

Maximiliano Firtman
@firt firt.dev

Introduction to Kotlin and Android App Development



firtman.github.io/intro-kotlin

Frontend *Masters*

mobile+web developer & trainer

A grayscale world map with the country of Argentina highlighted in a bright red color. The map shows the outlines of continents and countries.

Argentina



android



Maximiliano Firtman
@firt firt.dev

- Published 20+ apps
- Java in 2012
- Kotlin in 2019

Let's Start!



What we'll cover

Basics of Kotlin

Android Studio

Testing and Debugging

UI with JetPack Compose

Screen Navigation

Working with Data

Web Services

Web App

Pre-requisites

firtman.github.io/intro-kotlin

Questions?

Frontend Map

Native Clients

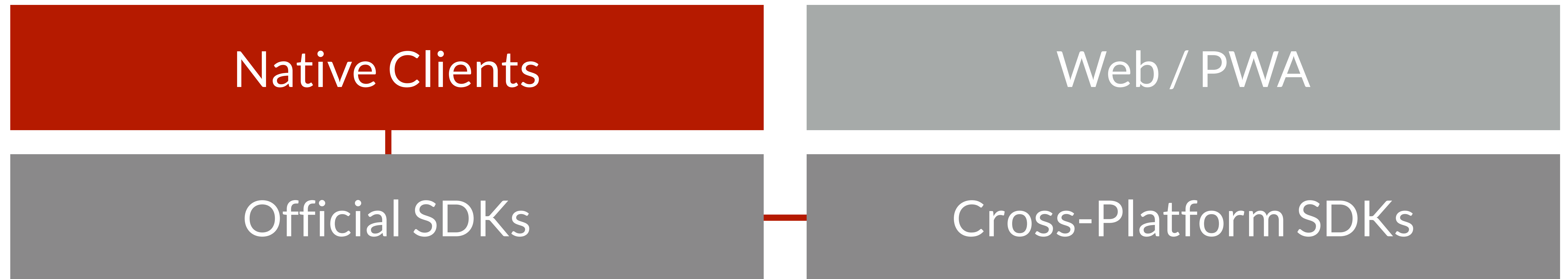
Web / PWA

Frontend Map

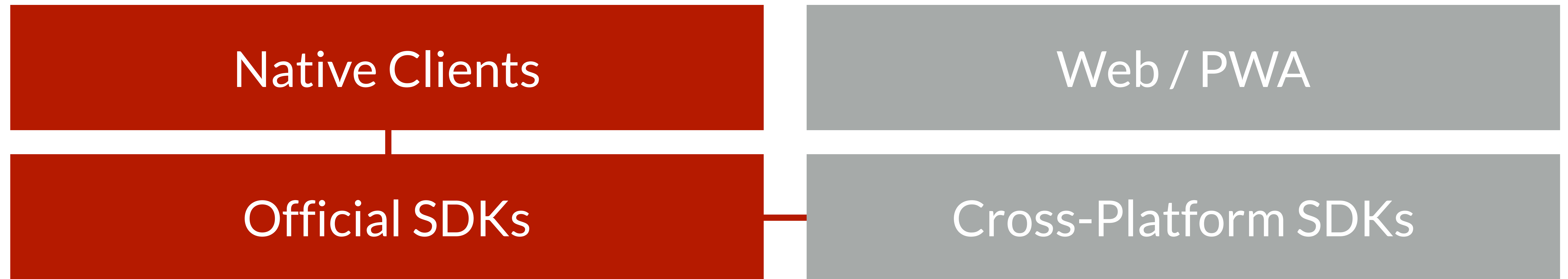
Native Clients

Web / PWA

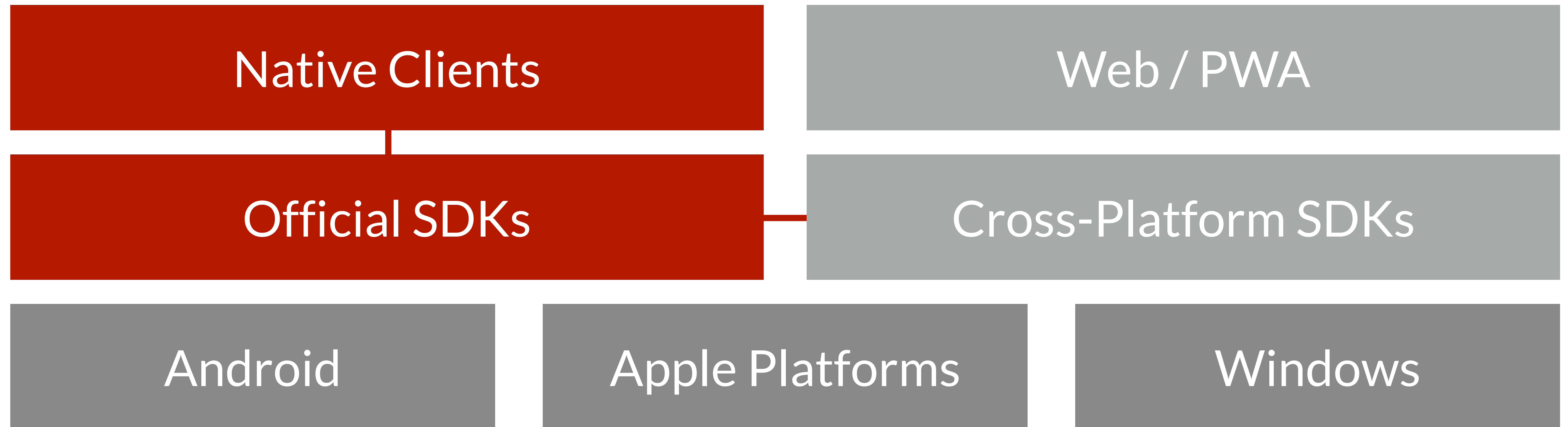
Frontend Map



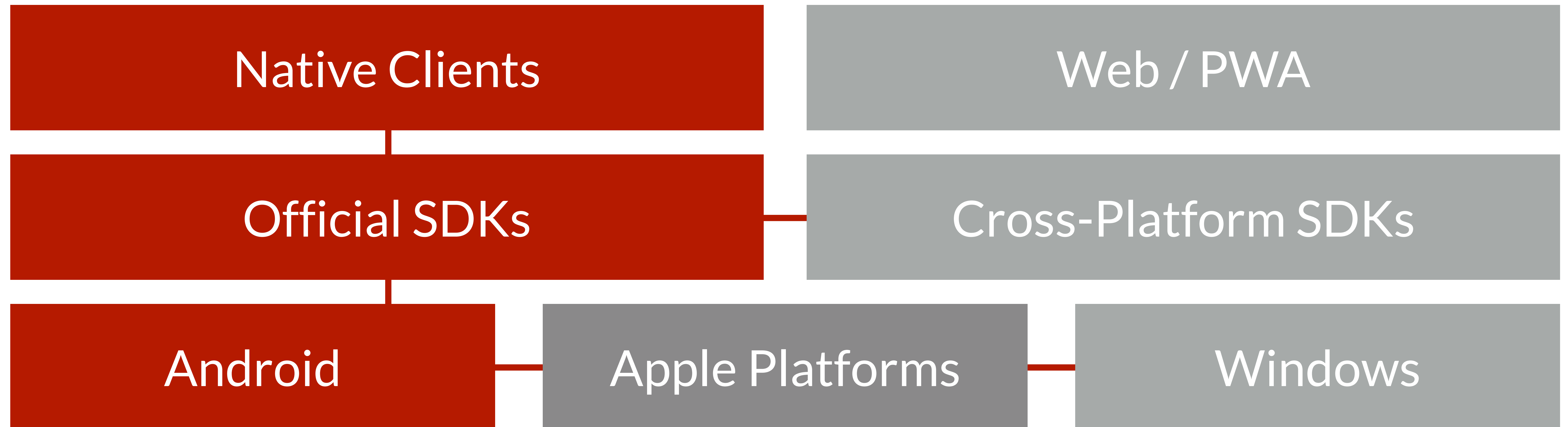
Frontend Map



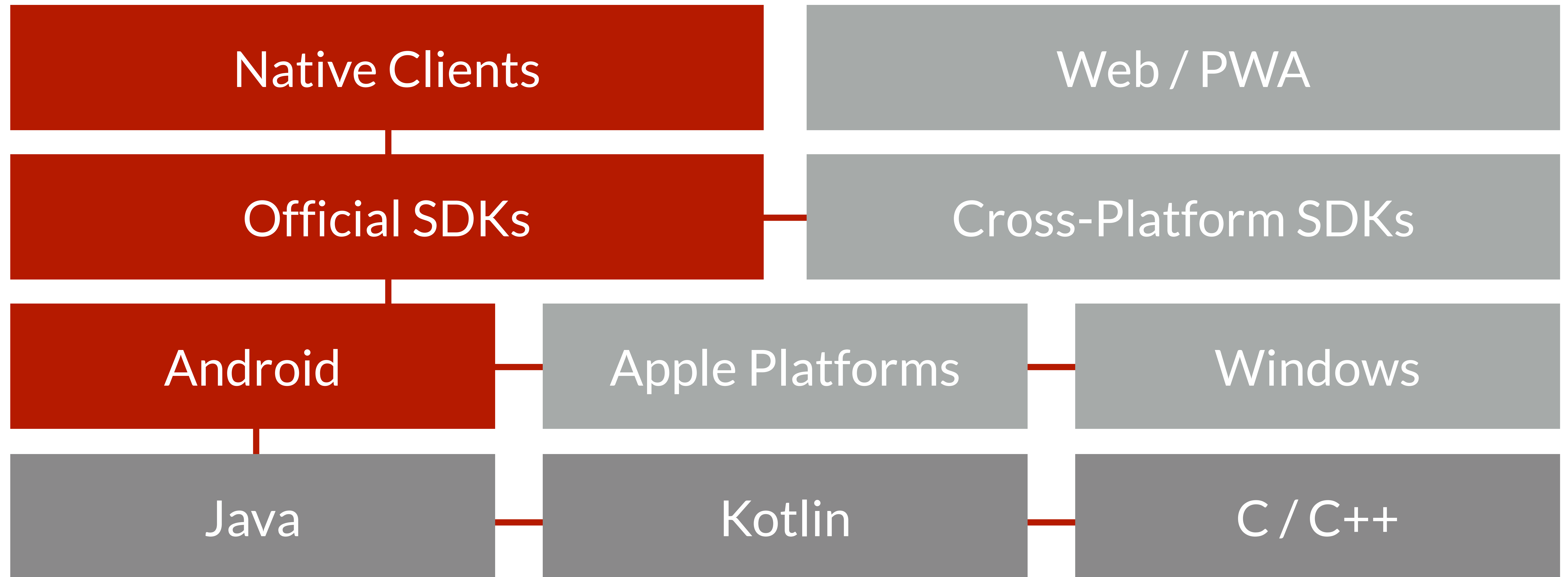
Frontend Map



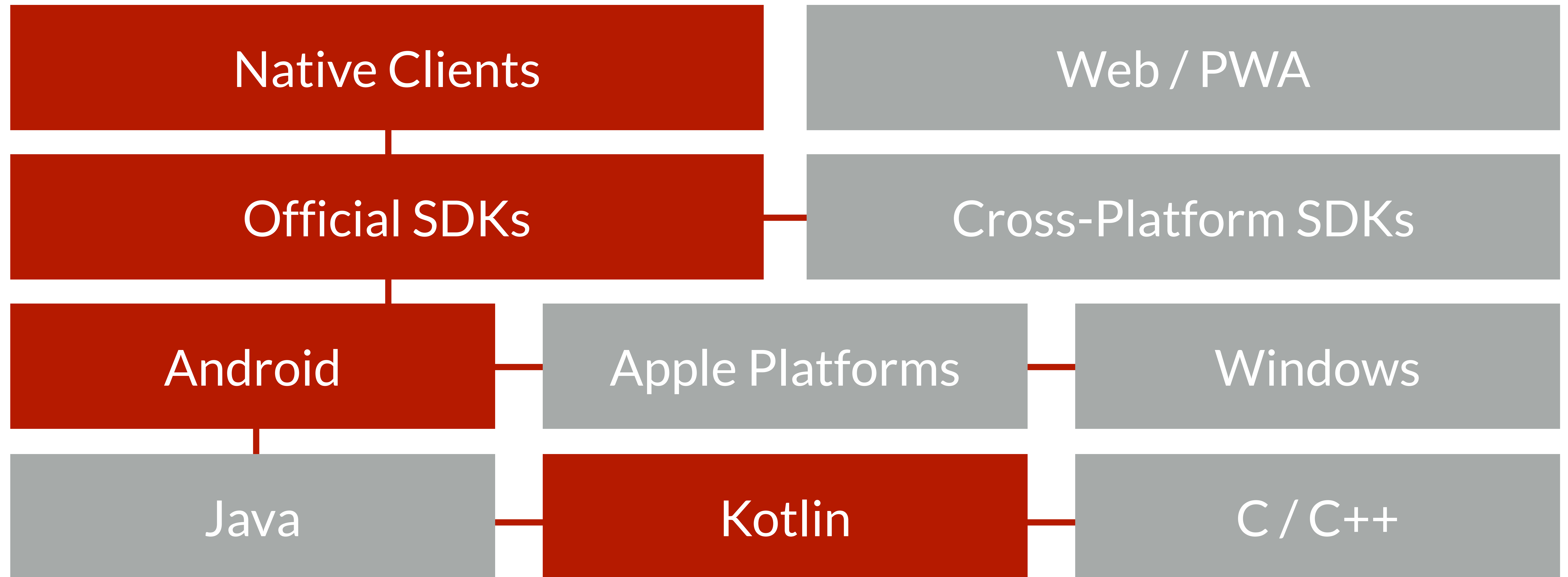
Frontend Map



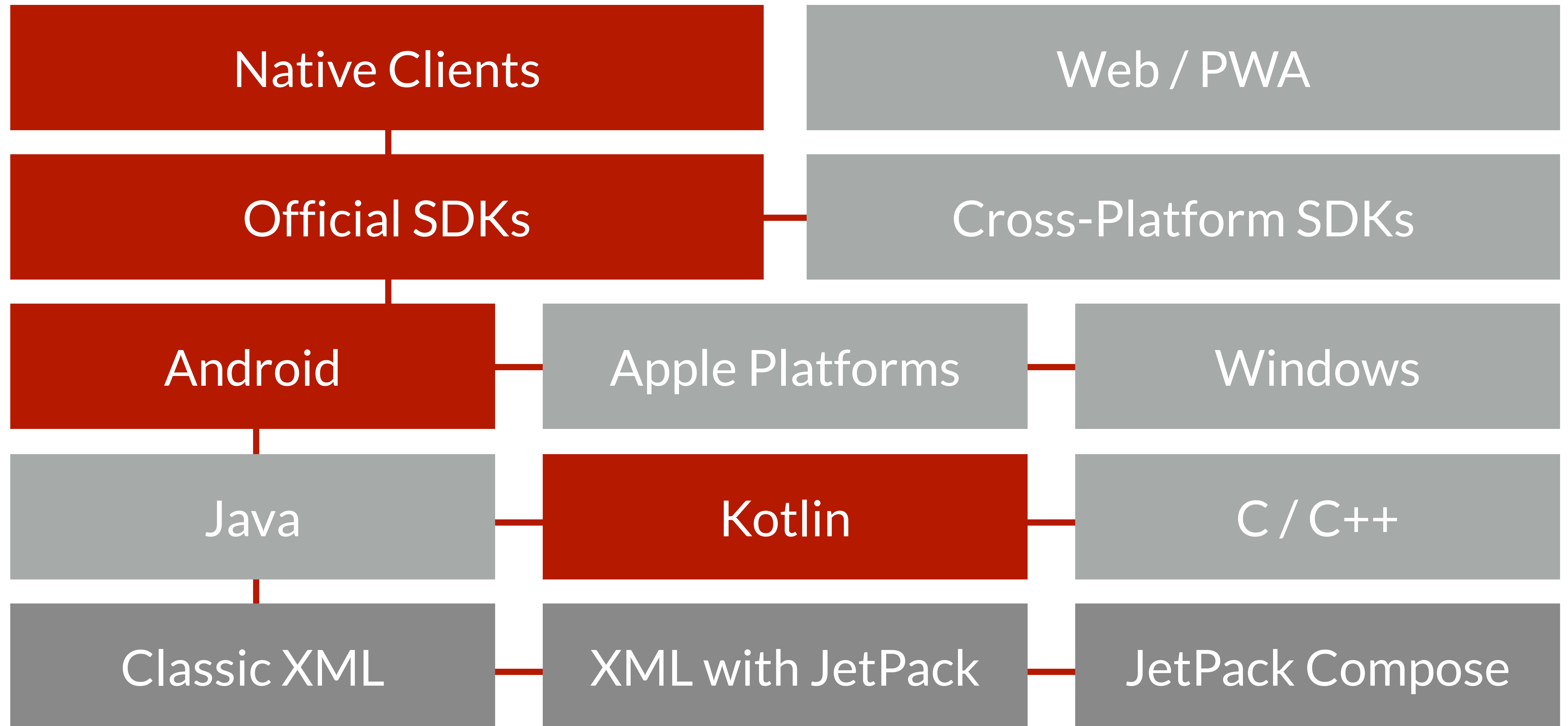
Frontend Map



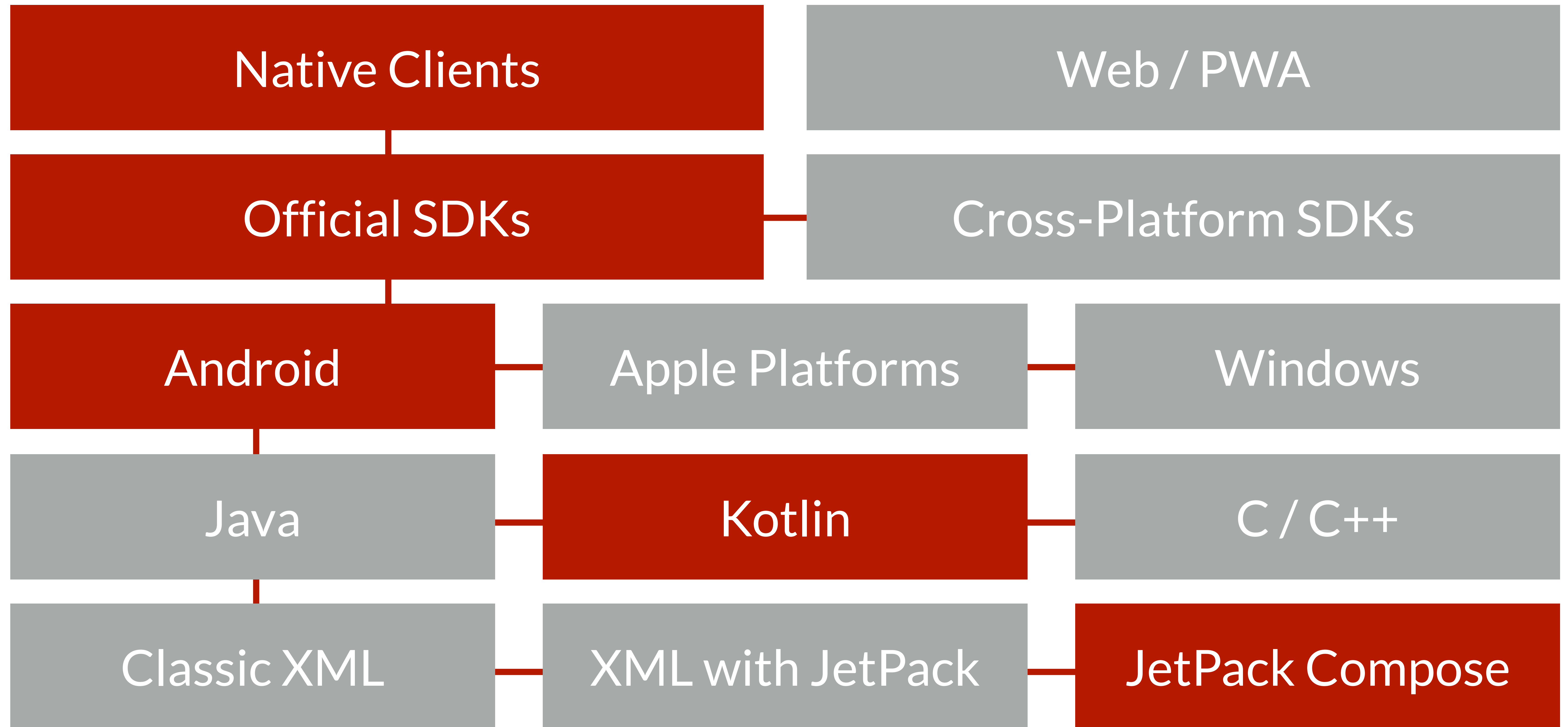
Frontend Map



Frontend Map



Frontend Map





Our Project

- App from scratch
- Coffee Store
- JetPack Compose
- Material Design
- Download assets and coding help

Android Operating System

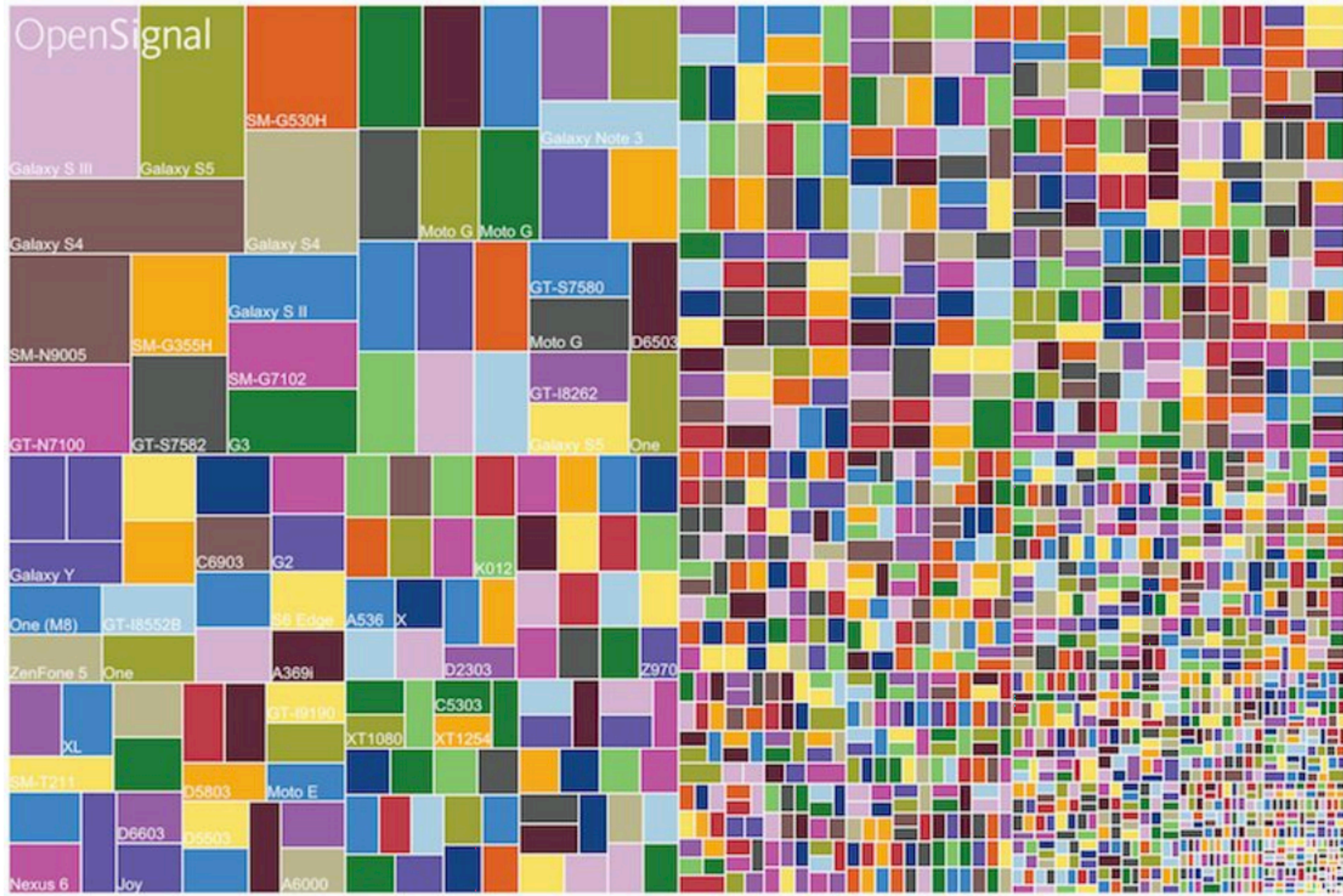
Vendor OEM Software	Proprietary
GMS (Google Mobile Services)	Proprietary
AOSP (Android Open Source Project)	Open Source

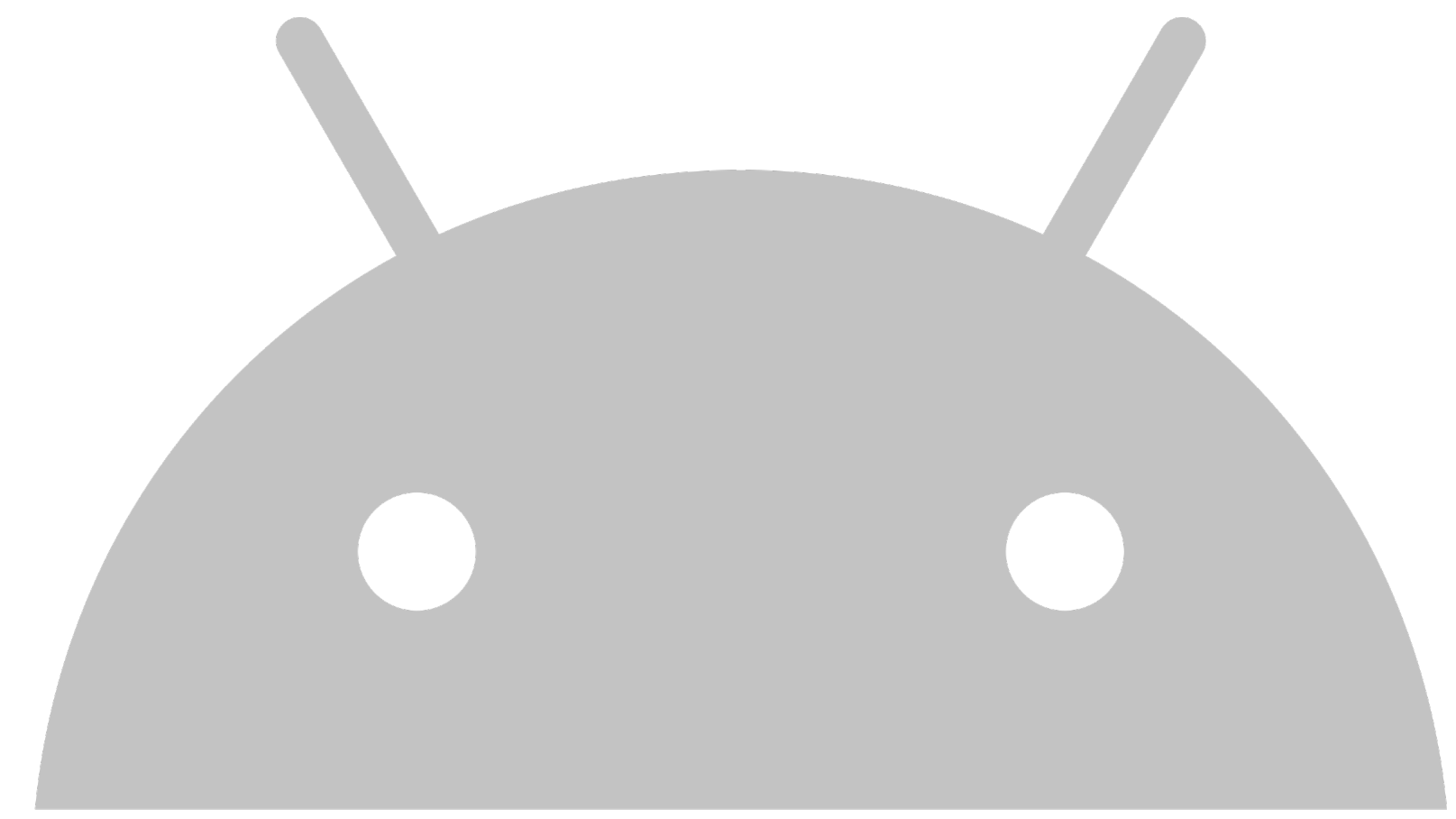
Android Operating System

Vendor OEM Software	Proprietary
AOSP (Android Open Source Project)	Open Source

No Gmail, Maps, Nice Camera, Chrome, Network Location Provider & more
Amazon FireOS, Huawei HMS

DEVICE FRAGMENTATION

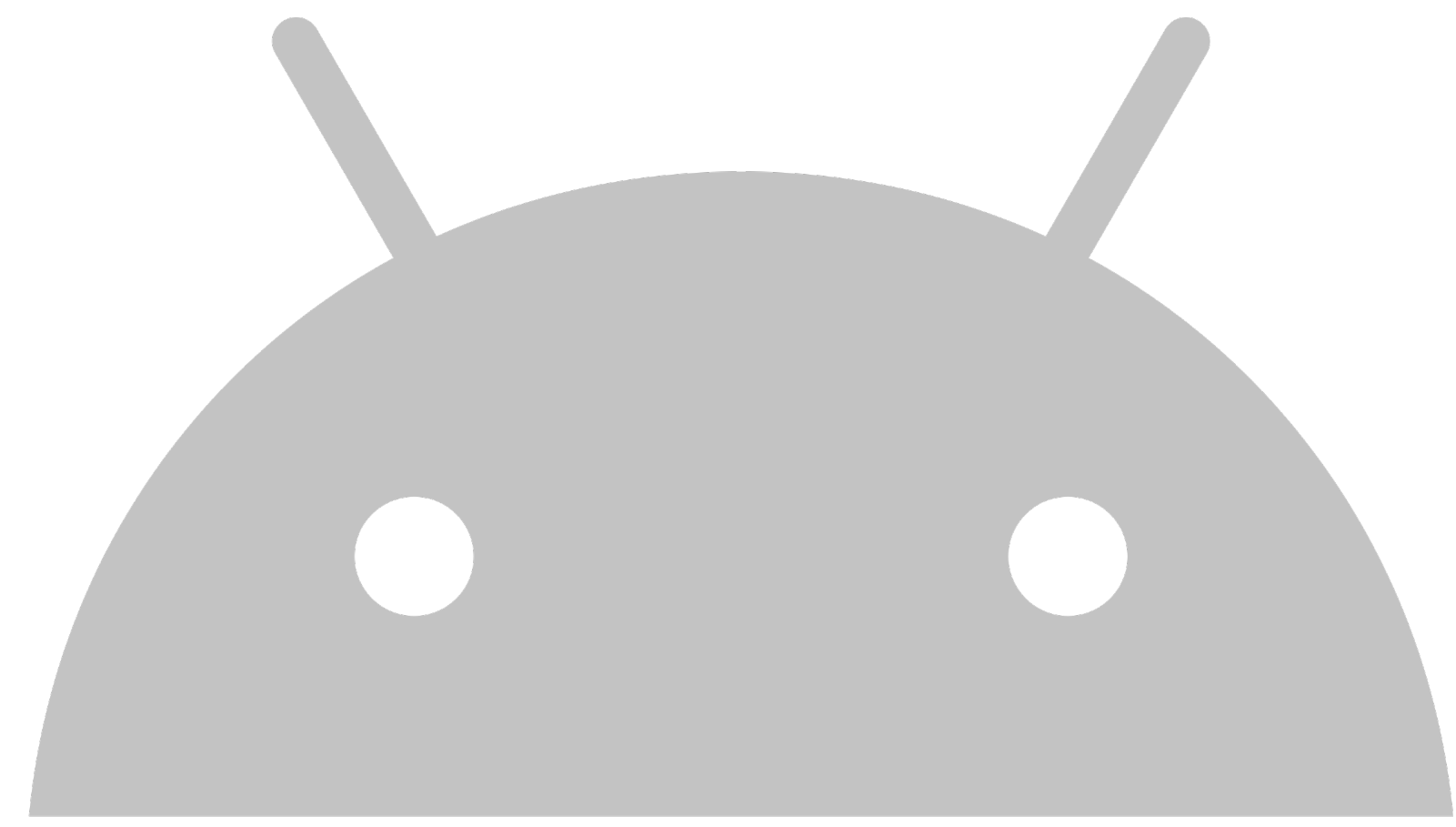




android



- **APK** - Android Package
- ZIP Package with:
 - compiled VM code
 - meta data
 - assets
 - signature
- Installable via:
 - Play Store **(AAB)** or other stores
 - Play for Enterprise
 - USB or other method



android

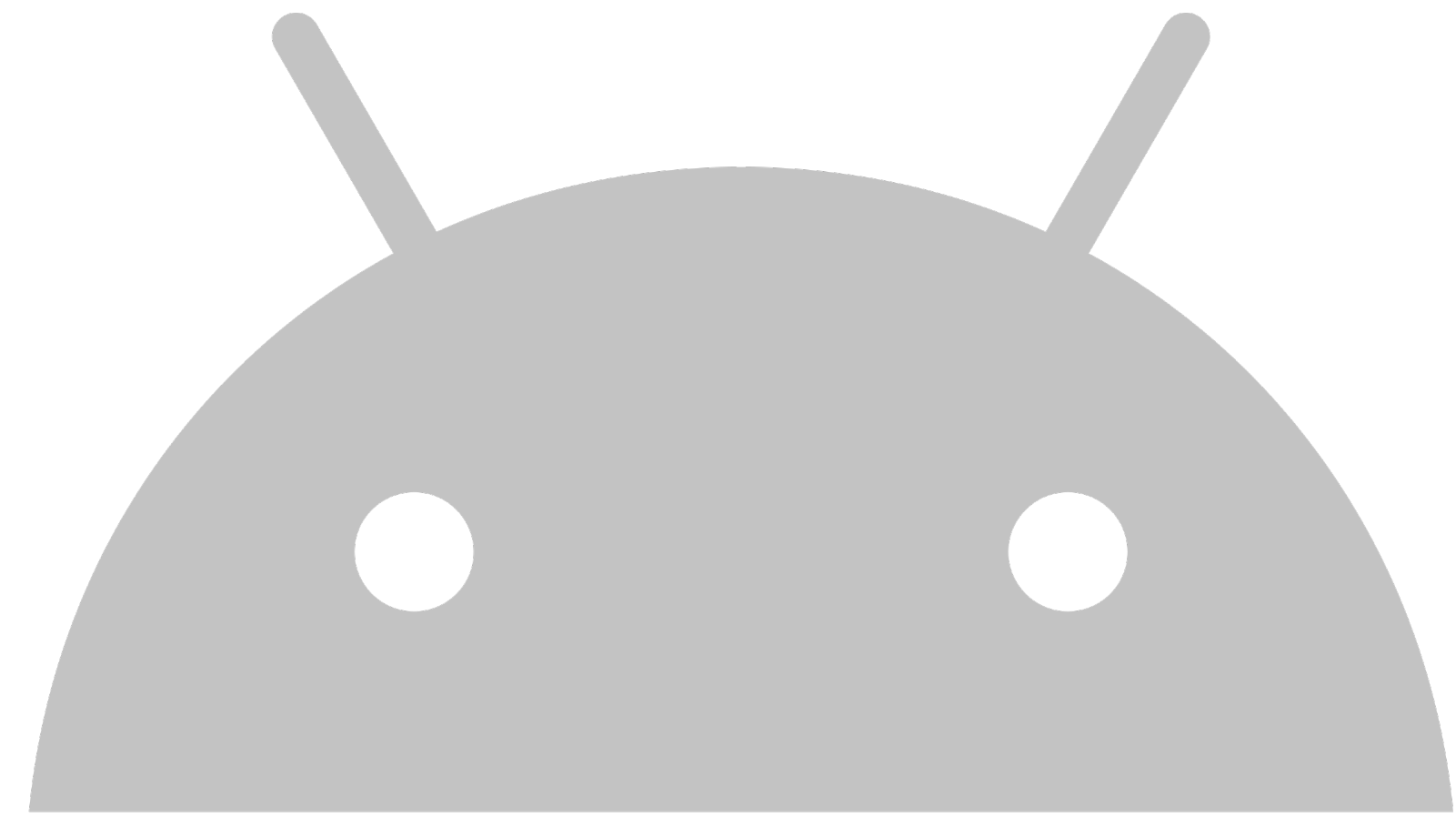
Android Versioning

Version number: 1.0 to 13.0

One major number per year

Codename: Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, ..., Oreo, Pie
(names deprecated from 10, but kind of still there sometimes)

API number (aka SDK): from 1 to 33



android

API versions
per app

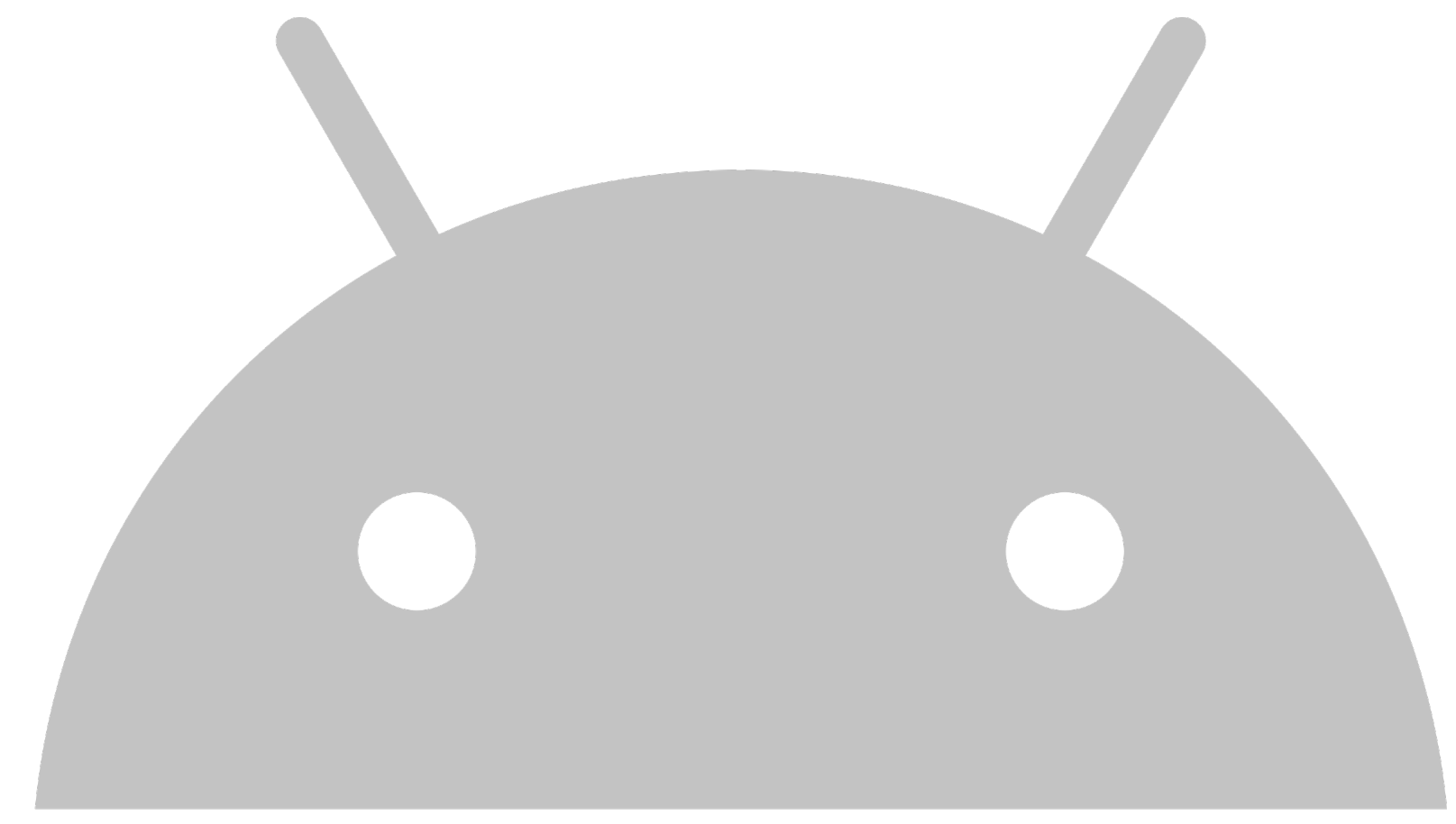
Minimum SDK version
(defines installability)

Target SDK version
(it changes behavior)

Compile SDK version
(not included in the app)

`minSdk <= targetSdk <= compileSdk`

Usually, target and compile matches



android

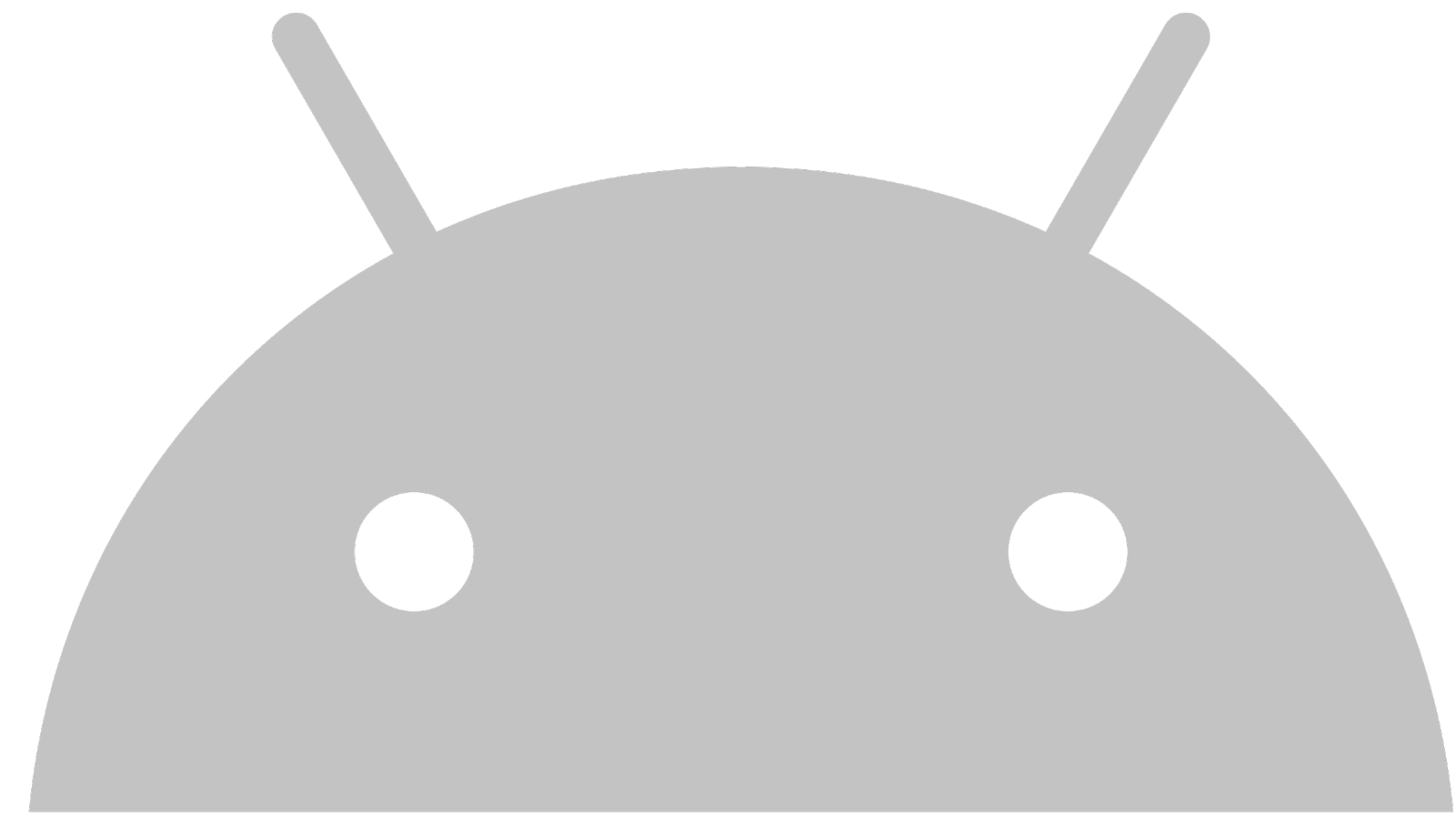
APIs
available

Every version includes new APIs

But if we target users on that new version, it will take years to get to the majority of users

For years we had something known as Support Libraries

In 2019, Support Libraries were replaced by **JetPack** (androidx packages)



android

Code
architecture

The official SDK is based on a Java VM and it's purely written in Java

Android doesn't use Oracle's JVM, but its own engines (known as Dalvik ART)

It doesn't compile native code

It compiles to Java Byte code

It does have Garbage collector

Kotlin also compiles to Java bytecode, so Java and Kotlin and 100% interoperable



Kotlin

- By JetBrains
- Open Source
- Statically typed language
- Type Inference
- Multi-platform
- Easy to Learn
- Concise - less ceremony
- Modern Ideas
- Null-safety

Language Types

Developers write code in

Then ship

Interpreted Languages



Source Code

Intermediate Languages



Bytecode

Compiled Languages



Machine Code



Language Types

Developers write code in

Then ship

Interpreted Languages



Source Code

Intermediate Languages



Bytecode



Compiled Languages



Machine Code





Kotlin

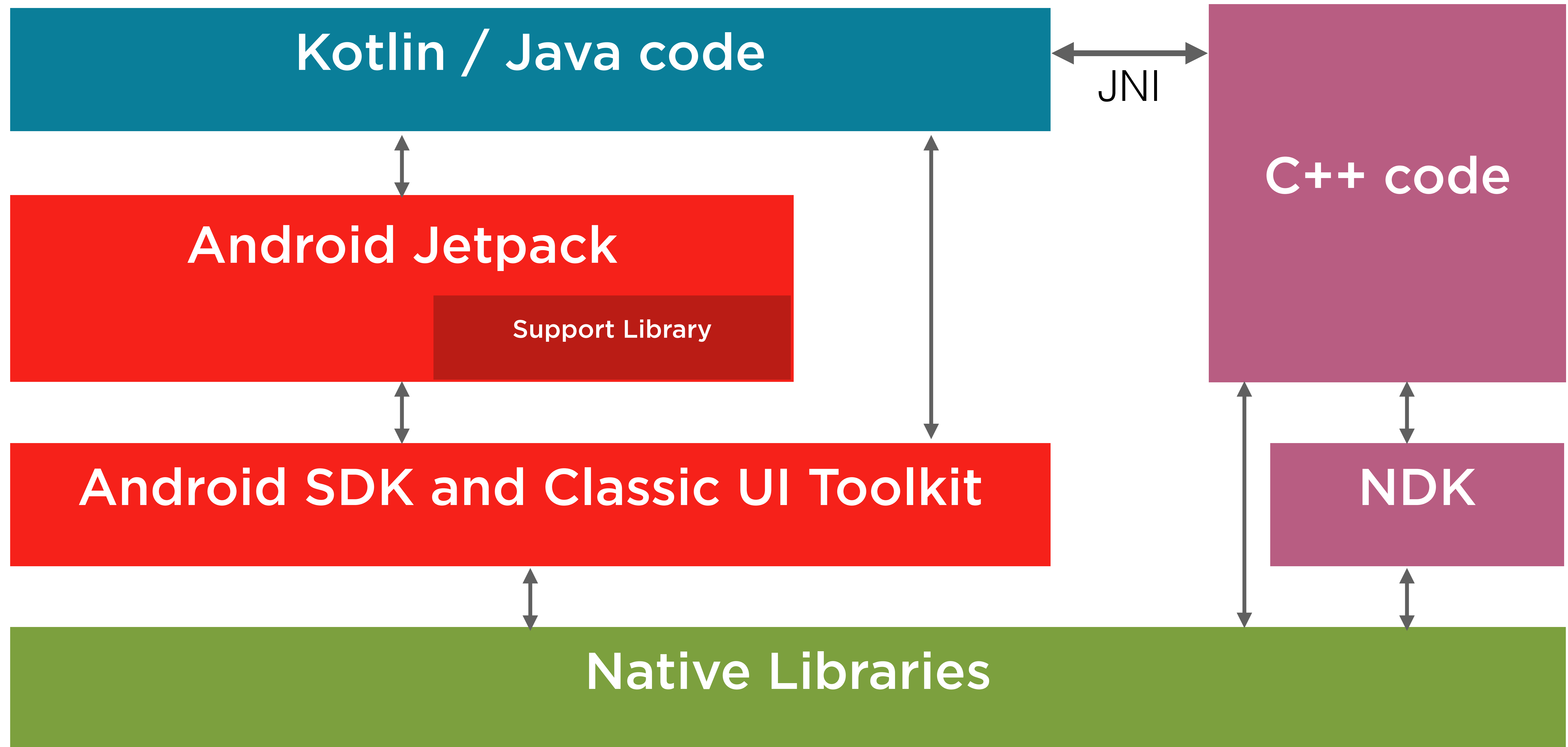
- JVM 6
- JVM 8
- Android - Java DEX
- JavaScript
- Native: iOS, macOS, Android, Windows, Linux
WebAssembly
- Kotlin Multiplatform Mobile?



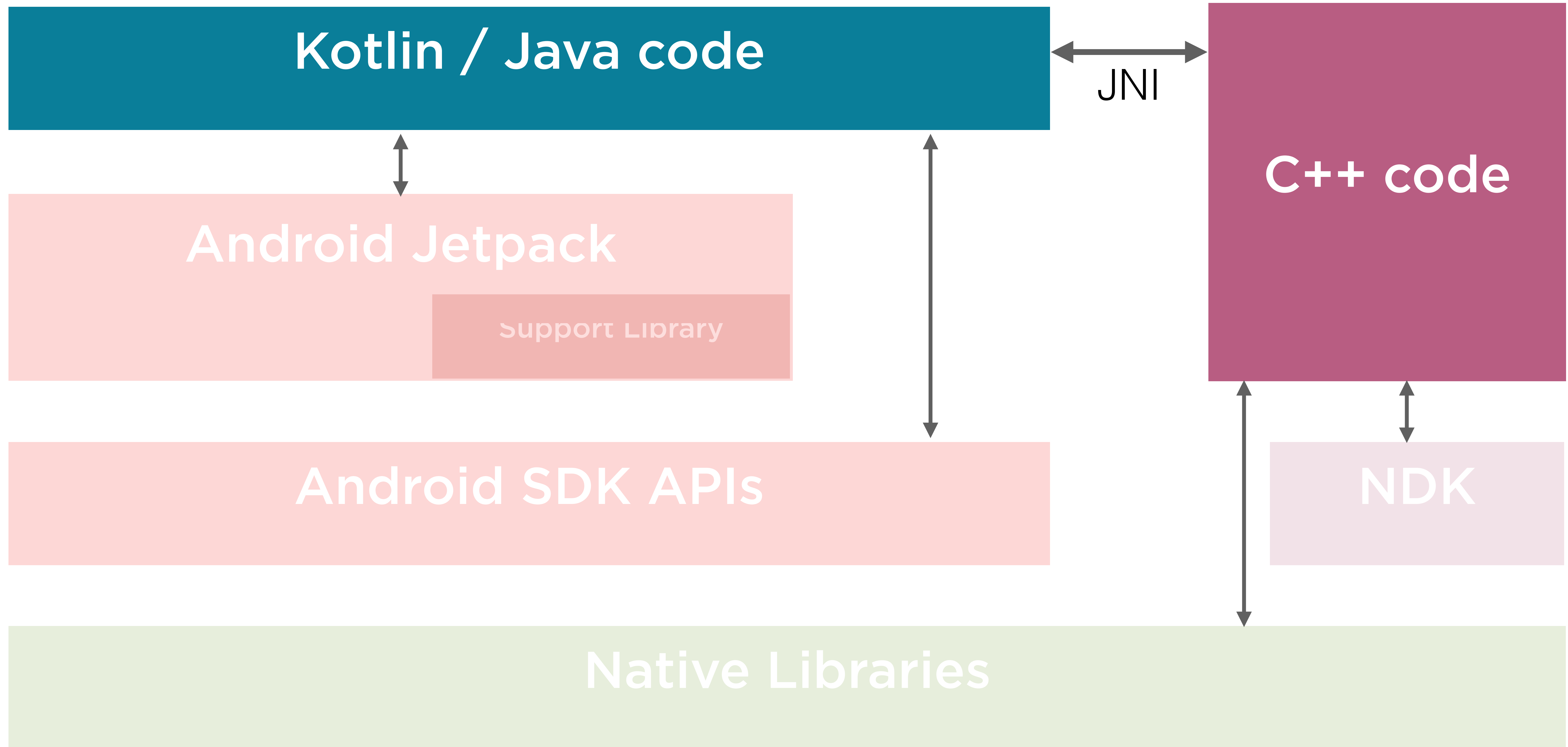
Kotlin

- JVM 6
- JVM 8
- **Android - Java DEX**
- JavaScript
- Native: iOS, macOS, Android, Windows, Linux
WebAssembly
- Kotlin Multiplatform Mobile?

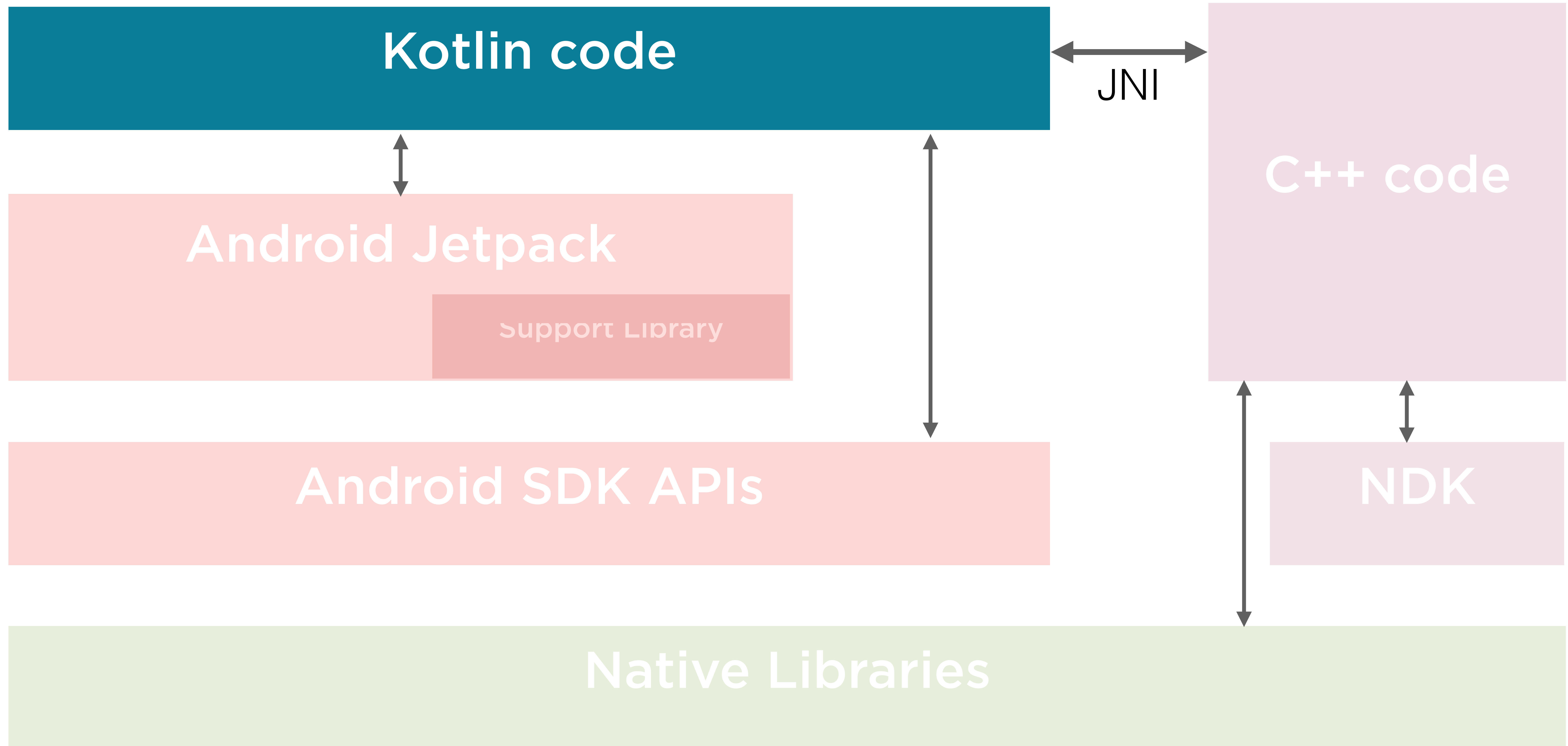
Android App Development



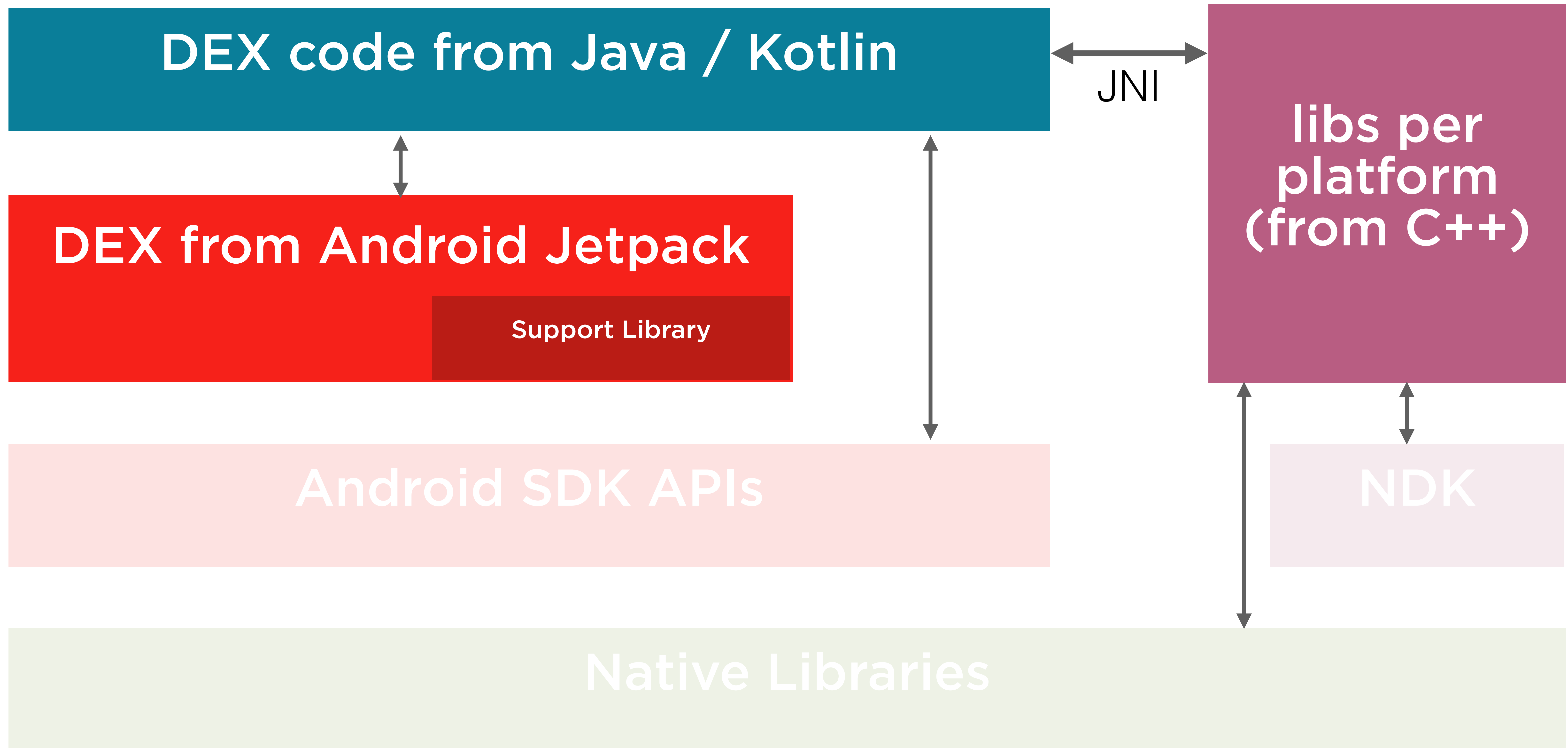
Our Code



Our Code



What goes in the APK file



Time to write some Kotlin code

Build better apps faster with Jetpack Compose

Jetpack Compose is Android's modern toolkit for building native UI. It simplifies and accelerates UI development on Android. Quickly bring your app to life with less code, powerful tools, and intuitive Kotlin APIs.

```
@Composable
fun JetpackCompose() {
    Card {
        var expanded by remember { mutableStateOf(false) }
        Column(Modifier.clickable { expanded = !expanded }) {
            Image(painterResource(R.drawable.jetpack_compose))
            AnimatedVisibility(expanded) {
                Text(
                    text = "Jetpack Compose",
                    style = MaterialTheme.typography.h2,
                )
            }
        }
    }
}
```



Jetpack
Compose



Jetpack Compose

- Declarative UI framework
- Launched in 2021
- Focus on modern UI patterns:
 - Single Source of Truth
 - Composable views
 - Multiplatform
 - Dependency Rendering
 - Data Binding & Reactive programming
 - UI expressed in Kotlin functions
 - Visual editor tooling in Android Studio

Time to create our project



Jetpack Compose

Main Concepts

- **Android Activity**
- **Theme**
- **Composable** (UI component)
- **Modifier**
- **Composition**
- **Mutable State & Remember**
- **Composition Local**
- **Side Effects**
- **Dimension Units:** px, dp, sp
- **Graphical Assets:** PNG, SVG



Jetpack Compose

**Some Built-in
composable**

- **Text**
- **Button**
- **Surface**
- **Card**
- **Column**
- **Row**
- **Box**
- **Spacer**
- **LazyColumn**
- **LazyRow**
- **TextField**

A blurred high-speed train in motion on a track, with a stone bridge in the background. The train is moving from left to right, and the background shows a stone bridge with multiple arches. The train is silver and has a red stripe. The tracks are on a gravel bed. There is a white signpost on the right side of the track.

hi@firt.dev

@firt

Fot