SSW 322: Software Engineering Design VI

Access Control
2020 Spring

Prof. Lu Xiao
lxiao6@stevens.edu
Office Hour: Monday/Wednesday 2 to 4 pm
https://stevens.zoom.us/j/632866976
Software Engineering
School of Systems and Enterprises

Today’s Topic – Access Control

- Access Control
  - Principles
  - Key Concepts---Subjects, Objects, Access Rights
  - Discretionary Access Control
  - Role-based Access Control
  - Attribute-based Access Control

- Computer Security---Principles and Practice 4th Edition, William Stallings and Lawire Brown, ISBN-10 1-292-22061-9

What is Access Control?

1. NISTIR 7298 (Glossary of Key Information Security Terms, May 2013)
   - The process of granting or denying specific requests to: 1) obtain and use information and related information processing services; 2) enter specific physical facilities
2. RFC 4949 (Internet Security Glossary)
   - A process by which use of system resources is regulated according to a security policy and is permitted only authorized entities (users, programs, processes, or other systems) according to that policy.

Access Control Principles

- NIST SP 800-171 (Protecting Controlled Unclassified Information in Nonfederal Information Systems and Organizations, December 2016)

  1. Limit information system access to authorized users, processes on behalf of authorized users, or devices.
  2. Limit information system access to the types of transactions and functions that authorized users are permitted to execute.
  3. Control the flow of CUI (controlled unclassified information) in accordance with approved authorization, such as firewalls, proxies, encryption, and other technologies.

Access Control Principles

4. Separate the duties of individuals to reduce the risk of malevolent activity without collusion.
5. Employ the principle of least privilege, including for specific security functions and privileged accounts.
6. Use non-privileged accounts or roles when accessing nonsecurity functions.
7. Prevent non-privileged users from executing privileged functions and audit the execution of such functions.
8. Limit unsuccessful logon attempts.
9. Provide privacy and security notices consistent with applicable CUI rules.
10. Use session lock with pattern-hiding displays to prevent access and viewing of data after period of inactivity.

Access Control Principles

11. Terminate a user session after a defined condition.
12. Monitor and control remote access session.
13. Employ cryptographic mechanisms to protect the confidentiality of remote access sessions.
14. Route remote access via managed access control points.
15. Authorize remote execution of privileged commands and remote access to security-relevant information.
16. Authorize wireless access prior to allowing such connections.
17. Protect wireless access using authentication and encryption.

Access Control Principles

18. Control connection of mobile devices
19. Encrypt CUI on mobile devices.
20. Verify and control/limit connections to and use of external information systems.
21. Limit use of organizational portable storage devices on external information systems
22. Control CUI posted or processed on publicly accessible information systems.

Subjects, Objects, and Access Rights

- A subject is an entity capable of accessing objects.
- A subject is typically held accountable for the actions they have initiated, and an audit trial may be used to record the associated of a subject with security-relevant actions.
- Basic access control systems typically define three classes of subjects:
  - **Owner:** the creator of the resource, such as a file.
  - **Group:** a named group of users may also be granted access rights, such that membership in the group is sufficient to exercise relevant access rights.
  - **World:** the least amount of access is granted to users who can access the system but are not included in the categorizes owner and group for this resource.

Subjects, Objects, and Access Rights

- An **object** is a resource to which access is controlled.
  - Examples include records, blocks, pages, segments, files, portions of files, directories, directory trees, mailboxes, messages, and programs.
  - Some access control even encompass bits, bytes, words, processors, communication ports, clocks, and network nodes.

Subjects, Objects, and Access Rights

- An **access right** describes the way in which a subject may access an object:
  - **Read**: subject may view information of the object, such as a file, selected fields. It also includes the right to copy or print.
  - **Write**: subject may add, modify, or delete data of the object.
  - **Execute**: subject may execute object programs
  - **Delete**: subject may delete certain system resources, such as files or records.
  - **Create**: subject may create new files, records, or fields.
  - **Search**: Subject may list the files in a directory or otherwise search the directory.

Access Control Policies

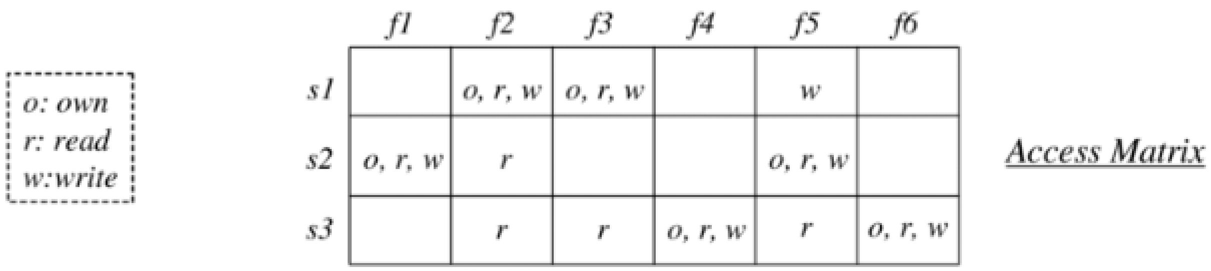
- Discretionary access control (DAC)
  - Controls access based on the identify of the requestor and on access rules stating that requestors are (or are not) allowed to do.
  - *An entity might have access rights that permit the entity to enable another entity to access some resource.*
- Mandatory access control (MAC)
  - Controls access based on comparing security labels with security clearances.
  - *An entity that has clearance to access a resource may **<u>not</u>** enable another entity to access resource.*

Access Control Policies

- Role-based access control (RBAC)
  - Control access based on the **roles** that users have within the system and on **rules** stating what accesses are allowed to users in given roles.
  - For example, with RBAC, a user in the accounts payable clerk position would automatically get added as a member (i.e. dynamic membership) to the AP Role, granting him or her access to AP functions in the accounting system.
- Attribute-based access control (ABAC)
  - Controls access based on **attributes** of the user, the **resource** to be accessed, and current environmental **conditions**.
  - An example of ABAC would be allowing only users who are type=employees and have department=HR to access the HR/Payroll system and only during business hours within the same timezone as the company.

Discretionary access control (DAC)

- An entity may grant access rights to a resource to another entity.
- A general approach to DAC, as exercised by operating systems and database systems, is that of an access matrix.



o: own
r: read
w:write

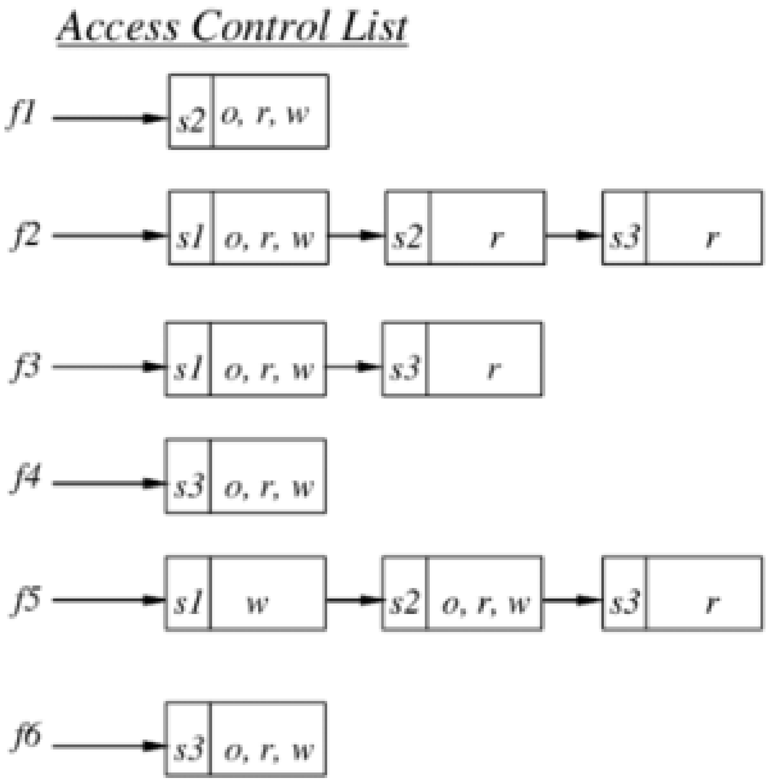
	f1	f2	f3	f4	f5	f6
s1		o, r, w	o, r, w		w	
s2	o, r, w	r			o, r, w	
s3		r	r	o, r, w	r	o, r, w

Access Matrix

- Rows: subjects
- Columns: objects
- Cells: access rights

Discretionary access control (DAC)

- An access matrix is often sparse, and may be decomposed by columns, yielding to ***access control lists***



Access Control List

f1 → s2 | o, r, w

f2 → s1 | o, r, w → s2 | r → s3 | r

f3 → s1 | o, r, w → s3 | r

f4 → s3 | o, r, w

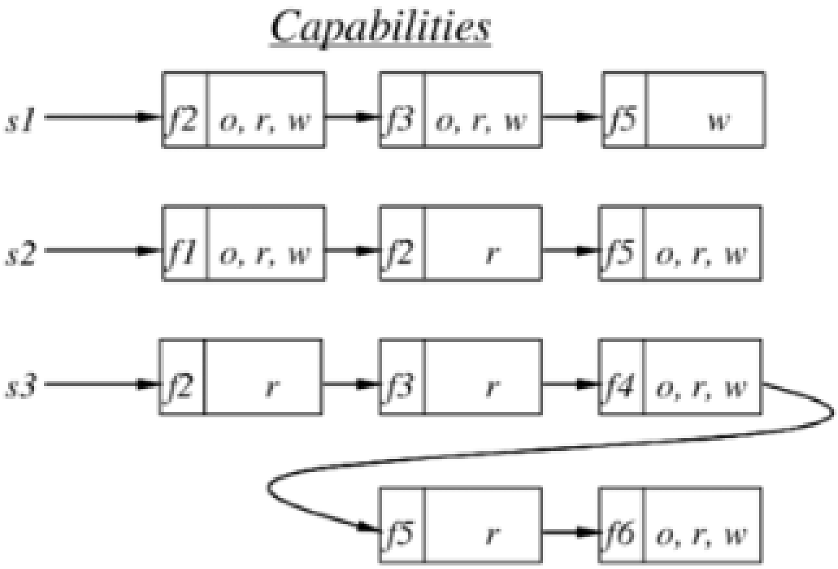
f5 → s1 | w → s2 | o, r, w → s3 | r

f6 → s3 | o, r, w

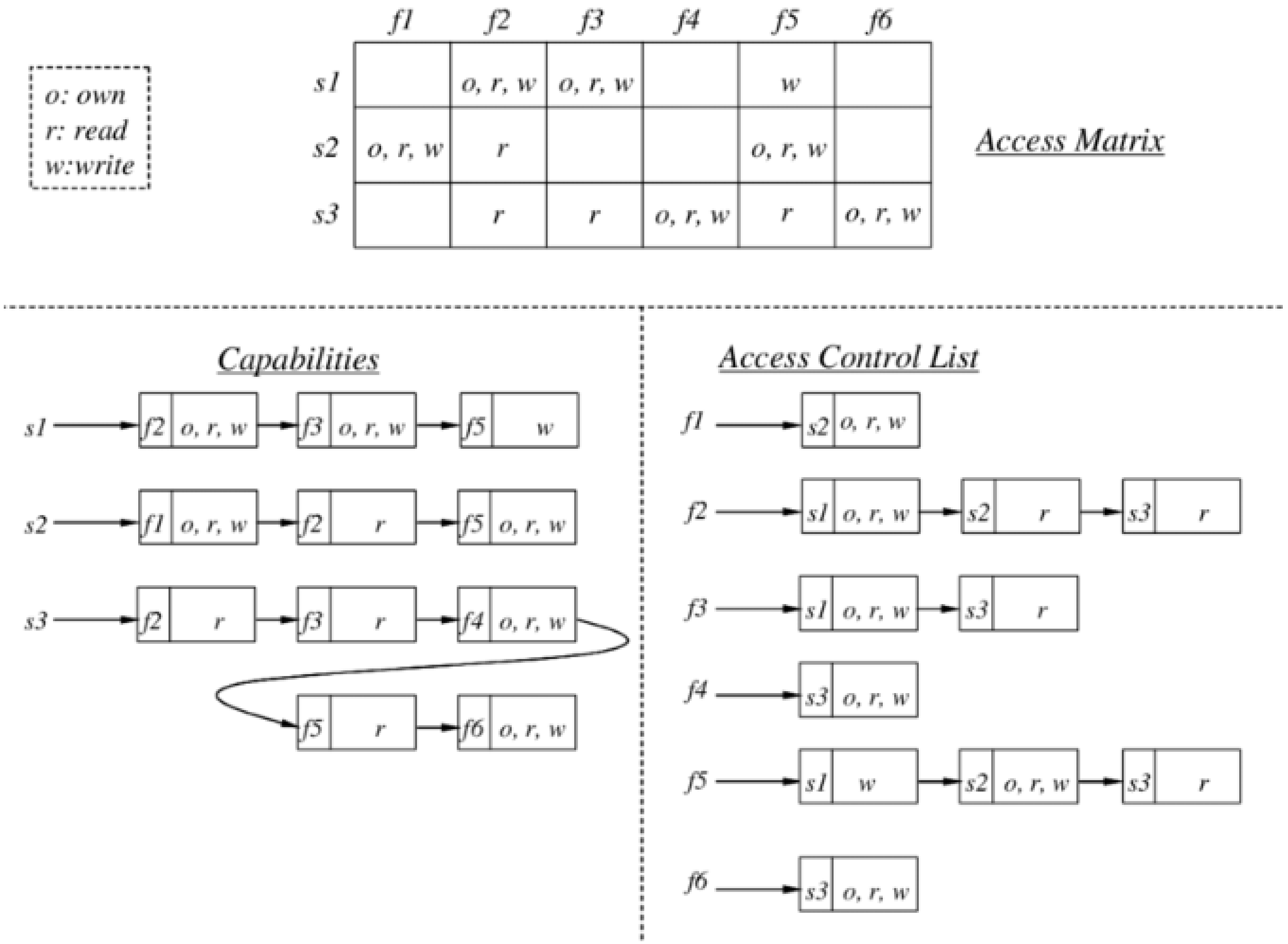
Discretionary access control (DAC)

- Decomposition by rows of an access matrix yields to ***capability tickets***

ACL lists users and their rights



Discretionary access control (DAC)



o: own
r: read
w:write

	f1	f2	f3	f4	f5	f6
s1		o, r, w	o, r, w		w	
s2	o, r, w	r			o, r, w	
s3		r	r	o, r, w	r	o, r, w

Access Matrix

Capabilities

s1 → f2 o, r, w → f3 o, r, w → f5 w

s2 → f1 o, r, w → f2 r → f5 o, r, w

s3 → f2 r → f3 r → f4 o, r, w → f5 r → f6 o, r, w

Access Control List

f1 → s2 o, r, w

f2 → s1 o, r, w → s2 r → s3 r

f3 → s1 o, r, w → s3 r

f4 → s3 o, r, w

f5 → s1 w → s2 o, r, w → s3 r

f6 → s3 o, r, w

What is the benefit of Access Control list and the capability tickets?

Discretionary access control (DAC)

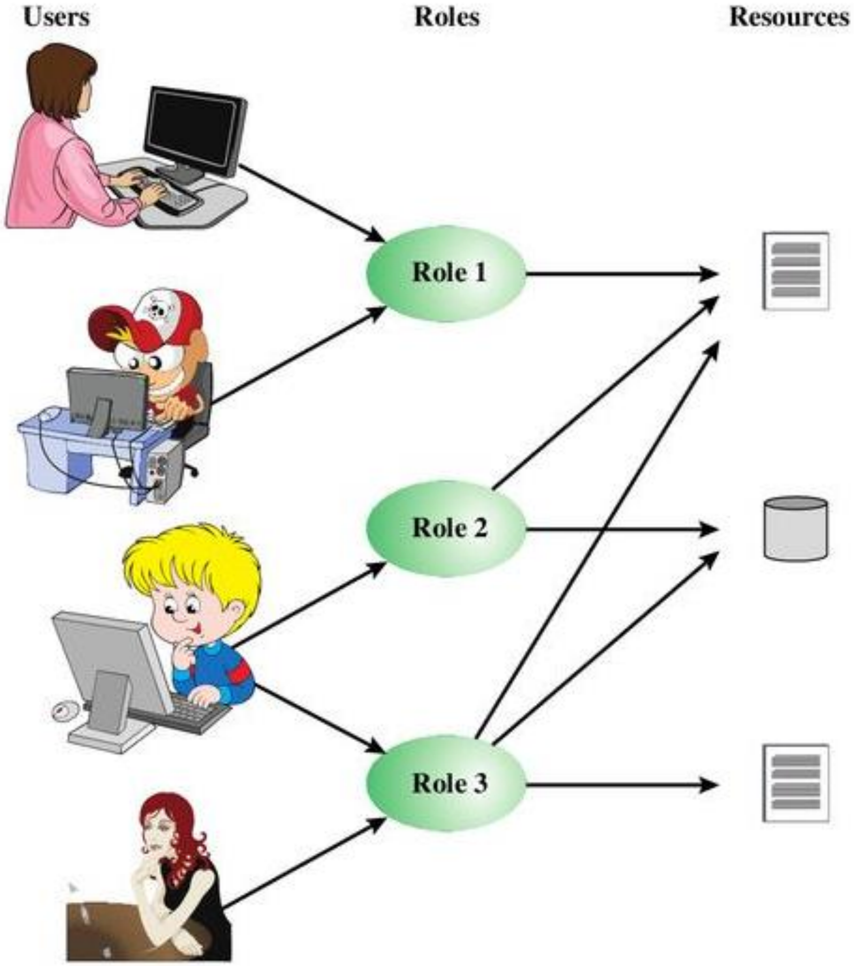
- A separate access control module is associated with each type of object.
- The module evaluates each request by a subject to access an object to check the access right.
- An access attempts triggers the following steps:
  1. A subject *S0* issues a request of type *a* for object *X*
  2. The system generate a message *(S0,a,X)* to the controller of *X*.
  3. The controller access the access matrix *A* to determine if a is in A[S0,X], if so, access is allowed; it not, access is denied.

Role-based access control (RBAC)

- DAC defines the access rights of individual users and groups of users.
- RBAC defines access rights based on the roles that users assume in a system rather than user identify.
- RBAC typically defines a role as a job function within an organization.
  - Users are assigned to different roles, either statically or dynamically, according to their responsibilities.
- RBAC enjoys widespread commercial use and remains an area of active research.

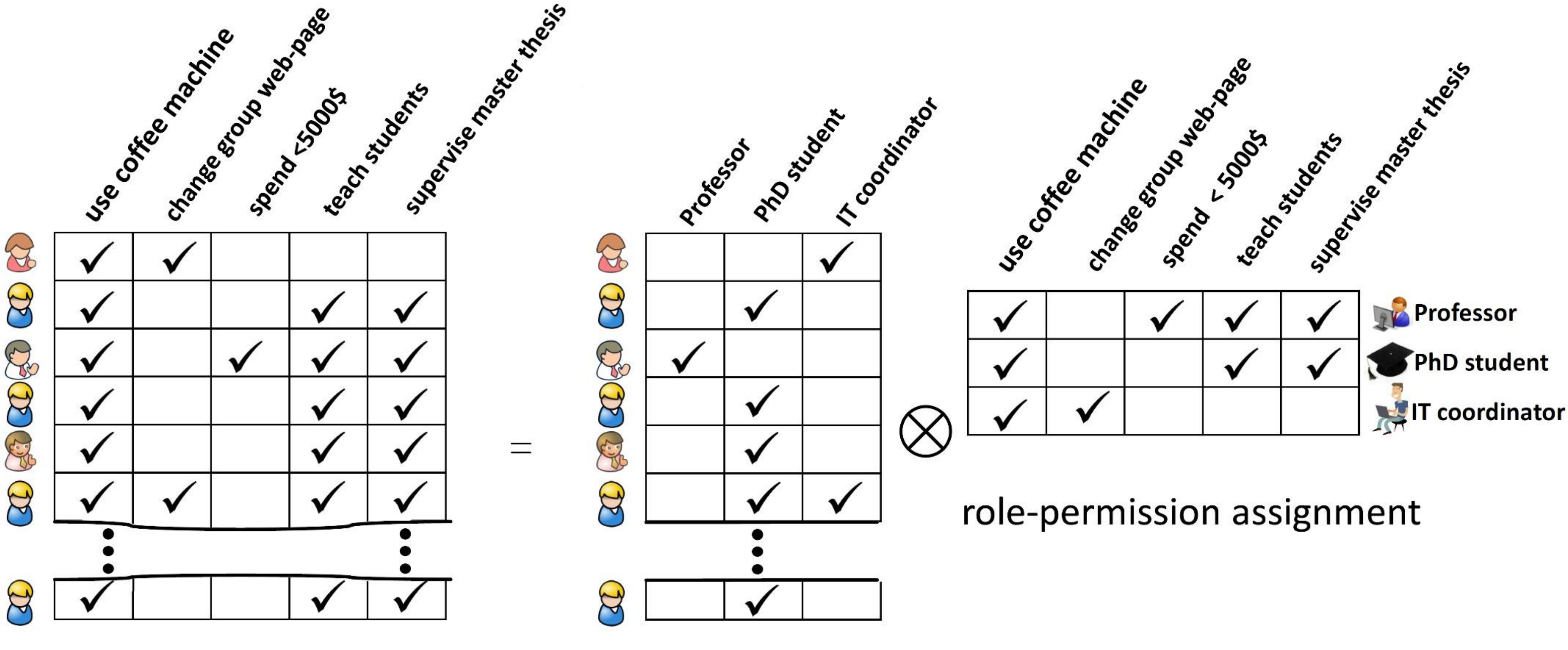
Role-based access control (RBAC)

- The relationship between users to roles is many to many, as is the relationship of roles to resources, or system objects.



Role-based access control (RBAC)

- We can use the access matrix representation to depict the key elements of an RBAC system



	use coffee machine	change group web-page	spend <5000$	teach students	supervise master thesis
	✓	✓			
	✓			✓	✓
	✓		✓	✓	✓
	✓			✓	✓
	✓			✓	✓
	✓	✓		✓	✓
	⋮				⋮
	✓			✓	✓

user-permission assignment

=

	Professor	PhD student	IT coordinator
			✓
		✓	
	✓		
		✓	
		✓	
		✓	✓
		⋮	
		✓	

user-role assignment

⊗

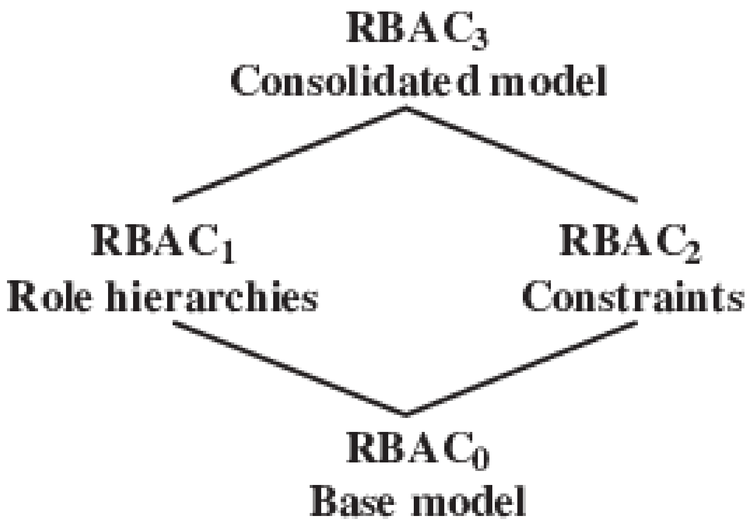
	use coffee machine	change group web-page	spend < 5000$	teach students	supervise master thesis
Professor	✓		✓	✓	✓
PhD student	✓			✓	✓
IT coordinator	✓	✓			

role-permission assignment

www.mariofrank.net

Role-based access control (RBAC)

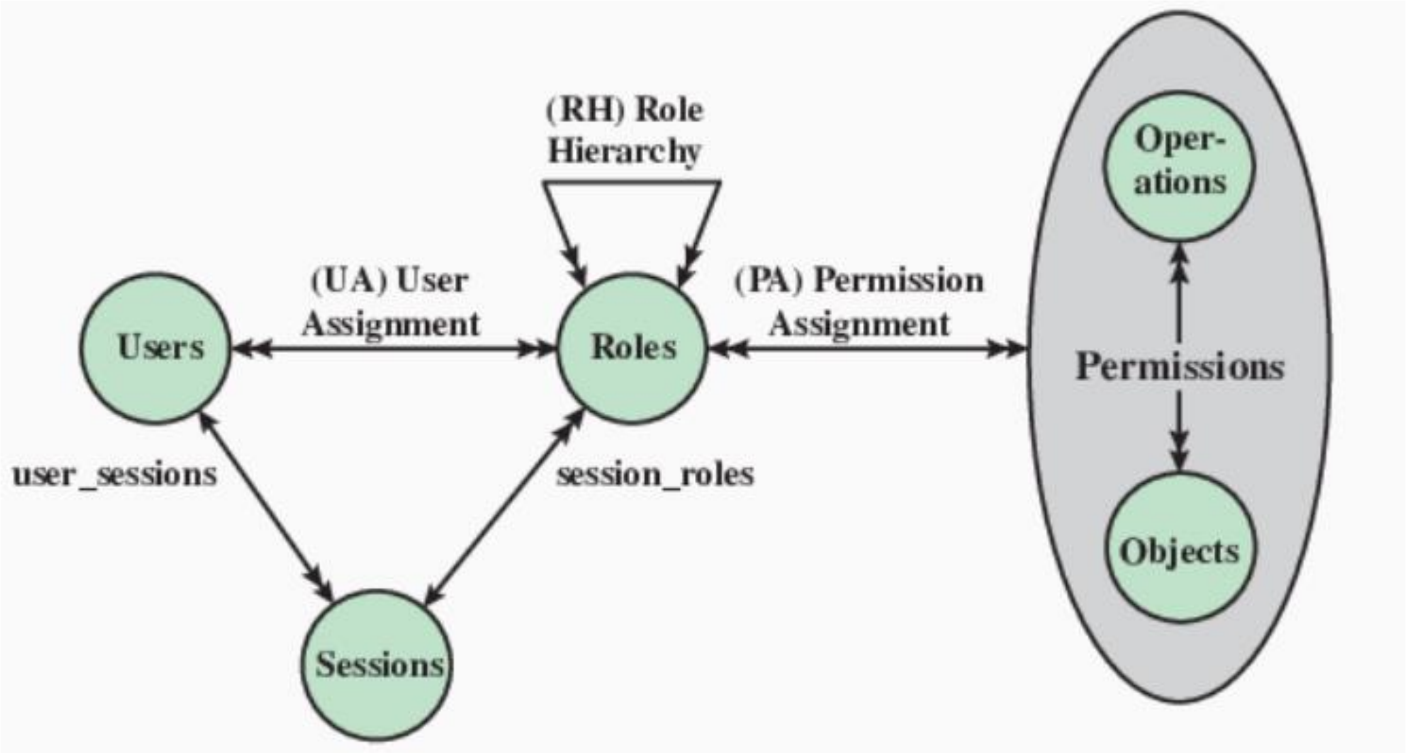
- A variety of functions and services can be included under the general RBAC approach.
- RBAC0: Base Model---Users, roles, permissions, and sessions.
- RBAC1: Role Hierarchies, e.g. director, project lead, engineer, etc.
- RBAC2: Constraints among roles or conditions, mutually exclusive roles and cardinality.



(a) Relationship among RBAC models

RBAC0: Base Model

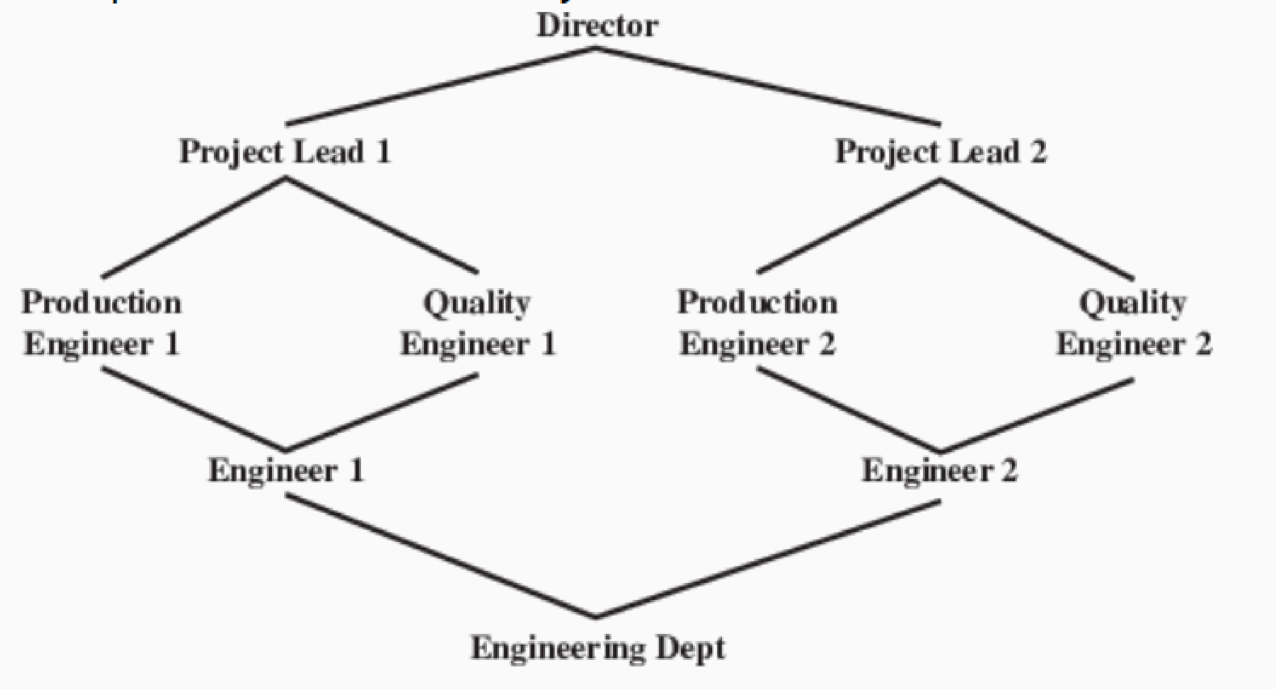
- User: an individual (with UID) with access to system
- Role: a named job function (tells authority level)
- Permission: equivalent to access rights
- Session: a mapping between a user and set of roles to which a user is assigned



RBAC1: Role Hierarchies

- Director has most privileges
- Each role inherits all privileges from lower roles
- A role can inherit from multiple roles
- Additional privileges can be assigned to a role

Example of role hierarchy



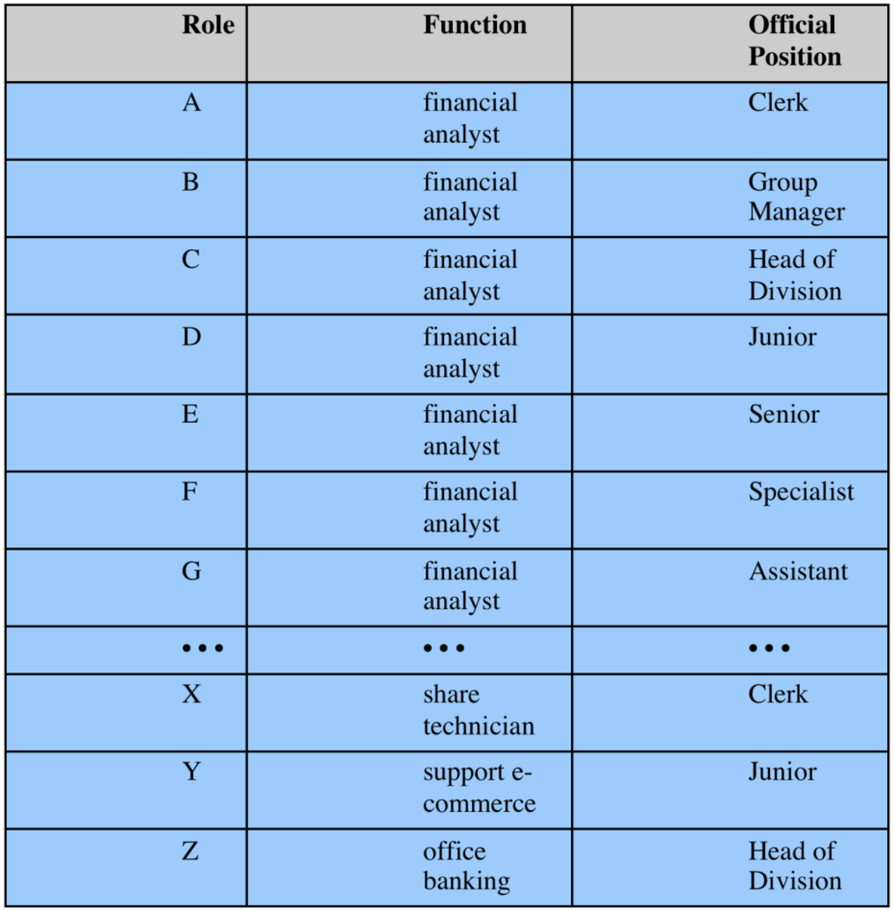
RBAC2: Constraints

- A condition (restriction) on a role or between roles
  1. Mutually exclusive
     - role sets such that a user can be assigned to only one of the role in the set
     - Any permission can be granted to only one role in the set
  2. Cardinality: set a maximum number (of users) with respect to a role (e.g., a department chair role)
  3. Prerequisite role: a user can be assigned a role only if that user already has been assigned to some other role

Case study: RBAC system for a bank

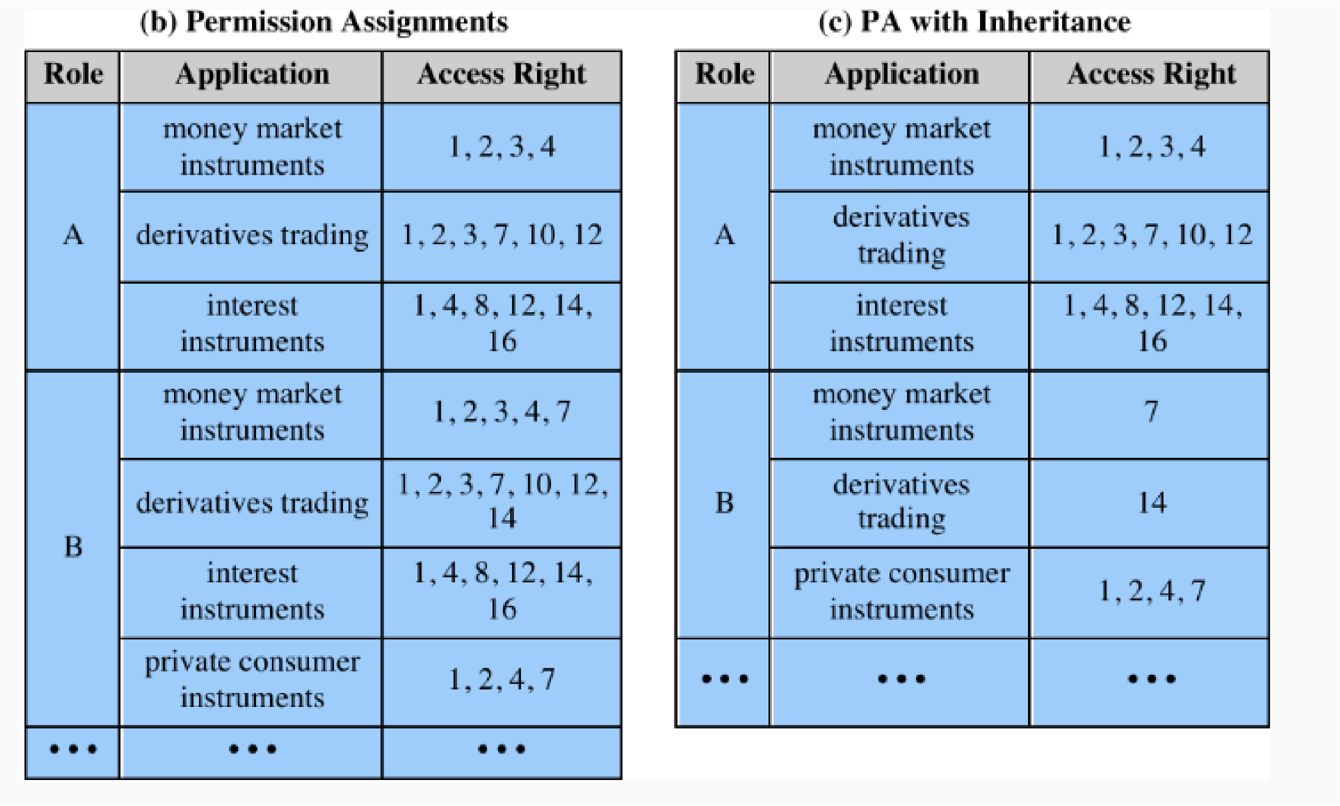
Table 4.4 Functions and Roles for Banking Example

(a) Functions and Official Positions



Role	Function	Official Position
A	financial analyst	Clerk
B	financial analyst	Group Manager
C	financial analyst	Head of Division
D	financial analyst	Junior
E	financial analyst	Senior
F	financial analyst	Specialist
G	financial analyst	Assistant
• • •	• • •	• • •
X	share technician	Clerk
Y	support e-commerce	Junior
Z	office banking	Head of Division

Case study: RBAC system for a bank



(b) Permission Assignments

Role	Application	Access Right
A	money market instruments	1, 2, 3, 4
	derivatives trading	1, 2, 3, 7, 10, 12
	interest instruments	1, 4, 8, 12, 14, 16
B	money market instruments	1, 2, 3, 4, 7
	derivatives trading	1, 2, 3, 7, 10, 12, 14
	interest instruments	1, 4, 8, 12, 14, 16
	private consumer instruments	1, 2, 4, 7
• • •	• • •	• • •

(c) PA with Inheritance

Role	Application	Access Right
A	money market instruments	1, 2, 3, 4
	derivatives trading	1, 2, 3, 7, 10, 12
	interest instruments	1, 4, 8, 12, 14, 16
B	money market instruments	7
	derivatives trading	14
	private consumer instruments	1, 2, 4, 7
• • •	• • •	• • •

Attribute-based access control (ABAC)

- Define authorizations that express conditions on properties of both the resource and the subject
  - Subject attributes
  - Object attributes
  - Environment attributes
- Strength: its flexibility and expressive power
- Drawback: high complexity
- Considerable interest in applying the model to cloud services

Subject attributes

- A subject is an active entity that causes information to flow among objects or changes the system state
- Attributes define the identity and characteristics of the subject: Name, Organization, Job title

Object attributes

- An object (or resource) is a passive information system-related entity containing or receiving information
- Objects have attributes that can be leveraged to make access control decisions: Title, Author, Date

Environment attributes

- Describe the operational, technical, and even situational environment or context in which the information access occurs
    - Current date
    - Current virus/hacker activities
    - Network security level
    - Not associated with a resource or subject
- These attributes have so far been largely ignored in most access control policies

ABAC Logical Architecture

1. A subject requests access to an object
2. A.C. is governed by a set of rules (2a):
   1. assesses the attributes of subject (2b),
   2. object (2c) and
   3. env (2d)
3. A.C. grants subject access to object, if authorized

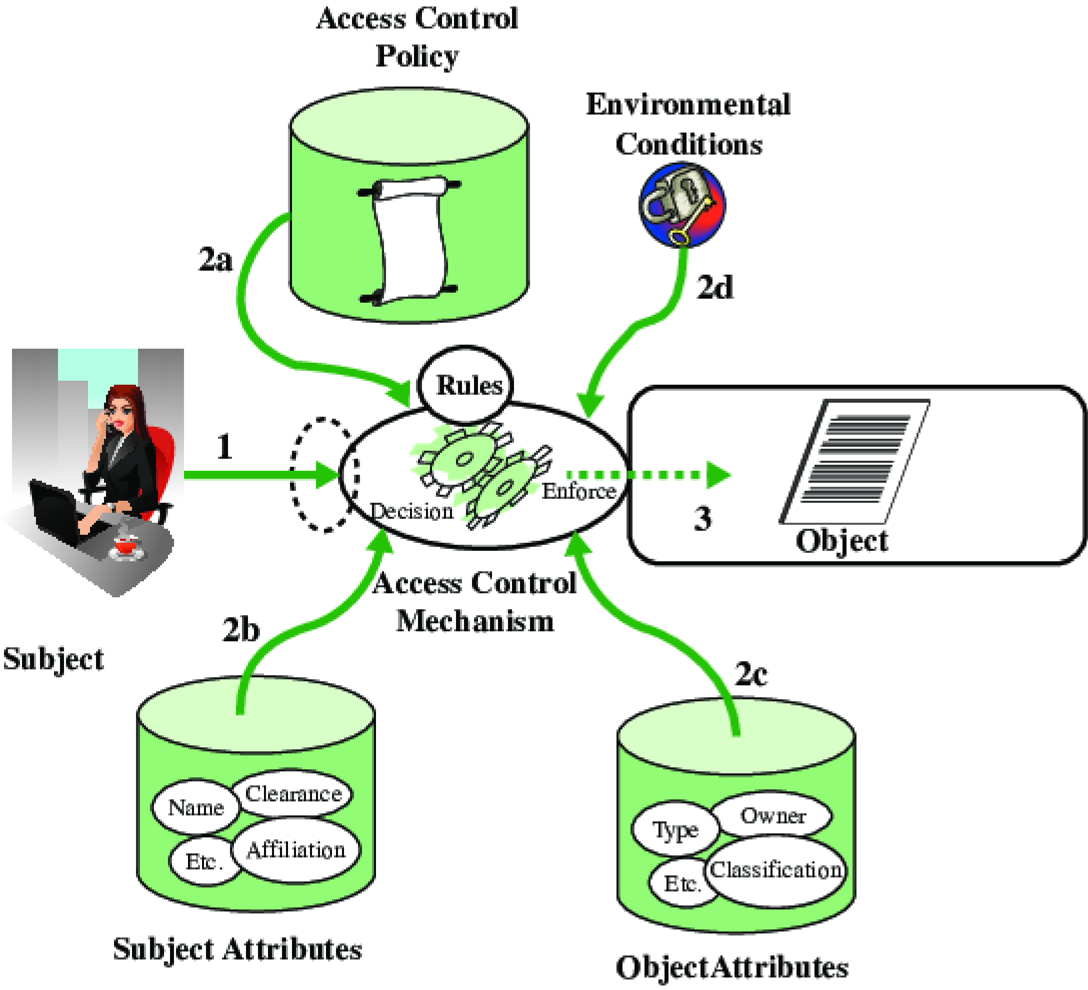


Figure 4.10 Simple ABAC Scenario

ABAC Highlights

- Distinguishable because it controls access to objects by evaluating rules against the attributes of entities, operations, and the environment relevant to a request
- Systems are capable of enforcing DAC, RBAC, and MAC concepts
- Relies upon the evaluation of attributes of the subject, attributes of the object, and a formal relationship or access control rule defining the allowable operations for subject-object attribute combinations in a given environment
- Allows an unlimited number of attributes to be combined to satisfy any access control rule

thank you