ECE458/ECE750T27: Computer Security
Access Control

Dr. Kami Vaniea
Electrical and Computer Engineering
kami.vaniea@uwaterloo.ca

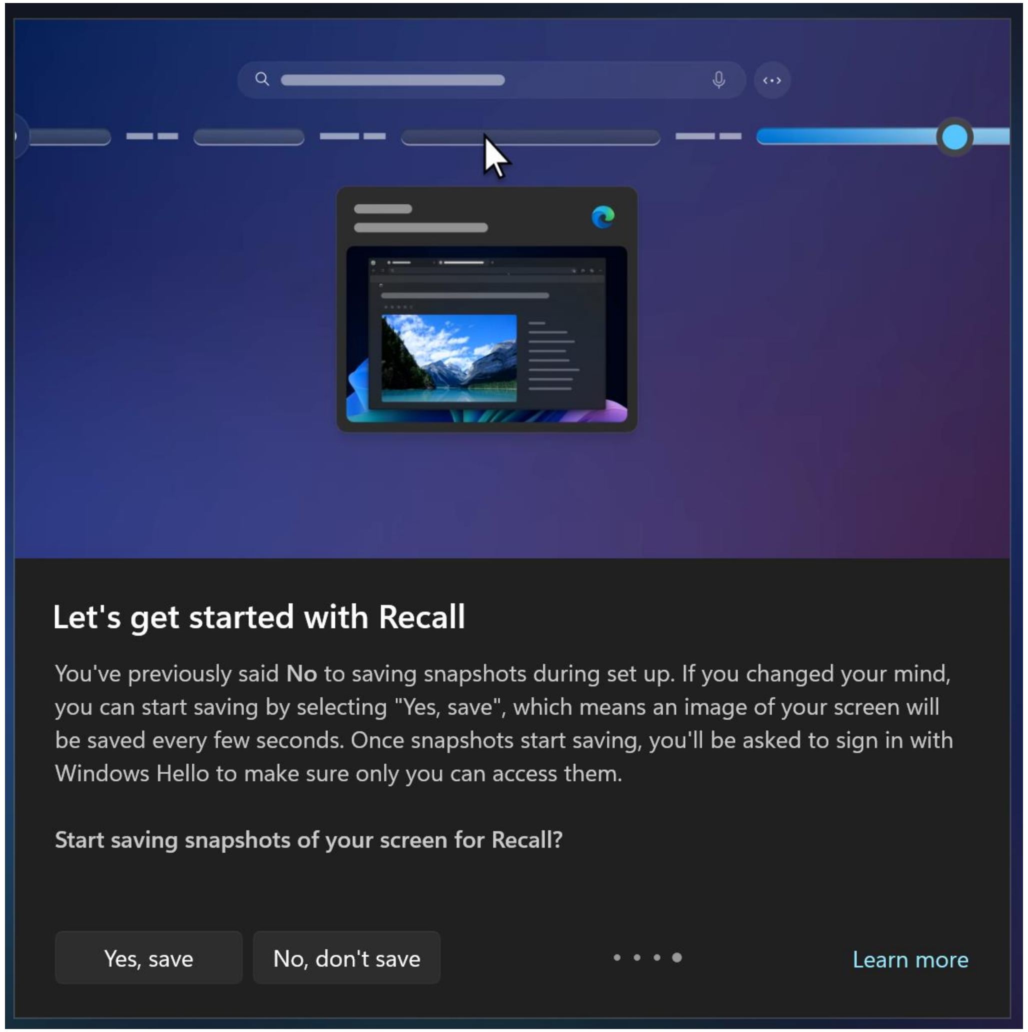
UNIVERSITY OF WATERLOO | FACULTY OF ENGINEERING

TULiPS
Technology Usability Lab in Privacy and Security

First, the news…

- First 5 minutes we talk about something interesting and recent
- You will not be tested on the news part of lecture
- You may use news as an example on tests
- Why do this?
  1. Some students show up late for various good reasons
  2. Reward students who show up on time
  3. Important to see real world examples

News...



Let's get started with Recall

You've previously said **No** to saving snapshots during set up. If you changed your mind, you can start saving by selecting "Yes, save", which means an image of your screen will be saved every few seconds. Once snapshots start saving, you'll be asked to sign in with Windows Hello to make sure only you can access them.

Start saving snapshots of your screen for Recall?

NO LECTURE MONDAY

Canada Day

TUTORIALS

Tutorials

- Scheduled
  - 8:30am Thursdays
- Options:
  - Run 8:30am tutorials as help sessions 2 weeks before assignments are due
  - Have TA office hours later in the day in weeks leading up to assignment deadlines

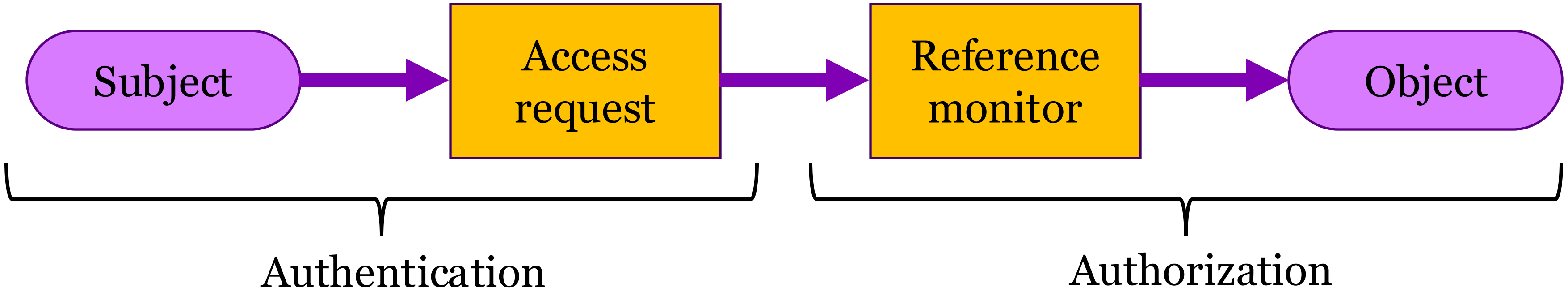
ACCESS AND INFORMATION FLOW

System security policies and models

- A security policy describes requirements for a system.
- A security model is a framework with which a policy can be described.
- There are two basic paradigms:
  - Access control
  - Information flow control

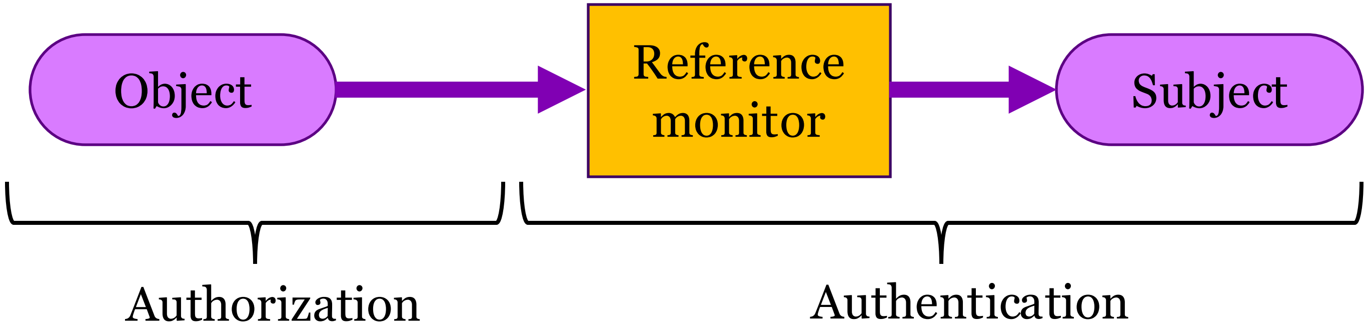
Access control

A guard controls whether a principal (the subject) is allowed access to a resource (the object).



Information flow control

A guard controls whether a principal (the subject) is allowed access to a resource (the object).



This is the dual notion, sometimes used when confidentiality is the primary concern.

The difference

- Access control
  - Starts with the subject (user) and asks if the user has access to the object.
- Information flow control
  - Starts with the object (information) and asks if that information can be known to the subject.

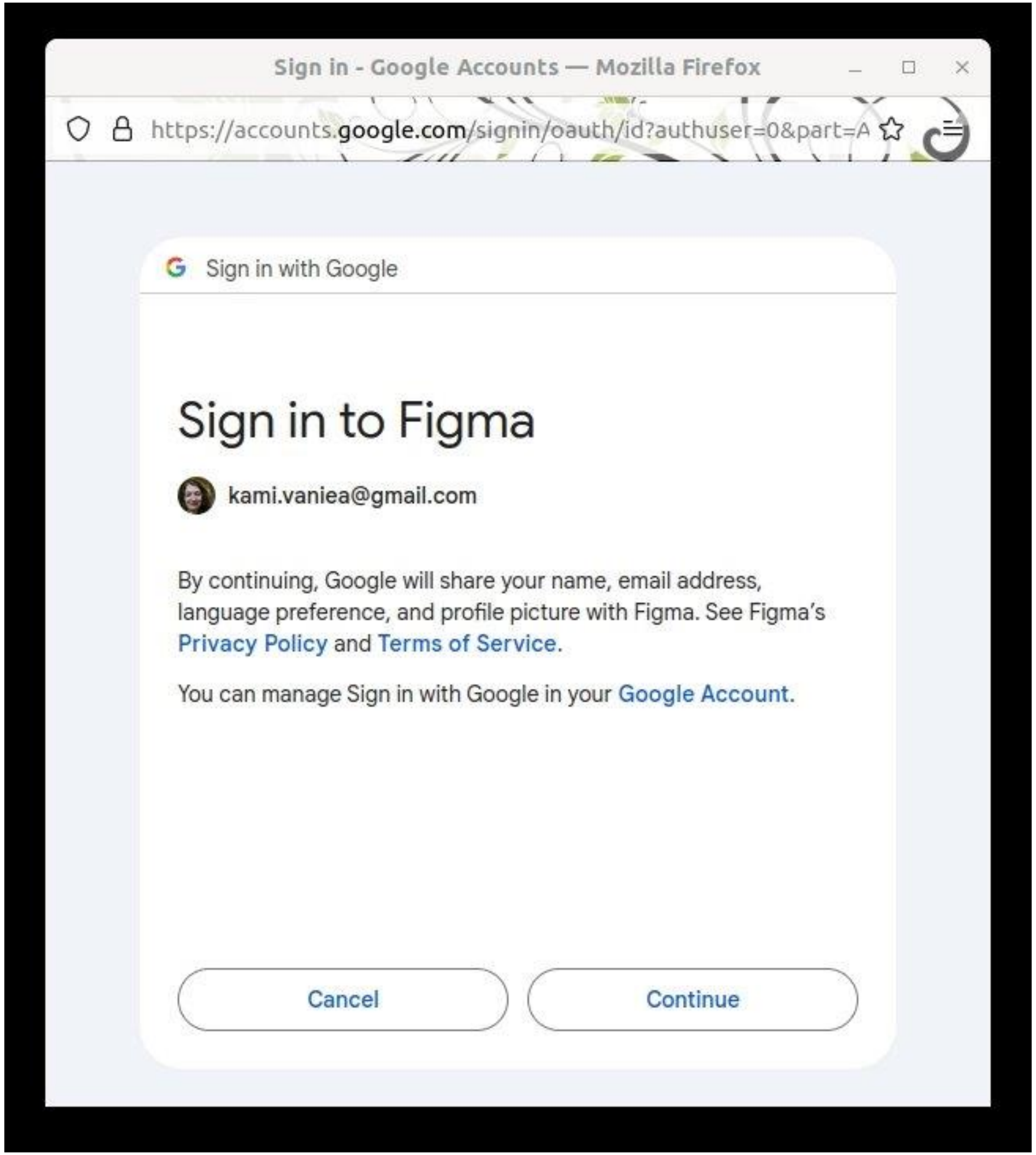
ACCESS CONTROL

Access Control

- Ensure that certain users can use a resource in one way (i.e. read-only), others in a different way (i.e. write), and still others not at all.
- **Subjects** – human users who are often represented by surrogate programs running on behalf of the users.
- **Objects** – things on which an action can be performed. Such as files, database tables, programs, memory objects, hardware, network connections, and processors. User accounts can also be objects since they can be added to the system, removed, and modified.
- **Access modes** – any controllable actions of subjects on objects, including read, write, modify, delete, execute, create, destroy, copy export, and import.

Example: Figma

- Subjects
  - Figma
- Objects
  - My name, email address, language preferences, and profile picture
  - Any future updates to the above
- Access Modes
  - Read



Sign in - Google Accounts — Mozilla Firefox

https://accounts.google.com/signin/oauth/id?authuser=0&part=A

Sign in with Google

Sign in to Figma

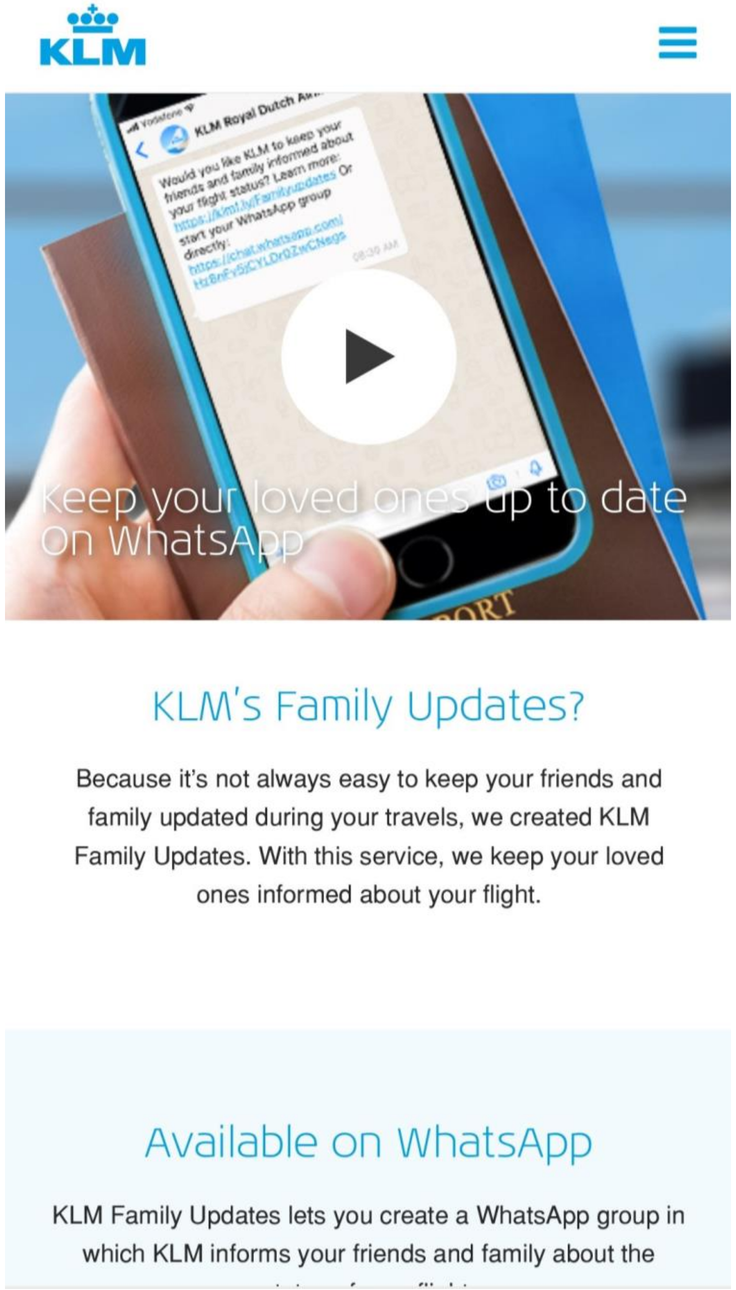
kami.vaniea@gmail.com

By continuing, Google will share your name, email address, language preference, and profile picture with Figma. See Figma's Privacy Policy and Terms of Service.

You can manage Sign in with Google in your Google Account.

Access Modes can have wide variety

- Read, write, execute
- Delete, append, create file, create folder, see folder contents
- Ability to read live update feed



Would you like KLM to keep your friends and family informed about your flight status? Learn more: https://klmf.ly/Familyupdates Or start your WhatsApp group directly: https://chat.whatsapp.com/HzBnFy5jCYLDr0ZwCNegs

Keep your loved ones up to date On WhatsApp

KLM's Family Updates?

Because it's not always easy to keep your friends and family updated during your travels, we created KLM Family Updates. With this service, we keep your loved ones informed about your flight.

Available on WhatsApp

KLM Family Updates lets you create a WhatsApp group in which KLM informs your friends and family about the

Principle of least privilege

- A subject should have access to the smallest number of objects necessary to perform their task.
    - Example: A program does not need access to the absolute memory addresses to which a page number reference translates
    - Permissions should match what is possible. Impossible actions should not be granted
- Permissions should be reviewed and adjusted over time
    - New job, means new permissions and removal of old ones
    - Analyzing logs can show what permissions are not being used and can possibly be removed

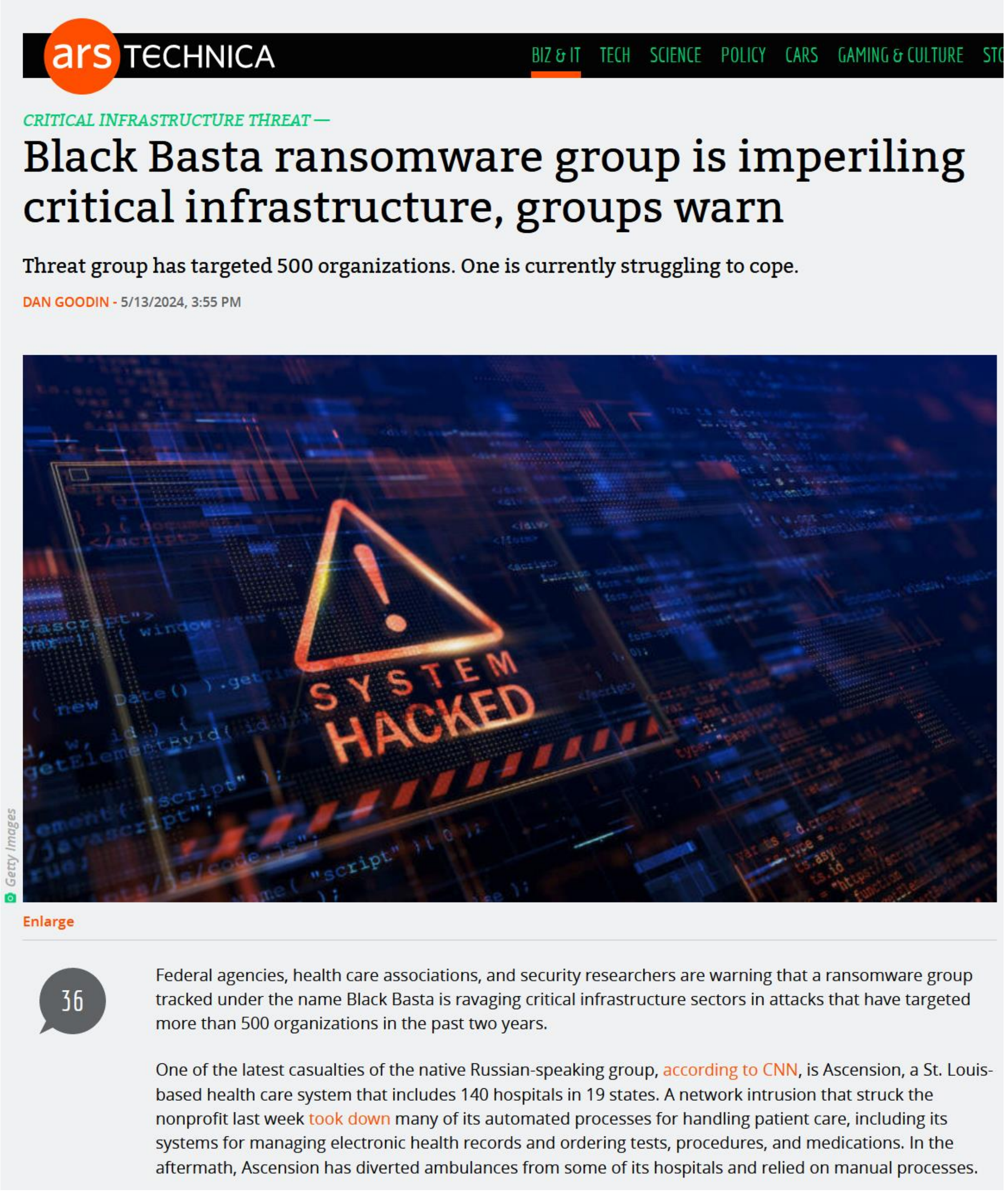
Example: Elevator

- Key card controls what floors the elevator will go to based on guest room number.
- Weak version of principle of least privilege.

6
19
17
15
13
11
9
7
5
3
1
-1
18
16
14
12
10
8
6
4
2
0
-2
In case of emergency, please push ☎ to get in contact with the front desk

Example: Ransomware

- Attacker sends lots of spam to victim employee email
- Attacker calls employee claiming to be IT Services and offering to help fix spam problem
- Suggests employee download a tool that allows attacker to interact with employee's computer (RMM)
- Employee downloads
- Attacker installs ransomware which then tries to install itself everywhere
- After a set time, ransomware encrypts every file it can access



CRITICAL INFRASTRUCTURE THREAT —

Black Basta ransomware group is imperiling critical infrastructure, groups warn

Threat group has targeted 500 organizations. One is currently struggling to cope.

DAN GOODIN - 5/13/2024, 3:55 PM

Federal agencies, health care associations, and security researchers are warning that a ransomware group tracked under the name Black Basta is ravaging critical infrastructure sectors in attacks that have targeted more than 500 organizations in the past two years.

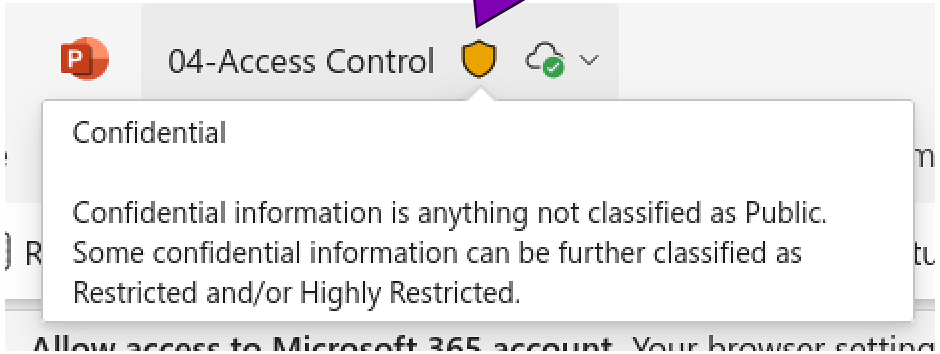
One of the latest casualties of the native Russian-speaking group, according to CNN, is Ascension, a St. Louis-based health care system that includes 140 hospitals in 19 states. A network intrusion that struck the nonprofit last week took down many of its automated processes for handling patient care, including its systems for managing electronic health records and ordering tests, procedures, and medications. In the aftermath, Ascension has diverted ambulances from some of its hospitals and relied on manual processes.

Who sets the policy?

- Discretionary Access Control (DAC)
  - The **owner** of a resource decides who may access that resource. Policy set on a case-by-case basis.
- Mandatory Access Control (MAC)
  - The decision of accessing resources is controlled **system-wide** by a uniform policy.
- Role Based Access Control (RBAC)
  - Similar to MAC, but users are assigned roles and permissions are granted to roles, not people.

In practice a mixture of DAC and MAC may apply.

University added a MAC setting on OneDrive on top of my DAC sharing ability.



04-Access Control

Confidential

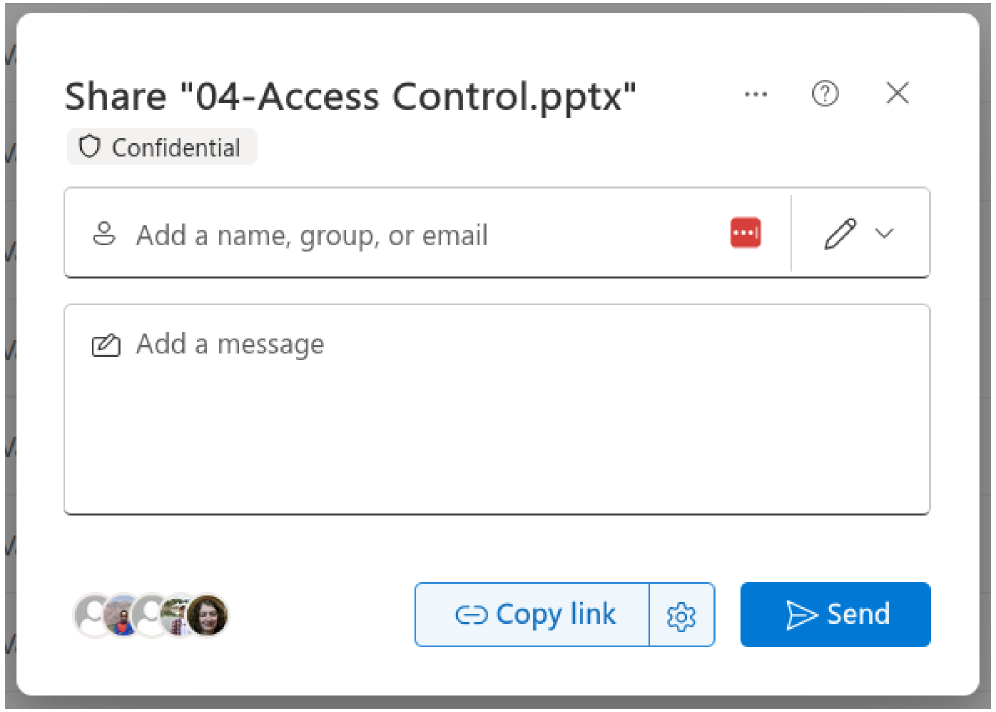
Confidential information is anything not classified as Public. Some confidential information can be further classified as Restricted and/or Highly Restricted.

Allow access to Microsoft 365 account. Your browser setting

Ownership and identity

- Owners of resources may be principles in the system: subjects themselves under access control.
- The identity of subjects is also flexible: e.g., identity changes during operations
  - SUID programs in Unix
  - Sudo

I can share files on OneDrive with other University accounts, but restrictions exist for external accounts.



Share "04-Access Control.pptx"

Confidential

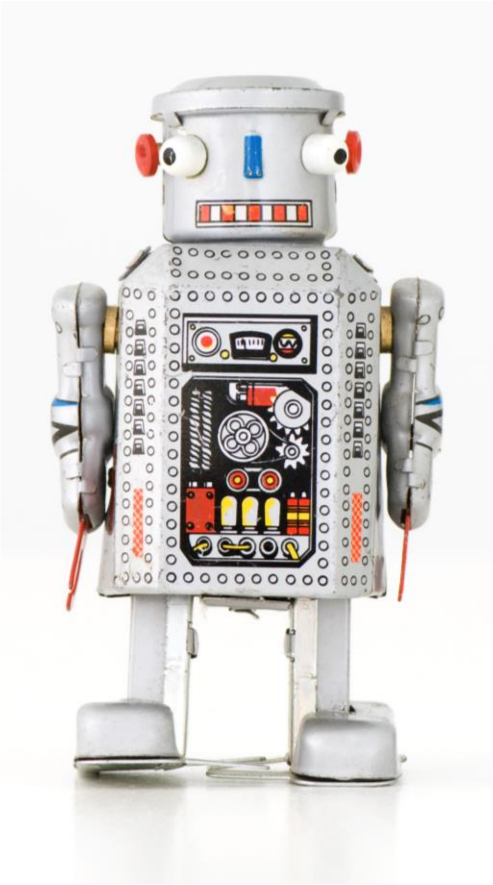
Add a name, group, or email

Add a message

Reference Monitor

Theoretical construct that manages what objects subjects can access and what actions they can perform on those objects.

1. Always invoked; so that it validates every access attempt
2. Immune from tampering
3. Assuredly correct



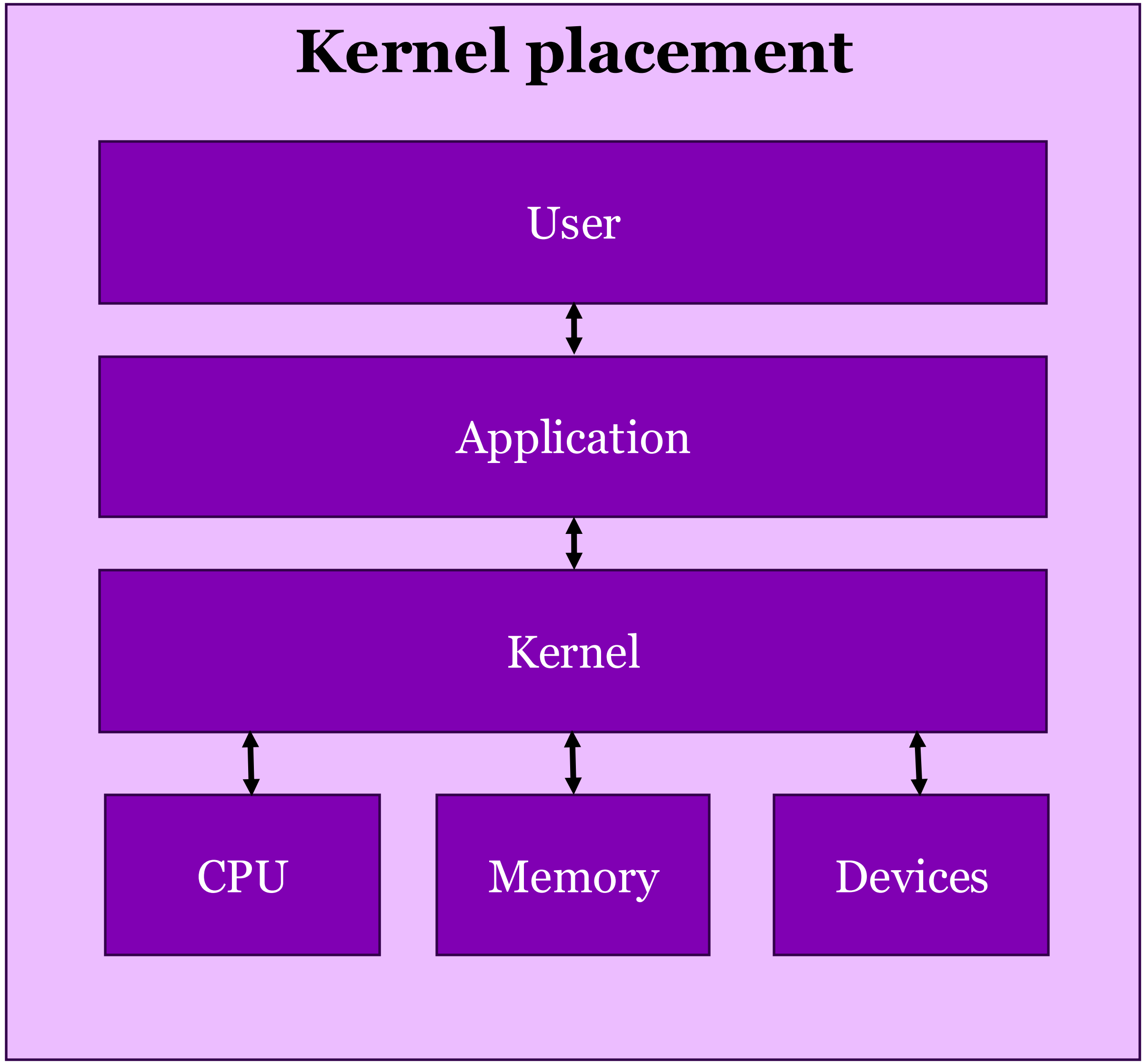
There are several ways to implement an access monitor

Here forward, I will mostly be talking about access control managed by an Operating System

- Operating systems mostly control access to files and to memory.
- Within-file access control is normally done by programs, for example, a database maintains and enforces access control separately from the operating system
- Network file system access control is also a bit different and depends on network-wide consistent user and file identifiers

How does an OS manage permission information efficiently?

- Decides what user/application is allowed to perform actions on resources like memory and devices
- Issues
  - Lookup speed
  - Clear answer (do they or don't they have access)
  - Cost to delete file/permission
  - DAC? MAC?
  - Storage space



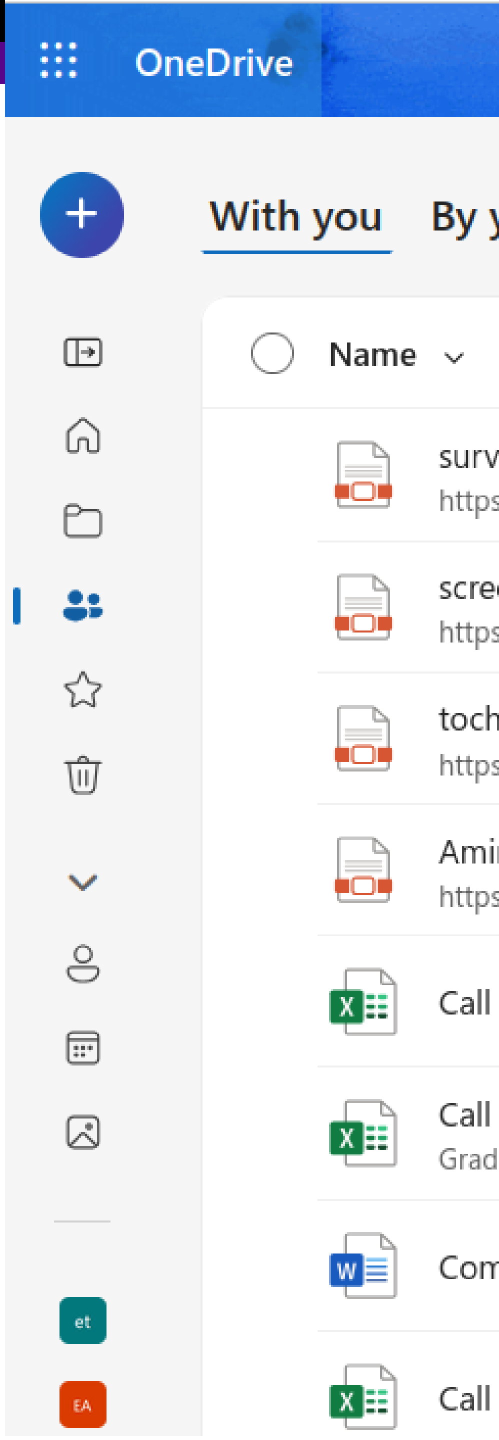
ACCESS CONTROL IMPLEMENTATIONS

Think-pair-share: Serendipity photo sharing

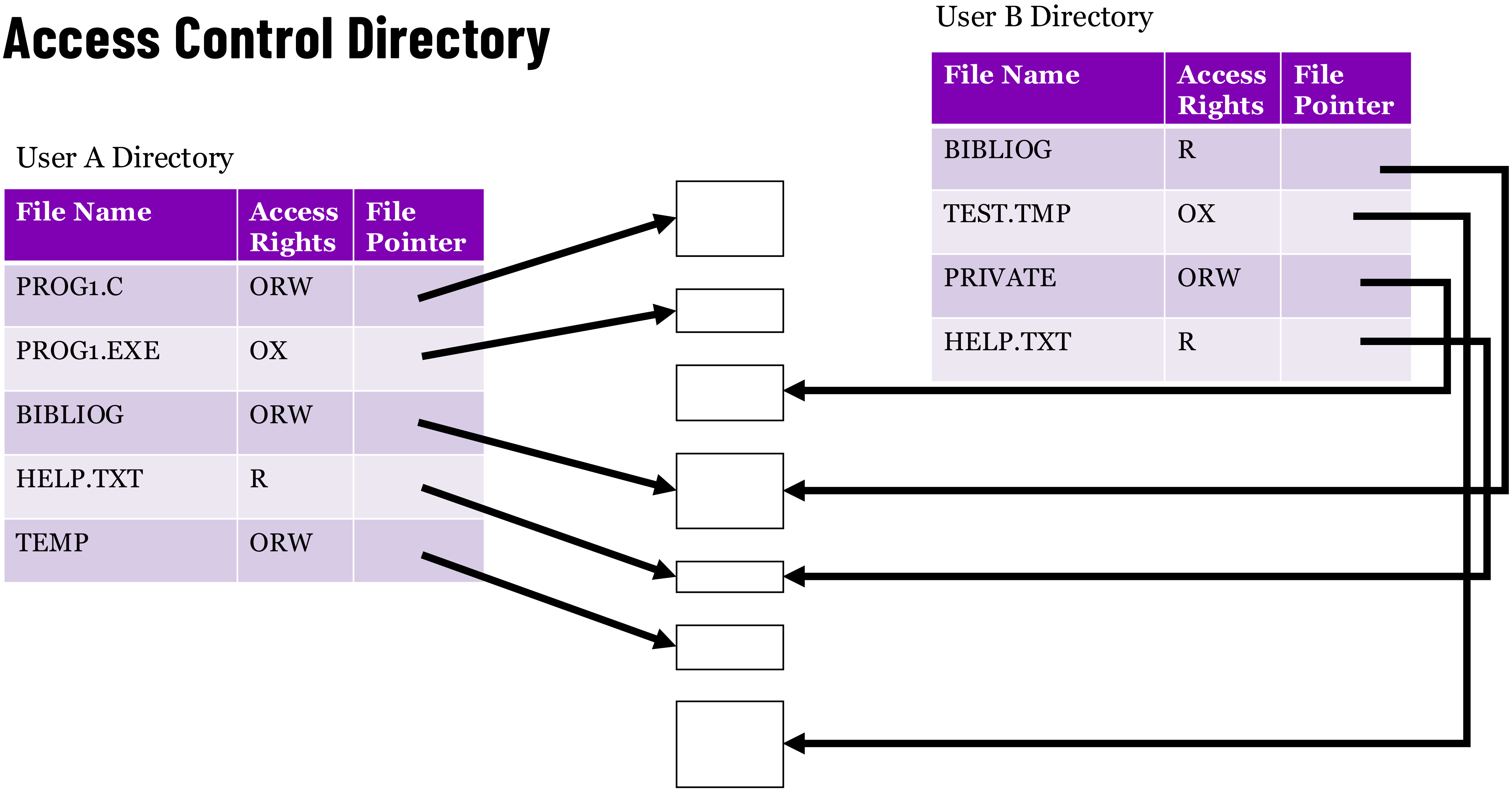
- You are building a new FooBar app for serendipitous photo sharing.
- Users can take photos and upload them. The system then grants view access of the photo to a randomly selected 5% of other users.
- Users can request edit access to photos which allows them to re-touch them.
- What access control implementation is best for this situation?

Access Control Directory

- Create a directory for each Subject (user) of all the files they can access
- Users can only access files in their directory



Access Control Directory



User A Directory

File Name	Access Rights	File Pointer
PROG1.C	ORW	
PROG1.EXE	OX	
BIBLIOG	ORW	
HELP.TXT	R	
TEMP	ORW	

User B Directory

File Name	Access Rights	File Pointer
BIBLIOG	R	
TEST.TMP	OX	
PRIVATE	ORW	
HELP.TXT	R	

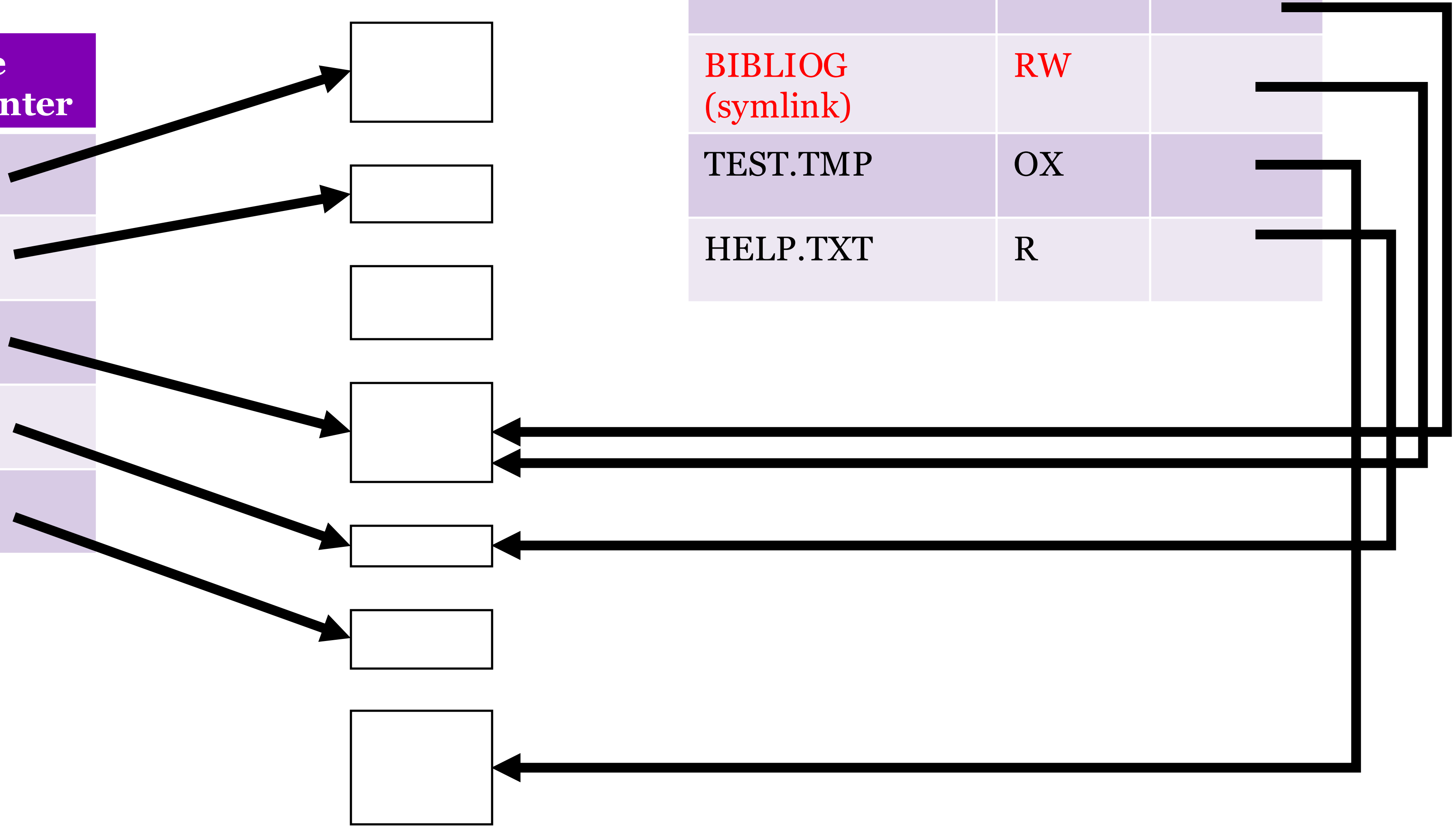
Access Control Directory
Permission granted twice

User A Directory

File Name	Access Rights	File Pointer
PROG1.C	ORW	
PROG1.EXE	OX	
BIBLIOG	ORW	
HELP.TXT	R	
TEMP	ORW	

User B Directory

File Name	Access Rights	File Pointer
BIBLIOG	R	
BIBLIOG (symlink)	RW	
TEST.TMP	OX	
HELP.TXT	R	



Access Control Matrix

- Matrix of all the Subjects (rows) and all the Objects (columns) with the access modes listed in the cells
- Clear and lookup is efficient
- Most of the matrix is empty since most Subjects do not have access to most Objects

	Bibliog	Temp	F	Help.txt	C_Comp	Linker	Clock	Printer
User A	ORW	ORW	ORW	R	X	X	R	W
User B	R			R	X	X	R	W
User S	RW		R	R	X	X	R	W
Sys Mgr				RW	OX	OX	ORW	O
User Svcs				O	X	X	R	W

Access Control Matrix (more formally)

How are access control rights defined? Many schemes but ultimately modelled by:

- A set S of subjects, a set O of objects
- A set A of operations (modeled by access rights), we consider $A=\{own, read, write\}$
- An **access control matrix**

$$M = (M_{so})_{s \epsilon S, o \epsilon O}$$

Where each entry $M_{so} \subseteq A$ defines rights for s to access o

	Bibliog	Temp	F	Help.txt
User A	ORW	ORW	ORW	R
User B	R			R
User S	RW		R	R
Sys Mgr				RW
User Svcs				O

Access Control Triples

- One row for each triple of <subject, object, access right> where one exists
- Solves the space problem
- Lookups are now expensive

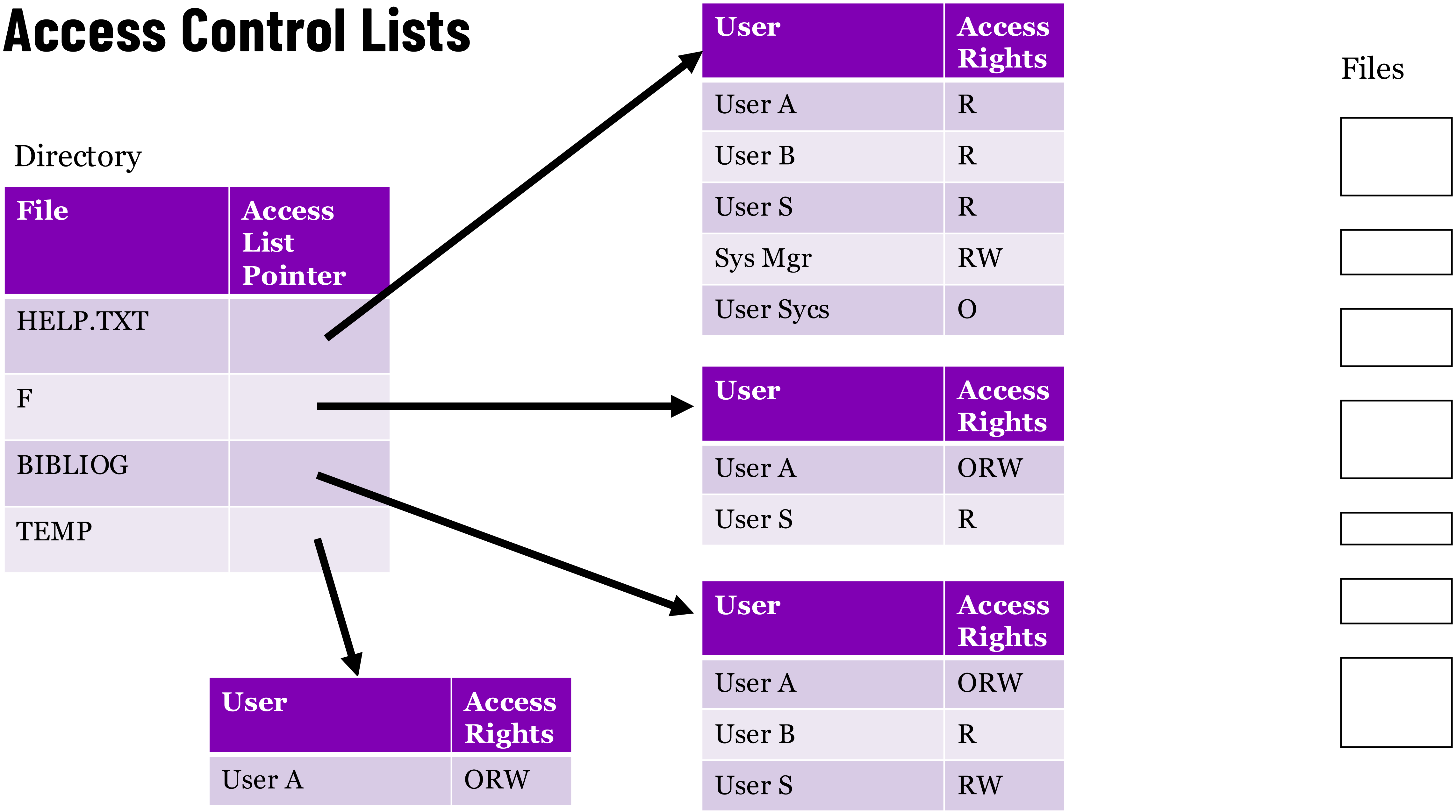
Subject	Object	Access Right
User A	Bibliog	ORW
User B	Bibliog	R
User S	Bibliog	RW
User A	Temp	ORW
User A	F	ORW
…	….	…

Access Control Lists

- Idea: store the column of the Access Control Matrix with each file
- Less space needed, though still quite a bit
- Faster lookup
- But still slow to make changes. Denying all access for a specific user requires searching and editing many lists

	Bibliog	Temp	F	Help.txt	C_Comp	Linker	Clock	Printer
User A	ORW	ORW	ORW	R	X	X	R	W
User B	R			R	X	X	R	W
User S	RW		R	R	X	X	R	W
Sys Mgr				RW	OX	OX	ORW	O
User Svcs				O	X	X	R	W

Access Control Lists



Directory

File	Access List Pointer
HELP.TXT	
F	
BIBLIOG	
TEMP	

User	Access Rights
User A	R
User B	R
User S	R
Sys Mgr	RW
User Sycs	O

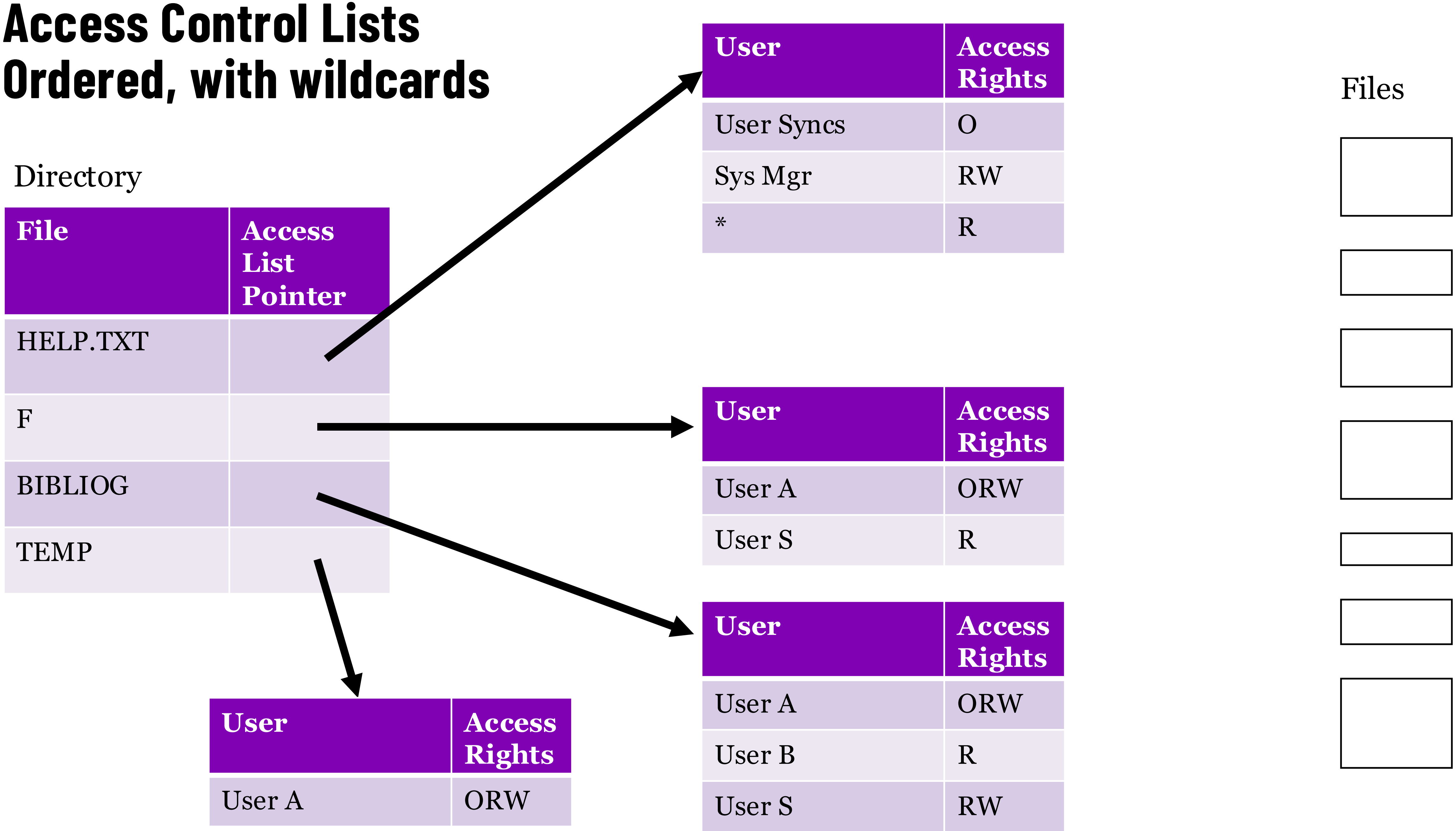
User	Access Rights
User A	ORW
User S	R

User	Access Rights
User A	ORW
User B	R
User S	RW

User	Access Rights
User A	ORW

Files

Access Control Lists
Ordered, with wildcards



Directory

File	Access List Pointer
HELP.TXT	
F	
BIBLIOG	
TEMP	

User	Access Rights
User Syncs	O
Sys Mgr	RW
*	R

User	Access Rights
User A	ORW
User S	R

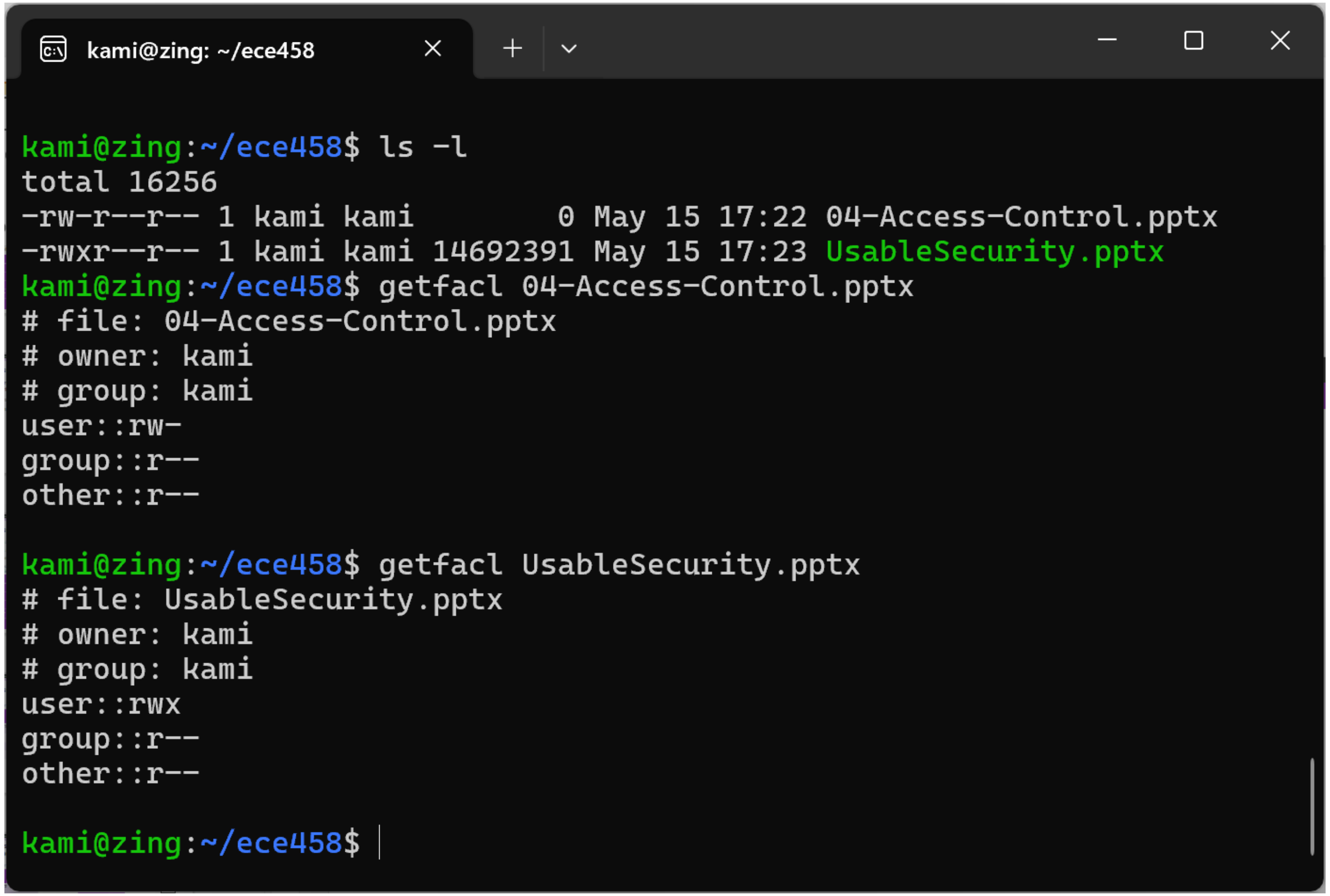
User	Access Rights
User A	ORW
User B	R
User S	RW

User	Access Rights
User A	ORW

Files

Linux uses the Access Control Lists model

- Every file has a **U**ser, **G**roup, and **O**ther
- The **U**ser is the owner
- The **G**roup is a list of users for whom these permissions will apply
- **O**ther refers to all users logged into this computer

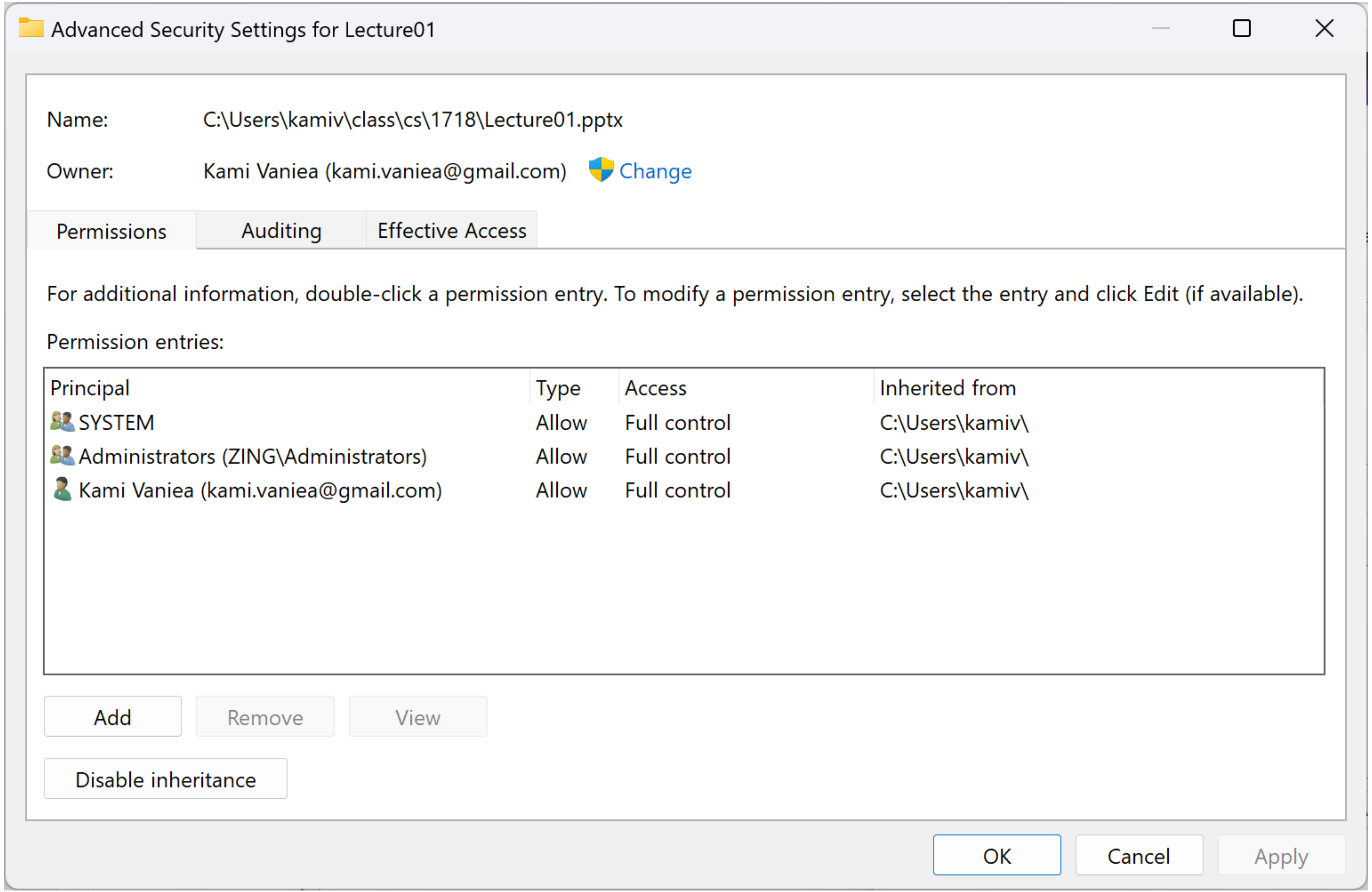


```
kami@zing:~/ece458$ ls -l
total 16256
-rw-r--r-- 1 kami kami        0 May 15 17:22 04-Access-Control.pptx
-rwxr--r-- 1 kami kami 14692391 May 15 17:23 UsableSecurity.pptx
kami@zing:~/ece458$ getfacl 04-Access-Control.pptx
# file: 04-Access-Control.pptx
# owner: kami
# group: kami
user::rw-
group::r--
other::r--

kami@zing:~/ece458$ getfacl UsableSecurity.pptx
# file: UsableSecurity.pptx
# owner: kami
# group: kami
user::rwx
group::r--
other::r--

kami@zing:~/ece458$
```

Windows also uses the Access Control List model



Advanced Security Settings for Lecture01

Name: C:\Users\kamiv\class\cs\1718\Lecture01.pptx

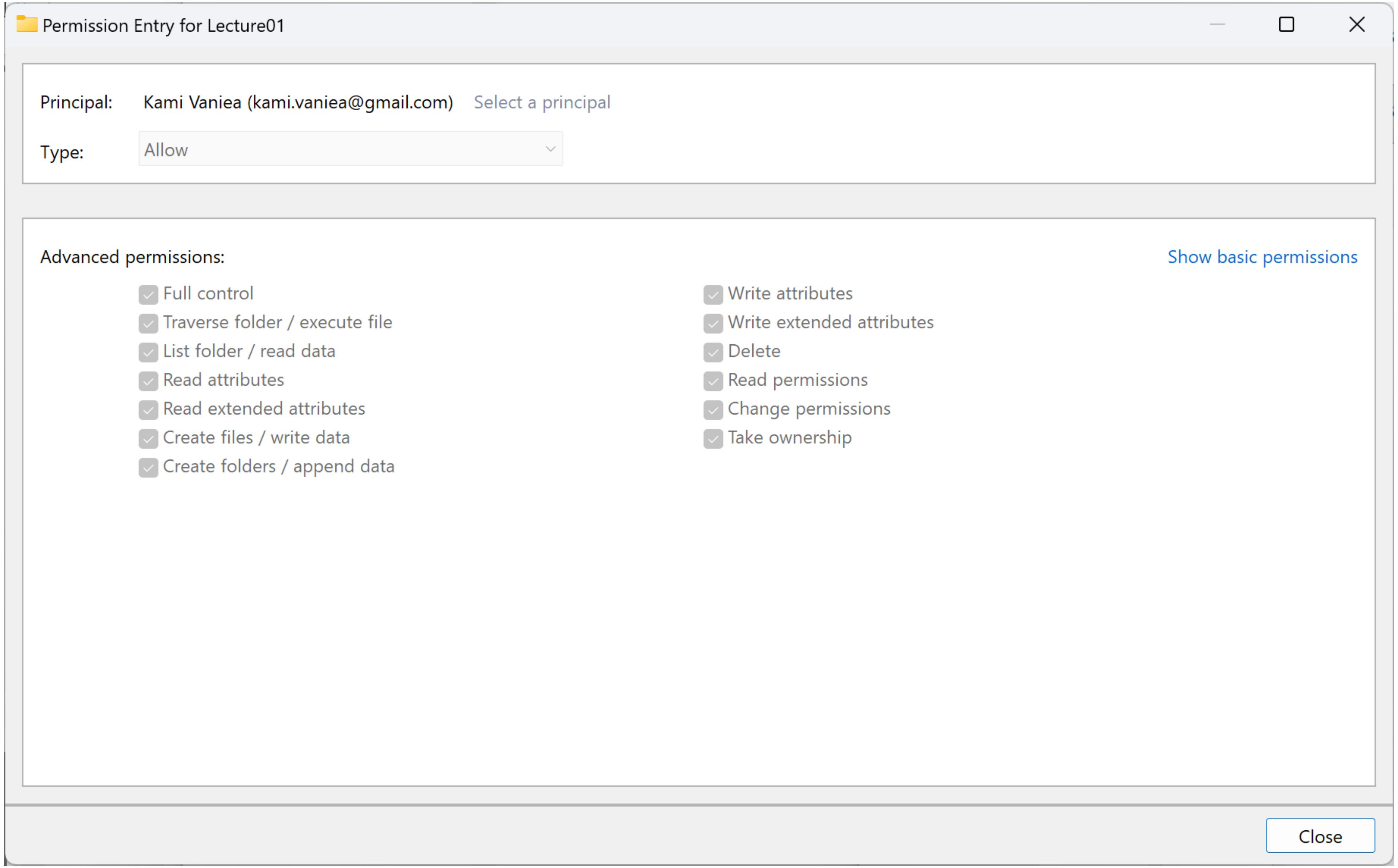
Owner: Kami Vaniea (kami.vaniea@gmail.com) Change

For additional information, double-click a permission entry. To modify a permission entry, select the entry and click Edit (if available).

Permission entries:

Principal	Type	Access	Inherited from
SYSTEM	Allow	Full control	C:\Users\kamiv\
Administrators (ZING\Administrators)	Allow	Full control	C:\Users\kamiv\
Kami Vaniea (kami.vaniea@gmail.com)	Allow	Full control	C:\Users\kamiv\

Windows also uses the Access Control List model



Permission Entry for Lecture01

Principal: Kami Vaniea (kami.vaniea@gmail.com) Select a principal

Type: Allow

Advanced permissions: Show basic permissions

☑ Full control
☑ Traverse folder / execute file
☑ List folder / read data
☑ Read attributes
☑ Read extended attributes
☑ Create files / write data
☑ Create folders / append data
☑ Write attributes
☑ Write extended attributes
☑ Delete
☑ Read permissions
☑ Change permissions
☑ Take ownership

Think-pair-share: Serendipity photo sharing

- You are building a new FooBar app for serendipitous photo sharing.
- Users can take photos and upload them. The system then grants view access of the photo to a randomly selected 5% of other users.
- Users can request edit access to photos which allows them to re-touch them.
- Original uploaders can delete photos.
- What access control implementation is best for this situation?

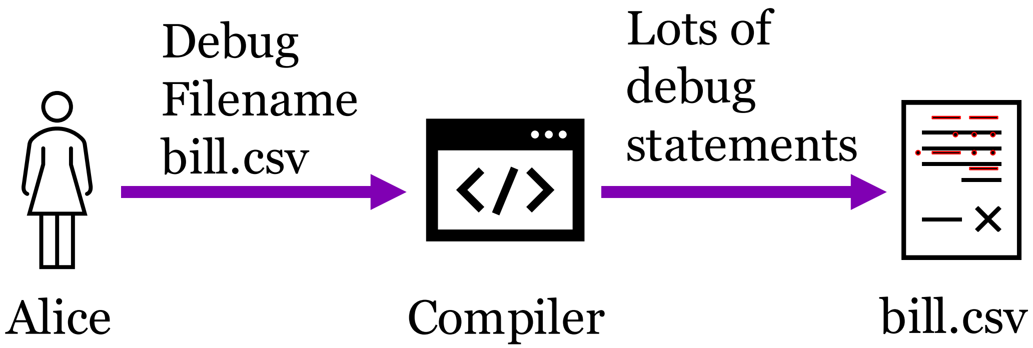
- Also consider:
  - User count
  - Resource count
  - What is the reference monitor?
  - Frequency of permission changes

Confused deputy problem

- Alice and the Compiler have different access rights to the file bill.csv
- Alice can ask the Compiler to write debug output to bill.csv which it has the right to do
- When designing a system access needs to be tracked for users AND processes they are running
- In a well-designed system the confused deputy problem should not happen

Access Control Matrix

	Compiler	bill.csv
Alice	x	--
Compiler	rx	rw



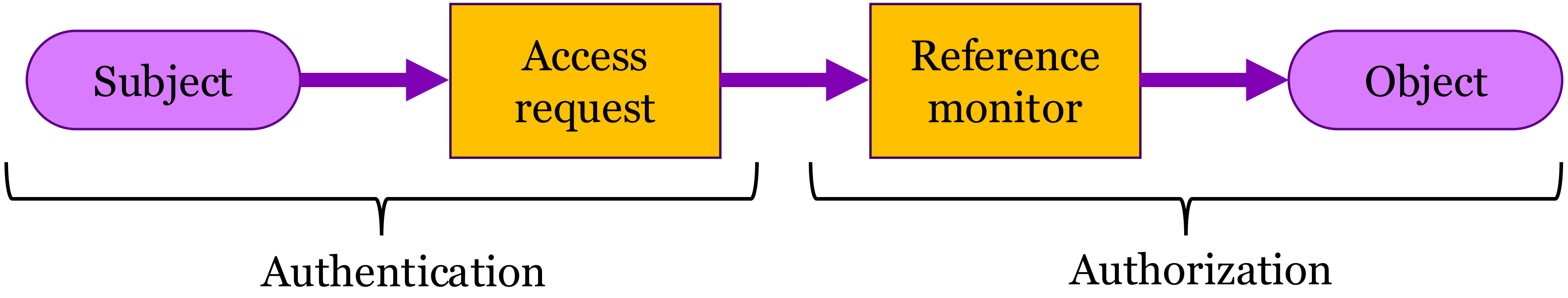
Capabilities

- Single- or multi-use ticket for a <subject, object, access mode> triple
- To access a resource the User or a process acting on the user's behalf presents a capability to the operating system
- One option is that the OS looks up the user's permissions and issues a process with a capability, at next access the process provides the capability making access rights lookup faster
- Capabilities are also easier to delegate, so easier to keep track of what each process is allowed to do on behalf of the user

INFORMATION FLOW CONTROL

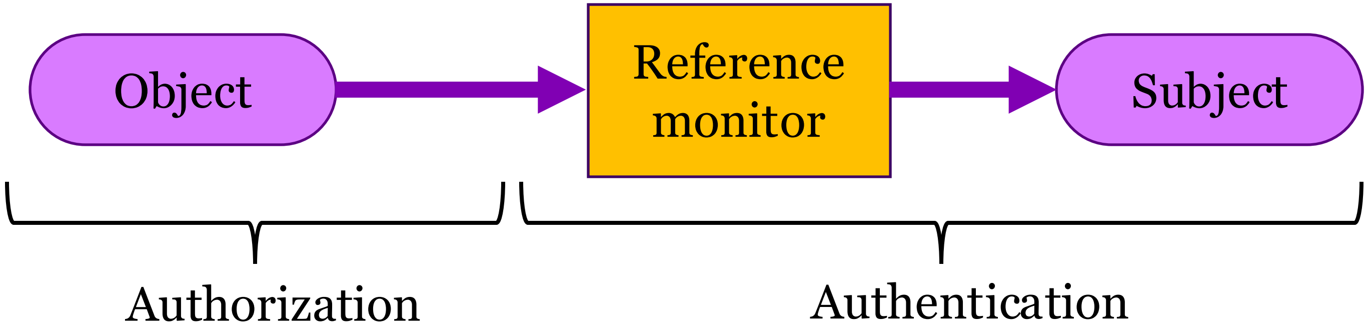
Access control

A guard controls whether a principal (the subject) is allowed access to a resource (the object).



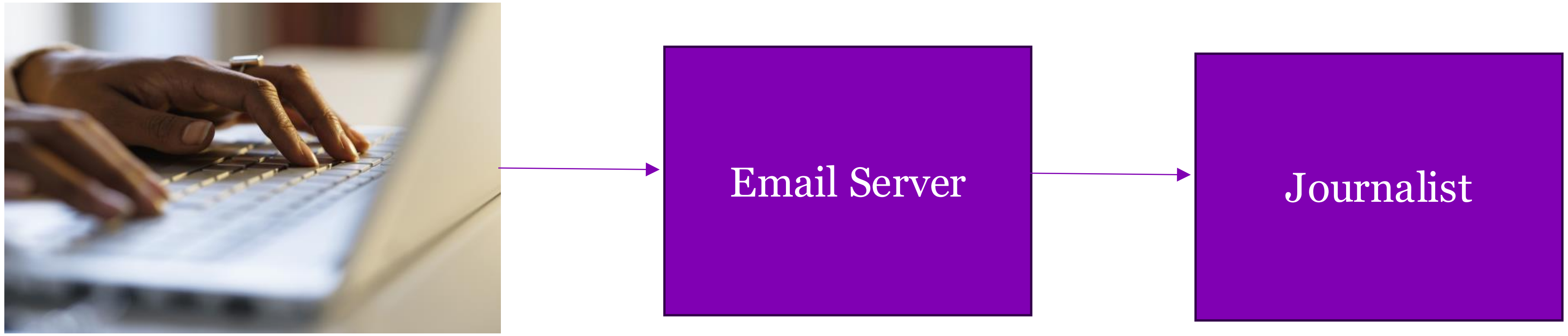
Information flow control

A guard controls whether a principal (the subject) is allowed access to a resource (the object).



This is the dual notion, sometimes used when confidentiality is the primary concern.

Information flow control



Email Server

Journalist

Biba Integrity Model

- Focus on the integrity of the data rather than the confidentiality
- Subjects S and Objects O have Integrity values
- **Simple Integrity Property** – subjects at a given level of integrity must not read data at a lower integrity level (no read down)
- * **Integrity Property** – subjects at a given level of integrity must not write to data at a higher level of integrity (no write up)
- **Invocation Property** – processes from below cannot request higher access; only with subjects at an equal or lower level

MULTILEVEL SECURITY MODELS

Multi-level security

- **Multi-level security** (MLS) systems originated in the military. A **security level** is a label for subjects and objects to describe a policy.
- These are models or ways of thinking about the problem of access control logically and are not implementations
- Security levels are ordered

  Unclassified ≤ Confidential ≤ Secret ≤ Top Secret

- Ordering is important since it can express policies like “no write down” to prevent a subject with high-level clearance from writing secrets into a low-level document

Running Example

Classifications (H)

- Admin
- Manager
- User

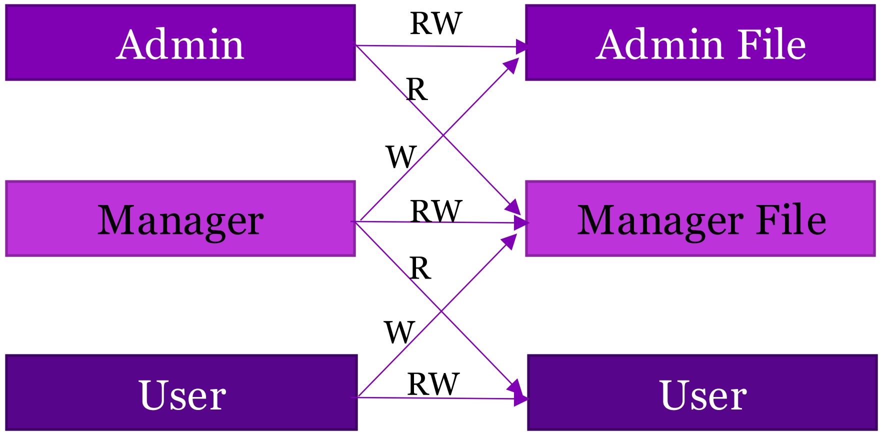
Bell-LaPadula

- Simple model of MLS designed to promote academic thought
  - **Simple Security Condition** – Subject S can read object O if and only if $L(O) \leq L(S)$
  - ***-Property (star property)** - Subject S can read object O if and only if $L(S) \leq L(O)$
- In other words:
  - No read up
  - No write down

Running Example

Classifications (H)

- Admin
- Manager
- User



Problem: most real systems don't fit neatly into clearence levels. It is rare that someone with a Top Security clearance really needs access to all Top Security files.

Solution: categories

Security lattice

- A lattice is a set L equipped with a partial ordering ≤ such every two elements a, b ∈ L has a least upper bound a ∨ b and a greatest lower bound a ∧ b. A finite lattice must have top and bottom elements.
- take a set of classifications H and linear ordering ≤H
- take a set C of categories; compartments are subsets of C
- security levels are pairs (h, c) with h ∈ H and c ⊆ C
- ordering (h1, c1) ≤ (h2, c2) ⇐⇒ h1 ≤ h2, c1 ⊆ c2 gives a lattice.

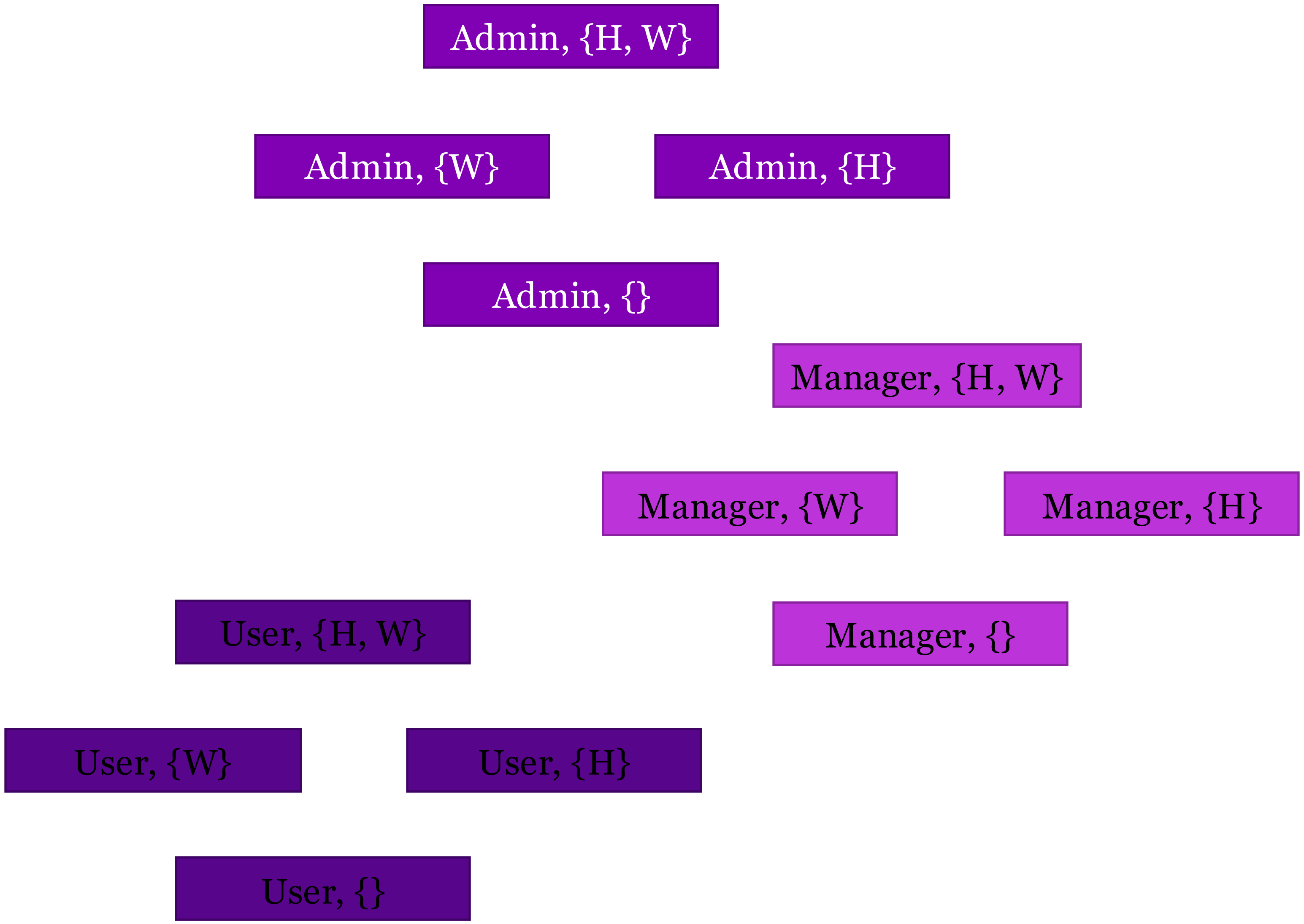
Running Example

Classifications (H)

- Admin
- Manager
- User

Categories (C)

- H (Hippo project)
- W (Walrus project)



Admin, {H, W}

Admin, {W}

Admin, {H}

Admin, {}

Manager, {H, W}

Manager, {W}

Manager, {H}

Manager, {}

User, {H, W}

User, {W}

User, {H}

User, {}

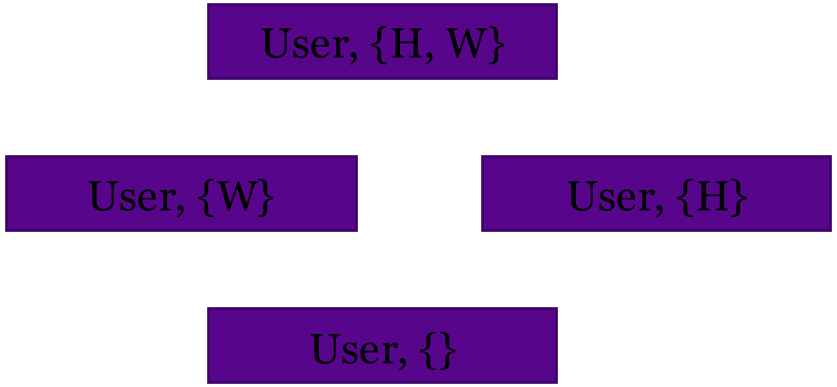
Running Example

Classifications (H)

- Admin
- Manager
- User

Categories (C)

- H (Hippo project)
- W (Walrus project)



User, {H, W}

User, {W}

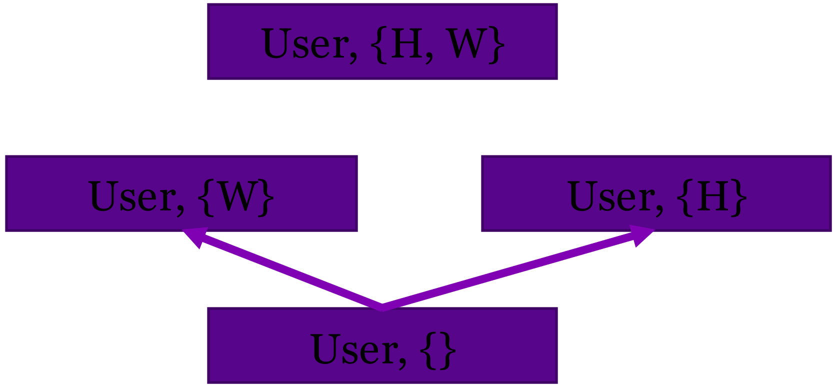
User, {H}

User, {}

Orderings:

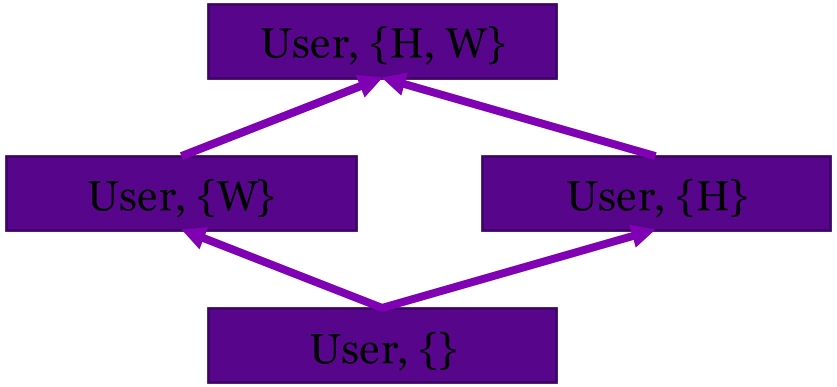
(User,{}) ≤ (User, {W})

(User,{}) ≤ (User, {H})



Orderings:

(User,{}) ≤ (User, {W})
(User,{}) ≤ (User, {H})
(User,{W}) ≤ (User, {H, W})
(User,{H}) ≤ (User, {H, W})



User, {H, W}

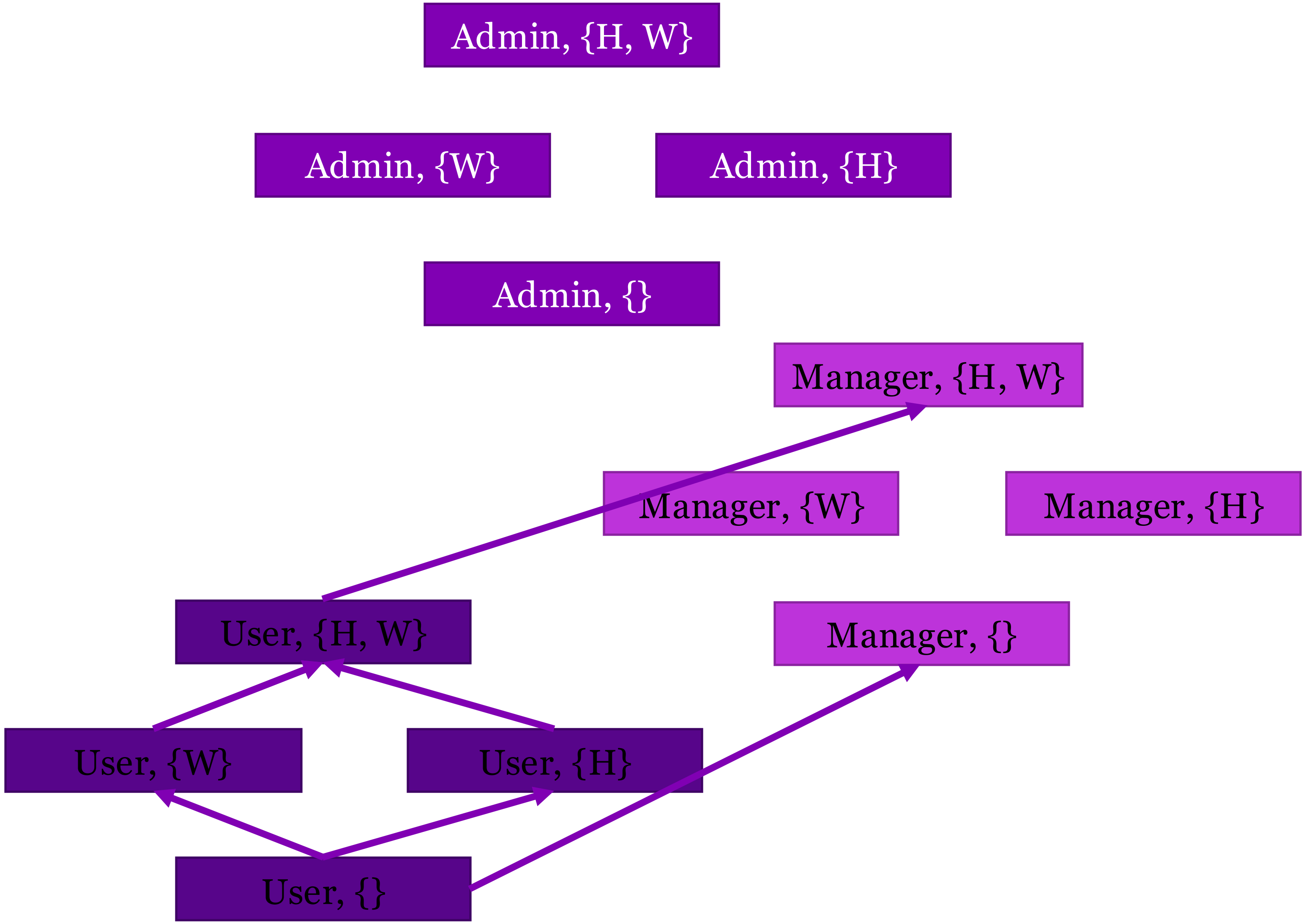
User, {W}

User, {H}

User, {}

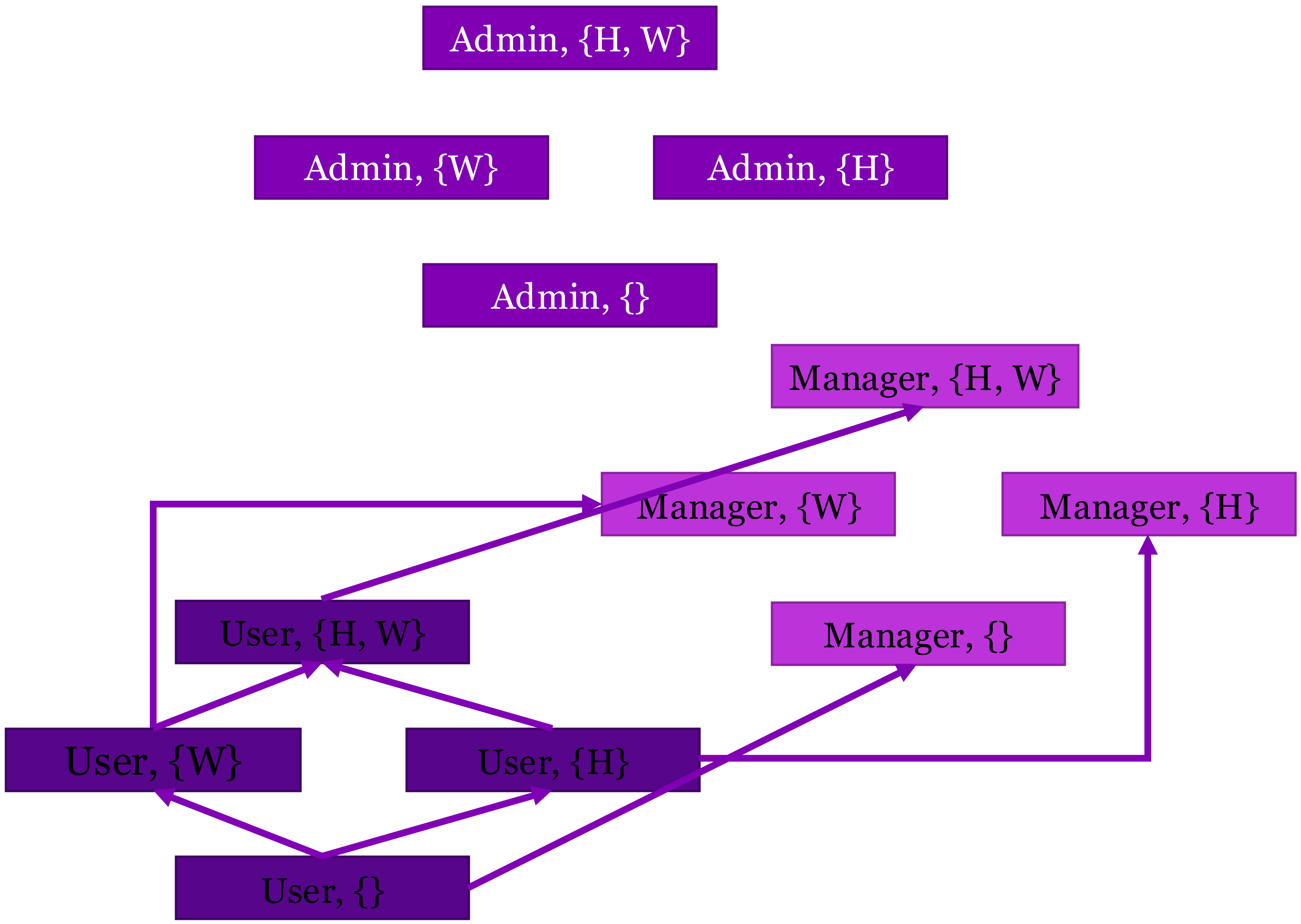
Orderings:

(User,{}) ≤ (User, {W})
(User,{}) ≤ (User, {H})
(User,{W}) ≤ (User, {H, W})
(User,{H}) ≤ (User, {H, W})
(User,{}) ≤ (Manager,{})
(User,{H,W})≤(Manager,{H,W})



Orderings:

(User,{}) ≤ (User, {W})
(User,{}) ≤ (User, {H})
(User,{W}) ≤ (User, {H, W})
(User,{H}) ≤ (User, {H, W})
(User,{}) ≤ (Manager,{})
(User,{H,W})≤(Manager,{H,W})
(User,{W}) ≤ (Manager,{W})
(User,{H}) ≤ (Manager,{H})



Admin, {H, W}

Admin, {W}

Admin, {H}

Admin, {}

Manager, {H, W}

Manager, {W}

Manager, {H}

User, {H, W}

Manager, {}

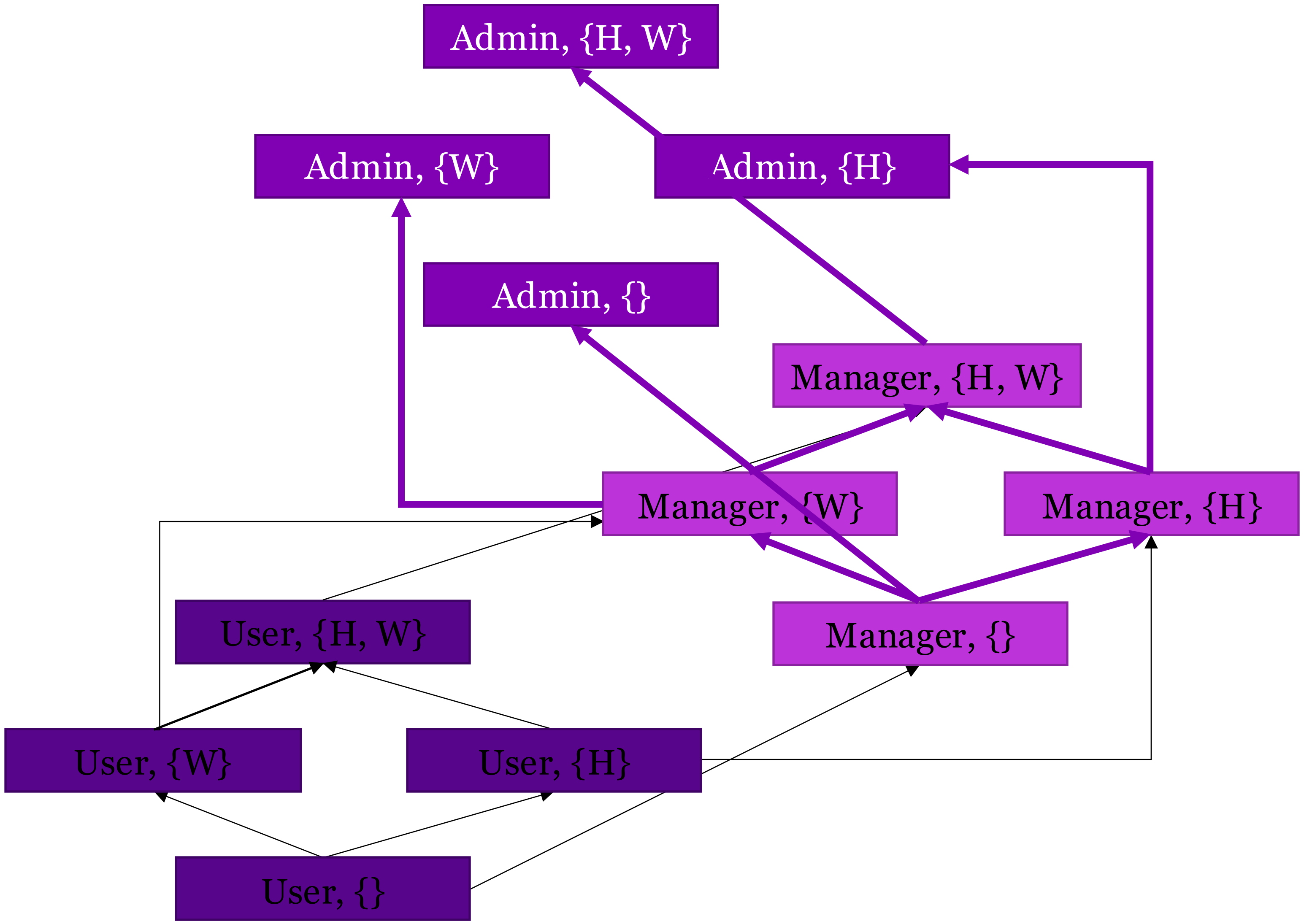
User, {W}

User, {H}

User, {}

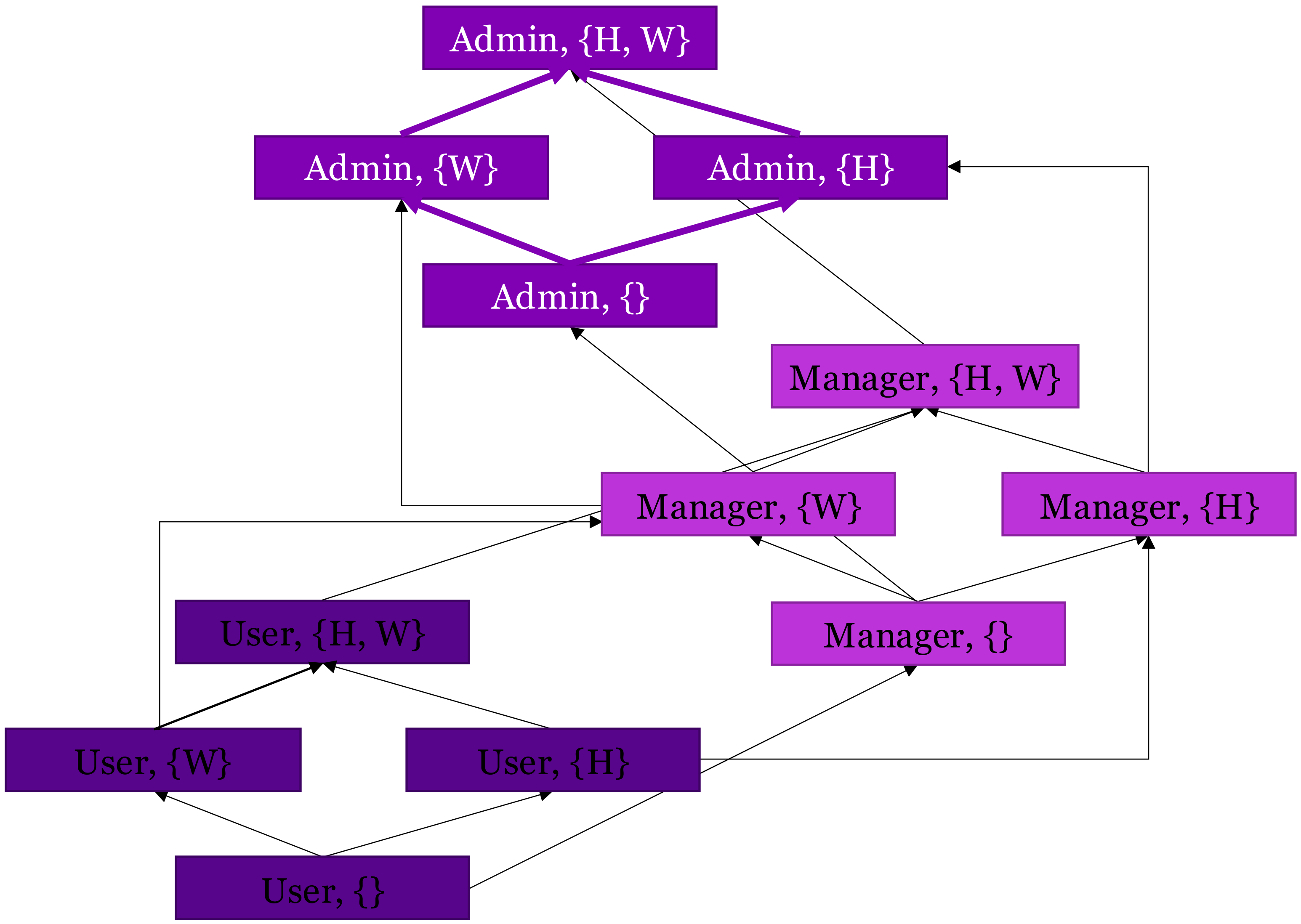
Orderings:

(Manager,{}) ≤ (Manager, {W})
(Manager,{}) ≤ (Manager, {H})
(Manager,{W}) ≤ (Manager, {H, W})
(Manager,{H}) ≤ (Manager, {H, W})
(Manager,{}) ≤ (Admin,{})
(Manager,{H,W})≤(Admin,{H,W})
(Manager,{W}) ≤ (Admin,{W})
(Manager,{H}) ≤ (Admin,{H})



Orderings:

(Admin,{}) ≤ (Admin, {W})
(Admin,{}) ≤ (Admin, {H})
(Admin,{W}) ≤ (Admin, {H, W})
(Admin,{H}) ≤ (Admin, {H, W})



Biba Integrity Model

- Focus on the integrity of the data rather than the confidentiality
- Subjects S and Objects O have Integrity values
- **Simple Integrity Property** – subjects at a given level of integrity must not read data at a lower integrity level (no read down)
- * **Integrity Property** – subjects at a given level of integrity must not write to data at a higher level of integrity (no write up)
- **Invocation Property** – processes from below cannot request higher access; only with subjects at an equal or lower level

Think-pair-share

- How can covert channels still happen under:
- Bell-LaPaula
- Biba

QUESTIONS