Authorization

Segurança Informática em Redes e Sistemas
2022/23

Ricardo Chaves

Ack: Miguel Pardal, Carlos Ribeiro, Miguel P. Correia

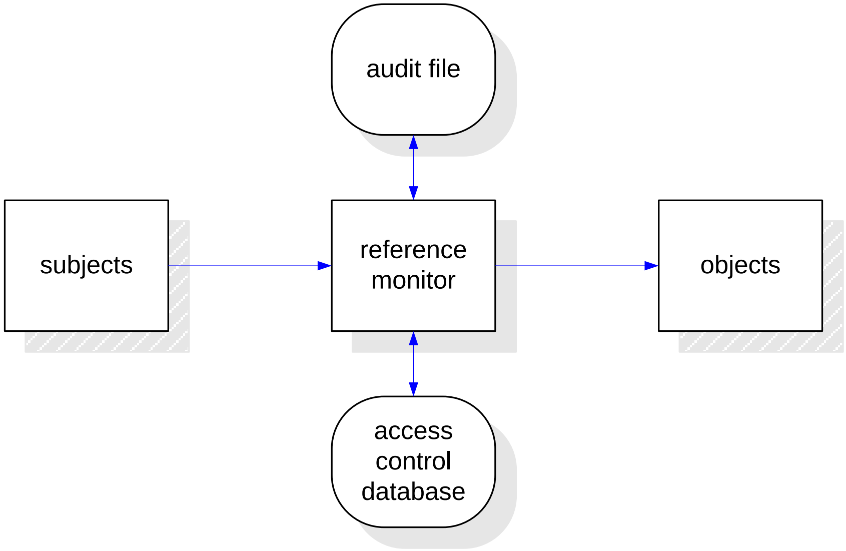
Roadmap

- Authorization / access control
- Access control models

Roadmap

- **Authorization / access control**
- Access control models

Access control



- Subjects
  - People, processes
  - Active entities
- Objects
  - Files, processes
  - Passive entities
- Accesses
  - Operations on objects

- **Reference Monitor**
  - Small and verifiable
  - Total mediation

Access actions

- There are several options; they can be:
  - **Specific**
    - Each object has a set of specific operations
    - Large policy size ☹
  - **Generic**
    - Only write and read (change or not the object’s state)
  - **Mixed**
    - (next slide)

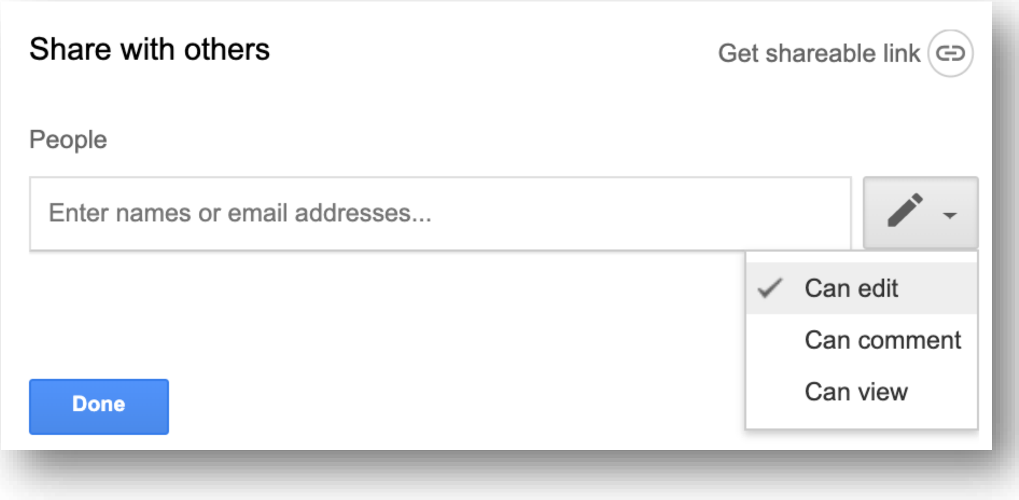
Mixed access actions

- **Mixed**
  - Bell LaPadula (BLP): execute, append, read, write
  - Different objects, different meanings; example from Unix:

Operation	Meaning for directories	Meaning for files
Read	List content	Read content
Write	Create / rename	Create / rename
Execute	Enter (cd) and access files/dirs	Run program in the file

Specific access actions example: Google Drive

- Actions: edit, comment, view



Share with others

Get shareable link

People

Enter names or email addresses...

✓ Can edit

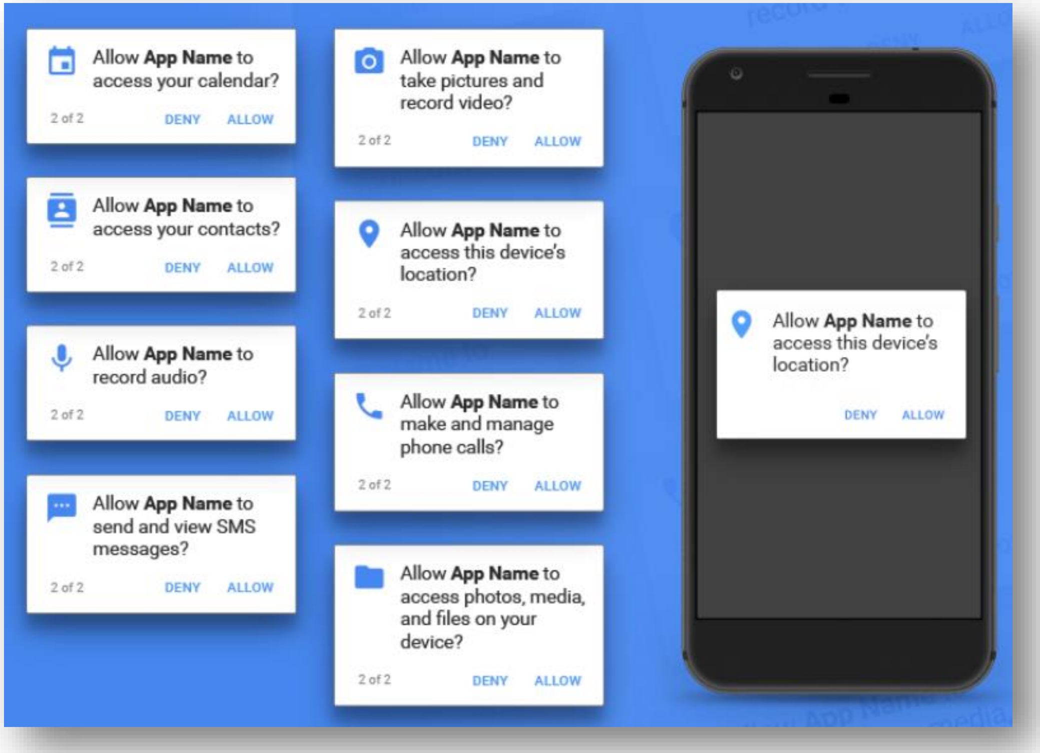
Can comment

Can view

Done

Specific access actions example: Android

- Actions: access, record, send/view, ...
- Many objects: calendar, contacts, audio, SMSs, location,...



Allow **App Name** to access your calendar?

Allow **App Name** to access your contacts?

Allow **App Name** to record audio?

Allow **App Name** to send and view SMS messages?

Allow **App Name** to take pictures and record video?

Allow **App Name** to access this device's location?

Allow **App Name** to make and manage phone calls?

Allow **App Name** to access photos, media, and files on your device?

Allow **App Name** to access this device's location?

Management and Ownership

- Owner
  - Usually the creator of the object
- **Discretionary access control systems (DAC)**
  - Management made by the owner
- **Mandatory access control systems (MAC)**
  - Management made by a global policy
  - Less susceptible to malware

Caution: unrelated to Message Authentication Codes, Medium Access Control

Roadmap

- Authorization / access control
- **Access control models**

Model goals

Trustworthiness

Expressivity

Performance

Administration

Discretionary Access Control

Access matrix

- **Access control matrix**
  - Theoretical model, very sparse
- **Access Control List (ACL)**
  - Matrix columns
- **Capability**
  - Matrix rows

	Obj 1	Obj 2	Obj 3
Suj 1	{read}		{read, write}
Suj 2		{read, write}	
Suj 3	{read,write}		

Access Control List

- **One ACL for each object**
  - ACL – non-empty element of each column in the matrix
  - ACE (Access Control Entry) – a cell of the matrix
- Can be assigned to **groups of subjects**
  - Minimize policy
  - Negative permissions may be required
- Hard to manage individual subject permissions

Capabilities

- **One list of capabilities for each subject**
  - Each capability is a non-empty element in a matrix row
- It is hard to:
  - Know who can access an object
  - Revoke a capability
- Use:
  - Traditionally less used than ACLs
  - In distributed systems
  - Example: access document with link (URL)
    - Whoever knows the link, can access the object

Roadmap

- Authorization / access control
- **Access control models**
  - **Role-Based Access Control**

Role-Based Access Control (RBAC)

- Performs access control based on **roles**
- Allows the description of complex policies
  - Segregation of duties (static and dynamic)
    - Static: allows role memberships that are mutually exclusive
    - Dynamic: allows same subject having 2 roles but not using both in same operation
  - Least privilege:
    - possible to assign the least privileges the subject needs to the role
  - Delegation:
    - possible to transfer privileges
  - Restrictions based on: time, context, history

RBAC mechanism

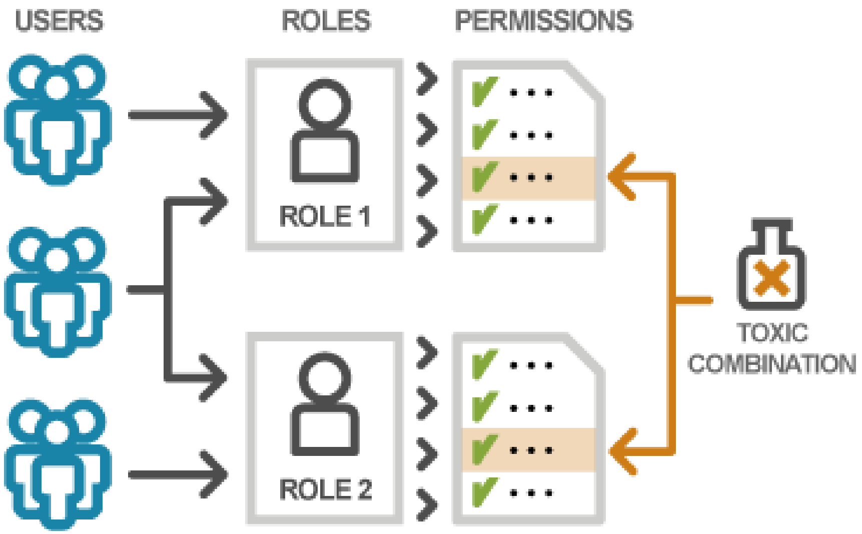
- Users are associated with roles
- Roles are associated with permissions
  - A user has a permission if he has an association with a role that has an association with a permission
- Reduces size of policy
  - Better scalability
  - Reduces error probability
  - Simplifies administration

RBAC administration benefits

- Associations between roles and permissions are more stable than associations between users and permissions
  - So easier to administrate: just associate users to roles
- May be associated with different types of concept
  - Position
  - Authority
  - Skill
  - Responsibility
- Allows propagating rights across hierarchies
  - Similar to inheritance in object oriented programming

RBAC overall assessment

- RBAC models categorize users based on similar needs and groups them into roles
  - The role concept uses approximations for the sake of simplicity
  - There is a never-ending struggle to refine the definition of a role
    - and to maintain a sound segregation of duties

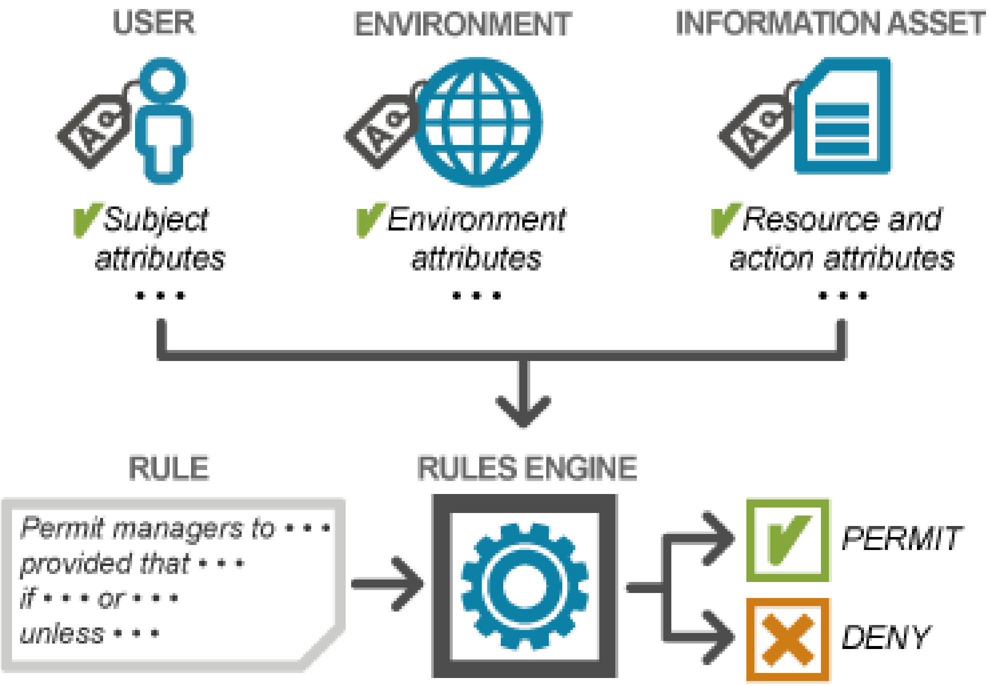


Roadmap

- Authorization / access control
- **Access control models**
  - **Attribute Based Access Control**

Attribute-Based Access Control (ABAC)

- Permissions are granted or denied depending on the values of named attributes



ABAC assessment

- **Dynamic permissions**
  - Current business rules
  - Risk mitigating precautions
  - Context-related security measures
- **Fine-grained authorization**
- Major disadvantages:
  - Complex model – big attack surfaces
  - Lower performance (higher delay)

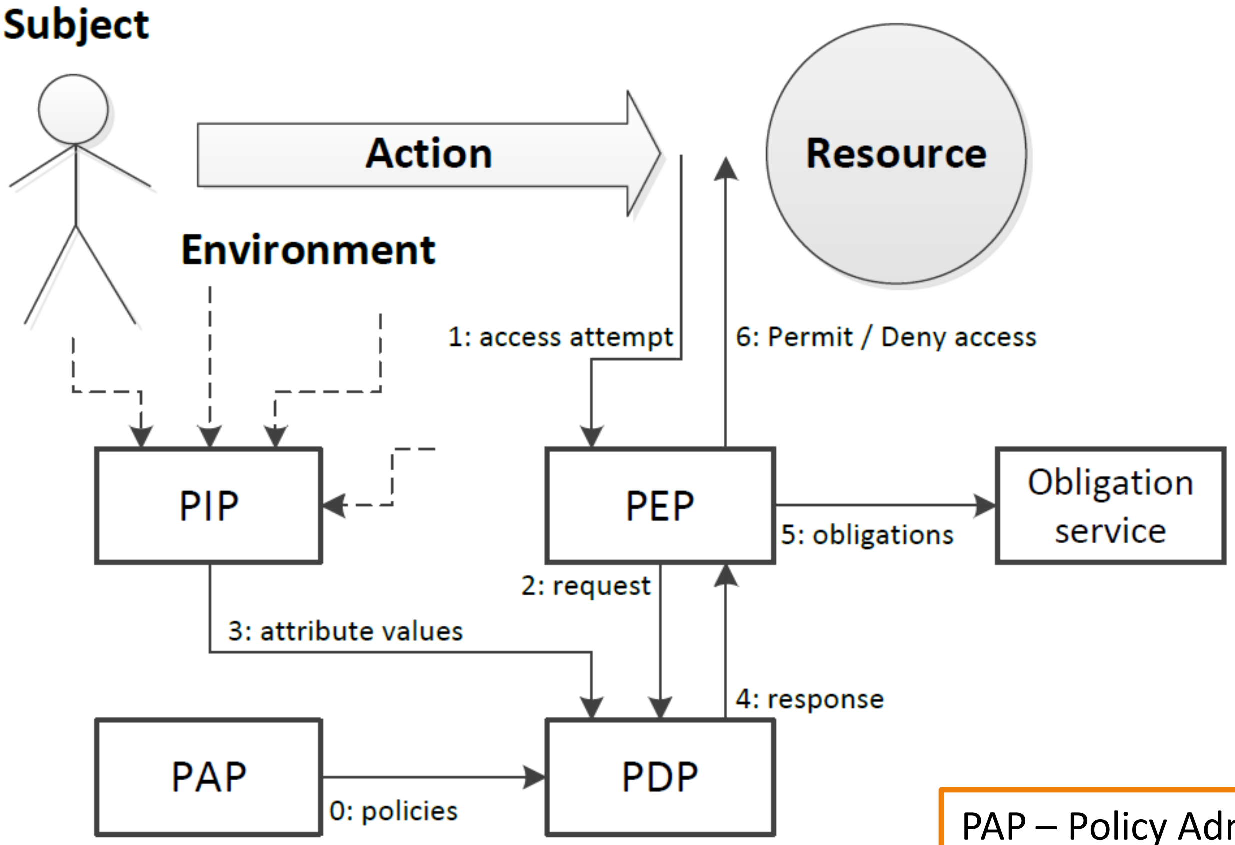
XACML

- **eXtensible Access Control Markup Language**
- Language that allows implementing ABAC
- Standard proposed by OASIS
  - Processing model
  - Policy format
  - Request/response formats

XACML Processing Model

- **Components**
  - PAP – Policy Administration Point
  - PEP – Policy Enforcement Point
  - PDP – Policy Decision Point
  - PIP – Policy Information Point
- **Obligations**
  - Actions that must be executed when the request is processed
    - Typically used to write audit logs

XACML processing model



PAP – Policy Administration Point
PEP – Policy Enforcement Point
PDP – Policy Decision Point
PIP – Policy Information Point

Roadmap

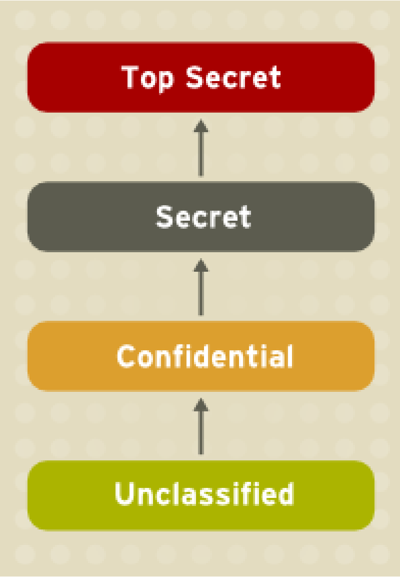
- Authorization / access control
- **Access control models**
  - **Bell-LaPadula model**

Bell-LaPadula (BLP) model

- Mandatory Access Control (MAC)
- State machine

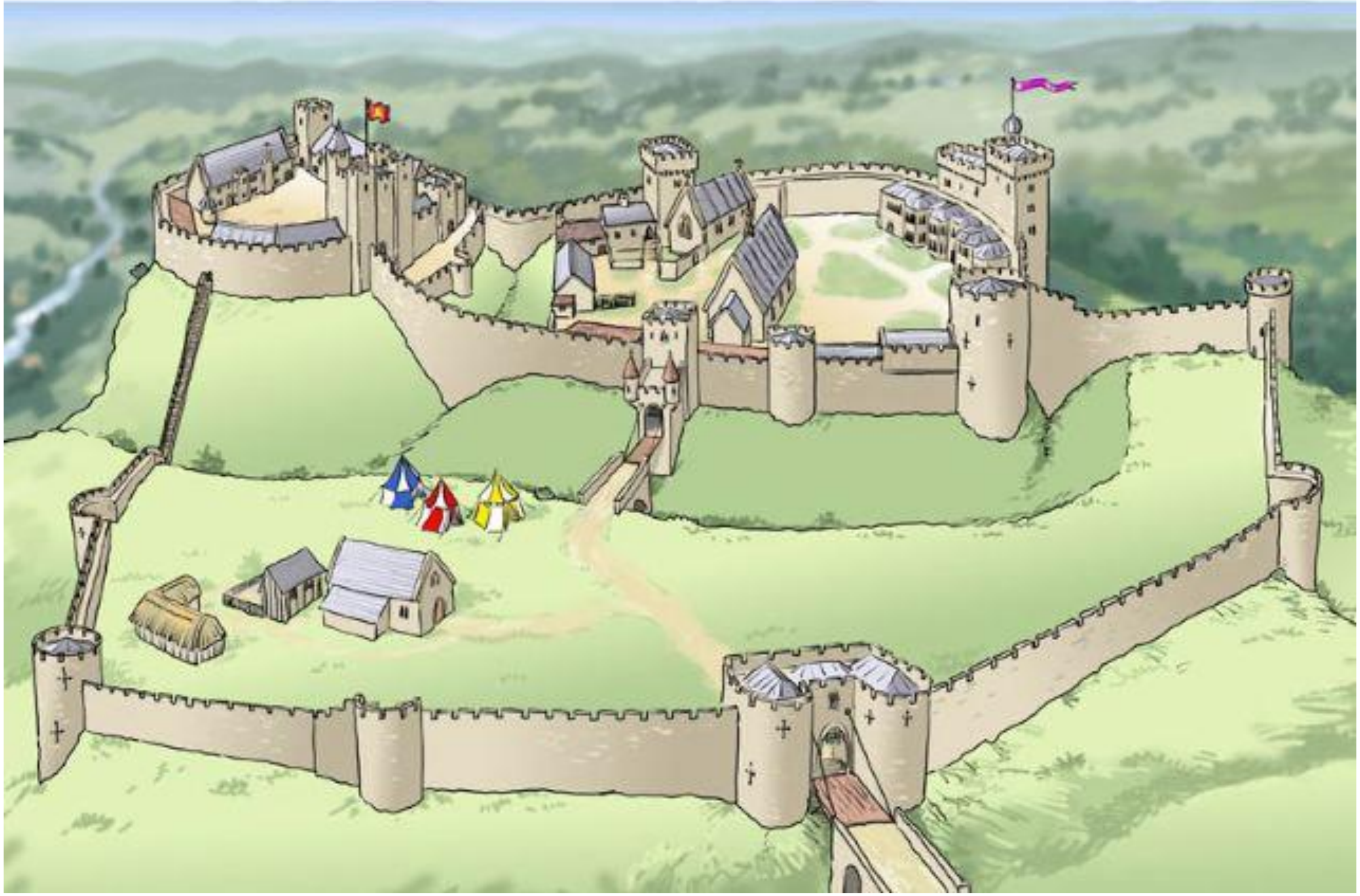
  ls *– clearance of* ***<u>subject</u>*** *s*

  lo *– classification of* ***<u>object</u>*** *o*



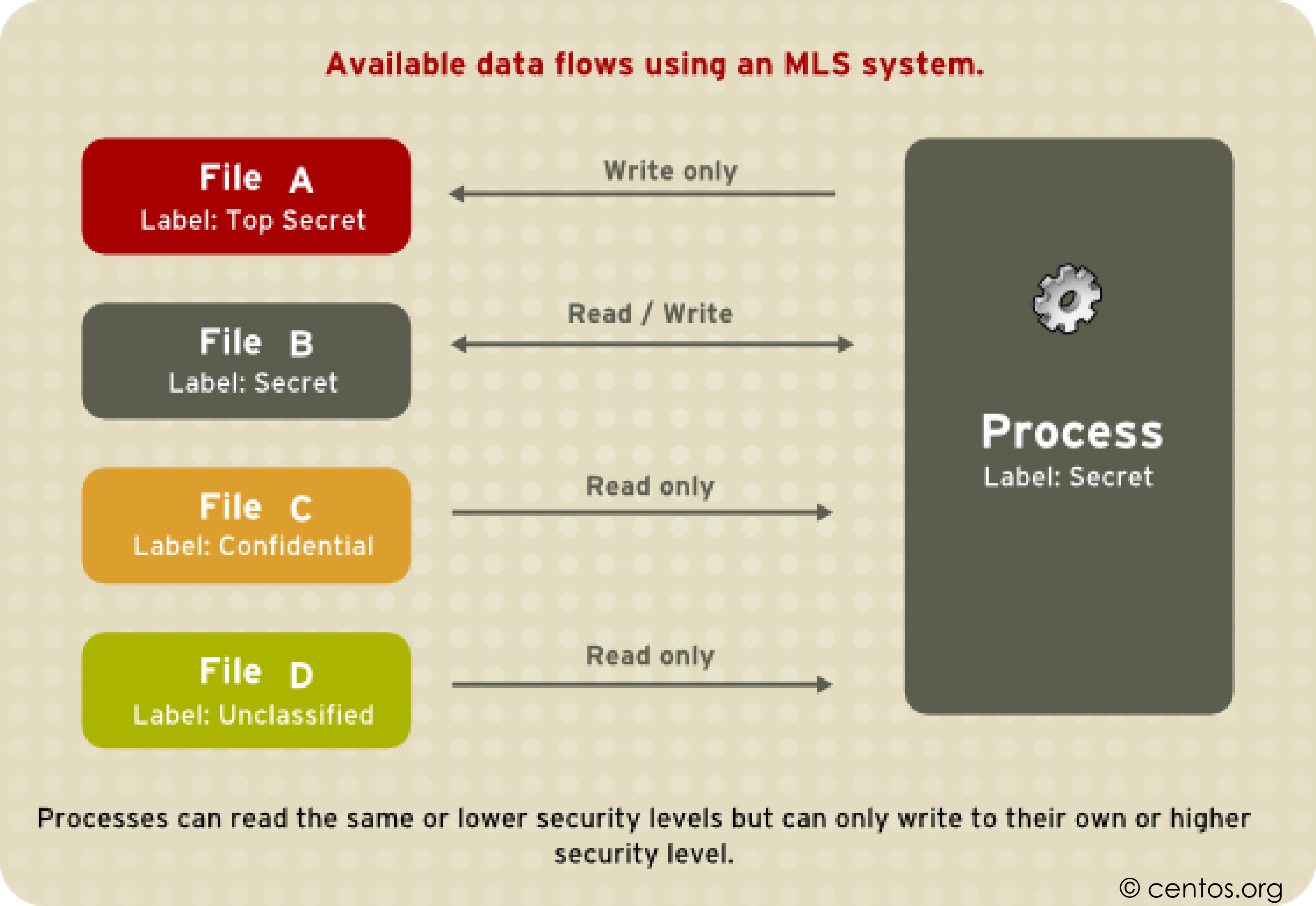
- Simple Security Property
  - **No read up:** access granted iff *lo <= ls*
- Confinement Property (★-Property)
  - **No write down:** access granted iff *ls <= lo*

Bell-LaPadula (BLP)



© selfless security

BLP: file access example



Processes can read the same or lower security levels but can only write to their own or higher security level.

© centos.org

Bell-LaPadula (BLP) overall assessment

- Confidentiality only
  - No integrity
- Data changes only through specific programs
- There are covert channels
  - e.g. file names
- Does not allow management (rights are fixed)
  - Nor delegation

Summary

- Authorization / access control
- Access control models