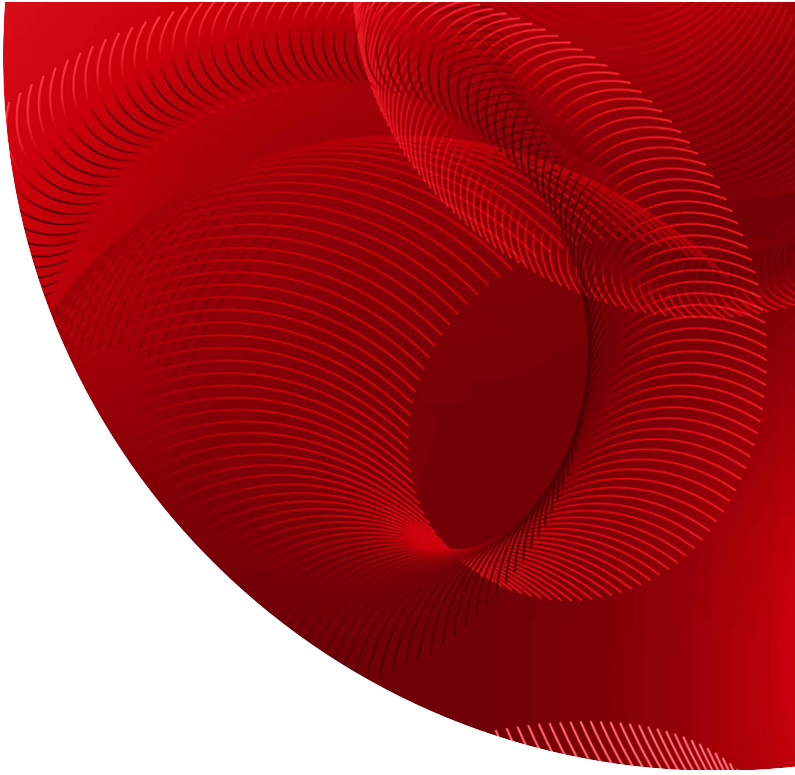

OTT/CTV User Agent Preliminary Guidelines

December 2019

Background

Accurate device information – type, operating system (OS), client software, and other data – is crucial in the advertising ecosystem. It allows for precise targeting, better creative delivery, analytics, and measurement among other things. Knowing the screen size and other capabilities enables the best creative to be sent to the device. Reliable device identification in connected TV (CTV) / over-the-top (OTT) is especially important for measurement because TV screen inventory is considered more valuable than inventory on other devices. Today, according to our IAB Tech Lab members, a large number of CTV impressions are incorrectly reported as not CTV, making buyers question the campaign validity and sometimes denying publishers of proper credit for their inventory.

The primary reason behind this problem is that there is no reliable technology to identify the device type independently. The only standard facility that transports device and client software information is User-Agent HTTP request header. Currently, ad serving participants have no choice but to depend on the User-Agent header for the device determination solutions. However, the header value, its content, accuracy, and, at times, even availability is at the mercy of the device manufacturers, app developers, and software vendors.

In this document, we will go over some of the common issues the OTT Technical Working Group members have seen, and also propose some recommendations on how to address them. However, we are calling these "preliminary guidelines" since we believe that a deeper conversation with the device platforms and vendors needs to occur.

Problematic User Agent Strings:

The following are some of the common problematic User Agent strings seen from various OTT device platforms.

1. **Non descriptive user agents**

   The user agent string provided does not contain enough information to determine the device.

   For example, on the Xbox the user agent string returned is *"Mozilla/4.0 (compatible; NativeHost)"*

2. **User agents from built-in SDKs**

   Sometimes the device surfaces a user agent defined by an underlying SDK.

   For example, on some Vizio devices leveraging the Chromecast Built In SDK, the user agent string returned is: *"Mozilla/5.0 (X11; Linux armv7l) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.87 Safari/537.36 CrKey/1.34.149937"*

© 2019 IAB Technology Laboratory, Inc

3. User agents shared across different device types for an OS platform

In cases where the OS platform is shared by multiple devices types, the user agent sometimes lacks the information needed to distinguish between devices.

For example, user agents generated in Apple app environments, regardless of whether it is an iPhone, iPad or Apple TV, the user agent string returned is: *"[appname]/5 CFNetwork/975.0.3 Darwin/18.2.0"*

4. User agents lacking device identifiers

Examples:

- The exoplayer library returns the UA *"<app name>/<app version> (Linux;Android <android version>) ExoPlayerLib/<exoplayer version>"*
- The Samsung Smart-TV / Tizen devices return *"samsung-agent/1.1"*
- The Comcast X1 Box returns the default WebPlatformEmbedded (WPE) user agent *"Mozilla/5.0 (Linux; x86_64 GNU/Linux) AppleWebKit/601.1 (KHTML, like Gecko) Version/8.0 Safari/601.1 WPE"*

5. Model Collisions

Sometimes a model number is included but no identification of the brand. With this, the risk of collisions occurs where different brands utilize the same model numbers.

For example, there are more than a dozen brands with a model *"M3"*.

6. Other issues

Our members have seen a number of other issues that might not directly impact device detection, but unnecessarily complicate User-Agent string parsing. For example:

- Many User-Agent headers include irrelevant information (not related to product version)
- Many User-Agent headers have encoding issues such as spaces being replaced by a '+' symbol, or characters getting encoded too early (before being inserted into an HTTP request).

© 2019 IAB Technology Laboratory, Inc

Solving the User Agent problem

Addressing these user agent problems is not easy. The ideal solution is for device platforms to provide a well-structured and usable user agent (as described in #1 below), which the publisher cannot change. This increases the level of trust regarding the validity of the user agent in systems that consume the information. While standardization will take some time to develop, we understand that having some guidance in place now is better than none. So, we offer the following three recommendations to improve current methods for identifying user agents.

1. **Recommendation for CTV device platforms and app developers/publishers:**

   Provide enough details in the User-Agent header to allow it to be consistently differentiated from the user agent of other devices. Using the following pattern to build the user agent will offer a consistent structure for all parties who consume the details:

   <device info> <os name>/<os version> <app name>/<app version> <other info>

   For example:

   DeviceBrand DeviceModel OSName/1.2.3 AppName/1.2.3 LibName/1.2.3

   We also recommend that platforms be conservative in adding unnecessary information to the User Agent string, and also in encoding practices.

   Finally, we recommend that platforms submit their User-Agent header value to the IAB Spiders and Bots white list so that it is not considered a bot, and can be a signal used to determine the device information.

   Even with the above guidance, we are aware that this recommendation might not work for all device platforms. To that end, we invite all the OTT device platforms to discuss the challenges with the IAB Tech Lab so that we can develop a solution that best addresses the concerns of the advertising ecosystem.

2. **Recommendation for device detection vendors:**

   One of the challenges is the inconsistent device types returned across various device detection vendors. Ask vendors to use the same mechanism to provide the user agent. One option is the use of the IAB Tech Lab's Spiders and Bots white list as a starting point, but other options are available. We invite suggestions for further discussion.

© 2019 IAB Technology Laboratory, Inc

3. **Recommendation for alternate or supporting signals:**

- On device platforms where user agent values are not consistently provided, publishers should attempt to provide the UA themselves, following the guidance provided above. When VAST is used, the "DEVICEUA" macro should be populated correctly.

- Exchanges should be consistent with the user agent values used between bid requests and tracking pixels so that classification is consistent between parties to the transaction – i.e. Demand-side platform (DSP), third-party ad server, verification vendor, etc.

- DSPs can consume the "type" field from the device object of the OpenRTB bid request to use as a hint when conventional user-agent string parsing fails to return a result.

As more parties follow these recommendations, the quality of OTT inventory will improve as will its value. Success is limited by adoption, so we are calling on the OTT advertising ecosystem to follow these recommendations as soon as possible. The OTT Technical working group is looking into additional recommendation (like updates to the AdCOM object model to better support video/OTT inventory). We welcome everyone to join our working group and add to these conversations.

***Note** – we are aware that the terms OTT and CTV have been used interchangeably even though they have differences and overlaps. The guidance above is intended for all internet connected devices that are connected to (or part of) a television set - and not intended for mobile and desktop inventory.*

© 2019 IAB Technology Laboratory, Inc