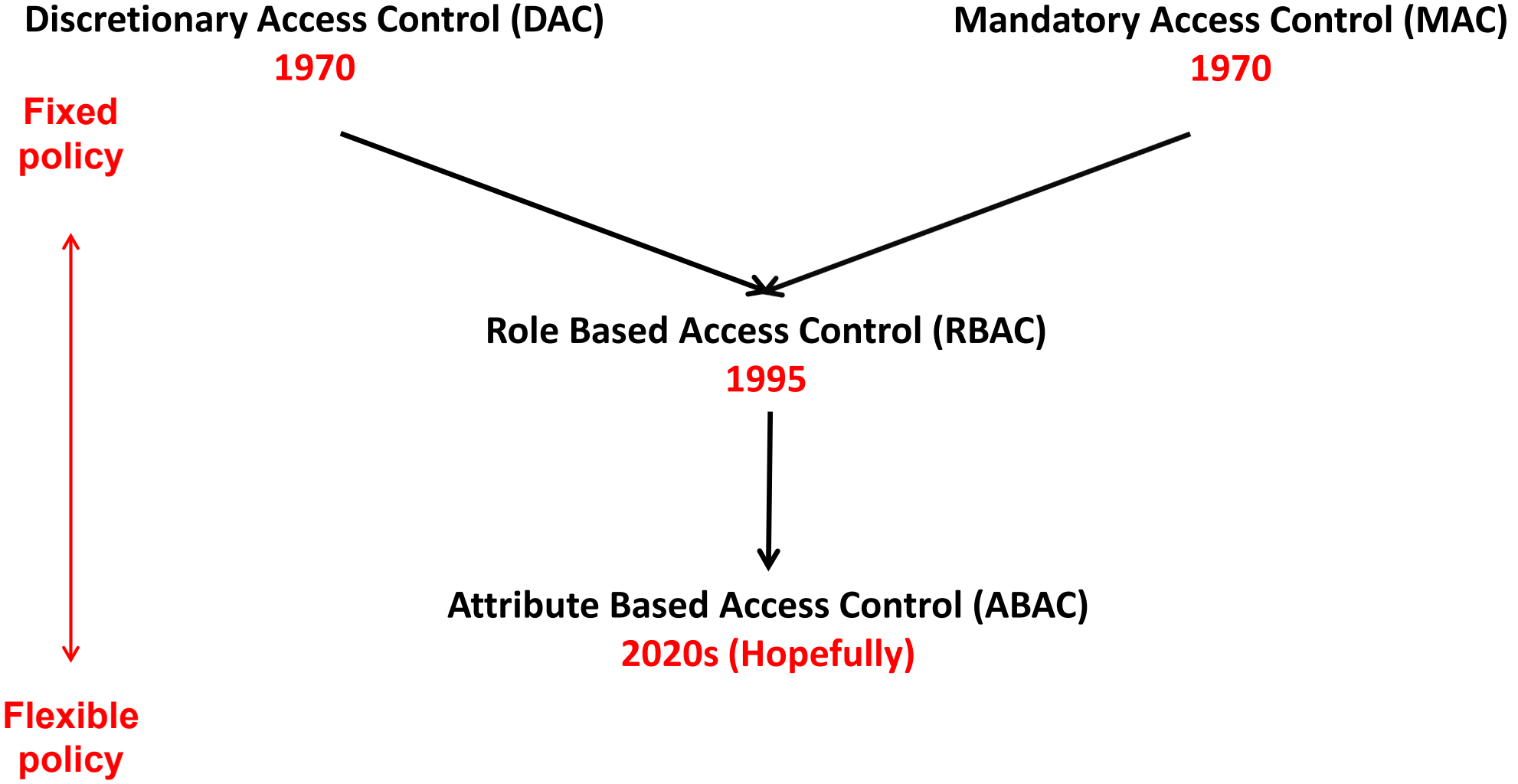
Module 4.1
Role-Based Access Control (RBAC)

Ravi Sandhu

Spring 2021

Access Control Models



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!

Role-Based Access Control (RBAC)

- Core concept:
  - All accesses are mediated through Roles
- Core drawback:
  - Roles are a natural concept for human users
  - Not so natural for:
    - Information objects
    - Smart objects (Internet of Things)
    - Contextual attributes
- Sophistication:
  - Role hierarchies
  - Role constraints

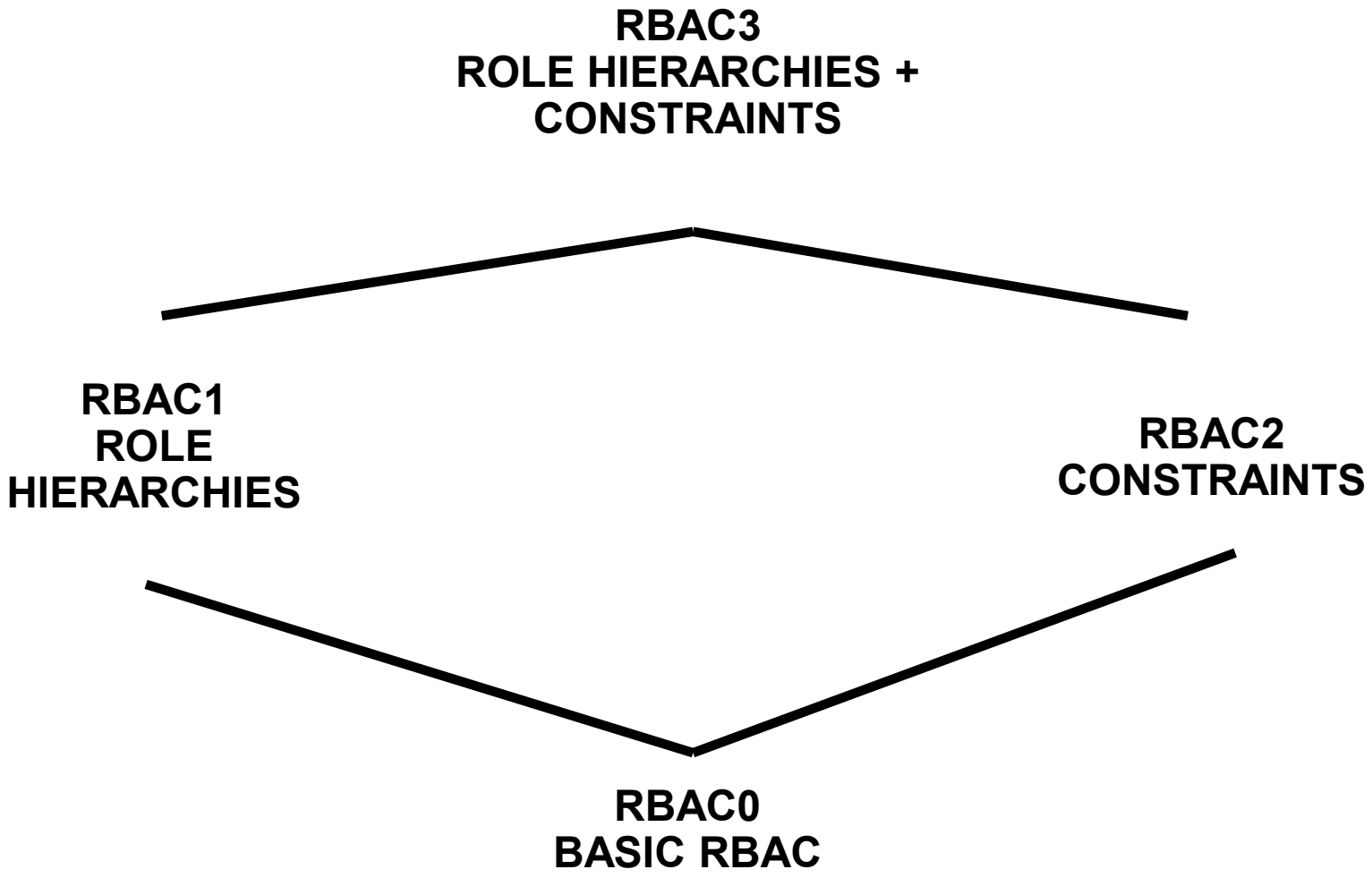
RBAC

- Operational model — **Our RBAC focus**
  - specify the decision function for the access decision triple or quad
- Administrative — **DAC, RBAC, …**
  - specify the model's dynamics
  - dynamics change the system state and modify the outcome of some access decision triple or quads

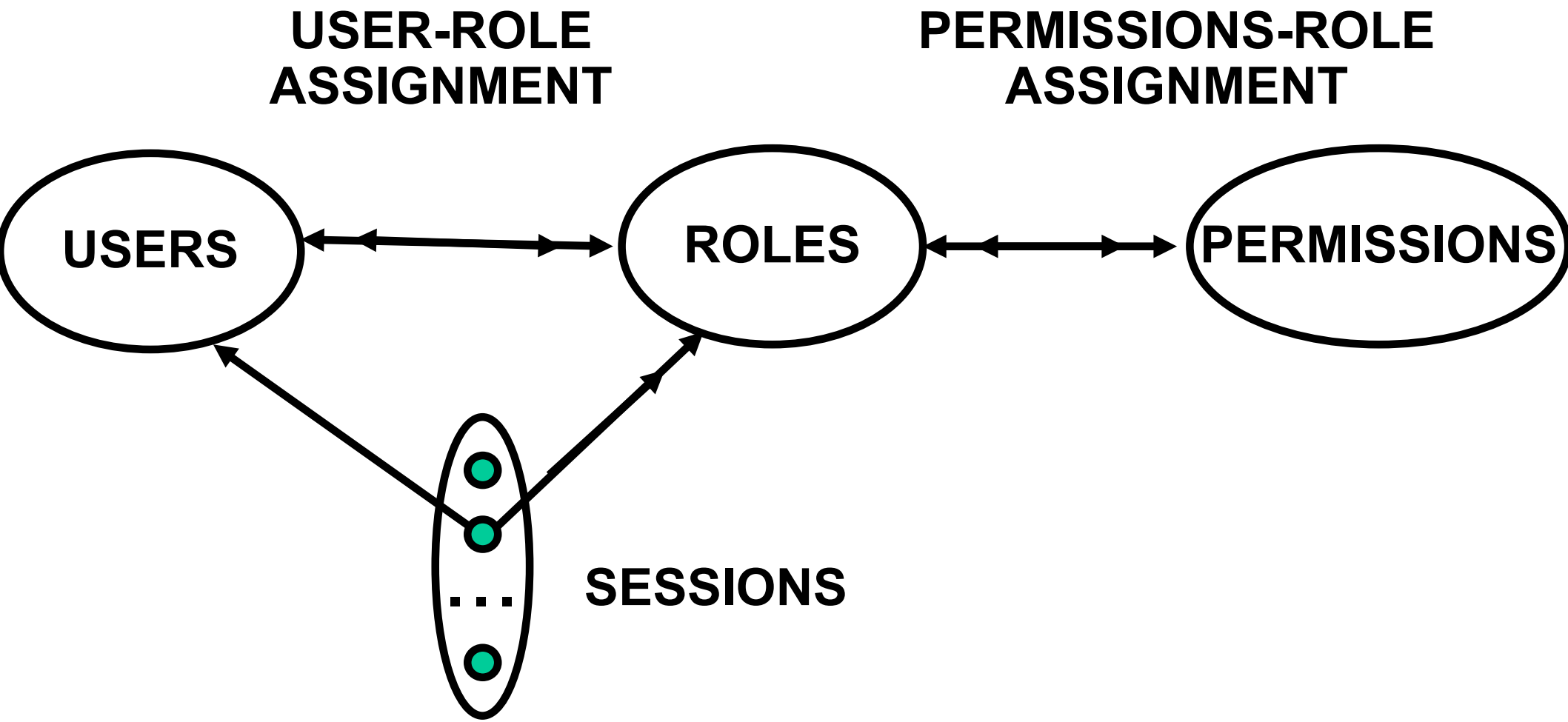
Founding Principles of RBAC96

- **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
  - Limit roles of a user
  - Limit users in a role
- **Separation of Duty**
  - Static separation: purchasing manager vs accounts payable manager
  - Dynamic separation: cash-register clerk versus cash-register manager

RBAC96 Model Family



RBAC96 Model Family: RBAC0

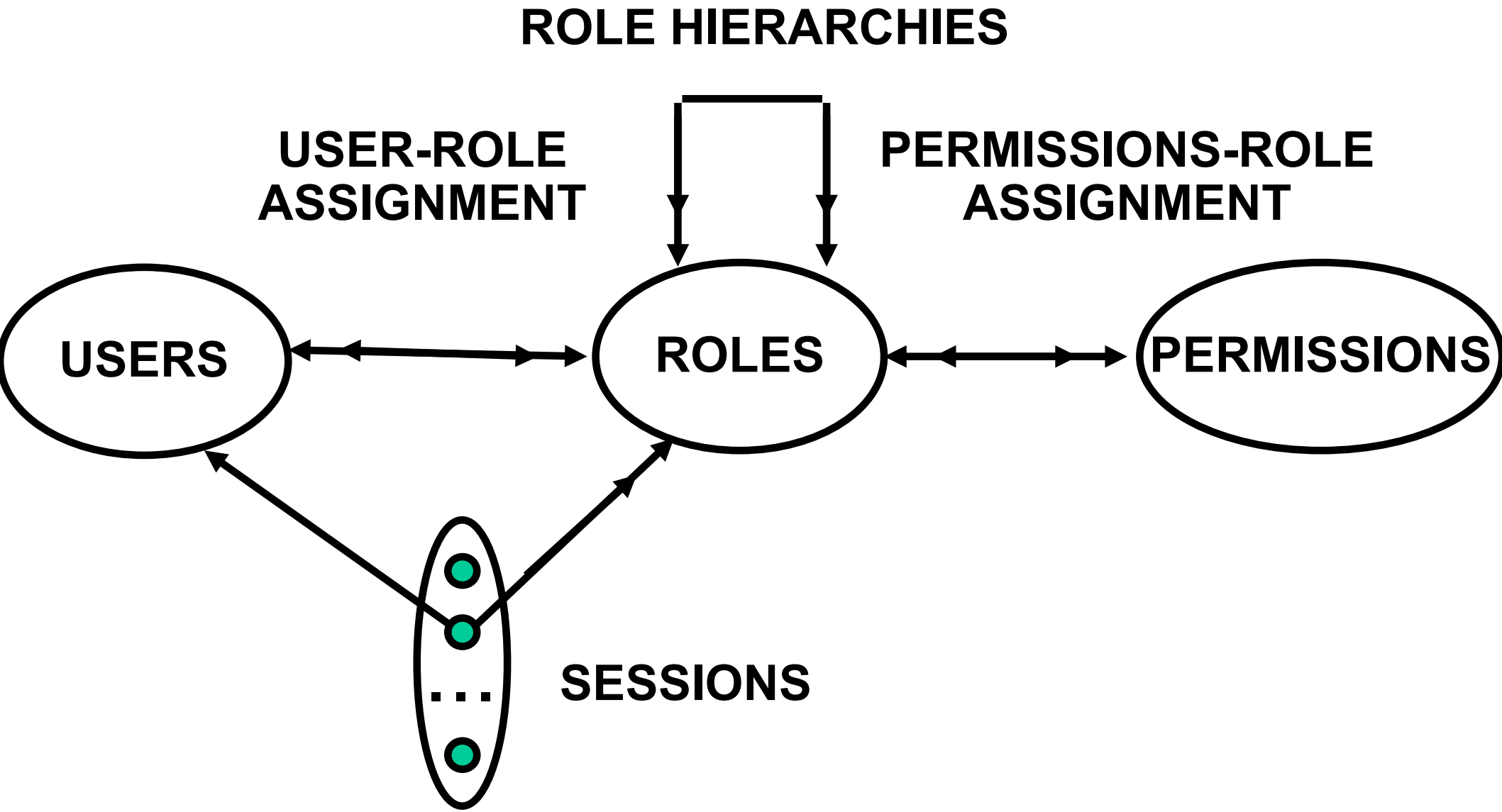


ROLES AS POLICY ANCHORS

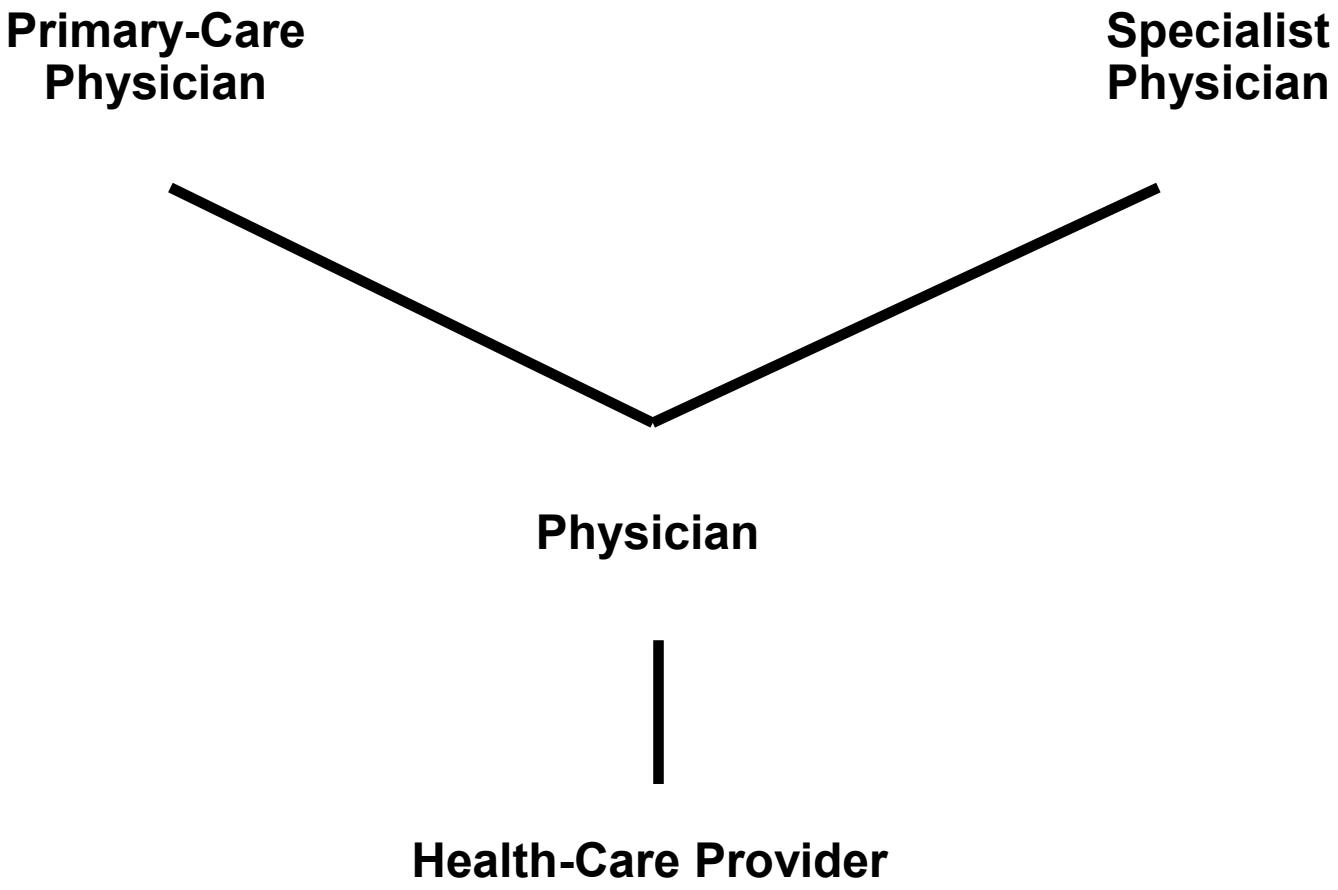
- A role brings together
  - a collection of users and
  - a collection of permissions
- These collections will vary over time
  - A role has significance and meaning beyond the particular users and permissions brought together at any moment

- Roles versus Operating System (OS) groups
  - Most OS's support groups as ACL entries
  - An OS group is a collection of users
  - Selective activation typically not supported

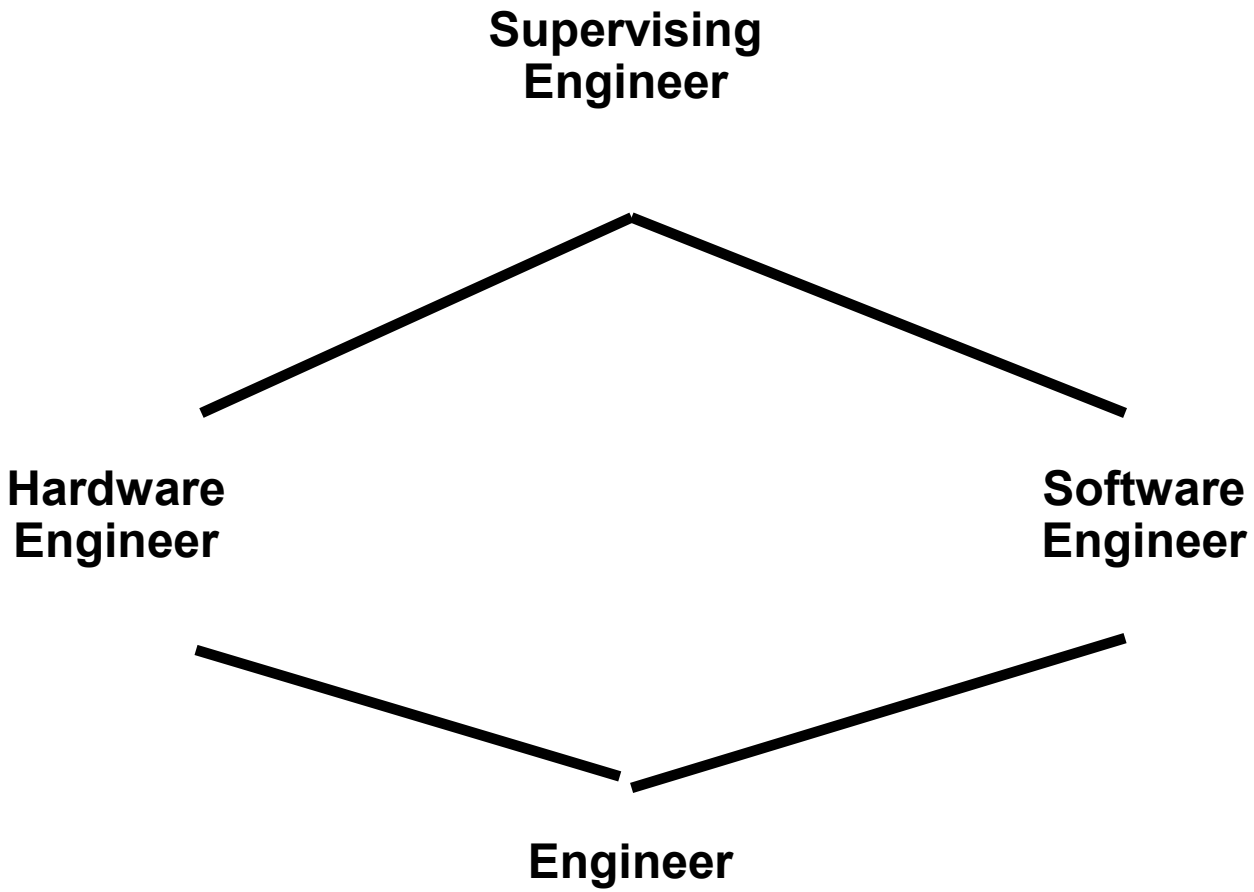
RBAC96 Model Family: RBAC1



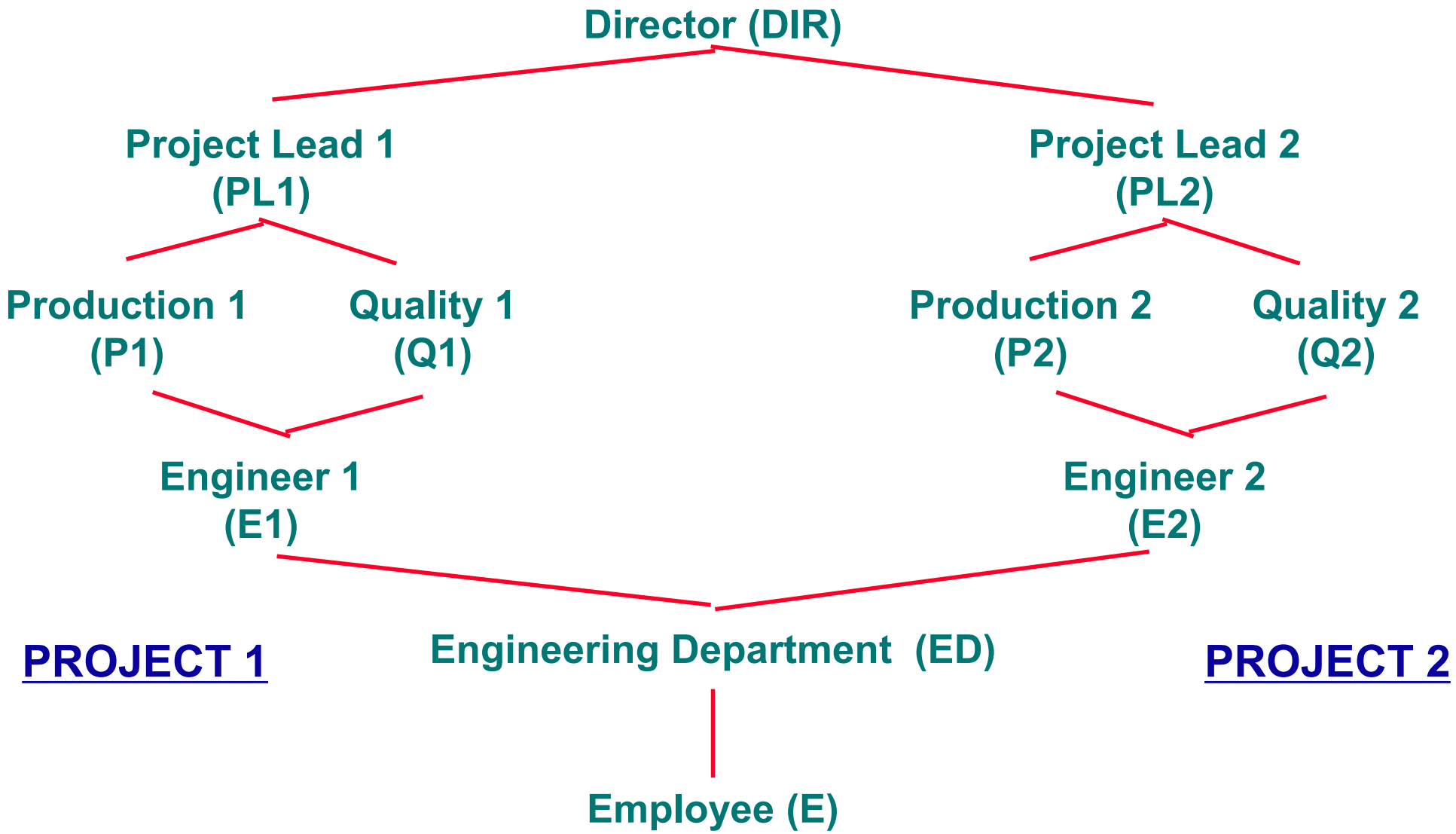
HIERARCHICAL ROLES



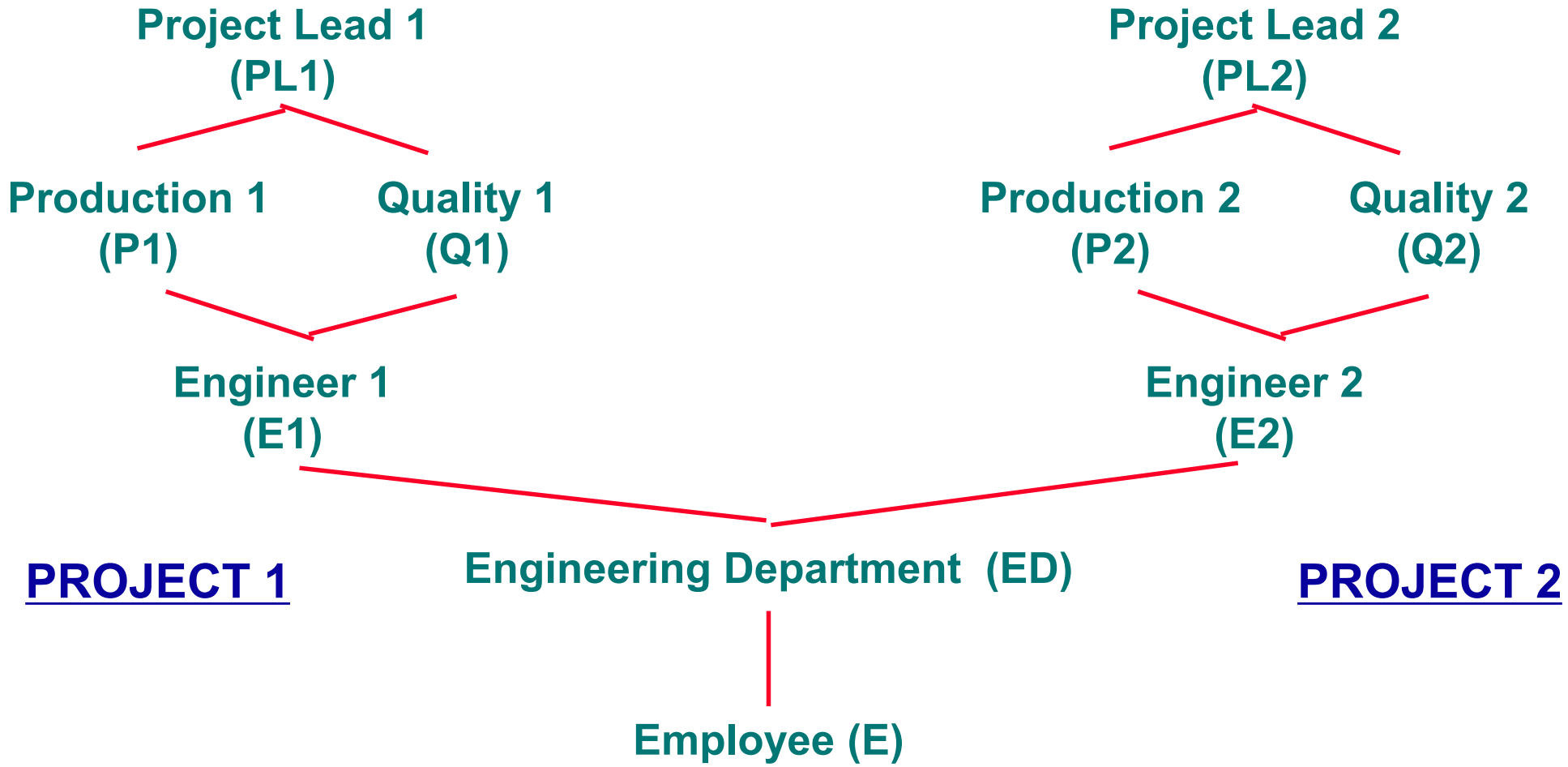
HIERARCHICAL ROLES



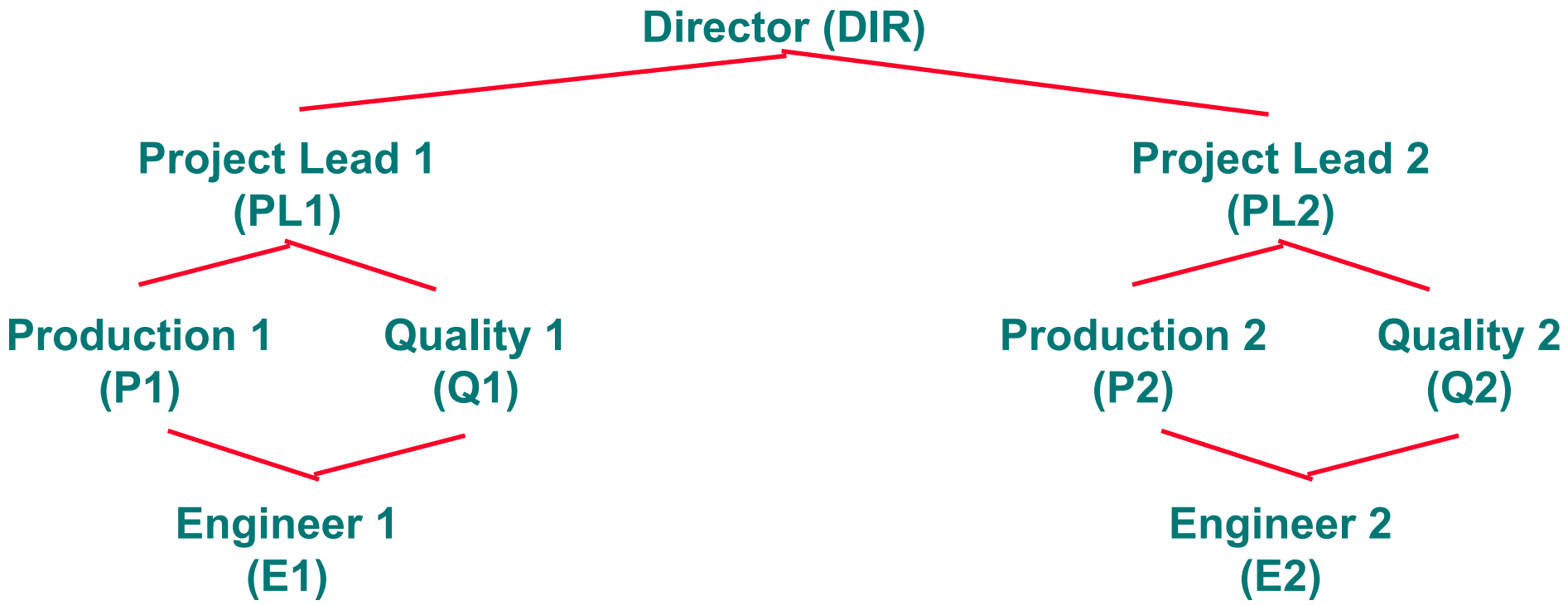
EXAMPLE ROLE HIERARCHY



EXAMPLE ROLE HIERARCHY

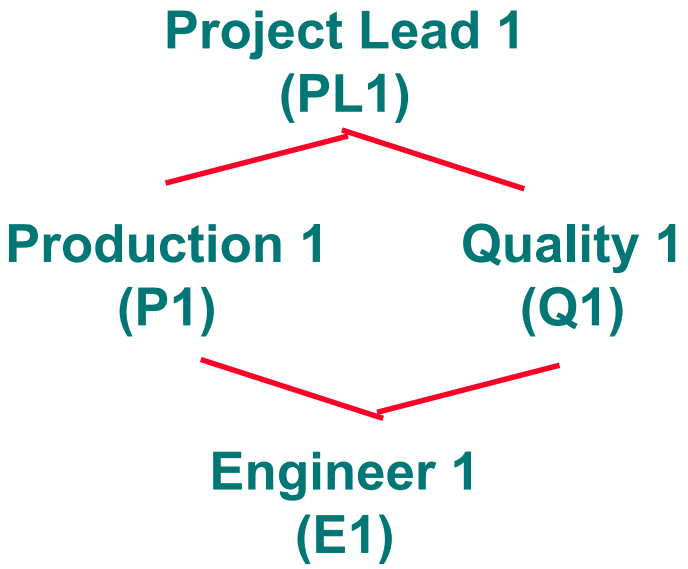


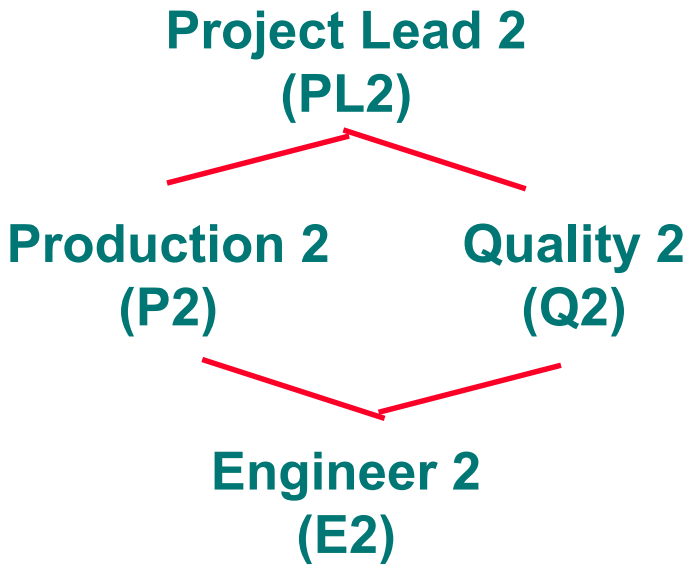
EXAMPLE ROLE HIERARCHY



PROJECT 1 **PROJECT 2**

EXAMPLE ROLE HIERARCHY

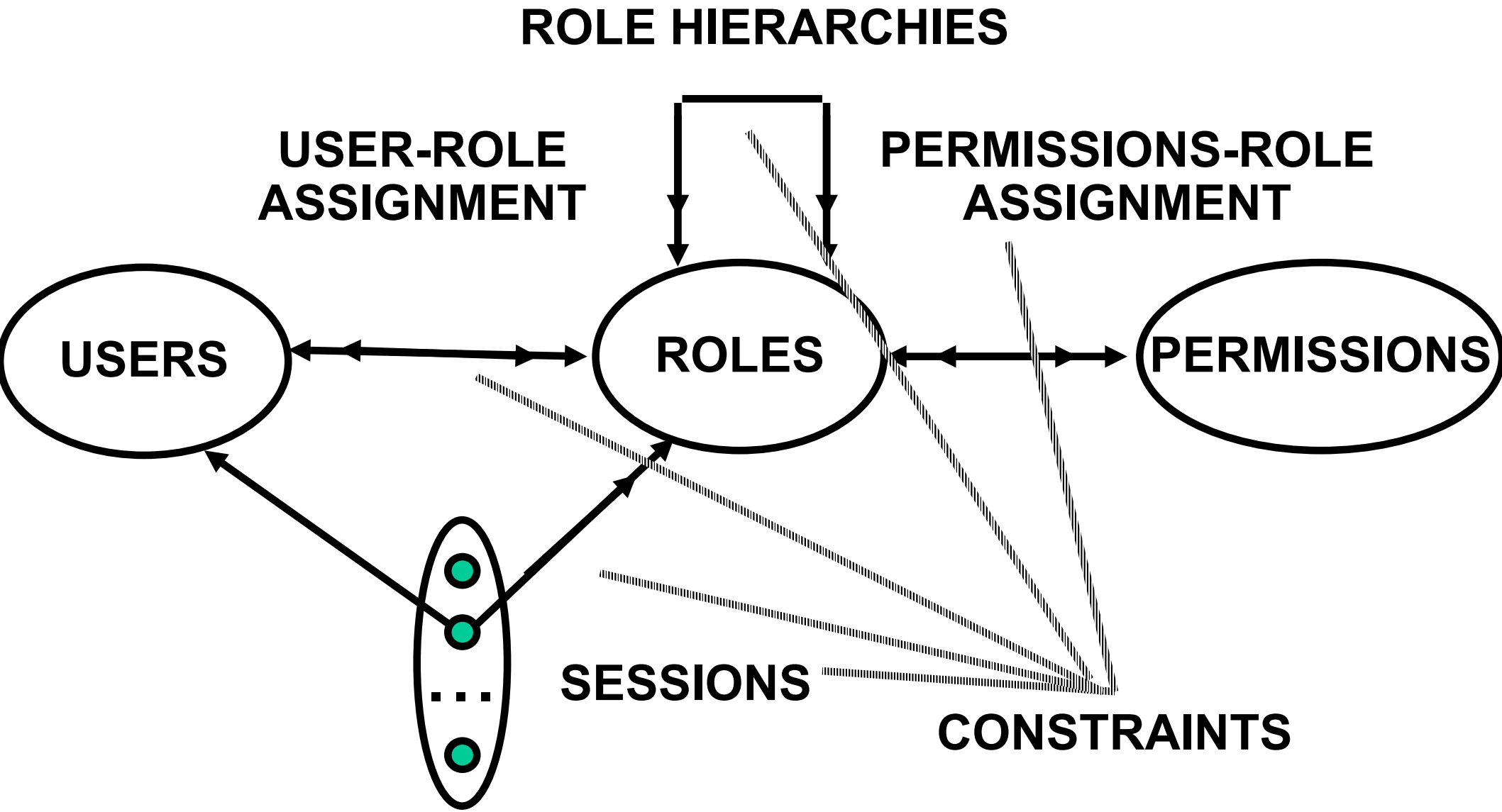




PROJECT 1

PROJECT 2

RBAC96 Model Family: RBAC3



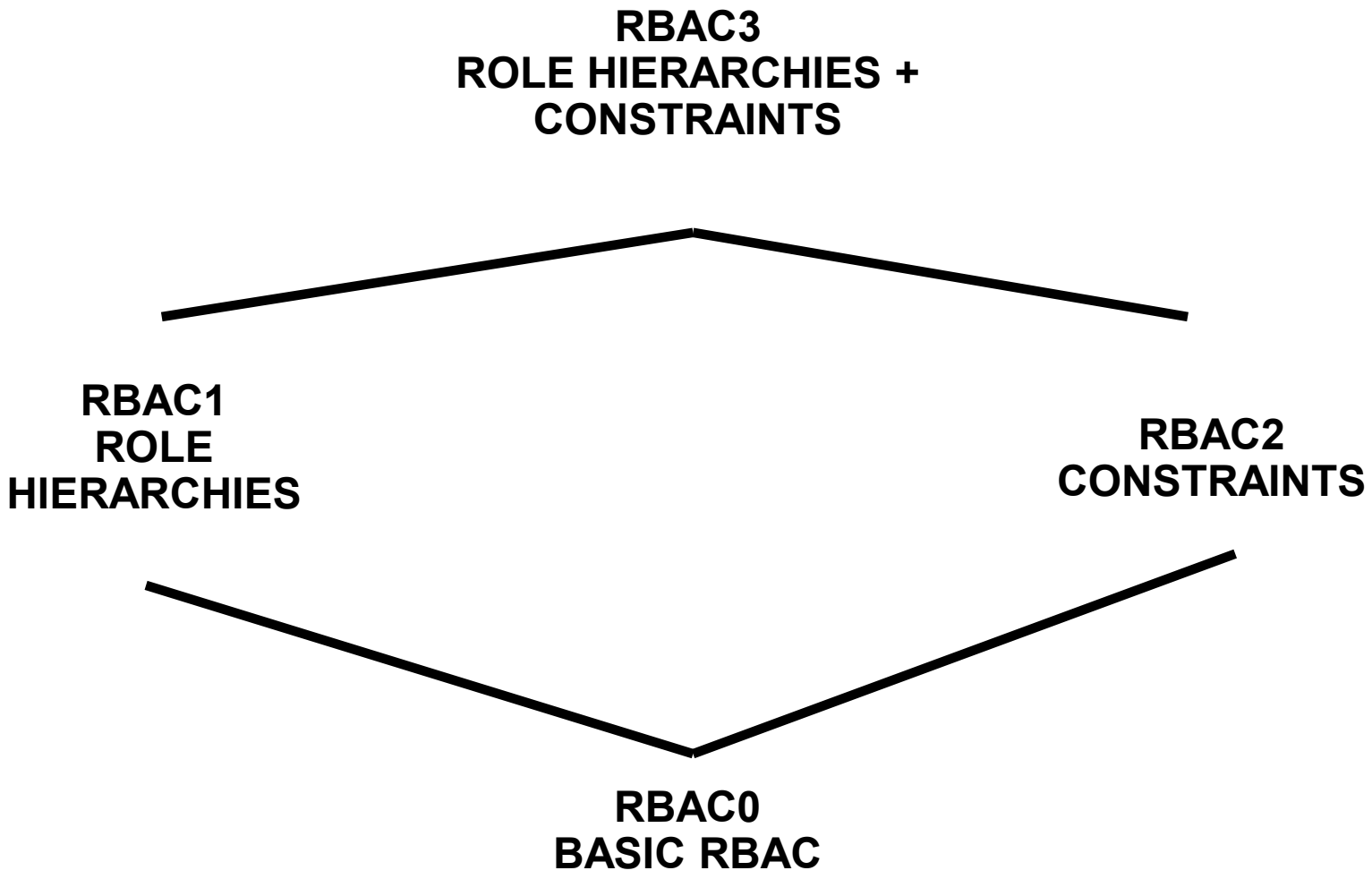
SEPARATION OF DUTY (SOD)

- Static Separation of Duty
  - The same individual can never hold both roles
  - Applies to User-Role Assignment
  - Example: Purchasing Manager, Accounts Payable Manager
- Dynamic Separation of Duty
  - The same individual can never hold both roles in the same session
  - Applies to Session-Role Activation
  - Example: Cash-Register Clerk, Cash-Register Manager
  - Example: Course-Teaching-Assistant, Course-Student

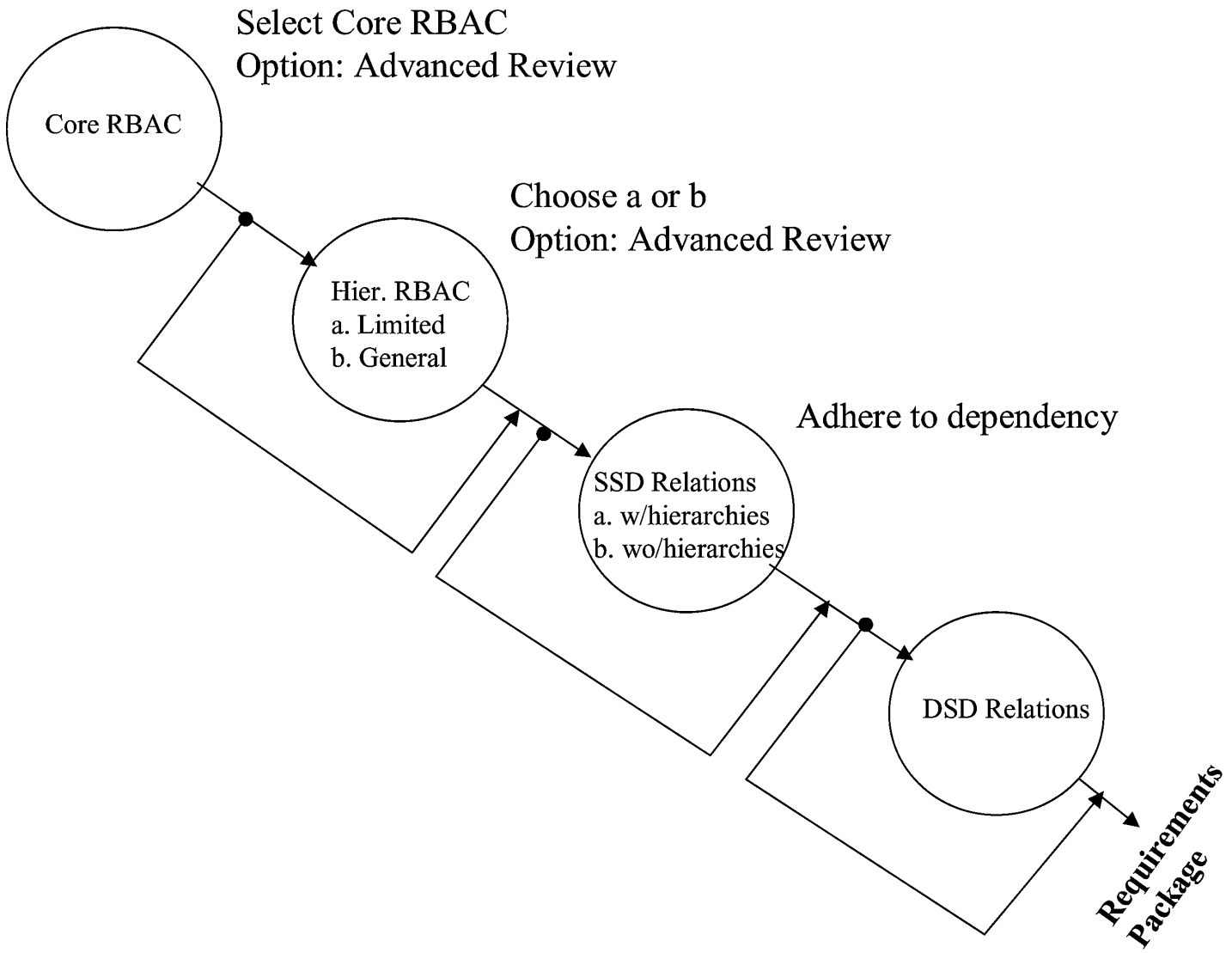
CARDINALITY CONSTRAINTS

- 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

RBAC96 Model Family



NIST MODEL FAMILY



ROLE HIERARCHIES