Monolithic V/s Microservices

- Monolithic architecture and microservices architecture are two distinct approaches to building software applications. Monolithic architecture involves a single, unified codebase, while microservices architecture breaks down the application into smaller, independent services. The choice between them depends on the specific needs and goals of the application, with each approach offering advantages and disadvantages.

Monolithic Architecture:

Definition:

- A monolithic application is built as a single unit, with all components tightly coupled and deployed together.

Characteristics:

- Single codebase.
- All components are interdependent.
- Deployable as a single unit.

Advantages:

- Simpler to build and deploy initially, especially for smaller projects.
- Can be more performant for simpler applications.
- Easier to manage and maintain in its early stages.

Disadvantages:

- Can become difficult to maintain and scale as the application grows.
- Changes in one part of the application can impact the entire system.
- Deployment and updates are slower and can be more disruptive.
- Less resilient to failures, as a failure in one part can bring down the entire application.

Microservices Architecture:

Definition:

- A microservices architecture breaks down the application into smaller, independent services that communicate with each other through APIs.

Characteristics:

- Independent services.
- Loose coupling between services.
- Deployable and scaleable independently.

Advantages:

- Improved scalability, as individual services can be scaled independently.
- Faster development cycles, as teams can work on individual services concurrently.
- Increased fault tolerance, as failures in one service are less likely to impact the entire application.
- Easier to adapt to changing requirements.

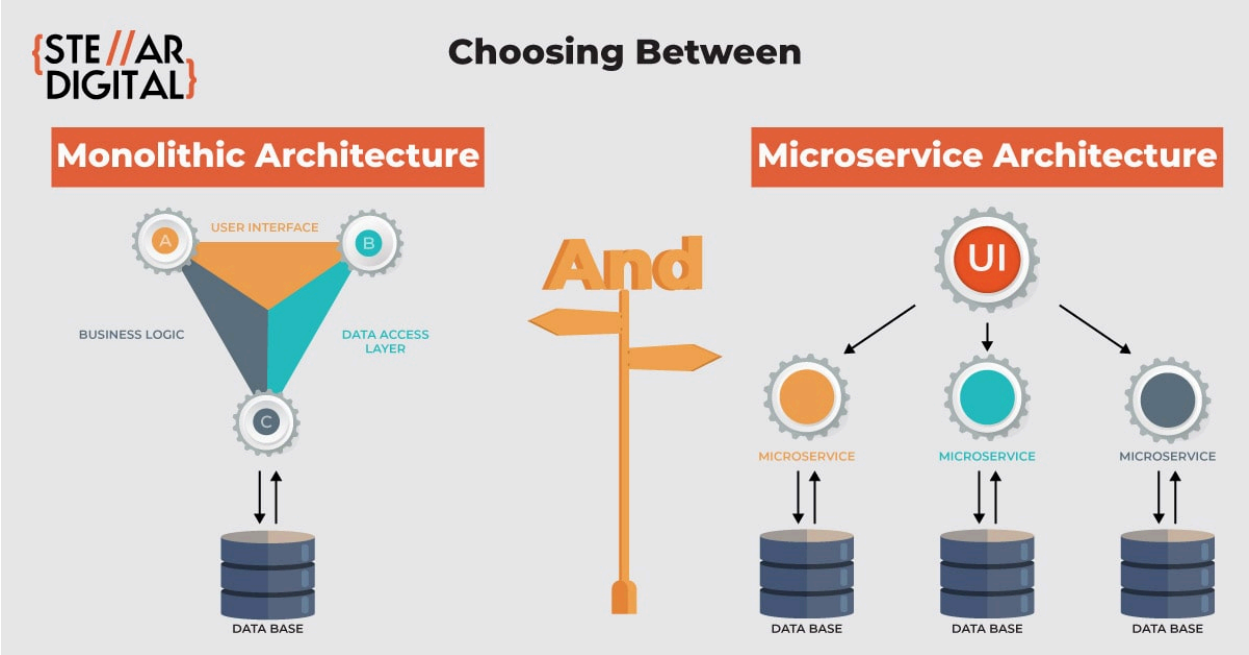
Disadvantages:

- Increased complexity, as managing and coordinating multiple services can be challenging.
- Potential for increased latency due to network communication between services.
- Requires more expertise in distributed systems and DevOps.

Key Differences Summarized:

Feature	Monolithic Architecture	Microservices Architecture

Codebase	Single, unified codebase	Multiple, independent services
Coupling	Tightly coupled	Loosely coupled
Deployment	Single deployment unit	Independent deployments
Scalability	Scalable as a whole	Scalable independently by service
Complexity	Lower initial complexity, but can increase over time	Higher initial complexity, but potentially lower overall
Fault Tolerance	Less resilient, failure in one part can affect the whole	More resilient, failures are isolated


{STE//AR DIGITAL}
Choosing Between
Monolithic Architecture
Microservice Architecture
USER INTERFACE
A
B
C
BUSINESS LOGIC
DATA ACCESS LAYER
DATA BASE
And
UI
MICROSERVICE
MICROSERVICE
MICROSERVICE
DATA BASE
DATA BASE
DATA BASE