DSCI 519: Foundations and Policy for Information Security

Policy Composition, TNI Composition of MAID Components

Tatyana Ryutov

Outline

- ORCON
- Policy composition
  - Intro
  - TCB subsets
  - TCB partitions
- TNI Composition of MAID Components

Reminders

- Quiz 3 to be completed by October 30th
  - Covers lectures 7 and 8
- Semester project is due December 4th
  - Substantial grade penalty for late submission
    - Cumulative of 10% times number of days late
      - 1 day late: lose 10%
      - 2 days late: lose 30% (10% + 20%)
      - 3 days late: lose 60% (30% + 30%)
      - Greater than 4 days late not accepted

Presentation 11

cloud security

CCAF:
Cloud Computing Adoption Framework

Tianyi Wang
Xiaoke Li

USC Viterbi
School of Engineering

Your Questions

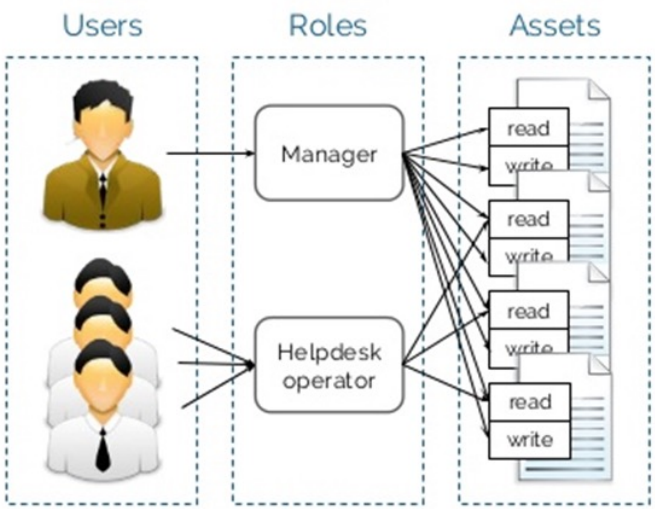
- Can you review how to work through examples of Clark-Wilson user permissions in a methodical manner?
- Why is "separation of duty" considered to be done outside of the system, when E2 says that the system must control the user's ability to access certain CDI & TP pairs? Is that not the system doing "separation of duty"?
- Differences between BLP/Biba and Clark-Wilson
- RBAC and ABAC, which is better?

ABAC Advantages

- ABAC rules can be extremely fine-grained and contextual
- Requires no advance knowledge of requestors
- An individual's attributes can be correlated from multiple sources to create a unified identity
- Highly adaptable to changing needs
  - Efficient for agencies where individuals come and go frequently
  - Support for dynamic groups
- Example: ABAC for AWS
  - https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_attribute-based-access-control.html

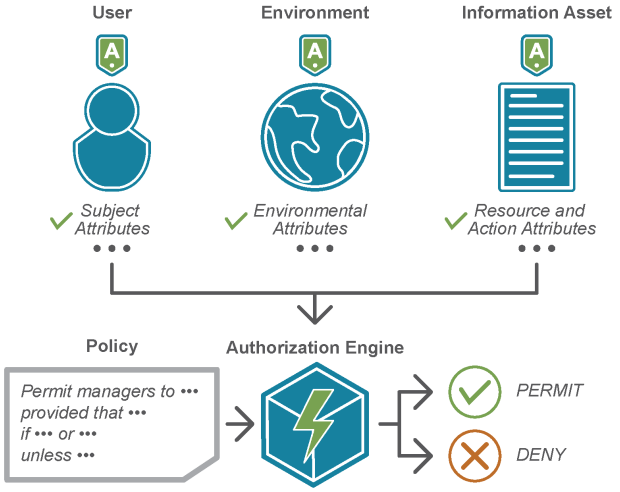
RBAC vs. ABAC

RBAC



- Initial effort of structuring roles
- Advantages in administration and auditability

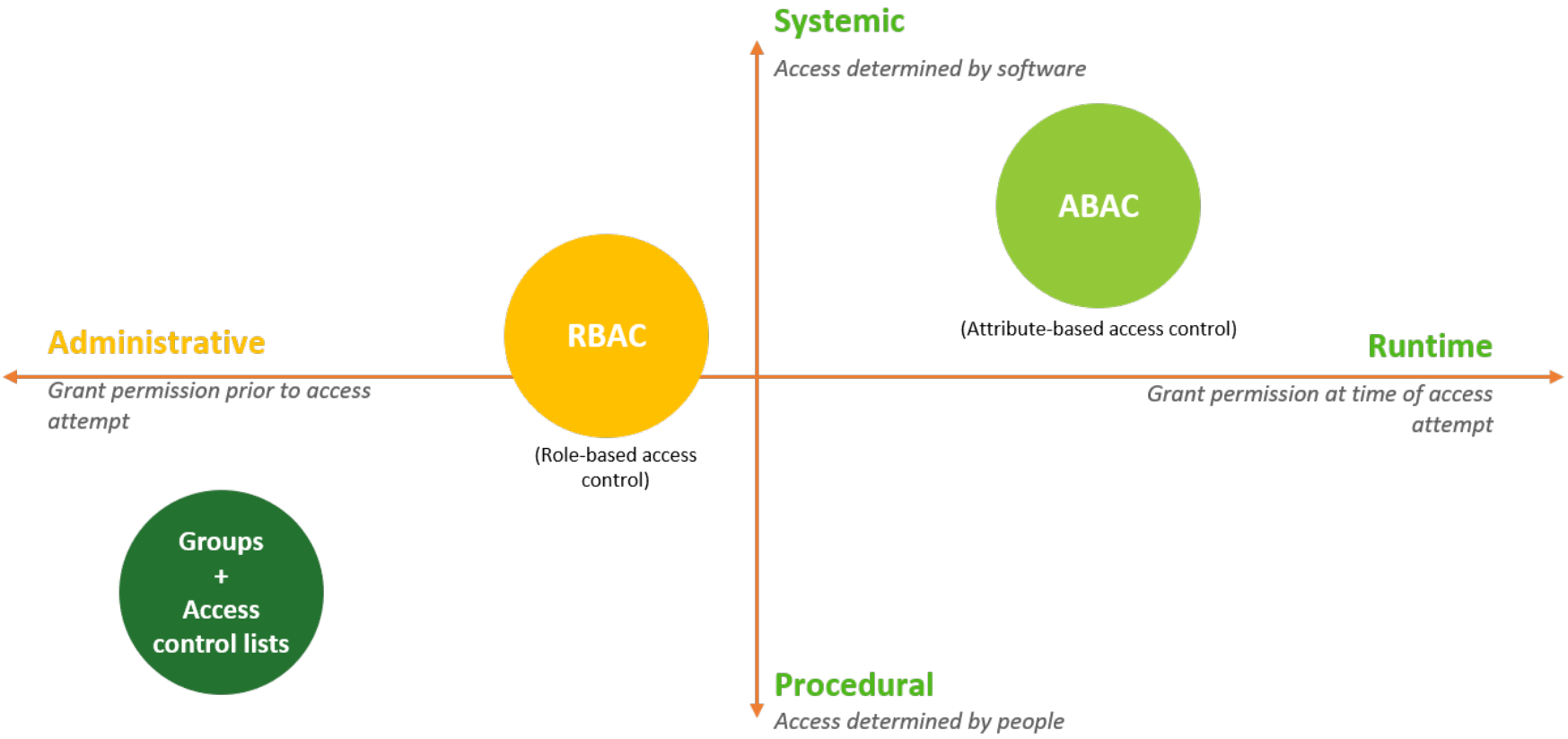
ABAC



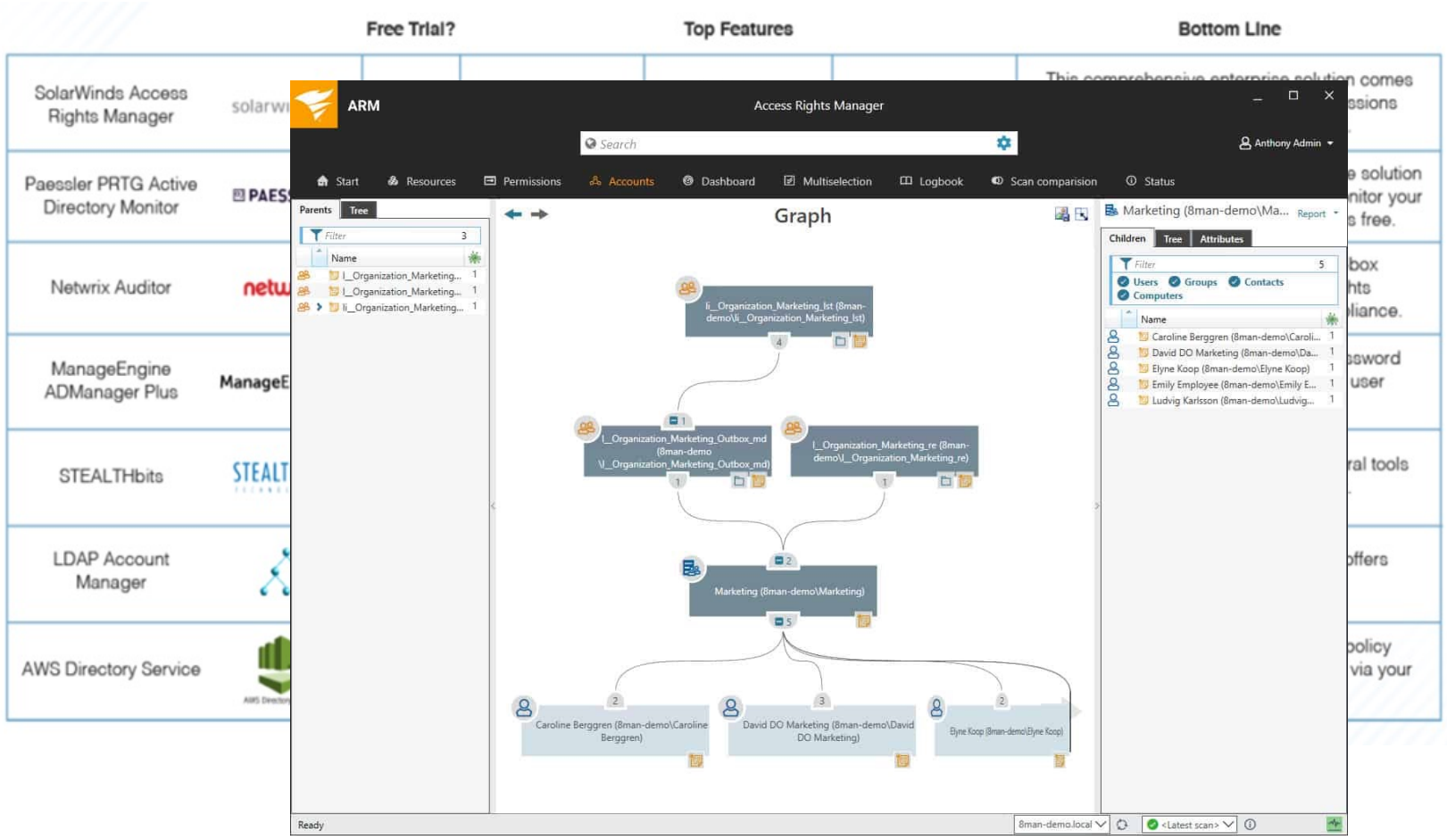
- Immediate creation of rule policies
- Complicated management and audit of user permissions

USC Viterbi School of Engineering

Groups, RBAC, ABAC



Best Access Rights Management Tools



Source: https://www.dnsstuff.com/access-rights-management-tools

Outline

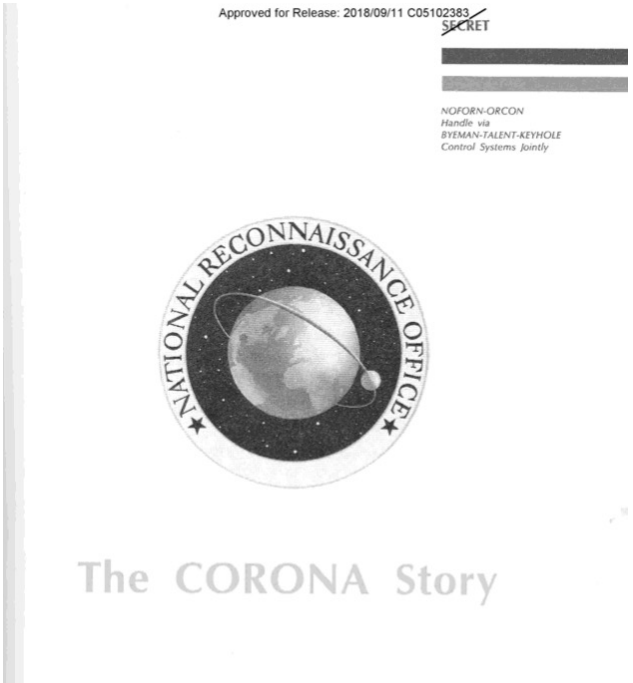
- ORCON
- Policy composition
  - Intro
  - TCB subsets
  - TCB partitions
- TNI Composition of MAID Components

ORiginator **CON**trol

- **Problem**: organization creating document wants to control its dissemination
- Example: manager writes a memo for distribution to her immediate subordinates, and she must give permission for it to be disseminated to anyone else
- In the paper world, ORCON is one of the control markings for restriction of document distribution defined by the Director of Central Intelligence Directive (DCID) 1/7
  - Security Controls on the Dissemination of Intelligence Information
  - https://fas.org/irp/offdocs/dcid1-7.html

ORCON: Example

- The ORCON policy has been used in the U.S. intelligence community
- Objective:
  - Desired “Originator Control” in Closed-Network Information Sharing
  - Examples: rescind access; prevent forwarding
  - https://www.dni.gov/index.php/who-we-are/organizations/ic-cio/ic-cio-related-menus/ic-cio-related-links/ic-technical-specifications/orcon-need-to-know-access?highlight=WyJvcmNvbiJd



WARNING NOTICE
INTELLIGENCE SOURCES OR METHODS INVOLVED
(WNINTEL)

NATIONAL SECURITY INFORMATION
UNAUTHORIZED DISCLOSURE SUBJECT TO CRIMINAL SANCTIONS

CLASSIFIED BY:BYE-1
DCLL: OADR

DISSEMINATION CONTROL ABBREVIATIONS

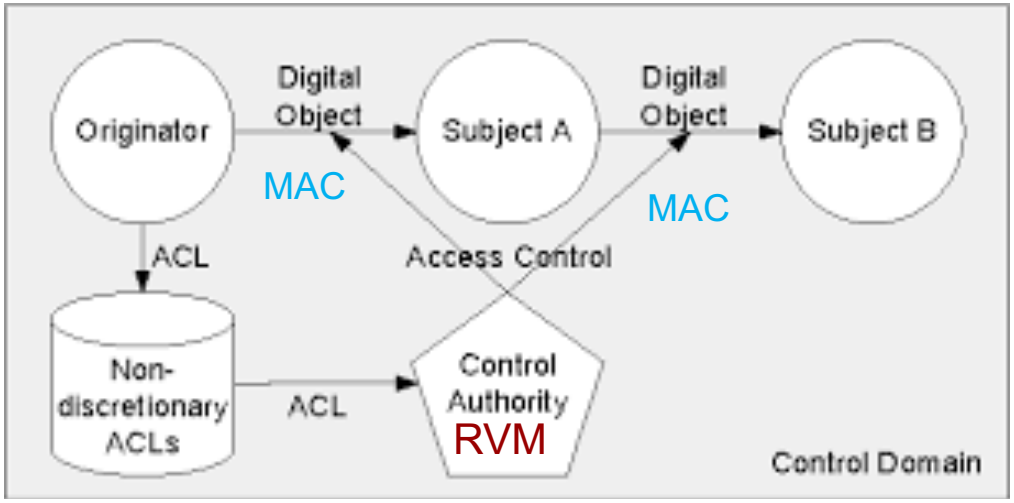
NOFORN - NOT RELEASABLE TO FOREIGN NATIONALS
ORCON - DISSEMINATION AND EXTRACTION CONTROLLED BY ORIGINATOR

ORCON: More Recent Examples

- Some examples:
  - S. Kirkman, R. Newman, ”Bridging the Cloud Trust Gap: Using ORCON Policy to Manage Consumer Trust between Different Clouds”, *2017*
  - Y.-Y. Chen and R. B. Lee, “Hardware-assisted application-level access control,” 2009
  - J. Park and R. Sandhu, “Originator control in usage control,” *2002*
- ORCON is related to DRM
  - Typically use cryptographic mechanisms for copy protection of digital objects
    - Issue: once data is decrypted, all control is lost
  - Example: Oracle Entitlement Server

ORCON Requirements

- Subject $s \in S$ marks object $o \in O$ as ORCON (in organization *X)*
- *X* allows *o* to be disclosed to subjects acting on behalf of another organization *Y* with the following restrictions:
  1. o cannot be released to a subject in another organization without *X* 's permission; and
  2. Any copy of *o* must have the same restrictions placed on it
- DAC?
- MAC?



ORCON Combines MAC and DAC

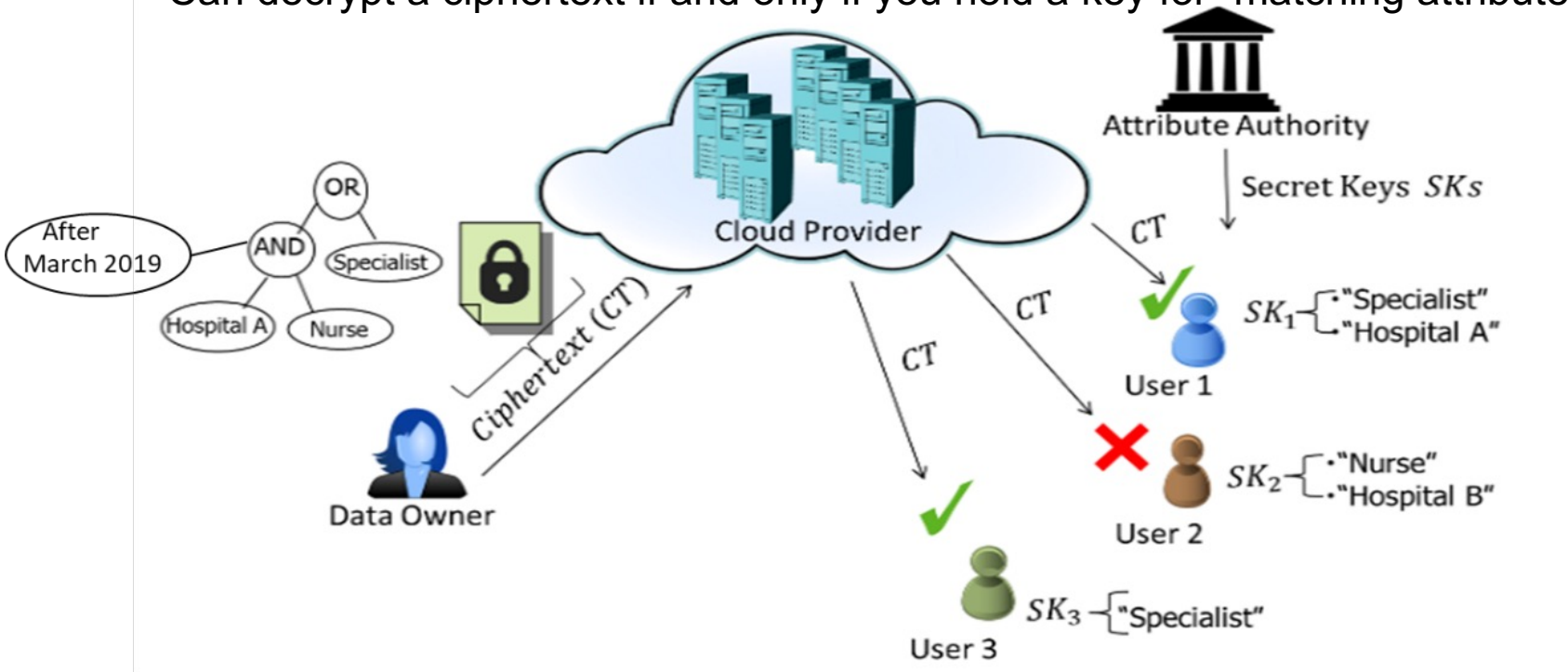
- DAC allows owner to set any permission
- MAC depends on centralized control
- ORCON is inherently decentralized:
  1. Owner does not control access after the object is copied
     - This is MAC
  2. Access control restrictions are copied with the object
     - This is MAC
  3. Creator (Originator) can alter access control restrictions on a per-subject and per-object basis
     - This is DAC

How can we enforce ORCON?

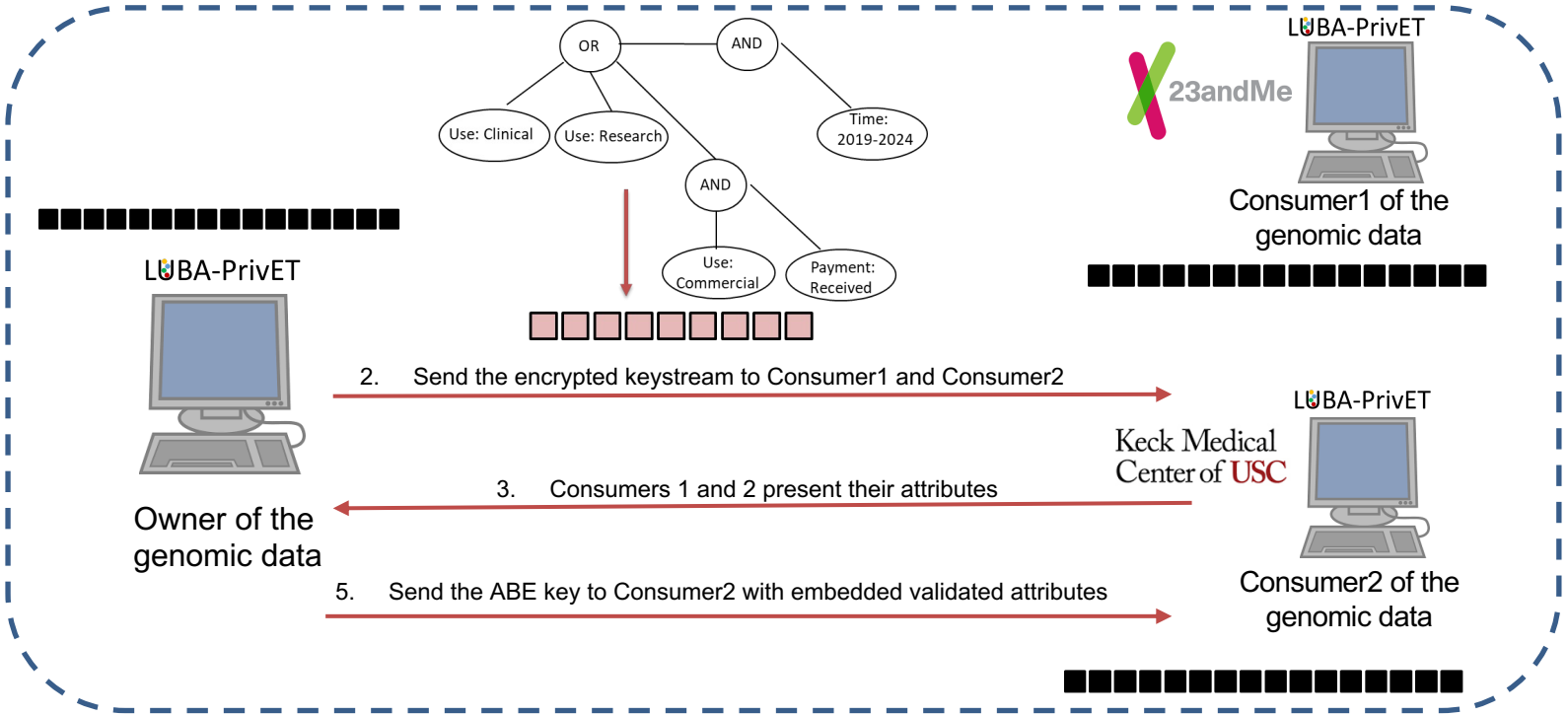
- DAC?
  - With DAC, the owner (=! creator) can change the permissions at will, without considering the creator (originator) original permissions
  - DAC fails!
- MAC?
  - Theoretically possible but...
    - Separate category is needed for every <Object, Creator, Owner> tuple
    - Category explosion problem!
    - Requires a centralized solution for category creation and access control, impossible to implement in real world
  - MAC fails!
- One solution: Attribute-Based Encryption
  - The attribute-based policies are embedded in the encrypted data objects (ciphertext), thus making it impossible to remove or modify the policies
  - The data requestor will need to prove the possession of the requestor-related attributes, in order to decrypt the ciphertext

Attribute-based Encryption (ABE)

- Attribute-based encryption (ABE) is an extension of the IBE
  - Supports fine-grained access control policies that are cryptographically enforced
  - Data objects are encrypted using a set of attributes typically expressed as logical combinations (**access structure**) referencing attributes of the requester, environment, or the data object
  - Can decrypt a ciphertext if and only if you hold a key for "matching attributes"



ORCON Example: Genomic Data Protection



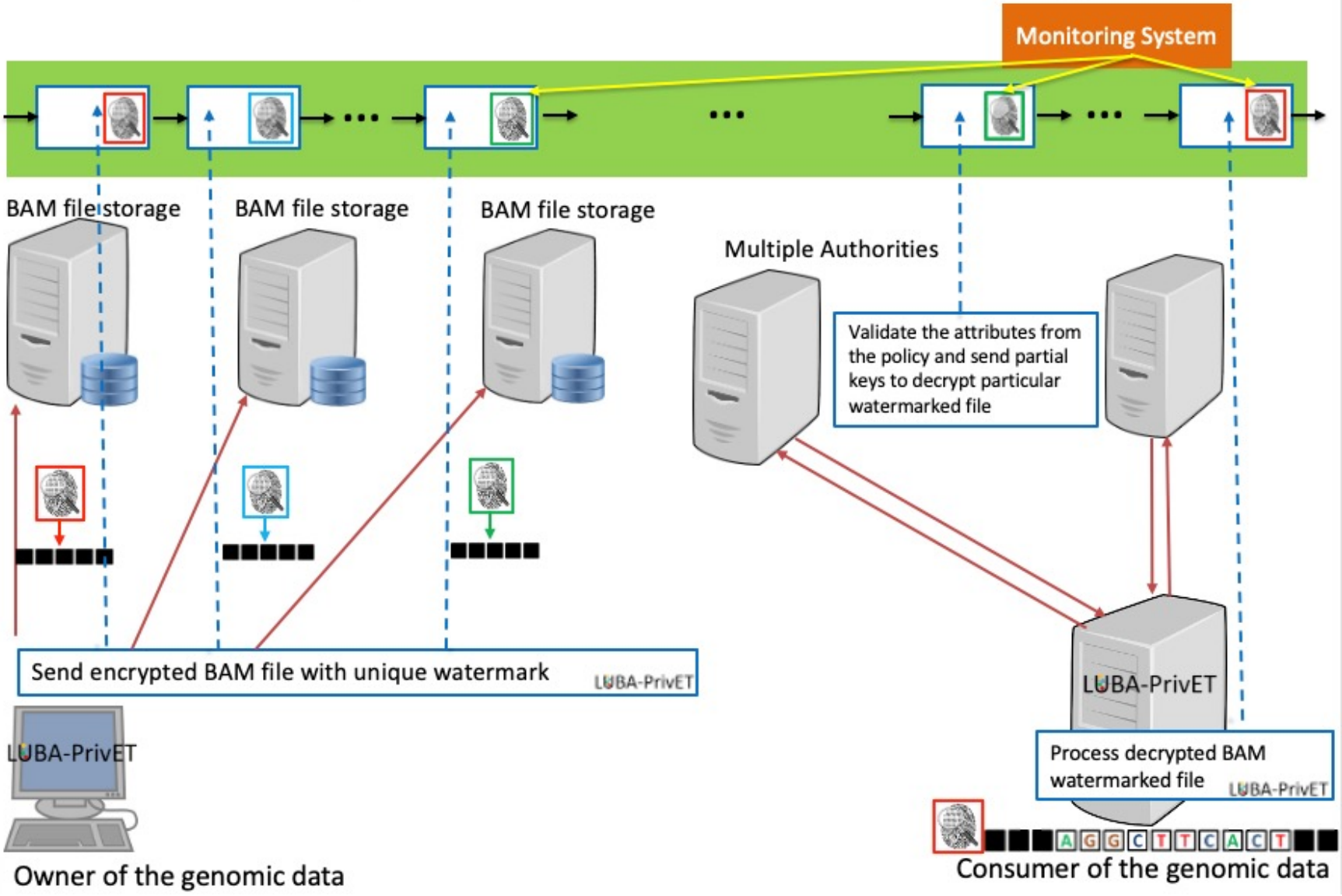
1. Encrypt the keystream that opens a genomic region (e.g., APOE gene) using ABE, the access control policy is imbedded in the ciphertext

4. Validate the attributes, decrypt only if Consumer's attributes satisfy the policy

6. Decrypt the APOE gene

A G G C T T C A C T

ORCON Example: Genomic Data Protection



Hybrid Policy Models: Key Points

- Chinese Wall policy focuses on conflict of interest
  - Information is stored in hierarchically arranged levels
  - Subjects are only allowed access to information which is not in conflict with any other information that they already possess
  - CW is based on access history
- RBAC is a policy-neutral model
  - Uses role to simplify administration of access control
  - Base access control on job functions
  - Family of models (add role hierarchy, inheritance, role activation constraints)
- ABAC rules can be extremely fine-grained and contextual
- ORCON is a combination of DAC and MAC
  - Enforcement is a much bigger issue

Outline

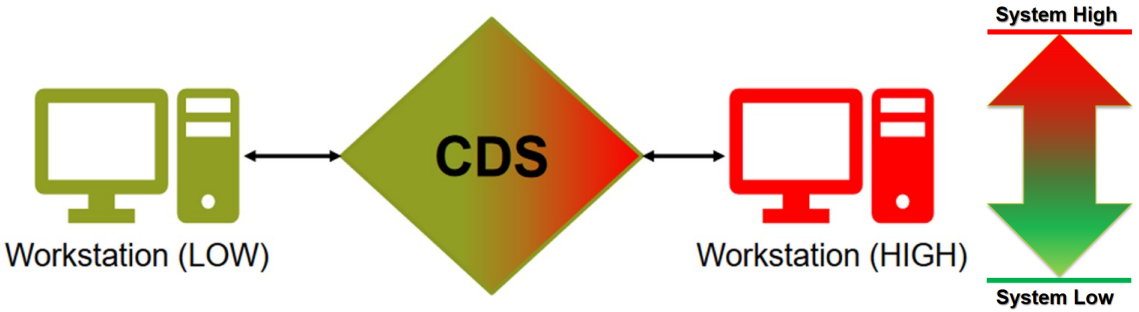
- ORCON
- Policy composition
  - Intro
  - TCB subsets
  - TCB partitions
- TNI Composition of MAID Components

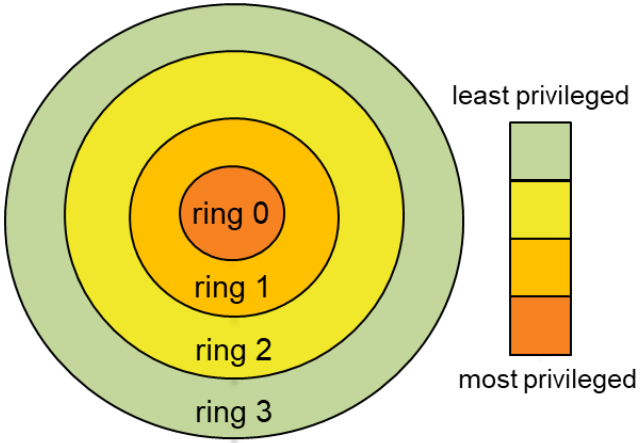
Monolithic Policy Enforcement

- Baseline is a monolithic reference monitor (RM)
  - Complete access control policy abstraction
- Hardware and software implements the RVM
  - Defined as a “security kernel”
  - Implement monolithic trusted computing base (TCB)
  - TCB is responsible for enforcing a security policy
- Formal security policy model (FSPM) for kernel
  - Discretionary access control (DAC) policy
  - Mandatory access control (MAC) policy
- A subject is “technically a process/domain pair”
  - Ring integrity policy

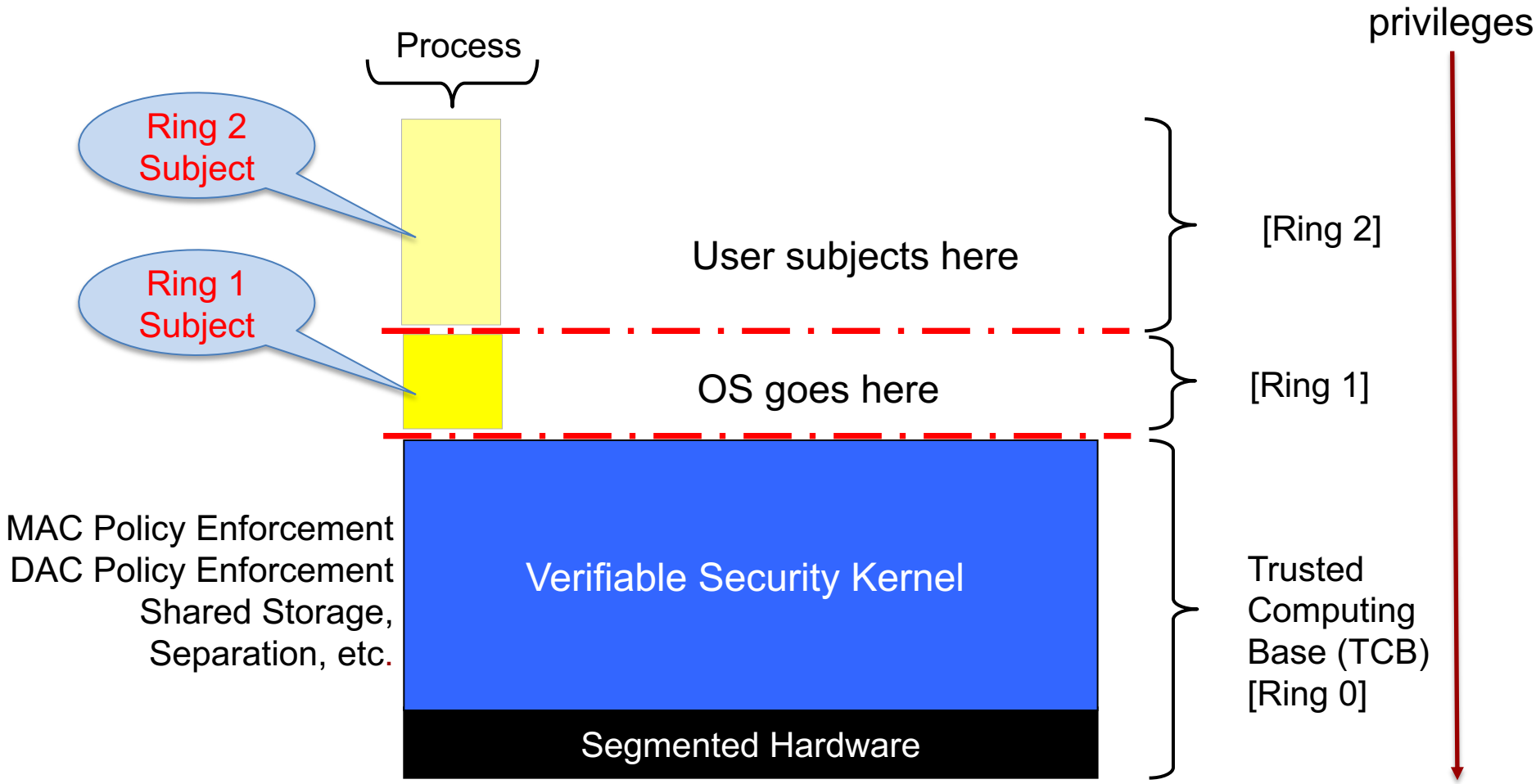
Side Note: Domains and Domains

- Note that the term “Domain” is used in two ways:
  1. **Security domain** – An environment in which all users and objects have uniform security labels (e.g., dedicated mode computer)
     - Interfaces between these domains are cross-domain solutions
  2. **Execution domain** – Kernel/user modes, or an ordered set of protection rings



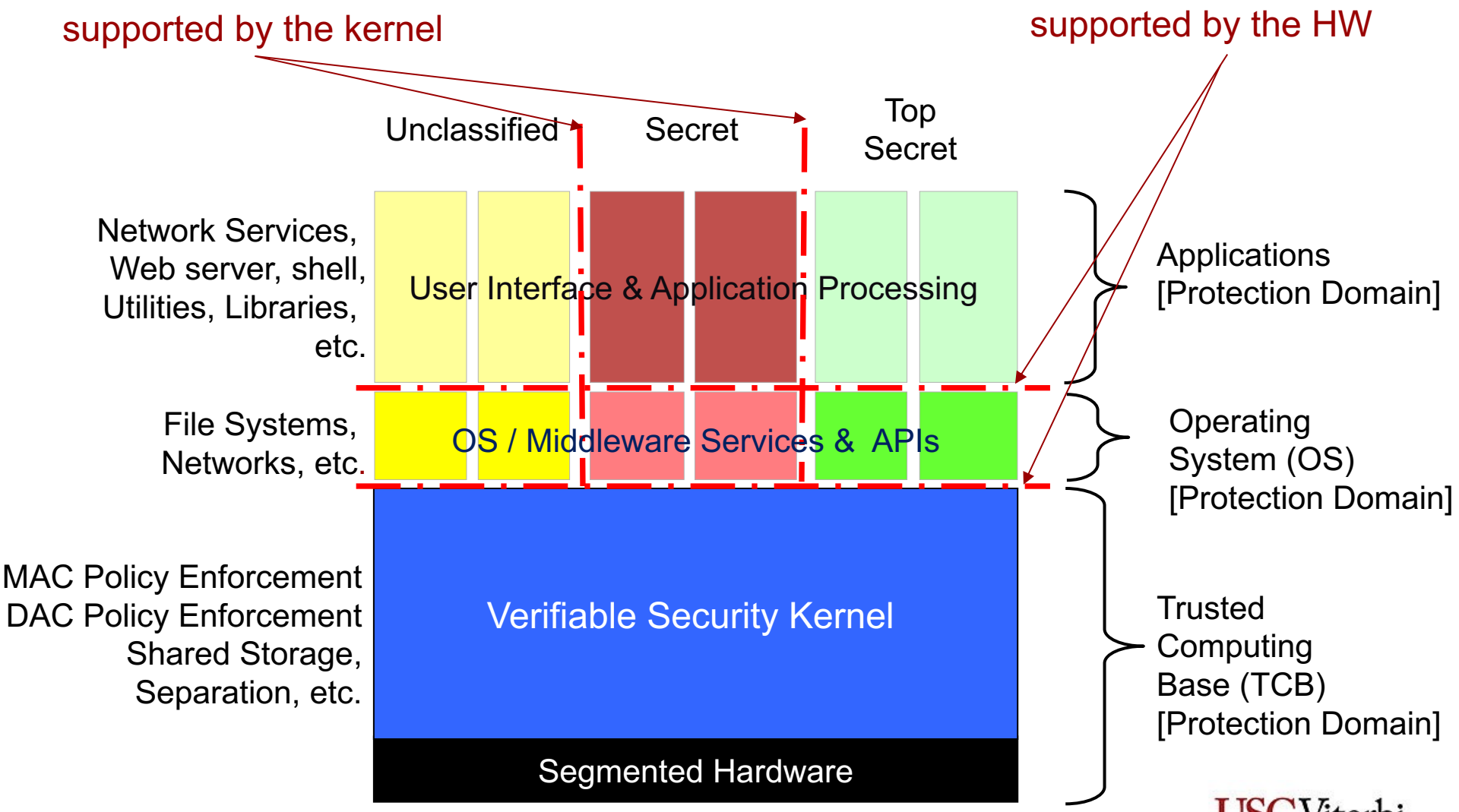


Process With Subjects



Want at least 3 rings, why?

Monolithic Policy Enforcement



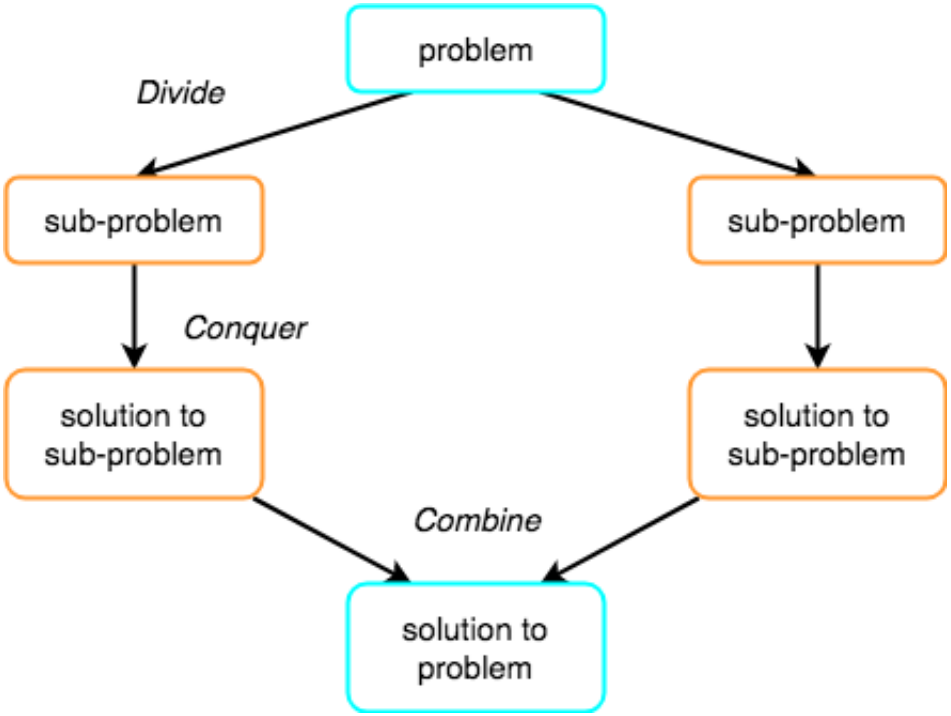
The "Composition Problem"[1]

- Composition of several secure systems
  - Each with known security properties
  - What security properties for interconnected systems?
- Bell conjecture: no closed-form solution (computable in finite number of steps)
  - Limited success for unconstrained composition
  - Unavoidable in heterogeneous networks
- Solution from Trusted Network Interpretation (TNI) (red book) and Trusted Database Interpretation (TDI) is tightly constrained
  - Need network-view (TNI)
  - Includes interfaces and protocols (TDI)
- Uses **only** (1) Partitioned TCB and (2) TCB Subset

1. Bell, David Elliott. "Looking back at the Bell-La Padula model." In Computer Security Applications Conference, 21st Annual, IEEE, 2005.

"Divide and Conquer" Strategy

- If we can break a single big problem into smaller sub-problems, solve the smaller sub-problems and combine their solutions to find the solution for the original big problem, it becomes easier to solve the whole problem
- Issue: **emergent properties**
  - Result from the interaction of the component properties within a composite system
    - The whole is more than the sum of its parts

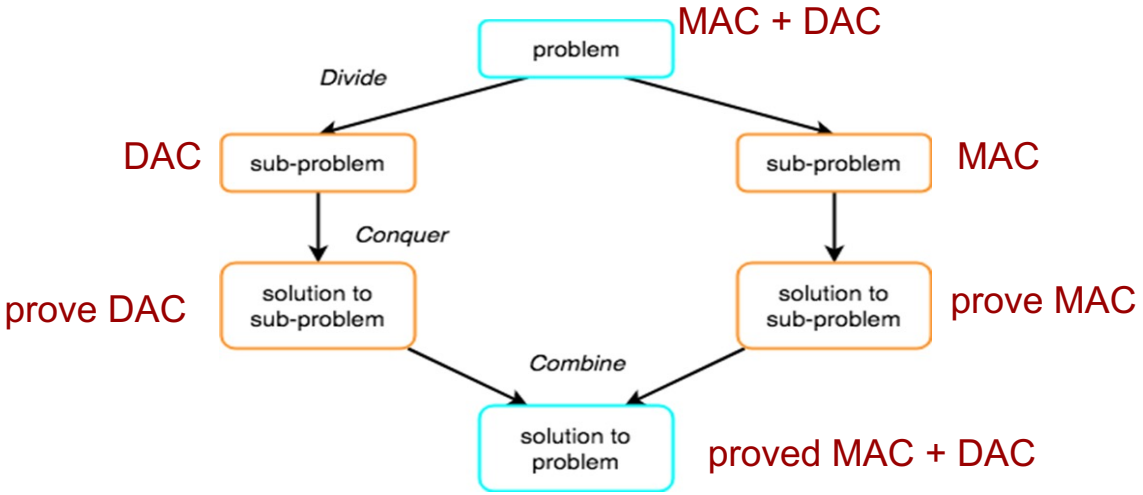


Outline

- ORCON
- Policy composition
  - Intro
  - TCB subsets
  - TCB partitions
- TNI Composition of MAID Components

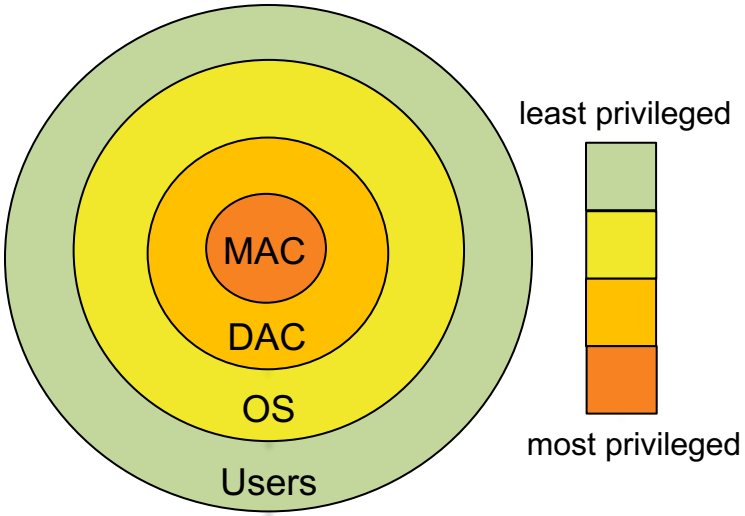
TCB Subset Introduction

- Want to validate enforcement of complex policy
- "Divide and conquer" strategy
  - Divide policy into disjoint security policy subsets
  - Must prove they compose into single system policy
- A TCB subset enforces each policy subset
  - We say a policy subset is allocated to a TCB subset
  - Must verify that each subset enforces its policy
- Divide the system TCB into simpler parts
  - Verify each part (subset) separately
  - Composed subsets enforce a single system policy
  - Must validate parts are **composed in correct way**



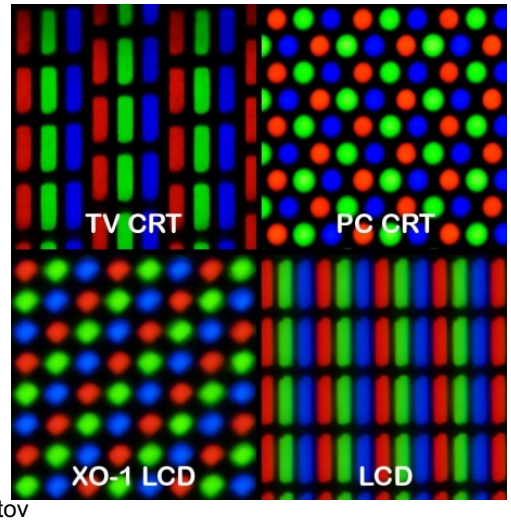
TCB Subset Introduction

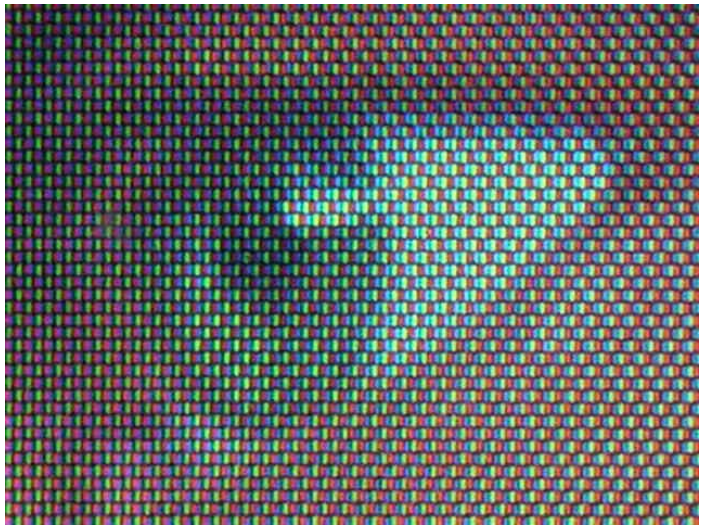
- Leverage protection domain structure
  - Purely hierarchical domains ordered by privilege
  - Satisfied by the classic “protection ring” case
- Compose complex TCB from simpler TCBs
  - Each partial TCB (subset) is in individual execution domain
    - Each must satisfy RM properties!
  - Allows for a chain of simpler evaluations
  - Called an “incremental evaluation”: do simpler evaluations multiple times
  - Must be able to verify that composite TCB is correct
- Suited to complex policies over virtual objects
  - Objects different from physical hardware storage, e.g., files



Virtual Objects

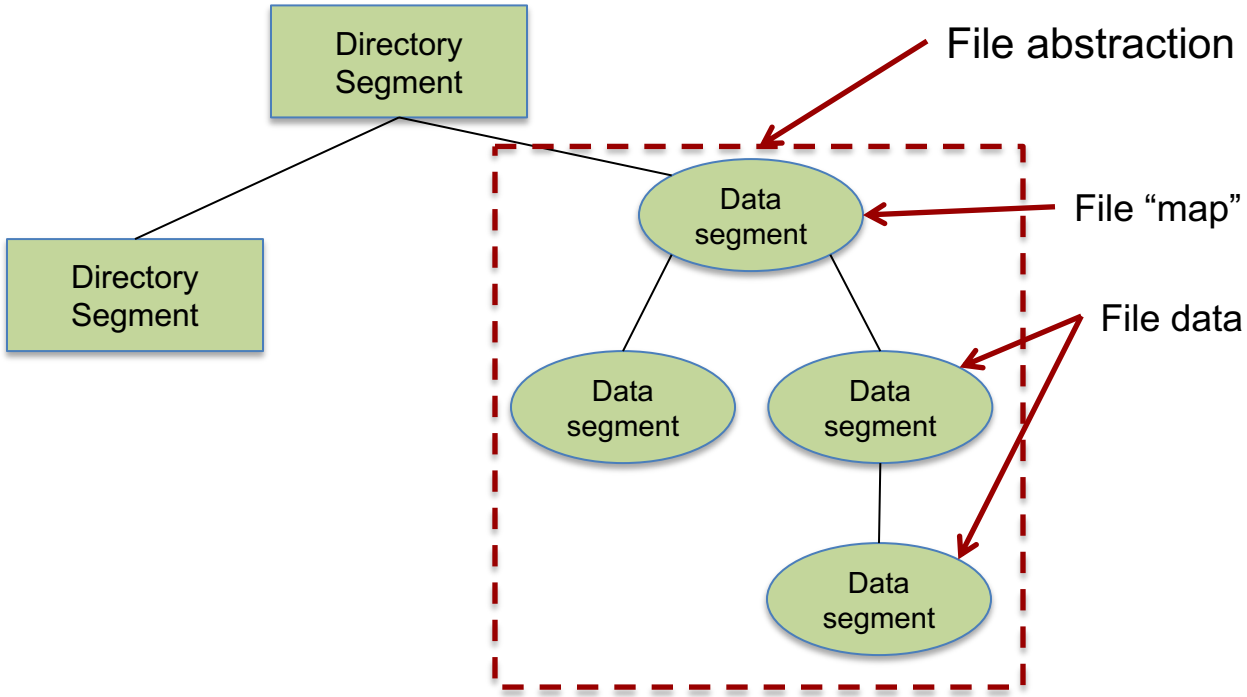
- Abstractions created out of “concrete” objects
- Example:
  - Think of pictures made out of pixels
    - “Real” image is pixels
    - Virtual objects are images created out of pixels
- “Real” objects are segments
- “Virtual objects”
  - Files
  - Relational database tables





Virtual Objects: Example

- "Real" objects are segments supported by HW



TCB Subset Abstraction

- TCB subset enforces some access control policy
  - Subjects attempt to access objects under its control
    - Subjects = (process, domain)
    - Objects = real or possibly virtual objects
  - Every TCB subset meets reference monitor properties
    - Isolation
    - Completeness
    - Verifiability
- Monolithic trusted system has single TCB subset
  - "Degenerate case" – TCB subset is the whole TCB
- RM implementation is distinguished from other subsets
  - It is the only subset to directly access platform hardware
    - System objects & resources – memory, devices, etc.

Converting Allowed Access to Prohibited Access

Consider 3 access modes: r,w,x

Allowed Access

	Segment 1	Segment 2
Alice	rw	
Bob		x
Carol	r	w

Prohibited Access

	Segment 1	Segment 2
Alice	x	rwx
Bob	rwx	rw
Carol	wx	rx

Prohibited Access by MAC enforced by TCB subset 1

	Segment 1	Segment 2
Alice		w
Bob	w	w
Carol	w	

Prohibited Access by DAC enforced by TCB subset 2

	Segment 1	Segment 2
Alice	x	rx
Bob	rx	r
Carol	x	rx

Overall system policy is a union of TCB subsets 1 and 2

Acess Control Policy Abstraction

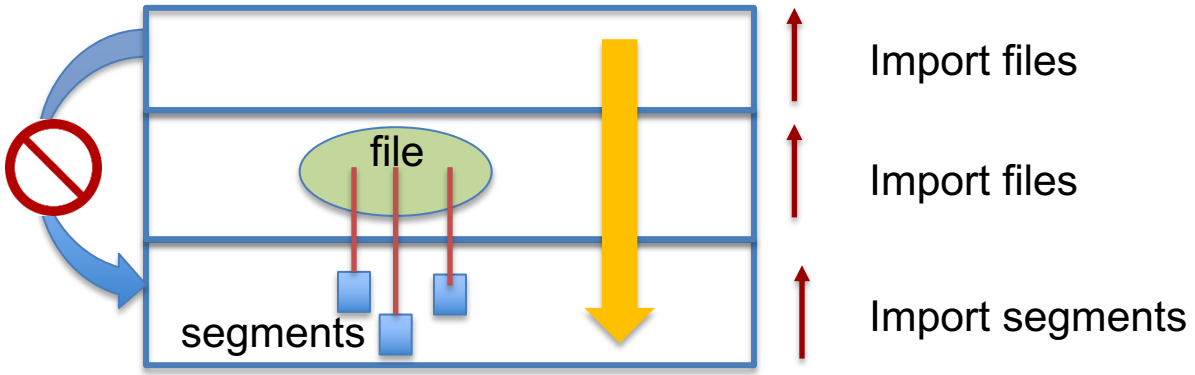
- Policy P is a set of ordered triples <s, o, m>
  - s is a particular subject
  - o is a particular object
  - m is a particular access mode that is prohibited
- Policy is represented as **prohibited** accesses
  - NOTE: failure to grant allowed access is not insecure
- Central idea: set P replaced by collection of P(i)
  - For each subset policy P(i), there is a TCB subset M(i)
  - The system policy P is the **union** of all P(i)
- Every access request submitted to every subset
  - Every access prohibited in P must exist in some P(i)
- Every prohibited access has to be prohibited by at least one TCB subset

Example: DAC and MAC Subsets

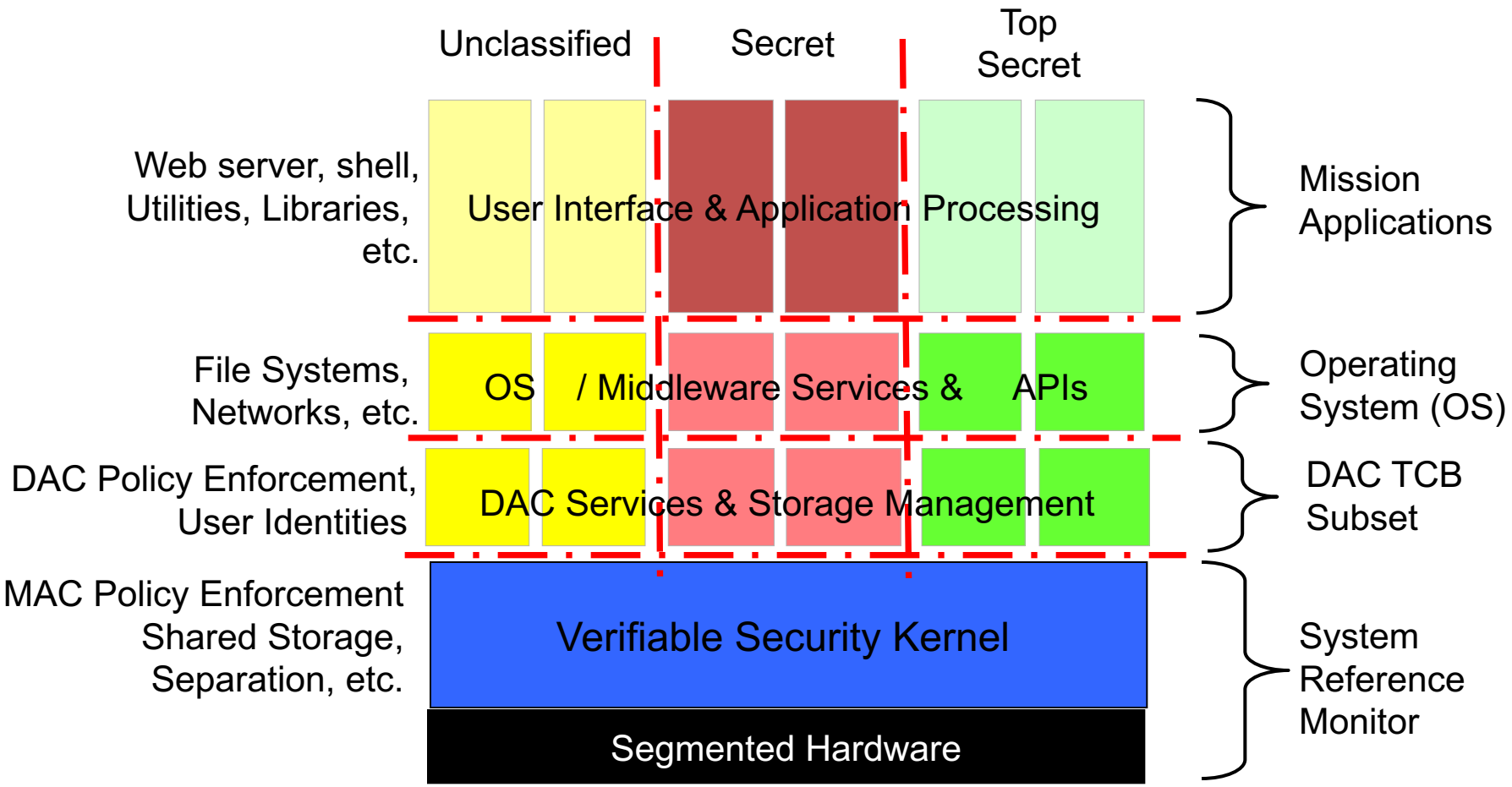
- In BLP:
  - DAC implemented by M
  - MAC implemented by f
  - M can contain some modes prohibited by f
  - f may permit some modes prohibited by M
- Union of all modes prohibited by f and M is overall system access policy

Organization of Collection of Subsets

- Ensure every TCB subset is consulted on every access
  - Do not bypass layers, Why?
- Subsets within series of hierarchical domains
  - Protected objects exported to next less-privileged TCB subset
  - Least privileged subset exports to application domain



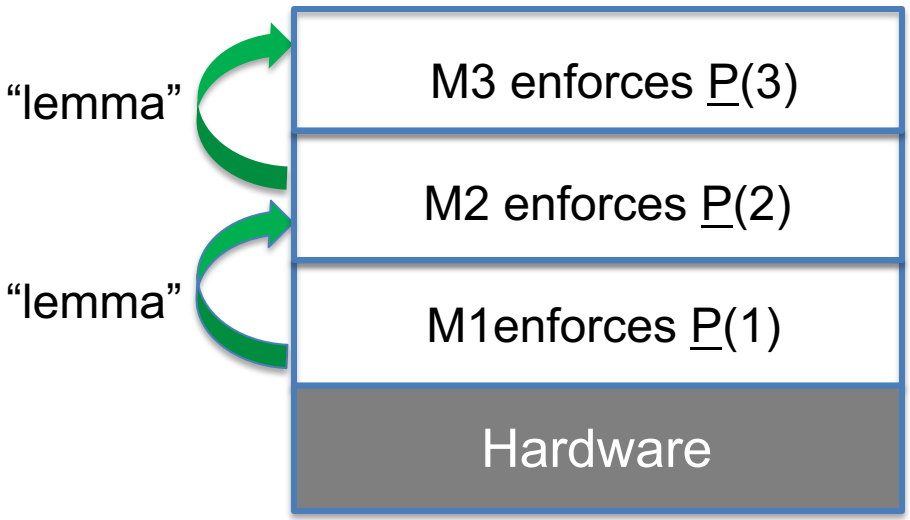
DAC on MAC TCB Subset Example



Correctness of Collection of Subsets

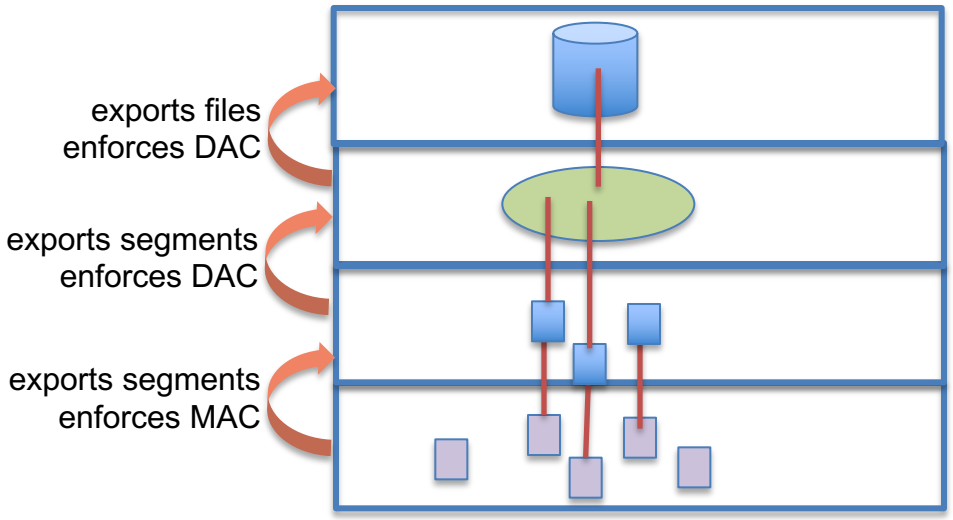
- Complex policy enforced over virtual objects that considerably differ from physical storage object provided by HW
- Incremental evaluation
  - Correctness of a subset depends only on correctness of previous subset
- Exactly one subset does not depend on any other
  - This subset, M(1), must directly access the real system objects and resources (hardware)
  - This is by definition the system reference monitor
  - Means M(1) must be tamperproof WRT any other
- Thus, M(2) must reside in less-privileged domain
  - Argument applies recursively to all remaining subsets

• • •



Subset "Protected Objects"

- Hierarchical subsets enforce policy at interface
- Some objects may be inherited with no change
  - Additional policy restrictions may be enforced
- Some objects are built from primitive underlying objects
  - Often called "interpretively/indirectly accessed" or "abstract"
  - TCB subset interface supports operation on abstract objects
    - e.g., observe, modify a file
- Subset can inherit only some objects from the more privileged subset
  - The more privileged subset can remove some objects for its own use



MLS Databases

- In an MLS database, each object has a classification and maybe a compartment
  - Object can be element, column, row, or aggregate (e.g., view)
- Views can be used for access control
  - A user's view of a database consists of only the data that the user is allowed to access
  - Different users at different levels may get different query results
- **Polyinstantiation** - single relation may be mapped to multiple instances of a row of data to assist users with different security levels

Name	λ_N	Dept	λ_D	Salary	λ_S
Bob	U	Dept1	U	100K	U
Jim	U	Dept1	U	100K	U
Ann	S	Dept2	S	200K	S
Sam	U	Dept1	U	150K	S

Visible to a user with Secret level

Name	λ_N	Dept	λ_D	Salary	λ_S
Bob	U	Dept1	U	100K	U
Jim	U	Dept1	U	100K	U
Sam	U	Dept1	U	–	U

Visible to a user with Unclassified level

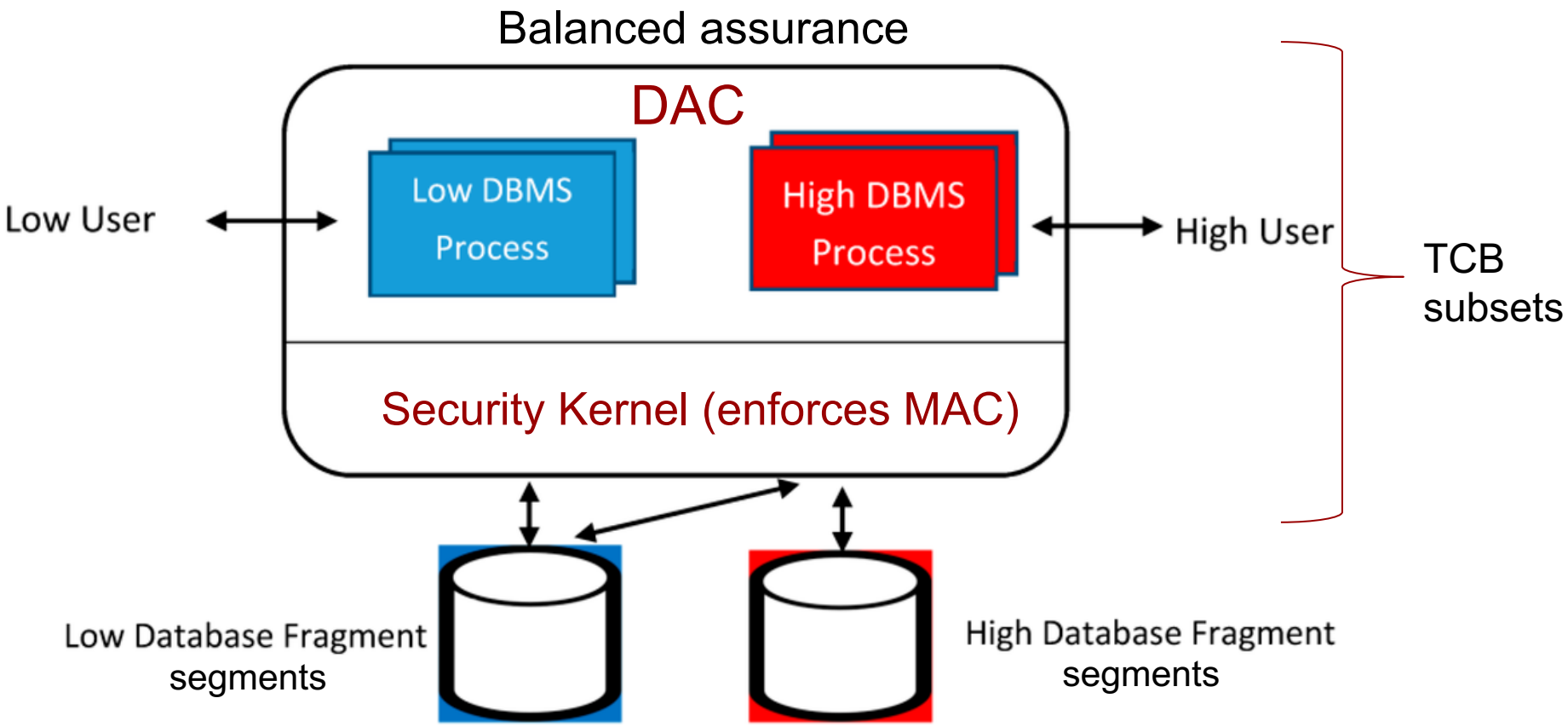
USC Viterbi School of Engineering

Case Study:
High Assurance MLS SeaView DBMS

- Solved persistent hard problem of MLS DBMS
  - Highest level of reference monitor assurance (Class A1)
  - Element-level labels for most flexible functionality
- Preserves major properties of relational DB model
- Decompose all MLS relations to single-level objects
  - Supports content and context dependent operations
  - Reconstruct original MLS relation from fragments; has to yield same results
- Achieving Class A1 assurance for a multilevel database system is possible only if the portion of the system enforcing mandatory security is small and isolated and does not depend on the large database machinery for its correct enforcement

SeaView TCB Subset Architecture

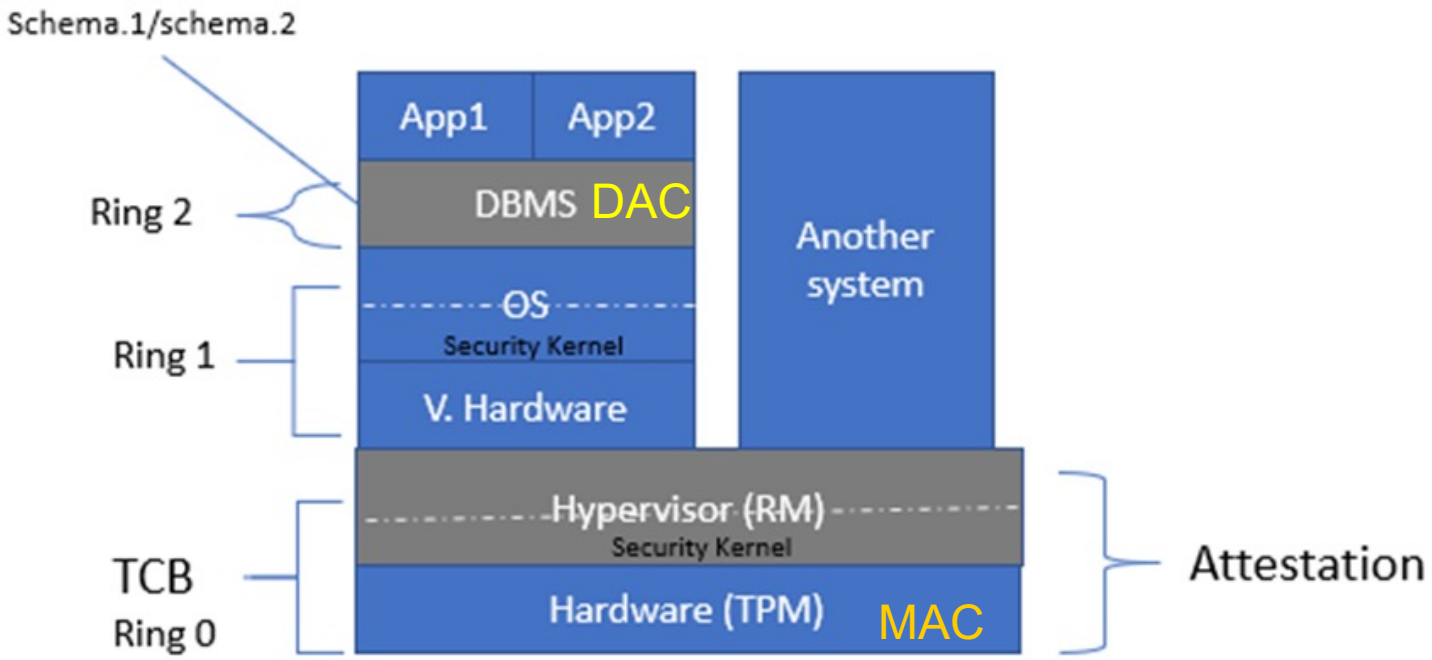
- Relations stored in segments (real objects) and enforce DAC on columns of a relation (virtual objects)



More details: Linda L. Vetter, Gordon Smith, Teresa F. Lunt: TCB subsets: the next step. ACSAC 1989

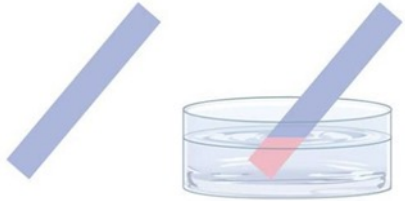
Question

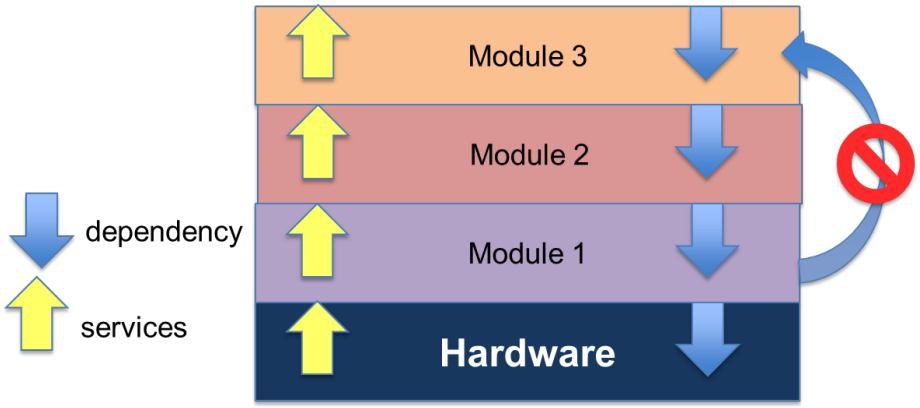
- Is the deployment method shown in the picture correct? Why or why not?

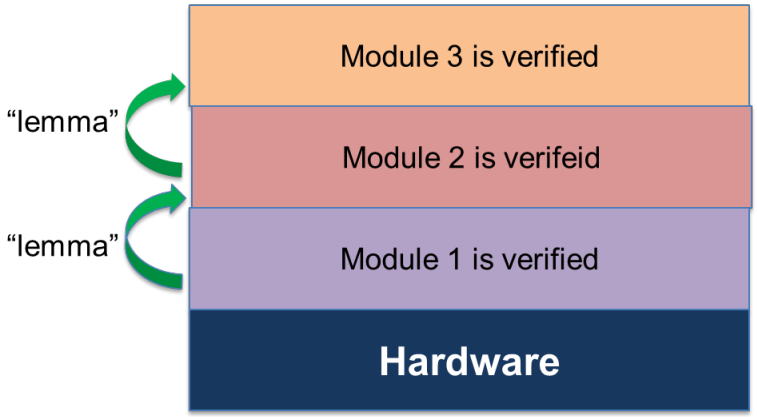


Trusted OS: Loop-free Independent Module Layers

- Litmus test of “high assurance” OS kernel
  - Separate module for each distinct function
  - Each module has precisely defined interface
- Each module is part of a single loop free layer
  - Each layer uses only facilities of lower layers
    - Subject to restrictions of all lower layers
  - Module services available to any module above it
  - Experience shows OS kernel needs about a dozen layers







Linux global table demonstrates bad layering

Composition State of the Art

- **Only two** validated composition methods
  - Require tight constraints on the system architecture
  - Known as (1) Partitioned TCB and (2) TCB Subset
- We defined **TCB subsets** – for single computer
  - Leverage hierarchical domains ordered by privilege
  - Each TCB subset resides in an individual **execution domain**
  - Most privileged domain (e.g., ring 0) is security kernel
  - System policy is the union of all TCB Subset policies
  - Decomposing a policy into subsets is “art” not science
- **Partitioned TCB** for network of computers
  - Constrained to address loosely connected cases

Policy Composition Strategy Summary

- "Divide and conquer" strategy for policy
  - Divide system policy into disjoint security policy parts
  - Must prove they compose into a single system policy
- Divide the system TCB into simpler parts
  - Verify each part (subset) separately
  - Compose parts to enforce a single system policy
- Each part of the TCB enforces part of a policy
  - We say a part of policy is **allocated** to a part of TCB
  - Must verify that each part enforces its policy
  - Must validate parts are composed in correct way
- Each subset meets reference monitor conditions

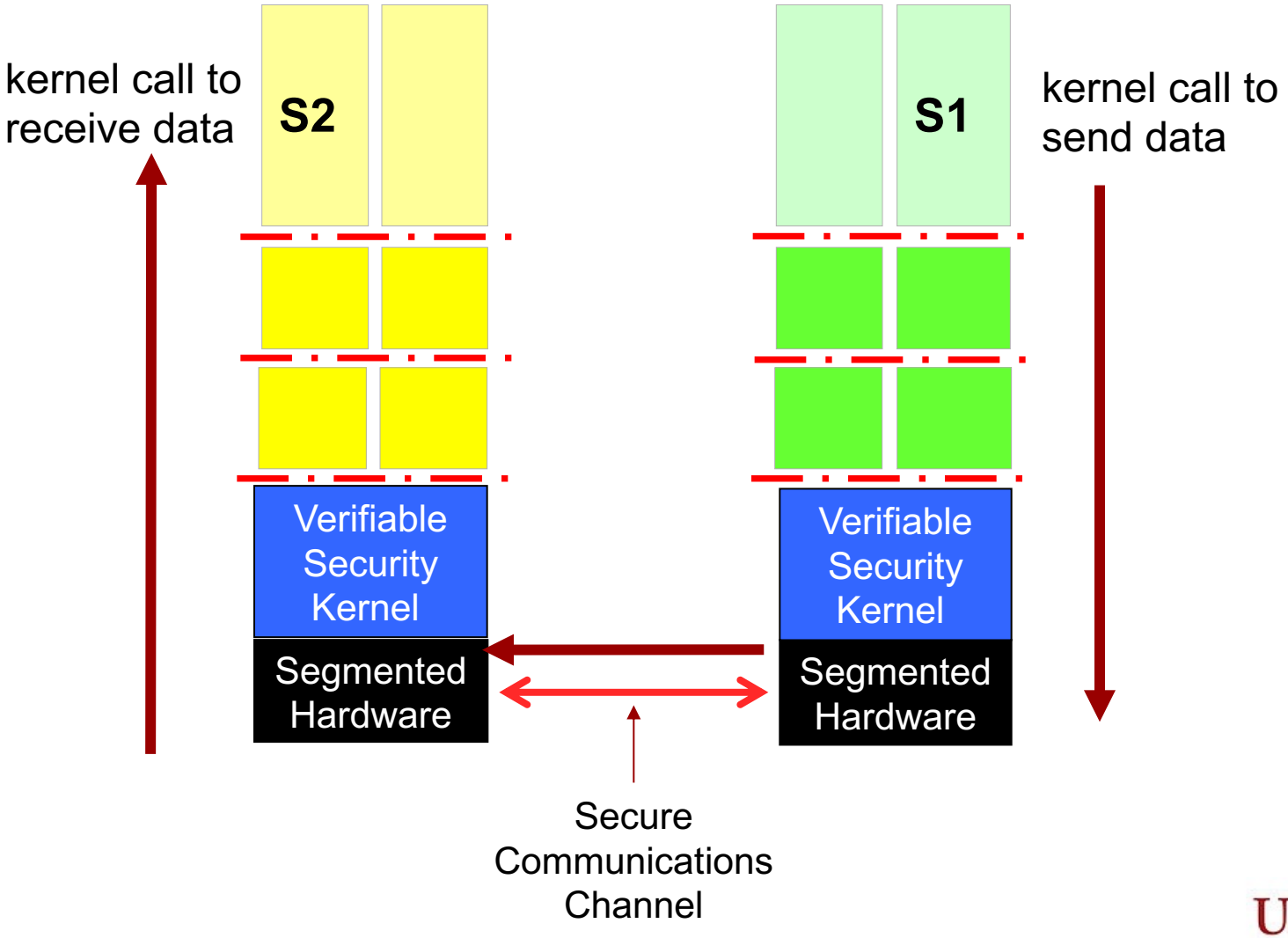
Outline

- ORCON
- Policy composition
  - Intro
  - TCB subsets
  - TCB partitions
- TNI Composition of MAID Components

Partitioned TCB Introduction

- Network interpretation of RM using Trusted Network Interpretation (TNI) concepts
- For network of loosely connected "**components**"
  - Computers and communication devices
- Subsets alone do not address such a network
  - Important to view network as a single system
  - Important to have single network security policy
- A single Trusted Computing Base (TCB) for net
  - Called the Network Trusted Computing Base (NTCB)
  - Physically & logically partitioned among components
- Each TCB partition enforces its allocated policy
  - How to cleanly partition net into components is "art"
  - Network security architecture and design

NTCB Components

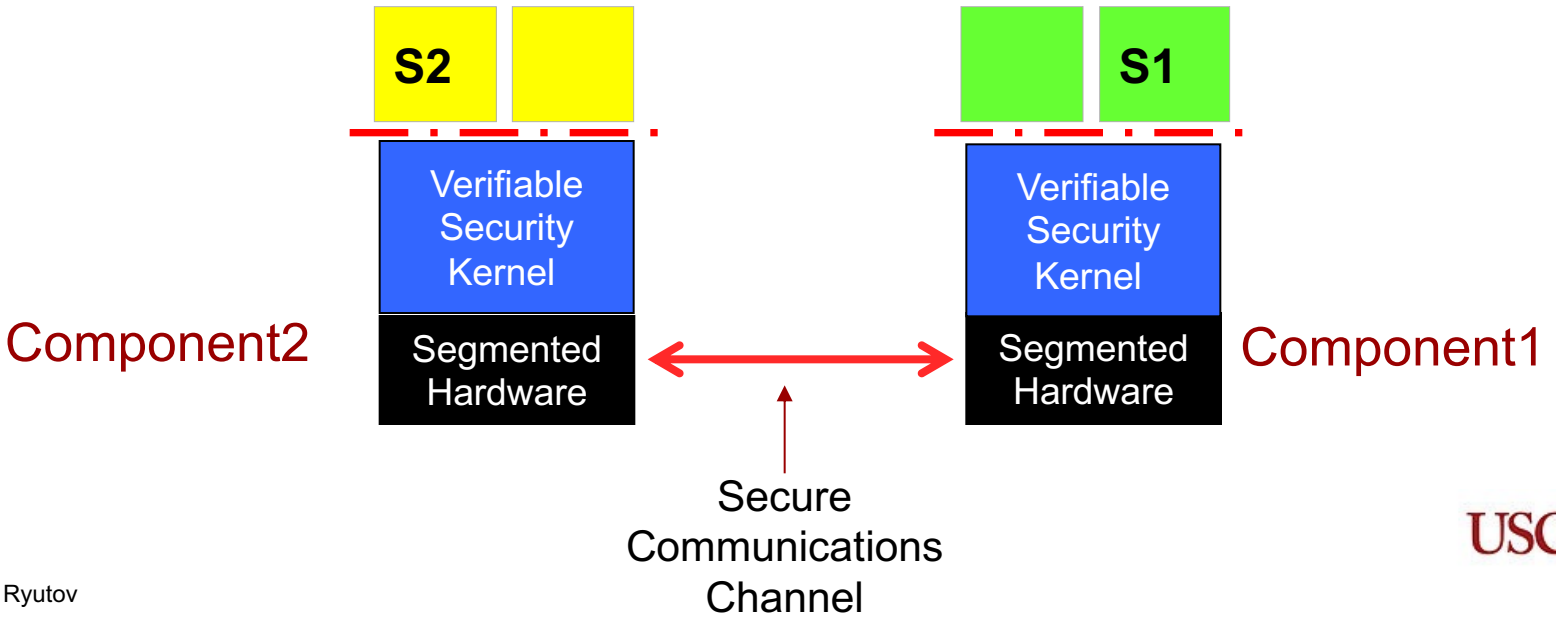


Ideal Communication Channels

- Ideal communications channels are secure
  - Do not compromise security of information entrusted
  - Build perfect network first!
- Subjects on component cause info exchange
  - point-to-point communication, disregard network complexity
- Channels maintain proper associations between
  - Labels – connections may be multilevel
  - User identifications
  - Clearances
- Each component must translate its labels

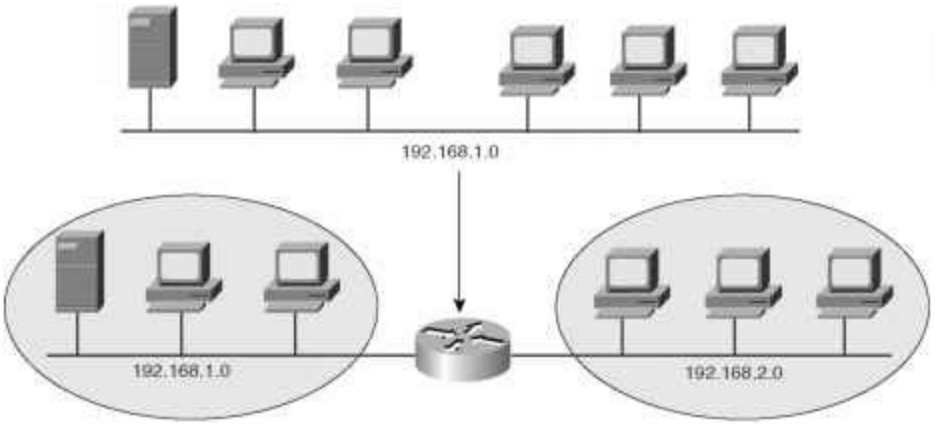
Confinement of Subjects and Objects

- Each TCB partition enforces it's policy, subjects and objects must be disjoint from other TCBs
  - Otherwise cannot enforce policy independently of other TCB partitions
- Subject is confined to single network component
- Subjects access objects within same network component
- Cannot "move" subjects or objects from one component to another
  - "remote process" is subject on a different component, e.g., RPC call
  - "transferred object" is a new object on another component that contains the same info, e.g., FTP



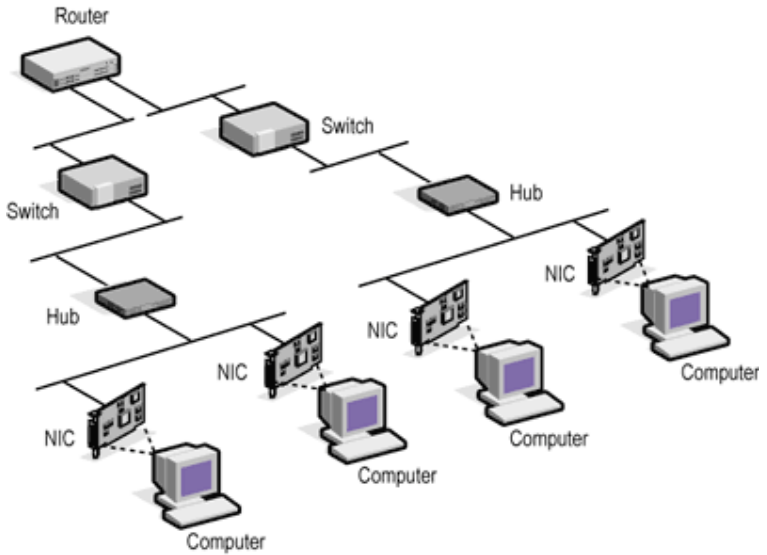
Internal Subjects

- Subjects are either
  - surrogates for users or
  - internal subjects (e.g., OS demons)
- Internal subjects provide services
  - E.g., services for communications between users and the system, such as protocol handlers
  - Some components may only have internal subjects
    - E.g., packet switch, router
- Since no user surrogate don't need DAC
  - Organizational policy only applies to individuals



Local Nature of Objects

- Subjects access objects on the same component
  - All objects for the system are in disjoint partitions
  - No objects are shared between components
- Information is transmitted between components
  - Have “ideal communications channel” between them
  - Information transferred from one device to another
    - Without existence of an intermediate object
  - No subject can access information being transmitted
- Communication is viewed as an operation that copies information from an object at one end of a communication path to an object at the other end



Partitioned Access Control Policy

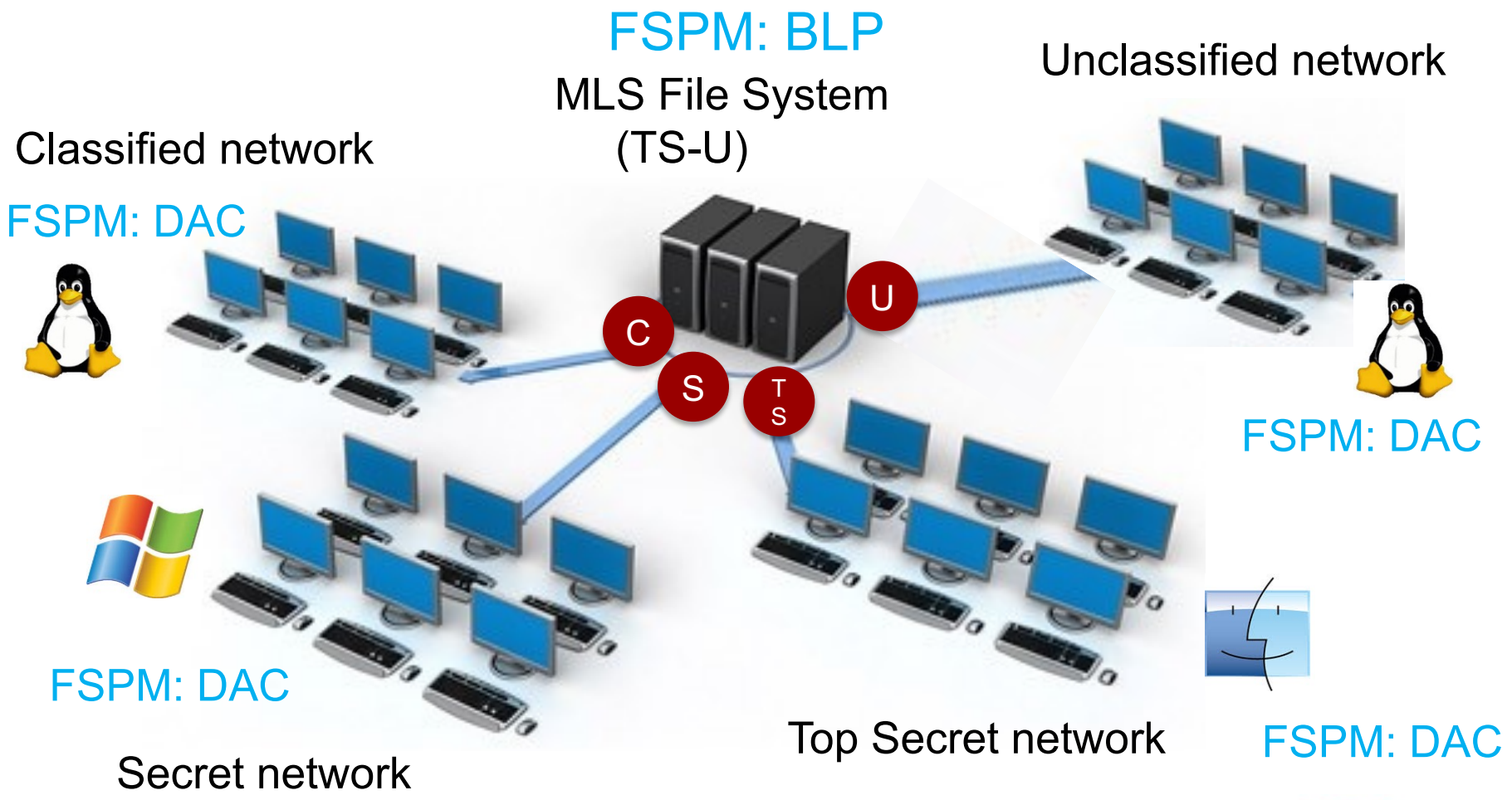
- Policy formulation similar to that for subsets
- System policy P is a set of triples <s, o, m>
  - s is a particular subject
  - o is a particular object
  - m is a particular access mode which is prohibited
- Policy is represented as **prohibited** accesses
- Central idea: set P replaced by collection of P(i)
  - Each partition has a unique policy subset P(i)
  - The system policy P is the union of all P(i)
- An access request is submitted to some partition
  - Every partition enforces the P(i) allocated to it

Constructing TCB Partitions

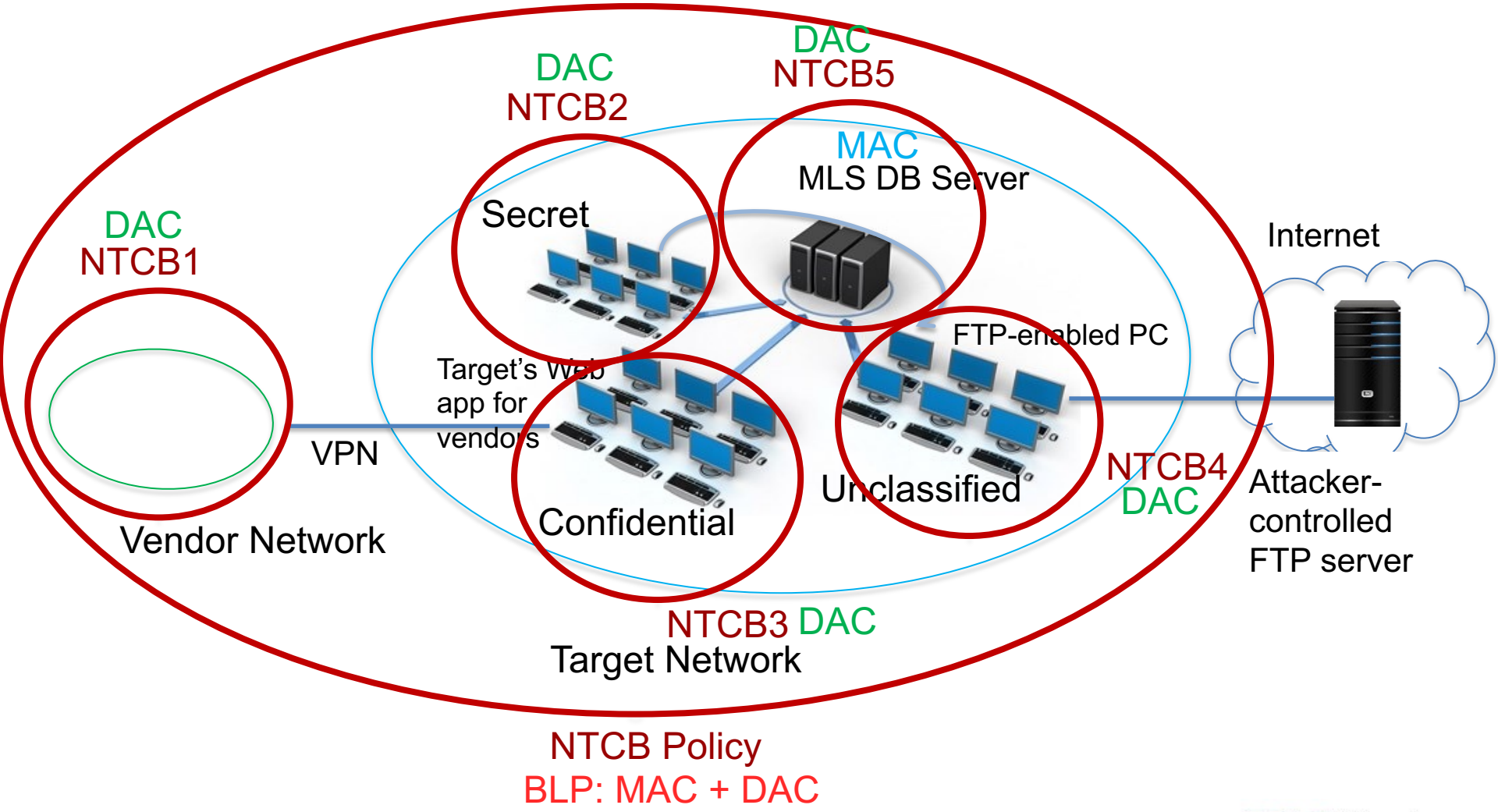
- Key is properly formulating policy to be enforced
  - Need to adopt single, uniform network security policy
  - Must be the policy for the complete NTCB to enforce
  - It is distinct from formal security policy model (FSPM)
    - Each component may have different FSPM!
- Policy must control the establishment of authorized connections across the network
- Policy implies a "distributed" reference monitor for entire net
  - Collection of individual component security kernels
  - Implement as locally autonomous reference monitors
- Every component contains a reference monitor
  - Abstraction for subjects and objects on component
  - Enforces allocated access control policy for them
  - Meets RM properties
- All TCB partitions are evaluated independently
  - Correctness of one **does not depend** on the other

MLS Network Example

Does this system as a whole effectively behave like single MLS system?



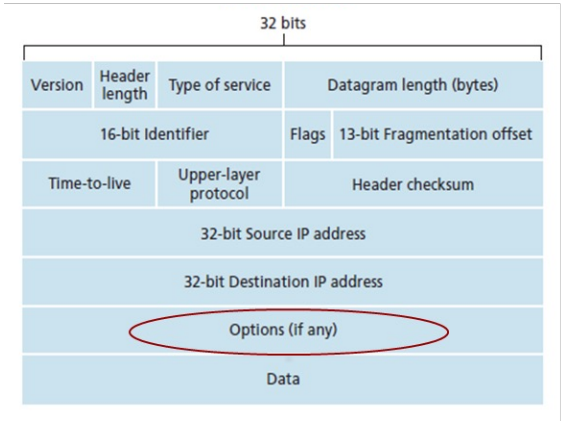
MLS Network Example



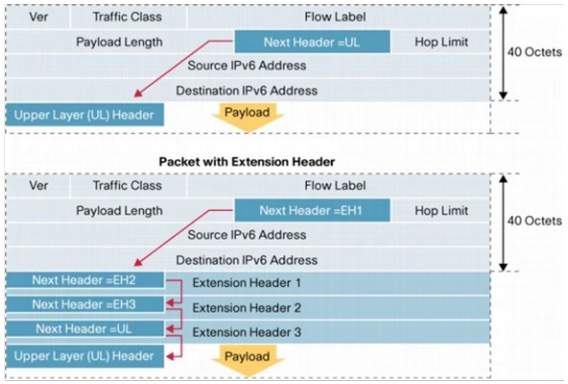
How can we implement this?

- Example: low assurance PitBull Network Polyinstantiation
- PitBull provides polyinstantiated network labeling
  - Implemented at a kernel network-stack level
  - Polyinstantiation isolates data to prevent data from bypassing MAC enforcement
  - Allows software to share a single network port at multiple Sensitivity Levels and Categories
  - Network Traffic Labeling (**`netrule`** command)

IPv4 header



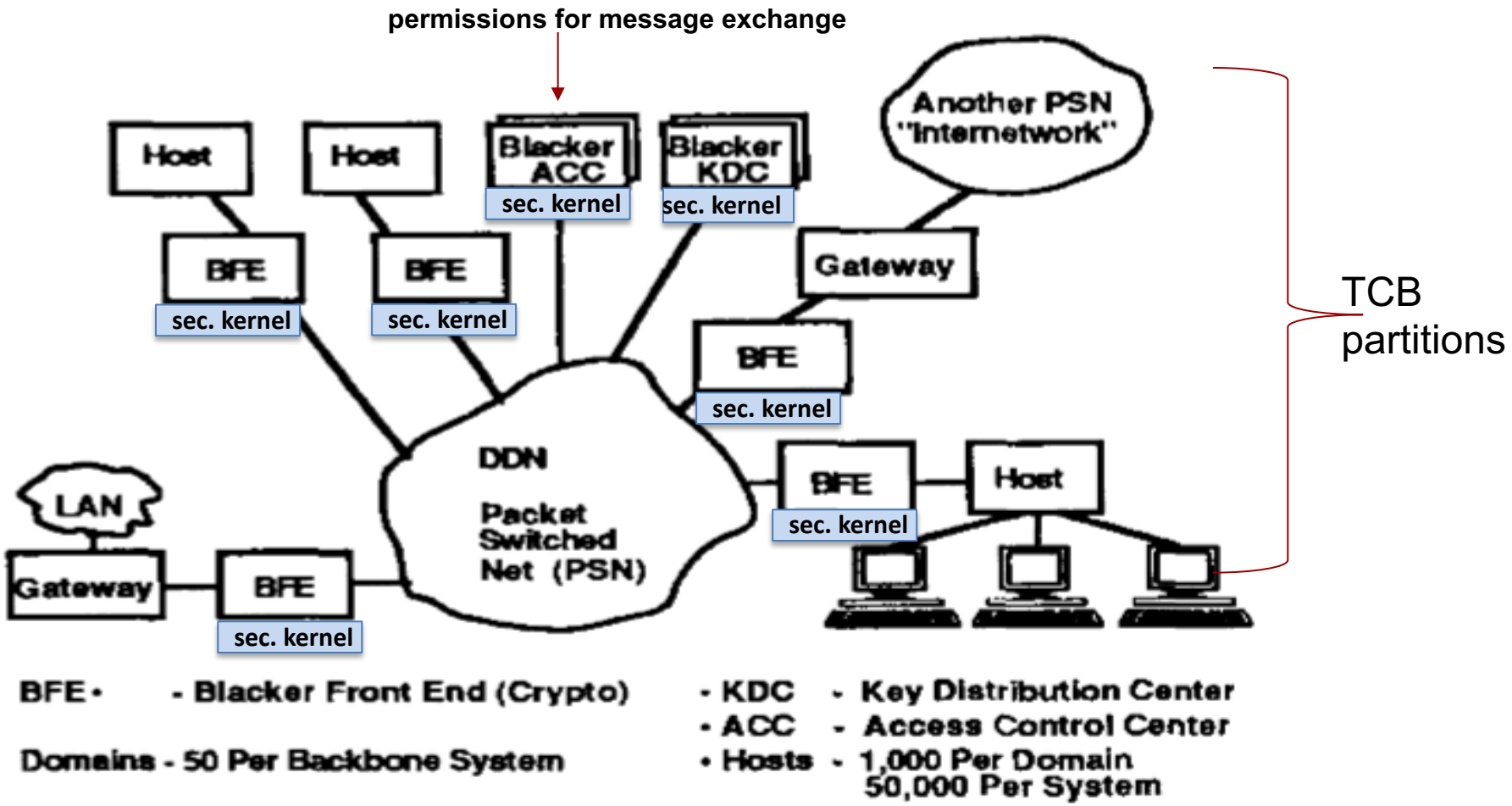
IPv6 header



Case Study: Blacker System

- **It can be done!**
- Class A1 BLACKER system complex and large (up to 50,000 hosts) distributed system
  - Essentially MLS virtual private network for government TCP/IP networks
- The first secure system with trusted End-to-End encryption using government encryption
- The security of the entire system had to be evaluated, not just individual pieces
- Demonstration of the power of the partitioned TCB
- GEMSOS security kernel and trusted components used as the basis for trust for critical cryptographic and key distribution functions
- Each of the four BLACKER devices are a TNI component in the **partitioned TCB**
- Bell-LaPadula security model
  - host computer served as a subject, encrypted network was the object

BLACKER Defense Data Network



Summary of NTCB Partitions

- NTCB is cooperating, loosely-coupled partitions
  - Includes routers and other network devices
- All functions of the NTCB must be allocated
  - In some coherent way to the various components
- Network components & channels are exhaustive
  - Parts are disjoint (none shared between components)
- Access mediated by component security kernel
  - Totality of security kernels mediates all accesses
- Partitioning may be applied recursively
  - A single TCB partition may itself be a NTCB
- A network-wide Security Policy is enforced

Outline

- ORCON
- Policy composition
  - Intro
  - TCB subsets
  - TCB partitions
- TNI Composition of MAID Components

Review: RM, RVM, TCB, and Security Kernel

- RM is an abstract concept: an abstract machine that mediates all access to objects by subjects
- A reference validation mechanism (RVM) is an implementation of the RM concept
- Security kernel is the hardware, firmware, and software of a TCB that implements RM concept, example of RVM
- TCB (trusted computing base) is the totality of protection mechanisms within a computer system that work together to enforce a security policy
- In general: RM implementation $\subseteq$ security kernel $\subseteq$ TCB

Nature of Supporting Policies

- Supporting policy "supports" the tie of people in access control policies
- Identification/Authentication and Audit are Supporting Policies
  - No "theory" that helps verify the implementation of supporting policies
- Provide an environment for basic access policies
  - Allows effective enforcement and monitoring
- May be enforced **outside** the security kernel
  - Some services within local component, e.g., login
  - Some are network-related, e.g., protocols in-between

TNI Taxonomy of Four Policy Elements

- Access control policies
  - Mandatory access control (MAC) – Designated "M"
  - Discretionary access control (DAC) – Designated "D"
- Supporting policies
  - Identification and authentication – Designated "I"
  - Audit – Designated "A"
- Components in taxonomy combine from "MAID"
  - Sixteen combinations:
    - {null}, M, A, I, D, MA, MI, MD, AI, AD, ID, MAI, MAD, MID, AID, and MAID

USC Viterbi School of Engineering

Network Security Architecture Issues

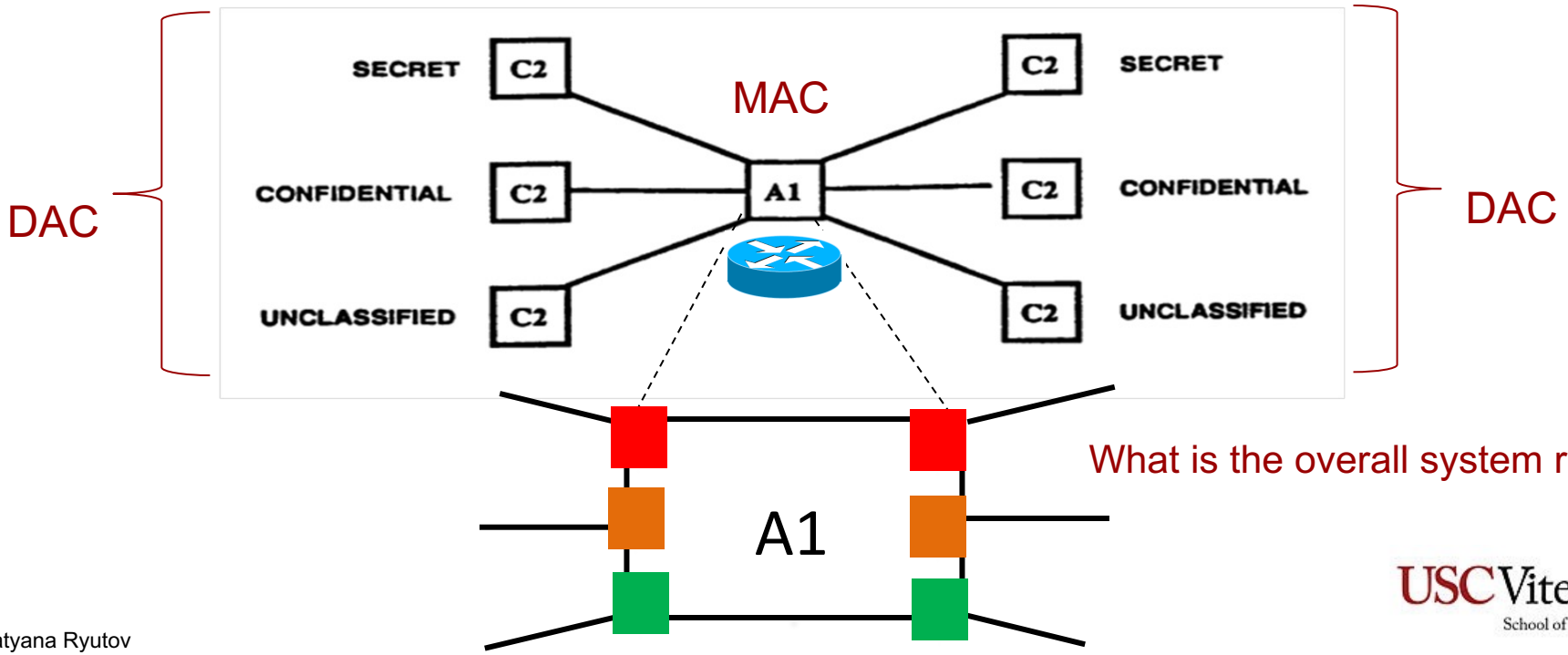
- How is the network to be partitioned [TNI App B]
  - Ensure eventual policy verification of entire network
  - Rationale theory "by construction" for multiprocessor
    - processes running on separate processors and communicating via shared memory
    - replace shared memory with ideal network.
- Basis for verifying component enforces its policy
  - Decompose policy into policy elements for component
    - Access control policies, i.e., MAC and DAC
    - Supporting policies – environment for access control policies
- Verifying policy enforcement of composition, including
  - M-component, D-component, I-component, and A-component

Mandatory – M-component

- Need support environment for MAC in network
- Composing two directly connected components
  - Must consider physical security of medium
  - Must consider transmission communication security
- Non-M-Component is **directly** connected
  - M-component end has single-level device
  - Level associated with two devices must be the same

M-Component: High Assurance Experience

- M-component is dominant choice of type
- Balanced assurance is major motivation
  - Recognize inherent robustness limitations of DAC
  - A, I, and D elements of policy as lower assurance
  - MAC components must be in more privileged domain



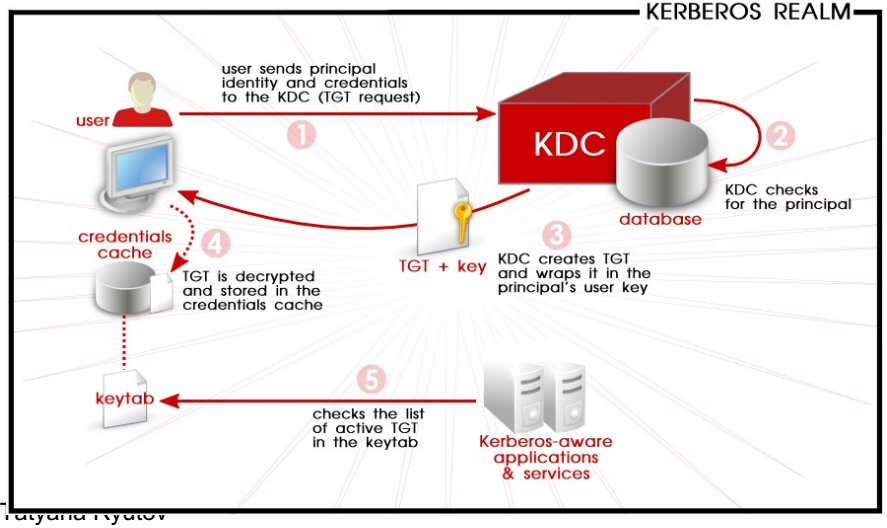
What is the overall system rating?

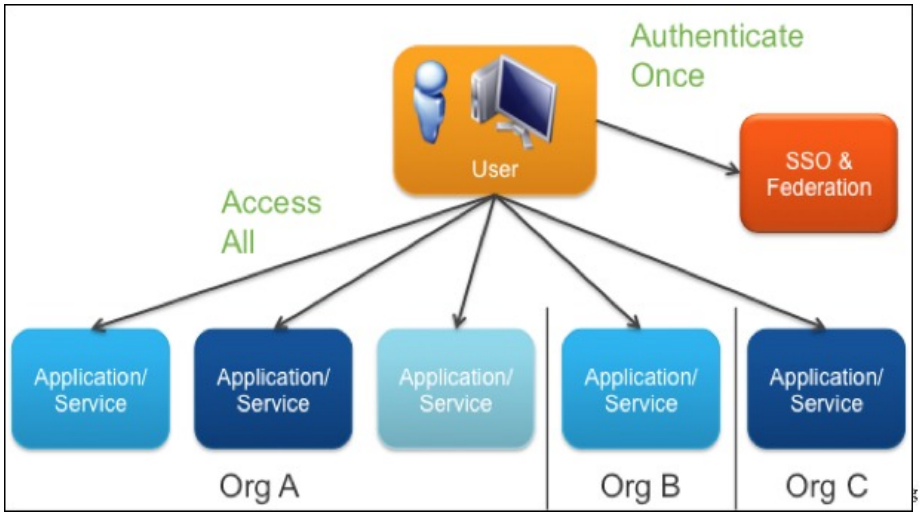
Discretionary – D-component

- Need support environment for DAC in network
  - Physical and transmission security for medium
- Composing two directly connected components
  - Passed identifiers used as basis for DAC decisions
  - Could map each user into:
    - a single network ID
    - a specific group of users (e.g., RBAC)
- Composed DAC enforces network DAC Policy
  - E.g., USC NetID

Identification & Authentication – I-component

- I-component enforces its allocated policy
  - Network identification-authentication policy
- Need statement of the supported protocol
  - Communicating user identification
  - Interfaces provided by the I-component
- Can compose an I-component with others
  - Significant engineering and system architectural work
- Policy assurance is lowest of any I-component



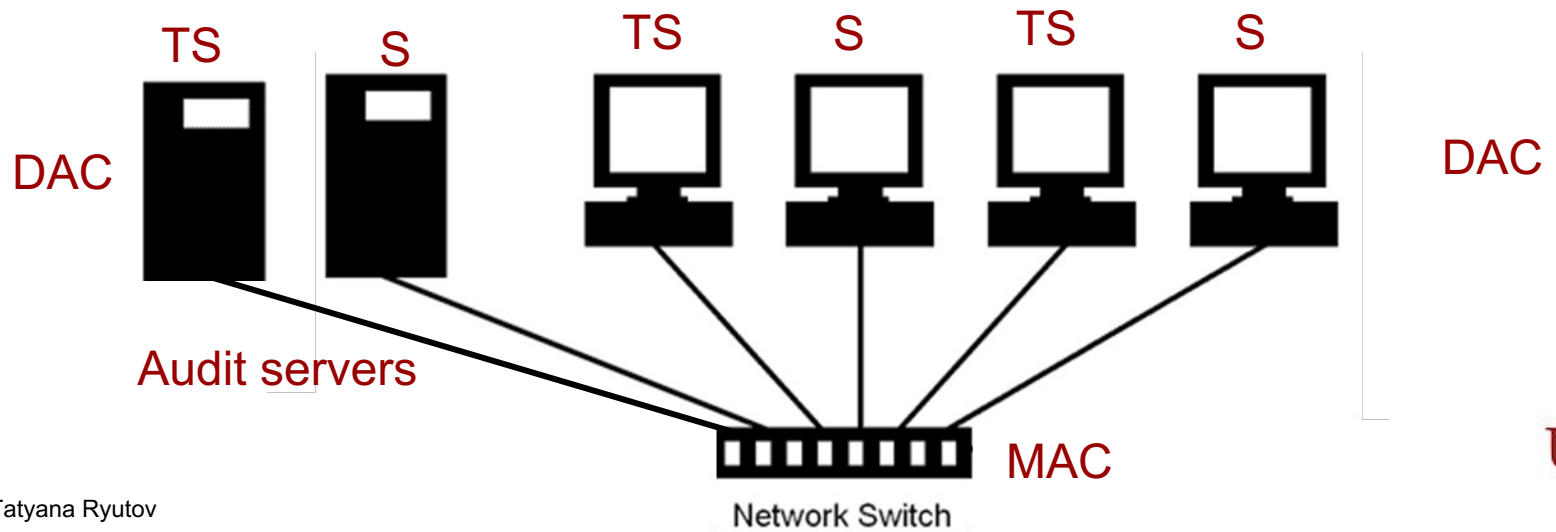


Audit – A-component

- A-component enforces its allocated audit policy
  - A partition of network audit policy – for accountability
  - May include real-time alarms for audit events
- Statement of the supported audit protocol
  - Data generated by actions on other components
- Can compose an A-component with others
  - May provide audit services to other components
  - Strong integrity policy for audit file
- Physical and transmission security for medium

TNI A-Component: Example

- MLS packet switch
  - Verified security kernel, enforces MAC
- Can configure several single-level channels
  - Single-level channels are either Top Secret or Secret
  - Single-level channels directly connected to C2 hosts (DAC)
    - Some hosts running at dedicated Secret
    - Some hosts running at dedicated Top Secret
- C2 hosts collect audit records from M-Component
  - At least one Secret host for Secret records
  - A least one Top Secret host for Top Secret records



Summary of TNI "MAID" Composition

- TNI MAID Taxonomy is not unique
  - Has endured for decades as sufficiently practical
  - Few careful, systematic alternatives have emerged
- Is neither more nor less than TCSEC policy
  - Considered "operationally complete and consistent"
- Supports both known composition technologies
  - TCB subsets can be in any single component
  - TCB partition is on any policy-enforcing component