

# [EXTERNAL EXPORT] Banner Refresh Policy

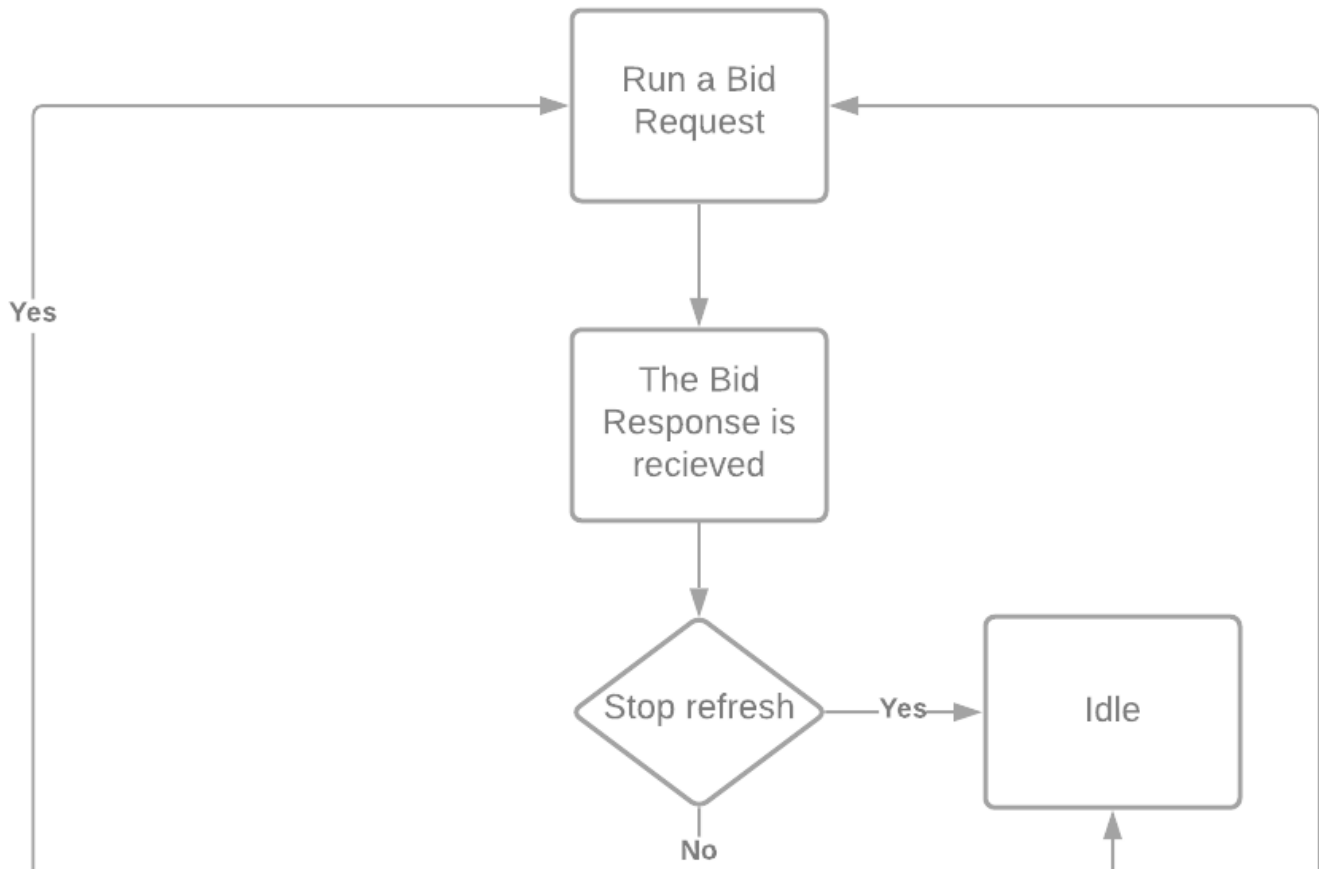
As usual, the banner ads should be refreshed in some period of time. At the first glance, it looks obvious but there are a lot of circumstances that can influence this process. This document regulates the policy of refreshing the banner ad respectively to the different possible situations.

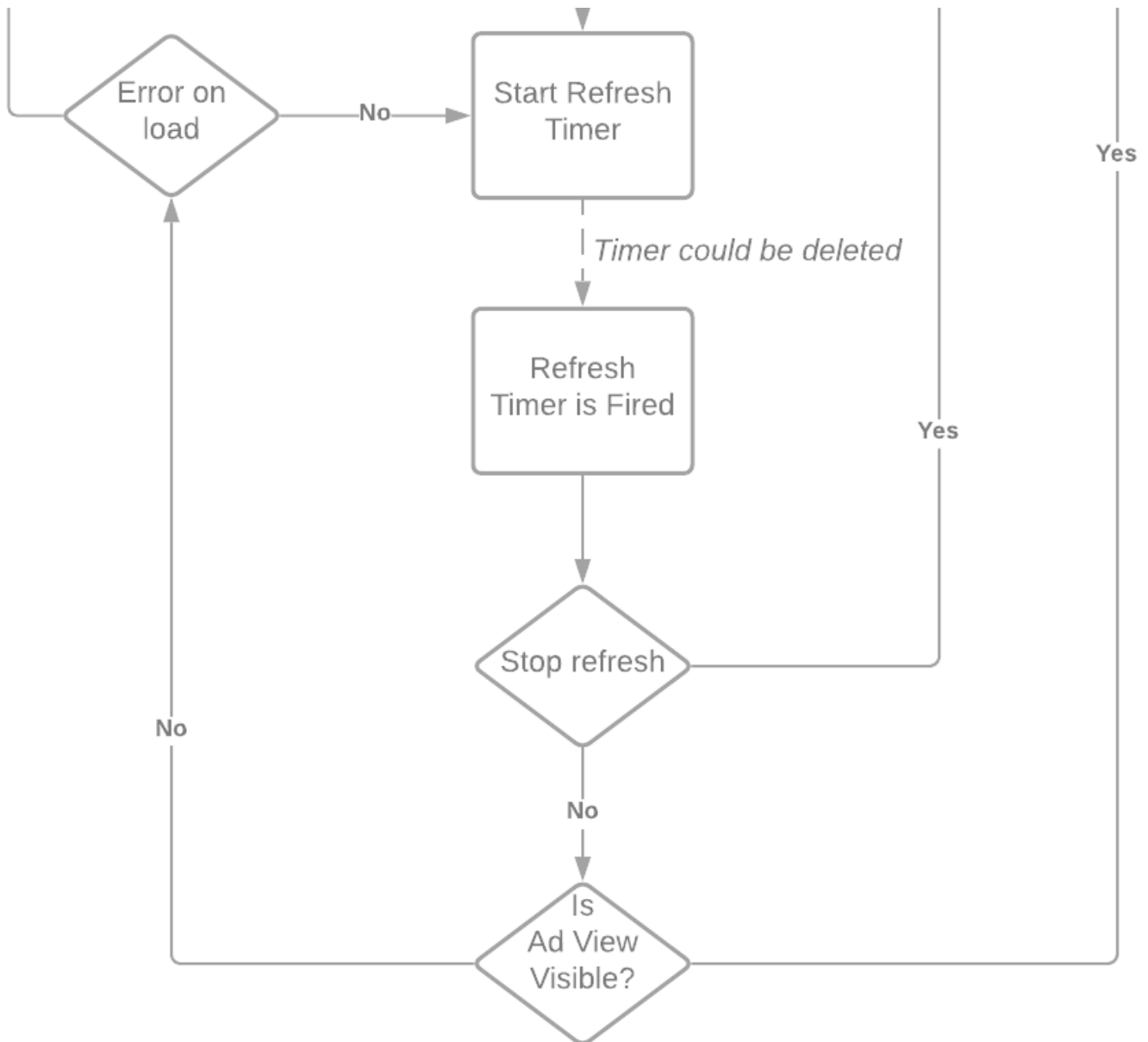
## General Rules

The rules:

1. The banner should be refreshed at a certain period of time.
2. The default refresh period is **60 sec**.
3. Publisher can change the refresh rate via the Banner API
  - a. the minimal refresh rate is **15 sec**
  - b. the maximum refresh rate is **120 sec**
  - c. If the publisher set the value which is out of range, the SDK logs a warning and changes the refresh rate to the *min* or *max* value respectively to the direction of change
4. The refresh timer starts when the bid response is received, despite the response status.
5. The refresh timer should not initiate the bid request if the ad view is not present on the screen. In this case, the timer should be started again and check the viewability on the next iteration. The circumstances which lead to skipping the refresh after firing the timer are:
  - a. The banner was scrolled out of the screen
  - b. The user changed the tab or switched the carousel page and the banner with the entire page was removed from the screen
  - c. The user tapped on the banner, and the modal view or external browser is presented on the screen covering the banner
6. Here is the list of cases when the Ad View is "not visible" but still should be refreshed:
  - a. The Ad View is shrunk, f.e, after an error
  - b. A Container for the Ad View is shrunk, f.e, after an error
  - c. The Ad View has been withdrawn from the view hierarchy due to the custom orchestration of the ad views by publishers code
7. The publisher should be able to:
  - a. Stop refresh
  - b. Resume refresh and loading via the loadAd() method

The flow:





### Pitfalls and corner cases

The controversial points in the above algorithm are:

- The ad state. It is hard to define the state for all integration kinds in the same way.
  - For the **MoPub** we don't have the ability to track the ad state, so it is hard to say if the ad was loaded on the previous iteration or failed.
- The publisher's code. The app could change the view hierarchy due to the status of an ad request

To handle these cases properly we should give publishers an opportunity to tune the refreshing flow by:

- For MoPub integration Inform the SDK that the ad was not loaded due to the error. It will help to understand the current state of the ad unit.
- Stop/Resume refreshing. It will help publishers to adapt the banner's behavior to the app's logic. The *resume* should lead to the new cycle of the algorithm - requesting the ad so publishers should use the `loadAd()` / `fetchDemand()` method.

### Policies of third-party SDK's

Depending on the testing time the behaviour of third party SDKs may be different but for now, I see the next one (iOS):

- **Prebid** - prebid SDK does not check any visibility params, it reloads the ads according to the refresh interval. Primary ad server SDK loads the ad respectively (even if ad view is invisible and no impression on the last iteration).
- **MoPub** - does not load the next ad until the impression of the previous one is fired. If the ad is not visible on the screen (on a different tab or scrolled out) the impressions will not be tracked. Hence there won't be any loading.
- **GAM** - It is hard to describe the GAM's behaviour precisely. We do not know their protocol for requesting ads and other events. Basing on the SDK delegates I see that they stop the reloading when the banner is out of the display. The delegate methods for confirming that ad is loaded are not called when the ad is not visible.