Choosing the Right Framework for Your Cross-Platform Development

Dominik Titl

What is cross-plat and why you might want that?

- attract more users
- costs
- easier for developers
- making your life easier... well, sometimes

Expectations vs. reality

write once, debug everywhere

Jiri Charousek • 2nd
CIO and Member of the BoD at Partners Banka ● Advisory bank ● Clien...
17h •

+ Follow

Developing one mobile app for two different systems with one team? Many companies would be patting themselves on the forehead.

We did it. And it makes perfect sense to us.

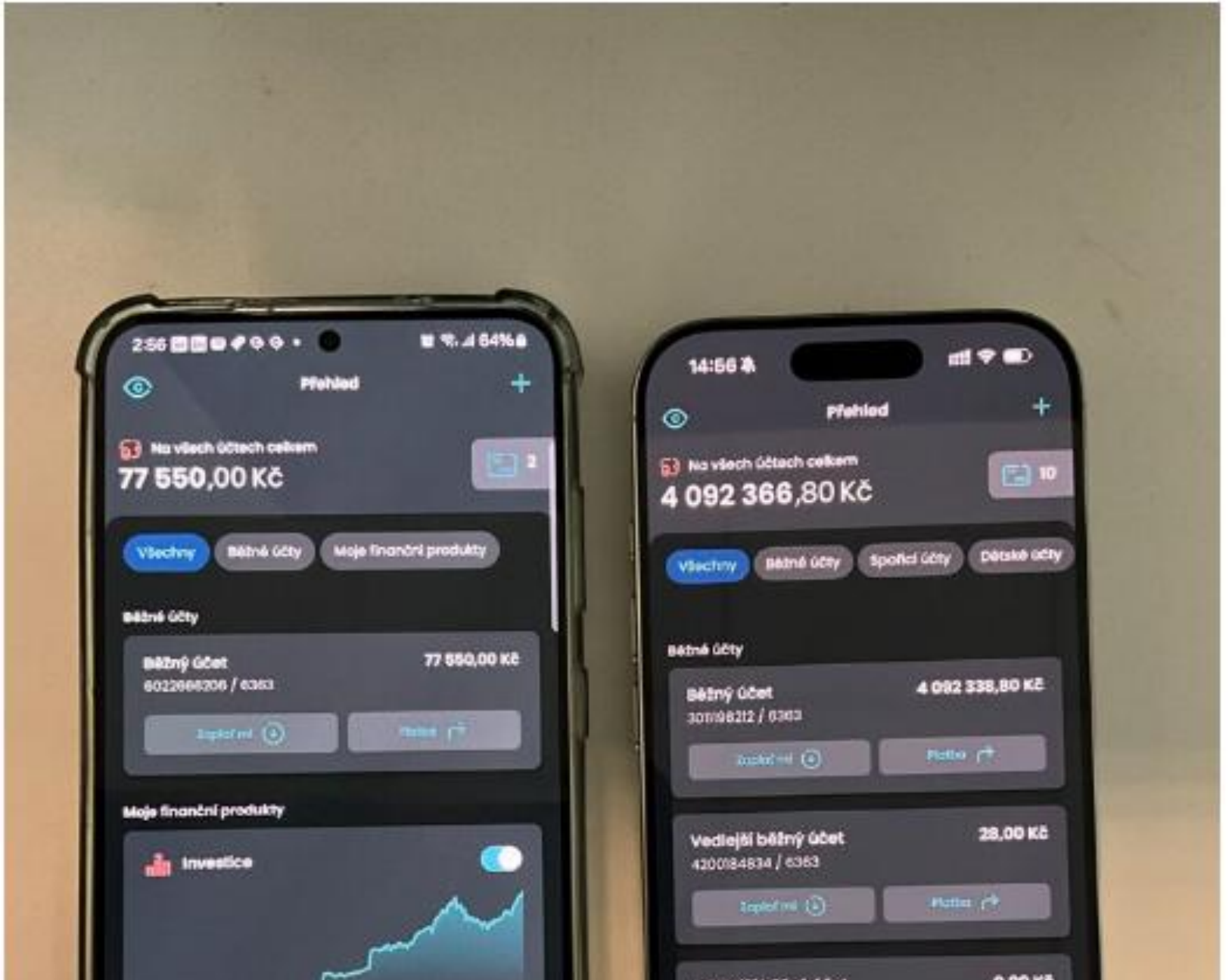
Most companies have separate teams – one for iOS, one for Android. Two different approaches, two different paths, many things twice...

We went the other way: one team, one mobile app, same approach...

👉 Same design, same logic. Whether you pick up an iPhone or a Samsung, the mobile app behaves the same.
👉 Fewer people, more efficiency. We keep know-how together and don't waste our efforts.
👉 Faster development. When we deploy changes, we only do them once – not twice.

Sure, it has its challenges. Sometimes the mobile app behaves differently on different platforms, sometimes you need to debug specific bugs for one system. But the benefits outweigh...

What do you prefer? One unified approach, or each system separately?
#PartnersBanka #JsmePartners



Kotlin and Swift have the same reason why they are declining. They are both mainly used for one particular mobile platform, Android and iOS, respectively, whereas there are other sufficiently good languages and frameworks to develop cross platform nowadays.

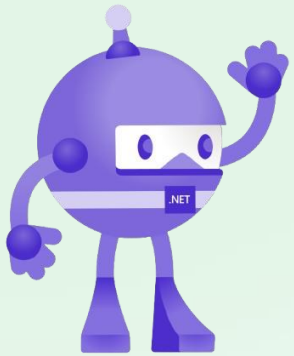
- Paul Jansen, Founder & CEO TIOBE Software


IT DEPENDS

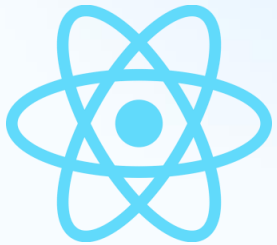
Key Decision Factors

- performance
- ecosystem
- development experience
- ui control
- stability & maturity
- company backing

Our contenders

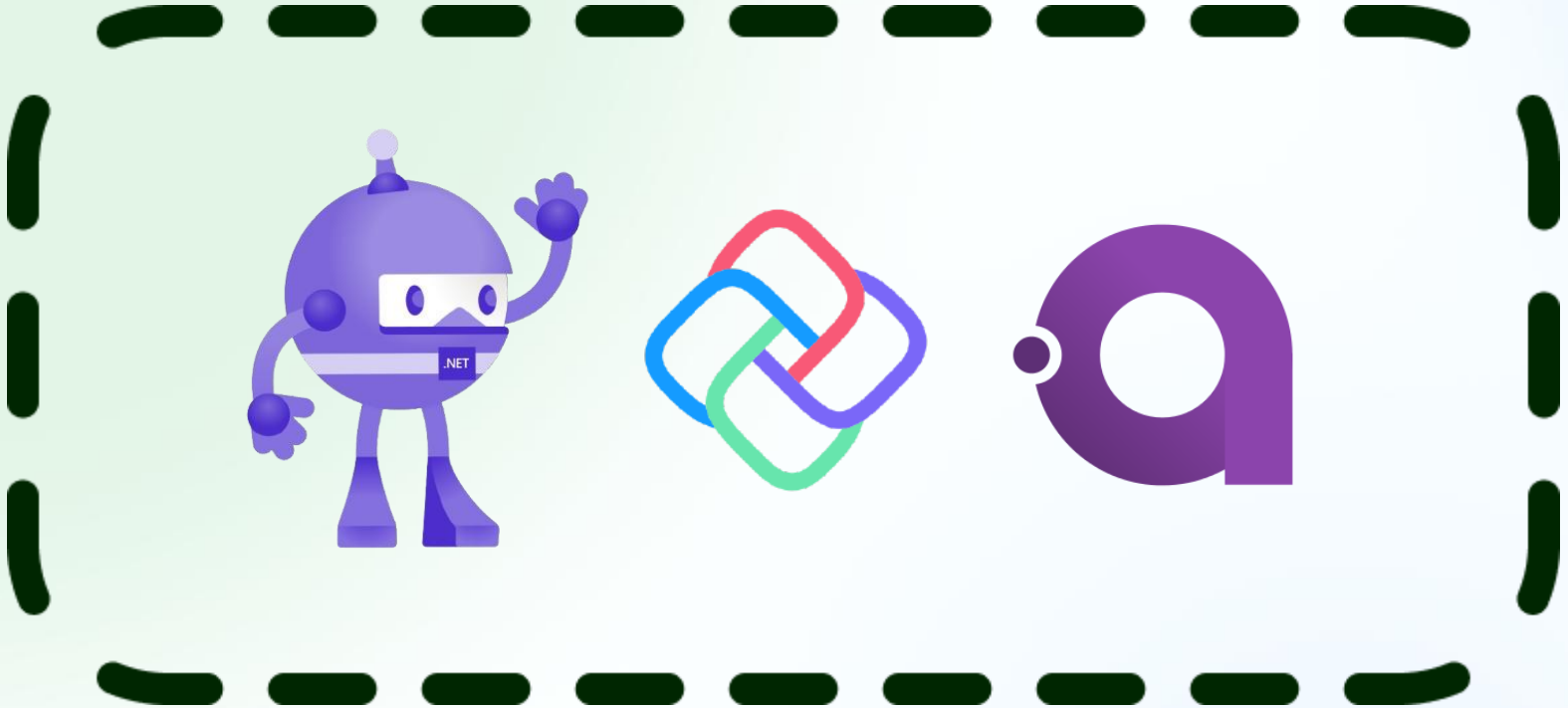




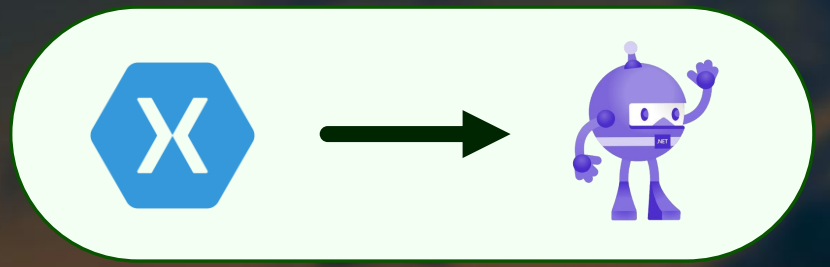
*Ionic, NativeScript, Kotlin Multiplatform, Electron, etc.

Our .NET contenders



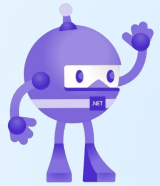
support ended on May 1, 2024

MAUI



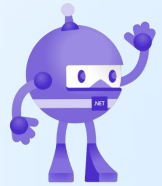
- Successor to Xamarin.Forms
- Microsoft’s cross-platform framework
- Android, iOS, macOS, and Windows with a single codebase
- Native rendering

+ (pros)



- Full C#/.NET stack
- Native controls = native UI feel
- Good mobile support
- Tight integration with Visual Studio
- Easy for existing Xamarin.Forms developers

- (cons)



- Mobile performance not as optimized as others
- UI customization sometimes verbose
- Android/iOS platform quirks
- No web/Linux desktop support
- Smaller ecosystem than Flutter/React Native

Apps with MAUI

Scribzee

Azure App

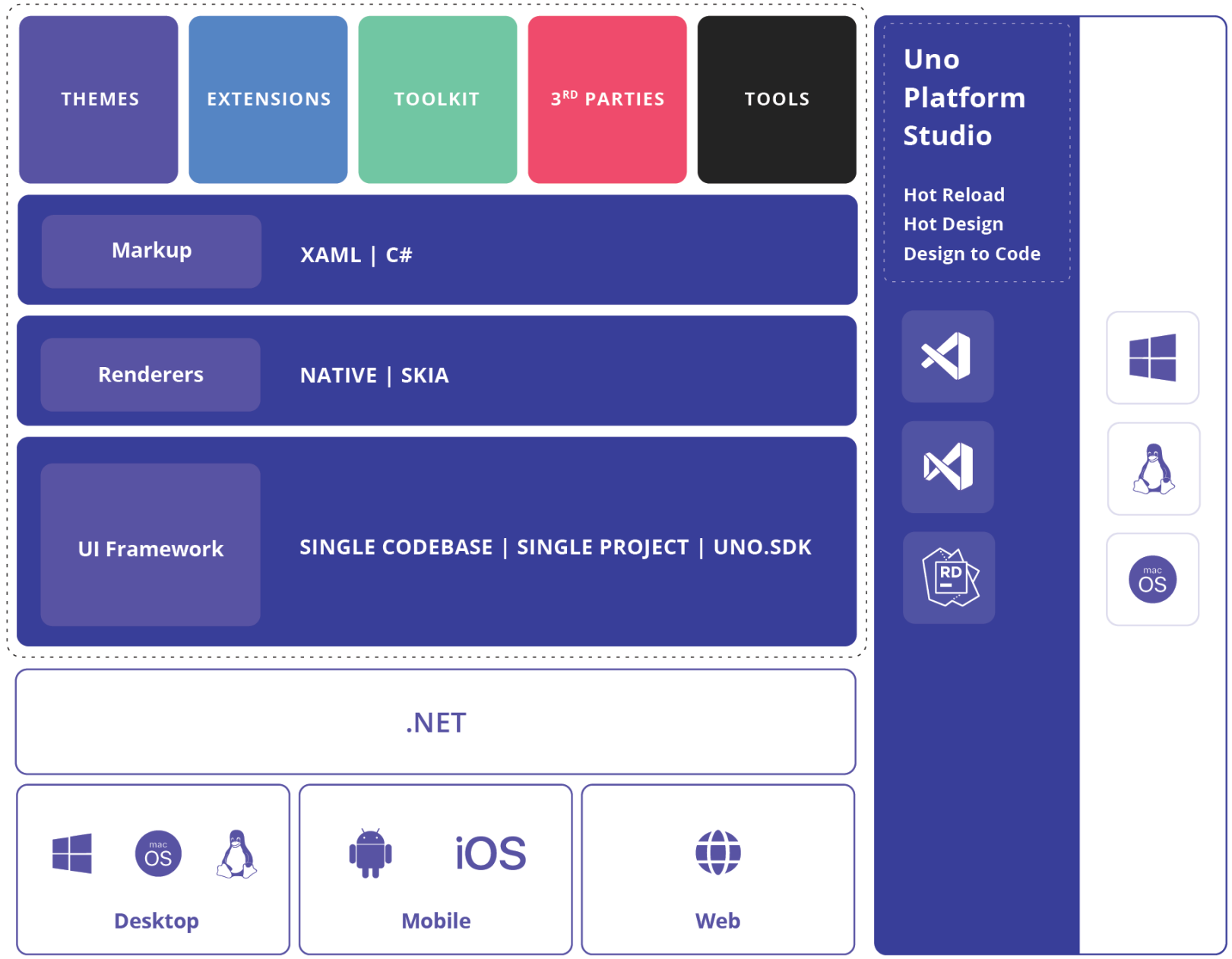
Berichtenbox

Red-Point (by Alpha Outdoors)

Ticketsphere

dotnet.microsoft.com/platform/customers/maui


uno®
PLATFORM


THEMES
EXTENSIONS
TOOLKIT
3RD PARTIES
TOOLS
Markup
XAML | C#
Renderers
NATIVE | SKIA
UI Framework
SINGLE CODEBASE | SINGLE PROJECT | UNO.SDK
.NET
Desktop
Mobile
iOS
Web
Uno Platform Studio
Hot Reload
Hot Design
Design to Code
RD
mac OS

+ (pros)

- Web (Wasm), Windows, Android, iOS, macOS, Linux
- Full-fledged WinUI styling
- Optional visual designer inside the running app
- Reuse UWP/WinUI XAML & C# skills
- Fluent or Material design out of the box

- (cons)

- Smaller ecosystem/community, smaller enterprise adoption
- Steeper learning for non-WinUI developers
- Some platform APIs limited on non-Windows
- WinUI API surface dependency

Apps with Uno

Hug App

Kahua

Eppie-App

nuget.info

Toyota
(internal app)

TradeZero

platform.uno/case-studies

Avalonia

- Community-driven cross-platform UI framework
  Windows, macOS, Linux
- Mobile & web in preview
- Single codebase with XAML (WPF flavor) and C#
- GPU-accelerated, canvas-based rendering

+ (pros)

- Strong desktop focus (Windows, Linux, macOS)
- Many success stories for desktop apps
- Good MVVM/XAML support
- Fast canvas-based UI rendering
- Cross-platform theming/styling

- (cons)

- Experimental mobile and web support
- Increasingly more components/features are under a subscription (Accelerate)
- Smaller ecosystem than MAUI/Uno
- Some platform-specific features missing
- Newer devs may need time learning Avalonia XAML differences

Apps with Avalonia

JetBrains Rider
(plugin host UI components)

Prismatik (Lightpack fork)

Core2D

Beutl

Suki Assistant

PassWinMenu

avaloniaui.net/showcase

Flutter

- Google’s cross-platform UI framework
- Community-driven with strong ecosystem
- Android, iOS, Web, Windows, macOS, Linux
- Single codebase with Dart
- GPU-accelerated, canvas-based custom rendering

+ (pros)

- Fast UI performance (Skia engine)
- Huge ecosystem & plugins
- Strong mobile support (iOS/Android)
- Great tooling & hot reload
- Growing desktop & web support

- (cons)

- Web/Desktop still catching up
  (I wouldn’t recommend for desktop dev at all)
- Dart language (extra learning curve)
- Native platform APIs need bridging
- Big reliance on community pubs 🍻 (plugins)

Apps with Flutter

ByteDance

eBay Motors

Google Ads

Nubank

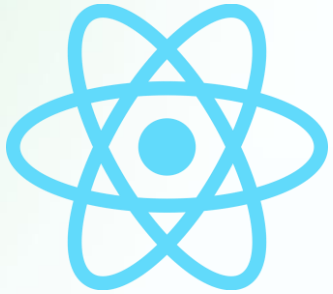
Alibaba/Xianyu

Google Pay

Reflectly

BMW / My BMW App

flutter.dev/showcase



React Native

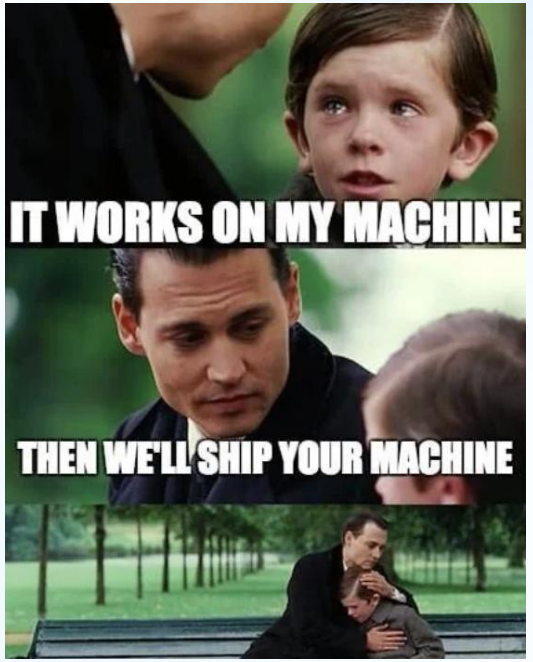
- Meta’s (Facebook) cross-platform UI framework
- Large community and rich plugin ecosystem
- Android, iOS — Windows & macOS via community support
- Single codebase with JavaScript or TypeScript
- Bridges to native UI components for performance

+ (pros)

- JavaScript/TypeScript = popular & accessible
- Fast development & live reload
- Large open-source ecosystem
- Easier 3rd-party integrations
- Native-feel UI with native components

- (cons)

- Performance bottlenecks for complex Uis
- Reliance on community plugins
- More manual native bridging for deep APIs
- Web needs extra frameworks (e.g., Expo, React Native Web)
- Debugging native issues can be tricky



Apps with React Native

Bloomberg

Skype

Uber Eats (partially)

Facebook (partially)

Walmart

Instagram (partially)

Tesla

Discord

Pinterest

reactnative.dev/showcase

Who is the WINNER?

Who is the WINNER?

YOU, not having to use Java Swing anymore...

My recommendation

Make an MVP using at least two of your selected choices.

Developing a desktop app?

Avalonia

Developing a mobile-only app?

Flutter, MAUI/Uno

Performance-heavy Apps

Flutter

Creating an app targeting all platforms including web?

Uno Platform

Developing a desktop app?

Avalonia

Teams Skilled in UWP/WinUI?

Uno Platform

Additional resources

Thank YOU!

bento.me/morning4coffe