3 major benefits about server side tagging product:

1. Conversion API
2. Custom Audiences
3. Offline conversion API

Basically it uses AdTech platform’s server side API to send the information to the said platform, bypassing the need of third party cookies, let me take example of FB’s Conversion API (CAPI).

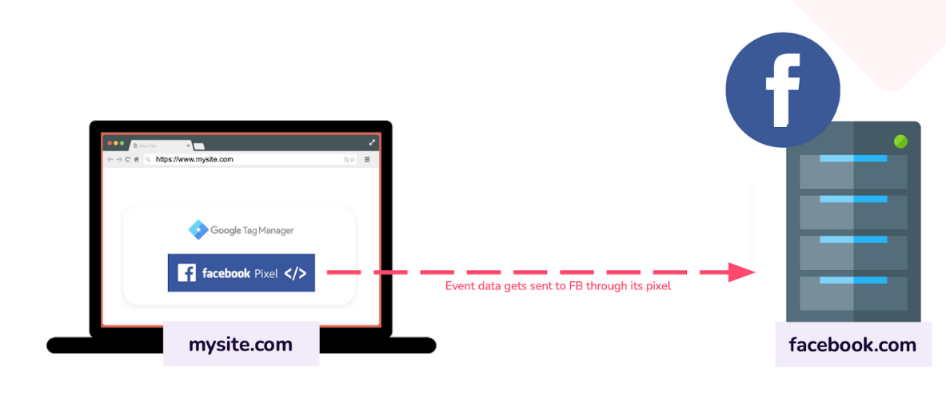
Why do we need conversion API?

It allows us to send data to platforms like FB, to map onsite conversions with the ads, thereby helping us understand user behavioral and segment data along with the conversion data. This further enables us to create better targeting strategies, personalization initiatives and optimization techniques. This way we can strategize to show our ads to people who would be most likely to convert.

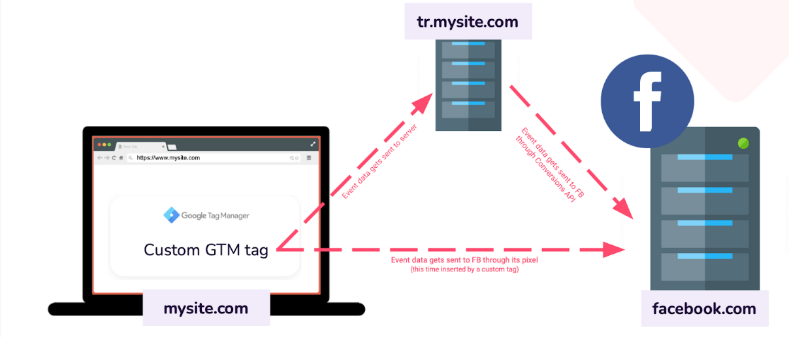
Server-side tagging

With absence of 3rd party cookies, it can’t be done through the traditional method of putting conversion tracking code within a Tag Management tool like GTM or Launch. This is the reason why AdTech platforms have introduced server side tracking APIs. It, as name suggests, consists of sending events through server, rather than client side (through tag management tools).

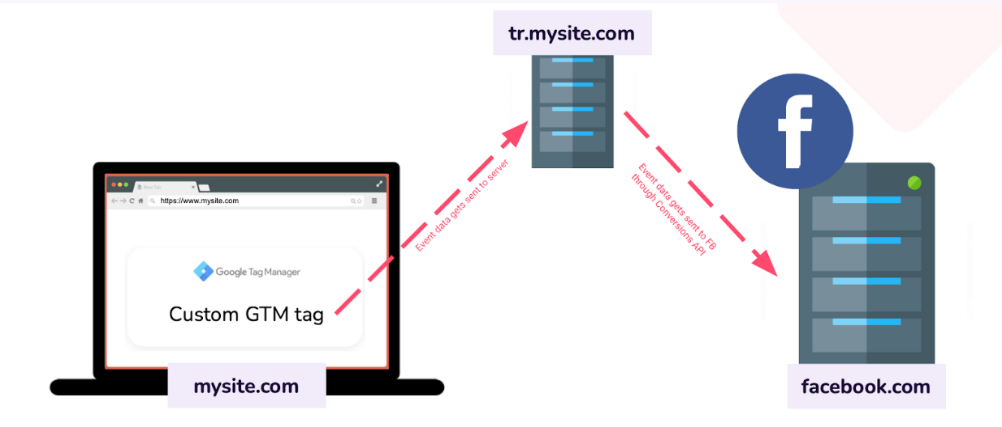
Traditional Method:



Current CAPI methodology:



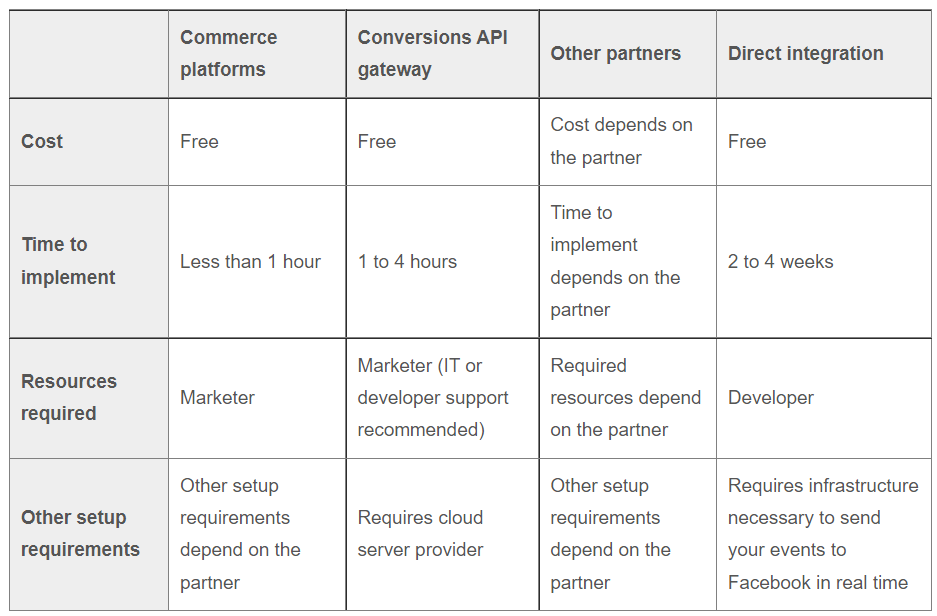
Future method when 3rd party cookies are deprecated:



Typically what you need to implement server side Conversion API:

* Pixel ID
* Business Manager
* Access Tokens

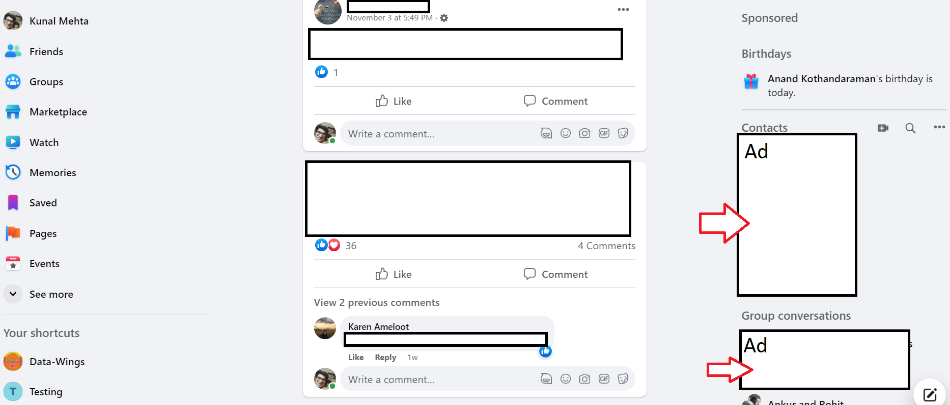
How long does it take to implement the same:



I would also need to understand what do we want to do, the specific use cases we might want to solve, so that we can ideate on the same. I think that replicating a tool like TagHash won’t be an issue for us, but what is the next step for us? That is where we might need to brainstorm and get the ball rolling.

Facebook Platform:

FB or Meta has now become one of the biggest advertising platform, for all kinds of advertisers. Let’s look into how people generally interact with it.



Someone clicks on the ad to land on your platform, and that creates a third party cookie with 2 IDs:

* Click ID
* Browser ID

Since all conversion tracking works on third party cookie, so at the time of conversion, when we want to fire the conversion pixel, this is where integration can break when 3rd party cookies go away.

To address this issue, Meta and other AdTech platforms have launched a server side Conversion API (CAPI)

Conversion API (CAPI) Details:

Here cookie is created on advertisers’ own subdomain, making it a first party cookie. Within this first party cookie, we will be able to find the values of Click ID and Browser ID in the following respective variables:

* \_fbc
* \_fbp

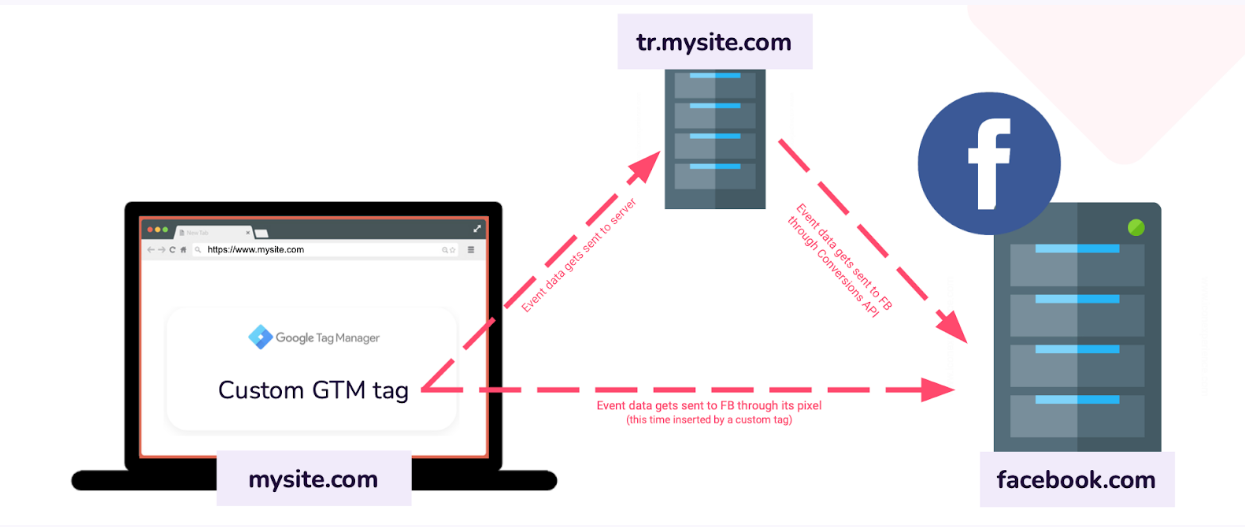
At the time of conversion FB would expect us to pass the information to Meta server along with the following parameters:

* Email: Hashed
* Phone Number: Hashed
* First Name: Hashed
* Last Name: Hashed
* Gender: Hashed
* DOB: Hashed
* City: Hashed
* State: Hashed
* Zip: Hashed
* Country: Hashed
* Client IP Address: Not hashed
* Client User Agent: Not hashed
* Click ID: Not hashed
* Browser ID: Not hashed
* Subscription ID: Not Hashed
* FB Login ID: Not Hashed
* Lead ID: Not hashed

Now, there may be a possibility that not all values are available, so send as much as possible. More values, more accurate values, will help us get a better match rate quality score.

The PII data listed above needs to be hashed at the source itself, and should be passed on to the Meta Conversion API through server side, not dataLayer. Even if it is passed through dataLayer, it has to be passed in dataLayer in the hashed format only.

Current tracking methodology:



For now, Meta will get data both from the cookie and server side conversion API, and will deduplicate data from its own end. If and when it encounters the situation of third party cookies being defunct, like in case of Safari and Firefox, it automatically showcases results from server side information.

What we would need to switch on CAPI:

In addition to the parameters listed in the section above, we will be needing the following:

* Business Manager Account
* Pixel ID
* Access Tokens

Where Tag Hash fits in:

TagHash is a tool which can help us create the connection between advertiser server and Meta server to create a code, that can fetch the parameter information from the advertiser website, and then relay the same to Meta server, once they are provided with the Pixel ID and Access Token. In addition to this, Advertiser’s IT team will also have to add a subdomain provided by Taghash within their folder.

TagHash provides a maintenance free, serverless pipeline to pass data from Advertiser server to the Meta server. In the whole process, TagHash doesn’t store or manipulates any data at their own end.

TagHash claims that their event match rate score is 7 and above consistently, on a band of 1-9, where anything 5 and below is poor, while 8 and above is best.

New issue:

Webkit browsers, especially Safari, has come up with this 7 day cookie expiration window/ In case the IP address of the website that the user is browsing, doesn’t match the IP address of the cookie, then the cookie will expire in 7 days.

It will impact all first party cookie implementations, server side implementation, and will have massive impact on attribution.

High Level arch:

1. A front end to help users (marketers) implement CAPI with no code methodology
   1. Onboarding client 🡪 One time
   2. Onboarding platforms 🡪 Multiple times
   3. Reports 🡪 Role based
2. Input subdomain which will be mapped in the domain of the client
   1. Some homework by the clients that they need to do
   2. Either they give us the name, or we do it programmatically
3. Implementation code will be sitting on our server
4. Create database to help clients input their customer PIIs
5. There might be pipelines, or csv files in a specific template, for initial phase
6. Create governance system, since PII data will be uploaded
7. This PII data will be passed over to Meta and other AdTech tools like Google Ads, through CAPI
8. Reports and match scores should be made available to the clients
9. Create continuous maintenance program

**Google Enhanced Conversion API**

Enhanced conversions for web is a feature that can improve the accuracy of our conversion measurement. It supplements out existing conversion data by sending hashed first-party customer data in a privacy-safe way.

The hashed customer data is compared to hashed customer data of signed-in Google accounts and attributed to ad events such as clicks and video views to help measure our campaign conversions.

Benefits:

* First-party data can be provided after the conversion event: Unlike other enhanced conversions options like tagging that require us to send first-party data at the time of the conversion, we can use the API to send hashed first-party data up to 24 hours after the conversion event. This also enables us to locate first-party data from other sources, such a customer database or CRM system.
* More choices for our data source: Using the API enables us to locate first-party data from a variety of sources, such as a customer database or CRM system.

Google Ads API, we can leverage enhanced conversions by sending first-party customer data in the form of conversion adjustments.

In order to implement enhanced conversions using the Google Ads API, we must record conversions using a regular conversion tracking tag on our website. After a conversion is recorded, we can then send hashed first-party data (such as email address, phone number, name, and/or home address) via an API connection up until 24 hours after the conversion.

The hashed data we send us is matched against Google logged-in user data and a conversion is reported in our account.

We have to use SHA-256 to hash customer data.

Parameters used for Google’s Enhanced conversion API

1. order\_id (required)
2. adjustment\_type must be ‘ENHANCEMENT’
3. The conversion\_action is the resource name of a ConversionAction with a type of WEBPAGE.
4. The user\_identifiers collection must contain between one and five identifiers.
5. The user\_identifier\_source of each identifier is optional.
6. A gclid\_date\_time\_pair with a conversion\_date\_time
7. The user\_agent string is optional
8. restatement\_value with adjusted\_value and currency\_code

Normalization and hashing

* Remove leading/trailing whitespaces.
* Convert the text to lowercase.
* Format phone numbers according to the E164 standard.
* Remove all periods (.) that precede the domain name in gmail.com and googlemail.com email addresses.