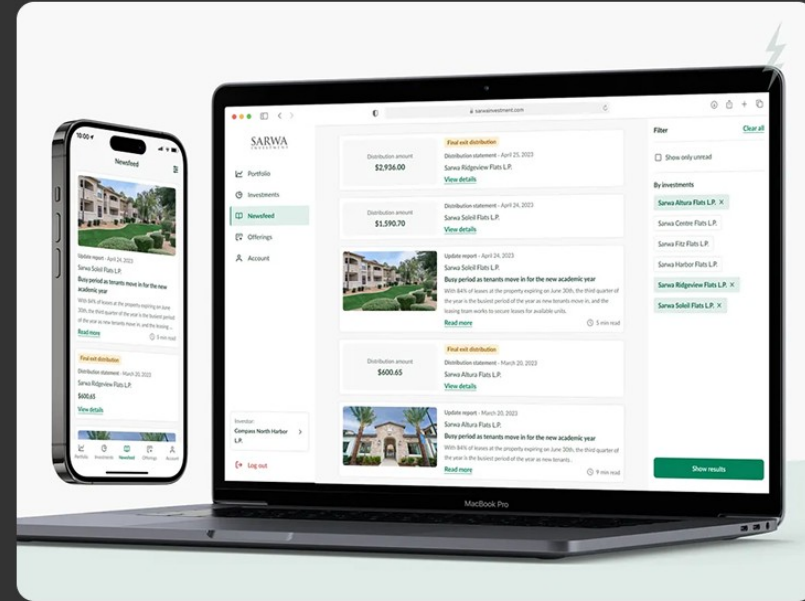# MAUI

Cross platform apps

- Supports multiple platforms from a single codebase

- Cross platform frameworks are used to build these applications

- Preffered for making multi platform projects

Cross platform frameworks

- Allows sharing of business logic between different plaforms

- Access to native api's and UI elements are handled by the framework

- Updating the application is much more streamlined

- React Native, Flutter ,UNO and MAUI are the popular examples
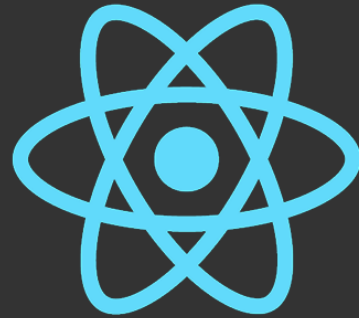
React Native

React Native

- Developed by facebook in order to create iOS and android apps using a single codebase

- Uses React JS for creating native UI elements

- Lightweight and fast prototyping

- Fast UI refresh in development environment

- Allows creation of native elements using platform specific code

Disadvantages

- Large app sizes

- Limited support for native Api access

- Over reliance on third party npm packages

- JavaScript is notorious for heavy memory usage

- Version mismatch for packages can cause bugs or incompatibility

Flutter

- Developed by Google in order to create cross platform applications

- Uses Dart, which is also made by Google, for creating GUI applications

- Dart is compiled to native machine code on production builds, so fast and responsive UI

- Uses Skia, a custom C++ renderer, which renders every UI element the same on every platform

- Made for iOS, Android, MacOS, Linux, Web and Windows application development

Disadvantages

- Isolates uses more memory and are slow to spawn

- Lack of support for a lot of native api's

- Slow industry adoption

- Limited third party libraries

- Large app sizes

MAUI

what is MAUI?

○ Multi platform App UI (MAUI) is an application framework made by Microsoft

○ Built for cross platform app development

○ Powered by the **.NET** framework

○ Uses CSharp and XAML

why use MAUI?

○ Supports iOS, android, macOS and windows app development

○ Provides access to platform specific api's

○ Uses native UI components

○ Runs on the .NET CLR (Common Language Runtime)

Advantages

- Supports true multi-threading

- Small App sizes

- Almost full native api access

- Rich third party libraries

- Allows complex UI and computations

- Uses native renderer

How does MAUI work?
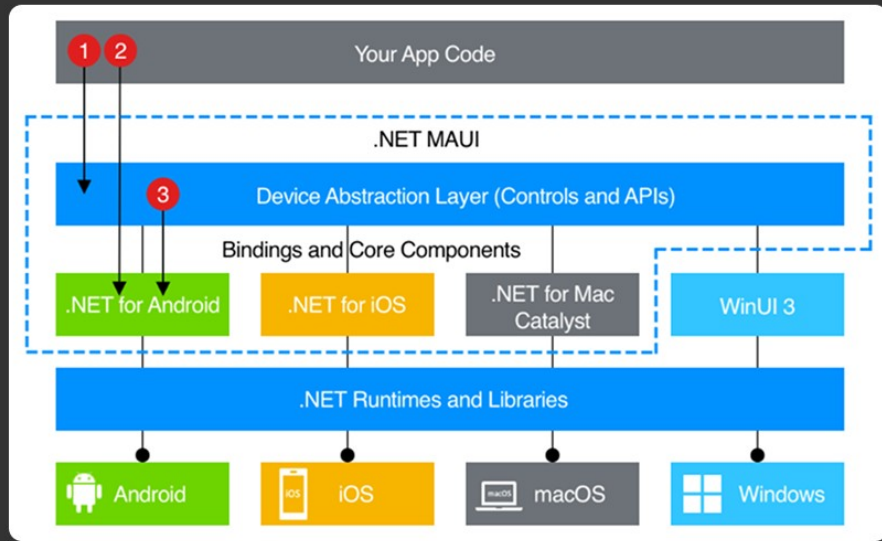
- Program code accesses native api's using built in abstraction layers

- Abstraction layer is made up of MAUI Controls and MAUI api's

- Abstraction layers gets mapped to platform specific api's

In conclusion . . .

- Most cross platforms have limited access to native api's

- Large app sizes

- Lack of third party support

- Doesn't even have multi-threading

- Communication between native api and the framework code is slow

In conclusion . . .

- MAUI has direct access to native apis

- Small app sizes

- Rich third party libraries

- Has multi-threading

- Fast inter communication between native api's and the CLR

Thank you!