# Open Measurement SDK: Native App Capabilities and Limitations

The Open Measurement software development kit (OM SDK) enables third-party ad measurement services to collect signals regarding ad impressions and performance. These signals are sent by the Open Measurement Interface Definition (OMID) API.

Components: The OM SDK includes both native libraries and JavaScript.

- OM SDK Native Libraries: platform specific libraries for iOS, tvOS,Android or AndroidTV. Implemented natively on the device for collecting and publishing viewability signals that support viewable impressions as defined by MRC.

- OM SDK JavaScript: JavaScript libraries used to implement OMID API. Includes details for building an adapter that listens to signals provided by the OM SDK. Using the provided JS libraries provided in the OM SDK, the OMID API should be universal across iOS, Android, and Web browsers. Additional platform support is planned for future iterations.
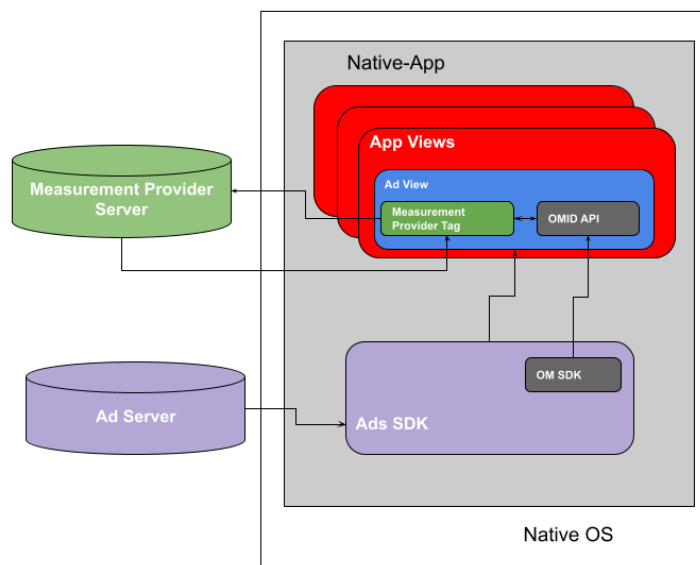
Video and Platform Support

OM SDK video is supported in Nativepp, whether it's delivered in a video player or as part of the creative for a display ad. OM SDK can be delivered and implemented using the IAB Video Ad-Serving Template (VAST), but VAST is only designed to serve video content in the context of a video platform.

Video and platform technology supported in OM SDK is described in the following table.

Technology	Native App Display	Native App Video
Webview	✔	✔
Native	✔	✔
iOS/tvOS	✔	✔
Android/AndroidTV	✔	✔
VAST 2.0	Not Supported	✔
VAST 3.0	Not Supported	✔
VAST 4.0	Not Supported	✔
VAST 4.1	Not Supported	✔
VPAID	Not Supported	Not Supported

For full compatibility mapping between target platform and OM SDK please see:

https://iabtechlab.com/standards/open-measurement-sdk/native-app-compatibility-mapping
https://iabtechlab.com/standards/open-measurement-sdk/web-video-compatibility-mapping

CTV Display Connection Status

It's critical for CTV measurement to understand if, when an application is running, the display screen is on.

- A new value indicating that there is no display connection will be added to the Reason enum.
- When the OM SDK determines that there is **no display connection**, this value will be added to the list of reasons provided in the event data of ad view-related events.
- In this case, the pixelsInView and percentageInView values will be zeroed out like they are when the app is backgrounded.

To make this no display connection determination OMSDK uses similarly exposed attributes from supported operating systems about the Audio route for HDMI.

[Android](#)
[tvOS](#)

OM SDK Limitations

While OM SDK facilitates features like brand safety and fraud detection, logic for execution is not built-in. The limited support for these features is listed below.

- **Brand Safety:** Performed by Measurement Provider tag. No logic in OM SDK.
- **Fraud Detection:** Performed by Measurement Provider tag. No logic in OM SDK.
- **Advertising ID:** No retrieval logic using OM SDK.

MEASUREMENTS REPORTED (OM SDK v.1.4)

NOTE: Full API Documentation available at: https://tools.iabtechlab.com. Specific API calls may vary by implementation.

Ad Session information

Measurements reported for the ad session identify details for session start, finish, and any errors that occur. Items marked with an asterisks (*) are new in version 1.4.

- **sessionStart:** apiVersion, environment, accessMode, videoElement, slotElement, adSessionType, adServingId, transactionId, podSequence, adCount, omidNativeinfo - partnerName, partnerVersion; omidJSInfo - omidImplementer, serviceVersion, sessionClientVersion, partnerName, partnerVersion; app - libraryVersion, appID, deviceInfo- deviceType, os, osVersion; supports, customReferenceData, creativeType, mediaType, impressionType, supportsLoadedEvent, contentURL, lastActivity*
- **sessionError:** errorType, message
- **sessionFinish**
- **loaded:** creativeType, mediaType, impressionType
- **impression:** creativeType, mediaType, impressionType, mediaEventAdaptorType, mediaEventAdaptorVersion, viewport width and height, adView,

Impression Events

OM SDK 1.3 added full support of the MRC Begin to Render definition for impressions. Verification scripts can receive all the data needed to determine the begin-to-render criteria using the updated `loaded` event. In addition, new parameters for 'impression type' enables the integration partner to declare how the impression event was triggered. With this improved feature, integration partners and verification vendors can better understand and manage discrepancies.

These impression types are described below:

Impression Type	Description
loaded	Integration is using count-on-download criteria
beginToRender	Integration is using begin-to-render criteria
onePixel	Integration is using one-pixel criteria, which is when at least one pixel of the creative is visible.
viewable	Integration is using viewable criteria, which is 1 second for display or 2 seconds while playing for video.
audible	Integration is using audible criteria, which is 2 continuous seconds of media playback with non-zero volume. Only for use with `audio` creative type.
other	Integration criteria not met on any of the defined impression types.
unspecified	No criteria declared and therefore no impression type can be defined. Default impression type for OMID 1.2 and integrations not declaring impression type.

User Interactions

Measurements reported for user interactions include metrics for direct interactions, like clicks, and view dimension changes.

- **adUserInteraction:** click, invitationAccept
- **geometryChange:** percentageInView, geometry,onScreenGeometry, measuringElement containerGeometry, onScreenContainerGeometry, viewport width and height, adView, reasons, pixel, friendlyObstructions, declaredFriendlyObstructions

Video and Audio Ad Lifecycle

Measurements reported for the video or audio ad lifecycle include metrics track media player and ad events.

- **Media Player State:** Minimized, Collapsed, Normal, Expanded, Fullscreen
- **Media Interaction Types:** Click, Invitation Accepted
- **Media Events:** start, first Quartile, midpoint, thirdQuartle, complete, pause, resume, bufferStart, bufferFinish, skipped, volumeChange, playerStateChange, adUserInteraction