

## Progressive Web-Apps (PWA): Was versprechen progressive Apps?

21.02.2022, 08:47 | IT, New Media & Software

Pressemitteilung von: *Inars Vitols*  
Presseagentur: *appFactory GmbH*

---

Die Progressive Web-App (PWA) ist die neueste Begebenheit in der digitalen Welt. Eine PWA verknüpft Eigenschaften von Websites mit vielen Merkmalen nativer mobiler Apps. Dadurch soll die Applikation befreit vom Betriebssystem gleichermaßen als Website und als App funktionieren.

Doch können progressive Web-Apps wirklich den etablierten nativen Apps Konkurrenz machen? Und was ist überhaupt das Progressive an diesen Apps bzw. was unterscheidet sie von klassischen Web-Apps? Darüber haben wir mit dem Tech-Experten Inars Vitols aus Hamburg gesprochen.

### Was ist eine Progressive Web-App?

Der Begriff „Progressive Web-App“ bezeichnet eine neuartige Methode des App-Developments, die vielversprechende Features mit sich bringt. Sie stellt ein Update des schon länger existierenden Formats der Web-App dar.

Für das neue App-Format ist vor allem Google verantwortlich. Daher überrascht es nicht, dass PWAs bisher vorrangig für die Nutzung mit dem hauseigenen Android-Betriebssystem entwickelt sind – auf iOS-Geräten des Wettbewerbers Apple laufen sie bisher nur mit Einschränkungen. Die Technologie befindet sich aber noch in einer frühen Phase und kann auch deshalb noch nicht ihr volles Potenzial entfalten. Trotzdem ist schon jetzt absehbar, dass Google in ihre Weiterentwicklung investieren wird. Die stetige Optimierung von PWAs ist deshalb sehr wahrscheinlich.

### Definition Progressive Web-App, Teil 1: Website und App in einem

Da sich PWAs noch mitten in der Entwicklung befinden, ist eine Begriffsklärung selbst für unseren Tech-Experten Inars Vitols momentan nur in Ansätzen möglich. Trotzdem kann das Fundament als gesetzt angesehen werden. Eine PWA ist über eine URL im Internet erreichbar und wird im Browser ausgeführt, wodurch sie grundsätzlich auf verschiedenen Betriebssystemen läuft. Außerdem ist sie dadurch unabhängig von einem App-Store verfügbar und benötigt daher keine Installation.

Im Unterschied zu einer herkömmlichen Web-App kann man eine Progressive Web-App aber auch offline ausführen. Dafür legt man auf dem Homescreen eines internetfähigen Device einen Shortcut mit der entsprechenden URL an. Damit lässt sich die App dann einfach aufrufen – selbst wenn keine Internetverbindung besteht (Voraussetzung dafür ist ein Offline-Cache im Browser). Die Applikation präsentiert sich dann jedoch oft nicht wie eine Website, sondern vermittelt vielmehr den Eindruck einer nativen App. Dank ihres responsiven Designs passt sie sich an jegliche Displaygröße an.

### Definition Progressive Web-App, Teil 2: PWA vs. Native App

Eine PWA kann man sich also wie eine responsive Webanwendung vorstellen, die den Anschein einer nativen App erweckt. So kann die Applikation native Funktionen des verwendeten Device wie Kamera und Mikrofon, GPS-Ortung oder Push-Notifications nutzen und in das Programm einbeziehen. Dabei prüft die Anwendung sowohl den genutzten Browser als auch das Device auf Kompatibilität. Auch das „Look and Feel“ einer PWA verhält sich auf einem Smartphone und Tablet ähnlich wie bei einer Native Mobile App (flüssige Verhaltensweise bei der Bedienung, gute Reaktionszeiten bei Wischbewegungen etc.).

Eine progressive Web-App online zu öffnen, hat aber den Vorteil, dass man stets auf die neueste Version zurückgreifen kann. Anders als bei einer Native App müssen sich die Nutzer also nicht um das Update kümmern – sobald eine

Anbindung an den Server der PWA besteht, wird die Applikation auf mögliche Updates überprüft.

Weitere Vorteile gegenüber einer nativen App sind der geringere Arbeitsaufwand für die Entwicklung einer PWA (denn sie ist zugleich eine Website und eine auf verschiedenen Betriebssystemen laufende App) und damit einhergehenden geringeren Kosten – und das bei einer für viele Szenarien ähnlich guten Reaktionszeit bei der Wiedergabe und Bedienung. Ausserdem benötigt eine PWA kaum Speicherplatz auf dem Device des Users (was allerdings auch von der Größe des Offline-Caches abhängt).

Einige der genannten Funktionen bieten jedoch auch bereits klassische Web-Apps sowie mit Einschränkungen auch responsive und mobile Websites. Deswegen haben wir den App-Spezialisten Inars Vitols gefragt: Was ist neu an Progressive Web-Apps gegenüber klassischen Web-Apps?

### **Definition Progressive Web-App, Teil 3: PWA vs. traditionelle Web-App**

Bereits klassische Web-Apps können browserbasiert (also ohne Installation) auf verschiedenen mobilen Betriebssystemen wie Android, iOS und Windows Phone funktionieren. Der Ansatz einer progressiven Web-App ist also keineswegs neu, präsentiert sich aber grundlegend ausgereifter als bei traditionellen Web-Apps: Der Funktionsumfang einer PWA passt sich an den Nutzungsrahmen an, was bedeutet, dass auch Nutzer, deren Devices und/oder Browser nicht mit allen Funktionen der Anwendung kompatibel sind, diese dennoch nutzen können (wenn auch nur in abgespeckter Form).

Ein Beispiel: Eine Progressive Web-App, mit der sich Fotos auf eine Onlineplattform hochladen lassen, erlaubt die Nutzung einer Kamera. Hierdurch sollen die User direkt in der Anwendung Fotos erstellen und hochladen können. Allerdings unterstützen nicht alle Browser und Devices dieses Feature auf technischer Ebene. Sobald man die Anwendung in einem Browser startet, prüft sie deshalb, ob eine Kamera angeschlossen ist und diese von der PWA und vom Browser unterstützt wird. Sollte das nicht der Fall sein, lässt sich zwar die Kamera-Funktion nicht nutzen, aber immerhin alle anderen kompatiblen Features der Webanwendung. Wenn die Kompatibilität mit der Kamera nicht gegeben ist, könnte man also beispielsweise immer noch durch eine Upload-Funktion der Progressive Web-App eigene Smartphone-Aufnahmen in diese integrieren.

Progressiv sind die Apps also in dem Sinne, dass sie in jedem Browser lauffähig sind, ihr Funktionsumfang sich jedoch an die eingesetzte Hardware und die Browserfunktionen anpasst. Die Funktionen einer PWA wachsen also mit den Möglichkeiten des genutzten Browsers und Endgeräts.

### **Definition Progressive Web-App, Teil 4: PWA vs. Hybrid-App**

Neben der Web-App gibt es mit der schon länger existierenden Hybrid-App ein weiteres Format, das einige Funktionen nativer Applikationen nachahmt und einige ihrer Defizite bewältigt. So kann eine hybride App – ähnlich wie eine PWA – mit einem vergleichsweise geringen Arbeitsaufwand für verschiedene mobile Betriebssysteme entwickelt werden. Allerdings verfolgen Hybrid-Apps dabei einen anderen Ansatz: Sie werden per Installation in Android oder iOS integriert und kopieren so die Funktionsweise von nativen Apps – was Vor- und Nachteile hat.

Neben der weniger aufwendigen Entwicklung bildet die Integration in das Betriebssystem einen wesentlichen Pluspunkt hybrider Apps. Sie profitieren dabei davon, dass sich durch die Installation mehr native Gerätefunktionen einbeziehen lassen als bei (progressiven) Web-Apps – allerdings immer noch weniger als bei nativen Apps.

Doch die notwendige Installation birgt auch Nachteile. Hybrid-Apps lassen sich nicht so einfach testen, wie es das Web-App-Format erlaubt. Außerdem bleibt die Applikation plattformabhängig – und selbst wenn hier das Development für ein einzelnes Betriebssystem schneller von der Hand geht als bei Native Apps, so bedeutet dies im Endeffekt dennoch deutlich mehr Programmieraufwand als bei einer Progressive Web-App. Denn bei der Entwicklung einer PWA müssen keine Spezifikationen für Android und iOS vorgenommen werden – hier rückt stattdessen die Kompatibilität der Hardware und des Browsers in den Mittelpunkt.

### **Technische Hintergründe einer PWA**

Eine PWA basiert ausschließlich auf offenen Webstandards und wird – wie eine konventionelle Web-App – prinzipiell in den Sprachen HTML, CSS und JavaScript geschrieben. Progressive Web-Apps sind äußerst agil und passen ihre Features stets den jeweiligen Rahmenbedingungen an. Wozu genau eine PWA möglich ist, hängt dabei immer vom eingesetzten Browser und Device (und zum Teil auch vom Betriebssystem) ab. Doch welche Technologien kommen hier zum Einsatz?

### **Service Worker**

Eine Errungenschaft der PWAs ist die Usage der Service Worker, einer Weiterentwicklung der Web Worker. Ein Service Worker wird als JavaScript im Hintergrund des Browsers ausgeführt (in Form eines von der Website losgelösten Threads). Beim ersten Visit der Anwendung lädt der PWA-Server den Service Worker und versucht diesen zu installieren. Nach der erfolgreichen Installation ist der Service Worker bei jedem Neustart der App einsatzbereit und wird über jegliche Netzwerkabfrage auf der betreffenden Domain informiert. Dies funktioniert aber nur unter Nutzung von HTTPS, andernfalls bestünden erhebliche Sicherheitsrisiken.

Der Service Worker und der dazugehörige Cache werden im verwendeten Browser gespeichert – sofern dieser die Funktion unterstützt (was derzeit bei Opera, Google Chrome und Mozilla Firefox der Fall ist). Das Besondere hierbei: Ein Service Worker ermöglicht die Nutzung einer PWA auch ohne Internetverbindung, da die Inhalte aus dem Cache geladen werden. Dieser Vorgang lässt sich massiv beschleunigen, indem man das App-Fundament cacht und somit nur aktuelle Daten frisch heruntergeladen werden müssen.

### **Trennung von Application Shell und Inhalt**

Beim Development von Progressive Web-Apps spielt neben den Service Workern die „Application Shell Architecture“, die auch bei Native Apps eingesetzt wird, eine wichtige Rolle. Durch diese unterscheidet die Anwendungssoftware zwischen der Application Shell (dem festgelegten Aufbau der Anwendung, kurz: App Shell) und dem Content. Die PWA-Benutzeroberfläche beruht dabei auf der App Shell, die zuerst geladen und dargestellt wird. Der hier abgebildete Content verhält sich dynamisch und wird aus dem Internet geladen.

Die App Shell wird nach dem ersten View der PWA im Cache des Service Workers gespeichert. Das Ablegen im Service-Worker-Cache hat den Vorteil, dass der Ladevorgang der App Shell weniger Zeit in Anspruch nimmt, wodurch sich letztlich die Performance der PWA erhöht. Zusätzlich lässt sich der Content der PWA im Cache bzw. via IndexedDB sichern. Dabei müssen sie (genau wie die App Shell) bei einem vorherigen Zugriff auf die Progressive Web-App heruntergeladen werden. Es ist somit möglich, auch den bereits heruntergeladenen Content einer PWA offline abrufbar zu machen.

### **WebAPKs**

Bei WebAPKs handelt es sich um Server, über die man eine im Browser aufgerufene progressive Web-App in das Dateiformat APK (Android Package) umwandeln kann. Damit lässt sich die PWA noch besser in das Betriebssystem integrieren – zumindest auf Android-Geräten.

Eine im APK-Format verpackte PWA lässt sich so in den App Drawer integrieren und dadurch ähnlich wie eine native Android-App auf dem Gerät installieren. Dadurch kann die PWA noch mehr native Features, Zugriffsrechte auf andere Anwendungen sowie Ressourcen des Device verwenden. Beispielsweise führt Android eine PWA nach der Installation als eigene Anwendung aus und nicht mehr als Tab in einem Browser.

Bisher unterstützen aber nur wenige Beta-Versionen von Android-Browsern die WebAPK-Funktion und die Konvertierung in das APK-Format ist noch sehr schwierig. In der momentanen Variante ist die Installation von WebAPKs also noch nicht massentauglich. Eine zukünftige Verbreitung dieser Funktion auf alle gängigen Browsern ist aber realistisch, findet Inars Vitols. Doch schon jetzt – auch ohne die Installationsfunktion – überzeugen PWAs durch ihr großes Leistungsvermögen.

### **Status quo und Zukunft von PWAs**

Generell waren der Fortschritt in der Browser-Technologie im Allgemeinen und die hiermit verbundene Implementierung der Service Worker im Speziellen ausschlaggebend dafür, dass die neuartigen Web-Apps überhaupt performen konnten. Schon jetzt lassen sich mit den oben beschriebenen PWA-Technologien sehr unterschiedliche und eindrucksvolle Apps umsetzen – die aber nur mit Browsern, die Service Worker unterstützen, ihr volles Potenzial entfalten können. Progressive Web-Apps sind momentan nicht in der Lage, Native Apps in allen Bereichen zu ersetzen – und so wird es auch in der Zukunft sein. Denn nativ für ein Betriebssystem entwickelte Applikation haben immer die meisten Möglichkeiten, um auf Endgeräte, Systemressourcen etc. zugreifen zu können. Allerdings benötigen die allermeisten Apps gar keine derart tiefe Einbeziehung der Soft- und Hardware. Insofern ist das PWA-Format für die meisten Anwendungen durchaus interessant. Professionelle Entwickler (wie z.B. Inars Vitols aus Hamburg) bereiten sich daher schon jetzt auf diesen neuen Trend vor.

Viele Vorteile progressiver Web-Apps sind offensichtlich, weshalb sowohl App-Anbieter als auch User große Hoffnungen in das PWA-Format setzen. Unter dem Strich stehen der massenhaften Verbreitung und Herstellung von PWAs nur wenige, dafür aber gravierende Faktoren im Weg.

### **Können sich PWAs gegenüber nativen Apps durchsetzen?**

Neue App-Formate hatten es in der Vergangenheit schwer: Sowohl die herkömmlichen Web-Apps als auch Hybrid-Apps haben es nicht geschafft sich gegenüber Native-Apps durchzusetzen. Progressive Web-Apps hingegen könnten dies dank ihrer ausgeklügelten technischen Grundlage und der großen Unterstützung von vielen Seiten erstmals schaffen, findet Inars Vitols. Die wichtigste Frage bleibt im Moment, ob sich Apple dem PWA-Format mehr öffnen wird. Dies würde progressive Web-Apps wahrscheinlich in kürzester Zeit zu einem ernst zu nehmenden Feind für native Apps machen.

Apple hat aber gute Gründe, sich weiterhin auf das native App-Format zu konzentrieren und PWAs in seinen Betriebssystemen Steine in den Weg zu legen. Immerhin hat Apple mit seinem App-Store eine große Einnahmequelle und bindet darüber hinaus viele Customer an seine Services. Auch die über die nativen Apps und den App-Store erhobenen Daten spielen hier eine wichtige Rolle. Wenn Apple PWAs in einem solchen Maß unterstützen würde, dass sich viele Anwendungen auf iOS und macOS auch als Progressive Web-App anstatt als native App umsetzen ließen, wären die Apple-Developer und -User nicht mehr vom App-Store abhängig. Diese Kopplung an den eigenen Store würde Apple nur dann aufgeben, wenn PWAs in der Usage und Verbreitung Native Apps übertreffen sollten.

Allerdings scheint der Markt für Progressive Web-Apps auch ohne eine volle Unterstützung von Apple rentabel genug zu sein. Für den Erfolg des Formats müssten gängige Browser es dennoch besser unterstützen. Es ist zu erwarten, dass bei der Weiterentwicklung des Chrome-Browsers eine bessere Vereinbarkeit mit PWAs sowie eine einfachere PWA-Integration für Android in Planung sind. Neben Chrome sind aber u. a. auch Firefox und Opera Fans des PWA-Formats – die teils schon vorhandene Kompatibilität ihrer Browser mit PWAs wird demnach noch stärker in den Fokus rücken. Die Befürworter von progressiven Web-Apps sind zumeist Akteure aus der Internet- und Software-Industrie, die von einer offenen Computer- und Informationstechnologie profitieren.

### **Fazit: Sind Progressive Web-Apps die Apps der Zukunft?**

Die Bezeichnung „Progressive Web-App“ ist nicht nur ein Marketing-Begriff – mit PWAs betreten die App-Entwickler technisches Neuland. Die Weiterentwicklung der „normalen“ Web-Apps trägt eine durchaus revolutionäre Chance: Progressiv sind die Apps deshalb, weil sie sich an die Fähigkeiten der Anwendungsumgebung angleichen – je mehr Leistung das Device und der verwendete Browser mit sich bringen, umso mehr Features kann eine PWA abrufen.

Öffnen Sie eine PWA auf einem Desktop-PC oder Notebook, verhält sie sich wie eine gewöhnliche Webanwendung. Rufen Sie sie aber mit einem Smartphone oder Tablet auf, erhalten Sie eine Webanwendung, die wie eine Native App auftritt. Abhängig von der Ausstattung Ihres mobile Device ist es dann möglich, dass die Applikation native Gerätefunktionen nutzt (wie etwa Kamera, Mikrofon, Push-Notifications oder GPS).

Die Vorteile progressiver Web-Apps sind offensichtlich und vielversprechend: Sie benötigen weder Download noch Update, werden bei jedem Visit automatisch aktualisiert, sparen Gerätere Ressourcen und brauchen nicht zwingend eine Internetverbindung. Die Plattformunabhängigkeit ermöglicht es Ihnen außerdem, eine PWA erst einmal online zu testen,

bevor Sie sie tiefer in ihr Gerät eingliedern. Darüber hinaus bildet die Reaktionszeit der Applikation einen wichtigen Faktor: Bislang boten native Apps meist das bessere „Look and Feel“, doch auch in diesem Bereich sind PWAs inzwischen in der Lage, Ähnliches zu leisten.

Der geringere Aufwand und Preis in der Herstellung (man braucht nur eine Anwendung zu programmieren, die sowohl als Website wie auch als App für alle Plattformen fungiert) bilden auf Entwicklerseite einen großen Pluspunkt. Die Beschaffenheit einer PWA spielt insofern App-Anbietern wie auch Usern in die Karten. Trotzdem scheitert es noch an der Unterstützung aller nativen Funktionen mobiler Endgeräte: Auf funktioneller Ebene treten sie erst dann in einen ernsthaften Wettbewerb mit Native Apps, wenn Progressive Web-Apps die Möglichkeiten eines Geräts in annähernd gleicher Weise nutzen können (auch wenn sie wohl nie mit den Nutzungsmöglichkeiten nativer Apps mithalten werden).

Wesentlich problematischer ist die fehlende Bereitschaft von Apple, das App-Format zu unterstützen. Die Browser der meisten Anbieter (Chrome, Firefox, Microsoft, Opera) werden viel daran setzen, dass eine bestmögliche Kompatibilität des offenen PWA-Formats schnell Einzug in ihre Produkte findet. Apple hingegen hat grundsätzlich weit weniger Interesse daran, da das Unternehmen erheblich vom isolierten App-System der OS-Plattformen und den daran angeschlossenen App-Store profitiert.

Die Zukunft wird zeigen, ob Progressive Web-Apps wirklich mit Native Apps mithalten können – die Chancen hierfür stehen dank der namhaften Unterstützer denkbar gut. Ob es jemals eine volle Kompatibilität des PWA-Formats in den Betriebssystemen und dem Browser von Apple geben wird, bleibt ungewiss. Doch sollte das PWA-Format in Zukunft mehr Aufmerksamkeit und Zuspruch von Entwicklern und Nutzern erhalten, dürfte auch Apples Zurückhaltung gegenüber dem progressiven App-Format mehr und mehr schwinden. Denn in der Regel kann es sich kein Global Player leisten, sich einer richtungsweisenden und erfolgreichen Technologie komplett zu verschließen.

Sie möchten mehr über Progressive Web-Apps erfahren? Dann fordern Sie gerne ein kostenloses Beratungsgespräch bei Inars Vitols und seinen Kollegen von der appFactory GmbH in Hamburg an. Geschäftsführer Inars Vitols ist seit über 20 Jahren in der digitalen Branche aktiv, stets auf dem neuesten Stand und berätet Sie gerne bei Ihrem Vorhaben.

## **Portrait**

Die appFactory ist eine Tech Agentur der mobilen Welt. Wir begleiten unsere Kunden auf dem Weg zum Erfolg. Unsere Kernbereiche sind dabei iOS, Android, tvOS, Android TV, AR, VR, Bots und Skills und unser Leistungsportfolio erstreckt sich von Strategieberatung, Design Thinking, Agile Coaching, UX, UI-Design, über die Software-Entwicklung bis hin zum Aufbau und Betrieb Cloud basierter Backend-Services.

---

News-ID: 1224797 • Views: 869 (Stand: 18.04.2026)

Link zur Pressemitteilung:

<https://www.openpr.de/news/1224797/Progressive-Web-Apps-PWA-Was-versprechen-progressive-Apps.html>