Configuring data flows in organizations and the Internet of Things for security and privacy

Luigi Logrippo

Abdelouadoud Stambouli

Université du Québec en Outaouais

luigi@uqo.ca

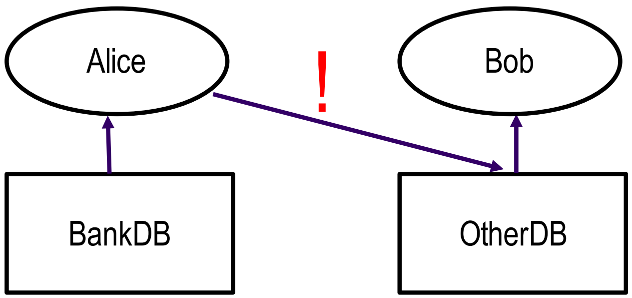
SECREV May 2020

Abstract

- Data flow control for data security and privacy is an important issue in organizations and in the Internet of things:
  - Where can data end up?
  - Who can change them?
- We show that fairly simple and efficient solutions exist, based on old ideas that have not yet been exploited to their full potential

Data flow control vs access control

- Access control:
  - Controls access of subjects to objects
- Data flow control:
  - Controls where data can end up in a network



- By access control:
  - Bob is authorized to read only from Other DB
  - Alice is authorized to both read BankDB and write on OtherDB
- But there is no data flow control, so Bob might get to know the data in BankDB although it has no direct access to it
  - see Trojan horse etc.

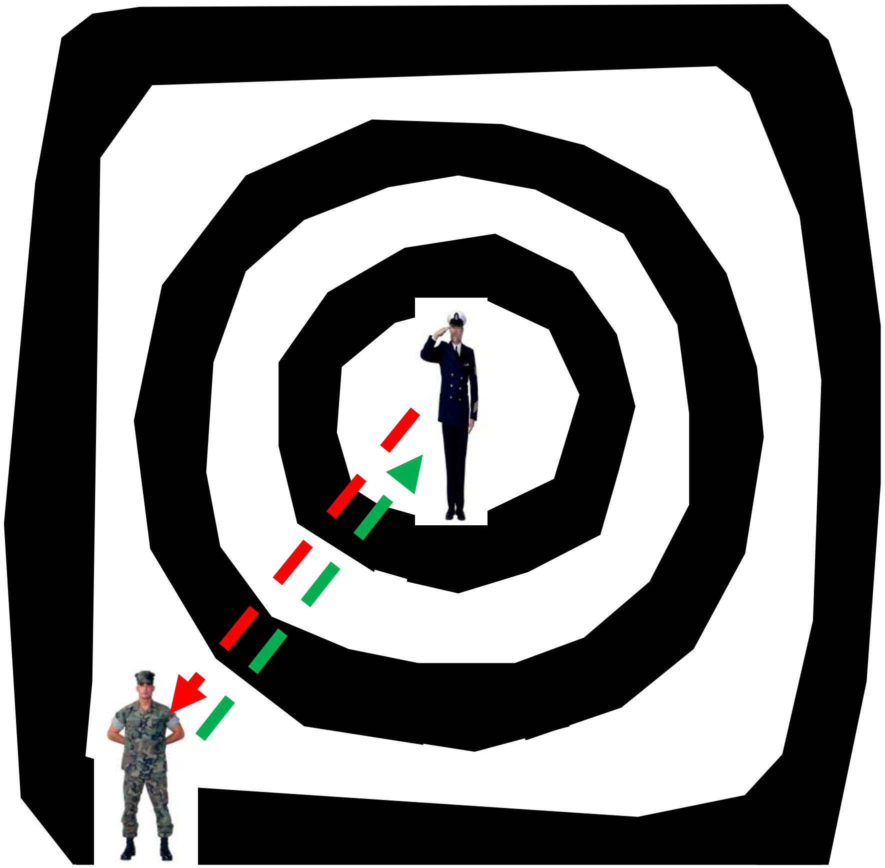
Existing access control methods

- The most used access control method today is RBAC, Role-Based Access Control, in its many variants
- But RBAC does not do data flow control
- It can be configured for data flow control,
  - but only if methods such as the one presented here are used
- Similarly for ABAC-XACML and many other data protection methods

Traditional remedy for flow control

- Label subjects and data in order to be able to express policies on who should know what
- Access control and flow control
- Many organizations use labeling methods
  - Banks, government, the military
    - Classify documents by levels of secrecy
    - Classify employees by their security clearance

Historical example



- Data flow
  - Can be unrestricted from low to high
  - Must be restricted from high to low

MAC

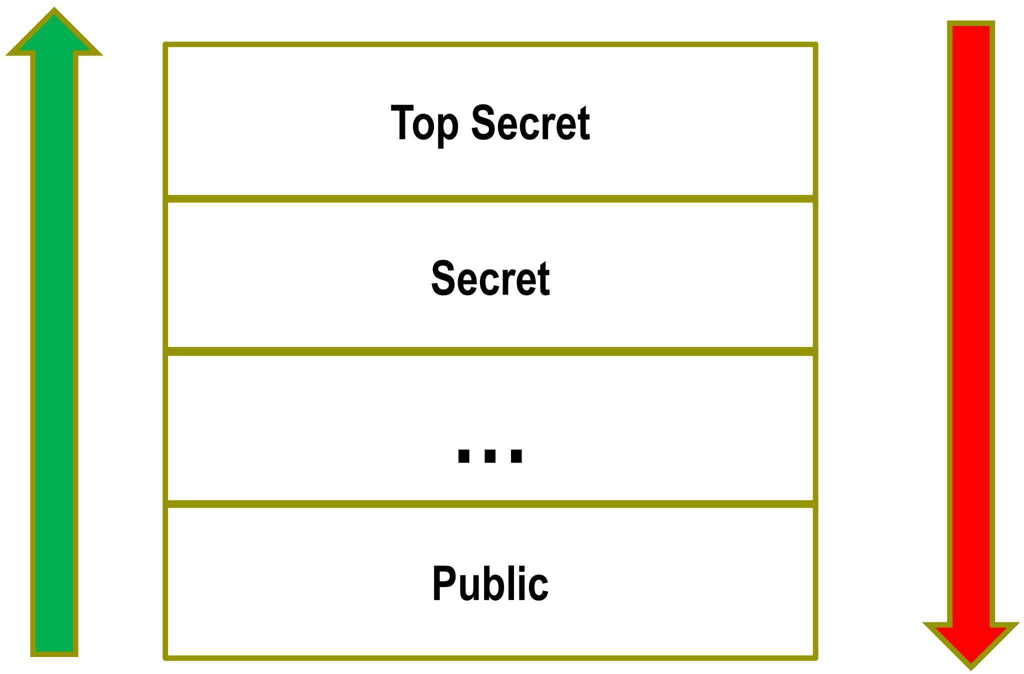
Mandatory Access Control data security models

- Subjects and objects are labelled
  - Subjects are labelled by the data that they can read
  - Objects are labelled by the data that they can contain
- There are label-based policies that determine
  - Which subjects can read which objects
  - Which subjects can write on which objects
- Simultaneously guarantees access control and flow control

The Bell-La Padula model

Read down

Write up



No read up

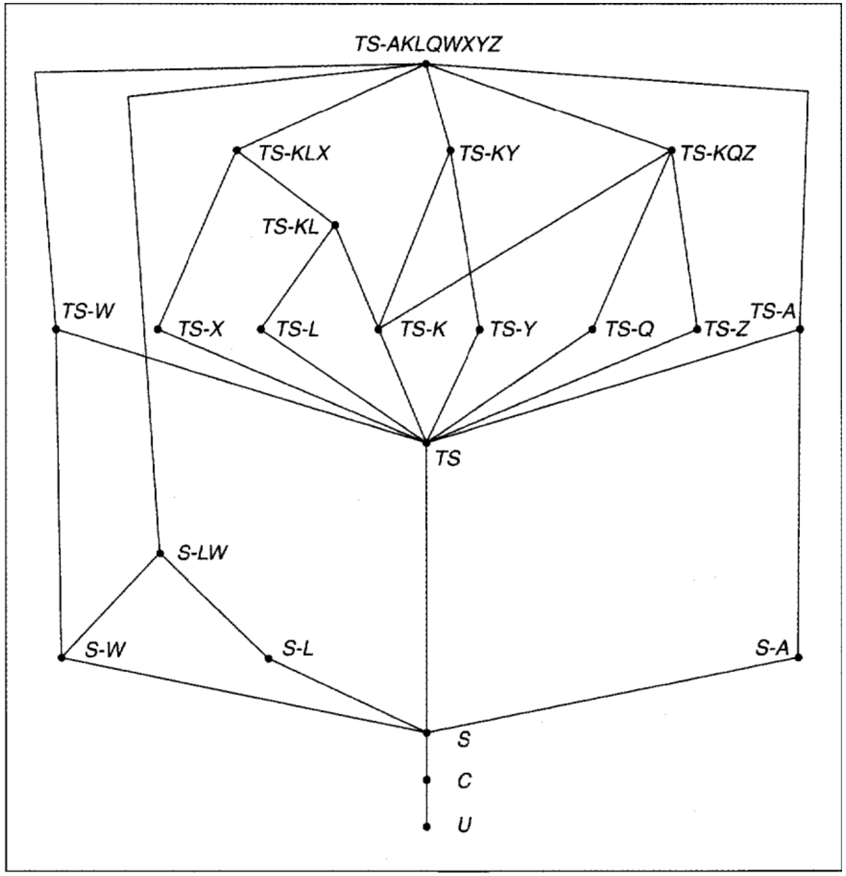
No write down

Data can flow only upwards

We generalize this model

An established generalization: Lattice model
(Denning 1974)

- Data can move only upwards in the lattice



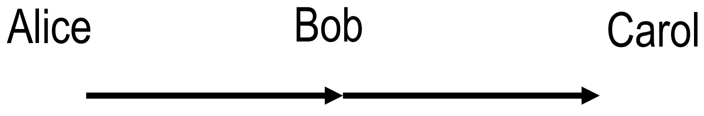
Source: Sandhu, 1993

Success and critique of the lattice model

- **All** security data flow models in the literature today are based on the lattice model
- But: it may oblige to include inexistent or impossible entities in order to have a lattice structure
  - It may be necessary to postulate the existence of
    - an entity that can know everything and
    - another that can know nothing
    - possibly, entities that contradict security constraints
      - E.g. if no entity is supposed to know both Bank1 and Bank2 data, it is still necessary to assume the existence of an entity that knows both!
- Data networks are very rarely designed as lattices!
  - The lattice model has found limited application in practice
- **Happily, a *simpler and more general* model is possible by using the concept of *partial order instead of lattice***

Reflexivity and Transitivity of data flows

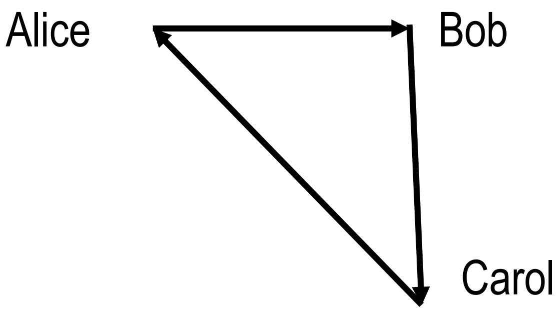
- Reflexivity: every entity knows the data that it contains
- Transitivity: if Alice talks to Bob, and Bob talks to Carol, then I should assume that whatever Alice knows, can also be known by Carol
    - Happily, it is possible also to assume that entities limit their conversations to certain subjects, this is very useful for refinements



A data flow digraph

Equivalence classes of knowledge

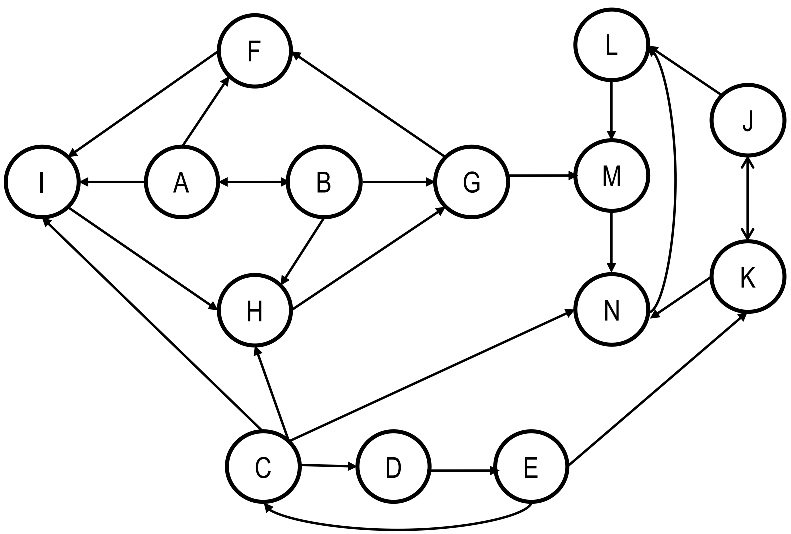
- Under the transitivity assumption, any strongly connected data flow digraph describe a set of entities that can have the same knowledge
- We call them ***components***



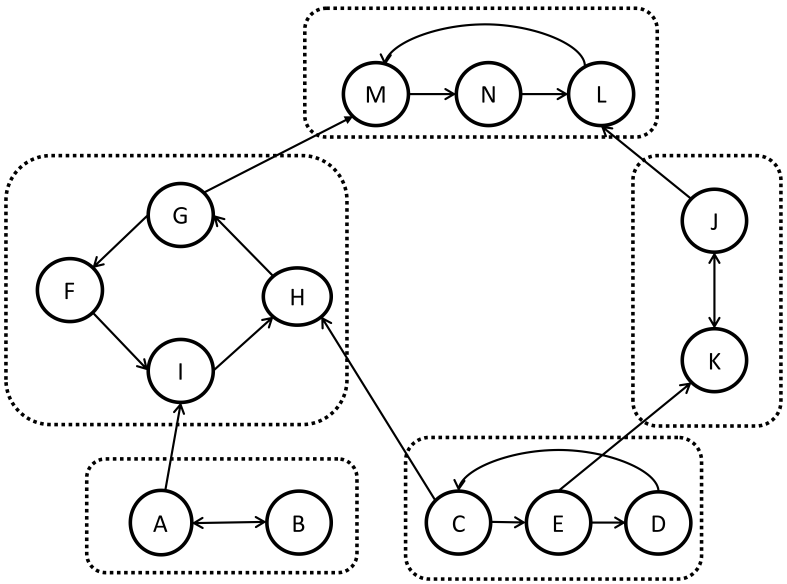
- We see three entities that by transitivity can know the same data
  - If some data are sent to any of them, then all of them can know the data
- From this point of view, they can be identified, they are a single ***component***
  - Note that some or all data flows above can be bidirectional, the entities can still know the same data

Finding the components in any digraph

- A digraph

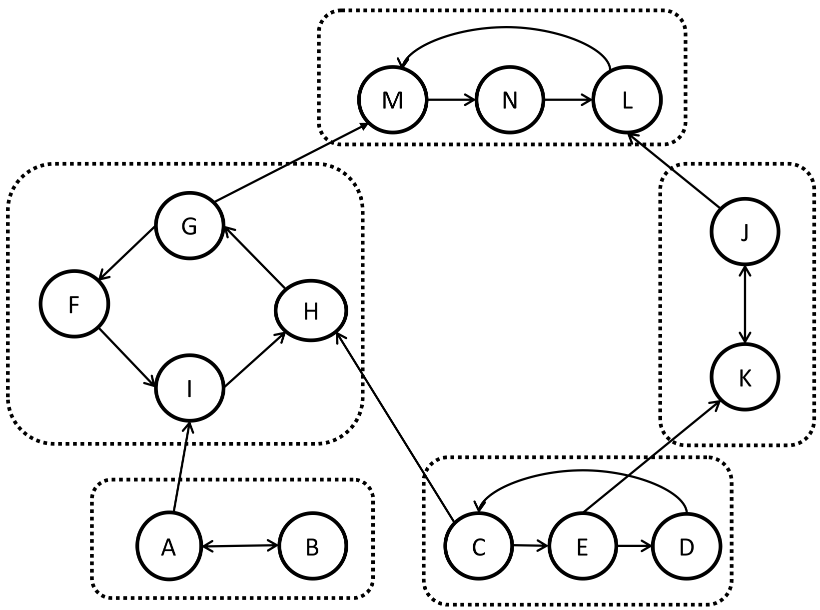


- Identifying its components

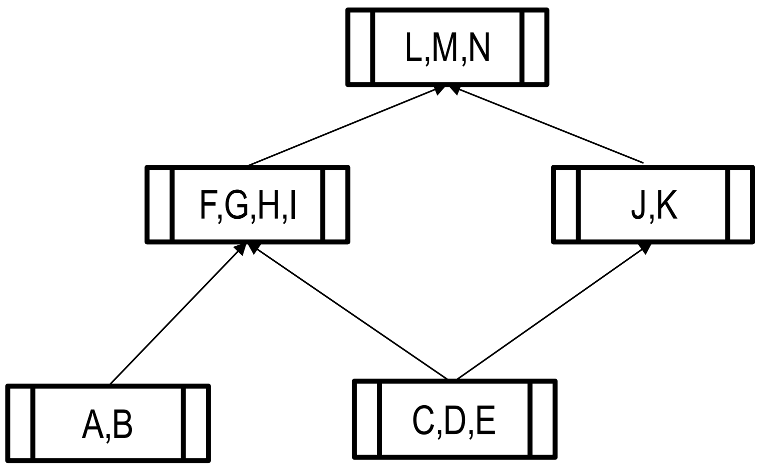


Finding a partial order

- The previous digraph

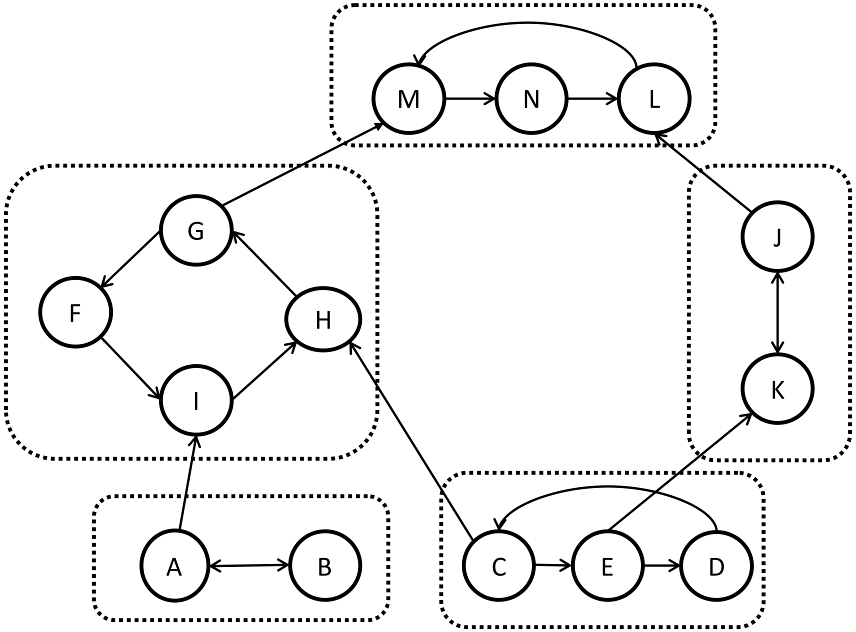


- Concise view of the partial order



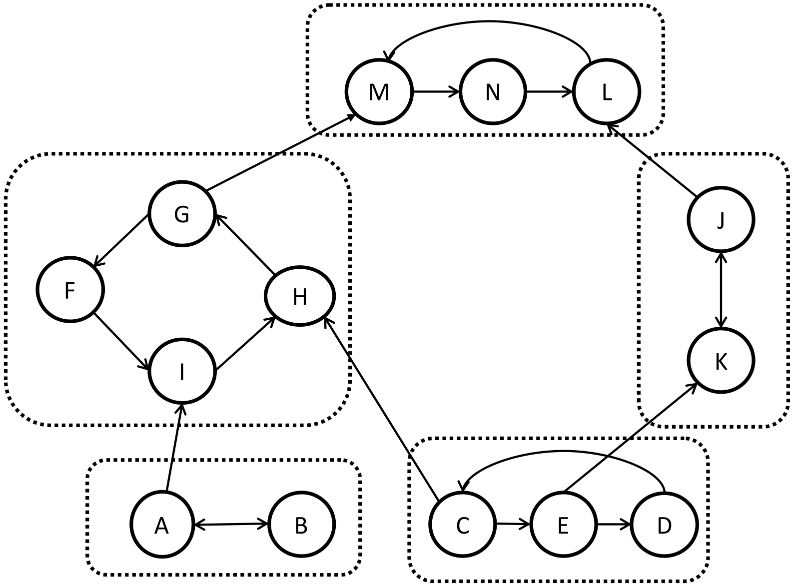
Graph theory results

- For any digraph, it is possible to find its partial order of components
  - any digraph will have sources and sinks
- There are efficient algorithms to do this



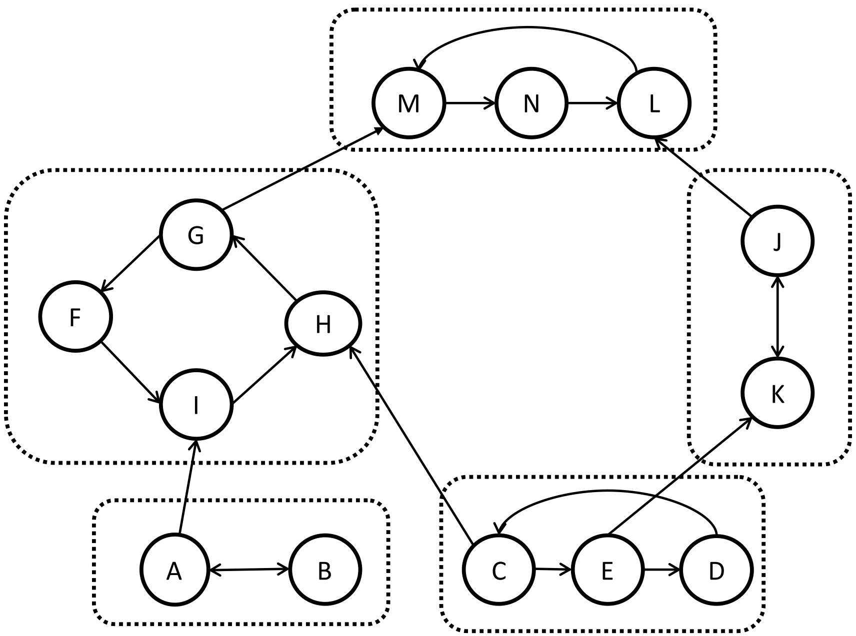
Secrecy and integrity in partial orders

- Secrecy: an entity's secrecy is characterized by the fact that its data cannot flow to other entities
  - Entities towards the top of the partial order have highest secrecy
  - As desired, since usually they are executive-level data
- Integrity: an entity's integrity is characterized by the fact that extraneous data cannot be injected in it
  - Entities towards the bottom of the partial order have highest integrity
  - As desired, since they are usually data gathered from the field



Application to the Internet of Things

- The entities at the bottom can be the detectors
  - Measuring equipment, cash registers…
  - They have highest integrity
- The entities at the top can be the final users of the data and of all the processing that has been done on them
  - They have highest security



Using labels

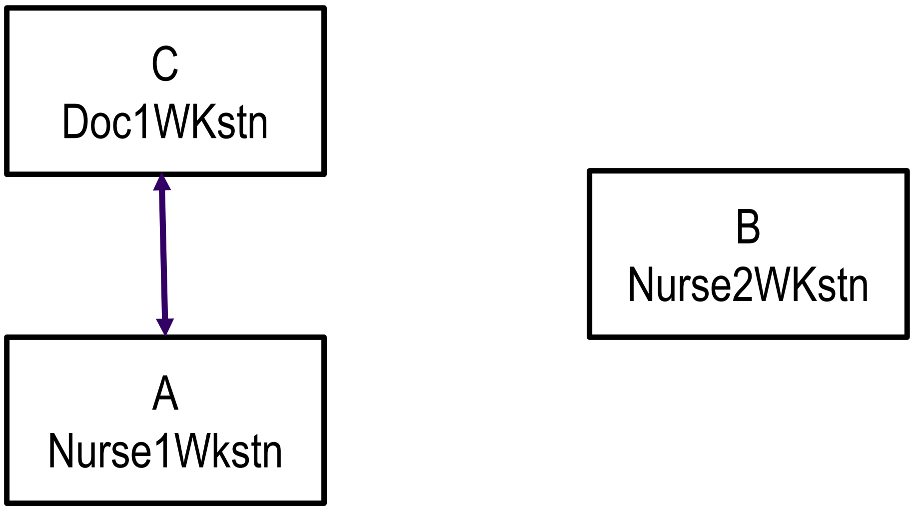
- Security labels are usually partial orders
  - E.g. in traditional security:
    - Secret > Classified > Public
- By assigning labels to entities, entities are given their position in partial orders

Hospital devices example (1)

- In this example, we construct a hospital IoT network, where each entity is labeled with the type of data it can contain
  - E.g.**Nurse1Wkstn{SamPress, BobPulse, Stats1}** means that this workstation can only contain data about the blood pressure of Sam, the pulse of Bob and some statistics
  - E.g. **Nurse2Wkstn{SallyPulse,Stats2}** means that this workstation can only contain data about the pulse of Sally and some different statistics
  - No data flow is possible between the two, they have to be mutually disconnected

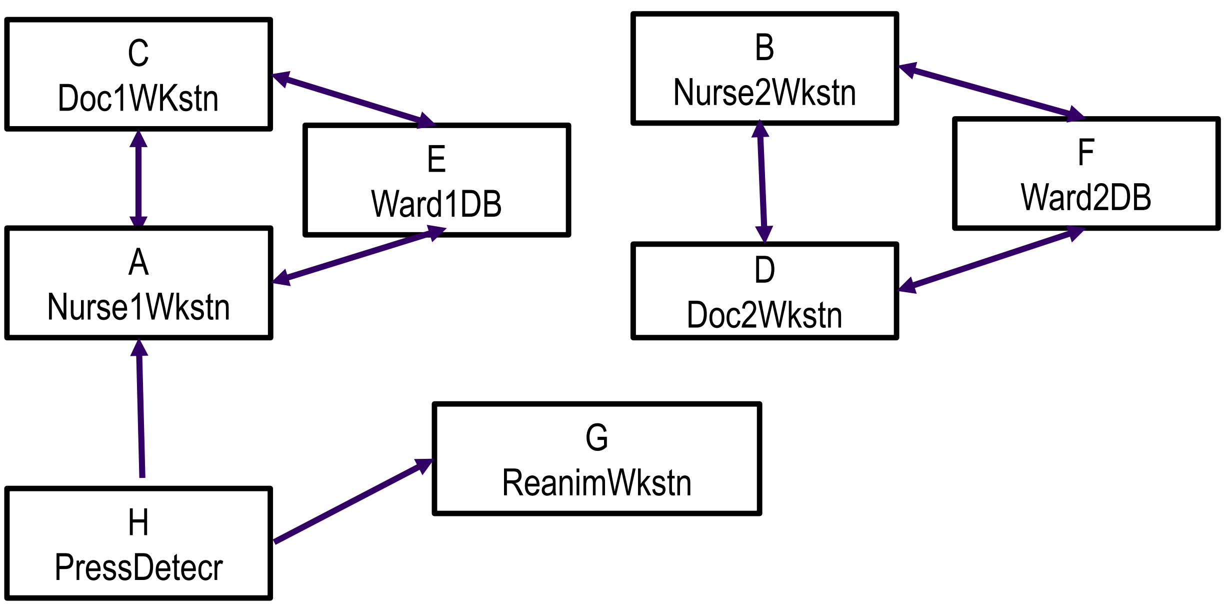
Hospital devices example (2)

- We now add an entity C, whose name and label are **Doc1Wkstn{SamPress, BobPulse, Stats1}**
- Since A and C have exactly the same label, we can establish a bidirectional flow between the two

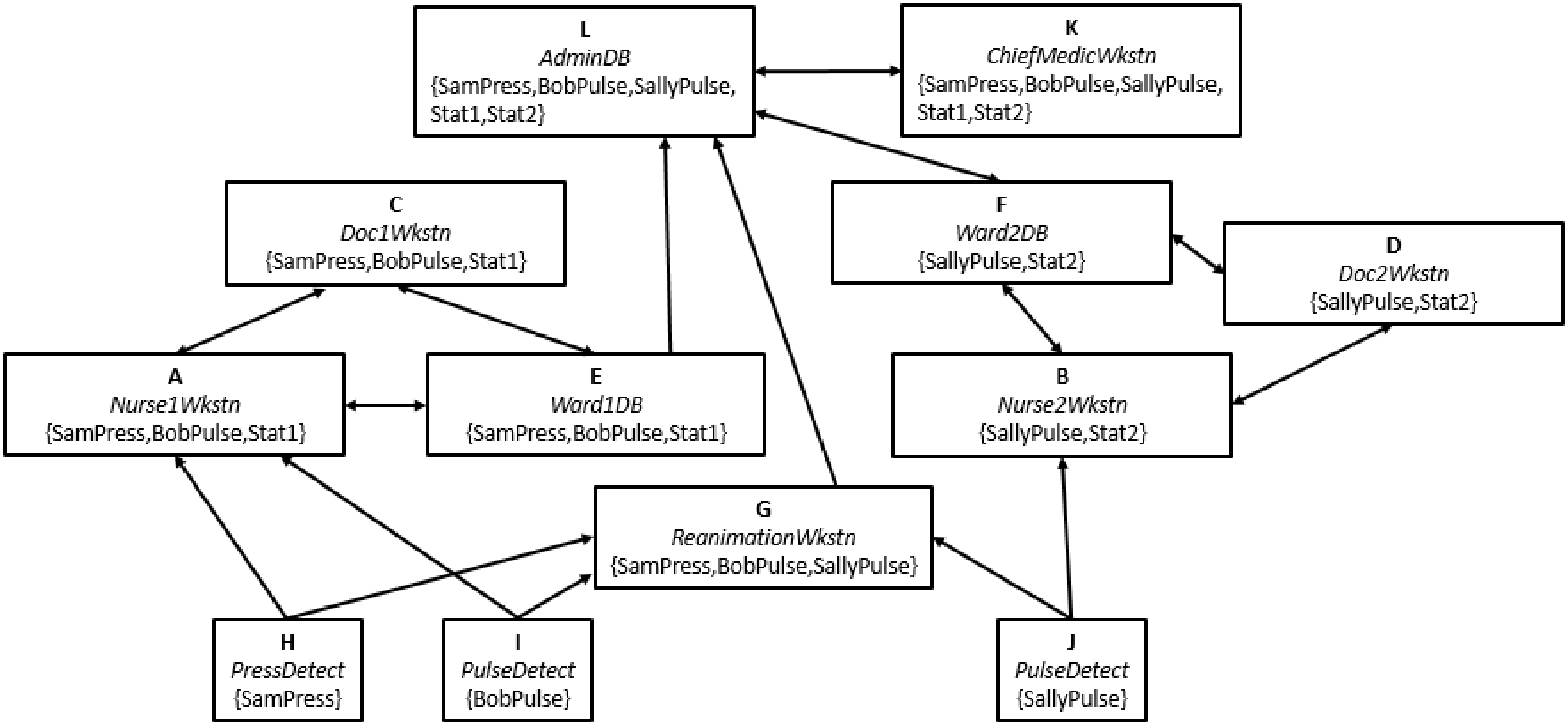


Hospital devices example (3)

- We keep going, creating new entities which various labels that create new data flows
  - Create(D) = Doc2Wkstn{SallyPulse,Stats2}
  - Create(E) = Ward1DB{SamPress, BobPulse, Stats1}
  - Create(F) = Ward2DB{SallyPulse,Stats2}
  - Create(G) = ReanimationWkstn{SamPress, BobPulse, SallyPulse}
  - Create(H) = PressDetect{SamPress}

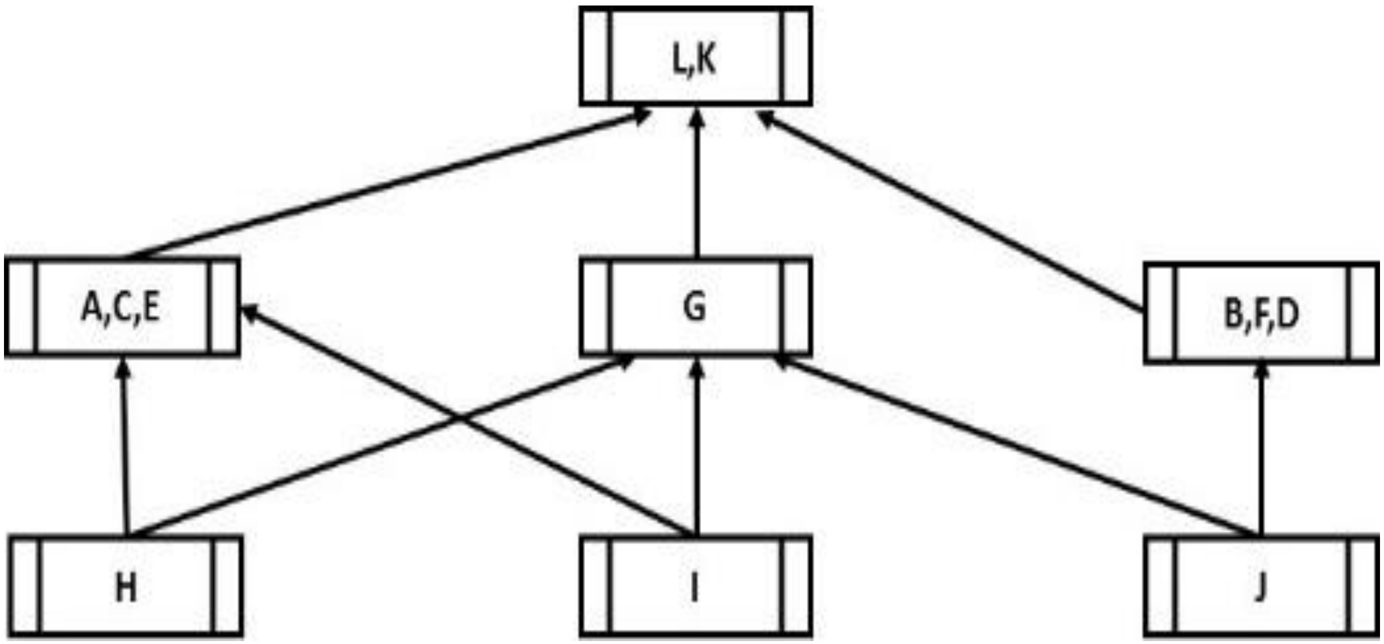


The full example



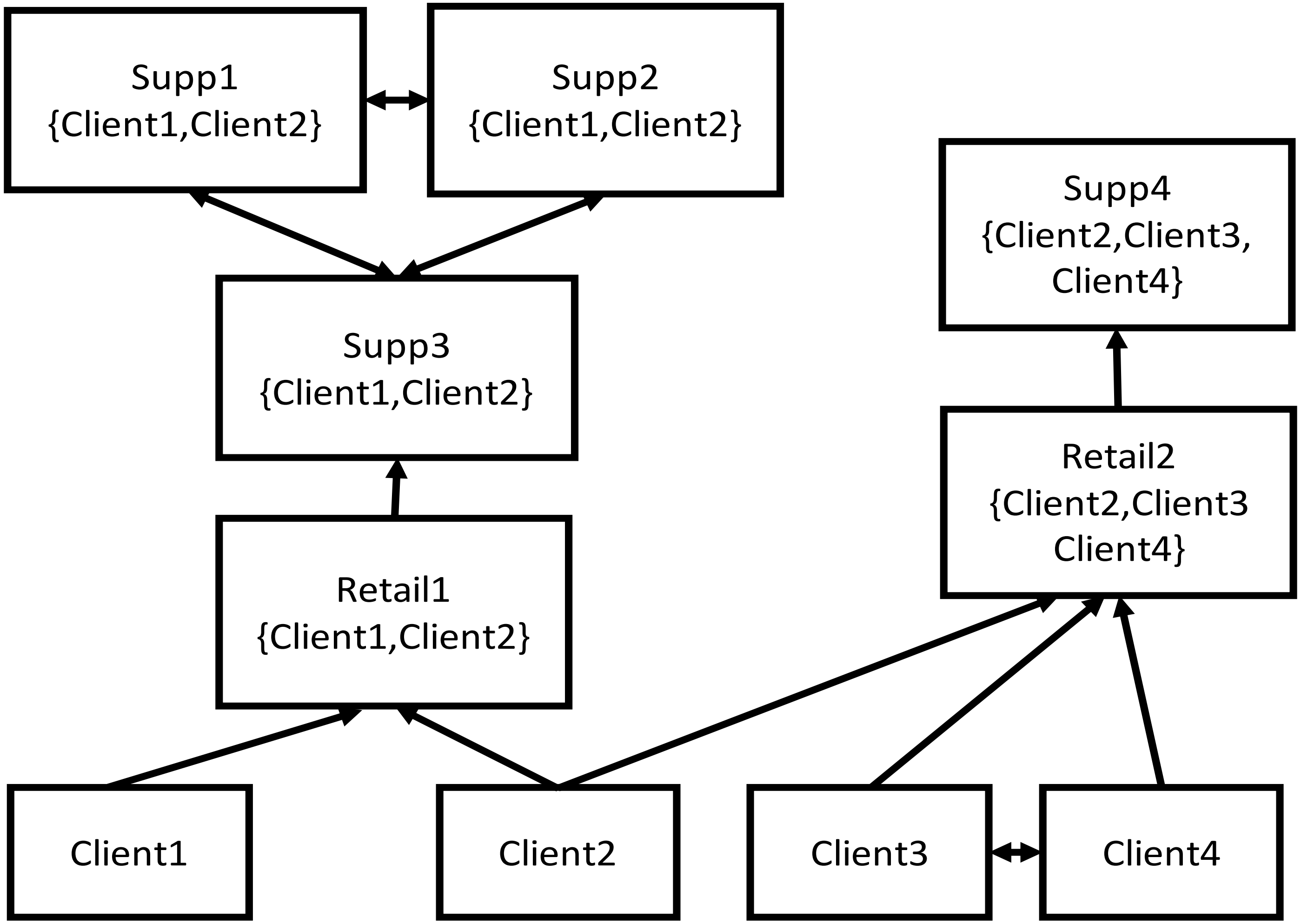
We will get to this structure independently of the order of creation of the entities

Its partial order of components

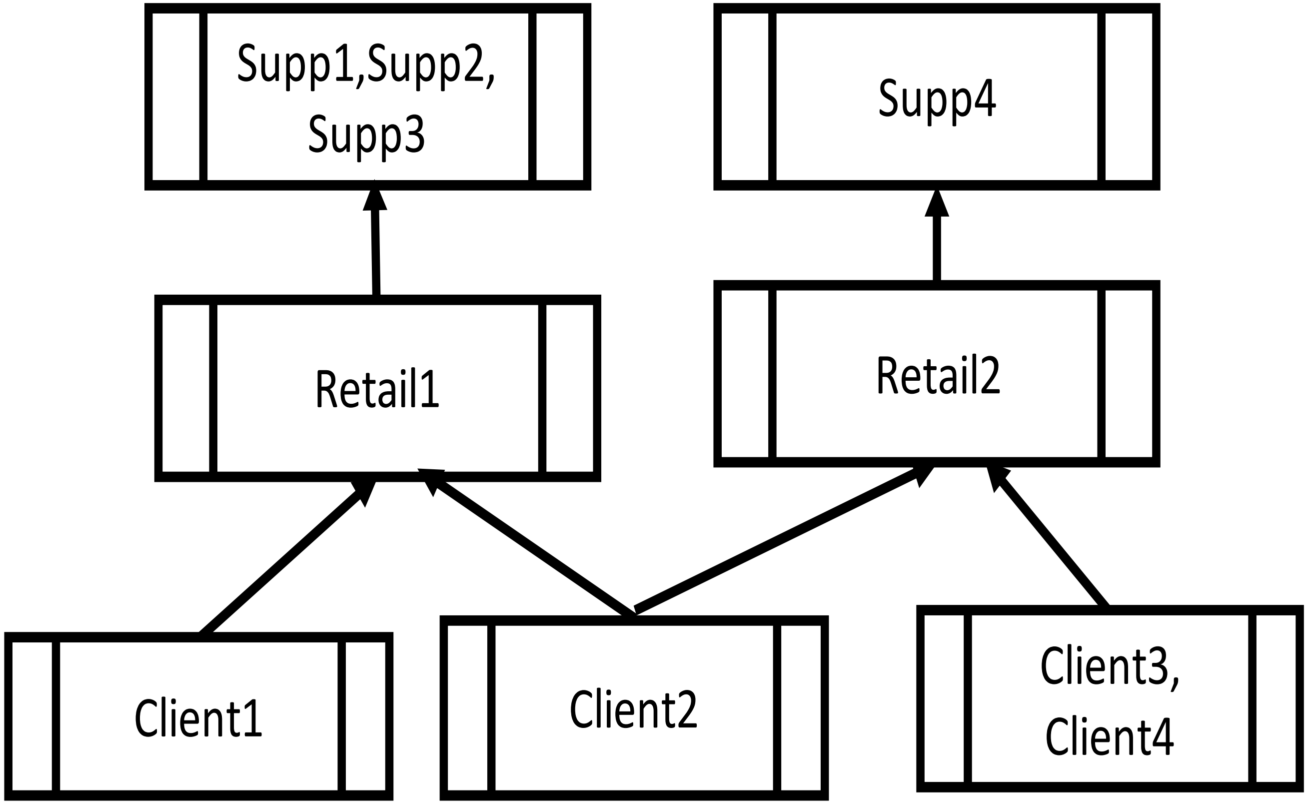


Secrecy grows together with knowledge as we move up
Integrity grows with basic knowledge as we move down

E-commerce example: orders data flow

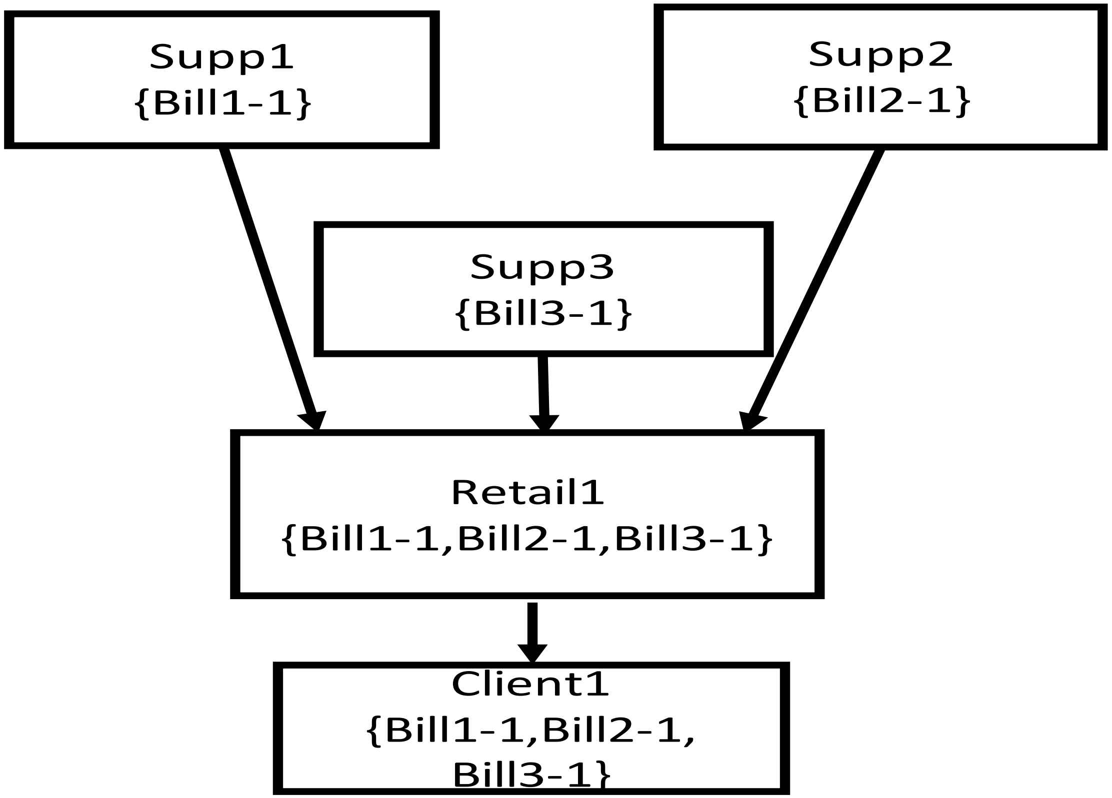


Partial order of components in e-commerce example



Another possible data flow

billing data flow for Client1



Coexisting data flows

- So, several data flows can coexist in a network
- Our method can handle them, by tagging data according to the data flow to which they belong
- Entities must be trusted to keep separate different flows

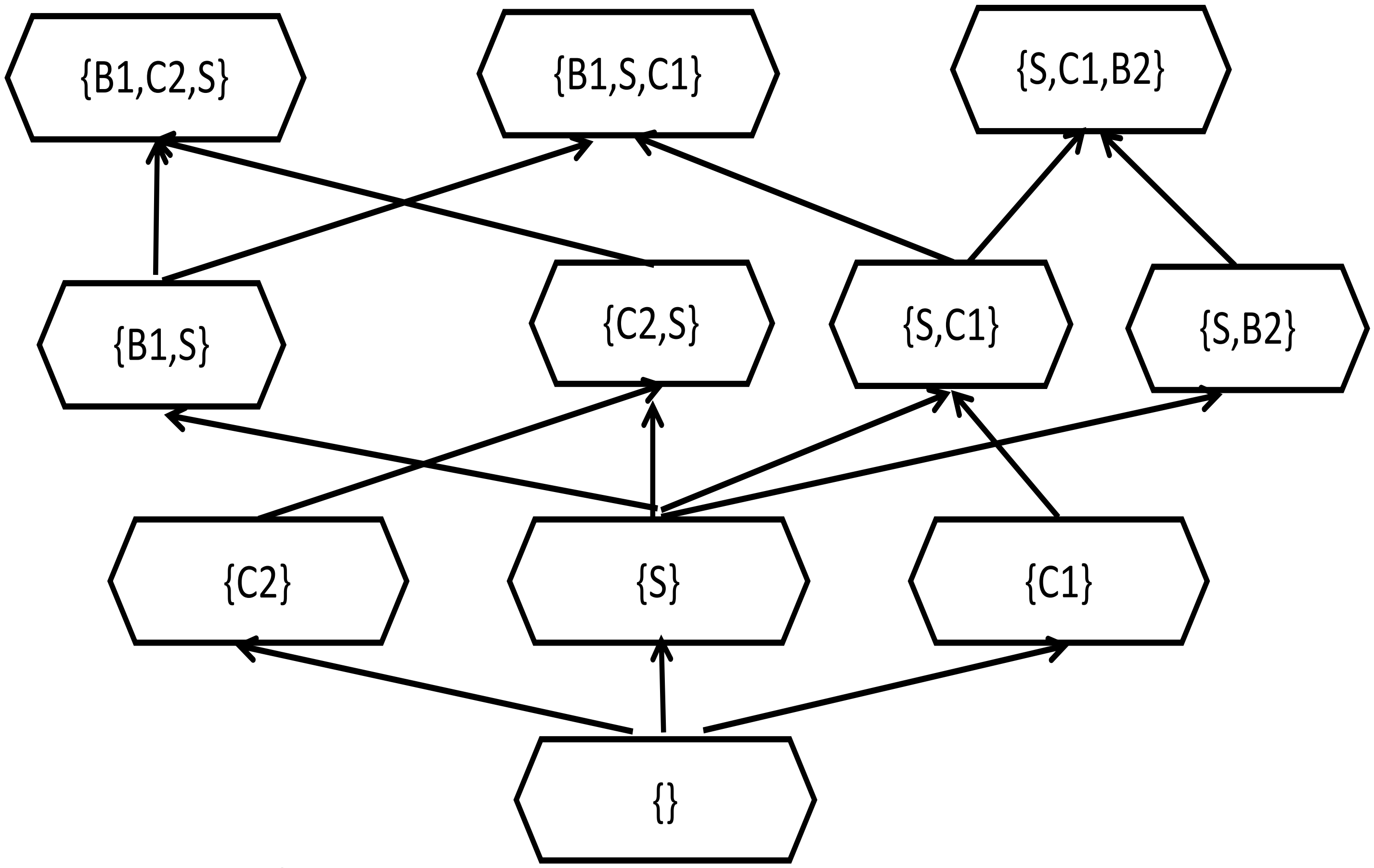
Complex label sets

- Complex labeling schemes can be invented, to express complex sets of constraints
- This is possible by relaxing the conventional constraint that labels be organized in lattices
  - We only need partial orders!
- Labeling methods have great expressive power, which so far has been little used in practice

Example: a business network

- Network requirements:
  - Two banks in conflict of interest, their data are labeled B1, B2
    - So labels containing {B1,B2} are not allowed
  - Two companies also in conflict, their data are labeled C1,C2
    - So labels containing {C1,C2} are not allowed
  - Also Bank2 is in conflict with Company2
    - So labels containing {B2,C2} are not allowed
  - The two banks need the data S from a server, so B1 and B2 will always have to appear with S

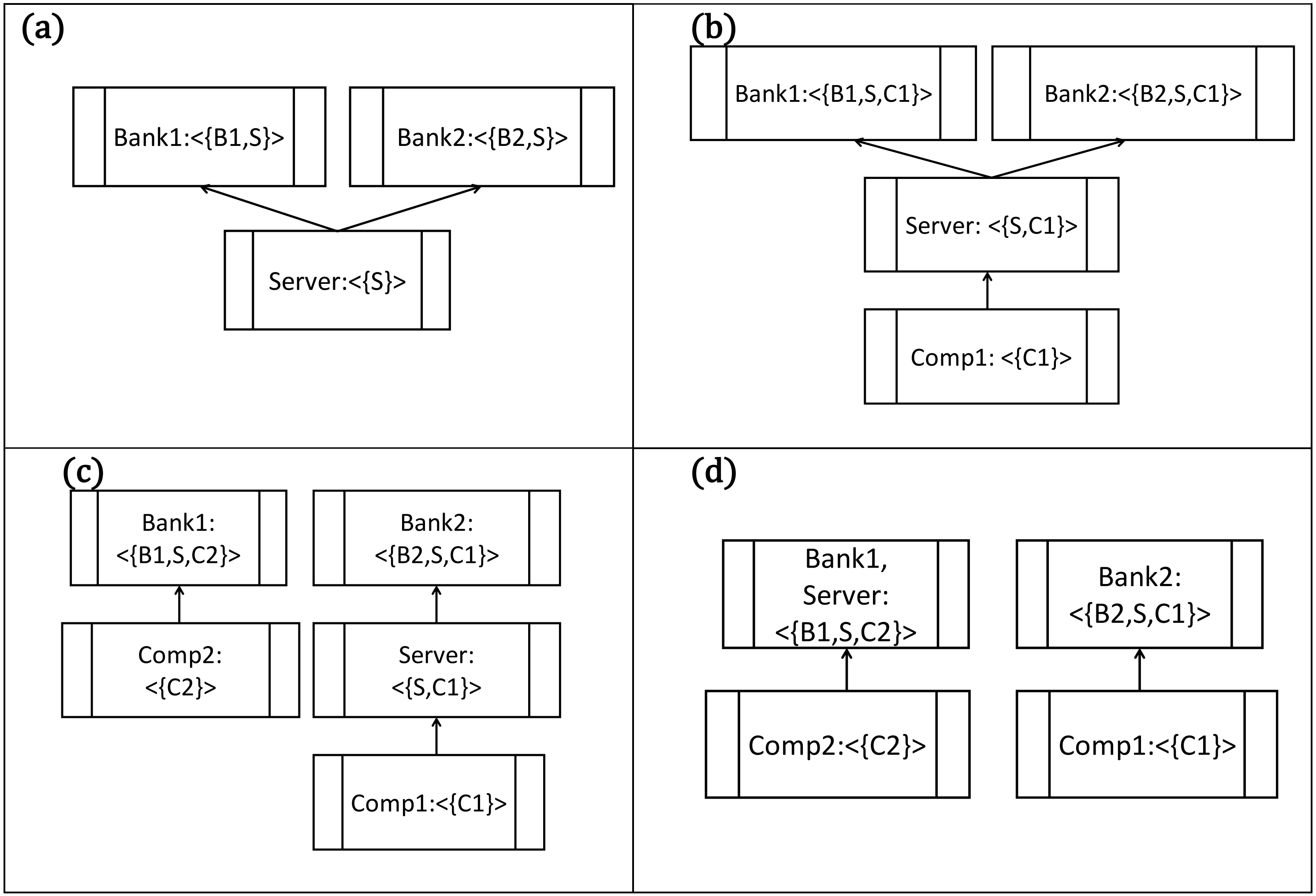
Partial ordered set of allowed labels



Note the absence of forbidden labels

Network re-configurations

- IoT systems should be able to continuously reconfigure
  - Data de-classification and other updates due to changing requirements
  - Entity creation, deletion
  - Channel creation, deletion
- Entities can be relabeled, but in order to maintain security requirements, the relabeling must follow the given partial order diagram



(a) Is the initial configuration; in (b) Company1 comes in; in (c) Company2 comes in and is associated with Bank1, which must lose its association with Company2, and so on.

How to implement this?

- By access control mechanisms
- By routing mechanisms
- By encryption, to implement secure channels
  - If data flow is from A to C through B, but B cannot read it, then we can say that the channel is only from A to C
- SE-Linux

Conclusions

- **Necessary and sufficient** conditions for data security in networks can be obtained by generalizing mandatory access control and lattice concepts
  - Use the concept of partial order instead
  - Entities of **high secrecy or high integrity** can be found in any network
- **Exactness:** By positioning entities in IoT networks according to the type of data they can hold, it is possible to configure data transfer channels so that **all and only** logically allowed flows are possible
  - Both **secrecy and integrity** are taken care of
- **Scalability:** Efficient algorithms exist, which makes the solution **scalable** and practical
- **Supporting methods:** Partial orders can be implemented in networks in many ways, according to the nature of each network.

Related work

- Although the literature in security and access control in the IoT is vast, there are few papers with solutions for data flow control in IoT networks
- Previous to us, they were all based on the lattice model
  - Which is needlessly restrictive
- Many papers on security in IoT do not provide specific solutions

For more information

- A draft paper with references can be found in https://www.site.uottawa.ca/~luigi/papers/20_Multilevel.pdf
- Authors will be pleased to hear from you:
  - luigi@uqo.ca