

Database Application Development

DBS201

Lecture 06

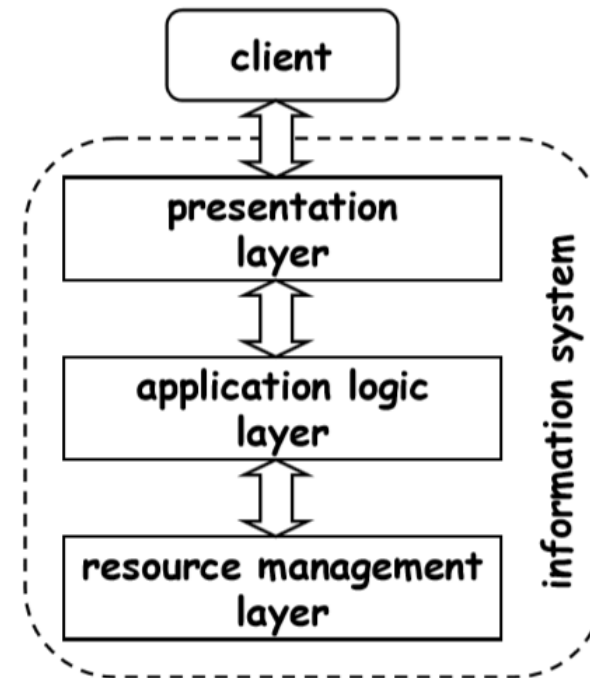
Application Architecture and Design

Web Services: Concepts, Architectures and Applications

By Gustavo Alonso, Fabio Casati, Harumi Kuno, Vijay Machiraju. Chapter 1

Application Layers

- Applications are typically broken into three layers:
 - Presentation and user interface
 - Business (application) logic
 - Data access (resource management)



Presentation Layer

- The presentation or the user interface provides the user interaction.
- Different form of the presentation layers could be
 - Web browsers
 - Mobile phone user interface

Business Layer

- The business or application logic layer provide the high level view and access of data.
- In this layer, business rules are enforced.

Data Access Layer

- The data access later is the interface between business layer and data resources. It deals with data storage and retrieval to support the business layer.

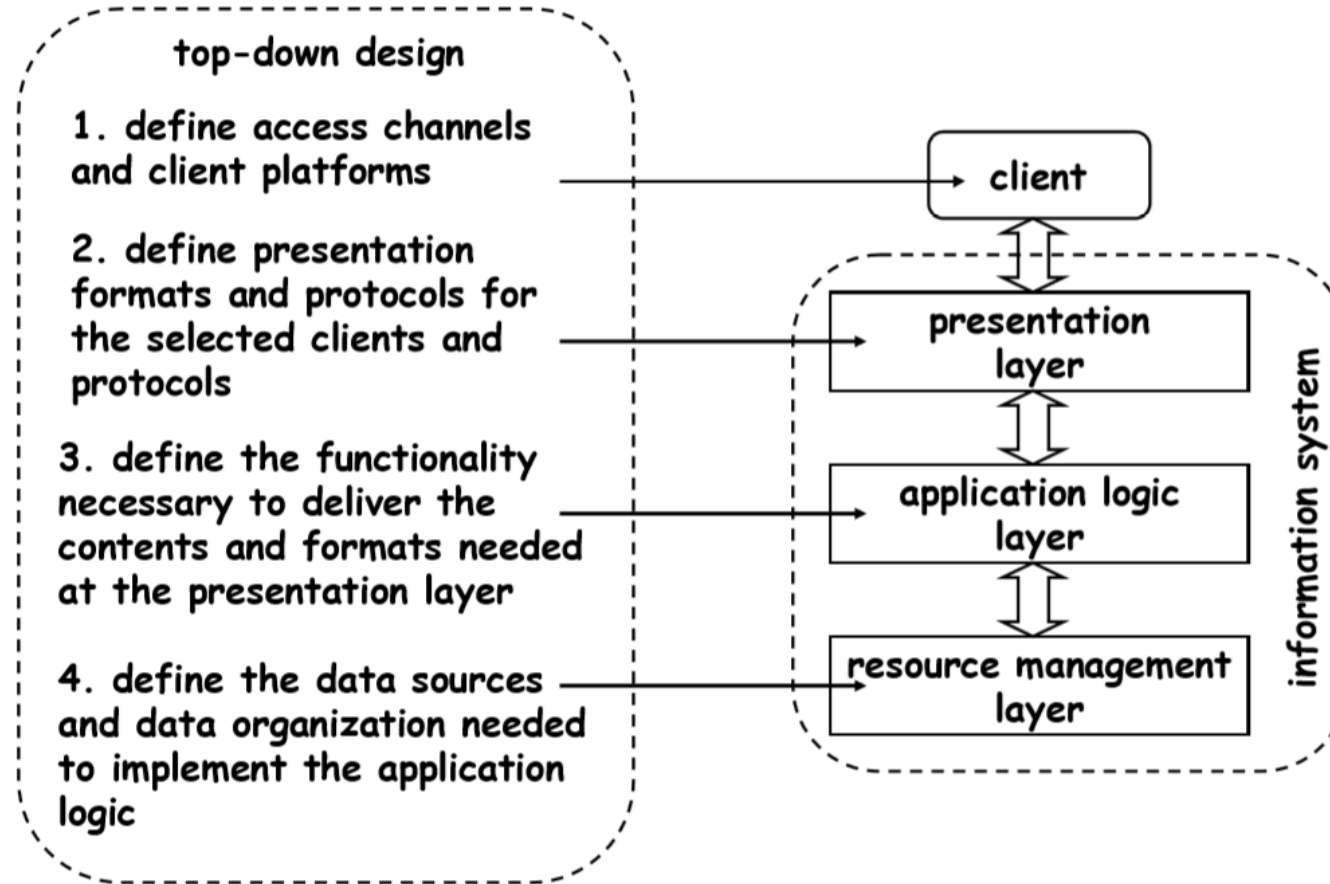
Application Design

- Top-down Design
- Bottom-up Design

Top-down Design

- The functionality of the system is defined from the client's point of view.
 - It determines how client interact with the system.
 - It focuses on high-level requirements.
- Advantages:
 - It focuses on final goals.
 - It addresses
 - Functional issues – the operations that are supported
 - Non-functional issues – the performance and availability
- Disadvantages
 - The system has to be entirely developed from scratch.

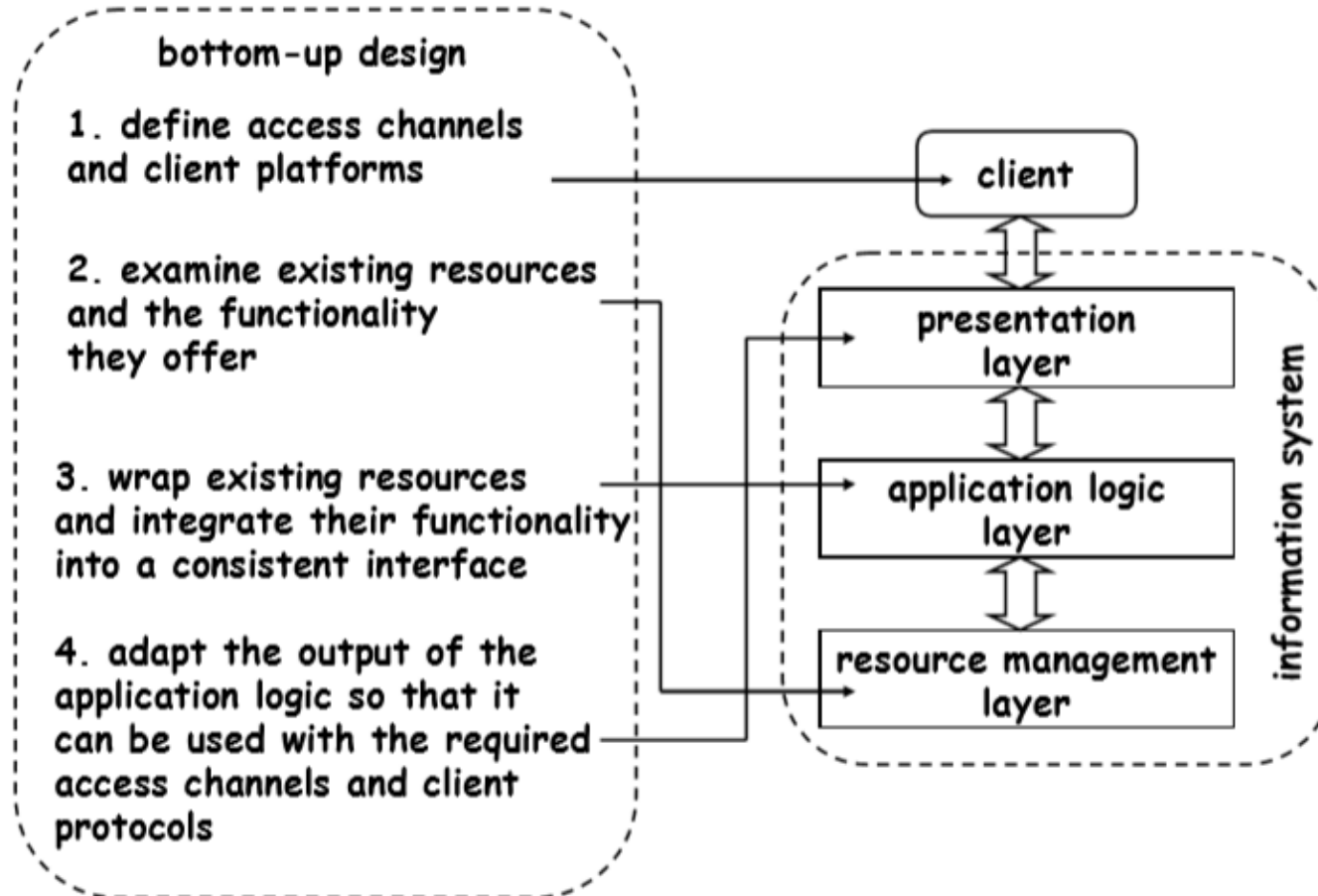
Top-down Design



Bottom-up Design

- The system components and parts are designed.
- The smaller parts and components of the system are then combined to achieve a higher goal.
- New systems can be created by integrating existing systems.
- The bottom-up approach includes:
 - Investigating existing applications and processes
 - Analyzing the problem domains
 - Redefining and reconstructing the problems
 - Determining the high-level objectives achieved by integrating these applications

Bottom-up Design



Application Architecture

- Two-tier architecture
- Three-tier architecture

2 Tiered Architecture

- The two-tier application is based on client server approach.
- There is a direct communication between client and server.
- Client request data from server. Further data processing can be done on the client side.
- The two-tier architecture has two main parts:
 - Database
 - Client application
- Advantages:
 - Since there is not any intermediate layer between client and server, two-tier application runs faster. Business and data managements are in one layer.
- Disadvantages:
 - It supports limited number of clients.
 - The server cannot respond multiple request same time

3 Tiered Architecture

- In this architecture, there is an intermediate layer between presentation (user interface) and data layers.
- Three separate layers:
 - Presentation
 - In this layer, the users send requests and receive data.
 - Business/application logic
 - In this layer, business rules are applied on
 - data calculation
 - data validation
 - data input
 - Data
 - This layer includes methods to
 - connect to the data resource (database)
 - manipulate data such as insertion, deletion, and modification

Database Application Development

A First Course in Database Systems_3rd Edition

[https://dev.mysql.com/doc/connector-cpp/8.0/en/Oracle document](https://dev.mysql.com/doc/connector-cpp/8.0/en/Oracle-document)

MySQL Document

Three Tiered Architecture

- It has three interactive parts:
 - Web Server
 - includes components that connect clients to the database systems.
 - Application Server
 - performs the business logic.
 - Database Server
 - This component runs the database and performs the queries requested from the application server.

Database Application

- How SQL fits into a programming environment?
- How a typical application can access a database to fetch or manipulate data in a database?
- How to embed SQL to programs with programming languages such as C++?

Database Access from Application

- How SQL commands can be executed in a host language?
 - **Embedded SQL**
 - **Cursors**
 - **Dynamic SQL**

Embedded SQL

- SQL statements and commands can be used in the host language. In this course, we use C++ as the host language.
- In an Embedded SQL approach, the host language can consist of either static or dynamic SQL or a combination of both.
- One approaches to embed SQL statements in a host language:
 - Call Level Interface (CLI):
 - In this approach, a program uses some libraries to access the database. All SQL statements are set of strings that are passed to the database using a CLI function call.

SQL Application

- Connect to DBMS
- Retrieve data from the database
- Manipulate data
- Modify data in the database