Information System - Role Based Access Control (RBAC)

Dr J Paul Gibson

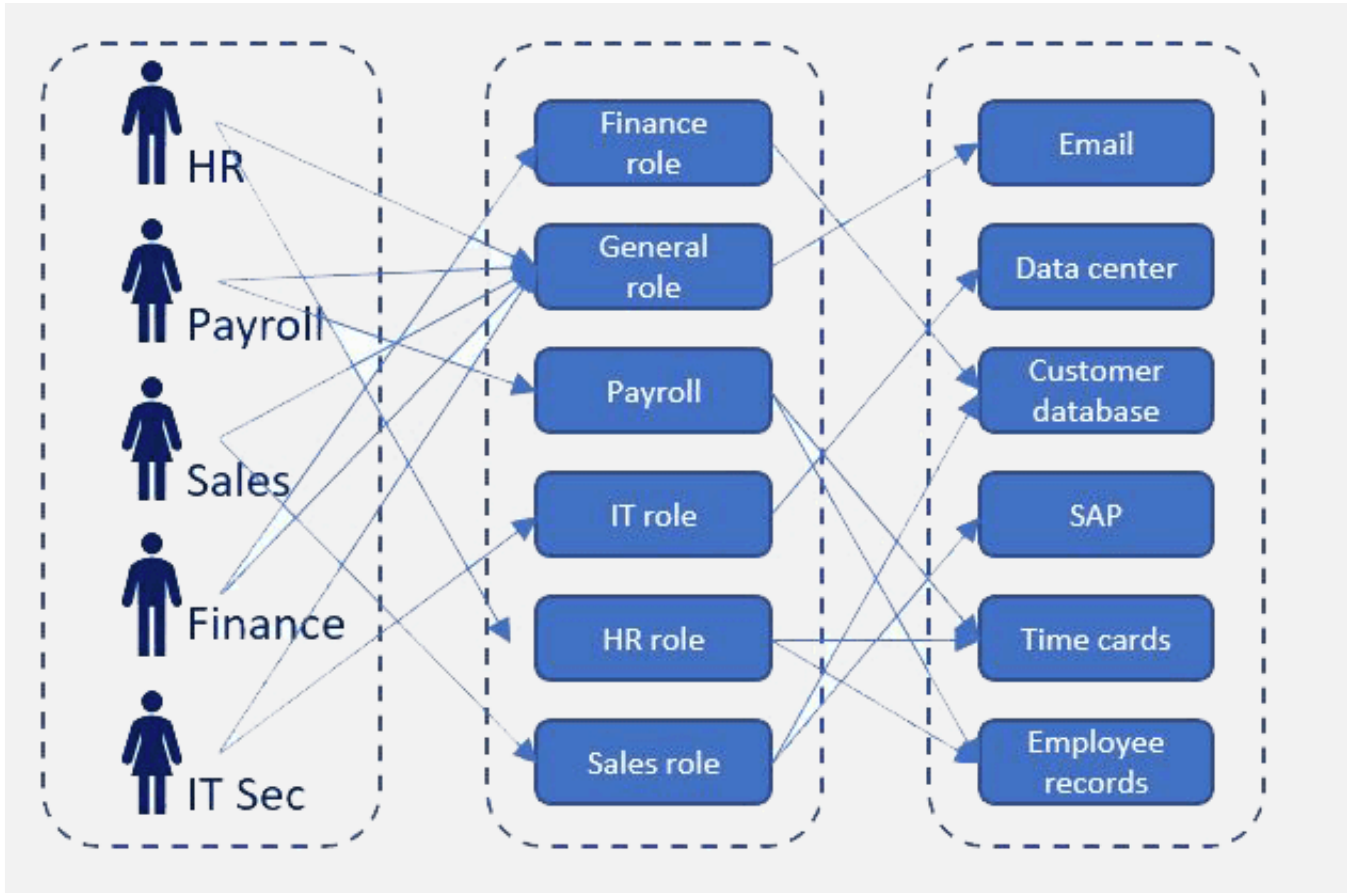
Dept. INF

Office D311

paul.gibson@telecom-sudparis.eu

http://jpaulgibson.synology.me/~jpaulgibson/TSP/Teaching/CSC4104/CSC4104-InformationSystems-RBAC.pdf

RBAC - **USERS** **ROLES** and **OPERATIONS**



https://www.bettercloud.com/monitor/the-fundamentals-of-role-based-access-control/

Role-based access control (RBAC) is a widely used security framework claimed to be especially appropriate for commercial settings.

Unlike access control policies that assign permissions to subjects, RBAC associates permissions with functions/jobs/roles within an organization.

A *role* is a collection of job functions. Roles within a bank might include: `president`, `manager`, `trainer`, `teller`, `auditor`, `janitor`, etc.

The following are the three primary RBAC rules:

- **Role assignment:** A subject can execute a transaction only if the subject has an active role.
- **Role authorization:** A subject's active role must be an authorized role for that subject.
- **Transaction authorization:** A subject can execute a transaction only if the transaction is authorized for one of the subject's active roles.

Note that a subject can have multiple roles. For example, in a pinch a bank president might also act as a teller.

One role may *subsume* another, meaning that anyone having role r_j can do at least the functions of r_i.

Example: a `trainer` can perform all of the actions of a `trainee`, as well as some others.

RBAC can also model *separation of duty* (one individual cannot assume both roles r_1 and r_2).

Example: if `teller` is among S's authorized roles, `auditor` cannot be.

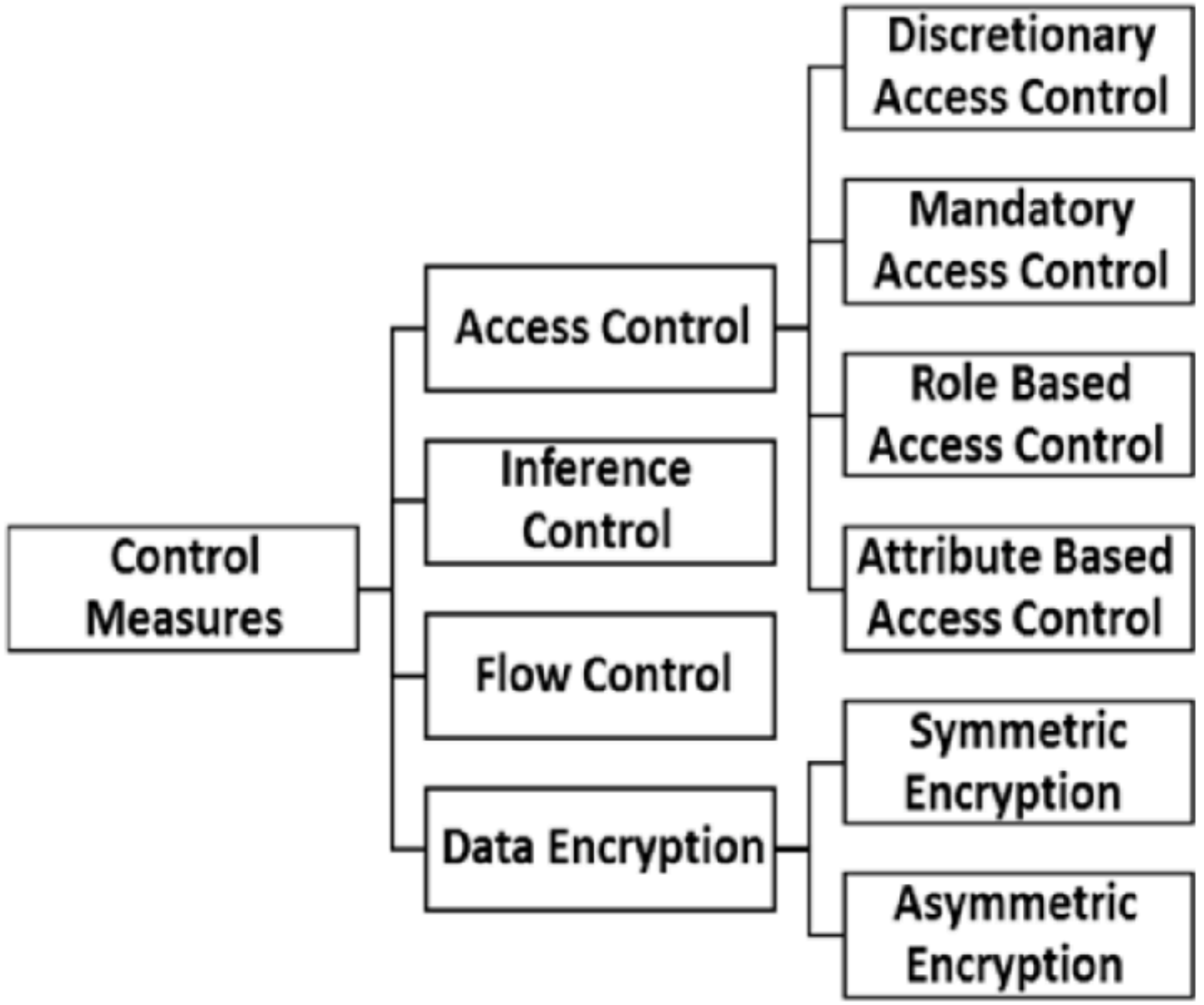
RBAC is generally more flexible than standard access control policies:

- RBAC is easy to administer. Everyone in role `teller` has the same permissions.
- Permissions are appropriate to the organization—"open an account" rather than "read a file."
- RBAC recognizes that a subject often has various functions within the organization.
- RBAC allows a subject to transition between roles without having to change identities.

- RBAC associates access permissions with a job/function/role rather than with individual subjects.
- This provides a flexible approach to modeling the dynamism of commercial organizations.

Further Reading

A Survey of Access Control and Data Encryption for Database Security, Emad F. Khalaf and Mustafa M. Kadi

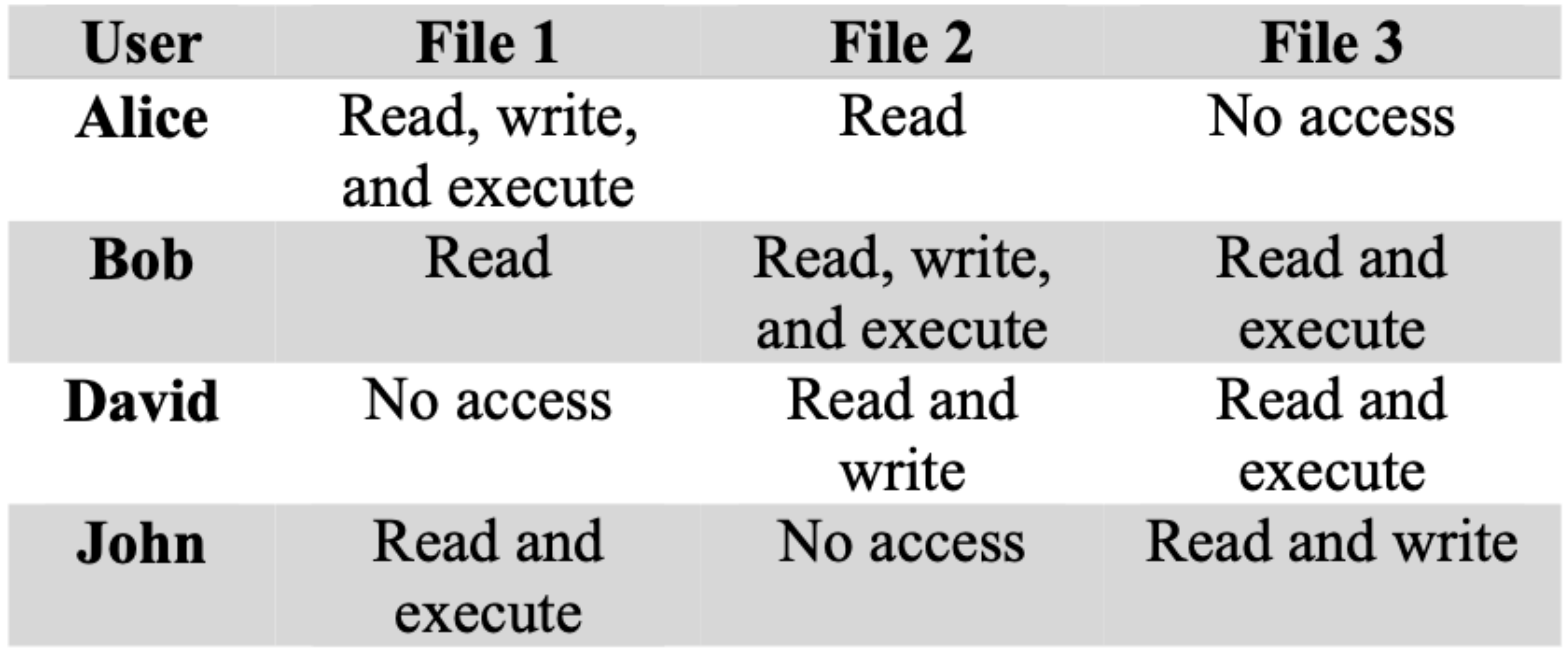


The recommended model for our study

Sometimes called rule-based

Discretionary Access Control Policy Example

Table 1. Example of access matrix.



User	File 1	File 2	File 3
Alice	Read, write, and execute	Read	No access
Bob	Read	Read, write, and execute	Read and execute
David	No access	Read and write	Read and execute
John	Read and execute	No access	Read and write

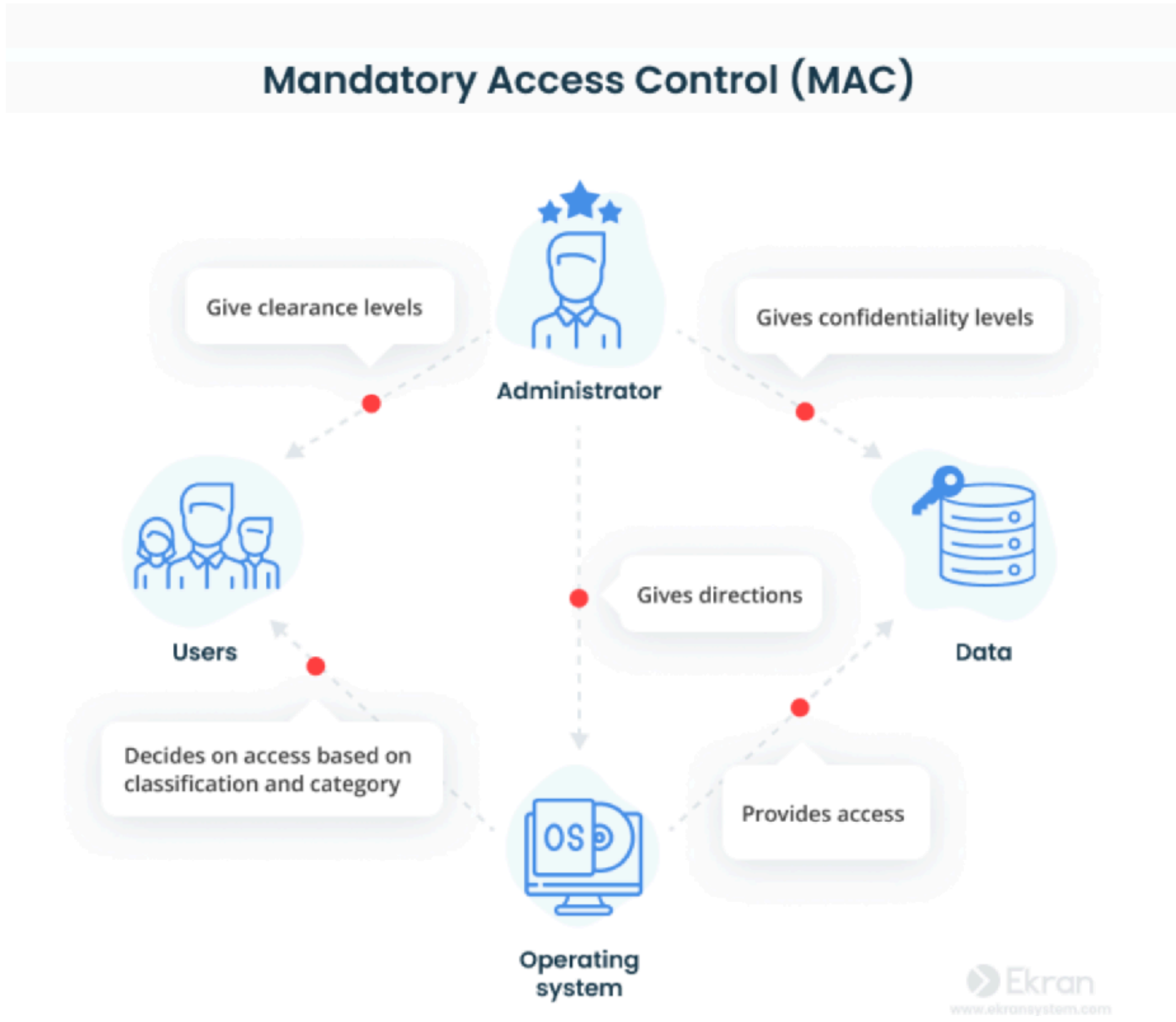
Simplest approach - owners of data control access

When you can’t trust the owners

In a MAC system, access to resources is determined by centrally-defined rules that users cannot override. Access to resources is strictly controlled and cannot be changed by individual users.

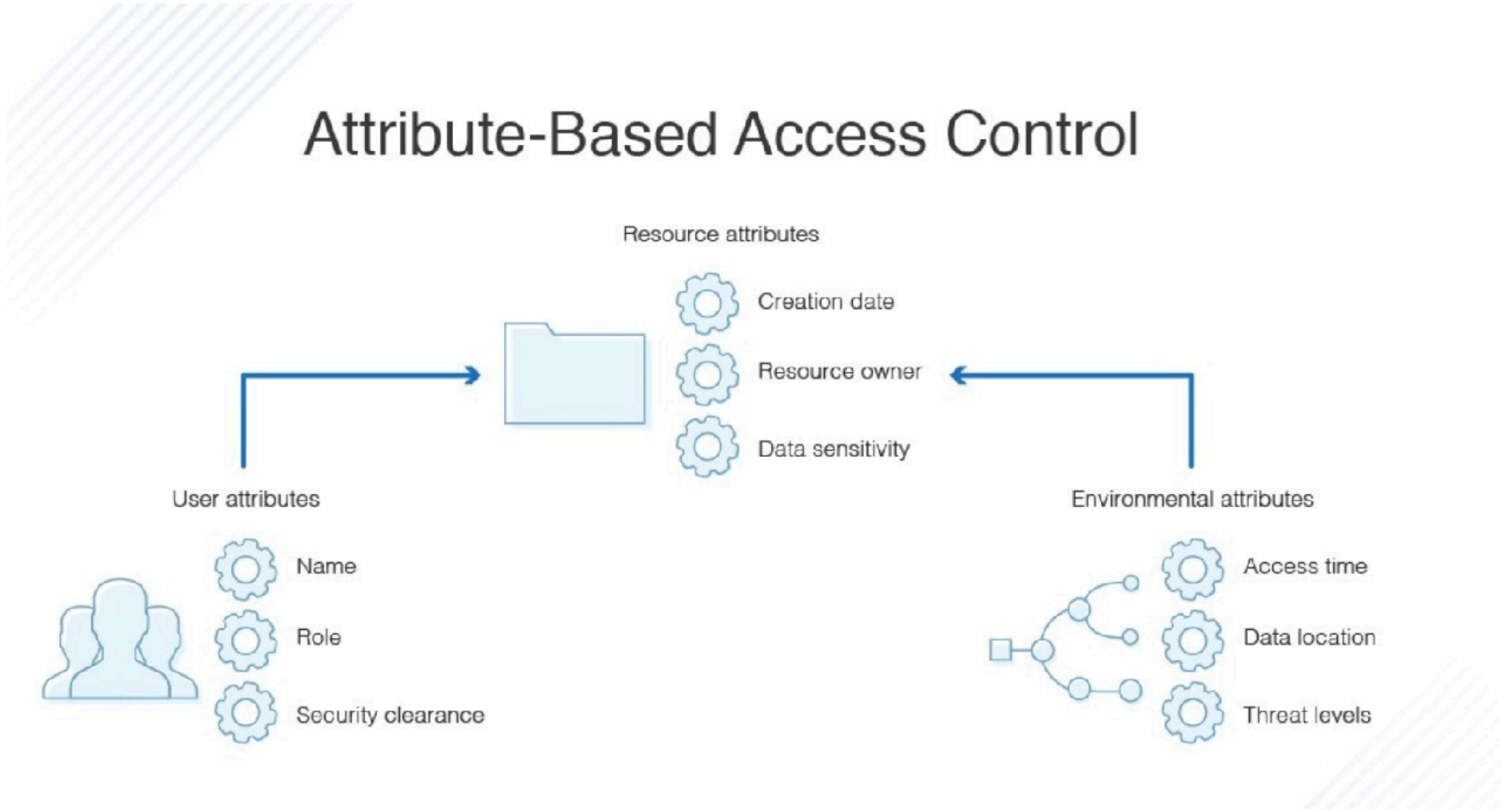
On the other hand, DAC is a security model where the resource owner determines its access. In a DAC system, owners can control who has access to their resources and their access level.

One key difference between MAC and DAC is that MAC is considered a more secure model because individual users cannot change access to resources.


Mandatory Access Control (MAC)
Give clearance levels
Administrator
Gives confidentiality levels
Users
Gives directions
Data
Decides on access based on classification and category
Provides access
Operating system
Ekran

https://www.ekransystem.com/en/blog/mac-vs-dac

Rules and/or attributes

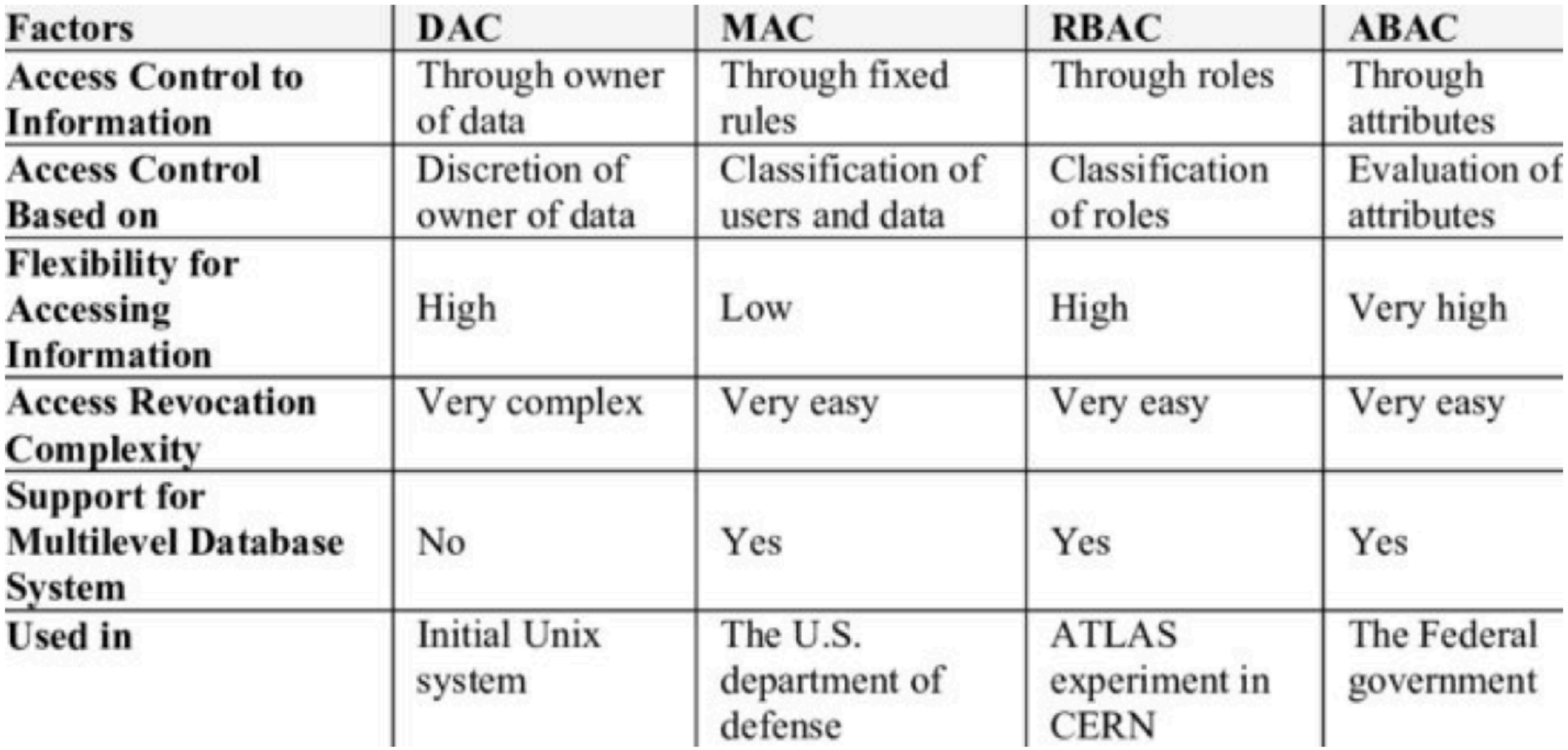


The most powerful, but also the most complex

It is usually dynamic rather than static

https://www.dnsstuff.com/rbac-vs-abac-access-control

Summary



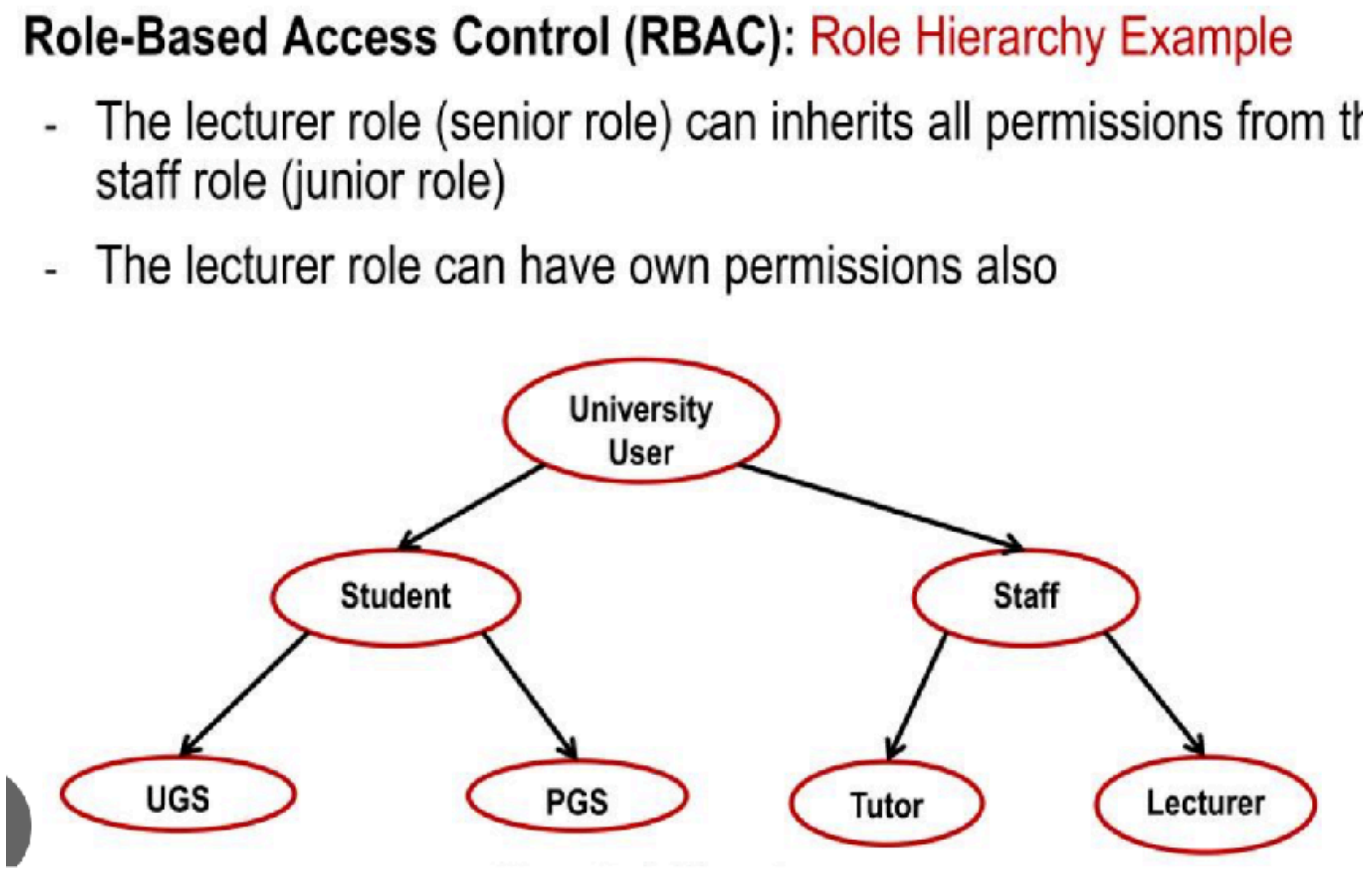
Factors	DAC	MAC	RBAC	ABAC
Access Control to Information	Through owner of data	Through fixed rules	Through roles	Through attributes
Access Control Based on	Discretion of owner of data	Classification of users and data	Classification of roles	Evaluation of attributes
Flexibility for Accessing Information	High	Low	High	Very high
Access Revocation Complexity	Very complex	Very easy	Very easy	Very easy
Support for Multilevel Database System	No	Yes	Yes	Yes
Used in	Initial Unix system	The U.S. department of defense	ATLAS experiment in CERN	The Federal government

Role Based Access Control (RBAC)

The suggested model for your case study

TO DO - Model the access control requirements for the case study

Maybe use your inheritance hierarchy from your use cases? eg:



Role-Based Access Control (RBAC): Role Hierarchy Example

- The lecturer role (senior role) can inherits all permissions from tl staff role (junior role)
- The lecturer role can have own permissions also

$$\begin{tikzcd}
 & & \text{University User} \arrow[dll] \arrow[drr] & & \\
\text{Student} \arrow[d] \arrow[dr] & & & & \text{Staff} \arrow[dl] \arrow[d] \\
\text{UGS} & \text{PGS} & & \text{Tutor} & \text{Lecturer}
\end{tikzcd}$$