INFS 767 Fall 2001

The RBAC96 Model

Prof. Ravi Sandhu
George Mason University

AUTHORIZATION, TRUST AND RISK

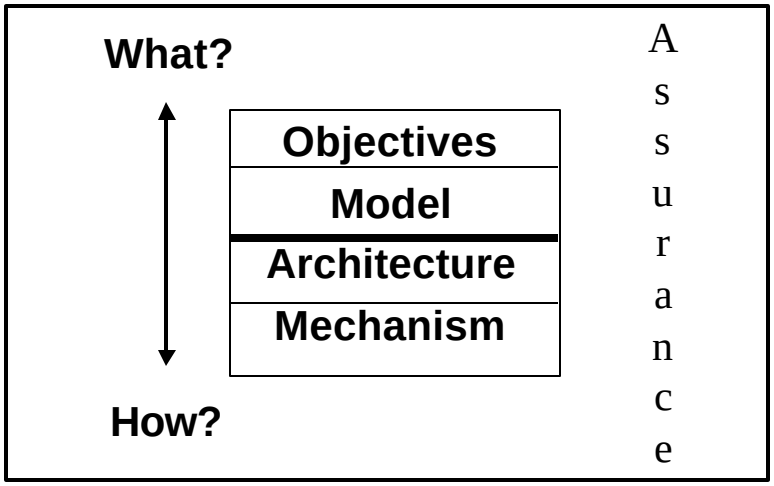
- **Information security is fundamentally about managing**
  - **authorization and**
  - **trust**

  so as to manage risk

SOLUTIONS

- **OM-AM**
- **RBAC**
- **PKI**
- **and others**

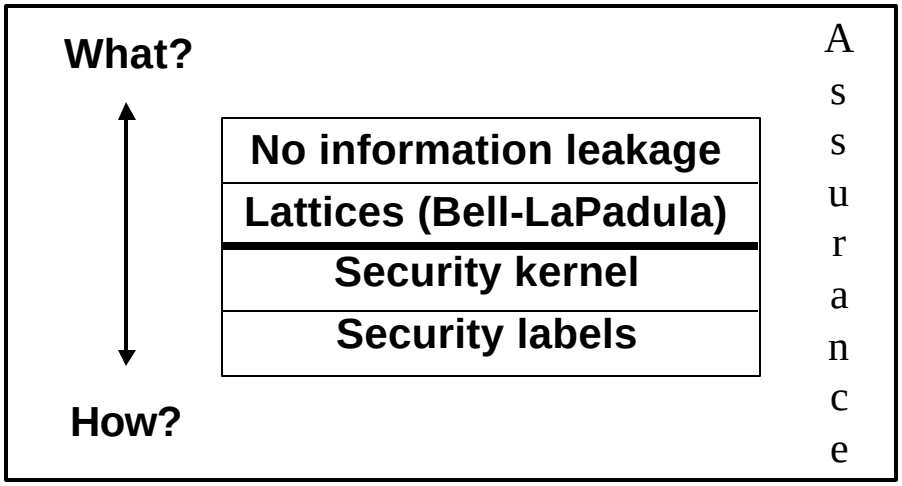
THE OM-AM WAY



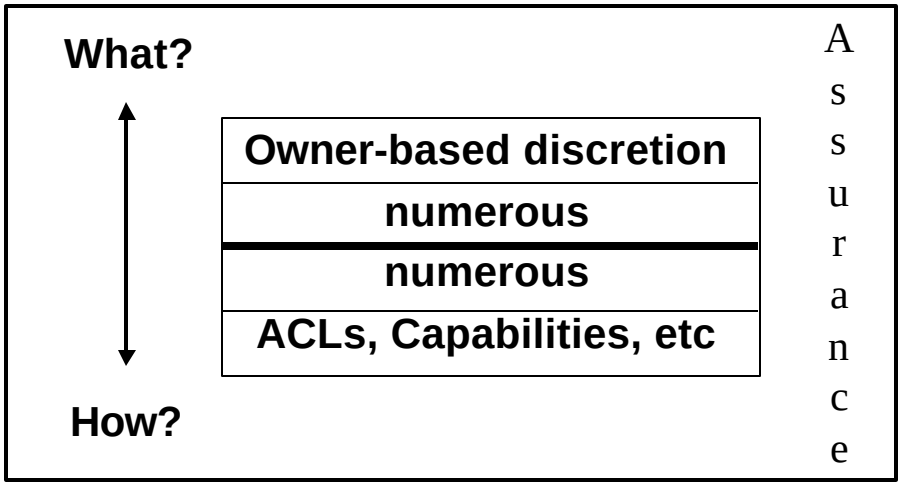
LAYERS AND LAYERS

- **Multics rings**
- **Layered abstractions**
- **Waterfall model**
- **Network protocol stacks**
- **OM-AM**

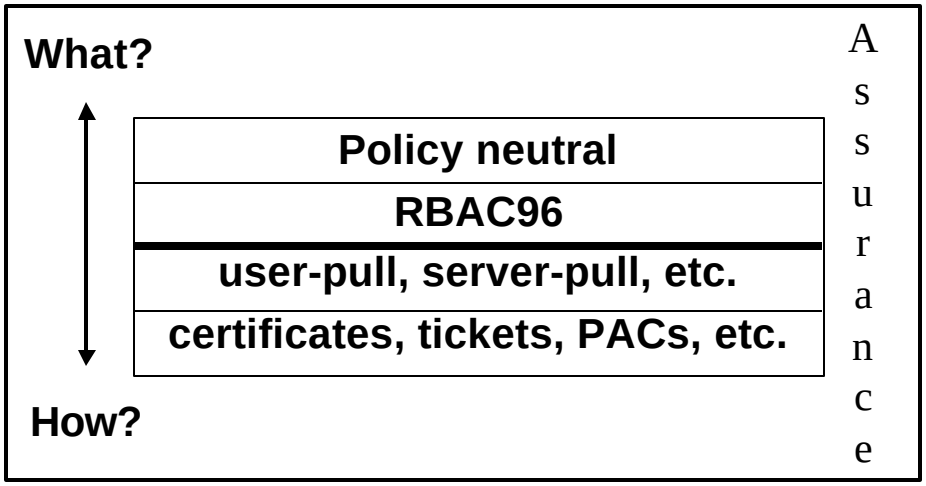
OM-AM AND MANDATORY ACCESS CONTROL (MAC)



OM-AM AND DISCRETIONARY ACCESS CONTROL (DAC)



OM-AM AND ROLE-BASED ACCESS CONTROL (RBAC)



ROLE-BASED ACCESS CONTROL (RBAC)

- **A user's permissions are determined by the user's roles**
  - **rather than identity or clearance**
  - **roles can encode arbitrary attributes**
- **multi-faceted**
- **ranges from very simple to very sophisticated**

WHAT IS THE POLICY IN RBAC?

- **RBAC is a framework to help in articulating policy**
- **The main point of RBAC is to facilitate security management**

RBAC SECURITY PRINCIPLES

- **least privilege**
- **separation of duties**
- **separation of administration and access**
- **abstract operations**

RBAC96 IEEE Computer Feb. 1996

- **Policy neutral**
- **can be configured to do MAC**
  - **roles simulate clearances (ESORICS 96)**
- **can be configured to do DAC**
  - **roles simulate identity (RBAC98)**

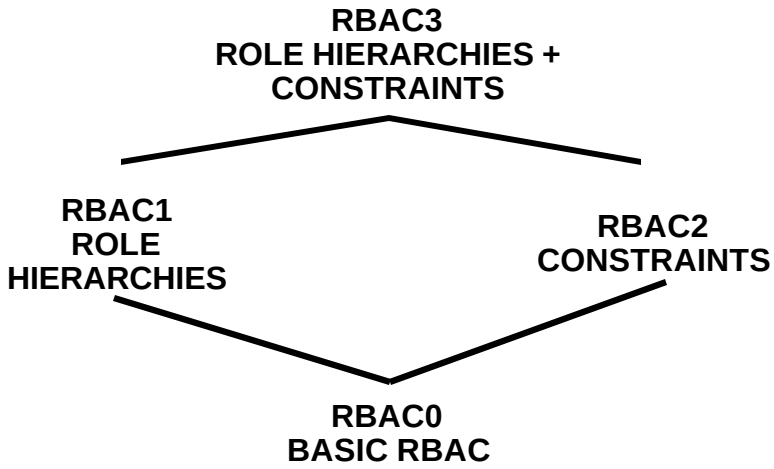
WHAT IS RBAC?

- **multidimensional**
- **open ended**
- **ranges from simple to sophisticated**

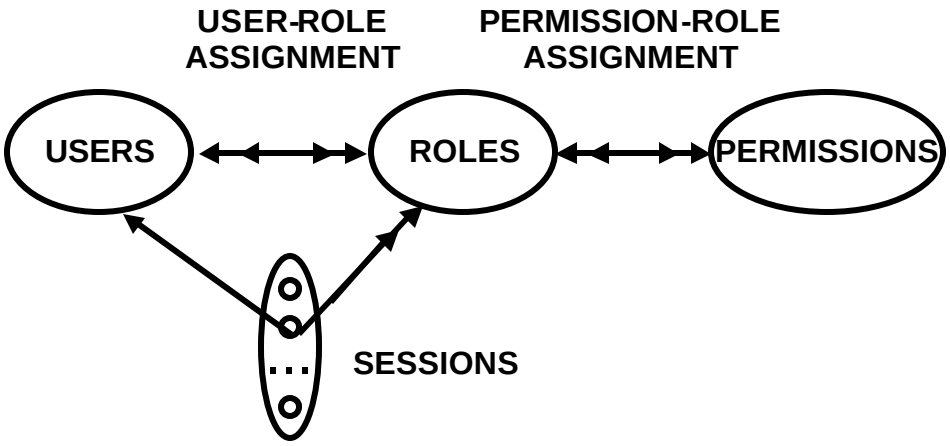
RBAC CONUNDRUM

- **turn on all roles all the time**
- **turn on one role only at a time**
- **turn on a user-specified subset of roles**

RBAC96 FAMILY OF MODELS



RBAC0



PERMISSIONS

- **Primitive permissions**
  - **read, write, append, execute**
- **Abstract permissions**
  - **credit, debit, inquiry**

PERMISSIONS

- **System permissions**
  - **Auditor**
- **Object permissions**
  - **read, write, append, execute, credit, debit, inquiry**

PERMISSIONS

- **Permissions are positive**
- **No negative permissions or denials**
  - **negative permissions and denials can be handled by constraints**
- **No duties or obligations**
  - **outside scope of access control**

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**

USERS

- **Users are**
  - **human beings or**
  - **other active agents**
- **Each individual should be known as exactly one user**

USER-ROLE ASSIGNMENT

- **A user can be a member of many roles**
- **Each role can have many users as members**

SESSIONS

- **A user can invoke multiple sessions**
- **In each session a user can invoke any subset of roles that the user is a member of**

PERMISSION-ROLE ASSIGNMENT

- **A permission can be assigned to many roles**
- **Each role can have many permissions**

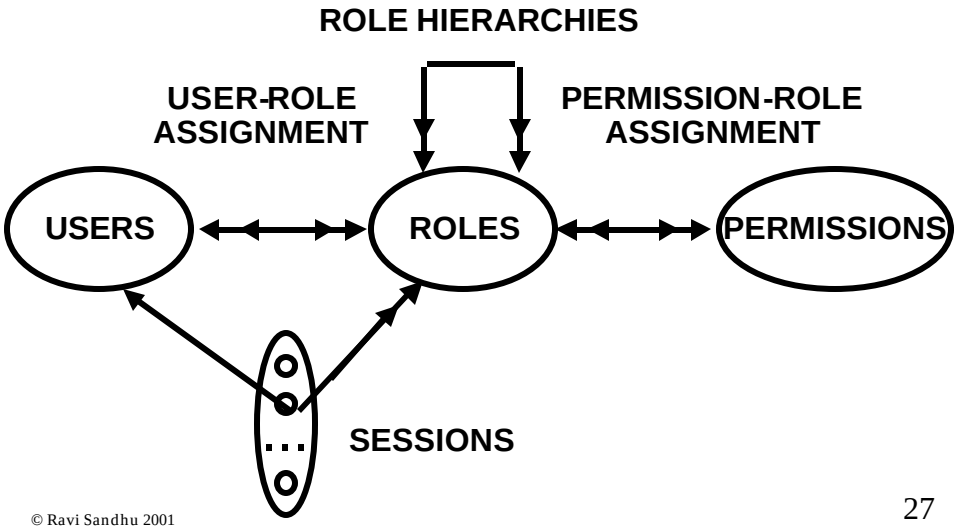
MANAGEMENT OF RBAC

- **Option 1:**

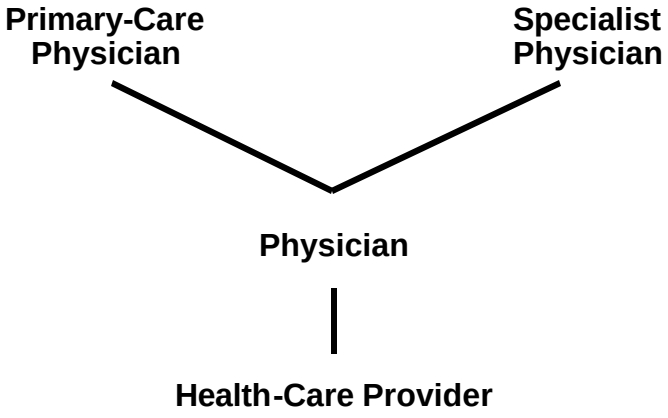
  USER-ROLE-ASSIGNMENT and PERMISSION-ROLE ASSIGNMENT can be changed only by the chief security officer
- **Option 2:**

  Use RBAC to manage RBAC

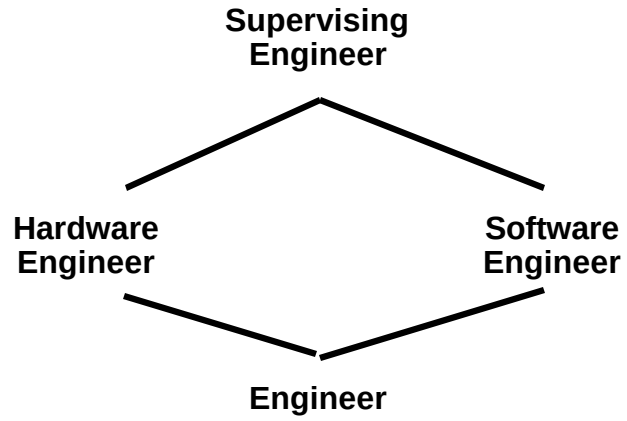
RBAC1



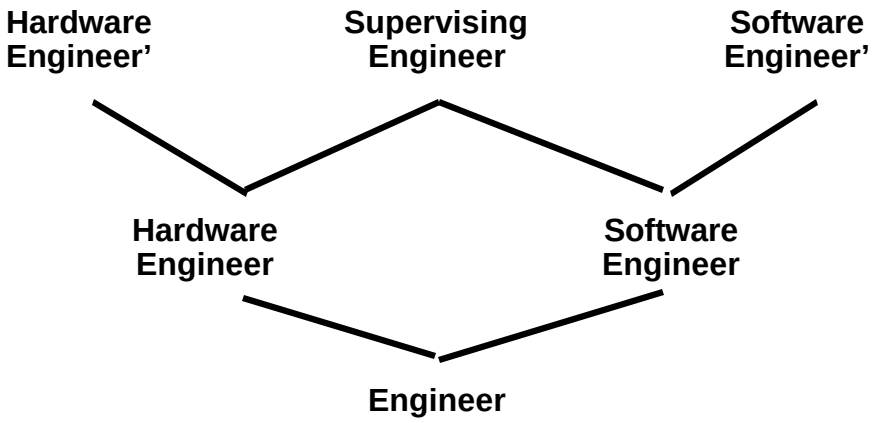
HIERARCHICAL ROLES



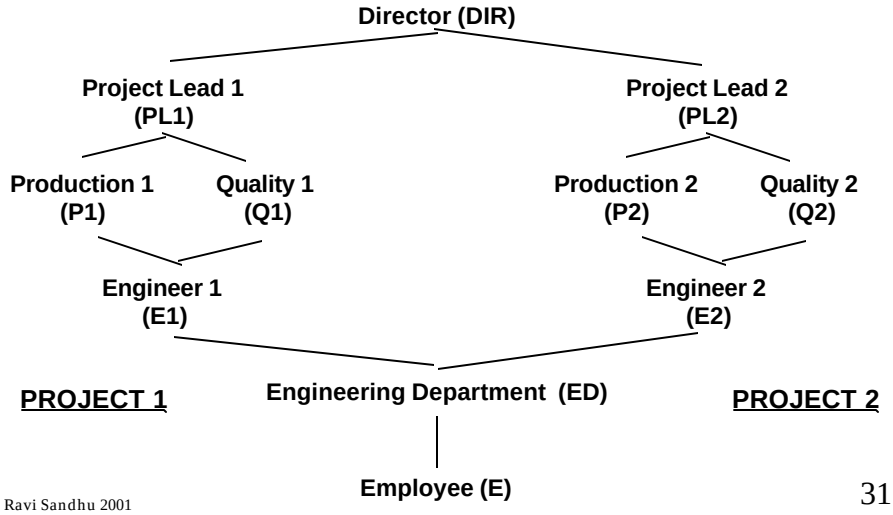
HIERARCHICAL ROLES



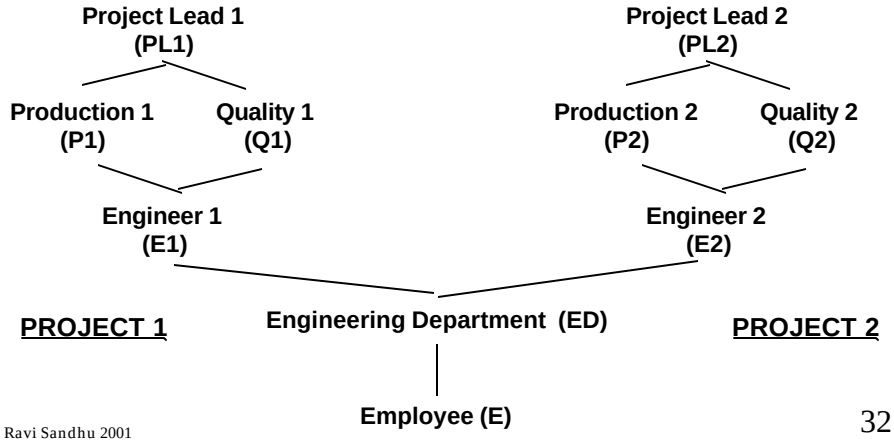
PRIVATE ROLES



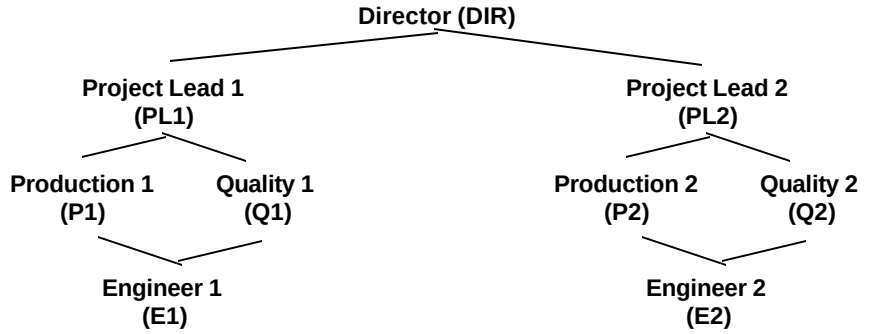
EXAMPLE ROLE HIERARCHY



EXAMPLE ROLE HIERARCHY

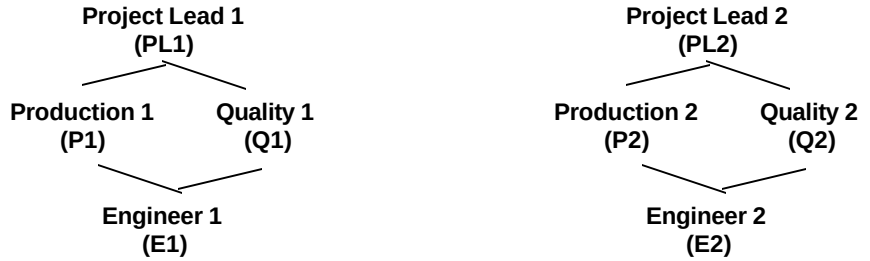


EXAMPLE ROLE HIERARCHY



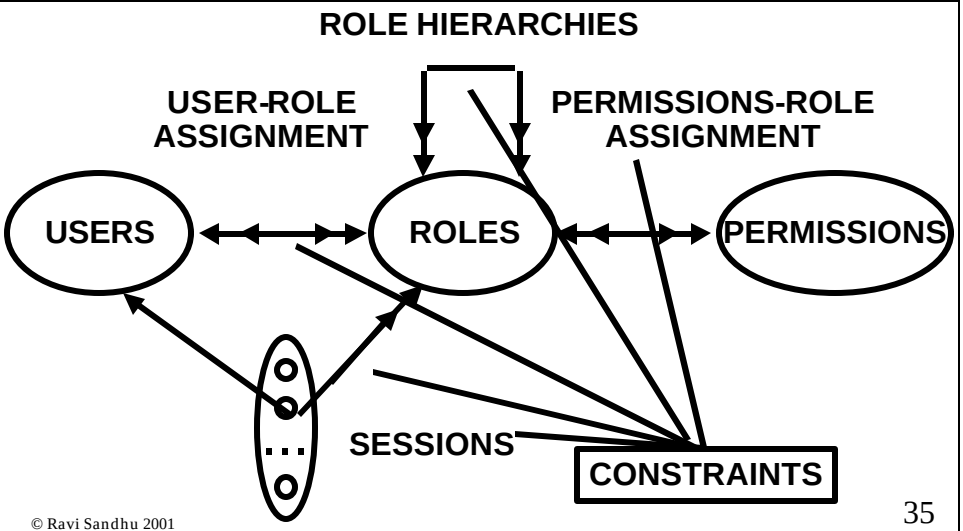
PROJECT 1 **PROJECT 2**

EXAMPLE ROLE HIERARCHY



PROJECT 1 **PROJECT 2**

RBAC3



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**

CONSTRAINTS

- **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**

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**

CONSTRAINTS

- **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**