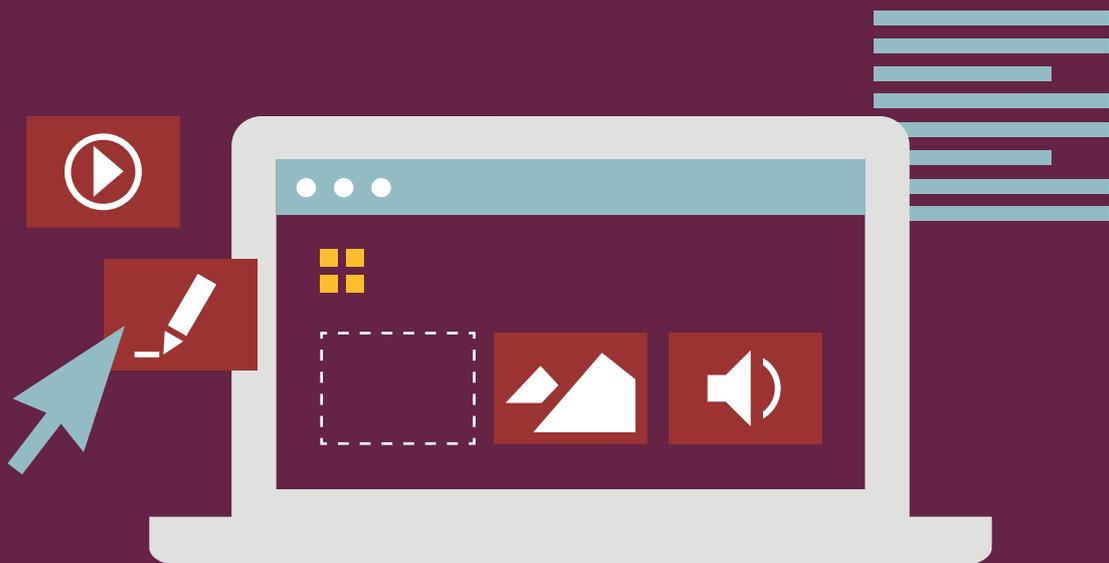


# Headless CMS: How to find the best fit for your enterprise



## Table of contents

Finding where headless fits in your content management architecture	<b>3</b>
What is "headless" and why the buzz?	<b>3</b>
<hr/>	
Advantages of headless	<b>5</b>
<hr/>	
Drawbacks of headless	<b>6</b>
<hr/>	
Types of headless approaches	<b>6</b>
When to go pure headless	<b>6</b>
Enterprise headless	<b>7</b>
Hybrid headless	<b>8</b>
<hr/>	
Magnolia for headless	<b>8</b>
Incremental migration	<b>8</b>
Personalization	<b>9</b>
The stories app: How Magnolia brings creative freedom back to headless	<b>11</b>
Headless with no limits	<b>12</b>
<hr/>	
Links to resources	<b>12</b>
<hr/>	
Contact information	<b>13</b>
<hr/>	

## Finding where headless fits in your content management architecture

Headless content management systems (CMS) are becoming increasingly popular. Key reasons for this are the high degree of speed, agility, choice and flexibility that they give businesses in creating and implementing their information architectures.

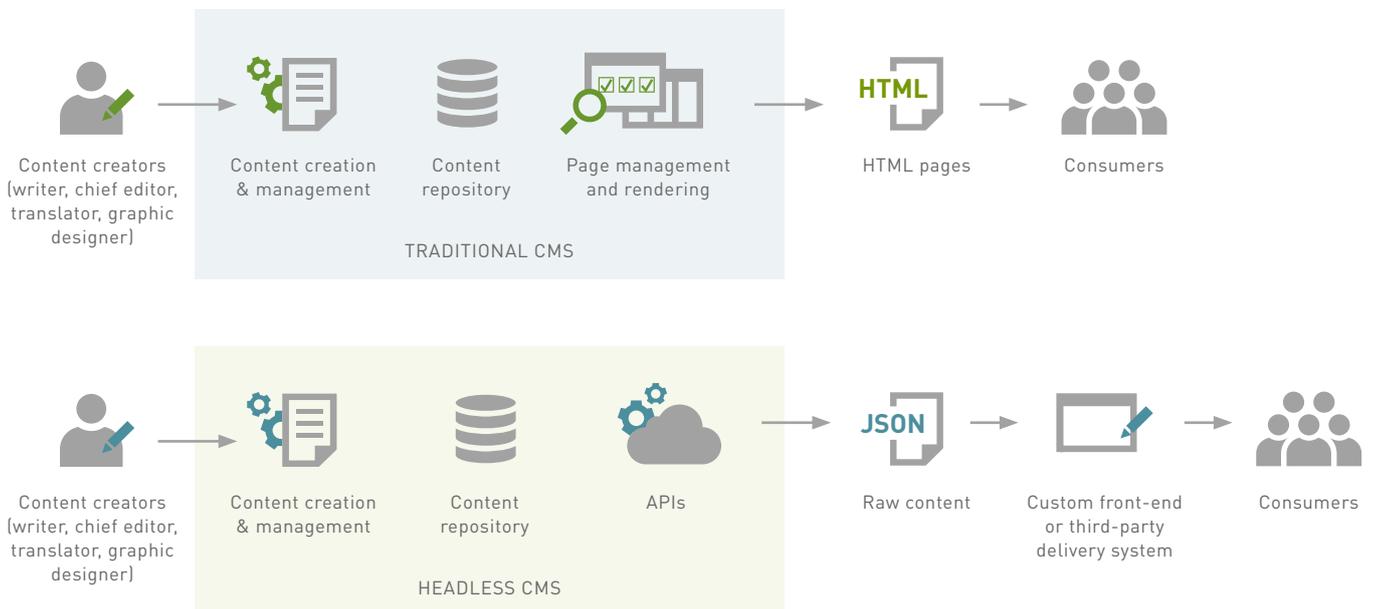
This paper looks at the advantages and drawbacks of pure headless systems and explains other approaches such as enterprise headless and hybrid headless. It provides decision-makers, managers and practitioners wearing the “digital” hat with pointers on how to tap the benefits of headless in their current architecture, while addressing the drawbacks, in order to keep their systems robust and future-ready.

### What is “headless” and why the buzz?

A headless CMS is one that does not do any rendering of content and where the rendering is decoupled from the management interface. Because of this, some use the terms “headless” and “decoupled” interchangeably, although “decoupled” generally refers to parts of the

architecture being separate. The figurative “head” refers to the front-end rendering or presentation of the content, while the “body” refers to the back-end storage and administration of the content.

A headless CMS only captures and stores content, provides a UI for creating and managing content, and provides a mechanism to access the content, typically via REST APIs as JSON. It is thus also known as “API-first” CMS. The content can be delivered to and integrated with a third-party system, such as an e-commerce tool, or delivered to and displayed using any front-end technology, whether in a browser, mobile application, or syndication service. Many headless CMS are also cloud-based solutions, also called “content as a service” (CaaS).



4

A traditional CMS deals with content creation, delivery and display. It usually has a back-end where users can enter content. This content is stored in a database, retrieved from the database, rendered into HTML on the server and then delivered as fully rendered pages to the browser.

A headless CMS decouples the rendering and presentation system and allows you to replace it with front-end or other technologies of your choice. The CMS is simply a content store and web application for content creators, and the

content is delivered to the front-end or other system via an API (application programming interface).

Strictly speaking, headless is not so much a specific technology, but rather an approach to content management as a whole. A headless CMS can be integrated into an existing IT infrastructure. It does not necessarily replace an existing CMS, as it can be interlinked with other systems.

Headless ensures that you store content in a pure format, ready for use in any context. Use content for multiple delivery channels, from desktop web to native mobile or touchscreen applications.



#### Company-internal channels

Product catalog, intranet, partner marketing, consumer website

#### External channels

Common: websites, mobile sites, native mobile apps, e-mail newsletters, RSS feeds, games

Advanced: social posts, media (AMP, Apple news, Facebook Instant Articles), kiosks (in-store, museum), travel entertainment systems (flights, cruises)

Digital signage: in-store (walls, checkouts, aisles), hospitals and medical centers, convention centers, hotels, sporting events, airports, cinemas

Innovations: vehicle systems, airplane cockpits, VR, IoT devices, interaction interfaces (chatbots, digital assistants), wearables and watch interfaces

## Advantages of headless

### Start instantly with no obstacles

The barriers to getting started with headless are very low. Headless is often available as a service, provides a basic authoring interface that is fast and easy to learn, and your front-end team can integrate it into your existing website.

### Fit into existing infrastructure

You do not need to change your existing delivery tier. Headless is open to all other systems such as CRM and e-commerce that consume content from it—access is provided through APIs.

### App-focused projects

Headless is ideal for developing web and mobile apps, as it allows practically any application—web, mobile, IoT, smart TV, touchscreens—to pull and push content. You are no longer working with a CMS that renders HTML from templates, but with one that provides content in its simplest form via content APIs your team needs for app development.

### Going into the cloud

Headless is often provided on cloud-based platforms. For enterprises aiming to outsource their infrastructure, this means less upfront costs, rapid iteration for digital projects and scalability.

### Work in parallel

Different teams—front-end developers, back-end developers, marketing and content editors—can get started right away and work independently of one another.

### Take a front-end approach

Empower front-end devs—they do not have to learn the inner workings of a CMS or its templating system because they simply work with a content API.

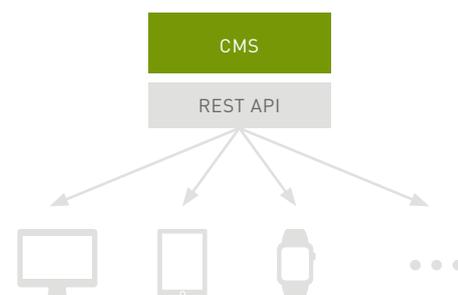
The developer team can use its preferred frameworks such as Handlebars, Angular and React, and have complete control over the end-user experience.

### Re-use content: create once, publish everywhere

Stop creating content in different places and having to deal with the problems of updating and versions getting out of sync. Headless lets you create content in one place and use it everywhere.

### Future-proof content: multichannel, omnichannel

Headless supports content that does not contain any presentation formatting, and that can be used and re-used on any channel, now and in the future.



### Microservices

Headless works well in a microservices environment and supports cross-functional teams working in an agile mode to get tasks done quickly. The front-end is uncoupled from the back-end. You get more flexibility and it's easier to switch and replace various pieces of functionality in the principle of best-of-breed.



## Drawbacks of headless

Headless brings lots of freedom, but with freedom comes responsibility. As a project owner, you might find that you increasingly have to do it yourself with writing, debugging, maintaining and troubleshooting code. You and your team will need to supply the features that would have come with a traditional CMS.

While many vendors in the headless space are developing their capabilities, there are things that pure headless solutions don't do well, such as asset management, navigation, security, workflow, access control, categorization and link management.

Content authoring tends to be primitive, as editors are often restricted to entering content in rigid forms. The experience of creating content is limited and editors may have to get used to doing without WYSIWYG.

A preview function is non-existent. There is no way of checking what a page would look like, let alone checking on previous versions of a page.

Marketers may no longer be able to create layouts without developer assistance. While they can manage content, they are not able to orchestrate experiences and map content to customer journeys—one of the key things that a full-featured CMS makes possible.

Personalization is difficult in a pure headless setup and you might need to rewrite a full personalization content engine. Eventually, you might need to consider implementing new types of content analytics.

---

## Types of headless approaches

While the distinction between types of headless approaches might seem artificial, it is useful in showing the tradeoff between different capabilities.

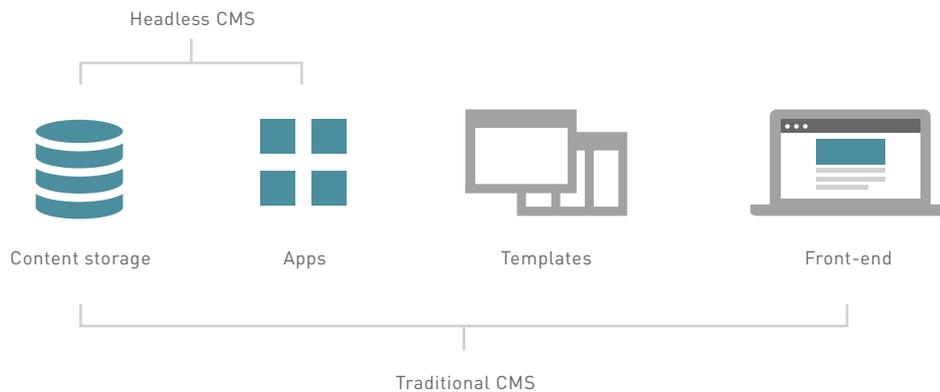
### **When to go pure headless**

Headless is ideal if initial project rollout speed is your main concern, and speed of updates and changes is secondary. Headless is well-suited for developer-centric projects, where devs can quickly adjust the presentation when needed,

and empowering non-technical, marketing teams is less important. It is also good for app-centric projects where presentation updates are seldom, and for smaller projects with a short life, e.g. a low-stakes marketing microsite.

7

If speed is your main concern, go headless. If you have a bigger project, invest more upfront for the longer-term advantages you get.



### Enterprise headless

Pure headless CMSes are in the process of building up capabilities, but they lack the robust features demanded by enterprise-level requirements. An enterprise headless CMS provides the benefits of headless, plus the features required for enterprise environments.

### Authentication and access control

A headless CMS does not offer the user groups, fine-grained controls or advanced user management that an enterprise needs. With an enterprise CMS, you can integrate with your existing user management infrastructure such as LDAP and provide convenient single sign-on for your users.

### Configurable workflow

While pure headless CMSes typically provide a simple workflow out-of-the-box, an enterprise CMS enables you to create the custom workflows that match your organization's actual processes.

### Personalization

Personalization is particularly challenging with headless CMSes, whereas an enterprise CMS has capabilities for setting traits, creating

content variants and serving these in the right contexts.

### Fully customizable REST endpoints

Headless CMSes provide standard endpoints out-of-the box, but an enterprise CMS enables you to create the exact endpoints that would best serve the needs of your projects.

### Translation

A good enterprise CMS provides support for the processes that professional translation companies employ and support integrated translation services via API.

### Hosting

An enterprise CMS gives you the option of hosting on your own infrastructure.

### Complex integration with existing systems

You can only manage the content stored in a headless CMS, while an enterprise CMS allows you to integrate with other content in your existing systems or third-party tools.



### Hybrid headless

Many CMSes are now positioned as hybrid to tap the advantages of headless while overcoming its drawbacks. A hybrid CMS is a combination of a headless CMS and a traditional CMS.

It combines:

1. A UI for entering structured content, and a UI for building experiences (such as pages)
2. A great REST API to access both structured content and experiences, and a server-side rendering system

Some benefits of taking a hybrid approach are:

#### Author empowerment

Marketers are free to create pages or campaigns they need and are less dependent on developers for layout changes.

#### WYSIWYG when needed

Content editors can still work with a page builder when they need it and they can preview the work —this improves the UX for marketing.

#### Speed to update website

Depending on the scale of the updates, non-technical teams can get things done with fewer obstacles.

#### Versatile use

A hybrid approach covers you for whatever needs are called for—you can deliver content using both the rendering pipeline and the REST endpoints.

---

## Magnolia for headless

Headless architectures have been technically possible for several years, also with Magnolia, a proven enterprise WCMS that leading global brands use in headless and hybrid scenarios. Magnolia lets you combine the benefits of headless with robust enterprise WCM features, and build with a no-limits, future-ready CMS that scales easily to your business demands.

Here are two scenarios where Magnolia solved problems for clients using headless approaches.

### Incremental migration

In this case, a customer was using an e-commerce system that included a built-in CMS. Over time, the built-in CMS became cumbersome and expensive to manage, and did not provide the features needed to run campaigns, create personalized experiences, rapidly publish and re-use content. The customer couldn't risk a complete migration to a better solution as it was tied to business-critical elements of the e-commerce system, and it also lacked the technical resources.

Magnolia filled the gaps: the customer incrementally migrated marketing content from its current system into Magnolia, then used Magnolia to deliver chunks of rendered HTML content via a JSON API that could be included by the existing e-commerce CMS. More specifically, the customer:

- Created content structures in Magnolia that solved the major pain points it had with its current systems
- Built templates and content apps in Magnolia that matched the structure of content that it needed to migrate out of the current system
- Copied the content from its current system to Magnolia, both manually and automatically



- Replaced the content in its current system with content managed by Magnolia using include mechanisms (via JSP and dynamic includes)

The migration process was low-risk, controlled and incremental. The customer could use a robust and mature enterprise CMS in a headless manner to manage content and campaigns, and access features such as personalization, translation and workflows. It could also re-use the content in other channels (like mobile apps, kiosks) and publish through matching formats such as JSON and XML. The customer didn't have to disrupt its existing infrastructure or re-implement the entire system. Magnolia's open, modular architecture allows easy adaptation to current requirements and future demands.

### Personalization

To understand how personalization is possible in a headless mode with Magnolia, let's take a brief look at how content is managed in Magnolia. First, you can manage content directly in the pages app, a WYSIWYG authoring interface. Content is stored there in a semi-structured way, meaning that the number and the type of used components per page can differ. Second, you can manage content in dedicated content apps. These allow you to mass-manage certain types of content in a pre-defined structure, e.g. teasers, jobs or company events.

Magnolia's architecture allows you to serve content in all kinds of formats and granularities. For a start, you can serve content from the pages app in a headless manner. One of Magnolia's hidden gems is that you can directly access components within a page. This means that you can call up specific components in HTML without the CSS and JS from the page.

This opens up interesting options for enterprises. Your existing back-end systems can do exactly that—call up different components and integrate them in the overall rendering. The upside compared to a pure headless solution is that you get full visual feedback on how the content is going to look like in your target system.

On top of that, you can serve personalized content. When the system calls for the relevant component and the right parameters are set to match the right traits, then Magnolia automatically sends the right personalized component variant.

If you need the content in JSON format and not in HTML, you can use Magnolia's REST API or you can use a component variant. A component template can serve either rendered HTML or JSON, depending simply on the .html or .json extension.

10

Magnolia combines the benefits of headless, plus the power of an enterprise CMS, in a way that no other CMS can offer. All this goes back to Magnolia's core strength in integration and customization.

The benefits of headless:

- Get started right away
- Fit in your existing infrastructure
- Work in parallel
- Future-proof, multi-channel content

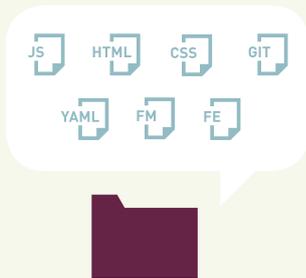
Plus the power of an enterprise CMS:

- Workflow, roles, security, personalization

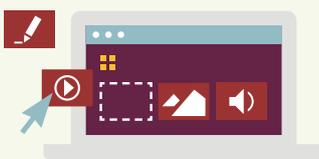
- Freedom to install on own infrastructure
- Page editor with integrated preview, analytics and personalization

In a way unique from other CMSes:

- Front-end, light development
- Fast setup and deployment via bundles and light modules
- Integration both via API or rich Java interfaces, plus an integration UX via back-end apps that you define



1. Magnolia's use of open APIs and data standards create rich integration and customization opportunities that make it easy to share and manage content across multiple sources. Magnolia's representational state transfer (REST) Web service allows you to manipulate content through a Web API. Developers can call any Magnolia command via REST and expose their own custom REST endpoints via Magnolia. Flexible REST APIs support headless scenarios and enable you to use semi-structured content.



2. Magnolia's content apps allow you to pull in content from different sources, manage content and publish to any format. These specialized web-based tools allow you to manage structured data productively and can be tailored exactly to your needs. Build a content hub to cover any structured content creation task that you have, such as managing event pages or your store's catalog.



3. Magnolia's light development approach allows developers to extend and customize Magnolia using common front-end development tools and technologies. Developers simply create directories and edit plain text files in common and easy-to-learn formats like HTML, YAML and Freemarker.



4. Magnolia's cloud offering provides flexible deployment and a rich feature set that includes easy authoring, rapid development and continuous delivery.

### A tale of two CMSes

There are cases where it might make sense to have a central CMS and then use a headless CMS in parallel. For instance, a CMS already widely used in your enterprise could be supplemented by a headless CMS that supports a new business area where the focus is on pulling out and rendering structured content, e.g. retail store locations or a dealer network.

### The stories app: How Magnolia brings creative freedom back to headless

A pure headless CMS is a double-edged sword. Headless makes it fast and easy to create content. It does this by using form-based data entry to enforce structured content that is easy to retrieve and re-use via REST. However, that efficiency and rigidity come with a price. Headless also limits marketers and editors to working with inflexible forms and a lean UI, and restricts their creativity in telling stories and building experiences.

Magnolia bridges the gap by combining the best of its two main authoring tools: the more rigid form-based content app and the more flexible page-based editor. The result is the stories app—an authoring interface that gives marketers the power and flexibility to create rich content beyond the limits of a strict form. It still generates clean JSON for developers to use in headless scenarios and to deliver structured content which can be rendered differently in any framework or channel.



#### Stories app features

- Generous, full-page editing experience
- Metadata section for structured content
- Flexible blocks section for free-form content
- Deliver as JSON via REST API
- Configurable block types
- Efficient keyboard shortcuts
- Content linking and previews
- Fast publishing

#### Use cases

- Articles for media companies, news sites
- Story-driven content commerce, rich product descriptions
- Landing pages
- Content curation (curating pre-produced assets)
- Blogging
- Headless content delivery

The stories app serves all the content—both the structured metadata and the free-form blocks—in clean JSON format via REST API. As the content is not delivered in one chunk—but delivered in clean blocks—the REST client, whether it be a web or mobile app, a kiosk or any third-party system—can use and present the content in any way needed. For instance, you can intersperse advertisements between blocks, arrange the blocks in a certain order, display only image or video blocks, or use the blocks in completely different contexts. At the same time, you can also use Magnolia's standard server-side templating to render the content to pages.



12

### Headless with no limits

Headless opens up interesting possibilities and challenges conventional ways of doing things. For many enterprises, it is no longer a question of whether to go headless, but where headless sits in their information systems. The trend towards microservices—whether that service involves content, content management, engagement, optimization or analytics—will

continue to fuel headless approaches. In this vision, the CMS acts as a central hub, processing and curating content and data from other tools and services, while at the same time sharing its own content and data via APIs. By staying true to the principles of flexibility, scalability, and empowering their content authors, enterprises will have no limits on creativity and innovation as they build their digital architectures.

---

## Links to resources

Check out the stories app in our technical documentation:

<https://documentation.magnolia-cms.com/display/DOCS/Stories+app>

## About Magnolia

Magnolia is a digital business platform with a CMS at its core. Enterprises such as Airbus Group, Al Arabiya, Avis and Virgin America use it as the central hub for their web, mobile and IoT initiatives.

Magnolia's unique open suite approach allows you to go to market fast, adapt quickly to changing business priorities and integrate easily with Commerce and other third-party systems. This enables them to create outstanding multi-channel customer experiences while maintaining central control over all content assets.

Magnolia CMS ranks on top position in the prestigious Emerce best 100 companies in e-businesses 2018, commercial CMS category.

### Contact Magnolia

Oslo-Strasse 2  
4142 Münchenstein (Basel)  
Switzerland  
+41 61 228 90 00  
[info@magnolia-cms.com](mailto:info@magnolia-cms.com)  
[www.magnolia-cms.com](http://www.magnolia-cms.com)

---

## About Info.nl

Info.nl has been a certified Benelux partner of Magnolia since 2008. Next to that for numerous years we have been listed in the EMERCE Top 100 Best digital agencies in The Netherlands.

We craft digital products that people love to use. We enjoy working with forward thinking companies actively looking for the next step forward. Our self-driven teams serve as agile product development partners, enabling our clients to use new techniques and injecting a different way of thinking into their business process. We offer expertise in the core disciplines of agile product development and digital transformation.

It is this passion that drives every unique project we complete, no matter how large or small. Our successful history includes clients such as Greenwheels, BVA Auctions, T-Mobile, Graydon, Adyen and many others.

### Contact Info.nl

Sint Antoniesbreestraat 16  
1011 HB Amsterdam  
Netherlands  
+31 (0)20 530 9100  
[newbusiness@info.nl](mailto:newbusiness@info.nl)  
<https://www.info.nl/>