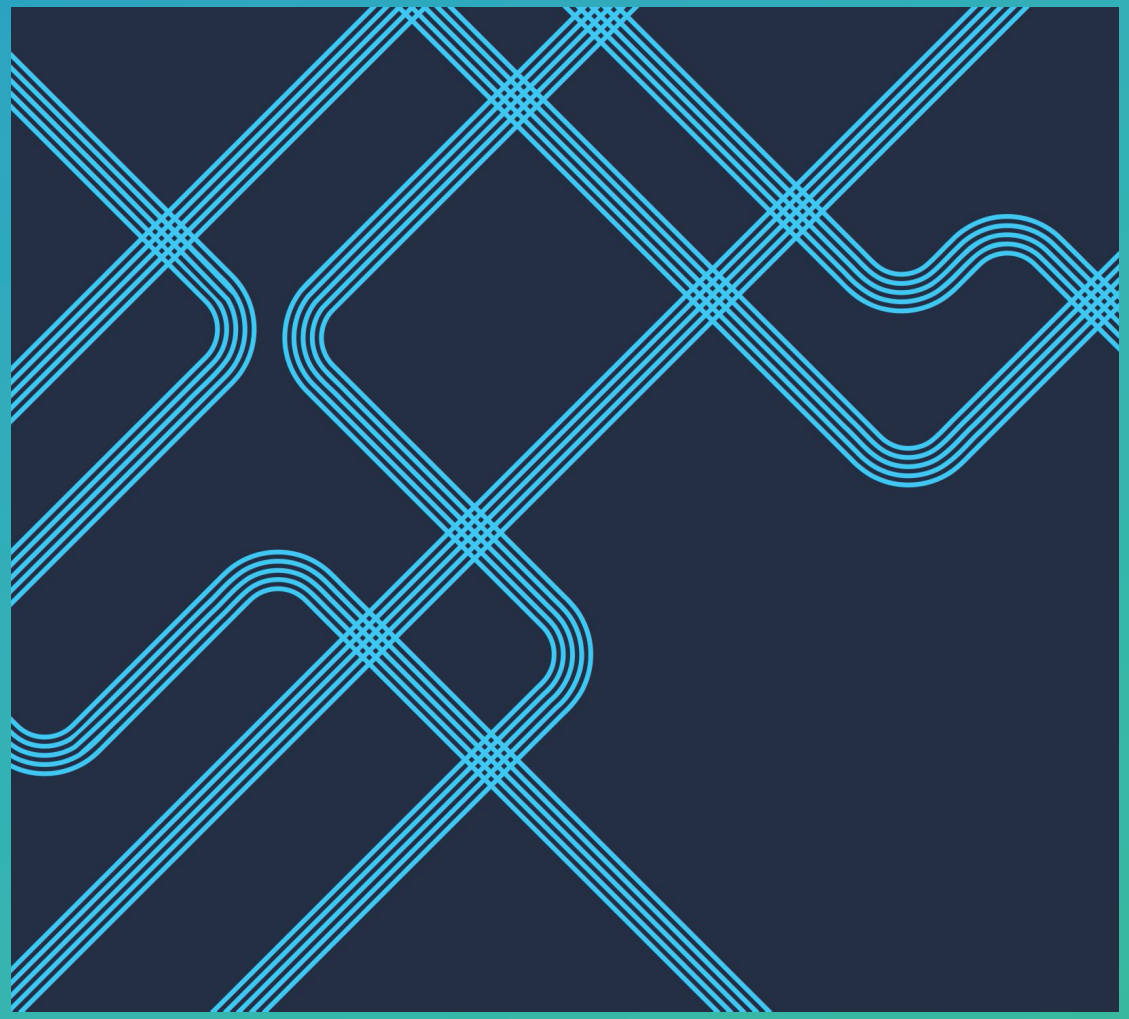
THE PROTECTION OF INFORMATION

Jamie Tan



Motivation

Bottom-up examination of information protection systems

BASIC PRINCIPLES

Security Violations

Confidentiality

Unauthorized information release

Integrity

Unauthorized information modification

Accessibility

Unauthorized denial of use

Protection Schemes by Function

Unprotected

All-or-nothing

Controlled sharing

User-programmed sharing

Strings on Information

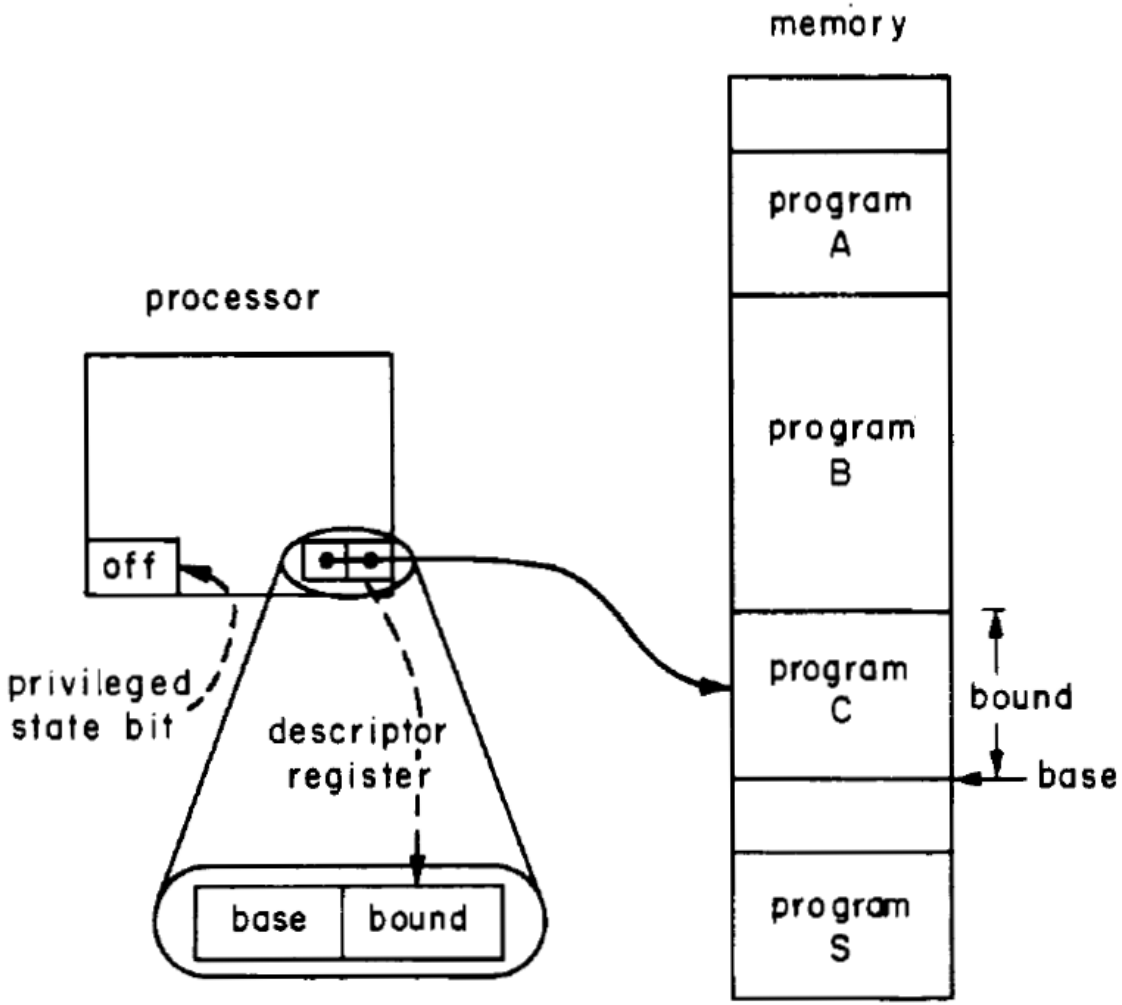
Dynamics of use

- *How to establish and change access specifications*

Design Principles

- Economy of Mechanism
- Fail-safe Defaults
- Complete Mediation
- Open Design
- Separation of Privilege
- Least Privilege
- Least Common Mechanism
- Psychological Acceptability

Basic Separation



Authentication

Passwords

- ✅ Simple
- ✅ Intuitive
- 🚫 pass123
- 🚫 Must expose

Unforgeable object

- ✅ Convenient
- ✅ Hard to misuse
- 🚫 Need physical security

Enciphering

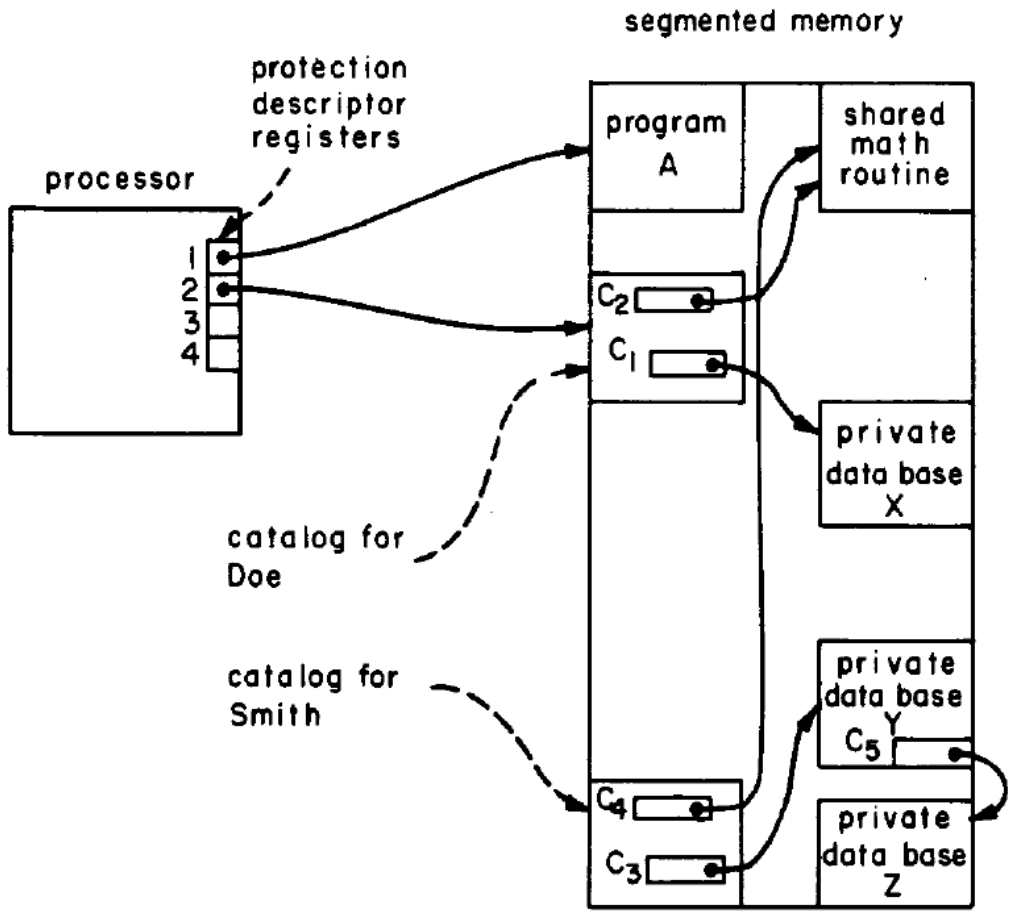
- ✅ Two-way system
- 🚫 Most complex

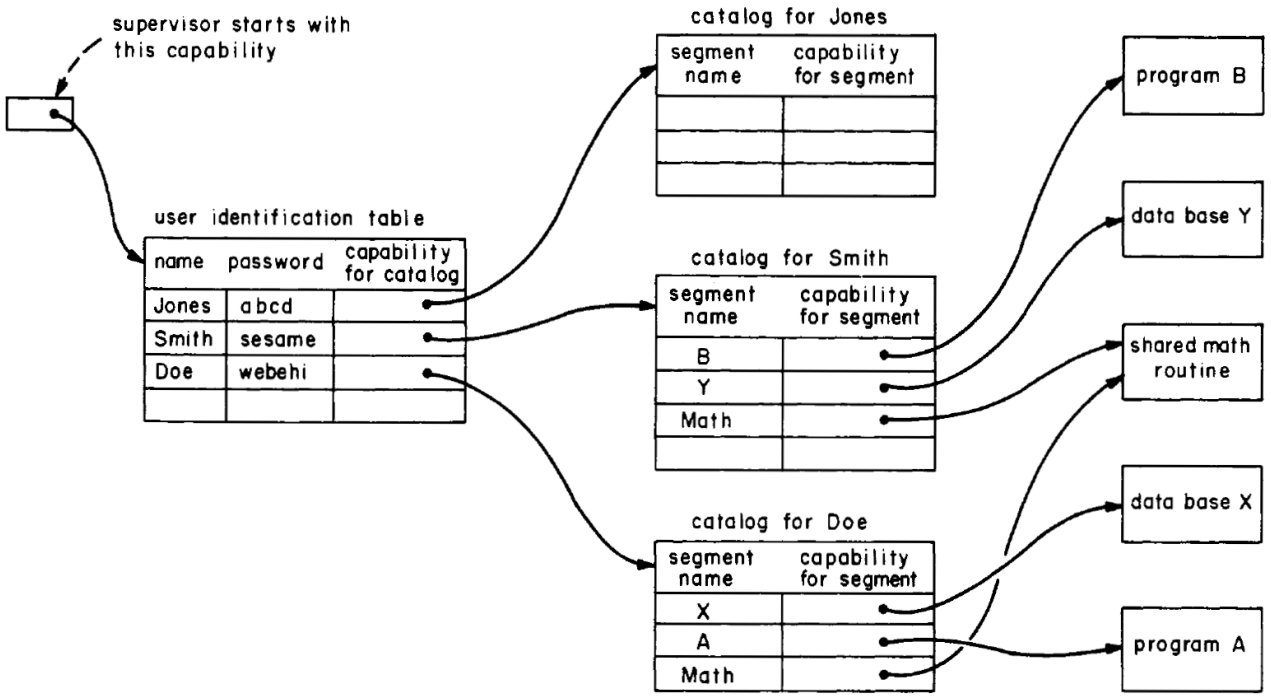
Shared Information Mechanisms

List	Ticket
List-oriented • Access allowed based on identifiers of authorized principles • Authenticated = id on list	Ticket-oriented • Access allowed if requestor has appropriate (unforgeable) ticket • Authenticated = has a ticket

DESCRIPTOR-BASED SYSTEMS

Capability System





Capability System

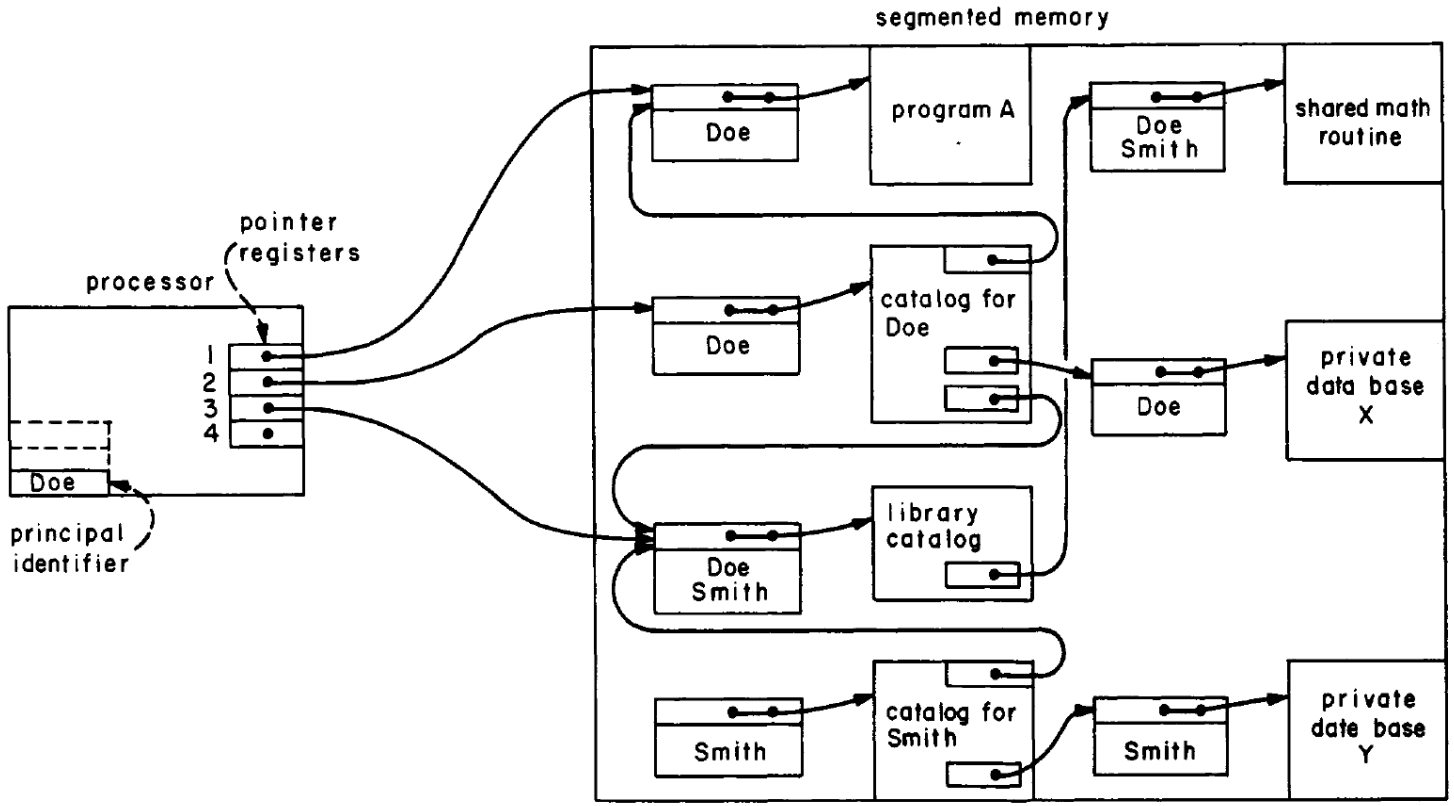
Benefits

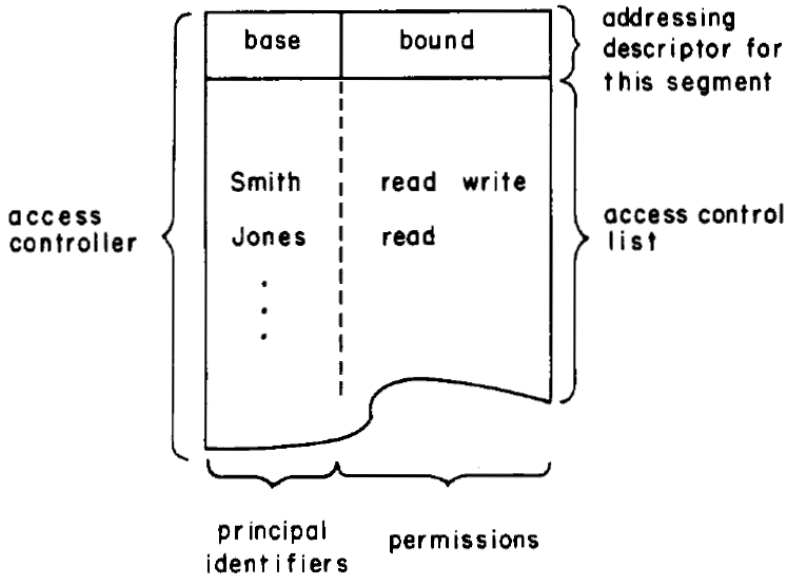
- Efficiency
- Simplicity
- Flexibility

Drawbacks

- Revocation of access
- Control of propagation
- Review of access

Access Control List System





ACL System

Benefits

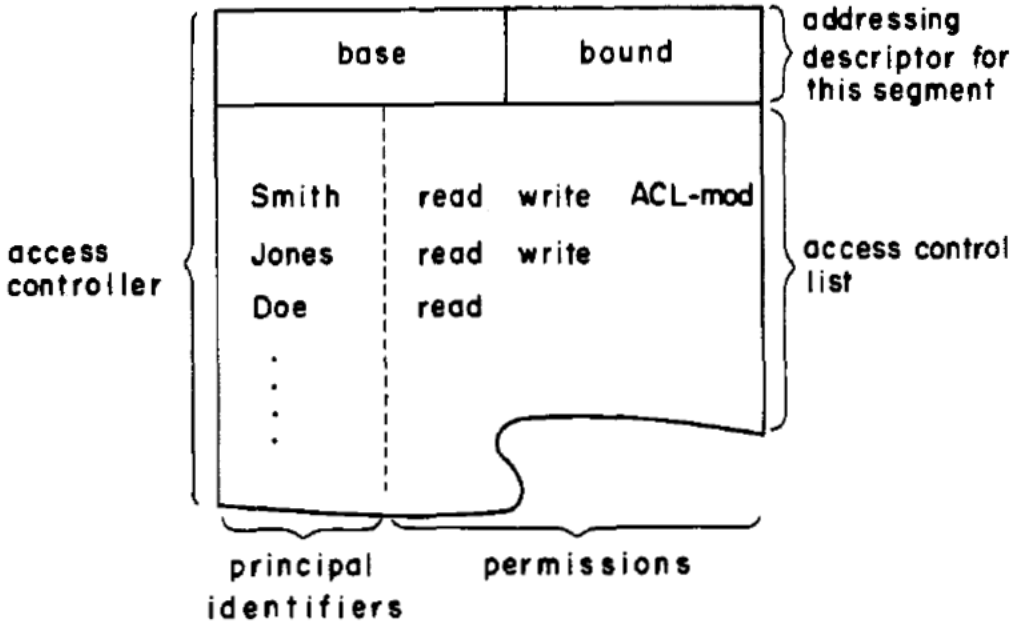
- Prevents unauthorized copying
- Easy revocation & review
- No unnecessary data-user association

Drawbacks

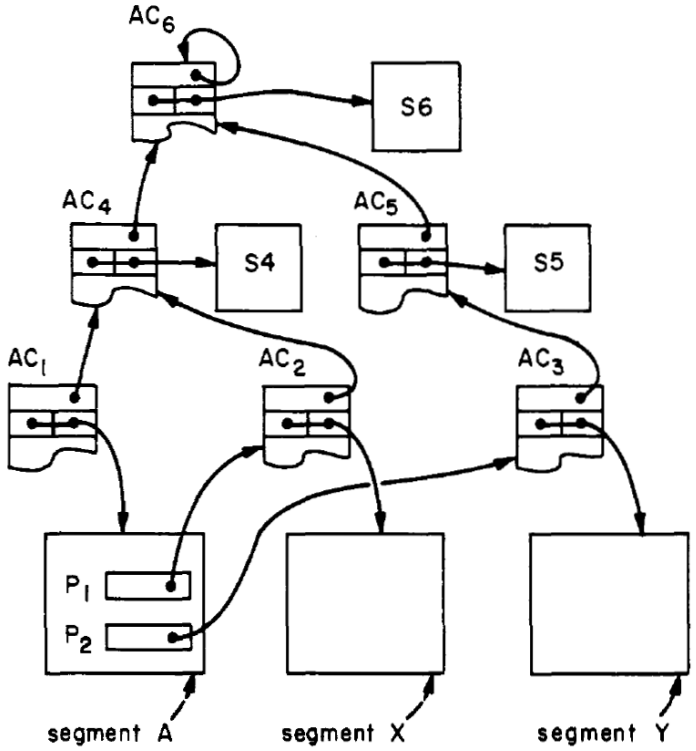
- Slow memory references
- ACL search complexity
- Space allocation

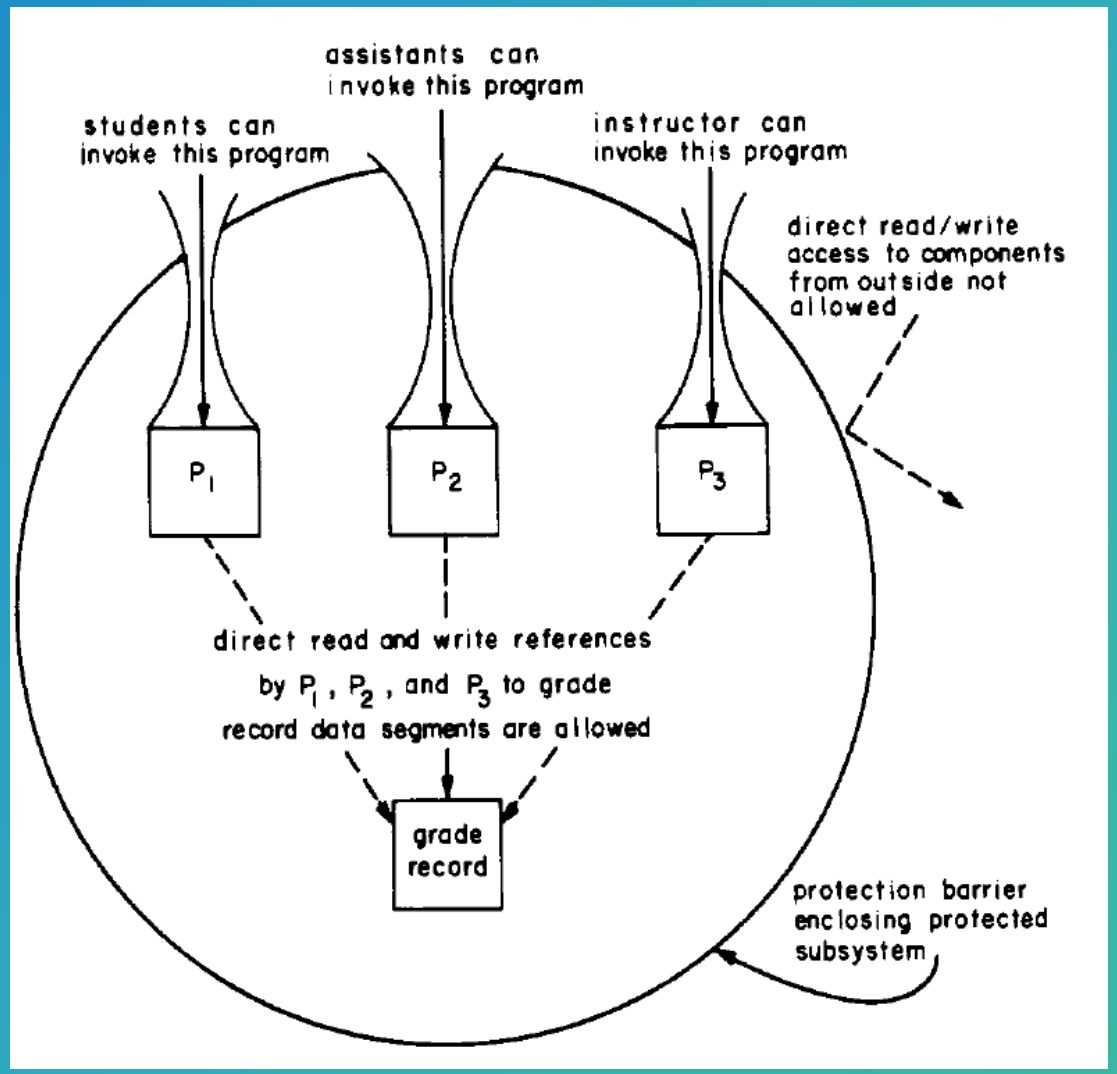
Dynamics of ACLs

Self-Control



Hierarchical Control





Protected Subsystems

- Program and data encapsulated from other programs
- Can be implemented using ACL or capacities
- Bare essentials for domain switching

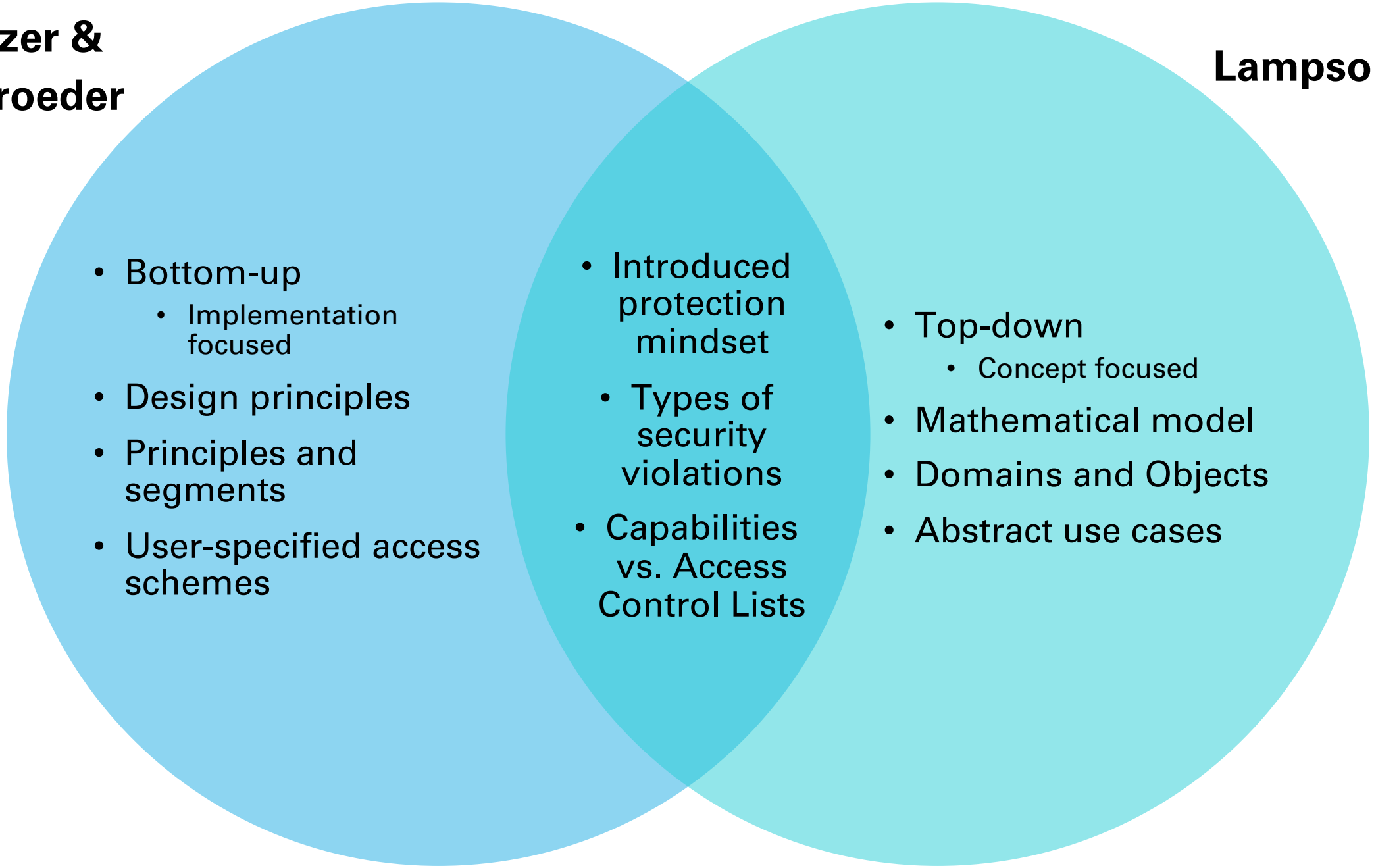
Special Use Cases/Extensions

- Prescript
  - Specific rules for modifying another user's ACL (ex. court order)
- Protecting non-segment objects
  - Lists, Queues, ACLs

ANALYSIS

Key Takeaways

- Certain design principles should guide the structure of a secure protection system.
  - Still integral in security design to this day.
- Protection systems are motivated by use cases.
  - Essentially distinguished by how each handles dynamic updates
  - Mechanisms and architectures should be well matched to user's image/model of the problem to be solved.
- Most systems use a hybrid capability and ACL system.
  - Capability (ticket) for hardware implementation
  - ACL (list) for human interface


Saltzer & Schroeder
Lampson
• Bottom-up
• Implementation focused
• Design principles
• Principles and segments
• User-specified access schemes
• Introduced protection mindset
• Types of security violations
• Capabilities vs. Access Control Lists
• Top-down
• Concept focused
• Mathematical model
• Domains and Objects
• Abstract use cases

Discussion Questions

- What are the advantages of a top-down compared to a bottom-up approach? Vice versa?
- Are certain security violation types more dangerous in certain protection schemes? Why or why not?
- Why are the user-programmed sharing and strings on information protection schemes so rare?
  - Are there any modern systems that account for these schemes?
- For each of the design principles, which of the systems we've studied this year best embodies that principle?
  - What principles do modern systems seem to value the most? The least?
- Why have systems tended toward a hybrid ACL-Capacity system? Is this evolution pattern present on an OS level? (ex. Monolithic + Micro -> Hybrid)