A Unique Approach to FACE conformance DDC-I/OAR

US Army Aviation FACE™ TIM Paper by:

Gary Gilliland/Joel Sherrill

Gary Gilliland, Technical Marketing Manager DDC-I

Joel Sherrill, Ph.D., Director of Research and Development, OAR Corporation

Febuary, 2016

Table of Contents

Executive Summary

A primary goal of the FACE standards effort is to move military aerospace systems development from a "stove-pipe" approach to a set of interoperable and reusable avionics components. Instead of redesigning modules and software for each new system, the prime contractor can instead leverage a set of FACE software building blocks. The benefits to the rapid deployment of future systems should not be underestimated.

The heart of any safety critical avionics system is the operating system on which it is hosted. FACE conformant operating systems (Security, Safety Base, and Safety Extended) are expected to provide hard partitioning between software subsystems as well as a subset of POSIX APIs. These three profiles, targeted at systems typically expected to undergo a safety certification, require a combination of the features historically found in certified ARINC 653 operating systems, as well as the POSIX APIs historically found in real-time operating systems. For many vendors, including DDC-I and OAR, this presents a challenge.

DDC-I develops Deos™, an ARINC 653 conformant operating system which has been the base for hundreds of Level A system certifications, first certified in 1998. There is only one flavor of Deos, and Deos is always built with Level A certification in mind. Deos supports minimal POSIX APIs. RTEMS, similarly, is a native POSIX 52 compliant operating systems (single process/multiple threads), designed to be a classic hard-real time executive. RTEMS was originally released in 1990, and like Deos, enjoys and long history of reliability. RTEMS does not support ARINC 653 partitioning.

Deos and RTEMS are both mature real-time operating systems which have previously targeted different segments of the real-time application space. Each RTOS has capabilities not found in the other. Deos has ARINC 653 partitioning and APIs, while RTEMS has focused on POSIX compatibility to enable application portability while maintaining a deterministic real-time environment. Using the unique approach of combining the two products (Deos for ARINC 653 partitioning and RTEMS for POSIX APIs), FACE OSS safety base conformance becomes achievable.

Operating System Backgrounds

RTEMS

A hard real-time system requires that a set of tasks, or threads, interleave with predictable execution time bounds, known as deadlines. For hard real-time tasks, every deadline must be met, while soft real-time tasks might miss deadlines or put bounds on the maximum tardiness of meeting the deadline. Often, the set of tasks contain a mix of periodic and aperiodic tasks. Periodic tasks have a period that controls the rate at which jobs from the task are released to execute, and a job has a worst-case execution time (WCET); all other tasks are aperiodic. A useful sub-class of aperiodic tasks is sporadic tasks, which distinguish from other aperiodic tasks by having a minimum inter arrival time, thus providing an upper-bound on the rate at which jobs release. Schedulability analysis gives theoretic guarantees about real-time behavior of periodic and sporadic tasks subject to a scheduling algorithm and the task set WCET, deadline, and rate variables.

The primary function of a hard real-time operating system (RTOS) is to help developers of embedded system applications to ensure schedulability guarantees are met for hard real-time tasks by supporting preemptive multithreading. The secondary function is to achieve high quality-of-service for soft real-time and other aperiodic tasks. Tertiary RTOS functions aim to provide services that ease application development, such as synchronization, memory management, file systems, block devices, networking, programming language runtimes, testing support, debug aids, and more.

The Real-Time Executive for Multiprocessing Systems, better known as RTEMS, was conceived in response to RTOS needs identified by the U.S. Army Missile Command (MICOM), which is now the Aviation and Missile Command (AMCOM). AMCOM systems have long life cycles often exceeding those of commercial products and even the companies behind those products. Army engineers wanted the ability to examine, analyze, fix, and improve RTOSs in their systems, but they were closed source, required hefty fees for source code access, and had proprietary APIs which led to vendor lock in. Although per-system licensing costs were a factor, the cost of ensuring license compliance for the life of the system was higher than the licenses themselves. Thus came the motivation for an open source RTOS based on open standards.

Today RTEMS is a successful open-source hard RTOS used in projects such as satellites, space probes, unmanned vehicles, robotics frameworks, automotive data logging, military weapons systems, building automation, medical devices, networking appliances, particle accelerators, and industrial controllers.

A Unique Approach to FACE conformance

Deos

Deos is a real-time operating system (RTOS) that has been developed to the RTCA/DO-178C Level A avionics safety standard, supporting numerous applications since 1998. Deos is optimized to support the development processes and deployment of safety critical applications, especially for those applications that require safety certification, high integrity, high availability, or temporal determinism. For example, Deos features directly support binary reuse of application components and consequently this enables binary reuse of all lifecycle data from plans through certification ("PSAC to SAS" in DO-178C terms). Binary reuse greatly reduces the cost of system development.

Deos is both reliable and robust. We expect Level A software to be reliable. However, Deos is also expected to withstand application failures. The failure of an application cannot corrupt the execution of the kernel or its ability to provide services to other applications. Kernel robustness is most evident during software test. Deos continues correct delivery of services despite the varied and awkward failures that are typical of untested software. Deos builds walls around applications contained in a process/partition. An application within a process/partition cannot break out of its walls to steal another application's processing resources or otherwise interfere with its execution. The strength of these walls allows Deos to concurrently execute software at different DO-178C levels, A through E.

Deos manages computing resources including processing time, physical memory, IO and interrupts. In addition, it manages kernel resources such as processes, threads, semaphores, mutexes, events and mailboxes. Requirements for guaranteed resources are specified using the Deos Integration Tool and the Deos tools will not build a system unless there are sufficient system resources to meet all guarantees.

Deos's design incorporates features to reduce product schedules, schedule risks, and overall program costs for its users by providing the following features:

- Proven RTOS, Middleware and Certification Artifact structure designed for reuse that reduces the amount of software that would otherwise have to be developed
- RTOS partitioning – allows for separation of DAL levels for applications and drivers developed to the required DAL level of the necessary service.
- Sophisticated SLACK scheduler – with an ability to recoup unused but budgeted processing time enabling the highest possible processor utilization.
- Development tools - compilers, debuggers, profilers, time saving DO-178 qualified configuration tools, and a code coverage tool that provides MC/DC equivalence capability.
- Off-the-shelf simulation tooling, and COTS BSPs allow for application development before the hardware is completed
- Product Training performed by DDC-I's engineers experienced in DO-178 certification
- Stage of Involvement (SOI) defense of all RTOS supplier deliverables
- Optional engineering services for customized RTOS services and BSP development
- I/O abstraction interface – delivers application reuse, allows applications to be developed or tested in a black box simulation environment, and provides a lightweight mechanism for interfacing device drivers and end customer applications.

Deos and RTEMS for FACE OSS Safey Base Conformance

Both Deos and RTEMS have long histories of addressing the requirements of real-time safety-critical applications. However, based upon the application domain, there are a variety of RTOS standards available. The FACE Technical Standard is unique in that it requires one RTOS solution to support both the ARINC 653 and POSIX standards to provide the capabilities of the Operating System Segment (OSS). Neither Deos nor RTEMS alone can meet both API requirement sets, but together they can exceed the capabilities required of the Safety Base Profile.

Combining Deos' certified ARINC 653 implementation with RTEMS' robust POSIX API implementation, enables the combined solution to meet the FACE OSS requirements. A combined Deos/RTEMS solution for FACE conformance is shown in the figure below:

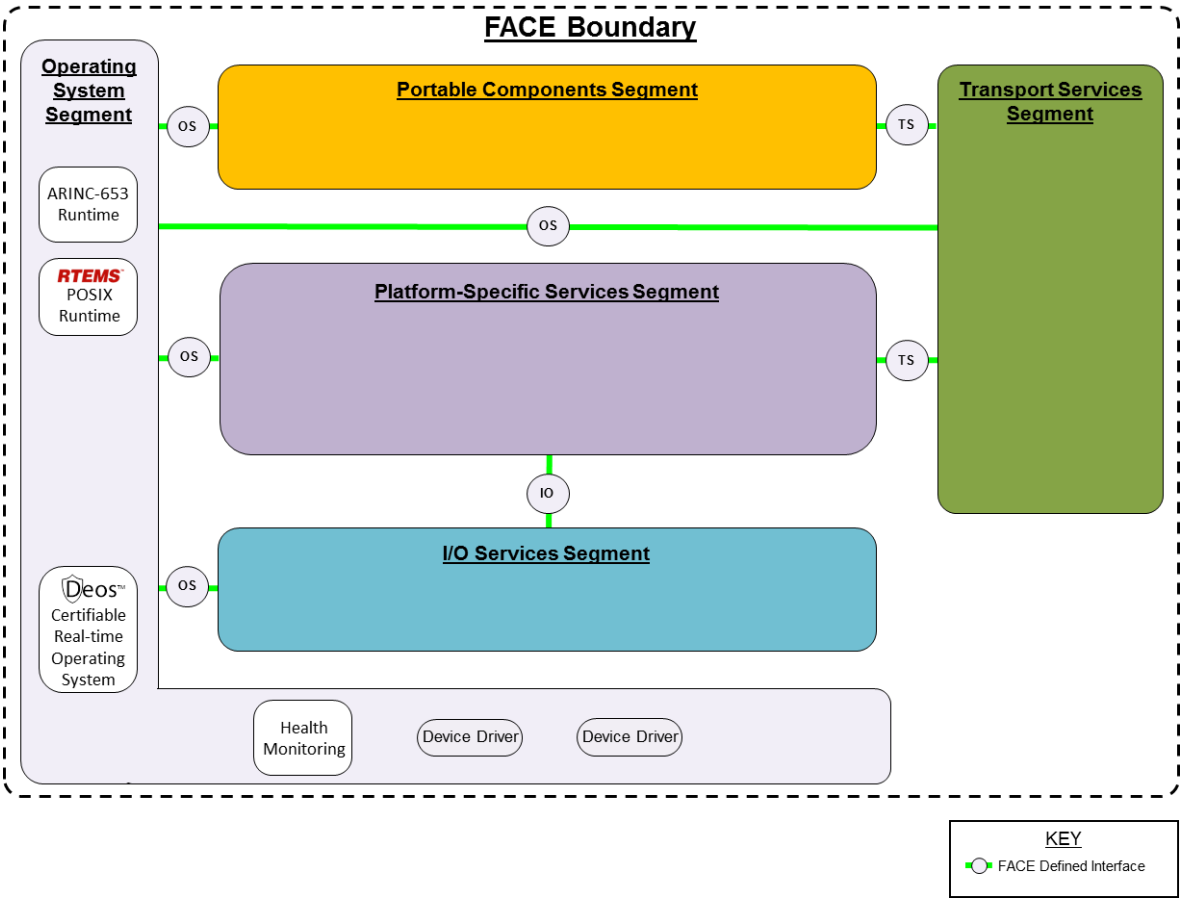

Figure 1 - Deos/RTEMS in FACE Architecture

A Unique Approach to FACE conformance

The figure below illustrates that the Deos partitioning kernel will provide time and space partitioning services for the user developed applications within those partitions. It also illustrates which services will be provided by existing Deos and RTEMS capabilities. RTEMS will be responsible for providing POSIX API services for applications and make selected ARINC 653 services available per the FACE Technical Standard. Just as existing mature RTEMS POSIX support is being leveraged, the mature DO-178C Level A Deos ARINC 653 and TCP/IP services are being used to meet those FACE Safety Base Profile requirements. In addition, other non-POSIX RTEMS services such as stack checking, performance monitoring, a shell, and the Classic API based on the RTEID and ORKID specifications will also be available. Many of these services are helpful while debugging and tuning a system but should not be included in production builds by applications which must be FACE conformant.

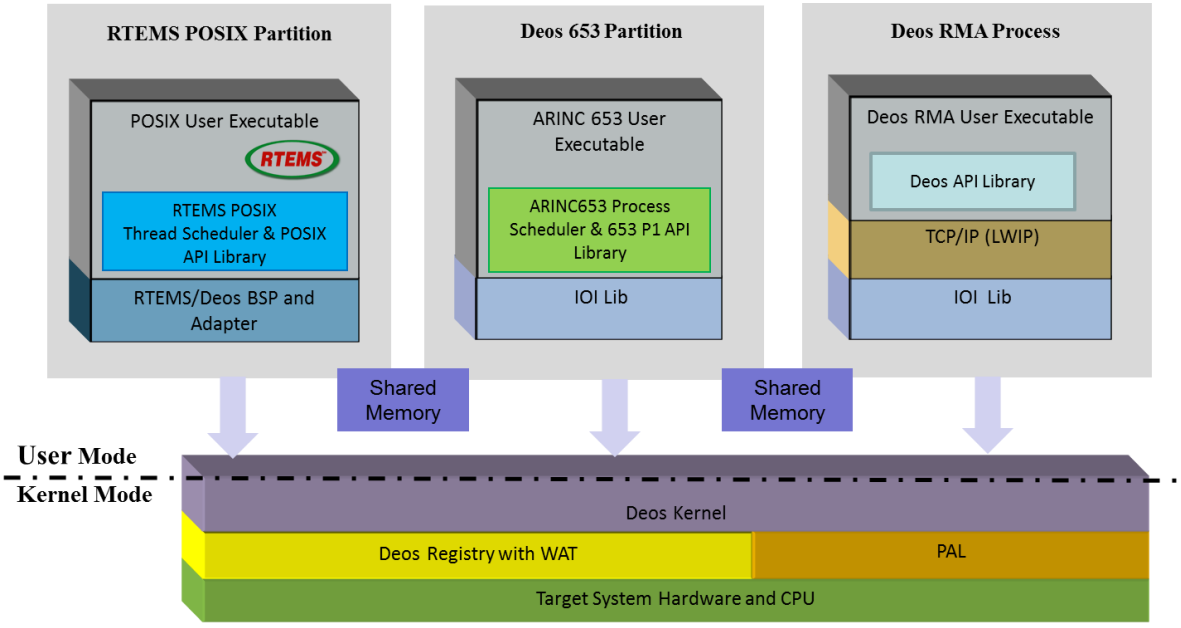

Figure 2 – Detail on Deos/RTEMS in OSS Architecture Diagram

The existing Deos ARINC 653 and TCP/IP services are mature and already available on the Deos kernel. The key element of integrating Deos and RTEMS is the "RTEMS/Deos BSP and Adapter." RTEMS Board Support Packages (BSPs) are the hardware abstraction layer which tailors RTEMS for a specific target board. In this case, the *target board* is the virtualized environment provided by the Deos kernel. RTEMS and applications hosted on RTEMS are executing in a protected address space. This requires that the RTEMS be built in paravirtualized mode to allow it to execute without the use of privileged instructions. The BSP in this case performs simple initialization of the stack and registers and provides a logical abstraction which ends up using Deos kernel services to implement services which would normally be done via direct hardware access. With proper adaptation and sufficient device abstraction, all RTEMS services can be supported in this environment. However, the primary focus is on multithreading including communication and synchronization services.

A Unique Approach to FACE conformance

The Deos kernel will provide the time, space and resource partitioning required to keep the applications in each partition safe from anything running in other partitions in the system. The scheduling of the partitions will follow the ARINC 653P1 specification as shown in the figure below:

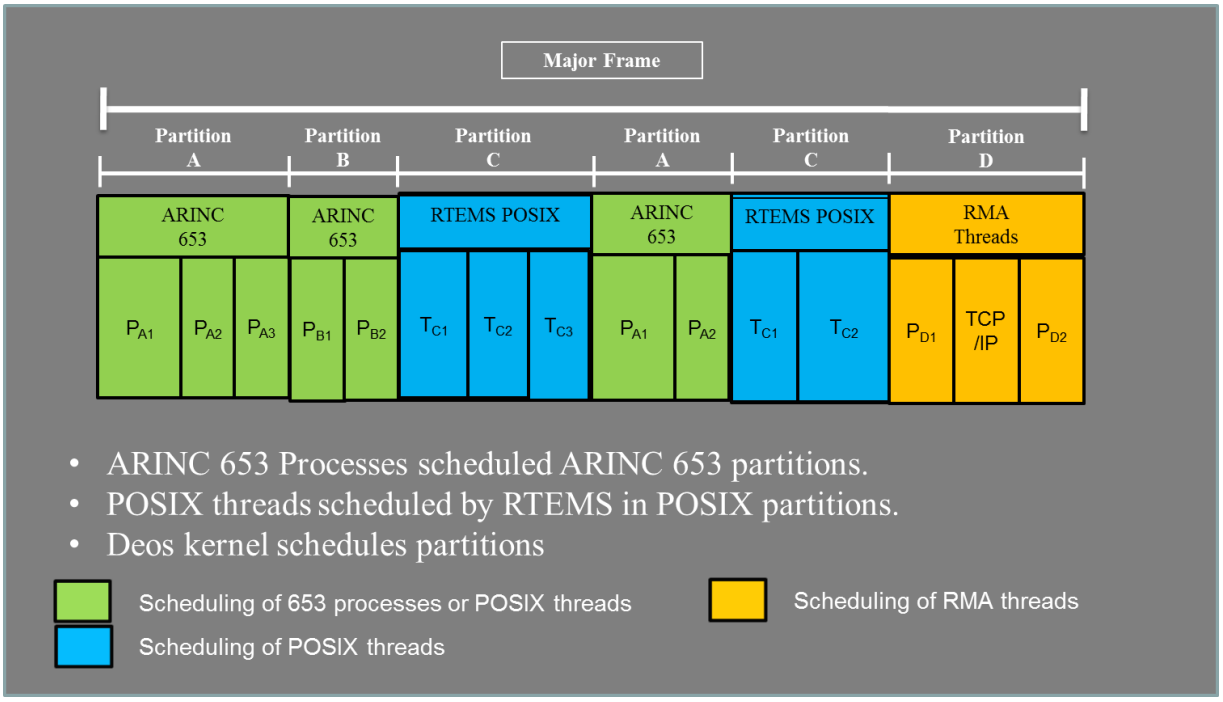

Figure 3 - Partition Scheduling Diagram

Partitions will be scheduled in a time line per major frame. Within each partition the units of execution will be scheduled via a second level scheduler such as specified by ARINC 653, RTEMS, or by Deos in the case of RMA for the TCP/IP stack.

The Security and Safety Base profiles are based on years of industry experience in the development of hard real-time applications. This is apparent when one examines RTEMS POSIX API support. Most of Safety Base was supported before the FACE Consortium existed because the real-time development community expects these POSIX services to be available. Deos was first certified to DO-178B in 1998 and has a long history of safety certifications in the avionics industry. Thus the combination of the two products, Deos for hard memory partitioning per DO-178 and ARINC 653 and RTEMS for the Safety Base POSIX APIs, immediately fulfils the vast majority of the API requirements for the FACE OSS Safety Base profile.

However some functions still must be provided:

POSIX Function	Functional Group
pthread_condattr_getclock()	**POSIX_CLOCK_SELECTION**
pthread_condattr_setclock()	**POSIX_CLOCK_SELECTION**
mmap() methods	**POSIX_MAPPED_FILES**
shm_open() methods	**POSIX_SHARED_MEMORY_OBJECTS**
pthread_setschedprio()	**POSIX_THREAD_PRIORITY_SCHEDULING**
posix_devctl()	**IEEE Std 1003.26, device control**
pthread_getconcurrency()	**XSI_THREADS_EXT**
pthread_setconcurrency()	**XSI_THREADS_EXT**

Table 1 – Missing POSIX Services

The missing services tend to group into two categories. The first category is services that cannot be implemented as intended by the POSIX standard in an environment without a memory management unit (MMU). RTEMS has limited support for MMUs and does not support virtual memory. The memory map and shared memory services cannot be implemented as expected without an MMU. When RTEMS is hosted in a partition by the Deos kernel, the Deos kernel services can be used to provide the necessary virtual memory support.

The second category is methods that have not been requested by the open RTEMS community. This is typically indicative of methods added in newer editions of the POSIX standard that are not in common use. The posix_devctl() is a special case in that it is not part of the primary POSIX standard and not implemented by open source operating systems such as GNU/Linux and FreeBSD. Both of these tend to discourage the method from being commonly used. All of these methods in this category can be implemented; it was just a matter of a user driven requirement. The posix_devctl() method can be implemented within the RTEMS environment and map to normal device driver capabilities provided by RTEMS and Deos device drivers.

The beauty of this design is that the Level A certifiability of Deos remains unchanged as does the open source nature of RTEMS. In fact, each RTOS can be used in its existing form when desired. This results in no impact on existing application for either RTOS. RTEMS can still be used to host applications on bare metal and could be used to host FACE conformant applications (with restrictions) that can be easily migrated to the fully FACE conformant Deos/RTEMS environment.

References

(Please note that the links below are good at the time of writing but cannot be guaranteed for the future.)

G. Bloom, J. Sherrill. Scheduling and Thread Management with RTEMS. ACM SIGBED Review - Special Issue on the 3rd Embedded Operating System Workshop (EWiLi 2013). Volume 11 Issue 1, February 2014.

RTEMS on-line library. http://rtems.org/onlinedocs/doc-current/share/rtems/html/, 2015.

RTEMS: Real-Time Executive for Multiprocessor Systems. http://www.rtems.org/, 2015.

Deos: A time and Space Partitioned DO-178 Level A Certifiable RTOS. http://www.ddci.com/products_deos.php

ARINC Specification 653P1-4, Avionics Application Software Standard Interface, Part 1: Required Services, August 2015

IEEE Std 1003.1-2013: IEEE Standard for Information Technology – Portable Operating System Interface (POSIX®) – Base Specifications, Issue 7, April 19, 2013

Software Considerations in Airborne Systems and Equipment Certification, ED-12B/DO-178B, December 1992

Software Considerations in Airborne Systems and Equipment Certification, ED-12C/DO-178C, January 2012

About the Author(s)

Gary Gilliland, Technical Marketing Manager, DDC-I, Inc.

Gary Gilliland is a Technical Marketing Manager at DDC-I, where he is responsible for technical marketing functions with an emphasis on the safety-critical Deos real-time operating system. He has over 25 years of experience in development and marketing of hardware and software solutions for embedded systems. He has extensive experience with military and commercial avionics and real-time operating systems. Gary is a graduate of the University of Texas at Arlington, where he earned a degree in Electrical Engineering.

Joel Sherrill, Ph.D., Director of Research and Development, OAR Corporation

Dr. Sherrill is Director of Research and Development for OAR Corporation with over twenty-five years of experience in the design, development, and fielding of real-time embedded applications in a variety of commercial, research, and military domains. Dr. Sherrill has been an active member of the free software community for over twenty years. As a principal author and current maintainer of the open-source real-time operating system RTEMS, he has been deeply involved in numerous RTEMS related efforts including the GNAT/RTEMS validation. In addition to being a contributor to numerous free software projects, he is a founding member of the Steering Committee for the Free Software Foundation's GNU Compiler Collection and a board member of the Network Time Foundation. Dr. Sherrill has served as organization administrator and mentor for the RTEMS Project as a participant in the Google Summer of Code™ and European Space Agency Summer of Code In Space for university students as well as the Google Code-In™ for high school students.

Dr. Sherrill has been a key participant in many U.S. Army efforts including the Avenger Slew To Cue upgrade. Dr. Sherrill is currently serving as a representative for the U.S. Army to the Open Group Future Airborne Capability Environment (FACE™) Consortium. As the software architect for the U.S. Army's FC-NET program, Dr. Sherrill oversaw the design and development of the FC-NET architecture. FC-NET is a component based software architecture that provides reusable building blocks for constructing tactical fire control applications. It has been used as the foundation for multiple fire control applications.

As an experienced software developer, Dr. Sherrill likes to focus on the practical application of software engineering research to the everyday life of the developer. In this light, he is currently leading process improvement and implementation efforts for the RTEMS Project.

Education: BS Computer Science, University of Tennessee at Chattanooga. MS Computer Science, University of Alabama in Huntsville. PhD Computer Science, University of Alabama in Huntsville.

About The Open Group FACE™ Consortium

The Open Group Future Airborne Capability Environment (FACE™) Consortium, was formed in 2010 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 FACE Consortium is available at www.opengroup.org/face.

About The Open Group

The Open Group is a global consortium that enables the achievement of business objectives through IT standards. With more than 500 member organizations, The Open Group has a diverse membership that spans all sectors of the IT community – customers, systems and solutions suppliers, tool vendors, integrators, and consultants, as well as academics and researchers – to:

- Capture, understand, and address current and emerging requirements, and establish policies and share best practices
- Facilitate interoperability, develop consensus, and evolve and integrate specifications and open source technologies
- Offer a comprehensive set of services to enhance the operational efficiency of consortia
- Operate the industry's premier certification service

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