Security Policies

Security Definitions

Below are give some (relatively) formal definitions[1]:

- a ***security policy*** is a statement that partitions the states of a system into a set of *authorized*, or secure, states and a set of *unauthorized*, or non-secure, states.
- a ***secure system*** is a system that starts in an authorized state and cannot enter an unauthorized state.
- a ***breach of security*** occurs when a system enters an unauthorized state.
- a ***security mechanism*** is an entity or procedure that enforces some part of the security policy.
- a ***security model*** is a model that represents a particular policy or set of policies.

[1]Matt Bishop: Computer Security

The Military Security Policy (1)

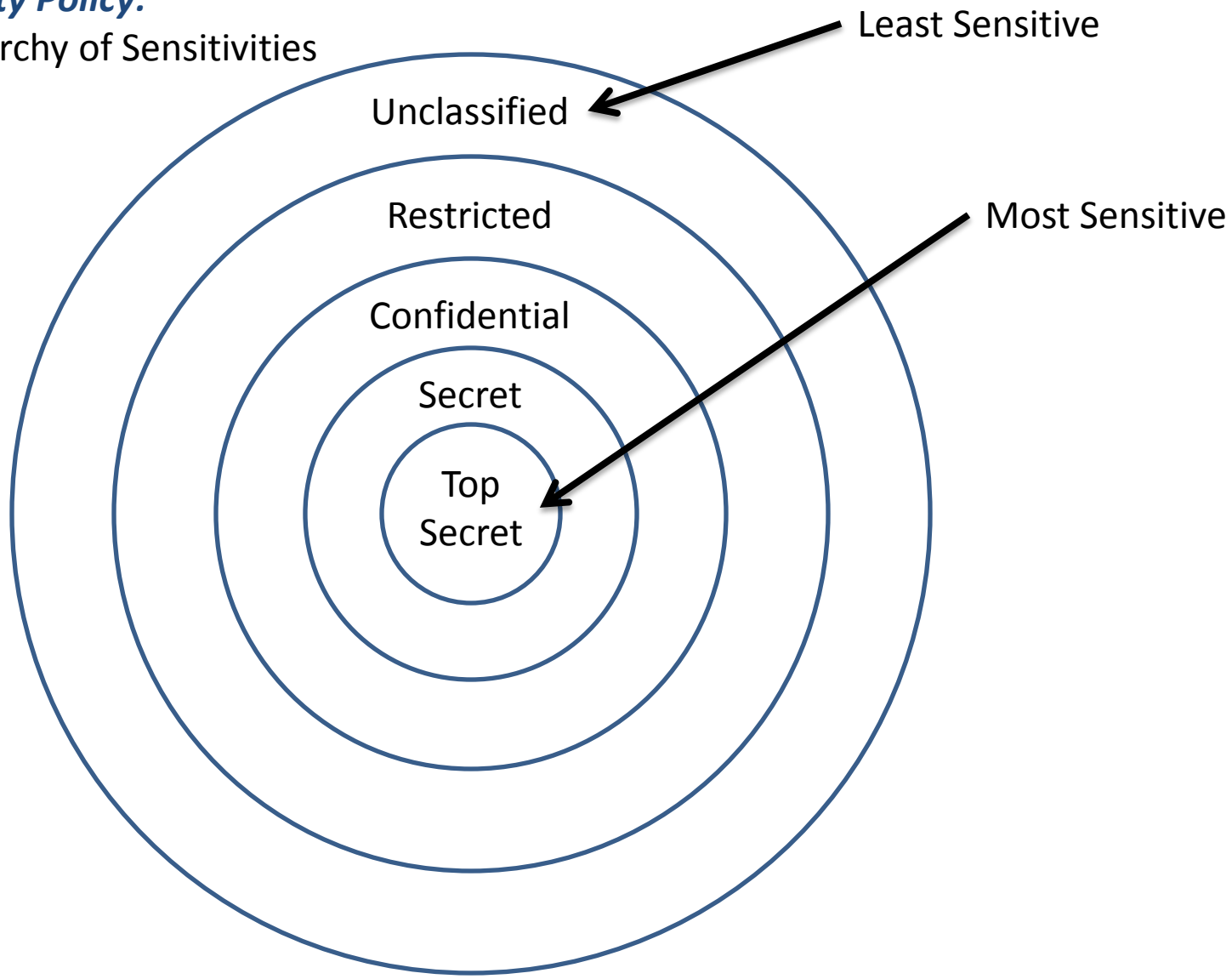
- the **Military Security Policy** is based on protecting classified information with respect to *confidentiality*.
- each piece of information is ***rank***ed at a particular *sensitivity* level:
  - unclassified
  - restricted
  - confidential
  - secret
  - top secret
- each piece of information may be associated with one (or more) projects, called ***compartments***.
- The combination **<rank; compartments>** is called the **classification** or **class** of a piece of information.

The Military Security Policy (2)

- a person has a **clearance** to access information up to a certain level of sensitivity.
- The clearance of a person has the same form as the classification of a piece of information:
  <rank; compartments>
- the **need-to-know** rule (principle of least privilege) means that individuals shall only have access to those data that they need in order to perform their jobs.
- the use of compartments helps to enforce the need-to-know rule.
- the user may ***not*** alter classifications, i.e. the policy requires Mandatory Access Control (MAC).

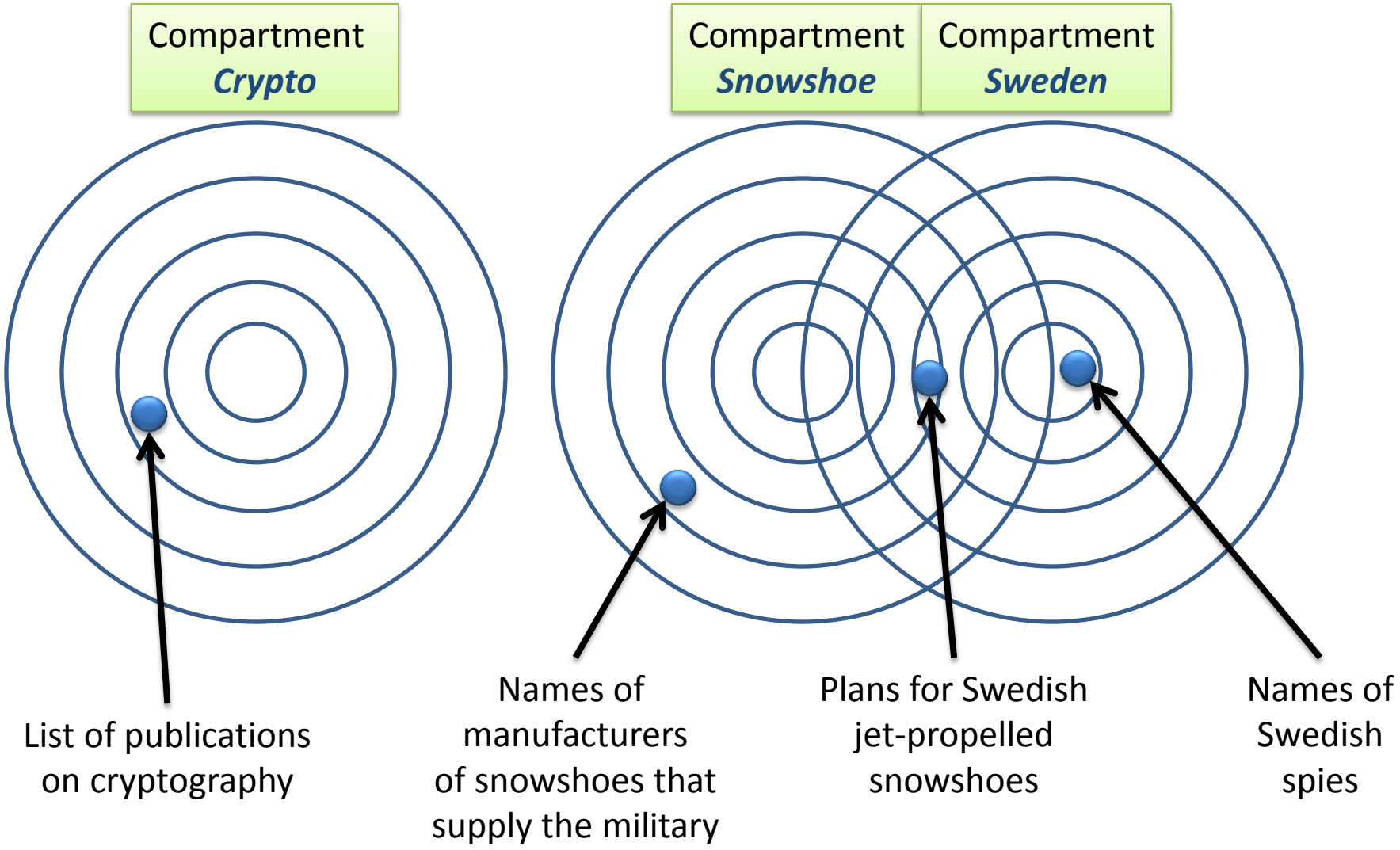
Military Security Policy:

Hierarchy of Sensitivities



Note: Here an object has a single rank, but may belong to several compartments.

Plans for Swedish jet-propelled snowshoes = <Confidential; {Snowshoe,Sweden}>



Commercial Security Policies (1)

- commercial security policies generally have a broader scope than the military security policy.
- they may address issues such as industrial espionage, conflicts of interest and rules for how activities must be performed within a company. Also they ***extend*** the scope to *integrity* and *availability*.
- they are normally less formal. There is no formalized notion of clearance and consequently are the rules for allowing access less regularized.
- the degrees of sensitivity are normally (but variants exists):
  - public
  - proprietary
  - internal

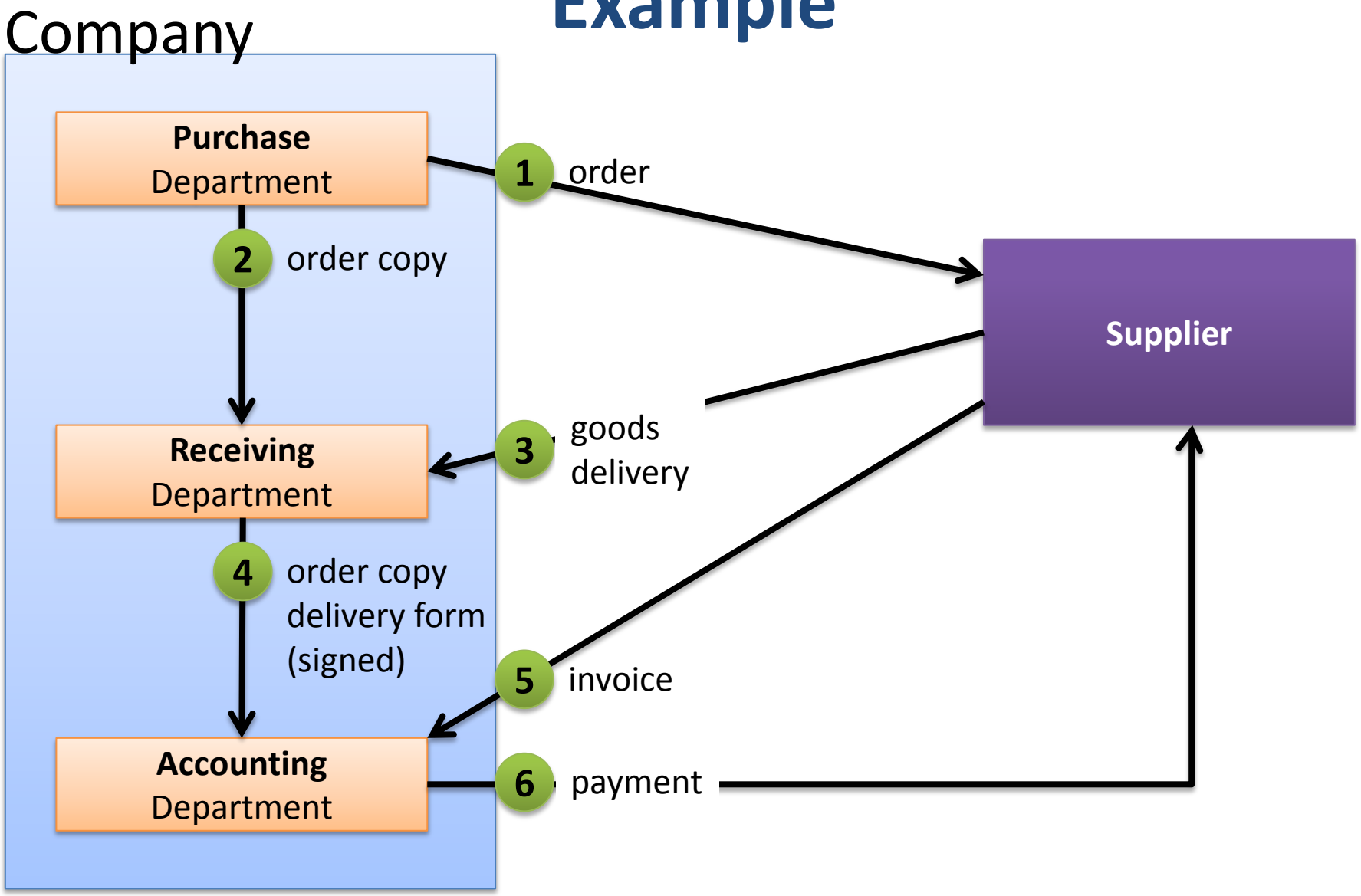
Commercial Security Policies (2)

- the **Clark-Wilson security policy:**
  - proposes a policy for *well-formed transactions*, which gives rules for the logistic process within the company, in terms of which steps must be performed by which person with a specified authority and in which order. Thus it addresses the *integrity aspect*.
- the Clark-Wilson security policy is defined in terms of access triples:

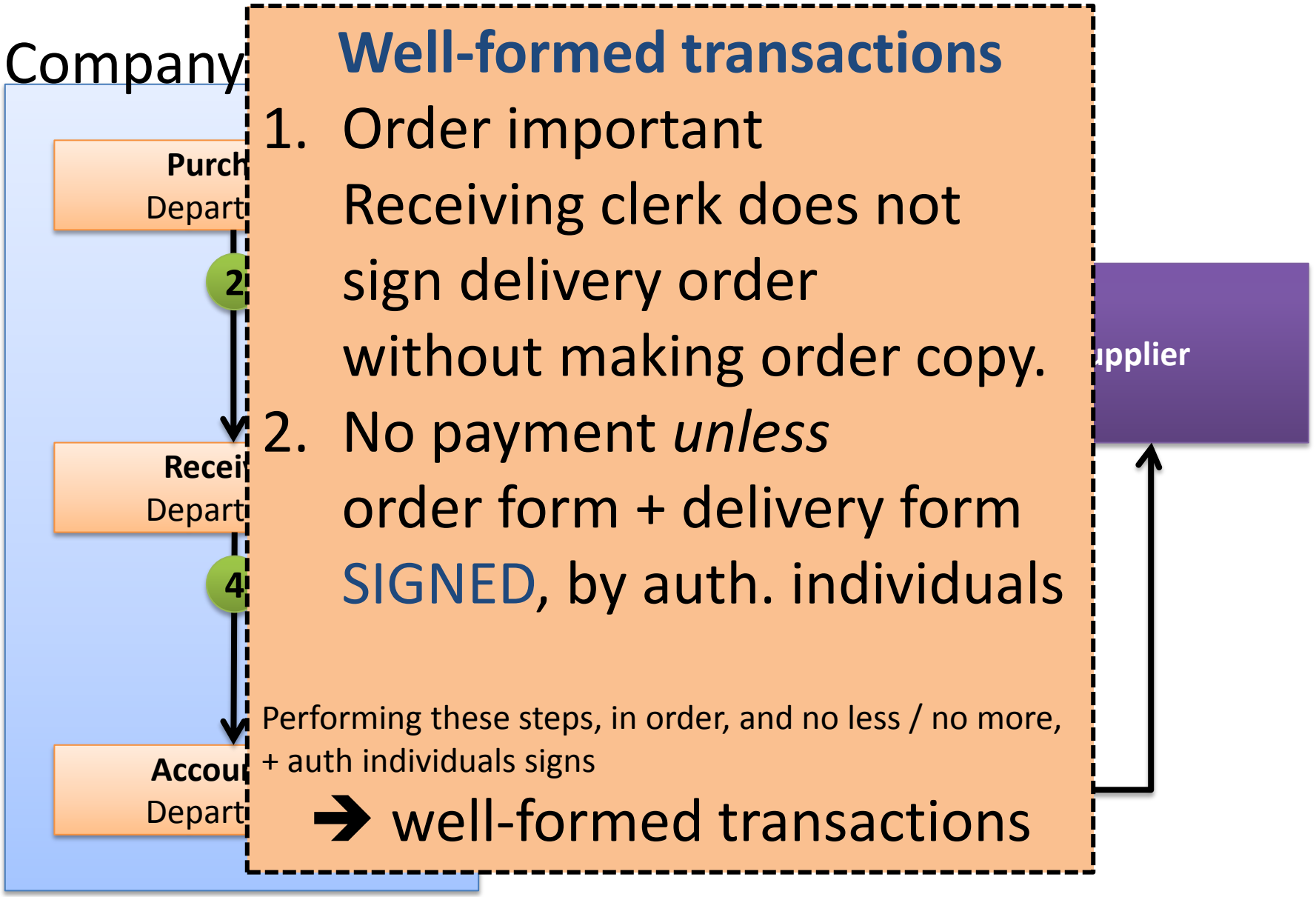
  <UserID; TP; {CDI_i,CDI_k, ...}>,

  which stands for
  - **User ID**entification,
  - **T**ransformation **P**rocedure and
  - **C**onstrained **D**ata **I**tems resp.

Clark-Wilson security policy Example



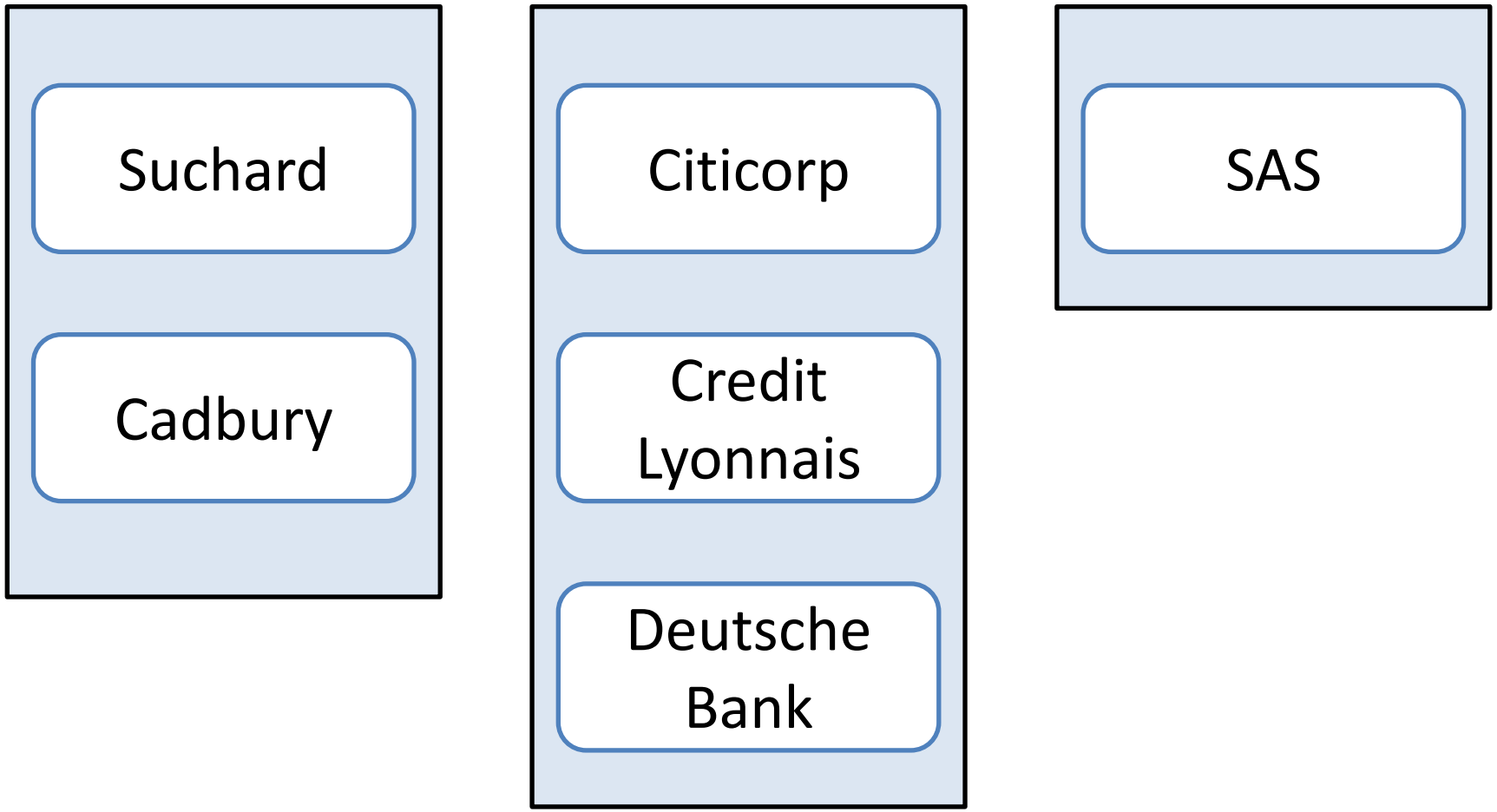
Clark-Wilson security policy



Commercial Security Policies (3)

- **Lee, Nash and Poland** suggested an addition to the *Clark-Wilson* policy that involves *separation of duty*. The aim is to prevent abuse that can arise when the same person performs too many related actions in a company.
- the **Chinese Wall policy** [by Brewer and Nash] enforces rules that prevents flow of information between companies that may have conflicting interests, e.g. competing.
  - the policy is defined in terms of three primitives:
    - objects,
    - company groups, and
    - conflict classes.
  - and the same employee may not access information from different companies in the same conflict class. Thus it *addresses confidentiality*.

Chinese Wall Policy Example



Bell-La Padula Security Model Overview

- The BLP is a formalization of the Military Security Model (described a mathematical notation).
- The BLP model is a formal description of the allowable paths of information flow in a secure system.
- The BLP defines security requirements for systems that concurrently handles data at different sensitivity levels.
- The BLP addresses confidentiality.

Bell-La Padula Security Model Formalism

- System is described as a set of subjects **S** and objects **O**
  - For each o∈**O**, there is a security class L(o) [classification]
  - For each s∈**S**, there is a security class L(s) [clearance]

Bell-La Padula Security Model Properties

- **Simple Security Property**:
  A subject *s* may have *read* access to an object *o* only if L(*o*)≤L(*s*).

- ***-Property**:
  A subject *s* who has *read* access to an object *o* may have *write* access to an object *p* only if L(*o*)≤L(*p*).

Bell-La Padula Security Model Extension with categories

Adding the *need-to-know* property with categories (projects).

Each subject s have a security clearance, l_s, and the need-to-know to access a number of categories, c_s.
CMP: Eve: <TS,{snowshoes, sweden}>

Definition:

The security level (L,C) dominates the security level (L',C') if and only if $L' \leq L$ and $C' \subseteq C$.

Bell-La Padula Security Model Properties revisited

- **Simple Security Property**:
  A subject *s* may have *read* access to an object *o* only if *s dominates o*.

- ***-Property**:
  A subject *s* who has *read* access to an object *o* may have *write* access to an object *p* only if *p dominates o*.

Bell- La Padula pros and cons

- **Advantages:**
  - A subject may not downgrade information
- **Problems:**
  - ”High” users can never talk to ”low” users
  - Addresses only confidentiality
  - Anyone can create an object with a higher classification
  - ”Float-up” (i.e. down-grade needed)
  - Does not address access control
  - Does not address covert channels
- **Principle of tranquility:**
  - Subjects and objects may not change their security level once they are instantiated