FARABI UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Object-Oriented Analysis and Design

Lecture 1
Introduction to Software Architecture and Design



Dr. Bazargul Matkerim, Professor-Researcher

Learning Outcomes

Understand

Explain the purpose of architecture.

Distinguish

Separate analysis from design.

Apply

Model a small library system.

Learning path: from system structure to collaborating objects.

Architecture Organizes a Whole



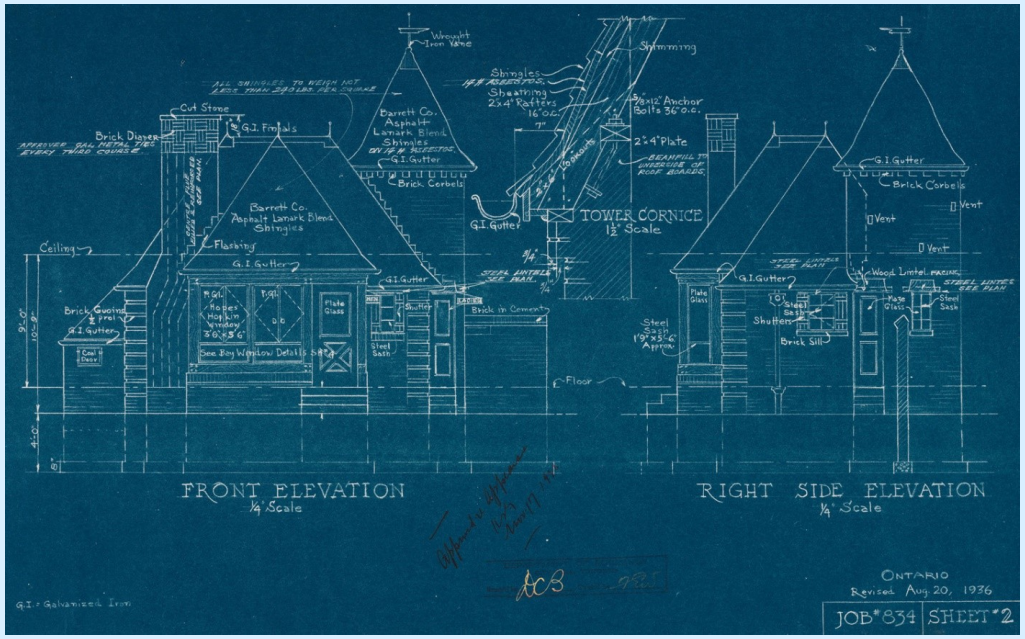


- Parts have distinct purposes.
- Connections enable movement and use.
- Constraints influence the whole design.

Building examples: structure supports purpose.

Software architecture also organizes parts, relationships and constraints.

From Blueprint to Software Model



Software model

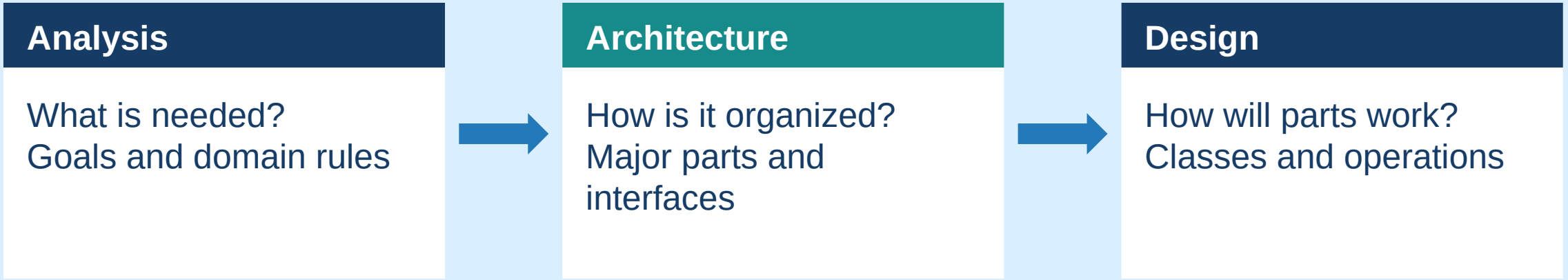
Components
Interfaces
Interactions

A building plan communicates spatial decisions.

A software model communicates system decisions.

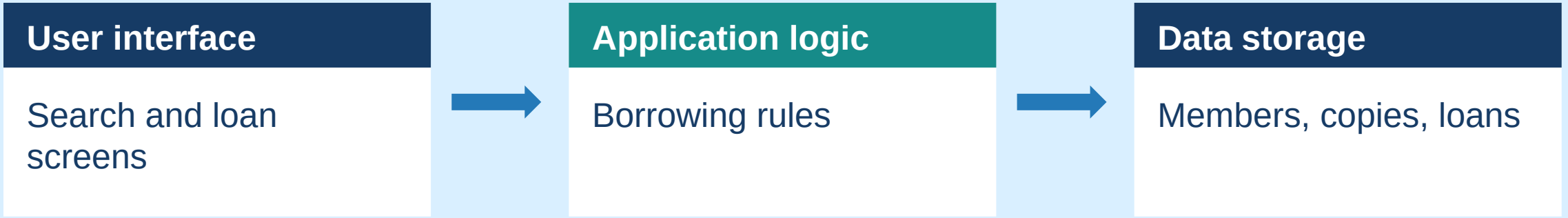
Models simplify a system so people can discuss and evaluate it.

Analysis, Architecture and Design



Feedback can send the team back to refine any earlier decision.

A System Has Parts and Connections



Architecture defines responsibilities and the interfaces between parts.

Requirements Shape the Architecture

Functional requirement

A member can borrow an available book copy.

Quality requirement

Only authorized staff can change member records.



Design implications

Loan rules in one place
Access checks
Recorded transactions

Requirements provide reasons for architectural decisions.

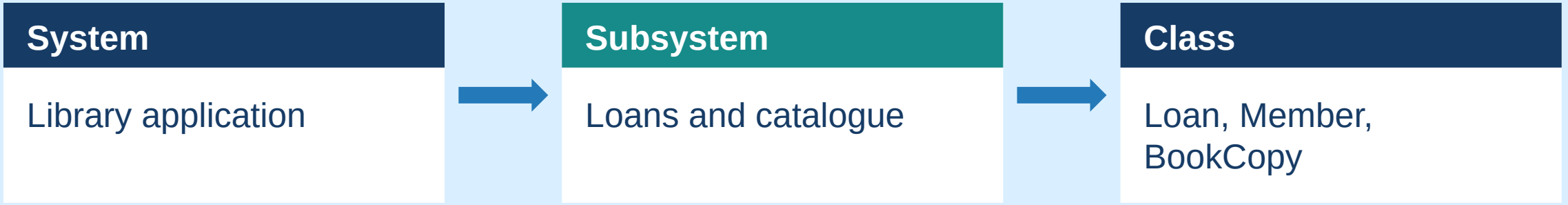
Quality Goals Require Trade-offs

Performance	Security	Maintainability
Fast search responses	Controlled access	Changes stay localized

Example: caching can speed up search, but it introduces rules for refreshing data and avoiding stale availability information.

Evaluate trade-offs against the system’s priorities.

Architecture at Different Levels



Broad decisions

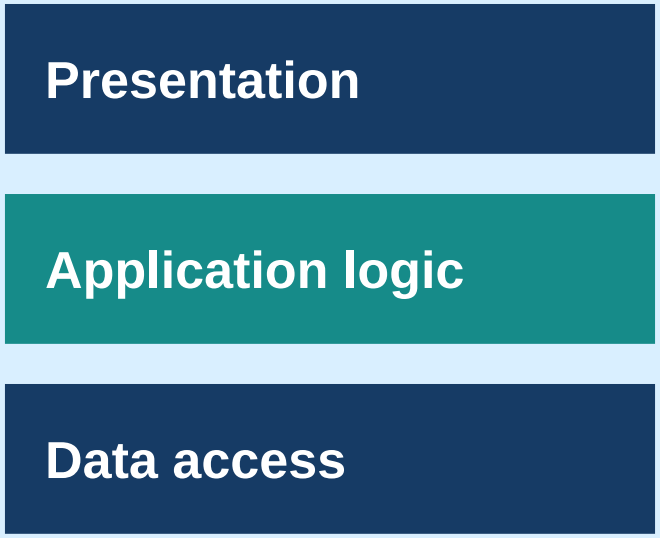
Responsibility boundaries

Detailed operations

Choose the level of detail that answers the current design question.

Common Architectural Styles

Layered



Client–server

Client requests
Server responds

Service-oriented

Separate services
Defined interfaces
Coordinated workflows

Architectural styles describe organization. Select them for a reason.

Several Views Describe One System

Business view

Goals and workflows

Application view

Software responsibilities

Information view

Data and relationships

Technology view

Runtime and deployment

These perspectives answer different questions about the same system.

Software Design Assigns Responsibilities



- Define operations and interfaces.
- Allocate behaviour to suitable classes.
- Guide implementation and testing.

A design plan explains how the elements work together.

Object-Oriented Analysis

Scenario: A member borrows an available book copy.

Member	BookCopy	Loan
Who borrows?	Which physical copy?	When is it due?

Analysis identifies domain concepts and the rules connecting them.

Object-Oriented Design

Member

Knows membership status
Can check eligibility

BookCopy

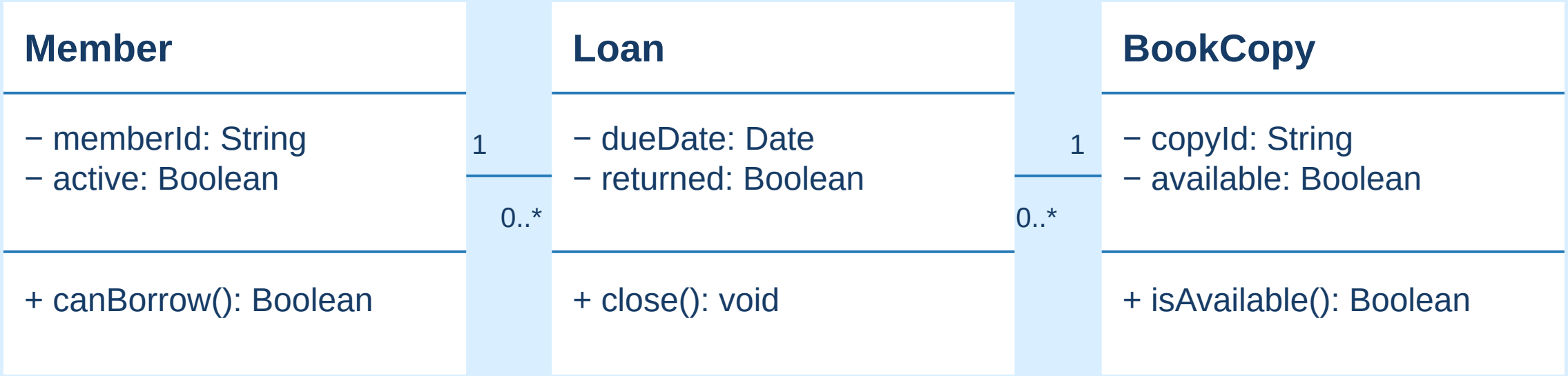
Knows copy identity
Can report availability

Loan

Records borrowing
Tracks the due date

Design assigns state and behaviour to cooperating software objects.

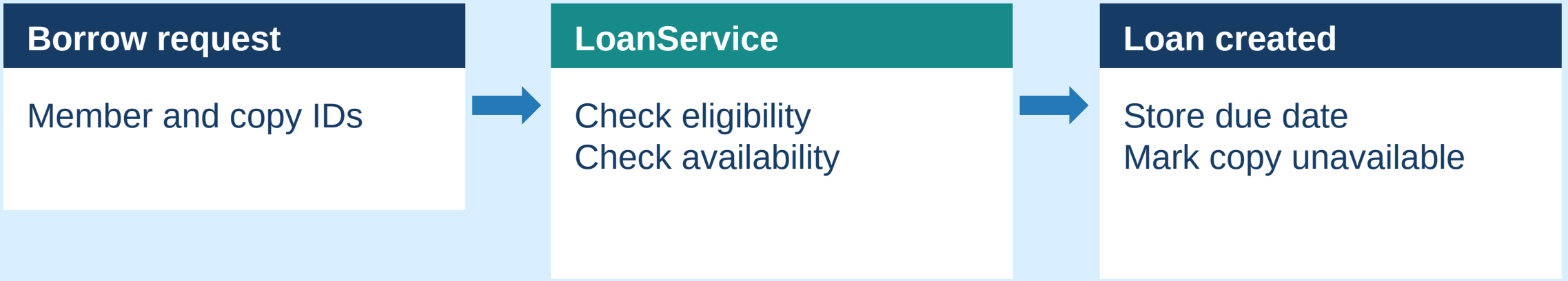
UML Makes the Design Visible



Each loan links one member to one copy. Historical loans may be many.

UML visibility: + public operation − private attribute

Objects Collaborate to Fulfil a Goal



An interaction view shows order, collaboration and alternative outcomes.

The Software Architect’s Role

Domain knowledge

Clarify needs and rules

Design knowledge

Evaluate structures

Technology knowledge

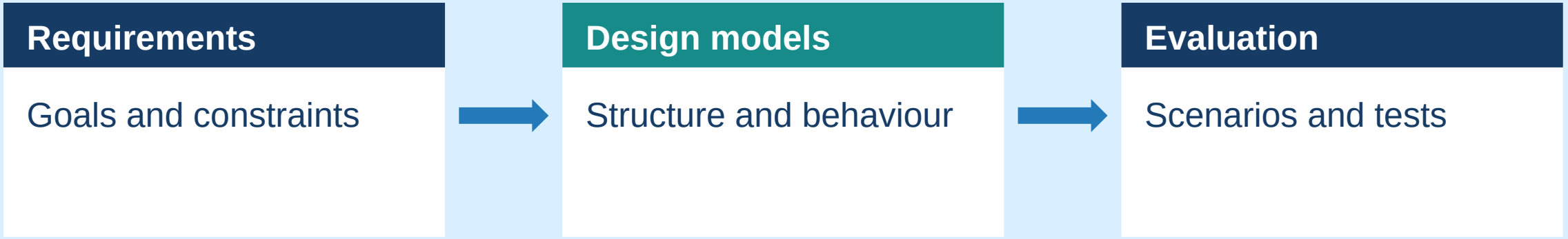
Assess feasible options

Team coordination

Explain decisions and risks

The architect connects stakeholder goals with an implementable solution.

Design Deliverables and Feedback



Decision records explain the options considered, the chosen approach, and the consequences that future maintainers need to understand.

Review the design as requirements and implementation evolve.

Design Risks and Practical Responses

Risk	Response
Unclear requirements	Validate scenarios
Tightly coupled parts	Define clear interfaces
Untested assumptions	Prototype risky decisions



Models support reasoning. Tests and feedback provide additional evidence.

Practice and Review

Draw	Explain	Evaluate
Sketch the main parts of a library application.	Assign responsibilities to Member, Loan and BookCopy.	Describe what happens when a copy is unavailable.

Review: How do analysis, architecture and detailed design differ?