Security Architecture and Design

Domain 6

Overview

The security architecture and design domain addresses the concepts, principles, structures, frameworks, and standards used to design, implement, monitor, and secure individual systems and the IT enterprise.

This domain focuses on the architecture of security services used to enforce various levels of confidentiality, integrity, and availability in key information assets.

Key Areas of Knowledge

A. Understand the fundamental concepts of security models (e.g., Confidentiality, Integrity, and Multi-level Models)

B. Understand the components of information systems security evaluation models

B.1 Product evaluation models (e.g., common criteria)

B.2 Industry and international security implementation guidelines (e.g., PCI – DSS, ISO)

C. Understand Security Capabilities of Information Systems (E.G., Memory Protection, Virtualization, Trusted Platform Model)

D. Understand the vulnerabilities of security architectures

D.1 System (e.g., covert channels, state attacks, emanations)

D.2 Technology and process integration (e.g., single point of failure, service oriented architecture)

Key Areas of Knowledge(continued)

E. Understand software and system vulnerabilities and threats

E.1 Web-based (e.g., XML, SAML, OWASP)

E.2 Client-based (e.g., applets)

E.3 Server-based (e.g., data flow control)

E.4 Database security (e.g., inference, aggregation, data mining, warehousing)

E.5 Distributed systems (e.g., cloud computing, grid computing, peer to peer)

F. Understand countermeasure principles (e.g., defense in depth)

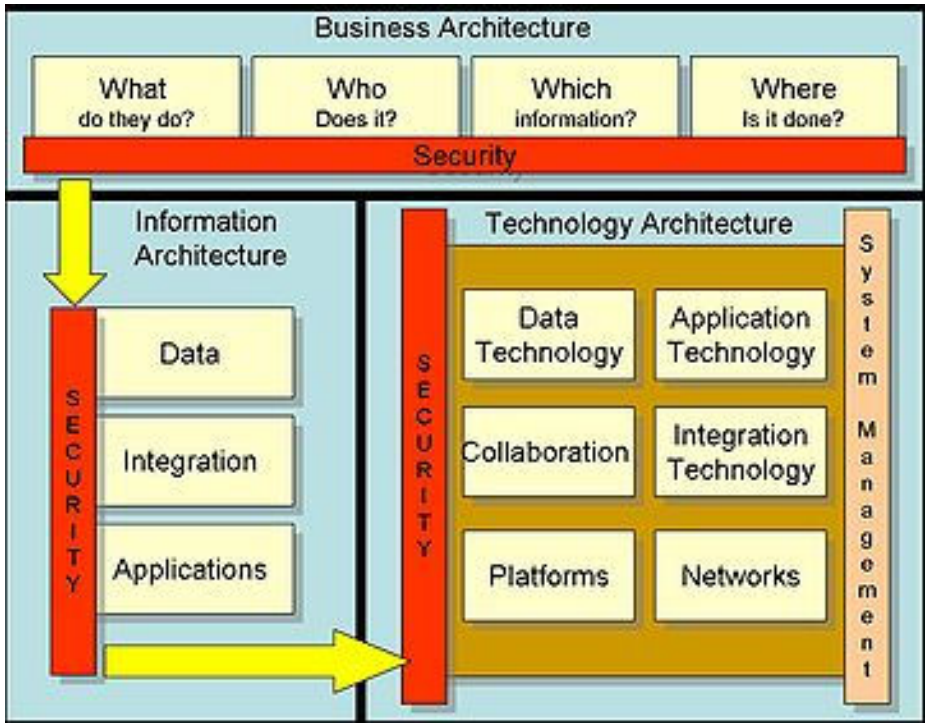
Agenda

- **Fundamental concepts of security models**
- Components of information systems security evaluation models
- Security capabilities of information systems
- Vulnerabilities of security architectures
- Software and system vulnerabilities and threats
- Countermeasure principles

Security Definitions

Security Architecture

- The practice of applying a comprehensive and rigorous method for describing a current and/or future structure and behavior for an organization's security processes, information security systems, personnel and organizational sub-units, so that they align with the organization's core goals and strategic direction.



Security Definitions

Framework

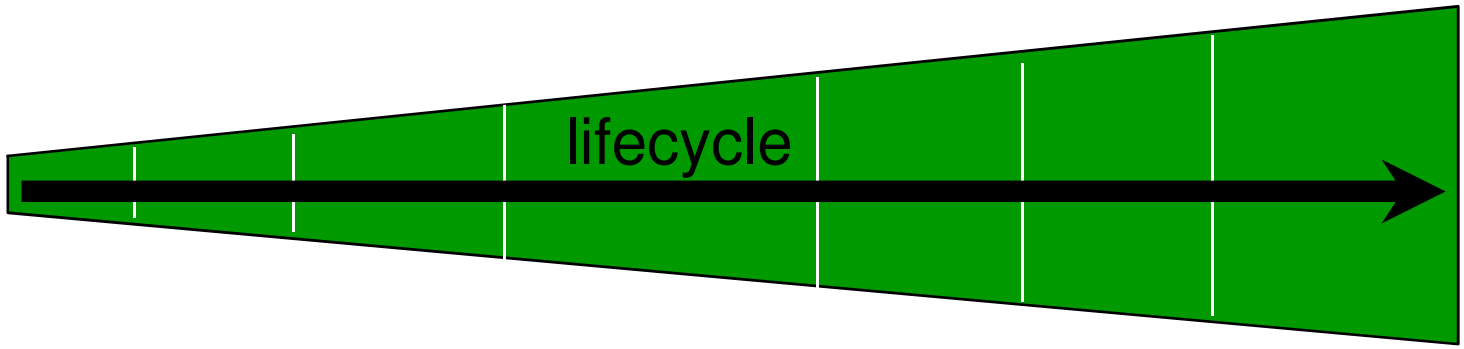
- A defined approach to the process used to achieve the goals of an architecture, based on policy, and reflecting the requirements and expectations of the various stakeholders.

Blueprint

- The functional definition for the integration and development of technology infrastructure into the business process.

Why Architect in the First Place?

- **Solutions are the result of concerted efforts**
- **Saves Money**
  - fixing up front cheaper than fixing in the field
- **Improves Quality and Productivity**
  - re-use is good, architecture is about re-use



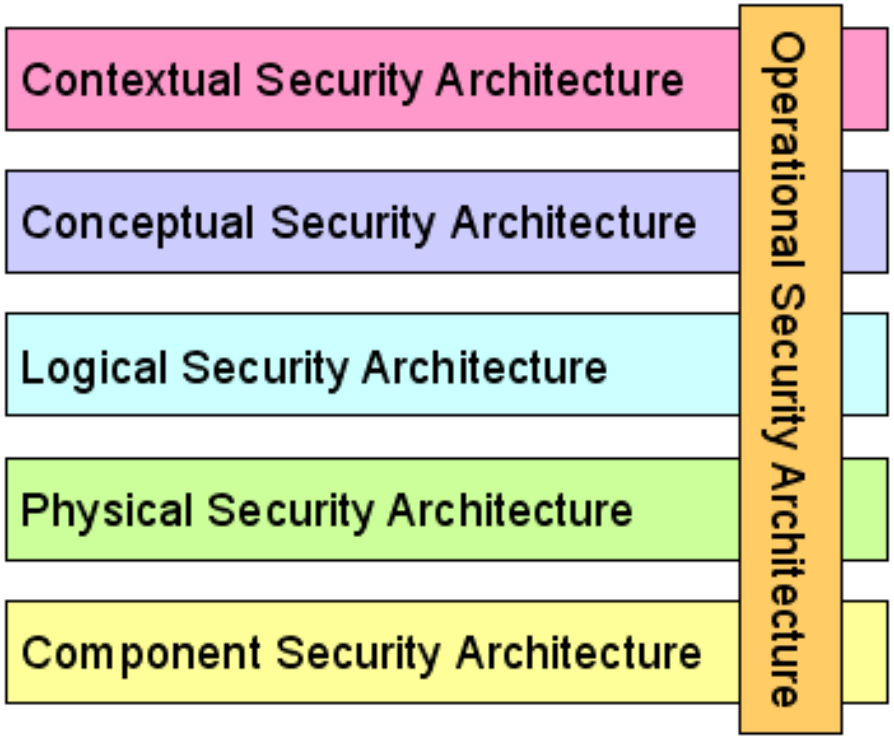
Common Architecture Frameworks

Zachman Framework

	Why	How	What	Who	Where	When
Contextual	Goal List	Process List	Material List	Organizational Unit & Role List	Geographical Locations List	Event List
Conceptual	Goal Relationship	Process Model	Entity Relationship Model	Organizational Unit & Role Rel. Model	Locations Model	Event Model
Logical	Rules Diagram	Process Diagram	Data Model Diagram	Role relationship Diagram	Locations Diagram	Event Diagram
Physical	Rules Specification	Process Function Specification	Data Entity Specification	Role Specification	Location Specification	Event Specification
Detailed	Rules Details	Process Details	Data Details	Role Details	Location details	Event Details

Common Architecture Frameworks (cont.)

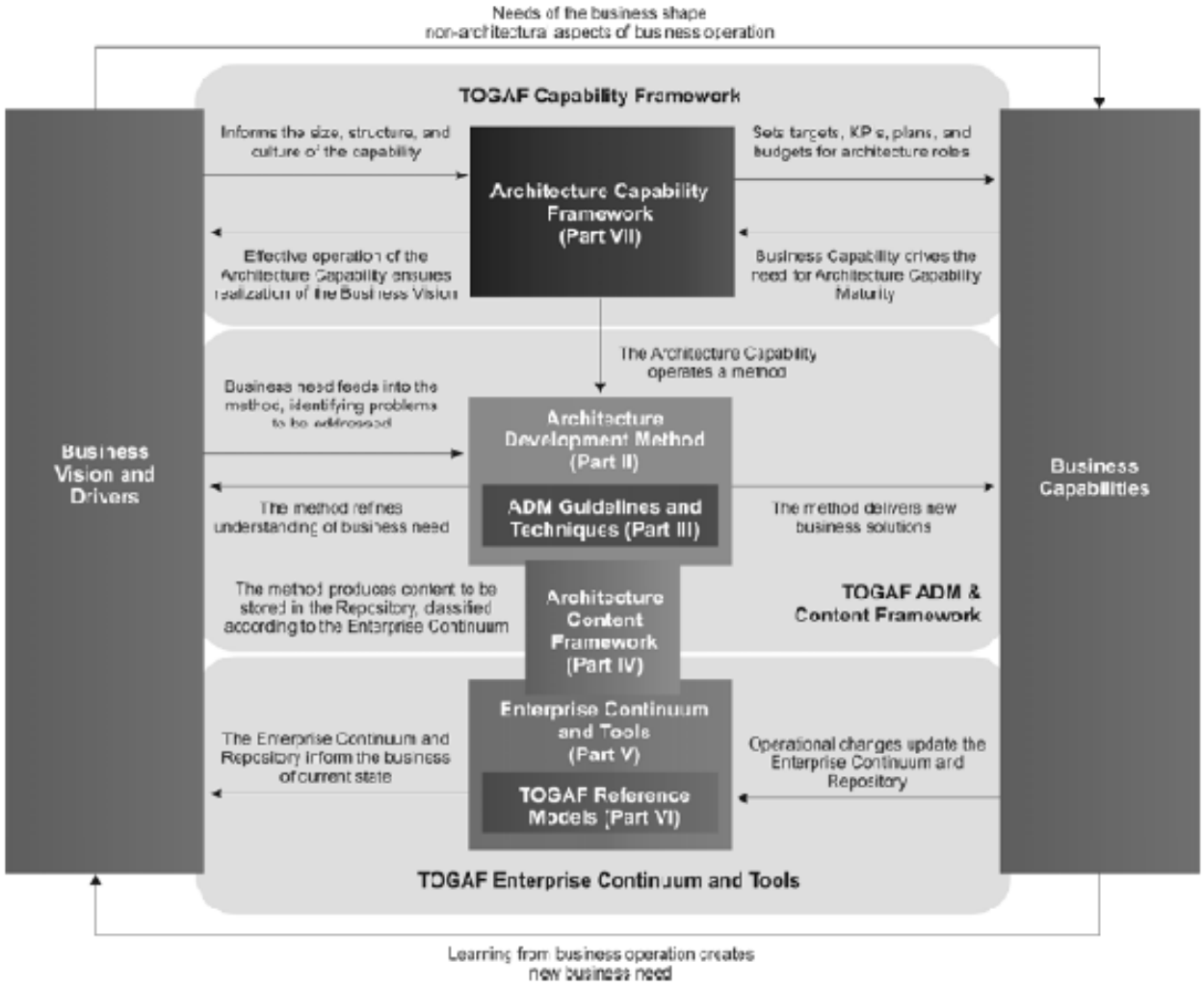
Sherwood Applied Business Security Architecture (SABSA) Framework



The SABSA Model for Security Architecture Development

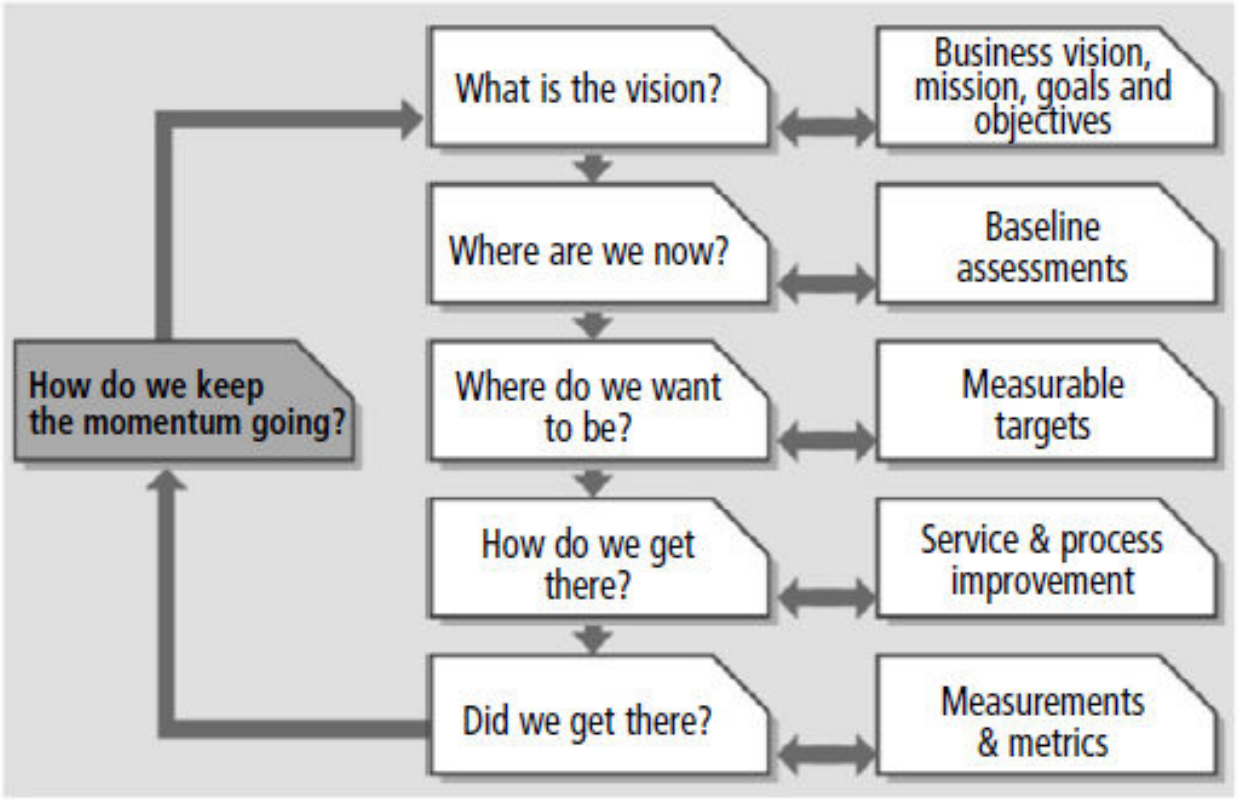
Common Architecture Frameworks (cont.)

The Open Group Architecture Framework (TOGAF)



Common Architecture Frameworks (cont.)

ITIL Continual Service Improvement Model



Creating and Documenting Security Architecture

Once the requirements have been captured and signed off, the security architect can create suitable designs based on those requirements.

Following the SABSA framework from the previous slide, a complete security architecture can be represented by six layers of designs.

Contextual security architecture	The business view: assets to be protected in context.
Conceptual security architecture	The architect's view: high-level view of services to protect the assets.
Logical security architecture	The designer's view: node level view of the services showing how services will be deployed and how they relate to each other at a high level.
Physical security architecture	The builder's view: detailed, node level view of all services and how they will be deployed against physical assets.
Component security architecture	The tradesman's view: component view of individual security services.
Operational security architecture	The facility manager's view: security operations view of all security services in scope.

ISO/IEC 27001

ISO/IEC 27001 has at its core five key focus areas.

1) General requirements of the ISMS
2) Management responsibility
3) Internal ISMS audits
4) Management review of the ISMS
5) ISMS improvement

ISO/IEC 27002

While ISO/IEC 27001 focuses on security governance, ISO/IEC 27002 provides a "Code of Practice for Information Security Management." It has 11 focus areas.

1) Security policy
2) Organization and Information Security
3) Asset Management
4) Human Resources Security
5) Physical and Environmental Security
6) Communications and Operations Management
7) Access Control
8) Information Systems Acquisitions, Development, and Maintenance
9) Information Security Incident Management
10) Business Continuity Management
11) Compliance

Control Objects for Information and Related Technology (COBIT)

COBIT is a framework for IT management which was created by the Information Systems Audit and Control Association (ISACA), and the IT Governance Institute (ITGI). COBIT provides a set of generally accepted processes to assist in maximizing the benefits derived through the use of information technology (IT) and developing appropriate IT governance.

COBIT documents 34 high level processes that cover 210 control objectives categorized in four domains:

1) Planning and Organization (11 high-level processes)
2) Acquisition and Implementation (6 high-level processes)
3) Delivery and Support (13 high-level processes)
4) Monitoring and Evaluation (4 high-level processes)

Common Formal Security Models

A formal security model describes and verifies the ability to enforce security policy and mathematical terms. In general, most security models will focus on defining allowed interactions between subjects and objects at a particular moment in time. The three following types of security models approach the problem in slightly different ways.

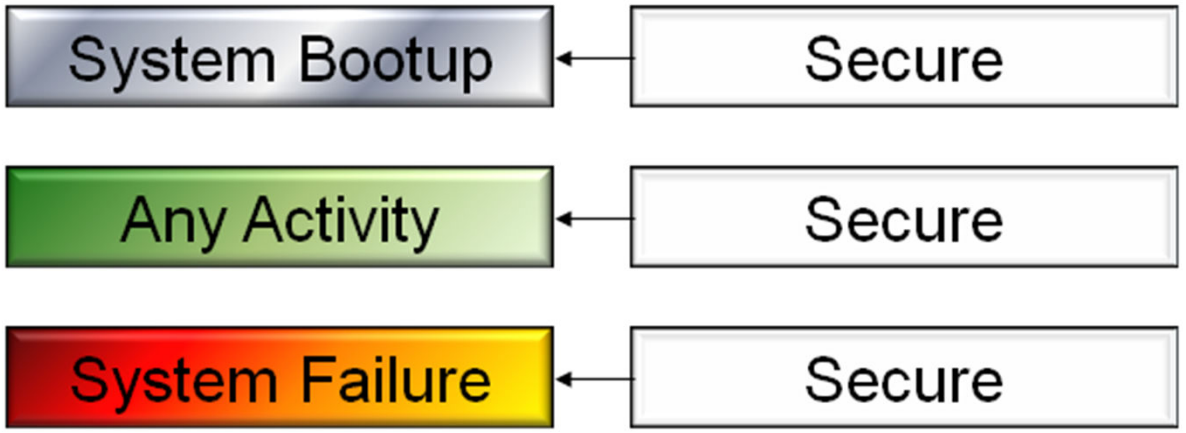
1) State Machine Model
2) Multilevel Lattice Model
3) Noninterference Models
4) Information Flow Models

State Machine Model

- **Deals with states**
  - Set of possible initial states
- **For each possible initial state, there is an execution sequence for each possible state transformation**
- **Outputs and next state of system is dependent upon inputs and present state**
- **Model for all computer systems**
  - Abstract in nature; does not specify protection mechanisms or means of enforcing the model

State Machine Model Characteristics

- No matter what input, output, or processing tasks take place, all states are secure
- If initialized in a secure state, all further states will be secure
- Bell-LaPadula and others were based on this model



State Machine Model Characteristics

State Machine Model

Security Levels | Classifications | Clearances

- **Single-state Machine**
  - Processes data of a single security level
  - No way to separate classifications
  - All users are assumed to have full clearance
- **Multi-state Machine**
  - Processes data at two or more security levels without risk of compromising the system's security
  - Data can be classified
  - Not all users require full clearance

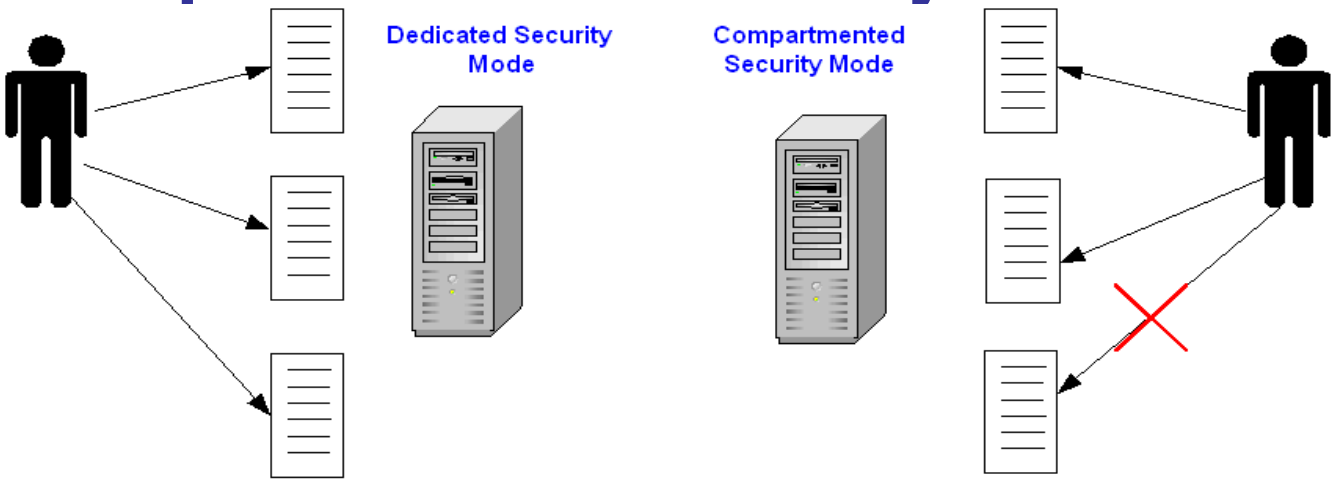
System Security Modes of Operation

Categories| Need-to-Know

Dedicated Security Mode

- Cannot separate compartments or categories
- All users have need-to-know to access all data

Compartmented Security Mode



Operating States and Security Modes

Combinations:

- Dedicated Security Mode on a Single State Machine
- Dedicated Security Mode on a Multi-State Machine
- Compartmented Security Mode on a Single State Machine
- Compartmented Security Mode on a Multi-State Machine

Multilevel Lattice Models

Lattice Characteristics

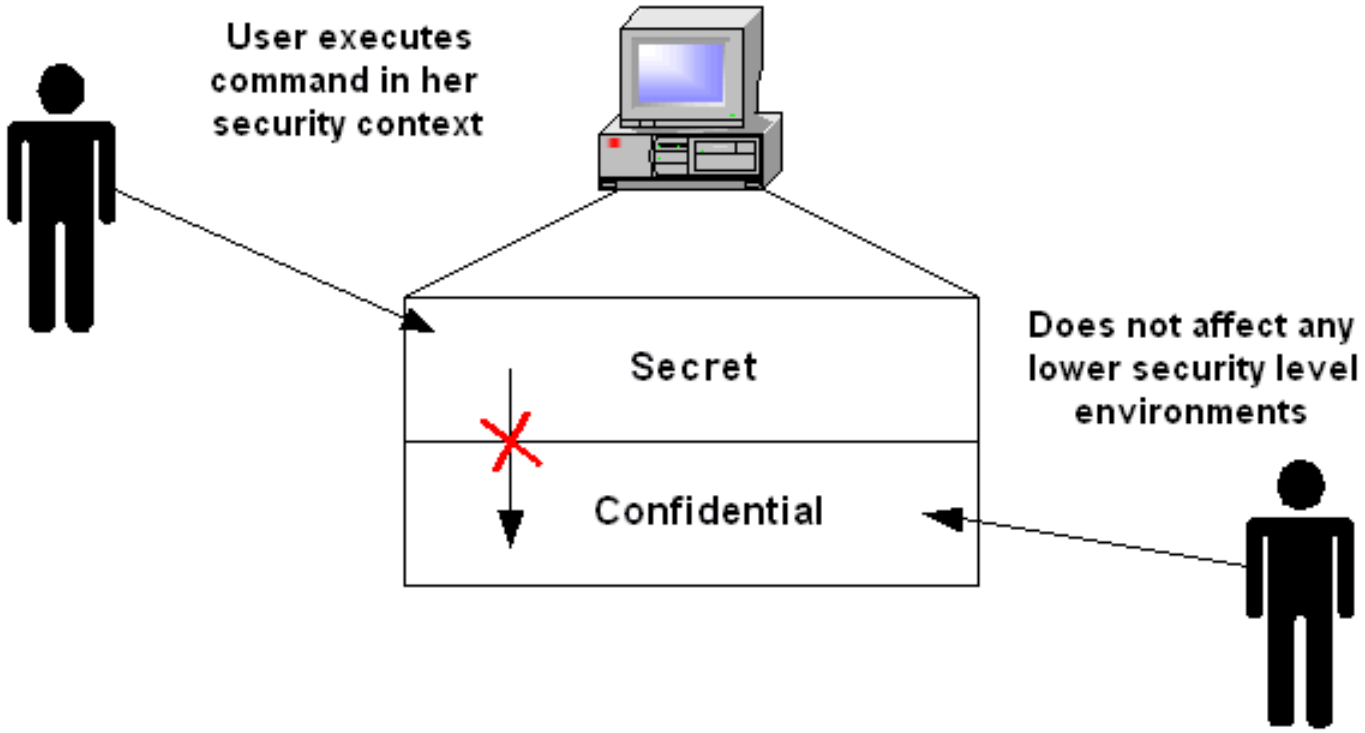
- Every pair of elements (subject and object) has a partially ordered set with a greatest lower bound and least upper bound of access rights

- Bounds can be confidentiality levels (classifications and clearances) or integrity levels

Non-Interference Model

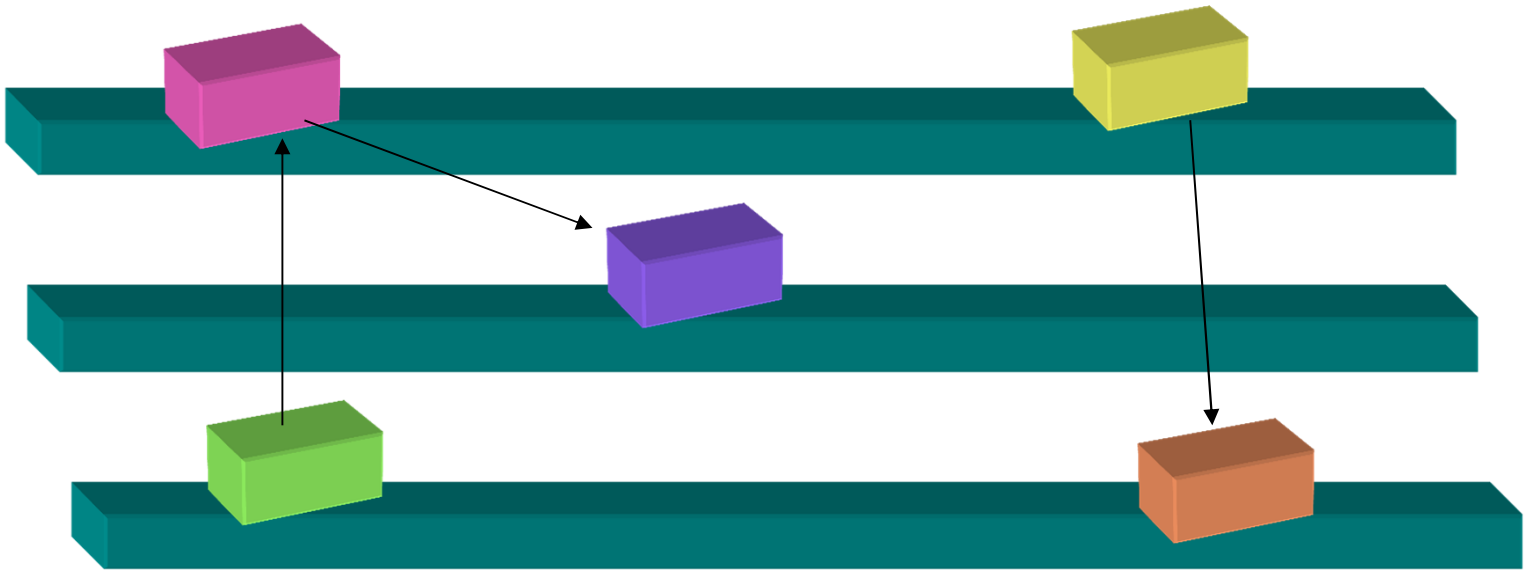
- **Based on a set theory where the users are separated into different domains**
  - Domain: a set of objects that a user can access
- **Uses a State Machine approach that keeps track of which actions are allowed for which users**
- **Users' actions in one domain cannot affect or interfere with users in another domain**
- **A subject cannot be influenced by the behavior of other subjects at higher security levels**

Non-Interference Model



Information Flow Model

- **Looks at the information flows in a State Machine**
  - Each input induces a state transition and a specific output
  - Restricts information from flowing in ways that would go against the security policy



Examples of Security Models

There are several different security models. The following are a few examples that have had a major impact on the way that security has been developed over the years.

1) Bell-LaPadula Confidentiality Model
2) Biba Integrity Model
3) Clark-Wilson Integrity Model
4) Brewer-Nash (Chinese Wall) Model
5) Graham-Denning Model

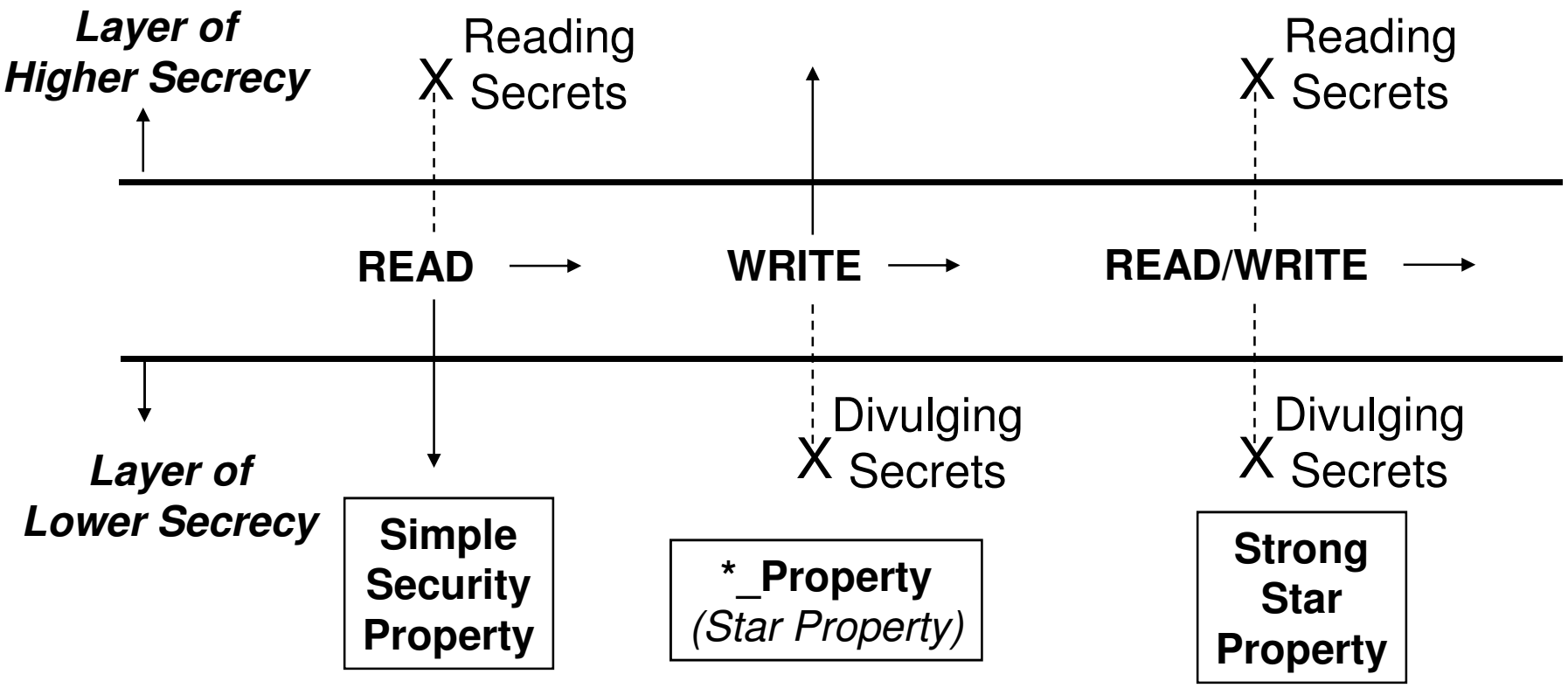
Bell-LaPadula Model

- Formal state transition model that divides entities into subjects and objects
- The model outlines how to keep a secure state in every transaction by only allowing subjects certain access rights
- The clearance of the subject attempting to access an object is compared with that object's classification
- The clearance/classification scheme is expressed in terms of a lattice
  - Upper and lower access rights and bounds

Bell-LaPadula Model

- **Information-flow Security Model**
  - Information should not flow to an object of lesser or non-comparable classification
- **Confidentiality model**
  - Does not secure integrity or availability
- **Developed for military systems to protect national secrets**
  - Multilevel secure databases often use this model

Bell-LaPadula Confidentiality Model



Biba Model

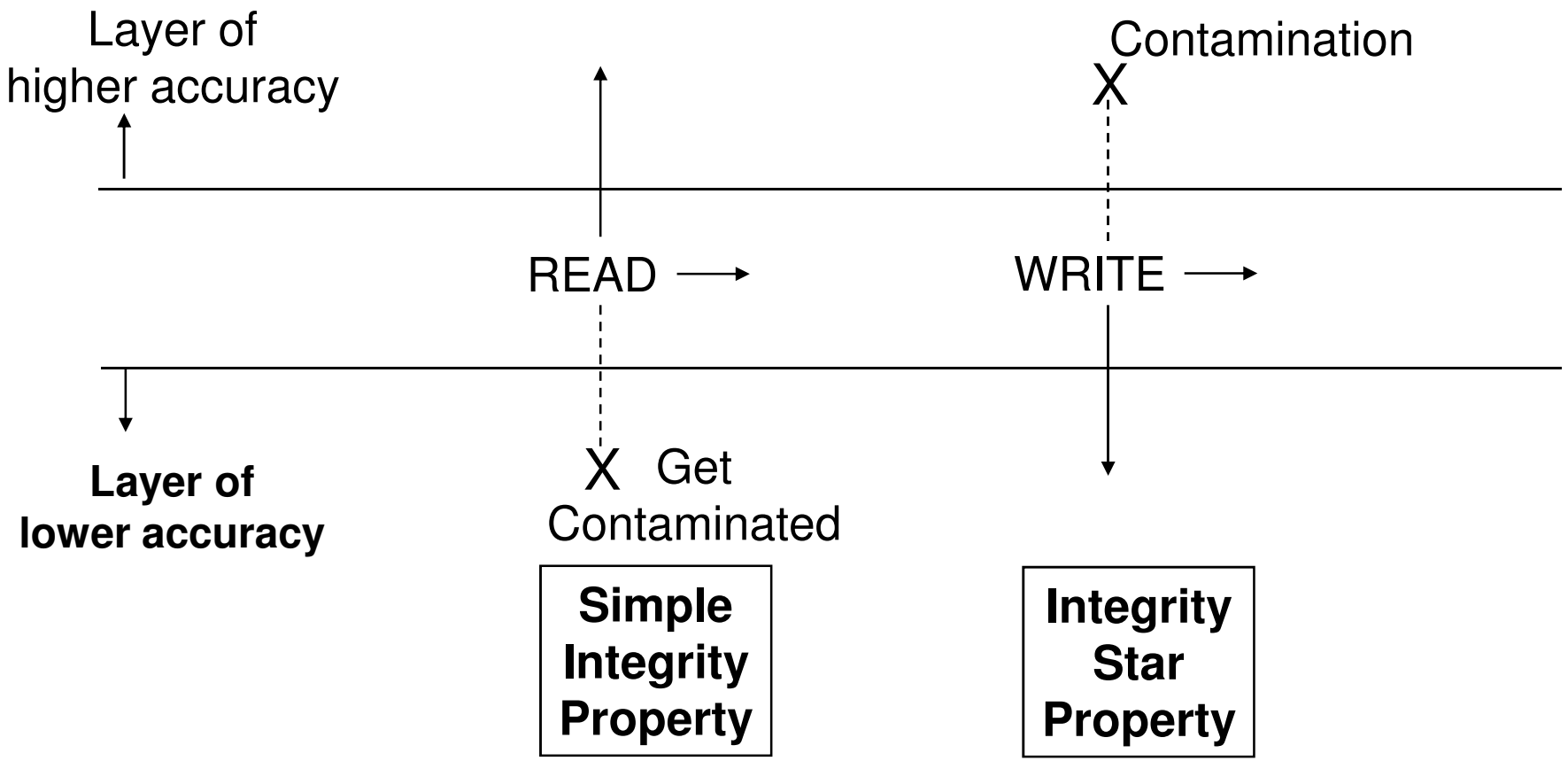
Integrity Model

- No subject can depend on a less trusted object
- Based on a hierarchical lattice of integrity levels

Two Main Rules:

- Subject cannot write data to an object at a higher integrity level
  - "no write up" (integrity axiom)
- Subject cannot read data from an object at a lower integrity level
  - "no read down" (simple integrity axiom)

Biba Integrity Model



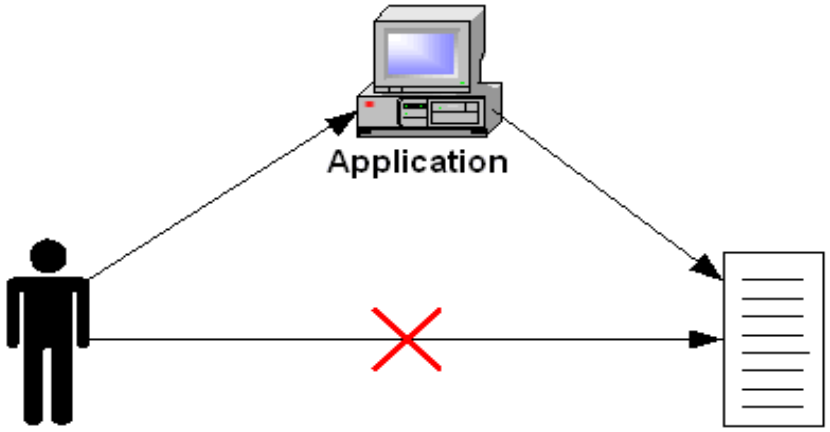
Clark-Wilson Model

- **Requires well-formed transactions and separation of duty**
  - Well-formed transactions: constraints on user to ensure the internal consistency of data is not affected
  - Separation of duties: ensures the consistency of data
    - Prevents users from making improper modification
    - Forces collusion to commit fraud
- **This model partitions objects into programs and data**
  - Biba and Bell-LaPadula use the lattice approach
- **Access triple: subject must go through a program to access and modify data**

Clark-Wilson Model

Deals with all three integrity goals:

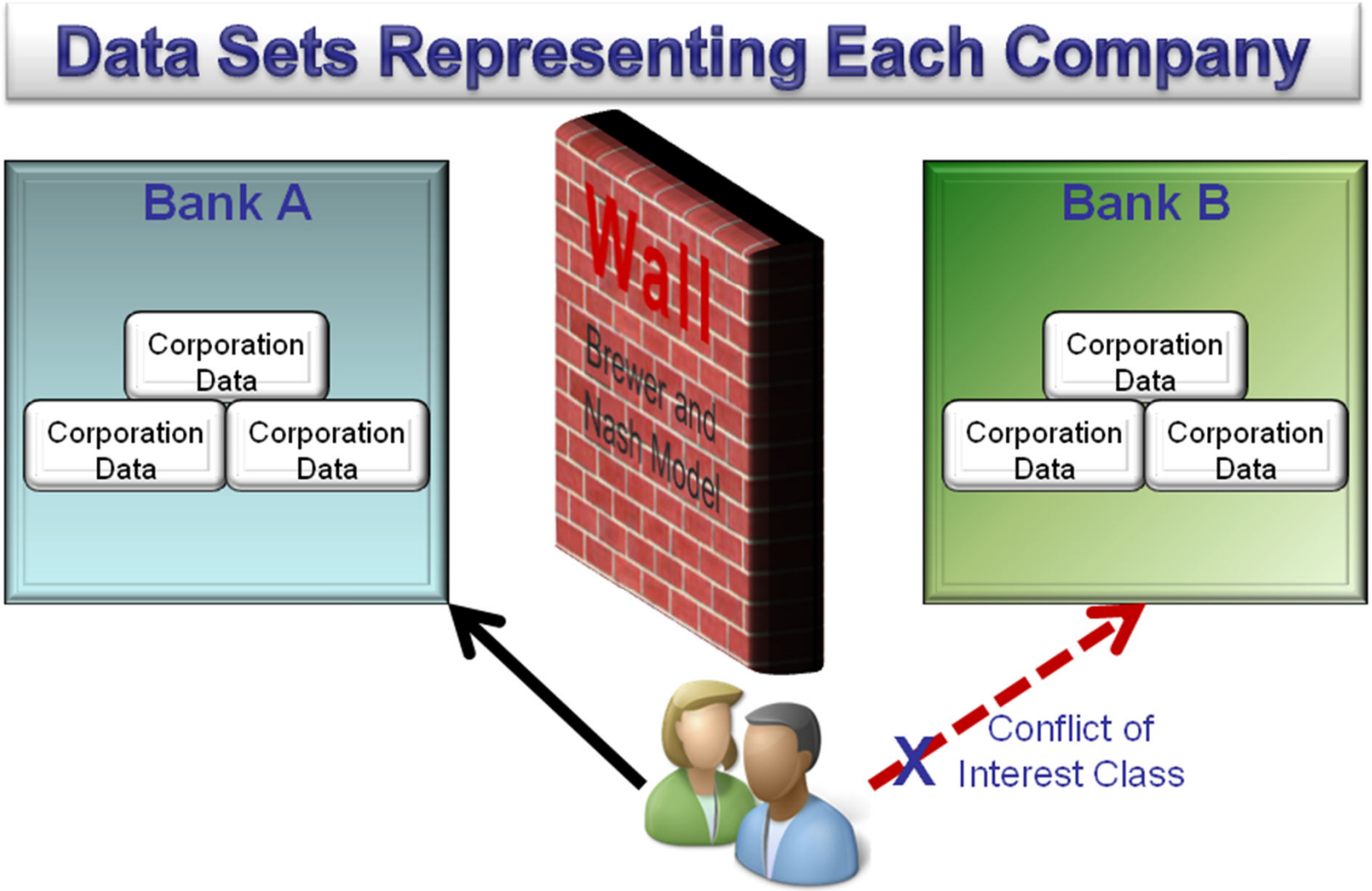
- Prevents unauthorized users from making modifications
- Prevents authorized users from making improper modifications
- Maintains internal and external consistency



Brewer and Nash Model – Chinese Wall

- **Mathematical theory used to implement dynamically changing access permissions**
- **Defines a wall and develops a set of rules that ensure that no subject accesses objects on the other side of the wall**
- **Individuals are only allowed to access data that is not in conflict with data they accessed previously**
  - It allows controls to be put into place to ensure that there is no conflict of interest
  - If a user accesses one company's data, the competitor's data can automatically be deemed "off limits"
- **Way of separating competitors' data within the same integrated database**
- **Tries to ensure that users do not make fraudulent modifications to objects**

Brewer and Nash Model



Graham-Denning Model

- Primarily concerned with how subjects and objects are created, how subjects are assigned rights or privileges and how ownership of objects is managed. This model defines eight primitive protection rights:

1) Create object
2) Create subject
3) Delete object
4) Delete subject
5) Read access right
6) Grant access right
7) Delete access right
8) Transfer access right

Agenda

- Fundamental concepts of security models
- **Components of information systems security evaluation models**
- Security capabilities of information systems
- Vulnerabilities of security architectures
- Software and system vulnerabilities and threats
- Countermeasure principles

Evaluation of the Security of Products

- Helps vendor develop a product to meet the market's demand
- Third party verifying the security mechanisms and acclaimed protection in products
- Provides a common metric to understand and talk about protection provided in products
- A "grading system"

Product Evaluations

- **Degree of independence of the evaluation team is crucial**
- **Evaluation criteria needs to reflect security features**
  - Higher security increases rigor and completeness of testing
- **Accreditation is environment- and system-specific**
- **Balance of risk and benefits:**
  - One product may be appropriately accredited for one environment but not another

Trust vs. Assurance

- **Assurance and trust are evaluated through inspecting development practices and testing**
- **A trusted system**
  - All protection mechanisms work to process sensitive data for many types of users and maintain the same level of protection
- **Assurance**
  - Degree of confidence that the system will act in a correct and predictable manner in each and every computing situation

Security Evaluations

Evaluation Standards:

- Trusted Computer System Evaluation Criteria (TCSEC)
- Information Technology Security Evaluation Criteria (ITSEC)
- ISO/IEC 15408 Common Criteria

TCSEC

- **Developed by the National Computer Security Center (NCSC) for U.S. DOD**
- **Based on the Bell-LaPadula model**
- **Rainbow Series**
  - Orange Book rated operating systems - Standalone
  - Red Book - Trusted Network Interpretation of the TCSEC (TNI)

TCSEC Divisions Breakdown

- **TCSEC addresses confidentiality, but not integrity or availability**
- **Functionality and assurance of the security mechanisms are not evaluated separately, but combined and rated as a whole system (TCB)**
- **Graded classification of systems that is divided into hierarchical divisions of security levels:**
  - A - Verified Protection (Formal Methods)
  - B1, B2, B3 - Mandatory Protection (Security Labels)
  - C1, C2 - Discretionary Security Protection
  - D - Minimal Protection

TCSEC Levels

- **C1 – Discretionary Security Protection**
  - Separation of Users and Data
  - Cooperating users processing data at the same sensitivity
- **C2 – Controlled Access Protection**
  - Object reuse
  - Protect audit trail
    - Considered the minimum level operating system for a firewall
- **B1 – Labeled Security Protection**
  - Labels and mandatory access control
  - Process isolation in system architecture
  - Design specifications and verification
  - Device labels

TCSEC Levels

- **B2 – Structured Protection**
  - Device labels and subject sensitivity labels
  - Trusted path
  - Separation of operator and administrator functions
  - Covert channel analysis
- **B3 – Security Domains**
  - Security administrator role defined
  - Trusted recovery
  - Monitor events and notify security personnel
- **A1 – Verified Design**
  - Formal methods

ITSEC

- **Developed in Europe to establish one standard for evaluating the security of systems**
- **Evaluates functionality and assurance separately**
  - Two systems can provide the same functionality but different assurance levels.
- **F1-F10 rates for functionality**
- **E0-E6 rates for assurance**
  - Assurance is a measurement of correctness and a judgment of its effectiveness of security functionality.
  - Assurance provides confidence.

Common Criteria

- **ISO/IEC 15408**
- **Rainbow series was too rigid and did not take many things into account**
- **ITSEC provided more flexibility, but added more complexity with its attempts**
- **Made up from:**
  - TCSEC
  - ITSEC
  - Canadian Trusted Computer Product Evaluation Criteria (CTCPEC)
  - Federal Criteria

Common Criteria Components

- **Protection Profile**
  - Description of needed security solution
- **Target of Evaluation**
  - Product proposed to provide needed security solution
- **Security Target**
  - Written by vendor explaining security functionality and assurance mechanisms that meet the needed security solution
- **Packages – Evaluation Assurance Levels (EAL)**
  - Security requirements are bundled into packages for re-use
  - Describes what must be met to achieve specific EAL ratings

First Set of Requirements

Security Functional Requirements

- Identification and authentication
- Audit
- Resource utilization
- Trusted paths/channels
- User data protection
- Security management
- TOE access
- Communications
- Privacy
- Protection of the TOE security functions
- Cryptographic support

Second Set of Requirements

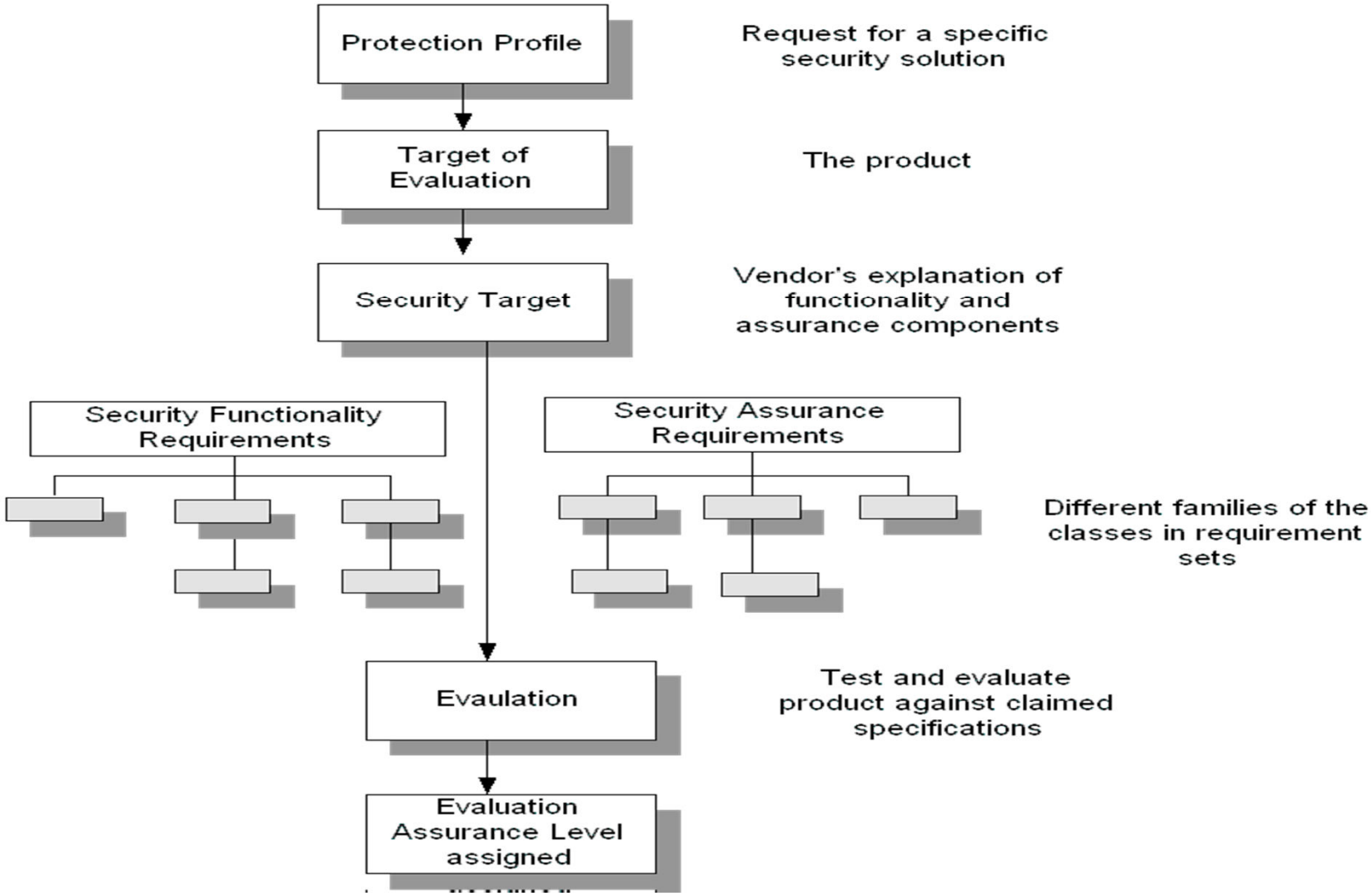
Security Assurance Requirements

- Guidance documents and manuals
- Configuration management
- Vulnerability assessment
- Delivery and operation
- Life cycle support
- Assurance maintenance
- Development
- Testing

Evaluation Ratings

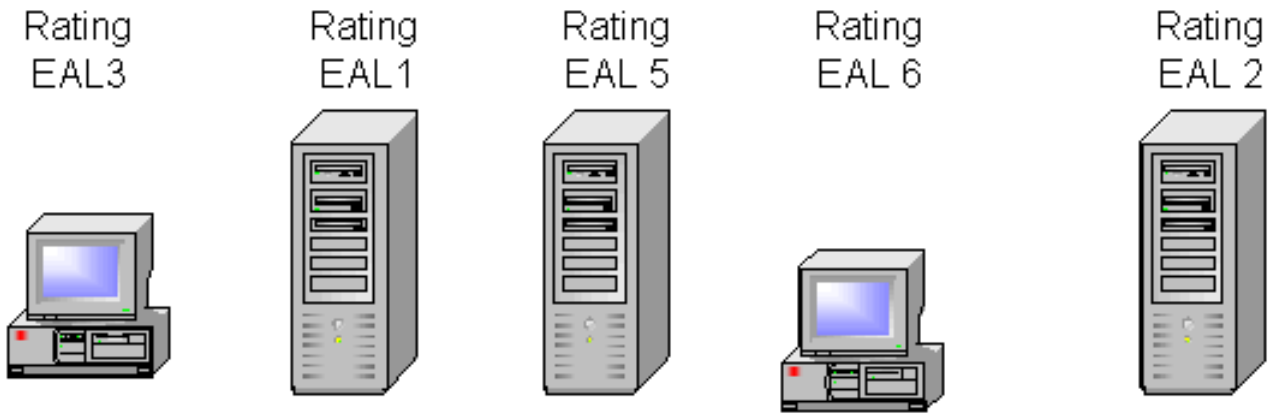
- EAL 1 – Functionally tested
- EAL 2 – Structurally tested
- EAL 3 – Methodically tested and checked
- EAL 4 – Methodically designed, tested, and reviewed
- EAL 5 – Semi-formally designed and tested
- EAL 6 – Semi-formally verified, designed, and tested
- EAL 7 – Formally verified, designed, and tested

Common Criteria Outline



CC – Evaluation Components

- Uses protection profiles
- Product is assigned an Evaluation Assurance Level (EAL)
- Product is put on Evaluated Products List (EPL)
- Provides more flexibility than previous evaluation criterion



Certification and Accreditation

- **Relationship of C&A to Enterprise Architecture**
  - Technical Reference Model
  - Standards Profile
- **Communication and Validation of Security Requirements**

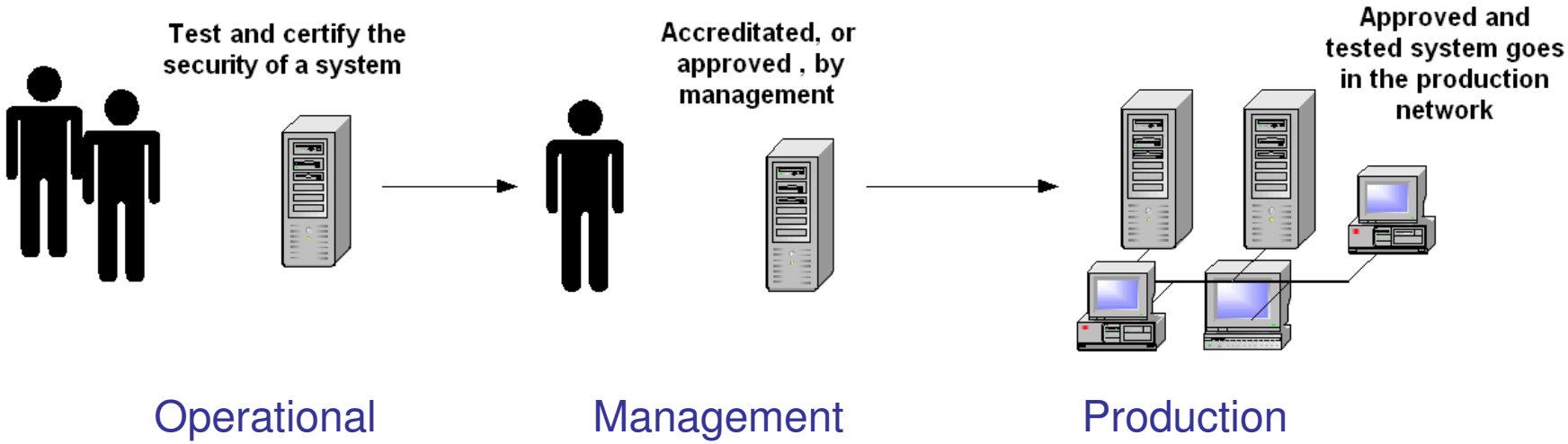
- **Consistent Taxonomy**
  - Rules of Behavior
  - System Controls
  - Risk Management

Certification vs. Accreditation

Security rating is just one step:

- Certification is the technical evaluation of the security components within a product… *in my environment.*

- Accreditation is the formal acceptance of the product's overall security by management.



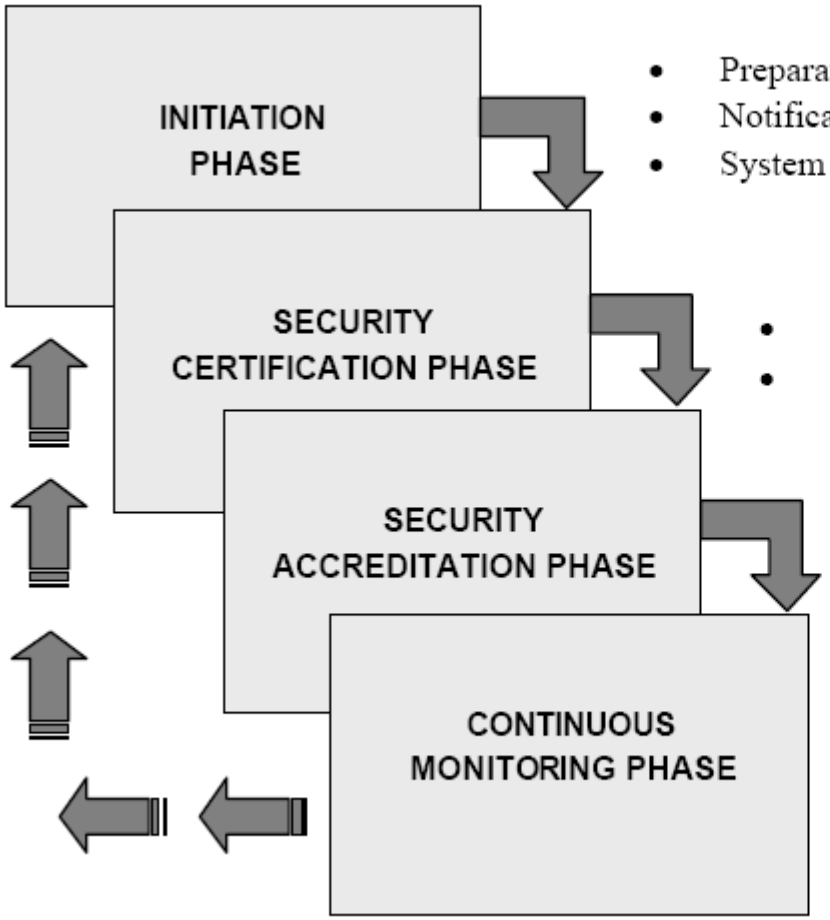
Certification Characteristics

- Formal process for testing systems against a set of security requirements
- Normally performed by an independent reviewer instead of someone who was involved with building the system
- Less formal security testing can be performed for lower-risk systems

Accreditation Characteristics

- **The decision given by a *senior* agency official to authorize operation of an information system:**
  - In a particular security mode
  - Using a prescribed set of controls
  - Against a defined threat
  - At an acceptable level of risk
  - For a specific period of time
- **The official explicitly accepts the risk to agency assets based on the implementation of these security conditions.**

Certification and Accreditation Phases



- Preparation
- Notification and Resource Identification
- System Security Plan Analysis, Update, and Acceptance

- Security Control Assessment
- Security Certification Documentation

- Security Accreditation Decision
- Security Accreditation Documentation

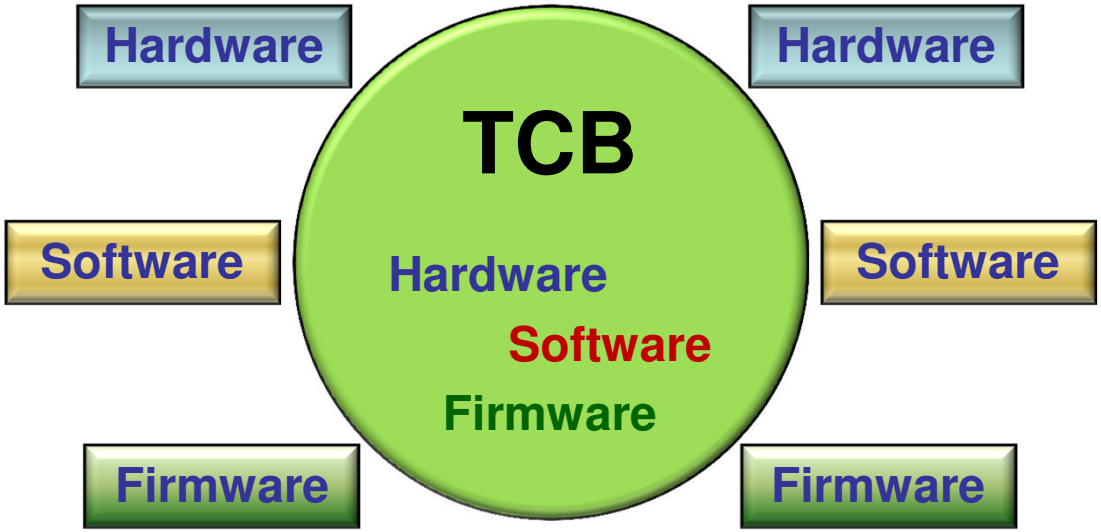
- Configuration Management and Control
- Security Control Monitoring
- Status Reporting and Documentation

Agenda

- Fundamental concepts of security models
- Components of information systems security evaluation models
- **Security capabilities of information systems**
- Vulnerabilities of security architectures
- Software and system vulnerabilities and threats

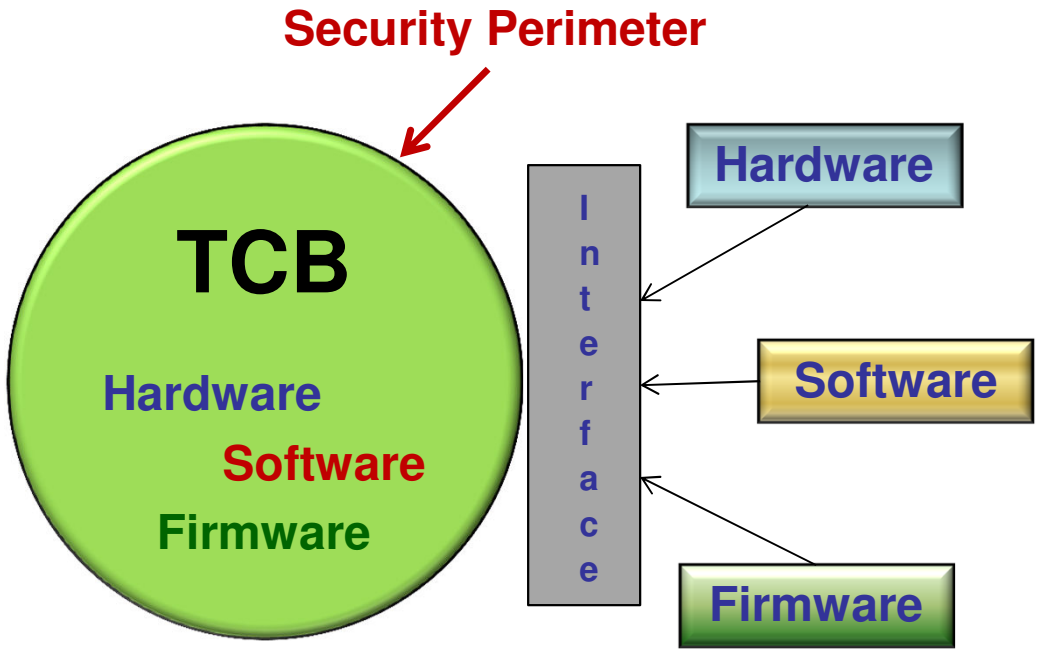
Trusted Computing Base

- Defined as the total combination of protection mechanisms within a computer system
- Elements of the TCB enforce the security policy of the system and do not violate it



Security Perimeter

- The security perimeter separates the TCB and non-TCB objects



Computer Components

Parts of a Computer System

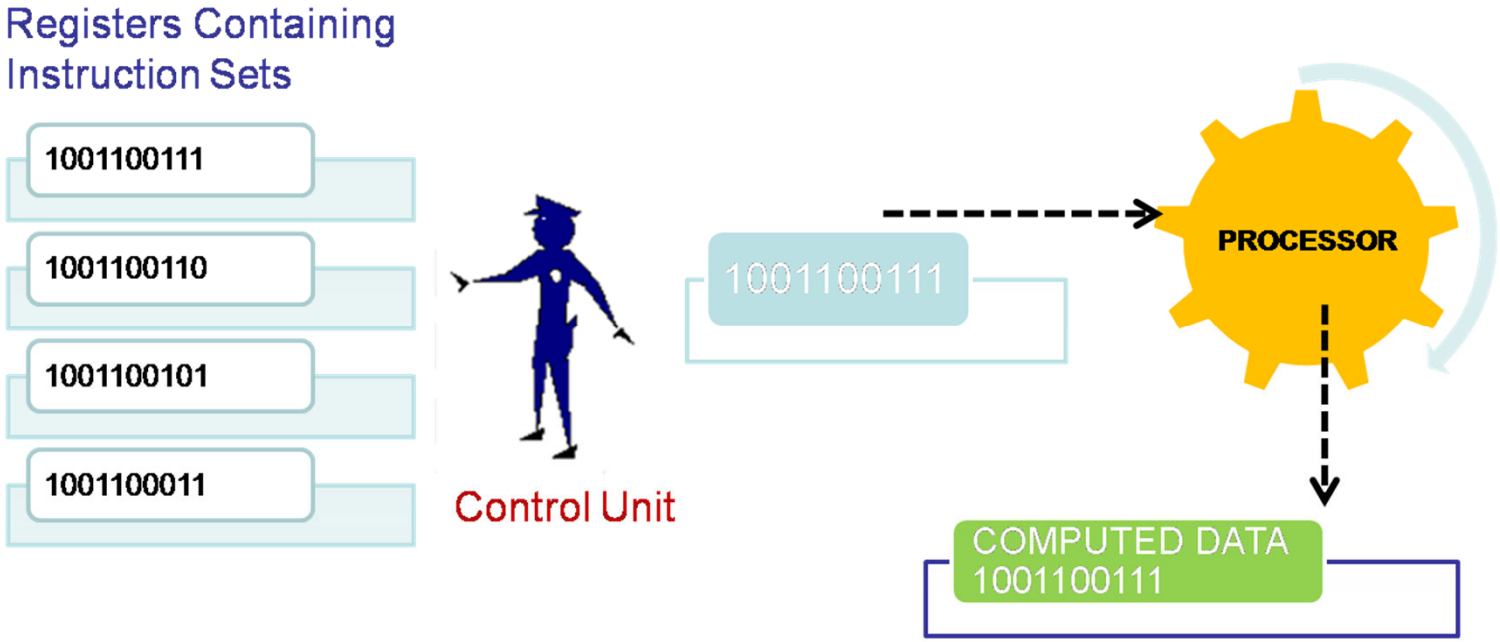
- Processor
- Memory
- Storage
- Buses
- Networking Components
- Operating System

Functionality and security of the system depends upon the interaction of these components

Central Processing Unit (CPU)

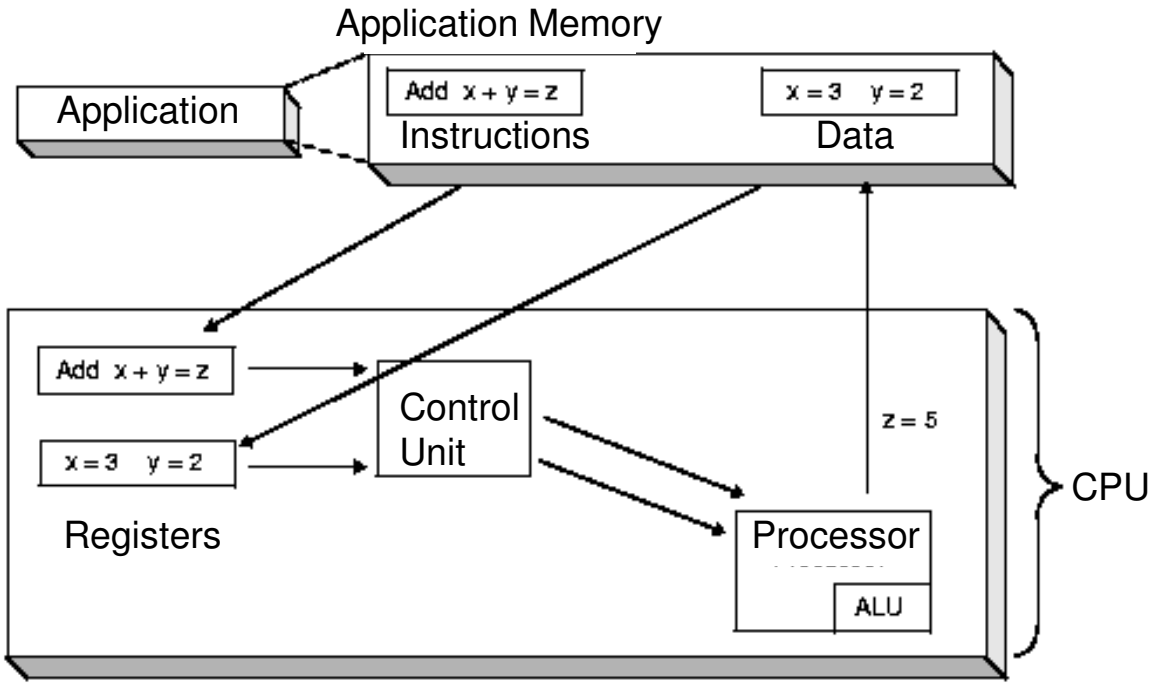
- **Control Unit**
  - Manages synchronization of data being processed

- **Arithmetic Logic Unit (ALU)**
  - Performs mathematical and logical functions on data



Processing Data

- Instructions and data are passed to registers for processing
- After the CPU performs functions on them, they are returned to the original application's memory



CPU States

Supervisor State

- Kernel / Protected / Privileged mode
- Ring Zero
- Program can access entire system
- Both privileged and non-privileged instructions

Problem State

- User / Program mode
- Ring Three
- Only non-privileged instructions are executed
- Intended for application programs

Storage Types

Primary Storage

- Memory directly addressable by the CPU used to store an application's data for processing
- Distinguished from virtual memory

Secondary Storage

- Slower access, non-volatile
- Hard disk, tape, ZIP, optical media drives, thumb drives

More Storage Types

Volatile Memory

- Random access memory (RAM)
- Loses data when system is powered off

Non-volatile Storage

- Does not lose data when system is powered off
- Read-only memory (ROM), erasable and programmable read-only memory (EPROM), Flash, etc.

More Storage Types

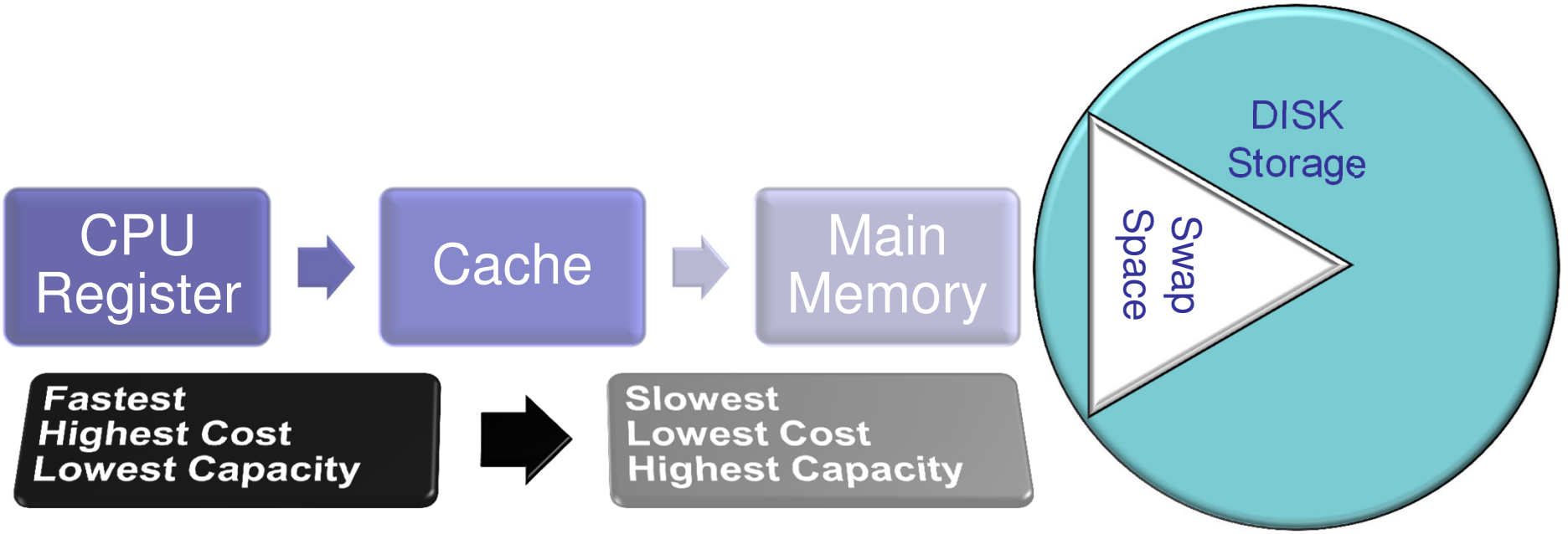
Cache Storage

- Part of RAM used for high-speed writing and reading activities
- Cache logic attempts to predict what instructions will be most needed and stores those for overall increase in system performance

Virtual Storage

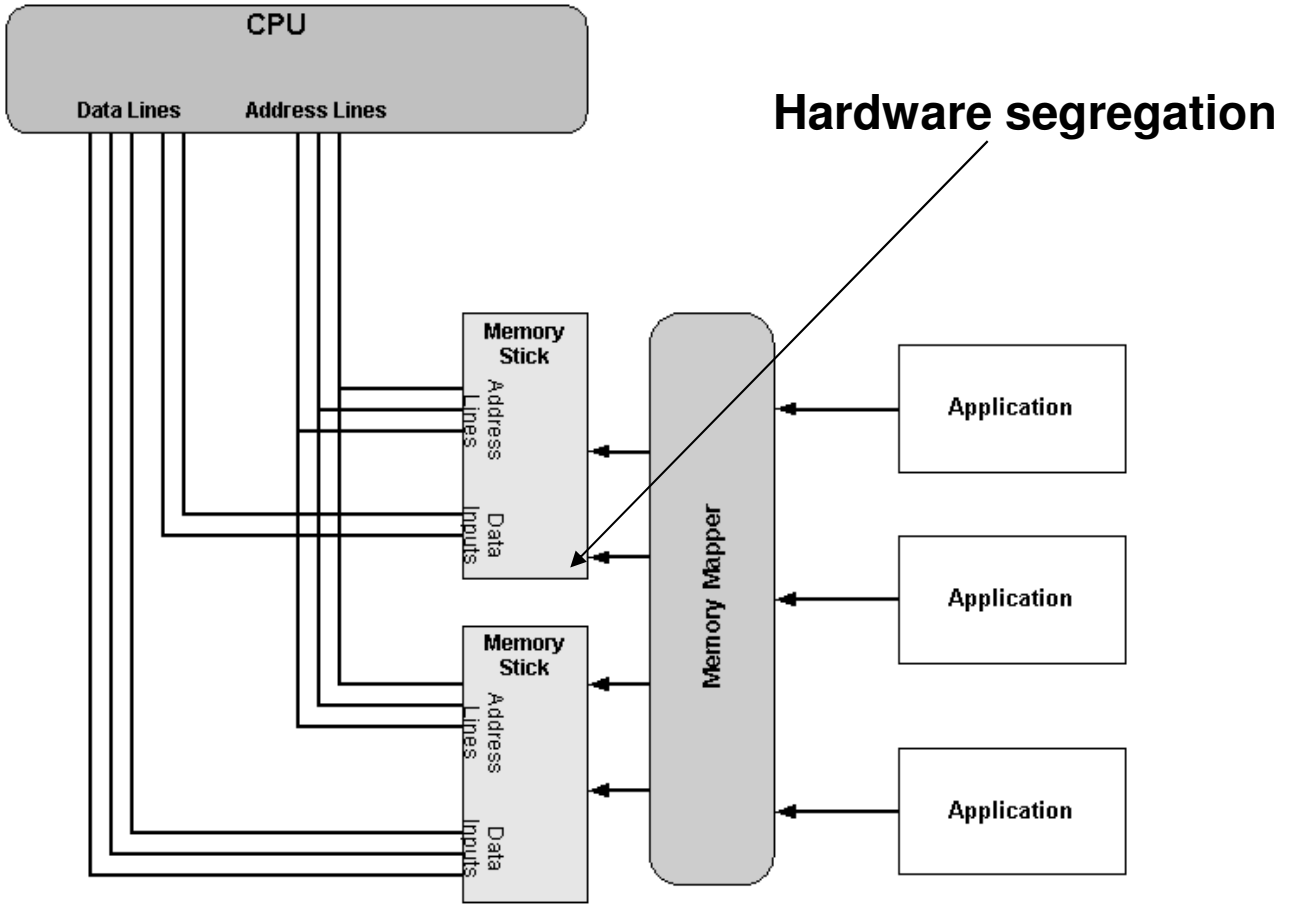
- Secondary storage plus RAM, extends system's overall memory
- Applications access through virtual addresses

Types of Memory

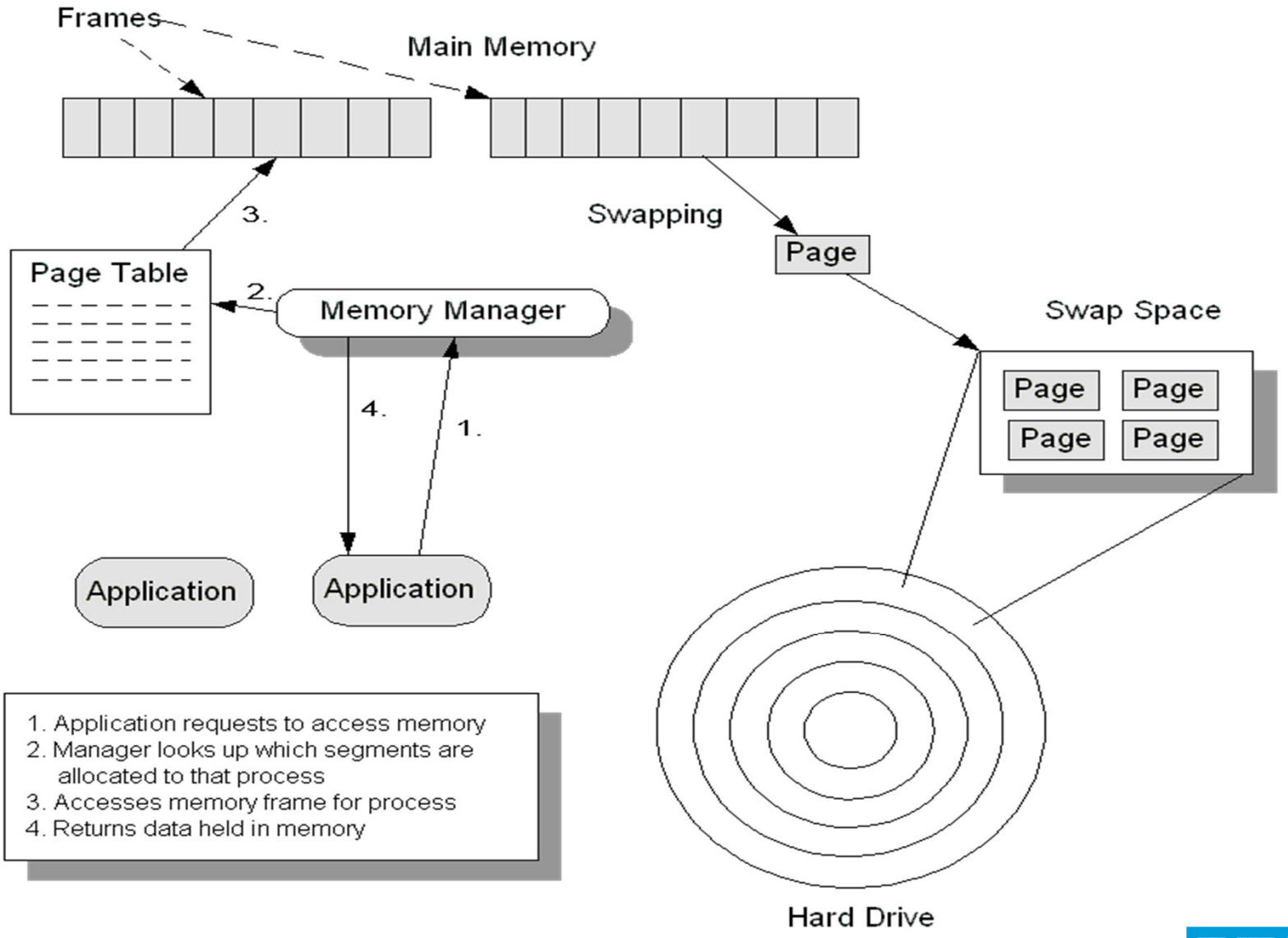


Memory Mapping

- Only the trusted processes can access RAM directly; everything else must be mapped



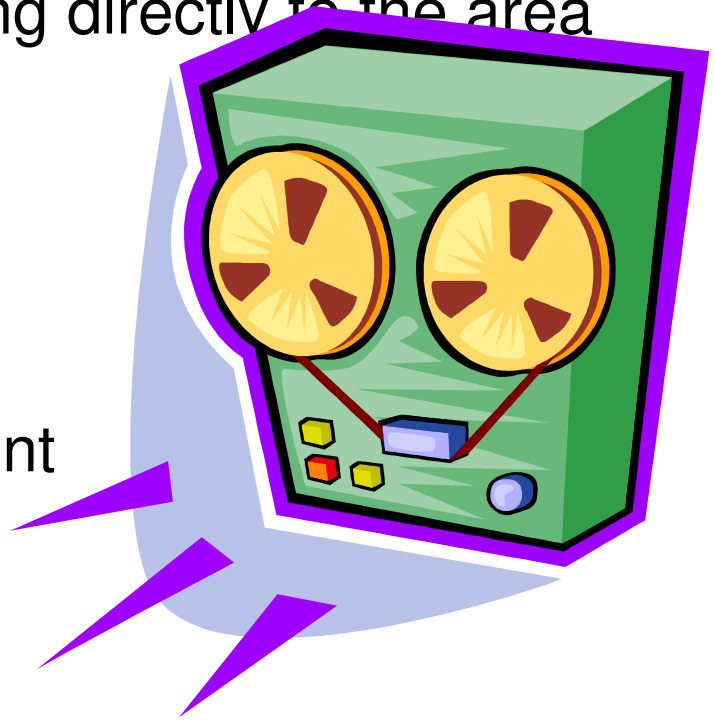
Virtual Storage



Accessing Files

Sequential Storage

- Media that holds data that must be searched from beginning to end instead of going directly to the area where requested data is held
  - Magnetic tape



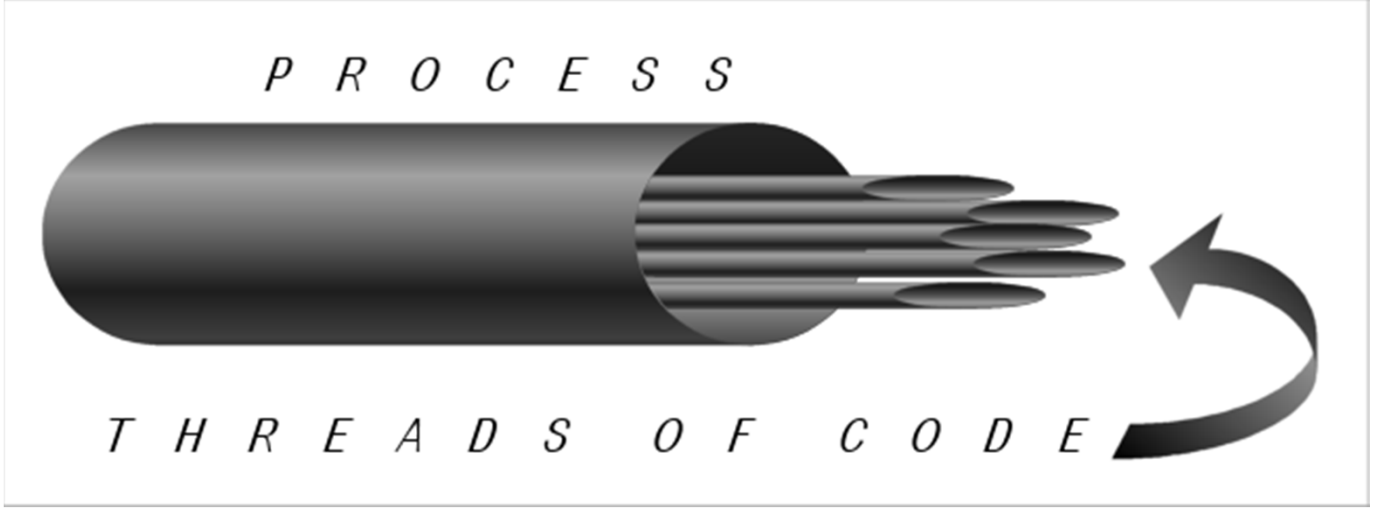
Direct or Random Access

- Files can be accessed in different sequences
  - Dividing media into tracks and sectors
  - Accessing memory

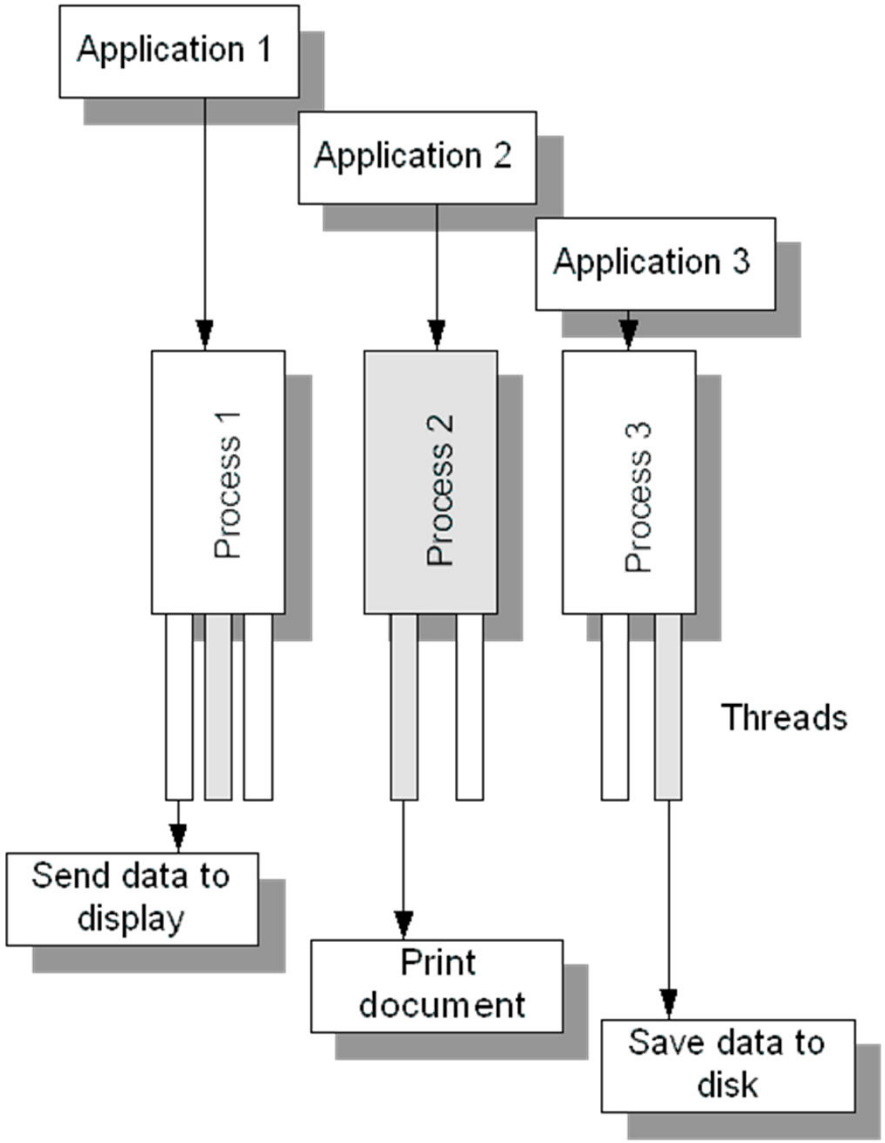
Process vs. Thread

Process

- The context for a program
- Has its own virtual memory space
- A process can contain several threads of code
- Thread is the basic entity that can be scheduled
- Each thread has a kernel and user mode stack

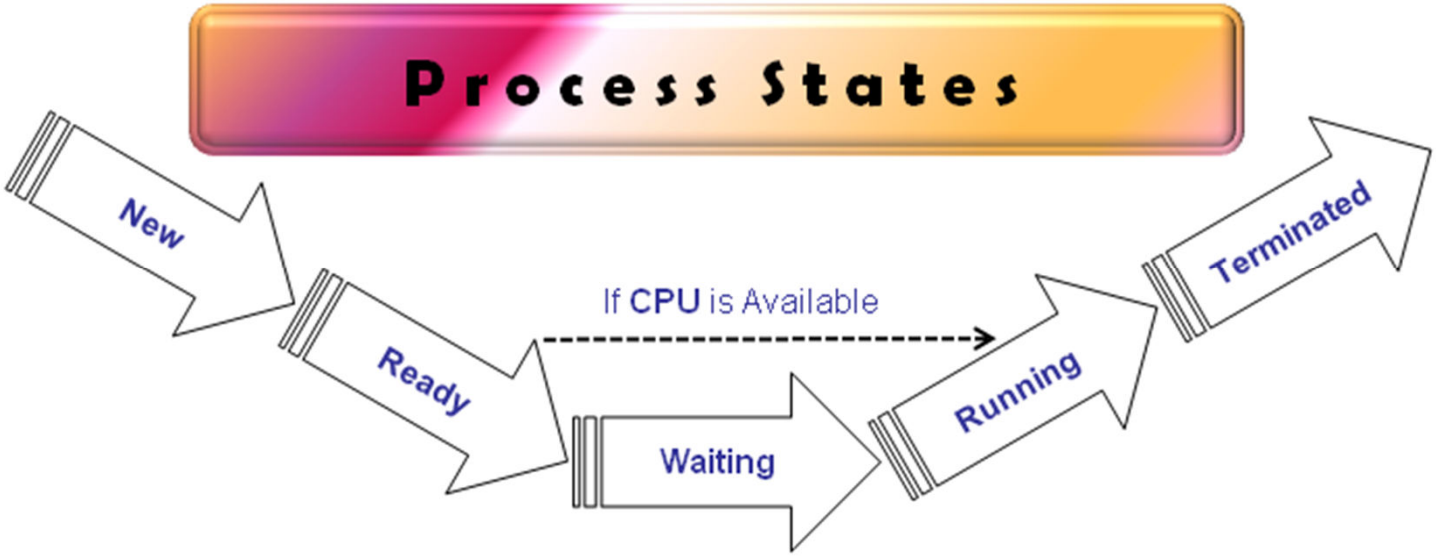


Processes and Threads



Process States

- Stopped – *process is not running*
- Ready – *available for use and waiting for instructions*
- Waiting – *process is waiting for an interrupt to be able to be processed by the CPU*
- Running – *process instructions are being executed by the CPU*

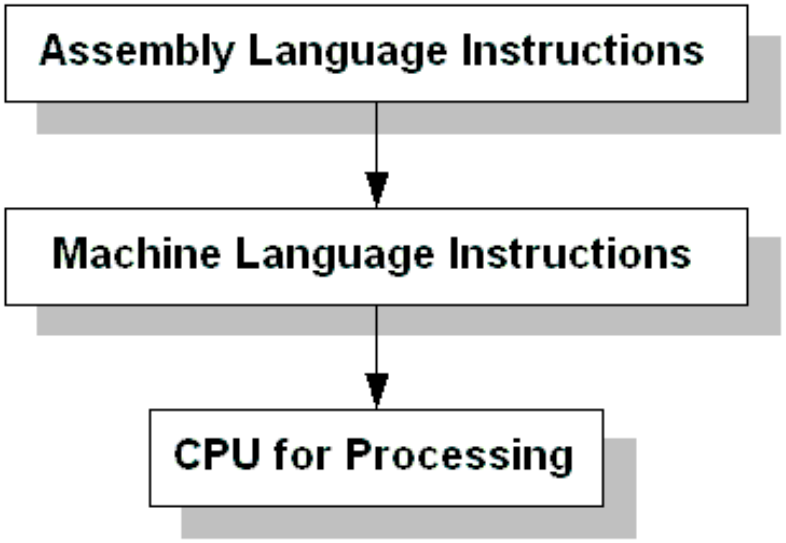


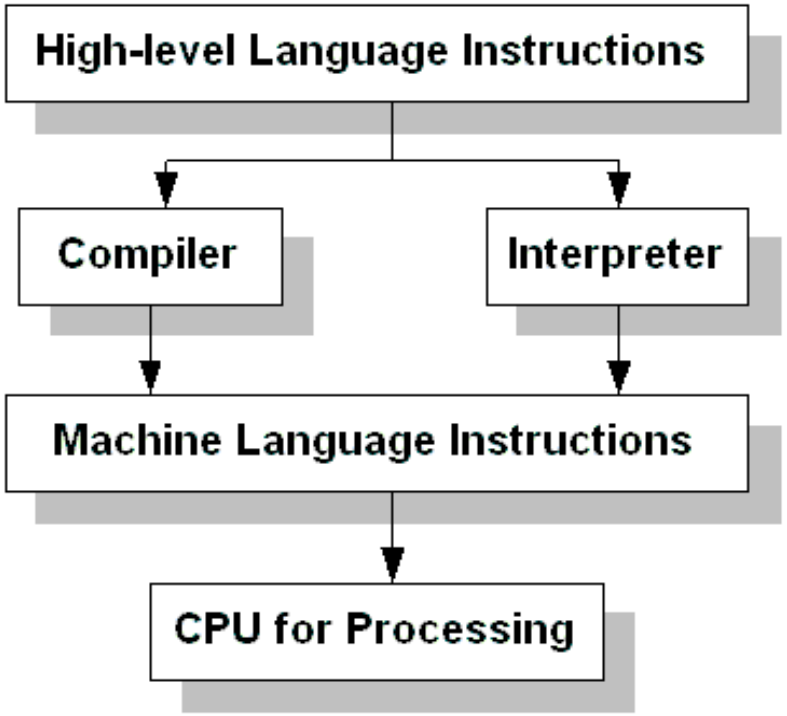
System Functionality

- **Multi-threading**
  - The operating system can process several pieces of code at one time
  - Concurrent processing of several tasks inside same program
- **Multi-tasking**
  - Simultaneous execution of two or more programs
- **Multi-programming**
  - Interleaved execution of two or more programs by a CPU
  - Running more than one application, but not necessarily multithreading
- **Multi-processing**
  - More than one CPU and they can process requests in parallel
- **Multi-core**
  - A processer composed of two or more independent cores

Processing Instructions

- The type of programming language used determines the necessary steps of decoding and encoding before the instructions reach





System Self-Protection

An operating system uses several mechanisms to protect itself:

- **Memory segmentation**
  - Process isolation
  - Security domains
  - Virtual machines
- **Layering and data hiding**
  - Levels of access to resources
  - Protection rings
- **Protection techniques are performed by elements of the Trusted Computing Base**

Security Definitions

- **Subject**
  - Active entity that requests access to an object or the data within an object
  - Examples: User, process, machine instruction
- **Object**
  - Passive entity that contains information
  - Examples: file, record, memory location

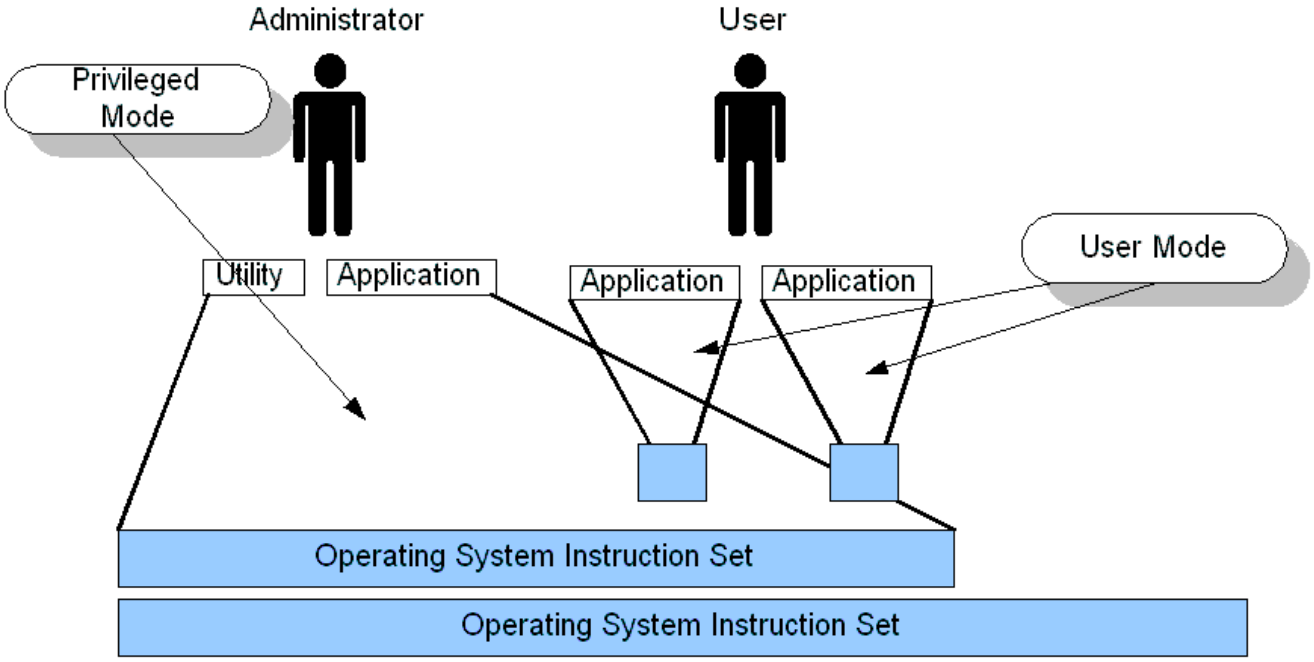
- **Access**
  - Ability of subject to perform a task
  - Flow of information between a subject and an object

Resource Access

Level of Privilege

- Subjects of higher trust can access more system instructions and operate in privileged mode

- Controls must be placed on utilities that require Ring



Process Isolation – *What and How?*

- **Preserves object's integrity and subject's adherence to access controls**
- **Prevents objects from interacting with each other and their resources**
- **Actions of one object should not affect the state of other objects**
- **Techniques used to enforce isolation:**
  - **V**irtual mapping
  - **E**ncapsulation of objects
  - **N**aming distinctions
  - **T**ime multiplexing of shared resources

System Protection Mechanisms

Layering

- Code operating at one layer can only communicate with other layers through interfaces

Data Hiding

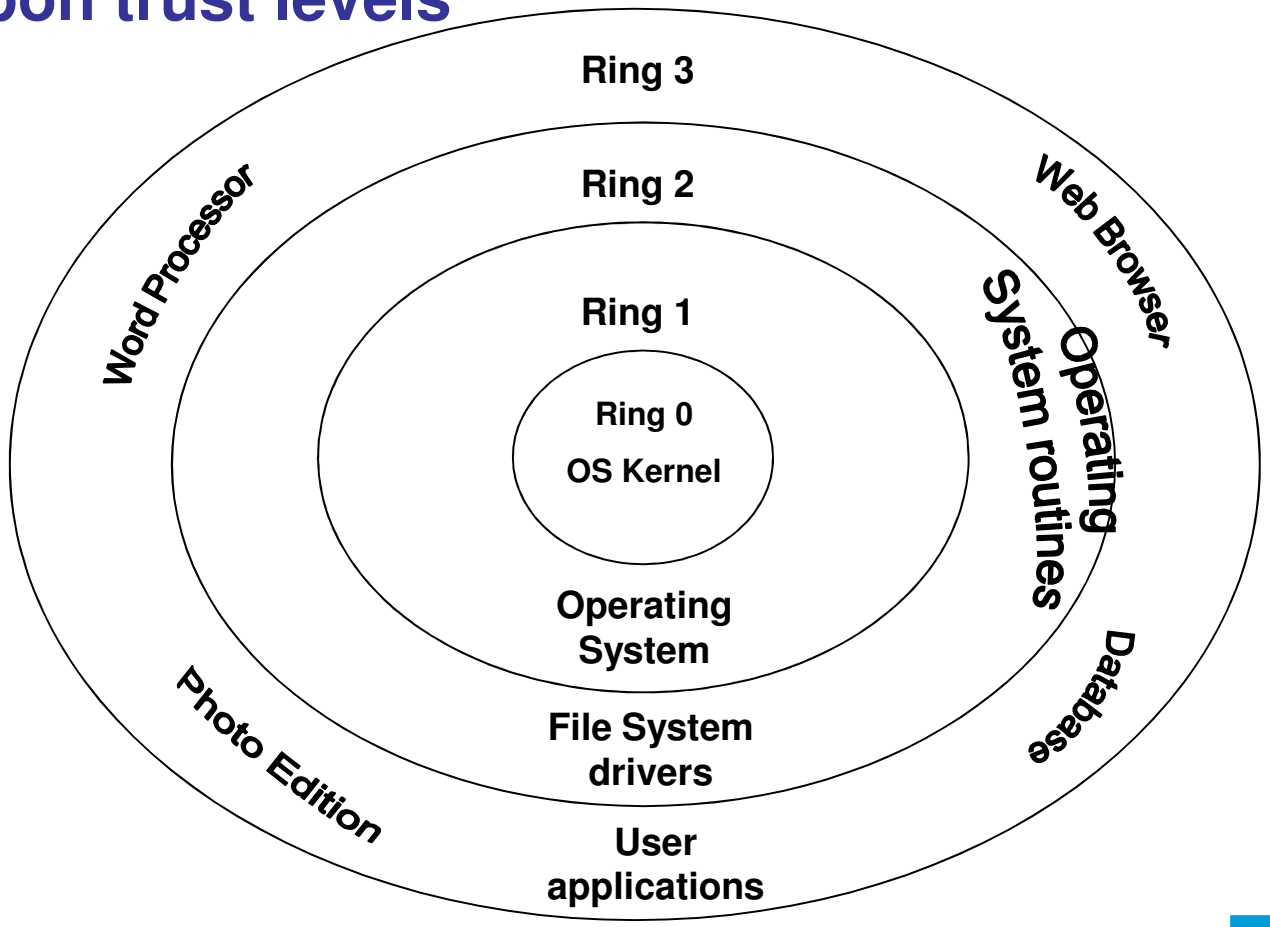
- Data in one layer cannot be accessed by code in a different layer

Abstraction

- Regarding only required information; the big picture

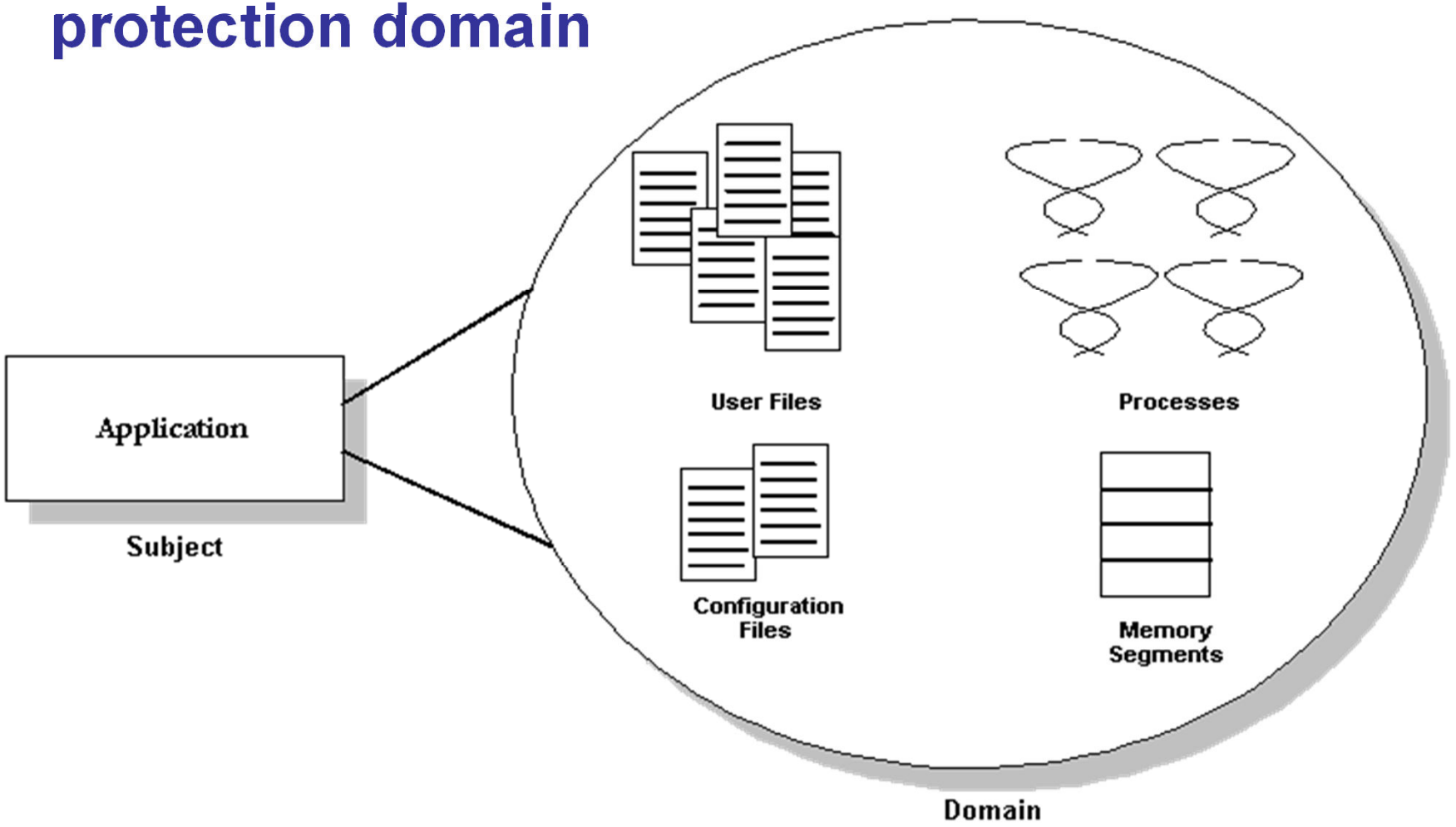
Protection Rings

- An operating system protects itself by implementing protection rings and placing components in each ring based upon trust levels



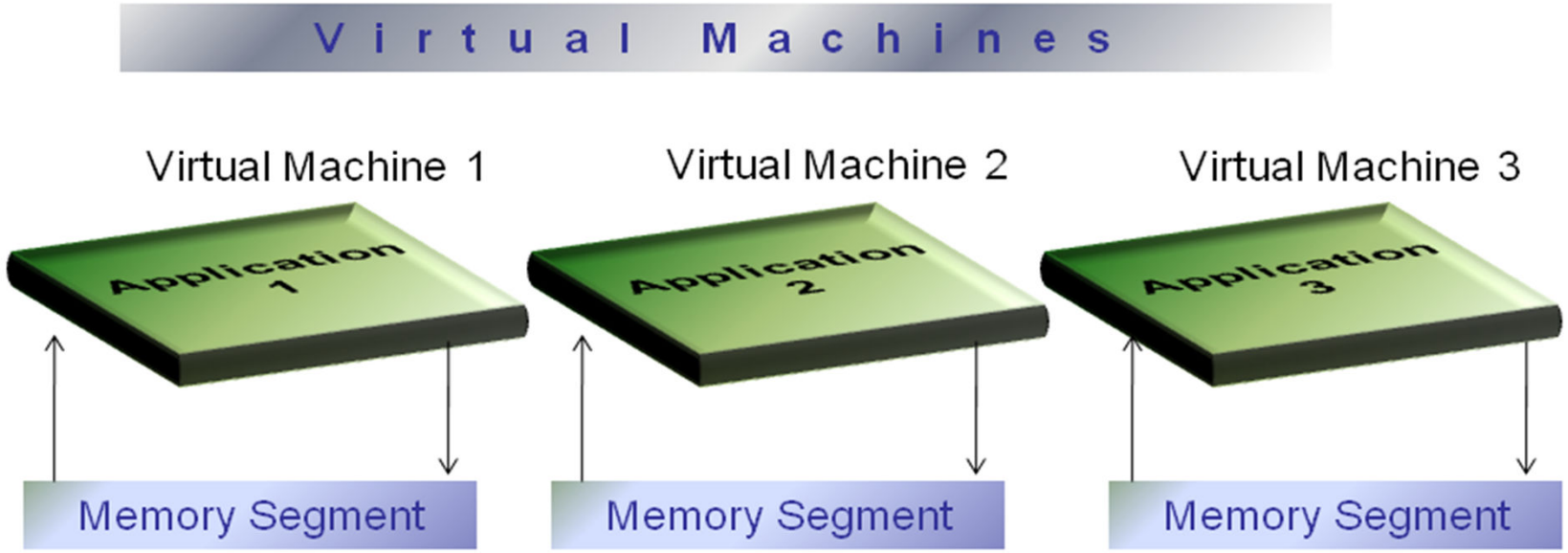
Security Domain of a Process

- A security domain can be called an execution or protection domain



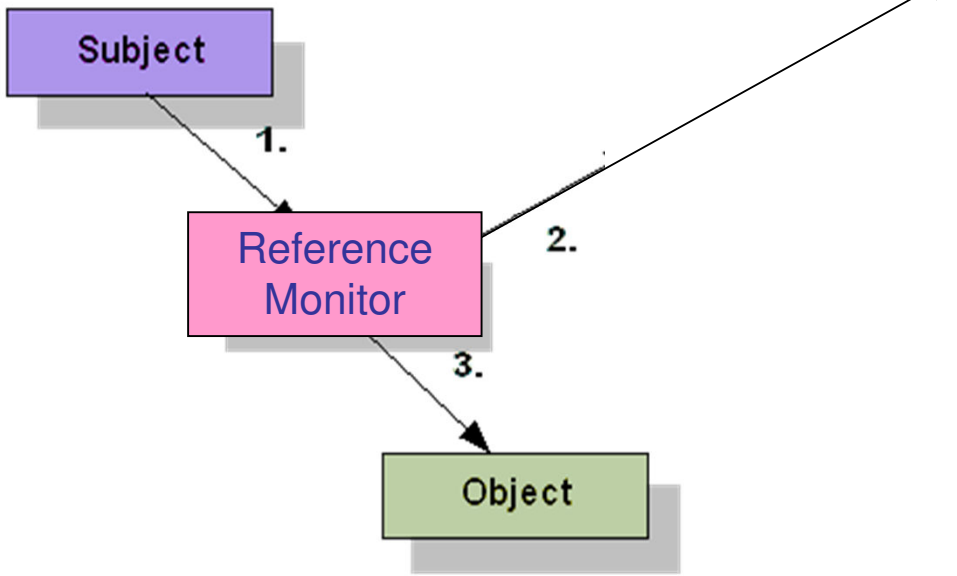
Virtual Machines

- An operating system provides an application with a working environment.
- Virtual machines mimic the architecture of the actual system.
- On multilevel systems, they can run at different security levels.



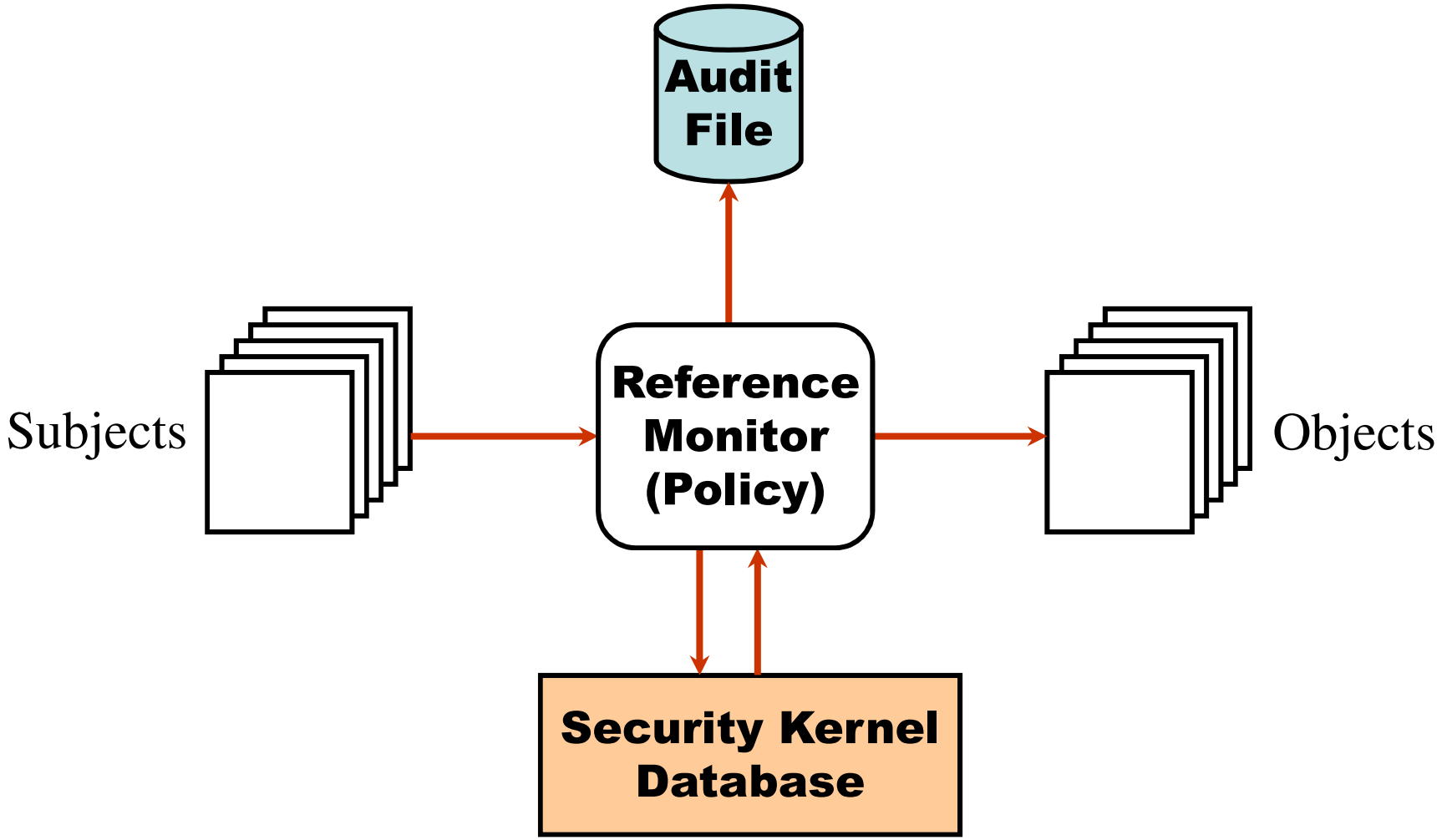
Reference Monitor Concept

- **Reference Monitor –** Abstract machine that controls the access subjects have to objects
- **Security Kernel –** Components in system that enforce the rules of the reference monitor



John	Full Control
Bob	No Access
David	Read
Marge	Full Control
George	Write
Sean	No Access

Reference Monitor



Requirements

- **Three main requirements of the Security Kernel and Reference Monitor:**
  - The security kernel must provide isolation for the processes carrying out the reference monitor concept, and it must be tamperproof.
  - The reference monitor must be invoked for every access attempt and must be impossible to circumvent.
  - The reference monitor must be small enough to be tested and verified in a complete and comprehensive manner.

Agenda

- Fundamental concepts of security models
- Components of information systems security evaluation models
- Security capabilities of information systems
- **Vulnerabilities of security architectures**
- Software and system vulnerabilities and threats
- Countermeasure principles

Threats to Security Architectures

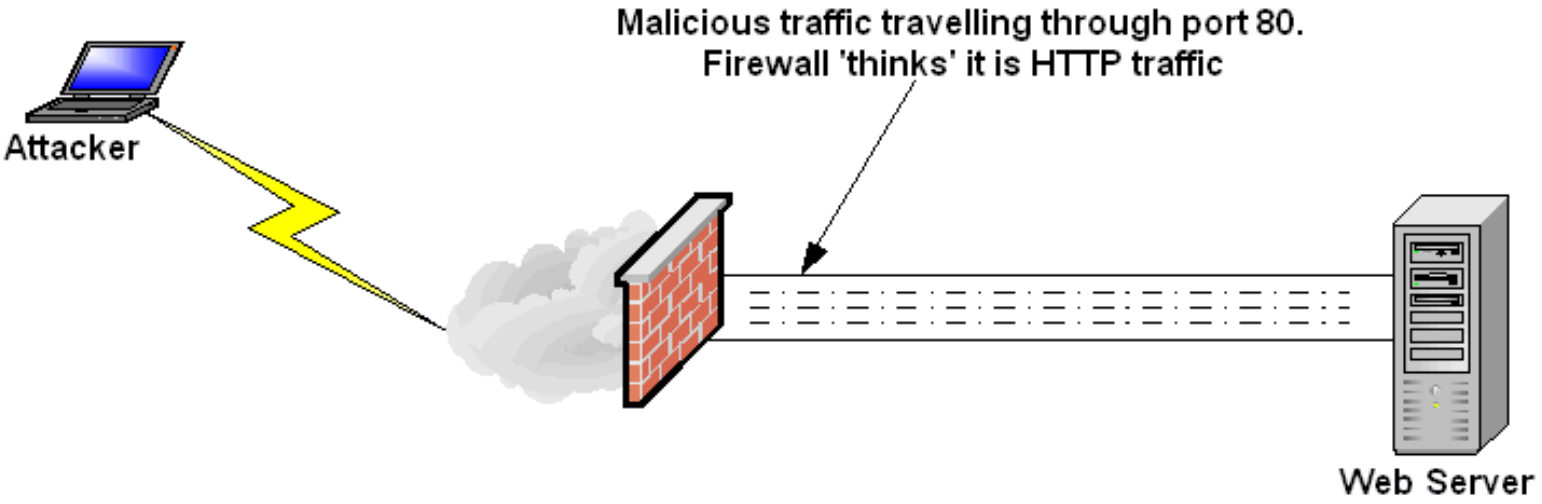
- Covert Channels
- Emanations

Covert Channels

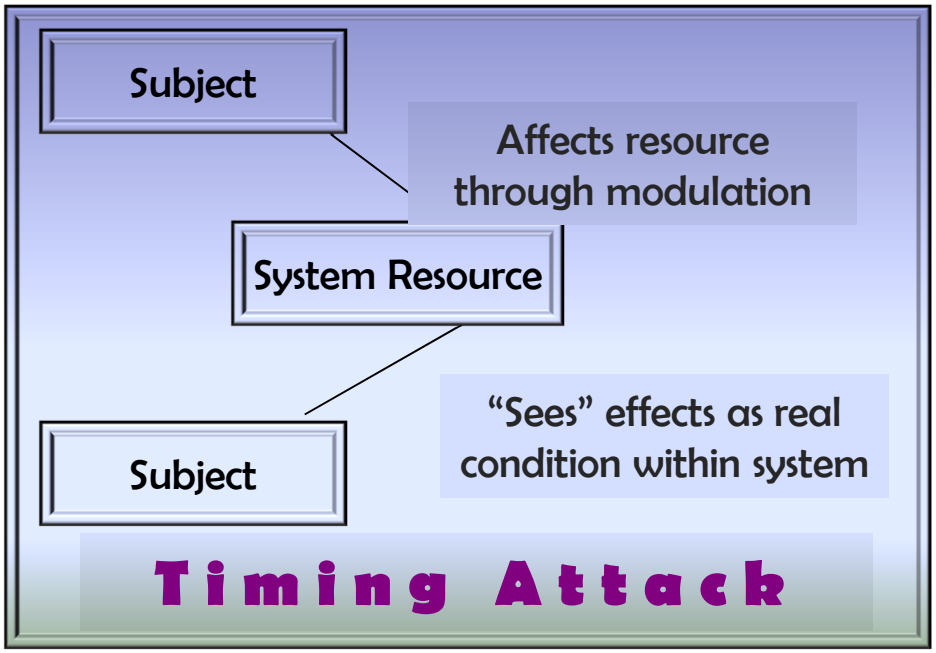
- **Sending information in an unauthorized manner using a medium in an unintended way**

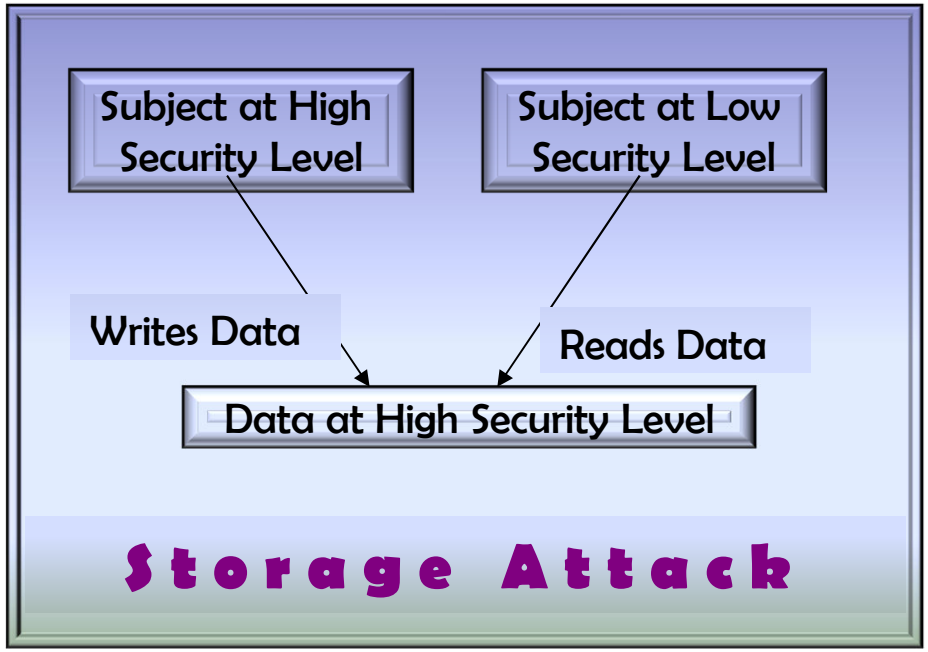
  - Covert Timing Channel – a process relays information to another by *modulating* its use of system resources.

  - Covert Storage Channel – a process writes data to a storage location, and another process of *lower* clearance reads it.



Covert Channel Attack Types





Emanations

- CRT emanation captured from 25 meters distance

Figure 6.8: Testchart displayed on the target monitor (VESA 640 × 480@85Hz video mode)

Agenda

- Fundamental concepts of security models
- Components of information systems security evaluation models
- Security capabilities of information systems
- Vulnerabilities of security architectures
- **Software and system vulnerabilities and threats**
- Countermeasure principles

Threats to Software and Systems

- Backdoors
- Timing Attacks
- Buffer Overflows
- Inference
- Aggregation

Circumventing Access Controls

Back Doors

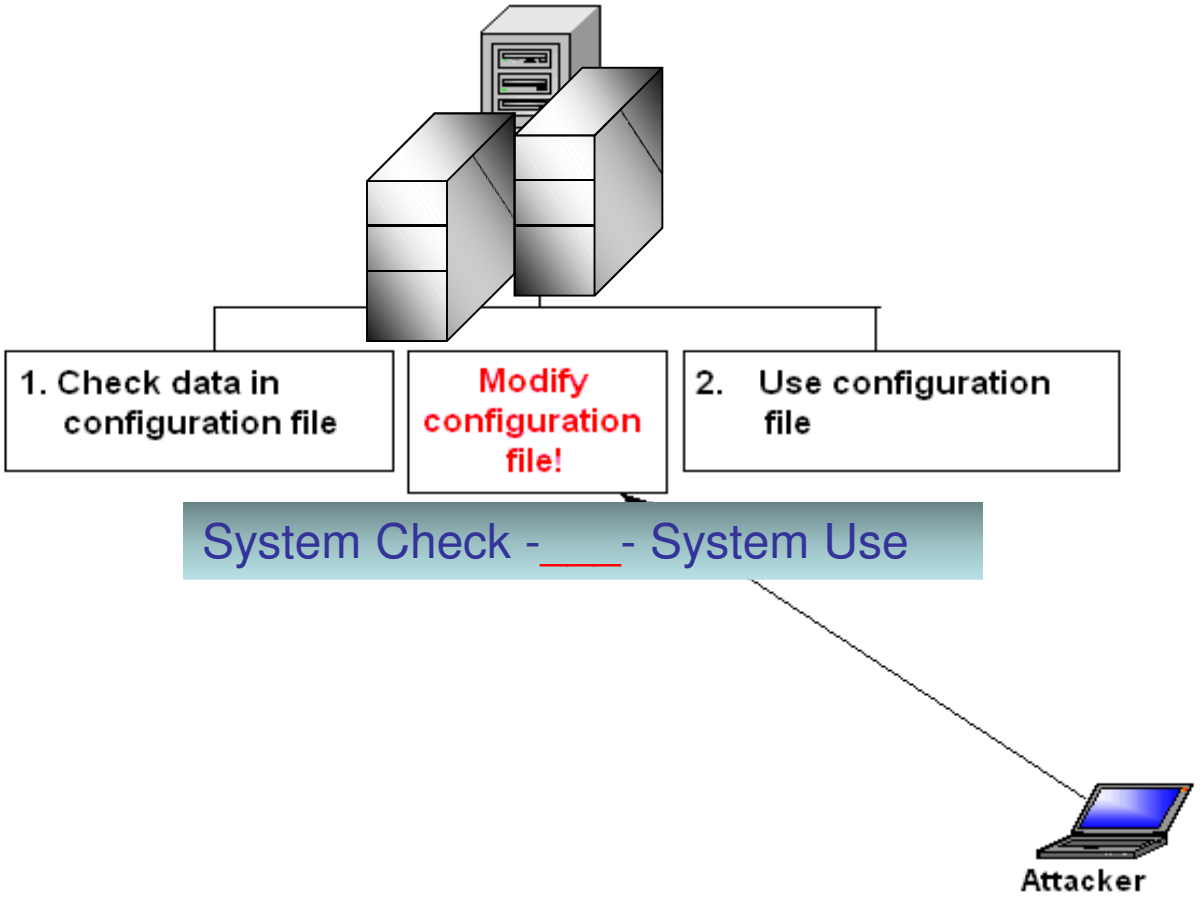
- Accessing a system by bypassing the access controls
- Allows attacker to enter the computer at any time
- Maintenance Hook/ Trapdoor/ Rootkits
- Can be inserted by a Trojan horse

Asynchronous Attacks - Timing

- **Take advantage of the time between events in a sequence**
  - Time of Check/Time of Use (TOC/TOU)
    - Attack takes place after the system checks a specific file of the system and before the system actually uses that file
  - Race Conditions
    - Two processes race to carry out conflicting actions at the same time
    - By slowing down one process or speeding up another process, an attacker can take advantage of these conditions within a system

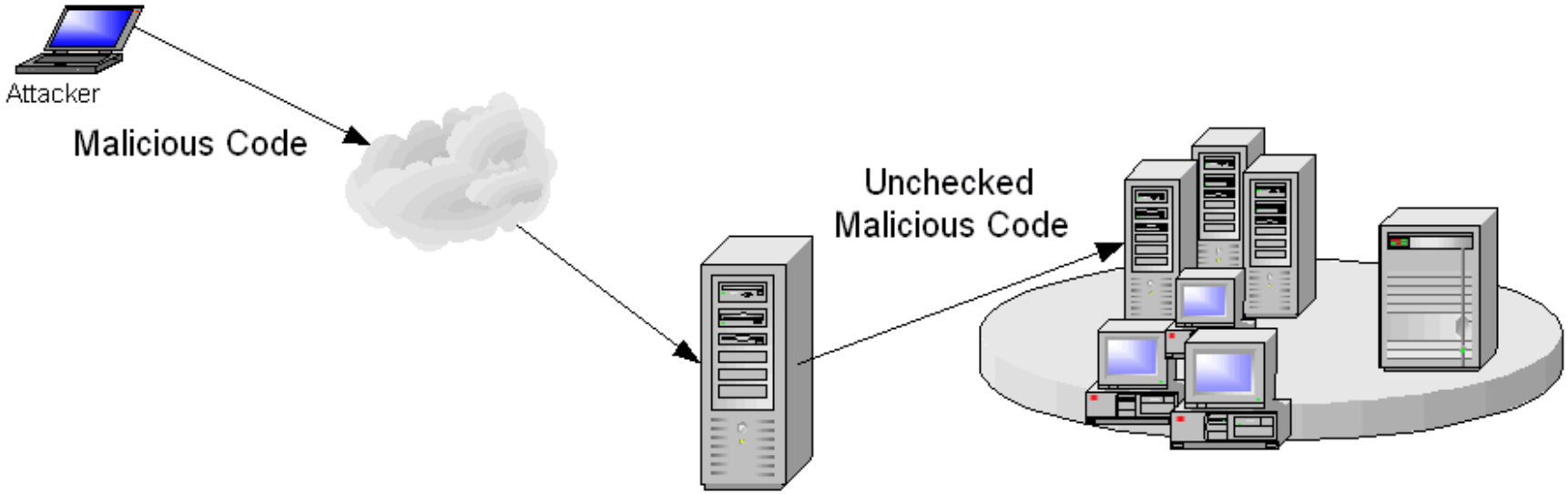
Asynchronous Attacks

Taking advantage of the time between events in a sequence



Data Validation

- **Confirms Criteria of Acceptable Data**
  - Validation is the process of reviewing data against a pre-established set of criteria.
  - Security checks on data along with validity checks to ensure it is in the proper format.



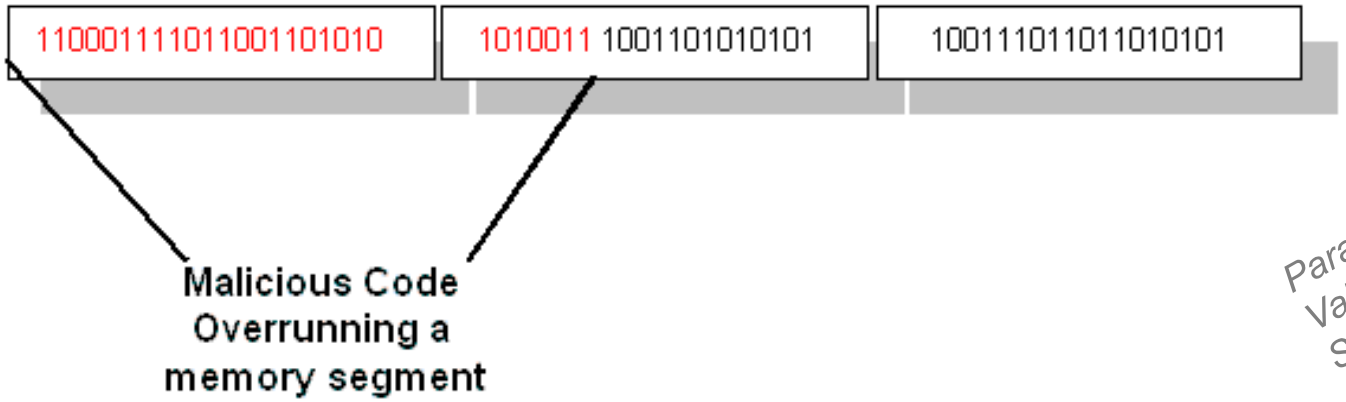
Attacking Through Applications

Code Injection

- Input must be validated for range/ type/ length
- Injecting code like SQL statements into input buffers

Buffer Overflow

- If an application does not verify the amount of information being input, the data can overwrite other memory segments
- An attack might cause code to execute in a privileged mode



Parameter Checking
Validation
Segmentation

Inference

- Definition - Inference is the act or process of deriving logical conclusions from premises known or assumed to be true. The conclusion drawn is also called an idiomatic. The laws of valid inference are studied in the field of logic.

- It could be “inferred” that since a large shipment of Dell computers was made, that an equally large number of returns would be made in the near future.

- From that it could be “inferred” that some of those old computers might not have had their hard drives sanitized and posing as a re-cycler, a malicious user could gain access to sensitive data.

Aggregation

- Definition - A massing together or clustering of independent but similar units, such as data elements.
- Combining information from different databases, could expose sensitive information, which, when looked at separately would not present an exposure.
- For, an example, if one data element showed a ship departing a US port, and another data element showed a shipment of weapons, and another database element showed customs clearance for a foreign country, the aggregated information could be considered confidential.

Agenda

- Fundamental concepts of security models
- Components of information systems security evaluation models
- Security capabilities of information systems
- Vulnerabilities of security architectures
- Software and system vulnerabilities and threats
- **Countermeasure principles**

Countermeasure Principles

Defense in Depth

Countermeasure Principles

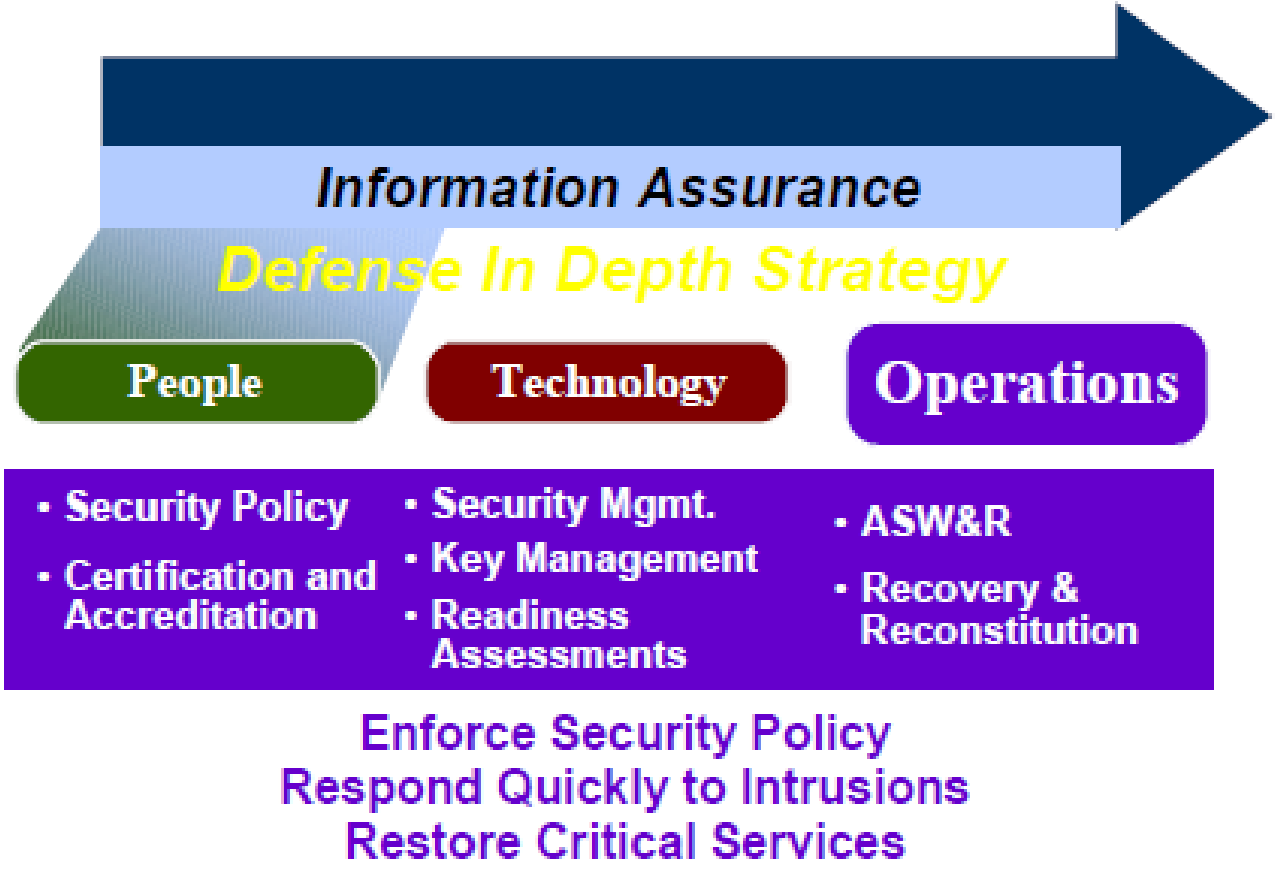
- Defense in Depth – people perspective

Countermeasure Principles

- Defense in Depth – technology perspective

Countermeasure Principles

Defense in Depth – Operations perspective



Bringing Things Together

- Many different components within the system must help protect the system overall – the TCB
- Operating systems have several protection mechanisms to protect themselves
- There are many different models that can be implemented to protect a system's confidentiality or integrity
- Security architectures, software and systems vulnerabilities and threats must be understood
- Defense-in-depth provides the best protection