# Access Control Models

SIO

André Zúquete

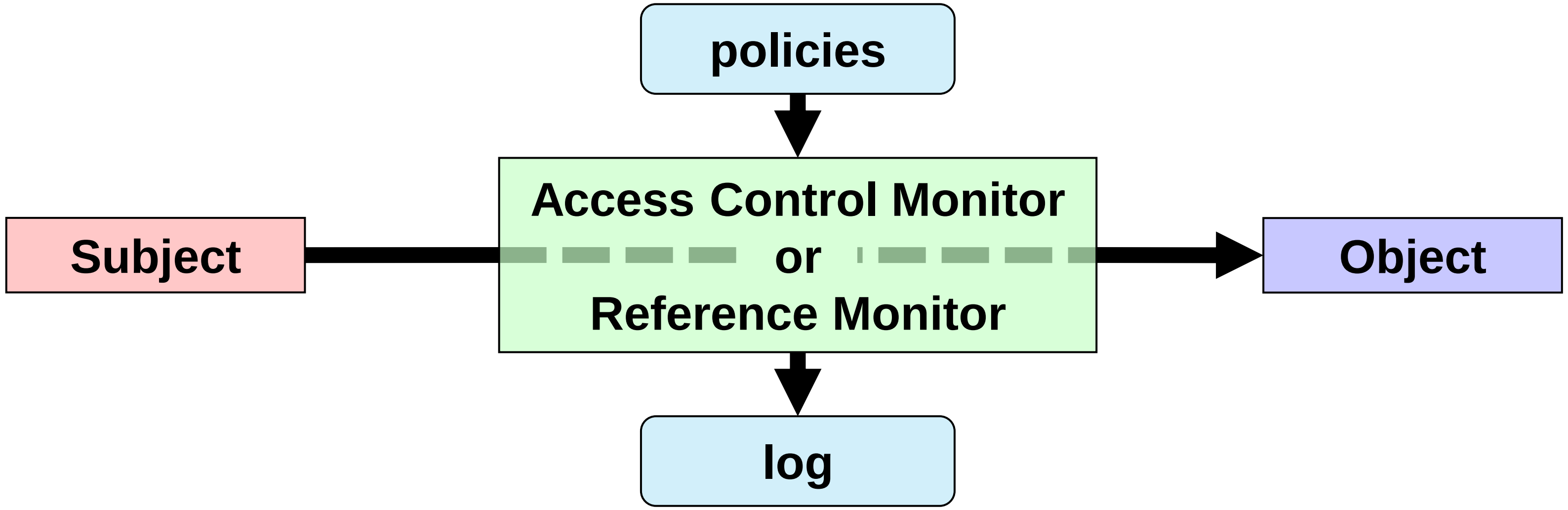
deti universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

# Access types

- Physical access
  - Physical contact between a subject and the object of interest
    - Facility, room, network, computer, storage device, authentication token, etc.
  - Out of scope of this course …

- Informatic or electronic access
  - Information-oriented contact between a subject and the object of interest
    - Contact through request-response dialogs
  - Contact is mediated by
    - Computers and networks
    - Operating systems, applications, middleware, devices, etc.

# Access control




- Definition
  - Policies and mechanisms that mediate the access of a subject to an object

- Normal requirements
  - Authentication
    - With some Level of Assurance (LoA)
  - Authorization policies
  - Accountability → logging

AAA

# Access control: Subjects and Objects

- Both are digital entities
- Subjects are something exhibiting activity:
  - Processes
  - Computers
  - Network elements
- Objects are targets of actions (resources):
  - Stored data
  - CPU time
  - Memory
  - Processes
  - Computers
  - Networks
- An entity can be both subject & object

# Least privilege principle

**“Every program and every user of the system should operate using the least set of privileges necessary to complete the job”**

J. H. Saltzer, M. D. Schroeder, Proc. of The protection of information in computer systems, IEEE, 63(9) 1975

- Privilege:
  - Authorization to perform a given task
  - Similar to access control clearance

- Subjects should have, at any time, the exact privileges required to their assigned tasks
  - Less privileges than the required create unsurpassable barriers
  - More privileges than the required create vulnerabilities
    - Damage resulting from accidents or errors
    - Potential interactions among privileged programs
    - Misuse of a privileges
    - Unwanted information flows
      - "need-to-know" military restrictions

# Access control models

- Access control matrix
  - Matrix with all access rights for subjects relatively to objects
  - Conceptual organization

| | O1 | O2 | ... | Om-1 | Om |
|---|---|---|---|---|---|
| S1 | | Access rights | | | |
| S2 | | | | | |
| ... | | | | | |
| Sn-1 | | | | | |
| Sn | | | | | |

# Access control models

| | O1 | O2 | … | Om-1 | Om |
|---|---|---|---|---|---|
| S1 | | Access rights | | | |
| S2 | | | | | |
| … | | | | | |
| Sn-1 | | | | | |
| Sn | | | | | |

- ACL-based mechanisms
  - ACL: Access Control List
  - Matrix column
- List of access rights for specific subjects
  - Access rights can be positive or negative
  - Default subjects may often be used
- Usually, ACLs are stored along with objects
  - e.g., for file system objects

# Access control models

|  | O1 | O2 | … | Om-1 | Om |
|---|---|---|---|---|---|
| S1 |  | Access rights |  |  |  |
| S2 |  |  |  |  |  |
| … |  |  |  |  |  |
| Sn-1 |  |  |  |  |  |
| Sn |  |  |  |  |  |

- Capability-based mechanisms
  - Capability: unforgeable authorization token
  - Matrix row
  - Contains object references and access rights
- Access granting
  - Transmission of capabilities between subjects
  - Mediated / non-mediated
- Usually, capabilities are kept by subjects
  - e.g., OAuth 2.0 access tokens

# Access control models: MAC and DAC

- Mandatory access control (MAC)
  - Fixed access control policy implemented by the access control monitor
  - Access control rights cannot be tailored by subjects or object owners
- Discretionary access control (DAC)
  - Some subjects can update rights granted or denied to other subjects for a given object
  - Usually this is granted to object owners and system administrators

# Access control models: Role-Based Access Control (RBAC)

D.F. Ferraiolo and D.R. Kuhn, "Role Based Access Control“,
15th National Computer Security Conference, Baltimore, Oct. 1992

- Not DAC or MAC
  - Roles are dynamically assigned to subjects
  - For access control what matters is the role played by the subject
    - And not the subject’s identity
    - Identity is mostly relevant for role access and logging
- Access control binds roles to (meaningful) operations
  - Operations are complex, meaningful system transactions
    - Not the ordinary, low-level read/write/execute actions on individual objects
  - Operations can involve many individual, lower-level objects

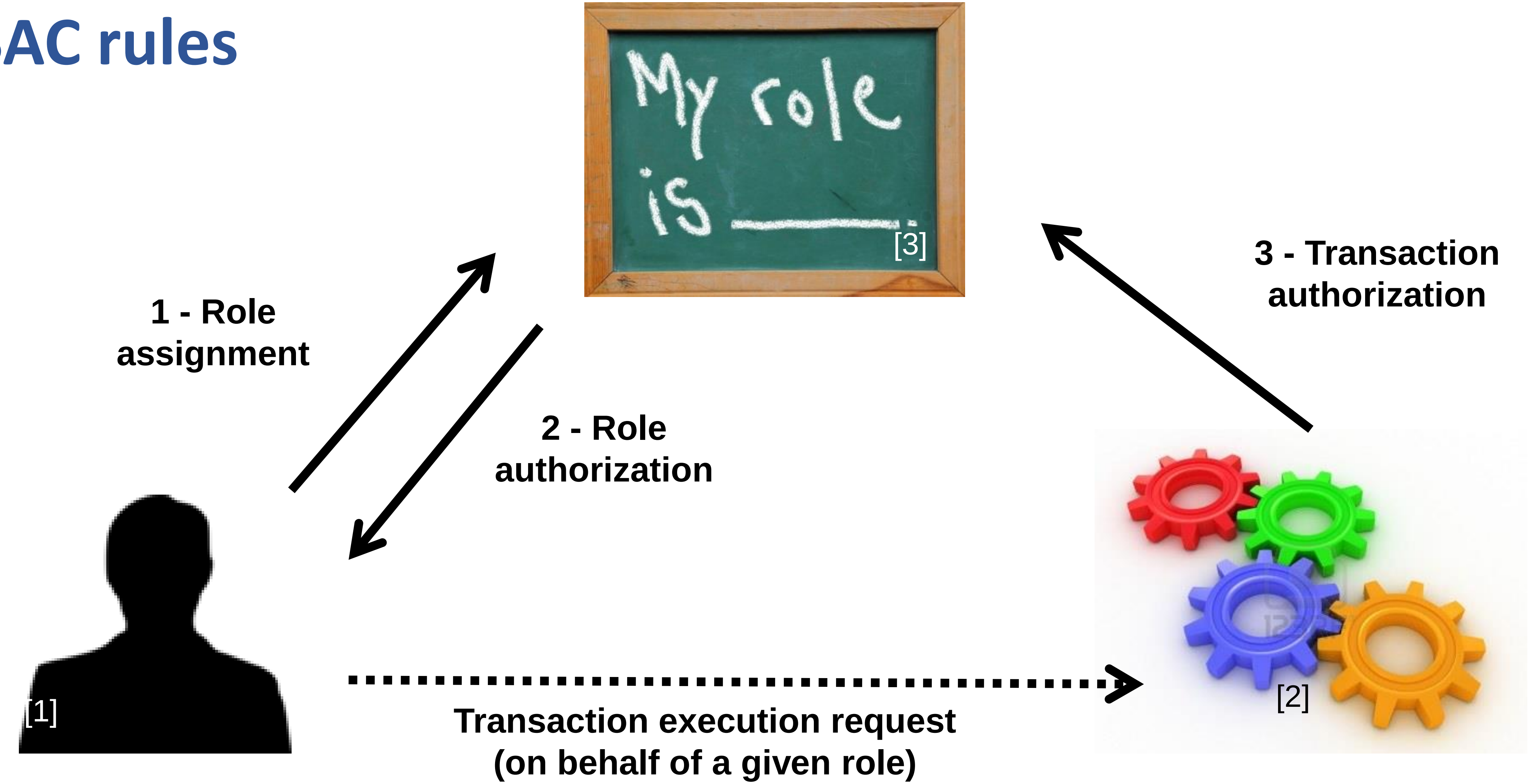
# Access control models: RBAC role assignment

- All subject activity on the system is conducted through transactions
  - And transactions are allowed to specific roles
  - Thus, all active subjects are required to have some active role
- A subject can execute a transaction iff it has selected or been assigned a role which can use the transaction

# Access control models: RBAC role authorization

- A subject's active role must be authorized for the subject
  - In that case, the subject may assume the role
- Transaction authorization:
  - A subject can execute a transaction iff
    - the transaction is authorized through the subject's role memberships
  - and
    - there are no other constraints that may be applied across subjects, roles, and permissions

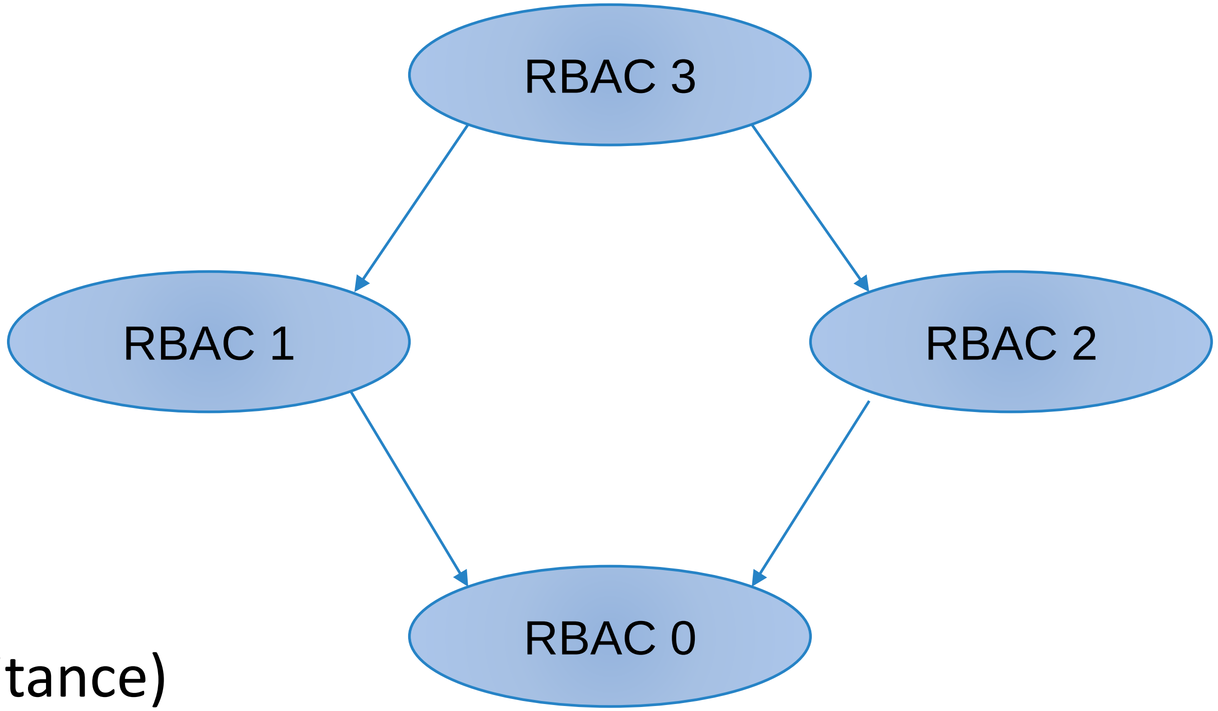
# RBAC rules




[1] From http://www.clker.com/clipart-24011.html
[2] From http://www.123rf.com/photo_12115593_three-dimensional-colored-toothed-wheels.html
[3] From http://www1.yorksolutions.net/Portals/115255/images/MyRoleIs.jpg

# RBAC variants

- RBAC 0
  - No role hierarchies
  - No role constraints
- RBAC 1
  - RBAC 0 w/ role hierarchies (privilege inheritance)
- RBAC 2
  - RBAC 0 w/ role constraints (separation of duties)
- RBAC 3
  - RBAC 1 + RBAC 2



# NIST RBAC model

- Flat RBAC
  - Simple RBAC model w/ user-role review
- Hierarchical RBAC
  - Flat RBAC w/ role hierarchies (DAG or tree)
  - General and restricted hierarchies
- Constraint RBAC
  - RBAC w/ role constraints for separation of duty
- Symmetric RBAC
  - RBAC w/ permission-role review

<u>User-role review</u>

Which users can have a role?

Role $\xrightarrow{?}$ users

Which roles can a user have?

User $\xrightarrow{?}$ roles

<u>Permission-role review</u>

Which permissions has a role?

Role $\xrightarrow{?}$ permissions

Which roles have a permission?

Permission $\xrightarrow{?}$ roles

# Access control models: Context-Based Access Control (CBAC)

- Access rights have an historical context
  - The access rights cannot be determined without reasoning about past access operations

Ver o histório do subject antes de dar role

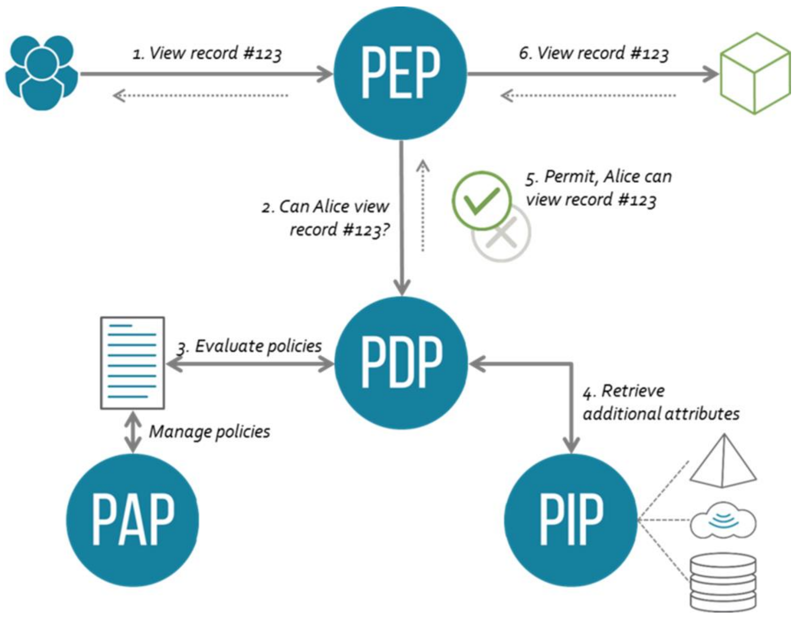
- Example: Stateful packet filter firewall

- Example: Chines Wall policy
  - Conflict groups
  - Access control policies need to address past accesses to objects from members of conflict groups

D.F.C. Brewer and M.J. Nash,
"**The Chinese Wall Security Policy** “,
IEEE Symposium on Security and Privacy, 1989

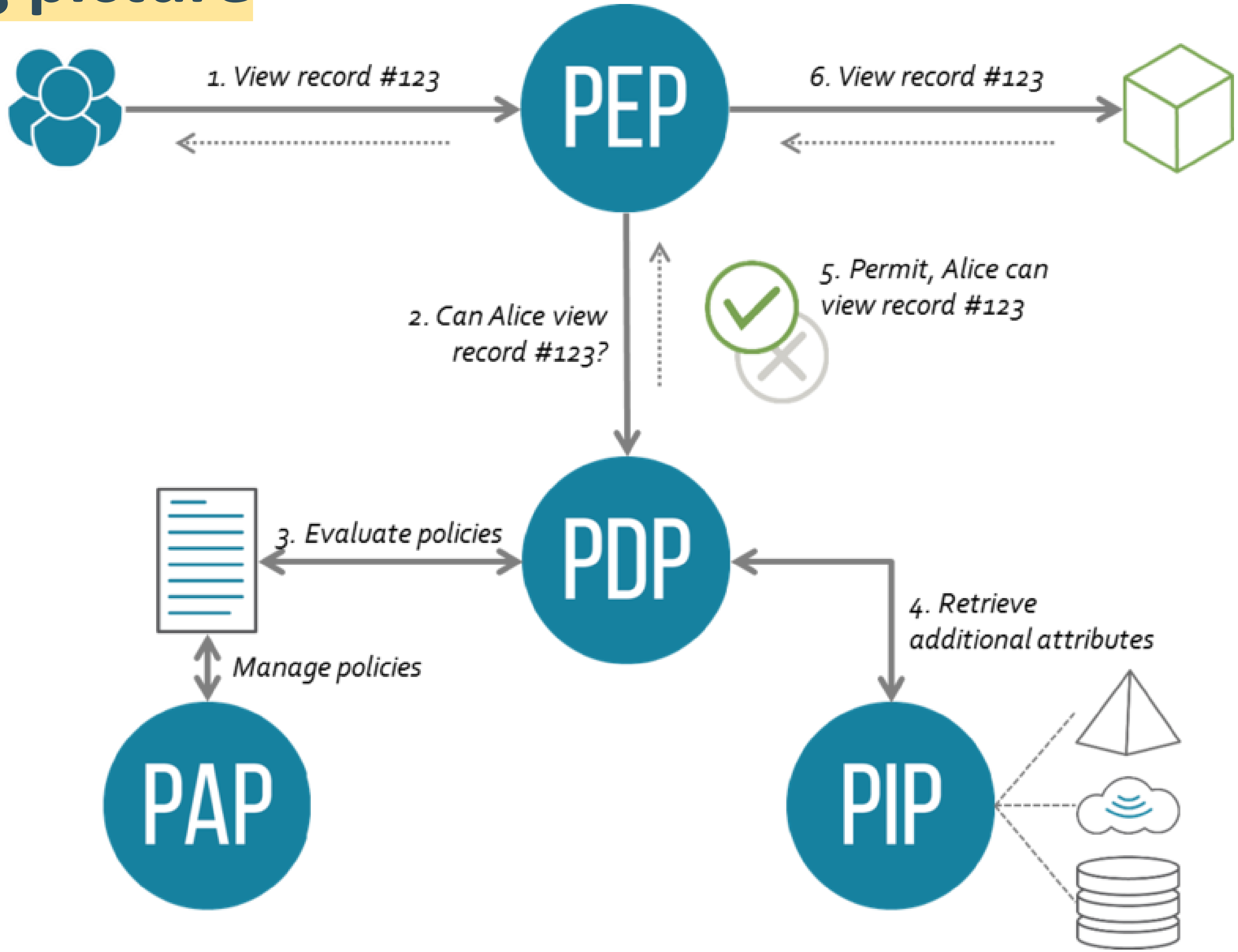
# Access control models: Attribute-Based Access Control (ABAC)

- Access control decisions are made based on attributes associated with relevant entities




- OASIS XACML architecture
  - Policy Administration Point (PAP)
    - Where policies are managed
  - Policy Decision Point (PDP)
    - Where authorization decisions are evaluated and issued
  - Policy Enforcement Point (PEP)
    - Where resource access requests are intercepted and confronted with PDP’s decisions
  - Policy Information Point (PIP)
    - Where the PDP gets external information

# XACML big picture



# XACML: Access control with PEP and PDP

- A subject sends a request
  - Which is intercepted by the Policy Enforcement Point (PEP)
- The PEP sends an authorization request to the Policy Decision Point (PDP)
  - With some subject's attributes
- The PDP evaluates the authorization request against its policies and reaches a decision
  - Which is returned to the PEP

  - Policies are retrieved from a Policy Retrieval Point (PRP)
  - Useful attributes are fetched from Policy Information Points (PIP)
  - Policies are managed by the Policy Administration Point (PAP)

# Access control models: Break-the-glass

- In some scenarios it may be required to overcome the established access limitations
  - e.g., in a life-threatening situation
- In those cases, the subject may be presented with a break-the-glass decision upon a deny
  - Can overcome the deny at their own responsibility
  - Logging is fundamental to prevent abuses

# Separation of Duties

R.A. Botha, J.H.P. Eloff, “Separation of duties for access control enforcement in workflow environments”, IBM Systems Journal, 2001

- Fundamental security requirement for fraud and error prevention
  - Dissemination of tasks and associated privileges for a specific business process among multiple subjects
  - Often implemented with RBAC

- Damage control
  - Segregation of duties helps reducing the potential damage from the actions of one person
  - Some duties should not be combined into one position

# Segregation of duties

## ISACA (Inf. Systems Audit and Control Ass.) matrix guideline

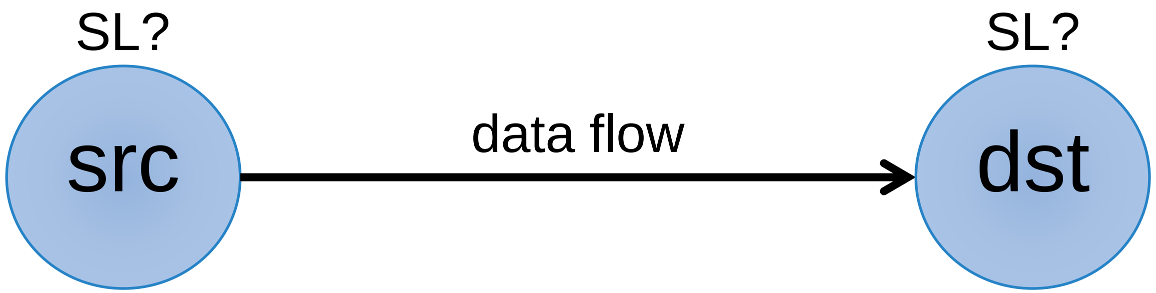
**Exhibit 2.9—Segregation of Duties Control Matrix**

| | Control Group | Systems Analyst | Application Programmer | Help Desk and Support Manager | End User | Data Entry | Computer Operator | Database Administrator | Network Administrator | Systems Administrator | Security Administrator | Systems Programmer | Quality Assurance |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Control Group | | X | X | X | | X | X | X | X | X | | X | |
| Systems Analyst | X | | | X | X | | X | | | | X | | X |
| Application Programmer | X | | | X | X | X | X | X | X | X | X | X | X |
| Help Desk and Support Manager | X | X | X | | X | X | | X | X | X | | X | |
| End User | | X | X | X | | | X | X | X | | | X | X |
| Data Entry | X | | X | X | | | X | X | X | X | X | X | |
| Computer Operator | X | X | X | | X | X | | X | X | X | X | X | |
| Database Administrator | X | | X | X | X | X | X | | X | X | | X | |
| Network Administrator | X | | X | X | X | X | X | X | | | | | |
| System Administrator | X | | X | X | | X | X | X | | | | X | |
| Security Administrator | | X | X | | | X | X | | | | | X | |
| Systems Programmer | X | | X | X | X | X | X | X | | X | X | | X |
| Quality Assurance | | X | X | | X | | | | | | | X | |

X—Combination of these functions may create a potential control weakness.

# Information flow models

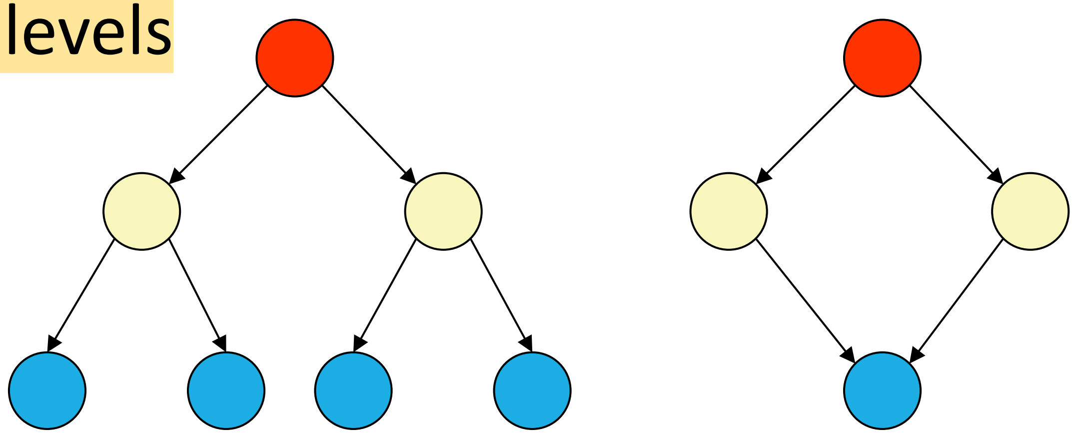
- Authorization is applied to data flows
  - Considering the data flow source and destination
  - Goal: avoid unwanted/dangerous information flows




- Src and Dst security-level attributes
  - Information flows should occur only between entities with given security level (SL) attributes
  - Authorization is given based on the SL attributes

# Multilevel security

- Subjects (or roles) act on different security levels
  - Levels do not intersect themselves
  - Levels have some partial order
    - Hierarchy
    - Lattice

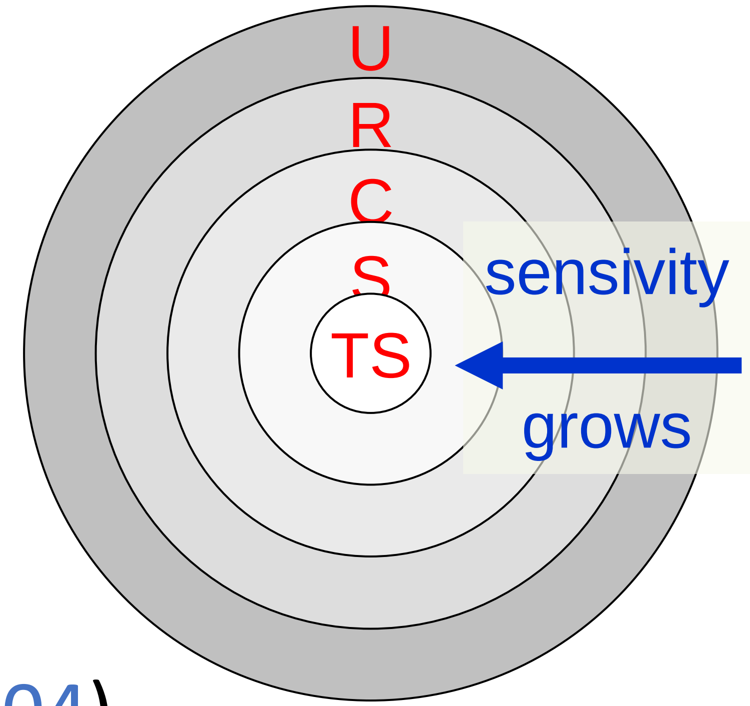


- Levels are used as attributes of subjects and objects
  - Subjects: security level clearance
  - Objects: security classification

- Information flows & security levels
  - Same security level → authorized
    - Still, restricted to a "need to know"
  - Different security levels → controlled

# MS levels: Military / Intelligence organizations

- Typical levels
  - Top secret
  - Secret
  - Confidential
  - Restricted
  - Unclassified

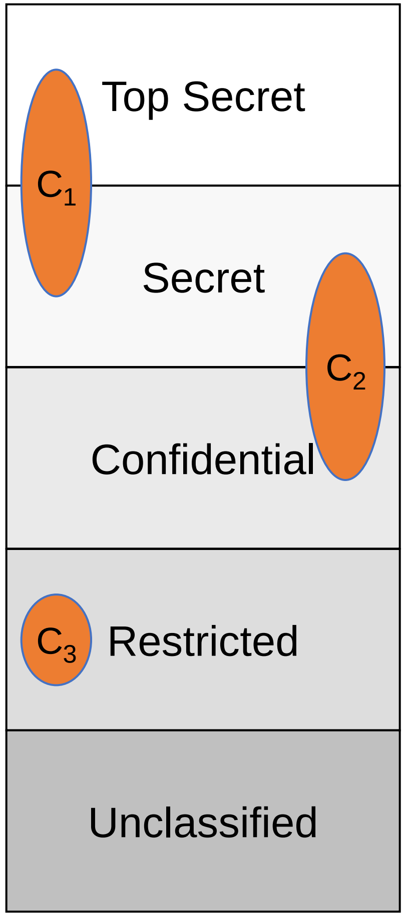



- EU example
  - EU TOP SECRET
  - EU SECRET
  - EU CONFIDENTIAL
  - EU RESTRICTED
  - EU COUNCIL / COMMISSION

- Portugal (NTE01, NTE04)
  - Muito Secreto
  - Secreto
  - Confidencial
  - Reservado

- NATO example
  - COSMIC TOP SECRET (CTS)
  - NATO SECRET (NS)
  - NATO CONFIDENTIAL (NC)
  - NATO RESTRICTED (NR)

# Security categories (or compartments)

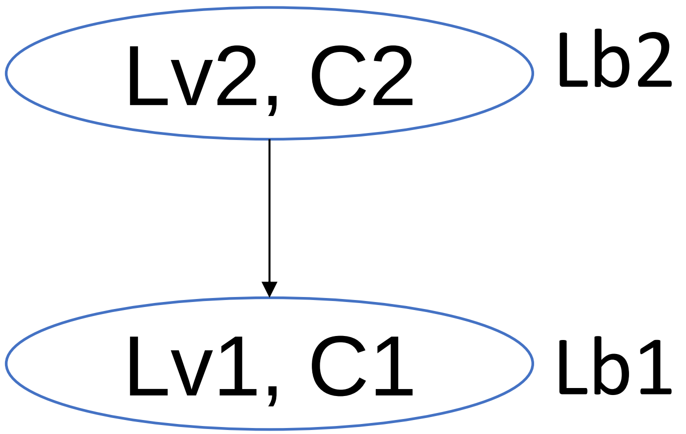



- Self-contained information environments
  - May span several security levels
- Military environments
  - Military branches, military units
- Civil environments
  - Departments, organizational units
- An object can belong to different compartments and have a different security classification in each of them
  - (top-secret, crypto), (secret, weapon)

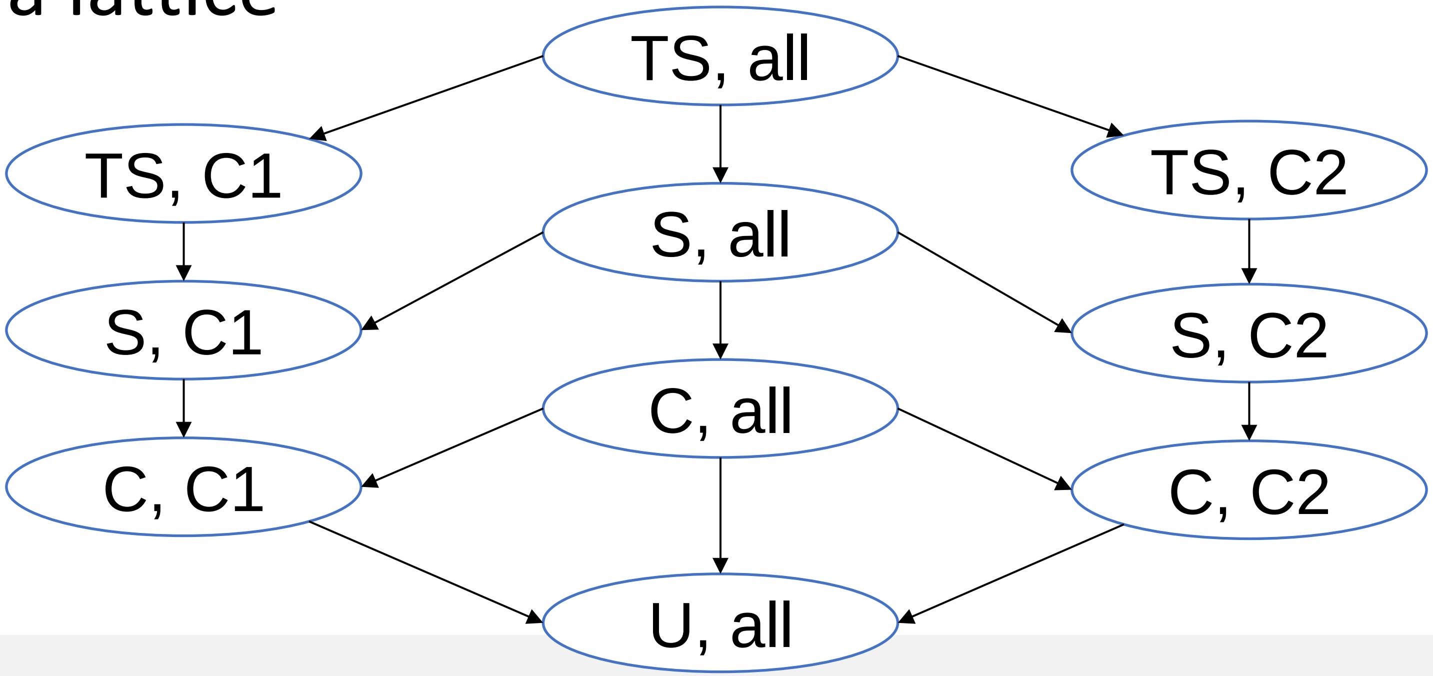
# Security labels

- Label = Category + Level

- Relative order between labels
  - Lb1 ≤ Lb2 ⇒ C1 ⊆ C2 ∧ Lv1 ≤ Lv2




- Labels form a lattice



# Bell-La Padula MLS Model

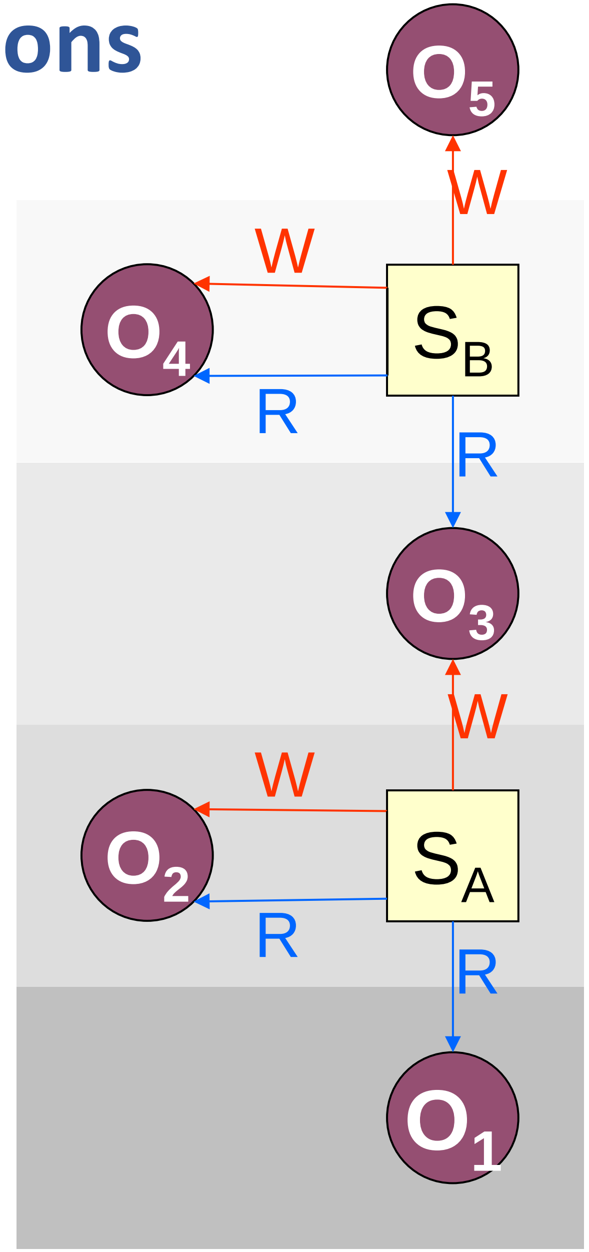
D. Elliott Bell, Leonard J. La Padula, "Secure Computer Systems: Mathematical Foundations", MITRE Tech. Report 2547, Vol. I, 1973

- Access control policy for controlling information flows
  - Addresses data confidentiality and access to classified information
  - Addresses disclosure of classified information
    - Object access control is not enough
    - One needs to restrict the flow of information from a source to authorized destinations

- Combines access control matrixes with MLS

# Bell-La Padula MLS Model: secure state transitions

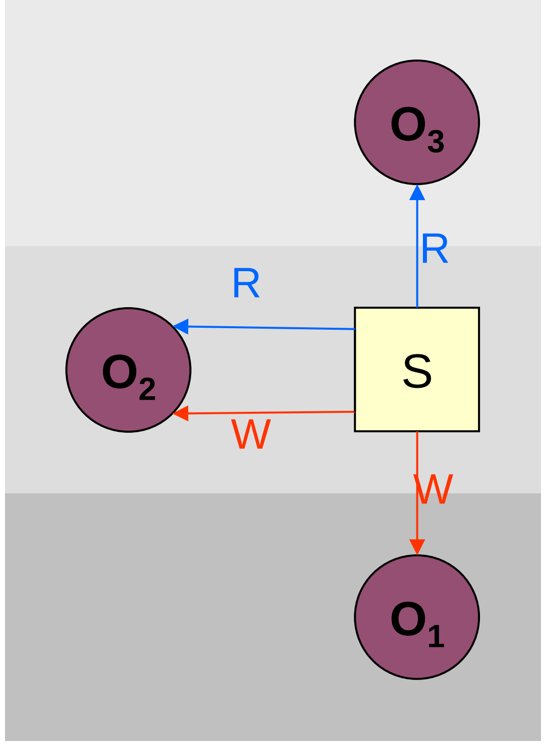
- Simple security condition (<u>no read up</u>)
  - S can read O iff L(S) ≥ L(O)
- *-property (<u>no write down</u>)
  - S can write O iff L(S) ≤ L(O)
  - aka confinement property
- Discretionary Security Property
  - DAC-based access control within the same security level



# Biba Integrity Model

K. J. Biba, "Integrity Considerations for Secure Computer Systems", MITRE Technical Report 3153, The Mitre Corporation, April 1977

- Access control policy for enforcing integrity control over data flows
  - Uses integrity levels, not security levels
  - Similar to Bell-La Padula, with inverse rules
- Simple Integrity Property (no read down)
  - S can read O iff $I(S) \le I(O)$
- Integrity *-Property (no write up)
  - S can write O iff $I(S) \ge I(O)$



# Windows mandatory integrity control

- Allows mandatory (priority and critical) access control enforcement prior to evaluate DACLs
  - If access is denied, DACLs are not evaluated
  - If access is allowed, DACLs are evaluated

# Windows mandatory integrity control: integrity labels

- Untrusted
- Low (or AppContainer)
- Medium
- Medium Plus
- High
- System
- Protected Process

# Windows mandatory integrity control: users and processes

- User integrity level
  - Medium: standard users
  - High: elevated users

- Process integrity level
  - The minimum associated to the owner and the executable file
  - User processes usually are Medium or High
    - Except if executing Low-labeled executables
  - Service processes: High

# Windows mandatory integrity control

- Securable objects mandatory label
  - NO_WRITE_UP (default)
  - NO_READ_UP
  - NO_EXECUTE_UP