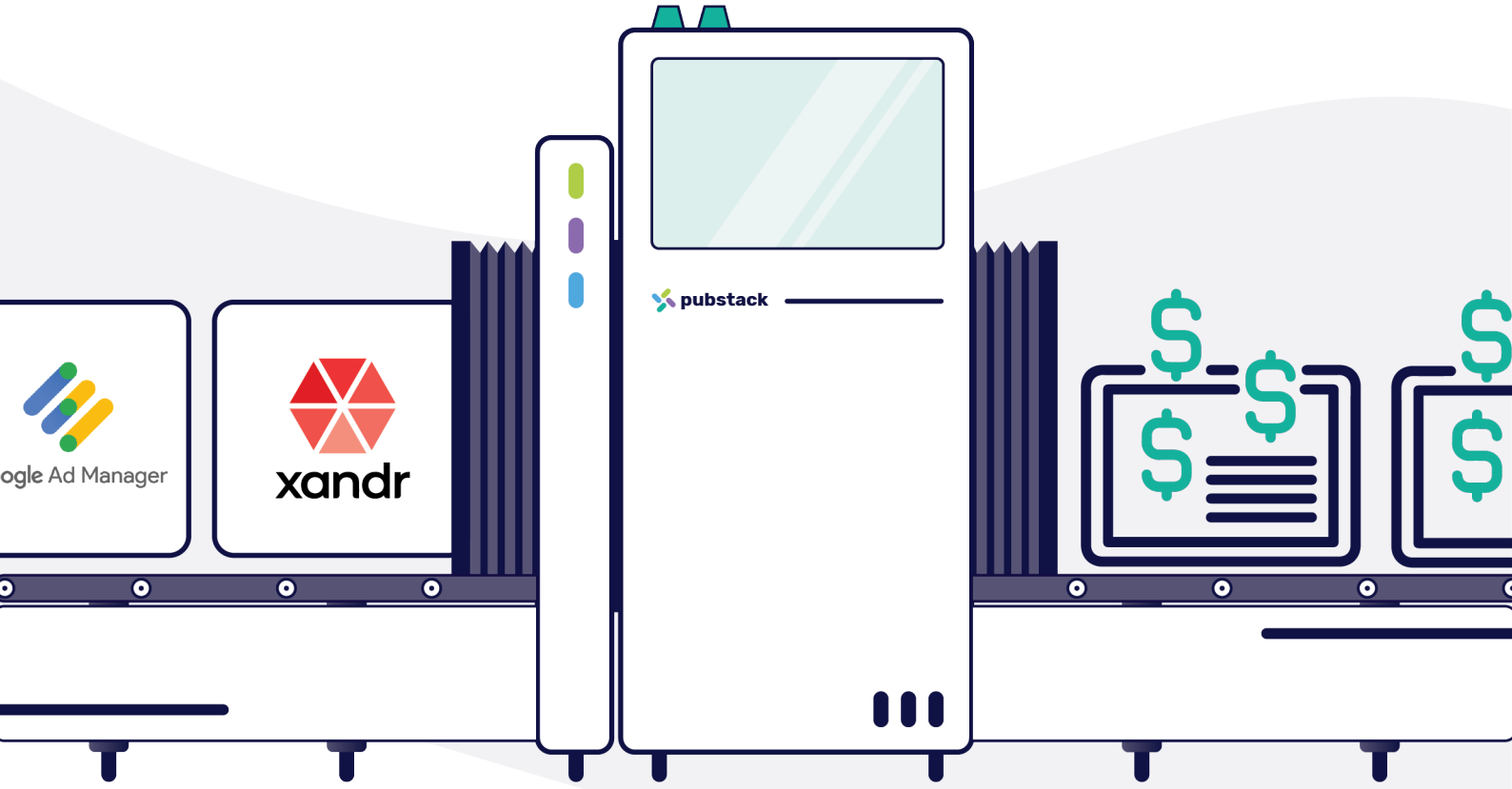
pubstack

How to build the optimal monetization machine?

A complete guide for building, monitoring and optimizing a publisher wrapper



pubstack

How to build the optimal monetization machine?

A complete guide for building, monitoring and optimizing a publisher wrapper

Introduction

Over the last few years, the number of demand partners has significantly increased, as has the number of SSPs used by each publisher, and it is now common for Publishers to work with over 20 different SSPs.

What this has meant is that Publishers have plenty of options to build their monetization machine, but they also need to navigate increasingly complex options to find the most profitable monetization machine for their business.

This guide will cover the 3 main steps to build an optimal monetization machine.

1. **Building the wrapper and carefully selecting Prebid partners.**
2. **Implementing the right monitoring tools.**
3. **Optimizing the wrapper.**

PART 1

Building my Wrapper

Part 2 - Monitoring the wrapper

Part 3 - OPTIMIZING THE WRAPPER

Part 1: Building my Wrapper

Selecting the right Prebid Partners for Optimized Ad Revenue and Page Performance

When building a wrapper, a publisher needs to decide what SSPs to connect to it. Finding the best options means considering factors such as revenue and page performance. This can be difficult and requires specific data in order to make the most accurate decisions. Because there are so many SSPs today, there is an abundance of different options. This raises very common questions from Publishers, and we have compiled the top 3:

- **How can I choose which SSPs to work with?**
- **How can I know if I have enough or too many SSPs?**
- **How can I decide which partner to part from?**

The average number of bid requests that are being sent out for each auction

(based on Pubstack's data)

How to identify the best-performing partners in a given market?

As a Publisher, when selecting your SSP, you need to make sure that you will not be going through the trouble of adding a new partner, only to realize that that partner has no demand for your inventory. What we recommend is to gather market data. By pulling data from Pubstack's large database, we are able to see what SSPs have the best results in a given region. We do not only look at their performances in terms of revenue – as this metric can be influenced by factors such as the format – but also at the SSP's internal performance. By analyzing its bid rate and bid CPM, we collect valuable insights on whether the SSP is bidding, and at what level. The share of voice on the requested inventory is also an important metric to look at as this lets you know how much demand the SSP fills. One thing to draw your attention to is that we choose to only consider the requested inventory. The reason is that if all inventory is taken into account, some SSPs that are only requested on a restricted number of ad units will have a very low apparent Share of Voice, but a much higher one on the ad units they're requested on.

Region: **UK**	Inventory type: **Display**		Device: **Mobile**
	Avg Bid Rate	Avg Bid CPM	SoV on requested inventory
SSP4	64.8%	1.48e	8.3%
SSP3	23.0%	0.62e	4.9%
SSP2	49.6%	2.15	0.7%
SSP1	37.3%	1.01	6.3%

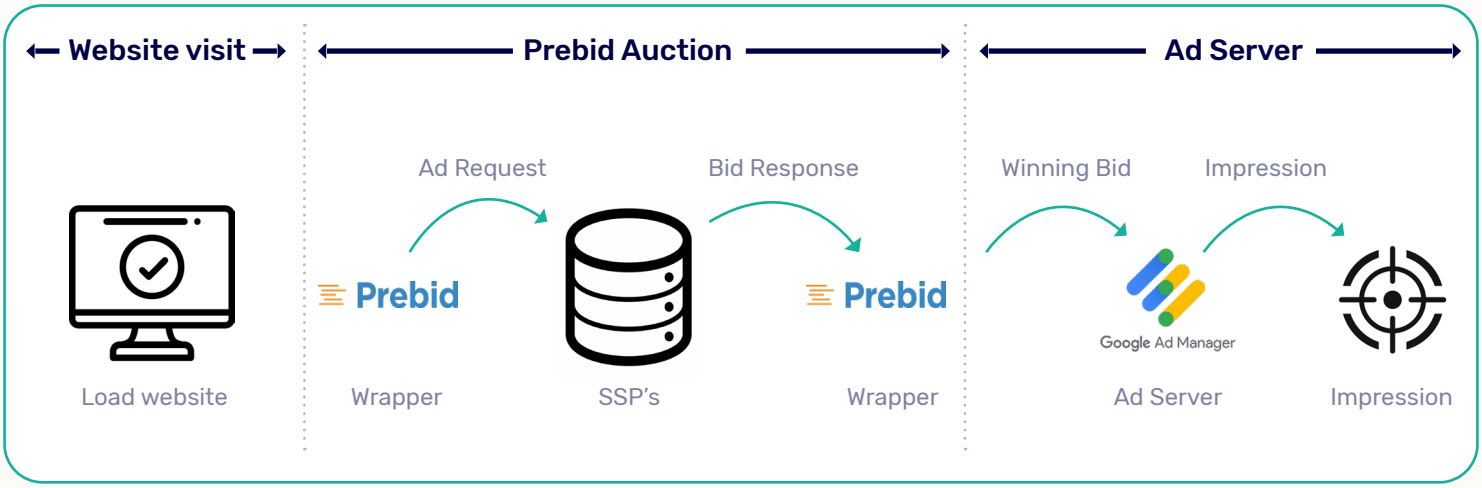
It is important for a Publisher to have a measure of how much value the SSP is bringing, and how much inventory it is filling, especially when the wrapper does not contain many SSPs.
The second-best option for publishers who do not have access to this data is to use a light implementation. A lot of our publishers first implement their SSPs through Open Bidding or other server-side solutions (Magnite server, PSP, TAM Wrapper,..) because it is much easier. This allows them to evaluate whether their new partner actually has demand for their inventory in their region.

How to accurately assess the value brought by your SSPs?

To assess the value SSPs are bringing to your ad stack, you should not only consider the revenue. If you only look at the revenue, your data will be biased because there are other actors in the monetization process. Let's take an extreme example: an SSP can have no revenue on some ad units if there is a direct campaign running on that ad unit. In that case, the revenue will be 0€, yet it is not indicative of poor performance. For that reason, it is important to consider the whole process: not only should you look at what happens in the ad server (i.e. the impression), but also analyze what happens in the Prebid auction (i.e. bid responses and winning bids).

You can also extend this measurement to assess whether your Prebid wrapper as a whole is performing well. The number of winning bids divided by auctions for example can be interesting to look at. It shows how often Prebid is participating in the competition against the ad server. If there is no winning bid coming from Prebid in the auction, Prebid is not present at all in the

auction. If there are many auctions in which Prebid is not present, it is likely that there are not enough SSPs in the ad stack, or that the current SSPs aren't able to bid properly.



Once you have broken down your process horizontally like in the example above, it can also be done vertically. Of course, when looking at metrics such as bid rate and bid CPM, you shouldn't only look at global averages because there could be some ad units where the SSPs are performing poorly, and others where it is performing very well. This is why Publishers should have the right granularity in their data. Some SSPs only bid on a single ad unit per page. If you plug them on two ad units, there will never be any bid on the second one: there will be a normal bid rate on the first one and a 0% bid rate on the second one. The example below shows a case where the global average bid rate of the SSP is about 10%, but where granularity is necessary to understand what is really going on. This SSP only bids on one ad unit with a 31.5% bid rate and practically never bids on the other ones. This is a very common trap in bid data and a lesson to remember: averages hide a lot of information.

SSP 2 | Breakdown

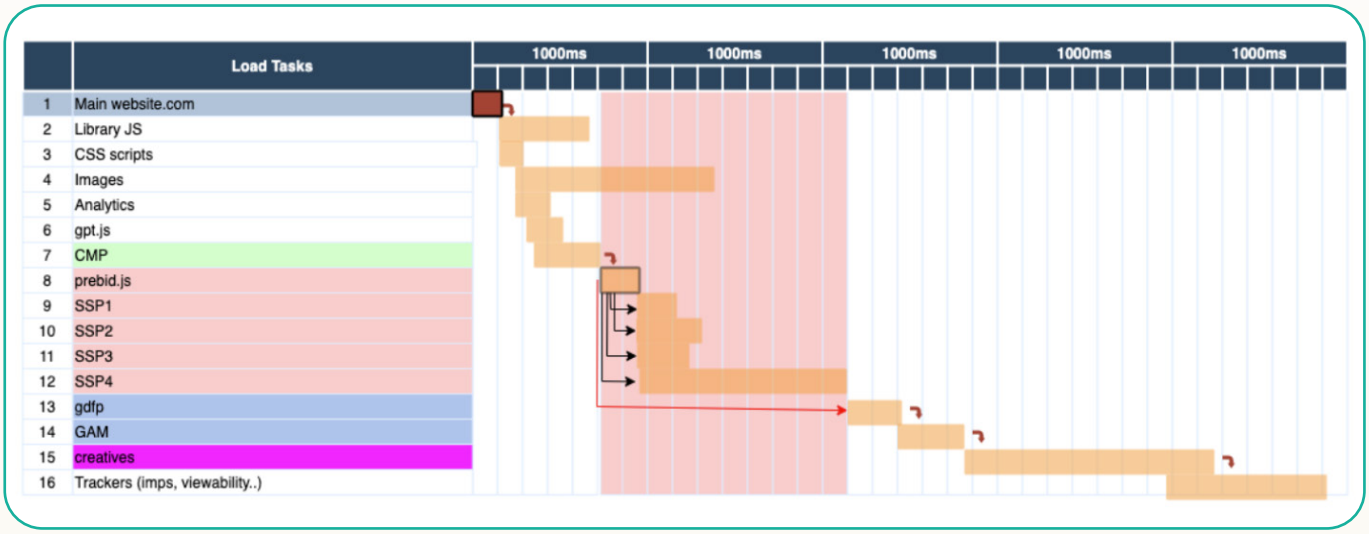
	BID REQUESTS	HB REVENUE	BID RATE	BID CPM	HB WIN RATE	TIMEOUT RATE
Ad unit 1	4.7M	523.94€	31.5%	1.81	21.3%	5.6%
Ad unit 2	2.8M	1.35€	0.3%	0.4	38.4%	8.7%
Ad unit 3	1.5M	5.84€	1.2%	0.52	75.9%	8.1%
Ad unit 4	923.2K	2.6€	1.9%	0.37	50.3%	6.9%
Ad unit 5	830.5K	6.39€	4.5%	0.46	55.7%	5.8%
Ad unit 6	572.6K	1.32€	1.5%	0.5	35.6%	8.7%

Similar cases occur when looking at the country level, with SSPs that mostly bid on specific countries. It often happens in English-speaking countries, such as the USA and the UK, where many SSPs will only bid there and not outside of these zones. By plugging these SSPs in another region, the bid rate will considerably decrease.

How to accurately assess the drawbacks brought by each SSP?

A common fear for publishers who add a new SSP is to see their auction slow down. This is usually one of the biggest drawbacks of adding SSPs: page performance tends to decrease, and having too many SSPs in a wrapper can negatively impact page speed for example. There are several aspects to consider on this topic.

In order to visualize it better, we have broken down the monetization process below:



What we see here are the main events of a monetization process. Highlighted in red are the events that occur in Prebid. This red part is only a fraction of the total time as, on average, Prebid is not what takes the most time (what does is the printing of the creatives). Furthermore, what we can see is that the bid requests sent to the SSPs are sent simultaneously. They all start approximately at the same time, right after Prebid has been called. As for the Prebid auction, it only lasts for as long as the slowest SSP takes to answer or when timeout is reached. This is very important because it means that if you add a new SSP, as long as the SSP is not slower than the ones you already have, it will not add any significant latency to your auction.

In this example, SSP4 takes 1 full second longer than all the other SSPs to answer the request. Any number of SSPs that will take less than that time will not add any delay to the auction.

Knowing this, publishers should evaluate which of their existing SSPs are the slowest. It requires monitoring the timeout rate of each SSP. In the example below, in the set of SSPs, we can easily identify that one of the SSPs has a high timeout rate of 42.3%. Almost one auction out of two is going to time-out when this SSP is called because it is taking too much time to answer.

The Publisher should ask himself whether or not it is worth keeping this SSP in the wrapper, depending on the value it is bringing. Another option is to isolate it, for example by cutting it from the ad units where it is not performing.

	HB REVENUE	BID REQUESTS	BID RATE	BID CPM	HB WIN RATE	TIMEOUT RATE
SSP 4	495.09$	86M	17.1%	0.08	73.7%	42.3%
SSP 3	345.15$	83.3M	2.3%	2.19	98.8%	6.4%
SSP 2	200.26$	91.8M	4.5%	0.43	49.5%	7.0%
SSP 1	192.7$	91.8M	3.2%	0.4	55.1%	7.8%

Part 1: Building my Wrapper

PART 2

Monitoring the wrapper

Part 3 - OPTIMIZING THE WRAPPER

Part 2 - Monitoring the wrapper

Making sure my SSPs are optimally performing at all times

Once the wrapper is built and the SSPs carefully selected and in place, Publishers need to make sure that the SSPs are performing in an optimal way at all times and that their full potential is yielded.

Some of the most frequently asked questions we receive from our publishers regarding optimization of SSP performance include the following:

- How to make sure that nothing is broken?
- How to make sure that my SSPs are performing well?
- What should I look out for in terms of the most common issues?
- And how do I fix them?

Maximizing bidding pressure in the auction

To maximize SSP performance, publishers need to find ways to maximize bidding pressure within the auction. High bidding pressure leads to high prices. This also means that Prebid will be competing against the ad server more often, thus generating more value.

To do so, publishers can focus on these levers:

PinPoint and **fill gaps** in the mapping

Uncover and fix broken AdSlots with **Reports and Alerts**

Identify and implement **missing sizes**

React right away thanks to **real time data**

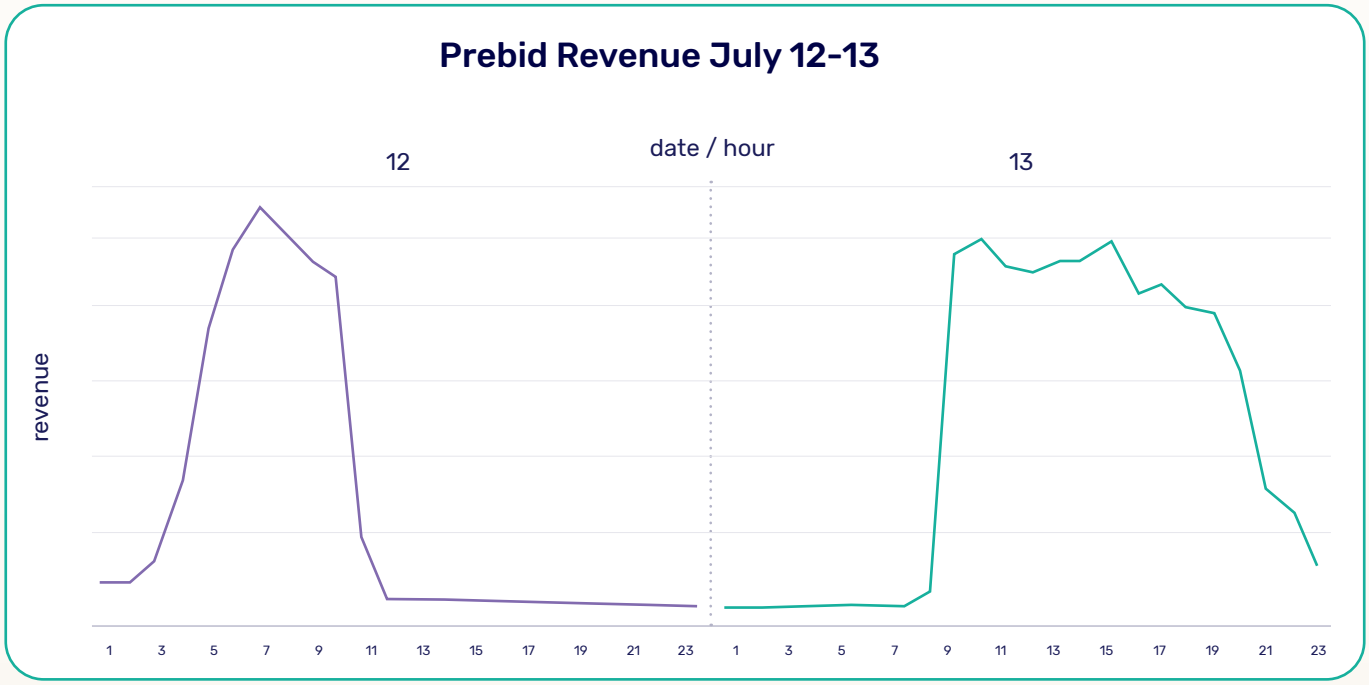
Spot **underperforming partners** on a specific context

Use **high granularity** to get to the root of any issue

A couple of examples

Use case 1
Real-time data helps you save money

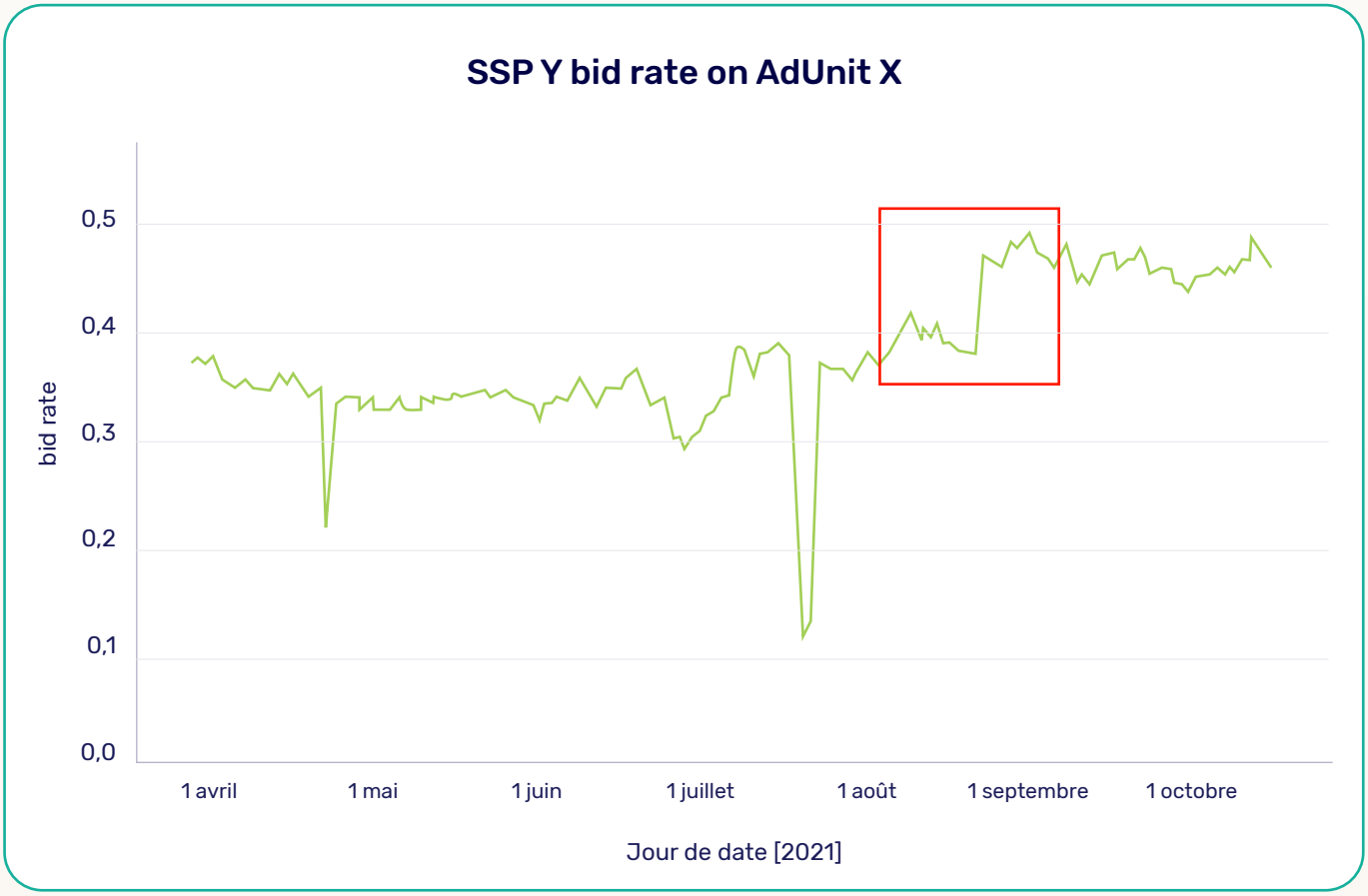
In the straightforward use case below, we see that around midday, the wrapper breaks and the revenue drops to 0 at around 11 pm. Obviously, the publisher quickly needs to spot this. The longer it remains broken for, the higher the revenue loss.



Use case 2

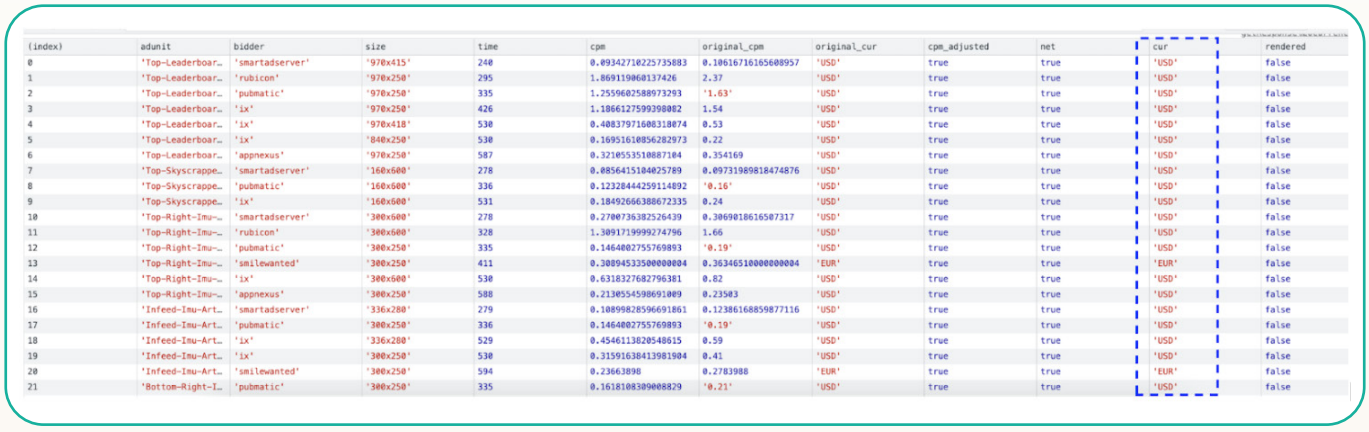
Cross-SSP metrics in the same context can help improve SSP performance

In this second use case, we analyzed cross-SSP performances within the same context. Using Pubstack's high granularity, we compared for one specific ad unit the performances of all SSPs. It appears that one SSP had a lower performance in comparison to the rest. While this is not always an issue in itself, in this case, once we had shared this information with the publisher, they uncovered a problem in that SSP's bidding pattern that was responsible for the lower performance. After fixing it, as we can see on the graph, the bid rate increased significantly and performances was once again similar to those of the wrapper's other SSPs.



Ensure the value declared is being paid

Once potential is maximized, the following step ensures that it is turning into revenue. Often, we see that what is being declared on the page is not actually what is being paid at the end of the month. There can be several reasons for that. The most common one is due to a constant discrepancy. A gap can sometimes be observed day after day between what the publisher is reporting and what the SSP is reporting and therefore paying. This gap can be due to the fact that the SSP is bidding in gross and are only paying in net, so the publisher needs to adjust those bids. It can also be due to the fact that the SSP is bidding in the wrong currency. For example, they can be bidding in USD in an auction in euros, resulting in an unfair advantage.



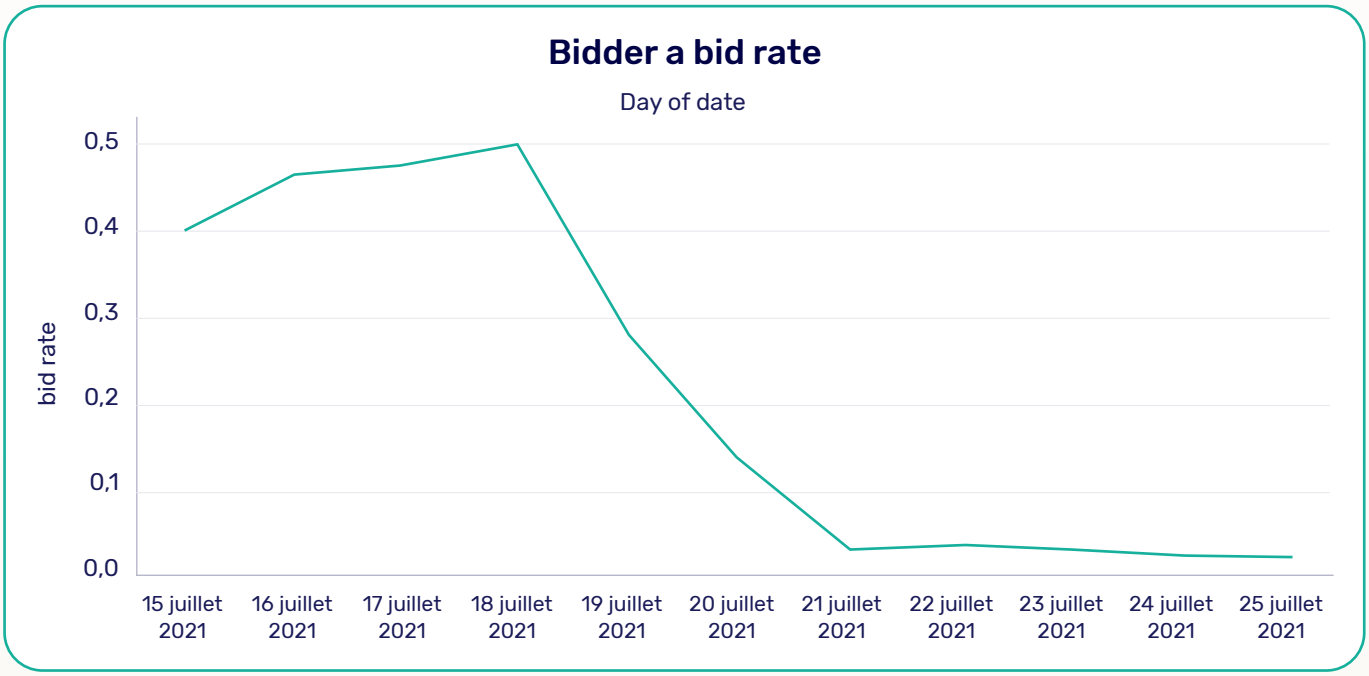
(index)	adunit	bidder	size	time	cpm	original_cpm	original_cur	cpm_adjusted	net	cur	rendered
0	'Top-Leaderboar…	'smartadserver'	'970x415'	240	0.09342710225735883	0.10616716165608957	'USD'	true	true	'USD'	false
1	'Top-Leaderboar…	'rubicon'	'970x250'	295	1.869119060137426	2.37	'USD'	true	true	'USD'	false
2	'Top-Leaderboar…	'pubmatic'	'970x250'	335	1.2559602588973293	'1.63'	'USD'	true	true	'USD'	false
3	'Top-Leaderboar…	'ix'	'970x250'	426	1.1866127599398082	1.54	'USD'	true	true	'USD'	false
4	'Top-Leaderboar…	'ix'	'970x418'	530	0.40837971608318074	0.53	'USD'	true	true	'USD'	false
5	'Top-Leaderboar…	'ix'	'840x250'	530	0.16951610856282973	0.22	'USD'	true	true	'USD'	false
6	'Top-Leaderboar…	'appnexus'	'970x250'	587	0.3210553510887104	0.354169	'USD'	true	true	'USD'	false
7	'Top-Skyscrappe…	'smartadserver'	'160x600'	278	0.0856415104025789	0.09731989818474876	'USD'	true	true	'USD'	false
8	'Top-Skyscrappe…	'pubmatic'	'160x600'	336	0.12328444259114892	'0.16'	'USD'	true	true	'USD'	false
9	'Top-Skyscrappe…	'ix'	'160x600'	531	0.18492666388672335	0.24	'USD'	true	true	'USD'	false
10	'Top-Right-Imu-…	'smartadserver'	'300x600'	278	0.2700736382526439	0.3069018616507317	'USD'	true	true	'USD'	false
11	'Top-Right-Imu-…	'rubicon'	'300x600'	328	1.3091719999274796	1.66	'USD'	true	true	'USD'	false
12	'Top-Right-Imu-…	'pubmatic'	'300x250'	335	0.1464002755769893	'0.19'	'USD'	true	true	'USD'	false
13	'Top-Right-Imu-…	'smilewanted'	'300x250'	411	0.30894533500000004	0.36346510000000004	'EUR'	true	true	'EUR'	false
14	'Top-Right-Imu-…	'ix'	'300x600'	530	0.6318327682796381	0.82	'USD'	true	true	'USD'	false
15	'Top-Right-Imu-…	'appnexus'	'300x250'	588	0.2130554598691009	0.23503	'USD'	true	true	'USD'	false
16	'Infeed-Imu-Art…	'smartadserver'	'336x280'	279	0.10899828596691861	0.12386168859877116	'USD'	true	true	'USD'	false
17	'Infeed-Imu-Art…	'pubmatic'	'300x250'	336	0.1464002755769893	'0.19'	'USD'	true	true	'USD'	false
18	'Infeed-Imu-Art…	'ix'	'336x280'	529	0.4546113020548615	0.59	'USD'	true	true	'USD'	false
19	'Infeed-Imu-Art…	'ix'	'300x250'	530	0.31591638413981904	0.41	'USD'	true	true	'USD'	false
20	'Infeed-Imu-Art…	'smilewanted'	'300x250'	594	0.23663898	0.2783988	'EUR'	true	true	'EUR'	false
21	'Bottom-Right-I…	'pubmatic'	'300x250'	335	0.16181083090008829	'0.21'	'USD'	true	true	'USD'	false

Some more tedious issues can also occur. More impressions can be counted as viewed due to the SSP having specific conditions. For example, some SSPs will only count an impression as viewed if it has been viewed for a set amount of time. Furthermore, there can be some technical issues, such as rendering issues, where the impression is detected but does not actually appear on the page. The problem can also come from the creative, thus preventing it from rendering.

The following use case is a detailed example of why it can be valuable to have different perspectives and data-collection measurements when trying to get to the bottom of a problem.

Access to different data-collection perspectives can help find the root cause of issues

Pubstack's perspective:



SSP perspective:

Here, we noticed a drop in terms of revenue for one SSP on the publisher's full inventory. What we saw from our side was that the SSP was not bidding anymore, so the bid rate went down to 0, but it was still receiving bid requests. On the other hand, after some troubleshooting efforts, the SSP was seeing a drop in the number of bid requests that they were receiving. So we see the bid requests being sent out to the SSP, but it never actually receives the requests. This means there is an issue in the process, between the moment the Prebid wrapper is issuing the requests and the moment the SSP is receiving them. When we visited the web page, thanks to the Prebid debug tool, we saw that there was an issue in the bid adapter. The bid adapter was unable to parse the requests, leading to an issue with multi-sizing and the way it was handled by the bid adapter. This was responsible for a very consequential loss of revenue for the publisher.

Part 1: Building my Wrapper

Part 2: Monitoring the wrapper

PART 3
OPTIMIZING THE WRAPPER

Part 3 - OPTIMIZING THE WRAPPER

Generating as much value as possible in the existing setup with ad refresh

Once the bidding pressure is maximized, publishers can go one step further to build the optimal monetization machine. There are many ways publishers can continue optimizing their wrapper. We will deep-dive into one of them, which is ad refresh. When we go through this step with our clients, their main questions are the following:

- **Is it possible to refresh without damaging my inventory's quality?**
- **What should the timer be? Should there be a single timer or a granular strategy?**
- **How to measure refresh-oriented metrics? And what for?**
  (first call vs refreshed impression CPM, bidder behavior, refresh rank data)

Ad Refresh: Increasing Revenue without harming the inventory

We advise our publishers to refresh their inventory in a way that sustains the value of their inventory in the long run. To do so, there are a few easy steps that publishers can follow that will allow them to yield this enormous boost in revenue while sustaining their inventory's quality. Granular rules and features to prevent layout shifts are required to implement an ad refresh strategy that doesn't harm the user experience.

Granular rules help the publisher avoid having too many refresh occurrences over the same session. Having too many refreshes may upset the users. Publishers can adjust their timers depending on the probability that the ad

unit has a refresh. A sticky ad unit will have much more opportunity to refresh than any common ad unit, especially on mobile. Therefore, it will generate much more user fatigue in comparison to other ad units. Ideally, the timer would be higher on the sticky ad unit than on others. Furthermore, having a higher timer on desktop than on mobile – since ad units are more often in view on desktop – can make a lot of sense as well. These factors should be taken into consideration when setting the timer for each ad unit.

The viewability booster effect

In terms of viewability, ad refresh actually increases viewability when refreshes are done in view. It will generate a pool of highly viewable impressions that will contribute to boosting the overall quality of the inventory. However, if the refreshes are done out of view, many of the generated impressions will not be viewed, and this will decrease the viewability.

Ad Refresh effect on CPMs

When increasing the average viewability level, the CPMs usually remain steady. There can be some gaps between the CPM of the first impression and the CPMs of the refreshed impressions. But once the publisher starts refreshing, the CPM from the first auction is not impacted. We also strongly recommend activating bid caching in order to generate more revenue from in-view ad refresh.

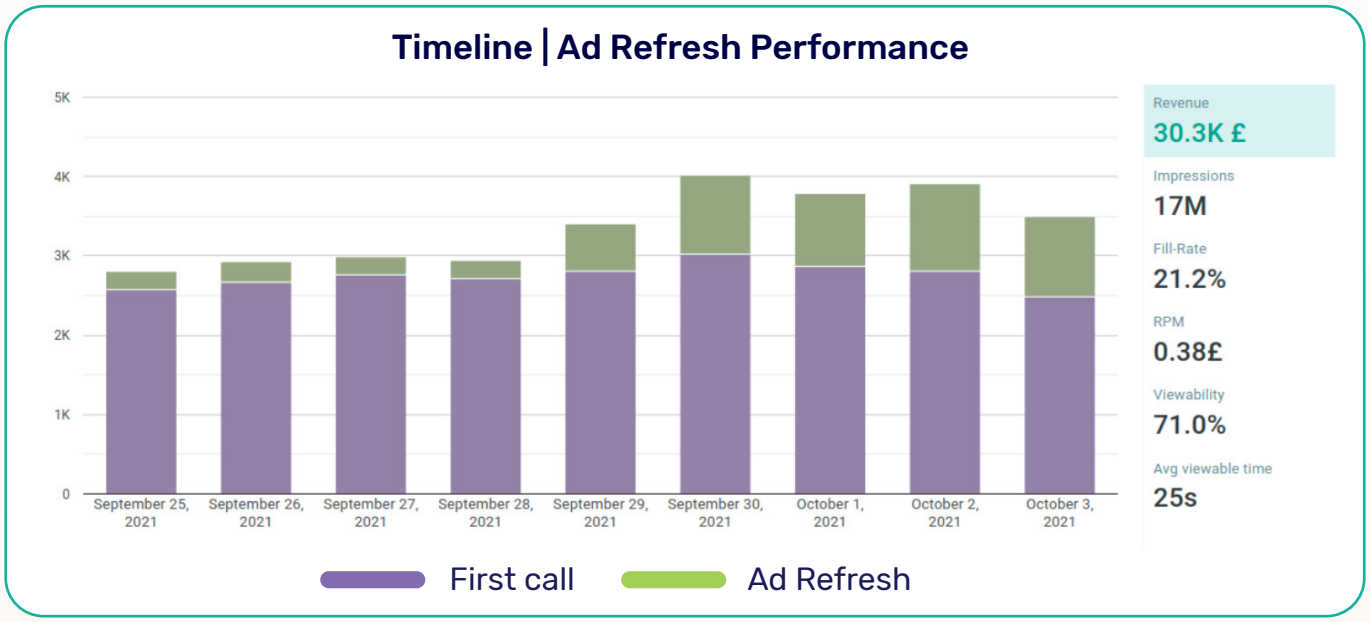
Maintaining good relationships with SSPs

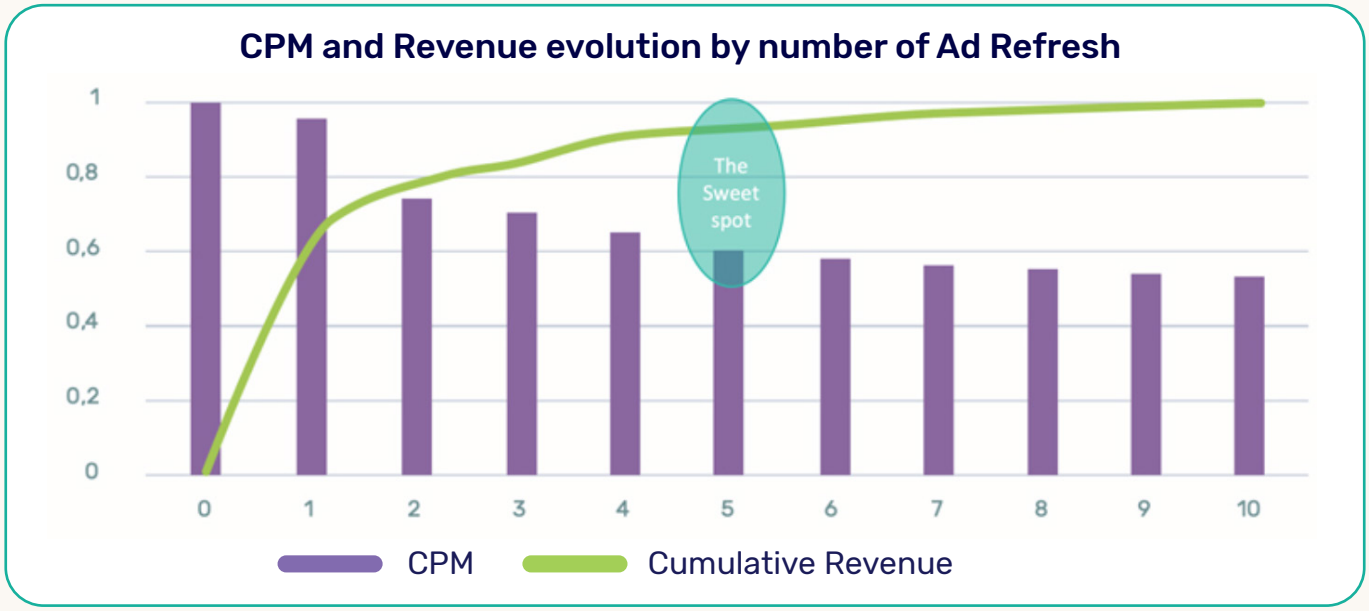
Finally, it is possible to refresh and keep a good relationship with demand partners. To do so, the behavior of the SSPs need to be monitored after the refreshes are implemented. Pubstack has helped many publishers to moni-

tor this. Once the in-view ad refresh solution is activated, we make sure that there are no adverse effects on the SSPs, either by monitoring their bidding patterns on the first call and on the refresh, or for example by monitoring the CPMs. In the rare cases where behavioral shifts are seen, the publisher can immediately adapt, by setting up a granular rule for the ad units on which the issue was detected, or by only setting it up for the SSP in question.

Use case

Finding the sweet spot between revenue and CPM





In this use case, we looked at the data to decide what the capping should be on the refreshes. On the first graph, the refreshes are generating a significant revenue uplift. We looked at the cumulative revenue based on the refresh ranks to understand what the optimal number of refreshes would be in order to set the optimal capping. The graph of the CPM and revenue evolution by the number of ad refreshes shows how the CPM decreases slightly for every additional refresh. The first refresh has an almost identical CPM to the one of the first auction, and it decreases as the number of refreshes increase. Opportunities after a certain refresh rank have much less value. In this case, after the fifth refresh, 95% of the refresh revenue had already been generated. This data allows us to put a capping at the sweet spot between CPM and revenue.

CONCLUSION

Access to granular data is key for Publishers to make the best possible decisions when building their wrapper. Many factors need to be considered by publishers before selecting their partners. A precise understanding of the value and drawbacks partners will bring to the ad stack is necessary to avoid harming overall performance. By carefully monitoring and tweaking the right levers, publishers can aim to increase the bidding pressure within their ad stack, which will result in high performance from their partners and higher prices for their inventory. Maximizing ad revenue for a publisher is a continuous process. Even with a fully optimized setup and monitoring, there are still many ways to reach significant uplifts. In-view ad refresh has proven to be extremely effective. When done correctly, this technique can allow a massive revenue uplift, while having benefits for the overall quality of the inventory thanks to viewability boost.