Access Control Models

Prof. Ravi Sandhu
Executive Director and Endowed Chair

January 25, 2013
&
February 1, 2013

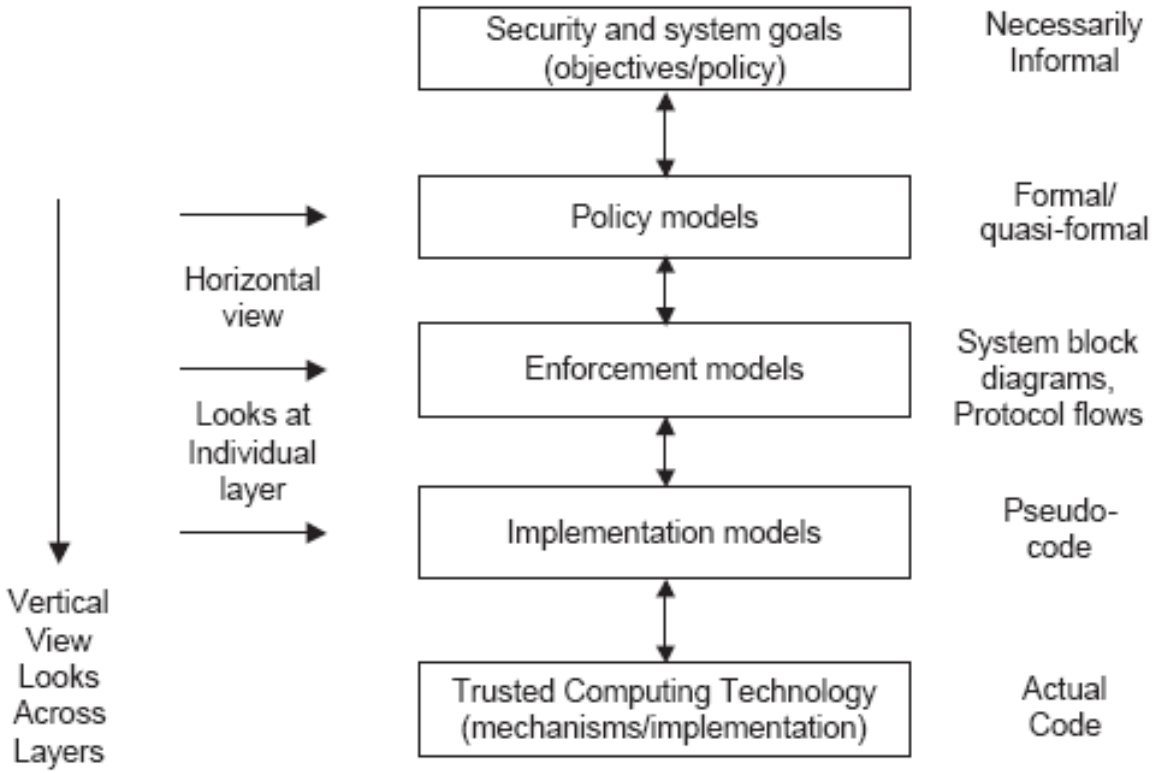
ravi.sandhu@utsa.edu
www.profsandhu.com

Foundational Security Assumptions

- Information needs to be protected
  - In motion
  - At rest
  - In use
- Absolute security is impossible and unnecessary
  - Trying to approximate absolute security is a bad strategy
  - "Good enough" security is feasible and meaningful
  - Better than "good enough" is bad
- Security is meaningless without application context
  - Cannot know we have "good enough" without this context
- Models and abstractions are all important
  - Without a conceptual framework it is hard to separate "what needs to be done" from "how we do it"

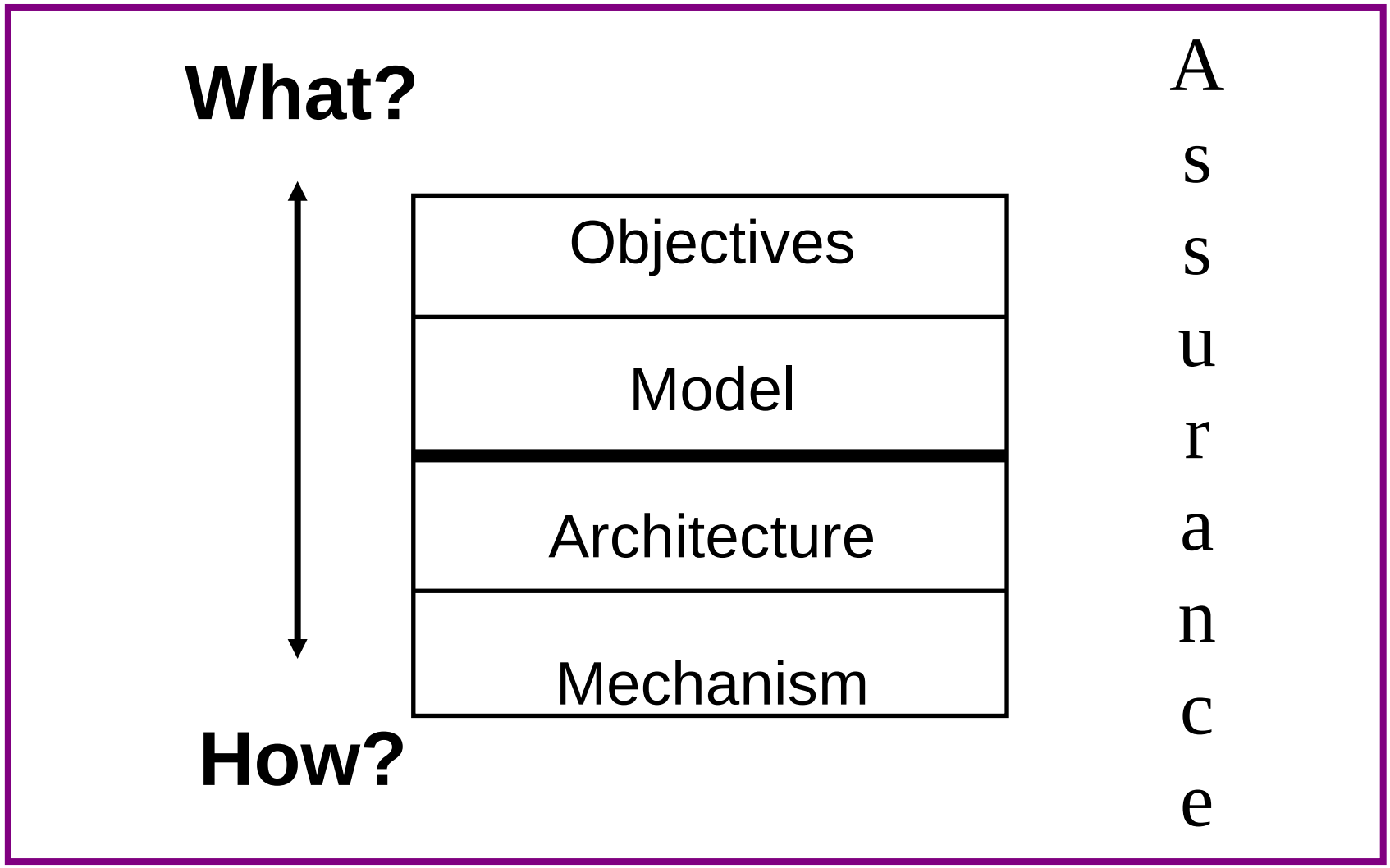
We are not very good at doing any of this

PEI Models



At the policy layer security models are essentially access control models

OM-AM (Older Version of PEI)



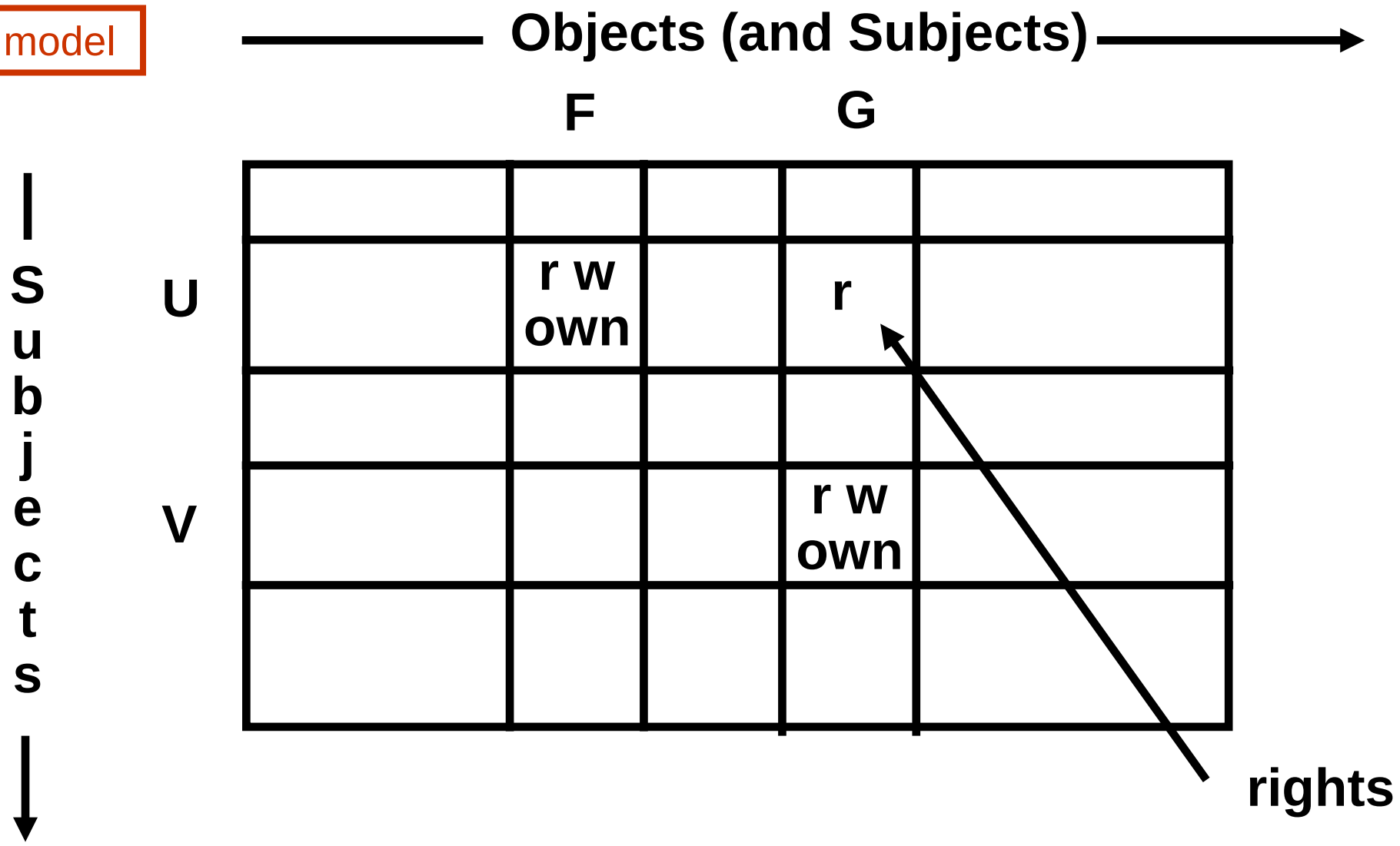
Access Control Models

- Discretionary Access Control (DAC)
  - Owner controls access but only to the original, not to copies
- Mandatory Access Control (MAC)
  Same as Lattice-Based Access Control (LBAC)
  - Access based on security labels
  - Labels propagate to copies
- Role-Based Access Control (RBAC)
  - Access based on roles
  - Can be configured to do DAC or MAC
  - Generalizes to Attribute-Based Access Control (ABAC)

Numerous other models but only 3 successes

DAC: ACCESS MATRIX MODEL

P model



ACCESS CONTROL LISTS (ACLs)

E model

F

U:r

U:w

U:own

G

U:r

V:r

V:w

V:own

each column of the access matrix is stored with the object corresponding to that column

CAPABILITY LISTS

E model

U **F/r, F/w, F/own, G/r**

V **G/r, G/w, G/own**

each row of the access matrix is stored with the subject corresponding to that row

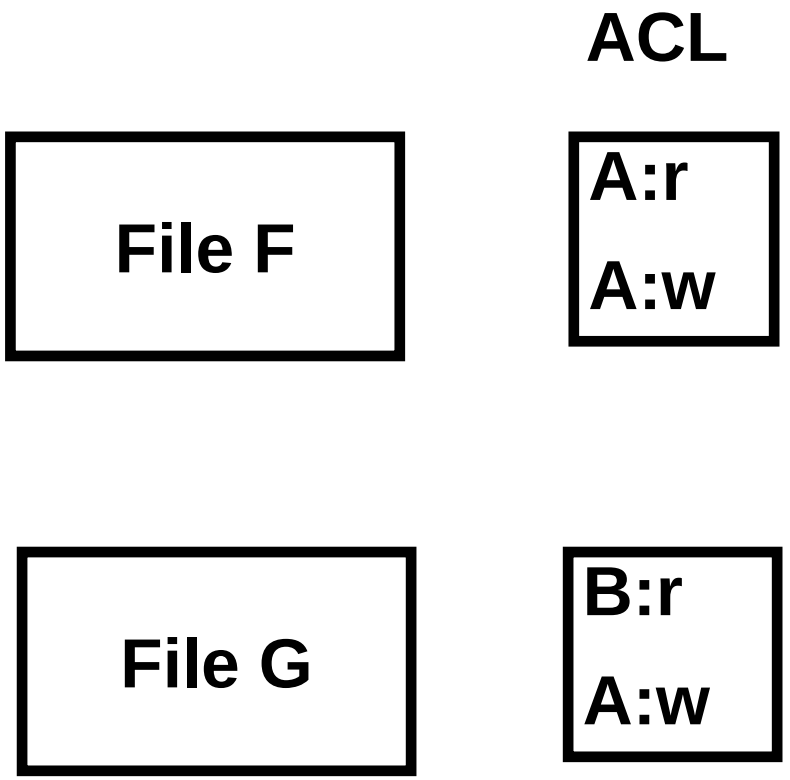
ACCESS CONTROL TRIPLES

E model

Subject	Access	Object
U	r	F
U	w	F
U	own	F
U	r	G
V	r	G
V	w	G
V	own	G

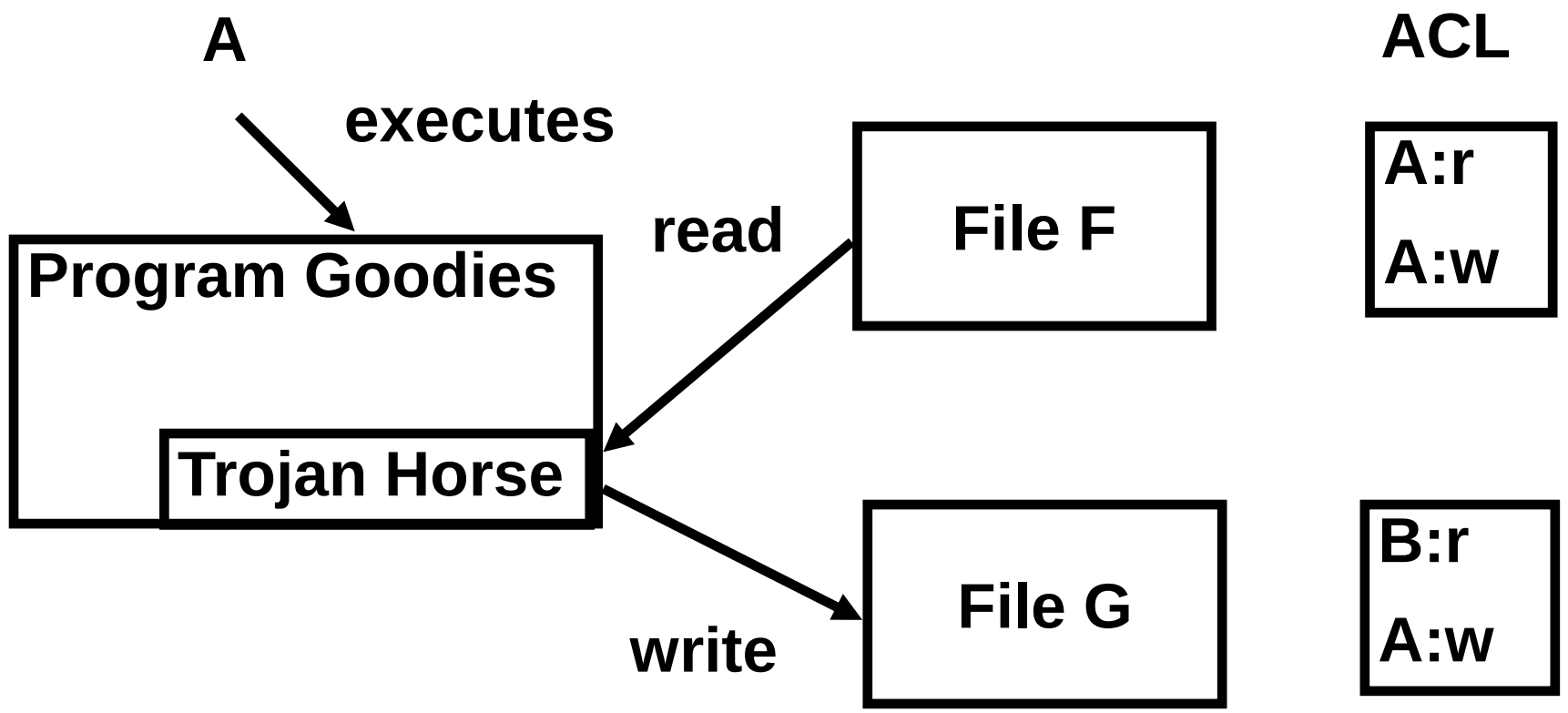
commonly used in relational database management systems

DAC: TROJAN HORSE EXAMPLE



B cannot read file F

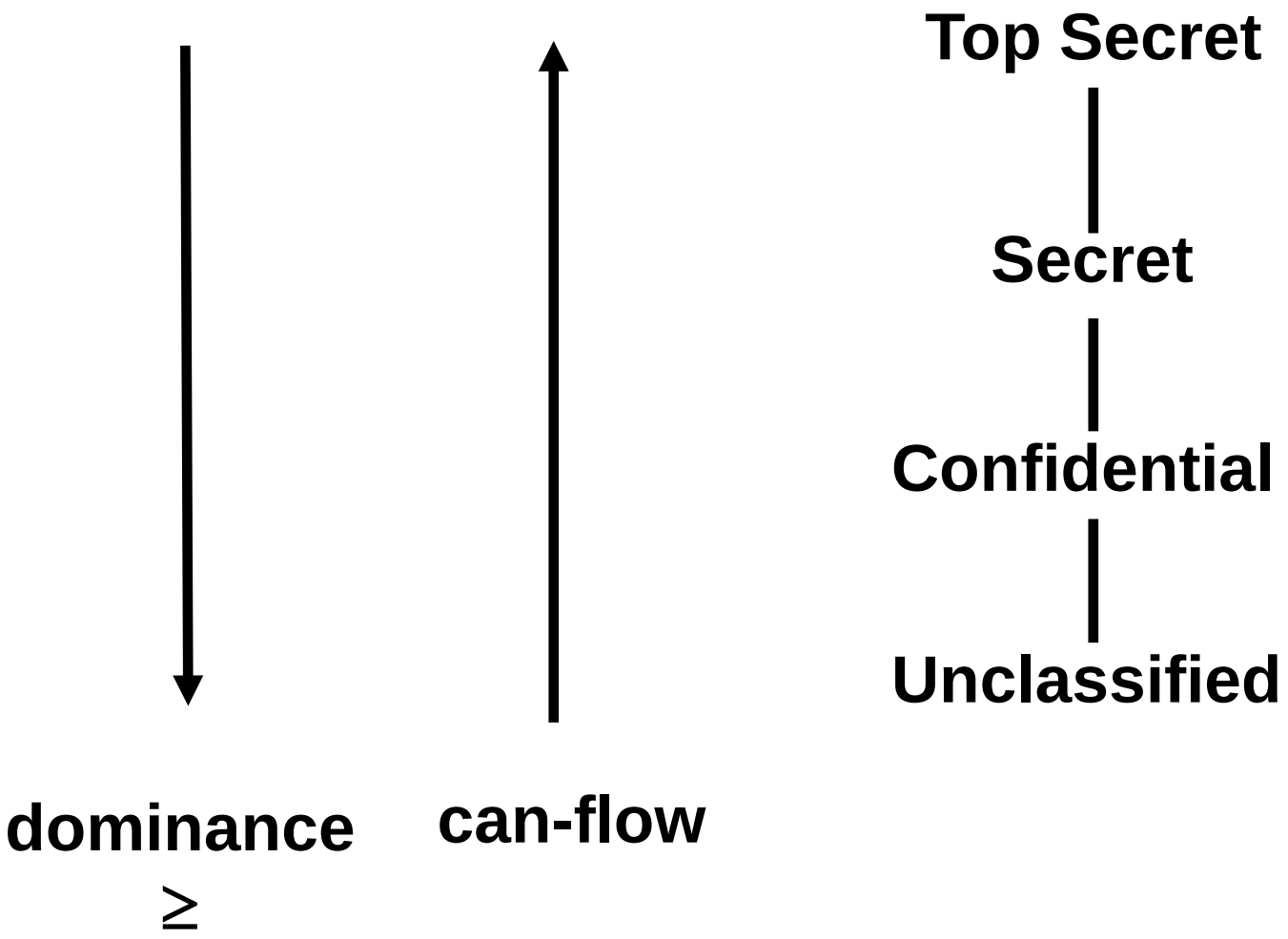
DAC: TROJAN HORSE EXAMPLE



B can read contents of file F copied to file G

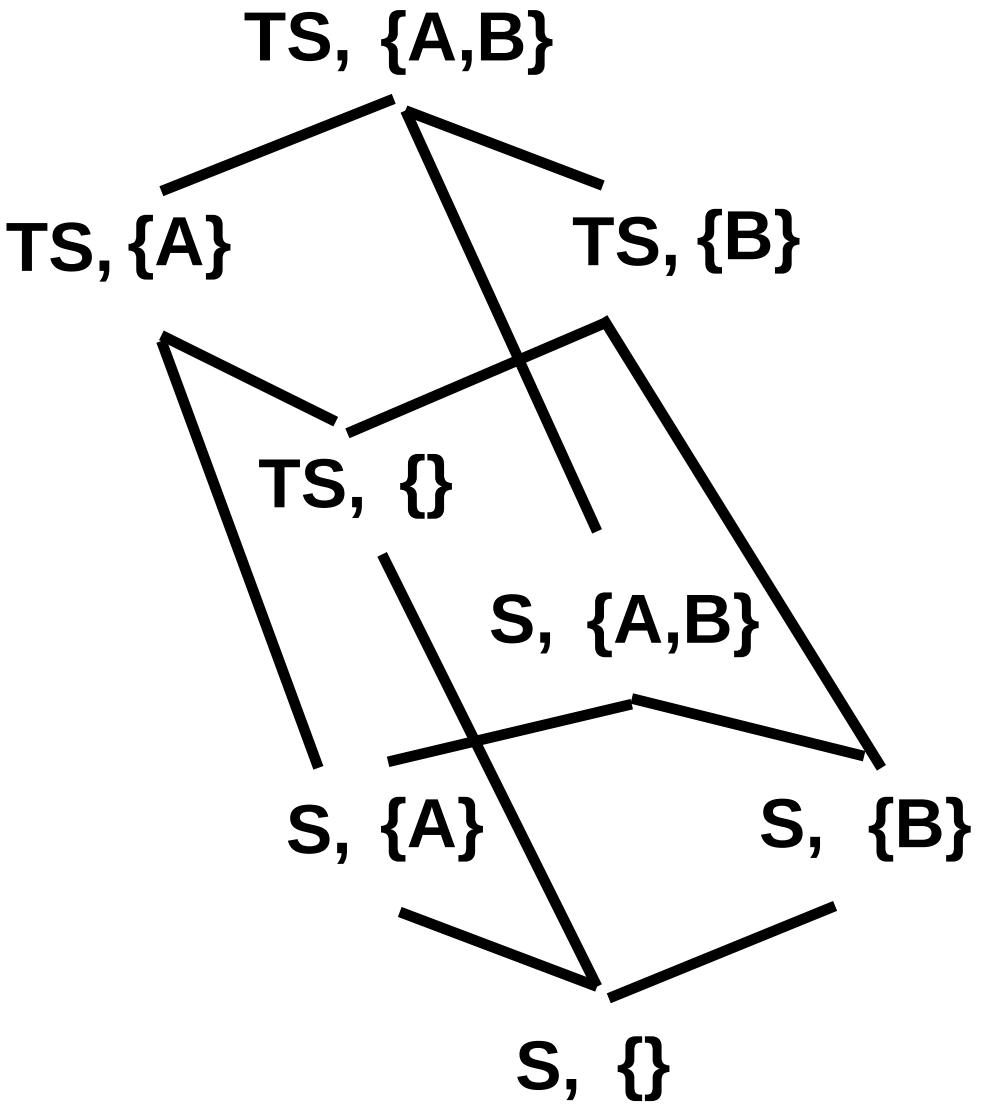
LBAC: LATTICE STRUCTURES

P model



LBAC: LATTICE STRUCTURES

P model



Hierarchical Classes with Compartments

LBAC: LATTICE STRUCTURES

SIMPLE-SECURITY

Subject S can read object O only if

- label(S) dominates label(O)

STAR-PROPERTY (LIBERAL)

Subject S can write object O only if

- label(O) dominates label(S)

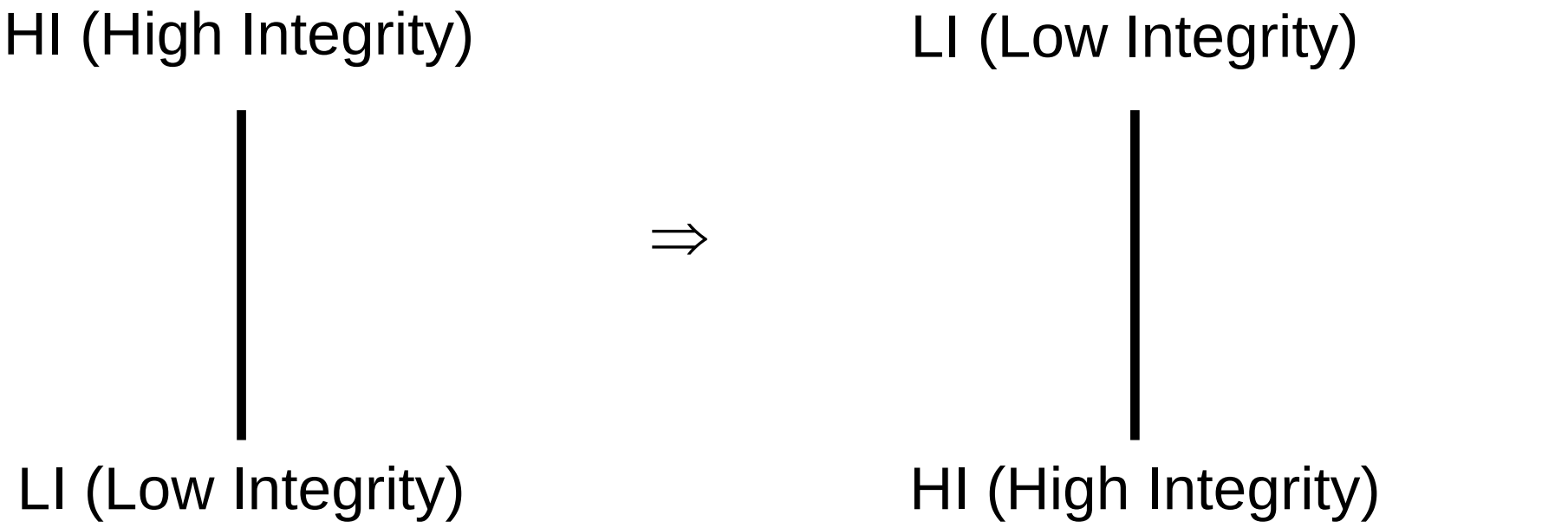
STAR-PROPERTY (STRICT)

Subject S can write object O only if

- label(O) equals label(S)

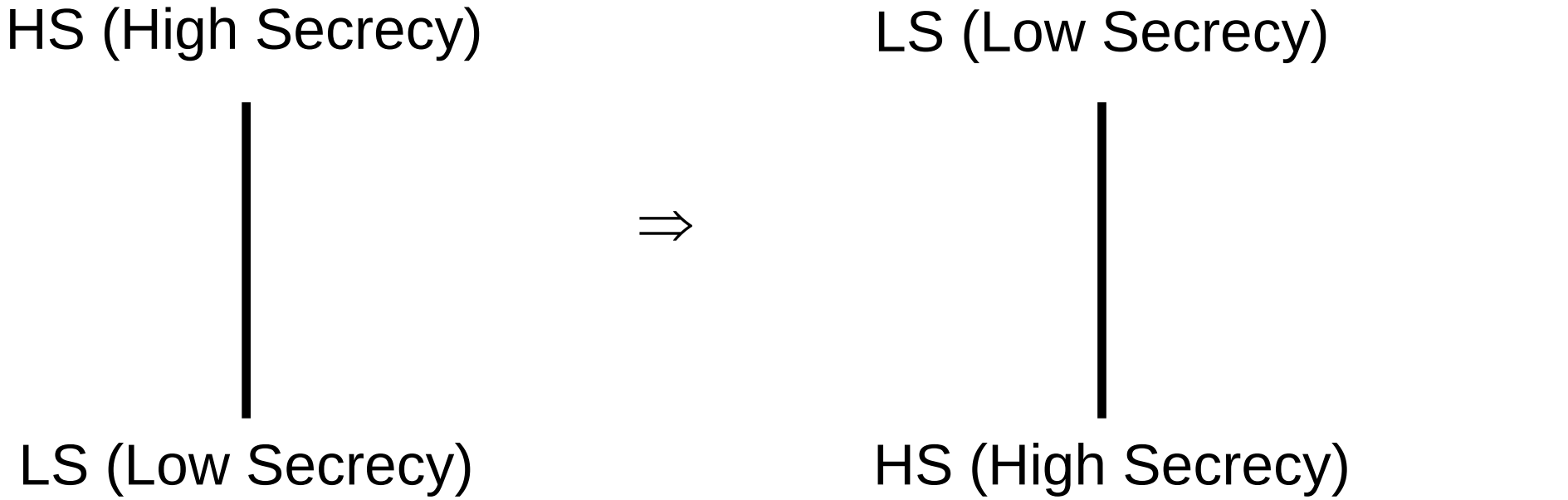
EQUIVALENCE OF BLP AND BIBA

P model



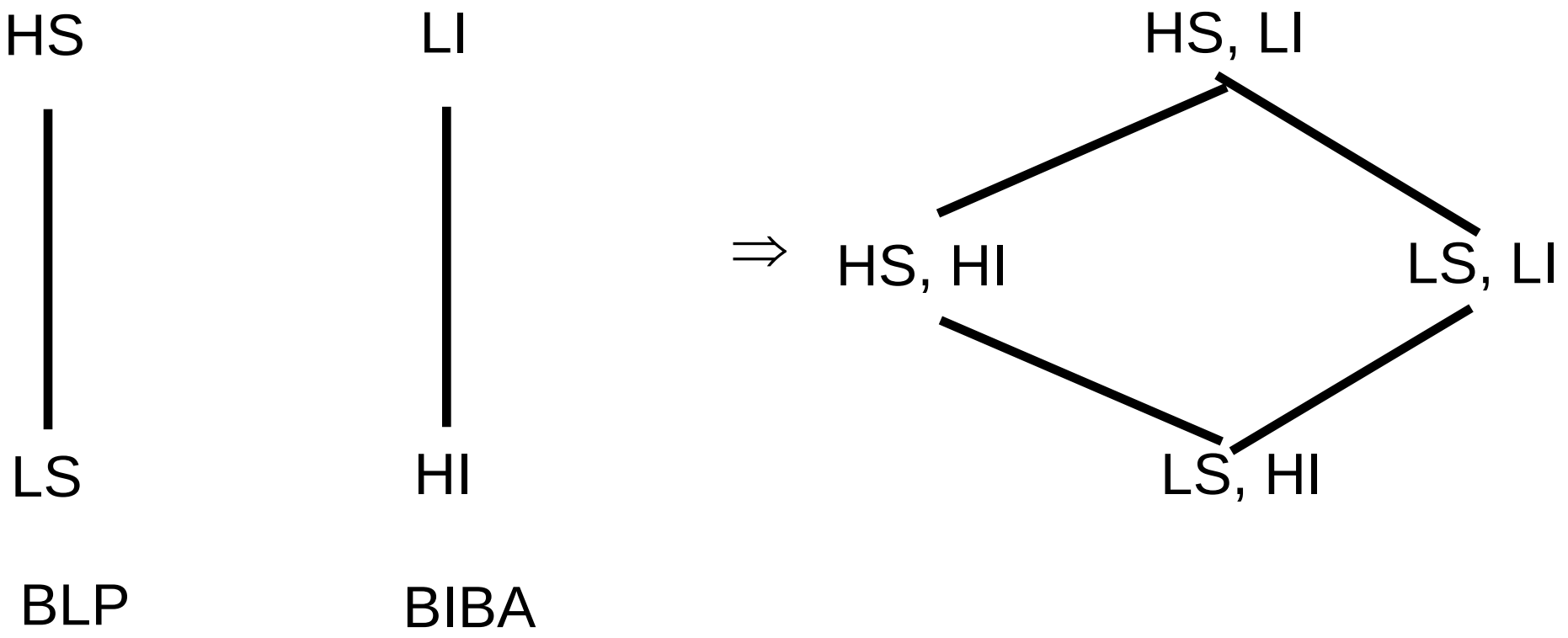
EQUIVALENCE OF BLP AND BIBA

P model



COMBINATION OF DISTINCT LATTICES

P model



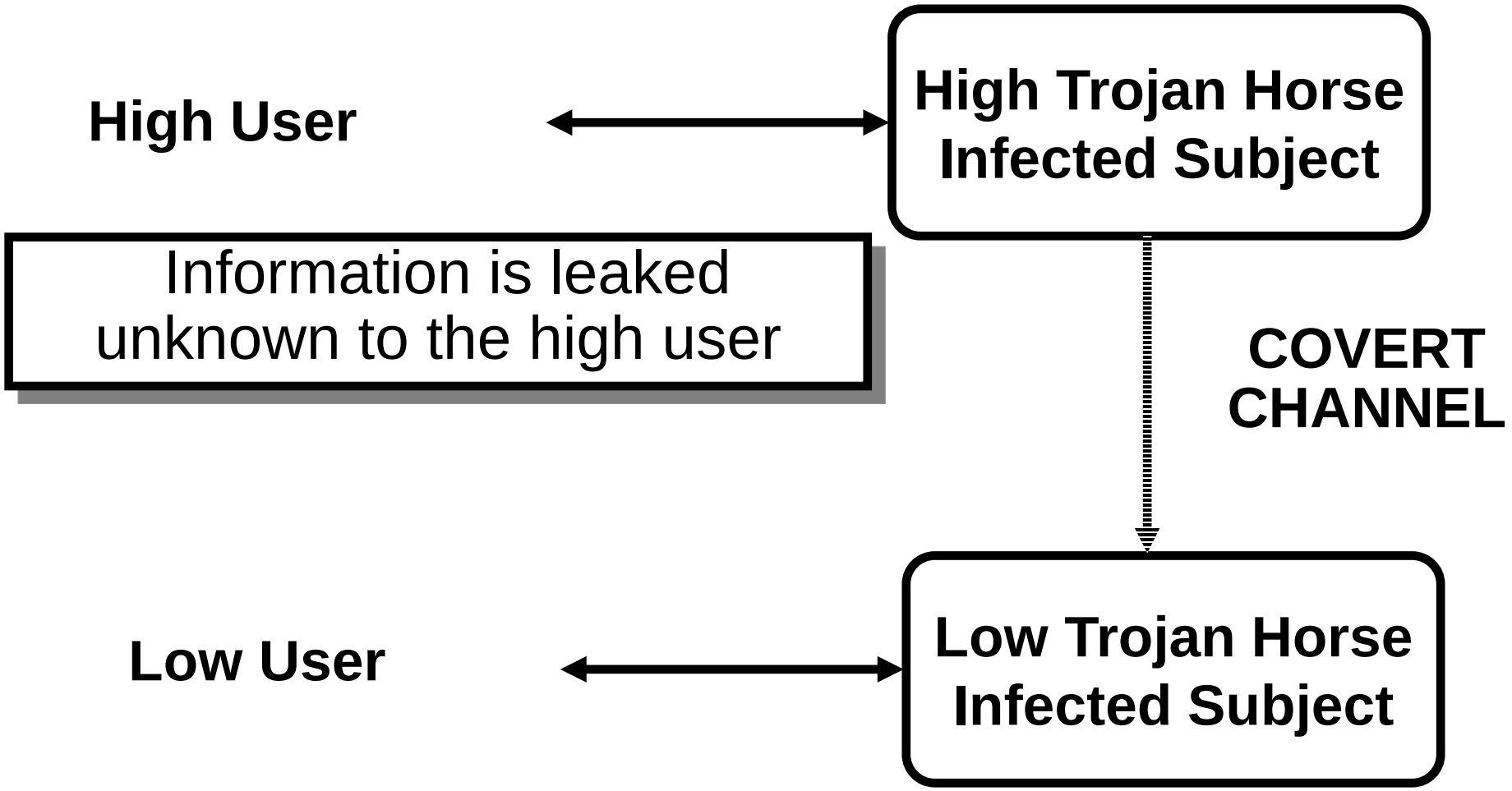
GIVEN

Information flow upwards

EQUIVALENT BLP LATTICE

Information flow upwards

LBAC: LATTICE STRUCTURES



RBAC: Role-Based Access Control

- Access is determined by roles
- A user's roles are assigned by security administrators
- A role's permissions are assigned by security administrators

First emerged: mid 1970s
First models: mid 1990s

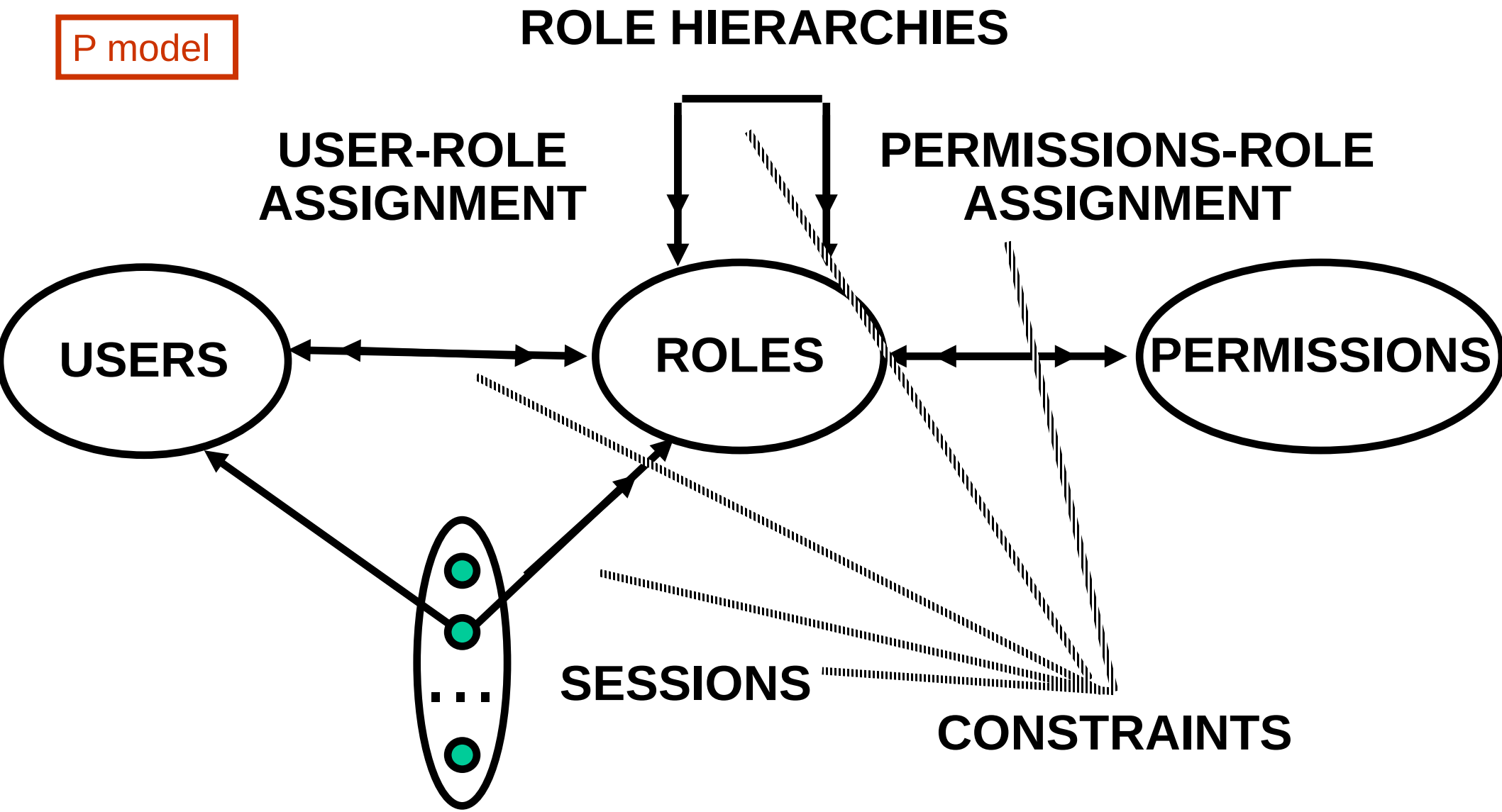
Is RBAC MAC or DAC or neither?

- RBAC can be configured to do MAC
- RBAC can be configured to do DAC
- RBAC is policy neutral

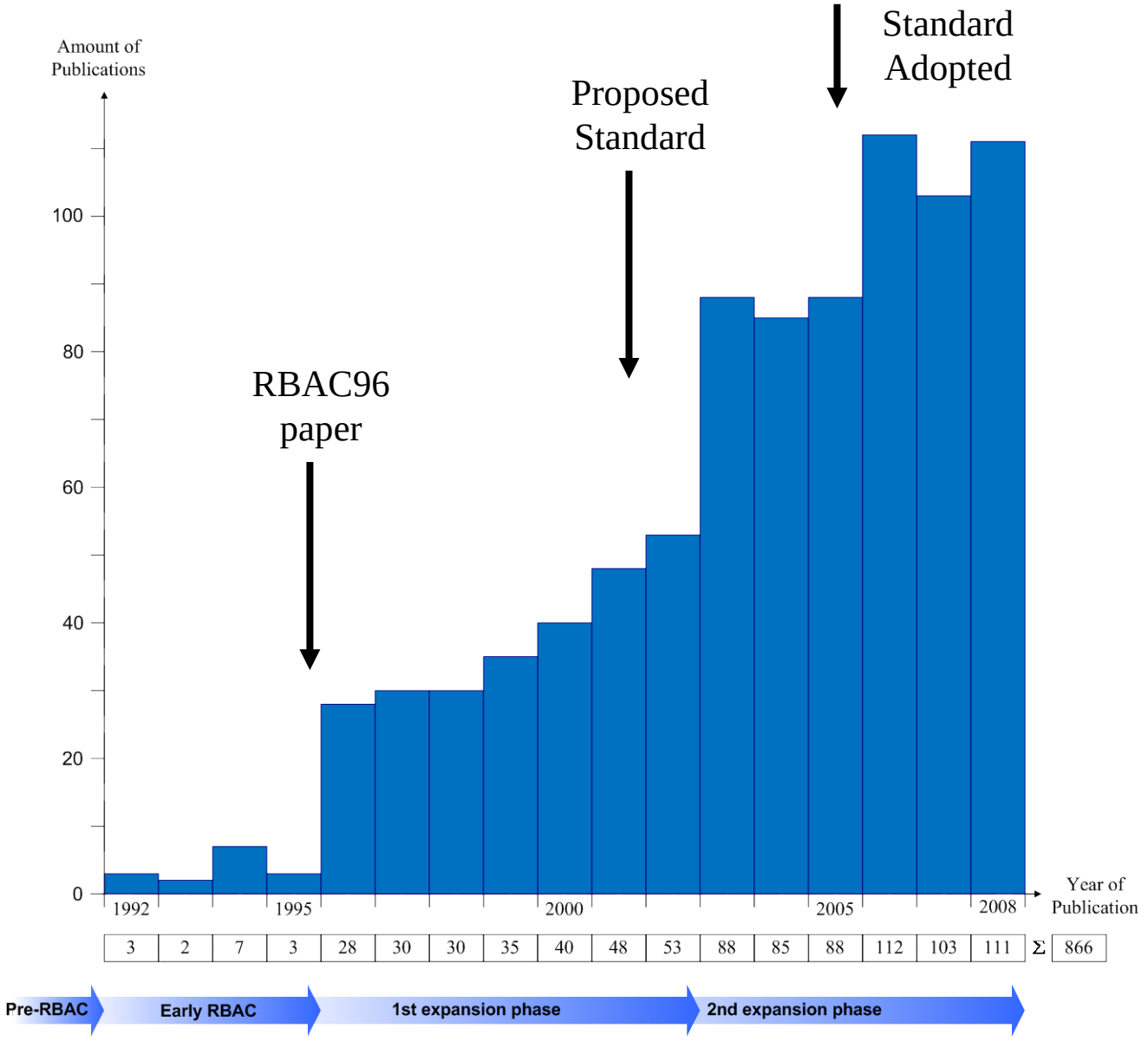
RBAC is neither MAC nor DAC!

RBAC: RBAC96 Model

P model

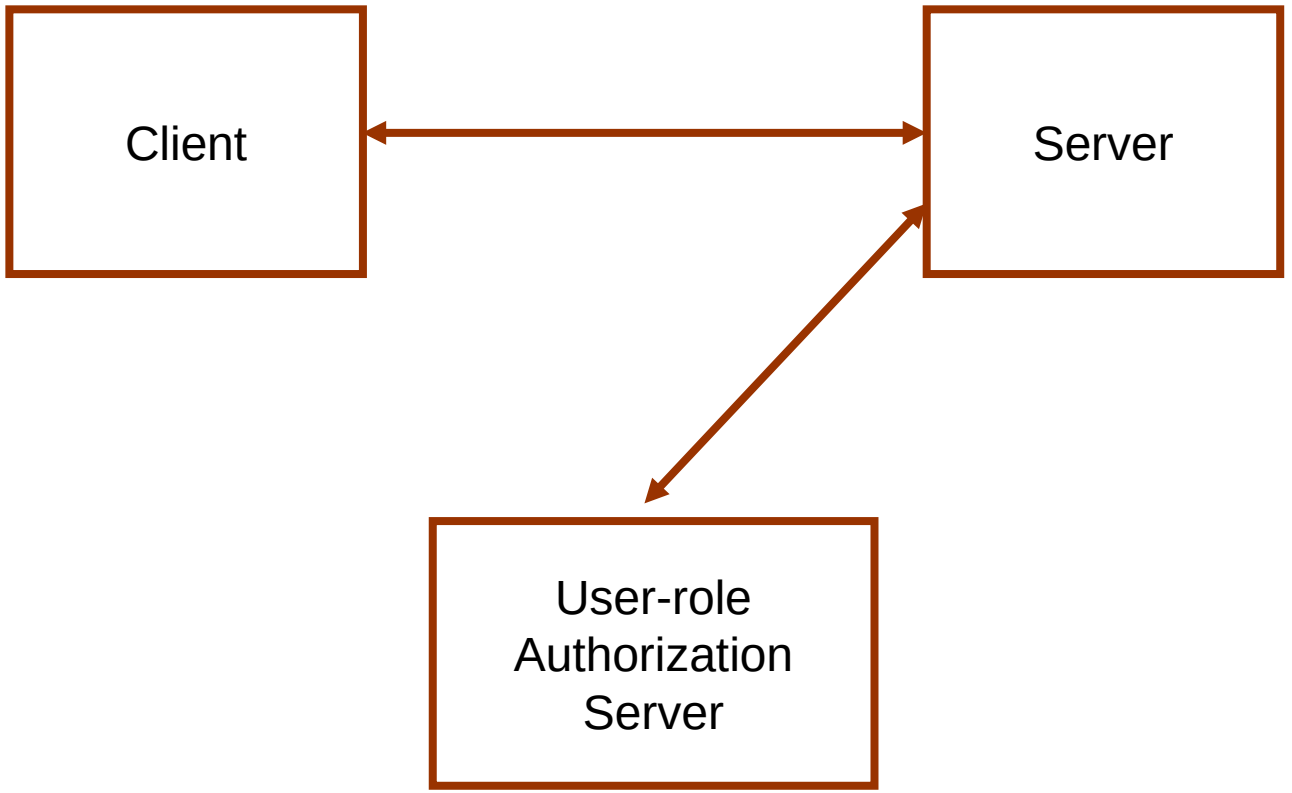


The RBAC Story



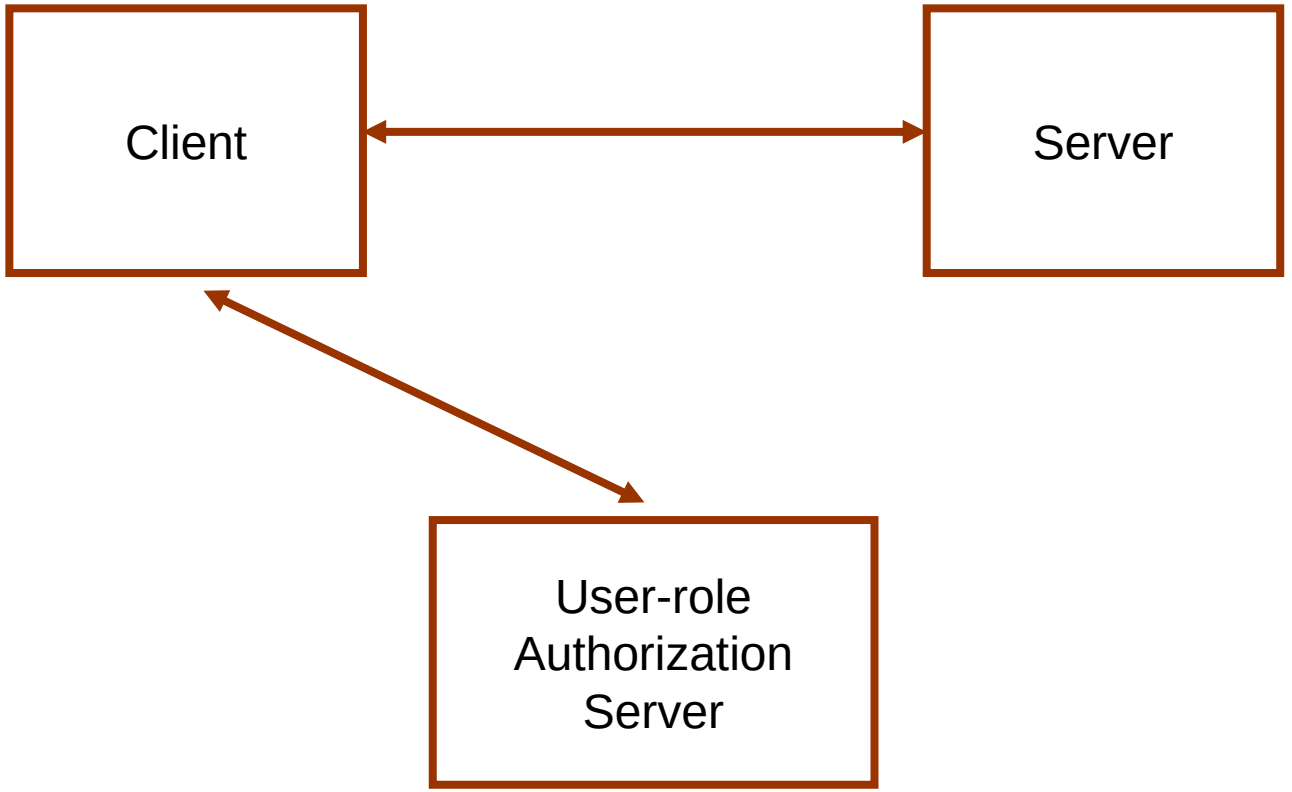
RBAC: SERVER PULL

E model



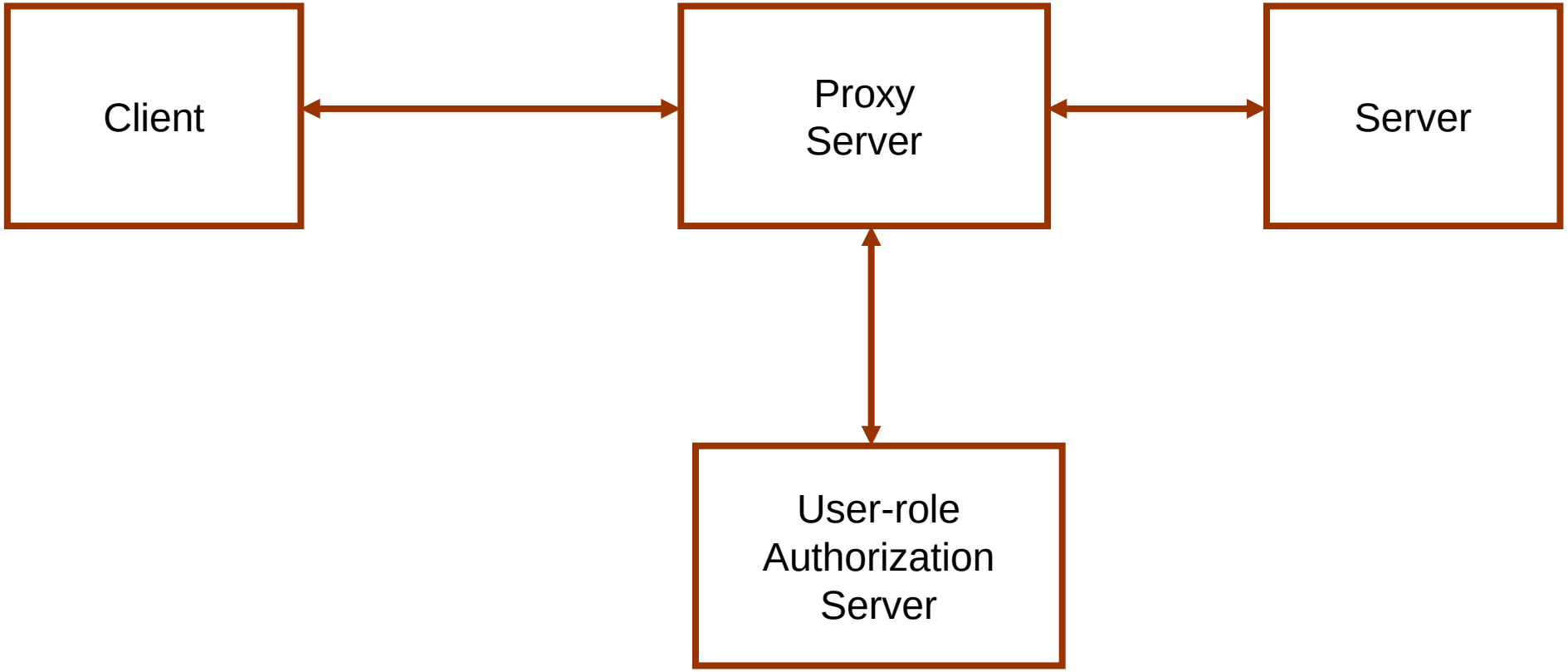
RBAC: CLIENT PULL

E model

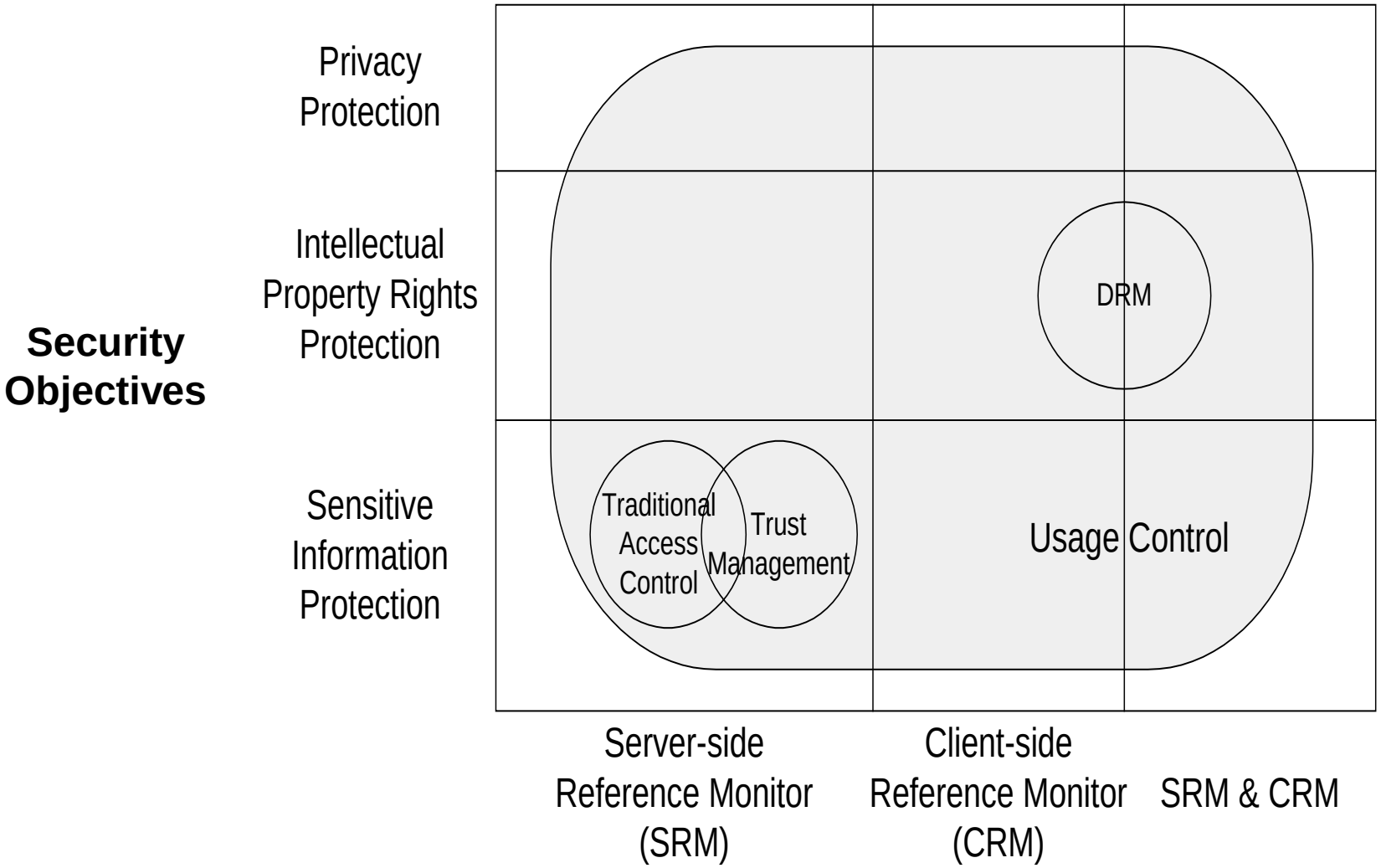


RBAC: PROXY-BASED

E model

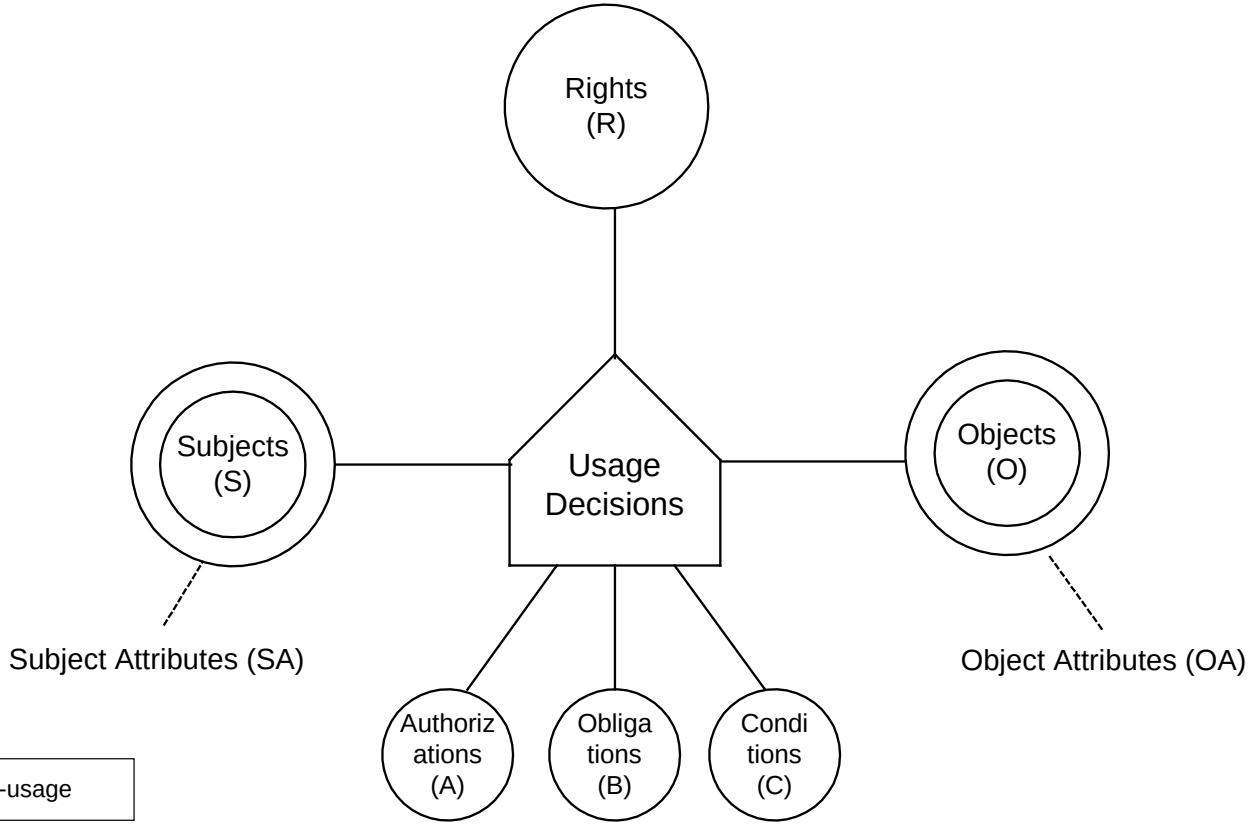


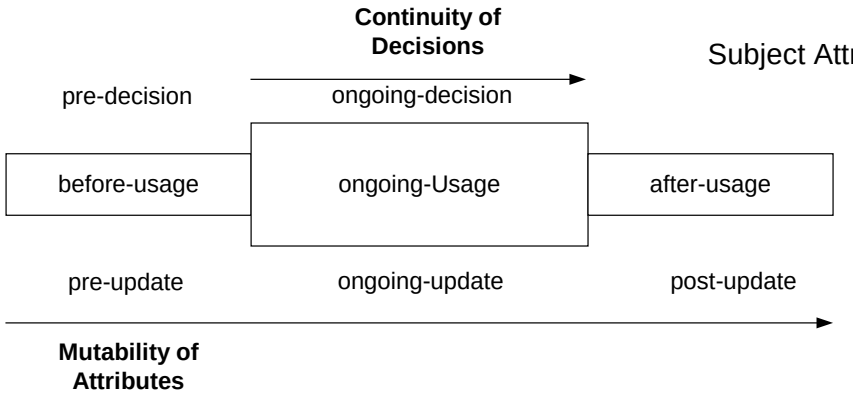
UCON: Usage Control Scope



UCON: Usage Control Model

- unified model integrating
  - authorization
  - obligation
  - conditions
- and incorporating
  - continuity of decisions
  - mutability of attributes



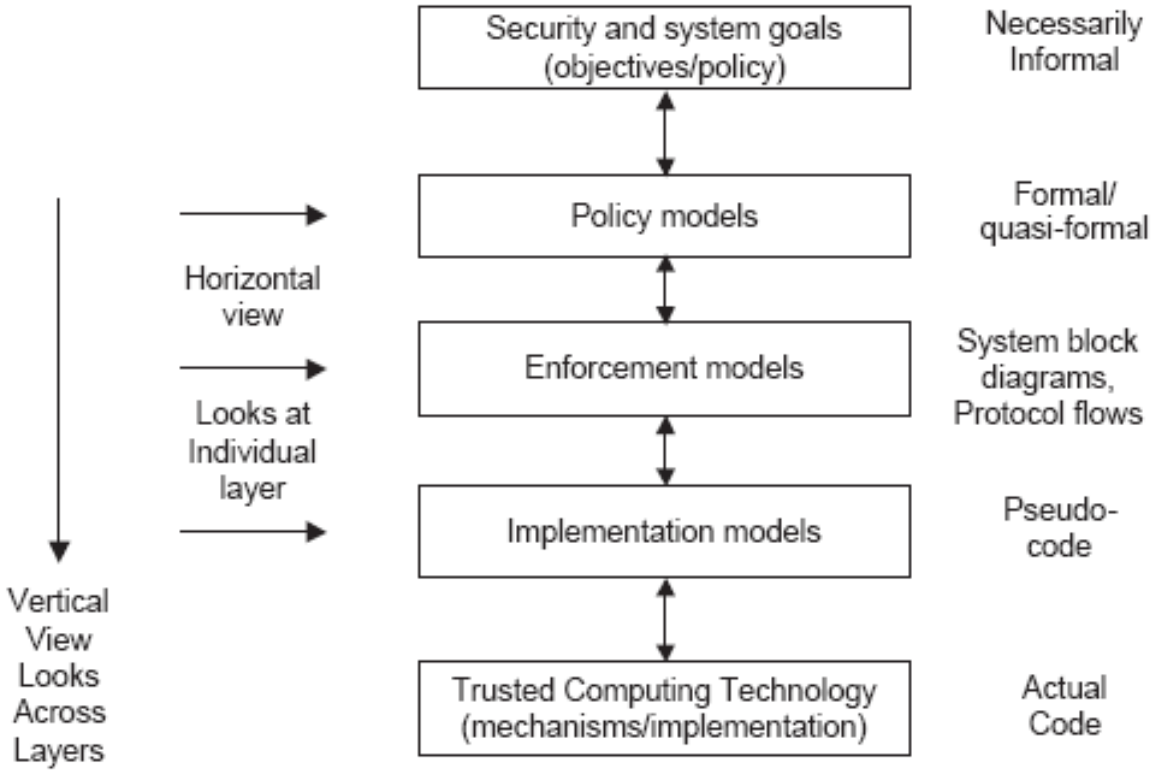


UCON is ABAC on steroids

Application-Centric Security Models

- Our Basic Premise
  - There can be no security model without application context
- So how does one customize an application-centric security model?
  - Meaningfully combine the essential insights of
    - DAC, LBAC, RBAC, ABAC, UCON, etcetera
  - Directly address the application-specific trade-offs
    - Within the security objectives of confidentiality, integrity and availability
    - Across security, performance, cost and usability objectives
  - Separate the real-world concerns of
    - practical distributed systems and ensuing staleness and approximations (enforcement layer) from
    - policy concerns in a idealized environment (policy layer)

PEI Models



At the policy layer security models are essentially access control models