Expo

By Marius Uktveris and Andojas Lapinskas

Phase 1

→ **Intro & Research Outcomes**

About Expo and advantages of using it

Key features

Key features of expo

Demo

Short demo

Why it exists?

Expo exists to simplify mobile development with React Native.

Removes native setup complexity

Gives prebuilt access to native APIs

Enables fast, iterative development

Fast refresh and unified toolchain help expo stand out in ease of development

Handles builds, updates, and configuration

Tools like EAS and Expo Go make building, testing and shipipng apps easy from the get-go.

What problem does it solve?

Expo solves the “native friction” of mobile development.

No need for native build tools

Expo handles everything you need to develop on any platform

No manual linking native modules

Expo provides native APIs

Faster onboarding + faster iteration

Tools like EAS and Expo Go make building, testing and shipipng apps easy from the get-go.

Why is it relevant?

Expo Router

Expo handles everything you need to develop on any platform

Cloud builds

EAS Build + Submit`

Custom native support

Dev client provided by EAS help you test your app quickly

Expo vs Alternatives

Expo

vs

React Native

capacitor

Flutter

Kotlin

Swift

Jetpack Compose

Expo vs Alternatives

Flutter

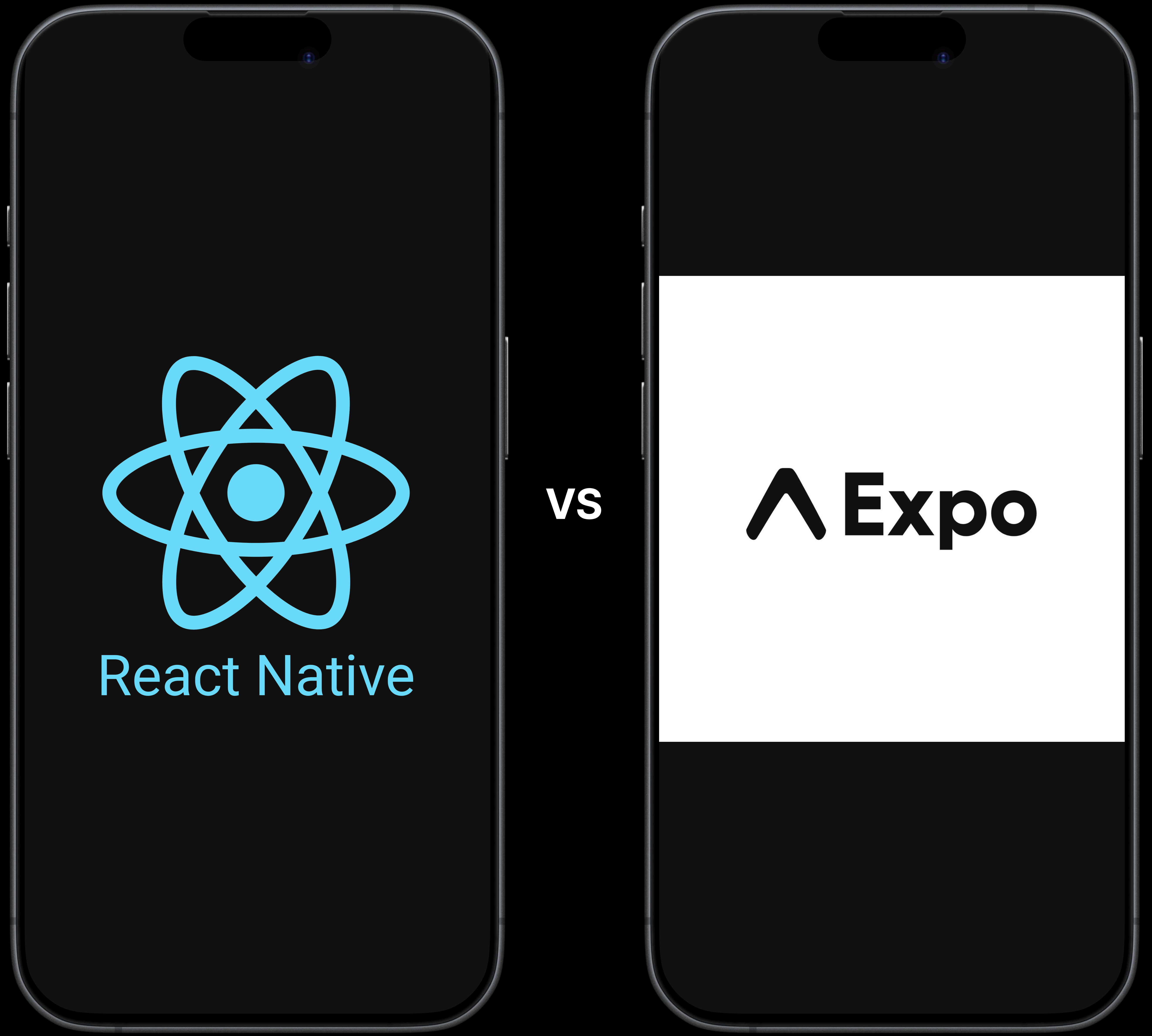
- **Flutter is better** when you need highly polished native performance or custom native rendering that Flutter’s engine provides out of the box.
- **Expo is better** when you want to use JavaScript/React and get rapid, web-ready development with minimal native setup.

Expo vs Alternatives

React Native CLI

- **React Native CLI is better** when you need full control of native code and custom platform configuration without relying on Expo's abstraction.

- **Expo is better** when you want a smoother setup, built-in native modules, and easier building/deployment without touching Xcode or Android Studio.



Expo vs Alternatives

Capacitor/Ionic

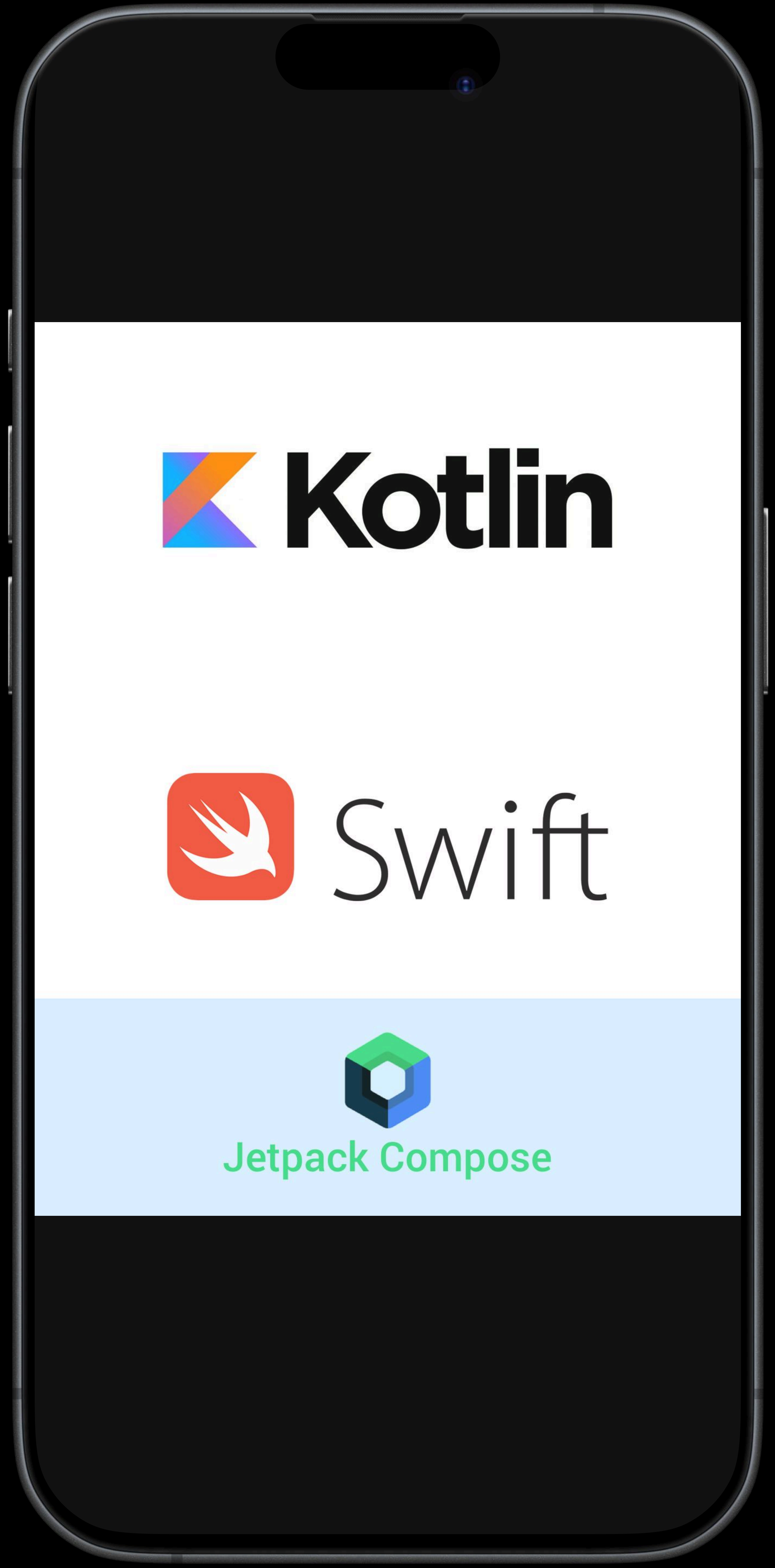
- **Capacitor/Ionic is better** when you need a web-first development model and want to write mostly HTML/CSS/JS rather than React Native components.
- **Expo is better** when you want truly native UI components and performance rather than web views wrapped in a mobile shell.

Expo vs Alternatives

Native (Swift / Kotlin / Jetpack Compose)

- **Native is better** when you need maximum performance, deep platform integration, or access to cutting-edge native APIs the moment they're released.

- **Expo is better** when you want to build cross-platform apps faster with one React codebase and far less platform-specific setup.



vs

When to use Expo

✓ **Fast setup**
Start coding without native tools.

✓ **Cross-platform**
iOS, Android, Web with shared code.

✓ **Built-in APIs**
Camera, Sensors, Haptics, File System ready to use.

✗ **Custom native code**
Requires modules not in Expo Go.

✗ **Unsupported SDKs**
Proprietary or niche SDKs.

✗ **High-performance apps**
Games, AR/VR, or low-level optimization.

Phase 1

Intro

About Expo and advantages of using it

→ ### Key features

Key features of expo

Demo

Short demo

Fast Refresh / Real-Time Updates

Saving a file will update both web and mobile in real-time, saving the struggle of manually refreshing each platform after a change.

Expo Router

File-based routing makes navigating between screens simple and predictable, similar to Next.js for the web.

Web Support

Your Expo app can run in a browser with the same code you use for mobile.

Expo Go

Scan a QR code to load your app instantly on a phone without any installation or setup.

Asset & Font Management

Import images, fonts, and icons easily, with everything working automatically across platforms.

Phase 1

Intro

About Expo and advantages of using it

Key features

Key features of expo

→ ### Demo

Short demo

Demo

Phase 2

→ **CNG**

Concepts of native code generation

EAS

Features and benefits of EAS

Task

Practical task with Expo APIs

Continuous Native Generation

One source of truth

Both native projects are generated from one source - JS / TS codebase.

Prebuild command

One command can (re)generate both /ios and /android directories on demand.

Config plugins

Custom native configuration can be managed by small JS / TS functions.

Flexibility

CNG can be easily adopted by native projects.

Phase 2

CNG

Concepts of native code generation

→

EAS

Features and benefits of EAS

Task

Practical task with Expo APIs

Expo Application Services

Workflows & CI/CD

Provides infrastructure for CI/CD integration with builds, updates, etc.

Development tools

Builds, submits, insights, etc.

OTA updates

Over-The-Air updates, contributing to faster development.

Phase 2

CNG

Concepts of native code generation

EAS

Features and benefits of EAS

→ **Task**

Practical task with Expo APIs

Task

The task is to implement a function using one expo API.

- expo-camera
- expo-location
- expo-sensors
- expo-video

Task reflection

Trivia / Kahoot

Questionnaire about the topics of Expo

Summary & questions