Access Control

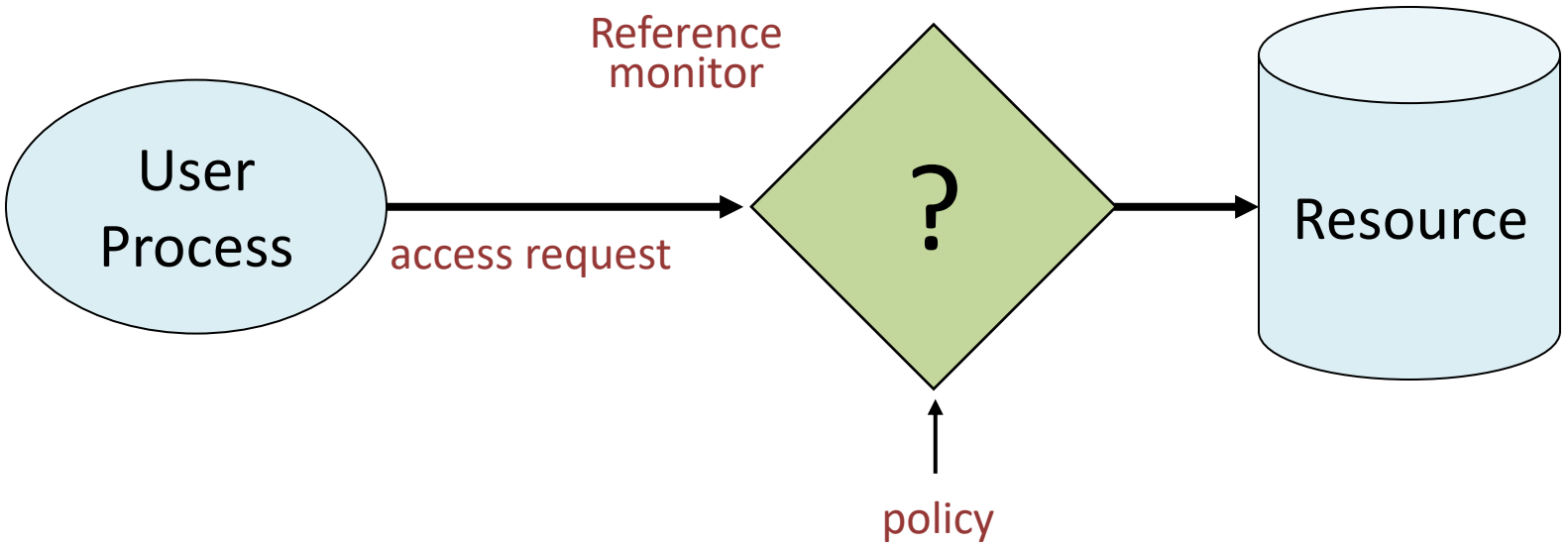
Dr. Chen Zhang

Department of Computer Science
The Hang Seng University of Hong Kong

Slides credits in part from D. Boneh, J. Mitchell, M. Goodrich, and W. Stallings

Access control

- Assumptions
  - System knows who the user is
    - Authentication via name and password, other credential
  - Access requests pass through gatekeeper (reference monitor)
    - System must not allow monitor to be bypassed



Relationship Among Access Control and Other Security Functions

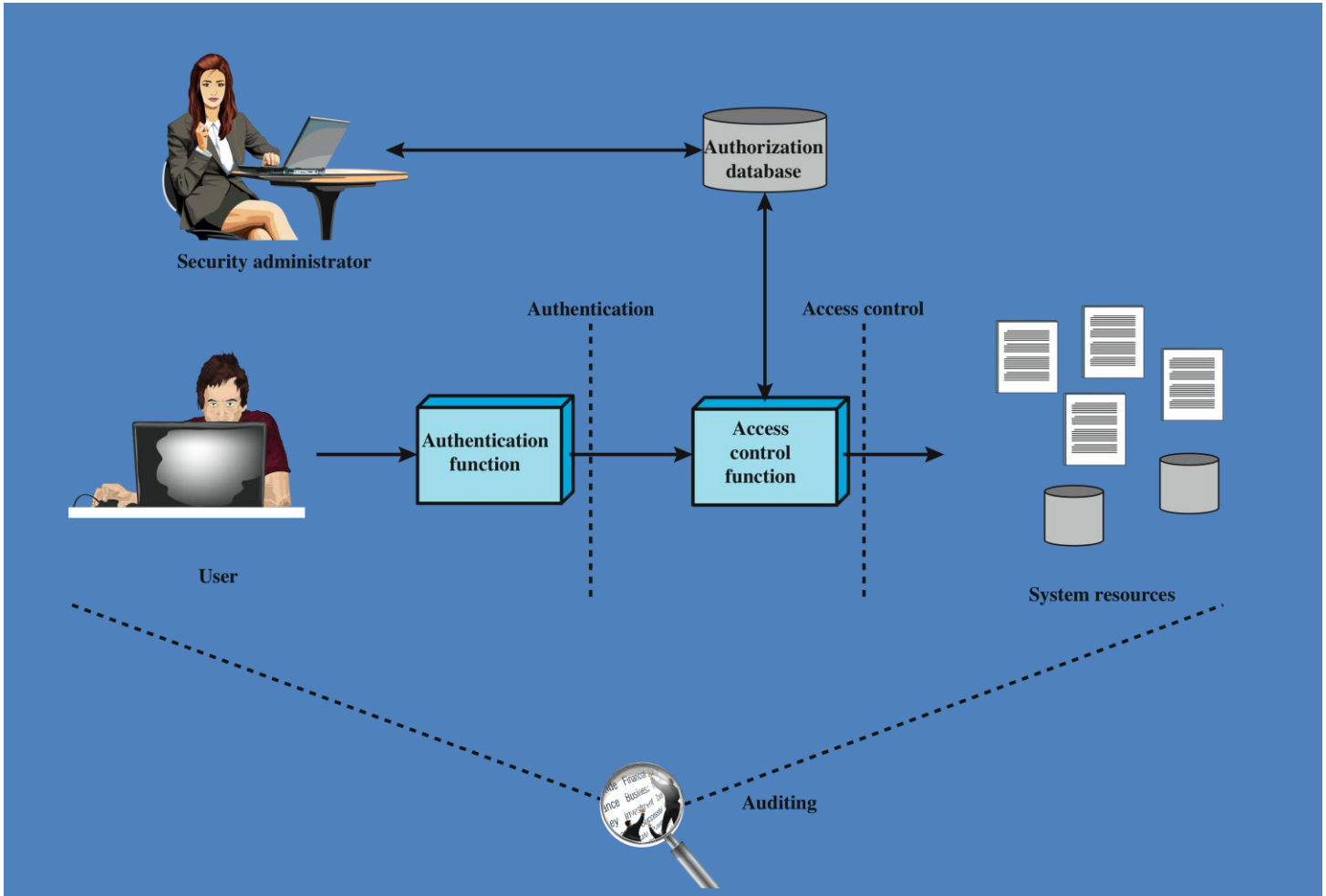


Figure 4.1 Relationship Among Access Control and Other Security Functions

Functions related to Access Control

- **Authentication: Verification that the credentials of a user or other system entity are valid.**

- **Access Control: The granting of a right or permission to a system entity to access a system resource.**
  - This function determines who is trusted for a given purpose.

- **Audit: An independent review and examination of system records and activities.**
  - In order to test for adequacy of system controls, to ensure compliance with established policy and operational procedures, to detect breaches in security, and to recommend any indicated changes in control, policy and procedures.

Access control matrix

Access Control Matrix

- Scenario: an entity may enable another entity to access some resource
- Often provided using an access matrix
  - one dimension consists of identified **subjects** that may attempt data access to the resources
  - the other dimension lists the **objects** that may be accessed
- each entry in the matrix indicates the access rights of a particular subject for a particular object

Access Control Matrix

Objects

Subjects

	File 1	File 2	File 3	...	File n
User 1	read	write	-	-	read
User 2	write	write	write	-	-
User 3	-	-	-	read	read
...					
User m	read	write	read	write	read

Access Control Matrix for Unix

	/etc/passwd	/usr/bin/	/u/roberto/	/admin/
root	read, write	read, write, exec	read, write, exec	read, write, exec
mike	read	read, exec		
roberto	read	read, exec	read, write, exec	
backup	read	read, exec	read, exec	read, exec
. . .	. . .	. . .	. . .	. . .

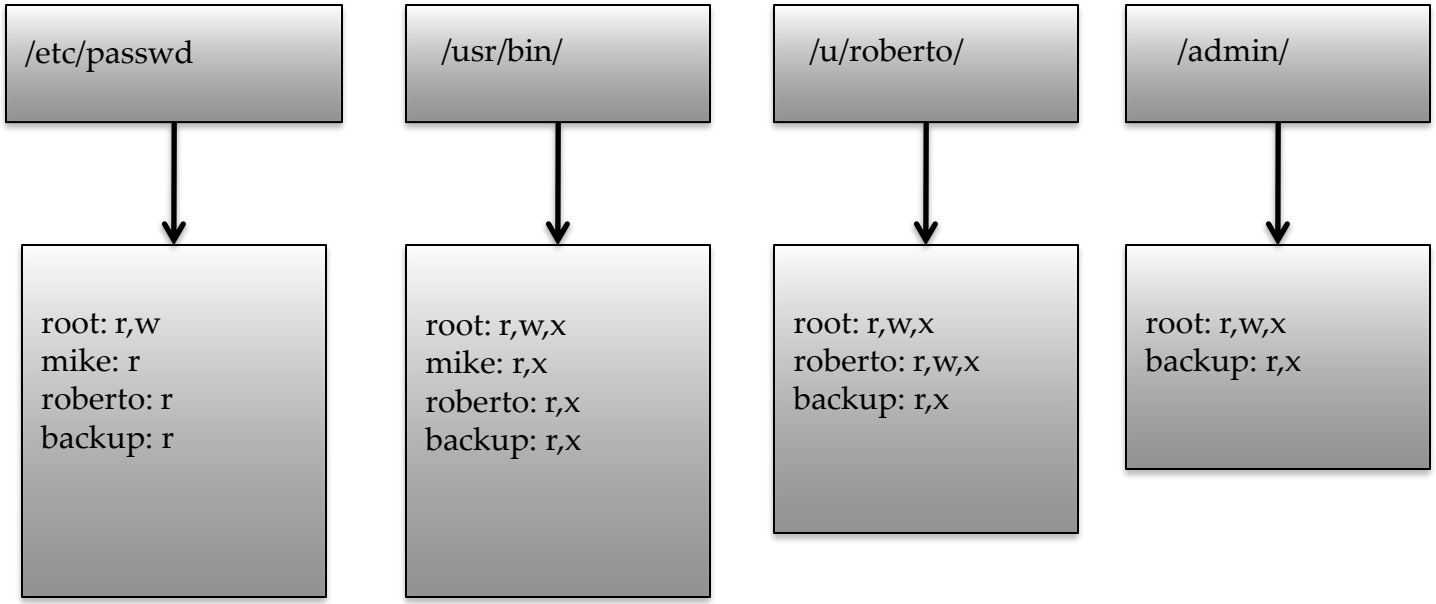
Two implementation concepts

- Access control list (ACL)
  - Store column of matrix with the resource
- Capability
  - User holds a “ticket” for each resource
  - Two variations
    - store row of matrix with user, under OS control
    - unforgeable ticket in user space

	File 1	File 2	…
User 1	read	write	-
User 2	write	write	-
User 3	-	-	read
…			
User m	Read	write	write

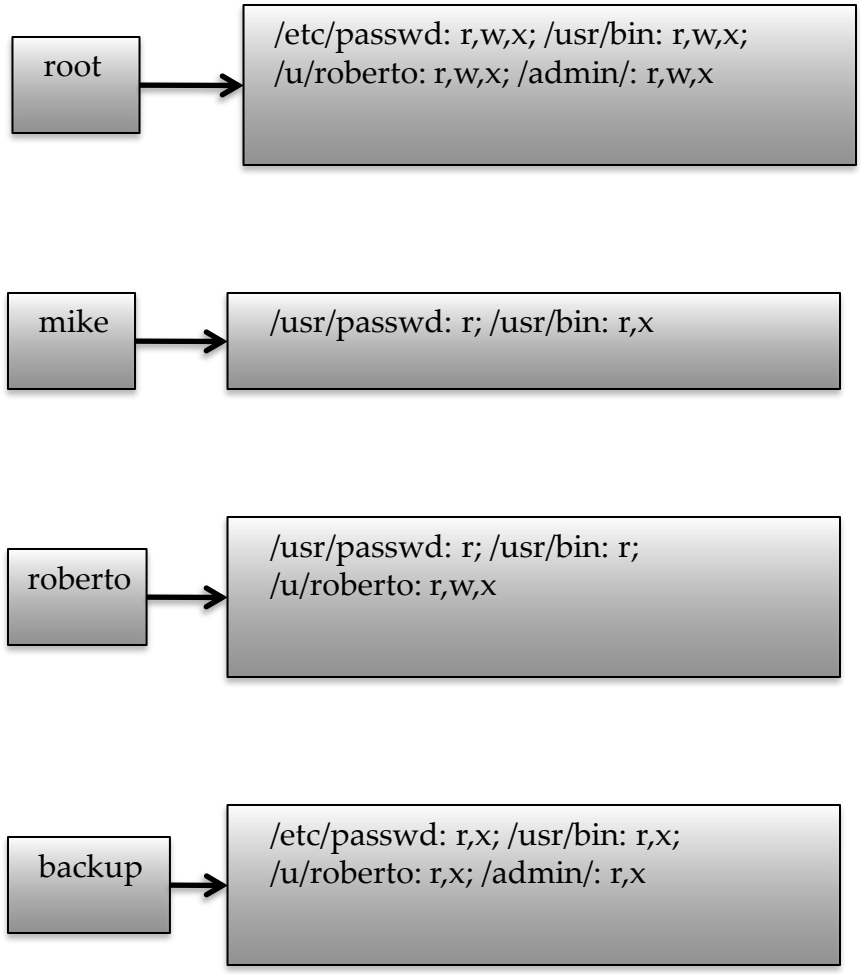
Access Control List Examples

- It defines, for each object, o, a list, L, called o's access control list, which enumerates all the subjects that have access rights for o and, for each such subject, s, gives the access rights that s has for object o.



Capability Examples

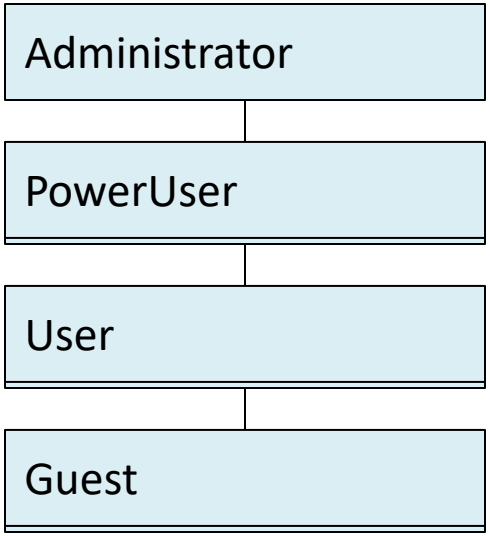
- Takes a subject-centered approach to access control. It defines, for each subject s, the list of the objects for which s has nonempty access control rights, together with the specific rights for each such object.



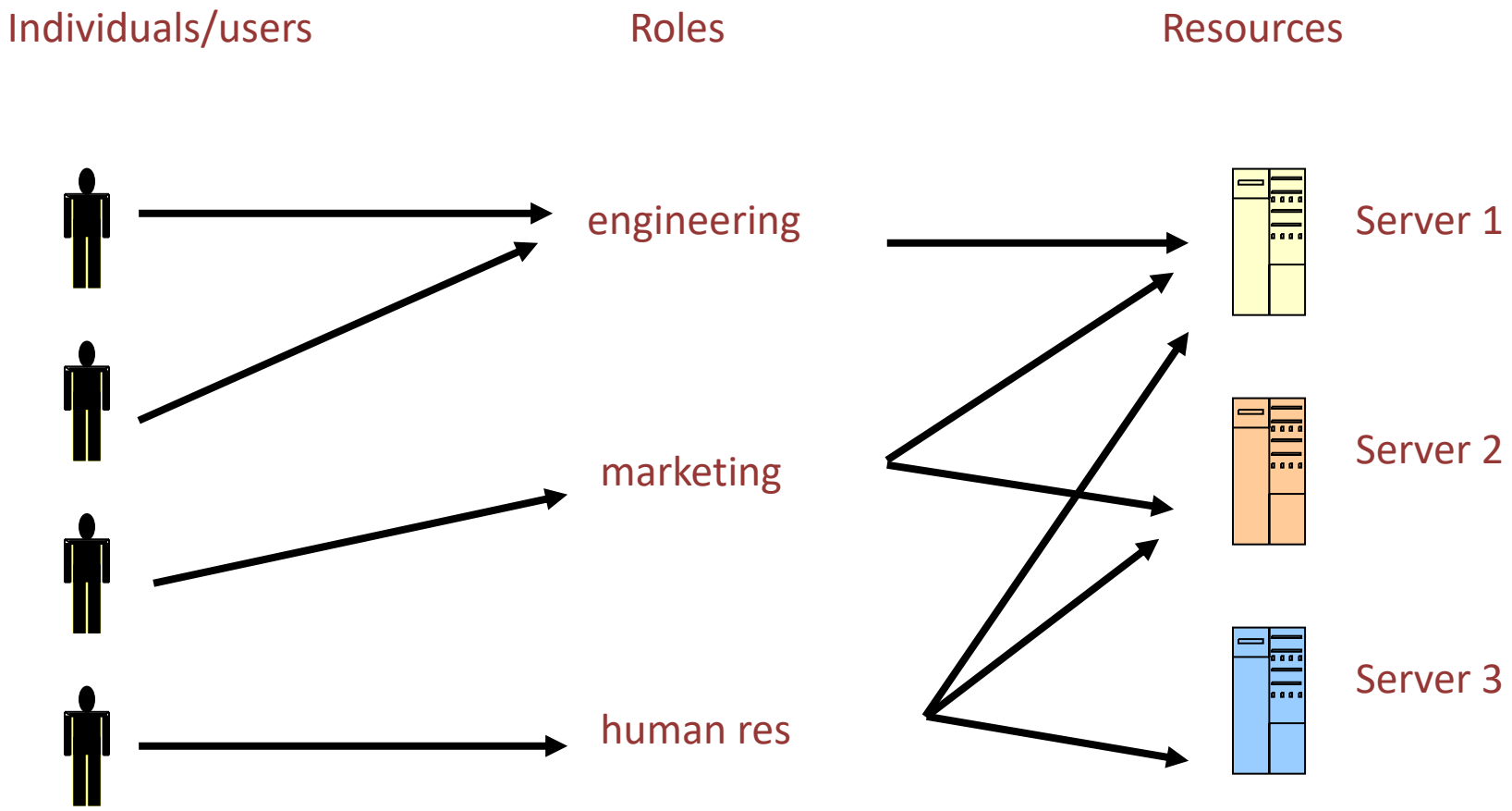
Role based Access Control (RBAC)

Roles (also called Groups)

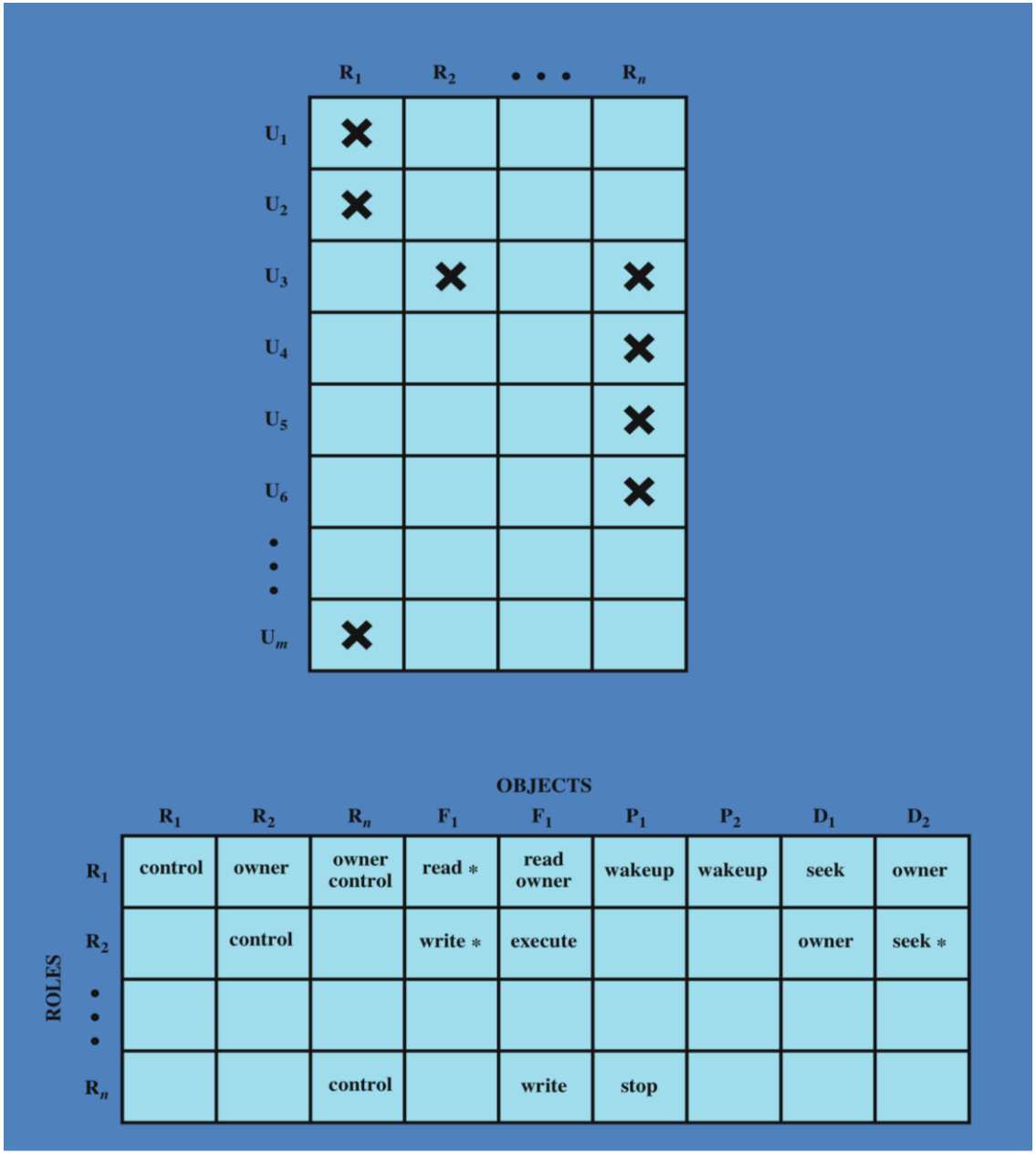
- Role = set of users
  - Administrator, PowerUser, User, Guest
  - Assign permissions to roles; each user gets permission



Role-Based Access Control



Advantage: users change more frequently than roles



The matrix relates individual users to roles.

Access control matrix for roles instead of users

Efficiency of RBAC

- RBAC has the potential to offer greater administrative efficiency for:
  - giving permissions to new users;
  - reviewing and removing old privileges;
  - changes in a user's job assignment;
  - removal of privileges for leaving employees.
- There is usually a direct relationship between the cost of administration and the number of associations that must be managed.
- The larger the number of associations, the costlier and more error-prone access control administration.
- In most organisations RBAC reduces the number of associations that must be managed.