The Difference between Web Services and Micro Services

Micro Services and Web Services are two different concepts of Application Development Architecture, which can be differentiated from its layered architecture and development style. This blog gives more details about these concepts and the difference between Web Services and Micro Services.

What is Web Service?

Web Service is a way to expose the functionality of an application to other application, without a user interface. It is a service which exposes an API over HTTP.

Web Services allow applications developed in different technologies to communicate with each other through a common format like XML, Jason, etc. Web services are not tied to any one operating system or programming language. For example, an application developed in Java can communicate with the one developed in C#, Android, etc., and vice versa.

Web Service is a connection technology, a way to connect services together into a Service Oriented Architecture (SOA).

What is Micro Service?

Micro Service is independently deployable service modeled around a business domain. It is a method of breaking large software applications into loosely coupled modules, in which each service runs a unique process and communicates through APIs. It can be developed using messaging or event-driven APIs, or using non-HTTP backed RPC mechanisms.

Micro Services are designed to cope with failure and breakdowns of large applications. Since multiple unique services are communicating together, it may happen that a particular service fails, but the overall larger applications remain unaffected by the failure of a single module.

Use-Case Representation

Let us understand these concepts with the help of an example of Online Shopping Center.

In figure-1: The Online Shopping Center Web Application is developed in Monolithic Architecture. In this application, there is one Web Service which communicates with web application and database. So this web service might be performing many functional tasks related to database operations.
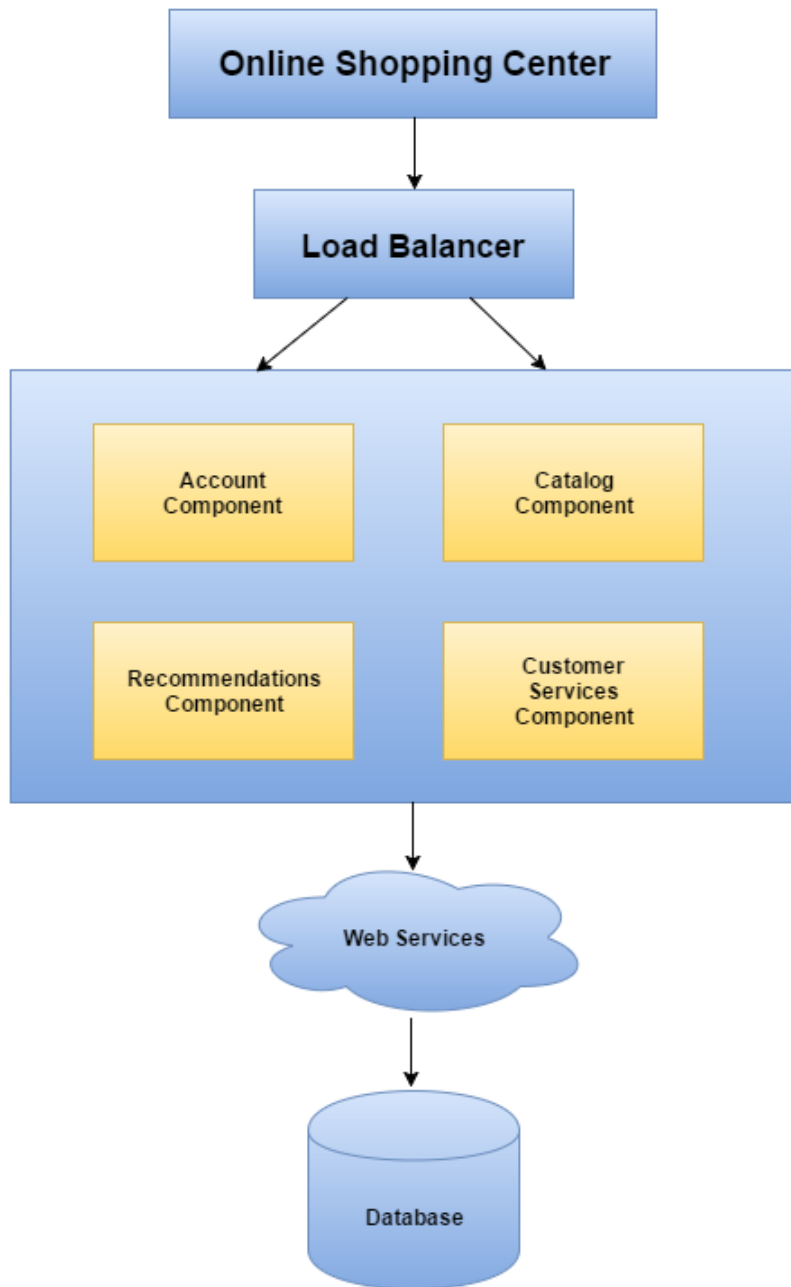
Figure 1: Conventional Approach

In figure-2: The Online Shopping Center Web Application is developed in Micro Services Architecture. All the components of the web application are developed independently, single functional responsible, fine-grained clearly scoped services.
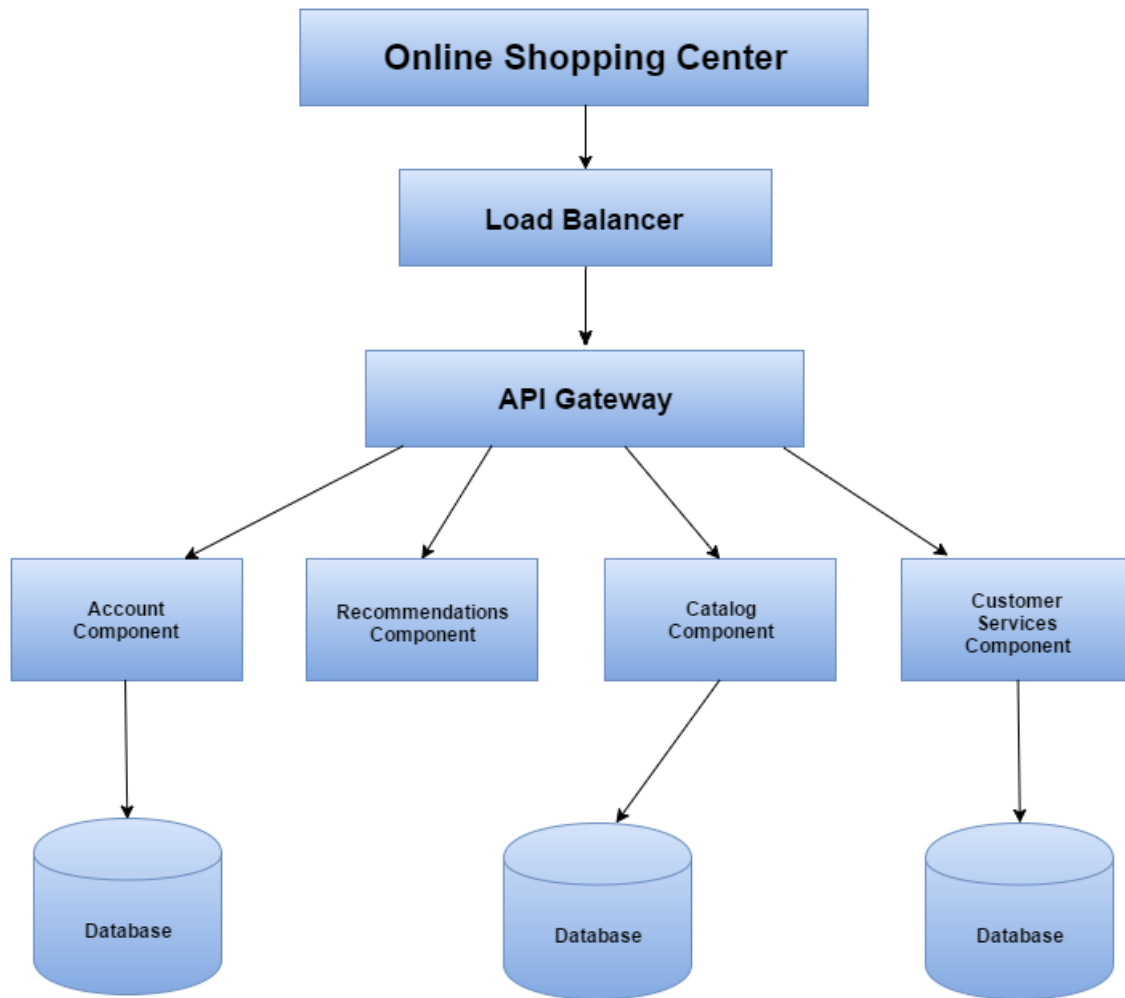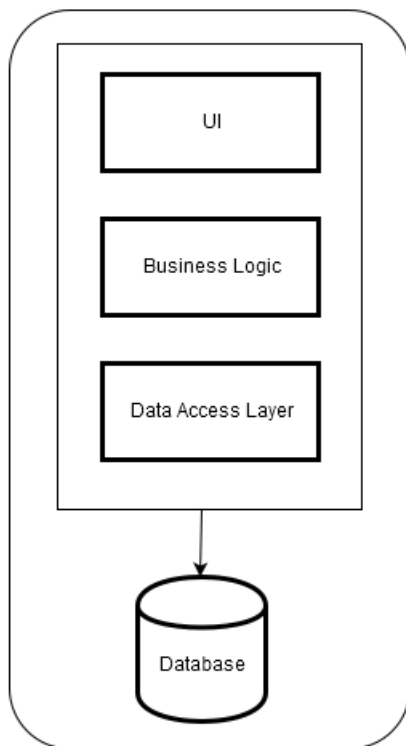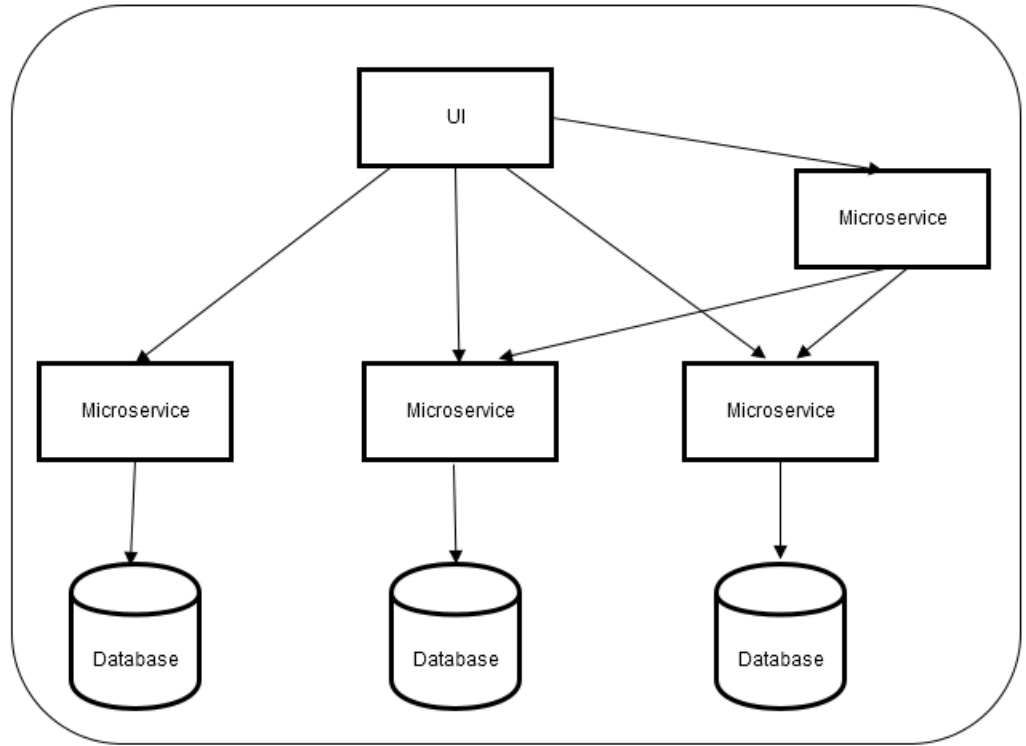
Figure 2: Micro Services Approach

Web Services could be of any size, including large enterprise apps retrofitted with APIs that too many other apps depended on. Although "micro" in Micro Services, the basic concept is that each service performs a single function.

For example, one of the largest eCommerce portal, Amazon, has migrated to Micro Services. They get countless calls from a variety of applications, including applications that manage the Web Services API as well as the portal, which would have been simply impossible to handle for their old, two-tiered architecture.

Applications built as Micro Services can be broken into multiple component services and this service can be a Web Service, which should run unique process and then redeployed independently without compromising the integrity of an application.

Monolithic Architecture   Microservices Architecture

Micro Services style is usually organized around business capabilities and priorities. Unlike a traditional monolithic development approach, where different teams have a specific focus on, say, UIs, databases, technology layers, or server-side logic, Micro Services architecture utilizes cross-functional teams. The responsibilities of each team are to make specific products based on one or more individual services communicating via message bus. It means that when changes are required, there won't necessarily be any reason for the project, as a whole, to take more time or for developers to have to wait for budgetary approval before individual services can be improved. Most development methods focus on projects: a piece of code that has to offer some predefined business value must be handed over to the client, and is then periodically maintained by a team. But in Micro Services, a team owns the product for its lifetime.

In a monolithic service oriented architecture deployment, each small change meant that the entire monolith needed to be rebuilt and this, in turn, meant that re-builds weren't happening as rapidly as they should.

A Web Service is a service offered by an application to another application, communicating with each other via the World Wide Web.
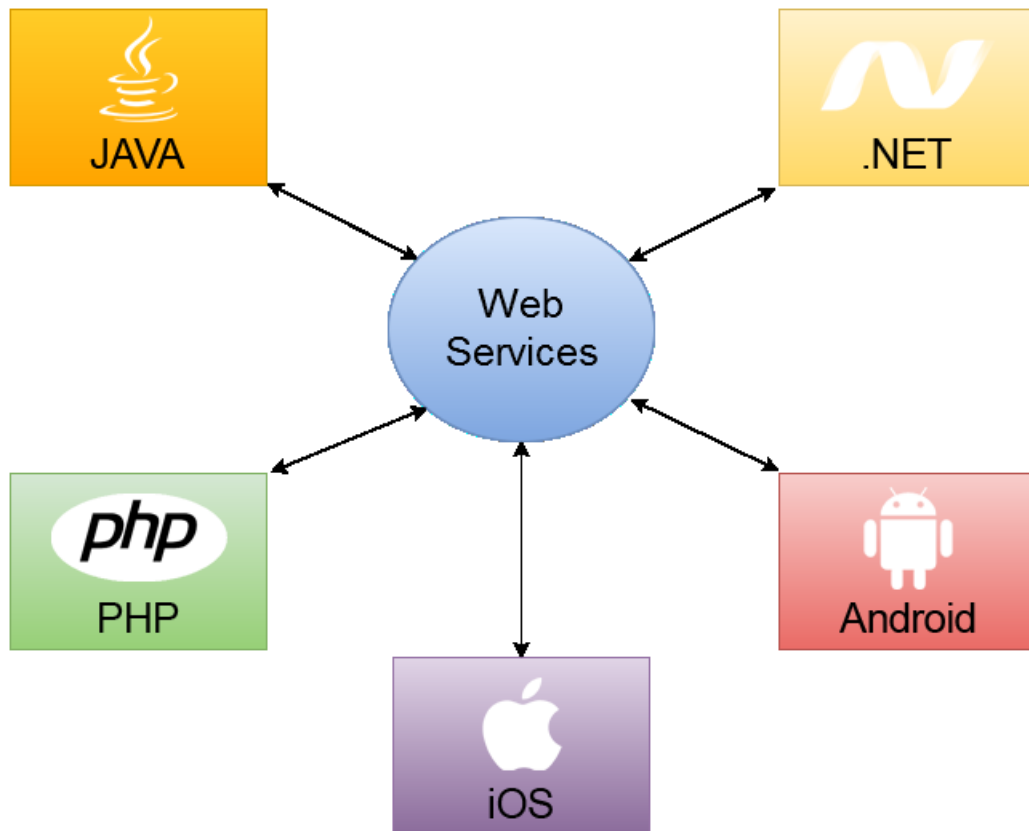
The Web Service typically provides an object-oriented web-based interface to a database server, utilized by another web server, or by a mobile application, that provides a user interface to the end user. Another common application offered to the end user may be a mash-up, where a web server consumes several web services at different machines and compiles the content into one user interface.