



*Smart Push, Smart Pull, Sensor
to Shooter in a Multi-Level
Secure/Safe (MLS)
Infrastructure*

Smart Push, Smart Pull, Sensor to Shooter in a Multi-Level Secure/Safe (MLS) Infrastructure

Gordon M. Uchenick

Senior Mentor / Principal Engineer
Objective Interface Systems, Inc.

W. Mark Vanfleet

Senior Mathematician
Senior INFOSEC Security Analyst
National Security Agency

- The Vision: Seamless Data Flow
 - Sensor to Shooter
 - Sensor to Weapon
- Information Assurance Requirements of the Vision
- Design Guidance
- Reference Monitors
- Candidate Technologies
- Multiple Independent Levels of Security

- Each sensor type reveals different information, nominally (source: wikipedia.org):
 - Radar
 - Sonar and other acoustic
 - Infra-red / Thermal imagery
 - HDTV imagery
 - Seismic sensors
 - Magnetic sensors
 - Electronic Support Measures (ESM)
 - Phased Array
- Direct fusion from disparate sources results in better electronic information
 - More accurate
 - More complete
 - More dependable
- Indirect fusion merges electronic information with human input, merging:
 - ELINT: Electronic Intelligence
 - HUMINT: Human Intelligence
 - COMINT: Communications Intelligence
 - SIGINT: Signals Intelligence
 - IMINT: Imagery Intelligence

- Data derived from Direct Fusion (*contrived*)
 - What is it?
 - *T-72 Tank*
 - What is its condition?
 - *Lightly Damaged*
 - Where is it now?
 - *Longitude / Latitude*
 - Where has it been?
 - *Track*
- Characteristics of the Data
 - Multiple sensor devices on a surveillance platform
 - Sensor devices produce giga-gobs of raw data
 - Real-time transmission of all raw sensor data is impractical
 - Direct Fusion likely to be performed on the platform
 - Raw sensor data likely to be TOP SECRET
 - Derived data likely to be SECRET NOFORN
 - Data derived from Direct Fusion shared via **Smart Push**

- Surveillance platforms use SOA to populate MLS Web Server database
 - MLS Web Server database likely to be SECRET NOFORN
- Merged with data about each threat derived from Indirect Fusion:
 - Who controls it?
 - What is its threat potential?
 - What are its intentions?
- Many different types of users need the data:
 - Cleared US Military
 - At various levels
 - Multiple Communities of Interest
 - *Services, Job Titles, etc.*
 - Uncleared US Military in vicinity of the threat
 - Cleared coalition partners
 - At various levels
 - Multiple Communities of Interest for each partner
 - *Canadian Army vs. UK Army Vs. UK Special Air Service*
 - Uncleared coalition partners in vicinity of the threat

- SOA applications query the database searching for threats that meet certain characteristics **Smart Pull**
 - Threat type
 - Threat nationality
 - Proximity to Coalition assets
- When an applicable threat is found, Command and Control personnel are notified **Smart Push**
- The database is “Googled” by a human who makes the decision to prosecute the threat **Smart Pull**
 - Humans make decisions that we would not defer to automation

- Command and Control creates an ad-hoc group of available assets to prosecute the threat
- Ad-hoc task force requires ad-hoc networking for command and control
- Task force comprised of assets from various US services and coalition partners
- Multiple security levels and communities of interest
- Data shared according to security policy
 - Downgraded
 - Guarded
 - Filtered
- After threat prosecution, the task force is dissolved

- Fixed Black IP addresses for Web Servers
 - Communications via Type-1 HAIPE and/or JTRS
- Type-1 Crypto identifies and authenticates registrant
 - Also identifies and authenticates registrant's Domain
- Registrant provides its own Black IP address
 - Also can provide credentials, geo-location, and capabilities
- Red side provides
 - Available services list
 - Red IP addresses for SOA / Web portals
 - Security Policy for information release to other members of the ad-hoc network or other ad-hoc networks

- The Vision: Seamless Data Flow
 - Sensor to Shooter
 - Sensor to Weapon
- Information Assurance Requirements of the Vision
- Design Guidance
- Reference Monitors
- Candidate Technologies
- Multiple Independent Levels of Security

- Controlled Information Flow to users in multiple Security Domains
- Controlled Information Flow requires trustworthy enforcement of appropriate Security Policies.
- Security Policy enforcement must be trustworthy so that the mission is not compromised
 - Even more important, Information Sharing can't be allowed to endanger the Warfighters
 - Information Assurance is all about making sure that the Warfighters' systems can't be used against them.
- Trust is earned, never assumed
 - Certification and Accreditation are the ways to earn Trust.

What identifies a Security Domain?

*Smart Push, Smart Pull, Sensor
to Shooter in a Multi-Level
Secure/Safe (MLS)
Infrastructure*

- Nationality
 - US, Canada, UK, etc
- Classification/Clearance
 - SCI, TS, SECRET, UNCLASSIFIED, etc.
- Community of Interest
 - Functional Organization
- Geo-Location
 - Iraq, Afghanistan, CONUS, the Pentagon, etc.
- Safety
 - Critical, Non-critical, etc.

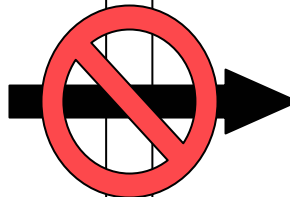
- Cross Domain Server components that enforce the Security Policy
- Downgraders
 - Input: Data at a given classification level
 - Output: Data at a lower classification level
 - Rule Sets
 - Configured for each data stream
 - Field deletion and obfuscation
- Access Control Guards
 - IBAC: Identity Based Access Control
 - RBAC: Role Based Access Control
 - Protocol Specific Access Control
 - CORBA/GIOP
 - DDS
 - HTTP
 - etc.

- Content Guards
 - Document Type Specific Guarding (notional)

▪ .doc	.ppt	.xls
▪ .pdf	.jpg	.mpeg
▪ .xml	.avi	.mov
▪ .html	.mp3	.ps/eps
▪ .tex	.dvi	.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. (U) BACKGROUND	1
A. (U) Administrative Matters	1
1. (U) Appointing Authority	1
2. (U) Brief Description of the Incident	1
B. (U) Constraints and Limitations	2
C. (U) Format of the Report	2
II. (U) ATMOSPHERICS	4
A. (U) Introduction	4
B. (U) Local Security Situation	4
1. (U) Iraq	4
2. (U) Baghdad	4
3. (U) Route Irish	4
C. (U) Known Insurgent Tactics, Techniques, and Procedures	5
1. (U) Methods of Attack	5
2. (U) Insurgent TTPs for IEDs	5
3. (U) Insurgent TTPs for VBIEDs	6
4. (U) Effectiveness of Attacks	7
D. (U) Recent Incidents in the Vicinity of Checkpoint 541	8
E. (U) Unit Experience in the Baghdad Area of Responsibility	8
1. (U) [REDACTED] Division	8
2. (U) [REDACTED] Brigade, [REDACTED] Division	9
3. (U) [REDACTED] Battalion	9
4. (U) [REDACTED] Battalion	10
F. (U) Findings	10
III. (U) TRAFFIC CONTROL POINTS, BLOCKING POSITIONS, AND TRAINING	12
i	
UNCLASSIFIED	



UNCLASSIFIED	
TABLE OF CONTENTS	
I. (U) BACKGROUND	1
A. (U) Administrative Matters	1
1. (U) Appointing Authority	1
2. (U) Brief Description of the Incident	1
B. (U) Constraints and Limitations	2
C. (U) Format of the Report	2
II. (U) ATMOSPHERICS	4
A. (U) Introduction	4
B. (U) Local Security Situation	4
1. (U) Iraq	4
2. (U) Baghdad	4
3. (U) Route Irish	4
C. (U) Known Insurgent Tactics, Techniques, and Procedures	5
1. (U) Methods of Attack	5
2. (U) Insurgent TTPs for IEDs	5
3. (U) Insurgent TTPs for VBIEDs	6
4. (U) Effectiveness of Attacks	7
D. (U) Recent Incidents in the Vicinity of Checkpoint 541	8
E. (U) Unit Experience in the Baghdad Area of Responsibility	8
1. (U) Third Infantry Division	8
2. (U) Second Brigade, 10 th Mountain Division	9
3. (U) 1-69 Infantry Battalion	9
4. (U) 1-76 Field Artillery Battalion	10
F. (U) Findings	10
III. (U) TRAFFIC CONTROL POINTS, BLOCKING POSITIONS, AND TRAINING	12
i	
UNCLASSIFIED	

- “Googler” Characteristics
 - Nationality
 - Clearance
 - Job Title
 - Location
- Threat Characteristics
 - Classification of the Threat(s)
 - Location of the Threat
- Security Policies
 - Releasability of Threat Data
 - Down Grade Policy

- Requires anticipation of the unauthorized events that the system must prevent
 - e.g., No SECRET cleared users allowed to read TOP SECRET information
- System Security Policy usually consists of a collection of sub-policies which define the security services offered by the system.
- Example sub-policy: User Access Control
 - “A correct user name, password, and fingerprint must be entered into the system prior to user access”

- Notional Security Policy for Information Flows
 - P1A: There shall be no infiltration of data among flows
 - P1B: There shall be no infiltration of data within flows
 - P2A: There shall be no exfiltration among flows
 - P2B: There shall be no exfiltration within flows
 - P3: There shall be no unauthorized use of authorized flows
 - Example: No third party is allowed to cause information belonging to “A” to flow to “B” even if the security policy allows “A” to communicate with “B”
- Applicable to Security Enforcing components
 - HAIPE
 - JTRS
 - PCS
 - Etc.

- High Robustness is, in general, equivalent to Common Criteria EAL6+
 - There is no official definition of High Robustness yet.
 - Working definition in SKPP V0.71 (draft)
- DCID 6/3 applies to all entities that process, store, or communicate intelligence information
 - An information system operates at Protection Level 5 when at least one user lacks any clearance for access to some of the information in that system
- DO-178B applies to software for airborne systems and equipment.
 - Software that can cause a catastrophic failure is certified at Level A
- There is significant overlap and synergy among these standards

- Interdomain Security Policy Management
 - How do we define it?
 - How do we update it?
 - How do we distribute it
- Domain Policy Management
 - How do we include a new actor into a domain?
 - How do we revoke privileges of an actor?
 - How do we detect and exclude a compromised actor?
- Threat-based Domain construction and destruction
 - Multilevel
 - Multinational
 - Multiple COIs

- Transparency
 - Warfighters are supposed to expend their resources on fighting wars, not enforcing security policies
 - If it is too hard to follow, nobody will follow it
 - “Get the job done” attitude
 - If it is too hard to administer, nobody will administer it
 - Security can be compromised

- The Vision: Seamless Data Flow
 - Sensor to Shooter
 - Sensor to Weapon
- Information Assurance Requirements of the Vision
- Design Guidance
- Reference Monitors
- Candidate Technologies
- Multiple Independent Levels of Security

- Bell-LaPadula to focus on Confidentiality
 - Read Down, Write Up
 - Protects against unauthorized disclosure
- Biba to focus on Integrity
 - Read Up, Write Down
 - Protects against unauthorized modification
- Other security policies:
 - Brewer-Nash (access control)
 - Information flow model provides controls to mitigate conflict of interest
 - Clark-Wilson (integrity)
 - Well formed transactions transition system from one secure state to another
 - Graham-Denning (rights)
 - Define rights on how subjects execute security functions on objects

- Identify the unauthorized events that the system must prevent
- Typically, systems must protect against:
 - Unauthorized Disclosure
 - *Confidentiality*
 - Unauthorized Modification
 - *Integrity*
 - Unauthorized Access
 - *Access Control*
 - Masquerade or Replay
 - *Authentication*
 - Denial of Transmission or Reception
 - *Non-repudiation*
 - Denial of Service
 - *Availability*

- Input: System Security Policy
- Input: Unauthorized Events
- Use these inputs to derive a list of requirements which the system must meet
- Result: A written System Requirements Document (SRD)
- When dealing with classified data, seek NSA IAD guidance
 - Engage them **EARLY**
 - Engage them **OFTEN**

Step 1: Assess Information Value

*Smart Push, Smart Pull, Sensor
to Shooter in a Multi-Level
Secure/Safe (MLS)
Infrastructure*

- Consult the *Information Assurance Technical Framework*
 - Best practices document, available on <http://iatf.net>
- Value assessed by evaluation the consequences of security policy violation with respect to:
 - Security
 - Safety
 - Financial Posture
 - Infrastructure
- The IATF identifies five levels:
 - **V1:** Negligible effect
 - **V2:** Minimal Damage
 - **V3:** Some Damage
 - **V4:** Serious Damage
 - **V5:** Exceptionally Grave Damage

Step 2: Determine Threat Levels

*Smart Push, Smart Pull, Sensor
to Shooter in a Multi-Level
Secure/Safe (MLS)
Infrastructure*

- Best practices also in the IATF
- Threats are ranked by assessing:
 - Capability
 - Resources
 - Motivation
 - Risk Willingness
- The IATF identifies seven levels:
 - **T1:** Inadvertent or accidental events
Tripping over a power cord
 - **T2:** Minimal resources – willing to take little risk
Passive, casual eavesdropper
 - **T3:** Minimal resources – willing to take significant risk
Unsophisticated hacker
 - **T4:** Moderate resources – willing to take little risk
*Organized crime, sophisticated hacker,
international corporations*
 - **T5:** Moderate resources – willing to take significant risk
International terrorists
 - **T6:** Abundant resources – willing to take little risk
*Well funded national laboratory,
nation-state, international corporation*
 - **T7:** Abundant resources – willing to take significant risk
Nation-states in time of crisis

- Confidentiality
 - Encryption algorithms
- Integrity
 - Hashing algorithms
- Access Control
 - Identification and Authentication
- Authentication
 - Certificates
- Non-repudiation
 - Digital Signatures
- Availability
 - Redundancy

Step 4: Strength and Assurance Level

*Smart Push, Smart Pull, Sensor
to Shooter in a Multi-Level
Secure/Safe (MLS)
Infrastructure*

- From the IATF, Strength of Mechanism and Assurance Level mapped to Information Value and Threat Level

Information Value	Threat Levels						
	T1	T2	T3	T4	T5	T6	T7
V1	SML1 EAL1	SML1 EAL1	SML1 EAL1	SML1 EAL2	SML1 EAL2	SML1 EAL2	SML1 EAL2
V2	SML1 EAL1	SML1 EAL1	SML1 EAL1	SML2 EAL2	SML2 EAL2	SML2 EAL3	SML2 EAL3
V3	SML1 EAL1	SML1 EAL2	SML1 EAL2	SML2 EAL3	SML2 EAL3	SML2 EAL4	SML2 EAL4
V4	SML2 EAL1	SML2 EAL2	SML2 EAL3	SML3 EAL4	SML3 EAL5	SML3 EAL5	SML3 EAL6
V5	SML2 EAL2	SML2 EAL3	SML3 EAL4	SML3 EAL5	SML3 EAL6	SML3 EAL6	SML3 EAL7

- Architecture Policy: INFOSEC boundaries shall be designed using the Principle of Least Privilege
- Principle of Least Privilege: Each subject is granted only the most restrictive set of privileges (or clearance) needed to perform its authorized tasks
 - Minimum memory footprint
 - Only what is needed and ***nothing more***
 - Minimum hardware features
 - Smallest capability set and ***nothing more***
 - Minimum invocation of rights
 - Only necessary privileges ***only when needed***
 - Maximum separation
 - Necessary data disclosed and ***nothing more***

Step 6: Utilize the Common Criteria

*Smart Push, Smart Pull, Sensor
to Shooter in a Multi-Level
Secure/Safe (MLS)
Infrastructure*

- Utilize the Functional Requirements in Part 2 to help define the system and meet the System Requirements Document
- Utilize the Assurance Requirements in Part 3
 - Configuration Management
 - Delivery and Operation
 - Development
 - Guidance Documents
 - Testing
 - Life Cycle Support
 - Vulnerability Assessment
 - Maintenance of Assurance

- Use a defined/structured process (e.g., SEI/CMMI)
 - Produce software that does only its intended task and is evaluable
 - NSA requires at least CMMI Level 3
- For software that is not security enforcing or security relevant
 - Develop the code with good quality control techniques, in small, well-structured units, and thoroughly test it

- Code that is Security Enforcing or Security Relevant
- Develop the code from an abstract finite state machine (when it makes sense)
- Use formal tools (e.g. model checkers) to evaluate the state machine and other critical code
- Develop a mapping between the state machine and the code
- Boot process, with digitally signed copies of ALL software running on the system, should be stored in the system on ROM and protected accordingly
- Meet ALL Non-Trusted development requirements

- The Vision: Seamless Data Flow
 - Sensor to Shooter
 - Sensor to Weapon
- Information Assurance Requirements of the Vision
- Design Guidance
- Reference Monitors
- Candidate Technologies
- Multiple Independent Levels of Security

- Common Criteria Definition (Version 2.2, Part 1, page 14)
 - The concept of an abstract machine that enforces TOE access control Policies
- The enforcement point for the Security Policy
- The Reference Monitor is *not* always a software module
- The Reference Monitor is an abstraction
- The best Reference Monitor is no Reference Monitor
 - Because the design of the system itself makes violation of the Security Policy impossible
 - (e.g., separation by air gap)
 - It isn't always practical, affordable, or achievable to design systems that way
 - Potentially user unfriendly
 - Cost
 - Size, Weight, and Power

To be effective, Security Policy Enforcement must be:

- **Non-bypassable**
 - Security functions cannot be circumvented
- **Evaluatable**
 - Security functions are small enough and simple enough for mathematical verification
- **Always Invoked**
 - Security policy is enforced each and every time
- **Tamperproof**
 - Subversive or errant code cannot alter the security data or functions

- Reference Monitor is the heart of the TOE Security Function (TSF)
 - TSF: TOE Security Function
 - TOE: Target of Evaluation
- Common Criteria class FPT: Protection of the TSF
- Decomposed into:

AMT	Underlying abstract machine test	RPL	Replay detection
FLS	Fail Secure	RVM	Reference mediation
ITA	Availability of exported TSF data	SEP	Domain separation
ITC	Confidentiality of exported TSF data	SSP	State synchrony protocol
ITI	Integrity of exported TSF data	STM	Time stamps
ITT	Internal TSF data transfer	TDC	Inter-TSF data consistency
PHP	TSF physical protection	TRC	Internal TOE TSF data replication consistency
RCV	Trusted Recovery	TST	TSF self test

- The Vision: Seamless Data Flow
 - Sensor to Shooter
 - Sensor to Weapon
- Information Assurance Requirements of the Vision
- Design Guidance
- Reference Monitors
- **Candidate Technologies**
- Multiple Independent Levels of Security

- All security is policy is performed by the security kernel
 - Originally for performance reasons
 - No other was to ensure enforcement is non-bypassable
- As security policy becomes more complex:
 - Code grows in security kernel
 - Certification efforts become unmanageable
 - Evaluatability of kernel code decreases
 - Maintainability of kernel code decreases
 - Policy decisions can be based on incomplete or unauthenticated information

Monolithic Security Kernel

Smart Push, Smart Pull, Sensor
to Shooter in a Multi-Level
Secure/Safe (MLS)
Infrastructure

Monolithic Applications

**User
Mode**

**Monolithic
Application
Extensions**

Monolithic Kernel

Network I/O

Information Flow

Data isolation

Auditing

DAC

MAC

Device drivers

**Privilege
Mode**

Fault Isolation

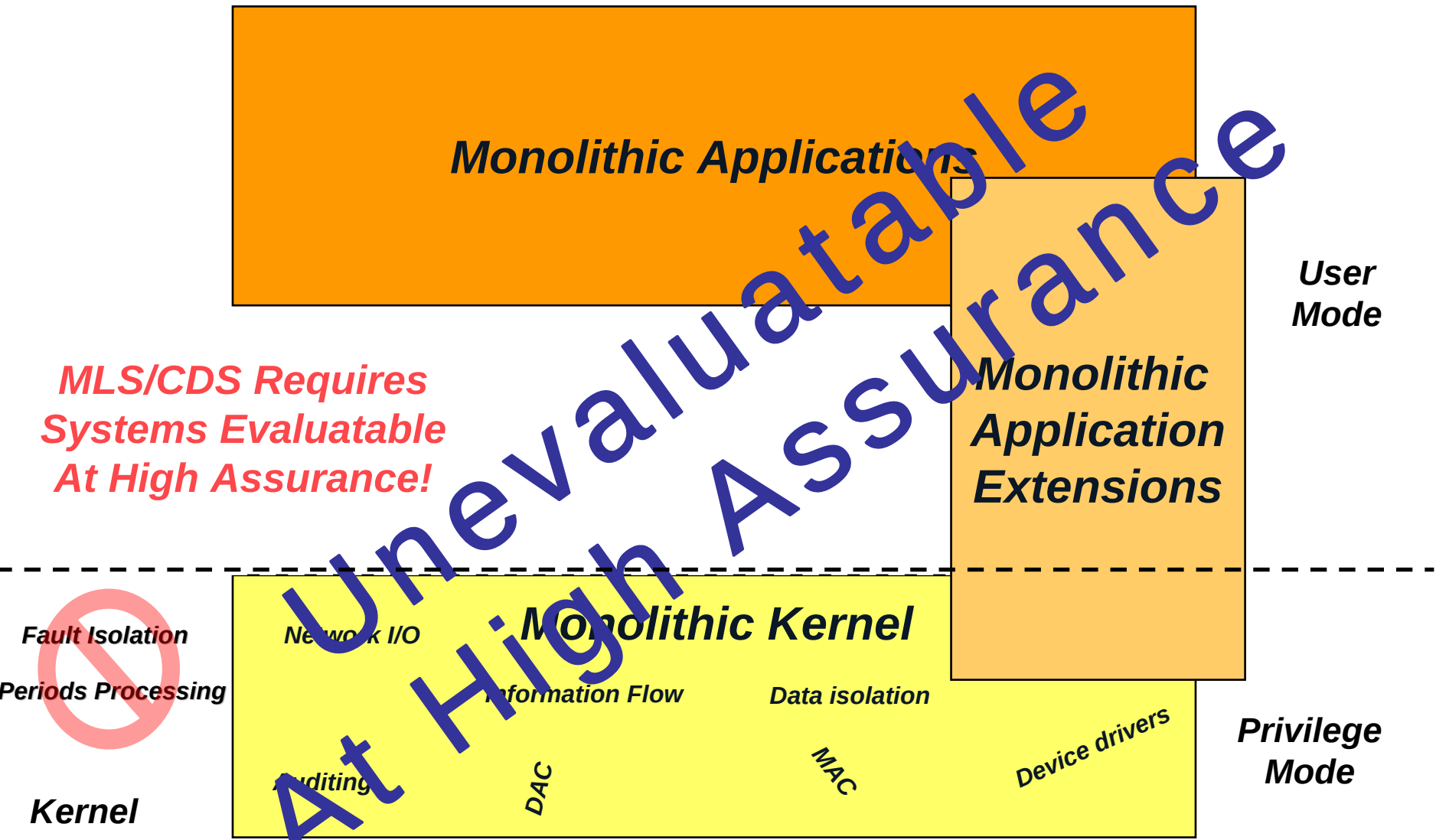
Periods Processing

Kernel

- Most commercial computer security architectures
 - The result of systems software where security was an afterthought
 - Operating systems
 - Communications architectures
 - **Reactive** response to problem
 - 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

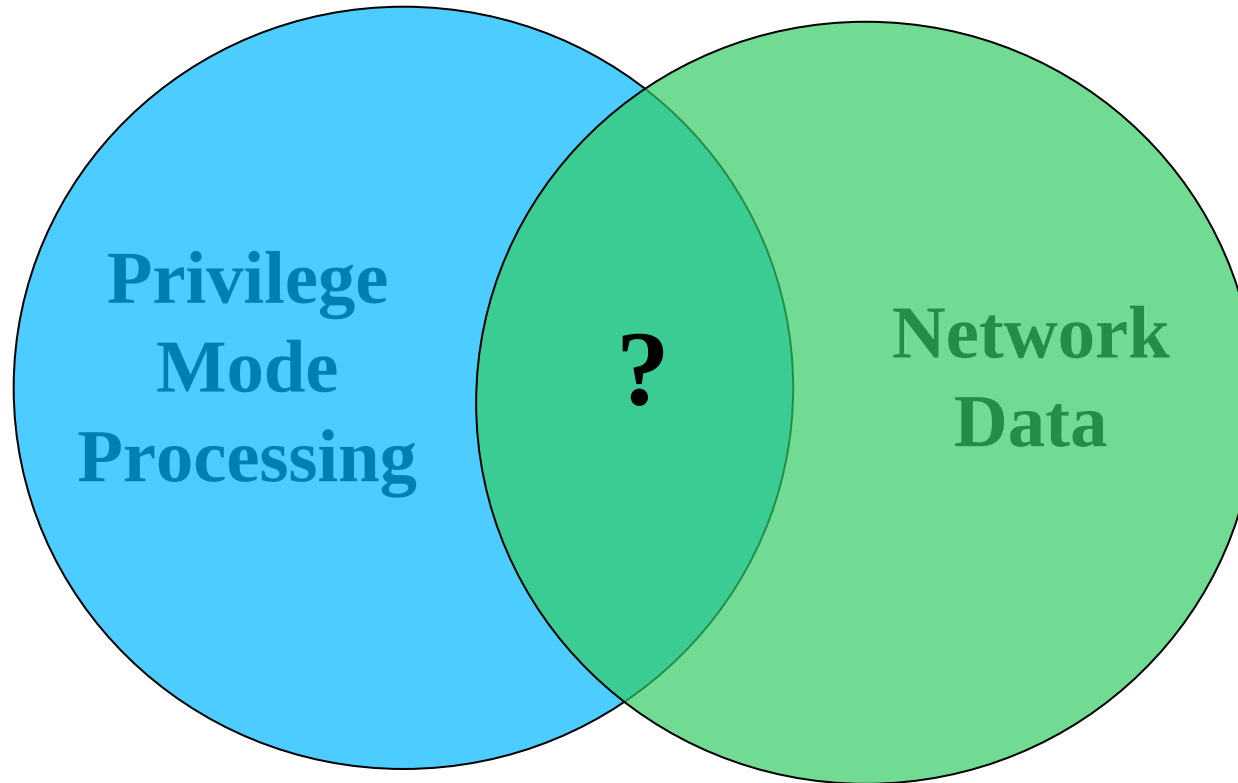
High Assurance Monolithic Kernel?

Smart Push, Smart Pull, Sensor
to Shooter in a Multi-Level
Secure/Safe (MLS)
Infrastructure

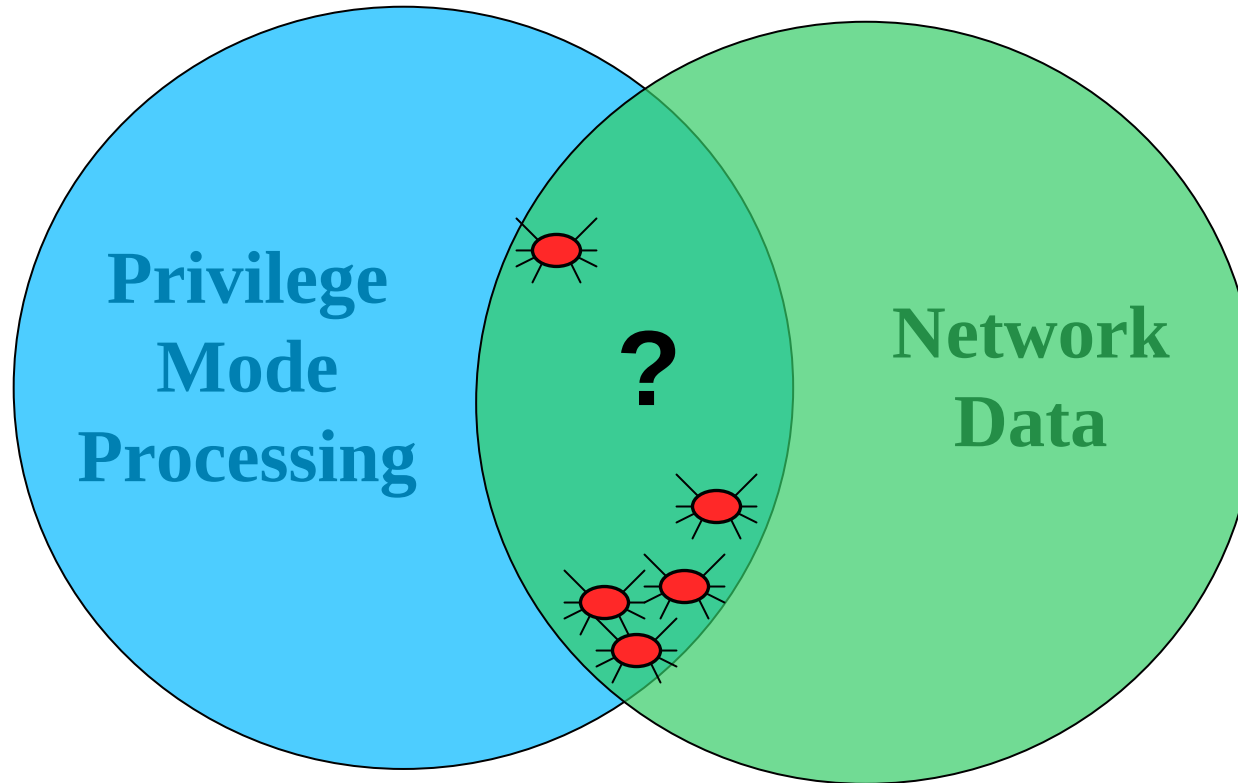


MLS/CDS Requires
Systems Evaluatable
At High Assurance!

Unevaluatable
At High Assurance



**What happens when network headers
are processed in privilege mode?**

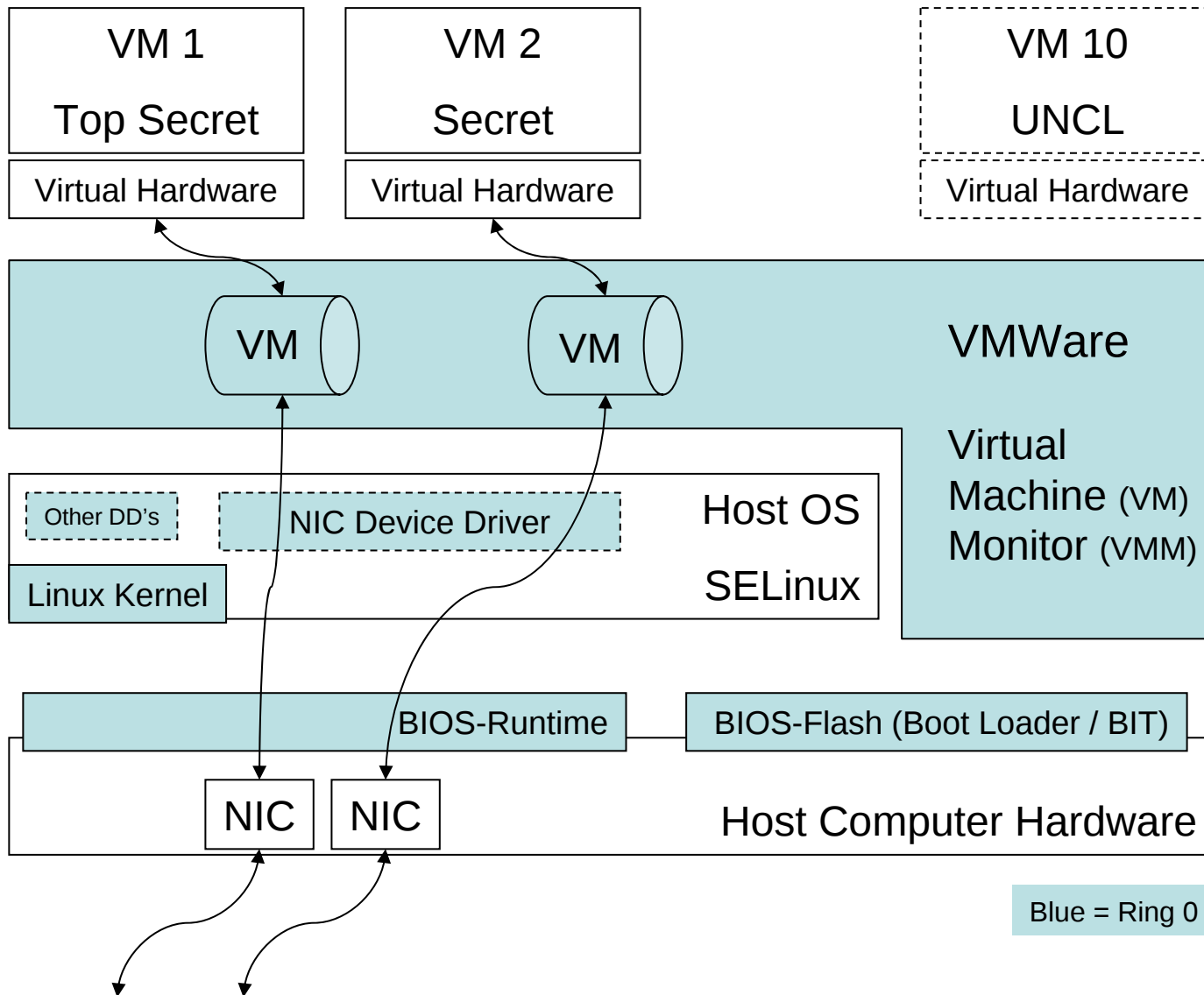


Wild Creatures of the Net: Worms, Virus, . . .

- Developed by NSA “R” Group for internal use, later licensed for unlimited distribution by HP and TCS
- Assembled from readily available software components
 - Device drivers from SELinux
 - Separation from VMware®
 - Virtual machines run Windows® or Linux®
 - Virtual machines communicate via virtual NICs
- Originally approved by the NSA for internal use to provide separation of TOP SECRET from SECRET without respect to compartments or need to know, only for users with TOP SECRET clearance
 - Intended to connect internal NSANet (TS) to SIPRNET (S) for users with TS clearance
- Accredited by NSA to run in DCID 6/3 PL4 environments
 - Extends original certification to allow users with Secret clearances

NetTop Architecture

*Smart Push, Smart Pull, Sensor
to Shooter in a Multi-Level
Secure/Safe (MLS)
Infrastructure*



- Readily available on generic PC hardware
 - A desktop solution, no plans for embedded support
 - Not applicable to weapon systems or platforms
- Meets NSTISSP-11 validation requirements
 - Not certified via CCEVS (NIAP)
 - CCRA not applicable
- Applicable to low threat environments
 - Trusted people in secure facilities
- Provides a moderately robust level of separation
 - COTS components do not meet least privilege high robustness design requirements

- The Vision: Seamless Data Flow
 - Sensor to Shooter
 - Sensor to Weapon
- Information Assurance Requirements of the Vision
- Design Guidance
- Reference Monitors
- Candidate Technologies
- Multiple Independent Levels of Security

- 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.

Separation Kernel

- The only code that runs in privileged mode
- **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, ...

Really very simple:

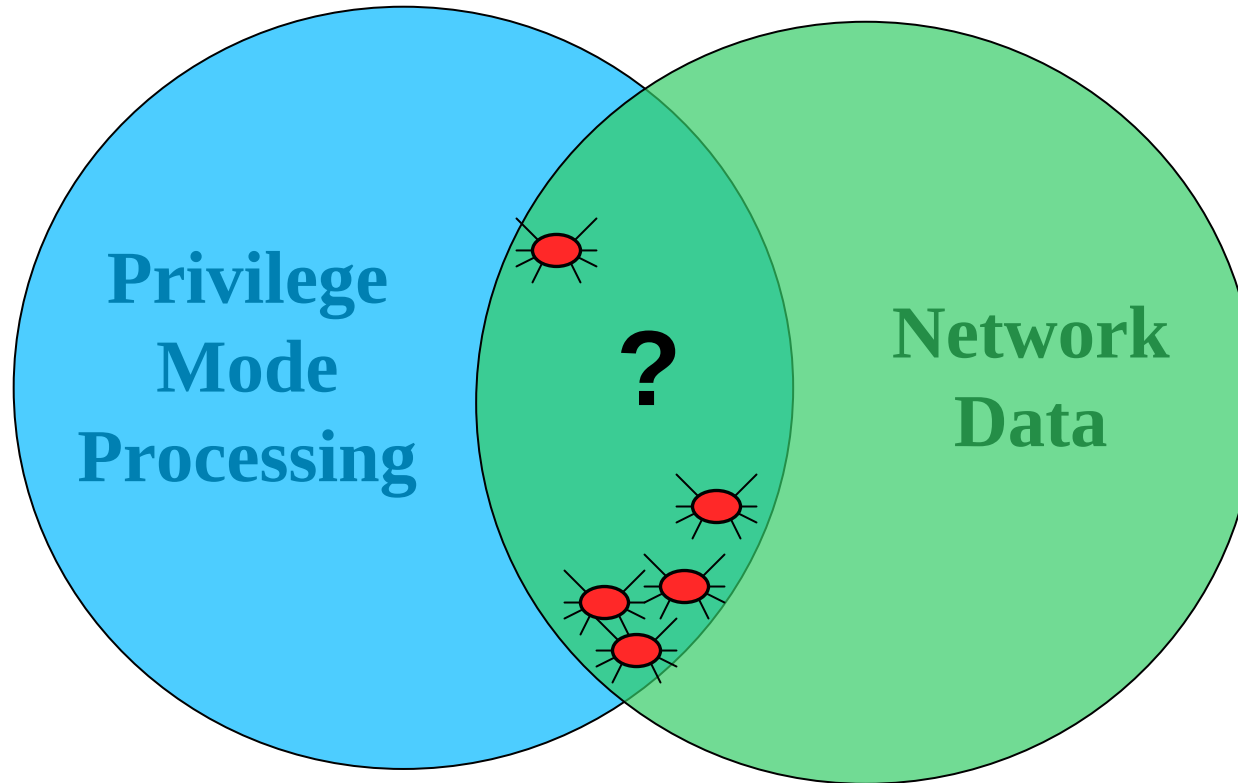
- Dramatically **reduce the amount of**
security critical code

So that we can

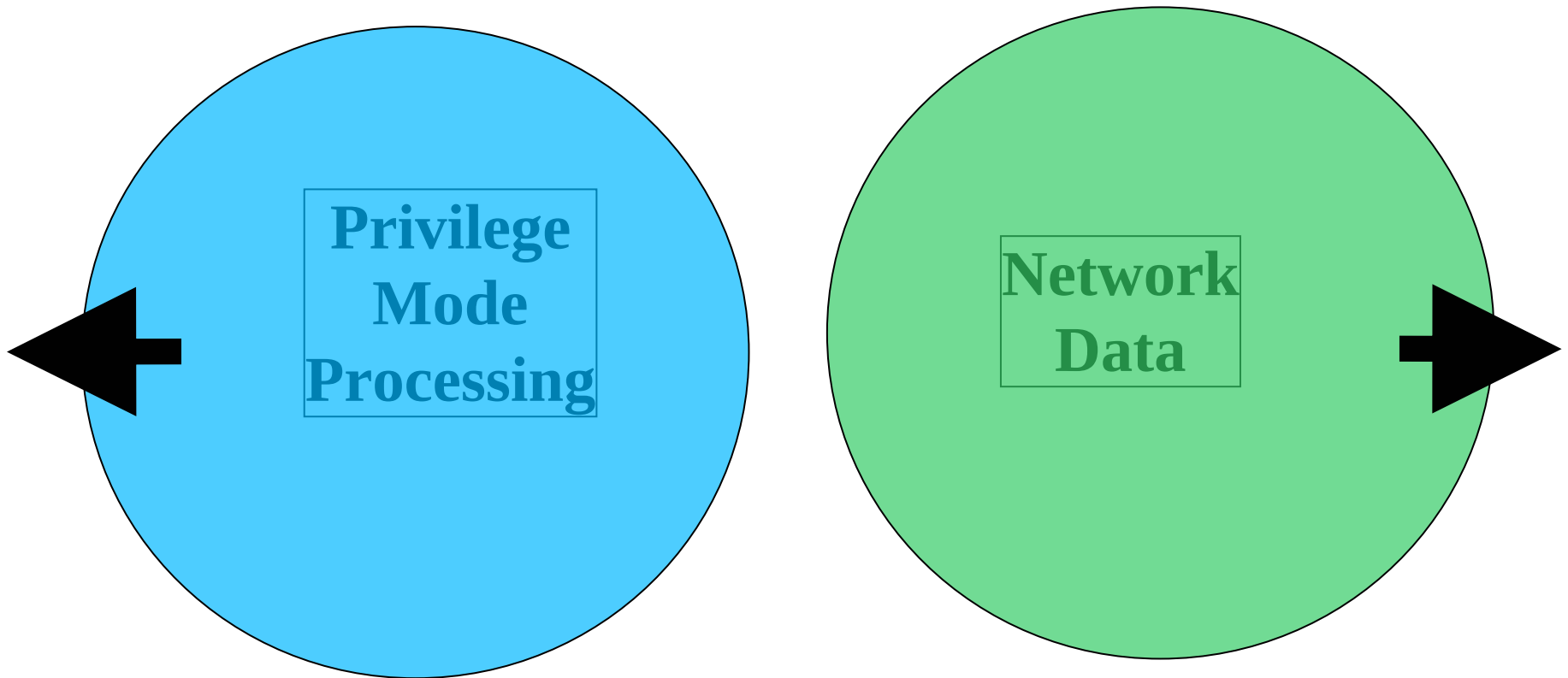
- Dramatically **increase the scrutiny of**
security critical code

To make

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



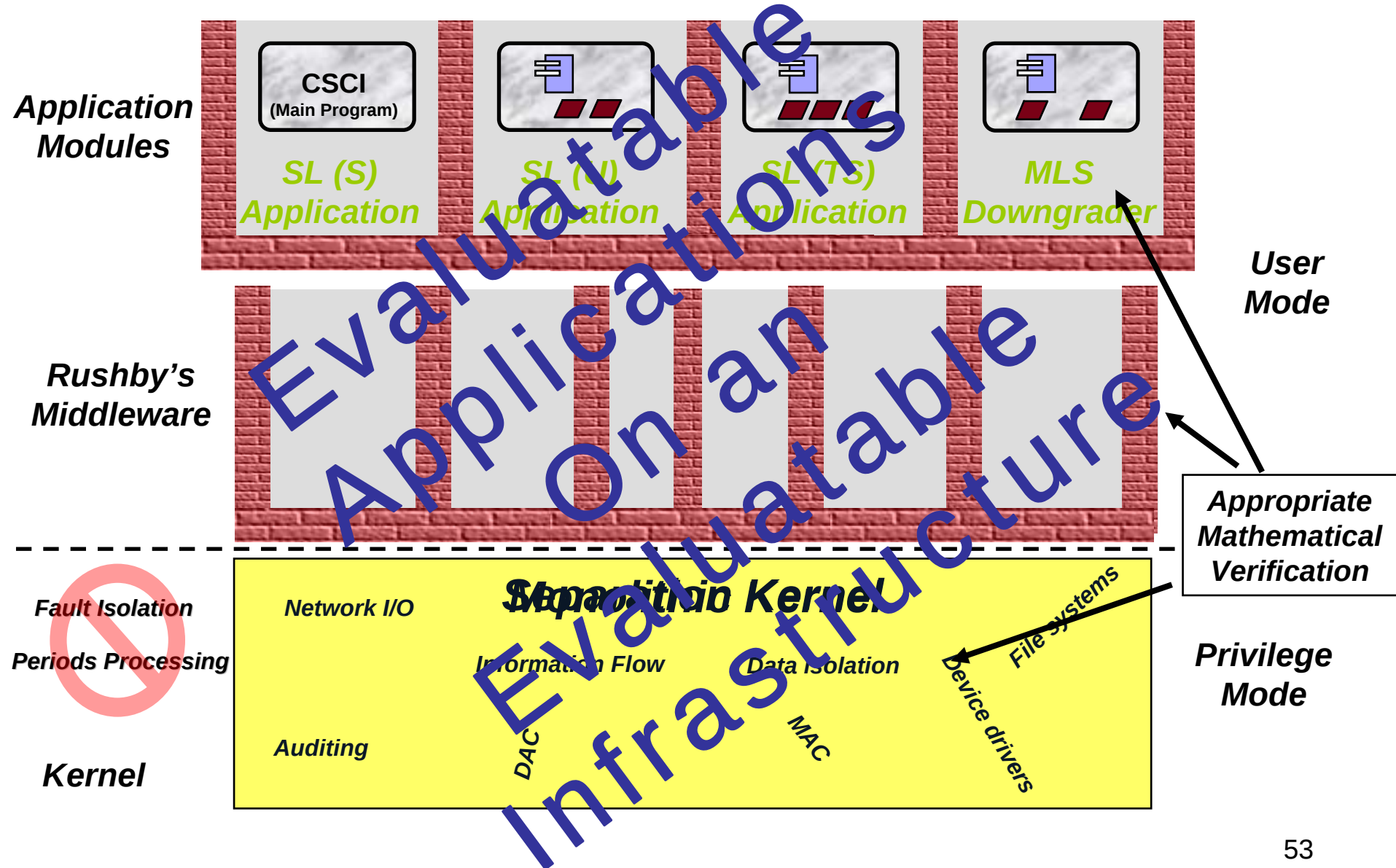
Wild Creatures of the Net: Worms, Virus, . . .

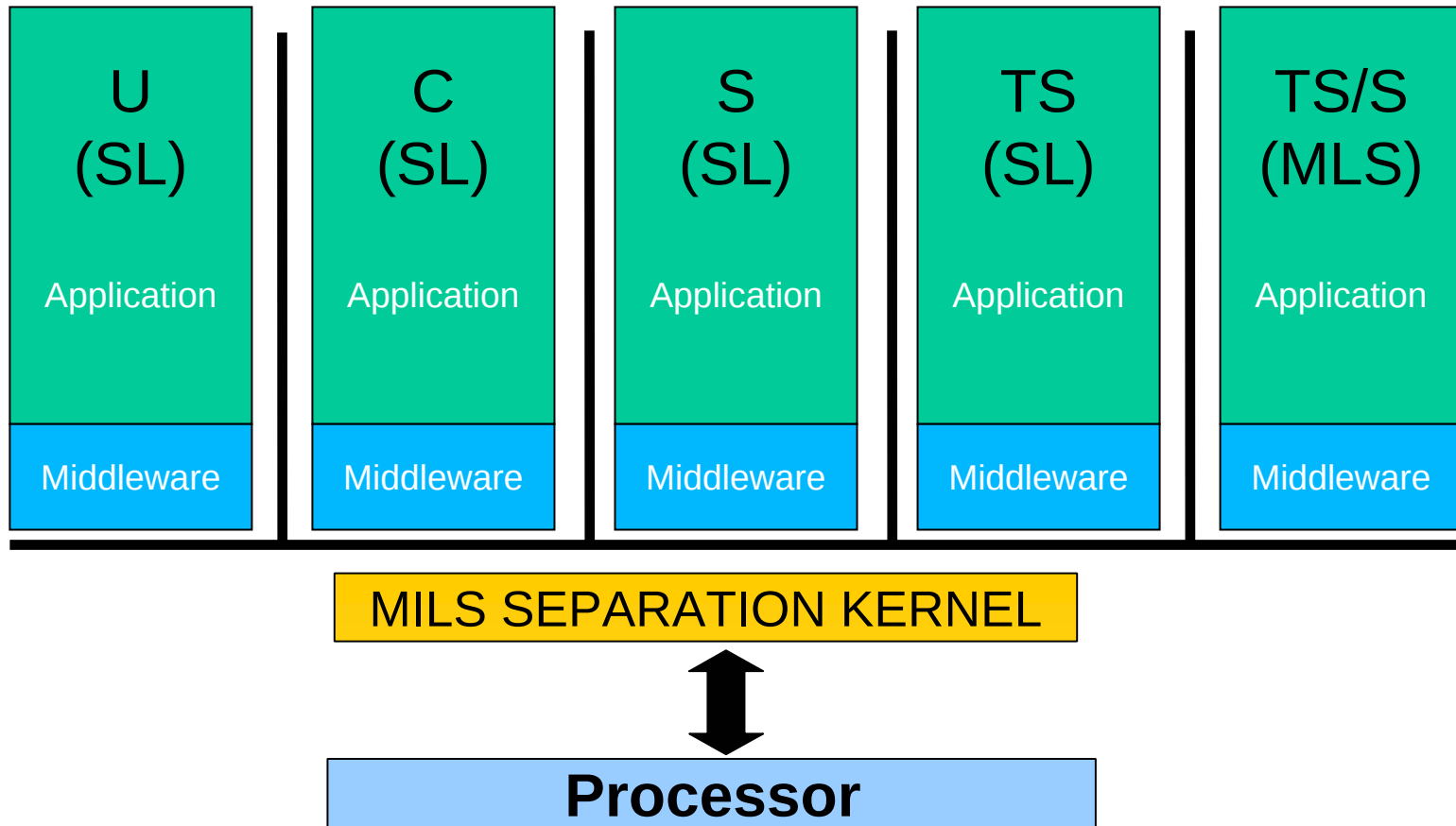


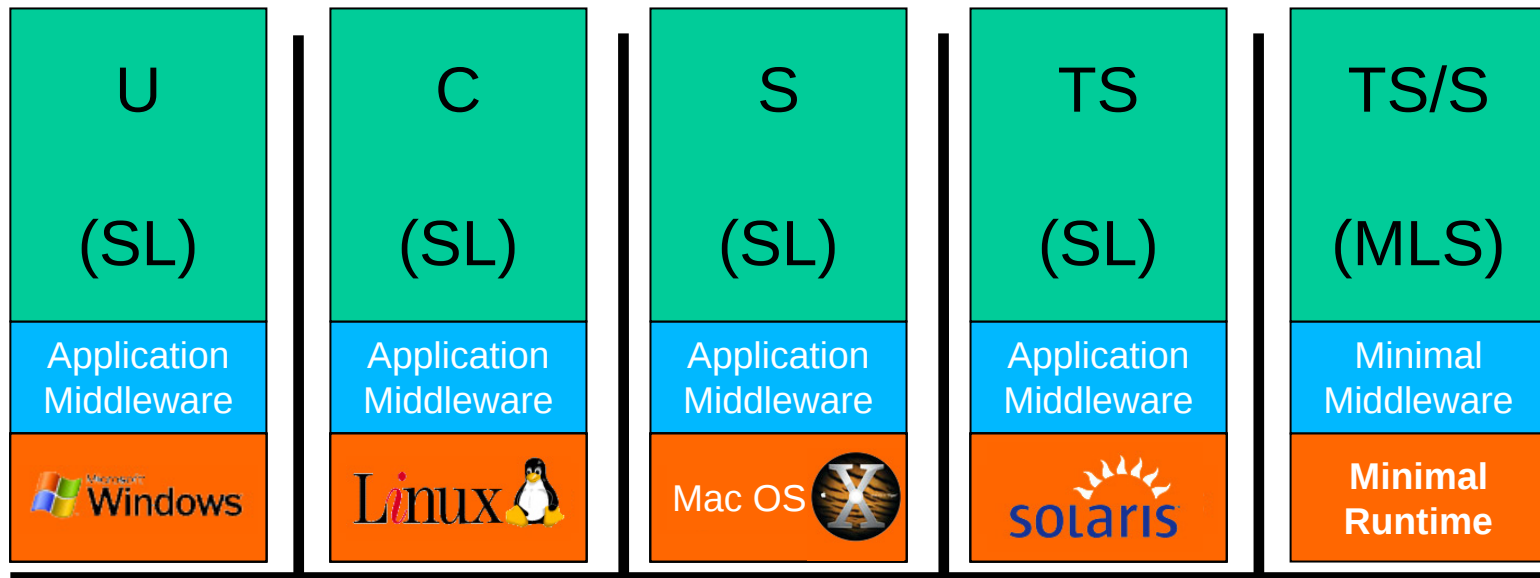
**Under MILS, network header and
privilege mode processing are separated**

MILS Architecture Evolution

Smart Push, Smart Pull, Sensor
to Shooter in a Multi-Level
Secure/Safe (MLS)
Infrastructure







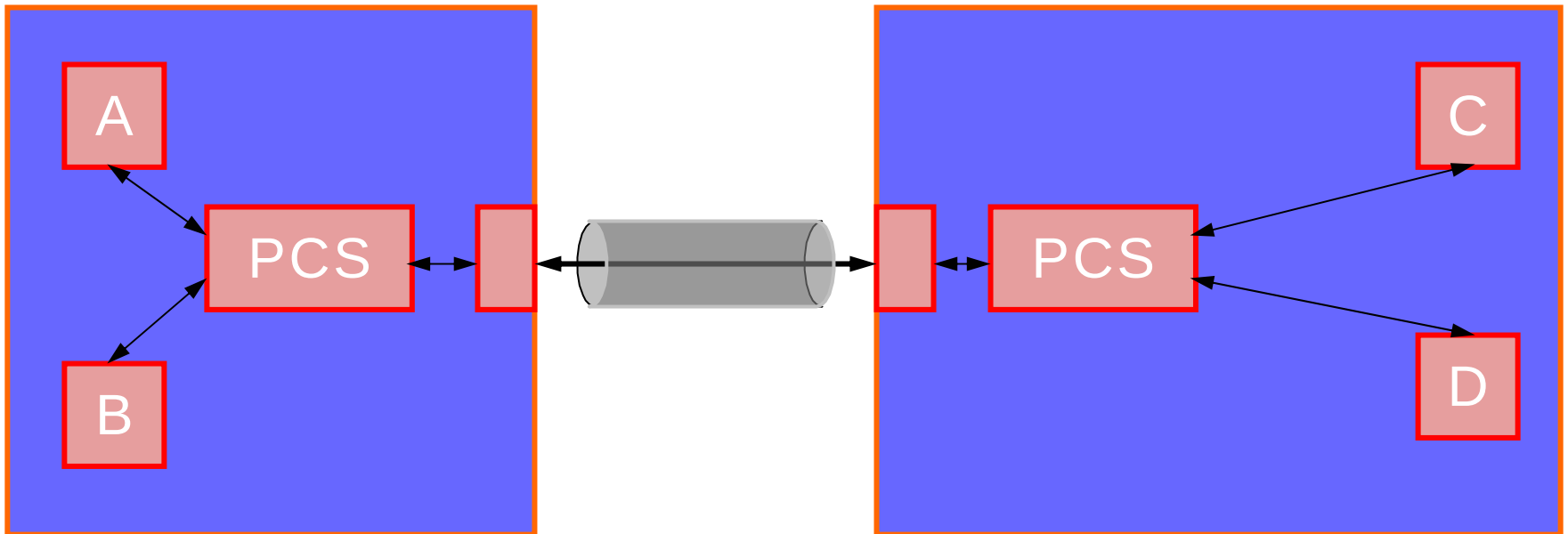
A MILS Workstation?



Processor

- Extend single node security policy enforcement to multiple nodes
- 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

- Part of MILS Middleware
- Responsible for all communication between MILS nodes
- Specific Requirements:
 - Strong Identity
 - Nodes, applications, and application instances
 - Separation of Levels/Communities of Interest
 - Secure Configuration of all Nodes in Enclave
 - Bandwidth provisioning & partitioning
 - Secure Clock Synchronization
 - Suppression of Covert Channels
 - Network resources: bandwidth, hardware resources, buffers
 - Secure Loading: signed partition images



- Notional Security Policy for Information Flows
 - P1A: There shall be no infiltration of data among flows
 - P1B: There shall be no infiltration of data within flows
 - P2A: There shall be no exfiltration among flows
 - P2B: There shall be no exfiltration within flows
 - P3: There shall be no unauthorized use of authorized flows
 - Example: No third party is allowed to cause information belonging to “A” to flow to “B” even if the security policy allows “A” to communicate with “B”

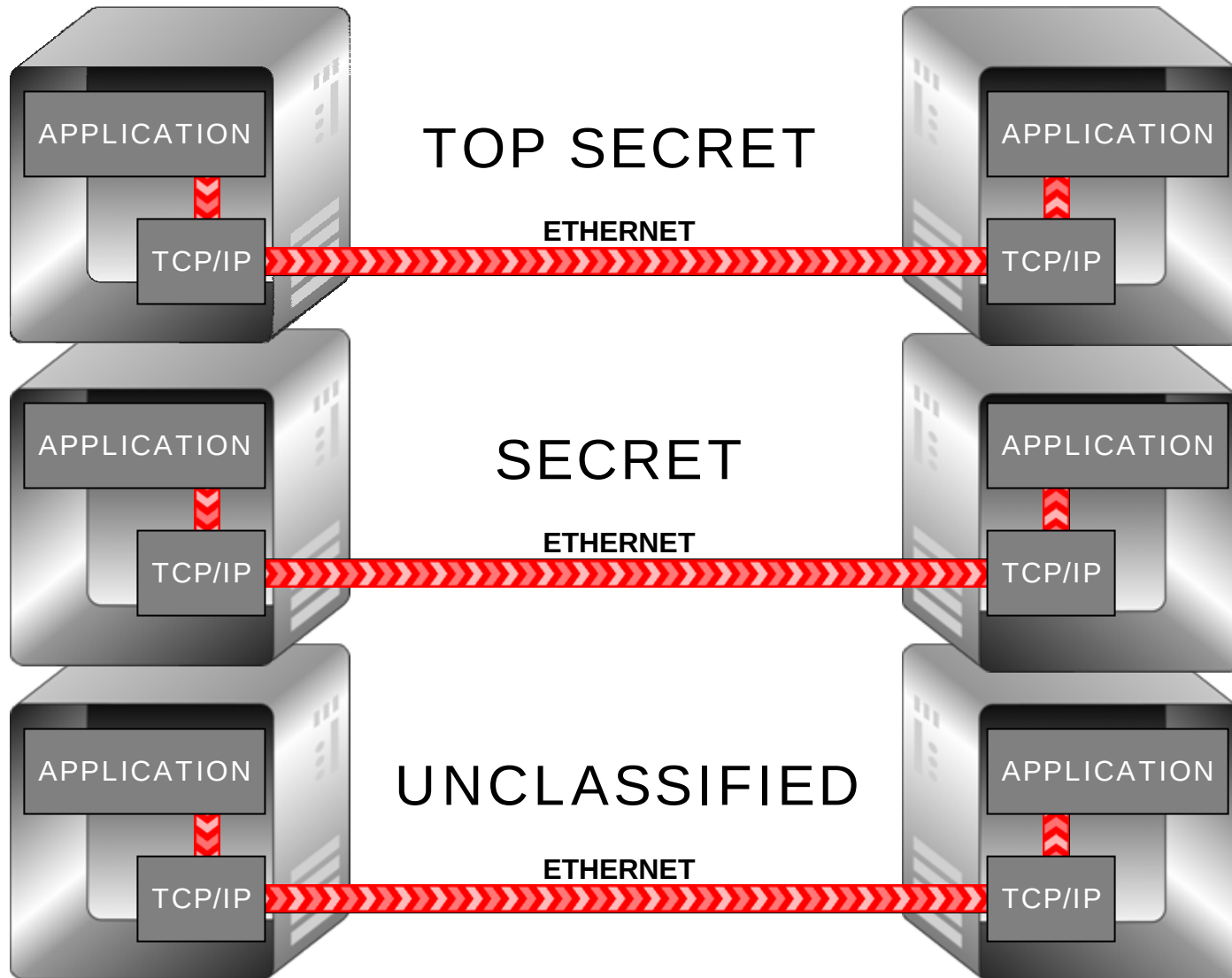
Smart Push, Smart Pull, Sensor to Shooter in a Multi-Level Secure/Safe (MLS) Infrastructure



- 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

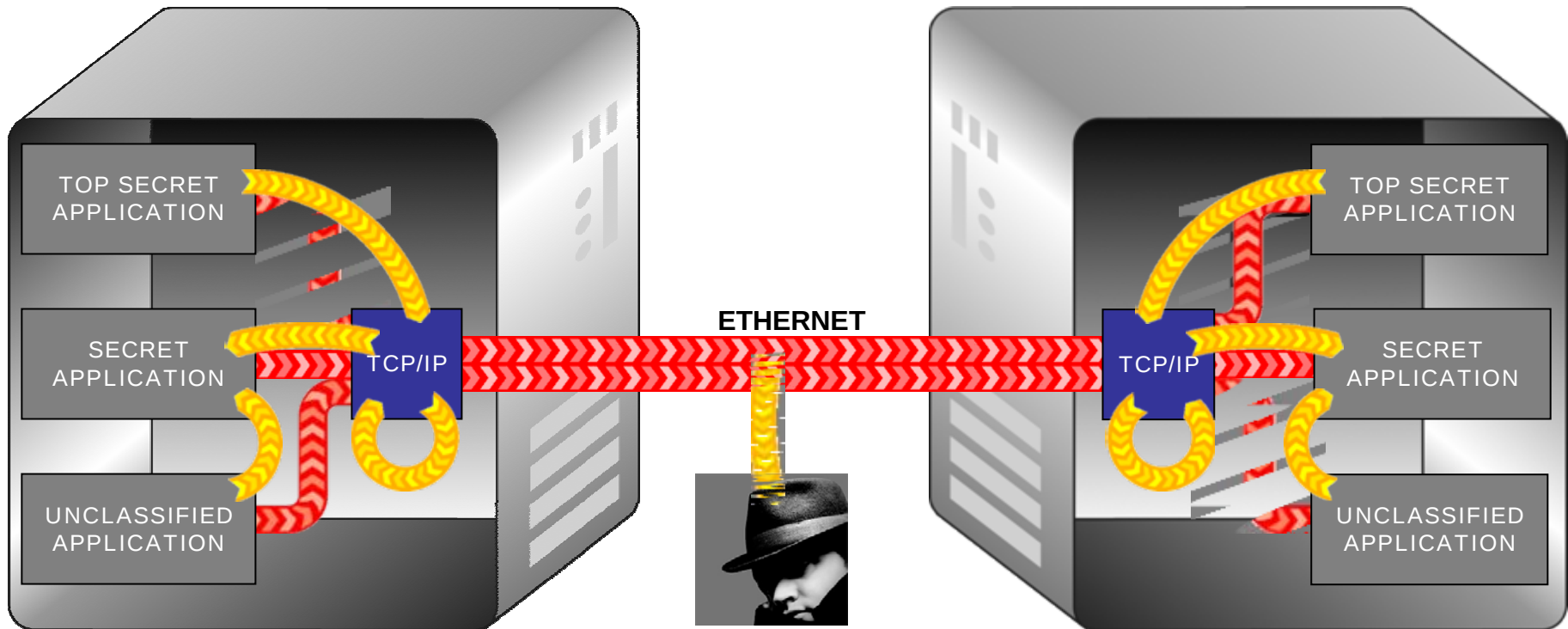
*Air Gap Works But....
Costly, Inflexible, & Awkward*

*Smart Push, Smart Pull, Sensor
to Shooter in a Multi-Level
Secure/Safe (MLS)
Infrastructure*



Combining Levels On Medium Assurance Platforms Is Unsafe

Smart Push, Smart Pull, Sensor
to Shooter in a Multi-Level
Secure/Safe (MLS)
Infrastructure

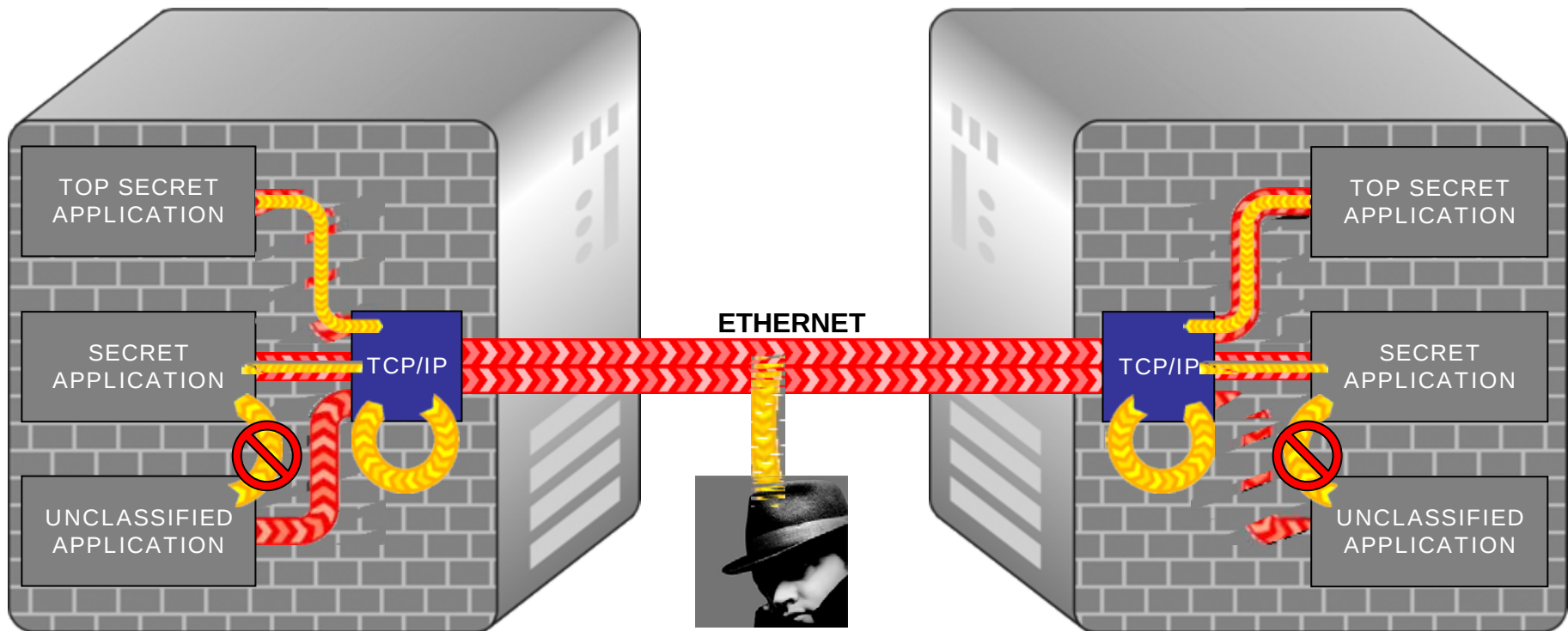


LEGEND

Yellow wavy arrow Vulnerabilities

MILS Separation Kernels Counter Most Internal Threats

*Smart Push, Smart Pull, Sensor
to Shooter in a Multi-Level
Secure/Safe (MLS)
Infrastructure*

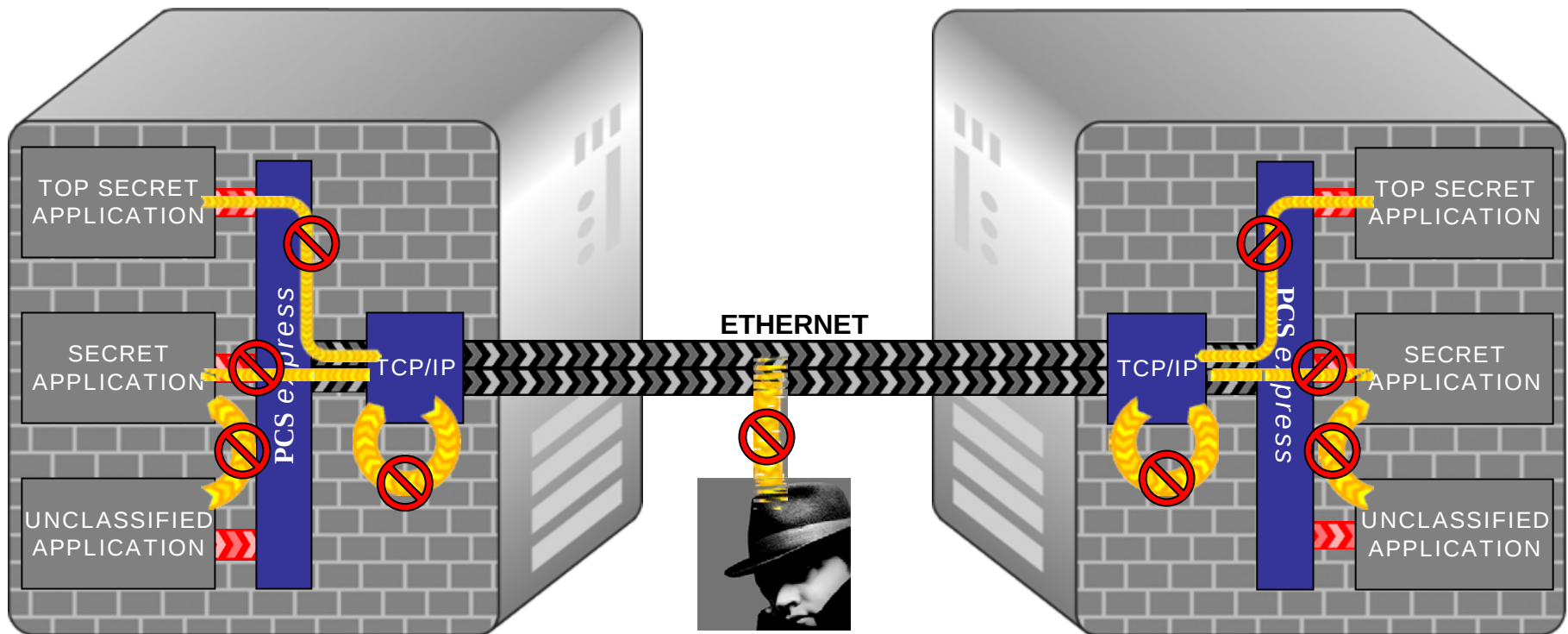


LEGEND

-  Vulnerabilities
-  Reduced Vulnerabilities

PCSexpress Completes MILS Separation Kernel

*Smart Push, Smart Pull, Sensor
to Shooter in a Multi-Level
Secure/Safe (MLS)
Infrastructure*

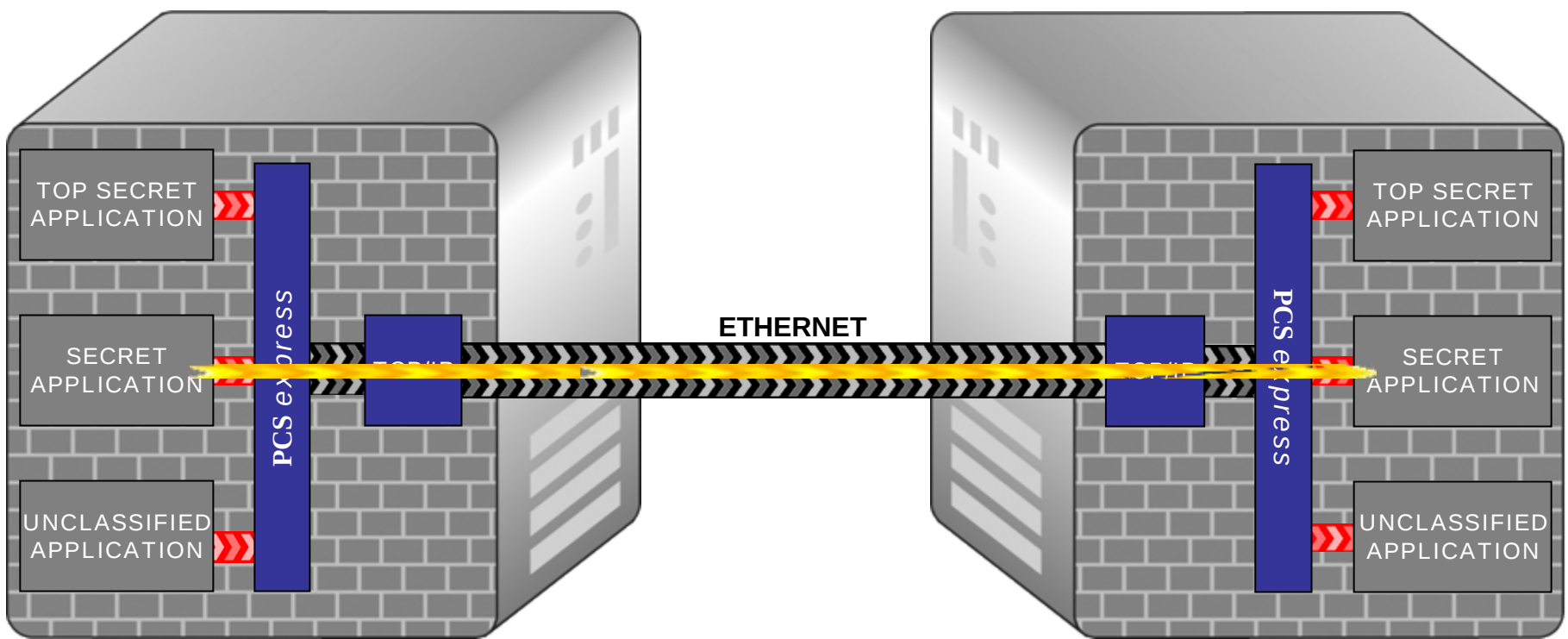


LEGEND

-  Vulnerabilities
-  Reduced Vulnerabilities

Guards Still Needed for Intra-level Threats

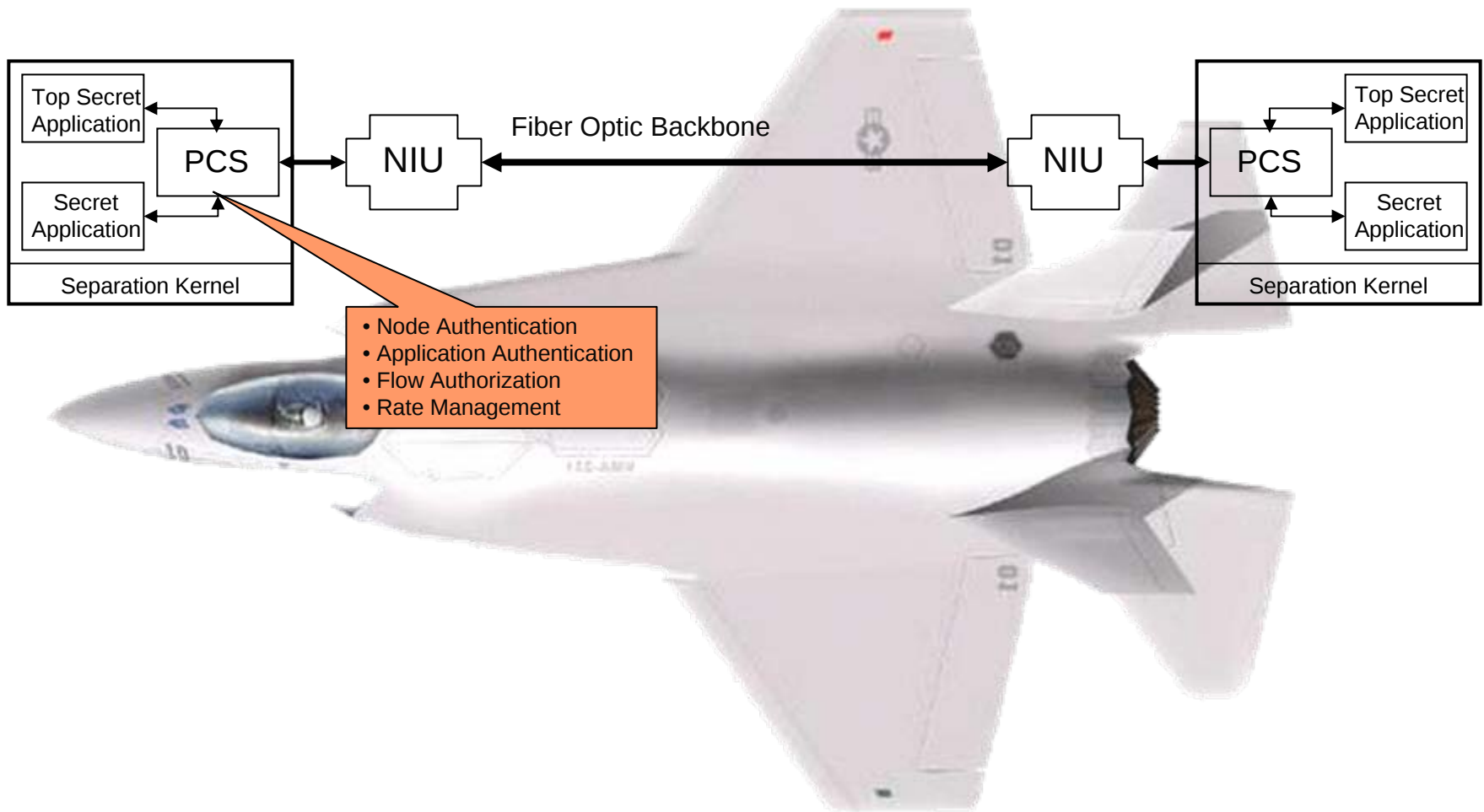
Smart Push, Smart Pull, Sensor
to Shooter in a Multi-Level
Secure/Safe (MLS)
Infrastructure

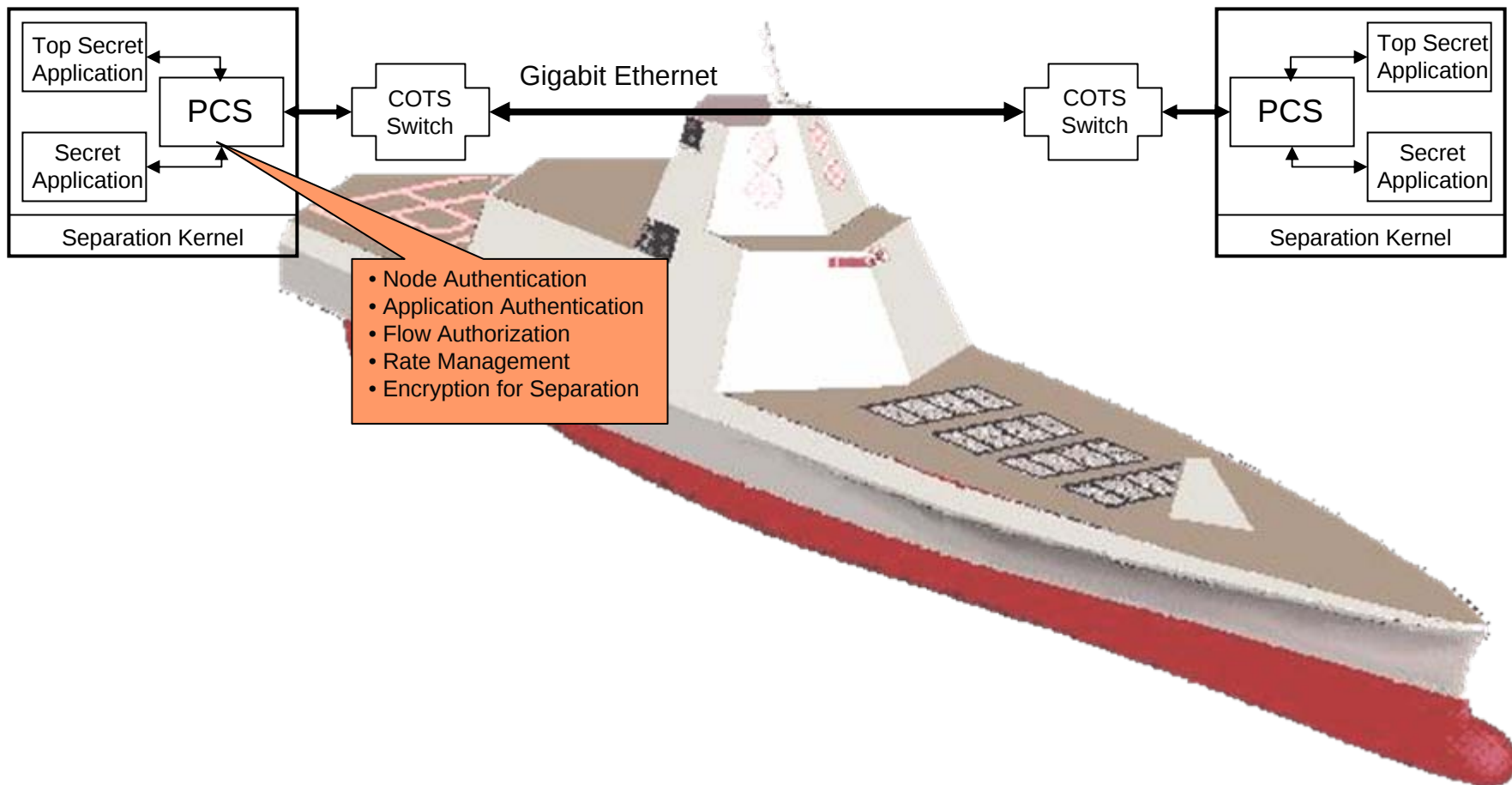


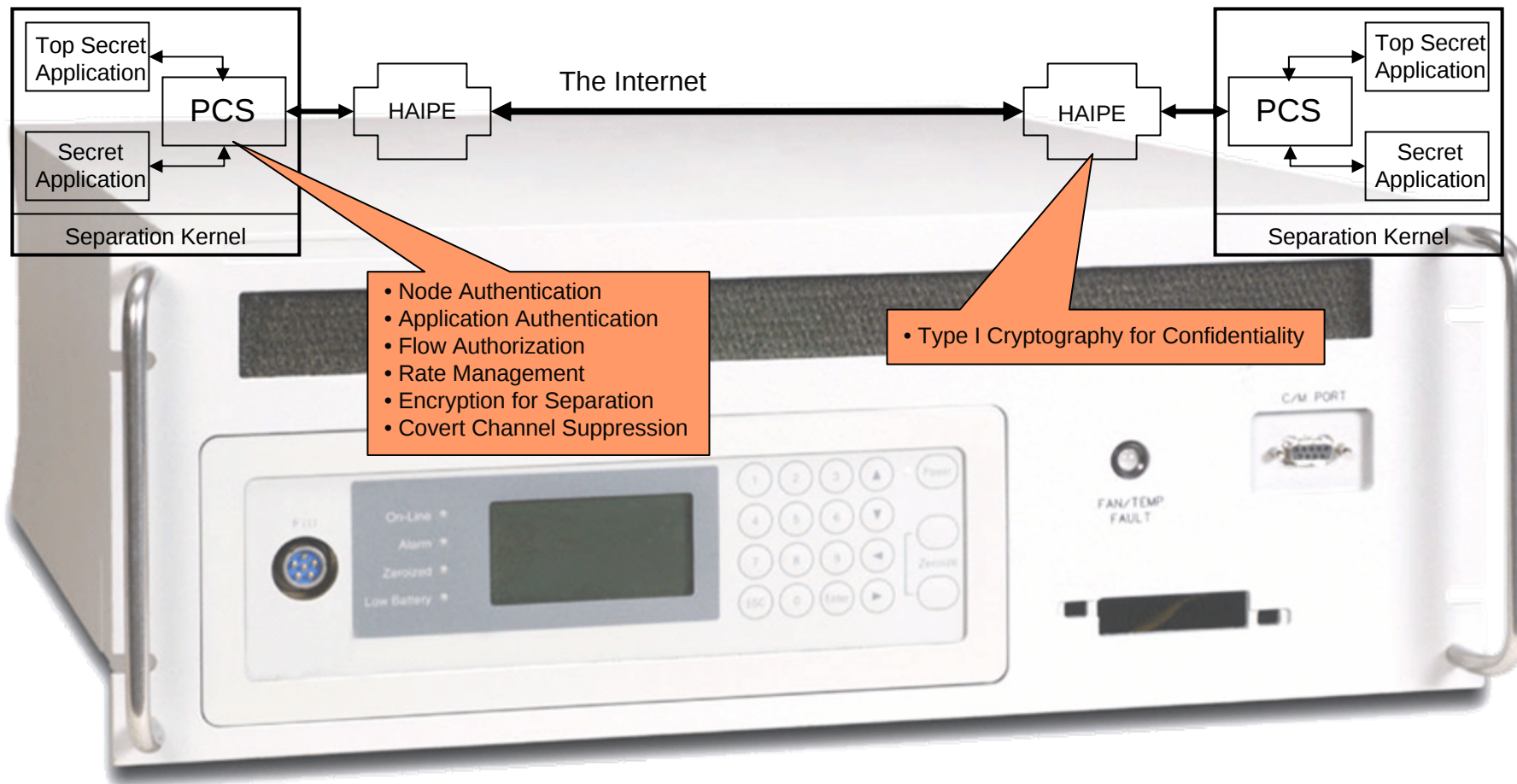
LEGEND

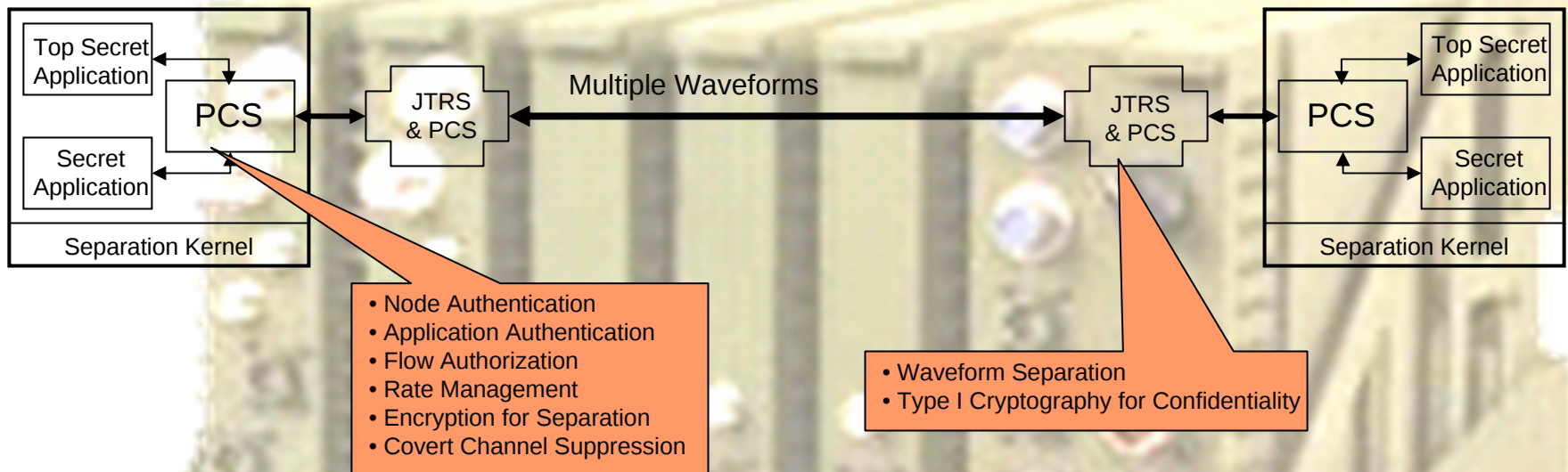
- Multiple Vulnerabilities
- Data Vulnerabilities

- Trusted Transport
 - Communications system can be trusted to maintain separation by level and Community of Interest
- Untrusted Transport
 - Communications system cannot be trusted to maintain separation by level and Community of Interest
- Gray Sky
 - Threats to communications confidentiality are acceptably low
 - Example: Front to back of an airplane or submarine; within an FCS tank
- Blue Sky
 - Threats to communications confidentiality are unacceptably high
 - Example: Radio transmission; the Internet





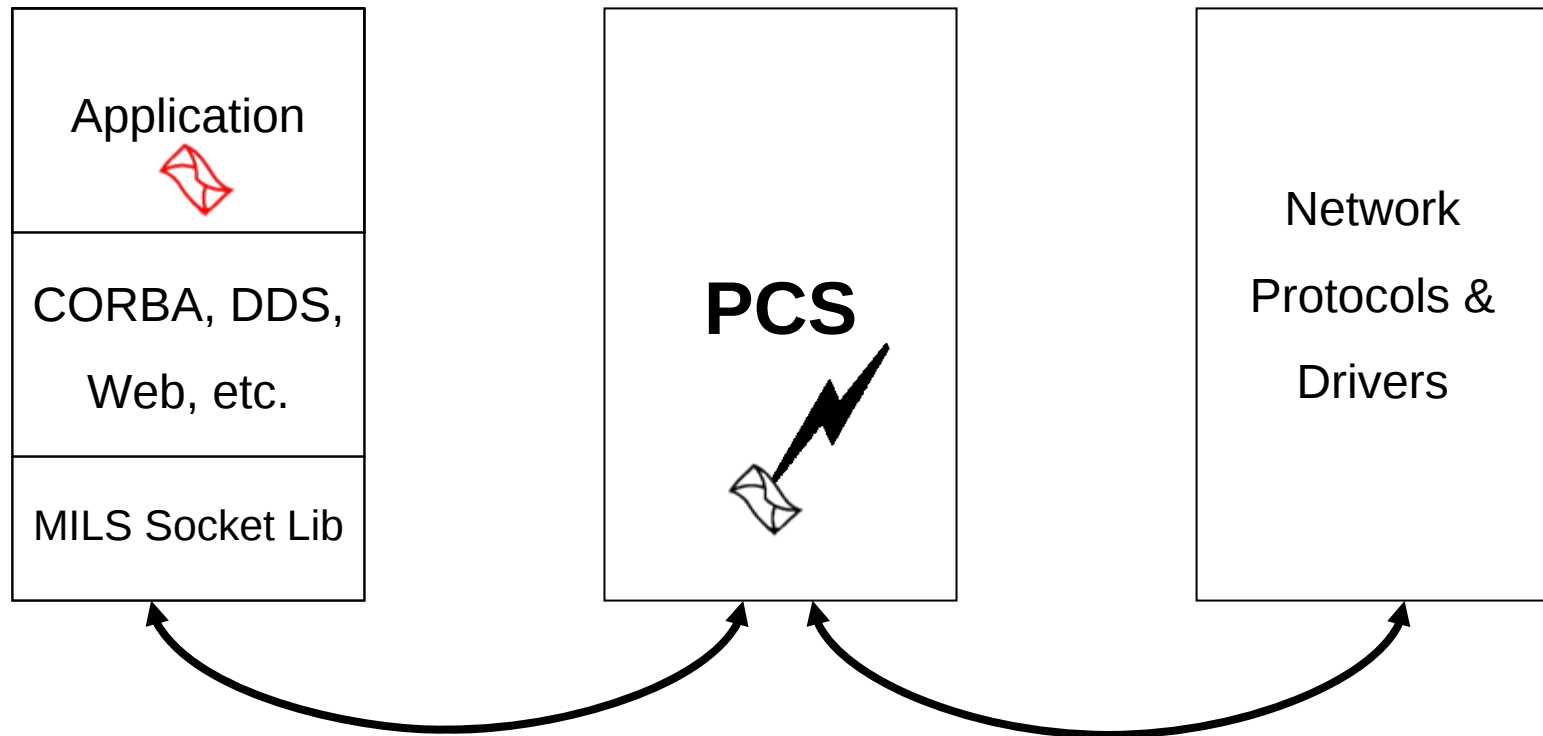




- Real-time CORBA can take advantage of PCS capabilities
 - Real-time CORBA + PCS = Real-time MILS CORBA
 - Additional application-level security policies are enforceable because of MILS SK and PCS foundation
- Real-time MILS CORBA represents a single enabling application infrastructure

- 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

- 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 to**
 - *Rearrange system functions to take advantage of protection without introducing new threats to data confidentiality and integrity*



▪ CCEVS:	Common Criteria Evaluation Scheme
▪ CCRA:	Common Criteria Recognition Arrangement
▪ CMMI:	Capability Maturity Model Integration
▪ COL:	Community of Interest
▪ COMINT:	Communications Intelligence
▪ CONUS:	Continental United States
▪ CORBA:	Common Object Resource Broker Architecture
▪ DCID:	Director of Central Intelligence Directive
▪ DDS:	Data Distribution Service
▪ EAL:	Evaluation Assurance Level
▪ ELINT:	Electronic Intelligence
▪ GIOP:	General Inter-Orb Protocol
▪ HAIPE:	High Assurance Internet Protocol Equipment
▪ HTTP:	Hypertext Transfer Protocol
▪ HUMINT:	Human Intelligence
▪ IAD:	Information Assurance Directorate
▪ IATF:	Information Assurance Technical Framework
▪ IBAC:	Identity Based Access Control
▪ IMINT:	Imagery Intelligence
▪ JTRS:	Joint Tactical Radio System
▪ MILS:	Multiple Independent Levels of Security
▪ MLS:	Multi-Level Security/Safety
▪ NSA:	National Security Agency
▪ PCS:	Partitioning Communications System
▪ RBAC:	Role Based Access Control
▪ SEI:	Software Engineering Institute (Carnegie Mellon)
▪ SIGINT:	Signals Intelligence
▪ SKPP:	Separation Kernel Protection Profile
▪ SOA:	Services Oriented Architecture
▪ SRD:	System Requirements Document
▪ TOE:	Target of Evaluation
▪ TSF:	TOE Security Functions