Software Engineering

Lesson #05 - Lecture

Lesson #04 - Lecture

Your KBTU 202309 Software Engineering class information is updating …

Lesson #05 update is in progress

This will take around 2 hours to complete

Please, don’t turn off your head

Architectural design

Architectural design

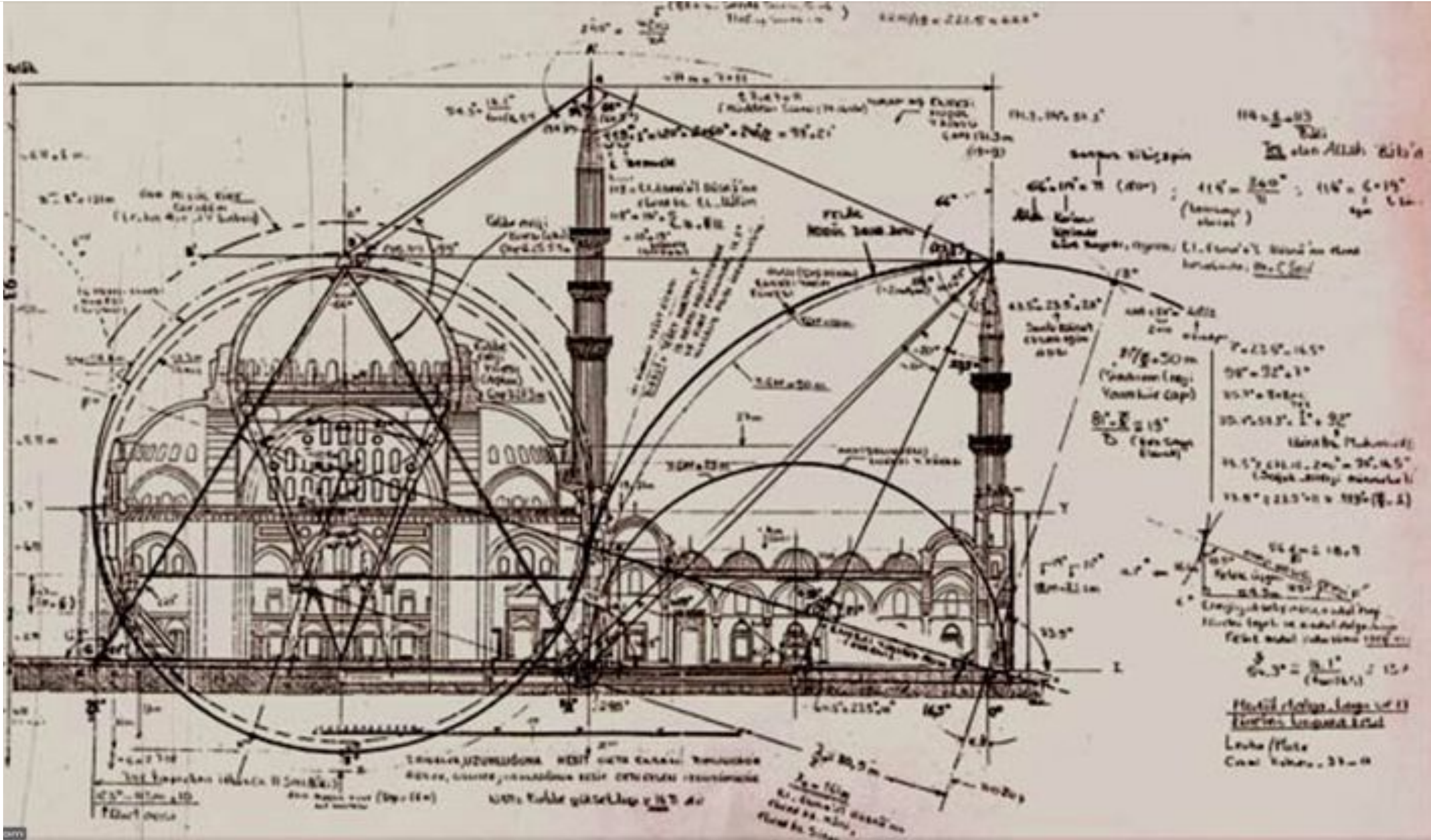
Architectural design

```
#include<iostream>
Using namespace std;

int main()
{
      cout << “Architectural design” << endl;

      return 0;
}
```

MIMAR SINAN ESERI



Agenda: Lesson #05 - Software Engineering - Lecture

1 Architectural design decisions

2 Architectural views

3 Architectural patterns

4 Application architectures

Agenda: Lesson #05 - Software Engineering - Lecture

1 **Architectural design decisions**

2 Architectural views

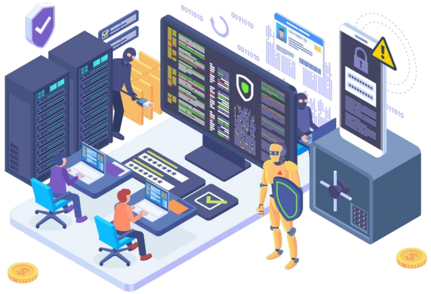
3 Architectural patterns

4 Application architectures

Architectural design decisions

Architectural design

Architectural design is concerned with understanding how a software system should be organized and designing the overall structure of that system



Architectural design

Architectural design is the first stage in the software design process

It is the critical link between design and requirements engineering, as it identifies the main structural components in a system and the relationships between them

Architectural design

You can design software architectures at two levels of abstraction, which I call architecture in the small and architecture in the large:

- Architecture in the small is concerned with the architecture of individual programs. At this level, we are concerned with the way that an individual program is decomposed into components.

Architectural design

You can design software architectures at two levels of abstraction, which I call architecture in the small and architecture in the large:

- Architecture in the large is concerned with the architecture of complex enterprise systems that include other systems, programs, and program components. These enterprise systems may be distributed over different computers, which may be owned and managed by different companies.

Architectural design

Bass et al. (Bass, Clements, and Kazman 2012) suggest that explicitly designing and documenting software architecture has three advantages:

- Stakeholder communication
- System analysis
- Large-scale reuse

Architectural design

The apparent contradictions between architectural theory and industrial practice arise because there are two ways in which an architectural model of a program is used:

- As a way of encouraging discussions about the system design
- As a way of documenting an architecture that has been designed

Architectural design

Block diagrams are a good way of supporting communications between the people involved in the software design process

They are intuitive, and domain experts and software engineers can relate to them and participate in discussions about the system

Architectural design decisions

Architectural design is a creative process in which you design a system organization that will satisfy the functional and non-functional requirements of a system

There is no formulaic architectural design process

Architectural design decisions

During the architectural design process, system architects have to make a number of structural decisions that profoundly affect the system and its development process

Architectural design decisions

The architecture of a software system may be based on a particular Architectural pattern or style

Architectural design decisions

Because of the close relationship between non-functional system characteristics and software architecture, the choice of architectural style and structure should depend on the non-functional requirements of the system:

- Performance
- Security
- Safety
- Availability
- Maintainability

Architectural design decisions

Evaluating an architectural design is difficult because the true test of an architecture is how well the system meets its functional and non-functional requirements when it is in use

Architectural design decisions

Architectural Styles

Types of Architectural Styles - Georgia Tech - Software Development Process

https://www.youtube.com/watch?v=JLbo9Lvvy5M&t=84s

Agenda: Lesson #05 - Software Engineering - Lecture

1 Architectural design decisions

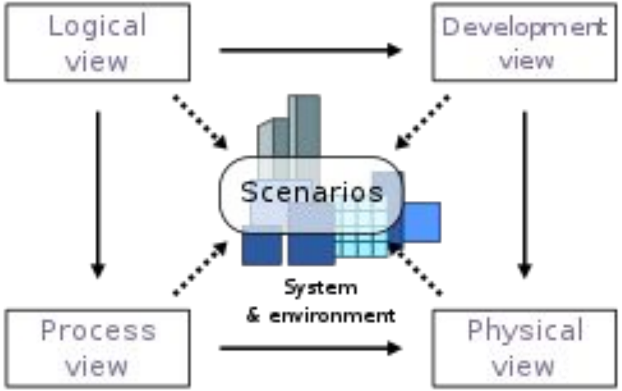
2 Architectural views

3 Architectural patterns

4 Application architectures

Architectural views

It is impossible to represent all relevant information about a system's architecture in a single diagram, as a graphical model can only show one view or perspective of the system



Architectural views

Architectural views

Architectural Views - Udacity

https://www.youtube.com/watch?v=xDi_6vwfhIY

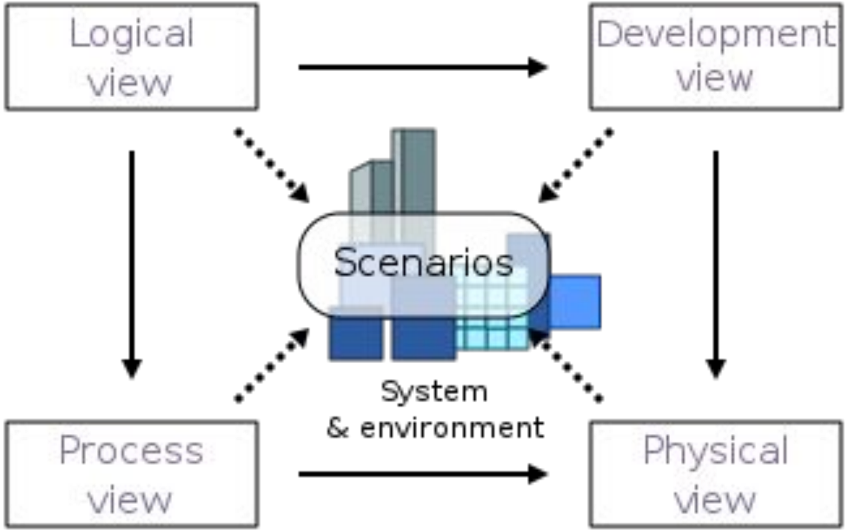
Architectural views

Architectural Views - Udacity

https://www.youtube.com/watch?v=be9JThe7Ul8

Architectural views

Well-known 4 +1 view model of software architecture



Architectural views

4.6 SEM: 4+1 view into software architecture

https://www.youtube.com/watch?v=5r60GuRu8V0&t=25s

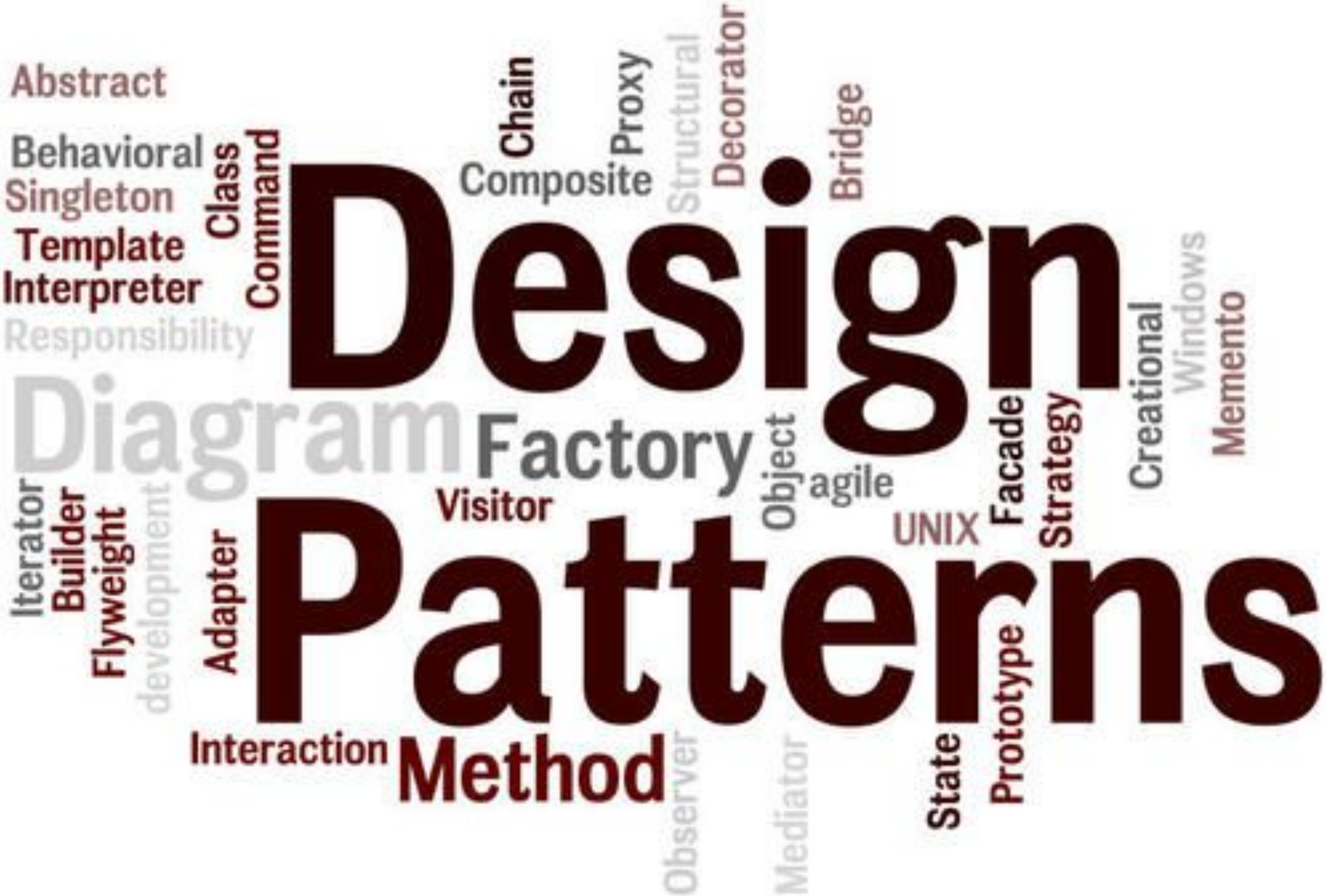
Architectural views

In practice, conceptual views of a system’s architecture are almost always developed during the design process

They are used to explain the system architecture to stakeholders and to inform architectural decision making

Agenda: Lesson #05 - Software Engineering - Lecture

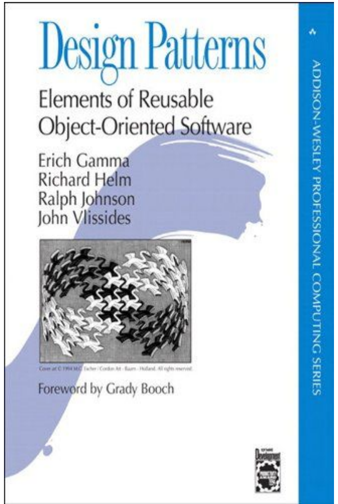
1. Architectural design decisions
2. Architectural views
3. **Architectural patterns**
4. Application architectures


Design
Patterns
Abstract
Behavioral
Singleton
Template
Interpreter
Responsibility
Class
Command
Chain
Composite
Proxy
Structural
Decorator
Bridge
Diagram
Factory
Object
agile
Visitor
UNIX
Facade
Strategy
Creational
Windows
Memento
Iterator
Builder
Flyweight
development
Adapter
Interaction
Method
Observer
Mediator
State
Prototype

Architectural patterns

The idea of patterns as a way of presenting, sharing, and reusing knowledge about software systems has been adopted in a number of areas of software engineering

The trigger for this was the publication of a book on object-oriented design patterns (Gamma et al. 1995)


Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

Architectural patterns

Patterns may be described in a standard way using a mixture of narrative description and diagrams

You can think of an Architectural pattern as a stylized, abstract description of good practice, which has been tried and tested in different systems and environments

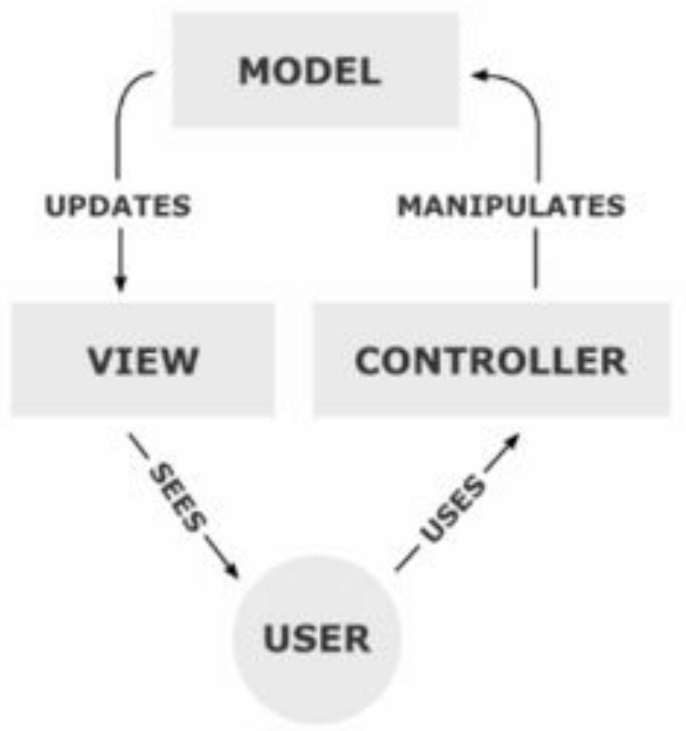
Architectural patterns

Architectural Patterns

https://www.youtube.com/watch?v=exJ8oLnq4dM

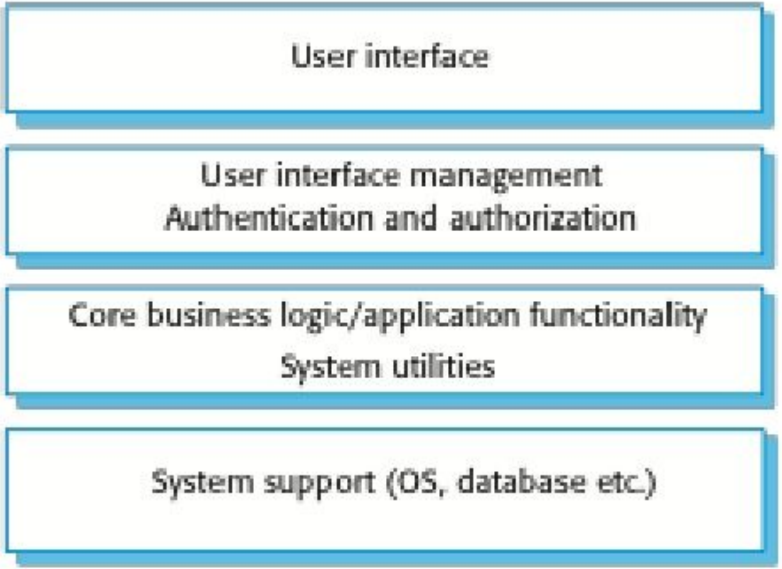
Architectural patterns

MVC (Model-View-Controller) pattern



Architectural patterns

Layered Architecture



User interface

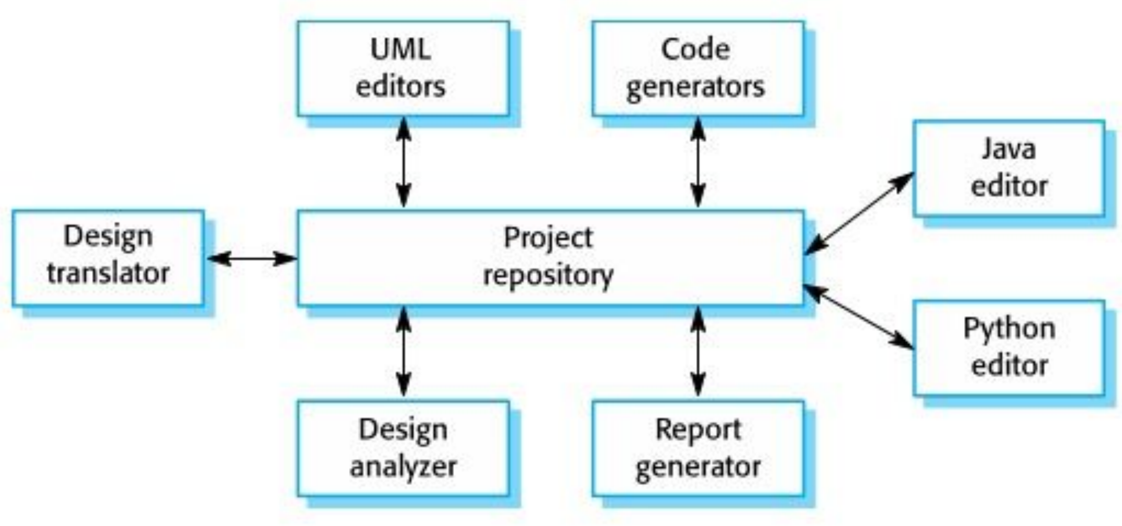
User interface management
Authentication and authorization

Core business logic/application functionality
System utilities

System support (OS, database etc.)

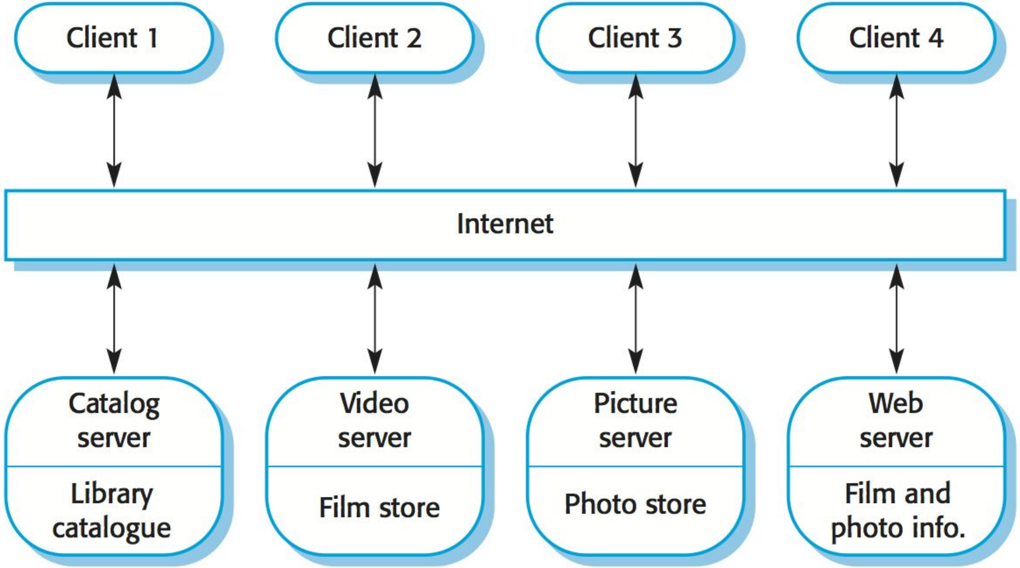
Architectural patterns

Repository Architecture



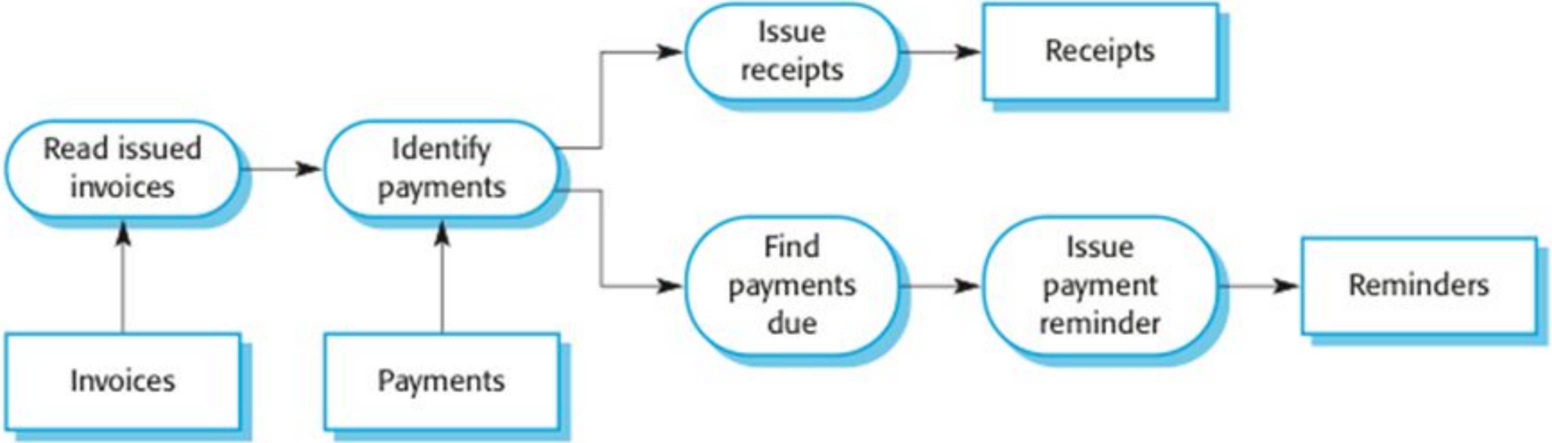
Architectural patterns

Client-Server Architecture



Architectural patterns

Pipe & filter Architecture



Architectural patterns

Architectural patterns for real-time systems

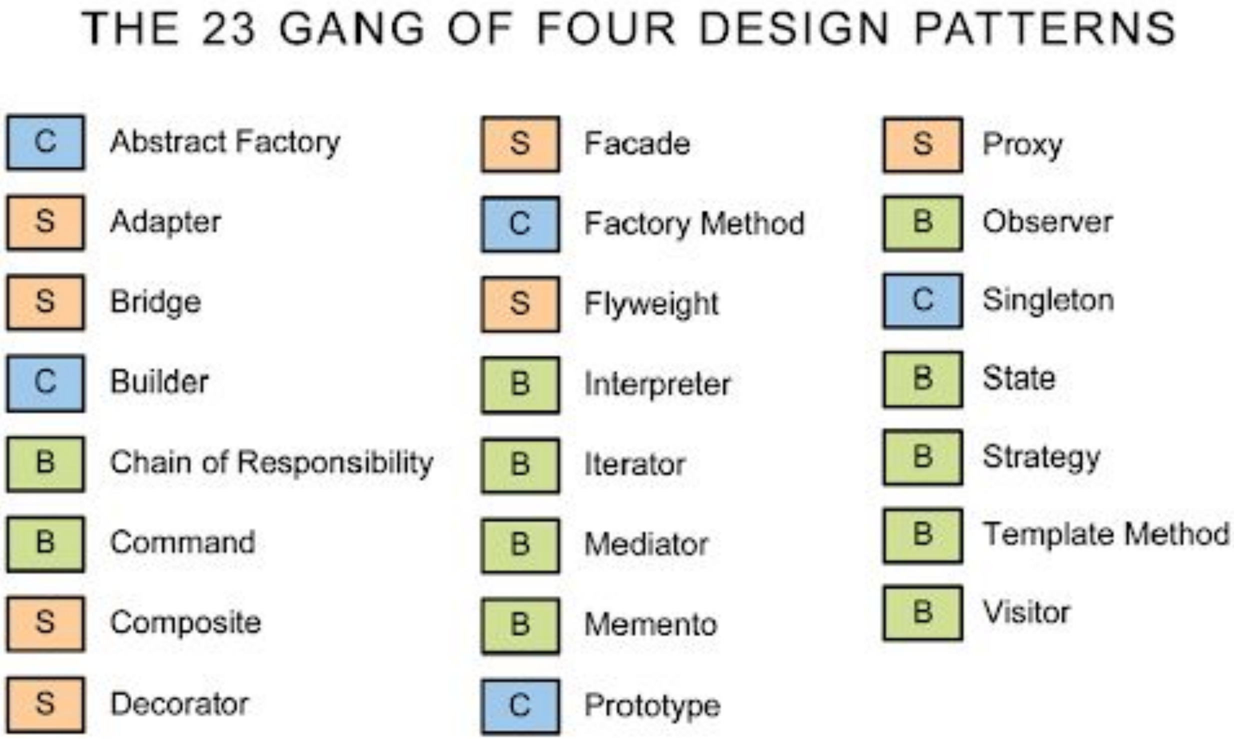
https://www.youtube.com/watch?v=OmRVgmVgt4Y&t=118s

Architectural patterns

Software Design Patterns and Principles (quick overview)

https://www.youtube.com/watch?v=WV2Ed1QTst8&t=427s

Architectural patterns



THE 23 GANG OF FOUR DESIGN PATTERNS

C	Abstract Factory
S	Adapter
S	Bridge
C	Builder
B	Chain of Responsibility
B	Command
S	Composite
S	Decorator
S	Facade
C	Factory Method
S	Flyweight
B	Interpreter
B	Iterator
B	Mediator
B	Memento
C	Prototype
S	Proxy
B	Observer
C	Singleton
B	State
B	Strategy
B	Template Method
B	Visitor

Agenda: Lesson #05 - Software Engineering - Lecture

1 Architectural design decisions

2 Architectural views

3 Architectural patterns

4 Application architectures

Application architectures

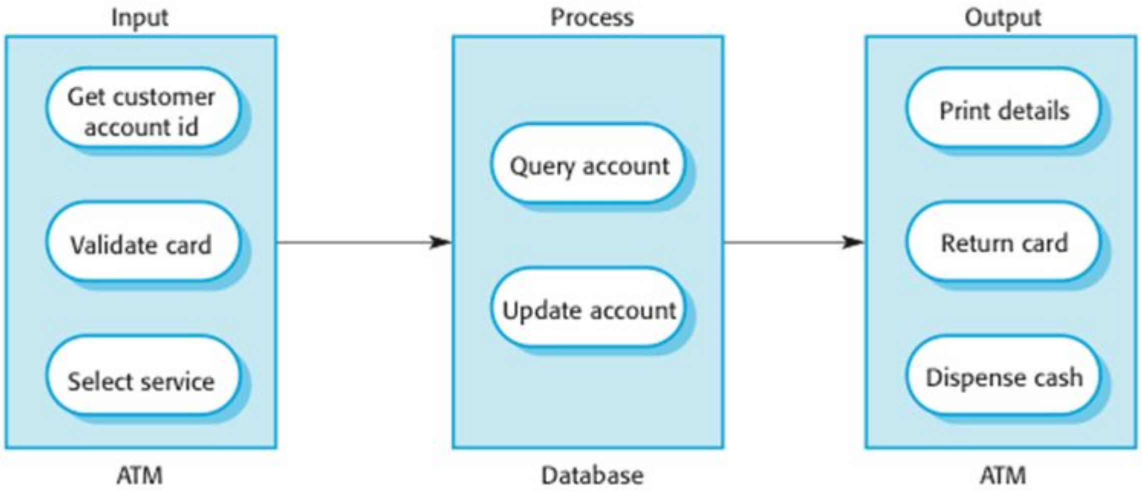
Application architectures encapsulate the principal characteristics of a class of systems

For example, in real-time systems, there might be generic architectural models of different system types, such as data collection systems or monitoring systems

Although instances of these systems differ in detail, the common architectural structure can be reused when developing new systems of the same type

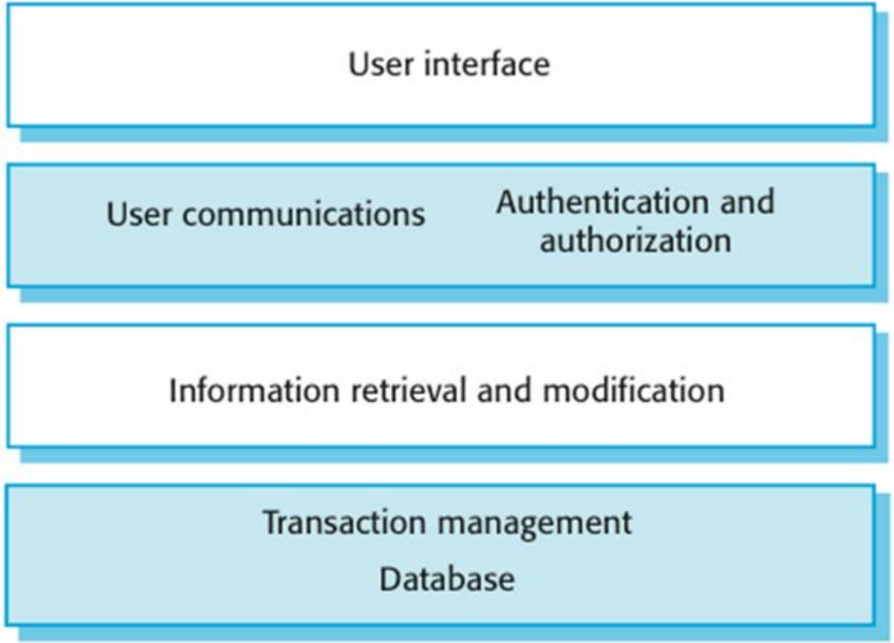
Application architectures

Transaction processing systems



Application architectures

Information systems



User interface

User communications

Authentication and authorization

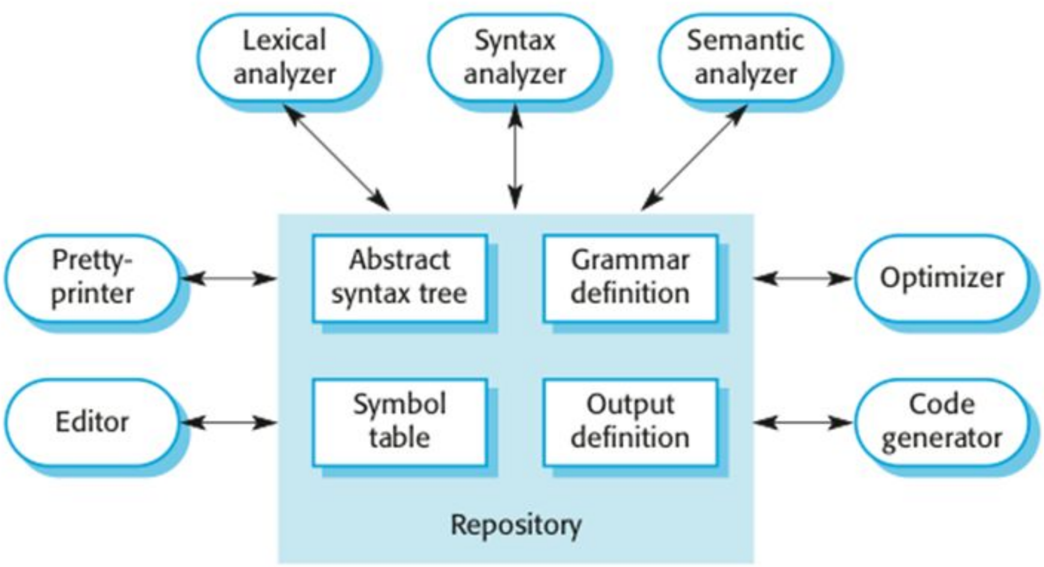
Information retrieval and modification

Transaction management

Database

Application architectures

Language processing systems



Application architectures

Getting the Basics - Software Architecture Introduction (part 1)

https://www.youtube.com/watch?v=8UlLgOf20Ho

Agenda: Lesson #05 - Software Engineering - Lecture

Q & A