Kotlin Multiplatform

Shareable code base

Nattapon Pondongnok

Contents

- What is Kotlin Multiplatform

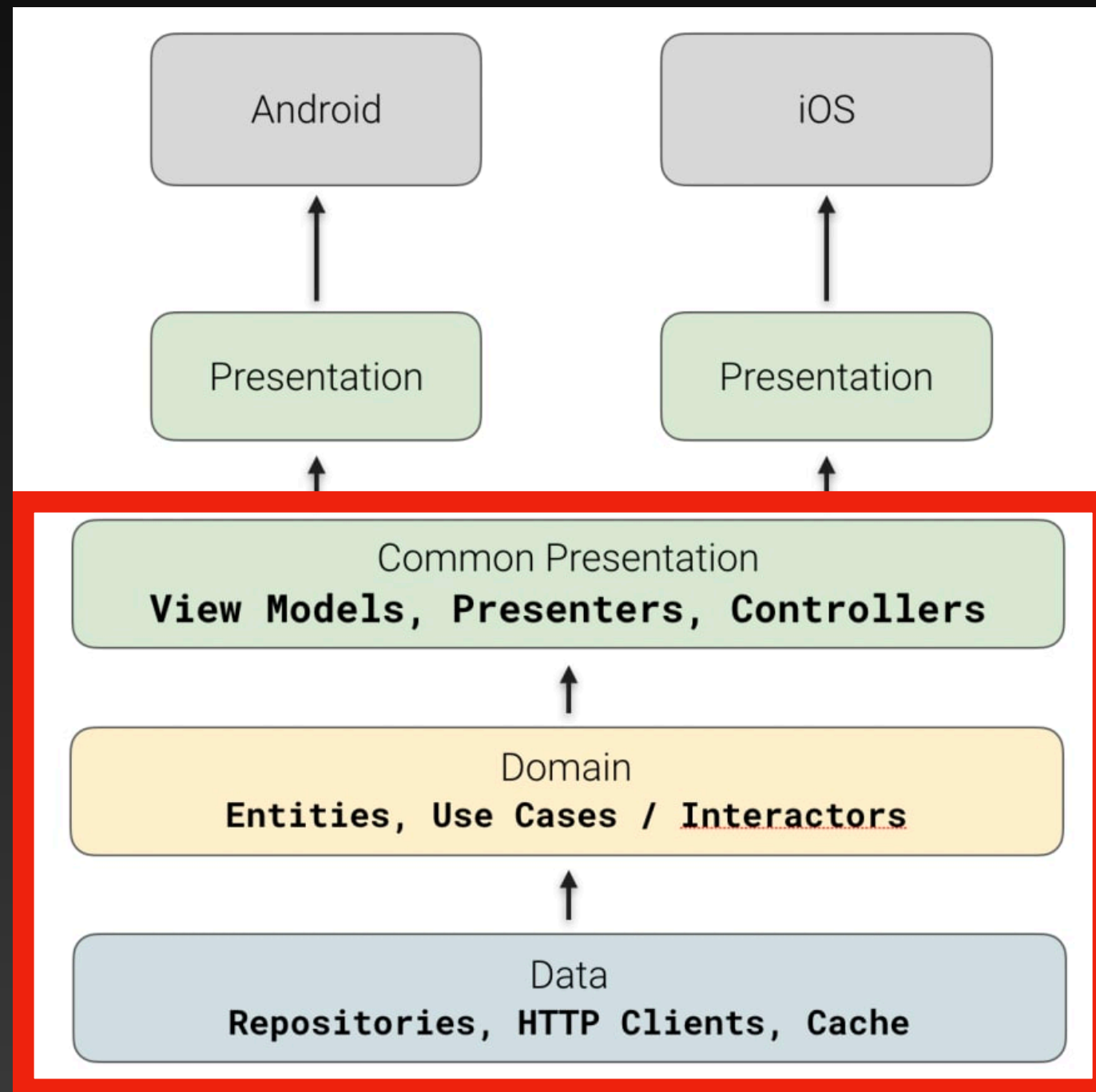- Demo Simple App

- Conclusion

Kotlin Multiplatform

- Share business logic with Kotlin language

- Can integrate to exiting Native App

- Easy to maintenance

- Low learning curve

- Alpha stage

Project Structure

- Main business logic in **commonMain**

- Specific platform code in **androidMain** and **iosMain**

- Main unit test in **commonTest**

> 📁 androidApp
> 📁 build
> 📁 gradle
> 📁 iosApp
∨ 📁 shared
  > 📁 .gradle
  > 📁 build
  ∨ 📁 src
    > 📁 androidMain
    > 📁 commonMain
    > 📁 commonTest
    > 📁 iosMain
    📄 build.gradle.kts
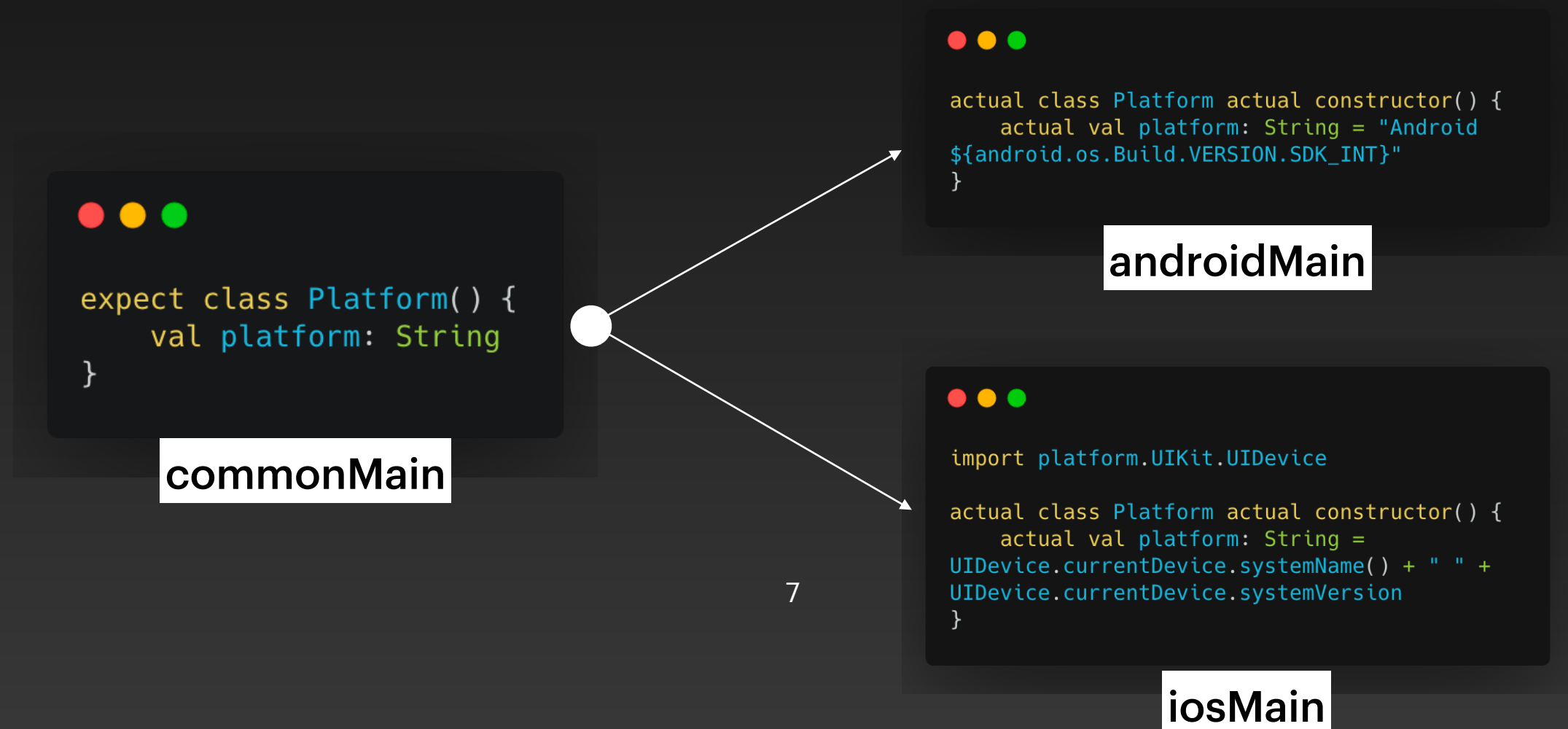   📄 shared.podspec

4

What is can share

How it can share

- For **android** it just **Kotlin library**

- For **iOS** it compile to **Objective-C Library**

How it can share

- Specific code platform use **expect** and **actual** keyword

```
expect class Platform() {
    val platform: String
}
```
commonMain

```
actual class Platform actual constructor() {
    actual val platform: String = "Android
${android.os.Build.VERSION.SDK_INT}"
}
```
androidMain

```
import platform.UIKit.UIDevice

actual class Platform actual constructor() {
    actual val platform: String =
UIDevice.currentDevice.systemName() + " " +
UIDevice.currentDevice.systemVersion
}
```
iosMain

7

Requirement for use

- Android Studio
  - with Kotlin Multiplatform Plugin

- Xcode 13
  - Cocoapods

Demo

Conclusion