Headless Commerce:

How a Headless + Jamstack Architecture Yields High Performance and Maximum Conversions

Abstract

As the coronavirus pandemic has crippled economies around the globe, e-commerce has boomed. Consumers have embraced online shopping and digital storefronts rather than brick-and-mortar ones. These have become the primary, if not only, source of sales conversions for most retailers. In fact, COVID-19 has accelerated e-commerce growth by 10 years[1], and has thrust web site performance into the business-critical spotlight.

Today's digital consumers demand an exceptional digital experience. This includes extremely rapid loading of pages, services, and images, speedy and reliable checkout processes, extensive personalization of the shopping experience, and inventory updates that keep pace with a customer's browsing. As a case in point, Amazon estimates that a slowdown of just one second in their site would cost the company $1.6 billion in sales each year[2]. The impetus to continuously optimize web site performance is enormous, and the challenge of doing so is becoming increasingly more complex. This is especially the case as consumers rely more heavily on mobile platforms. Needless to say, massive amounts of enterprise resources are being devoted to this effort.

The premise of this white paper is that simply by choosing the right architectural approach—the right commerce stack for its website—an e-commerce company can increase conversions by 20% or more, which for many businesses can mean the difference between success and failure. By adopting a new, headless architecture with Jamstack, enterprises can not only improve e-commerce performance but revolutionize it.

Problem

As brick-and-mortar stores have closed due to the coronavirus pandemic, retailers have seen hundreds of thousands of points-of-sale reduced to a single one—the digital storefront. The digital storefront can be a difficult place to convert potential customers to buying customers. For starters, how does a digital storefront create "experiential shopping" the way a brick-and-mortar location can? Where's the ambiance? The lighting, the smells, the smiling staff, not to mention the opportunity to see, hear, touch, and try on or test the product? Secondly, with digital storefronts, pricing is transparent. With a matter of clicks, anyone can comparison shop for better prices from competitors around the world. This drives down margins and puts the

[1] https://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales

[2] Source tbd

emphasis on volume. That, in turn, makes lost conversion opportunities more costly than ever before. Another challenge of digital commerce is capturing and keeping the attention of the shopper. A shopper in a brick-and-mortar store may be content to browse or stand in line for 15 minutes; a digital shopper is much more impatient and easily distracted. For online shoppers, milliseconds matter.

These challenges explain why e-commerce managers are acutely aware of the need to improve website performance. Downtime, slow load times, bounce-rates, etc., all directly and negatively affect the bottom line.

- **Google, for instance, found that improving site load time by a tenth of a second boosts conversion rates by 8%[3].**
- **According to a survey by bytes.co, 56 percent of customers leave after 3 seconds if a site or image fails to load, and 20% never return[4].**
- **Amazon estimated in 2012 that a slowdown of just one second on their site would cost the company $1.6 billion in sales each year[5]. Imagine what the loss would be today, after eight years of revenue growth.**

Ironically, consumer adoption of innovative technology makes the e-commerce challenge even more complex. **MachMetrics**, for instance, describes how mobile phone use impacts e-commerce experiences:

> In 2017, mobile internet usage surpassed desktop as the majority. [Three] years later, this is still the case with 53% of website visits coming from mobile devices. Projections for the next few years expect the mobile market share to increase as mobile devices become more powerful and more prevalent. Unfortunately, the data from **Backlinko**[6] paints a bleak picture when it comes to mobile internet browsing. They found that the average web page takes 87% longer to load on mobile vs. desktop. In addition, the average desktop Time to First Byte (TTFB) speed is 1.28 seconds on desktop and 2.5 seconds on mobile. Finally, they found that the average time it takes to fully load a webpage is 10 seconds on desktop and 27 seconds on mobile[7].

It's no wonder that web developers and IT operators are pulling out the stops, using many different resources to "optimize everything" in the quest for better website performance. Typical approaches include horizontal scaling, Varnish caching, Redis caching, load balancing, and geo-redundant clusters. "Unfortunately though," to quote Retail Dive, "providing great performance is becoming more complex than these obvious approaches."[8]

---

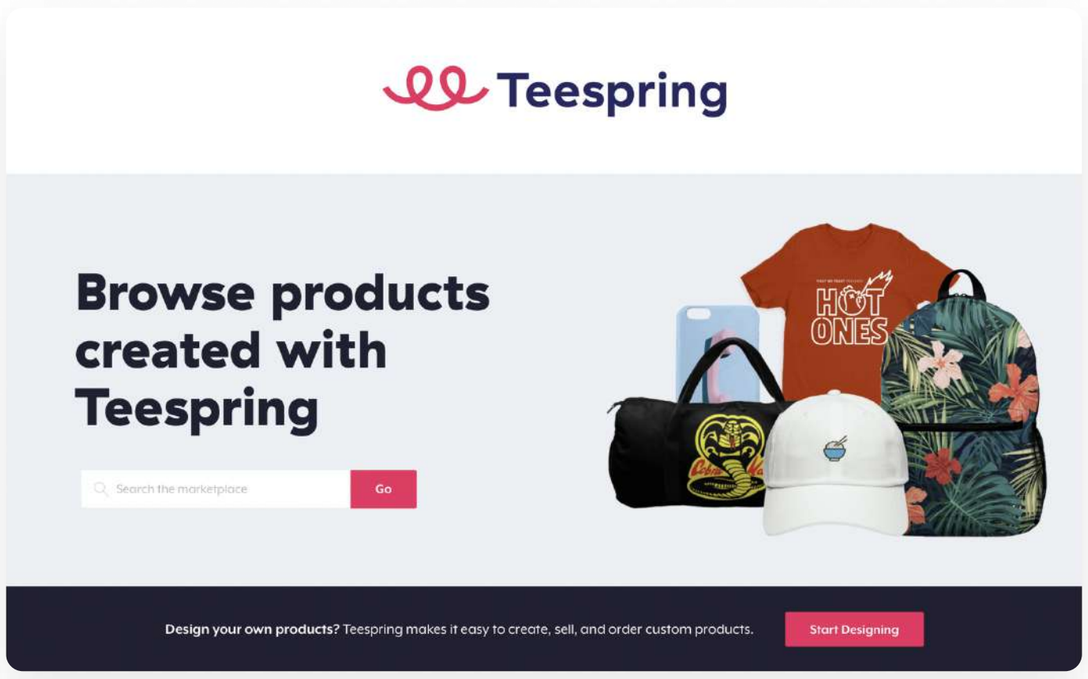
[3] https://www2.deloitte.com/content/dam/Deloitte/ie/Documents/Consulting/Milliseconds_Make_Millions_report.pdf
[4] https://bytes.co/blog/hosting-support/people-throwing-phones-slow-load-times-impacting-business/
[5] https://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales
[6] https://backlinko.com/page-speed-stats
[7] https://www.machmetrics.com/speed-blog/average-page-load-times-for-2020/
[8] https://www.retaildive.com/news/amazon-is-crushing-retailer-load-times/507370/

A Preview of the Possibilities

The good news is that converting to a headless architecture approach with Jamstack offers radical improvement, not only in performance but also in developer flexibility, speed of experimentation, and time to market with a secure, compliant product.

The story of Teespring provides an excellent example of the transformation that is possible.

Teespring

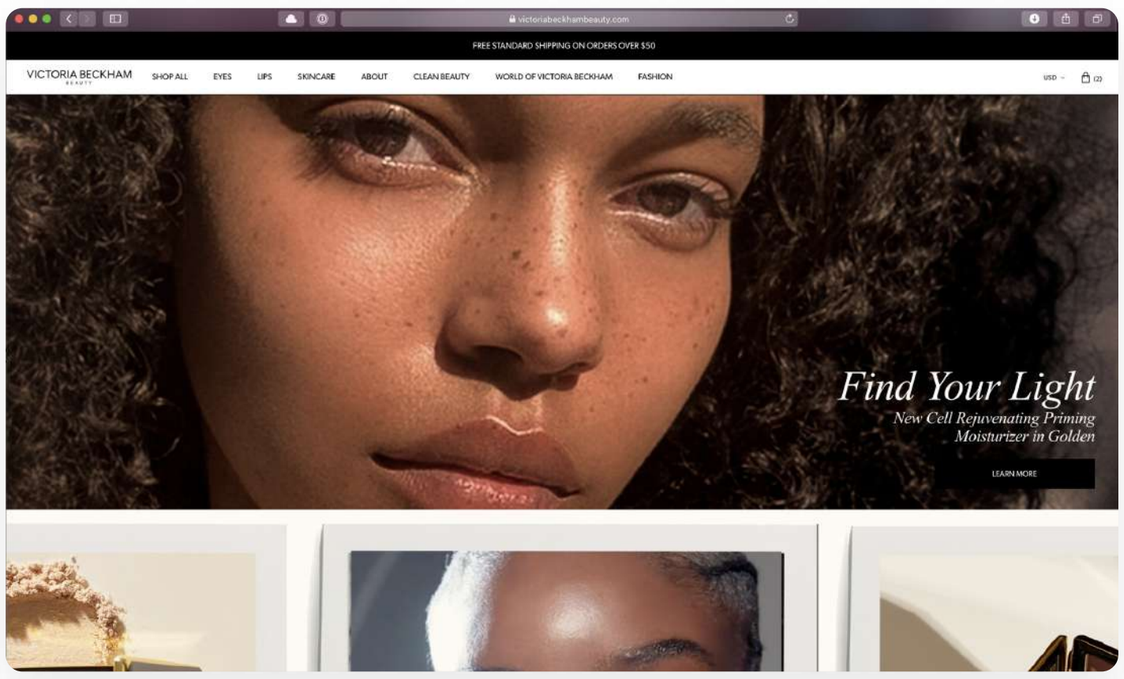


Teespring is a large e-commerce marketplace where individual designers and content creators can launch their own turnkey storefronts for custom-branded apparel and merchandise. The Teespring community is huge, serving more than a million merchants and hosting tens of thousands of storefronts. As the company was experiencing massive growth, Director of Engineering Rick Takes realized their monolithic Ruby on Rails web architecture could not scale to support their expanding web team and initiatives. He set out to prototype a Jamstack approach using Netlify, breaking its monolithic e-commerce app into microservices.

As a result of the prototype and gaining buy-in, he has improved the developer experience and productivity for his team and enabled them to launch an ambitious new initiative called Branded Stores. The early results from the new Branded Stores on the Jamstack have been promising, delivering double the performance compared with the former architecture, especially on mobile, and increasing conversion rates.

Victoria Beckham Beauty is another company demonstrating how moving to a modern website architecture directly impacts the bottom line.

Victoria Beckham Beauty



Victoria Beckham Beauty (VBB) launched a new site for its cosmetics line using a headless architecture approach, and the site has set a new performance bar for the fashion industry. Ryan Foster, head of VBB's web agency Fostr, said "Other fashion brands benchmarking against VBB want to understand how it's so much faster." [9] Indeed, the site has impressive performance, but raw speed is far from the only benefit VBB has achieved by "going headless."

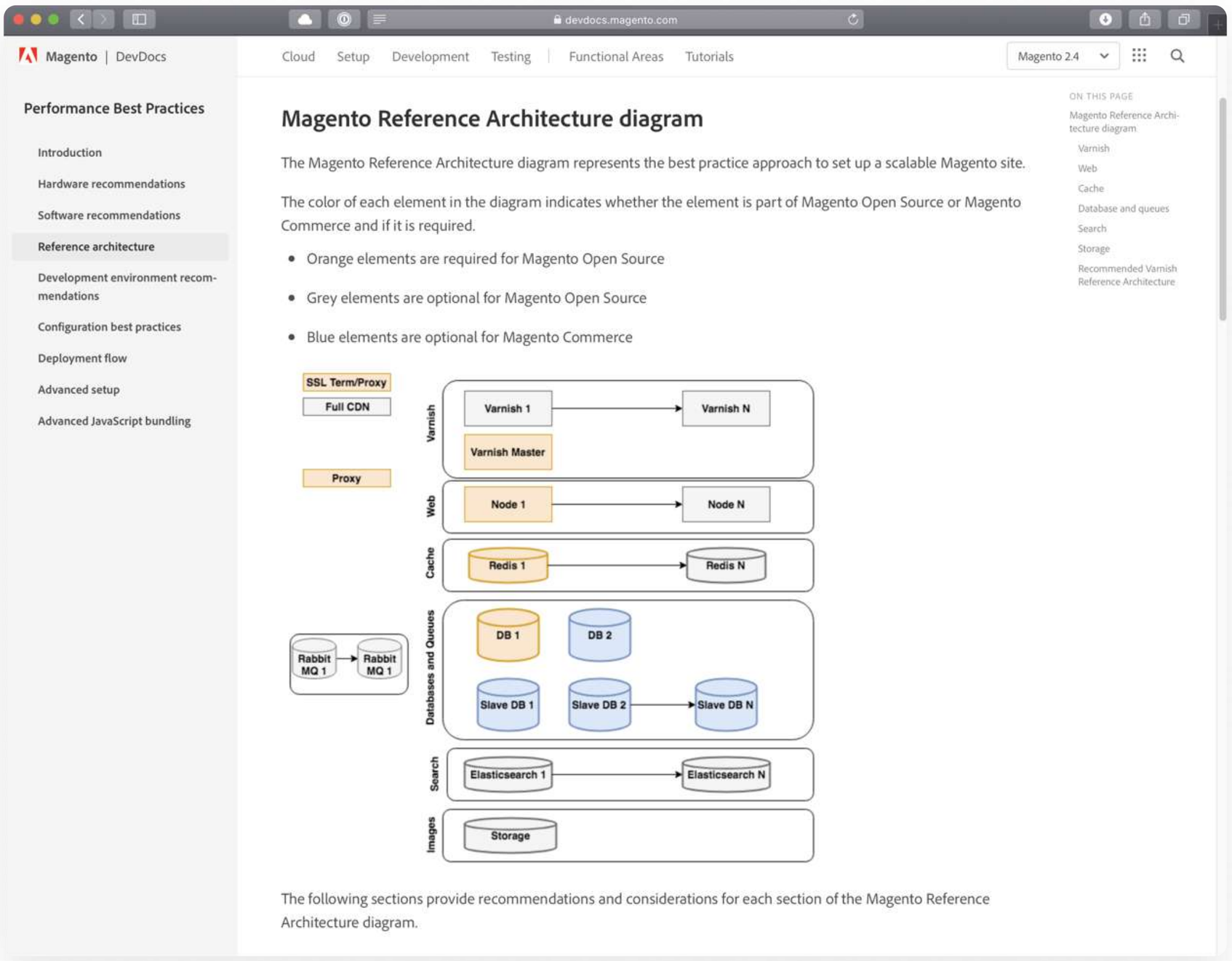
In fact, VBB's primary objective in adopting a headless architecture was not speed but flexibility. Rather, Foster and the VBB team wanted fine-grained control over the brand experience—from visual design to user journeys, even down to the exact text of each URL. Midway through a site redesign, Fostr was struggling to find this flexibility using Shopify's traditional templates and monolithic approach, so Ryan Foster made the rather bold decision to convert to a headless architecture, despite the understandable concern that switching architectures in the middle of a launch might slow development and jeopardize schedules. What the team discovered, however, is that development velocity actually increased because the headless approach eliminated the design and development constraints of the former, traditional architecture.

[9] https://www.netlify.com/customers/victoria-beckham-beauty/

Background

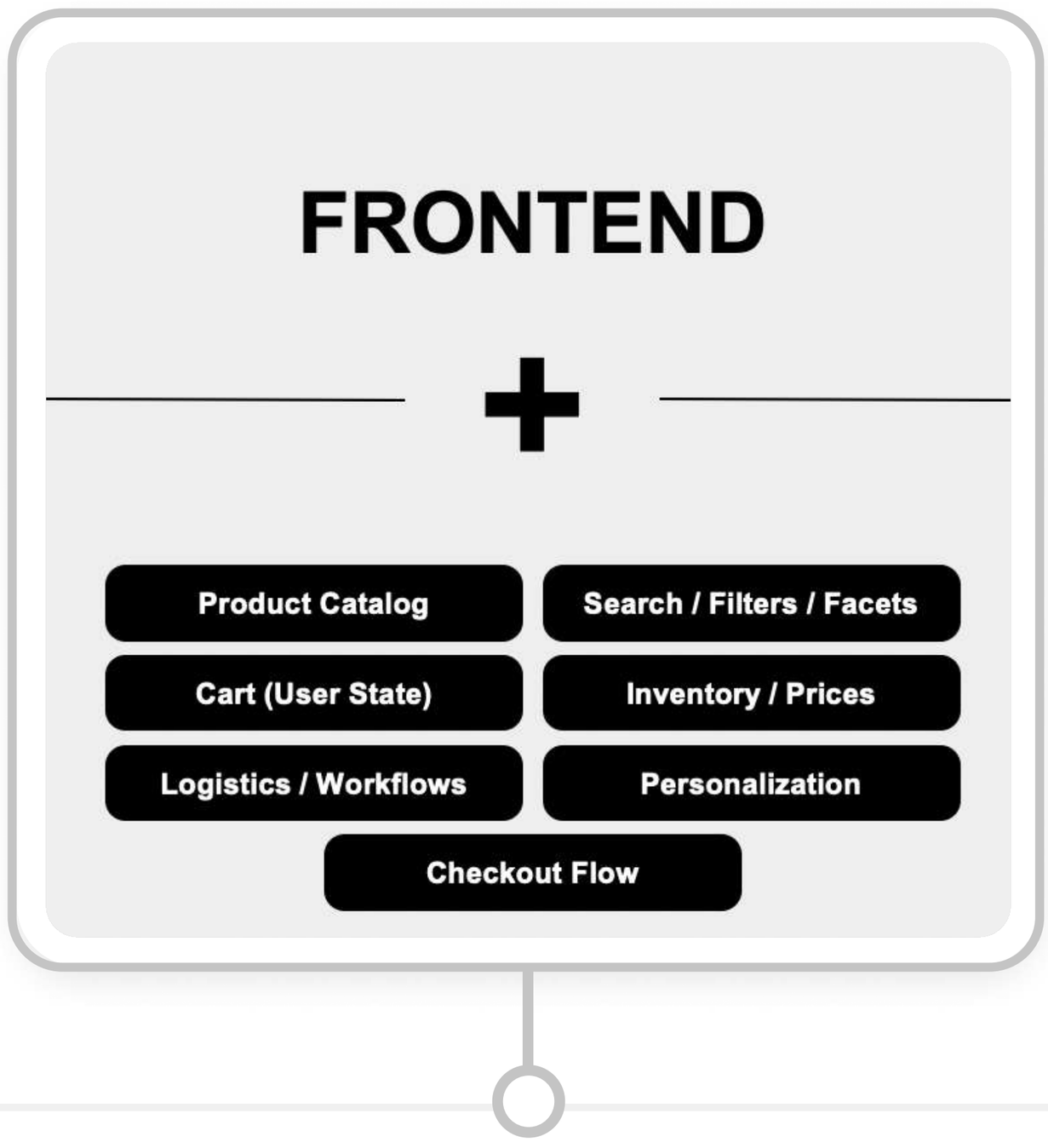
The traditional approach to website design is based on a monolithic architecture. One set of tightly coupled infrastructure components is responsible for the application. This approach worked for developers in the past but now is breaking down under the complexity and demands of modern users, development needs, and technologies.

To illustrate, pictured below is Adobe's recommendation for a fairly traditional stack to run Magento, one of the traditional, open source e-commerce platforms.



You can see here the many different components that are required to optimize the performance of this system. These include a Varnish cluster for caching generated HTML, several web nodes, a caching layer built with Redis to lighten the load on the database, a large database cluster with replicates, a dedicated search pod on top of elasticsearch, and dedicated storage layer for images as well as messaging queues, CDNs, proxying, and so on. All of these complicated components are required to operate at a fundamental level of performance for a modern e-commerce experience.

Inside the backend of this monolithic system are the product catalog; search, filters, and facets; inventory and prices; the user cart with user states; the personalization engine; logistics and workflows; and checkout flow.



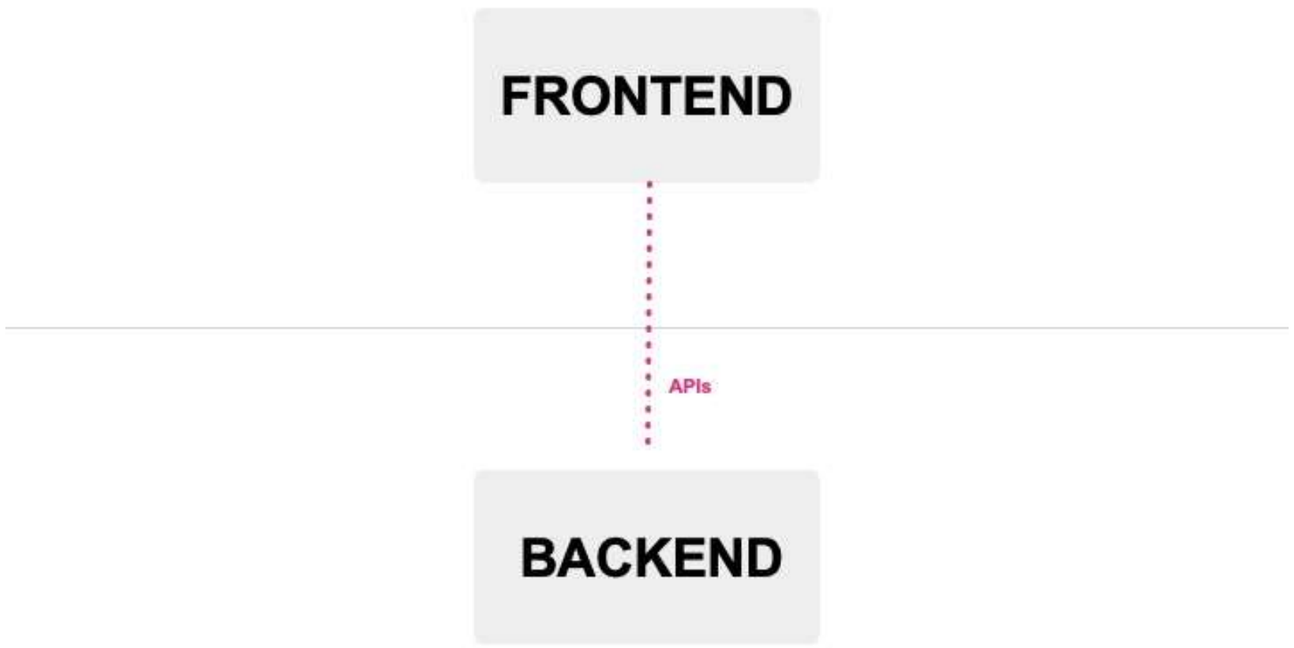
With this architecture, you can see two major limitations:

1. **Everything is in one stack:** the backend services are tied to and serve the entire frontend as a unit. Thus the stack is strongly opinionated, which makes it hard for developers to customize the site and express the company's unique brand. Rather than developers choosing the features they want, the platform tells developers what they can have.

2. **Every web request has to traverse this complex stack:** all requests from all visitors hit this common origin. This creates a significant performance hurdle.

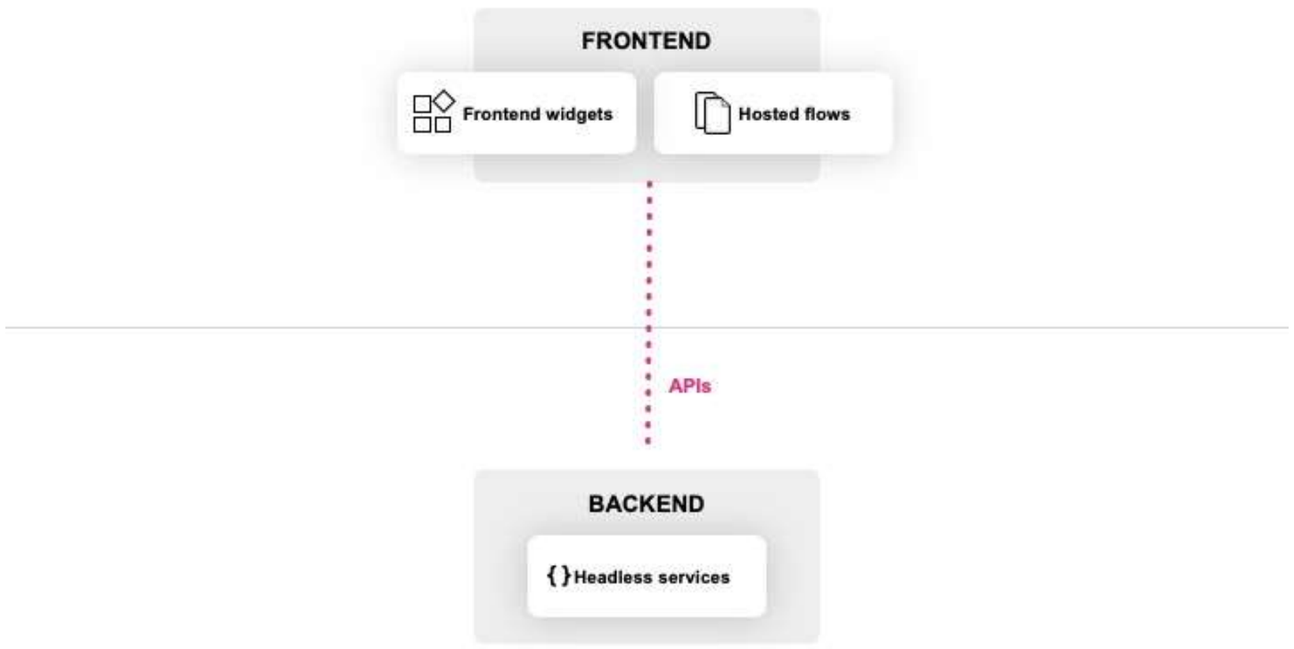
Solution

The headless approach, by contrast, disaggregates the monolithic stack, decoupling the frontend from the backend, tying them together with APIs. By so doing, headless gives us the unique opportunity to combine best-of-breed services for all the components of e-commerce.



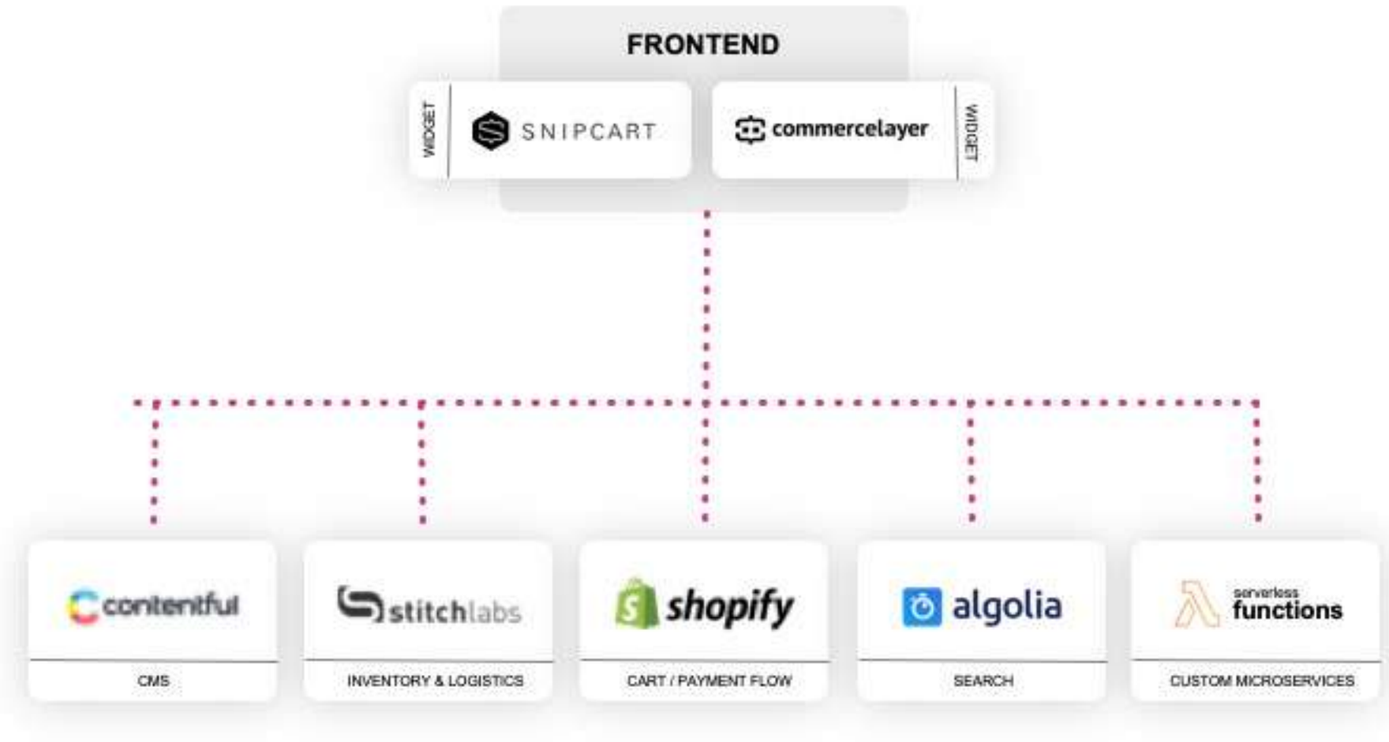
Frontend

A self-standing frontend enables developers to take many different architectural approaches. Some may give their users a luxury experience by completely customizing how the site looks and performs. Or, in the case of smaller projects, a developer can quickly add features by pulling in widgets to power the shopping cart or the checkout flow. Others may take the approach of hosted flows, where you build a fully customizable product catalog and search, but the actual checkout takes place somewhere else.



Backend

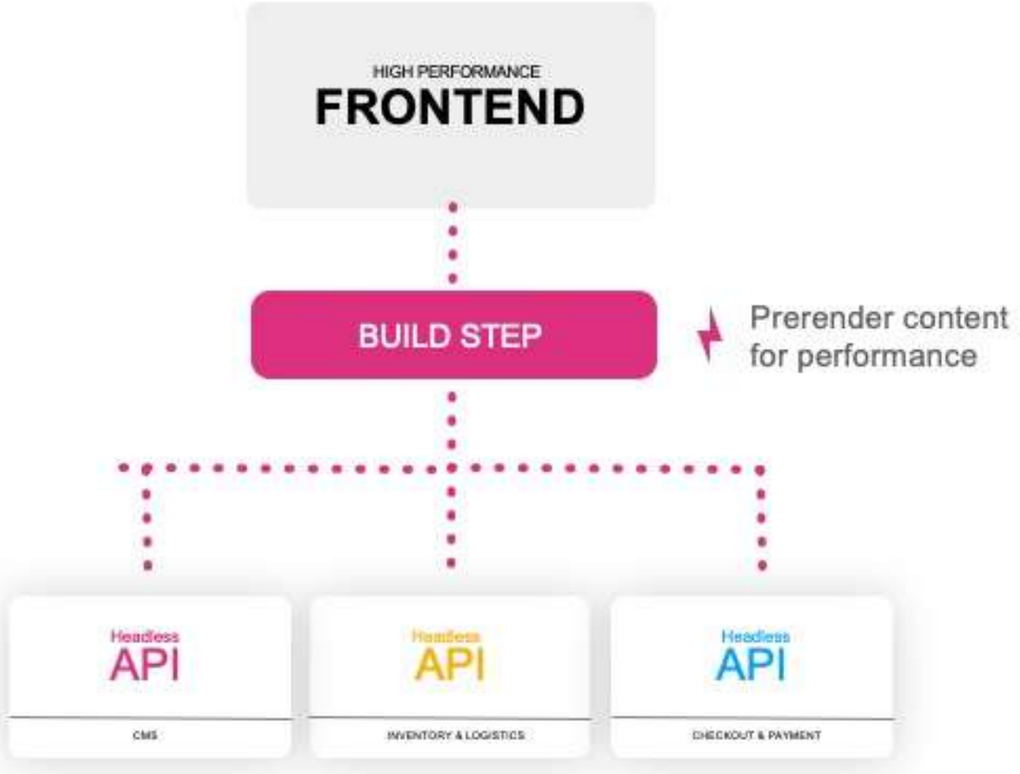
Going further, the real power of headless is that there's rarely only one backend. For example, you might use a content management system like Contentful to manage your product catalog. There are a wealth of choices for CMS systems. You could use Stitchlabs to manage inventory and logistics or run payments through Shopify. Algolia is a common option for faceted search and filtering. Serverless Functions can be used as a custom layer to glue all of these different services together. With this flexibility, web developers building the frontend can tailor the API calls to a variety of different backend headless services based on what they are building, not what the platform will allow.



This flexibility truly transforms website development and performance. That's why about 56% of all enterprises are currently exploring these types of headless architectures, according to Gartner. Enterprises are looking to achieve the performance, conversion rates, and developer experience that headless architecture can offer.

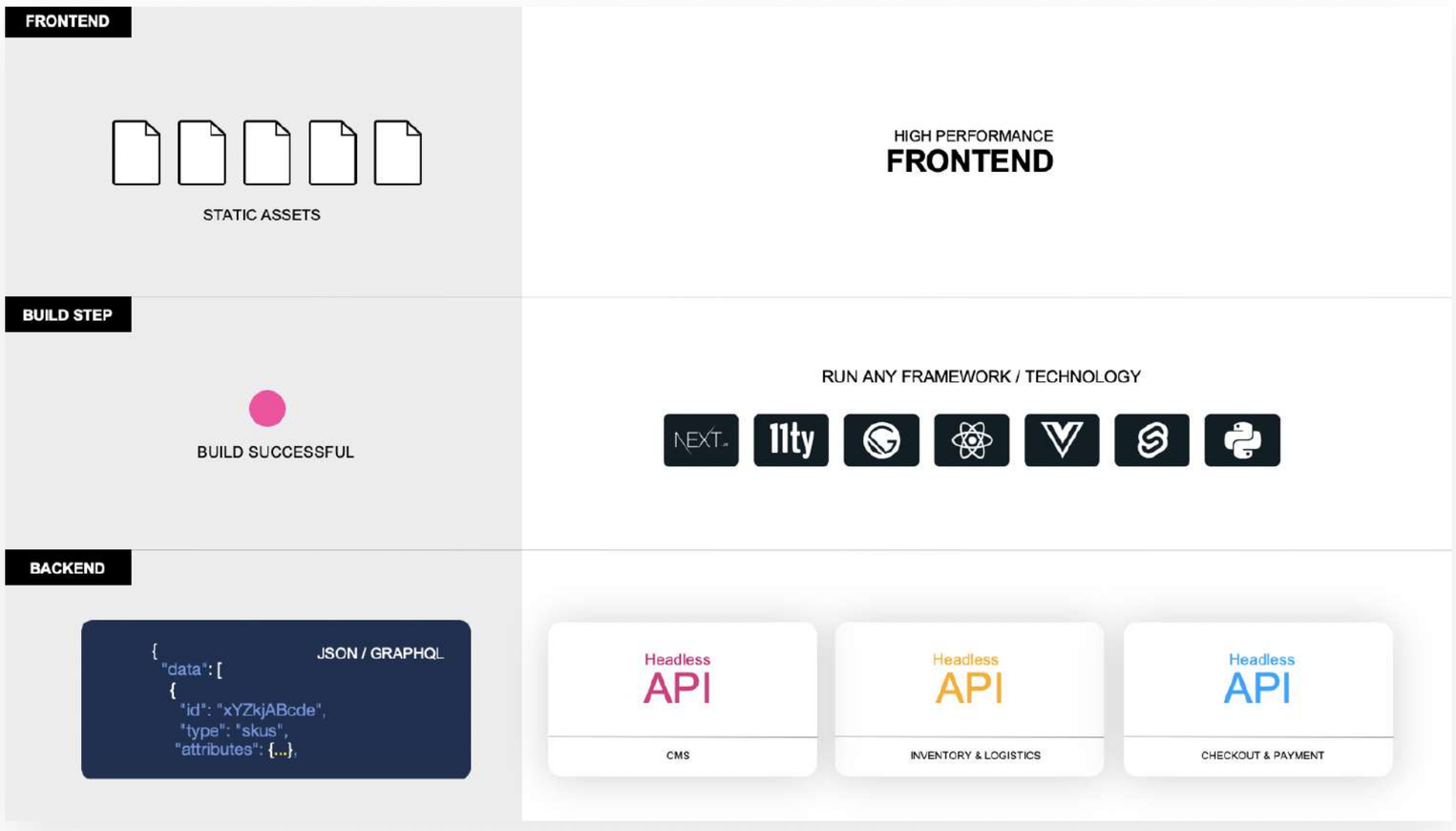
But there's one more ingredient in the secret sauce–the Jamstack.

Headless alone does not guarantee site performance achieved by Teespring and Victoria Beckham Beauty. What these two companies are doing differently with Netlify is pairing a headless commerce backend with a Jamstack approach to the frontend.



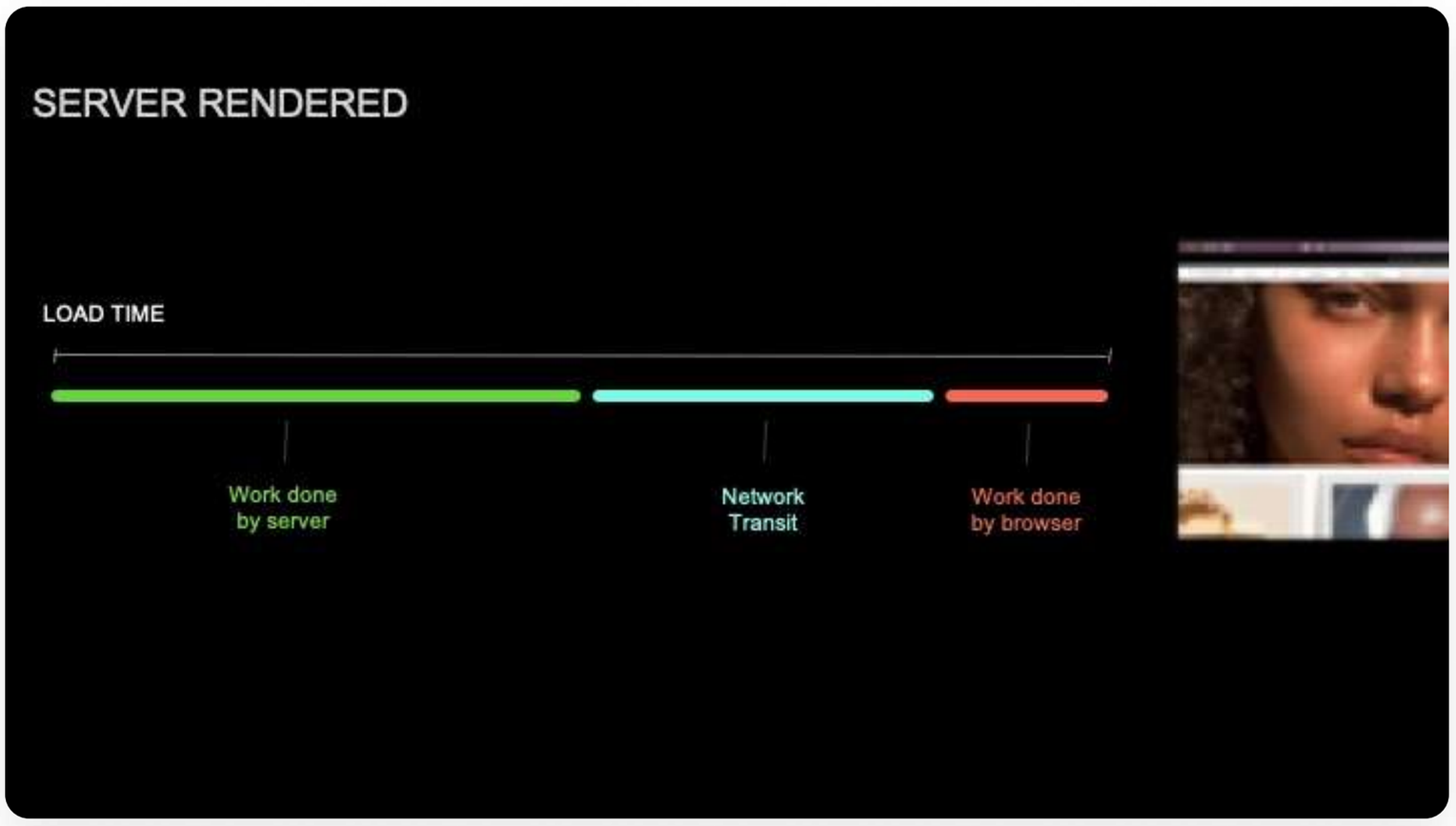
The essence of the Jamstack approach is to minimize the work done by the website server and by the user's browser, accelerating the delivery of customized content to the user. This is done by pre-rendering content when building the application.

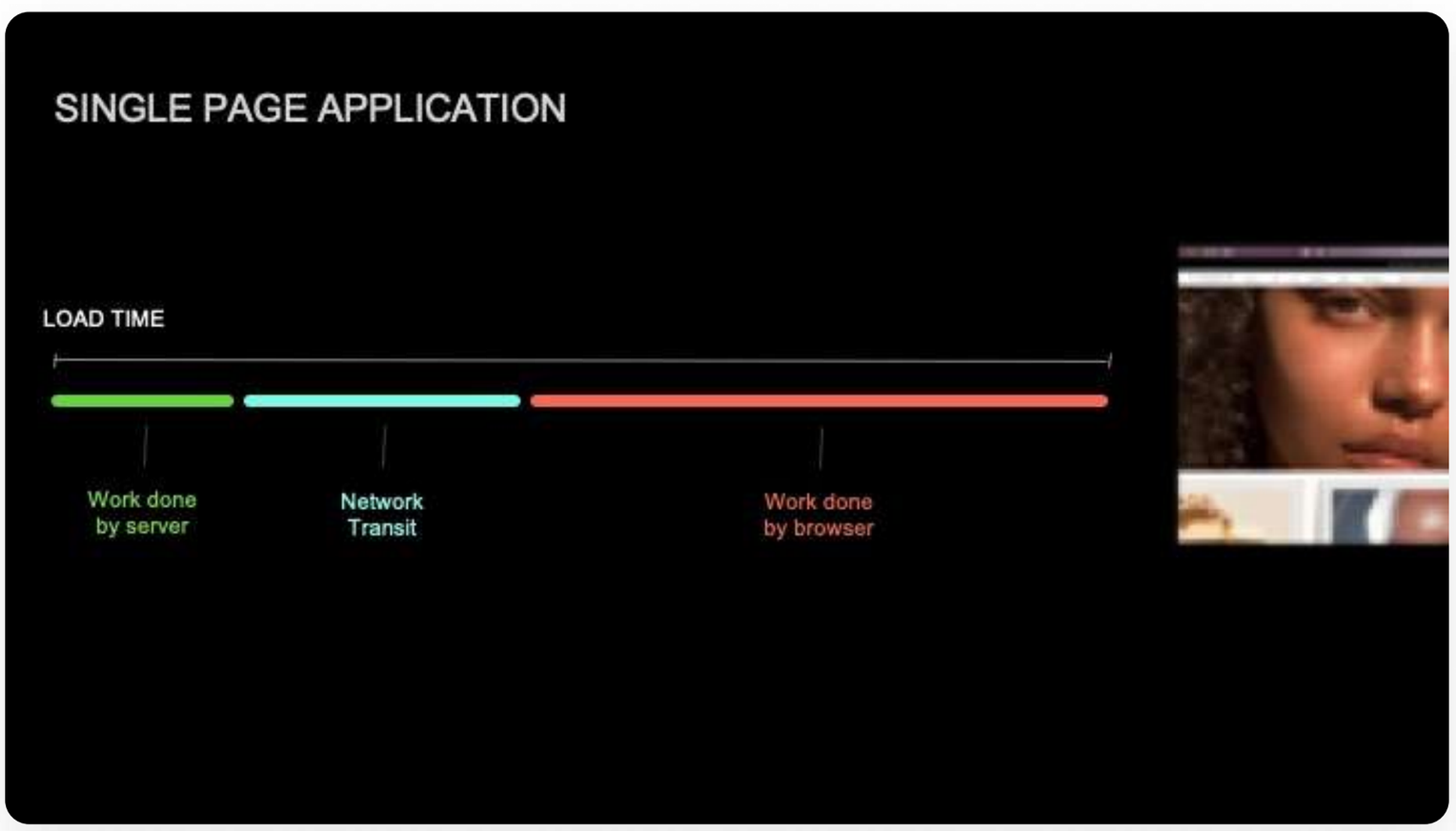
The backend headless services—such as the CMS, inventory, logistics, checkout, and payment services—are exposed through a REST API or a GraphQL API as JSON documents. Developers can pick any site-generating framework or technology—for example, Next.js, Eleventy, Gatsby, Gridsome or Nuxt—that provides the right user experience for their customers. By pre-rendering assets such as product catalogs and detail pages, they can be distributed across a global network and delivered instantly to the user's browser.



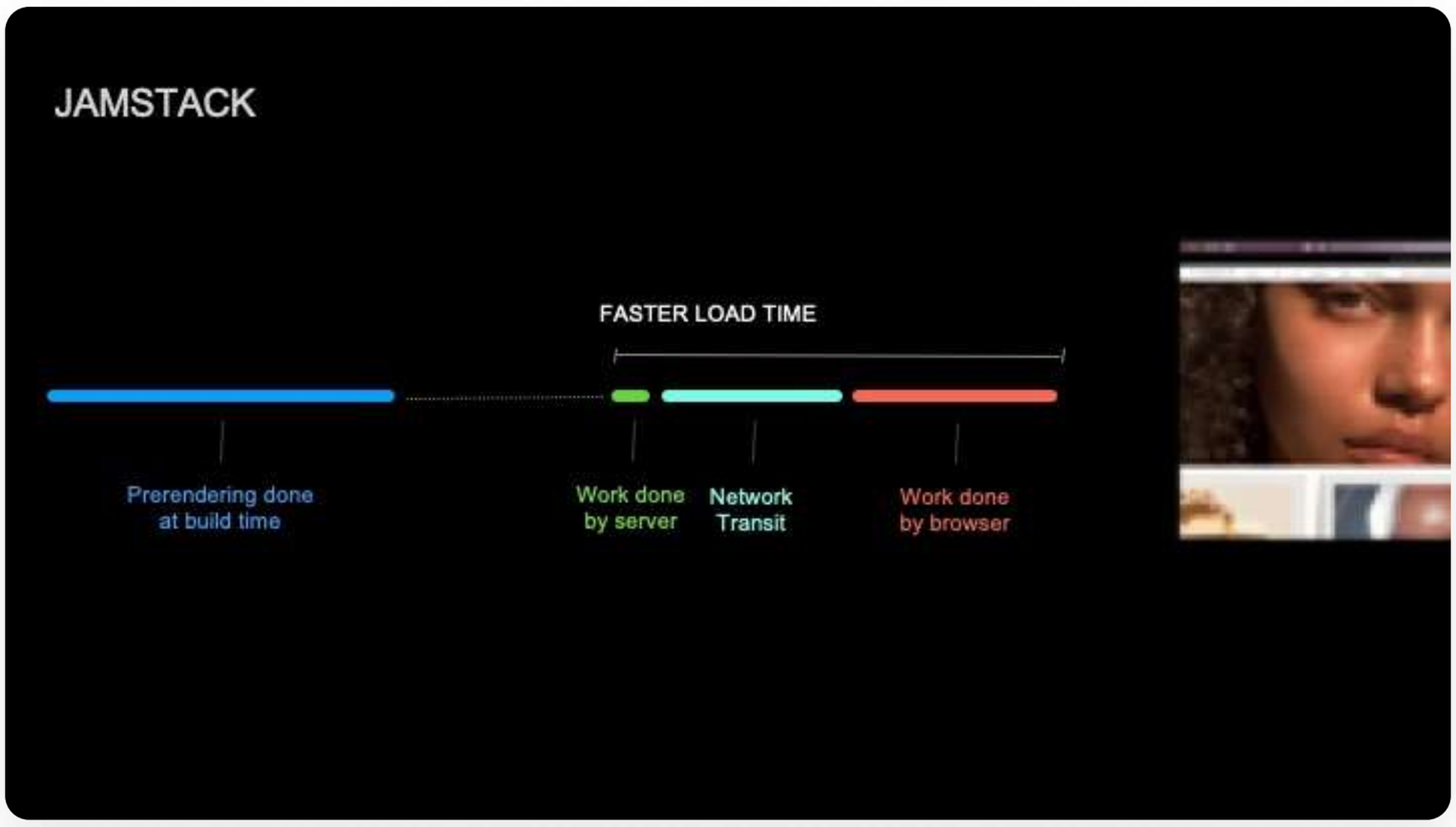
A Stark Contrast to the Norm

In the traditional approach, a website request requires that the server do a lot of work, pass the content via network transit to the user's browser, and the user's browser engine also does a bit of work on the content before it is finally made available to the user. Even in the case of single-page applications where very little work is required of the server, the user's browser essentially receives an empty app shell, and the user's browser then does all the work of stitching together the view for the user.



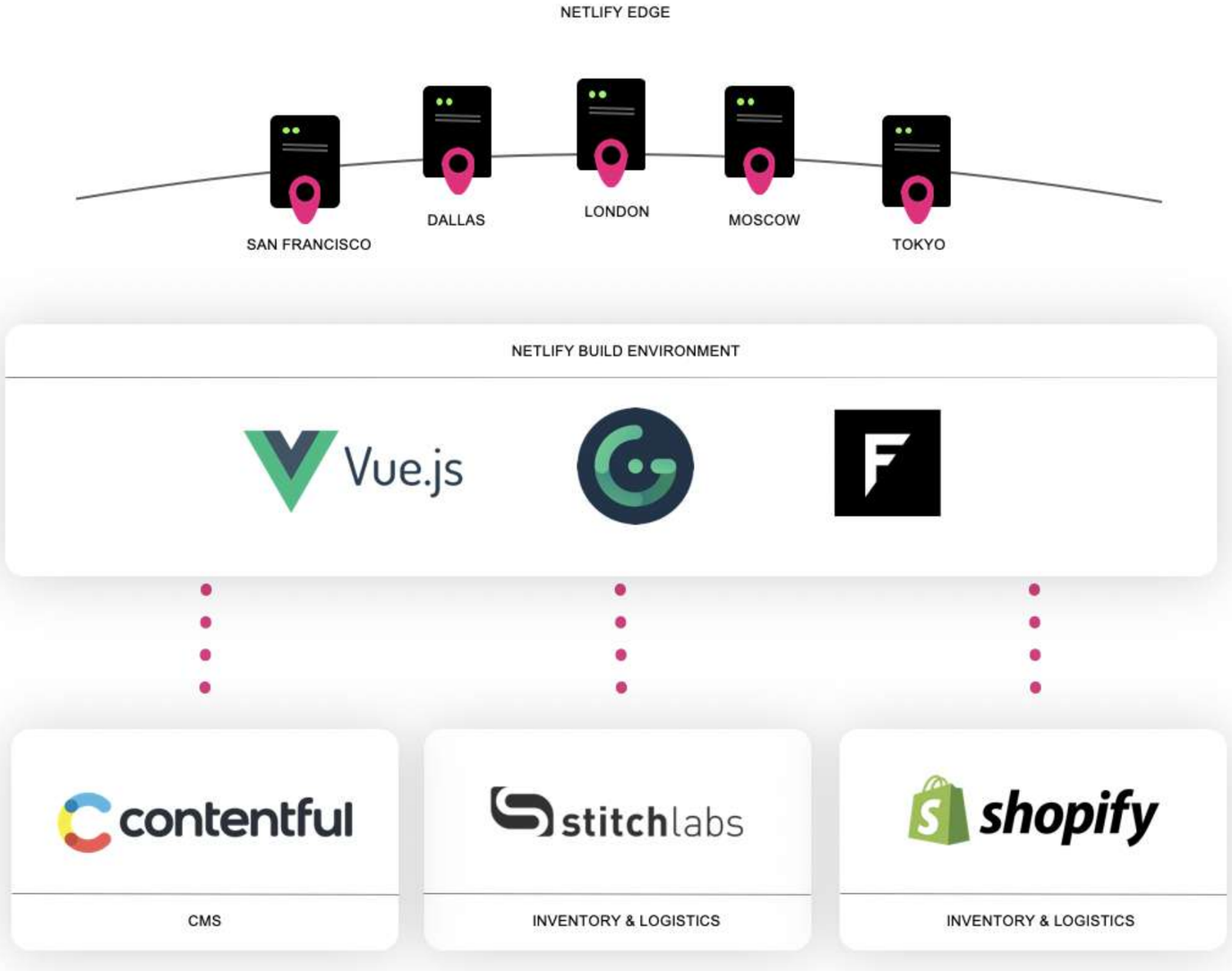


In contrast, pre-rendered content is delivered so rapidly that the user experiences a near-instantaneous response to requests, and a user interface that is custom tailored.



A Closer Look at Victoria Beckham Beauty

VBB adopted a Jamstack using Gridsome and Vue as the layer that allowed their developers to build exactly the experience they wanted for their users. VBB used Contentful for the product catalog, Shopify for the actual checkout flows, and Stitch Labs for inventory and logistics.



As mentioned above, the speed of VBB's site has become an object of industry envy, but more important to VBB is the flexibility it gives developers to choose the integrations that provide precisely the brand experience VBB wants their customers to experience. No longer does the monolithic platform dictate what developers can or cannot do. VBB has also experienced a boost in developer productivity and velocity, because the ease of working with the tooling of hand-picked and refined applications far exceeds the cumbersome experience of working with PHP-based templates inside a big monolithic system. Other advantages of the Jamstack approach taken by Netlify are speed of experimentation, security, and PCI compliance, and the potential to take personalization of content all the way to the edge nodes.

Conclusion

E-commerce managers are acutely aware of how website performance impacts business objectives: downtime, slow load times, high bounce-rates, and other performance challenges all directly and negatively affect the bottom line.

Headless commerce with Jamstack offers:

- **Site performance**
- **Flexibility of microservices**
- **Content integration**
- **Developer productivity**
- **Speed of experimentation**
- **Security & PCI**
- **Personalization**

That is why headless commerce with Jamstack is an architectural approach that’s right for a new era in which e-commerce system performance has emerged as a critical component in the success of retailers everywhere.


netlify