

Improve Customer Experience & Decrease Cybersecurity Risk with Headless CMS & E-commerce

September 2023

Contents

The Next “Big Thing” in Web Development is Here	3
The Current State of Web Development	4
Monolithic Architecture Performs Admirably, but There are Downsides	5
Single Point of Failure	5
Inability to Scale	5
Marketing and Sales Constraints	6
Human Resources Challenges	6
What is “Headless”?	7
The Benefits of Headless Architecture	8
Virtually No Point of Failure	8
Ability to Scale Indefinitely	9
Marketing and Sales Optimizations	10
Human Resources Enablement	10
Embracing Change	11
A Word of Caution	11
Get in Touch	11

The Next “Big Thing” in Web Development is Here

For a business to grow and prosper in the digital age, a robust online presence is a must-have. While many stakeholders questioned the value of investing in a website just two short decades ago, today, a business' website is considered one of its most critical sales and marketing components.

In the early days of the Internet, websites were built from scratch, each requiring hundreds or thousands of hours of development. Then, we saw the emergence of the “customizable out of the box” platforms like WordPress and Shopify, which quickly became the de facto standard.

The purpose of this white paper is to explain the new gold standard of web development, referred to as “Headless” CMS and e-commerce, and how it can be a cost-effective, more secure, and scalable solution for your business.

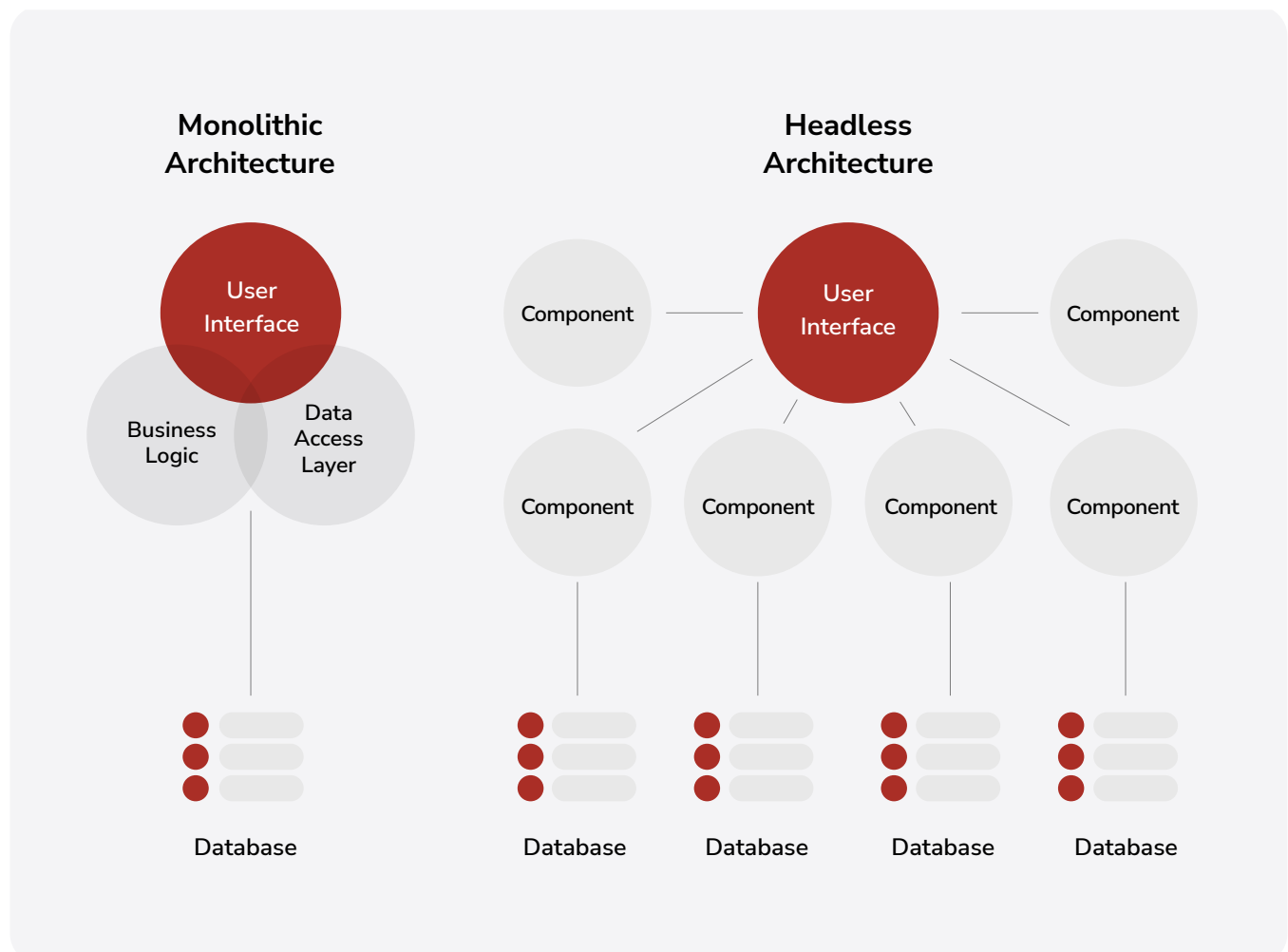


The Current State of Web Development

A standard commercial website is made using three main ingredients, namely HTML, JavaScript, and CSS. For years, these components have done a satisfactory job of empowering businesses of all sizes in reaching global markets via the Internet. However, as website adoption increased, it became painfully clear that websites constructed manually were laborious to create, and virtually impossible to scale.

Fortunately, several technologies emerged that enabled more efficient production and scaling: server-side processing, Content Management Systems (CMS's), and dedicated purpose-built (e.g. e-commerce) platforms. Thanks to these advancements, brands could build, deploy and maintain rich online presences.

But as useful as these tools were, they all suffer from one serious flaw; they are built using what is called 'Monolithic' Architecture. Using the example of an e-commerce store, this means that everything – from the storefront UI, inventory, and payment gateways – were all built in one large, self-contained system. In short, with Monolithic Architecture, all the code and data used to build a website exists in a single environment, meaning that everything is intrinsically tied. All of your digital eggs are in one digital basket.



Monolithic Architecture Performs, but With Downsides

Depending on the scope, applications built using a Monolithic framework are relatively easy to build, test, deploy, and monitor. But this simple architecture comes at a cost (or several).

Single Point of Failure

For all the potential benefit offered by Monolithic Architecture, there is one glaring vulnerability that has been made apparent amidst the staggering increases in cyber attacks¹. The unified structure of a Monolithic build means that once a hacker has gained access to the system, all a business' confidential information, including customer data, can be stolen or held for ransom. While these types of attacks can deal a financial blow to a company to the tune of hundreds of thousands if not millions of dollars, the true cost to the business can be measured in the loss of trust in their brand and the opportunity cost associated with the downtime required to correct the vulnerability.

Security vulnerability isn't the only potential for a single point of failure with Monolithic Architecture. Due to the single codebase, a bug located in any of the component applications can create a cascading effect that can bring down the entire website.

Inability to Scale

As stated previously, the initial development of a website can be easier with Monolithic Architecture, but that ease can quickly turn to a burden once the needs of the business scale. Consider for a moment the following scenario:

Company XYZ's server can handle 5,000 daily transactions. Initially, this volume is more than adequate since they are new to the market and considered a small cap business. After several years of sales, the company plans an exciting marketing campaign around Black Friday, when daily transactions peak at 25,000. Despite the campaign being an initial success, the volume of traffic crashes the server, effectively stopping sales. Company XYZ must quickly add more server resources, a cost that must be paid even during non-peak sale periods.

What makes scalability such an issue?

Since the code, data and presentation layers all reside in the same single environment, they are all competing for resources. And those resources can only expand to the extent that memory and processing capacity can be assigned to the environment.

Additionally, Monolithic code is tightly coupled. As more features are added to support the business, the entire codebase needs to be updated. Want to implement a new user-friendly payment gateway? It's likely the remaining code will need to be revisited, tested, and redeployed. The larger the codebase, the more time and money will go into making even the smallest changes.

¹ Jack Gillum. Ransomware Attacks on Industrial Firms Increased by 87% in 2022. February 14, 2023. Accessed March 20, 2023.
<https://www.bloomberg.com/news/articles/2023-02-14/ransomware-attacks-on-industrial-firms-increased-by-87-in-2022?leadSource=uverify%20wall>



Marketing and Sales Constraints

At a fundamental level, a website is only as good as it is easy for users to access. To clarify, this means that if a website is slow, doesn't rank well (due to poor SEO), it doesn't matter how beautiful or functional it is. In a Monolithic Application, if just one part of the system is under heavy load it will slow down the entire site. This directly correlates to a reduction in customer conversions or even ranking decreases.

The reality with any form of any type of application is that bugs happen. If there's a bug that is negatively impacting your CX, Monolithic Architecture can make debugging very difficult. Since all code is deployed as a single process, isolating the buggy code can be a challenge, and a particularly costly one if that bug has made its way to production!

Cost of a Software Bug

\$100

If found in **Gathering Requirements** phase

\$1,500

If found in **QA testing** phase

\$10,000

If found in **Production**

Human Resources Challenges

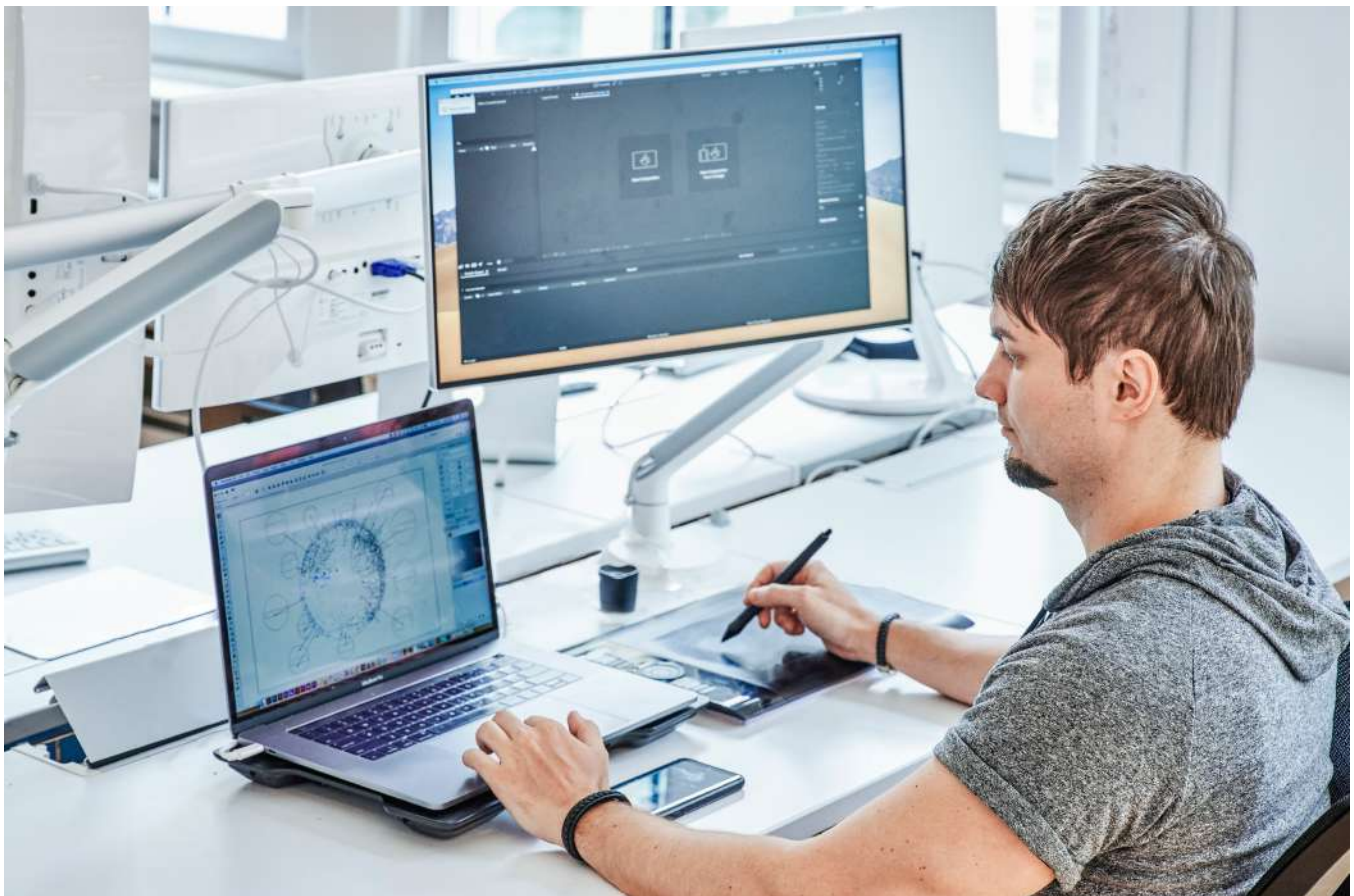
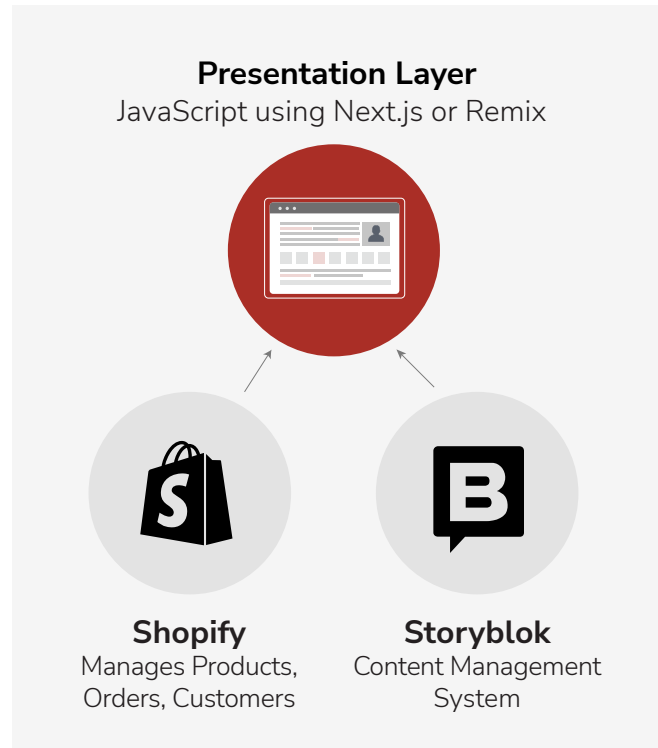
Assembling the right team to build and maintain a Monolithic application can be a logistical challenge. Monolithic applications require multiple programming disciplines respective to front-end and back-end development (e.g. PHP or Ruby for back-end and HTML, CSS, and JavaScript for front-end). It is increasingly unlikely that an HR team currently hiring will be able to find a single developer that possesses enough domain knowledge to adequately fulfill the roles of both a front-end and back-end developer.

What is “Headless”?

Headless Architecture is a modern approach to application development.

Unlike Monolithic Architecture, each component of the application is developed separately and independently of the others, existing in a decoupled state. The benefits of utilizing this architecture are significant to businesses that wish to remain digitally competitive.

Also referred to as microservice architecture, Headless Architecture consists of a presentation layer that contains a user interface that is separated from the data and processing back-end of a website. As we'll see in the sections that follow, implementing Headless Architecture for your website will reduce operating costs, increase uptime, and improve the effectiveness of your marketing strategy.



What is “Headless”?

The Benefits of Headless Architecture

The benefits of Headless Architecture are staggering. Since components function independently of one another, each can individually be developed, tested, deployed, and built-up without impacting the other services. Updates can be performed granularly rather than aggregately, with greater flexibility, speed, and reliability.

Virtually No Point of Failure

As described previously, one of the most flawed aspects of Monolithic Architecture is the vulnerability of a single point of failure. In stark contrast, with Headless Architecture, all of the data and functionality is integrated through microservices or APIs. Therefore, exposure to cyber threats is significantly reduced as the absence of sensitive data on your server means that there is *virtually zero* risk of a security breach.

Provided the application leverages a modern framework like Next.js, the codebase is also protected from unauthorized access, as the site is rendered in prebuilt static HTML pages. Lastly, the site can then be replicated across a global, cloud-based Content Delivery Network (CDN), allowing it to remain functional and fast, even in the highly unlikely event a hacker is able to disable a server node.



What is “Headless”?

Ability to Scale Indefinitely

Headless Architecture allows a website to meet the needs of the business as they occur. This is due to the use of the CDN as well as serverless cloud computing, both of which are designed to handle massive spikes in bandwidth with ease.

To explain at a high level, your code is housed within third party cloud containers. As traffic increases (e.g. during a marketing blitz for Black Friday) the computational power of the cloud-based servers will increase commensurately to meet the need. Once traffic returns to normal volumes, the server resources required diminish. The increase in capacity will cost more but, unlike Monolithic Architecture, this increase in cost is temporary.

Adding new features is also faster with Headless Architecture as developers are able to leverage existing component libraries and third-party APIs, meaning they won't have to build features from scratch.



What is “Headless”?



Marketing and Sales Optimizations

In the past, a company could use SEO (“Search Engine Optimization”) to gain a competitive advantage. However, most current platforms are optimized out of the box, and additionally most digital marketers are all using the same SEO tools. So, in a world where brands all have access to the same tools, how can a business even expect to gain an edge over its competitors?

The short answer is *speed*.

Websites that leverage a JavaScript powered front-end will notice that their site loads instantly. This may seem trivial, but a fast website can actually outrank slower sites (i.e., a site with a perfect Google PageSpeed score of 100 can outrank a similar site with a score of 86). Additionally, navigating a very fast website is a pleasant experience for the user, which impacts conversions. In fact, conversion rates have been proven to directly correlate to the speed of a website. A 2019 study of over 100 million pages views across 20 different websites concluded that a site that loads in one second has a conversion rate 5x higher than a site that loads in 10 seconds².

In short, speed plays a critical role in search rankings as well as generating leads and revenue. Embracing Headless Architecture is vital to ensure a website performs optimally.

Human Resources Enablement

Headless Architecture can make it easier for HR teams to fill developer positions. Since all the services in a Headless Architecture are considered “pluggable” via APIs, a single programming language such as JavaScript can be used across the entire environment (from back-end to front-end). There’s no need to find (and frustrate) developers that need to use different languages to maintain the stack, making hiring and retention smoother.

² Michael Wiegand, Site Speed is (Still) Impacting Your Conversion Rate. April 20, 2022. Accessed March 27, 2023.
[https://www.portent.com/blog/analytics/research-site-speed-hurting-everyones-revenue.htm#:~:text=The%20first%205%20seconds%20of,\(between%20seconds%200%2D5\)](https://www.portent.com/blog/analytics/research-site-speed-hurting-everyones-revenue.htm#:~:text=The%20first%205%20seconds%20of,(between%20seconds%200%2D5))

Embracing Change

There is no getting around it – change can be *hard* - most people prefer to stick with what they find comfortable. Monolithic Architecture may seem like the “safe” play for a new or existing application – it’s how we’ve done things for the last two decades after all. However, particularly in the world of technology, change aversion carries the risk of becoming outdated.

It's important to weigh the potential benefits of adopting Headless Architecture versus the opportunity cost of pursuing a Monolithic approach. The former will allow the business to remain agile and responsive to change, expedite the application’s time to market, improve performance and customer/developer satisfaction, and keep the focus of the business on growth and the future. The latter (while more comfortable) will make change difficult, hinder growth, and create missed opportunities.



A Word of Caution

You should now have an understanding of how Headless Architecture combines services; much like putting together a Lego® set, selecting best-of-breed components for a successfully ‘composed’ stack.

To build upon the Lego® analogy, when considering a Headless Architecture, you need to lay all of your pieces out beforehand and make sure you’ve selected all the right pieces. Failing to sufficiently plan could result in selecting a suboptimal technology that makes up the tech architecture stack. The result would be an application that doesn’t meet the project’s requirements, or one that cannot accommodate changing business needs.

It should also be noted that the more complex the project, the initial development costs will likely be higher than a Monolithic project, but the long term benefits of Headless Architecture will lessen the blow of this cost. Headless Architecture does require resources to develop and replace the existing architecture. For these reasons, it is best to work with a provider experienced in the Headless Architecture ecosystem. 9thCO has been in the web development industry for over a decade; during this time we’ve completed many Headless projects with clients located all over the world.

Want to find out more?

9thCO is an experienced web application development and digital marketing agency, based in Canada and servicing clients globally.

[Get in touch](#)

