

DAC models - DBMS vs OS

- Increased number of objects to be protected
- Different granularity levels (relations, tuples, single attributes)
- Protection of logical structures (relations, views) instead of real resources (files)
- Different architectural levels with different protection requirements
- Relevance not only of data physical representation, but also of their semantics

The Trojan Horse

- DAC models are unable to protect data against Trojan Horses embedded in application programs
- MAC models were developed to prevent this type of illegal access

MAC

- MAC specifies the access that subjects have to objects based on subjects and objects classification
- This type of security has also been referred to as *multilevel security*
- Database systems that satisfy multilevel security properties are called multilevel secure database management systems (MLS/DBMSs)
- Many of the MLS/DBMSs have been designed based on the Bell and LaPadula (BLP) model
- Rules specify granting of access
- Also called rule-based access control

MAC

- Access control is based on rules (policy) set by administrator
- AC policy is enforced by the reference monitor and cannot be changed by users
- Subjects cannot leak access rights to others, e.g.
 - User can read a secret file but cannot copy, print or email it
 - Process can download data from the Internet or write to the file system, but not both → users cannot download malware
- MAC is also called rule-based AC

Origins of MAC

- MAC originates from **military policies**
 - Officer can read a secret document but not make a copy or remove it from the premises
 - Intelligence officer may not be allowed to read his own reports after submitting them
 - Officer who had contact with foreign agents may lose access to classified information

MAC in commercial systems

- **Isolation policies:** Apps in a phone cannot communicate with each other unless permitted by a policy
- **Digital Rights Management (DRM):** User can play the music he/she purchased, but cannot share
- **Secure document viewers:** Protected PDF files do not allow editing, printing or cut-and-paste; Snapchat supports sharing of photos for a limited time

MAC outside computers

- Examples of MAC-like policies outside computers:
 - Biometric authentication prevents sharing of capabilities, e.g. photo on driver's license, face recognition in ATMs
 - Admit-one event ticket; amusement park wristband
 - In UK, jurors must not read newspapers or watch TV about the court case to avoid external influence

Confidentiality Policy: Bell-LaPadula Model

- Formally models military-style classification
 - Multi-level access control
- Mandatory access control (Security Level)
 - Subject has clearance $L(S) = I_s$
 - Object has classification $L(O) = I_o$
 - Clearance/Classification ordered
 - $I_i < I_{i+1}$
- Discretionary access control
 - Matrix: Subject has read (write) on Object

Bell and LaPadula Model

Elements of the model:

- *objects* - passive entities containing information to be protected
- *subjects*: active entities requiring accesses to objects (*users, processes*)
- *access modes*: types of operations performed by subjects on objects
 - read: reading operation
 - append: modification operation
 - write: both reading and modification

Bell and LaPadula Model

- Subjects are assigned **clearance** levels and they can operate at a level up to and including their clearance levels
- Objects are assigned **sensitivity** levels
- The clearance levels as well as the sensitivity levels are called **access classes**

BLP Model - access classes

- An access class consists of two components
a security level a category set
- The security level is an element from a totally ordered set - example
 $\{\text{Top Secret (TS), Secret (S), Confidential (C), Unclassified (U)}\}$ where
 $\text{TS} > \text{S} > \text{C} > \text{U}$
- The category set is a set of elements, dependent from the application area in which data are to be used - example
 $\{\text{Army, Navy, Air Force, Nuclear}\}$

BLP Model - Access classes

Access class $c_i = (L_i, SC_i)$ **dominates** access class $c_k = (L_k, SC_k)$, denoted as $c_i \geq c_k$, if both the following conditions hold:

- $L_i \geq L_k$ The security level of c_i is greater or equal to the security level of c_k
- $SC_i \supseteq SC_k$ The category set of c_i includes the category set of c_k

BLP Model - Access classes

- If $L_i > L_k$ and $SC_k \subset SC_i$, we say that c_i **strictly dominates** c_k
- c_i and c_k are said to be **incomparable** (denoted as $c_i < > c_k$) if neither $c_i \geq c_k$ nor $c_k \geq c_i$ holds

BLP Model - Examples

Access classes

$$C_1 = (TS, \{Nuclear, Army\})$$

$$C_2 = (TS, \{Nuclear\})$$

$$C_3 = (C, \{Army\})$$

- $C_1 \geq C_2$
- $C_1 > C_3$ $(TS > C \text{ and } \{Army\} \subset \{Nuclear, Army\})$
- $C_2 < > C_3$

Example

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 })

BLP Model - Axioms

- **Simple Security Condition:** *S can read O if and only if*
 - *S dom O and*
 - *S has discretionary read access to O*

Example

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 })

Paul is cleared into security level (SECRET, { EUR, US, NUC })

BLP Model - Axioms

- **Simple Security Condition:** *S can read O if and only if*
 - *S dom O and*
 - *S has discretionary read access to O*
- ***-Property:** *S can write O if and only if*
 - *O dom S and*
 - *S has discretionary write access to O*
- **Secure system:** *One with above properties*
- **Theorem:** *Let Σ be a system with secure initial state σ_0 , T be a set of state transformations. If every element of T preserves the simple security condition, and the *-property, then every state σ_i , $i \geq 0$, is secure*

BLP Model - Axioms

- The simple security property prevents subjects from reading data with access classes dominating or incomparable with respect with the subject access class
- It therefore ensures that subjects have access only to information for which they have the necessary access class

BLP Model - Axioms

- The *-property has been defined to prevent information flow into objects with lower-level access classes or incomparable classes
- For a system to be secure both properties must be verified by any system state

Problem - No write-down

Cleared subject can't talk to non-cleared subject

- Any write from l_i to l_k , $i > k$, would violate *- property
 - Subject at l_i can only write to l_i and above
- Any read from l_k to l_i , $i > k$, would violate simple security property
 - Subject at l_k can only read from l_k and below
- Subject at level i can't write something readable by subject at k
 - Not very practical

Problem

- Colonel has (Secret, {Nuclear, Army}) clearance
- Major has (Secret, {Army}) clearance
- The Colonel needs to send a message to the Major. The Colonel cannot write a document that has access class (Secret, {Army}) because such a document would violate the *-property
- To address this problem the model provides a mechanism; each subject has a *maximum access class* and a *current access class*
- A subject may change its access class; the current access class must however be dominated by the maximum access class

Changing Access classes

Simple Security and the *-property constrain accesses to objects by subjects according to the relationship between their labels.

But what if the labels are allowed to change?
Assume you could somehow change an object's label from (Top Secret: { Crypto }) to (Unclassified: {})
independent of the object's contents. This would clearly violate confidentiality. Why?

Tranquility Properties

We clearly need an additional rule that governs changing access classes.

- The Strong Tranquility Property: Subjects and objects do not change access class during the lifetime of the system.
- The Weak Tranquility Property: Subjects and objects do not change access class in a way that violates the “spirit” of the security policy

An Example of Application

The DG/Unix B2 System

- DG/Unix Provides mandatory access controls
 - MAC label identifies security level
 - Default labels, but can define others
- Initially
 - Processes (users) assigned MAC label of parent
 - Initial label assigned to user, kept in Authorization and Authentication database
 - Object assigned label at creation
 - Explicit labels stored as part of attributes
 - Implicit labels determined from parent directory

Directory Problem

- Process p at access class MAC_A tries to create file $/tmp/x$
- $/tmp/x$ exists but has access class MAC_B
 - Assume $MAC_B \geq MAC_A$ (MAC_B dominates MAC_A)
- Create fails
 - Now p knows a file named x with a higher label exists
- Fix: only programs with same MAC label as directory can create files in the directory
 - This solution is too restrictive

Multilevel Directory

- Directory with a set of subdirectories, one per label
 - Not normally visible to user
 - *p* creating */tmp/x* actually creates */tmp/d/x* where *d* is directory corresponding to MAC_A
 - All *p*'s references to */tmp* go to */tmp/d*
- The directory problem illustrates an important point:

Sometimes it is not sufficient to hide the contents of objects. Also their existence must be hidden.

Bell and LaPadula Model

- It is a significant model and it has been used in both OS and DBMS
- Some criticisms:
 - Only dealing with confidentiality, not with integrity
 - Containing covert channels

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

Biba model

- In computer systems, integrity of data and the system is often more important than confidentiality

Biba is a MAC policy for protecting integrity of data

Biba model

1. $s \in S$ can read $o \in O$ if and only if $i(s) \leq i(o)$.
2. $s \in S$ can write to $o \in O$ if and only if $i(o) \leq i(s)$.
3. $s_1 \in S$ can execute $s_2 \in S$ if and only if $i(s_2) \leq i(s_1)$.

Chinese Wall Model

- Supports confidentiality and integrity
- Models conflict of interest
 - object sets CD
 - conflict of interest sets COI
- Principle: Information can't flow between items in a COI set

Conflict of Interest

- It is a well known concept
- An example in the financial world is that of a market analyst working for a financial institution providing corporate business services
- Such analyst must uphold the confidentiality of information provided to him by his firm's client; this means he/she cannot advise corporations where he/she has *insider knowledge* of the plans, status and standing of a competitor
- However the analyst is free to advice corporations which are not in competition with each other, and also to draw on general market information

Chinese Wall Policy

Introduced by Brewer and Nash in 1989

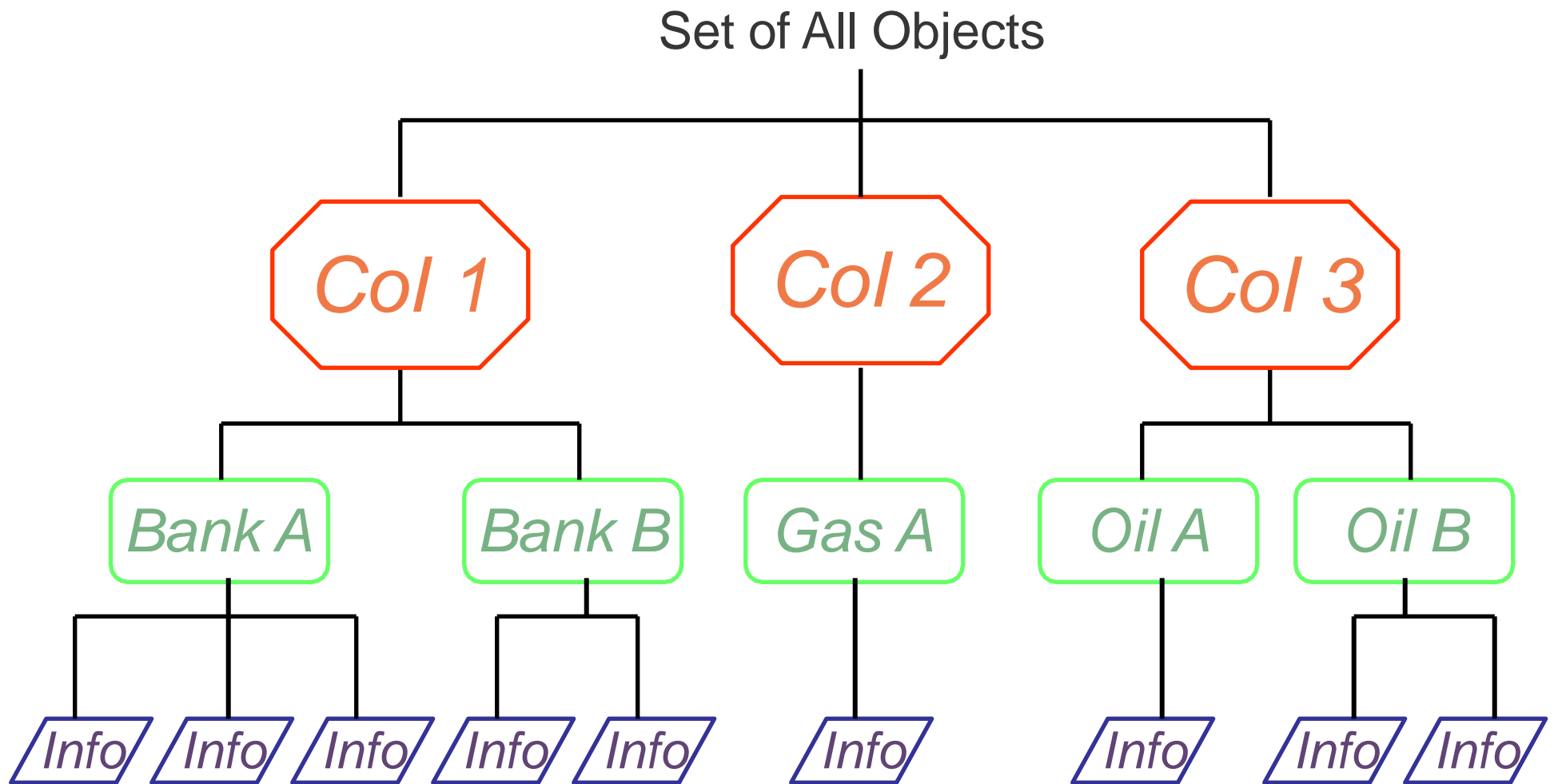
The motivation for this work was to avoid that sensitive information concerning a company be disclosed to competitor companies through the work of financial consultants

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

Chinese Wall Policy - Definitions

- *Objects*: items of information related to a company
- *Company dataset* (CD): contains objects related to a single company
 - Written $CD(O)$
- *Conflict of interest class* (COI): contains datasets of companies in competition
 - Written $COI(O)$
 - Assume: each object belongs to exactly one COI class

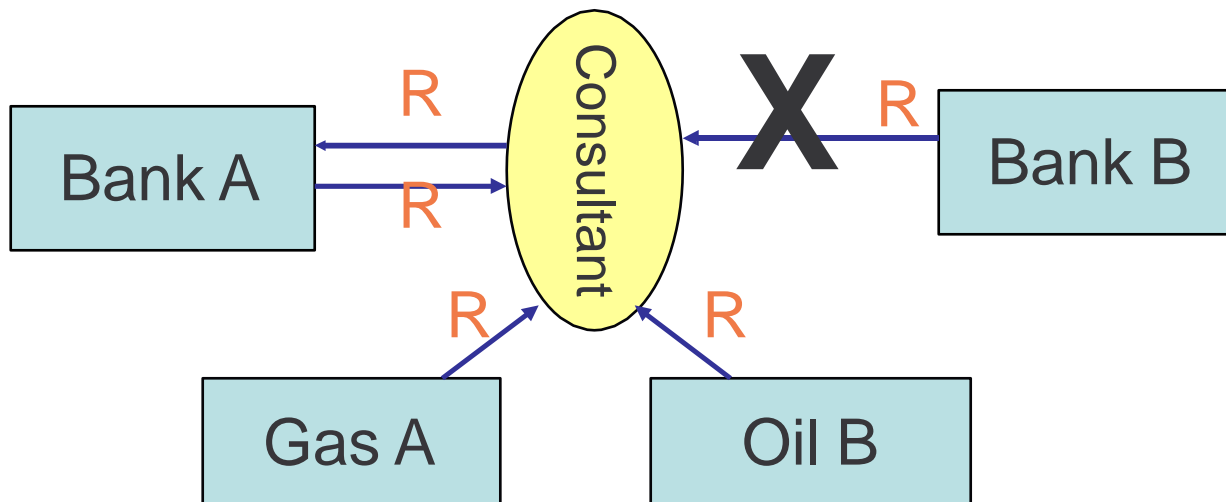
Data Classification




Read Rule

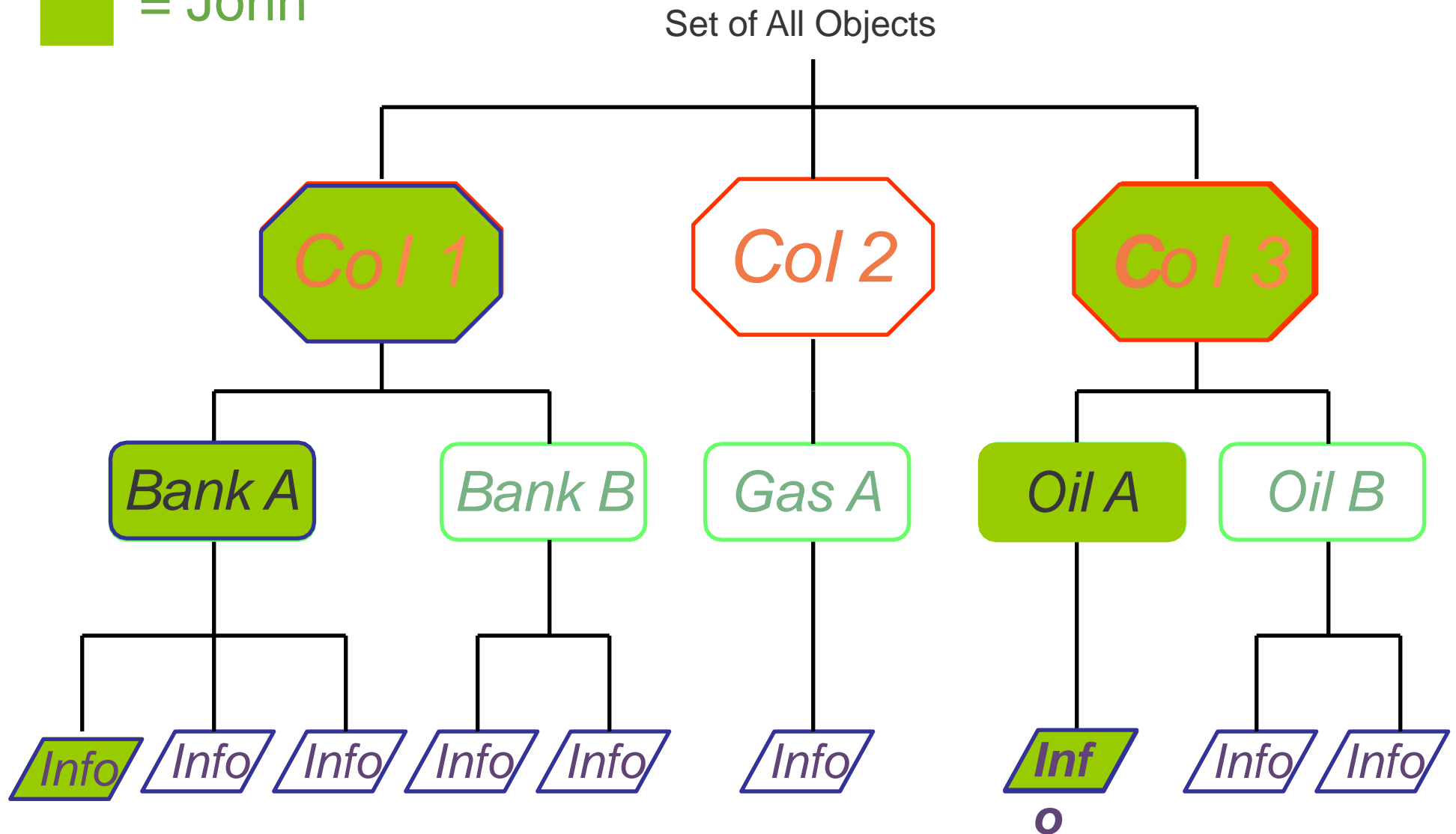
Read Rule: A subject S can **read** an object O if :

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



Read Rule

 = John




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

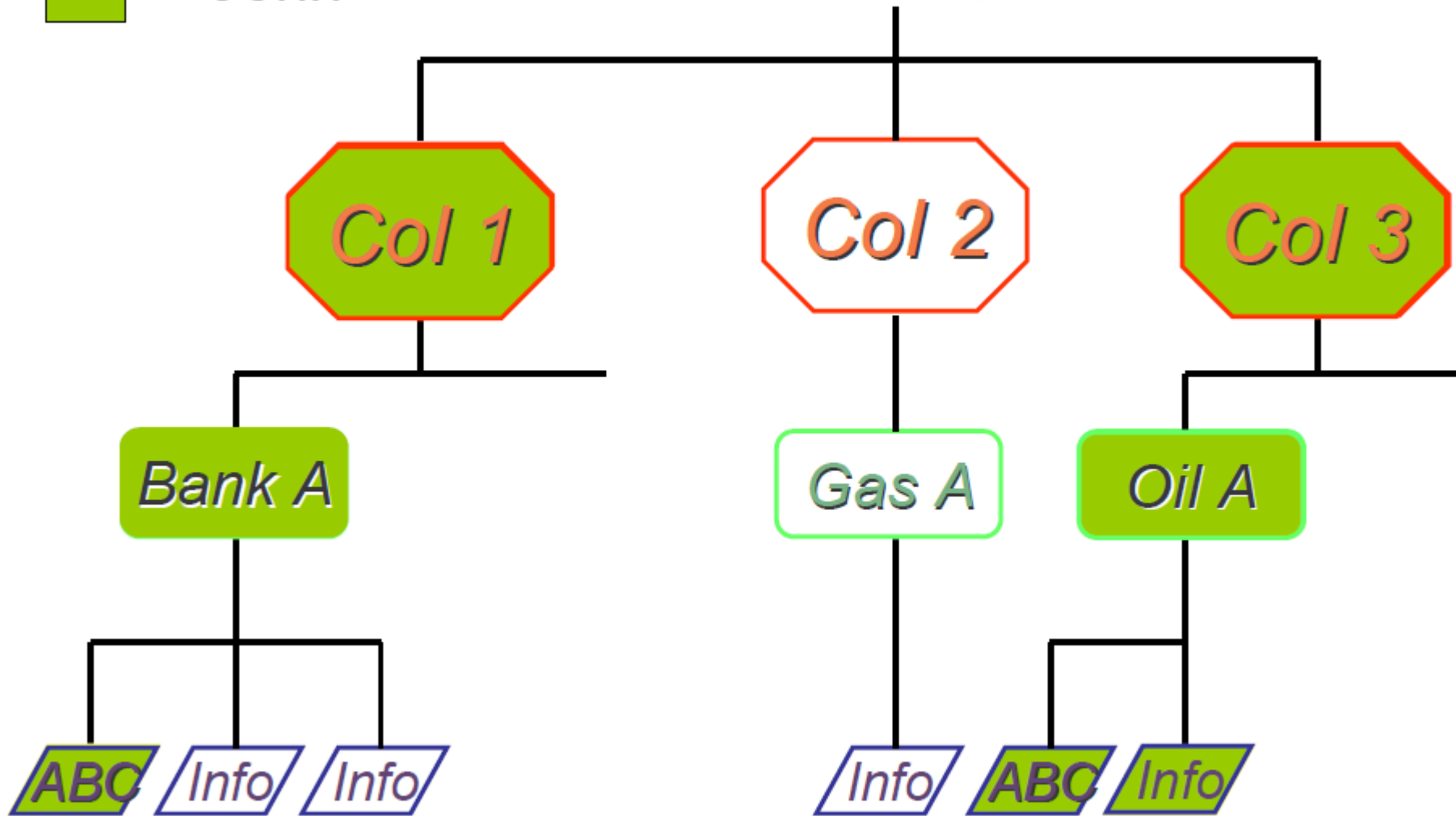
Write Rule

- 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

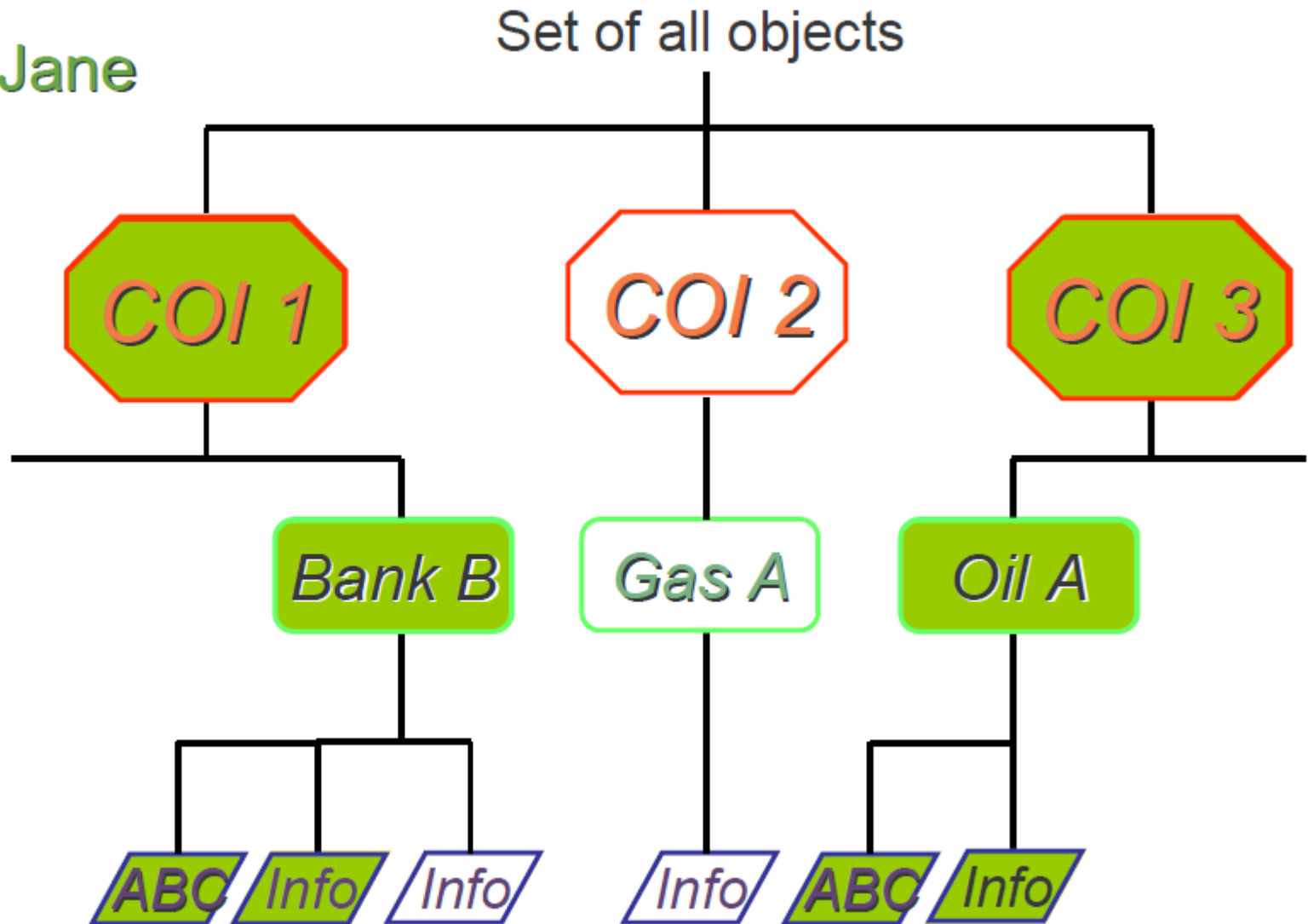
 = John

Set of all objects



Write Rule

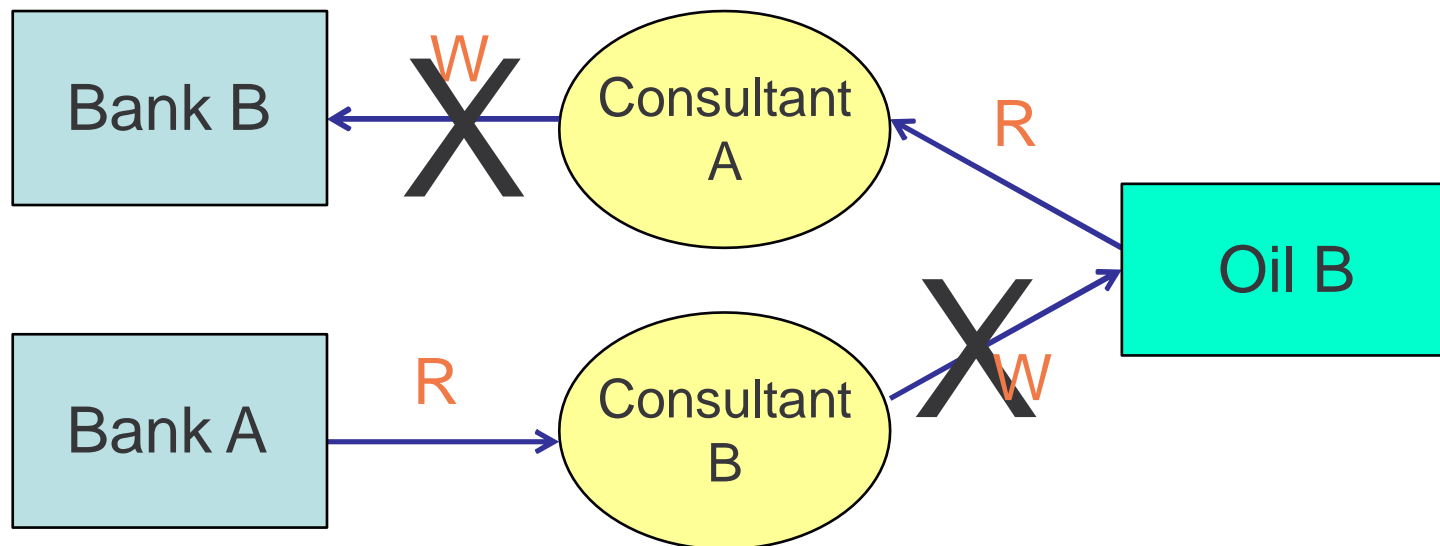
 = Jane



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

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