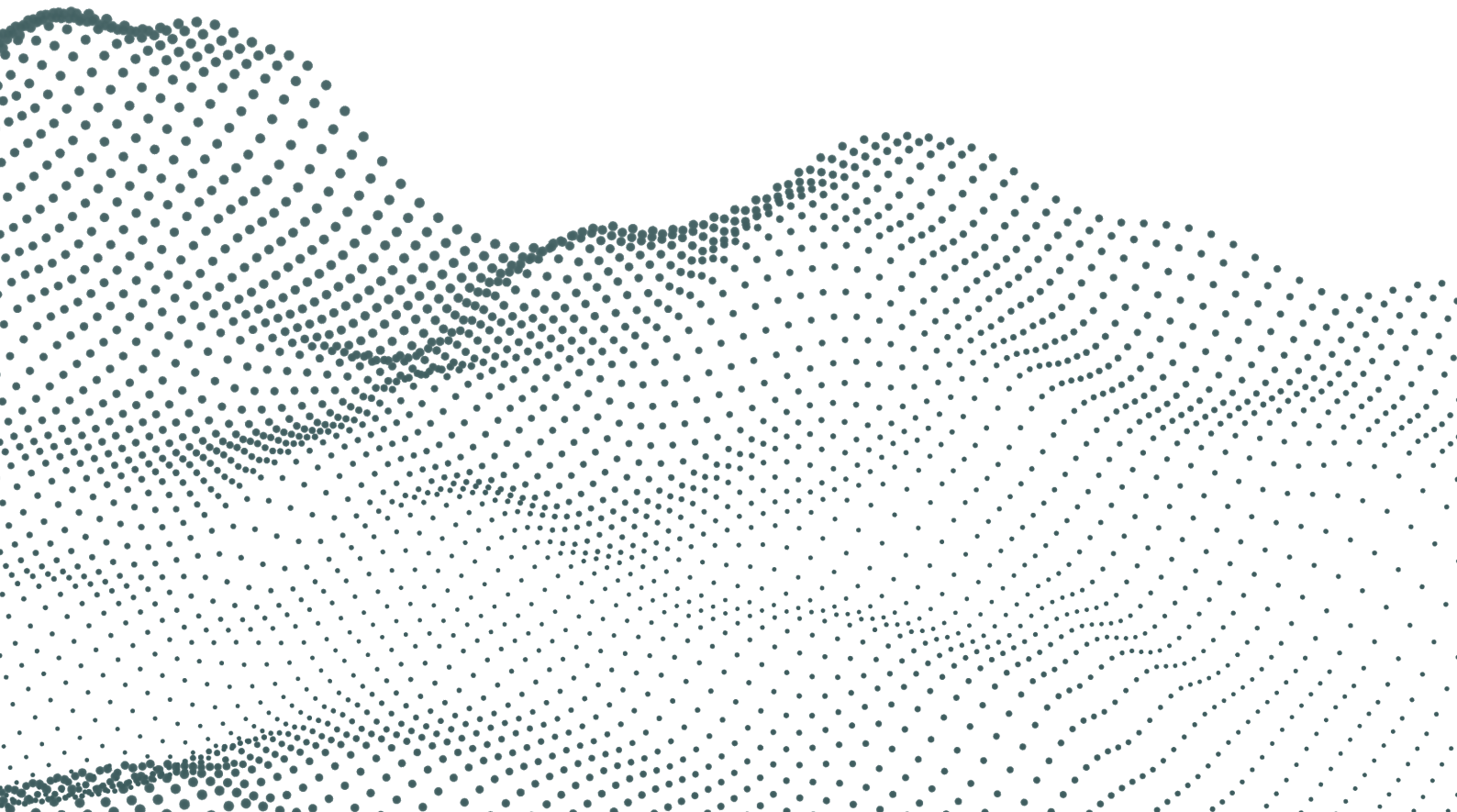
mohr stade

Marketing Attribution für Mobile Apps

Möglichkeiten und Herausforderungen

Whitepaper



Inhalt

Veröffentlichung dieses Whitepapers: November 2021

1. Marketing Attribution im Wandel

Die digitale Analyse und das Marketing verzeichneten in den letzten Jahren eine Zunahme datenschutzrechtlicher Reglementierungen für die Datenverarbeitung im Third Party-Kontext. Der Austausch von Nutzerdaten über mehrere Daten-Endpunkte hinweg wird laufend erschwert - angeführt von großen Browser-Anbietern sowie Soft- und Hardware-Riesen wie Google und Apple. Den größten Einfluss auf mobile Endgeräte nimmt dabei das von Apple entwickelte App Tracking Transparency (ATT)-Framework, das erfordert, dass iOS-Nutzer vorab in den Einstellungen des Betriebssystems selbst für jegliche Art von Tracking zustimmen. Eine granulare Identifikation von Gerät und Nutzer für Drittanbieter wird mit der Ablehnung des Trackings somit verhindert. Die Geräte-Id wird nicht mehr übertragen. Dies stellt natürlich gerade für werbefinanzierte Modelle ein großes Problem dar, da Marketingaktivitäten nicht mehr valide beurteilt werden können. In diesem Whitepaper widmen wir uns daher dem aktuellen Stand der Möglichkeiten und Herausforderungen, die für die Marketing Attribution mobiler Apps bestehen und die es zu beachten gilt.

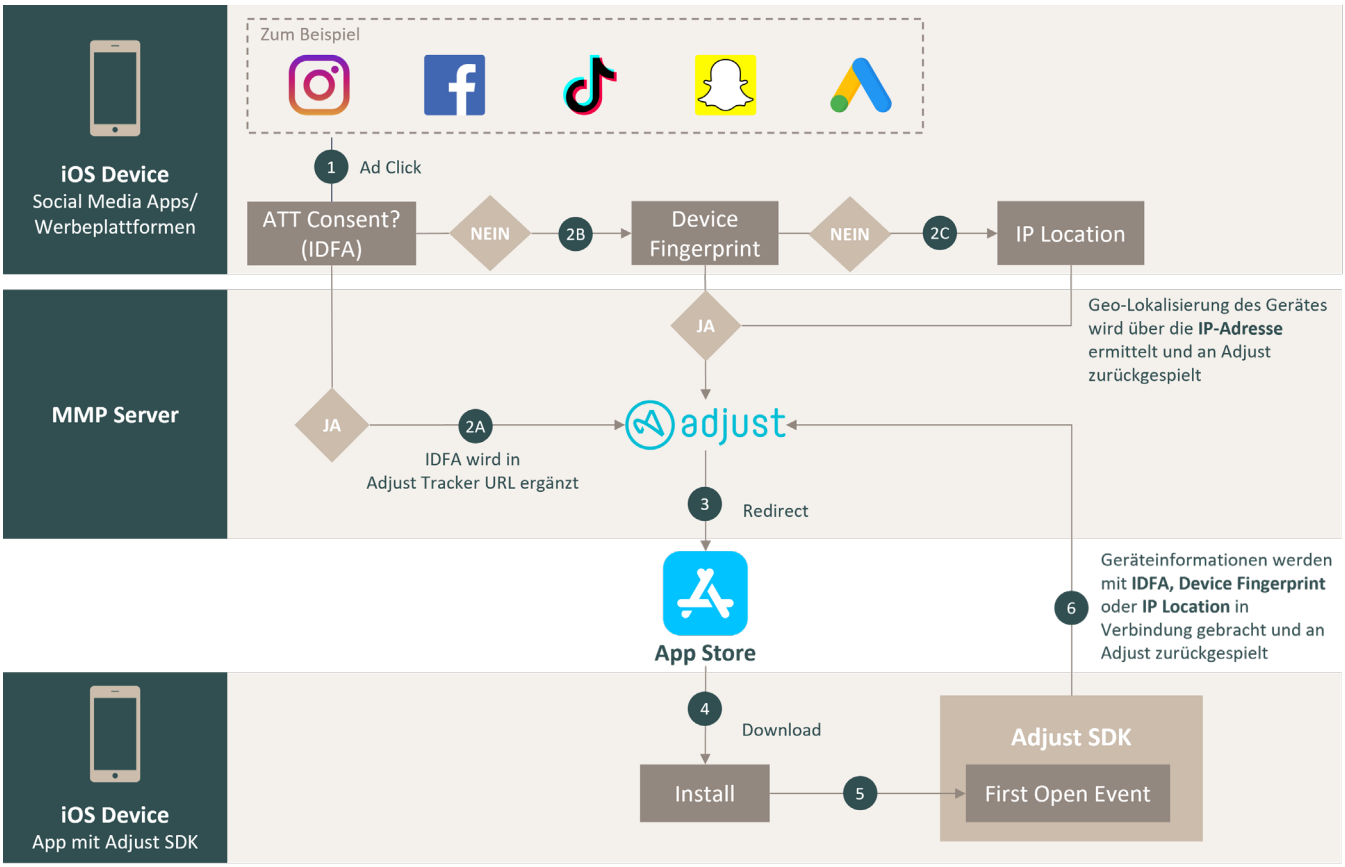
2. Marketing Attribution mobiler Apps

Die Zuordnung und Verknüpfung aller Marketing-Aktivitäten über die Customer Journey eines einzelnen Nutzers stellt die Grundlage jeder datenbasierten Budget-Allokation dar. Mit einer technisch korrekten Marketing Attribution lassen sich festgelegte Conversion-Ziele (z. B. Downloads, First Opens) zuverlässig der entsprechenden Marketingmaßnahme zuordnen. Dadurch lässt sich optimal beurteilen, welchen Einfluss ein Touchpoint auf das jeweilige Conversion-Ziel hatte. Jedoch bringt die korrekte Attribution einzelner Maßnahmen in Abhängigkeit der technischen Ausgangslage mittlerweile einige Herausforderungen mit sich, die wir im Folgenden anhand der aktuellen Möglichkeiten und Services für die Mobile Marketing Attribution vorstellen.

2.1 Mobile Marketing Platforms (MMP)

Standardmäßig lassen sich Aktivitäten außerhalb der App nicht mit der vorherigen Aktivität des Nutzers verbinden, die zu einem Download und dem erstmaligen Öffnen des Spiels geführt haben. Mobile Marketing Platforms (MMP) wie Adjust oder Appsflyer bieten hier eine Lösung, um die Lücke zwischen Erstkontakt und initialem Start der App zu schließen. Dabei erfolgt eine Attribution auf Geräteebene anhand spezifischer Identifier auf Basis eines Last-Click Modells. Aus rechtlicher und technischer Sicht gibt es jedoch ein paar Fallstricke, die bei der Entscheidung für eine Mobile Marketing Platform (MMP) unbedingt beachtet werden sollten. Anhand des folgenden Beispiels wird der technische Workflow für eine korrekte Zuordnung und Funktionsweise durch den Anbieter Adjust einmal genauer für Android- und iOS-Geräte dargestellt.

iOS-Geräte (< iOS 15.0)



1. Klick auf eine Werbeanzeige	Der Nutzer sieht in einem sozialen Netzwerk oder auf einer Werbeplattform die Anzeige zu einer App, welche zum Download in den App Store führt und klickt auf den Call-to-Action.

Fortlaufend besteht nun ein relevanter Unterschied zwischen den Möglichkeiten für eine gerätebasierte Attribution zwischen Android- und iOS-Geräten. Das seit iOS 14.5 von Apple entwickelte App Tracking Transparency (ATT)-Framework erfordert, dass der Nutzer vorab in den Einstellungen des Betriebssystems selbst für jegliche Art von Tracking zustimmt. Ist diese Funktion nicht aktiviert oder lehnt der Nutzer das Tracking aktiv ab, darf die Advertising ID (IDFA) nicht vom Ad Network erfasst werden. Eine Identifikation des Nutzers wird mit der Ablehnung des Trackings somit verhindert und die IDFA genullt. In diesem Fall bietet Adjust Fallback-Lösungen für alle iOS-Geräte < iOS 15.0 an, die in der Reihenfolge ihrer Reliabilität für die Attribution geprüft und angewendet werden.

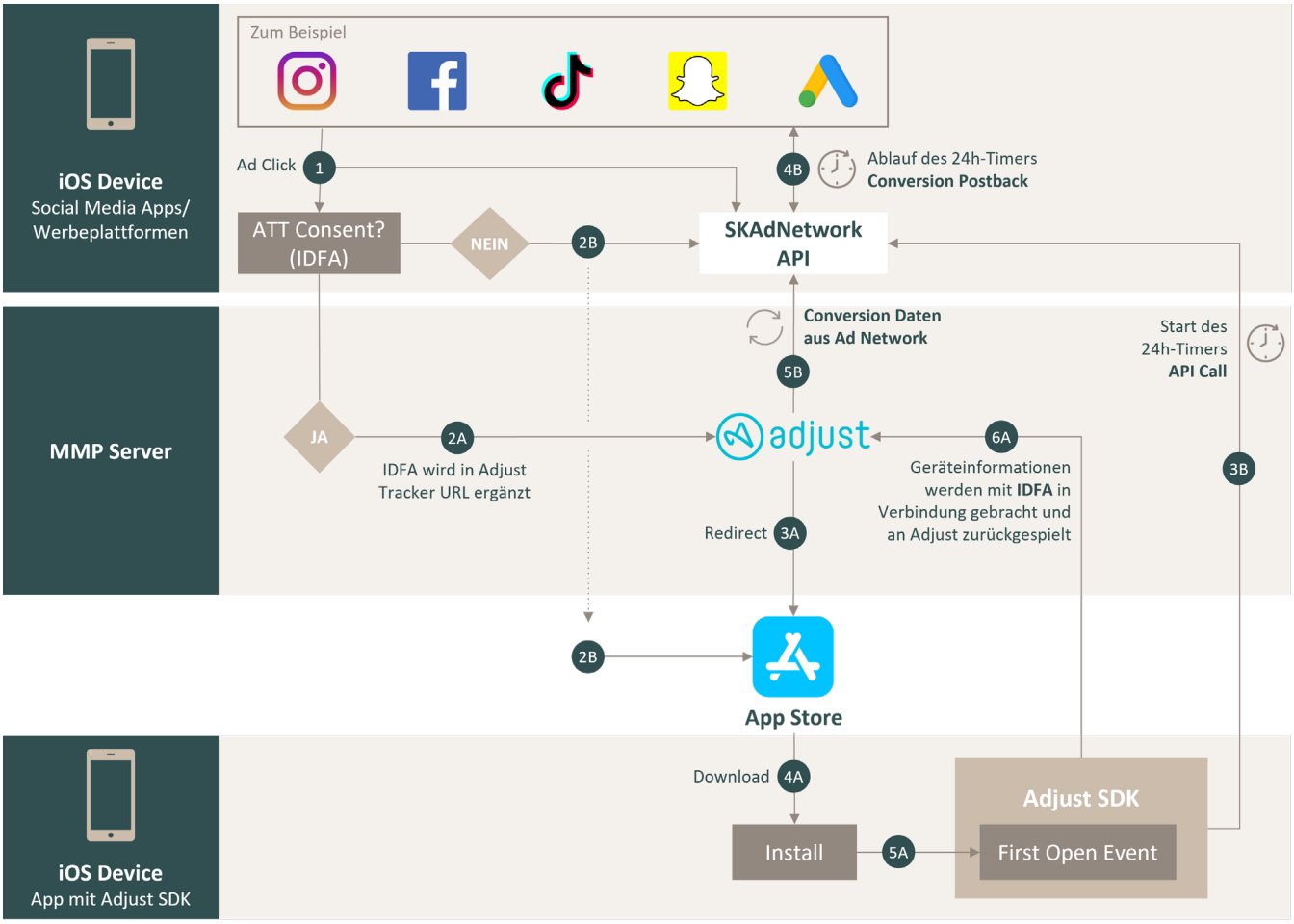
Attribution ohne IDFA-Zugriff

2b. Device Fingerprinting	Lehnt das ATT SDK die Erfassung der Advertising ID (IDFA) ab, wird versucht, ein Device Fingerprint des Gerätes als Identifier zu erstellen. Dieser enthält Informationen aus dem User Agent und der IP-Adresse des Klicks, welcher mit den Daten der Installation abgeglichen wird. Fingerprinting wird bspw. auch standardmäßig als Attributions-Methode für E-Mail Kampagnen verwendet, bei denen keine Advertising ID (IDFA) zur Verfügung steht.
2c. IP Lokalisierung	Ist es ebenfalls nicht möglich, einen Device Fingerprint zu erstellen, wird als letzte Lösung die IP Lokalisierung des Nutzers für das Matching der Informationen zwischen Ad-Click und Install verwendet. Diese Methode sollte jedoch unbedingt als Fallback-Lösung betrachtet werden, da sich Datenpunkte wie die IP-Adresse häufig ändern, was es wiederum erschwert, eine sichere Übereinstimmung zu erzielen.

Attribution mit IDFA-Zugriff

2a. ATT Consent	Liegt der Consent für die Erfassung der Advertising ID (IDFA) vor, wird diese vom Ad Network in der Adjust Tracker URL ergänzt und an Adjust übergeben.
3a. Adjust Netwerk	Im Adjust Network werden die Informationen über die Anzeigen-Interaktion und die Advertising ID (IDFA) erfasst, miteinander verknüpft und gespeichert. In einem Bruchteil von Sekunden wird der Nutzer automatisch an den App Store zum Download weitergeleitet.
4a. Downlaod der App	Der Nutzer lädt die App aus dem App Store herunter. Die App wird erfolgreich auf dem Gerät installiert.
5a. Initiale Öffnen der App (Adjust SDK)	Sobald der Nutzer die App initial öffnet (Event: First Open), wird das in der App integrierte Adjust SDK getriggert. *Bei initialem Öffnen der App findet erneut die Abfrage einer Einwilligung für Tracking statt, sofern in den ATT-Einstellungen des OS noch keine Informationen über die App erfasst wurden oder das Tracking grundsätzlich für ausgewählte Apps zugelassen wird. Ist jedoch bereits ein Tracking der Advertising ID (IDFA) durch die Anzeige im Marketing-Kanal abgelehnt worden, kann selbst bei Einwilligung für das Tracking innerhalb der App die Advertising ID (IDFA) nicht mit dem Marketingkanal verknüpft werden.*
6a. Matching der Informationen	Anschließend sammelt das Adjust SDK Geräteinformationen inklusive der Advertising ID (IDFA), des Device Fingerprints und/oder IP-Location und übermittelt diese an das Adjust Ad Network. Im Adjust Ad Network findet nun ein Matching anhand des jeweiligen Identifiers statt, wodurch die Interaktion mit der Anzeige auf dem Marketingkanal mit der erfolgreichen Installation der App in Verbindung gebracht werden kann.

iOS-Geräte (≥ iOS 15.0)



1. Klick auf eine Werbeanzeige	Der Nutzer sieht in einem sozialen Netzwerk oder auf einer Werbeplattform die Anzeige zu einer App, welche zum Download in den App Store führt und klickt auf den Call-to-Action.

Ab diesem Punkt besteht ein Unterschied in der Anwendbarkeit der zuvor beschriebenen Attributions-Logik für Geräte auf denen bereits iOS 15.0 oder eine neuere Version installiert wurde. Seit der mit iOS 15 eingeführten iCloud Private Relay-Funktion greift nun auch nicht mehr der von Adjust entwickelte Attribution Fallback.
Die Funktion verschlüsselt hierbei DNS-Einträge und erzeugt eine temporäre IP-Adresse für den gesamten unverschlüsselten HTTP-Traffic zwischen Gerät und der aufgerufenen Webseite im Safari-Browser. Zwar hat Apple angedeutet, dass nur HTTP-Traffic innerhalb von Apps durch Private Relay geleitet wird und es somit das App-to-App-Fingerprinting nicht behindert, dies jedoch nicht die Datenkommunikation an Third Party-Anbietern wie Adjust oder Appsflyer mit einschließt. Eine gerätespezifische Attribution wird somit unmöglich, falls kein Opt-In für die Nutzung der IDFA im ATT SDK registriert wurde. Als Alternative für die Messung von Kampagnen ohne Zugriff auf die IDFA wurde bereits 2018 mit der Einführung von iOS 14 das sog. SKAdNetwork (StoreKit Ad Network) von Apple vorgestellt. Das SKAdNetwork kann hierbei parallel als auch bei Verweigerung des Zugriffs auf die IDFA für die Attribution von Installationen genutzt werden.

Attribution ohne IDFA-Zugriff

2b. SKAdNetwork	Bei Ablehnung der Erfassung der Advertising ID (IDFA) durch das ATT SDK, kann auf das SKAdNetwork zurückgegriffen werden. Dieses ersetzt im Wesentlichen die Funktionalität der MMP, indem es die gerätebasierte Attribution in eine datenschutzkonforme Attribution anhand aggregierter Conversion Daten (Installationen) übersetzt. Dem Werbenetzwerk bleibt es somit weiterhin möglich, Rückschlüsse darüber zu treffen, ob eine bestimmte Kampagne zu einer Installation führte, jedoch nicht, welcher spezifische Nutzer diese Installation durchgeführt hat. Nach dem Klick auf die Werbeanzeige wird der Nutzer auf die App-Detailseite des App Stores weitergeleitet. *Zur Nutzung des SKAdNetworks muss die App sowie das verwendete Werbenetzwerk, auf dem die App beworben wird, über die AD Network ID Request Form von Apple zuvor registriert werden. Nach erfolgreicher Registrierung werden spezifische Key-Identifier für die spätere Attribution im Conversion Postback zur Verfügung gestellt. Darunter fallen die Ad Network ID, ein kryptografischer Schlüssel, um eine Signatur für jede Anzeige zu erstellen, die von der App ausgespielt wird sowie eine unique URL für das Empfangen von Postback-Requests für die Validierung der Installationen.*
3b. Start des 24h-Timers	Wurde eine über das SKAdNetwork registrierte App erfolgreich installiert, wird beim initialen Öffnen der App mit einem Call an die SKAdNetwork-API ein 24h-Timer in der App gestartet, welcher nach Ablauf des Zeitfensters die Conversion erfolgreich verifiziert. Wird innerhalb des gesetzten Zeitfensters eine weitere Conversion erfasst, wird der Conversion-Value mit einem weiteren Call an die SKAdNetwork API geupdated und der bestehende 24h-Timer zurückgesetzt. *Beim initialen Öffnen der App wird über die registerAppForAdNetworkAttribution() method ein Request für die Registrierung der Conversion gesendet. Über die updateConversionValue(_:) method kann der Conversion-Value über die registrierte App jederzeit geupdated werden.*

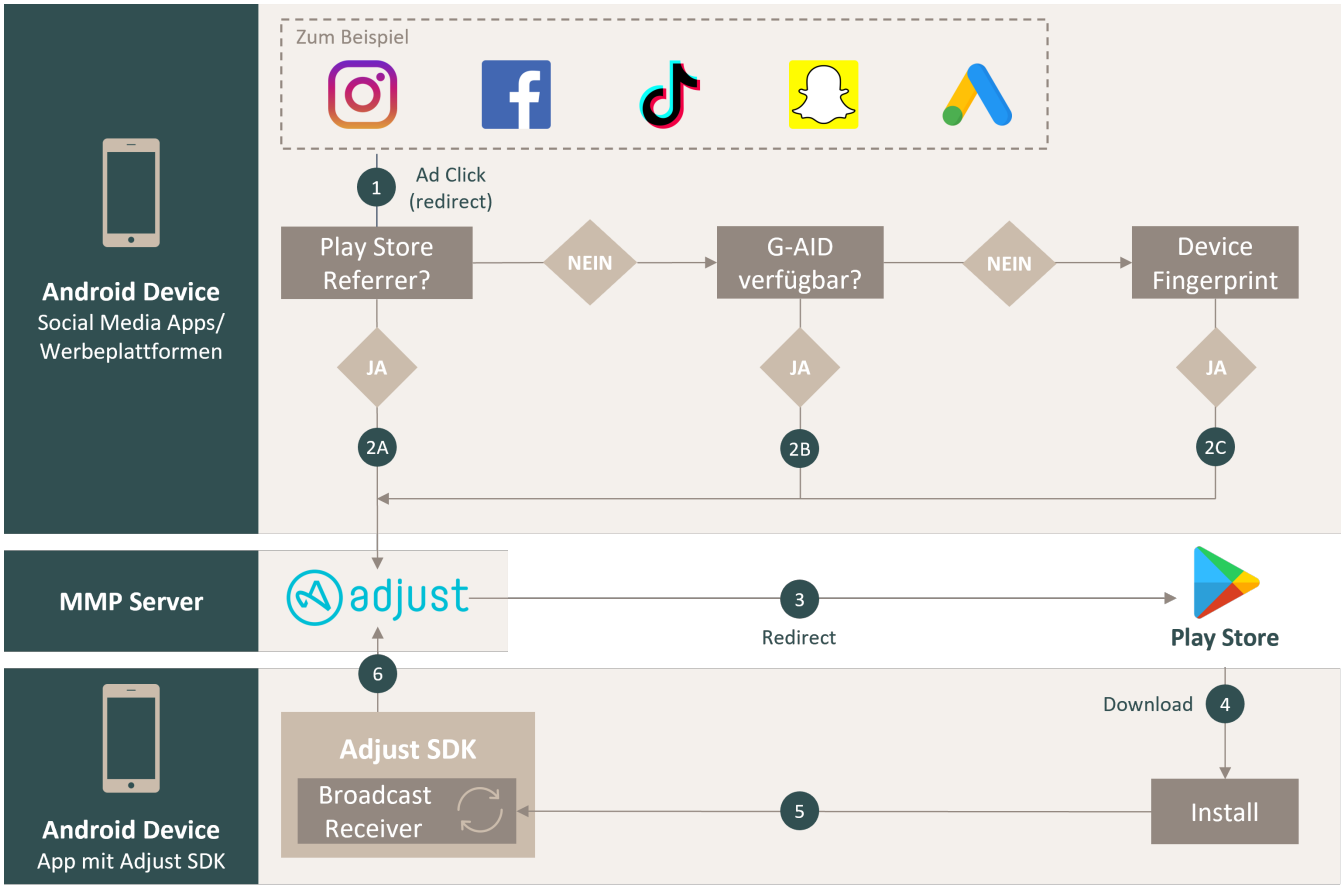
4b.
Conversion Postback
	Nach Ablauf des 24h-Timers kommuniziert die App den endgültigen Conversion-Value an das SKAdNetwork (API), welches die Conversion Daten in einem Conversion Postback an das zugewiesene Werbenetzwerk sendet.
	
	Beispiel: Conversion Postback
	`{`
	`"version" : "2.2",`
	`"ad-network-id" : "com.example",`
	`"campaign-id" : 42,`
	`"transaction-id" : "6aafb7a5-0170-41b5-bbe4-fe71dedf1e28",`
	`"app-id" : 525463029,`
	`"attribution-signature" : "MDTuQ1Z3aEKbxLl5cqZikuY/AO27QIhAJAaiXJffciEHpcV8ZgbKwV9/sY",`
	`"redownload": true,`
	`"source-app-id": 1234567891,`
	`"fidelity-type": 1`
	`"conversion-value": 20`
	`}`

5b. Übermittlung der Conversion Daten zu Adjust	Adjust erhält von dem jeweiligen Ad Network anschließend eine Kopie des Conversion Postbacks mit dem aktuellsten Conversion-Value.

Attribution mit IDFA-Zugriff

2a. ATT Consent	Liegt der Consent für die Erfassung der Advertising ID (IDFA) vor, wird diese vom Ad Network in der Adjust Tracker URL ergänzt und an Adjust übergeben.
3a. Adjust Netwerk	Im Adjust Network werden die Informationen über die Anzeigen-Interaktion und die Advertising ID (IDFA) erfasst, miteinander verknüpft und gespeichert. In einem Bruchteil von Sekunden wird der Nutzer automatisch an den App Store zum Download weitergeleitet.
4a. Downlaod der App	Der Nutzer lädt die App aus dem App Store herunter. Die App wird erfolgreich auf dem Gerät installiert.
5a. Initiale Öffnen der App (Adjust SDK)	Sobald der Nutzer die App initial öffnet (Event: First Open), wird das in der App integrierte Adjust SDK getriggert. *Bei initialem Öffnen der App findet erneut die Abfrage einer Einwilligung für Tracking statt, sofern in den ATT-Einstellungen des OS noch keine Informationen über die App erfasst wurden oder das Tracking grundsätzlich für ausgewählte Apps zugelassen wird. Ist jedoch bereits ein Tracking der Advertising ID (IDFA) durch die Anzeige im Marketing-Kanal abgelehnt worden, kann selbst bei Einwilligung für das Tracking innerhalb der App die Advertising ID (IDFA) nicht mit dem Marketingkanal verknüpft werden.*
6a. Matching der Informationen	Anschließend sammelt das Adjust SDK Geräteinformationen inklusive der Advertising ID (IDFA) und übermittelt diese an das Adjust Ad Network. Im Adjust Ad Network findet nun ein Matching anhand der Advertising ID (IDFA) statt, wodurch die Interaktion mit der Anzeige auf dem Marketingkanal mit der erfolgreichen Installation der App in Verbindung gebracht werden kann.

Android



1. Klick auf eine Werbeanzeige	Der Nutzer sieht in einem sozialen Netzwerk oder auf einer Werbeplattform die Anzeige zu einer App, welche zum Download in den Google Play Store führt und klickt auf den Call-to-Action.

Ab diesem Punkt - wird wie bei iOS-Geräten auch - eine Validierung vorhandener Identifier für die Identifizierung des Gerätes durch das Ad Network ausgelöst. Adjust bezeichnet diesen Fallback-Mechanismus als “Attribution Waterfall”. Hierbei wird die Verfügbarkeit der Identifier in der Reihenfolge ihrer Reliabilität für die Attribution geprüft und angewendet.

Attribution mit reftag

2a. Play Store Referrer	Erkennt Adjust einen Play Store Referrer-Link, wird dem Referrer ein sogenanntes reftag hinzugefügt, welches zur Identifizierung der Interaktion dient. Anschließend wird der Nutzer an den Google Play Store weitergeleitet.

Attribution ohne reftag

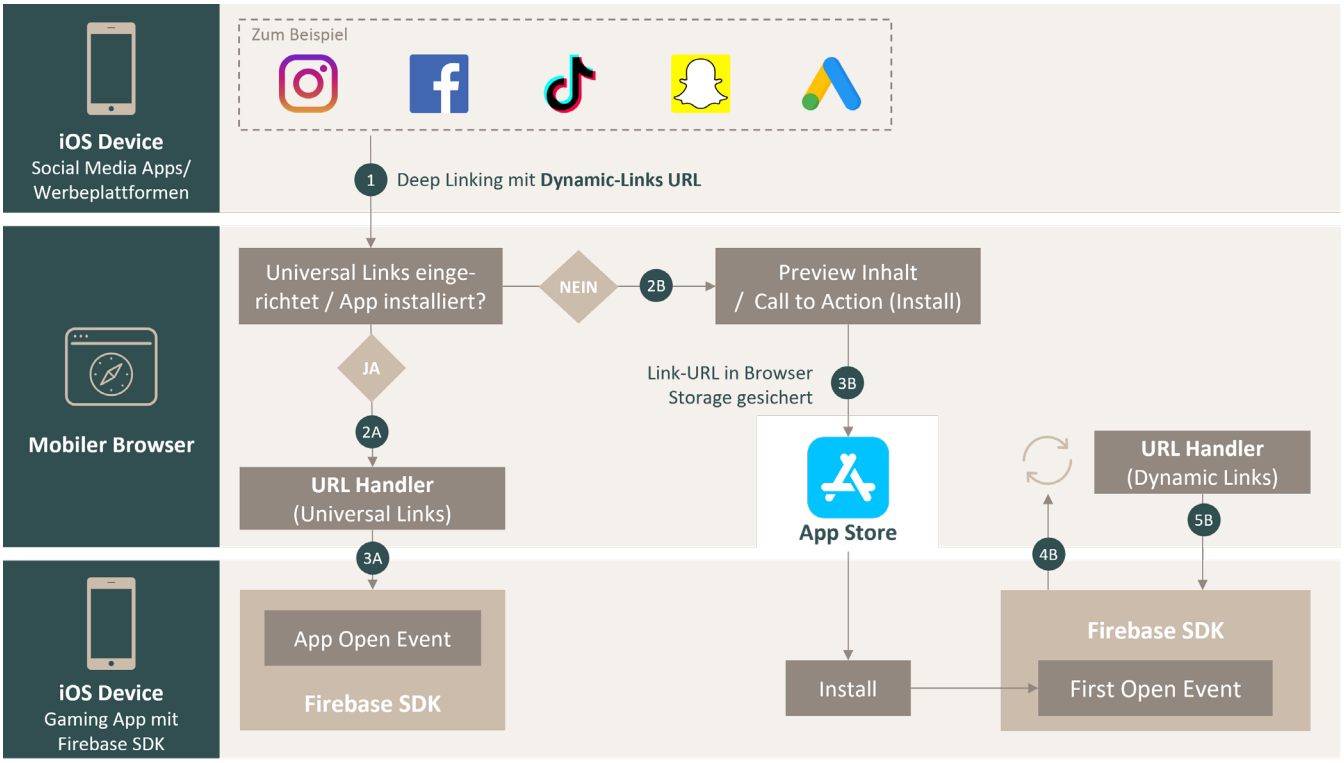
2b. Google Advertising ID (GAID)	Die Google Advertising ID (GAID) ist in Android das Pendant zu dem Identifier for Apps (IDFA) von iOS. Stehen dem Adjust Ad Network keine Informationen zu einem Play Store Referrer zur Verfügung wird die Google Advertising ID (GAID) für die Identifizierung des Gerätes genutzt. *Aktuell kann die Google Advertising ID (GAID) noch problemlos durch Third Party-Anbieter wie Adjust ohne aktive Einwilligung durch den Nutzer von dessen Gerät abgefragt werden. Über Android-Geräte ist eine Attribution daher auch bei fehlendem Play Store Referrer aktuell noch ohne rechtliche Einschränkungen möglich. Orientiert am App Tracking Transparency (ATT) Framework von Apple, soll jedoch nun auch mit der kommenden Version (12.0) des Android-Betriebssystem der Zugriff auf die Google Advertising ID (GAID) vom Hinzufügen einer zusätzlichen Berechtigung in der AndroidManifest-Datei von Android-Apps abhängig sein.* *Auf Android 12 lässt sich die Google Advertising ID (GAID) daher ebenfalls nicht mehr auslesen, wenn keine Berechtigung erteilt wurde. Ist der Parameter isLimitAdTrackingEnabled() auf true gesetzt, gibt die API automatisch eine Reihe von Nullen für die GAID zurück.* *Der Nutzer hat die Möglichkeit, direkt über die Privacy Settings des Betriebssystems entweder die GAID zurückzusetzen oder vollständig zu löschen.*
2c. Device Fingerprinting	Besteht kein Zugriff auf die Google Advertising ID (GAID) wird versucht, ein Device Fingerprint des Gerätes als Identifier zu erstellen. Dieser enthält Informationen aus dem User Agent und der IP-Adresse des Klicks, welcher mit den Daten der Installation abgeglichen wird.
3. Google Play Store	Im Playstore wird der mit dem reftag deklarierte Play Store Referrer temporär gespeichert. Kommt es anschließend zum Download, werden die Informationen mit der Installation in die App weitergegeben.
4. Adjust SDK	Mit der Installation der App wird auch das in der App installierte Adjust SDK ausgeführt, welches über einen Broadcast Receiver verfügt. Dieser untersucht empfangene Messages laufend nach Referral-Informationen.
5. Matching der Informationen	Der aus dem Play Store übermittelte reftag wird vom Broadcast Receiver erkannt. Daraufhin wird die Information über den erfolgreichen Install an Adjust übermittelt und anhand des reftag mit dem initialen Klick auf die Werbeanzeige in Verbindung gebracht. Wurde zu diesem Zeitpunkt für die Attributions-Logik bspw. die Google Advertising (ID) oder ein Device Fingerprinting gewählt, da keine Play Store Referrer-Informationen verfügbar waren, werden diese Informationen vom Adjust SDK im Gerät abgefragt und nach Übermittlung an das Adjust Ad Network mit dort vorhandenen Informationen abgeglichen, um die Attribution zuzuordnen.

2.2 Deep Linking (Dynamic Links)

Ein sogenannter Deep Link beschreibt die Ziel-URL in einer Anzeige, mit der man auf eine bestimmte Seite oder einen bestimmten Inhalt in einer App weitergeleitet wird. Oftmals treten jedoch Problematiken in der Übergabe der Deep Link-Parameter auf, wenn die App vom Nutzer bisher noch nicht installiert wurde. Link-Parameter gehen zwischen Interaktion mit der Anzeige und Installation der App verloren. Es entsteht ein Bruch in der Attribution, wodurch die Maßnahme nicht mehr zielführend mit dem Download verknüpft wird.
Google Firebase hat für dieses Problem das Dynamic Links SDK entwickelt. Deep Links lassen sich somit mithilfe der Firebase-Konsole, einer REST-API, iOS- oder Android-Builder-API oder mittels einer URL erstellen, indem dynamische Link-Parameter zu einer für die App spezifischen Domain hinzugefügt werden. Wenn der Nutzer nun einen Dynamic Link öffnet, die App jedoch noch nicht installiert ist, wird der Nutzer an den Play- oder App Store gesendet, um die App zu installieren. Anschließend wird der initial an die App übergebenen Dynamic-Link beim Öffnen vom Dynamic Links SDK abgefragt und der Nutzer auf den spezifischen Inhalt weitergeleitet.

In dem Umgang mit Dynamic Links existieren zwischen den Betriebssystemen iOS und Android jedoch einige Unterschiede. Anhand der folgenden Schaubilder wird deshalb der technische Ablauf der Attribution von Dynamic Links detailliert für beide Betriebssysteme dargestellt.

iOS



1. Klick auf eine Werbeanzeige	Der Nutzer sieht in einem sozialen Netzwerk oder auf einer Werbeplattform die Anzeige zu einer App, welche einen Deep Link enthält, der auf spezifische Inhalte in der App verlinkt. An dieser Stelle findet eine Prüfung statt. Ist die App bereits installiert, wird der Nutzer zu dem ausgewiesenen Inhalt weitergeleitet. Ist die App dem Gerät nicht bekannt, wird der Nutzer zum Download in den App Store weitergeleitet.

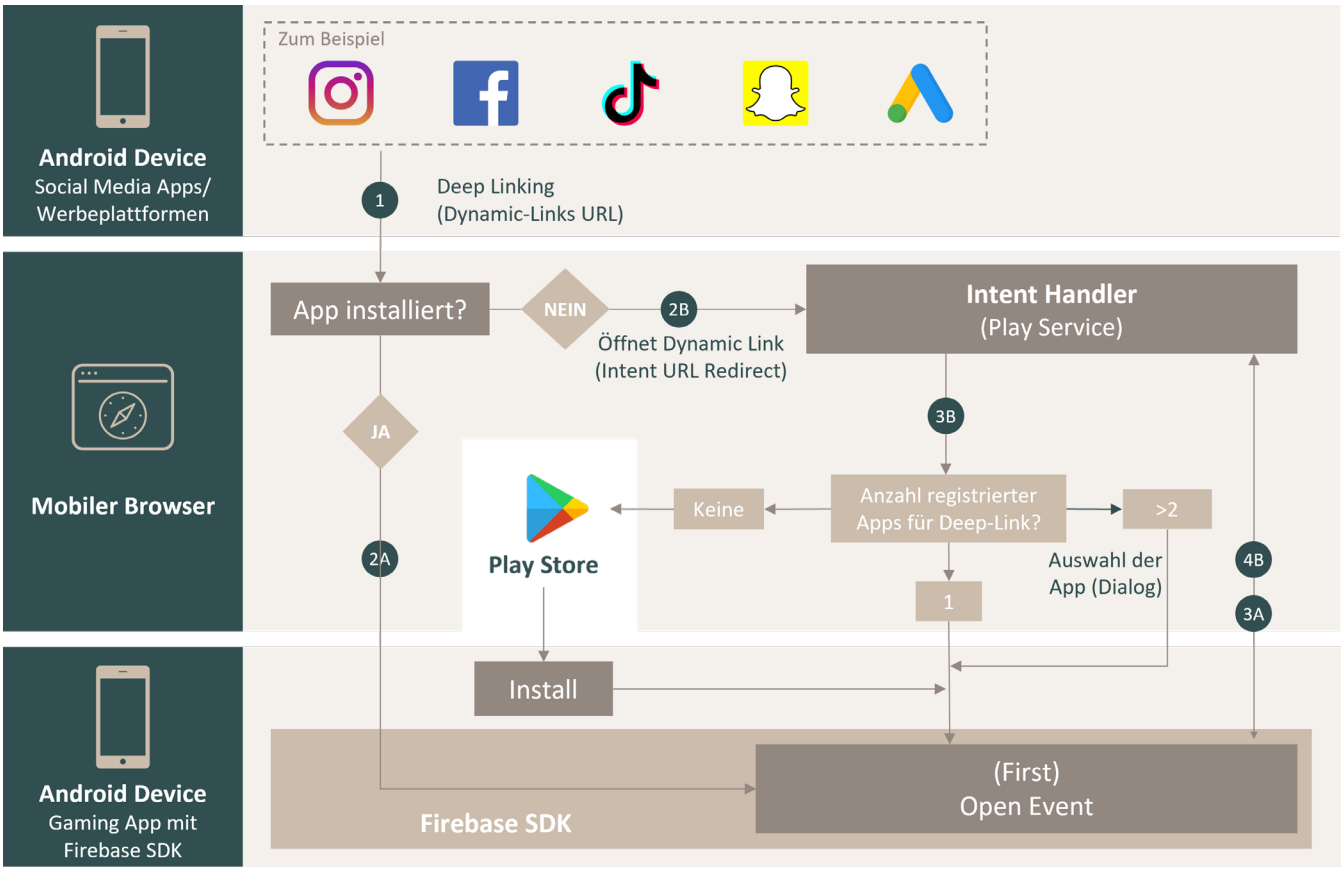
Deep Linking bei abgeschlossener Installation

2a. URL Handler	Ist die App bei Klick auf die Anzeige bereits auf dem Gerät installiert, wird der Dynamic Link über einen URL Handler (*handleUniversalLink*) im Dynamic Links SDK als normaler Universal Link behandelt. *Universal Links bieten die Möglichkeit, spezielle Website-Inhalte mit Inhalten in einer App zu verknüpfen. Hierfür wird von Firebase automatisch ein apple-app-site-association-File auf der Website mit Preview-Inhalten der App implementiert, welches iOS mitteilt, dass der auf der Website verfügbare Content ebenfalls in der App wiederzufinden ist. Ist die App also installiert, erkennt iOS automatisch, dass es sich um In-App Inhalte handelt und redirected den User in die App, anstelle auf die Preview-Inhalte des mobilen Browsers (Safari).*
3a. Attribution des Dynamic Links	Der URL Handler ruft eine Funktion im Firebase SDK der App auf, welche die Deep Link URL aus der Dynamic Links URL extrahiert und mit der Firebase Instance-ID als Identifier verknüpft.

Deep Linking bei ausstehender Installation

2b. Mobile Preview der Inhalte	Ist die App noch nicht auf dem Gerät installiert, wird nach dem Klick auf den Dynamic Link automatisch der mobile Browser des Gerätes geöffnet. Dort wird dem Nutzer eine Vorschau der spezifischen App-Inhalte mit einem Call-to-Action zum Download präsentiert.
3b. Download und Installation der App	Während der Nutzer über den Download Link in der Vorschau zum App Store weitergeleitet wird, speichert der Browser die Dynamic Link-URL in der Zwischenablage des mobilen Browsers ab. Anschließend installiert der Nutzer die App auf dem Gerät.
4b. Initiale Öffnen und Aufruf des URL Handlers	Wird die App nach erfolgreicher Installation initial geöffnet, erstellt das in der App implementierte Firebase SDK einen Dynamic Link nach dem App-spezifischen Schema, welches es aus der Dynamic Link URL des Browser Storage extrahiert und ruft anschließend den URL Handler auf.
5b. Attribution des Dynamic Links	Der URL Handler ruft eine Funktion im Firebase SDK der App auf, welche die Deep Link URL aus der Dynamic Links URL extrahiert und mit der Firebase Instance-ID als Identifier verknüpft. *Die Firebase Instance-ID ist ein spezifischer Identifier der App, welcher einmalig für jeden Install generiert wird. Er dient lediglich zur Identifikation von Interaktionen innerhalb der App und bietet daher keine Aufschlüsse über geo- und demografische Informationen wie der Identifier for Apps (IDFA). Werden diese Daten jedoch mit Drittanbietern wie bspw. Google Analytics oder Mobile Marketing Platforms (MMP) geteilt, ist die Einwilligung durch den Nutzer über das ATT SDK gefordert. In dem beschriebenen Beispiel wird lediglich die Verknüpfung des Dynamic Linkings mit der App-Instance ID innerhalb des Firebase Analytics eingegangen.*

Android



1. Klick auf eine Werbeanzeige	Der Nutzer sieht in einem sozialen Netzwerk oder auf einer Werbeplattform die Anzeige zu einer App, welche einen Deep Link enthält, der auf spezifische Inhalte in der App verlinkt. Klickt der Nutzer nun auf die Anzeige, findet eine Prüfung statt.

Deep Linking bei abgeschlossener Installation

2a. Weiterleitung in die App	Ist die App bereits erfolgreich auf dem Gerät des Nutzers installiert und der Link zur App ist in Dynamic Links registriert, wird die App geöffnet und der Nutzer auf den ausgewiesenen Inhalt weitergeleitet.
3a. Initiale Öffnen und Aufruf des Intent Handlers	Beim Öffnen der App wird der Intent Handler getriggert und nach der für die App hinterlegten Dynamic Link URL abgefragt.
4a. Attribution des Dynamic Links	Der Intent Handler ruft anschließend eine Firebase Funktion auf, um die Deep-Link URL aus dem Handler zu extrahieren und mit der Firebase Instance-ID zu verknüpfen.

Deep Linking bei ausstehender Installation

2b. Aufruf des Intent Handlers	Ist die für den Dynamic Link registrierte App noch nicht auf dem Gerät installiert, öffnet das Betriebssystem automatisch den Standard-Webbrowser des Nutzers über den Dynamic Link aus der Anzeige. Die durch Dynamic Links automatisch generierte Website leitet den Nutzer direkt zu einer Android Intent URL, wodurch ein Intent Handler aufgerufen wird.
3b. Handling des Deep-Links	Der Intent Handler prüft, wie viele Apps sich auf dem Gerät befinden, die bereits für Dynamic Links registriert sind und die spezifische Deep Link-URL händeln können. Das Prüfungsschema des Handlers sieht dabei folgende drei Szenarien vor: **Keine registrierte App:** Befindet sich keine App auf dem Gerät, die für den Deep-Link registriert ist, leitet der Intent Handler den Nutzer automatisch in den Play Store zum Download weiter. **Eine registrierte App:**Ist bereits eine App auf dem Gerät installiert, welche über Dynamic Links registriert wurde, um den Deep-Link zu händeln, wird diese automatisch geöffnet. **Mehr als zwei registrierte Apps:** Sind mehrere Apps auf dem Gerät installiert, die für Dynamic Links registriert sind, wird dem Nutzer ein Pop-up ausgeliefert, welches in einem Dialog zur Auswahl der gewünschten App auffordert. Wählt der Nutzer die gewünschte App aus, wird diese vom Intent Handler anschließend geöffnet.
4b. Aufruf des Intent Handlers	Das in der App implementierte Firebase SDK und Dynamic Links SDK wird geladen und stellt eine Abfrage für die Deep Link URL an den Intent Handler. Der Handler ruft anschließend eine Firebase Function auf, um die Deep Link-URL aus dem Intent zu extrahieren.

2.3 UTM Campaign Tracking

Im Gegensatz zu Google Analytics, verfügt die Firebase Analytics Console nicht über ein Default Channel Grouping für Traffic Quellen. Stattdessen werden die Werte für Quelle, Medium und Kampagne gespeichert und direkt in den Firebase Conversion-Berichten angezeigt. Standardmäßig gruppiert Firebase Analytics Traffic Quellen wie folgt:

- **Direct/None**
  Stehen keine Informationen zu einer Kampagne (Bspw. Link-Parameter) zur Verfügung, wird der Traffic von Firebase Analytics als *Direct/None* klassifiziert. Dies kann der Fall sein, wenn der Nutzer auf die App zugreift, aber von einer Quelle kommt, die von Firebase Analytics nicht erkannt wurde oder Firebase Analytics nicht in der Lage war, die Informationen über die Quelle abzurufen. Einige Beispiele hierfür sind organische Apple Search-Links, Google Ads iOS Search App-Install Kampagnen und andere nicht deklarierte Kampagnen. Die Klassifizierung als *Direct/None* kann auch erfolgen, falls der Nutzer über einen Link zum Play Store weitergeleitet wurde, der keine UTM-Parameter enthält.

- **google-play/referral**
  Diese Klassifizierung erfolgt, wenn der Nutzer über einen Link, der einen Referrer enthält, auf die Play Store-Liste geleitet wird oder die App organisch durch eine Suche im Google Play Store gefunden wurde.

- **Apple/Search**
  Eine App wurde als Ergebnis eines Klicks auf eine Apple Search Ad installiert.

  Für eine korrekte Zuordnung von Apple Search Ads ist die Integration des iAd-Framework in der Xcode-Projektdatei der App notwendig.

- **(not set)**
  Wenn einer der Werte für das Quellmedium, die Kampagne, den Typ des Werbenetzwerkes oder das Werbemittel fehlt, klassifiziert Firebase Analytics den Traffic als *(not set).*

Firebase kann automatisch den Traffic von einer begrenzten Anzahl von Quellen erfassen. Bei anderen Plattformen wird die Kampagnenquelle als direct eingestuft. Für die Analyse gibt es jedoch einen Workaround, mit dem Daten korrekt der Kampagnenquelle zugeordnet werden können. Dazu müssen wir zwei Schritte befolgen:

1. **UTM-Kennzeichnung der Deep-Links**
   Um Traffic toolseitig klassifizieren zu können, muss sichergestellt werden, dass alle Links mit UTM-Parametern gekennzeichnet sind. Hierfür bietet Google Play den sogenannten Google Play URL Builder an, um das nötige Schema einzuhalten.

2. **Fetching und Übermittlung der UTM Parameter an Firebase Analytics**
   Je nach verwendetem Betriebssystem unterscheidet sich die Vorgehensweise punktuell. Der grundlegende technische Vorgang bleibt jedoch relativ gleich. Folgend wird die Vorgehensweise zur Übermittlung der UTM Parameter zu Firebase Analytics am Beispiel des Android Betriebssystem erläutert:

A. Hinzufügen der Dynamic Links Library im Code der App

```
implementation 'com.google.firebase:firebase-dynamic-links-ktx:19.1.0'
implementation 'com.google.firebase:firebase-analytics:17.3.0'
```

B. Akzeptieren der Nutzungsbedingungen

Nach Hinzufügen der Dynamic Links Library im Code der App muss im jeweiligen Firebase-Projekt über den Abschnitt Dynamische Links in der Firebase-Console den Nutzungsbedingungen von Dynamic Links zugestimmt werden.

C. Hinzufügen eines Intent-Filters in der AndroidManifest.xml

```
<intent-filter>
    <action android:name="android.intent.action.VIEW"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <category android:name="android.intent.category.BROWSABLE"/>
    <data
        android:host="example.com"
        android:scheme="https"/>
</intent-filter>
```

Ein *Intent* ist ein Messaging-Objekt, mit dem sich eine Aktion von einer anderen App-Komponente (Aktivität, Service, Broadcasting etc.) anfordern lässt. Um *Intents* für eine Komponente zu definieren, wird ein sog. *Intent-Filter* in der Manifestdatei der App erstellt. Indem bspw. ein *Intent-Filter* für eine Aktivität deklariert wird, wird es anderen Apps ermöglicht, die Aktivität direkt mit einem *Intent* zu starten.

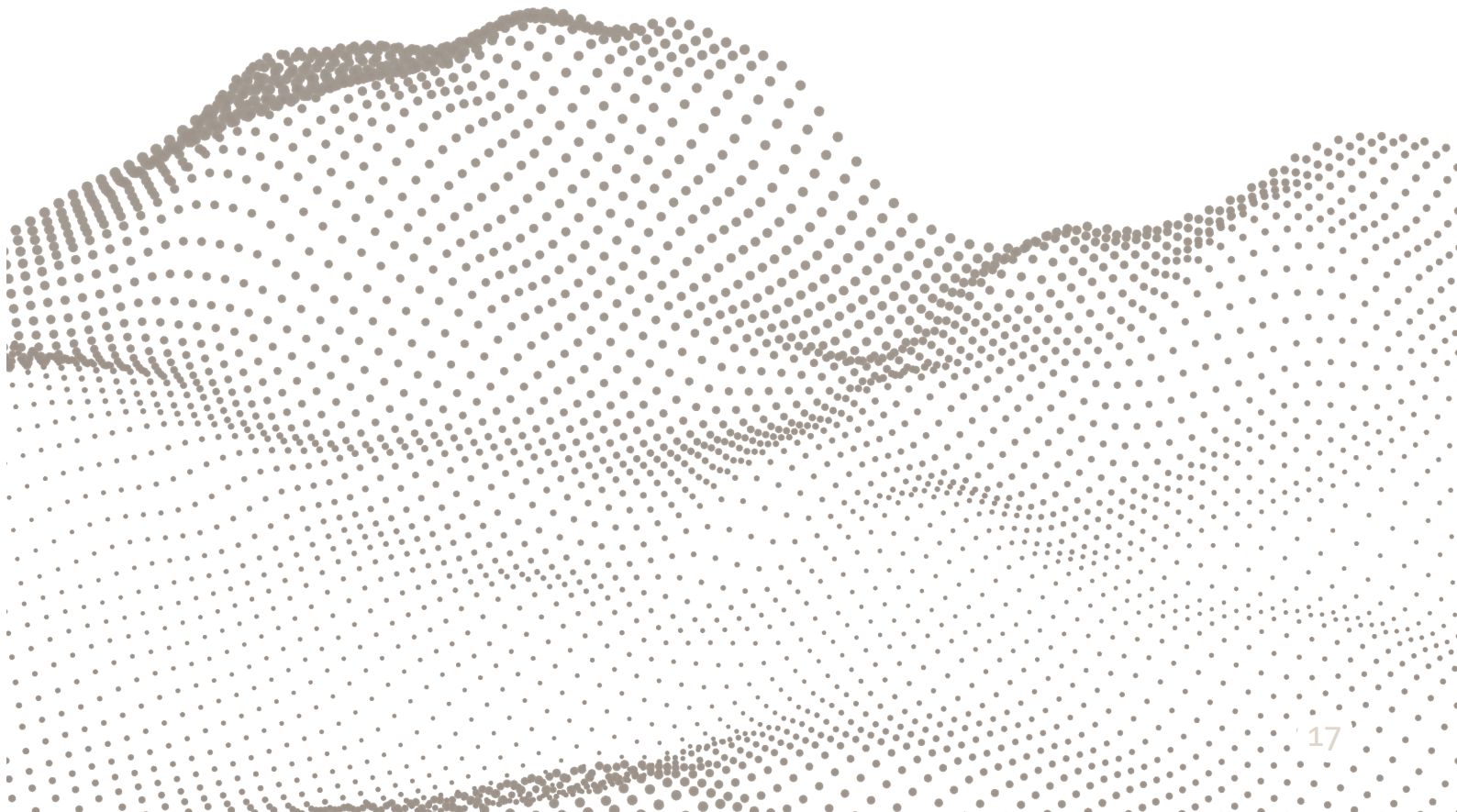
D. Handling der UTM-Parameter im Deep Link

```
Firebase.dynamicLinks
.getDynamicLink(intent)
.addOnSuccessListener(this) { pendingDynamicLinkData ->
    // Ausgabe des Deep Links (null falls nicht vorhanden)
    var deepLink: Uri? = null
    if (pendingDynamicLinkData != null) {
        deepLink = pendingDynamicLinkData.link
    }
    // Auslesen der UTM-Parameter aus dem deeplink object

    String source = String.valueOf(deepLink. getQueryParameters("utm_source"));
    String medium = String.valueOf(deepLink. getQueryParameters("utm_medium"));
    String campaign = String.valueOf(deepLink. getQueryParameters("utm_campaign"));
    if(cs !=null && cn !=null){

        // Hinzufügen der UTM-Parameter als Event Properties im params object
        Bundle params = new Bundle();
        params.PutString(FirebaseAnalytics.param.SOURCE, source);
        params.PutString(FirebaseAnalytics.param.MEDIUM, medium);
        params.PutString(FirebaseAnalytics.param.CAMPAIGN, campaign);
        mFirebaseAnalytics.logEvent(FirebaseAnalytics.Event.CAMPAIGN_DETAILS, params);
        mFirebaseAnalytics.logEvent(FirebaseAnalytics.Event.APP_OPEN, params);
}
.addOnFailureListener(this) { e -> Log.w(TAG, "getDynamicLink:onFailure", e) }
```

Der Code sendet die UTM-Parameter mit den Events *app_open* und *campaign_details*. Die Vorgehensweise kann jedoch auf jedes beliebige Event angepasst werden, mit dem die Kampagnen-Parameter verschickt werden sollen.

3. Ausblick

Eine lückenlose Attribution ist und war schon immer eine Herausforderung für die digitale Analyse und das digitale Marketing. Jedoch stößt der einstige “Wilde Westen” des User Trackings auf wachsende Hürden, die ein grundlegendes Umdenken in der Erhebung und Weiterverarbeitung von Daten erfordern. Gerade deshalb ist das Thema Mobile Attribution relevanter denn je, da in Zukunft herkömmliche Methoden an Reliabilität verlieren. Vorgestellte Attributions-Provider, wie Mobile Marketing Platforms, die bisher auf die Weiterverarbeitung und Nachverfolgung von Daten aus Third Party-Zusammenhängen setzen, stellen durch steigende Anforderungen an die ePrivacy von mobilen Nutzern langfristig ein aussterbendes Modell dar. (Mobile) Attribution kann zukünftig daher nicht mehr nur mit verfügbarer Datenbasis reliabel betrieben werden, sondern erfordert intelligente Messverfahren, um Lücken in der Customer Journey zu schließen. Der Einsatz probabilistischer Methoden wird für die Attribution daher zunehmend relevanter. Probabilistische Modelle berücksichtigen hierbei zusätzlich Zufallsvariablen und Wahrscheinlichkeitsverteilungen. Anhand bestimmter Touchpoint-Kombinationen auf der Grundlage historischer Customer Journey-Daten, berechnet das Modell Übergangswahrscheinlichkeiten für nachgelagerte Touchpoints. Somit lassen sich lückenhafte Customer Journeys anhand errechneter Touchpoint-Kombinationen intelligent abbilden. Da also die Quantität an Daten über die Customer Journey weiter abnimmt, wird die Qualität verfügbarer Customer Journey Daten zunehmend relevanter, um die Aussagekraft probabilistischer Modelle zu erhöhen.



Über mohrstade

Unternehmen

mohrstade ist eine Beratung für Marketing Technologie in München und Hamburg. mohrstade ist spezialisiert auf Projekte in den Bereichen Data Collection, Data Management, Analytics, Marketing Activation und Data Visualization. Diese Services bietet mohrstade in zertifizierten Partnerschaften mit Marketing Software Herstellern an.

Managing Partner

Patrick Mohr

Co-Founder & Managing Partner

Patrick ist Gründer und Geschäftführer von mohrstade. Bereits während seines Studiums für BWL, Finance und Information (MSc) sammelte er Erfahrungen im Management Consulting. Später arbeitet er als SEA Manager, Data Scientist und Analytics Consultant bei Rocket Internet, Group M und UDG. 2017 baute er schließlich den Münchner Standort von Trakken auf. Parallel arbeitet er als Dozent an Universitäten. Darüber hinaus ist er Co-Organisator von Analytics Pioneers - der größten Analytics Community im DACH-Raum.

patrick@mohrstade.de

Marcus Stade

Co-Founder & Head of Analytics

Marcus ist Gründer von mohrstade und Head of Analytics. Darüber hinaus ist er Co-Organisator von Analytics Pioneers - der größten Analytics Community im DACH-Raum. Zuvor hat er im Bereich Web-Development und Online-Marketing gearbeitet. Auf seinem Blog www.marcusstade.de schreibt er regelmäßig zu Themen der Digitalen Analyse.

marcus@mohrstade.de

mohr stade

Mohr & Stade GmbH
Schillerstraße 14
80336 München

www.mohrstade.de