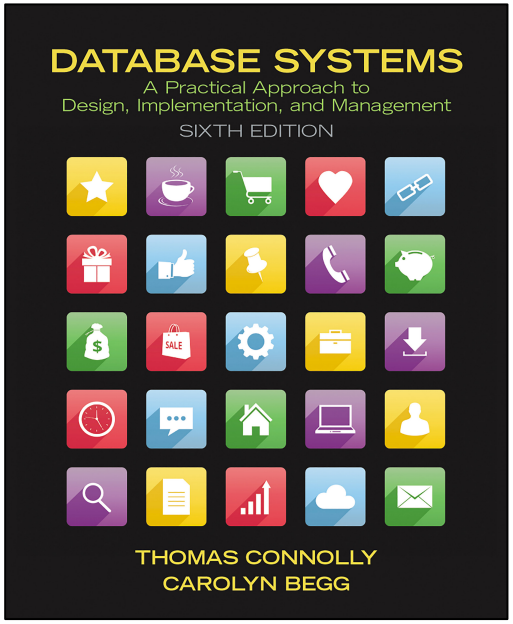
Database Models and Architectures

Topic 1

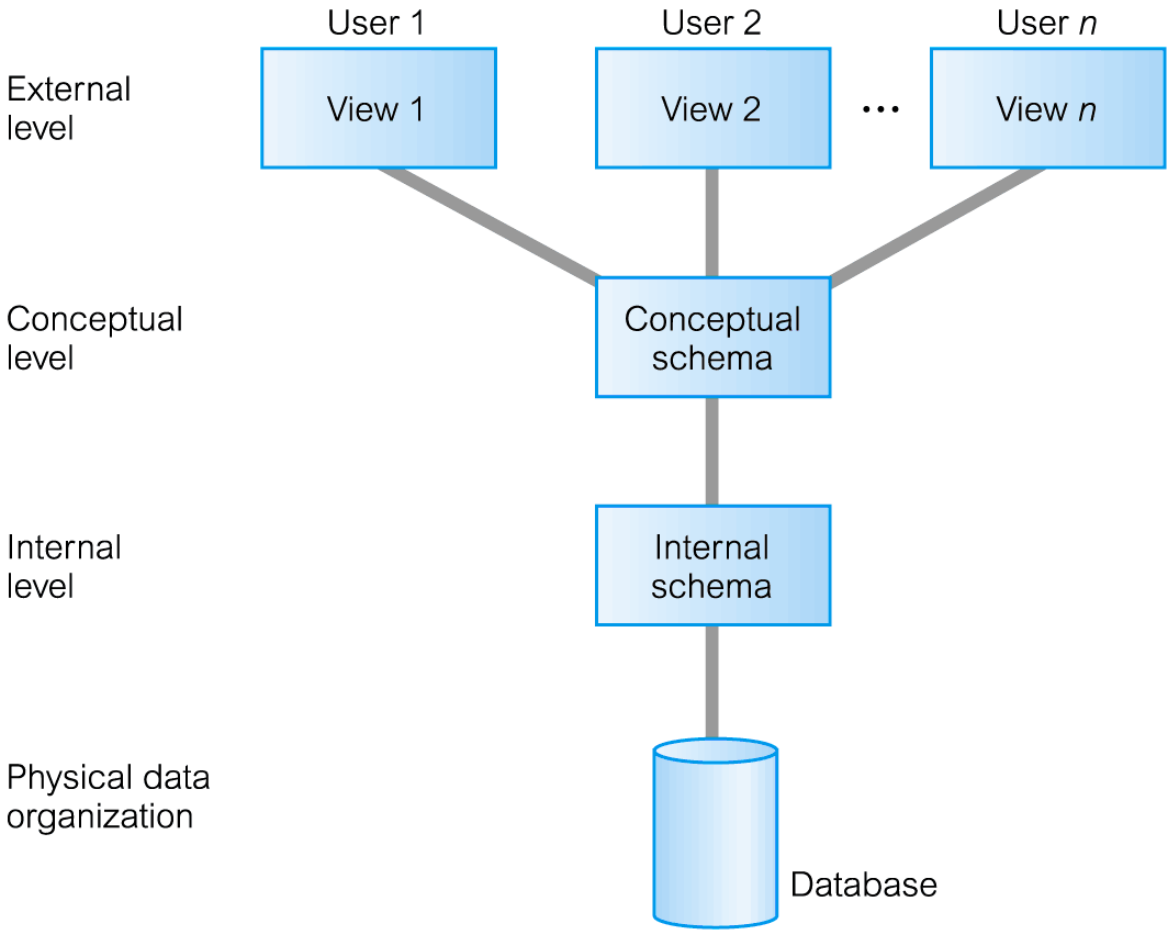
Lesson 2 – Software and hardware architectures

Chapter 2


DATABASE SYSTEMS
A Practical Approach to
Design, Implementation, and Management
SIXTH EDITION
SALE
THOMAS CONNOLLY
CAROLYN BEGG

The ANSI-SPARC model

Promoting data independence



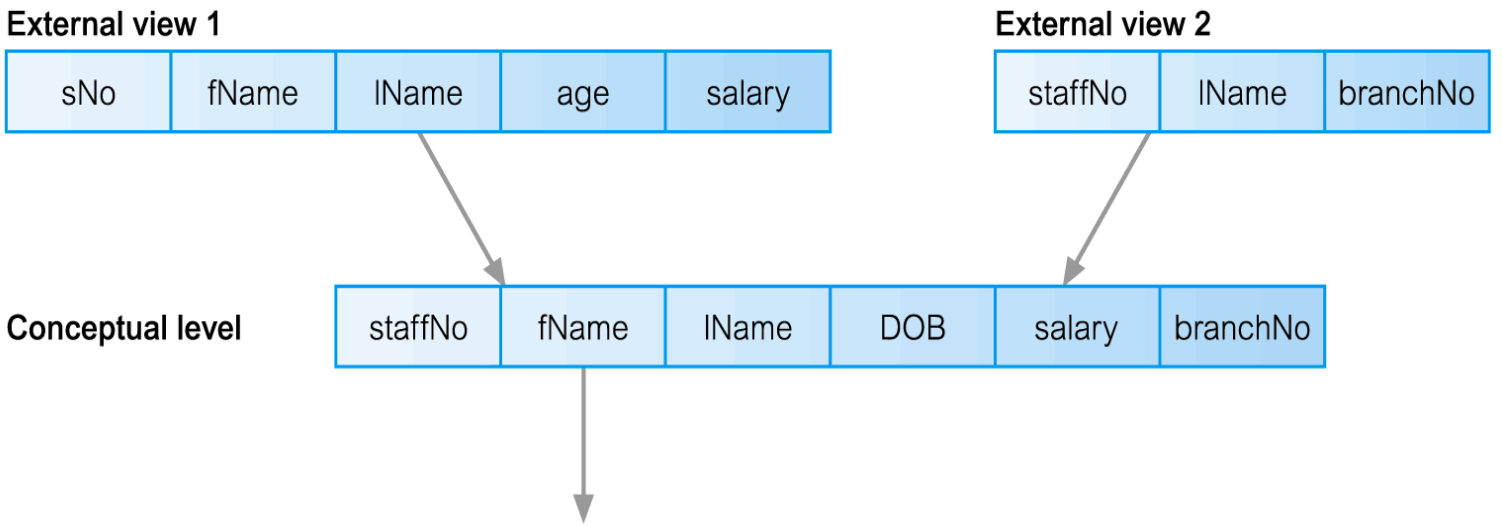
Objective for the ANSI-SPARC Architecture

Data Independence: immunity of a data application to changes in the data

- Logical data independence: immunity of external schemas to changes in conceptual schema.
- Physical data independence: immunity of conceptual schema to changes in the internal schema.

Example: Schemas for the three levels

Schema: a description of the database



Discussion

For relational database systems:

We know that the description of the structure of the data is stored in the system catalog. How does a user define the structure of their data?

How does a user manipulate their data?

ANSI-SPARC levels

External Level

- Users' view of the database.
- Describes that part of database that is relevant to a particular user.

Conceptual Level

- Community view of the database.
- Describes what data is stored in database and relationships among the data.

Internal Level

- Physical representation of the database on the computer.
- Describes how the data is stored in the database.

Conceptual Schema

- The conceptual schema is the core of a system supporting all user views.
- It should be complete and an accurate representation of an organization's data requirements.
- Conceptual modeling is the process of developing a model of information use that is independent of implementation details.

Data sublanguages

DDL or data **definition** language: used to specify the structure of the database schema. The data generated from DDL commands change the system catalog.

DML or the data **manipulation** language is used to read and update the data. The data generated from DML commands changes the data in a user table(s).

Application programmers can embed DDL or DML statements in a high-level programming language such as Python or Java

DDL and DML commands

How does a user define the structure of their data?

DDL Commands

- Create a new table (CREATE command)
- Update structure of table (ALTER command)
- Delete a table (DROP command)

How does a user manipulate their data?

DML Commands

- Create a new tuple (INSERT command)
- Read a tuple (SELECT command)
- Update a tuple (UPDATE command)
- Delete a tuple (DELETE command)

System catalog

- Repository of information (metadata) describing the data in the database. It is updated by DDL commands
- The system catalog is one of the fundamental components of DBMS.
- The system catalog typically stores:
  - names, types, and sizes of data items;
  - constraints on the data;
  - names of authorized users;
  - data items accessible by a user and the type of access granted to a user;
  - usage statistics.

Using a model to represent a data schema

A model is a **representation** of real-world objects and events, and their associations.

What is a data model?

A **data model** is a higher-level description of a data schema. It is an integrated collection of concepts for describing the data, relationships between data and constraints on the data

The purpose of a data model is to facilitate the understanding of the data.

A data model consists of:

1. A structural part
2. A manipulative part
3. A set of integrity constraints that ensures that the data is accurate and sound

What functionality should a DB have?

Let's identify the functionality expected from a fully functioning database system. Chris Date did this exercise back in 1982.

Database functionality

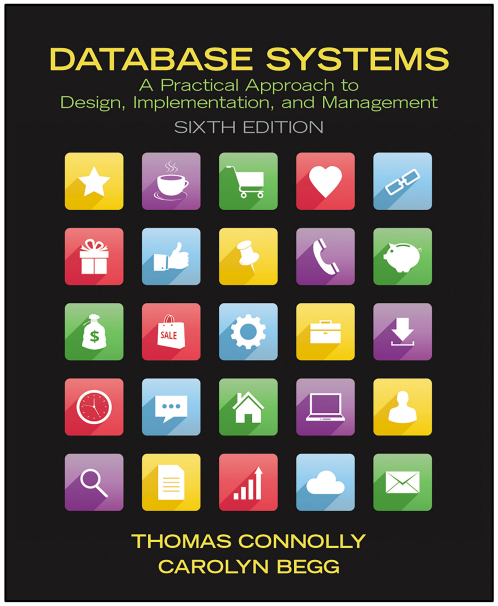
- Data Storage, Retrieval, and Update.
- A User-Accessible Catalog.
- Transaction Support.
- Concurrency Control Services.
- Recovery Services.
- Authorization Services.
- Support for Data Communication.
- Integrity Services.
- Services to Promote Data Independence.
- Utility Services.

Summary

In this module you learned:

- The ANSI-SPARC architecture and the types of data independence it provides
- The use of DDL commands to update the system catalog
- The use of DML commands to manipulate user data
- The expected functionality of a DBMS

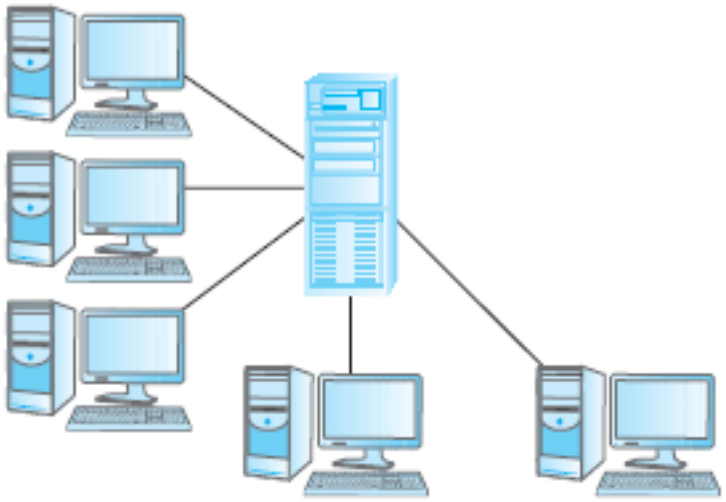
Chapter 3 Connolly & Begg

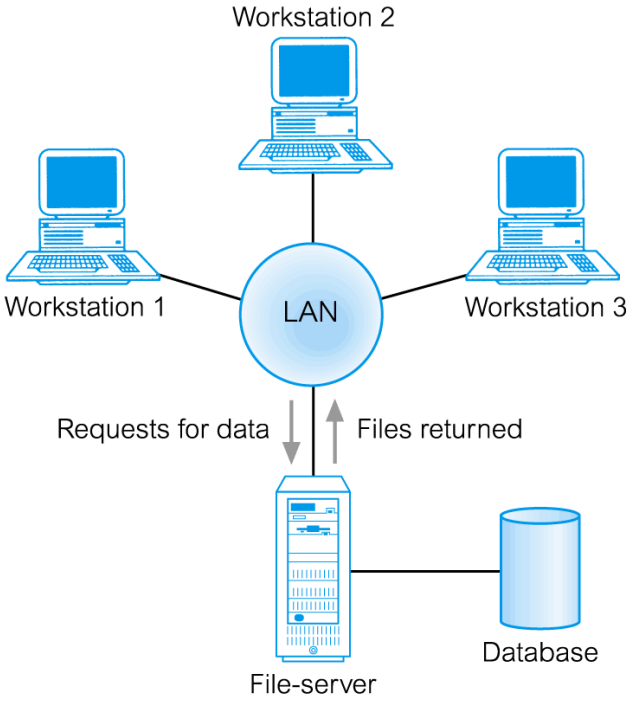


Hardware housing databases

Old Architectures

Teleprocessing





File Server Architecture

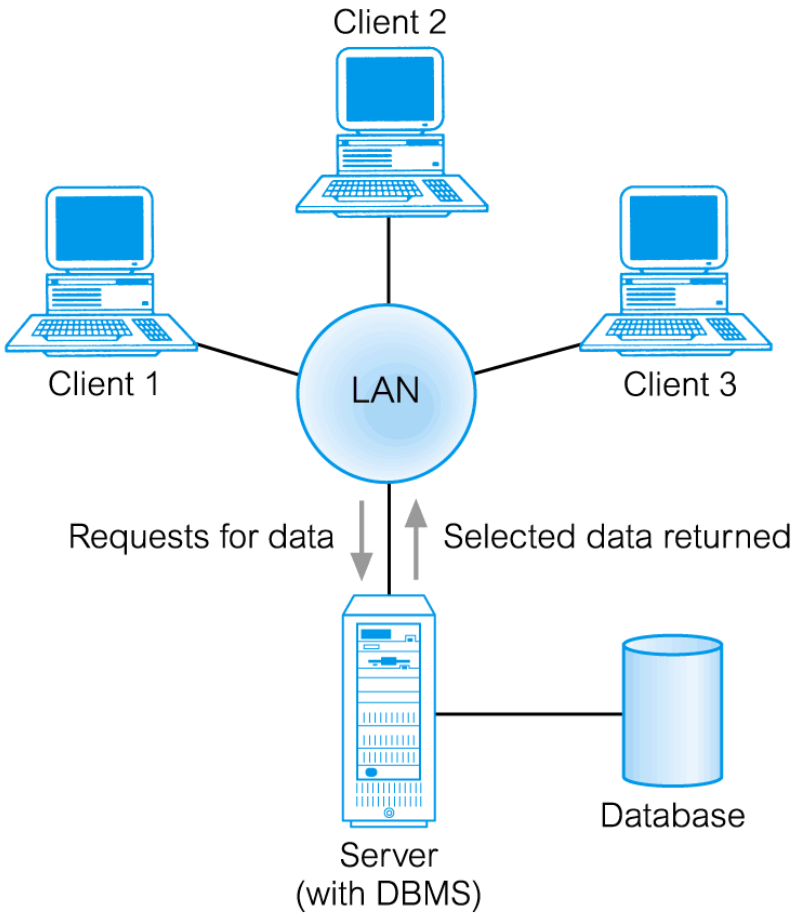
What are the modern hardware systems?

Think about the web, and the way computers communicate with each other. Are there any general protocols used? Any specific protocols used for databases?

Spend a few minutes discussing this with your classmates, feel free to use any available resources

Client-Server Architecture

Portions of the database application deployed on potentially different hardware with a well-defined method for communication



Discussion: client server architecture

In the client server architecture, the database application is spread across at least two different processes, residing on potentially two different platforms.

Identify the functionality that the client would need to provide for a database application?

Identify the functionality that the database server would need to provide for a database application?

Server and Client Functionality

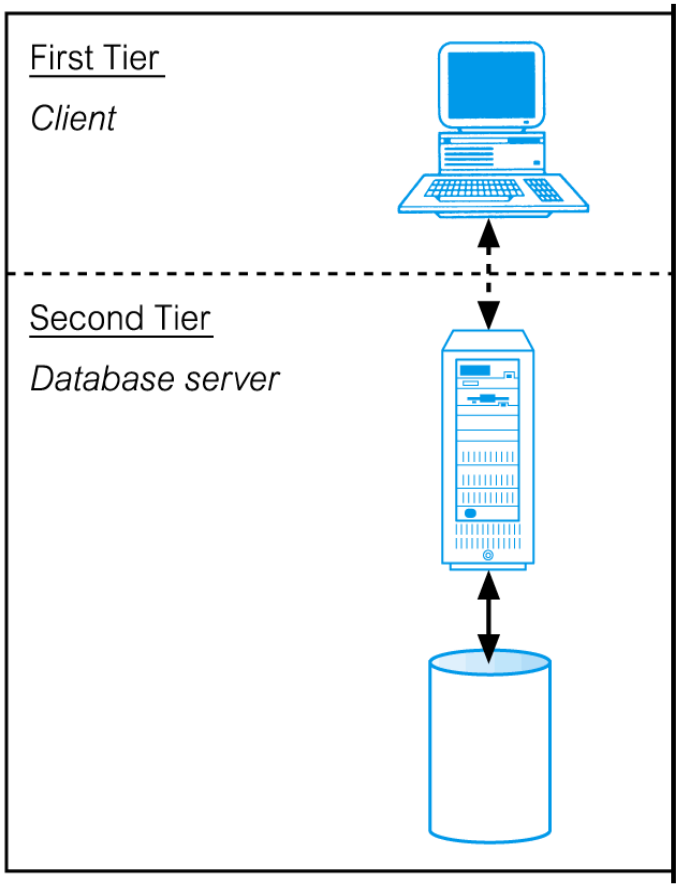
Client	Server
Manages the user interface	Accepts and processes database requests from clients
Accepts and checks syntax of user input	Checks authorization
Processes application logic	Ensures integrity constraints not violated
Generates database requests and transmits to server	Performs query/update processing and transmits response to client
Passes response back to user	Maintains system catalog Provides concurrent database access Provides recovery control

Machines for the DB application

Are we limited to two machines?

Is there any benefit to having more machines as running part of the database application?

What type of work would they do?

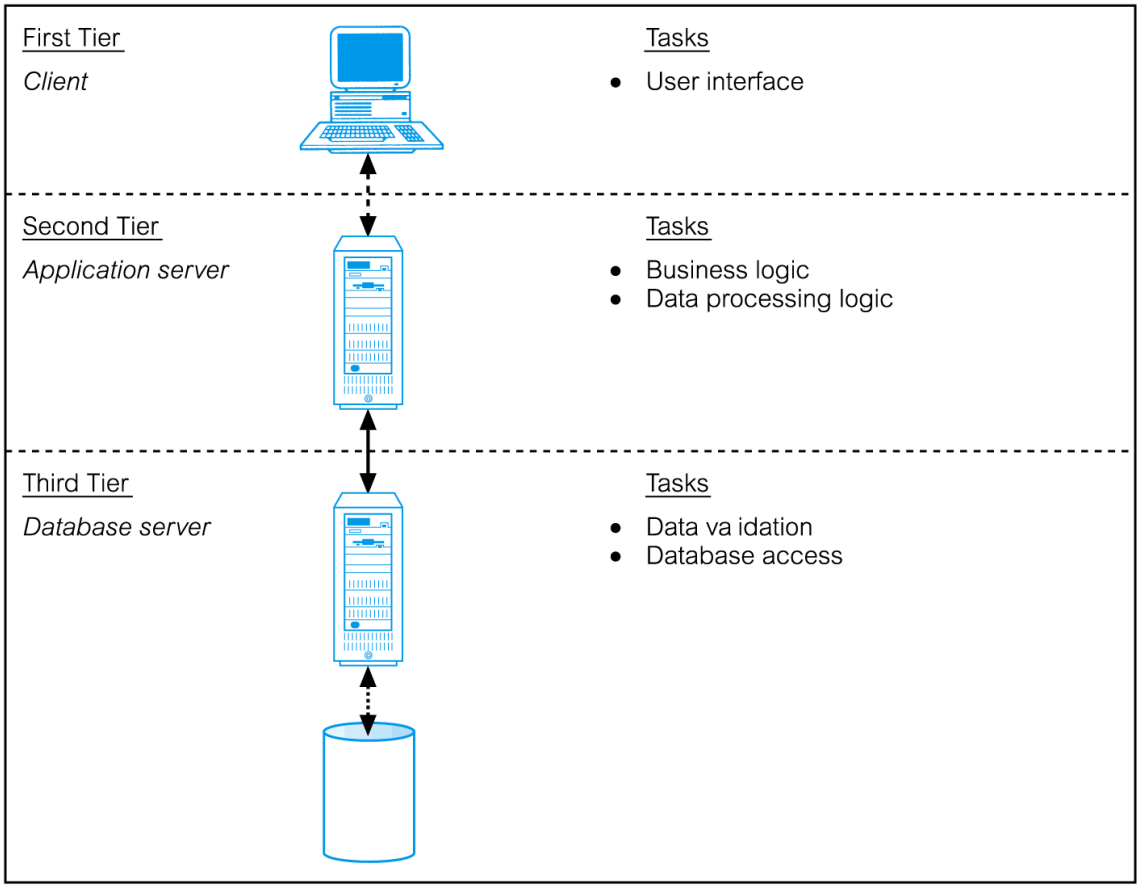


Add more hardware to lighten a load

- Client side presented two problems preventing true scalability:
  - ‘Fat’ client, requiring considerable resources on client’s computer to run effectively.
  - Significant client administration overhead.

- By 1995, three layers proposed, each potentially running on a different platform.

Additional tiers for a DB application



3-Tier Database applications

Benefits?

3-Tier Database applications

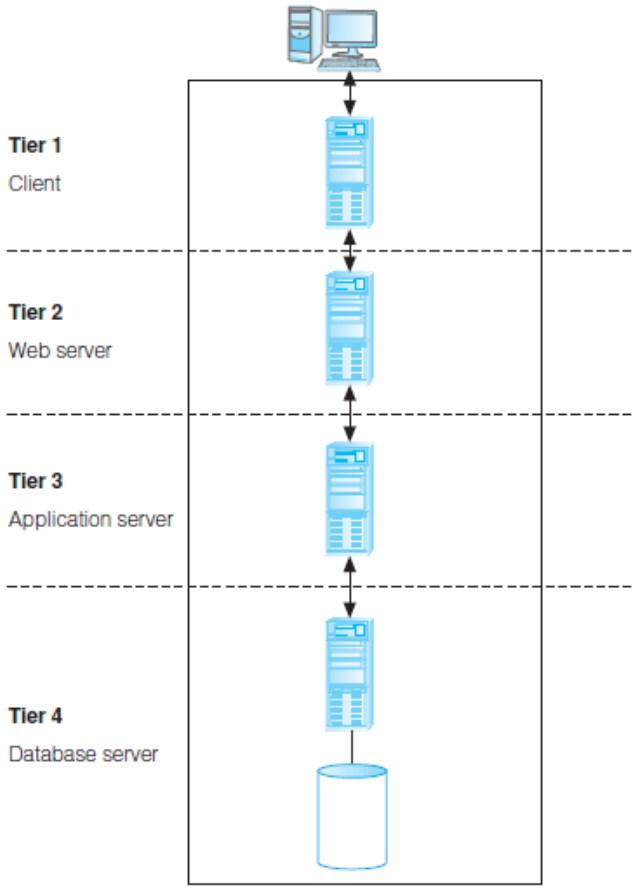
Benefits:

- 'Thin' client, requires less expensive hardware.
- Application maintenance centralized.
- Easier to modify or replace one tier without affecting others.
- Separates business logic from database functions, making it easier to implement load balancing.
- Maps quite naturally to Web environment.

N-tier architectures

Additional tiers add scalability and flexibility

Application servers host API to expose business logic and business processes for use by other applications

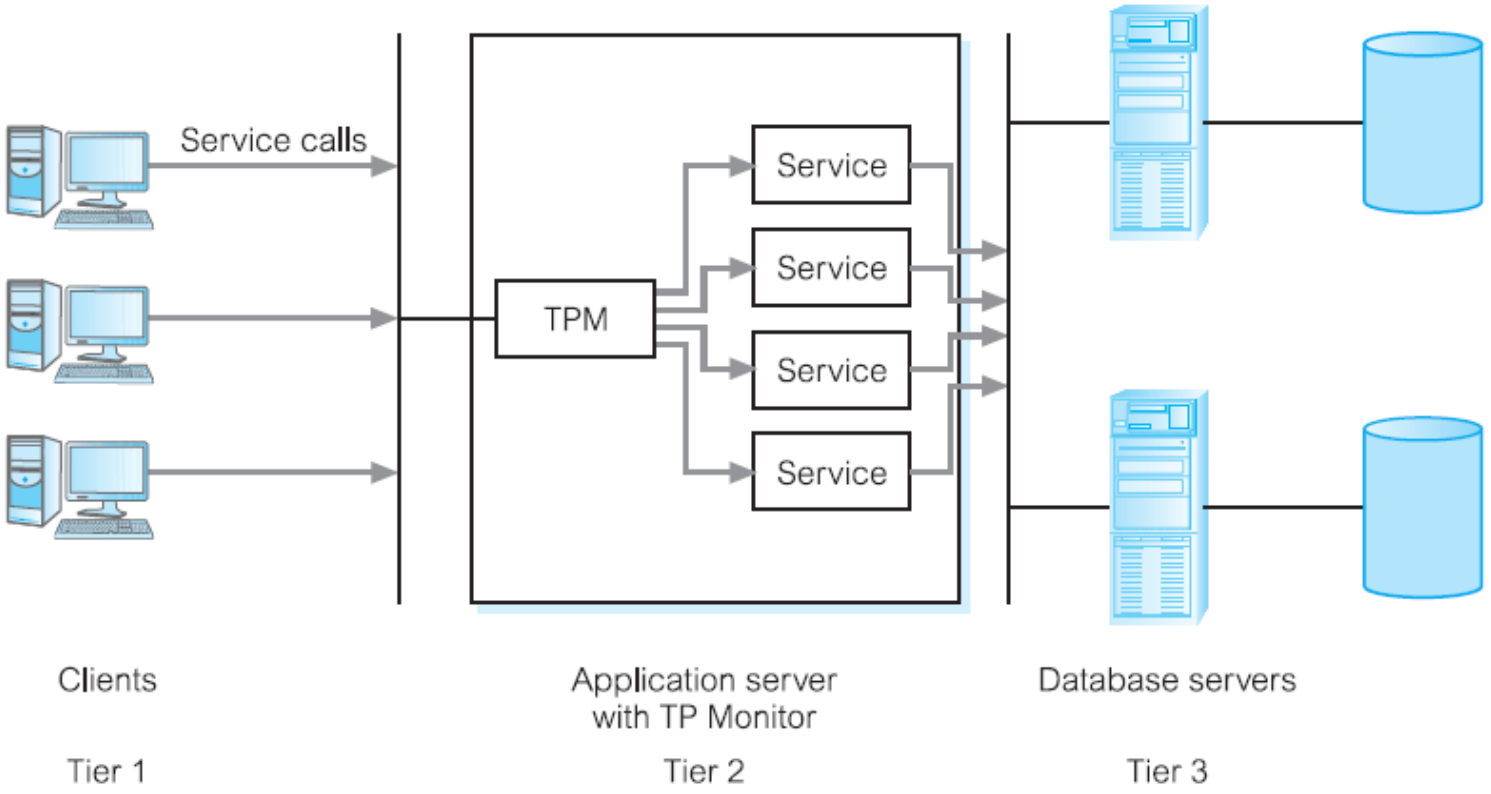


How do all these levels communicate?

Middleware is a generic term used to describe software that mediates with other software and allows for communication between disparate applications in a heterogeneous system.

The need for middleware arises when distributed systems become too complex to manage efficiently without a common interface.

Transaction Processing Monitor



Cloud computing

"A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

Cloud computing

What do these picture have in common with cloud computing?

Discussion: Cloud characteristics

Discuss some ways you interact with the cloud.

List some characteristics of cloud computing.

Cloud computing characteristics(1)

On-demand self-service

Consumers can obtain, configure and deploy cloud services without help from the cloud provider.

Broad network access

Accessible from anywhere, from any standardized platform (e.g. desktop computers, laptops, mobile devices).

Cloud computing characteristics (2)

Resource pooling

Provider's computing resources are pooled to serve multiple consumers, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. Examples of resources include storage, processing, memory, and network bandwidth.

Cloud computing characteristics (3)

Rapid elasticity

Provider's capacity caters for customer's spikes in demand and reduces risk of outages and service interruptions. Capacity can be automated to scale rapidly based on demand.

Measured service

Provider uses a metering capability to measure usage of service (e.g. storage, processing, bandwidth, and active user accounts).

Cloud service models

Infrastructure as a Service (IaaS)

Provider's offer servers, storage, network and operating systems – typically a platform virtualization environment – to consumers as an on-demand service, in a single bundle and billed according to usage.

Cloud service models

Platform as a Service (PaaS)

Allows creation of web applications without buying/maintaining the software and underlying infrastructure. Provider manages the infrastructure including network, servers, OS and storage, while customer controls deployment of applications and possibly configuration.

Cloud service models

Software as a Service (SaaS):

Software and data hosted on cloud. Accessed using thin client interface (e.g. web browser). Consumer may be offered limited user specific application configuration settings.

Data and database in the cloud

- As a type of Software as a Service (SaaS), cloud-based database solutions fall into two basic categories:
  - Data as a Service (DaaS) and
  - Database as a Service (DBaaS).
- Key difference between the two options is mainly how the data is managed.

Database as a service

- DBaaS
  - Offers full database functionality to application developers.
  - Provides a management layer that provides continuous monitoring and configuring of the database to optimized scaling, high availability, multi-tenancy (that is, serving multiple client organizations), and effective resource allocation in the cloud, thereby sparing the developer from ongoing database administration tasks.

Data as a service

DaaS

Services enables data definition in the cloud and subsequently querying.

Does not implement typical DBMS interfaces (e.g. SQL) but instead data is accessed via common APIs.

Enables organization with valuable data to offer data access to others. Examples Urban Mapping (geography data service), Xignite (financial data service) and Hoovers (business data service.)

Discussion: Cloud computing

Identify the benefits and drawbacks of cloud computing.

Benefits of cloud computing

Cost-Reduction: Avoid up-front capital expenditure.

Scalability/Agility: Organisations set up resources on an as-needs basis.

Improved Security: Providers can devote expertise & resources to security; not affordable by customer.

Improved Reliability: Providers can devote expertise & resources on reliability of systems; not affordable by customer.

Access to new technologies: Through use of provider's systems, customers may access latest technology.

Benefits of cloud computing

- **Faster development:** Provider's platforms can provide many of the core services to accelerate development cycle.
- **Large scale prototyping/load testing:** Providers have the resources to enable this at a lower cost point.
- **More flexible working practices:** Staff can access files using mobile devices so can work anywhere.
- **Increased competitiveness:** Allows organizations to focus on their core competencies rather than their IT infrastructures.

Drawbacks to cloud computing

- **Network Dependency:** Power outages, bandwidth issues and service interruptions.
- **Provider System Dependency:** Customer's dependency on availability and reliability of cloud provider's systems.
- **Cloud Provider Dependency:** Provider could become insolvent or acquired by competitor, resulting in the service suddenly terminating.
- **Lack of control:** Customers unable to deploy technical or organisational measures to safeguard the data. May result in reduced availability, integrity, confidentiality, intervenability and isolation.
- **Lack of information on processing transparency**

Summary

In this module you learned:

- Hardware architectures used to deploy databases
- Client-server model
- Characteristics and benefits of cloud computing
- Cloud-computing service models