MILS ARCHITECTURE: MULTIPLE INDEPENDENT LEVELS OF SECURITY

HIGH-ASSURANCE SECURITY AT AN AFFORDABLE COST

By Objective Interface Systems

On April 23, 2000, in Queensland, Australia, a man named Vitek Boden, age 48, became an environmentalist's worst nightmare. Using a computer and radio transmitter, he released millions of liters of untreated sewage along Australia's Sunshine Coast, where it contaminated parks and hotel grounds, blackened creeks, and killed marine life. Boden was a disgruntled ex-employee of a company that supplied remote control and telemetry equipment to the Australian Water Utility. Using a laptop computer, Boden had command of 300 control nodes governing sewage and drinking water, and could easily have done far worse damage if that were his intent. Apprehended by police on his 46th attempt, Boden admitted he was angling for a consulting job to "fix" the problems he had created.

Although cases like this are rare, they illustrate the far-reaching consequences of embedded systems without adequate security measures, such as authentication of administrators and operators. "The problem is that programmable logic controllers, digital control systems, and supervisory control and data acquisition, or SCADA, systems were never designed with security in mind," according to a report entitled "SCADA vs. the hackers" in *Mechanical Engineering*, December 2002.

As the above example illustrates, information networks are vulnerable to catastrophic failures and even deliberate attacks. Military, public utility, transportation, and other mission-critical operations are not immune to the dangers. The National Institute of Standards and Technology (NIST) has identified 30 distinct categories of threats to information infrastructures, ranging from operator errors to hacker intrusions to viruses.

Now that the Internet and high-speed communications have made it possible to connect military and aerospace systems throughout the world through the U.S. Department of Defense's Global Information Grid (GIG), information networks are more vulnerable than ever. Computers on fighter aircraft,

unmanned vehicles, tanks, and aircraft carriers, as well as embedded processors in radios and wireless devices used in combat situations, will all be nodes on a global network. The National Security Agency (NSA), CIA and coalition forces all provide and share intelligence on this global grid, requiring dynamic policy management based on rapidly changing political realities. International coalitions may be formed to address the threat of the moment. When the threat disappears, the coalition may dissolve.

The connectivity that enables this fluid and dynamic policy management also dramatically escalates security risks to defense systems. A single infected node on the GIG could spread like a cancer, putting lives at risk. With the threat of global cyber-terrorism as well as pandemic viruses and worms, the stakes have never been higher for creating high-assurance security systems.

At what cost?

As security pressures mount, budgetary pressures in both the military and commercial sectors are also creating a rising demand for commercial off-the-shelf (COTS) systems that can meet high-assurance security requirements within reasonable costs. Competition in the commercial sector holds down costs, and industry-standard solutions enable software updates and changes much more cost-effectively than with proprietary systems.

The NIST and the NSA have established a program under the National Information Assurance Partnership (NIAP) to evaluate IT product conformance with international standards.

> "September 11 was a wakeup call. It changed the face of the world. It's why there's a new synergy between COTS vendors and defense contractors. We knew we could do better with cost-effective commercial products that meet security standards."
>
> – *Dr. Ben A. Calloni, research program manager, Lockheed Martin Aeronautics Co.*

The program, officially known as the NIAP Common Criteria Evaluation and Validation Scheme for IT Security (CCEVS), is a partnership between the public and private sectors. Its goal is to establish a national program for the evaluation of information technology products for conformance to the International Common Criteria for Information Technology Security evaluation.

A Department of Defense Directive (DoDD 8500.1) of 2002 mandates that any commercial communications product or service used in the Global Information Grid MUST be NIAP-certified to the appropriate level of security. Whether a desktop application or a sophisticated weapons control system, *any commercial technology used in high-risk environments must meet NIAP*

standards – a surprisingly little known requirement with sweeping implications for the IT industry.

Until recently, the cost of evaluating and certifying security at the highest Evaluation Assurance Level (EAL7 of the Common Criteria) has been prohibitive because the size of the software code to be evaluated was too large, making the process too slow and too expensive for all but the most mission-critical systems. An evaluation might cost over $100 million and be almost impossible to complete. This makes the evaluation of such new technologies improbable at best.

The emergence of MILS

An enabling architecture known as **Multiple Independent Levels of Security (MILS)** is in the process of dramatically reducing the size and complexity of security-critical code, thus allowing faster and more cost-effective development and evaluation. MILS is based on work initiated by John Rushby in the early 1980s, and has evolved in a cooperative effort among government, education and commercial organizations including the U.S. Air Force Research Laboratory, National Security Agency, SRI International, University of Idaho, Lockheed Martin, Boeing, Rockwell Collins, MITRE, Object Management Group (OMG), The Open Group, Objective Interface Systems, Green Hills Software, LynuxWorks, Wind River, and others.

> "The whole point of MILS is really simple: to dramatically increase the *scrutiny* of security-critical code and dramatically reduce the *amount* of security critical code."
>
> *– W. Mark Vanfleet, Senior Cryptologic Mathematician and Senior Information Security (INFOSEC) Systems Security Analyst, National Security Agency*

MILS is a departure from operating system architectures that were designed prior to the Internet, when there was little threat of network attacks. As a result, these early systems did not incorporate security as *a design requirement*. In response to inevitable failures and intrusions, patches were developed over time to plug specific security holes. This "fail-first, patch-later" approach is unacceptable for any mission-critical system.

The central idea behind MILS is to partition a system in such a way that (1) the failure or corruption of any single partition cannot affect any other part of the system or network, and (2) each partition can be security-evaluated and certified separately, so that no partition needs to be evaluated at a higher level than is required for its particular function. For the first time, developers will be able to base their applications on secure, high-assurance foundations.

Multiple levels of security for multifunctional systems

In the early 1980s, the DoD issued the "Orange Book," a set of criteria used for evaluating the security features of computer systems. It became widely used in the IT industry as a benchmark for security standards. However, Orange Book security fell short in two areas:

1. Higher assurance levels required both mathematical verification of trusted system components, as well as significant security functionality in those trusted system components. The code size made mathematical verification almost impossible.
2. Intersystem communication was not addressed by the Orange Book. Trusted components and device drivers ran in privileged mode for performance reasons. Security-critical application code also ran in privileged mode. This was a nightmare to evaluate, and typical evaluations cost on the order of $100 million.

As a result, implementing Orange Book standards became expensive and problematic, mainly because of the limitations of microprocessors in the 1980s. The tremendous increase in microprocessor performance has enabled new paradigms of security.

Often, one system has the job of performing several different functions, especially as processors increase in performance. If such a multi-functional system must meet different levels of safety or security criteria for each of its functions, there must be some guarantee that lower-security functions cannot interfere with higher-level functions – under any circumstances.

Such systems require Multiple Independent Levels of Security, or MILS, as the NSA designates them. MILS system designers must guarantee that unintended interactions are not possible. Otherwise, systems integrators would have to integrate each function individually on a separate processor, which would increase costs and system complexity. In some applications, such as fighter aircraft, separate processors would also add weight, take up space, and consume power – a serious design drawback. MILS implementation on a single processor is both cost-effective and possible with today's technology.

MILS is not a revolution of new ideas over old, but old ideas coming of age – now that technology has caught up.

One size does not fit all

MILS combines the best of the safety and security worlds to create a better solution than either could have devised. It draws upon FAA DO-178B Level A Safety technology and Common Criteria EAL7 Security technology to enable

MILS Web and network services for mission-critical embedded and real-time systems including high-assurance weapons, training and communications systems and C4I platforms.

MILS is founded on the understanding that security is not a one-size-fits-all proposition, and that the security level should be appropriate to the application. The Common Criteria's Evaluation Assurance Levels range from EAL1, the very basic level, to EAL7, the highest level of assurance. Various military systems require EAL assurance levels according to the value of their data and the threat that they encounter. A set of assurance requirements, between EAL6 and EAL7, called "High Robustness" is required when top secret, secret, confidential, and unclassified data reside on the same node.

Military command centers derive information from a variety of sources, from weather forecasting systems to fighter jets to commanders and allied forces in the field. Users within intelligence agencies and the DoD wrestle with information on multiple computers handling information at varying security levels.

> "In aircraft, space, weight, and power are critical factors. If security measures require duplicated hardware, this means additional weight and that's a problem. MILS enables critical safeguards without adding hardware."
>
> *– Jahn A. Luke, Senior Program Manager, Embedded Information Systems Branch, Information Directorate, Air Force Research Laboratory*

An operating system that can simultaneously support ubiquitous commercial applications running on Windows or Linux, along with a variety of mission-critical or high-assurance applications, is the holy grail of computing.

Without such a capability, system designers need to use multiple hardware devices to meet varying security requirements. This type of hardware separation is costly and awkward. An architecture that can support secure partitioning, commercial or legacy applications, multi-level communication, secure user authentication and trusted path, and secure cross-domain information transfer – in a single processor – is the promise of MILS.

Minimum code = affordable cost for high assurance

MILS architecture separates security mechanisms into manageable components. Processes are isolated into partitions that comprise a collection of data objects, code and system resources. These individual partitions can be evaluated separately. This approach substantially reduces the proof effort for secure systems.

To support these partitions, the MILS architecture is divided into three layers:

- *Separation Kernel*
- *Middleware, including the Partitioning Communications System (PCS)*
- *Applications*

Figure 1 is a basic architecture diagram.

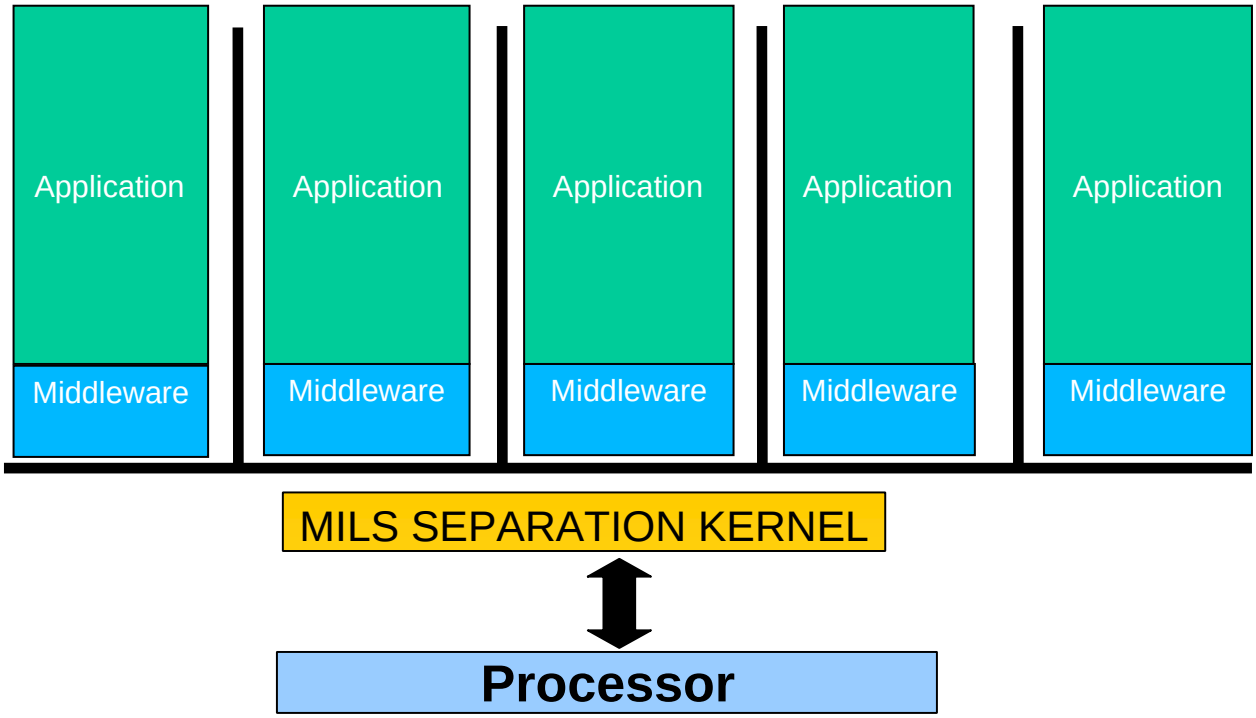


Figure 1

While these terms have been used since the days of the PDP-11, what is different is the assignment of functions to these layers.

- **Separation Kernel.** The MILS separation kernel divides the computer into separate address spaces and scheduling intervals, guarantees isolation of the partitions, and supports carefully controlled communications among them. Because the separation kernel performs these functions and only these functions, the source code can be small – roughly 4,000 lines of C language code. This makes it fast and practical to verify using formal analysis methods (mathematical verification) and to do the exhaustive testing and comprehensive documentation required for the highest level certifications. The separation kernel requires the highest level of authentication, and is the only piece of software that runs in privileged mode. Therefore, no

other code, *not even device drivers*, has the ability to affect the processor's protection mechanisms. Everything else, including all middleware, runs in user mode.

The small size of the separation kernel is a manifestation of the most important MILS design objective: *Dramatically reduce the amount of policy enforcement code so that we can dramatically increase the inspection of that code.* It is because of this rigorous inspection and evaluation that the MILS separation kernel can be *trusted.*

- **Middleware.** Most of the traditional operating system functions have been moved from the operating system to "middleware services," e.g., file systems, device drivers, trusted path, etc. Middleware services include a Partitioning Communications System (PCS) to extend the scope of the separation kernel to inter-system communication. It also includes traditional middleware like CORBA (Common Object Request Broker Architecture), DDS (Data Distribution Service) and Web services. Middleware resides in the same kind of partition as the application that it supports, either co-resident with the application or in a partition by itself. Middleware runs in unprivileged (user) mode, making these services subject to separation kernel policy enforcement. The services that previously ran in privileged mode as part of the operating system, such as memory allocation, device drivers, I/O primitives, file systems, and network stacks now run in user mode in the MILS middleware layer. Some middleware components don't need to be certified at the highest level, and because they can be confined to one partition, they can be evaluated and certified at the appropriate level at much less cost. For example, if a component such as real-time CORBA is running in a classified partition and another instance of it is running in an unclassified partition, only the classified instance of CORBA needs to be certified.

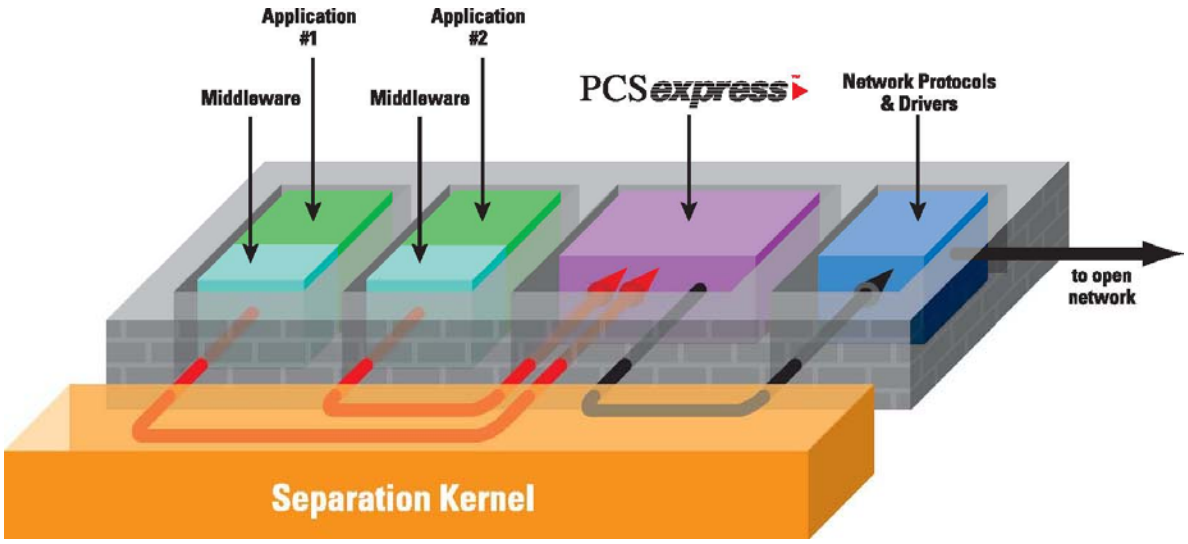
- **Applications.** The application level entities manage, control, and enforce their own application-level security policies, such as firewalls, crypto services and guards. Instead of the fail-first-patch-later approach, trusted components are mathematically verified so that they are:

  - **N**on-bypassable,
  - **E**valuatable,
  - **A**lways invoked, and
  - **T**amperproof.

  Taken together, these form the acronym NEAT. In order to be effective, all system protection must be NEAT.

To satisfy the High Robustness requirements of EAL6/7, engineers must design the system with security in mind from the start and must make it possible to decompose every system function into successively smaller subsets, down to a simple, provable module, each step demonstrating that mechanisms are “NEAT.” This formal proof requires extensive analysis, documentation, and review. It is economically infeasible to achieve High Robustness unless the system is designed from the inception to be "provable." This cannot be added on after the fact.

When we create a distributed system configuration, we would like it to be as safe or secure as if it were just a single processor. We accomplish this by implementing *end-to-end* enforcement of the basic MILS separation kernel policies. The Partitioning Communications System (PCS) is the enforcement mechanism. The collection of MILS nodes in a distributed system is called an *enclave,* and the PCS is present in each node in the enclave. The PCS fits between the applications and the partitions implementing network protocols. Figure 2 is an illustration in which the separation kernel has been omitted for simplicity.



The secure separation kernels developed by companies such as Green Hills, Wind River, and LynuxWorks provides the ability to separate multiple address spaces. In one millisecond, the system may perform a safety-critical task, the next millisecond, not; the non-safety-critical and safety-critical won’t interfere with each other.

The separation kernel is microprocessor-centric. On this microprocessor, one can build a firewall that separates applications – top-secret from not, safety-critical from not – and guarantee that those applications won’t talk to each other without an application-centric firewall. The separation kernel makes

decisions about what goes on at the microprocessor level, but it knows nothing of the network. It just secures this one node.

The PCS takes this secure environment in the separation kernel and extends it to an enclave of computers – two, 100, or 10,000 computers. There will still be an application-centric firewall that separates applications, but it must by NEAT. Partitions are no longer restricted to being on the same processor. There could be hundreds of microprocessors, but one can still guarantee the firewall is tamperproof and nonbypassable. The MILS architecture makes it possible to secure tens of thousands of computers in a global information grid – on fighter aircraft, tanks, aircraft carriers and destroyers.

MILS enables protection against malicious software, internal mistakes and failure

Malicious software can successfully attack the system's hardware or software foundations and render any form of security useless. Security "patches" that do not address security at the foundation level are vulnerable to the following forms of attack:

- *Bypass* – An attacker uses a flaw in a security system to circumvent security mechanisms to get system or network access. The actual point of entry is through either a hardware device or a program that enables the user to access the system without going through security clearance procedures such as authentication. A bypass may be put in place by an attacker, or it may be a design flaw, or even a diagnostic facility accidentally left in place by developers.

- *Compromise* – An invading program reads private data. An example is spyware. If invasive software can monitor the data of programs running on the system then security has been breached.

- *Tamper* – An attack that makes unauthorized modifications to data or program code. If tampering is possible then no application is safe from viruses and worms.

- *Cascade* – Malicious users or software cause failures to cascade from one system component to another. If the failure of one application can cause another application to fail, then it may be possible to bring down the whole system.

- *Covert Channel* – Information is leaked to an unauthorized recipient through a communication channel that is accidental or unintended. By detecting the presence or absence of a message, for example, an unauthorized observer can derive information about the activity of the communicating parties.

- *Virus* – Malicious software invades privileged functions to infect all parts of the system and spread to other connected systems.
- *Subversion* — Malicious software is innocently loaded into the system by an authorized user who mistakenly believes the software is legitimate.

Secure systems are built on secure foundations

The MILS architecture enables the construction of applications that can protect against all the threats named above through the implementation of four key security policies – information flow, data isolation, periods processing, and damage limitation – and only these policies:

- **Information flow** - Information flow from one partition to another is from an authenticated source, to authenticated destinations, and to nowhere else.
- **Data isolation** – Memory that is allocated to a partition can only be accessed by the software in that partition; private data remains private.
- **Periods processing** – The microprocessor and any networking equipment will not be used as a covert channel to leak information to listening third parties. For example, one of the many functions of the PCS is to suppress covert storage and timing channels on a communications link.
- **Damage limitation** – Damage is limited by preventing a failure in one partition from cascading to any other partition. Failures are detected, contained, and recovered from locally.

By reducing core functionality to these four key security policies, MILS not only provides increased security across the board, it also makes mathematical verification simpler and more cost-effective.

Case in point: JTRS and Software-Defined Radio

Why is MILS important? Imagine you're a field commander in the most demanding security environment: a combat situation. You have secret information to share with allied commanders. You have top-secret communications to send to Central Command, and unclassified information sent to soldiers in the field.

Radio often provides the only means of communication in high-risk military environments. Unfortunately, military personnel have been unable to trust

that radios will be effective at separating multiple levels of classified and unclassified transmissions. They need to know that secret communications on one channel intended for U.S. forces only (classified as “NOFORN” or not foreign) won’t bleed into unclassified channels, or be intercepted by hostile third parties.

Further complicating the matter is the wide number of incompatible devices in the field, including aging legacy technology. Ensuring interoperability among different types of field-based radios operating at different frequencies is mission-critical.

Until recently, manual separation of classified messages and hand delivery have been the only secure options available. Now, however, the Joint Tactical Radio System (JTRS) has been initiated by the Department of Defense to provide a flexible new approach to meet diverse warfighter communications needs – through high-assurance software programmable radio technology, or software-defined radio (SDR).

MILS is a perfect match for the JTRS because it ensures a high level of security while enabling modularity of new capabilities, scalability of bandwidth and channels, and backwards compatibility with legacy radios. It also supports the dynamic intra- and inter-network routing of data transport that is transparent to the radio operator.

Objective Interface Systems views MILS as the secure foundation to protect and enable real-time CORBA-based SDR systems – initially for building the most effective secure software-based radio system possible for mil/aero use, and later for providing flexible and secure wireless communications for the commercial markets. As an active member of the OMG, Objective Interface is leading the development of a real-time and MILS-compliant profile for CORBA. Objective Interface’s ORB*express* middleware solution offers a commercially available SDR platform that enables interoperability through software modifications, not hardware changes. As a result, future software radios will be interoperable much like the international phone system.

Beyond embedded systems

For military applications, for homeland security, or for commercial purposes, MILS is a critically important architecture and is currently under active development. For example, the Air Force Research Laboratory, Information Directorate (Wright Research Site), is working on multiple MILS-related contracts with contractors such as Boeing, Lockheed Martin and Raytheon.

Beyond weapons and defense systems, MILS security could be used in medical applications. Today, if there are not enough surgeons in Iraq to attend to wounded soldiers, a doctor can operate on a patient from the U.S.

using special eyeglasses and gloves that remotely control robotic arms and instruments. If this were a MILS-based system, this delicate procedure could occur in a secure environment, without malicious interference.

Banks are showing increasing interest in MILS for protecting ATM networks from costly fraud and abuse. In manufacturing and process control applications, automated warehouses and assembly lines are controlled by computer networks; in a matter of seconds, a single hacker could inflict millions of dollars in damage to sophisticated equipment as well as to materials. MILS is used today to a limited extent in process control, but the potential for wider use is growing.

Nuclear reactors present another example of systems that must be absolutely tamperproof. MILS has already been proposed for use in controlling the U.S. power grid – to prevent blackouts like that of August 2003 that shut down much of the Northeast and Midwest.

Objective Interface is playing a leading role in a consortium of government organizations and commercial vendors focused on developing new safety and security standards for high-assurance applications. In the first public presentation of the security requirements for MILS middleware made to The Open Group, a vendor-neutral standards body, Objective Interface introduced a Protection Profile for the PCS that defines MILS middleware requirements. In practical terms, the PCS allows engineers to design more flexible and affordable highly-secure distributed systems. It also eases the certification process required to process and transmit multiple levels of classified data across a network.

Summary

Past efforts at making software truly secure usually added complexity and high cost. Layers of protection were added on top of the operating systems, middleware, and the applications. Sometimes these layers interfered with each other, had unintended side affects, or were not completely consistent with each other, giving both bugs and attackers the initial crack in the wall they needed to inflict damage.

The MILS approach is precisely the opposite. Systems are made more secure by making the protection simpler. Because it is simpler, it can be trusted to work under all conditions. The processor, via the MILS separation kernel, is tightly controlled. All protections built into the system will be *composable* – that is, the components will work the way they were designed to work and information will flow between them only the way that it should. The PCS provides the same assurances for distributed systems.

In collaboration with its partners, including the U.S. National Security Agency, U.S. Air Force Research Laboratory, the University of Idaho,

Lockheed Martin, Raytheon, Boeing and Rockwell Collins, Objective Interface is working to integrate several MILS security separation kernels with Objective Interface's high-performance implementation of the PCS architecture, PCS*express*. Objective Interface is developing PCS*express* as well as real-time MILS versions of its signature products, ORB*express* and DDS*express*.

The MILS Separation Kernel Protection Profile (SKPP) is under final review by members of The Open Group. Once evaluated and endorsed by The Open Group, the SKPP will be officially evaluated and endorsed by the National Information Assurance Partnership (NIAP) as a validated protection profile, probably during the end of 2005. Developers can use the draft SKPP to plan MILS-based systems. The draft is available for download from

http://www.niap.nist.gov/pp/draft_pps.

The MILS architecture is being applied today and will continue to be important in the most demanding applications where failure is unthinkable: airborne software and national security systems. Because it is both secure and affordable, it will be practical to use this architecture in commercial applications and anywhere system failure or unauthorized access will have significant or even life-threatening consequences.

For more information about MILS and current news of MILS developments, visit http://mils.ois.com.

Acknowledgments

Objective Interface is grateful to the following MILS experts and leaders for their input into this white paper:

- Dr. Ben A. Calloni, Research Program Manager, Lockheed Martin Aeronautics Co.
- Jahn A. Luke, Senior Program Manager, Embedded Information Systems Branch, Information Directorate, Air Force Research Laboratory
- W. Mark Vanfleet, Senior Cryptologic Mathematician and Senior Information Security (INFOSEC) Systems Security Analyst, Department of Defense