

The ultimate guide to headless CMS





The Ultimate Guide to Headless CMS

How the next generation of agile CMS options can work for you

Let's take a minute to talk about terminology. Headless and decoupled describe the architecture behind modern CMSes, and differentiate these systems from traditional CMSes and their limitations. Unfortunately, these technical labels do little to help decision-makers and non-technical users understand what modern CMSes do differently. This is often exacerbated by communication challenges between developers and other team members.

When it comes to selecting a CMS, it's important that product managers need to consider the future needs of their products and how the CMS will impact them. Traditional CMSes were made for websites, and despite efforts to modernize them and make them more API-friendly, they are falling behind competitors who have reimagined content management from an API-first approach. Businesses that cling to the familiar CMS model will find themselves falling behind too.

Forrester uses a more descriptive term for the future of CMS in "It's The End Of Web CMS As We Know It (And I Feel Fine)," Mark Grannan, Forrester Research, Inc, 15 November 2018.

"We need a new container to meet tomorrow's challenges. Building from the architecture of headless CMS, Forrester believes that that new container is already taking shape: Agile CMS. Forrester defines Agile CMS as: A solution for collaboratively curating, creating, and delivering content across channels and campaigns via iterative development and deployment processes."

Key terms to know

1. HEADLESS CMS

Any type of content management system where the content repository "body" is separated or decoupled from the presentation layer "head."

2. HEADLESS VS DECOUPLED CMS

There is some debate about the differences between headless and decoupled CMSes, since a lot of people assume that they are the same thing. The key difference is that a headless CMS gives developers complete control over how content is displayed, whereas a decoupled CMS prepares content for presentation and pushes it to an application. [Learn more here >](#)

3. APIs

How programs interact with each other. Headless CMSes use APIs to deliver content to digital platforms.



Instead of tightly marrying content to a web page or app, headless and decoupled CMSes provide more flexibility for where and how you deliver your content. They empower you to use content across your digital portfolio more efficiently, so even a single piece of content works harder and reaches farther. Your team enjoys more efficient content workflows. And you can focus on what's next for your product.

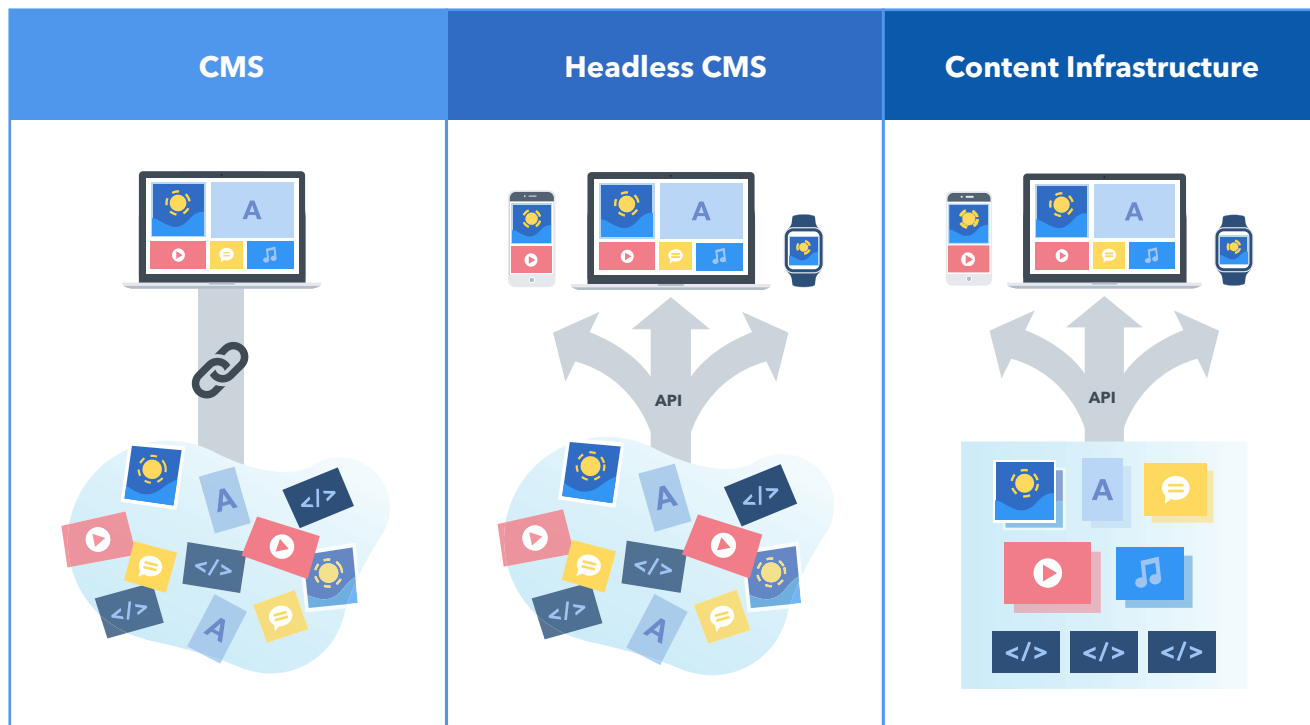
While it is good to understand what the terms mean, what you really need to know is how and why this next generation of content management solutions are the smart choice for your business.

In this white paper we'll explain in non-technical terms what makes a CMS "headless" and, more importantly, how being headless makes a CMS more agile and better able to meet your growing digital needs. We'll provide a checklist of key criteria for determining whether or not headless is right for you and then take a deeper dive into how to evaluate the sea of headless CMS vendors.

WHAT IS HEADLESS?

Web pages used to be built by developers or other savvy individuals who learned to code content into a page. The first CMSes hid this code behind a friendly user interface, making it easy for non-developers to update and add to web pages. Today's traditional CMSes have a lot more features, but behind the scenes the content and code are still combined in web-centric frameworks, making it nearly impossible to modify or adapt them to new digital applications. That means for every new digital product, you need another CMS.

Headless CMSes take a different approach by storing content separately from the code used to build the presentation layer. Thus, one system can manage all of your content, no matter its destination. Each piece of content can be pulled via APIs into any digital endpoint, without directly changing the original content. The beauty of this is that it streamlines the content management process, while simultaneously multiplying the reach of each piece of content.





WHY GO HEADLESS?

Traditional CMSes were built for websites and remain very channel-centric, resulting in situations where a business uses different CMSes for each channel. But modern businesses aren't organized around delivery channels. Channel-centric technology is being replaced by more agile technology that can be leveraged across products and teams to deliver omnichannel experiences.

"Moving legacy technology barriers further out of the way will require de-emphasizing channel-centric technologies. Even in martech – an area of profound technology fragmentation – channel-centric technologies are on the decline as businesses evolve their processes to be more Agile." – "It's The End Of Web CMS As We Know It (And I Feel Fine)," Mark Grannan, Forrester Research, Inc, 15 November 2018.

With the plethora of digital channels that are available, outdated legacy CMSes are becoming huge agility roadblocks.

The two biggest things driving change are:

1. Frustration with too many CMSes. With the explosion of digital platforms, enterprises are often crippled by a proliferation of CMS instances – dozens, or even hundreds. As a result, they have to copy and paste content from a website CMS to an app CMS and then to a digital display CMS. For these companies, consolidating content into a headless content hub is a game changer.

2. A desire to build and ship faster.

For companies undertaking a digital transformation, the question of CMS vs. headless CMS isn't just an IT choice; it's a strategic decision for digital leaders and their teams on how to upgrade tech stacks to be more agile and competitive.

WHY NOT?

For small businesses that don't have digital teams, headless CMS is often times not the best choice. It requires technical proficiency or capacity to handle the presentation layer. For enterprises currently wedded to their legacy suites, headless CMS can present a challenge in change management, in the same way migration from on-premises servers to the cloud required changes in people, processes and technology.

Legacy-bound enterprises have a great opportunity to pilot a headless CMS in parallel. Digital teams to test a greenfield project as a proof of concept, without being forced to immediately commit to ripping and replacing their legacy CMS. These tests demonstrate the business value and result in more teams and departments adopting a headless approach to content management.

"Moving legacy technology barriers further out of the way will require de-emphasizing channel-centric technologies...as businesses evolve their processes to be more Agile."

– "It's The End Of Web CMS As We Know It (And I Feel Fine)," Mark Grannan, Forrester Research, Inc, 15 November 2018.



Is headless right for you?

1

You're already using multiple CMSes

Cutting and pasting content between CMSes is inefficient and leads to inconsistencies. A headless CMS could help you gain speed and improve quality.

2

You are heavily dependent on developers to make changes and publish content

Waiting on developers to edit or publish content creates unnecessary delays, and distracts developers from their core functions. A headless CMS creates a healthy amount of empowerment and separation.

3

Product launches always seem to hit content-related delays and content operations take longer and require more resources

Piling complex demands onto a traditional CMS or managing multiple CMSes can result in time-consuming workflows as you take on manual work to compensate for the CMSes shortcomings. A headless CMS can help you streamline content operations and get to market faster.

4

It's getting harder to find and keep developers to work on your CMS.

A CMS that requires specific programming languages or is tedious for developers to maintain limits the talent pool. A headless CMS lets developers work in the languages they prefer and eliminates the grunt work associated with traditional CMSes.

5

The CMS has become more of a challenge than an enabler.

Traditional CMSes can feel like a rigid box with walls popping up to block your best ideas. A headless CMS removes these barriers by making content more flexible and accessible wherever you want to use it.

Adding, upgrading or changing CMSes might ease these problems in the short term, but that perpetuates the cycle of choosing a CMS >> implementing the CMS >> outgrowing the CMS >> replacing it (painfully) with a new CMS. The old method was to repeat this process ad infinitum, wasting time and money.

Product managers with an eye to the future should consider modern CMS options with the potential to grow with your business. A headless CMS can provide all the benefits you'd expect from a new CMS, plus it will continue to add value by adapting to your future needs. Instead of stopping to find a CMS just for your next product or an emerging channel, you can speed past the competition and go straight to market.



Choosing the right vendor: Not all headless CMSes are created equal

Now that you've decided to include headless in your CMS search, let's narrow down the options. Headless CMS is a term for a rapidly evolving category, so not all vendors offer the same type of product. Some offer enterprise-grade features and functionality, some are open-source, and some have limited features that can facilitate experimentation but can't scale effectively.

When evaluating headless CMSes, consider:

1. Architecture
2. Content
3. Content operations
4. APIs and extensibility

For each of these, we'll help you understand what sets the best headless CMSes apart. You can weed out vendors who are just scratching the surface of what headless can do, and focus on those who are positioning themselves to be the agile CMS of the future.

ARCHITECTURE

When comparing headless CMS vendors, it's helpful to remember that headless means that the content repository and the front end where content is displayed are decoupled. All of the configurations in the illustration below can be considered headless, but the freedom they give developers, product managers and editors varies.

CONTENT

Headless usually focuses on how content flows out of the CMS to different delivery platforms, but product managers also want to look at how content is managed within the CMS. Separating content delivery from content management only solves part of the challenge. The question remains: How do you organize and manage content for use across different platforms?

A traditional CMS organizes content as pages, using rigid templates to tie headlines, body

copy and images together into specific layouts. This page-centric approach limits your ability to reuse those content elements in different layouts for different channels. This problem can persist with headless CMSes that still use a page-centric approach.

Content infrastructure takes a different approach. Instead of organizing content around pages, it starts with a content model – a framework for organizing types of content and defining how each type relates to another.

A content model is custom-built for each organization, so content creators aren't stuck with the preprogrammed models offered by a traditional CMS. A content model breaks down your content into individual elements, such as a blog post headline or the copy on a call-to-action button. You can then define how each element relates to others. In doing so, you create a flexible model that can be molded to fit any digital container.

CONTENT OPERATIONS

Your chosen CMS should support a fast flow of content from creation through delivery.

First, take a deeper look into the basic features you expect in a CMS.

- Is the **editorial interface** easy to use? Can it be customized to your team's preferences? What time-saving features does it include? Contentful offers a number of UI extensions that allow you to customize the editorial interface based on your teams' needs and workflows. You can change the appearance of fields, add dialogue boxes with helpful tips for editors, create drop downs to give editors more options and integrate with other tools so that editors can access forms, videos and other assets without jumping between screens.



- How granular are the **permissions and roles**? Content governance and consistency are important for maintaining a strong brand image. Does the CMS offer enough control to give everyone the level of access they need while protecting content from unauthorized changes?
- Is it easy to **preview content**? Editors like to be able to see how content will look as they're putting a page together. Having to wait for a developer to publish content for previews and approvals is time-consuming. Look for CMSes that let you preview changes immediately and make it easy to share that preview with anyone who needs to approve content and make edits, all before anything goes live.
- Can editors **publish content** on their own? Putting publishing power into the hands of editors and other end users speeds up your time to market. The CMS should support publishing from the editorial interface, allowing for changes to live content that take minutes, not days.

Next, look beyond the basic features for ways in which the CMS supports agile practices.

- **Parallel workflows:** Can teams work in parallel instead of taking turns working on a project? With Contentful, your content creators can input content, while developers work on the front end. An outside agency can build a new digital platform and pull content in via APIs. You can even fine-tune which elements local teams can edit, giving them flexibility where they need it while protecting universal brand elements.
- **Iterative development:** Instead of lumping changes into periodic product releases, leading companies release multiple updates daily. Does the CMS include tools and features that make it easy to experiment and test new ideas without putting your live site at risk? Is there a process for pushing these changes live without locking users out of the

CMS? Can you update the CMS itself without a content freeze?

- **Scalability:** Starting small and being able to scale later allows you to get to market and start getting customer feedback faster. Some CMSes require a big investment just to implement the platform, others let you start building right away and give you a clear path for scaling successes quickly.

APIS AND EXTENSIBILITY

Headless CMSes use APIs to deliver content to different digital platforms, but APIs can do much more than just content delivery. APIs and extensions help the CMS integrate with the other tools you use (PIM, personalization, automation, etc.), making the CMS more responsive to the way you want to work.

Choosing a CMS that is extensible helps you get more value out of your tech investments. Instead of adapting to rigid workflows or cutting and pasting content and data between systems, a headless CMS that leverages APIs and extensions can help your content flow from a central hub through your personalization, automation, PIM tools, and other technology.

When looking at headless vendors, you will see some described as "purpose-built" or "API-first." Despite some traditional CMS vendors trying to capitalize on the trend by adding on a content delivery API, API-first CMSes built their content management and delivery processes that way from the very start.

API-first headless CMSes can go beyond the basic content delivery API (a requirement of headless) to offer more flexibility and integration possibilities. For example, in addition to a content delivery API, Contentful includes APIs for content management, content preview, image management and [GraphQL](#). These additional APIs give product managers more flexibility to integrate tools, customize the CMS and build innovative digital touchpoints and products.



Getting buy-in for headless

Developers often champion headless CMS options because they readily grasp the implications of APIs, CDNs, SDKs and other acronyms that can make their colleagues' eyes glaze over. But the conversation also needs to focus on how a headless CMS helps you create, manage and use content.

Fortunately, there is some common ground where developers and other stakeholders can evaluate CMS options: content structure.

Some headless CMSes leave content in the digital equivalent of a bucket, with no additional structure to support the transition from single-channel to multi-channel content delivery. Adding structure takes content out of the bucket and provides a framework for organizing it within the CMS for use in multiple digital applications.

Good content infrastructure should be:

- **Unified, with all content in a single content hub.** This eliminates content silos and duplication. Editing is faster and easier – change the copy or image in one place, and that change is pushed anywhere the content is used. Content is consistent and can quickly be published across all channels. Unifying content benefits end users and developers by reducing the number of CMSes they have to learn and use, streamlining content operations and improving consistency.
- **Modular for easy to use and reuse across digital platforms.** Instead of lumping content together into pages, page components should be stored in logical chunks. This modular approach makes it easy to reuse content in different layouts and on different platforms. Structuring content for reuse respects content creators' time and increases the reach and value of every piece of content. It also makes it easier to quickly spin up new landing pages and microsites.
- **Organized to support collaboration and parallel workflows.** Modular content is only one part of the equation. Clearly categorized content empowers people to work in parallel, because everyone knows where to find different types and how they relate to each other. Content creators can create, edit, preview and publish using existing components, while developers build new features, software, etc. This offers a competitive advantage for enterprises that need to move fast and decrease time to market.
- **Extensible, so that it fits your needs both now and in the future.** Your content needs and preferences are as unique as your business. Start with a custom content model, then integrate your preferred tools and programming languages. The content model should be easy to modify as your needs change, without the dependencies and code bloat that result from customizing legacy CMSes. An extensible CMS should feel limitless instead of limiting.



Contentful is a headless CMS for the whole team

Contentful has all the features that developers love in a headless CMS, plus an editorial interface that will win over product, marketing and editorial teams. Notable features include search features that help you quickly find content for editing or reuse, collaboration tools and a long list of integrations that let you work from one dashboard instead of clicking back and forth between programs.

You'll love how fast developers can build and ship new products and new digital features for your product. But you'll also love how much more you and your team can do without developers.

Often, CMS decisions are a compromise between end users and developers. Contentful's leading content management API allows engineering teams to organize and manage even the most complex digital ecosystems. Our powerful APIs are well-documented, and developers have straightforward resources and support every step of the way.

Contentful is the only cloud-native vendor to be featured in the latest Forrester Wave. They've agreed that our "API-first and cloud-native approach excels," and that we're "a good fit for progressive digital initiatives that want to unify content services across channels and projects."

Learn more about why Contentful stands head and shoulders above other headless platforms.
[Talk to a Contentful expert today](#) and see how user friendly a headless CMS can be.

