

[EXTERNAL EXPORT] Components

- Integration Scenarios
 - Banner
 - Native
 - Native Event Trackers
 - Interstitial
 - Rewarded
- Internal Modules
 - Bid Response
- Configuration and Targeting
 - Ad Unit Configuration
 - Native Ad Unit Configuration

Integration Scenarios

SDK supports the integration with Google Ad Manager (**GAM**), **MoPub**, and pure In-App bidding scenario just with the **Prebid Server** server. Integration scenarios have different APIs but at the same time, they work based on common components.

Integration with **MoPub** is based on the Mediation feature. The ad views are instantiated via mediation adapters when the Prebid Network line item won in the waterfall.

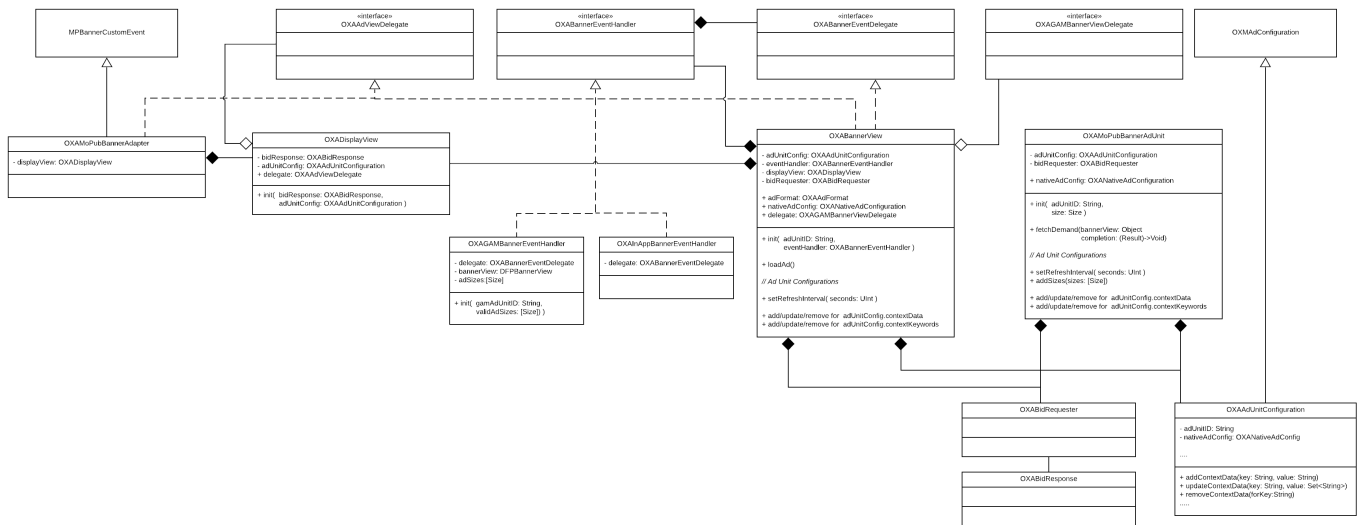
Integration with **GAM** is based on the App Events feature, and SDK manages the ad views in the UI. The ad views are instantiated when the Prebid line item won in the waterfall and GAM returned a meta-information about the necessity of rendering the cached creative via the app event. Otherwise, if another line item won, SDK will display an instant MA SDK Ad View.

In-App Bidding Integration works as a regular advertising SDK. Publisher integrates the Ad View or Interstitial Controller into the app's UI and this view renders ads from the winning bid.

SDK supports these APIs to support particular ad kinds:

- Banner - the API for the inventory which is able to show **display ads**, **outstream video ads**, **native style ads**.
- Native - the API for the inventory which could display *Unified Native Ads*, *Native Templates*.
- Interstitial - the API for the inventory which is able to display **HTML** or **Video** interstitials
- Rewarded Video - the API for **Rewarded Video** ads

Banner



Native

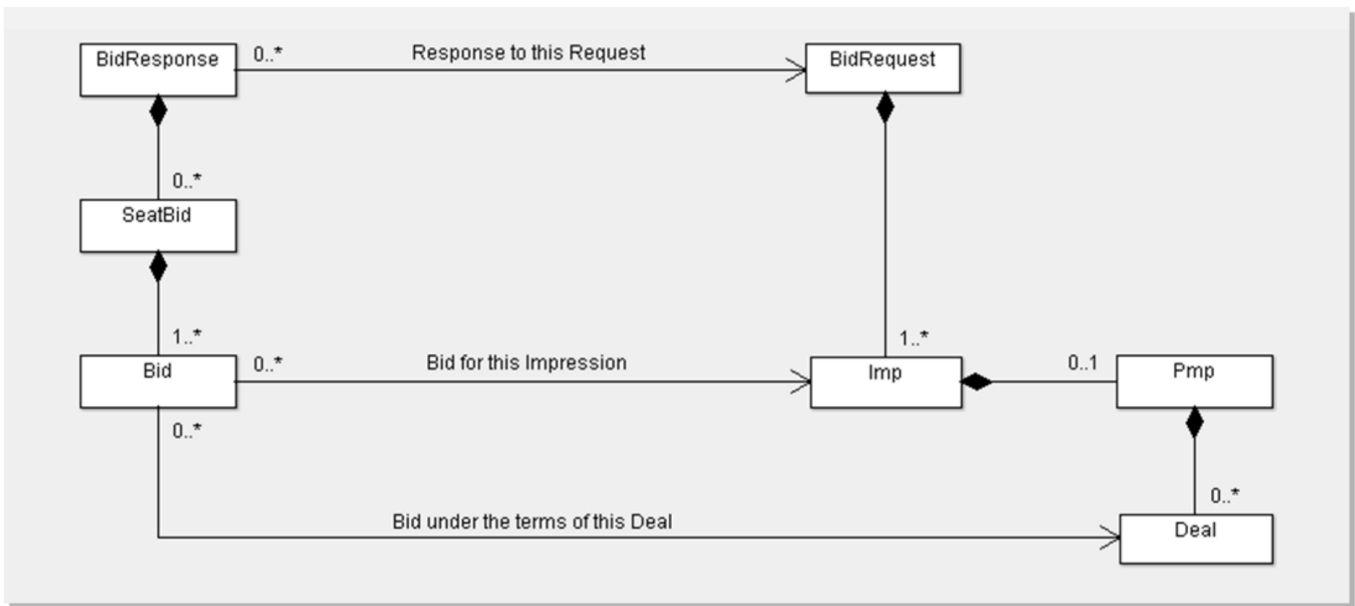


Figure 4: Bid Response object model.

The appliance of received response depends on the integration kind:

- GAM - set up keywords into the ad unit. Store the creative in the cache.
- MoPub - set up keywords into the ad unit. Pass creative to the adapter via extras.
- In-App - render the ad.

In any case, there is no reason to deserialize the bid response partially and cache the raw data.

Configuration and Targeting

Ad Unit Configuration

All properties, which a publisher could assign to the ad unit, are transformed to the particular OpenRTB params in the bid requests. There are obligatory, optional, and ad unit-specific properties for prebid ad units. The chart below shows the set of supported properties and which ad unit they belong to. This table defines the interface for customizable properties of ad units.

	Banner	HTML Interstitial	Video Interstitial	Rewarded Video	Outram Video	Native Styles	Native
configID	Obligatory	Obligatory	Obligatory	Obligatory	Obligatory	Obligatory	Unsupported Ad Format
adUnitSizes	Obligatory	Not Used	Not Used	Not Used	Not Used	Not Used	Unsupported Ad Format
minWidthPerc	Not Used	Optional	Not Used	Not Used	Not Used	Not Used	Unsupported Ad Format
minHeightPrec	Not Used	Optional	Not Used	Not Used	Not Used	Not Used	Unsupported Ad Format
nativeAdConfig	Not Used	Not Used	Not Used	Not Used	Not Used	Obligatory	Obligatory
contextData	Optional	Optional	Optional	Optional	Optional	Not Used	Unsupported Ad Format
contextKeyword	Optional	Optional	Optional	Optional	Optional	Not Used	Unsupported Ad Format

Native Ad Unit Configuration

Native ads assume a special set of configuration properties according to the IAB standards. In order to keep API as clean as possible, these configs are assembled in a separate class - NativeAdConfig, which could be set up by the publisher into a particular Ad Unit and get held in the AdUnitConfig.

Looking at this property bid requestor can build and perform native ad requests.

The configurable parameters are defined by [IAB specs](#) (4.1 Native Markup Request Object).