

# Modified

## Exclusion Targeting

**Supported**

~~Not Supported~~ Degraded

### Description

Exclusion (or negative or anti) targeting, in which a decision is made not to bid on the avail or show an ad creative, is a core component of many digital marketing strategies. Consider a brand executing a new user acquisition campaign, in this case, users who have previously engaged with the brand by visiting its Owned and Operated (O&O) properties, purchasing its products, etc., should be excluded from the campaign.

### Assessment

As stated in PAAPI 5.2: “Additional bids are commonly triggered using contextual signals” meaning that either:

1. A buyer must submit all additional bids to all Protected Audience auctions as there is no way to pre-filter potential additional bids based on Interest Groups that are included in the auction.
2. A buyer must attempt to know the Interest Groups that will be available in the upcoming Protected Audience auction at the time of the ORTB auction, which is specifically prevented by the implementation of Protected Audience.

Moreover, the term additional bids does not have a normative description.

The sections 5.2, 5.3, and 5.3.1 of PAAPI confirm that the only ability to leverage negative interest groups is by leveraging the non-normative additional bids feature.

### Remarks

There are three possible ways in which exclusion targeting could be achieved by a buyer, on behalf of a brand, within the Protected Audience framework, however all are blocked by current API restrictions.

1. Consider an exclusion targeting interest group at bid time

Exclusion targeting could be achieved by allowing the buyer to check if the current browser is a member of an exclusion targeting interest group (such as “HAS VISITED BRAND X”) in addition to the interest group that is the subject of the current auction. This workflow is prevented by the restriction: “the generateBid() function is called once for each interest group that the browser is a member of” with no exposure to other interest groups supported.

<http://web.archive.org/web/20231127020721/https://developer.chrome.com/docs/privacy-sandbox/protected-audience-api/interest-groups/#generatebid>

2. Conditionally manage a browser's inclusion within the target interest group by considering its membership of an exclusion targeting interest group

Exclusion targeting within the Protected Audience framework could be achieved by conditionally managing the inclusion and removal of a browser from the target interest group based on their inclusion within an exclusion targeting interest group. For example, when a browser visits BRAND X the buyer adds this browser to the exclusion interest group "HAS VISITED BRAND X". Elsewhere within the buyer's publisher network, inclusion within the "HAS VISITED BRAND X" interest group is checked when considering if a browser should be included in the target interest group "BRAND X AUDIENCE ACQUISITION". This workflow is prevented by the absence of a "listInterestGroups()" function within the API.

<http://web.archive.org/web/20231127020721/https://developer.chrome.com/docs/privacy-sandbox/protected-audience-api/interest-groups/#joinadinterestgroup>

3. Leverage userBiddingSignals to maintain an exclusion targeting subset of a target interest group

Exclusion targeting within the Protected Audience framework could be achieved by leveraging the userBiddingSignals within a target interest group to record when a browser has met the exclusion targeting criteria. Consider: across a buyer's publisher network, browsers that visit any publisher and meet the targeting criteria are added to the "BRAND X AUDIENCE ACQUISITION" interest group, any browser that then visits BRAND X has the userBiddingSignal within this interest group (for example "NEGATIVE") set to true. At bid time, the buyer could check this userBiddingSignal to exclude the negative targeting set. This workflow is prevented by the absence of a UPDATE operation associated with the joinAdInterestGroup() function. Consider: A qualifying browser that visits BRAND X would be added to the interest group "BRAND X AUDIENCE ACQUISITION" with the userBiddingSignal "EXCLUDED" set to true, the next time this browser visits any other publisher within the buyer's publisher network, the buyer would add the browser to the "BRAND X AUDIENCE ACQUISITION" interest group without setting the userBiddingSignal "EXCLUDED", however as all joinAdInterestGroup() function calls are SET not UPDATE, the original value of the "EXCLUDED" userBiddingSignal is lost and can no longer be considered at bid time.

<https://web.archive.org/web/20231121101019/https://wicg.github.io/turtledove#joining-interest-groups>

**Update Comments**

Per Public Comment Analysis - Working Group majority vote to update supported designation to Impractical 3/18/2024

## Create and Modify an Audience Across Domains

### Supported

Not Supported

### Description

A buyer wants to create a custom audience across multiple domains, not necessarily owned by the same publisher. The buyer wants to further segment that audience in real time based on their behavior across those domains.

### Assessment

The PA-API section 2.1 describes how buyers can register the interest group in the browser using the `joinAdInterestGroup()` function. PA-API section 2.1 subsection "[check-interest-group-permissions](#)" describes the requirements of registering the interest groups from the site with a different origin. Once the interest group is registered it will take a part in the Protected Audience auction conducted later on any other site (including the sites of the multibrand publishers). According to the section 4.1 subsection "validate and convert auction ad config" browser will add all interest groups of all buyers listed in the `interestGroupBuyers` property of the auction config.

If the interest group is registered in the browser and the buyer is present in the auction config on any other site the group will participate in the auction.

### Remarks

See Interest Group section in Introduction and the Target a Single Campaign use case for more information

Further the following must be noted:

- It is only possible to change the composition of an Interest Group from the point in time the update was made and moving forward
- Updates to Interest Groups can only be done once per 24 hours
- It is functionally impossible to manage conflicts if site visitation of more than one site is considered in the composition of an Interest Group
- ~~Creatives must be associated with an Interest Group meaning that a brand's activation strategy must be pre-determined at audience building time~~

### Update Comments

Per Public Comment, creatives do not need to be associated with an Interest Group Prior to adding users, so some text was removed.

It should be noted that it is possible to join Interest Groups across sites, but it is not possible to append an existing user's membership details based on backwards-facing attributes within an Interest Group.

There are mechanisms to send and receive user attributes at bid time, but they are limited only to actions taken on a single publisher or site. A mechanism for buyers to segment a user within an Interest Group

## Target a Single Campaign to My Online Audience

### **Supported**

Supported

### **Description**

As a brand, I want to run a campaign targeted to users who have previously visited my website.

### **Assessment**

The PAAPI provides the one-directional registration of the interest groups without the ability to check and reuse the already registered ones. Advertisers can't see the existing Interest Groups. They can only use the following methods to manage interest groups:

- `joinAdInterestGroup()` to add the interest group to the browser (section 2 of PAAPI)
- `leaveAdInterestGroup()` to remove the interest group from the browser (section 3 of PAAPI).
- In addition, at the end of each auction (section 4.1 of PAAPI), the interest groups can be updated (section 8).

These methods can only manipulate the data stored in the browser and don't provide any information to the advertiser. As a result advertisers can't read interest groups. However this does not impact this narrow use case.

### **Remarks**

Brands may message users who have visited their owned and operated website on that device across the web but it should be noted that it may only be used for a single Interest Group. ~~which is limited to a single campaign.~~

See Create and Modify an Audience, and all other use cases in this section for additional information.

### **Update Comments**

Per Public Comment, Interest Groups may be associated to multiple campaigns

## Bid Using a Deal ID

### Update Comments

No updates were made, but the Working Group deemed it necessary to have more in depth conversations around specific mechanisms as Feature Requests.

## Receive a “No Bid” Response from a DSP

### Update Comments

This use case was not updated, but will move to a workstream dedicated to feature requests for further analysis.

## Invalid Traffic

### Supported

Impractical

### Description

As an advertiser, I wish to ensure that traffic where my ads are shown originates from humans.

A publisher with low fraud and fast performance is more valuable to advertisers and more trustworthy to partners.

### Assessment

Privacy Sandbox is designed to make **users and/or devices traffic** unidentifiable. Advertisers and Publishers might be able to distinguish traffic from humans using the Private State Token (PST) with a suitable issuing party.

### Update Comments

Updated the first sentence of the assessment to clarify that Privacy Sandbox was designed to make users and/or their devices, not traffic, unidentifiable.

## Publisher Revenue Accrual and Impression Validation

### Description

As a publisher, I want to be able to fire a pixel that directly logs an event into my own database, that tells me that an ad rendered, who the advertiser was, and what Supply Path the advertisement took so I can generate reports related to advertising activity on my website.

### Supported

~~Temporarily Supported~~ Not Supported

### Update Comments

Per Public Comment, changed from Temporarily Supported to Not Supported. Anything that makes publishers reliant on their ad tech partners is considered not supported.

## Bid Loss Reporting

### **Supported**

Not Supported

### **Description**

As either a DSP or an advertiser, I want to understand why my bid did not win to inform and optimize my future bidding strategy.

### **Assessment**

~~See assessment of Second Price Auction Reporting.~~

PAAPI contains no reference to functionality that could be considered useful for reporting lost auctions to all auction participants. As the use case does not relate only to the winner of the auction, but all auction participants including those that lose, we do not consider the use case supported.

### **Remarks**

From the point of view of an existing server-side bid request and associated auction, a DSP (and therefore the advertisers it represents) has no way of knowing exactly which Interest Groups are eligible to participate prior to the execution of the client-side auction within the Protected Audience worklet. Through the [Event-Level Reporting API](#) (supported at least until 2026), a DSP will be informed of each winning bid along with the name of the Interest Group that won the auction.

Understanding the precise auction behavior (bids submitted, bid price, bid losses and reason for loss) is critical for bidder development and improvement, debugging, customer support and machine learning for automatic optimization.

Currently, Protected Audience API exposes this data through a temporary event level API and a long-term reporting API based on Private Aggregation.

- (1) The [Temporary Auction Reporting API](#) allows bidders to log arbitrary data relating to a single Interest Group auction (including the internal state of the bidding function). This API is only intended for technical support and debugging of DSP integration to Privacy Sandbox, and not as a mechanism for ongoing monitoring or automatic optimization. For privacy reasons, the amount of data is highly rate limited; it is not recommended (and may be impossible) to record an accurate representation of the bid landscape across all Chrome browsers
- (2) The [Private Aggregation API](#) also enables exposure of internal auction state (including non-winning bids), but delivers reports through a private aggregation service with dimension limits, metric representation, noise, and delay. Although subject to privacy

restrictions through aggregation, this API is intended to capture the overall state of all auctions across all Chrome browsers.

Because both APIs record data at the individual IG level, there is no simple way for a DSP to understand behavior at the overall auction level (for instance: understanding how multiple IGs owned by the same DSP interact when bidding against each other).

#### **Update Comments**

Removed reference to 'Second Price Auction' because it was removed from the assessment. The mechanism is the same for both, so copied from the removed use case into this use case.

## Managing Infrastructure Costs

#### **Supported**

Not Supported

#### **Description**

One of the key challenges of the ad-tech ecosystem is the problem of scale. To meet scale requirements, publishers, DSPs, SSPs, and other members of the ecosystem need to provide network, compute and services, as well as manage those services. These services in turn require resources that have real-world implications - rack space with its physical limitations, power (with its implications for the environment and heat constraints), network capacity and manufacturing costs mean that it is critical that companies scale their resources in the most efficient way possible. Corporations have spent billions of dollars to stand up and maintain this infrastructure. We need to ensure that we can provide advertising services with a similar cost model and similar scaling model to existing auctions by leveraging this infrastructure and processing the new demands of PA-API with a minimum of new compute and network load on the system.

#### **Assessment**

Privacy Sandbox specifications keep silent about what new services and infrastructure will be required to support it. ~~So that the publishers, DSPs, SSPs, and other members can't forecast the cost of adopting and supporting the infrastructure and services.~~

#### **Update Comments**

Removed line referencing the difficulty forecasting costs because a [Cost Forecasting Tool](#) did exist at the time of the analysis.

# Reassessment based on Additional Information

## Multi-touch Attribution

**Supported**

~~Not Supported~~ Impractical

### Description

As a Brand I want to know the relative contribution of prior ad exposures across publishers' ad inventory in driving marketing outcomes (e.g., binary action such as a purchase or continuous value such as dollars spent relative to media spend) for the following purposes:

- 1) Timeliness requirements. Rapidly informing and optimizing how to purchase future ad inventory after winning a prior exposure opportunity (i.e. "next-click" or less than 3 seconds) via the use of unaggregated data;
- 2) Accuracy requirements. Accurately informing the optimization of next-click media budget allocation decisions across different OS, browser, media properties, ad creatives by geographic region, day of week and time of day via the use of unaggregated data.
- 3) Scale requirements. To ensure the improvement in effectiveness is maximized, I want a solution that covers the majority of my Display ad spend across publishers' properties.

### Assessment

It is possible for an advertiser to utilize the [Shared Storage API](#) in combination with the Private Aggregation API to achieve this use case. Google provides an example on how to do this [here](#).

However, as with other use cases that use the Private Aggregation API there are a few [limitations](#) and degradations that are introduced. Private Aggregation API does not support event-level notifications, introduces delay, introduces noise, requires an aggregation service, and sets the limit of an aggregation key of 128 bits.

Due to the removal of IPs the ability to measure a user's journey across devices is severely degraded.

The Shared Storage API does not support Web to App conversions.

The Attribution Reporting API does not currently support multiple touch points, and only allows [prioritization](#) of a single event during a user's journey.

### Remarks

It is possible for an advertiser to utilize the [Shared Storage API](#) to track touch points for a given user across sites, with the caveat of being constricted to the same user agent.

The Shared Storage API can be accessed both by the top level frame and a fenced frame/iframe in which an ad is rendered in. An advertiser can use the Shared Storage API's "append" method to record a user's history of where a certain ad/campaign/etc was viewed.



When a conversion occurs, the advertiser can then use the Shared Storage API's "run" method to retrieve the user's touch points and then report it via the Private Aggregation API.

Note that due to the nature of the Shared Storage API's security feature, it is possible to track a conversion without using the Attribution Reporting API.

As per Google's documentation on the Private Aggregation API, reports are sent with a "[random delay up to one hour](#)."

### **Update Comments**

Use case was re-evaluated because the Working Group did not take into consideration the Shared Storage API in the original assessment. Omni-channel is Not Supported, this use case pertains only to Multi-Touch Attribution within a single browser.

After further review, the Supported designation was updated to Impractical. Additional functionality contained in an update to [HTTP Response Headers](#) was released after the cut-off date for the analysis but had it been live, the designation would have been Degraded. Future versions of this report will have an updated designation.

Note that we consider the removal of IP addresses for this assessment, not just the removal of 3rd party cookies.

ORIGINAL FOR COMPARISON PURPOSES

### **Supported**

Not Supported

### **Description**

As a Brand I want to know the relative contribution of prior ad exposures across publishers' ad inventory in driving marketing outcomes (e.g., binary action such as a purchase or continuous value such as dollars spent relative to media spend) for the following purposes:

- 4) Timeliness requirements. Rapidly informing and optimizing how to purchase future ad inventory after winning a prior exposure opportunity (i.e. "next-click" or less than 3 seconds) via the use of unaggregated data;
- 5) Accuracy requirements. Accurately informing the optimization of next-click media budget allocation decisions across different OS, browser, media properties, ad creatives by geographic region, day of week and time of day via the use of unaggregated data.
- 6) Scale requirements. To ensure the improvement in effectiveness is maximized, I want a solution that covers the majority of my Display ad spend across publishers' properties.

## Assessment

Multi-Touch - section 12.9 of the [Attribution Reporting API](#) states:

*An event-level report a is lower-priority than an event-level report b if any of the following are true:*

- *a's trigger priority is less than b's trigger priority.*
- *a's trigger priority is equal to b's trigger priority and a's trigger time is greater than b's trigger time."*

....

*Remove lowestPriorityReport from the event-level report cache.*

Only the last event or the one with the highest priority will be attached to the report resulting in MTA support not being available.

Timeliness - section 8 of the [Attribution Reporting API](#) defines a minimum time delay for any report to be 1 hour.

*Min report window is a positive duration that controls the minimum duration from an attribution source's source time and any end in aggregatable report window or event-level report windows. Its value is 1 hour.*

It will not be possible to obtain a report in less than 1 hour from the source time which is greater than the 3 seconds required.

Accuracy - section 18.2.2 of the [Attribution Reporting API](#) includes an issue titled "add links to the aggregation service noise addition algorithm". As such we are unable to assess how the accuracy of the report will be impacted. The service interface is not defined and we're unable to determine if the caller will have control over the amount of noise added which would be important for those with smaller data sets.

The designers have considered the need to disable noise for testing purposes in section 16.1 of the [Attribution Reporting API](#) where the document explains how noise can be turned off.

*Without this, reports would be subject to noise and delays, making testing difficult.*

It is not clear if the disability of noise or the selection of the noise aggregation service will be available to advertisers and publishers using the API.

App-to-Web - section 14 of the [Attribution Reporting API](#) describes optional app and web support when it states "If the user agent supports web/OS registrations". There has been some consideration for app-to-web but support is not mandatory. Tighter language concerning the requirements of the implementor is required to assess this.

Scale - the implication of the [Attribution Reporting API](#) is that scale requirements will not be met due to the inherent limitations of the design.

Cross-Device - [Attribution Reporting API](#) is silent concerning an explicit statement concerning cross-device support. Given the context of PS we consider that cross-device is not supported.

## Remarks

1. Content of the header Attribution-Reporting-Register-Source
  - a. Google in the [developer guides](#) provides some examples of the header value. We can take them as a source for searching [Attribution Reporting API](#).
2. Requirements for using the attributes on registering the attribution trigger
  - a. Google in the [developer guides](#) provides the table with requirements for using the attributionsrc and attributionReporting attributes.
3. Explainer: [Attribution Reporting with event-level reports](#)
  - a. Creating [event-level reports limitations](#):
    - i. Each navigation source is allowed to schedule only a maximum of three reports, while each event source is only allowed to schedule a maximum of one.
    - ii. If a source has already scheduled the maximum number of reports when a new report is being scheduled, the browser will compare the priority of the new report with the priorities of the scheduled reports for that source. If the new report has the lowest priority, it will be ignored. Otherwise, the browser will delete the scheduled report with the lowest priority and schedule the new report.
  - b. The section [Multiple sources for the same trigger \(Multi-touch\)](#) says:
    - i. *If multiple sources were registered and associated with a single attribution trigger, the browser schedules reports for the one with the highest priority. If no priority is specified, the browser effectively performs last-touch.*
    - ii. *There are many possible alternatives to this, like providing a choice of rules-based attribution models. However, it isn't clear the benefits outweigh the additional complexity. Additionally, models other than last-click potentially leak more cross-site information if sources are clicked across different sites.*
4. Explainer: [Attribution Reporting API with Aggregatable Reports](#)
  - a. [Storage limits](#)
    - i. The browser may apply storage limits in order to prevent excessive resource usage.
    - ii. Strawman: There should be a limit of 1024 pending aggregatable reports per destination site.
    - iii. Note: The storage limits for event-level and aggregatable reports are enforced independently of each other.
  - b. [Contribution bounding and budgeting](#)

- i. Each attribution can make multiple contributions to an underlying aggregate histogram, and a given user can trigger multiple attributions for a particular source / trigger site pair. Our goal in this section is to bound the contributions any source event can make to a histogram.
- ii. This bound is characterized by a single parameter: L1, the maximum sum of the contributions (values) across all buckets for a given source event. L1 refers to the L1 sensitivity / norm of the histogram contributions per source event.
- iii. Exceeding these limits will cause future contributions to silently drop.
- iv. While exposing failure in any kind of error interface can be used to leak sensitive information, we might be able to reveal aggregate failure results via some other monitoring side channel in the future.
- v. For the initial proposal, set L1 = 65536. Note that for privacy, this parameter can be arbitrary, as noise in the aggregation service will be scaled in proportion to this parameter. In the example above, the budget is split equally between two keys, one for the number of conversions per campaign and the other representing the conversion dollar value per geography. This budgeting mechanism is highly flexible and can support many different aggregation strategies as long as the appropriate scaling is performed on the outputs.
- vi. The browser also applies a limit on the number of contributions within a single report.
- vii. Strawman: There should be a limit of 20 contributions per aggregatable report.

5. The [Report Schedules](#) doc says:

- a. Regardless of report type, reports are only sent when the browser is running and online. If the browser is online and fails to send a report, it tries to send the report again after 5 minutes. After the second failure, the browser tries to send the report again after 15 minutes. After that, it's not sent and the report is deleted.
- b. Aggregatable reports are sent by the browser to the ad tech provider or advertiser with a random delay between zero and 10 minutes, or with a small delay after the browser starts again.
- c. View-through conversion reports are sent by the browser to the ad tech approximately one hour after the view event (that the conversion is attributed to) is no longer eligible for attribution.
- d. Click-through conversion reports are sent following a more complex schedule of attribution windows; they are sent at least two days after conversion.
- e. Click reports are sent during one of three windows: 2, 7, or 30 days after conversion.
- f. After the initial ad click or view, a schedule of built-in reporting windows begins. Each reporting window has a deadline. Conversions registered before that deadline are sent at the end of that window.
- g. If you need to filter out conversions happening after a certain window, you can use [custom report windows](#).

6. [Combating Fingerprinting with a Privacy Budget](#)
  - a. There will be [no way to opt out](#) of the budget.
  - b. Publisher Sites can [limit third parties](#) use of the budget
  - c. Also Google is going to restrict rivals access to IP address by routing through a Google controlled proxy server to [anonymize the user's IP](#) address from others
    - i. Google's Privacy Proxy server will restrict geography from rivals and assign IP addresses that represent the user's [coarse location](#), including country.
7. [Cross App and Web Attribution Measurement](#)
  - a. *Currently, the [Attribution Reporting API](#) supports attributing events within a single browser instance. This proposal expands the scope of attribution to allow attributing conversions that happen on the web to events that happen off the browser, within other applications.*
  - b. *Currently, the [Attribution Reporting API](#) (classically referred to as the Conversion Measurement API) supports attributing events on a single device, within a single browser instance. With this proposal, browsers that support a "sign-in" feature can allow attribution across all the user's devices, as if they had a unified storage.*

## Removed

The below use cases were removed because they relate to auctions that are not widely used today

### Second Price Auction

#### **Supported**

Degraded

#### **Description**

As either an advertiser or publisher, I want to be able to participate in a traditional second price auction.

#### **Assessment**

The section 4.1 of PA-API, states that the configuration property auctionSignals, that was passed to the runAdAuction() function via AuctionAdConfig structure, will be passed to the generateBid() function (subsection "generate a bid") and later to the scoreAd() function (subsection "generate and score bids"). So that data will be available to buyers and sellers and can contain information about the auction kind.

The section 12.4 of PA-API defines the auctionSignals as:

*"Null, a string, a Promise, or failure. Opaque JSON data passed to both sellers' and buyers' script runners."*

For reporting, PAAPI (section 4.1 subsection “report win”) highestScoringOtherBid is available which contains the bid value of the second highest scoring bid. Additionally, madeHighestScoringOtherBid is also available that lets the buyer know if they were the owner of the highest and second highest bid.

Through reportResult() or reportWin() both the seller and buyer can determine what the final bid price actually is. This is done by adding one cent to the highestScoringOtherBid value. See the assessment of Bid Price Reporting for Winners for the details.

In order to run a truly second price auction, a seller must score bids purely on price meaning that they are unable to score by other criteria. Sellers will only have access to the price of the second highest overall scoring bid that may contain non-price based factors but may not be the second highest price. For some ad servers/SSPs/auctioneers, this actually matches current behavior; for others it is a degradation.

### **Remarks**

This is markedly different from traditional OpenRTB bids in which a seller can support second price auctions while still using different scoring criteria beyond pricing.

### **Business Impact**

Private Marketplace (PMP) mechanics currently available in programmatic advertising solutions are hindered. Certain buyers of inventory have preferred pricing structures that can sometimes, but not always, supersede second price auctions. The optionality to support this prioritization of lower bids is not available in PAAPI for second price auctions. This degraded support impacts the utility of second price auctions.

## Second Price Auction Reporting

### **Supported**

Not Supported

### **Description**

As an auction participant, I want to get reporting on the winning and second-highest bid.

### **Assessment**

PAAPI contains no reference to functionality that could be considered useful for reporting lost auctions to all auction participants. As the use case does not relate only to the winner of the auction, but all auction participants including those that lose, we do not consider the use case supported.

### **Remarks**

Temporary reporting for this use case is supported in PAAPI via the reportAdAuctionLoss function. It allows auction participants to learn the winning bid value and second-highest scoring bid value. Additional information is also provided, such as whether the interest group owner was

the one to submit the second-highest scoring bid. However, the support for this function is only temporary to help the industry test and adopt PAAPI via the for-debugging endpoint.

As described in the executive summary, we do not consider any functions under forDebugOnly because as the name implies they are only for testing and debugging purposes.

The above doesn't apply to auction winners. As documented in "Bid Price Reporting for Winners" use case above, auction winners will be able to get reporting on the second-highest scoring bid value along with their own winning bid value.