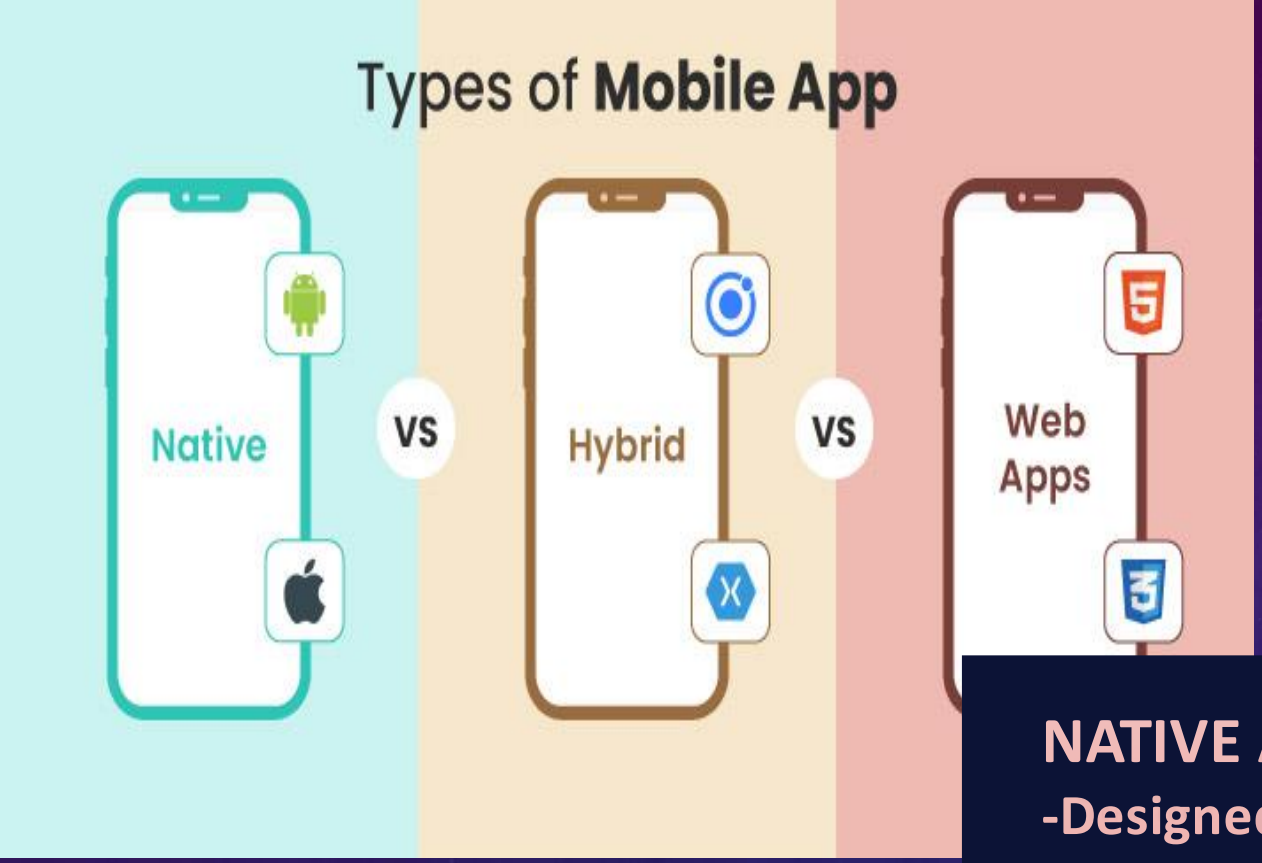
MOBILE APPS

By GROUP 2|
Supervised by DR Nkemeni Valeri



NATIVE APPS

-Designed for ios and android
-utilizes swift or objective-C for ios
-java and kotlin for android
-offers full access to device

PROGRESSIVE WEB APPS

-Designed for use across different platforms
-Limited access to device features

HYBRID APPS

-Developed using web technologies (html, css, javascript)
-cross platform compatability
-limited acces to device features

Java/Kotlin:

Used for Android app development. Kotlin is now the preferred language by Google for Android development due to its modern features and interoperability with Java.

Swift/Objective-C:

Used for iOS app development. Swift is the newer and preferred language by Apple for iOS development due to its safety features and performance improvements over Objective-C.

Mobile app programming languages

JavaScript/TypeScript:

Commonly used for hybrid app development using frameworks like React Native or Ionic.

Dart:

Used for developing mobile apps with the Flutter framework, which allows for building apps for both iOS and Android from a single codebase.

Mobile App Development Frameworks

React Native:

Language: JavaScript/TypeScript

Flutter:

Language: Dart

Xamarin:

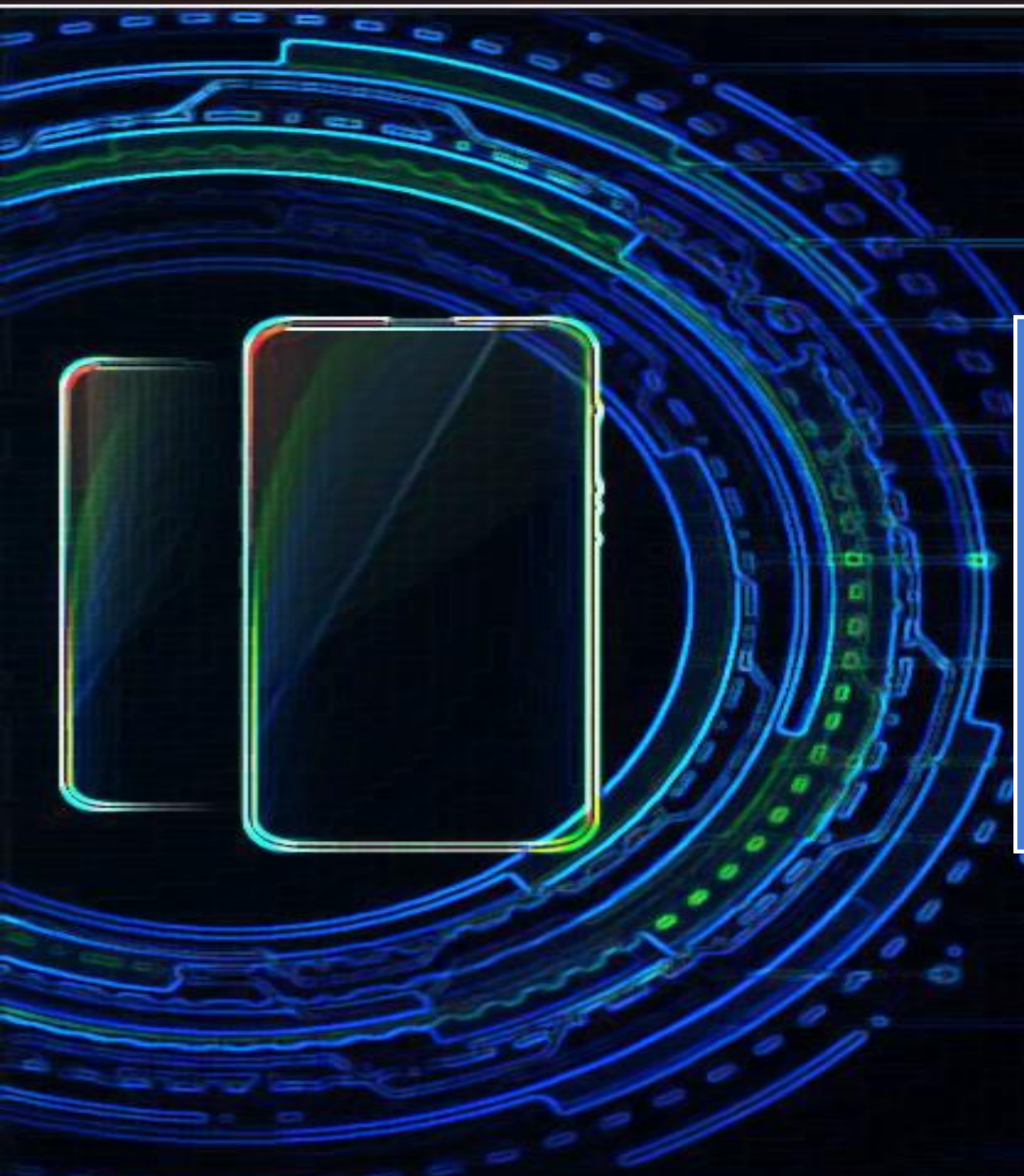
Language: C# with .NET framework

Nativescript:

Javascript/Typescript

Ionic:

Language: HTML, CSS, JavaScript/TypeScript



MOBILE APPLICATION ARCHITECHTUES AND DESIDN PATTERNS

4.1 MOBILE APPLICATION ARCHITECTURE

DEFINITION

- Mobile App Arch it’s a structural design and organization of a mobile app arch that show how the component and module are interconnected and work together to achive a specific goal by in consideration the sustainability , manageability , reusability , security and performance of the mobile app

KEY COMPONENTS

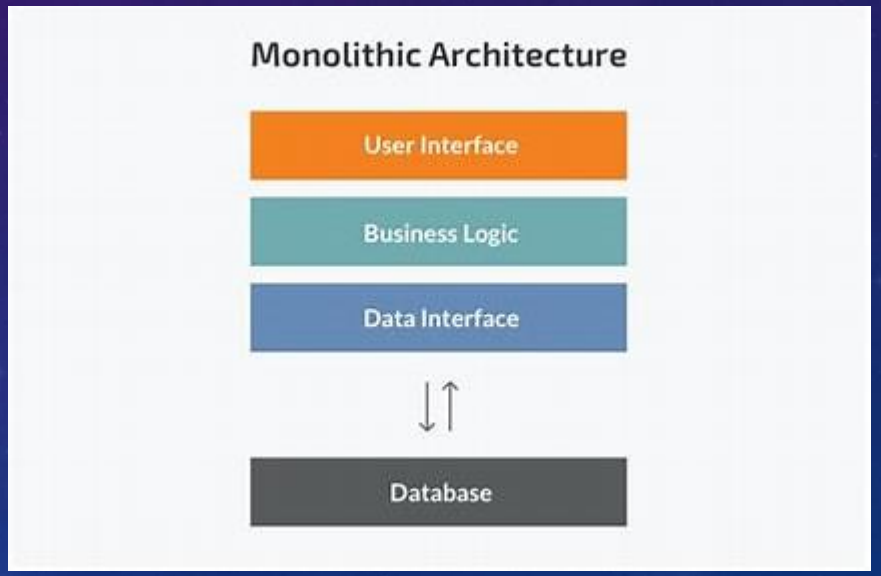
- User Interface (UI)
- App Logic Layer
- Data Layer

EXAMPLE OF MODEL OF MOBILE ARCHITECTURE

1.Monolitic Architecture

- Advantage: Simple to develop, test and deploy

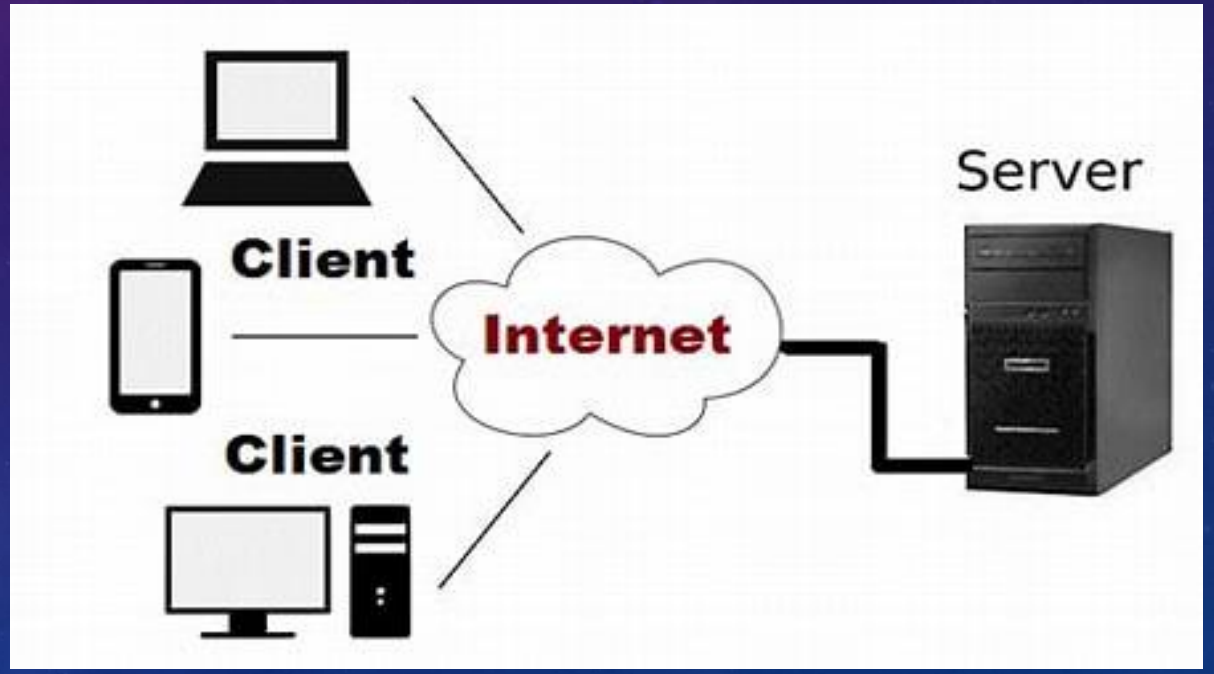
- Consequence: Lacks flexibility and scalability for complex apps



2. Client-Server Architecture

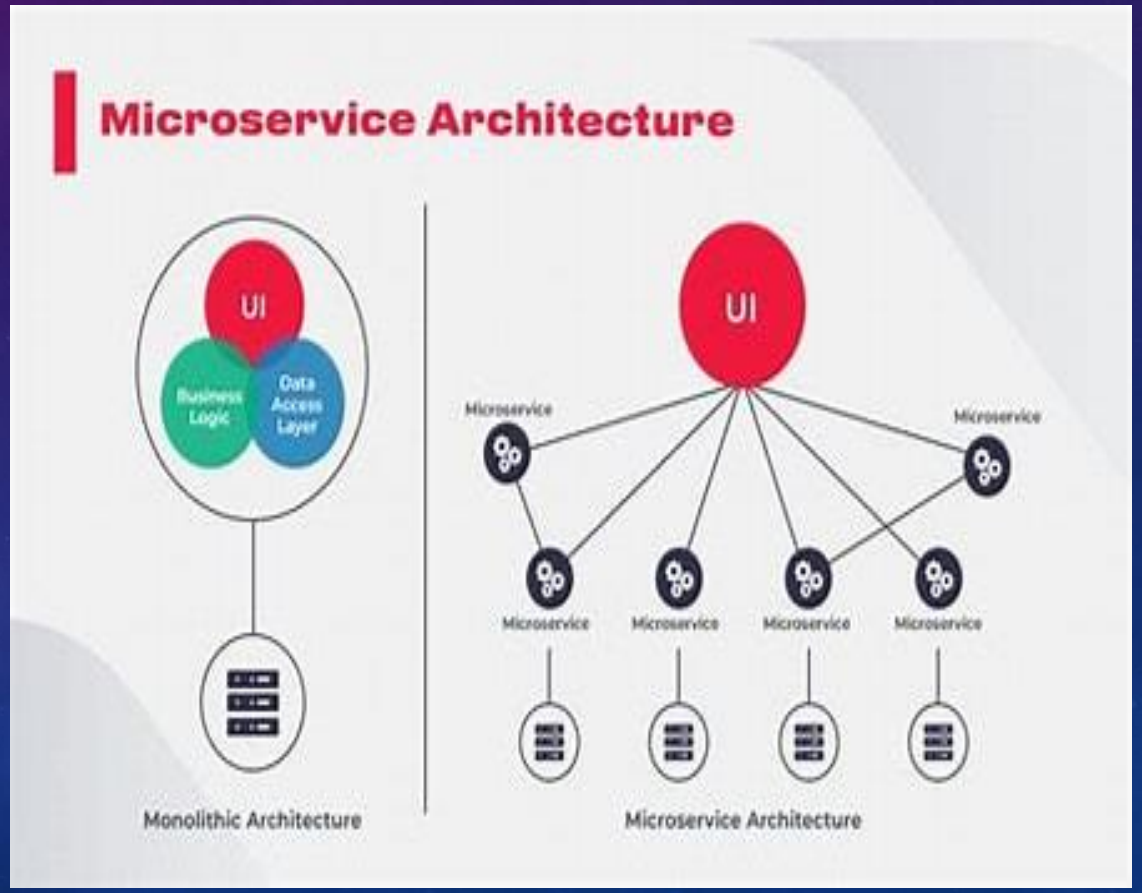
- Advantage: Promotes modularity, parallel workflows

- Consequence: Complex to implement for more superficial apps



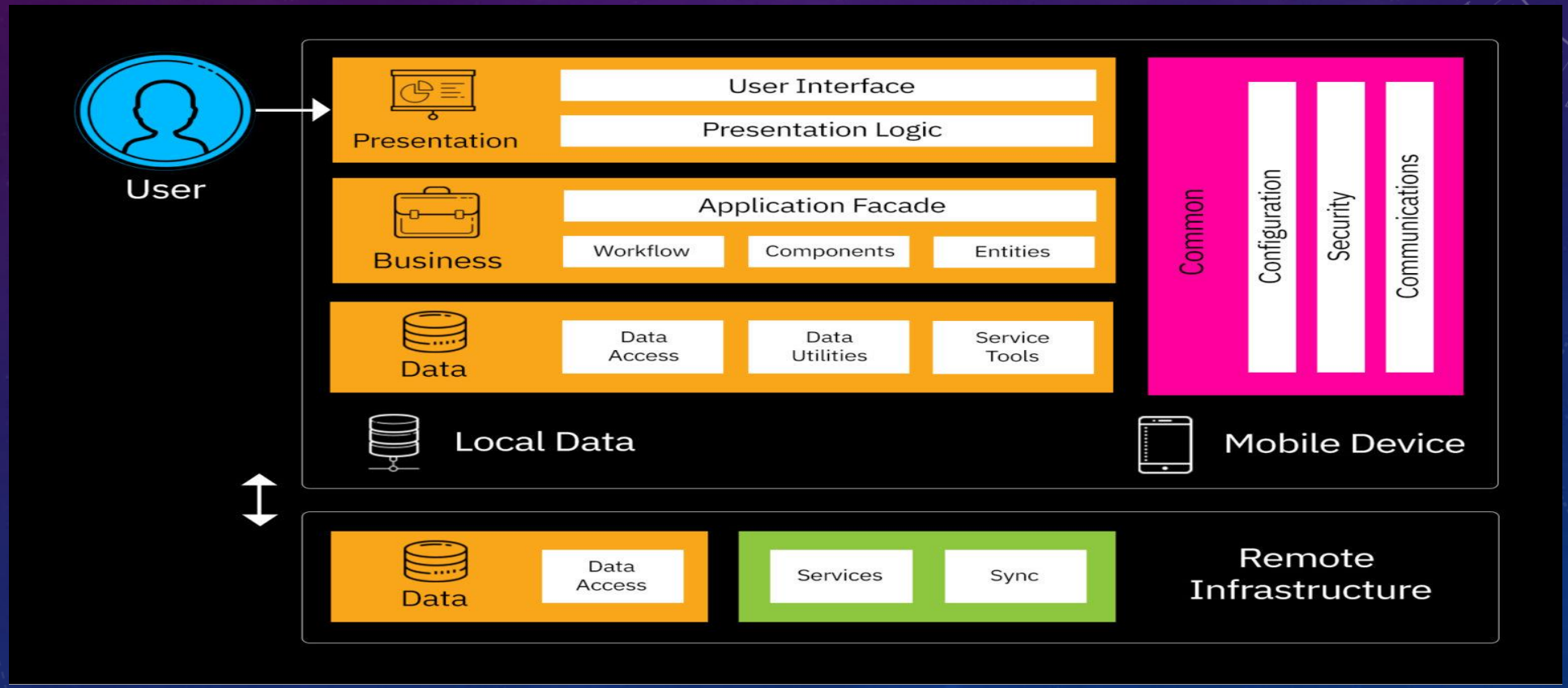
4. Microservice Architecture

- Advantage: Highly scalable. Enables continuous delivery
- Consequence: Complex to create, orchestrate, test and monitor



4.2 DESIGN OF MOBILE ARCHITECTURE

A. MOBILE APP ARCHITECTURE DIAGRAM : THE MOST COMMON



B. FACTORS TO CONSIDER DURING THE MOBILE APP DESIGN

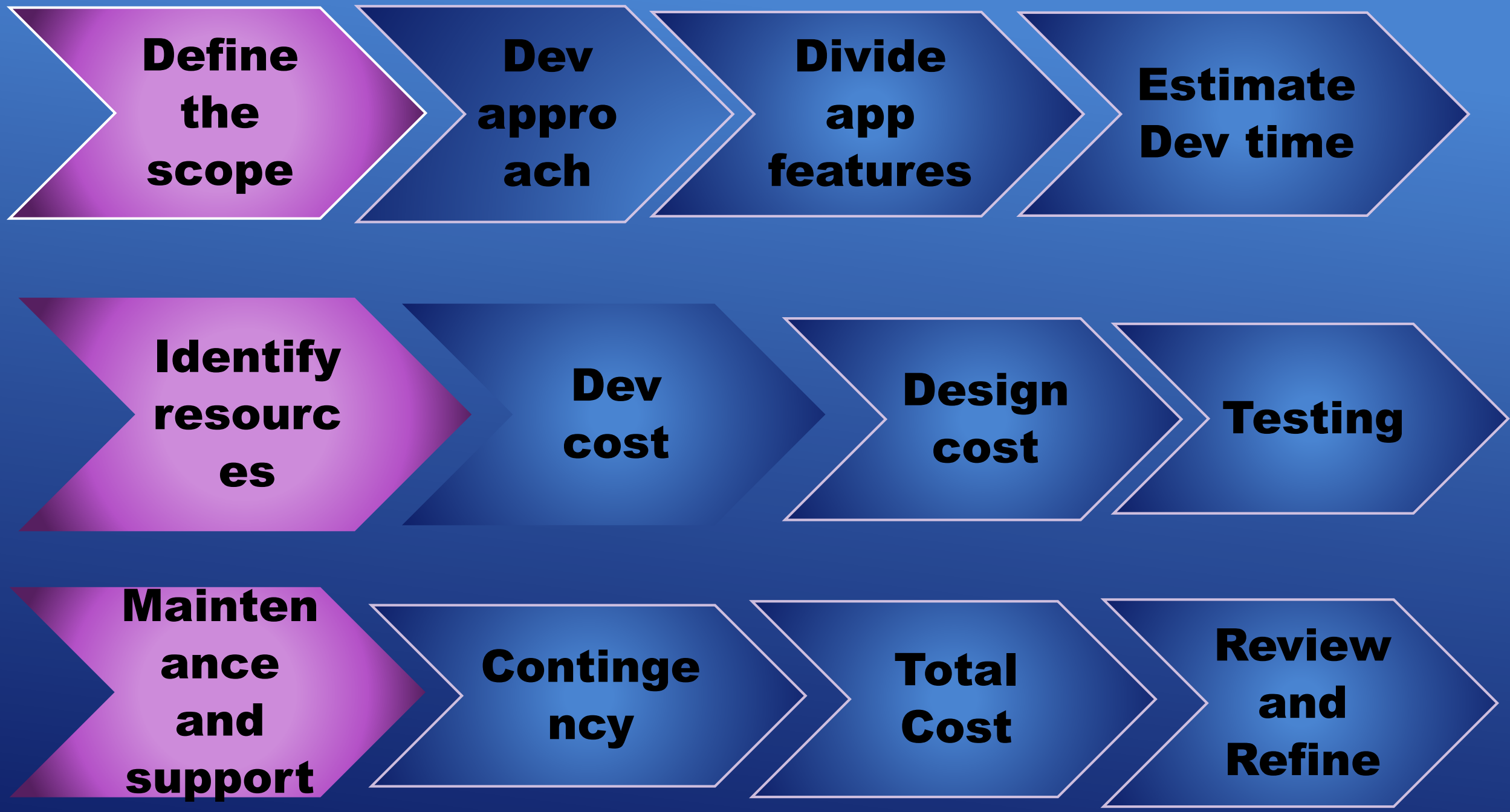
- Device Characteristic
- Development Framework
- User Interface / User Experience Design (UI/UX Design)
- Navigation
- **…etc**

REQUIREMENT ENGINEERING

- Identify stakeholders
- Gather initial information
- User Personas
- Document functional requirements
- Document non-functional requirements
- Use cases and user stories
- Prototyping and mockups

- Prioritize Requirements
- Review and validation
- Traceability and management
- Iterative process

Estimating mobile App Development Cost


Define the scope
Dev appro ach
Divide app features
Estimate Dev time
Identify resourc es
Dev cost
Design cost
Testing
Mainten ance and support
Contingе ncy
Total Cost
Review and Refine