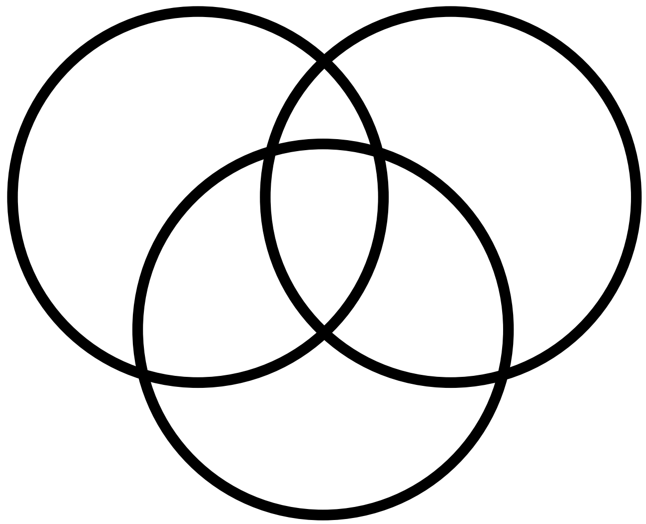
Security Models: Past, Present and Future

Prof. Ravi Sandhu
Executive Director and Endowed Chair
Institute for Cyber Security
University of Texas at San Antonio
July 2009

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

THE BIG PICTURE

Security Objectives

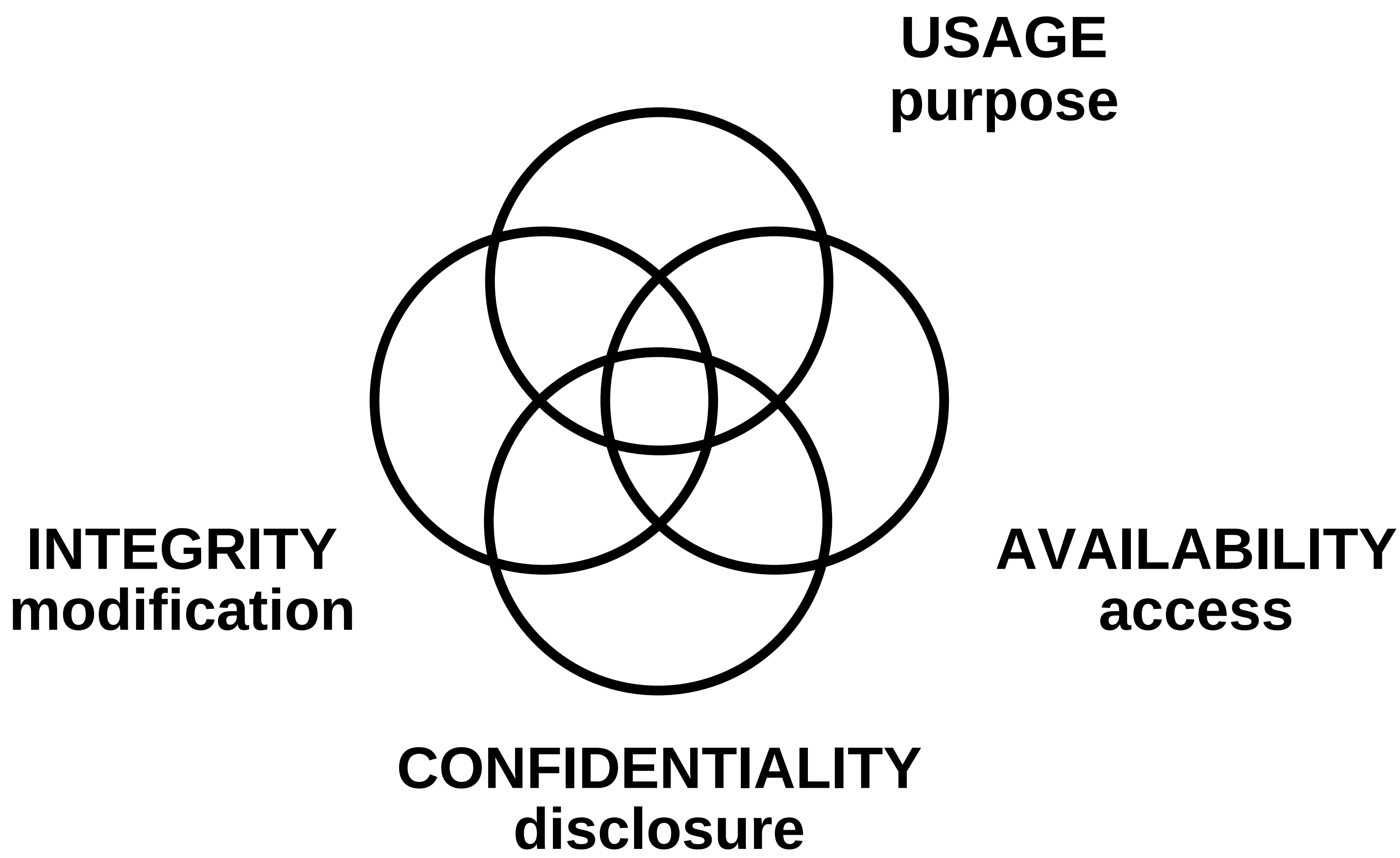


INTEGRITY
modification

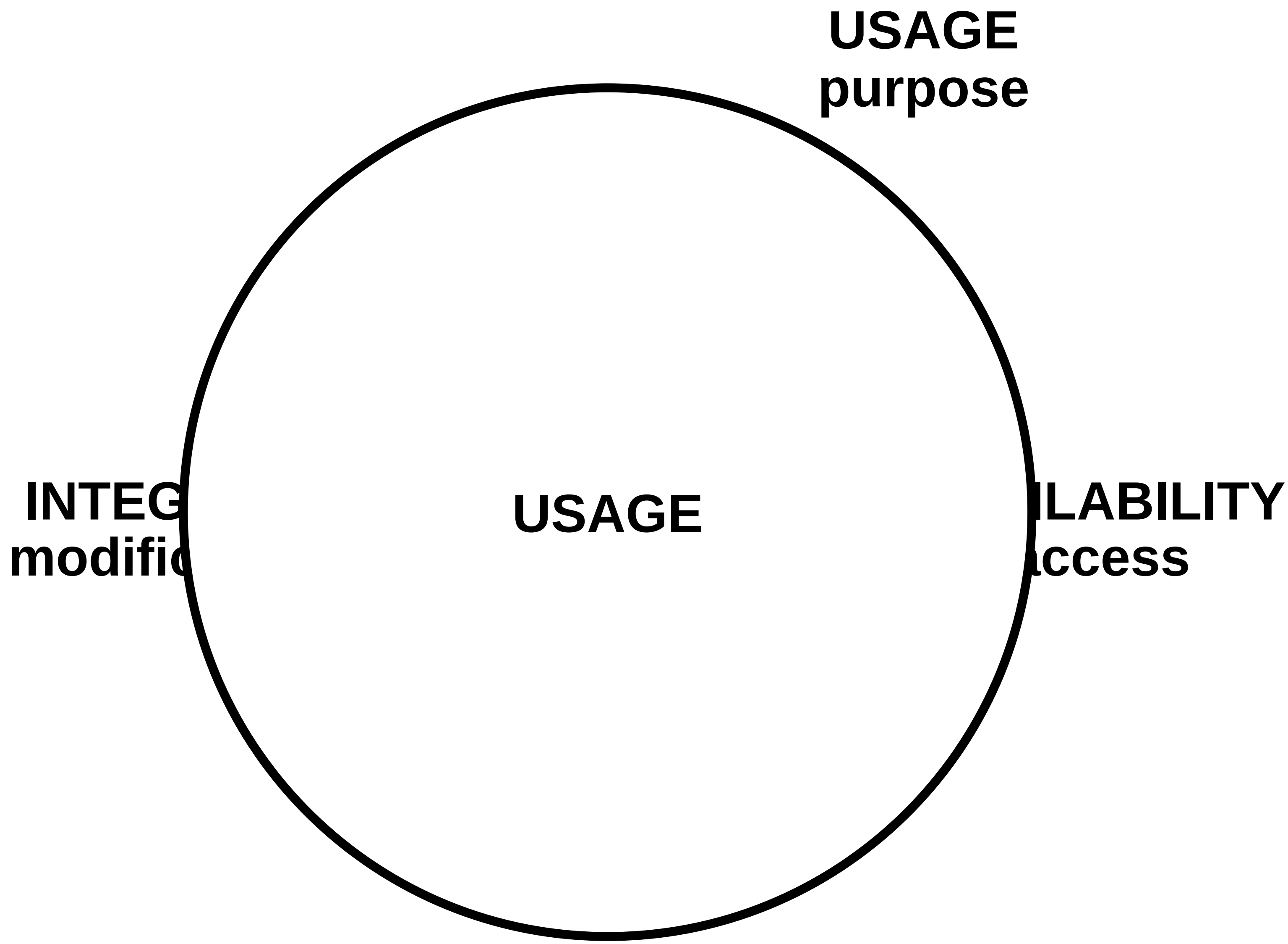
AVAILABILITY
access

CONFIDENTIALITY
disclosure

Security Objectives



Security Objectives



Security Trends and Change Drivers

Stand-alone computers	→	**Internet**
Vandals	→	**Criminals, Nation states, Terrorists**
Enterprise security	→	**Mutually suspicious yet mutually dependent security**
Few standard services	→	**Many and new innovative services**

We are at an inflection point

Diffie on Information Security … 2007

- "Now we face a new challenge to security, a world of shared computing and web services. As with radio, this technology is too valuable to go unused, By contrast with radio, which could be protected with cryptography, there may be no technology that can protect shared computation to the degree we would call secure today. In a decade or a generation, there may be no secure computing."

Need to be realistic in our security expectations

Butler Lampson Paraphrased (I think)

- Computer scientists could never have designed the web because they would have tried to make it work.
  But the Web does "work."
  What does it mean for the Web to "work"?
- Security geeks could never have designed the ATM network because they would have tried to make it secure.
  But the ATM network is "secure.
  What does it mean for the ATM network to be "secure"?

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

Application Context

- Our Basic Premise
  - There can be no security without application context
  - Courtney's Law (1970s, 1980s ??):
    - You cannot say anything interesting (i.e. significant) about the security of a system except in the context of a particular application and environment
- Corollary
  - There can be no security model without application context
- Reality
  - Existing security models are application neutral
    - Assumption is they can be readily "configured" or "policy-fied" to suit application context

There is also a notion of technology context for security models but out of scope for this lecture

Application Context

Software-Architect	Project	% Time	Label
Alice	Vista	25%	U
Alice	SecureVista	75%	S
Bob	XP	100%	U

- What precisely is Secret?
  - There exists a SecureVista project
  - Alice works on SecureVista
  - Alice's effort on SecureVista is 75%
  - All or some of the above
- How do we maintain integrity of the database?
  - Depends

Much work and $$$ by researchers and vendors, late 80's-early 90's

Emerging Application-Centric Era (ACE)

ECE Enterprise-Centric Era	→	ACE Application-Centric Era

Applications are cyber analogs of previously existing enterprise-centric applications

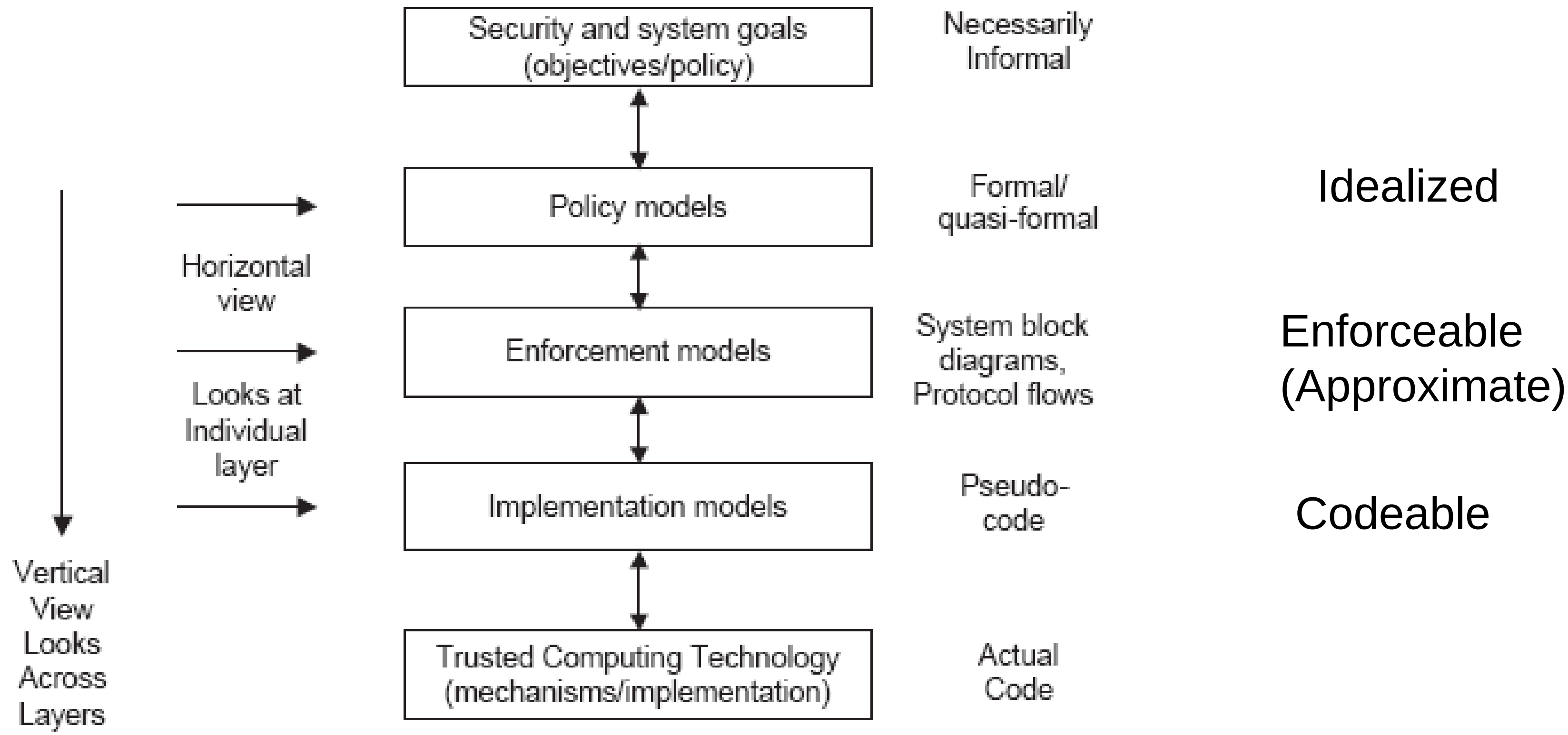
- on-line banking
- brokerage
- e-retail
- auctions
- search engines

Future applications will be fundamentally different

- ?
- ?
- ?
- ?
- ?

PEI Models: 3 Layers/5 Layers

This lecture is focused on the policy models layer



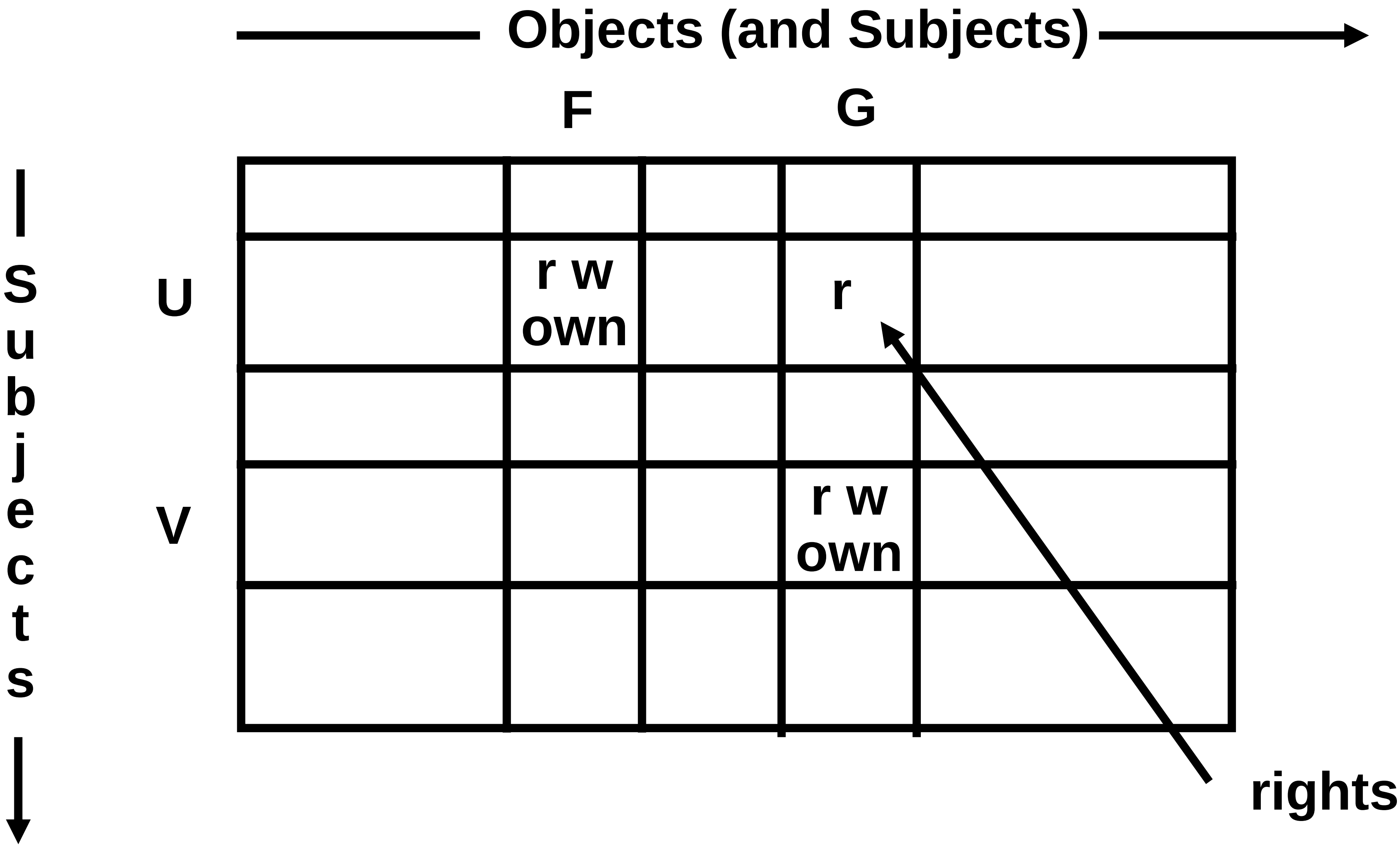
At the policy layer security models are essentially access control models

THE PAST

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

ACCESS MATRIX MODEL



ACCESS CONTROL LISTS (ACLs)

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

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

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

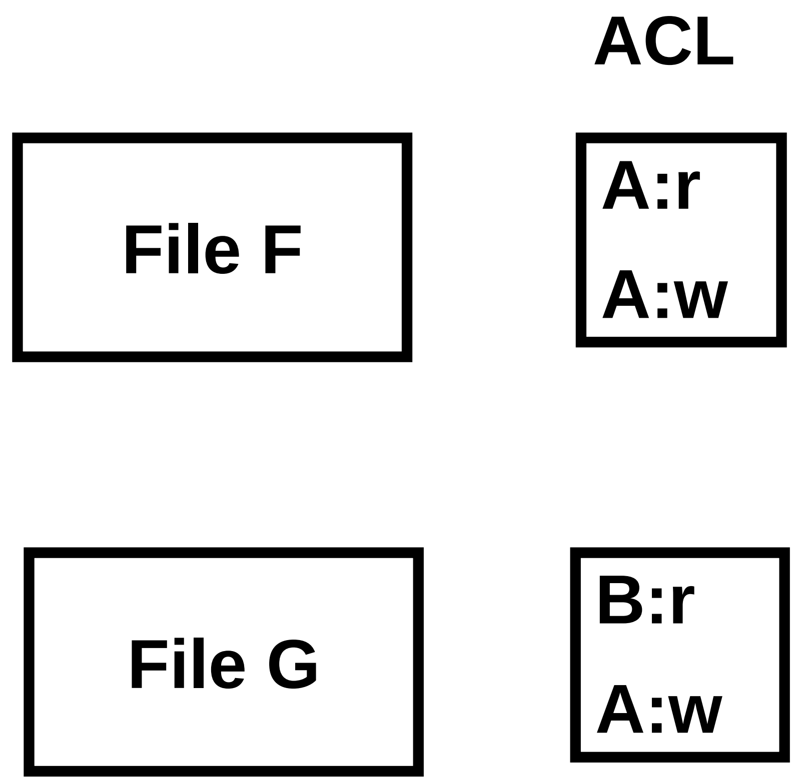
each row of the access matrix is stored with the subject corresponding to that row

ACCESS CONTROL TRIPLES

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

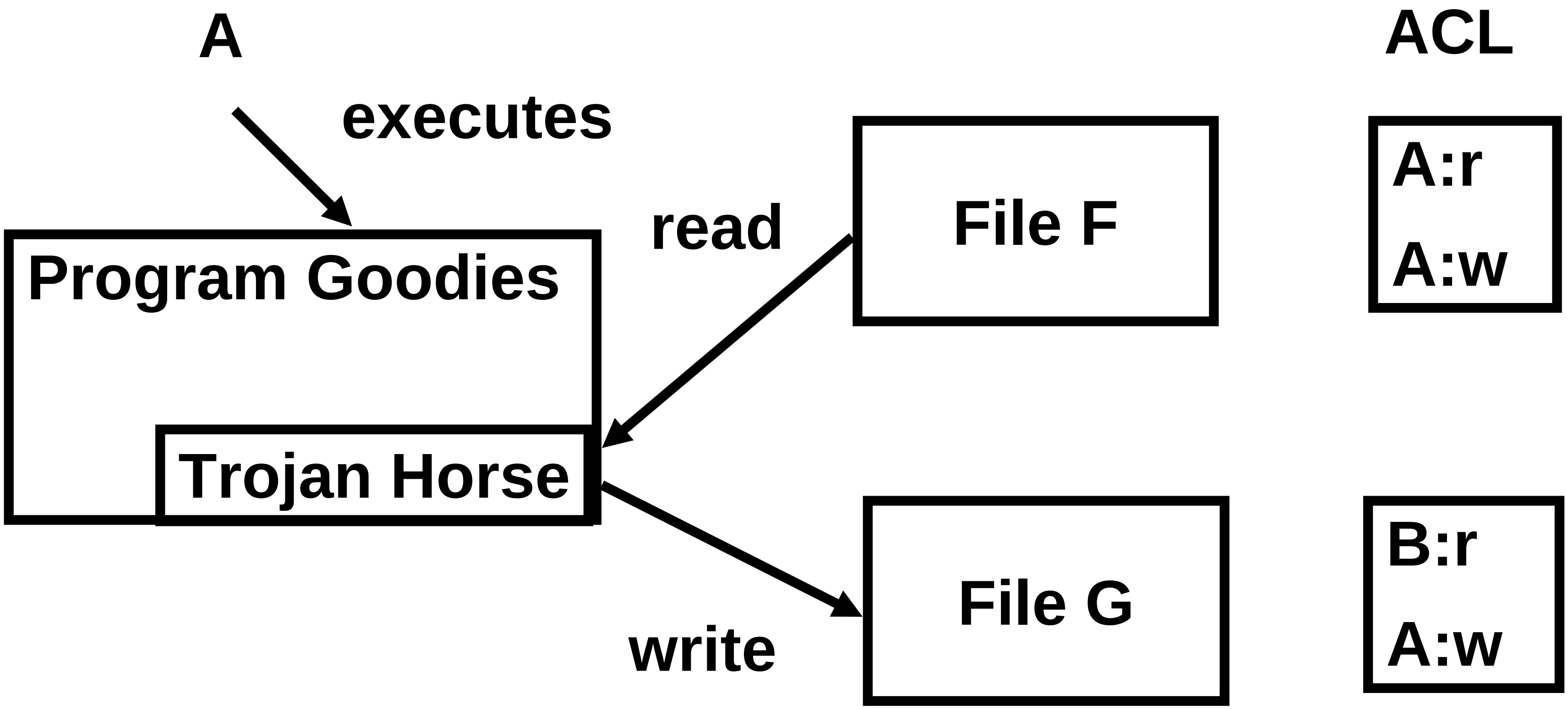
commonly used in relational database management systems

TROJAN HORSE EXAMPLE



B cannot read file F

TROJAN HORSE EXAMPLE

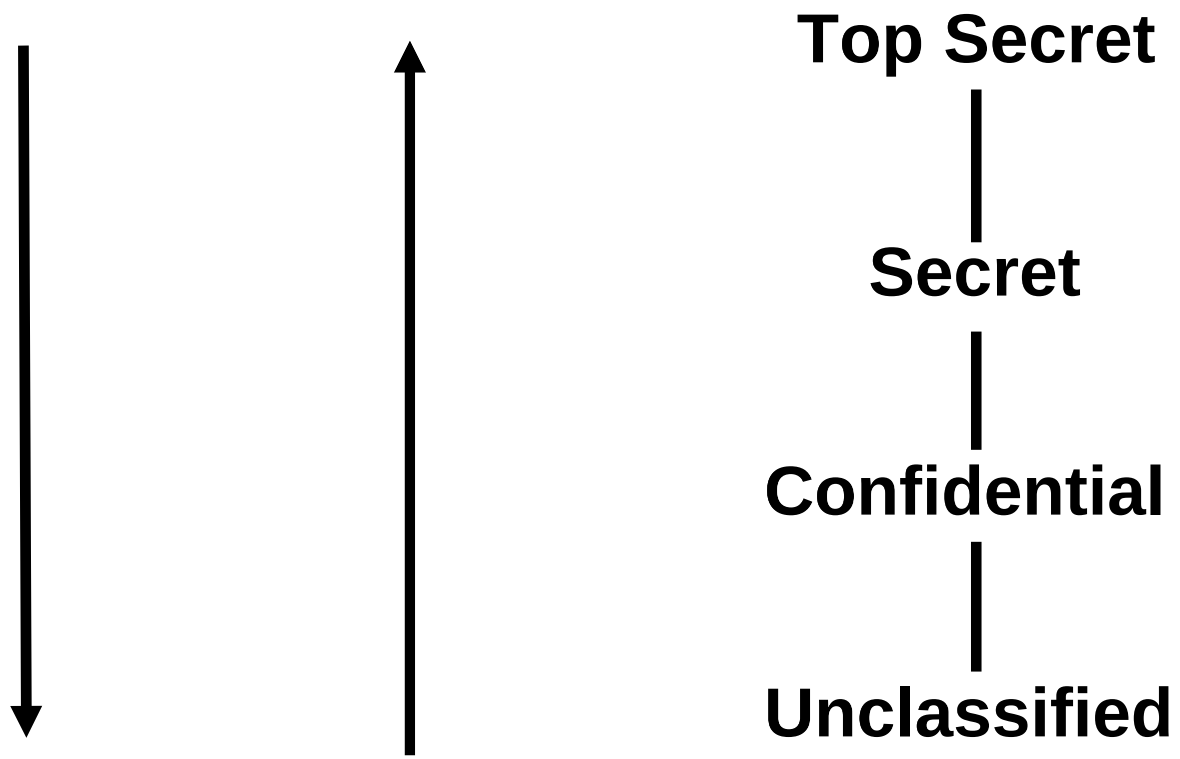


B can read contents of file F copied to file G

DAC Summary

- Traditional DAC does not prevent copies from being made and there is no control over copies
  - Modern approaches to information sharing and trusted computing seek to maintain control over copies
- Traditional DAC is weak with respect to confidentiality but may have value with respect to integrity

LATTICE STRUCTURES



dominance
≥

can-flow

BELL LAPADULA (BLP) MODEL

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)

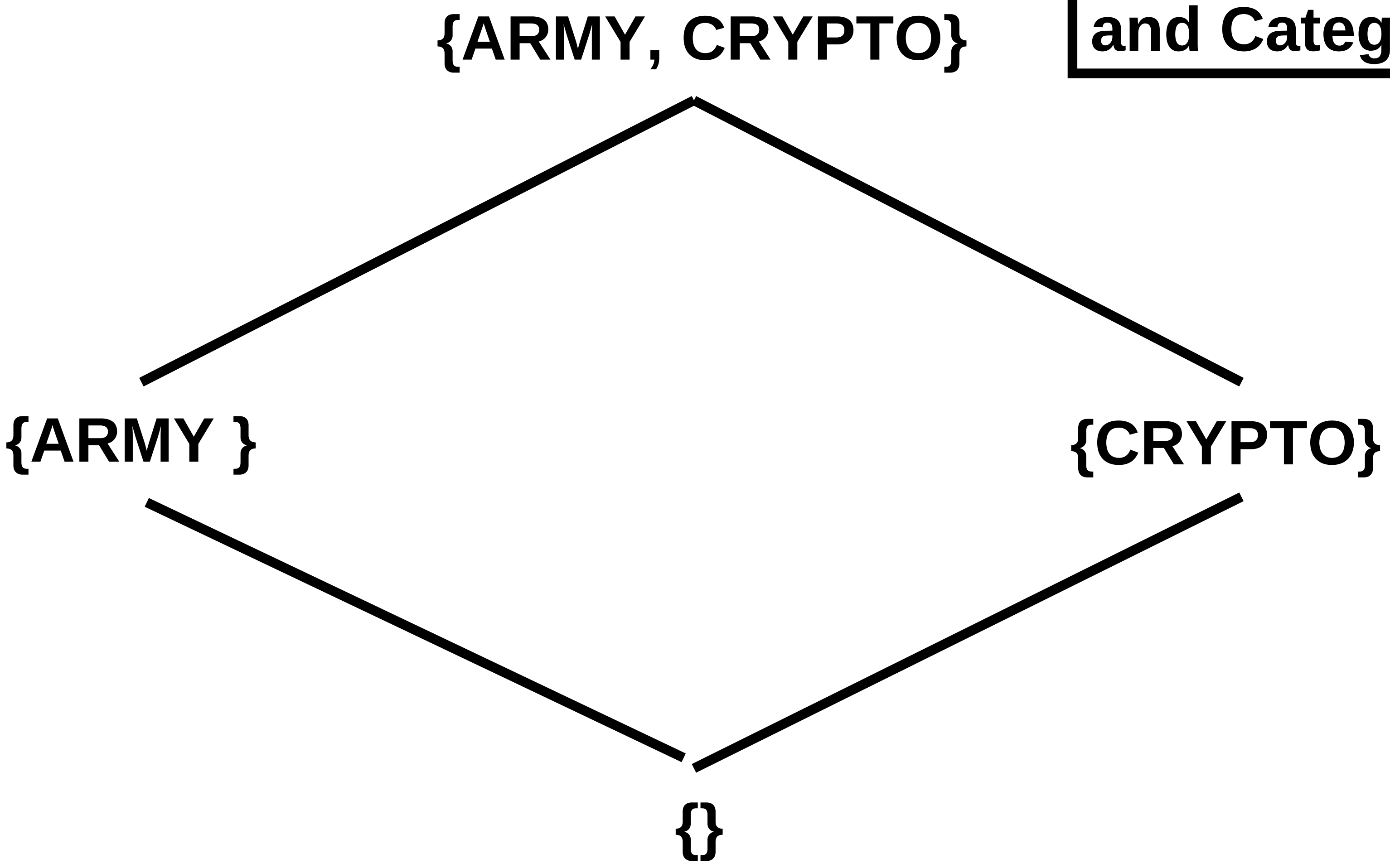
STAR-PROPERTY (STRICT)

Subject S can write object O only if

- label(O) equals label(S)

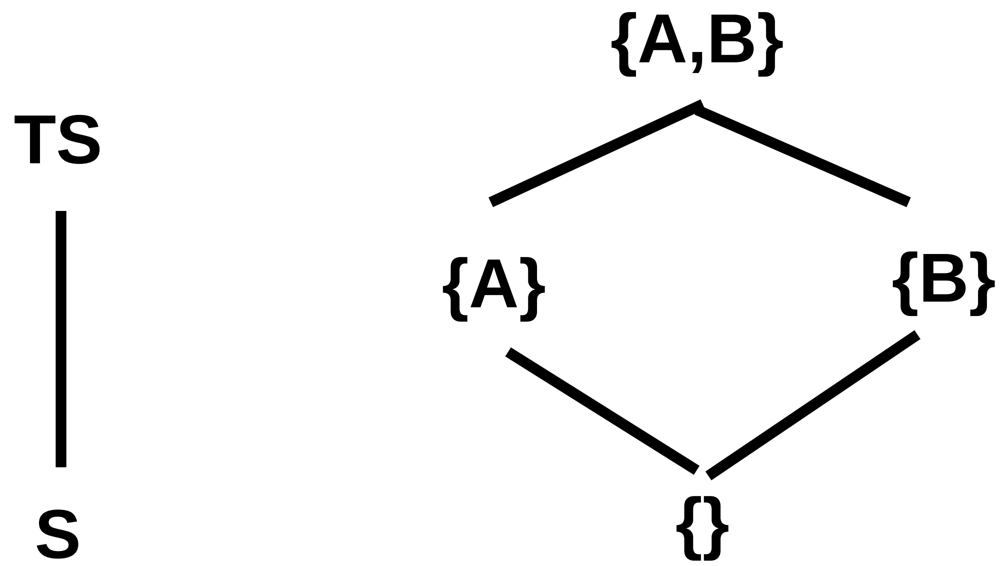
LATTICE STRUCTURES

Compartments and Categories



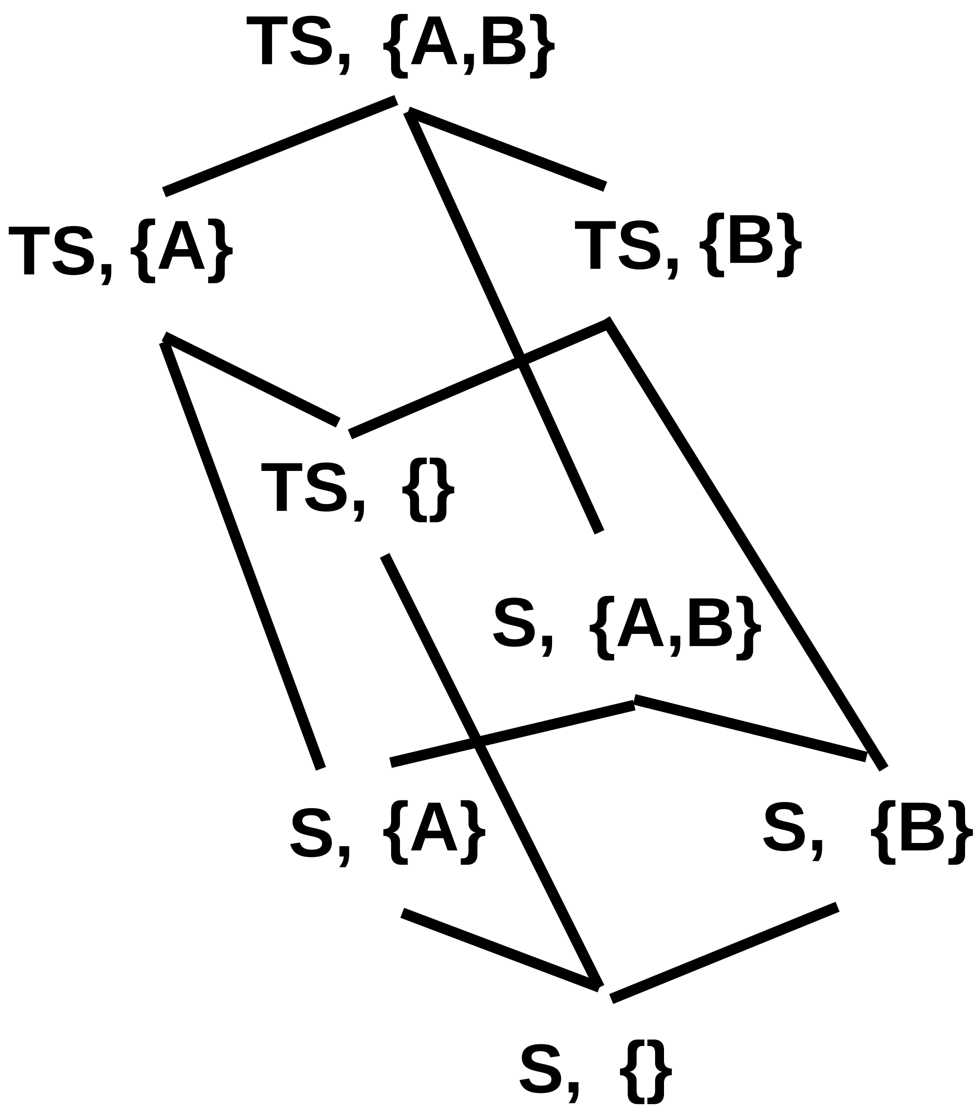
LATTICE STRUCTURES

Hierarchical Classes with Compartments



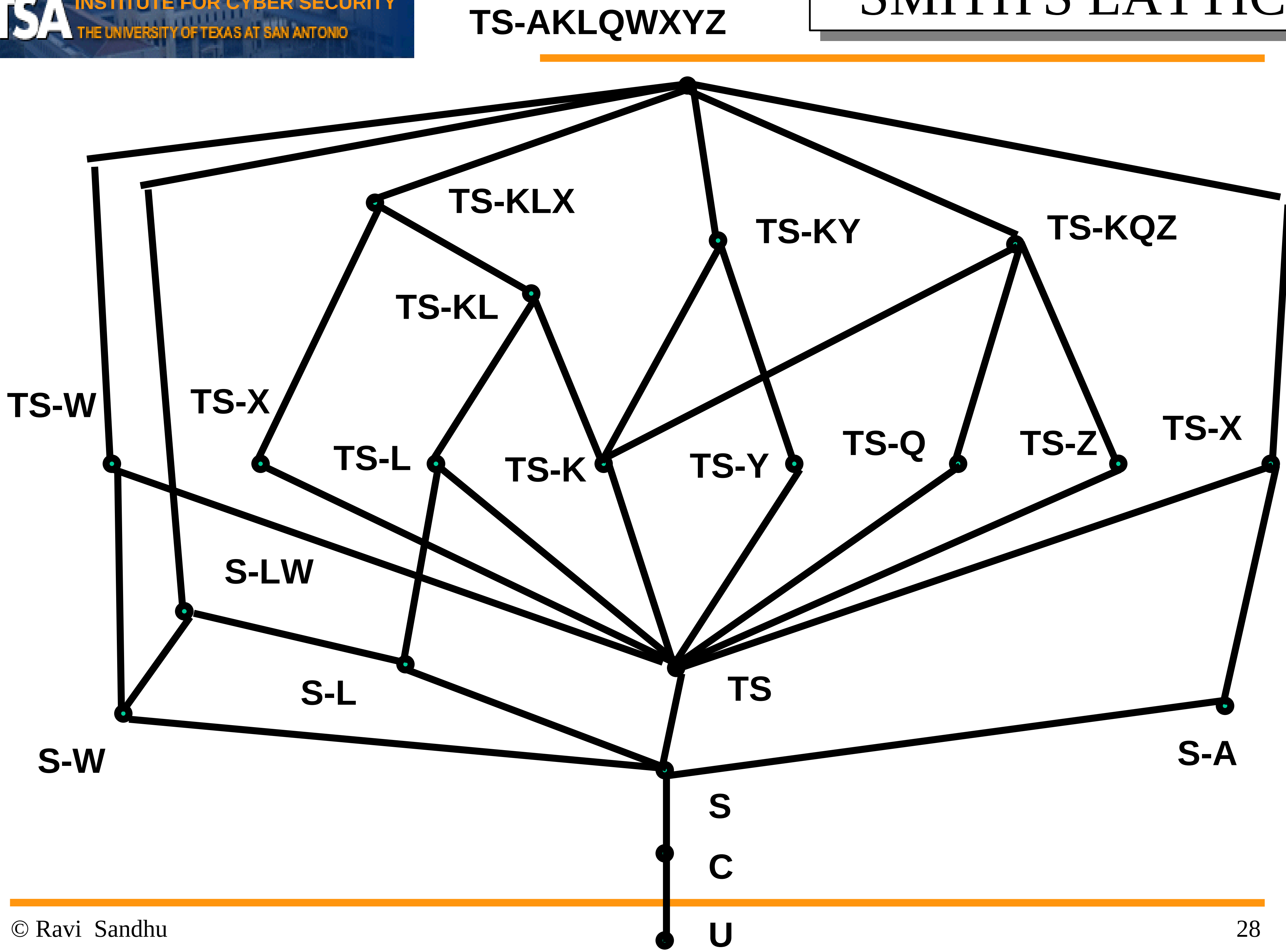
product of 2 lattices is a lattice

LATTICE STRUCTURES



Hierarchical Classes with Compartments

SMITH'S LATTICE



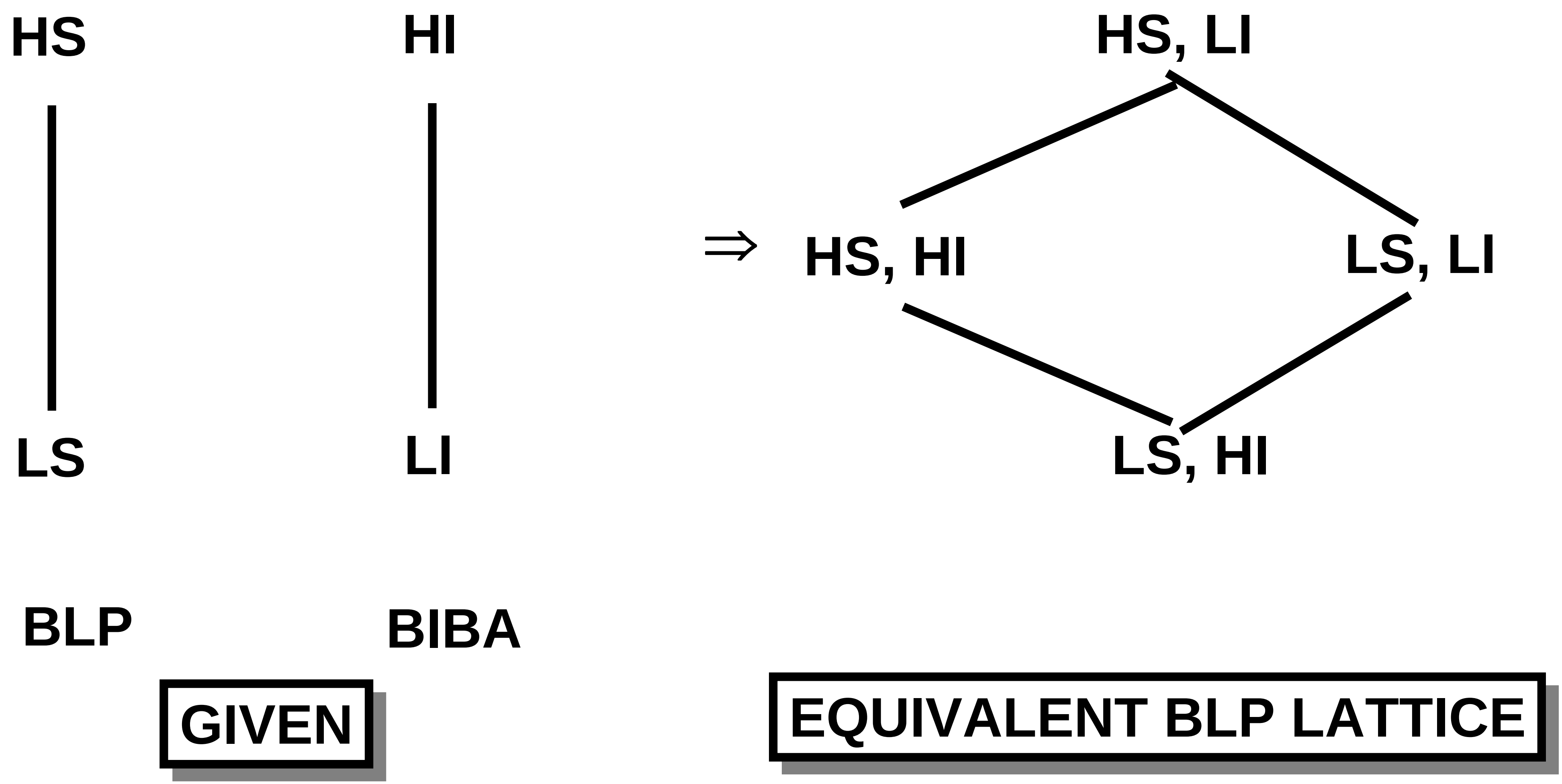
EQUIVALENCE OF BLP AND BIBA

BIBA lattice		Equivalent BLP lattice
HI (High Integrity)		LI (Low Integrity)
|	$\Rightarrow$	|
LI (Low Integrity)		HI (High Integrity)
BIBA LATTICE		**EQUIVALENT BLP LATTICE**

EQUIVALENCE OF BLP AND BIBA

BLP		BIBA
HS (High Secrecy)		**LS (Low Secrecy)**
|	⇒	|
LS (Low Secrecy)		**HS (High Secrecy)**
BLP LATTICE		**EQUIVALENT BIBA LATTICE**

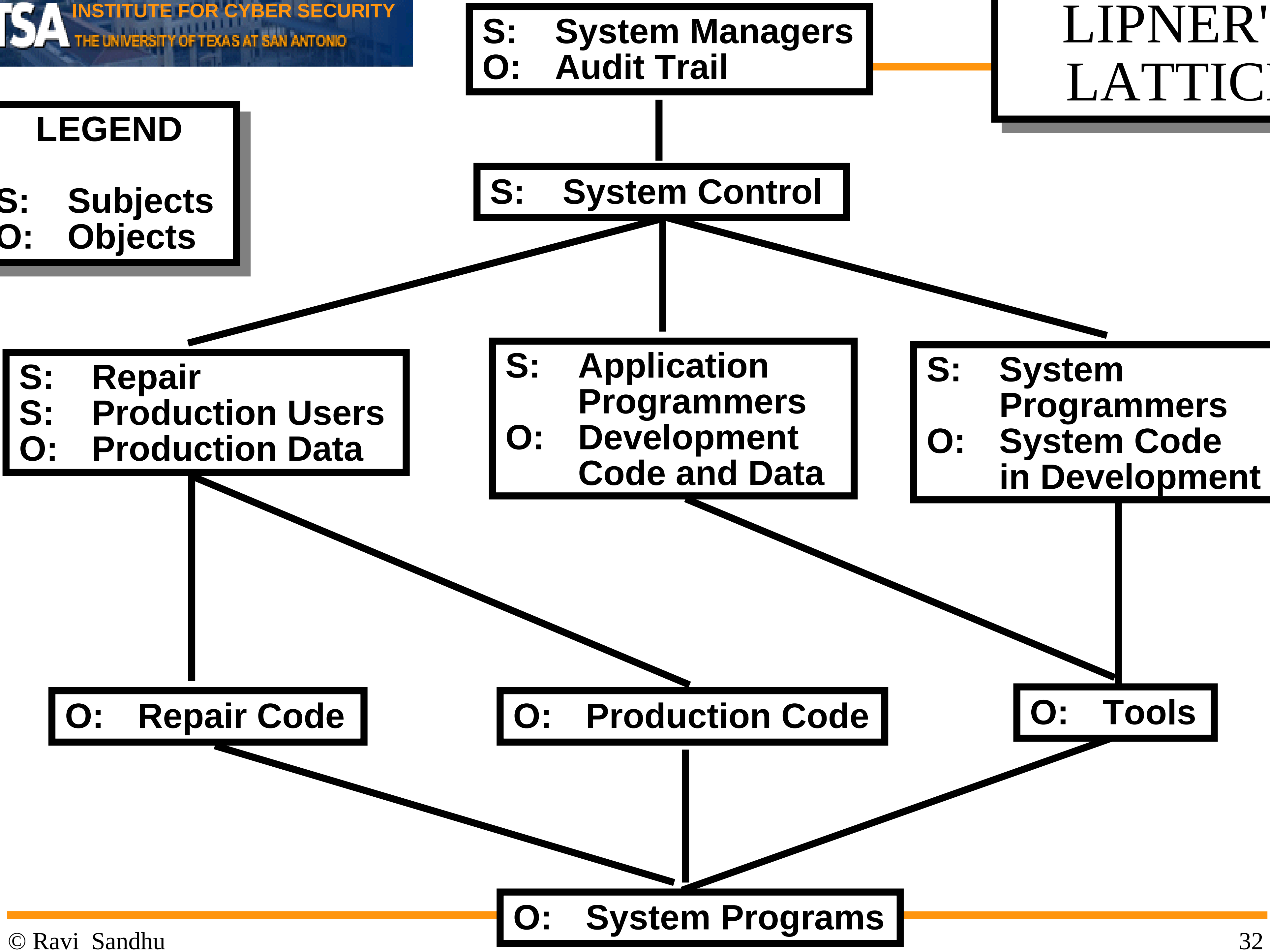
COMBINATION OF DISTINCT LATTICES



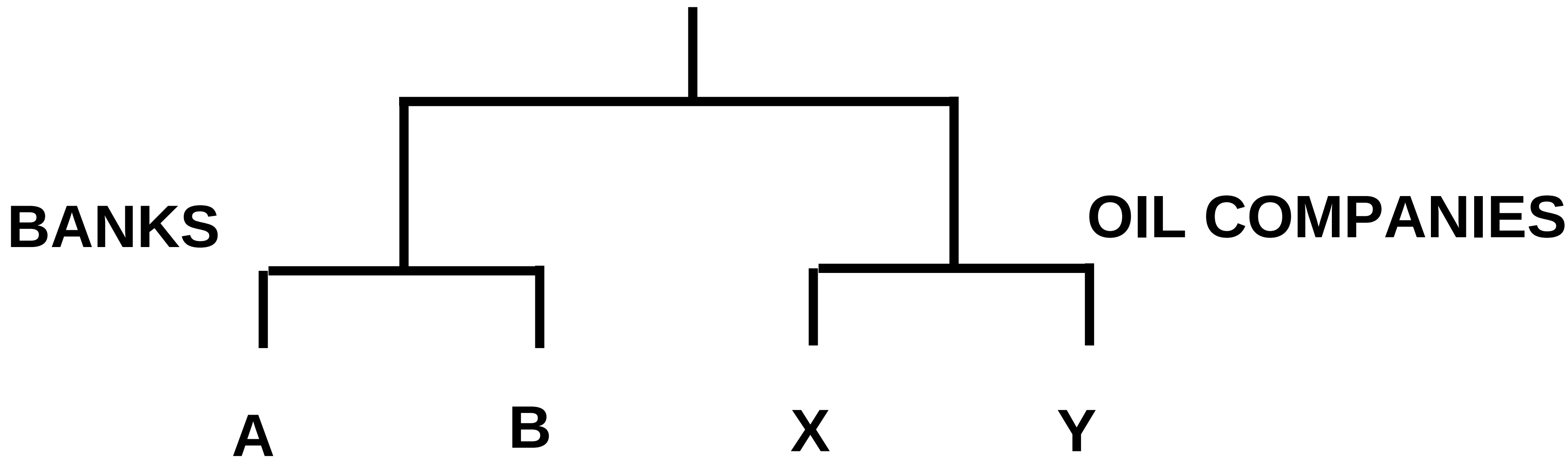
LIPNER'S LATTICE

LEGEND

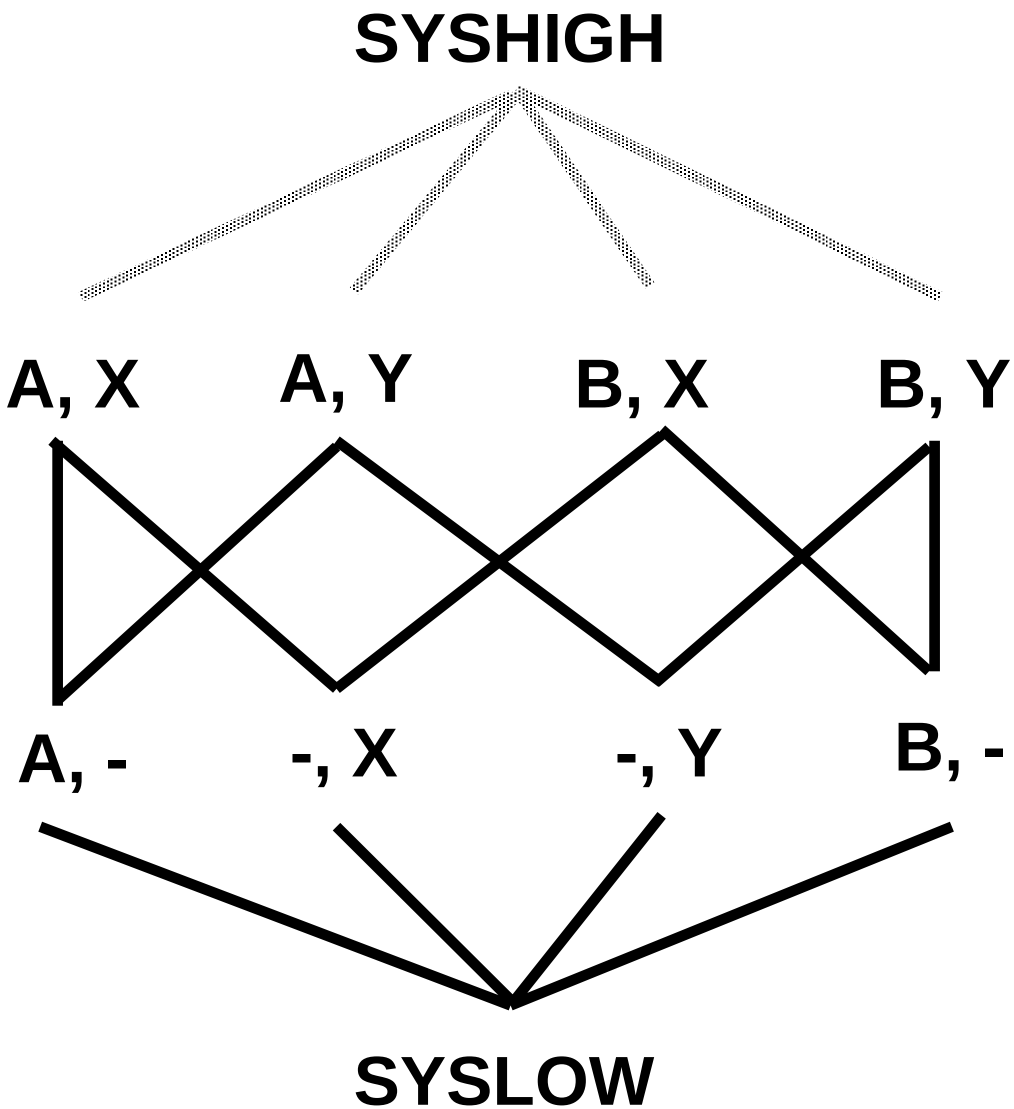
S: Subjects
O: Objects

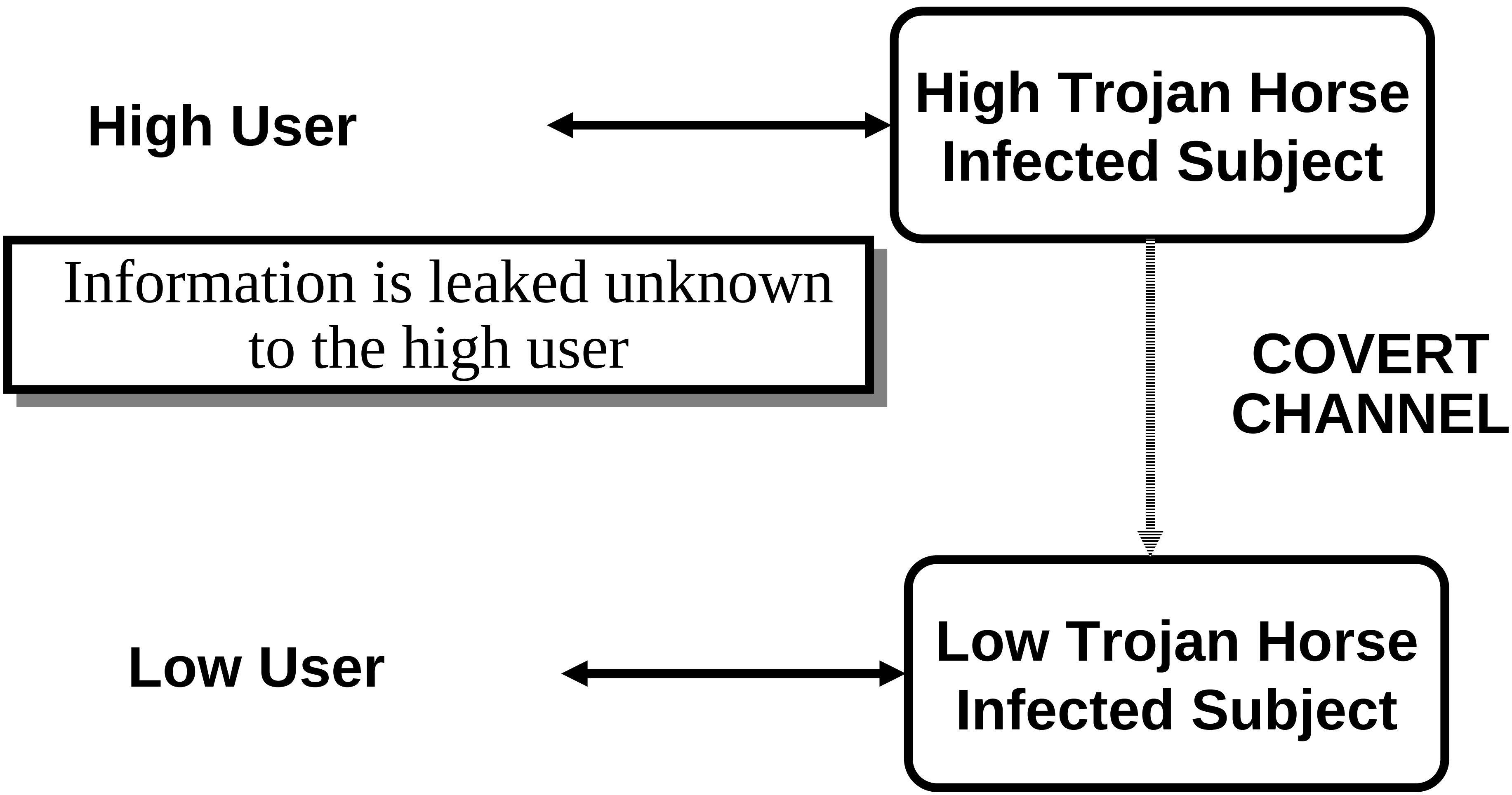


CHINESE WALL EXAMPLE



CHINESE WALL LATTICE




High User
High Trojan Horse Infected Subject
Information is leaked unknown to the high user
COVERT CHANNEL
Low User
Low Trojan Horse Infected Subject

MAC/LBAC Summary

- LBAC fails to control covert channels
- LBAC fails to control inference and aggregation
- It is too rigid for most commercial applications
- It has strong mathematical foundations

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

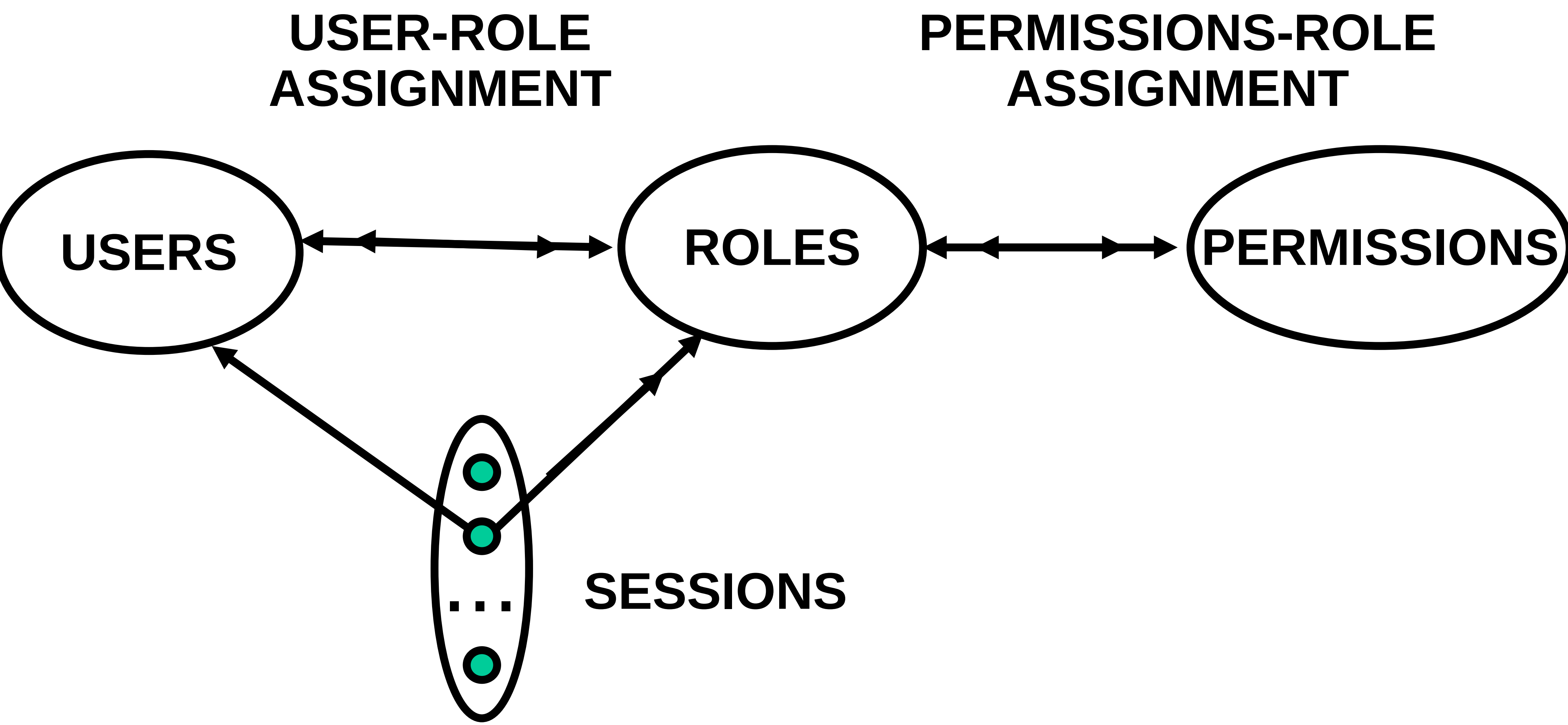
Is RBAC MAC or DAC or neither?

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

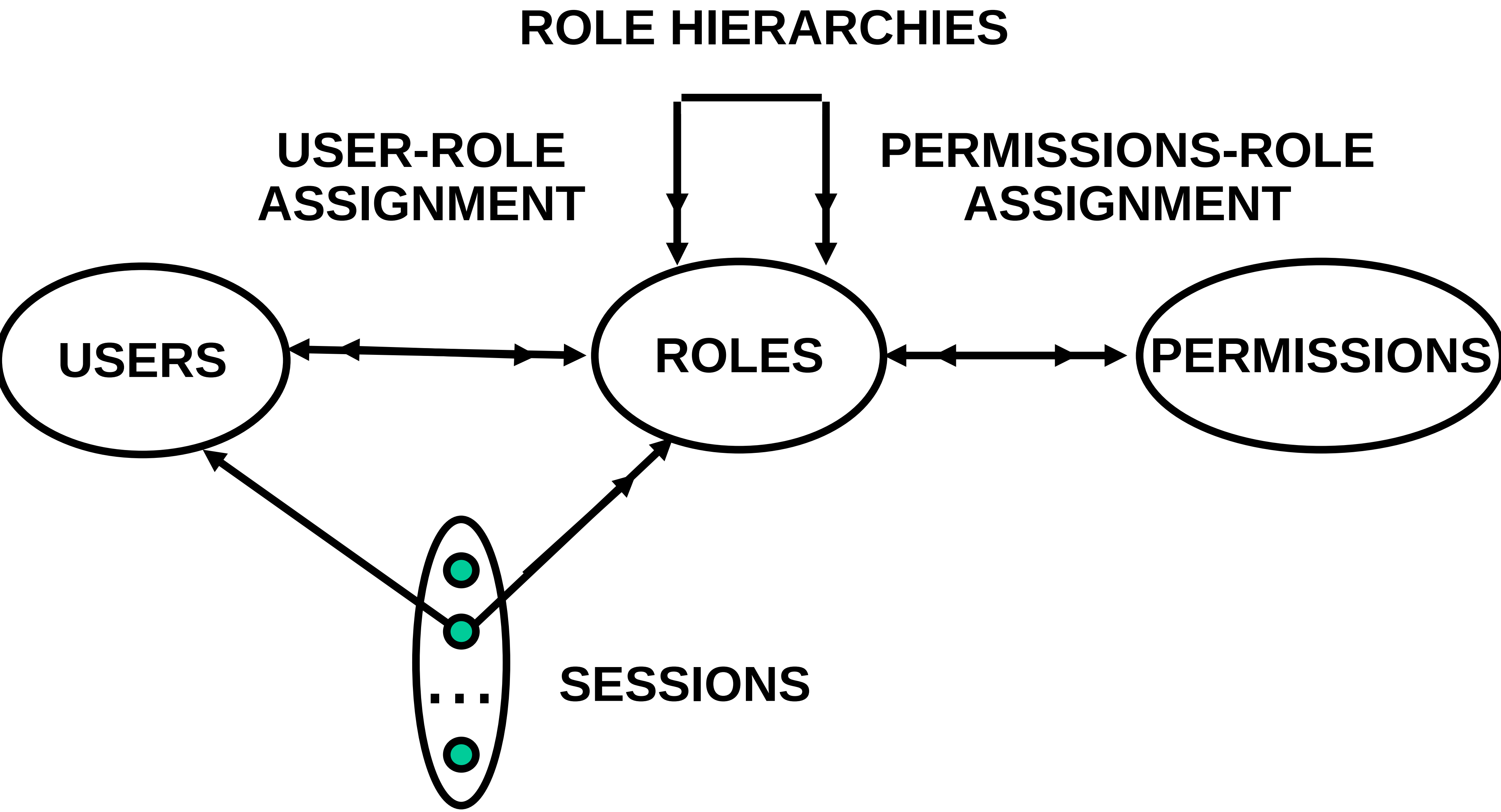
Fundamental Theorem of RBAC

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

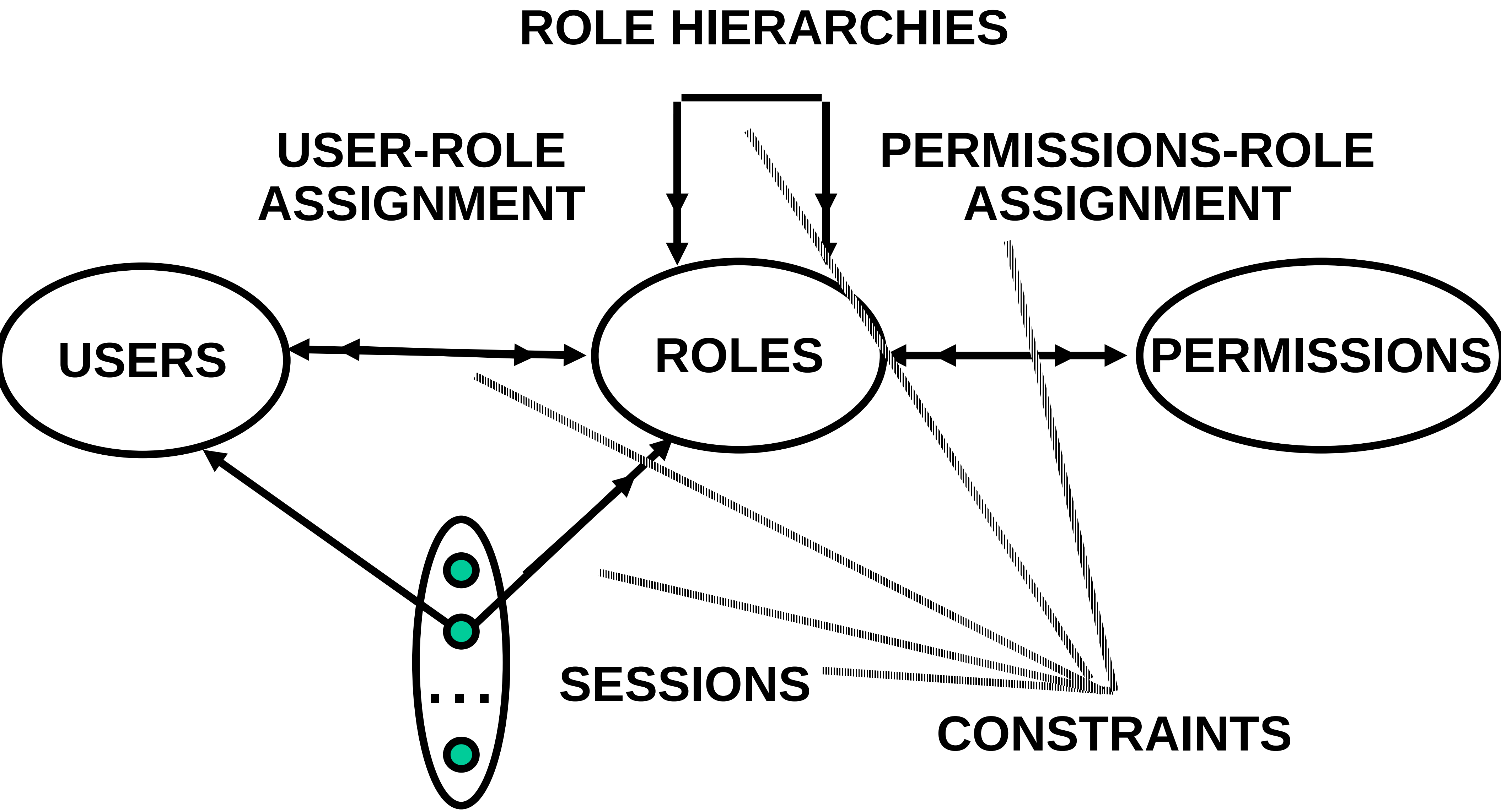
RBAC is neither MAC nor DAC!


USER-ROLE
ASSIGNMENT
PERMISSIONS-ROLE
ASSIGNMENT
USERS
ROLES
PERMISSIONS
. . .
SESSIONS

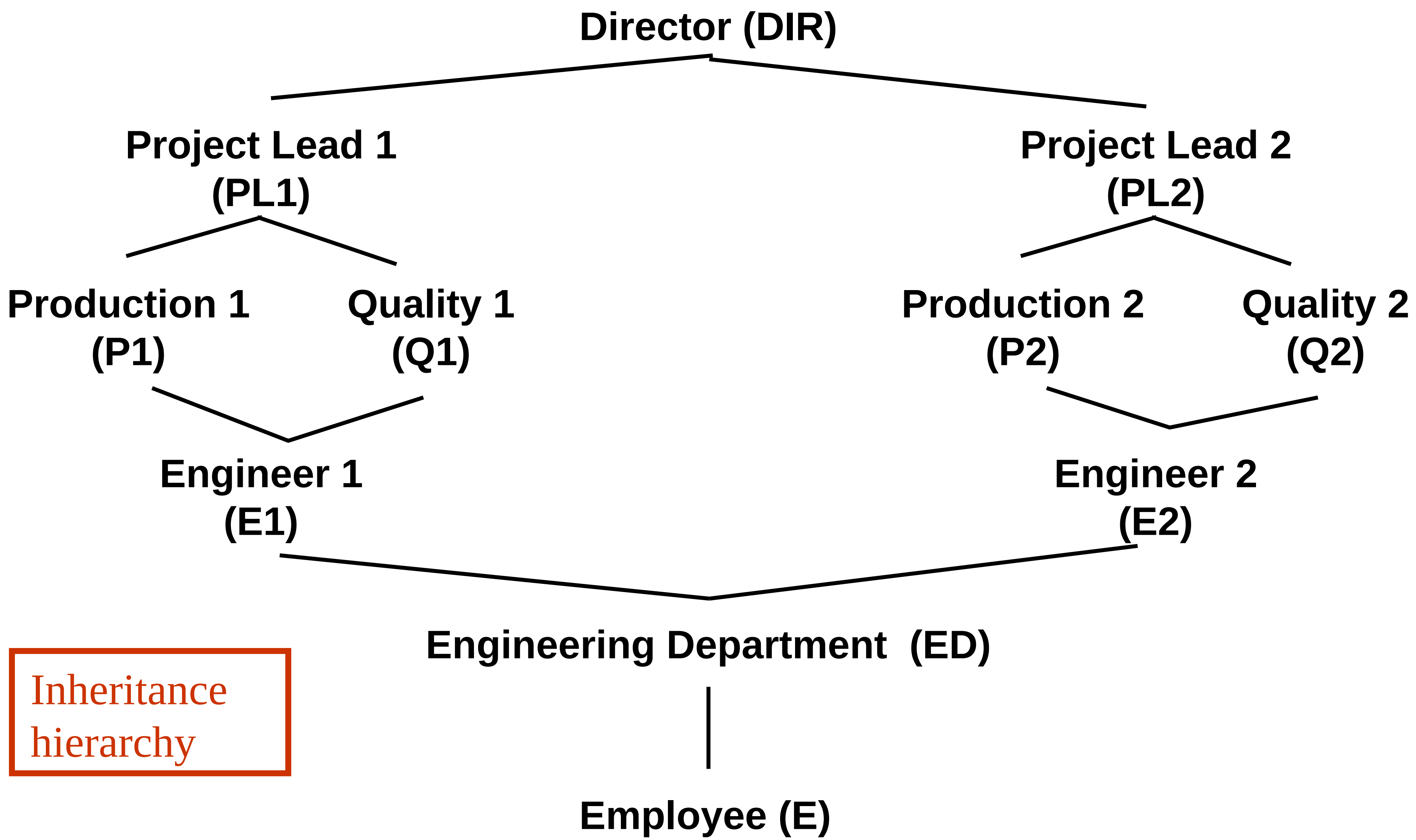
RBAC96 Model



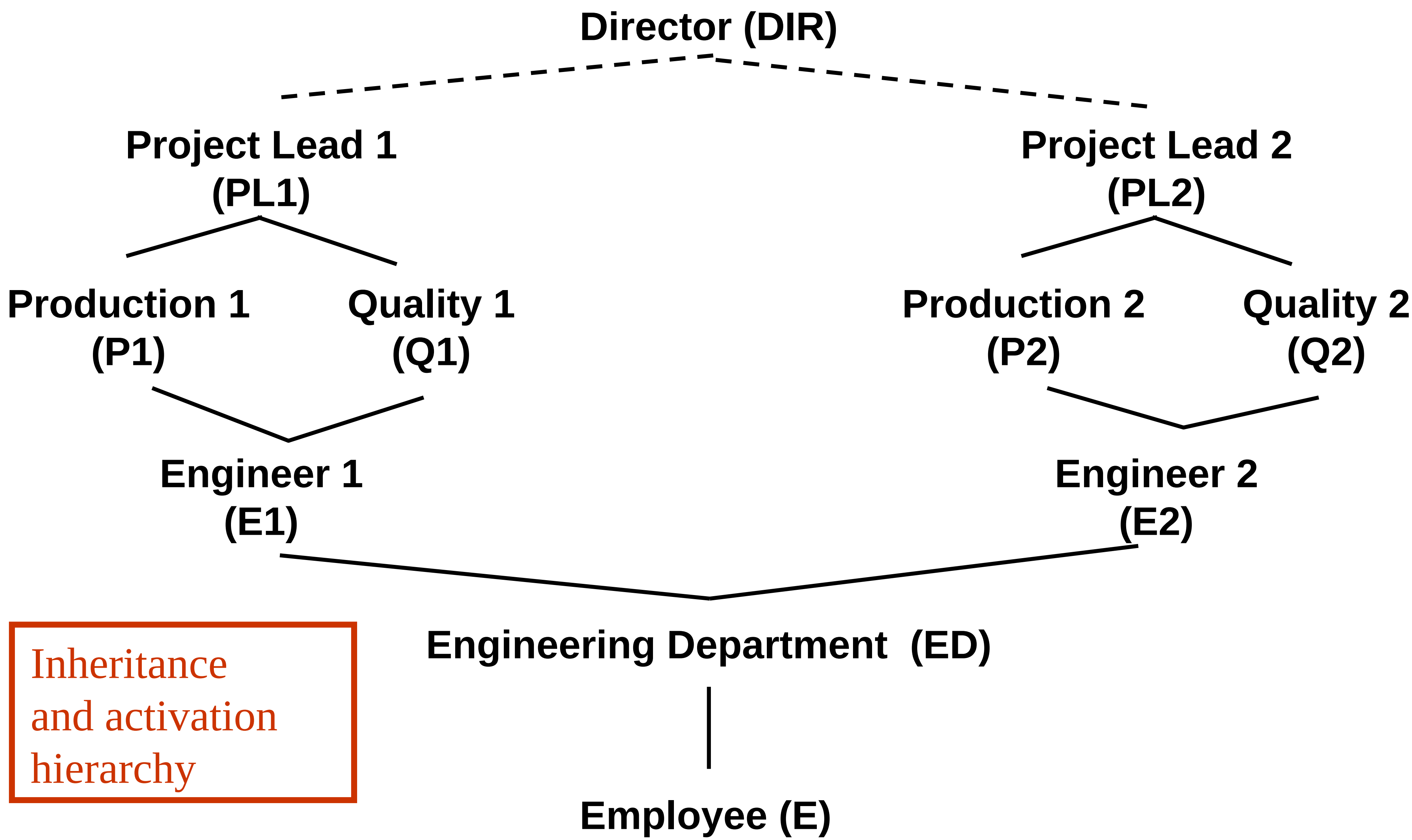
RBAC96 Model



Example Role Hierarchy

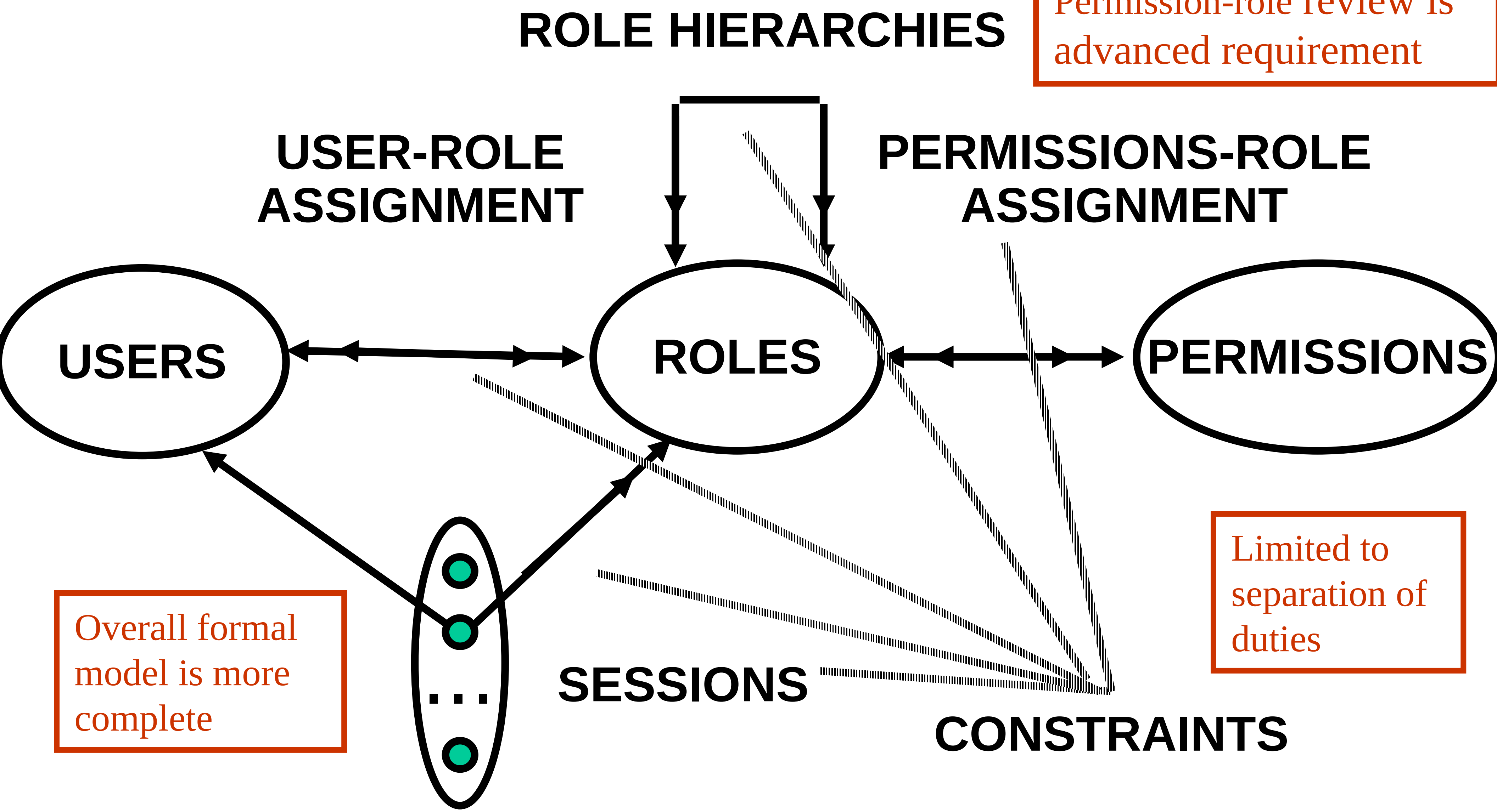


Example Role Hierarchy



NIST/ANSI RBAC Standard Model 2004

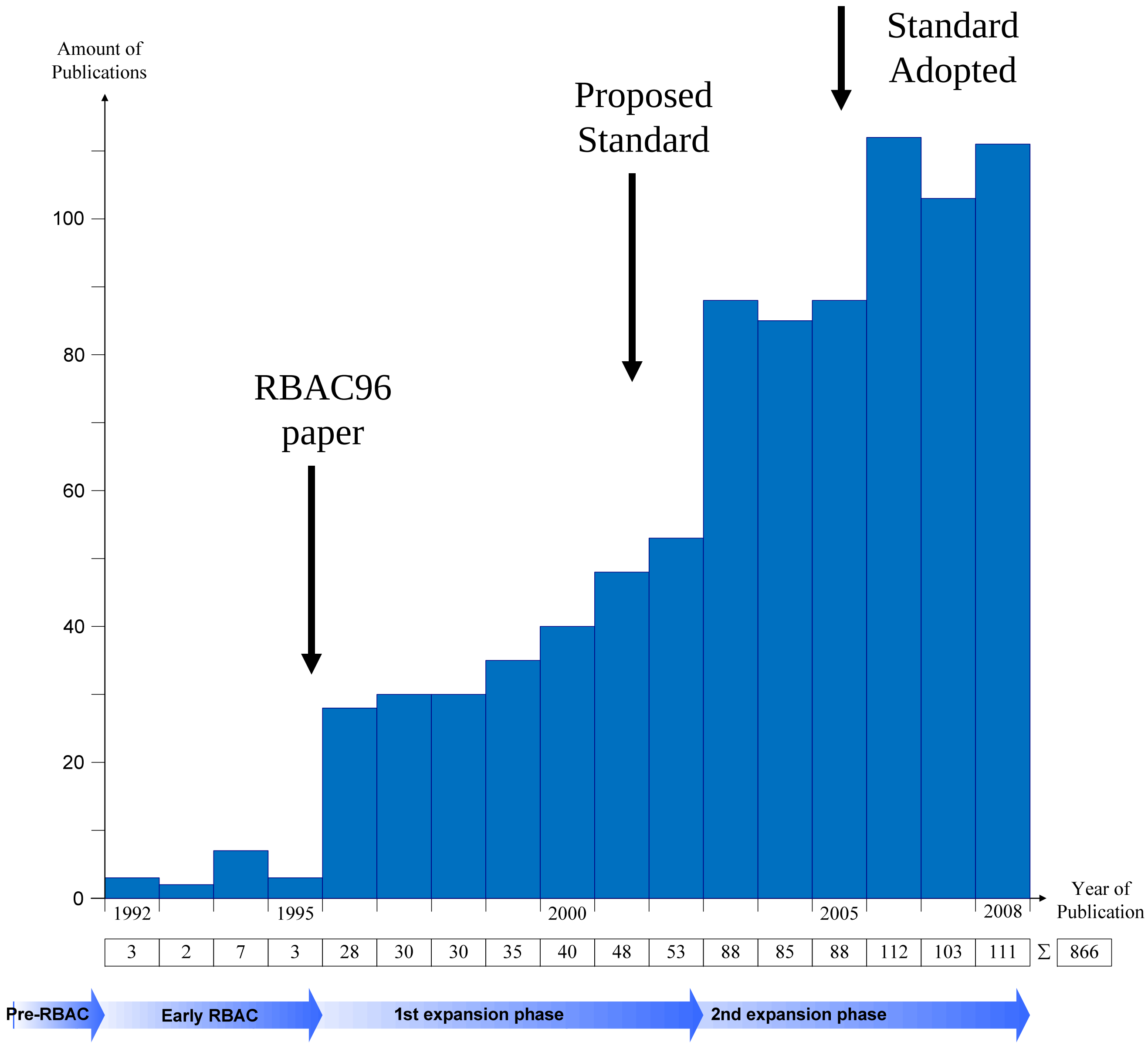
Permission-role review is advanced requirement



Limited to separation of duties

Overall formal model is more complete

The RBAC Story



1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	Σ
3	2	7	3	28	30	30	35	40	48	53	88	85	88	112	103	111	866

THE PRESENT

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
- Attribute-Based Access Control (ABAC)
  - Access based on attributes, to possibly include roles, security labels and whatever

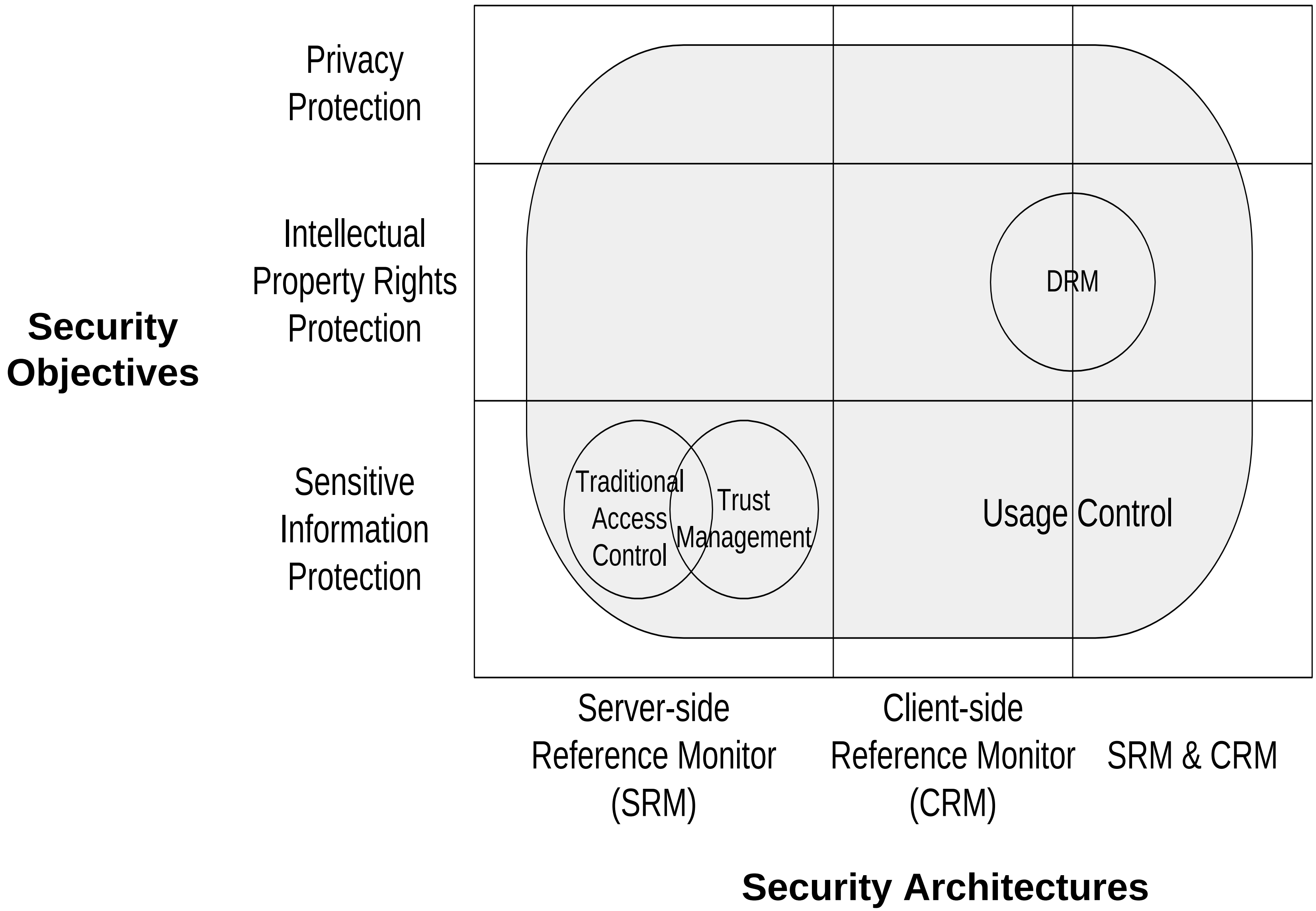
Founding Principles of RBAC96

- **Abstraction** of Privileges
  - Credit is different from Debit even though both require read and write
- **Separation** of Administrative Functions
  - Separation of user-role assignment from role-permission assignment
- **Least Privilege**
  - Right-size the roles
  - Don't activate all roles all the time
- **Separation of Duty**
  - Static separation: purchasing manager versus accounts payable manager
  - Dynamic separation: cash-register clerk versus cash-register manager

ASCAA Principles for Future Access Control

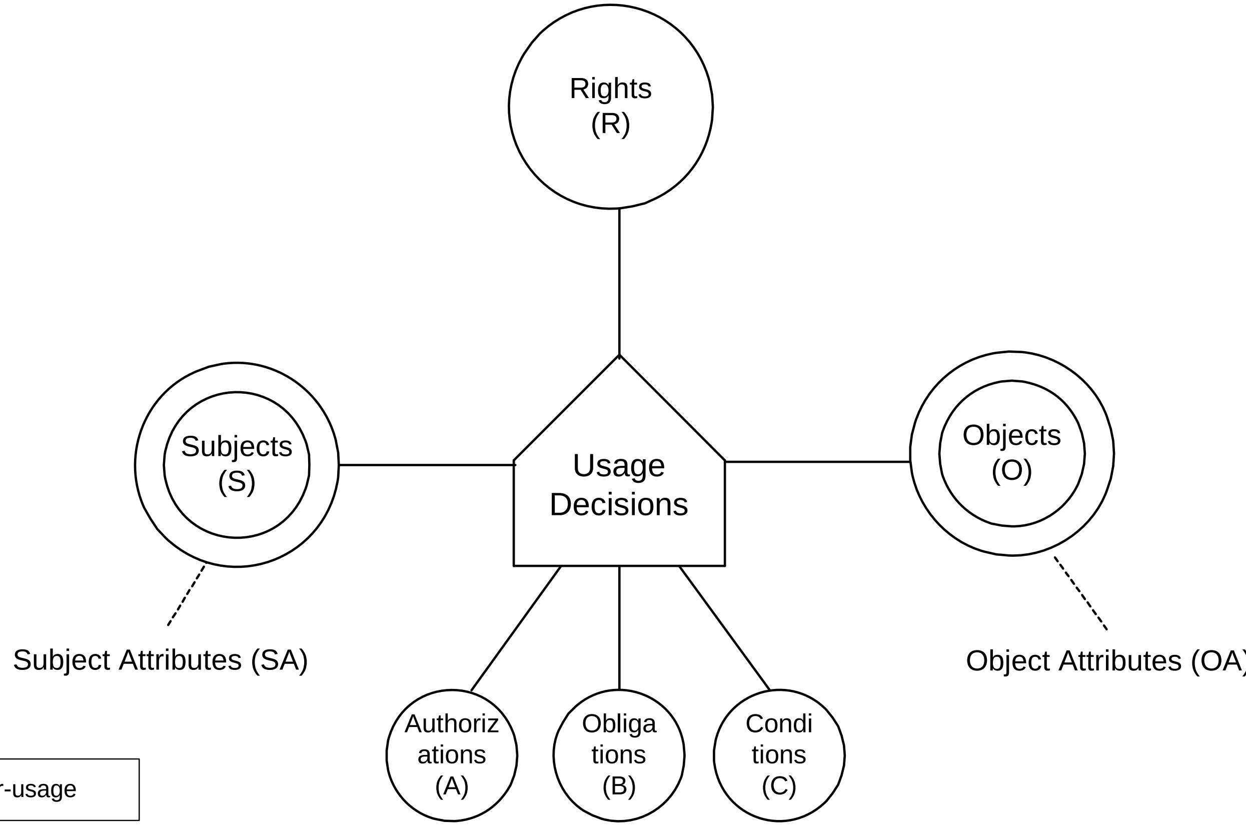
- **Abstraction** of Privileges
  - Credit vs debit
  - Personalized permissions
- **Separation** of Administrative Functions
- **Containment**
  - Least Privilege
  - Separation of Duties
  - Usage Limits
- **Automation**
  - Revocation
  - Assignment: (i) Self-assignment, (ii) Attribute-based
  - Context and environment adjustment
- **Accountability**
  - Re-authentication/Escalated authentication
  - Click-through obligations
  - Notification and alerts

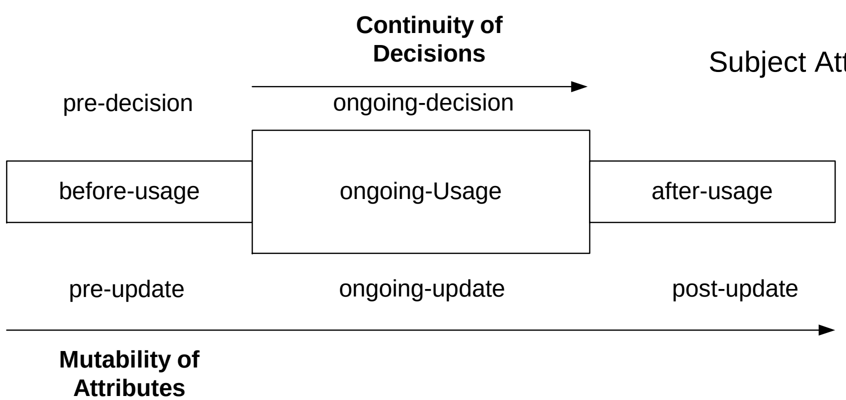
Usage Control Scope



Usage Control Model (UCON)

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





Usage Control Model (UCON)

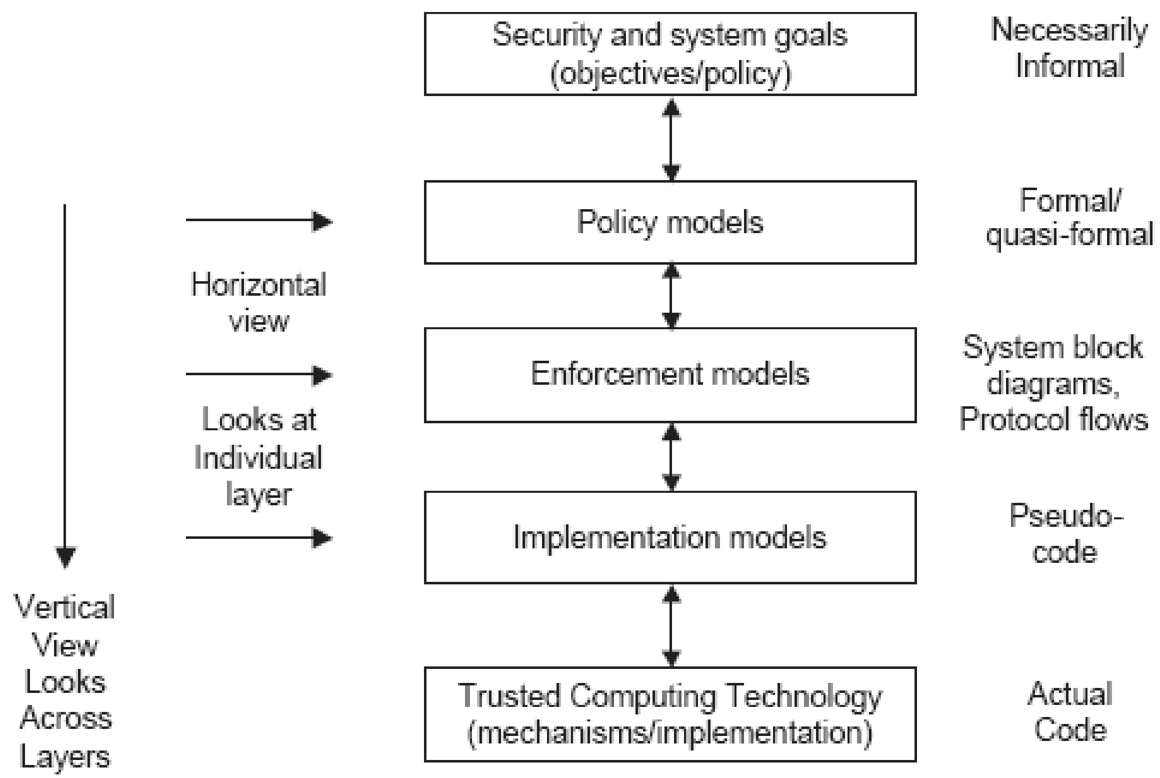
- DAC
- LBAC
- RBAC
- ABAC
- … and many, many others
- UCON
  - ABAC on steroids
  - Simple, familiar, usable and effective use cases demonstrate the need for UCON
    - Automatic Teller Machines
    - CAPTCHAs at Public web sites
    - End User Licencse Agreements
    - Terms of Usage for WiFi in Hotels, Airports
    - Rate limits on call center workers

THE FUTURE

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: 3 Layers/5 Layers



Idealized

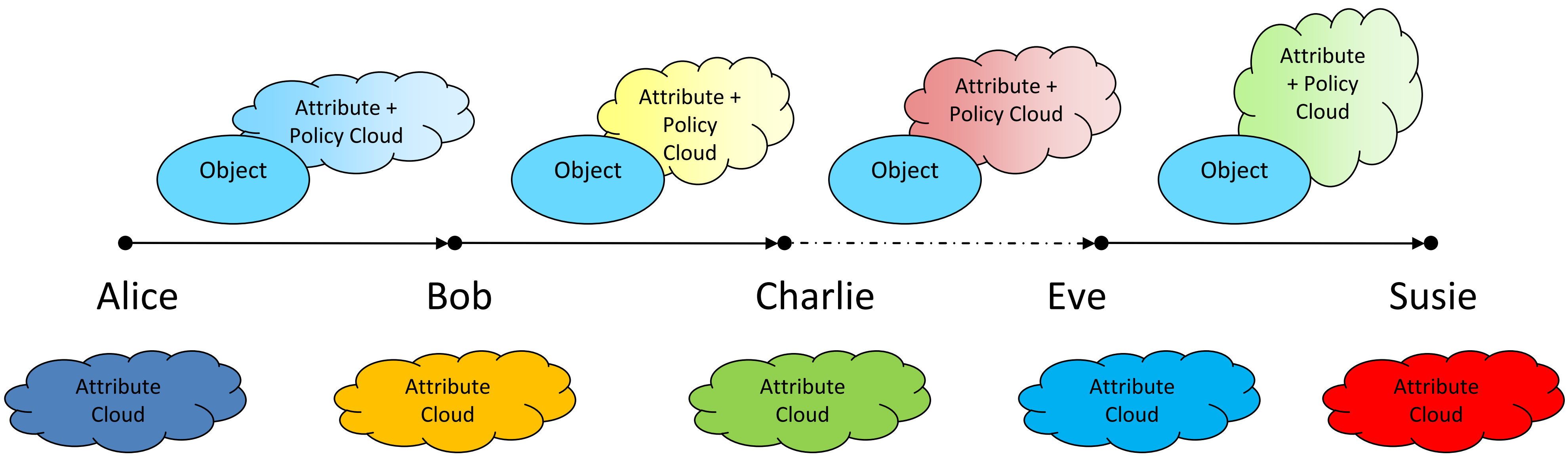
Enforceable (Approximate)

Codeable

This lecture is focused on the policy models layer

Dissemination-Centric Sharing

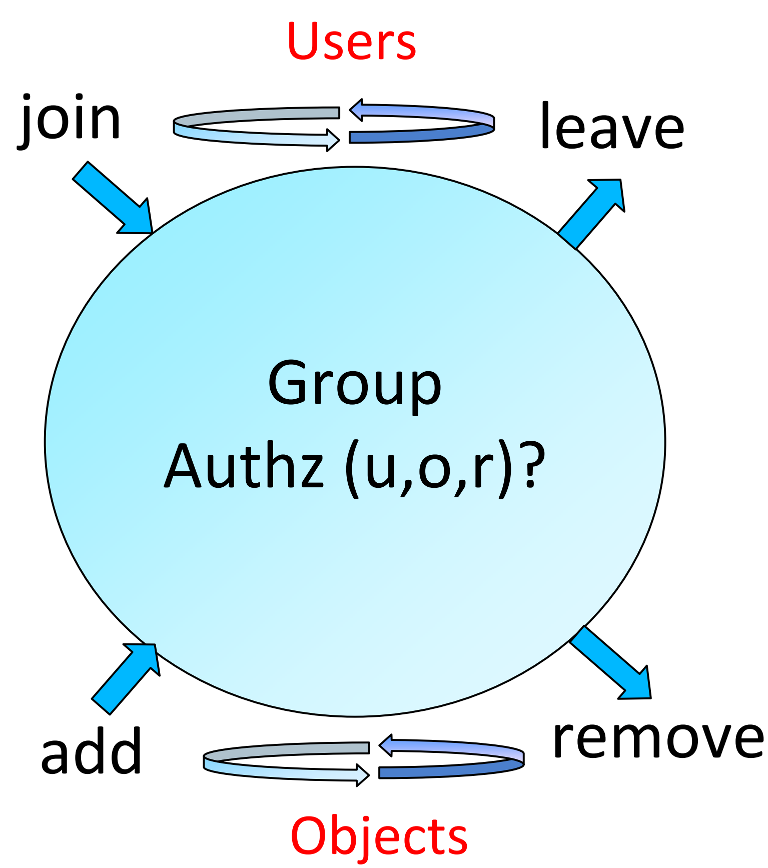
- Extensive research in the last two decades
  - ORCON, DRM, ERM, XrML, ODRL, etc.
- Copy/usage control has received major attention
- Manageability problem largely unaddressed



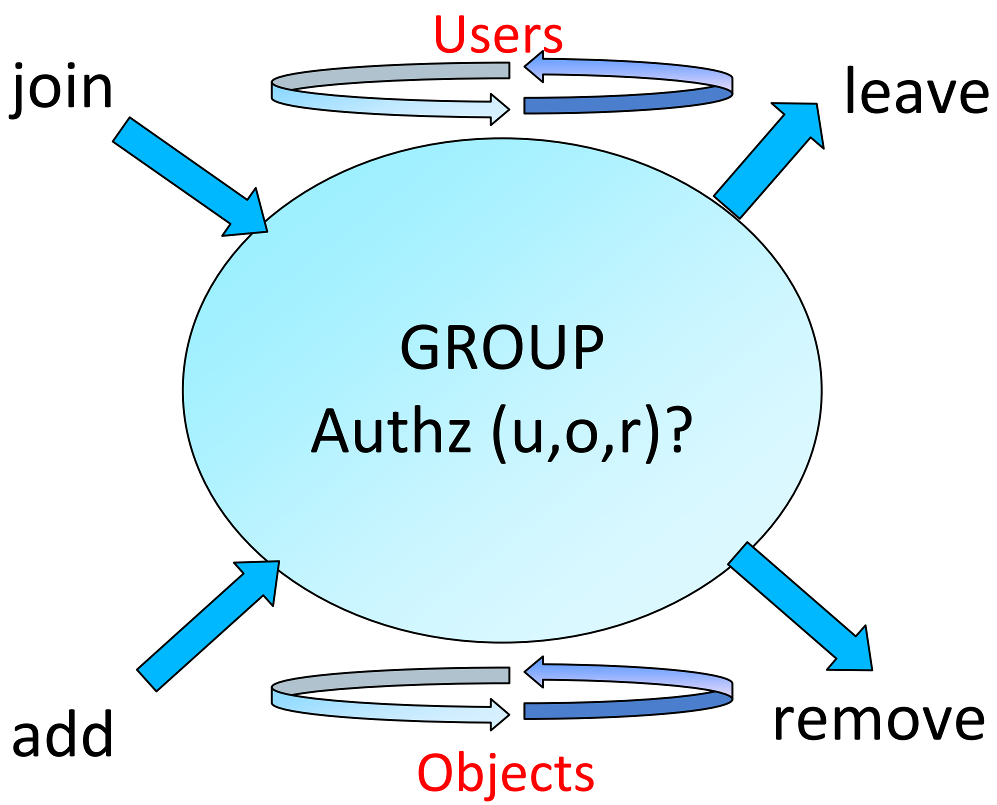
Dissemination Chain with Sticky Policies on Objects

Group-Centric Sharing (g-SIS)

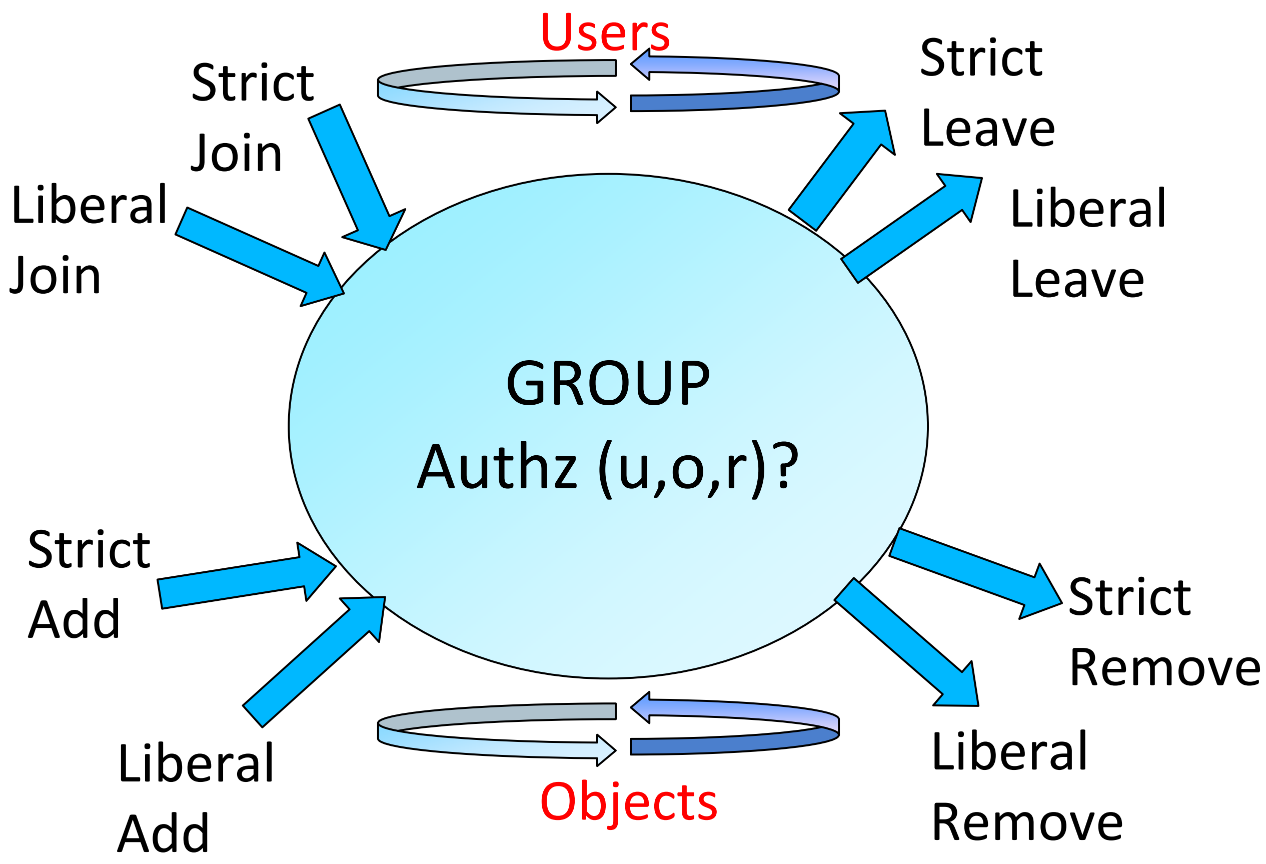
- Brings users & objects together in a group
  - Focuses on manageability using groups
  - Co-exists with dissemination-centric
  - Two metaphors
    - Secure Meeting Room (E.g. Program committee)
    - Subscription Model (E.g. Secure multicast)
- Operational aspects
  - Group characteristics
    - E.g. Are there any core properties?
  - Group operation semantics
    - E.g. What is authorized by join, add, etc.?
  - Read-only Vs Read-Write
- Administrative aspects
  - E.g. Who authorizes join, add, etc.?
  - May be application dependant
- Multiple groups
  - Inter-group relationship



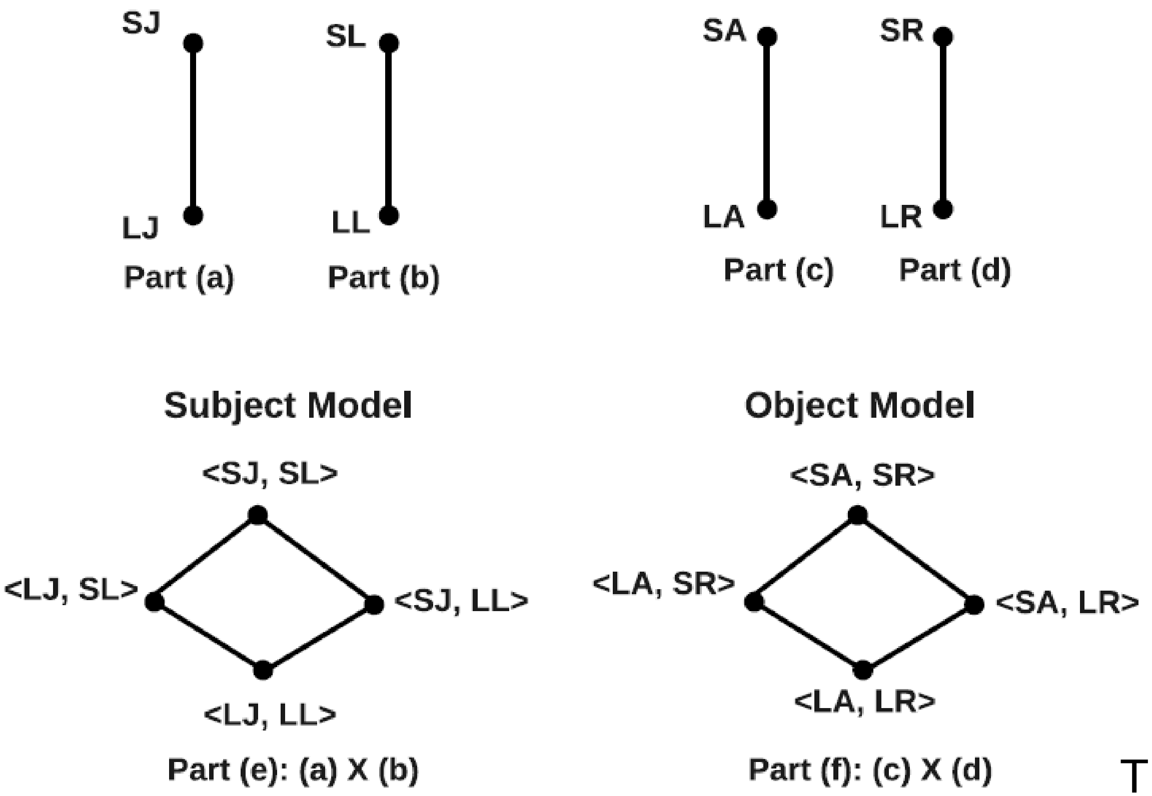
g-SIS Operation Semantics



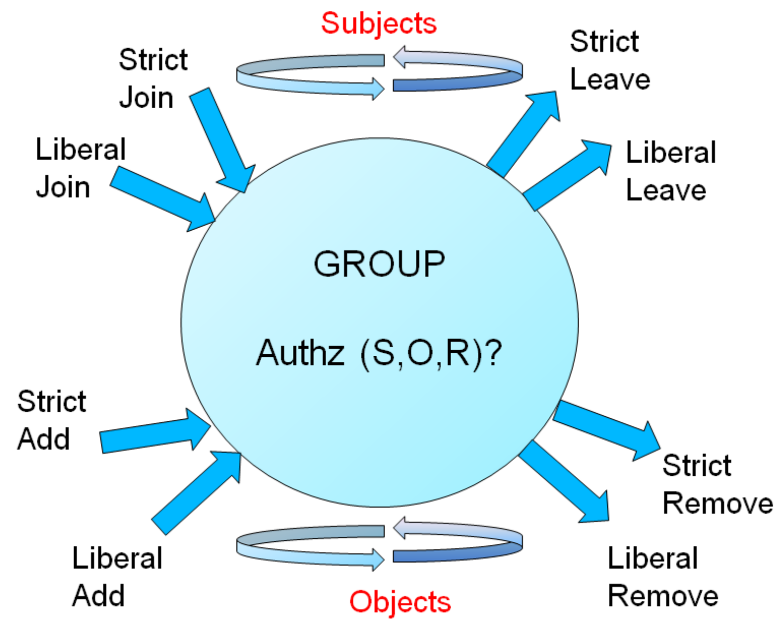
g-SIS Operation Semantics



Family of g-SIS Policy Models



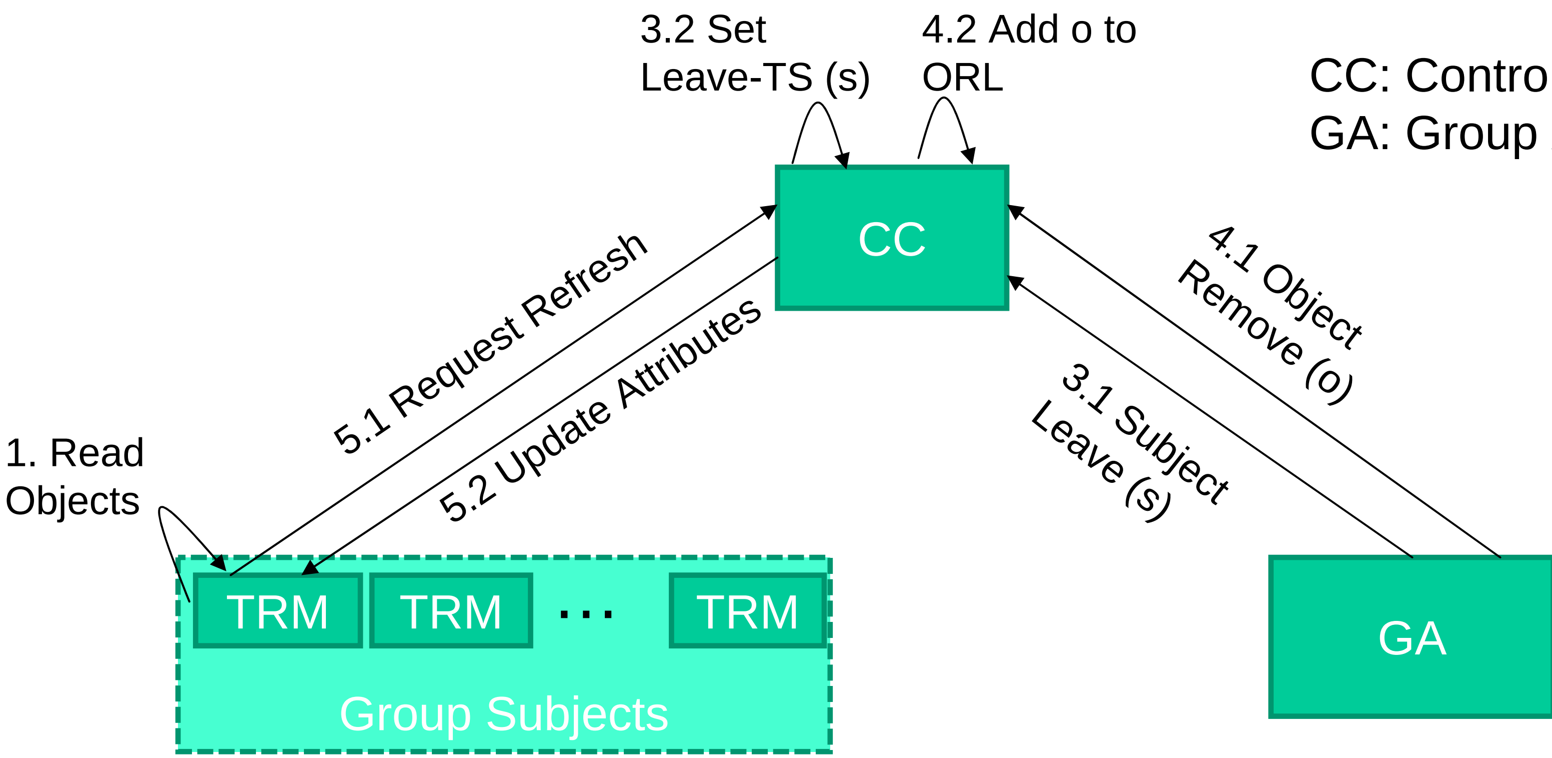
g-SIS models: (e) X (f)



Traditional Groups: <LJ, SL, LA, SR>
Secure Multicast: <SJ, LL, LA, *>

Most Restrictive g-SIS Specification: $\Box(\text{Authz} \leftrightarrow (\neg \text{SR} \wedge \neg \text{SL})\ \mathcal{S}\ (\text{SA} \wedge (\neg \text{SL}\ \mathcal{S}\ \text{SJ})))$

g-SIS Enforcement Model



CC: Control Center
GA: Group Administrator

Subject Attributes: {id, Join-TS, Leave-TS, ORL, gKey}

ORL: *Object Revocation List*

gKey: *Group Key*

Object Attributes: {id, Add-TS}

Refresh Time (RT): TRM contacts CC to update attributes

From Policy to Enforcement

- Additional Trusted/Semi-Trusted Servers
- Approximate Enforcement

- Finally, the Implementation layer models spell out protocol details and details of TRM algorithms

CONCLUSION

Conclusion

THE PAST

- Discretionary Access Control (DAC)
- Mandatory Access Control (MAC)
  - Equivalently Lattice-Based Access Control (LBAC)
- Role-Based Access Control (RBAC)

THE PRESENT

- Usage Control (UCON)
  - Attribute-Based Access Control (ABAC) on steroids

THE FUTURE

- Application-Centric Access Control Models
- Technology-Centric Access Control Models

Models are all important
A Policy Language is not a substitute for a good model