# PRE-CACHING

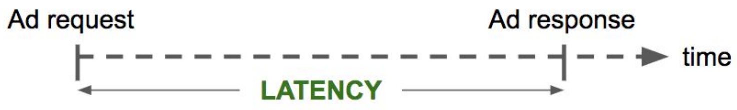
## WHAT IS PRE-CACHING?

Pre-caching allows the ad to be loaded in the background of the application and shown to the user once the app calls for the ad to be displayed. Pre-caching allows for an ad to be loaded off-screen as part of an application’s buffer to help provide a smooth user experience. Pre-caching is always recommended. Loading an ad and not showing it does not have any negative impact.

Interstitial and rewarded video ad formats provide caching methods. We can’t use pre-cache with banners.

## WHY PRE-CACHING? WATERFALL LATENCY

Each item in your waterfall is a call to an ad source. If no ad is returned by the ad source, the next item in your waterfall is chosen. For each ad source, the mediation has to make a call internally or to a partner to retrieve ads, which takes time and increases the duration between when the ad request is initiated and the ad response is returned. This duration is called **latency**.



From a monetisation perspective latency becomes a problem when we lose the opportunity to show an ad because we are still requesting an ad and going through the waterfall at the moment that we should have been displaying the ad.

From a user experience perspective latency becomes a problem when the user engages with a rewarded video placement and there are no ads available to be displayed because we are still requesting an ad and going through the waterfall.

## PRE-CACHING BEST PRACTICES

- Pre-cache the first ad at the app start (make sure to wait until the initialisation of the SDK to make the request - we recommend you to delay the first request 100ms after initialisation is fully completed;

- Request the next ad after the previous one is displayed and cache it. As long as we have an ad cached no new request should be done.

- Request a new ad only when the previous one closed. When you load a new ad (no matter if it's right after receiving close, failure or any other callback), delay the firing of the next request by 100ms.