[Microservice in .Net Core Example: Microservice Tutorial in C# (webtrainingroom.com)](https://www.webtrainingroom.com/aspnetcore/microservices)

<https://dotnettutorials.net/lesson/microservices-using-asp-net-core/>

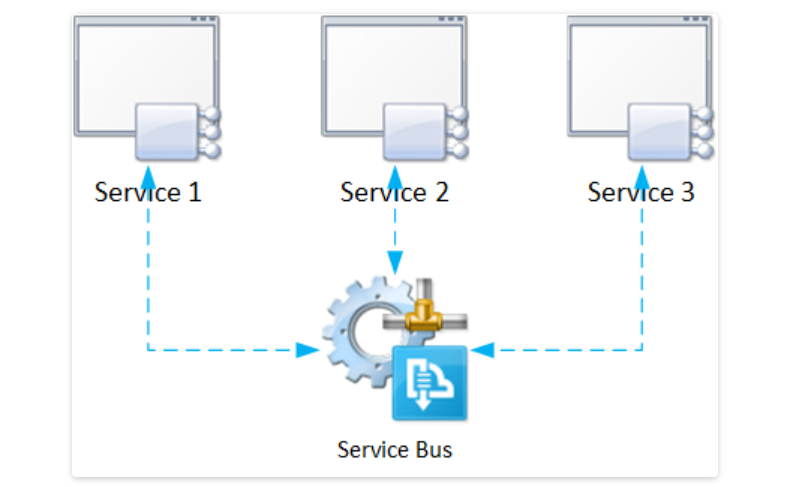
<https://codewithmukesh.com/blog/microservice-architecture-in-aspnet-core/>

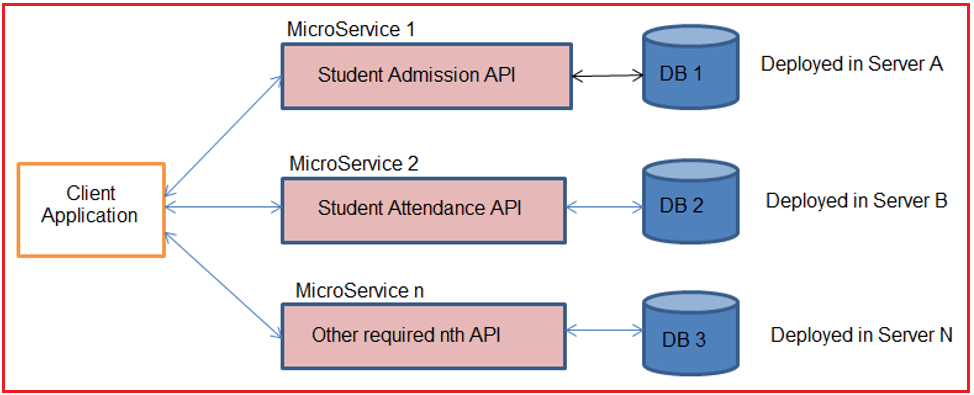
What is Microservice

A collection of small services forms an architecture called microservice architecture. Each and every service has their own business logic and are deployed independently and may run independently. In the microservice architecture the services communicate through the API only.

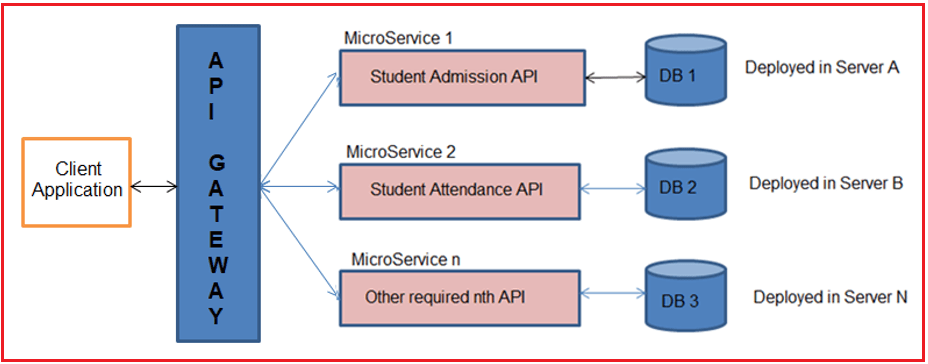
Unlike the older way of application development in Monolithic architecture which causes high maintenance cost, more downtime during upgrades made to existing monolithic architected software is not reliable. So, the Microservices Architecture of developing applications came into the picture.

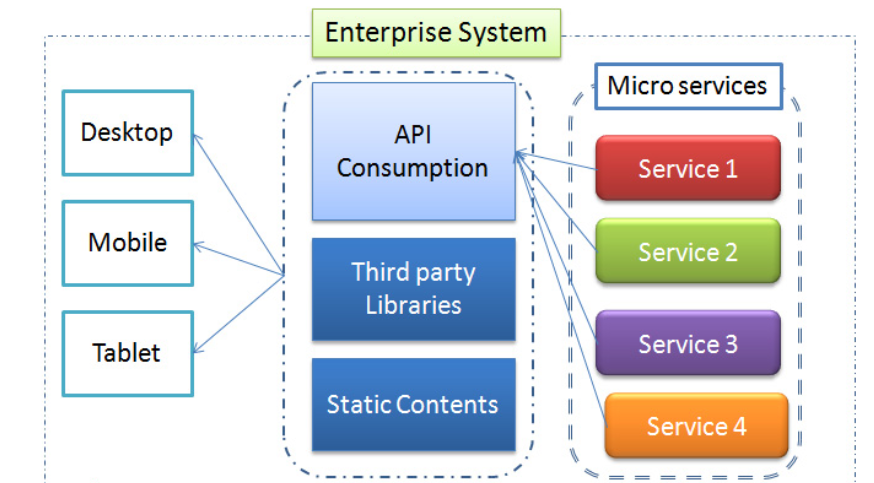
Microservice is to split the application into small services and run them independently.

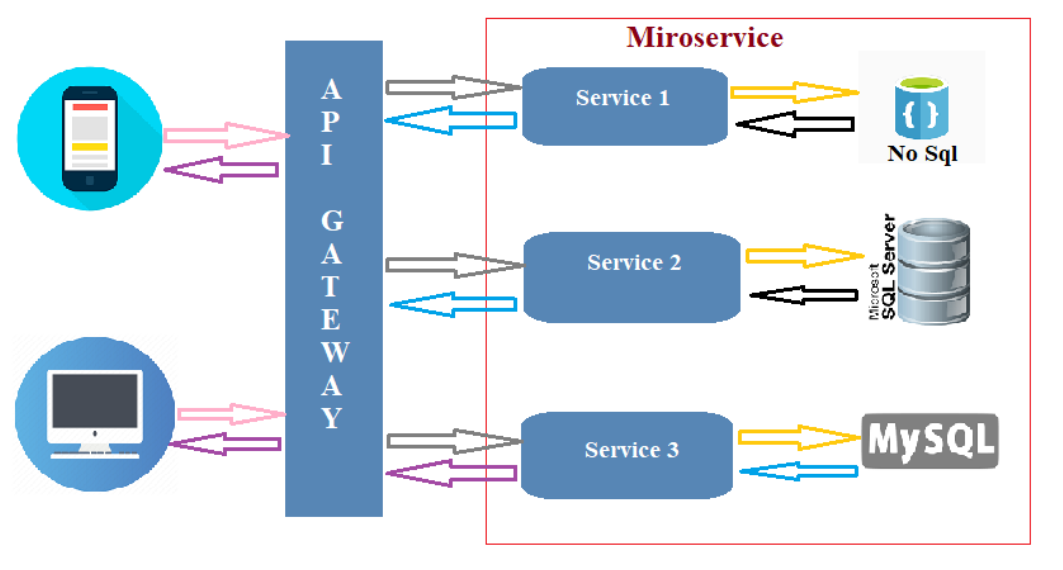




##### **API Gateway (Ocelot Gateway with example)**





****

##### **Features of API Gateway**

**API virtualization:**API Gateways acts as a single point entry for all the microservices configured, and avoids direct availability of microservices to clients, and hides versioning details of microservices.

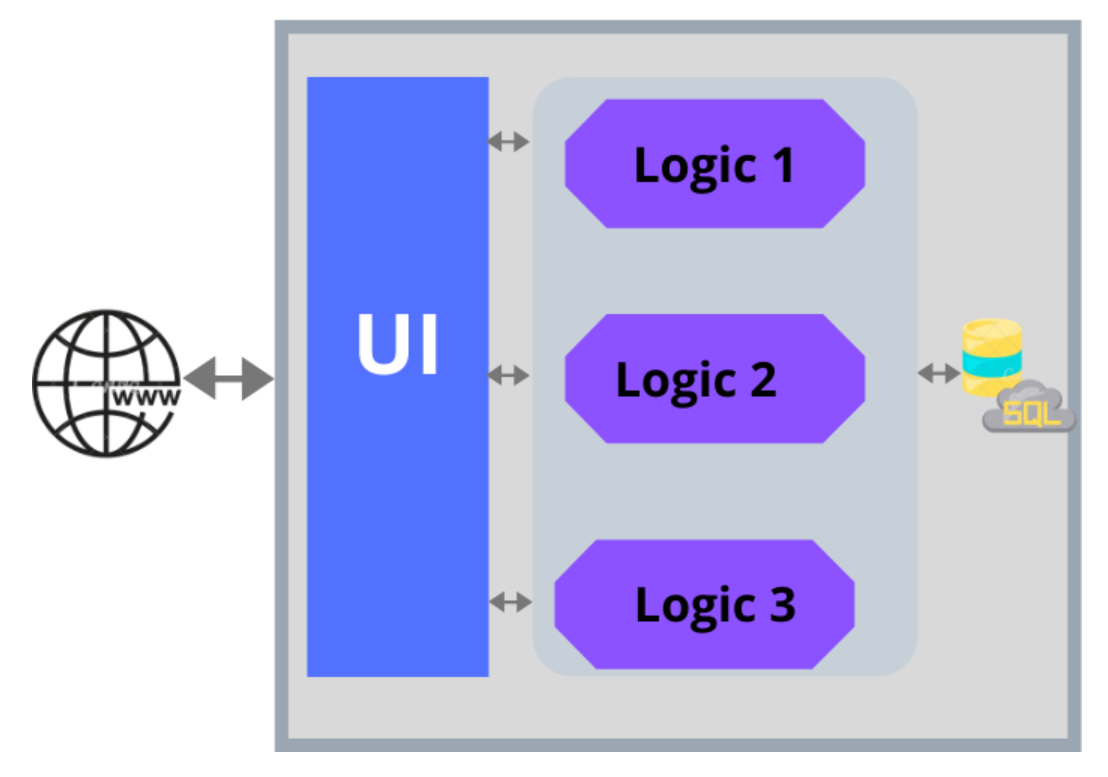
**Serves as an additional layer of security microservices:**API gateways prevent malicious attacks by providing an additional layer of protection from attack vectors and hackers like SQL Injection, XML Parser exploits, and denial-of-service (DoS) attacks, and forged form data submissions.

**Advantages**

1. Faster than monolithic architecture
2. No need to take down the entire application for the deployment
3. Technology Independent (i.e. front end may be developed in angular and backend service may developed in .net or any other technology.)
4. Independent deployment. So there is  no need to take down entire application.
5. If any one of the services fails another one will run without any issue.
6. The service may be written different languages.
7. It enables continuous delivery.
8. Increase application performance.
9. Dockerization and containerization is possible.
10. The service may run in different servers.

**Monolithic Architecture**

Monolithic Architecture is single tier architecture, which means all the backend logics and the frontends are in the single environment.



**Monolithic Architecture Disadvantages**

1. It may slow down the application when the application size increases.
2. Need to take down the entire application during the deployment.
3. The developer may have difficulty in continuous deployment.
4. If a small piece of code changes in the application the developer needs to redeploy the entire application.
5. Monolithic application is tightly coupled.