Content management systems are undergoing a shift to accommodate front-end web development and multichannel experiences at scale.

Headless, Hybrid, and Beyond: Demystifying the New Wave of Web Development Technology

April 2022

Written by: Marci Maddox, Research Director, Digital Experience Strategies

Introduction

The global pandemic was a turning point for many organizations to urgently modernize their online presence to attract and retain customers. Technologies related to the creation of content and the channels of customer interaction that previously were nice-to-haves suddenly became a mission-critical foundation for future growth, innovation, and agility.

Web technologies have continuously evolved over the past 20 years to accommodate the rise of the web developer and the introduction of JavaScript and API frameworks that support a multichannel world. Separating the core content creation task from the publication process gives web teams the freedom to improve the quality and timeliness of website content. Web teams can create dynamic and engaging customer experiences for a growing number of connected devices and applications with their own distinct formats.

Content authors want the agility to create content quickly at the same time web designers require the freedom to control the presentation, performance, and responsiveness of the digital experience. As marketers and web developers adopt modern agile content creation and delivery practices, they are turning to their IT counterparts to assist in selecting, implementing, and supporting a modernized digital experience technology stack. In February 2022, IDC survey research found that 64% of organizations want access to the developer tools and technology necessary to improve efficiency in and delivery of innovative digital products and services.

The content management industry is undergoing a shift from on-premises platforms to modular cloud-based systems. New headless and hybrid content management systems (CMSs) are combined with front-end web development tools to offer flexibility in delivering multichannel experiences at scale but come with a few caveats for the traditional content creator. Investment in new web technologies necessitates consideration of the pros and cons of each approach and the impact it has on the editor, developer, and IT owner.

AT A GLANCE

KEY STAT

64% of organizations want access to the developer tools and technology necessary to improve efficiency in and delivery of innovative digital products and services.

WHAT'S IMPORTANT

New headless and hybrid content management systems, combined with front-end web development tools, offer flexibility in delivering multichannel experiences at scale.

Growing Demand for Technology Options

As content management systems advance, organizations are measuring their success in terms of business improvements. In a February 2022 IDC survey, business leaders identified customer satisfaction (36%), operational efficiency (36%), and innovation (29%) as their top 3 business priorities. As a result of investments in digital transformation, organizations saw a 25% improvement in business agility and a 24% improvement in time to market.

Marketers and web developers should consider the following core elements of content creation and delivery:

» **Content repository/back end:** The back-end system is the interface to a repository for organizing, governing, and storing content.

» **Front-end presentation layer:** The front-end system controls the design elements of how the content will be displayed to the end user. HTML controls the layout on a web page, the design is controlled by cascading style sheets (CSSs), and basic logic would be enabled through JavaScript. Developers can build single-page applications (SPAs) and progressive web applications (PWAs) to control the end-user experience.

» **Application programming interfaces (APIs):** API frameworks allow communication between back-end and front-end software as well as connection between data sources using RESTful-, JSON:API-, or GraphQL-based interchanges.

» **Unified/traditional CMS or web content management (WCM) platform:** The CMS or WCM platform, traditionally deployed on premises, offers a unified content authoring environment, workflow engine, and presentation layer to create, edit, and publish digital content such as text, images, and interactive graphics to websites and mobile web apps. A modernized WCM platform based on a model-view-controller (MVC) software design or web services can loosely decouple the content business logic from the presentation code, providing a separate back end (data layer) of structured content services accessible via APIs.

» **Headless CMS:** A headless CMS is an API-based back-end application used to manage, store, and track structured design-agnostic raw content elements. The pure headless CMS does not generate any front-end code, allowing web developers to use the APIs to send and retrieve content and apply presentation logic (e.g., localized language display or accessibility controls for text size or closed captioning) or personalized preferences (e.g., style themes or shopping recommendations) to dynamically generate the digital experience.

» **Hybrid CMS:** A hybrid CMS is a blend of the headless back-end content services architecture with the front end–aware content authoring environment of the traditional CMS. Web developers code the front-end experience separately, and marketers use visual tools to create, edit, approve, and preview content in the context of the designed web channels. The hybrid CMS can be deployed as a unified CMS with a combined back end and front end or as a standalone headless option with API access to basic content services.

» **Static site generators:** These front-end systems allow developers or tech-savvy marketers to build web pages using the content pulled from back-end systems. For example, static site generators such as Hugo or Gatsby are often deployed alongside headless or hybrid CMSs for their ability to minimize security risks, reduce complexity for developers, and improve static web page performance through a content delivery network (CDN).

» **Front-end development frameworks:** These include Angular, React, and Vue.js, which have reusable components for coding web pages. Pre-processors allow web developers to build complex definitions to avoid routine tasks and inefficiencies in writing pure JavaScript, HTML, or CSS output. Modern developers seek auto-deployment from source repositories such as GitHub and high-level languages to address more complex tasks, such as Haml for processing into HTML and Sass for processing into CSS.

Gaining a Competitive Edge with the "Best Fit" Technology Stack

Each CMS category can be characterized by the level of control and technical skills needed at the content, design, and administration layers summarized in Table 1 and as follows:

» **Unified WCM systems** focus on codeless tools for business users to create and manage web, mobile web, and other HTML browser-based experiences. It is a misconception that modern WCM systems cannot distribute content to touch points such as mobile apps, digital kiosks, or smart wearables. Unified WCM systems empower marketing teams with the greatest usability in creating website content using highly visual interfaces and prebuilt design components. The content repository may include digital asset management capabilities and workflow collaboration for content reviews, approvals, and translations. Editors can create simple websites without IT help, but more complex designs or intricate data transactions may require customization.

The complexity of unified WCM systems requires technical skills and time to set up and configure the system and a higher degree of education for content editors to learn all the componentry. One downside of the combined layers is the predefined structures that limit how the site will render and the channels to which the content can be delivered. Adding new site definitions or device channels requires new presentation templates to be created by developers, customization with plug-ins, or updates from the vendor. Another downside is the dependency of code development to be tested against the back-end system for compatibility, which limits the ability to quickly iterate changes to the front-end design.

» **Headless CMSs** address the needs of the web developers for content reuse and delivery to multiple front-end designs, including websites, commerce applications, mobile apps, or smart devices. The API-first architecture supports a wide array of data connections and integration of best-of-breed tools. Front-end developers can easily add new interfaces to customer preferred channels, and the back-end content workflows can ensure content is formatted correctly before publication. Headless solutions also support SPAs by no longer requiring a full web page to load from the server before the user engages with the site. Additionally, headless delivered PWAs act like regular websites but offer the user local functionality such as push notifications, offline work environments, or browser-based plug-in actions. A headless CMS also adds an extra layer of security for the website because the content database is not accessible directly from the publishing platform.

From the authoring perspective, marketing teams can create content autonomously and independent of the front end being completed. Headless solutions pose a challenge to the authoring experience because of the lack of visual editing tools and content preview templates found in the traditional WCM platform. Organizations also need to build the content model independent of the delivery model and structure the content as componentized elements that support dynamic assembly at runtime. Marketers lack the ability to make changes to the layout or add new content blocks without a front-end developer to manage the code changes.

» **Hybrid CMS solutions** strive to offer the best of both worlds: a headless developer-friendly, API-based back end with the content personalization and visual authoring environment that marketers crave. A hybrid CMS, also referred to as a progressively decoupled or agile CMS, allows nontechnical users to build and update digital experiences and offers users flexibility, ease of use, connectivity, security, and governance at scale. Hybrid CMSs allow for back-end extensibility and ease of integration to both legacy and newer applications. This interoperability enables business users to seamlessly interact with the systems with minimal IT involvement. The hybrid CMS fosters collaboration between marketing and development for easier content authoring/preview and better control to connect multiple delivery channels. With some of the development work for content synchronization, templates, and publishing workflows already in the base system, the cost to maintain and extend the application is reduced. The disadvantages of a hybrid CMS are the content preview is often limited only to website channels and personalization of the content requires custom development. As the coupling between the back end and the front end comes closer, creating iterative presentation models becomes more limited.

» **Static sites and front-end frameworks** offer the most control of the user experience but lack the support and reusability of a commercial application. On one hand, generating static sites is fast and web developers are given the freedom to design the experience as needed. On the other hand, developers need to be aware of the performance of the page and support as much server-side rendering as possible — a task traditional WCM systems have tackled already.

TABLE 1: *Content Management Systems Pros and Cons*

	Advantage	Disadvantage
Unified WCM system	Codeless content creation and publication processing; easy for nontechnical users to adopt; templates and native integration to other apps; API access to content; provides features for translation, navigation and link management, access control, personalization, and caching; delivery to predetermined channels; library of preconfigured content components, templates, and themes	Monolithic applications in which the user interface and data access code are combined into a single platform; proprietary tech stack and MVC architecture with limited functionality accessible via APIs; front-end content display may be limited by back-end control; custom coding cannot easily iterate separate from the back-end CMS; large install size and difficult to maintain due to breadth of functionality
Headless CMS	Front-end presentation design flexibility; add new channels faster; scalability and offline content creation; easier content reuse; content creators work independently of developers; publishing content to any Internet of Things (IoT) device; easily integrate with other best-of-breed applications; developers use the frameworks of their choice	Separate content repository with structured content model; lack of content delivery preview; no visual content editor; training for nontechnical content authors; limited workflow or content governance; developer skills required; reliance on IT to push content live or change layouts/designs of the web page
Hybrid CMS	All functionality is accessible via APIs and can be used in both a unified state and a headless state; visual content authoring and preview environment; developer APIs for front-end design and application integration; multichannel workflow; ready-made publishing templates; developers use modern front-end frameworks	Limits content and design controls because it contains both back-end and front-end layers; personalization dependent on custom development; content preview limited to website channels; more difficult to leverage microservices with ties to a front-end presentation layer; content is tied to a traditional publication workflow

Static site generator	Fast site creation and rendering performance; modern front-end frameworks used	Lack of dynamic personalization; limited integration to other apps; requires some technical knowledge to build web pages
Front-end development framework	Acts as a decoupled application with server-side rendering; large dev pool and career path options; JavaScript frameworks can work with both unified and headless CMSs	Upgradability due to custom code not connected to the CMS; introduces duplication of components for different site instances; no content preview functions

Source: IDC, 2022

Bringing It All Together

IDC survey research shows that organizations are increasing spend on digital experience delivery, of which 25% are investing in making their sites more interactive and engaging with video, audio, or rich internet applications. IDC predicts that by 2024, DevOps maturity will shift further left, with 50% of DevOps teams evaluated primarily on business metrics such as customer satisfaction, margins, cloud costs, and other business outcomes. Organizations are defining key metrics that center on the speed and agility with which content can be created and the level of engagement by the end customer. The number of channels managed across multiple regions adds to the deployment complexity and the level of skills required to maintain them. The goal should be to find a balance between the content authoring and delivery needs of the organization. When considering a headless or hybrid CMS solution, organizations should keep in mind the following areas of distinction:

» Level of API dependency means more access needed by developers to the content.

» Structure of the content is modular and ready for reuse.

» The experience is led by iterative development with modern code-centric applications.

» Flexible level of control is needed on the front-end design.

» Channel or device connectivity changes frequently.

Considering Acquia's Hybrid CMS

Leveraging its open source heritage, Acquia offers a hybrid CMS option boasting interoperability with static site generators and front-end developer frameworks alike. Acquia can be deployed as a headless system with support for front-end JavaScript tools or as a unified CMS with packaged solutions via the Acquia Marketing Cloud and Drupal Cloud. Acquia maintains a front-end editorial capability while allowing the back end to serve as an API-based repository. As a hybrid CMS, Acquia provides low-code tools and prebuilt components for the authoring environment, allowing any user of the system to publish content to multiple channels without developer assistance.

Acquia also supports both structured and unstructured content to accommodate the API demands of a headless system and the unified view of content delivery desired by the marketer. Organizations that need the power of iterative development coupled with the flexibility of front-end design will find that a hybrid CMS offers a good balance of nontechnical and technical user support. Acquia's open architecture supports seamless integration with legacy enterprise applications and best-of-breed modern cloud-based apps with a community of Drupal developers behind it. As an enterprise-grade CMS, Acquia provides advanced security, content workflows, personalization, analytics, and role-based permissions, typically offered by a traditional unified CMS.

Challenges

Acquia offers SaaS or managed private cloud options for an open source CMS solution built on Drupal. Organizations that adopt Acquia's hybrid CMS will need to have Drupal developers or a partner in place to accommodate any customizations or coding tasks. Despite the performance and agility found in hybrid CMS solutions, some organizations have avoided open source options because of concerns over maintenance and security. The best commercial open source CMS platforms maintain security-compliant architectures, extensive documentation, sound support packages, and endorsed partner ecosystems. As with other commercial headless CMSs, when Acquia is deployed in a headless mode, organizations must also invest in web developer skills to design the front end that marketers use to deliver rich, immersive digital experiences across multiple channels.

Advice on Evaluating Headless and Hybrid Solutions

Organizations are continuing to invest aggressively in digital experience delivery. IDC survey research shows that 74% of organizations are increasing spend on technology to holistically manage customer engagement processes. For most organizations, however, delivering a great experience on a website is no longer enough. Millennials have replaced baby boomers as the cohort with the greatest purchasing power and want to engage with the brand through a growing array of channels that did not exist a decade ago, including mobile apps, social networks, wearables, and IoT devices.

> 74% of organizations are increasing spend on technology to holistically manage customer engagement.

Interest in headless and hybrid content management has grown in response to these challenges — driven even more by the popular static site generator and front-end frameworks that allow for fast turnaround to address the ever-growing list of digital customer touch points and control in delivering an online presence that aligns with the organization's brand and operational goals. It is imperative to understand the level of content authoring features, front-end web development, or administration skills required of these newer systems.

Organizations with simpler web publishing needs will be better served by a traditional WCM system as a headless content management approach requires them to build and maintain the client-side code that renders pages. Ultimately, the greatest benefit organizations will reap from headless and hybrid CMSs is future proofing their investments in both their content delivery systems and their channel delivery systems. In their simplest form, the different CMSs can be characterized as:

» **Hybrid CMS:** Built for marketing and developers to manage multichannel content

» **Headless CMS:** API centric for developers to distribute content to multiple front-end channels

» **Unified CMS:** Design centric for marketers to manage content structure and presentation

IDC believes that while a headless CMS offers more flexibility in design controls and front-end management, the addition of nontechnical user usability to manage back-end content in the context of front-end delivery is the appeal of a hybrid approach. To the extent that Acquia can address the challenges described in this paper, the company has a significant opportunity for success with its hybrid CMS offering.

About the Analyst

Marci Maddox, *Research Director, Digital Experience Strategies*

Marci Maddox is Research Director for IDC's Digital Experience Management Software program, responsible for research related to content and media assets that drive relevant, personalized, and engaging digital experiences research. Marci's core research coverage includes creative tools, web content management systems, customer communications, digital asset management, and video platform solutions.

MESSAGE FROM THE SPONSOR

More About Acquia

Acquia's software and services are built around Drupal to give enterprises the ability to build, operate, and optimize websites, apps, and other digital experiences.

For organizations considering a headless architecture, Drupal gives developers freedom to use the front end tools of their choice to design experiences. Developers may use headless Drupal to provide content for channels as varied as websites, mobile devices, native mobile applications, IoT devices, digital signage, virtual reality programs, chat platforms, CRM systems, fitness trackers, smart watches, and smart speakers, and more.

With Drupal, developers can choose the right frontend tools for each job.

To learn more, visit https://www.acquia.com/products/drupal-cloud/cloud-platform

IDC Custom Solutions

The content in this paper was adapted from existing IDC research published on www.idc.com.