# MILS Middleware: High Assurance Security for Real-time, Distributed Systems


Bill Beckwith

bill.beckwith@ois.com

Objective Interface Systems, Inc.
13873 Park Center Road, Suite 360
Herndon, VA 20171-3247
703/295-6500 (voice)
703/295-6501 (fax)
http://www.ois.com/
info@ois.com

# Agenda

- **Introduction**
  - Sponsors
  - Objective
  - Real-time Security Challenge
  - Distributed Security Requirements
- **Partitioned Communications System**
- **Real-time MILS CORBA**
- **MILS Middleware Realization**


# Introduction

# Sponsors

- **Sponsors for PKPP and RT CORBA PP effort**
  - Wright-Patterson AFB (WSSTS Project)
  - Air Force – F22 and JSF (via LM Ft. Worth)
  - Objective Interface

# Objective

- **An applications and communications infrastructure that provides:**
  - Safety — high correctness (does what it is supposed to do)
  - Security — high integrity (… and nothing else!)
  - High performance — low latency/high bandwidth
  - Fault resilience — high reliability
  - Real-time — high predictability
  - Standards-based — high independence

# Real-time Security Challenge Requires New Thinking

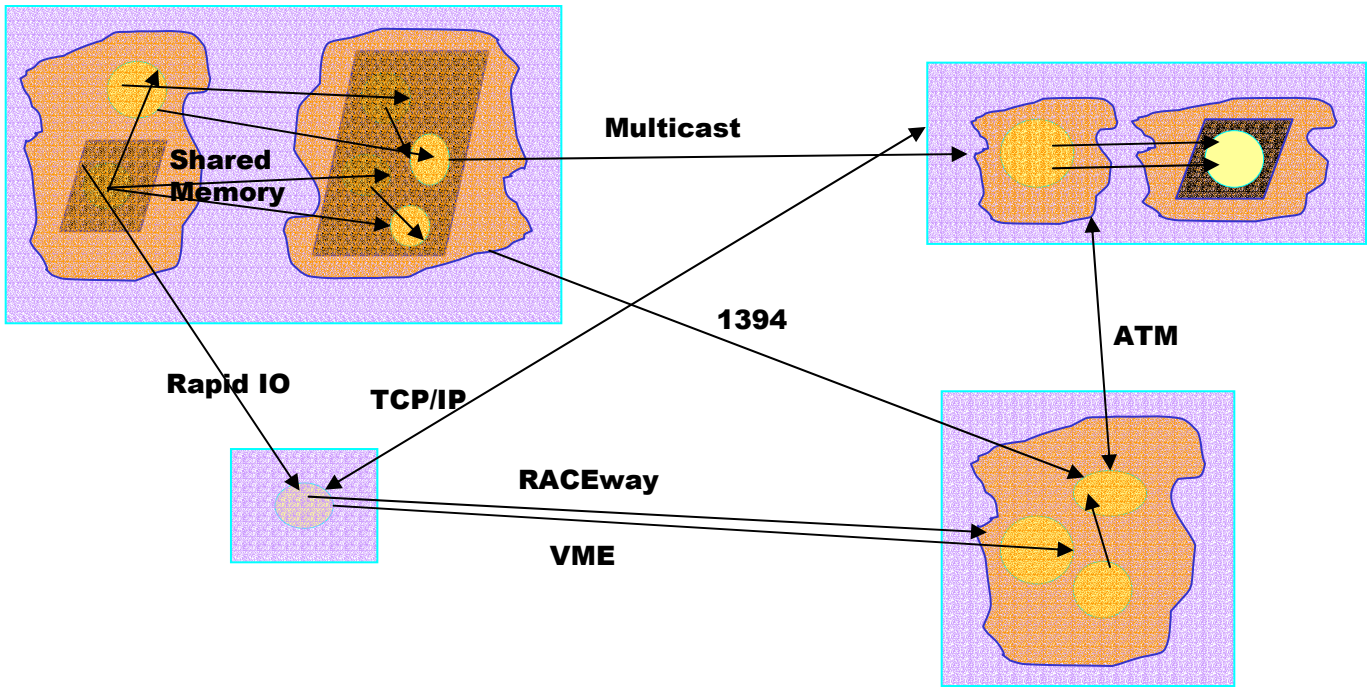
- **Old approaches to security don't work for real-time**
  - Monolithic security kernels
    - Too large
    - Too slow
    - Unpredictable
    - Don't support inherent distribution of embedded systems
    - Can't get above EAL 4 (medium assurance)
  - CORBA Security
    - Weak trust model
    - Too large
    - Too slow
    - Unpredictable
    - Can't get above EAL 4 (medium assurance)
- **New thinking required**
  - MILS: Multiple Independent Levels of Security

# Distributed Security Requirements

- **Rely upon partitioning kernel to enforce middleware security policies *on a given node***
  - Information Flow
  - Data Isolation
  - Periods Processing
  - Damage Limitation
- **Application specific security requirements**
  - must not creep down into the middleware (or kernel)
  - ensure the system remains supportable and evaluatable
- **Optimal inter-partition communication**
  - Minimizing added latency (first byte)
  - Minimizing bandwidth reduction (per byte)
- **Fault resilience**
  - No single point of failure

# Distributed System

- **Distributed object communication**
  - Partition Local – same address space, same machine
  - Machine Local – different address space, same machine
  - Remote – different address space, on a different machine



# Inter-object Communication

- **Unrestricted information flow between objects creates a significant security concern**
- **Three types of inter-object communication**
  - Intra-partition: between objects in same partition
    - left to application
  - Inter-partition: between objects in partitions on the same node
    - Node = one CPU or tightly coupled group of CPUs
    - Unencrypted communication
    - Middleware controls
  - Inter-partition: between objects in partitions on different nodes
    - Encrypted communication
    - Middleware controls

# Secure Communications Framework

- **Security concerns include:**
  - The correct processes occur within an application
    - e.g., the correct waveform and applications are instantiated
  - Inter-object messages flow between the correct objects
    - Information flow security policy
  - "Trusted" objects are capable of instantiating and tearing down applications
  - Permissions enforced on read, write, & execution of files
  - Boots correctly with integrity
    - Implementation dependent
      - Are all of the puzzle pieces present and correct?
      - Is my piece of the puzzle correct?
  - Audit records are understandable
    - Implementation dependent
  - Audit record protected from unauthorized changes or reading

# Why CORBA Security Is Not Used

- **Can't accommodate MLS**
- **Provides a overly wide trust model**
  - Commercially available CORBA security implementations place CORBASEC within the ORB and application
  - ORBs reside within the application components' process spaces
  - Security mechanisms can be modified by other software objects within process space (e.g. application)
    - Security mechanisms can be bypassed
    - Security becomes an application option
- **CORBA Security services**
  - More of a collection of security APIs than a security architecture
- **CORBASEC implementations too large to evaluate**
- **Non-security issues related to message latency**
- **Not appropriate for medium or high assurance systems**

# Layer Responsibilities

## MILS Partitioning Kernel Functionality

- Time and Space Partitioning
- Data Isolation
- Inter-partition Communication
- Periods Processing
- Minimum Interrupt Servicing
- Semaphores
- Timers
- Instrumentation

## MILS Middleware Functionality

- **RTOS Services**
  - Device Drivers
  - CORBA
  - File System
  - …
- **Partitioned Communication System**
  - Inter-processor Communication


# Partitioned Communication System

# PCS: Partitioned Communication System

- **Partitioned Communication System**
  - A portion of MILS Middleware
  - Responsible for all communication between MILS nodes
- **Purpose**
  - Extend the protected environment of MILS kernel to multiple nodes
- **Similar philosophy to MILS Partitioning Kernel**
  - Minimalist:
    - Only that functionality needed to enforce end-to-end versions of policies
      - *End-to-end* Information Flow
      - *End-to-end* Data Isolation
      - *End-to-end* Periods Processing
      - *End-to-end* Damage Limitation
  - Designed for EAL level 7 evaluation

# PCS Objective – What?

- **Just like MILS:**
  - Enable the Application Layer Entities to
    - Enforce, Manage, and Control
  - their own
    - Application Level Security Policies
  - in such a manner that the Application Level Security Policies are
    - Non-Bypassable,
    - Evaluatable,
    - Always-Invoked, and
    - Tamper-proof.
  - Desire is an architecture that allows the Security Kernel and PCS to share the RESPONSIBILITY of Security with the Application.
- **Extended:**
  - To all inter-partition communication within a group of MILS nodes (*enclave)*

# PCS Requirements

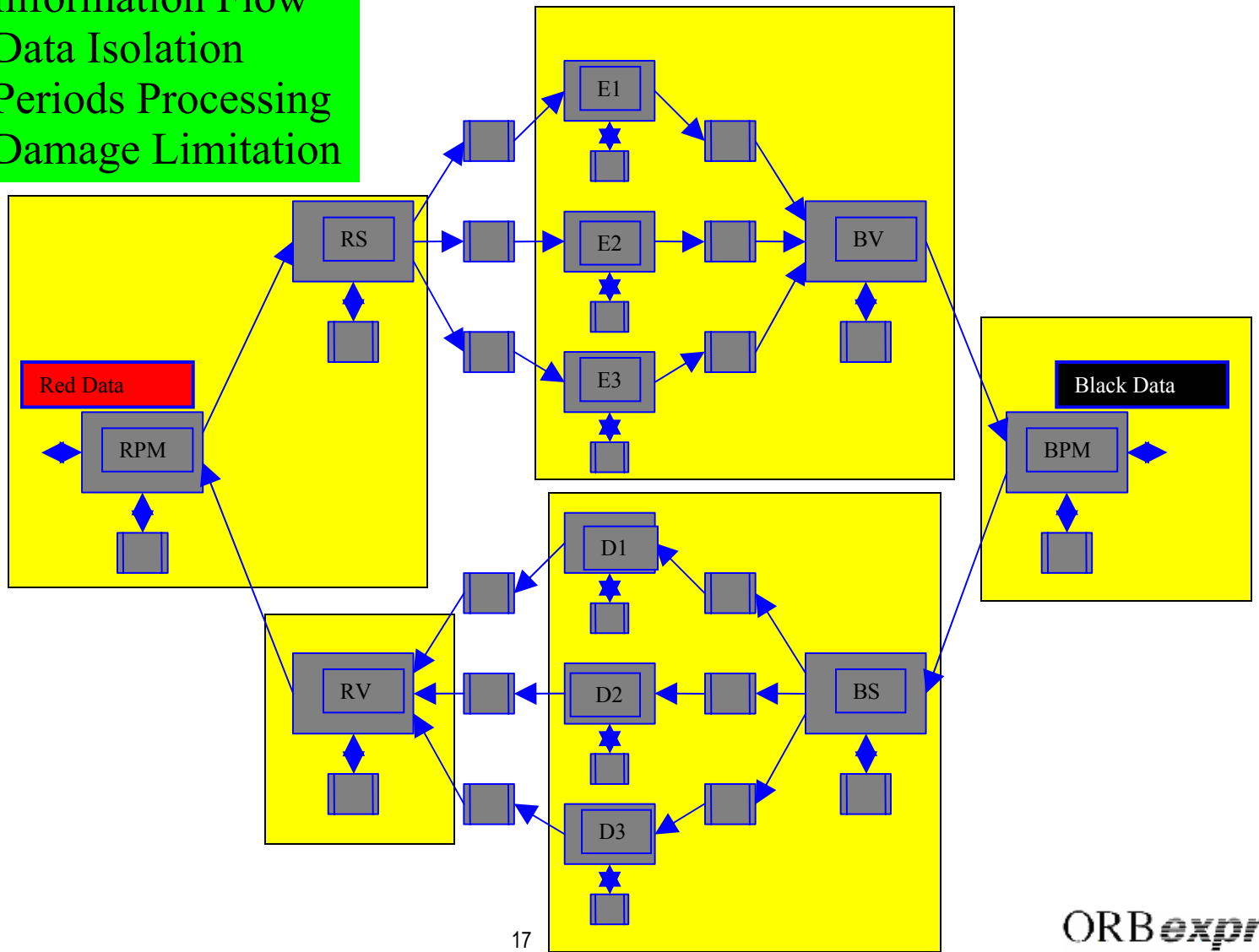
- **PCS must provide**
  - Strong identity of nodes within *enclave*
  - Separation of Levels
    - Need cryptographic separation
  - Separation of Communities of Interest
    - Need cryptographic separation
  - Bandwidth Provisioning & Partitioning
  - Secure Configuration of all Nodes in Enclave
    - Federated information
    - Distributed (compared) vs. Centralized (signed)
  - Secure Clock Synchronization
  - Secure Loading
    - Signed partition images
- **PCS must eliminate**
  - Covert channels
    - Network resources (bandwidth, hardware resources, buffers, …)

# MILS PCS Security Policy Application Example

**Policy Enforcement Independent of Node Boundaries**

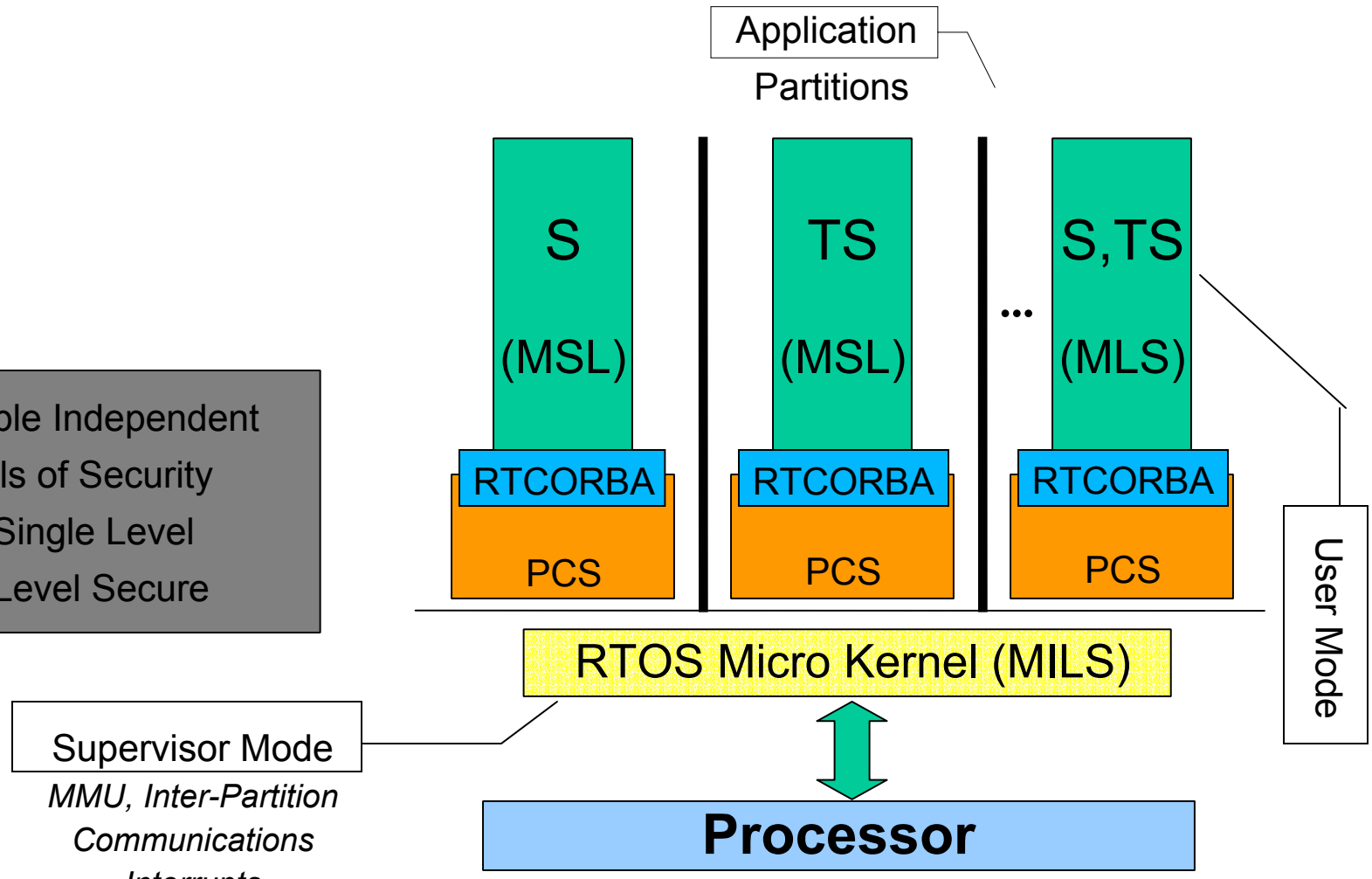
PCS Provides:

- *End-to-end* Information Flow
- *End-to-end* Data Isolation
- *End-to-end* Periods Processing
- *End-to-end* Damage Limitation



# High Assurance MILS Architecture

MILS - Multiple Independent Levels of Security
MSL - Multi Single Level
MLS - Multi Level Secure



# Confines Malicious Code

- With MILS Partitioning Kernel Architecture we know
  - Where inputs came from for each object
  - Where outputs are going for each object
  - Data for each object remains private
- Impossible to TEST security system for all possible user applications
- In addition to testing, high-assurance communication system requires
  - Rigorous design methods
  - Proven software engineering process methods
  - Rigorous use of mathematics


# Real-time MILS CORBA

# Real-Time MILS CORBA

- **Real-time CORBA can take advantage of PCS capabilities**
  - Real-time CORBA + PCS = Real-time MILS CORBA
  - Additional application-level security policies are enforceable because of MILS PK and PCS foundation
- **Real-time MILS CORBA represents a single enabling application infrastructure**
- **Can address the following key cross-cutting system requirements:**
  - High-assurance, MILS-based distributed security (with high-integrity support required for safety critical systems)
  - Real-time (fixed priority as well as dynamically scheduled)
  - High performance distributed object communications (low latency, high bandwidth)

# Security Policies for RT CORBA

- **Unauthorized Discloser of Data - >** Information Flow
  - Critical tasks not bypassed
- **Compromise/Corruption of Sensitive Data->**Data Isolation
  - Data segments not read or corrupted by unauthorized entities
- **Covert Storage Channels - >** Periods Processing
  - ORB is not a covert storage channel on separate messages
- **Presence of Unevaluated Code - >** Damage Limitation
  - Unevaluated code will not compromise processing or data

# Threat Examples

- **T.CONFIG_CORRUPT**
  - A malicious or faulty subject may attempt to modify or corrupt configuration data used by the ORB to enforce information flow policy by accessing the contents of an ORB.
  - A malicious or faulty subject may attempt to bypass the information flow policy enforced by an ORB by using configuration data that, while valid, differs from that being used by another ORB.
- **T.DOS**
  - A malicious or faulty subject may attempt to block other subjects from sending or receiving communications by exhausting or monopolizing shared resources or the resources of another ORB.


# MILS Middleware Realization

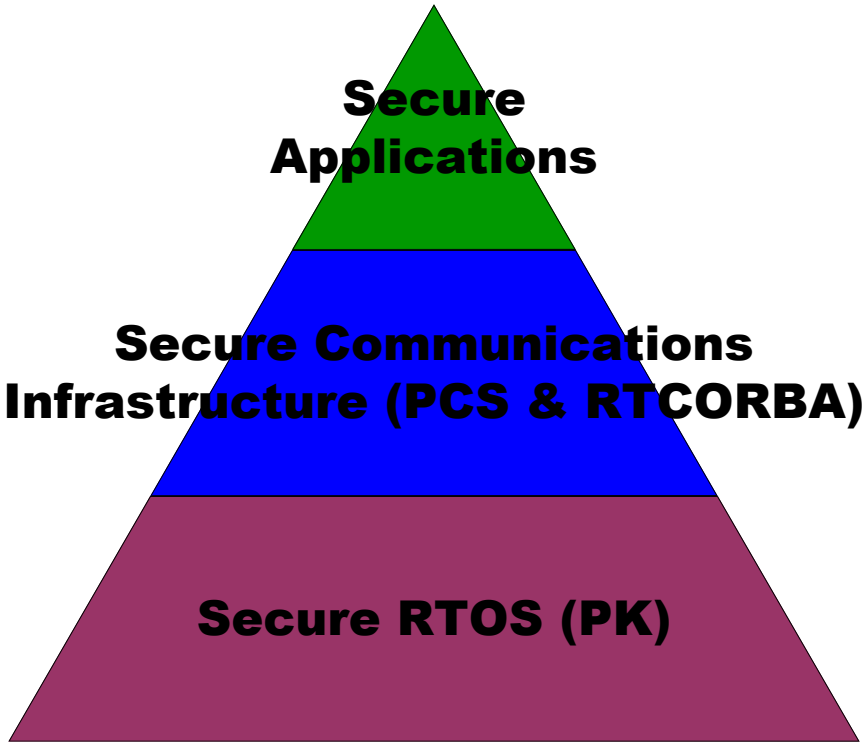
# MILS Partnering

- **Objective Interface is partnered with**
  - Boeing
  - Lockheed-Martin
  - MITRE
  - National Security Agency
  - Rockwell Collins
  - U. S. Air Force
  - University of Idaho

  - To integrate
    - Several MILS security partitioning kernels with
    - A Real-time CORBA middleware implementation, ORB*express* RT

# Business Dependencies

- **Success depends on**
  - Strong business and technical commitment by RTOS vendors
    - So far so great
  - Customer need
    - There's a difference between *wanting* security and *buying* security
    - Performance, size, and predictability are *key*
    - Ease of use is essential
  - Logistics of standards groups
    - Reconciliation of business and technical perspectives



# MILS Implementors

- Hardware Based Kernel
  - AAMP7 — Rockwell-Collins
- Software Based Kernel
  - Integrity-178 — Green Hills Software
  - LynuxOS — LynuxWorks
  - VxWorks AE — Wind River Systems
- Middleware
  - ORB*express* — Objective Interface

# Summary

- **MILS =**
  - High-assurance
  - Multi-level
  - Evaluatable
  - Distributed
- **PCS provides distribution for MILS RTOSes**
- **DoD needs multi-level systems for the war-fighter**
- **Partitioning kernel provides the lowest risk, quickest development time to provide high-assurance MILS systems**
- **MILS + PCS + Real-time CORBA =**
  - A secure deployment platform for distributed, real-time applications