

---

# Access Control Models

## Part II

Elisa Bertino  
*CERIAS and CS &ECE Departments*  
*Purdue University*

# Introduction

---

## Other models:

- The Chinese Wall Model – it combines elements of DAC and MAC
- RBAC Model – it is a DAC model; however, it is sometimes considered a policy-neutral model
- The Biba Model – relevant for integrity

---

# The Chinese Wall Access Control Model

# Table of Contents

---

- Conflict of Interests
- Chinese Wall Policy
  - Information classification
  - Read Rule
  - Write Rule
- Criticisms to this model (R. Sandu)

OCTOBER 14, 2002

▶ [DAILY BRIEFING](#)
[Today's News](#)  
[News Archive](#)
[Markets](#)  
[Stocks](#)  
[Funds](#)  
[Sectors](#)  
[Economy & Bonds](#)  
[Investing Glossary](#)
[Newsletter Sign-Up](#)
[LIFESTYLE](#)
[COLUMNS](#)
[FORUMS & CHATS](#)
[NEWSLETTERS](#)
[PERSONAL FINANCE](#)
[SEARCH & BROWSE](#)
[SPECIAL REPORTS](#)
[TOOLS & SCOREBOARDS](#)
[VIDEO VIEWS](#)
**STREET WISE**

By Amey Stone

## A Chinese Wall -- or Several Fences?

**Strictly dividing Wall Street research from investment banking may be too painful. However, smaller but still useful steps can be taken**

It sounds like a no-brainer: The best way to end conflicts of interest between research and investment banking would be for firms to separate the two divisions, right? If research operates independently, the theory goes, analysts could express their true views on stocks without fear of cramping the style of investment bankers, who typically earn the bulk of most firms' income.

The problem is that this obvious solution creates a lot of problems. Wall Street research operations are so costly that firms couldn't keep as many analysts on staff or cover as many companies if their work wasn't subsidized to some extent by investment banking. Most analysts would end up either paid a lot less or be out of a job. Even then, the new



[Amey Stone](#) is an  
associate editor of  
BusinessWeek Online

▶ [Printer-Friendly Version](#)

▶ [E-Mail This Story](#)

### MARKET INFO

<a href="#">DJIA</a>	8141.92	+282.20
<a href="#">Nasdaq</a>	1392.27	+51.94
<a href="#">S&amp;P 500</a>	862.79	+29.52

[Create / Check Portfolio](#)  
[Launch Popup Ticker](#)

Stock Lookup

GO

Enter name or ticker



### RELATED ITEMS



**This Week's Issue** | **Past Issues** | **Subscribe Now** | **Free Newsletters** | **Advertise** | **Log In**

Search **usnews.com**

 **Go**

**Advanced Search**

**Rankings & Guides**

▼ **Money & Business**

Auto

Biz Buzz

Economy/Market

Investing

Retirement

Taxes

Tools

Education

Health

Opinion

Technology

Washington Whispers

Work & Career

News Briefings

News Quiz

Photography

**Business & Technology 6/25/01**

## When the fix was in

*How Wall Street's storied Chinese wall failed investors*

**By Richard J. Newman and Peter Basso**

On Wall Street, there are bulls and bears. And now—well, now there are bloodhounds. With the collapse of the tech sector and the nose dive in the Dow, regulators, shareholders, and lawyers are baying for blood. Investigators from the Securities and Exchange Commission and the Department of Justice are examining whether questionable—and maybe criminal—behavior occurred at some of the top-tier underwriting firms that fueled the white-hot market for initial public offerings from 1998 to 2000. "I am deeply troubled by evidence of Wall Street's erosion of the bedrock of ethical conduct," said Republican Rep. Richard Baker of Louisiana last week at congressional hearings. "Our first goal . . . is to begin a process of rebuilding confidence in the market."

That's why the Securities Industry Association unveiled a new code of ethics for Wall Street stock pickers last week. The new rules are designed to prevent analysts from thinking of self-gain or their bosses' approval when they make "buy" or "hold" calls. But for many investors, the push for "integrity" comes way too late. Some of the same folks who once sued for access to IPOs are now joining class action suits,

Interested in quick quotes, statistics and industry trends? Get them in the [Money & Business](#) section.

### email tools



[Subscribe to e-newsletter](#)



[E-mail this page to a friend](#)

### from the archives

[Dark clouds over a guru to the stars:](#) Were celebrities victims of a Ponzi scheme? (5/28/01)

[Trading in false tips exacts a price:](#) A pump-and-dump perpetrator gets jail time. (2/5/01)

[Blame the pundits:](#) After the dot bombs, Wall Street has a credibility problem. (10/2/00)

# Chinese Wall Policy

---

Introduced by Brewer and Nash in 1989

The motivation for this work was to avoid that sensitive information concerning a company be disclosed to competitor companies through the work of financial consultants

- It dynamically establishes the access rights of a user based on what the user has already accessed

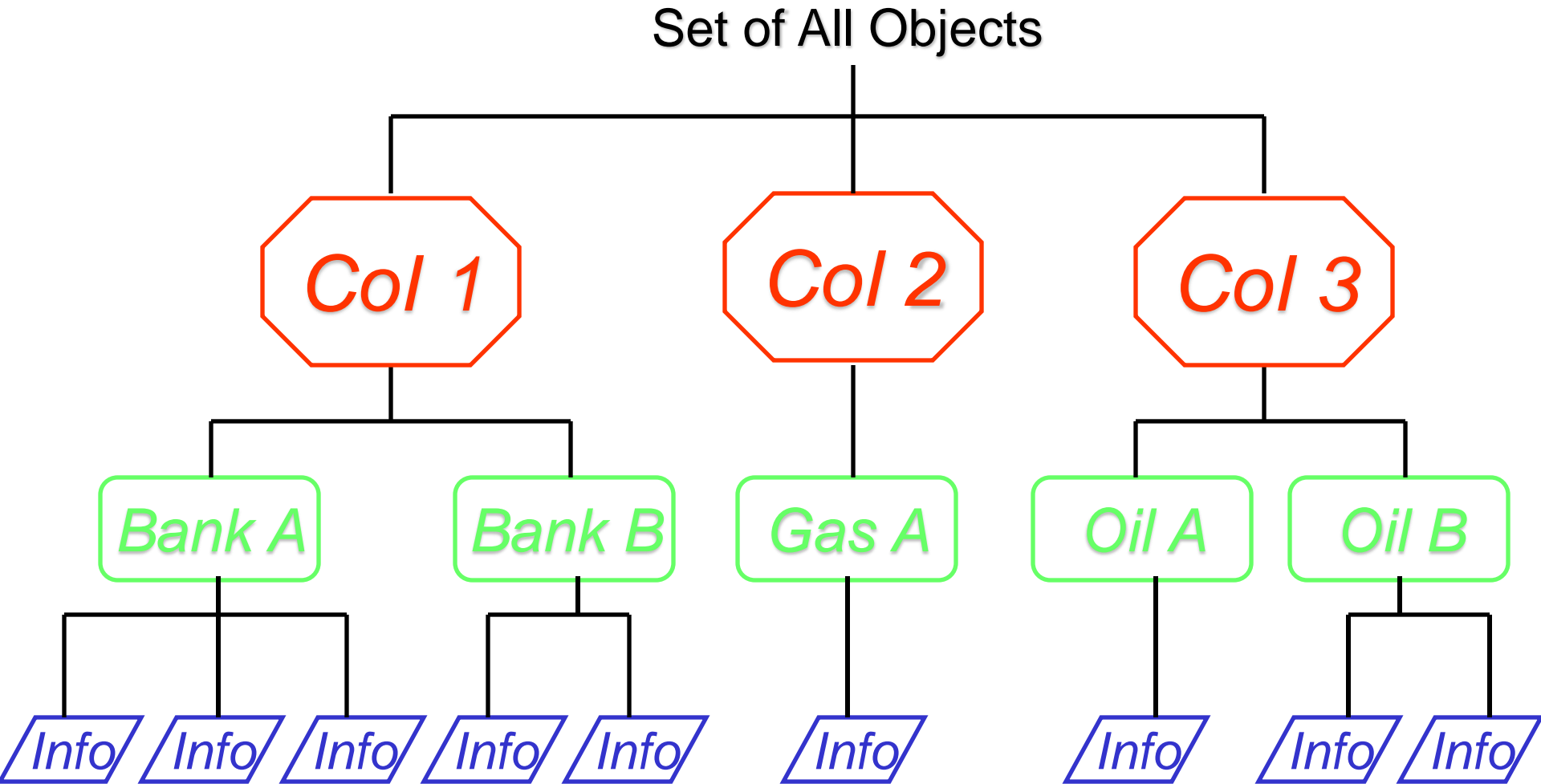
# Chinese Wall Policy

---

- *Subjects*: Active entities accessing protected objects
- *Objects*: Data organized according to 3 levels
  - » Information
  - » DataSet
  - » Conflict-of-Interest (CoI) classes
- *Access Rules*
  - » Read rule
  - » Write rule



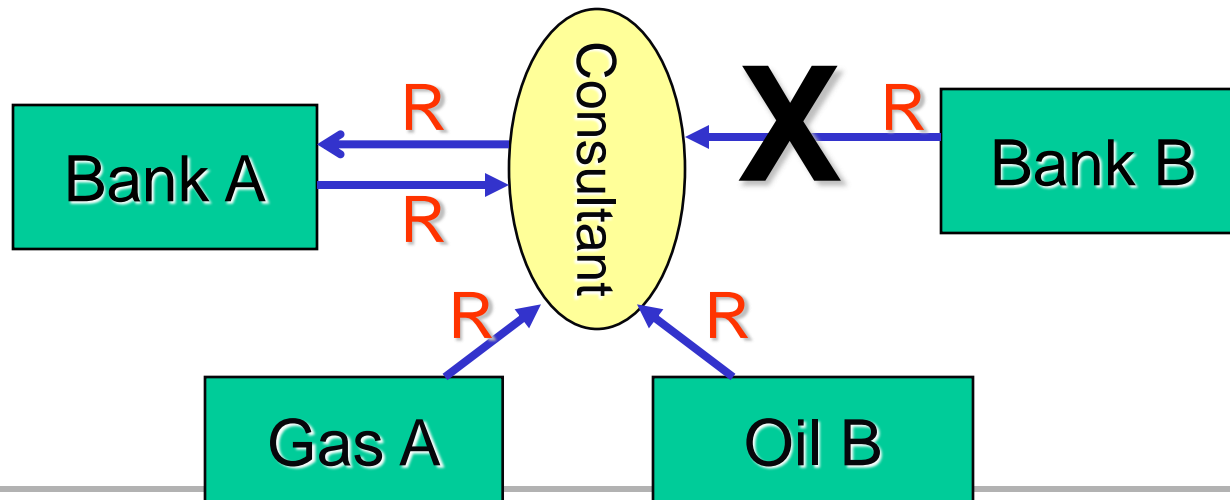
# Data Classification



# Read Rule

**Read Rule:** A subject S can **read** an object O if :

- O is in the same Dataset as an object already accessed by S OR
- O belongs to a CoI from which S has not yet accessed any information





# Comparison with Bell-LaPadula

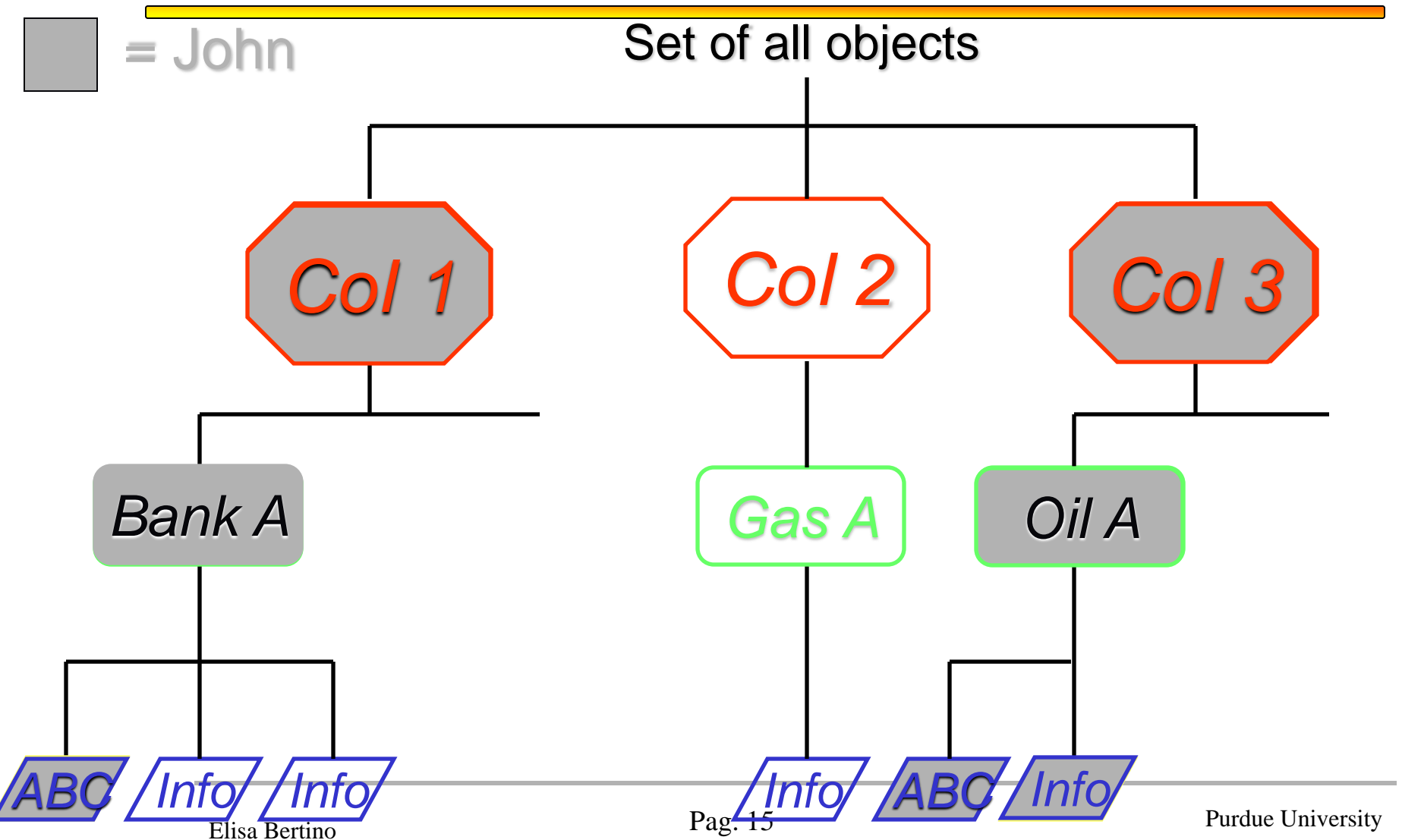
---

- The Chinese Wall Policy is a combination of free choice and mandatory control
- Initially a subject is free to access any object it wishes
- Once the initial choice is made, a *Chinese Wall* is created for that user around the dataset to which the object belongs
- Note also that a Chinese Wall can be combined with DAC policies

# Write Rule

- 
- The Read Rule does not prevent indirect flow of information
  - Consider the following case:
    - John has access to
      - Oil A and Bank A
    - Jane has access to
      - Oil B and Bank A
    - If John is allowed to read Oil A and write into Bank A, it may transfer information about Oil A that can then be read by Jane

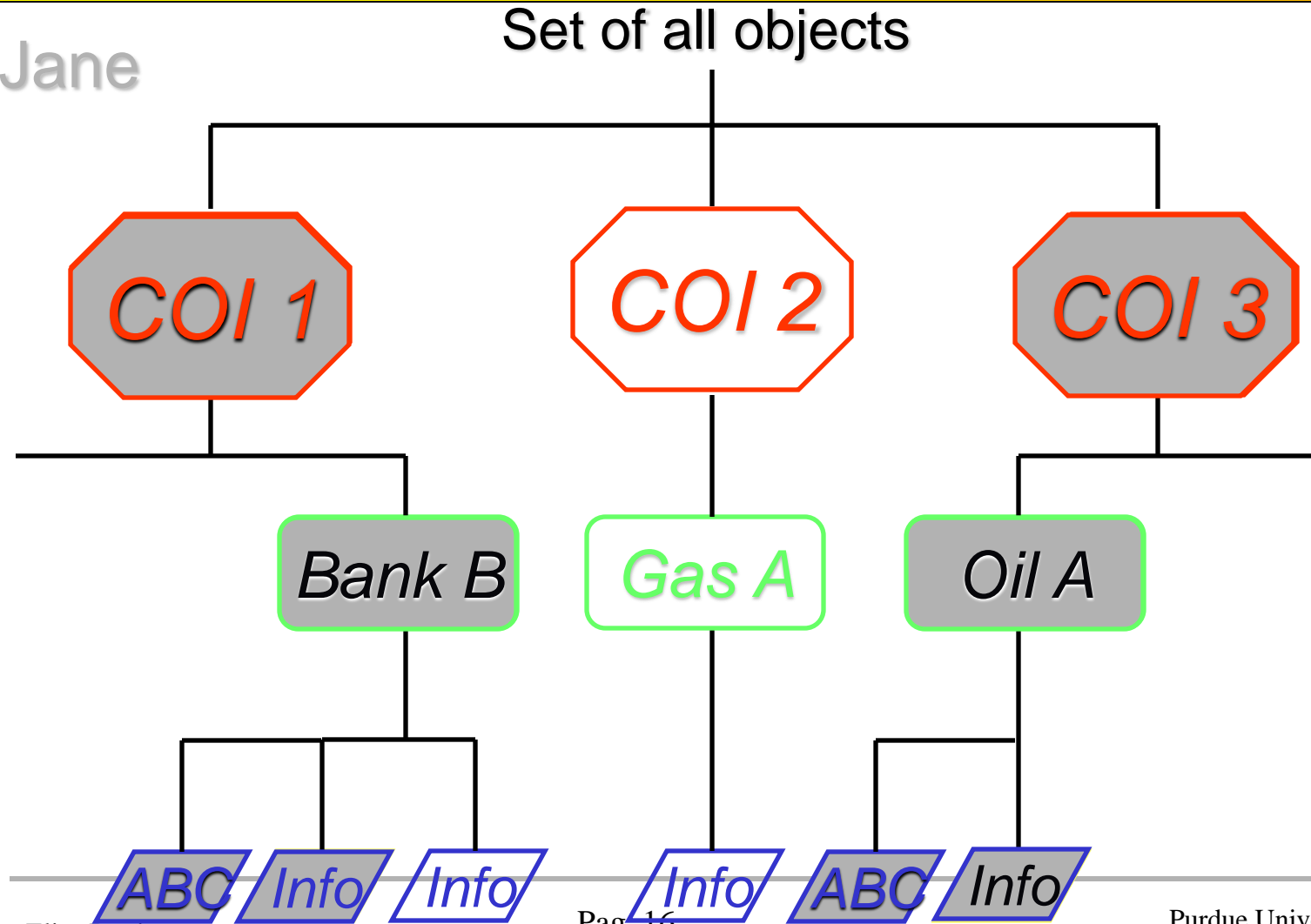
# Write Rule



# Write Rule



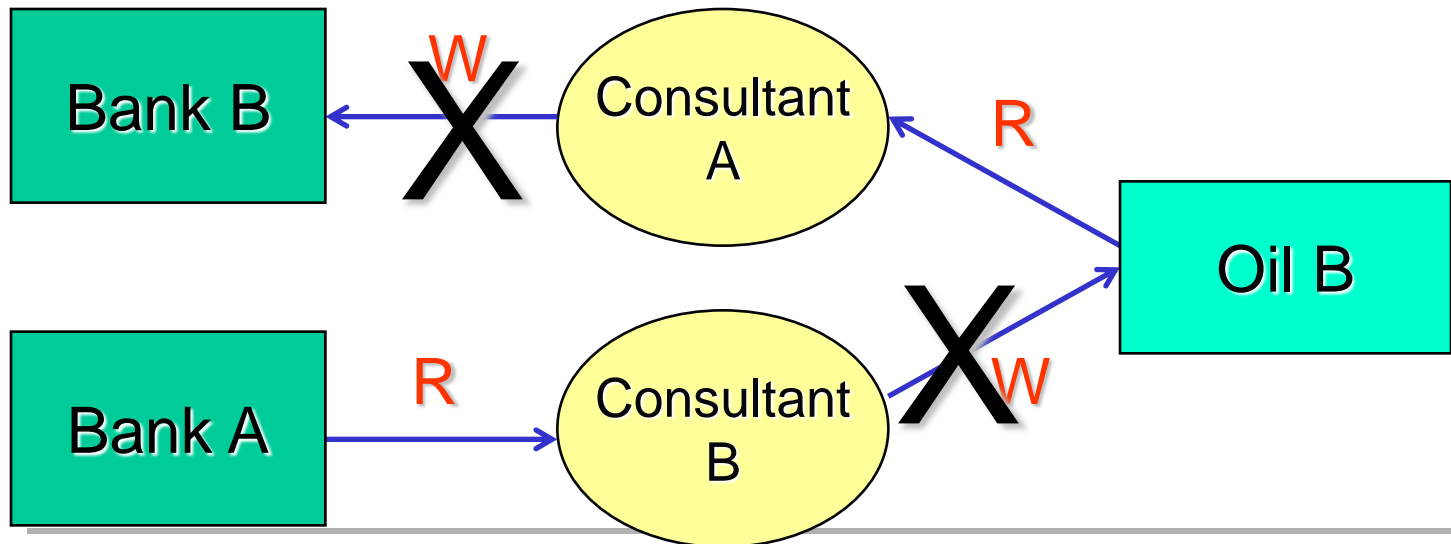
= Jane



# Write Rule

**Write Rule:** A subject S can **write** an object O if:

- S can read O according to the Read Rule AND
- No object has been read by S which is in a different company dataset to the one on which write is performed





# Write Rule

---

Thus, according to the write rule:

*The flow of information is confined to its own company dataset*

# Sanitized Information

---

- Brewer and Nash recognize the need for analysts to be able to compare information they have with that relating to other corporations
- Thus they recognize that access restriction can be lifted for **sanitized information**
- Sanitization takes the form of disguising a corporation's information, so to prevent the discovery of that corporation identity

# Criticisms to the Model (R. Sandhu)

---

The Write Rule of BN is very restrictive:

- A user that has read objects from more than one dataset is not able to write any object
- The user can only read and write objects from a single dataset

# Compare CW to Bell-LaPadula

---

- Chinese Wall (CW) is fundamentally different
  - CW has no security labels, B-LP does
  - CW has notion of past accesses, B-LP does not
- Bell-LaPadula cannot track changes over time
  - Susan becomes ill, Anna needs to take over
    - CW history lets Anna know if she can
    - No way for Bell-LaPadula to capture this
- Access constraints change over time
  - Initially, subjects in CW can read any object
  - Bell-LaPadula constrains set of objects that a subject can access
    - Can't clear all subjects for all categories, because this violates CW-simple security condition

---

# Role Based Access (RBAC) Control Model

[http://en.wikipedia.org/wiki/Role-based\\_access\\_control](http://en.wikipedia.org/wiki/Role-based_access_control)

# RBAC: Motivations

---

- One challenging problem in managing large systems is the complexity of security administration
- Whenever the number of subjects and objects is high, the number of authorizations can become extremely large
- Moreover, if the user population is highly dynamic, the number of grant and revoke operations to be performed can become very difficult to manage

# RBAC: Motivations

---

- End users often do not own the information for which they are allowed access. The corporation or agency is the actual **owner** of data objects
- Control is often based on employee functions rather than data ownership
- RBAC has been proposed as an *alternative* approach to DAC and MAC both to simplify the task of access control management and to directly support function-based access control

# RBAC: Basic Concepts

---

- Roles represent functions within a given organization and authorizations are granted to roles instead of to single users
- Users are thus simply authorized to "play" the appropriate roles, thereby acquiring the roles' authorizations



# RBAC: Benefits

---

- Because roles represent organizational functions, an RBAC model can directly support security policies of the organization
- Granting and revoking of user authorizations is greatly simplified
- RBAC models have been shown to be policy-neutral

# RBAC

---

- DBMS vendors have recognized the importance and the advantages of RBAC, and today most of the commercial DBMSs support RBAC features at some extents
- There is some consensus on a standard RBAC model
- The NIST model [Sandhu,Ferraiolo,Kuhn 00] has been the first step towards the definition of a standard
- A recent definition is by ANSI. American national standard for information technology – role based access control. ANSI INCITS 359-2004, February 2004

# NIST Model

---

- Three main levels of increasing functional capabilities
  - Core RBAC – also called Flat RBAC
  - Hierarchical RBAC
  - Constrained RBAC

# RBAC- Basic concepts

---

- *User* – is defined as a human being, a machine, a process, or an intelligent autonomous agent, etc.
- *Role* – is a function within the context of an organization with an associated semantics regarding its authority and responsibility

# RBAC- Basic concepts

---

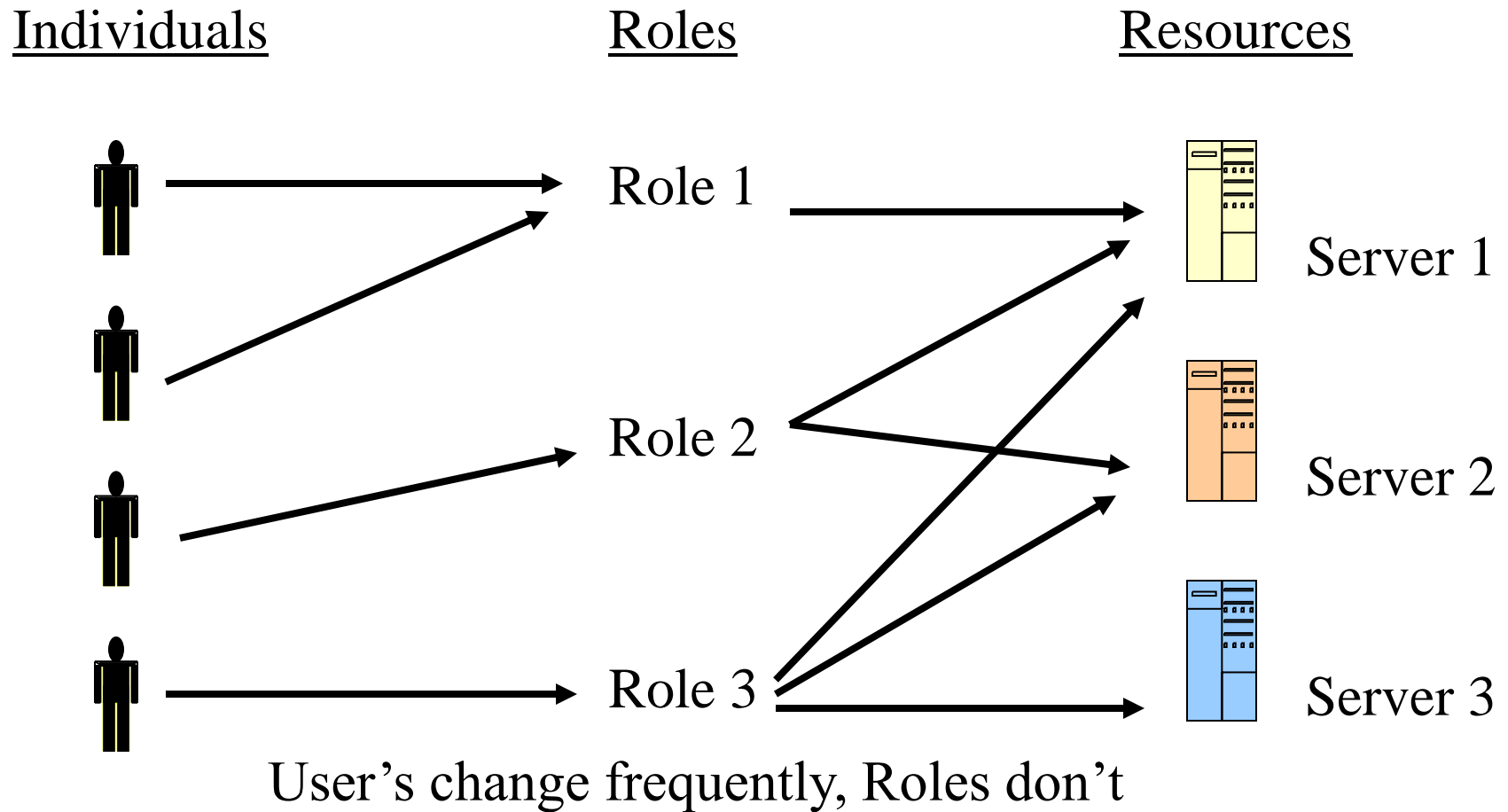
- *Permission* – is an access mode that can be exercised on objects in the system. Both objects and access modes are domain dependent. For example, in the case of databases, the object set includes tables, columns, and rows, and the access mode set includes insert, delete, and update operations.

# RBAC- Basic concepts

---

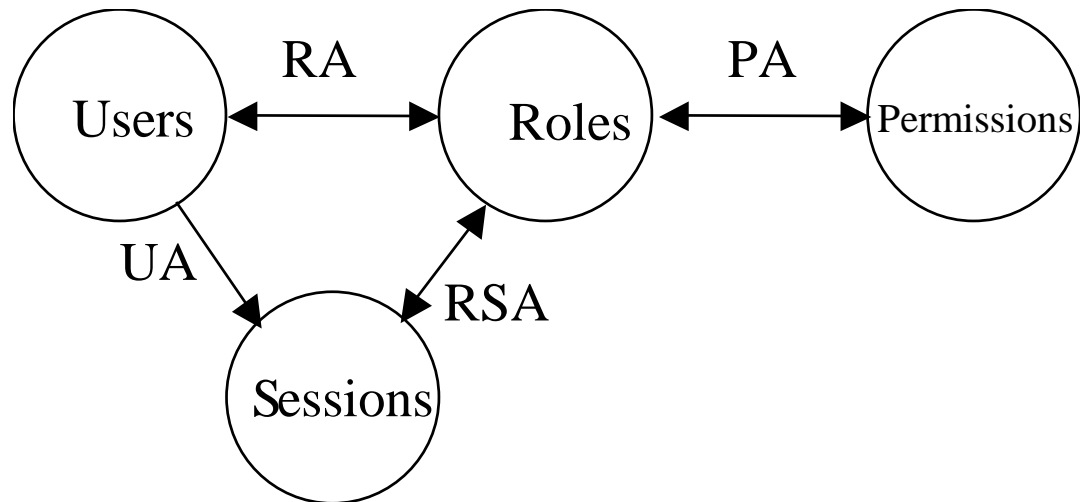
- *Session* – it is a particular instance of a connection of a user to the system and defines the subset of activated roles. At each time instant, different sessions for the same user can be active. When a user logs in the system, he/she establishes a session and, during this session, can request to activate a subset of the roles he/she is authorized to play. The user obtains all permissions associated with the role he/she has activated in the session

# Role-Based AC



# Core RBAC

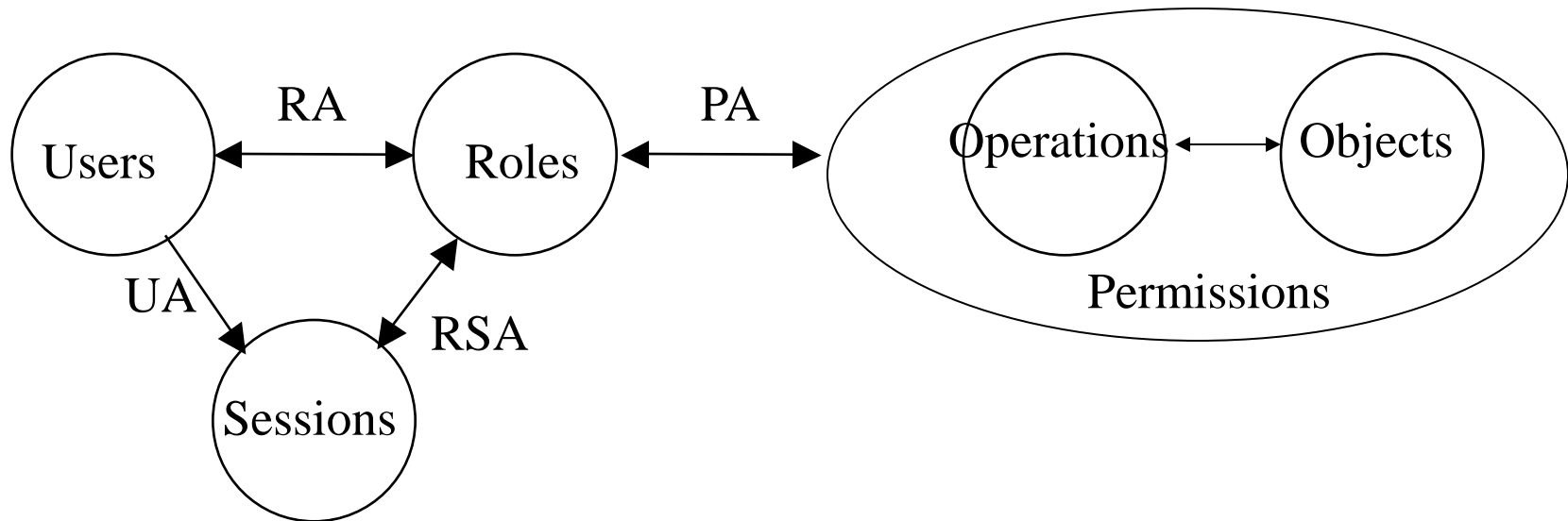
---





# Core RBAC - Permissions

---



# Core RBAC

## Sets, Functions, and Relations

---

- $USERS$ ,  $ROLES$ ,  $OPS$ , and  $OBS$
- $UA \subseteq USERS \times ROLES$ , a many-to-many mapping user-to-role assignment relation
- $assigned\_users: (r: ROLES) \rightarrow 2^{USERS}$ , the mapping of a role  $r$  onto a set of users. Formally:
  - $assigned\_users(r) = \{u \in USERS \mid (u, r) \in UA\}$
- $PRMS = 2^{(OPS \times OBS)}$ , the set of permissions
- $PA \subseteq PRMS \times ROLES$ , a many-to-many mapping permission-to-role assignment relation
- $assigned\_permissions: (r: ROLES) \rightarrow 2^{PRMS}$ , the mapping of a role  $r$  onto a set of permissions. Formally:
  - $assigned\_permissions(r) = \{p \in PRMS \mid (p, r) \in PA\}$

# Core RBAC

## Sets, Functions, and Relations

---

- $Op(p : PRMS) \rightarrow \{op \subseteq OPS\}$ , the permission to operation mapping, which gives the set of operations associated with permission  $p$
- $Ob(p : PRMS) \rightarrow \{op \subseteq OBS\}$ , the permission to object mapping, which gives the set of objects associated with permission  $p$
- $SESSIONS$  = the set of sessions
- $session\_users(s : SESSIONS) \rightarrow USERS$ , the mapping of session  $s$  onto the corresponding user
- $session\_roles(s : SESSIONS) \rightarrow 2^{ROLES}$ , the mapping of session  $s$  onto a set of roles. Formally:
  - $session\_roles(s) = \{r \in ROLES \mid (session\_users(s), r) \in UA\}$
- $avail\_session\_perms(s : SESSIONS) \rightarrow 2^{PRMS}$ , the permissions available to a user in a session. Formally:
  - $avail\_session\_perms(s) = \bigcup_{r \in session\_roles(s)} assigned\_permissions(r)$

# Core RBAC - Sessions

---

- The notion of session is quite abstract – it is defined as “*a mapping between a user and an activated subset of roles that are assigned to the user*”
- Basic distinction:
  - *Single-role activation (SRA)* Only one role can be activated
  - *Multi-role activation (MRA)* Multiple roles can be activated in one session, and dynamic separation of duty constraints may be used to restrict concurrent activation of some roles
  - There are trade-offs between the use of these two types of session

# Hierarchical RBAC - Motivations

---

- Role hierarchies are a natural means for structuring roles to reflect an organization's line of authority and responsibility

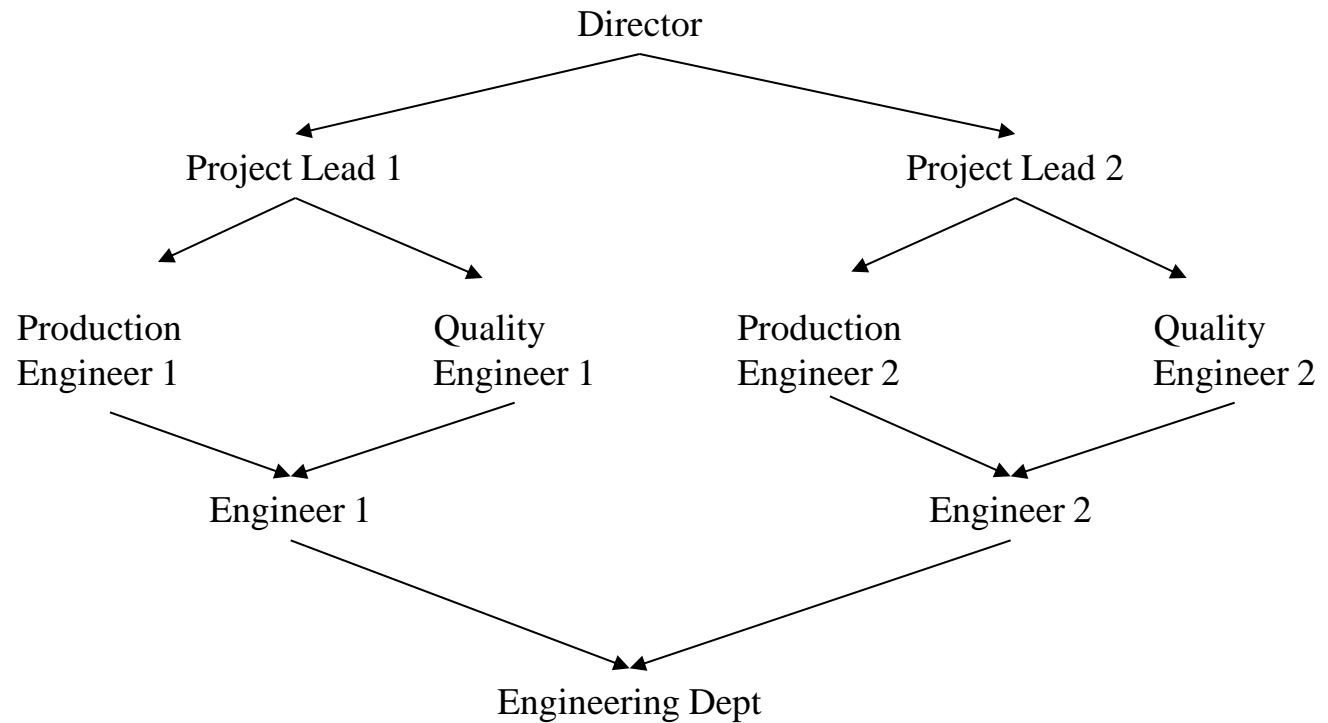
# Hierarchical RBAC - Model

---

- $RH \subseteq ROLES \times ROLES$  it is a relation defined on the set of roles in the system
- It has to be irreflexive and acyclic. We refer to this relation as *dominance relation*; if  $(r_i, r_j) \in RH$  we say that  $r_i$  dominates  $r_j$
- We also define a partial order  $\geq$  which is the reflexive and transitive closure of  $RH$ .
- An RBAC system may choose to store  $\geq$  or to compute it when needed

# Example of RH

---



# Hierarchical RBAC - Semantics

---

- User Inheritance (*UI*): All users authorized for a role  $r$  are also authorized for any role  $r'$  if  $r \geq r'$
- Permission Inheritance (*PI*): A role  $r$  is authorized for all permissions for which any role  $r'$ , such that  $r \geq r'$ , is authorized
- Activation Inheritance (*AI*): Activating a role  $r$  automatically activates all roles  $r'$ , such that  $r \geq r'$ . This semantics can be used only if MRA sessions are used



# Constrained RBAC

---

- Constrained RBAC is an RBAC model with the capability of supporting *Separation of Duties* policies
- Two main categories:
  - Static SoD
  - Dynamic SoD

# RBAC - SoD Policies

---

- Enforces conflict of interest policies employed to prevent users from exceeding a reasonable level of authority for their position.
- Ensures that failures of omission or commission within an organization can be caused only as a result of collusion among individuals.

# SoD Definitions

---

- ANSI: “Dividing responsibility for sensitive information so that no individual acting alone can compromise the security of the data processing system”
- The U.S. Office of Management and Budget’s Circular A-123: “Key duties and responsibilities in authorizing, processing, recording, and reviewing official agency transactions should be separated among individuals”.

# RBAC – Static SoD Constraints

---

- SSoD places restrictions on the set of roles and in particular on their ability to form *RA* relations.
- No user is assigned to  $t$  or more roles in a set of  $m$  roles
- Prevents a person being authorized to use too many roles
- These constraints can be enforced based on the users assigned to each role

# RBAC – Dynamic SoD Constraints

---

- These constraints limit the number of roles a user can activate in a single session
- Examples of constraints:
  - No user may activate  $t$  or more roles from the roles set in each user session.
  - If a user has used role  $r1$  in a session, he/she cannot use role  $r2$  in the same session
- Enforcement of these roles requires keeping the history of the user access to roles within a session

# RBAC System and Administrative Functional Specification

---

- Administrative Operations
  - Create, Delete, Maintain elements and relations
- Administrative Reviews
  - Query operations
- System Level Functions
  - Creation of user sessions
  - Role activation/deactivation
  - Constraint enforcement
  - Access Decision Calculation

# Integrity Policies: BIBA's model

---

- Overview
- Requirements
- Biba's models

# Requirements of Policies for Commercial Applications [Lipner 1982]

---

1. Users will not write their own programs, but will use existing production programs and databases.
2. Programmers will develop and test programs on a non-production system; if they need access to actual data, they will be given production data via a special process, but will use it on their development system.
3. A special process must be followed to install a program from the development system onto the production system.
4. The special process in requirement 3 must be controlled and audited.
5. The managers and auditors must have access to both the system state and the system logs that are generated.

*The emphasis of these requirements is on integrity.*



# Requirements and Principles of Operation

---

- The requirements suggest several principles
  - *Separation of duty*. If two or more steps are required to perform a critical function, at least two different subjects should perform them. Moving an application from the development system to the production system is an example of critical function.
  - *Separation of function*. Different functions are executed on different sets of data. For example, developers do not develop new programs in the production environment. Also they do not process production data in the development environment. If they need data, depending on the sensitivity of data, sanitized versions of these data may be given to them.
  - *Auditing*. It is the process of analyzing systems to determine what actions took place and who performed them

# Biba Integrity Model

---

- The Biba model associates an integrity level with both objects and subjects
- These levels form the basis for expressing integrity policies that refer to the corruption of ‘clean’ high level entities by ‘dirty’ low level entities
- In the integrity lattice, information may only flow downwards

# Biba Integrity Model

---

- Set of subjects  $S$ , objects  $O$ , integrity levels  $I$ , relation  $\leq \subseteq I \times I$  holding when the second dominates the first
- $i: S \cup O \rightarrow I$  gives integrity level of an object or of a subject
- $\underline{r}: S \times O$  means  $s \in S$  can read  $o \in O$
- $\underline{w}$ ,  $\underline{x}$  defined similarly

# Intuition for Integrity Levels

---

- The higher the level, the more confidence
  - That a program will execute correctly
  - That data is accurate and/or reliable
- Important point: *integrity levels are **not** security levels*
- Integrity labels are assigned and maintained separately, because the reasons behind the labels are different

# Biba's Model

---

- Access Control Rules
  1.  $s \in S$  can read  $o \in O$  iff  $i(s) \leq i(o)$
  2.  $s \in S$  can write to  $o \in O$  iff  $i(o) \leq i(s)$
  3.  $s_1 \in S$  can execute  $s_2 \in S$  iff  $i(s_2) \leq i(s_1)$
- No actual implementations in real products have been reported for the Biba model
- The problem of integrity of information requires articulated solutions

# References (from Wikipedia)

---

- [http://en.wikipedia.org/wiki/Discretionary\\_access\\_control](http://en.wikipedia.org/wiki/Discretionary_access_control)
- [http://en.wikipedia.org/wiki/Mandatory\\_Access\\_Control](http://en.wikipedia.org/wiki/Mandatory_Access_Control)
- [http://en.wikipedia.org/wiki/Bell-La\\_Padula\\_model](http://en.wikipedia.org/wiki/Bell-La_Padula_model)
- [\*\*http://en.wikipedia.org/wiki/Role-based\\_access\\_control\*\*](http://en.wikipedia.org/wiki/Role-based_access_control)