The Open Group Guide

FACE™ Conformance Contributions to Airworthiness, Version 1.0

How Artifacts and Activities for FACE Conformance Contribute to Flight Certification Evidence

FACE™

Future Airborne Capability Environment™

THE Open GROUP

A Technical Report prepared by the FACE Consortium Enterprise Architecture Airworthiness Committee and The Open Group

This document is provided "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

Any publication of The Open Group may include technical inaccuracies or typographical errors. Changes may be periodically made to these publications; these changes will be incorporated in new editions of these publications. The Open Group may make improvements and/or changes in the products and/or the programs described in these publications at any time without notice.

Should any viewer of this document respond with information including feedback data, such as questions, comments, suggestions, or the like regarding the content of this document, such information shall be deemed to be non-confidential and The Open Group shall have no obligation of any kind with respect to such information and shall be free to reproduce, use, disclose, and distribute the information to others without limitation. Further, The Open Group shall be free to use any ideas, concepts, know-how, or techniques contained in such information for any purpose whatsoever including but not limited to developing, manufacturing, and marketing products incorporating such information.

If you did not obtain this copy through The Open Group, it may not be the latest version. For your convenience, the latest version of this publication may be downloaded at www.opengroup.org/library.

The Open Group Guide

FACE™ Conformance Contributions to Airworthiness, Version 1.0:
How Artifacts and Activities for FACE Conformance Contribute to Flight Certification Evidence

ISBN: 1-957866-12-3

Document Number: G230

Published by The Open Group, March 2023.

Comments relating to the material contained in this document may be submitted to:

The Open Group, 177 Huntington Avenue, Suite 1703, PMB 54820, Boston, MA 02115-3153, US

or by electronic mail to:

ogface-admin@opengroup.org

Contents

Preface

The Open Group

The Open Group is a global consortium that enables the achievement of business objectives through technology standards. With more than 900 member organizations, we have a diverse membership that spans all sectors of the technology community – customers, systems and solutions suppliers, tool vendors, integrators and consultants, as well as academics and researchers.

The mission of The Open Group is to drive the creation of Boundaryless Information Flow™ achieved by:

- Working with customers to capture, understand, and address current and emerging requirements, establish policies, and share best practices
- Working with suppliers, consortia, and standards bodies to develop consensus and facilitate interoperability, to evolve and integrate specifications and open source technologies
- Offering a comprehensive set of services to enhance the operational efficiency of consortia
- Developing and operating the industry's premier certification service and encouraging procurement of certified products

Further information on The Open Group is available at www.opengroup.org.

The Open Group publishes a wide range of technical documentation, most of which is focused on development of Standards and Guides, but which also includes white papers, technical studies, certification and testing documentation, and business titles. Full details and a catalog are available at www.opengroup.org/library.

The Open Group FACE™ Consortium

The Open Group Future Airborne Capability Environment™ Consortium (the FACE™ Consortium), was formed as a government and industry partnership to define an open avionics environment for all military airborne platform types. Today, it is an aviation-focused professional group made up of industry suppliers, customers, academia, and users. The FACE Consortium provides a vendor-neutral forum for industry and government to work together to develop and consolidate the open standards, best practices, guidance documents, and business strategy necessary for acquisition of affordable software systems that promote innovation and rapid integration of portable capabilities across global defense programs.

Further information on the FACE Consortium is available at www.opengroup.org/face.

This Document

This Technical Report identifies artifacts and activities related to achieving Future Airborne Capability Environment™ (FACE) Conformance that can contribute to the evidence of airworthiness. After identifying key standards for civilian and military flight certification, this document focuses on mappings from FACE artifacts to requirements in the MIL-HDBK-516C “Airworthiness Certification Criteria” standard and in DO-178C “Software Considerations in Airborne Systems and Equipment Certification”. This document will benefit system integrators planning to incorporate FACE Units of Conformance (UoCs) as well as the developers and suppliers of the UoCs, as an aid for planning and development of their software. This guidance can reduce the cost and effort of achieving FACE Conformance in addition to flight certification, ultimately reducing the acquisition costs of airworthy systems that incorporate FACE UoCs.

This document was developed and is maintained by The Open Group FACE Consortium.

Trademarks

ArchiMate, DirecNet, Making Standards Work, Open O logo, Open O and Check Certification logo, Platform 3.0, The Open Group, TOGAF, UNIX, UNIXWARE, and the Open Brand X logo are registered trademarks and Boundaryless Information Flow, Build with Integrity Buy with Confidence, Commercial Aviation Reference Architecture, Dependability Through Assuredness, Digital Practitioner Body of Knowledge, DPBoK, EMMM, FACE, the FACE logo, FHIM Profile Builder, the FHIM logo, FPB, Future Airborne Capability Environment, IT4IT, the IT4IT logo, O-AA, O-DEF, O-HERA, O-PAS, Open Agile Architecture, Open FAIR, Open Footprint, Open Process Automation, Open Subsurface Data Universe, Open Trusted Technology Provider, OSDU, Sensor Integration Simplified, SOSA, and the SOSA logo are trademarks of The Open Group.

Java is a registered trademark of Oracle and/or its affiliates.

Linux is the registered trademark of Linus Torvalds in the US and other countries.

MISRA C is a registered trademark of MISRA Consortium.

POSIX is a trademark of the Institute of Electrical and Electronic Engineers, Inc.

All other brands, company, and product names are used for identification purposes only and may be trademarks that are the sole property of their respective owners.

Acknowledgements

(Please note affiliations were current at the time of approval.)

The Open Group gratefully acknowledges the contribution of the following people in the development of this document:

- Ned Bent
- Ben Brosgol
- Glenn Carter
- Maria Christiano
- Stephen DiCamillo
- Chris Edwards
- Karen Limardo
- Mike Linnig
- Kristopher Riley
- Joel Sherrill
- Stephen Simi
- Eric Smith
- Bill StClair
- Alicia Taylor
- Jeff VanDorp
- Joe Wlad

The following individuals actively served on the Enterprise Architecture Airworthiness Committee that oversaw this work:

- Ben Brosgol – AdaCore
- Marie Christiano – WisEngineering
- Chris Crook – Army PEO Aviation
- Stephen Di Camillo – LDRA Tech
- Jeffry Howington – Collins
- Dave Johnson – GTRI, NAVAIR

- Karen Limardo – BAE Systems
- Michael Linnig – Raytheon
- Michael Majors – DCS
- Fred Maymir-Ducharme – IBM
- Ron O’Guin – OAR
- Steven Price – Army CCDC AvMC
- Kristopher Riley – CS Communication & Systems, Inc.
- Stephen Simi – TES-SAVI
- Dudrey Smith – AdaCore
- Eric Smith – L3Harris
- Todd Stempel – ENSCO Avionics
- Alicia Taylor – Army PEO Aviation
- Matthew Tkac – CS Communication & Systems, Inc.
- Joyce Tokar – Raytheon
- Mike Tolfree – ENSCO Avionics
- Steve VanderLeest – The Boeing Company
- Jeff VanDorp – GE Aviation

About the Authors

Steven H. VanderLeest is the Boeing Linux® Chief Technologist at the Boeing Company. He holds a PhD from the University of Illinois and was formerly a Professor of Engineering at Calvin University. He is a senior member of the IEEE. He has published on avionics safety and security in the *IEEE Aerospace and Electronic Systems Magazine*, the IEEE/AIAA Digital Avionics Systems Conference, and SAE Aerotech. He has served as principal investigator for Small Business Innovation Research contracts with the US Navy, Army, and DARPA. Dr. VanderLeest currently chairs the FACE Enterprise Architecture Airworthiness Committee and served as the editor of this document.

Ben Brosgol is a senior member of the technical staff at AdaCore. He has been involved with programming language design and implementation throughout his career, concentrating on languages and technologies for high-assurance systems with a focus on Ada and safety certification (DO-178B/C). Dr. Brosgol is Vice-Chair of The Open Group FACE Consortium Technical Working Group.

Karen Limardo is the Software Engineering Lead for the Tactical Systems Business Area at BAE Systems. She has over 30 years of experience in embedded software engineering development life-cycle including requirements, design, code, test, integration, qualification, and maintenance. She currently leads the FACE development efforts for the Tactical Systems business area and is responsible for numerous TSO-C166 and TSO-C112 certified Identification Friend or Foe (IFF) products which follow the DO-178 software life-cycle process.

Matthew Tkac is President and Chief Engineer at CS Communication & Systems, Inc. He is certified by the International Council of Systems Engineering (INCOSE) as an Expert Systems Engineering Professional. He directs engineering teams on software qualification and certification efforts in various industries. He has over 34 years of experience in systems, hardware, and software qualification and certification and has presented on the topics of systems engineering, test strategies, test automation, testing coverage, requirements development, and change management.

Referenced Documents

The links below were valid at the time of writing but cannot be guaranteed for the future.

ARINC ARINC Specification 653 (ARINC 653): Avionics Application Software Standard Interface: Safety-Critical Avionics Real-Time Operating Systems (RTOS), published by ARINC, August 2019; refer to: https://www.aviation-ia.com/products/653p0-2-avionics-application-software-standard-interface-part-0-overview-arinc-653

C207 FACE™ Technical Standard, Edition 3.1, a standard of The Open Group (C207), published by The Open Group, July 2020; refer to: www.opengroup.org/library/c207

DoD Airworthiness Certification Criteria, published by the US Department of Defense, MIL-HDBK-516C, December 2014; refer to: https://daytonaero.com/wp-content/uploads/MIL-HDBK-516C-from-ASSIST.pdf

EASA Easy Access Rules for Acceptable Means of Compliance for Airworthiness of Products, Parts and Appliances (AMC-20), published by European Union Aviation Safety Agency (EASA), published March 2021; refer to: https://www.easa.europa.eu/en/document-library/easy-access-rules/online-publications/easy-access-rules-acceptable-means?page=22

FAA 2004 Reusable Software Components (AC 20-148), published by US Federal Aviation Administration, December 2004; refer to: https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_20-148.pdf

FAA 2013a Integrated Modular Avionics Development, Verification, Integration, and Approval Using RTCA/DO-297 and Technical Standard Order (TSO)-C153 (AC 20-170), published by US Federal Aviation Administration, November 2013; refer to: https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_20-170_with_chg_1.pdf

FAA 2013b Airborne Software Assurance (AC 20-115C), published by US Federal Aviation Administration, July 2013; refer to: https://www.faa.gov/documentlibrary/media/advisory_circular/ac_20-115c.pdf

FAA 2017a Airborne Software Development Assurance Using EUROCAE ED-12 and RTCA DO-178 (AC 20-115D), published by US Federal Aviation Administration, July 2017; refer to: https://www.faa.gov/documentLibrary/media/Advisory_Circular/AC_20-115D.pdf

FAA 2017b Airworthiness Certification of Aircraft (8130.2J), published by US Federal Aviation Administration, July 2017; refer to: https://www.faa.gov/documentLibrary/media/Order/FAA_Order_8130.2J.pdf

FACE 2020 FACE™ Conformance Test Suite, Version 3.0.1, published by The Open Group, June 2020; refer to: https://www.opengroup.org/face/conformance-testsuites

FACE FACE™ Group Charters, published by The Open Group; refer to: https://www.opengroup.org/content/future-airborne-capability-environment-face/group-charters

G209 Reference Implementation Guide for FACE™ Technical Standard, Edition 3.0 (G209), published by The Open Group, May 2020; refer to: www.opengroup.org/library/g209

RTCA 2000 Design Assurance Guidance for Airborne Electronic Hardware (DO-254), published by RTCA, March 2000; refer to: https://my.rtca.org/productdetails?id=a1B36000001IcjTEAS

RTCA 2005 Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations (DO-297), published by RTCA, November 2005; refer to: https://my.rtca.org/productdetails?id=a1B36000001IchFEAS

RTCA 2011a Formal Methods Supplement to DO-178C and DO-278A (DO-333), published by RTCA, December 2011; refer to: https://my.rtca.org/productdetails?id=a1B36000001IcfeEAC

RTCA 2011b Model Based Development and Verification Supplement to DO-178C and DO-278A (DO-331), published by RTCA, December 2011; refer to: https://my.rtca.org/productdetails?id=a1B36000001IcfiEAC

RTCA 2011c Object Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A (DO-332), published by RTCA, December 2011; refer to: https://my.rtca.org/productdetails?id=a1B36000001IcfgEAC

RTCA 2011d Software Considerations in Airborne Systems and Equipment Certification (DO-178C), published by RTCA, December 2011; refer to: https://my.rtca.org/NC__Product?id=a1B36000001IcmqEAC

RTCA 2011e Software Tool Qualification Considerations (DO-220), published by RTCA, December 2011; refer to: https://my.rtca.org/productdetails?id=a1B36000001IcfkEAC

SAE 1996 Guidelines and Methods of Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, published by SAE Aerospace, December 1996; refer to: https://webstore.ansi.org/standards/sae/saearp47611996arp4761

SAE 2010 Guidelines for Development of Civil Aircraft and Systems, published by SAE Aerospace, December 2010; refer to: https://webstore.ansi.org/standards/sae/saearp4754a2010arp4754a

X1608 FACE™ Conformance Policy, Version 2.0 (X1608), published by The Open Group, September 2016; refer to: www.opengroup.org/library/x1608

X205 FACE™ Conformance Verification Matrix, Edition 3.1 (X205), published by The Open Group, September 2020; refer to: www.opengroup.org/library/x205

1 Introduction

1.1 Purpose and Scope

This document identifies artifacts and activities associated with pursuit of conformance to The Open Group Future Airborne Capability Environment (FACE) Technical Standard that can contribute towards evidence of airworthiness according to selected flight certification guidance. The current version of this document targets two standards: MIL-HDBK-516C and DO-178C. Other standards may be added in future revisions of this document.

This document focuses on suppliers that are developing FACE conformant software in parallel or prior to development of Verification Evidence for flight certification. Ideally, FACE Conformance and airworthiness evidence are both planned and pursued from the beginning of the program. This document can aid software suppliers that have already developed flight certification evidence and then choose to obtain FACE Conformance. In most cases, the existing flight certified software must be modified (tailored) to achieve FACE Conformance. These modifications will often trigger rework of some certification artifacts, and thus the work towards FACE Conformance and airworthiness evidence proceeds in parallel in this case. The audience for this document includes the software supplier, and any role that has a stake in achieving both FACE Conformance and flight certification for the software, such as a system integrator, FACE Verification Authority (VA), FACE Certification Authority (CA), airworthiness qualification authority, software acquirer, etc.

This document is the primary deliverable of the Enterprise Architecture Airworthiness Committee, whose charter is to address airworthiness-related considerations across the FACE Technical Standard. The purpose of the Committee is to ensure that FACE conformant software, associated artifacts, and the process to produce them:

- Contribute towards evidence of airworthiness where possible, explicitly citing common requirements, processing, and guidance
- Address both commonality and variance at a high level
- Do not prevent the ability to meet airworthiness requirements

A related group in the FACE Consortium is the Technical Working Group (TWG) Safety Subcommittee on Software Safety (TWG-Safety group), also known as TWG-Airworthiness. The TWG-Safety group is cognizant of the efforts of the Enterprise Architecture Airworthiness Committee and participated in an early review of this document. In contrast to the Enterprise Architecture Airworthiness mandate, the purpose of the FACE TWG-Safety group is "to ensure that FACE conformant software and artifacts don't preclude the ability to meet airworthiness requirements". [FACE 2020]

This document uses the term certification in the context of an approval by an airworthiness authority. This approval could come from a civil authority or a military authority depending on the context of the approval. Some military airworthiness authorities use the term qualification or release instead of certification. This document uses the terms qualification and certification interchangeably and recognizes that some readers may apply them differently.

The scope of this document covers the entire FACE Technical Standard and associated documents such as the FACE Reference Implementation Guide (RIG) and FACE Conformance Verification Matrix (CVM). Flight certification (or release) guidance is only covered to the extent where artifacts or activities from the FACE Conformance process are identified as contributing to airworthiness evidence. Tailoring of existing flight certification artifacts is not addressed in this document.

Unless stated otherwise, the following editions/versions of FACE documents are implied. Note that the listed documents are not necessarily aligned with each other. These are the released versions available at the time of writing this document.

Document	Version	Release Date
FACE Reference Implementation Guide (RIG)	3.0	May 2020
FACE Conformance Test Suite (CTS)	3.0.1	June 2020
FACE Conformance Verification Matrix (CVM)	3.1	September 2020
FACE Technical Standard	3.1	July 2020
FACE Conformance Policy	2.0	September 2016

1.1.1 Benefits and Value to the Department of Defense (DoD)

Programs of record under the Department of Defense (DoD) are increasingly requiring software components that are FACE conformant due to the benefits of modularity, portability, and ease of integration.[1] The benefit of this document to the DoD is that it reduces acquisition costs by providing guidance to software suppliers and other product stakeholders. This guidance can reduce the cost and effort of achieving FACE Conformance in addition to flight certification. Specifically, this guidance aims to cross-reference activities and artifacts for FACE Conformance that can contribute to airworthiness evidence required for flight certification.

1.1.2 Benefits and Value to Developers and Other Stakeholders

The life-cycle cost of developing software that undergoes evaluation for requiring flight certification is more costly than for non-safety-critical software development. Suppliers, integrators, and purchasers of flight software that is certified as FACE conformant will benefit from cost reductions in two ways. One benefit is that during initial development, the effort spent on FACE Conformance can be used as evidence of airworthiness. Ideally, these two goals are pursued simultaneously, though this document focuses on the first goal as it maps to the second. Another benefit is that software components certified as FACE conformant are meant to be reused, which means the flight certification evidence can be made reusable as well as providing recurring savings. Additionally, to support software product lines, these artifacts can be revised, and have reuse benefits when extended into other product lines, such as document templates.

[1] For example, Title 10 U.S.C. 2446a.(b), §805: Requirement for modular open system approach in major defense acquisition programs; definitions.

1.2 Document Status

This document is not official guidance from any airworthiness authority. It includes guidance from the FACE Consortium documenting how efforts towards FACE Conformance contribute to the evidence of airworthiness. Applicants for flight certification should consult with their airworthiness authority early in project development to have the best likelihood of acceptance.

1.3 Document Organization

This document is organized as follows:

- Chapter 2 provides a brief overview of airworthiness assurance and points out important differences between certification of FACE Conformance and flight certification or similar airworthiness processes
- Chapter 3 identifies some of the key standards related to assurance of airworthiness
- Chapter 4 and Chapter 5 map FACE artifacts to the evidence required for the targeted standards
- Chapter 6 is a conclusion
- Chapter 7 provides cross-reference matrixes

2 Background

This chapter first provides a short introduction on airworthiness in Section 2.1 to introduce airworthiness concepts referenced throughout this document, and secondly points out important distinctions between FACE Conformance and flight certification in Section 2.2.

2.1 Airworthiness Background

This section starts with an overview of airworthiness, then in subsections identifies some typical evidence (artifacts) generated to show airworthiness, ending with an overview of the software life-cycle.

An aircraft is not airworthy unless it is safe for flight. The definition of airworthy paraphrased from the Federal Aviation Administration (FAA) Order 8130.2 is that an aircraft configuration, including the engine, propeller, and articles installed, is consistent with the drawings, specifications, and other data, and that the aircraft is in a condition for safe operation relative to wear and deterioration.

Approval of airworthiness is provided by an airworthiness authority. For civilian flight in the United States, this authority is the FAA. Each branch of the US military has its own airworthiness authority. Assurance of airworthiness must consider every system of the aircraft, including software. The focus in this document is on flight software; i.e., software that is executing on the flying aircraft, as distinguished, for example, from software used to analyze the aircraft design, or software running on a ground station.

Flight software must be carefully designed and rigorously checked through a process of validation and verification, to provide high confidence that it works as expected and avoids unexpected behavior. Software qualification is a mitigation of the risks inherent in critical software functionality. It should come as no surprise that developing safety-critical software is costly. Development of software that is assured for use in safety-critical systems is estimated to cost ten times more[2] than the development of software that does not have safety requirements.

Several philosophies of assurance can be observed across the different standards for airworthiness. Some depend on a correct development process (one that conforms to what is required by the relevant standards); some depend on fulfillment of objectives. Some attempt to build confidence through testing; others look to formal proof of correctness by design. For example, the MIL-HDBK-516C standard states criteria in the form of verification objectives along with means of compliance. Another example is DO-178C from RTCA, Software Considerations in Airborne Systems and Equipment Certification, which claims to be objectives-based but in practice can dictate or constrain the processes used. Developing software for airborne systems under this standard requires that the software be developed according to a planned development process including requirement definition, design, code, integration, and then that the software has been thoroughly tested to meet its requirements.

[2] Refer to: https://betterembsw.blogspot.com/2018/.

The purpose of DO-178C is to ensure that flight software meets its functional and safety (airworthiness) requirements. This is accomplished in DO-178C through the definition of objectives for the software life-cycle processes along with their accompanying activities and required evidence that the objectives have been satisfied. The DO-178C standard defines five Software Levels, also known as Design Assurance Levels (DALs) in other standards, which are applied to each software component through the performance of an airworthiness assessment during the system life-cycle processes. These levels are:

- Level A: software failure could result in a catastrophic aircraft failure; e.g., flight control or display
- Level B: software failure could result in a hazardous aircraft failure; e.g., flight management
- Level C: software failure could result in a major aircraft failure; e.g., supplemental navigation
- Level D: software failure could result in a minor aircraft failure; e.g., cabin communication
- Level E: software failure has no effect on aircraft operational capability; e.g., aircraft maintenance

In real terms, catastrophic typically means loss of life and equipment, or significant environmental harm. The term hazardous often refers to a large impact to the pilot's ability to safely land the aircraft. The levels then work down in severity to the level of minimal or no impact to the pilot or crew.

The software activities named in DO-178C are consistent with the Software Engineering Institute (SEI) Capability Maturity Model Integration (CMMI) process and accepted software development processes of other industry standards. These activities include specific development objectives and criteria so that software can be evaluated and certified to meet safety requirements. Verification, with an emphasis on requirements-based testing, is at the heart of this process. However, it is worth noting that a high CMMI rating is not sufficient – without direct experience with developing software for flight, organizations can struggle to complete the audits of an airworthiness authority, such as the Stage of Involvement (SOI) audits performed by the FAA for civilian flight certification.

2.1.1 Common Artifacts

A certification artifact is data presented to an airworthiness authority as assurance evidence. The artifact may take the form of a document, database, spreadsheet, etc. Example types of artifacts that are commonly expected by most airworthiness standards include:

- Requirements data (including tracing data between levels of requirements)
- Design/architecture (e.g., narrative descriptions, diagrams, interface definitions)
- Implementation (e.g., source code, models)
- Test cases and procedures
- Test results
- Configuration records

- Quality records
- Review records (e.g., review of requirements, design, implementation, tests, documents)
- Memory/timing margin
- Code review (e.g., automated static analyses or manual inspections that verify specific program properties such as compliance with coding standards, absence of references to uninitialized variables)
- Data Coupling and Control Coupling (DC/CC) analyses
- Coverage analysis (demonstrating that tests exercised all the software, with a level of granularity commensurate with the required assurance/software level)
- The specification of well-defined processes for configuration management, quality assurance (problem reporting and correction), etc.

2.1.2 Software Life-Cycle

DO-178C divides software life-cycle activities into processes:

- Planning
- Development
- Verification
- Configuration Management
- Quality Assurance
- Certification

This document is focused on the development and verification phases, described in more detail below.

2.1.2.1 Software Development

In DO-178C, the four phases of software development are defined as requirements, design, coding, and integration. DO-178C §5.5 further defines traceability activities with the goal of ensuring complete implementation of system requirements and visibility into any undocumented functionality. The activities during each of the development life-cycle phases are as follows:

Requirements: High-level software requirements are developed from the systems requirements. These requirements include functional, performance, interface, and safety-related requirements.[3] Bi-directional traceability is required between high-level software requirements and system requirements.

Design: High-level requirements are refined to develop both software architecture and software low-level requirements. Bi-directional traceability is required between low and high-level software requirements.

[3] See DO-178C §11.9.

Coding: Source code is implemented based upon the software architecture and low-level requirements as determined during design, in accordance with the documented coding standards. Bi-directional traceability is required between source code and low-level software requirements.

Integration: Consists of both software integration and hardware/software integration to test that the software system is operating properly in the target hardware configuration.

2.1.2.2 Software Verification

The purpose of the software verification process is to detect and remove defects introduced during development. Verification occurs throughout the software life-cycle. Verification activities include reviewing requirements and design data, analyzing software implementation, and testing software at the unit level, integrating software modules together, and testing on the target hardware. Process verification is most often accomplished by review, while functional verification is typically accomplished by testing.

The purpose of software verification is to provide evidence that software satisfies requirements. Consequently, requirements-based testing is advocated by both DO-178C (§6.4.2) and MIL-HDBK-516C (§14.2). According to DO-178C, software testing consists of verifying low-level requirements, verifying correct operation of software in the target environment, and verifying the implementation of the software requirements and components within the architecture. The DO-178C standard introduces the idea of independence, requiring that certain critical objectives are verified by a team that is independent from the team that authored the item being verified.

Similarly, from MIL-HDBK-516C, verification methods include analysis, test, and inspection of documents. Inspection of software architecture design documentation ensures that the software architecture is defined. Analysis and test of the software architecture and design mechanization ensures software/system requirements are properly allocated, are met, and provide a safe implementation for the applications supported. Analysis ensures software design supports the overall software architecture and does not conflict with system integrity requirements. [DoD, p.472]

MIL-HDBK-516C recommends the use of DO-178C as a reference for software development and assurance. MIL-HDBK-516C §15.6 (Software Qualification and Installation) describes a test methodology to verify that the software package is completely and correctly tested and integrated into the system. §15.6.1 further describes 12 levels of testing associated with this approach, starting with unit testing, and ending with flight testing. It notes three types of testing: requirements coverage, failure condition testing, and regression testing. [DoD, p.479]

2.2 FACE Conformance Compared to Flight Certification

The term certification can have many meanings. With regards to FACE Conformance, a UoC is first verified by a FACE VA, then certified by a FACE CA as meeting the FACE Technical Standard (resulting in a FACE Conformance Certificate), and finally can be listed in the FACE Registry. With regards to airworthiness, flight certification (also called qualification, depending on the airworthiness authority) means that the software was part of an aircraft that was approved by an airworthiness authority to meet airworthiness requirements with a sufficient level of confidence. While these two types of certification are different, this document identifies activities and artifacts required to achieve a FACE Conformance Certificate that can be used in whole or part to support airworthiness certification.

2.2.1 Relationship Between FACE Technical Standard and Airworthiness Guidance

The FACE Technical Standard does not provide guidance regarding airworthiness and flight certification, noting that there are safety engineering practices that are important but are outside the domain of the FACE Technical Standard [C207, p.184]. However, each edition is reviewed by the TWG-Safety group to ensure the standard does not preclude any planned flight certification activities. By contrast, the purpose of this document is to identify activities and artifacts required for conformance to the FACE Technical Standard that contribute to evidence of airworthiness. Figure 1 illustrates these conceptual relationships, showing that activities and artifacts driven by the FACE Technical Standard requirements overlap with the evidence needed for software aspects of military airworthiness criteria described in MIL-HDBK-516C and with the civilian military airworthiness criteria described in DO-178C.

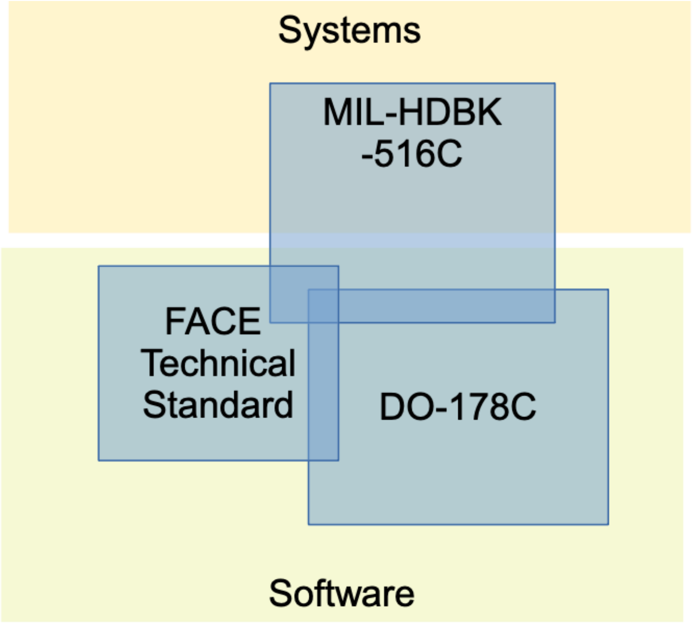

Figure 1: Conceptual Relationship Between Standards

2.2.2 FACE Conformance Applies to a UoC

Certification of FACE Conformance is conferred to a UoC, which may be a subset of a FACE architectural segment or form an entire segment, as defined in the FACE CVM. The FACE Technical Standard describes a layered architecture, where the layers are called segments and notes that the “FACE Reference Architecture is comprised of logical segments where variance occurs. The structure created by connecting these segments together is the foundation of the FACE Reference Architecture.” [C207, p.11] The FACE Technical Standard defines three profiles for the Operating System Segment (OSS) that tailor the included functionality: security (most restricted), safety, and general purpose (least restricted). FACE Conformance is granted against a particular profile.

To be certified as conformant, a UoC must meet its applicable requirements of the FACE Technical Standard as reflected in the CVM. FACE requirements are primarily focused on interfaces, not internal functionality, behavior, or assurance properties such as safety or security. FACE Conformance activities produce artifacts (data supporting demonstration of conformance), which are examined by a FACE VA. By contrast, the DO-178C standard goes beyond this type of "black box" verification at the higher software levels (i.e., more rigorous design assurance), requiring "white box" verification, such as structural coverage of code.

Software suppliers are responsible for providing the VA with their Software Verification Package that includes the Statement of Conformance. The Statement of Conformance includes the conditionals that apply to the UoC. It provides the justification for meeting each requirement or explains why a requirement is not applicable based on options chosen. The characterization found on the Statement of Conformance reflects design intent and system boundaries. For requirements which the software supplier intends to meet, the justification includes specific references into UoC design artifacts, requirements, test reports, certification artifacts, etc.

One of the goals and benefits of conformance is reuse. However, reuse will generally be at the granularity of the UoC. There is no defined concept of FACE Conformance for an entire system. For example, the system might be composed of components that are UoCs certified as FACE conformant connected to other components that may or may not be conformant. The UoCs certified as FACE conformant may have been reused from earlier systems.

2.2.3 Airworthiness Certification Applies to Entire Aircraft

In contrast to FACE Conformance, flight certification can only be performed for an entire aircraft (e.g., under US Code of Federal Regulations Title 14 Parts 21, 23, 25, and 27) and not for individual software or hardware components of the aircraft. This certification is done to a particular assurance level. Certification involves verifying properties such as safety, security, correct functionality, etc. Because the entire system is the only granularity that can be certified, products – such as a Real-Time Operating System (RTOS), middleware, or a file system – as components of an aircraft can only be "flight certifiable", meaning they have associated artifacts that will contribute to the flight certification effort for an aircraft, as illustrated in Figure 2.

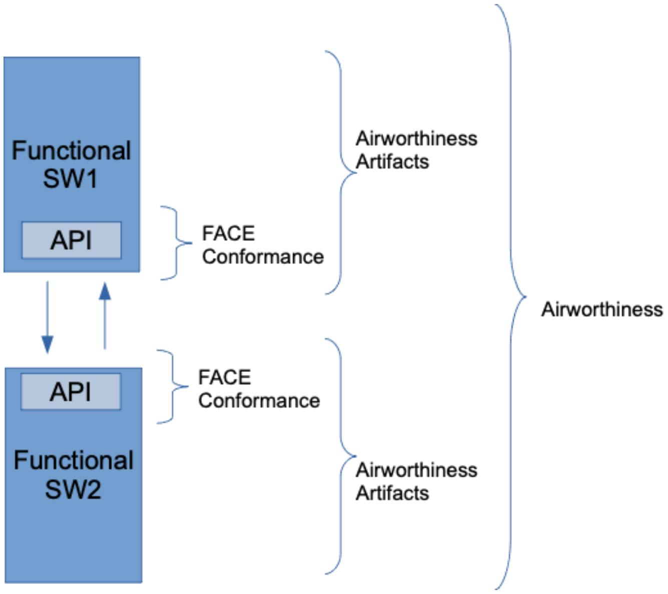

Figure 2: FACE Conformance Related to Airworthiness

2.2.4 Flight Certification of Reusable Components

Airworthiness artifacts may apply to specific components, as long as it is clear that these artifacts ultimately must be generated and aggregated for the entire aircraft in order to seek flight certification. This approach forms the basis for reuse of the software itself, and for reuse of the attendant artifacts. The FAA Advisory Circular (AC) 20-148 on Reusable Software Components provides detailed instructions guiding the development of software intended for reuse and guiding the integrator intending to reuse the software component. This includes an outline of the data to support reuse and identification of common issues encountered when attempting to reuse a component.

With UoC reuse being one of the primary benefits of the FACE architecture, it follows that FACE UoCs can be treated as reusable components for flight certification. A component can have a certification data package that is independently reviewed, though this does not make it airworthy until that package is examined in the context of the overall system to ensure its suitability. Certification artifacts related to a particular component may be reused on a new project, but a change impact analysis must be performed on this component which is now considered previously developed software.

Figure 3 illustrates how the FACE artifacts can contribute to the verification of these development life-cycle processes.

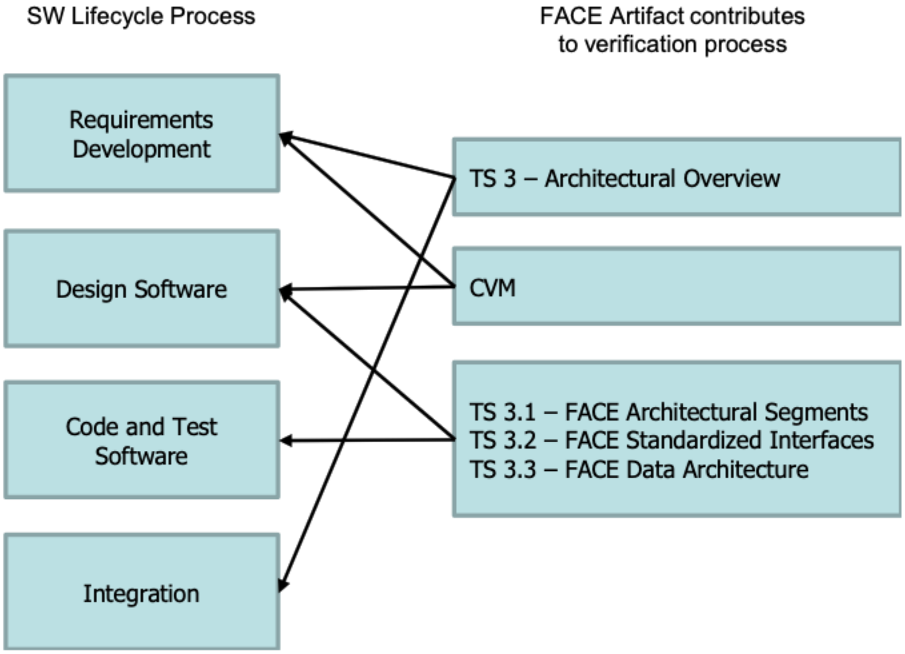
SW Lifecycle Process
FACE Artifact contributes to verification process
Requirements Development
Design Software
Code and Test Software
Integration
TS 3 – Architectural Overview
CVM
TS 3.1 – FACE Architectural Segments
TS 3.2 – FACE Standardized Interfaces
TS 3.3 – FACE Data Architecture

Figure 3: FACE Artifact Contributions to Software Life-Cycle

3 Airworthiness Guidance

A variety of standards, criteria, and guidance are used by airworthiness authorities in deciding whether to certify an aircraft for flight. This chapter describes some of the most significant documents regarding flight certification. The first section focuses on international standards and the second focuses on US-specific standards.

3.1 International Standards

DO-178C/ED-12C Software Considerations in Airborne Systems and Equipment Certification

This standard was jointly published in 2011 in the US by Radio Technical Commission for Aeronautics (RTCA) as DO-178C and by the European Organisation for Civil Aviation Equipment (EUROCAE) as ED-12C. It is referenced in the US by the FAA in AC 20-115C and in Europe by the European Union Aviation Safety Agency (EASA) as "an acceptable means, but not the only means, for showing compliance with the applicable airworthiness regulations for the software aspects of airborne systems." [FAA 2013b, EASA] The DO-178C standard provides overall guidance on processes and artifacts for development of software systems. Although the document is considered guidance, it is often viewed as the definitive step-by-step description for which activities must be accomplished for airworthiness. In the US, a Designated Engineering Representative (DER) appointed by the FAA may be used to provide the overall assessment of aircraft systems for flight certification, under the supervision of the FAA. The document describes software levels from A (the most rigorous) through E (the least rigorous) and the required artifacts and separation of responsibilities recommended for each of those levels. Four supplements were developed to accompany revision C of the DO-178 standard: DO-330, 331, 332, and 333. These supplements are intended to be used in conjunction with DO-178C and not as standalone guidance.

DO-330/ED-215 Software Tool Qualification Considerations

This standard is applicable to Software Tool Qualification for tools used to automatically generate software (such as model-based development software) as well as tooling used for automated verification. Tools must be qualified (note, this does not mean certified in the context of this document) whenever they are used to take credit for an objective in DO-178C that would normally be achieved through a manual means. For example, a test harness may be used to determine whether a specific test passes or fails. Normally, a manual review of correctness of each test result would be performed, but in some cases the test harness can be qualified to do so. Within the FACE ecosystem, the use of automated code generation and/or automated verification can greatly reduce the qualification efforts of porting software to new systems. The qualification of the tools used to develop and verify portable components should be considered for any software that may later be used in support of DAL-A to DAL-D functions.

DO-331/ED-218 Model-Based Development and Verification Supplement to DO-178C and DO-278A

This standard addresses the use of model-based development techniques in developing and verifying software under the DO-178C standard. The FACE Technical Standard defines a number of Data Models in §3.3 "FACE Data Architecture" and §4.9 "Data Architecture".

DO-332/ED-217 Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A

This standard addresses applicability of object-oriented software techniques, and related techniques such as generic template programming, and their impacts on certification.

DO-333/ED-216 Formal Methods Supplement to DO-178C and DO-278A

This standard describes the use of formal (mathematical) arguments as a verification technique, and how it can be used in conjunction with and/or as a replacement for certain testing activities.

DO-297/ED-124 Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations

Published in 2005, this standard describes the concept of IMA as a path for the development of artifacts for a computing platform separately from the capabilities that can be hosted on that platform. Based on isolation provided by a partitioning platform, IMA enables incremental acceptance, providing modularity and easing integration.

DO-254/ED-80 Design Assurance Guidance for Airborne Electronic Hardware

Published in 2000, this standard describes a process for development of complex electronic hardware (e.g., programmable logic) similar to that described in DO-178C for software.

ARP4754A Guidelines for Development of Civil Aircraft and Systems

Published by the Society of Automotive Engineers (SAE) in 2010, this standard is referenced in the US by the FAA in AC 20-174 and in Europe by EASA in ED-79. It describes a process for development of avionics systems with a focus on safety. It discusses "validation of requirements and verification of the design implementation for certification and product assurance. It provides practices for showing compliance with the regulations and serves to assist a company in developing and meeting its own internal standards by considering the guidelines".[4]

ARP4761 Guidelines and Methods of Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment

Published by SAE in 1996, this standard provides recommended practices for modeling and evaluating the safety of an aircraft, including functional hazard assessment, fault tree analysis, Failure Mode and Effects Analysis (FMEA), etc. The document is primarily associated with showing compliance with FAR/JAR 25.1309. The methods identify a systematic means, but not the only means, to show compliance. A sub-set of this material may be applicable to non-

[4] https://www.sae.org/standards/content/arp4754a/

25.1309 equipment. The concept of Aircraft Level Safety Assessment is introduced and the tools to accomplish this task are outlined. The overall aircraft operating environment is considered.[5]

3.2 US Standards

3.2.1 US Military

MIL-HDBK-516C Airworthiness Certification Criteria

This standard was published by the US DoD in 2014. It is referenced by the airworthiness authorities of each US military service. It provides overall guidelines regarding processes and artifacts for development of systems: mechanical, electrical, computer hardware, etc. §14 covers system safety (including software safety). §15 focuses on computer systems and software. The standard describes a Computer System Integrity Level (CSIL), which dictates a corresponding development process with an appropriate amount of rigor but does not define specific levels.

3.2.2 US Civilian

AC 20-115D Airborne Software Development Assurance Using EUROCAE ED-12 and RTCA DO-178

This AC issued by the FAA in 2017 describes DO-178C as "an acceptable means, but not the only means, for showing compliance with the applicable airworthiness regulations for the software aspects of airborne systems and equipment in type certification or [Technical Standard Order] (TSO) authorization". [FAA 2017a, p.1]

This AC aids in developing new software under the older DO-178B or the current DO-178C, and assistance with modifying or reusing software using any of the earlier revisions of DO-178. Guidance includes:

- Objectives for life-cycle processes
- Activities that provide means for satisfying development objectives
- Descriptions of evidence that indicate the objectives have been satisfied

AC 20-170 Integrated Modular Avionics Development, Verification, Integration, and Approval Using RTCA/DO-297 and TSO-C153

This AC issued by the FAA in 2013 describes DO-297 as "an acceptable means of compliance for aircraft and engines that utilize [IMA] systems". [FAA 2013a]

This AC provides guidance to developers of IMA systems applications and components by providing a method to show compliance to obtain approval of IMA components. As a reference, along with RTCA/DO-297, this AC provides an acceptable means of compliance for the development, integration, verification, and installation approval of IMA systems.

[5] https://www.sae.org/standards/content/arp4761/

AC 20-148 Reusable Software Components (RSC) Guidance

This AC issued by the FAA in 2004 describes “one acceptable means of compliance, but not the only means, for … RSC developers, integrators, and applicants to gain acceptance of software component and credit for reuse of a software component”. [FAA 2004, p.1]

This AC provides developers with a method to show that their RSC is compliant with RTCA/DO-178B. In conjunction with AC 20-115D, this AC provides methods/procedures that can be followed to ensure that the reuse of software components and software life-cycle data can aid in using RSC.

4 Mapping FACE Artifacts to MIL-HDBK-516C Flight Certification Evidence

This chapter identifies artifacts generated during FACE Conformance activities and correlates them to sections of MIL-HDBK-516C, in support of the specified airworthiness requirements. This chapter is organized into subsections for each FACE Consortium document with subsections for each specific mapping to requirements in MIL-HDBK-516C, ending with a final section covering a list of certification documents from the MIL-HDBK-516C.

4.1 FACE RIG

The RIG provides best practices and examples for implementing UoCs and more detailed information on certain services, addresses safety and security concerns during the planning and design phases of UoC development, provides programming language mapping rules, and explains the FACE Data Architecture.

RIG Volume 1, §6 on “Safety Guidance” describes overall activities and processes that should be used for development of flight-worthy software. This is a high-level overview, without specific mappings of FACE artifacts to specific requirements of an airworthiness standard, thus §6 of the RIG should be considered a companion to this document.

4.2 FACE CTS

The CTS is a framework for testing candidate FACE UoCs and Data Models to ensure they meet the interface requirements of the FACE Technical Standard. Data Models must be provided for Portable Components Segment (PCS), Platform-Specific Services Segment (PSSS), and Transport Services Segment (TSS) UoCs (see the FACE Technical Standard §3.3.1). All requirements tested by the CTS are defined in the CVM, which is maintained by the FACE Consortium. However, the CTS only tests a portion of the total requirements listed in the CVM. All types of UoCs may be tested with the CTS, with specific sections devoted to each type of UoC as well as testing a Data Model.

4.2.1 Entire CTS Contributes to MIL-HDBK-516C §4.2.1

FACE Artifact	**Contributes to MIL-HDBK-516C: §4.2.1 Tool and Database Processes**	**Analysis**
FACE CTS	Criterion: Verify that all tools, methods, and databases used in the requirements management, design, risk control, and assessments of safety are applied appropriately and exhibit accuracy commensurate with their application.	The CTS can be considered a verification tool to confirm FACE Conformance when the FACE Application Programming Interface (API) is part of the system requirements. The CTS is not qualified, so the output must be manually reviewed. MIL-HDBK-516C §4.2.1 requires
	Standard: Processes are in place to	

FACE Artifact	Contributes to MIL-HDBK-516C: §4.2.1 Tool and Database Processes	Analysis
	ensure that all analysis, modeling, and simulation tools and databases are of appropriate accuracy and fidelity, are validated for the intended applications, and are configuration controlled. Requirements definition/traceability, design and performance analysis tools, prediction methods, models, and simulations are applied appropriately, and exhibit accuracy commensurate with their applications.	configuration control. While the CTS itself is under configuration management, the UoC tested may or may not be. The certification of FACE Conformance must be tied to configuration management of the UoC to provide baseline control.

4.2.2 Entire CTS Contributes to MIL-HDBK-516C §15.2.5

FACE Artifact	Contributes to MIL-HDBK-516C: §15.2.5 Simulations, Models, and Tools	Analysis
FACE CTS	Criterion: Verify that simulators, models, and tools used in the development, integration, and testing of software and hardware supporting Safety-Critical Functions (SCFs) have been appropriately qualified and validated.	The CTS is a tool provided by the FACE consortium to verify FACE Conformance. CTS development and life-cycle management are summarized in §6 of the FACE Conformance Policy. The CTS is a tool used to verify FACE Conformance, but it does not otherwise verify correctness of the UoC. The CTS does verify correct usage of the UoC with the FACE APIs; however, the results would not be suitable to support flight certification without human review of the output. Thus, suppliers should trace their designs to the FACE Technical Standard elements and show airworthiness verification through the testing of those interfaces in a traditional way. The CTS might be useful as supplemental evidence that the interface requirements are addressed properly.
	Standard: All simulators, models, and tools used for design, development, integration, and test have been identified and a qualification and validation approach has been defined. Each simulator, model, and tool is identified as off-the-shelf (commercial or government), modified, or developed for the application, and has been adequately qualified (for the Navy, accredited) and validated for its intended use to the appropriate qualification level. Verification tools do not mask safety issues within products they are used to verify.	

4.3 FACE Conformance Policy

The FACE Conformance Policy defines the processes and policies that govern the FACE Conformance program, the goal of which is: “to provide a trusted, accessible, and fair process for achieving FACE Conformance Certification”. [X1608, p.5]

4.3.1 Entire Conformance Policy Contributes to MIL-HDBK-516C §4.2.1

FACE Artifact	Contributes to MIL-HDBK-516C: § 4.2.1 Tool and Database Processes	Analysis
FACE Conformance Policy	Criterion: Verify that all tools, methods, and databases used in the requirements management, design, risk control, and assessments of safety are applied appropriately and exhibit accuracy commensurate with their application.	As directed in §2.3.1 of the FACE Conformance Policy, CTS testing for FACE verification must be performed and witnessed by (or at the direction of) a FACE VA.
	Standard: Processes are in place to ensure that all analysis, modeling, and simulation tools and databases are of appropriate accuracy and fidelity, are validated for the intended applications, and are configuration controlled. Requirements definition/traceability, design and performance analysis tools, prediction methods, models, and simulations are applied appropriately, and exhibit accuracy commensurate with their applications.	

4.3.2 Entire Conformance Policy Contributes to MIL-HDBK-516C §15.2.5

FACE Artifact	Contributes to MIL-HDBK-516C: §15.2.5 Simulations, Models, and Tools	Analysis
FACE Conformance Policy	Criterion: Verify that simulators, models, and tools used in the development, integration, and testing of software and hardware supporting SCFs have been appropriately qualified and validated.	The CTS is a tool provided by the FACE Consortium to verify FACE Conformance. CTS development and life-cycle management are summarized in §6 of the FACE Conformance Policy. The CTS is a tool used to verify FACE Conformance, but it does not otherwise verify correctness of the UoC. The CTS does verify correct usage of the UoC with the FACE APIs; however, the results would not be suitable to support flight certification without human review of the output.
	Standard: All simulators, models, and tools used for design, development, integration, and test have been identified and a qualification and validation approach has been defined. Each simulator, model, and tool is identified as off-the-shelf (commercial or government), modified, or developed for the application, and has been adequately qualified (for Navy, accredited) and validated for its intended use to the appropriate qualification level. Verification tools do not mask safety issues within products they are used to verify.	

4.4 FACE CVM

The FACE CVM provides the Product Standard that clarifies the requirements from the FACE Technical Standard that a product must meet in order to be certified as FACE conformant. It specifies the technique(s) to be used to verify each of these requirements. The CVM categorizes each element of the FACE Technical Standard, identifying which FACE segments are applicable, which verification methods should be used, etc.

Although the CVM indicates how requirements should be verified (by test, by inspection), it is not a verification tool. The CVM is an analysis or accounting tool that acts as a checklist. Thus, in the sections below that map some or all portions of the CVM to airworthiness standards, the listing of "FACE CVM" is meant to indicate Verification Evidence showing conformance to the CVM.

4.4.1 Entire CVM and FACE Technical Standard Contributes to MIL-HDBK-516C §15.5.1

FACE Artifacts	**Contributes to MIL-HDBK-516C: §15.5.1 Software Architecture**	**Analysis**
FACE CVM FACE Technical Standard	Criterion: Verify that the software architecture and design are defined, properly implement the system/software requirements, and are safe.	The FACE architecture is well defined and documented via the Technical Standard, the Shared Data Model (SDM), and CVM. FACE software provides functionality through the FACE architecture. UoC software is developed to the requirements for a single segment of the FACE architecture and provides no external interfaces beyond those defined by the chosen segment(s). The FACE Conformance process assures that the software adheres to the requirements, which are defined in the Technical Standard and outlined in the CVM. The CVM identifies the FACE architecture and interfaces, which helps demonstrate that the requirements regarding the architecture and interface design are defined and implemented.
	Standard: Computer system software architecture is defined. System-level requirements are allocated to the sub-system and software requirements. All software and associated safety requirements are defined and allocated to software components. Software architecture is consistent with the software functional requirements; especially functions that ensure system integrity (e.g., partition schemes). The software architecture and design mechanizations implement proven techniques.	

FACE Artifacts	Contributes to MIL-HDBK-516C: §15.5.1 Software Architecture	Analysis
	Method of Compliance: Verification methods include analysis, test, and inspection of documents. Inspection of software architecture design documentation ensures that the software architecture is defined. The analysis and testing of the software architecture and design mechanization ensures software/system requirements are properly allocated, are met, and provide a safe implementation for the applications supported. Analysis ensures software design supports overall software architecture and does not conflict with system integrity requirements.	The SDM provides a concise measurement description for the base modeling elements on which all models must build. The CVM does not check the safety properties of the architecture. However, use of the OSS safety profile can help limit the scope of effort and the CVM does verify that the UoC provides the proper APIs to conform to the profile.

4.4.2 CVM Items F-27, F-29, F-33, and F-34 Contribute to MIL-HDBK-516C §15.1.10 (Partitioning)

Table 1: FACE CVM Partitioning Verification Items, Including F-27, F-29, F-31, F-33, and F-34

Row ID	FACE Segment	Technical Standard Requirements	Verification Method	Conformance Artifacts
F-27	OSS	A General-Purpose Profile OSS UoC shall provide space partitioning.	Inspection	UoC Designs
F-29	OSS	A Safety Profile OSS UoC shall provide space partitioning.	Inspection	UoC Designs
F-33	OSS	A Security Profile OSS UoC shall provide space partitioning.	Inspection	UoC Designs
F-34	OSS	A Security Profile OSS UoC shall provide time partitioning.	Inspection	UoC Designs

FACE Artifacts	Contributes to MIL-HDBK-516C: §15.1.10 Physical and Functional Separation	Analysis
Refer to Table 1	Criterion: Verify that physical and functional separation between Software Supporting Elements (SSEs) and non-SSEs are accounted for in the System Processing Architecture (SPA).	The CVM specifies that time and space partitioning must be verified by inspection. Inspection implies that the OSS supplier provides evidence of time and space partitioning, which would ensure partitioning breaches are prevented and ensure partitioning integrity. Much of the compliance evidence necessary to support certification would be provided by the OSS supplier.
	Standard: Partitioning schemes and operating systems utilized to separate SSEs from non-SSEs are developed and tested at the highest CSIL supported. Partitioning schemes ensure partitioning breaches are prevented and that derived requirements are defined to ensure partitioning integrity is maintained.	

4.4.3 CVM Inspection of Fully Tested Requirements Contribute to MIL-HDBK-516C §4.1.1

FACE Artifact	Contributes to MIL-HDBK-516C: §4.1.1 Requirements Allocation	Analysis
FACE CVM items marked with Verification Method of “Inspection” and Conformance Artifacts listed as “Fully Tested Requirement”	Criterion: Verify that the design criteria, including requirements and ground rules, adequately address airworthiness and safety for mission usage, full permissible flight envelope, duty cycle, interfaces, induced and natural environment, inspection capability, and maintenance philosophy.	Where the FACE CVM dictates verification by inspection of fully tested requirements (there are 60 such entries in the CVM), this implies traceability from the FACE Technical Standard requirements to verification tests, thus supporting: “traceability is documented among requirements, design criteria, design, and verification”. The CVM does not directly address traceability to the full set of life-cycle artifacts, which may include designs, requirements, and other documentation. The CVM only verifies requirements in the FACE Technical Standard. These requirements may be system requirements, if for example the system
	Standard: Allocated high-level airworthiness and safety requirements through the design hierarchy are defined. Allocated design criteria for all system elements and components result in required levels of airworthiness and safety throughout the defined operational flight envelope, environment, usage, and life.	

FACE Artifact	Contributes to MIL-HDBK-516C: §4.1.1 Requirements Allocation	Analysis
	Method of Compliance: Inspection of process documentation verifies allocation of airworthiness and safety requirements and design criteria. Traceability is documented among requirements, design criteria, design, and verification. Consistency between design criteria and airworthiness and safety requirements is confirmed by the inspection of documentation.	must integrate a FACE UoC, but it is possible that they may not trace to any system requirements if the FACE architecture is a low-level design choice. In the cases where the FACE requirement can only be proved through functional testing, the software supplier will perform the functional tests and supply the test procedures and results to the VA. Functional testing is considered part of the software development life-cycle and is not conducted during the FACE Conformance Program.

4.5 FACE Technical Standard

The Technical Standard is the key product of the FACE Consortium. It describes the FACE Reference Architecture, interfaces, and requirements that must be met. "The FACE approach is to develop a Technical Standard for a software Common Operating Environment (COE) designed to promote portability and create software product lines across the military aviation domain." [C207, p.xv] The standard was developed using industry standards for distributed communications, operating systems, and other applicable domains. The FACE Reference Architecture is defined in §3 of the FACE Technical Standard, with sections on architectural segments, standardized interfaces, the FACE Data Architecture, programming language run-times, component frameworks, OSS profiles, and an explanation of UoCs.

4.5.1 Technical Standard §3 Contributes to MIL-HDBK-516C §15.1.2

FACE Artifact	Contributes to MIL-HDBK-516C: §15.1.2 SPA Requirements	Analysis
FACE Technical Standard §3 "Architectural Overview"	Criterion: Verify that the SPA safety requirements are fully defined and documented.	Although the FACE Reference Architecture does not directly contribute to safety and performance requirements, it does define functional requirements related to the API, defines a Data Architecture related to data flow, and an API related to interfacing. When performing the SPA analysis as required by MIL-HDBK-516C §15.1.2, the FACE Reference Architecture can be referenced to provide evidence for the attributes of data flow and interfacing elements. The details of how FACE UoCs meet these requirements would be
	Standard: Safety and performance requirements are allocated to the architecture. The SPA is defined. An analysis of the SPA is performed to address attributes such as functional requirements, processing demands, timing criticalities, data flow, interfacing elements, and fault tolerance. Federated and integrated elements of the SPA are identified.	

FACE Artifact	Contributes to MIL-HDBK-516C: §15.1.2 SPA Requirements	Analysis
	Method of Compliance: Verification methods include analysis and inspection. Verify that the analysis of the SPA is complete and is documented. Analysis determines if the SPA supports program safety and performance requirements. Ensure all technical and safety SPA risks are appropriately mitigated/captured.	documented in requirements and design material. Software suppliers may want to follow guidance from FAA AC 20-148 for advice on reusability.

4.5.2 Technical Standard §3 Contributes to MIL-HDBK-516C §15.2.3

FACE Artifact	Contributes to MIL-HDBK-516C: §15.2.3 Integration Methodology	Analysis
FACE Technical Standard §3 "Architectural Overview"	Criterion: Verify that the integration methodology used for the SPA SSEs is defined, documented, and provides complete verification coverage of SCFs at all levels, for each flight configuration release.	The FACE Reference Architecture is designed to support reuse of FACE UoCs. The modular design of the architectural segments will support efficient integration and effective testing, particularly if the supplier is providing a test suite with the UoC. Each UoC that has achieved conformance will have demonstrated adherence to the FACE architecture, thereby supporting the integration methodology described in MIL-HDBK-516C. Each of the segments of the FACE Reference Architecture contributes to the advantages of a modular design that is straightforward to integrate. Like any reusable component, a FACE UoC will have a pedigree; e.g., DO-178C certification artifacts. This pedigree can save cost/schedule in the certification of the integrated system. The ease of integration is in part due to the UoC being developed to an open standard that includes descriptive interface documentation and independently enforced consistent use of that interface. System integrators will still be responsible for combining UoCs, where the overall system must meet the safety criteria defined for the aircraft. Reuse of a UoC will still require addressing specific safety criteria.
	Standard: The entire SPA is developed, integrated, and verified using a defined, documented, and proven process which includes complete test coverage (requirements, functions, failure conditions, and mission validation) at all levels. Verification methodology includes end-to-end testing of SCF threads. SSEs, along with non-SSEs, are developed and tested individually; then integrated to form an SPA or multiple SPAs. Multiple SPAs may be integrated to form the system. Software residing on an SPA or multiple SPAs forms a build. Each build, for planned flight release, has all the safety-critical functionality necessary to ensure safe flight. The integration methodology identifies conditions (adequacy of partitioning, decision points, areas of testing) for which a subset of testing is permissible *in lieu* of complete testing. Regression process addresses adequacy of partitioning, the areas of change, other areas affected by the change(s), and core testing required independent of the change(s) (comprehensive verification of SCFs that are supported by modified SSEs).	

FACE Artifact	Contributes to MIL-HDBK-516C: §15.2.3 Integration Methodology	Analysis
	Method of Compliance: Verification method includes inspection of integration planning and test documentation. Inspection of integration plans and process documentation verifies that the integration methodology has been implemented and provides complete verification coverage. Verify that the SSEs of the SPA are clearly identified such that any dependencies that safety-critical systems (e.g., vehicle management system) have on other systems are addressed. Ensure integration plans show a reasonable build-up approach. Verify that the test documents reflect all levels of testing throughout all levels of the architecture/system. Verify the plans and results of end-to-end SCF testing.	

4.5.3 Technical Standard §3.1 to §3.3 Contribute to MIL-HDBK-516C §15.1.9

FACE Artifacts	**Contributes to MIL-HDBK-516C: §15.1.9 Data and Control Flow**	**Analysis**
FACE Technical Standard §3.1 “FACE Architectural Segments”, §3.2 “FACE Standardized Interfaces”, and §3.3 “FACE Data Architecture”	Criterion: Verify that interfaces (control and data flow) supporting SPA SSEs are clearly defined and documented.	Verification by a FACE VA is required to ensure that a UoC, representing an SSE, has been tested to meet the FACE Interface and Data Model requirements found in the Technical Standard. To achieve FACE Verification, the data exchanged between UoCs and the method for that transport is defined and documented. In addition, each software function is assigned to a segment, which requires separation from platform-specific data sources. This improves the reliability of the design to meet the data flow requirements. The requirements of the FACE Technical Standard only partially contribute to MIL-HDBK-516C §15.1.9 since the FACE Technical Standard does not include any of the actual data and control flow requirements. The lower-level details of the actual data and control flow must be supplied by the FACE UoC vendor to complete the set of artifacts. On the other hand, the segregation of the POSIX™ and ARINC 653 APIs into profiles, and the specification of the safety and security profiles (to exclude functionality that could be problematic in a safety-critical system) contribute to meeting the MIL-HBK-516C requirements. The FACE Technical Standard identifies and documents these interfaces. The FACE Conformance process further ensures that the UoC does not use interfaces that are prohibited in the targeted profile. However, interfaces to higher-level safety-related functionality are outside the scope of the FACE Technical Standard; e.g., services that relate to ensuring the continued safe operation of the aircraft.
	Standard: All SSE interfaces which handle control and data associated with safety-critical functions have been clearly defined and documented. Interface requirements for safety are defined and accounted for throughout the development process (e.g., drive-specific test processes). Safety-critical interfaces are designed to ensure that data/calculation/system-timing dependencies do not impede system performance in any operational mode or degrade architectural safety coverage.	

4.5.4 Technical Standard §3.1 to §3.3 Contribute to MIL-HDBK-516C §15.5.1

FACE Artifacts	Contributes to MIL-HDBK-516C: §15.5.1 Software Architecture	Analysis
FACE Technical Standard §3.1 “FACE Architectural Segments”, §3.2 “FACE Standardized Interfaces”, and §3.3 “FACE Data Architecture”	Criterion: Verify that the software architecture and design are defined, properly implement the system/software requirements, and are safe.	The objective “verify that the software architecture and design are defined” is partly met, since the software architecture and interfaces design are defined in the FACE Technical Standard. Detailed design aspects will likely need further elaboration and documentation to support certification. The objective “properly implement the system/software requirements” is partly fulfilled by §1.2.2 “Technical Approach” of the FACE Technical Standard, as it defines a set of principles that can guide development of software requirements. The FACE Reference Architecture (§3) provides an organizing architectural structure, including interface requirements for portability and interoperability, which could be considered white box (rather than black box) internal design requirements. The FACE Data Models may play a role here as well. Note that FACE Conformance checks for correct interfaces and correct usage of the API but does not test internal behavior of UoCs. Thus, the supplier of a UoC will need to provide verification artifacts that validate behavior requirements. The objective “are safe” is only partly fulfilled. Fully demonstrating that the architecture and design for a UoC are safe is beyond the scope of what the FACE approach intended to achieve. However, at the same time, the FACE architecture does not compromise safety. This is because the architecture is a design choice that is independent of function. This design choice does not preclude traditional software reliability mechanisms like redundancy, data integrity checks, and interlocks. The FACE Technical Standard defines profiles and capability sets with restricted functionality. These profiles do not impose specific safety requirements but rather are in the interest of promoting analyzability and
	Standard: Computer system software architecture is defined. System-level requirements are allocated to the sub-system and software requirements. All software and associated safety requirements are defined and allocated to software components. Software architecture is consistent with the software functional requirements; especially functions that ensure system integrity (e.g., partition schemes). The software architecture and design mechanizations implement proven techniques and are safe. The software architecture is compatible with the target hardware architecture.	

FACE Artifacts	Contributes to MIL-HDBK-516C: §15.5.1 Software Architecture	Analysis
		avoiding potentially unsafe constructs. Nonetheless it can be claimed that the FACE Reference Architecture can be implemented safely. The evidence of this must be generated and is likely beyond the artifacts generated for FACE Conformance.

4.5.5 Technical Standard §3.1.1 Contributes to MIL-HDBK-516C §15.1.10 (Partitioning)

See CVM Items F-27 to F-34 for details of this contribution.

4.5.6 Technical Standard §4.1.3 Contributes to MIL-HDBK-516C §15.5.4

FACE Artifact	Contributes to MIL-HDBK-516C: §15.5.4 Dynamic Operation	Analysis
FACE Technical Standard §4.1.3 "OSS Health Monitoring and Fault Management"	Criterion: Verify that the following are designed to safely operate under all dynamic conditions anticipated: mode inputs, operational flight modes, failure monitoring and detection techniques, failure management functions, redundancy management, voting schemes, self-checks, built-in-tests, safety interlock mechanizations, SCF interfaces, health status interfaces, reconfiguration capabilities, and switchover of command-and-control data links. Standard: The Safety Supporting Software Element (SSSE) designs account for all dynamic conditions anticipated for the system. Ensure the transient effects of mode switching and condition changes are acceptable and do not introduce unacceptable hazards. Switchover of command-and-control data links does not result in loss of control. Flight test features and software hooks for laboratory testing cannot be activated in any unintended flight mode. Design techniques employed in the software, hardware, and computer system architecture mechanization of the system do not introduce unacceptable hazards. Typical areas to address are the techniques for detecting, monitoring, isolating, and accommodating failures;	The FACE Technical Standard requires health monitoring APIs for the safety and security profiles (see §4.2.1.3 and §4.2.1.4). The OSS Health Monitoring and Fault Management (HMFM) software component provides the interfaces to allow the software to detect faults, determine the scope of faults or errors, and transition to a fault recovery state, as described in §4.1.3.1 of the Technical Standard. When a fault or error is detected, the HMFM is then initiated to return the system to a healthy or fail-safe state. The HMFM partially supports MIL-HDBK-516C §15.5.4 through the support of failure management functions and the interface with health status once the isolation state has determined the source of the fault and the best action to perform to address it. The HMFM does not provide for failure monitoring and detection techniques but does use the detection of these failures as input to drive the cycle. The HMFM does not verify the safe operation, only that the system transitions to the proper states based on the customer-provided software. It defines transitions between fault recovery states and allows the customer-provided software to decide how to react. Safe operation is determined by the customer – the FACE Technical

FACE Artifact	Contributes to MIL-HDBK-516C: §15.5.4 Dynamic Operation	Analysis
	the entire redundancy management/fault tolerance scheme (from the lowest level through the system level); the techniques for assessing self-health; the techniques employed for determining other channels and external dependent sub-system/system health status; the voting scheme mechanization; the flow of all mode unique inputs through the system; and the safe implementation of internal/functional/sub-system/system interfaces throughout all dynamic conditions/modes/envelopes expected.	Standard only defines the system to interface between the detection, isolation, and recovery states (and repair state, if implemented). HMFM is critical to all systems. The ability to monitor the health of the system and manage faults is key to proper and sustained operation. The challenge with this task is that the platform on which the UoCs run can vary. However, the task could be split into UoC-specific *versus* platform-specific HMFM. The FACE architecture requires an explicit interface to HMFM support for UoCs in other segments. The HMFM provides "standardized methods for detecting, reporting, and handling faults and failures" (§4.1.3). This enables HMFM to be platform-specific without imposing assumptions about the HMFM on UoCs that could be inappropriate to a different platform.

4.5.7 Technical Standard §4.1.3.2 Contributes to MIL-HDBK-516C §15.5.7

FACE Artifact	Contributes to MIL-HDBK-516C: §15.5.7 Restart and Reset Capabilities	Analysis
FACE Technical Standard §4.1.3.2 "OSS HMFM Requirements"	Criterion: Verify that SSSE designs have the necessary provisions to restart and/or reset the system safely while in flight. Standard: The system SSSEs are designed in conjunction with the digital hardware to reset/restart the computer system safely without inducing unacceptable effects (e.g., transients). Aspects of the design include channel/data resynchronization, the system interrupt structure, the system reinitialization, re-check of system health, and reconfiguration to safe states. The design of flight-critical functions accommodates transient time limits dependent on the air system's inherent stability/safety margins, envelope, and flight maneuvers before unrecoverable departure.	The FACE Technical Standard requires health monitoring APIs for the safety and security profiles (see §4.2.1.3 and §4.2.1.4). §4.1.3.2 "OSS HMFM Requirements" outlines that an OSS UoC is required to include a health monitoring capability to restart the module or restart a partition. These capabilities are controlled and managed directly by the OSS UoC. The FACE HMFM system provides the interfaces to invoke the restart capabilities, while the actual restart capabilities are managed by the developed UoC. The FACE OSS HMFM software component must transition to the proper Fault Management (FM) states according to the preconfigured FM policies. The FACE architecture dictates the interface between detection states and allows the UoC to define how to react. Safe operation is determined by the platform-specific requirements and the FACE

FACE Artifact	Contributes to MIL-HDBK-516C: §15.5.7 Restart and Reset Capabilities	Analysis
		architecture requires the system to interface between the detection, isolation, and recovery states (and repair state, if implemented). The reaction to the reset and whether the reset induces unacceptable effects is mitigated by the software component; however, the FACE OSS HMFM requires the software component to complete the cycle once the action has been taken.

4.6 MIL-HDBK-516C Flight Certification Documents

The following table of documents/data taken from §14 of MIL-HDBK-516C indicates which ones are likely to benefit from the use of FACE UoCs in the system. Those marked “N” are less likely to significantly benefit from use of FACE UoCs.

Flight Certification Document	FACE Influence	Rationale
1. System Safety Program Plan (SSPP)	N	
2. Preliminary Hazard Analysis (PHA)	N	
3. Sub-System Hazard Analyses (SSHA)	N	
4. System Hazard Analyses (SHA)	Y	FACE Technical Standard requires features related to health monitoring, fault isolation, and fault management.
5. Operating and Support Hazard Analysis (O&SHA)	Y	FACE Technical Standard requires features related to health monitoring, fault isolation, and fault management.
6. Test Hazard Analyses	Y	FACE Technical Standard requires features related to health monitoring, fault isolation, and fault management.
7. Occupational Health Hazard Assessment (HHA)	N	
8. Specialized analyses such as a sneak circuit analyses and software hazard analyses	Y	FACE Technical Standard requires features related to health monitoring, fault isolation, and fault management.
9. Modification documentation (for correction of safety deficiencies)	N	

Flight Certification Document	FACE Influence	Rationale
10. Component/system test results (waivers/deviations and equipment conditional usage documents)	Y	Although not qualified, CTS provides informal test results demonstrating FACE Conformance.
11. Minutes of system safety group meetings (open items)	N	
12. Minutes of system safety program reviews (open items)	N	
13. Engineering change proposals (safety-related)	N	
14. Hazard identification, evaluation, and correction-tracking system files	N	
15. Safety Assessment Reports (SARs)	N	
16. Test plans and test results	Y	CVM outlines verification methods for each requirement for FACE Conformance. Although not qualified, CTS provides informal test results demonstrating FACE Conformance.
17. Test temporary engineering orders	N	
18. Failure Modes, Effects, and Criticality Analysis (FMECA)/Failure Modes and Effects Analysis (FMEA)	N	
19. Hazard risk index	N	
20. MIL-STD-882, System Safety Program Requirements	N	
21. Test review board reports	N	
22. Safety review board reports	N	
23. Flight readiness review reports	N	
24. Safety requirements traceability matrix (both hardware and software)	Y	FACE Technical Standard requires features related to health monitoring, fault isolation, and fault management.
25. Software System Safety Program Plan (SwSSPP)	Y	FACE Technical Standard requires features related to health monitoring, fault isolation, and fault management.
26. Functional Hazard Analysis or Assessment (FHA)	N	

Flight Certification Document	**FACE Influence**	**Rationale**
27. System of Systems Hazard Analysis	N	
28. Safety-Critical Functions/Safety-Critical Items list	N	
29. Systems Engineering Plan (SEP)	N	
30. Proof of Risk Acceptance	N	

5 Mapping FACE Artifacts to DO-178C Flight Certification Evidence

This chapter identifies artifacts generated during FACE Conformance activities and correlates them to sections of DO-178C, in support of the specified airworthiness requirements. This chapter is organized into sections for each FACE Consortium document, with subsections for each specific mapping to requirements in DO-178C.

5.1 FACE RIG

The RIG provides best practices and examples for implementing UoCs and more detailed information on certain services, addresses safety and security concerns during the planning and design phases of UoC development, provides programming language mapping rules, and explains the FACE Data Architecture.

The RIG Volume 1, §6 on "Safety Guidance" describes overall activities and processes that should be used for development of flight-worthy software. This is a high-level overview, without specific mappings of FACE artifacts to specific requirements of an airworthiness standard, therefore §6 of the RIG should be considered a companion to this document.

5.2 FACE CTS

The CTS is a framework for testing candidate FACE UoCs and Data Models to ensure they meet the interface requirements of the FACE Technical Standard. Data Models must be provided for PCS, PSSS, and TSS UoCs (see the FACE Technical Standard §3.3.1). All requirements tested by the CTS are defined in the CVM, which is maintained by the FACE Consortium. However, the CTS only tests a portion of the total requirements listed in the CVM. All types of UoCs may be tested with the CTS, with specific sections devoted to each type of UoC as well as testing a Data Model.

5.2.1 CTS Test Report Contributes to DO-178C A-4.12

FACE Artifact	Contributes to DO-178C: Objective A-4.12	Analysis
CTS Test Report	Objective A-4.12 is to verify that the software architecture conforms to standards.	A system designed consistently around the FACE Reference Architecture using only FACE UoCs supports the objective. The FACE Conformance verification procedures demonstrate that only the permitted interfaces were used. Passing the CTS contributes toward objectives of DO-178C 11.14 (a) and (c).

5.2.2 Entire CTS Contributes to DO-178C A-5.4

FACE Artifact	Contributes to DO-178C: Objective A-5.4	Analysis
FACE CTS	Objective A-5.4 is to verify that source code conforms to standards.	The output of the CTS provides evidence that the UoC conforms to the coding standards as defined in the FACE Technical Standard. Passing the CTS contributes toward objectives of DO-178C 11.14 (a) and (c). However, because the CTS only checks linking, not compilation, it only verifies the correctness of interfaces. Additional analysis will be needed for other aspects of code standards.

5.3 FACE CVM

The FACE CVM provides the Product Standard that clarifies the requirements from the FACE Technical Standard that a product must meet in order to be certified as FACE conformant. It specifies the technique(s) to be used to verify each of these requirements. The CVM categorizes each element of the FACE Technical Standard, identifying which FACE segments are applicable, which verification methods should be used, and so forth.

Although the CVM indicates how requirements should be verified (by test, by inspection), it is not a verification tool. The CVM is an analysis or accounting tool that acts as a checklist. Thus, in the sections below that map some or all portions of the CVM to airworthiness standards, the listing of “FACE CVM” is meant to indicate Verification Evidence showing conformance to the CVM.

5.3.1 Entire CVM Contributes to DO-178C A-3.1

FACE Artifact	Contributes to DO-178C: Objective A-3.1	Analysis
FACE CVM, as tailored by project	Objective A-3.1 is to verify software requirements comply with system requirements.	The FACE artifacts only contribute to this objective if the project system requirements (or derived requirements) require conformance to the FACE Technical Standard. The CVM focuses mostly on interfaces rather than functionality. Therefore, it will not be sufficient to verify compliance with the project’s system requirements. However, the CVM aids in demonstrating that the system functions (or derived requirements) related to the FACE Technical Standard are satisfied by the software. Another challenge in using the CVM Verification Evidence is that the FACE Technical Standard covers a range of requirements from system to software, and from high-level to low-level.

5.3.2 Entire CVM Contributes to DO-178C A-4.8

FACE Artifact	Contributes to DO-178C: Objective A-4.8	Analysis
FACE CVM, as tailored by project	Objective A-4.8 is to verify software architecture is consistent with software requirements.	The CVM is the authoritative guide to the requirements for conformance to the FACE Technical Standard and is thus the authoritative confirmation that software requirements based on the CVM are consistent with the FACE Reference Architecture. For systems that incorporate UoCs that are not FACE conformant, the associated requirements would need to be verified as consistent by other means. This mapping may cover interoperability requirements, but it does not check consistency of functional or behavioral requirements. Use of this mapping implies that FACE Conformance is a high-level requirement. The partitioning requirements of the FACE Technical Standard are particularly relevant to this objective of the DO-178C standard. The Verification Evidence for the CVM will include evidence of ARINC 653 partitioning for the safety and security profiles. The FACE Data Models may provide evidence of architecture consistency.

5.3.3 Entire CVM Contributes to DO-178C A-4.11

FACE Artifacts	Contributes to DO-178C: Objective A-4.11	Analysis
FACE CVM, as tailored by project Test report generated by CTS	Objective A-4.11 is to demonstrate the software architecture is verifiable.	Although the CTS is not a qualified tool, it provides means to informally confirm that the requirements of the FACE CVM verified by test are met. Software requirements based on the CVM are then easily shown to be verifiable.

5.3.4 Entire CVM Contributes to DO-178C A-5.1 and A-5.2

FACE Artifacts	Contributes to DO-178C: Objectives A-5.1 and A.5-2	Analysis
FACE CVM, as tailored by project Test report generated by CTS	Objectives A-5.1 and A-5.2 are to verify source code complies with low-level requirements and is consistent with software architecture.	The CVM is the authoritative guide to the requirements for conformance to the FACE Technical Standard and is therefore the authoritative confirmation that source code written to meet software requirements based on the CVM are consistent with the FACE Reference Architecture. For systems that incorporate UoCs that are not FACE conformant, other means of verification would be needed for the parts of the architecture beyond the FACE Reference Architecture. The CTS links but does not actually run code. A successful link shows some measure of software consistency but may not be sufficient.

5.4 FACE Technical Standard

The Technical Standard is the key product of the FACE Consortium. It describes the FACE Reference Architecture, interface, and requirements that must be met. “The FACE approach is to develop a Technical Standard for a software COE designed to promote portability and create software product lines across the military aviation domain.” [C207, p.xv] The standard was developed using industry standards for distributed communications, operating systems, and other applicable domains. The FACE Reference Architecture is defined in §3 of the FACE Technical Standard, with sections on architectural segments, standardized interfaces, the FACE Data Architecture, programming language run-times, component frameworks, OSS profiles, and explanation of UoCs.

5.4.1 Entire Technical Standard Contributes to DO-178C A-1.5

FACE Artifact	Contributes to DO-178C: Objective A-1.5	Analysis
FACE Technical Standard	Objective A-1.5 is to demonstrate software development standards are defined.	A system designed consistently around the FACE Reference Architecture using only FACE UoCs supports the objective. The FACE Technical Standard provides design standards for Data Architecture and data modeling. The FACE Reference Architecture defines standardized interfaces between the segments. Coding standards for UoC communication and implementation in various software languages are contained in the appendixes of the FACE Technical Standard. The FACE Technical Standard is under change control. This mapping can be used if the project adopts the FACE Technical Standard as a software development standard, but is not sufficient, as other standards would need to cover the full software life-cycle.

5.4.2 Technical Standard §3 Contributes to DO-178C A-4.10

FACE Artifact	Contributes to DO-178C: Objective A-4.10	Analysis
FACE Technical Standard §3 “Architectural Overview”	Objective A-4.10 is to verify software architecture is compatible with target.	The FACE architecture intentionally abstracts the target hardware and provides an OSS standard interface to enhance portability. UoCs other than the OSS should require little or no change (beyond recompiling) to be compatible with the hardware. FACE Conformance does not contribute to the actual verification activity but makes it more likely that such verification proceeds smoothly.

5.4.3 Technical Standard §3 Contributes to DO-178C A-2.3

FACE Artifact	Contributes to DO-178C: Objective A-2.3	Analysis
FACE Technical Standard §3 "Architectural Overview"	Objective A-2.3 is to develop a software architecture.	A system designed consistently around the FACE Reference Architecture using only FACE UoCs supports the objective. If UoCs that are not FACE conformant are used, the FACE Reference Architecture provides a strong starting point. The following DO-178C sections are referenced from this objective: §5.2.1.a: Architecture is developed from high-level requirements. This is met if the high-level requirements include FACE Conformance. §5.2.2.a: Architecture conforms to software design standards, is verifiable, and consistent. Meeting this section will depend on what design standards were adopted. Consistency is provided by the architecture and defined interfaces. Verification is aided by the CVM and CTS. The CTS is not a qualified tool but passing the CTS may demonstrate that the system architecture is verifiable and consistent to the FACE software architecture. The SDM may help with conformance to design standards and consistency. §5.2.2.d: Interfaces between components defined and consistent between the components. The FACE architecture clearly defines consistent interfaces between UoCs.

5.4.4 Technical Standard §3 Contributes to DO-178C A-4.9

FACE Artifact	Contributes to DO-178C: Objective A-4.9	Analysis
FACE Technical Standard §3 "Architectural Overview"	Objective A-4.9 is to verify software architecture is consistent (DO-178C §6.3.3.b).	The FACE Reference Architecture has a ten-year pedigree demonstrating its consistency. Component relationships are defined in the standard. For systems that incorporate UoCs that are not FACE conformant, verification of the extended architecture would be required. §3 of the FACE Technical Standard describes the architecture. Thus, the standard itself becomes part of the evidence for meeting objective A-4.9. This is not sufficient, however. The project's software architecture description artifacts (which include or point to the FACE architecture) must be provided, along with peer review results verifying consistency.

5.4.5 Technical Standard §4.1.1 Contributes to DO-178C A-4.13

FACE Artifact	Contributes to DO-178C: Objective A-4.13	Analysis
FACE Technical Standard §4.1.1 "Operating System Segment Requirements"	Objective A-4.13 is to verify partitioning.	A FACE OSS under the safety or security profiles requires time and space partitioning. The CVM dictates that these partitioning requirements are verified by inspection. Typically, this would be FACE VA inspection of partitioning verification results provided by the applicant.

5.4.6 Technical Standard §4.2.3 and Annex A Contribute to DO-178C A-1.5

FACE Artifacts	Contributes to DO-178C: Objective A-1.5	Analysis
FACE Technical Standard §4.2.3 "Programming Language Run-Time" FACE Technical Standard Annex A "OSS Profile Details"	Objective A-1.5, from the Software Planning Process, specifies that software development standards are defined.	Objective A-1.5 encompasses the detailed objective (4.1.e) and its associated activities (4.2b, 4.2g, 4.5) and outputs (11.6, 11.7, 11.8). The requirements in the referenced sections of the FACE Technical Standard precisely delineate the subsets of the C, C++, Ada 95, Ada 2012, and Java® programming languages that are allowed for the various capability sets and OSS profiles. These requirements comprise the specification of a software development standard that meets objective 4.1.e; namely, that it is a standard (known as a "Code Standard") that reflects the systems safety objectives for the software. Selecting FACE Technical Standard §4.2.3 and Annex A as an operative Code Standard is a direct application of activity 4.2b. (Note: there may be other Code Standards; e.g., MISRA C® or SPARK Ada, that supplement and further constrain the allowed source constructs). The entire FACE Technical Standard, and thus the sections comprising the Code Standard, are under change control and subject to a well-defined review process, thus satisfying 4.2g. The FACE Technical Standard sections comprising of the Code Standard contribute toward meeting the objectives defined in DO-178C §4.5 ("Software Development Standards"): 4.5a: The Code Standard complies with the relevant subsection of §11 ("Software Life-Cycle Data") as described below. 4.5b: The Code Standard encourages uniformity of implementation, by constraining the code to

FACE Artifacts	Contributes to DO-178C: Objective A-1.5	Analysis
		the specified language subsets. 4.5c: The Code Standard excludes features that are not compatible with safety-related requirements. An output of Objective A-1.5 is the specification of Software Code Standards. The selection of the FACE Technical Standard §4.2.3 and Annex A as a Code Standard directly satisfies 11.8a, which relates to the choice of programming languages and subsets. The other clauses of 11.8 deal with lexical issues, naming conventions, coding style, and tool support. These are outside the scope of the FACE Code Standard. Likewise, outputs 11.6 (Software Requirements Standards) and 11.7 (Software Design Standards) are outside the scope of the FACE Code Standard.

6 Conclusion

Programs of record under the DoD are increasingly requiring components that are FACE conformant due to the benefits of modularity, portability, and ease of integration. The life-cycle cost of developing software that undergoes evaluation for requiring flight certification is typically estimated to be ten times the cost for non-safety-critical software development. This document identifies artifacts and activities associated with pursuit of conformance to the FACE Technical Standard that can contribute towards evidence of airworthiness according to selected flight certification guidance.

This document provides an analysis of sections of both MIL-HDBK-516C and DO-178C for which the FACE documents can contribute to evidence of compliance towards airworthiness. See Chapter 7 for a cross-reference table between the FACE documents and the applicable sections of DO-178C and MIL-HDBK-516C. In addition, Chapter 5 provides a mapping of MIL-HDBK-516C certification documents with an indication of which can be influenced using a FACE UoC. While the FACE Technical Standard does not dictate specific operation of software components, it does provide interface and architecture requirements which contribute to both airworthiness standards and this document provides guidance for those areas which intersect.

The benefit of this document is that it provides additional return on investment to achieve FACE Conformance by leveraging the resulting artifacts and activities to contribute to the airworthiness evidence required for flight certification. Suppliers and purchasers of FACE conformant flight software will benefit from cost reductions in two ways. First, during initial development, the effort spent on FACE Conformance will contribute to evidence of airworthiness. Second, FACE conformant software components are meant to be reused, which means the flight certification evidence can be made reusable as well as providing recurring savings.

7 Cross-Reference Matrixes

7.1 MIL-HDBK-516C

MIL-HDBK-516C Section	Cross-Reference Section	FACE Document
4.1.1	Section 4.4.3	CVM inspection of fully tested requirements
4.2.1	Section 4.2.1	CTS
	Section 4.3.1	Conformance Policy
15.1.2	Section 4.5.1	Technical Standard §3
15.1.9	Section 4.5.3	Technical Standard §3.1 to §3.3
15.1.10	Section 4.4.2, Section 4.5.5	CVM Items F-27, F-29, F-33, F-34
15.2.3	Section 4.5.1	Technical Standard §3
15.2.5	Section 4.2.2	CTS
	Section 4.3.2	Conformance Policy
15.5.1	Section 4.4.1	CVM and Technical Standard
	Section 4.5.4	Technical Standard §3.1 to §3.3
15.5.4	Section 4.5.6	Technical Standard §4.1.3
15.5.7	Section 4.5.7	Technical Standard §4.1.3.2

7.2 DO-178C

DO-178C Section	Cross-Reference Section	FACE Document
A-1.5	Section 5.4.1	Technical Standard
	Section 5.4.6	Technical Standard §4.2.3, Annex A
A-2.3	Section 5.4.3	Technical Standard §3
A-3.1	Section 5.3.1	CVM

DO-178C Section	Cross-Reference Section	FACE Document
A-4.8	Section 5.3.2	CVM
A-4.9	Section 5.4.4	Technical Standard §3
A-4.10	Section 5.4.2	Technical Standard §3
A-4.11	Section 5.3.3	CVM, CTS Test Report
A-4.12	Section 5.2.1	CTS Test Report
A-4.13	Section 5.4.5	Technical Standard §4.1.1
A-5.1	Section 5.3.4	CVM
A-5.2	Section 5.3.4	CVM
A-5.4	Section 5.2.2	CTS

Acronyms & Abbreviations

Term	Meaning
AC	Advisory Circular
API	Application Programming Interface
CMMI	Capability Maturity Model Integration
COE	Common Operating Environment
CSIL	Computer System Integrity Level
CTS	Conformance Test Suite
CVM	Conformance Verification Matrix
DAL	Design Assurance Level
DC/CC	Data Coupling and Control Coupling
DER	Designated Engineering Representative
DoD	Department of Defense
EASA	European Union Aviation Safety Agency
EUROCAE	European Organisation for Civil Aviation Equipment
FAA	Federal Aviation Administration
FACE	Future Airborne Capability Environment
FHA	Functional Hazard Analysis
FM	Fault Management
FMEA	Failure Mode and Effects Analysis
FMECA	Failure Mode, Effects, and Criticality Analysis
HHA	Health Hazard Assessment
HMFM	Health Monitoring and Fault Management
IFF	Identification Friend or Foe
IMA	Integrated Modular Avionics

Term	**Meaning**
INCOSE	International Council of Systems Engineering
O&SHA	Operating and Support Hazard Analysis
OSS	Operating System Segment
PCS	Portable Components Segment
PHA	Preliminary Hazard Analysis
PSSS	Platform-Specific Services Segment
RIG	Reference Implementation Guide
RSC	Reusable Software Components
RTCA	Radio Technical Commission for Aeronautics
RTOS	Real-Time Operating System
SAE	Society of Automotive Engineers
SAR	Safety Assessment Report
SCF	Safety-Critical Function
SDM	Shared Data Model
SEI	Software Engineering Institute
SEP	Systems Engineering Plan
SHA	System Hazard Analyses
SOI	Stage of Involvement
SPA	System Processing Architecture
SSE	Software Supporting Elements
SSHA	Sub-System Hazard Analyses
SSPP	System Safety Program Plan
SSSE	Safety Supporting Software Element
SwSSPP	Software System Safety Program Plan
TSO	Technical Standard Order
TSS	Transport Services Segment

Term	Meaning
TWG	Technical Working Group
UoC	Unit of Conformance
US	United States
VA	Verification Authority

Index