Web Technology and Services

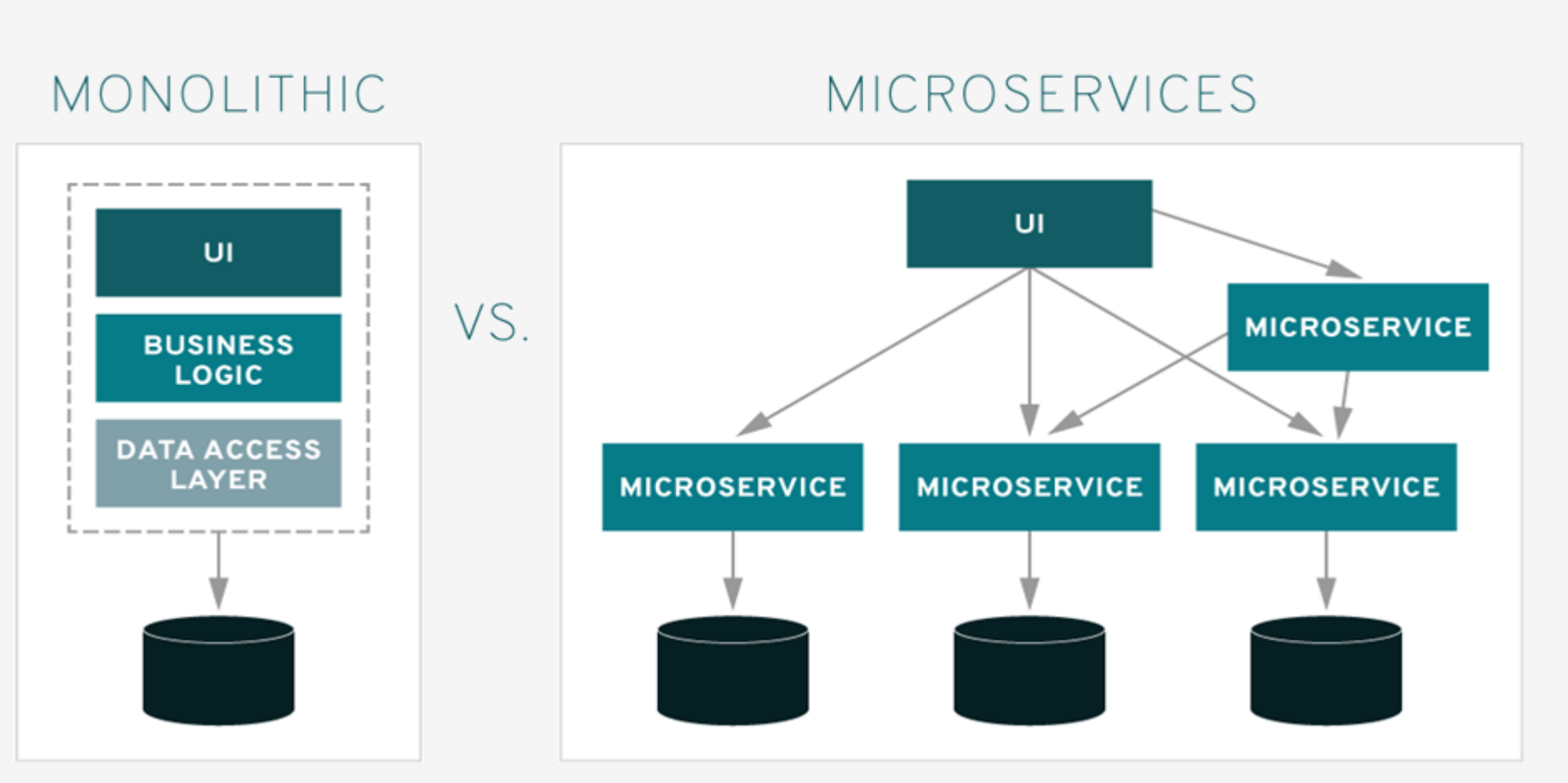
Betim Gashi

betim.gashi@ubt-uni.net

Understand Monolithic & Microservice Architecture

- Monolithic Architecture
- Microservice Architecture
- Docker - Enterprise Application Container Platform
- Serverless Architecture

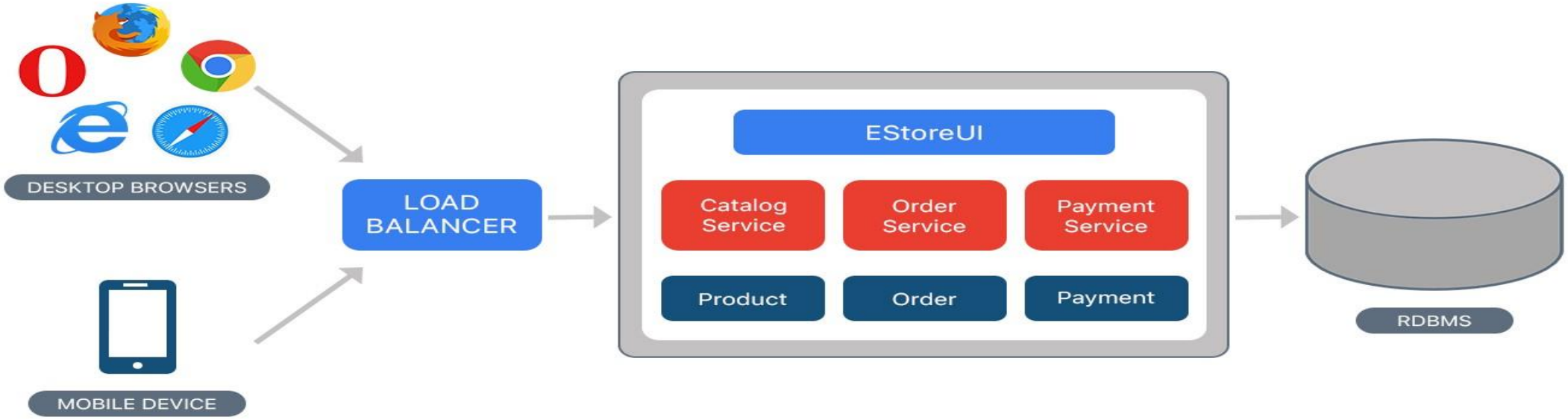
Monolithic & Microservice Architecture



Monolith means composed all in one piece. The Monolithic application describes a single-tiered software application in which different components combined into a single program from a single platform.

Monolithic Approach

Example of Ecommerce application, that authorizes customer, takes an order, check products inventory, authorize payment and ships ordered products.



Benefits

Simple to develop –At the beginning of a project it is much easier to go with Monolithic Architecture.

Simple to test. For example, you can implement end-to-end testing by simply launching the application and testing the UI with Selenium.

Simple to deploy. You have to copy the packaged application to a server.

Simple to scale horizontally by running multiple copies behind a load balancer.

Definition

A microservices architecture consists of a collection of small, autonomous services. Each service is self-contained and should implement a single business capability.

Microservices are an approach to application development in which a large application is built as a suite of modular services (i.e. loosely coupled modules/components).

Each module supports a specific business goal and uses a simple, well-defined interface to communicate with other sets of services.

Microservices Approach

Example of the e-commerce application, which consists of several components/modules.

Microservices Architecture (for E-Commerce Application)

Benefits:

Microservices Enables the continuous delivery and deployment of large, complex applications.

Better testability –services are smaller and faster to test.

Better deployability –services can be deployed independently.

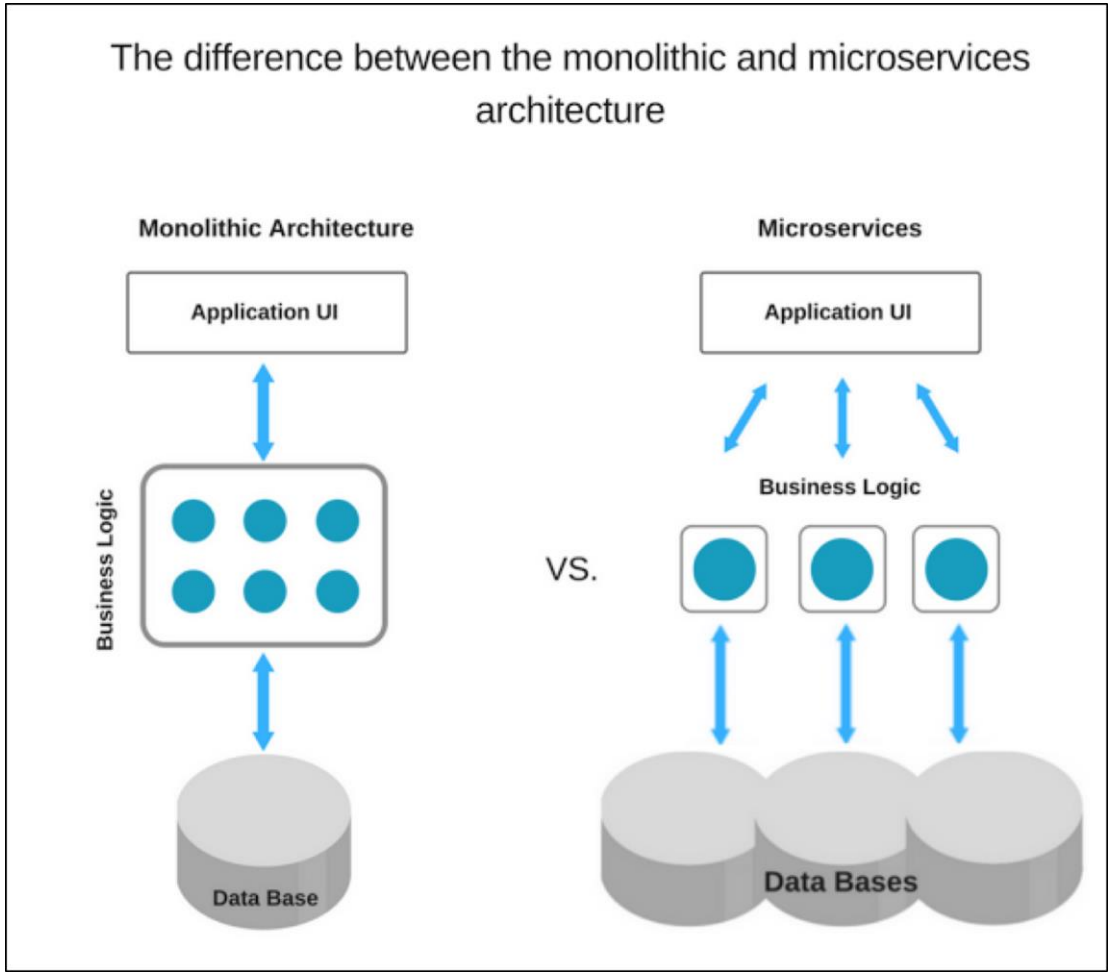
Each microservice is relatively small

Comfortable for a developer to understand

Microservices Eliminates any long-term commitment to a technology stack.

Both approaches have their pros and cons, but it depends on each scenario or product/project requirements and which tradeoff you choose.

As monolithic approach best suits for lightweight applications It is recommended to adopt Monolithic approach first and depending on the needs/requirement gradually shift towards Microservices approach.

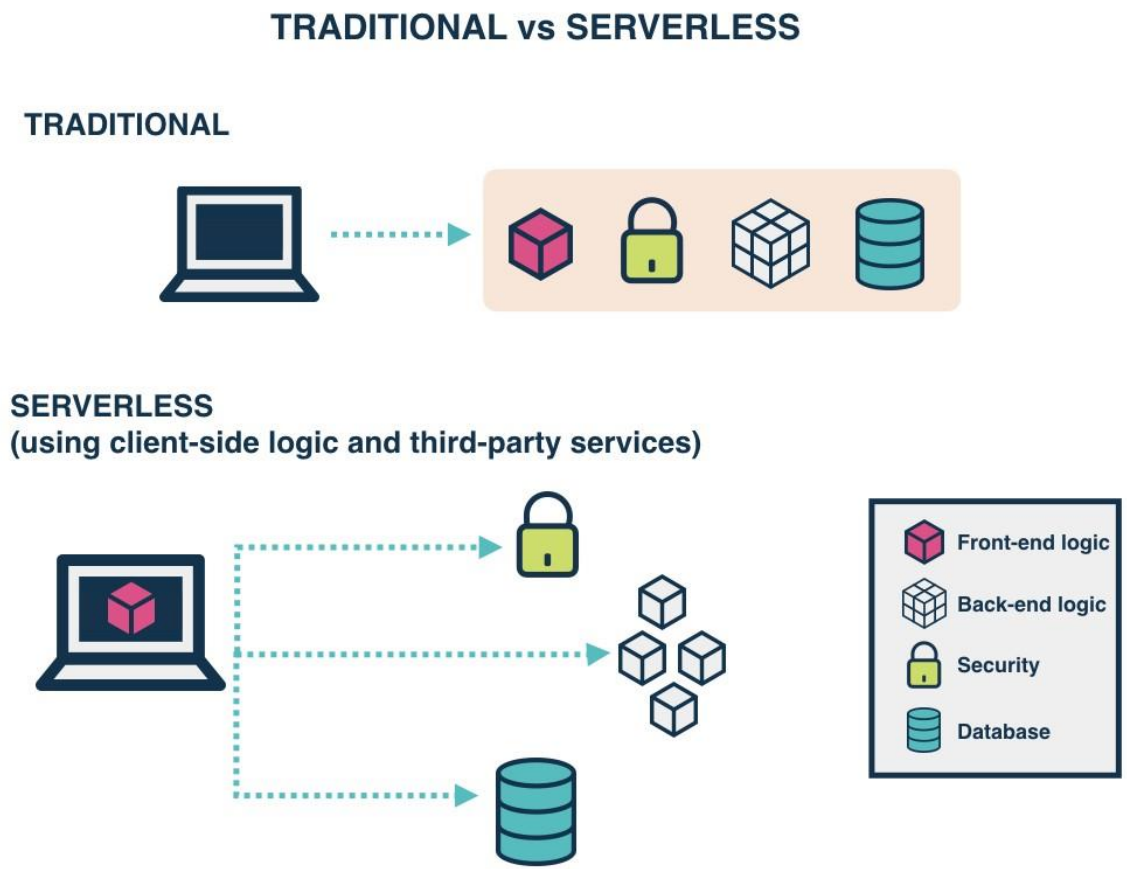


Serverless Architecture

- Serverless allows you to build and run applications and services without thinking about servers. It eliminates infrastructure management tasks such as server or cluster provisioning, patching, operating system maintenance, and capacity provisioning.

- No Server management
- Flexible Scaling
- Automated high availability

Serverless Architecture



Serverless Architecture

Pros

- Though not a magical wand serverless architecture does offer a couple of amazing advantages.
- Pricing. One of the biggest advantages of Serverless computing is, of course, low pricing. You pay only for what is used. On top of that you are reducing operational costs ie years of server-related fees and capacity that has been pre-purchased - basically, all of that is now money saved.
- Reduced maintenance. With no backend infrastructure to manage, as it all falls to Cloud vendors, you are free of maintenance problems you'd have if you're running a server on your own. That includes updates, patches and taking care of potential hacks.
- Scalability. Serverless architectures are inherently scalable allowing your application to, for example, scale automatically as the user base grows or usage increases.
- Faster delivery. For example, FaaS functions are a lot more straightforward when deploying, as opposed to the deployment of a huge server.
- Setting up different environments. Much easier than with the traditional approach.
- Security. But only in the sense that you are responsible for your application and code.

?