

Marcus Stade

# APP-TRACKING MIT GOOGLE ANALYTICS 4 UND FIREBASE

Apps bauen oft auf unterschiedlichen Technologien auf. Zum Beispiel zeigen sogenannte hybride Apps Teile einer Website innerhalb der App an. Reine Web-Apps basieren hingegen ausschließlich auf Webtechnologie und native Apps sind speziell für das Betriebssystem des Endgerätes programmiert. Der Beitrag von Marcus Stade beschreibt am Beispiel der Tools Google Firebase und Google Analytics 4, welche App-Tracking-Set-ups sich für die jeweilige technische Ausgangslage eignen.

218 Milliarden Apps wurden 2020 weltweit heruntergeladen, was zu einem Umsatzvolumen von 148 Mrd. USD allein für die App-Stores von Apple und Google führte (<http://einfach.st/apps3>). Der mobile App-Markt wuchs auch 2021 kräftig weiter (<http://einfach.st/apps4>). Inzwischen verfügen über 62,6 Mio. Nutzer in Deutschland über ein Smartphone (Quelle: VuMA Touchpoints 2022) und die Deutschen sind in Europa die aktivsten App-Nutzer (<http://einfach.st/absatz7>). Ein riesiges Marktpotenzial für deutsche Unternehmen.

Doch Apps deutscher Anbieter findet man kaum auf den Listen der am meisten heruntergeladenen Apps hierzulande, wenn man von der Corona-App einmal absieht. Viele Unternehmen haben zwar inzwischen eine eigene mobile App oder planen die Einführung, allerdings sehen Vorstände und Finanzmanager das Investment oft kritisch. Sie scheuen hohe Einführungs-, Infrastruktur- und Instandhaltungskosten zusätzlich zur schon bestehenden Website und verlangen Belege für Wachstumschancen und Rentabilität, was ein umfassendes Tracking und Reporting der Apps erfordert. Über Werbung und neue Funktionalitäten müssen neue Nutzer gewonnen werden. Gleichzeitig gilt es, bestehende Nutzer durch

eine Verbesserung der Nutzerfreundlichkeit und des Angebots an Interaktionsmöglichkeiten aktiv zu halten und nicht zu verlieren. Die Optimierung von Werbemaßnahmen und Bedienbarkeit der Apps setzt eine korrekte technische Erfassung der Nutzerinteraktionen voraus. Diese weicht in einigen wichtigen Punkten vom klassischen Website-Tracking ab. Was ist also bei der Auswahl der Lösungen und beim Set-up zu beachten?

## Mobile-App-Tracking: die Wahl des richtigen Set-ups

Vor der Wahl des richtigen Set-ups für das App-Tracking sind vorerst die Anforderungen an die Datenerhebung zu definieren. Im Klartext – welche Daten sollen innerhalb der App generiert und erfasst werden? Welche Verwendung finden sie und an welche Tools und Systeme sollen sie weitergereicht werden? Ist dies klar kommuniziert, hängt die Wahl des Set-ups für die Implementierung eines App-Trackings, wie bereits erwähnt, grundlegend von der technischen Architektur der App und ihres verwendeten Betriebssystems ab. Grundsätzlich kann bei mobilen Endgeräten zwischen reinen Web-Apps, nativen Apps und einem hybriden Ansatz unterschieden werden.

### DER AUTOR



**Marcus Stade** ist Head of Analytics bei mohrstade. Darüber hinaus ist er Co-Organisator der Community Analytics Pioneers. Auf seinem Blog [www.marcusstade.de](http://www.marcusstade.de) schreibt er regelmäßig Fachartikel zu Themen der digitalen Analyse.

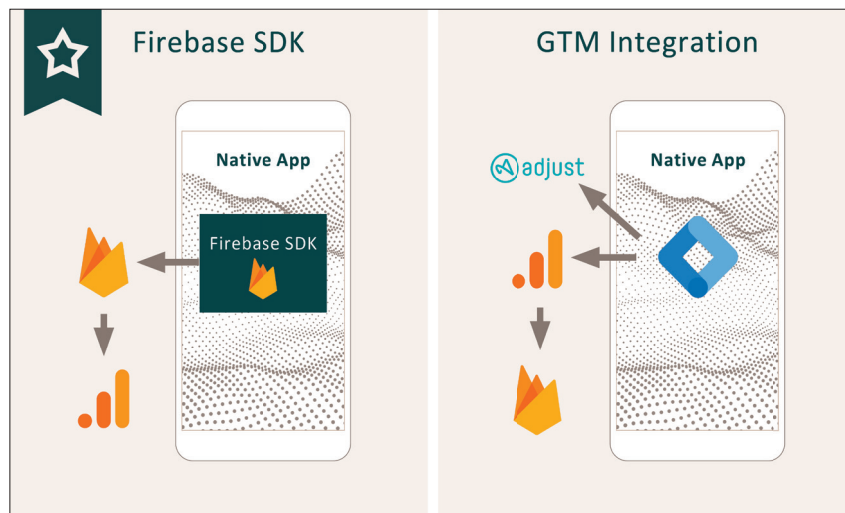


Abb. 1: Optionen für das Tracking nativer Apps

### Tracking nativer Apps

Eine native App wird speziell für das Betriebssystem (OS) eines mobilen Endgerätes entwickelt und über dessen dedizierten Marketplace vertrieben. Als Grundlage für die Datenerfassung innerhalb einer nativen App wird idealtypisch auf ein sog. Software Development Kit (SDK) zurückgegriffen. Bekannte Anbieter von App-Tracking-SDKs sind unter anderem Adobe mit dem Adobe Experience Platform Mobile SDK oder das von Google entwickelte Firebase SDK. In Abbildung 1 haben wir zwei Optionen für das Tracking nativer Apps anhand der Nutzung des Firebase SDK sowie des Google-Tag-Managers (GTM) dargestellt:

- » (1) Um Events innerhalb der App zu erfassen, wird das Firebase SDK in dem Quellcode der App eingebunden. Wird nun ein Event ausgelöst, sendet das Firebase SDK dieses an das zugehörige Firebase-Projekt. Innerhalb von Firebase kann eine Verknüpfung zu einer Google-Analytics-4-Property vorgenommen werden, wodurch sich die generierten In-App-Events anschließend zusammen mit Daten aus anderen App- und Web-Streams zusammenführen und analysieren lassen.
- » (2) Die Implementierung eines Tag-Management-Systems (TMS)

wie Google-Tag-Manager (GTM), Adobe Launch oder Tealium iQ bietet eine flexiblere Integration und Kommunikation der In-App-Events an ausgewählte Daten-Endpunkte von Werbeplattformen und/oder sonstigen Drittanbietern. In einem gewissen Rahmen ermöglicht ein TMS, nachträgliche Konfigurationen an erfassten Events vorzunehmen. Kleinere Anpassungen lassen sich somit unabhängig von Releasezyklen der IT durchführen. In unserem Beispiel übernimmt der implementierte GTM-Container die Aufgabe, die erfassten In-App-Events in das spezifische Datenformat des jeweiligen Daten-Endpunktes zu übersetzen

### INFO

Ein Software Development Kit (SDK) ist eine Sammlung von hilfreichen Tools, Schnittstellen, Dokumentationen technischer Abläufe und Programm-bibliotheken, die zur Entwicklung der App und ihrer verwendeten Software dienen.

und an diesen Daten-Endpunkt zu übermitteln. Bei der Verwendung von Google Analytics 4 stehen die erhobenen Daten anschließend ebenfalls in Firebase zur Verfügung.

### Tracking von Web-Apps

Die Basis von Web-Apps bilden Webtechnologien wie HTML, CSS und JavaScript. Die Inhalte werden dabei über einen sog. Web-Container (Webview) dargestellt, welcher sich über einen Browser von jedem internetfähigen Endgerät aufrufen lässt. Die Bereitstellung, Verarbeitung und Auswertung der Daten findet nach dem sogenannten Client-Server-Modell statt und erfolgt somit nicht auf dem Endgerät selbst. Die Parallelen zu einem klassischen Web-Tracking sind dabei sehr stark. Die folgenden zwei Optionen (Abbildung 2) stellen unsere Favoriten für das Tracking von Web-Apps dar:

- » (1) Auch bei dieser technischen Ausgangslage wird die Implementierung eines Tag-Management-Systems

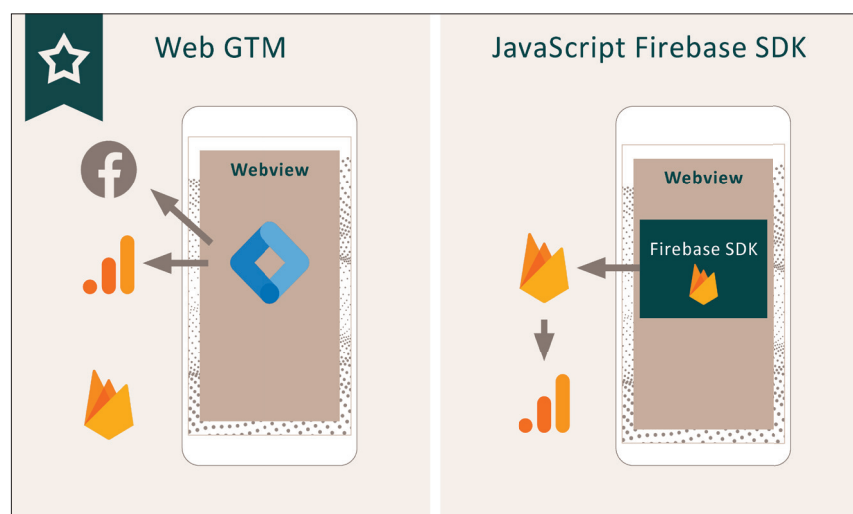


Abb. 2: Optionen für das Tracking von Web-Apps

(TMS) wie des Google-Tag-Managers (GTM) aufgrund der gegebenen Flexibilität favorisiert. Im HTML-Code des Webviews implementiert, reagiert der GTM-Container auf Veränderungen im Data-Layer-Objekt. Wird der Container durch ein Event getriggert, leitet er die Event-Daten an die jeweiligen Daten-Endpunkte (z. B. Facebook, Google Ads etc.) weiter. Für die Analyse der In-App-Events werden diese an eine Google-Analytics-4-Property im Firebase-Format weitergeleitet.

- » (2) Ist es aufgrund technischer Gegebenheiten innerhalb der App nicht möglich, ein Tag-Management-System (TMS) im Code zu integrieren, wird JavaScript als Programmiersprache jedoch grundlegend in der Anwendung unterstützt, lässt sich optional auf das Firebase JavaScript SDK zurückgreifen. Bei dieser Lösung werden die Daten ebenfalls mithilfe einer JavaScript-Bibliothek an Firebase weitergeleitet. Nach der Erfassung der Events in Firebase lassen sich diese bei Verknüpfung mit einer Google-Analytics-4-Property ebenfalls analysieren.

### Tracking hybrider Apps

Das technische Grundgerüst einer hybriden App baut auf den Bestandteilen einer nativen App auf. Hybride Apps können auf native Programmierschnittstellen (APIs) zugreifen, damit einzelne Funktionen des Betriebssystems nutzen, basieren in ihrer Kernfunktionalität jedoch auf Web-Technologien (HTML, CSS und JavaScript), welche in einem Web-Container (Webview) ausgeführt werden. Einfach gesagt stellt die App also eine Kombination aus nativer App und Web-App dar. Die Optionen in Abbildung 3 lassen sich für das Tracking hybrider Apps festhalten:

- » (1) In diesem Beispiel wird für die Erfassung der Events aus dem nati-

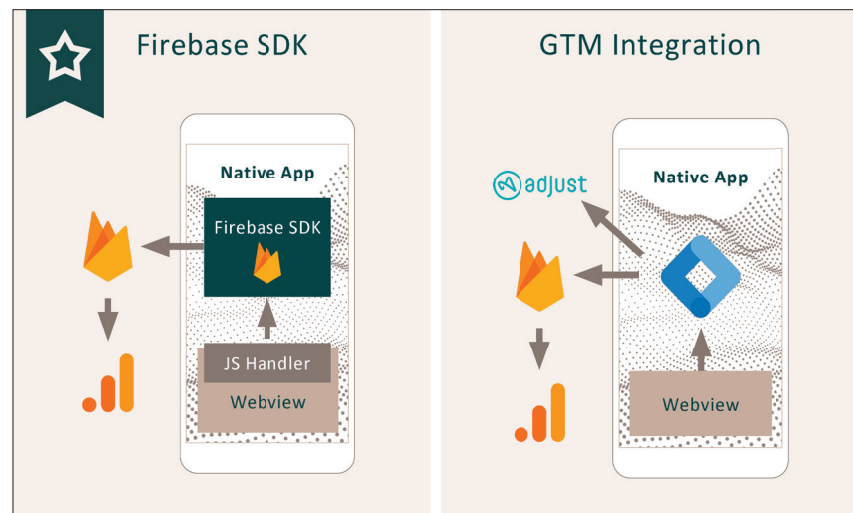


Abb. 3: Optionen für das Tracking hybrider Apps

ven Teil der App auf das Firebase SDK zurückgegriffen. Um jedoch auch die Events aus dem Web-Container (Webview) zu erfassen und dem gleichen Nutzer zuordnen zu können, ist es nötig, diese in ein einheitliches Datenformat zu strukturieren. Hierfür wird eine sog. App Bridge (JavaScript Handler) verwendet, da der native Teil der App technisch nicht mit dem Web-Container (Webview) kommuniziert. Der JavaScript Handler reagiert auf Veränderungen im Data-Layer-Objekt des Web-Containers (Webviews) und sendet diese in einem einheitlichen Datenformat via postMessage an das im nativen Teil der App eingebundene Firebase SDK. Anschließend können die zu Firebase übermittelten Daten in Google Analytics 4 analysiert werden.

- » (2) Wie bereits als Lösungsansatz für eine rein native App dargestellt, besteht auch bei einem hybriden Ansatz die Möglichkeit, ein Tag-Management-System wie den Google-Tag-Manager in den nativen Teil der App zu integrieren. Auch hier bietet diese Option Raum für Flexibilität in der Kommunikation von In-App-Events und Kampagnen-Informationen an die Daten-Endpunkte von Werbeplattformen und weiteren Drittanbietern. Der Verzicht auf zusätzlichen Code innerhalb der App nimmt dabei positiven Einfluss auf

die Komplexität technischer Abläufe. Vergleicht man dieses Vorgehen mit dem vorgestellten Ansatz der ersten Option, übernimmt der Google-Tag-Manager (GTM) hierbei gleiche Funktion wie der JavaScript Handler und bringt die im Web-Container (Webview) ausgelösten Events in das passende Firebase-Format. In Firebase erfasst, lassen sich die In-App-Events nach erfolgreichem Linking in Google Analytics 4 analysieren.

### Sonderlösung: Measurement Protocol

Das aus Google Analytics bekannte Measurement Protocol stellt in gewisser Art und Weise eine Sonderlösung für das Tracking von Events aus jeder HTTP-fähigen Umgebung dar. Diese Option beschränkt sich daher nicht nur auf das Tracking mobiler Apps. Über das Measurement Protocol lassen sich mit der HTTP-POST-Request-Methode einzelne Events direkt an die Server von Google Analytics 4 senden. Besteht aufgrund technischer Gegebenheiten also nicht die Möglichkeit, eine der bisher vorgestellten Optionen für das App-Tracking zu nutzen, können die Events manuell über das Measurement Protocol aus der App heraus verschickt werden. Des Weiteren lassen sich über das Measurement Protocol auch In-App-Events messen, falls die App nicht mit dem Internet in Verbindung steht. Generell sollte diese



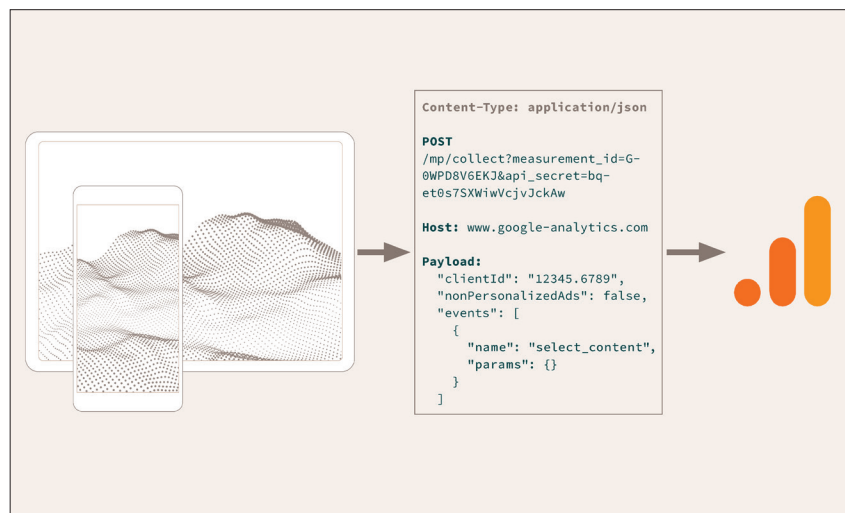


Abb. 4: Measurement Protocol

Option jedoch als Notlösung betrachtet werden, da sie mit einem erheblichen manuellen Aufwand in der Implementierung aufseiten der App-Entwickler verbunden ist und eine gewisse Fehleranfälligkeit in der validen Übermittlung der einzelnen Requests liegt.

### Empfehlung

Die Entscheidung für oder gegen eine der oben genannten App-Tracking-Optionen ist abhängig von den technischen Gegebenheiten der App. Grundsätzlich lassen sich einzelne Optionen für die jeweilige technische Ausgangs-

lage empfehlen.

Für das In-App-Tracking nativer Apps ist die Verwendung des Firebase SDK einer Integration des Google-Tag-Managers innerhalb der App vorzuziehen. Für reine Webviews empfehle ich die Integration des GTM. Wie auch bei reinen Webseiten ermöglicht er mehr Flexibilität und Unabhängigkeit von der IT in der Datenerfassung.

Für das Tracking einer hybriden App ist die Verwendung des Firebase SDK in Kombination mit einem Event-Handler, welcher die Daten aus dem Data-Layer des Webviews in das notwendige Firebase-Format umstrukturiert, die sicherste Variante für die Datenerfassung. ¶



# BLINDVK – KOMMT, SCHENKT EUCH WEIN.

BLINDVERKOSTUNG.DE



## SPANNENDES TEAMEVENT GESUCHT?

**Incentive für die Mitarbeiter? Online oder in Präsenz?**

Wir machen euch mit einer Blindverkostung zum Hobbysommelier und bringen jede Menge Spaß mit! Versprochen!!  
Auf Wunsch kommen wir mit Sack und Pack zu Euch oder wir organisieren für Euch ein Event der Extraklasse in einer Location Eurer Wahl ...  
Meldet Euch einfach – wir kümmern uns um einen unvergesslichen Abend.

☎ 0931 / 404 767 67 // ✉ [events@blindverkostung.de](mailto:events@blindverkostung.de)  
Weitere Infos und fertige Pakete unter [www.blindverkostung.de](http://www.blindverkostung.de)