CS 547: Foundation of Computer Security

S. Tripathy
IIT Patna

Previous Class

- Access Control
  - Discretionary Access Control
  - Mandatory Access Control

Present class

- Access Control
  - Role-Based Access Control

Limitations with BLP

- Deal only with confidentiality, does not deal with integrity at all

  - Addressed by integrity models (such as Biba, Clark-Wilson)

- Does not deal with information flow through covert channels

What is integrity?

- Attempt 1: Critical data do not change.
- Attempt 2: Critical data changed only in "correct ways"
  - E.g., in DB, integrity constraints are used for consistency
- Attempt 3: Critical data changed only through certain "trusted programs"
- Attempt 4: Critical data changed only as intended by authorized users.

The Biba Model

- Kenneth J. Biba: "Integrity Considerations for Secure Computer Systems", MTR-3153, The Mitre Corporation, April 1977.

- Motivated by the fact that BLP does not deal with integrity

Biba: Integrity Levels

- Each subject (program) has an integrity level
- Each object has an integrity level
- Integrity levels are totally ordered

- Integrity levels different from security levels in confidentiality protection
  - a highly sensitive data may have low integrity

Five Mandatory Policies in Biba

- Strict integrity policy
- Subject low-water mark policy
- Object low-water mark policy
- Low-water mark Integrity audit policy
- Ring policy

Strict Integrity Policy (BLP reversed)

- Rules:
  - s can read o iff $i(s) \leq i(o)$
    - no read down
    - stops indirect sabotage by contaminated data
  - s can write to o iff $i(s) \geq i(o)$
    - no write up
    - stops directly malicious modification

- Fixed integrity levels

Subject Low-Water mark Policy

- Rules
  - s can always read o; after reading
    $i(s) \leftarrow \min[i(s), i(o)]$
  - s can write to o iff $i(s) \geq i(o)$

- Subject's integrity level decreases as reading lower integrity data

Object Low-Water Mark Policy

- Rules
  - s can read o; iff $i(s) \leq i(o)$
  - s can always write to o; after writing
    $i(o) \leftarrow \min[i(s), i(o)]$
- Object's integrity level decreases as it is contaminated by subjects

Low-Water Mark Integrity Audit Policy

- Rules
  - s can always read o; after reading
    $i(s) \leftarrow \min[i(s), i(o)]$
  - s can always write to o; after writing
    $i(o) \leftarrow \min[i(s), i(o)]$

The Ring Policy

- Rules
  - Any subject can read any object
  - s can write to o iff $i(s) \geq i(o)$
- Integrity levels of subjects and objects are fixed.
- Intuitions:
  - subjects are trusted to process low-level inputs correctly

Object Integrity Levels

- The integrity level of an object may be based on
  - Quality of information (levels may change)
    - Degree of trustworthiness
    - Contamination level:
  - Importance of the object (levels do not change)
    - degree of being trusted
    - Protection level: writing to the objects should be protected

- What should the relation between the two meanings, which one should be higher?

Integrity vs. Confidentiality

Confidentiality	Integrity
Control reading • preserved if confidential info is not read	Control writing • preserved if important obj is not changed
For subjects who need to read, control writing after reading is sufficient, no need to trust them	For subjects who need to write, has to trust them, control reading before writing is not sufficient

Advantages and Disadvantages

- Advantages:
  - The Biba model is simple and easy to implement.
  - The Biba model provides a number of different policies that can be selected based on need.
- Disadvantages:
  - The model does nothing to enforce confidentiality.
  - The Biba model doesn't support the granting and revocation of authorization.
  - To use this model all computers in the system must support the labeling of integrity for both subjects and objects. To date, there is no network protocol that supports this labeling. So there are problems with using the Biba model in a network environment.

MAC Pattern Known Uses

- Security-Enhanced Linux (SELinux) kernel
  - enforces theMAC pattern to implement a flexible and fine-grained MAC architecture called Flask (Flux advanced security kernel) which operates independently of the traditional Linux access control mechanisms.

- TrustedBSD developed by the FreeBSD
  - TrustedBSD contains modules that implement MLS (Multi-Level Security) and fixed-label Biba integrity policies which is a variant of the MAC pattern.

- GeSWall (General Systems Wall) is the Windows security project developed by GeSWall. It implements the MAC pattern to provide OS integrity and data confidentiality transparent and invisible to user.

MAC Pattern Consequences

Advantages:

- The MAC pattern facilitates enforcing access control policies based on security levels.
- The assignment of a classification and category to users and objects is centralized by a mediator.

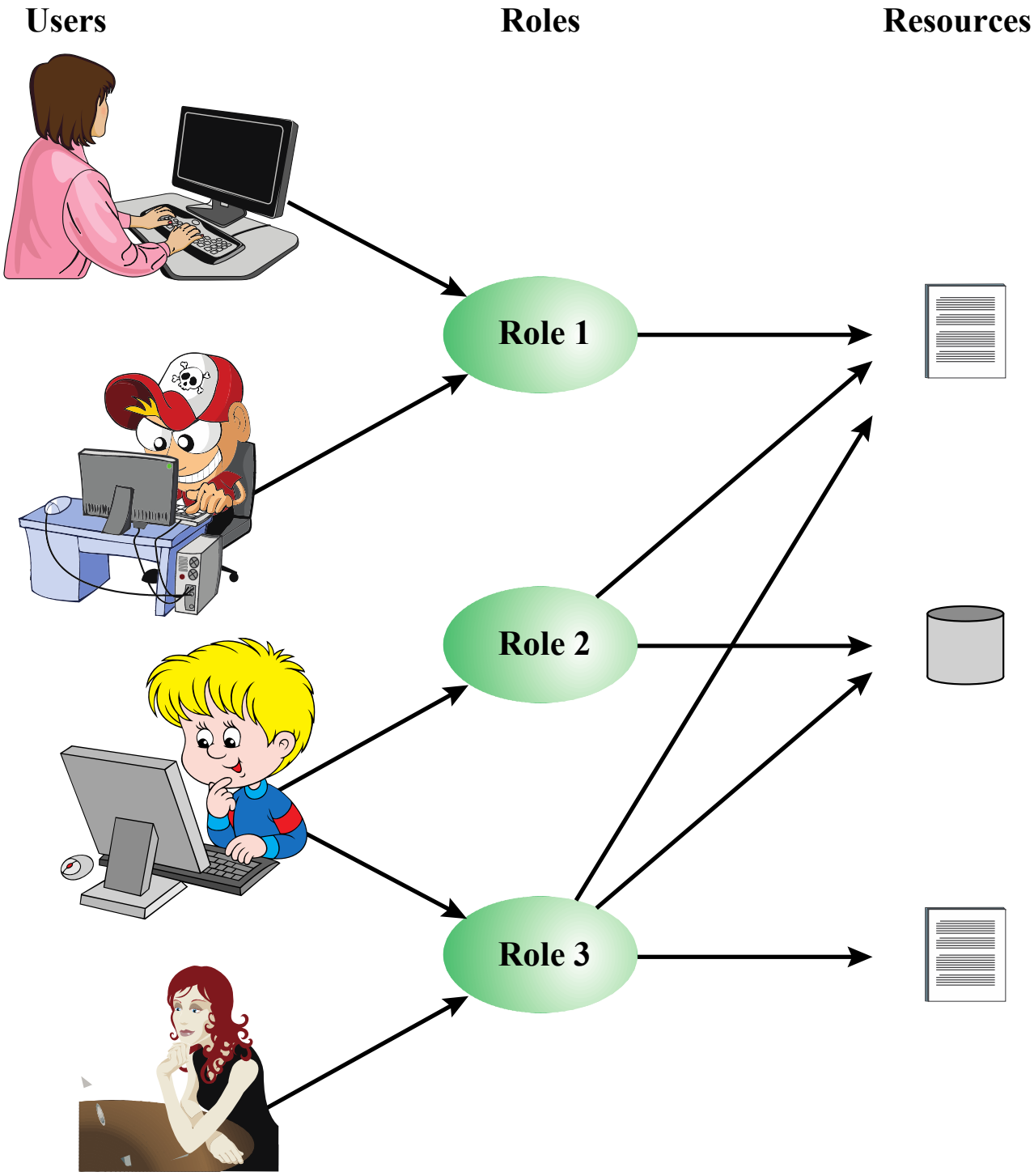
Disadvantages:

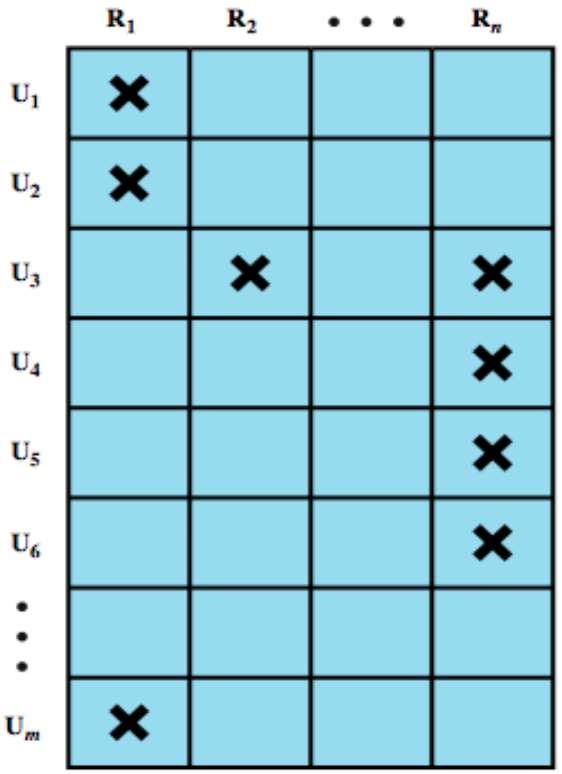
- Introducing a new object or user requires a careful assignment of a classification and category.
- The mediator who assigns classifications to users and objects should be a trusted person.

Role-based Access Control (RBAC)

- In a company, objects that a user can access often do not depend on the identity of the user, but on the user's job function (role) within the company
  - Salesperson can access customers' credit card numbers, marketing person only customer names
- In RBAC, administrator assigns users to roles and grants access rights to roles
- When a user takes over new role, need to update only her role assignment, not all her access rights
- Available in many commercial databases

Role-Based Access Control (RBAC)

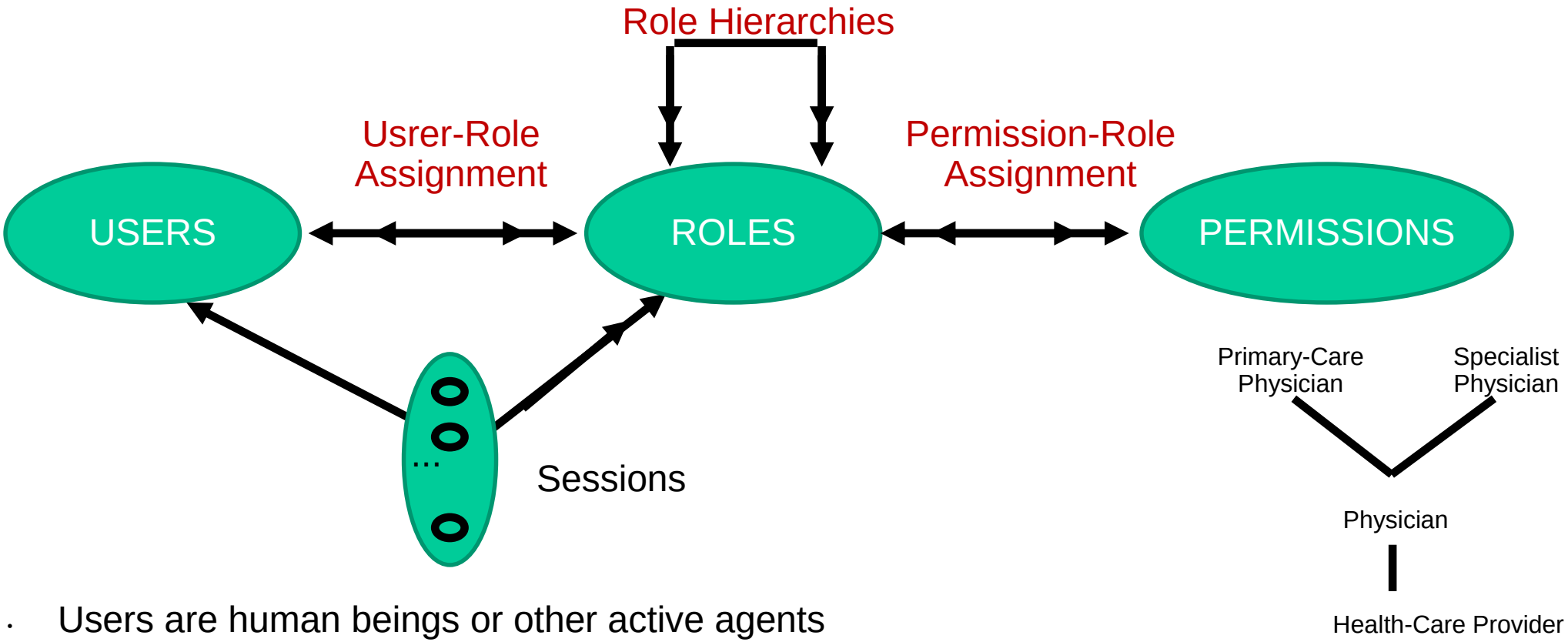




Access Control Matrix

ROLES \ OBJECTS	R_1	R_2	R_n	F_1	F_1	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			

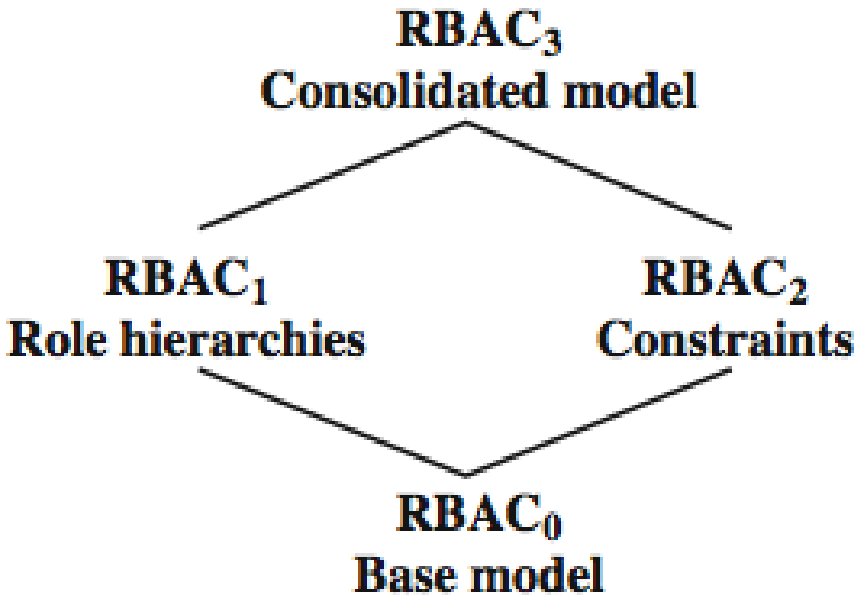
Role-Based Access Control



- Users are human beings or other active agents
- Business function the user perform is role
- A user can be a member of many roles
- Each role can have many users as members
- A user can invoke multiple sessions
- In each session a user can invoke any subset of roles that the user is a member of

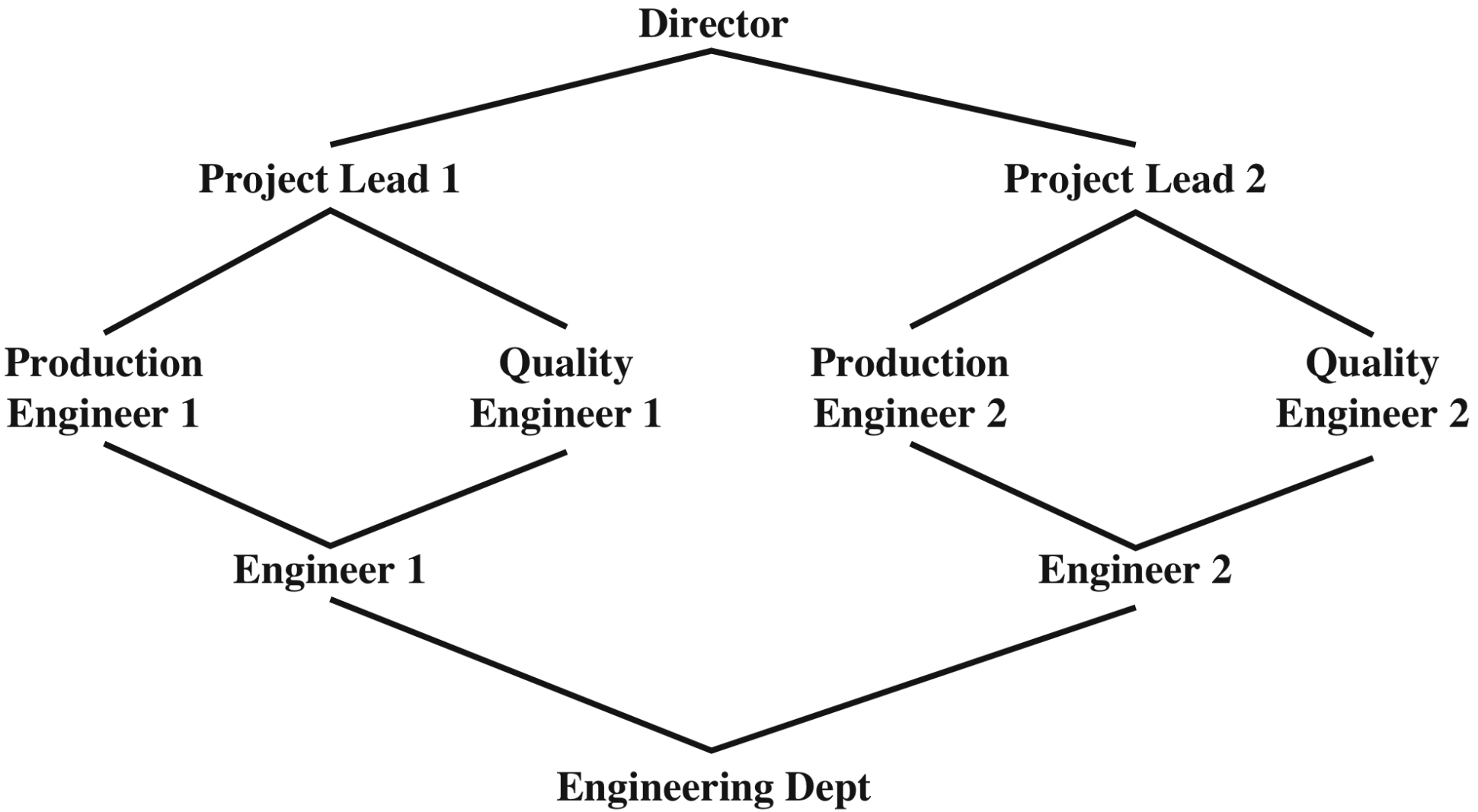
- A permission can be assigned to many roles
- Each role can have many permissions
  - read, write, append, execute

Scope RBAC Models



Models	Hierarchies	Constraints
$RBAC_0$	No	No
$RBAC_1$	Yes	No
$RBAC_2$	No	Yes
$RBAC_3$	Yes	Yes

Example of Role Hierarchy



- Thanks