# Access Control

Vítor Francisco Fonte, vff@di.uminho.pt, University of Minho, 2025

## Overview

- Access Control Principles
- Subjects, Objects and Access Rights
- Discretionary Access Control
- Role-based Access Control
- Attribute-based Access Control

## Definitions of Access Control

- NISTIR 7298: The process of granting or denying specific requests:
  - to obtain and use information and related information processing services;
  - and to enter specific physical facilities.
- RFC 4949: The process by which:
  - use of system resources is regulated according to a security policy;
  - and is permitted only by authorised entities (users, programs, processes, or other systems) according to that policy.

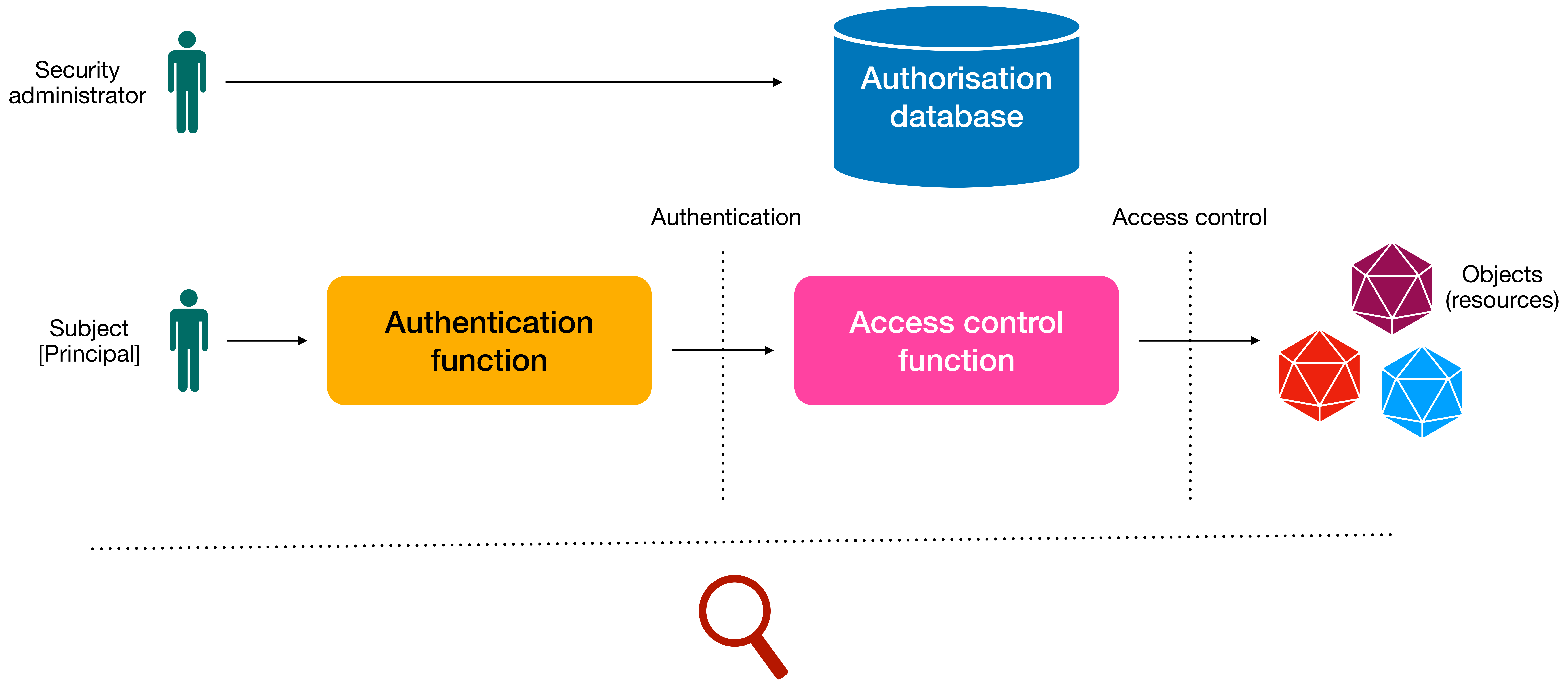
## Our focus regarding Access Control

- Central element to the principal objectives of computer security:
    - to prevent unauthorised users from gaining access to resources;
    - to prevent legitimate users from accessing resources in an unauthorised manner;
    - and to enable legitimate users to access resources in an authorised manner.
- Access control implements a security policy that specifies who or what (e.g., in the case of a process) may have access to each specific system resource, and the type of access that is permitted in each instance.

## Access Control Context

- **Authentication:** Verification that the credentials of a user or other system entity are valid.
- **Authorisation:** The granting of a right or permission to a system entity to access a system resource. This function determines who is trusted for a given purpose.
- **Audit:** An independent review and examination of system records and activities in order to test for adequacy of system controls, to ensure compliance with established policy and operational procedures, to detect breaches in security, and to recommend any indicated changes in control, policy, and procedures.

# Overview of the Basic Access Control Model



## Access Control Policies

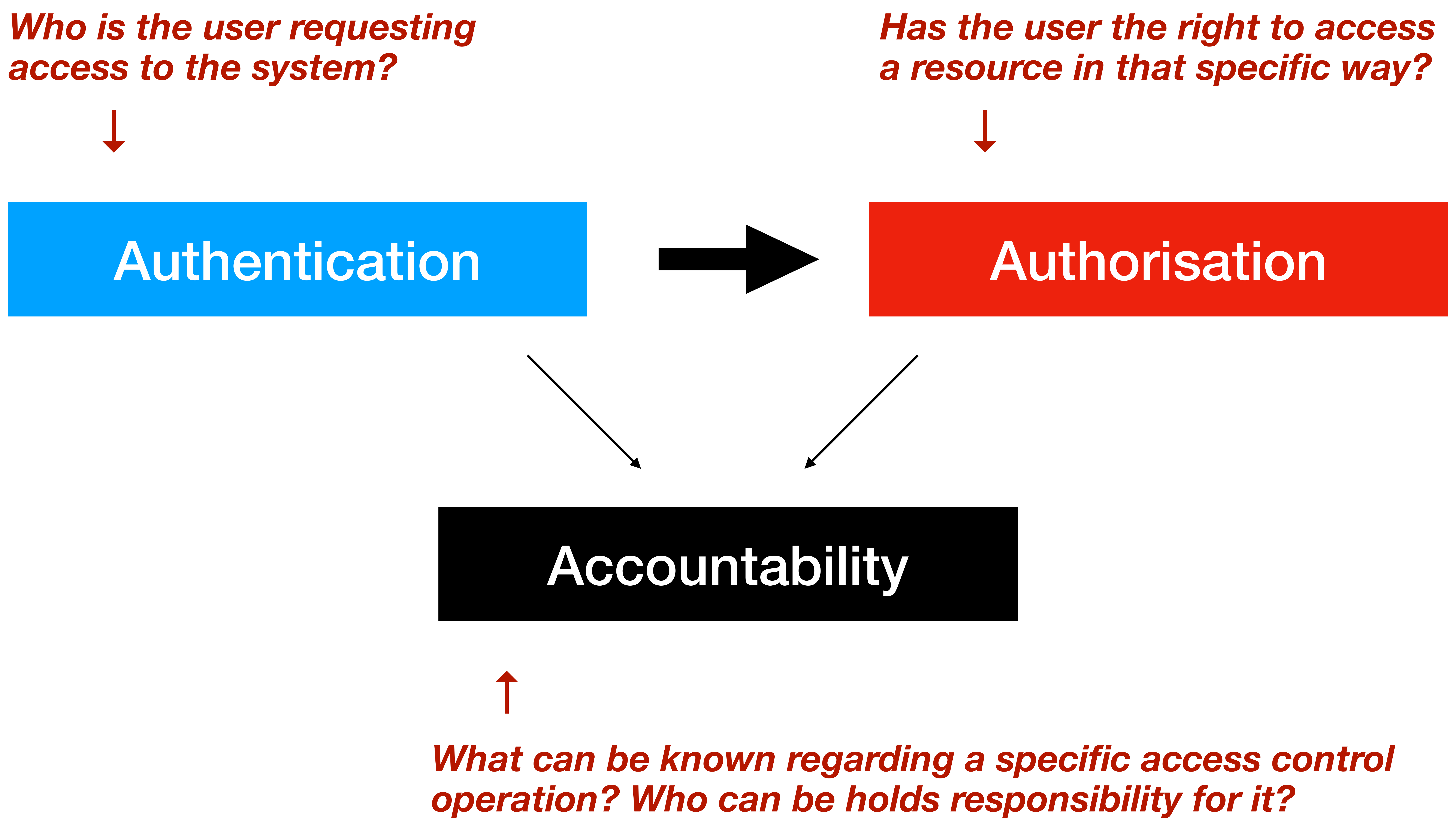
- **Discretionary access control (DAC):** Controls access based on the identity of the requestor and on access rules (authorisations) stating what requestors are (or are not) allowed to do. This policy is termed discretionary because an entity might have access rights that permit the entity, by its own volition, to enable another entity to access some resource.
- **Mandatory access control (MAC):** Controls access based on comparing security labels (which indicate how sensitive or critical system resources are) with security clearances (which indicate system entities are eligible to access certain resources). This policy is termed mandatory because an entity that has clearance to access a resource may not, just by its own volition, enable another entity to access that resource.

## Access Control Policies (cont.)

- **Role-based access control (RBAC):** Controls access based on the roles that users have within the system and on rules stating what accesses are allowed to users in given roles.

- **Attribute-based access control (ABAC):** Controls access based on attributes of the user, the resource to be accessed, and current environmental conditions.

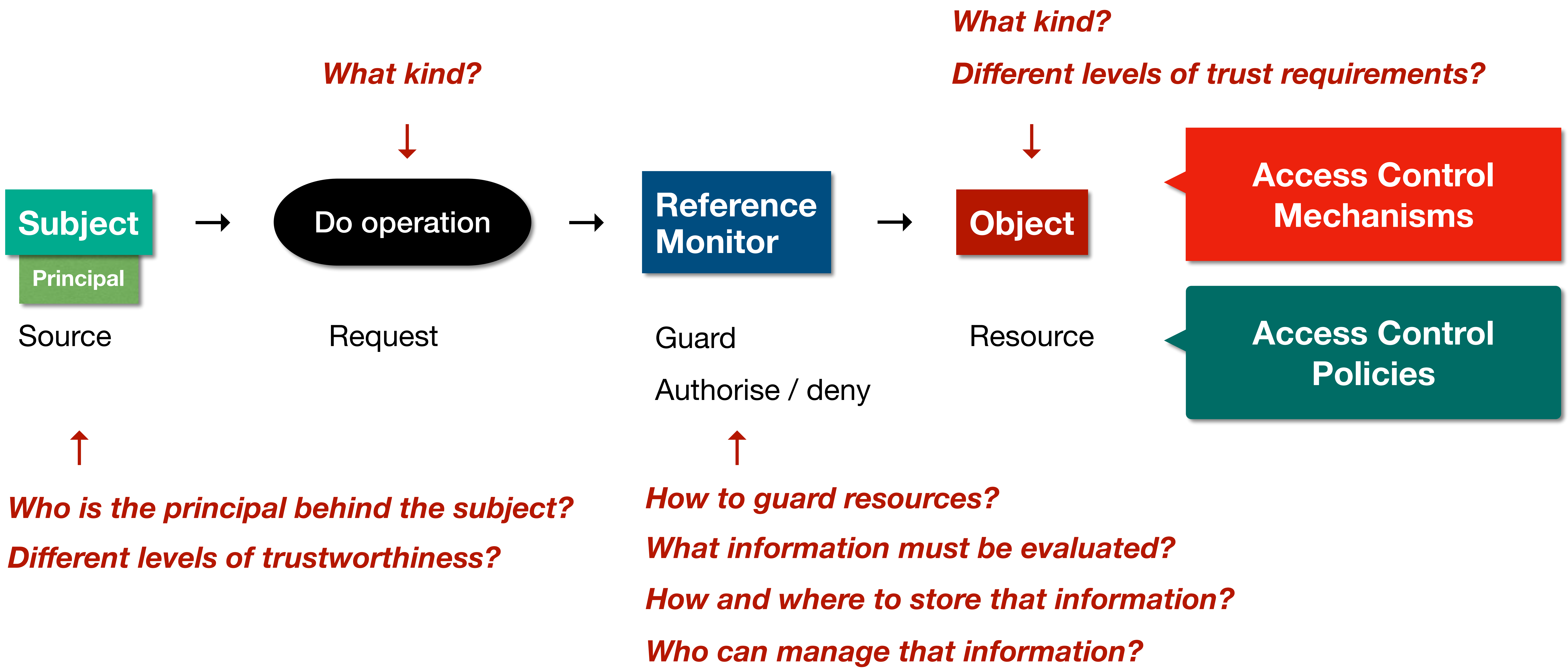
# Fundamental Model of Access Control



*Who is the user requesting access to the system?*
*Has the user the right to access a resource in that specific way?*
Authentication
Authorisation
Accountability
*What can be known regarding a specific access control operation? Who can be holds responsibility for it?*

# Fundamental Model of Access Control

## Authorisation


What kind?
Different levels of trust requirements?
What kind?
Subject
Principal
Do operation
Reference Monitor
Object
Access Control Mechanisms
Access Control Policies
Source
Request
Guard
Authorise / deny
Resource
Who is the principal behind the subject?
Different levels of trustworthiness?
How to guard resources?
What information must be evaluated?
How and where to store that information?
Who can manage that information?

## Subject

- A **subject** is an entity capable of accessing objects.
  - Typically is a process executing on behalf of a user (human or virtual).
  - A subject may be held accountable for the actions it has initiated.
  - An audit trail may be used to record the association of a subject with security-relevant action performed on an object.

## Classes of subject

- Basic access control systems typically define three classes of subject, with different access rights for each class:
  - **Owner:** This may be the creator of a resource, such as a file. For system resources, ownership may belong to a system administrator. For project resources, a project leader may be assigned ownership.
  - **Group:** In addition to the privileges assigned to an owner, a named group of users may also be granted access rights, such that membership in the group is sufficient to exercise these access rights. In most schemes, a user may belong to multiple groups.
  - **World:** The least amount of access is granted to users who are able to access the system but are not included in the categories owner and group for this resource.

## Object and types of objects

- An **object** is a resource to which access is controlled.
  - In general, an object is an entity used to contain and/or receive information. Examples include records, blocks, pages, segments, files, portions of files, directories, directory trees, mailboxes, messages, and programs.
  - Can also encompass, bits, bytes, words, processors, communication ports, clocks, and network nodes.
- The number and types of objects to be protected by an access control system depends on:
  - the environment in which access control operates;
  - and the desired tradeoff between security, complexity, and ease of use.

## Access Right

- An **access right** describes the way in which a subject may access an object, i.e. the action the subject can perform on a object. Some examples:
    - Read: User may view information in a system resource (e.g., a file, selected records in a file, selected fields within a record, or some combination). Read access includes the ability to copy or print.
    - Write: User may add, modify, or delete data in system resource (e.g., files, records, programs). Write access includes read access.
    - Execute: User may execute specified programs.
    - Delete: User may delete certain system resources, such as files or records.
    - Create: User may create new files, records, or fields.
    - Search: User may list the files in a directory or otherwise search the directory.

# Discretionary Access Control

- A discretionary access control scheme is one in which an entity may be granted access rights that permit the entity, by its own volition, to enable another entity to access some resource.
- **Access Control Matrix**
  - subjects (and groups of subjects) X objects (and groups of objects)
  - subjects can be users, devices, applications, …
  - since they are usually sparse they are often decomposed by:
    - columns, resulting in Access Control Lists (ACLs)
    - rows, resulting in Capability Tickets

## Access Control Lists (ACLs)

- The **Access Control List (ACL)** scheme decomposes the Access Control Matrix by column.
    - Each object specifies the access rights of a set of users (or groups of users).
- An ACL may contain a default entry setting access rights for subjects not explicitly listed.
    - Should always follow the rule of least privilege.
- The integrity of the ACL must be protected, and guaranteed (usually by the operating system).
    - OS keeps the ACL on a reserved area, only modifiable by the object owner.

## Capability Tickets

- The **Capability Ticket** scheme decomposes the Access Control Matrix by row.
  - A **capability ticket** specifies authorised objects and operations for a particular user.
- Each user has a number of tickets and may be authorised to loan or give them to others.
- Because tickets may be dispersed around the system, they present a greater security problem than access control lists.
- The integrity of the ticket must be protected, and guaranteed (usually by the operating system).
  - OS may hold all tickets on behalf of the subjects (on private area).
  - OS may include an unforgeable token (e.g. large password or a cryptographic message authentication code) and may be verified by the relevant resource whenever access is requested. Useful in a distributed environment.

## ACLs vs. Capability Tickets

- ACLs are convenient for determining which users have access rights to a particular object.
- Capability Tickets are convenient for determining the access rights available to a particular user.
- **Authorisation Table:** not sparse and more convenient than ACLs and Capability Tickets:
    - Each row: a subject, an access right, an object.
    - Sorting by object or subject equivalent to ACLs or Capability Tickets, respectively.
    - Easily implemented as relational database.

## Role-Based Access Control (RBAC)

- Traditional DAC systems define the access rights of individual users and groups of users.
- In contrast, RBAC is based on the roles that users assume in a system rather than the user's identity.
- Typically, RBAC models define a role as a job function within an organisation.
- RBAC systems assign access rights to roles instead of individual users.
- In turn, users are assigned to different roles, either statically or dynamically, according to their responsibilities.

## Role-Based Access Control (RBAC)

- Many to many relation for roles and users, and for roles and resources
  - A user can be assigned multiple roles
  - A role can be held by multiple users
  - A role can grant access rights to multiple resources
  - Access rights to a resource can be granted by multiple roles
- A role can be treated as an object, allowing the definition of role hierarchies

## RBAC Reference Models

- RBAC0: base model (minimum requirement for an RBAC system)
- RBAC1: base model + role hierarchies
- RBAC2: base model + constraints
- RBAC3: base model + role hierarchies + constraints

## RBAC Base Model

- **User:** An individual that has access to this computer system. Each individual has an associated user ID.
- **Role:** A named job function within the organisation that controls this computer system. Typically, associated with each role is a description of the authority and responsibility conferred on this role, and on any user who assumes this role.
- **Permission:** An approval of a particular mode of access to one or more objects. Equivalent terms are access right, privilege, and authorisation.
- **Session:** A mapping between a user and an activated subset of the set of roles to which the user is assigned.

## RBAC Base Model

- The many-to-many relationships between users and roles and between roles and permissions provide a flexibility and granularity of assignment not found in conventional DAC schemes.
  - Examples: Users may need to list directories and modify existing files without creating new files, or they may need to append records to a file without modifying existing records.
- Without this flexibility and granularity, there is a greater risk that a user may be granted more access to resources than is needed because of the limited control over the types of access that can be allowed.
  - Violation of the principle of least privilege.

## RBAC Role Hierarchies

- Role hierarchies provide a means of reflecting the hierarchical structure of roles in an organisation.
    - Typically, job functions with greater responsibility have greater authority to access resources.
    - A subordinate job function may have a subset of the access rights of the superior job function.
- Role hierarchies make use of the concept of inheritance to enable one role to implicitly include access rights associated with a subordinate role.

## RBAC Constraints

- Constraints provide a means of adapting RBAC to the specifics of administrative and security policies in an organisation.
- A constraint is a defined relationship among roles or a condition related to roles:
    - mutually exclusive roles
    - cardinality
    - prerequisite roles

## RBAC Constraints

### Mutually exclusive roles

- Users can be assigned to only one role in a mutually exclusive set of roles.
- Constraint can be applied statically or dynamically (in a session).
- Supports a separation of duties and capabilities within an organisation.
- The purpose is to increase the difficulty of collusion among individuals of different skills or divergent job functions to thwart security policies.

## RBAC Constraints

### Mutually exclusive roles

- Separation can be further reinforced by use of mutually exclusive permission assignments.

  1. A user can only be assigned to one role in the set (either statically or dynamically).

  2. Any permission (access right) can be granted to only one role in the set.

- Thus, the set of mutually exclusive roles have non overlapping permissions.

  - If two users are assigned to different roles in the set, then the users have non overlapping permissions while assuming those roles.

# RBAC Constraints

## Cardinality

- Cardinality refers to setting a maximum number with respect to roles.
- Maximum number of users that can be assigned to a given role.
  - For example, a project leader role or a department head role might be limited to a single user.
- Maximum number of roles that a user is assigned to, or can activate in a session.
- Maximum number of roles that can be granted a particular permission.
  - Risk mitigation technique for a sensitive or powerful permission.

# RBAC Constraints

## Prerequisite Role

- A system might be able to specify a prerequisite role, which dictates a user can only be assigned to a particular role if it is already assigned to some other specified role.
- A prerequisite can be used to structure the implementation of the least privilege concept.
  - A user can be assigned to a higher role only if it is already assigned an immediately lower role.
  - Then, if the user does not need all of the permissions of a higher role for a given task, the user can invoke a session using only the required subordinate role.
  - Prerequisite role tied to the concept of hierarchy implies the RBAC Consolidated Model.

## Attribute-Based Access Control (ABAC)

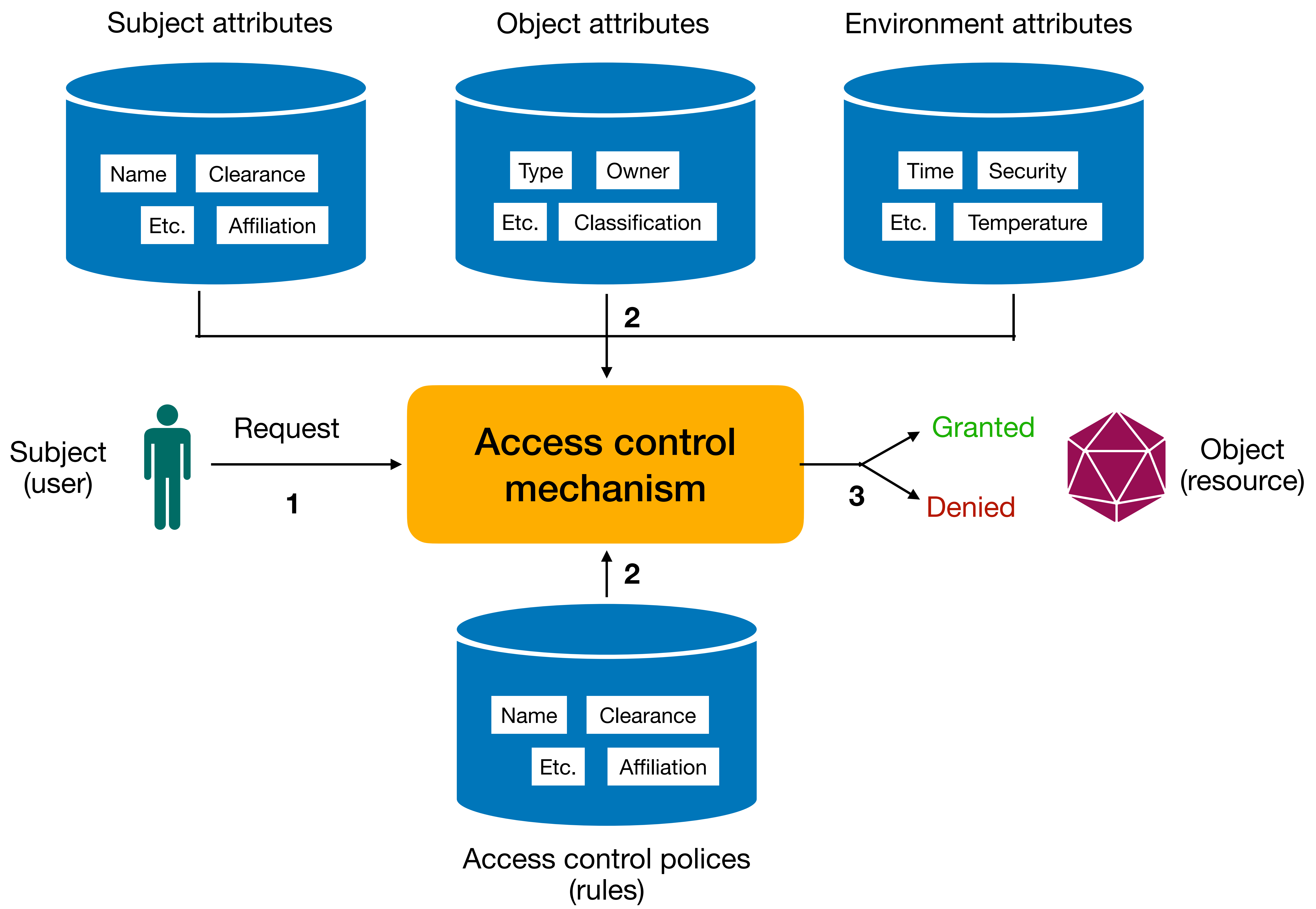
- A relatively recent development in access control technology.
- Defines authorisations that express conditions on properties of the resource, the subject and the environment.
  - Flexibility and expressiveness.
  - Performance penalty can be an obstacle for general usage.
- Well-suited for web services and cloud computing systems.
  - E.g. eXtensible Access Control Markup Language (XAMCL).
- Elements of an ABAC model:
  - Attributes, which are defined for entities in a configuration;
  - A policy model, which defines the ABAC policies;
  - The architecture model, which applies to policies that enforce access control.

## Attribute-Based Access Control (ABAC)
### Attributes

- Characteristics that define specific aspects of the subject, object and environment conditions that are predefinedby an authority.
- Contain information that indicates the class of information given by the attribute, a name, and a value.
    - E.g., Class = HospitalRecordsAccess, Name = PatientInformationAccess, Value = BusinessHoursOnly.
- **Subject attributes:** A subject is an active entity (e.g., a user, an application, a process, or a device) that causes information to flow among objects or changes the system state. A subject role can also be an attribute.
- **Object attributes:** An object, also referred to as a resource, is a passive (in the context of the given request) information system–related entity (e.g., device, file, record, table, processe, program, network, domain) containing or receiving information.
- **Environment attributes:** They describe the operational, technical, and even situational environment or context in which the information access occurs (often ignored in most access control policies). E.g. Current date and time, current security level.

# Attribute-Based Access Control (ABAC)

## Logical Architecture



## Attribute-Based Access Control (ABAC)

### Logical Architecture

1. A subject requests access to an object. This request is routed to an access control mechanism.

2. The access control mechanism is governed by a set of rules that are defined by a preconfigured access control policy. Based on these rules, the access control mechanism assesses the attributes of the subject, object, and current environmental conditions to determine authorisation.

3. The access control mechanism grants the subject access to the object if access is authorised, and denies access if it is not authorised.

## Attribute-Based Access Control (ABAC)

### Policies

- A **policy** is a set of rules and relationships that govern allowable behaviour within an organisation, based on the privileges of subjects and how resources or objects are to be protected under which environment conditions.

- **Privileges** (aka rights, authorisations, entitlements) represent the authorised behaviour of a subject; they are defined by an authority and embodied in a policy. Policy is typically written from the perspective of the object that needs protecting, and the privileges available to subjects.

## Attribute-Based Access Control (ABAC)

- ABAC is a logical access control model that is distinguishable because it controls access to objects by evaluating rules against the attributes of entities (subject and object), operations, and the environment relevant to a request.
- ABAC systems are capable of enforcing DAC, RBAC, and MAC concepts.
- How to model the access control to movies available from a streaming service?
    - Movies can be age-restricted
    - Only premium users can view recent released movies
    - Regular users can watch recent released movies during a promotional periods