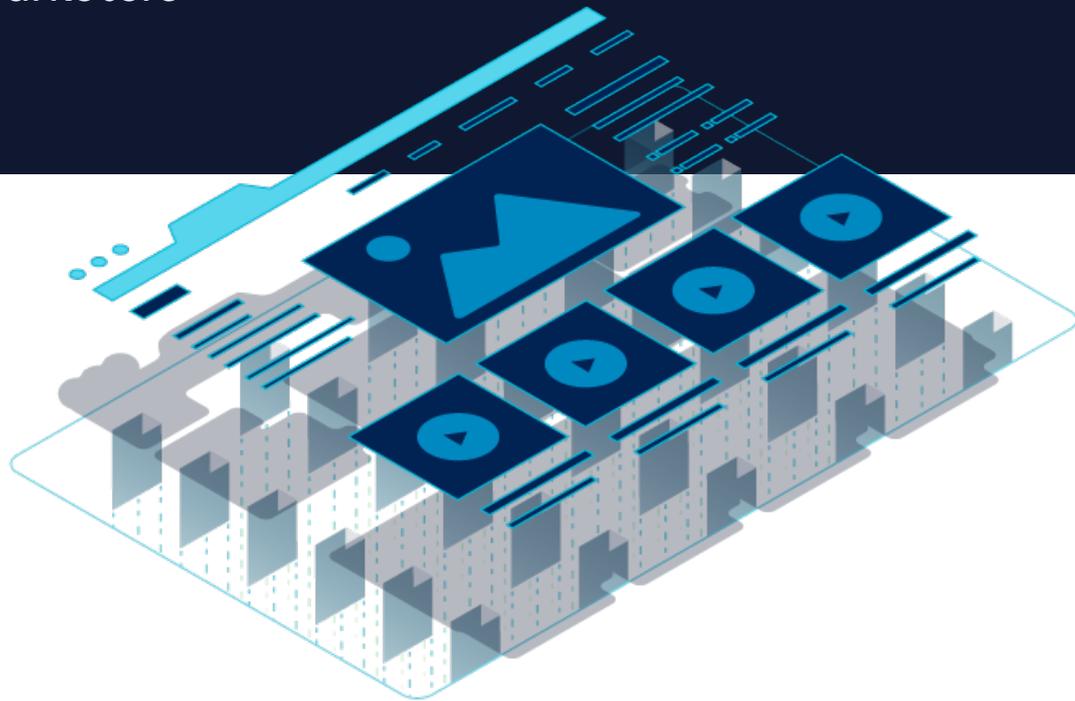


# How to use Single Page Applications with a CMS

CMS + SPA = Happy  
Developers and Marketers



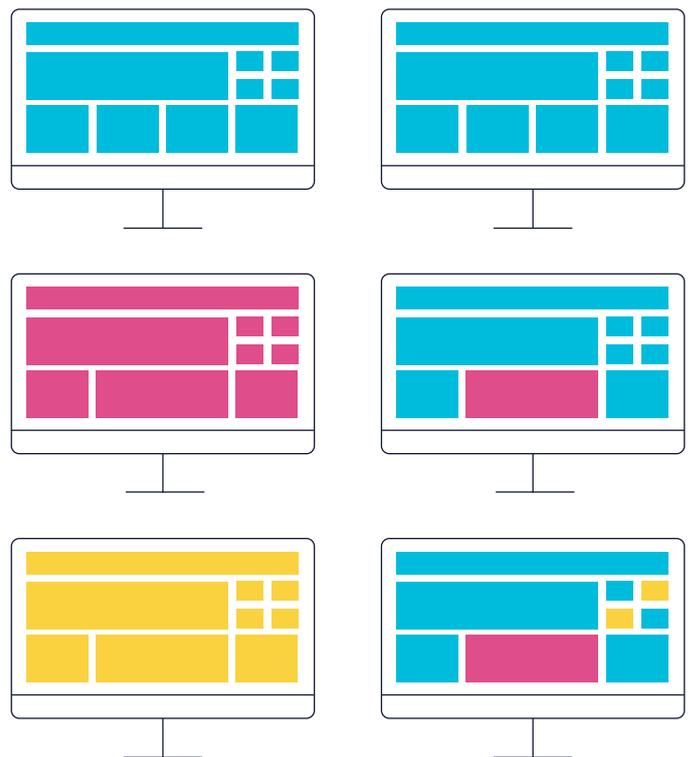
Single Page Applications (SPAs) are a great tool for making incredibly engaging and unique experiences for your users. Gmail, Google Maps, AirBNB, Netflix, Pinterest, Paypal, and many more are using SPAs to build a fluid, scalable experience. However, in the past SPAs have left marketers out in the dark when it comes to managing content. Luckily, it's now possible to pair your SPA with the right CMS to give both developers and marketers the level of control they need.

## SPA 101

The core principle of an SPA is that it is a single page (hence the name) where a lot of information stays the same and only a few pieces need to be updated at a time. For example, when you browse through your email you'll notice that not much changes during navigation - the sidebar and header remain untouched as you go through your inbox. The SPA only sends what you need with each click, and your browser renders that information. This is different to a traditional page load where the server re-renders a full page with every click you make and sends it to your browser.

This piece by piece, client side method makes load time much faster for users and makes the amount of information a server has to send a lot less and a lot more cost efficient. A win-win.

Along with the efficiency, SPAs are a big hit among developers because that separation of back-end services and front-end display means that building a new front-end can be done much quicker, without disturbing critical back-end functionality. It also allows developers to work in the language they prefer. Many SPA frameworks, such as Angular, React, Ember, and Vue are JavaScript but the beauty of that decoupled setup between back and front end means they can work with systems built with other languages. Bottom line, SPAs provide a lot of freedom for developers.



### Traditional

Every request for new information gives you a new version of the whole page.

### SPA

You request just the pieces you need.

# The New Way to Pair CMS & SPA

If SPAs are so great, why isn't everyone using them all the time? Well, historically, the freedom they gave developers came at the cost of losing critical marketing features such as preview, editing, content reuse and personalization.

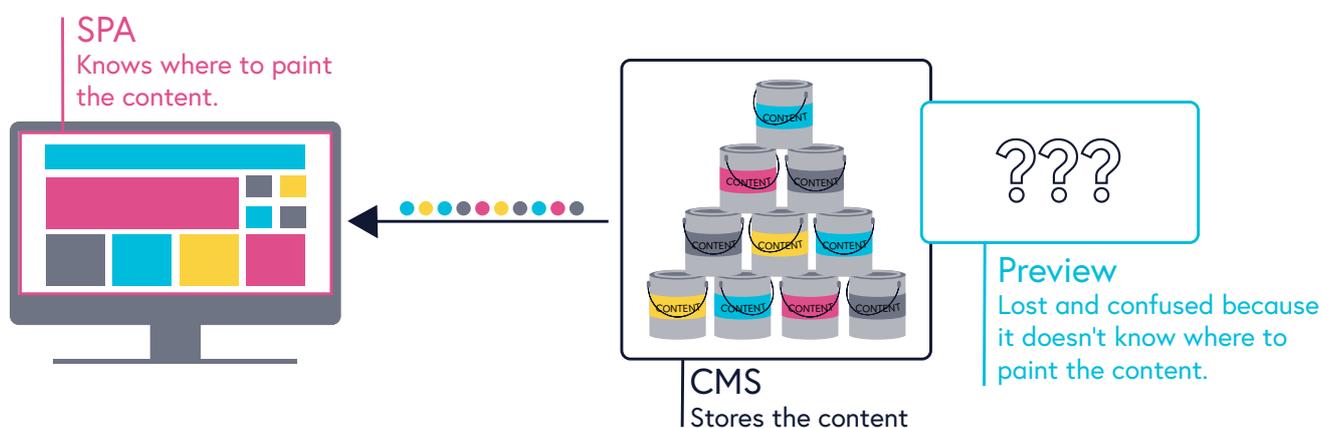
With the decoupled structure SPAs need, the CMS sends content that is presentation-neutral to the SPA and the SPA decides how it looks. This is sometimes called "headless" delivery, because the CMS has no presentation layer - no head. Problem is, the what-you-see-is-what-you-get editing that marketers are used to needs that presentation layer. Without it, marketers are thrown back into the digital-stone age of putting content in a form, clicking publish, and hoping for the best.

Additionally, for any new additions or changes to that SPA, like building a new landing 'page' (known as a route in an SPA), marketers have to go through a developer for each change which creates a bottleneck. Also, because a lot of CMSs do their personalization page based (not helpful in an SPA), or on the client-side (JavaScript personalization rules fighting with the SPA JavaScript) relevant content delivery has been very difficult when using SPAs.

So are marketers doomed forever in an SPA world? Of course not, you simply need a CMS that has an architecture ready for SPA use. One that ticks the following boxes:

- API-based
- Decouples content from presentation
- Can provide live preview & editing
- Supports a hybrid setup
- Personalization on the server side

## Traditional Headless Delivery



## BloomReach's SPA Support

We added SPA support directly into our product because developers like SPAs and we think an Enterprise tool should encourage new, efficient ways of working. We figured the best way to give developers the SPA freedom they want is to ensure Marketing gets to keep all the editing features they are accustomed to - and to make it easy to do so.

## Developers Get Flexibility

The BloomReach CMS (brXM) has always had a core separation of content, code, and presentation which already made it an API-friendly CMS. The architecture was designed for reuse and content, data, and structure can be shared cleanly between sites, devices, and 3rd party systems.

Not only can BloomReach dish out content in that API-friendly manner, but it can also dish out the logic behind that content in the same clean way. On the server side, the BloomReach platform can use browsing behavior, 3rd party data, and links between content to determine the right "content kit" to deliver - and pump out this content and logic in a presentation-neutral, JSON format. This is the key to what makes BloomReach so SPA-friendly.

### The solution addresses two main factors:

1. Developers should have a built-in API solution that is framework agnostic.
2. Marketers should be able to edit the way they are used to (drag and drop, adding/deleting new pages, personalization rules) whether that content is displayed in an SPA, a traditional page, or both.

### To accomplish this, the two key additions are:

**Page Model REST API:** Sends the SPA a JSON/HTTP map of the component hierarchy, along with the content to fill those components. It's a generic representation of resources that can be integrated with the framework of your choice.

**Aggregate Page Model:** The map that the Page Model API sends. It contains the page identifier, follow-up links, page representation (components and domain specific models), and content models.

## Hybrid Support

Many companies use SPAs on one part of their site while keeping other parts as a more traditional server-side page. BloomReach's separation of content and presentation makes this possible. Another key aspect of this hybrid support is the ability to link within and between SPAs and traditional pages. With BloomReach, this is handled by the ability to map each piece of content to an individual URL. When creating content, you'll have a choice to make an "internal" or "external" link. An internal link means the content can be represented within the current SPA, and the SPA will only do a partial page update instead of making a full page request. An external link is used to navigate to content outside of the SPA.

This linking is communicated through the Page Model API, which outputs links for:

- The current URL (the page)
- Per component
- Per component model (eg a menu)
- For each content item
- For rich content links

For example, let's say your blog is done via an SPA while your homepage is a traditional server side page. In the setup, you can say that everything below `"/blog"` has the same application ID so is part of the blog app, aka an inherited child item. Now if you are linking

between blogs that is an "internal" link and your SPA will only call for the new blog text and images. Linking from the blog to the homepage will be an "external" link and a traditional page load (non-SPA) will be made.

With this link setup, and the decoupling of content and code, you can combine an SPA with server side pages, or multiple SPAs, or server side pages and a React App and an Angular App, or a combination of any of the above. Your architecture is more scalable, content and logic can be shared, and marketing teams don't need to worry if their content is in an SPA or not, they can create and link as usual.

## Marketers Keep the Features They Need

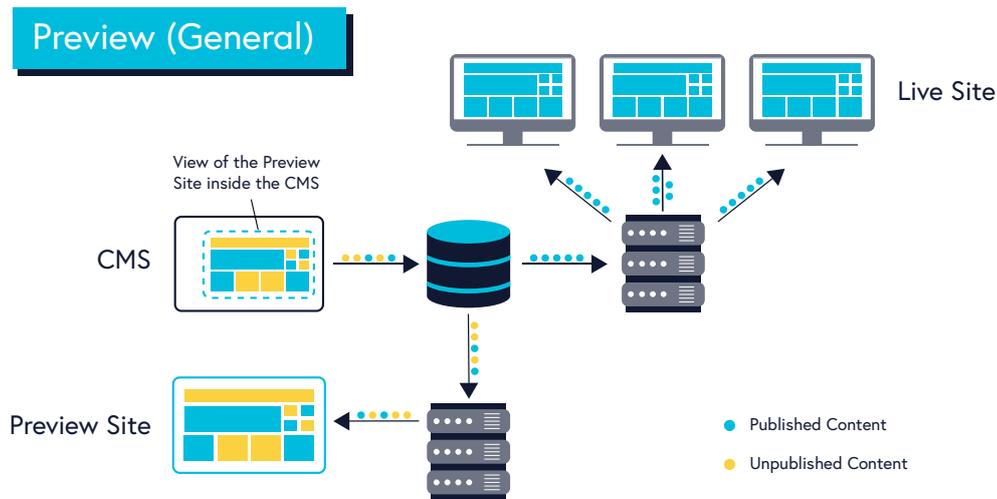
Regardless of how flexible a system is to build, if marketers can't do their job without going through developers for every update there is going to be a bottleneck. This is why one of the main factors of BloomReach's SPA support is ensuring marketers can update and edit SPA driven experiences in the way they are used to managing traditional pages.

## Preview

To be able to edit in the what-you-see-is-what-you-get (wysiwyg) way that Marketers are used to, you first need to be able to see what can be edited. However, the crux of an

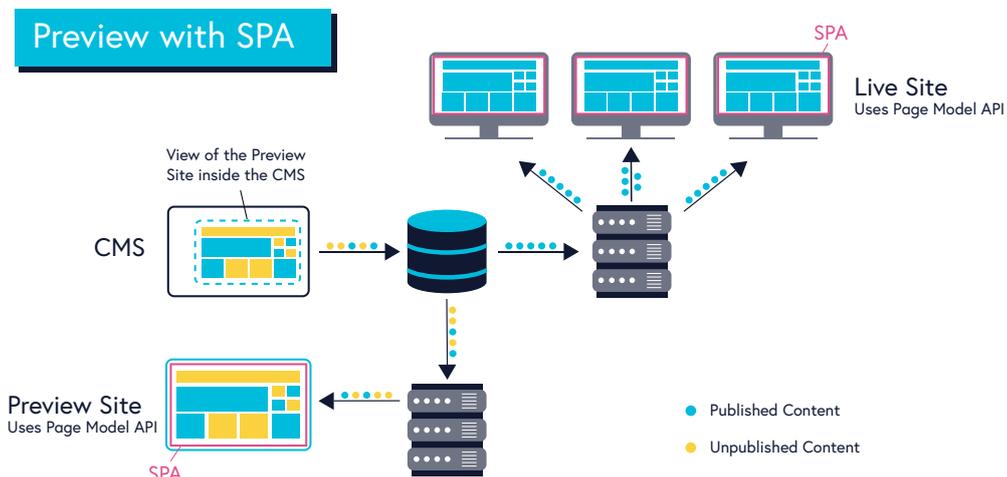
SPA is that the rendering is done in the application on the front-end, so this means that your back-end CMS has to have, well, a front-end. Luckily, BloomReach has always handled it's preview functionality in this sort of way.

So let's talk pre-SPA real quick. To provide a preview in the CMS, you use a special preview site that sits in a server. It acts like any other site - a browser asks it for information, it runs to the database to check if it has the most updated version of that information, and it sends back the certified-fresh information - the key difference is that it is set up so that it can find all the unpublished content too. In a way, the preview in the CMS is acting as a browser to display content.



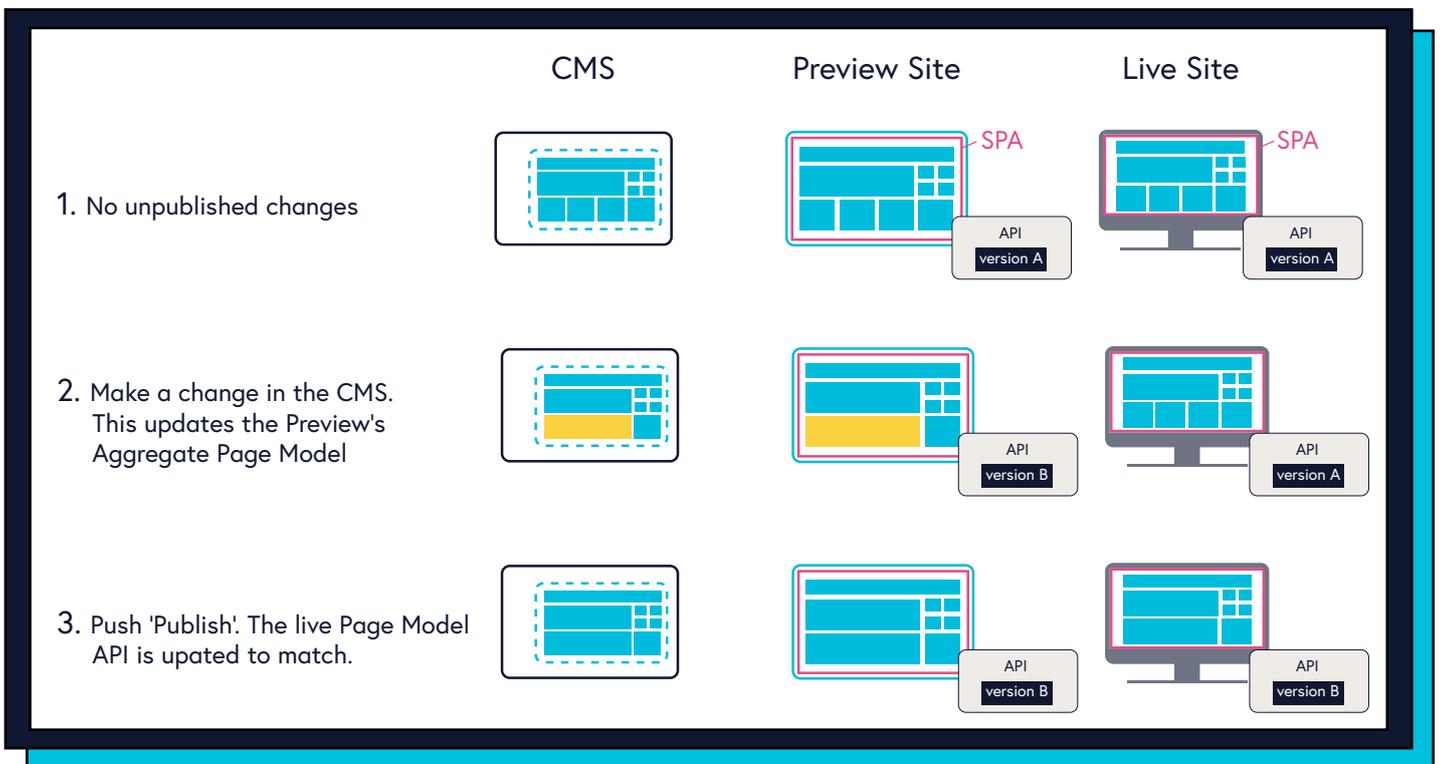
With this set-up already long in place, being able to preview SPAs isn't too big of a jump. The SPA and the CMS have an API contract (defined during integration) on what components can be edited in the CMS - the meta-data from the API tells the SPA which of these components are present, the properties of each component, and a map containing all of the content referenced by the components present.

That special preview site mentioned above now has an SPA on it which powers what you see in your back-end CMS.



The SPA generating your Preview has its own Aggregate Page Model, aka the map of components and content. When you make a change by editing the preview, the CMS compiles a new map based on those changes. This only changes the map for your preview, so the map the public is getting, and the content they are seeing, is not effected.

The SPA takes the new map and renders the updated preview. You push publish. Now you've replaces the old map in the database with the new one, and when a public visitor asks for the map your servers will hand them the new one.



## New Pages

With Single Page Applications you create a new view, or a new route, instead of an actual page, but with BloomReach's SPA support a Marketer would be able to create these new views in the same way they are used to creating pages. Create a new page in the Editing Mode (like you normally would), and

you are given a fresh template in the Preview that you can add content and containers too. Behind the scenes this new template and the content you add is being serialized and added to the Aggregate Page Model, but as an editor you can create (and delete) like usual.

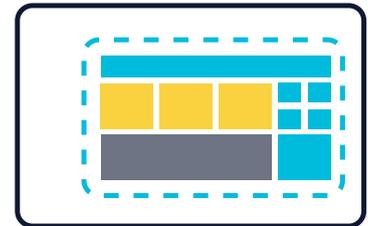
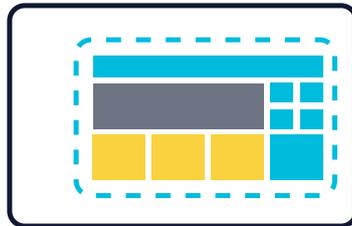
## Drag and Drop

Thanks to meta-data tags the API uses, the list of components and content the CMS gives the SPA is split into pieces with clear "begin" and "end" points in the map. If you drag a component to a new place in the preview,

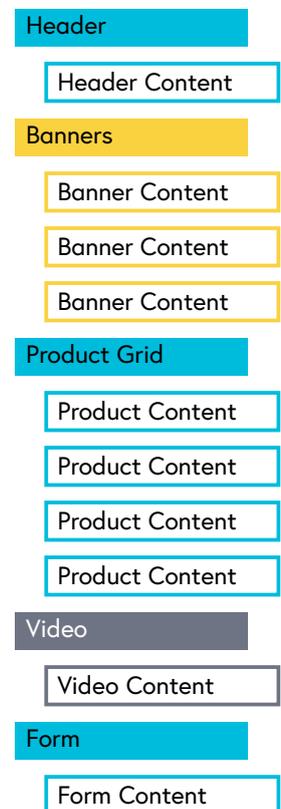
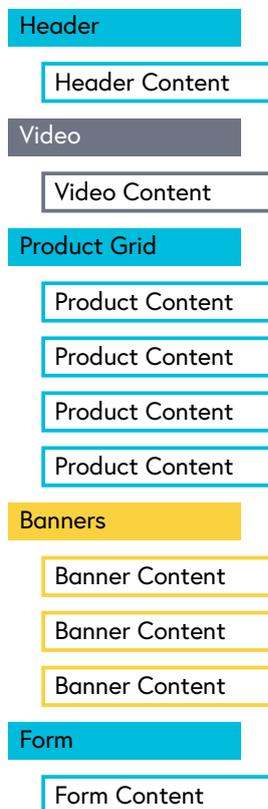
the CMS notices the order has changed and updates the order of that list. The SPA then renders the list in a new order. You can move components around in the standard wysiwyg way, and the API flexes to make it work for the SPA - no development work needed.

## Drag and Drop with SPAs

Editing in CMS



Preview API map



## Personalization

This is where BloomReach's server-side, component-based personalization really comes in clutch. Content personalization is done by creating variants of content and components, gathering the right data natively and via APIs, and segmenting visitors to determine who sees what.

When a visitor comes onto the page, the BloomReach Relevance module personalizes the page in real time:

1. The browser (or in this case the SPA) does the request for the pieces it needs.
2. The BloomReach server receives the request and the BloomReach Experience Relevance module kicks in.
3. The Relevance module will fetch all the information and load the right content based on the relevance profile that is created.

This happens every time anything new is requested, ensuring that the BloomReach Experience Relevance Module will constantly update the information that is known about the visitor.

Determining the right content to deliver is done on the back-end, so the map the Page Model API delivers to the SPA already has personalization in there. This means that there is no Javascript, front-end

personalization tool fighting with your SPA when rendering the page. And you can test all the different variations as well, by leveraging the Experiments feature.

So what information can you use to personalize within an SPA? Literally anything you can collect - browser behavior, time of day, day of the week, information from your other tools like marketing or e-commerce platforms. A common ways we see our customers start using relevance (with or without an SPA) is with inbound marketing campaigns. If a customer clicks an ad about "How to start saving for retirement in your 20s" their view of the homepage will show information on 401Ks, buying your first home, and more bold investing options. If a customer comes in through an add on "How to start a college fund for your baby" they will see a homepage that has information on life insurance, budgeting tips, and more low-risk investing options.



## A Win Win Win

Ultimately, with the core decoupling of content and presentation, the hybrid setup, editing features, and server-side personalization, developers can use SPAs at an Enterprise level, Marketers are happy to use SPAs because editing works exactly how they are used to, and your visitors get a incredibly engaging experience that feels cohesive at all times. A win win win.



**bloomreach**

BloomReach software enables highly personalized digital experiences for retailers, banks, distributors, manufacturers, public service agencies and other enterprises. BloomReach has combined a super CMS, digital merchandising, site search and SEO into the world's first open and intelligent Digital Experience Platform. brXperience stimulates customer engagement and loyalty, even as it boosts conversion and increases revenue.

With a global network of certified partners, BloomReach serves hundreds of enterprise customers such as Neiman Marcus, Staples, REI, Mailchimp, FC Bayern München, and the UK's NHS Digital. Founded in 2009 and backed by NEA, Battery Ventures, Lightspeed Ventures, Bain Capital Ventures and Salesforce Ventures, BloomReach is headquartered in Silicon Valley with offices worldwide.

Learn more: [www.bloomreach.com](http://www.bloomreach.com)