

Lecture 3

Access Control

Information &
Communications Security
(WS 2008)

Prof. Dr. Kai Rannenberg

Guest Lecturer Dr. Martin Reichenbach

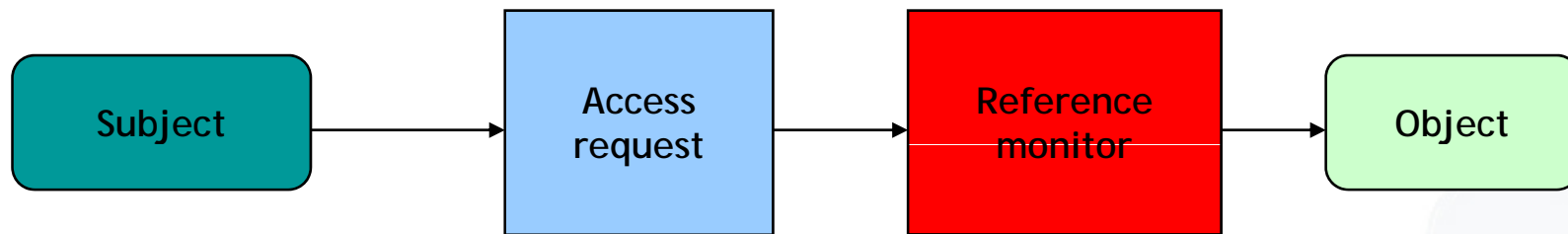
T-Mobile Chair for
Mobile Business & Multilateral Security
Johann Wolfgang Goethe University Frankfurt a. M.
www.whatismobile.de



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

Access Control in real life

- There is an active *subject* accessing a passive *object* with some specific *access operation*, while a *reference monitor* grants or denies access.
- Typical subjects are users and processes.
- Typical objects are files or resources.
- Depending on circumstances, an entity can be a subject in one access request and an object in another.



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.

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

- 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 will be used.

	Execute	Append	Read	Write
Observe			X	X
Alter		X		X

- 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.

- Few systems implement the append function.
- However, sometimes the append function could be useful.
 - Example: 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 observe or alter mode
 - Example: File encryption

- 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.

- 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.

- 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]

- The owner of a resource decrees 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.

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

- Access rights can be defined individually for each combination of subject and object.
- Therefore, we have:
 - a set S of subjects,
 - a set O of objects,
 - a set A of access operations.

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

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

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

Access Control Matrix - Example

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

- 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.

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

- 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.

- 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.exe

Alice: execute, read;
Bob: execute, read,
write

[Go99]

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

- 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]

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

- Bell-LaPadula (BLP) is probably the most famous 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]

- 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.

- We have:
 - A set of subjects S ;
 - A set of objects O ;
 - The set of access operations
 $A = \{\text{execute, read, append, write}\}$;
 - A set L of security levels with a partial ordering \leq .

TOP SECRET



SECRET



CONFIDENTIAL



UNCLASSIFIED

Tamara, Thomas



Sally, Samuel



Clairra, Clarence



Ursula, Ulf

Personal Files



Electronic Mail Files



Activity Log Files



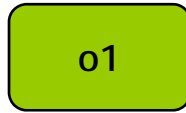
Telephone List Files



[Bi05]

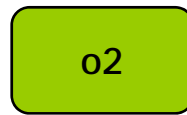
- Simple Security (SS) Property: s can read o if and only if $l_o \leq l_s$ and s has discretionary read access to o .
- * Property (Star Property): s can write o if and only if $l_s \leq l_o$ and s has discretionary write access to o .

Top Secret

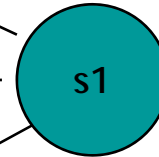


append

Secret



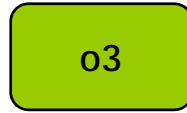
read-write



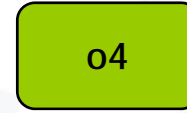
read
execute

append,
read-write

Confidential



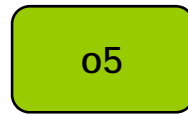
append



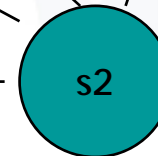
read

read-write

Unclassified



read-write



- 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.

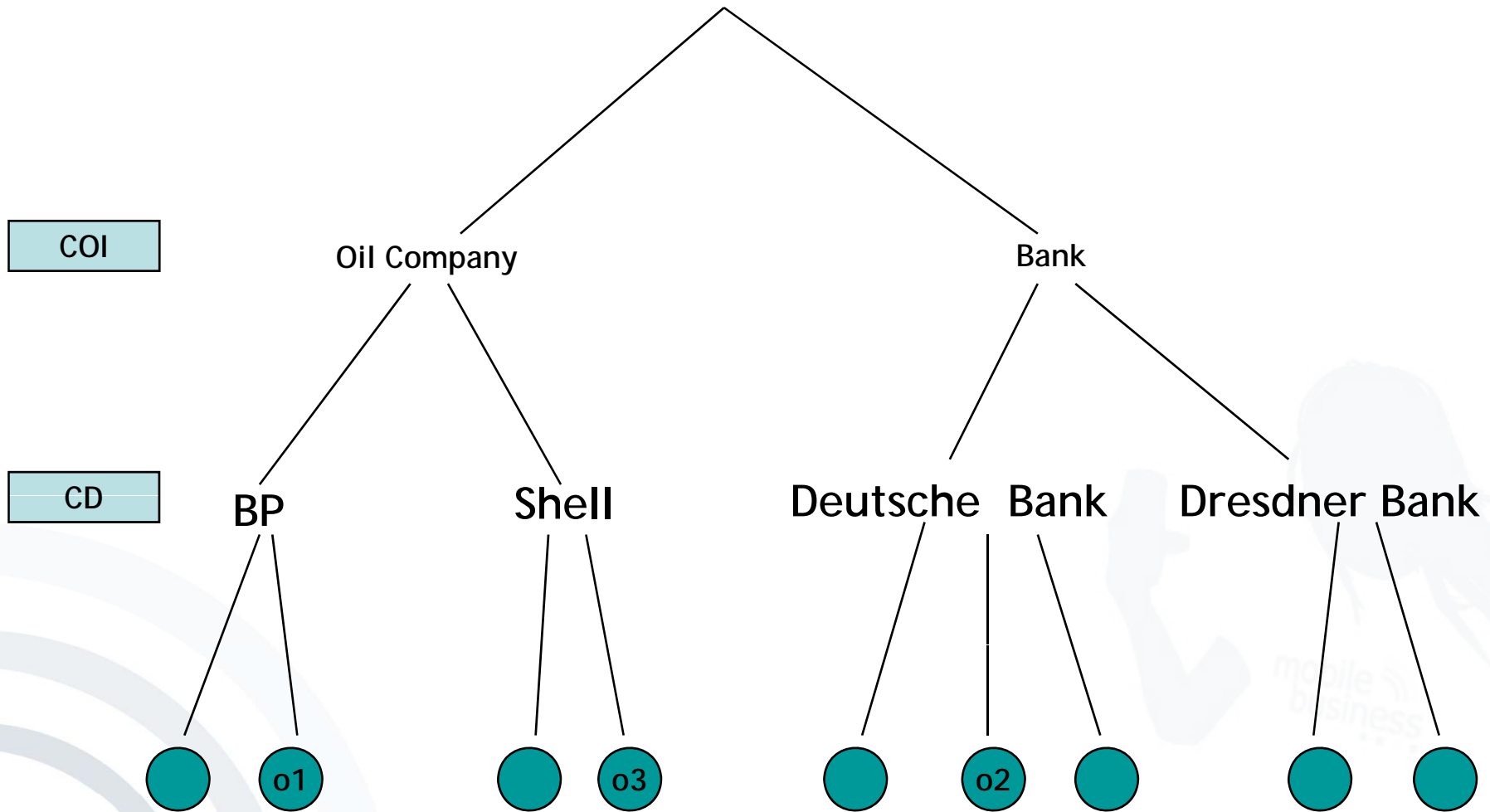
- Introduction
- Access attributes
- Access control matrix
- Security models
 - Bell-LaPadula
 - Chinese wall
 - Role based access control

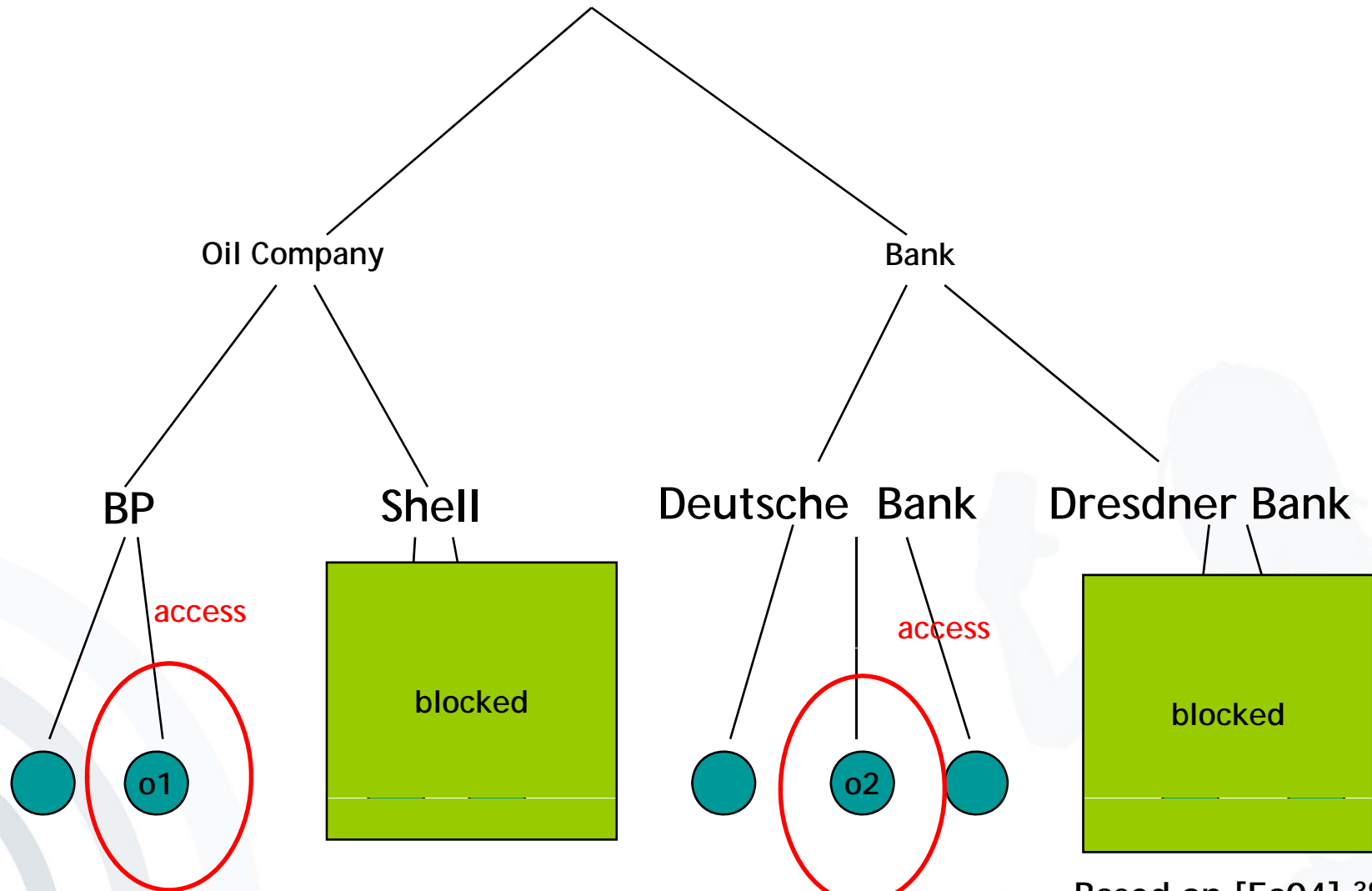
- The Chinese Wall (CW) model is a model of a security policy that refers equally to confidentiality and integrity.
- It describes policies that involve a conflict of interest in business.
- The environment of a stock exchange or investment house is the most natural environment for this model.
- In this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

- 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.

- 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' , $o' \in PR(s)$
 $\Rightarrow COI(o') \neq COI(o)$
 3. o is a sanitized object.

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



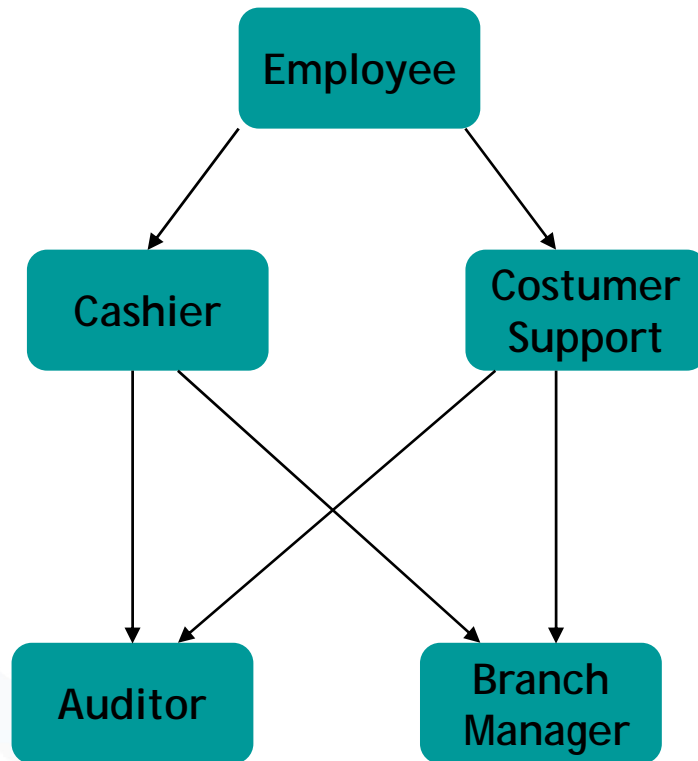


- Introduction
- Access attributes
- Access control matrix
- Security models
 - Bell - LaPadula
 - Chinese wall
 - Role based access control

- 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.

- 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.

- If a subject can execute any 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.



Customer

role
 $R1 \rightarrow R2$, R2 has also the rights of R1

- Bishop, Matt. *Introduction to Computer Security*. Boston: Addison Wesley, 2005. pp. 27-35, 62-69, 83-87, 92-94
- Eckert, Claudia. *IT-Sicherheit*. München, Wien: Oldenbourg, 2004.
- Gollmann, Dieter. *Computer Security*. Chichester, New York, Weinheim, Brisbane, Singapore, Toronto: John Wiley & Sons, 1999. pp. 30-54