Data Security & Privacy

CIS 545

Security & Integrity Policies

Birhanu Eshete
birhanu@umich.edu

Lecture Goals

- Security policy w.r.t the CIA Triad
- Confidentiality: BLP
- Integrity: Biba, Clark-Wilson
- Hybrid: Chinese Wall

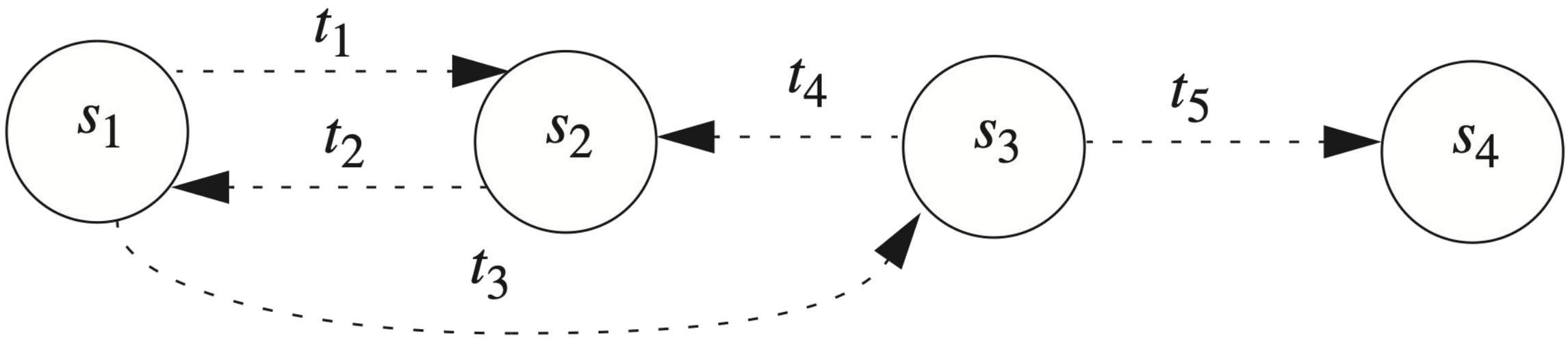
Security Policy

- A computer system: a FSA with a set of transition functions that change the state
- Security Policy: a statement that partitions the states of the system into a set of authorized (secure) states and a set of unauthorized (nonsecure) states
  - sets the context in which we can define a secure system
- Note: what is secure under one policy may not be secure under a different policy

Secure System

- Def: a system that starts in an authorized state and cannot enter an unauthorized state

- Example: 4-states (s_1-s_4), 5-transitions (t_1-t_5), A = {s_1, s_2}, UA = {s_3, s_4}



- Is the above system secure?

- **Def**: a breach of security occurs when a system enters unauthorized state

Security Policy w.r.t. Confidentiality

- Identifies states in which info. leaks to those not authorized to receive it
- Dynamic changes of authorization (includes temporal element)
- Example: contractor's access to proprietary info during and after a NDA

Security Policy w.r.t. Integrity

- Identifies authorized ways in which information may be altered

- Example: separation of duties in transaction completion

Security Policy w.r.t. Availability

- what services must be provided to subjects entitled to the services (often with service parameters)

- **Example**: 24/7 availability for web servers, <=30sec response time for login

- **Context sensitive**: 30min. delay at DMV vs. 30min. delay in the ER

Types of Security Policies

- Confidentiality:
  - e.g., Bell-LaPadula (we've covered this)
- Integrity:
  - e.g., Biba, Clark-Wilson
- Hybrid:
  - e.g., Chinese Wall+ Bell-LaPadula, Chinese Wall + Clark-Wilson

Integrity Policies

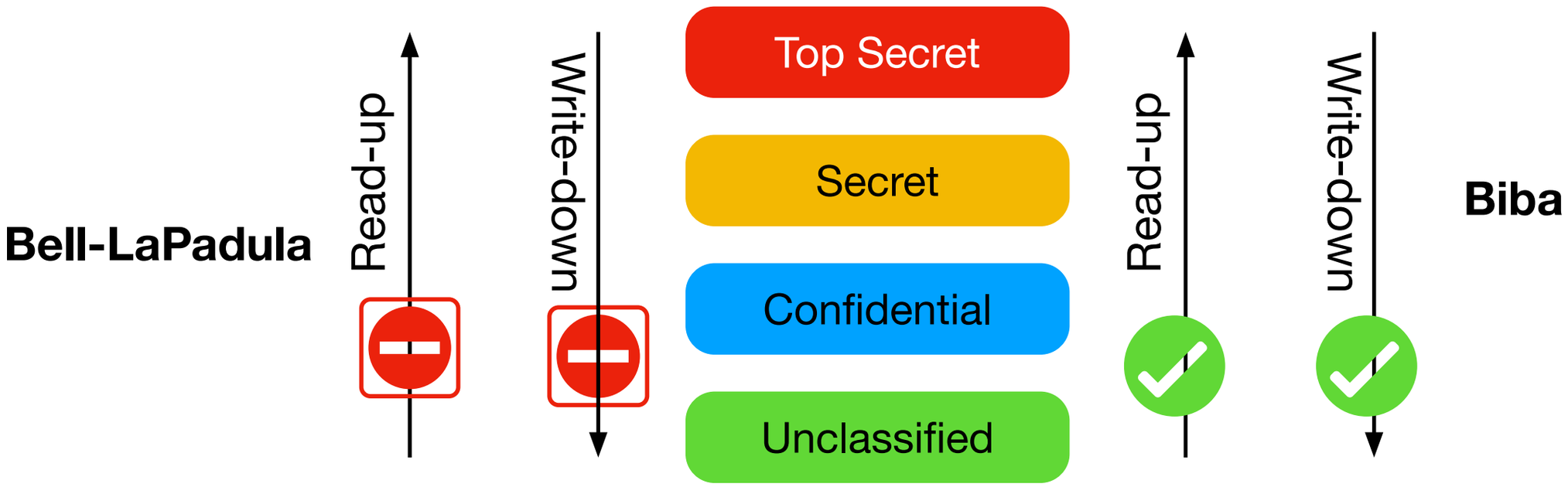
- Focus: on integrity, not confidentiality
- Example: an inventory control system
  - data disclosure (important)
  - data randomly changed (paramount)
- Commercial and industrial firms are more concerned with accuracy than disclosure

Integrity Policy Requirements

1. Users will not write their own programs, but will use existing production programs and databases (separation of duty)
2. Programmers will develop and test programs on a non-production system (separation of function)
3. A special process must be followed to install a program from the development system onto the production system (e.g., separate teams for dev & testing)
4. The special process in requirement 3 must be controlled and audited (recovery and accountability)
5. Managers and auditors must have access to both the system state and the system logs that are generated

The Biba Integrity Model

- Entities: S: set of subjects, O: set of objects, I: set of integrity levels (ordered)
- Simple Property (read up): $s \in S$ can read $o \in O$ iff $i(s) \leq i(o)$
- Star Property (write down): $s \in S$ can write to $o \in O$ iff $i(o) \leq i(s)$
- Invocation Property (execute down): $s_1 \in S$ can execute $s_2 \in S$ iff $i(s_2) \leq i(s_1)$



Example: Read

- Users: S1(C) , S2 (U), S3 (C), S4 (S), S5 (TS)
- Files: F1, F2, F3, F4, F5
- Note: F1-F5 are created by S1-S5 respectively
- Question: allow/deny for a read attempt

	F1	F2	F3	F4	F5
S1	A				
S2		A			
S3			A		
S4				A	
S5					A

Example: Write

- **Users**: S1(C) , S2 (U), S3 (C), S4 (S), S5 (TS)
- **Files**: F1, F2, F3, F4, F5
- **Note**: F1-F5 are created by S1-S5 respectively
- **Question**: allow/deny for a write attempt

	F1	F2	F3	F4	F5
S1	A				
S2		A			
S3			A		
S4				A	
S5					A

Example: Integrity and Confidentiality Levels

- Integrity Levels:

Benign Authentic (BA)	From strongly vetted source
Benign(B)	Believed to be benign, but source not well vetted
Suspicious(SP)	Not sure whether to trust or not
Unknown (U)	No reasonable basis to trust this source

- **Confidentiality Levels**:

Secret(SC)	Highly sensitive (e.g., /etc/shadow)
Sensitive(SN)	Disclosure has security impact, but less than disclosed secrets
Private (PR)	Loss may not pose a direct security threat
Public(PB)	Widely available (e.g., on public websites)

Example: Breach Detection Policies

- Untrusted Execution (UE):
  - Subject with higher integrity level executes (loads) an object with lower integrity level
- Suspicious Modification (SM):
  - Subject with lower integrity level modifies (content, attributes) an object with higher confidentiality level
- Data Leak(DL):
  - Untrusted subject writes confidential data to network
- Sensitive Read (SR):
  - A subject with lower integrity level reads an object with a higher confidentiality level

Policy Violation Tests

- Subjects: firefox(BA), bash(BA), sudo (BA), cp(BA), myprogram(B), downloaded(U) U<SP<B<BA
- Network sockets: a.a.a.a:80 (U), b.b.b.b:80 (U)
- Files: /etc/passwd(SC), /home/alice/code.cpp(SN), mycv.pdf(PB)
- **Apply the breach detection policies on the following:**

  - downloaded.write(b.b.b.b:80): DL
  - myprogram.read(/etc/passwd): SR
  - downloaded.write(/etc/passwd): SM
  - firefox.exec(myprogram): UE
  - firefox.read(a.a.a.a:80): OK
  - bash.exec(sudo): OK
  - bash.exec(myprogram): UE
  - firefox.write(b.b.b.b:80): OK

The Clark-Wilson Model

- Transactions as the basic operations of the model
- Integrity is modeled in terms of operations on data
- Data in a consistent or valid state if it satisfies given properties

Consistency Intuitively

- Example: a bank account
- D: $ deposited so far today
- W: $ withdrawn so far today
- YB: $ in all accounts at the end of yesterday
- TB: $ in all accounts so far today
- Consistency property: D+YB-W = TB
- Before and after each operation (e.g., transfer, withdrawal, deposit), D+YB-W = TB should hold

Well-Formed Transaction

- **Def**: a series of operations that transition the system from one consistent state to another

- **Goal**: transfer $100 from account A to B
  - op-1: A.deduct($100)
  - op-2: B.deposit($100)

- **Note**: op-1 or op-2 may leave the system in an inconsistent state, but op-1 followed by op-2 must preserve consistency

Integrity of Transactions

- Who examines and certifies that transactions are performed correctly?
- Example: UM-Dearborn receives an invoice for an airfare by staff
  - service requested ->account determined ->invoice validated ->account debited ->check written & signed
- How many people should do these? One? Two? Three?
  - One person doing all of the above (risk: pay phony invoices?)
  - Two people doing all of the above (risk: conspire to defraud the company?)
- Principle of Separation of duty: the certifier and the implementors need to be different people

Clark-Wilson: Key Concepts

- Constrained Data Items (CDIs): objects whose integrity is protected (e.g., balances of accounts)
- Unconstrained Data Items (UDIs): objects not covered by the integrity policy (e.g., gifts to account holders)
- Integrity Verification Procedures (IVPs): test that the CDIs conform to the integrity constraints when IVPs are executed (e.g., D+YB-W = TB)
- Transformation Procedures (TPs): the only procedures allowed to modify CDIs, or take arbitrary user input and create new CDIs (e.g., account creation, depositing, withdrawing, transferring)

Clark-Wilson: Rules (1/3)

- Certification Rule 1 (CR1): when any IVP is run, it must ensure that all CDIs are in a valid state
- Certification Rule 2 (CR2): for some associated set of CDIs, a TP must transform those CDIs in a valid state into a (possibly different) valid state
  - defines a certified relation C that associates a set of CDIs with a particular TP (TP_i, (CDI_1, CDI_2, CDI_3, …, CDI_n))
  - e.g., (*check_balance()*, (Acc_1, Acc_2, Acc_3))
  - A TP may corrupt a CDI if it is not certified to work on that CDI

Clark-Wilson: Rules (2/3)

- Enforcement Rule 1 (ER1): the system must maintain the certified relations, and must ensure that only TPs certified to run on a CDI manipulate that CDI
  - if a TP *f* operates on a CDI *o*, then *(f, o)* $\in C$
- Enforcement Rule 2 (ER2): associate a user with each TP and set of CDIs
  - a set of triples *(user, TP, { CDI set })* capture the association of users, TPs, and CDIs
  - Call this relation allowed A. Of course, these relations must be certified
- Certification Rule 3 (CR3): the allowed relations must meet the requirements imposed by the principle of separation of duty

Clark-Wilson: Rules (3/3)

- Enforcement Rule 3 (ER3): The system must authenticate each user attempting to execute a TP

  - Unauthenticated users?

- Certification Rule 4 (CR4): All TPs must append enough information to reconstruct the operation to an append-only CDI

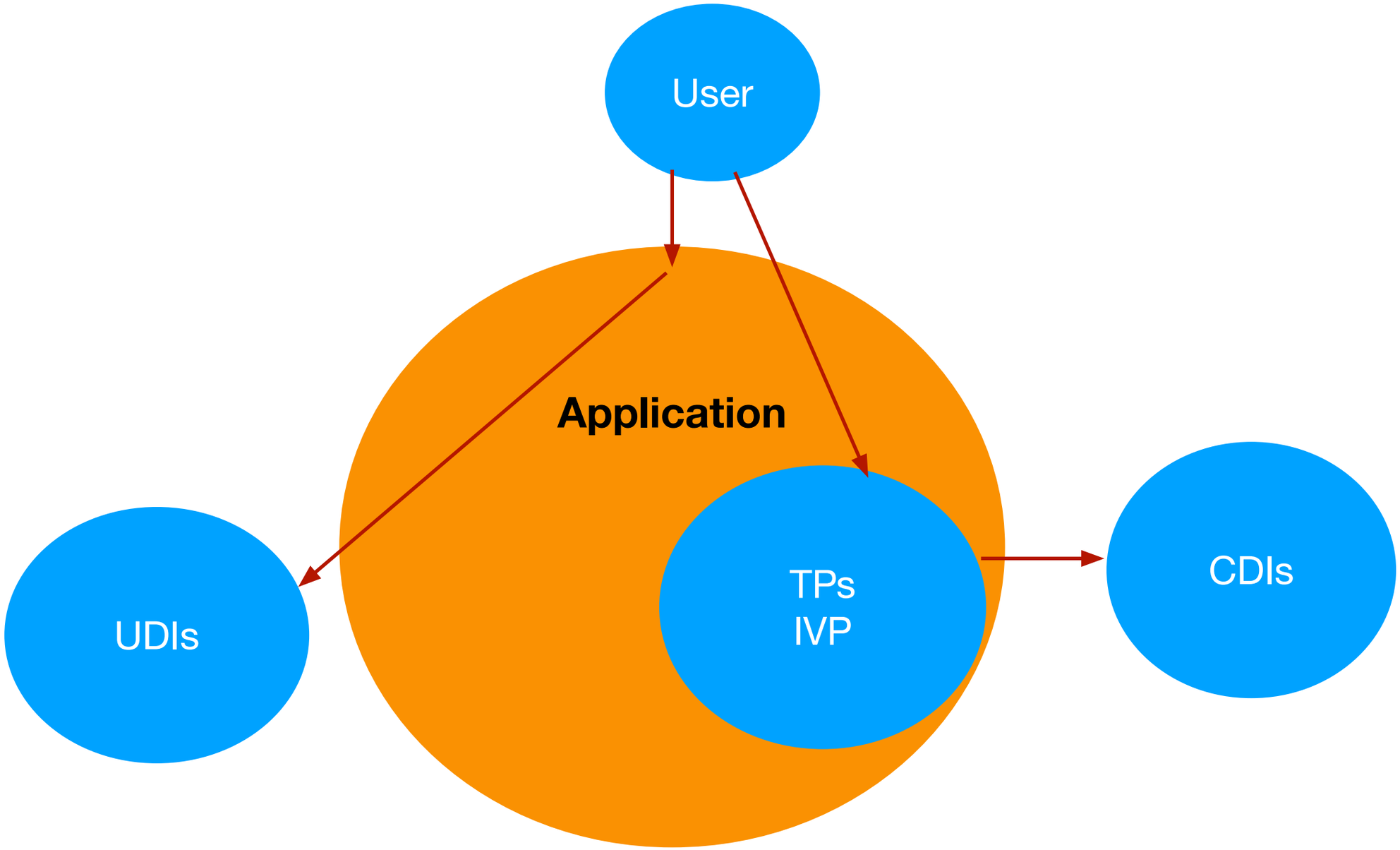
  - no TP can overwrite the log

- Certification Rule 5 (CR5): Any TP that takes as input a UDI must either reject it or transform it to a CDI

- Enforcement Rule 4 (ER4): Only the certifier of a TP may change the list of entities (e.g., CDIs) associated with that TP

  - Certifier cannot execute

Clark-Wilson Summary



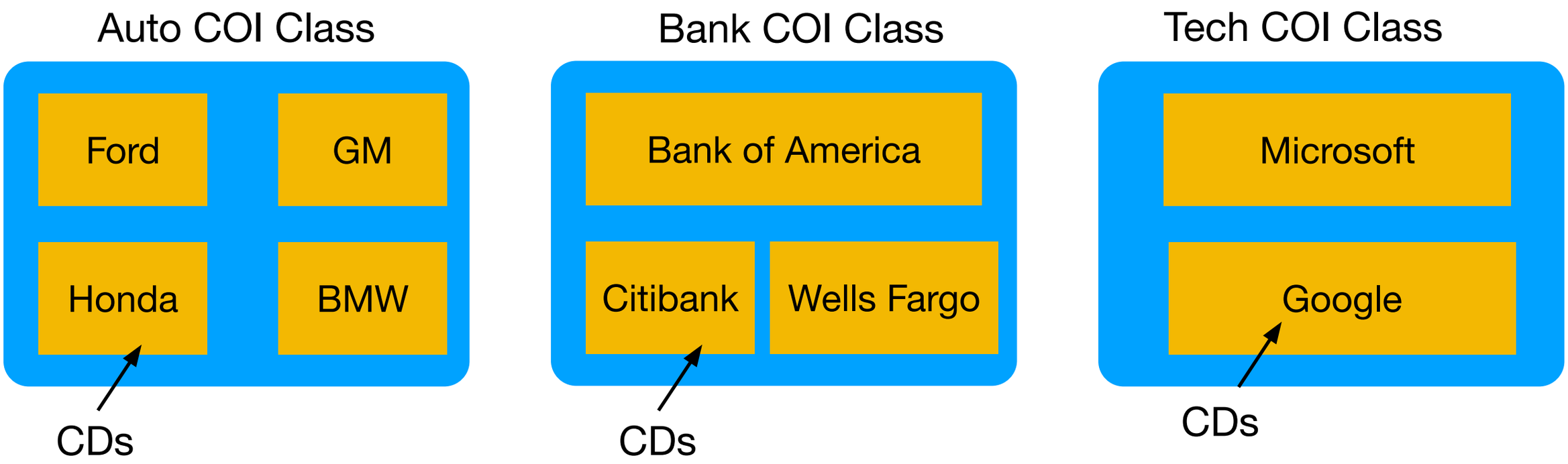
Hybrid Models

- The policies we saw so far are fairly general
- Specific commercial concern: the potential for conflicts of interest and inadvertent disclosure of information by a consultant or a contractor
- Example: A lawyer who specializes in product liability and consults for American Airlines. It could be a breach of confidentiality for her to consult also for United Airlines.
- **Why?**
- How about a simultaneous contract with Starbucks?

The Chinese Wall Policy

- Refers equally to confidentiality and integrity
- Levels of abstraction:
  - Objects (e.g., files): information about only one company
  - Company Dataset (CD): all objects concerning a particular company
  - Conflict of Interest (COI) classes: cluster the groups of objects for competing companies

Examples on COI Classes



- **Policy**: A subject S may access CDs from any company as long as S has never accessed CDs from a different company in the COI class.
- **Example**:
  - if you access a file from Ford, you subsequently will be blocked from accessing any files from GM, Honda, or BMW
  - you are free to access files from companies in any other conflict class (e.g., Citibank)

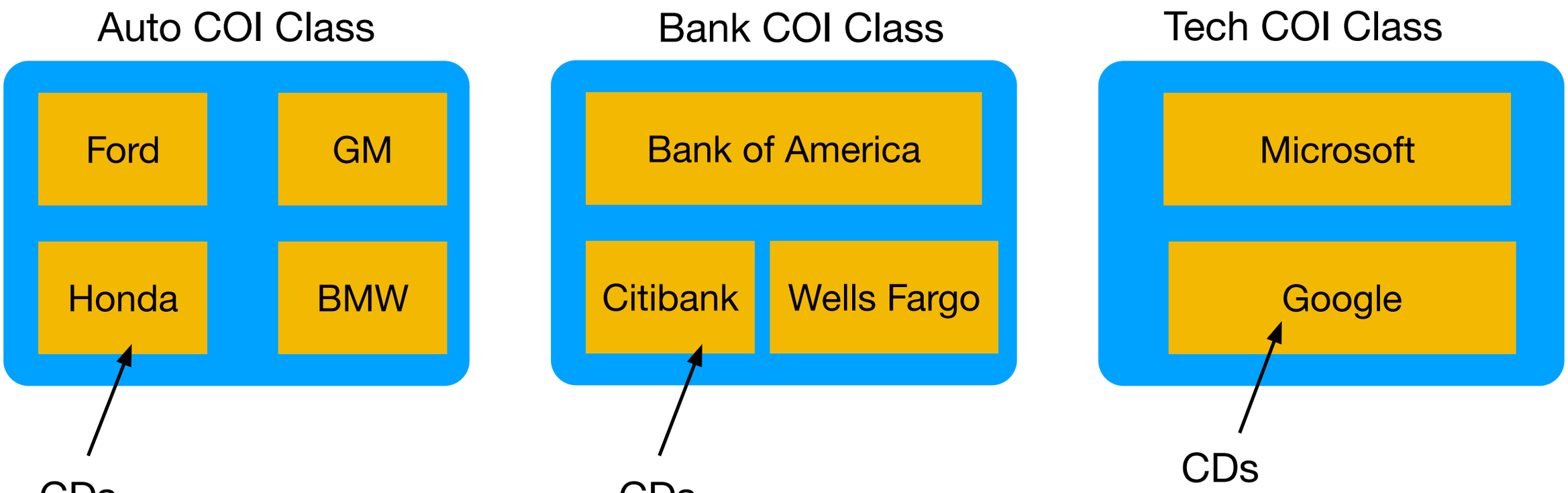
Chinese Wall Formally

- (Chinese Wall) Simple Security Rule: *S* can read *O* iff either a), b), or c) is true:
  - a) ∃ *O´: S* has accessed *O´* and *CD*(*O´*) = *CD*(*O*)
  - b) ∀ objects *O'*, *O´* ∈ *PR*(*S*) ⇒ *COI*(*O´*) ≠ *COI*(*O*)

PR(S): the set of objects that S has read

  - c) *O* is a sanitized object (ready for public consumption)
  - **Note:** Initially PR(S) = ∅, initial read request is assumed to be granted
- (Chinese Wall) *-property: *S* may write *O* iff both a) and b) hold:
  - a) The (Chinese Wall) simple security condition permits *S* to read *O*
  - b) ∀ unsanitized objects *O´*, *S* can read *O´* ⇒ *CD*(*O´*) = *CD*(*O*)

Chinese Wall Policy Violation Check



- PR(S1) = ∅
- PR(S2) = {BMW, Citibank}
- S1.read(GM): Yes
- S2.read(Microsoft): Yes
- S1.read(Ford): No
- S1.read(GM): Yes
- S2.read(Google): Yes
- S1.write(Ford): No
- S2.read(Honda): No
- S2.write(Bank of America): No

Lecture Summary

- CIA: w.r.t. security and integrity policies
- BLP: confidentiality focus
- Biba: integrity focus (BLP upside down)
- Clark-Wilson: transactions and consistency
- Chinese Wall: confidentiality = integrity