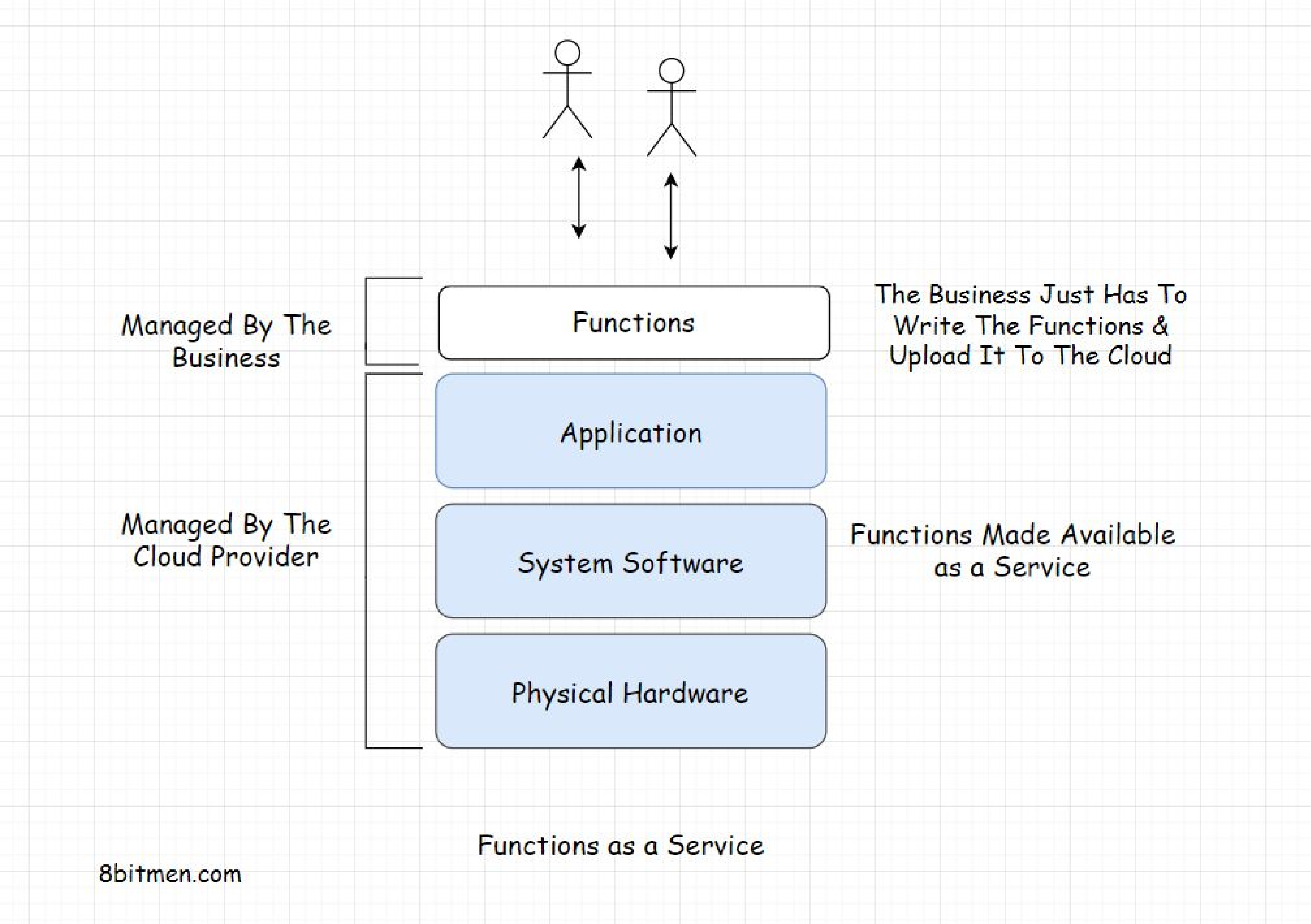
Functions as a Service (FaaS) – Part 1

This lesson covers Function as a Service (FaaS) cloud models in detail.

We'll cover the following

Functions as a Service

FaaS is a cloud service model that enables developers to run their code as *functions* in the cloud. When using *FaaS*, a developer simply has to upload the business logic in functions to the cloud and the cloud platform takes care of the rest of the stuff such as running the code, managing the servers, scaling, provisioning of resources, monitoring, integration with other services and so on.



Managed By The Business
Functions
The Business Just Has To Write The Functions & Upload It To The Cloud
Application
Managed By The Cloud Provider
System Software
Functions Made Available as a Service
Physical Hardware
Functions as a Service
8bitmen.com

The idea behind using *FaaS* is to have the privilege to focus solely on the business logic and let the cloud take care of the rest. This eventually speeds up the development process. Besides allowing you to completely focus on creating value, there is also a cost-related reason associated with the FaaS service model.

When using *FaaS*, a business doesn't have to provision the servers; there is no need to run the servers continually. *FaaS is an event-driven service model*. The functions are only triggered when an event occurs. That event can be an *HTTP* web request, a backend job, a message being delivered from a queue, a user clicking on an element on a web page, and so on.

Therefore, the service gets triggered and the platform spins up a server only when an event occurs. As a result of this event-driven flow, the business only pays for the compute the service consumes during the time that particular event occurred, as opposed to paying for an instance that runs the entire day.

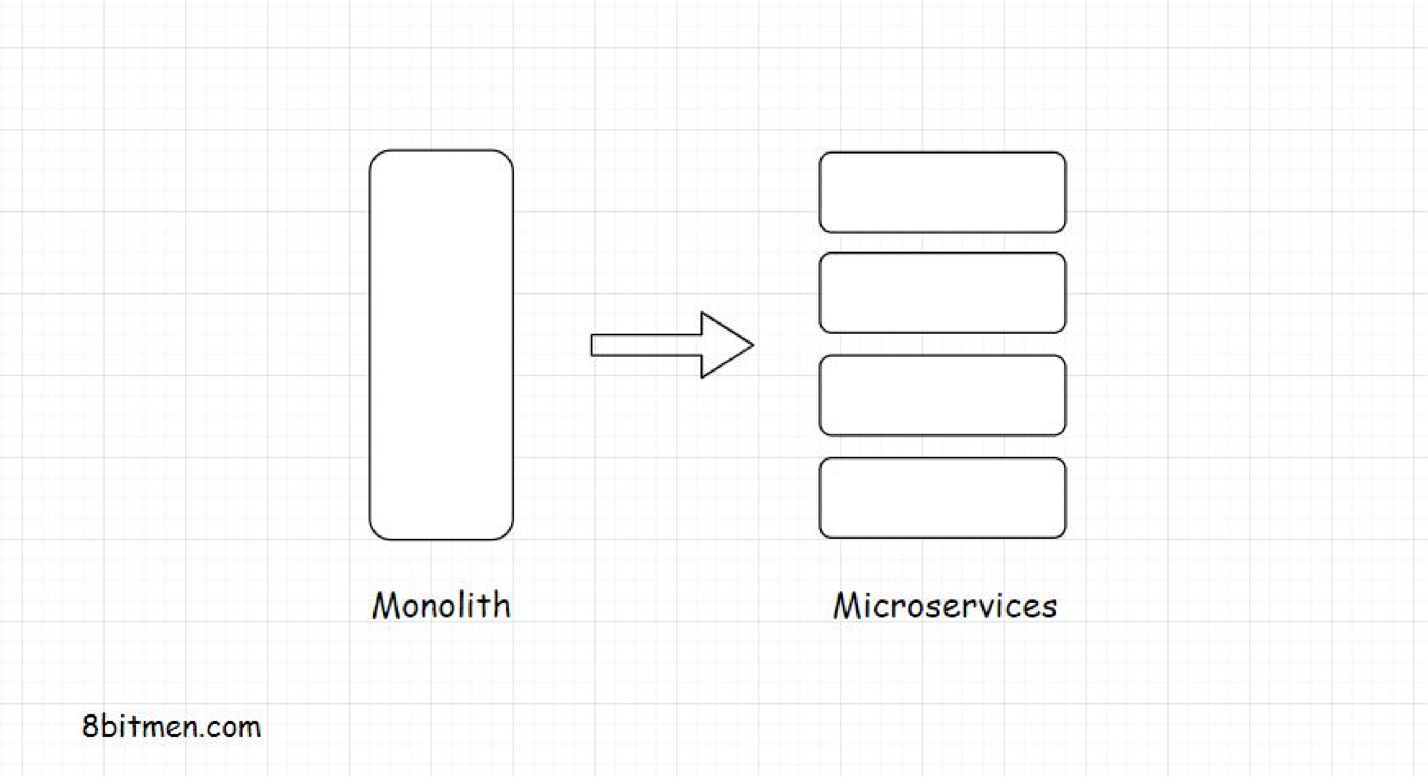
The platform tracks this compute time very minutely, even upto milliseconds, so that businesses don't have to pay anything extra.

Some examples of *FaaS* are *AWS Lambda, Google Cloud Functions*, and *Azure Functions*. Now, let's have a quick insight into the *FaaS* architecture.

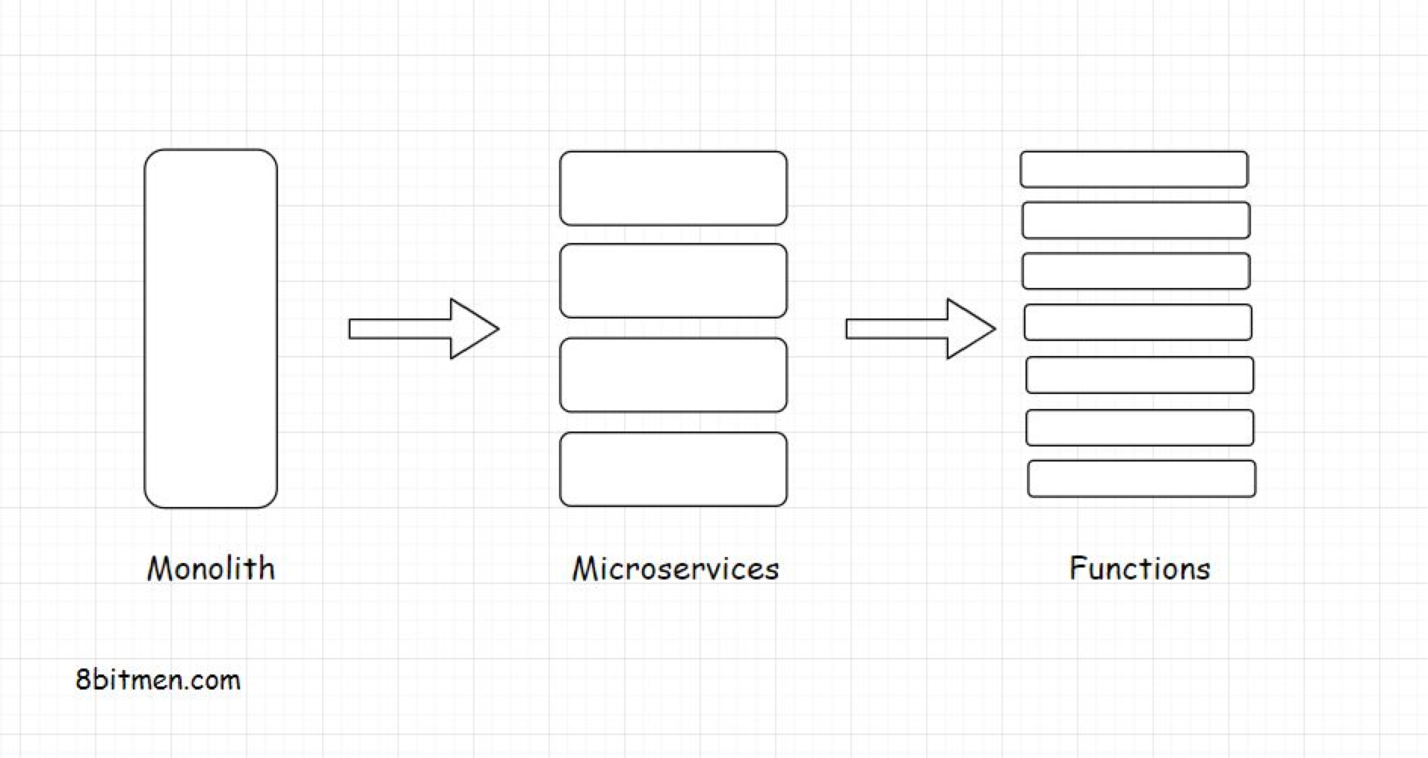
Functions as a Service architecture

FaaS - A step further from microservices

Originally, we had application monoliths. Then, we split those *monoliths* into *microservices* that helped us scale and make our architecture loosely coupled. It gave us the confidence that making a change to one module of the system won't negatively impact or even bring down the entire service for that matter. The split enables us to manage the system more efficiently.



FaaS is a step further from microservices. It’s a more granular approach. With this service model, a microservice is split into functions, giving away deployment control to the cloud platform. When using FaaS, we do not have to manage the *containers*, as we did earlier, to deploy our microservices or worry about implementing monitoring and so on.



Let’s continue this discussion in the next lesson.