



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

Technical Specification for Virtual Interoperable Simulation for Tests of Avionics Systems in Virtual or Hybrid Bench

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-247

October 2017

Technical Specification for Virtual Interoperable Simulation for Tests of Avionics Systems in Virtual or Hybrid Bench

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-247

October 2017

FOREWORD

1. This document was prepared by EUROCAE Working Group 97 “Interoperability of virtual avionic components” and was approved by the Council of EUROCAE on 10 November 2017.
2. EUROCAE is an international non-profit making organisation in Europe. Membership is open to manufacturers and users 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 amongst Members of EUROCAE and in collaboration with RTCA Inc, Washington D.C., and/or the Society of Automotive Engineers (SAE), Warrendale PA, U.S.A., through their appropriate committees.
4. EUROCAE performance specifications and other documents 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.
5. Copies of this document may be obtained from:

EUROCAE

9-23 rue Paul Lafargue

93200 SAINT-DENIS

France

Telephone: 33 1 49 46 19 65

Email: eurocae@eurocae.net

Website: www.eurocae.net

TABLE OF CONTENTS

FOREWORD	I
INTRODUCTION	8
SCOPE OF THE DOCUMENT	9
REFERENCE DOCUMENTS	10
CHAPTER 1 GENERAL PRESENTATION	11
1.1 WHAT IS VIRTUAL TESTING	11
1.2 VIRTUAL TESTING ADDED VALUE	11
CHAPTER 2 VIRTUAL TESTING CONCEPT AND DEFINITIONS	13
2.1 VIRTUAL TESTING CONCEPT	13
2.2 DEFINITIONS	14
2.2.1 System under test	14
2.2.2 Virtual Component / Virtual Unit	15
CHAPTER 3 VISTAS WG-97 STANDARD GOAL	18
3.1 PURPOSE OF VISTAS WG-97 STANDARD	18
3.1.1 Frame	18
3.1.2 Philosophy	18
3.2 VISTAS WG-97 PERIMETER	18
3.3 VISTAS WG-97 TECHNICAL ASSUMPTION	18
CHAPTER 4 VISTAS WG-97 DEFINITIONS AND CONCEPTS	19
4.1 VISTAS WG-97 BUSES AND FRAME DEFINITIONS	19
4.1.1 Virtual Test Bench VISTAS WG-97 architecture	19
4.1.2 Data buses	19
4.1.3 Communication buses definition	20
4.1.3.1 Avionics Virtual Bus (AV Bus)	20
4.1.3.2 VC Commands & Controls Bus (VCC Bus)	20
4.1.3.3 Non avionics data Bus (NAD Bus)	21
4.1.4 VISTAS WG-97 frame description	21
4.2 VIRTUAL TEST BENCH OPERATING OVERVIEW	22
4.2.1 Detailed view of VTB Architecture	22
4.2.2 Description of the Virtual communication model	23
4.2.3 Virtual component Configuration and Control	24
4.2.3.1 Physical configuration	24
4.2.3.2 Virtual bench connection through firewall	24
4.2.3.3 Bench and general purpose	24
4.2.3.4 Bench network directly inserted in general network	25
4.2.4 Types of avionics data	25
4.3 VIRTUAL COMPONENT CATEGORIES	26
4.3.1 Application test case	26
4.3.2 Modular avionics case	26
4.3.3 Specifying model	28

4.3.3.1	Global specifying model	28
4.3.3.2	Software specifying model	28
4.3.4	Other categories	29
CHAPTER 5	INTERFACE STRUCTURES AND FORMATS	30
5.1	XML FILES	30
5.1.1	Local delivered file	30
5.1.2	Global built file	30
5.1.3	Network.xml configuration file definition and rules	30
5.1.3.1	Structure of Network.xml configuration file	30
5.1.3.2	Xml tree rules and attributes definition	32
5.1.3.3	Rules	33
5.1.4	Configuration of ARINC 664 virtual bus	33
5.1.4.1	Recommendations for channel names in the Network.xml file	33
5.1.4.2	A664 configuration example	34
5.1.5	Configuration of ARINC 429 virtual bus	37
5.1.5.1	Recommendations for channel names in the Network.xml file	37
5.1.5.2	A429 configuration example	37
5.1.6	ARINC 825 (CAN) virtual bus Configuration	38
5.1.6.1	Recommendations for channel names in the Network.xml file	38
5.1.6.2	ARINC 825 configuration example	38
5.1.7	MIL-STD 1553B virtual bus Configuration	38
5.1.7.1	Recommendations for channel names in the Network.xml file	39
5.1.7.2	MIL-STD 1553B configuration example	39
5.1.8	Discrete virtual bus Configuration	40
5.1.8.1	Recommendations for channel names in the Network.xml file	40
5.1.8.2	Discrete configuration example	40
5.1.9	Analogue virtual bus Configuration	40
5.1.9.1	Recommendations for channel names in the Network.xml file	40
5.1.9.2	Analogue configuration example	41
5.1.10	NAD virtual bus Configuration	41
5.1.10.1	Recommendations for channel names in the Network.xml file	41
5.1.10.2	NAD configuration example	41
5.1.11	Signal Overriding	41
5.1.12	Behaviour models	41
5.2	DATA MESSAGES	41
5.2.1	General consideration for avionics data transportation	41
5.2.2	ARINC 664 virtual bus	41
5.2.2.1	Transport of A664 payload in VISTAS WG-97 frame	41
5.2.2.2	A664 virtual bus Topology	42
5.2.2.3	Failure injection on A664 virtual bus	43
5.2.3	A429 virtual bus	43
5.2.3.1	Transport of A429 labels in VISTAS WG-97 frame	43
5.2.3.2	A429 virtual bus Topology	43
5.2.3.3	Failure injection on A429 virtual bus	43
5.2.4	ARINC 825 (CAN) communication	44
5.2.4.1	Transport of A825 frames in VISTAS WG-97 frame	44
5.2.4.2	ARINC 825 virtual bus Topology	44
5.2.4.3	Failure injection on A825 virtual bus	46
5.2.5	1553B communication	46

5.2.5.1	Transport of 1553B frames in UDP datagram	46
5.2.5.2	1553B virtual bus Topology	46
5.2.5.3	Failure injection on 1553B virtual buses	47
5.2.6	Discrete virtual bus	47
5.2.6.1	Transport of Discrete in VISTAS WG-97 frame	47
5.2.6.2	Discrete virtual bus Topology	47
5.2.6.3	Failure injection of Discrete virtual bus	48
5.2.7	Analogue virtual bus	48
5.2.7.1	Transport of Analogue in VISTAS WG-97 frame	48
5.2.7.2	Analogue virtual bus Topology	49
5.2.7.3	Failure injection of Analogue virtual bus	49
5.2.8	NAD bus	49
5.2.8.1	Transport of NAD in VISTAS WG-97 frame	49
5.2.8.2	NAD bus Topology	50
5.2.8.3	Failure injection of NAD bus	50
5.3	CONTROL MESSAGES	50
5.3.1	Command & Control of models and Virtual Components	50
5.3.1.1	Simulation control description	50
5.3.1.2	Typical control sequences	52
5.3.1.2.1	Model or Virtual components without management of states and time	52
5.3.1.2.2	Model or Virtual components with time management only	53
5.3.1.2.3	Virtual components with state management only	54
5.3.1.2.4	Virtual components with state and time management	55
5.3.2	Preamble	56
5.3.3	Description of control commands	56
5.3.4	Description of state commands	57
5.3.5	Description of time management commands	60
5.3.5.1	Logic synchronization example presentation:	62
5.3.5.2	Synchronization example ACCELERATED:	62
5.3.5.3	Synchronization example SLOWERED:	62
5.3.6	Description of power supply management commands	63
5.4	FAILURE CONTROLLER CAPABILITY ON AVIONICS VIRTUAL BUS (AV BUS)	65
5.5	CONSTRAINTS ON MODELS REALISATIONS	66
5.5.1	Endianness treatment	66
5.5.2	Ethernet addressing rules	66
CHAPTER 6	SPECIFIC TOPICS	67
6.1	QUALITY OF SERVICE	67
6.1.1	General considerations	67
6.1.2	Producer ID	67
6.1.3	QoS Sequence number	68
6.1.4	QoS TimeStamp	68
6.2	DATA ACQUISITION TIMESTAMP	68
6.3	DYNAMIC SIGNAL OVERRIDING (FAILURE INJECTION)	68
6.3.1	Global definition	68
6.3.2	Virtual component failure injection capabilities	68
6.3.2.1	Failure simulation	68
6.3.2.2	Injection of failures on avionics virtual buses	69

6.3.2.2.1	Principle of stopping data transmission and reception	69
6.3.2.2.2	Principle of Dynamic signal overriding	69
6.3.2.3	Failure commands	71
6.3.2.4	Configuration of failure commands	73
6.3.3	Monitoring of data on avionics virtual buses	73
6.4	STATE MACHINE	74
6.4.1	State “not configured”	75
6.4.2	State “configured”	75
6.4.3	State “RUNNING”	75
6.4.4	State “running in steps”	76
6.4.5	State “HELD”	76
6.5	DYNAMIC BEHAVIOUR (I/O BOXES)	77
6.5.1	Avionics I/O function	77
6.5.2	Specific I/O function	77
6.5.3	Virtual component pin programming	77
6.5.4	Dysfunctional modes	78
6.6	TEST BENCH HEALTH MONITORING	78
6.7	VIRTUAL COMPONENT COMPOSITION	78
6.7.1	Definition file general purpose	78
APPENDIX A - ACRONYMS AND GLOSSARY OF TERMS		79
A.1	ACRONYMS	79
A.2	GLOSSARY OF TERMS	82
APPENDIX B - VIRTUAL TESTING APPLICATION FIELD (TESTS CATEGORIES)		85
B.1	DISTRIBUTION OF TESTS DEPENDING ON TEST BENCH TYPES	85
B.2	TARGETED ERRORS DURING USE OF VTB	86
APPENDIX C - VIRTUAL TESTING IMPACT ON DEVELOPMENT PROCESS		87
C.1	LISTS OF ACTORS	87
C.2	DEVELOP AIRCRAFT	89
C.2.1	Aircraft Systems Suppliers Selection	91
C.2.1.1	Objective	91
C.2.1.2	Activities linked to Virtual	91
C.2.2	IMA Platform Supplier selection	91
C.2.2.1	Objective	91
C.2.2.2	Activities linked to Virtual	91
C.2.3	Develop IMA Platform	91
C.2.3.1	Objective	91
C.2.3.2	Activities linked to Virtual	91
C.2.4	Develop Avionics Systems	91
C.2.4.1	Objective	91
C.2.4.2	Activities linked to Virtual	92
C.2.5	Integrate Avionics System(s)	92
C.2.5.1	Objective	92
C.2.5.2	Activities linked to Virtual	92
C.2.6	Set up System Test Bench(es)	92
C.2.6.1	Objective	92

C.2.6.2	Activities linked to Virtual.....	93
C.2.7	Validate System(s)/Function(s)	93
C.2.8	Transverse Activities	93
C.2.8.1	Configuration Management	93
C.2.8.2	Change Request Management	93
C.2.8.3	Traceability Management	93
C.2.9	Calendar Aspects	94
C.2.10	Development and delivery of AS	94
C.2.11	Development and delivery of VC	94
C.2.12	Development and delivery of models of non-IMA equipment	95
APPENDIX D - WG-97 MEMBERSHIP		96
IMPROVEMENT SUGGESTION FORM		97

LIST OF FIGURES

FIGURE 2-1 TEST BENCH FUNCTIONAL DIAGRAM	13
FIGURE 2-2 SYSTEM UNDER TEST COMPONENTS	15
FIGURE 2-3 EXAMPLE OF THE DIFFERENT ELEMENTS.....	16
FIGURE 4-1 VISTAS WG-97 TEST BENCH ARCHITECTURE	19
FIGURE 4-2 ENCAPSULATION EXAMPLE OF A PAYLOAD ON ETHERNET	21
FIGURE 4-3 VISTAS WG-97 FRAME DEFINITION	22
FIGURE 4-4 VIRTUAL TEST BENCH ARCHITECTURE	22
FIGURE 4-5 OSI MODEL AND VIRTUAL MODEL COMPARISON	23
FIGURE 4-6 VIRTUAL BENCH SEGREGATED BY FIREWALL.....	24
FIGURE 4-7 VIRTUAL BENCH SEGREGATED BY NETWORK CARD	24
FIGURE 4-8 VIRTUAL BENCH TOPOLOGY NOT ALLOWED	25
FIGURE 4-9 VIRTUAL COMPONENT INSERTED IN GENERAL PURPOSE NETWORK	25
FIGURE 4-10 MODULAR AVIONICS ARCHITECTURE	27
FIGURE 4-11 VIRTUAL RDC ARCHITECTURE	27
FIGURE 4-12 GLOBAL SPECIFYING MODEL INTEGRATION ON A VIRTUAL TEST BENCH.....	28
FIGURE 4-13 SOFTWARE SPECIFYING MODEL ARCHITECTURE	29
FIGURE 5-1 EXAMPLE OF A REAL A664 TOPOLOGY	34
FIGURE 5-2 VIRTUAL MODEL ASSOCIATED TO THE A664 EXAMPLE	35
FIGURE 5-3 EXAMPLE OF A REAL A429 TOPOLOGY	37
FIGURE 5-4 VIRTUAL MODEL ASSOCIATED TO THE A429 EXAMPLE	37

FIGURE 5-5 VIRTUAL MODEL ASSOCIATED TO THE MIL 1553B EXAMPLE	39
FIGURE 5-6 ENCAPSULATION OF A664 PAYLOAD ON ETHERNET	42
FIGURE 5-7 ENCAPSULATION OF A429 LABELS ON ETHERNET	43
FIGURE 5-8 ENCAPSULATION OF A825 PAYLOAD ON ETHERNET	44
FIGURE 5-9 ENCAPSULATION OF 1553B PAYLOAD ON ETHERNET	46
FIGURE 5-10 DISCRETE ENCAPSULATION ON ETHERNET	47
FIGURE 5-11 ANALOGUE ENCAPSULATION ON ETHERNET	48
FIGURE 5-12 NAD ENCAPSULATION ON ETHERNET	50
FIGURE 5-13 NOMINAL SEQUENCE BETWEEN BENCH CONTROLLER AND VC	52
FIGURE 5-14 VTB CONTROLLER AND MODEL LINKED ONLY WITH TIME MANAGEMENT	53
FIGURE 5-15 BENCH CONTROLLER AND VC LINKED ONLY WITH STATE MANAGEMENT	54
FIGURE 5-16 VTB CONTROLLER AND VC LINKED WITH STATE AND TIME MANAGEMENT	55
FIGURE 5-17 F_MODE_REGISTER COMMAND DESCRIPTION EXAMPLE	56
FIGURE 5-18 R_MODE_REGISTER COMMAND DESCRIPTION EXAMPLE	57
FIGURE 5-19 F_CONFIG COMMAND DESCRIPTION EXAMPLE	57
FIGURE 5-20 R_CONFIG COMMAND DESCRIPTION EXAMPLE	58
FIGURE 5-21 F_STATE COMMAND DESCRIPTION EXAMPLE	59
FIGURE 5-22 R_STATE COMMAND DESCRIPTION EXAMPLE	59
FIGURE 5-23 F_SYNC_REGISTER COMMAND DESCRIPTION EXAMPLE	60
FIGURE 5-24 R_SYNC_REGISTER COMMAND DESCRIPTION EXAMPLE	61
FIGURE 5-25 F_SYNCHRO COMMAND DESCRIPTION EXAMPLE	61
FIGURE 5-26 R_SYNCHRO COMMAND DESCRIPTION EXAMPLE	62
FIGURE 5-27 F_POWER_REGISTER COMMAND DESCRIPTION EXAMPLE	63
FIGURE 5-28 R_POWER_REGISTER COMMAND DESCRIPTION EXAMPLE	64
FIGURE 5-29 F_POWER_SUPPLY COMMAND DESCRIPTION EXAMPLE	65
FIGURE 5-30 R_POWER_SUPPLY COMMAND DESCRIPTION EXAMPLE	65
FIGURE 6-1 FAILURE INJECTION AT EMISSION	69
FIGURE 6-2 FAILURE INJECTION AT RECEPTION	70
FIGURE 6-3 F_INSTRUM_REGISTER COMMAND DESCRIPTION EXAMPLE	71
FIGURE 6-4 R_INSTRUM_REGISTER COMMAND DESCRIPTION EXAMPLE	72
FIGURE 6-5 F_FAILURE COMMAND DESCRIPTION EXAMPLE	73
FIGURE 6-6 R_FAILURE COMMAND DESCRIPTION EXAMPLE	73

FIGURE 6-7 STATES OF THE VIRTUAL COMPONENT.....	74
FIGURE C-1 RELATIONSHIP BETWEEN IDENTIFIED ACTORS & A/C COMPONENTS	88
FIGURE C-2 DEVELOPMENT AIRCRAFT	89

INTRODUCTION

Physical test benches used in the aircraft development process are complex platforms with high initial and recurring costs. They are generally on the critical path of the development and cannot be easily multiplied to increase the validation capacity.

One way to avoid these difficulties can be Virtual Testing approach. It is a promising solution that has already demonstrated its benefits in other industries like Cell phone. However, in the avionics industry, it brings specific challenges as systems are complex, hardware heterogeneous and coming from multiple suppliers with different infrastructures.

Therefore, guidance material with the goal to become an internationally recognized standard is necessary:

- To provide a framework that supports the demonstration of the interoperability of virtual avionics components.
- To define a global process that describes the exchange, integration and exploitation of virtual avionics components, their environments and connectivity.

Virtual Component definition subject aims at improving system simulation models fidelity through the definition of new standards for the representation of its different components such as middleware, operating system or hardware of the equipment under test. It assumes that modelling of the system environment is already covered by the existing practices for physical test benches.

This standard issue doesn't address certification credit as per ED-79A using the virtual testing for validation and verification activities.

SCOPE OF THE DOCUMENT

This document aims to standardize the “Virtual testing” approach.

Two major aspects are considered in this standard:

- The Virtual Component definition.
- The standardization of the interfaces (virtualization rules for the different A/C signals and buses, commands, etc.).

Virtual testing is considered for avionics systems. It's more convenient to have in mind IMA architectures when considering this approach. However, there is no contradiction for widely using it across all kind of computer based systems embedded in the aircraft, commonly called Line Replaceable Unit (LRU).

The term “IMA platform” or “LRU platform” will be used as far as the real avionics equipment is concerned.

Any other component shall be considered as virtual and their classifications/definitions are given further in the document.

Concerning interfaces standardization between components, the use of specific physical layer is not enforced by this document even though Ethernet is considered throughout the document in the requirements specification. It has to be taken as an example.

This document is organized as follows:

- CHAPTER 1 introduces the intended use and added value of Virtual testing,
- CHAPTER 2 describes the Virtual testing concept and definitions,
- CHAPTER 3 makes an overview of Virtual testing goals,
- CHAPTER 4 defines VISTAS WG-97 Standard concepts and definitions,
- CHAPTER 5 defines the Interface structures and formats,
- CHAPTER 6 deals with specific topics, such as Quality of service, Dynamic signal overriding (failure injection), etc.

Given in APPENDIX C, a proposal of how “Virtual testing” impacts the development phase is also provided in a general environment. The description does not aim at reminding all the regular existing activities and artefacts necessary for avionics developments. It focuses on specificities led by the Virtual Testing in the development cycle.

Virtual components in this document are considered as "black box" and the emphasis is made on the exchanges which have to respect the ICDs defined for the real components and its interfaces with the Virtual Test Bench (VTB).

REFERENCE DOCUMENTS

Name	Title	Origin	Reference
[01]	ARINC 825 (CAN) specification	ISO / ARINC	ARINC 825-3
[02]	1553B specification	DoD NATO / MAS	MIL-STD-1553B / STANAG3838
[03]	Understanding the Value of Business Objects (BI) Platform Standardization	CRIMSON Group	-
[04]	Guidance for use of avionics equipment and software in simulators	ARINC	ARINC 610C
[05]	Guidelines for development of Civil Aircraft and Systems	EUROCAE	ED79A
[06]	A Proposed Taxonomy for Software Development Risks for High-Performance computing (HPS) Scientific/Engineering Applications	SEI	CMU/SEI-2006-TN-039
[07]	ARINC SPECIFICATION 664 (AIRCRAFT DATA NETWORK)	ARINC	ARINC 664
[08]	ARINC SPECIFICATION 429	ARINC	ARINC 429

CHAPTER 1

GENERAL PRESENTATION

1.1 WHAT IS VIRTUAL TESTING

In a classical V development, integration phase is performed after detail design and realization. For this step, System integrator had to wait both for hardware and software availabilities.

In the Virtual Testing approach, the system or equipment under test is replaced by a Virtual Component (VC). Virtual Component is a representative simulation of a system, a system of systems, one Avionics equipment, more than one or part of it.

Associated to this approach, Virtual Test Benches (aka Test Rigs) are developed. They intend to enable testers to perform integration tests between VCs.

“Virtual Testing” consists in, prior to real components (equipment or hardware) availabilities:

- Building virtual test benches,
- Integrating Virtual Components,
- Performing Tests on Virtual Components.

1.2 VIRTUAL TESTING ADDED VALUE

The following are the main added value related with standardizing Virtual Component (VC) definition and the interfaces (protocols) between the test bench and the VCs:

- Cost reduction:
 - o Integration through Virtual components:
 - Helps reducing the need of real avionics equipment, associated bench(es) and maintaining and operating them,
 - Simplifies virtual or hybrid test bench architecture (e.g. wiring aspects),
 - Allows having multiple benches at reasonable cost and so improving productivity
 - Enabling early validation of the system under development and obtaining feedback from relevant stakeholders through quick prototyping.
 - o By discovering system & software errors earlier in the development,
 - o Having a library of virtual hardware at your fingertips: e.g. emulators, simulators, oscilloscopes, memory monitors, multi-meters, etc. reusable for other projects.
- Improving final product quality with:
 - o Full regression testing using virtual potentialities, exercising both the object Software and the real Time Operating System, like:
 - Automatic testing,
 - Using capacity to faster or slower real time operation
 - o Verification of system design changes rapidly,
 - o More relevant tests as programmer can run them on the whole representative system,

- Improving schedules by:
 - o Increasing programmer productivity,
 - o Increasing of flexibility by copying virtual test environment,
 - o Testing earlier as programmer have a mean to execute the SW,
- Enable geographically distributed development and testing.

Virtual testing doesn't aim to replace tests done on real equipment. It's just a mean to be more efficient considering software and integration issues.

CHAPTER 2

VIRTUAL TESTING CONCEPT AND DEFINITIONS

2.1 VIRTUAL TESTING CONCEPT

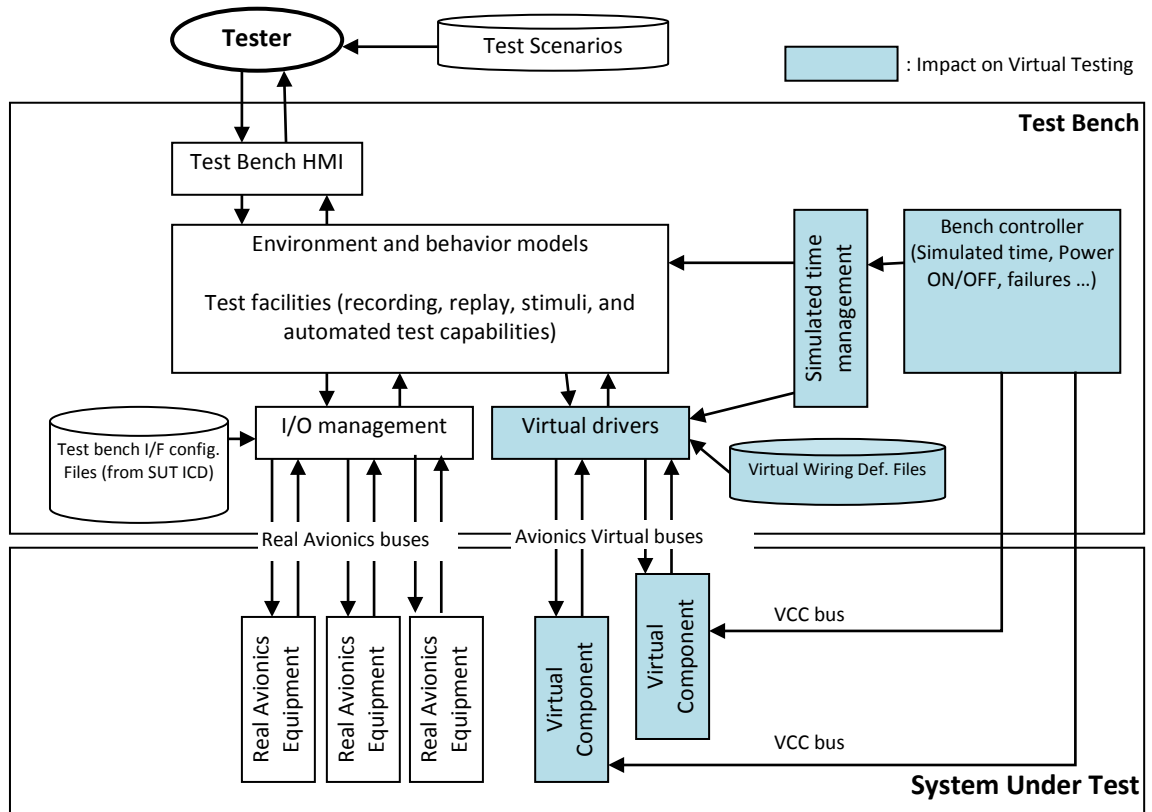


FIGURE 2-1 TEST BENCH FUNCTIONAL DIAGRAM

[Definition: “Virtual Test Bench (VTB)”

A **Virtual Test Bench** (aka VTB) is a bench with no avionics hardware in the loop. The only hardware considered is the bench infrastructure itself (workstations, network, mass storage...)].

[Definition: “Hybrid Test Bench (HTB)”

A **Hybrid Test Bench** (aka HTB) is composed of both real avionics equipment and virtual components.

A Test Bench, either virtual or hybrid, is made up of several elements:

- Management part with:
 - o Control functions (configuration, time management, etc.),
 - o Command functions (test launch, Power On/Off, failure injection, etc.),
 - o Monitoring capacities.
- Environmental models for the SUT,
- Facilities (recording, automated test capabilities, replay functions, test scenarios libraries),
- HMI for the Tester,
- ...]

System Under Test (SUT) on this bench can either be:

- Real Avionics Equipment: classical integration testing.
- Or Virtual Component(s) (VC). A VC hosts one or a set of avionics functions. The aim is to have these functions with the real software to be more relevant. From a logical point of view, the VC may differ from the avionics equipment as it can be a part of one, more than one or a subset of different equipment functions. In that last case, it may not have an equivalent in the real world.
- Or a mix of both (hybrid).

On Virtual Test Bench, virtual drivers and UDP/IP wiring replace real drivers and real wiring to interface Virtual Components. Test Bench “virtual wiring” is detailed within Virtual Wiring Definition Files which provide information on which message or bus is transported on which UDP/IP packet. This operation is detailed in CHAPTER 5.

For avionics buses based on the Internet Protocol (IP), it is also recommended to reuse the real A/C network-addressing plan. Test Bench Virtual Wiring Definition Files (VWDF) are produced from Aircraft Interface Control Document (ICD) and from Test Benches architecture and network topology.

VWDF follow a standard format that all exchanged virtual components must comply with.

2.2 DEFINITIONS

2.2.1 System under test

[Definition: “System Under Test (SUT)”

Next figure depicts the components considered in the standard in order to build up a System Under Test (SUT). Colour convention is the following one:

- *Blue depicts items only relevant to Real Avionics Equipment,*
- *Green corresponds to items only relevant to Virtual Components,*
- *Orange corresponds to items used in both cases.*

Indeed, a Real Avionics Equipment, whether it's an LRU or an IMA-based one consists of:

- *An Avionics Platform (i.e. the hardware resource),*
- *One Avionics Equipment configuration,*
- *None, one or several Avionics Software (indeed, a RDC will not host any Avionics Software beyond that part of the Avionics Platform, on the other hand a CPM may host several ones).]*

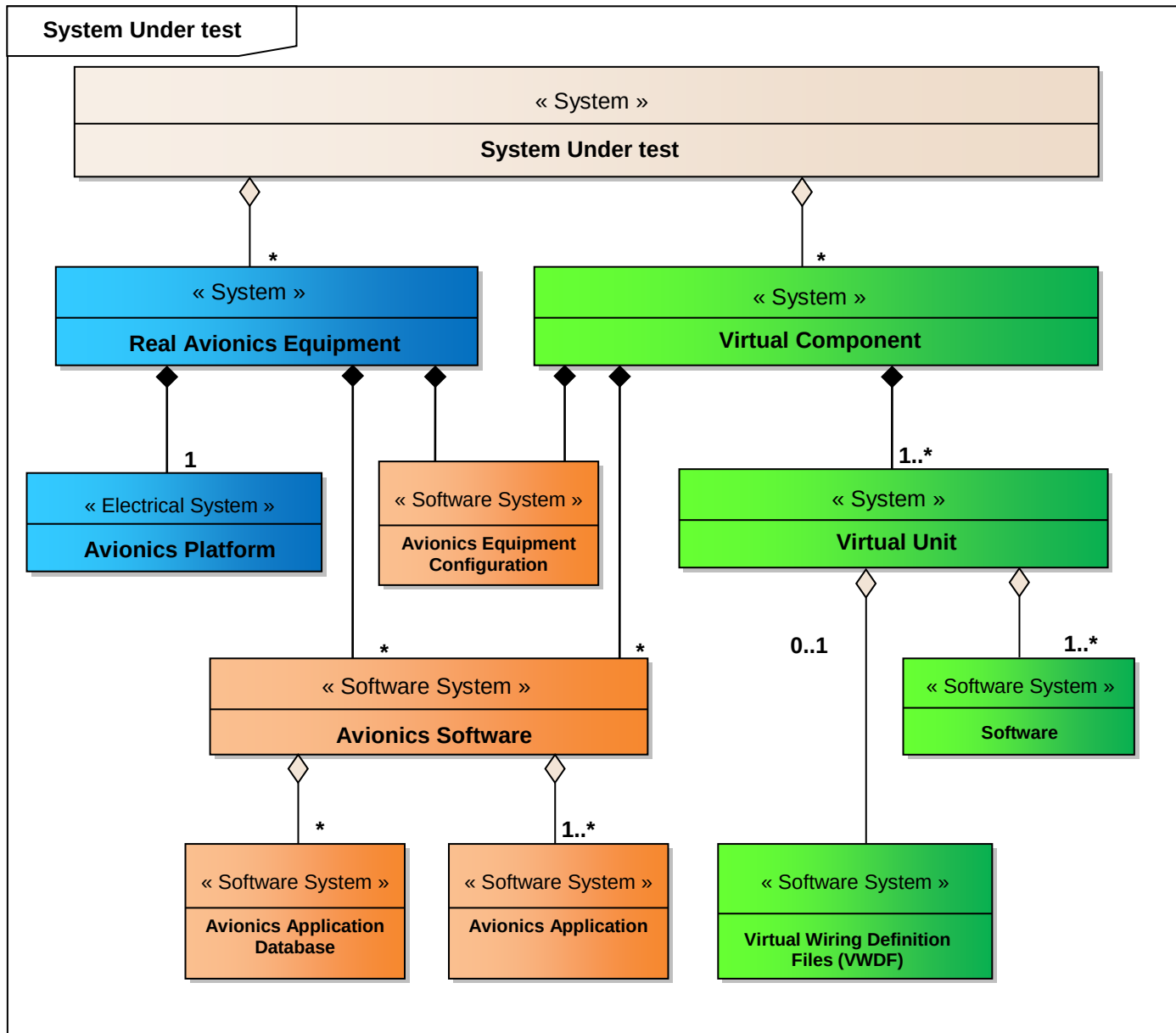


FIGURE 2-2 SYSTEM UNDER TEST COMPONENTS (UML)

2.2.2 Virtual Component / Virtual Unit

The goal of this standard is to propose a common way to integrate components which represent an Aircraft function in a simulated world.

To achieve this, other kinds of components may be needed to perform the required simulated representation of the function which can be part of / whole / several A/C Systems.

All virtual test bench components shall be plugged in the same way, no matter its type: A/C Systems (part of, whole or several systems), a function needed for the bench (e.g. mass storage, spying tools...) or physical or topological objects (ATA00, network map...).

Therefore, from a logical point of view, all components are integrated in the same way.

Definition: “Virtual Component (VC)”

Virtual components are the basic building blocks of a virtual test bench.

For reuse purpose and to be representative of Real Avionics Equipment, a virtual component can be broken down in elementary parts called Virtual Unit.]

[Definition: “Virtual Unit (VU)”

Virtual Unit (VU) is the simulation of a computational unit with its embedded software. The VU embeds the application AND the hardware simulation. The VU does not necessarily reflect complete referenced A/C equipment: several VU may be combined to represent the whole real equipment (e.g. one VU per IMA partition).]

A VU can be obtained through re-host, retarget or simulation (i.e. modelling).

It may include the Virtual Wiring Definition File (VWDF), consistent with the test bench network architecture.

So, Virtual Component may be composed of:

- Virtual Unit(s)
- Behavioural model (BM), which is the modelling of a function needed for the bench.

From a logical point of view, there is no difference between a VU and a BM as both are encapsulated in a VC.

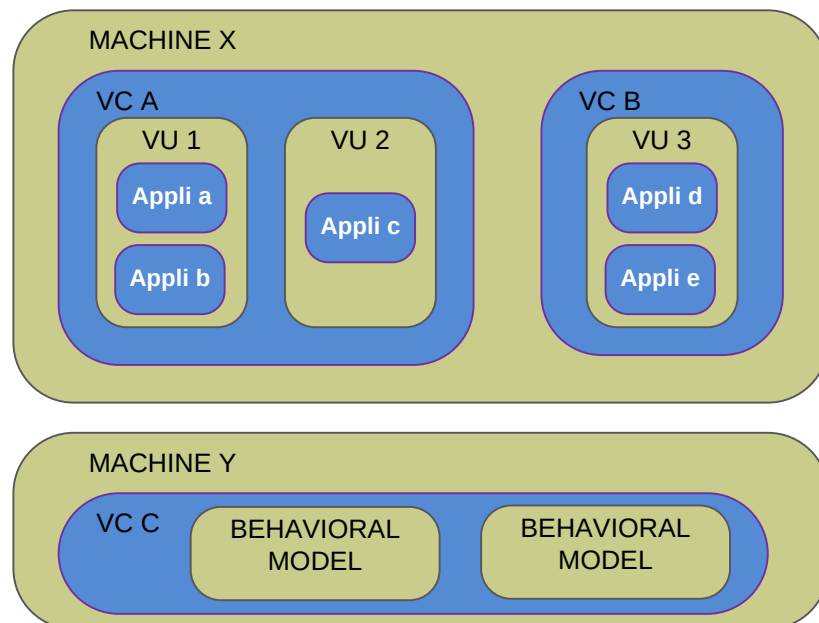


FIGURE 2-3 EXAMPLE OF THE DIFFERENT ELEMENTS

Virtual Unit can be based upon the final Avionics Application Software and Avionics Equipment simulation or upon pure prototyping software in order to support specification phase also known as Specifying Model.

Fidelity of VU vs Real Avionics Equipment relates to the reuse of Real Avionics Equipment artefacts (Application Software, Databases, and Configuration Files).

NOTE 1: *When the Real Avionics Equipment is a LRU, the boundary between the Avionics Software and the computational part of the VU may not be so clear depending on the volume of functions assumed by the hardware platform. Thus, LRU VC includes Avionics Software and Equipment simulation in a single delivery.*

NOTE 2: *Avionics Equipment Configuration data used in both Real and Virtual cases are processed by dedicated tools to generate the relevant artefact for either Real Avionics Equipment or Virtual Component configuration.*

[Definition: “Behavioral Model”

A specific category of VCs that consists in providing components that have, from the AVB and NAD, bus point of view the same behaviour as those of the real component for test objectives purposes .]

CHAPTER 3

VISTAS WG-97 STANDARD GOAL

3.1 PURPOSE OF VISTAS WG-97 STANDARD

3.1.1 Frame

This document proposes a solution to integrate Virtual Components in a “plug and play” manner. By extension, the hybridation between real and virtual components is also described throughout each technical description.

The target of this standard is to allow the use of virtual testing for avionics, in the aeronautical industrial frame encountered worldwide.

3.1.2 Philosophy

- The VISTAS WG-97 Standard does not provide rules to build nor develop Virtual Unit, Component or Virtual or Hybrid Test Bench.
- The VISTAS WG-97 Standard considers that the smallest part of a bench to be considered is the Virtual Component.
- The VISTAS WG-97 Standard considers the virtual component as a black box that falls under the responsibility of its provider.
- However, the VISTAS WG-97 Standard defines for a Virtual Components:
 - o How it shall exchange data,
 - o How it is controlled,
 - o Its states.

Between each other so that plug and play and representative simulation of part of/whole/several avionics system may be properly set.

- The VISTAS WG-97 Standard defines functionality at data exchange level to ensure that:
 - o Data exchange is representative regarding test objectives,
 - o Test bench made of VISTAS WG-97 Virtual Components is consistent with real Avionics,
 - o All real test bench capabilities are met by a VISTAS rev. A HTB.

3.2 VISTAS WG-97 PERIMETER

VISTAS WG-97 Standard defines protocols on:

- Exchanging Avionics data,
- Exchanging simulation data,
- Exchanging other than Avionics Data (Non Avionics Data), see § 5.2.8,
- Ensuring consistent construction of a Virtual or Hybrid Test Bench,
- Enabling enough capabilities to satisfy tests objectives.

3.3 VISTAS WG-97 TECHNICAL ASSUMPTION

Virtual Components are communicating through Ethernet.

CHAPTER 4

VISTAS WG-97 DEFINITIONS AND CONCEPTS

4.1 VISTAS WG-97 BUSES AND FRAME DEFINITIONS

4.1.1 Virtual Test Bench VISTAS WG-97 architecture

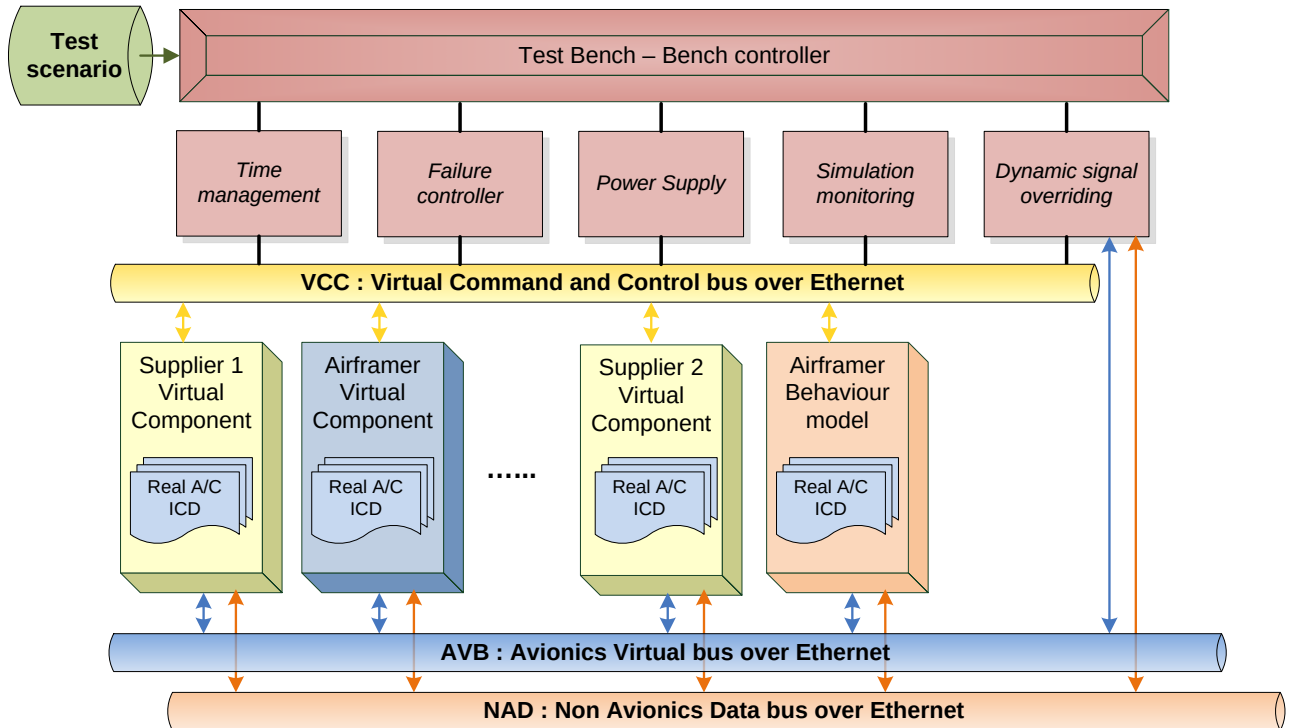


FIGURE 4-1 VISTAS WG-97 TEST BENCH ARCHITECTURE

4.1.2 Data buses

VISTAS WG-97 defines three buses over Ethernet through which a VC interfaces other VCs, in the frame of virtual test bench:

- Virtual avionics bus for Avionics protocols covered by this document,
- Command and Control bus to operate properly a Virtual component,
- Non avionics data bus for environment flow.

These buses are non-exclusive: a Virtual Component may embed other kind of simulation protocols for communication with other standards (e.g. FMI, HLA...).

For easier understanding, in the rest of the document, when not specified, VC Command and Control bus and Non avionics Data bus are mentioned as simulation bus reflecting the opposition of the data transported: avionics data vs non avionics data.

These buses are logical and they all use data transfer protocol defined in CHAPTER 5.

All communication is made through Ethernet (for wired, use 100Mbps or higher) and uses IPV4 or V6 addresses. It is not mandatory to use separate physical Ethernet media to transport those buses (e.g. use of VPN).

When built as described on the previous figure, same tests can be run for Physical Tests, Hybrid and Virtual Tests (manual and automated ones - with restrictions described later on).

4.1.3 Communication buses definition

4.1.3.1 Avionics Virtual Bus (AV Bus)

[Definition: “Avionics Virtual Bus (AVB)”

The aim of avionics virtual bus is to support real avionics data transfer representation. The current types of avionics data that may be transferred on this bus are defined in CHAPTER 5:

- ARINC 664 (AFDX)
- ARINC 429
- ARINC 825 (CAN)
- Discrete signals
- Analogue signals

Preliminary release with this issue, still under construction

- Mil-Std-1553

Not yet defined and not exhaustive for

- Serial buses
- ARINC 818
- Wireless communications]

Remarks:

- When instantiated on a specific avionics bus, *Avionics* is replaced by the bus name: *A664 virtual bus* for ARINC 664 Aircraft Data Network, *A429 virtual bus* for ARINC 429, *Discrete virtual bus* for discrete signals, etc.
- Specific categories such as synchro and resolver hardwares and connectivity are not treated in this issue.

4.1.3.2 VC Commands & Controls Bus (VCC Bus)

[Definition: “VC Commands & Controls Bus (VCC)”

The VCC bus is the bus on which circulates messages that are needed to drive any VC present within the VTB. Those commands cover all Bench controller commands.

This includes:

- Commands of the VC Configuration,
- Commands of the VC State Machine,
- Commands for VC Network modification,
- Commands for VC Health Monitoring,
- VC specific commands].

4.1.3.3

Non avionics data Bus (NAD Bus)**[Definition “Non avionics data Bus (NAD)”**

The NAD bus is the bus on which circulates messages that are neither part of a specific A/C ICD nor part of dedicated command and control messages of the Virtual Component.

- Environment data
- Behavioural Model data and corrupted information

There are two types of environment data (not exhaustive lists):

- Internal to A/C produced by behaviour model:
 - o A/C physics
 - o Fuel
 - o Hydraulics
 - o Electricals
 - o Mechanicals
 - o ...
- External to the A/C:
 - o Atmospheric data
 - o Weather
 - o Traffic
 - o ATC and Comm.
 - o Tactical data, (Radar emissions, IR signature)
 - o Video streaming...]

4.1.4

VISTAS WG-97 frame description

All VISTAS WG-97 frames shall be encapsulated in the same way in the protocol used on the Virtual testing system. In case of Ethernet, the encapsulation is explained in the figure hereafter:

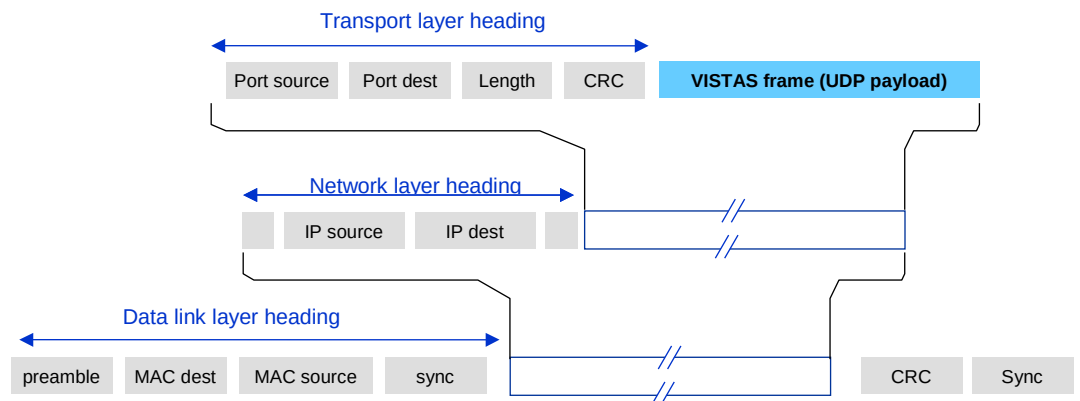


FIGURE 4-2 ENCAPSULATION EXAMPLE OF A PAYLOAD ON ETHERNET

A VISTAS WG-97 message is identified by its multicast/unicast destination address. All messages to this address have the same structure.

A VISTAS WG-97 message is basically a UDP message. VISTAS WG-97 frame shall have the hereafter structure:

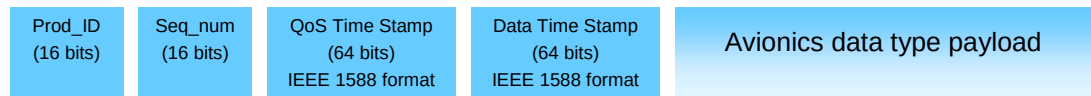


FIGURE 4-3 VISTAS WG-97 FRAME DEFINITION

Where:

- The content of all four header fields is optional whereas the fields are required in the frame,
- QoS Time Stamp is the time of the packet emission on Ethernet,
- Data Time Stamp is the time of the first data computed (e.g. by a VC).

These header fields are detailed in § 6.1 and the avionics data payload in § 5.2 for each type of avionics data listed in § 4.1.3.1.

Time is expressed as a TAI time in IEEE 1588 format with 1µs precision.

Resolution related to the timetag of each parameter (e.g. in case a VC is transferring large size of data) will be considered in a further issue of the standard.

4.2 VIRTUAL TEST BENCH OPERATING OVERVIEW

4.2.1 Detailed view of VTB Architecture

A detailed view of components of virtual test bench (VTB) logical architecture is illustrated in FIGURE 4-4.

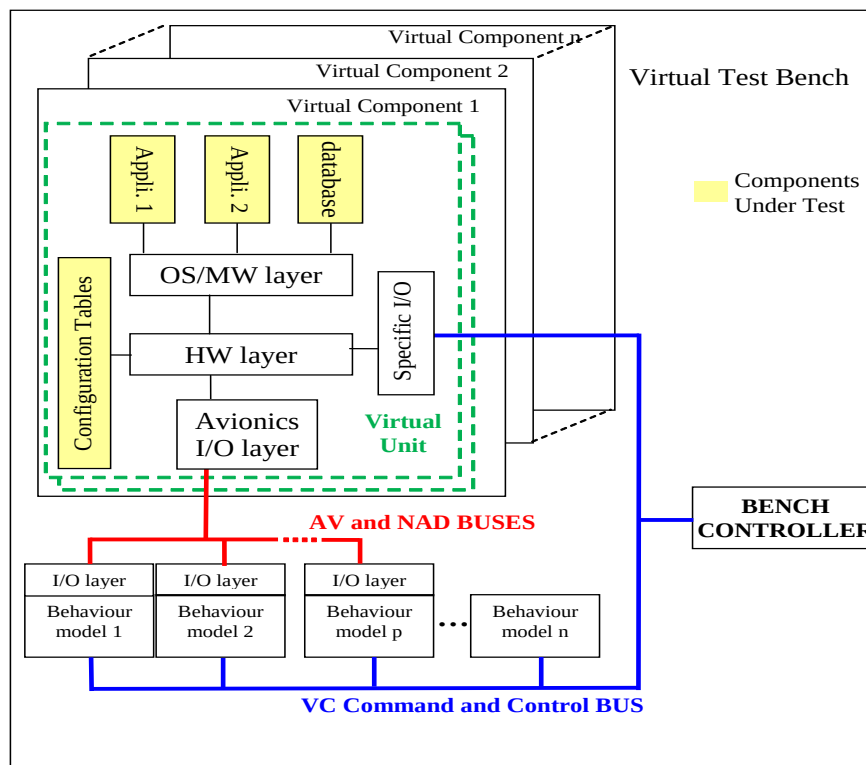


FIGURE 4-4 DETAILED VIEW OF VTB ARCHITECTURE

The VTB can be used for several purposes:

- Integration test bench. In this case the components under test are the Virtual Component, which may be composed of one or several Virtual Units.
- Specification test bench. In this case the component under test is a Virtual Component called a specifying model (see § 4.3.3 for further details)

4.2.2

Description of the Virtual communication model

Reminder: The physical bus that supports the three types of logical buses (AVB, VCC and NAD) is the Ethernet bus.

The communication model used is based on the transport layer (TCP, UDP) of the TCP/IP model. The following figure shows the model layer, in front of the layers of the OSI model:

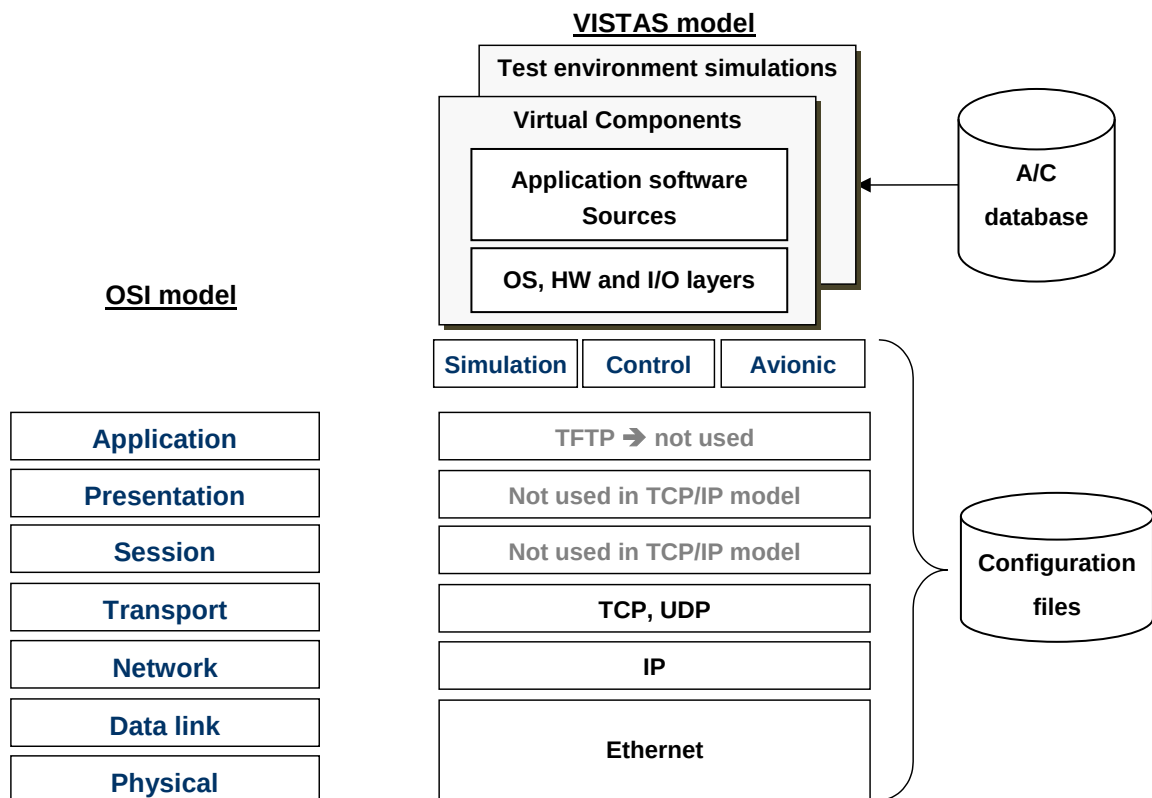


FIGURE 4-5 OSI MODEL AND VIRTUAL MODEL COMPARISON

Multicast IP is mostly used to transport data, especially Avionics data.

For each category of data flow:

1. **Avionics Bus (AVB):** this bus is used to encapsulate Avionics data flow in VISTAS WG-97 frames. The communication topology of the *Avionics Virtual bus* that supports the Avionics data flow is described in § 5.2.
The way to introduce errors on avionics data flow is described in § 6.3.
2. **Control layer (VCC):** this layer is used to control and command Virtual components. Bench control data flow on the VC Command and Control bus is described in § 5.3.
3. **Test environment Simulation layer (NAD)** is used to encapsulate external data of the simulated aircraft. The communication topology of the *NAD bus* that supports the NAD flow is described in § 5.2.8.

Reminder: VISTAS WG-97 is not proposing process for VC development. As a consequence, the software API of the communication layers is not treated in this standard. The Virtual communication model is presented to explain the specification organization and how Ethernet frames are constructed. The implementation details of the Virtual communication layers are left to the responsibility of the Virtual Component supplier.

4.2.3 Virtual component Configuration and Control

Virtual Test Bench (VTB) needs a bench controller to configure and control each Virtual Component connected on the VCC bus.

4.2.3.1 Physical configuration

The following considerations must be taken into account to limit possibilities for network topology of virtual benches:

- The avionics virtual bus aims to connect real time simulated systems with predictive performance. The expected performance on data exchange is most of the time incompatible with a merger between virtual bench network and company general-purpose network, on which data flows are not mastered.
- Company rules for networking often forbid usage of multicast IP addresses.

As a consequence, two categories of physical configurations are treated in this paragraph:

4.2.3.2 Virtual bench connection through firewall

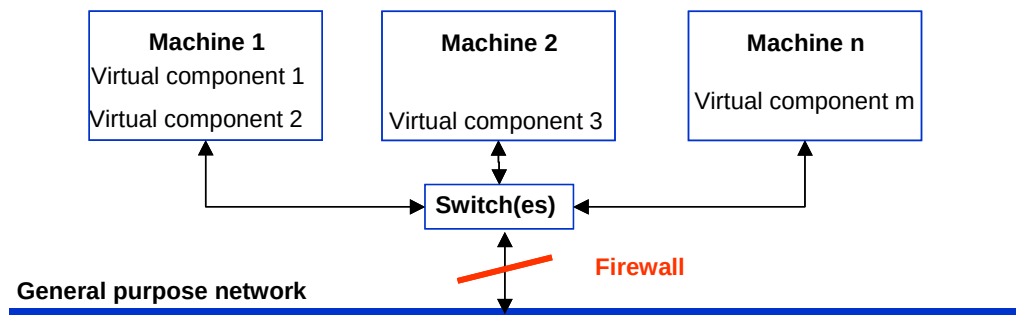


FIGURE 4-6 VIRTUAL BENCH SEGREGATED BY FIREWALL

In this topology, the segregation between avionics virtual bus and general purpose network is ensured by a firewall on the company network or by configuring the switch of the avionics virtual bus.

4.2.3.3 Bench and general purpose

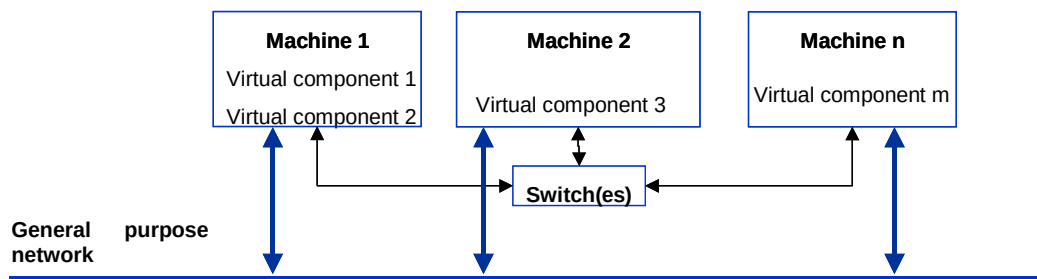


FIGURE 4-7 VIRTUAL BENCH SEGREGATED BY NETWORK CARD

One possible network solution for this topology: the segregation between avionics virtual bus and general purpose network is ensured by usage of distinct Ethernet boards on machines that host virtual components.

4.2.3.4

Bench network directly inserted in general network

The avionics communication specified in this document may not be compliant with the following physical configuration, mainly because multicast IP addressing is used for avionics communication and that this addressing mode is not allowed on most of general purpose networks.

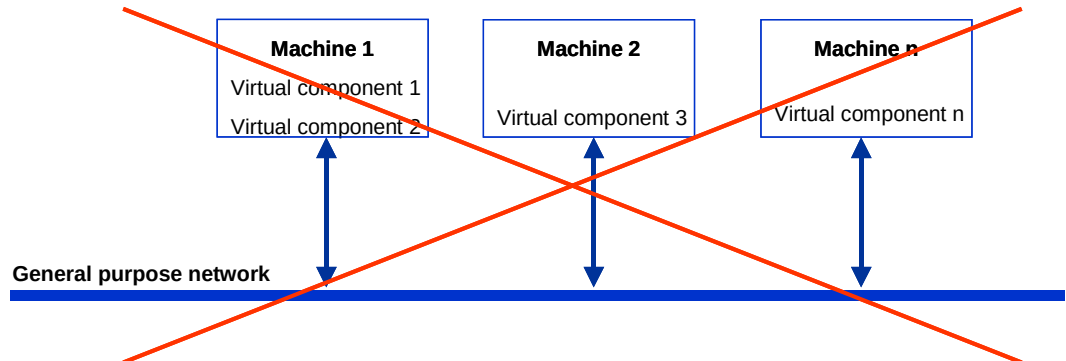


FIGURE 4-8 VIRTUAL BENCH TOPOLOGY NOT ALLOWED

Nevertheless, in some simple virtual benches configurations, it may be interesting to insert virtual components in the general-purpose network, in order to merge virtual component with developers' environments.

For that, it is recommended to insert virtual component through a gateway that performs translation of multicast addresses into unicast addresses. This gateway shall ensure data repetition on unicast addresses linked to a same multicast address:

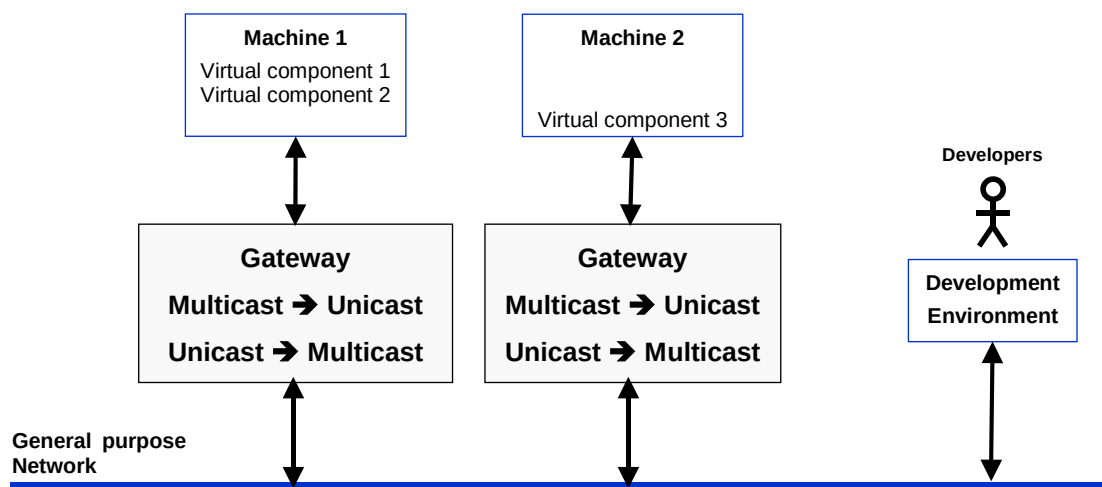


FIGURE 4-9 VIRTUAL COMPONENT INSERTED IN GENERAL PURPOSE NETWORK

The Gateway that ensures addresses translation from multicast to unicast is not treated in this document.

4.2.4

Types of avionics data

The following types of avionics data are considered in release A of this document: ARINC 664 (AFDX), ARINC 825 (CAN), ARINC 429, MIL-STD 1553B (proposal), discrete and analogues.

The other types of avionics data (RS232, A/C Ethernet, etc.) will be addressed in a further issue of this document.

4.3 VIRTUAL COMPONENT CATEGORIES

A classification, not exhaustive, is proposed here after. A component may not fall into one of these categories but in most of the case, its integration as a VC suits completely this standard.

4.3.1 Application test case

The Virtual component is made of the components listed hereafter:

- One or several Application software. It is the part of the virtual component which is under test. The application software present in virtual component must be built from the same source code files as the application software present in real equipment.

In case the application software present in virtual component is:

- o Identical to the one present in real equipment (same binary source), the application software will be qualified as re-hosted,
- o Otherwise, when the application software present in virtual component is built from recompiled source, it will be qualified as retargeted application software.

- Configuration tables if needed (see § 4.3.2 for further details).
- Database (if available on real equipment) retargeted or re-hosted

Virtual Component hosts the following functions:

- Avionics I/O function (i.e. receive or transmit avionics data through Virtual Avionics Bus),
- Specific I/O function (i.e. receive or transmit environmental data through NAD Bus),
- HW function,
- MW/OS function.

No restriction is applied on the delivery format of the Virtual component.

The Virtual Component is configured through simulation configuration tables. The aim of these tables is to configure simulation options of Virtual Component. Simulation configuration tables are used to enable the connection of Virtual Component to test bench (see CHAPTER 5) and for the "APPLY CONFIG" command (see § 5.3.1 "Command & Control of models and Virtual Components" and § 6.4 "State machine" for further details).

The Virtual Test Bench shall be able to host several Virtual Components simultaneously.

4.3.2 Modular avionics case

In Modular avionics architecture, execution platform functioning can be configured through configuration tables (CT). Therefore, in modular avionics architecture, the same configuration tables as for real equipment should be used to configure the functioning of the Virtual Component.

The parameters included in the configuration tables are mainly associated to the Application Software. They are defined to configure the core-processing module which is the execution platform in the IMA context. In that case, the behaviour validation is done on both the Application Software and its configuration tables.

Consequently, for virtual component under modular avionics architecture, the parts under test are the application software and the configuration tables.

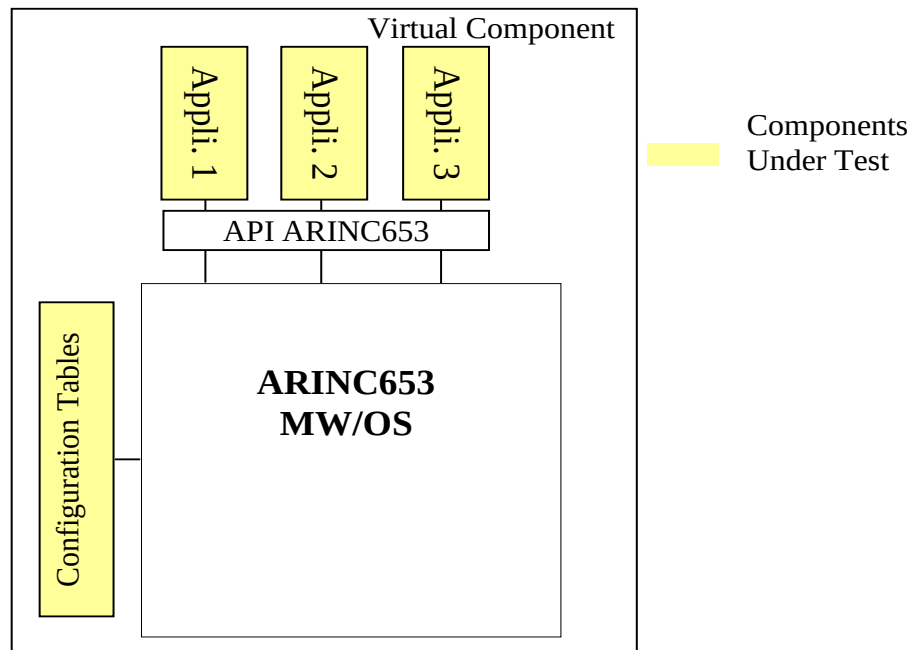


FIGURE 4-10 MODULAR AVIONICS ARCHITECTURE

Moreover, 2 types of IMA components should be considered:

- Core Processing Modules whose architecture of corresponding virtual component is illustrated in [FIGURE 4-10](#) above.
- Remote Data Concentrators (RDC) whose function is to be a gateway between I/O on various transport lines (ARINC 429, ARINC 825, Analogue / Discrete in the present example) and the A/C Data Network based on Ethernet technology. The corresponding virtual component can be seen as a special Virtual Unit (VU) allowing testing of real RDC configuration files. A functional architecture of it can be illustrated by [FIGURE 4-11](#).

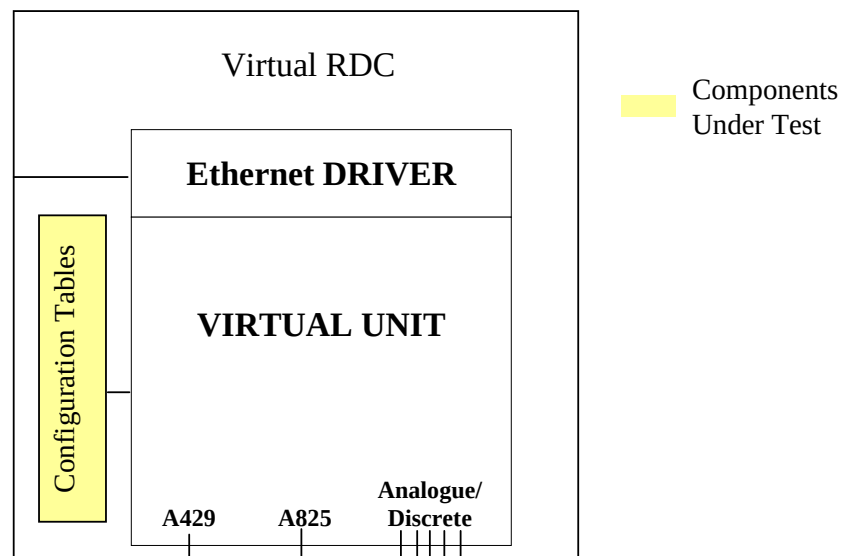


FIGURE 4-11 VIRTUAL RDC ARCHITECTURE

NOTE: The Virtual Unit includes generic RDC software.

Considering CT volume and complexity, CT are usually produced by generation tools from the definition Database used to produce the real CT.