



THE UNIVERSITY OF BRITISH COLUMBIA

Access Control

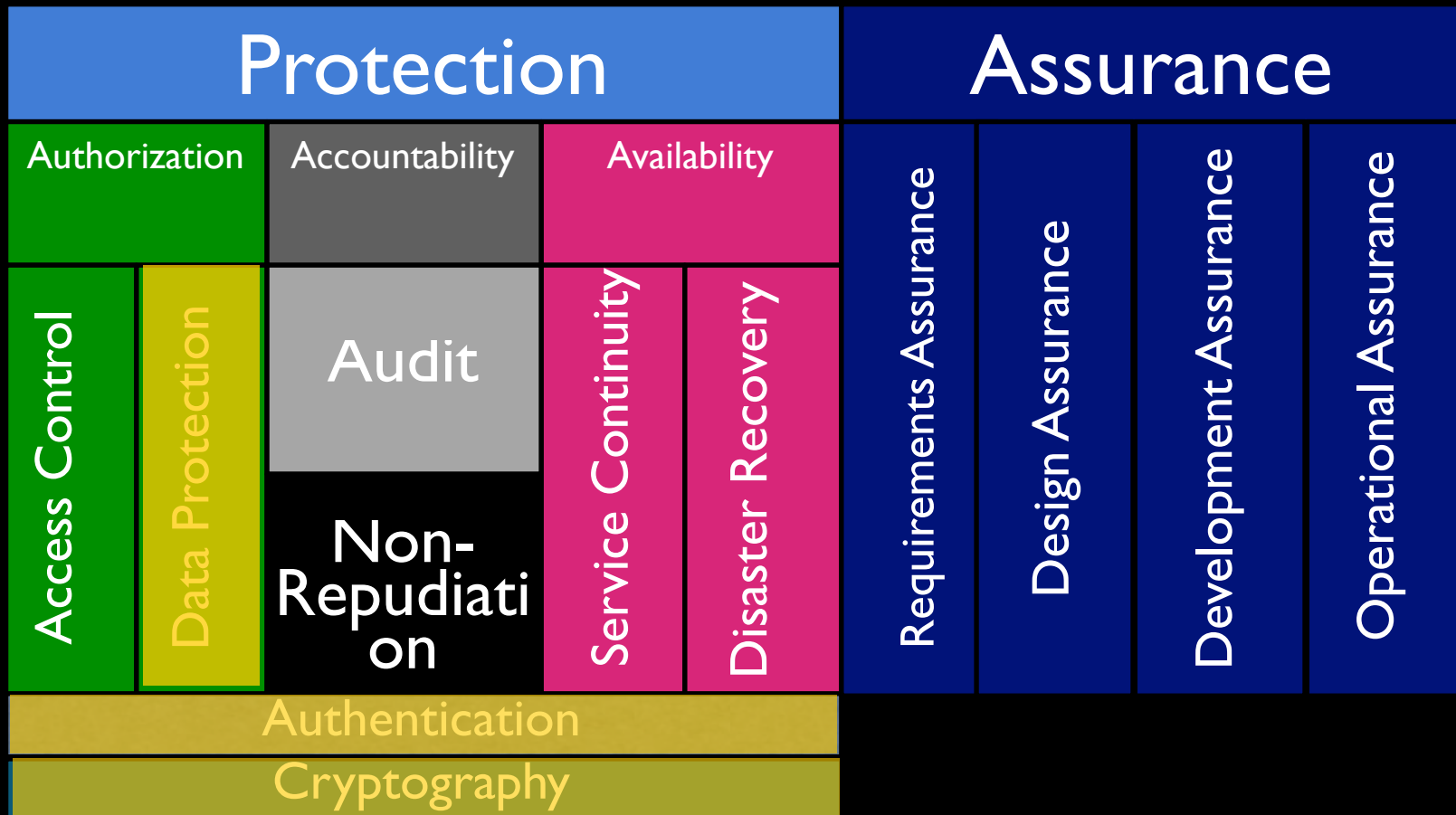
EECE 412

learning objectives

you should be able to

- explain confidentiality and integrity in terms of security policies
- explain c-lists and ACLs and differences between the two
- explain main access control poly models (BLP, CW, RBAC, DAC)
- convert a policy from one model to another

Where We Are





Ross
Anderson

“If you say that your problem can be solved with cryptography, then you don't understand your problem and you don't understand cryptography.”



Roger
Needham

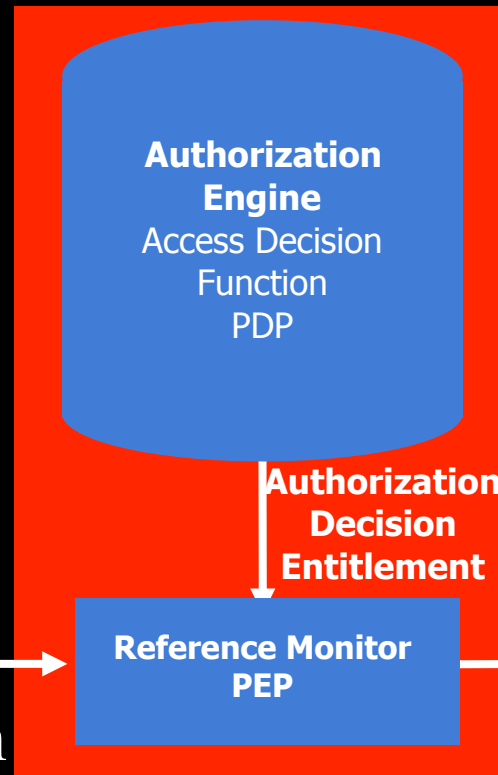
Authorization Mechanisms: Access Control

Definition: **enforces the rules,
when rule check is
possible**

Subject
Principal
User, Client
Initiator



Action



Security
Subsystem



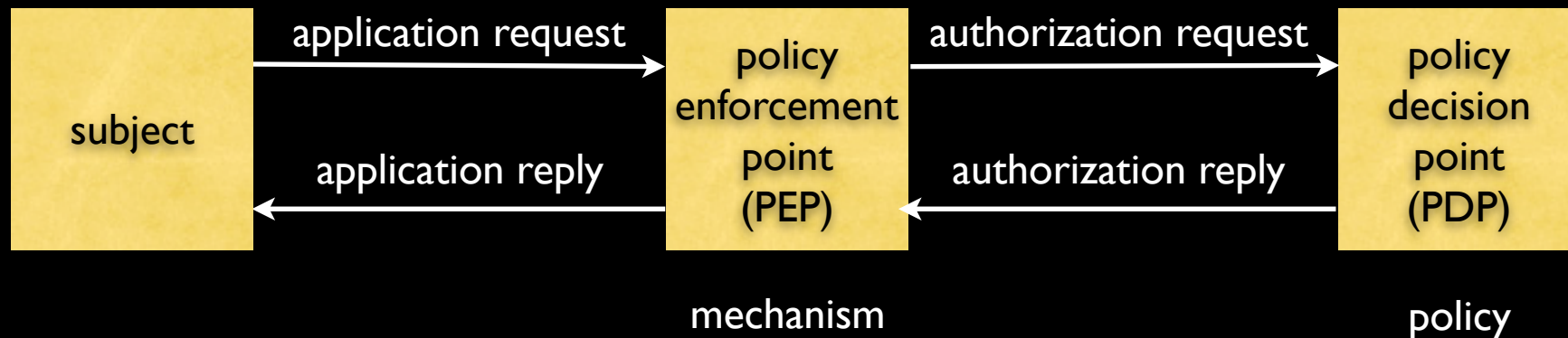
Mix of terms:

Authorization == Access Control Decision

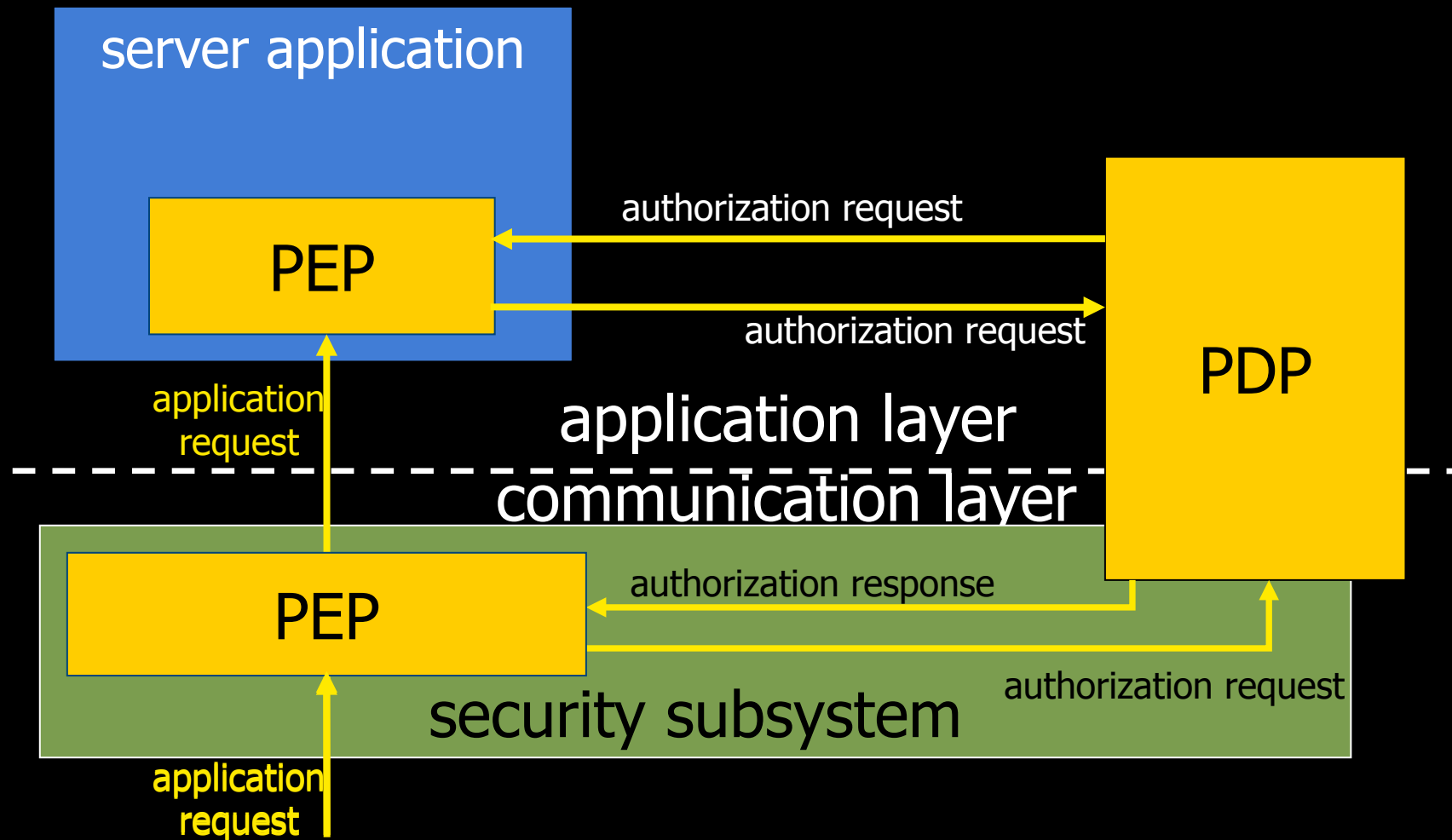
Authorization Engine == Policy Engine

policies and mechanisms

- Policies describe what is allowed
- Mechanisms control how policies are enforced



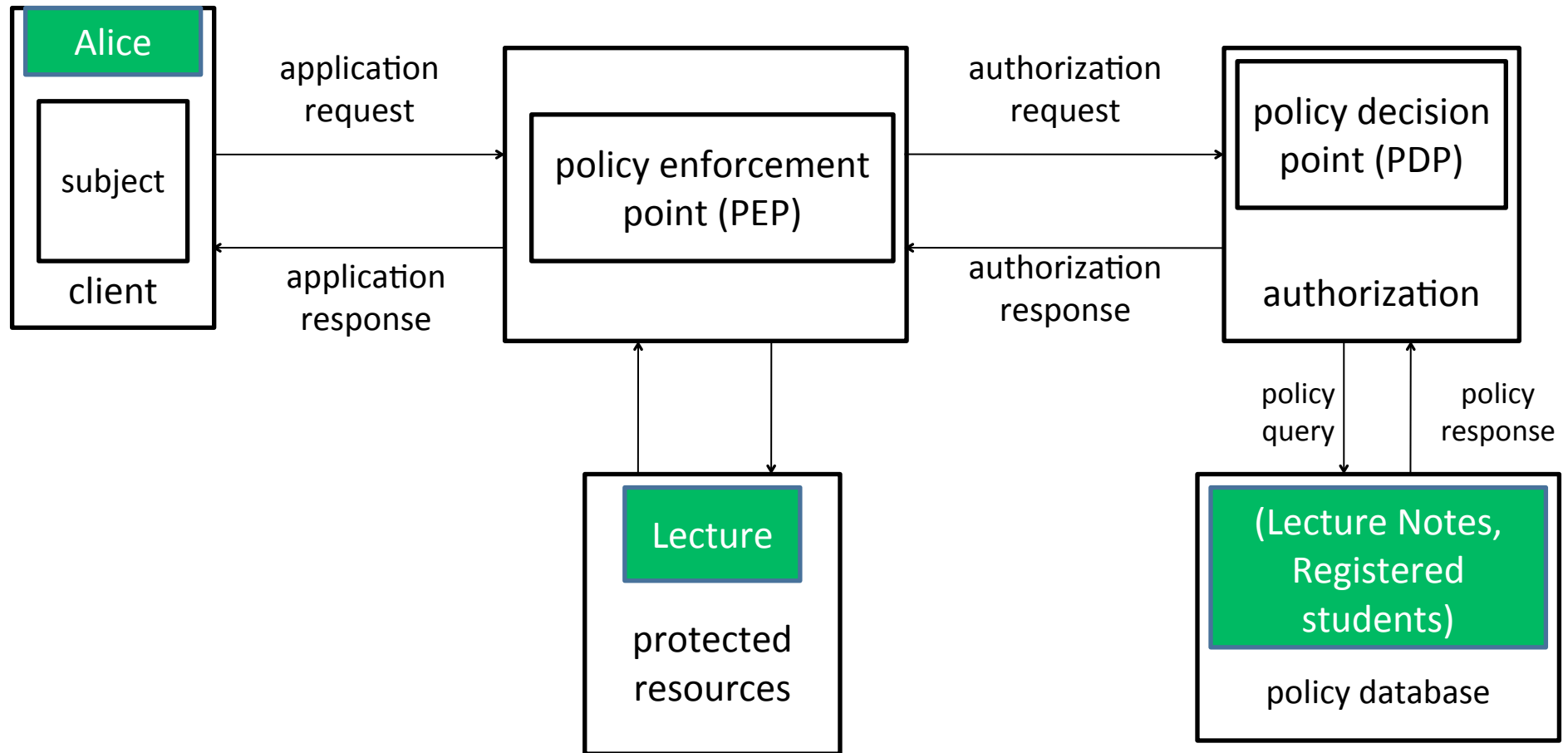
request-response paradigm



case study of research @
LERSSE:

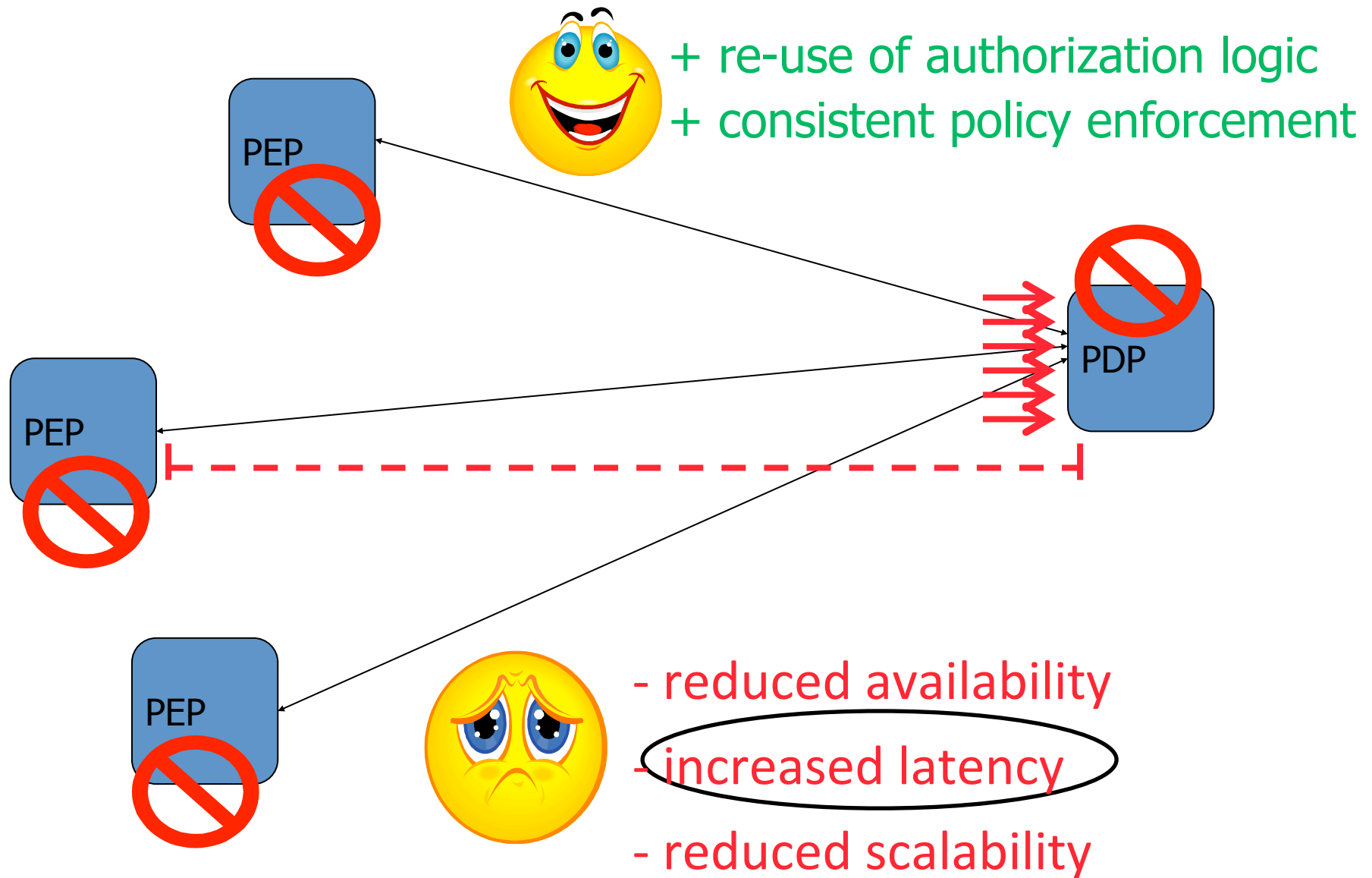
improving performance and
availability of enterprise
authorization architectures

authorization architecture

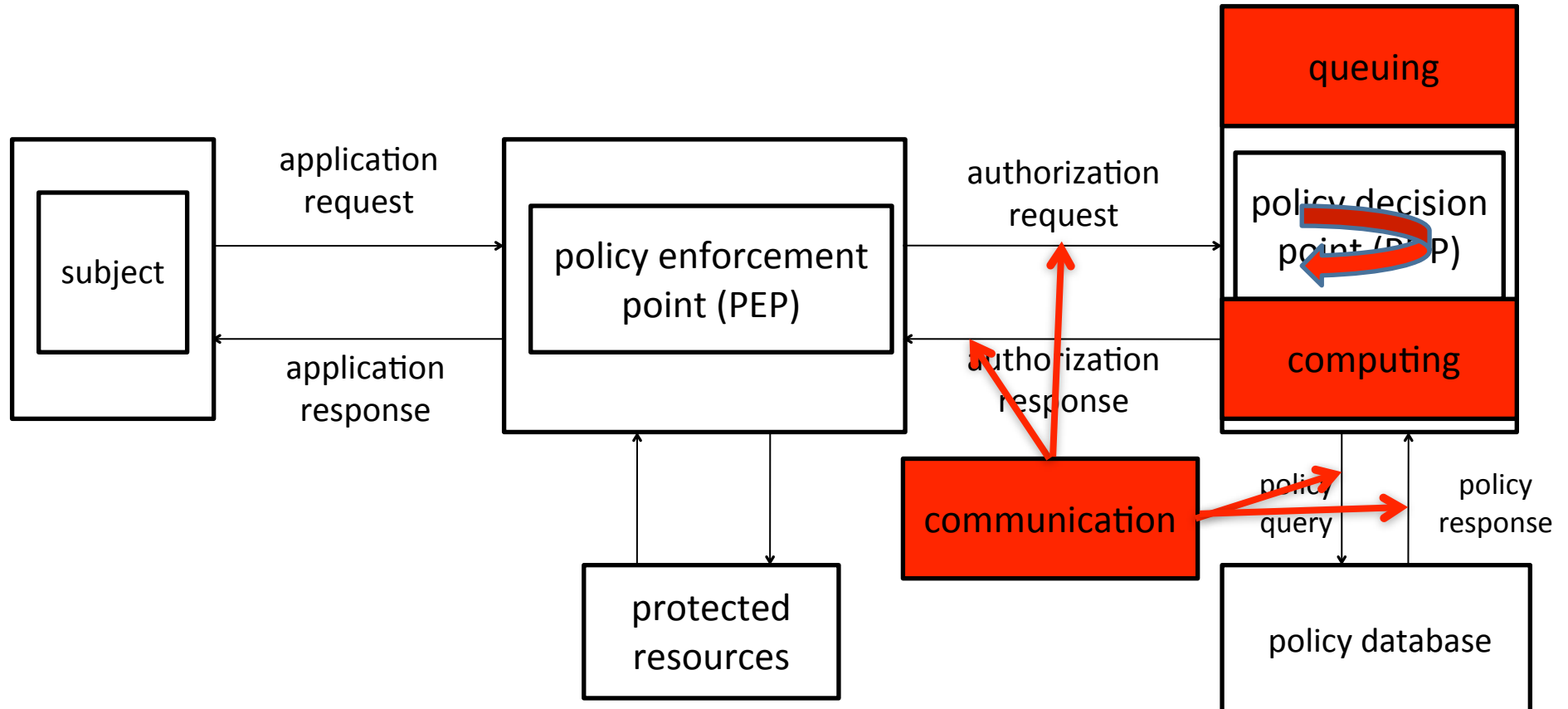


- also known as the request-response model
- used by IBM Access Manager, Entrust GetAccess, CA SiteMinder

pros and cons



problem – authorization latency



authorization latency is the sum of the three delays

existing approaches

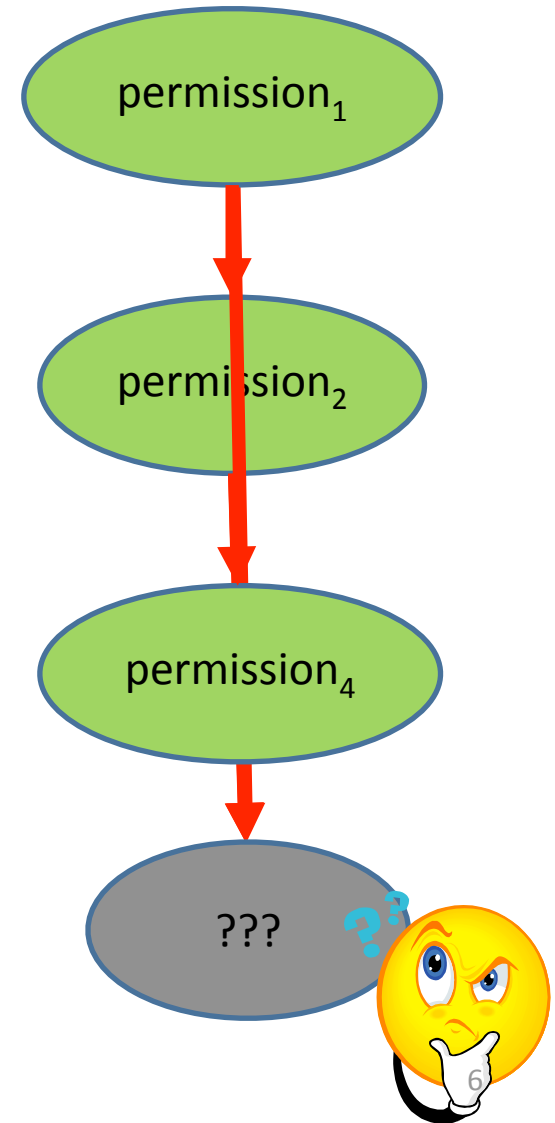
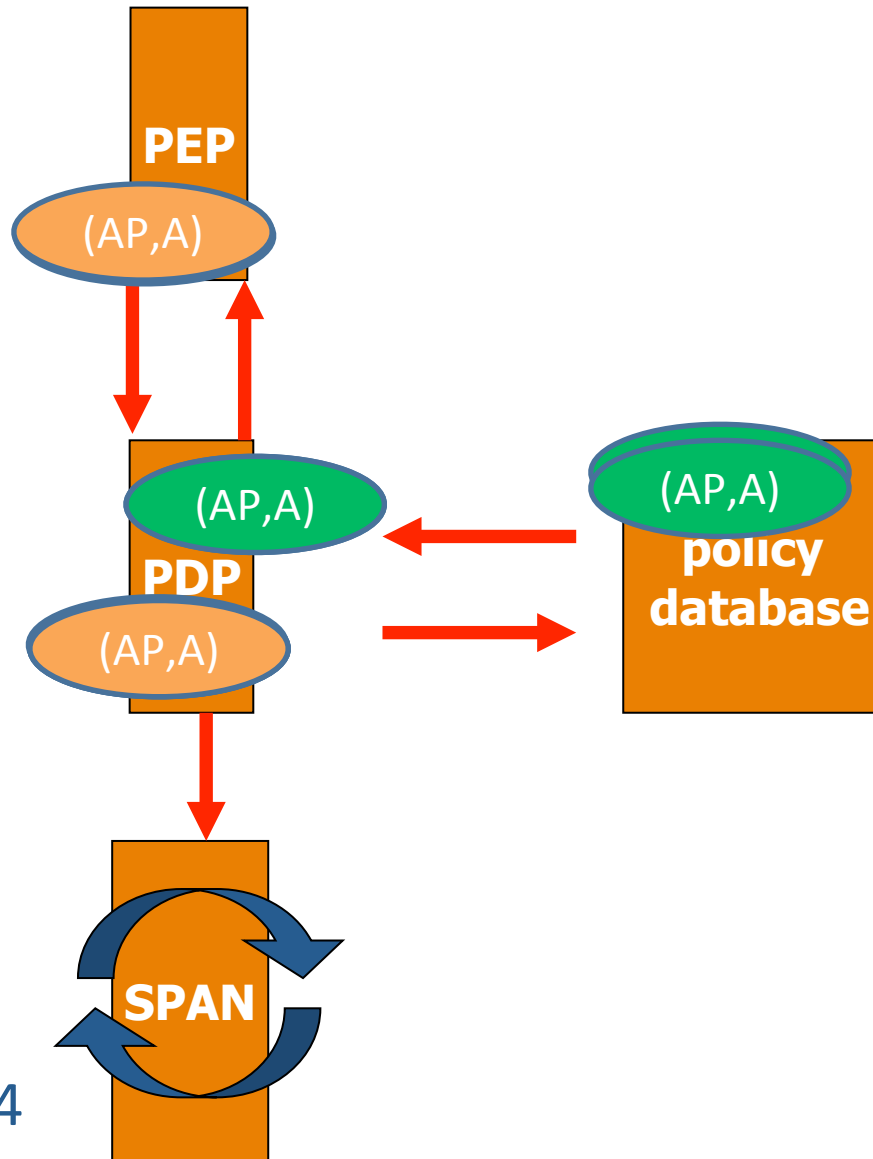
- group replication
 - + reduces the queuing delays
 - require specialized OS/middleware
 - poorly scale on large populations
 - communication delays still exist
- caching previous authorizations
 - + simple, inexpensive
 - + improves overall latency
 - serves only returning requests
- SAAM and its variants [1, 2]
 - + improve availability and performance
 - delay incurred for computing responses remains unchanged
 - designed for policies that are defined using the BLP model.

•[1]Crampton J, Leung W, and Beznosov. K Secondary and approximate authorizations model and its application to Bell-LaPadula policies. In Proceedings of the 11th ACM Symposium on Access Control Models and Technologies SACMAT'06), pages 111-120, Lake Tahoe, CA, USA, June 7-9 2006. ACM Press

•[2]Wei Q, "Towards Improving the Availability and Performance of Enterprise Authorization Systems," PhD dissertation, Department of Electrical and Computer Engineering, THE UNIVERSITY OF BRITISH COLUMBIA, October, 2009

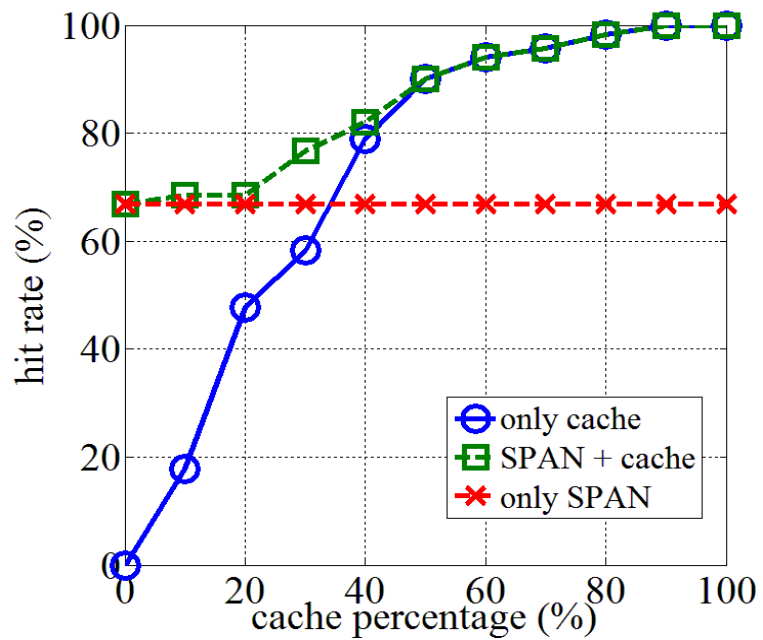
13

speculative authorization (SPAN)

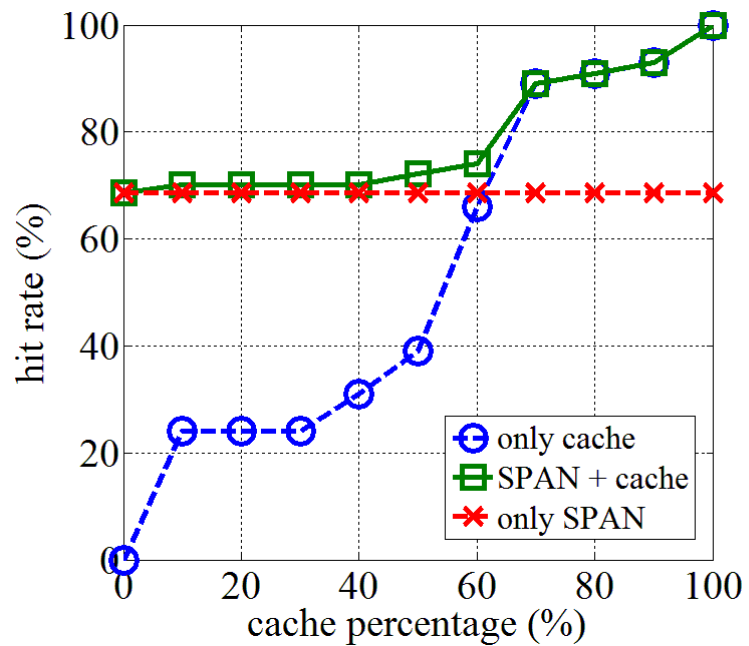


14

caching and SPAN in same system



WebCT



FC



Access Matrix

Lampson's Access Control Matrix

Subjects (users) index the rows

Objects (resources) index the columns

| | OS | Accounting program | Accounting data | Insurance data | Payroll data |
|--------------------|-----|--------------------|-----------------|----------------|--------------|
| Bob | rx | rx | r | --- | --- |
| Alice | rx | rx | r | rw | rw |
| Sam | rwX | rwX | r | rw | rw |
| Accounting program | rx | rx | rw | rw | rw |

why access matrix is not used

- **Access control matrix** has all relevant info
- But how to manage a large access control (AC) matrix?
- Could be 1,000's of users, 1,000's of resources
- Then AC matrix with 1,000,000's of entries
- Need to check this matrix before access to any resource is allowed
- Hopelessly inefficient

Access Control Lists

- ACL: store access control matrix by **column**
- Example: ACL for **insurance data** is in **yellow**

| | OS | Accounting program | Accounting data | Insurance data | Payroll data |
|--------------------|-----|--------------------|-----------------|-----------------------|--------------|
| Bob | rx | rx | r | --- | --- |
| Alice | rx | rx | r | rw | rw |
| Sam | rwX | rwX | r | rw | rw |
| Accounting program | rx | rx | rw | rw | rw |

example: MacOS X

```
beznosov — bash — 106x27
Last login: Fri Oct 19 10:26:56 on ttys000
beznosov@Konstantin-Beznosovs-MacBook-Pro-2.local:~> ls -l
total 24
drwx-----@  5 beznosov  staff    170  7 Jun 17:00 Applications
drwx-----@ 25 beznosov  staff    850 17 Oct 18:54 Data
drwx-----@  5 beznosov  staff    170  6 Jul 03:53 Desktop
drwx-----@ 25 beznosov  staff    850 13 Jan 2010 Documents
drwx-----@ 375 beznosov  staff  12750 21 Oct 20:52 Downloads
drwx-----@ 26 beznosov  staff    884  9 Oct 22:04 Dropbox
drwxr-xr-x@  70 beznosov  staff   2380  9 Oct 22:04 Google Drive
drwxr-xr-x  134 beznosov  staff   4556 13 Aug 15:08 GoogleDocs
drwxrwxr-x   5 beznosov  staff    170 20 Jul 2011 Incompatible Software
drwxr-xr-x@  3 beznosov  staff    102 26 Mar 2009 InstallShield
drwx-----@ 82 beznosov  staff   2788 27 Jul 00:55 Library
drwx-----@  2 beznosov  staff    68 24 Dec 2007 Login Items
drwx-----@ 31 beznosov  staff   1054 12 Mar 2012 Movies
drwx-----@  8 beznosov  staff    272  5 Apr 2009 Music
drwx-----@ 10 beznosov  staff    340  3 Jan 2012 Pictures
drwxr-xr-x@  7 beznosov  staff    238  8 Oct 2009 Public
drwxr-xr-x@ 20 beznosov  staff    680  2 Feb 2007 Sites
-rw-r--r--@  1 beznosov  staff    248 17 Dec 2008 id_rsa.pub
drwx-----@  3 beznosov  staff    102 22 Jun 2004 poseidon2
--w-----+  1 beznosov  staff     0 25 Jan 2010 test.txt
-rw-r--r--  1 beznosov  staff    277 20 Feb 2010 texput.log
drwxr-x---@ 186 beznosov  staff   6324 17 Oct 18:58 tmp
-rw-r--r--@  1 beznosov  staff    130 15 Oct 16:53 webct_upload_applet.properties
beznosov@Konstantin-Beznosovs-MacBook-Pro-2.local:~> █
```

Capabilities (or C-Lists)

- Store access control matrix by **row**
- Example: Capability for **Alice** is in **blue**

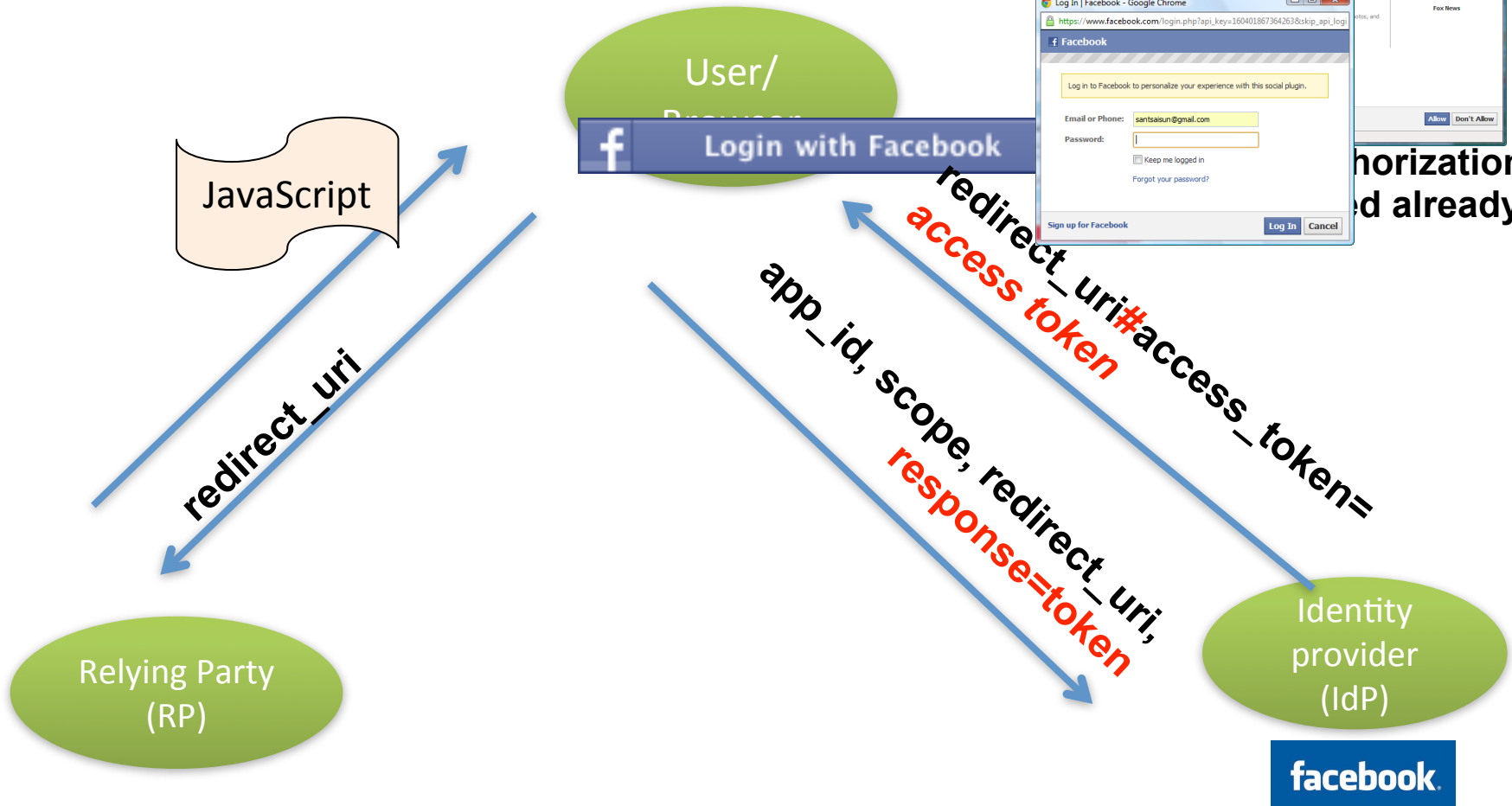
| | OS | Accounting program | Accounting data | Insurance data | Payroll data |
|--------------------|----|--------------------|-----------------|----------------|--------------|
| Bob | rx | rx | r | --- | --- |
| Alice | rx | rx | r | rw | rw |
| Sam | rx | rx | r | rw | rw |
| Accounting program | rx | rx | rw | rw | rw |

introduction →
evaluation →
recommendation →
conclusion

example: OAuth v2

```
access_token = document.location.hash
```

skip authentication if user has logged into the IdP in the same browser

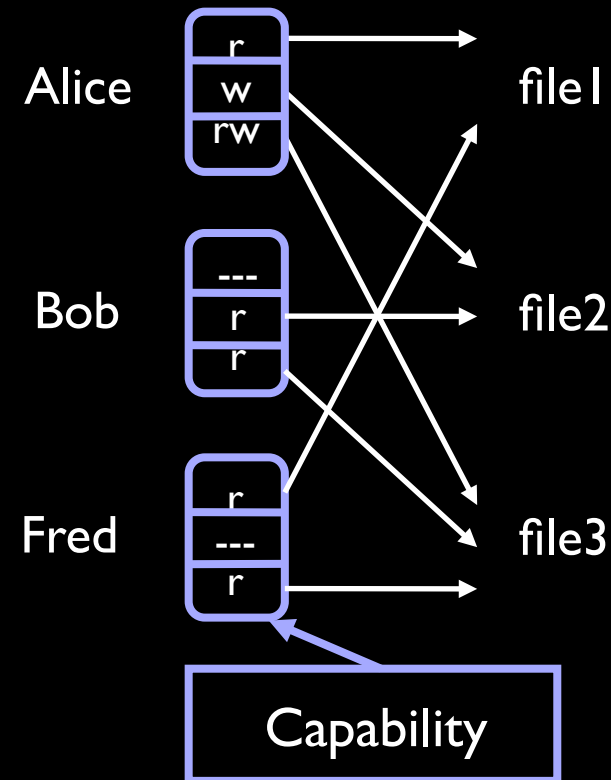
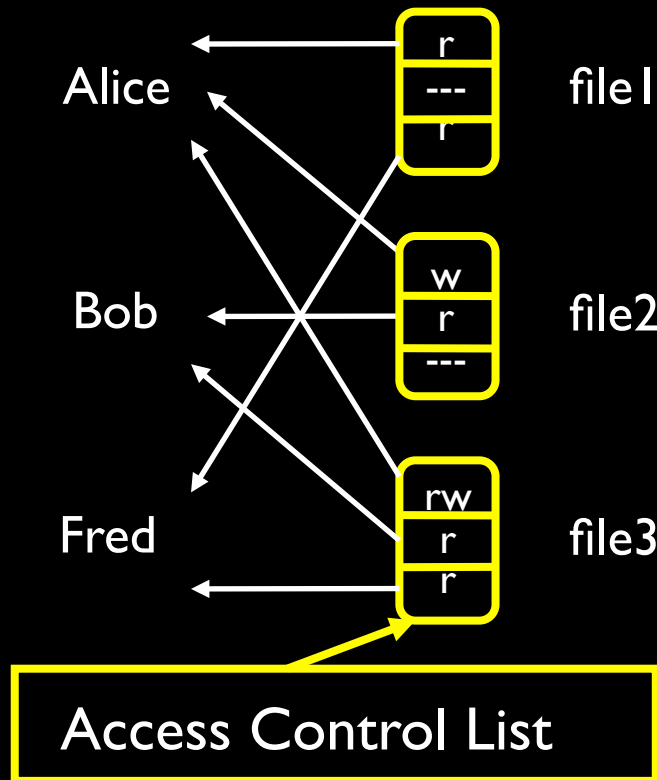


Authorization if already



ACLs

Capabilities



- Note that arrows point in opposite directions!
- With ACLs, still need to associate users to files

ACLs vs Capabilities

- ACLs
 - Good when users manage their own files
 - Protection is data-oriented
 - Easy to change rights to a resource
- Capabilities
 - Easy to delegate
 - Easy to add/delete users
 - Easier to delegate rights
 - Harder to control the delegation
 - More difficult to implement
 - The “Zen of information security”

can jana read Four-part Harmony.doc?

can jana read Four-part Harmony.doc?

The screenshot shows the 'the eXpandable grid' application window. The legend indicates that a green square means 'Allow', a red square means 'Deny', and a yellow square means 'Some access allowed'. The grid shows permissions for various users across different files. The file 'Four-part Harmony.doc' is highlighted, and its permissions are shown in a subgrid. The subgrid shows that 'jana' has 'RW' (Read and Write) permissions, while other users have different levels of access.

| File | chan | edna | henry | jana | kavita |
|-----------------------|---------------------|---------------------|-------|---------------------|--------|
| Theory 101 | Some access allowed | Some access allowed | Deny | Some access allowed | Deny |
| Admin | Some access allowed | Allow | Deny | Allow | Deny |
| Handouts | Some access allowed | Allow | Deny | Some access allowed | Deny |
| Four-part Harmony.doc | Some access allowed | Allow | Deny | RW | Deny |
| Musical Analysis1.doc | Some access allowed | Allow | Deny | Allow | Deny |
| Musical Analysis2.doc | Some access allowed | Allow | Deny | Allow | Deny |
| Ditch Training.doc | Some access allowed | Allow | Deny | Allow | Deny |

source: <http://www.robreeder.com/projects/xgrids.html>



Security Policies

what's secure system?

- **Secure system**
 - Starts in authorized state
 - Never enters unauthorized state
- If the system enters any of these states, it's a security violation
- Authorized state in respect to what?
- Policy partitions system states into:
 - Authorized (**secure**)
 - These are states the system can enter
 - Unauthorized (**nonsecure**)



THE UNIVERSITY OF BRITISH COLUMBIA

CIA

What's Confidentiality?

- X set of entities, I information
- I has **confidentiality property** with respect to X if **no $x \in X$** can obtain information from I
- I can be disclosed to others
- Example:
 - X set of students
 - I final exam answer key
 - I is confidential with respect to X if students cannot obtain final exam answer key

what's confidentiality policy?

- Goal: **prevent the unauthorized disclosure of information**
 - Deals with information flow
 - Integrity incidental
- Multi-level security models are best-known examples
 - **Bell-LaPadula Model** basis for many, or most, of these

What's Integrity?

- X set of entities, I information
- I has **integrity** property with respect to X if all $x \in X$ trust information in I
- Examples?

Types of Access Control Policies

- Discretionary Access Control (**DAC**, IBAC)
 - individual user sets access control mechanism to allow or deny access to an object
- Mandatory Access Control (**MAC**)
 - system mechanism controls access to object, and individual cannot alter that access
- Originator Controlled Access Control (**ORCON**)
 - originator (creator) of information controls who can access information

Multilevel Security (MLS) Models

Classifications and Clearances

- **Classifications** apply to **objects**
- **Clearances** apply to **subjects**
- US Department of Defense uses 4 levels of classifications/clearances

TOP SECRET

SECRET

CONFIDENTIAL

UNCLASSIFIED

Clearances and Classification

- To obtain a **SECRET** clearance requires a routine background check
- A **TOP SECRET** clearance requires extensive background check
- Practical classification problems
 - Proper classification not always clear
 - Level of granularity to apply classifications
 - Aggregation — flipside of granularity

Subjects and Objects

- Let O be an **object**, S a **subject**
 - O has a classification
 - S has a clearance
 - Security **level** denoted $L(O)$ and $L(S)$
- For DoD levels, we have
TOP SECRET > SECRET > CONFIDENTIAL > UNCLASSIFIED

Multilevel Security (MLS)

- MLS needed when subjects/objects at different levels use same system
- MLS is a form of **Access Control**
- Classified government/military information
- **Business example:** info restricted to
 - Senior management only
 - All management
 - Everyone in company
 - General public
- Network firewall
 - Keep intruders at low level to limit damage
- Confidential medical info, databases, etc.

Example

| security level | subject | object |
|----------------|---------|-----------------|
| Top Secret | Alice | Personnel Files |
| Secret | Bob | E-Mail Files |
| Confidential | Chiang | Activity Logs |
| Unclassified | Fred | Telephone Lists |

Alice can read all files

Chiang cannot read Personnel or E-Mail Files

Fred can only read Telephone Lists

Bell-LaPadula

- BLP security model designed to express essential requirements for MLS
- BLP deals with **confidentiality**
 - To prevent unauthorized reading
- Recall that O is an object, S a subject
 - Object O has a classification
 - Subject S has a clearance
 - Security level denoted $L(O)$ and $L(S)$

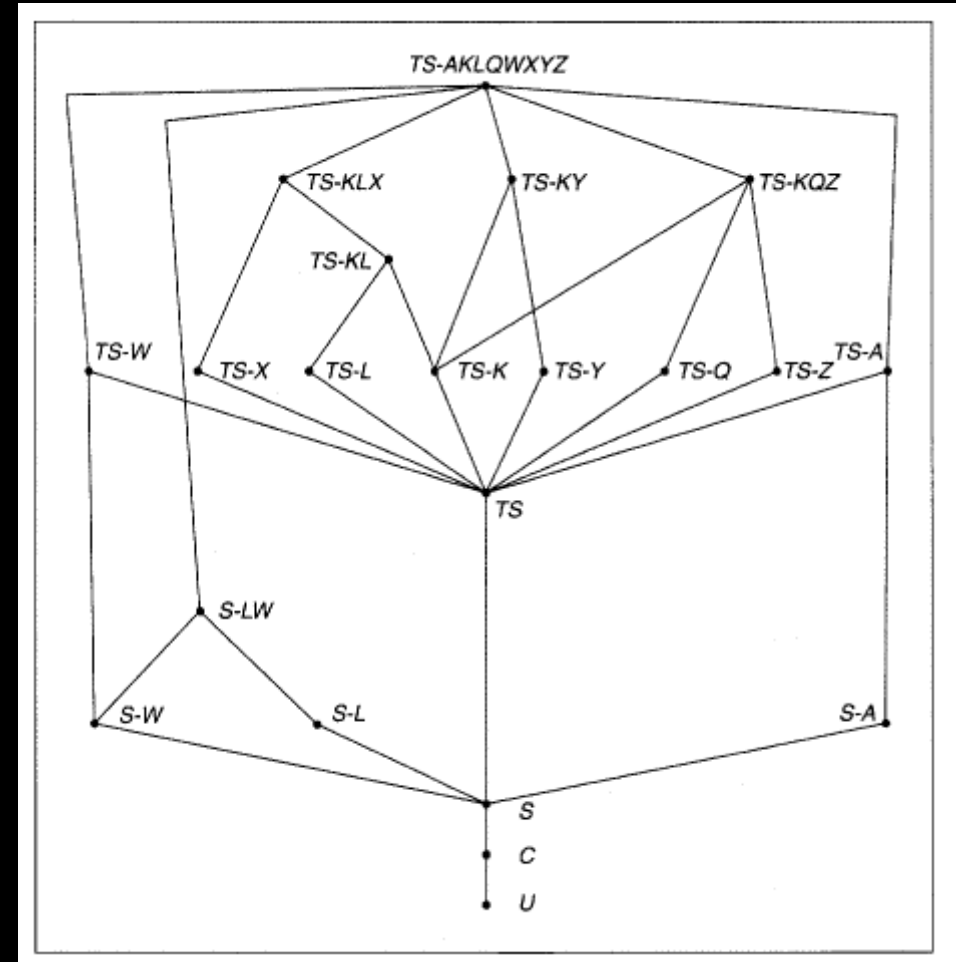
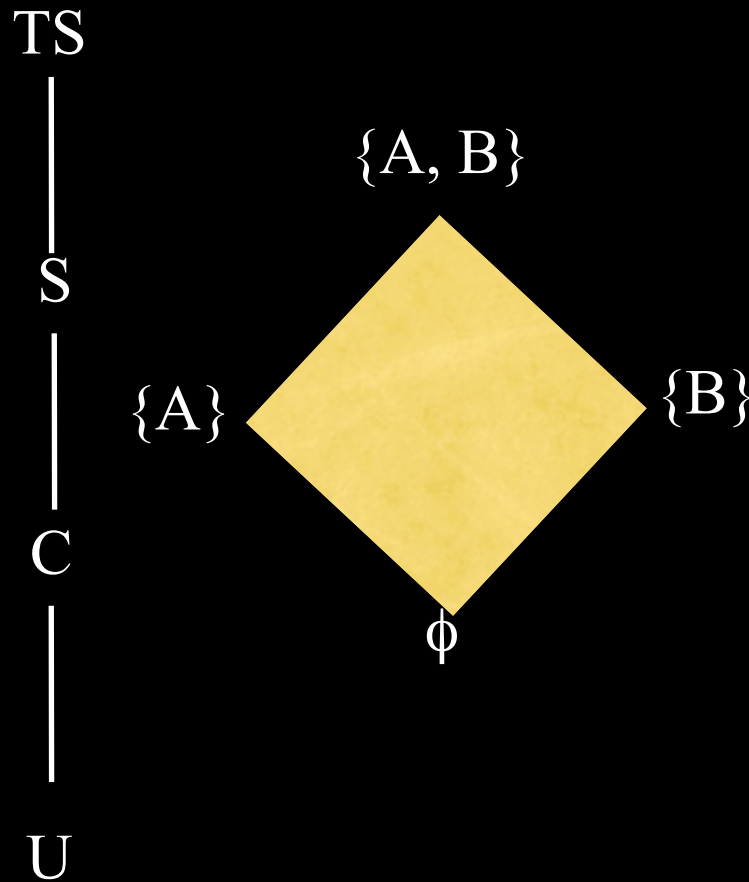
BLP rules

Simple Security Condition: S can **read** O
if and only if $L(O) \leq L(S)$

***-Property (Star Property):** S can **write**
 O if and only if $L(S) \leq L(O)$

- **No read up, no write down**

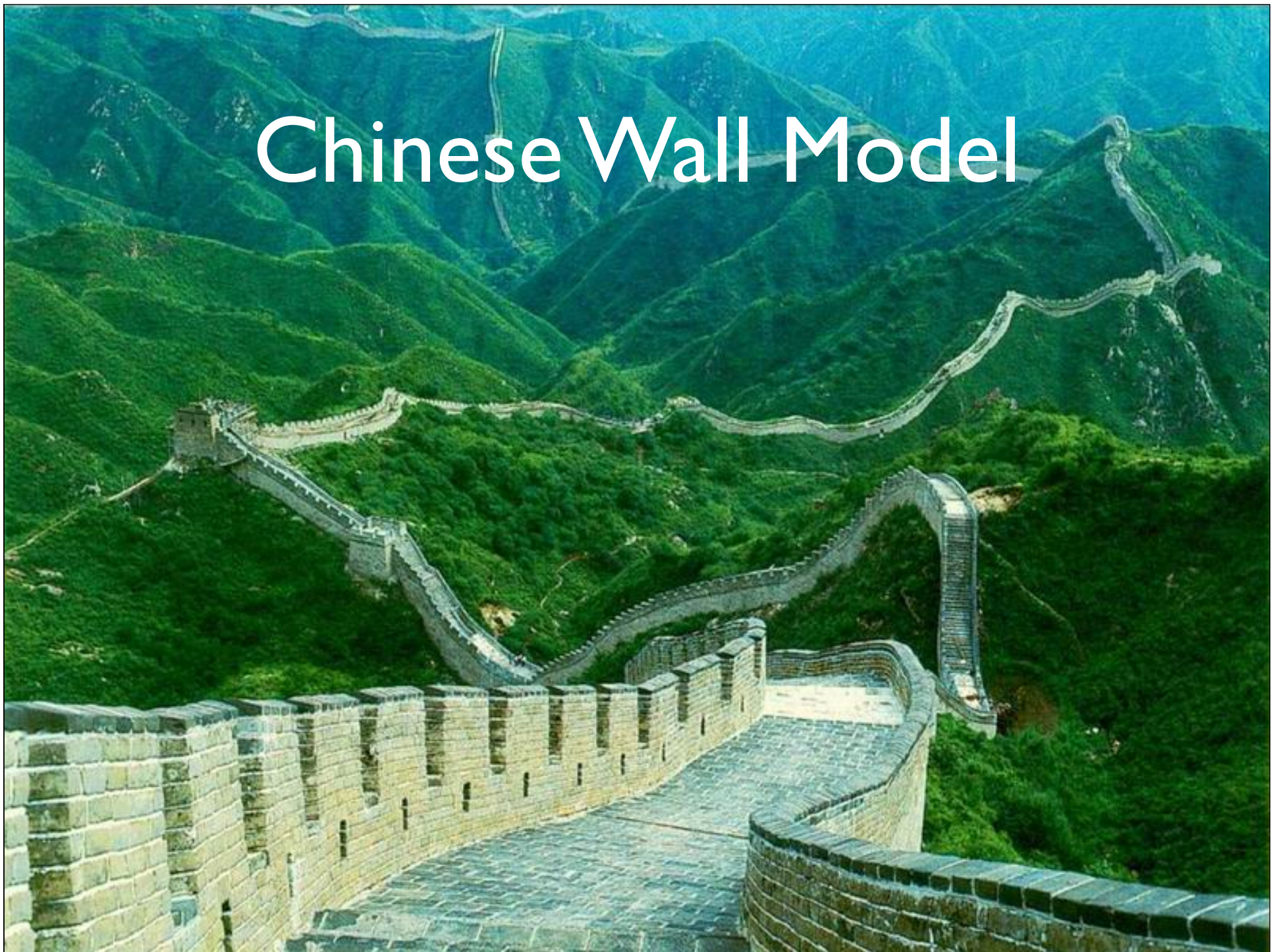
The Military Lattice



Key Points Regarding Confidentiality Policies

- Confidentiality policies restrict flow of information
- Bell-LaPadula model supports **multilevel security**
 - Cornerstone of much work in computer security

Chinese Wall Model



What's Chinese Wall Model

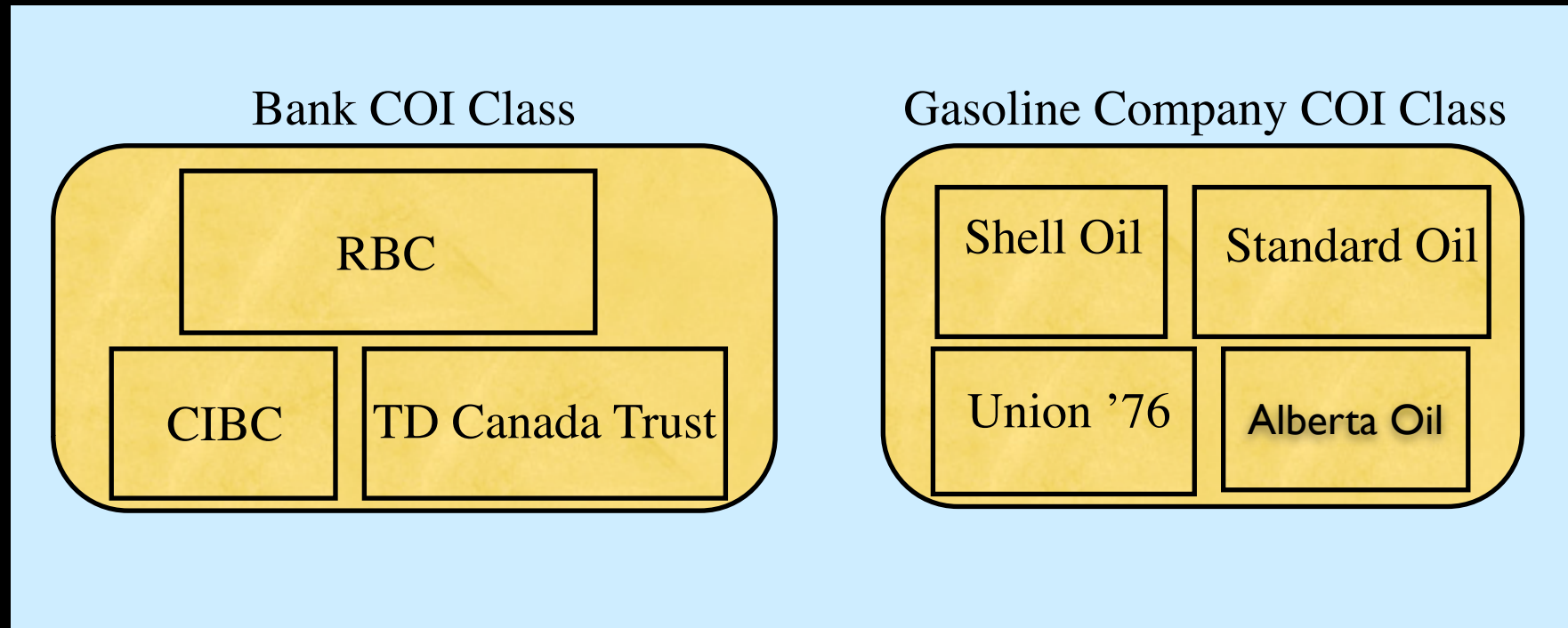
Problem:

- Tony advises American Bank about investments
- He is asked to advise Toyland Bank about investments
- **Conflict of interest** to accept, because his advice for either bank would affect his advice to the other bank

Organization

- Organize entities into “conflict of interest” classes
- Control subject accesses to each class
- Control writing to all classes to ensure information is not passed along in violation of rules
- Allow sanitized data to be viewed by everyone

Example



- If Anthony reads any **Company dataset** (CD) in a **conflict of interest** (COI), he can never read another CD in that COI
 - Possible that information learned earlier may allow him to make decisions later

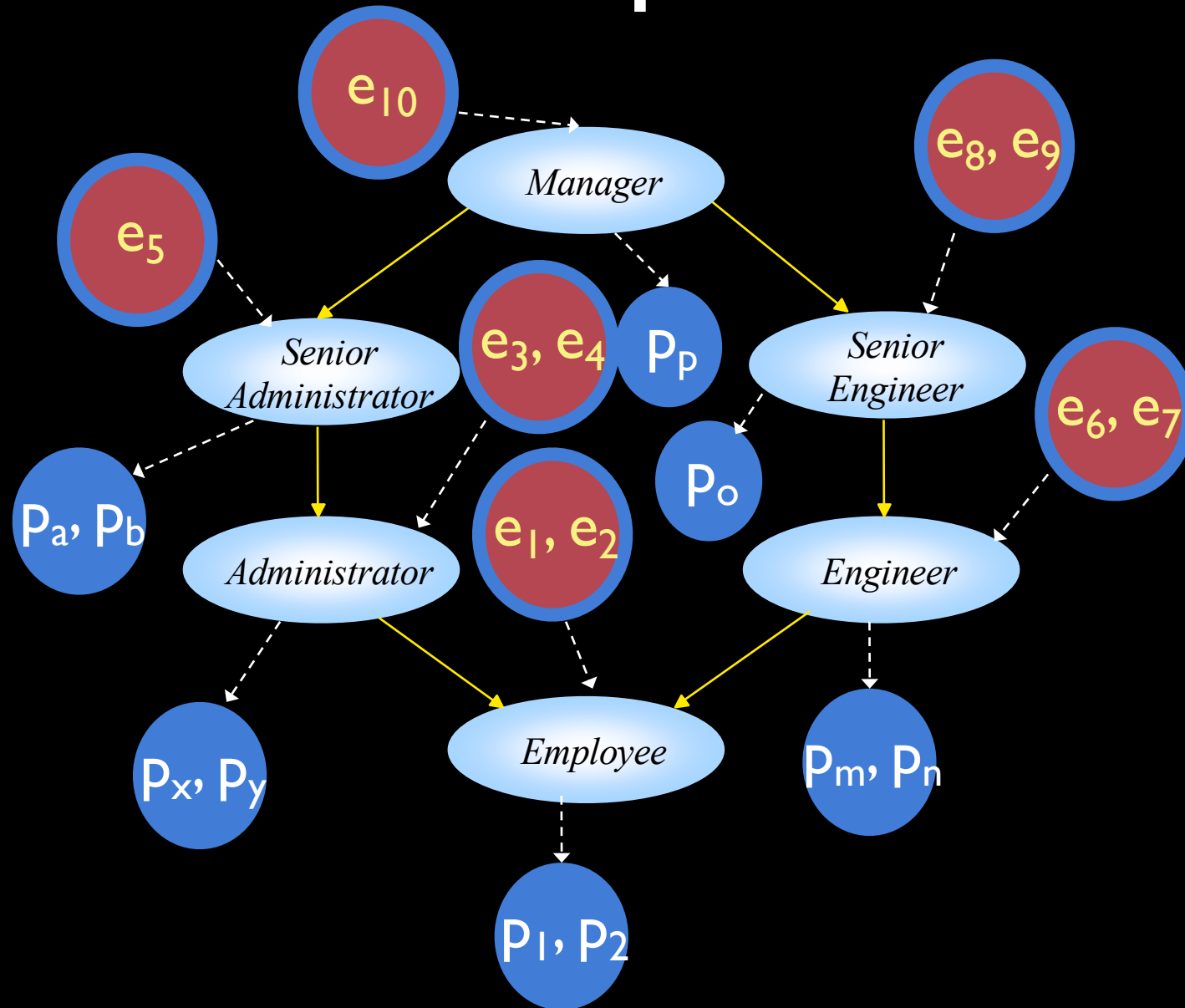


Role-based Access Control (RBAC)

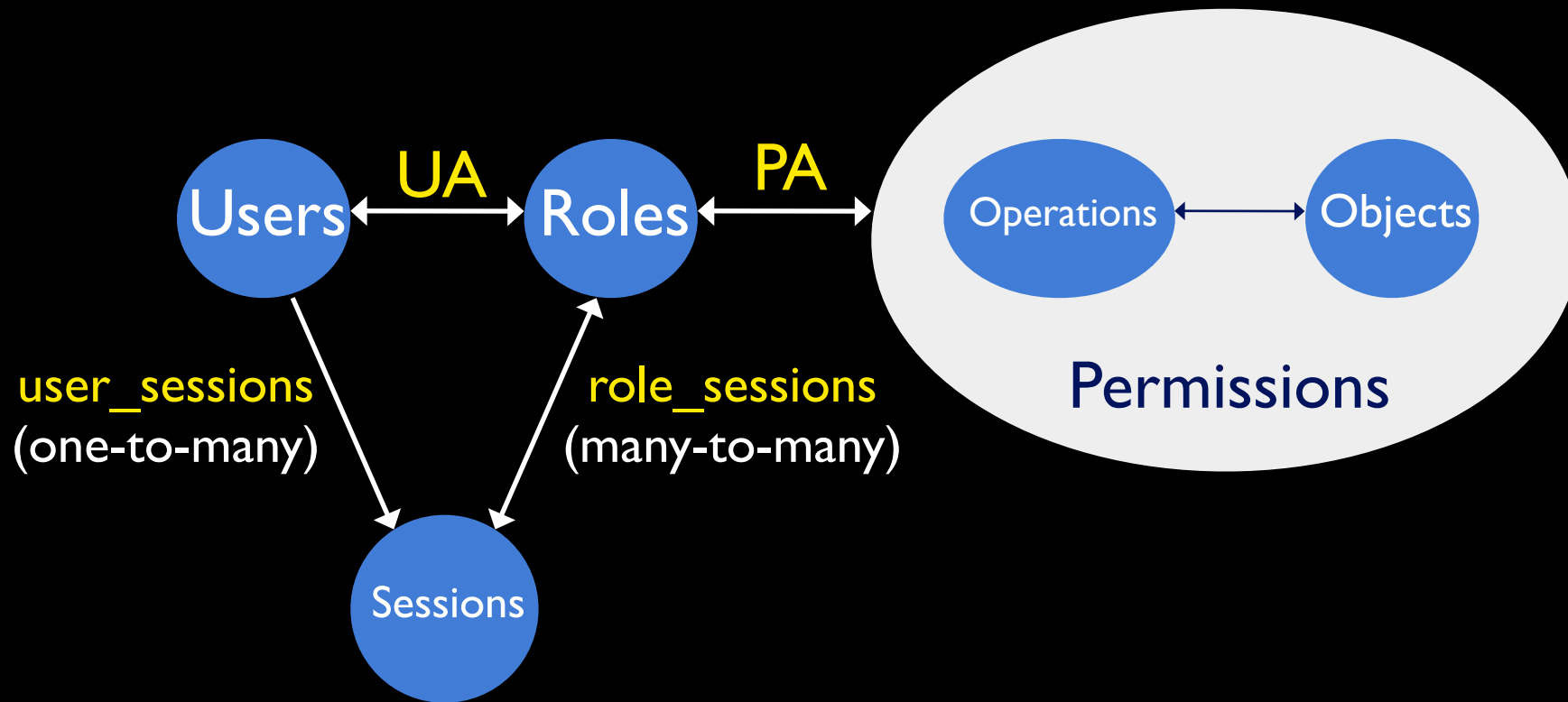
RBAC

- Access depends on **role**, not identity or label
 - Example:
 - Allison, **administrator** for a department, has access to financial records.
 - She leaves.
 - Betty hired as the new **administrator**, so she now has access to those records
 - The role of “administrator” dictates access, not the identity of the individual.

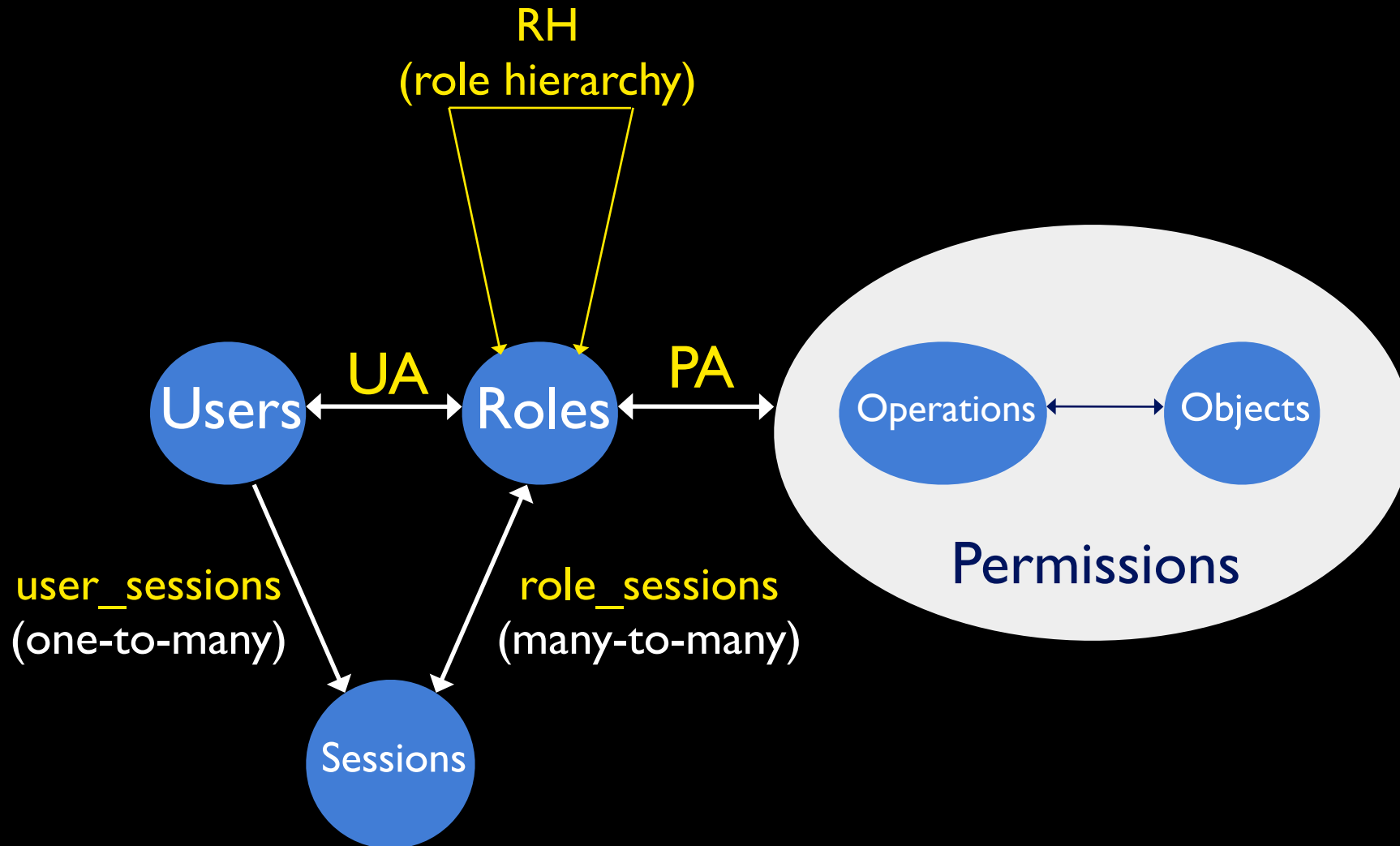
Example



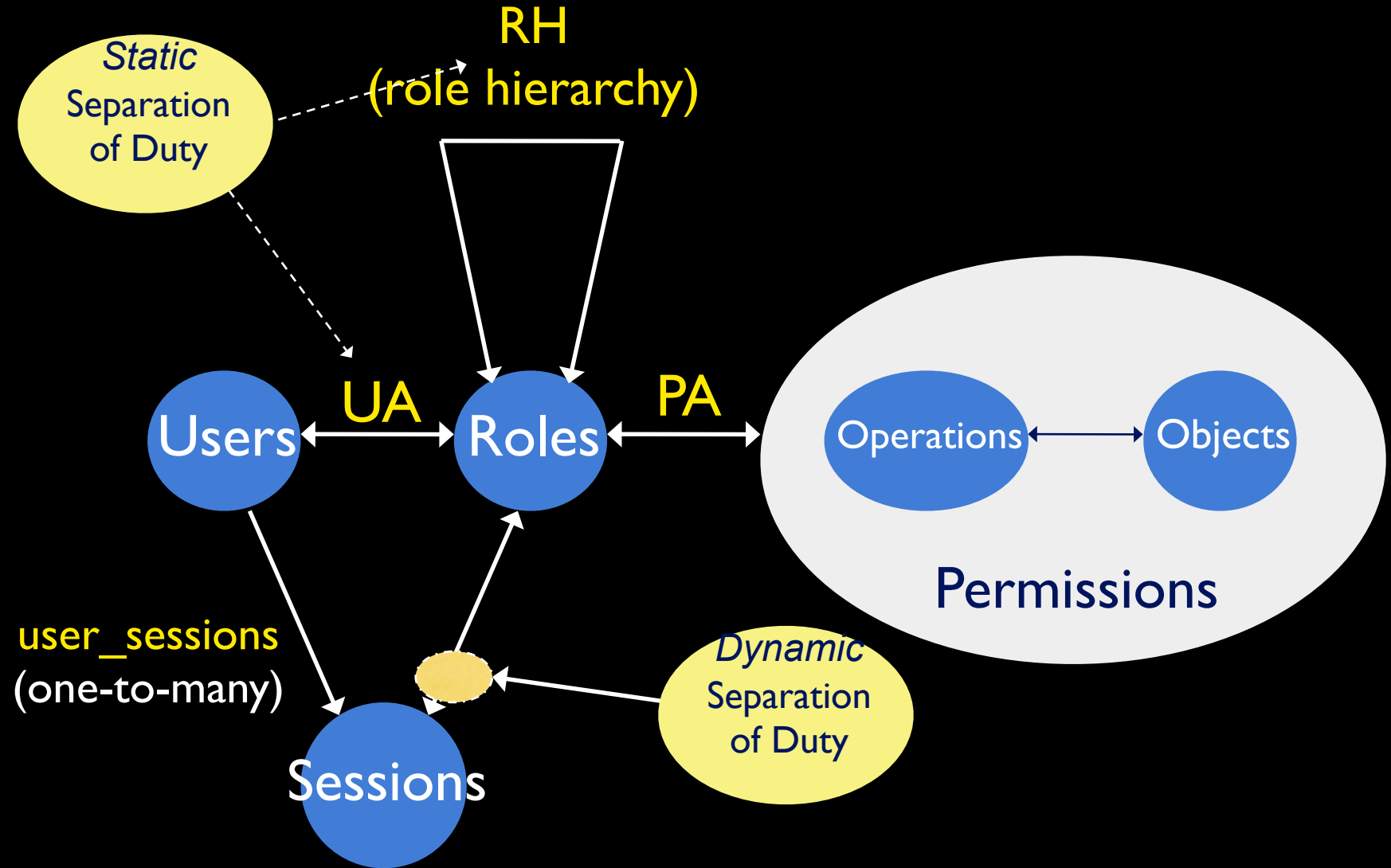
RBAC (ANSI Standard)



RBAC with General Role Hierarchy



Constrained RBAC



what we learned so far

- structure of access controls (PEP & PDP)
- access matrix
 - ACLs and capability lists
- security policies
 - confidentiality & integrity
 - types of policies (DAC, MAC, OrCon)
- BLP model
- Chinese Wall model
- RBAC model

case study: access control in Android