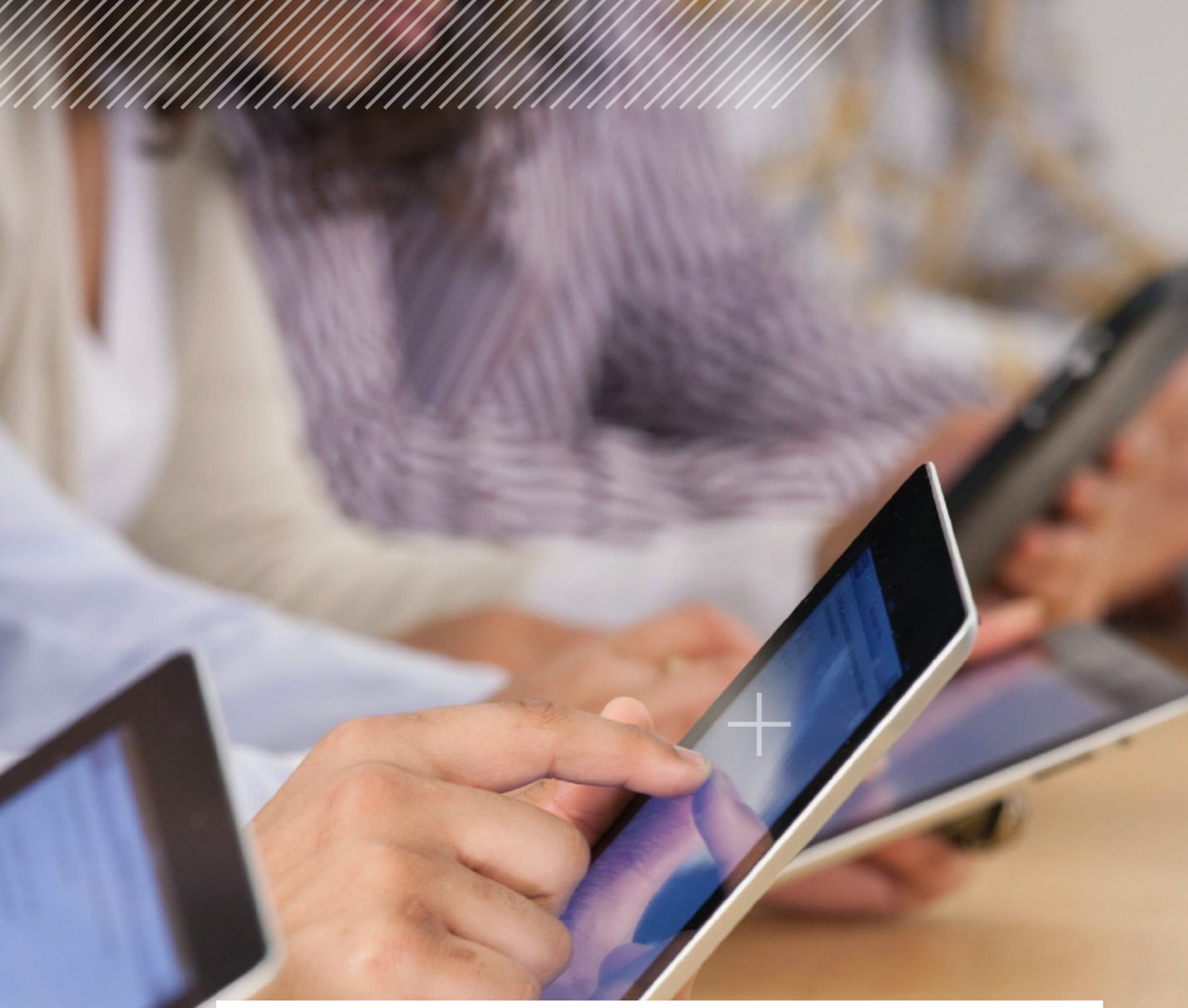




Flutter

als zentrales Framework für die
plattformübergreifende Frontend-Entwicklung



Was ist eine Flutter-App?

Flutter ist ein Open-Source-Framework von Google. Die zugrunde liegende Programmiersprache ist Dart, die von Java, C++ und C beeinflusst wurde und Ähnlichkeiten mit diesen aufweist. Sie bietet die Möglichkeit, nativ kompilierte Applikationen für das Internet, Windows, macOS, Linux, iOS

und Android mittels einer einzigen Codebasis zu erstellen. Flutter wird von Google aktiv entwickelt und genutzt. Ebenfalls gibt es eine sehr grosse Community von Open-Source-Entwicklern aus aller Welt welche das Framework weiterentwickeln und erweitern.

Marktprävalenz von Flutter

Es gibt derzeit mehr als 500'000 Apps¹, die mit Flutter erstellt wurden, Tendenz rasch steigend. Zahlreiche grosse Firmen vertrauen auf Flutter als zentrales Framework für ihre Applikationen. Dazu zählen Marken der verschiedensten Branchen, von Social-Media-Apps wie WeChat über Finanzinstitute und Banken wie Betterment und Nubank sowie Lifestyle-Apps wie Fastic, Tabcorp und trip.com bis hin zu Companion-Apps wie BMW. Selbst staatliche Institutionen wie die brasilianische Regierung nutzen Flutter für ihre Dienste.

Es nutzen jedoch nicht nur zahlreiche grosse Firmen das Flutter-Framework und passen es an ihre Applikationen an. Flutter hat auch recht schnell eine Führungsposition in der Branche übernommen. Dies lässt sich auch an den Google-Suchergebnissen erkennen, wenn man Flutter beispielsweise unmittelbar mit anderen Frameworks vergleicht, welche die gleiche Zielsetzung haben.

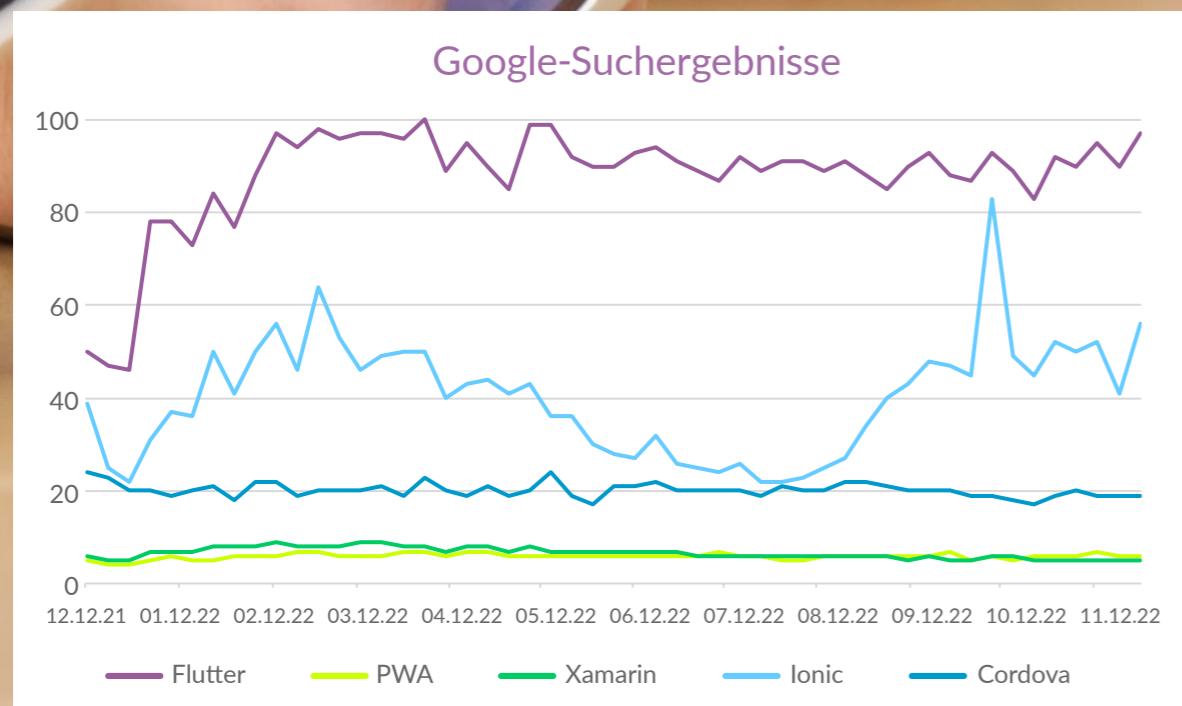


Abb. 1: Vergleich der Google-Suchergebnisse (weltweit) (Quelle: trends.google.com)

¹ <https://flutter.dev/showcase>

Was zeichnet eine Flutter-App aus?

Eine einzige Codebasis:

Das wichtigste Merkmal einer Flutter-App ist, dass sie unabhängig von der Anzahl ihrer Zielplattformen – namentlich Mobile, Web und/oder Desktop – nur eine einzige Codebasis hat. Dies kann den gesamten Entwicklungsprozess vereinfachen, da der Code für unterschiedliche Plattformen erneut verwendet werden kann. Layout-UI-Elemente können mit etwas Planung über verschiedene Plattformen hinweg wiederverwendet werden und die gesamte in der App enthaltene Geschäftslogik muss nur einmal geschrieben werden.

Weitreichende Auswahl an Widgets:

Das Grundkonzept einer Flutter-App ist die Synergie verschiedener Widgets. Widgets sind Elemente, die kombiniert werden können, um die Grafikoberfläche der App aufzubauen. Flutter bietet eine weitreichende Auswahl an Widgets, die Entwickler nutzen oder sogar ergänzen können. Es ist zudem möglich, vollständig benutzerdefinierte Widgets zu entwickeln, um ein pixelgenaues Design zu erreichen.

Schnelle App-Entwicklung:

Mit Funktionen wie Hot Reload für Plattformen kann man Änderungen sichtbar und für Entwickler sofort zugänglich machen, um Applikationen schnell zu entwickeln, zu testen und anzupassen. Widgets machen es möglich, schnell zu reagieren und neuartige Änderungen umzusetzen, ohne die App gross zu beeinträchtigen. Diese Funktion gibt es zwar auch bei anderen Frameworks, sie funktioniert bei Flutter jedoch aussergewöhnlich gut.

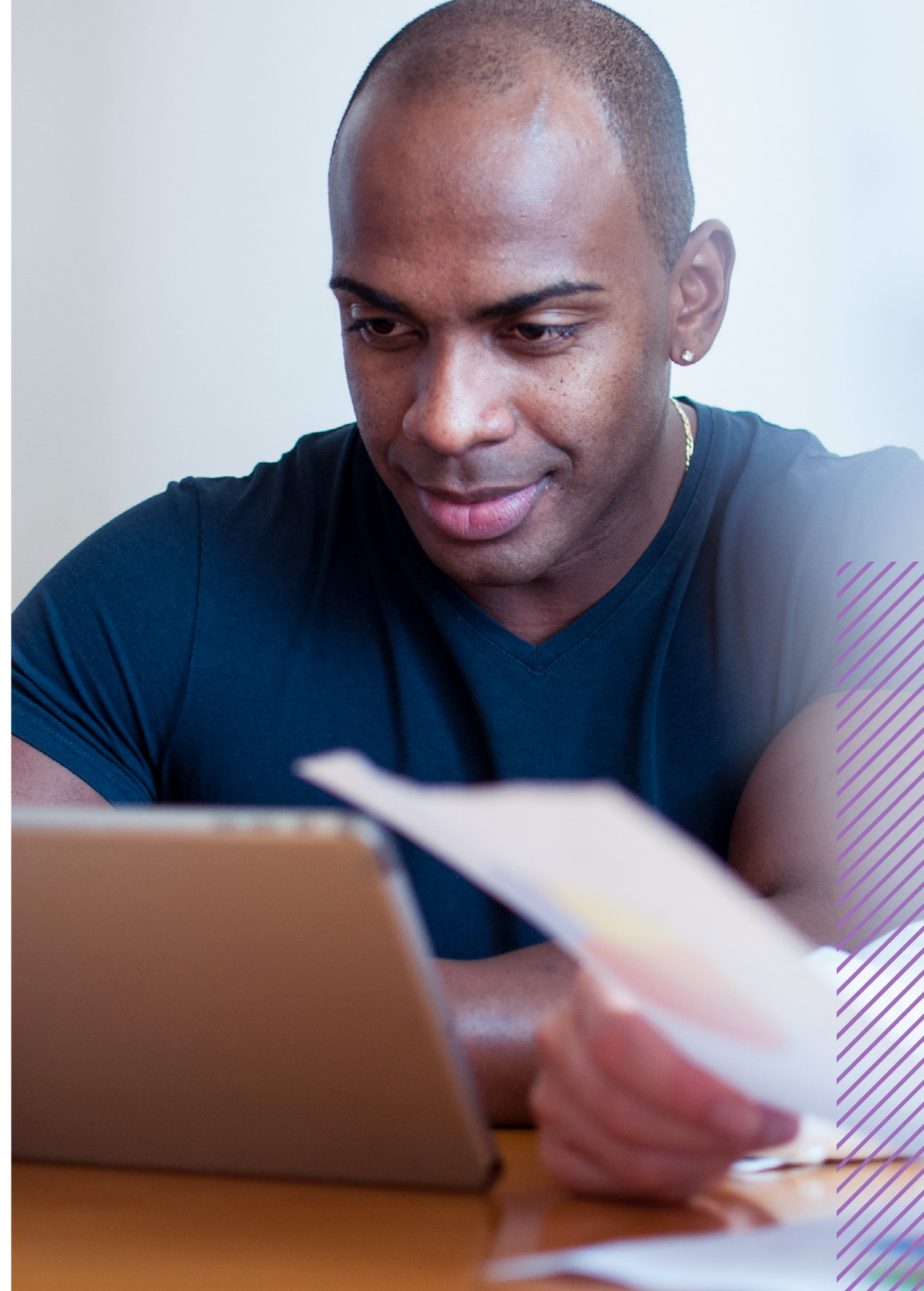
Massgebliche Unterstützung der Community:

Eine grosse Community aktiver Entwickler, das steigende Interesse am Framework und die Unterstützung von Google, das die weitere Entwicklung von Flutter voranbringt, sorgen für die Verlässlichkeit der Plattform. Eine breite Auswahl an Plugins und Bibliotheken, die durch die Community entwickelt und gepflegt werden, trägt zur raschen Beschleunigung der Entwicklung bei.

Vielfältige Plattformen

Mit dem Flutter-Framework kann der Benutzer die App für sämtliche Arten von Zielplattformen kompilieren. Flutter kann für Desktop, Web und Mobile eingesetzt werden und ermöglicht es Benutzern darüber hinaus, die Kernfunktionalität dieser Zielplattformen ganz ohne Kompromisse zu

nutzen und gleichzeitig das völlig native Erscheinungsbild und die Leistungsfähigkeit beizubehalten. Diese Funktion macht Flutter zu einem der wichtigsten Frameworks für plattformübergreifende Entwicklung, da Code-Redundanz für verschiedene Zielplattformen vermieden werden kann.





Vor- und Nachteile von Flutter



- Plattformübergreifende Leistungsfähigkeit für nahezu alle gewünschten Zielplattformen
- Schnell wachsende Community
- Open-Source-Projekt
- Riesige Auswahl an verfügbaren Widgets
- Support für Mobile, Web und Desktop
- Sehr gute Dokumentation von Google
- Durch Google genutzt, unterstützt und kontinuierlich weiterentwickelt



- Stetig wachsende Auswahl an Tools und Bibliotheken
- Neue Funktionen für iOS und Android könnten später unterstützt werden
- Junge Technologie. Dies kann sich auf die Anzahl der verfügbaren Experten und existierenden Bibliotheken auswirken
- Die Herstellung des plattformspezifischen Look and Feels erfordert zusätzlichen Aufwand
- Apps benötigen mehr Speicherplatz, da die Render-Engine mit verteilt werden muss.

Unterschied zu anderen plattformübergreifenden Frameworks

Häufig verwendete plattformübergreifende Frameworks wie Cordova oder Ionic basieren auf JavaScript (JS) und Web-Ansichten und zeigen im Grunde eine Webseite in einer nativen App an. Diese Apps werden mithilfe von HTML erstellt und dann in den Apps als Web-Ansichten angezeigt, wodurch eher eine Mobile-First-Webseite als eine tatsächliche App entsteht. Ein grosser Nachteil dieses Konzepts ist, dass zusätzlich zur Langsamkeit von Browsern und JS die Implementierung nativer Gerätefunktionen manchmal gänzlich untersagt ist. Flutter beruht, beim Gebrauch von Gerätefunktionen, auch auf Bibliotheken, welche die nativen Gerätefunktionen aufrufen. Es stehen bereits eine grosse Vielzahl Bibliotheken zur Auswahl. Gibt es keine Open-Source-Bibliothek, welche die Funktionalität abdeckt, kann sie in jedem Fall mit nativem Code durch das Entwicklungsteam implementiert werden. Einige alternative plattformübergreifende Frameworks wie z. B. React Native nutzen eine Bridging-Layer, um das in JavaScript beschriebene Layout in das native Format der Zielplattformen

zu übertragen. Dies führt oftmals zu Leistungsbeeinträchtigungen, wenn schnell viele Aufrufe über dieses Bridging-Layer gemacht werden. Flutter hingegen nutzt eine eigene Rendering-Engine. Die Engine bietet eine eigene Low-Level-API für das Darstellen des App-Layouts, der Texte und von allem anderen. Dies verschafft Flutter den Vorteil, jedes einzelne Pixel zuverlässig rendern und kontrollieren zu können.

Die folgende Abbildung zeigt Flutter im Vergleich zu einer nativen Entwicklung in Bezug auf die Code-Redundanz für verschiedene Layer der Apps. Wie bereits erwähnt, besteht der hauptsächliche Nachteil der nativen Entwicklung darin, für alle Zielplattformen eine Benutzeroberfläche entwickeln zu müssen. Es gibt Frameworks wie z. B. Kotlin Multiplatform, die das Teilen der Geschäftslogik einer App zwischen unterschiedlichen mobilen Plattformen zuverlässig ermöglichen. Die Abbildung zeigt, wie viele Codebasen für eine App notwendig sind, die für verschiedene Ziele entwickelt wurde.

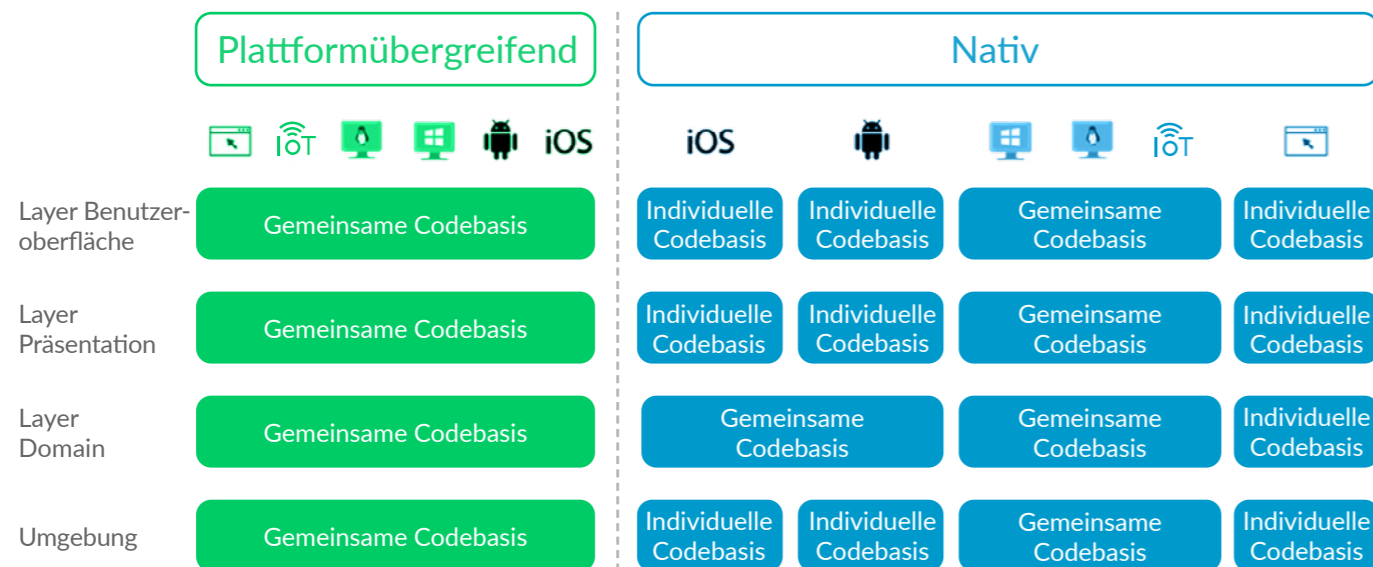
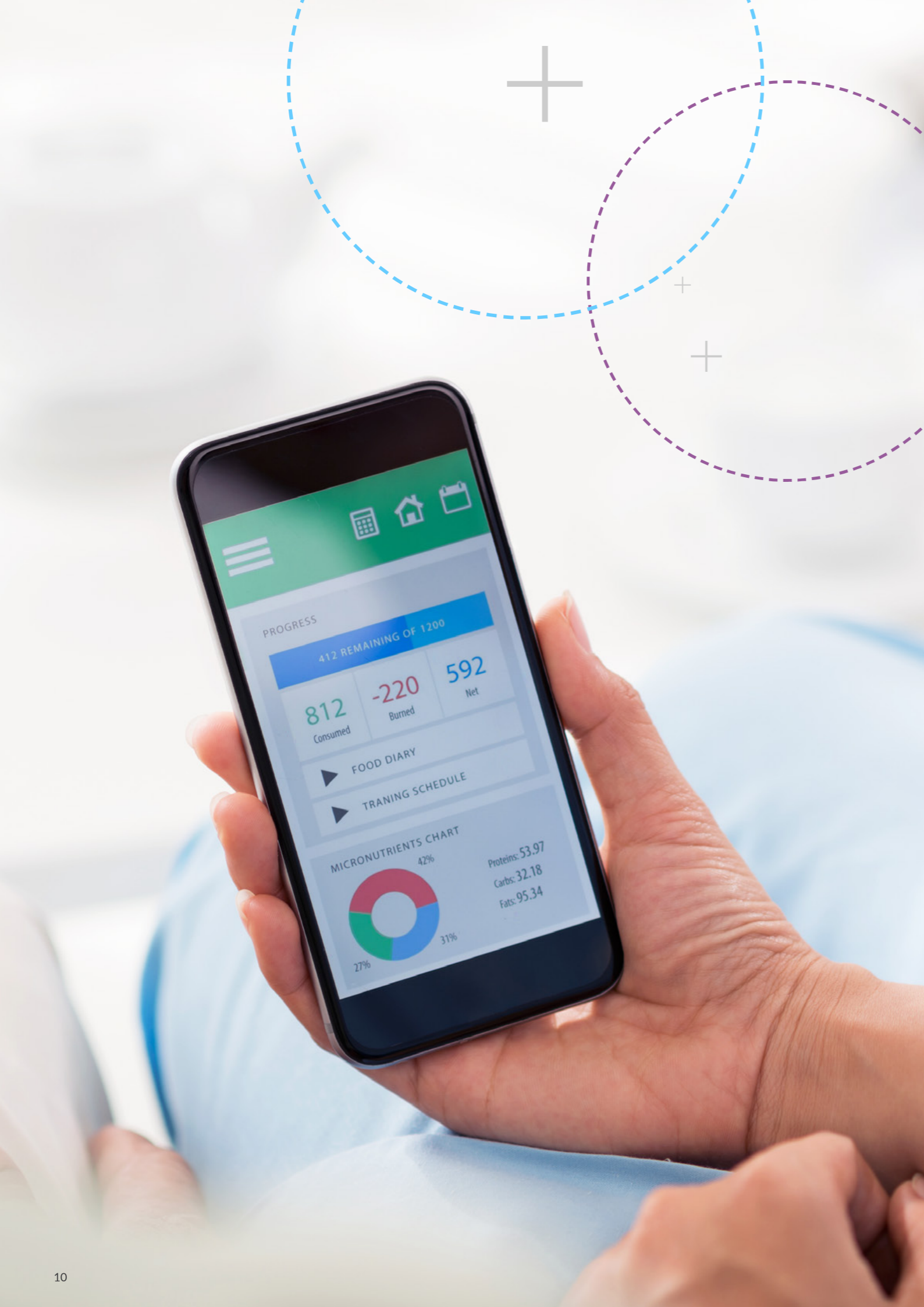


Abb. 2: Vergleich der erforderlichen Codebasis für Flutter vs. nativ





Flutter compared to other cross-platform solutions

	Flutter	React Native	Cordova/Ionic	Xamarin
macOS	x		x	
Windows	x		x	x
iOS	x	x	x	x
Android	x	x	x	x
Linux	x		x	x
Web	x	x	x	
Natives Look and Feel	x	x		x
Native Performance	x			x
Zugriff auf plattformspezifische Dienste	x	x	x	x
Hot-Reload-Support	x	x	x	x
SDK built-in testing infrastructure	x	x	x	x

Tabelle 1: Vergleich plattformübergreifender Lösungen

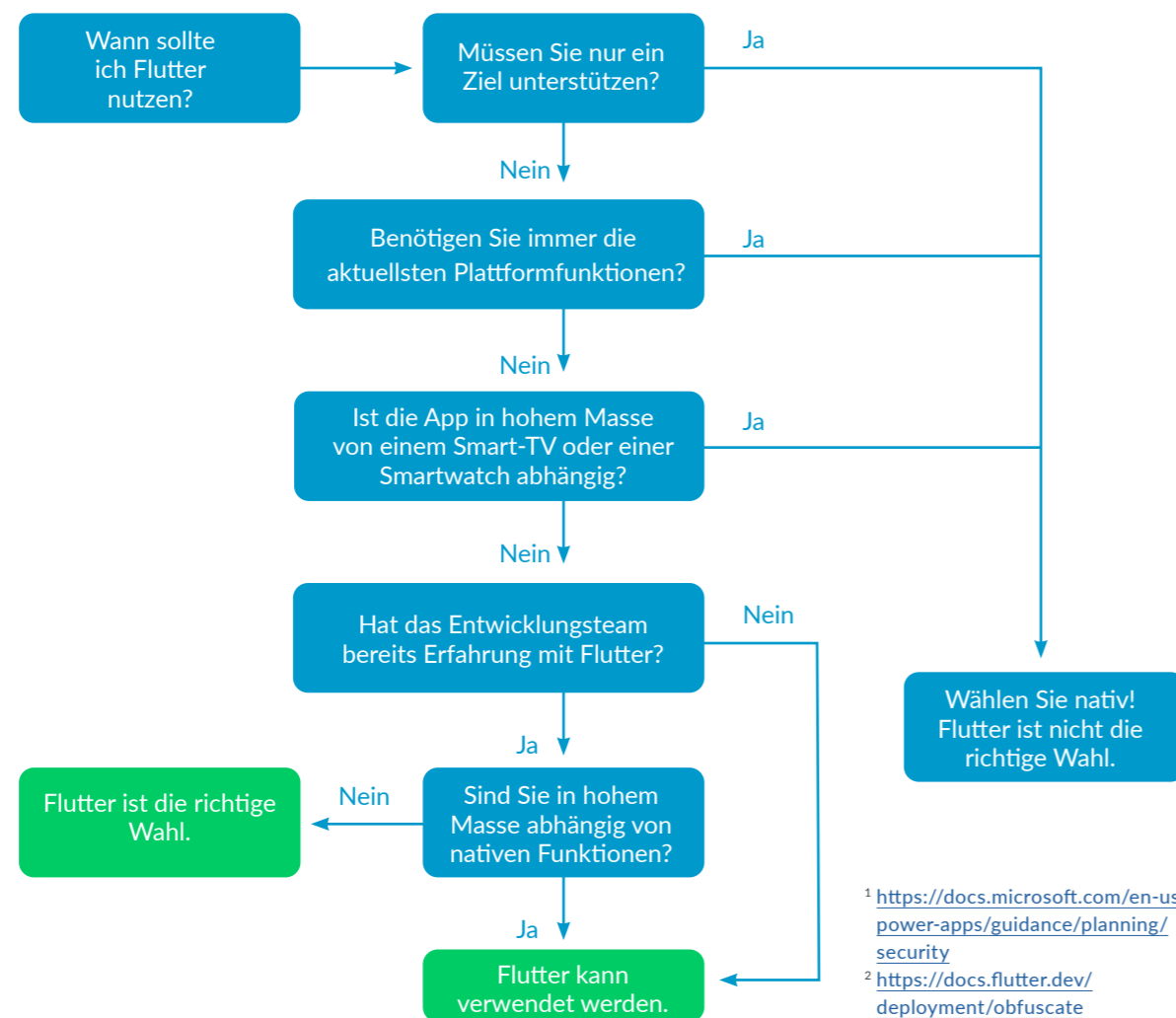
* Erfordert zusätzlichen Aufwand

Sicherheit

Flutter bietet im Hinblick auf die Sicherheit einige Tools und Funktionalitäten. Zusätzlich zum sicheren App-Design¹, das immer ein entscheidender Faktor sein sollte, ist bei Flutter-Projekten eine Code-Verschleierung möglich. Code-Verschleierung verbirgt die Funktionen, Klassen und alles weitere des Quellcodes und erstellt kompilierten Dart-Code, der das Reverse-Engineering der App zusätzlich erschwert. Diese Funktion wird aktuell für mobile und macOS-Apps² unterstützt. Die Implementierung nativer Sicherheitselemente, die durch das Zielsystem bereitgestellt werden, ist jederzeit möglich. Diese können, sofern sie nicht bereits standardmässig implementiert sind,

mittels Erweiterung der Bibliothek integriert werden und dadurch tatsächlich den Sicherheitsstandard einer nativen App erreichen. Bibliotheken aus nicht vertrauenswürdigen Quellen oder mit veralteten Implementierungen stellen oftmals das höchstmögliche Sicherheitsrisiko dar. Es ist daher wichtig, bei der Nutzung von Drittanbieter-Bibliotheken immer die Quelle zu prüfen. Gleichwohl gibt es auch ausgezeichnete Bibliotheken, welche die Sicherstellung und Aufrechterhaltung der App-Sicherheit unterstützen. Die Sicherheitsvorkehrungen von Flutter entsprechen denen anderer Frameworks und es sind keine Kompromisse bei der Sicherheit erforderlich.

Entscheidungshilfe





Geräteeigenschaften

Wie viele andere Frameworks, die auf die Entwicklung von Mobilgeräten ausgerichtet sind, hinkt auch Flutter hinterher, wenn es um die Implementierung neuer Funktionen geht, wie sie Apple oder Google jedes Jahr einführen. So ist beispielsweise die kürzlich von Apple vorgestellte Widget-Funktion für den Startbildschirm noch nicht nativ in Flutter verfügbar. Solche fehlenden

Anforderungen lassen sich immer mit dem nativen Code oder den Bibliotheken im Flutter-Projekt implementieren. In der Regel unterstützt Flutter die Implementierung aller Arten von nativ programmierten Funktionen für alle seine Zielplattformen. Auf diese Weise können fehlende Plattformfunktionen mit geringem Mehraufwand ausgeglichen werden.

Barrierefreiheit

Barrierefreiheit ist bei der modernen Softwareentwicklung ein wichtiges Thema. Daher ist es nicht überraschend, dass das Flutter-Framework alle möglichen Funktionen unterstützt, um diesbezüglich Lösungen zu finden. Am wichtigsten ist die Verfügbarkeit und Integrierbarkeit von Bildschirmlesegeräten. Dank der Umsicht der Entwickler können sehbehinderte Nutzer mithilfe von Bild-

schirmlesegeräten bei der Navigation durch eine App unterstützt werden. Das Bildschirmlesegerät kann dem Nutzer den Text vorlesen und ihm mitteilen, welche Schaltfläche aktuell ausgewählt ist oder was passieren wird, sobald die Schaltfläche gedrückt wird. UI-Funktionen wie adaptive Textgröße oder ausreichender Kontrast werden ebenfalls unterstützt.

Tests

Bei der Entwicklung hochwertiger, skalierbarer Flutter-Anwendungen zählt das Testen zu den wichtigsten Faktoren. Flutter verfügt über eine integrierte Testinfrastruktur für Komponenten-, Integrations- und End-to-End-Tests ohne Notwendigkeit einer Darstellung auf dem Bildschirm. Zudem besteht die Möglichkeit, Tests mit dem Flutter-Treiber zu implementieren. Der Flutter-

Treiber stellt eine Verbindung zur entwickelten Anwendung her und interagiert mit ihr mithilfe definierter Testfälle wie ein Benutzer. Die Software ist mit dem Framework Selenium vergleichbar, das eingesetzt wird zur Automatisierung von Webanwendungen für Testzwecke. All diese Tests können auch auf modernen CI-/CD-Pipelines wie Real Device Cloud oder BrowserStack laufen.



Erfahrung mit Flutter

Zeit-Kosten-Vergleich

Um Flutter mit anderen plattformübergreifenden Frameworks bezüglich Zeitaufwand und Kosten vergleichen zu können, wurde ein interner Workshop organisiert, an dem verschiedene Teams aus Mobile-Entwicklern mit begrenzten Vorkenntnissen zu Flutter eine App zu entwickeln versuchten. Die App sollte sowohl auf Mobilgeräten als auch im Internet nutzbar sein und ein möglichst einheitliches Erscheinungsbild aufweisen. Das erklärte Ziel

der App war es, mithilfe von Diagrammen den besten Zeitpunkt für die Nutzung von Strom aus dem Netz anzuzeigen. Anhand der Daten aus dem Stromnetz wurde ermittelt, wann das Stromnetz mit den meisten erneuerbaren Energien versorgt wurde. Dieser Zeitraum wurde als optimal dargestellt. Die Flutter-App wurde extrem schnell entwickelt, und es entstand innerhalb kurzer Zeit ein Prototyp für iOS, Android und das Web.

Flutter bei Zühlke

Bei Zühlke Engineering haben wir im Rahmen unterschiedlicher Projekte umfassende Erfahrungen mit Flutter gesammelt. Wir ermöglichen Kunden mit umfangreichen Anwendungen den Übergang von nativen Anwendungen zu Flutter. Anderen geben wir die Möglichkeit, ihre Anwendungen von Grund auf in Flutter zu entwickeln. Flutter eignet sich ideal für einen Übergang, da der

Flutter-Code integriert werden und den bestehenden Code in einer nativen Anwendung Schritt für Schritt ersetzen kann. Dies gewährleistet einen problemlosen und nahtlosen Übergang. Aber nicht nur unsere Kunden, auch unsere Entwickler sind von Flutter begeistert. Benutzerfreundlichkeit, Entwicklungsgeschwindigkeit und eine angemessene Lernkurve erleichtern den Übergang.





Low-Level-API der Plattform

Wir haben eine App entwickelt, die für mobile Anwendungen und Windows eingesetzt werden könnte. Die App könnte serielle Bytes von und auf NFC-Gerätespeichern lesen und schreiben. Zu diesem Zweck mussten wir die NFC-Funktionen der Geräte nutzen. Dies machte die Nutzung von Low-Level-API erforderlich. Für mobile Geräte konnten wir ein bereits vorhandenes Plugin verwenden, das die Anbindung und Nutzung der

Android- und iOS-API in Flutter ermöglicht. Für die Windows-App nutzen wir das Dart FFI, womit sich unsere Befehle an die zugrunde liegende Windows-API übermitteln lassen. Die App dient als Beispiel dafür, wie gerätespezifische Funktionen und Optionen mittels Flutter implementiert werden können und wie eine einzige Geschäftslogik für die gesamte App verwendet werden kann.

Schnelle App-Entwicklung für eine bestehende Webseite

Wir haben zudem eine App für einen Kunden in der Medizinbranche entwickelt, die ein gutes Beispiel für Flutter als schnelle plattformübergreifende Lösung ist. Die bereits vorhandene Webanwendung wurde innerhalb des vorgegebenen Zeitrahmens erfolgreich von Grund auf in iOS- und Android-Apps nachgebaut. Die existierenden Funktionen der Web-App, wie z. B. Video- und Audio-Player, wurden ganz einfach mit den bereits vorhandenen Flutter-Plugins in der

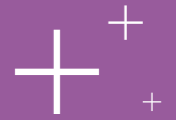
mobilen App gespiegelt, und zwar zusätzlich zu den Netzwerkaktivitäten mit dem aktuellen Backend. Die Änderungswünsche des Kunden wurden in beiden Plattformen umgesetzt und innerhalb kurzer Frist zu Testzwecken geliefert. Die finale App zeigte, dass eine rasche Entwicklung und Kostenreduktion möglich sind, indem Flutter anstelle von anderen Technologien verwendet wird.

Kontakt



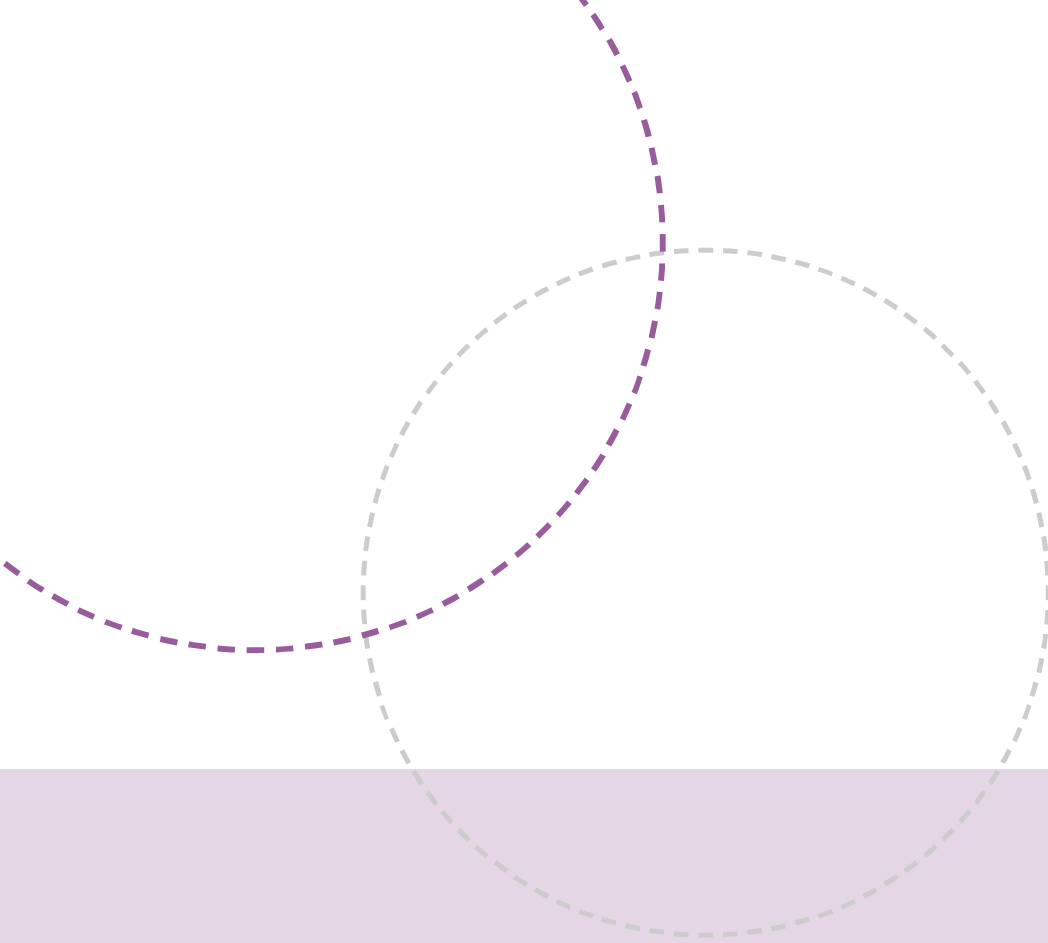
Doris Rogger, Head of Mobile

doris.rogger@zuehlke.com



Zühlke – Empowering Ideas.

Zühlke ist ein globaler Anbieter von Innovationsdienstleistungen. Wir erarbeiten Ideen und schaffen neue Geschäftsmodelle für unsere Kunden, indem wir Dienstleistungen und Produkte auf der Grundlage neuer Technologien entwickeln – von der Vision über die Konzeption bis hin zur Umsetzung, zur Produktion und zum Betrieb. Zühlke beschäftigt 1700 Mitarbeitende in Österreich, Bulgarien, Deutschland, Hongkong, Portugal, Serbien, Singapur, der Schweiz, dem Vereinigten Königreich und Vietnam.



Impressum

Zühlke Engineering AG
Zürcherstrasse 39J
8952 Schlieren (Zürich)
Schweiz

info@zuehlke.com
CEO:
Nicolas Durville

Bilder:
Getty Images Deutschland GmbH

© Zühlke 2023 Alle Rechte vorbehalten.

