Access Control 2

Dr. Asem Kitana

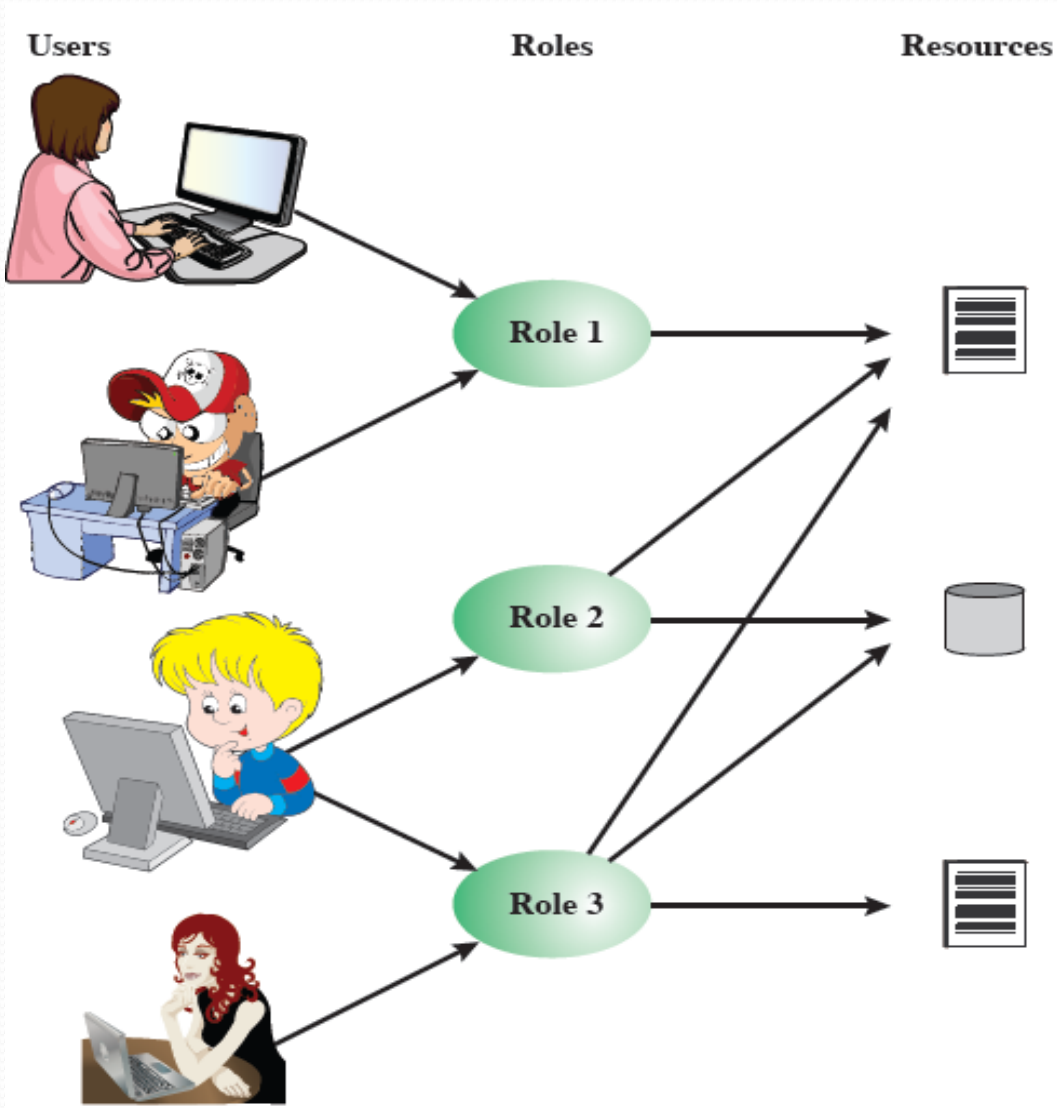
Role-Based Access Control (RBAC)

Role-Based Access Control

Access based on ‘role’, not identity.

Many-to-many relationship between users and roles.

Roles often static, but could be dynamic.



Role-Based Access Control

- RBAC: users are assigned to roles; access rights are assigned to roles.
- Roles typically job functions and positions within organization, e.g. faculty member in a university, doctor in a hospital.
- Users may be assigned multiple roles.
- Roles could be static or dynamic.
- Sessions are temporary assignments of user to role(s).
- RBAC access control matrix has two variations:
  - Users-Roles matrix
  - Roles-Objects matrix

Example of Users-Roles matrix

- This matrix relates individual users to roles.
- Typically, there are many more users than roles.
- Each matrix entry /cell is either blank or marked.
- The marked entry indicating that this user is assigned to this role.
- A single user may be assigned multiple roles.
- Multiple users may be assigned to a single role.

	R_1	R_2	• • •	R_n
U_1	×			
U_2	×			
U_3		×		×
U_4				×
U_5				×
U_6				×
⋮				
U_m	×			

Example of Roles-Objects matrix

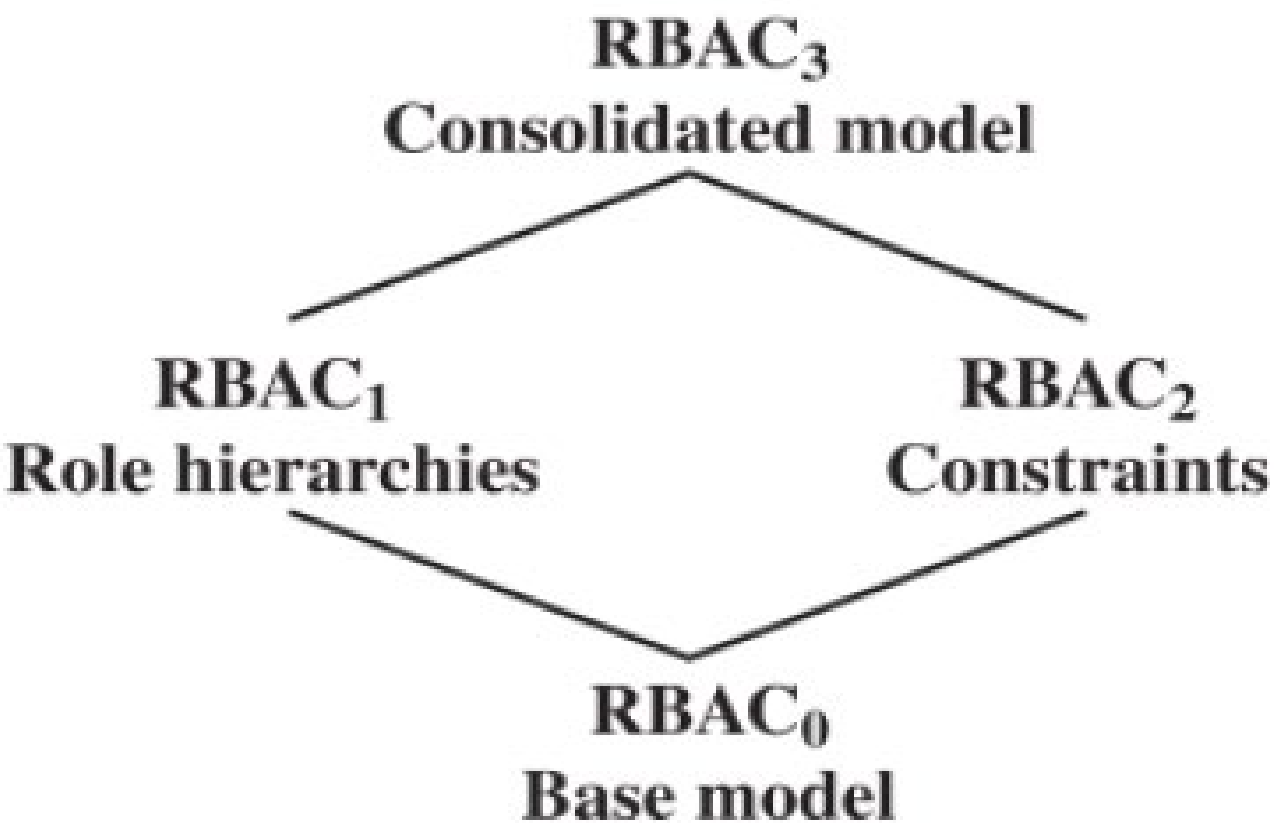
ROLES \ OBJECTS	R_1	R_2	R_n	F_1	F_2	P_1	P_2	D_1	D_2
R_1	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
R_2		control		write *	execute			owner	seek *
• • •									
R_n			control		write	stop			

- This matrix has the same structure as the DAC access matrix, but with roles as subjects.
- Typically, there are few roles and many objects.
- Entries/cells represent access rights assigned to the roles.

RBAC Models

- RBAC0: contains the minimum functionality for an RBAC system
- RBAC1: includes the RABC0 functionality + role hierarchies (which enable one role to inherit permissions from another role).
- RBAC2: includes the RABC0 functionality + constraints (which restrict the ways in which the components of an RBAC system may be configured).
- RBAC3: contains the functionality of RBAC0 + RBAC1 + RBAC2.

RBAC Models



RBAC0 - Base Model

Role hierarchies and constraints are eliminated from this model, RBAC0 model contains the following four types of entities:

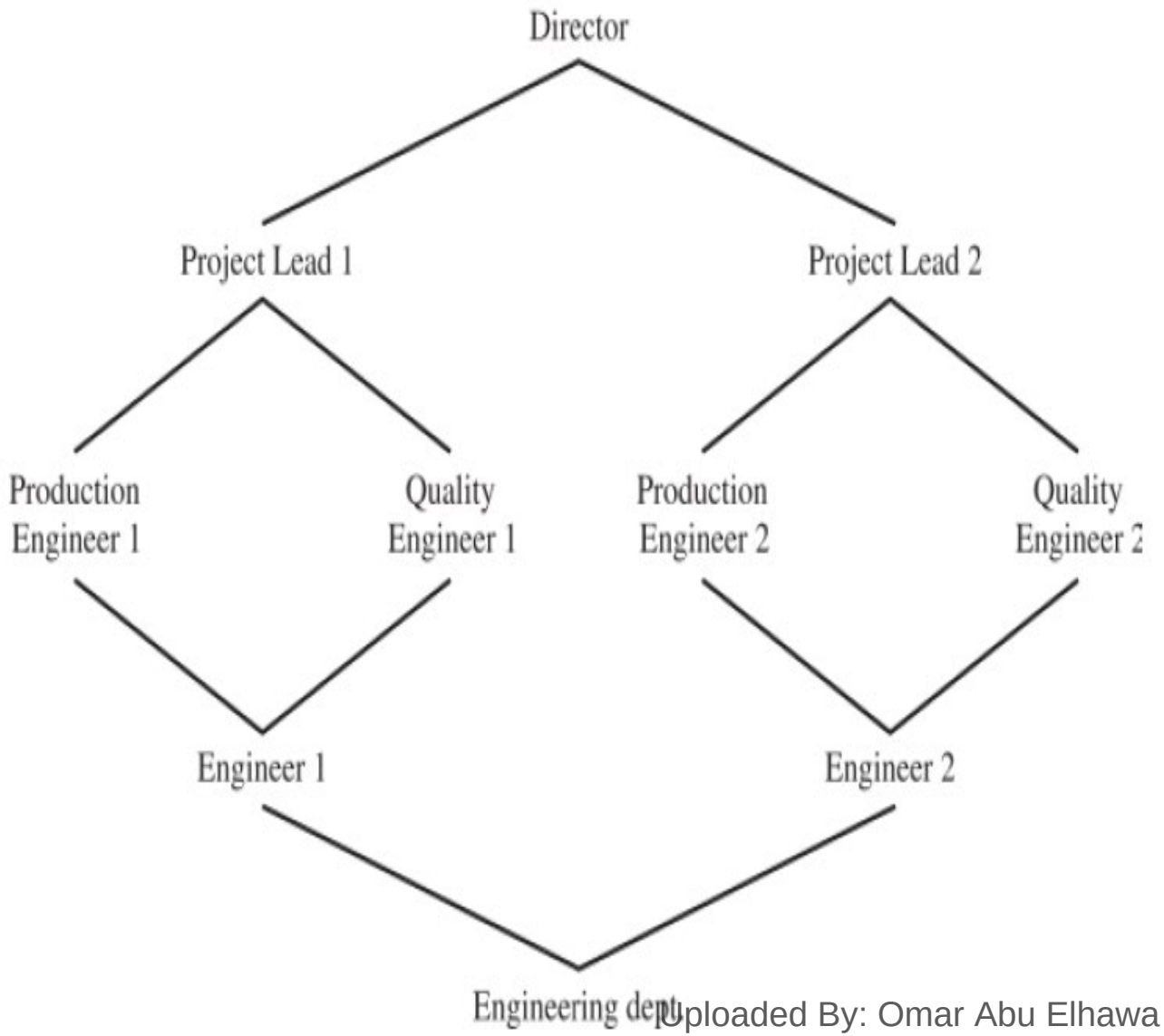
- **User:** An individual with user ID has access to a system.
- **Role:** A named job function within an organization. Typically, each role has a level of authority and responsibility.
- **Permission:** An equivalent term to *access right* or *privilege*.
- **Session:** A temporary mapping/relationship between a user and a set of roles to which the user is assigned.

RBAC1 - Role Hierarchies

- Role hierarchies provide a methods of reflecting the hierarchical structure of roles in an organization (i.e. hierarchy of an organization can be reflected in roles).
- Typically, job functions with greater responsibility have greater authority to access resources.
- Role hierarchies use of the concept of inheritance to enable one role to implicitly include access rights associated with a lower role.
- A higher role includes all access rights of lower role.

Example of Role Hierarchy

- Director has most privileges.
- Each role inherits all privileges from lower roles.
- A role can inherit from multiple roles.
- Additional privileges can be assigned to a role.



RBAC2 - Constraints

- Constraints define conditions between roles or conditions on roles.
- Types of constraints:
  - Mutually exclusive roles.
  - Cardinality.
  - Prerequisite roles.

RBAC2 - Constraints

- **Mutually exclusive roles:**
  - A user can only be assigned to one role in the set.
  - Any permission (access right) can be granted to only one role in the set.
  - The mutually exclusive constraint supports a separation of duties and capabilities within an organization.
  - 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.

RBAC2 - Constraints

- **Cardinality:**

refers to setting a maximum number with respect to roles, for instance:

- Setting maximum number of users assigned to a role
- Setting maximum number of roles a user can be assigned to
- Setting maximum number of roles that can be granted particular access rights.

Setting this kind of limitation, makes it easy to manage systems.

- **Prerequisite roles:**

- a user can be assigned a role only if that user already has been assigned to some other specified role.
- e.g. user can only be assigned a senior role if already assigned a junior role.

Scope of RBAC Models

Models	Hierarchies	Constraints
RBAC0	No	No
RBAC1	Yes	No
RBAC2	No	Yes
RBAC3	Yes	Yes

Attribute-based Access Control (ABAC)

Attribute-based access control

- ABAC represents a relatively recent development in access control technology.
- Defines authorizations that express conditions on properties of both the resource and the subject
  - Each resource has an attribute (e.g., the subject that created it).
  - a single access rule can specify the ownership privilege for all the creators of every resource.
- Strength: its flexibility and expressive power
- Considerable interest in applying the ABAC model to cloud services and web services (via the eXtensible Access Control Markup Language (XACML)).

Elements of ABAC

There are three key elements to an ABAC model:

- Attributes: which are defined for entities in a configuration.
- Policy model: which defines the ABAC policies.
- Architecture model: which applies to policies that enforce access control.

Attributes of ABAC

- Attributes are characteristics that define specific aspects of the subject, object, environment conditions, and/or requested operations that are predefined by an authority.
- There are three types of attributes in the ABAC model:
  - Subject attributes
  - Object attributes
  - Environment attributes

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.
- Each subject has associated attributes that define the identity and characteristics of the subject.
- Subject attributes such as subject's identifier, name, organization, and job title.

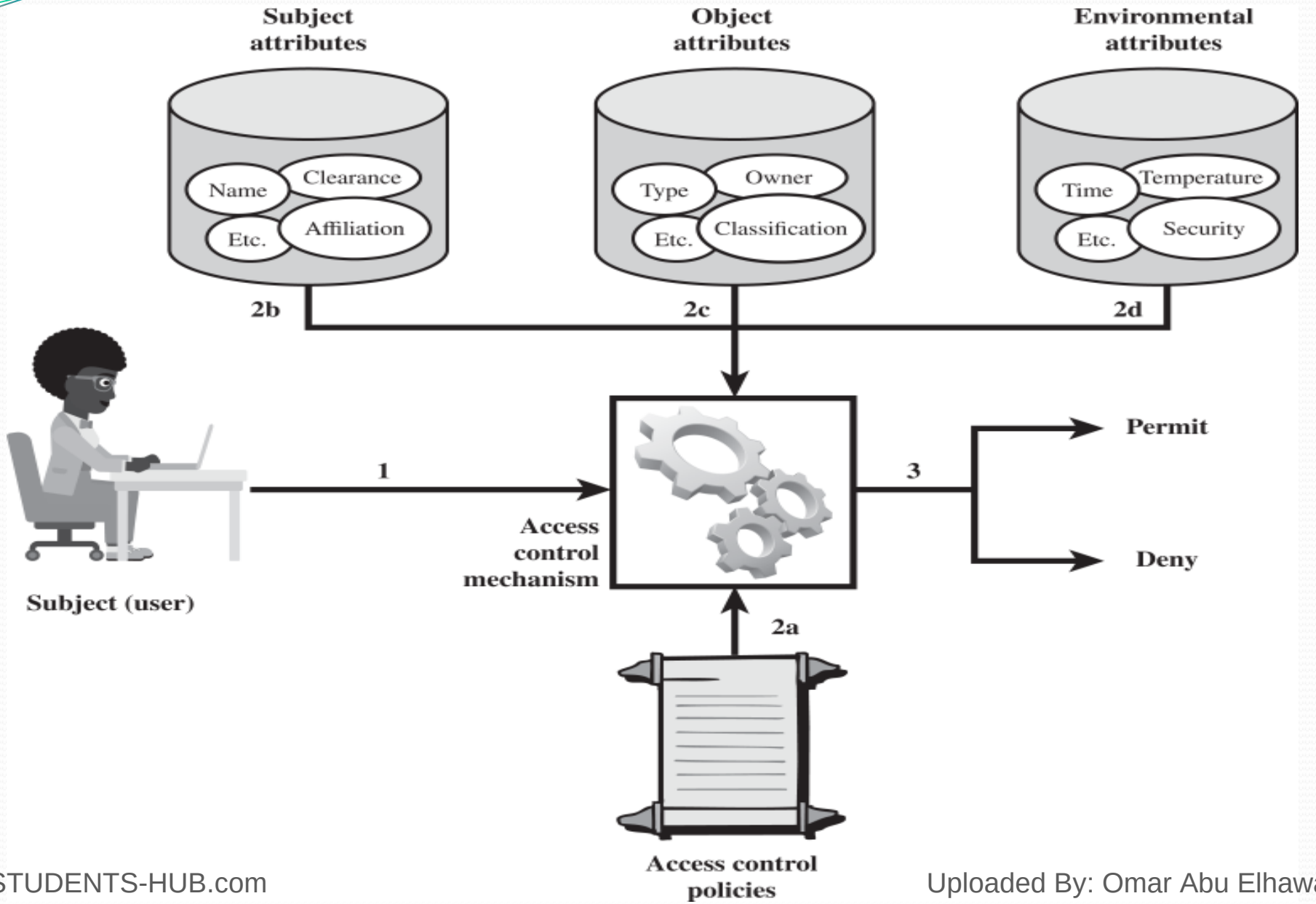
Object Attributes

- Object (a.k.a. resource) is a passive (in the context of the given request) entity (e.g., devices, files, records, processes, programs, networks, domains) containing or receiving information.
- Objects have attributes that can be used to make access control decisions.
- A Microsoft Word document, for example, may have attributes such as title, subject, date, and author.

Environment Attributes

- Describe the operational, technical, and even situational environment or context in which the information access occurs.
- For example, attributes, such as
  - Current date
  - Current time
  - Current location
  - Current temperature
  - Not associated with a particular subject nor a resource, but may used in applying an access control policy.

ABAC Scenario



ABAC Scenario

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 (2a) that are defined by a preconfigured access control policy. Based on these rules, the access control mechanism assesses the attributes of the subject (2b), object (2c), and current environmental conditions (2d) to determine authorization.

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

ABAC Granular Policy

SUBJECT	OBJECT	ENVIRONMENTAL
Title	Project	Geo-location
Division	PII	Network
Certifications	Sensitivity	Time of day
Training		

(Auditor + financial + during work hours) = Grant