Enhanced E-Commerce

Development Model Draft

Ver. 1.1 (3 Nov 2017)

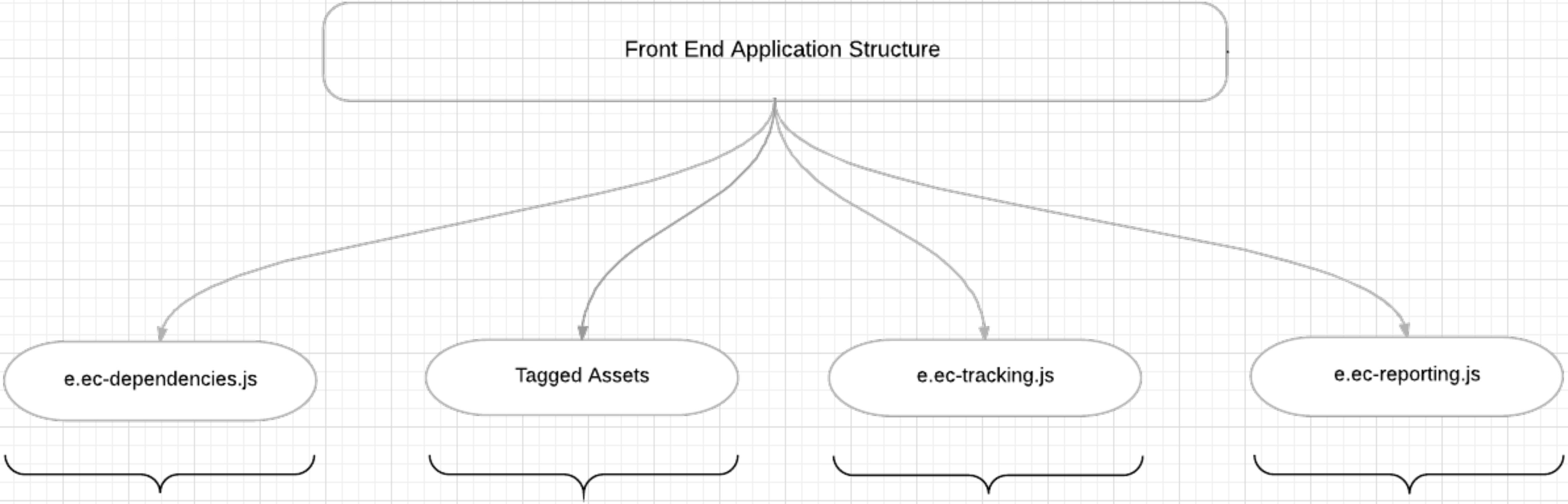
Project Objectives

- Move from Marketo-Forms to Drupal-Forms (For Speed & Conversion)
- Consolidate AF 3-step process into 1-step to stop 70% abandonment
- Fix old bugs: Ex. Appointment booking time zone issues in NZ etc.
- Improvements: Speed, Validation, Tracking,
- Ground preparation for moving away from Marketo
- Enhanced e-comm. Unite web-activity data in GA & lead data in Marketo
- Enhanced e-comm. Assign $ value to each lead

Frontend structure

The following slides explain the modular structure of an enhanced e-commerce application

On the front-end the application consists of 4 components: dependencies, tagged assets, tracking & reporting



Dependencies module

Responsible for loading essential external assets & dependencies

Tagged assets

A list of assets tagged for monitoring via inline HTML

Tracking module

Responsible for real time event listening & interaction monitoring

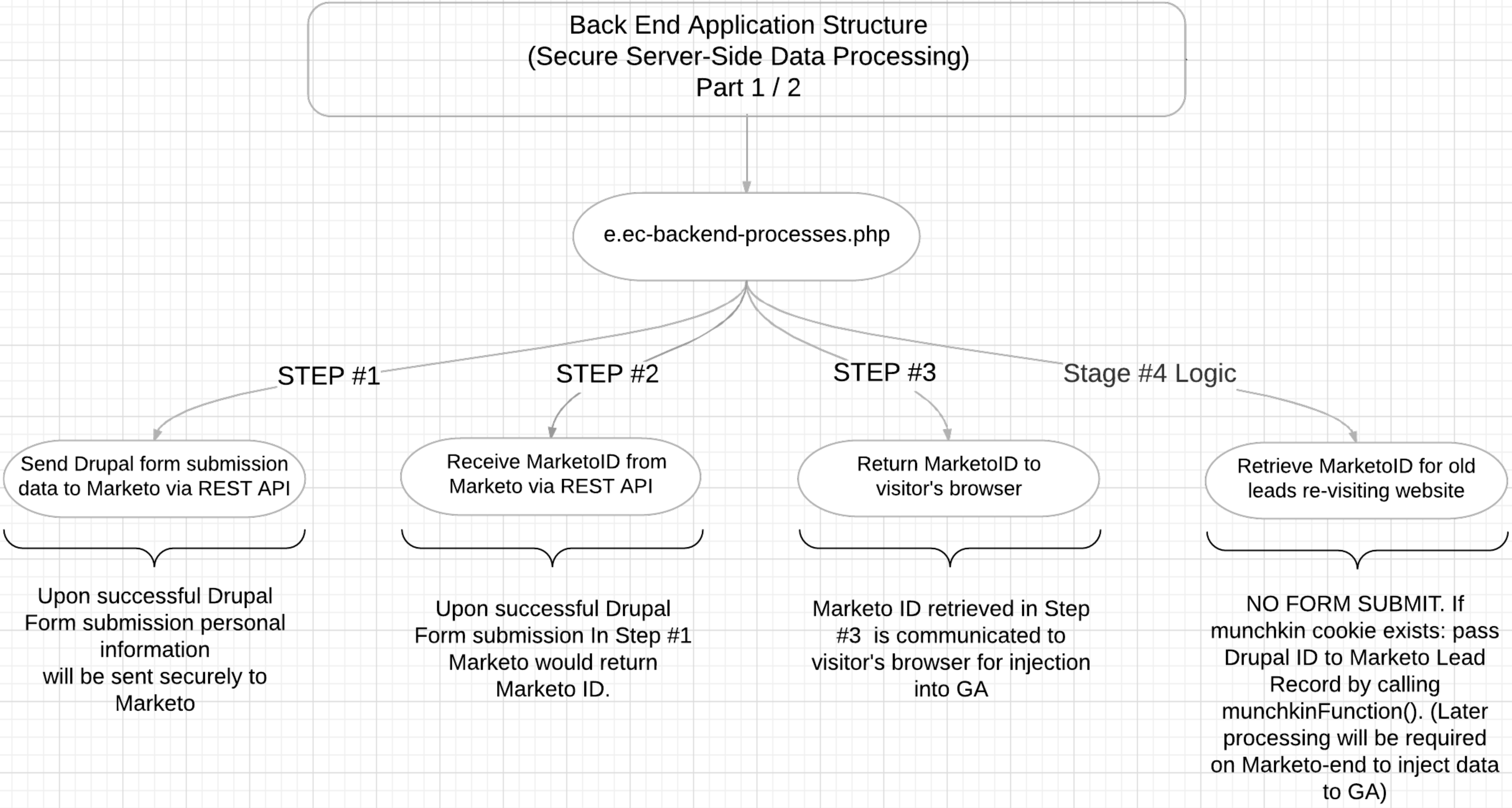
Reporting module

Responsible for real time event listening & interaction monitoring

Backend structure

The following slides explain the back end structure and functionality of an enhanced e-commerce application

On the back end the application consists of 2 PHP Scripts. We use PHP scripts to communicate Drupal Forms submission data to and from Marketo using secure REST API.



Back End Application Structure
(Secure Server-Side Data Processing)
Part 1 / 2

e.ec-backend-processes.php

STEP #1

Send Drupal form submission data to Marketo via REST API

Upon successful Drupal Form submission personal information
will be sent securely to Marketo

STEP #2

Receive MarketoID from Marketo via REST API

Upon successful Drupal Form submission In Step #1 Marketo would return Marketo ID.

STEP #3

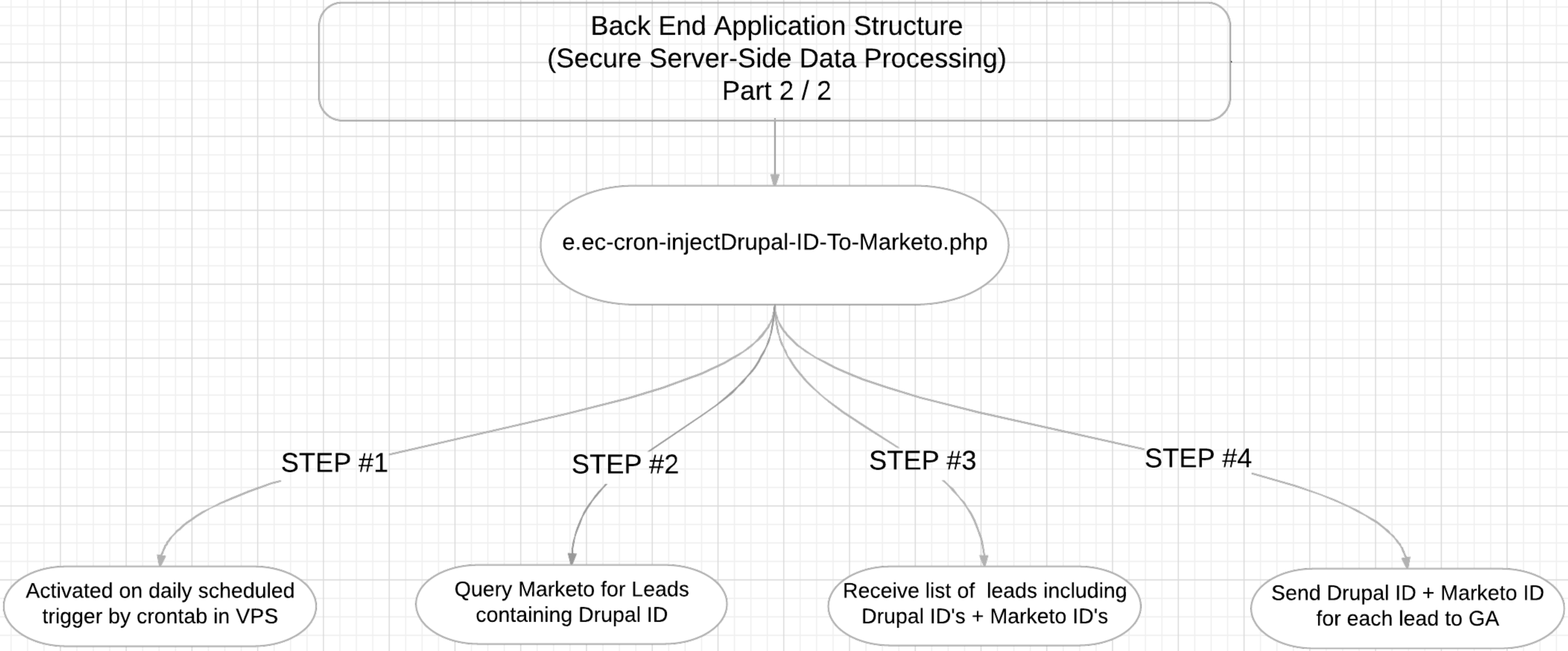
Return MarketoID to visitor's browser

Marketo ID retrieved in Step #3 is communicated to visitor's browser for injection into GA

Stage #4 Logic

Retrieve MarketoID for old leads re-visiting website

NO FORM SUBMIT. If munchkin cookie exists: pass Drupal ID to Marketo Lead Record by calling munchkinFunction(). (Later processing will be required on Marketo-end to inject data to GA)



Back End Application Structure
(Secure Server-Side Data Processing)
Part 2 / 2

e.ec-cron-injectDrupal-ID-To-Marketo.php

STEP #1

Activated on daily scheduled trigger by crontab in VPS

STEP #2

Query Marketo for Leads containing Drupal ID

STEP #3

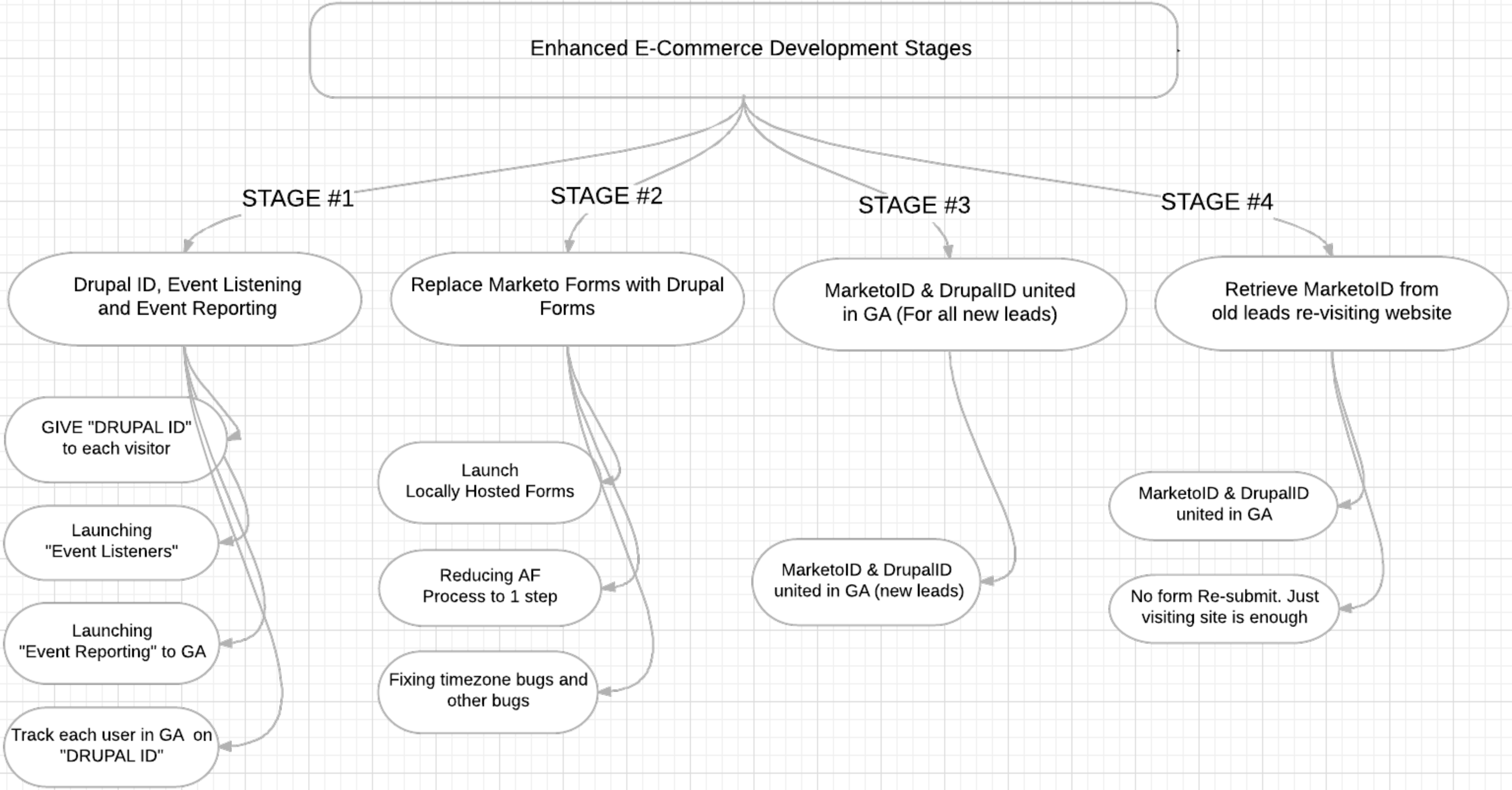
Receive list of leads including Drupal ID's + Marketo ID's

STEP #4

Send Drupal ID + Marketo ID for each lead to GA

Project Development Stages

The following slides explain the 4 stages of project delivery



Enhanced E-Commerce Development Stages

STAGE #1

Drupal ID, Event Listening and Event Reporting

- GIVE "DRUPAL ID" to each visitor
- Launching "Event Listeners"
- Launching "Event Reporting" to GA
- Track each user in GA on "DRUPAL ID"

STAGE #2

Replace Marketo Forms with Drupal Forms

- Launch Locally Hosted Forms
- Reducing AF Process to 1 step
- Fixing timezone bugs and other bugs

STAGE #3

MarketoID & DrupalID united in GA (For all new leads)

- MarketoID & DrupalID united in GA (new leads)

STAGE #4

Retrieve MarketoID from old leads re-visiting website

- MarketoID & DrupalID united in GA
- No form Re-submit. Just visiting site is enough

Development Stage 1:

Drupal ID, Event Listening and Event Reporting

Upon completion of the first development stage the website will automatically assign a unique ID to each visitor. Stage 1 completion will allow us to gain the ability to tag any element on the website (Images, Buttons, Forms, Pages, Promotions etc.) to record the following: visitor's view of a tagged element (impressions), element clicks, user intention to fill in the Drupal forms (User starts typing inside a form field), user initiating form submission and successful form submissions. All these actions will be tied to a unique ID (Drupal ID) and pushed to GA

Development Stage 2:

Replacing Marketo Forms with Drupal Forms

Upon completion of the second development stage the use of Marketo embedded forms will be discontinued. We will start using Drupal forms. This stage will fix a multitude of bugs (such NZ Time zone issues). This stage will also consolidate our existing 3-step booking process into a single step (tackling an existing 70% AF drop-off rate). In addition the above, upon elimination of embedded Marketo Forms we can expect a significant AF loading speed improvements.

Development Stage 3:

Marketo ID + Drupal ID are united in GA (New Leads)

Upon completion of the third development stage, the moment form submission is inituated by the user, the application will connect to Marketo on the back-end and retrieve user's Marketo ID. Market ID will then be sent to Google Analytics to be merged will all pre-existing Google Analytics data and user's Drupal ID (Unique ID) that was already generated and pushed to GA (Stage 1)

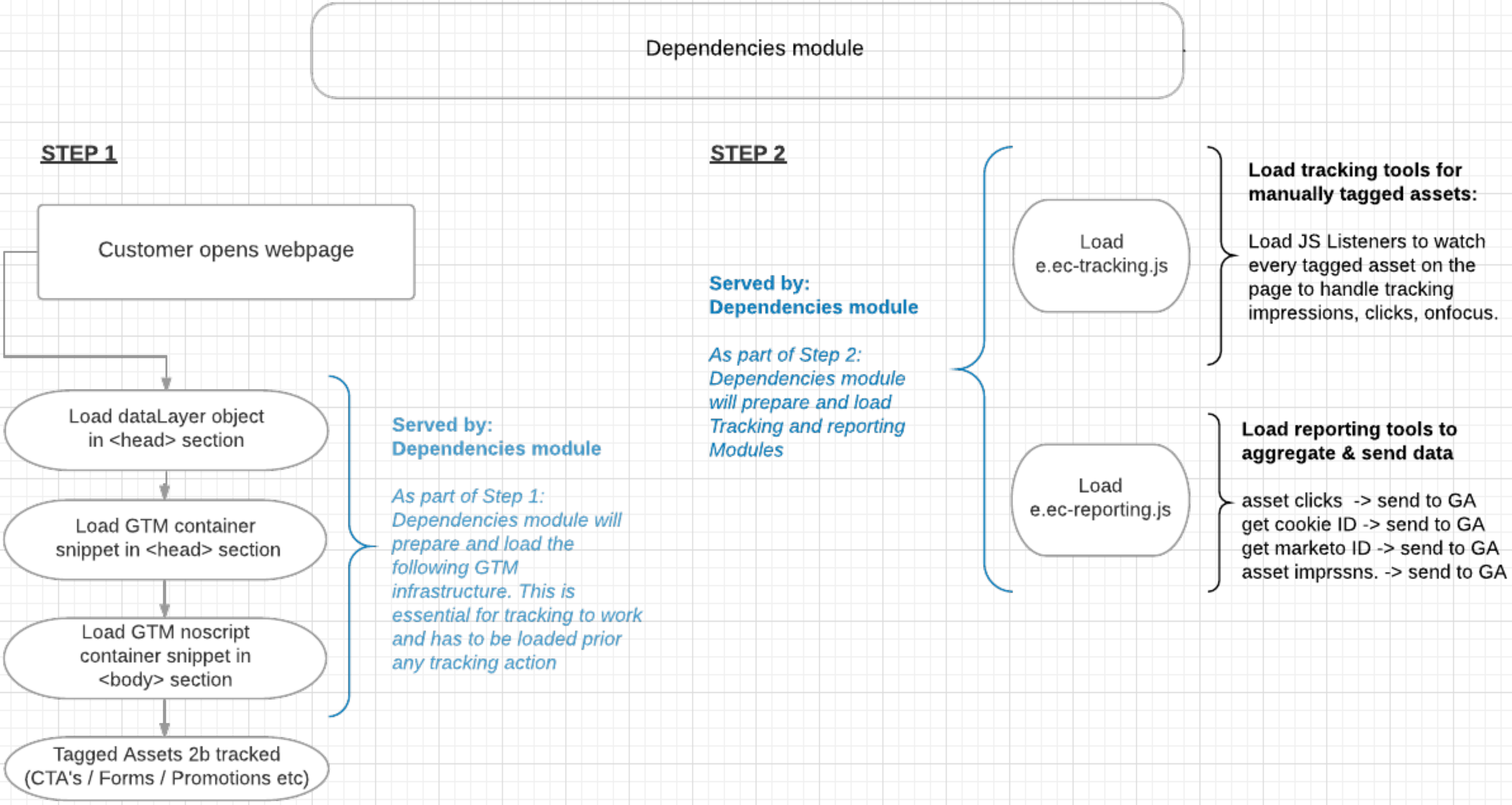
Development Stage 4:

Marketo ID pushed to GA for For leads that already exist in Marketo (To be processed without form submission)

The objective of development Stage 4 is to pass lead’s pre-existing Marketo ID to Google Analytics upon website re-visit. This process will happen behind the scenes and will not require form submission. This reply on Munchkin Tracking code for this to happen. We pass Drupal ID into Marketo using Munchkin API and then querying Marketo once per day via REST API for to retrieve all leads that contain Drupal ID. Upon retrieval we then send Marketo ID + Drupal ID into Google Analytics by simulating Data Layer Push.

Front End Logic & Functions

The following slides explain the functionality of 4 front end modules



Dependencies module

STEP 1

Customer opens webpage

Load dataLayer object in <head> section

Load GTM container snippet in <head> section

Load GTM noscript container snippet in <body> section

Tagged Assets 2b tracked (CTA's / Forms / Promotions etc)

Served by: Dependencies module

As part of Step 1: Dependencies module will prepare and load the following GTM infrastructure. This is essential for tracking to work and has to be loaded prior any tracking action

STEP 2

Served by: Dependencies module

As part of Step 2: Dependencies module will prepare and load Tracking and reporting Modules

Load e.ec-tracking.js

Load tracking tools for manually tagged assets:

Load JS Listeners to watch every tagged asset on the page to handle tracking impressions, clicks, onfocus.

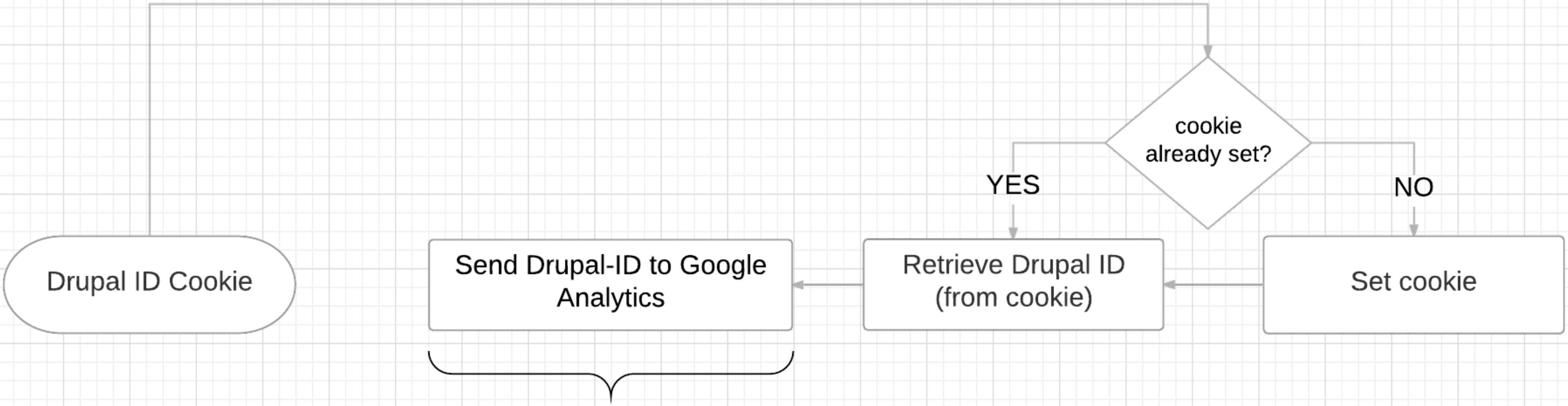
Load e.ec-reporting.js

Load reporting tools to aggregate & send data

asset clicks -> send to GA
get cookie ID -> send to GA
get marketo ID -> send to GA
asset imprssns. -> send to GA

Tracking module

PART 1 / 2 - Generating "Drupal ID"



cookie already set?
YES
NO
Drupal ID Cookie
Send Drupal-ID to Google Analytics
Retrieve Drupal ID (from cookie)
Set cookie

This logic is handled by reporting module

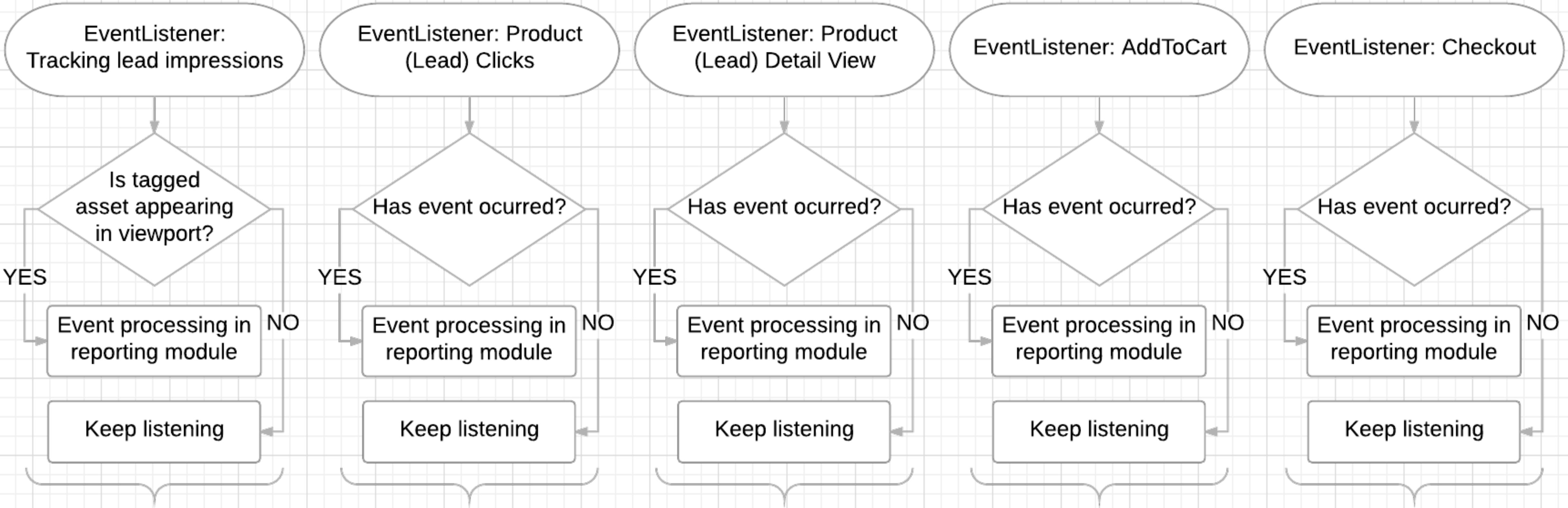
- Diagram demonstrates the logic of generating & assigning custom-ID's to each new visitor. (Later referred to as "Drupal ID").

- Proposed method ensures our independence from Marketo due to Marketo-ID not being our primary identifier in GA.

- Upon form submission, app retrieves Marketo ID for each new lead and associates Marketo ID with visitors Drupal ID in GA.

Reporting module

PART 2 / 2 - Initialising "Event Listeners"



EventListener: Tracking lead impressions

Is tagged asset appearing in viewport?

YES

Event processing in reporting module

NO

Keep listening

EventListener: Product (Lead) Clicks

Has event ocurred?

YES

Event processing in reporting module

NO

Keep listening

EventListener: Product (Lead) Detail View

Has event ocurred?

YES

Event processing in reporting module

NO

Keep listening

EventListener: AddToCart

Has event ocurred?

YES

Event processing in reporting module

NO

Keep listening

EventListener: Checkout

Has event ocurred?

YES

Event processing in reporting module

NO

Keep listening

Listening for Lead Impressions & Position

For all tagged assets, an event listener is created. When event condition is met reporting module sends data to GA

Listening for Clicks Events

For all tagged assets, an event listener is created. When event condition is met reporting module sends data to GA

Listening for lead form views

For all tagged assets, an event listener is created. When event condition is met reporting module sends data to GA

Listening for user starting to fill out a Drupal Form field

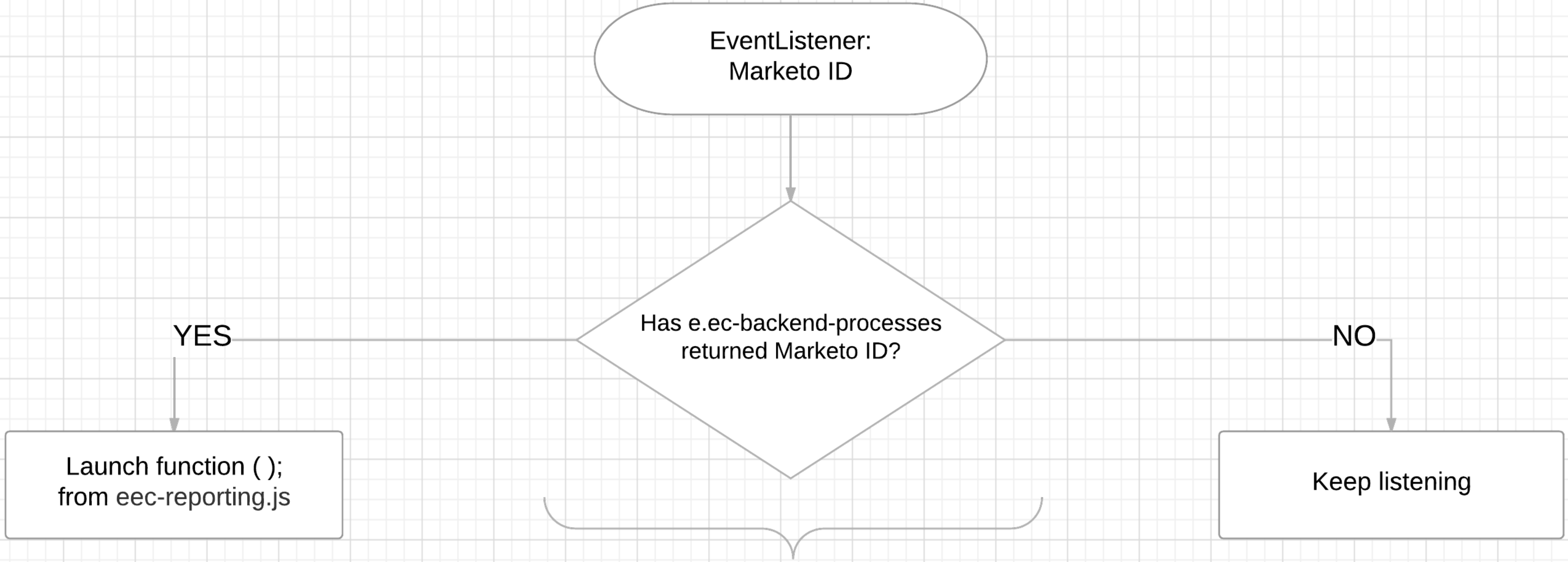
For all tagged assets, an event listener is created. When event condition is met reporting module sends data to GA

Listening for Checkout Initialise & Checkout Success

For all tagged assets, an event listener is created. When event condition is met reporting module sends data to GA

Tracking module

Listening for Marketo ID (to be returned by PHP Script)



EventListener:
Marketo ID

Has e.ec-backend-processes returned Marketo ID?

YES

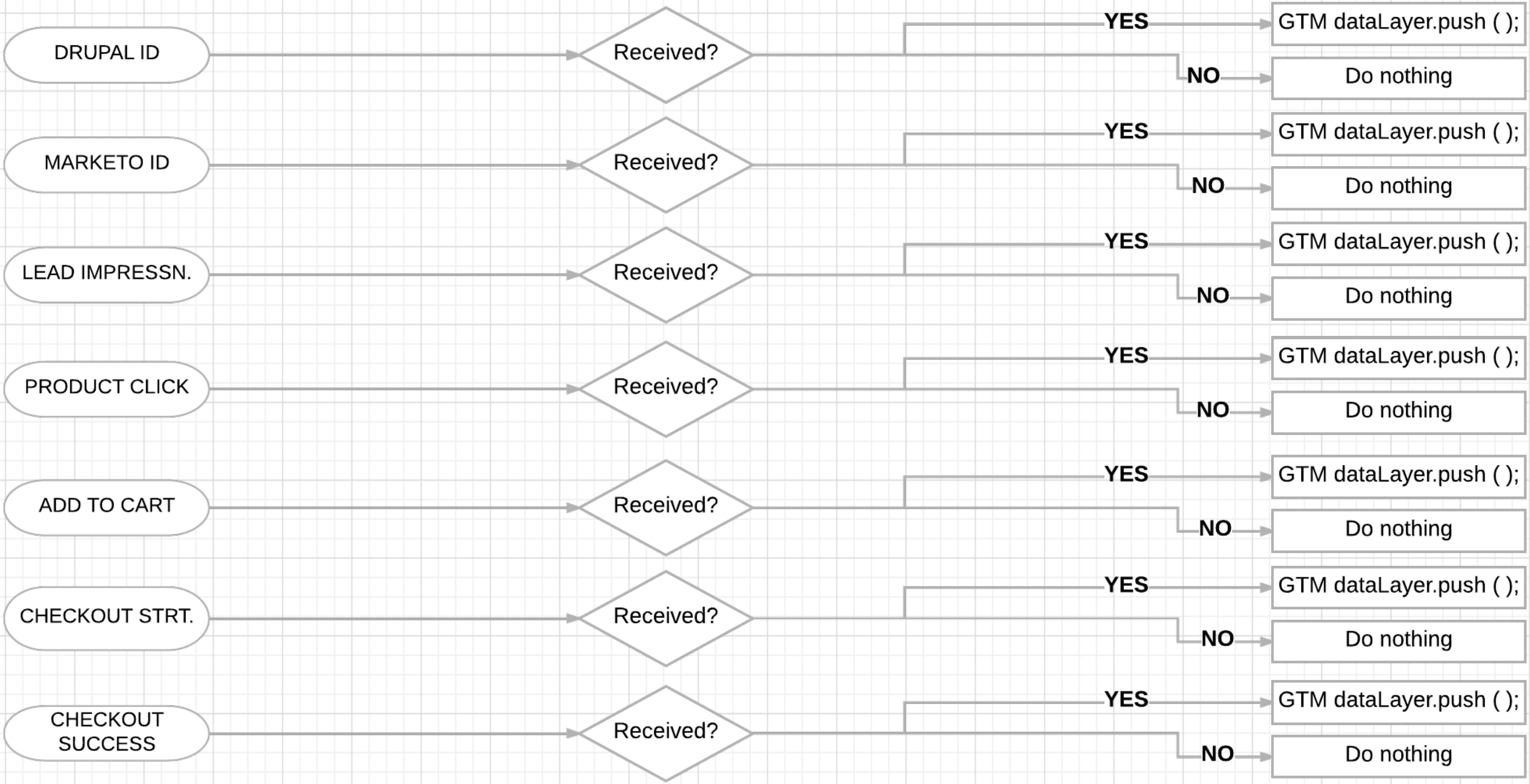
Launch function ( );
from eec-reporting.js

NO

Keep listening

Marketo ID will be returned on successful Drupal Form Submit
Upon Marketo ID being returned, it will be immediately sent to reporting module for processing: to be sent to Google Analytics (and associated with visitor Drupal ID)

Reporting module



DRUPAL ID

Received?

YES

GTM dataLayer.push ( );

NO

Do nothing

MARKETO ID

Received?

YES

GTM dataLayer.push ( );

NO

Do nothing

LEAD IMPRESSN.

Received?

YES

GTM dataLayer.push ( );

NO

Do nothing

PRODUCT CLICK

Received?

YES

GTM dataLayer.push ( );

NO

Do nothing

ADD TO CART

Received?

YES

GTM dataLayer.push ( );

NO

Do nothing

CHECKOUT STRT.

Received?

YES

GTM dataLayer.push ( );

NO

Do nothing

CHECKOUT SUCCESS

Received?

YES

GTM dataLayer.push ( );

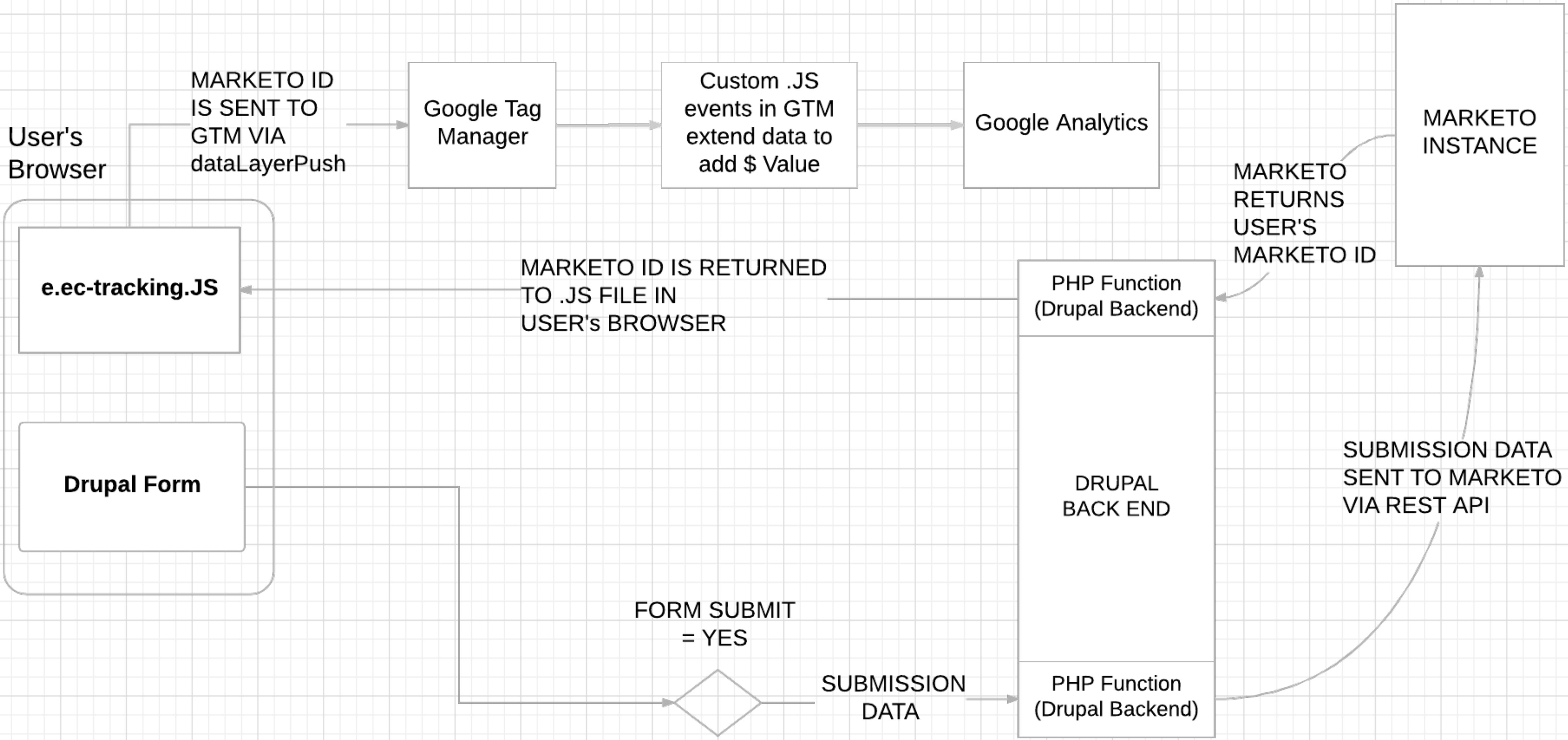
NO

Do nothing

Drupal Form.Submit Data-Flow

The following slides explain the data-flow occurring between Drupal, Marketo & GA when AF form is submitted.

Data Flow:
Retrival of Marketo ID when Drupal Form is submitted



User's Browser

e.ec-tracking.JS

Drupal Form

MARKETO ID IS SENT TO GTM VIA dataLayerPush

Google Tag Manager

Custom .JS events in GTM extend data to add $ Value

Google Analytics

MARKETO INSTANCE

MARKETO RETURNS USER'S MARKETO ID

MARKETO ID IS RETURNED TO .JS FILE IN USER's BROWSER

PHP Function (Drupal Backend)

DRUPAL BACK END

PHP Function (Drupal Backend)

SUBMISSION DATA SENT TO MARKETO VIA REST API

FORM SUBMIT = YES

SUBMISSION DATA

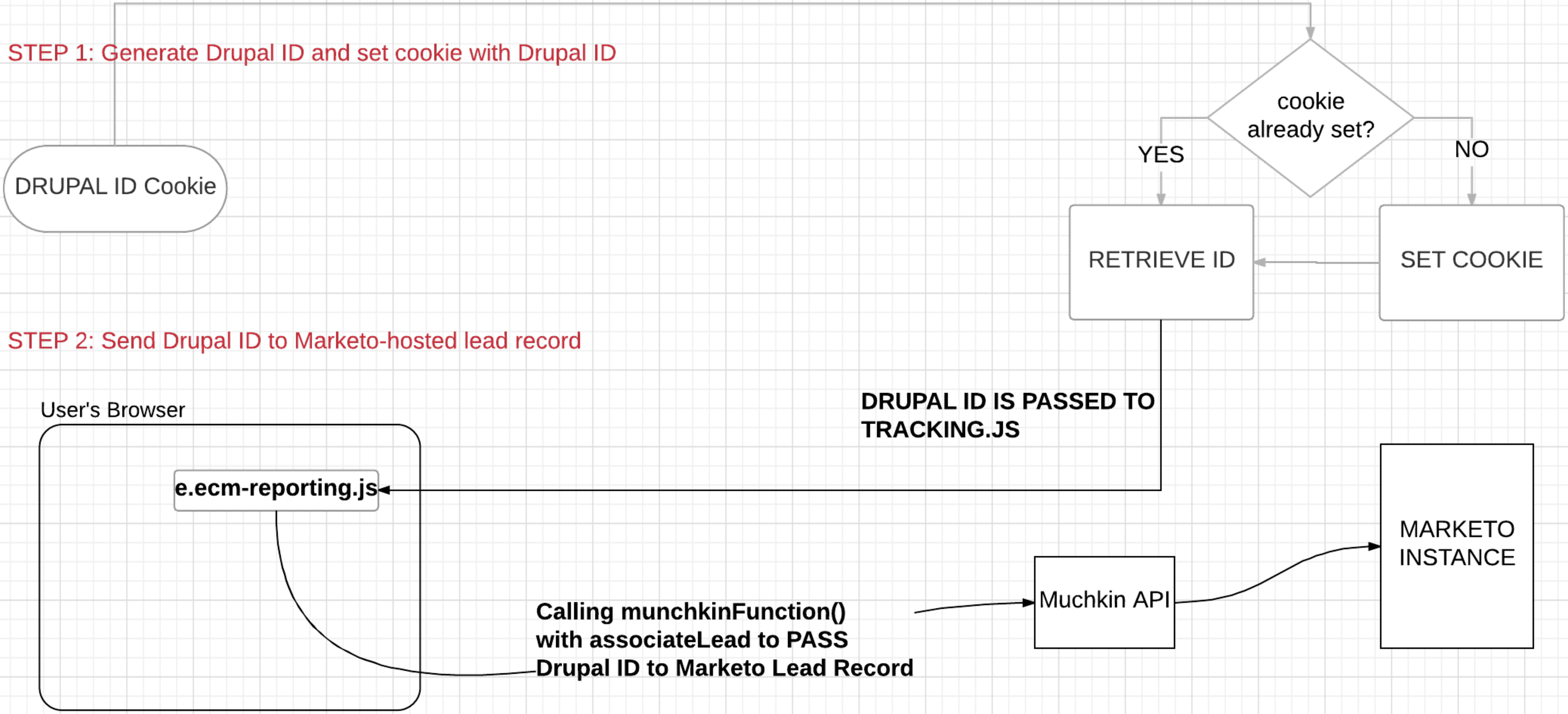
Stage 4 Data-Flow

The following slides explain the data-flow occurring between Drupal, Marketo & GA when pre-existing Marketo lead revisits the website.

The objective of development Stage 4 is to pass lead's pre-existing Marketo ID to Google Analytics upon website revisit. This process will happen behind the scenes and will not require form submission.

The process consists of 2 steps: first step utilizes Marketo Muchkin API to pass newly generated Drupal ID into an existing Lead record inside Marketo. The second step relies to php-script triggered by cron job hosted by Digital Pacific VPS. The script, when triggered will connect to Marketo via REST API, query lead records containing Drupal ID, pull their data and then conect to GA using Universal GA API to add Marketo ID to GA.

Stage 4 Data Flow:
Retrival of Marketo ID for pre-existing leads
PART 1 / 2 (Place Drupal ID into Marketo lead record)



STEP 1: Generate Drupal ID and set cookie with Drupal ID

DRUPAL ID Cookie

cookie already set?

YES

NO

RETRIEVE ID

SET COOKIE

STEP 2: Send Drupal ID to Marketo-hosted lead record

User's Browser

e.ecm-reporting.js

DRUPAL ID IS PASSED TO TRACKING.JS

Calling munchkinFunction() with associateLead to PASS Drupal ID to Marketo Lead Record

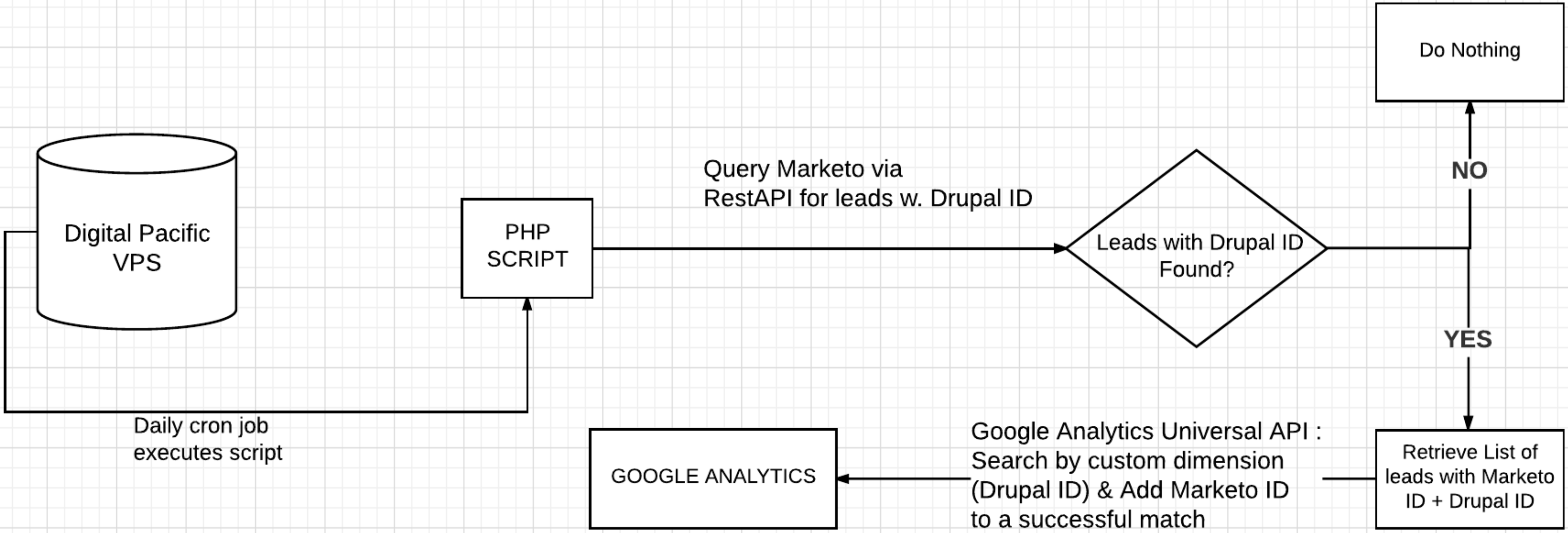
Muchkin API

MARKETO INSTANCE

Stage 4 Data Flow:
Retrival of Marketo ID for pre-existing leads
PART 2 / 2 (Place Drupal ID into Marketo lead record)

STEP 3:

PHP Script triggered by cron job accesses Marketo API,
to generate list of leads containing Drupal ID. Script then simulated dataLayer.Push to send Drupal ID + Marketo ID to Google Analytics


Digital Pacific VPS
PHP SCRIPT
Query Marketo via RestAPI for leads w. Drupal ID
Leads with Drupal ID Found?
NO
Do Nothing
YES
Retrieve List of leads with Marketo ID + Drupal ID
Google Analytics Universal API : Search by custom dimension (Drupal ID) & Add Marketo ID to a successful match
GOOGLE ANALYTICS
Daily cron job executes script

Known Constraints

The following slides will explore existing constraints

Known Constraints

Constraint #1

- Synchronizing with Marcus on product naming convention (Product numbers and names)
  Action taken: *Raised with Marcus, Marcus to supply naming convention samples*

Constraint #2

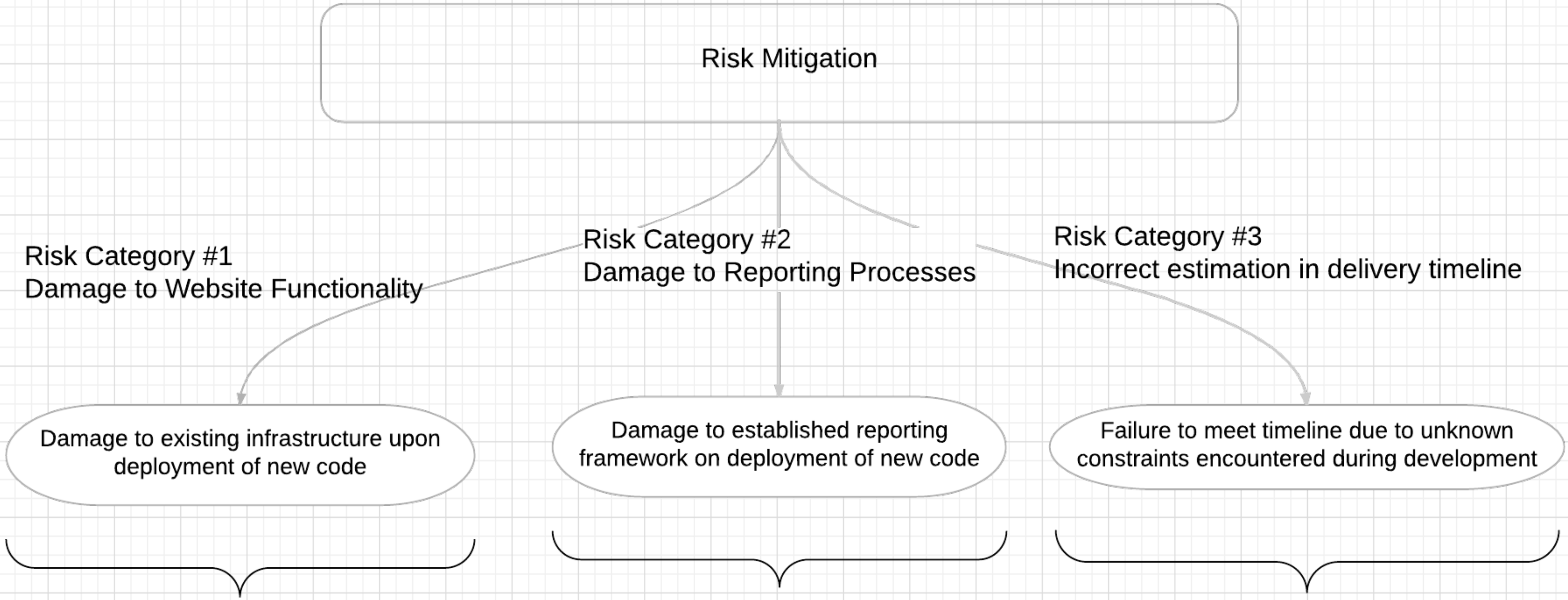
- Assigning $ value to leads that are injected into GA programmatically via GA Universal API is problematic. Data flow will bypass GTM entirely, therefore also bypassing event based JavaScript written in GTM by Marcus that is responsible for assigning $ value to leads.
  Action taken: *Raised with Marcus, Marcus to supply samples / workaround*

Constraint #3

- Providing accurate development timeline estimation for each stage of the project. Time estimation has been provided is ”preliminary” and pending careful technical overview with the US team
  Action taken: *Scheduled meeting with Kory + Tal to discuss unknown constraints that might potentially increase / decrease project development involvement*

Risk mitigation

The following slides will explore 3 categories of risk associated with this project and implemented mitigation strategies for each category



Risk Mitigation

Risk Category #1
Damage to Website Functionality

Damage to existing infrastructure upon deployment of new code

STATUS: Mitigated

Risk mitigated by creating a secondary staging server specifically for this project. In addition we have implemented stepwise deployment strategies & effective rollback mechanisms for immediate recovery options in case of new systems unexpected behavior

Risk Category #2
Damage to Reporting Processes

Damage to established reporting framework on deployment of new code

STATUS: Mitigated

A new feature GA Universal (Ecommerce-TAB) will be utilised for this project. Existing tracking methods and data will not be affected

Risk Category #3
Incorrect estimation in delivery timeline

Failure to meet timeline due to unknown constraints encountered during development

STATUS: In Progress

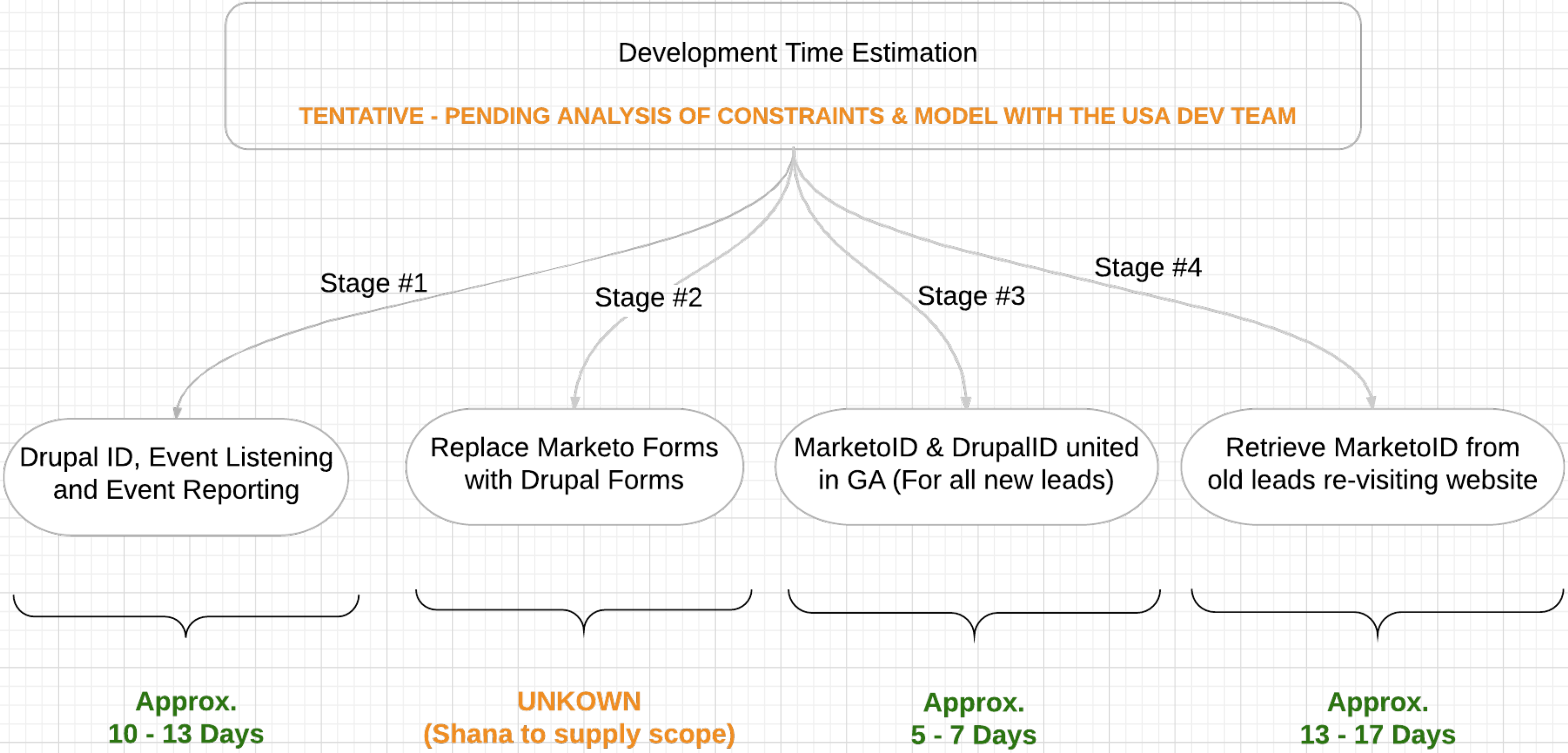
To mitigate this risk a call has been scheduled with the US team to go over development model and evaluate constraints that are currently unknown and might increase the amount of time required to complete this project

Project Delivery Timelines

Exploring development effort required per project stage

Development time is an estimate and pending model review and constraints analysis with US and pending constraints resolution by Marcus. Development time is given in days.

An assumption is made that 15-20% of web developer resource time can be allocated for BAU & emergency tasks during completion of this project



Development Time Estimation

TENTATIVE - PENDING ANALYSIS OF CONSTRAINTS & MODEL WITH THE USA DEV TEAM

Stage #1

Drupal ID, Event Listening and Event Reporting

Approx.
10 - 13 Days

Stage #2

Replace Marketo Forms with Drupal Forms

UNKOWN
(Shana to supply scope)

Stage #3

MarketoID & DrupalID united in GA (For all new leads)

Approx.
5 - 7 Days

Stage #4

Retrieve MarketoID from old leads re-visiting website

Approx.
13 - 17 Days