ObservePoint

The Elusive Data Layer

What Is It and Why Do I Need One?

What is a data layer, and what can it actually do for you?

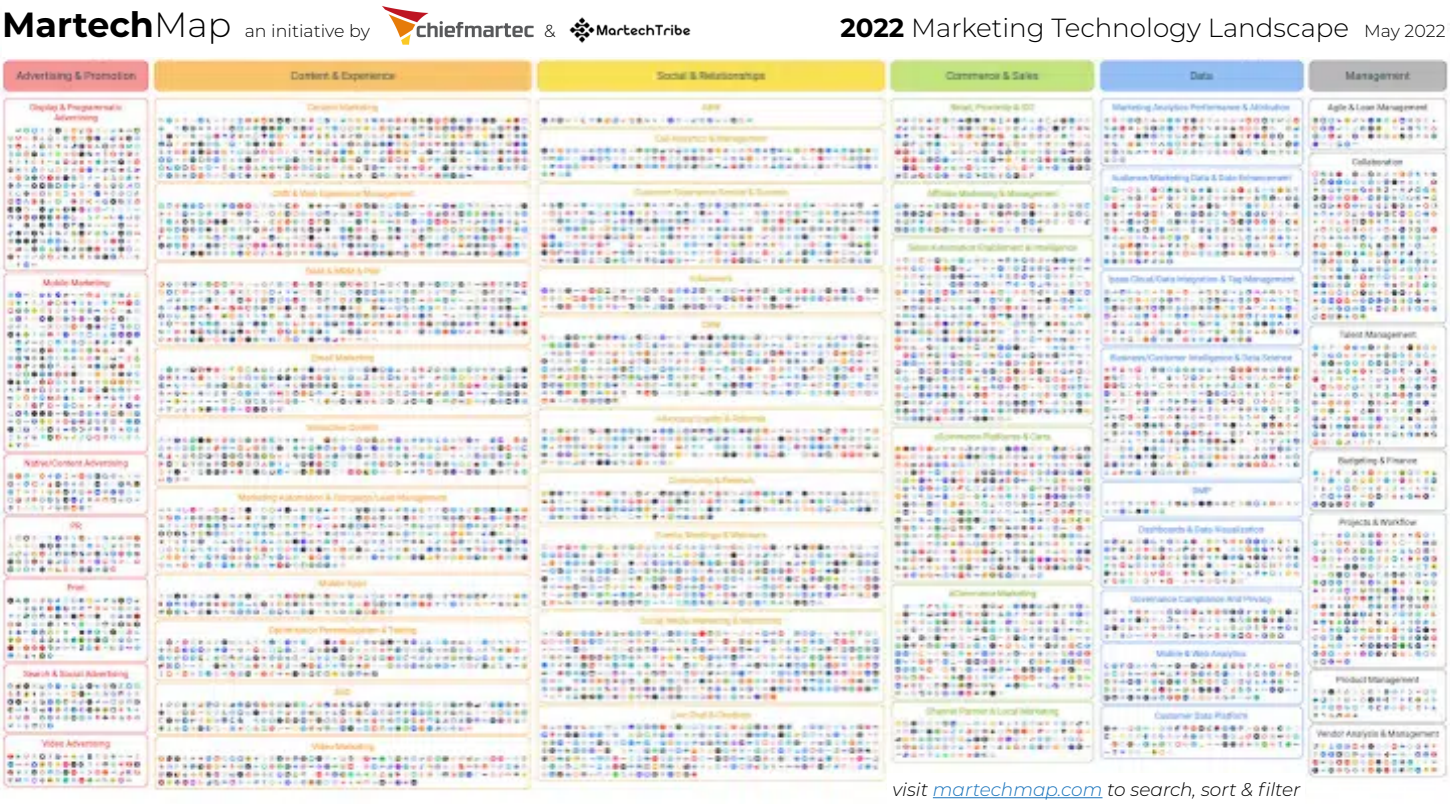
In this tip sheet, we'll walk through the key technical components of applying a data layer and demonstrate how it positively impacts data collection practices.

Hint:

This tip sheet is a comprehensive guide to understanding what a data layer is and how it can benefit your business. For a more advanced guide detailing best practices for data layer implementation, see ***The Data Layer: From Novice to Expert in 2.5 Seconds With Red Bread Lab.***

Why the Data Layer Matters

Ever seen this before?



visit martechmap.com to search, sort & filter

There are more than 9,900 marketing technologies out there, and the number is constantly growing. Of course, your company will never use all of them, but according to **Chiefmartec**, the average enterprise uses an astounding 323 SaaS apps.

These technologies all need data, much of which is common across technologies, such as page name, page type, visitor ID, visitor state, traffic source, and more.

Unfortunately, without a data *layer*, each marketing vendor is responsible for capturing its own data.

With one or two marketing technologies, this may not be too hard to manage—but what about with 10 or 20 or even 323?

A data layer helps you manage all this data across technologies. Let's dive a bit deeper into how it works.

Data Collection Without a Data Layer

Marketing technologies gather data through the use of tags.

Tags can either be hard-coded to a site or deployed dynamically by a tag management system. When a page loads or a user performs an action, variables contained within the tags are populated with values, which trigger responses by the marketing technologies (e.g. initiating a remarketing campaign or gathering analytics data).

A vendor-specific tag will produce a collection of variables that looks something like this:

```
<script>
   s.pageName="Shop>Holiday";
   s.eVar55="Shop";
   s.prop1=s.eVar55;
   s.events="event33,event45";
   s.prop14="guest";
   ...
<script>
```

"So what?" you may ask.

Relying on siloed, vendor-specific data collection has its issues. Here are the problems with this method

These variables only apply to one vendor.

That means you would need to have a process of variable declaration and population for each and every marketing technology. Repeating this process is redundant.

Deploying 10 times the code will slow down your website.

"A one second delay in page response can result in a 7% reduction in conversions".

Each marketing technology will have slightly different definitions of user events.

As a result, the data they collect may be inconsistent across your MarTech stack.

The naming conventions are not very intuitive.

When you have multiple teams across business units trying to work with less-than-intuitive variable names, things can get lost, forgotten or broken.

Data Layer to the Rescue

A data layer is a JavaScript object with a consistent structure that holds all the information you want to collect and then passes that information to a tool, such as an ad or analytics tool.

In other words, a data layer expedites the data collection process by gathering data into one central location and then redistributing it to your various marketing technologies (both first-party and third-party).

By using a data layer to centralize data before it's distributed, you can:

- ☑ Standardize data across channels
- ☑ Ensure greater governance over data collection
- ☑ Increase the extensibility of your technologies

```
var dataLayer = {
“pageTitle” : “Receipt Page”,
“pageURL” : “/pages/checkout/receipt”,
“pageCat” : “Checkout Pages”,
“PageCat2” : “”,
“tranID” : “17658726382”,
“tranTotal” : “34.95”,
“tranTax” : “0.00”,
“tranShipping” : “0.00”,
“tranShippingMethod” : “USPS”,
“tranCurrency” : “USD”,
“tranProds” : “249|398”,
“tranSKUs” : “249-32|398-12”,
“tranProdNames” : “Kids Onsie|Kids Lava
Lamp”,
“tranCategories” : “Kids|Kids”,
“tranPayMethod” : “VISA”,
“visitorType” : “RETURN”,
“visitorState” : “Logged In”,
“visitorFirstPurchDate” : “20111205”,
“visitorFirstProds” : “822”
}
```

A Simple Data Layer

Remember what a vendor-specific tag would look like?

```
<script>
      s.pageName=“Shop>Holiday”;
      s.eVar55=“Shop”;
      s.prop1=s.eVar55;
      s.events=”event33,event45”;
      s.prop14=”guest”;
      ...
<script>
```

Now, compare the following data layer object:

```
<script>
dataLayer=[{
  ‘pageCategory : Shop’,
  ‘categoryType : Holiday’,
  ‘memberStatus : Guest’
}];
</script>
```

“How is this better?” you may ask. Let me show you.

1. The key-value pairs within the data layer are strings with intuitive labels. This means less confusion among developers and across business units.

2. The variables are vendor-agnostic. You only have to deploy this once, and then delineate within your third-party tool which variable corresponds with each metric. You might think to yourself, “Well this is just reversing the process! Instead of writing 10 different codes on 1 site, I have to configure 10 tools to match my 1 code.” That’s true, but consider this:

a. The data layer puts you in control, allowing you to work with intuitive naming conventions without having to learn vendor-specific coding.

b. If you use a tag management solution, you can configure all of those technologies within the TMS itself.

3. Using a data layer allows you to standardize data definitions because data is only gathered once.

4. Now you have 1 code instance instead of 10, which means you have a faster website.

So what’s the ultimate result?

- ☑ Improved data quality
- ☑ Enhanced customer experiences
- ☑ Increased operational efficiency

Need I say more?

Populating Data Layer Variables & Values

Why Implement a Data Layer

One of the key arguments in favor of implementing a data layer is the way in which its values are populated, which allows you to 1) standardize data collection practices, and 2) avoid using DOM-scraping.

DOM-scraping is a method in which marketing tags collect values from the Document Object Model (DOM)—the HTML structure—using JavaScript selectors (tag name, ids, and classes).

For example, a marketing tag could use JavaScript or jQuery to pull the value from a form field and assign it to a variable:

HTML:

```
<form>
    <input id='form-field'/>
</form>
```

JAVASCRIPT:

```
<script>
    s.pageName = document.getElementBy
    ID('form-field').value;
</script>
```

Eliminate DOM-Scraping

DOM-scraping can be handy, because as long as a website's HTML elements are well identified, values can be easily gathered from a page with some basic knowledge of JavaScript.

But DOM-scraping is also problematic. If website structure changes due to a redesign or new release, analytics script that relies on JavaScript selectors for DOM-scraping may no longer be able to pull the correct values into your variables.

Enter the data layer. The data layer's centralized data is built separately from the DOM, using a combination of three methods:

1. Hard-coding variable values. Variables and values that don't need to be dynamic can be hard-coded into the data layer.

2. Back-end variable population. When using a template-based CMS, values can be pushed from the CMS database into the data layer as the page is being built.

3. Front-end variable population. Using event listeners like onclick within HTML tags allows you to push values into the data layer when an event occurs:

```
<button id='button1' onclick='dataLa-
yer.push({'event': 'button1-click'})'
/>
```

NOTE

Front-end variable population may just seem like DOM-scraping in reverse, pushing data instead of scraping data. But as long as your company establishes protocol around pushing events into the data layer, the marketing technologies will always have the data they need.

Basic Checklist for Implementing a Data Layer

Below are some basic milestones that will need to be discussed among the developers, marketers, and other stakeholders:

- ☑ Decide on data layer structure and naming conventions **(see blog post)**.
- ☑ Develop and deploy code to populate the key-value pairs:
    a. Hard-code
    b. Back-end
    c. Front-end
- ☑ Remove the vendor-specific code from your website pages, templates, or header.
- ☑ Update variable documentation, mapping your data layer elements to business/vendor require ments.
- ☑ Perform **regular audits** for data layer quality assurance.

To learn more specifics about implementing a data layer, check out the on-demand webinar:

"How to Ensure a Healthy Data Layer"
by Napkyn Analytics and ObservePoint

or read our eBook:

The Data Layer: From Novice to Expert in 2.5 Seconds With Red Bread Lab.

ObservePoint & the Data Layer

ObservePoint automatically validates and monitors analytics and digital marketing technologies across websites and apps, ensuring functionality and helping enterprises be more efficient and confi- dent in their data-driven decisions. ObservePoint's platform can also be used to audit and validate that your data layer is appropriately formatted and loading correctly.

You've made a large investment in MarTech. Don't you want to make sure that it's all working as intended? With ObservePoint you can shave down on QA time and rest assured that the data you have is reliable.

SCHEDULE DEMO