THE LINUX FOUNDATION | Research

Sponsored by OPEN MOBILE HUB

Expanding the Vision for the Mobile Industry

How to Gain More Choice, Innovation, and Productivity with Open Source

August 2025

Gordon Graham

Foreword by Mike Woster, The Linux Foundation

Expanding the Vision for the Mobile Industry

Android with GMS & iOS dominate the mobile market. But platforms like Fire OS, Harmony OS, and Horizon OS, have 1.75+ billion installs with great potential.

Every mobile platform supports default services. Any competing service struggles to gain traction, which limits innovation & choice.

The number of mobile platforms keeps growing. But building a native version for each one costs more, delays releases, adds risk, & slows time to market.

The key is to **stop building the same thing twice**—by using a cross-platform framework like **Flutter or React Native.**

Cost-effective dev teams are mostly cross-platform, with a few native specialists. That way, 80% of the codebase can be shared & 20% can stay native where needed.

There is an estimated **30%+ productivity gain** for a mobile development team that switches to a cross-platform framework.

Cross-platform frameworks aren't new—and they've been maturing for years.

- 2015: React Native (Meta)
- 2017: Flutter (Google) and KMP (JetBrains)

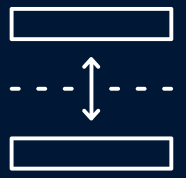
Cross-platform development delivers clear wins: Faster onboarding, faster updates, less time spent testing & debugging, and lower costs across the board.

Many major companies use cross-platform frameworks, such as Alibaba, Amazon, BMW, Duolingo, eBay, Google, Meta, Microsoft, Shopify, Tencent & more.

All major cross-platform frameworks are open source. Each is backed by a global community that contributes libraries, plug-ins, and fixes to fill in the gaps.

Cross-platform frameworks still have gaps. They mainly support Android with GMS and iOS, but not emerging platforms & rarely support alternatives to default services.

The Open Mobile Hub aims to fill the gaps in today's frameworks with open source to support more choice in key mobile services.



Contents

Foreword

The mobile industry is at an inflection point. For years, developers have worked within the constraints of just a few dominant platforms—conforming to their APIs, building to their specifications, and too often duplicating work across ecosystems.

While Android (with Google Mobile Services [GMS]) and iOS have defined much of the mobile experience, they are not the only platforms in play. A new generation of mobile environments is already here, with more than 2 billion devices offering real opportunities for innovation.
What has held us back is not the lack of devices but the lack of open, flexible tools that can bridge these environments without locking developers into one ecosystem or default service stack. That is where open source can change the game.

This report—resulting from 24 interviews with mobile developers and stakeholders across 10 countries—is a timely contribution to the global mobile conversation. It reveals just how much time developers spend rewriting the same features across different platforms, and how cross-platform frameworks, such as Flutter, Kotlin Multiplatform (KMP), and React Native, have already begun to address these inefficiencies.

Yet, it also shows where current frameworks fall short—especially when it comes to supporting non-GMS Androids or offering alternatives to built-in services, such as maps, payments, or authentication.

Open Mobile Hub (OMH), an open source initiative, is stepping up to meet that challenge. By building a set of intelligent APIs that can detect the platform a device is running and switch services on the fly, OMH opens the door to real interoperability—and with it, true competition and user choice.

We believe the future of mobile is open, and with OMH, that future is within reach today more than ever.

Mike Woster

Chief Revenue Officer
The Linux Foundation

Executive summary

At first glance, regular Android and iOS appear to be the only platforms on the mobile horizon. However, a closer look reveals that **more than 2 billion devices have shipped for other platforms**. Among these are devices based on the Android Open Source Project (AOSP), such as Fire OS and Meta Horizon OS, and emerging platforms such as HarmonyOS.

Some of these platforms dominate in certain regions due to cost, features, or geopolitics. At the same time, regulators are prying open the once-locked app stores. Alternate services are challenging the platform defaults.

This broader selection of platforms, channels, services, and workflows calls for a fresh look at the mobile landscape.

The old model is no longer sustainable

For too long, mobile development followed a costly pattern: Build native applications for each target platform, accept the default services, and distribute through the platform's app store. This added costs, delayed time-to-market, and limited choice for developers and consumers alike.

Cross-platform development offers a sharper focus. Frameworks such as Flutter, Kotlin Multiplatform (KMP), and React Native enable developers to reuse 80% or more of their code across platforms. Teams that adopt these frameworks report productivity gains of 30% to 50%. These frameworks speed up onboarding, building features, releasing updates, and fixing bugs.

That is why enterprises from Alibaba to Google to Walmart rely on cross-platform frameworks for mission-critical apps. So do many startups and smaller firms. The growing ecosystem delivers robust libraries, plug-ins, and tools to meet most common needs.

A clear vision of what's next

However, many gaps remain. Most frameworks target Android with GMS and iOS, with limited support for non-GMS Android, emerging platforms, and alternate services for areas such as authentication, cloud storage, maps, and payments.

The OMH, a project of the Linux Foundation, aims to close these gaps by helping frameworks support more platforms and services through unified APIs and open standards.

For mobile decision-makers, now is the time to expand your vision. The landscape is changing. A better way to build and deploy apps is coming into focus—with lower costs, more innovation, and more choice for everyone.

Mobile developers face many challenges

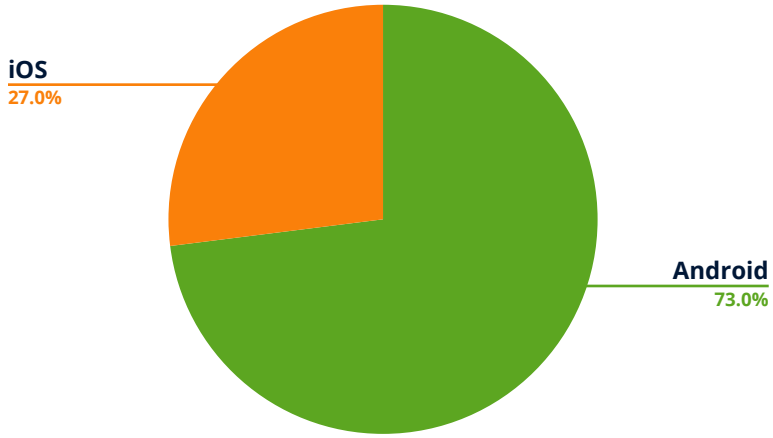
At first glance, Android and iOS appear to be the only players on the mobile landscape. Yet, new platforms and devices appear all the time. Native development for each platform drains precious resources. Well-meaning regulations have unintended consequences. Regional differences mean that an app that succeeds in Silicon Valley might struggle in Sokoto.

All of these challenges blur together to change our view. However, this report can help executives look at the mobile landscape differently and glimpse some exciting new opportunities.

Challenge 1: The simple view does not match the landscape

When most of us look at the mobile landscape, we only see Android and iOS. When we add up the installed base for the whole world, Android emerges as far more popular than iOS, as **FIGURE 1** shows. However, this view is outdated and incomplete.

FIGURE 1: ANDROID AND IOS GLOBAL MARKET SHARES, 2025



Source: Statcounter, May 2025 [1]

TWO VERSIONS OF ANDROID

Android comes in two distinct versions:

- **Standard Android:** Phones from Samsung, Xiaomi, Vivo, and other companies use this version.
- **AOSP:** This is the open source version without GMS, the proprietary layer that provides key features like authentication, cloud storage, maps, and purchases. While apps like Gmail, Google Maps, and Google Drive rely on GMS, so do most third-party Android apps.

Most stats for Android tend to lump both versions together. So, our pie chart really should have another slice for AOSP, which most analysts peg at roughly 7% of all Android shipments.[2]

MANY DIFFERENT VERSIONS OF AOSP

Any original equipment manufacturer (OEM) can fork AOSP to create their own version of Android. However, without access to GMS, they also need to replace the essential services that GMS provides, including app sign-in, location, push notifications, and security. Most Android apps depend on these services.

Replacing GMS means more than swapping out a few apps; it means recreating an entire service layer that the ecosystem relies on. Dozens of AOSP forks have been created, with varying levels of success.

"There used to be 50 different Android OEMs. I think it's down to 10 big ones today," says David Zeuthen, a senior staff software engineer at Google. That means some things may not look or feel the same on every phone. ***"Even on iPhones, you have different sizes, different features. As an app developer, you have to know all this, and that's a lot of work,"*** he notes.[3]

NEW DEVICES KEEP APPEARING

Beyond smartphones, developers face an ever-expanding array of form factors:

- AI-enabled devices, such as glasses and pendants
- Flip-phones, which are resurging as retro or minimalist devices
- Headsets from Meta (Horizon OS) and Apple (Vision Pro)
- Smartwatches and other wearables, such as smart rings

In addition, hundreds of millions of connected cars are rolling onto our highways, each one a computer on wheels that includes up to 200 Internet of Things (IoT) sensors that generate a constant stream of data.[4] The simple pie chart in **FIGURE 1** really does not show the whole picture.

Challenge 2: New platforms are coming into sight

In fact, more than 2 billion devices have been sold without standard Android or iOS. Our research—detailed in Appendix A—shows that at least 500 million AOSP devices have been shipped.

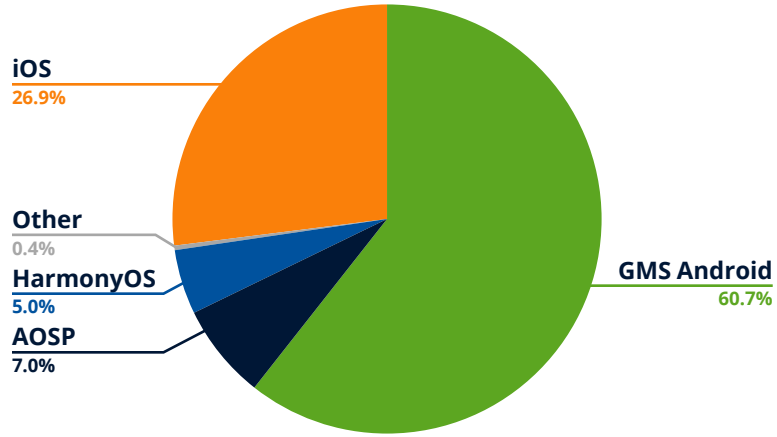
Also, a third global platform has appeared: HarmonyOS. Huawei created the platform in 2019, and it has evolved quickly. While versions 1 and 2 used Android and versions 3 and 4 used Linux, HarmonyOS NEXT (i.e., version 5) is a completely proprietary version. OpenHarmony is the open source version.

Harmony and OpenHarmony are on an estimated 1.1 billion devices. In China, HarmonyOS smartphones have outsold iOS since the start of 2024.[5] Huawei will soon roll out the platform to Africa and Europe.

As shown in **FIGURE 2**, a more realistic pie chart must include new slices for AOSP (non-GMS Android) and HarmonyOS, plus a sliver for other platforms. These three new additions now account for more than 12% of the world market for mobile devices, and that number is growing.

"I think that's a big opportunity for developers," says Ted Miracco, CEO of Approov, a mobile security service provider. ***"If you're selling to consumers in Europe or the Middle East or Latin America or India or China, you're going to need to think about this from a global perspective."***[7]

FIGURE 2: AN EXPANDED VIEW OF THE WORLD MOBILE MARKET, 2025



Sources: See note 6 in the References section.

Challenge 3: Default services limit choices

Every mobile platform provides a set of pre-installed apps like Apple Maps, Apple Pay, Apple iCloud, Google Maps, Google Pay, and Google Drive. These apps appear on the home screen on a brand-new device.

They're built-in and tightly integrated into the OS. So these apps gain an instant out-of-the-box advantage, shaping user expectations and habits from day one.

Underneath these apps are platform services such as authentication, cloud storage, maps, and payments. Developers rely on default services to build third-party apps from Airbnb to Uber. These services are deeply integrated into the OS SDKs, which makes them convenient to use but harder to replace.

This creates a self-reinforcing cycle, where users rely on the defaults because they're always there, and developers build around these services because they're the path of least resistance.

This makes it difficult for alternate apps or services to gain traction—even if they offer a better fit, privacy, security, or relevance to a certain region.

FROM MOBILE PAYMENTS TO CASUAL GAMES

This dilemma affects every app, from mobile payments to casual games.

"The payment systems that Google and Apple provide are integrated into the OS," notes Krysztof Magiera, director of engineering for Software Mansion, a development shop based in Poland. ***"When you use something else, consumers need to go through additional steps to pay,"*** he says. ***"So there's more incentive for them to select the default because it's easier."***[8]

Mobile game publisher Curro Alvarez from Spain sees the same tendency when developers choose services.

"Everybody uses Firebase for analytics and cloud storage because Google provides it for free," he says. ***"But once you have users in Firebase, if you want to move them to PlayFab from Microsoft, that's a big task."***[9]

This creates high switching costs for developers, making it tough for any alternate service to gain attention. The result? Both developers and consumers end up with fewer choices.

SWITCHING BETWEEN MOBILE SERVICES IS NOT EASY

Developers can have compelling reasons not to use a default service in their app. Google Maps may not have the best maps for some regions. Apple Pay may not work in some countries. In addition, if any service has an outage, a developer might want to switch to a backup service on the fly.

However, making that switch is never simple.

Every service relies on an API, and the provider can code it and write the documentation however it likes. An API can take a unique approach to authentication, data formats, and error handling.[10] An API can expose secrets and open security risks. When an API is updated, any app using the previous version could break.

To integrate any new service into an app, a developer has to figure all that out.

WORKING WITH MULTIPLE APIS IS DIFFICULT

An average app today uses 25 to 50 third-party APIs.[11] On top of that, Android ships with dozens of platform defaults for features such as cloud sync, push messaging, and payments. Each of those exposes their own APIs.

A developer may need to manage 100 or more different APIs: a mind-numbing chore.

Integrating numerous APIs ***"can be complicated, take valuable time, and make one vulnerable to problems such as errors, delays, downtime, data loss, security, and performance risks,"*** confirms the Hackmamba blog.[12]

It is no wonder that mobile developers see working with numerous APIs as one of their top frustrations.[13]

Challenge 4: Native development is inefficient

Once upon a time, every software firm had two teams with two codebases supporting two native apps for Android and iOS. These teams lived in separate worlds, used different languages, and seldom mixed.

However, supporting two platforms with separate teams means building every feature twice, fixing every bug twice, and releasing every update twice. That is a vast duplication of effort that adds obvious costs and delays. Also, if a company wants to develop for non-GMS Android, that means building the app a third time.

NATIVE DEVELOPMENT ADDS COSTS AND DELAYS

Using separate teams for each platform adds costs and delays across the app lifecycle:

- Onboarding slows down as new developers learn platform-specific tools.
- Features lag between platforms.
- Inconsistent features or behaviors spark support calls.
- Testing costs spiral from separate device farms and test rigs.

"I think platform-specific teams are leaving money on the table and leaving velocity on the table," says Matt Dyor, Google's product manager for KMP.[14]

"I don't see the point of maintaining separate teams," agrees Ian Hickson, who co-founded the Flutter cross-platform framework at Google. ***"Why not put the whole team on one thing? You'll be more productive."***[15]

SOME ADDED COSTS ARE HIDDEN DEEPER

Amazon supports a mind-boggling ecosystem, with thousands of engineers working in-house and for different partners, suppliers, and customers. Over a year, Amazon might have 100 or 200 contributors to a single app, some with no mobile experience. Moreover, it has identified a "coordination tax" or the overhead of keeping so many people in sync.

"A huge amount of effort goes into coordinating efforts between teams, and it scales with the size of the team," notes Erin Brasch, a principal engineer at Amazon responsible for the Prime Video app. ***"If you have 10 people, you get 10 developers' worth of output. But if you have 100 people, you probably get 30 developers' worth of output."*** [16]

Software development doesn't scale in a linear way. The more people involved, the greater the drag from communications, context-switching, and meetings.

Everyone from Fred Brooks in "The Mythical Man-Month" [17] to Amazon founder Jeff Bezos has made this point. The latter proposed that any team should be small enough to be fed with two pizzas.[18]

A SHADOW ON STRATEGY

High-cost native development can also cast a shadow on strategic decisions. Companies may delay entering new markets or shelve new features because implementing them in multiple codebases costs too much. They may avoid new platforms that could yield valuable users.

This conservative approach limits innovation and narrows the choices available to consumers. In short, native development slows everything down, ties up talent, and makes an organization more cautious than it needs to be.

CASE STUDY: WHY ONE STARTUP WILL NEVER BUILD NATIVE APPS

The company Devicethread is a hospitality startup bringing digital smarts to the hotel business.

Founder Sandeep Bhat faces a daunting task: abstracting away the messy complexity of smart devices such as locks, fridges, lights, and thermostats. Each device has its own API and protocol. Some are easy to work with, and others are opaque.

His goal is to give operators ***"a single pane of glass to control everything in the hotel."*** [19] For now, devicethread is using web-based apps. If and when mobile apps are necessary for guests, Bhat is firm: They will go cross-platform. Native development is not an option.

"Why would I take on all the pain I already have with hardware APIs and then do the same on mobile?" he says. ***"We can't afford to build the same thing twice. You have two apps, and they slowly start to drift apart. Now you need two teams, two testers, two sets of tribal knowledge. It's a nightmare."***

In past ventures, Bhat used Flutter with success. ***"Cross-platform is the only sane option for startups,"*** he says. ***"I want to build once and deploy globally."***

He's just as pragmatic about hiring. ***"You have a huge pool of web developers. Why would you limit yourself to the small group that does native development?"***

For devicethread, cross-platform development is the best strategy to stay lean and sane.

Challenge 5: Regulations have unintended consequences

Mobile technology is essential for billions of people around the world, connecting us to work, leisure, friends, and family. We carry our devices everywhere and use them for hours every day.

WHEN TWO PLATFORMS LEAD THE MARKET

Despite its significance, the mobile market today is primarily shaped by just two major platforms. Developers face high switching costs, restricted distribution channels, and minimal influence on platform roadmaps.

These structural inefficiencies suppress innovation, competition, and meaningful choice for developers and consumers alike.

Governments are stepping in with legislation that aims to open up the market. The E.U.'s Digital Markets Act (DMA)—and similar efforts in other jurisdictions—are forcing Apple and Google to allow third-party app stores.

"The DMA is just the beginning," says one European automotive executive. ***"Similar legislation is likely to emerge globally, forcing platform providers to open app distribution and APIs. This will improve developer access and grant OEMs more freedom to innovate."*** [20]

HELPFUL REGULATIONS CAN ACTUALLY HURT

However, there is a paradox: Regulations that lawmakers create to help developers can end up making their lives harder. While new rules promote competition, they also introduce more complexity and risk:

- Multiple app stores to contend with
- Alternative payment systems to integrate
- Different rules for compliance, depending on jurisdiction
- Various security models across channels

These unintended consequences add another burden, especially for smaller developers. Almost all of our sources agree that regulations will likely continue to grow. However, not everyone sees this as a clear win.

"There's always going to be more regulation," says Flutter co-founder Hickson. ***"But the problem with regulation is that you can't simply regulate a competitor into existence."*** [21]

Governments may widen the frame, but developers still need someone to show them what's possible. Sometimes, that someone is a community-led alternative to a platform default.

Challenge 6:
Different regions need different approaches

People in different regions tend to prefer different devices, often due to cost. These regional differences add yet another layer of complexity for any app developer.

IN SUB-SAHARAN AFRICA, MOBILE APPS ARE NOT POPULAR

You cannot succeed in Africa by repeating what you do in America, notes market consultant Bola Afuye. Originally from Nigeria, Afuye now runs Perform Africa Partnerships near London to help fintech and telecom firms do business in sub-Saharan nations. With more than 1 billion people now, that region is projected to reach 2 billion by 2050.[23]

For many people in the region, non-GMS Android phones are the only affordable choice, Afuye notes. Storage is precious on those devices, sometimes only 16 gigabytes. That is barely enough for half a dozen apps beyond the OEM defaults. So, instead of deleting emails, photos, or other apps to make room for a new app, most people use the Web instead.

CASE STUDY: AN ALTERNATE SERVICE FOR MAPS

The Overture Maps Foundation (OMF) is building open source map data to give developers another option for creating their own map apps. The next challenge is to help developers see how that fits into their roadmap.

OMF Executive Director Marc Prioleau sees open maps as a way for developers to regain control over their proprietary data.

"People have strong preferences for what map service they like, and they can be rather unforgiving. They want a seamless, unbroken experience," he says. ***"If I route you perfectly, you think, 'Well, that's your job.' If I route you a little bit wrong, you'll remember that the rest of your life."***[22]

That means there is a constant pressure to obtain high-quality, up-to-date map data.

Apple and Google have invested billions into their map services. These platforms also collect location searches from developers, which they then monetize for their own advertising and services, such as Waymo robotaxis. This dynamic can limit opportunities for independent developers.

"If you give developers access to high-quality open maps, you can help them build something themselves and take control of their destiny," notes Prioleau.

Projects like OMF help developers see beyond the platform gatekeepers. Regulations may redraw the map, but developers still need open services to see a different path forward.

"Some developers think they absolutely need an app to go to market, but I tell them that's foolishness," says Afuye. *"I've seen web services amass millions of users by URL and then struggle to reach 10,000 downloads of their app. The most sensible thing to do is a website that's mobile compatible."*[24]

GEOPOLITICS CAN GET IN THE WAY

Russia is off limits for Western companies since the start of the Russo-Ukrainian war. As the world splits into separate camps, not every app is welcome on the other side.

It is not easy for outsiders to sell mobile apps in China. Spanish game publisher Curro Alvarez opened an office in Shanghai to introduce his games there. However, he noted that doing business in that region was different.

"You need to offer your partner the source code, so they compile it and launch the game with their brand," he says. *"For each dollar that is generated there, I think we receive less than 10 cents."*[25]

Still, Alvarez is enjoying success in other regions. In South America, he found that less than 10% of consumers have credit cards. So, he worked out a way for buyers to pay through their mobile providers. Now he's exploring how to get his games into cars and planes.

The key is for developers to recognize regional differences and take a fresh look at the possibilities beyond the big two platforms.

AI makes us rethink everything

In addition to every other challenge facing developers, there is AI.

"The barrier to entry to writing code is dropping drastically every day," says Andrew Denta, a Y Combinator grad and startup advisor. *"What used to take a day now takes an hour."*[26]

AI does much more than help developers code faster. It challenges teams to take another look at everything from tooling to teams to what it means to write software.

- *"AI is forcing businesses to radically rethink the way they do software development,"* says Ian Cairns, CEO of the mobile studio Freeplay.[27]
- *"The future of software development is being shaped by trends that prioritize efficiency, intelligence, and adaptability,"* says Igor Fedulov, CEO of the software firm Intersog.[28]
- *"There's a new kind of coding I call 'vibe coding', where you fully give in to the vibes, embrace exponentials, and forget that the code even exists,"* says Andrej Karpathy, co-founder of OpenAI.[29]

Everyone is watching AI, but no one has a clear view of where it leads, for mobile developers or anyone else.

In mobile security, more channels mean more risks

Until recently, the two major app stores acted as the primary channels, promising a certain baseline of security. However, with millions of apps to monitor, some bad actors still slipped through the blind spots.

"App stores use curated listings, user reviews, and automated scans to give users confidence that an app is 'safe'," says Ted Miracco, CEO of the mobile security firm Approov. ***"But many of these checks are superficial and won't stop more capable adversaries."*** [30]

Now even that fuzzy baseline is disappearing. The E.U.'s DMA and similar policies around the world are forcing Apple and Google to allow third-party app stores and alternate payments. However, more channels mean more risks.

"Where there's complexity, there's opportunity for hackers," notes Miracco. ***"The mobile app is very exposed, and the device is not controlled."***

He warns that hackers can clone or modify a mobile app or write scripts that mimic a genuine app to interfere with the device or install malware. Every new store, channel, or payment system adds another surface to protect.

Developers must safeguard their own signing keys, validate update mechanisms, and make sure that users are not installing compromised builds.

Traditional defenses, such as obfuscating secrets, no longer hold up. People can download, disassemble, and scrape apps to access API keys, encryption credentials, and other embedded secrets.

"So where do you focus your security budget? Do you try to build high walls around the mobile app? Is that where the valuable data is?" asks Miracco. ***"No, the valuable data is in the cloud."*** To protect that data, he says, you need to continuously monitor and validate every device, every app, and every transaction.

Cross-platform development does not erase every security risk. However, it does reduce the cost of managing separate codebases, which frees up resources to invest in stronger defenses.

As app distribution becomes more open, developers can no longer rely on someone else to take care of security. That job is now squarely in their hands.

Cross-platform development brings the future into focus

After years of working with siloed platform teams, many mobile leaders can now see a better way forward. Using a cross-platform framework, such as Flutter or React Native, enables developers to save time and money, consolidate teams, and most of all, build once and deploy everywhere.

This approach was once considered a compromise that led to less-performant apps with quirky user interfaces (UIs). However, over the past decade, these frameworks have evolved and matured. Today, nimble companies use them to outpace competitors still using traditional native development.

Proven productivity

Cross-platform development delivers productivity gains through every step of the software lifecycle, from onboarding new developers to fixing bugs. Here is how:

- **More efficient onboarding:** New developers learn one codebase, often in a language they already know.
- **Faster implementation of new features:** Core business logic is written once and shared across platforms.
- **Streamlined releases:** Updates and bug fixes deploy at the same time for all.
- **Version syncing:** Releases for different platforms stay compatible and in sync.
- **Reduced testing:** Shared code means fewer test scenarios and edge cases.

Since time is money, all these benefits add up to lower costs for development. In fact, a compelling consensus of experts and published sources pegs these savings at somewhere between 30% and 50%.[31]

Some find this number low. After all, if a company goes from two 100% separate codebases to one 80% shared codebase, would not that productivity gain be closer to 80% than 30%? One developer guesstimated the gain was 2X; another said, in his experience, it was 3X.[32]

“Those estimates you’ve got around 30% are credible,” agrees Dyor, Google’s KMP product manager.[33] However, he cautions that measuring developer productivity is not an exact science, and he hopes to see further research in this area. So do we.

Trimming app development budgets

Developing a basic app with geolocation, payments, and integrated social media likely costs $20K to $50K, estimates Software Executive Ignor Trunov. A more complex app that supports A/V, multiple languages, and real-time sync with multiple screens may come in between $75K and $250K, he says.[34] Of course, these are just approximations.

The precise budget for any mobile app depends on many factors, such as complexity, need for compliance, number of target platforms, testing requirements, UI/UX, and use of phone hardware, such as the camera or near-field communication (NFC).

Whatever those numbers look like, shaving one-third off that total looks good to everyone.

A 6-for-1 deal, or better

Cross-platform development has been called a 2-for-1 deal because teams get both Android and iOS apps from the same codebase.[35] If they support another platform, such as the Web or Windows, they get a 3-for-1 deal. Finally, if they support the full array of Android, iOS, Linux, macOS, the Web, and Windows, this can run all the way up to a 6-for-1 deal.

"One of the projects I'm working on in my 'spare time' is a 2D game for desktop, mobile, and possibly the Web," says Flutter co-founder Hickson. ***"So I'm targeting six platforms, and there's no native code there at all."***[36]

Seeing the number of platforms on the landscape, developers may be able to squeeze even more out of a cross-platform approach at some point. Imagine getting a 12-for-1 deal by adding support for AOSP, Fire OS, HarmonyOS, Meta Horizon OS, watchOS, and Wear OS.

No company could likely justify paying for 12 separate native teams. Supporting that many platforms would only be possible with a cross-platform approach to make it cost-effective.

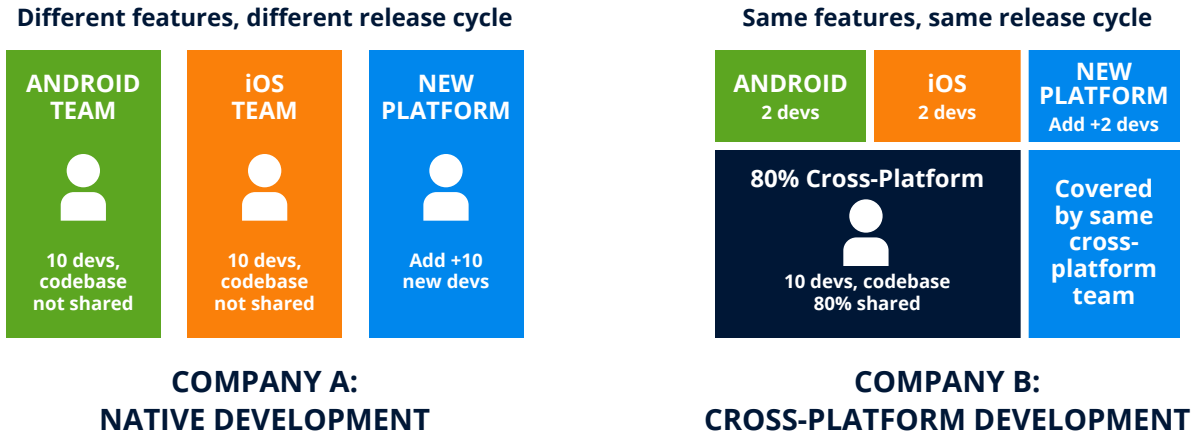
Consolidated teams, sharper focus

Adopting a cross-platform framework changes how companies structure development teams and allocate budgets. IT teams are routinely pressed to do more with less, and this is one way to actually achieve that.

FIGURE 3 shows two typical companies with existing Android and iOS versions of their apps. Now both are drawing up their budgets to add support for a third platform. Notice how:

- Company A needs to add another native team of 10 more developers, for a total headcount of 30 for three platforms.
- Company B can use the same cross-platform team of 14 developers and add just two more native developers for a total headcount of 16, becoming 46% smaller.

FIGURE 3: PRODUCTIVITY GAINS FROM CROSS-PLATFORM DEVELOPMENT



Source: The Linux Foundation, courtesy of Gordon Graham

Yet, Company B can deliver the same features across all platforms, with shared release cycles and far less duplicate effort. At the same time, Company B can access the unique capabilities of each platform when required.

These savings fall within the 30% to 50% productivity boost documented by industry sources.

Structuring teams this way generates significant gains in efficiency, flexibility, and responsiveness. This also lessens risk; if the new platform does not gain traction, Company B is only risking two new hires, while Company A is risking 10. That also means that Company B reaches the break-even point on the third platform with far fewer sales than Company A.

Any way you look at it, these practical and economic advantages show why the old model of separate Android and iOS teams is giving way to a cross-platform approach.

"I got interested in the frameworks because I wanted to support more platforms faster," says Hyo Jang, creator of the popular in-app payments library React Native IAP.

"Most business apps today benefit more from fast iteration and A/B testing than from native performance," he adds. ***"Rewriting the same UI and business logic twice slows teams down—so for most, cross-platform is a better fit."*** [37]

When to use cross-platform development

Some apps are a natural fit for cross-platform development. If the use case is simple and speed matters, this approach gets an app to market faster and cheaper.

FOUR CASES FOR CROSS-PLATFORM

Some apps lend themselves very well to cross-platform development:

- Minimum viable products (MVPs) or early prototypes
- Apps with simple UIs
- Apps with little need for native features
- Apps with users on both Android and iOS

CASE STUDY: SHOPIFY WINS WITH CROSS-PLATFORM

Back in 2020, Shopify—a leading e-commerce platform—made a pivotal decision: It would migrate all of its mobile apps to the cross-platform framework React Native. The company hoped to become ***"more effective as a team"*** by consolidating ***"iOS and Android development onto a single stack."*** [38]

Rather than dictating a timetable, engineering leaders let each team decide when and how much code to move. The transition was steady. By late 2024, the move was complete: 86% of the codebase was shared across platforms, and only 14% remained native.

Along the way, Shopify dropped 1.8 million lines of obsolete code. The apps still run fast, with reported screen loads below 500ms (P75), and both Android and iOS versions now ship with complete feature parity.[39]

Shopify's Director of Engineering Mustapha Alli says this approach benefits everyone: current engineers, new hires, and the company as a whole.

"This flexibility not only opens up more growth opportunities for engineers," he says, ***"but also increases staffing flexibility and enables teams to accomplish more with the same number of developers."*** [40]

For one of the world's most respected e-commerce platforms, the move to cross-platform development was a strategic win for productivity, performance, and people.

"If you're a fast-moving startup, hitting the market with a working product is your top priority," says a recent article on the pros and cons of React Native. Using a framework to build your MVP ***"allows you to see if your idea is, in fact, viable while spending the least amount of time and money."*** [41]

Some apps have users divided between Android and iOS. In this case, cross-platform development can generate apps that look and work essentially the same on both platforms. That keeps both versions in sync with features and releases. It also lightens the load on tech support and bug fixing and eliminates complaints about missing features.

Cross-platform frameworks still have some limitations. However, for many teams, these offer the most sensible way to write once and deploy everywhere.

When not to use cross-platform development

Despite its benefits, cross-platform development is not for everyone. Some groups, such as OMF, never build mobile apps. Others deliver core infrastructure, back-end services, or libraries that sit below the UI, so they sidestep most platform issues.

If an app needs lots of animations, A/V, and frequent calls to native APIs for hardware, such as the camera, GPS, and NFC, native development can still offer better performance today.

THE DILEMMA OF LEGACY APPS

Some teams do not begin with cross-platform in mind. They might start with an Android app, build some traction, and then want to add an iOS version.

"In that case, it's probably easier to develop a tailored experience for the new platform," says Richard Sikang Bian, an advisor to Open Infra Asia who has worked at Square and Microsoft. ***"If you build your new app with a cross-platform framework, you'll end up with two stacks anyway."*** [43]

In that scenario, Bian recommends staying native and optimizing each app for its own platform. The same applies to any firm with legacy apps. Converting those to cross-platform requires significant effort upfront, so the savings will not start to appear for some time.

Some companies do not need cross-platform

Some firms have no need for mobile apps, or no need for more than one platform.

For example, one product line from Siemens Digital Industries Software is mathematical models that developers use mainly in architectural rendering apps. No GMS? Odd screen? The models do not care; they sit below the application layer.

The company's only cross-platform challenge is testing, since all the models must work smoothly on Android, iOS, Linux, macOS, and Windows.

Another example: The remote-controlled Louma Crane helps get challenging shots for film and TV productions. The Oscar-winning system includes hardware and software tied together with a Google-free version of Android from eFoundation.

And it runs on a specific Samsung tablet with a private network, no Internet connection, and no services like maps or payments. Since the entire stack uses only one platform, Louma has no need for cross-platform development.

So cross-platform frameworks are irrelevant to outlier mobile products like these.

Cross-platform frameworks at a glance

Four popular cross-platform frameworks that developers use in mobile development today are Flutter, KMP, NativeScript, and React Native. All four are open source and free to use, and each has attracted a loyal following.

Flutter (from Google), KMP (from JetBrains), and React Native (from Meta) are strategic investments by major tech firms to build loyalty to specific platforms, ecosystems, or toolsets. NativeScript is a community-driven project under the OpenJS Foundation—and the Linux Foundation—with no single commercial owner.

The following table compares these four frameworks across some key dimensions such as launch date, sponsors, major users, native language, and tooling. All four frameworks can output apps that run on Android with or without GMS, iOS, and the Web. Support for emerging platforms, such as HarmonyOS or Fire OS, varies.

MAJOR CROSS-PLATFORM FRAMEWORKS, 2025

	Flutter	KMP	NativeScript	React Native
Launched	2017	2017	2015	2015
IP holder	Google	JetBrains	OpenJS Foundation / Linux Foundation	Meta
Ecosystem	Backed by Google, strong tooling, large plug-in library	Backed by JetBrains, fast-growing, strong IDE integration	Backed by OpenJS, smaller but active community	Backed by Meta, mature, broad npm ecosystem
Major users	• Alibaba • BMW • eBay • Google Pay • The New York Times • TikTok	• Autodesk • Forbes • Google Docs • McDonald's • Netflix • Philips Hue	• Debeka • eLichens Air • FanFiktion.de • Globe Telecom • SAP Enterprise • Vocal Image	• Bloomberg • Instagram • Pinterest • Shopify • Uber Eats • Walmart
Native language	Dart	Kotlin	JavaScript / TypeScript	JavaScript / TypeScript
Tooling / IDE	VS Code, Android Studio	IntelliJ, Android Studio	VS Code, Chrome DevTools. JetBrains	VS Code, Chrome DevTools (via Expo)
More info	https://flutter.dev/	https://kotlinlang.org/docs/multiplatform.html	https://nativescript.org	https://reactnative.dev/

Sources: See note 44.

More frameworks on the horizon

While the established frameworks are popular today, more open source, cross-platform frameworks are on the horizon. These three emerging frameworks are worth watching: the Godot game engine, Lynx from TikTok's owner, and Tauri.

GODOT, THE UP-AND-COMING GAME ENGINE

Named after the celebrated play "Waiting for Godot," this free, open source game engine can create 2D and 3D games for Android, iOS, Linux, macOS, the Web, Windows, and AR/VR headsets. It uses its own scripting language (GDScript) as well as C#, C++, and visual scripting. The Godot Foundation hosts the engine in the Netherlands.

Godot got a major boost in 2023 when Unity introduced controversial runtime fees. In protest, game developers rallied around Godot and gained wide attention. The backlash forced Unity to reverse course, fire its CEO, and cancel the fees.[45]

Since then, Godot has attracted new funding from major tech firms and studios interested in more transparent, community-driven tools.

For more info, visit **https://godotengine.org/**

LYNX FROM TIKTOK'S OWNER

ByteDance launched Lynx in 2024. It is an open source framework that the company designed to rival React Native in performance, flexibility, and cross-platform support. The company built it in Rust and rendered it with Skia. Lynx aims to give developers a unified toolset to write high-performance apps for Android, iOS, the Web, Windows, and macOS.

Lynx avoids the JavaScript bridge, opting instead for a zero bridge architecture that compiles layout and logic to native code. This reduces runtime overhead while still enabling developers to write UIs in JSX—a JavaScript-like syntax for building UIs—and to reuse business logic across platforms.

ByteDance is already using Lynx in production to power TikTok's search panel, Studio app, in-app shop, and major live event pages. This validates Lynx's performance on high-traffic features. The company is actively recruiting contributors to expand its open source ecosystem. The roadmap for 2025 includes full VS Code plug-in support, expanded documentation, and community governance.

For more info, visit **https://lynxjs.org/**

TAURI FOR SECURE, SUSTAINABLE PERFORMANCE

First released in 2022, Tauri is an open source framework for security-conscious, privacy-respecting, and environmentally aware software engineers. As of 2025, the stable V2 release supports Android and iOS, with support for Linux, macOS, and Windows coming soon.

Tauri enables developers to build lightweight, secure apps using any front-end framework that compiles to HTML, JavaScript, and CSS. For back-end logic, developers can use the programming languages Kotlin, Rust, or Swift.

The project emphasizes small binary sizes, low memory use, and minimal system access. All these features appeal to teams focused on sustainability and performance.

For more info, visit **https://tauri.app/**

Open source expands what is possible

Cross-platform frameworks, such as Flutter, React Native, and KMP, are open source at their core. Their success would not be possible without vibrant global communities that maintain and extend these tools every day.

Open source development has fueled a growing ecosystem of libraries, plug-ins, and software development kits (SDKs). These tools reduce complexity, streamline platform differences, and give developers access to prebuilt components for everything from authentication to payments.

***"This generation of programmers was not brought up on proprietary software*,"** says Michal Pierzchala, principal engineer with Callstack, which specializes in cross-platform development with React Native.

"When we started coding, all the tools were open. There was Linux. There was GitHub. There was version control. This is why we're pushing a lot of our work to open source. We know it helps us, it helps others, and it's like a social proof of expertise for Callstack." [46]

This shared innovation accelerates progress. Instead of waiting for a roadmap, developers can contribute directly or choose from community-maintained alternatives that already solve common problems.

Just as important, open source offers strategic independence. Teams are not locked into a proprietary vendor's pricing, priorities, or permissions. They can fork a tool, modify it, or swap it out entirely when business needs change. In this way, open source expands the potential for today's frameworks. It helps developers build for more environments, adopt new technologies faster, and stay resilient as the mobile landscape evolves.

This flexibility is key to long-term success. As the market diversifies and new platforms appear, open source tools—and the communities behind them—will be the engine that delivers productivity and choice.

Yet, even with strong communities and mature frameworks, developers still encounter gaps, especially when it comes to supporting new platforms and alternate services.

Cross-platform frameworks still have gaps

Despite their widespread use, today's cross-platform frameworks still fall short of the vision of "write once, run everywhere." Most offer only limited support for emerging platforms and alternatives to the platform default services.

Limited support for new platforms

Most frameworks focus squarely on Android with GMS and iOS. That leaves major blind spots:

- **Non-GMS Android phones**, common in Africa, China, and India
- **Emerging platforms**, such as Fire OS, HarmonyOS, and Meta Horizon OS
- **Specialized targets**, such as in-car systems, smart TVs, and AR/VR headsets

For developers trying to reach millions of new users—or expand into new categories such as AR/VR headsets, connected cars, or smart TVs—these gaps are hard to ignore.

Smart TVs, for example, are not mobile devices, but users often reach for their phones or tablets while the TV is on. While screen sharing is seamless on regular Android and iOS phones, it is not often available on non-GMS devices. Sharing a mobile app to the big screen feels natural, but not every platform supports this out of the box.

Locked into default services

Cross-platform frameworks also tend to assume that all developers want to use the platform defaults for essential services, such as:

- **Authentication**
- **Cloud storage**
- **Payments**
- **Mapping**

This narrows the options for developers and consumers alike. Teams often face vendor lock-in, a poor service fit for their region, or extra work to integrate third-party alternatives.

Even popular features, such as alternate payments or region-specific maps, often require custom native modules. This breaks the promise of a single shared codebase.

A thriving ecosystem helps fill the gaps

Fortunately, the open source community and commercial vendors are stepping up. They have created a vibrant ecosystem with tools, plug-ins, and modules to help bridge the gaps in platform and service coverage.

These solutions show the power of shared interfaces and modular add-ons. Some provide unified APIs for core services that work across platforms. Others offer hardened solutions for compliance or security.

For example, Approov's mobile app security tools integrate quickly into several frameworks, including HarmonyOS. This is just one of the four useful contributions from the ecosystem that appeared in the sidebar.

One ambitious open source project aims to go further. The OMH seeks to go beyond patchwork fixes to deliver a consistent developer experience across platforms and services. The goal is to save developers time and give consumers more choices.

Some contributions from the ecosystems

Here are four helpful contributions from the mobile ecosystems that provide authentication, robust security, in-app payments, and in-app messaging.

Each of these four showcases the power of abstraction layers that hide platform complexity and provide consistent APIs to developers. Their popularity proves the need for solutions that work across platforms and service providers.

APPAUTH FOR ANDROID

- Use case: OAuth 2.0 and OpenID Connect authentication flows
- Why it is notable: Developers can avoid vendor-specific SDKs by using standard endpoints
- Abstracts: Any standards-compliant OAuth2/OpenID provider, including Auth0, Google, Microsoft, and Okta
- Docs: **AppAuth GitHub**

APPROOV MOBILE SECURITY SDK

- Use case: Runtime mobile app protection and API attestation
- Why it is notable: Provides a unified security layer regardless of back-end provider
- Abstracts: Integrating multiple API gateways and mobile back ends
- Docs: **Approov Docs**

REACT NATIVE IAP

- Use case: In-app purchases
- Why it is notable: Provides a unified API that abstracts in-app purchase mechanisms across Android, Fire OS, and iOS.
- Abstracts: Amazon Appstore, Apple App Store, and Google Play Store
- Docs: **https://react-native-iap.hyo.dev/docs/get-started/**

MULTIPAZ

- Use case: Digital identity, especially for wallets
- Why it is notable: Provides APIs to simplify creating a digital identity
- Abstracts: Android, AOSP, and iOS
- Docs: **https://github.com/openwallet-foundation-labs/identity-credential/blob/main/README.md**

An ambitious project: Open Mobile Hub

OMH is an ambitious open source project that aims to empower developers to ***"build apps once—and run them anywhere."***

The core mission is to develop a comprehensive suite of SDKs that work smoothly across all mobile environments, including iOS and every flavor of Android, with or without GMS. OMH wants to expand to cover emerging platforms, such as Fire OS and HarmonyOS, in the future.

INTEGRATES WITH EXISTING FRAMEWORKS

The OMH architecture is built on a modular and transparent layer that connects a wide array of mobile services across all platforms. Rather than creating yet another development framework, OMH provides modular APIs that integrate with existing cross-platform frameworks.

That means developers can carry on with their chosen framework with no learning curve while accessing a wider range of platforms and services. This preserves a team's investment in Flutter, KMP, or React Native and reduces friction by supporting the familiar workflows.

SUPPORTS A WIDER CHOICE OF SERVICES

This innovative design enables developers to swap in different mobile services with minimal effort. They can even switch services on the fly during an outage to avoid lost revenue.

For instance, if a developer wishes to use an alternative to Google Maps, OMH enables the developer to pick alternative map services, such as Azure Maps, Mapbox, or OpenStreetMap. This plug-and-play capability extends to other crucial services, such as authentication (supporting providers such as Google, Microsoft, Facebook, and Dropbox) and cloud storage (including Google Drive, OneDrive, and Dropbox).

This approach enables a granular service-by-service choice and ensures consistent behavior regardless of the back-end provider. OMH blocks any confusing errors or fails and passes those calls on to the proper service provider.

PROVEN COMMUNITY AND GOVERNANCE

As an open source project under the Linux Foundation, OMH benefits from a proven governance model that has successfully managed many other multi-vendor initiatives. To maximize developer reach, OMH plans to contribute SDKs to cross-platform development frameworks, such as Flutter, KMP, and React Native.

By reducing development costs, expanding choice in mobile services, and fostering a truly open and collaborative environment, OMH is fueling innovation and empowering a more interconnected and accessible global mobile ecosystem.

VALIDATION FROM MOBILE INDUSTRY EXPERTS

"I think it's a brilliant idea!" says consultant Bola Afuye, who says that African developers often complain about the high cost of creating apps. ***"I can see how this makes sense to a developer if you can do it one time and know that it works perfectly. That should eliminate the additional costs."*** [47]

"I could see that being useful if it works well and it looks native," says David Zeuthen, project lead for the OpenWallet Foundation's Multipaz digital identity function. ***"And it's got to be private; it's got to be secure. Those will be the main concerns of developers."*** [48]

"By offering a consistent, unified set of APIs for common services, OMH offers the promise to reduce costs and complexity for developers," says the vice president of engineering at a European automotive company. ***"And for OEMs, OMH would free up resources, allowing us to focus on creating differentiated experiences rather than repeatedly building standard APIs."*** [49]

"We're 100% aligned with that vision," notes Matt Dyor from Google's KMP team. ***"This allows a service provider such as Firebase to write a library once in KMP so that KMP developers can use it on Android and iOS. And then any iOS or Android developer could use that library as well."*** [50]

FEATURED AT A GLOBAL GATHERING

The OMH was featured at the first-ever Global Digital Collaboration conference in Geneva in July 2025. This event drew more than 1,000 participants from government, industry, and open-source communities to explore the future of digital identification (ID), credentials, and verifiable data.

OMH presented alongside the Google Multipaz team, showing how these two open source projects can help solve key challenges in cross-platform interoperability and digital ID.

Multipaz provides a full set of open source building blocks for digital ID, covering everything from credential wallets to verifiable data flows. This enables companies, governments, and projects to create robust ID solutions without starting from scratch.

The two projects complement each other nicely. While Multipaz handles the identity layer, OMH provides a unified API for core mobile services, such as authentication, cloud storage, maps, and payments. Together, they will give developers a modular, interoperable foundation for secure mobile apps that work seamlessly across platforms.

A WIN FOR EVERYONE

The project's commitment to interoperability addresses some key challenges for those in every role in the industry: app developers, service providers, and OEMs.

For app developers, OMH reduces the steep learning curve associated with handling multiple APIs and provides a wider choice of service providers.

For service providers, OMH opens up distribution for alternative service to millions more devices.

For OEMs, OMH supports creating devices that run alternate operating systems and services, yet still offer a compatible development experience to attract developers.

OMH is following an evolving roadmap shaped by real-world feedback, with a mission to help mobile developers save time and gain more choices across every platform.

Conclusions: Expanding our vision of mobile

Most people see just two mobile platforms: Android and iOS. However, more than 2 billion devices have been shipped with other systems, such as non-GMS Android, Fire OS, HarmonyOS, and Meta Horizon OS.

The world has changed, yet many teams still build native apps one platform at a time. That traditional model does not scale. As developers face more platforms, channels, services, and workflows, the need for cross-platform development is coming into sharper focus.

Frameworks such as Flutter, KMP, and React Native offer proven gains in productivity, cost savings, and reach. Major enterprises from Alibaba to Walmart already rely on them for mission-critical apps. With a thriving ecosystem, the vision of write once, deploy everywhere is closer than ever.

Still, current frameworks do not support every platform or service out of the box. Gaps remain in areas like authentication, cloud storage, maps, and payments. Fortunately, the open source community is helping to fill those gaps with tools, plug-ins, and ambitious projects, such as OMH.

What happens next depends on your role

- **Developers** can explore new platforms, adopt cross-platform frameworks, and contribute to open source ecosystems.
- **Service providers** can invest in cross-platform SDKs, support open APIs, and reduce integration friction for developers.
- **OEMs** can support frameworks and APIs that go beyond the dominant platforms to expand access to their devices.
- **Platform vendors** can embrace open standards, reduce lock-in, and compete on innovation and developer experience.

The future of mobile development is cross-platform, open source, and choice-driven.

The technology is ready. The community is growing. The business case is proven. We no longer have to squint to see the future of mobile. It's already in clear view.

Acknowledgments

This report is based on interviews with 24 experts in 10 countries on various aspects of the mobile industry, plus extensive research into analyst reports, articles, blogs, podcasts, and press releases. Special thanks to Bola Afuye (Perform Africa Partnership), Curro Alvarez (Viva Games), Sandeep Bhat (devicethread), Richard Sikang Bian (OIF Asia Advisory Board), Erin Brasch (Amazon), John Dallman (Siemens Software), Andrew Denta (Mangrove Technology), Matt Dyor (Google), Daniel Goldsheider (OpenWallet Foundation), Ian Hickson (Ian Hickson Consulting), Mark Hotchkiss (Louma), Hyo Jang (React Native IAP creator), Raymond Lai (Amaze app contributor), Krzysztof Magiera (Software Mansion), George McGregor (Approov), Ted Miracco (Approov), Michal Pierzchala (Callstack), Andrew Pinkus (Amazon), Marc Prioleau (Overture Maps Fondation), Oscar Valenzuela (Amazon), and David Zeuthen (Google) for being interviewed for this project.

Several others shared their views but asked to remain nameless.

Special thanks to the Open Mobile Hub team—including Ravi Belwal, Sunny Cheng, Saurabh Goyal, Daniel Ljunggren, David Lu, Kevin Niu, Dr. June Wang, and Diego Zuluaga—who contributed valuable input to this report and provided introductions to many experts.

A big thanks to the Linux Foundation team, including Mike Woster for inspiring leadership, Hilary Carter for expert editorial advice, and Anna Hermansen for most helpful coordination.

About the author

Gordon Graham—also known as That White Paper Guy—is an award-winning writer who has worked on 324 white papers for clients from Switzerland to Silicon Valley. Gordon has written on everything from choosing enterprise software to designing virtual worlds for kids and for everyone from tiny startups to major enterprises such as 3M, Google, and Verizon. Since 2018, he has worked on about 20 white papers and case studies for the Linux Foundation.

Appendix A: Installed base for other platforms

This appendix outlines the research behind our estimate of the global installed base for mobile platforms beyond standard Android and iOS. These are conservative guesstimates, based on all available market research and multiple deep research sweeps using ChatGPT and Gemini. In every case, we rounded down the numbers to remain conservative.

GLOBAL INSTALLED BASE FOR OTHER MOBILE PLATFORMS, 2025

Devices	Platform	Notes on OS and Devices Counted
1 billion (a)	HarmonyOS (Huawei)	All versions in cars, laptops, tablets, and watches in China
500 million (b)	AOSP	Non-GMS Android smartphones, tablets, and watches
270 million (c)	Samsung Tizen OS	Watches and fitness devices (no TVs)
200 million (d)	Fire OS (Amazon)	E-readers and tablets (no FireTV or Echo)
175 million (e)	KaiOS	Feature phones, mainly in India
120 million (f)	watchOS	Apple Watches only
100 million (g)	OpenHarmony	Open source version of HarmonyOS in China
50 million (h)	Wear OS	Watches from Samsung and others
30 million (i)	Meta Horizon OS	VR/XR headsets and devices
TOTAL: ~2.4 billion mobile devices sold without standard Android or iOS		

a. "Huawei steps up overseas push for HarmonyOS with Grab, Emirates apps," Nikkei Asia, 22 Oct 2024. **https://asia.nikkei.com/business/technology/huawei-steps-up-overseas-push-for-harmonyos-with-grab-emirates-apps**

b. Sales reports and analyst data for non-GMS Android device shipments plus known volumes for AOSP forks such as LineageOS yields conservative guesstimate of 500 million active smartphones, tablets, and watches.

c. "Tizen OS 270m Devices Run on Samsung Platform," Invidis, 23 June 2024. **https://invidis.com/news/2024/06/tizen-os-270m-devices-run-on-samsung-platform**

d. "Worldwide Tablet Shipments Increase by 20.4% in Q3 2024," IDC, 1 Nov 2024. **https://my.idc.com/getdoc.jsp?containerId=prUS52695624** and Shehraj Singh, "Kindle Users and Usage Statistics", 18 July 2023. **https://ereader.blog/kindle-statistics**

e. "Our Story," Kaiostech.com, 2024. **https://www.kaiostech.com/company/our-story**

f. 24% Apple Watch share of 155M units with 3-year lifecycle. "Wearables Market Continues to Grow Amid Shifts in Consumer Preferences," IDC, 28 Apr 2025. **https://www.idc.com/promo/wearablevendor**

g. "Chinese tech companies supported open-source project OpenHarmony…", ijiwei.com, 27 Apr 2024. **https://jw.ijiwei.com/n/816275**
h. 27% Wear OS share of 155M units outside China plus modest sales in China with 3-year lifecycle. "Wear OS, HarmonyOS to Register Strong Growth in Global Smartwatch Market in 2024," Counterpoint Research, 29 Apr 2024. **https://www.counterpointresearch.com/insights/global-smartwatch-market-2024** and "Wearables Market Continues to Grow Amid Shifts in Consumer Preferences", IDC, 28 Apr 2025. **https://www.idc.com/promo/wearablevendor**
i. 20 million in early 2023 plus sales of 10 million over past two years. "Meta Has Sold Nearly 20 Million Quest Headsets," UploadVR, 1 Mar 2023. **https://www.uploadvr.com/meta-sold-20-million-quests**

Appendix B: How to pick the right cross-platform framework

When choosing a framework, most mobile developers and team leads consider five key factors: familiarity, maturity, libraries, documentation, and community.

Familiarity

"Once developers choose an ecosystem, they typically don't plan on adopting any other technology or toolset," concluded the "State of the Developer Ecosystem Report 2024" from Jetbrains, based on a survey of more than 23,000 developers. ***"People tend to pick what they like and stick with what they know."***[51]

Most developers strongly prefer frameworks that support their existing skills and knowledge. This explains why React Native appeals to web developers already comfortable with JavaScript and React, while KMP attracts Android developers who already know Kotlin.

Maturity

A framework must demonstrate stability and real-world validation. Developers want to see years of production use and high-profile applications built with the technology. The 10-year history of React Native—and eight years for the other three—shows a solid track record.

Major enterprises now rely on these frameworks to create business-critical apps. Google uses Flutter for Google Pay and KMP for Google Docs. React Native powers Instagram, Pinterest, Shopify, Uber Eats, and Walmart.

These are not niche apps or side projects. They are massive platforms that billions of people use, which is solid proof that the cross-platform frameworks are ready for prime time.

When choosing a framework, most mobile developers and team leads consider five key factors: familiarity, maturity, libraries, documentation, and community.

Libraries

A strong framework needs a rich ecosystem of third-party libraries. Developers especially look for support with analytics, authentication, cloud storage, networking, and payments. Without them, teams must build their own, at considerable cost.

React Native benefits from the entire npm ecosystem, while Flutter's package repository continues to grow quickly. Platform-specific modules that handle integration with native SDKs are particularly important.

Documentation

Code to download from GitHub is one thing, but what about the docs to go with it? As a general rule of thumb, the more documentation, the better. A developer may not look at the docs every day—but when they do, they expect everything to be crystal clear.

"We are very serious about the quality of the documentation," says Flutter co-founder Hickson, who also worked as an editor of the HTML standard. In the early days, he recalls how Flutter project lead Adam Barth spent months working full-time writing and refining the docs. ***"I don't know any other project that has done that,"*** he adds.[52]

Community

Active communities contribute libraries, provide support, and drive the evolution of any framework. Developers often assess community health by checking activity on Stack Overflow, contributions to GitHub, and the presence of the framework at industry events and conferences.

In the end, every executive needs to choose what is right for their app, team, and company. However, they should not delay making a decision.

As one tech newsletter puts it, ***"It's a tough world in mobile apps, and only those who act quickly stand a chance. Cross-platform frameworks can provide a key benefit by bringing together speed, scalability, and affordability. Right now, you should update your mobile strategy."***[53]

References

1. "Mobile Operating System Market Share Worldwide", Statcounter, May 2025. https://gs.statcounter.com/os-market-share/mobile/worldwide
2. AOSP market share includes all uncertified variants, regional builds, and devices using forks such as Fire OS and LineageOS.
3. David Zeuthen, interviews by author, Zoom, 16 May 2025 and 6 June 2025.
4. Chris Dickert and Sabrina Fortin, "Network Overload? Adding Up the Data Produced By Connected Cars", VisualCapitalist.com, 15 March 2023. https://www.visualcapitalist.com/network-overload
5. "Global Smartphone Sales Share by Operating System", Counterpoint Research, 2 June 2025. https://www.counterpointresearch.com/insights/global-smartphone-os-market-share
6. In Figure 2, Fire OS is counted under AOSP. The 7% allocated to AOSP is consistent with analyst estimates suggesting AOSP is 8% to 10% of total Android shipments. Figure 2 is assembled from numerous sources, including:
    - "Android Market Share," SOAX.com, January 2025. https://soax.com/research/android-market-share
    - "Android vs iOS Statistics," Tekrevol.com, 2025. https://www.tekrevol.com/blogs/android-vs-ios-statistics
    - "Global Smartphone Operating System (OS) Market," Maximize Market Research, 2025. https://www.maximizemarketresearch.com/market-report/global-smartphone-operating-system-os-market/66436
    - "Global Smartphone Sales Share by Operating System," Counterpoint, 12 March 2025. https://www.counterpointresearch.com/insights/global-smartphone-os-market-share
    - "HarmonyOS," Wikipedia, March 2025. https://en.wikipedia.org/wiki/HarmonyOS
    - "Mobile Operating System Market Share Worldwide," Statcounter, May 2025.
    - "Usage share of operating systems," Wikipedia, March 2025. https://en.wikipedia.org/wiki/Usage_share_of_operating_systems
    - "Wear OS, HarmonyOS to Register Strong Growth in Global Smartwatch Market in 2024," Counterpoint Research, 29 April 2024. https://www.counterpointresearch.com/insights/global-smartwatch-market-2024
    - "Worldwide Tablet Shipments Increase by 20.4% in Q3 2024," IDC, 1 November 2024. https://my.idc.com/getdoc.jsp?containerId=prUS52695624
7. Ted Miracco, interview by author, Zoom, 22 May 2025.
8. Krzysztof Magiera, interview by author, Zoom, 20 May 2025.

9. Curro Alvarez, interview by author, Zoom, 29 April 2025.

10. Ashraf Farid, "The Overview Role of APIs in Mobile & Web Development", Upstack Studio blog, 18 December 2024. https://upstackstudio.com/blog/api-mobile-web-development

11. "2024 State of the API Report", Postman, 2024. https://voyager.postman.com/doc/postman-state-of-the-api-report-2024.pdf

12. Akshat Virmani, "Common API Integration Challenges and How to Overcome Them", Hackmamba blog, 9 August 2024. https://hackmamba.io/blog/2024/08/common-api-integration-challenges-and-how-to-overcome-them

13. Unreleased survey of mobile app developers by the Linux Foundation, April 2025.

14. Matt Dyor, interview by author, Zoom, 2 June 2025.

15. Ian Hickson, interview by author, Zoom, 10 June 2025.

16. Erin Brasch, interview by author, Zoom, 16 April 2025.

17. Frederick P. Brooks Jr., "The Mythical Man-Month: Essays on Software Engineering", anniversary ed. (Boston: Addison-Wesley, 1995, pp25–27.

18. "Powering Innovation and Speed with Amazon's Two-Pizza Teams", AWS Executive Insights, Amazon Web Services, 2023. pp 6-7. https://d1.awsstatic.com/executive-insights/en_US/two_pizza_teams_eBook.pdf?utm_source=chatgpt.com

19. Sandeep Bhat, interview by author, Zoom, 23 May 2025.

20. Name withheld by request, interview by author, email, 29 March 2025.

21. Hickson interview.

22. Marc Prioleau, interview by author, Zoom, 31 March 2025.

23. Perform Africa blog. https://performafrica.co/blog/

24. Bola Afuye, interview by author, Zoom, 13 May 2025.

25. Alvarez interview.

26. Andrew Denta, interview by author, Zoom, 2 April 2025.

27. Ian Cairns, "Generative AI forces rethink of software development process", Deloitte Insights, 1 July 2024. https://www.deloitte.com/us/en/insights/topics/technology-management/tech-trends/2024/freeplay-ceo-discusses-process-change-ai-software-development.html

28. Igor Fedulov, "How AI Is Redefining The Way Software Is Built In 2025", Forbes, 31 January 2025. https://www.forbes.com/councils/forbestechcouncil/2025/01/30/how-ai-is-redefining-the-way-software-is-built-in-2025

29. Andrej Karpathy, "There's a new kind of coding I call "vibe coding", where you fully give in to the vibes, embrace exponentials, and forget that the code even exists..." X, 2 February 2025. https://x.com/karpathy/status/1886192184808149383?lang=en

30. All quotes from Miracco interview.

31. Many published sources estimate the time savings from cross-platform development at between 30% and 50%:

- "How Much Does It Cost To Make a Mobile App in 2025?", Mobisoft Infotech, 27 March 2025. https://dev.to/mobisoftinfotech/how-much-does-it-cost-to-make-a-mobile-app-in-2025-3j99
- Sandy, "Top Cross-Platform App Development Strategies", Data Science Society, 8 August 2024. https://www.datasciencesociety.net/top-cross-platform-app-development-strategies
- Buddhika Ranaweera, "React Native vs Flutter: Which Saves More Development Time in 2025?", Blott blog, 8 January 2025. https://www.blott.studio/blog/post/react-native-vs-flutter-which-saves-more-development-time
- George McGregor, "Using Cross-Platform Development? You Need Cross-Platform Security Too", Approov blog, 8 October 2024. https://approov.io/blog/using-cross-platform-development-you-need-cross-platform-security-too
- "Pros and Cons of React Native for Cross-Platform App Development", MindK, not dated. https://www.mindk.com/blog/react-native-pros-and-cons
- "The Future of Mobile Apps: How Cross-Platform Frameworks Drive Agility?", GMTA Software Solutions, LinkedIn newsletter, 22 May 2025. https://www.linkedin.com/pulse/future-mobile-apps-how-cross-platform-ct5ec
- "How Does Cross-Platform App Development Accelerate Time-to-Market?", AppVin Technologies, LinkedIn newsletter, 13 September 2024. https://www.linkedin.com/pulse/how-does-cross-platform-app-development-accelerate-time-to-market-iyp6c

32. Hyo Jang, interview by author, Zoom, 28 May 2025.

33. Dyor interview.

34. Ignor Trunov, "How to Save on Mobile App Development", Forbes. 30 January 2024. https://www.forbes.com/councils/forbesbusinesscouncil/2024/01/30/how-to-save-on-mobile-app-development

35. Oleksandr Kruglyak, "How Hybrid Apps Development Save Time and Money for Businesses in 2025?", Triare.net, 2 April 2025. https://triare.net/insights/hybrid-apps-development-2025

36. Hickson interview.

37. Jang interview.

38. Farhan Thawar, "React Native is the Future of Mobile at Shopify" blog post, Shopify.com, 29 January 2020. https://shopify.engineering/react-native-future-mobile-shopify

39. "Lessons from Shopify's Migration to React Native with Mustafa Ali", React Native at Scale podcast, Callstack, 13 January 2025. https://www.callstack.com/podcasts/lessons-from-shopifys-migration-to-react-native-with-mustafa-ali

40. Mustafa Ali, "Five Years of React Native at Shopify" blog post, Shopify.com, 13 January 2025. https://shopify.engineering/five-years-of-react-native-at-shopify

41. Maksym Churylov, "Pros and Cons of React Native for Cross-Platform App Development", MindK.com, not dated. https://www.mindk.com/blog/react-native-pros-and-cons

42. Alvarez interview.

43. Richard Bian, interview by author, Zoom, 30 April 2025.

44. Cross-platform framework summaries assembled from many sources including framework websites and product showcases, interview sources, and Wikipedia entries.

45. Dean Takahashi, "Unity cancels its much-hated Runtime fees, opting for traditional price increases instead", VentureBeat.com, 12 September 2024. https://venturebeat.com/games/unity-cancels-its-much-hated-runtime-fee

46. "Unity announces leadership transition", press release, Unity, 9 October 2023. https://s205.q4cdn.com/200788659/files/doc_news/Unity-Announces-Leadership-Transition-2023.pdf

47. Michal Pierzchala, interview by author, Zoom, 23 May 2025.

48. Afuye interview.

49. Zeuthen interview.

50. Name withheld, interview by author, email, 29 March 2025.

51. Dyor interview.

52. "The State of the Developer Ecosystem Report 2024", Jetbrains, 2025. https://www.jetbrains.com/lp/devecosystem-2024

53. Hickson interview.

54. "The Future of Mobile Apps: How Cross-Platform Frameworks Drive Agility", Tech Notes-Technology Insights newsletter on LinkedIn, GMTA Software Solutions, 22 May 2025. https://www.linkedin.com/pulse/future-mobile-apps-how-cross-platform-ct5ec

OPEN MOBILE HUB

Open Mobile Hub (OMH) is an open-source project hosted by the Linux Foundation, designed to provide a unified framework for mobile app development. OMH acts as an abstraction layer above various Software Development Kits (SDKs), standardizing and simplifying how applications interact with underlying services, regardless of the service provider. By aligning with the Linux Foundation's mission of fostering innovation and removing boundaries, OMH leverages the power of open source to reduce complexity, encourage flexibility, and drive technological advancement in the rapidly evolving landscape of mobile and web services.

www.openmobilehub.org

THE LINUX FOUNDATION | Research

Founded in 2021, **Linux Foundation Research** explores the growing scale of open source collaboration, providing insight into emerging technology trends, best practices, and the global impact of open source projects. Through leveraging project databases and networks, and a commitment to best practices in quantitative and qualitative methodologies, Linux Foundation Research is creating the go-to library for open source insights for the benefit of organizations the world over.

facebook.com/TheLinuxFoundation

x.com/linuxfoundation

linkedin.com/company/TheLinuxFoundation

To reference this work, please cite as follows: Gordon Graham, "Expanding the Vision for the Mobile Industry: How to Gain More Productivity, Innovation, and Choice with Open Source" foreword by Mike Woster, The Linux Foundation, August 2025.