

# Computer and Network Security (Security of Computer Systems)

Prof. Dr. Hasan Hüseyin BALIK

(4<sup>th</sup> Week)

# Outline

- 2. Computer security technology and principles
  - 2.1. Cryptographic Tools
  - 2.2. User Authentication
  - 2.3 Access Control
  - 2.4 Database and Data Center Security
  - 2.5 Malicious Software
  - 2.6. Denial-of-Service Attacks
  - 2.7 Intrusion Detection
  - 2.8 Firewalls and Intrusion Prevention Systems

## 2.3. Access Control

## 2.3.Outline

- Access Control Principles
- Subjects, Objects, and Access Rights
- Discretionary Access Control
- Example: UNIX File Access Control
- Role-Based Access Control
- Attribute-Based Access Control
- Identity, Credential, and Access Management
- Trust Frameworks

# Access Control Definitions

## 1/2

NISTIR 7298 (*Glossary of Key Information Security Terms* , May 2013) defines access control as:

“the process of granting or denying specific requests to: (1) obtain and use information and related information processing services; and (2) enter specific physical facilities”

# Access Control Definitions

## 2/2

RFC 4949 (*Internet Security Glossary*) defines access control as:

“a process by which use of system resources is regulated according to a security policy and is permitted only by authorized entities (users, programs, processes, or other systems) according to that policy”

### Basic Security Requirements

- 1 Limit information system access to authorized users, processes acting on behalf of authorized users, or devices (including other information systems).
- 2 Limit information system access to the types of transactions and functions that authorized users are permitted to execute.

### Derived Security Requirements

- 3 Control the flow of CUI in accordance with approved authorizations.
- 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.
- 11 Terminate (automatically) a user session after a defined condition.
- 12 Monitor and control remote access sessions.
- 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.
- 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.

CUI = controlled unclassified information

## Access Control Security Requirements ( SP 800-171)

# Access Control Principles

- In a broad sense, all of computer security is concerned with access control
- RFC 4949 defines computer security as:

“measures that implement and assure security services in a computer system, particularly those that assure access control service”



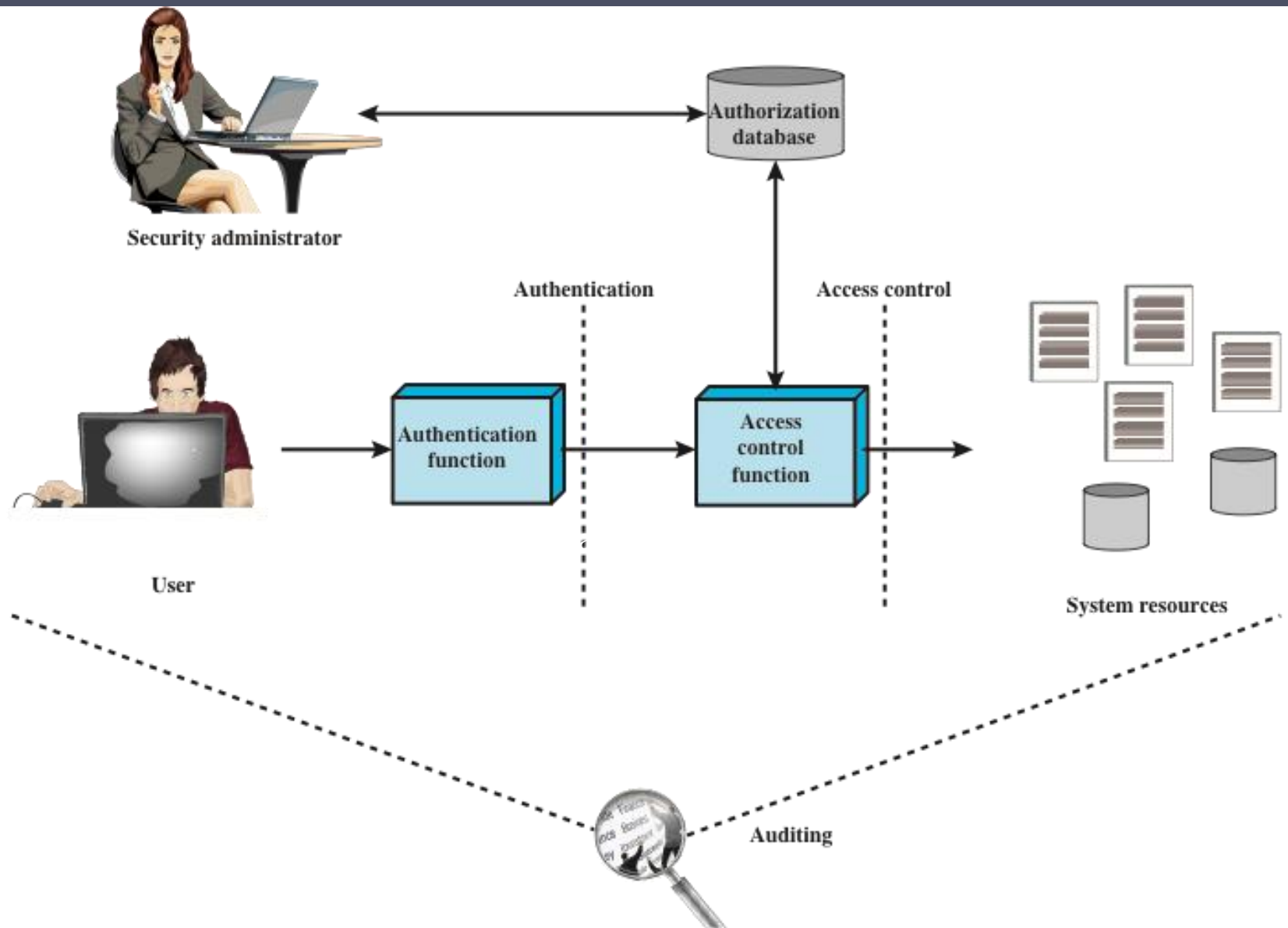


Figure 4.1 Relationship Among Access Control and Other Security Functions

# Access Control Policies

- Discretionary access control (DAC)
  - Controls access based on the identity of the requestor and on access rules (authorizations) stating what requestors are (or are not) allowed to do
  - This policy is termed *discretionary* because an entity might have access rights that permit the entity, by its own volition, to enable another entity to access some resource.
- Mandatory access control (MAC)
  - Controls access based on comparing security labels with security clearances
  - This policy is termed *mandatory* because an entity that has clearance to access a resource may not, just by its own volition, enable another entity to access that resource
- Role-based access control (RBAC)
  - Controls access based on the roles that users have within the system and on rules stating what accesses are allowed to users in given roles
- Attribute-based access control (ABAC)
  - Controls access based on attributes of the user, the resource to be accessed, and current environmental conditions

# Subjects, Objects, and Access Rights

## Subject

An entity capable of accessing objects

### Three classes

- Owner
- Group
- World

## Object

A resource to which access is controlled

Entity used to contain and/or receive information

## Access right

Describes the way in which a subject may access an object

### Could include:

- Read
- Write
- Execute
- Delete
- Create
- Search

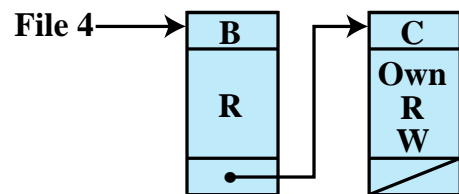
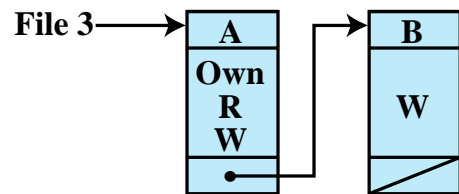
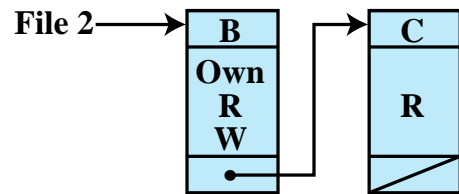
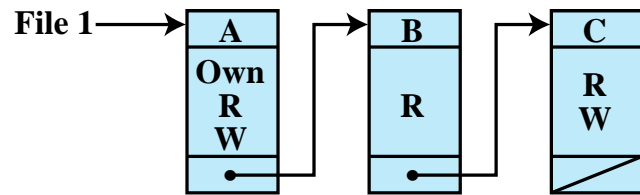
# Discretionary Access Control (DAC)

- Scheme in which an entity may be granted access rights that permit the entity, by its own violation, to 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

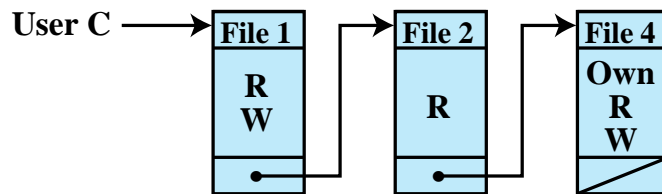
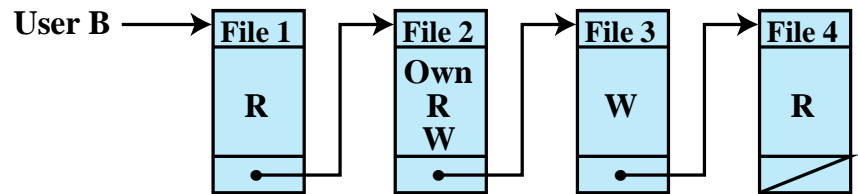
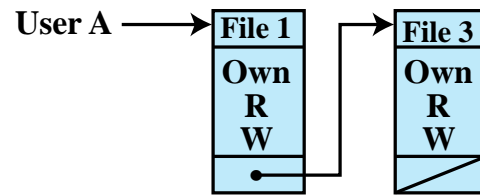
		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix

Figure 4.2 Example of Access Control Structures



(b) Access control lists for files of part (a)



(c) Capability lists for files of part (a)

**Figure 4.2 Example of Access Control Structures**

Subject	Access Mode	Object
A	Own	File 1
A	Read	File 1
A	Write	File 1
A	Own	File 3
A	Read	File 3
A	Write	File 3
B	Read	File 1
B	Own	File 2
B	Read	File 2
B	Write	File 2
B	Write	File 3
B	Read	File 4
C	Read	File 1
C	Write	File 1
C	Read	File 2
C	Own	File 4
C	Read	File 4
C	Write	File 4

Authorization  
Table  
for Files in  
Figure 4.2

# Protection Domains

- Set of objects together with access rights to those objects
- More flexibility when associating capabilities with protection domains
- In terms of the access matrix, a row defines a protection domain
- User can spawn processes with a subset of the access rights of the user
- Association between a process and a domain can be static or dynamic
- In user mode certain areas of memory are protected from use and certain instructions may not be executed
- In kernel mode privileged instructions may be executed and protected areas of memory may be accessed



# UNIX File Access Control

UNIX files are administered using inodes (index nodes)

- Control structures with key information needed for a particular file
- Several file names may be associated with a single inode
- An active inode is associated with exactly one file
- File attributes, permissions and control information are stored in the inode
- On the disk there is an inode table, or inode list, that contains the inodes of all the files in the file system
- When a file is opened its inode is brought into main memory and stored in a memory resident inode table

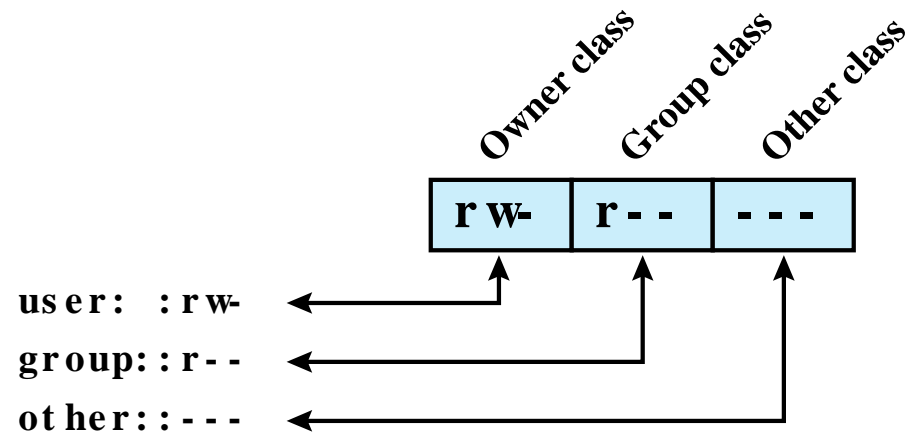
Directories are structured in a hierarchical tree

- May contain files and/or other directories
- Contains file names plus pointers to associated inodes

# UNIX

## File Access Control

- Unique user identification number (user ID)
- Member of a primary group identified by a group ID
- Belongs to a specific group
- 12 protection bits
  - Specify read, write, and execute permission for the owner of the file, members of the group and all other users
- The owner ID, group ID, and protection bits are part of the file's inode



(a) Traditional UNIX approach (minimal access control list)

Figure 4.5 UNIX File Access Control

# Traditional UNIX File Access Control

- “Set user ID”(SetUID)
- “Set group ID”(SetGID)
  - System temporarily uses rights of the file owner/group in addition to the real user’s rights when making access control decisions
  - Enables privileged programs to access files/resources not generally accessible
- Sticky bit
  - When applied to a directory it specifies that only the owner of any file in the directory can rename, move, or delete that file
- Superuser
  - Is exempt from usual access control restrictions
  - Has system-wide access

# Access Control Lists (ACLs) in UNIX

## Modern UNIX systems support ACLs

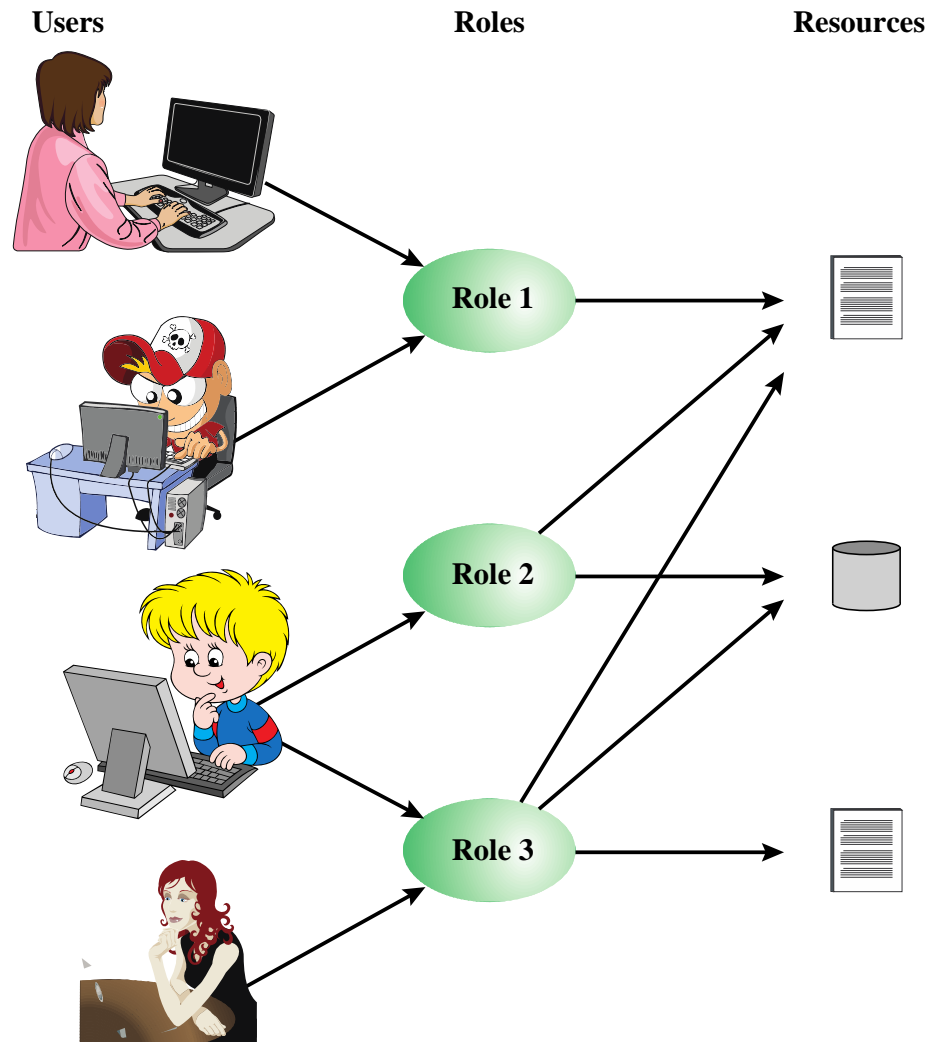
- FreeBSD, OpenBSD, Linux, Solaris

### FreeBSD

- Setfacl command assigns a list of UNIX user IDs and groups
- Any number of users and groups can be associated with a file
- Read, write, execute protection bits
- A file does not need to have an ACL
- Includes an additional protection bit that indicates whether the file has an extended ACL

When a process requests access to a file system object two steps are performed:

- Step 1 selects the most appropriate ACL
- Step 2 checks if the matching entry contains sufficient permissions



**Figure 4.6 Users, Roles, and Resources**

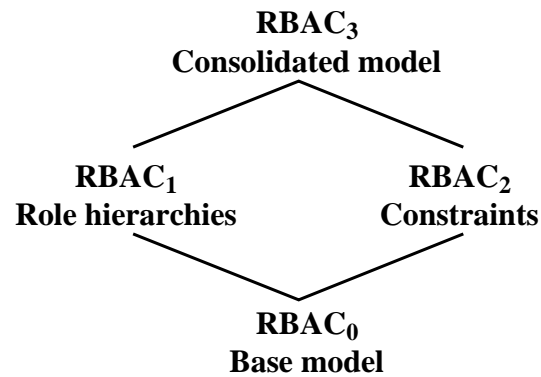
		$R_1$	$R_2$	$\cdot$	$\cdot$	$R_n$			
$U_1$		$\times$							
$U_2$		$\times$							
$U_3$			$\times$			$\times$			
$U_4$						$\times$			
$U_5$						$\times$			
$U_6$						$\times$			
$\cdot$									
$\cdot$									
$\cdot$									
$U_m$		$\times$							

		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 *
	$\cdot$									
	$\cdot$									
$R_n$			control		write	stop				

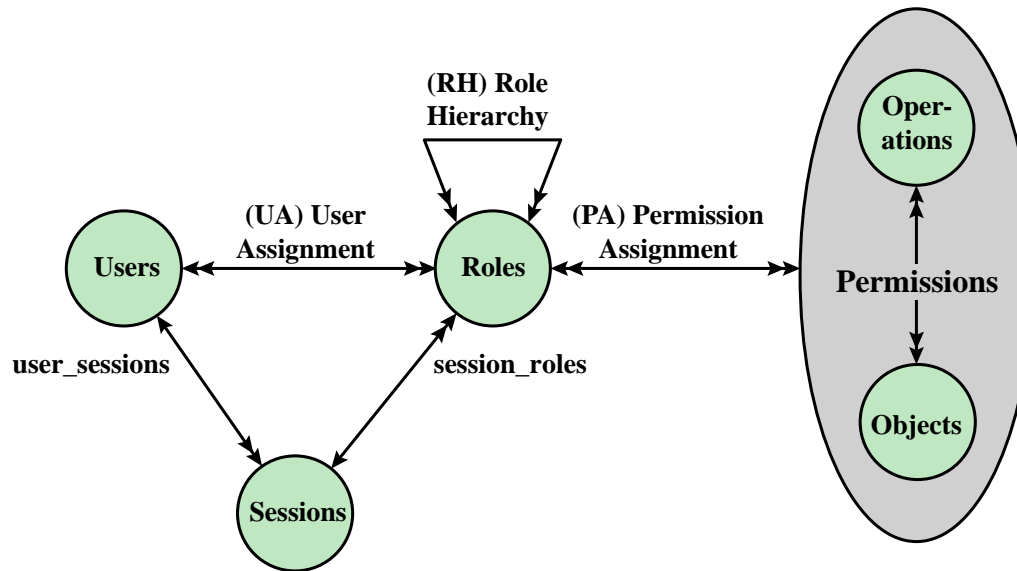
Figure 4.7 Access Control Matrix Representation of RBAC

# Scope RBAC Models

Models	Hierarchies	Constraints
RBAC <sub>0</sub>	No	No
RBAC <sub>1</sub>	Yes	No
RBAC <sub>2</sub>	No	Yes
RBAC <sub>3</sub>	Yes	Yes



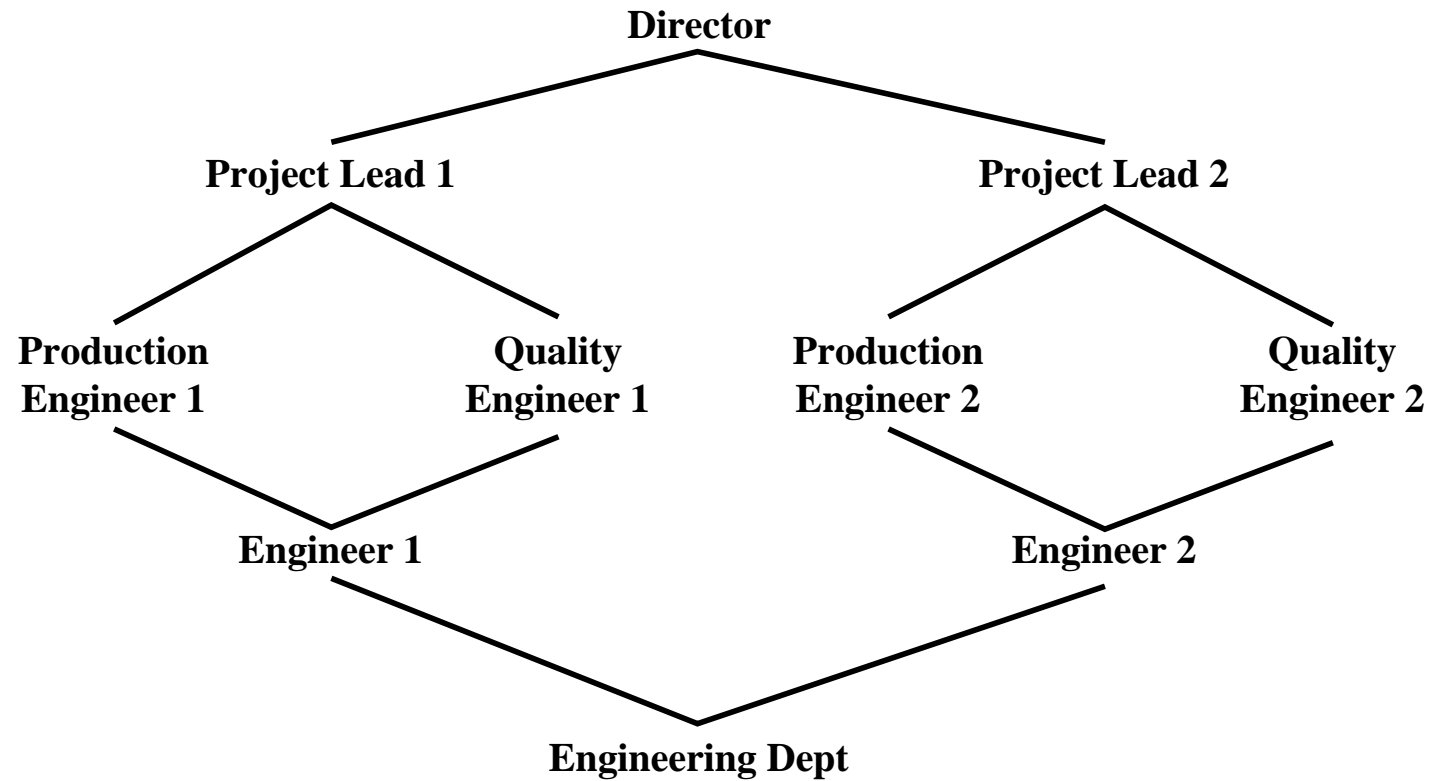
(a) Relationship among RBAC models



(b) RBAC models

**Figure 4.8 A Family of Role-Based Access Control Models.**





**Figure 4.9 Example of Role Hierarchy**

# Constraints - RBAC

- Provide a means of adapting RBAC to the specifics of administrative and security policies of an organization
- A defined relationship among roles or a condition related to roles
- Types:

## Mutually exclusive roles

- A user can only be assigned to one role in the set (either during a session or statically)
- Any permission (access right) can be granted to only one role in the set

## Cardinality

- Setting a maximum number with respect to roles

## Prerequisite roles

- Dictates that a user can only be assigned to a particular role if it is already assigned to some other specified role

# Attribute-Based Access Control (ABAC)

Can define authorizations that express conditions on properties of both the resource and the subject

Strength is its flexibility and expressive power

Main obstacle to its adoption in real systems has been concern about the performance impact of evaluating predicates on both resource and user properties for each access

Web services have been pioneering technologies through the introduction of the eXtensible Access Control Markup Language (XAMCL)

There is considerable interest in applying the model to cloud services

# ABAC Model: Attributes

## 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
- A subject's role can also be viewed as an attribute.

## 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

## Environment attributes

- Describe the operational, technical, and even situational environment or context in which the information access occurs
- These attributes have so far been largely ignored in most access control policies

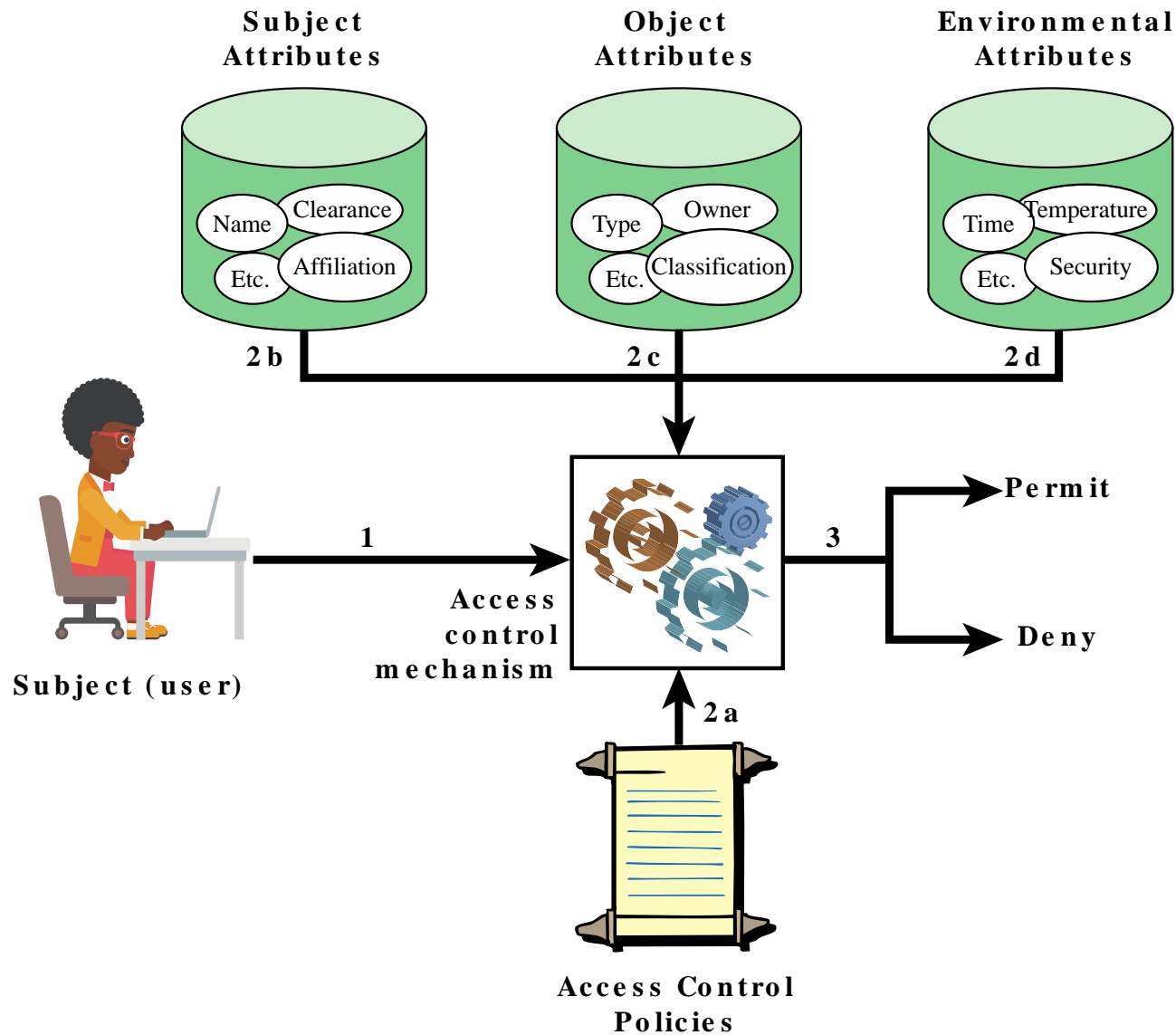
# ABAC

Distinguishable because it controls access to objects by evaluating rules against the attributes of entities, operations, and the environment relevant to a request

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

Systems are capable of enforcing DAC, RBAC, and MAC concepts

Allows an unlimited number of attributes to be combined to satisfy any access control rule



**Figure 4.10 ABAC Scenario**

# ABAC Policies

A policy is a set of rules and relationships that govern allowable behavior within an organization, based on the privileges of subjects and how resources or objects are to be protected under which environment conditions

Typically written from the perspective of the object that needs protecting and the privileges available to subjects

Privileges represent the authorized behavior of a subject and are defined by an authority and embodied in a policy

Other terms commonly used instead of privileges are: rights, authorizations, and entitlements

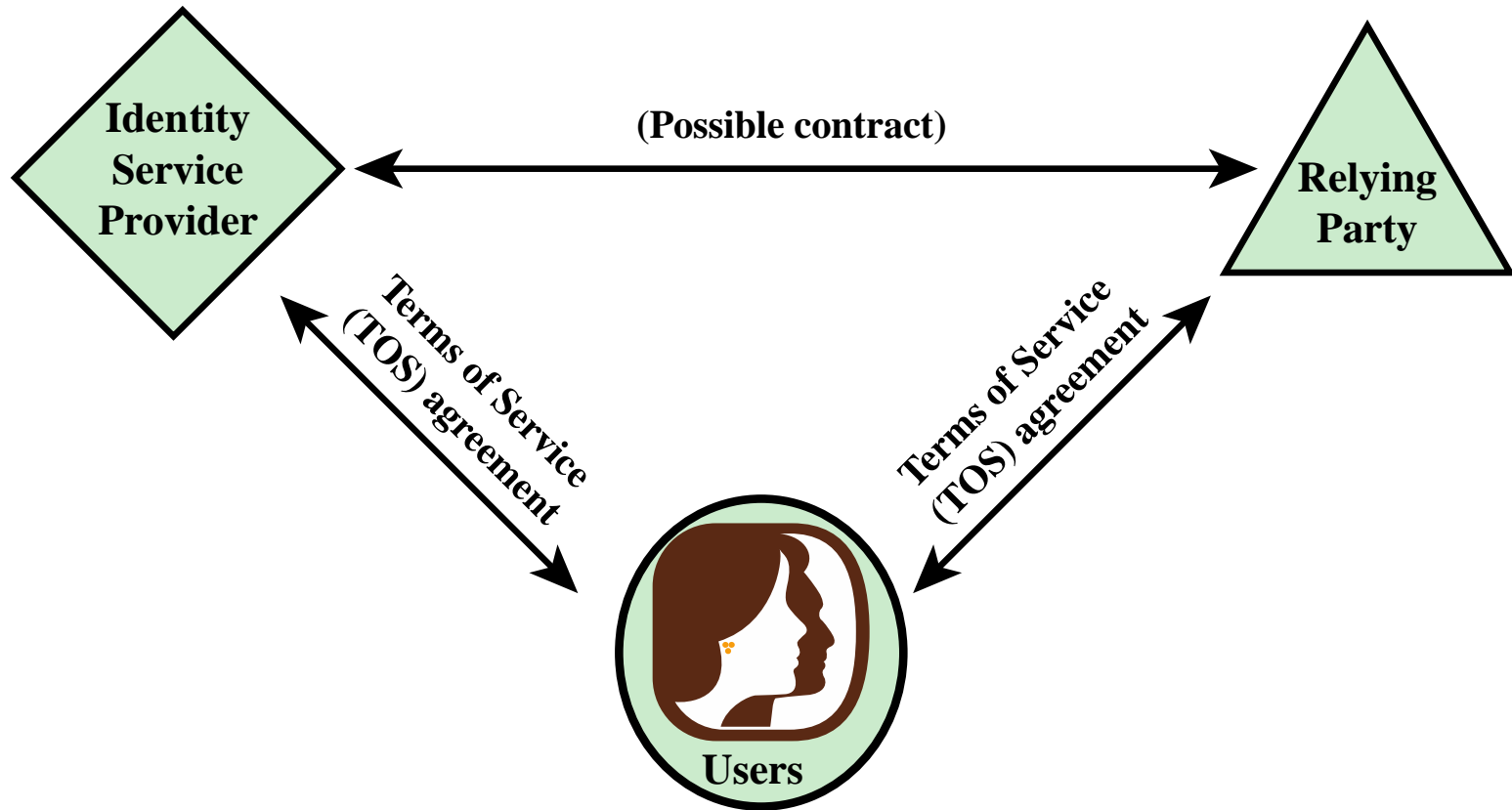
# Identity, Credential, and Access Management (ICAM)

- A comprehensive approach to managing and implementing digital identities, credentials, and access control
- Developed by the U.S. government
- Designed to:
  - Create trusted digital identity representations of individuals and nonperson entities (NPEs)
  - Bind those identities to credentials that may serve as a proxy for the individual or NPE in access transactions
    - A credential is an object or data structure that authoritatively binds an identity to a token possessed and controlled by a subscriber
  - Use the credentials to provide authorized access to an agency's resources



# Identity Federation

- Term used to describe the technology, standards, policies, and processes that allow an organization to trust digital identities, identity attributes, and credentials created and issued by another organization
- Addresses two questions:
  - How do you trust identities of individuals from external organizations who need access to your systems
  - How do you vouch for identities of individuals in your organization when they need to collaborate with external organizations



**(a) Traditional triangle of parties involved in an exchange of identity information**

# Open Identity Trust Framework

## OpenID

- An open standard that allows users to be authenticated by certain cooperating sites using a third party service

## OIDF

- OpenID Foundation is an international nonprofit organization of individuals and companies committed to enabling, promoting, and protecting OpenID technologies

## ICF

- Information Card Foundation is a nonprofit community of companies and individuals working together to evolve the Information Card ecosystem

## OITF

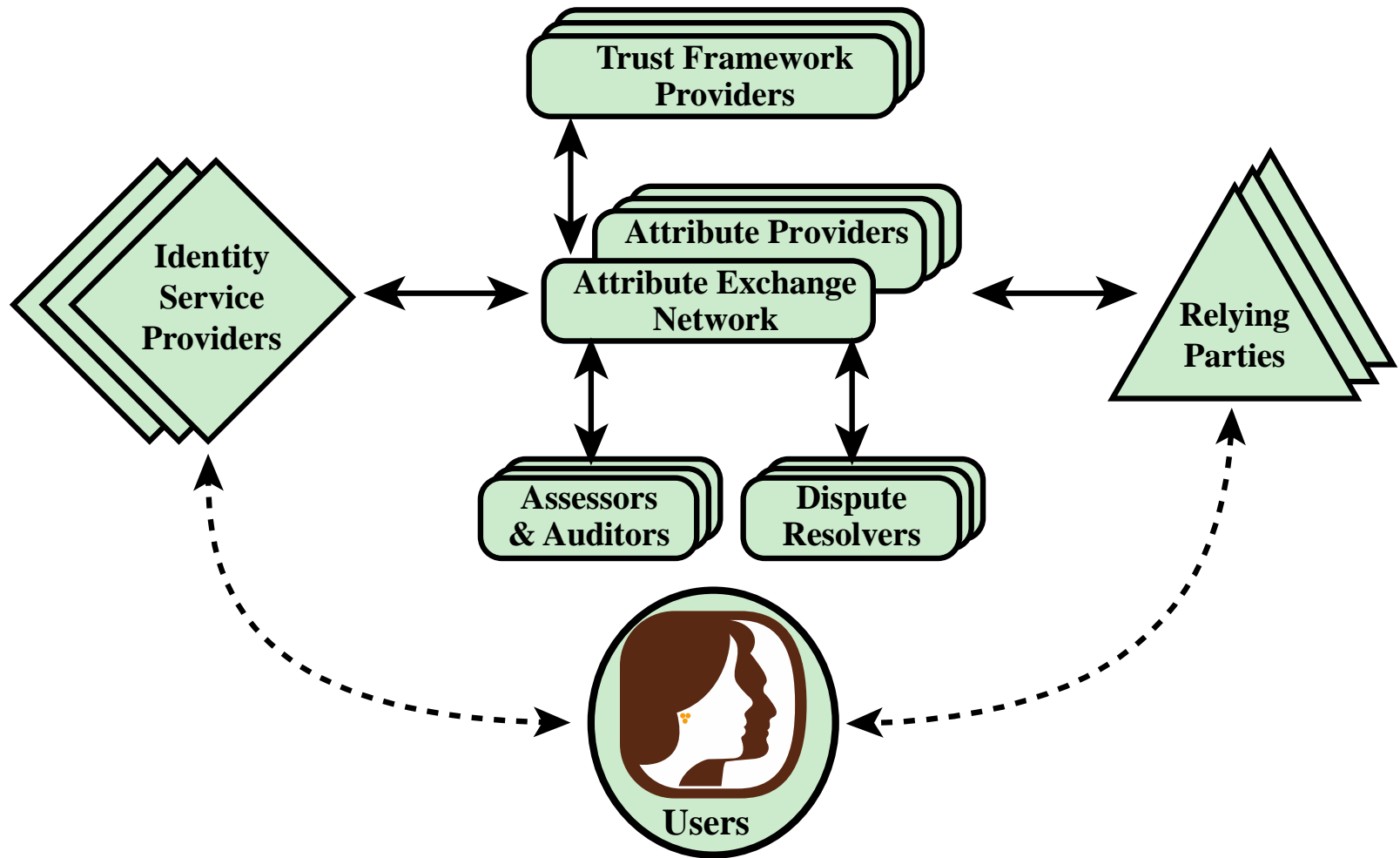
- Open Identity Trust Framework is a standardized, open specification of a trust framework for identity and attribute exchange, developed jointly by OIDF and ICF

## OIX

- Open Identity Exchange Corporation is an independent, neutral, international provider of certification trust frameworks conforming to the OITF model

## AXN

- Attribute Exchange Network is an online Internet-scale gateway for identity service providers and relying parties to efficiently access user asserted, permissioned, and verified online identity attributes in high volumes at affordable costs



(B) Identity attribute exchange elements

**Figure 4.13 Identity Information Exchange Approaches**