

Technical Disclosure Commons

Defensive Publications Series

November 13, 2017

A Cloud-Based Solution for Deploying Digital Advertising Bidding and Creative Selection Models Based on First Party Data

Nir Margalit

Don Greenberg

Stanislav Belov

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation

Margalit, Nir; Greenberg, Don; and Belov, Stanislav, "A Cloud-Based Solution for Deploying Digital Advertising Bidding and Creative Selection Models Based on First Party Data", Technical Disclosure Commons, (November 13, 2017)
http://www.tdcommons.org/dpubs_series/807



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

A CLOUD-BASED SOLUTION FOR DEPLOYING DIGITAL ADVERTISING BIDDING AND CREATIVE SELECTION MODELS BASED ON FIRST PARTY DATA

There is a major conflict of incentives between Demand Side Platforms (DSP) and data-rich advertisers. These advertisers want to deploy proprietary data, models, and insights in display advertising that give them as great a competitive advantage as possible, while DSPs want to build bidding solutions which are generic enough to be useful to as broad a group of advertisers as possible. As a result, advertisers who do not fit the data-light, vanilla-brand or vanilla-direct response use cases are generally underserved by DSPs. Therefore, they have to either hand-tune their bidding by targeting segment, with little to no machine learning, or accept a black box optimization solution with no room for control or competitive insight. In addition, sophisticated advertisers can't use most of what they know about a user in display advertising, largely due to concerns with sharing their valuable data with DSPs whose data and security practices are generally less stringent than the advertisers. Even more, a DSP in possession of such valuable data may succumb to the temptation of reusing the data or models to bid more effectively for other advertisers, including competitors.

Cloud Bidding to Enhance Display Advertising

Cloud bidding is a bidding solution for DSPs, agencies, and technical advertisers, providing all the advantages of owning a bidder, such as deploying first party data and models for bidding and creative selection, with none of the infrastructure overhead or data security concerns. The advertiser only focuses on the data and models, while cloud

bidding takes care of bid request ingestion, targeting, impression tracking, budgeting, pacing, creative filtering, logging, reporting, billing and more.

A Multi-Tenant Bidder (MTB) is the core of the cloud bidding platform offering a technology stack which enables lean bidding by hosting many “tenant” bidders. The MTB receives and ingests callouts from a number of advertising exchanges (e.g., SSPs), applies targeting, budgeting, and pacing, and requests bids by invoking custom bidding functions (campaign endpoints) for each of the eligible advertiser tenants.

Cloud bidding provides an advertiser with various advantages over conventional systems. For one, cloud bidding provides advertisers with a secure system to deploy their own models using proprietary data, which significantly increases the effectiveness of their display advertising. A cloud bidding system offers higher thinktime (e.g., 100ms) than any other full-control bidding solution on the market today. A cloud bidding system is also instantly scalable. That is, it allows an advertiser to build a system that can handle one bid request and then scaling seamlessly to millions.

The cloud bidding platform consists of the tenant bidding interface, several public APIs, logging and reporting via a cloud-based analytics platform, and a light-weight configuration User-Interface. Basic features of a cloud bidding system may be as follows:

1. Tenant sets up campaigns and creatives through the Cloud Bidding API, specifying for each campaign the targeting criteria for which they want to receive bid requests, the campaign bidding endpoint, and the eligible creatives.

2. The cloud bidding platform receives a request from a Supply-Side Platform (SSP), and checks which campaigns match the targeting criteria for the request. Cloud bidding calls the tenant endpoints for the matched campaigns with an OpenRTB bid request, including the `campaign_id`, eligible creatives, and other supplemental information (e.g., budgets, frequencies, etc.). If the request is eligible for multiple sizes or formats, Cloud bidding will submit multiple requests for each group of eligible creatives (although the advertiser can opt out of this feature).
3. The tenant's endpoint ingests the request, and selects a creative and bid using proprietary data or control flow of their choice. The bidder can call services outside of the cloud platform, as long as they can return their bid in under the timeout (e.g., ideally 100ms).
4. Cloud bidding collects bid responses and selects the highest bid to submit into the SSP auction.

Example Cloud Bidding System

Fig. 1 illustrates a cloud bidding platform system that executes a process for deploying digital advertising bidding and creative selection models based on first party data. The central component of the cloud bidding platform is a Multi-Tenant Bidder (MTB) that includes, for example, a Log Joiner, a Budget/Pacing Service, a Campaign Authenticator, a Campaign Extractor, a Multi-Tenant Bidding Service, a Campaign API Server, and a Campaign Database. The MTB communicates with a Click Server, an Ad Exchange Real-Time-Bidder (RTB), a Cloud Platform, and a Tenant Cloud Bidder.

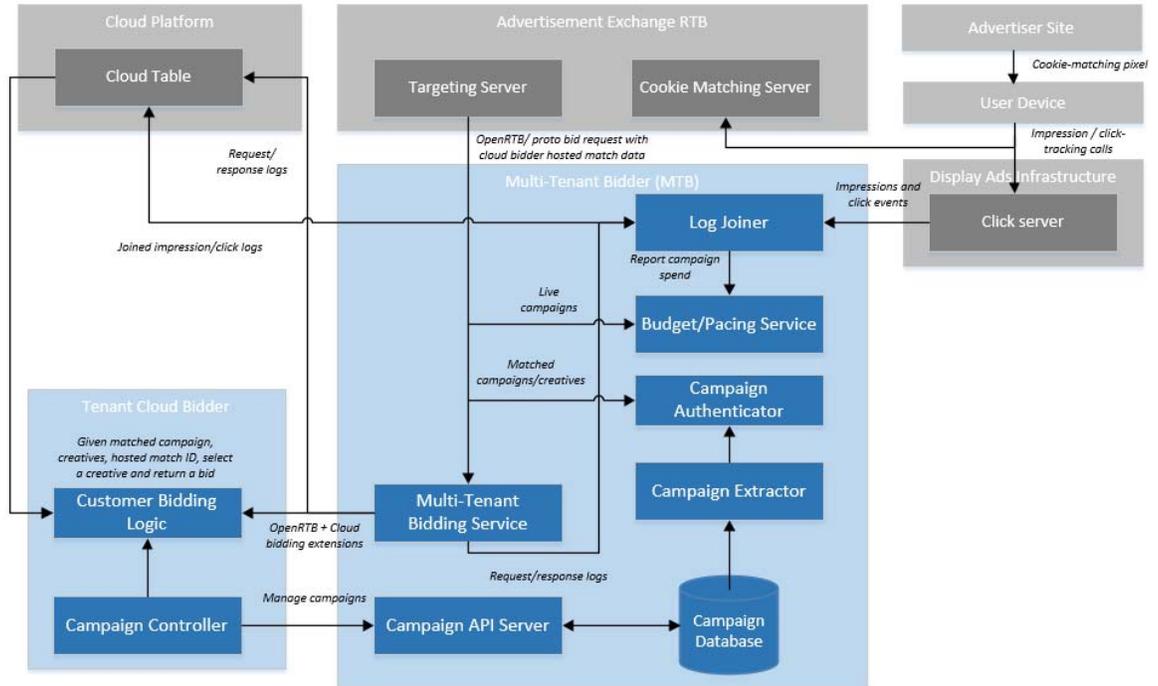


Fig. 1 - A cloud-bidding platform system including an MTB

In one implementation, the cloud bidding platform system has a number of phases that can run in parallel. Before a user device is served an advertisement, the system performs a cookie matching process. For example, a user device visiting an advertiser’s site will receive a redirect request from the advertiser’s site that causes the user device to send an HTTP request to the Cookie Matching Server of the Advertisement Exchange RTB. The Cookie Matching Server records the advertiser’s cookie in a table that maps the user ID of the Ad Exchange RTB with the user ID of the advertiser’s site for the user device. After the system serves the advertisement to the user device, the Click Server starts to record the user device’s interaction (e.g., clicks and impressions) with the served advertisement.

The Targeting Server delivers a bid request to the Multi-Tenant Bidding Service to initiate the cloud bidding service for injecting the single-bid request and calling the

individual advertising bidding function. In this case, the Targeting Server relies on the Cookie Matching Server to accurately populate the anonymous IDs that the Targeting Server sends with its bid request. Both the Multi-Tenant Bidding Service and the Click Server feed into a Log Joiner that stores all impressions and clicks. The system aggregates the stored data for populating real-time budgeting and pacing information used by the Budgeting / Pacing Service to check for budget-related triggers (e.g., exceeding budget, frequency capping of ads).

The Campaign Authenticator functions as a targeting match system that checks whether an advertiser's campaign on cloud bidding is eligible for that bid request. For example, the Campaign Authenticator may check whether the advertiser has a creative ID that is approved for this particular bid request or whether the advertiser has indicated that they want this type of bid request. The Campaign Authenticator receives a match request from the Multi-Tenant Bidding Service. In response, the Campaign Authenticator sends the list of campaigns across all advertisers that are matched to the Multi-Tenant Bidding Service, which then calls the cloud function free to the advertisers.

The Campaign Controller is a command line interface that the advertiser uses to create campaigns and associate creative IDs and cloud functions with the campaigns. The Campaign Controller plugs directly into a Campaign API Server, which stores the campaigns that the advertiser creates into the Campaign Database. On a real-time basis, the Campaign Extractor retrieves these campaigns and extracts it into the Campaign Authenticator, which uses it at bid-time to determine whether the campaigns match.

The Multi-Tenant Bidding Service calls the cloud function of the advertiser by sending a call to the Customer Bidding Logic. The cloud function can also read from

Cloud Table, which is a service provided by the Cloud Platform. The Log Joiner writes sanitized logs to the Cloud Platform, which contain reporting information (e.g., impressions/clicks, request/response, and amount spent) for the MTB.

Abstract

This document describes a cloud bidding method and system for DSPs, agencies, and technical advertisers, providing all the advantages of owning a bidder, such as deploying first party data and models for bidding and creative selection, with none of the infrastructure overhead or data security concerns. The advertiser only focuses on the data and models, while cloud bidding takes care of bid request ingestion, targeting, impression tracking, budgeting, pacing, creative filtering, logging, reporting, billing and more. The cloud bidding system includes a Multi-Tenant Bidder (MTB) which enables lean bidding by hosting many “tenant” bidders. The MTB receives and ingests callouts from a number of advertising exchanges (e.g., SSPs), applies targeting, budgeting, and pacing, and requests bids by invoking custom bidding functions (campaign endpoints) for each of the eligible advertiser “tenants.”