Topic 2:

Database Management System Architectures

What is an 'Architecture' of a DBMS?

Definition: Schema

Definition: Architecture

The ANSI/SPARC Architecture (1 / 4): Background

a.k.a. Three-Level Schema

- A product of the Standards Planning and Requirements Committee (SPARC) of the American National Standards Institute (ANSI)

- Never formally adopted as an ANSI or International Standards Organization (ISO) standard, but still very influential

- Created to standardize terms and concepts surrounding DBs and DBMSes

- Goals:

    – Allow for multiple views of the data to satisfy a range of users
    – Allow for a physical (disk-level) description of the database
    – Provide an abstraction layer to separate the two

The ANSI/SPARC Architecture (2 / 4): The Diagram

The ANSI/SPARC Architecture (3 / 4): The Levels

- External Level
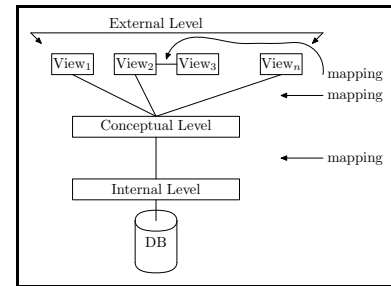
- Conceptual Level

- Internal Level

The ANSI/SPARC Architecture (4 / 4): The Mappings

The interfaces between the levels are known as *mappings*.

- External – External Mapping

- External – Conceptual Mapping

- Conceptual – Internal Mapping

Client - Server Architectures (1 / 3): Background

- Originally: DBMSes were built with a centralized architecture.

    – All components (OS, DBMS, compilers, etc.) on one
      computer

    – "All or nothing" with respect to failures

    – Often a performance bottleneck

- Decentralization became feasible when:

    – Computers became less expensive and more powerful

    – Broadband networking became commonplace

Client - Server Architectures (2 / 3): Two-Tier

User Apps	Network	DBMS	DB
Client		Server	
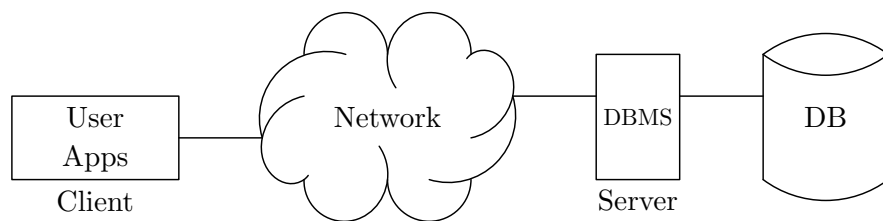
- One possible division of services:

Client - Server Architectures (3 / 3): Multi-Tier

Why add more tiers?

Example of a Four-Tier architecture:

Distributed DBMSes (1 / 2): Motivation

- A single DBMS server (with its single DB) is a single point of failure

- Solution: A DBMS can be operated by several servers.
    - Each server has all, some, or none of the DB stored locally (some replication exists for performance and reliability)

    - DDBMS sites communicate to handle nearly all tasks

    - Goal: Be completely transparent to the users

- Again, details are beyond the scope of this course

Distributed DBMSes (2 / 2): Diagram
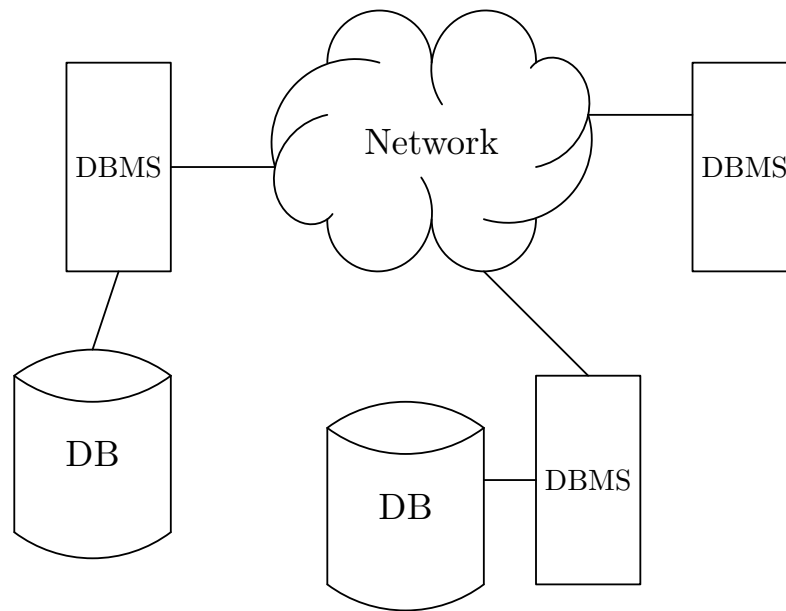
Extra Slides

The remaining slides in this topic are some that I no longer cover in class. I won't ask about them on an exam, but they could be referenced on a homework.

Service-Oriented Architectures (1 / 3): Motivation

- SOA is a software design technique:

  - Apps are built using pre-written service modules

    E.g., a data visualization module

  - Modules are located & accessed via a common interface

- Goal is to be flexible with the adoption of new business processes

- A web service is an interface used by service modules

  - That is, it can be a component of an SOA.

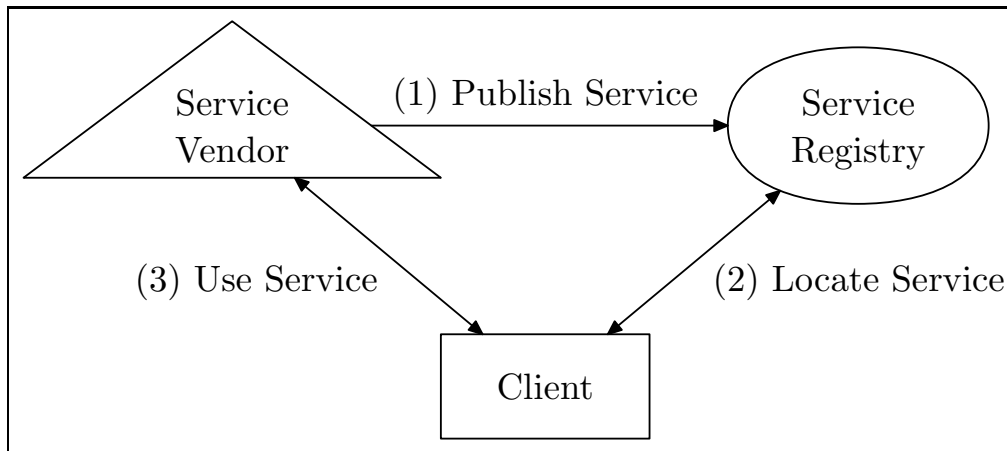- Further details are beyond the scope of this course

Service-Oriented Architectures (2 / 3): Before/After
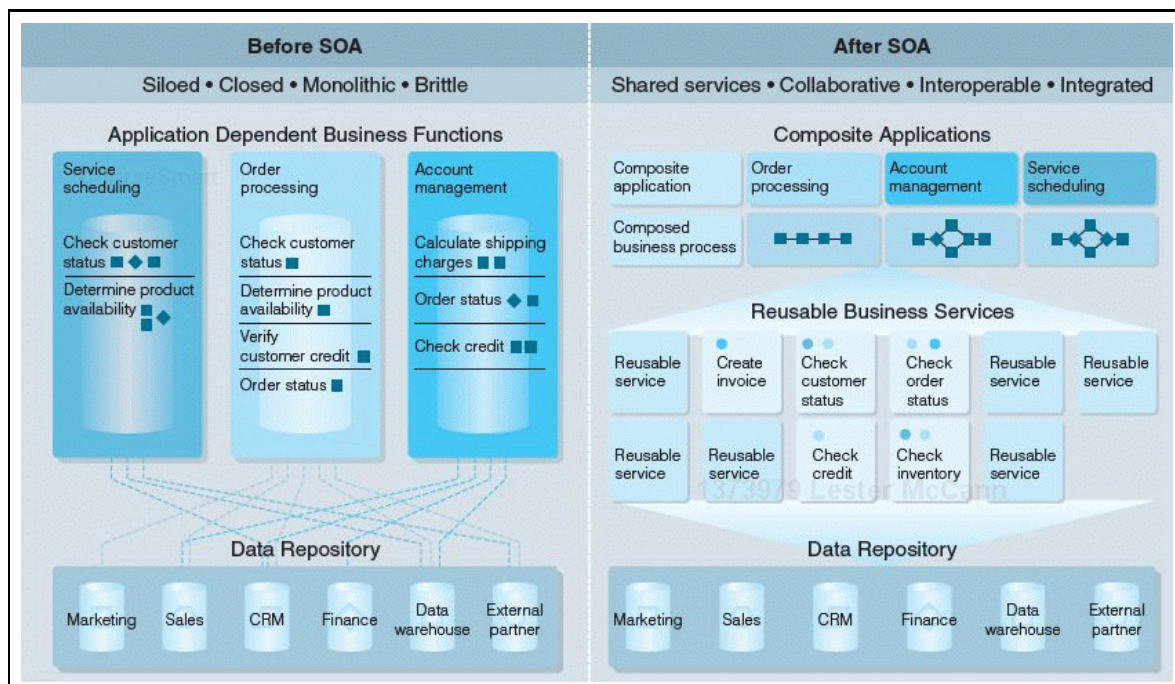
Credit: Connolly/Begg, "Database Systems," 6/e, p. 72.

Service-Oriented Architectures (3 / 3): Accessing

Advertising, Finding and Using a Service:

One example: Web browser plug-ins.

(This approach is typical for web services, too.)

Additional DB–Related Architectures

- Web Services
  - ○ Ex: Stock Quotations; Google Docs

  Two types:

  (a) Simple Object Access Protocol (SOAP)-based
      – Typically uses XML

  (b) RESTful (Representational State Transfer) – stateless
      – Ex: HTTP

- Data Warehouses
  - ○ Support decision–making

- Cloud Computing
  - ○ Provides dynamic resource provisioning (of DBMSes!)