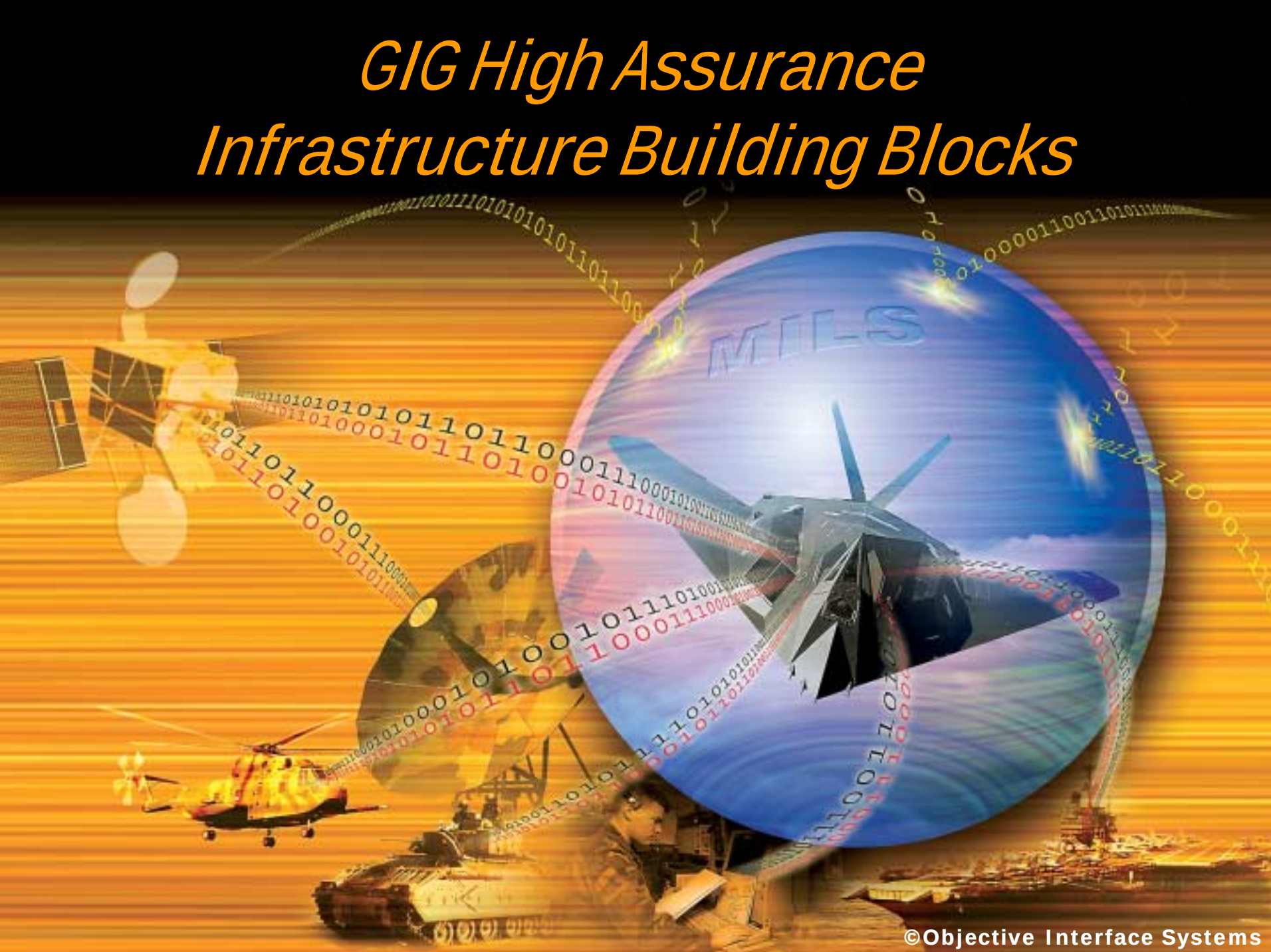
GIG High Assurance Infrastructure Building Blocks



Multiple Independent Levels of Security (MILS)

Gordon Uchenick

Sr. Mentor/Principal Engineer

Objective Interface Systems, Inc.

Acknowledgements

Acknowledgement of significant contributions from (alphabetically):

- Bill Beckwith, Objective Interface Systems
- Dr. Ben Calloni, P.E., Lockheed Martin
- Michael Dransfield, National Security Agency
- Jahn Luke, Air Force Research Laboratory
- W. Mark Vanfleet, National Security Agency

Agenda

- Vision and Benefits
- Security Evolution
- Foundational Threats
- MILS Architecture
  - Separation Kernel
  - Middleware
  - Applications
- Distributed Security
- Partitioning Communications System
- Network Middleware
  - MILS Real-time CORBA
  - MILS Data Distribution Services (DDS)
- Transition to MILS

The MILS Program:

U.S. Air Force Research Laboratory

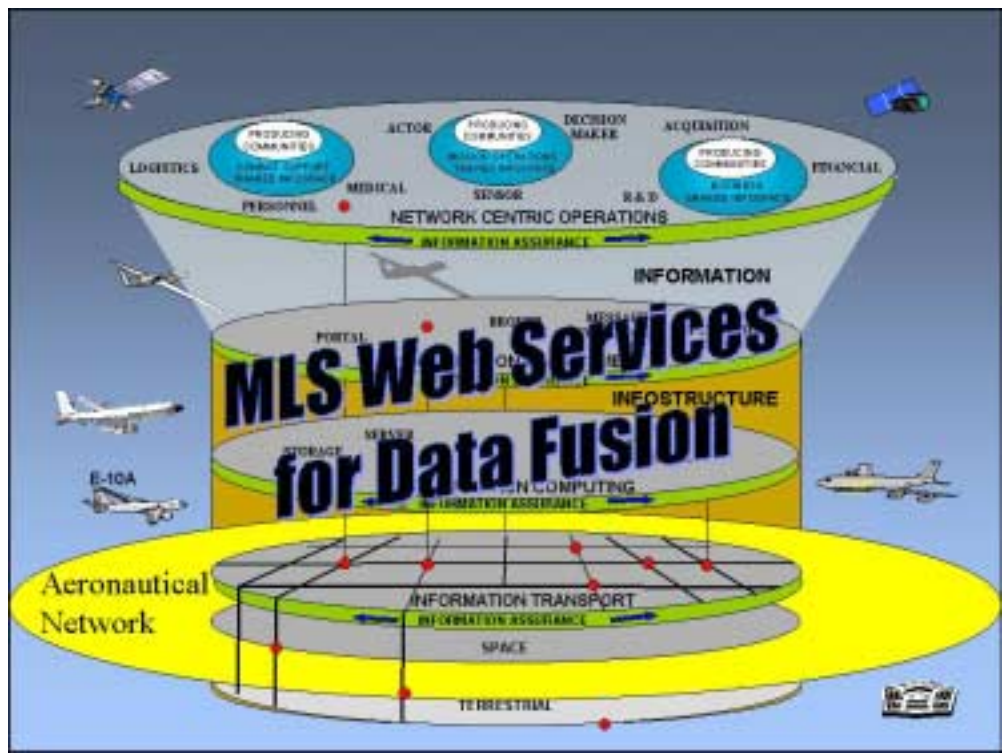
National Security Agency Information Assurance Directorate

Fuse the best from the Safety and Security technologies

- Safety
  - RTCA DO-178B Level A
  - ARINC-653
- Security
  - Common Criteria
  - High Robustness
  - DCID 6/3 Separation

to **ENABLE** provision of MSLS/MLS Computing, Web, and Network Services to

- Weapons Systems
- Communications Facilities
- Command & Control Platforms



- Multiple Independent Levels of Safety/Security: ***MILS***
- Each layer/application can be evaluated separately without impact to the evaluation of the other layers/applications
- High assurance applications
  - Can be developed
  - Can be evaluated
  - Can be maintained
- High assurance applications can become a full partner in enforcing complex Security Policies
- ***Goal: MLS/MSLS capabilities become more practical, achievable and affordable.***

Correspondence Between Safety and Security Certifications

- It is recognized that DO-178B and the Common Criteria have many of the same objectives
  - Does the Software meet its requirements?
  - Is it well designed and well implemented?
  - Has it been thoroughly tested?
  - Is there a process to control modification and maintain assurance?
- Several artifacts for DO-178B and the Common Criteria can be derived from the same documentation efforts (see next slides)
- MILS has many similarities with ARINC-653, *Avionics Application Software Standard Interface,* a standard for Integrated Modular Avionics (IMA)
- This presentation will focus on security because its certification requirements exceed those for safety.

DO-178B	Common Criteria
Software Configuration Management	**Configuration Management**
	Delivery and Operation
Software Development Process	**Development**
	Guidance Documents
Software Planning Process	**Life Cycle Support**
Software Verification Process	**Tests**
	Vulnerability Assesment
Software Quality Assurance	
	Assurance Maintenance

From “Towards Common Criteria Certification for DO-178B Compliant Airborne Software Systems, Executive Summary”, C. Taylor et. al., Center for Secure and Dependable Systems University of Idaho

DO-178B/CC Correspondence Detail 1

CC Class	DO-178B Correspondence					
	Software Configuration Management					Life Cycle Data
ACM – Configuration Management	**Activities**		**Data Control Processees**			
ACM_Aut CM Automation	X					Sect. 11.4
ACM_CAP Advanced support	X					Sect. 11.18
ACM_SCP Development Tools	X					Sect. 11.4, 11.18
ADO – Delivery and Operation						
ADO_Del Prevention Modification	(No correspondence with DO-178B Areas)					
ADO_IGS Installation and Start-up	(No correspondence with DO-178B Areas)					
	Software Development Processes					Life Cycle Data
ADV – Development	**Requ.**	**Design**	**Code**	**Integrate**	**Trace**	
ADV_FSP Functional Specification	X					Sect. 11.6, 11.9, 11.14
ADV_HLD High Level Design		X		X		Sect. 11.7, 11.10, 11.14
ADV_IMP Implementation of TSF			X			Sect. 11.14, 11.11
ADV_INT Minimization Complexity						
ADV_LLD Low Level Design			X			Sect. 11.7, 11.10
ADV_RCR Correspondence Demo Sect.						Sect. 11.14
ADV_SPM Security Policy Model	X					Sect 11.9, 11.14
AGD – Guidance Documents						
AGD_ADM Administrative	(No correspondence with DO-178B Areas)					
AGD_USR User	(No correspondence with DO-178B Areas)					
From “Towards Common Criteria Certification for DO-178B Compliant Airborne Software Systems”, Exec. Sum.						
C. Taylor et. al., Center for Secure and Dependable Systems University of Idaho						

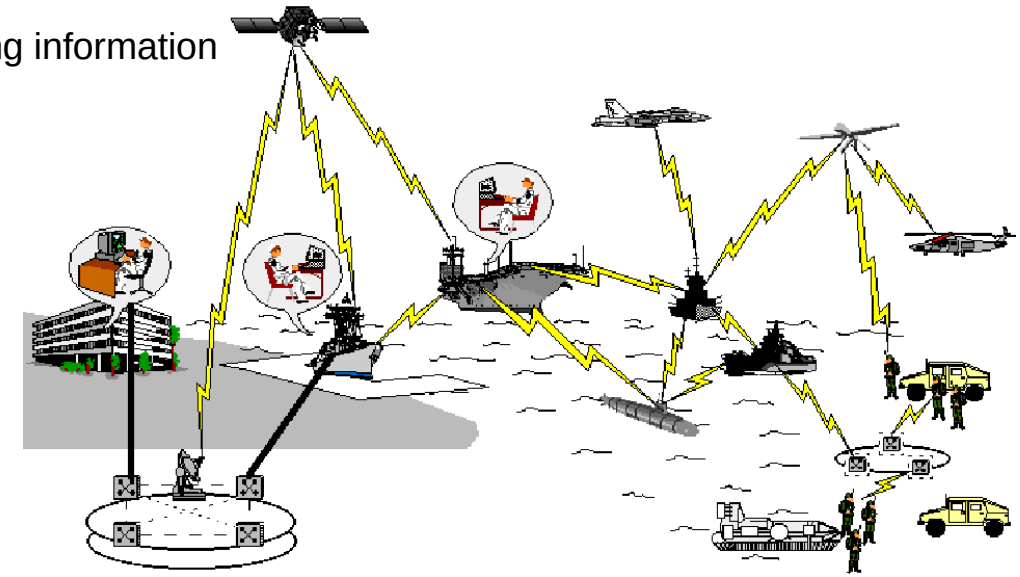
DO-178B/CC Correspondence Detail 2

CC Class		DO-178B Correspondence				
ALC – Life Support	**Activities**	**Plans**	**Life/Cycl /Env**	**Stand.**	**Plans**	
ALC_DVS Sufficiency Security						
ALC_LCD Measurable Model		**X**	**X**			Sect. 11.2
ALC_TAT Compliance Standards			**X**	**X**	**X**	Sect. 11.2
Software Verification Process		**Life Cycle Data**				
ATE – Tests	**Activities**	**Review and Analysis**			**Testing**	
ATE_Cov Coverage			**X**		**X**	
ATE _ DPT Impement. Represent			**X**		**X**	
ATE_FUN Functional Testing					**X**	
ATE_IND Independent Testing					**X**	Sect. 11.3, 11.13, 11.14
AVA – Vulnerability Assessment	(No correspondence with DO-178B Areas)					
AVA_CCA Covert Channel	(No correspondence with DO-178B Areas)					
AVA_MSU Insecure States	(No correspondence with DO-178B Areas)					
AVA_SOF Functional Eval	(No correspondence with DO-178B Areas)					
AVA_VLA Highly Resistant	(No correspondence with DO-178B Areas)					
From “Towards Common Criteria Certification for DO-178B Compliant Airborne Software Systems”, Exec. Sum.						
C. Taylor et. al., Center for Secure and Dependable Systems University of Idaho						

Vision Rationale

Multiple Independent Levels of Security (MILS)

- Modern warfare is all about sharing information
  - Network Centric Warfare
  - System of Systems
  - Global Information Grid
- Information must be shared securely to protect the warfighter and not compromise the mission
- Information is rapidly becoming more diverse
  - Coalition Force Operations
  - Multiple Levels and Communities of Interest
  - Smart Push / Smart Pull / Web Services
- **True MSLS/MLS capability is becoming more important**



MSLS/MLS is Difficult

- Current measures used to handle multilevel data
  - "System High" or "Single Level" operation
  - Physical Separation by Level and Community of Interest
    - Multiple servers in data centers
    - Multiple networks connecting the same endpoints
    - Multiple workstations on a single desk
    - Information sharing via the SneakerNet requiring significant human intervention
- Current MSLS/MLS capabilities
  - Difficult to implement and certify
  - Costly to maintain and reconfigure
  - Problematical to extend and interconnect

Today's Key Takeaway

- High Assurance Systems are needed by the War-Fighter ***and***
  - Home Land Defense
  - Safety Critical World
  - Process Control World
  - Financial World
  - Bio-Medical World

- ***The MILS Separation Kernel architecture provides the lowest risk, quickest development time technology to provide high assurance systems***

Agenda

- Vision and Benefits
- **Security Evolution**
- Foundational Threats
- MILS Architecture
  - Separation Kernel
  - Middleware
  - Applications
- Distributed Security
- Partitioning Communications System
- Network Middleware
  - MILS Real-time CORBA
  - MILS Data Distribution Services (DDS)
- Transition to MILS

Fail-first, Patch-later

- Most commercial computer security architectures
  - The result of systems software where security was an afterthought
    - Operating systems
    - Communications architectures
  - **Reactive** response to problems
    - Viruses, Worms, and Trojan Horses
    - Hackers and Attackers
    - Problems are only addressed ***after*** the damage has been done
- Inappropriate approach for mission critical systems
  - Does not safeguard information or the warfighter
  - **Proactive** measures are required to ***prevent*** damage

Fail-first, Patch-later (cont.)

- **Reactive** approach failures:
  - How many PC anti-virus programs can detect or quarantine malicious device drivers?
    - ***None!***
  - What can an Active-X web download do to your PC?
    - ***Anything!***

Where We've Been: Monolithic Security Kernels

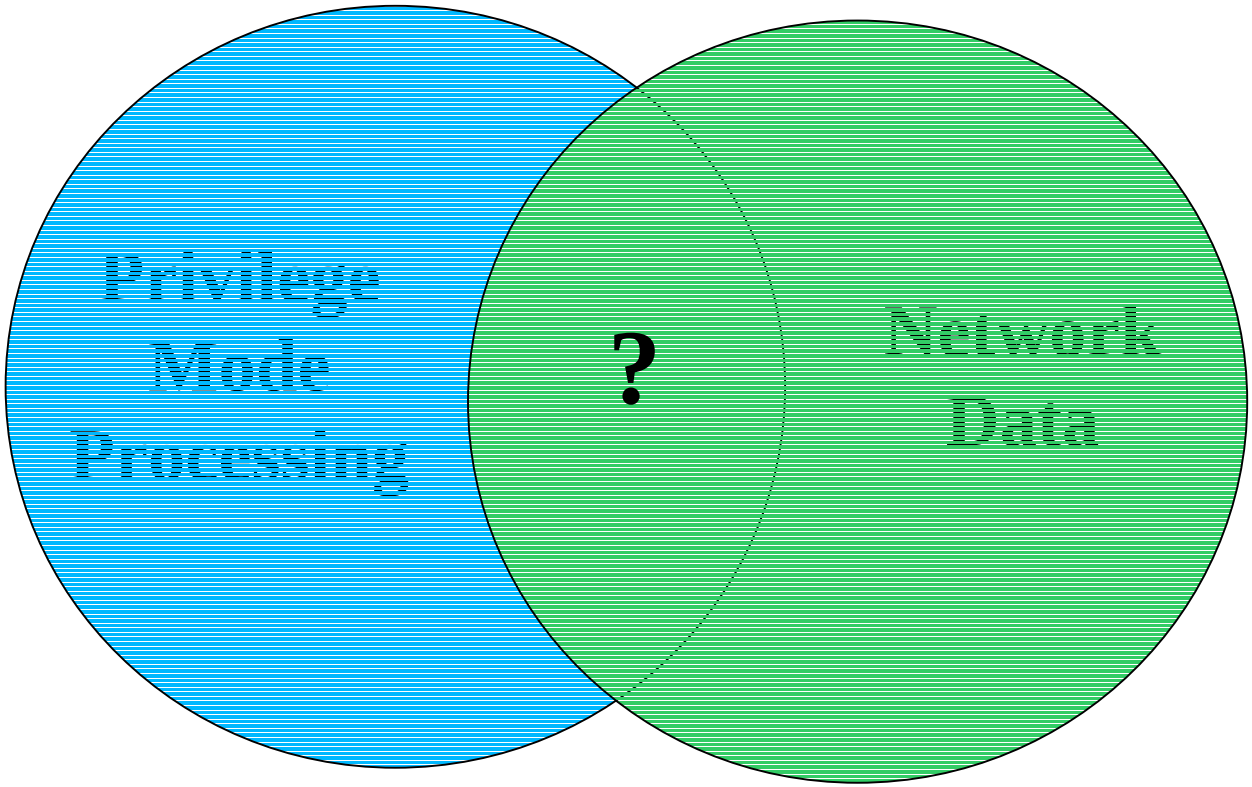
- All security policy enforcement ***was*** performed by the security kernel
  - For performance reasons
  - No other way to insure enforcement was nonbypassable
- As security policy became more complex:
  - Code grew in security kernel
  - Certification efforts become unmanageable
  - Evaluatability of kernel decreased
  - Maintainability of kernel code decreased
  - Policy decisions were based upon incomplete/unauthenticated information

Assurance Level Goals

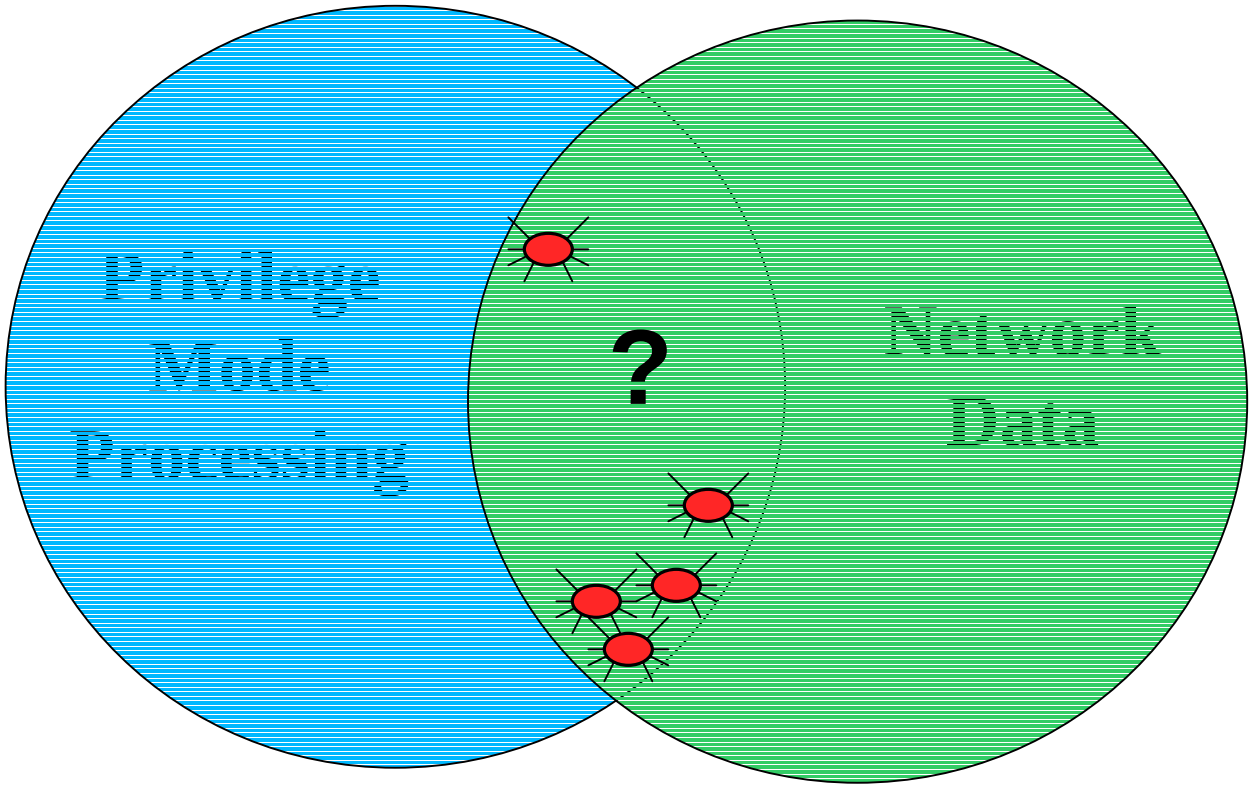
Common Criteria	***MSLS / MLS Separation Accreditation***
Basic Robustness (EAL3)	System High Closed Environment
Medium Robustness (EAL4+)	System High Open Environment
High Robustness (EAL6+)	Multi Level Separation
DCID 6/3 Protection Level 5	***Multi Nation Separation Accreditation***
DO-178B Level A	***Safety Critical***

Monolithic Security Kernel technology is problematical to evaluate above EAL4

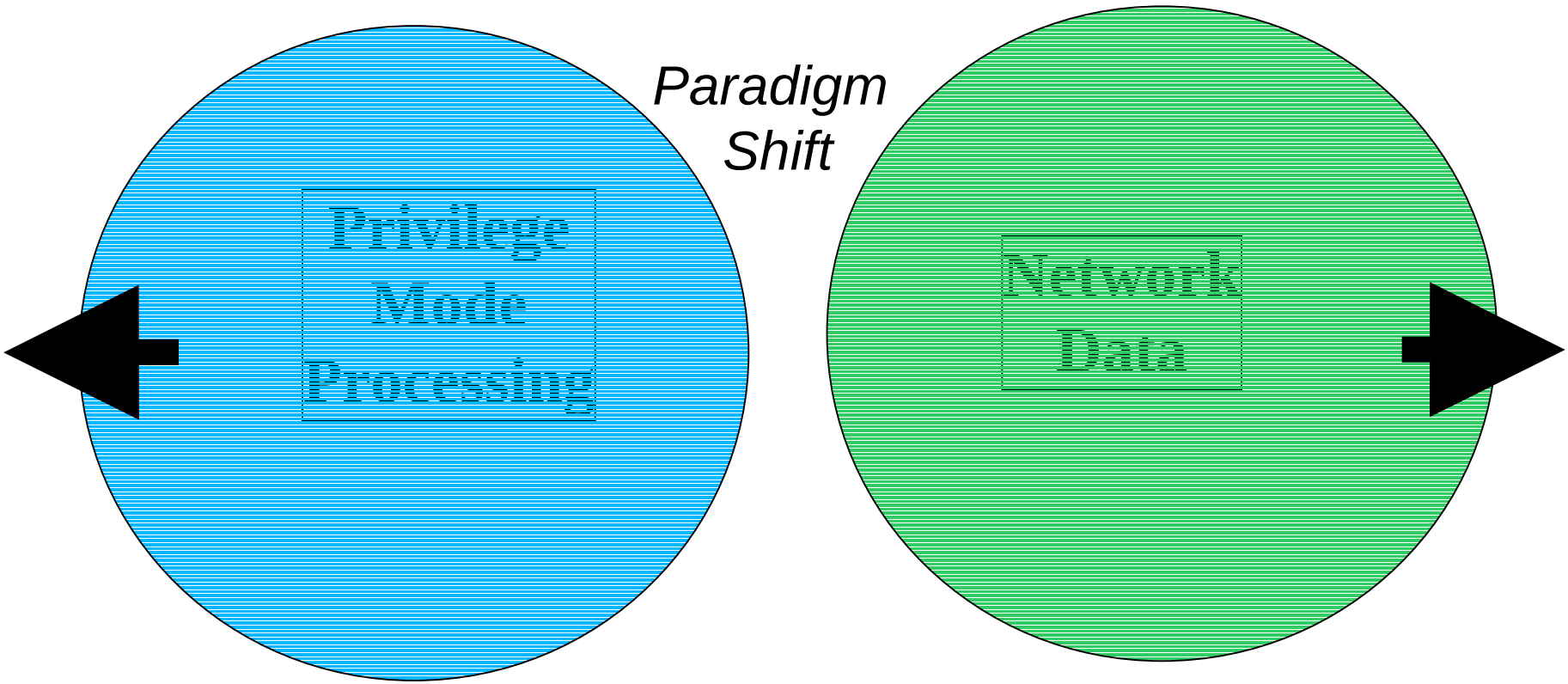
- MILS: Multiple Independent Levels of Security
- Security Kernel is the only privileged code
- Security Kernel enforces only four very simple security policies
- All other security policy enforcement is divided among middleware and the applications
- Enables application layer to enforce its own security policies in a manner that is "**N.E.A.T.**"
  - More about what that means later



What happens when network data is processed in privilege mode?



Wild Creatures of the Net: Worms, Virus, . . .



Under MILS Network Data and Privilege Mode Processing are Separated

- A Year in the Life of a Utility System
  - 100 - 150 hits/day on control network
  - 17 intrusions
  - 2 Denial of Service (DoS) events
  - 3 Loss of Control Events
    - Switchgear controller
    - Boiler Deaerator controls

- SCADA: Supervisory Control And Data Acquisition
- Australian Water Utility
  - Vitek Boden, 48, April 23rd, 2000, Queensland, Australia
    - Disgruntled ex-employee of equipment supplier
    - His vehicle became command center for sewage treatment
    - Controlled 300 SCADA water and sewage nodes
    - "*Was* the central control system" during intrusions
    - Released millions of liters of sewage
    - Killed marine life, blackened creek water, bad stench
  - Caught on 46th attempt
    - Was angling for a consulting job to "fix" the problems he caused
    - Only caught because police thought all the computers in his vehicle might have been stolen
- Result of embedded systems without security

- Vitek Boden was one man working alone, only a low level threat!
- Threat is ranked by assessment of
  - Capability: ***Low to moderate***
  - Resources: ***Low***
  - Motivation: ***Moderate***
  - Risk Willingness: ***Low***
- Higher level threats are organized crime, cyber terrorists, or nation-states.

Agenda

- Vision and Benefits
- Security Evolution
- **Foundational Threats**
- MILS Architecture
  - Separation Kernel
  - Middleware
  - Applications
- Distributed Security
- Partitioning Communications System
- Network Middleware
  - MILS Real-time CORBA
  - MILS Data Distribution Services (DDS)
- Transition to MILS

Foundational Threats

- Software can only be as secure as its foundation
- If the foundation can be successfully attacked, then any system security function that runs on that foundation can easily be rendered ineffective
- Foundational threats include:
  - Bypass
  - Compromise
  - Tamper
  - Cascade
  - Covert Channel
  - Virus
  - Subversion

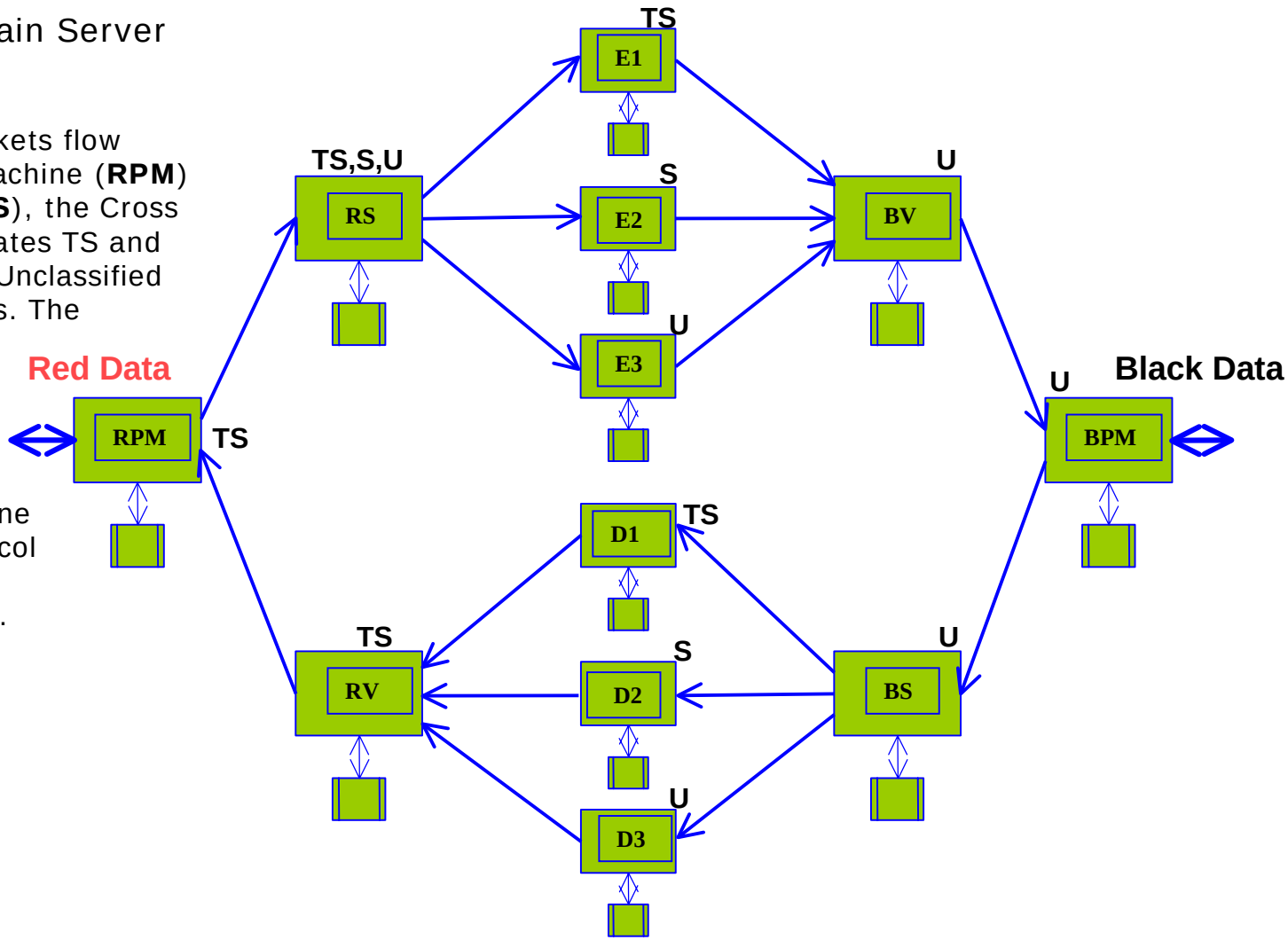
Multilevel Cross Domain Server

Outgoing data:

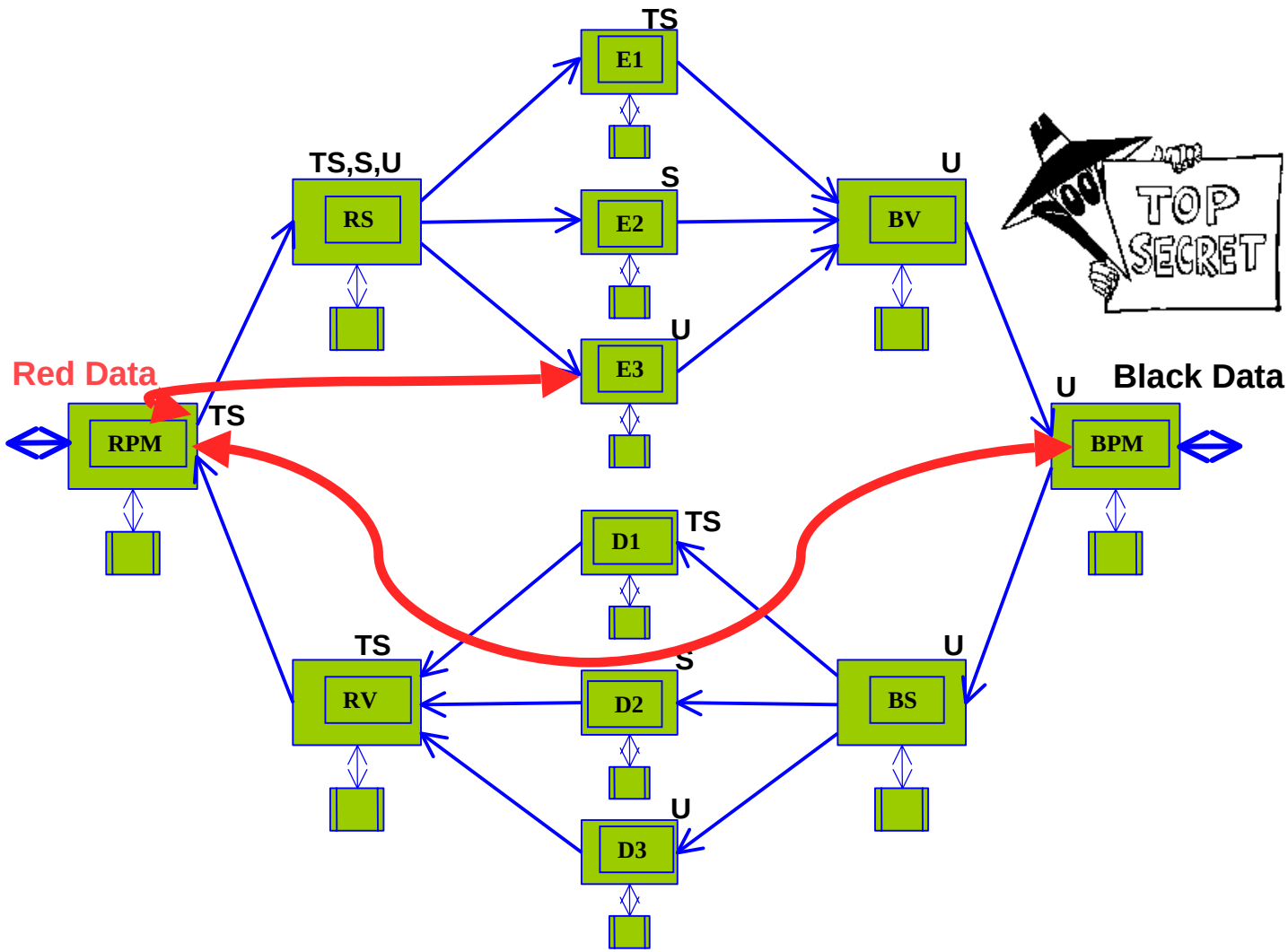
Top Secret cleartext packets flow from the Red Protocol Machine (**RPM**) to the Red Separator (**RS**), the Cross Domain Server, who creates TS and downgraded Secret and Unclassified versions of those packets. The packets are then routed to the appropriate Encryptor, according to level (**E1-E3**). The Black Verifier (**BV**) ensures that this was done properly.The Black Protocol Machine (**BPM**) then transmits the cyphertext.

Incoming data:

Similar to the above, but in the opposite direction.

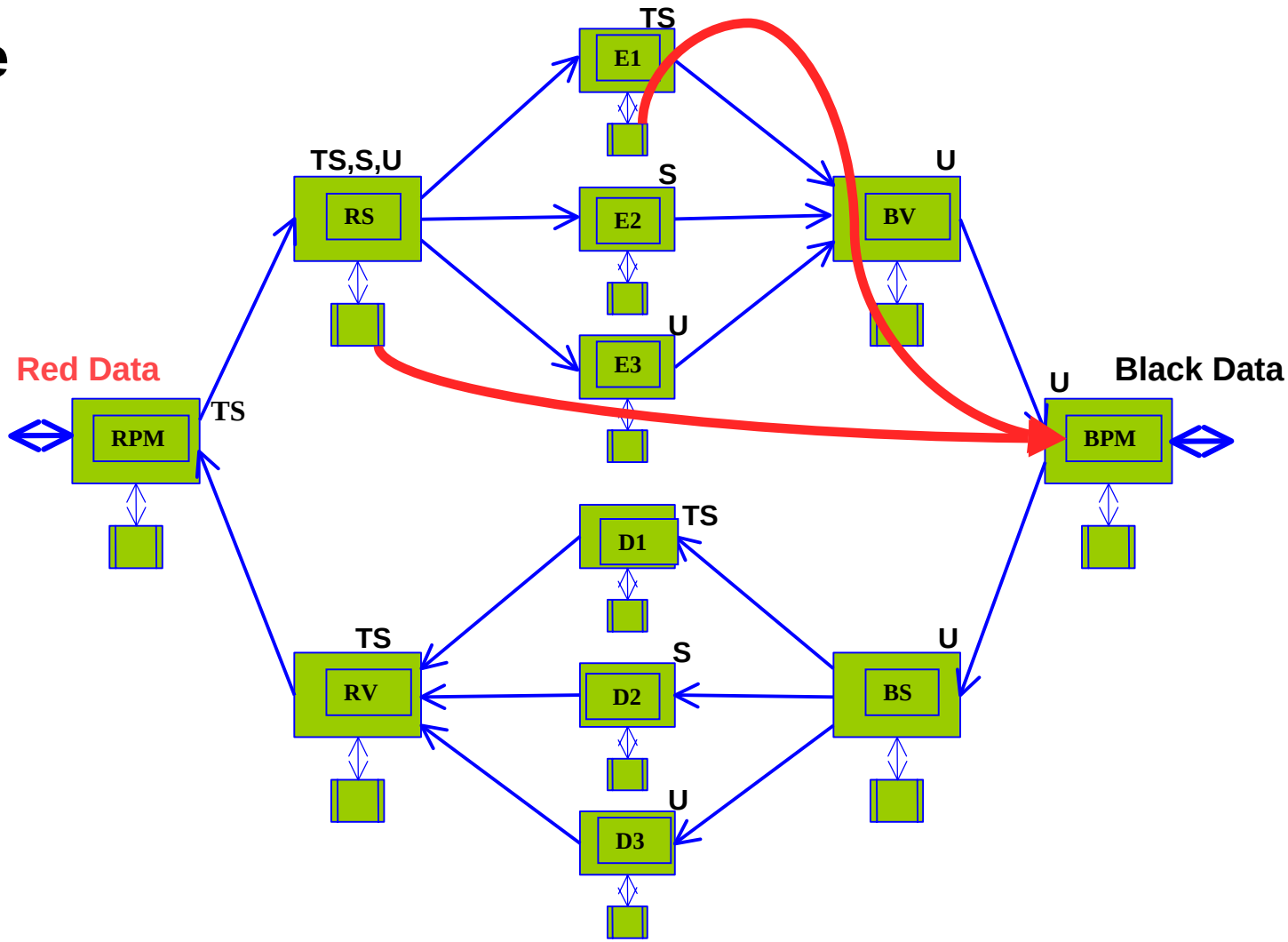


- ✓ **Bypass**
- ✓ Compromise
- ✓ Tamper
- ✓ Cascade
- ✓ Covert Channel
- ✓ Virus
- ✓ Subversion



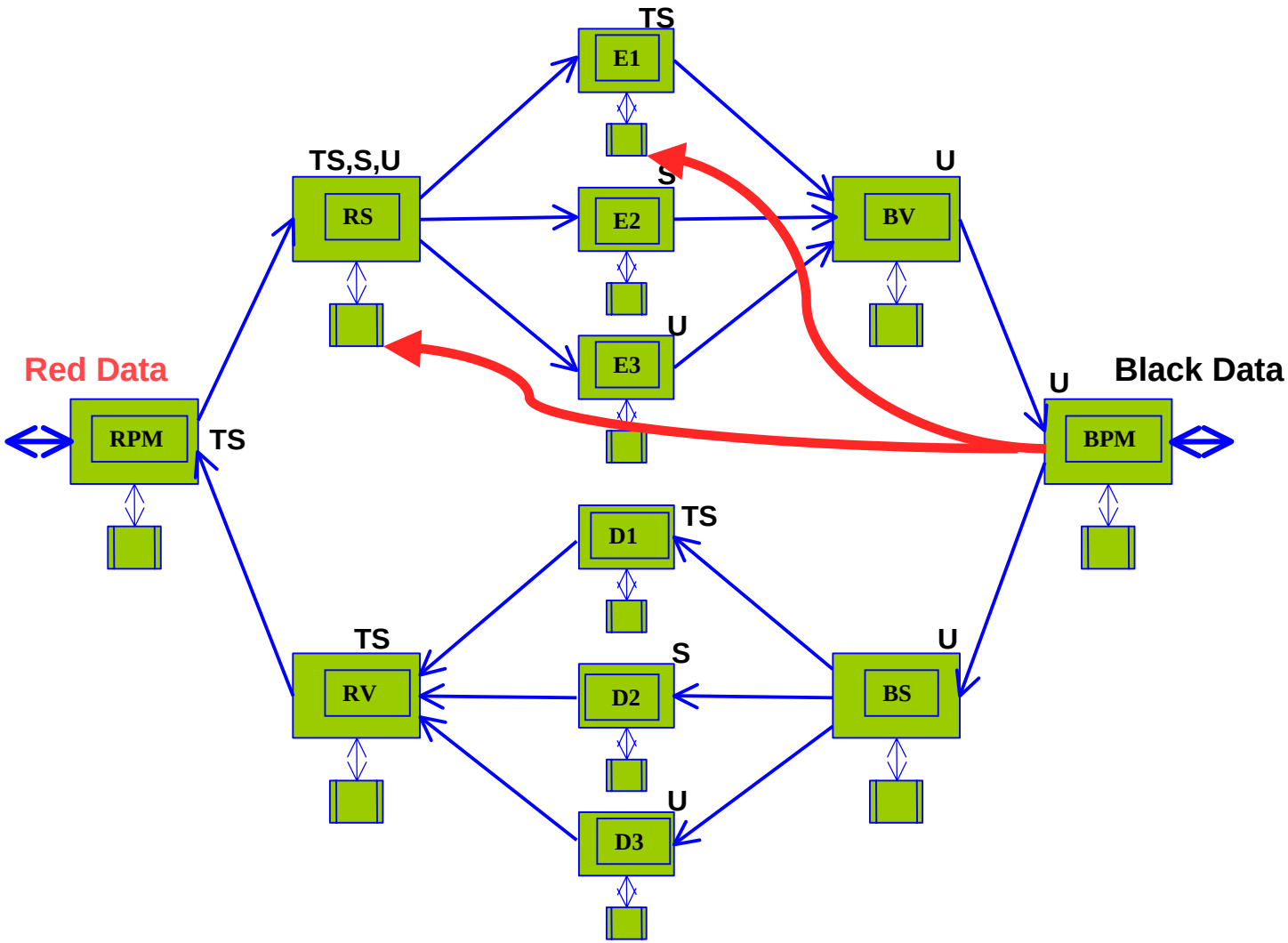
Foundational Threats: Compromise

- ✓ Bypass
- ✓ **Compromise**
- ✓ Tamper
- ✓ Cascade
- ✓ Covert Channel
- ✓ Virus
- ✓ Subversion



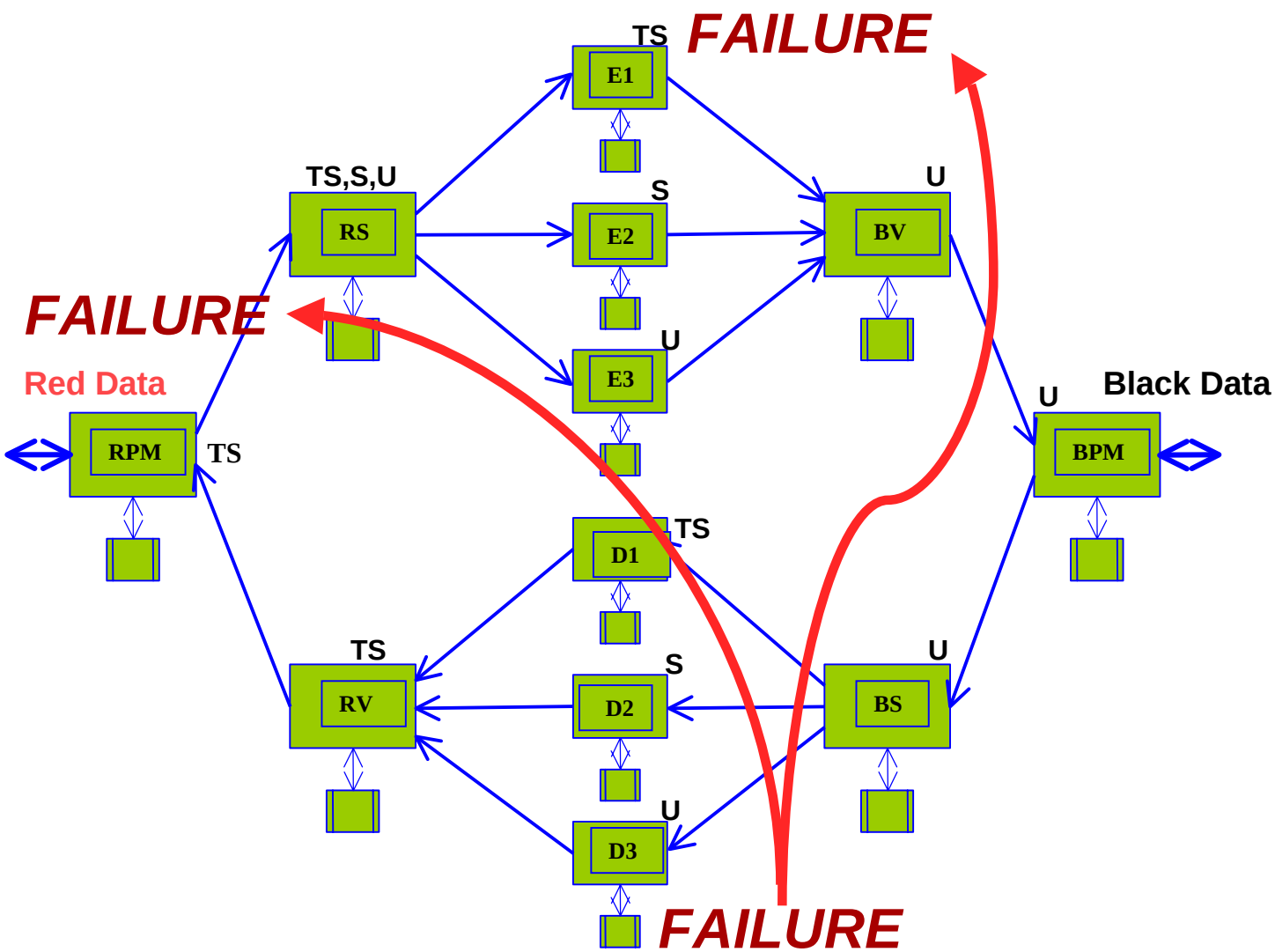
Foundational Threats: Tamper

- ✓ Bypass
- ✓ Compromise
- ✓ **Tamper**
- ✓ Cascade
- ✓ Covert Channel
- ✓ Virus
- ✓ Subversion



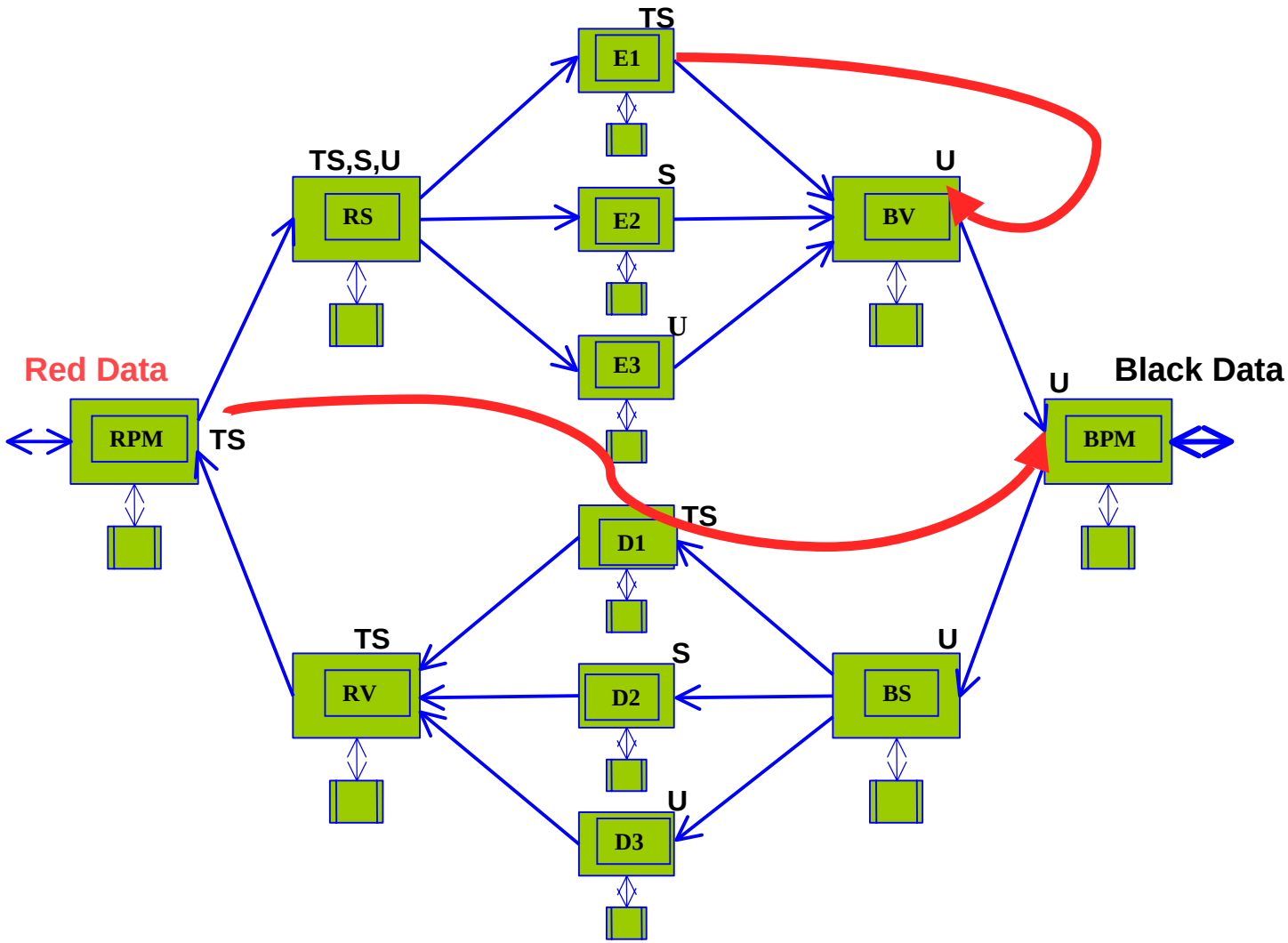
Foundational Threats: Cascade

- ✓ Bypass
- ✓ Compromise
- ✓ Tamper
- ✓ **Cascade**
- ✓ Covert Channel
- ✓ Virus
- ✓ Subversion



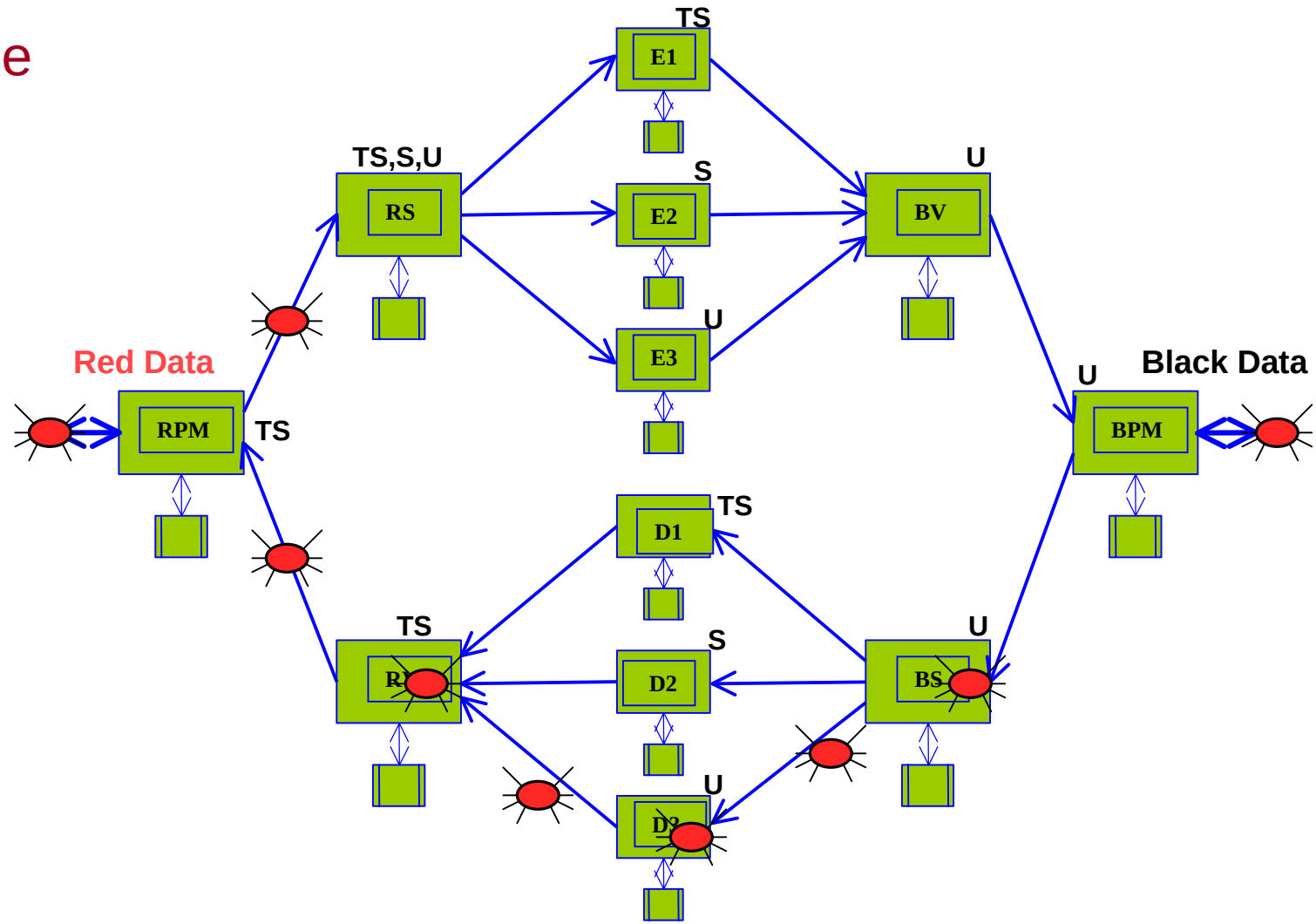
Foundational Threats: Covert Channel

- ✓ Bypass
- ✓ Compromise
- ✓ Tamper
- ✓ Cascade
- ✓ **Covert Channel**
- ✓ Virus
- ✓ Subversion



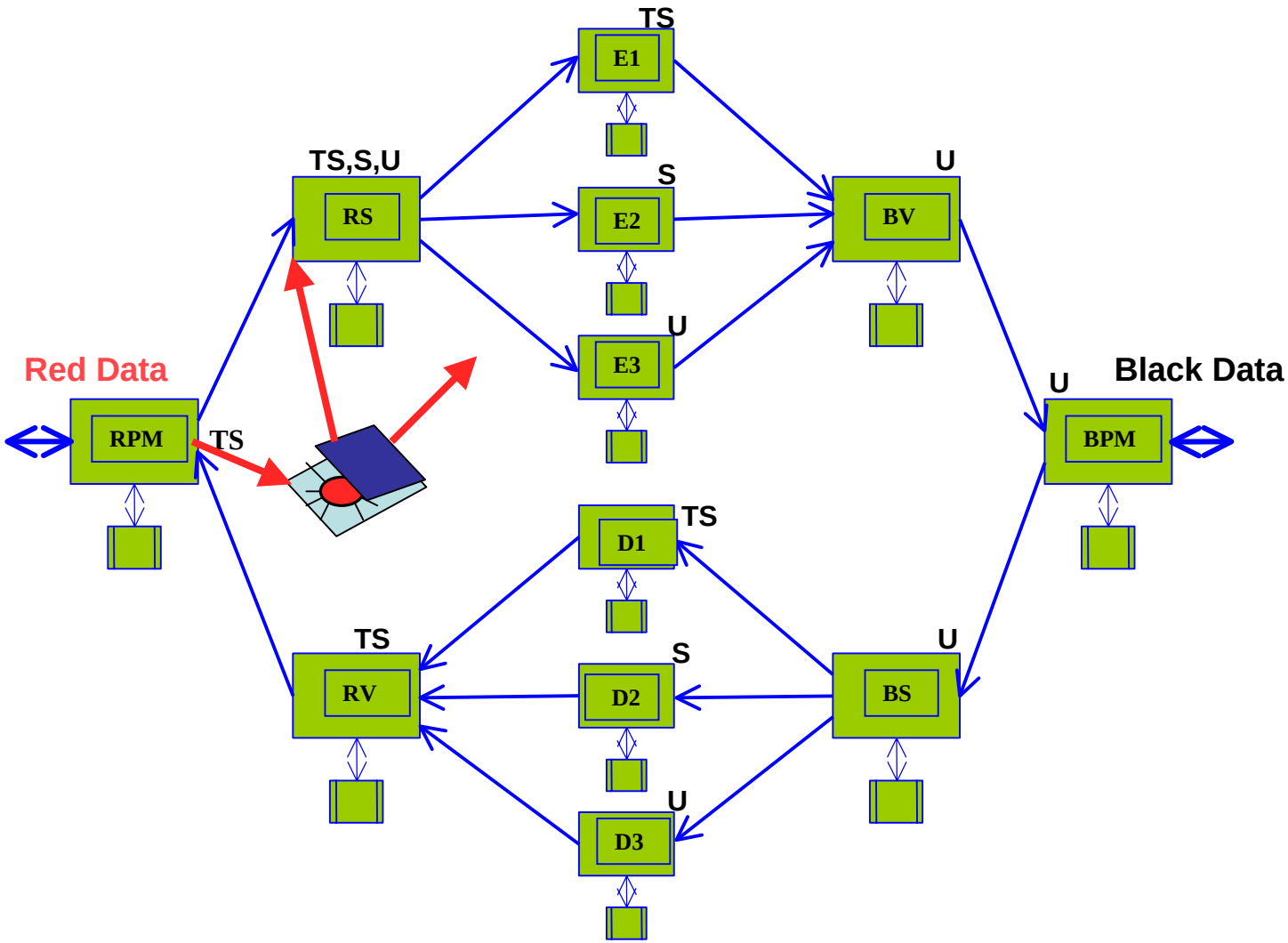
Foundational Threats: Virus

- ✓ Bypass
- ✓ Compromise
- ✓ Tamper
- ✓ Cascade
- ✓ Covert Channel
- ✓ **Virus**
- ✓ Subversion



Foundational Threats: Subversion

- ✓ Bypass
- ✓ Compromise
- ✓ Tamper
- ✓ Cascade
- ✓ Covert Channel
- ✓ Virus
- ✓ **Subversion**



Agenda

- Vision and Benefits
- Security Evolution
- Foundational Threats
- **MILS Architecture**
  - **Separation Kernel**
  - **Middleware**
  - **Applications**
- Distributed Security
- Partitioning Communications System
- Network Middleware
  - MILS Real-time CORBA
  - MILS Data Distribution Services (DDS)
- Transition to MILS

What does MILS do?

Enables **Application Layer Entities** to

Enforce, Manage, and Control

their own

Application Level Security Policies

such that enforcement of the Application Level Security Policies is

Non-bypassable

Evaluatable

Always-Invoked

Tamper-proof

Reference Monitor Concept

The MILS architecture allows the Security Kernel to *SHARE* the responsibility of Security with the Application.

How does MILS achieve its goals?

It Enforces an

Information Flow,

Data Isolation,

Periods Processing, and

Damage Limitation

Security Policy between multiple address spaces:

First, in a **Microprocessor Centric Manner**, i.e., MILS RTOS,

Second, in a **Network Centric Manner**, i.e., MILS Middleware,

in such a manner that the ***layered*** Security Policies are also

Non-bypassable

Evaluatable

Always-Invoked

Tamper-proof

Layered Reference Monitor Concept

N
E
A
T

Separation Kernel & Trusted Middleware must be:

- ***N*on-bypassable**
  - Security functions cannot be circumvented
- ***E*valuatable**
  - Security functions are small enough and simple enough for mathematical verification
- ***A*lways Invoked**
  - Security functions are invoked each and every time
- ***T*amperproof**
  - Subversive code cannot alter the security data or functions

Really very simple:

- Dramatically reduce the amount of *security critical code*

So that we can

- Dramatically increase the scrutiny of *security critical code*

To make

- Development, certification, and accreditation more practical, achievable, and affordable.

Three distinct layers (John Rushby, PhD)

- **Separation Kernel**
  - Separate process spaces (partitions)
  - Secure transfer of control between partitions
  - Really small: 4K lines of code
- **Middleware**
  - Application component creation
  - Provides secure end-to-end inter-object message flow
    - Device Drivers, File Systems, Network Stacks, CORBA, DDS, Attestation, …
- **Applications**
  - Implement application-specific security functions
    - Firewalls, Cryptomod, Guards, Mapplet Engine, CDS, Multi-Nation Web Server, etc.

The MILS Architecture (cont'd)

Separation Kernel

- **Microprocessor Based**
  - Multi-Core Time and Space Multi-Threaded Partitioning
  - Data Isolation
  - Inter-partition Communication
  - Periods Processing
    - Resource Sanitization
  - Minimum Interrupt Servicing
  - Semaphores
    - Multi-Core Synchronization Primitives
  - Timers

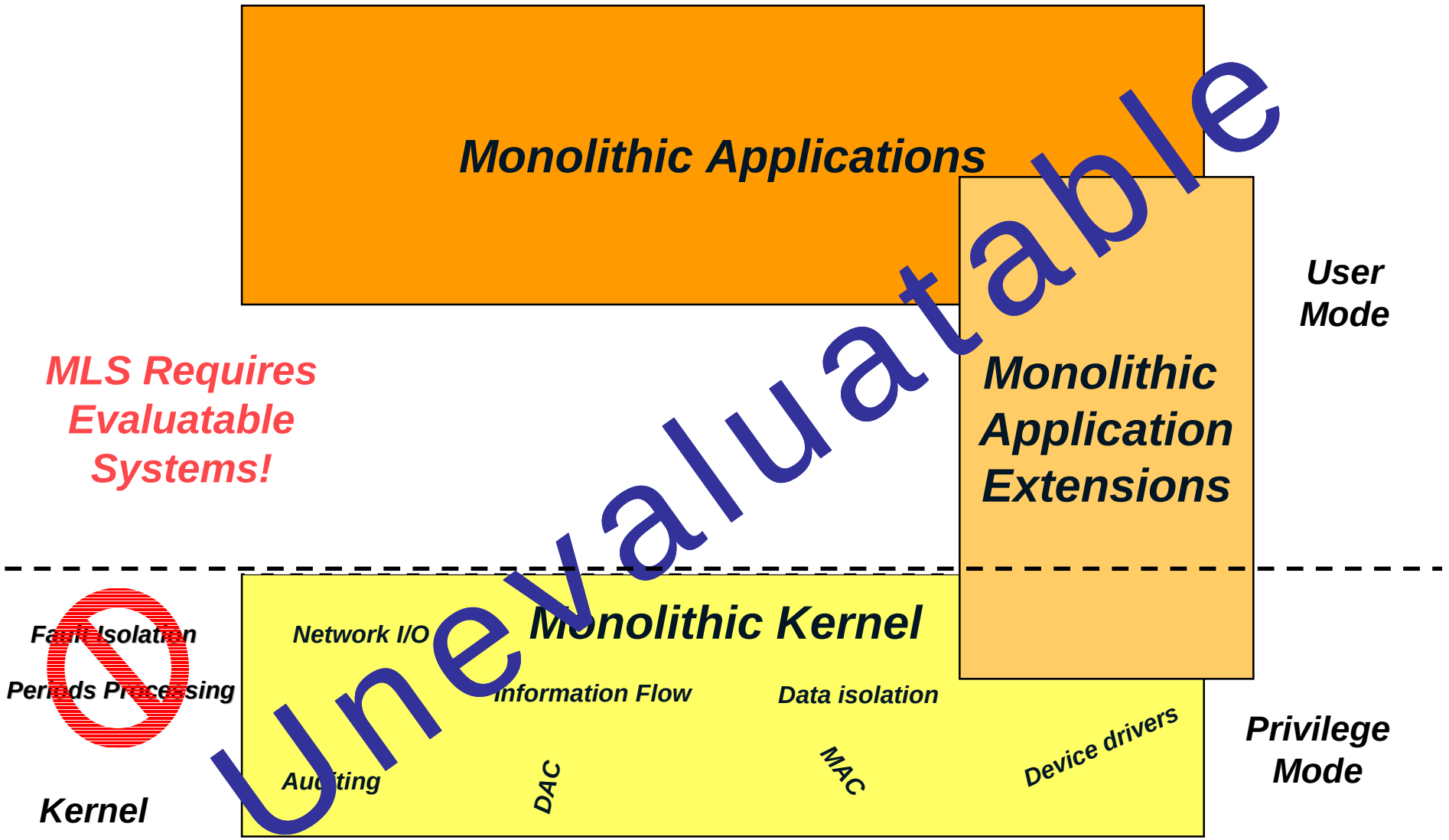
And nothing else!

MILS Middleware

- **Traditional RTOS Services**
  - Device Drivers
  - File Systems
  - Token and Trusted Path
- **Traditional Middleware**
  - CORBA (Distributed Objects)
  - Data Distribution (Pub-Sub)
  - Web Services
- **Partitioning Communication System (PCS)**
  - Global Enclave Partition Comm
    - TCP, UDP, Rapid-IO, Firewire, …
  - Partition Based Attestation

Where We've Been: Starting Point for Architectural Evolution

Multiple Independent Levels of Security (MILS)

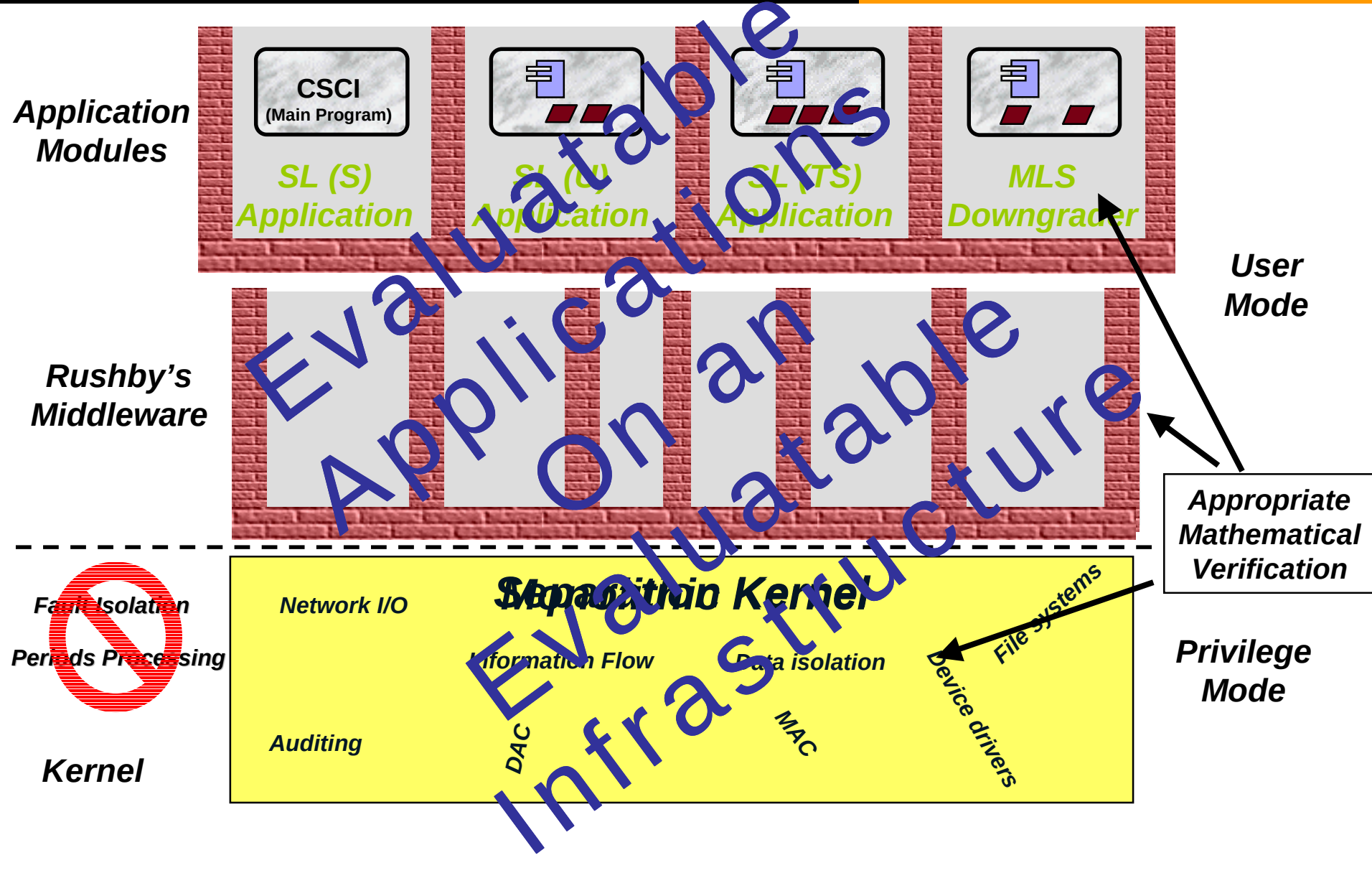


Separation Kernel

- Where should SK reside?
  - To be tamper-proof
    - Must be in a separate address space from ***any*** application code
  - To be non-bypassable
    - Must be part of every input or output service request issued by an application
- Why keep security functions out of the kernel?
  - Security functions are often application-specific
  - Any code co-resident with security functions could interfere with those security functions
  - Entire kernel must be analyzed for weaknesses and malicious code
- ***The SK must be the only code that runs in privileged mode***

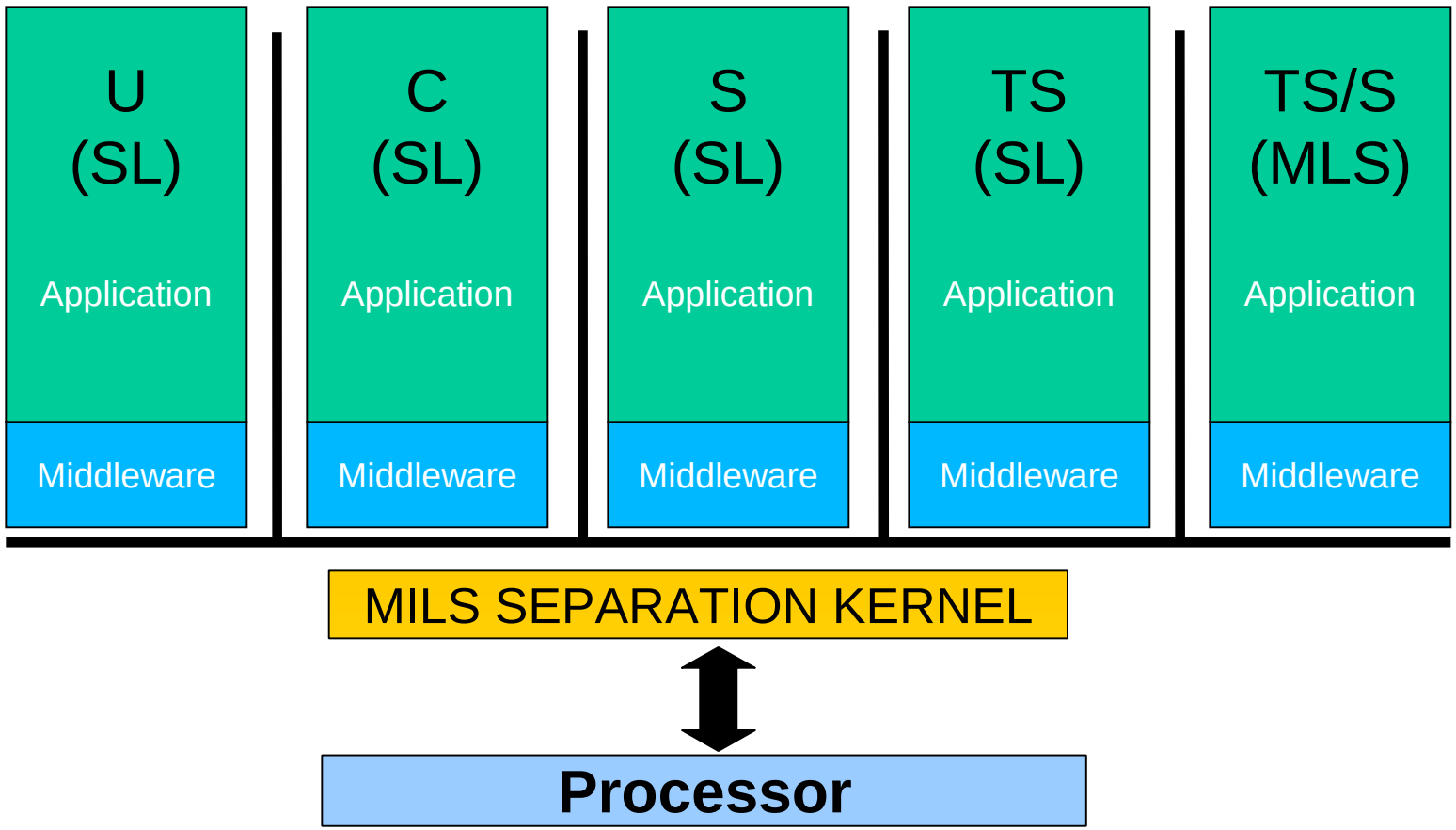
MILS Architecture Evolution

Multiple Independent Levels of Security (MILS)



The MILS Architecture

Multiple Independent Levels of Security (MILS)



- MILS makes mathematical verification of the core systems and communications software possible by reducing the security functionality to four key security policies
  - Information Flow … Policy
  - Data Isolation … Policy
  - Periods Processing … Policy
  - Damage Limitation … Policy

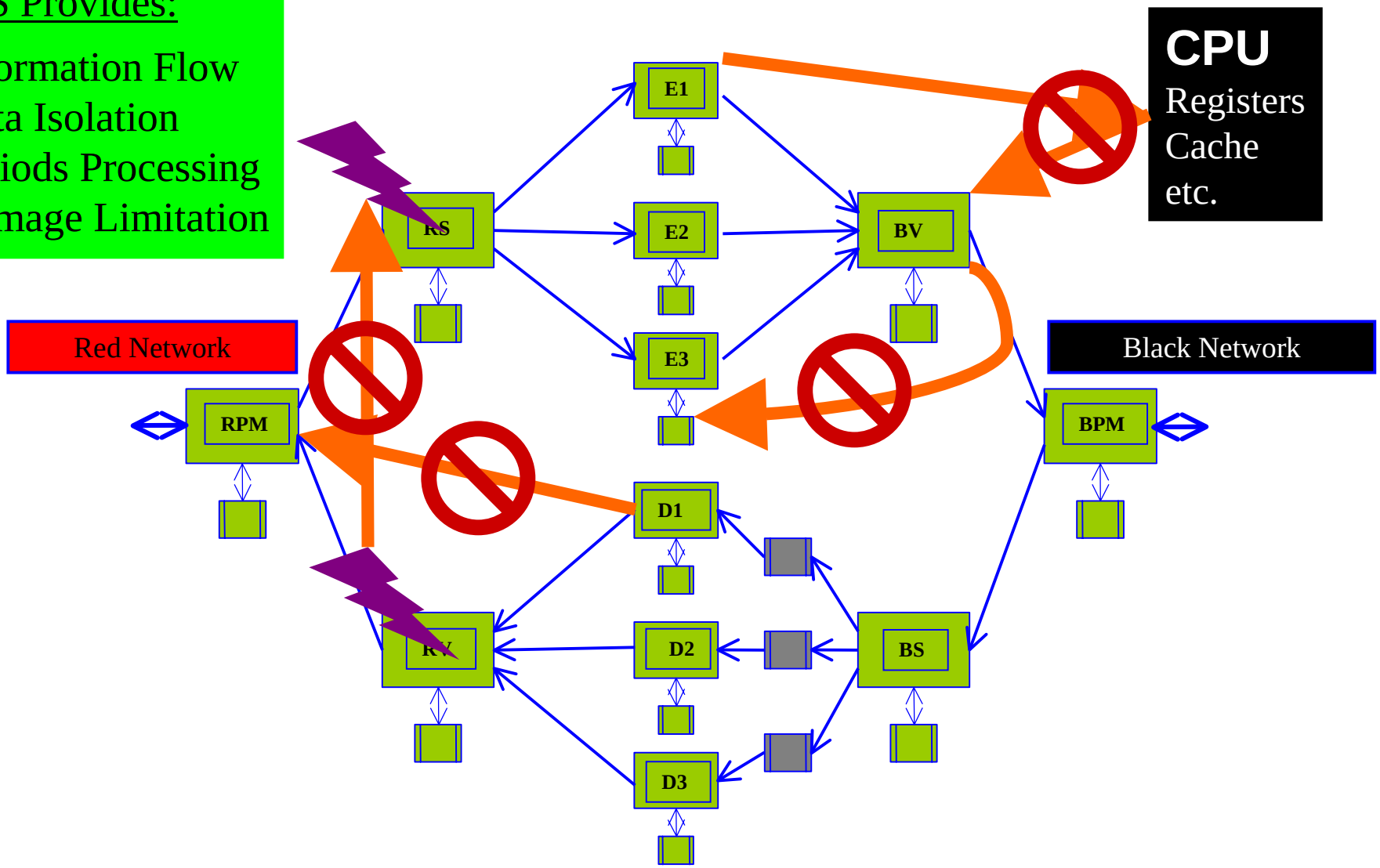
MILS Security Policies

- **Information Flow**
  - Information originates only from authorized sources
  - Information is delivered only to intended recipients
  - Source of Information is authenticated to recipient
- **Data Isolation**
  - Information in a partition is accessible only by that partition
  - Private data remains private
- **Periods Processing**
  - The microprocessor itself will not leak information from one partition to another as it switches from partition to partition
- **Damage Limitation**
  - A failure in one partition will not cascade to another partition
  - Failures will be detected, contained, & recovered from locally

MILS Security Policy Example

MILS Provides:

- Information Flow
- Data Isolation
- Periods Processing
- Damage Limitation

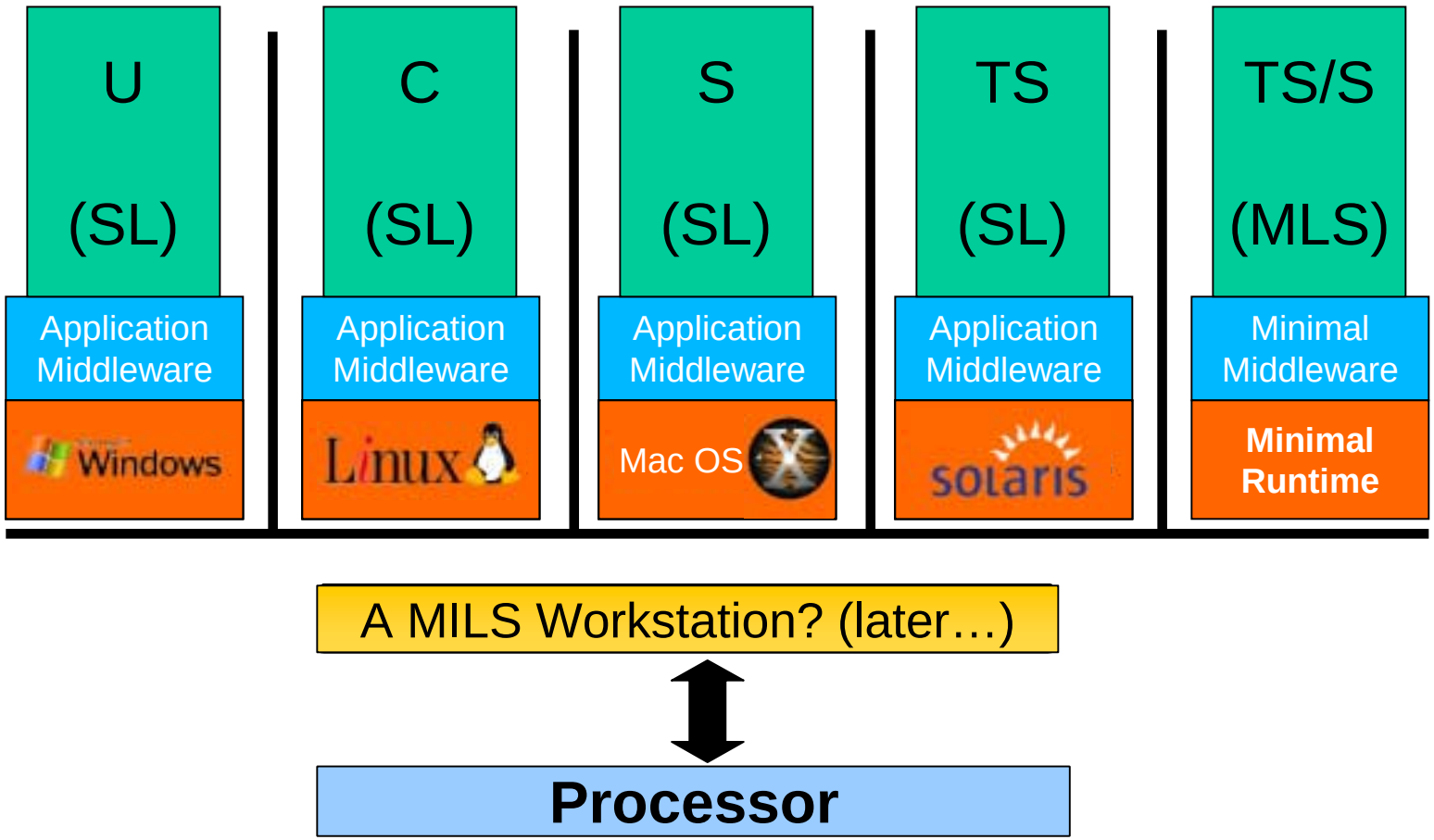


Guest OS Architecture

- Traditional Embedded OS/RTOS can run in a user mode MILS partition as a “Guest Operating System”
- Each Guest OS’s Hardware Abstraction Layer (HAL) “sees” the Separation Kernel as its hardware environment
- Effectively, a single real microprocessor supporting several virtual microprocessors, all robustly separated in time and in space
  - With tightly controlled facilities for inter-partition communications
- Advantages
  - Protect investment in existing code bases
  - Familiar API and environment for new application development
  - Both of the above in enhancement of a legacy system
  - Enable unit testing on commodity hardware

Guest OS Architecture

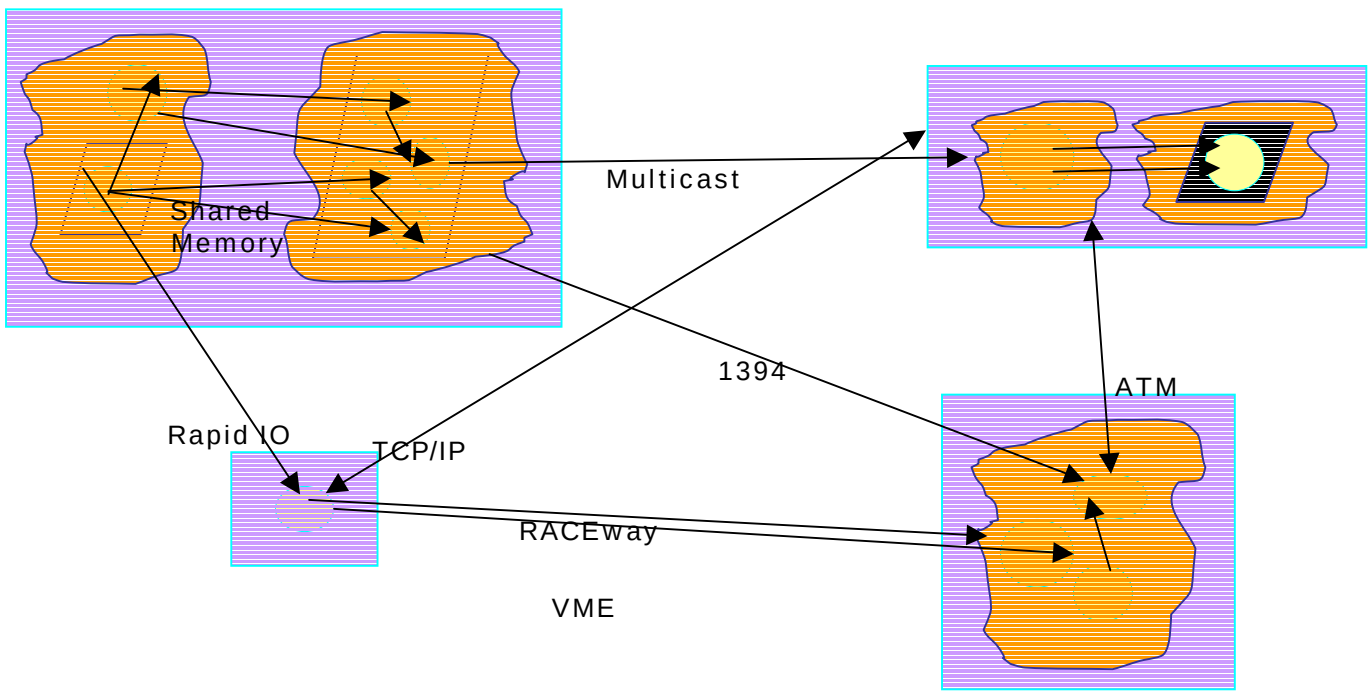
Multiple Independent Levels of Security (MILS)



Agenda

- Vision and Benefits
- Security Evolution
- Foundational Threats
- MILS Architecture
  - Separation Kernel
  - Middleware
  - Applications
- **Distributed Security**
- Partitioning Communications System
- Network Middleware
  - MILS Real-time CORBA
  - MILS Data Distribution Services (DDS)
- Transition to MILS

- Partition Local – same address space, same machine
- Machine Local – different address space, same machine
- Remote – different address space, on a different machine



Distributed Security Requirements

- Extend single node security policies to multiple nodes
  - Information Flow
  - Data Isolation
  - Periods Processing
  - Damage Limitation
- Do not add new threats to data Confidentiality or Integrity
- Enable distributed Reference Monitors to be **NEAT**
- Optimal inter-node communication
  - Minimizing added latency (first byte)
  - Minimizing bandwidth reduction (per byte)
- Fault tolerance
  - Security infrastructure must have no single point of failure
  - Security infrastructure must support fault tolerant applications

Agenda

- Vision and Benefits
- Security Evolution
- Foundational Threats
- MILS Architecture
  - Separation Kernel
  - Middleware
  - Applications
- Distributed Security
- **Partitioning Communications System**
- Network Middleware
  - MILS Real-time CORBA
  - MILS Data Distribution Services (DDS)
- Transition to MILS

Partitioning Communication System

- Extend MILS partitioning kernel protection to multiple nodes
- Part of MILS Middleware
- Responsible for all communication between MILS nodes
- Similar philosophy to MILS Separation Kernel
  - Minimalist: only what is 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 High Robustness (EAL6+) evaluation

PCS Objective

- **Just like MILS Separation Kernel:**
  - Enable the Application Layer Entities to
    - Enforce, Manage, and Control
  - Application Level
    - Security Policies
  - in such a manner that the Application Level Security Policies are
    - **N**on-Bypassable,
    - **E**valuatable,
    - **A**lways-Invoked, and
    - **T**amper-proof.
  - 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 Specific Requirements

- Strong Identity
  - Nodes within enclave
- Secure Configuration of all Nodes in Enclave
  - Federated information
  - Distributed (compared) vs. Centralized (signed)
- Separation of Levels/Communities of Interest
  - Need cryptographic separation
- Bandwidth provisioning & partitioning
  - Network resources: bandwidth, hardware resources, buffers
- Secure Loading: signed partition images
- Secure Clock Synchronization
- Suppression of Covert Channels

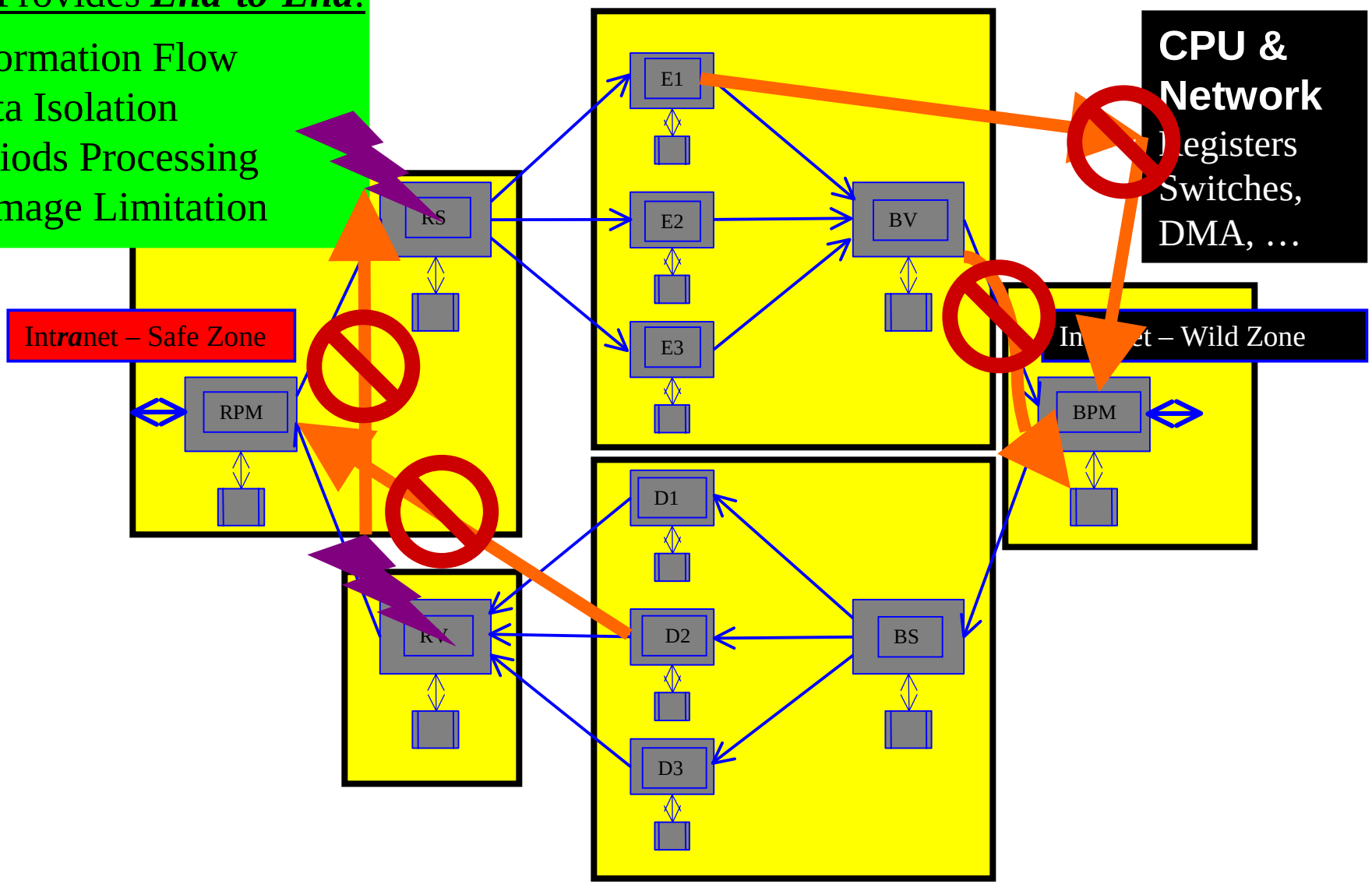
MILS Security Policy Example: Distributed Internet Firewall

Multiple Independent Levels of Security (MILS)

Policy Enforcement Independent of Node Boundaries

PCS Provides ***End-to-End***:

- Information Flow
- Data Isolation
- Periods Processing
- Damage Limitation



What the PCS Is *and* Is Not

- The PCS *is*
  - Like a super VPN configured between partitions in distributed nodes
  - Adds techniques for covert storage and time channel suppression

- The PCS *is not*
  - Application middleware like CORBA, DDS, or Web Services
  - A Guard or Application Firewall
    - Doesn't examine message content
    - Can't enforce security policies delegated to the application layer
  - A total, end-to-end security solution
    - Foundation for application level security
    - ***Not a replacement for*** application level security

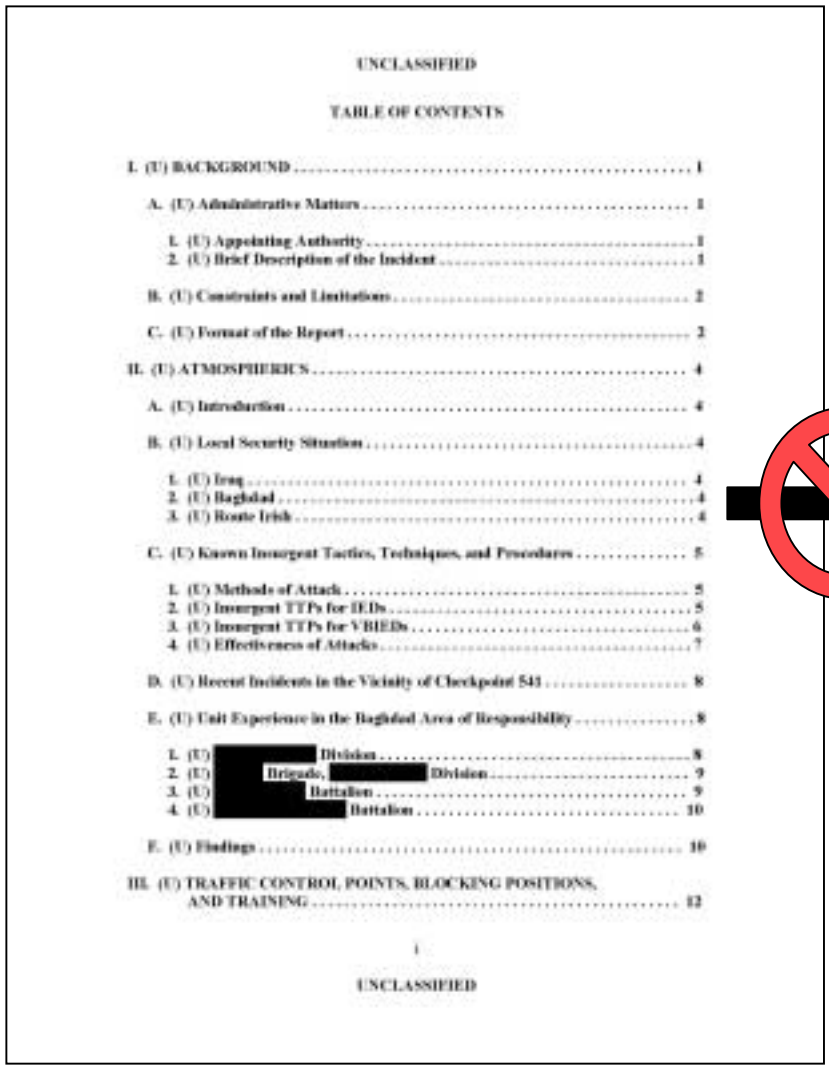
Not an Access Guard!

- Identity Based Access Control
- Protocol Specific Access Control
  - CORBA/GIOP (Client/Server) Access Guard
    - Determines if query is allowed based on method name, parameter values, security levels of client/server
    - Determines if response is expected
    - Error Message Response Policy
  - DDS (Publish/Subscribe) Access Guard
    - Determines if subscriber allowed to connect/receive from a particular label based on identity and security levels of label and subscriber
    - Determines if publisher allowed to connect/publish to a particular label based on identity and security levels of label and publisher
  - HTTP (Web) Access Guard

Not a Content Guard!

- Document Type Specific Guarding, including…
  - .doc
  - .pdf
  - .xml
  - .html
  - .tex
  - .ppt
  - .jpg
  - .avi
  - .mp3
  - .dvi
  - .xls
  - .mpeg
  - .mov
  - .ps/eps
  - .rtf
- Verify no Deleted Data in Document
- Verify no Hidden Data under Overlay
- No Non-displayed Annotation or Comments
- Verify Release Markings
- “Dirty” Word Search

- Italian Shooting Final Report (.pdf Guarding Failure)

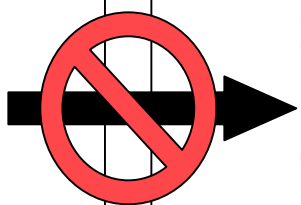


UNCLASSIFIED

TABLE OF CONTENTS

i

UNCLASSIFIED



UNCLASSIFIED

TABLE OF CONTENTS

i

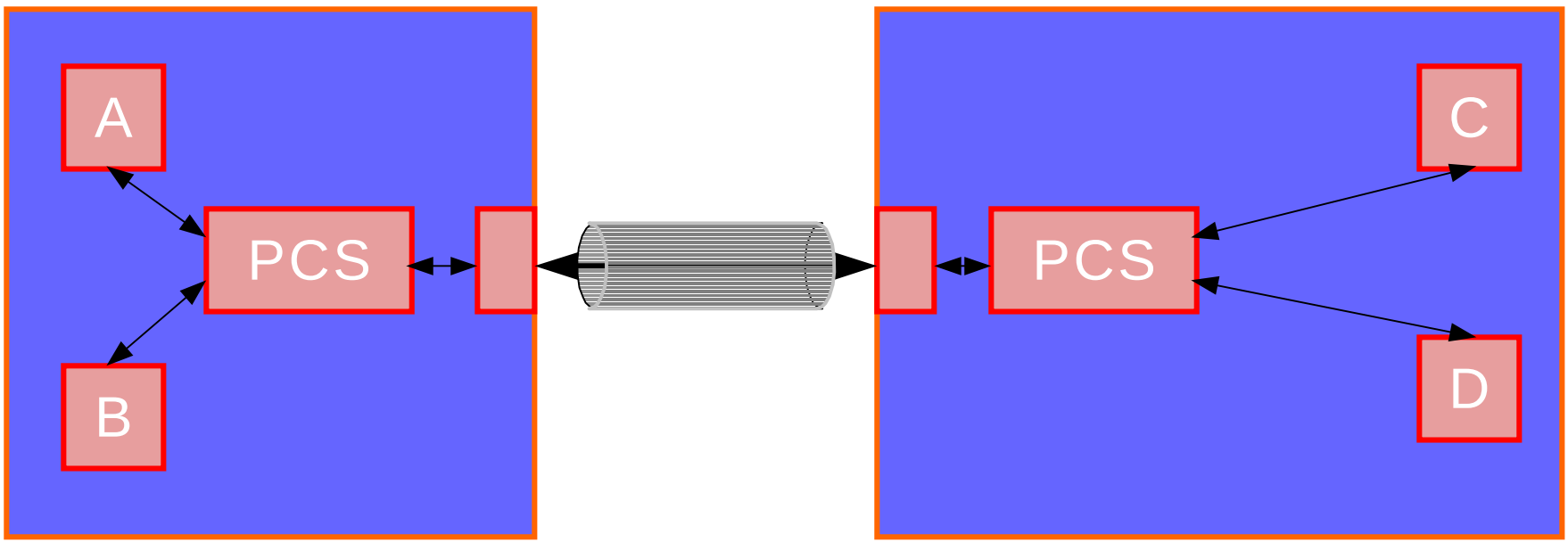
UNCLASSIFIED

PCS is Trusted Plumbing

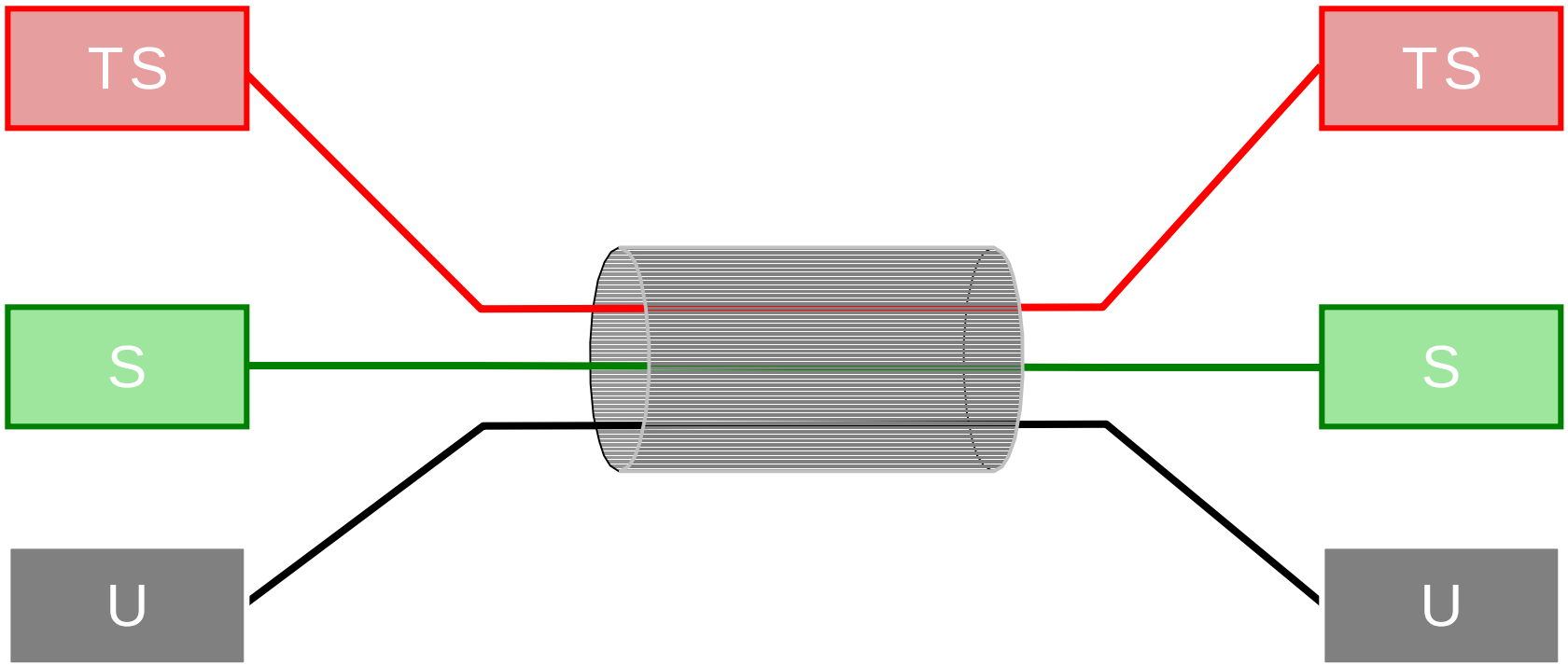
- PCS assumes the network can't be trusted
  - Leverage COTS stacks, NICs, media, switches, and routers
- PCS provides trusted data flow among distributed applications and guards
  - Code that was typically duplicated from partition to partition
- Access guards and data guards can be tightly focused on the data owner's specific requirements
- Trusted data flow enables higher assurance
  - Smaller code body
  - Simpler logic
  - Formal methods more practical

Where a PCS fits in MILS

- PCS is communications middleware for MILS
- Always interposed in inter-node communications
- Interposed in some intra-node communications also
- Parallels Separation Kernel's policies

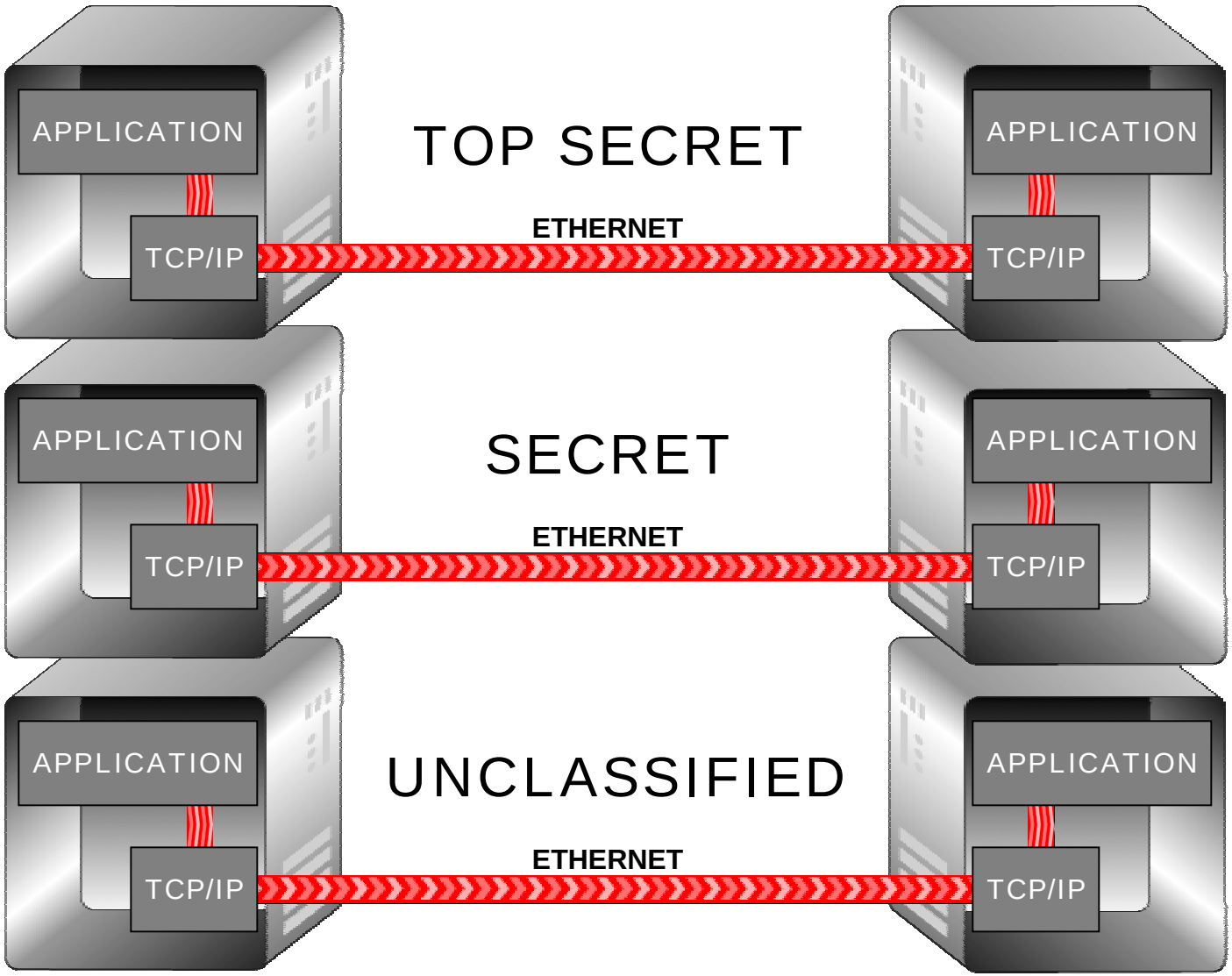

A
B
PCS
PCS
C
D

Partitioning the Channel



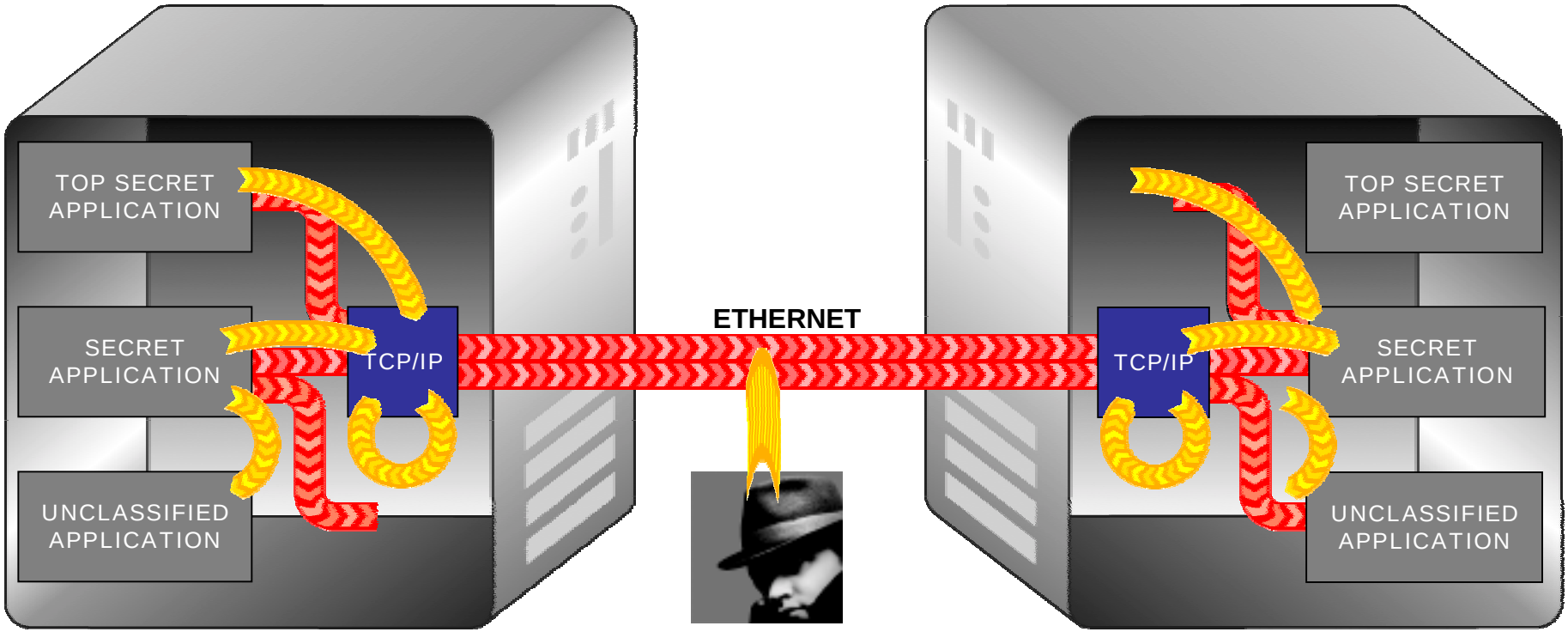
Air Gap Works But.... Costly, Inflexible, & Awkward

Multiple Independent Levels of Security (MILS)



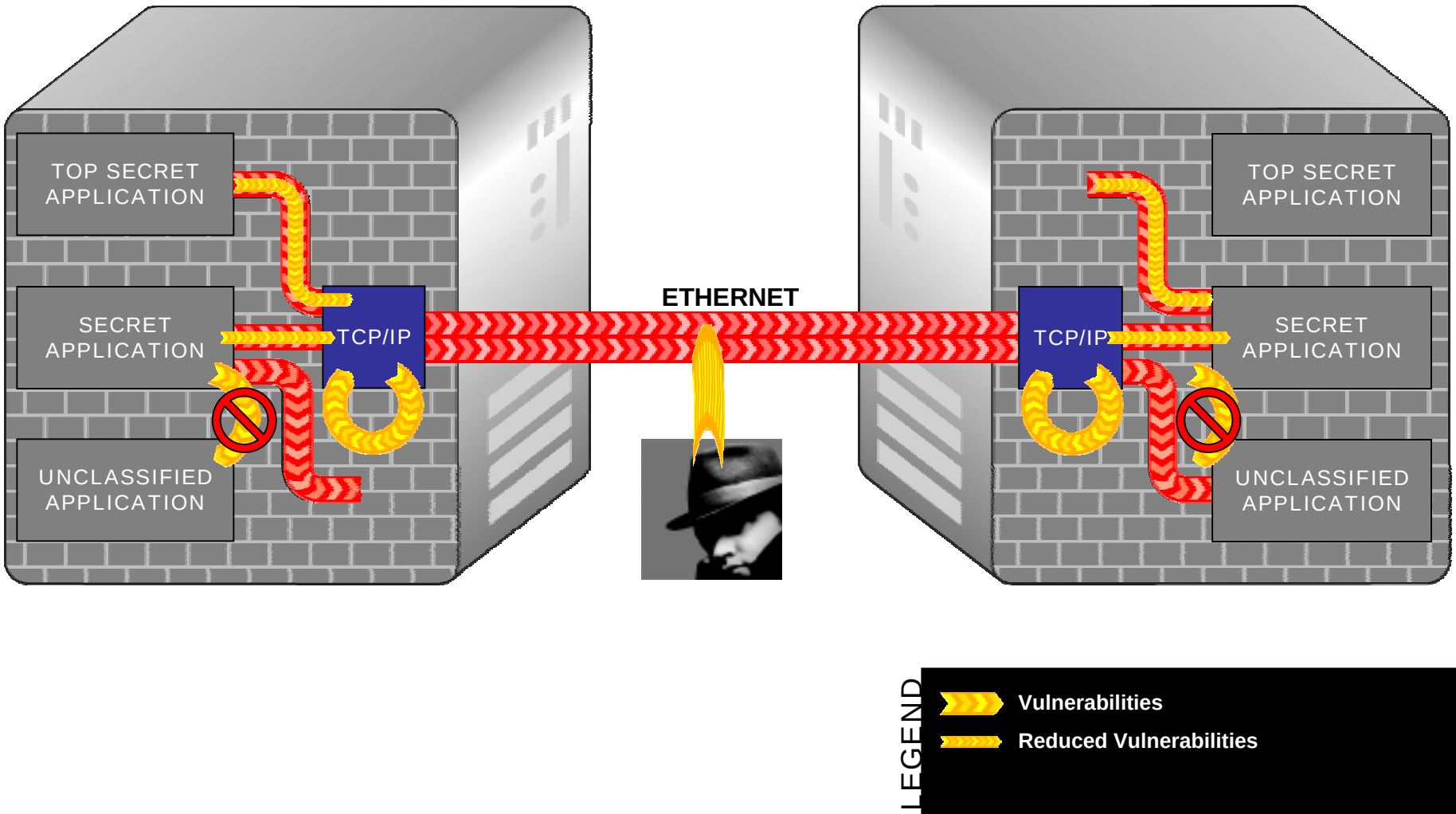
Combining Levels On Medium Assurance Platforms Is Unsafe

Multiple Independent Levels of Security (MILS)



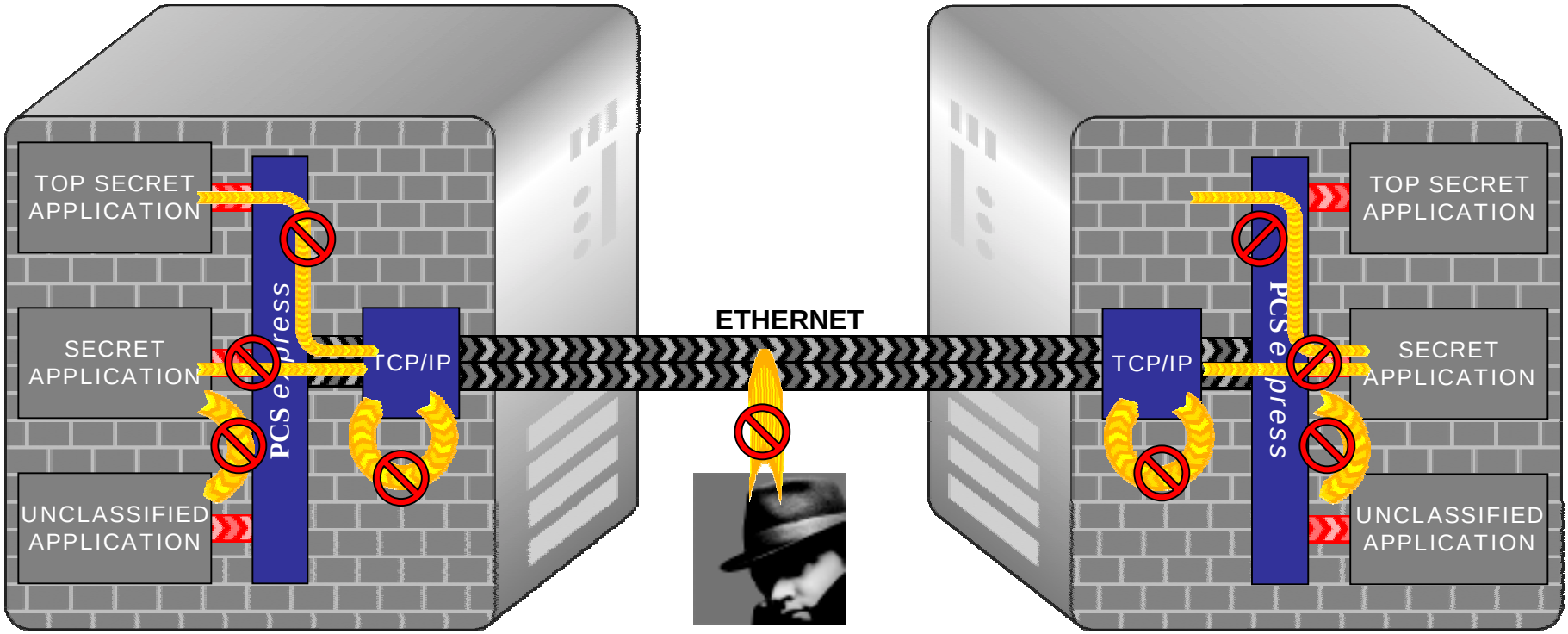
MILS Separation Kernels Counter Most Internal Threats

Multiple Independent Levels of Security (MILS)



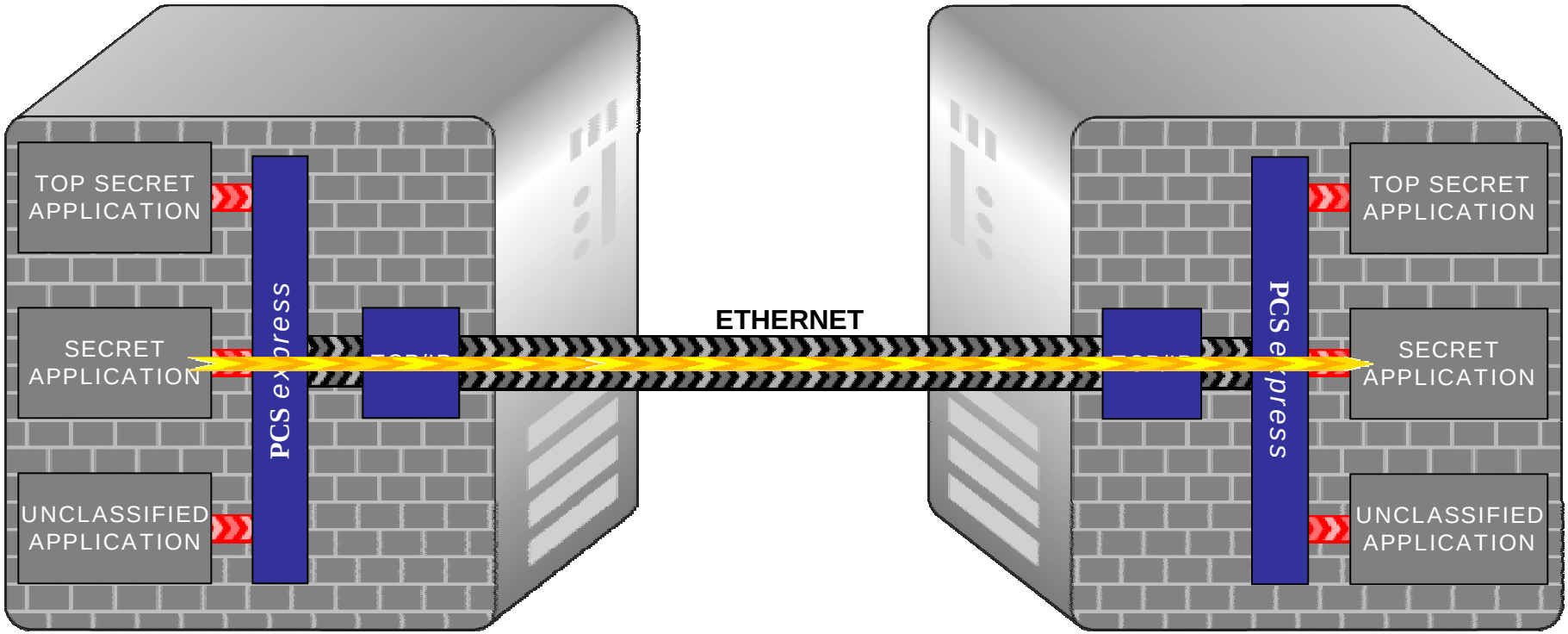
PCS Completes MILS Separation Kernel

Multiple Independent Levels of Security (MILS)



Guards Still Needed for Intra-level Threats

Multiple Independent Levels of Security (MILS)



Agenda

- Vision and Benefits
- Security Evolution
- Foundational Threats
- MILS Architecture
  - Separation Kernel
  - Middleware
  - Applications
- Distributed Security
- Partitioning Communications System
- **Network Middleware**
  - **MILS Real-time CORBA**
  - **MILS Data Distribution Services (DDS)**
- Transition to MILS

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 SK and PCS foundation
- Real-time MILS CORBA represents a single enabling application infrastructure

Real-time MILS CORBA (cont.)

- Can address key cross-cutting system requirements
- MILS-based distributed security
  - High-assurance
  - High-integrity (safety critical systems)
- Real-time
  - Fixed priority
  - Dynamic scheduling
- Distributed object communications
  - Predictable
  - Low latency
  - High bandwidth

RT CORBA & MILS Synergy

- Synthesis yields an unexpected benefit
  - Flexibility of Real-time CORBA allows realization of MILS protection
  - **MILS is all about location awareness**
    - Well designed MILS system separates functions into separate partitions
    - Takes advantage of the MILS partitioning protection
  - **Real-time CORBA is all about location transparency**
    - The application code of a properly designed distributed system built with Real-time CORBA will not be aware of the location of the different parts of the system.
    - CORBA flexibility allows performance optimizations by rearranging what partitions each system object executes in.
    - System layout can be corrected late in the development cycle
  - **Combination of MILS and Real-time CORBA allows system designer**
    - ***Rearrange system functions to take advantage of protection without introducing new threats to data confidentiality and integrity***

MILS DDS

- OMG Data Distribution Specification
  - Data-centric publish-subscribe
- PCS protects DDS implementations from
  - Attack by other partitions
  - Network attacks
  - Covert channels
- DDS can take advantage of PCS capabilities
  - PCS + DDS => MILS DDS
  - Application-level security policies are enforceable because of MILS SK and PCS foundation

Agenda

- Vision and Benefits
- Security Evolution
- Foundational Threats
- MILS Architecture
  - Separation Kernel
  - Middleware
  - Applications
- Distributed Security
- Partitioning Communications System
- Network Middleware
  - MILS Real-time CORBA
  - MILS Data Distribution Services (DDS)
- **Transition to MILS**

TYPICAL TRANSITION GUIDELINES:

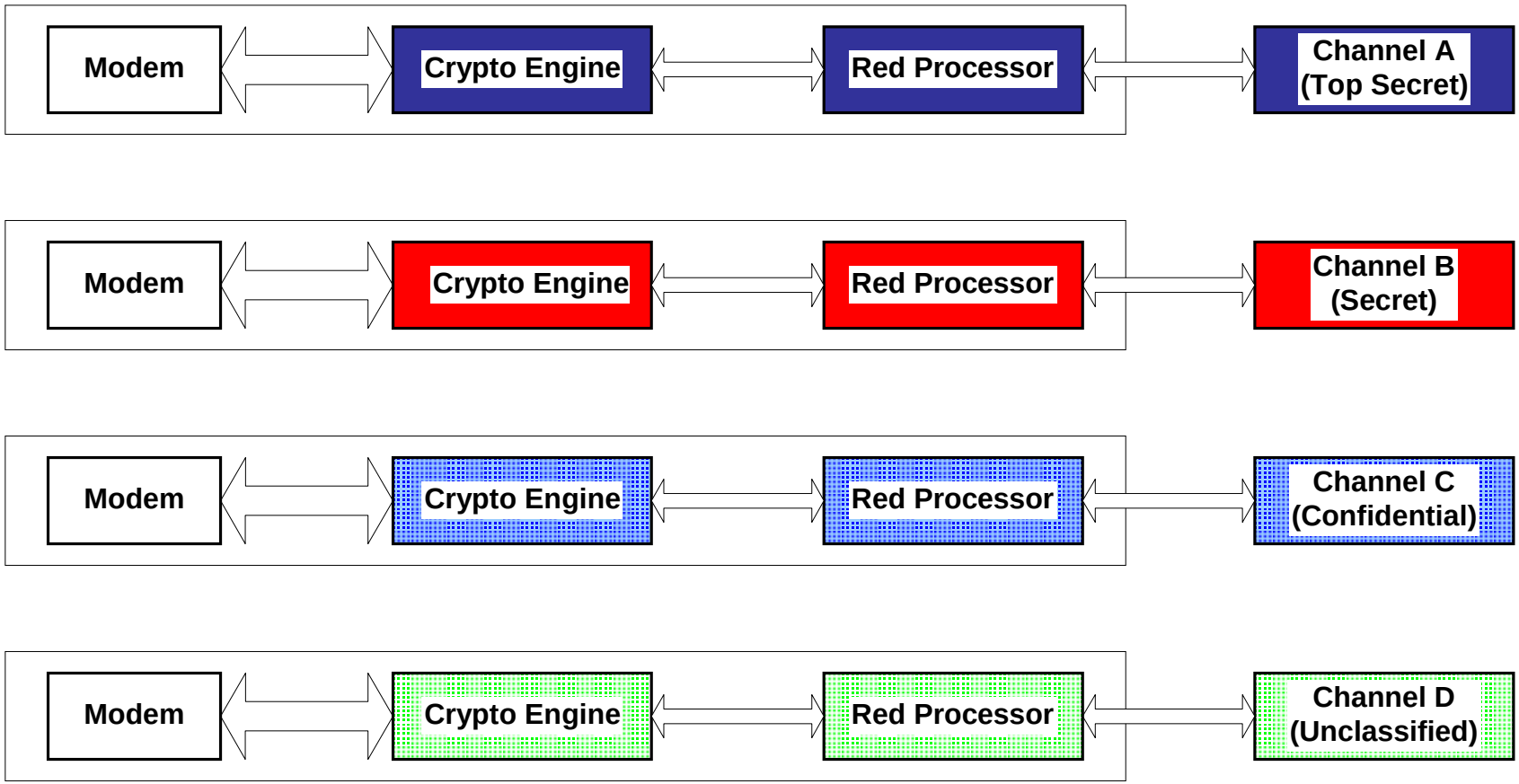
- Move Drivers from Privilege Mode to User Mode
- Import / Export PDU Labels from MLS Drivers

BUT WHAT WE REALLY WANT IS:

- MILS/MLS Intelligent IO Devices
  - IO Device Interface via User Mode Partitions ONLY
  - IO Device Supports Multiple User Mode Partitions
    - Each User Mode Partition has own Clearance
  - IO Device manages Clearance of User Mode Partitions
    - User Mode Partitions not trusted to report Clearance
  - IO Device Imports / Exports Security Label
    - Will not allow Write Down nor Read Up
  - Network Interface Unit (NIU) and Rapid-IO Examples

MILS Roadmap
Single Channel Legacy Systems

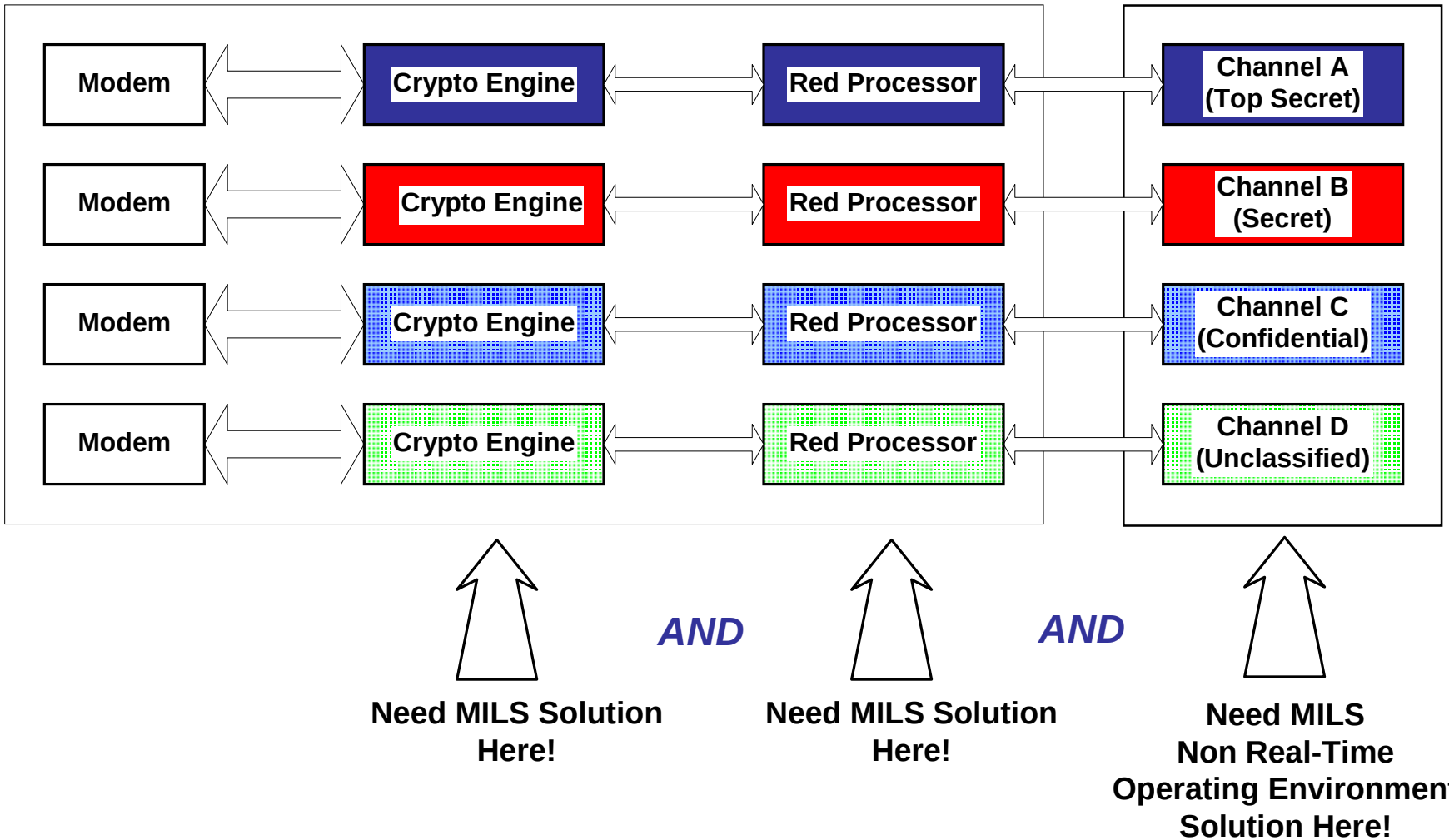
Multiple Independent Levels of Security (MILS)



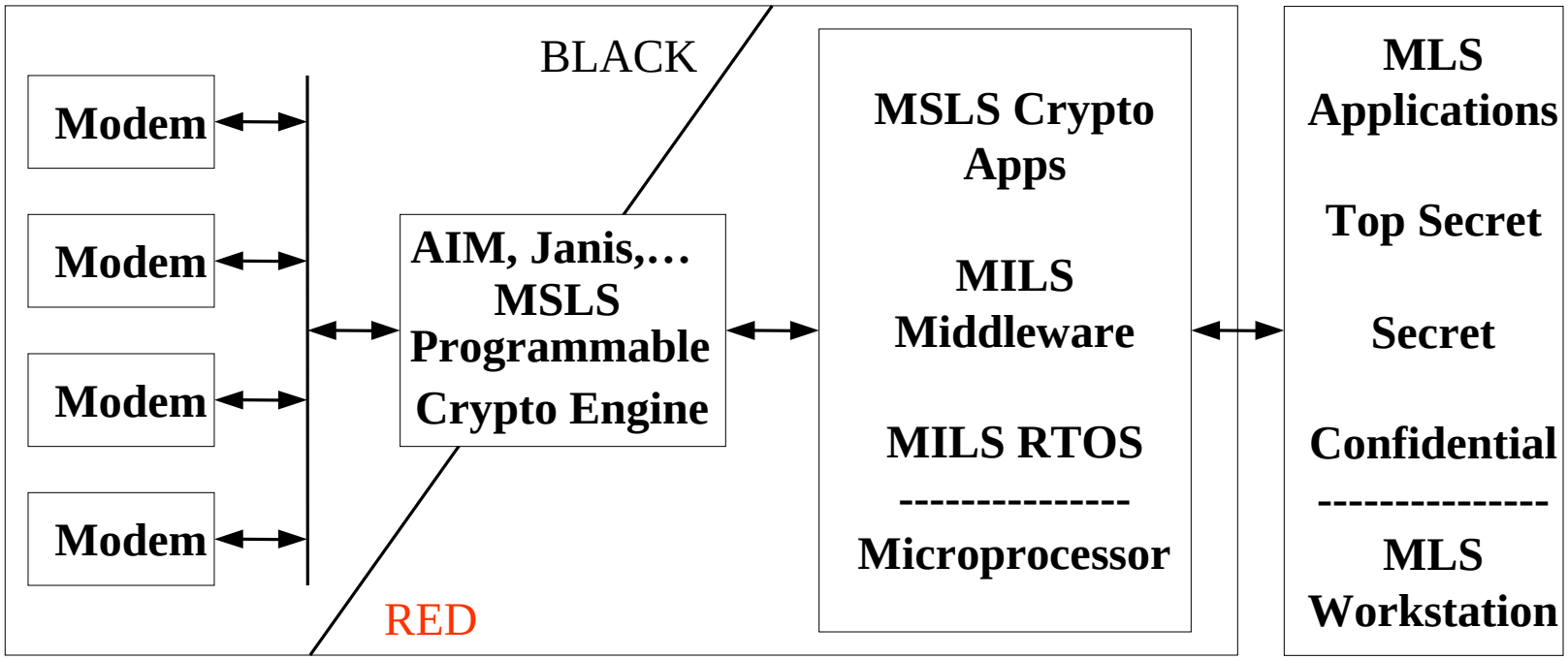
This Is Current Stovepipe Technology That Is Expensive And Inflexible

MILS Roadmap
Supports MILS via Physical Separation

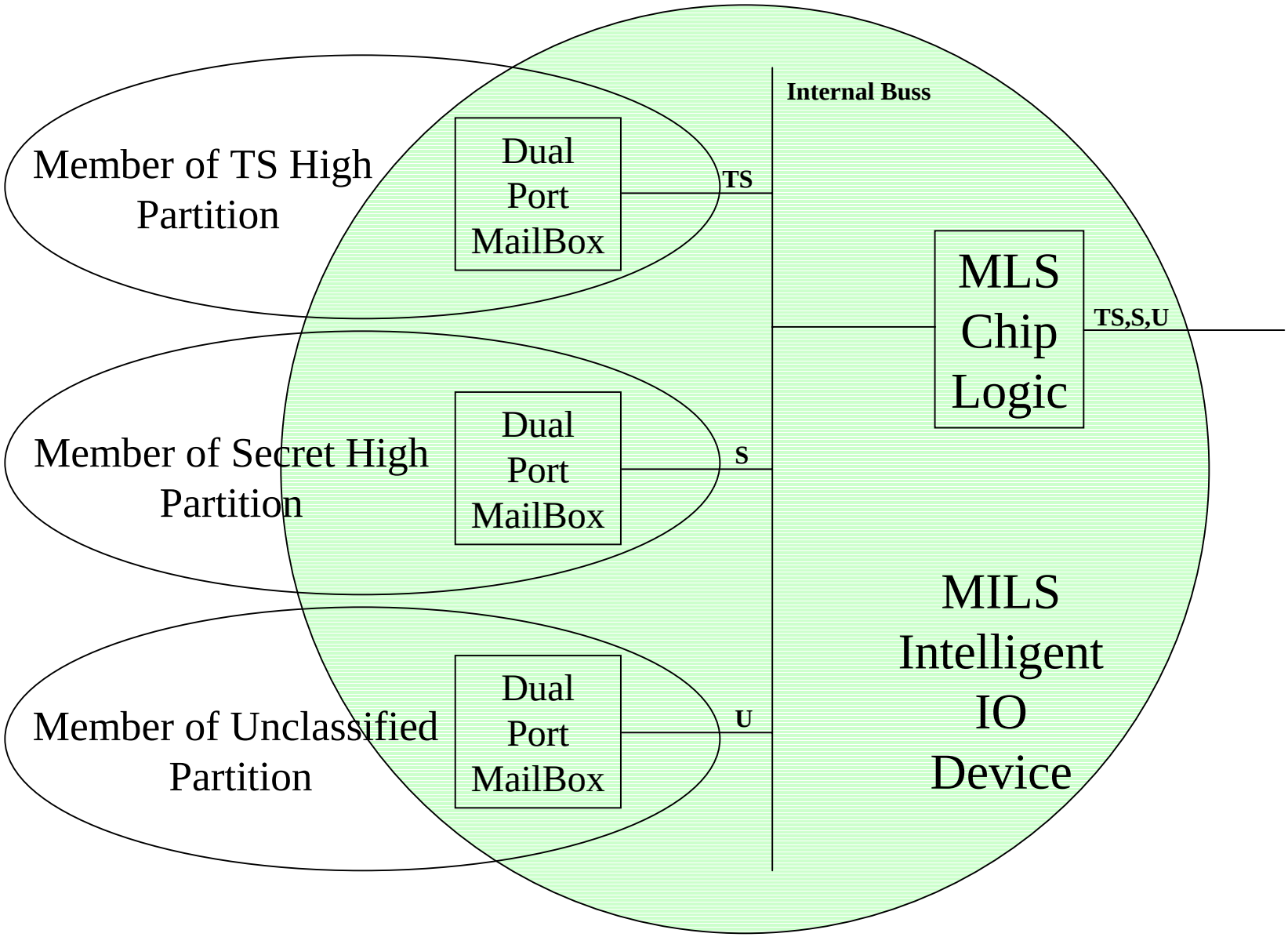
Multiple Independent Levels of Security (MILS)



Multiple Independent Levels of Security (MILS)



MILS Intelligent IO Device



Are you ready for the Global Information Grid?