

Publisher/SSP Controls & Reporting in PARAKEET

Publisher Inputs (Controls)

API is described on [GitHub](#).

1. Controls when creating Ad interest group using AdInterests
 - a. When an interest group is created, one can specify allowed readers.
`'readers': ['first-ad-network.com', 'second-ad-network.com']`
2. Controls when creating Ad request using adRequestConfig
 - a. Anonymized Proxy URL for requesting ad bundle: The origin that the anonymizing service request the ad bundle from. [privacy-preserving-ads/Parakeet.md at main · WICG/privacy-preserving-ads · GitHub](#) - Potential API shape
 - b. *anonymizedProxiedSignals: AnonymizedSignals[]*; - Specifies what information should be added or allowed to pass through with the ad request by the anonymizing request service.
 - c. Reporting logic URL:
`'reportLogicUrl': 'https://seller.example/scripts/reporting_worklet.js'`
 - d. Publisher code, ad unit and targeting interests

```
'publisherCode': '10931',  
'publisherAdUnit': 'publisher_ad_location_1',  
'targeting': { 'interests': ['music', 'sports'] },  
'geolocation': { 'lat': 41.5, 'lon': -81.7 } },  
'anonymizedProxiedSignals': ['coarse-geolocation', 'coarse-ua',  
'targeting', 'user-ad-interests']
```
 - e. Provide contextual targeting [interests, geolocation]. This will help inform ranking service on Ad Network.
 - i. Example: [GitHub](#)
 1. `'targeting': { 'interests': ['music', 'sports'] }`
 2. `'geolocation': { 'lat': 41.5, 'lon': -81.7 }`
3. Controls using myAuctionConfigForPlacement1
 - a. `'sellerSignals': {...}` See [FLEDGE](#) for details about 'sellerSignals'
4. Controls using generateBid
 - a. auctionSignals: The auctionSignals object from the finalizeAd call
 - b. perBuyerSignals: The respective ad server's signals from the finalizeAd call
 - c. browserSignals: Constructed by the browser. Can contain historical information like previous ad wins to allow generateBid to perform frequency capping and other user specified signals such as age restriction
5. *fallbackSource* : If anonymized ad request flows are not supported, either because of a global decision, a local user decision, or due to an infrastructure outage, this will be used to load the ad content as a fallback.

For Publisher to enable ad based on user features in local storage and request through service:

1. Include the PARAKEET polyfill either directly on a page or with other scripts/ad-tech (SSP)
2. Execute new Javascript API to request an ad with any contextual information or interests available. See: [Create an AdRequest and serve an ad](#)
 - a. On success, create an iFrame, set the src to the provided URL and insert into the Document to display the ad.
 - b. On failure, be sure to use [sendBeacon](#) or other mechanisms to log and track failures.
 - c. Add flooring bids in call – client side worklet - win/loss notification (delayed) - why lost
3. Provide JS logic that would select ads in various header bidding outcomes. Floor price is in the call.

As an alternative, programmatic SSP can enable PARAKEET ad requests on behalf of publisher. SSP needs to create and support service at endpoint specified in *proxiedAnonymizingOrigin*

Publisher Outputs (Reporting):

The following statistics can be provided to publisher/SSP about the ad calls received by the Parakeet service:

1. Number of ad calls – bucketed by proxy url – which ad network received which ad call
 - a. we may need to take into account feedback source as well
2. Number of successful ad returned calls
 - a. win notifications will not be provided (parakeet is not aware of whether an ad was returned. but on publisher JS, payload can have this information - return of the response is indicator of successful ad returned.)
3. Number of malformed Proxy Origin ads
4. Number of requests with malformed readers
5. Number of dropped calls/changed calls :
 - a. If the parakeet service drops the user-ad-interests attribute entirely because of multiple similar requests from user browsers – count of that, we should also have alerting on our end to see if such requests are coming from a specific SSP/Publisher and do they need to be informed

Monitor.json in Parakeet service logs most of these counters in Geneva/Jarvis – we can have a spark job that aggregates the data, and an API to return this information

Open Questions:

1. Can the SSP aggregate calls on targeting groups and see the ad call behavior for it?
2. Needs for Reporting on C' v/s needs for reporting on S' (Where C' is anonymized contextual information, and S' is anonymized user interest/segment information)
3. multiple request- challenges