Access control models

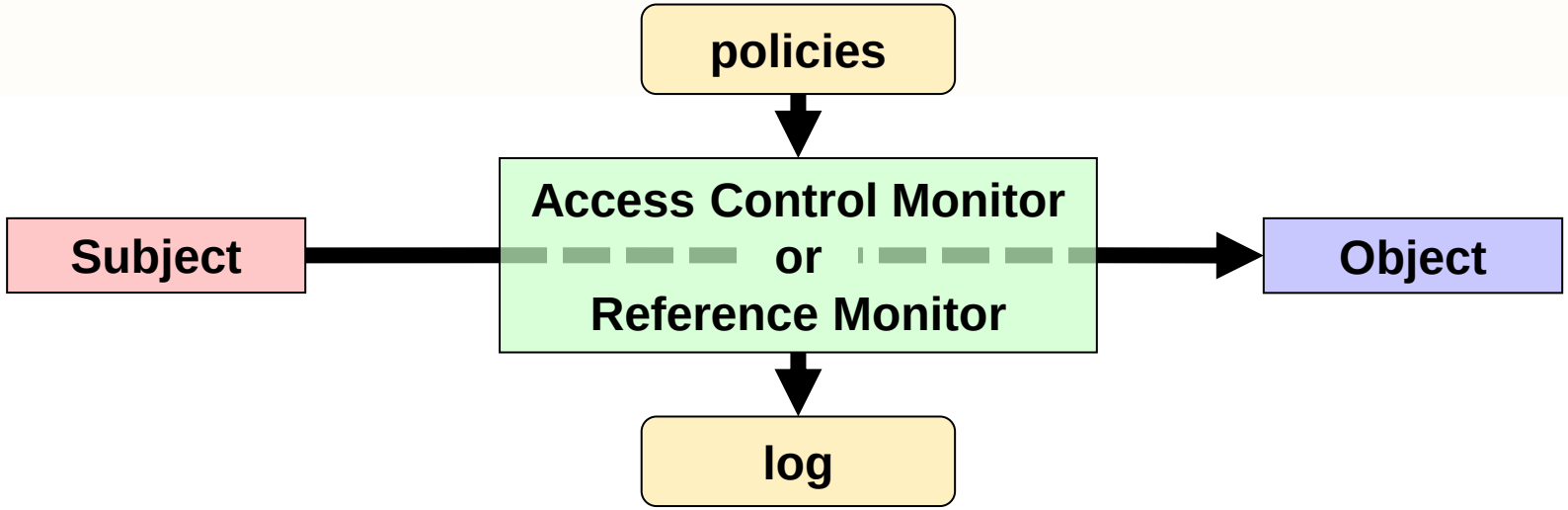
Access types

- 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 …**

MDA → contacto com o employer assim a dizer que não vai falar de um assunto fora de aula

- 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
  - The policies and mechanisms that mediate the access of a subject to an object
- Normal requirements
  - Authentication
    - With some Level of Assurance (LoA)
  - Authorization
  - Accountability ➔logging

AAA

Access control

- Subjects and objects
  - Both digital entities
  - Subjects can be something exhibiting activity :
    - Processes
    - Computers
    - Networks
  - Objects can be the target of an action :
    - Stored data
    - CPU time
    - Memory
    - Processes
    - Computers
    - Network
- An entity can be both subject and 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,
> The protection of information in computer systems, Proc. of the IEEE, 63(9) 1975

- Privilege:
  - Authorization to perform a given task
  - Similar to access control clearance
- Each subject should have, at any given time, the exact privileges required to the 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

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

- Access control matrix
  - Matrix with all access rights for subjects relatively to objects
  - Represents a conceptual model of the organization

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.
  - Rights are then mapped to specific actions
    - Same right may map to different actions on different contexts

Access control models

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

Linux uses this

restrictions for each user

- Capability-based mechanisms ≠ Groups
  - Capability: unforgeable authorization token (matrix row)
    - Contains object references and access rights
  - Access granting
    - Transmission of capabilities between subjects
  - Usually capabilities are kept by subjects
    - e.g. OAuth 2.0 access tokens

Access control kinds: MAC and DAC

root tem acesso a tudo

- Mandatory access control (MAC)
  - Access control policy statically 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 kinds:
Role-Based Access Control (RBAC)

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

- Not DAC or MAC
  - Roles are dynamically assigned to subjects
    - For access control it matters the role played by the subject and not the subject's identity

you can have only one role each time

- 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 kinds: RBAC rules (1/2)

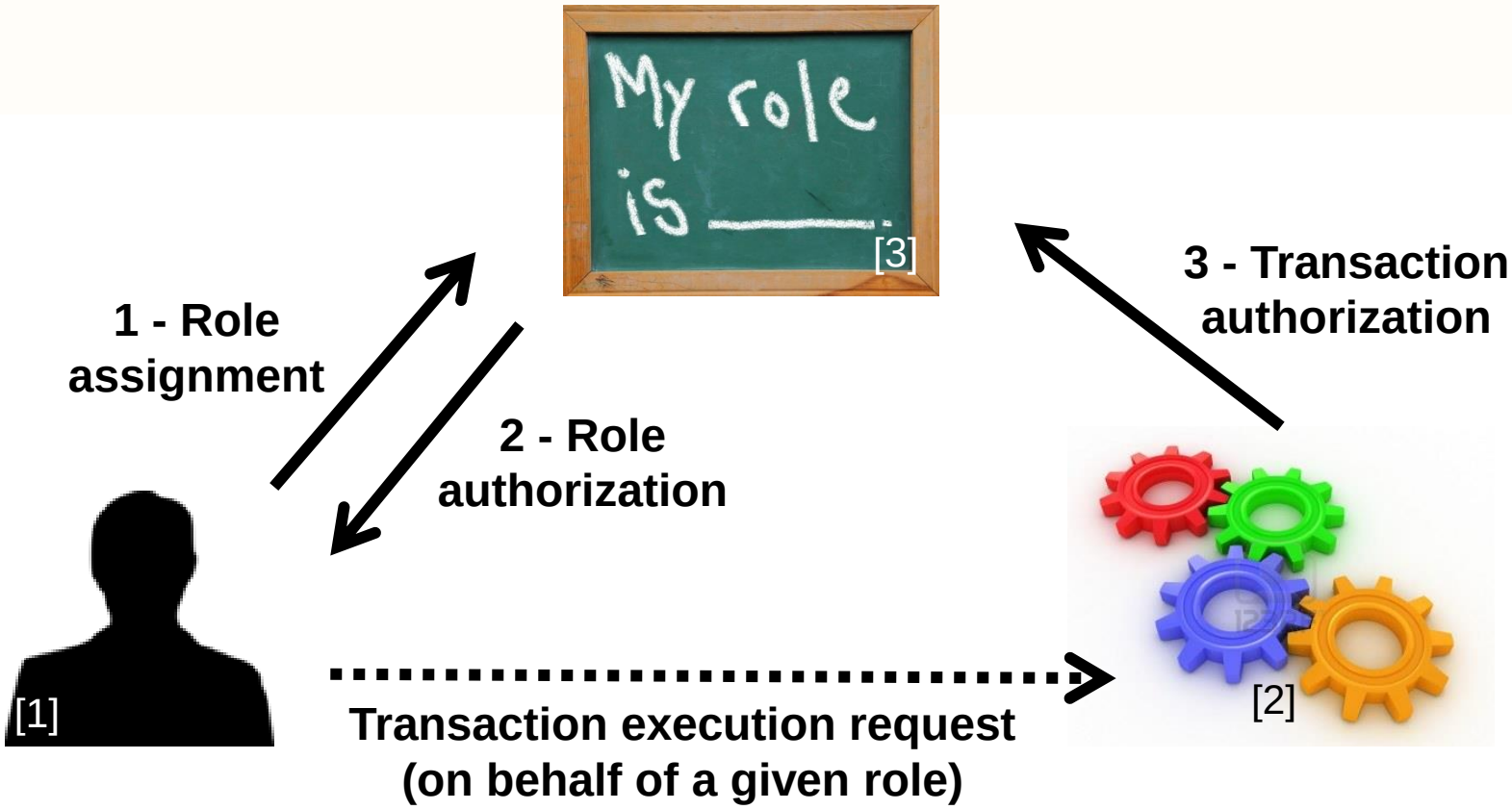
- Role assignment:
  - All subject activity on the system is conducted through transactions
    - And transactions are allowed to specific roles
    - Thus all active subjects are required to have some active role
  - A subject can execute a transaction **iff**
    - it has selected
  - or
    - been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

Access control kinds: RBAC rules (2/2)

- Role authorization:
  - A subject's active role must be authorized for the subject
- 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

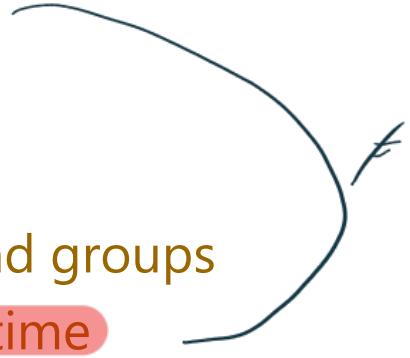
RBAC rules



[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:
Roles vs. groups

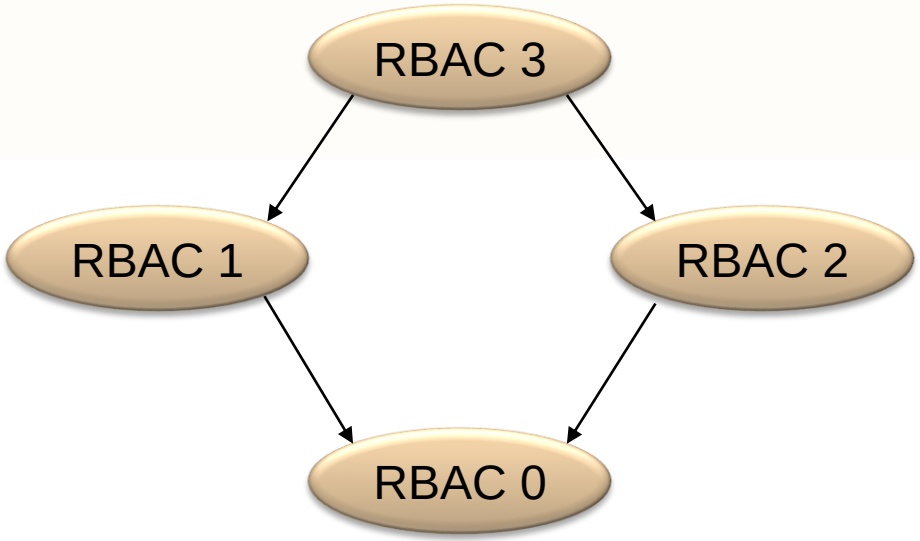
- Roles are a collection of permissions
  - The permissions are granted to the subjects that, at a given instant, play the role
  - A subject can only play a role at a given time



- Groups are a collection of users
  - And permissions can be granted both to users and groups
  - A subject can belong to many groups at a given time

- The session concept
  - Role assignment is similar to a session activation
  - Group membership is ordinarily a static attribute

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

→ Não pode ter roles com conflitos

NIST RBAC model

- Flat RBAC
  - Simple RBAC model w/ user-role review
  - Role provides specific permissions for the user
- Hierarchical RBAC
  - Flat RBAC w/ role hierarchies (DAG or tree)
  - General and restricted hierarchies, where Roles gain additional permissions from other roles
- Constraint RBAC
  - RBAC w/ role constraints for separation of duty
  - Static: Conflicting Roles cannot be assigned
  - Dynamic: Subject cannot activate conflicting Roles within session
- Symmetric RBAC → list with all the roles
  - RBAC w/ organization wide permission-role review
  - Allows review of a subject roles to prevent bloat → acesso a roles que já não precisa mais foram especificados

Access control kinds:
Context-Based Access Control (CBAC)

- Access rights have an historical context
  - The access rights cannot be determined without reasoning about past access operations
  - Example:
    - Stateful packet filter firewall
- Chinese Wall policy
  - Conflict groups
  - Access control policies need to address past accesses to objects in different 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 kinds: Attribute-Based Access Control (ABAC)

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

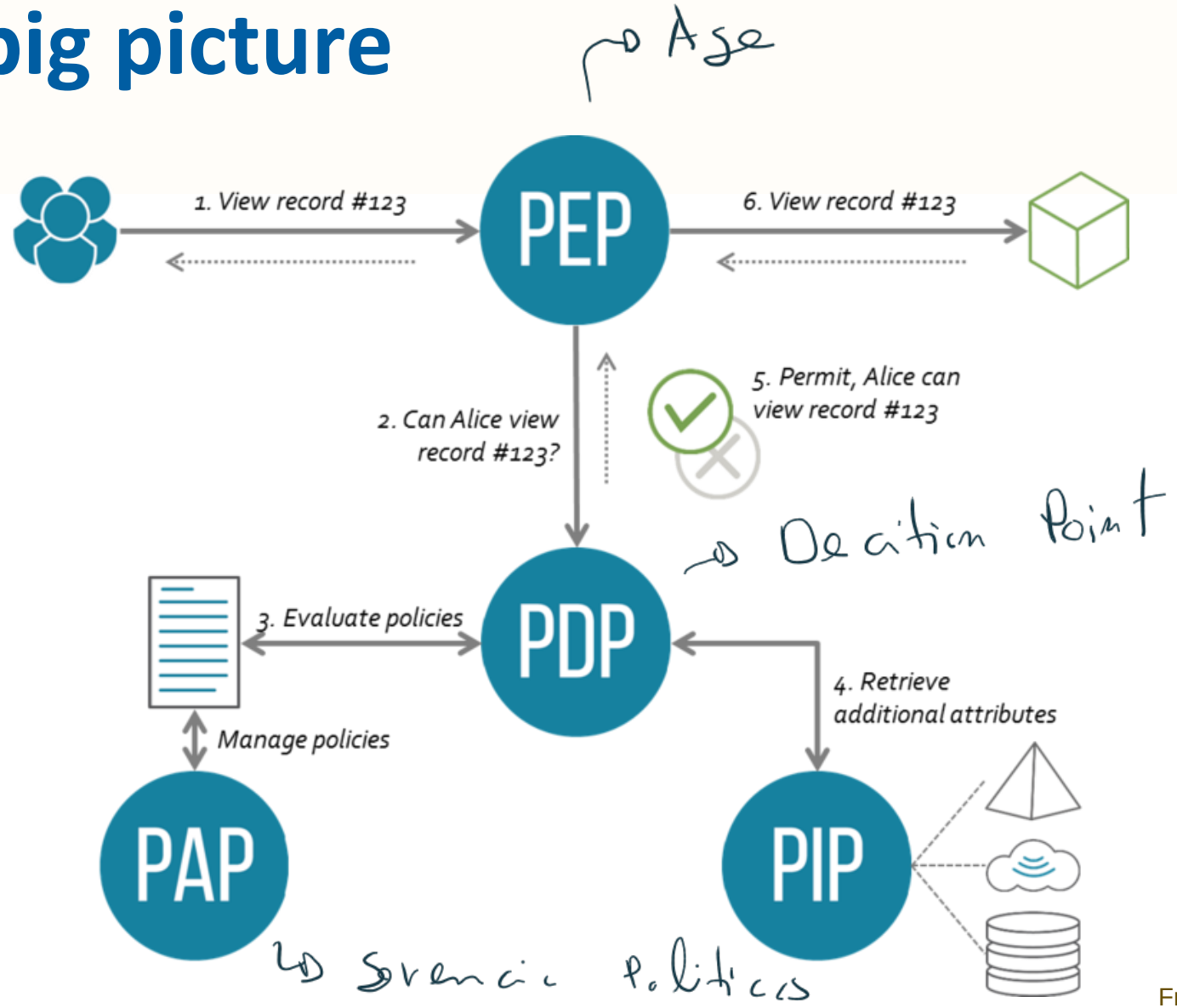
- OASIS XACML architecture
  - Policy Administration Point (PAP)
    - Where policies are managed
  - Policy Decision Point (PDP)
    - Where authorization decisions are evaluated and issued
  - Policy Enforcement Point (PEP)
    - Where access requests to a resource are intercepted and confronted with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

XACML:
Access control with PEP and PDP

- A subject sends a request
  - Which is intercepted by the Policy Enforcement Point (PEP)
- The PEP sends the authorization request to the Policy Decision Point (PDP)
- The PDP evaluates the 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)

XACML big picture



From https://en.wikipedia.org/wiki/XACML

Break-the-glass access control model

- It may be required to overcome the established access limitations
  - e.g. in a life threatening situation

- 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
    - Subject may have to justify action, after using the elevated right

If a user breaks policy. For better security on this cases the user should be asked on why he needed that action to be done.

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.

X marks an incompatibility

Segregation of duties:

Declaração de Práticas de Certificação da EC do Cartão de Cidadão

	Administração de Sistemas	Operação de Sistemas	Administração de Segurança	Auditoria de Sistemas	Custódia	Manutenção Sistemas de Suporte	Gestão
Administração de Sistemas			X	X	X	X	X
Operação de Sistemas			X	X	X	X	X
Administração de Segurança	X	X		X	X	X	X
Auditoria de Sistemas	X	X	X		X	X	X
Custódia	X	X	X	X		X	X
Manutenção Sistemas de Suporte	X	X	X	X	X		X
Gestão	X	X	X	X	X	X	

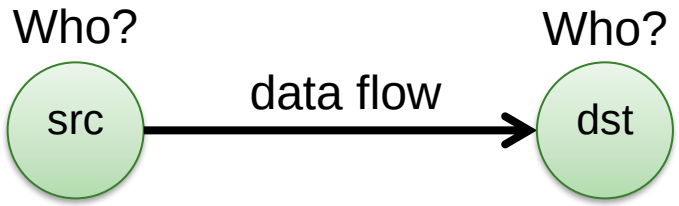
X marks an incompatibility

https://pki.cartaodecidadao.pt/publico/politicas/POL27.CPS_CC.pdf

Information flow models

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

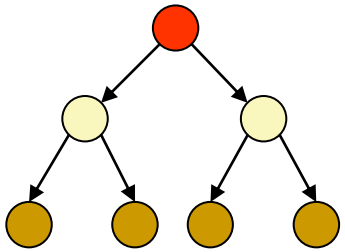
Pessoas de dep dif podem ser proibidos a falar



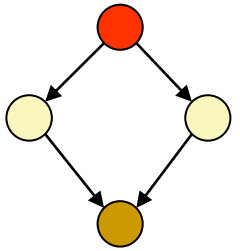
- Src and Dst security-level attributes
  - Information flows should occur only between entities with given security-level (SL) attributes
  - Authorization is given based on the SL attributes

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

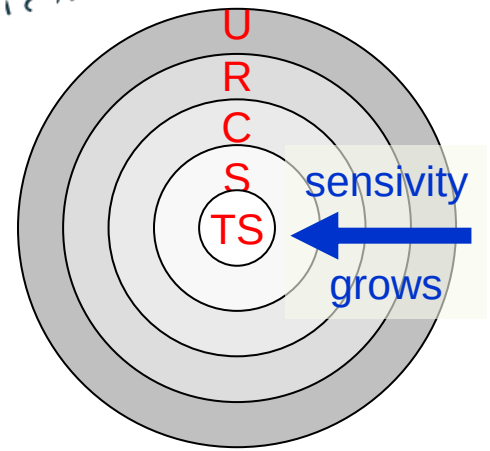


- Information flows & security levels
  - Same security level → authorized
  - Different security levels → controlled
    - Authorized or denied on a "need to know" basis

Multilevel security levels: Military / Intelligence organizations

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

→ Quem tem este acesso não tem necessariamente nos outros.



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

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

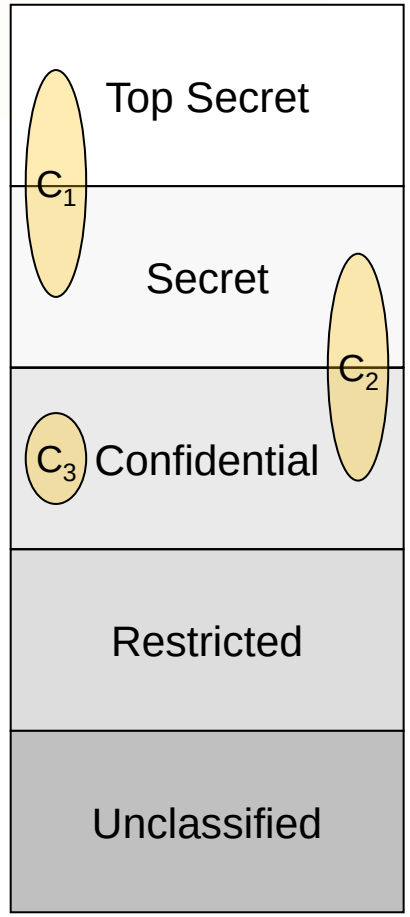
- NATO example:
  - COSMIC TOP SECRET (CTS)
  - NATO SECRET (NS)
  - NATO CONFIDENTIAL (NC)
  - NATO RESTRICTED (NR)

Multilevel security levels: Civil organizations

- Typical levels
  - Restricted
  - Proprietary
  - Sensitive
  - Public

Security categories (or compartments)

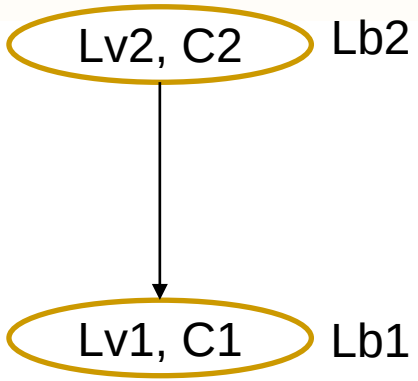
- Self-contained information environments
  - May span several security levels
- Military environments
  - Military branches, military units
- Civil environments
  - Departments, organizational units
- An object can belong to different compartments and have a different security classification in each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



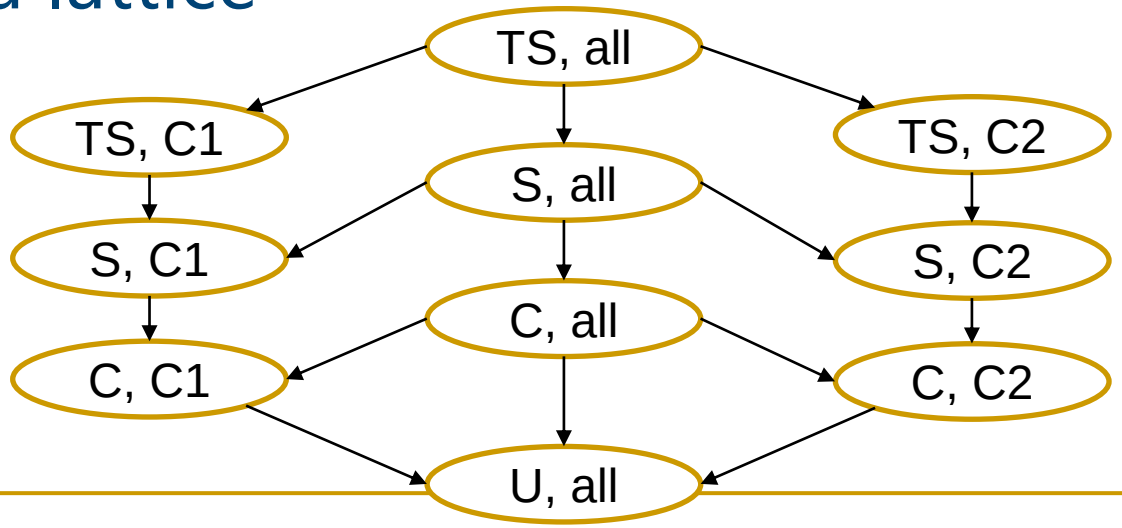
Security labels

- Label = Category + Level
- Relative order between labels

  $Lb1 \leq Lb2 \Rightarrow C1 \subseteq C2 \wedge Lv1 \leq Lv2$



- Labels form a lattice



Bell-La Padula MLS Model

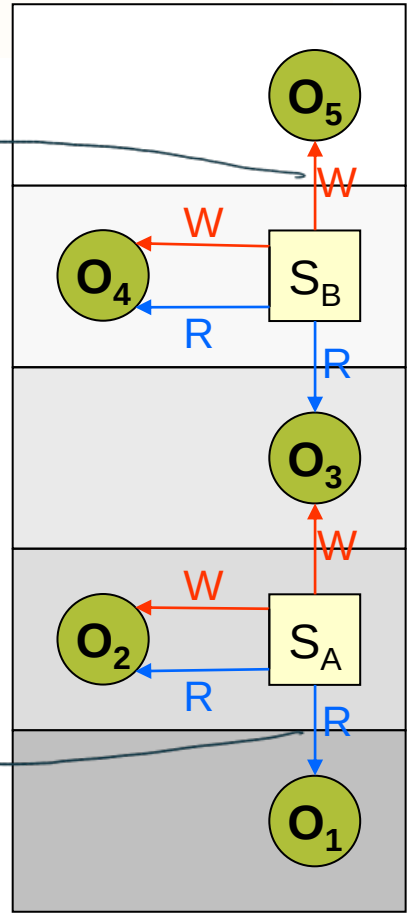
D. Elliott Bell, Leonard J. La Padula, "Secure Computer Systems: Mathematical Foundations", MITRE Technical Report 2547, Volume I, 1973

- 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

- Uses a state-transition model
  - In each state there are subjects, objects, an access matrix and the current access information
  - State transition rules
  - Security levels and clearances
    - Objects have a security labels
    - Subjects have security clearances
    - Both refer to security levels (e.g. CONFIDENTIAL)

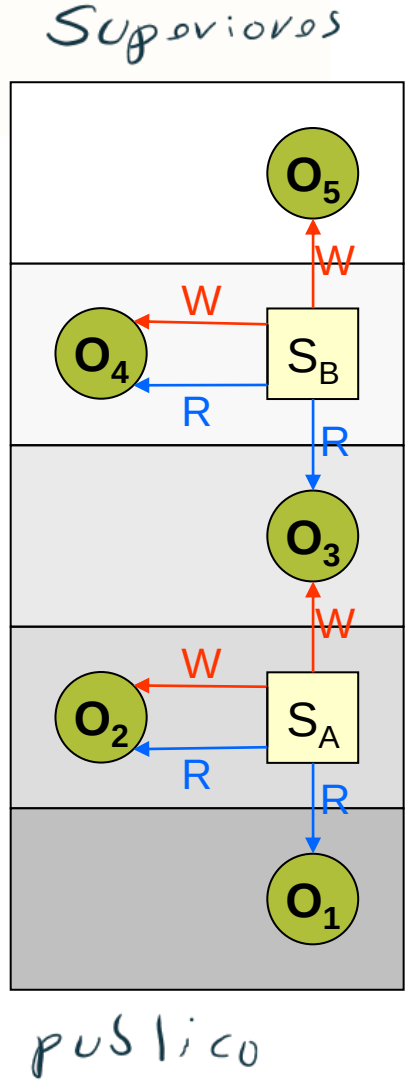
Bell-La Padula MLS Model: Secure state-transition model

- Simple security condition (no read up)
  - S can read O iff L(S) ≥ L(O)
- *-property (no write down)
  - S can write O iff L(S) ≤ L(O)
  - aka confinement property
- Discretionary Security Property
  - DAC-based access control



Bell-La Padula MLS Model: Secure state-transition model

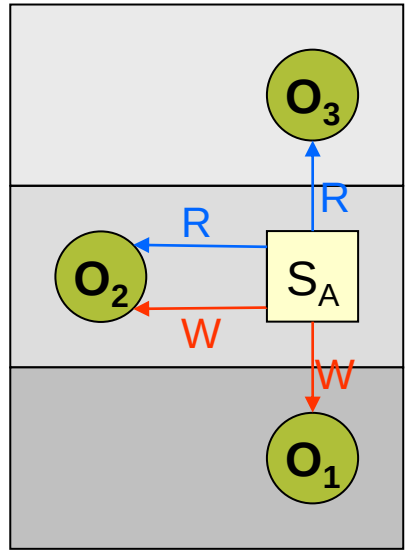
- Strong Star Property
  - S can read O iff L(S) = L(O)
- Tranquility Principle
  - Strong tranquility: S/O levels are static for the entire S/O lifetime
  - Weak tranquility: S/O levels may change if the security *spirit* of the system is not compromised
- Trusted Subjects
  - S can write to lower levels



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 controlling information flows
  - For enforcing data integrity control
  - Uses integrity levels, not security levels

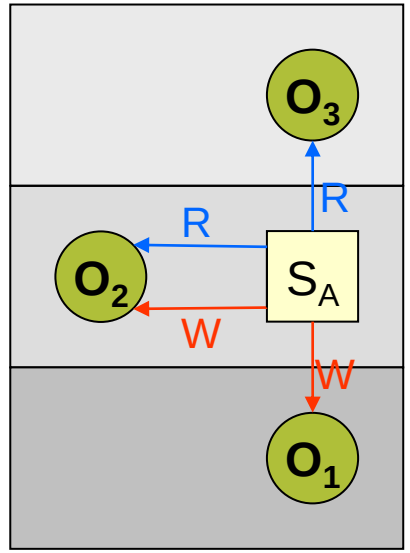


- Similar to Bell-La Padula, with inverse rules
  - Simple Integrity Property (no read down)
    - S can read O iff $I(S) \leq I(O)$
  - Integrity *-Property (no write up)
    - S can write O iff $I(S) \geq I(O)$

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 controlling information flows
  - For enforcing data integrity control
  - Uses integrity levels, not security levels
  - Subjects cannot corrupt objects at higher levels



- Similar to Bell-La Padula, with inverse rules
  - Simple Integrity Property (no read down)
    - S can read O iff I(S) ≤ I(O)
  - Integrity *-Property (no write up)
    - S can write O iff I(S) ≥ I(O)

- Invocation Property
  - S cannot request higher access

não pode ser dado mas não se pode pedir.

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

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

DACL: discretionary access control list

https://learn.microsoft.com/en-us/windows/win32/secauthz/mandatory-integrity-control

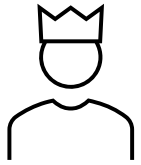
Windows mandatory integrity control

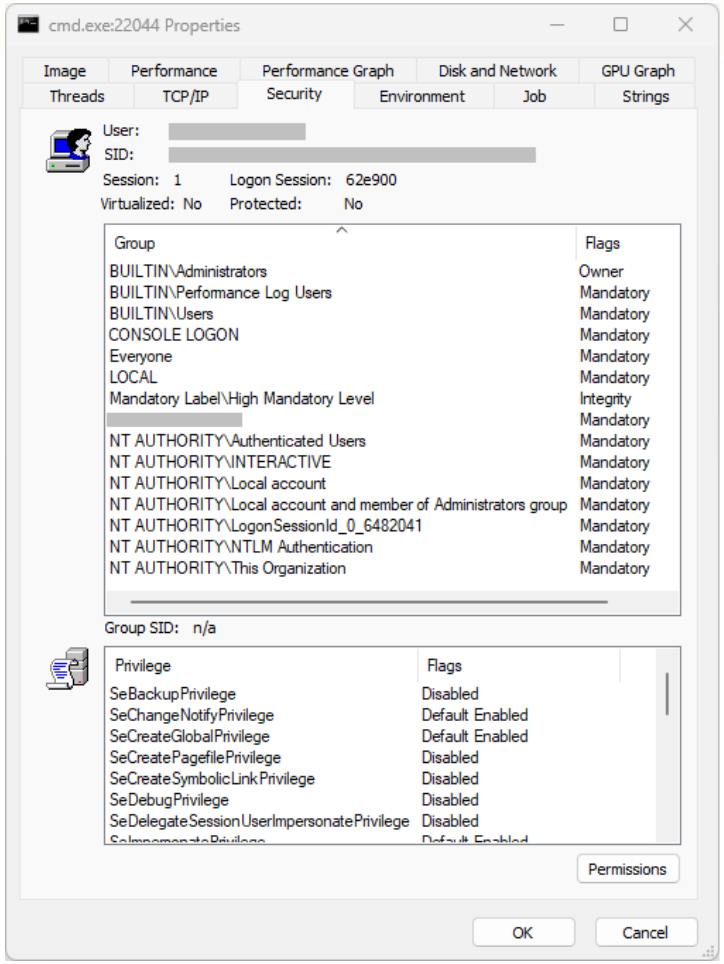
- Users
  - Medium: standard users
  - High: elevated users
- Process integrity level
  - The minimum associated to the owner and the executable file
  - User processes usually are Medium or High
    - Except if executing Low-labeled executables
  - Service processes: High

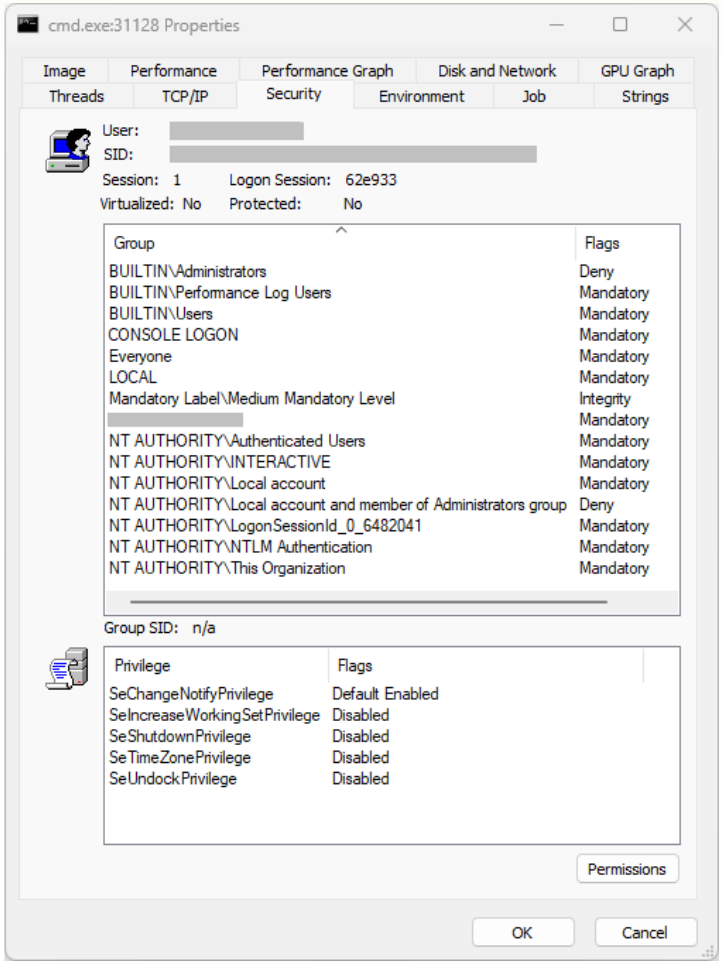
Windows mandatory integrity control

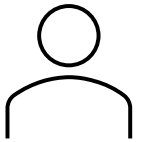
- Securable objects mandatory label
  - NO_WRITE_UP (default)
  - NO_READ_UP
  - NO_EXECUTE_UP

Windows mandatory integrity control

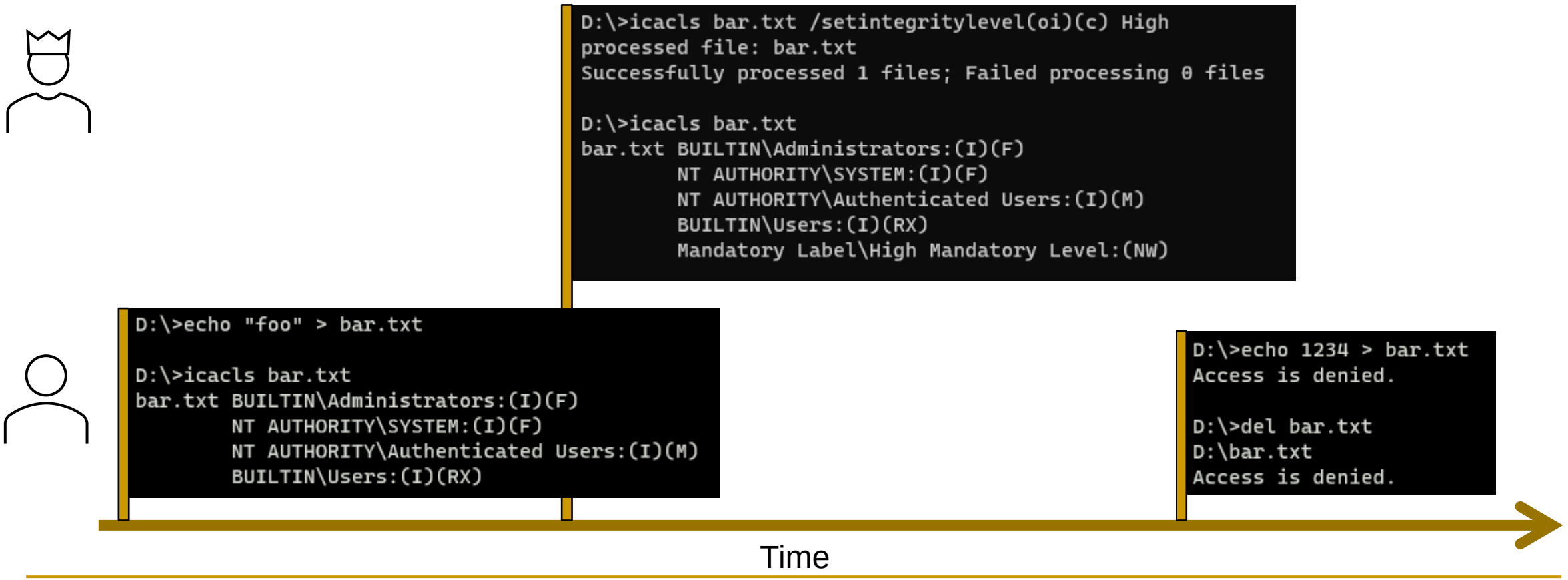








Windows mandatory integrity control



Clark-Wilson Integrity Model

D. D. Clark, D. R. Wilson, "A Comparison of Commercial and Military Computer Security Policies", IEEE Symposium on Security and Privacy, 1987

- Addresses information integrity control
  - Uses the notion of transactional data transformations
  - Separation of duty: transaction certifiers ≠ implementers
- Terminology
  - Data items
    - Constrained Data Item (**CDI**)
      - Can only be manipulated by TPs
    - Unconstrained Data Item (**UDI**)
  - Integrity policy procedures
    - Integrity Verification Procedure (**IVP**)
      - Ensures that all CDIs conform with the integrity specification
    - Transformation Procedure (**TP**)
      - Well-formed transaction
        - Take as input a CDI or a UDI and produce a CDI
      - Must guarantee (via certification) that transforms all possible UDI values to "safe" CDI values

Clark-Wilson Integrity Model: Certification & Enforcement

- Integrity assurance
  - Certification
    - Relatively to the integrity policy
  - Enforcement
- Two sets of rules
  - Certification Rules (C)
  - Enforcement Rules (E)

Clark-Wilson Integrity Model: Certification & Enforcement rules

- Basic rules
  - **C1**: when an IVP is executed, it must ensure that all CDIs are valid
  - **C2**: for some associated set of CDIs, a TP must transform those CDIs from one valid state to another
  - **E1**: the system must maintain a list of certified relations and ensure only TPs certified to run on a CDI change that CDI
- Separation of duty (external consistency)
  - **E2**: the system must associate a user with each TP and set of CDIs. The TP may access CDIs on behalf of the user if authorized
  - **C3**: allowed user-TP-CDI relations must meet "separation of duty" requirements
- Identification gathering
  - **E3**: the system must authenticate every user attempting a TP (on each attempt)
- Audit trail
  - **C4:** all TPs must append to a log enough information to reconstruct operations
- UDI processing
  - **C5**: a TP taking a UDI as input may only perform valid transactions for all possible values of the UDI. The TP will either accept (convert to CDI) or reject the UDI
- Certification constraints
  - **E4**: only the certifier of a TP may change the associated list of entities