

# The Ultimate Guide to Headless CMS





# **The Ultimate Guide to Headless CMS**



# The Ultimate Guide to Headless CMS

*Everything you need to know  
to choose the right CMS*





TL;DR (If you read nothing else, read this)	6
<b>Chapter I: The world has changed and so must the CMS</b>	<b>9</b>
<b>Chapter II: What is a cloud-first headless CMS?</b>	<b>17</b>
<b>Chapter III: Choosing your hosting model</b>	<b>33</b>
<b>Chapter IV: When to use a headless CMS</b>	<b>41</b>
<b>Chapter V: Increase your digital agility with a headless CMS</b>	<b>51</b>
Summary	59

## TL;DR (If you read nothing else, read this)

### Having a responsive website is no longer enough

Your customers are increasingly using mobile apps, virtual reality, augmented reality, chat, voice-controlled digital assistants, smart wearables, and various Internet of Things devices. If you're not connecting with them through these new channels, you're missing an opportunity.

### Traditional web-oriented Content Management Systems aren't ready for this

You need a new type of CMS that was built for omnichannel content management and delivery from the very beginning—a headless CMS. Headless CMS focuses on content management and provides an Application Programming Interface (API) that makes the content available on any device and any platform.

### Cloud-first headless CMS combines agility and flexibility

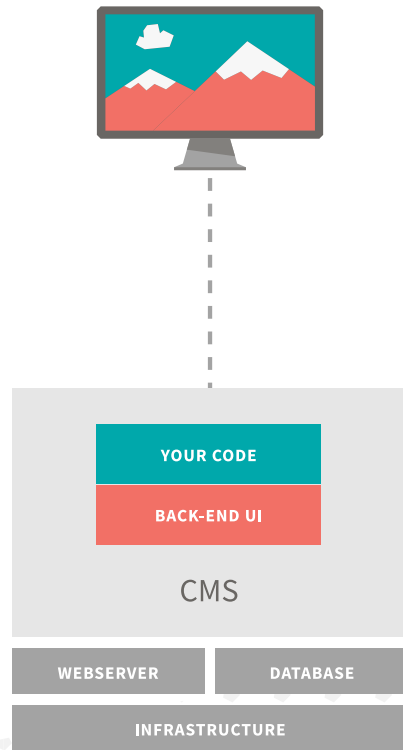
As part of your digital transformation, you need to be much more agile and test new ideas in a matter of days, not months. An on-premise CMS may slow you down as you find yourself waiting for IT to provision the infrastructure.

The headless architecture enables vendors to provide a CMS in a true Software as a Service model, which wasn't possible before, which means you can get started immediately and quickly deploy your applications.

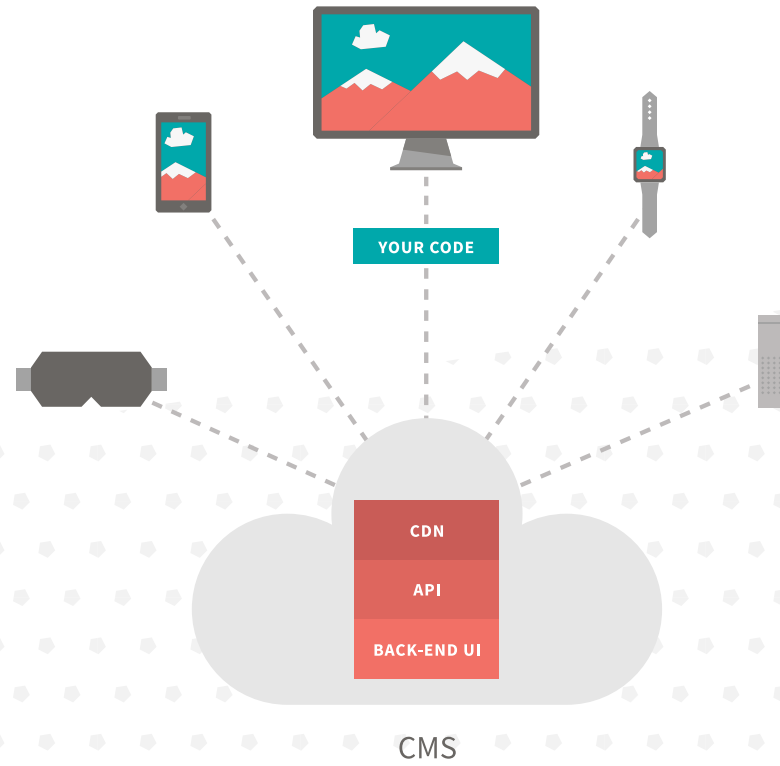
### Stay ahead of your competition

Cloud-first headless CMS is the next generation in content management for brands that want to stay ahead of the curve by engaging customers in the growing number of channels. This book will help you make the right decisions and choose a solution that will help you stand out.

### Traditional CMS



### Cloud-first Headless CMS



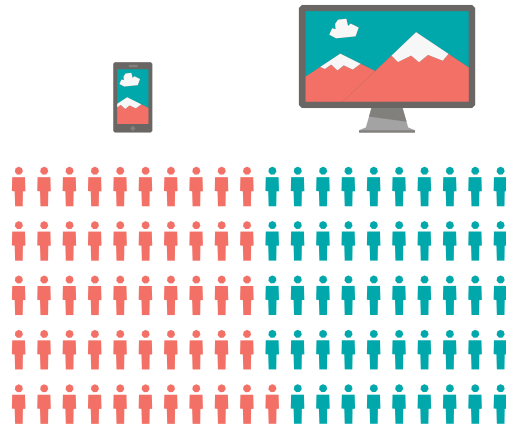


A scenic view of a snowy mountain range with a wooden structure in the foreground. The mountains are covered in snow and have some rocky peaks. The sky is a pale, hazy blue. In the foreground, there is a wooden structure, possibly part of a building or a bridge, with a dark circular object (possibly a light fixture) visible. The overall tone is soft and natural.

Chapter I

# The world has changed and so must the CMS

# 1 We now live in a multichannel world



## 51.3% mobile

51.3% of people now spend more time searching the Internet on mobiles than on desktops.

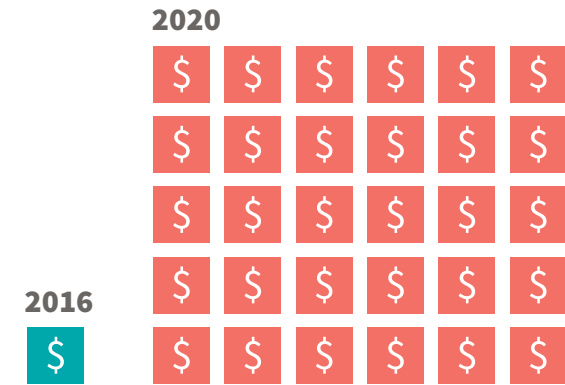
Source: StatCounter (2016)



## 3.5 devices per person

US Consumers use on average 3.5 connected devices each.

Source: Consumer Barometer (2016)



## \$5.2b to \$162b

Augmented reality and virtual reality (AR/VR) market will grow from \$5.2b in 2016 to more than \$162b in 2020.

Source: IDC (2016)

## 2 We need a new way of managing content in this multichannel world

We've been using traditional web-oriented CMS systems for years. But the new multichannel era requires a different approach to managing content. We see four major trends that create the need for a completely new generation of content management.

### Multichannel

In the new multichannel world, your content may be consumed on any number of devices. If you do not want to miss the opportunity of truly engaging with your customers, you need to make sure your content is ready for multiple channels. This requires a whole new mindset in structuring, organizing, managing, and presenting content.

Unfortunately, traditional CMS systems were built with just websites in mind and, although they do support structured content, the multichannel approach is an afterthought for them and they do not support the new paradigm well.



## Digital transformation

Companies realize they need to adapt to the new digital economy and embrace digital as part of their core strategy.

This change is never straightforward as companies need to rethink and reinvent their business models. In many ways, they need to act like a startup. They need to quickly test new ideas, learn, and adjust.

This requires a high level of agility that is not well supported by today's large monolithic CMS systems that were designed to support traditional rigid business processes.



## Cloud

Companies, waking up to the realization they need to be agile, are increasingly moving their systems to the Cloud. They recognize that running systems internally is less efficient and lacks the flexibility and speed they need in today's dynamic world.

While traditional CMS systems can be hosted in the cloud environment, they weren't designed as Software as a Service (SaaS) products and lack the benefits of a true cloud-first solution.

## Microservices

Microservices represent one of the major shifts in software architecture. Today's systems are increasingly built as a combination of both internal and external services connected via their API.

In the past, companies used to build websites on top of monolithic CMS platforms which meant they were locked in with a single vendor. When they decided to change their CMS, they had to throw away their investment and start pretty much from scratch.

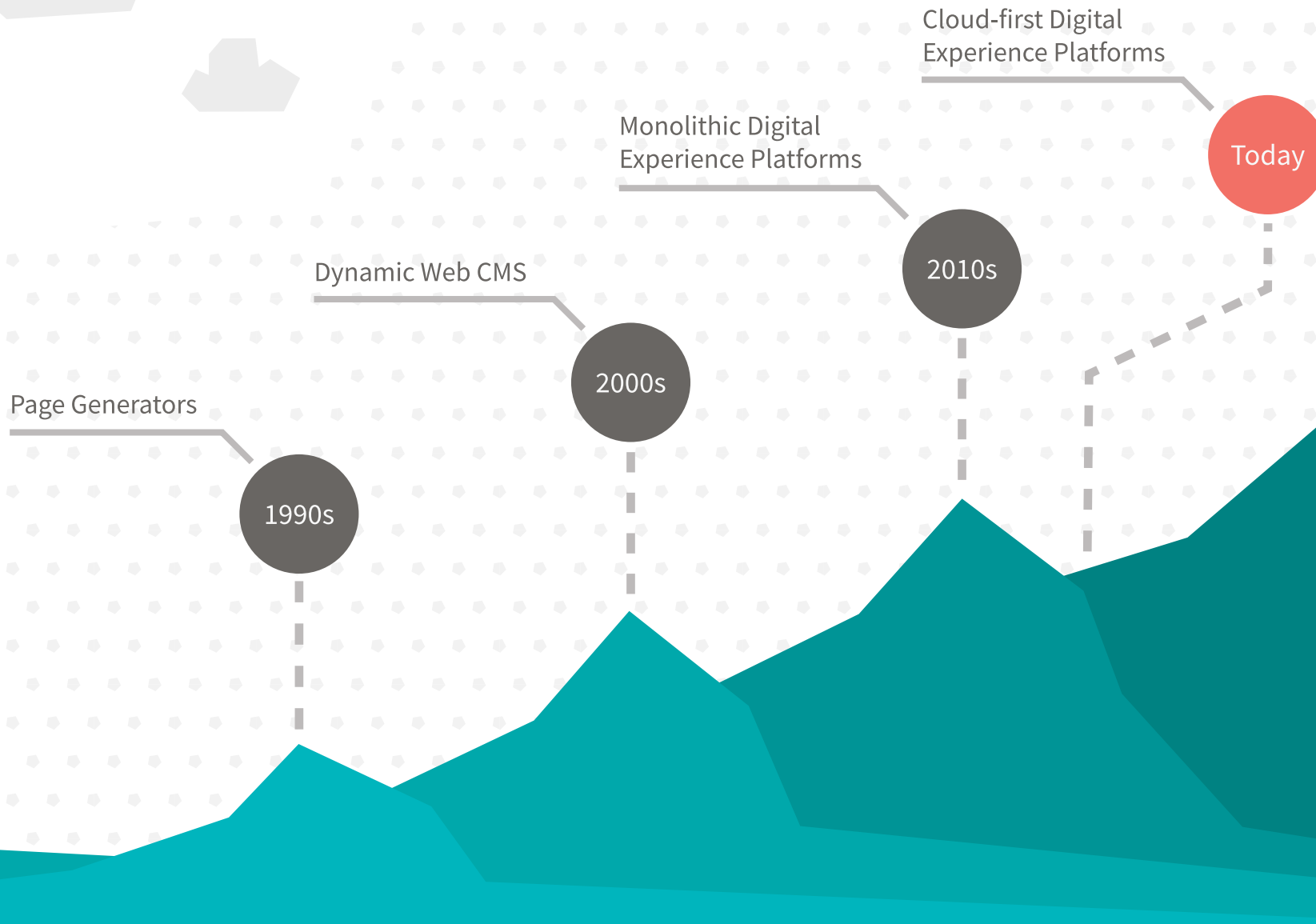
The new generation of CMS systems are built as "headless" or "API-first", which means they can be easily integrated into any application, no matter what technology it uses.



### 3 Cloud-first headless CMS: The next generation in content management

If you consider the four trends we've described, it's clear that the traditional web-oriented CMS is no longer enough. In order to stay competitive and connected with their customers, companies need a new generation of CMS—a cloud-first headless CMS. We're at the very beginning of this new era. If you're choosing a CMS today, you need to consider whether a monolithic digital experience platform or a new cloud-first headless CMS product is best for you.









Chapter II

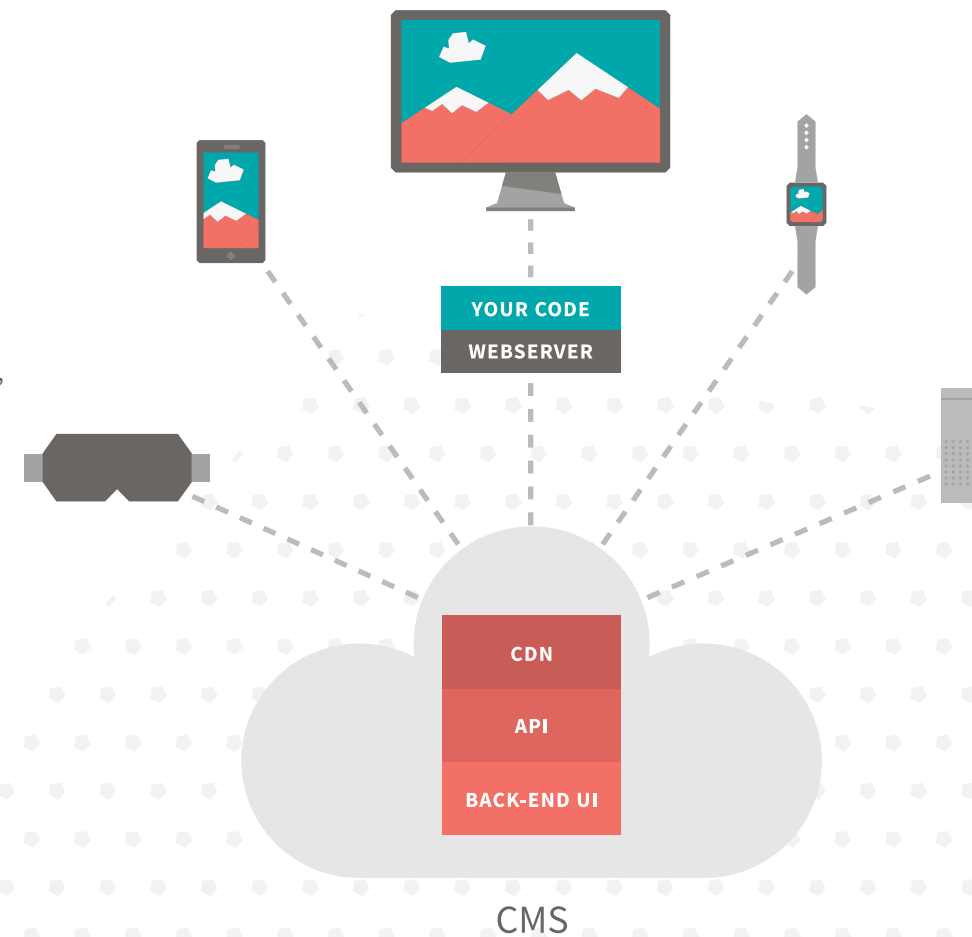
# What is a cloud-first headless CMS?

# 1 Cloud-first headless CMS in a nutshell

Headless CMS is a content management system that allows you to manage content and access it from your applications using an API.

Unlike traditional CMS solutions, headless CMS does without the presentation layer (the “head”) that would dictate how the content should be displayed. Instead, you control the presentation completely with your own code.

This not only enables a content-first approach to your engagement with your audience (as content creators no longer have to wait for development teams to catch up) but it means you can use the same content across multiple channels—website, mobile app, digital assistant, virtual reality, smart watches, etc.—making the headless CMS the ideal solution for the fast-paced multichannel world.



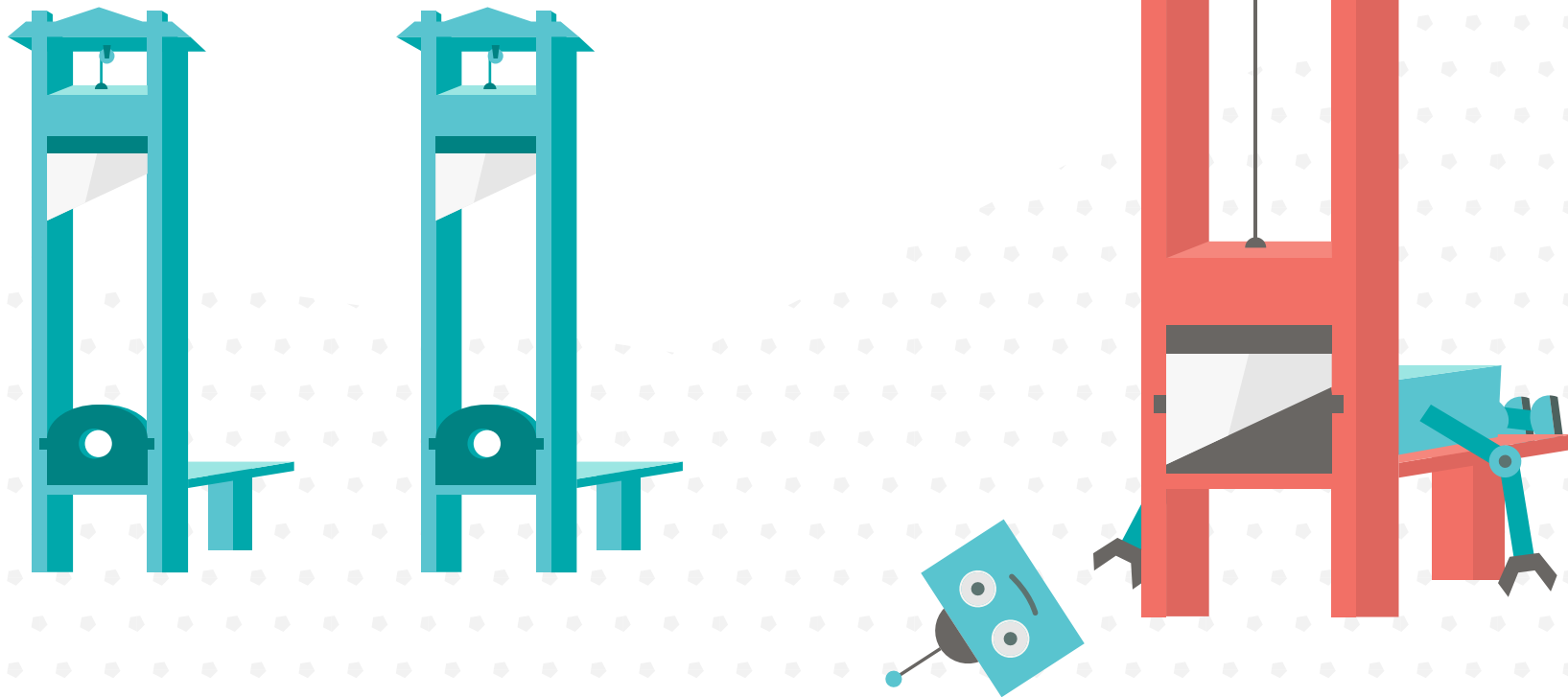
Theoretically, you can run an on-premise headless CMS, but you will get the most out of it if you use it as a cloud service. This is because the headless architecture allows CMS vendors to provide a true multitenant Software as a Service (SaaS) product and look after the CMS for you while you focus on your applications.

So the cloud-first headless CMS combines omnichannel content management and agile delivery within the comfort of a true cloud service.



## 2 Coupled, decoupled, or headless?

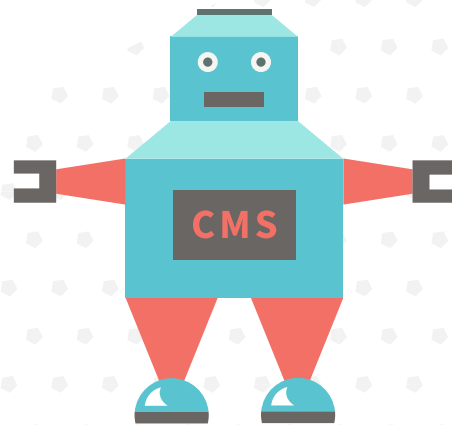
When you're choosing a new CMS, it's important that you understand the differences in various architectural approaches used by different products. While this may look like a technical detail, it has a big impact on how the CMS will support your business goals now and in the future.



## Coupled CMS

### Most traditional CMS systems are built as a coupled CMSs

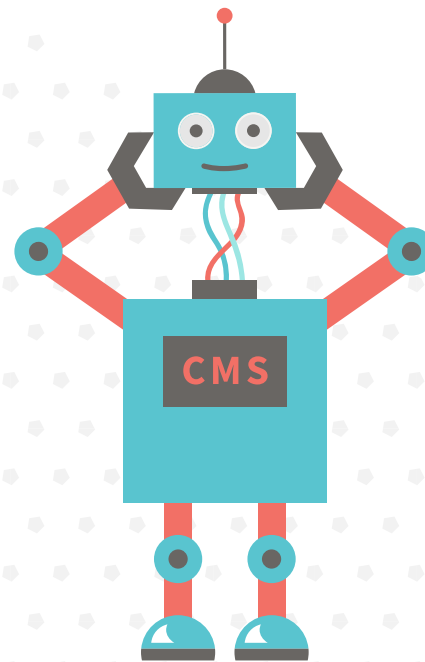
The coupled architecture combines everything in a single application: the content repository, back-end user interface for editors, templating system, as well as your custom code.



## Decoupled CMS

### A decoupled CMS for better separation of concerns

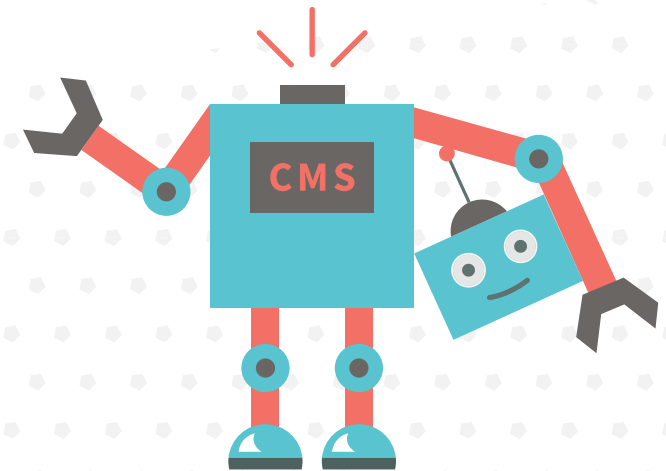
In this case, you make all modifications in the content management environment (typically behind the firewall) and synchronize the published content with the content delivery environment.



## Headless CMS (on premise)

### Headless CMS: no head, no limits

A headless CMS makes the presentation layer much more flexible as it eliminates the front-end part: by providing the content through its application programming interface (API), it can be used on any platform and on any device to retrieve and display the content.



### 3 Which architecture is right for you?

#### Coupled CMS

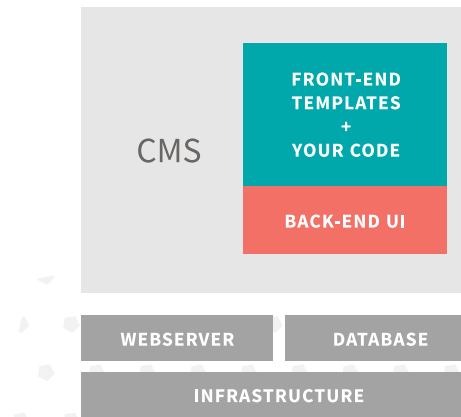
##### Pros

- This approach is very popular as it only requires a single environment and it's easy to set up and manage.

##### Cons

- The CMS code is tightly connected with your templates and custom code, meaning you need to look after your own code as well as the CMS code during the whole application life cycle (installation, upgrades, hotfixes, code versioning, continuous deployment, etc.).
- The CMS code is exposed on the public server which increases security risks.
- The load on your website impacts the performance of your content management back end, and vice versa.

While coupled CMSs may also provide an API for non-web channels, they weren't built as API-first solutions and do not provide the benefits of a true headless CMS (see chapter True headless-first versus "me too" headless CMS).



##### When to use

A coupled CMS may be a good fit for you when you plan to build a brochure website and you don't expect to publish to multiple channels.

## Decoupled CMS

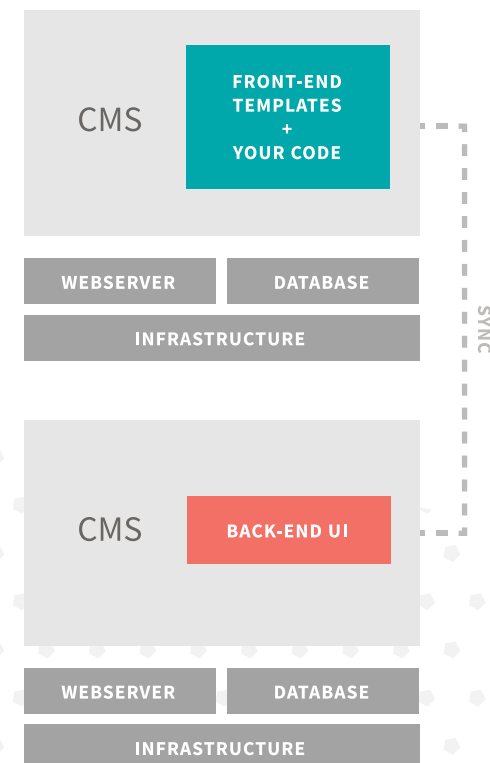
### Pros

- This model provides better security, higher (though not full) separation of your custom code from the CMS, and easier scalability.

### Cons

- You have to manage several environments, which multiplies the costs of infrastructure, ongoing maintenance, and software licenses.

The synchronization of content introduces potential issues, especially if your website enables members to contribute their own content that then needs to be synchronized back.



### When to use

The decoupled CMS enables similar use cases to the coupled CMS, just with a more robust architecture. However, by nature, it's still primarily a web-focused solution.

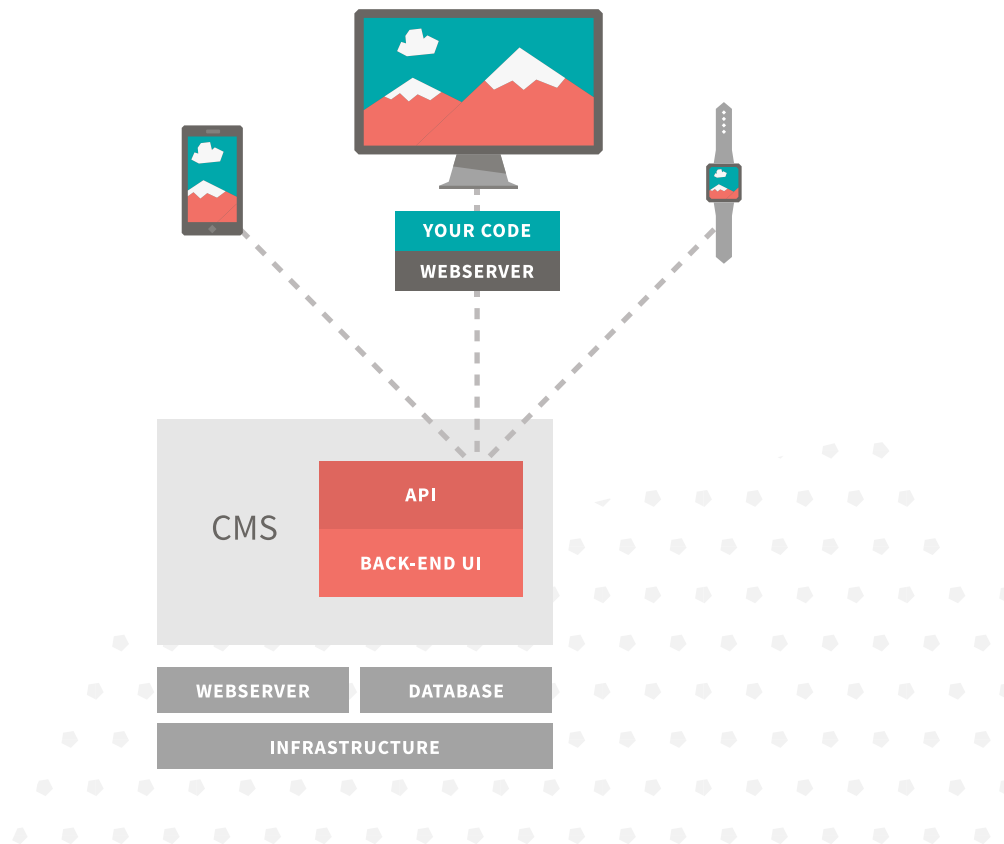
## Headless CMS

### Pros

- The API makes the content available through any channel and on any device and allows you to make the CMS part of your microservices architecture.
- You can write your websites or mobile applications using any programming language, your favorite tools, and your own development process.
- You have full control over the application lifecycle without having to interfere with any CMS code.
- It provides higher security and much easier scalability.

### Cons

- A pure headless CMS doesn't provide channel-specific support (especially for the web channel) which means developers may need to develop some web-specific functionality themselves.
- Marketers may be limited in what they can do with a pure headless CMS and rely more on developers in scenarios like creating a landing page with custom layout.



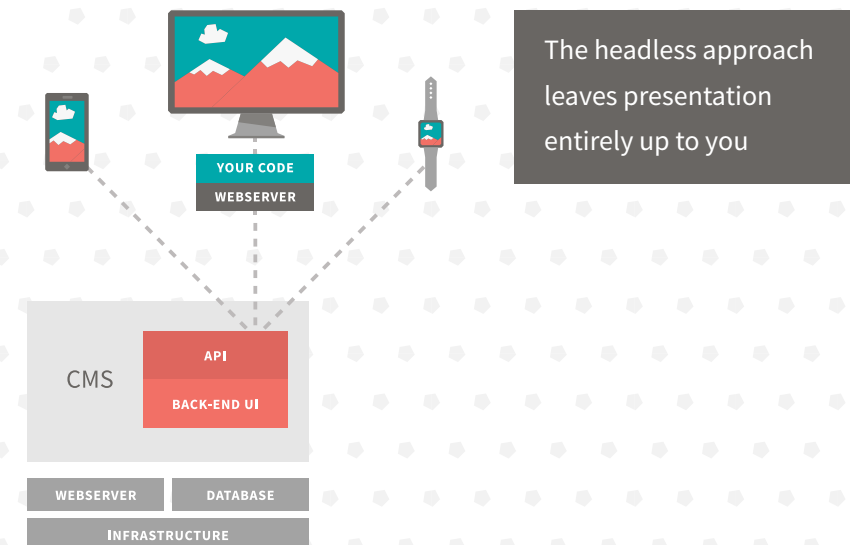
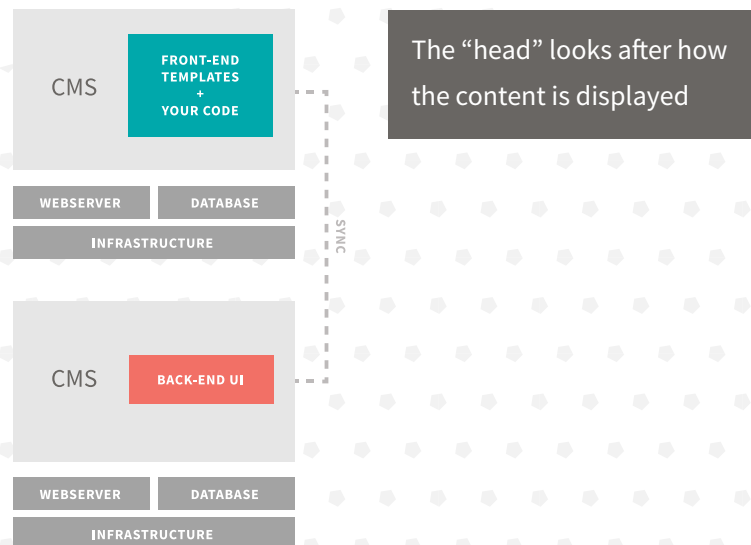
### When to use

If you plan on supporting multiple channels, a headless CMS is your best choice. You can also use it for traditional website projects—however, be aware of possible limitations when it comes to empowering marketers.

## 4 Understanding the differences: decoupled versus headless CMS

There's a lot of confusion around “decoupled” and “headless” in the market. People may tell you things like “a decoupled CMS is just another name for a headless CMS”. Unfortunately, that's a misunderstanding of the headless concept.

The decoupled CMS does separate the presentation layer (the “head”) and the content management back end, but it still uses a templating model and is responsible for how the content is displayed.



## 5 Understanding the differences: true headless-first versus “me too” headless CMS

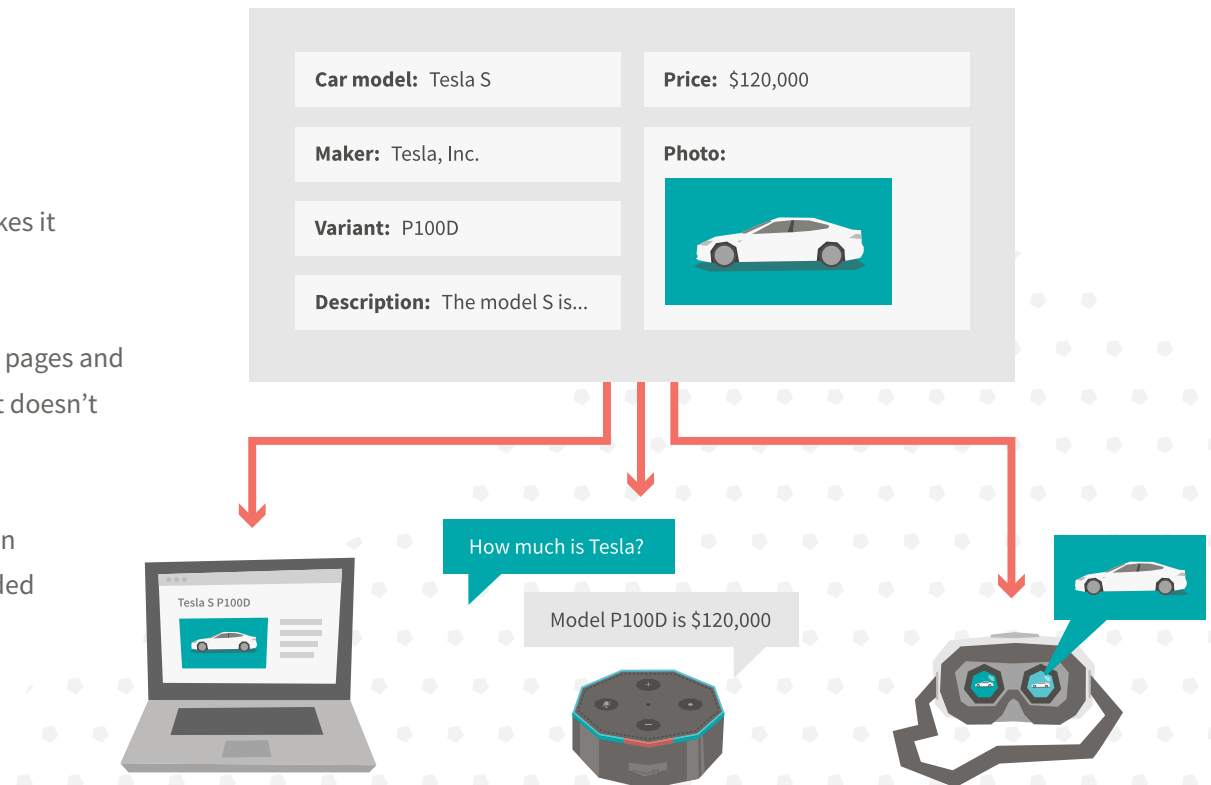
Many traditional coupled or decoupled CMS systems provide some kind of REST API and claim they are “hybrid” or “also headless”. You should be aware, however, of important details that make a big difference between a CMS that was designed with a headless or “API-first” approach in mind from the very beginning, versus a CMS for which the API was an afterthought.

### Content model

A true headless CMS works with all content in a way that makes it possible to use the content for any channel.

Traditional web-oriented CMS systems often use concepts of pages and drag-and-drop interfaces that create content in a format that doesn’t allow for content reuse.

Oftentimes you will discover that their REST API may not even make such content available or the way the content is provided makes it useless for other channels, like a mobile app.

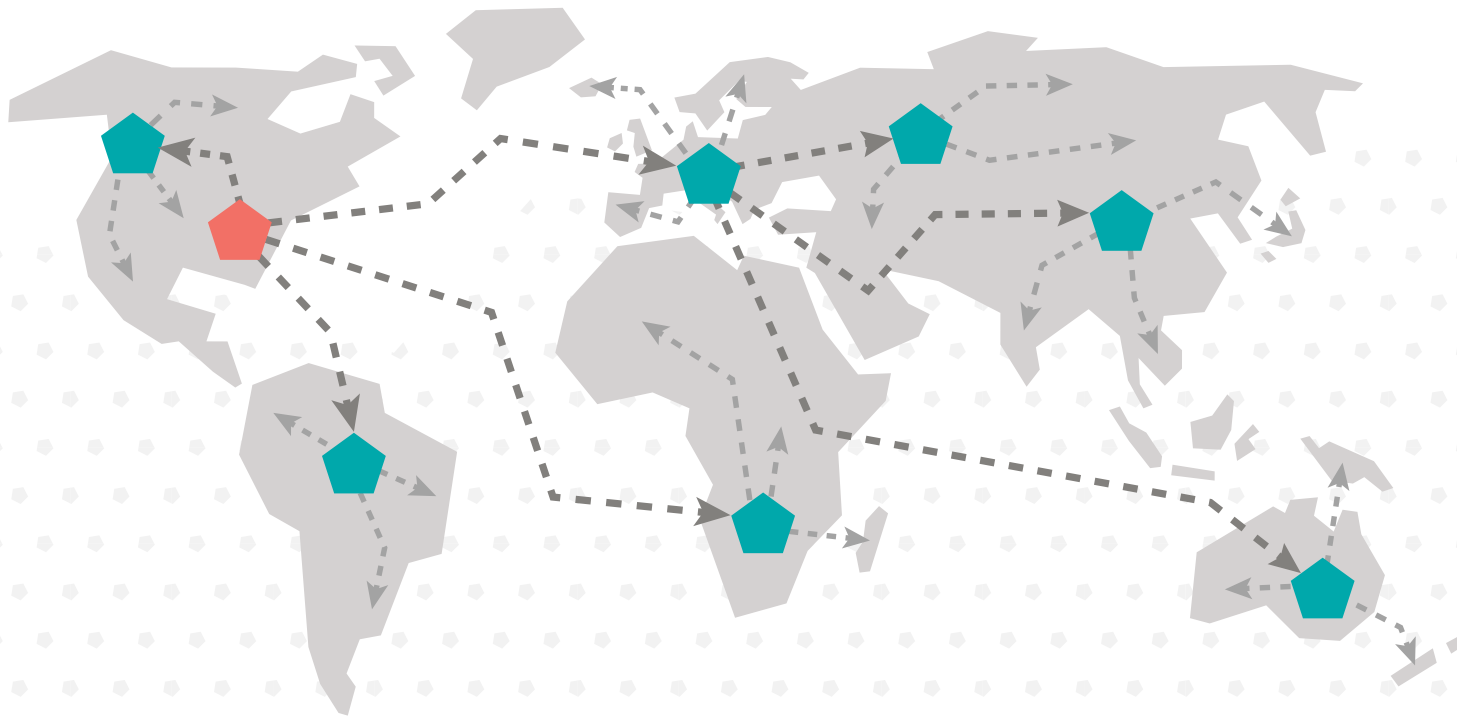


## Robustness and performance of the API

A true headless CMS was built with an API-first approach.

Its creators usually pay a lot of attention to the API and the whole solution is designed in a way that enables its full power.

Traditional CMS products often provide just a pure REST API, lacking the additional elements that ensure high scalability and security of the APIs in real-world applications. They usually don't provide a Content Delivery Network (CDN) to ensure global coverage, or high speed or high service availability, which has a considerable negative impact on its overall performance.



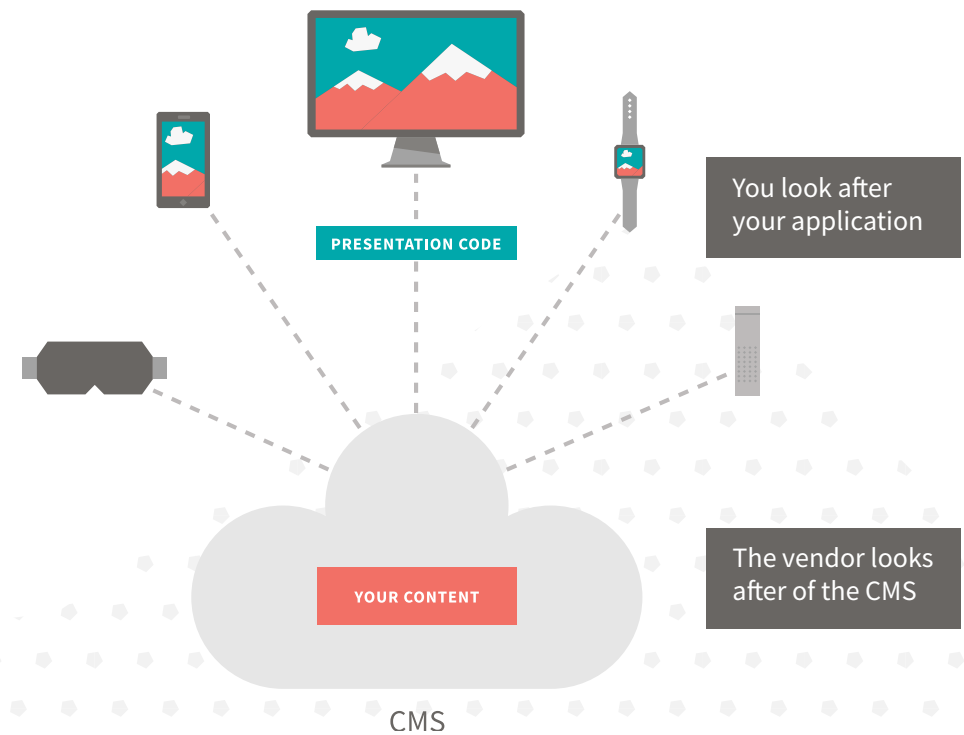
## 6 Understanding the differences: True SaaS CMS versus CMS hosted in the Cloud

### Software as a Service model

The most revolutionary aspect of the headless approach is that it enables CMS vendors to provide a CMS in a true multitenant Software as a Service model.

This wasn't possible with traditional CMS models that combined both content management and content presentation.

In fact, it's one of the main reasons traditional CMS vendors only provide single-tenant managed hosting or Platform as a Service (PaaS) hosting for their products. Such models, however, don't unlock all the benefits of the Cloud, as we explain later in the chapter "Choosing your hosting model".



## 7 Other options to consider

In order to give you a complete overview of your CMS options, we've included some alternatives that may fit certain scenarios.

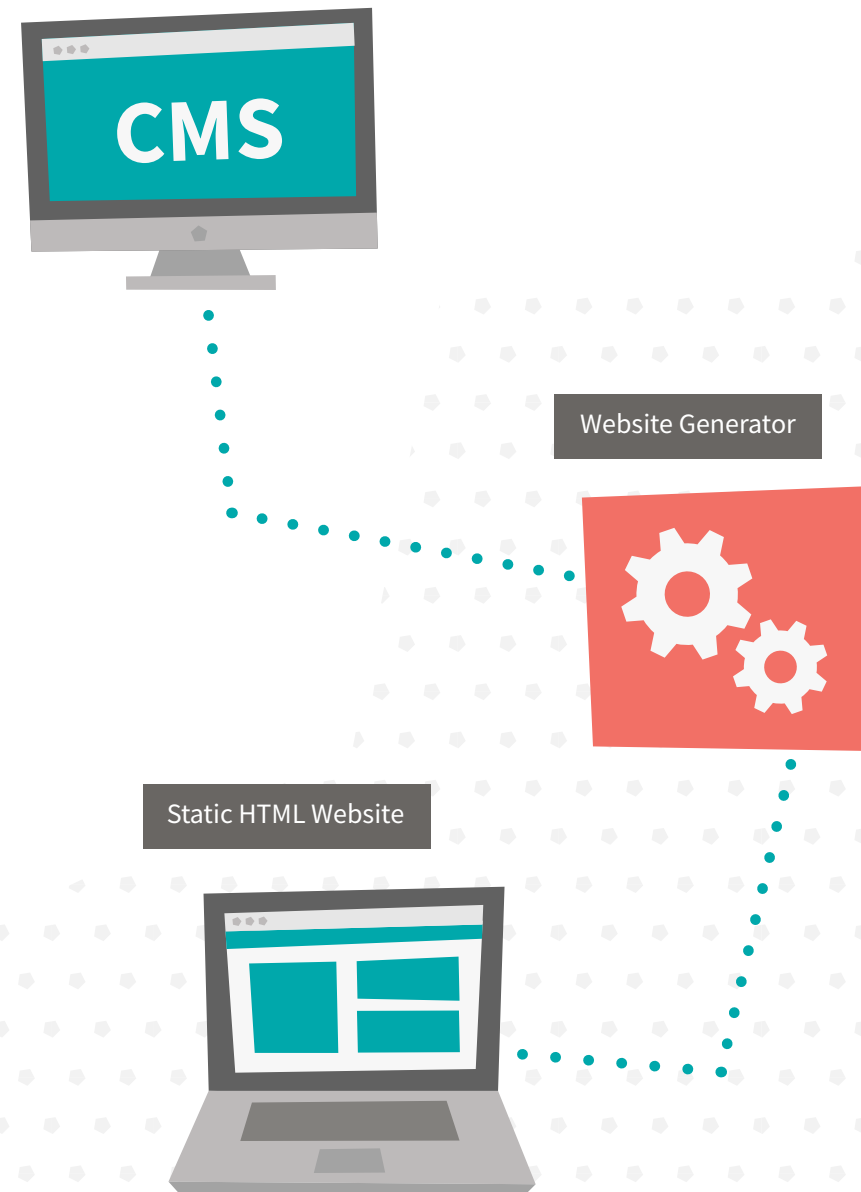
### Static site generators

Static site generators aren't actually a CMS. These are usually frameworks or scripts that take content provided in a specific format and use it to generate static HTML files that can be hosted on any server.

This option is popular for smaller websites that change infrequently and don't need to provide any kind of personalization or other dynamic elements.

In order to provide a user-friendly editing interface, it's common to use a headless CMS as the source of content for a static site generator. You can also use headless CMS webhooks that notify your site-generating code when content is updated so it can generate and upload a new version of the website to the server.

Static site generators are focused on HTML websites, so they do not support other channels.



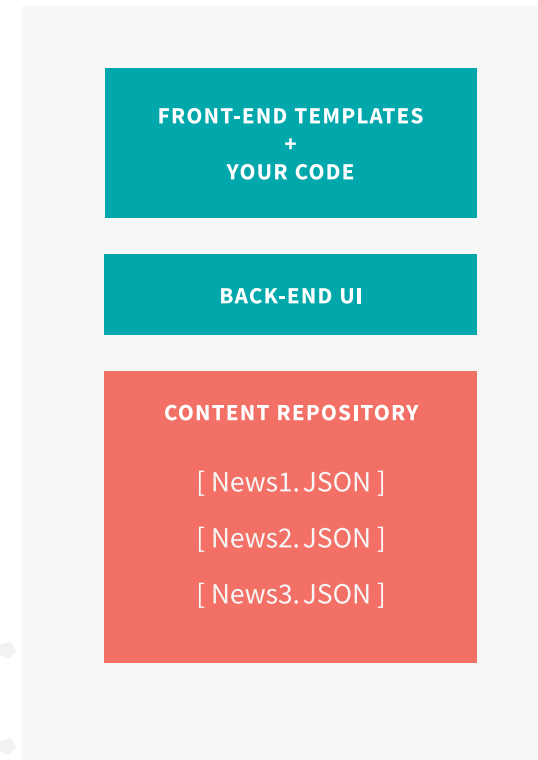
## Flat-file CMS

Flat-file CMS systems are usually coupled, web-oriented systems that use a file system to store the content in a structured format, such as XML or JSON files.

This means they don't use SQL databases, which makes their server footprint smaller and makes them easier to deploy or migrate to other servers.

This is an interesting option for companies that want a custom, dynamic, CMS-powered website with very low hosting costs.

This model, however, doesn't scale for large websites.



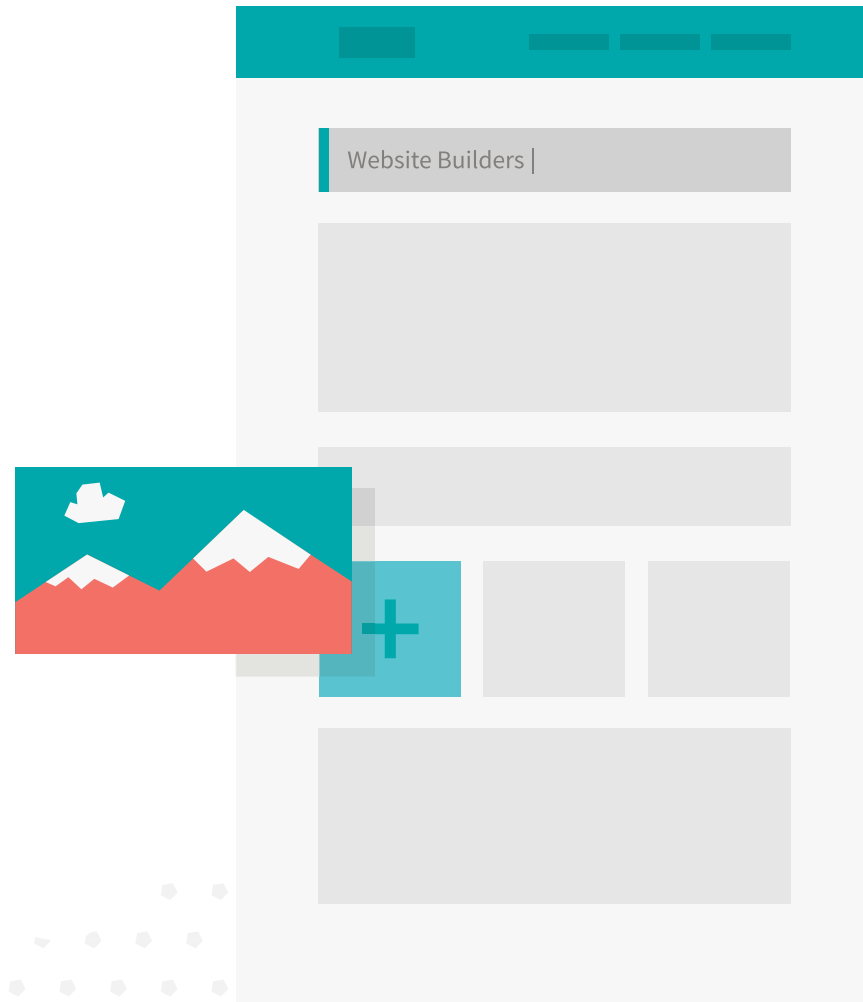
## Website builders

Website builders include SaaS solutions like Wix, Squarespace, Webnode, and others. These products are great for simple template-based websites. They're very easy to set up and even non-technical users can create a great-looking website in a couple of hours.

They are, however, not suitable for organizations that want a fully custom solution with specific functionality or integrations.

They also aren't a true CMS as they do not allow for management of structured content—they only support page editing and a limited number of predefined content types, such as blog posts or news, and don't allow you to define your own content types.

While these builders allow you to create responsive websites, they offer no (or very limited) capabilities for any other channels.





A green dome tent is pitched in a rocky, grassy field. The tent is the central focus, with a stone fire pit in front of it. Large, grey rocks are scattered around the tent, and a large rock formation is visible on the right side of the image. The ground is covered in sparse grass and small rocks.

Chapter III

# Choosing your hosting model

# 1 On premise, managed cloud, or SaaS?

You can choose from a number of CMS hosting models. Very few customers run their CMS as a true on-premise solution on their own servers. These days, most customers leverage a cloud hosting model. In most cases, however, they just manage a CMS installation in the Cloud themselves or have a vendor manage it for them. This approach doesn't allow customers to fully leverage all the benefits of the Cloud. The new headless model finally allows vendors to provide a CMS as Software as a Service (SaaS). Let's have a look at each option:

## Self-hosted on premise

If you run a traditional CMS on premise you need to manage the:

- configuring of a web server
- installation of the CMS
- application of hotfixes and upgrades
- ensurance of backup, security, and performance

... and only after that can you work on your website.

That's why more and more customers are looking for a better model.



## Self-hosted in the Cloud

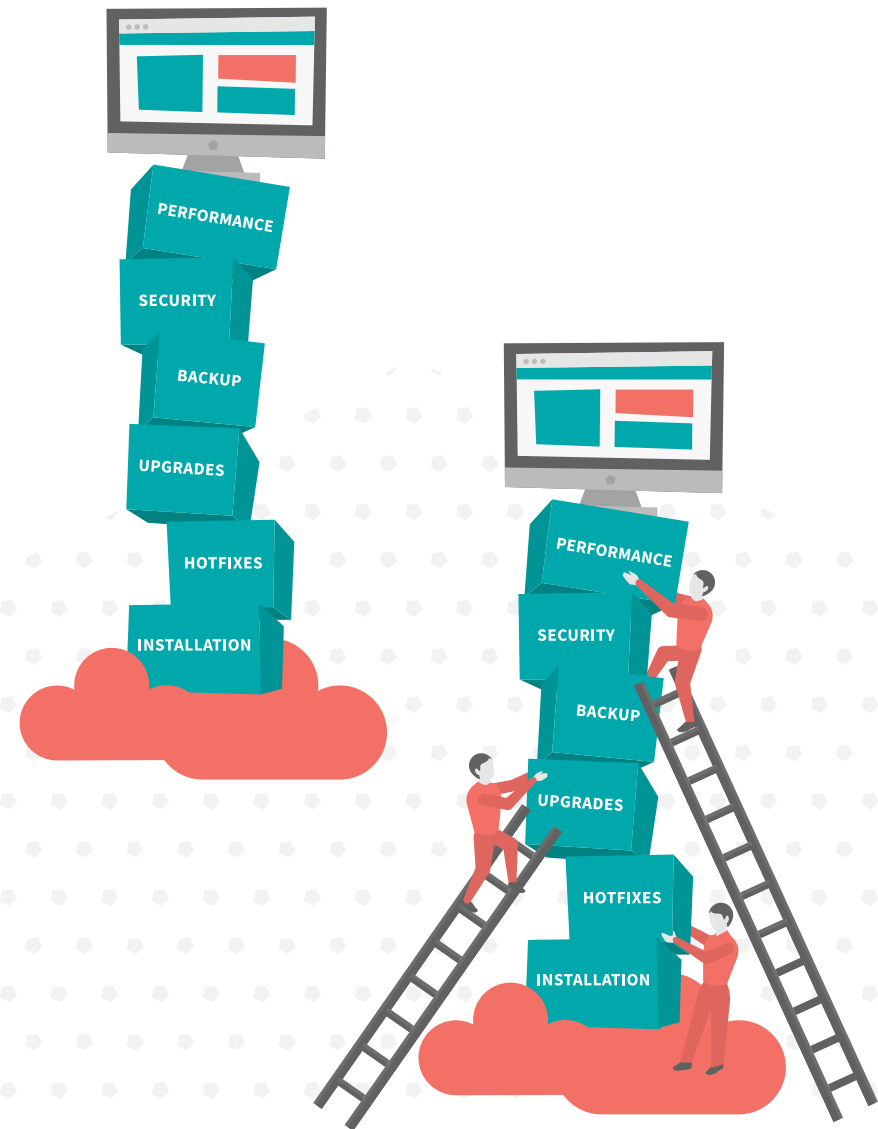
Even if you install your CMS in the Cloud, such as Amazon Web Services or Microsoft Azure, you still need to babysit it. The only advantage is that you no longer need to worry about the infrastructure or the underlying platform.

## Managed cloud hosting

Some traditional CMS vendors offer managed cloud hosting for their CMS.

While your first reaction might be “Wow, they take care of everything!”, it lacks the flexibility one would expect: with every change of the code, you typically need to talk to the vendor and ask them to deploy it to your production environment.

Moreover, while this model moves the inefficiencies from you to the vendor, at the end of the day, you pay for them.

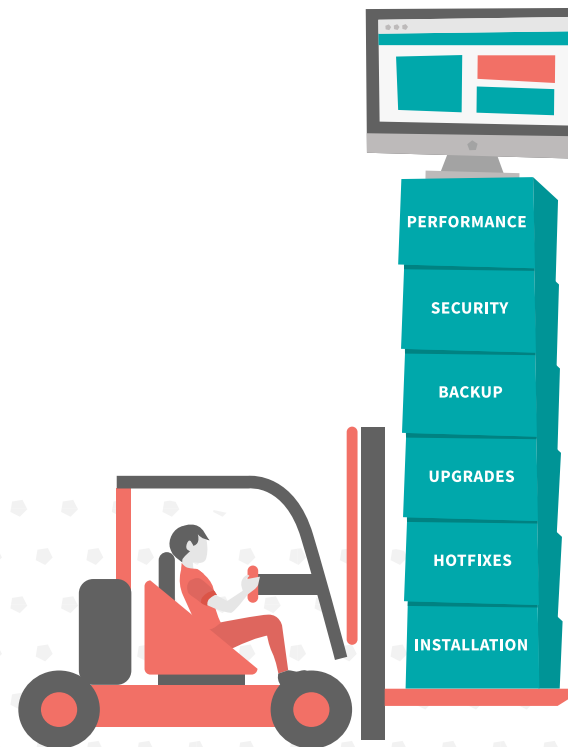


## Platform as a Service (PaaS)

This model is based on a unified CMS configuration that allows vendors to automate what would otherwise be manually managed cloud hosting.

The problem is, you may not be able to use certain plugins or create certain customizations. You also only have limited control over the hosting environment and you still have to test your website after each upgrade and hotfix to make sure it's not broken.

The PaaS model still lacks the elegance and efficiency you would expect from a true cloud service.



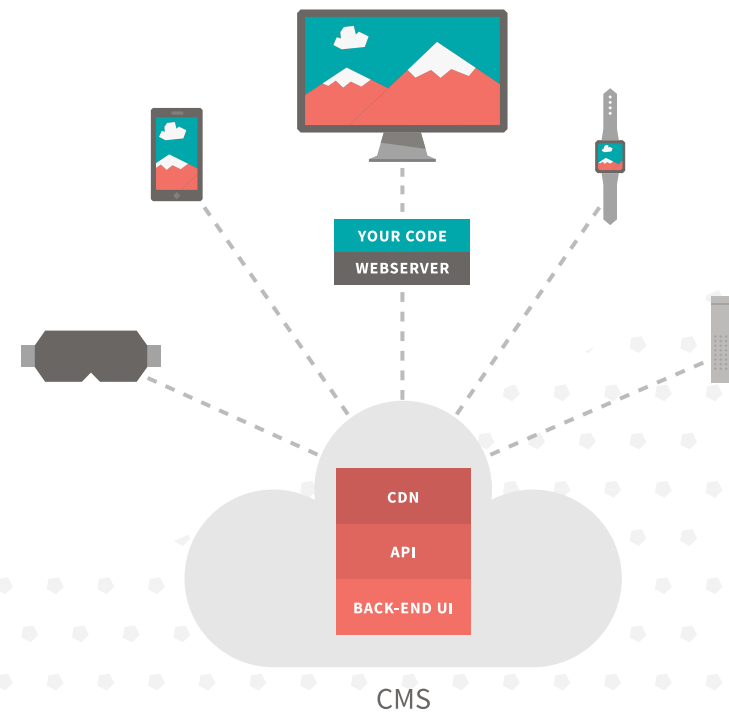
## Software as a Service (SaaS)

The headless approach separates the concerns of the vendor (running a CMS) and the client (creating a website or other application that consumes the content).

With SaaS, the vendor provides all customers with the same up-to-date CMS, high availability, security, and performance.

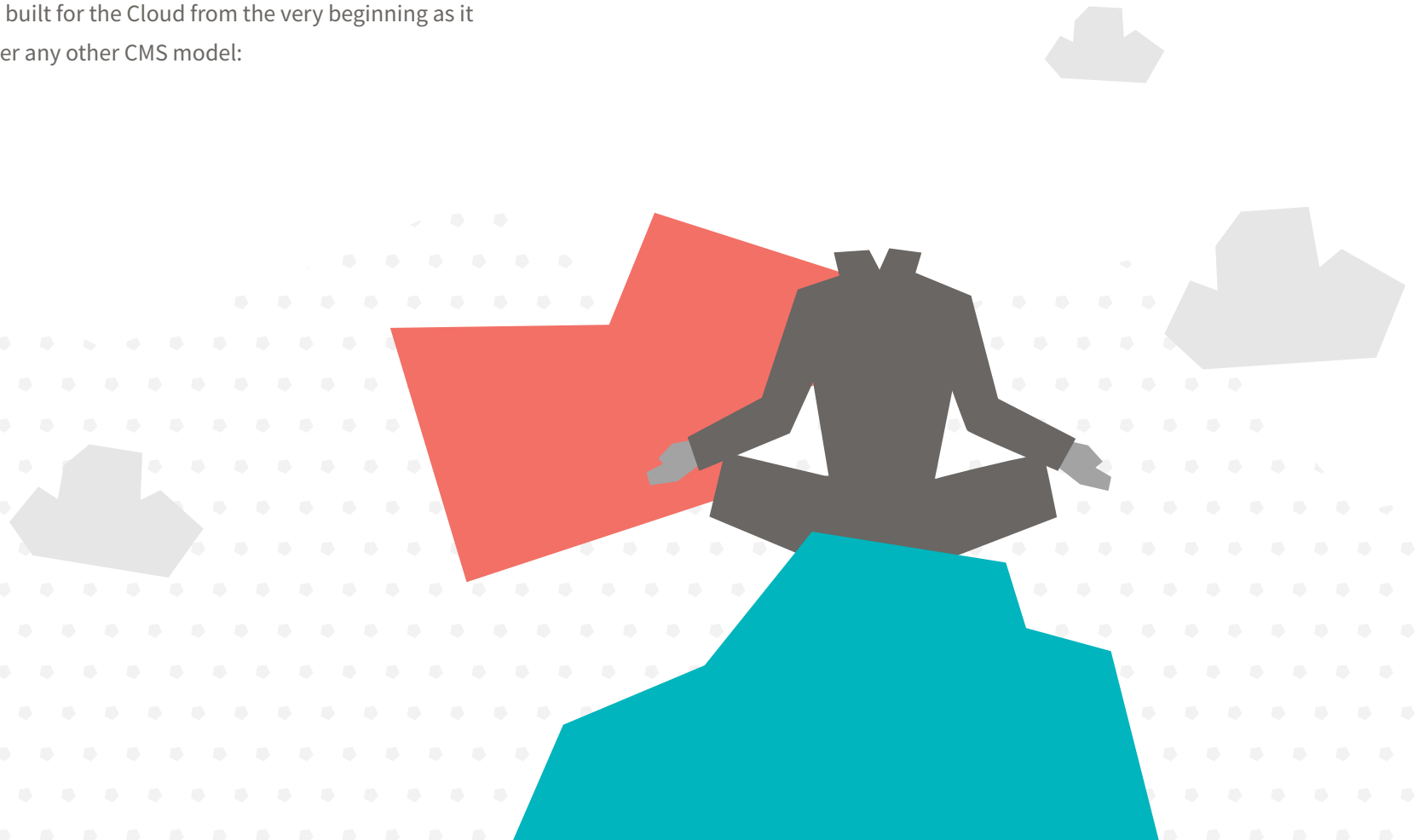
This means you only look after your own solution.

This model allows both vendors and customers to get all the benefits of the Cloud and achieve higher agility at lower operating costs.



## 2 Cloud-first headless CMS: your peace of mind

As previously mentioned, the headless model enables vendors to provide a CMS as a true Software as a Service (SaaS) solution. This wasn't possible before and it means a major revolution to the whole CMS industry. While you can use an on-premise headless CMS, it would be like buying a hybrid car and using only its combustion engine. You should use a cloud-first headless CMS that was built for the Cloud from the very beginning as it offers huge benefits over any other CMS model:



## No worries about infrastructure and CMS

The SaaS model means that you do not need to worry about any underlying infrastructure or middleware. You can forget about

- Web server maintenance
- Database server maintenance
- Backups

... and other pains connected with running a CMS. All you do is manage your content and retrieve it through an API!

## No painful upgrades

With a cloud-first headless CMS, you are always using the latest version of the CMS, without going through painful and costly upgrades. Is there a new version of a browser? A security threat new on the scene? Or new legislation to comply with? No worries, the vendor takes care of that and you benefit from any new functionality immediately.

## Better security

Since the cloud-first headless CMS is managed by the vendor who deeply understands its architecture and underlying infrastructure, it's much easier for the vendor to ensure proper security. Moreover, when the vendor fixes a security flaw, it's fixed immediately for all customers, giving hackers a much smaller window of opportunity. It's true that you still need to make sure your own code is secure, but it's much easier than securing a large monolithic CMS application as well as third-party plugins.

## Infinite scalability and high availability

The cloud-first headless CMS model allows vendors to fully leverage the power of the Cloud and build a highly scalable architecture with high availability—and you can very easily build your solutions on top of that architecture without any effort!

Moreover, such a CMS typically uses a global Content Delivery Network (CDN) to deliver the content to any application anywhere in the world in no time. The CDNs are built for extreme load which makes the content delivery unbreakable, even if you run a Super Bowl commercial or your campaign goes viral.





Chapter IV

# When to use a headless CMS

# 1 Should you use a headless or traditional CMS?

## Choose a traditional CMS if

- You want to host the CMS yourself on your servers or in the Cloud
- You only want to build a website and have no plans to support other channels
- You want to use the CMS as a development platform
- Your marketers can't change their "website-first" mindset

## Choose a cloud-first headless CMS if

- You want to create content for multiple channels, not just a website
- You want to leverage all benefits of a true SaaS solution provided as a cloud service
- You want to build your applications using a microservices architecture
- Your marketers are able to adopt an omnichannel content strategy



## 2 When to use a headless CMS: typical scenarios

### Content hub—one place for all your content

A cloud-first headless CMS makes for a perfect centralized repository for all your structured content. It can be used as a master content management system that serves as a single source of truth across the whole organization.

With a single repository, you can avoid creating content in departmental silos which causes inefficiency, inconsistency, and duplication of work.

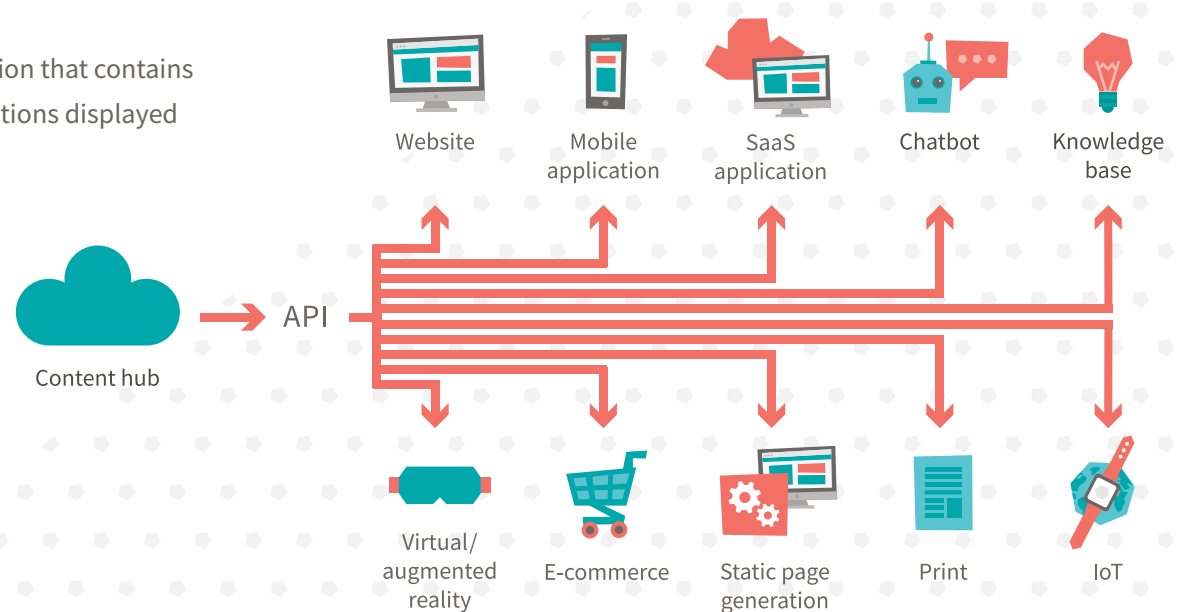
You can import content from various sources into a single cloud-based repository where you can collaborate on it with your colleagues. Then you can export it to other systems to ensure consistency or make it directly available to your website or custom applications using an API.



## Omnichannel content delivery

The API provided by a headless CMS allows you to deliver content to any channel, any device, and any platform. While many of these channels are obvious, let's have a look at some specific scenarios.

- Static page generation—you can use the headless CMS for content storage and then use script to generate static HTML files for your website.
- Conversational interfaces, such as digital assistants or chatbots, need a specific content structure that may not be well supported by traditional page-oriented systems.
- Microcontent for SaaS—think of a banking application that contains hints, short product descriptions, or various promotions displayed in the context of the application.



## Produce content in the Cloud, export it anywhere

In some cases, you may not be able to use the content API to serve the content directly as you may already have existing customer-facing systems in place that you can't replace immediately. You can still use a headless CMS to collaborate on the content and then export it to other systems using a custom integration. Here are some examples:

- **Preparing content for your website before a traditional CMS is fully set up**

You can use the cloud-based content hub to collect all your content and collaborate with others, while developers build the website, helping reduce time to market.

- **Handling the transition to a headless CMS**

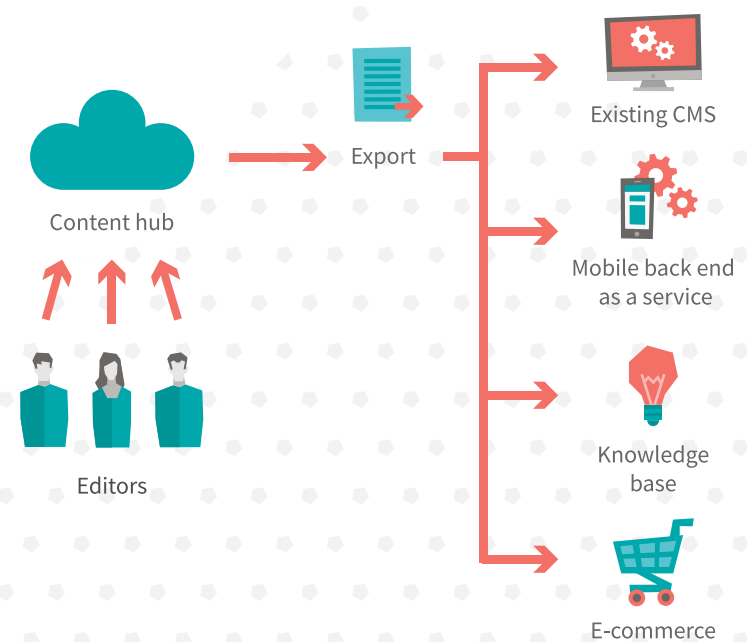
If you're in the transition towards a new headless CMS, you may need to centralize your content first and rebuild the digital experiences one by one. In the meantime, you can manage the content in the headless CMS and export it to existing systems.

- **Managing content for your mobile application that uses a mobile back end as a service (mBaaS)**

You can write script that takes the published content from the headless CMS and exports it to the mBaaS, which then distributes the content to your app. By doing this, you eliminate the need to modify the app while still reaping the benefits of a user-friendly CMS that supports the full content lifecycle.

- **Creating content for a knowledge base**

Instead of creating content for your website and knowledge base separately, you can create it once and then export it to both your CMS and knowledge base.



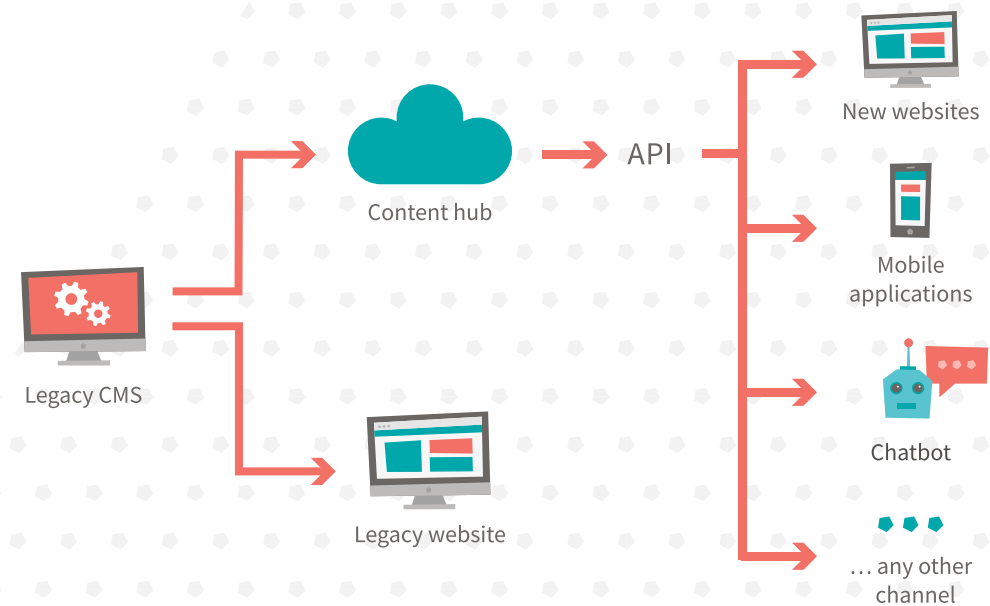
## Making a legacy CMS multichannel

If you invested big money into building highly customized solutions on top of a legacy CMS and cannot easily replace it, you can use a headless CMS as a proxy: simply export the content into the headless CMS and then benefit from its scalable API.

This way, you can leverage advanced content management available in traditional CMS systems, such as complex access control, translation management, or workflow, without having to retrain your end users on a new CMS.

At the same time, you can leverage the API to deliver the content to any channel and boost the performance of your legacy CMS, without spending more on servers and perpetual licenses.

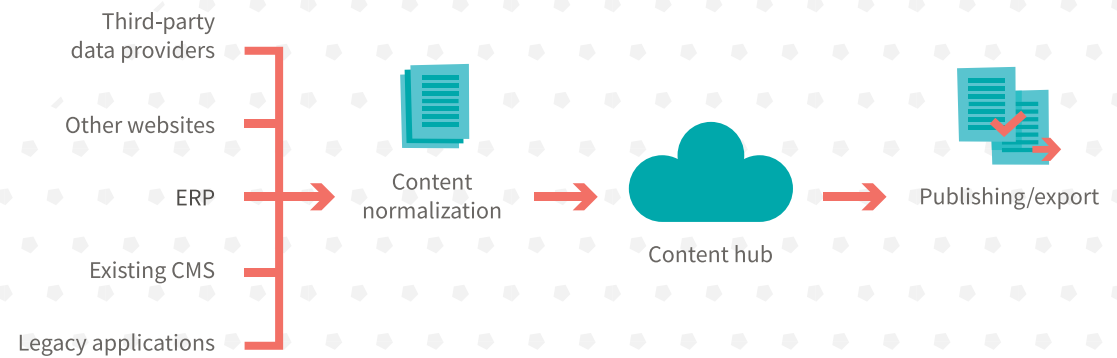
Still, you should see this as a temporary solution that allows you to test the headless concept before a full transition.



## Content aggregation

You may already manage your content in multiple applications—including ERP, PIM, legacy CMS, etc. You may also retrieve some content from external sources or scrape content from various websites, RSS feeds, or APIs.

In these cases, you may want to aggregate the content, normalize it into the required format, and use the headless CMS to make it available across any channel.



## Personalized omnichannel experiences

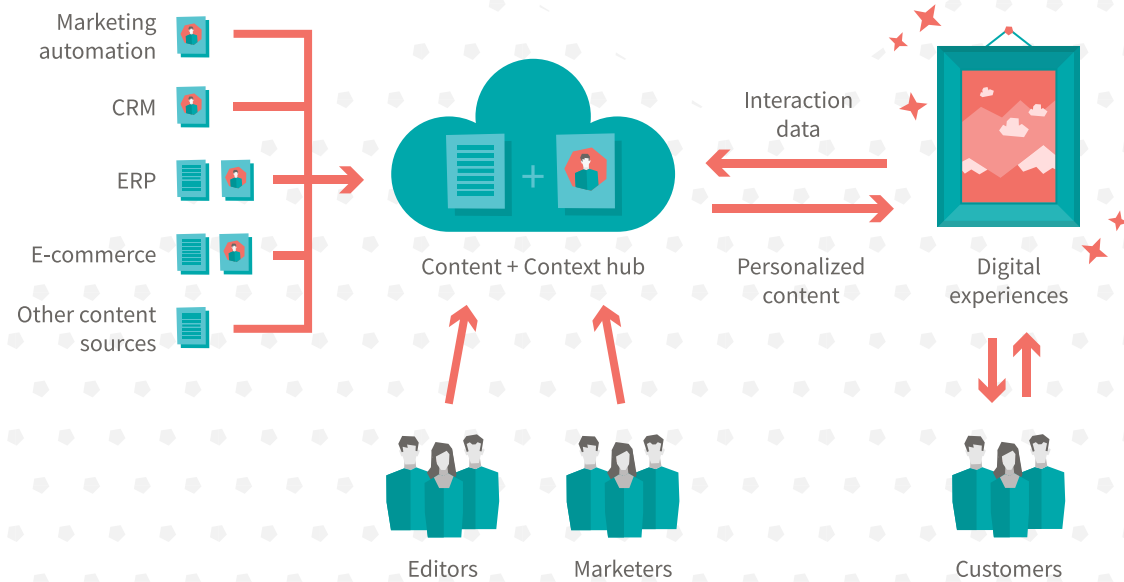
Originally, headless CMS was created by developers as a response to technical challenges they faced with traditional CMS systems in the emerging multichannel world.

However, as the concept becomes more widely adopted in business, marketers will try to get their own piece of the headless world.

They expect not only a friendly user interface and more empowerment, but they also need to measure the performance of their content and optimize it.

Marketers will benefit from a cloud-first, API-first digital experience platform that goes beyond a pure headless CMS and allows them to collect and evaluate data about customers and their interactions with content.

Effectively, such a digital experience platform will provide Content as a Service and Context as a Service in a single solution. This means each piece of content will come with detailed statistics on how it performs in each channel, providing invaluable feedback to content authors.







Chapter V

# Increase your digital agility with a headless CMS

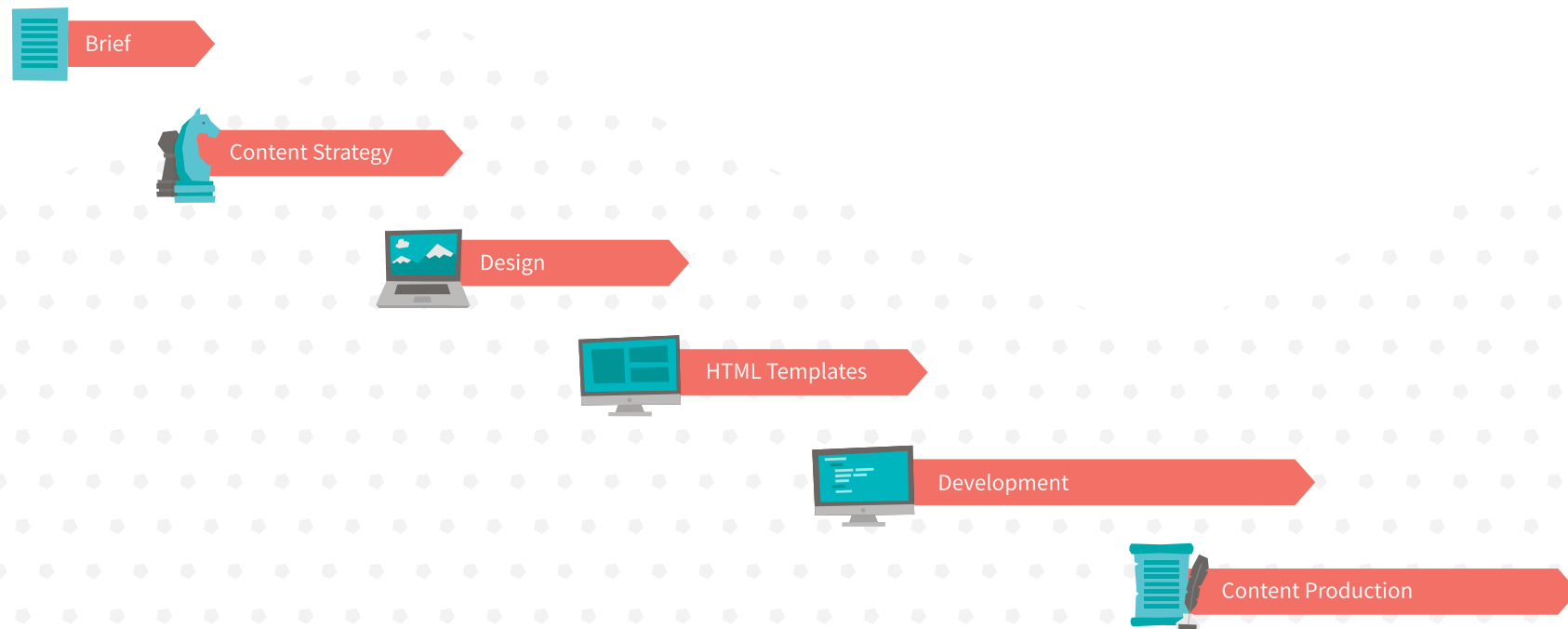
# 1 Replace waterfall with agile planning

## Traditional CMS

With the traditional CMS, the project planning was driven by the technology. Since your content was tightly connected to your website design, you needed to figure out what the website would look like before you could start working on the content.

Moreover, you couldn't start entering the content into the CMS before developers had implemented the page templates.

This led to project delays and inefficiencies in content production: you either waited until the website was ready or you had to create your content in a temporary format, such as excel spreadsheets, and then copy & paste the content into the new CMS.



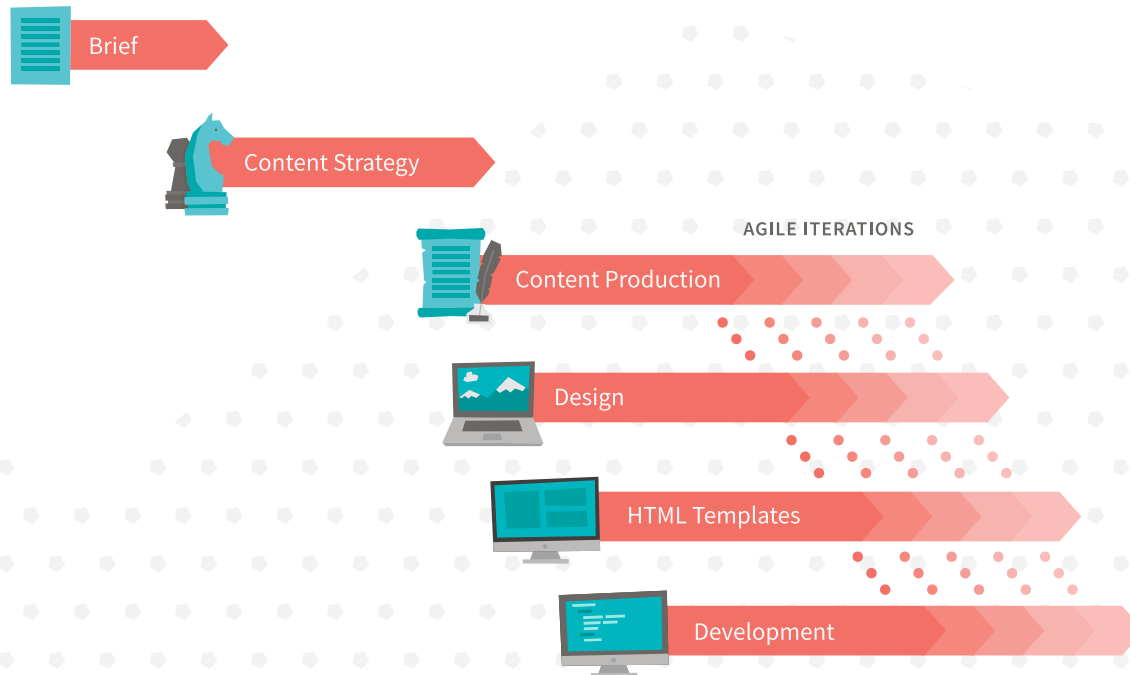
## Headless CMS

The headless CMS naturally leads you to a content-first approach. Once you have a proper content strategy in place, you can start creating content in a structured format that can be used for any channel.

This means you don't have to wait for the CMS to be set up—you can produce content while developers create the website and other applications.

Moreover, the headless approach enables front-end and back-end developers to work more independently, allowing for better people allocation on the project.

Overall, the headless approach shortens your time to market and makes your process much more agile.




## 2 Replace “website” with “omnichannel”

The new multichannel world requires a major shift in how you think about content. In the web-only world, it’s common that you organize your content based on pages and their layouts.

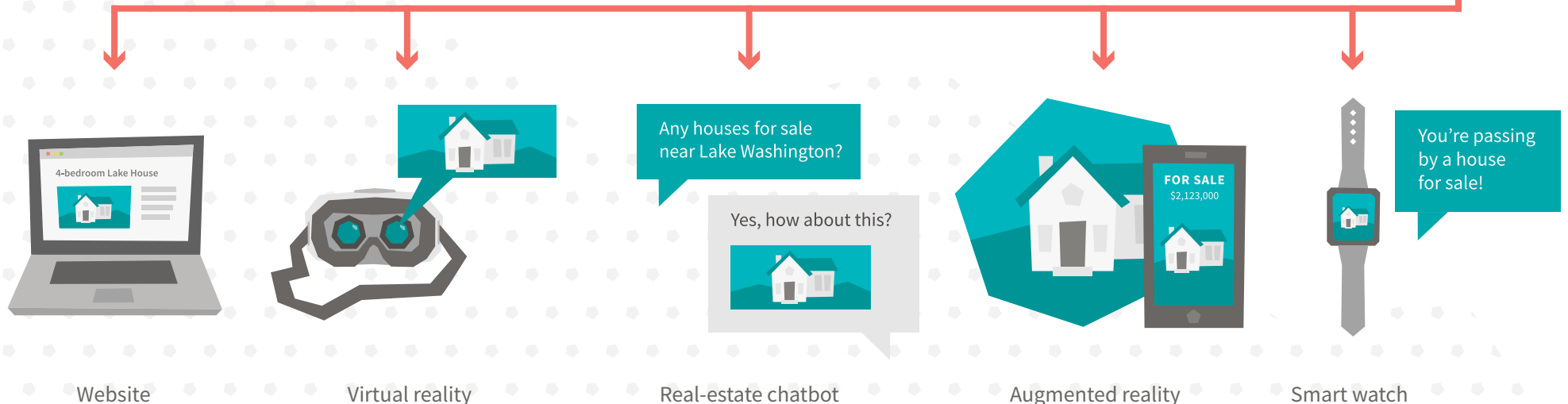
In the new omnichannel approach, content needs to be ready for any presentation, meaning you need to structure content in smaller chunks that provide semantics to each piece of content and allow for its reuse.

**Property**

<b>Name:</b> 4-bedroom Lake House	<b>Price:</b> \$2,123,000
<b>Bedrooms:</b> 4	<b>Photo:</b> 
<b>Location:</b> 47.609, -122.284	
<b>Suburb:</b> Lake Washington <input type="checkbox"/>	

**Suburb**

<b>Name:</b> Lake Washington
<b>Description:</b> This suburb is...



### 3 Replace your rigid IT process with mode 2 IT and DevOps

A traditional safety-first approach to IT has become a struggle for organizations that go through a digital transformation and need to become more agile.

Bringing up a new website or digital initiative in such an environment often takes many months, whereas marketers expect weeks.

As a result, marketers increasingly avoid their IT and choose sub-optimal technical solutions that consequently become a nightmare to manage.

The solution is what Gartner describes as bimodal IT: Mode 1 is focused on what is known making incremental improvements. Mode 2 is focused on exploring and experimenting.

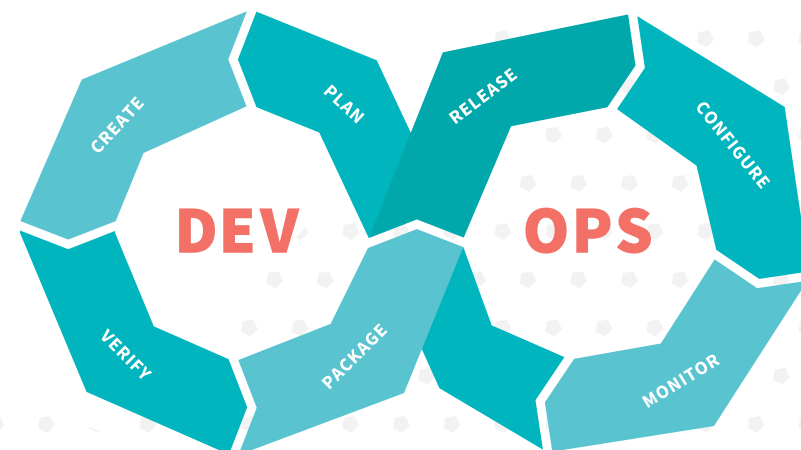
A prerequisite to Mode 2 IT is embracing the DevOps approach. DevOps is a crossfunctional mode of working between product management, software development, and operations that is focused on rapid development with frequent releases.

Such an approach allows you to quickly test new ideas and change your direction as often as you need to.

A cloud-first headless CMS represents a perfect fit for organizations who seek more agility.

As it's provided as a cloud service, you can avoid troublesome installation, maintenance, and upgrades and can rely on the vendor to take care of CMS security and performance for you. At the same time, you can easily integrate it into your environment through its API. The API-first architecture that clearly separates CMS from your code makes your continuous deployment process much easier compared to a traditional installed CMS.

A cloud-first headless CMS represents a perfect fit for organizations seeking more agility.



## 4 Replace a monolith with microservices

In the past, the only digital experience most companies provided was their website, typically built on top of a CMS. Any custom code was tightly connected to a CMS and glued to its monolithic architecture.

Today, brands need to provide a seamless experience across multiple channels, such as mobile applications, point-of-sale systems, social media, Internet of Things devices, virtual reality, digital assistants, and chatbots.

The growing number of customer touchpoints requires companies to provide a much broader set of applications.

This has lead developers to rethink their architecture and adopt the microservices approach where, instead of building a website on top of a CMS, they create their applications as a combination of multiple services.

They can create their own microservices, or they can use external services through their API as building blocks for their application. Instead of writing code from scratch or including third-party libraries in their code, they can simply call cloud-based services that do the job.

The question is no longer “Which plugins do we use for this website?”, but “Which APIs do we use for our applications?”

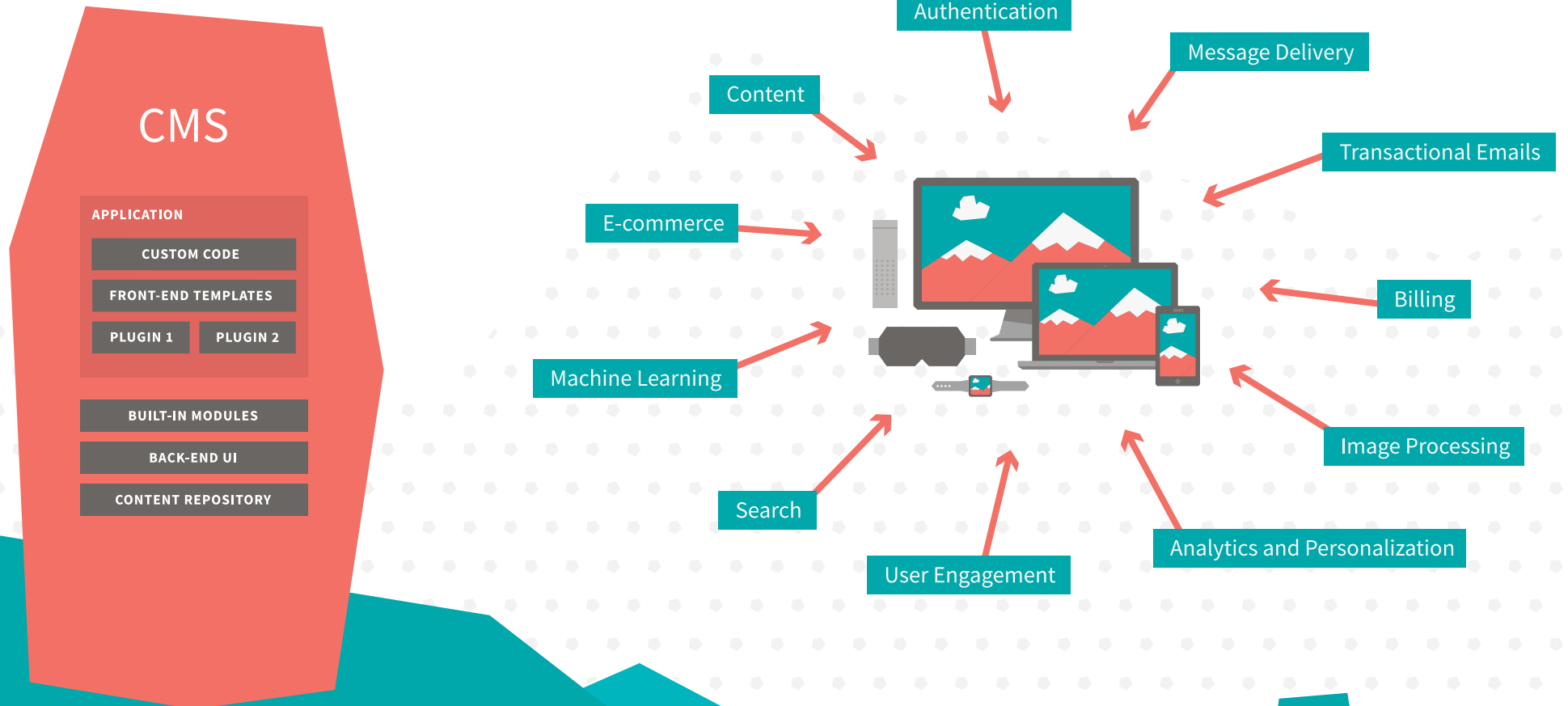
The use of microservices allows you to quickly assemble applications using multiple APIs from different vendors. What used to be a tedious integration task is now much easier.

These APIs are provided as a service which means you don’t have to worry about running any software and you can benefit from ongoing innovation without complex upgrades.

The fact that you no longer build your application on top of a proprietary CMS platform, but rather as a compilation of easily replaceable APIs, means that you avoid vendor lock-in and extend the life of your code, thus protecting your investment in development.

Monolith application built on top of CMS

Digital experiences built using APIs





A photograph of a waterfall cascading over dark, moss-covered rocks in a dense forest. The water is white and frothy as it falls. The surrounding rocks and foliage are dark green and black, creating a high-contrast scene.

# Summary



**Cloud-first headless CMS is the future.  
It combines the agility of a cloud-based  
solution with the unlimited flexibility of  
API-first architecture.**

**Now it's your turn: choose a headless  
CMS and deploy it for your first project!**

Go to [kenticocloud.com](https://kenticocloud.com) and start a free trial!





## About the Author

Petr Palas has one passion—Content Management Systems. In the past 15 years, he has written five CMSs. He's the founder and CEO of the leading CMS software company—Kentico Software.

## About Kentico Software

Since 2004, Kentico Software has helped over 10,000 brands in 100 countries tell their stories and engage with customers in the digital world. Its traditional CMS product—Kentico CMS—is one of the most popular solutions for mid-market and enterprise companies on the Microsoft .NET platform. In 2016, Kentico introduced a new product: Kentico Cloud—the cloud-first headless CMS. Kentico Cloud was built from scratch with a new vision: to create a CMS that would go beyond websites and allow people to create content and deliver personalized experiences across any channel. You can try Kentico Cloud at [www.kenticocloud.com](http://www.kenticocloud.com).

# **The Ultimate Guide to Headless CMS**

Written by Petr Palas

Published by Kentico Software s.r.o. in 2017  
Nové sady 996/25, 602 00 Brno, Czech Republic

First edition, 2017





[www.kenticocloud.com](http://www.kenticocloud.com)