storyblok

Headless CMS - How Does it Solve the Content Problem?

Abstract: The exponential growth of different digital channels and omnichannel publishing, coupled with the growing importance of content as the building-block of digital experiences, have led to a never-before-seen emphasis on the issues surrounding the creation and management of content. This whitepaper explores the most crucial problems documented by different reports, addresses them by defining their root cause, and aims to offer a substantial solution backed by data.

1. Content and managing digital experiences

Why is content so important?

At this point it is perhaps not necessary to emphasize how important content is, as it has been dominating online businesses for a while. However, it is still fascinating to see how recent data still supports the old "content is king" slogan. While in 2019 23% of companies reported having no content marketing strategy, the number shrank to 16% in 2020.[1] It is also worthy to note that in 2020 a total of 70% of marketers were actively investing in content marketing.[2]

Here's the interesting part: despite the obvious awareness of content's importance and the resources spent on its creation, **29% of companies reported no further success in 2020 compared to 2019**. It is also interesting to see that even in 2020, 38% of companies judged the performance of their content strategy to be average and/or below.[3]

So, do we know where this dissatisfaction comes from?

Inadequate experience

A collaborative report published in 2021 has an answer to the question of the root cause:
After surveying 400 digital customer experience executives, the report identified the top "digital customer experience challenges":[4]

1 - Limited budget/resources
2 - Siloed systems and/or fragmented customer data
3 - Limited cross-department alignment/ collaboration
4 - Outdated/limited technology, operations or processes
5 - Lack of in-house expertise/skills

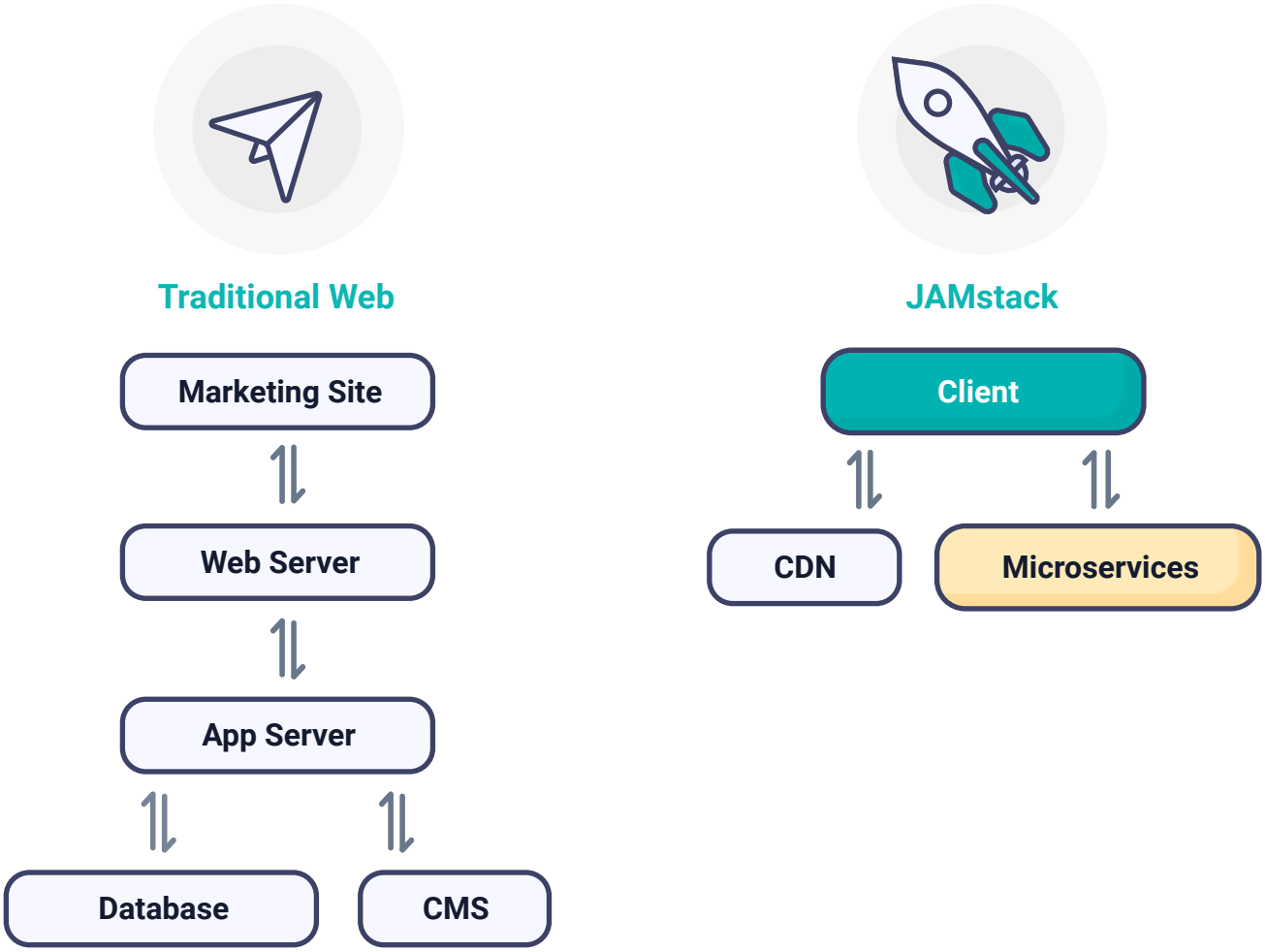
While 2 of these issues are directly resource related (1 and 5), the other 3 (2, 3, and 4) can potentially be addressed without having to waste any of the already-tight budget. These 3 points can be traced back to the way content is organized and distributed, or simply **the way a company's CMS (content management system) is set up**.

Let's discuss the relationship between these 3 points and the CMS in detail.

2. CMS and the organization of content

In the early years of the internet, a new webpage had to be created from scratch by either a developer or someone with a certain amount of coding experience. The early CMSs such as Drupal or Wordpress were created as a means for non-technical users to be able to publish content on a website. These CMSs are known as **traditional** or **monolithic**, referring to the way they handle content management.

These monolithic systems put a user-friendly graphic interface in front of the underlying code, meaning that the content and the code are still bound together, however non-technical users are now capable of publishing content. In other words, **a traditional CMS is called monolithic, because the back-end and the front-end are tightly linked together, forming a single monolith.**



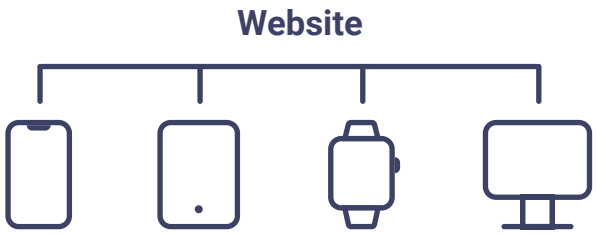
This approach was perfectly fine in earlier days when the connection to the internet was almost exclusively a desktop capability, and the content output of those websites were not particularly high. However, the users' interaction with the internet has changed dramatically since then. **By 2019, more than 50% of worldwide traffic came from mobile devices**, a 222% increase from just 7 years before in 2013.[5] People not only started to spend more time on the internet from different devices, but they also started to do online shopping from multiple devices. Business Insider reports that in 2019, more than 25% of total eCommerce sales came from mobile devices, a number that they predict is going to reach 44% by 2024.

This is where monolithic systems started to bend under the pressure and give way to the 3 major pain-points mentioned before:

1. Siloed systems and/or fragmented customer data

Monolithic systems were built specifically for content management on websites. With the rise of mobile devices and their popularity came the need for additional CMSs just to publish the same content across different platforms.

Take a typical online store as an example; while the business might have started with a single website catering to desktop users, gradually they have had to make changes. Now they need to optimize their website for different devices and screen sizes, apps for phones and tablets, displays on a kiosk or a screen somewhere, and not to mention running multiple accompanying or complimentary websites.



Each of these platforms had to be maintained separately. While the content being published in all of them might have been exactly the same (same product infos with the same information), due to the necessity of maintaining multiple CMSs the content had to be published and managed separately, leading to countless hours spent on duplications. This is what "**multiple content siloes**" or the "**siloed system**" refer to.

There are many reasons why companies try to avoid having multiple content siloes, for example:

- They lead to the fragmentation of the user/customer data, making it difficult to have clear analysis and subsequently hinder progress
- They slow down the whole process considerably
- They act as barriers to a seamless user experience
- They increase the chances of a security breach
- They make it harder to apply the same quality standards across the board

And as reported by those 400 executives mentioned before:

- **They discourage collaboration** (see next part)

2. Limited cross-department alignment/collaboration

This problem is the direct result of the previous one, a failure of monolithic systems to confine all the content in a central hub. As more and more silos are required to manage the content across different platforms, the cross-department collaboration becomes harder to maintain. In more extreme cases, the content pipeline can get so convoluted that collaboration can even become an issue within a single team

A few years ago, **Harvard Business Review reported that 75% of cross-functional teams are dysfunctional**. The study was led by a group of academics, including Stanford University's Behnam Tabrizi, and investigated 95 teams in 25 corporations.[7]

Let's imagine that an online store is about to offer a new targeted product. It doesn't really matter what the product is, it can be a t-shirt, a poster, or a gadget for a specific device. What is important is that our hypothetical product is created as an answer to a specific problem/demand. In situations like this, **the key is the speed with which the company moves from an abstract idea to a physical product ready to ship**. Now imagine for a second that the

product designers fail to communicate and work properly with the production line, the marketers launch their campaign too early and waste the initial potential, and the developers are placed so deep down the pipeline that all they can manage is a generic product page.

While a lack of collaboration itself is a huge problem for any company, it also results in its own challenges. The first one being a noticeable drop in both the productivity rate, and also the quality of the output. Retrieving information becomes time consuming, and subsequently, content repurposing degrades into simply producing flawed duplications.

The second one is that as silos grow and collaboration shrinks, more blindspots start to show up. This will eventually lead to lost potentials in areas that can be easily tackled, but are hard to spot because of the tangled web of content production.

3. Outdated/limited technology, operations, or processes

This problem directly points to the architecture of monolithic systems, so let's take a closer look.

As explained before, monolithic systems aim to offer everything in one single package (**back-end** and **front-end** capabilities). This initially sounds like a great solution, as everything is tied together and the companies have to only deal with a single vendor. This is why the monolithic systems are also known as **all-in-one suites**.

However, there is a huge drawback inherent in an all-in-one suite: **one does not have complete freedom to choose what goes in (and what goes out) of the package**. This means firstly, you are going to end up paying for some tools and technologies that you do not need, or even use. Secondly, your capability to quickly adjust to new technologies and benefit from early adaptation is dependent on the suite provider's decisions. In best case, you will end up receiving the newest technologies a bit later than you prefer, and in worst you may never get to have them.

While small-scale users may not be affected by these shortcomings, **they will definitely leave a mark on enterprise cases**. For example, in the competitive eCommerce scene, early adaptation to the latest trends makes the difference between success and failure. Since many of these trends require newer technologies, anything short of a swift action can have negative consequences.

How did the CMS pioneers respond? By going to the heart of the problem

At the center of all problems with monolithic systems is the way the architecture has been set up from the beginning. **The coupling of the front-end and the back-end was a necessity of it's time, and an obstacle in today's world**. So when developers looked for a solution, they aimed at the heart of the problem: the architecture.

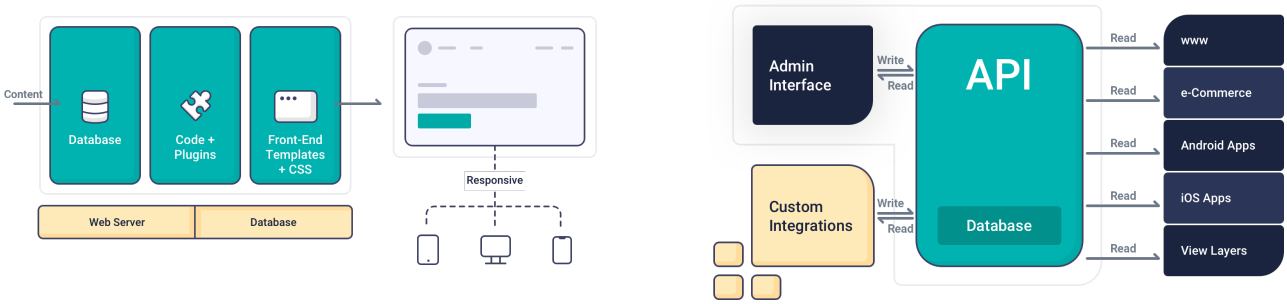
By breaking the link between the front-end and the back-end, most problems associated with the monolithic systems would be solved. This effort eventually led to a completely new way of managing content. These new systems were called **"headless"** specifically because of their architecture.

3. What is a Headless CMS?

A headless CMS simply cuts the link between the front-end and the back-end. With this approach, you can use the same core "body" (back-end) to create as many "heads" (front-ends) as you may need. This is where the name "headless" comes from, as there is no single fixed "head", but a countless number of different "heads" (**websites, phones, voice-activated assistants, smart watches, VR/AR technologies, etc.**) which can be put on the "body" based on your needs.

The back-end acts as a central content hub which hosts every piece of content that is going to be published on any platform. The content is delivered to each platform via **APIs**, which is why these systems are sometimes called **API-first**.

Monolitic vs Headless



Now, let's go back to the 3 major problems that monolithic systems were causing, and see how a headless system easily solves them:

1. With a headless CMS, any form of content **has to be created only once** in the back-end, and can be simultaneously published in multiple platforms. This obviously means that the **problem with multiple silos is immediately solved**, as content is stored in a central hub.

2. As content is no longer scattered across multiple silos, cross-department alignment improves considerably. Furthermore, customer/visitor data is also concentrated in the **central content hub**, instead of being fragmented in different silos.

3. The third major problem associated with monolithic systems was having to deal with outdated/limited technologies, something that we previously described as a result of the **all-in-one suites**. The headless architecture results in a fundamentally different approach to building a technology stack, one that is usually called **best-of-breed**.

Instead of being locked in a pre-fixed package, headless systems encourage building your own individual technology stack based on your specific needs. **This means having absolute control over which tools your team would be using, and which ones they do not need (anymore).**

A best-of-breed solution has many positive outcomes, for example:

- You can choose the technologies that you really need at the moment, and reduce your costs as a result.
- You are free to choose the technologies that your team already has experience with and/or is more comfortable with. This reduces the learning curve for the whole team considerably, saving time and other resources.
- As each tool is offered by an individual expert provider, updates come regularly and in small sizes. Since each technology is regularly updated with small batches, it is easy to stay ahead of the curve and quickly respond to newer trends (as opposed to the large-scale updates of monolithic suites). While regular updates for every technology reduces the overall security risk, if a problem shows up somewhere in the stack, it will not affect other parts of the system (**as opposed to monolithic suites, where a single breach can affect the whole stack**). Isolated issues have much lower risks and are easier to control.

We have previously written extensively about the nature of best-of-breed and all-in-one solutions, and mapped out their differences in detail. You can read the whole article here.

While headless systems address the 3 main problems with monolithic systems, they go even further in empowering content creation and management. In the next section, we will talk more about these other benefits.

4. Benefits of a headless system

Headless architecture's unique approach to content management benefits different user groups in different ways.

"Companies that take a headless approach to commerce (and even CMS) gain the agility to keep up with customer expectations, put their own unique touches on front-end experiences, and even expand into new channels and markets."
- Forrester, Jul. 2020

Let's take a closer look:

A. Developers

More flexibility: Because the front-end is handled separately, **developers are free to choose their favorite framework and tools**. This is in direct opposition to the monolithic approach, where the tools are dictated by the suite provider. Additionally in headless systems, the independence of both ends also means that any new design can be applied and implemented quickly.

More focus: Unlike monolithic systems, here developer teams can dedicate their full potential to the presentational layer: better aesthetics, superior user experience, and seamless accessibility. This developer-first experience is achieved thanks to the way content is managed in headless systems. Since content is delivered from the central hub in a "raw" format, it is completely up to the developers to decide how it is presented to the users.

Simpler maintenance: The separation of the front-end from the back-end completely changes the way updates work. In a headless CMS developers do not have to worry about the constant stream of updates that are associated with monolithic systems. With a headless system, updates are installed automatically and developers do not have to stress about uptime or broken updates, as is often the case with monolithic or on-premise systems.

Better security: With monolithic CMSs, developers usually end up with bloated code bases, containing considerable amounts of data. These huge code bases are necessary since the whole operation (front-end plus back-end) needs to be addressed in the same manner. Since headless systems separate the two ends, the code base ends up being noticeably smaller, as the content from the back-end is simply delivered with APIs. This in turn makes your code base a smaller target for security threats.

When Fundbox decided to move away from a traditional CMS to a headless solution, they chose Storyblok. The migration process to Storyblok took only 5 weeks. The previous migration to a monolithic system lasted 4.5 months.

B. Marketers

Independence: Headless systems empower editors and marketers to work independently from developers. Creating and updating landing pages only take minutes and they can go online effortlessly. This independence also enables marketing teams to respond quickly to any new trend, without having to worry about bottlenecks in other departments.

Omnichannel-first: The headless architecture is meant to be used for omnichannel projects. With the growing connectivity of different devices to the internet, and with the diverse platforms that consumers choose to buy products on, it is absolutely necessary to publish content simultaneously across all platforms. A true omnichannel system should

not only support desktop computers and phones/tablets, but also the emerging technologies such as voice activated assistants or VR headsets. Headless systems can easily support omnichannel publishing across all platforms, by separating the presentational layer from the back-end.

Personalized user experience: Traditionally, monolithic CMSs offer pre-built components for the presentational layer, meaning everything on the users' end would come from a generic template with only minor customization possible (**everything from the interface layout, to product categorization, and even contact forms**). A headless CMS on the other hand gives absolute freedom and flexibility to create each component with an endless possibility for customization. This means that your platform and every single bit of its components will be unique to your brand.

SEO and faster user experience:[9] One of the most important criteria in SEO is the website's response and loading time. Search engines prioritize websites with the shortest loading times, just as customers themselves do. For example, Google's emphasis on load times when it comes to ranking results is easily justified by their research, which shows that 40% of users exit a website if it takes more than 3 seconds to load. A headless CMS drastically reduces a website's loading time, separating the back-end traffic from that coming from the front-end. This simple fact can have considerable consequences.

UPC Business saw an 81% reduction in their new website's loading time by choosing Storyblok, a headless CMS. As a bonus, the whole process only required 3 developers! You can read more about their case here.

Another major factor in SEO is security, or as google marks it, a "top priority".[10] A central content hub by definition is much safer than having your content scattered around in multiple silos. Additionally, any threat to the front-end stays isolated and is not going to be a problem for the back-end.

C. Project Managers

Content automation: One of the major benefits of headless systems is their ability to take advantage of AI capabilities to drive content automation **while keeping a high quality**. In a monolithic system, an editor is forced to create full-fledged (finished) pieces of content every time. This means content creation either has to be from the ground up all the time or turn into uninspired copy-pasted material to save time.
An ideal headless CMS takes advantage of modular content blocks. These blocks of content are created and then stacked together as modules, in order to create new content. These blocks have the ability to be customized to any degree, and be connected to each other in any way. With the new system, marketers can create small blocks of content which can be stacked on top of each other to form different finished products. Additionally, modular content blocks make it possible to implement Inteligent Content and take content automation to the next level.

Low-maintenance infrastructure: Most headless systems take care of all the heavy lifting concerning the infrastructure and updates, leading to considerably less need for maintenance on your side. This means that you can reassign your developers to use their time for creative tasks, such as creating unique user experiences and personally tailored user journeys, instead of mundane maintenance work.

Scalability and future-proofing: The best-of-breed approach means you do not have to worry about sudden scaling issues, either as a result of unexpected traffic, or sudden shifts in strategy. **Since each technology comes as an individual component of the stack, scaling can be done quickly and without complications**. As the company expands , there is no need to constantly jump from one monolithic suite to another, instead you can simply scale as you grow.

Likewise, by having a flexible stack of technologies that can be modified at any moment, you can easily prepare for the unknown future. Any emerging technology can be easily added to the stack, without having to wait for a suite-provider to add the technology, or having to change other parts of the stack. As IoT grows and people continue to diversify their means of online presence, being able to quickly adjust to the newest trends is even more essential than before.

Resource management and cost reduction: All of the points mentioned above, including the benefits for developers and marketers, directly affect the amount of resources you have to spend to run your operations. Some, like content automation and simple maintenance, directly reduce your costs, while others, like faster user experience and omnichannel, result in more revenue/value.

Plus, the best-of-breed nature of headless CMS means that you can **"pay as you go"** with your services. Since a monolithic suite has to be paid all at once, migrations tend to be extremely costly and risky. With the best-of-breed model, you can start small and add more gradually as you move forward.

After moving to Storyblok, EF experienced 2-3X faster time to market.

5. Beyond headless: Storyblok as an agile digital user experience manager

There is a large variety of headless systems to choose from, each with their own specific focus. While many offer content management tools, some are moving forward and changing the way users experience content altogether.

In a 2021 report by Forrester, 3 criteria are marked as "key differentiators" between agile CMS systems:[11]

1. Omnichannel delivery
2. Developer tools
3. Collaborative planning

According to Forrester, these 3 criteria "will dictate which providers will lead the pack".

Storyblok has been offering unique digital experience management capabilities by going beyond what a typical headless CMS offers. Storyblok's modern approach to content management directly corresponds to these 3 criteria:

1. Omnichannel delivery

In addition to all the usual headless benefits, Storyblok takes the component-based content management to a new level by re-structuring

content into various forms of modular blocks. This makes the content creation (and editing) process even easier, resulting in lightning-fast publication of content across all digital touchpoints. Storyblok's internationalization and localization capabilities let you organize your content under one tree and distribute them quickly and seamlessly for your international audience. Storyblok does not charge extra for its internationalization capabilities!

Storyblok's one-of-a-kind **Image Service** allows you to optimize your images on the go, reducing the strain on your website and considerably lowering its load-time. The website and its images will be optimized for any device, with any screen size.

Visit here to learn more about Storyblok's omnichannel capabilities.

2. Developer tools

In addition to having comprehensive developer guides available, Storyblok keeps an engaging open community across different platforms, where developers can easily solve their issues.

Storyblok's own App Directory lets developers selectively add features to Storyblok. This keeps the UI clean and optimized. Additionally, developers can also build their own apps to be used.

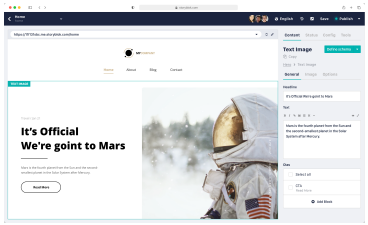
When it comes to technologies, Storyblok integrates with every framework so that developers are free to choose the best fit for their projects. The Technology Hub offers selected beginner tutorials, videos, boilerplates, and even cheat sheets all in one place, while more detailed guides and tutorials are available for further information. Lastly, Storyblok offers a completely free trial where developers can get hands-on experience before making any decisions.

3. Collaborative planning

Storyblok's **Content Workflows** let you define multiple workflows for your content and deploy only approved changes. **Workflow Stages** allow your team to use different stages like Drafting, Reviewing, or Ready for Publish. Additionally, **you can also define your own custom stages for maximum efficiency**.

Finally, Storyblok moves your marketing team one step ahead through it's **Visual Editor**. It is the simplest, yet most powerful editing experience that editors and marketers will get. The Visual Editor enables your editors to edit their content with an in context preview. This includes the possibility of clicking on components of the website and editing them in seconds.

Watch the Visual Editor in action!



Curious if a headless system is the right choice for you? If you want to dive deeper into the world of headless content management:

1. Take advantage of Storyblok's free trial and see how the process works.
2. Talk to specialists who can help you decided if you can benefit from one.
3. Learn more about content management by visiting our Guides Hub.

Endnotes

1. Semrush, Content Marketing Statistics You Need to Know for 2021. Jan. 2020, https://www.semrush.com/blog/content-marketing-statistics/

2. HubSpot, State of Marketing. 2020, https://www.hubspot.com/state-of-marketing

3. See 1

4. CMSwire, The State of Digital Customer Experience - 2021. 2021, https://www-cmswire.simplermedia.com/cw-cp-smg-dcx-rpt-2021-cx.html?_ga=2.204523280.1441690767.1614023096-437700128.1605213905

5. BroadbandSearch, Mobile Vs. Desktop Internet Usage. 2020, https://www.broadbandsearch.net/blog/mobile-desktop-internet-usage-statistics

6. Business Insider, Rise of M-Commerce: Mobile Ecomerce Shopping Stats & Trends in 2021. Dec. 2020, https://www.businessinsider.com/mobile-commerce-shopping-trends-stats?r=DE&IR=T#:~:text=M%2DCommerce%20Statistics,e%2Dcommerce%2C%20in%202024.&text=Business%20Insider%20Intelligence%20expects%20%2429,2024%20%E2%80%94%20an%2018.9%25%20CAGR.

7. Harvard Business Review, 75% of Cross-Functional Teams Are Dysfunctional. Jun. 2015, https://hbr.org/2015/06/75-of-cross-functional-teams-are-dysfunctional

8. Forrester, Digital Experience FAQ: Do I Need To Move To Headless Commerce?, Jul. 2020. https://www.forrester.com/fn/dvrYIEoIAOVLWSYgpsON4

9. Google, Find Out How You Stack Up to New Industry Benchmarks for Mobile Page Speed, Feb. 2017. https://www.thinkwithgoogle.com/intl/en-ca/marketing-strategies/app-and-mobile/mobile-page-speed-new-industry-benchmarks/

10. Google, HTTPS as a Ranking Signal. Aug. 2014, https://developers.google.com/search/blog/2014/08/https-as-ranking-signal

11. Forrester, The Forrester Wave: Agile Content Management Systems (CMSes), Q1 2021. Feb. 2021, https://www.forrester.com/report/The+Forrester+Wave+Agile+Content+Management+Systems+CMSes+Q1+2021/-/E-RES161667?objectid=RES161667