**Monolithic Architecture: A Brief Overview**

**Monolithic architecture** is a traditional way of building applications where all the components and services are tightly coupled and run as a single, indivisible unit.

**Key Characteristics**

* **Single Codebase:** All application functionalities reside in one codebase.
* **Interdependent Components:** Components are interconnected, and changes in one can affect others.
* **Single Deployment:** The entire application is deployed as a single unit.

**Advantages**

* **Simplicity:** Easier to develop initially due to a single codebase.
* **Lower Initial Overhead:** Less complex infrastructure is needed in the early stages.
* **Simplified Deployment:** Deploying a single unit is straightforward.

**Disadvantages**

* **Scalability Challenges:** Scaling specific features can be difficult; the entire application needs to be scaled.
* **Maintenance Complexity:** As the application grows, maintaining and updating the large codebase becomes cumbersome.
* **Slower Development:** Large codebases can slow down development cycles.
* **Technology Stack Limitations:** Developers are often locked into the initial technology stack.
* **Single Point of Failure:** A failure in one component can bring down the entire application.

**Comparison with Microservices**

Monolithic architecture is often contrasted with **microservices architecture**, where applications are structured as a collection of small, independent services.

**In summary**, monolithic architecture can be a good starting point for small or simple applications. However, as applications grow in complexity, the limitations of this approach can hinder scalability, maintainability, and development speed.