BCA Programme
Sem-V

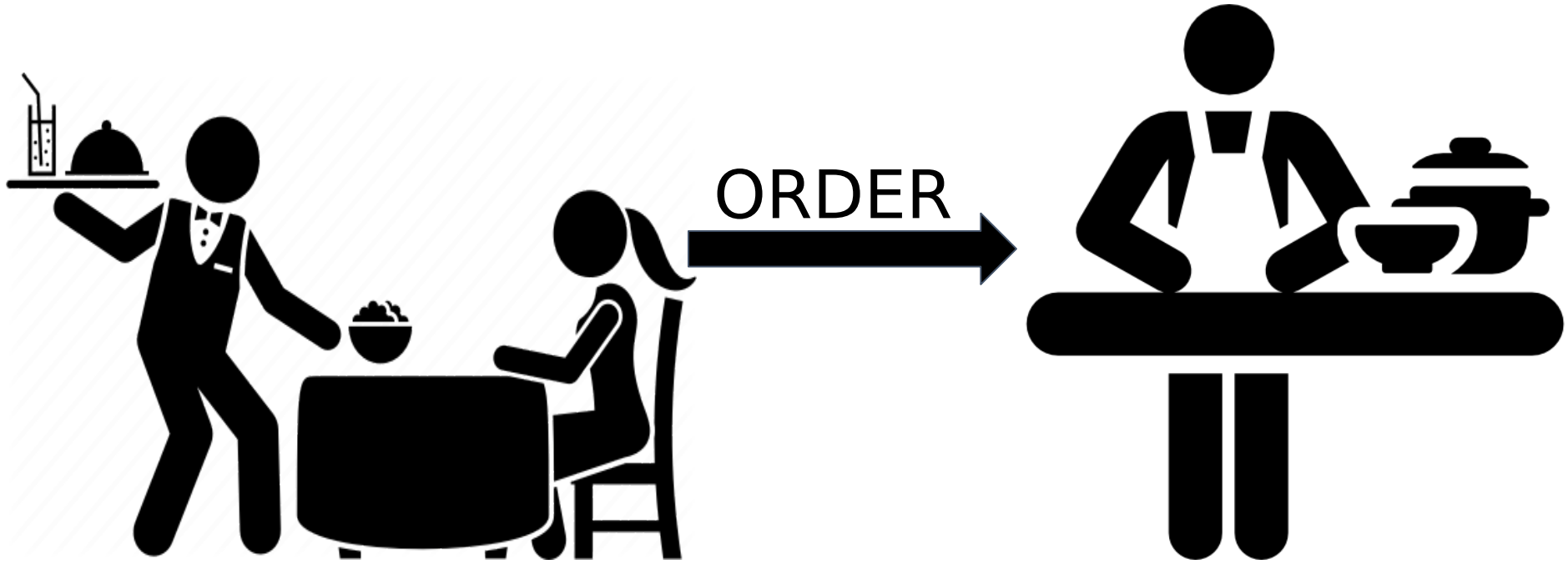
210302503

UNIT 1

WEB SERVICES

Example of Web Service

Example where a customer can directly place order.

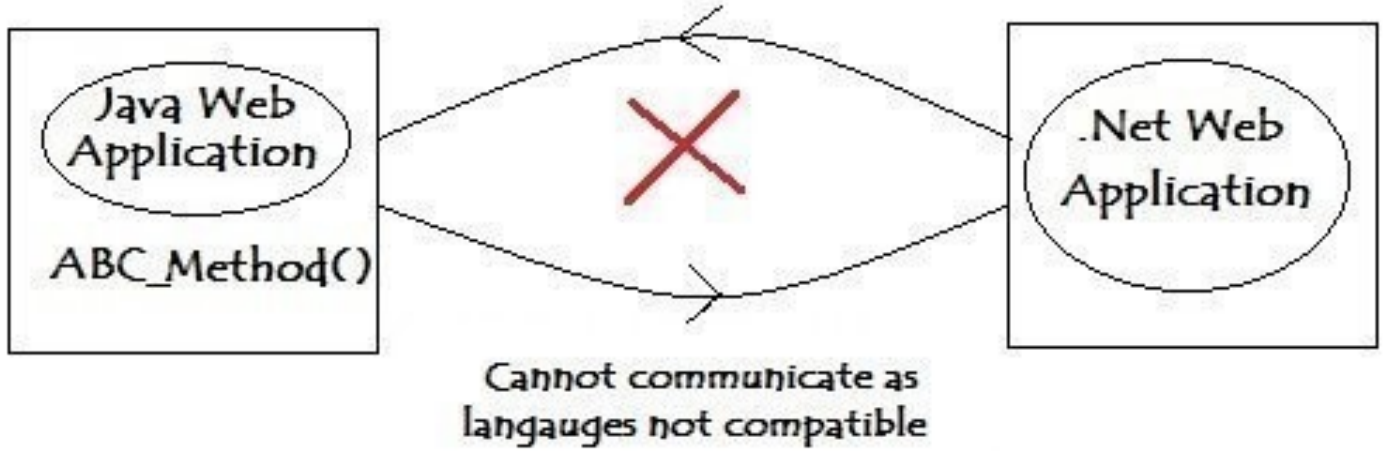


Case: Where the customer knows only Hindi and Chef knows only English

So Here the Waiter is playing role of web-service.

Why the Web Services Required

- Modern day business applications use variety of programming platforms to develop web-based applications. Some applications may be developed in Java, others in .Net, while some other in Angular JS, Node.js, etc.
- Most often than not, these heterogeneous applications need some sort of communication to happen between them.
- Since they are built using different development languages, it becomes really difficult to ensure accurate communication between applications.
- Here is where web services come in. Web services provide a common platform that allows multiple applications built on various programming languages to have the ability to communicate with each other.


Java Web Application
ABC_Method()
.Net Web Application
Cannot communicate as langauges not compatible


Call ABC Web Service
Java ABC Web Service
XML/HTTP
.Net Web Application
Fetch Results

- With its universal communication protocols, the web service method can help developers segment applications into components that can be reused for various needs.

- Businesses wanting to get revenue from data and compute services, often use data sources as a business service that can be provided to consumers.

- Web services can be used to **link data between different platforms**. Because all applications can include a web services component, this can turn any program into a fully interoperable one.

What is Web Service

- **Web service is a technology to communicate one programming language with another.**
- **Web services are XML-based information exchange systems that use the Internet for direct application-to-application interaction. These systems can include programs, objects, messages, or documents.**
- **A web service is a collection of open protocols and standards used for exchanging data between applications or systems.**

For example, java programming language can interact with PHP and .Net by using web services. In other words, web service provides a way to achieve interoperability.

A complete web service is

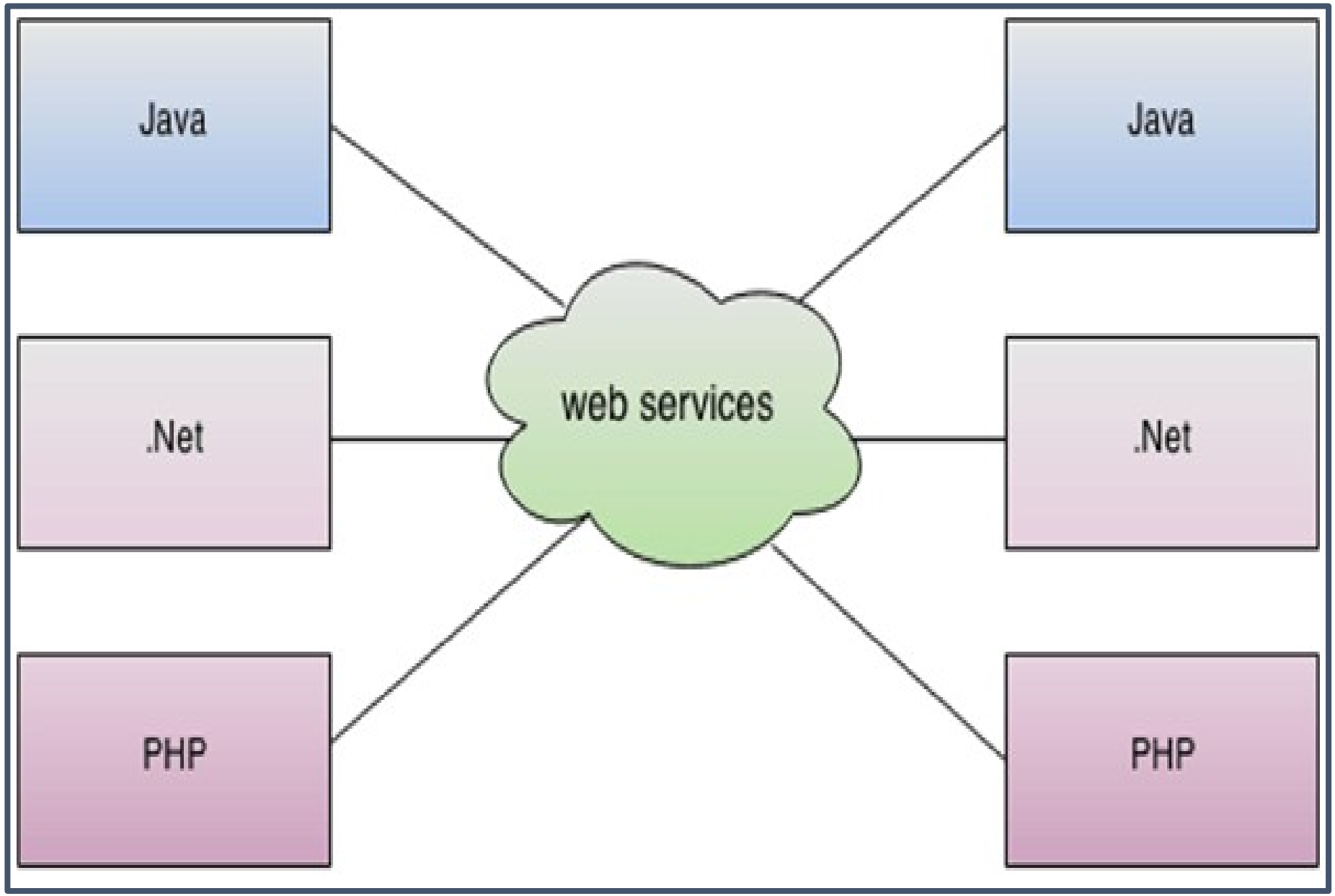
- Available over the Internet or private (intranet) networks
- Uses a standardized XML messaging system
- Not tied to any one operating system or programming language
- Self-describing via a common XML grammar
- Discoverable via a simple find mechanism

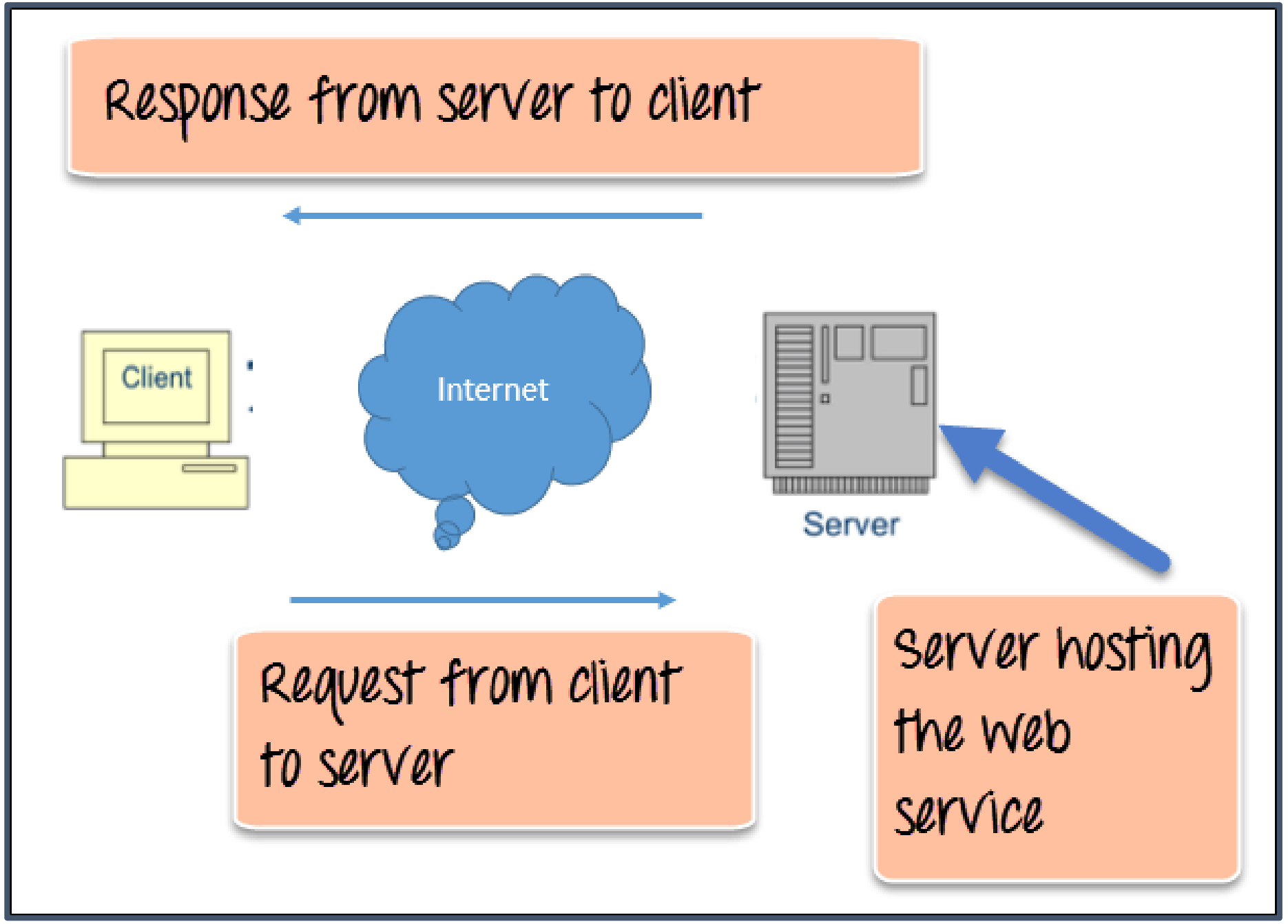
What is Web-service

- It is a client-server application or application component for communication.
- The method of communication between two devices over the network.

  It is a software system for the interoperable machine to machine communication.

  It is a collection of standards or protocols for exchanging information between two devices or application.


Java
.Net
PHP
web services
Java
.Net
PHP


Response from server to client
Client
Internet
Server
Request from client to server
Server hosting the web service

WEB SERVICE APPLICATION OPPORTUNITIES

- **Business Software Connection**

Web services help different business software systems work together smoothly.

- **Online Shopping and Payments**

They are used in online stores for payments, order tracking, and managing stock.

- **Cloud Services**

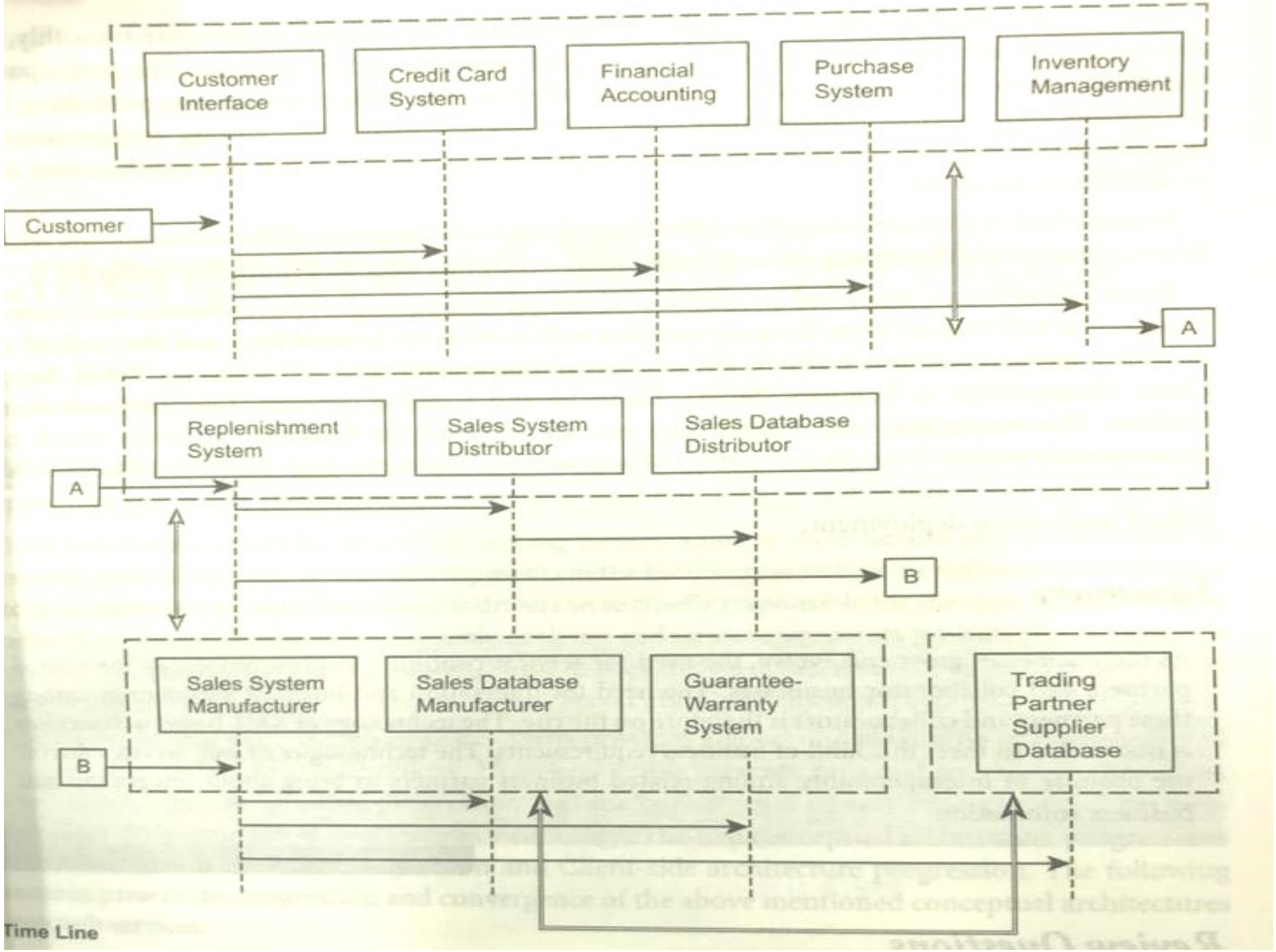
Web services let us use cloud platforms like Google Cloud or AWS to store data and run apps online.

- **Mobile Apps**
- Mobile apps use web services to log in users, send messages, and store data online.
- **Smart Devices (IoT)**
- Devices like smart TVs and health monitors use web services to send and receive data over the internet.
- **Social Media Connections**
- Apps use web services to connect with Facebook, Google, and other platforms for login and sharing.

- **Online Education**
- E-learning websites use web services to show videos, conduct live classes, and manage student data.

- **Healthcare**
- Hospitals use web services to share patient information, manage reports, and support telemedicine.

Examples


Customer Interface
Credit Card System
Financial Accounting
Purchase System
Inventory Management
Customer
A
Replenishment System
Sales System Distributor
Sales Database Distributor
A
B
Sales System Manufacturer
Sales Database Manufacturer
Guarantee-Warranty System
Trading Partner Supplier Database
B

Time Line

Steps

- Charging of the consumer's credit/debit card.
- Sending notification to the financial accounting application of the retail store.
- Sending notification to the inventory management application of the retail store.
- Sending notification to the replenishment application of the distributor, in case t inventory falls below certain level.
- Database updating of the retail store of the new purchase.
- Sending notification to the sales application of the distributor.
- Updating the database about the customer and purchase order details for custon support.
- Sending notification to the manufacturer's sales application program.
- Updating manufacturers database about the new sale and consumer information.
- Sending notification to the manufacturer's guarantee and warranty applications.
- Updating the databases of the suppliers and trading partners of the manufacturer.

Diagram shows the Multi-level business process using web services, showing how information flows between various systems in a supply chain and customer management environment. It is divided into three main layers, marked by labels A and B, and progresses along a timeline (bottom to top).

- Top Layer (Customer Interaction & Internal Systems)
- Middle Layer (Distributor-Level Operations)
- Bottom Layer (Manufacturer & External Systems)

Top Layer (Customer Interaction & Internal Systems)

This layer shows the initial customer interaction and the enterprise systems that process customer transactions:

Customer Interface: Entry point where a customer interacts (e.g., website or app).

Credit Card System: Processes payment details.

Financial Accounting: Handles accounting entries for the transaction.

Purchase System: Records customer orders.

Inventory Management: Checks item availability and updates stock levels.

When inventory is low or an order is placed, the replenishment system kicks in → communicates with the distributor's systems → sends data to A → moves down to B for manufacturer actions.

Bottom Layer (Manufacturer & External Systems)

This layer handles manufacturer systems and external partner systems:

Sales System Manufacturer: Handles sales operations at the manufacturer level.

Sales Database Manufacturer: Stores sales data.

Guarantee-Warranty System: Manages product warranty and customer support services.

Trading Partner/Supplier Database: Handles communication with suppliers and trading partners.

Diagram Concepts :

Web Services Enable Automation: All systems talk to each other automatically without manual work.

B2B Integration: Distributors, manufacturers, and suppliers are connected.

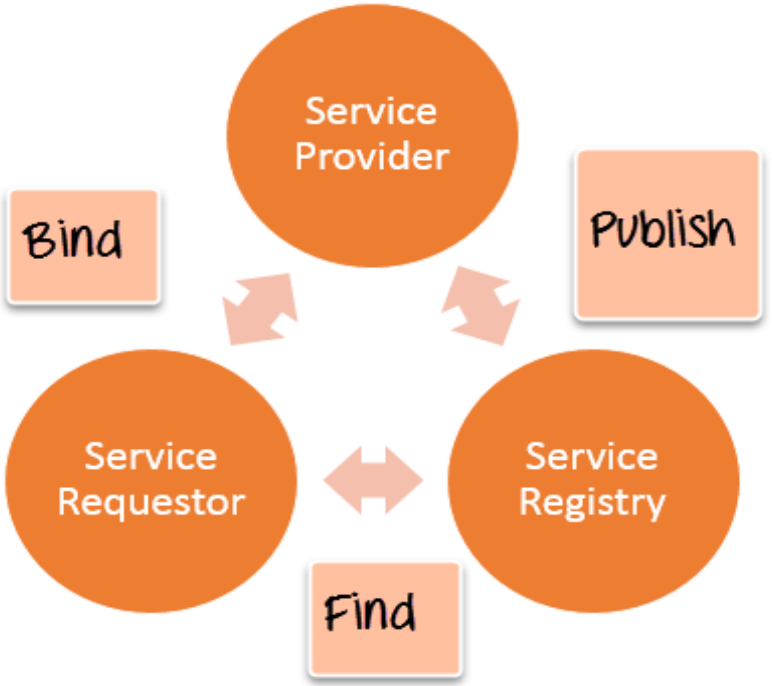
Real-Time Updates: Data like inventory levels, sales info, and warranties update instantly.

Layered Architecture: Shows how data flows from customer to supplier through intermediate layers.

Web Service Architecture

- **Requestor -** A requestor is the client application that needs to contact a web service. The client application can be a .Net, Java, or any other language based application which looks for some sort of functionality via a web service.

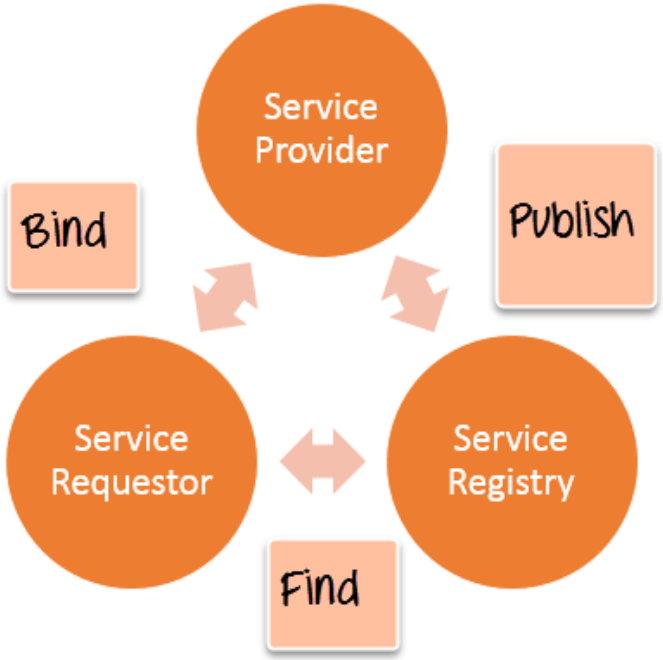
- **Registry (Broker) -** The broker is the application which provides access to the UDDI. The UDDI, as discussed in the earlier topic enables the client application to locate the web service.



- **Publish -** A provider informs the broker (service registry) about the existence of the web service by using the broker's publish interface to make the service accessible to clients

- **Find -** The requestor consults the broker to locate a published web service

- **Bind -** With the information it gained from the broker(service registry) about the web service, the requestor is able to bind, or invoke, the web service.



Advantages of Web Services

We already understand why web services came about in the first place, which was to provide a platform which could allow different applications to talk to each other. Some other advantages of why it is important to use web services.

- **Exposing Business Functionality on the network -** A web service is a unit of managed code that provides some sort of functionality to client applications or end users.
- **Simplicity -** Web services use standardized technologies such as WSDL, XML and HTTP.
- **Security-** Web services use authentication, authorization, encryption and other security measures to protect the data being transmitted between clients and servers.

Advantages of Web Services

- **Interoperability amongst applications** - Web services allow various applications to talk to each other and share data and services among themselves. All types of applications can talk to each other. So instead of writing specific code which can only be understood by specific applications, you can now write generic code that can be understood by all applications

- **A Standardized Protocol which everybody understands** - Web services use standardized industry protocol for the communication. All the four layers (Service Transport, XML Messaging, Service Description, and Service Discovery layers) uses well-defined protocols in the web services protocol stack.

- **Reduction in cost of communication** - Web services use SOAP over HTTP protocol, so you can use your existing low-cost internet for implementing web services.

Characteristics of Web services

Web services have the following special behavioural characteristics:

- **They are XML-Based -** Web Services uses XML to represent the data at the representation and data transportation layers. Using XML eliminates any networking, operating system, or platform sort of dependency since XML is the common language understood by all.

- **Loosely Coupled –** Loosely coupled means that the client and the web service are not bound to each other, which means that even if the web service changes over time, it should not change the way the client calls the web service. Adopting a loosely coupled architecture tends to make software systems more manageable and allows simpler integration between different systems.

- **Supports Document Exchange -** One of the key benefits of XML is its generic way of representing not only data but also complex documents. These documents can be as simple as representing a current address, or they can be as complex as representing an entire book.

Characteristics of Web services

- **Synchronous or Asynchronous functionality**- Synchronicity refers to the binding of the client to the execution of the service. In synchronous operations, the client will actually wait for the web service to complete an operation. An example of this is probably a scenario wherein a database read and write operation are being performed. If data is read from one database and subsequently written to another, then the operations have to be done in a sequential manner.

- Asynchronous operations allow a client to invoke a service and then execute other functions in parallel. This is one of the common and probably the most preferred techniques for ensuring that other services are not stopped when a particular operation is being carried out.

- **Ability to support Remote Procedure Calls (RPCs)** - Web services enable clients to invoke procedures, functions, and methods on remote objects using an XML-based protocol. Remote procedures expose input and output parameters that a web service must support.

Supports Document Exchange

One of the key advantages of XML is its generic way of representing not only data, but also complex documents. These documents can be as simple as representing a current address, or they can be as complex as representing an entire book or Request for Quotation (RFQ).

Web services support the transparent exchange of documents to facilitate business integration.

Applications of Web Services

Web services are composed of various languages and can still communicate with one another. A client sends a request to a web service, which then responds with an XML response.
It also allows developers or users to communicate with each other using various markup languages like HTML, XML, etc.
A web service that serves Java, PHP, or .Net apps can communicate with each other over a network. In terms of corporate operations, web services serve a variety of advantages.

Examples of Webservice

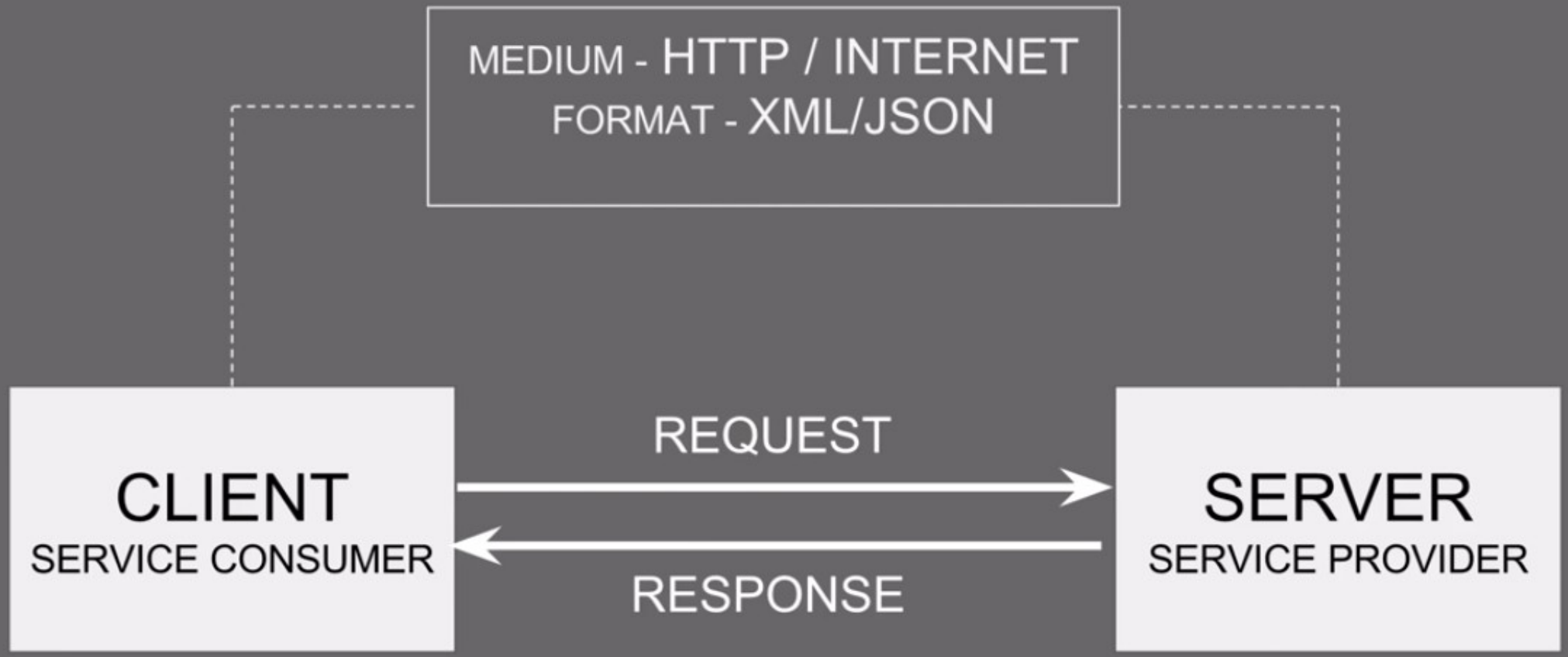
Google search : When computer users access the Google search engine, their search terms or query is passed to the Google search engine. The engine performs the search and returns search results to the user's web browser. The user then selects from the available search results or enters a new search query to refine and continue their browsing.

Google Maps: When using Google Maps, users provide their location and desired destination. The service calculates the most appropriate route and offers directions and a detailed map delivered to the user's web browser. Additional services can be invoked to add supplemental data, such as weather and traffic conditions.

Amazon: This large, well-known shopping site lets users detail the product type, model, manufacturer or other details for which they're looking. Amazon displays the available product options and lets users place orders and pay for using Amazon's web services.

WEB SERVICES

Services available over the web



WEB SERVICES

ervices available

SOAP
Simple Object Access Protocol

Medium : HTTP (POST)
Format : XML

EDIUM - HTTP
FORMAT - XI

REST
REpresentational State Transfer

Medium : HTTP (POST,GET,PUT, DELETE,...)
Format : XML/JSON/TEXT...

CLIENT
SERVICE CONSUMER

REQUEST

RESPONSE

SERVER
SERVICE PROVIDER

Applications of Web Services

Web Service Application in Supply Chain Management (SCM)

They enable seamless communication and integration between various entities such as

- Suppliers
- Manufacturers
- Distributors
- Retailers
- Customers.

standard internet protocols (like HTTP, XML, SOAP, and REST), web services support real-time data sharing and automate key SCM processes.
supply chain management, the process of planning, organising, and managing the flow of goods and services from suppliers to customers.

Supplier Integration:

Web services allow organizations to directly connect their ERP systems with suppliers.

Purchase orders, invoices, inventory status, and delivery schedules can be exchanged automatically.

This reduces manual effort, speeds up procurement cycles, and minimizes errors.

Inventory Management:

Real-time updates of stock levels are shared across locations.

Web services provide integration between warehouse management systems (WMS) and retail systems.

Helps prevent overstocking or stockouts through automated reordering.

Order Processing and Fulfillment:

Web services support order tracking across the entire supply chain.

Customers, retailers, and logistics providers can access order status updates in real time.

Improves customer satisfaction through timely deliveries and transparency.

Logistics and Transportation Coordination:

Web services connect shipping partners with supply chain systems.

Shipment scheduling, route optimization, and delivery confirmation are handled via integrated platforms.

Enables dynamic and efficient transportation planning.

Demand Forecasting and Planning:

Web services integrate data from sales, inventory, and market trends to aid in demand planning.

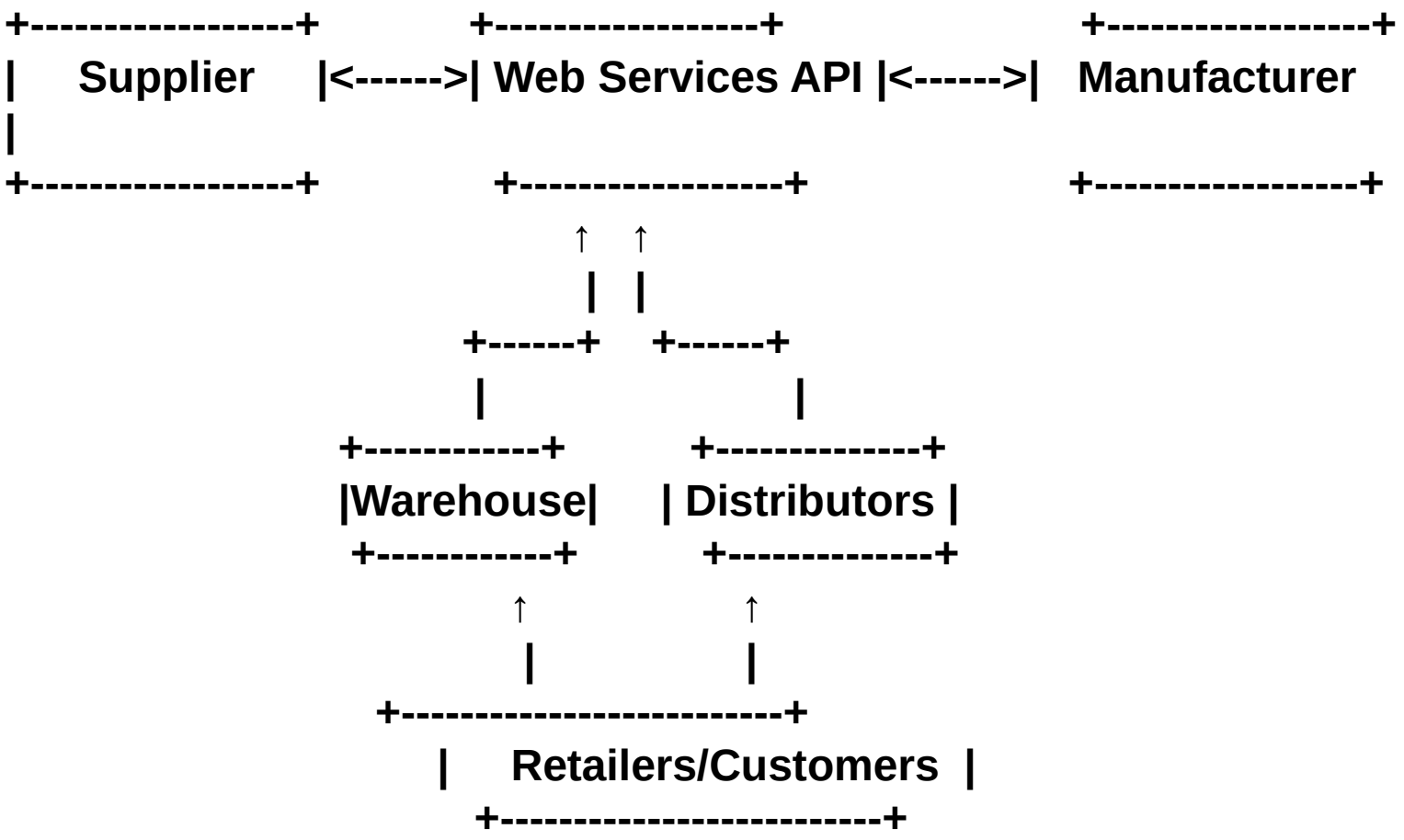
This helps optimize production schedules and reduce waste.

Returns Management:

Automates the return and replacement processes between customer service, logistics, and inventory systems.

Improves reverse logistics efficiency.

Web Services in SCM (Supply Chain Management)



The Web Services API acts as a central communication hub.

Web Services in Education

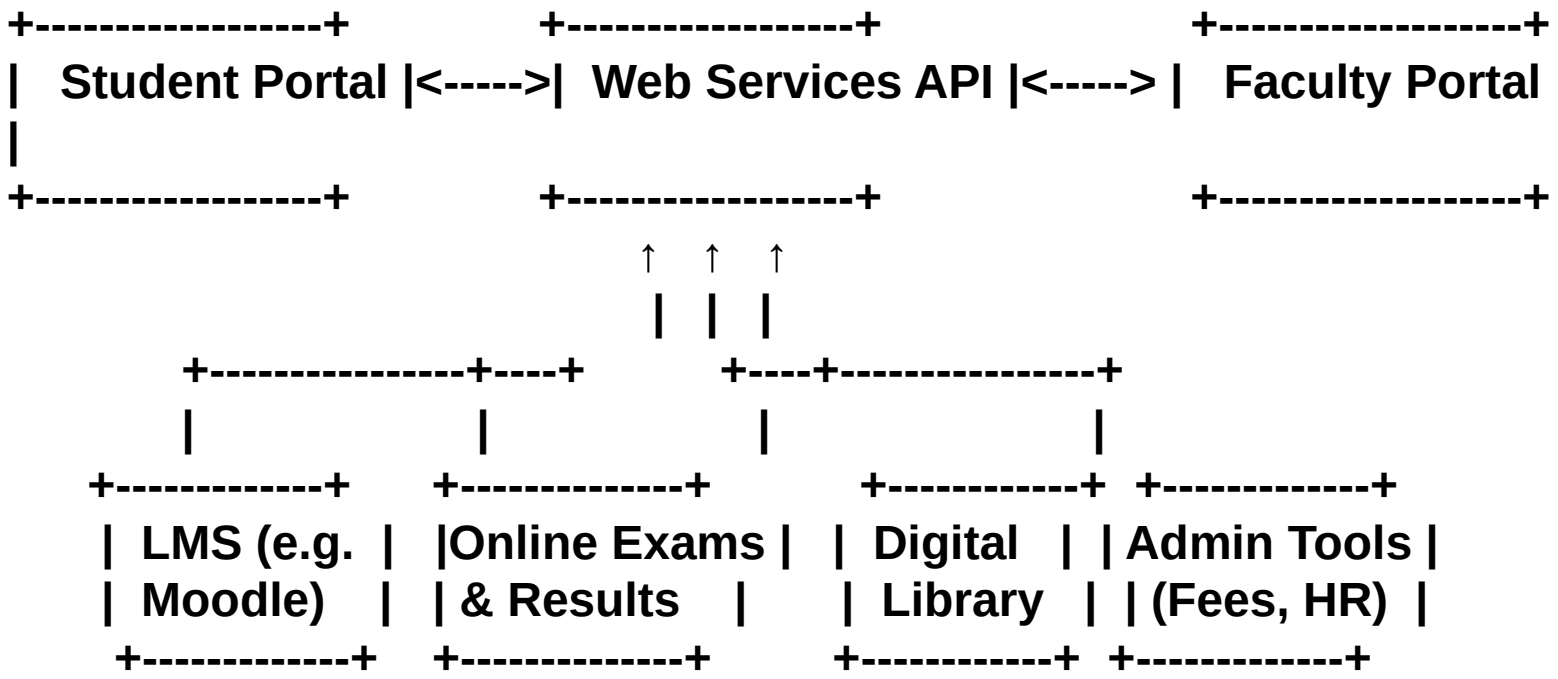
Web services enable smooth data sharing and integration between educational platforms like LMS, examination systems, and administrative tools using web protocols (SOAP, REST, XML, JSON).

Key Applications:

- LMS Integration: Access to course content, grades, attendance.
- Student Info System (SIS): Centralized student records and reporting.
- Online Exams: Automated exams, grading, and result sharing.
- Digital Libraries: Easy access to e-books, journals, and research.
- Virtual Classrooms: Integration of video platforms (Zoom, Meet).
- Administration: Automates fees, schedules, notifications.
- Analytics: Tracks performance and generates reports.

Benefits:

- Real-time data exchange
- Streamlined operations
- Personalized learning
- Better student-teacher-parent coordination
- Supports online & hybrid learning



- Central Web Services API connects all platforms.
- Enables two-way communication between students, teachers, and systems.

Key Applications (Summary):

- **Supplier Integration:** Automates order and invoice sharing.
- **Inventory Management:** Real-time stock updates across systems.
- **Order Fulfillment:** Tracks orders and updates status live.
- **Logistics Coordination:** Connects with transport partners.
- **Demand Forecasting:** Shares data for better planning.
- **Returns Management:** Simplifies reverse logistics.
-