Introduction To Prebid

This document provides an overview of Prebid, an open-source header bidding library.

Ad Tech Terms

Before we discuss Prebid, it is helpful to understand some of the terminology associated with the ad tech process.

Header Bidding:

Header bidding is a programmatic process for conducting an auction to fill ad slots on a publisher's website.

Publishers embed code in their web pages' header section that connects to an ad server. This code, known as an ad tag, requests an ad from an ad server and places it on a web page. The most popular ad tag is Google's Ad Manager (GAM).

Ad Server:

An ad server is a platform that stores, manages, and delivers ads to websites.

Ad Slot:

The ad slot is the space on a web page displaying an ad.

Programmatic:

Programmatic advertising automates buying and selling ad space. It uses technology and tools to automatically serve the right ads to the right segments.

The Ad Slot Auction

If an ad slot on a web page is not sold directly to an advertiser, the publisher typically auctions it off using embedded header bidding tags and an ad server. This programmatic auction occurs while the web page is loading in the user's browser.

The ad slot auction process works like so:

1. When a user visits a website, like the New York Times, the embedded code sends an ad request to the ad server as the web page loads.
2. The ad server recognizes the ad slots on the web page and conducts an auction based on targeting information in the Line Item section of the Insertion Order.

3. When the auction is complete, the ad server returns the winning ad's assets and code,
   collectively called the Creative, to the ad tag.
4. The ad tag displays the Creative on the web page in the correct ad slot.

The Development of Prebid

While this system benefits ad exchanges, it limits publishers' potential revenue. Many potential ad buyers were not included in the ad slot auction if they were not part of Google Ad Manager's (GAM) ecosystem.

Ad tech companies found that by pausing the call to the ad server, they could submit their own bids to the GAM's auction process. This pause and submission action could be conducted by Javascript (the ad tag) embedded in the header section of a page's HTML. This is why the process is known as header bidding.

Instead of competing against each other in creating individual header bidding libraries, the major ad tech firms, such as The Trade Desk, Appnexus, and Rubicon, collaborated to build an open-source platform that anyone could contribute to and use. This platform, known as Prebid, offers several advantages over other solutions, including increased participation, improved efficiency, and enhanced transparency.

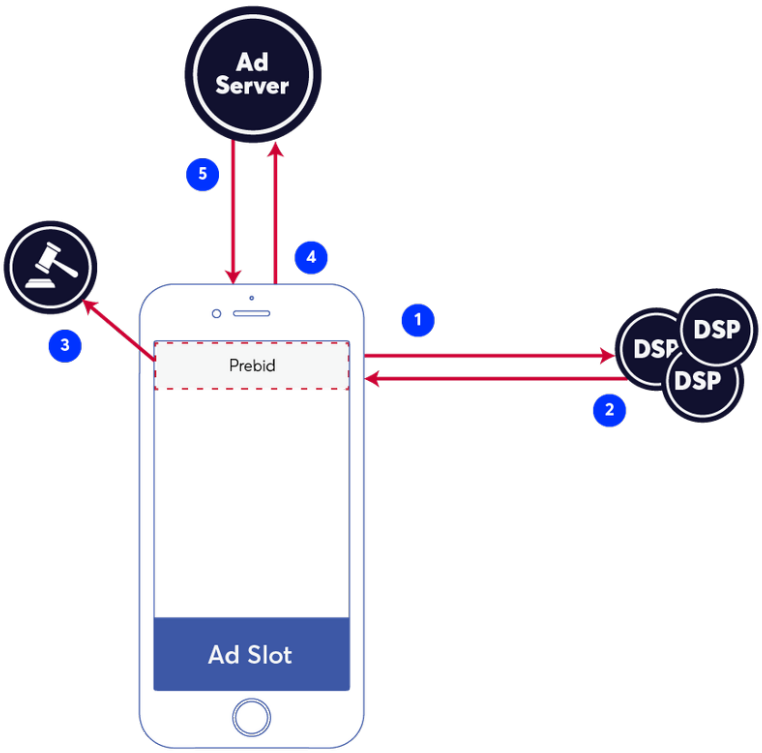
Installing Prebid

To install Prebid, publishers download and enhance the core code package with selected adapters and modules, tailoring the installation to their specific needs and preferences.

- Adapters are snippets of code written by Demand Side Partners (DSP). These third parties help manage ad agencies and individual advertisers' campaigns and enter bids into the ad server auctions on their behalf. The adapters connect Prebid and the DSPs, allowing them to communicate and exchange information.
- Modules are code written by Prebid to extend the library's functionality. They provide additional features and capabilities that enhance the library's performance and effectiveness.

Once downloaded, publishers can store the Prebid code and link it to the HTML pages of their website where they want to display advertising.

How Prebid Functions



The flow chart below illustrates the life of an ad call for Prebid.

1. The publisher's page is at the heart of the ad call process. As it loads, it initiates the ad call by sending ad requests to its associated Demand Side Partners (DSPs).
2. Upon receiving the ad request, the DSP plays a pivotal role. It processes the bid information provided by the publisher, such as ad slot dimension and costs, and, based on its clients' campaign goals, determines which ads to return as bids to the page.
3. Prebid (or another ad tag) receives the bids from the DSPs and conducts an auction to determine the best bid(s) to send to the publisher's ad server (typically, this is Google).
4. The ad server receives the bid(s) and adds them to its internal auction.

5. If the winning bid was submitted by Prebid, the ad server sends a signal back to the Prebid ad tag. The ad tag proceeds to render the ad to the page and begins to collect user event data.

This flow is a simplified overview of the Prebid process. The process is much more complex and involves several challenges. These include:

- Managing privacy consent and data security.
- Ensuring compliance with standards established by the International Advertising Bureau (IAB) and government regulations like GDPR.
- Dealing with potential issues such as latency and ad fraud.

Understanding and addressing these complexities is crucial for Prebid's successful implementation and operation.

Prebid Data

Each ad bid request and response generates valuable and actionable data for both the publisher and the advertiser.

Prebid provides a robust API to view and analyze this data. For example, developers can call the API's `getEvents` method to view all events since the web page loaded:

```
pbjs.getEvents().forEach(event => {
  console.log("event: "+event.eventType)
});
```

Some examples of events are:

- `requestBids`: Bids have been requested from adapters.
- `bidResponse`: A bid response has arrived.
- `auctionEnd`: The ad auction has ended.
- `bidWon`: A bid has won.

Next Steps

- Visit docs.prebid.org to learn more about Prebid.
- Contact Tech Comms to set up a learning session on Prebid.