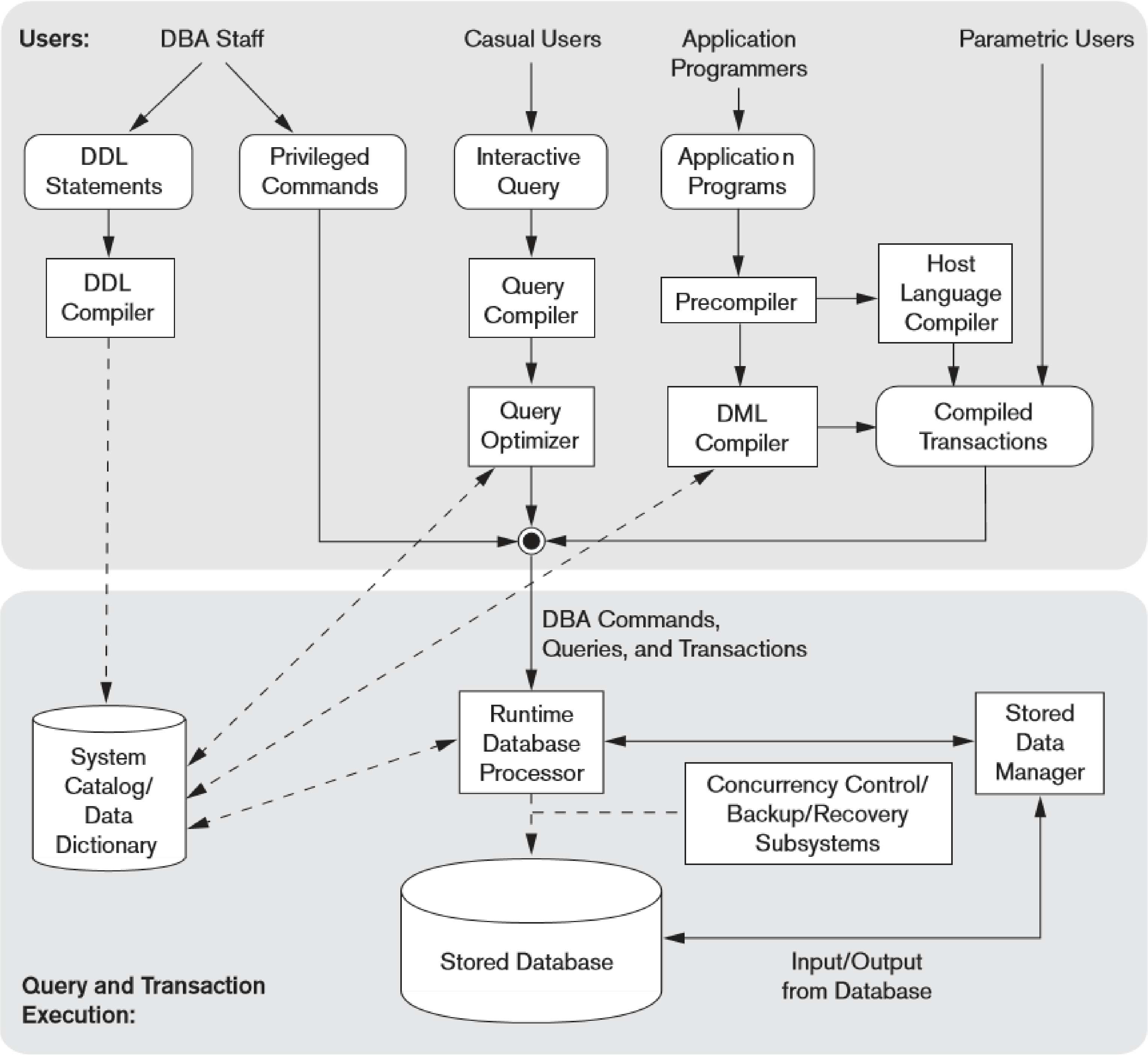
Database System Concepts and Architecture

Part 3

The Database System Environment

DBMS Component Modules


Users:
DBA Staff
Casual Users
Application Programmers
Parametric Users
DDL Statements
Privileged Commands
Interactive Query
Applicatio n Programs
DDL Compiler
Query Compiler
Precompiler
Host Language Compiler
Query Optimizer
DML Compiler
Compiled Transactions
DBA Commands, Queries, and Transactions
System Catalog/ Data Dictionary
Runtime Database Processor
Stored Data Manager
Concurrency Control/ Backup/Recovery Subsystems
Stored Database
Input/Output from Database
Query and Transaction Execution:

Database System Utilities

- Loading
- Backup
- Database storage reorganization
- Performance monitoring

Tools, Application Environments, and Communications Facilities

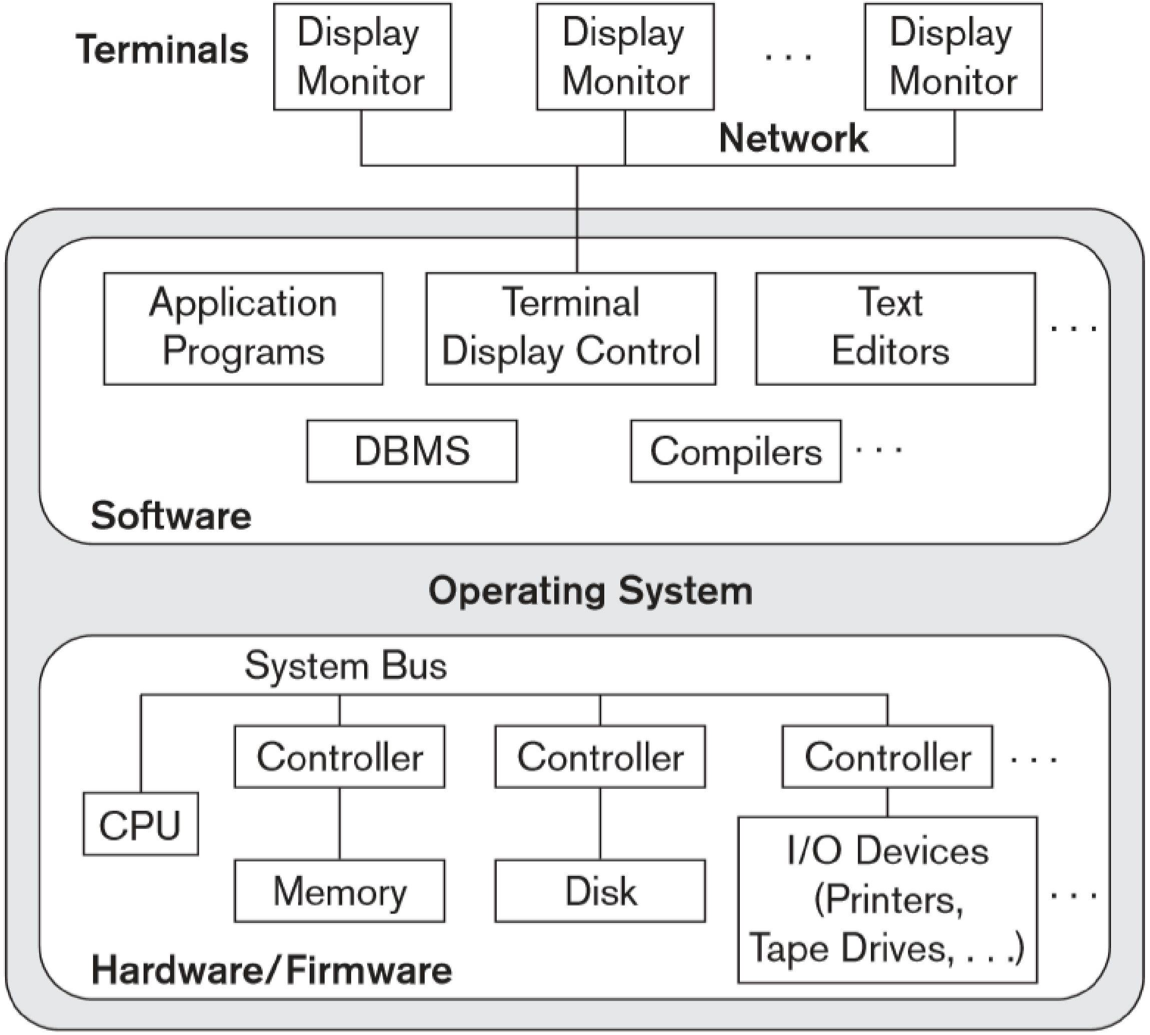
- CASE tools are used in the design phase of database systems.
- Another tool that can be quite useful in large organizations is an expanded **data dictionary** (or **data repository**) **system**.

- **Application development environments**, such as PowerBuilder (Sybase) or JBuilder (Borland),have been quite popular.
- These systems provide an environment for developing database applications and include facilities that help in many facets of database systems, including database design, GUI development, querying and updating, and application program development.

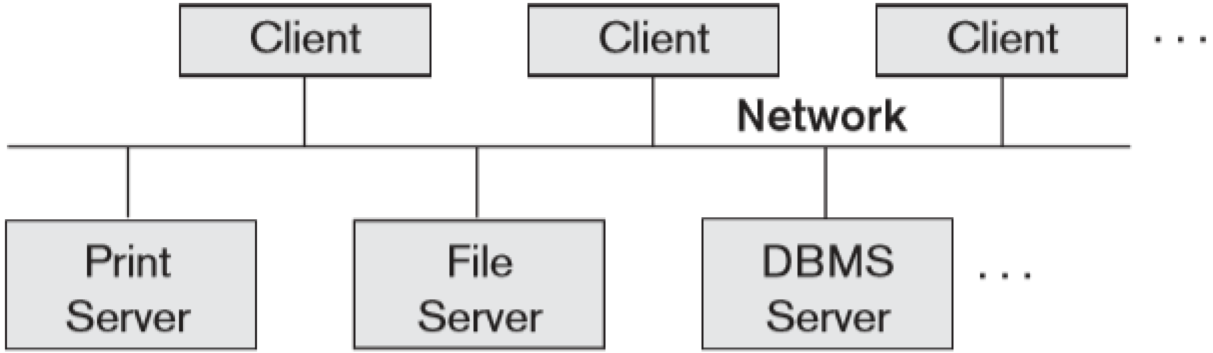
- The DBMS also needs to interface with **communications software**, whose function is to allow users at locations remote from the database system site to access the database through computer terminals, workstations, or personal computers.

Centralized and Client/Server Architectures for DBMSs

Centralized DBMSs Architecture



Basic Client/Server Architecture

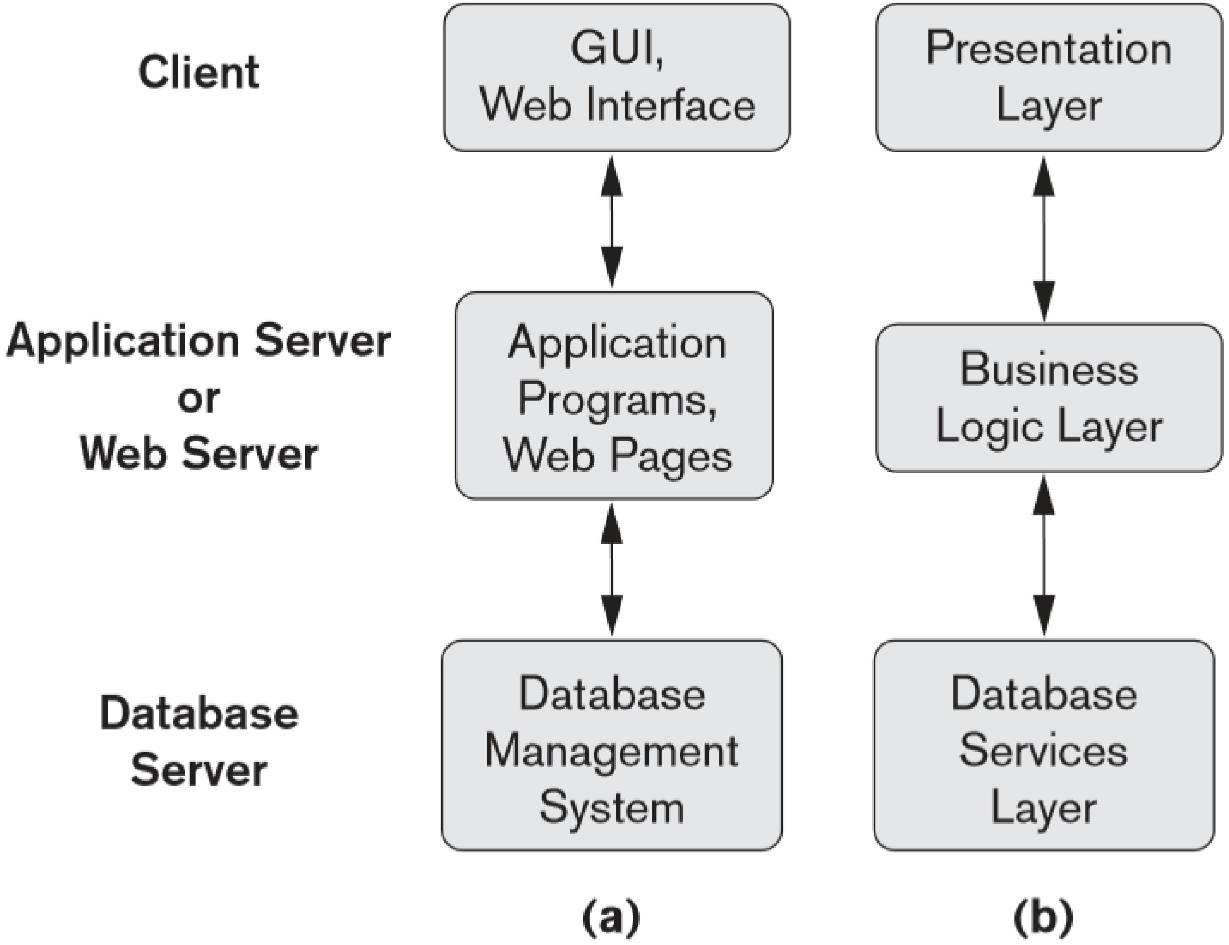


Two-Tier Client/Server Architectures for DBMSs

- Because SQL provided a standard language for RDBMSs, this created a logical dividing point between client and server.
- Hence, the query and transaction functionality related to SQL processing remained on the server side.
- In such an architecture, the server is often called a **query server** or **transaction server** because it provides these two functionalities.
- In an RDBMS, the server is also often called an **SQL server**.

- The user interface programs and application programs can run on the client side.
- When DBMS access is required, the program establishes a connection to the DBMS (which is on the server side); once the connection is created, the client program can communicate with the DBMS.
- A standard called **Open Database Connectivity** (**ODBC**) provides an application programming interface (**API**), which allows client-side programs to call the DBMS, as long as both client and server machines have the necessary software installed.
- A related standard for the Java programming language, called **JDBC**, has also been defined.
- This allows Java client programs to access one or more DBMSs through a standard interface.

Three-Tier and n-Tier Architectures for Web Applications



Classification of Database Management Systems

- Several criteria are normally used to classify DBMSs.
- The first is the **data model** on which the DBMS is based
  - Relational data model
    - Object-relational
  - Object data
  - Hierarchical data model
  - Network data model
  - XML model

- The second criterion used to classify DBMSs is the **number of users** supported by the system.
- **Single-user systems** support only one user at a time and are mostly used with PCs.
- **Multiuser systems**, which include the majority of DBMSs, support concurrent multiple users.

- The third criterion is the **number of sites** over which the database is distributed.
- A DBMS is **centralized** if the data is stored at a single computer site.
- A **distributed** DBMS (DDBMS) can have the actual database and DBMS software distributed over many sites, connected by a computer network.
- **Homogeneous** DDBMSs use the same DBMS software at all the sites, whereas **heterogeneous** DDBMSs can use different DBMS software at each site.
- It is also possible to develop **middleware software** to access several autonomous preexisting databases stored under heterogeneous DBMSs.
- This leads to a **federated** DBMS (or **multidatabase system**), in which the participating DBMSs are loosely coupled and have a degree of local autonomy.

- The fourth criterion is cost
  - Open source
  - Commercial

- We can also classify a DBMS on the basis of the **types of access path** options for storing files.
- One well-known family of DBMSs is based on inverted file structures.

- Finally, a DBMS can be **general purpose** or **special purpose**.