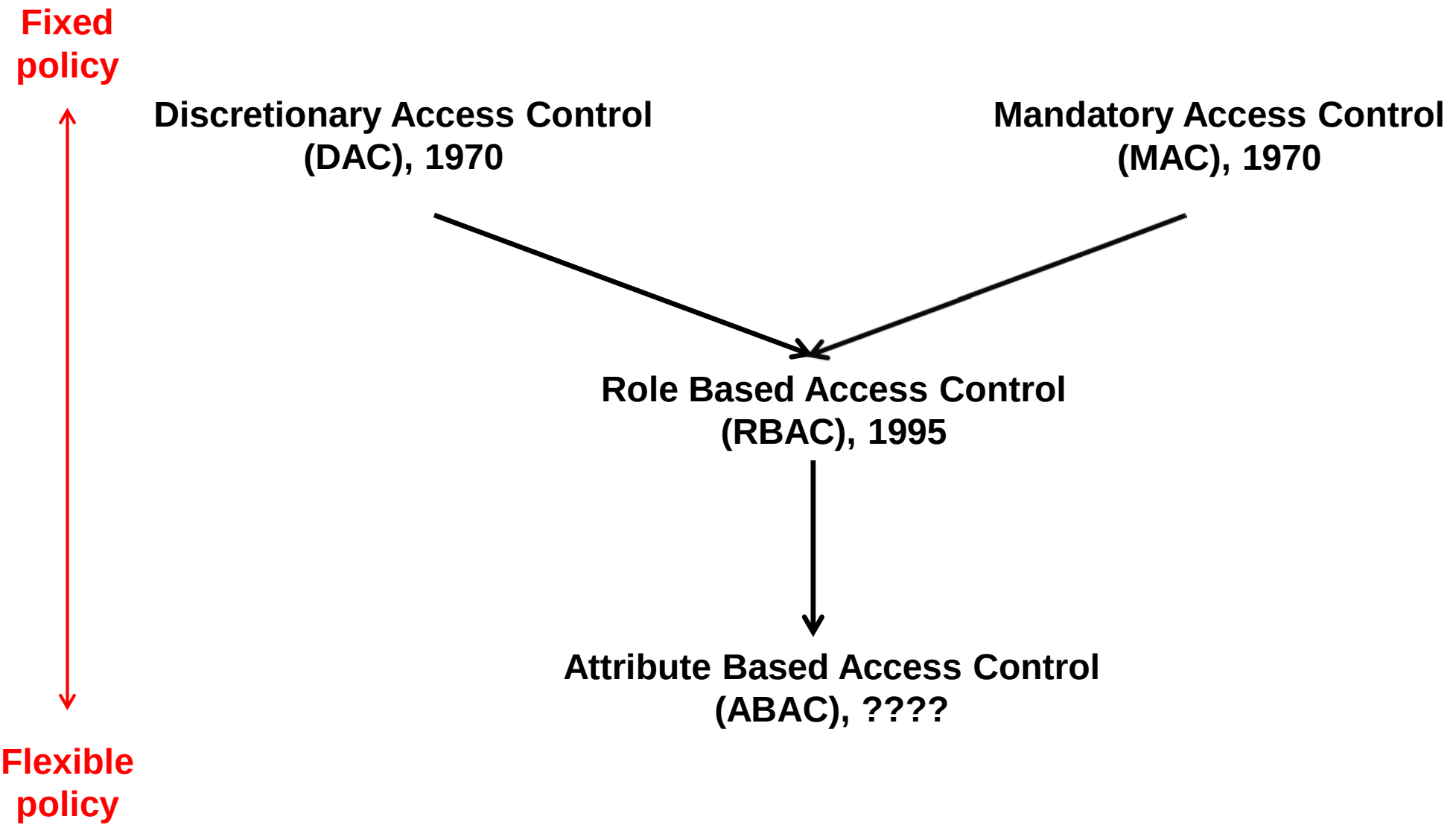
Role-Based Access Control (RBAC)

Prof. Ravi Sandhu
Executive Director and Endowed Chair

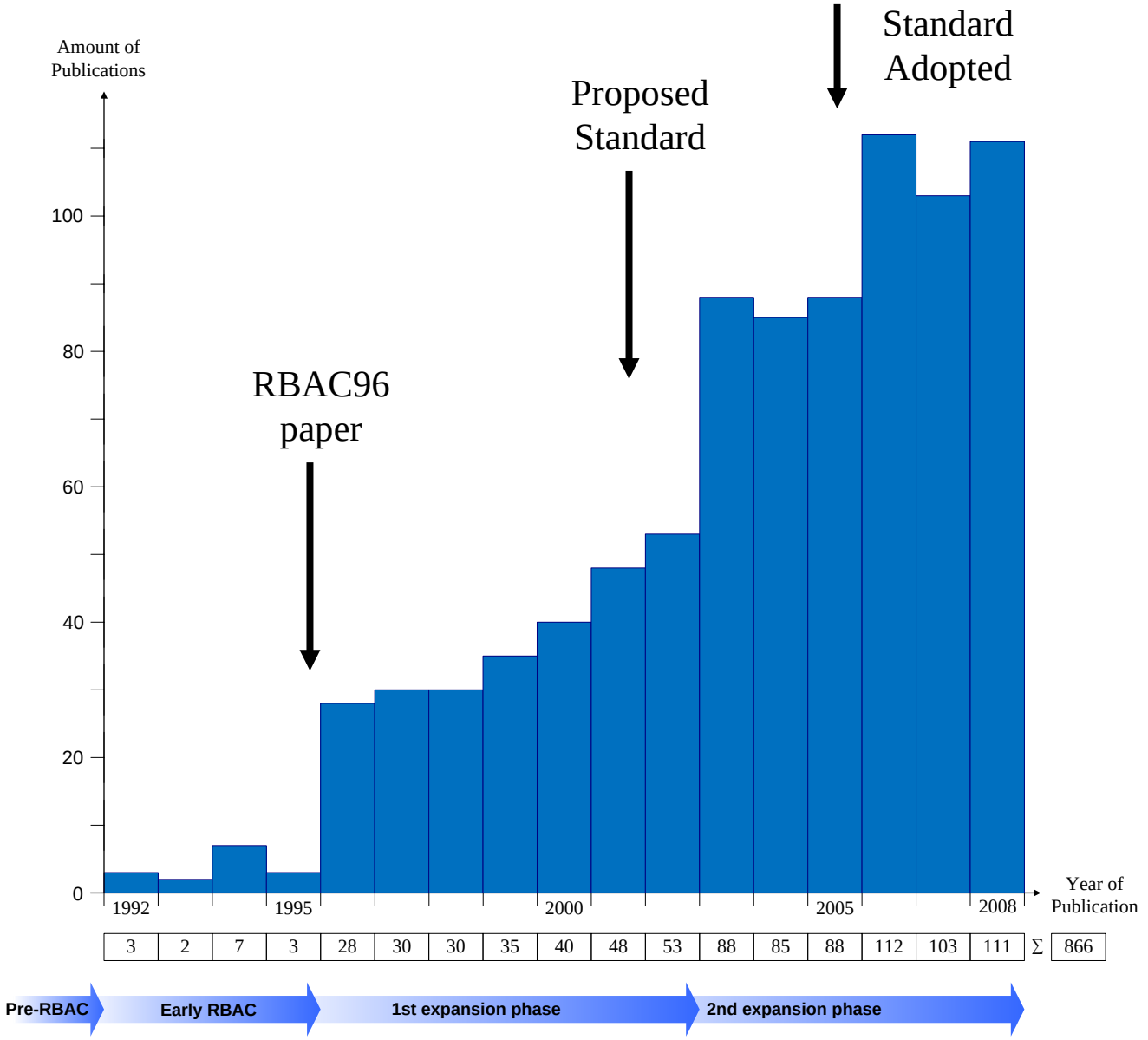
January 29, 2016

ravi.sandhu@utsa.edu
www.profsandhu.com

Access Control



The RBAC Story



RBAC: Role-Based Access Control

- Access is determined by roles
- A user's roles are assigned by security administrators
- A role's permissions are assigned by security administrators

First emerged: mid 1970s
First models: mid 1990s

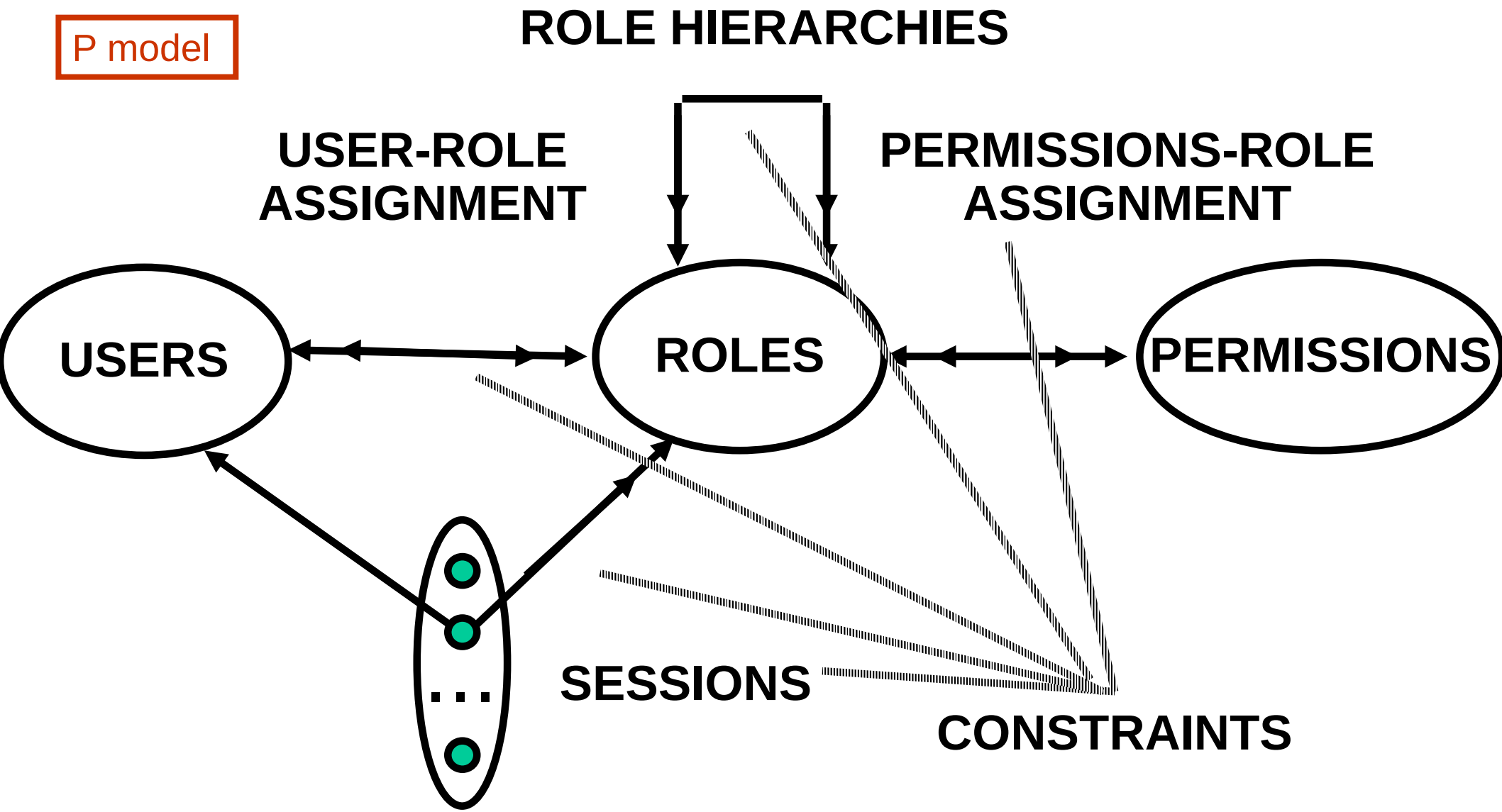
Is RBAC MAC or DAC or neither?

- RBAC can be configured to do MAC
- RBAC can be configured to do DAC
- RBAC is policy neutral

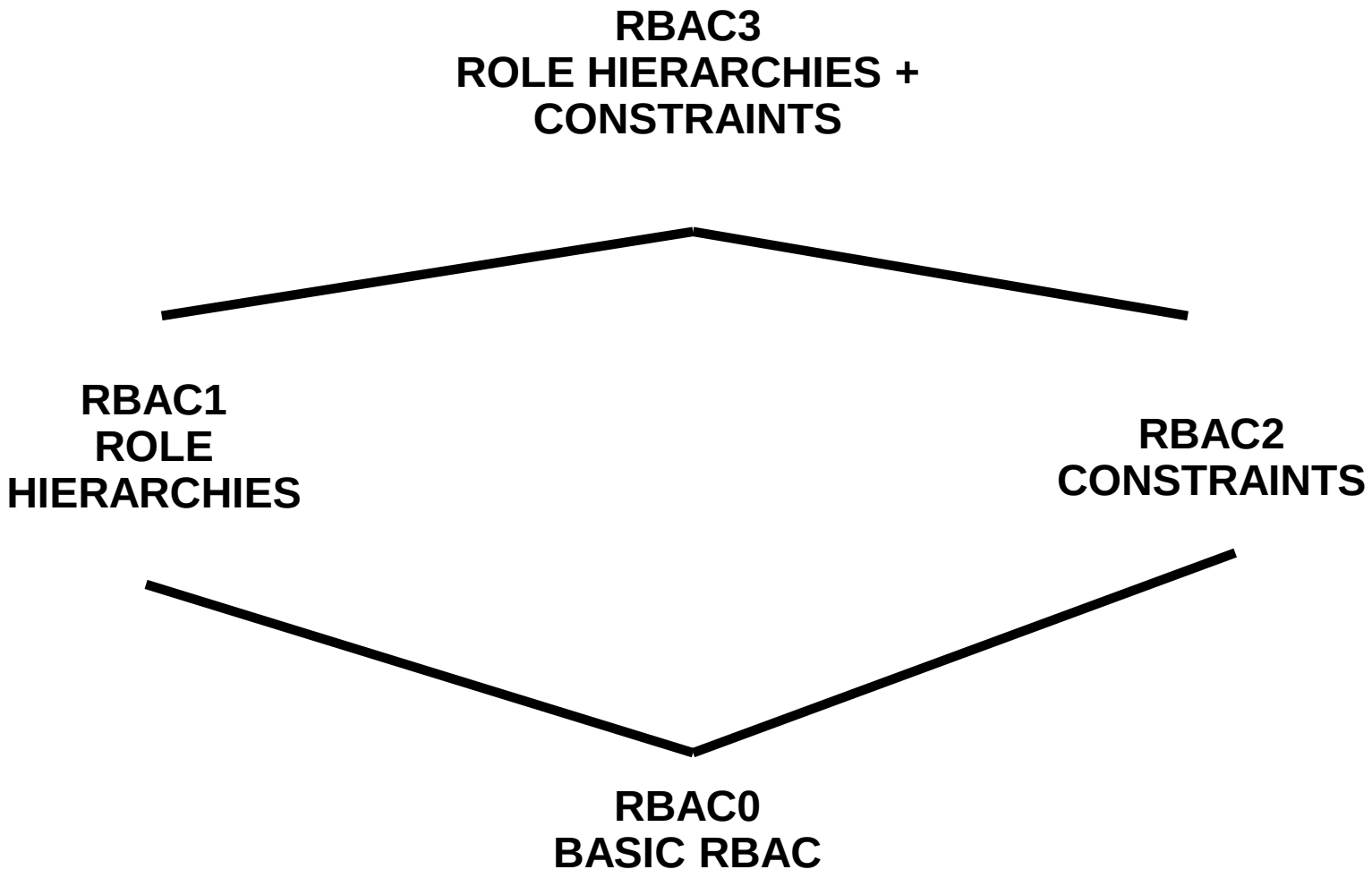
RBAC is neither MAC nor DAC!

RBAC96 Model Family

P model



RBAC96 Model Family



RBAC SECURITY PRINCIPLES

- **Abstraction** of Privileges
  - Credit is different from Debit even though both require read and write
- **Separation** of Administrative Functions
  - Separation of user-role assignment from role-permission assignment
- **Least Privilege**
  - Right-size the roles
  - Don’t activate all roles all the time
- **Separation of Duty**
  - Static separation: purchasing manager versus accounts payable manager
  - Dynamic separation: cash-register clerk versus cash-register manager

ROLES AS POLICY

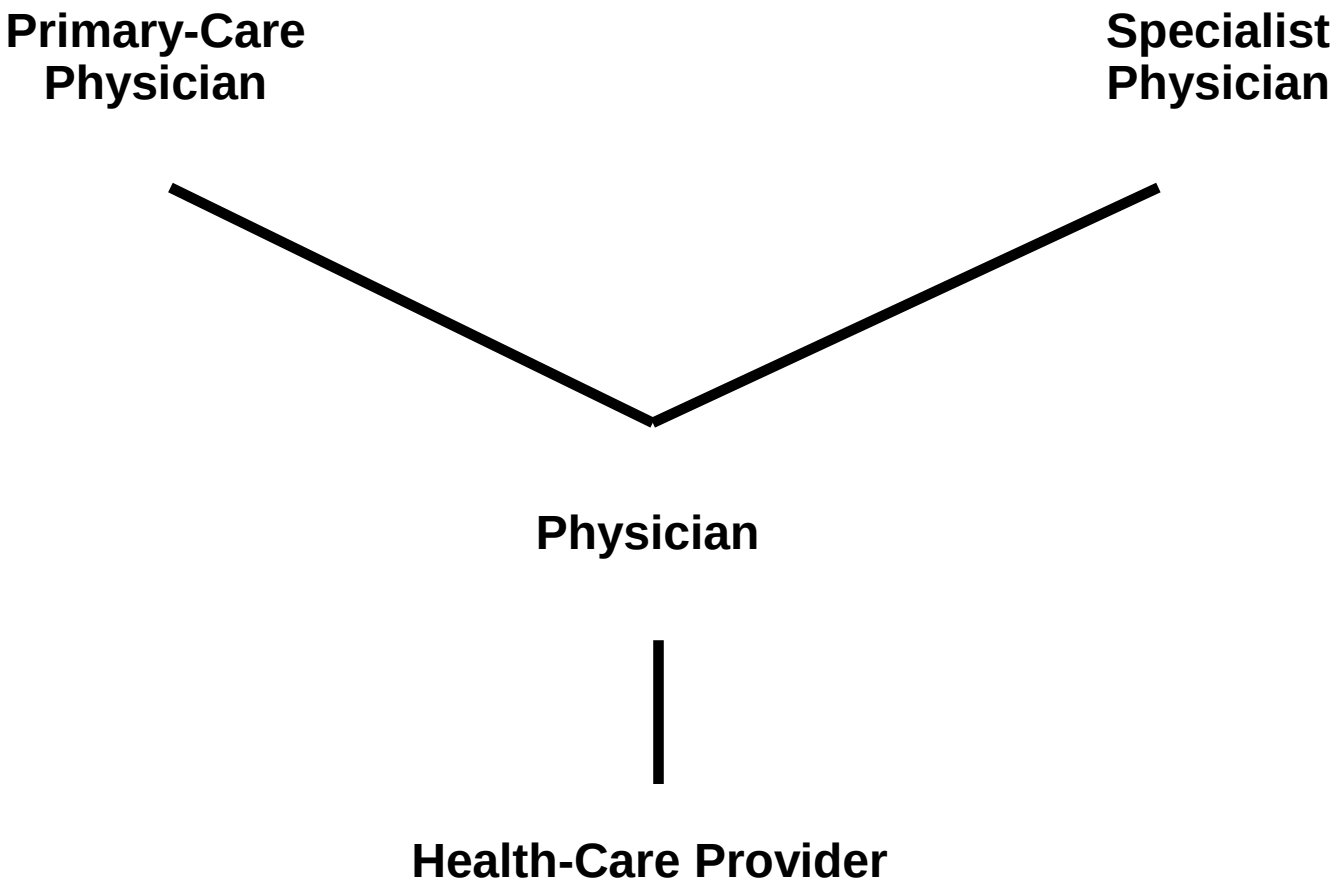
- A role brings together
  - a collection of users and
  - a collection of permissions
- These collections will vary over time
  - A role has significance and meaning beyond the particular users and permissions brought together at any moment

ROLES VERSUS GROUPS

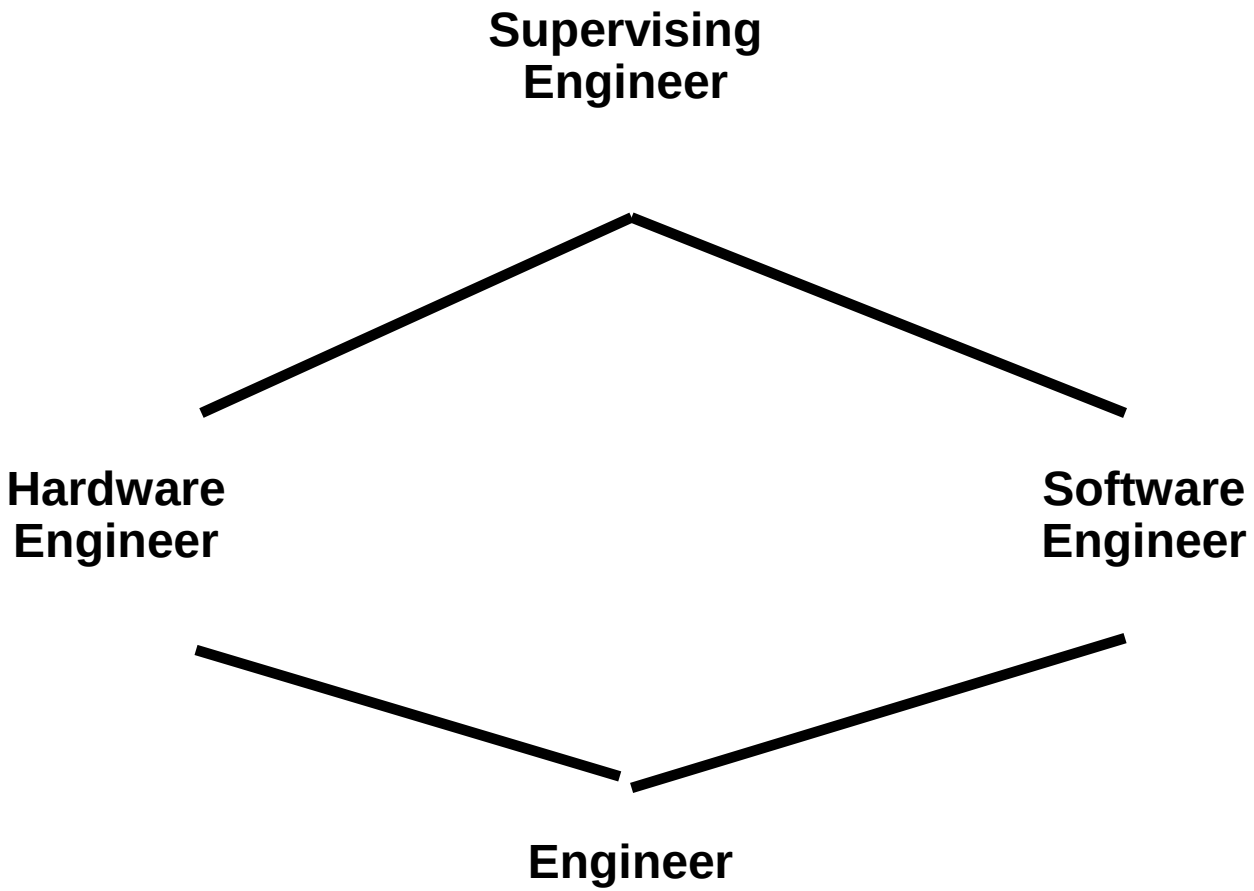
- Groups are often defined as
  - a collection of users
- A role is
  - a collection of users and
  - a collection of permissions
- Some authors define role as
  - a collection of permissions

- Most Operating Systems support groups
  - BUT do not support selective activation of groups
- Selective activation conflicts with negative groups (or roles)

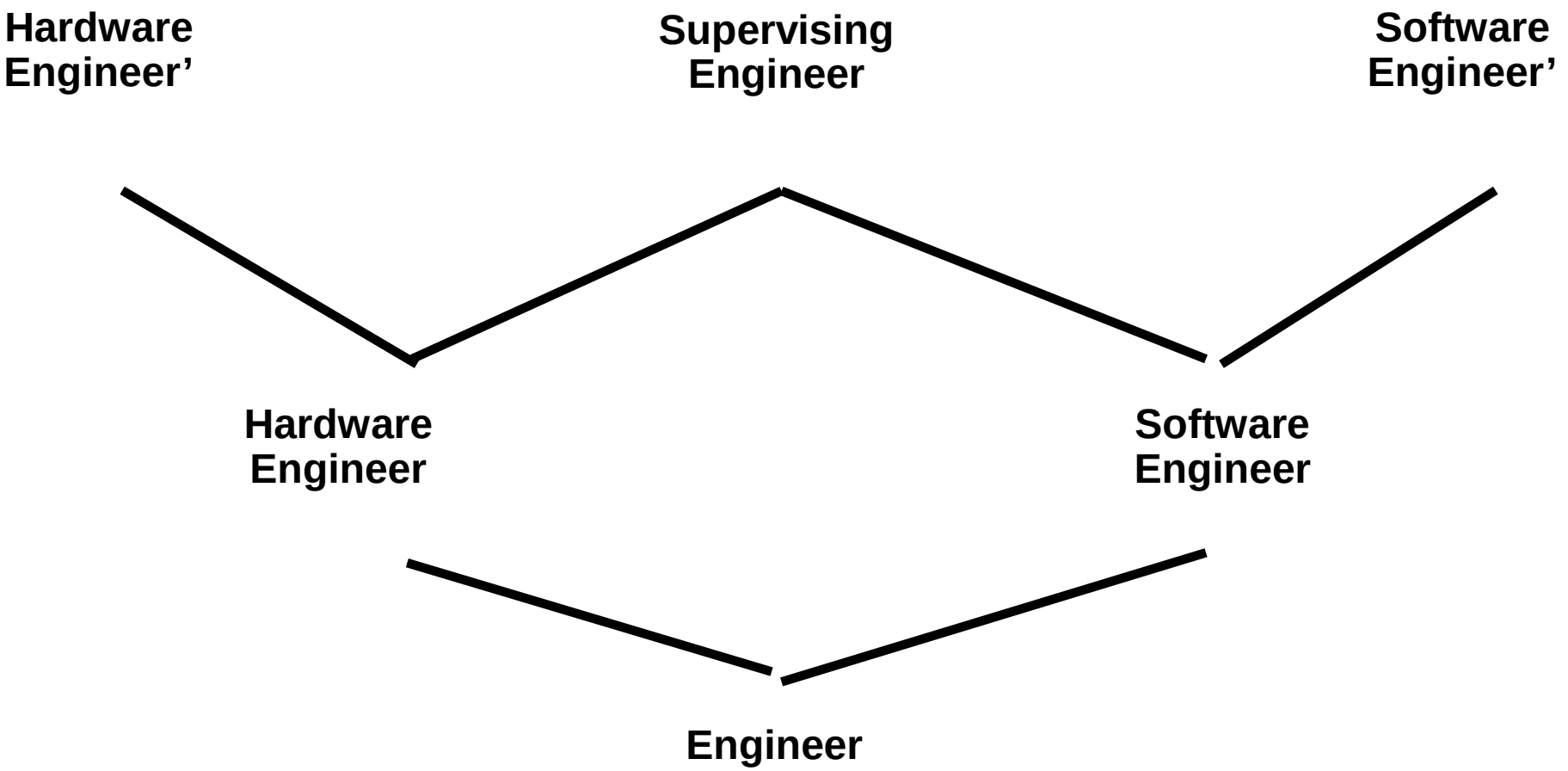
HIERARCHICAL ROLES



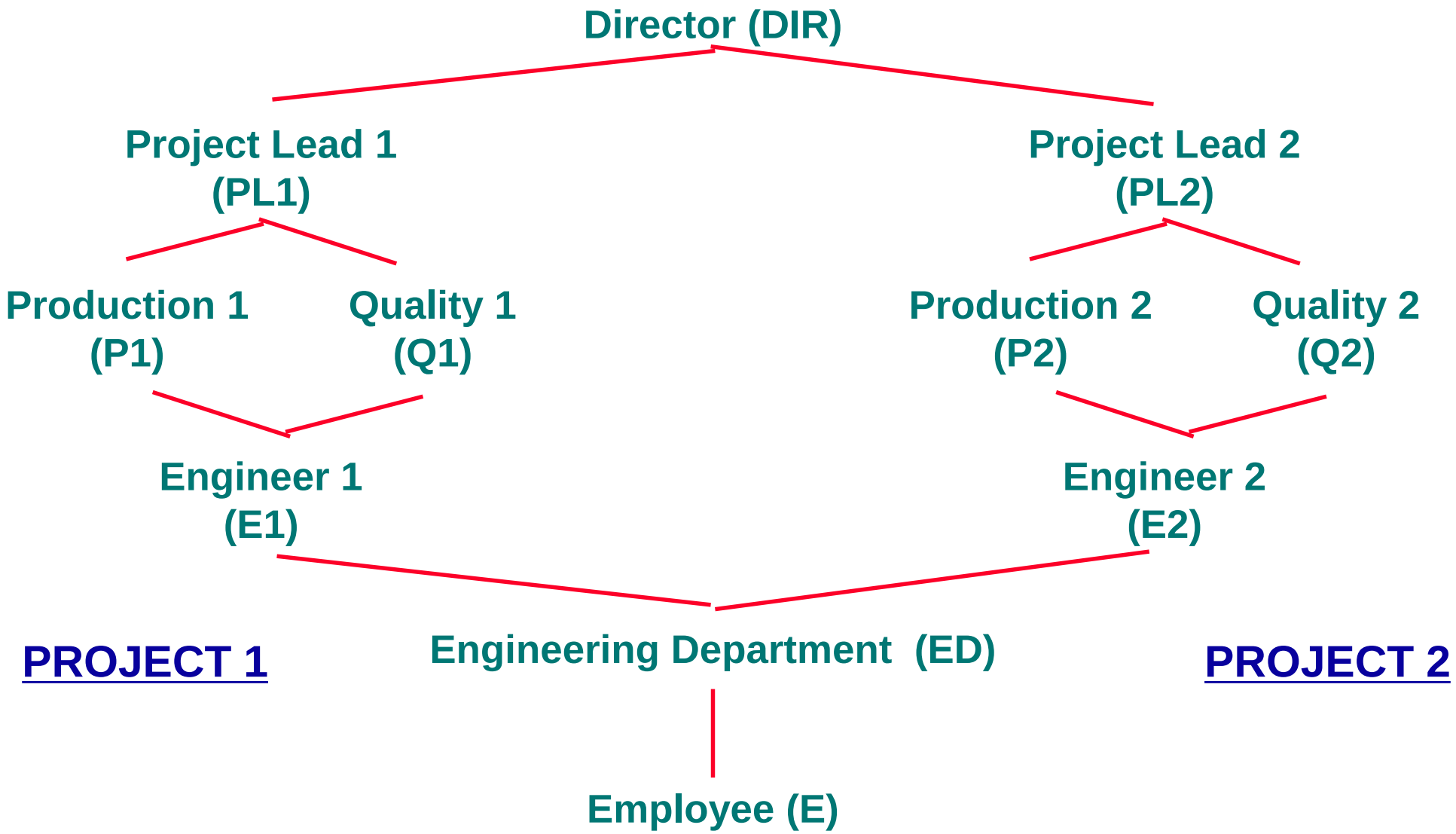
HIERARCHICAL ROLES



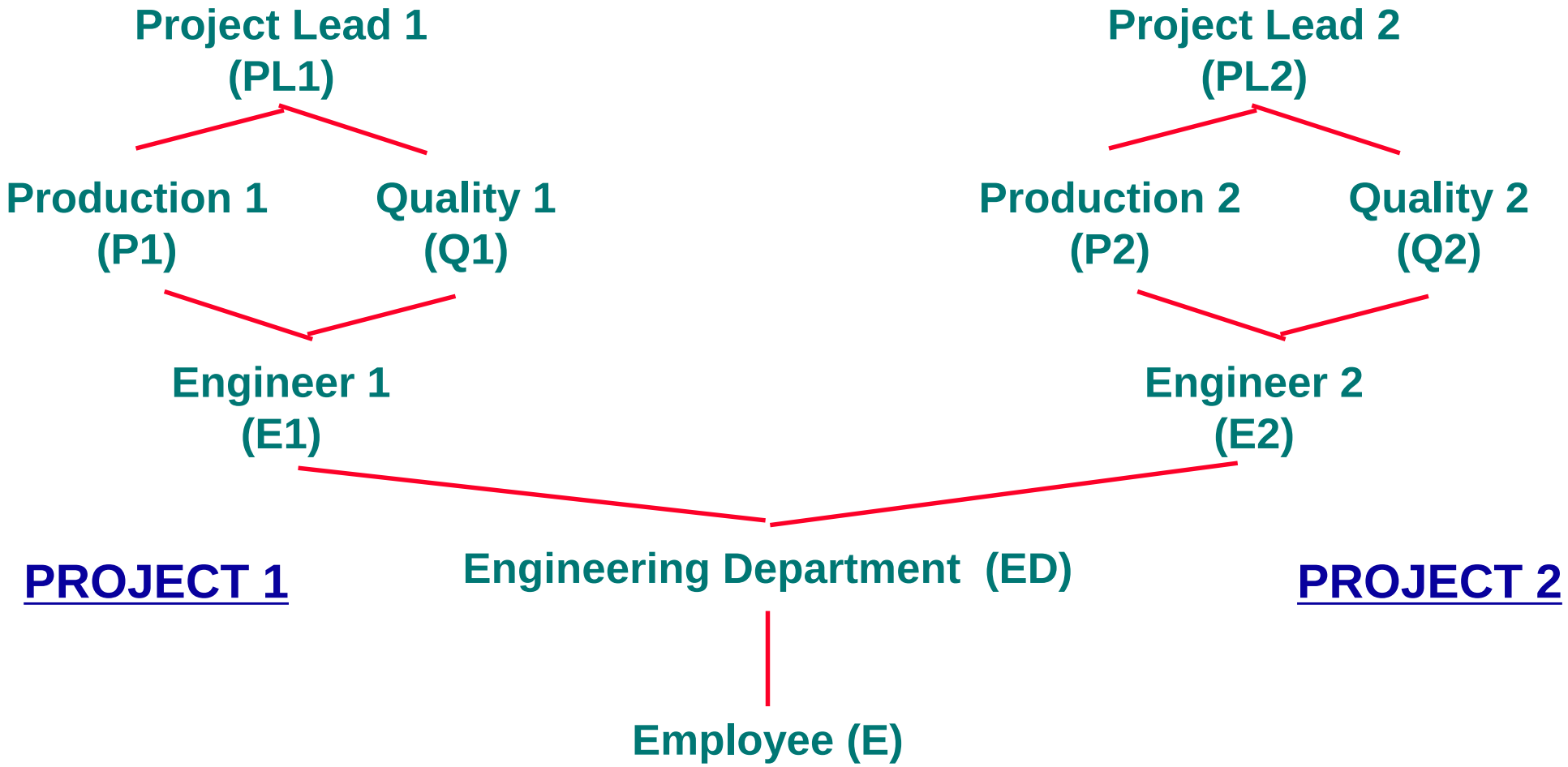
PRIVATE ROLES



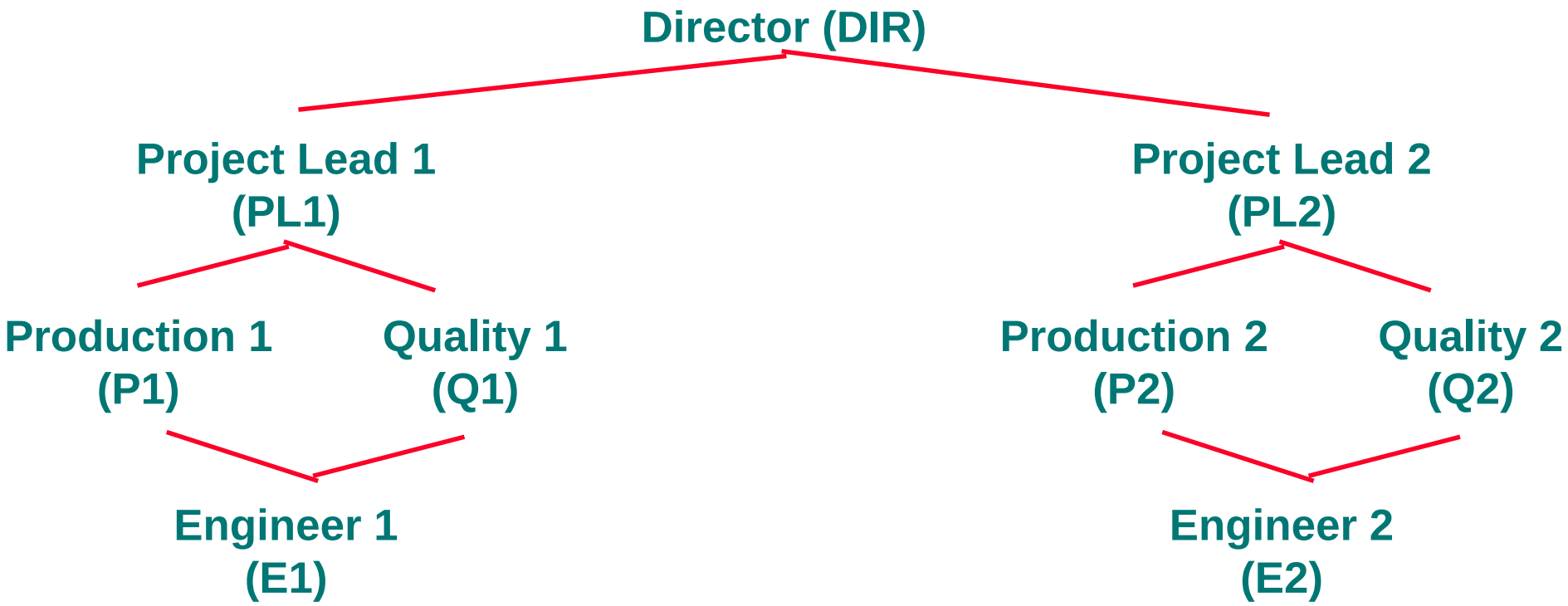
EXAMPLE ROLE HIERARCHY



EXAMPLE ROLE HIERARCHY



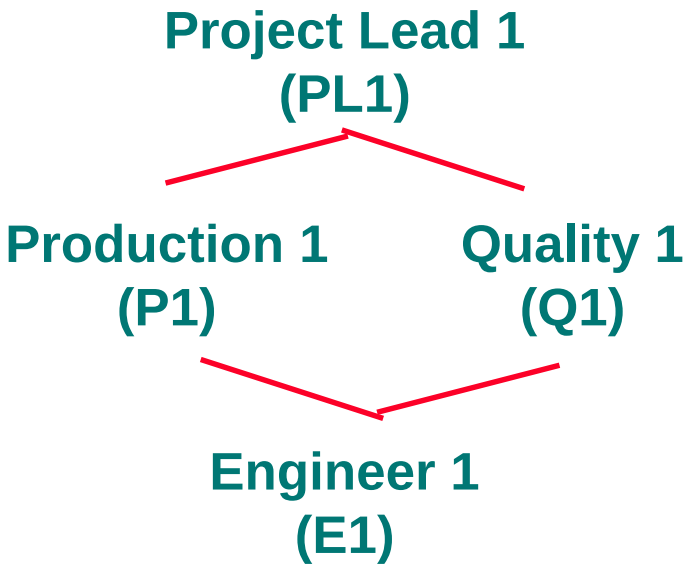
EXAMPLE ROLE HIERARCHY

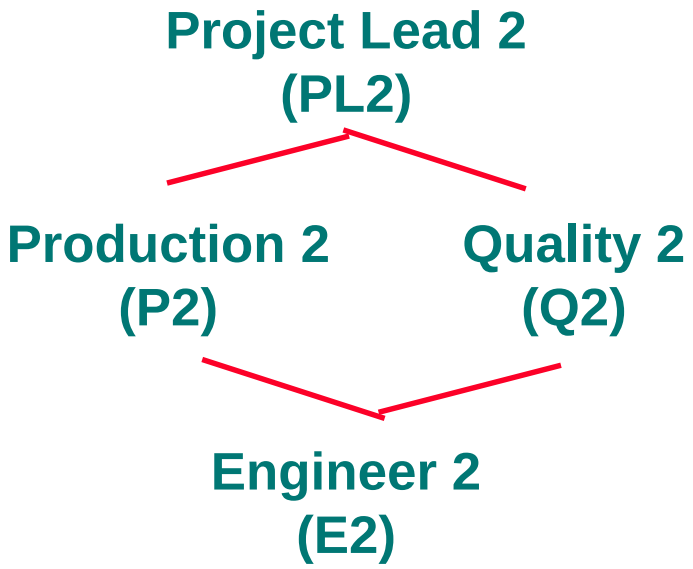


PROJECT 1

PROJECT 2

EXAMPLE ROLE HIERARCHY





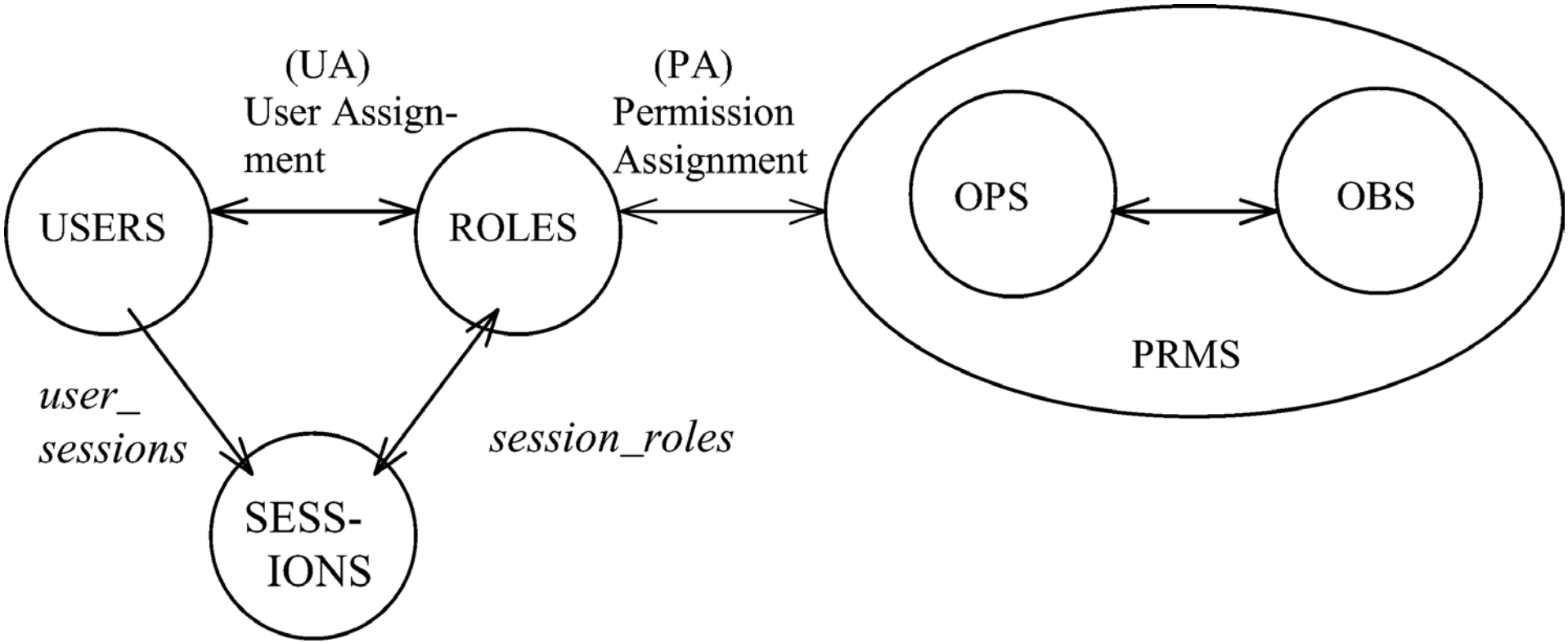
PROJECT 1

PROJECT 2

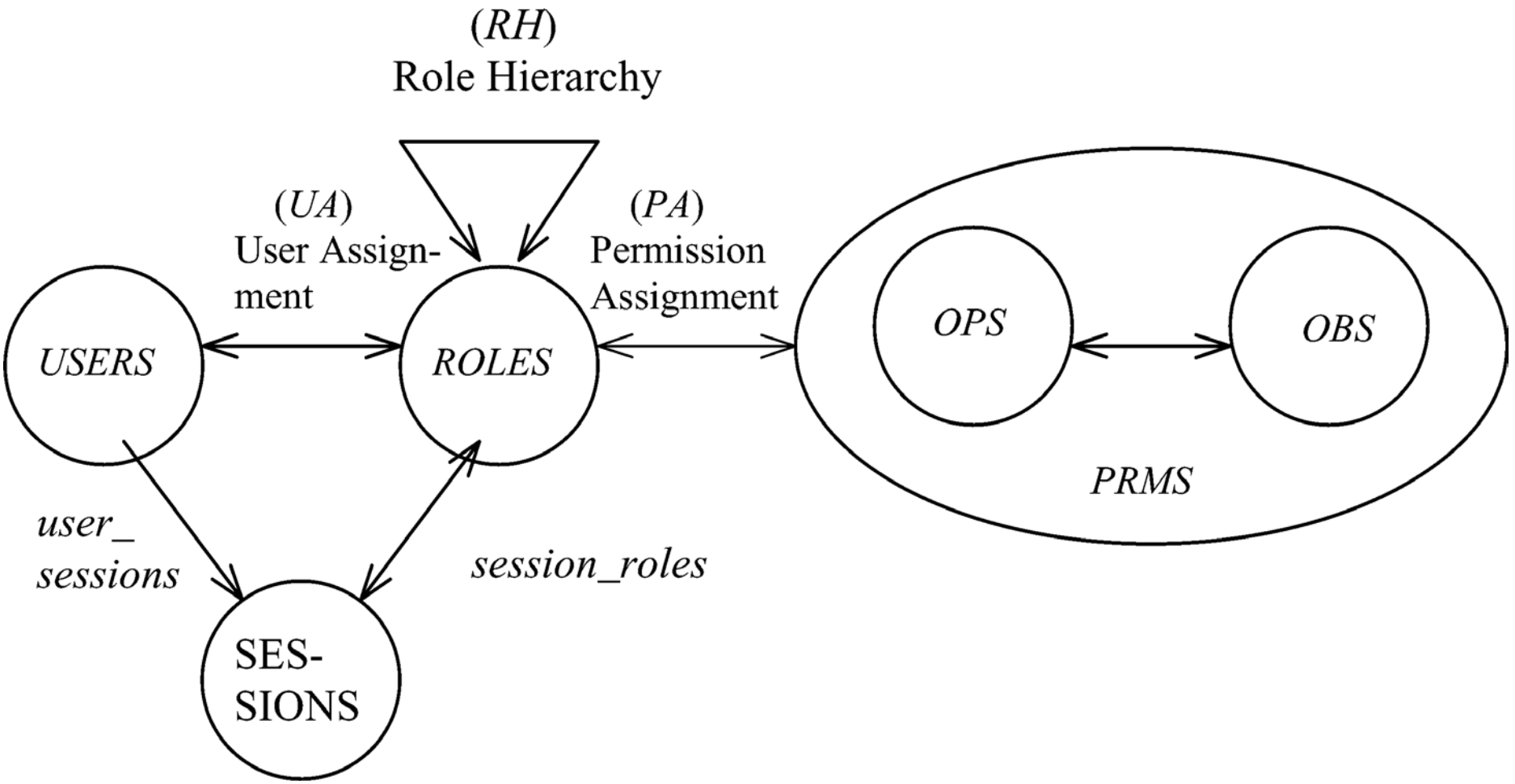
CONSTRAINTS

- Mutually Exclusive Roles
  - Static Exclusion: The same individual can never hold both roles
  - Dynamic Exclusion: The same individual can never hold both roles in the same context
- Mutually Exclusive Permissions
  - Static Exclusion: The same role should never be assigned both permissions
  - Dynamic Exclusion: The same role can never hold both permissions in the same context
- Cardinality Constraints on User-Role Assignment
  - At most k users can belong to the role
  - At least k users must belong to the role
  - Exactly k users must belong to the role
- Cardinality Constraints on Permissions-Role Assignment
  - At most k roles can get the permission
  - At least k roles must get the permission
  - Exactly k roles must get the permission

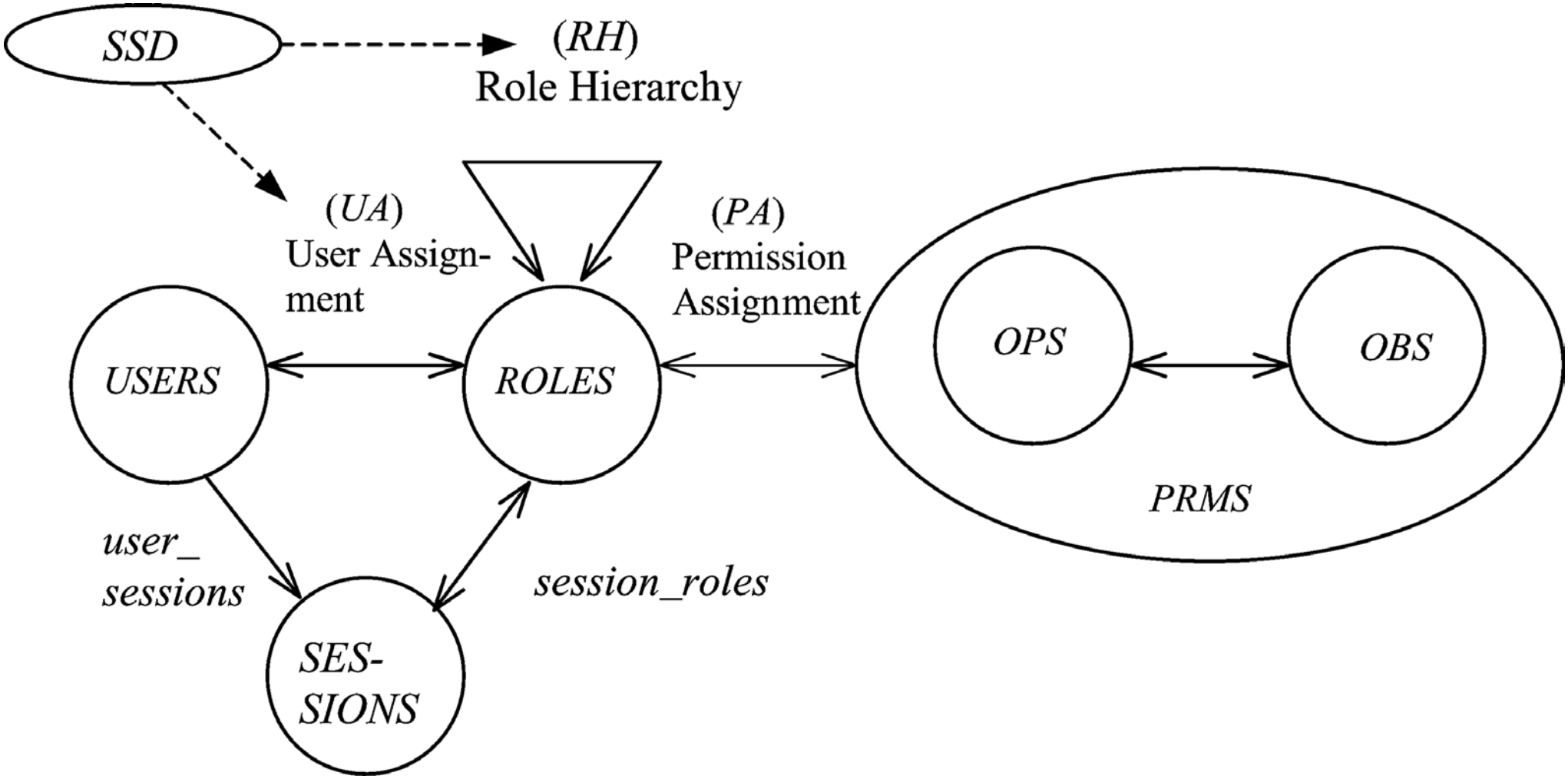
NIST MODEL: CORE RBAC



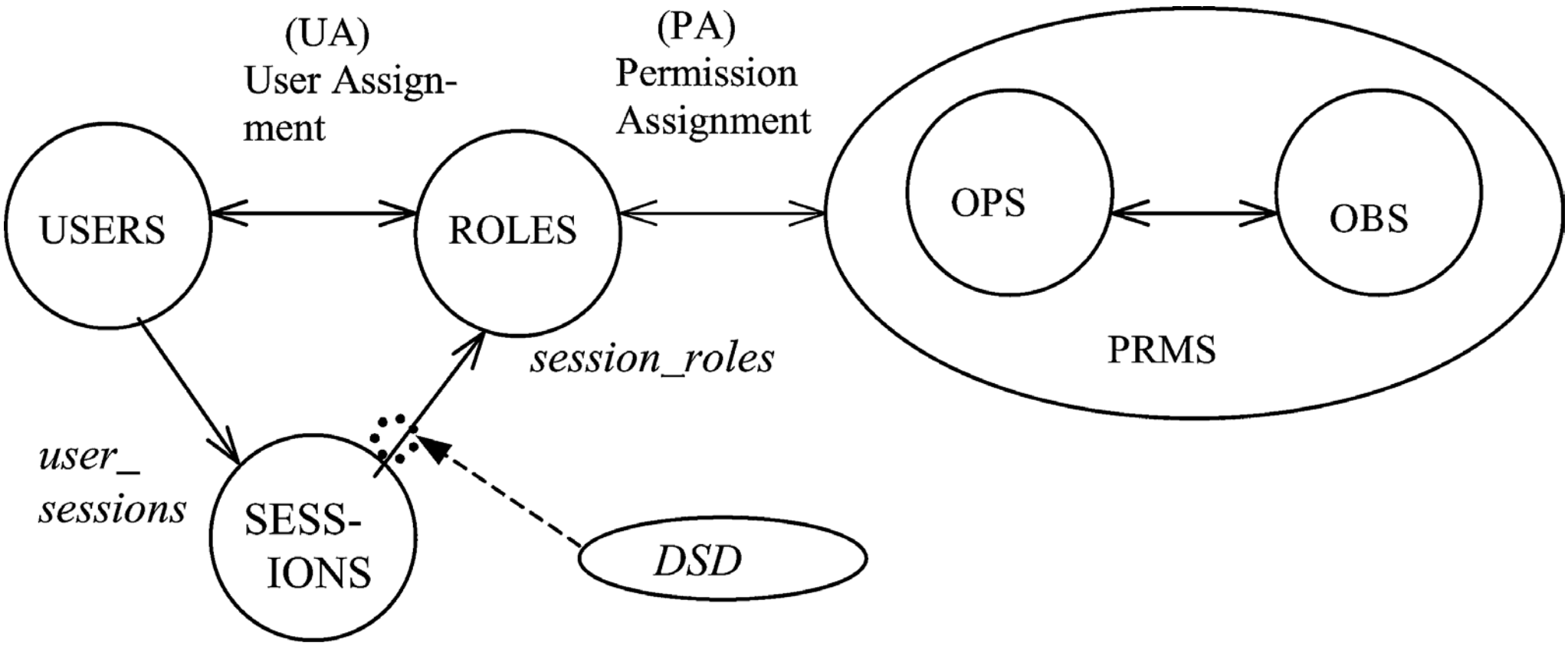
NIST MODEL: HIERARCHICAL RBAC



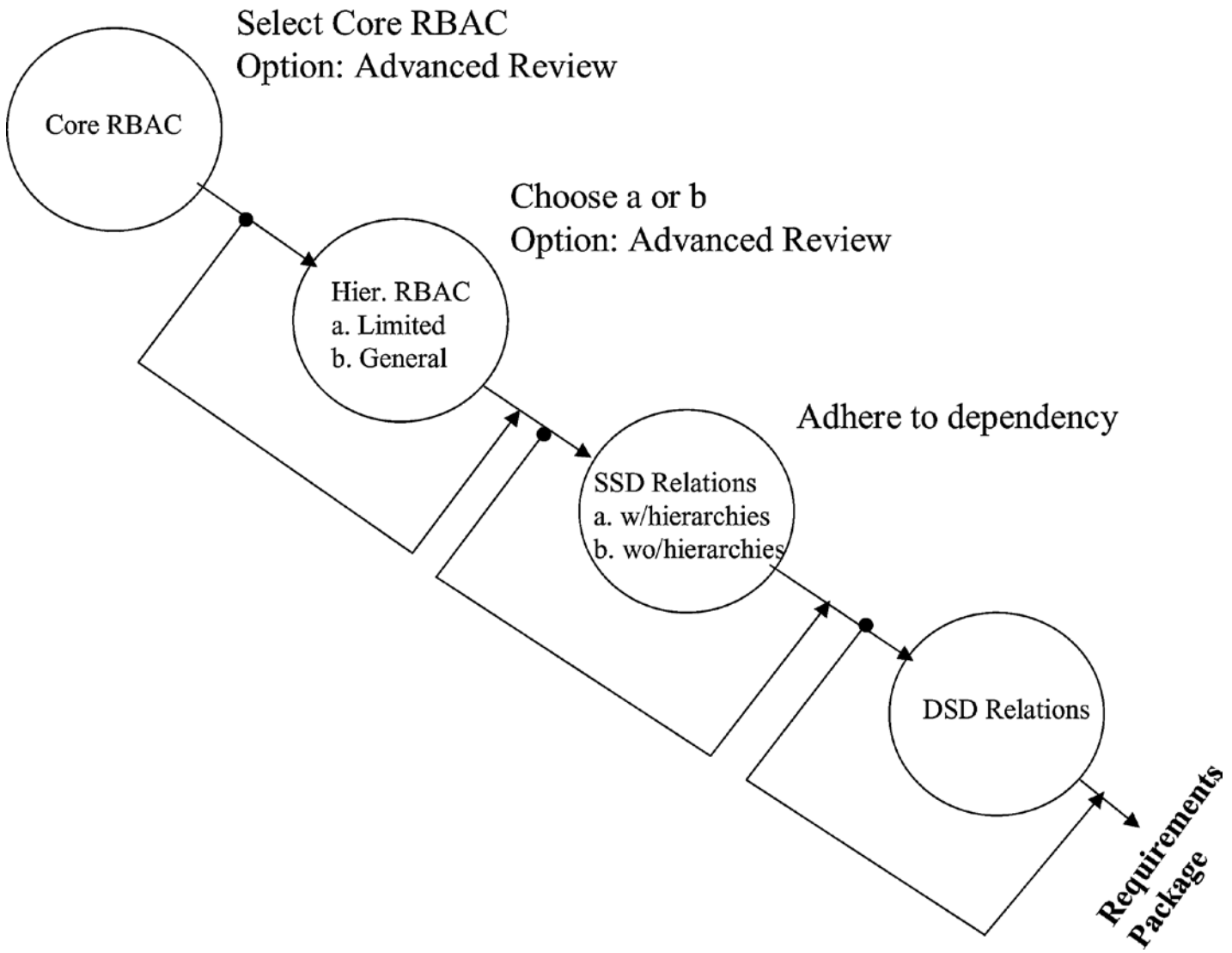
SSD IN HIERARCHICAL RBAC



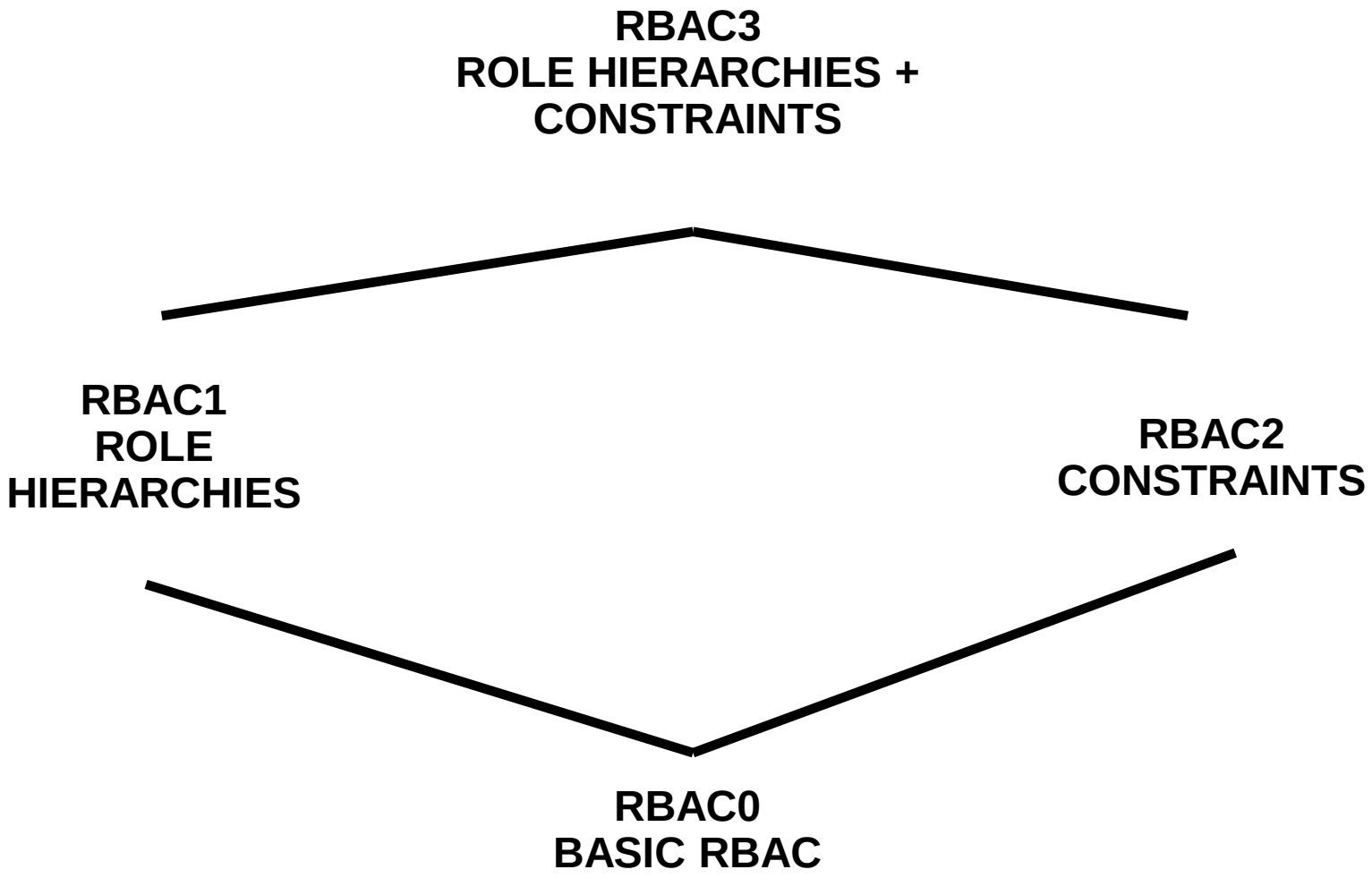
DSD IN HIERARCHICAL RBAC



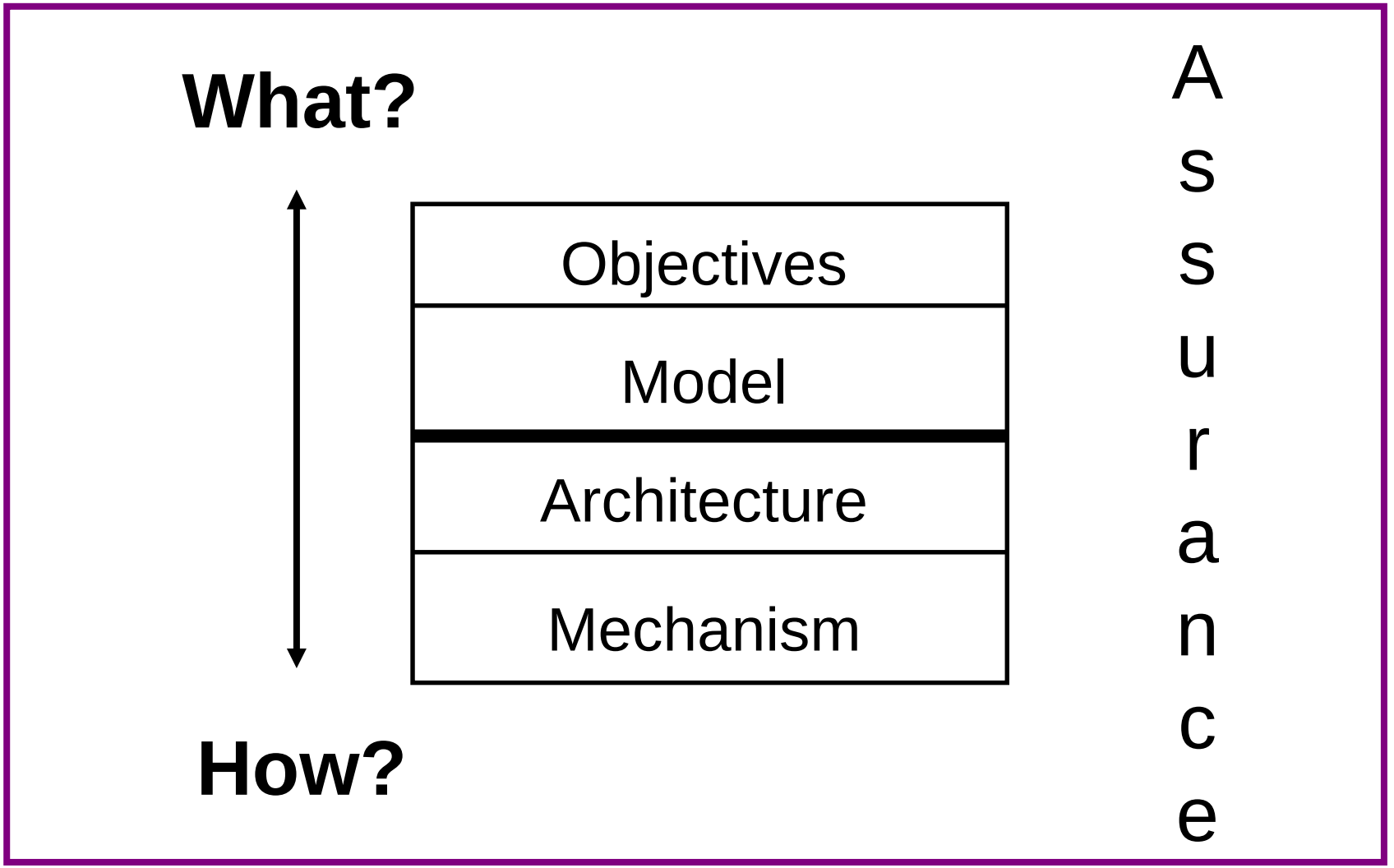
NIST MODEL FAMILY



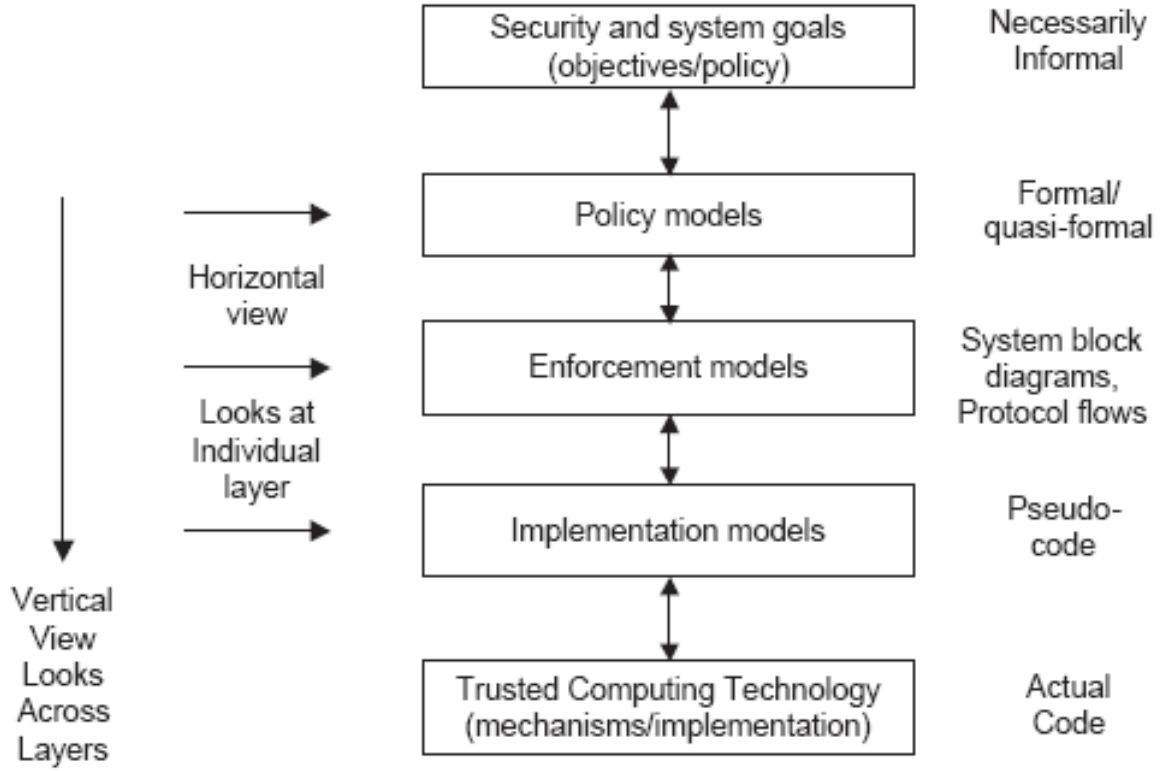
Compare RBAC96 Model Family



OM-AM


What?
Objectives
Model
Architecture
Mechanism
How?
Assurance

PEI Models



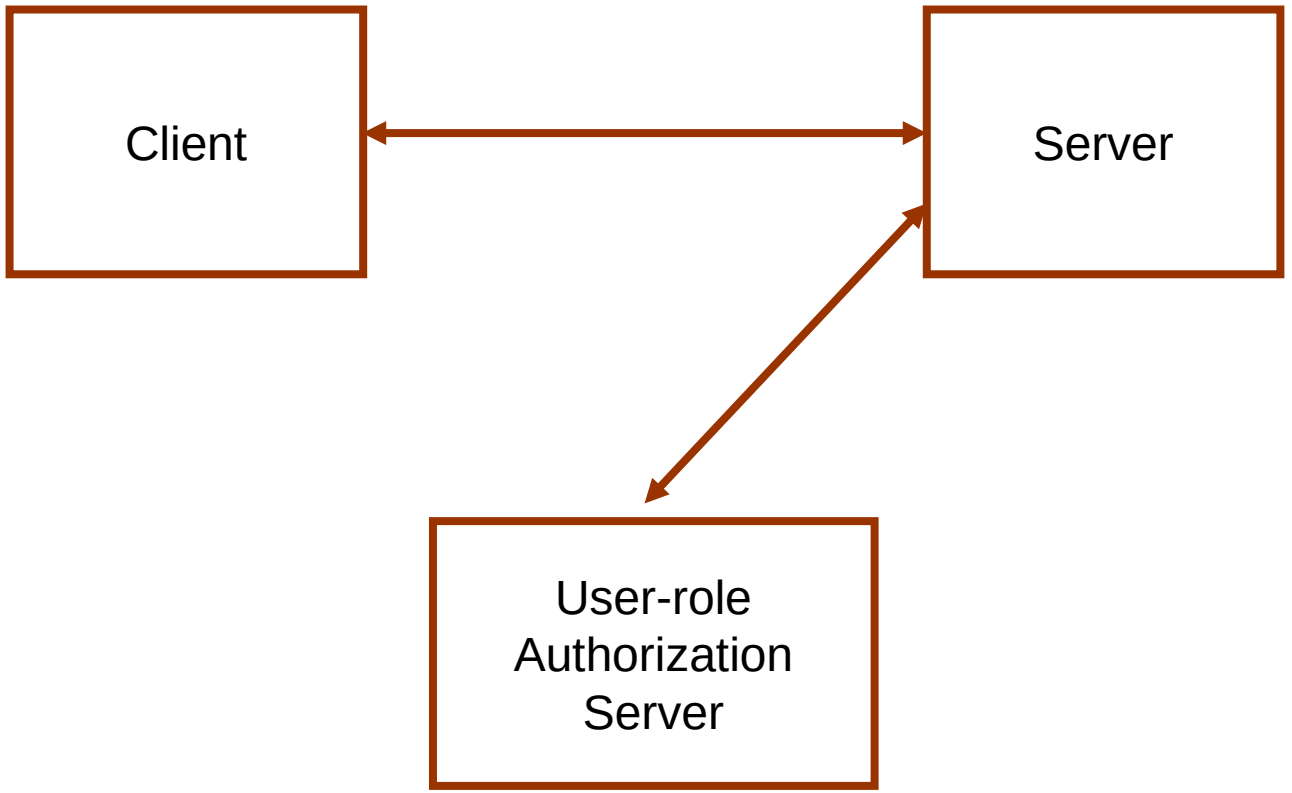
Idealized

Enforceable (Approximate)

Codeable

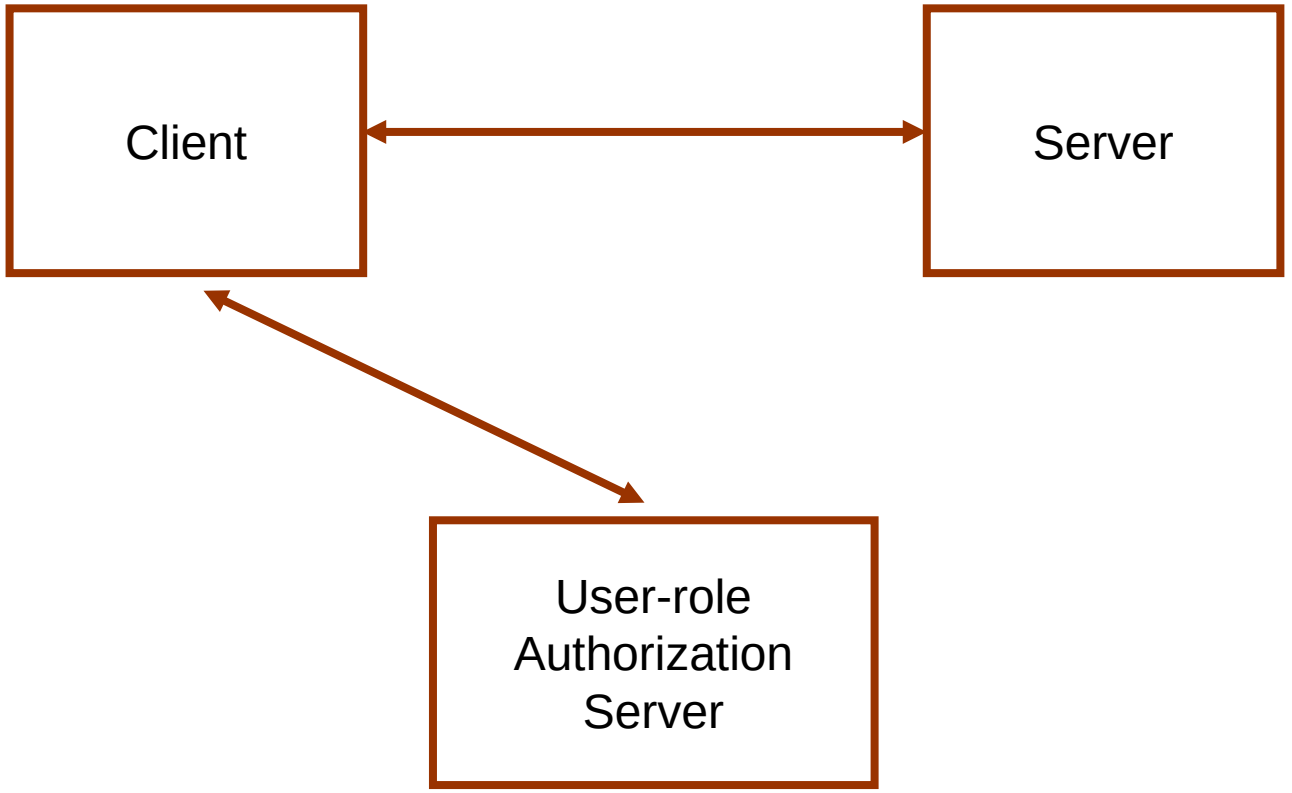
RBAC: SERVER PULL

E model



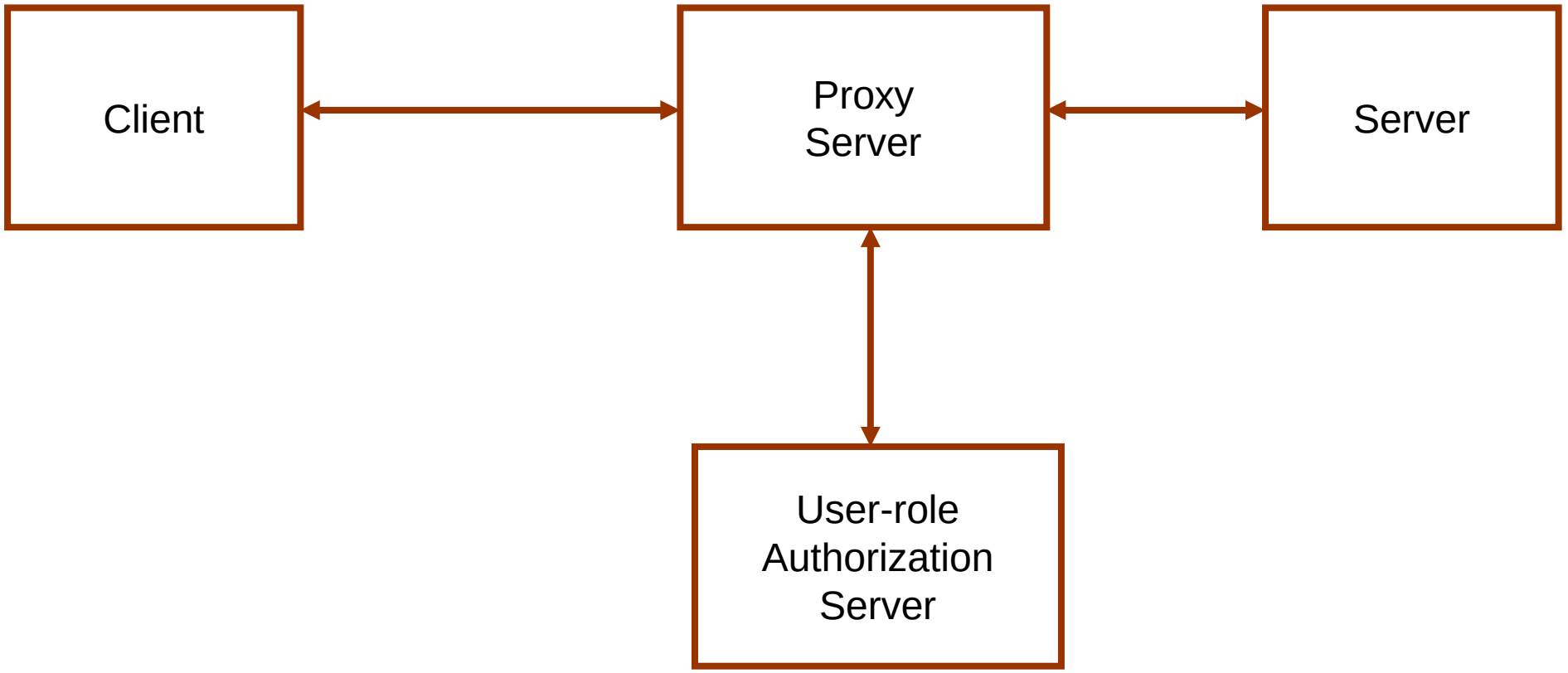
RBAC: CLIENT PULL

E model



RBAC: PROXY-BASED

E model



Discuss highlights of reference papers