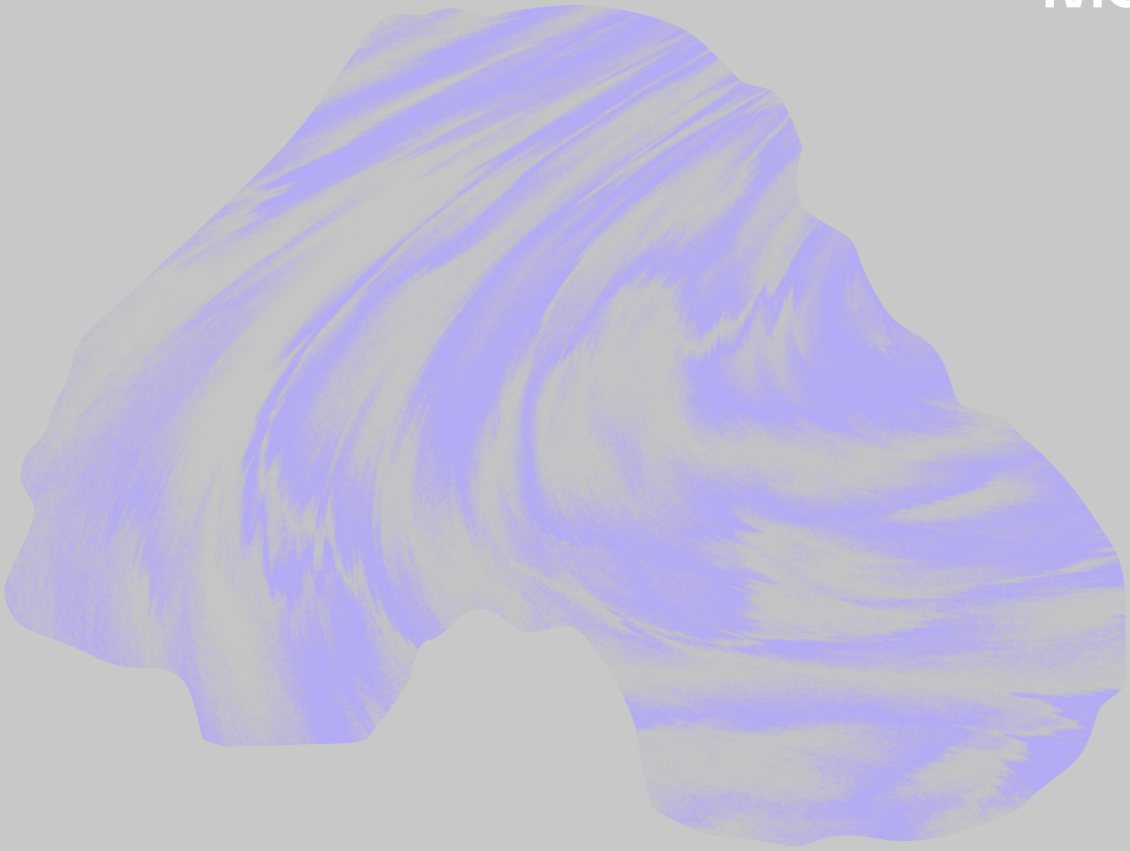
Magnite



# Microwrappers

# What we've learned so far

In June 2019, we launched Demand Manager with a simple goal: make Prebid easy to use for publishers big and small. Demand Manager allows publishers to earn revenue through header bidding across formats and device types without needing a team of developers dedicated to header bidding. It allows publishers to manage multiple Prebid configurations with a simple UI, and it keeps the Prebid.js and Prebid Server technology up-to-date and working on reliable, solid infrastructure.

With the logistics of hosting and maintaining wrappers taken care of, Demand Manager publishers have been able to develop sophisticated implementations of header bidding across large networks of diverse inventory. Some publishers have found ways to grow programmatic revenue that use specialized header bidding configurations for mobile web, tablet, and desktop inventory. Others have used Demand Manager's streamlined wrapper management tools to classify their ad units with precision for finer reporting, more granular deal targeting, and better signaling to demand partners. Others still have experimented with new identity solutions, of which Demand Manager supports nearly two dozen.

Put simply, Demand Manager freed up time and attention for our publishers, who put that time and attention to good use finding ways to optimize header bidding performance. We were excited to see Demand Manager deliver on its early promise, and we started to think about new features that would help publishers optimize the wrapper more easily.

# A new phase for Demand Manager

In late 2019, we began to design the next wave of features around the core purpose of maximizing the performance of their header bidding wrapper. We set out to design tools that allowed publishers to answer the questions they routinely asked:

- Should I add Bidder X to my wrapper? How much more revenue would I earn if I included the bidder? Would adding a new bidder cause any extra latency?
- Which of my bidders should I run client-side? Which should I run server-side?
- Who are the best bidders for my traffic from France? What about Japan?
- What's the best auction timeout for mobile web users on my gallery pages?
- How can I optimize the auction to maximize my ROI on paid traffic?
- Which identity solutions should I use? Which ones will lift my revenue?
- Does the size of the wrapper affect how it performs? How can I optimize the size of the wrapper for different device types and connection speeds?
- How does user experience impact monetization, and vice versa? How can I fine tune my wrapper to optimize for revenue as well as bounce rate, session depth, and user engagement?

Each of these questions starts the process by which the publisher grows their business through tinkering, testing, and measurement. With this new phase of Demand Manager, we aimed to deliver tools that make it possible for publishers to execute on their questions and be rewarded with concrete, measurable lift in programmatic revenue. In so doing, we defined a set of principles for ourselves:

1. **Wrappers must be flexible.** Maximizing performance means fine-tuning the configuration to each specific type of page, device, and user.

2. **Results must be measurable and meaningful.** It needs to be easy to get a clear measure of the value of any tweak or optimization. It should be able to apply scientific control and minimize disruptive variables to make sure that data is actionable.

3. **Performance must be measured holistically.** Most ad tech systems are woefully narrow-minded. Our ultimate goal is to maximize revenue across all demand sources, inside and outside of header bidding. We also need to be able to understand the effect the wrapper has on user experience.

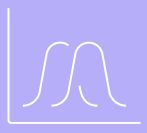
4. **Wrapper optimization must be easy** to do and must depend as little as possible on the publisher's developer team.

# A robust set of features

To deliver our publishers the capability to optimize effectively with flexibility, holistic measurable results, and a streamlined workflow, we determined that we would need to build three separate functions:

First, a flexible **wrapper targeting** system that allows publishers to tailor the wrapper's configuration to specific segments of inventory. The configurations must be completely customizable, and the segments must be able to be defined using a mixture of page and user characteristics.


Second, an **A/B testing framework** that allows the publisher to conduct tests of multiple variants of a single wrapper at the same time. The framework must make it easy to run experiments constantly, and it should be tightly integrated with analytics to provide measurable, actionable results. It must follow good scientific principles, minimizing variations in those variables that aren't being tested. It must also allow us to test bold, unproven ideas while minimizing risk.



Third, a revamped **analytics suite** that allows for holistic, multi-dimensional measurements of performance. Publishers should be able to see how their optimizations affect their returns across the entire ad stack on a revenue-per-session basis. The reports should allow the publisher to drill into granular inventory and audience segments, and they should allow publishers to measure user experience alongside monetization.

The first two functions made their way into a new feature we call **Microwrappers**. The third function is at the heart of our major recent overhaul of **Prebid Analytics**. From day one, we designed these features to be interoperable. They complement one another to form a complete solution. The rest of this paper will focus on Microwrappers, while another paper dives into Prebid Analytics.

# Optimize everything with Microwrappers

A major upgrade to the Demand Manager wrapper hosting architecture makes it possible to customize the wrapper's configuration to the page, environment and audience, and to A/B test anything. Microwrappers' innovative design also delivers those features without costs to auction execution in the browser, nor do they require ongoing engagement from the publisher's web development team.

Next, we'll describe the flexibility and the A/B testing in detail. It's best to use an example to guide our way, so let's consider a publisher we'll call XYZ Media:

- → XYZ operates a large global network with audiences in many parts of the world.
- → In some regions, XYZ has teams that source demand through localized partnerships and private marketplace deals.
- → XYZ's traffic is split between mobile and desktop devices.
- → XYZ already uses site analytics tools and focuses on optimizing their page layouts to maximize user experience.

## Flexibility

There are several tactics XYZ will want to use to maximize their revenue.

First, they want to **curate demand partners by region**. For example, XYZ's team in France would be able to curate a set of bidders that perform best on French inventory, while the team in São Paulo uses a bidder roster that performs best on Brazilian traffic.

Second, they want to **optimize ad latency**, which is the amount of time between the initial loading of the page and the rendering of ads. If ad latency is too long, XYZ will lose impressions and will negatively impact user experience. If it's too short, XYZ will see less potential demand, as bid responses time out.

With ad latency, the key is to hit the sweet spot. Where that sweet spot lies is not the same for all inventory.

- → **A desktop auction**, which has the benefit of generally faster connections and a more powerful processor, may be less pressed for time than a mobile web auction.
- → **A landing page**, on which users spend a relatively short amount of time, may require faster ad load times than an article page, where the user will spend several minutes.

→ **Traffic from paid sources** might be significantly more likely to bounce than organic traffic, causing the speed requirements to differ drastically between the two audience segments.

In this case, there is a three dimensional interaction among device type, page type, and traffic source that dictates how the auction latency should be tuned. Let's assume XYZ divides things up as follows:

| Dimension | Number of Segments | Segments | Optimization Focus |
|---|---|---|---|
| Device Type | 2 | Mobile, Desktop/Tablet | Latency |
| Page Type | 2 | Landing page, Home page | Latency |
| Traffic source | 2 | Paid, Organic | Latency |
| Region | 3 | North America, Europe, South America | Bidder roster |

This gives us 24 unique cells in our three dimensional matrix. 24 unique inventory segments, each of which will receive a customized header bidding configuration.

In Demand Manager, we call a single unique configuration bundle a "wrapper". In general, you'll use one wrapper for each unique inventory segment. So, XYZ will create 24 wrappers that will live in the Demand Manager UI.

Each wrapper has a curated set of regionally-appropriate bidders, and has three other parameters tailored to its inventory segment's latency requirements:

1. **Bidder timeout.** This is the most important parameter. It controls the amount of time Prebid gives bidders to return bid responses. A longer timeout means more bids and a slower ad load time, while a shorter timeout means more timeouts and a faster ad load time.

2. **Bidder roster.** How many bidders are competing in the auction, and who are they? How quickly is each capable of responding?

3. **Bid source.** This refers to how the bidder is integrated with Prebid: client-side using a Prebid.js bid adapter, or server-side using a Prebid Server bid adapter. Running some bidders server-side might help to capture more bids with faster auction times.

Now that we've described what publisher XYZ wants to do, let's talk about how Demand Manager makes it possible.

## Wrapper Management

Demand Manager's Wrapper Configuration section allows publishers to create all the customized wrappers they need. This is one of Demand Manager's earliest features, and we made significant enhancements for our Microwrappers release. Most of these enhancements were designed to improve ease of use for publishers who, like XYZ, use many wrappers. Almost always in ad tech, sophistication means complexity, and complexity often means confusion.

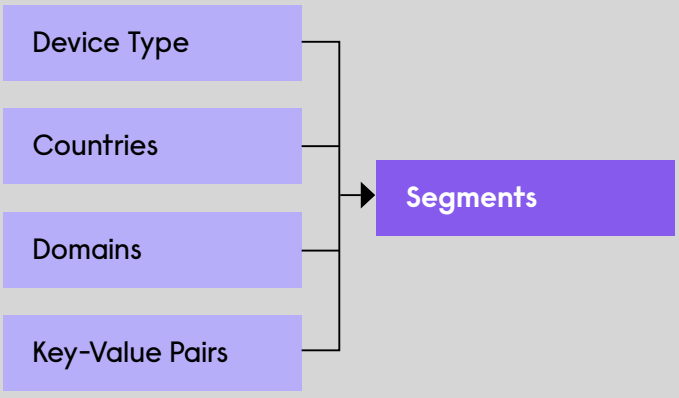
Knowing this, we've developed a long list of features that allow publishers to manage large multi-wrapper setups and remain sane. These include bulk editing features for ad slots, bidders, identity modules, and other wrapper settings like price granularity. Wrapper cloning, comparison, and version history features make it easier to keep track of things and ease data entry burdens. We delivered several of these small features with our Microwrappers release and have more on the way.

## Wrapper Targeting

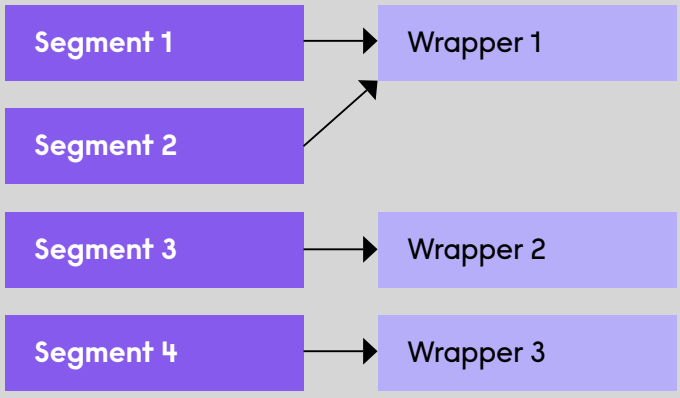
Once XYZ has their 24 wrappers set up, they'll need to map each wrapper to its dedicated slice of inventory.

Until today, publishers who wanted to customize the header bidding wrapper to the page and audience had to write custom code. At page load, Javascript running in the browser would identify the characteristics of the page and user, consult a mapping file to find the correct wrapper, then request that wrapper. The client-side decisioning logic must be developed and maintained by engineers and is difficult to adjust once it is in place. Not wanting to take on such a burden, many publishers use a generic configuration across all inventory, which costs revenue.

Demand Manager solves the problem by moving wrapper decisioning to the server-side. It replaces complex on-page code with a rules file that can be set up using the Demand Manager UI and is automatically deployed to the server-side alongside the wrappers. The publisher page implementation is drastically simplified: one universal tag will allow any wrapper to be delivered to any page. The publisher can define new inventory segments on the fly and deploy new wrappers to those segments without needing to involve their development team.




Create segments of inventory defined by device type, user location, domain, and first-party data.




Specify the Wrapper that should be used on each inventory segment.

Demand Manager's wrapper targeting solution supports the most common targeting dimensions out of the box, and it allows publishers to define segments using proprietary first party data. In XYZ's case, inventory segments are differentiated from one another by device type, user country, page type, and traffic source. Demand Manager supports the first two dimensions by default, while the second are specific to XYZ and are targetable in Demand Manager as first party data.

Once XYZ's 24 wrappers are set up and live on their 24 corresponding segments of inventory, it is time to optimize for ad latency on each one. They have a general suspicion that some situations need faster auctions and some pages need slower ones. But how fast is too fast, and how slow is too slow? To find out, they need to start testing.

## A/B Testing

### The case for A/B Testing

We doubt we'll need to spend much time here explaining what A/B testing is or why it's valuable. It is a widespread practice among successful internet businesses. Well-known **publishers** and **technology providers** in the Prebid ecosystem have been making the case for months. The problem has been that to do so has required a major investment in time and technical resources for publishers. In a world with dozens of managed Prebid services, not one has offered a robust A/B testing toolset, until now.

### Getting A/B Testing Right

Since the beginning, we have wanted to build A/B testing into Demand Manager, because testing is the key to unlocking sustainable, long-term growth in programmatic revenue. We began the design process with a long list of requirements:

- → **Test any parameter.** The framework should allow the publisher to test anything, not just the obvious things. Identity solutions, ad formats, bidders, wrapper settings, ad slot configurations, everything.
- → **Support multivariate testing.** Allow multiple parameters to be tested together. In our example, we've already investigated how timeout, bidder selection, and bid source interact to influence latency. These things must be tested together, not separately.
- → **Support "A/B/C/D/E/..." testing.** Allow multiple variants of the same experiment to be run simultaneously. This speeds up testing and gets you results much more quickly.
- → **Control extraneous variables.** True to the scientific method, the system should control any variables that aren't part of the test.
- → **Test without client-side code.** Use server-side to handle test decisioning. Don't burden the browser with the task of rolling the dice for A/B tests. Don't rely on the publisher's dev team for ordinary testing.

- → **Minimize risk with flexible traffic allocation.** Not all tests are going to work out, and publishers shouldn't be penalized for trying bold ideas.
- → **Measure test results precisely and holistically.** The A/B testing framework should be integrated tightly with analytics. It should be easy to isolate a test and see its results. Performance needs to be able to be evaluated holistically.
- → **Provide a clean, uncluttered, streamlined workflow.** Testing only works if you do it all the time. It needs to be easy to start, monitor, and conclude tests. It needs to keep things.

## A/B Testing with Demand Manager

Demand Manager's A/B testing framework is designed to be powerful and easy-to-use. It's designed to satisfy obvious short-term needs, but can be extended to satisfy sophisticated use cases in the future.

**Testing is a four step process:**

| | |
|---|---|
| 1 | Create test wrappers |
| 2 | Set the test wrappers live |
| 3 | Analyze the test results |
| 4 | Complete the test and start a new one |

### Test Wrapper Setup

We've already established that XYZ Media has 24 wrappers that represent the unique configurations for 24 inventory segments. Each of these wrappers is called a primary wrapper. It is a stable configuration that monetizes live traffic and represents the "A" state in an A/B test. To test things, XYZ can create what we call experiment wrappers: these are variations of the primary wrapper that represent states "B", "C", and so on.

Let's say XYZ wants to find the best bidder timeout setting for mobile web inventory. Their "A" wrapper is set to 1000 milliseconds, but they want to see what would happen if the value were raised or lowered. To find out, they'll use multiple experiment wrappers to test a wide range from 700ms and 1300ms.

### Traffic Allocation

Next, XYZ will set the experiment wrappers live on a slice of production traffic. With Demand Manager, it's possible to run experiment wrappers on production traffic in a true side-by-side test. To control the risk of running an unproven setup on production traffic, Demand Manager also offers fine control over the amount of traffic allocation. XYZ will allocate 80% of traffic to the primary wrapper, then remaining 20% will be split evenly among four experiment wrappers set to 700, 850, 1150, and 1300 milliseconds. This entire process happens in the Demand Manager UI and zero developers are required.

### Analytics

With the wrappers live, the test is live. XYZ can measure the performance of each of their experiment wrappers in Prebid Analytics. They are able to measure revenue lift across the entire ad stack, and they're also able to see user experience metrics. The fact that XYZ divided the traffic unevenly among the wrappers isn't a problem, because Prebid Analytics uses metrics like Revenue per Session and Ad Request CPM that normalize for differences in traffic volume. Prebid Analytics makes it easy to see what the differences in performance are among the wrapper variants, and its powerful segmentation capabilities make it possible to understand the reasons why those differences exist.

In another paper, we'll cover the benefits of Prebid Analytics in more detail.

### Test Completion

Having gathered the results, XYZ will complete their test. Demand Manager's A/B testing framework includes a test completion process that, in about four clicks, promotes the winning configuration to the primary wrapper, archives the experiment wrappers, and restores things to a clean slate so a new test can begin.

## Starting simple with A/B testing

XYZ Media is a complex example with its 24 wrappers and multiple segmentation dimensions, which allows it to demonstrate the power that a robust wrapper targeting and A/B testing framework are able to offer. But it's important to emphasize that the way to get to a complicated setup like that one is to grow slowly from a simpler one.

XYZ might start with just two wrappers: one for desktop, the other for mobile. They would be able to optimize as much as possible against device type, then, when those opportunities are exhausted, pick a new dimension (page type, let's say), and begin fine-tuning again. The trick is to make a series of small victories that add up to a significant amount of revenue over the course of weeks or months. The tight integration between Demand Manager and Prebid Analytics will allow you to measure those results and track your progress as you go.

The key point is: anybody who isn't A/B testing is missing an opportunity. Testing benefits everyone, from single-site publishers to giant networks. So, if you're an existing Demand Manager publisher or a prospective one, our advice is to start simple and run some basic tests. The Magnite team is actively working with publishers to optimize wrappers and grow revenue, and we'll spread the best practices far and wide. Together, we'll push header bidding further and further, finding answers and new problems to solve.