


Web Technology – CACS 205

Tier Technology-Unit2

What is Tier Technology?

- A Tier Architecture is a software architecture in which different software components, organized in tiers (layers), provide dedicated functionality.
- The most common occurrence of a multi-tier architecture is a three-tier system consisting of a data management tier (mostly encompassing one or several database servers), an application tier (business logic) and a client tier (interface functionality).
- Sometimes, a “tier” can also be referred to as a “layer”
- Three layers involved in the application namely Presentation Layer, Business Layer and Data Layer.

Presentation Layer

- It is also known as Client layer.
- This is the layer we see when we use a software.
- By using this layer we can access the web pages.
- The main functionality of this layer is to communicate with Application layer.
- This layer passes the information which is given by the user in terms of keyboard actions, mouse clicks to the Application Layer.
- For example, login page of Gmail where an end user could see text boxes and buttons to enter user id, password and to click on sign-in. In a simple words, it is to **view** the application.

Application Layer

- It is also known as Business Logic Layer which is also known as logical layer
- As per the Gmail login page example, once user clicks on the login button, Application layer interacts with Database layer and sends required information to the Presentation layer
- It controls an application's functionality by performing detailed processing.
- This layer acts as a mediator between the Presentation and the Database layer.
- Complete business logic will be written in this layer.

Application Layer task

- Performing all required calculations and validations
- Managing workflow
- State management: to keep track of application execution
- Session management: to distinguish among application instances
- User identification
- Service access: to provide application services in a consistent way

Application Layer task

- Managing all data access for the presentation layer
- It is also known as Business Logic Layer which is also known as logical layer
- As per the Gmail login page example, once user clicks on the login button, Application layer interacts with Database layer and sends required information to the Presentation layer
- It controls an application's functionality by performing detailed processing.
- This layer acts as a mediator between the Presentation and the Database layer.
- Complete business logic will be written in this layer.

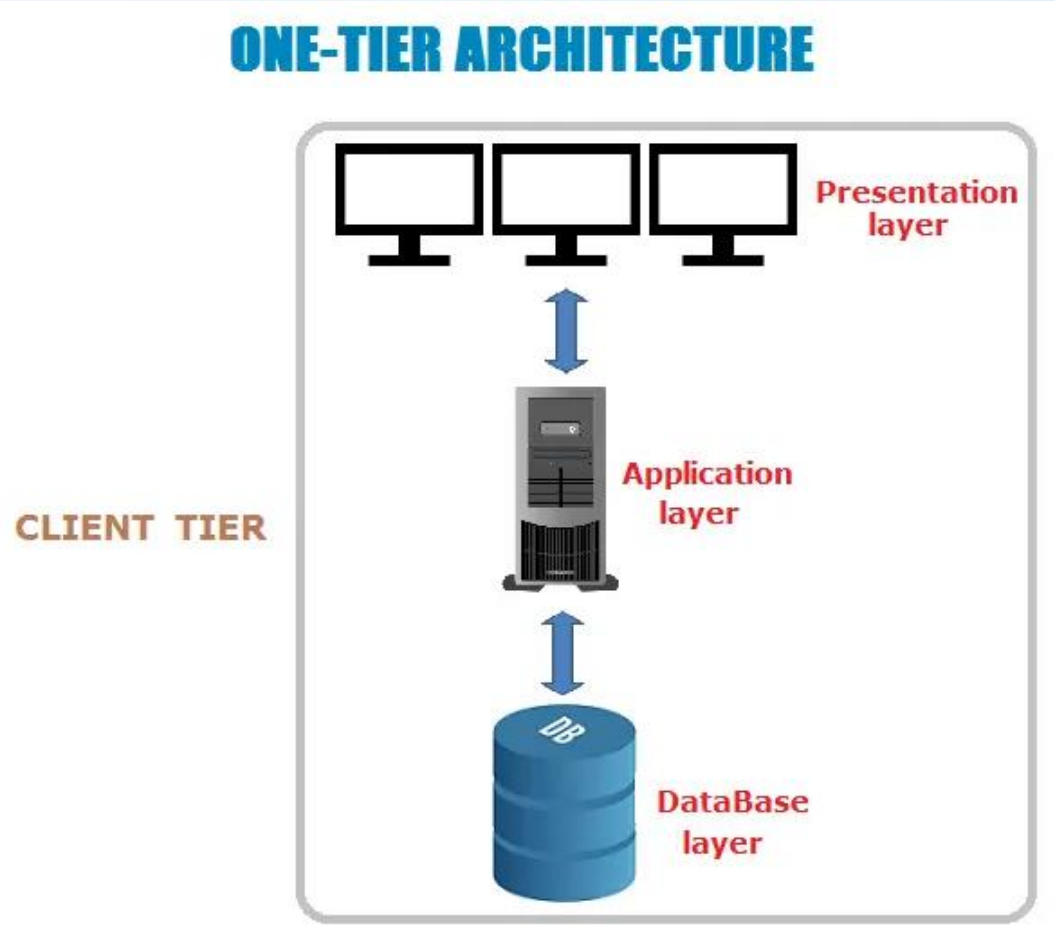
Data Layer

- The data is stored in this layer.
- Application layer communicates with Database layer to retrieve the data.
- It contains methods that connects the database and performs required action e.g.: insert, update, delete etc.
- In a simple words, it is to share and retrieve the data.

One Tier Architecture

- One Tier application also known as Standalone application.
- One tier architecture has all the layers such as Presentation, Business, Data Access layers in a single software package.
- Applications which handles all the three tiers such as MP3 player, MS Office are come under one tier application.
- The data is stored in the local system or a shared drive.

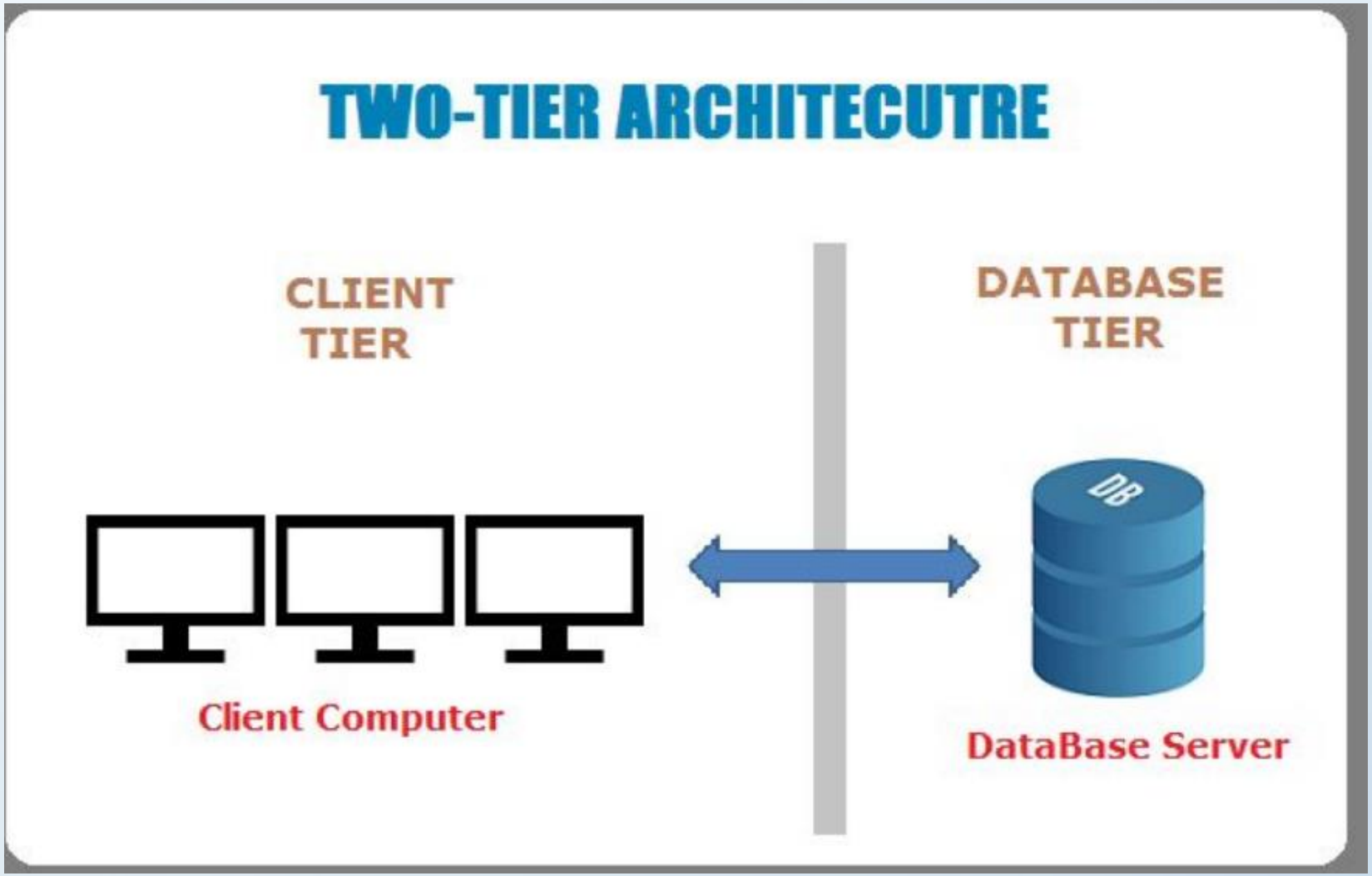
One Tier Architecture



Two-Tier Architecture

- Two Tier application also known as Client-Server application
- The Two-tier architecture is divided into two parts:
  - Client Application (Client Tier)
  - Database (Data Tier)
- Client system handles both Presentation and Application layers and Server system handles Database layer.
- It is also known as client server application.
- The communication takes place between the Client and the Server.
- Client system sends the request to the Server system and the Server system processes the request and sends back the data to the Client System

Two-Tier Architecture



Three-Tier Architecture

- Three Tier application also known as Web Based application.
- The Three-tier architecture is divided into three parts:
  - Presentation layer (Client Tier)
  - Application layer (Business Tier)
  - Database layer (Data Tier) Client system handles Presentation layer, Application server handles Application layer and Server system handles Database layer.

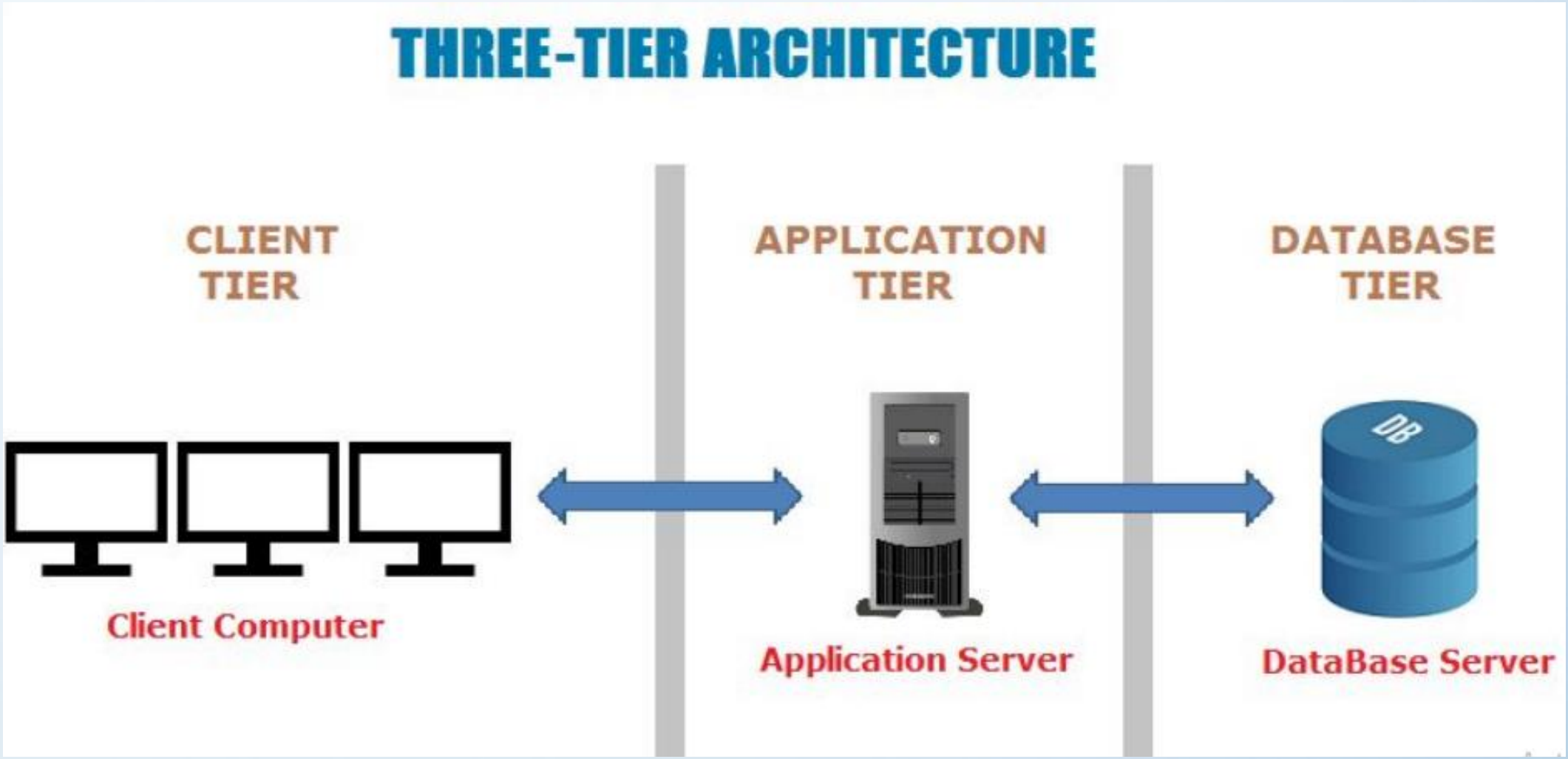
Benefits of a 3-tier app architecture

- The benefits of using a 3-layer architecture include improved horizontal scalability, performance and availability.
- With three tiers, each part can be developed concurrently by different team of programmers coding in different languages from the other tier developers.

Benefits of a 3-tier app architecture

- Because the programming for a tier can be changed or relocated without affecting the other tiers, the 3-tier model makes it easier for an enterprise or software packager to continually evolve an application as new needs and opportunities arise.
- Existing applications or critical parts can be permanently or temporarily retained and encapsulated within the new tier of which it becomes a component.

3-tier app architecture



N-Tier Architecture

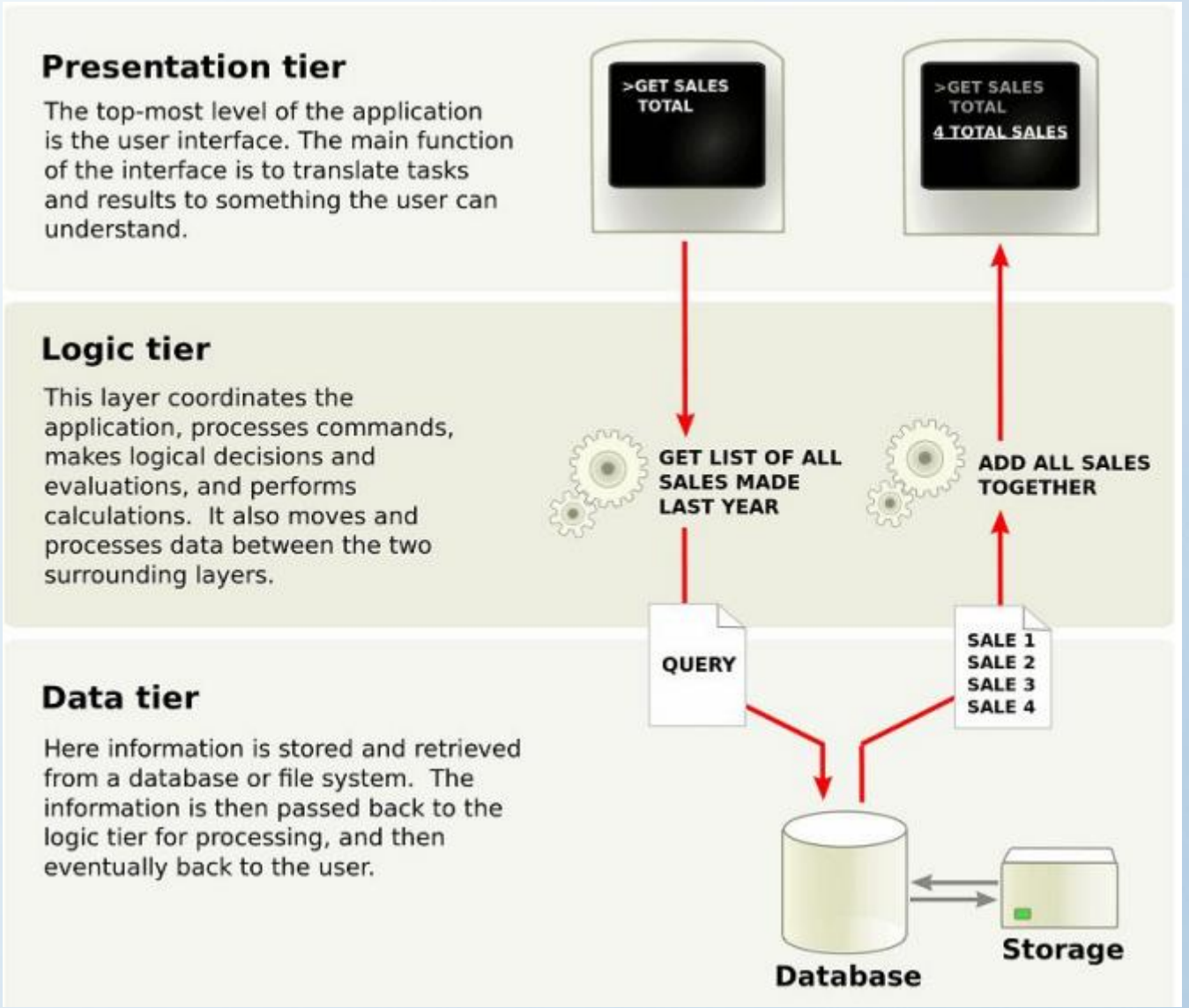
- N-tier architecture is also known as multi-tier architecture because the software is **engineered** to have the processing, data management, and presentation functions physically and logically separated.
- That means that these different functions are hosted on several machines or clusters, ensuring that services are provided without resources being shared and, as such, these services are delivered at top capacity
- The “N” in the name n-tier architecture refers to any number from 1.

N-Tier Architecture

- Not only does your software gain from being able to get services at the best possible rate, but it's also easier to manage.
- This is because when you work on one section, the changes you make will not affect the other functions.
- And if there is a problem, you can easily pinpoint where it originates.
- N-tier architecture would involve dividing an application into three different tiers

N-Tier Architecture

- Presentation tier
- Logic tier
- Data tier



Benefits of N-Tier Architecture

- Secure: You can secure each of the three tiers separately using different methods.
- Easy to manage: You can manage each tier separately, adding or modifying each tier without affecting the other tiers.
- Scalable: If you need to add more resources, you can do it per tier, without affecting the other tiers. With n-tier architecture, you can adopt new technologies and add more components without having to rewrite the entire application or redesigning your whole software, thus making it easier to scale or maintain.
- Flexible: Apart from isolated scalability, you can also expand each tier in any manner that your requirements dictate.

Benefits of N-Tier Architecture

- Meanwhile, in terms of security, you can store sensitive or confidential information in the logic tier, keeping it away from the presentation tier, thus making it more secure.
- More efficient development. N-tier architecture is very friendly for development, as different teams may work on each tier. This way, you can be sure the design and presentation professionals work on the presentation tier and the database experts work on the data tier.
- Easy to add new features. If you want to introduce a new feature, you can add it to the appropriate tier without affecting the other tiers.
- Easy to reuse. Because the application is divided into independent tiers, you can easily reuse each tier for other software projects. For instance, if you want to use the same program, but for a different data set, you can just replicate the logic and presentation tiers and then create a new data tier.

Advantages and Disadvantages of Multi-Tier Architectures

- **Advantages**
  - Scalability
  - Reusability
  - Reduced Distribution
  - Improved Security
  - Improved Availability
  - Data Integrity
- **Disadvantage**
  - Increase in Effort
  - Increase in Complexity

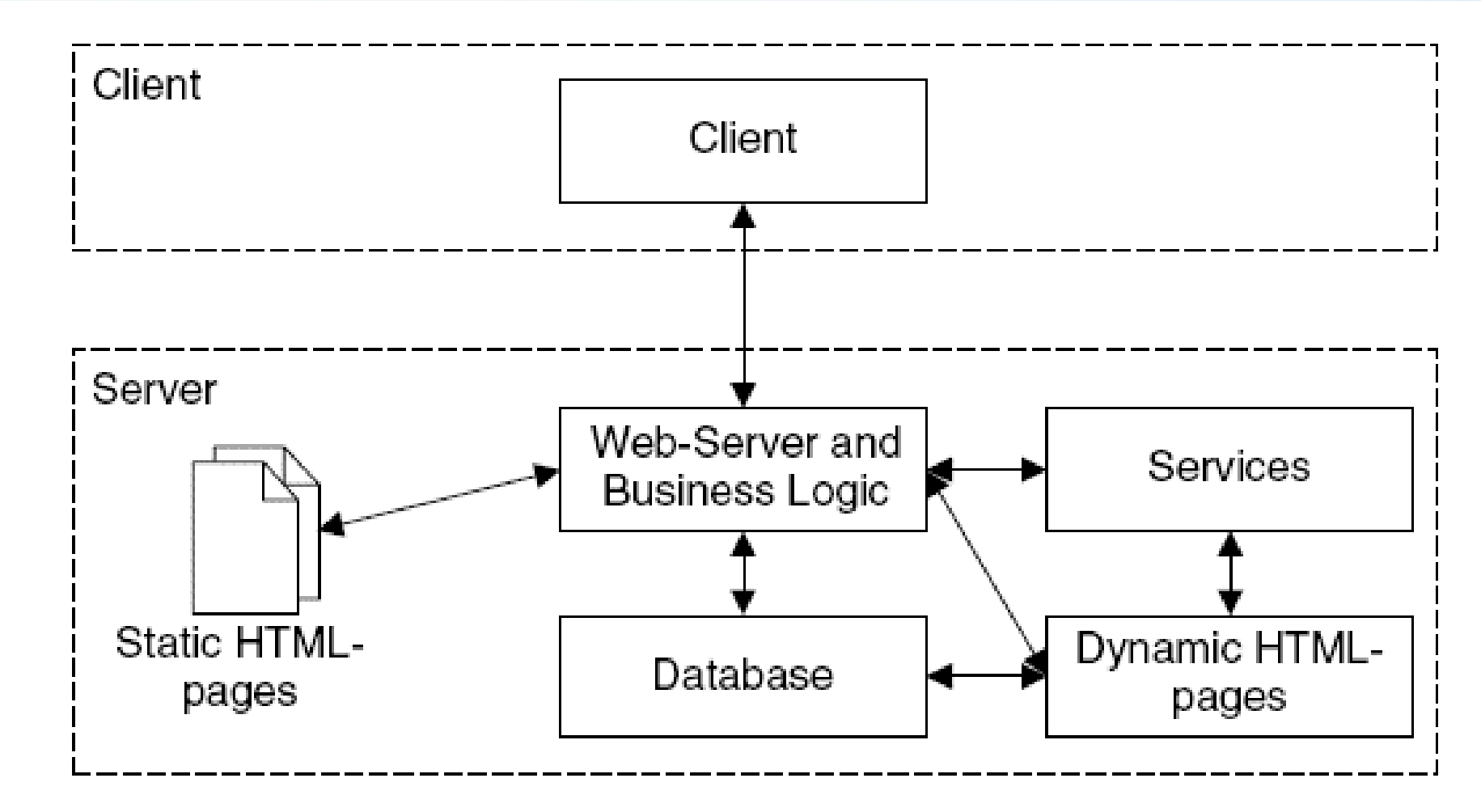
Architectural issues of web layer

- The web layer is also referred to as the **UI layer**. The web layer is primarily concerned with presenting the user interface and the behavior of the application (handling user interactions/events). While the web layer can also contain logic, core application logic is usually located in the services layer. The three Layers within the Web Layer are:
- **HTML-The Content Layer:** The content layer is where you store all the content that your customers want to read or look at. This includes text and images as well as multimedia. It's also important to make sure that every aspect of your site is represented in the content layer. That way, your customers who have JavaScript turned off or can't view CSS will still have access to the entire site, if not all the functionality.
- **CSS - the Styles Layer:** Store all your styles for your Web site in an external style sheet. This defines the way the pages should look, and you can have separate style sheets for various media types. Store your CSS in an external style sheet so that you can get the benefits of the style layer across the site.
- **JavaScript - the Behavior Layer:** JavaScript is the most commonly used language for writing the behavior layer; ASP, CGI and PHP can also generate Web page behaviors. However, when most developers refer to the behavior layer, they mean that layer that is activated directly in the Web browser - so JavaScript is nearly always the language of choice. You use this layer to interact directly with the DOM or Document Object Model.

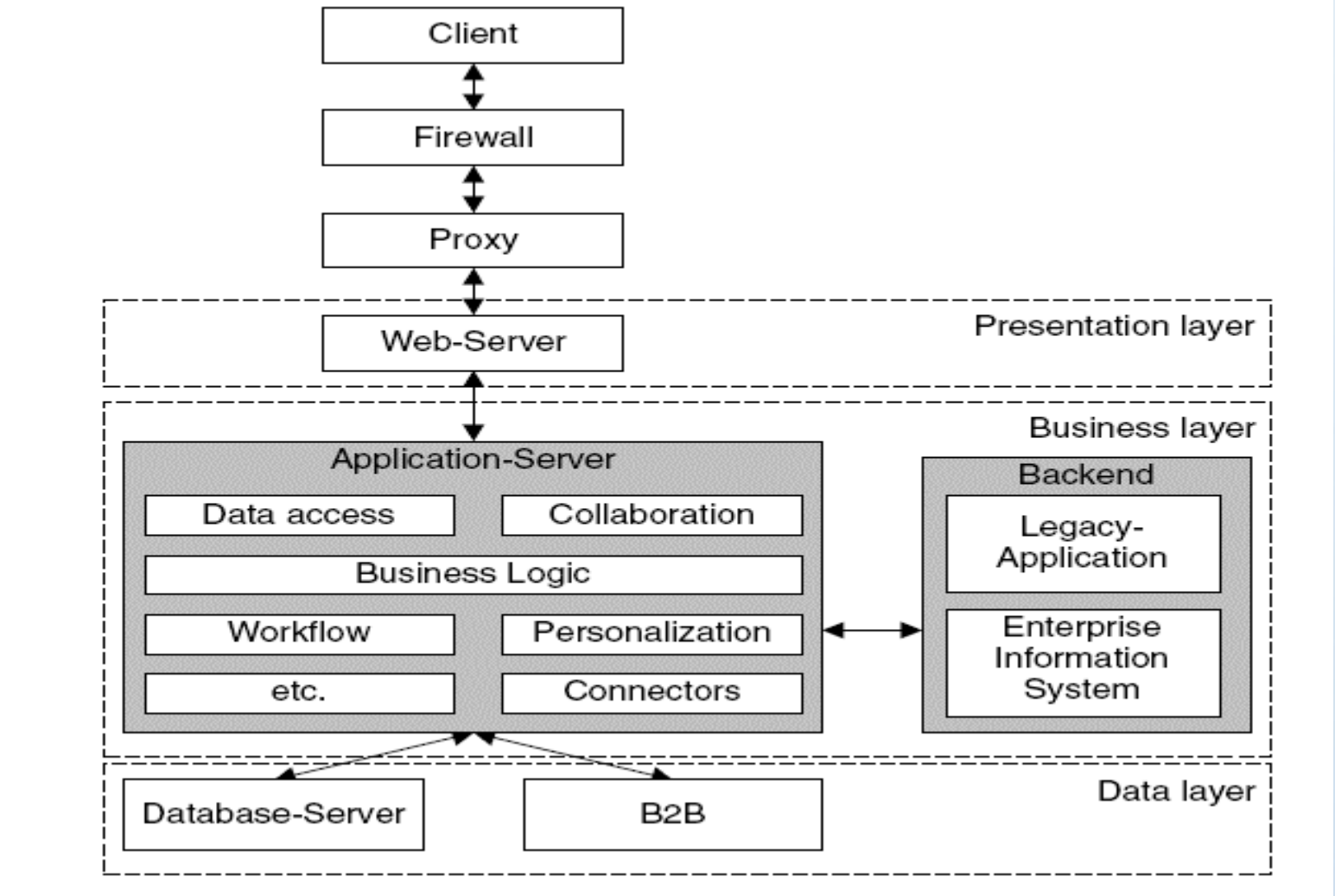
Architectural issues of web layer

- ***Benefits of separating the layers are:***
- Shared resources
- Faster downloads
- Multi-person teams
- Accessibility
- Backwards compatibility

The basic two parts for Web Applications



The standard three layered architecture for Web Applications



Thank you.