mobile business

Lecture 05

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Information & Communication Security (SS 2020)

Access Control

Prof. Dr. Kai Rannenberg

Chair of Mobile Business & Multilateral Security
Goethe-University Frankfurt a. M.

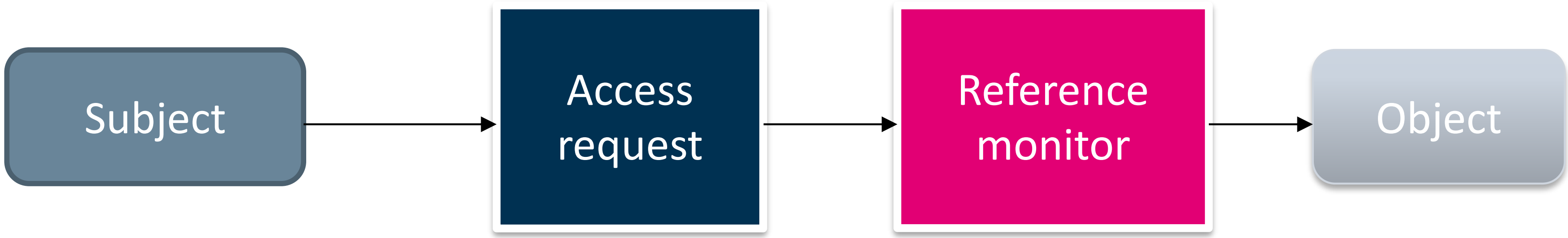
mobile business

Agenda

- Introduction
- Access attributes
- Access control matrix
- Security models
  - Bell-LaPadula
  - Chinese wall
  - Role based access control

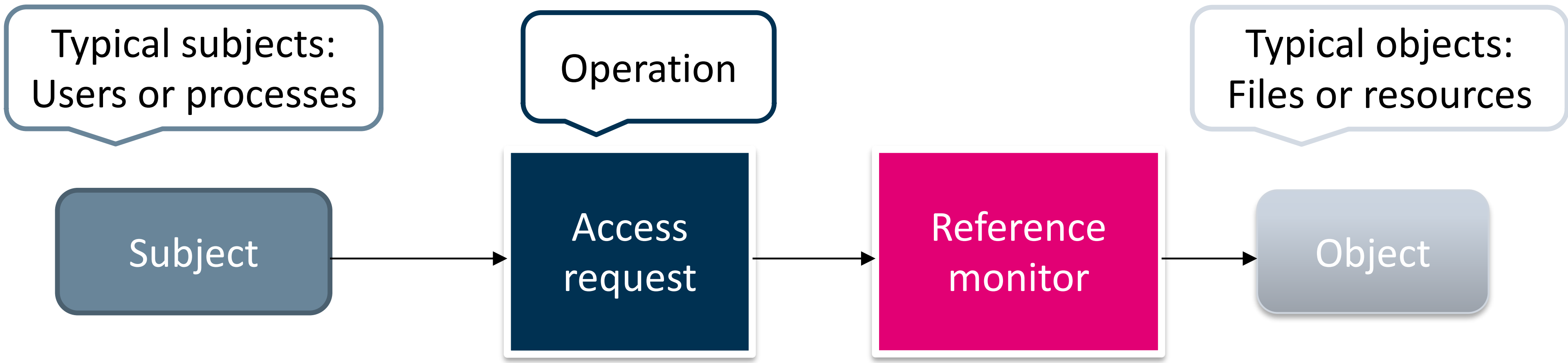
Subjects and Objects

- There is an active ***subject*** accessing a passive ***object*** with some specific ***access operation***, while a ***reference monitor*** grants or denies access.



[Go99]

Access Control Model



Depending on circumstances, an entity can be a subject in one access request, and an object in another.

Based on: B. Lampson. Protection. ACM Operating System Review, 8, 1974

Control Options

- The terms *subject* and *object* merely distinguish between the active and passive party in the access request.
- Subjects and objects present two options for **focusing control**:
  - You can either specify what a subject is allowed to do;
  - Or what may be done with an object.

[Go99]

mobile business

Agenda

- Introduction
- Access attributes
- Access control matrix
- Security models
  - Bell-LaPadula
  - Chinese wall
  - Role based access control

Access Modes (in the original Bell-LaPadula model)

- On the most elementary level, a subject may observe or alter an object.

- *Observe:* to look at the contents of an object
- *Alter:* to change the contents of an object

- Usually a richer set of access operations is used.

[BLP76, page 10; Go99]

Access Rights and Access Attributes (in the original Bell-LaPadula model)

	Execute	Append	Read	Write
Observe			x	x
Alter		x		x

[BLP76, page 10-11; Go99]

Access Rights and Access Attributes

- Usually files can be opened for read or write access.
- This way the operating system can avoid potential conflicts like two users trying to write the same file.
- Write access usually includes read access.

[Go99]

Access Rights and Access Attributes (in the original Bell-LaPadula model)

- Few systems implement the append function.
- An **example** of use of the append function
  - **Log files :** a process writing to a log file has no need to, and probably should not be able to read its contents.
- Operating systems can use files without opening them at all. Therefore, the execute right does not include the observer or alter mode:
  - **Example**: Cryptographic engine holding a **master key** in a special tamper-resistant register, so that the key can be invoked without being read.

[BLP76, page 10; Go99]

Example: Unix

- The Unix operating system expresses access control policies in terms of three operations:

- ***Read:*** reading from a file
- ***Write:*** writing to a file
- ***Execute:*** executing a (program) file

Note:
- Write access does **not** include read access.
- “Execute” does not match the Bell-LaPadula definition of “Execute”.

[Go99]

Example: Unix

- When applied to a directory, the access operations have the following meaning:

- ***Read:*** list directory contents
- ***Write:*** create or rename a file in the directory
- ***Execute:*** search the directory

- Unix controls who can create and delete files by controlling write access to the file's directory.

[Go99]

Example: Windows NT

- Windows NT (and the following versions) use the New Technology File System (NTFS) as the basis for access control.
- NTFS uses the following permissions:
  - Read
  - Write
  - Execute
  - Delete
  - Change permission
  - Change ownership
- NTFS does not rely on operations on directories to handle deletion of files or change of access rights.

[Go99]

Ownership

- The owner of a resource decides who is allowed to have access.
- Most operating systems support the concept of ownership of a resource.
- They consider ownership when making access control decisions.
- They may include operations to change the ownership of a resource.

[Go99]

mobile business

Agenda

- Introduction
- Access attributes
- Access control matrix
- Security models
  - Bell-LaPadula
  - Chinese wall
  - Role based access control

Access Rights

- Access rights can be defined individually for each combination of subject and object.

> - a set *S* of **subjects**
> - a set *O* of **objects**
> - a set *A* of **access operations**

[Go99]

Access Control Matrix

- Access rights are defined quite simply in the form of an access control matrix (***M***).

$$M=(M_{so})_{s \in S,\ o \in O} \qquad \text{with } M_{so} \subseteq A$$

- The entry M_{so} specifies the set of access operations subject ***s*** may perform on object ***o***.

[Go99]

Access Control Matrix - Example

Object / Subject	bob.doc	edit.exe	fun.com
Alice	—	{execute}	{execute, read}
Bob	{read, write}	{execute}	{execute, read, write}

A = {execute, read, write}

Based on [Go99]

Capabilities

- Access rights can be kept with the subjects or with the objects.
- In the first case every subject is given a ***capability***.
- A *capability* is an unforgeable token that specifies this subject's access rights.

[Go99]

Capabilities Example

- Alice's capability: edit.exe: execute;
  fun.com: execute, read

- Bob's capability: bob.doc: read, write;
  edit.exe: execute;
  fun.com: execute, read, write

[Go99]

Capabilities

- Complexity of security management by capabilities is very high.
- Operating systems are traditionally oriented towards managing objects.
- It is difficult to get an overview of who has permission to access a given object.
- It is very difficult to revoke a capability.

[Go99]

Access Control Lists (ACL)

- An access control list stores the access rights to an object with the object itself.
- Access rights of previous example:
  - bob.doc Bob: read, write
  - edit.exe Alice: execute; Bob: execute
  - fun.com Alice: execute, read; Bob: execute, read, write

[Go99]

Agenda

- Introduction
- Access attributes
- Access control matrix
- Security models
  - Bell-LaPadula
  - Chinese wall
  - Role based access control

Security Models

- Security models are an important concept in the design and analysis of secure systems.
- They capture the security policy that should be enforced by the system.
- Security models capture policies for confidentiality and for integrity.
- Some models apply to environments where policies are static (Bell-LaPadula).
- Others consider dynamic changes of access rights (Chinese Wall).

[Go99]

Agenda

- Introduction
- Access attributes
- Access control matrix
- Security models
  - Bell-LaPadula
  - Chinese wall
  - Role based access control

Bell-LaPadula (1)

- **Bell-LaPadula (BLP)** is probably the most popular of the security models.
- It was developed by Bell and LaPadula at the time of the first concerted efforts to design secure multi-user operating systems.
- It captures the **confidentiality** aspects of access control.
- **Access permissions** are **defined** both through an **access control matrix** and through **security levels.**

[Go99]

Bell-LaPadula (2)

- Security policies prevent information **flowing downwards** from a high security level to a low security level.
- These policies are commonly referred to as *multi level security*.
- BLP only considers the information flow that occurs when a subject observes or alters an object.

[Go99] [Bi05]

BLP Model

- We have:
  - A set of subjects S;
  - A set of objects O;
  - The set of access operations A={execute, read, append, write};
    - execute: no observe, no alter
    - read: observe, no alter
    - append: no observe, alter
    - write: observe, alter
  - A set L of security levels with a partial ordering ≤.

[Go99]

Security Levels Example

	A subject has a security clearance	*An object has a security classification*
TOP SECRET	Tamara, Thomas	Personnel Files
SECRET	Sally, Samuel	Electronic Mail Files
CONFIDENTIAL	Claire, Clarence	Activity Log Files
UNCLASSIFIED	Ursula, Ulf	Telephone List Files

[Bi05]

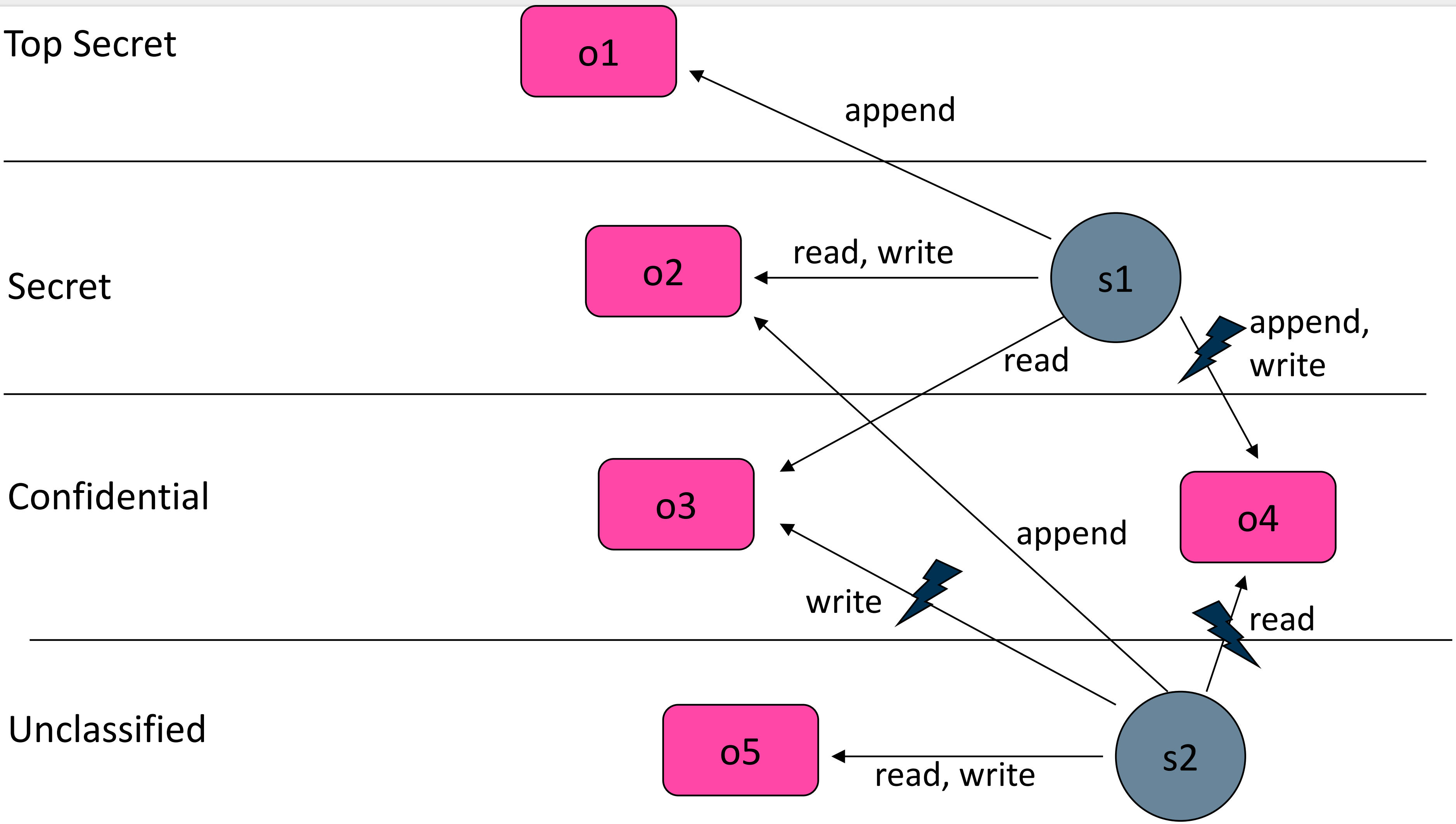
Properties

- **Simple Security Property (SS Property):**
  - **“No read up”** rule
  - ***s*** can **read** (observe) ***o*** if and only if
    - $l_o \leq l_s$ and
    - ***s*** has discretionary read access to ***o***.

- *** Property (Star Property):**
  - **“No write down”** rule
  - ***s*** can **append or write** (observe and alter) ***o*** if and only if
    - $l_s \leq l_o$ and
    - ***s*** has discretionary write access to ***o***.

[BLP76] [Bi05]

Example



Adapted from [Ec04]

Basic Security Theorem

- If all state transitions in the system are secure and if the initial state of the system is secure, then every subsequent state will also be secure, no matter which inputs occur.

[Go99]

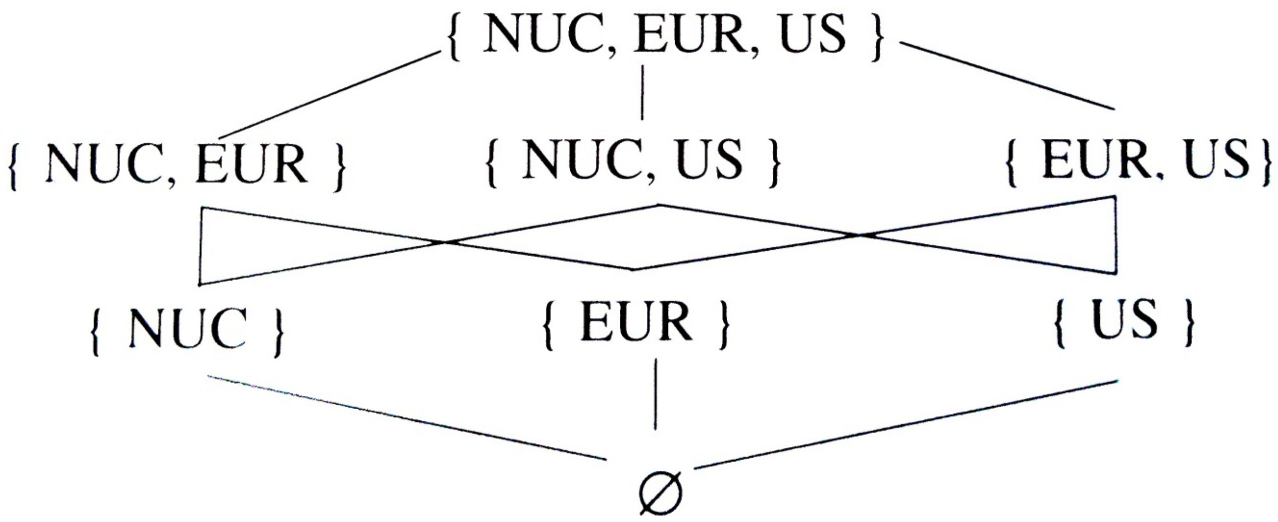
BLP with Categories

- Expanding the model by adding “Categories” to each security classification makes it richer and more powerful.
- The categories arise from the “need to know” principle:
  - No subject should be able to read objects unless reading them is necessary for that subject to perform its functions.

[Bi05]

Categories Example

- If the categories are NUC, EUR and US, the sets of categories form a lattice under the operation ⊆ (subset).



[Bi05]

Properties

- The security level (L_A, C_A) dominates (**dom**) the security level (L_B, C_B) if and only if
  - $L_B \leq L_A$ and
  - $C_B \subseteq C_A$.

- **Simple Security Property (SS Property):**
  s can read **o** if and only if
  - **s dom o** and
  - **s** has discretionary read access to **o**.

- *** Property (Star Property):**
  s can write to **o** if and only if
  - **o dom s** and
  - **s** has discretionary write access to **o**.

[Bi05]

mobile business

Example

- George is cleared into security level (SECRET, {NUC, EUR}).
- DocA is classified as (CONFIDENTIAL, {NUC}).
- DocB is classified as (SECRET, {EUR, US}).
- DocC is classified as (SECRET, {EUR}).

- George can read DocA and DocC but not DocB.

[Bi05]

Basic Security Theorem (revisited)

- If all state transitions in the system are secure and if the initial state of the system is secure, then every subsequent state will also be secure, no matter which inputs occur.

[Go99]

Agenda

- Introduction
- Access attributes
- Access control matrix
- Security models
  - Bell-LaPadula
  - Chinese wall
  - Role based access control

Chinese Wall Model

- The Chinese Wall (CW) model is a model of a security policy that refers equally to confidentiality and integrity.
- It describes policies that involve a conflict of interest in business.
- The environment of a stock exchange or investment house is the most natural environment for this model.
- In this context, ***the goal of the model is to prevent a conflict of interest in which a trader represents two clients.***

[Bi05]

Definitions

- The *objects* of the database are items of information related to a company.
- A *company dataset (CD)* contains objects related to a single company.
- A *conflict of interest (COI)* class contains the datasets of companies in competition.

[Bi05]

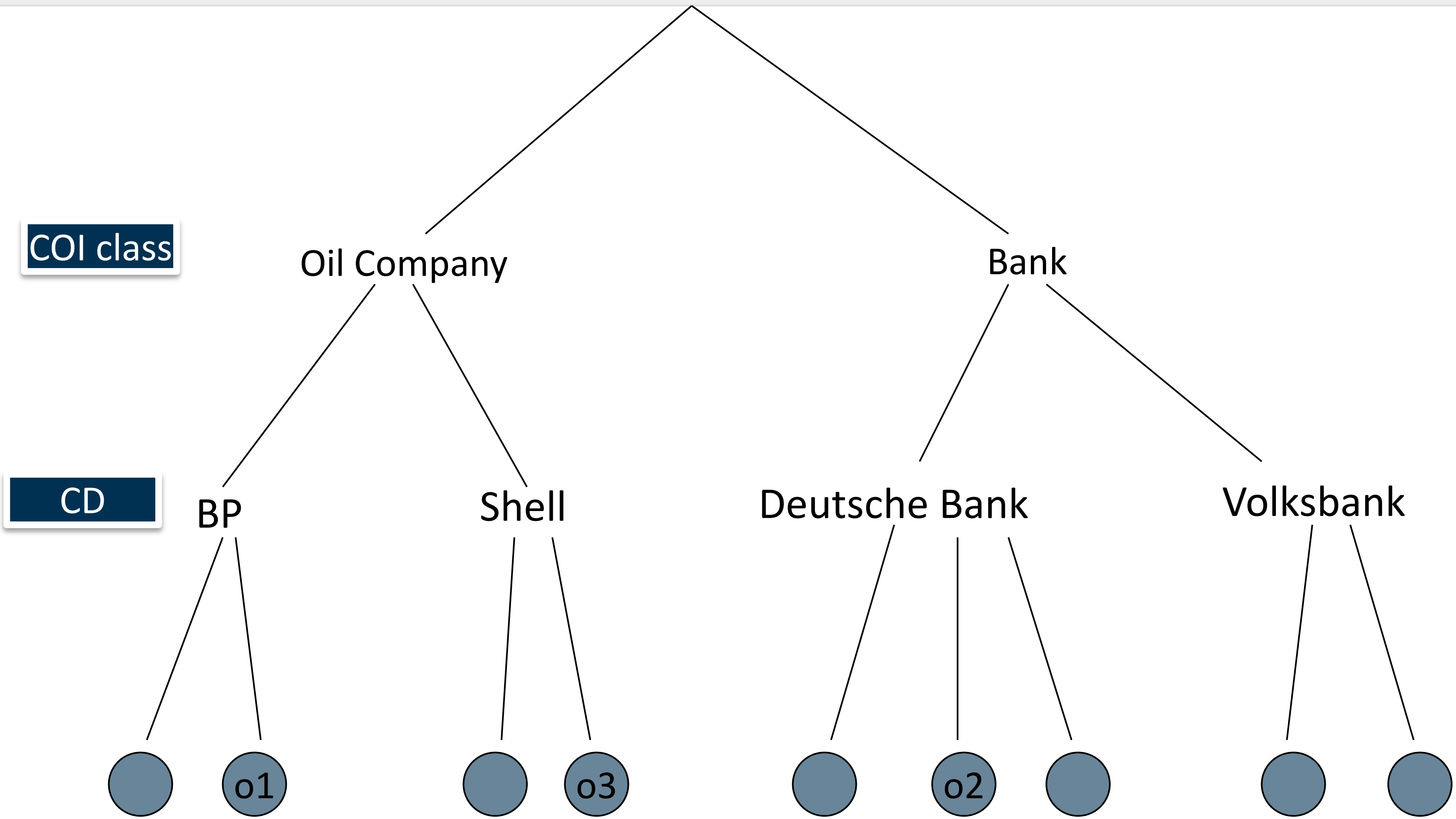
CW - Simple Security Condition

- s can read o if and only if any of the following holds:
  1. There is an object o´ such that s has accessed o´and CD(o´) = CD(o).
  2. For all objects o´, o´є PR(s)
     => COI(o´) ≠ COI(o)
  3. o is a sanitized object.

PR(s) are the files already opened by s.

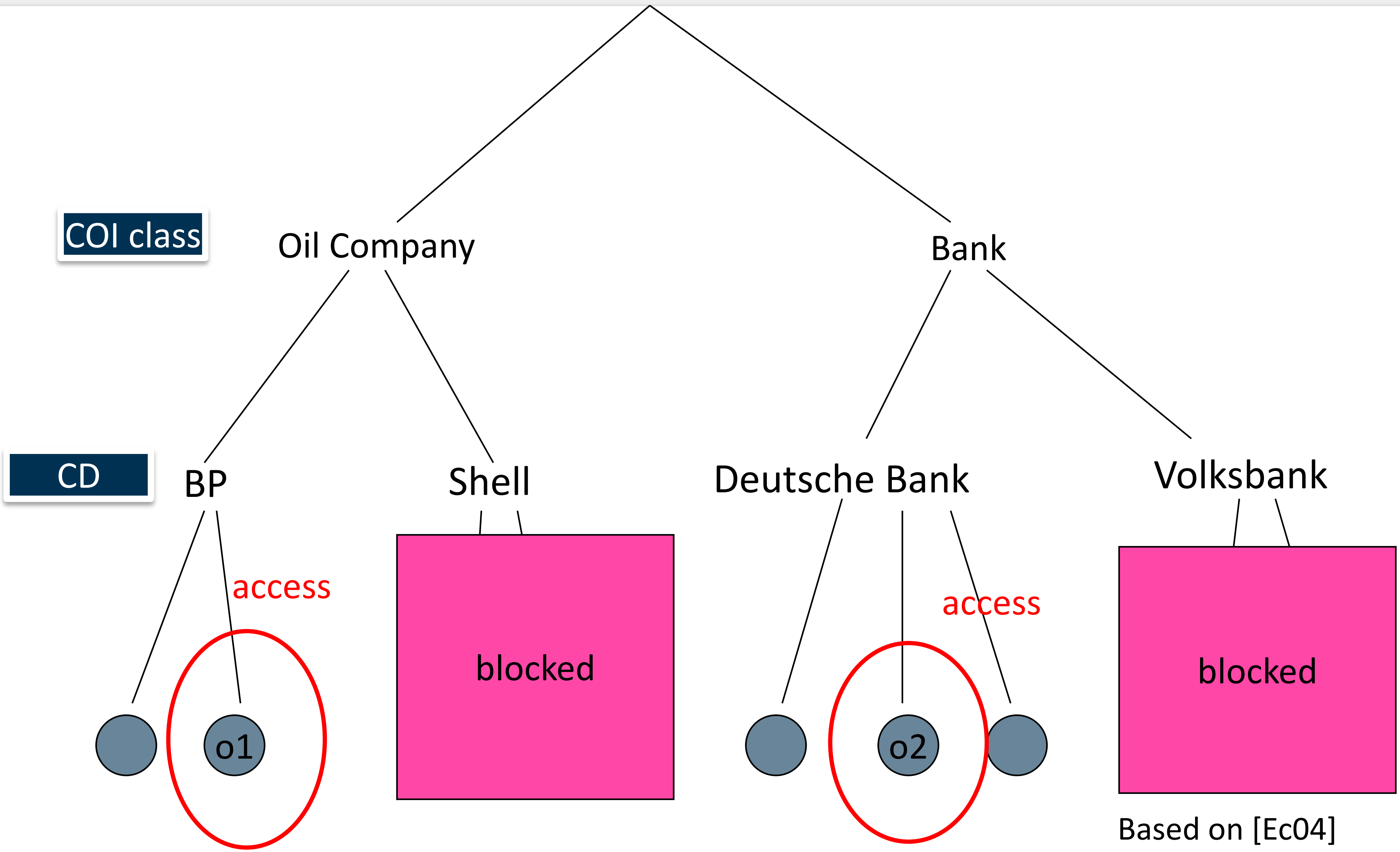
[Bi05]

Example



Based on [Ec04]

Example



Based on [Ec04]

Agenda

- Introduction
- Access attributes
- Access control matrix
- Security models
  - Bell – LaPadula
  - Chinese wall
  - Role based access control

Role Based Access Control

- The ability or need, to access information may depend on one’s job functions.
- This suggests associating access with the particular job of the user.

[Bi05]

Definitions

- A role is a collection of job functions. Each role ***r*** is authorized to perform one or more transactions. The set of authorized transactions for ***r*** is written ***trans(r).***
- The active role of a subject ***s***, written ***actr(s)***, is the role that ***s*** is currently performing.
- The authorized roles of a subject ***s***, written ***authr(s)***, is the set of roles that ***s*** is authorized to assume.
- The predicate ***canexec(s,t)*** is true if and only if the subject ***s*** can execute the transactions ***t*** at the current time.

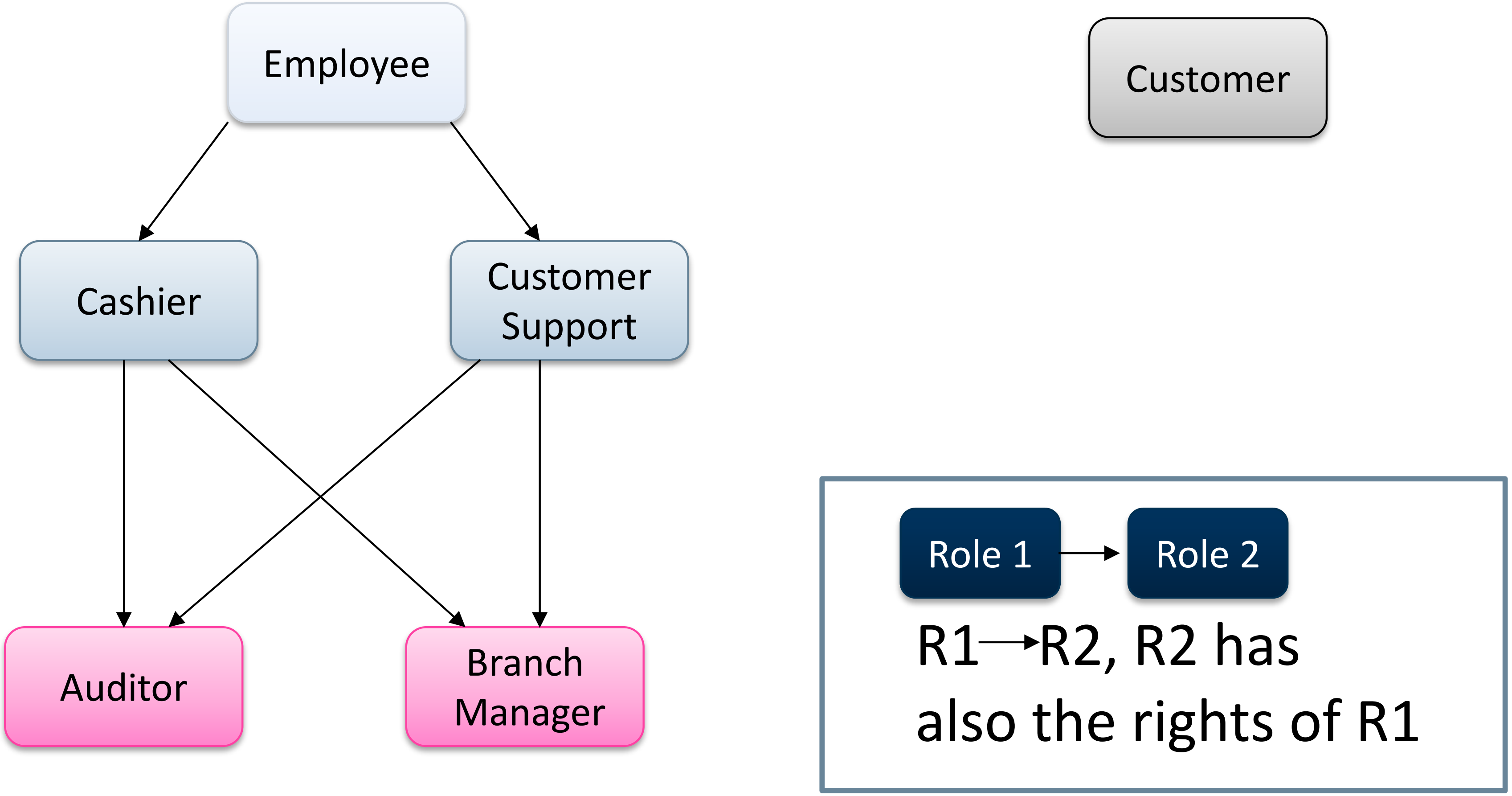
[Bi05]

Rules

- If a subject can execute at least one transaction, then the subject has an active role.
  - This binds the notion of execution of a transaction to the role rather than to the user.
- The subject must be authorized to assume its active role.
  - It cannot assume an unauthorized role.
- A subject cannot execute a transaction for which its current role is not authorized.

[Bi05]

Example: Bank



References

- **[BLP76] David Elliott Bell, Len La Padula.** Secure Computer System: Unified Exposition and Multics Interpretation, ESD-TR-75-306, 1976; http://csrc.nist.gov/publications/history/bell76.pdf
- **[Bi05] Matt Bishop.** *Introduction to Computer Security.* Boston: Addison Wesley, 2005. pp. 27-35, 62-69, 83-87, 92-94
- **[Ec04] Claudia Eckert.** *IT-Sicherheit*. München, Wien: Oldenbourg, 2004
- **[Go99] Dieter Gollmann.** *Computer Security.* Chichester, New York, Weinheim, Brisbane, Singapore, Toronto: John Wiley & Sons, 1999. pp. 30-54