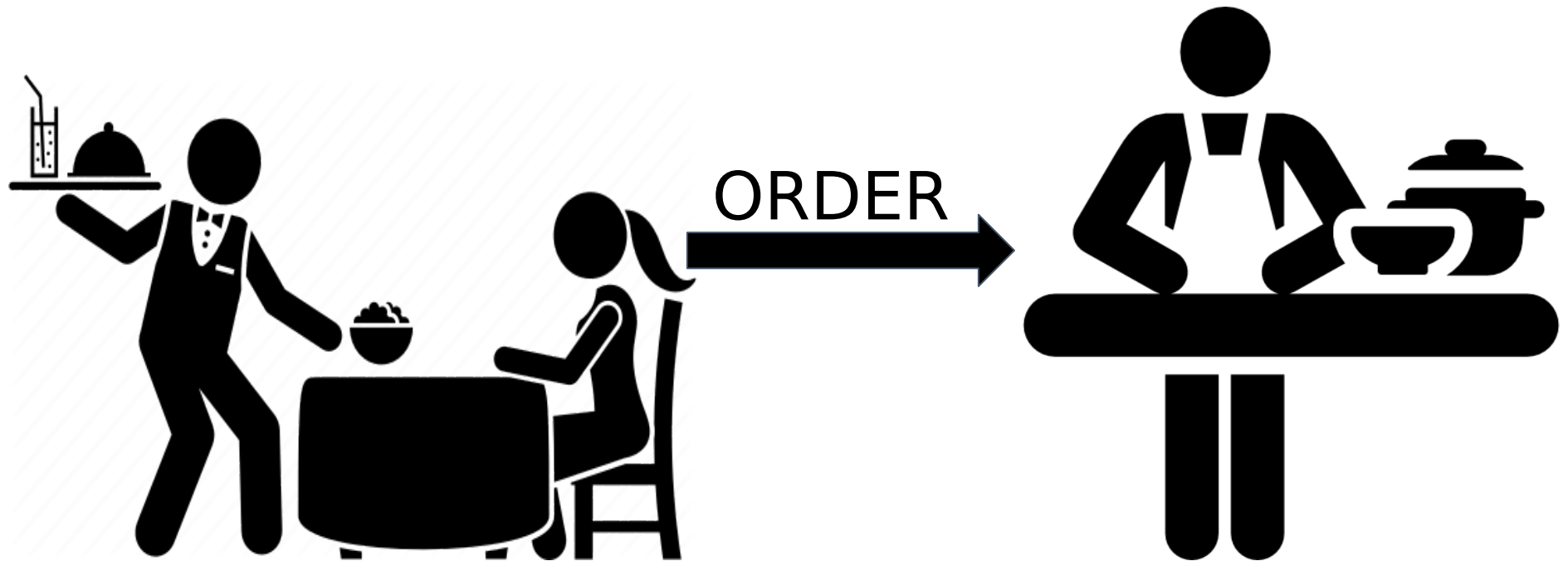An Introduction to.....

Web Services

Module – 1
Prof. Nirav Suthar

FCAIT-UG

Example of Web Service

Example where a customer can directly place order.

ORDER

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

Why do you need a Web Service?

- 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.
- 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.

So ….. 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

- Is available over the Internet or private (intranet) networks

- Uses a standardized XML messaging system

- Is not tied to any one operating system or programming language

- Is self-describing via a common XML grammar

- Is discoverable via a simple find mechanism

So ….. What is Web-service?

We can even say

● 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.
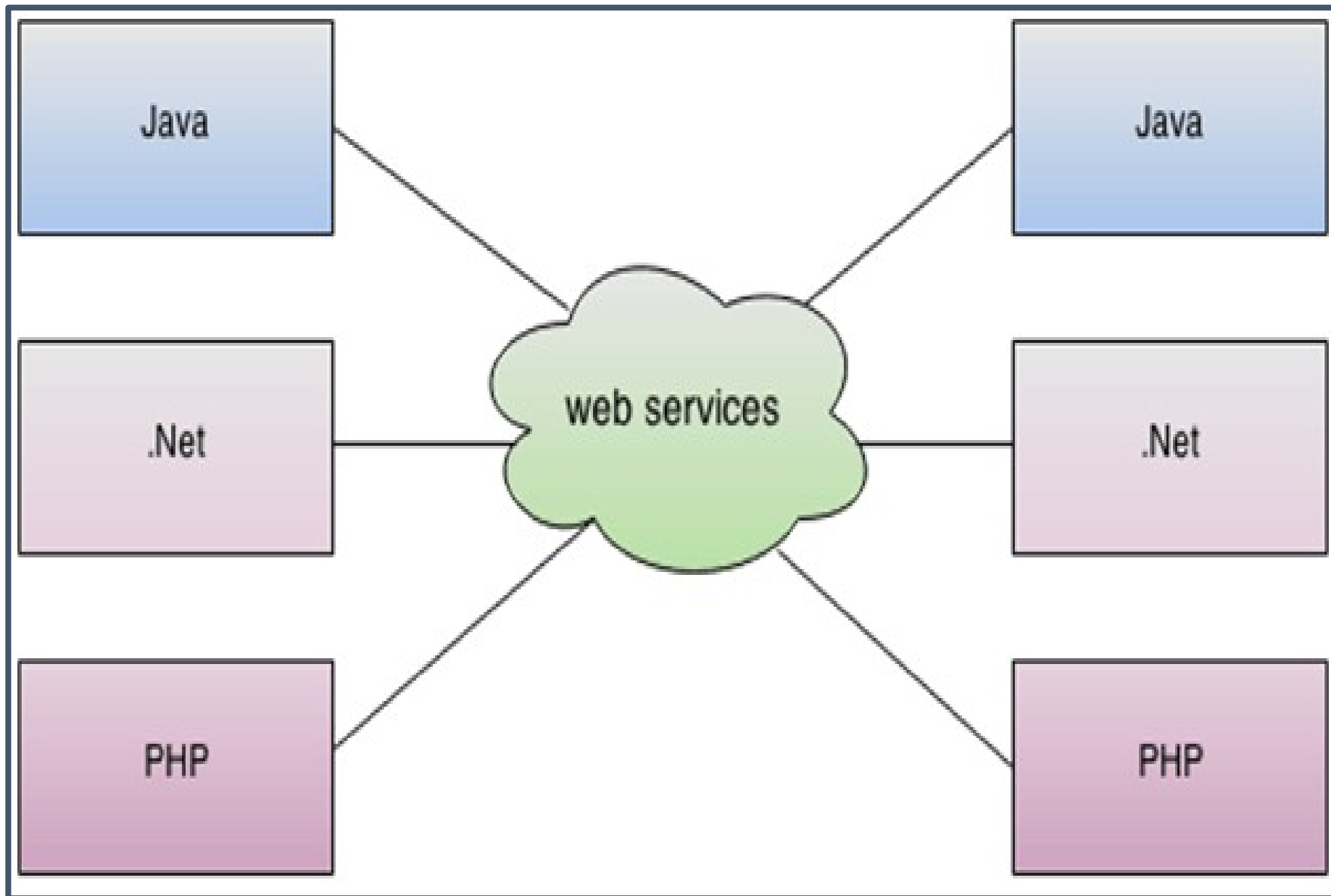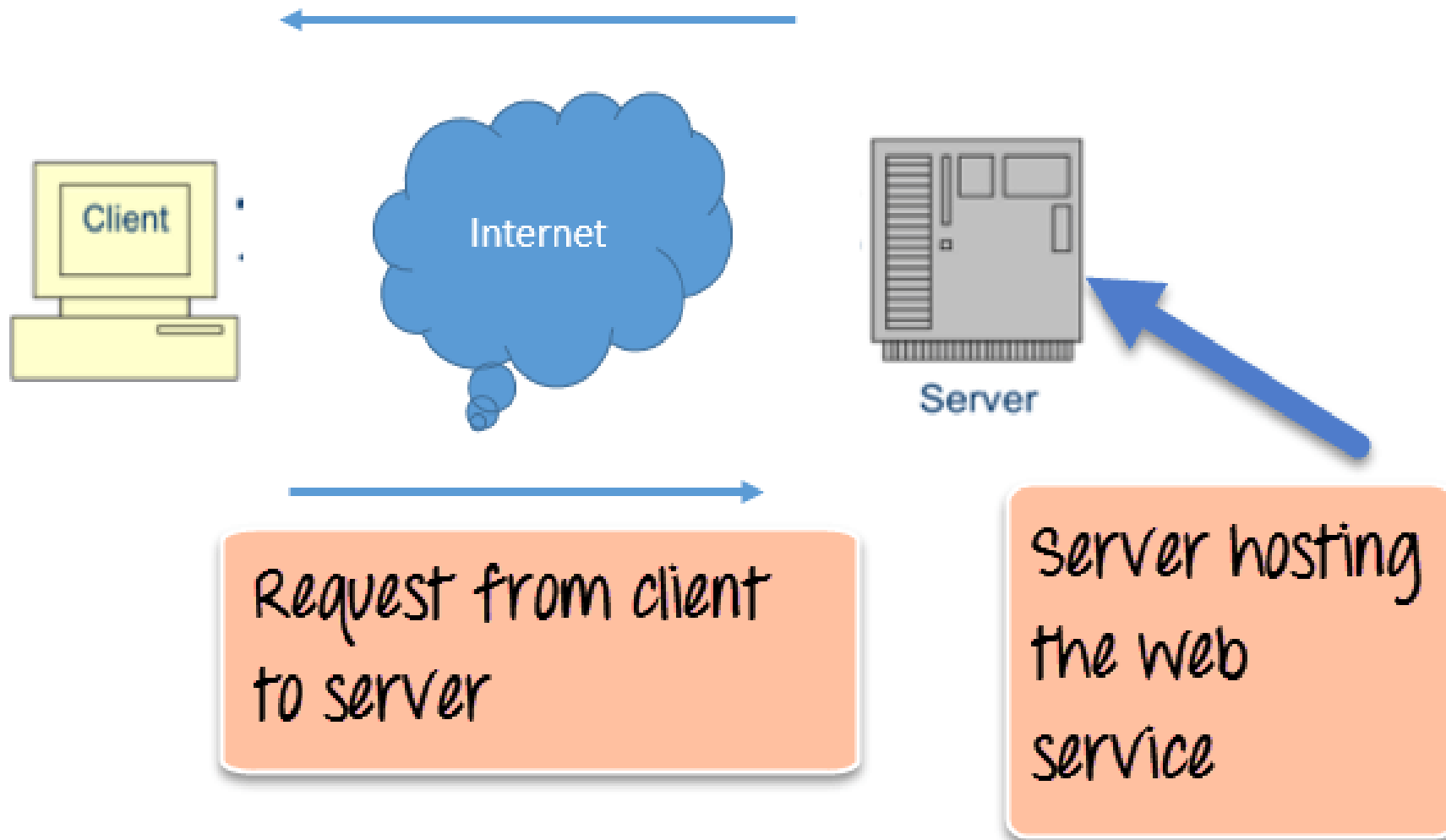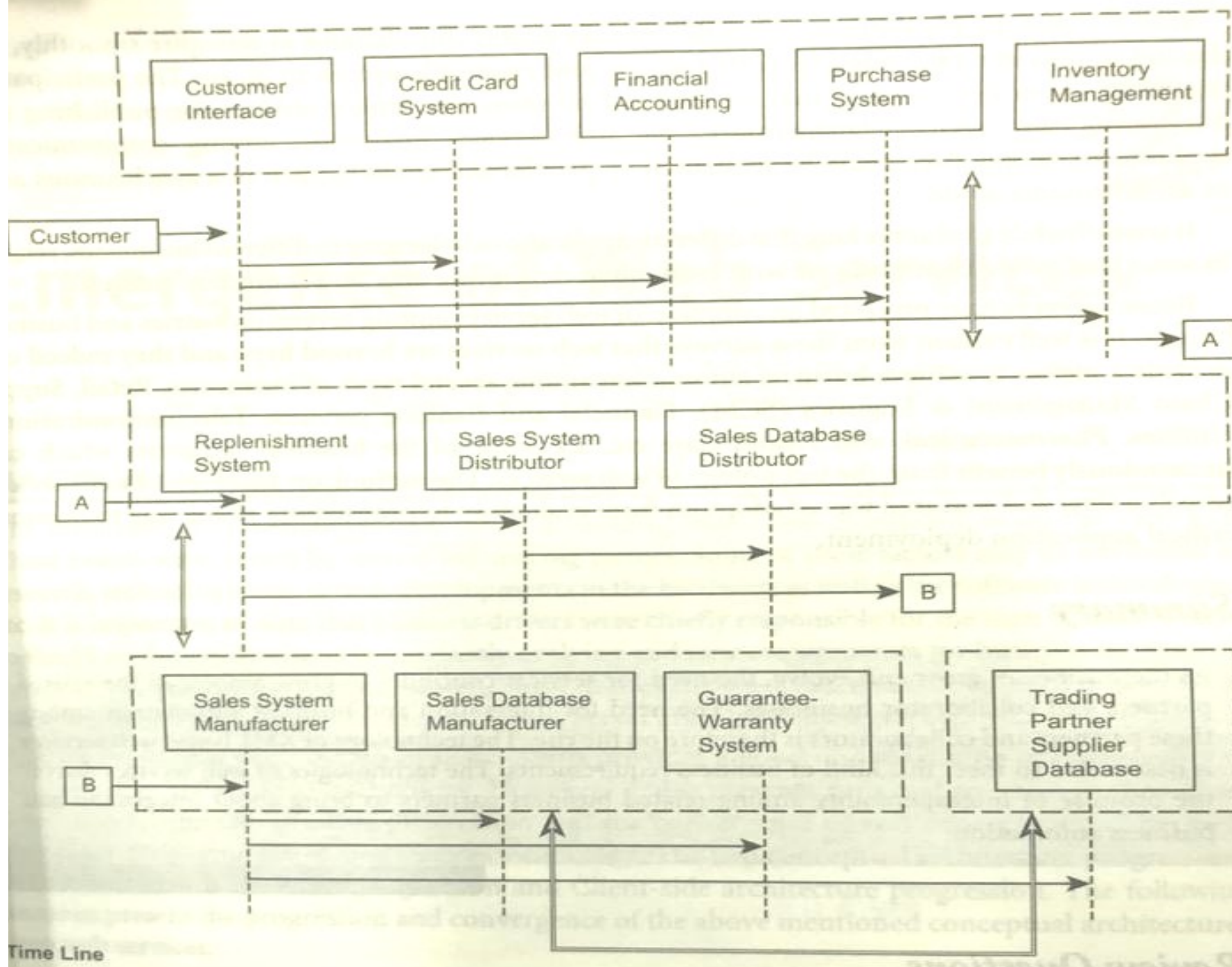
WEB SERVICE APPLICATION OPPORTUNITIES

- Web services heralded another significant milestone in the history of IT.
- Earlier Internet catered mostly B2C category of user.
- Web services enable B2B interaction.
- Possibility of business automation is most attractive aspect of Web Services.

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 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 custom 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.

Web Service Architecture

Every framework needs some sort of architecture to make sure the entire framework works as desired. Similarly, in web services, there is an architecture which consists of three distinct roles as given below

- **Provider** - The provider creates the web service and makes it available to client application who want to use it.
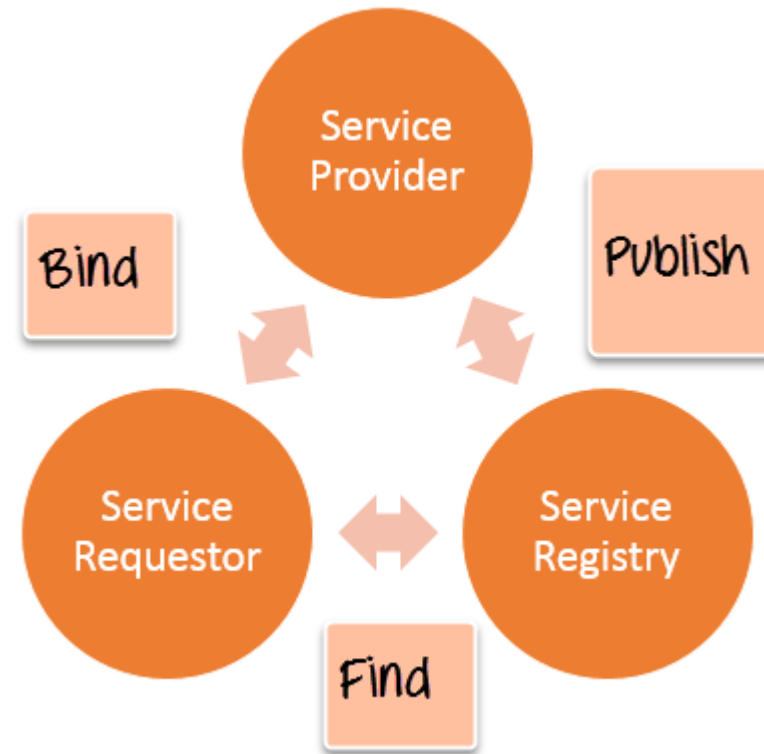
Web Service Architecture

- **Requestor -** A requestor is nothing but 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.

- **Broker -** The broker is nothing but 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.
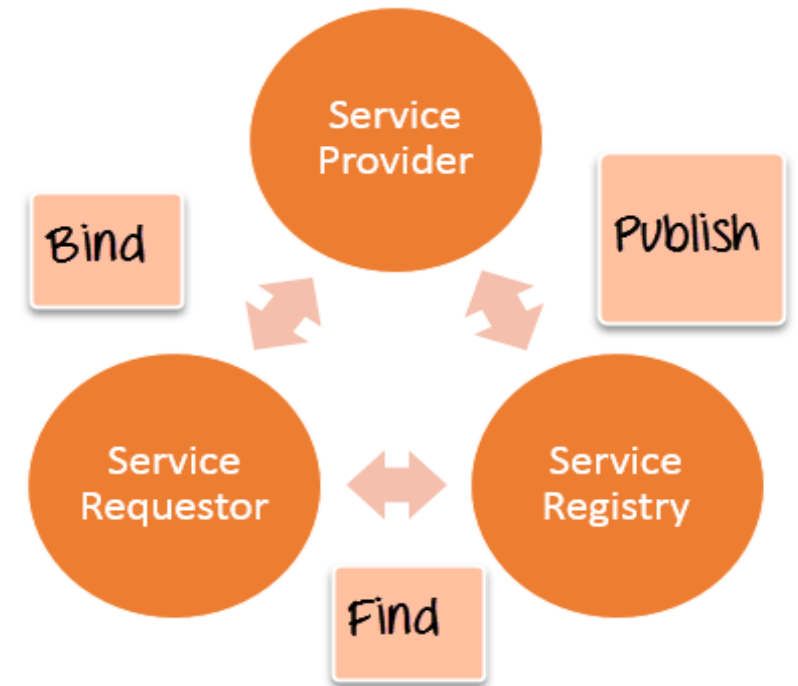
Web Service Architecture

- **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.
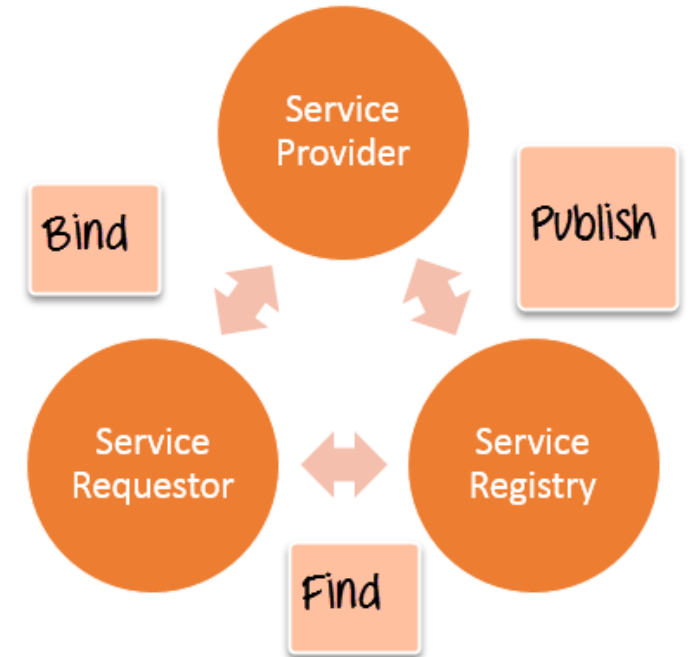
Advantages of Web Services

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. But let's look at 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.

XML RPC

- XML-RPC is a simple protocol that uses XML messages to perform RPCs.
- Requests are encoded in XML and sent via HTTP POST.
- XML responses are embedded in the body of the HTTP response.
- XML-RPC is platform-independent.
- XML-RPC allows diverse applications to communicate.
- A Java client can speak XML-RPC to a Perl server.
- XML-RPC is the easiest way to get started with web services.
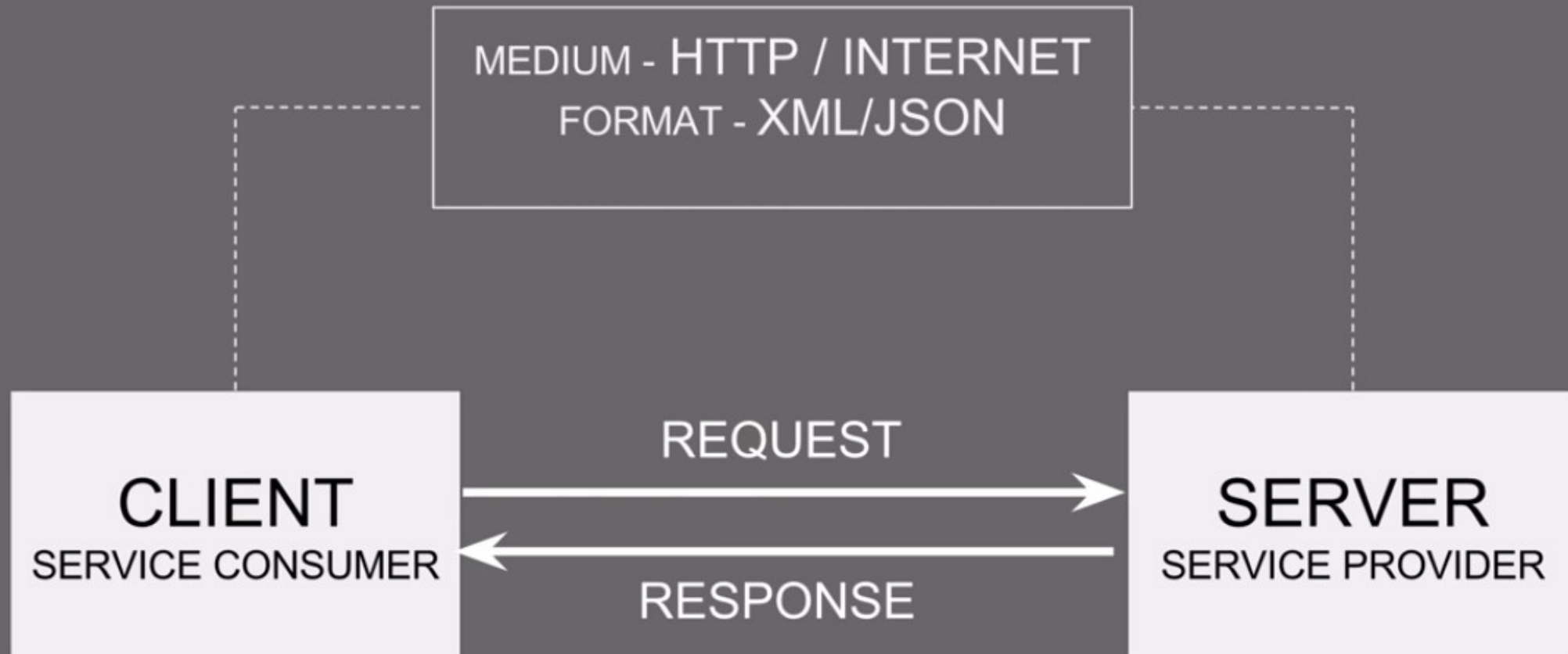
Examples of Webservice

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.