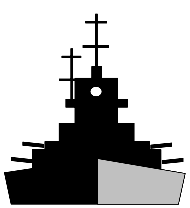
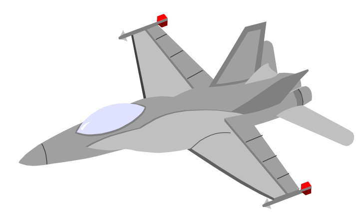
The MILS Partitioning Communication System + RT CORBA = Secure Communications for SBC Systems

Kevin Buesing
Objective Interface Systems
Field Applications Engineer
kevin.buesing@ois.com

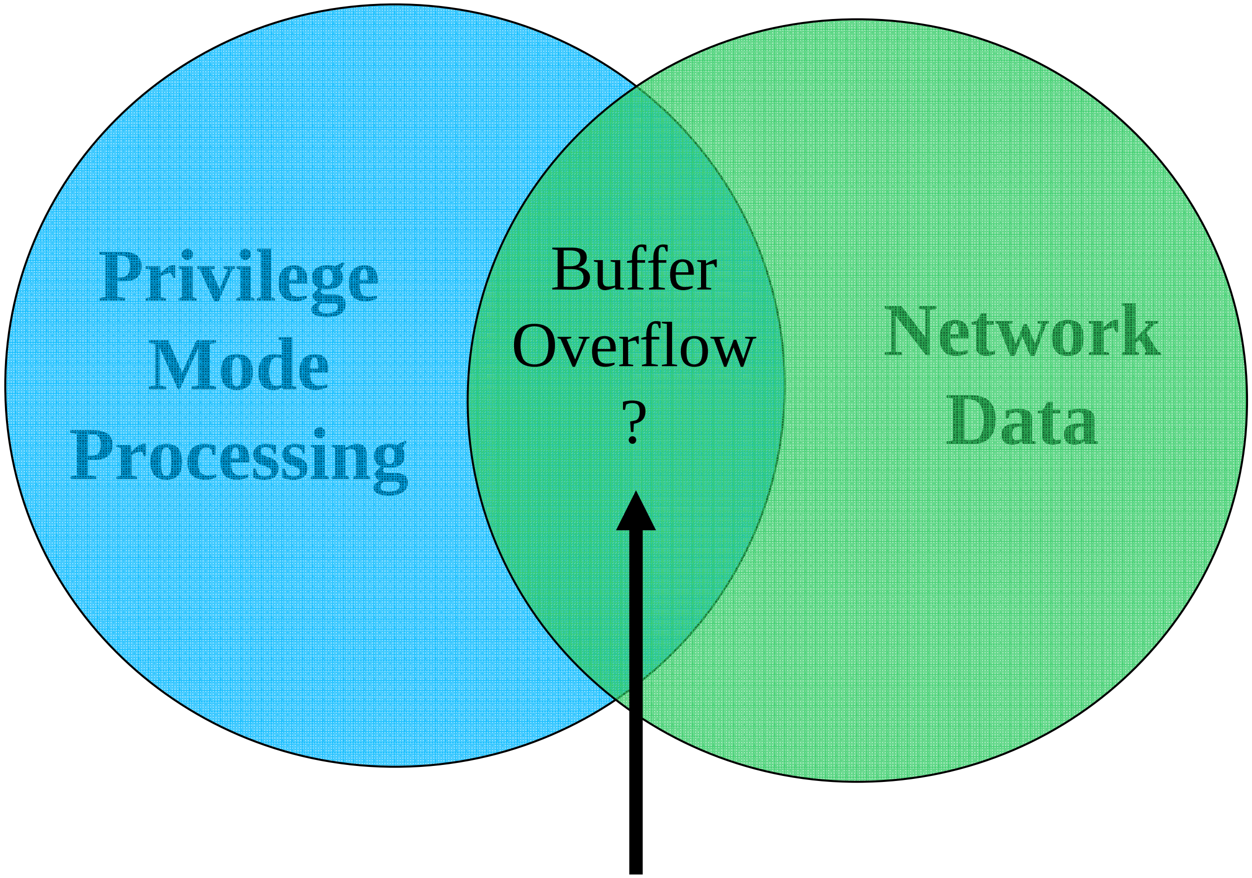
Jeff Chilton
Objective Interface Systems
Senior Product Engineer
jeff.chilton@ois.com



This presentation represents joint research between the
Air Force, Army, Navy, NSA, Boeing, Lockheed Martin, Objective Interface,
Green Hills, Lynux Works, Wind River, GD, Rockwell Collins, MITRE, U of Idaho



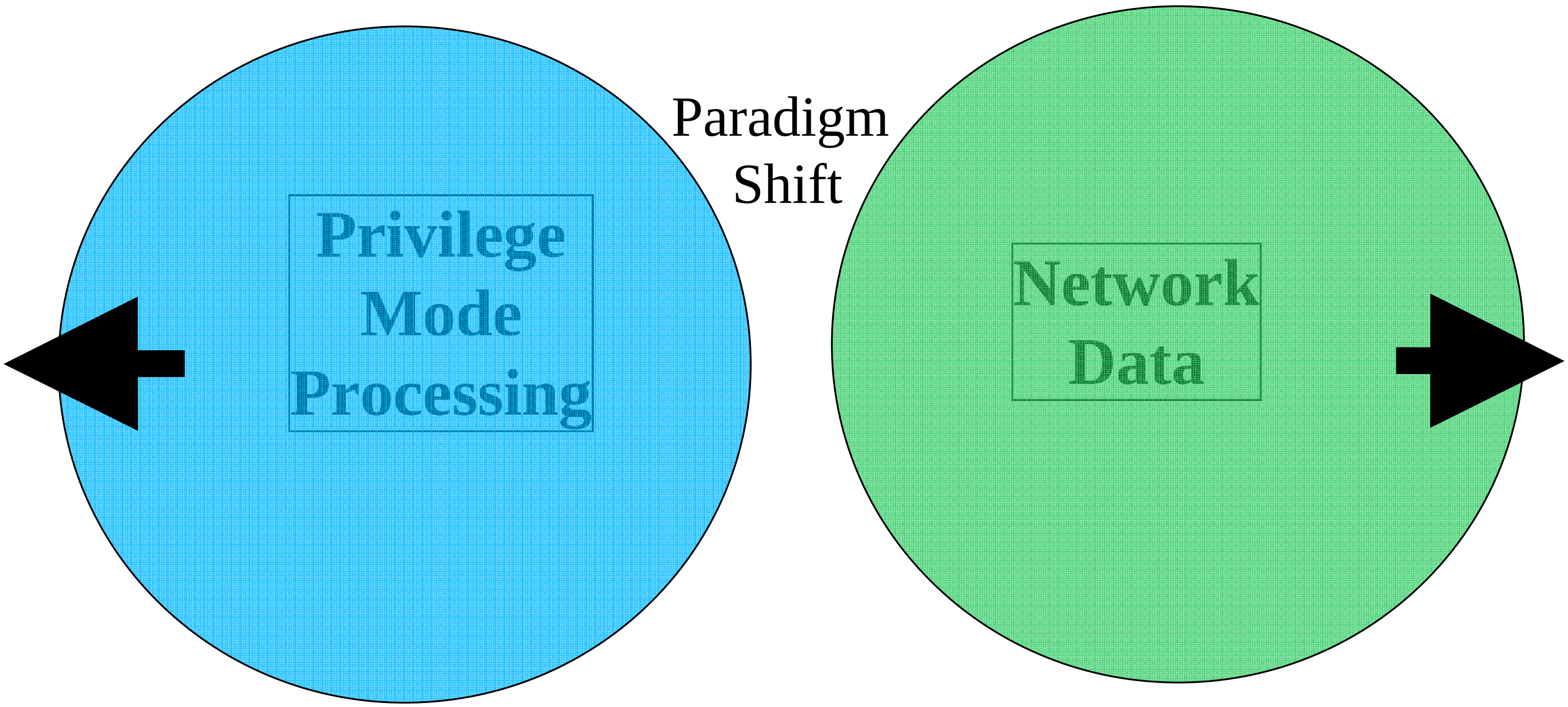
Foundational Threats



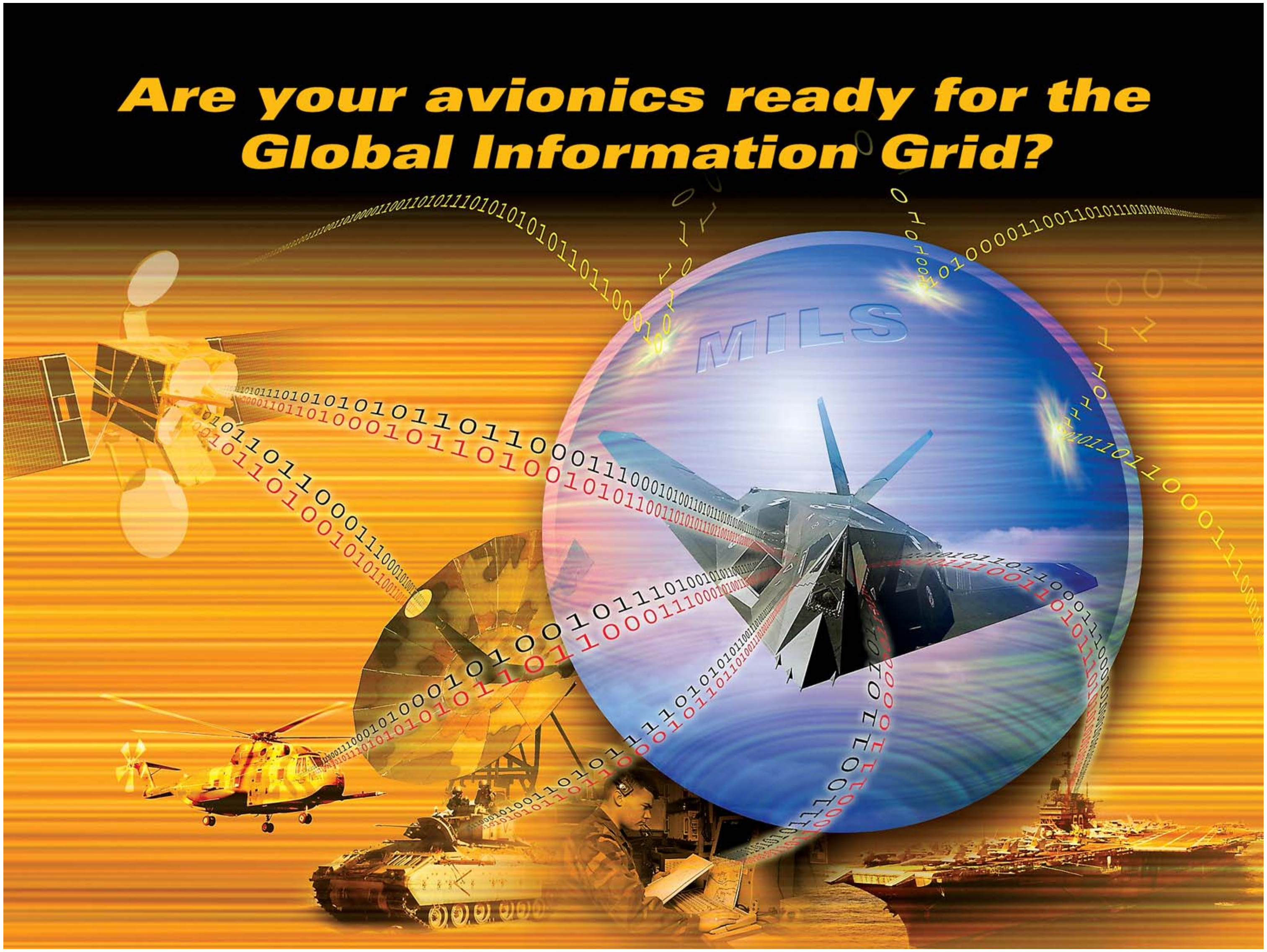
Wild Creatures of the Net, Worms, Virus, . . .

Foundational Threats

(That MILS Protects Against)



Under MILS Network Data and Privilege Mode Processing is Separated


Are your avionics ready for the Global Information Grid?
MILS

MILS Overview

The Whole Point of MILS

Really simple:

- Dramatically **increase the scrutiny** of *security critical code*
- Dramatically **reduce the amount** of *security critical code*

What does MILS do?

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*
***T**amper-proof*

Reference Monitor Concept

MILS = Multiple Independent Levels of Security/Safety

MILS Architecture Objectives

How does MILS achieve its objectives?

Enforce an

Information Flow,

Data Isolation,

Periods Processing, and

Damage Limitation ***Security Policy***

between multiple address spaces:

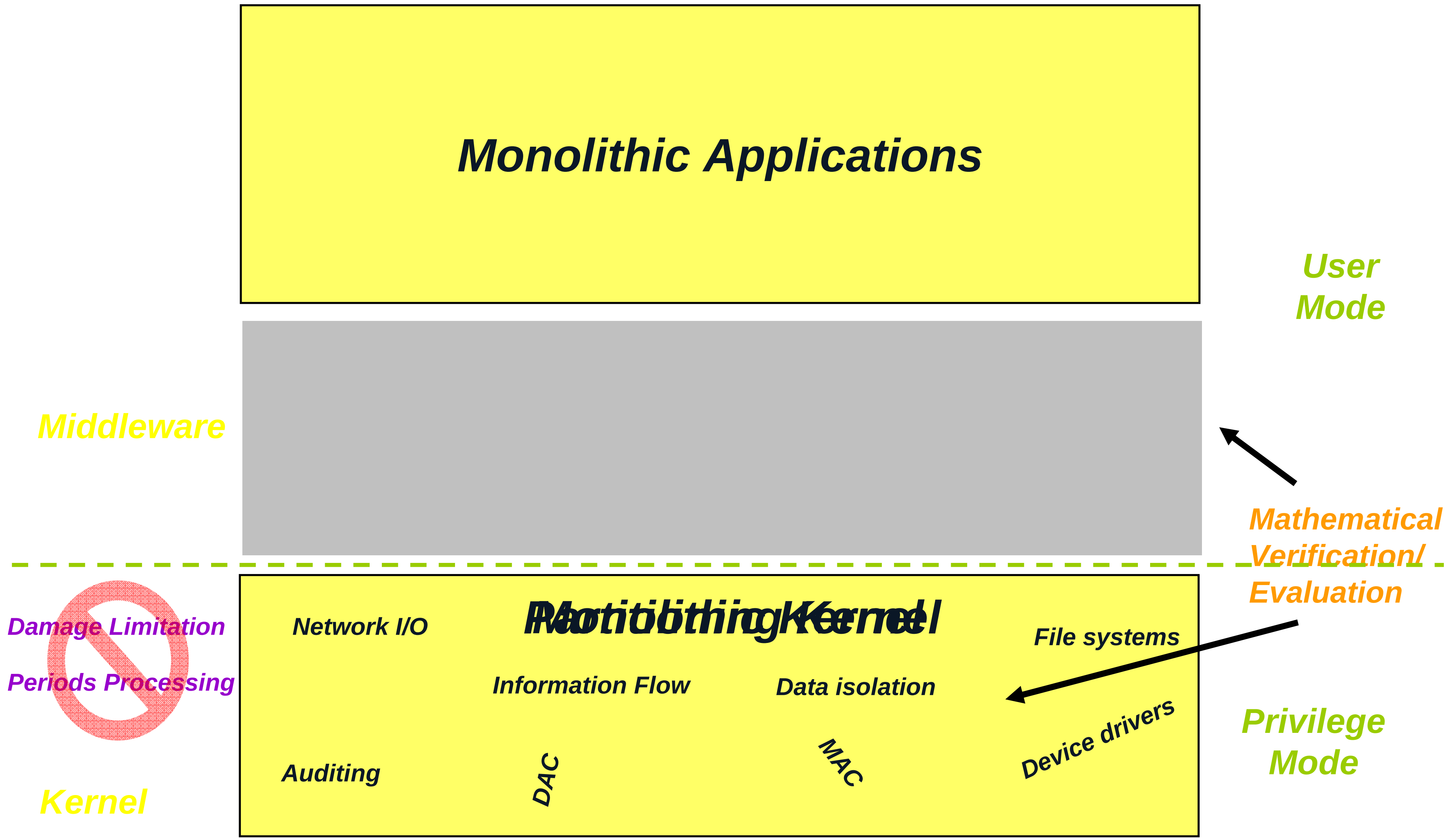
First, in a **Microprocessor Centric Manner**, i.e., MILS RTOS Kernel,

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

in such a manner that the layered Security Policies are

N E A T

Orange Book vs. MILS Architecture



Three distinct layers (John Rushby, PhD)

Partitioning Kernel

- Trusted to guarantee separation of time and space
  - Separate process spaces (partitions)
  - Time partitioning
- Secure transfer of control between partitions
- Really small: 4K lines of code

1. Middleware
   - Secure application component creation
   - Secure end-to-end inter-object message flow
   - Most of the traditional operating system functionality
     - Device drivers, file systems, etc.
   - Partitioning Communications System
     - Extends the policies of Partitioning Kernel to communication
     - Facilitates traditional middleware
       - Real-time CORBA, DDS, web services, etc.

2. Applications
   - *Can* enforce application-specific security functions
   - e.g., firewalls, crypto services, guards

Layer Responsibilities

Partitioning Kernel Functionality

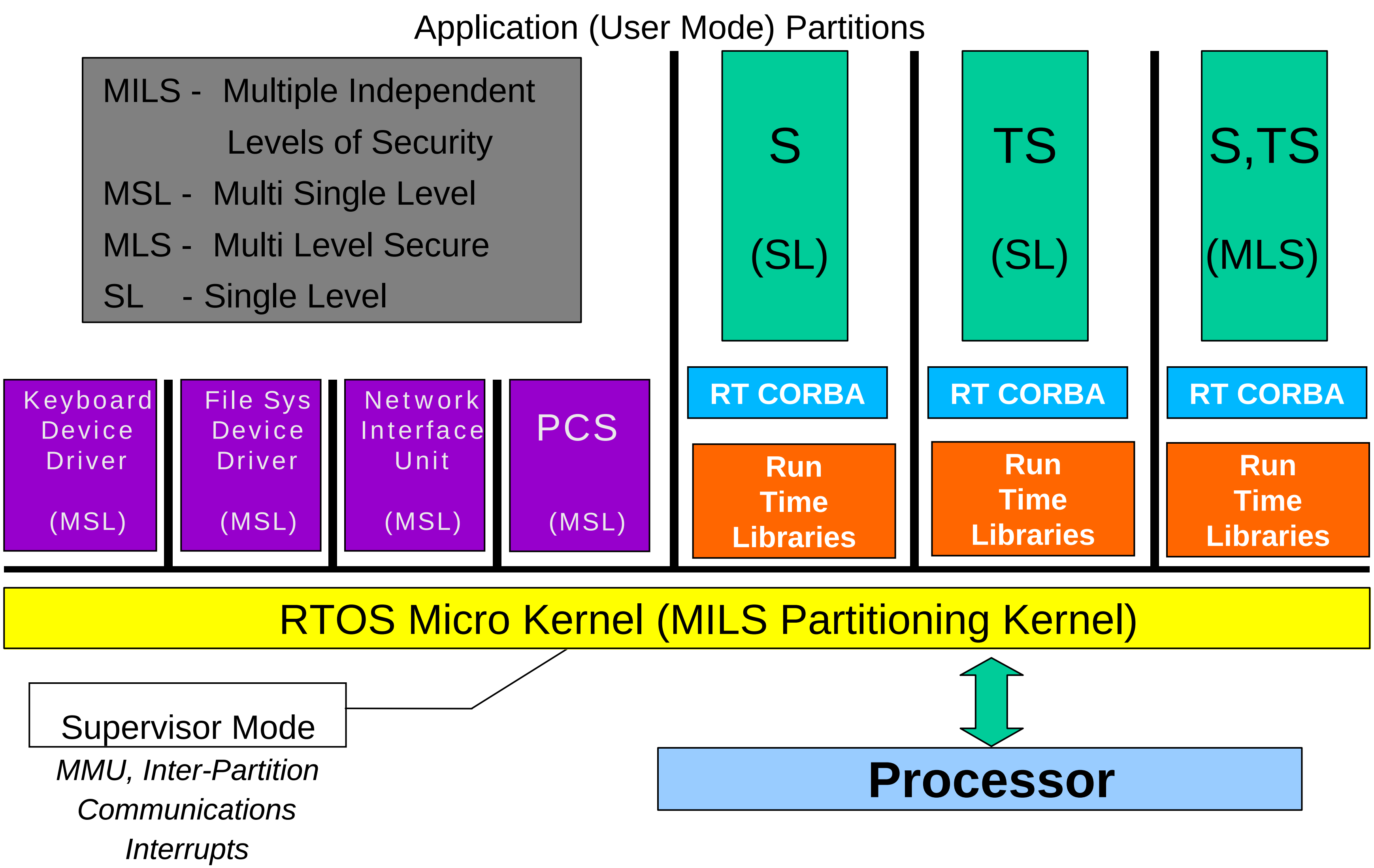
- Time and Space Partitioning
- Data Isolation
- Inter-partition Communication
- Periods Processing
- Minimum Interrupt Servicing
- Semaphores
- Timers
- Instrumentation

And nothing else!

MILS Middleware Functionality

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

Executive Overview
MILS Architecture – High Assurance



Partitioning Kernel: Just a Start …

- Partitioning Kernel provides
  - Secure foundation for secure middleware
- Secure Middleware provides
  - Most of traditional O/S capabilities
    - File system
    - Device drivers (*not* in the kernel, not special privileges)
    - Etc.
  - Secure intersystem communication (PCS)
  - Secure foundation for building secure applications
- Secure Applications can
  - Be built!
  - Be trusted to enforce application-level security policies!!!

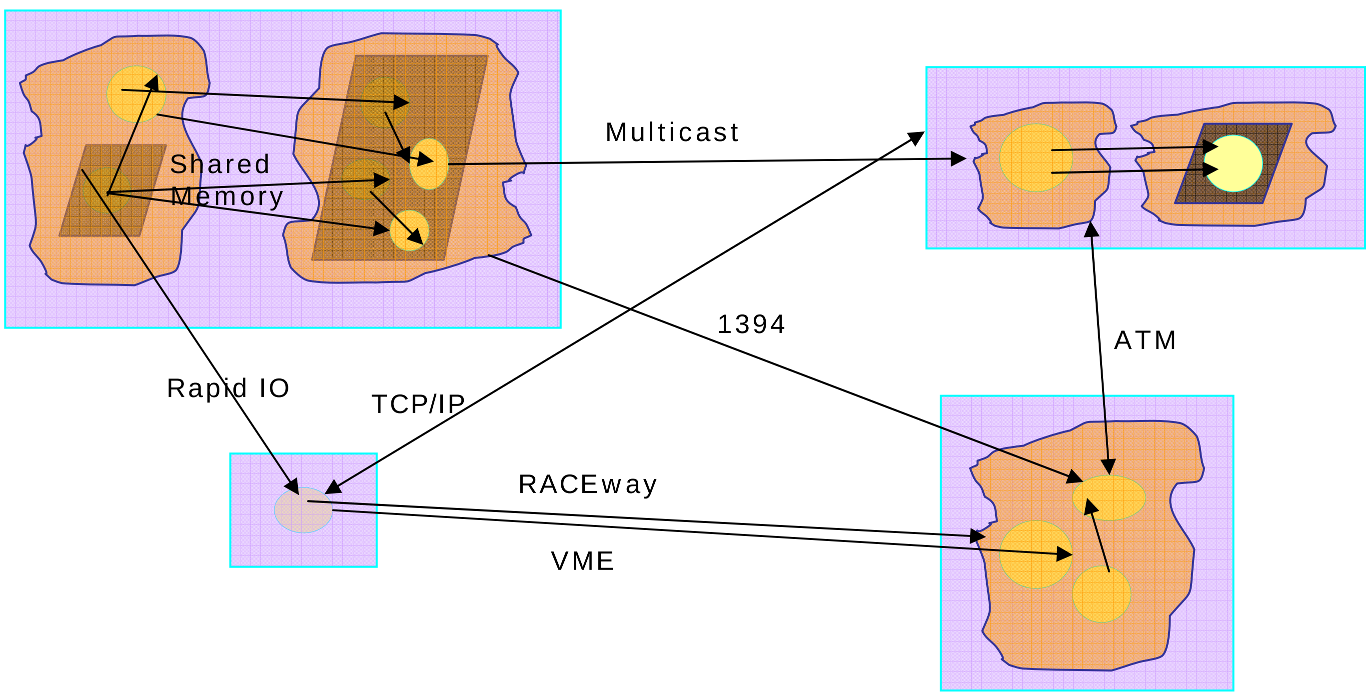
Distributed Security

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 tolerance
  - Security infrastructure must have no single point of failure
  - Security infrastructure must support fault tolerant applications

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



Partitioned Communication System

Partitioned Communication System

- Partitioned Communication System
  - Part of MILS Middleware
  - Responsible for all communication between MILS nodes
- Purpose
  - Extend MILS partitioning kernel protection to multiple nodes
- Similar philosophy to MILS Partitioning 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 EAL level 7 evaluation

PCS Objective

- Just like MILS Partitioning 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 Requirements

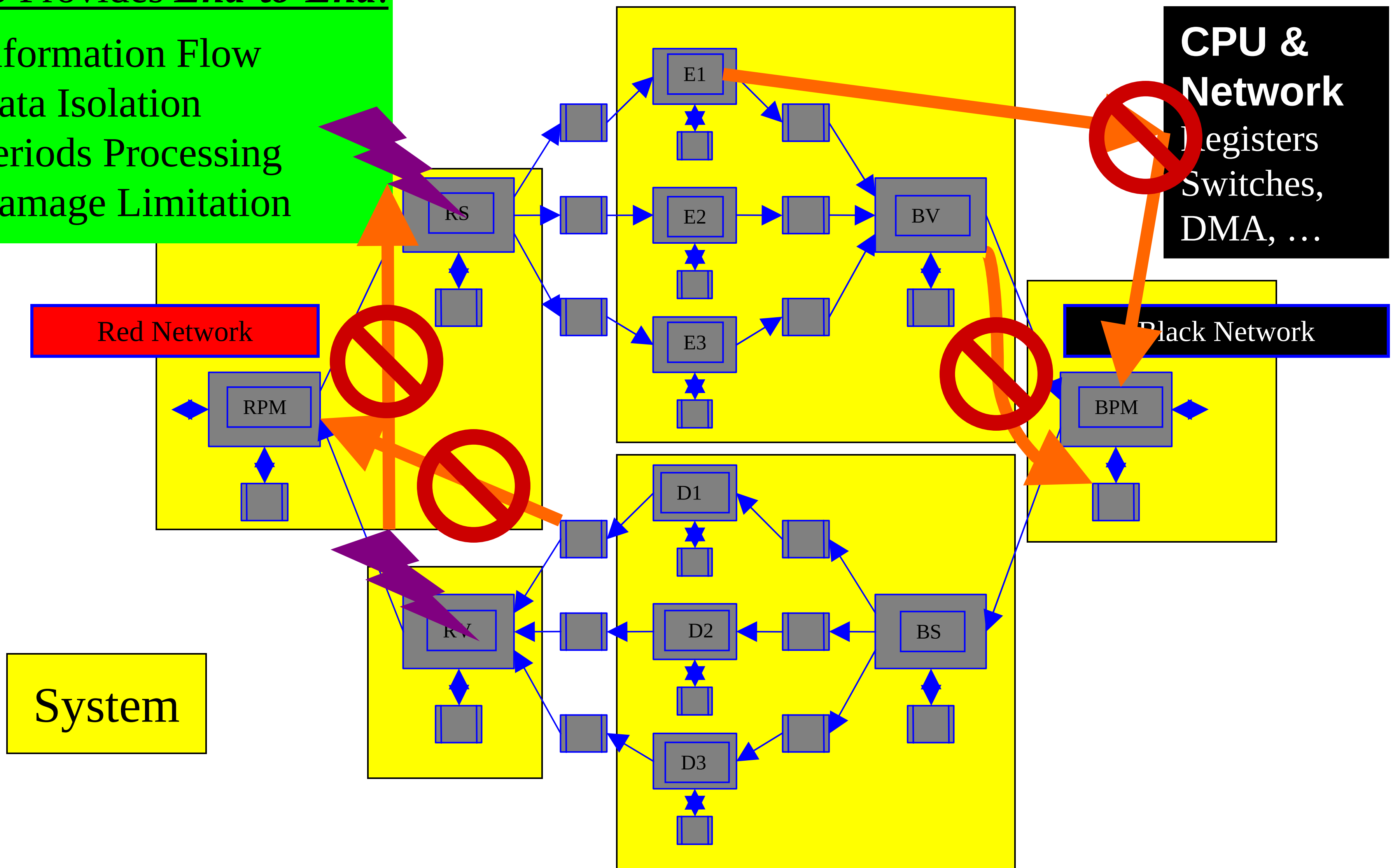
- Strong Identity
  - Nodes within enclave
- Separation of Levels/Communities of Interest
  - Need cryptographic separation
- Secure Configuration of all Nodes in Enclave
  - Federated information
  - Distributed (compared) vs. Centralized (signed)
- Secure Loading: signed partition images
- Suppression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

Executive Overview
MILS Network Security Policy
Example

Policy Enforcement Independent of Node Boundaries

<u>PCS Provides ***End-to-End***:</u>

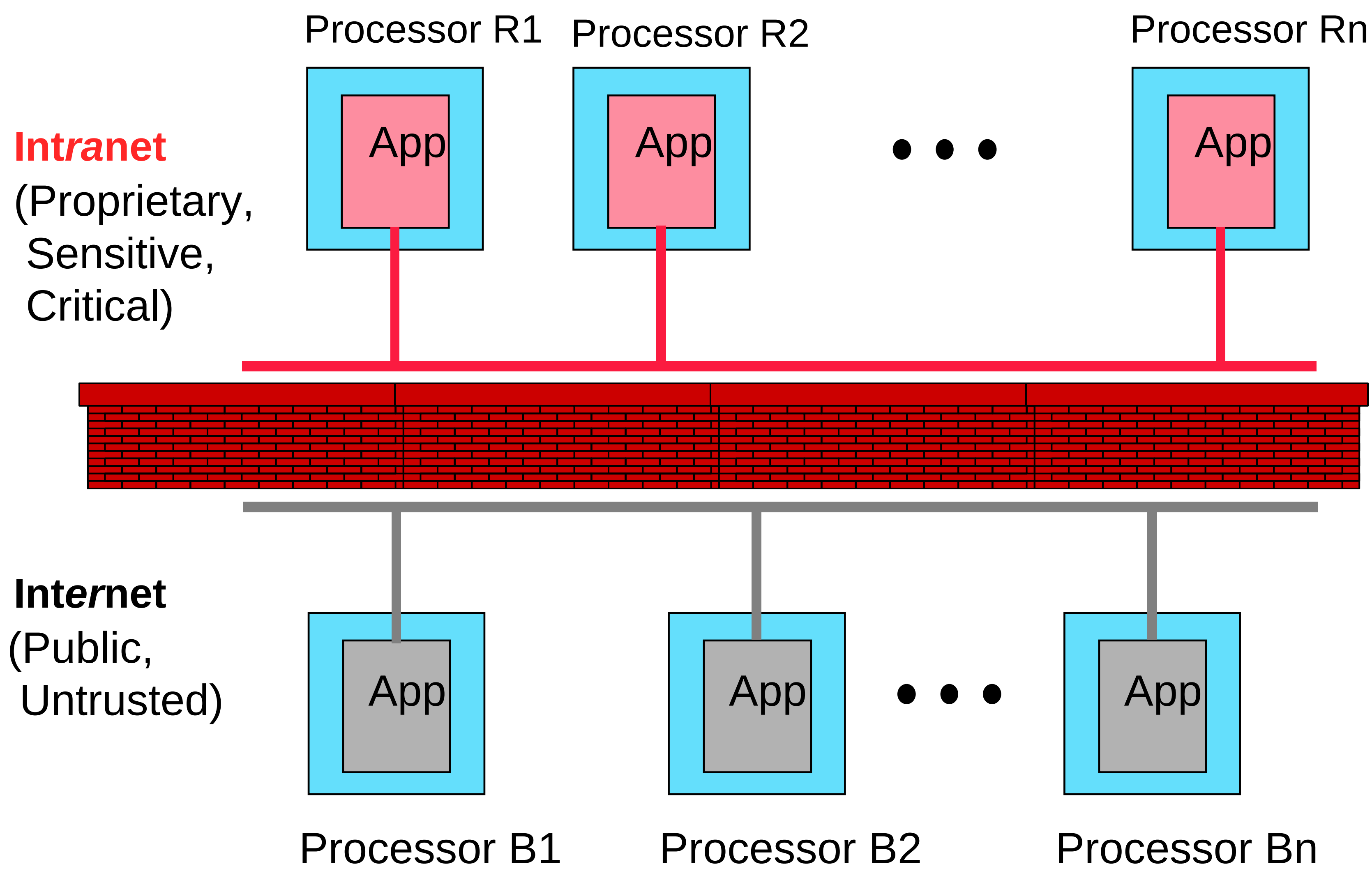
- Information Flow
- Data Isolation
- Periods Processing
- Damage Limitation



MILS Replaces Physical Separation

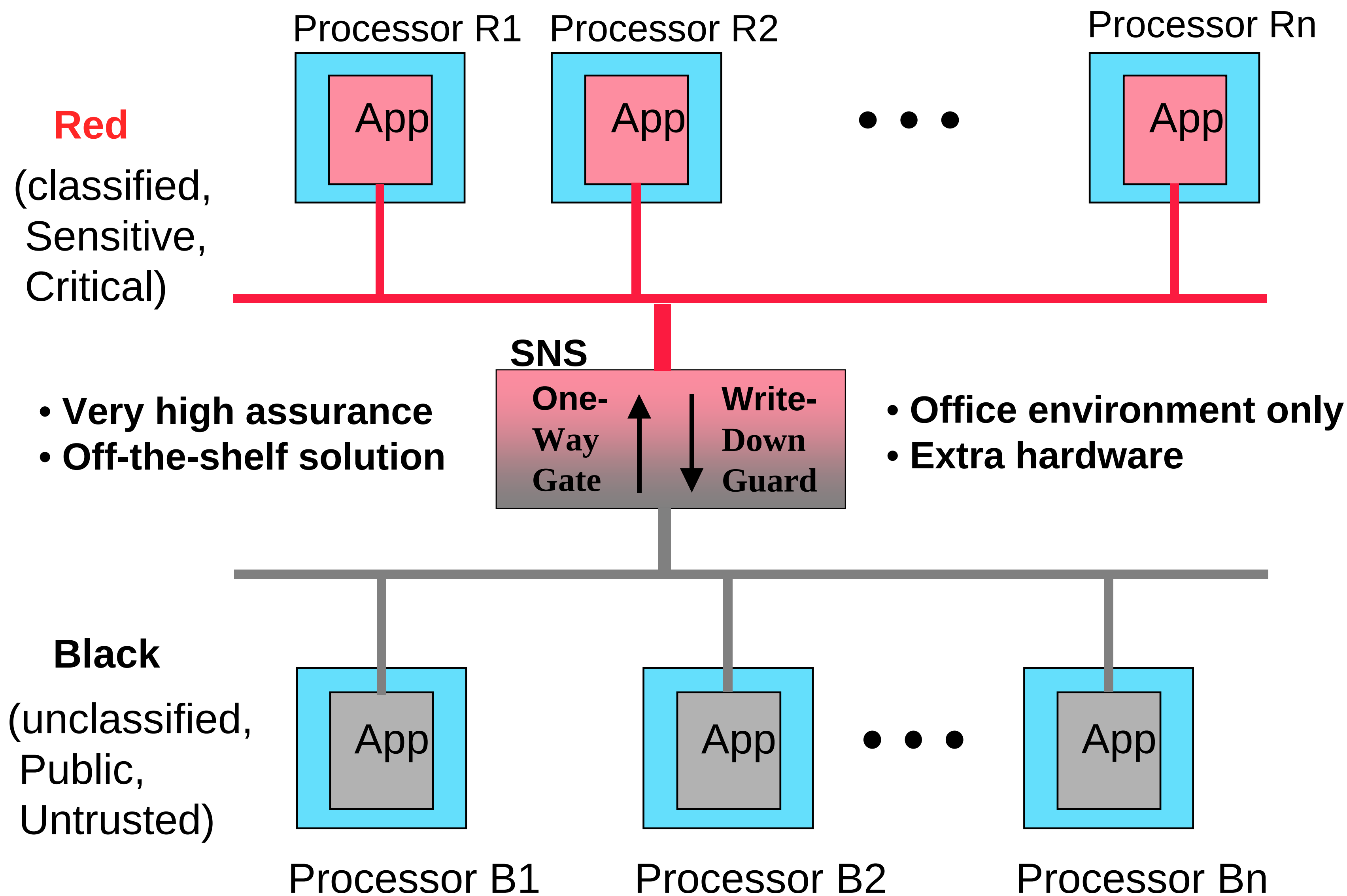
- MILS architecture allows computer security measures to achieve the assurance levels as “physically isolated” systems
  – All O/S code not necessary for performing Partitioning Kernel functions moved out of privileged mode
  – O/S service code moved to middleware layer
    - e.g. device drivers, file system, POSIX
  – Prevents software and network attacks from elevating a partition privilege to an unauthorized level

Best Security/Safety is Physical (Air Gap)

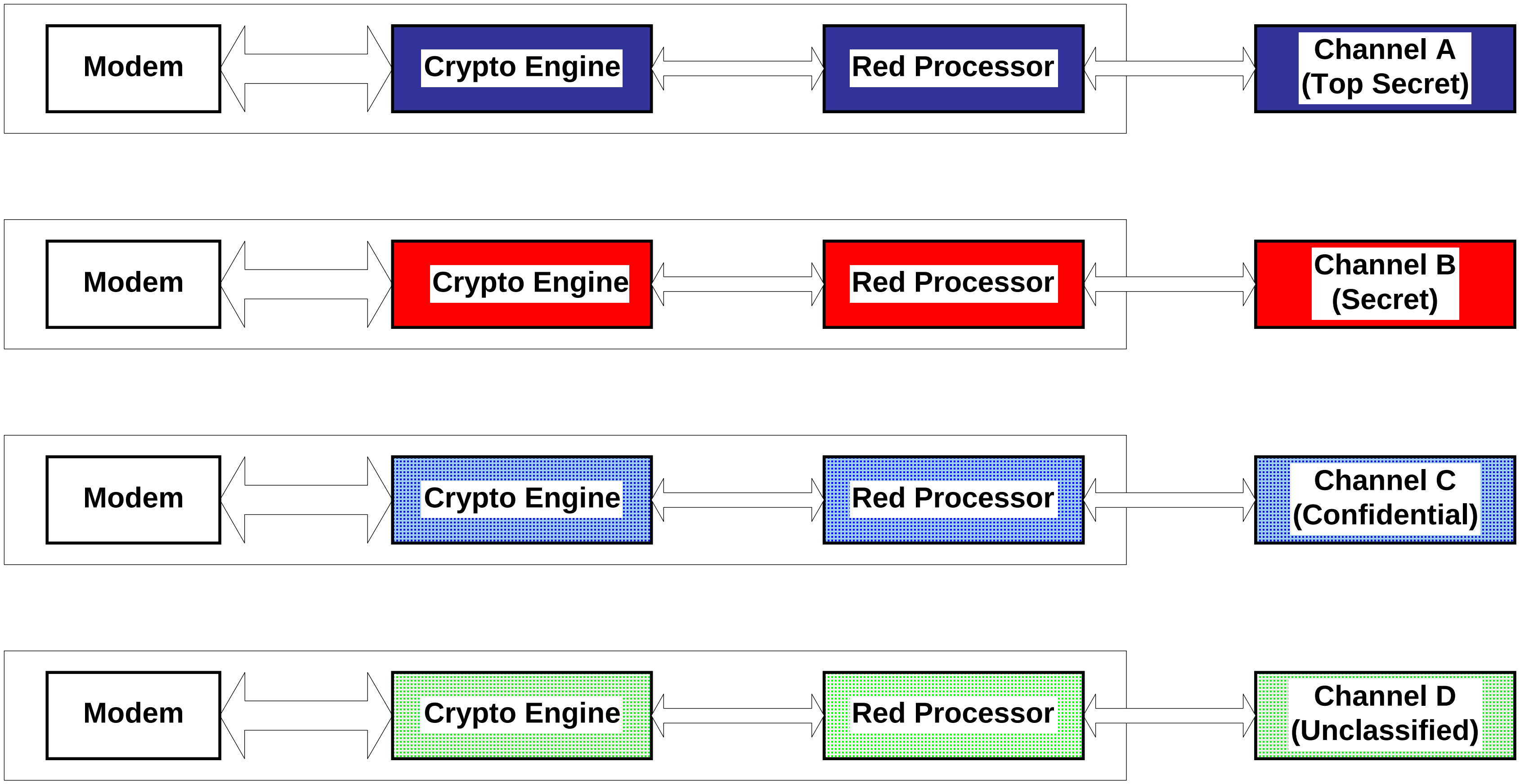


Legacy Approach to Bridging the Air Gap

(Good, Expensive, Physical Solutions Exist)

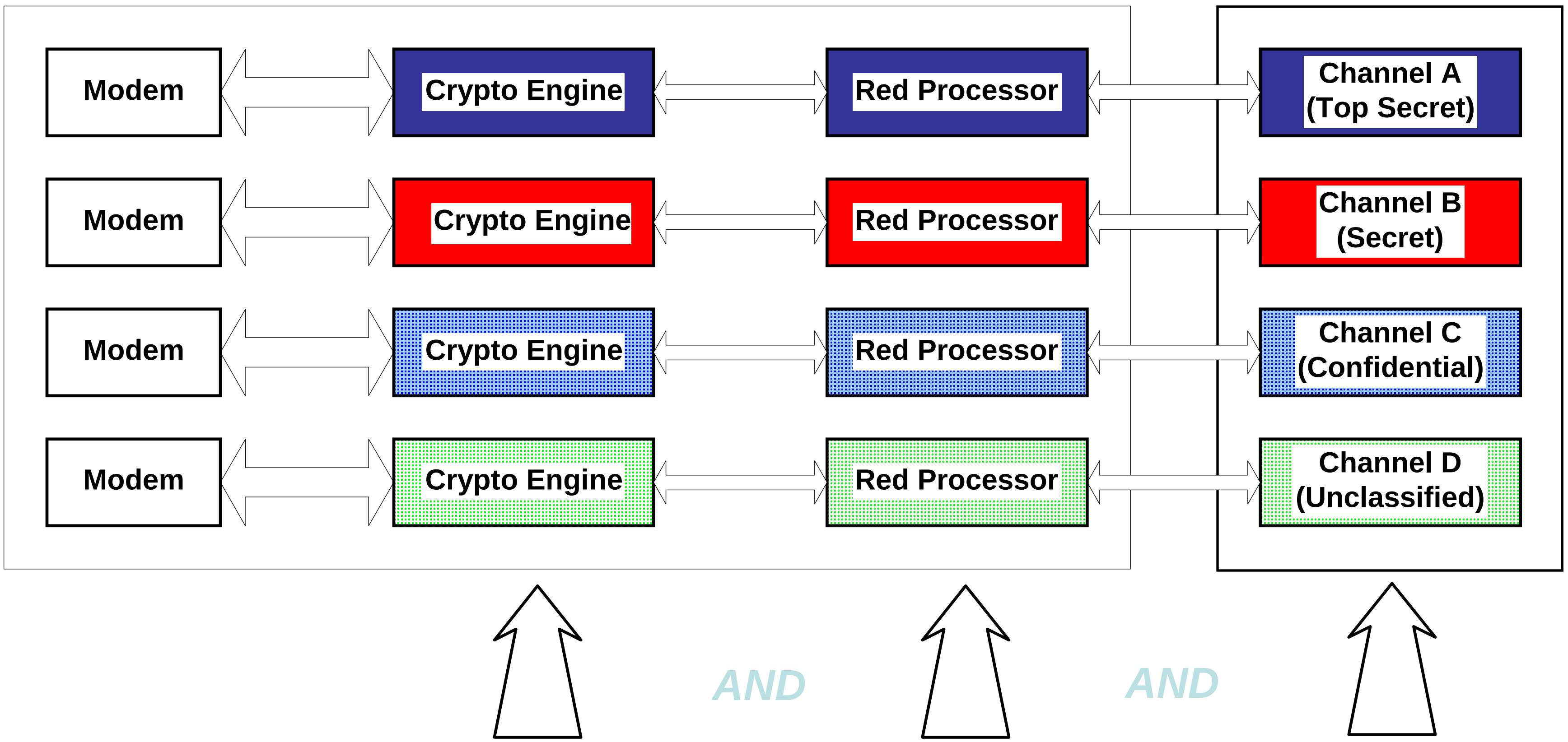


Air Gap Solution to SDR – Separate Hardware



This Is Current Stovepipe Technology That Is Expensive And Inflexible

A Simple Application of MILS to SDR – Separate Processor *Resources*



Need MILS Solution Here!

Need MILS Solution Here!

Need MILS Non Real-Time Operating Environment Solution Here!

Introduction – MLS/MSLS

Multi-Level Secure/Safe (MLS): Processes data of differing classifications/sensitivities securely/safely

- down graders
- data fusion
- guards
- firewalls
- data bases

Multi-Single Level Secure/Safe (MSLS): Separates data of differing classifications/sensitivities securely/safely simultaneously

- communications platforms
- infrastructures

MILS Can Handle MLS

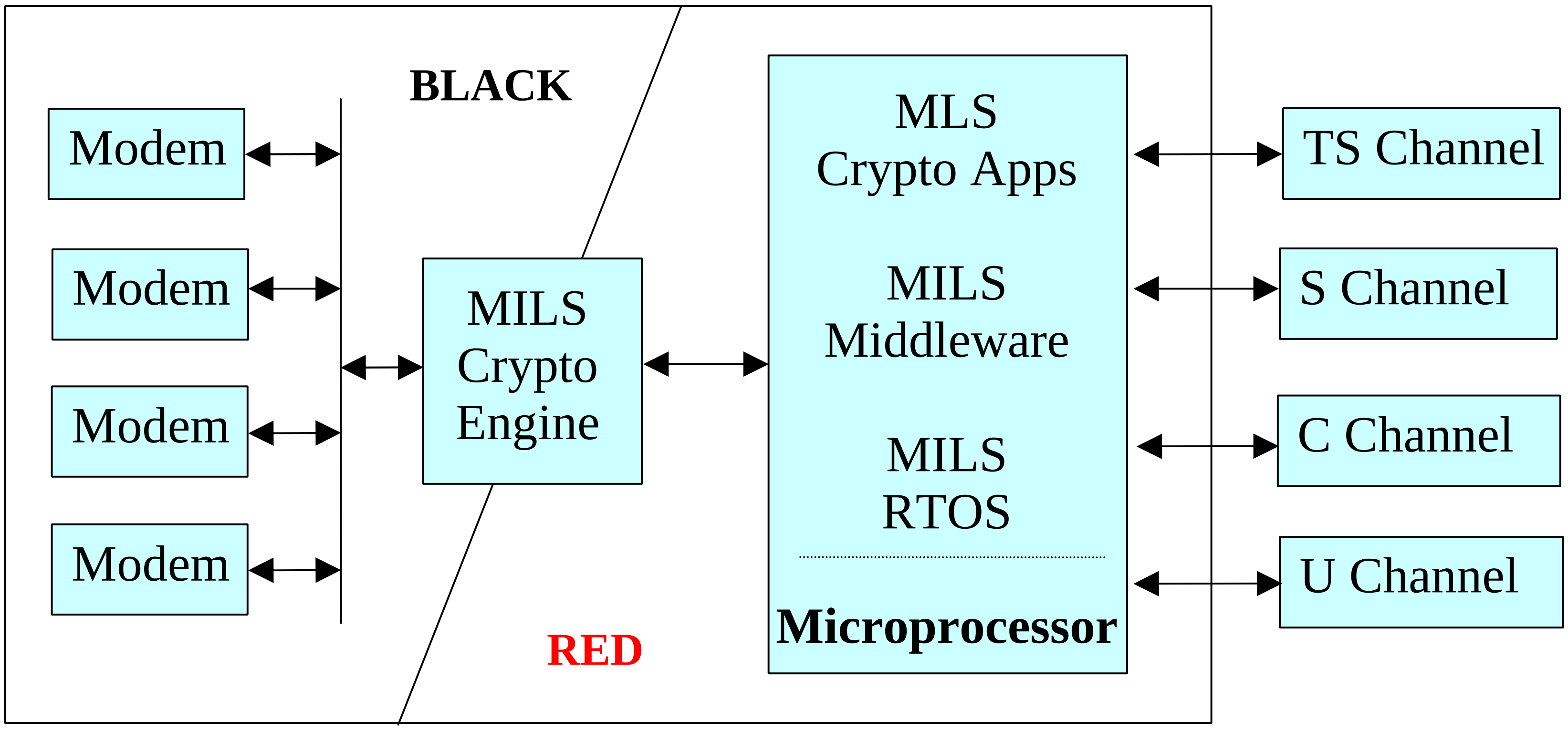
- A Partitioning Kernel is ignorant of traditional Multi-Level Security (MLS)
  - Requirement for military and intelligence systems
- However, MILS is quite capable of supporting MLS systems
- MILS can be used to construct MLS systems because of
  - Strong separation guarantees
  - Certification process

Applying MILS to Software Defined Radio

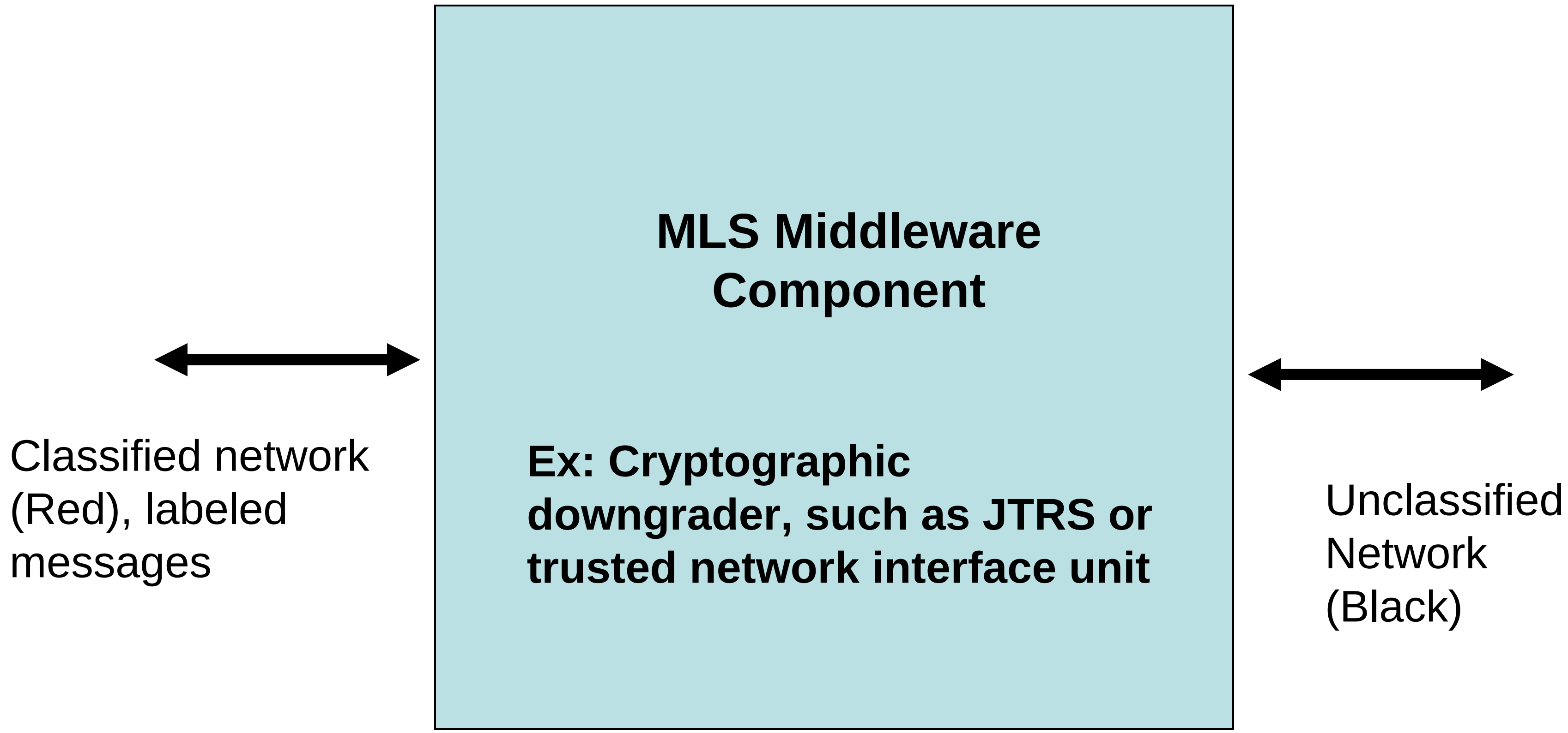
Example – JTRS
Joint Tactical Radio System

- Family of software programmable radios
- Design around Software Communications Architecture
- JTRS provides reliable multichannel voice, data, imagery, and video communications
- Eliminates communications problems of "stovepipe" legacy systems
- JTRS is:
  - Modular, enabling additional capabilities and features to be added to JTR sets
  - Scaleable, enabling additional capacity (bandwidth and channels) to be added to JTR sets
  - Backwards-compatible, communicates with legacy radios
  - Allowing dynamic intra-network and inter-network routing for data transport that is transparent to the radio operator

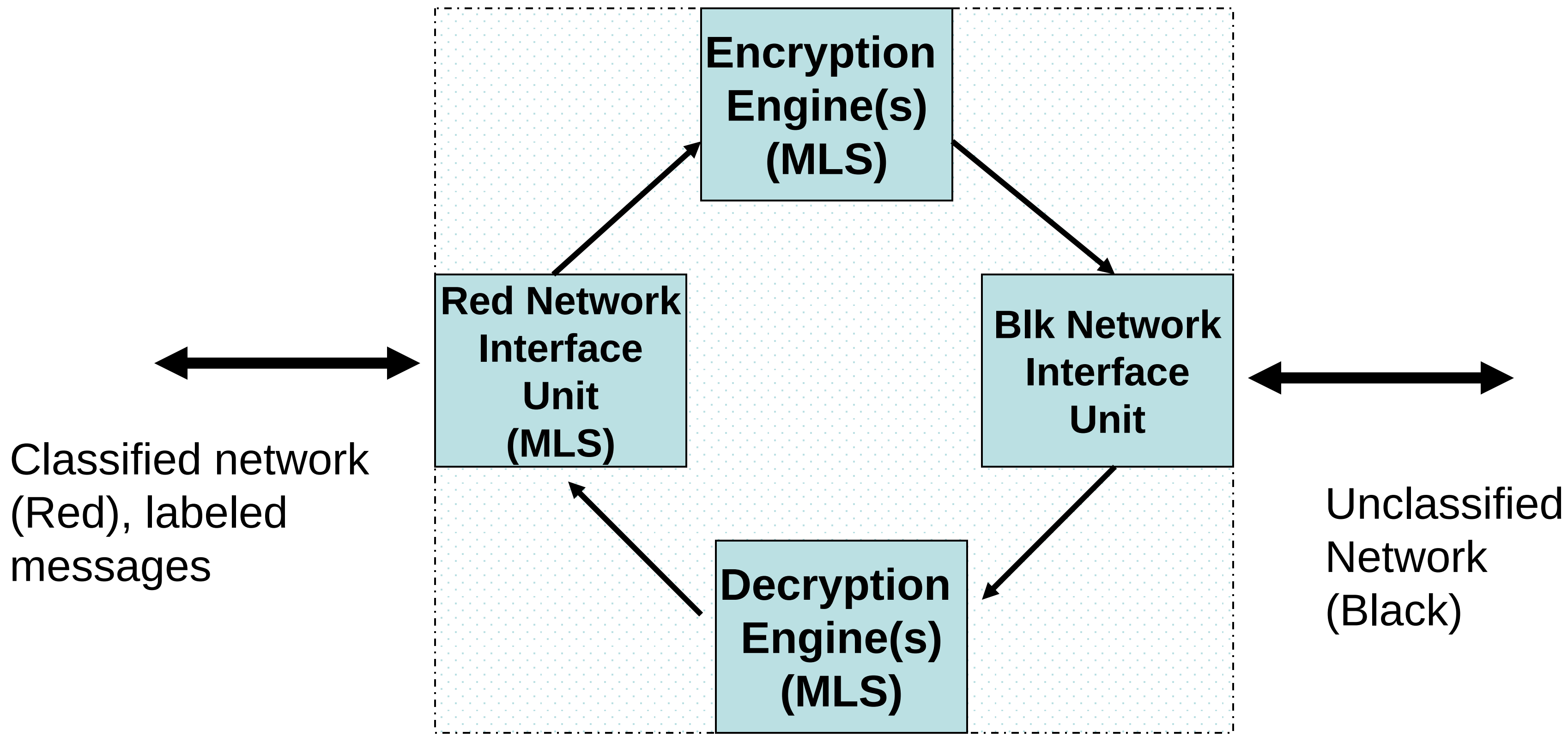
MILS Roadmap
MILS Crypto Engine & Emb OE



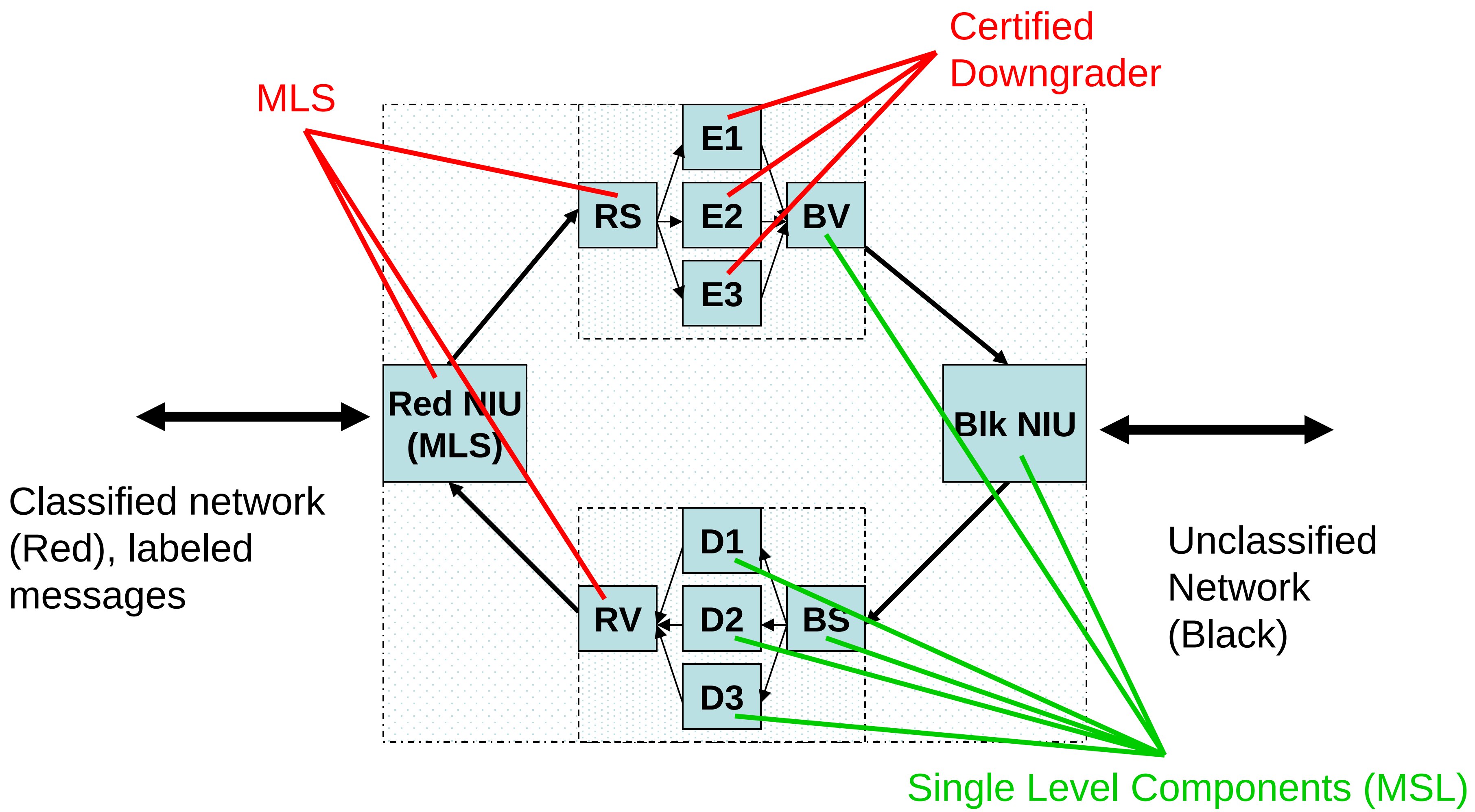
Designing an MLS Component



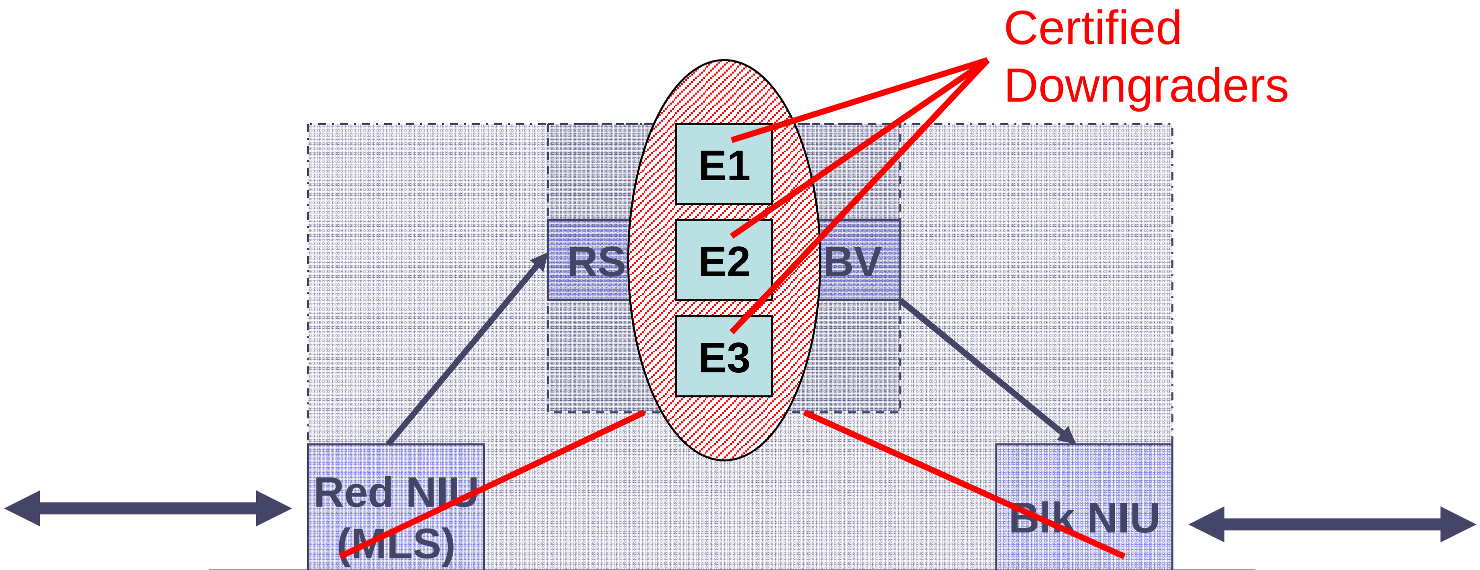
Designing an MLS Component



Designing an MLS Component



Designing an MLS Component



Classified networ
(Red), labeled
messages

Certification Requirements:

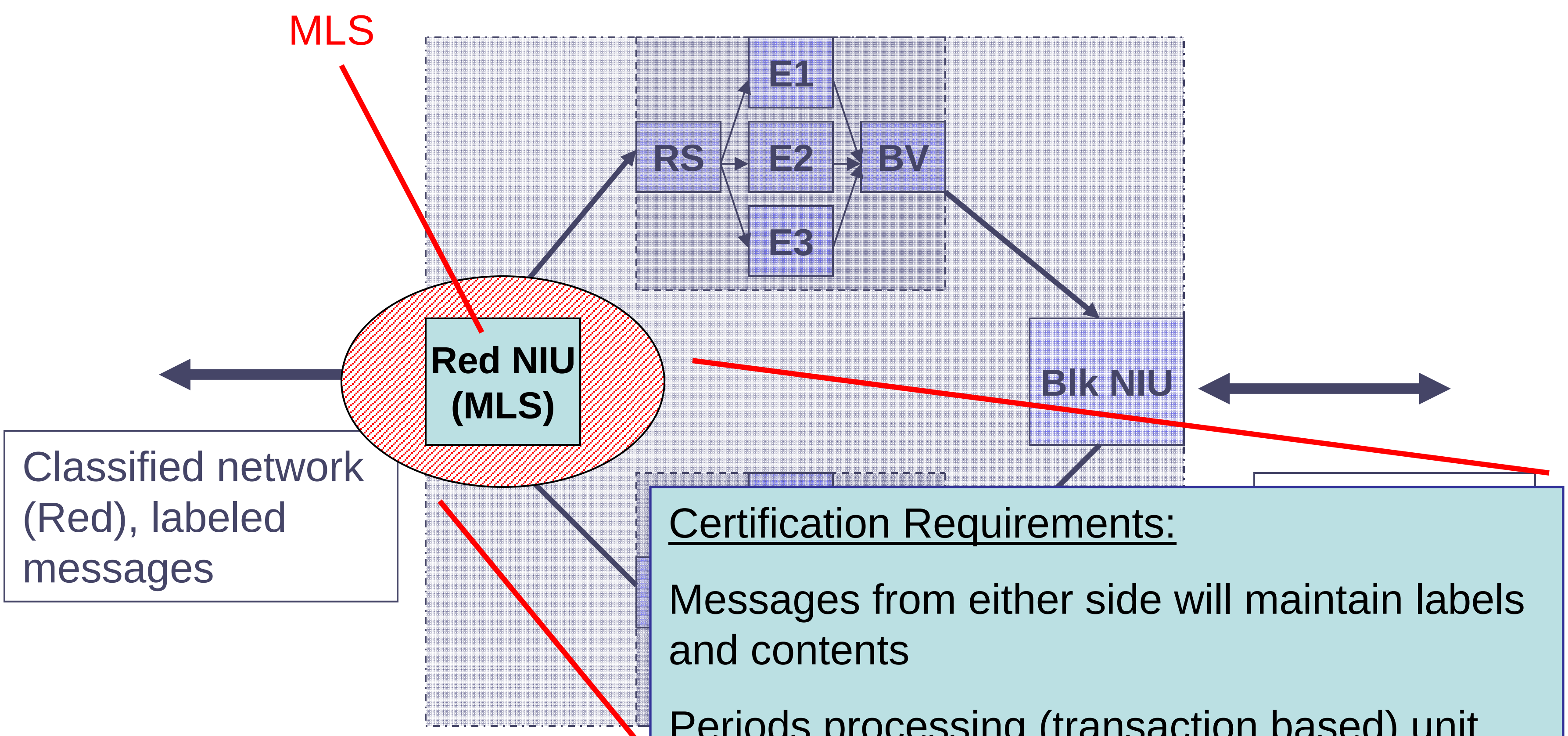
Incoming messages will be encrypted with the specified algorithm and key

Output is strongly encrypted

Each device downgrades from one specific level to unclassified

Unclassified
Network
(Black)

Designing an MLS Component

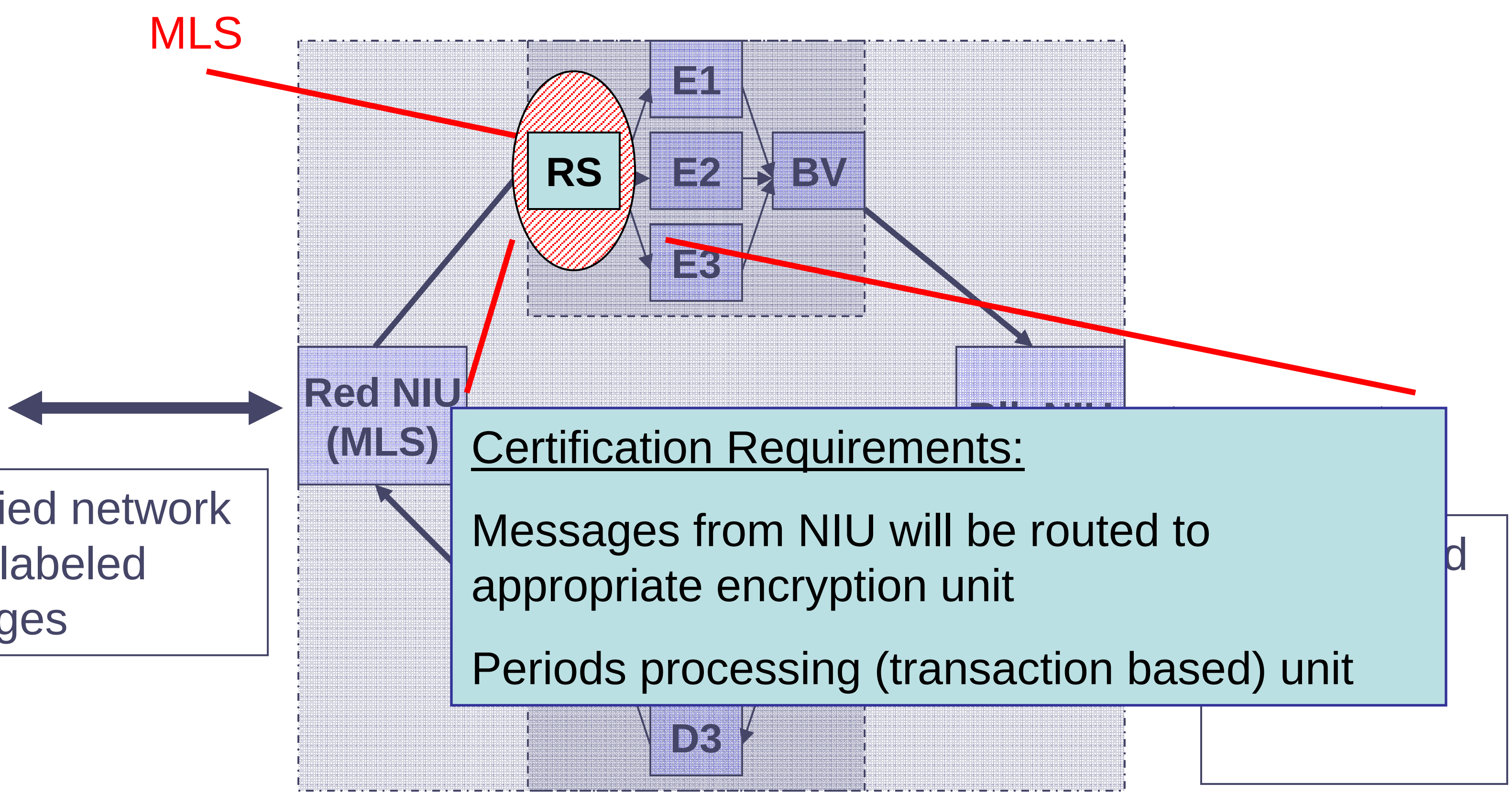


Certification Requirements:

Messages from either side will maintain labels and contents

Periods processing (transaction based) unit

Designing an MLS Component



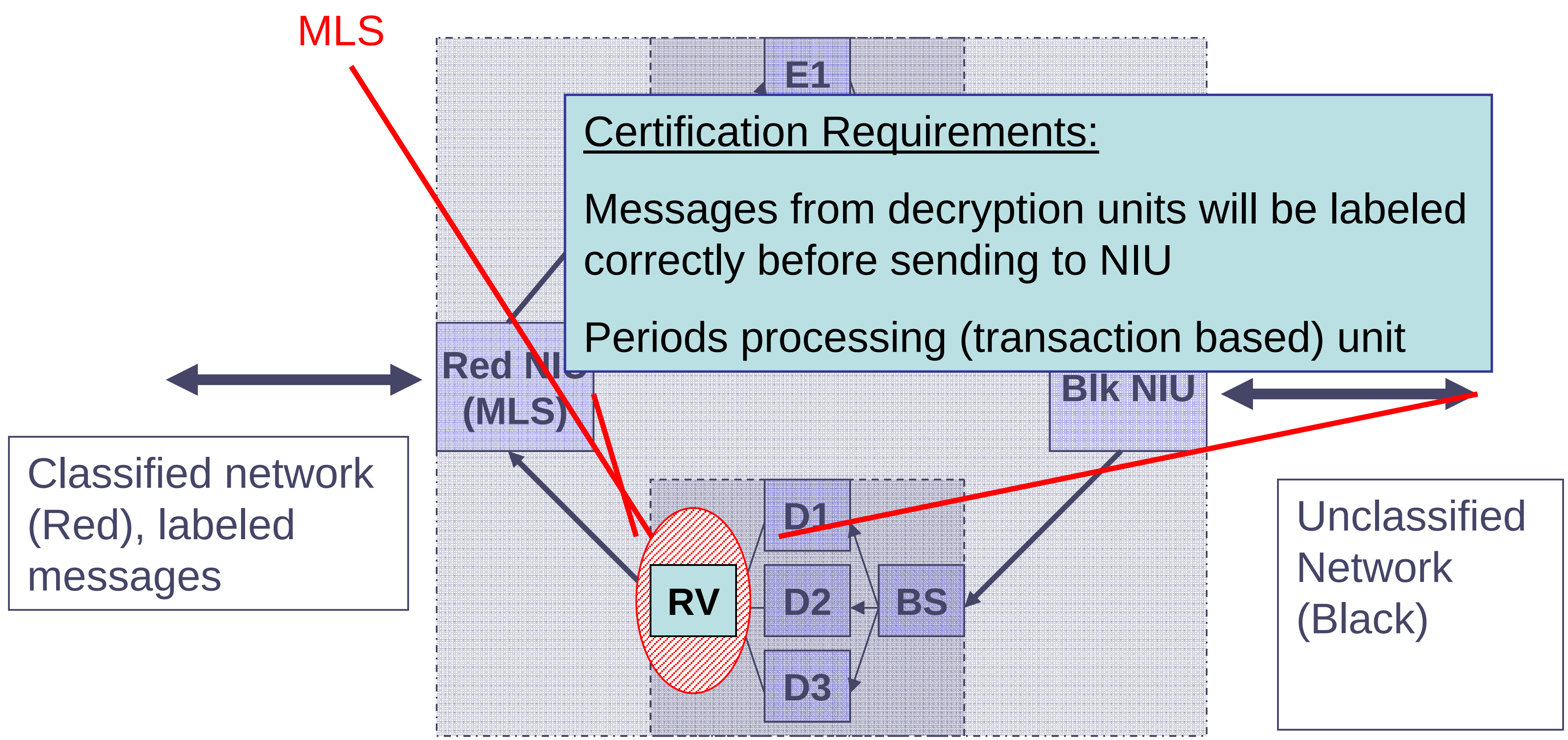
Classified network (Red), labeled messages

<u>Certification Requirements:</u>

Messages from NIU will be routed to appropriate encryption unit

Periods processing (transaction based) unit

Designing an MLS Component



Certification Requirements:

Messages from decryption units will be labeled correctly before sending to NIU

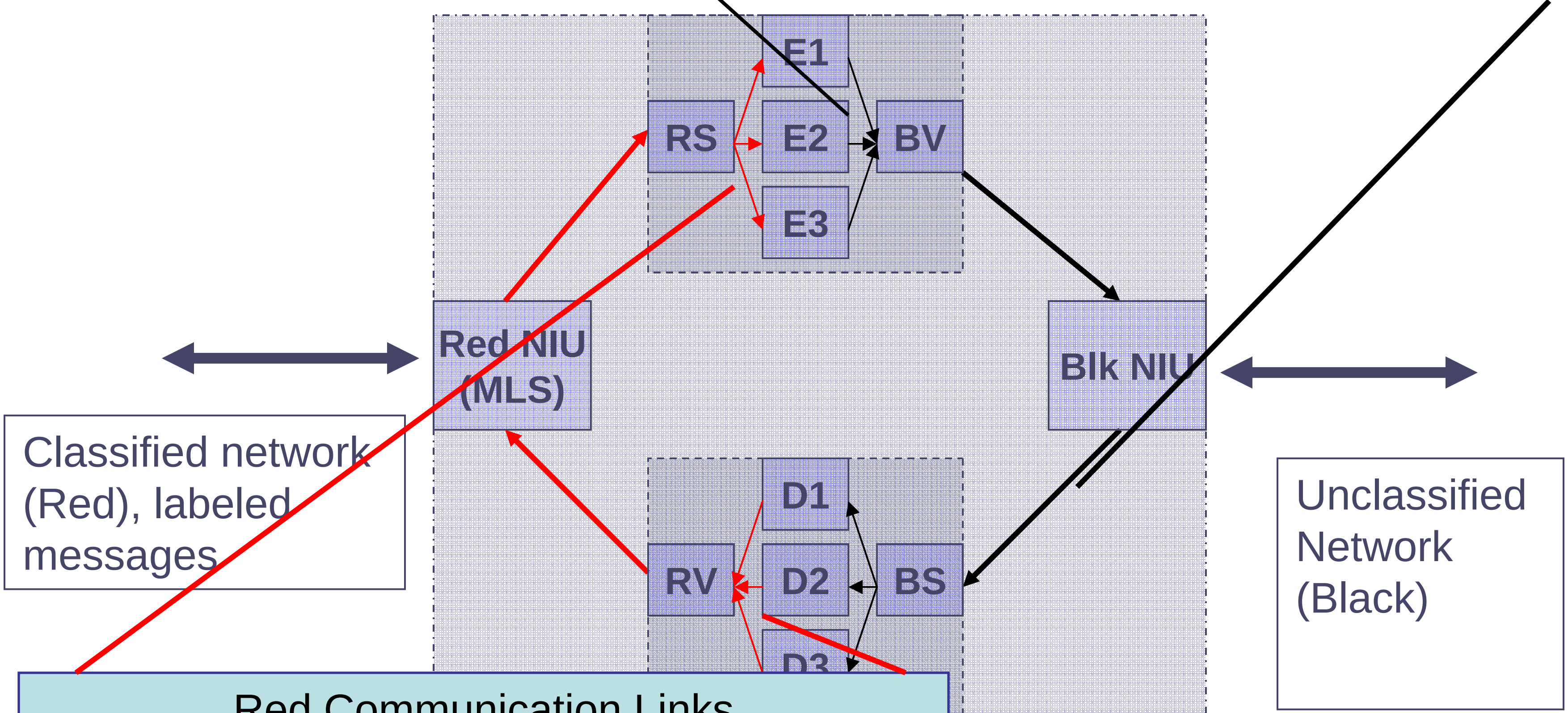
Periods processing (transaction based) unit

Designing

Black Communication Links

Certification Requirements???:

Tamperproof, Non bypassable, Evaluatable



Red Communication Links

Certification Requirements:

Tamperproof, Non bypassable, Evaluatable

The MILS Architecture Approach

- Describe the system in terms of communicating components
  - Designate the clearance of each component and label as MLS or MSL
  - Determine the flow between components with respect to policy
  - Install “boundary firewalls” that manage information up-flow and down-flow
    - these are MLS components

The MILS Architecture Approach

- For each MLS device, determine its type
  - Downgrader – will take data from one security level and send data at a lower level
  - Transaction processor – will process data one message at a time; stateless, may filter data or perform operation on single message
  - Collator – will combine data from many inputs
- Verification of each device may involve additional MILS componentization

Implementation

- Hierarchical Approach
  - Lowest level is separation kernel – enforces isolation, information flow, periods process, damage limitation on a single processor
  - Next level is middleware, to coordinate end-to-end separation
    - Need to create “trusted” components.
      - Verification of the components utilizes architectural support of lower layer
  - Next Level is application specific

Acronyms

MILS	Multiple Independent Levels of Security/Safety
MSLS	Multiple Single Level Security/Safety
MLS	Multi-Level Secure/Safe
PCS	Partition Communication System
CORBA	Common Object Request Broker Architecture
NEAT	Non-bypassable, Evaluatable, Always-invoked,Tamper-proof
NIU	Network Interface Unit
ORB	Object Request Broker
O/S	Operating System
CC	Common Criteria
EAL	Evaluation Assurance Level
ARINC 653	Safety Community Standard for Time and Space Partitioning
DMA	Direct Management Access
MMU	Memory Management Unit

OBJECTIVE INTERFACE

Partners

MILS Hardware Based Partitioning Kernel

AAMP7	**Rockwell Collins**

MILS Software Based Partitioning Kernel

Integrity-178	**Green Hills Software**
LynxOS-178	**LynuxWorks**
VxWorks AE Secure	**Wind River**

MILS Middleware

PCS and ORB*express*	**Objective Interface Systems, Inc.**
MILS TestBed	**University of Idaho**
MILS TestBed	**Naval Post Graduate School**