

# Native vs cross-platform app development

## Native apps

Native apps are built specifically for iOS or Android and offer top-notch performance and an unmatched user experience

### NOTABLE ADVANTAGES

- Top-level performance and speed
- Highest level of security and data protection
- Complete access to device resources
- Lower storage space requirements
- Full control over UX/UI design
- No third-party library restrictions

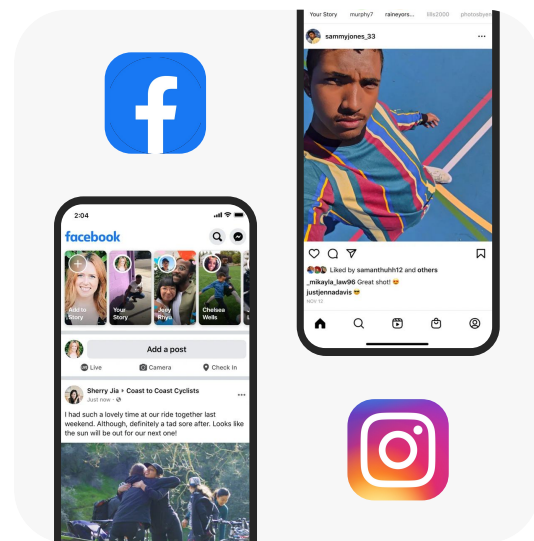
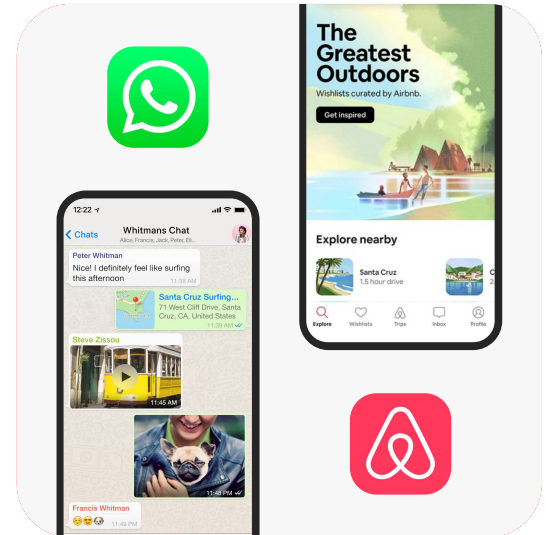
### DISADVANTAGES

- Higher app development and support cost
- Longer development time
- Require a high level of expertise to build

### BEST FIT FOR

- Telecommunication apps
- Mobile banking apps

- IoT apps
- Gaming apps



## Cross-platform apps

Cross-platform apps are built to work on multiple platforms/OS so they can run on smartphones, tablets, PCs, smartwatches, and even connected TVs.

### NOTABLE ADVANTAGES

- More cost-effective
- Shorter time to market
- 60% - 90% reusable codebase
- Easy maintenance and update synchronization
- Wider reach

### DISADVANTAGES

- Lack of platform-specific optimizations may cause performance issues
- Limited third-party library support
- Lacks access to some native OS features
- Dependent on framework providers
- Dependent on developer communities

### BEST FIT FOR

- Social media apps
- Simple MedTech apps

- Hospitality apps

## How to make the right decision?

### GO NATIVE IF...

- Performance, speed, and UX are top priorities.
- You're building a long-term product where sustainability matters.
- Safety and reliability are critical.
- You need device-specific functions (camera, GPS, microphone, etc.).
- Your app will run on one operating system (for now).
- You need a platform-specific UI design.
- Your app needs to perform tasks in the background.

### GO CROSS-PLATFORM IF...

- Time and/or budget are tight.
- You need a fast proof-of-concept.
- Both iOS and Android need to share complex logic.
- A consistent UI/UX across platforms is a key requirement.
- You want to use existing JavaScript logic or libraries (→ React Native).

	NATIVE	CROSS-PLATFORM	NOTE
Performance	+++	++	No doubt, native apps have better performance and are more responsive than cross-platform apps. This is mainly because native code interacts directly with the device's resources.
Development time	+	+++	Native apps take significantly more time to develop as you need separate teams for each platform. Cross-platform apps shine here since they share the same code across several platforms.
Development cost	+	+++	Longer development time and more personnel means higher development costs. Building for different platforms with one team will save you a lot of resources.
Hardware resources	+++	+	Native development gives you complete access to device resources. Cross-platform apps try to make up for it with plugins, but compatibility issues are still common and they often take up more space.
UX/UI design	+++	++	Cross-platform development also lags behind when it comes to UX/UI design. With native development, you get unrestricted access to UX/UI components and can deliver stunning visuals and an outstanding UX.
Product lifetime	+++	++	Native development is safer for long-term projects because the tools and frameworks are more stable, with fewer sudden changes.
Background processes	+++	+	Allows the app to perform tasks in the background, must be native.
Push notifications	+++	++	Handling push notifications can be tricky in cross-platform apps, native apps manage it much better.
Debugging	+++	++	Cross-platform apps don't have the same debugging capabilities, e.g. Crashlytics doesn't work with React Native and Flutter.
OS updates	+++	++	It takes a few weeks until cross-platform catches up with the latest OS updates.
Dependency	+++	++	Native apps are less dependent on open-source libraries and platforms.

Your strategic partner in creating exceptional software products

Let's talk



Ivor Cindric  
Strategic Partner