

Access Control

What is Access Control?

- NIST IR 7298, Glossary of Key Information Security Terms, 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.
- 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.

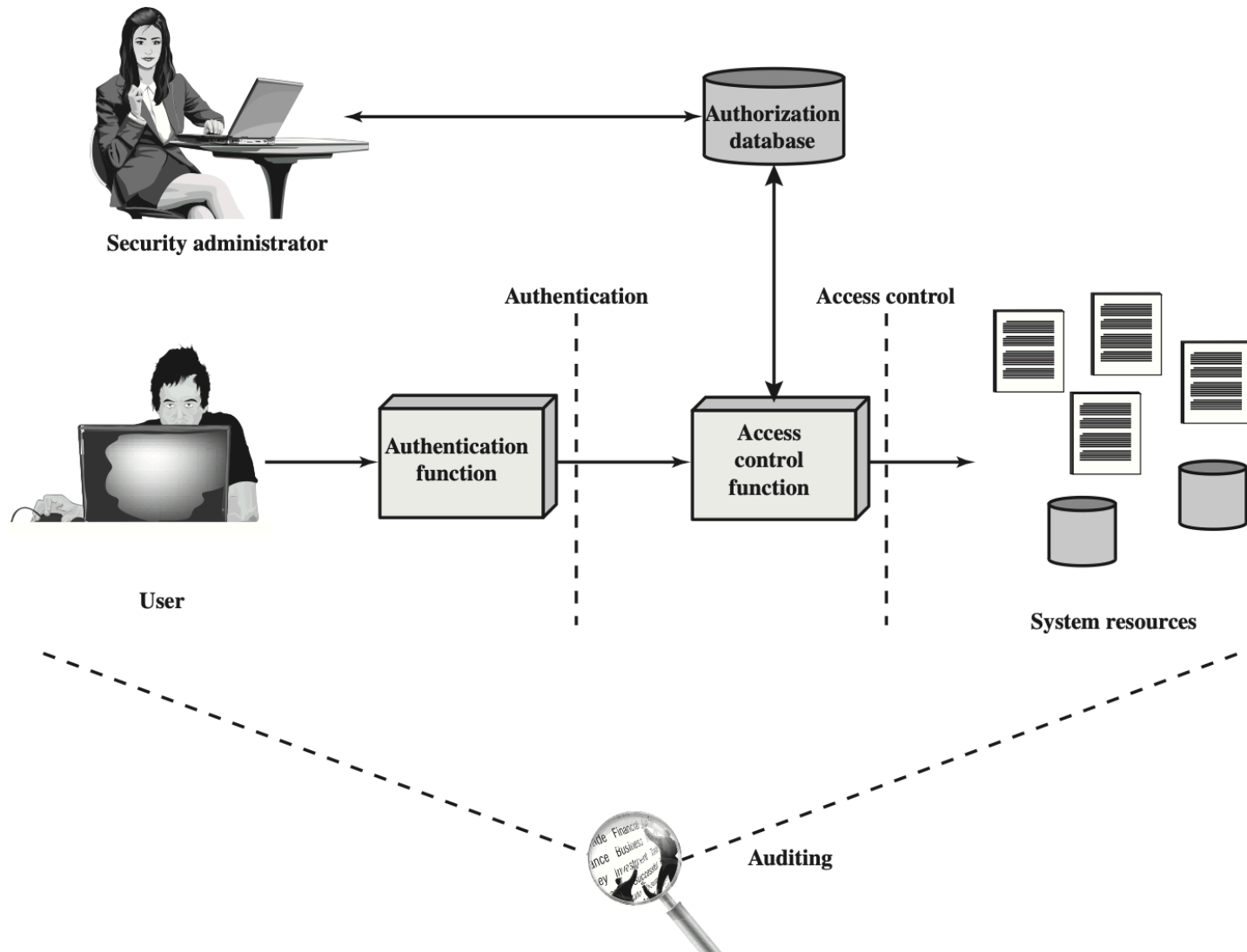
Access Control

- We can view **access control** as the **central element of computer security**. The principal objectives of computer security are to prevent unauthorized users from gaining access to resources, to prevent legitimate users from accessing resources in an unauthorized manner, and to enable legitimate users to access resources in an authorized manner.

Access Control Context

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

Relationship Among Access Control and Other Security Functions



Access Control Policies

- [Discretionary access control](#) (DAC): based on the identity of the requestor and access rules
- [Mandatory access control](#) (MAC): based on comparing security labels with security clearances (mandatory: one with access to a resource cannot pass to others)
- [Role-based access control](#) (RBAC): based on user roles
- [Attribute-based access control](#): based on the attributes of the user, the resources and the current environment

Access Control Requirements

- **Reliable input**: a mechanism to authenticate
- **Fine and coarse specifications**: regulate access at varying levels (e.g., an attribute or entire DB)
- **Least privilege**: min authorization to do its work
- **Separation of duty**: divide steps among different individuals
- **Open and closed policies**: accesses specifically authorized or all accesses except those prohibited
- **Administrative policies**: who can add, delete, modify rules

Access Control Elements

- Subject: entity that can access objects
 - a process representing user/application
 - often have 3 classes: **owner**, **group**, **world**
- Object: access controlled resource
 - e.g. files, directories, records, programs etc
 - number/type depend on environment
- Access right: way in which subject accesses an object
 - e.g. read, write, execute, delete, create, search

Discretionary Access Control

- Often provided using an access matrix
 - lists subjects in one dimension (rows)
 - lists objects in the other dimension (columns)
 - each entry specifies access rights of the specified subject to that object
- Access matrix is often sparse
- Can decompose by either row or column

Access Control Structures

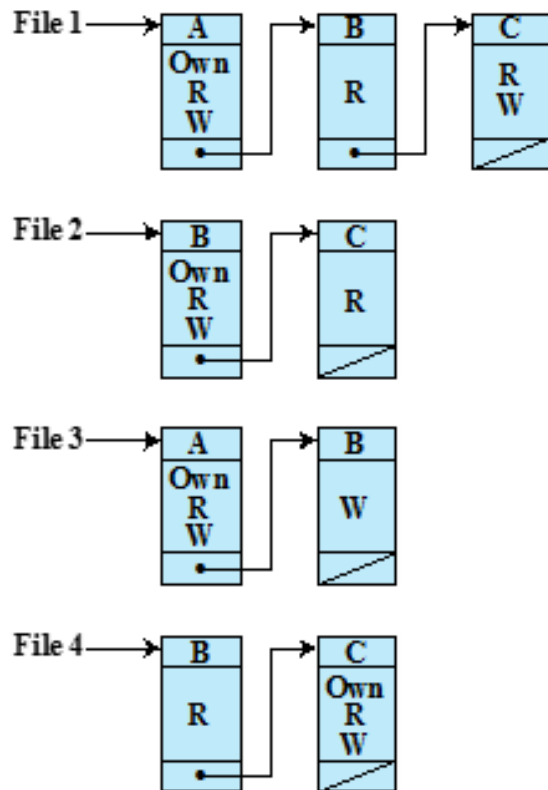
- Access control lists (decomposed by column)
- Capability tickets (decomposed by row)
- See page 119
- Also see alternative table representation on page 120 (tabular but not sparse)

An access matrix

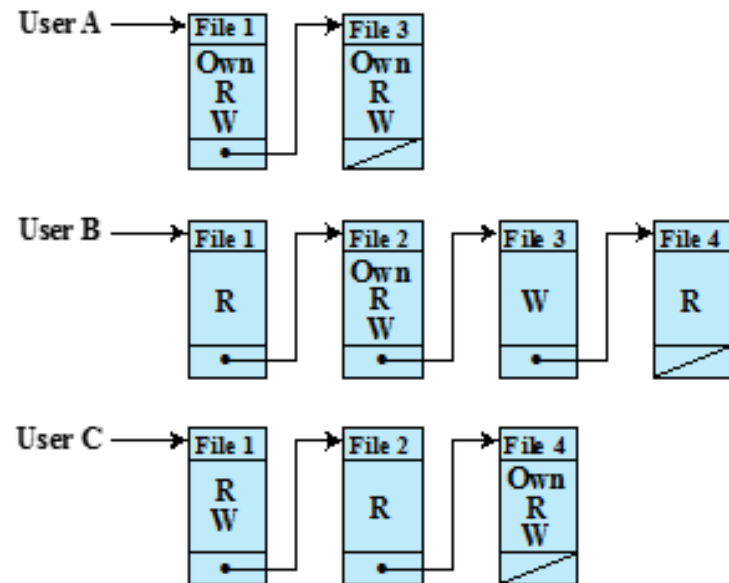
		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix

Access matrix data structures



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



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

Alternate Authorization Table

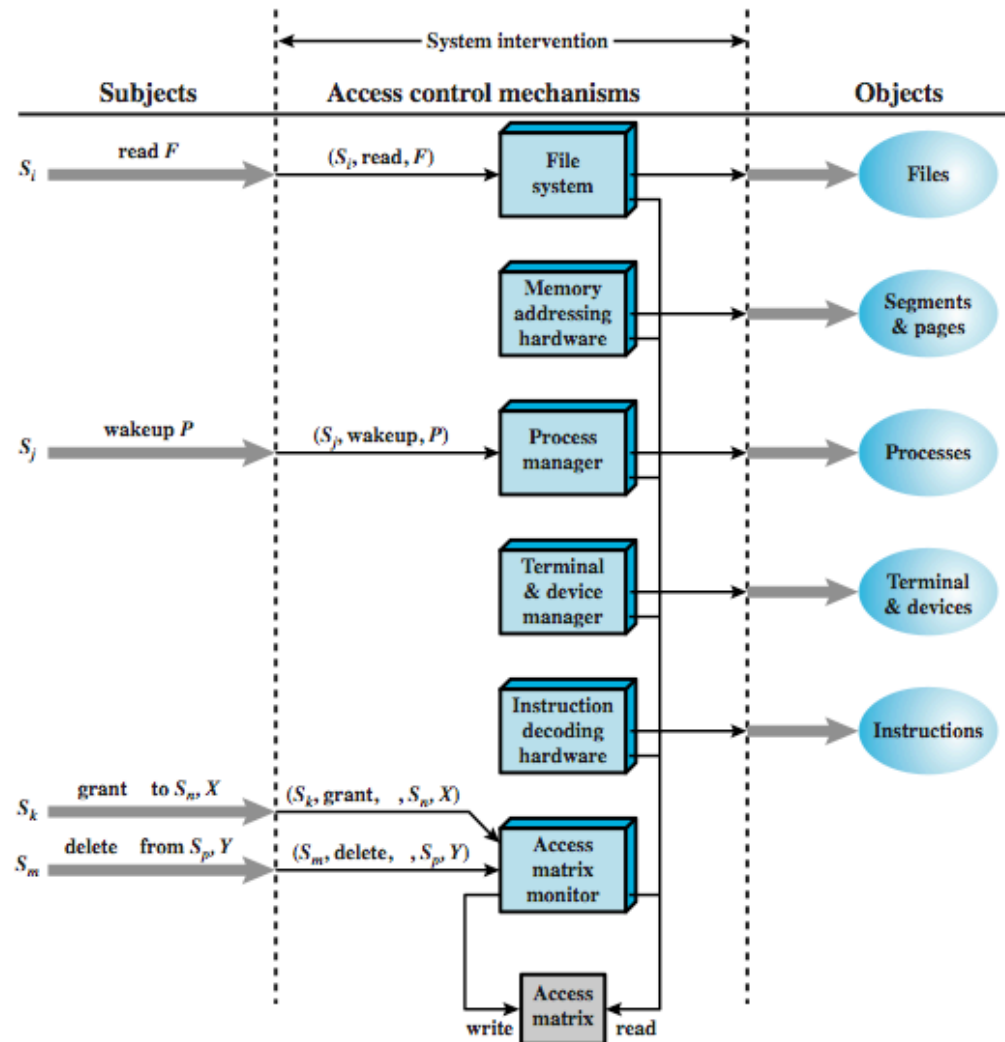
Subject	Access Mode	Object
A	Own	File 1
A	Read	File 1
A	Write	File 1
A	Own	File 3
A	Read	File 3
A	Write	File 3
B	Read	File 1
B	Own	File 2
B	Read	File 2
B	Write	File 2
B	Write	File 3
B	Read	File 4
C	Read	File 1
C	Write	File 1
C	Read	File 2
C	Own	File 4
C	Read	File 4
C	Write	File 4

An Access Control Model

		OBJECTS								
		subjects			files		processes		disk drives	
		S ₁	S ₂	S ₃	F ₁	F ₁	P ₁	P ₂	D ₁	D ₂
SUBJECTS	S ₁	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
	S ₂		control		write *	execute			owner	seek *
	S ₃			control		write	stop			

* - copy flag set

Access Control Function



Access control system commands

Rule	Command (by S_o)	Authorization	Operation
R1	transfer $\begin{Bmatrix} \alpha^* \\ \alpha \end{Bmatrix}$ to S, X	' α^* ' in $A[S_o, X]$	store $\begin{Bmatrix} \alpha^* \\ \alpha \end{Bmatrix}$ in $A[S, X]$
R2	grant $\begin{Bmatrix} \alpha^* \\ \alpha \end{Bmatrix}$ to S, X	'owner' in $A[S_o, X]$	store $\begin{Bmatrix} \alpha^* \\ \alpha \end{Bmatrix}$ in $A[S, X]$
R3	delete α from S, X	'control' in $A[S_o, S]$ or 'owner' in $A[S_o, X]$	delete α from $A[S, X]$
R4	$w \leftarrow$ read S, X	'control' in $A[S_o, S]$ or 'owner' in $A[S_o, X]$	copy $A[S, X]$ into w
R5	create object X	None	add column for X to A ; store 'owner' in $A[S_o, X]$
R6	destroy object X	'owner' in $A[S_o, X]$	delete column for X from A
R7	create subject S	none	add row for S to A ; execute create object S ; store 'control' in $A[S, S]$
R8	destroy subject S	'owner' in $A[S_o, S]$	delete row for S from A ; execute destroy object S

Protection Domains: More Useful

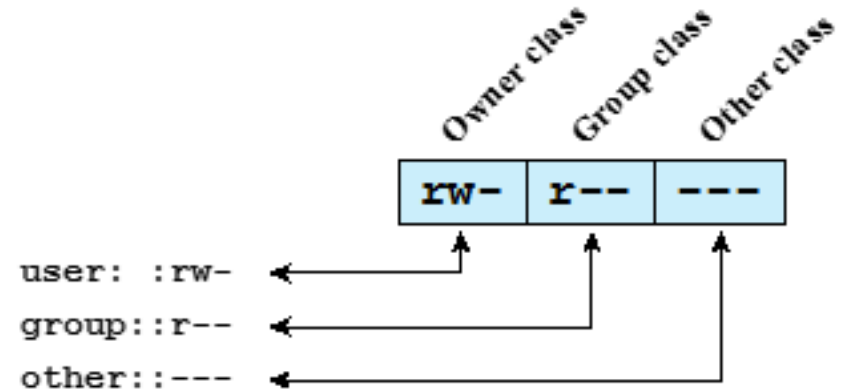
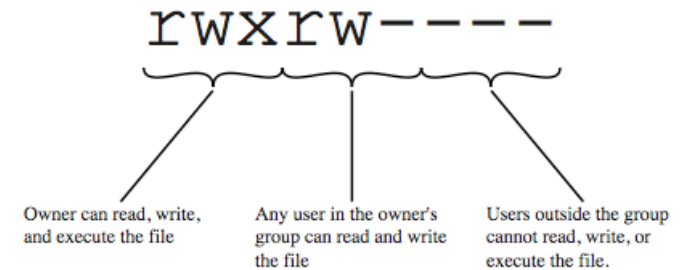
- Set of objects together with access rights to those objects
- More flexibility when associating capabilities with protection domains
- In terms of the access matrix, a row defines a protection domain
- User can spawn processes with a subset of the access rights of the user
- Association between a process 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 Concepts

- UNIX files administered using inodes (index nodes)
- An inode:
 - control structure with key info on file (attributes, permissions, ...)
 - on a disk: an inode table for all files
 - when a file is opened, its inode is brought to RAM
- Directories form a hierarchical tree
 - may contain files or other directories
 - are a file of names and inode numbers

UNIX File Access Control

- Unique user identification number (user ID)
- Member of a primary group identified by a group ID
- 12 protection bits
 - 9 specify read, write, and execute permission for the owner of the file, members of the group and all other users
 - 2 specify SetID, SetGID
 - 1 is the sticky bit (only owner can remove, delete, ..., a directory)
- The owner ID, group ID, and protection bits are part of the file's inode



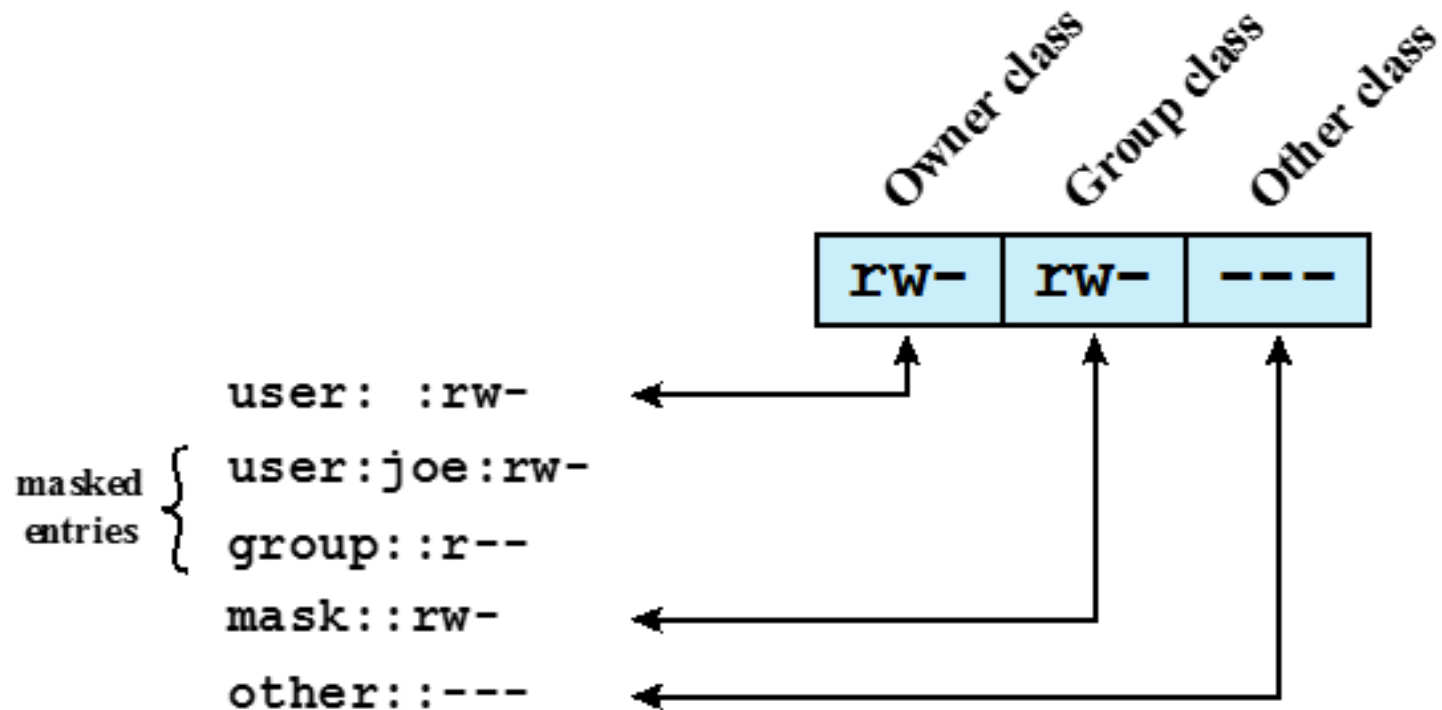
UNIX File Access Control

- “set user ID”(SetUID) or “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
 - on directory limits rename/move/delete to owner
- Superuser
 - is exempt from usual access control restrictions

UNIX Access Control Lists

- Modern UNIX systems support ACLs
- Can specify any number of additional users/groups and associated rwx permissions
- When access is required
 - select most appropriate ACL
 - owner, named users, owning/named groups, others
 - check if have sufficient permissions for access

UNIX extended access control list



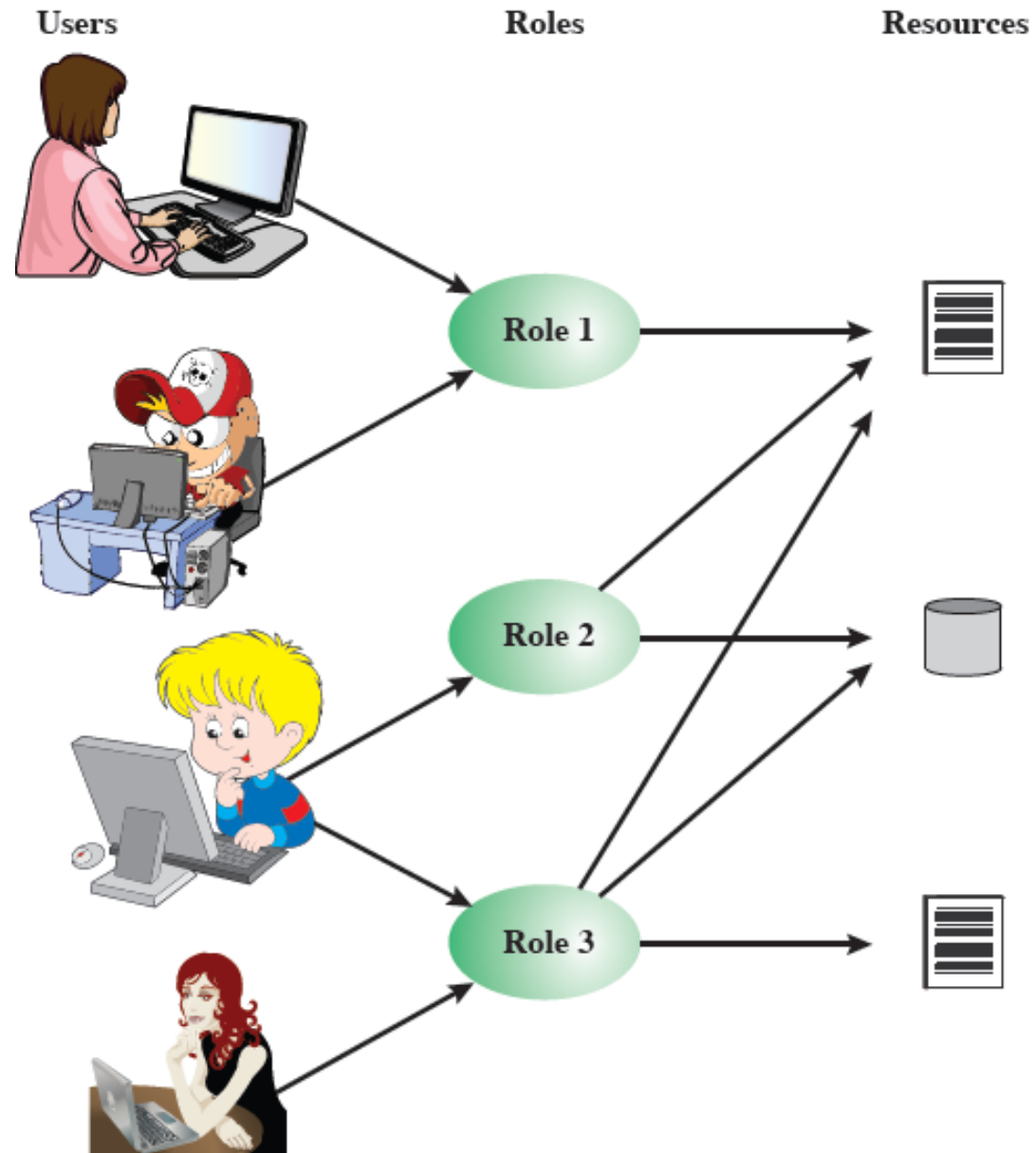
(b) Extended access control list

Role-Based Access Control

Access based on
'role', not identity

Many-to-many
relationship between
users and roles

Roles often static



Role-Based Access Control

Role-users and roles-object access matrix

	R_1	R_2	...	R_n
U_1	×			
U_2	×			
U_3		×		×
U_4				×
U_5				×
U_6				×
...				
U_m	×			

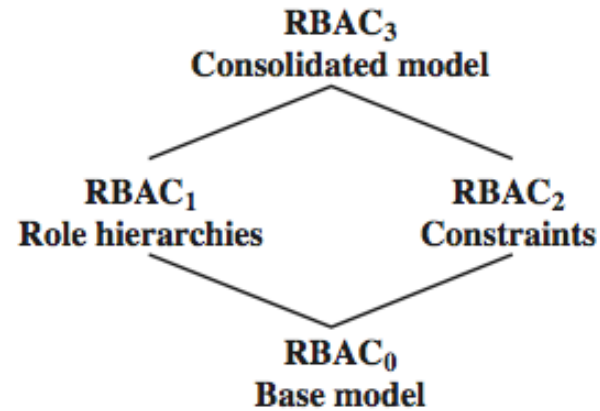
		OBJECTS								
		R ₁	R ₂	R _n	F ₁	F ₁	P ₁	P ₂	D ₁	D ₂
ROLES	R ₁	control	owner	owner control	read *	read owner	wakeup	wakeup	seek	owner
	R ₂		control		write *	execute			owner	seek *
	•									
	•									
	R _n			control		write	stop			

General RBAC, Variations

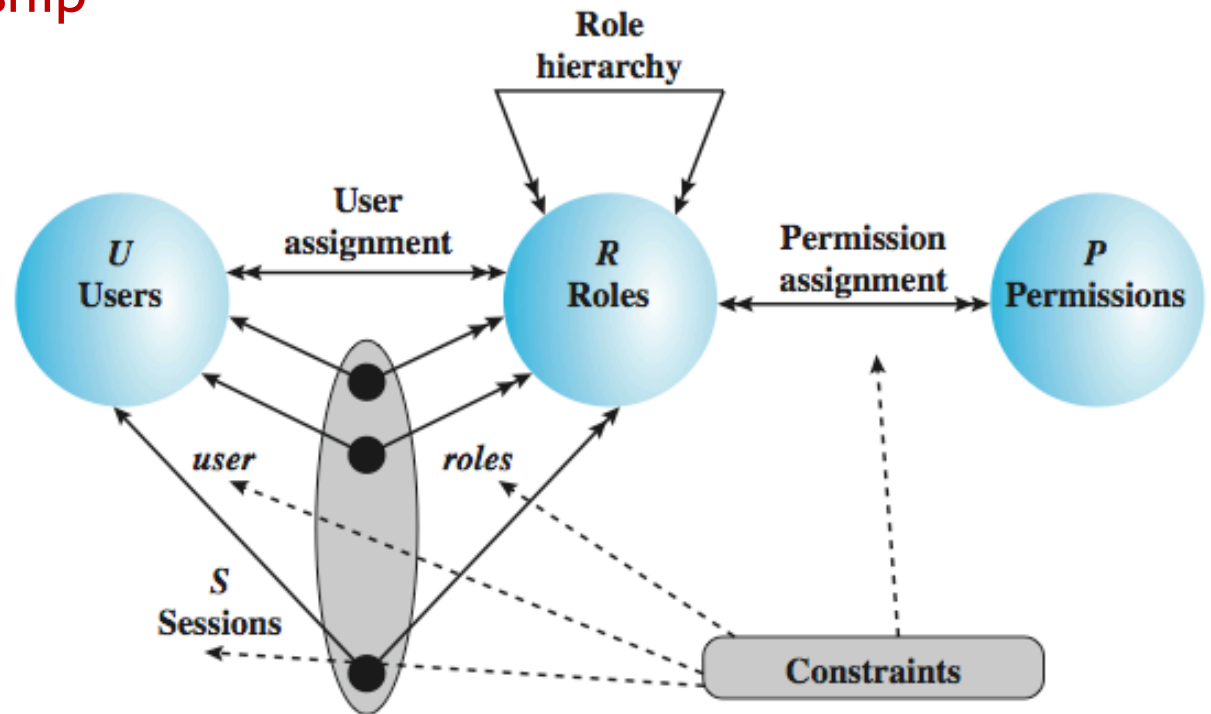
- A family of RBAC with four models
 1. RBAC0: min functionality
 2. RBAC1: RBAC0 plus role (permission) inheritance
 3. RBAC2: RBAC0 plus constraints (restrictions)
 4. RBAC3: RBAC0 plus all of the above
- RBAC0 entities
 - User: an individual (with UID) with access to system
 - Role: a named job function (tells authority level)
 - Permission: equivalent to access rights
 - Session: a mapping between a user and set of roles to which a user is assigned

Role-Based Access Control

Double arrow: 'many' relationship
Single arrow: 'one' relationship

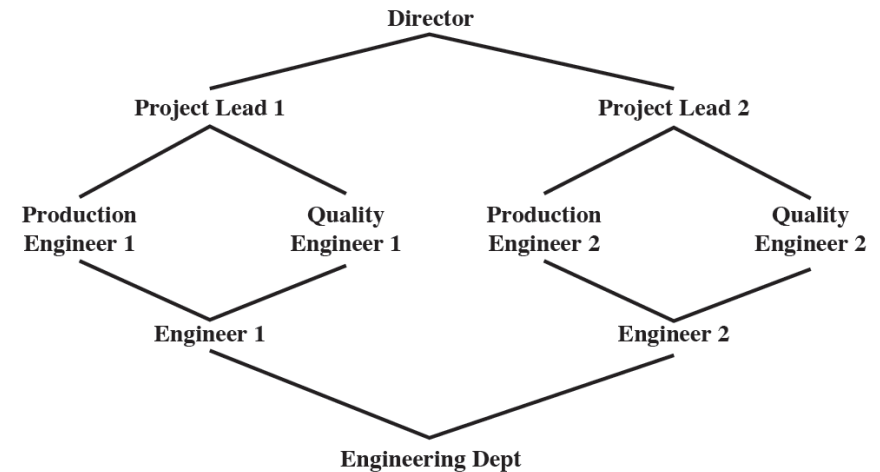


(a) Relationship among RBAC models



(b) RBAC models

Example of role hierarchy



- 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

Constraints

- A condition (restriction) on a role or between roles
 - 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
 - Cardinality: set a maximum number (of users) wrt a role (e.g., a department chair role)
 - Prerequisite role: a user can be assigned a role only if that user already has been assigned to some other role

Attribute-Based Access Control

- Fairly recent
- Define authorizations that express conditions on properties of both the resource and the subject
 - Each resource has an attribute (e.g., the subject that created it)
 - A single rule states ownership privileges for the creators
- Strength: its flexibility and expressive power
- Considerable interest in applying the model to cloud services

Types of attributes

- Subject attributes
- Object attributes
- Environment 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
 - Name
 - Organization
 - Job title

Object attribute

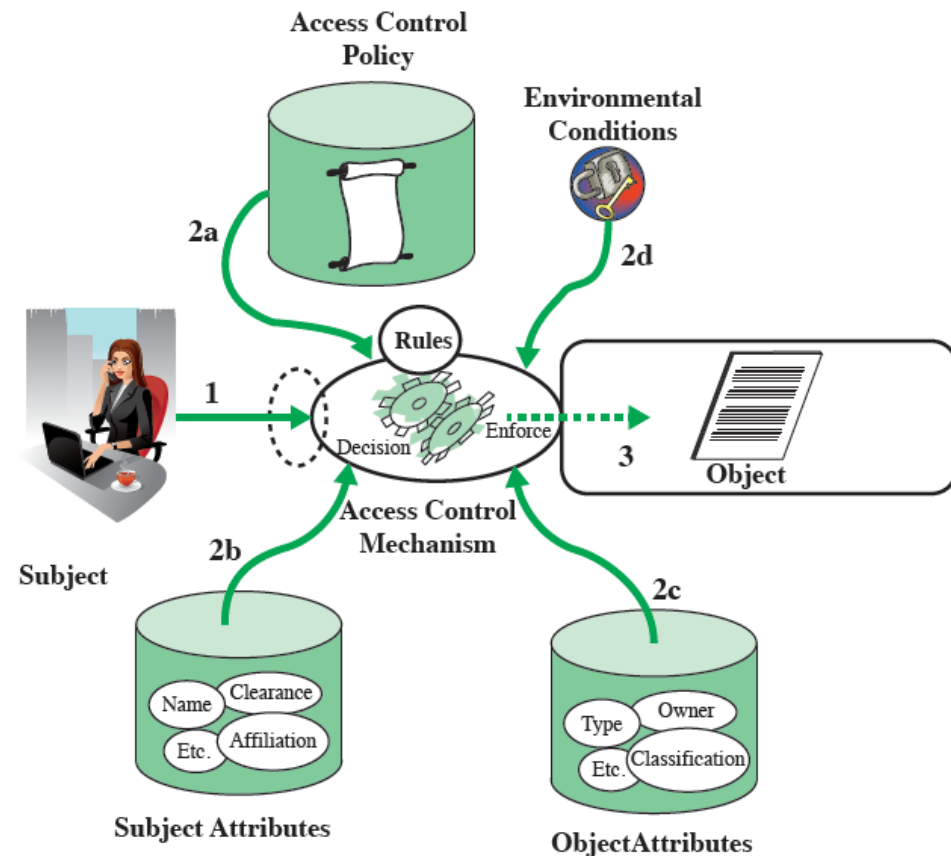
- An object (or resource) is a passive information system-related entity containing or receiving information
- Objects have attributes that can be leveraged to make access control decisions
 - Title
 - Author
 - Date

Environment Attributes

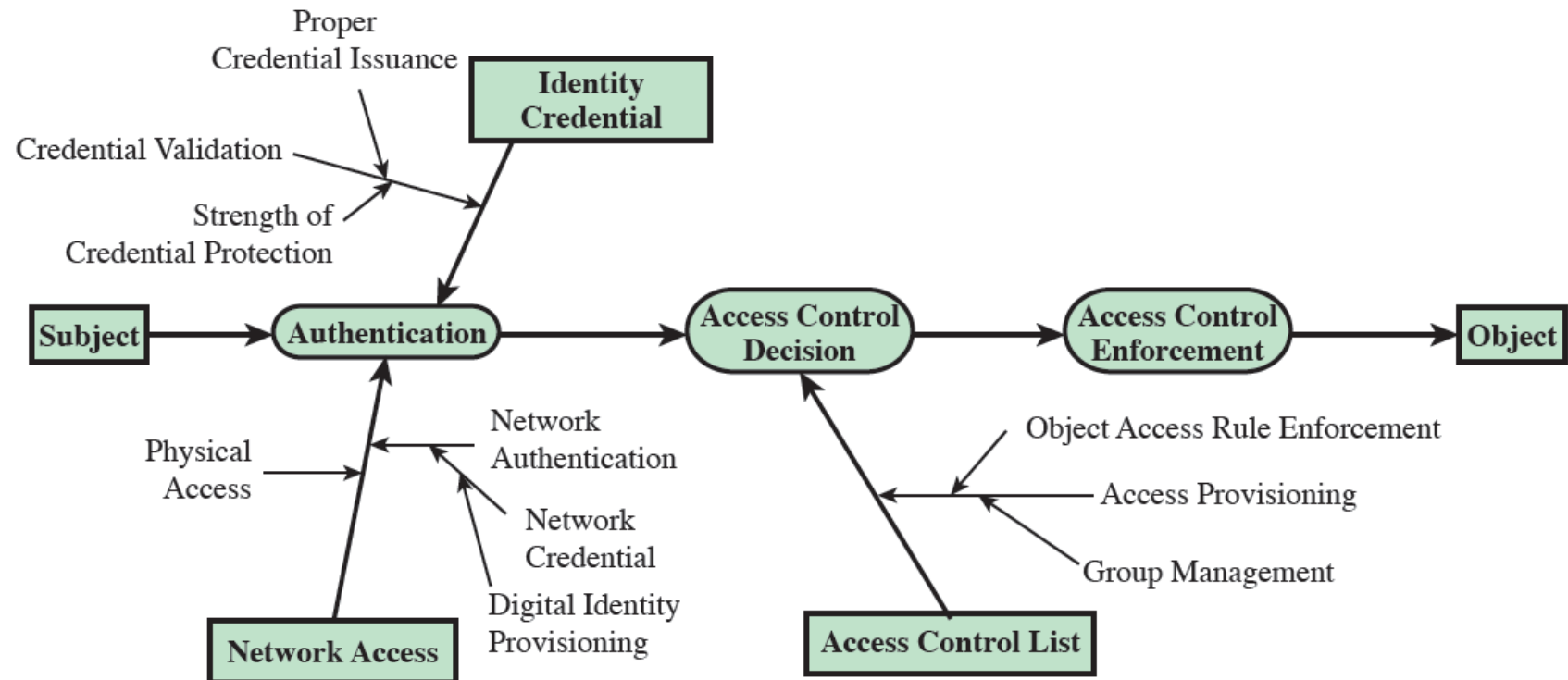
- Describe the operational, technical, and even situational environment or context in which the information access occurs
 - Current date
 - Current virus/hacker activities
 - Network security level
 - *Not associated with a resource or subject*
- These attributes have so far been largely ignored in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

Sample ABAC scenario

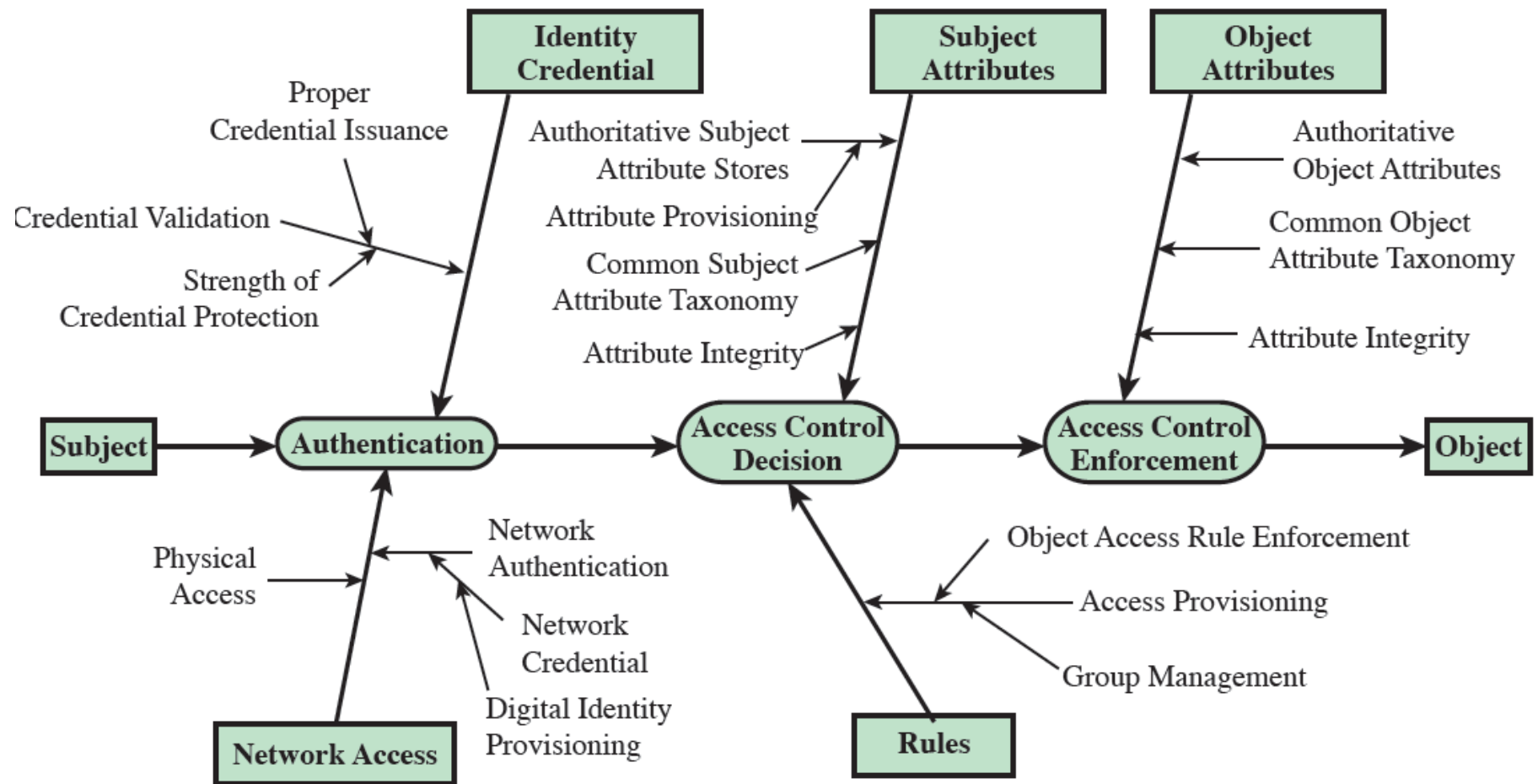
1. A subject requests access to an object
2. AC is governed by a set of rules (2a): assesses the attr of subject (2b), object (2c) and env (2d)
3. AC grants subject access to object if authorized



ACL vs ABAC trust relationships



ACL vs ABAC trust relationships



Identity, Credential, and Access Management (ICAM)

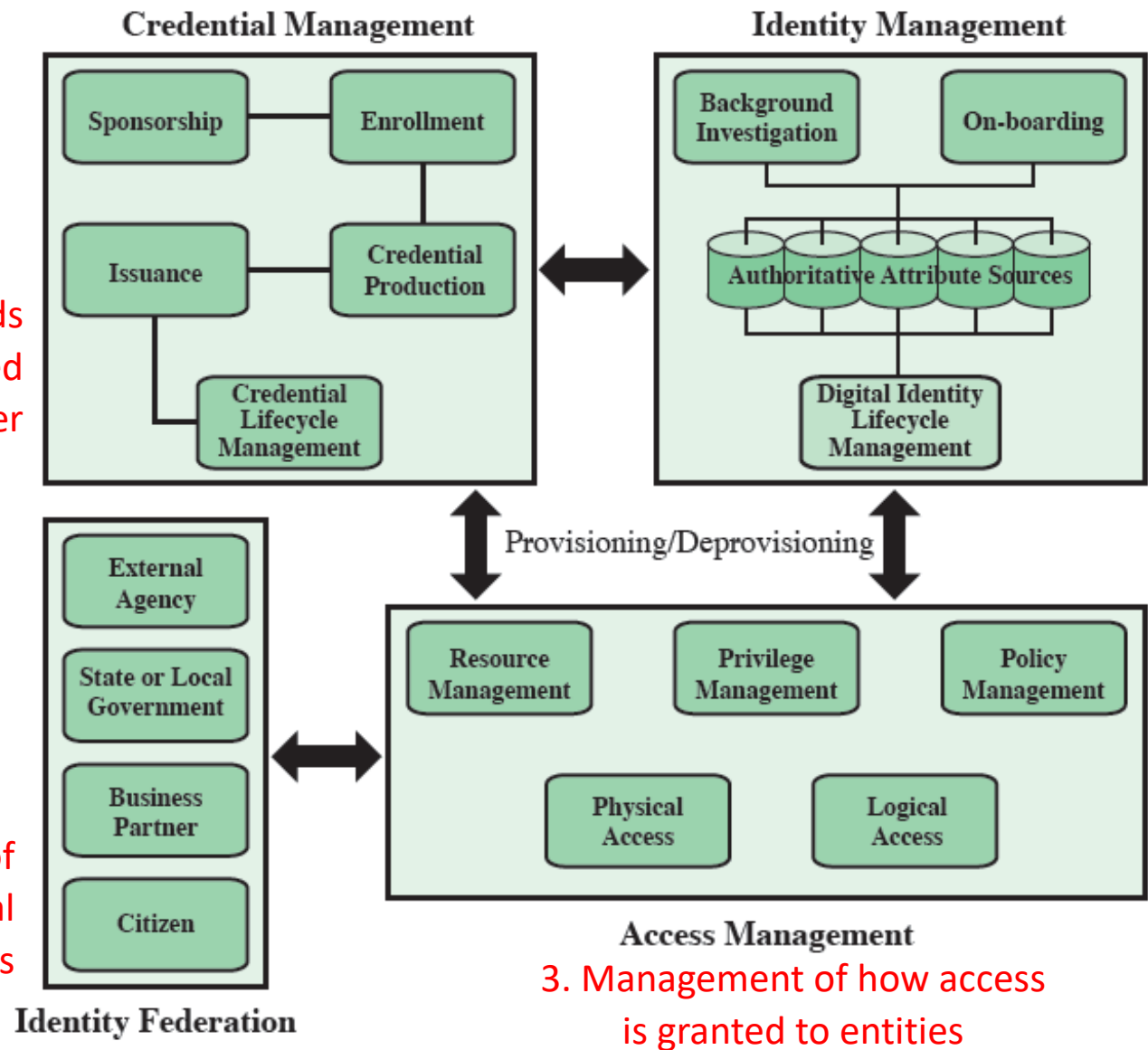
- A comprehensive approach to managing and implementing digital identities, credentials, and access control
- Developed by the U.S. government
- Designed to create trusted digital identity representations of individuals and nonperson entities (NPEs)
- A credential is an object or data structure that 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

ICAM

2. Data structures that binds a token possessed by a subscriber

4. Identity verification of individuals from external organizations

1. Connects digital identity to individuals



Trust frameworks

- Skip

Case study: RBAC system for a bank

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 has more access than A (strict ordering)
- Inheritance makes tables simpler

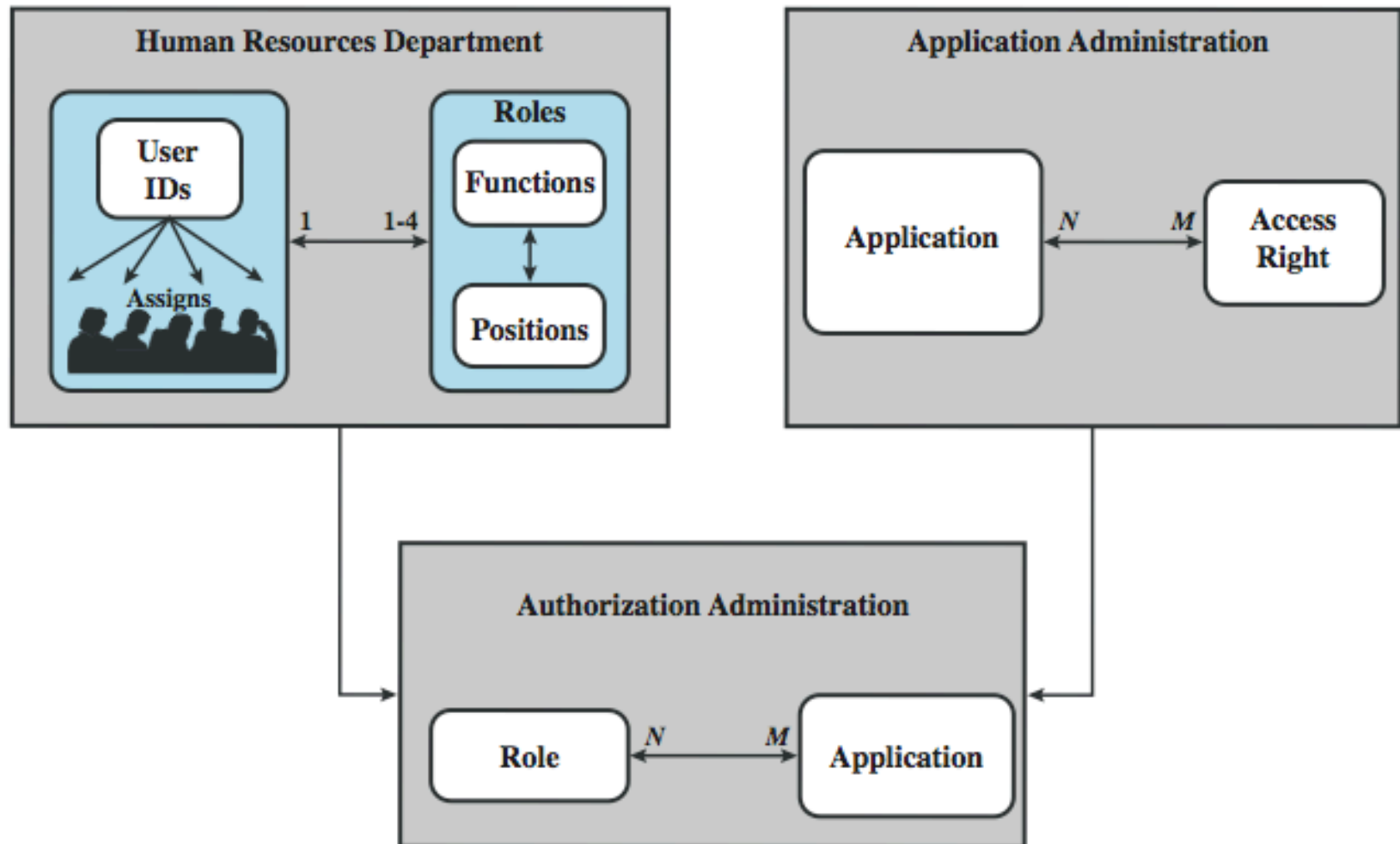
(b) Permission Assignments

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
...

Case study: RBAC system for a bank

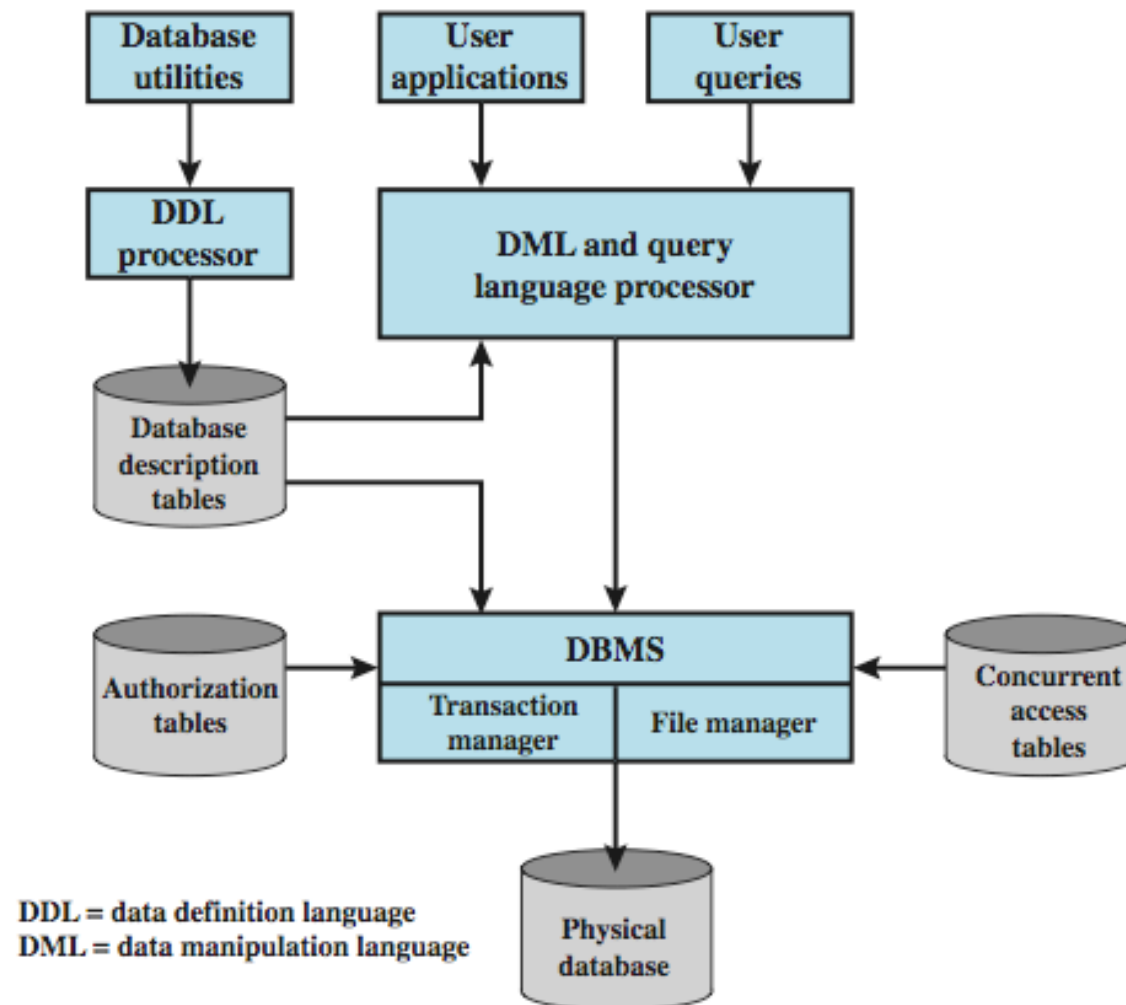


Database Security

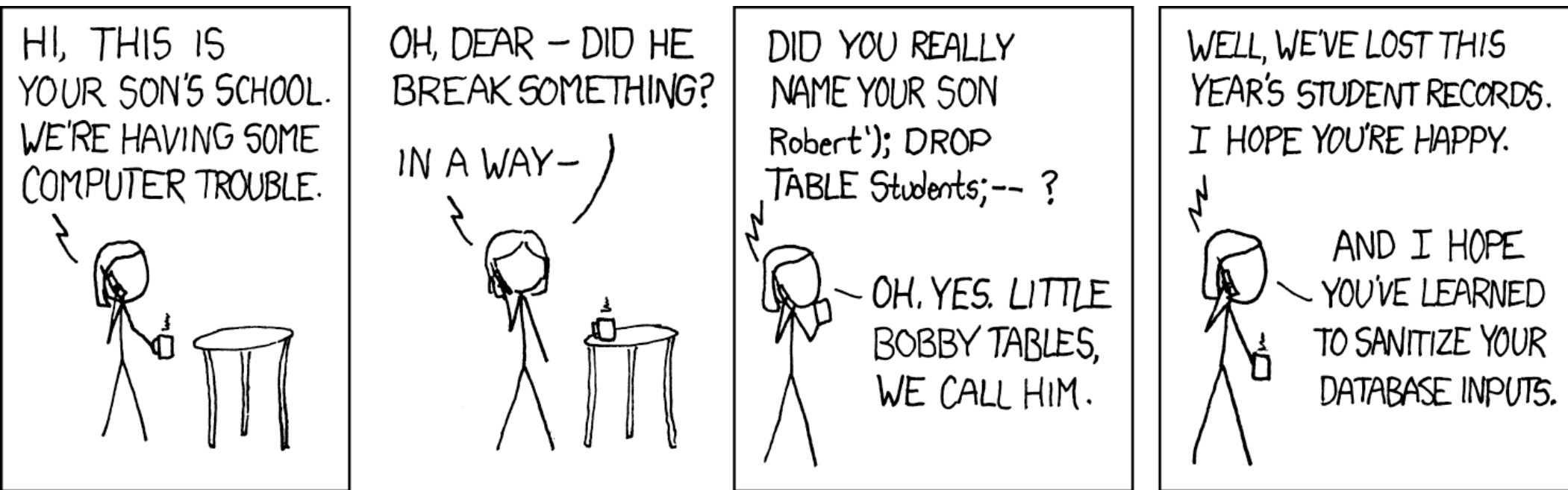
Database systems

- Structured collection of data stored for use by one or more applications
- Contains the relationships between data items and groups of data items
- Can sometimes contain sensitive data that needs to be secured
- Query language: Provides a uniform interface to the database

Database Security



xkcd 327



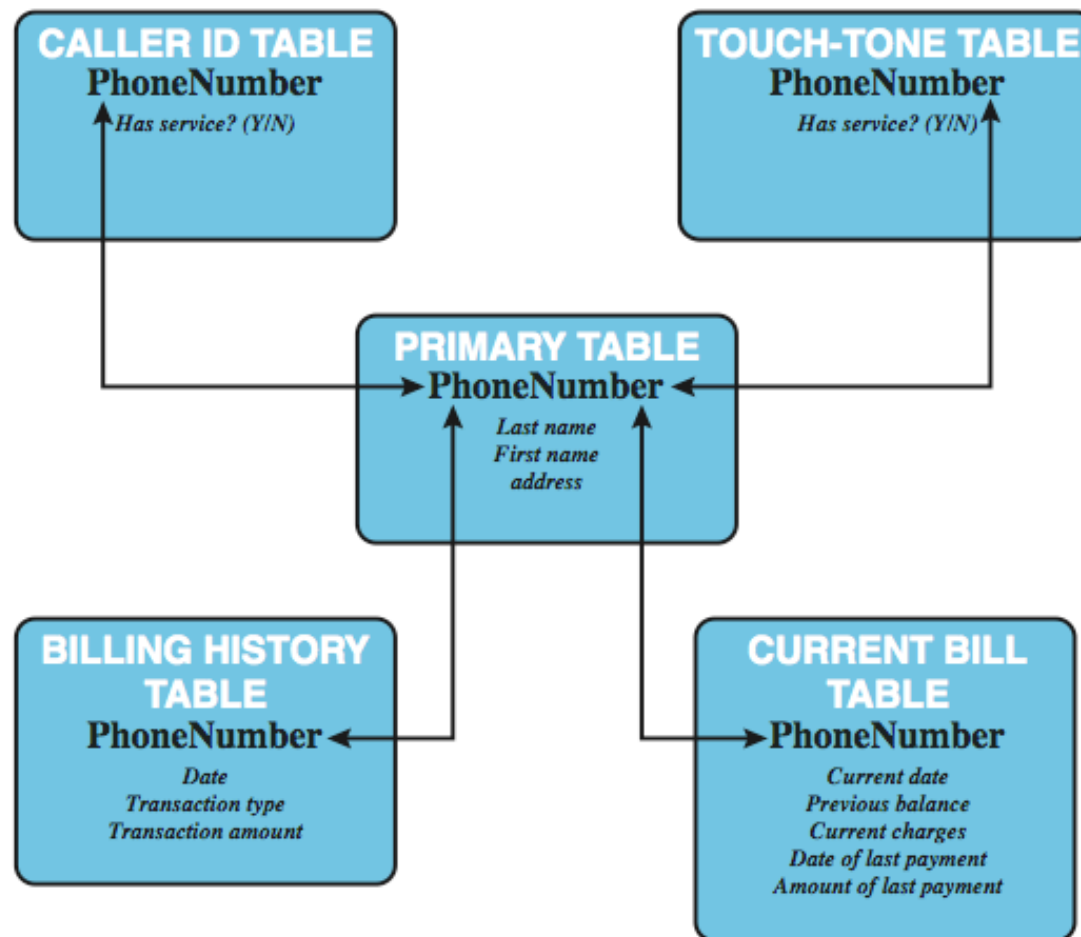
Relational Databases

- Constructed from tables of data
 - each column holds a particular type of data
 - each row contains a specific value these
 - ideally has one column where all values are unique, forming an identifier/key for that row
- Have multiple tables linked by identifiers
- SSE a query language to access data items meeting specified criteria

Relational databases

- Table of data consisting of rows and columns
 - Each column holds a particular type of data
 - Each row contains a specific value for each column
 - Ideally has one column where all values are unique, forming an identifier/key for that row
- Enables the creation of multiple tables linked together by a unique identifier that is present in all tables
- Use a relational query language to access the database
 - Allows the user to request data that fit a given set of criteria

A relational database example



Relational database terms

- Relation/table/file
- Tuple/row/record
- Attribute/column/field
- Primary key: uniquely identifies a row
- Foreign key: links one table to attributes in another
- View/virtual table: Result of a query that returns selected rows and columns from one or more tables

Abstract view of a relation

		Attributes								
		A_1	\bullet	\bullet	\bullet	A_j	\bullet	\bullet	\bullet	A_M
Records	1	x_{11}	\bullet	\bullet	\bullet	x_{1j}	\bullet	\bullet	\bullet	x_{1M}
	\bullet	\bullet			\bullet				\bullet	
	\bullet	\bullet			\bullet				\bullet	
	\bullet	\bullet			\bullet				\bullet	
	i	x_{i1}	\bullet	\bullet	\bullet	x_{ij}	\bullet	\bullet	\bullet	x_{iM}
	\bullet	\bullet			\bullet				\bullet	
	\bullet	\bullet			\bullet				\bullet	
	\bullet	\bullet			\bullet				\bullet	
	N	x_{N1}	\bullet	\bullet	\bullet	x_{Nj}	\bullet	\bullet	\bullet	x_{NM}

Relational Database Elements

Department Table			Employee Table				
Did	Dname	Dacctno	Ename	Did	SalaryCode	Eid	Ephone
4	human resources	528221	Robin	15	23	2345	6127092485
8	education	202035	Neil	13	12	5088	6127092246
9	accounts	709257	Jasmine	4	26	7712	6127099348
13	public relations	755827	Cody	15	22	9664	6127093148
15	services	223945	Holly	8	23	3054	6127092729
primary key			Robin	8	24	2976	6127091945
			Smith	9	21	4490	6127099380
			foreign key		primary key		

(a) Two tables in a relational database

Dname	Ename	Eid	Ephone
human resources	Jasmine	7712	6127099348
education	Holly	3054	6127092729
education	Robin	2976	6127091945
accounts	Smith	4490	6127099380
public relations	Neil	5088	6127092246
services	Robin	2345	6127092485
services	Cody	9664	6127093148

(b) A view derived from the database

Structured Query Language

- Structure Query Language (SQL)
 - originally developed by IBM in the mid-1970s
 - standardized language to define, manipulate, and query data in a relational database
 - several similar versions of ANSI/ISO standard

```
CREATE TABLE department (  
    Did INTEGER PRIMARY KEY,  
    Dname CHAR (30),  
    Dacctno CHAR (6) )
```

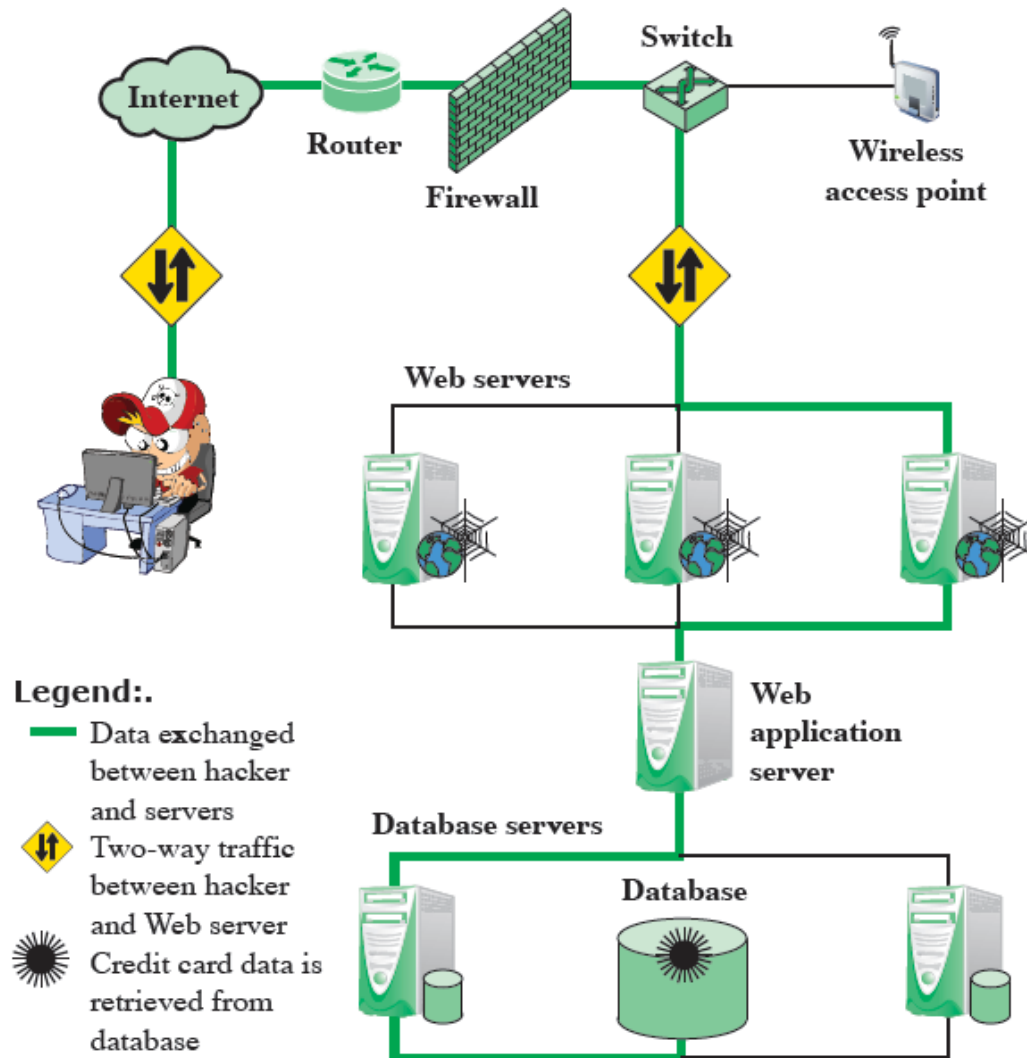
```
CREATE VIEW newtable (Dname, Ename, Eid, Ephone)  
AS SELECT D.Dname E.Ename, E.Eid, E.Ephone  
FROM Department D Employee E  
WHERE E.Did = D.Did
```

```
CREATE TABLE employee (  
    Ename CHAR (30),  
    Did INTEGER,  
    SalaryCode INTEGER,  
    Eid INTEGER PRIMARY KEY,  
    Ephone CHAR (10),  
    FOREIGN KEY (Did) REFERENCES department (Did) )
```

SQL injection attacks

- One of the most prevalent and dangerous network-based security threats
- Sends malicious SQL commands to the database server
- Depending on the environment SQL injection can also be exploited to:
 - Modify or delete data
 - Execute arbitrary operating system commands
 - Launch denial-of-service (DoS) attacks

A typical injection attack



Sample SQL injection

- The SQLi attack typically works by prematurely terminating a text string and appending a new command

SELECT fname

FROM student

where fname is 'user prompt';

User: John'; DROP table Course;--

Sample SQL injection: tautology

\$query= “

SELECT info FROM user WHERE name =

`\$_GET[“name”]’ AND pwd = `GET[“pwd”]`

”.
;

Attacker enters: ` OR 1=1 --

In-band attacks

- **Tautology:** This form of attack injects code in one or more conditional statements so that they always evaluate to true
- **End-of-line comment:** After injecting code into a particular field, legitimate code that follows are nullified through usage of end of line comments
- **Piggybacked queries:** The attacker adds additional queries beyond the intended query, piggy-backing the attack on top of a legitimate request

Database Access Control

- DBMS provide access control for database assume have authenticated user
- DBMS provides specific access rights to portions of the database
 - e.g. create, insert, delete, update, read, write
 - to entire database, tables, selected rows or columns
 - possibly dependent on contents of a table entry
- can support a range of policies:
 - centralized administration
 - ownership-based administration
 - decentralized administration

Inferential attack (gathering info)

- There is no actual transfer of data, but the attacker is able to reconstruct the information by sending particular requests and observing the resulting behavior of the Website/database server
 - Illegal/logically incorrect queries: lets an attacker gather important information about the type and structure of the backend database of a Web application

ASKING THE EMAIL SERVER IF AN ADDRESS IS CORRECT

ASKING THE EMAIL SERVER IF AN ADDRESS IS CORRECT

```
tbakker — tbakker@cse3320:~ — ssh tbakker@cse3320.org — 64x17
(base) [tbakker@cse3320 ~]$
```



Harvesting email addresses,
conference information, etc.

Out-band attack

- This can be used when there are limitations on information retrieval, but outbound connectivity from the database server is lax

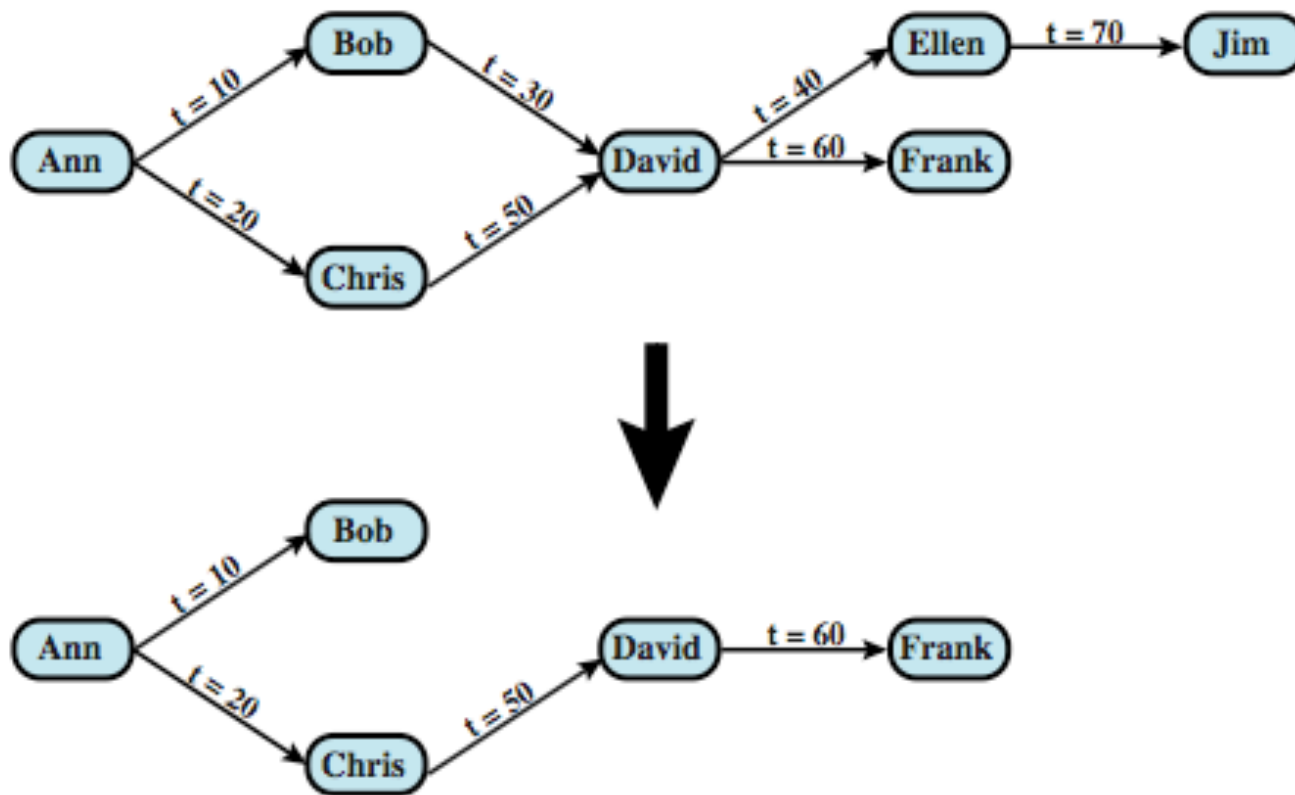
SQLi countermeasures

- Defensive coding: stronger data validation
- Detection
 - Signature based
 - Anomaly based
 - Code analysis
- Runtime prevention: Check queries at runtime to see if they conform to a model of expected queries

SQL Access Controls

- If the user has access to the entire database or just portions of it
- Two commands:
 - `GRANT {privileges | role} [ON table] TO {user | role | PUBLIC} [IDENTIFIED BY password] [WITH GRANT OPTION]`
 - e.g. `GRANT SELECT ON ANY TABLE TO john`
 - `REVOKE {privileges | role} [ON table] FROM {user | role | PUBLIC}`
 - e.g. `REVOKE SELECT ON ANY TABLE FROM john`
 - **WITH GRANT OPTION:** whether grantee can grant “GRANT” option to other users
- Typical access rights are:
 - `SELECT, INSERT, UPDATE, DELETE, REFERENCES`

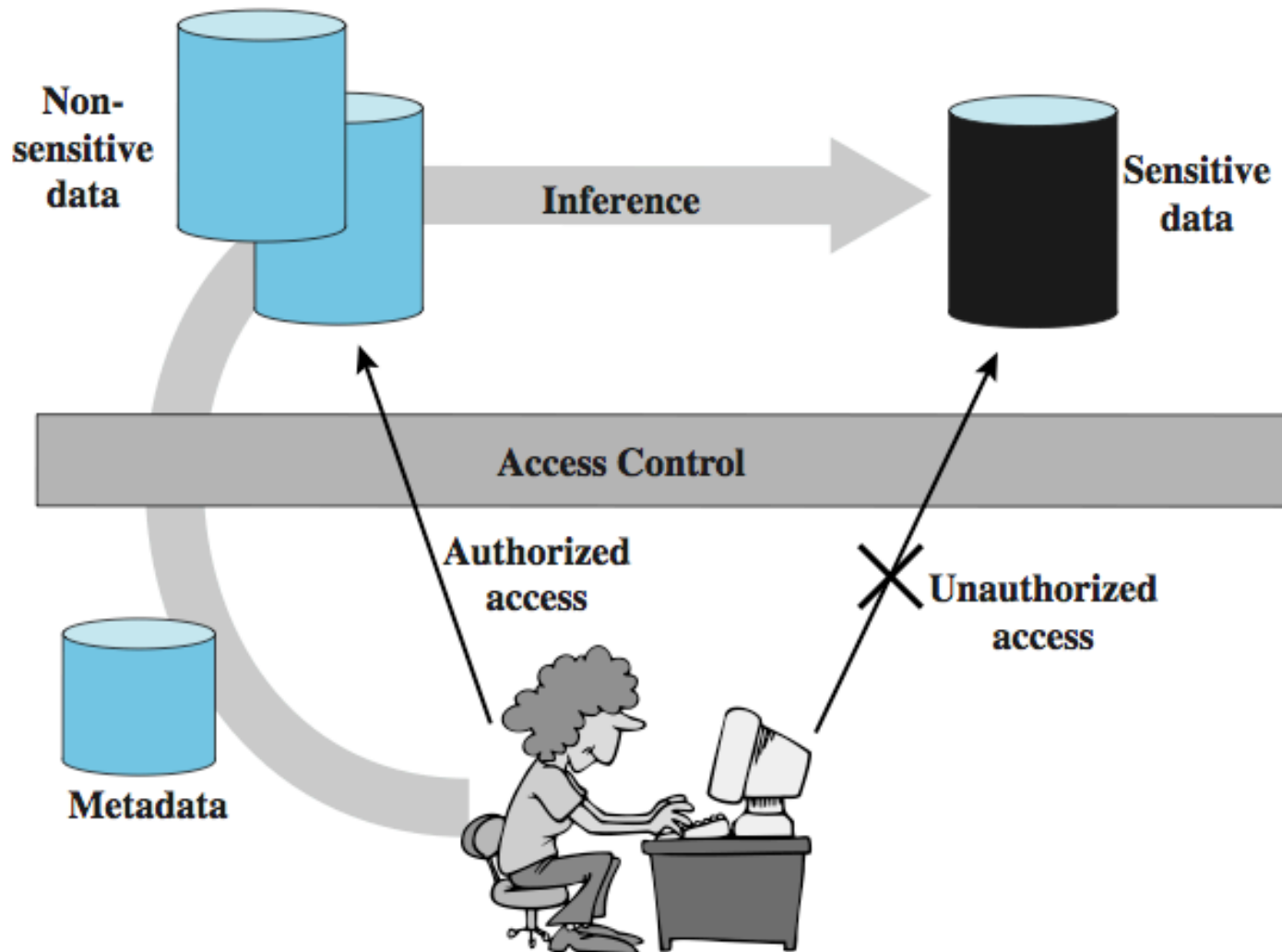
Cascading Authorizations



Role-Based Access Control

- Role-based access control work well for DBMS
 - eases admin burden, improves security
- Categories of database users:
 - application owner
 - end user
 - administrator
- DB RBAC must manage roles and their users

Inference



Inference Example

Name	Position	Salary (\$)	Department	Dept. Manager
Andy	senior	43,000	strip	Cathy
Calvin	junior	35,000	strip	Cathy
Cathy	senior	48,000	strip	Cathy
Dennis	junior	38,000	panel	Herman
Herman	senior	55,000	panel	Herman
Ziggy	senior	67,000	panel	Herman

(a) Employee table

Position	Salary (\$)	Name	Department
senior	43,000	Andy	strip
junior	35,000	Calvin	strip
senior	48,000	Cathy	strip

(b) Two views

Name	Position	Salary (\$)	Department
Andy	senior	43,000	strip
Calvin	junior	35,000	strip
Cathy	senior	48,000	strip

(c) Table derived from combining query answers

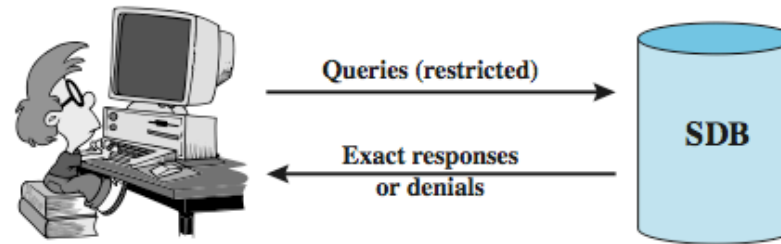
Inference Countermeasures

- Inference detection at database design
 - alter database structure or access controls
- Inference detection at query time
 - by monitoring and altering or rejecting queries

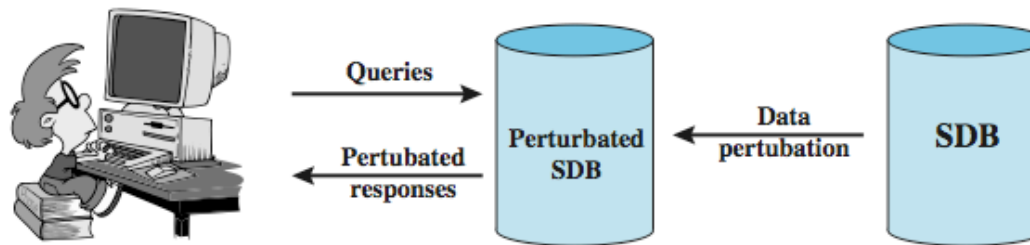
Statistical Databases

- Provides data of a statistical nature
 - e.g. counts, averages
- Two types:
 - pure statistical database
 - ordinary database with statistical access
 - some users have normal access, others statistical
- Access control objective to allow statistical use without revealing individual entries
- Security problem is one of inference

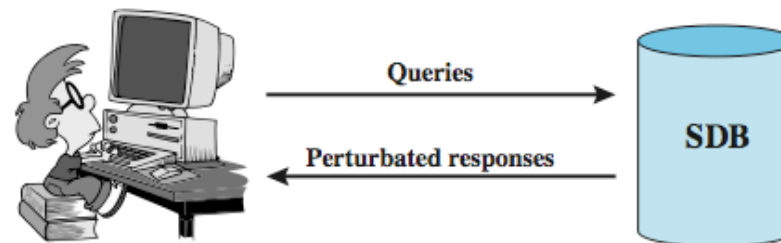
Protecting Against Inference



(a) Query set restriction



(b) Data perturbation



(c) Output perturbation

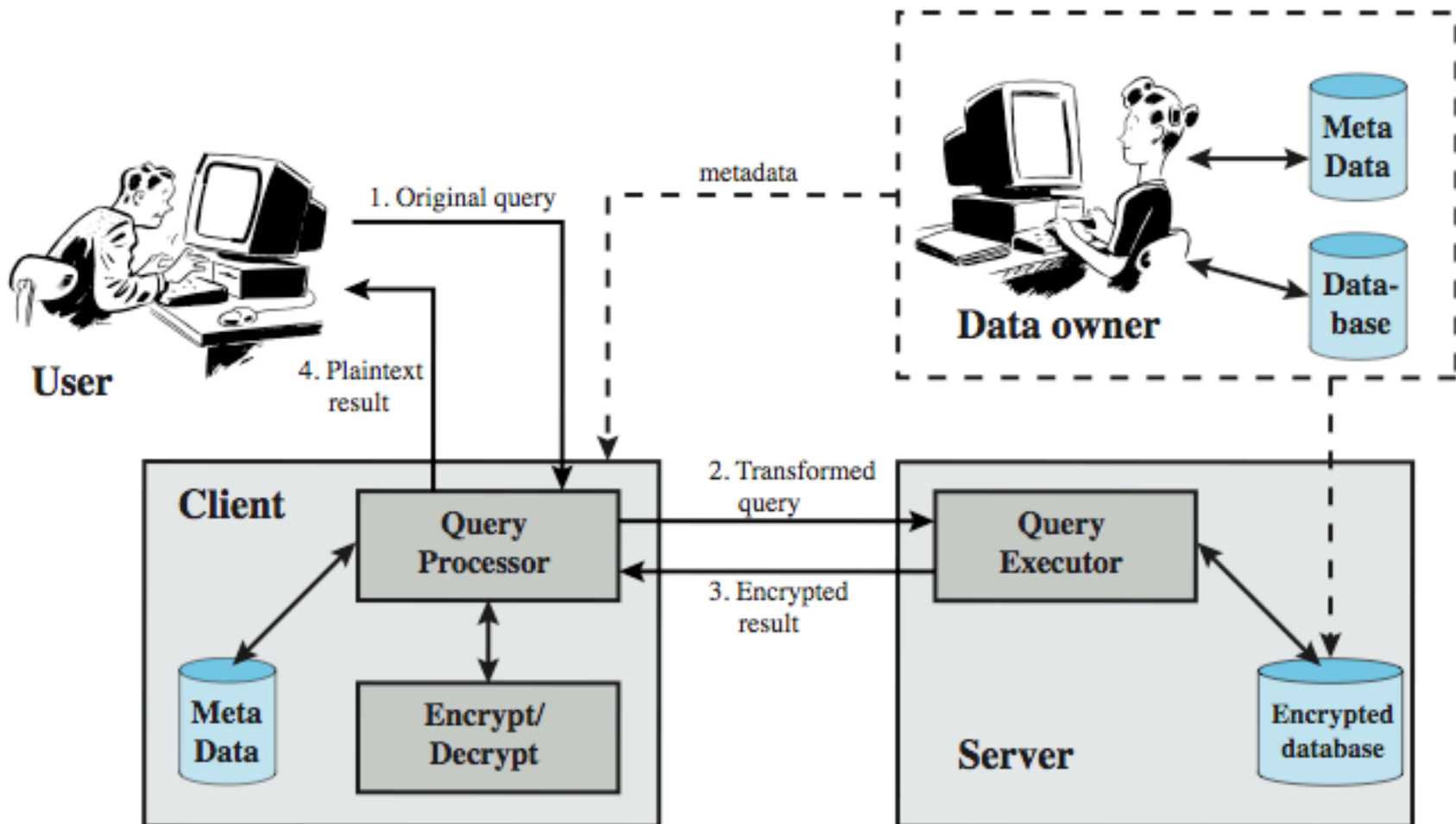
Perturbation

- Add noise to statistics generated from data
 - will result in differences in statistics
- Data perturbation techniques
 - data swapping
 - generate statistics from probability distribution
- Output perturbation techniques
 - random-sample query
 - statistic adjustment
- Must minimize loss of accuracy in results

Database Encryption

- Databases typical a valuable info resource
 - protected by multiple layers of security: firewalls, authentication, O/S access control systems, DB access control systems, and database encryption
- Can encrypt
 - entire database - very inflexible and inefficient
 - individual fields - simple but inflexible
 - records (rows) or columns (attributes) - best
 - also need attribute indexes to help data retrieval
- Varying trade-offs

Database Encryption

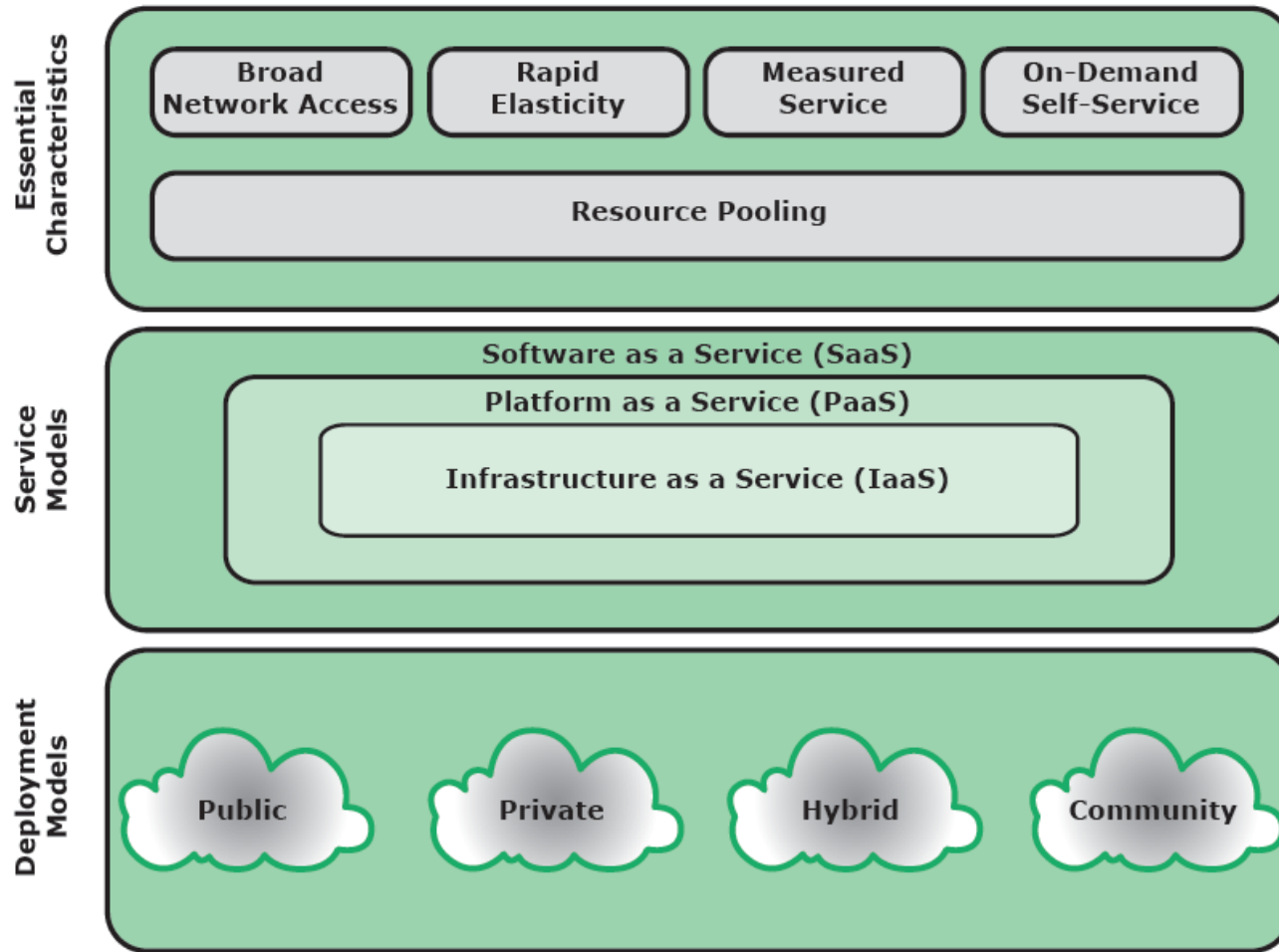


Cloud computing

- An increasing trend in many organizations to move a substantial portion (or all) of their IT to cloud computing
- NIST SP-800-145 defines cloud computing as: “A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. ...”



Cloud computing elements



Essential characteristics: Broad network access

- Capabilities available over a network
- Accessed thru standard mechanisms
- Use by heterogeneous client platforms (desktops, laptops, tablets, cell phones)

Essential characteristics:

Rapid elasticity

- The ability to expand/reduce services to specific requirements
- Large numbers of servers during the holidays
- Smaller number of resources during off-peak periods
- Release a resource when a task is completed

Essential characteristics: Measured service

- Cloud system automatically
 - Control and optimize resource use by leveraging a metering capability appropriate to the type of service
 - Storage, processing, bandwidth, active users

Essential characteristics: On-demand service

- A consumer can unilaterally provision computing capabilities without requiring human interaction
 - Server time, network storage
 - Resources won't have to be a permanent part of the client's IT infrastructure

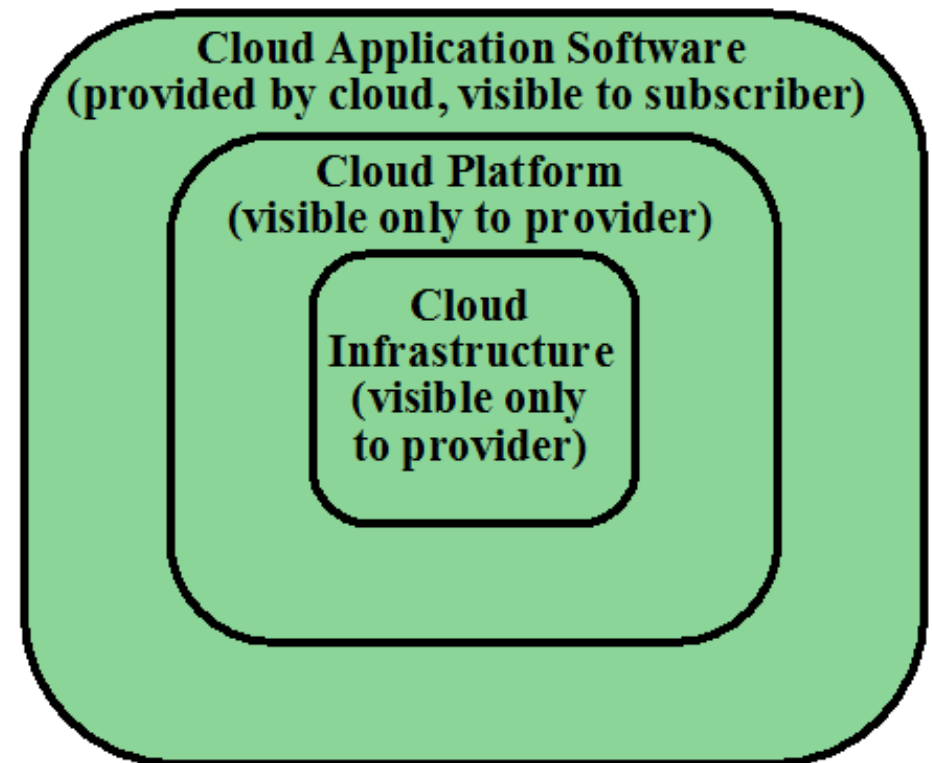
Essential characteristics:

Resource pooling

- As a consequence of the previous:
 - The providers resources are pooled to serve multiple consumers using a multi-tenant model
 - Multiple resources assigned and re-assigned to clients
 - Storage, processing power, bandwidth ...
 - Consumers need not to know the physical location of the service

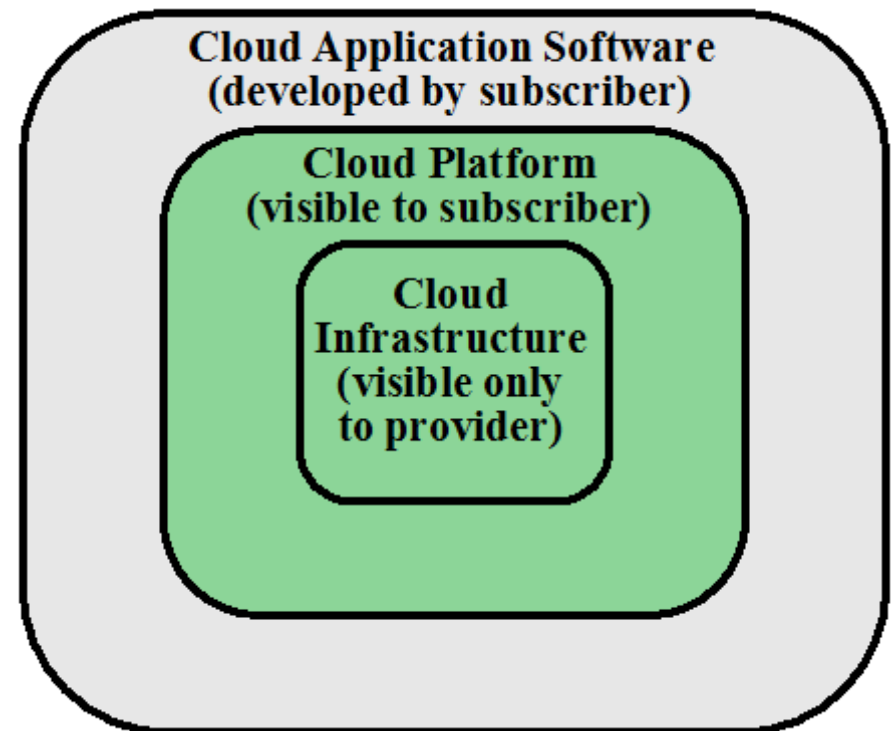
Service model: SaaS

- provides service to customers in the form of application software
- Clients need not to install
- Clients can access application via various platforms thru a simple interface (often a browser)



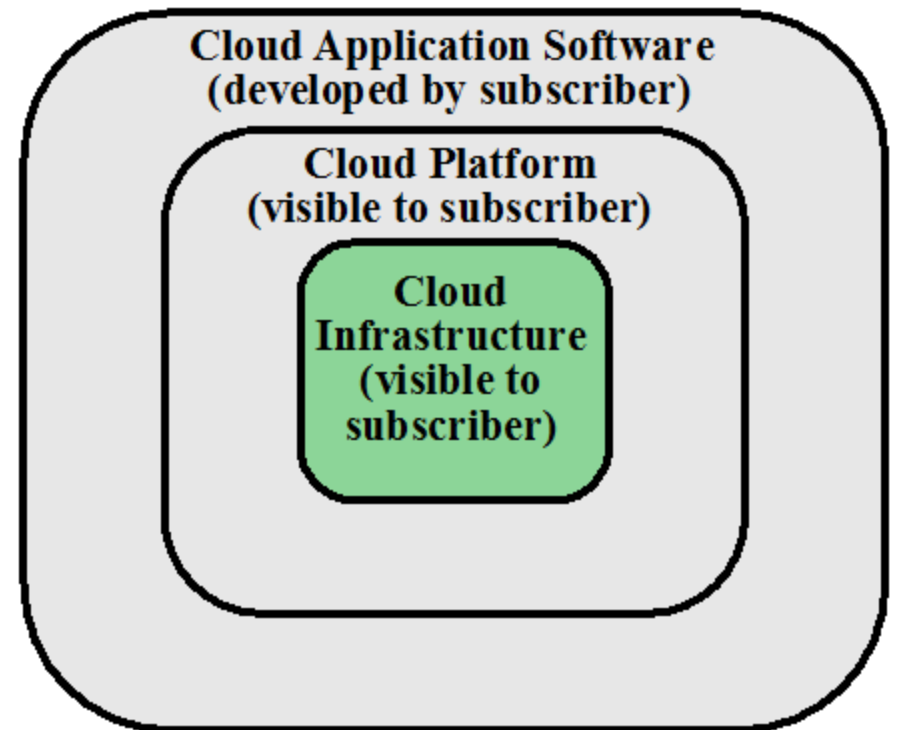
Service model: PaaS

- Provides service to customers in the form of a platform on the which the customer apps can run
- Clients deploy on the cloud
- PaaS provides useful building block/tools

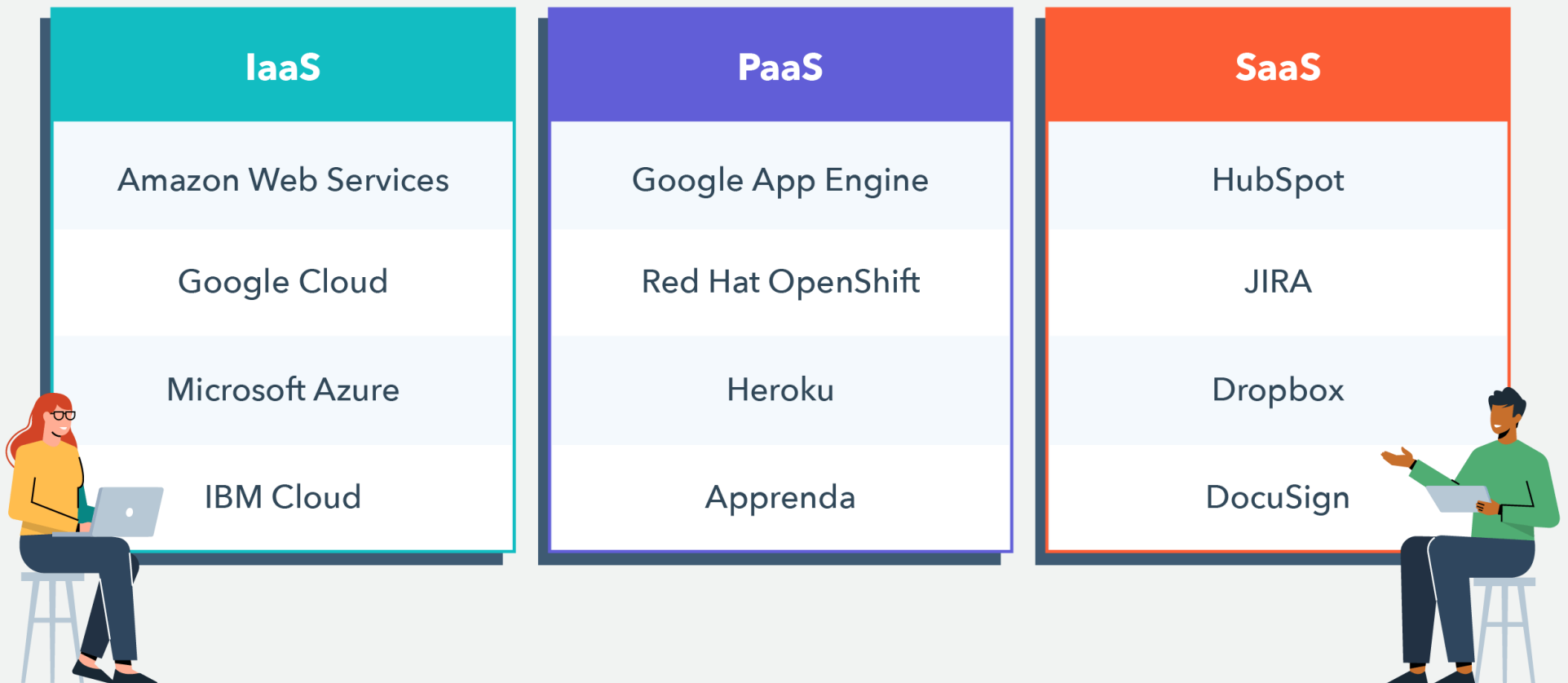


Service model: IaaS

- Provides clients access to the underlying cloud infrastructure
- VM and other abstracted hardware, OS, APIs
- Amazon Elastic Computing (EC2) and Windows Azure



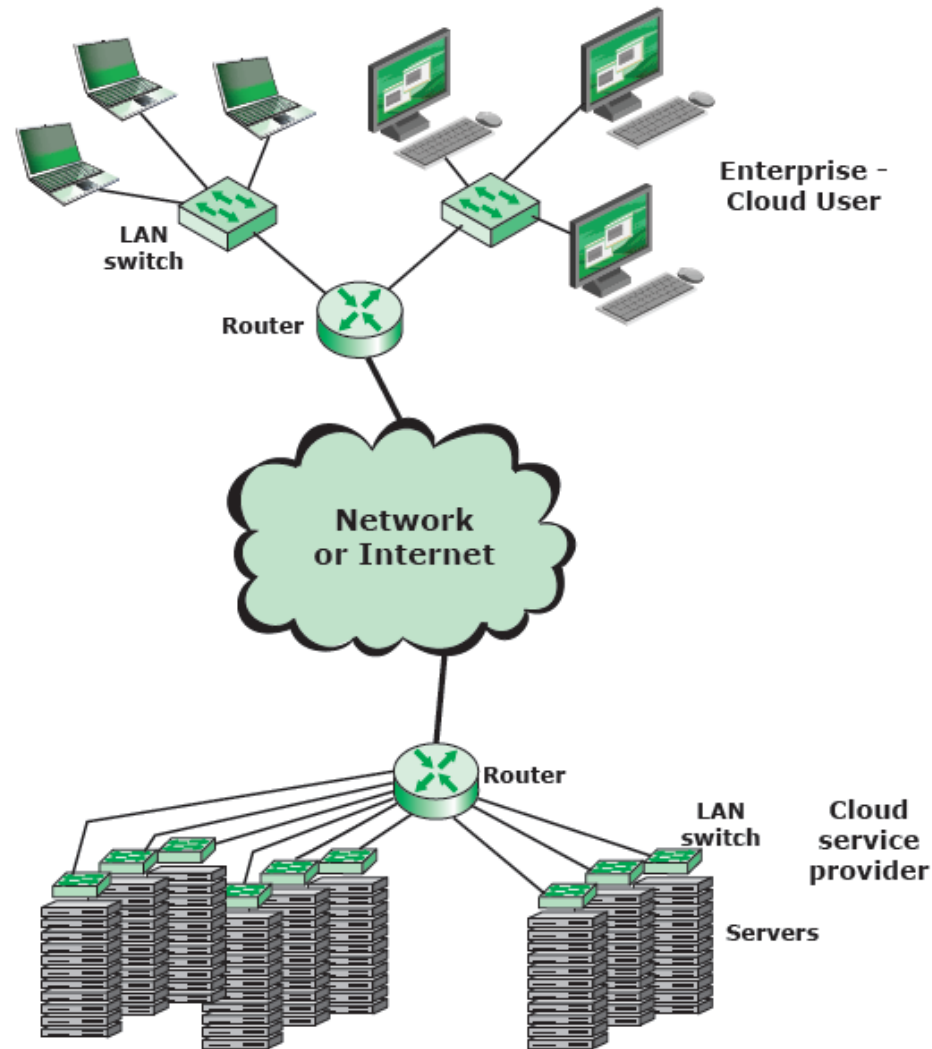
IaaS vs. PaaS vs. SaaS Examples



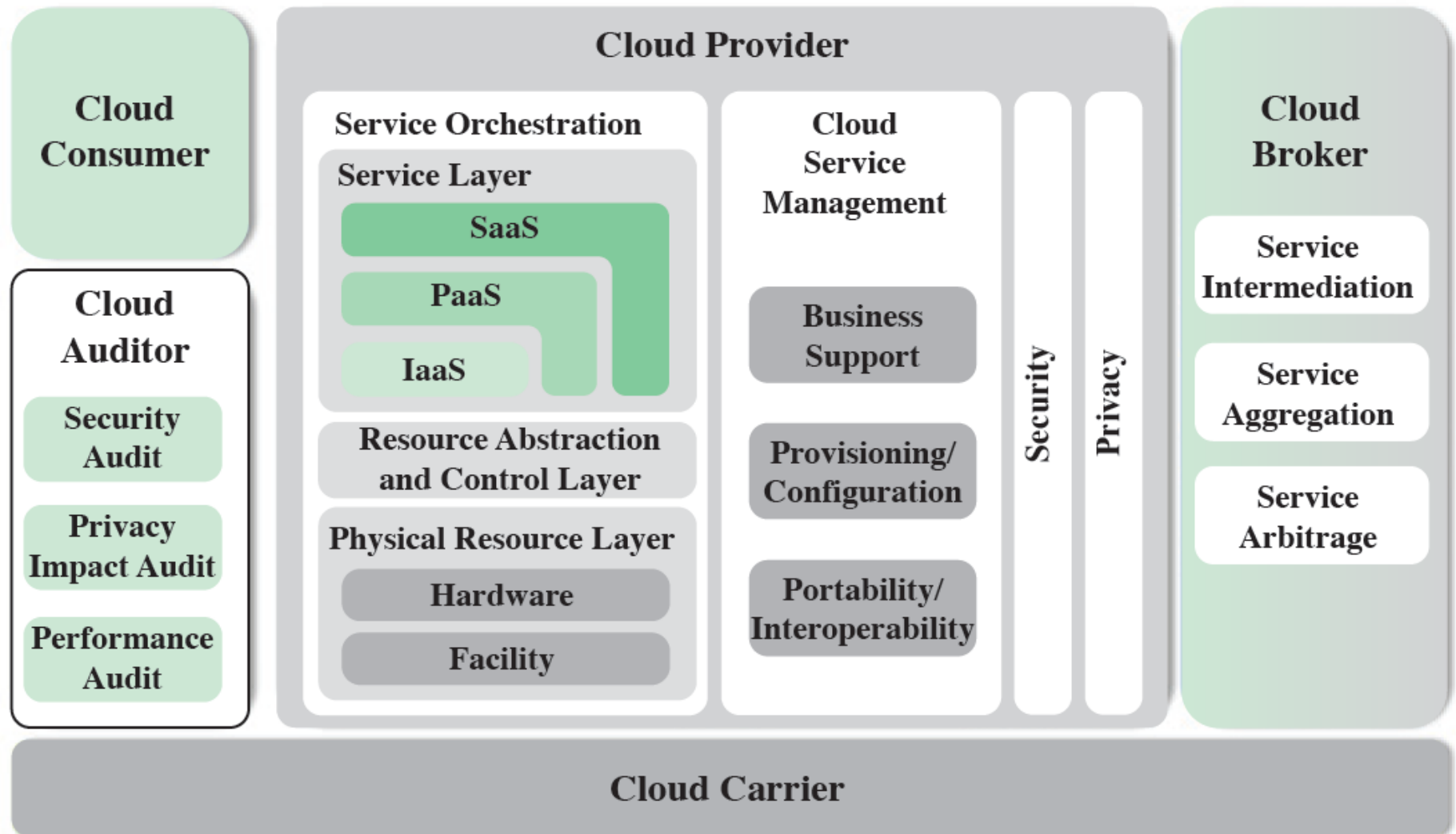
Deployment models

- **Public:** the cloud resources/services are available to the general public
- **Private:** The cloud infrastructure is solely for an organization
- **Community:** the cloud infrastructure is for a specific community and is shared by several organizations
- **Hybrid:** a composition two or more of the above

A typical cloud service



NIST's reference architecture (conceptual model)



NIST's reference architecture (conceptual model)

- **Cloud customer:** A person or organization that uses or is interested in cloud providers services
- **Cloud provider (CP):** a person or organization that makes cloud services available
- **Cloud auditor:** A party that conducts independent assessment of sources (e.g., security)
- **Cloud broker:** A party that manages use, performance,, negotiations
- **Cloud carrier:** An intermediary that provides connectivity and service transport

Cloud provider (CP)

- For SaaS: CP deploys, configures, maintains, updates app software
- For PaaS: CP manages the computing infrastructure for the platform (middleware, database, OS, programming languages, tools)
- For IaaS: CP acquires physical computing resources: servers, network, storage, ...

Cloud security risks

- **Abuse and nefarious use of cloud computing**
 - Relatively easy to register for the services
 - Many cloud services offer free trials
 - Enables hackers to get inside to conduct attacks (spamming, DoS, ...)
 - The burden on CP to provide protection
- Countermeasures
 1. Stricter/restrict registration
 2. Enhanced credit card monitoring
 3. Comprehensive monitoring (traffic)

Cloud security risks

- **Insecure interfaces of APIs**
 - CPs expose a set of APIs for customers apps
 - Security depends on APIs: must be secure (from authentication to encryption)
- Countermeasures
 1. Analyzing the security of APIs
 2. Ensuring strong authentication and access control
 3. Understanding the dependency chain between the APIs

Cloud security risks

- **Malicious insiders**
 - Client organization relinquishes many (or all) of its IT and gives to the CP
 - A grave concern: malicious insider activity
- Countermeasures (by client)
 1. Comprehensive supplier assessment
 2. Specify human resource requirements as part of legal contract
 3. Require transparency

Cloud security risks

- **Shared technology issues**
 - IaaS services are delivered in a scalable way by a shared infrastructure (CPU caches, GPUs, ...)
 - Probably not designed for multi-tenant architecture
 - Vulnerable to attacks (insider and outsiders)
- Countermeasures
 1. Implement best security practices during implementation, deployment, configuration
 2. Monitor environment for unauthorized activities
 3. Promote strong authentication and access control

Cloud security risks

- **Data loss or leakage**
 - Cloud storage may be the only backup storage
 - Could be devastating for many clients if data is lost
- Countermeasures
 1. Implement strong API access control
 2. Encrypt and protect integrity when in transit
 3. Analyze data protection at design and run time

Cloud security risks

- **Account or service hijacking**
 - Usually with stolen credentials
 - Compromise confidentiality, integrity and availability
- Countermeasures
 1. Prohibit sharing of credentials between users
 2. Leverage strong multi-factor authentication
 3. Employ proactive monitoring

Data protection in the cloud

- Data must be secured while at transit, in use, or at rest
 - Client can employ encryption for transit
 - Client can encrypt data for storage (rest) but can also be CP's responsibility (key management)

Cloud security as a service

- Identify management
- Data loss prevention
- Web security
- E-mail security
- Encryption
- Business continuity
- Network security
- ...

