

# COMPOSE MULTIPLATFORM

[STEPHEN.WHITE@CALLISTAENTERPRISE.SE](mailto:STEPHEN.WHITE@CALLISTAENTERPRISE.SE)

CADEC 2024.01.18 & 2024.01.24 | CALLISTAENTERPRISE.SE

# CALLISTA

## AGENDA

- What is Compose Multiplatform ( CMP ) ?
- Getting Started
- Demo
- Ongoing Development
- Final thoughts

**WHAT IS COMPOSE MULTI PLATFORM ?**



**WELL, IT'S ALREADY IN YOUR HAND**

## WHAT IS CMP

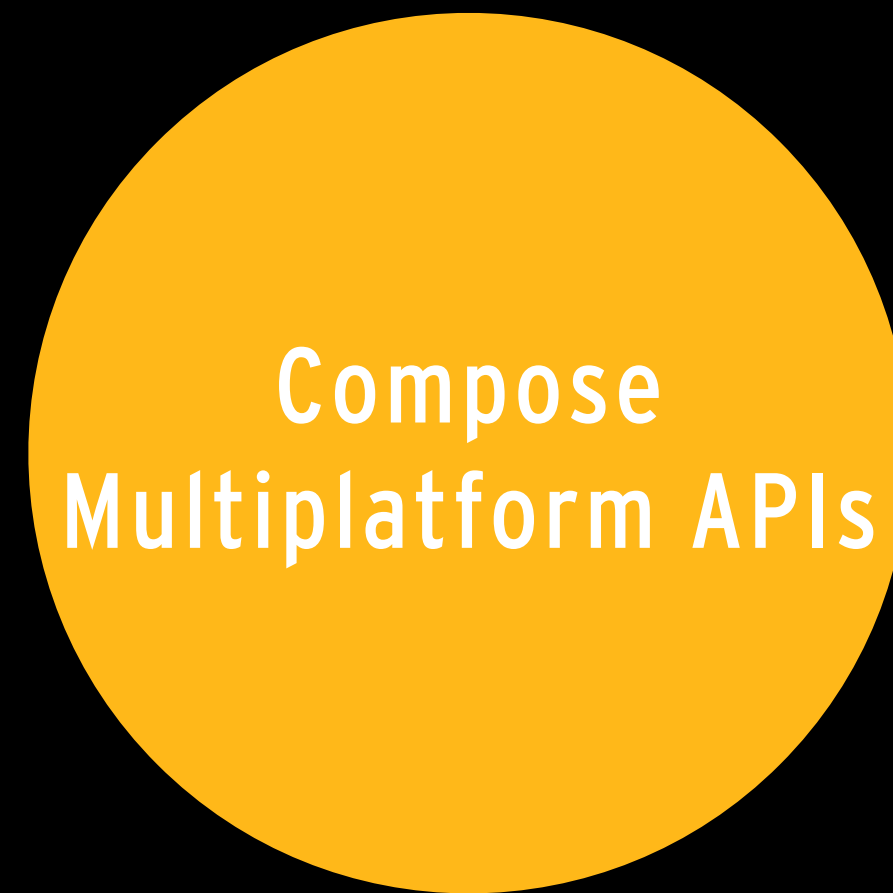
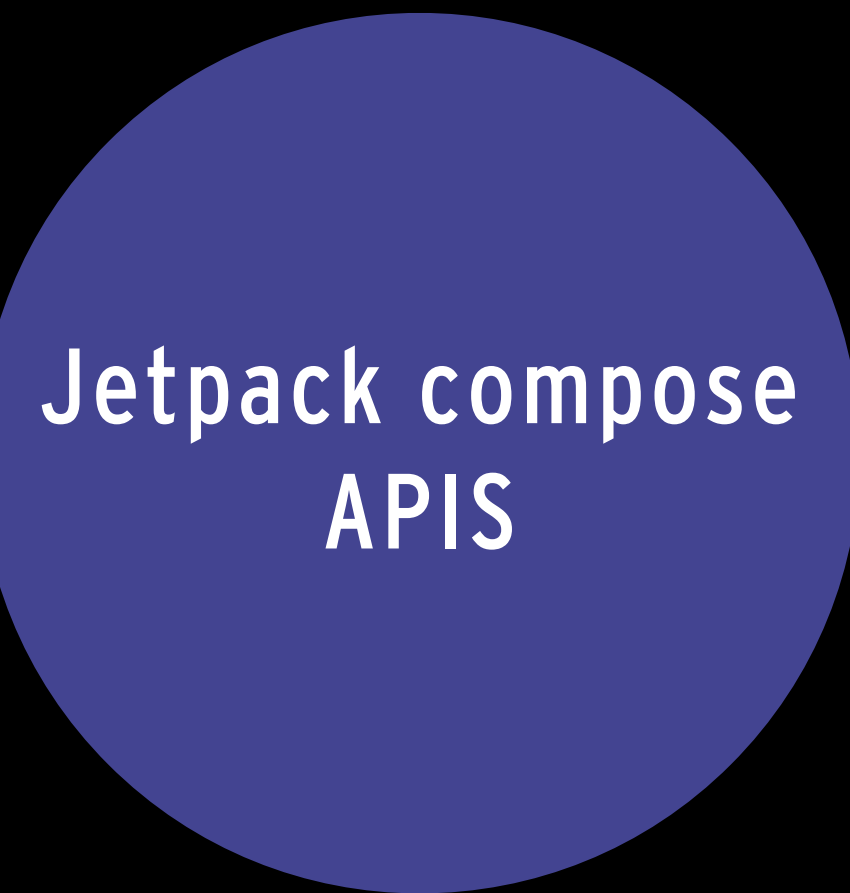
- It's a Multiplatform UI Framework built by JetBrains.
- On top of Jetpack compose, a UI toolkit by Google.
- Can be used for Multiplatform development.



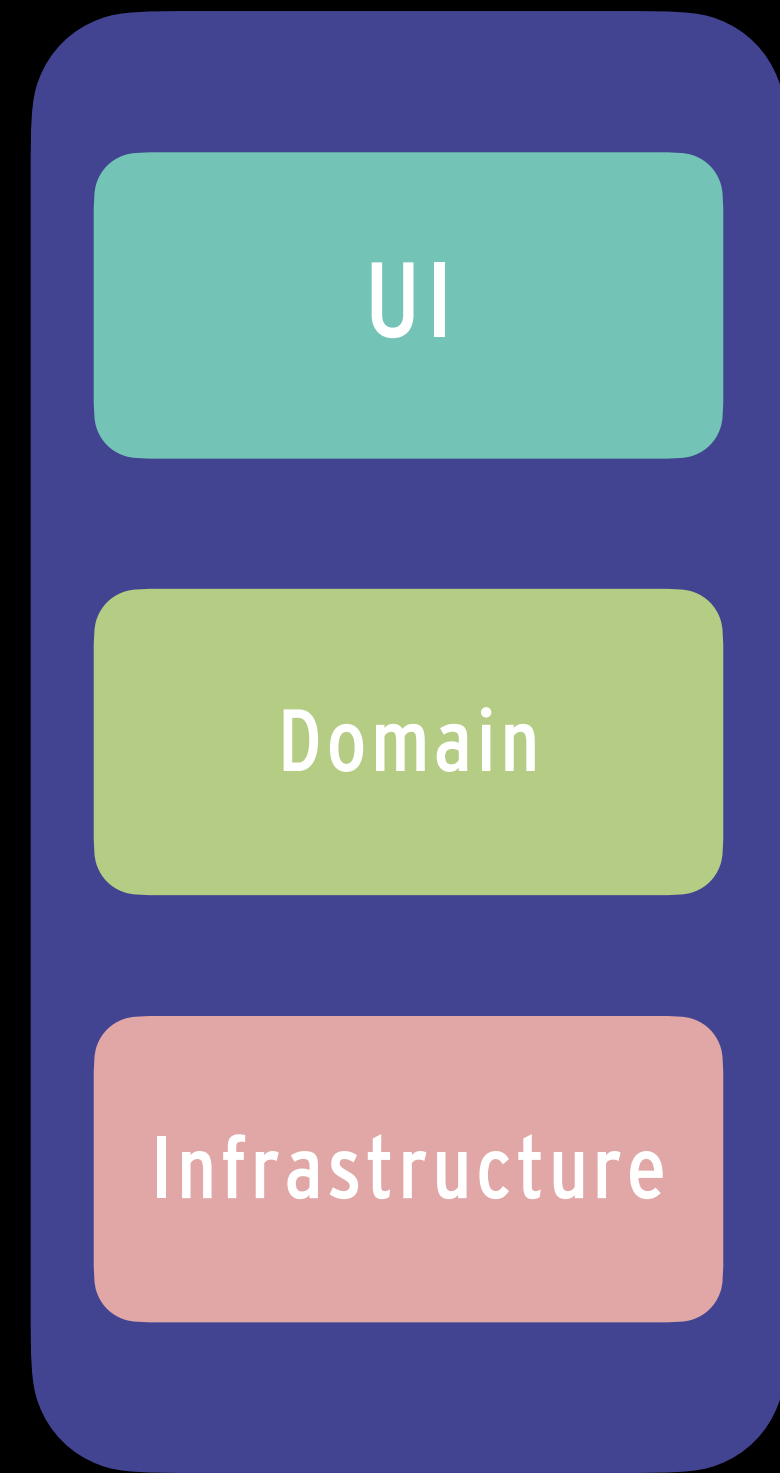
# Compose Multiplatform

Develop stunning shared UIs for Android, iOS, desktop, and web.

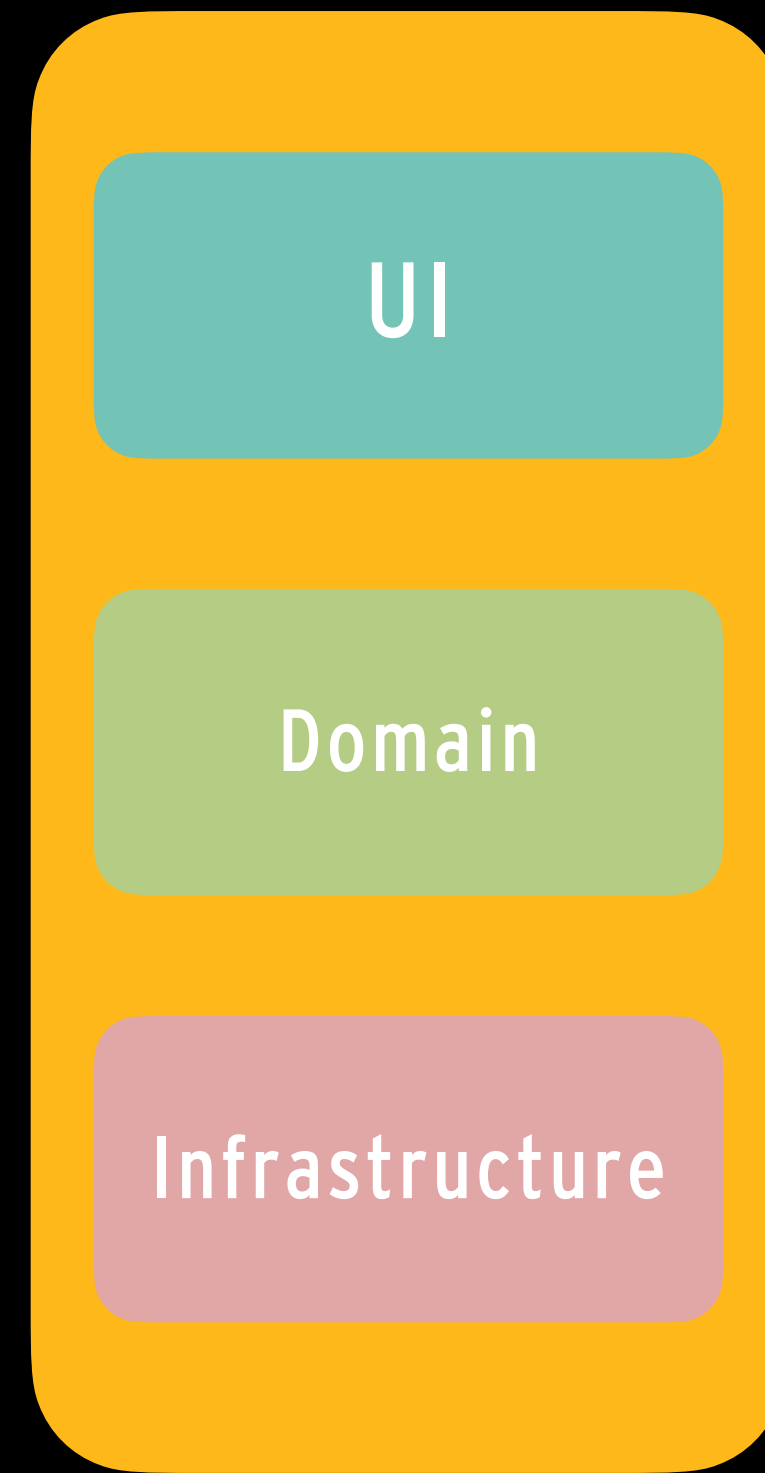
# WHAT IS CMP - JETPACK COMPOSE APIS



# WHAT IS CMP - TRADITIONAL APPROACH - SEPARATE CODE BASES

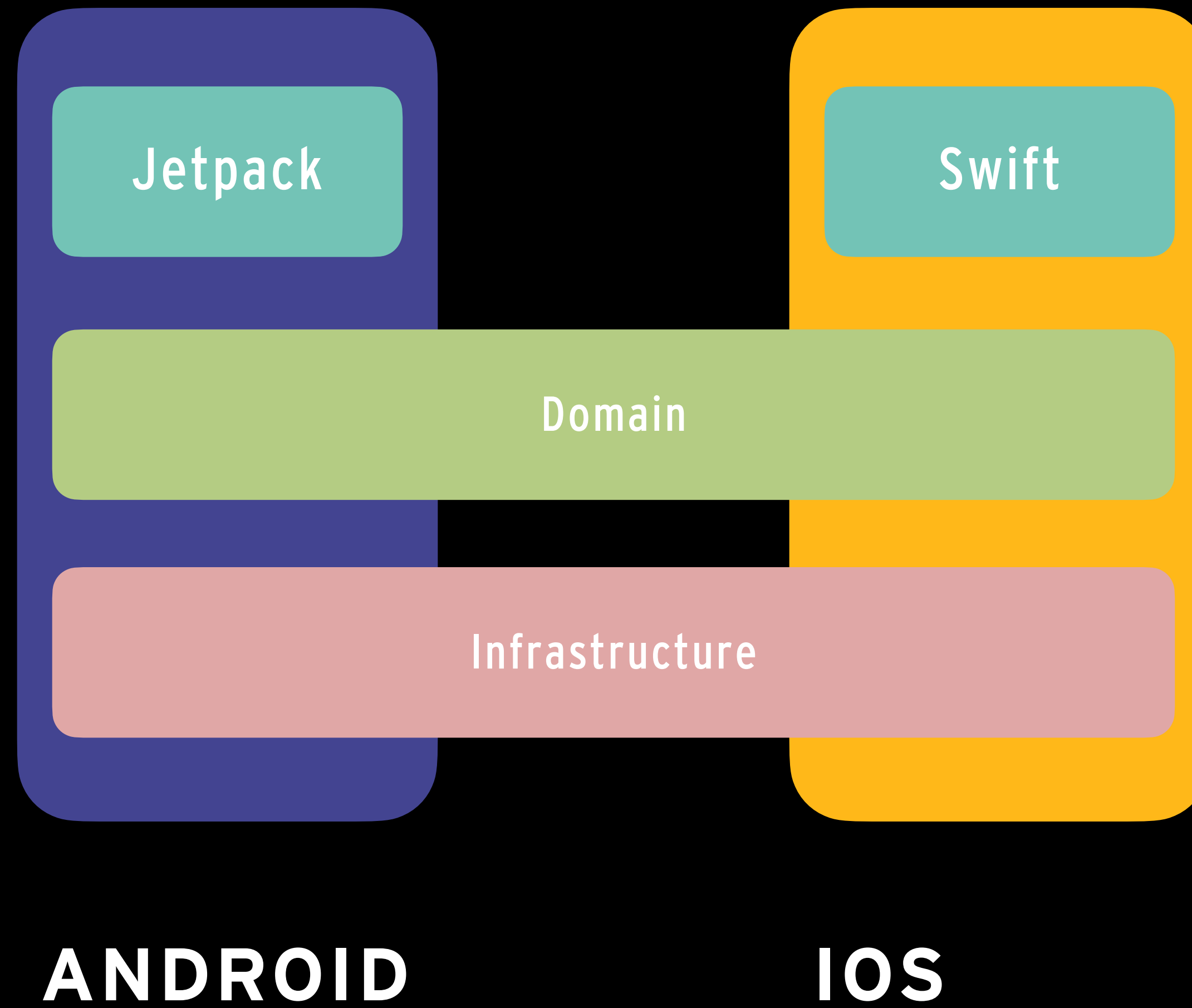


**ANDROID**



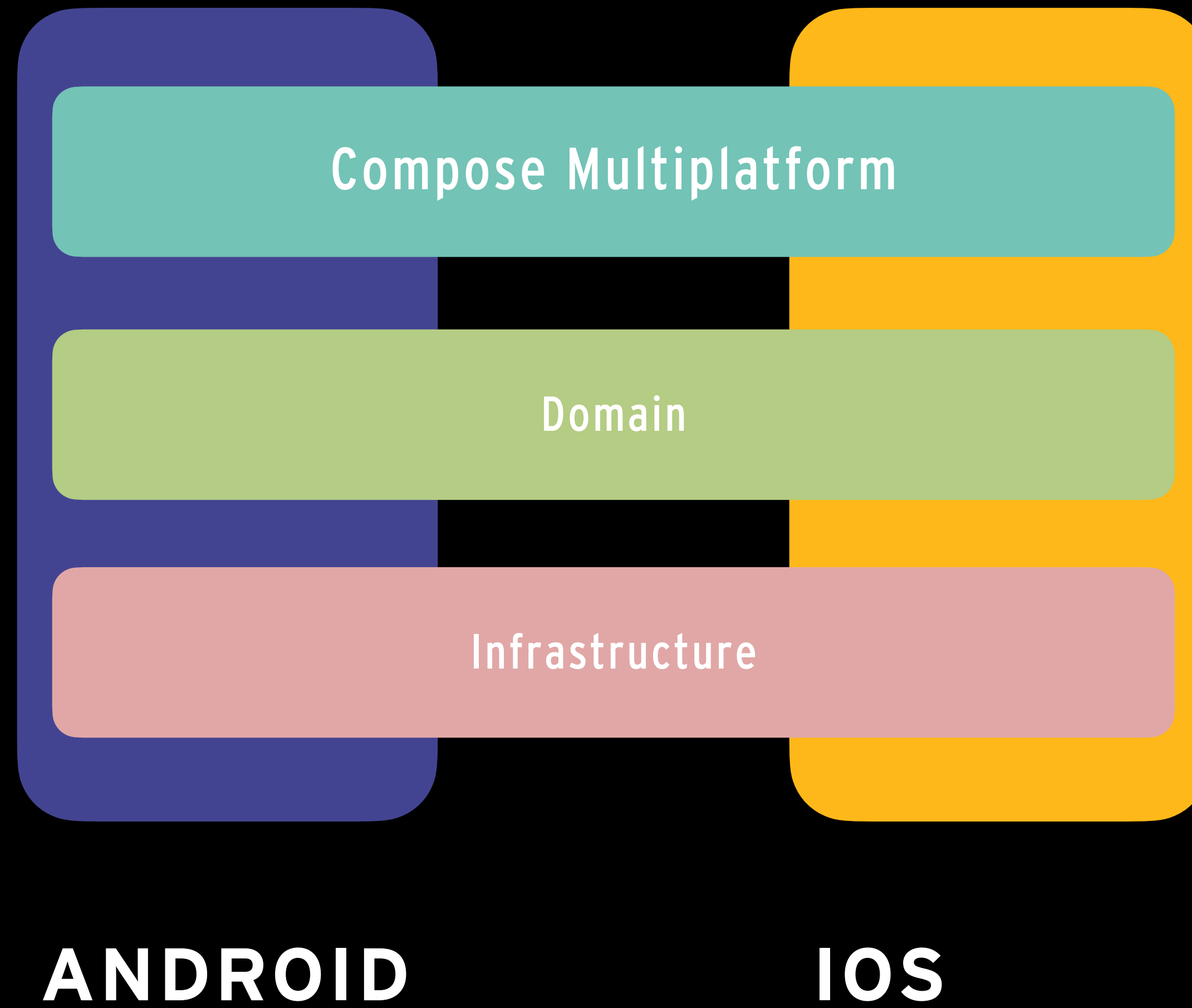
**IOS**

# WHAT IS CMP - KOTLIN MULTIPLATFORM

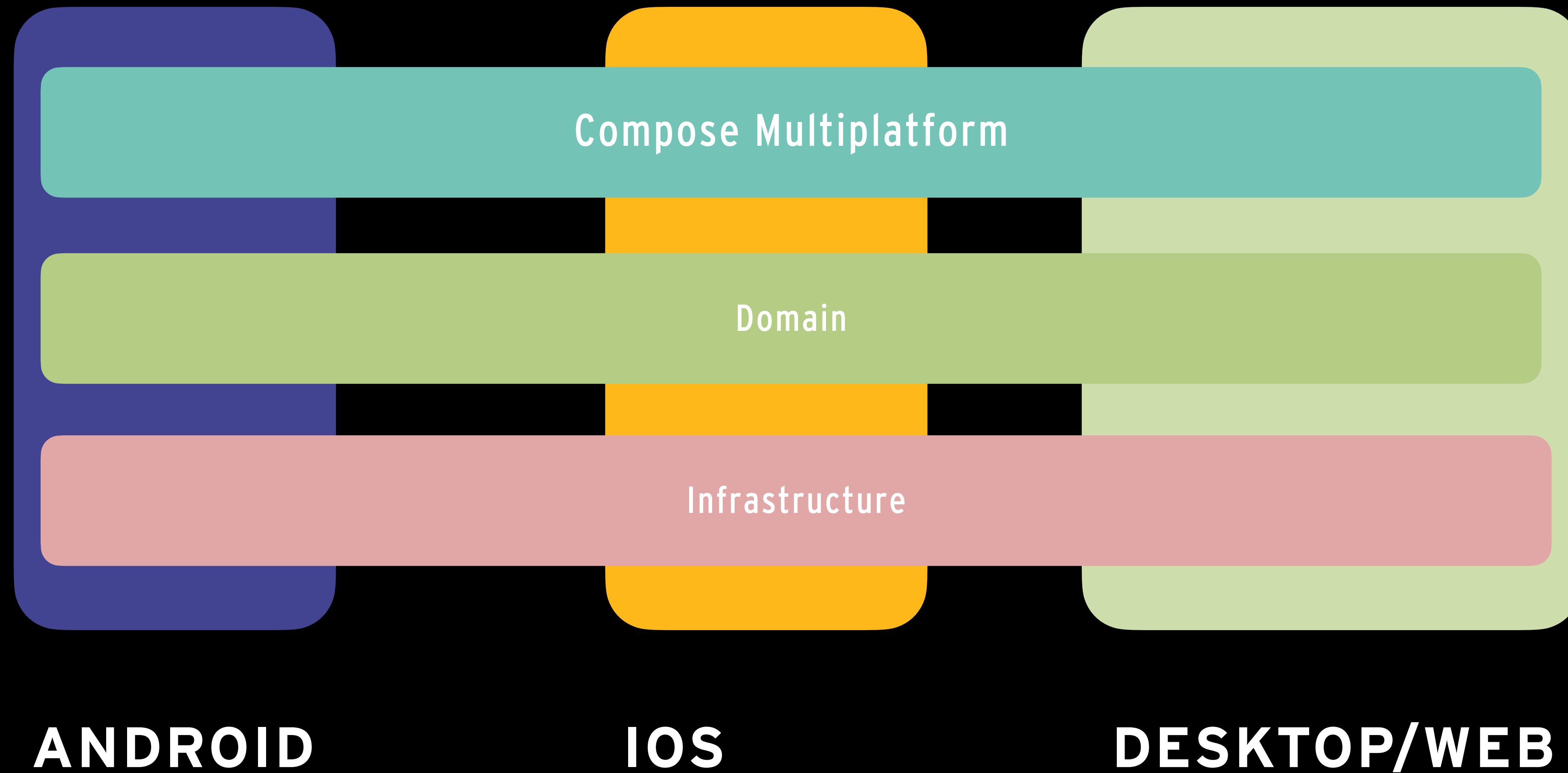




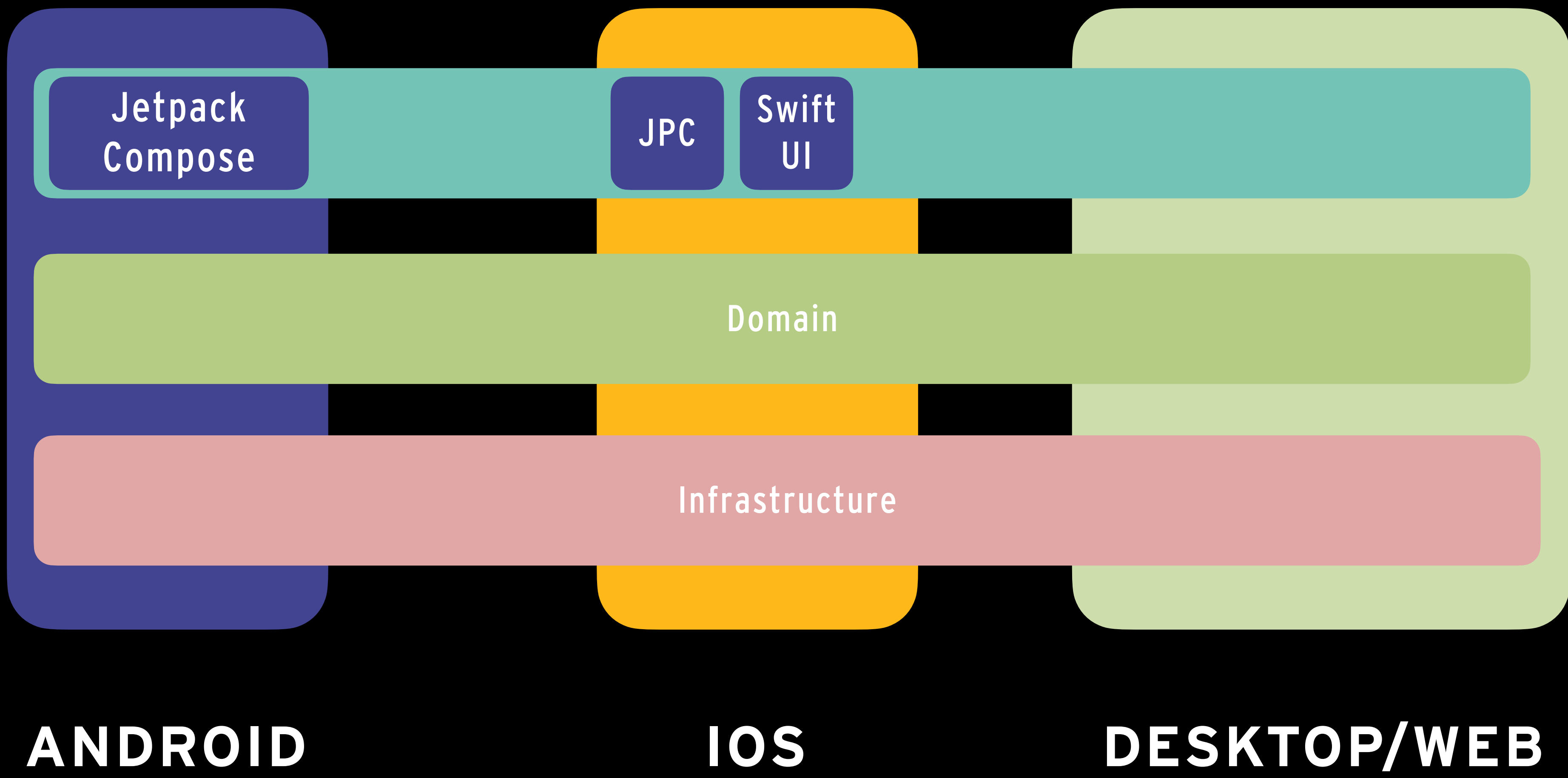
# WHAT IS CMP - WITH COMPOSE MULTIPLATFORM



# WHAT IS CMP - WITH COMPOSE MULTIPLATFORM.. EVEN MORE ..

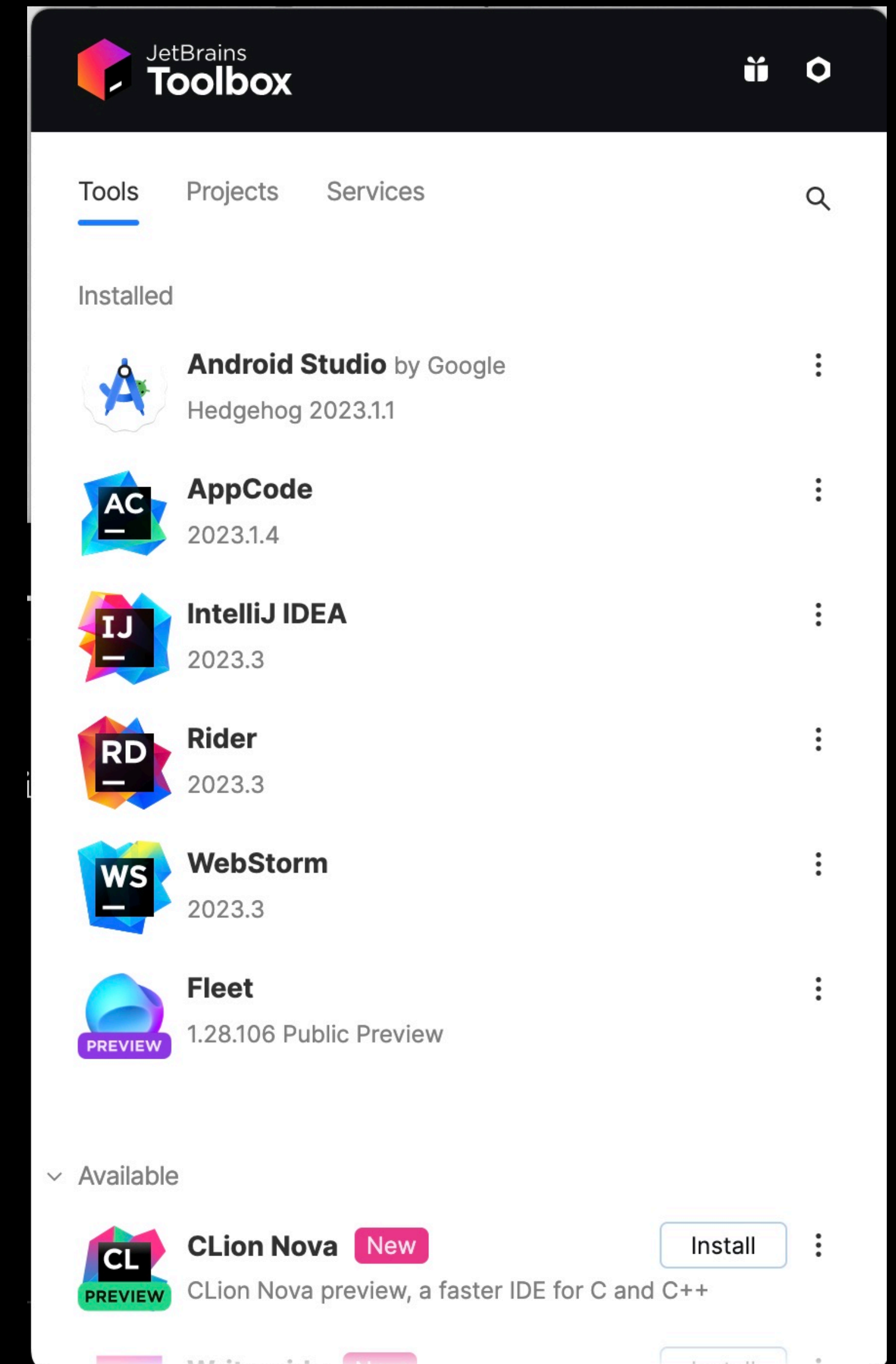
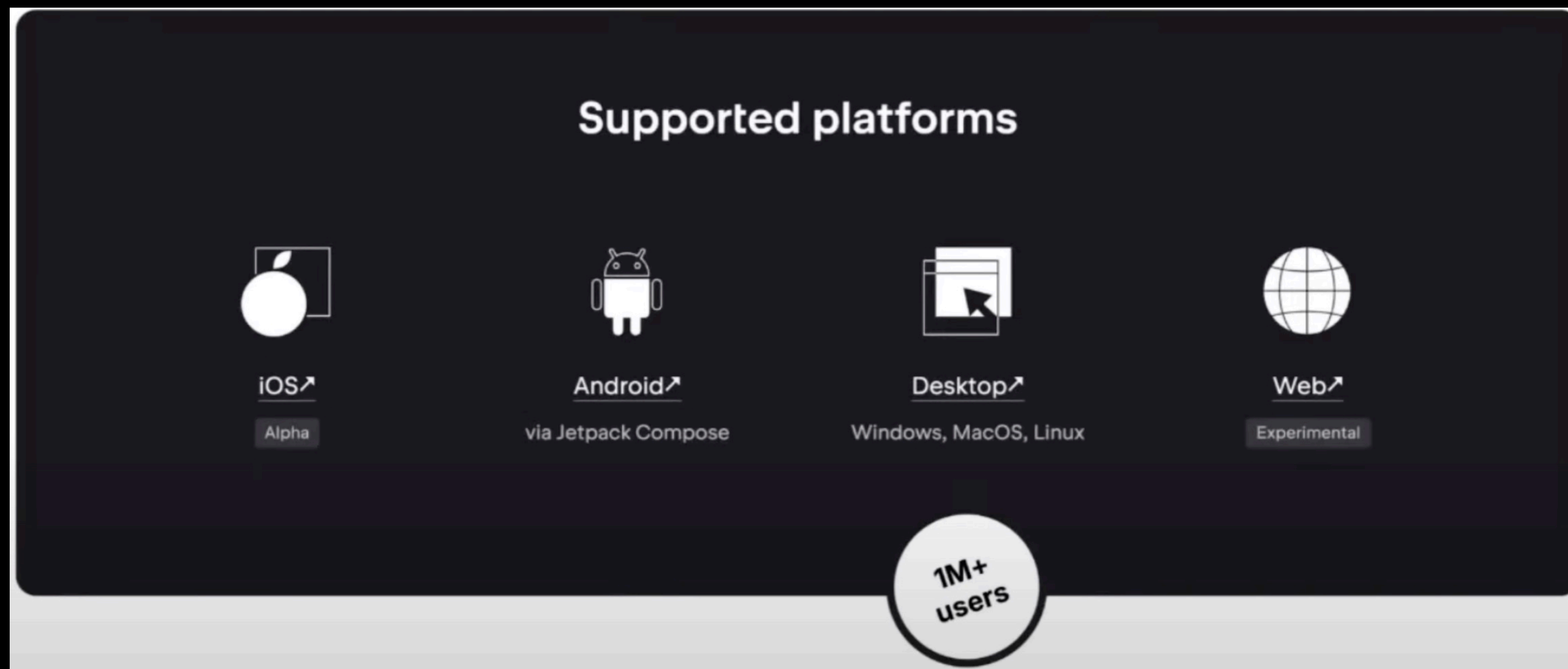


# WHAT IS CMP - WITH COMPOSE MULTIPLATFORM

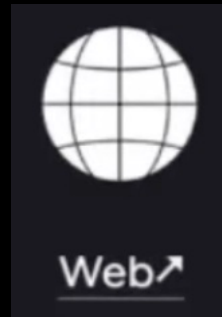


# WHAT IS CMP - SUPPORTED PLATFORMS

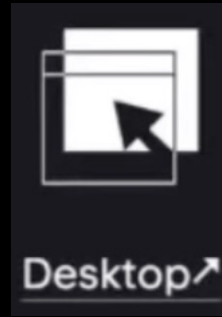
- IOS - alpha
- Android - via Jetpack Compose
- Desktop - Windows, MacOS, Linux
  - JetBrains toolbox
- Web - Experimental (WSM)



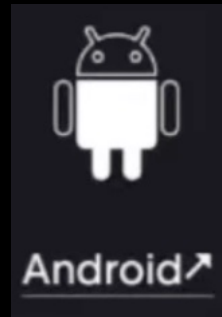
# WHAT IS CMP - TIMELINES



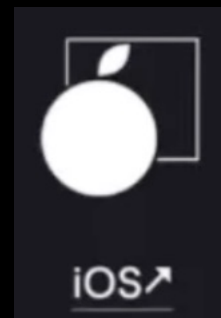
Experimental



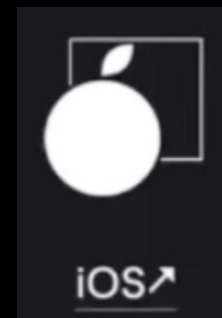
Stable



Stable



Alpha



iOS Beta



## WHAT IS CMP - INTEROPERABILITY

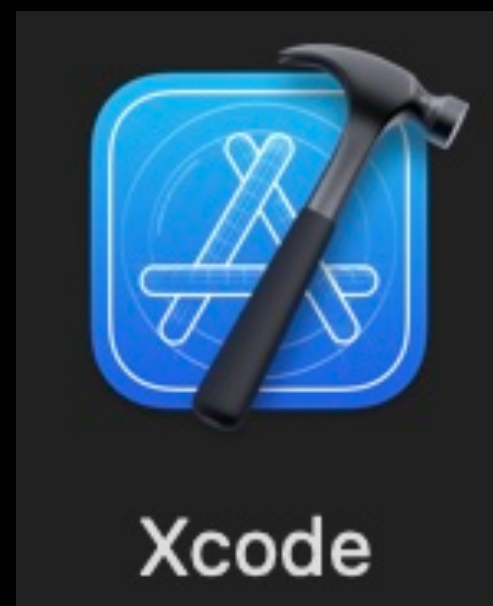
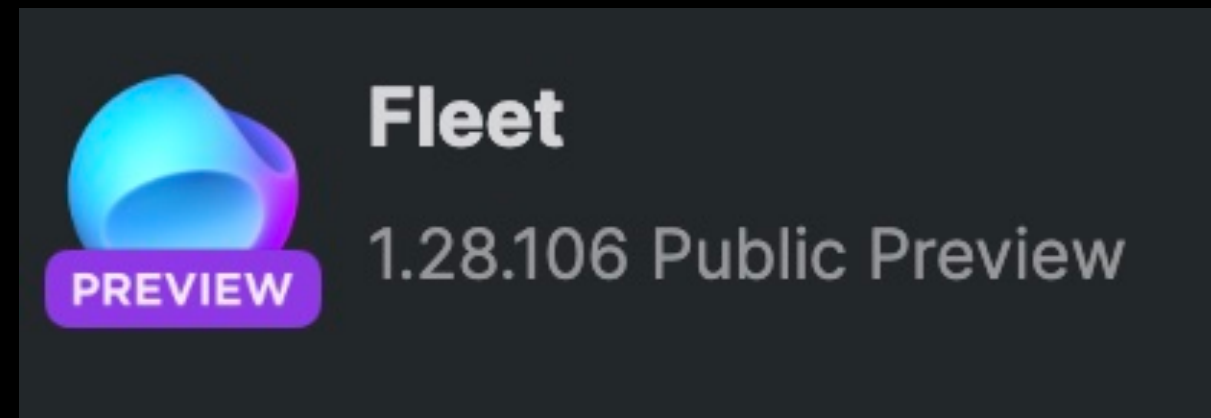
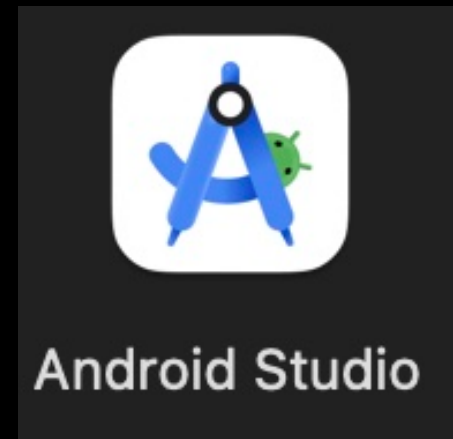
- Use Compose Multiplatform inside SwiftUI application
- Use Compose Multiplatform inside UIKit application
- Use UIKit inside Compose Multiplatform
- Use SwiftUI inside Compose Multiplatform
  
- Its flexible!

# GETTING STARTED

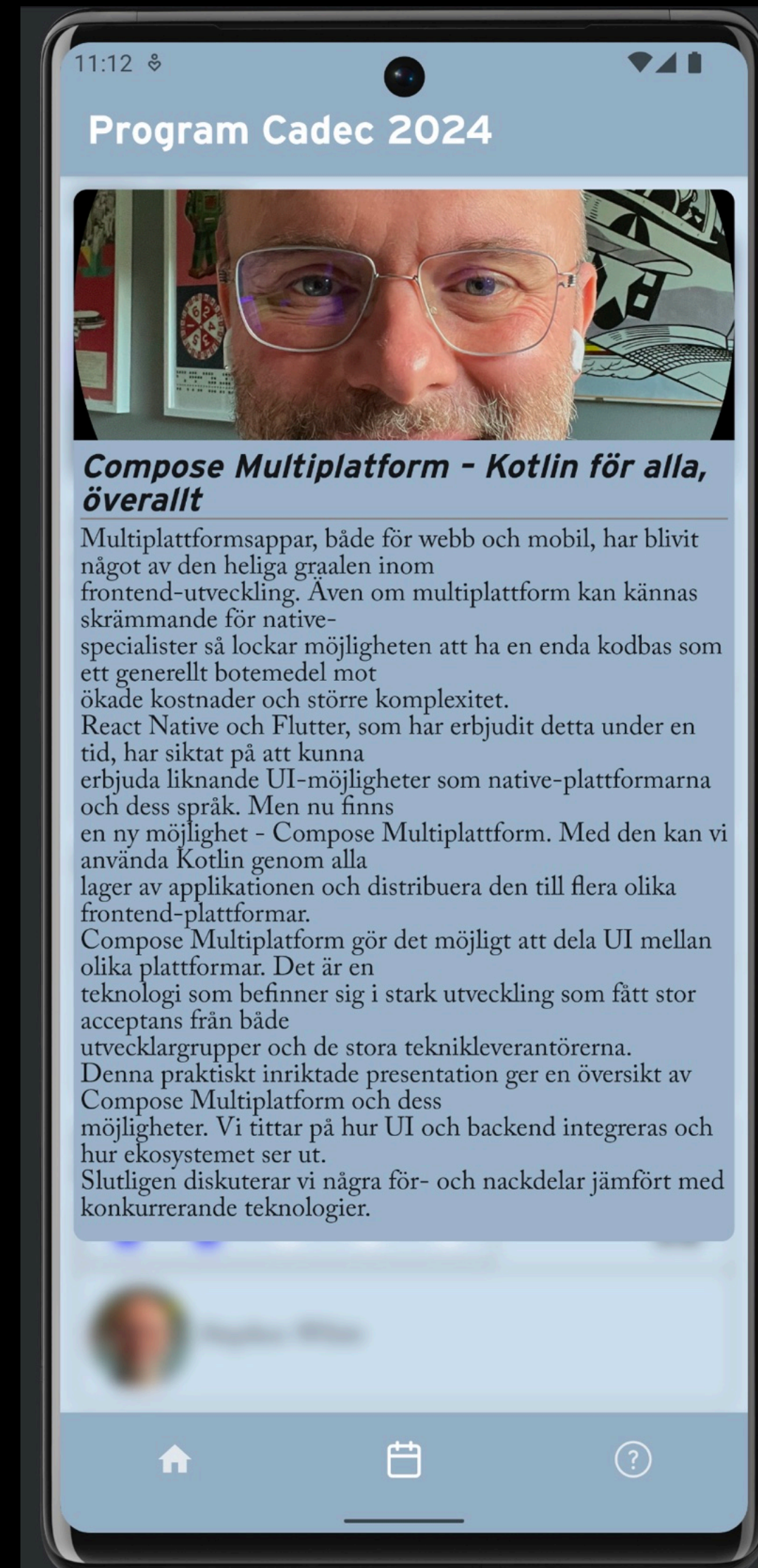


# GETTING STARTED - DEVELOPER TOOLS

## IDE'S

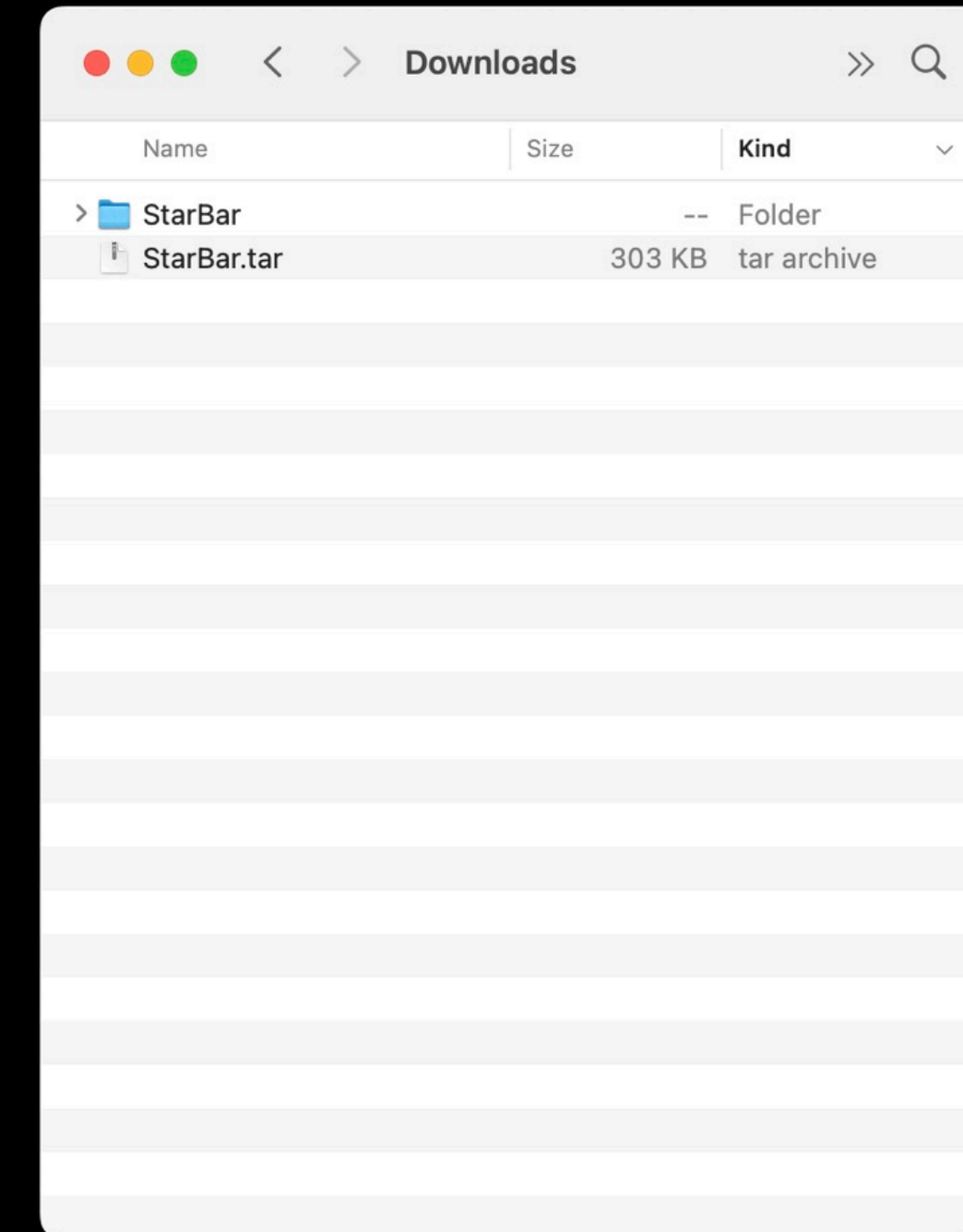
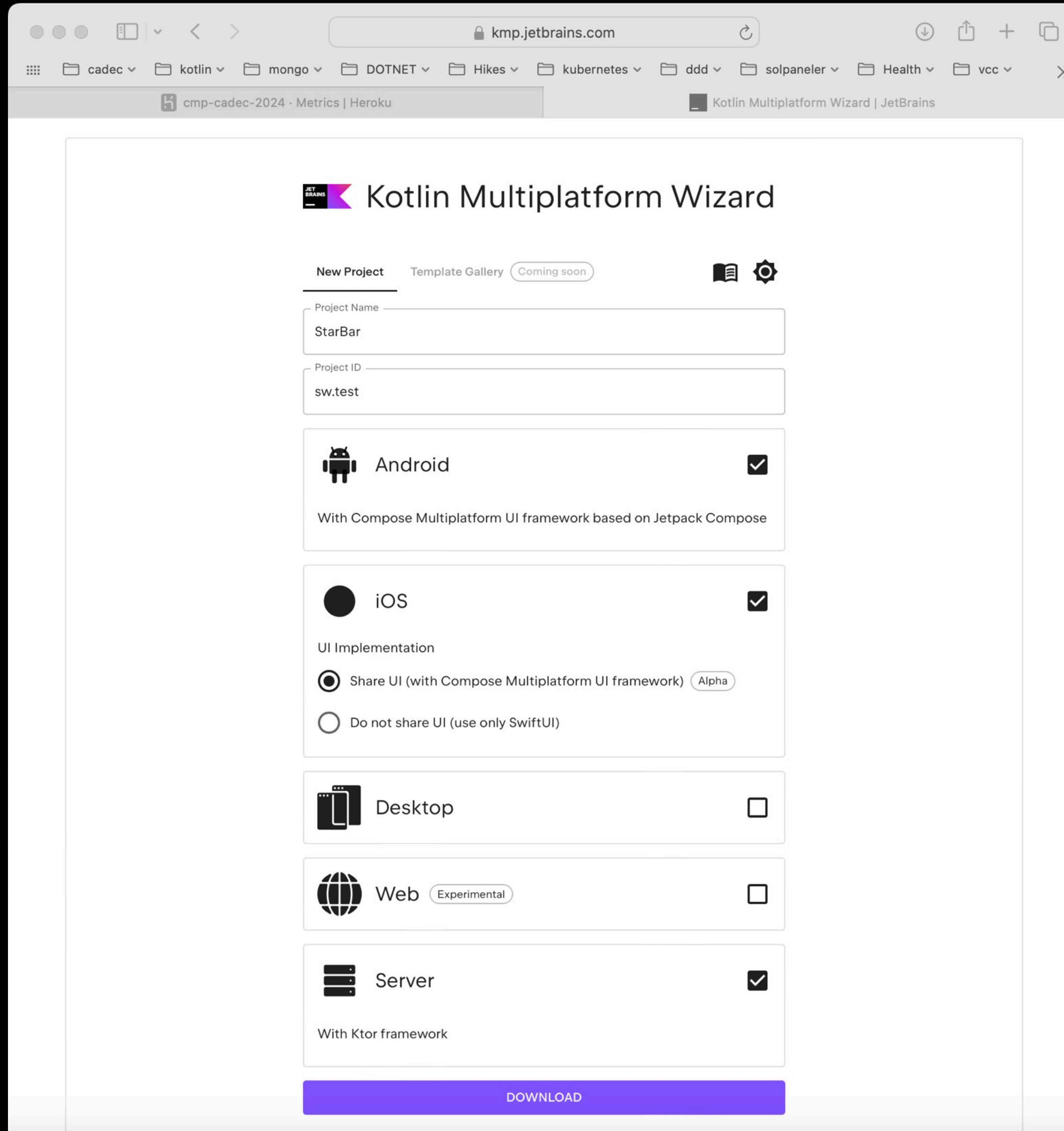


## SIMULATORS







# GETTING STARTED - CMP WIZARD

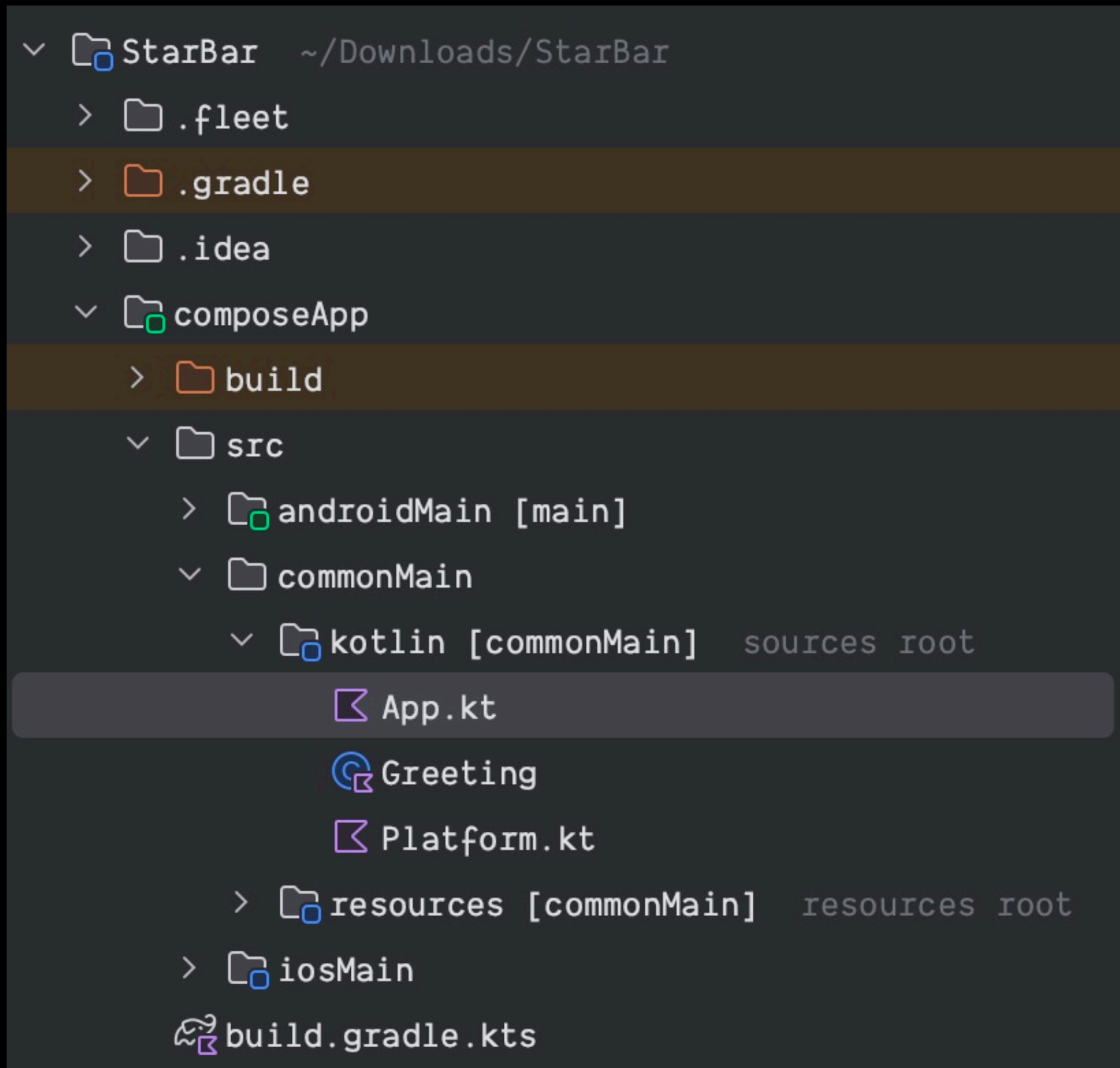


# GETTING STARTED - KDOCTOR

```
maitriyogin/kotlin/cmp-cadec-2024 via  v20.9.0 via  3.1.0
→ kdoctor
Environment diagnose (to see all details, use -v option):
[✓] Operation System
[✓] Java
[✓] Android Studio
[✓] Xcode
[✓] CocoaPods

Conclusion:
  ✓ Your operation system is ready for Kotlin Multiplatform Mobile Development!
```

# GETTING STARTED - CODE STRUCTURE



```
@OptIn(ExperimentalResourceApi::class)
@Composable
fun App() {
    MaterialTheme {
        var greetingText : String by remember { mutableStateOf( value: "Hello World!") }
        var showImage : Boolean by remember { mutableStateOf( value: false) }
        Column(Modifier.fillMaxWidth(), horizontalAlignment = Alignment.CenterHorizontally) { this: ColumnScope
            Button(onClick = {
                greetingText = "Compose: ${Greeting().greet()}"
                showImage = !showImage
            }) { this: RowScope
                Text(greetingText)
            }
            AnimatedVisibility(showImage) { this: AnimatedVisibilityScope
                Image(
                    painterResource( res: "compose-multiplatform.xml"),
                    contentDescription: null
                )
            }
        }
    }
}
```



# GETTING STARTED - RUN THE APP

The image shows an IDE interface with the following components:

- Project Structure:** A tree view on the left showing the project layout. The 'composeApp' module is expanded, showing 'src' with sub-directories 'androidMain', 'commonMain', and 'iosMain'. The 'App.kt' file is selected under 'commonMain'.
- Code Editor:** The central pane displays the content of 'App.kt'. The code includes imports for Compose libraries, plugin configurations for Kotlin Multiplatform, Android, and JetBrains Compose, and target configurations for Android and iOS. It also shows dependencies for Compose runtime, foundation, material, and UI.
- Running Devices:** On the right, a virtual device is shown running the application. The screen displays the date 'Mon, Jan 15' and a dock with icons for Play Store, Gmail, Photos, YouTube, Phone, Messages, Chrome, and the Android logo. A search bar with the Google logo is at the bottom.



# GETTING STARTED - OPEN AND RUN THE APP

The image shows an IDE interface with three main panes:

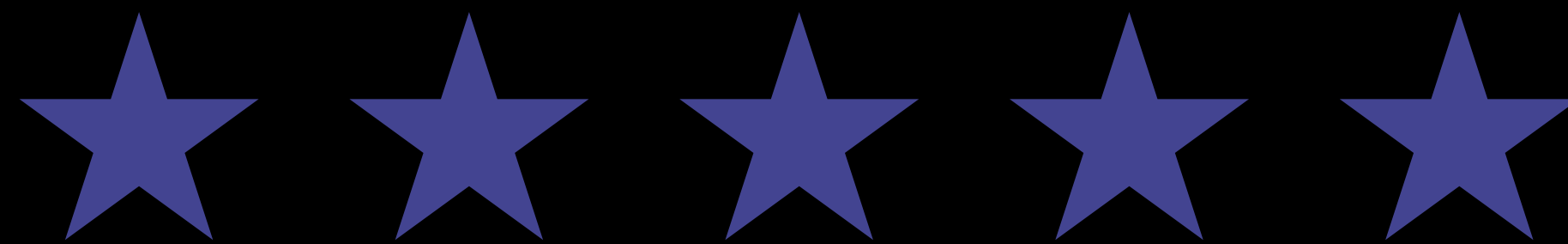
- Project Structure (Left):** Shows a project named "StarBar" with a "composeApp" module. The "App.kt" file is selected under "src > commonMain > kotlin".
- Code Editor (Middle):** Displays the content of "App.kt". The code includes imports, an annotation, a Composable function, a MaterialTheme, and UI components like a Button and an Image. The "AnimatedVisibility" function is highlighted in blue.
- Running Devices (Right):** Shows a virtual Android device running the app. The screen displays the date "Thu, Jan 18", a "Click me!" button, and a "Compose: \$greeting" text. The home screen also shows icons for Gmail, Photos, YouTube, Phone, Messages, Chrome, and the Android logo.

```
1 > import ...
17
18 @OptIn(ExperimentalResourceApi::class)
19 @Composable
20 fun App() {
21     MaterialTheme {
22         var showContent : Boolean by remember { mutableStateOf( value: false) }
23         val greeting : String = remember { Greeting().greet() }
24         Column(Modifier.fillMaxWidth(), horizontalAlignment = Alignment.CenterHorizontally) { th
25             Button(onClick = { showContent = !showContent }) { this: RowScope
26                 Text( text: "Click me!")
27             }
28             AnimatedVisibility(showContent) { this: AnimatedVisibilityScope
29                 Column(Modifier.fillMaxWidth(), horizontalAlignment = Alignment.CenterHorizonta
30                     Image(painterResource( res: "compose-multiplatform.xml"), contentDescription:
31                     Text( text: "Compose: $greeting")
32                 }
33             }
34         }
35     }
36 }
```

**DEMO**



# DEMO RECAP



## STAR BAR - RECAP

- Look and feel - Android
  - Animations
  - Blurs
- Patterns
  - Bottom Sheet
  - Navigation
    - » Stack
    - » Tab
- Resources - Images,
  - ▷ Fonts
- Local Storage - Multiplatform Settings
- Networking - HTTP / WS - Ktor - client
- Release to app stores ...
  
- Used it in the Cadec App!!



# ONGOING DEVELOPMENT

# ONGOING DEVELOPMENT

Material / compose-cupertino

Look and Feel

Moko

View Models

Moko

Resource  
Management

Ktor

Networking

Voyager

Navigation

Kamel

Image Loading



SQLDelight / Realm / KStore / Datastore etc ..

Persistence



# ONGOING DEVELOPMENT - HELP

**JET BRAINS** English

# Compose Multiplatform

Develop stunning shared UIs for Android, iOS, desktop, and web.

[Get started](#)

Image Viewer  
http://localhost:3000

20. Mar Mountain K2

20. Mar Mountain K2

3:03  
Note  
K2, at 8,611 meters above sea level, is the second-highest mountain on Earth, after Mount Everest.... (see more)

Related memories

Place

Mountain K2

Kina The Calico

Blue City

Tokyo Skytree

Taranaki

18:33

kotlinlang

Jump to or search...

Catch Up 7 new

Threads Caught up

Later 0 items

Drafts & Se 0 items

Unreads

- # android
- # announcements
- # compose
- # feed
- # getting-started
- # multiplatform
- # server

+ Add channel

Direct Messages

- Stephen White (you)
- + Start a new message

Home DMs Activity



# ONGOING DEVELOPMENT - REPLACEMENTS

- Awesome Kotlin Multiplatform
- <https://github.com/terrakok/kmp-awesome>
- Networking - ktor
- Resource Management - Mokko
- Navigation - Voyager
- Image Loading - Kamel
- Look and Feel - Material UI
- Persistence -
  - ▷ KStore
  - Realm
  - Multiplatform settings

The screenshot shows the GitHub repository page for 'kmp-awesome' by user 'mirzemehdi'. The repository is public and has 61 watchers. It is currently on the 'master' branch with 11 tags. The commit history shows a recent update to README.MD (#82) by mirzemehdi, dated last week. The repository contains files like 'img', '.gitignore', and 'README.MD'. The README content is visible, featuring the title 'Awesome Kotlin Multiplatform' and a list of categories: UI (Compose Multiplatform), Presentation, Business / Domain, and Data / Core. It also includes a description of Kotlin Multiplatform technology, a note about supporting iOS and Android targets, and a list of resources such as Website, KMP Web Wizard, Compose Multiplatform Wizard, Documentation, and Blog.

**FINAL THOUGHTS**

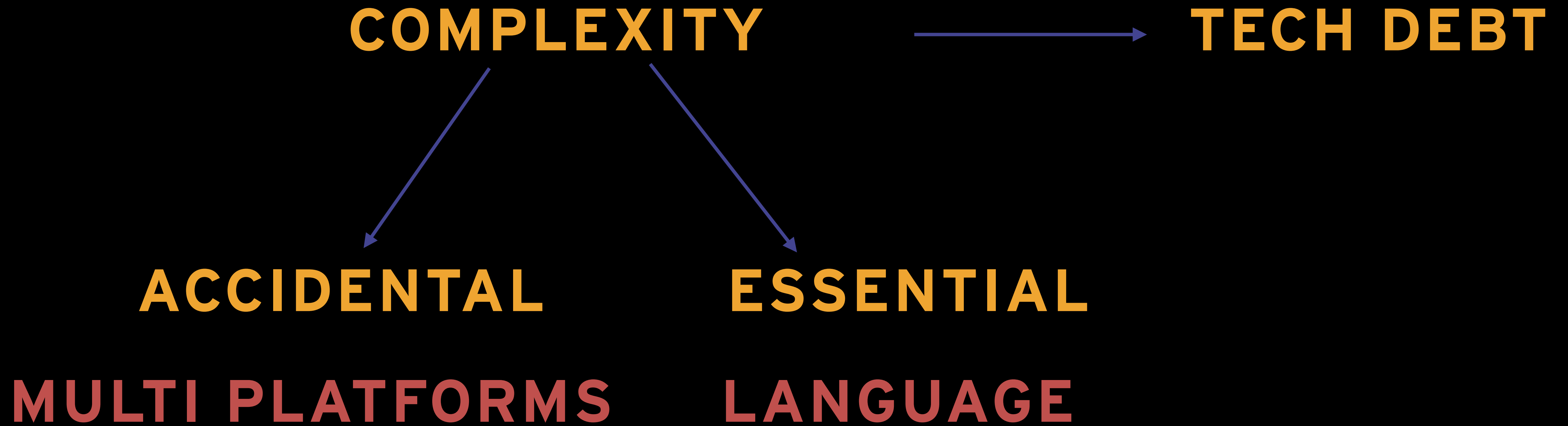
**FINAL THOUGHTS**

**EARLY DAYS ...**

**TARGET ANDROID**

**TRANSITION TO IOS**

# FINAL THOUGHTS



# DISSONANCE

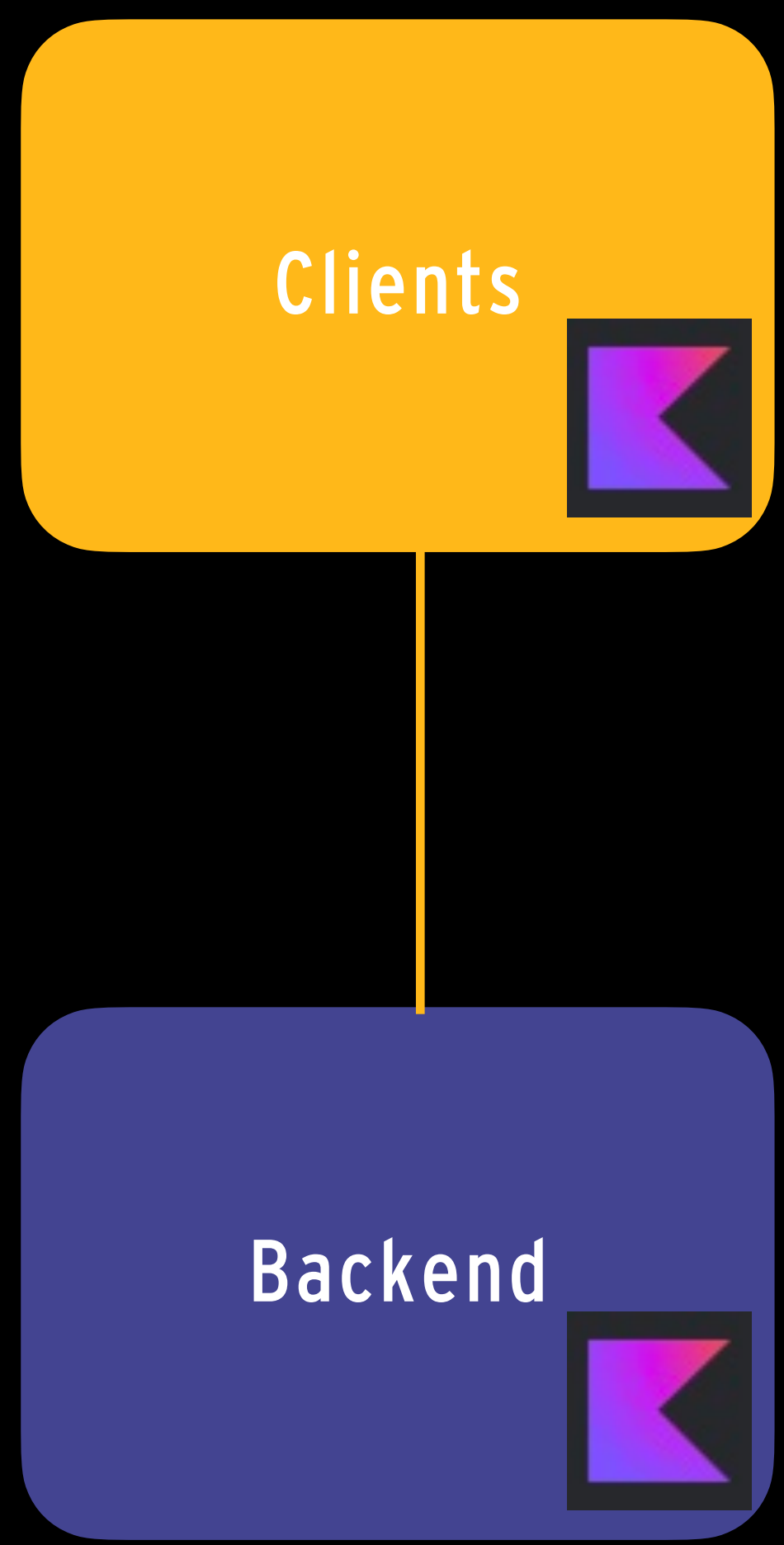




# FINAL THOUGHTS



# FINAL THOUGHTS



## | SUMMARY

- Star Bar - App
- Cadec - App
- Almost production ready!
- Jetpack compose
- Kotlin
- Great community

**END**