

Technical Disclosure Commons

Defensive Publications Series

January 14, 2019

Verified revenue sharing between distributor OEM and app developers

John Dukellis

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Dukellis, John, "Verified revenue sharing between distributor OEM and app developers", Technical Disclosure Commons, (January 14, 2019)
https://www.tdcommons.org/dpubs_series/1880



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Verified revenue sharing between distributor OEM and app developers

ABSTRACT

To increase the distribution of their apps, software developers often enter into agreements with mobile device original equipment manufacturers (OEMs) to distribute their apps, e.g., pre-installed on devices shipped by the OEM. OEMs control device software, including the home screen, and pre-install apps on devices upon upfront payment by the app developer. Smaller app developers may not be able to reach agreements with OEMs due to the cost of the sale, e.g., the upfront payment.

The techniques of this disclosure enable OEMs and app developers to reach agreements for the OEM to distribute an app for low or no upfront fee in exchange of a percentage of future ad revenue generated by the app. Total ad revenue generated is often split between an ad network and an app developer, and in this case the OEM can either (i) take a share of the ad revenue portion typically provided to the app developer, (ii) take a share of the ad revenue portion due to the ad network, (iii) take a share of both, or (iv) play the role of ad network or app developer while also taking a share of the other party's portion. The techniques ensure that the agreements are enforceable and protect both the OEM and the app developer against fraud or malicious action.

KEYWORDS

- App distribution
- Pre-installed software
- Channel partner
- Original equipment manufacturer (OEM)
- OS customization

- Ad network

BACKGROUND

To increase the distribution of their apps, software developers often enter into agreements with mobile device original equipment manufacturers (OEMs) to distribute their apps, e.g., pre-installed on devices shipped by the OEM. OEMs control device software, including the home screen, and pre-install apps on devices upon upfront payment by the app developer. Smaller app developers may not be able to reach agreements with OEMs due to the cost of the sale, e.g., the upfront payment. OEMs that are unable to find suitable apps to ship the device with are disadvantaged, for the revenue opportunities represented by home screens on their devices go unfilled.

DESCRIPTION

Per the techniques of this disclosure, OEMs agree to distribute an app for low or no upfront fee in exchange of a percentage of future ad revenue generated by the app. The main technical problem here is to ensure that the revenue split between app developer and OEM occurs in a foolproof and trustworthy manner, e.g., the revenue split occurs when the OEM has truly distributed the app, but not otherwise. Further, attempts by an app developer to avoid revenue-split arising from OEM-distributed apps, e.g., by re-downloading the app, are thwarted. Similarly, attempts by an OEM to force revenue-split for apps distributed by a non-OEM channel are prevented. Implementations include setting up the OEM and the app developer to share ad revenue, and subsequently, splitting ad revenue in a trustworthy manner.

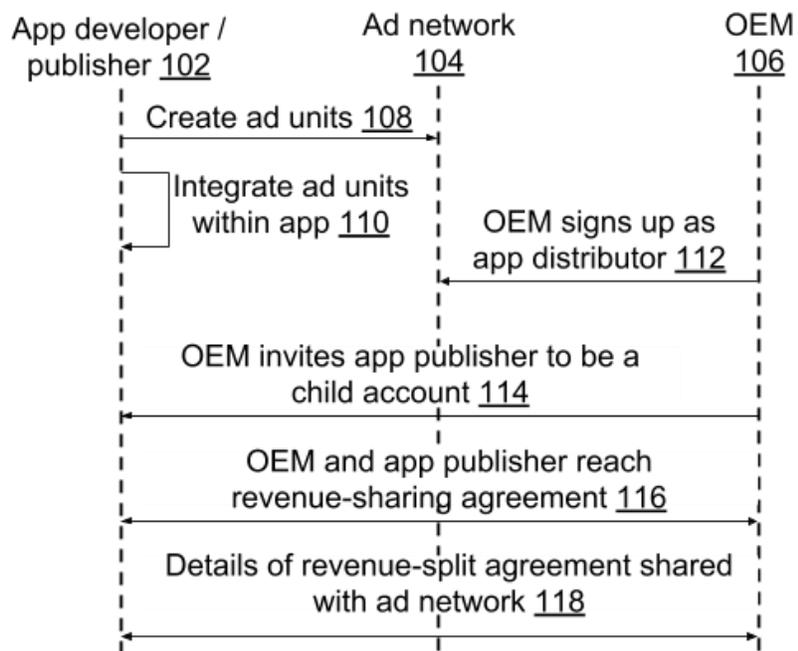


Fig. 1: Setting up the OEM and the app developer to split ad revenue

Fig. 1 illustrates an example process of setting up an OEM and an app developer to split ad revenue, per techniques of this disclosure. The app developer (or publisher) (102) creates ad units with an ad network (104) that administers the splitting of revenue. The app developer integrates ad units within the app (110). The OEM (106) signs up with the ad network as an app distributor (112) and invites the app developer to be a child account (114). The OEM and app developer reach a revenue-sharing (revenue-split) agreement (116). Details of the agreement are shared with the ad network (118).

The invitation and onboarding flow, e.g., as illustrated in the example of Fig. 1, is advantageously of self-service type. At the conclusion of the flow, the OEM is provided with access to a master screen where the OEM can see their accounts, activity, and revenue. The developer similarly is provided access to a screen that shows the revenues that are flowing to them, as well as (optionally) what the OEM is receiving as ad revenue. There is a provision for the OEM to enter the place where they want their revenue to be paid. Either party can cancel the

agreement via a button or setting, e.g., after 30 days or some other previously agreed upon grace period. Further, a setting that allows the OEM to subtract hard marketing costs, if any, from the pay-out to the developers can also be provided. As mentioned above, the ad network pays the developer and the OEM their respective shares.

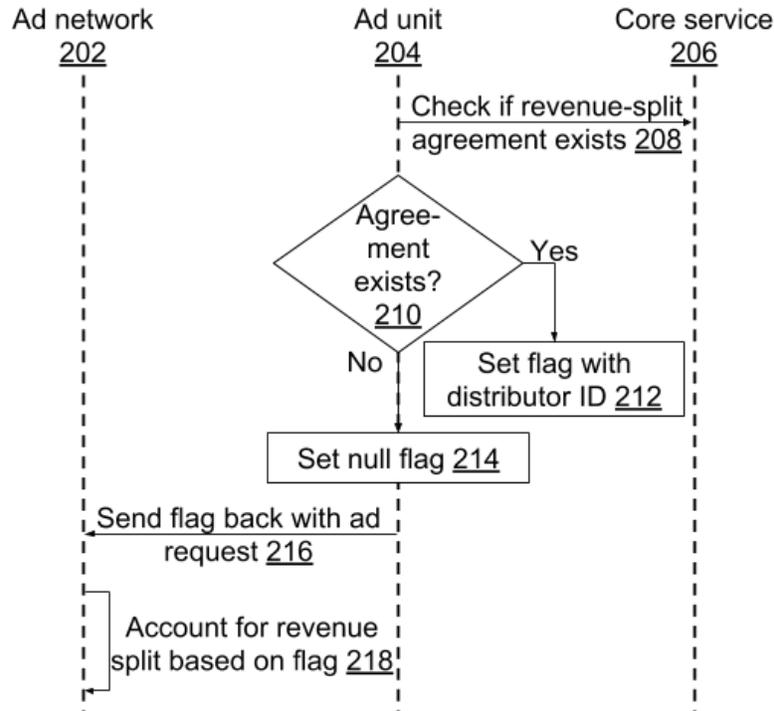


Fig. 2: Splitting ad revenue in a trustworthy manner

Fig. 2 illustrates an example process of splitting ad revenue between OEM and app developer in a trustworthy manner. An ad unit (204), which is integrated with the app, e.g., through the software development kit of the ad network (202), queries a core service (206) on the mobile device to see if a revenue-split agreement exists (208) for a given app. If an agreement exists (210) a flag is set on the mobile device with the distributor (OEM) identity. If no agreement exists, the flag is set to null (214).

Upon the request for an ad (216), the flag is sent back to the ad network (216). Based on the flag, e.g., credentials within and agreement details thereof, the ad network accounts for the revenue split between the OEM and the app developer (218). A change in the relationship between OEM and app developer causes the ad network to simply redirect the distributor ID for that app to the appropriate party.

As mentioned before, a malicious OEM can try to erase and re-download an app that was downloaded outside the OEM channel to get paid for an app that was not originally distributed through the OEM. Similarly, a malicious app developer may erase and re-download an app to try to move out of the OEM distribution channel, to avoid paying the OEM. To guard against such maleficent actions, and to ensure trustworthy and consistent revenue split, when an app updates, the ad unit retains and recalls the most recently updated relationship between OEM and app developer. Additionally, the ad network implements a server-side control wherein the ad-ID of the user is mapped against the OEM-app-developer relationship. For example, the ad network may tie the ad-id of the user to the app in the form of a table, thus establishing control over the revenue split procedure. The ad network can also detect how long ago an app was erased, e.g., by detecting the deletion date, and apply previously agreed-upon rules, e.g., after seven days, the re-install can be claimed by a new channel. Such operations by the ad network are performed with user permission, and do not make use of identifiable information regarding the user or user device.

Additionally, the techniques of this disclosure can be implemented in a product that coordinate multiple ad networks. For example, ads from different networks can be distributed via such a product, and the revenue-split between OEM and app-developer can be administered on behalf of the different ad networks.

CONCLUSION

The techniques of this disclosure enable OEMs and app developers to reach agreements for the OEM to distribute an app for low or no upfront fee in exchange of a percentage of future ad revenue generated by the app. Total ad revenue generated is often split between an ad network and an app developer, and in this case the OEM can either (i) take a share of the ad revenue portion typically provided to the app developer, (ii) take a share of the ad revenue portion due to the ad network, (iii) take a share of both, or (iv) play the role of ad network or app developer while also taking a share of the other party's portion. The techniques ensure that the agreements are enforceable and protect both the OEM and the app developer against fraud or malicious action.