

**Experiment No. 9**

**Title: Study Experiment on Web Services**



Batch: B-4

Roll No.: 16010422234

Name: Chandana Ramesh Galgali

**Experiment No.: 9****Aim: Study Experiment on Web Services**

---

**Theory:****Web Services:**

Web services are software mechanisms that communicate using pervasive, standards-based Web technologies including HTTP and XML-based messaging and this structure are based on a collection of standards and protocols that allow us to make handling requests to remote systems by delivering a standard, nonproprietary language and using conventional transport protocols such as HTTP and SMTP. The efficient e-business perception calls for a smooth integration of business processes, applications, and Web services over the Internet. Web service technology enables e-business and e-commerce to become a reality. It has become a competitive tool for companies by reducing cost through fast, efficient, and reliable services to clients, dealers, and partners over the Internet. It permits more efficient business processes via the Web and improves business chances for companies, Web services are planned to be accessed by other applications and differ in complication from primary activities, such as examine a banking account balance online, to complicated processes running CRM (customer relationship management) or enterprise resource planning (ERP) systems because these are based on open standards such as HTTP and XML-based protocols including SOAP and WSDL. Web services are powered by XML and three other core technologies: WSDL, SOAP, and UDDI. In a Web service model, a service supplier proposes Web services which deliver tasks or business operations which can be arranged over the Internet, in the hope that they will be invoked by partners or customers; a Web service requester defines requirements to trace service providers. Publishing, binding, and discovering Web services are three key tasks in the model. Discovery is the process of finding Web services provider locations which satisfy specific requirements. Web services are useless if they cannot be discovered. So, discovery is the most important task in the Web service model. The Web service model in Figure shows the interaction between a service requester, service providers, and a service discovery system.

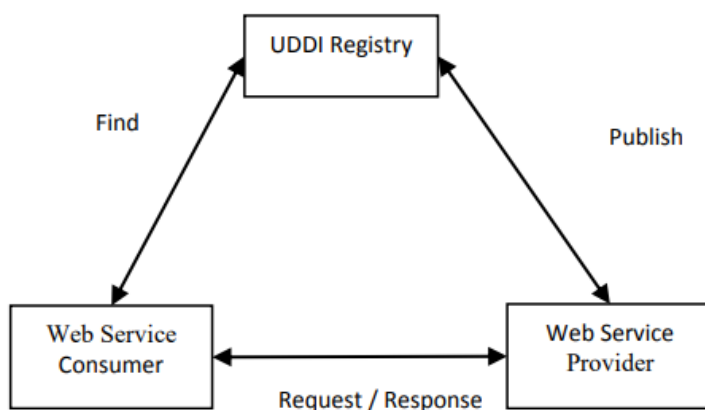


Figure: Web Services Model

1. The service provider's proposal Web services which deliver functions or business operations. They are formed by companies or societies. In order to be invoked, the Web services must be defined. This will facilitate discovery and arrangement. WSDL or service profile of semantic Web service is used to carry out this task.
2. The Web service requester defines requirements in order to locate service providers. Service requesters usually contain a description of the Web service, though it is not a Web service which can run on the Internet. The requirements are typically defined by WSDL, service template or service profile.
3. The Web service discovery or service registry is a broker that provides a registry and examines tasks. The service providers advertise their service info in the discovery system. This info will be kept in the registry and will be searched once there is a demand from the service requester. UDDI is used as a registry typical for Web service.

The above three mechanisms interact with each other via publishing, discovery, and binding operations. These operations are elaborated upon as follows:

1. **Publish:** the Web service providers publish their service information through the discovery system for requesters to discover. Through the publishing operation, the Web service provider stores the service description in the discovery system.
2. **Discovery:** the Web service requesters repossess service providers from the service archive. Based on service explanations, which describe the requests of the Web service clients, the discovery system will output a list of Web service suppliers which satisfy the requirements.
3. **Bind:** After discovering, the discovery system provides some Web service providers. The Web service requester invokes these Web service providers. The binding occurs at runtime. The Web service requesters and Web service providers will communicate via SOAP protocol which is an XML based protocol for Web service exchange information.

---

**Activity:** Case study of Google web services.

---

### Case Study: Google Web Services

Google offers a robust suite of web services designed to support various aspects of digital and cloud computing needs. Known as Google Cloud Platform (GCP), it provides scalable and

reliable infrastructure services to businesses and developers around the world. This case study will explore some of the key components of GCP and demonstrate their impact on modern computing solutions.

## **Overview of Google Cloud Platform**

Google Cloud Platform is a collection of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube. It offers services in computing, storage, networking, big data, machine learning, and the internet of things (IoT), as well as cloud management, security, and developer tools.

## **Key Services of GCP**

1. **Google Compute Engine (GCE):** This is an Infrastructure as a Service (IaaS) that provides users with virtual machine instances for workload hosting. It is highly customizable and scalable, offering various machine types depending on the user's requirements.
2. **Google App Engine (GAE):** This Platform as a Service (PaaS) allows developers to deploy applications without managing the underlying infrastructure. It supports popular programming languages, such as Python, Java, Ruby, and Go.
3. **Google Kubernetes Engine (GKE):** An orchestration service for Docker containers that handles scheduling and scaling applications based on containerized technology across a cluster of machines.
4. **Google Cloud Storage (GCS):** A service that offers object storage for live or archived data. It is highly durable and available, providing strong consistency, rich security features, and easy integration with other Google services.

## **Use Case: Spotify's Migration to Google Cloud**

Spotify, a leader in the music streaming industry, provides a practical example of leveraging Google Web Services. Faced with scaling issues due to its rapidly growing user base and the vast amount of data it handles, Spotify decided to migrate its services from its own data centers to Google Cloud.

## **Challenges Faced:**

- Scalability with growing data volumes and user base.
- Need for improved data analytics capabilities.
- Requirement for a more efficient way to manage resources and operational costs.

### **Solutions Implemented:**

- Spotify utilized Google BigQuery to handle its extensive data analytics needs, allowing for real-time insights into user behavior and song popularity.
- Google Compute Engine allowed Spotify to scale its computing resources dynamically, adjusting to changes in demand without the need for upfront capital investment.
- By adopting Google's data services, Spotify improved its operational efficiency and optimized its service delivery to millions of users worldwide.

### **Conclusion:**

Google's web services have demonstrated significant benefits in terms of scalability, reliability, and efficiency for a wide range of applications and industries. As exemplified by Spotify, transitioning to GCP can resolve substantial operational challenges, reduce costs, and enhance service delivery. Google continues to innovate in the space of web services, pushing the boundaries of what businesses can achieve with cloud technologies.

---

### **Conclusion:**

This experiment with web services illustrates the vital role they play in modern computing environments. The case study of Google's web services demonstrates how these technologies facilitate a powerful, scalable, and efficient method for handling diverse computing tasks through the internet. Web services are integral for businesses seeking to leverage the power of the internet for seamless integration across various platforms and services.

---

### **References:**

1. <https://www.geeksforgeeks.org/what-are-web-services/>
  2. [https://www.tutorialspoint.com/webservices/what\\_are\\_web\\_services.htm](https://www.tutorialspoint.com/webservices/what_are_web_services.htm)
-