# Unit 2 (ERP)

# ERP Architecture

**ERP Architecture** refers to the underlying structure and design of an **Enterprise Resource Planning (ERP) system**, which dictates how the system is organized and how different components communicate with each other. It defines the software layers, databases, interfaces, and integration points that enable various functional modules (such as finance, human resources, sales, supply chain, etc.) to work together.

**Key Components of ERP Architecture**

1. **User Interface Layer (Presentation Layer)**:
    - This is the **front-end** of the ERP system that users interact with. It includes all the tools and interfaces that allow users to input data and retrieve information.
    - The interface can be web-based (for cloud ERP systems) or desktop-based (for on-premises systems).
    - It may include dashboards, forms, and data visualizations that help users make informed decisions.
2. **Application Layer**:
    - The **application layer** contains the core business logic and processes of the ERP system. It defines how data is processed and how different modules of the ERP system (like finance, HR, manufacturing, and inventory management) interact with each other.
    - This layer contains the business rules, workflows, and processes that support daily business operations.
    - It might consist of a **middleware** that enables communication between the user interface and the database.
3. **Database Layer (Data Layer)**:
    - The **database layer** is the backbone of the ERP system, where all the business data is stored.
    - This layer includes relational databases (such as SQL databases) that hold all the information about customers, suppliers, employees, financial transactions, and inventory.
    - ERP systems ensure data consistency, security, and integrity across different modules, ensuring a unified view of the enterprise's operations.
4. **Integration Layer**:
    - The **integration layer** connects the ERP system to other external systems and third-party applications (like CRM, supply chain management, or payroll systems).
    - Integration is typically achieved through APIs (Application Programming Interfaces), middleware, and other communication protocols.
    - This layer helps integrate data and processes from various external sources and ensures that the ERP system operates seamlessly with other enterprise tools.
5. **Middleware Layer**:
    - **Middleware** acts as a bridge between the user interface, application layer, and database layer.
    - It allows data to be processed and moved between different layers of the ERP system. Middleware can also facilitate communication between different ERP systems or between the ERP and third-party software.

6. **Reporting and Analytics Layer**:
    - The **reporting and analytics layer** helps businesses gather insights from the data stored in the ERP system. It enables users to generate financial reports, inventory reports, sales data, and other key performance indicators (KPIs).
    - Modern ERP systems often include advanced data analytics tools, such as business intelligence (BI) features, predictive analytics, and real-time dashboards, helping businesses make data-driven decisions.
7. **Security Layer**:
    - **Security** is a crucial part of ERP architecture, ensuring that sensitive business data is protected from unauthorized access or breaches.
    - The security layer includes user authentication, authorization, role-based access control (RBAC), encryption, and audit trails, ensuring compliance with industry regulations and protecting intellectual property.
8. **Cloud Layer (for Cloud-Based ERPs)**:
    - In cloud-based ERP systems, the **cloud layer** refers to the cloud infrastructure used to host the system and its components.
    - Cloud ERP offers scalability, flexibility, and cost efficiency by hosting ERP software on cloud servers rather than on-premises hardware. It can be accessed via the internet from anywhere, offering remote access and collaboration features.

**Layered Architecture in ERP**

**Layered Architecture in ERP (Enterprise Resource Planning)** refers to a structural model where the different components of an ERP system are organized into distinct layers. Each layer is responsible for a specific function or role in the overall architecture. Layered architecture helps to modularize the system, improve maintainability, ensure scalability, and enhance flexibility by separating concerns across the different parts of the system.

**Key Layers in ERP Architecture:**

1. **Presentation Layer (User Interface Layer)**:
    - **Purpose**: This layer interacts directly with the users. It's the front-end of the ERP system where users input data, interact with reports, dashboards, and other visual interfaces.
    - **Components**:
        - User interfaces (UI)
        - Dashboards
        - Forms and reports
    - **Technology**: Web-based or desktop-based technologies (HTML, CSS, JavaScript for web-based UIs; desktop applications may use .NET, Java, etc.)
    - **Key Function**: To provide an intuitive and user-friendly interface for all users (employees, managers, etc.).
2. **Application Layer (Business Logic Layer)**:
    - **Purpose**: This is the core of the ERP system. It contains all the business rules, logic, and processing required to perform various enterprise functions, such as sales, finance, inventory management, human resources, etc.
    - **Components**:
        - Business logic for handling transactions (order processing, payroll, finance, etc.)

        - Workflow management (approval processes, notifications)
        - Data validation and rules enforcement
    - **Technology**: This layer typically uses application frameworks, programming languages (Java, C#, Python), and middleware to process requests.
    - **Key Function**: It ensures that all business operations are carried out correctly by applying the necessary logic to the incoming data.
3. **Data Layer (Database Layer)**:
    - **Purpose**: This layer is responsible for storing all the business data in a structured and secure way. It holds the information that drives the core ERP functionality.
    - **Components**:
        - Relational databases (such as SQL Server, Oracle, MySQL)
        - Data storage for various business modules (e.g., employee records, sales orders, inventory levels)
        - Data management systems
    - **Technology**: Relational Database Management Systems (RDBMS), Data Warehouses, NoSQL (depending on system requirements).
    - **Key Function**: Ensures data integrity, consistency, and availability for the ERP system. Data is stored centrally, allowing for efficient querying, reporting, and analysis.
4. **Integration Layer (Middleware Layer)**:
    - **Purpose**: The integration layer connects the ERP system to external systems, services, and applications. It facilitates communication between different layers within the ERP system and between the ERP and other enterprise systems (e.g., CRM, supply chain management, third-party services).
    - **Components**:
        - APIs (Application Programming Interfaces)
        - Web Services (RESTful, SOAP)
        - Middleware (Integration platforms like Apache Kafka, MuleSoft, or Oracle Fusion)
        - Data connectors
    - **Technology**: Middleware technologies, API management tools, service buses, and communication protocols.
    - **Key Function**: To ensure that data flows seamlessly between various external and internal systems. It allows for integration with third-party systems like CRM or eCommerce platforms and ensures real-time updates.
5. **Security Layer**:
    - **Purpose**: The security layer is responsible for ensuring the protection of sensitive business data and securing access to various ERP modules.
    - **Components**:
        - Authentication and authorization mechanisms (user login, roles, permissions)
        - Data encryption (for data storage and transmission)
        - Audit trails (to track who accessed what data and when)
    - **Technology**: Security protocols like SSL/TLS, OAuth for secure authentication, encryption standards, and firewalls.

- **Key Function**: Protect sensitive data, ensure compliance with regulations (such as GDPR), and restrict unauthorized access to the system.

6. **Reporting & Analytics Layer**:
   - **Purpose**: This layer handles the extraction of data from the ERP system, processes it, and generates reports or analytical insights to help users make business decisions.
   - **Components**:
     - Reporting tools (e.g., Crystal Reports, Microsoft Power BI, Tableau)
     - Analytical capabilities (predictive analytics, business intelligence tools)
     - Dashboards for monitoring KPIs (Key Performance Indicators)
   - **Technology**: BI tools, OLAP (Online Analytical Processing), data mining technologies.
   - **Key Function**: To provide real-time business intelligence, generate reports, and visualize data to support decision-making.

**Advantages of Layered Architecture in ERP:**

1. **Separation of Concerns**:
   - Each layer is responsible for a specific aspect of the ERP system. This modular approach allows for easy maintenance, upgrades, and troubleshooting. For example, changes to the user interface do not affect the business logic or data storage layers.
2. **Scalability**:
   - Since each layer operates independently, ERP systems built with layered architecture can scale more effectively. New modules, users, or features can be added with minimal disruption to existing systems.
3. **Flexibility**:
   - Layers can be modified or replaced independently. For example, a business may choose to replace the database layer with a different RDBMS without affecting the other layers. Similarly, the presentation layer could be updated to enhance user experience without impacting business logic.
4. **Enhanced Security**:
   - Security measures are implemented across multiple layers (e.g., user authentication in the presentation layer, data encryption in the database layer). This multi-layered approach ensures robust protection of sensitive business data.
5. **Improved Integration**:
   - The integration layer allows ERP systems to easily integrate with other external systems. This is crucial in modern businesses where multiple applications (CRM, SCM, third-party eCommerce platforms) need to communicate with the ERP system in real-time.
6. **Better Maintenance**:
   - Since the layers are decoupled, it's easier to maintain and update parts of the system without causing disruptions in other areas. For example, the application layer can be updated without changing how the data layer is structured.

**1. Types of ERP Architecture**

Enterprise Resource Planning (ERP) systems use various architectures to support the integration and management of business processes. These architectures can be categorized into different types based on their structure, deployment, and interaction mechanisms.

## Two-Tier Implementations

**Definition:**

- In a two-tier ERP architecture, the system is divided into two layers: a client tier and a server tier.

**Components:**

1. **Client Tier (User Interface Layer):**
    - This is the front-end layer where users interact with the ERP system.
    - It often involves desktop or web-based applications where users input data, view reports, and perform operations.
2. **Server Tier (Data and Business Logic Layer):**
    - This back-end layer processes the business logic and stores data.
    - It includes the database server, application server, and business logic processing.
    - The data is stored in a central database, and the application server processes requests from the client tier.

**Advantages:**

- Simpler architecture that is easier to implement and maintain.
- Lower infrastructure costs compared to more complex multi-tier architectures.

**Disadvantages:**

- Scalability issues with large datasets or high user concurrency.
- Limited flexibility in handling complex business processes.

**Use Cases:**

- Ideal for small to medium-sized organizations with fewer complex business operations.

## Three-Tier Client/Server Implementations

**Definition:**

- A three-tier ERP architecture adds an additional layer to the two-tier system, introducing a middle layer (business logic or application layer) between the client and server.

**Components:**

1. **Client Tier (Presentation Layer):**
    - The client provides the user interface (UI) for interacting with the ERP system.

    - The client sends user requests to the middle layer and receives data to display.
2. **Middle Layer (Business Logic/ Application Layer):**
    - This layer handles the business logic and application processing.
    - It serves as an intermediary between the client and server, ensuring that only the necessary data is sent to the client and that all logic is processed.
3. **Server Tier (Data Layer):**
    - This layer contains the database and handles data storage and retrieval.
    - It ensures data consistency and supports various data operations required by the application layer.

**Advantages:**

- Better scalability and performance compared to two-tier systems.
- Allows easier maintenance and updates to the business logic layer.
- More secure and flexible, suitable for large enterprises.

**Disadvantages:**

- More complex than two-tier implementations.
- Higher costs for infrastructure and maintenance.
- Increased complexity in managing multiple tiers.

**Use Cases:**

- Suitable for medium to large organizations with complex business processes and higher security or data management requirements.

**Web-based Architecture**

**Definition:**

- Web-based ERP architecture involves using web technologies for deploying ERP systems, where the user interface is web-based, and the system is hosted on a web server.

**Components:**

1. **Web Client:**
    - The user interface is a web browser, eliminating the need for dedicated client software.
    - It provides accessibility from any device with a web browser and internet connection.
2. **Web Server (Application Layer):**
    - A web server processes client requests and interfaces with the application logic.
    - It is responsible for serving web pages, data, and functionality.
3. **Database Server:**
    - The central data repository where all information is stored.
    - It can be hosted on a cloud infrastructure or on-premises.

**Advantages:**

- Accessible from anywhere with an internet connection.
- Reduces the need for managing client-side software.
- Easier deployment and updates since the application resides on the server.

**Disadvantages:**

- Potential security concerns with internet-based access.
- Performance may be dependent on internet bandwidth and server capacity.
- Requires stable internet connectivity for functionality.

**Use Cases:**

- Cloud-based ERP systems or organizations with a need for mobile accessibility and distributed users.

**Service-Oriented Architectures (SOA)**

**Definition:**

- Service-Oriented Architecture (SOA) is an architectural style where the application components (services) communicate over a network using
- standardized protocols, allowing the ERP system to be more flexible and modular.

**Components:**

1. **Services:**
    - Core ERP functionality is broken into smaller services that perform specific tasks, such as order management, inventory, or accounting.
    - These services can be deployed independently and interact with other services.
2. **Service Bus:**
    - A middleware layer that ensures communication between services.
    - It can include message queues, transformation services, and service orchestration to enable seamless integration.
3. **Data Layer:**
    - Each service accesses a shared data layer or database, ensuring that data is consistent across the system.

**Advantages:**

- Flexibility to update and replace individual services without impacting the entire system.
- Improved scalability and integration with other systems.
- Easy to adopt new technologies and third-party services.

**Disadvantages:**

- Higher complexity in managing the communication and interaction between services.
- Increased latency due to multiple service calls.

**Use Cases:**

- Large organizations needing high modularity, flexibility, and the ability to integrate multiple business processes across various platforms.

**Logical Architecture of an ERP System**

**Definition:**

- Logical architecture refers to the conceptual design of the ERP system, outlining how different system components (applications, modules, and data) interact with each other, independent of the physical hardware.

**Components:**

1. **Modules/Applications:**
    - Logical divisions of the ERP system into modules such as finance, procurement, manufacturing, HR, and sales.
2. **Integration Layer:**
    - Ensures communication between various modules and external systems, often through APIs or middleware.
3. **Data Flow:**
    - Outlines how data flows between modules, ensuring that information is shared seamlessly across business processes.

**Importance:**

- Logical architecture defines how ERP functions should work and interact.
- It serves as a blueprint for developing and deploying ERP applications.

**Advantages:**

- Helps in understanding the interrelationships of different business processes.
- Aids in defining the user experience and data flow across modules.

**Physical Architecture of an ERP System**

**Definition:**

- The physical architecture refers to the actual hardware and software resources that are used to support the ERP system. It includes servers, networks, storage, and database systems.

**Components:**

1. **Client Devices:**
    - Desktops, laptops, or mobile devices through which users access the ERP system.
2. **Application Servers:**
    - Physical or virtual servers running the ERP application software.
3. **Database Servers:**
    - Servers that store and manage the ERP system's data.
4. **Network Infrastructure:**

- Ensures communication between the client devices, application servers, and database servers.

**Importance:**

- The physical architecture ensures that the system is scalable, reliable, and secure.
- It determines how resources are allocated and managed across the network.

**Evaluation Framework for ERP Acquisition**

**Definition:**

- An ERP evaluation framework helps businesses assess and select an ERP system that best fits their requirements.

**Key Factors:**

1. **Business Needs:**
   - Define the primary objectives and business processes the ERP system must support.
2. **Cost:**
   - Evaluate both initial acquisition costs and long-term operational costs (licensing, maintenance, and training).
3. **Scalability and Flexibility:**
   - The ability of the ERP system to grow with the business and adapt to changing needs.
4. **Integration Capabilities:**
   - Ensure the system can integrate with existing software and third-party systems.
5. **Vendor Reputation and Support:**
   - Assess the reputation of the vendor, customer service, and post-implementation support.
6. **User-Friendliness:**
   - Evaluate the ease of use and training requirements for end users.
7. **Security:**
   - Ensure the system has robust security features to protect business data.