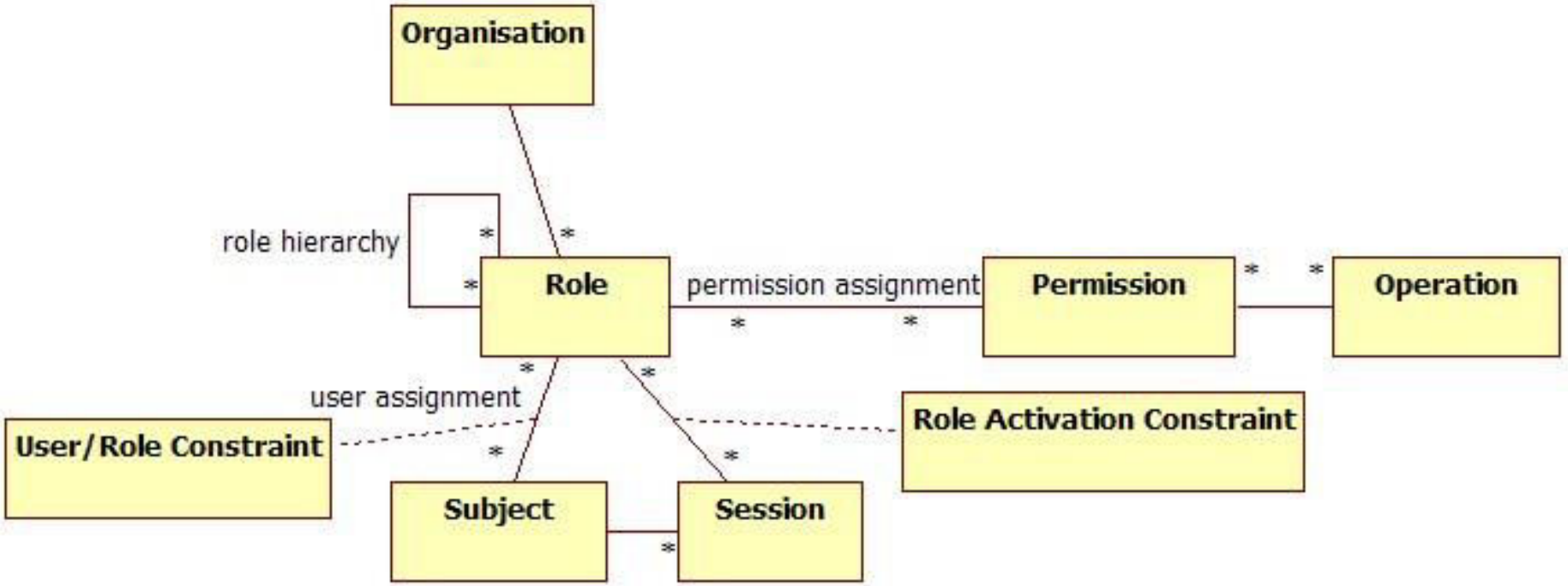
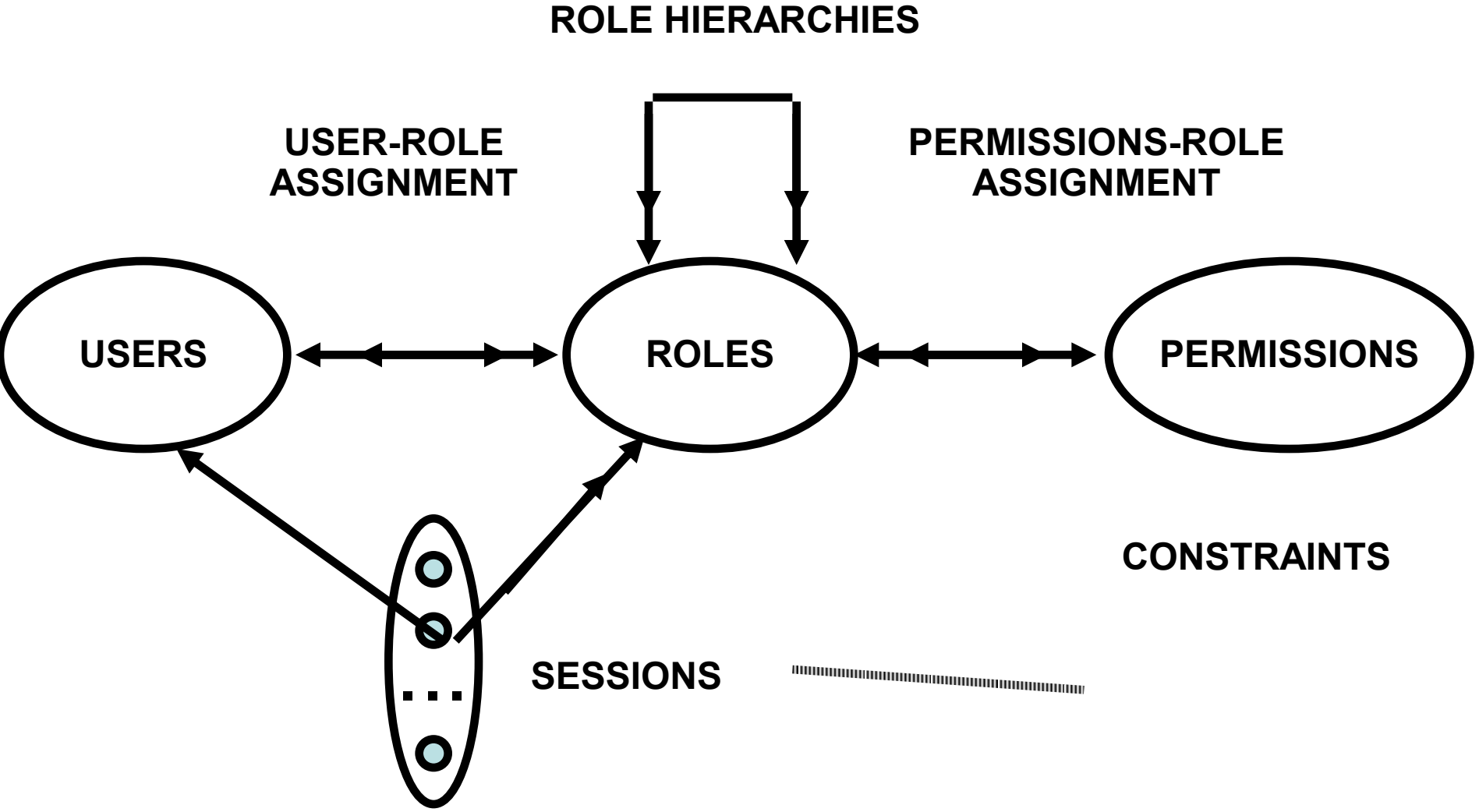
RBAC

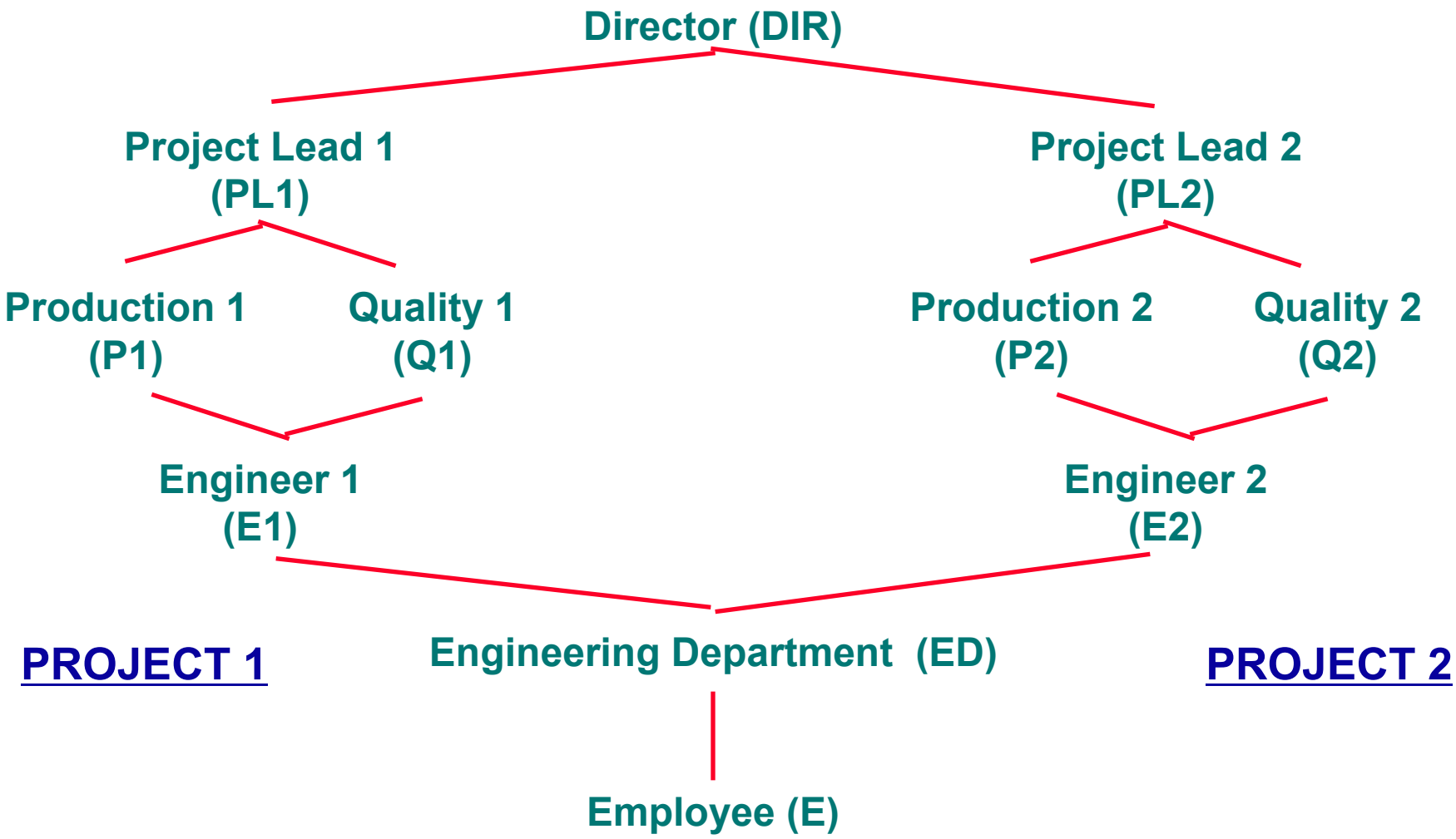
- Access depends on function, not identity
  - Example:
    - Allison, bookkeeper for Math Dept, has access to financial records.
    - She leaves.
    - Betty hired as the new bookkeeper, so she now has access to those records
  - The role of “bookkeeper” dictates access, not the identity of the individual.


Organisation
role hierarchy
Role
permission assignment
Permission
Operation
user assignment
User/Role Constraint
Role Activation Constraint
Subject
Session

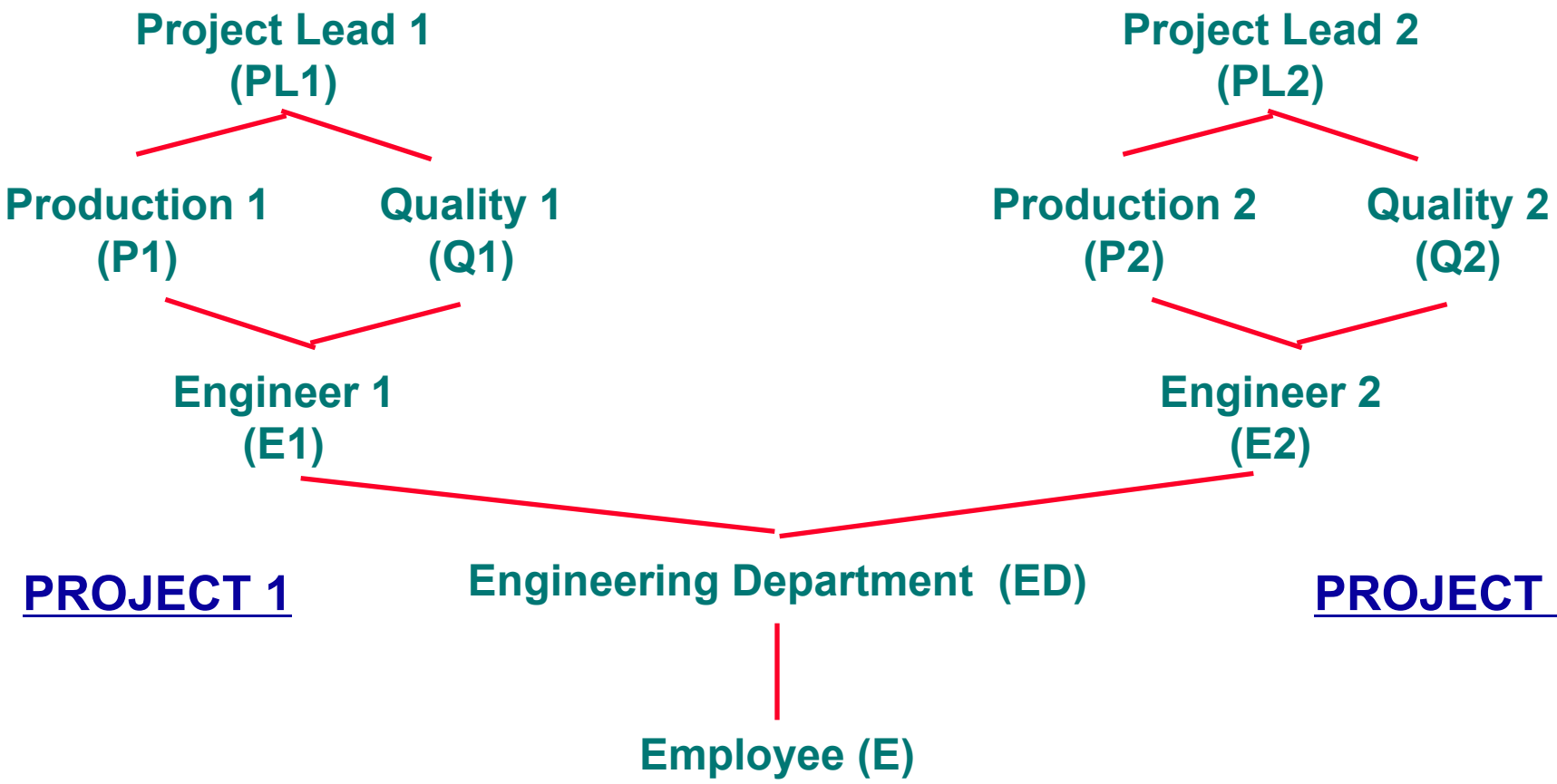
RBAC96 model (Currently foundation of a NIST/ANSI/ISO standard)



EXAMPLE ROLE HIERARCHY



EXAMPLE ROLE HIERARCHY



Definitions

- Role *r*: collection of job functions
  - *trans*(*r*): set of authorized transactions for *r*
- Active role of subject *s*: role *s* is currently in
  - *actr*(*s*)
- Authorized roles of a subject *s*: set of roles *s* is authorized to assume
  - *authr*(*s*)
- *canexec*(*s*, *t*) iff subject *s* can execute transaction *t* at current time

Axioms

- Let S be the set of subjects and T the set of transactions.
- *Rule of role assignment*:

$(\forall s \in S)(\forall t \in T)\ [canexec(s, t) \rightarrow actr(s) \neq \varnothing]$.

  - If s can execute a transaction, it has a role
  - This ties transactions to roles

- *Rule of role authorization*:

$(\forall s \in S)\ [actr(s) \subseteq authr(s)]$.

  - Subject must be authorized to assume an active role (otherwise, any subject could assume any role)

Axiom

- *Rule of transaction authorization*:

$$(\forall s \in S)(\forall t \in T)$$
$$[canexec(s, t) \rightarrow t \in trans(actr(s))].$$

  - If a subject *s* can execute a transaction, then the transaction is an authorized one for the role *s* has assumed

Containment of Roles

- Trainer can do all transactions that trainee can do (and then some). This means role *r* contains role *r′* (*r* > *r′*). So:

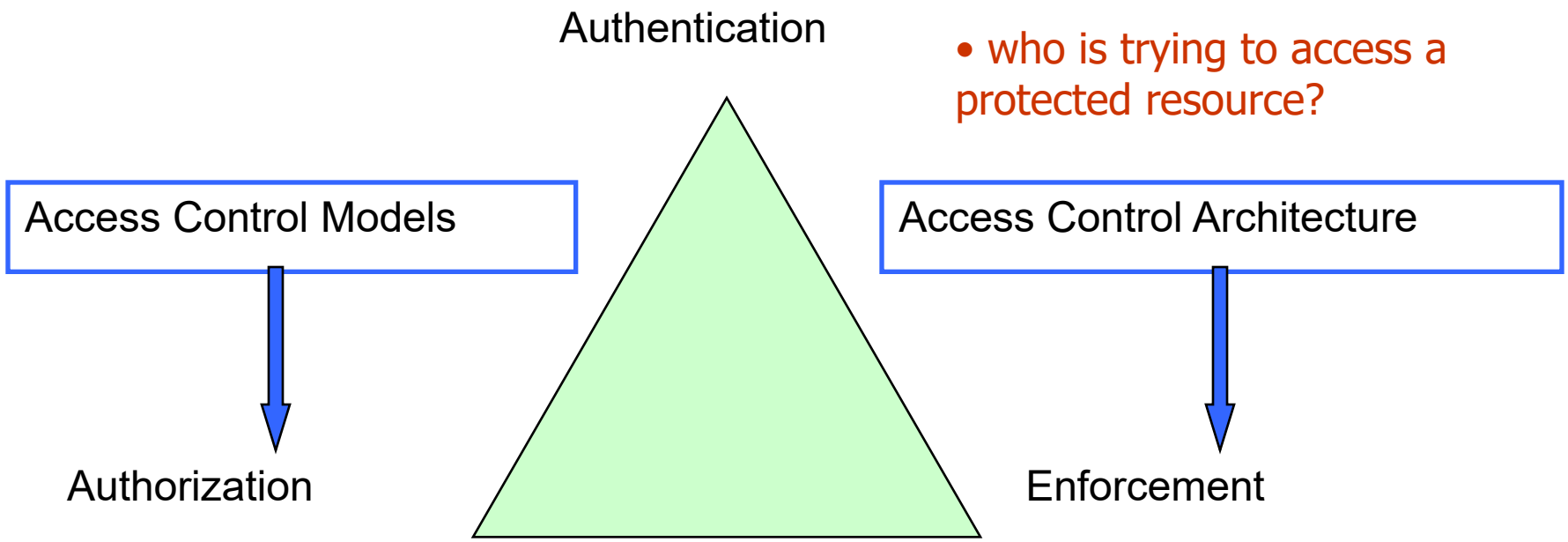
$$(\forall s \in S)[\, r' \in authr(s) \wedge r > r' \rightarrow r \in authr(s) \,]$$

Separation of Duty

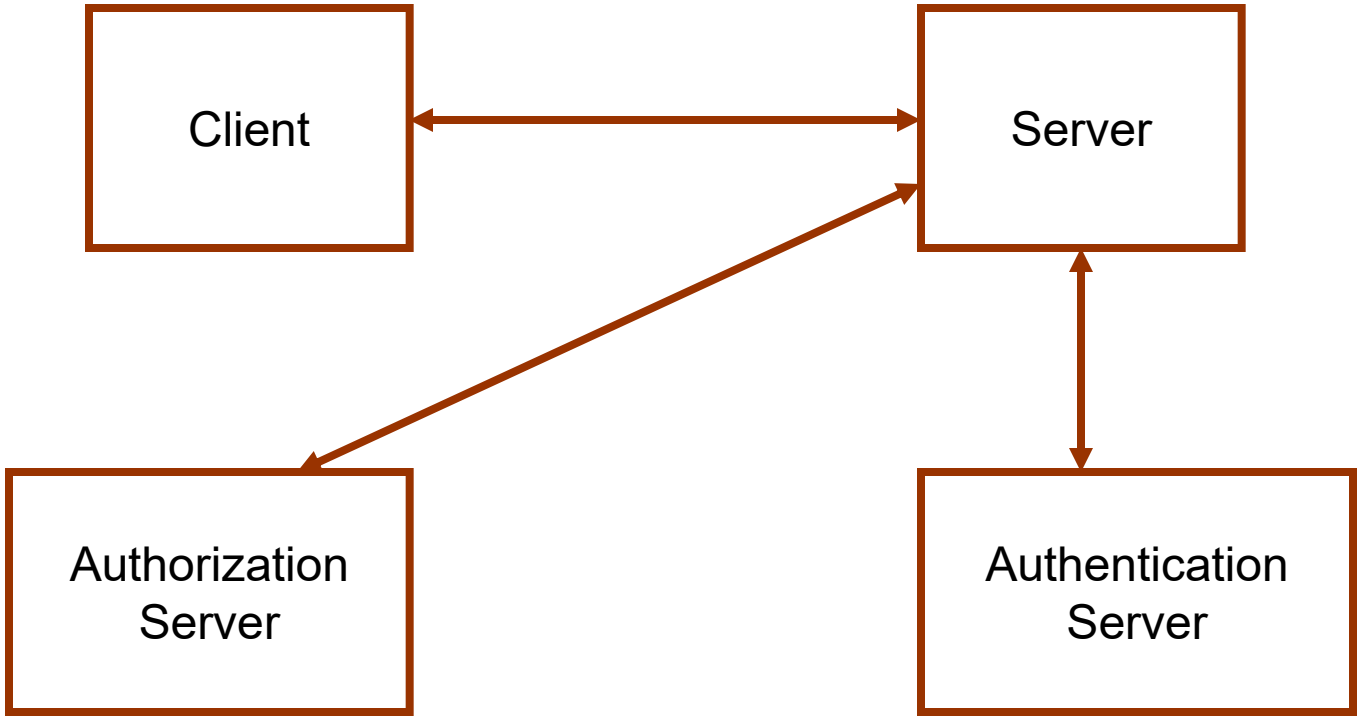
- Let *r* be a role, and let *s* be a subject such that $r \in auth(s)$. Then the predicate $meauth(r)$ (for mutually exclusive authorizations) is the set of roles that *s* cannot assume because of the separation of duty requirement.
- Separation of duty:

$$(\forall r_1, r_2 \in R)\ [\ r_2 \in meauth(r_1) \rightarrow$$
$$[\ (\forall s \in S)\ [\ r_1 \in authr(s) \rightarrow r_2 \notin authr(s)\ ]\ ]\ ]$$

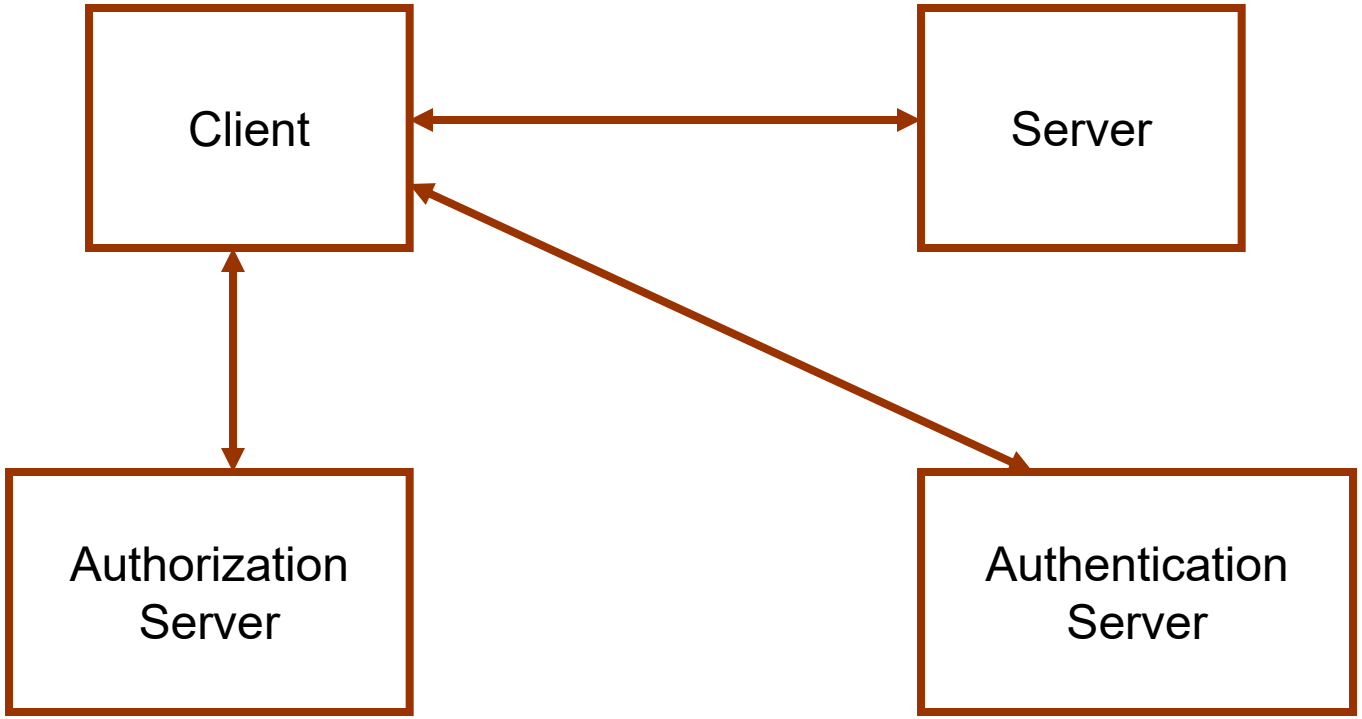
Safety in Access Control



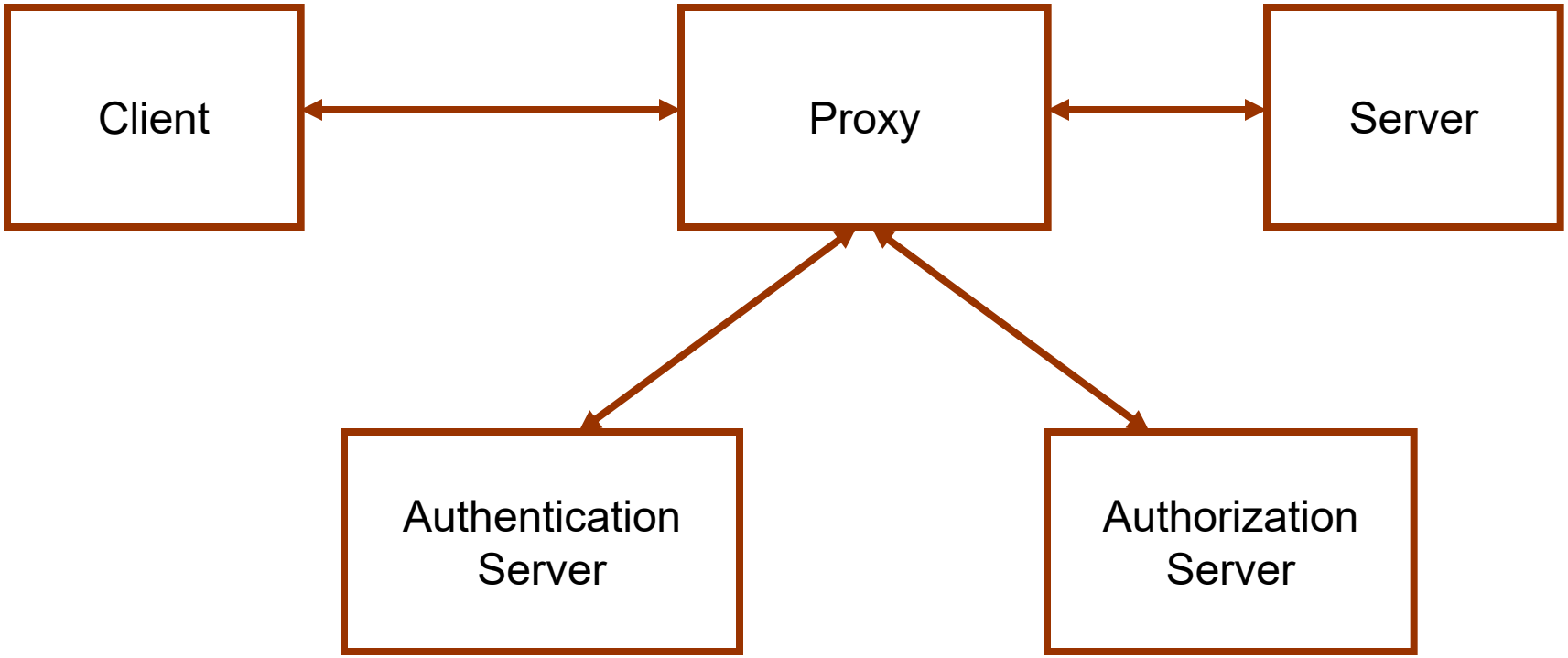
ACCESS-CONTROL ARCHITECTURE SERVER-PULL



ACCESS-CONTROL ARCHITECTURE USER-PULL



ACCESS-CONTROL ARCHITECTURE PROXY-BASED



Key Points

- Hybrid policies deal with both confidentiality and integrity
  - Different combinations of these
- ORCON model neither MAC nor DAC
  - Actually, a combination
- RBAC model controls access based on functionality

User Attributes

Attribute Based Management

- Attribute based encryption (ABE)
- Identity management
- Usage control

- Attribute Based Access Control (ABAC)

Attribute Administration

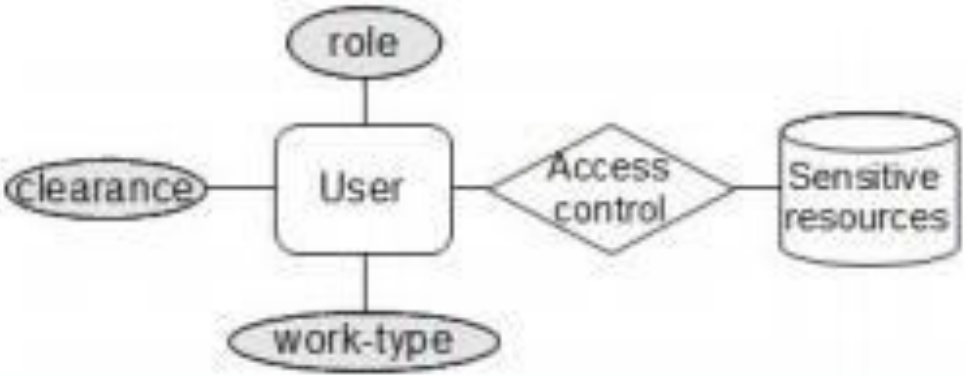
- In each organization, certain administrators have to assign user attributes values when the user is provisioned and modify user attributes values thereafter.

- Attributes of the same user constrain each other.
  - Administration rules are specified to regulate attribute modifications

Example Rule

clearance attribute of users can be assigned to "topsecret" **IF**: "officer" $\in$ role(u) $\wedge$ clearance(u) == "secret" $\wedge$ work-type(u) == "full-time".

Motivation for Reachability Problem



Example Authorization Policy

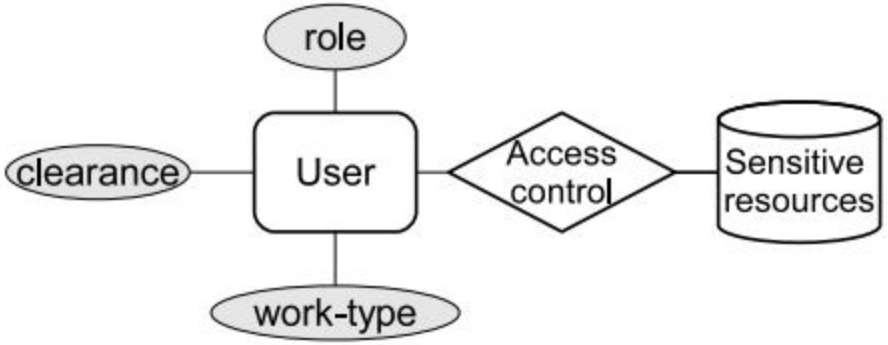
read(*sub*, *obj*) → ¬(clearance(u) == "topsecret" ∧ work-type(u) == "part-time")

Questions

Given a predefined administrative rules, will Alice ever be able to access *obj* in the future? It is equivalent to ask whether Alice's attribute can reach conditions which satisfies the authorization policy.

It might seem that a user can never be “part time” and with “topsecret” clearance at the same time

The policy may inadvertently allow that



(a) User attributes

Rule 1. Users clearance can be set as “topsecret” if:
user is with “officer” role **and**
“secret” clearance **and**
“full-time” work-type.

Rule 2. Users can be *unconditionally* assigned to “part-time” work-type.

(a) Sample attribute administration rules

A user may be “full time” and then assigned to “topsecret” clearance according to rule 1.
After that, he can be assigned to “part time” work-type according to rule 2.

Reachability Issues

References

- Matt Bishop -- see course details
- David E. Bell: Looking Back at the *Bell-La Padula Model*

- *Ross Anderson*

- Alexander Brodsky, Csilla Farkas, and Sushil Jajodia ,Database Security—Concepts,Approaches, and Challenges
  IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING. JANUARY-MARCH 2005
  http://ieeexplore.ieee.org/servlet/opac?punumber=8858