WHITE PAPER

How to choose the best Headless CMS for your project

A comparative view of slim headless solutions and Magnolia CMS

Contents

Summary

Headless is here. But there are significant differences in what the headless CMS vendors provide. In order to help you choose a solution, we at Magnolia have looked into the main reasons for going headless and compared how a project implementation looks with a slim headless CMS versus with a full-fledged CMS like Magnolia.

In this paper we use the term “slim headless CMS” as a class of software, we add the “slim” qualifier since Magnolia is often used in a pure headless fashion as well. Concretely, with this term we are referring to products like Contentful, Sanity, Contentstack, Butter, Craft CMS, Prismic, Forestry, Strapi, Scrivito, and so forth.

We’ve gathered our observations and advice on the topic in this whitepaper. The purpose is to help you think through your project before you decide whether a slim headless CMS, or a full CMS like Magnolia - which can function both in a pure headless way and a hybrid fashion, and also incorporates advanced enterprise CMS features - is the most appropriate for you.

After reading this whitepaper you will understand the full business case of implementing a project with Magnolia as compared to a slim headless CMS vendor. If you haven’t specified your requirements yet, we want to offer some suggestions on a couple of points you should think through before embarking on the headless journey.

Overall our conclusion is that all headless scenarios can be solved with Magnolia on equal terms as with slim headless CMS, in fact our lead architects state that: “Anything you can do with a headless CMS you can do with Magnolia and then some”.

The things you want to achieve by going headless can also be achieved with an integration-focussed full CMS. The main reason for this is that Magnolia is built on the fundamental ideas that separating the delivery of content from content management gives you flexibility, and that features and capabilities can be added as you need them, not all at once in the beginning.

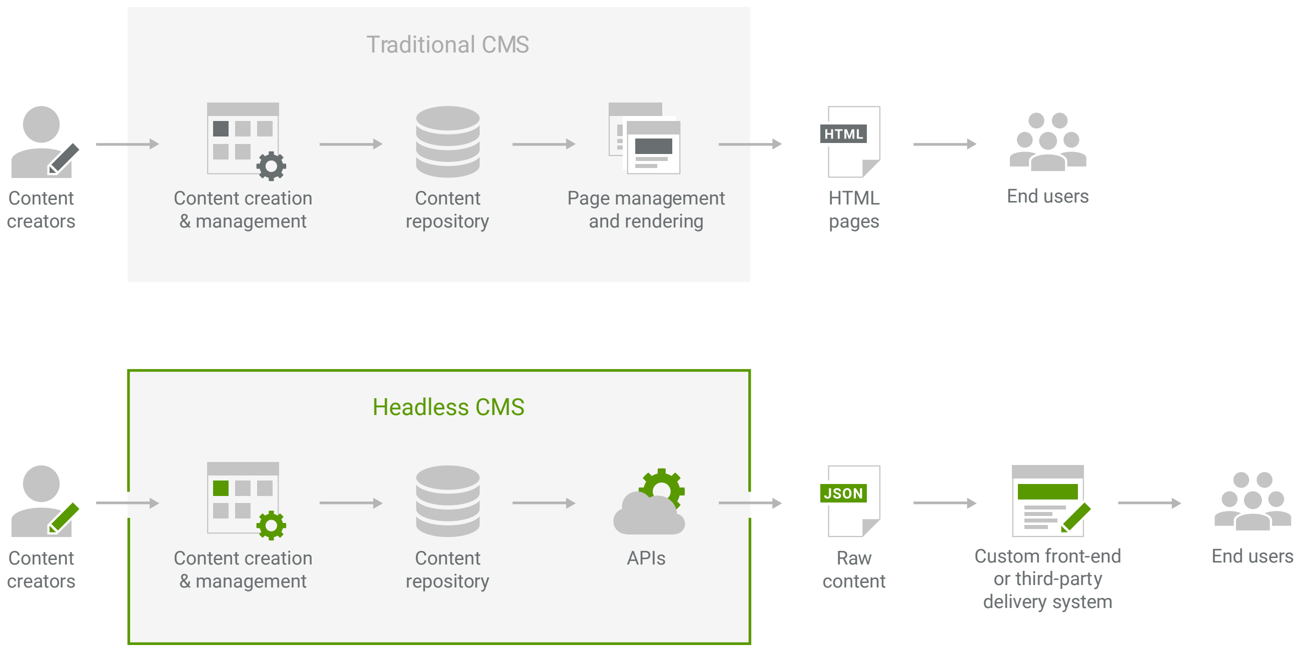
But with Magnolia those features are there when you need them, together with the professional services, support and documentation that comes with a vendor with a longer track record and a more fleshed out product. In fact, the first projects implemented with Magnolia were headless projects - but the term “headless” had not been coined yet. Those headless roots are still going strong and help you to implement first-class digital experiences.

What is a headless CMS?

A headless CMS does not generate a public-facing channel, such as a website. Its focus is on allowing content authors and business users to manage content. It then makes the content available to all digital destinations like websites, mobile devices, apps, in-store displays, print, IoT - typically via REST APIs.

Going headless nicely follows the “Content is king” credo and forces the project implementers to think of content-as-a-service that has to be delivered in a clean form that is usable at the point of delivery, regardless of what that delivery channel might be.

It is considered a good approach to create digital content that will work on multiple channels (aka frontends), where each channel often has control of how the content is presented.



Fictional scenario: why headless in the first place?

To help you make the business case for either a slim headless CMS or Magnolia, we'll start from a fictional scenario, then open up what you should consider every step of the way, and how the two types of solutions stack up against each other for these considerations.

Your old CMS has grown unmanageable. It is based on an old version of one of the large CMS systems so the license fee is fairly steep. However, because it has been allowed to homegrow without regular upgrades, it is now impossible to upgrade to a new version of the software and maintenance is 100% dependent on in-house knowledge about the platform. It has become a significant disadvantage in terms of time to market for new solutions, as well as in terms of managing the content.

The plan is to select a new content management technology with the main objective to create an efficient environment for managing and delivering content through a multitude of channels, and across an unpredictable variation of screens and digital devices.

The CMS is to be integrated with the existing e-commerce platform, CRM, and other business applications. It is appealing to treat content-as-a-service and implement a product that is scalable with growing needs and budgets.

The course of your headless project

CMS projects are usually about replacing a content management infrastructure with a new one and often changing the look and feel of delivery at the same time.

In a headless project, as soon as you have defined the content model, you can work on the frontend and on the content in parallel because they are decoupled. Even if your aim is to eventually completely replace a prior CMS, you normally start by building a smaller part of the new setup, maybe one website, an app, a shop or an interface for an IoT device, and then move to the next one.

Usually the first project is either the one that most urgently needs replacing, or a simple one that can be delivered quickly. Larger content repositories with more complexity are postponed for later.

Those could include channels that require language versioning, check points for compliance or ones with a large number of editors and publishers.

This incremental process will work for the next couple of products that are launched, but then comes the time to migrate all of the company websites and the customer extranets to the new content management system. Typically they have shared content that you want to create in a master version in English, and then localize via an external translation agency or the internal network of local marketing and sales reps who adapt the content to their markets.

Even with a full CMS, choosing and implementing the specifics of the workflow for such large projects can be cumbersome. With a slim headless CMS, while you have more freedom, it will be a lot more work, because the CMS does not offer support for this. It requires defining and implementing a workflow from scratch.

Similarly, the slim headless CMS approach gives you freedom when it comes to integrating with your best-of-breed selection of other tools and applications (shop, marketing engine, analytics tool, CRM) but you have to implement your connection to them. With Magnolia you still have that freedom, but you also have connectors available for the most important integration targets, via connector packs.

As the content management is spread out among an increasingly diverse group of editors and authors, not everyone will find it as easy to create content using a simple form-based editor of a slim headless CMS. They need preview functions and guidance in order to fill in the fields correctly. These too do not come out-of-the-box with slim headless CMS.

A quote that we frequently hear from failed enterprise headless projects is: “In the end our content authors got lost in decoupled ‘cockpit views’. It became impossible to manage the site”.

The thing is, the process and tools for creating content don’t just impact how efficiently it can be created, they also influence the quality of the content itself.

If you’re a large enterprise, you also need tools to ensure compliance with various laws, company policy, and brand guidelines. Even smaller enterprises prefer to be able to control their content, its quality and the versions that have been publically available at various times. This logic eventually has to be in place somewhere, but is it available out-of-the-box, or is it another feature that needs to be implemented?

So, as the project progresses, in many cases organisations who started out with a slim headless CMS discover that the development hours required to build out the logic that makes the system work not only for the content, but also for the organisation, are on par or exceeding the cost of using a full CMS with more of the functionalities coming out-of-the-box.

With a product like Magnolia you can also use a minimal headless approach to get started, but then add additional out-of-the-box features as the needs arise. Organisations increasingly take this approach with Magnolia.

So before you decide to go with a slim headless CMS, consider the following questions - for now and for future project phases:

1. **Form-based editing tools: Are they sufficient for your content and authoring teams?**
2. **Integrations: Should they be visible and usable by authors within their editing tools?**
3. **Developer resources: Will they be available on an ongoing basis?**
4. **Navigation: Should content authors be able to modify it, and make larger layout changes themselves?**
5. **Access control and security levels: Do you want to restrict access to certain collections of content?**
6. **Workflow: How much do you need and do you really want to create it all from scratch?**
7. **Personalisation: Will your frontends deliver personalized content? Do you want to build the personalization system and authoring UI?**
8. **Project size: Is it a large project, such that the initial setup time is less significant?**

In-depth comparison: slim headless or Magnolia CMS

In what follows, we'll take you through a deep dive and comparison of the features and capabilities that are relevant for your headless project.

Technical Foundation

REST APIs

Slim headless CMS	Magnolia CMS
A comprehensive set of API's for content delivery and content management. API's offer full control of the CMS. Typically "API-first" software, they have a well designed set of named APIs. A GraphQL endpoint is sometimes available.	A modern full-featured content delivery endpoint delivers content trees as well as items and lists. A low-level nodes API provides content and configuration management. A commands API allows triggering any Magnolia feature remotely.
APIs can not be added or customized.	All API's can be deeply customized. New API's for custom searches or business logic may be added with Java. In addition, Magnolia allows to deliver pre rendered, personalized content fragments on any channel.

Flexibility for developers

Slim headless CMS	Magnolia CMS
Beyond defining custom content types and fields, customization options of the authoring interface are very limited.	As an open-source full CMS, Magnolia offers deep options for customizing any aspect of the CMS, including custom user interfaces.

Content modelling features

Slim headless CMS	Magnolia CMS
The content model is based purely on content types. It has good support for these including custom content types and links between types.	Magnolia supports the same content type features, with custom types and links between types. Native support for hierarchical content - trees of folders with content items in them. Magnolia supports two additional models: • A flexible page model accommodates a free-form authoring and layout system. • A story model which supports a stack of blocks similar to the “Medium editor”. This model combines the simplicity of a content-type with some of the freedom of a page editor.

Managing content models

Slim headless CMS	Magnolia CMS
Includes a GUI which enables users to create and edit the model of a content type, even without development skills. Developers can export and import the underlying content model to files with an external tool, or interact with them via API.	In Magnolia, defining a content type is considered a developer task. Content types are stored in simple configuration files, which are automatically loaded. This has many advantages, they are easy to share and copy, and it’s easy to track and manage changes with standard developer tools like Git. Magnolia’s content type definition is the slimmest on the market.

Editorial Experience

Editing tools

Slim headless CMS	Magnolia CMS
The editorial experience is form-based, and very clean and simple. Editing via a form, completely removed from the final experience is sometimes too abstract for some content authors, and too limited for some forms of content.	Three editing tools expose the three Magnolia content models: • Content apps provide a form-based editor. • Pages apps provide a flexible visual page editor. • Stories apps provide a block-based editor. You provision the tools that are most appropriate for your content and the skills and needs of your content authors. For any of the tools you can also provision folder trees or other content hierarchies. This is a familiar and convenient way for authors to organize their content.

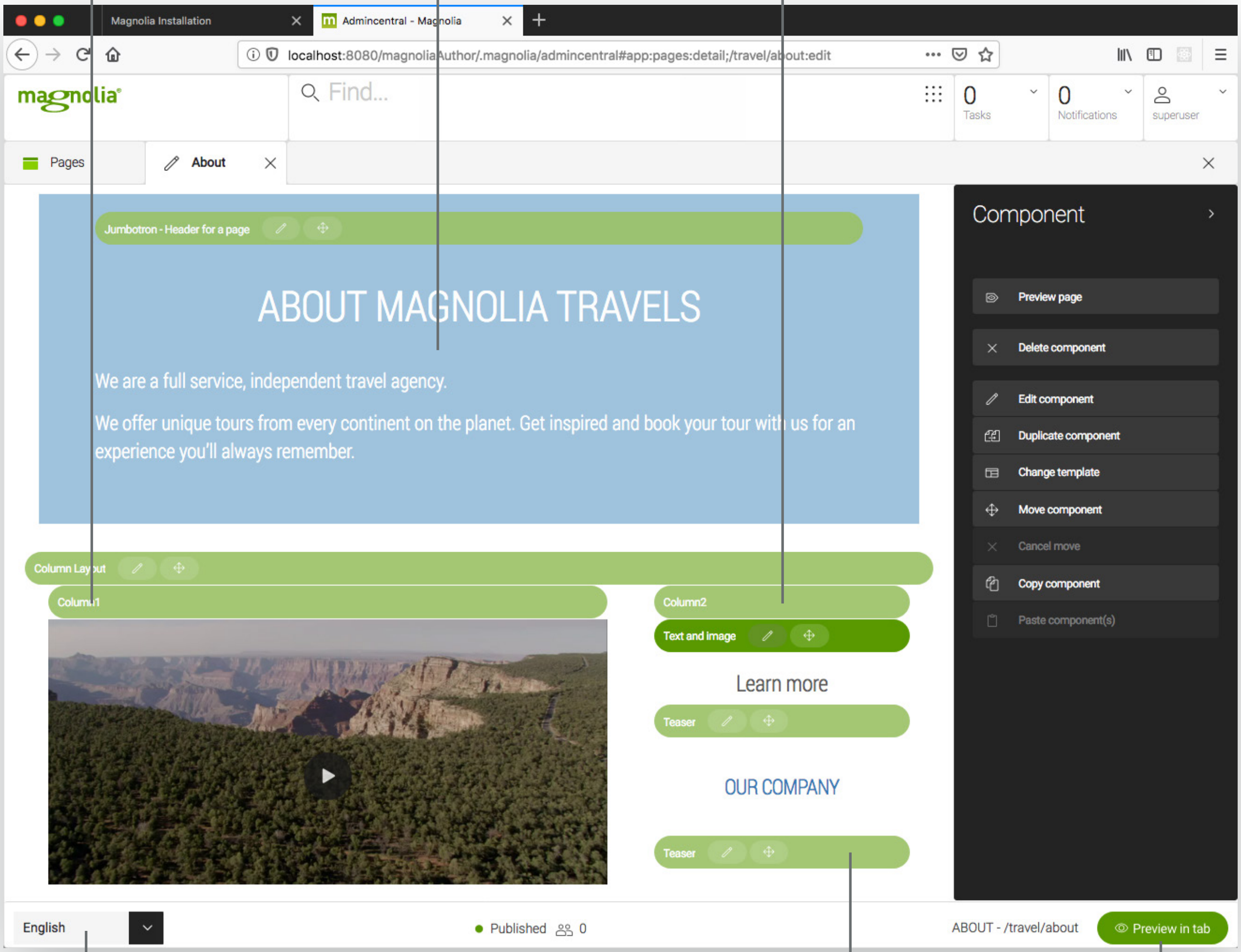
Support for editing SPAs (Angular, React, Vue)

Slim headless CMS	Magnolia CMS
With the form-based editor, authors can edit the items that are listed in a SPA, but they cannot edit the SPA itself. It's technically possible to also manage rich content like a landing page, but the author won't see a clear connection to the content they are creating and the workflow is awkward and not intuitive. In general, developers will be needed to make any navigation or structural changes within the SPA.	Magnolia's visual page editor, with live preview supports editing SPA as well as static pages. It is perfect for editing landing pages and other rich content. Authors have total flexibility to place the content components they want, where they want. And they can see exactly what they are creating, because the page editor loads the actual SPA. Support for trees of content gives authors intuitive editing of the navigation within the SPA.

Creative control - choose the components you want, place them where you need

Visual editing - authors work in context

Drag and drop components



Visual SPA Editor in Magnolia CMS

See the interface in your own language

Copy and paste components

Preview the experience as end users see it.

Custom UI

Slim headless CMS	Magnolia CMS
Some support custom field and item editor widgets.	Supports custom field and item editor widgets. Supports custom item list views, and custom widgets in column rows. Supports custom controls for filtering and querying items in lists. Action buttons with custom behaviour can be added to any editing interface. Integrations to 3rd party systems can modify the UI to add data views and controls right where and when content authors can use it, for example displaying analytics, or inventory from an ecommerce system. This saves authors from needing to login to, and jump between different systems to accomplish tasks.

Localized User Interface

Slim headless CMS	Magnolia CMS
UI is available only in English, with no option to provide an additional language.	UI is available in English, German and Spanish and you can provide your own custom language files as well.

Platform and Key Features

Saas and Cloud versus on-premise

Slim headless CMS	Magnolia CMS
Most are Cloud SaaS only. As a SaaS it is very easy and fast to get started. Just sign up and a trial instance is provisioned for you. If you want local backups of your solution and content it usually can be done, but you have to set it up for yourself.	Magnolia is available both as SaaS and On-Premise, providing flexibility in terms of server configuration and backup systems, including a hybrid cloud approach. Magnolia's SaaS offering includes Integration, UAT and Production environments for best practice continuous delivery, and automated backups.

Platform maturity

Slim headless CMS	Magnolia CMS
Young platforms with exciting prospects. In some cases customers will find themselves as part of a development process where features and documentation for new scenarios will be available on-demand and not out-of-the-box. Best Practice can be difficult to pinpoint because it is something that grows over time and requires that the vendor takes an active strategic interest in its' continuous definition.	Seasoned platform. All common digital experience features are available. Extensive and deep documentation with many examples and tutorials. Three SLA's are available. Standard developer trainings and certifications are available. Custom trainings and project support are available. There is an established network of certified implementation partners all over the world. Magnolia trainings and documentation provide Best Practice recommendations based on partner experience and customer consultations.

Analytics

Slim headless CMS	Magnolia CMS
Offer some direction on how to set your CMS up with analytics but doesn't come with a clear cut direction on how to manage analytics and related topics such as marketing tags and personalisation of content.	Marketing Tags feature makes it easy for marketers to add JS tag-based tracking systems for analytics. Our analytics visualizers give authors the information about content performance, right where they need it, at their content.

Personalization

Slim headless CMS	Magnolia CMS
No personalization facilities in the product. Personalization can be implemented by developers in the frontend, for example by loading only specific content based on tags or other attributes.	Allows authors to define audience segments based on implicit traits, or traits from external systems like connected CRM. Authors target which audiences receive which content. They can even preview the entire digital experience based on convenient personas. All without developer intervention. REST endpoints then deliver the personalized content based on the traits of the current visitor.

Publishing workflows

Slim headless CMS	Magnolia CMS
You will have to define and build functionality and integrations to support the workflow you need considering aspects such as: Content State, Publishing rules, Editor rights, version control, integration with translation services. Scheduled publication must usually be added via an external integration or tool.	Provides a 4-eye publishing workflow out-of-the-box and supports the creation of custom workflows, all based on the popular JBPM workflow engine. Workflows can contain a mix of steps for authors, publishers, other roles and even other systems. Use Magnolia as the cornerstone, or integrate Magnolia as part of a business process managed architecture. Scheduled publication is supported out-of-the-box.

Maintenance

Slim headless CMS	Magnolia CMS
Because many features need to be implemented outside of the CMS, that custom software will need to be maintained. If not by the original developer, then new developers will need to learn the custom in-house code.	Any customizations implemented in Magnolia will need to be maintained. Any developers who know Magnolia will be able to maintain the customizations. Special tooling helps by reporting outdated configurations.

Conclusions

We hope this document has helped you visualize your upcoming CMS project, and in light of that to understand what the relative strengths of both Magnolia and slim headless CMS are, and in particular what hidden costs and efforts can be associated with them.

It comes down to a few key factors:

Who is on your team?

And who will be joining that team? If you have technically savvy content authors, and the authoring team will be well trained and using the CMS daily, then a slim headless CMS can work because they can overcome some of the shortfalls of the limited UI. If needed they can also log into the other tools of any integrated systems, and use tools together that way.

But otherwise, if the authoring experience is important in your organization, if they need creative content tools such as to optimize landing pages, or if some of your authors are less technically-savvy, Magnolia is the better choice.

For a successful slim headless CMS project, you will likely need strong developer resources ongoing. This is because when content authors are limited in what they can do, then you will need developers to make most updates to the frontend channels.

How big is your project, and what is your timeline?

A key benefit of a slim headless CMS is the fast ramp-up. You can have a running instance in a few minutes, and the delivery API means that developers can be productive right away. This makes them a great fit for smaller projects, because you may not need the extra depth that Magnolia brings.

For larger projects, the length of the startup phase is less significant. More importantly, they are more likely to require the enterprise-class CMS features that slim headless CMS do not include. On a larger project, you are likely to launch sooner with Magnolia because the features are available out-of-the-box.

What are your most important requirements?

Both solutions provide:

- Full REST APIs,
- Content models,
- Form-based editing tools.

But when it comes to your unique combination of needs and requirements, then we recommend that you take a closer look at the differentiators:

Slim headless CMS

Very fast setup

- Provisioned in minutes
- GUI to create content tools

Low prices for smaller projects

- Pricing scales with usage

Magnolia CMS

Superior authoring

- Visual SPA Editing - Preview, Context, Creativity
- Stories app
- IUX - Usable integrations. Authors see data and controls from 3rd party systems, right where they need it, at their content.
- Intuitive content trees

Enterprise features you will need

- Publication Workflow
- Personalization
- Advanced Roles and Permissions
- Connectors for E-commerce, Analytics, external DAM, CRM and more

Depth - unlimited customization

- Customizable authoring UI
- Custom REST Endpoints
- Deep integrations

Deploy anywhere

- On-premise or Saas

Meet the author

Christopher Zimmerman

Product Manager, Magnolia

Christopher is a product manager at Magnolia with an emphasis on developer experience & productivity. He helped introduce the 'light development' paradigm and is now focused on headless, hybrid headless and making integrations easier to implement. While trained in Physics in university, the buzz and wild west openness of software development drew him to a career as a programmer in product companies, creative web agencies, freelancing and startups.

Christopher is an outdoors enthusiast who got started with backcountry camping in the USA, but is slowly getting the hang of finding a coffee and cake in a rustic hut at 3000+ meters altitude in the Swiss Alps.

Take the next step

Magnolia is a leading digital experience software company. We help brands outsmart their competition through better customer experiences and faster DX projects. Get full headless flexibility and seamless workflows across best-of-breed digital experience stacks. Global leaders such as Tesco, Avis, Generali and the New York Times all rely on Magnolia for maximum reliability, high-speed project implementation and exceptional omnichannel experiences.

To learn more about what Magnolia can do for your headless projects, visit: https://www.magnolia-cms.com/product/headless.html.