**MICROSERVICES ARCHITECTURE**

**INTRODUCTION:**

Microservices architecture refers to **an architectural style for developing applications**. Microservices allow a large application to be separated into smaller independent parts, with each part having its own realm of responsibility.

WHY MICROSERVICES?

* The most popular architecture paradigm.
* Not tied to specific technology
* Solves real problems
* In high demand in the job market
* Generates a lot of buzz

Applications were traditionally built as monolithic pieces of software. Traditional monolithic applications have long lifecycles, are updated infrequently and changes usually affect the entire application.

**Problems with Monolith and SOA:**

* Single Technology
* Inflexible Deployment
* Inefficient Compute Resources
* Large and Complex
* Complicated and Expensive ESB
* Lack of Tooling

**Characteristics of MicroServices:**

* Componentization via services
* Organized around Business Capabilities
* Products not Projects
* Smart endpoints and Dumb pipes
* Decentralized Governance
* Decentralized Data Management
* Infrastructure Automatic
* Design for failure
* Evolutionary Design

**5 Components of Microservices architecture:**

* Microservices
* Containers
* Service Mesh
* Service Discovery
* API gateway

Following are some rules that we need to keep in mind while developing a Microservice-oriented application.

1. **Independent:**

Each microservice should be independently deployable.

1. **Coupling:**

All microservices should be loosely coupled with one another such that changes in one will not affect the other.

1. **Business Goal:**

Each service unit of the entire application should be the smallest and capable of delivering one specific business goal.