section structure in chapter 2. The second column is the identification of the attribute, while the third column describes the attribute. The fourth column "Implementation" may be a value, range, rate, or other objective measure. The last column, means of verification, refers to life cycle data describing test, analysis, inspection or other means used for verification.

For the ACR system services (referring to section 2.1) the "Implementation" column contains an example (in italic) of the system services taken from APEX API as described in Annex B. If another API is being used, an appropriate specification of the services will be entered in this column.

Table A-1: ACR data sheet template “Function and Operation”				
Group	ID	Attribute	Implementation	Means of Verification
General Considerations Section 2.1	GEN1	Development Environment (e.g. programming language, compilers available, etc.)		
	GEN2	Integration Environment		
	GEN3	Configuration Environment		
Data Structures Section 2.1.1	SRV1	API dependent data structures	<i>APEX_TYPES</i>	
Partition Management Section 2.1.1.1 / 2.5.1.1	SRV2.1	Read the current partition status.	<i>GET_PARTITION_STATUS</i>	
	SRV2.2	Set the mode of the current partition.	<i>SET_PARTITION_MODE</i>	
	SRV2.3	Define Partition Modes	<i>type OPERATING_MODE_TYPE is (IDLE, COLD_START, WARM_START, NORMAL);</i>	

Table A-1: ACR data sheet template “Function and Operation” (continued)				
Group	ID	Attribute	Implementation	Means of Verification
Process Management Section 2.1.1.2 / 2.5.1.2	SRV3.1	Obtain the ID of the current process	<i>GET_PROCESS_ID</i>	
	SRV3.2	Obtain the status of the current process	<i>GET_PROCESS_STATUS</i>	
	SRV3.3	Define a new process	<i>CREATE_PROCESS</i>	
	SRV3.4	Set the priority of the current process.	<i>SET_PRIORITY</i>	
	SRV3.5	Suspend a process	<i>SUSPEND_SELF</i> <i>SUSPEND</i>	
	SRV3.6	Reactivate a suspended process	<i>RESUME</i>	
	SRV3.7	Stop/halt a process	<i>STOP_SELF</i> <i>STOP</i>	
	SRV3.8	Activate a process	<i>START</i>	
	SRV3.9	Prevent the current process from being pre-empted (turn preemption off)	<i>LOCK_PREEMPTION</i>	
	SRV3.10	Preemption on	<i>UNLOCK_PREEMPTION</i>	

Table A-1: ACR data sheet template “Function and Operation” (continued)				
Group	ID	Attribute	Implementation	Means of Verification
Time Management Section 2.1.1.3 / 2.5.1.3	SRV4.1	Delay processing for a specified duration	<i>TIMED_WAIT</i> <i>PERIODIC_WAIT</i>	
	SRV4.2	Read the current time from the system clock	<i>GET_TIME</i>	
	SRV4.3	Refresh the timer countdown value	<i>REPLENISH</i>	
Memory Management Section 2.1.1.4 / 2.5.1.4	SRV5.1	Allocate program store to each partition		
	SRV5.2	Allocate data store to each partition		
Communications Section 2.1.1.5 / 2.5.1.5 Inter-Partition Communications	SRV6.1	Define a communications port	<i>CREATE_SAMPLING_PORT</i> <i>CREATE_QUEUEING_PORT</i>	
	SRV6.2	Write to a communications port	<i>WRITE_SAMPLING_MESSAGE</i> <i>SEND_QUEUEING_MESSAGE</i>	
	SRV6.3	Read from a communications port	<i>READ_SAMPLING_MESSAGE</i> <i>RECEIVE_QUEUEING_MESSAGE</i>	
	SRV6.4	Obtain the communications port ID	<i>GET_SAMPLING_PORT_ID</i> <i>GET_QUEUEING_PORT_ID</i>	
	SRV6.5	Obtain the status of the communications port.	<i>GET_SAMPLING_PORT_STATUS</i> <i>GET_QUEUEING_PORT_STATUS</i>	

Table A-1: ACR data sheet template “Function and Operation” (continued)				
Group	ID	Attribute	Implementation	Means of Verification
Communications Section 2.1.1.5 / 2.5.1.5 Intra-Partition Communications	SRV7.1	Define inter-process communications object	<i>CREATE_BUFFER</i> <i>CREATE_BLACKBOARD</i>	
	SRV7.2	Synchronized write mechanism to inter-process communications object	<i>SEND_BUFFER</i> <i>DISPLAY_BLACKBOARD</i>	
	SRV7.3	Read from inter-process communications object	<i>RECEIVE_BUFFER</i> <i>READ_BLACKBOARD</i>	
	SRV7.4	Obtain ID of inter-process communications object	<i>GET_BUFFER_ID</i> <i>GET_BLACKBOARD_ID</i>	
	SRV7.5	Obtain status of inter-process communications object	<i>GET_BUFFER_STATUS</i> <i>GET_BLACKBOARD_STATUS</i>	
	SRV7.6	Clear the inter-process communications object	<i>CLEAR_BLACKBOARD</i>	
Communication Section 2.1.1.5 / 2.5.1.5 Data and Process Synchronization Support: Semaphores	SRV8.1	Control access to resources	<i>CREATE_SEMAPHORE</i>	
	SRV8.2	Suspend the current process pending resource availability	<i>WAIT_SEMAPHORE</i>	
	SRV8.3	Change the resource availability	<i>SIGNAL_SEMAPHORE</i>	
	SRV8.4	Obtain the ID of the resource control mechanism	<i>GET_SEMAPHORE_ID</i>	
	SRV8.5	Query the status of the resource control mechanism	<i>GET_SEMAPHORE_STATUS</i>	

Table A-1: ACR data sheet template “Function and Operation” (continued)				
Group	ID	Attribute	Implementation	Means of Verification
Communication Section 2.1.1.5 / 2.5.1.5 Data and Process Synchronization Support: Events	SRV9.1	Synchronize processes	<i>CREATE_EVENT</i>	
	SRV9.2	Set the synchronization event	<i>SET_EVENT</i>	
	SRV9.3	Clear the synchronization event	<i>RESET_EVENT</i>	
	SRV9.4	Suspend the current process until the synchronization event occurs	<i>WAIT_EVENT</i>	
	SRV9.5	Obtain the ID of a synchronizing event	<i>GET_EVENT_ID</i>	
	SRV9.6	Obtain the status of a synchronizing event	<i>GET_EVENT_STATUS</i>	
Health Monitoring Section 2.1.1.6 / 2.5.1.6	SRV10.1	Record application defined messages	<i>REPORT_APPLICATION_MESSAGE</i>	
	SRV10.2	Define an application specific error handler	<i>CREATE_ERROR_HANDLER</i>	
	SRV10.3	Obtain the current system health/error status	<i>GET_ERROR_STATUS</i>	
	SRV10.4	Record application detected failures	<i>RAISE_APPLICATION_ERROR</i>	
Field Loadable Software Section 2.1.1.7 / 2.5.1.7	SRV11	Mechanism for the application software and the core software to be field loadable		

Table A-1: ACR data sheet template “Function and Operation” (continued)				
Group	ID	Attribute	Implementation	Means of Verification
Extensions to API Services Section 2.1.2 / 2.5.2	EXT1	Extensions to API Services		
Interrupts Section 2.1.3 / 2.5.3	INT1	Min/Max Interrupt Rates		
	INT2	Interrupt Type		
	INT3	Interrupt Latency		
	INT4	Masking		
	INT5	H/W S/W interrupts		
	INT6	Pre-assigned interrupts		
Fault Tolerance Section 2.1.4 / 2.5.4	FLT1	Type of Fault Tolerance		
	FLT2	Effect on hosted applications		

Table A-2: ACR data sheet template “Performance Parameters”				
Group	ID	Attribute	Implementation	Means of Verification
Processing Performance Section 2.2.1 / 2.6.1	PRO1	Hardware CPU bandwidth and performance		
	PRO2	Cache Memory		
	PRO3	Configuration of Cache		
	PRO4	Instruction mix benchmark		
	PRO5	Maximum Partitions Supported		
	PRO6	Minimum Partition Duration		
	PRO7	Maximum Partition Duration		
	PRO8	Maximum Partition Jitter		
	PRO10	Partition context switch		
	PRO11	Process context switch		
	PRO12	Instruction per clock cycle		
	PRO13	Instruction Set Architecture		
	PRO14	On Chip Power Management		

Table A-2: ACR data sheet template “Performance Parameters” (continued)				
Group	ID	Attribute	Implementation	Means of Verification
Memory Performance Section 2.2.2 / 2.6.2	MEM1	Technology		
	MEM2	Addressing Capacity		
	MEM3	Minimum Addressing Unit		
	MEM4	Access Time		
	MEM5	Memory in ACR		
	MEM6	Used by the core software (includes executive services)		
	MEM7	Cache		
	MEM8	Allocatable Size For Partitions		
	MEM9	Data Storage		
	MEM10	Program Storage		
	MEM11	Allocation Policy		
Communication Performance Section 2.2.3 / 2.6.3	COM1	Type		
	COM2	Quantity		
	COM3	Capacity		
	COM4	Delay		
	COM5	Rate		
	COM6	Boundary Conditions		

Table A-2: ACR data sheet template “Performance Parameters” (continued)				
Group	ID	Attribute	Implementation	Means of Verification
I/O Performance Section 2.2.4 / 2.6.4	IOP1	Identification		
	IOP2	Propagation delay from application to physical layer		
	IOP3	Direction		
	IOP4	Device driver queuing properties		
	IOP5	Number of devices supported		
	IOP6	Latency properties		
	IOP7	Buffer size		
	IOP8	Data buses		
	IOP9	Capacity		
	IOP10	Rate		
	IOP11	Delay		

Table A-2: ACR data sheet template “Performance Parameters” (continued)				
Group	ID	Attribute	Implementation	Means of Verification
Logical Connections Performance Section 2.2.5 / 2.6.5	LGX1	Validation		
	LGX2	Buffering		
	LGX3	Storage		
	LGX4	Unbundling		
	LGX5	Transmission		
	LGX6	Reception		
	LGX7	Capacity		
	LGX8	Rate		
	LGX9	Delays		
	LGX10	Boundary Conditions		
	LGX11	Queuing		
	LGX12	Latencies		
	LGX13	Protocol		

Table A-3: ACR data sheet template “Power-On Initialization and Power Interruptions”				
Group	ID	Attribute	Implementation	Means of Verification
Power-On Characteristics Section 2.3 / 2.7	POW1	Initialization characteristics		
	POW2	Power interrupt response		

Table A-4: ACR data sheet template “Environmental Qualification”				
Group	ID	Attribute	Implementation	Means of Verification
Environmental Qualification Section 2.4	ENV1	Tested in accordance with ED-14D / DO-160D		
	ENV2	Compliance categories		

ANNEX B

APPLICATION INTERFACE SPECIFICATION EXAMPLEB.1 INTRODUCTION

This annex contains an example of an Application Software Interface that meets the objectives of this Requirements Specification.

B.1.1 Relationship to ARINC Specification 653

Several technical concepts were considered for the application software interface definition. The APEX model defined in ARINC Specification 653 was the most suitable model for use as an example. ARINC provided source material for Appendix B with the stipulation, that the ARINC 653 material be reviewed in a regulatory context. The material which follows is a modification of ARINC Specification 653, based on that review, that meets the objectives of the ACR. The modifications are minor, non-technical changes to highlight the key features of the application software interface definition. Appendix B is independent of ARINC 653. It is included in this document to emphasize the requirement to establish a comprehensive, well documented API for the ACR.

B.2 SYSTEM OVERVIEWB.2.1 System Architecture

This interface standard has been developed for use in an Integrated Modular Avionics (IMA) environment and traditional federated avionics architectures such as a single function LRU. The Application Executive (APEX) architecture supports partitioning in accordance with the IMA philosophy.

In an IMA cabinet, each core module can contain one or more individual processors. The core module architecture has influence on the O/S implementation, but not on the APEX interface used by the application software of each partition. The processes of a partition (and therefore the partition itself) cannot be distributed over multiple processors, either in the same core module or in different core modules. Application software should be portable between core modules and between individual processors of a core module without modifying its interface with the O/S.

To ensure portability, communication between partitions must be independent of the location of both the source and destination partition. An application sending a message to, or receiving a message from, another application should not contain explicit information regarding the location of its own host partition or that of its communications partner. The information required to enable a message to be correctly routed from source to destination is contained in configuration tables.

To achieve the required functionality and conform to specified timing constraints, the constituent processes of a partition may operate concurrently. The O/S provides services to control and support the operational environment for all processes within a partition. In particular, concurrency of operation is provided by the partition level scheduling model.

The underlying architecture of a partition within an IMA system is similar to that of a multitasking application within a general-purpose mainframe computer. Each

partition consists of one or more concurrently executing processes, sharing access to processor resources based upon the requirements of the application. All processes are uniquely identifiable, having attributes that affect scheduling, synchronization, and overall execution.

B.2.2 Hardware

In order to isolate multiple partitions in a shared resource environment, the hardware should provide the O/S with the ability to restrict memory spaces, processing time, and access to I/O for each individual partition.

Partition timing interrupt generation should be deterministic. Any interrupts required by the hardware should be serviced by the O/S. Time partitioning should not be disturbed by the use of interrupts.

Federated avionics (LRUs) may include other types of backplane bus or none at all. Federated systems that support multiple partitions should internally generate interrupts at regular time intervals to schedule those partitions.

There are several basic assumptions made about the processor:

- a. The processor provides sufficient processing to meet worst-case timing requirements.
- b. The processor has access to required I/O and memory resources.
- c. The processor has access to time resources to implement the time services.
- d. The processor provides a mechanism to transfer control to the O/S if the partition attempts to perform an invalid operation.
- e. The processor provides atomic operations for implementing processing control constructs.

B.2.3 System Functionality

This section describes the functionality to be provided by the O/S and its interface with the application software.

At the core module level, the O/S manages partitions (partition management) and their interpartition communication, the latter being conducted either within or across module boundaries. At the partition level, the O/S manages processes within a partition (process management) and communication between the constituent processes (intra-partition communication). The O/S may therefore be regarded as multi-level according to its scope of operation. O/S facilities (e.g., scheduling, message handling) are provided at each level, but differ in function and content according to their scope of operation. Definition of these facilities therefore distinguishes between the level at which they operate.

B.2.3.1 Partition Management

Central to the IMA philosophy is the concept of partitioning, whereby the functions resident on a core module are partitioned with respect to space (memory partitioning) and time (temporal partitioning). A partition is therefore a program unit of the application designed to satisfy these partitioning constraints.

The O/S provides robust partitioning. A robustly partitioned system allows partitions with different criticality levels to execute in the same core module, without affecting one another spatially or temporally.

The core module O/S is responsible for enforcing partitioning and managing the individual partitions within that core module.

Specific partitions within an IMA system are scheduled on a fixed, cyclic basis. To assist this cyclic activation, the O/S maintains a major time frame of fixed duration which is periodically repeated throughout the module's runtime operation. Partitions are activated by allocating the partition to one or more partition windows within this major time frame, each partition window being defined by its offset from the start of the major time frame and expected duration. The order of partition activation is defined at configuration time using configuration tables. This provides a deterministic scheduling methodology whereby the partitions are furnished with a predetermined amount of time to access processor resources. Temporal partitioning therefore ensures each partition uninterrupted access to common resources during their assigned time periods.

Each partition has predetermined areas of memory allocated to it. These unique memory spaces are identified based upon the requirements of the individual partitions, and vary in size and access rights. At most, one partition only has write access to any particular area of memory. Memory partitioning is ensured by prohibiting memory accesses (at a minimum, write access) outside of a partition's defined memory areas.

Configuration of all of the partitions throughout the whole system is controlled by the system integrator and maintained according to system level configuration tables.

There are limited partition management services available to the application software.

B.2.3.1.1 Partition Attributes Definition

In order for partitions to be supported within an IMA core module, a set of unique attributes needs to be defined for each partition. These attributes enable the core module O/S to control and maintain the partition's operation.

Partition attributes are as follows:

FIXED ATTRIBUTES

- a. Identifier - uniquely defined on a system-wide basis, and used to facilitate partition activation and message routing.
- b. Memory Requirements - defines memory bounds of the partition, with appropriate code/data segregation.

- c. Period - defines the activation period of the partition, and is used to determine the partition's runtime placement within the core module's overall time frame.
- d. Duration - the amount of processor time given to the partition every period of the partition.
- e. Criticality Level - denotes criticality level of partition.
- f. Communication Requirements - denotes those partitions and/or devices with which the partition communicates.
- g. Partition Health Monitor Table (health monitor reconfigurations) - denotes instructions to the health monitor on the actions required.
- h. Entry Point (i.e., partition initialization) - denotes partition elaboration restart address.

VARIABLE ATTRIBUTES

- i. Lock Level - denotes the current lock level of the partition.
- j. Operating Mode - denotes the partition mode, either IDLE, COLD_START, WARM_START, or NORMAL.

In IDLE mode the partition is not executing any processes within its allocated partition windows.

In COLD_START mode the partition is executing cold start initialization.

In WARM_START mode the partition is executing warm start initialization.

In NORMAL mode the process scheduler is activated.

The resource allocation necessary for each partition is specified at system build time. The corresponding objects are created during the initialization of the partition.

B.2.3.1.2 Partition Control

The O/S starts the application partitions when it enters operational mode.

The resources used by each partition (channels, processes, queues, semaphores, events, etc.) are specified at system build time. The corresponding objects are created during initialization phase of this operational mode, and then the partitions enter normal operating mode.

The Health Monitor supervisory function can restart a partition in response to a fatal fault.

A power failure fault function may force the partitions to terminate.

B.2.3.1.3 Partition Scheduling

Scheduling of partitions is strictly deterministic over time. Based upon the configuration of partitions within the core module, overall resource requirements/availability and specific partition requirements, a time-based activation schedule is generated that identifies the partition windows allocated to the individual partitions. Each partition is then scheduled according to its respective partition window.

The schedule is fixed for the particular configuration of partitions on the processor. The main characteristics of the partition scheduling model are:

- a. The scheduling unit is a partition.
- b. Partitions have no priority.
- c. The scheduling algorithm is predetermined, repetitive with a fixed periodicity, and is configurable by the system integrator only. At least one partition window is allocated to each partition during each cycle.
- d. The core module level O/S exclusively controls the allocation of the resources to the partition.

B.2.3.2 Process Management

Within the IMA concept, a partition comprises one or more processes that combine dynamically to provide the functions associated with that partition. According to the characteristics associated with the partition, the constituent processes may operate concurrently in order to achieve their functional and real-time requirements. Multiple processes are therefore supported within the partition.

An application requires certain scheduling capabilities from the operating system to accurately control the execution of its processes in a manner that satisfies the requirements of the application. Processes may be designed for periodic or aperiodic execution, the occurrence of a fault may require processes to be reinitialized or terminated, and a method to prevent rescheduling of processes is required in order to safely access resources that demand mutually-exclusive access.

A process is a programming unit contained within a partition that executes concurrently with other processes of the same partition. It comprises the executable program, data and stack areas, program counter, stack pointer, and other attributes such as priority and deadline. For references within this specification, the term "process" will be used in place of "task" to avoid confusion with the Ada task construct.

Although enforcing partitioning between constituent processes of a partition is not an IMA requirement, the APEX interface does not preclude designs that operate in this capacity.

The partition level O/S is responsible for managing the individual processes within that partition.

Access to process management functions is via the utilization of APEX services. The set of services called in the application software should be consistent with

the certification/validation criticality level of the associated partition. Partition code executes in User mode only (i.e., no privileged instructions are allowed).

The partition should be responsible for the behavior of its internal processes. The processes are not visible outside of the partition.

B.2.3.2.1 Process Attributes Definition

In order for processes to be supported in IMA, a set of unique attributes needs to be defined for each process within the system. These attributes differentiate between unique characteristics of each process as well as define resource allocation requirements.

A tabular description of the process attributes is given below. Fixed attributes are statically defined and cannot be changed once the partition has been loaded. Variable attributes are defined with initial values, but can be changed through the use of service requests once the system has been started.

FIXED ATTRIBUTES

- a. Name - Defines a value unique to each process in the partition.
- b. Entry Point - Denotes the starting address of the process.
- c. Stack size - Identifies the overall size for the runtime stack of the process. Depending on the selected architecture, multiple stack requirements may be required for each process.
- d. Base Priority - Denotes the capability of the process to manipulate other processes.
- e. Period - Identifies the period of activation for a periodic process. A distinct and unique value should be specified to designate the process as aperiodic.
- f. Time Capacity - Defines the elapsed time within which the process should complete its execution.
- g. Deadline - Specifies the type of deadline relating to the process, and may be "hard" or "soft". For a "hard" deadline, failure of the process to meet that deadline within the specified time period will cause the O/S to take some remedial action. Failure of a process to meet a "soft" deadline is less severe and would typically result in the failure being recorded and processing being allowed to continue.

VARIABLE ATTRIBUTES

- h. Current Priority - Defines the priority with which the process may access and receive resources. It is set to base priority at initialization time and is dynamic at runtime.
- i. Deadline Time - The deadline time is periodically evaluated by the operating system to determine whether the process is satisfactorily completing its processing within the allotted time.
- j. Process State - Identifies the current scheduling state of the process. The state of the process could be either dormant, ready, running or waiting.

Once the process attribute information is defined for each process within the partition, a method is required to provide this information to the O/S. Normally, the linker provides a certain amount of attribute information (i.e., starting address, stack size, etc.) pertinent to the main (or initial) process of the system. Some linkers also allow for the creation of user-defined data and/or code segments during the link procedure.

B.2.3.2.2 Process Control

The system resources required to manage the processes of a partition are statically defined at build time and system initialization.

Processes are created (e.g., resources are allocated) and initialized at partition initialization. This implies that all the processes of a partition be defined in such a way that the necessary resource utilization for each process be determined at system build time. Each process is created only once during the life of the partition.

A partition should be able to reinitialize any of its processes at anytime, and should also be able to prevent a process from becoming eligible to receive processor resources. Certain fault and failure conditions may necessitate a partition to restart or terminate any of its processes. These conditions may arise due to either hardware or software faults, or as a result of a distinct operational phase of the application.

Each process has a priority level, and its behavior may be synchronous (periodic) or asynchronous. Both types of processes can co-exist in the same partition. Any process could be preempted at any time by a process with a higher current priority. The process in the ready state with the highest (most positive) current priority is always executing while the partition is active.

Preemptibility control allows sections of code to be guaranteed to execute without preemption by other processes within the partition. If a process within a section of code is interrupted by the end of a partition window, it is guaranteed to be the first to execute when the partition is resumed.

An O/S may or may not support static or dynamic creation for processes or any other communication mechanisms. This should be transparent to the application software. APEX assumes static process creation.

The mechanisms used by the processes, for inter-process communication and synchronization, are also created during the initialization phase, and are not destroyed.

To become active, a process not only needs to be created, it also needs to be started. At least one process in the partition is started shortly after creation. Active processes can start others, stop themselves or others, and restart as required by the application; this is governed by the process state transition model within the O/S. Processes are never destroyed (i.e., the memory areas assigned to the process are not deallocated).