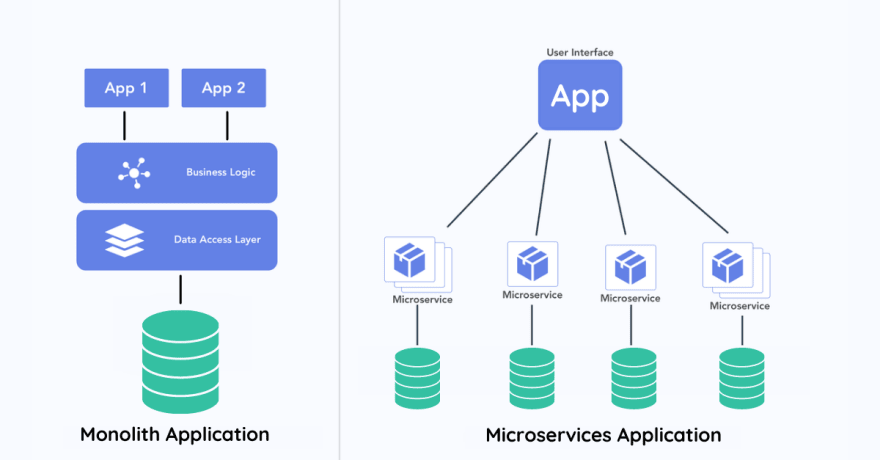
Micro Services

Monolithic VS Microservices



* Monolithic - Monolithic architecture is a traditional software design model where all the components of an application are integrated into a single, unified codebase. This single executable or deployable unit contains all the functionality of the application, including the user interface, business logic, and data access layers. Changes or updates to any part of the application typically require redeploying the entire application, making it challenging to scale, maintain, and adapt as the application grows.
* Micro Services - Microservices are a software development approach where an application is built as a collection of small, independent, and loosely coupled services, each responsible for a specific business function. These services communicate with each other through well-defined APIs and can be developed, deployed, and scaled independently. This architecture enhances flexibility, maintainability, and scalability, allowing teams to develop, test, and deploy features rapidly and efficiently.

Key Components

* Service Registry - Enables microservices to register themselves and discover other services dynamically. Each microservice registers with Eureka upon startup, facilitating communication without hardcoded dependencies.
* Configuration Server - Centralize configuration management. Microservices fetch configurations from the Config Server, allowing updates without redeployment.
* API Gateway - Serve as a single-entry point for client requests. Routes requests to appropriate microservices based on URI and HTTP method, leveraging Eureka for dynamic service discovery.
* Services – These are the services that interact with the API Gateway.