

# Monolithic Applications

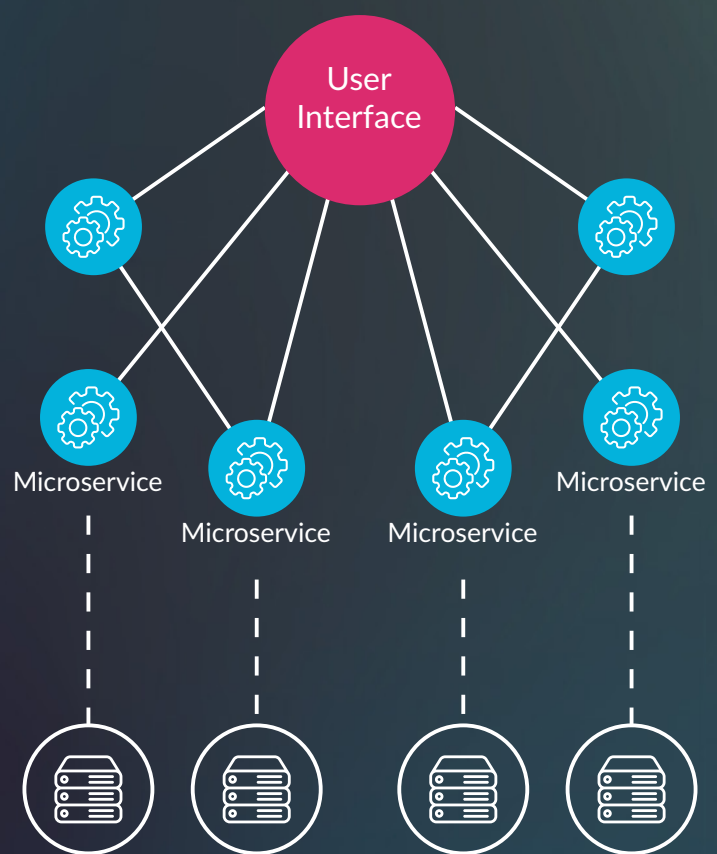
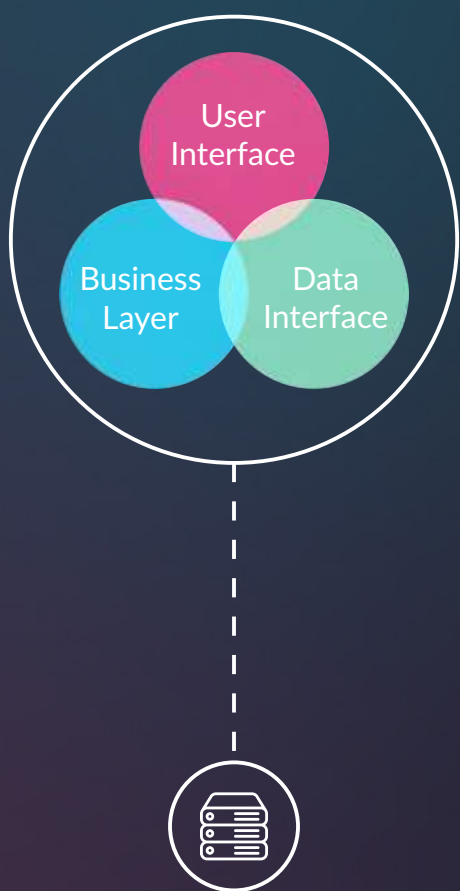
# VS

# Microservices

Wondering about the differences between monolithic applications vs microservices? As more companies move to a cloud-first architecture, it's important to understand these terms.

The traditional monolithic enterprise application is usually presented as a three-tier architecture – a client-side application, a database, and a server-side application. The server-side application itself consists of a web interface, a business logic layer, and a data layer.

It may be built either as one large “monolithic” block of code, or as a collection of small, independent, and reusable pieces called microservices. Most legacy applications have been built as monoliths. Converting them to microservices has both benefits and challenges.



## Monolithic Applications

A monolithic architecture is one where all the components of an application are located within a single code base. The web interface, business logic layer, and data layer are all combined in a single instance of the application. The application is deployed as a single unit, such as a JAR, WAR, exe, or something similar.

Monolithic code is organized as a single unit with various functionalities and dependencies interwoven throughout the code. The coding shortcuts, ad hoc patches, and documentation inadequacies that are typical sources of technical debt in legacy applications are embedded in the code in ways that are extremely difficult for humans to unravel. Worse, because of hidden dependencies in the code, any changes aimed at upgrading functions or adding features may ripple throughout the codebase in unexpected ways, potentially causing the entire application to fail.

## Microservices

A microservices architecture consists of services that are small, independent, and loosely coupled. Microservices are small chunks of code that perform a single task. Every service can be an independent application with its own programming language, development and deployment framework, and database. Each service can be modified independently and deployed by itself. This allows developers to change a specific function in an application by updating the associated microservice without the risk of unintentionally impacting the codebase as a whole. A Gartner study shows that microservices can deliver better scalability and flexibility.



vFunction

[Request a Demo](#) ▶

### About vFunction

vFunction is the first and only AI-driven platform for architects and developers that intelligently and automatically transforms complex monolithic applications into microservices, restoring engineering velocity and optimizing the benefits of the cloud. Designed to eliminate the time, risk and cost constraints of manually modernizing business applications, vFunction delivers a scalable, repeatable factory model purpose-built for cloud native modernization. With vFunction, leading companies around the world are accelerating the journey to cloud-native architecture and gaining a competitive edge. vFunction is headquartered in Palo Alto, CA, with offices in Israel. To learn more, visit [vFunction.com](https://vFunction.com).