# Web Engineering

## Lecture 4
## N-Tier Architecture

Dr. Zulfiqar Ahmad
Department of CS & IT
Hazara University Mansehra
zulfiqarahmad@hu.edu.pk

## N-Tier Architecture

N-Tier architecture

- is an industry-proved software architecture model,
- suitable to support enterprise-level client/server applications by resolving issues like scalability, security, fault tolerance and etc.

## Significance of “Tiers”

**N-tier architectures have three components**

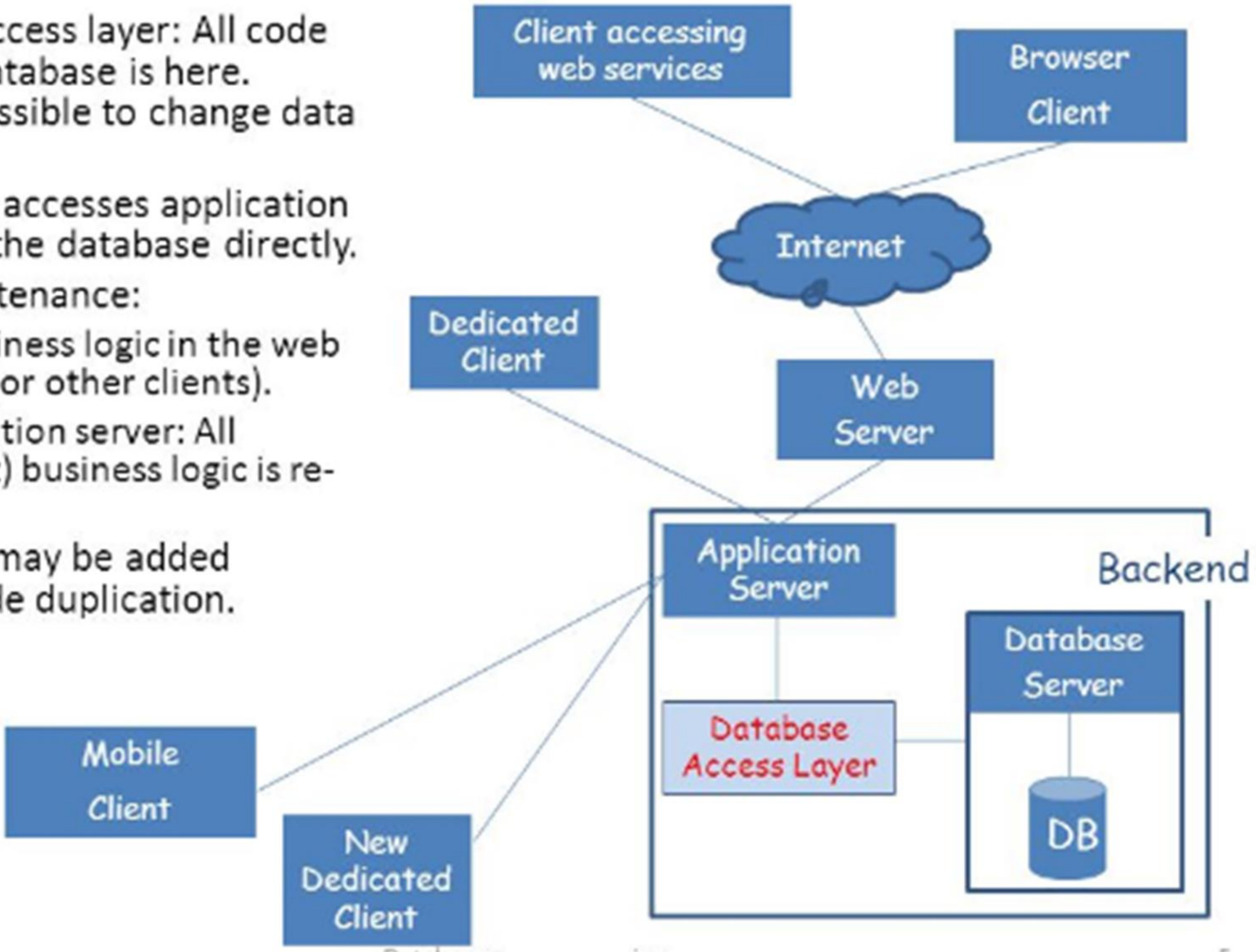
- Presentation
- Business/Logic
- Data

**N-tier architectures try to separate the components into different tiers/layers**

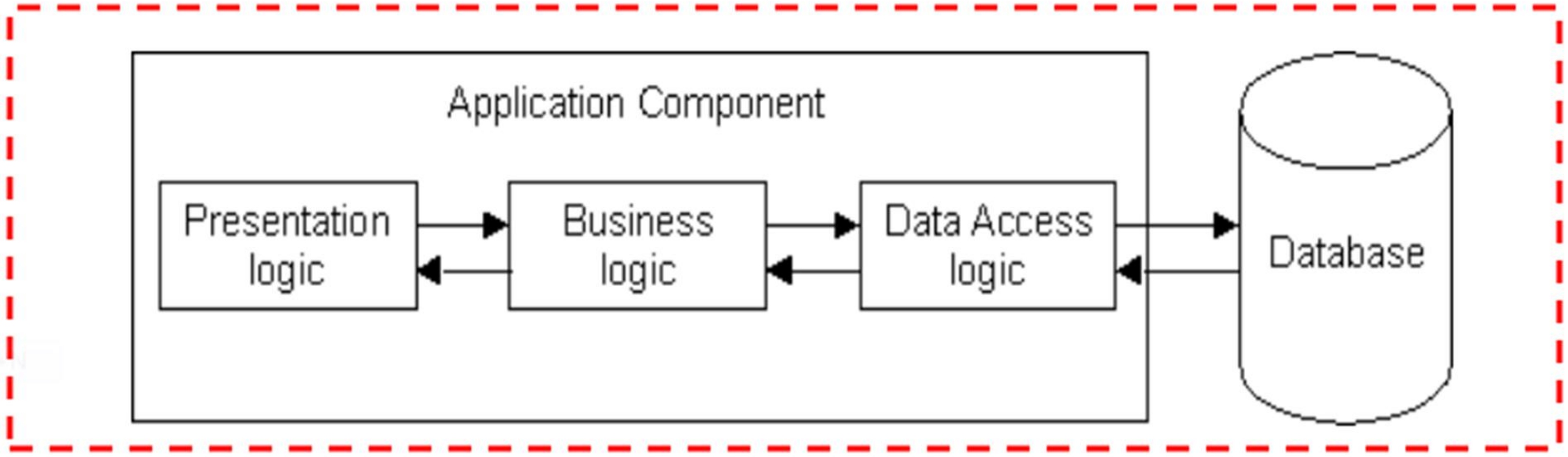
- Tier: physical separation
- Layer: logical separation

# N-tier (multi-tier) Architecture

- Database access layer: All code to access database is here. Makes it possible to change data store.
- Web server accesses application layer – not the database directly.
- Easier maintenance:
  - No business logic in the web server (or other clients).
  - Application server: All (almost) business logic is re-used.
- New client may be added without code duplication.



## 1-Tier Architecture



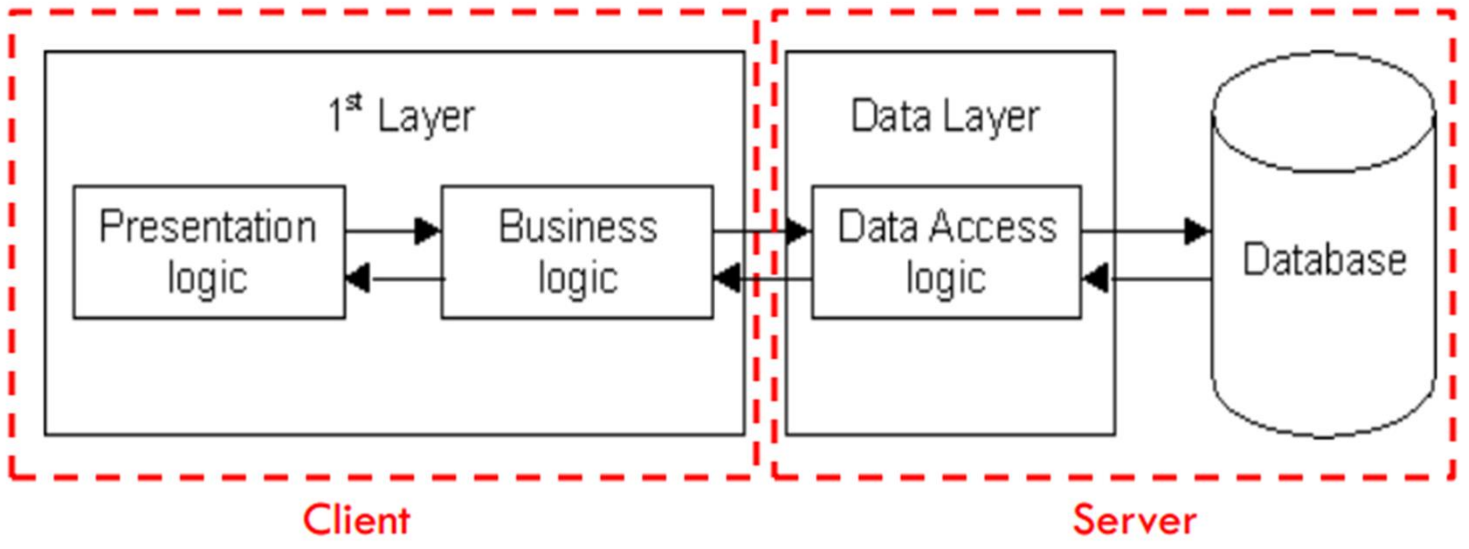
**All 3 layers are on the same machine**

- All code and processing kept on a single machine

**Presentation, Logic, Data layers are tightly connected**

- Scalability: Single processor means hard to increase volume of processing
- Portability: Moving to a new machine may mean rewriting everything
- Maintenance: Changing one layer requires changing other layers

## 2-Tier Architecture



**Database runs on Server**

- Separated from client
- Easy to switch to a different database

**Presentation and logic layers still tightly connected (coupled)**

- Heavy load on server
- Potential congestion on network
- Presentation still tied to business logic

## 7.0 Client/Server 2-Tier Architecture

Two-tier client/server architectures have 2 essential components

1. A Client PC and
2. A Database Server

**2-Tier Considerations:**

- Client program accesses database directly
    - Requires a code change to port to a different database
    - Potential bottleneck for data requests
    - High volume of traffic due to data shipping
- Client program executes application logic
    - Limited by processing capability of client workstation (memory, CPU)
    - Requires application code to be distributed to each client workstation

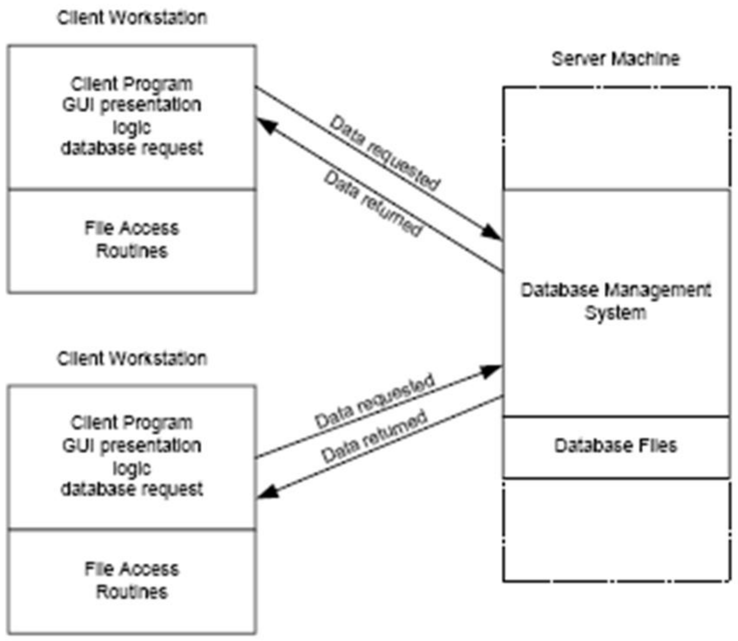
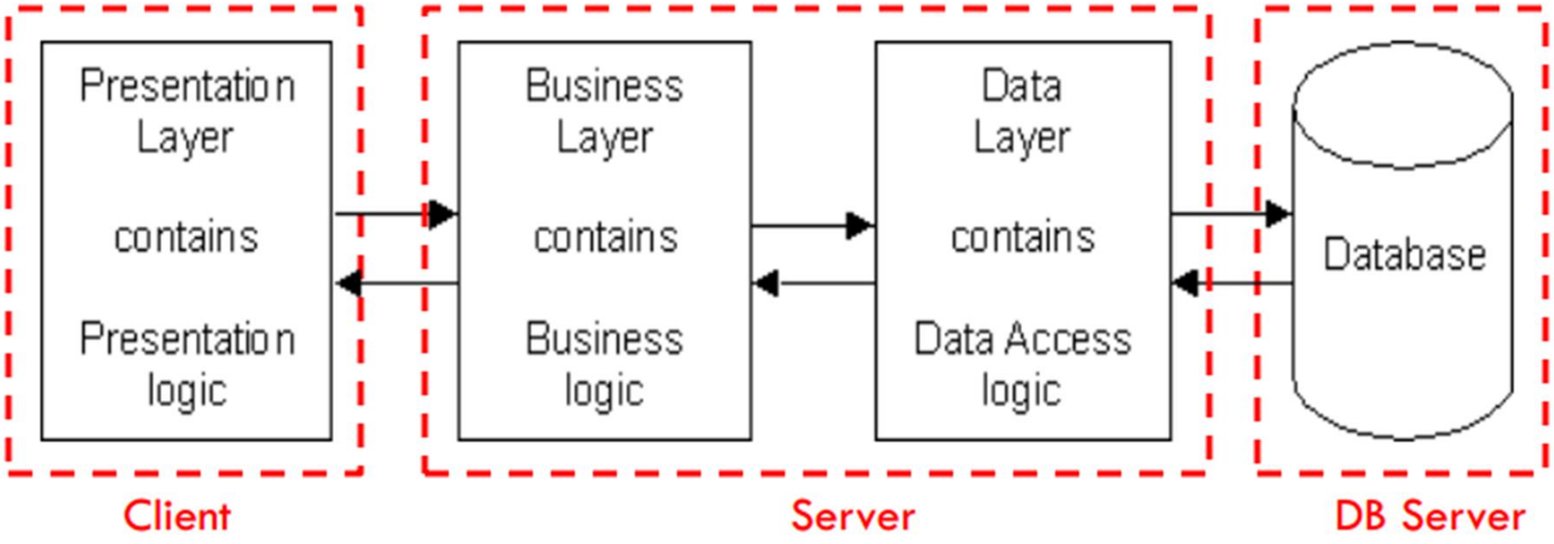


**Figure 7.1 Client/Server 2-Tier Architecture**

## Two – Tier Pros and Cons

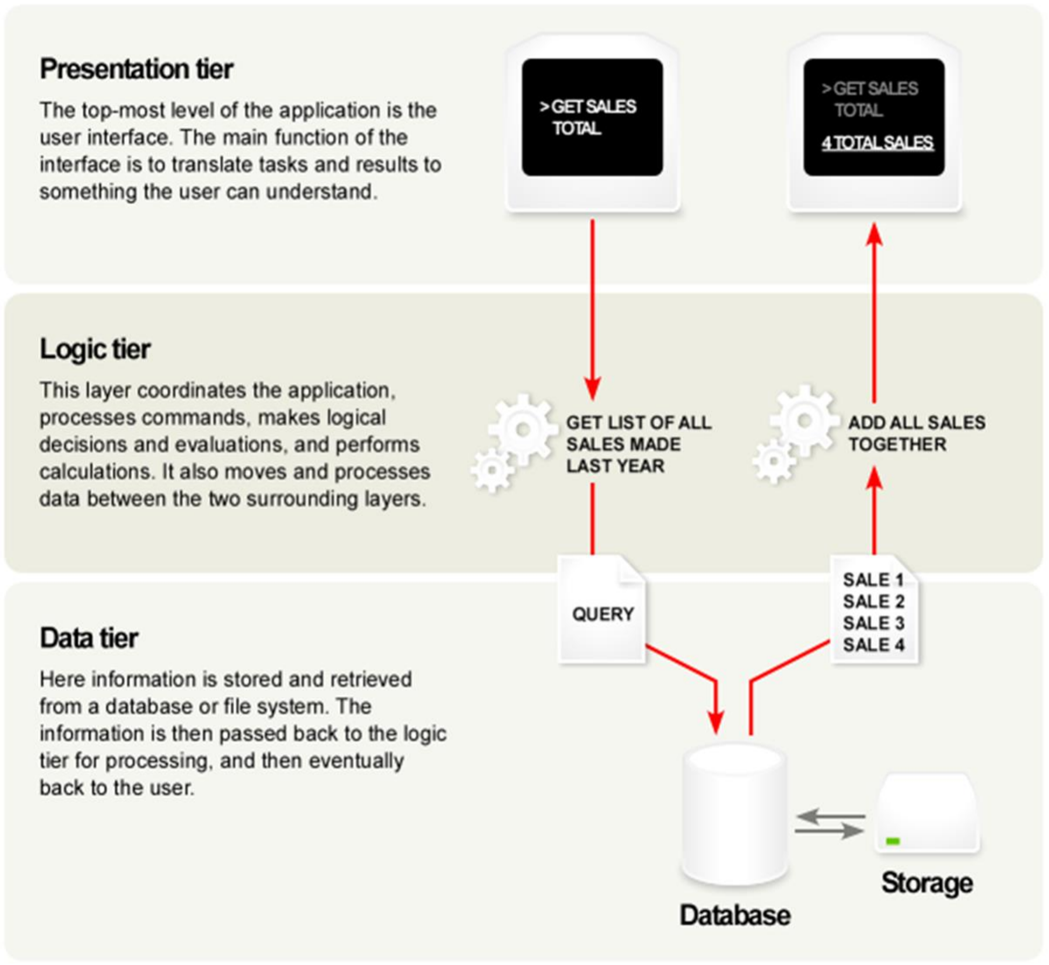
| Advantages | Disadvantages |
|---|---|
| *Development Issues:*<br>• Simple structure<br>• Easy to setup and maintain | *Development Issues:*<br>• Complex application rules difficult to implement in database server – requires more code for the client<br>• Complex application rules difficult to implement in client and have poor performance<br>• Changes to business logic not automatically enforced by a server – changes require new client side software to be distributed and installed<br>• Not portable to other database server platforms |
| *Performance:*<br>• Adequate performance for low to medium volume environments<br>• Business logic and database are physically close, which provides higher performance. | *Performance:*<br>• Inadequate performance for medium to high volume environments, since database server is required to perform business logic. This slows down database operations on database server. |

## 3-Tier Architecture



- Each layer can potentially run on a different machine
- Presentation, logic, data layers disconnected

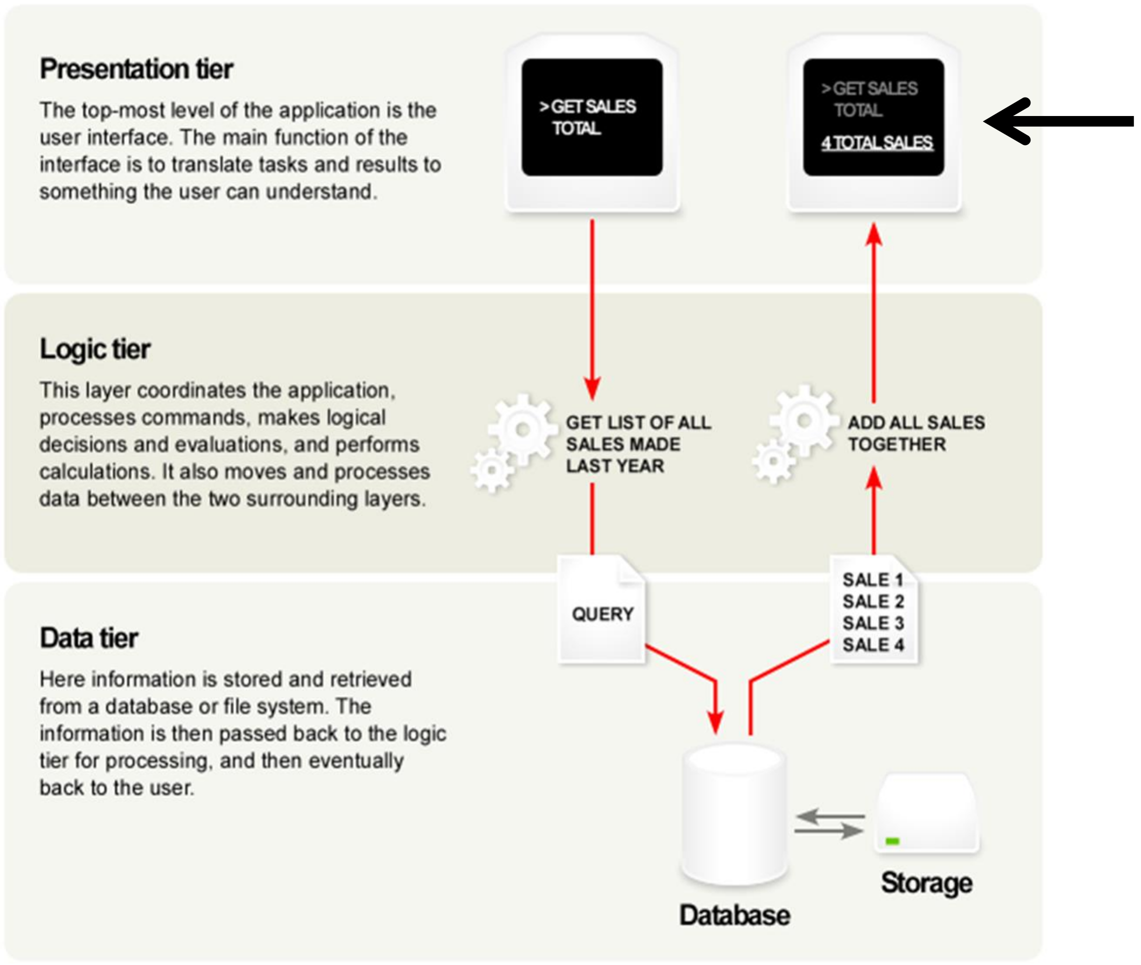
# A Typical 3-tier Architecture


Presentation tier
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.
>GET SALES TOTAL
>GET SALES TOTAL
4 TOTAL SALES
Logic tier
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.
GET LIST OF ALL SALES MADE LAST YEAR
ADD ALL SALES TOGETHER
Data tier
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.
QUERY
SALE 1
SALE 2
SALE 3
SALE 4
Database
Storage

## Architecture Principles

- Client-server architecture
- Each tier (Presentation, Logic, Data) should be independent and should not expose dependencies related to the implementation
- Unconnected tiers should not communicate
- Change in platform affects only the layer running on that particular platform

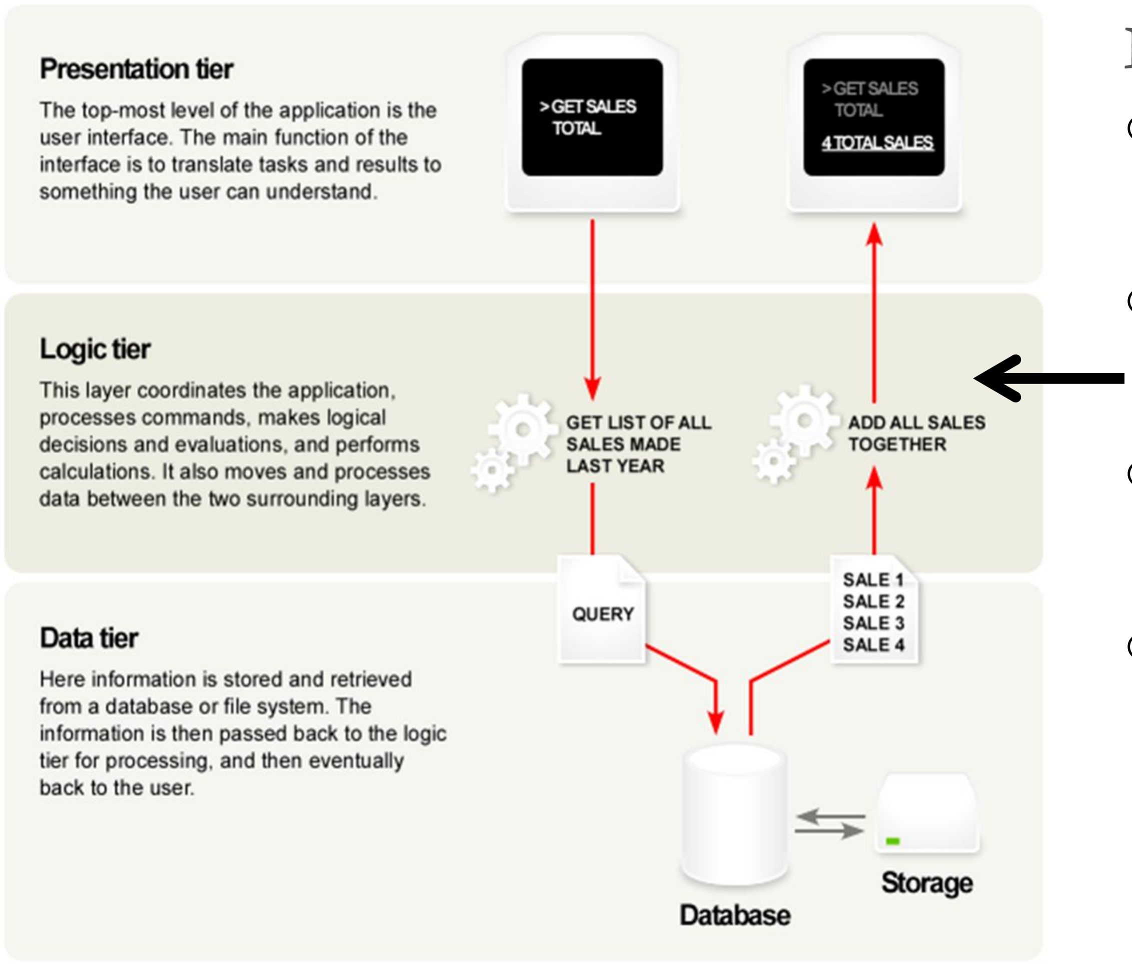
# A Typical 3-tier Architecture


Presentation tier
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.
>GET SALES TOTAL
>GET SALES TOTAL
4 TOTAL SALES
Logic tier
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.
GET LIST OF ALL SALES MADE LAST YEAR
ADD ALL SALES TOGETHER
QUERY
SALE 1
SALE 2
SALE 3
SALE 4
Data tier
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.
Database
Storage

## Presentation Layer

- Provides user interface
- Handles the interaction with the user
- Sometimes called the GUI or client view or front-end
- Should not contain business logic or data access code

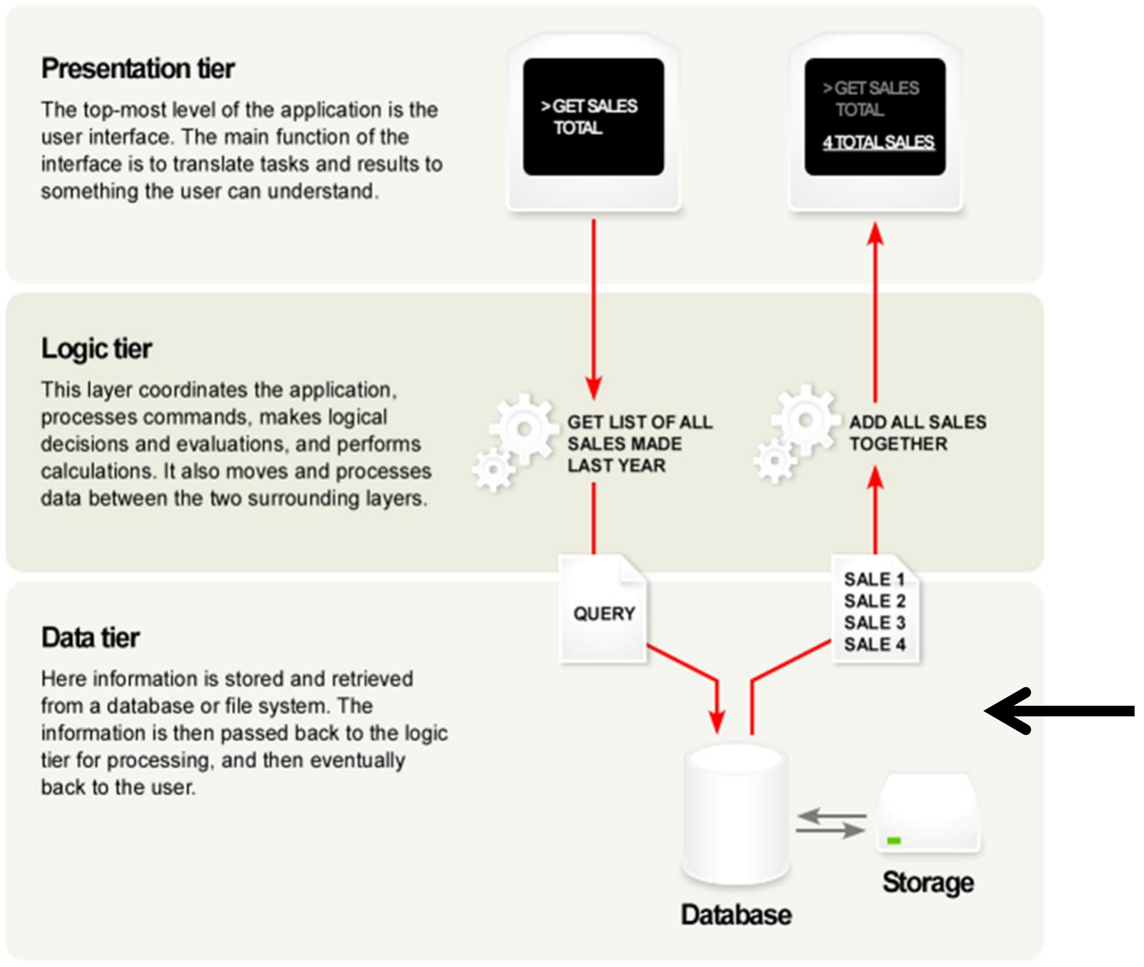
# A Typical 3-tier Architecture


Presentation tier
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.
>GET SALES TOTAL
>GET SALES TOTAL
4 TOTAL SALES
Logic tier
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.
GET LIST OF ALL SALES MADE LAST YEAR
ADD ALL SALES TOGETHER
QUERY
SALE 1
SALE 2
SALE 3
SALE 4
Data tier
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.
Database
Storage

## Logic Layer

- The set of rules for processing information
- Can accommodate many users
- Sometimes called middleware/ back-end
- Should not contain presentation or data access code

## A Typical 3-tier Architecture


Presentation tier
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.
>GET SALES TOTAL
>GET SALES TOTAL
4 TOTAL SALES
Logic tier
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.
GET LIST OF ALL SALES MADE LAST YEAR
ADD ALL SALES TOGETHER
QUERY
SALE 1
SALE 2
SALE 3
SALE 4
Data tier
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.
Database
Storage

**Data Layer**

- The physical storage layer for data persistence
- Manages access to DB or file system
- Sometimes called back-end
- Should not contain presentation or business logic code

## The 3-Tier Architecture for Web Apps

- **Presentation Layer**

  Static or dynamically generated content rendered by the browser (front-end)

- **Logic Layer**

  A dynamic content processing and generation level application server, e.g., Java EE, ASP.NET, PHP, ColdFusion platform (middleware)

- **Data Layer**

  A database, comprising both data sets and the database management system or RDBMS software that manages and provides access to the data (back-end)

## 8.0 3-Tier Client/Server Architecture

3-Tier client-server architectures have 3 essential components:

1. A Client PC
2. An Application Server
3. A Database Server

3-Tier Architecture Considerations:

- Client program contains presentation logic only
    - Less resources needed for client workstation
    - No client modification if database location changes
    - Less code to distribute to client workstations
- One server handles many client requests
    - More resources available for server program
    - Reduces data traffic on the network

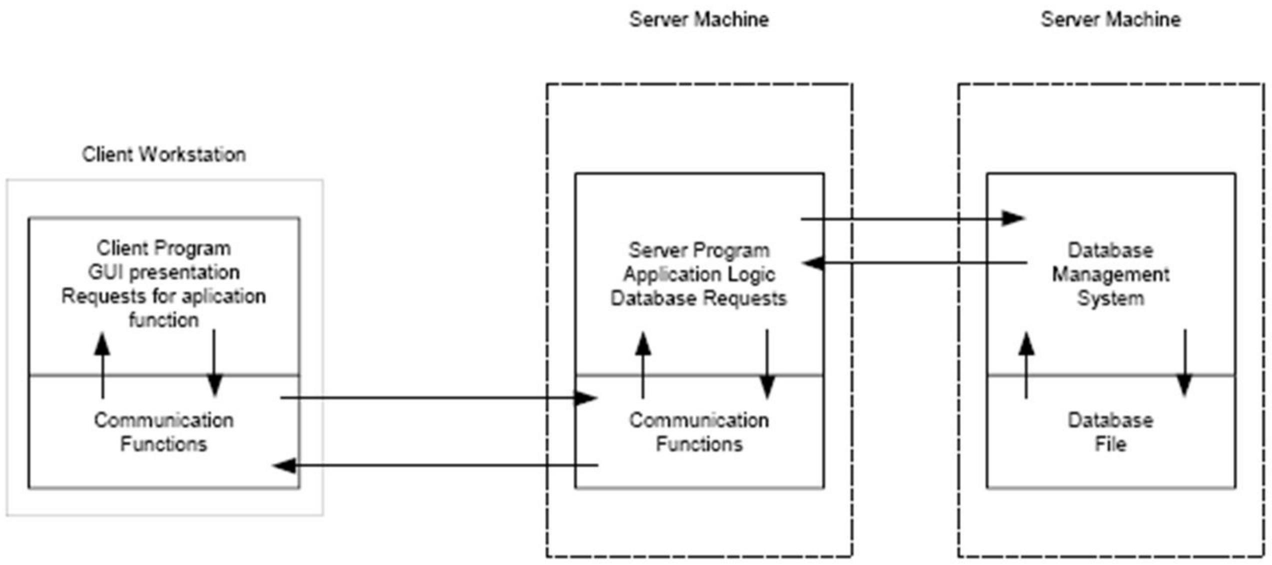
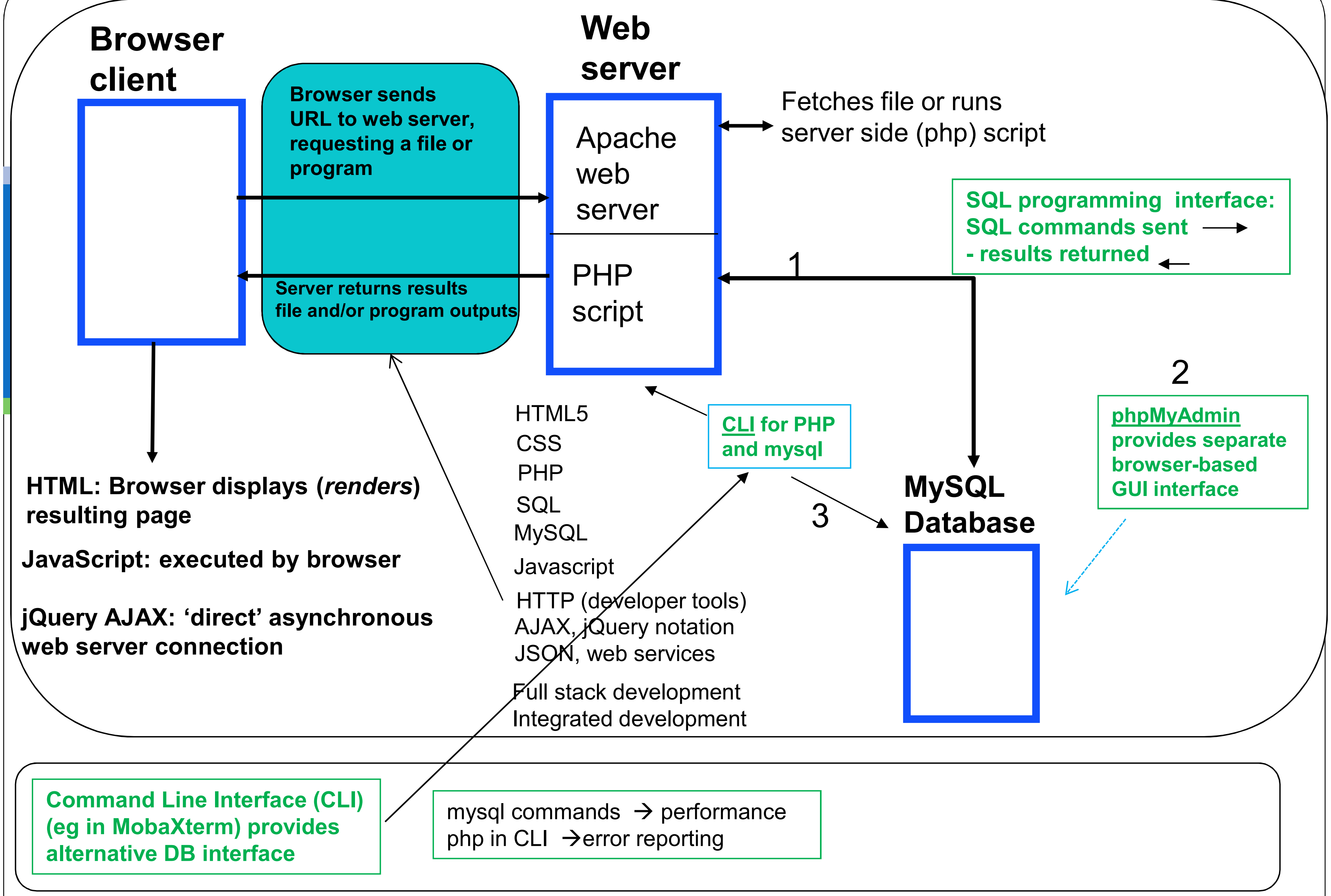


Figure 1.8. Typical 3 – Tier Architecture

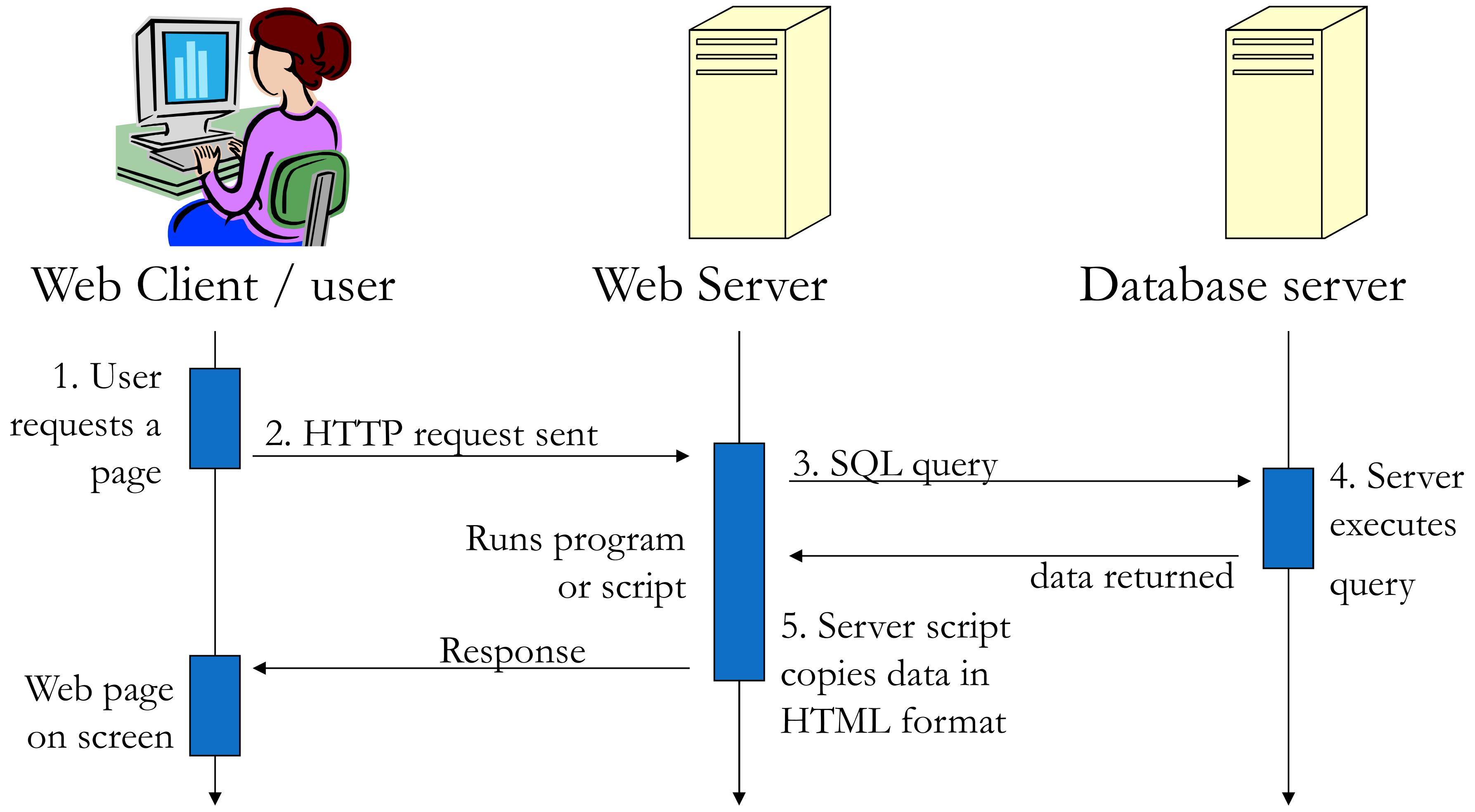
Run as slide show to see animated display.

## 3-Tier Architecture


Browser client
Browser sends URL to web server, requesting a file or program
Server returns results file and/or program outputs
Web server
Apache web server
PHP script
Fetches file or runs server side (php) script
SQL programming interface: SQL commands sent →
- results returned ←
1
2
3
phpMyAdmin provides separate browser-based GUI interface
CLI for PHP and mysql
MySQL Database
HTML: Browser displays (*renders*) resulting page
JavaScript: executed by browser
jQuery AJAX: 'direct' asynchronous web server connection
HTML5
CSS
PHP
SQL
MySQL
Javascript
HTTP (developer tools)
AJAX, jQuery notation
JSON, web services
Full stack development
Integrated development
Command Line Interface (CLI) (eg in MobaXterm) provides alternative DB interface
mysql commands → performance
php in CLI →error reporting

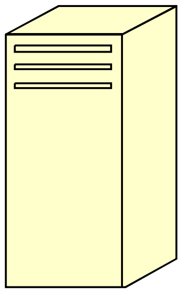
Also online meandeviation.com PHP syntax tool

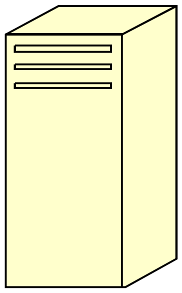
## Web sites based on data



## The “three tier architecture”

## Some technologies to use

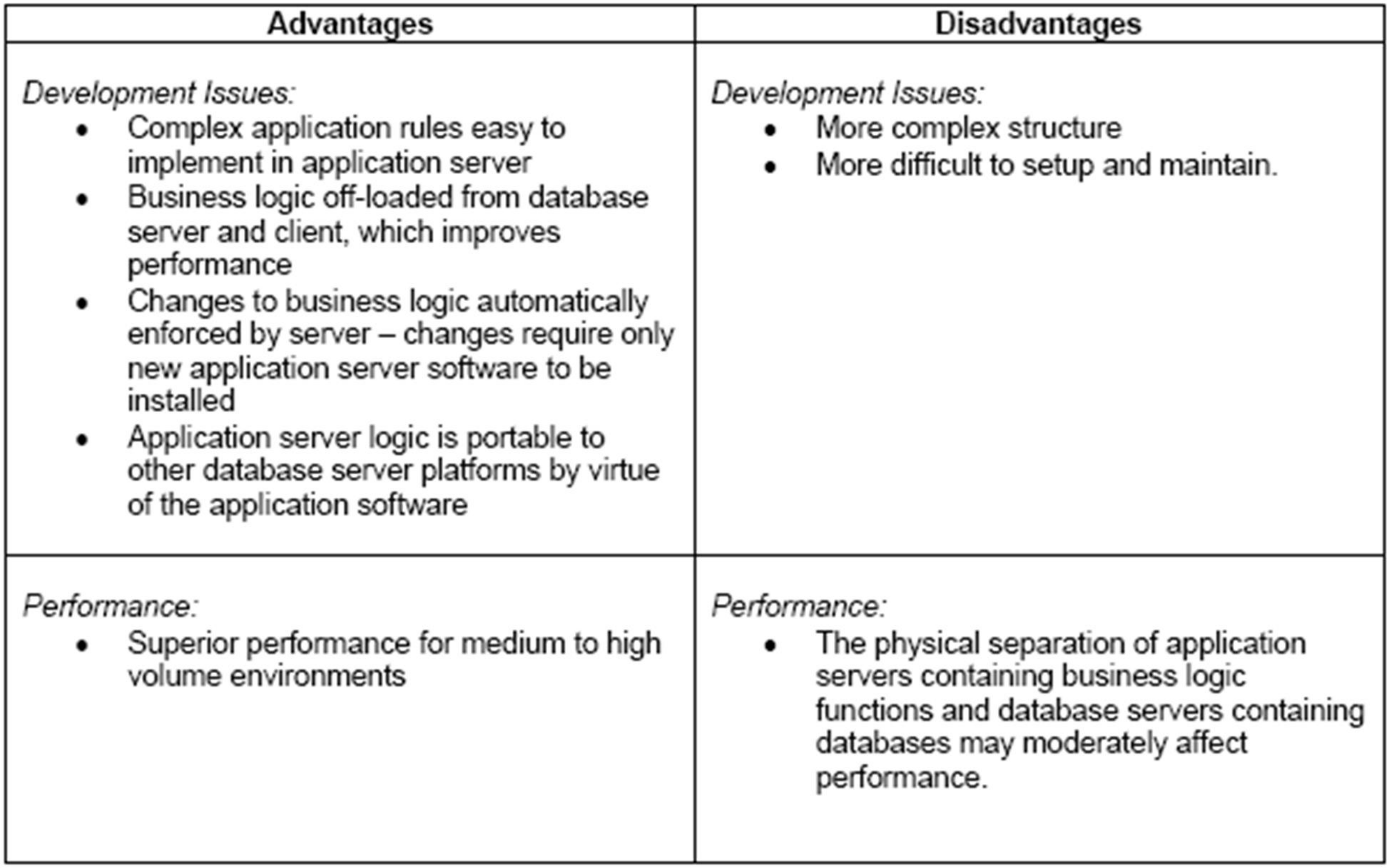




| Web Client / user | Web Server | Database server |
|---|---|---|
| Any Web browser | | |
| Client languages: HTML, CSS, JavaScript | Apache (most popular) | |
| | Server language: PHP | |
| | | MySQL |
| | | Query language: SQL |

## Bundled in the XAMPP package

## 3 – Tier Pros and Cons



| Advantages | Disadvantages |
|---|---|
| *Development Issues:*<br>• Complex application rules easy to implement in application server<br>• Business logic off-loaded from database server and client, which improves performance<br>• Changes to business logic automatically enforced by server – changes require only new application server software to be installed<br>• Application server logic is portable to other database server platforms by virtue of the application software | *Development Issues:*<br>• More complex structure<br>• More difficult to setup and maintain. |
| *Performance:*<br>• Superior performance for medium to high volume environments | *Performance:*<br>• The physical separation of application servers containing business logic functions and database servers containing databases may moderately affect performance. |

## 9.0 Middleware

Simplifies 3-tier application development and administration by providing an extra application server layer to manage communication between components.

**Middleware Characteristics:**

- Simplifies partitioning of application processing among clients and servers
- Manages distributed transactions among multiple databases
- Communicates with heterogeneous database products within a single application.
- Supports application scalability
- Supports service requests prioritization, load-balancing, data dependant routing and queuing.

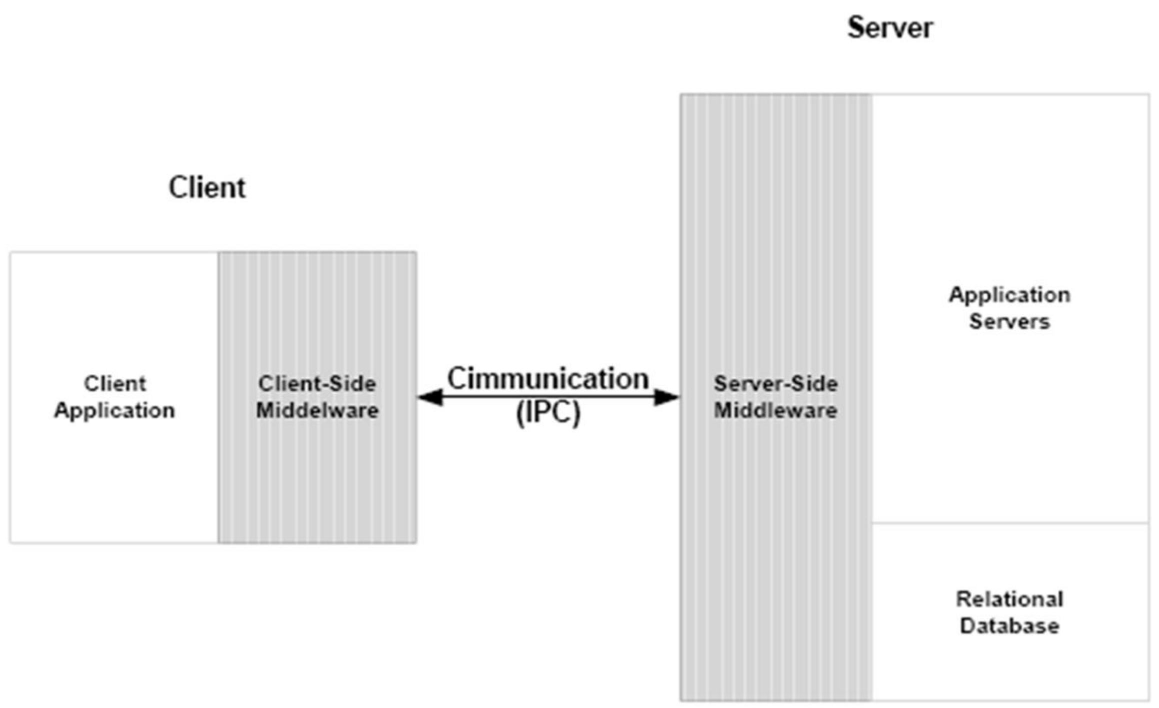


**Figure 1.9 Middleware**