jahia

WHY HEADLESS CMS AND MICROSERVICES ARE SYNONYMOUS WITH HYPER AGILITY IN CONTENT MANAGEMENT

TechTarget Custom Media

A NEXT GENERATION CMS takes the utilization of customer data to the next level in order to personalize user experiences. Headless technology and microservices are already revolutionizing content management.

Abstract

AT A time when mobile and Internet have become ubiquitous in consumers' lives, brands need to offer a high quality digital experience that differentiates themselves and the digital services they offer to their customers or prospects. This issue extends well beyond a few major luxury or high-tech brands, often cited as models in terms of customer experience. Many brands, and not only top-listed companies, have recognized the importance of this issue and are now also seeking to make improvements.

In this quest for the perfect user experience, CMS's have become key, with a new breed of CMS emerging as a direct response to these demands: the Headless CMS. "It differs from traditional Web CMSs in that its user interface is entirely focused on content production," explains Romain Gauthier, Product Owner at Jahia. "Contributors no longer have to worry about the way their content will be displayed. A Headless CMS therefore provides less functionality related to content layout and formatting, but instead focuses on making the content available, since another major feature of the Headless CMS is its API-first approach."

These APIs can typically be called from Single Page Apps, Progressive Web Apps, native mobile applications, traditional websites or even business applications.

AN ARCHITECTURE BUILT FOR HIGH-LEVEL USER EXPERIENCES
ROMAIN GAUTHIER points out that making a clear distinction between content and publishing offers the advantage of being able to fully exploit the capabilities of modern JavaScript frameworks. The current rise of Headless CMS comes with that of frameworks like React.js, Angular or Vue.js - technical solutions that make it possible to design very rich, optimized user experiences that are much more comprehensive than what could previously be done on standard HTML pages. What used to be simple HTML pages on a website has evolved into real applications with an enhanced user interface and sophisticated features. According to Romain, "This new approach makes it possible to create a 'single page,' which is actually a real application built using any modern framework." This versatility is all the more important as a growing number of front-end developers have highly specialized skills on specific development frameworks — an expertise that they promote to agencies and

SPECIAL REPORT

*WHY HEADLESS CMS
AND MICROSERVICES
ARE SYNONYMOUS
WITH HYPER AGILITY
IN CONTENT
MANAGEMENT*

organizations while refraining from specializing in any particular CMS. Being specialized in a particular framework allows them to focus on the customer experience aspect and to create high-performance applications in a particularly short time. "By using a high-performance framework and Jahia's Headless CMS, one of our clients was able to develop a Single Page application in just 8 days, based on 1.5 FTEs only," says Romain. "Headless CMS eliminates the need for developers to be trained on a specific CMS. The CMS itself will only communicate with its APIs, and GraphQL APIs offer the advantage of being self-descriptive which makes them easier to familiarize with."

HEADLESS TECHNOLOGY, A MAJOR STEP FORWARD FOR PERSONALIZED CUSTOMER EXPERIENCE

INTEGRATING THE CMS into business applications is a constant concern for any organization. Adding editorial content to pages in existing applications is crucial. "For example, one of our clients in the banking industry wanted to add highly personalized content to their private customer area. The bank pushes content onto the online banking site, one of its oldest

applications," explains Romain. These integrations are common when you want to push editorial content onto an event app, applications developed within a very short time frame by agencies who let the Headless CMS handle the content part. Jahia's CTO Serge Huber stresses the relevance of the Headless approach in these content integration scenarios involving existing applications. "When there are existing applications, Headless makes it easier to add editorial content without having to make substantial changes. There were other ways to achieve this type of content integration, but Headless CMS is a real step forward because of its flexibility compared to ESI (Edge Side Includes) or portlet technologies, for instance."

With its superior integration capabilities, Headless CMS opens the way for personalization that goes far beyond what traditional CMSs were capable of. Through massive, wide-ranging API calls, it is possible to generate content that is much more personalized because it is now possible to match user behavior data captured by the CMS with data from company applications. "The aggregation of API calls on the CMS platform is one of GraphQL's other strengths," Serge explains. "GraphQL makes it possible to easily extend the CMS's own API with data retrieved from

SPECIAL REPORT

*WHY HEADLESS CMS
AND MICROSERVICES
ARE SYNONYMOUS
WITH HYPER AGILITY
IN CONTENT
MANAGEMENT*

the company's information system (e.g., CDP, CRM, OMS, etc.).” As a software vendor, Jahia has made extensive internal use of this approach through small modules and microservices, which extend the original GraphQL API of its CMS. Serge Huber adds, “This approach simplifies the authentication and data access authorization processes in particular. This is a characteristic feature of GraphQL which does not have any direct equivalent when you only have a CMS based on REST APIs.”

ACHIEVING BALANCE BETWEEN THE ROLE OF OPERATIONAL STAFF AND THE AUTONOMY OF FRONT-END DEVELOPERS GRAPHQL TECHNOLOGY makes it possible to transfer the data integration task to the CMS. The aim is to simplify the developers' work in order to improve lead times, agility and customer experience quality. However, this is not a recipe that can be applied to all applications or companies, as explained by Romain Gauthier. “The most appropriate strategy regarding APIs will really depend on the type of project: in some cases, front-end developers may prefer to aggregate the microservices themselves from their development framework. This may be motivated by concerns about

application performance.” Based on the assumption that the performance of a processing chain depends on its weakest link, aggregating API calls at the application level allows developers to build into their applications the expectation that an API may not return the expected information until after a certain delay. They can then design their applications to hide this waiting time. “On systems subject to very high loads, it may be a good idea to split API calls so as not to create a bottleneck,” says Serge Huber, who adds, “However, GraphQL enables the standardization of API calls even if these calls are sent to separate systems, which simplifies the front-end logic significantly — and fully justifies this choice. In addition, a library like Apollo GraphQL further facilitates this standardization process for developers.”

Finally, organizational considerations are an important aspect to pay attention to when migrating to a next-generation CMS. Front-end developers can focus on developing the user experience and spend less time dealing with the complexity of internal enterprise systems and their APIs. One possible consequence of the autonomy given to developers is a loss of control over business-oriented tasks. Operational staff can no longer create custom pages themselves by producing

SPECIAL REPORT

*WHY HEADLESS CMS
AND MICROSERVICES
ARE SYNONYMOUS
WITH HYPER AGILITY
IN CONTENT
MANAGEMENT*

their own web pages. All they can do is provide their content assets to developers, who are then responsible for creating the container that will showcase this content. "Organizations should be very careful in this trade-off with developers. This is often the reason why our clients pursue both approaches in parallel on the same platform and choose to use a classic CMS approach for certain projects and a Headless CMS approach for ones which require greater agility," concludes Romain Gauthier. Headless will not be the endpoint for the future of CMS, but it is undeniably a key technique for more agility in development and a more sophisticated user experience.

SPECIAL REPORT

*WHY HEADLESS CMS
AND MICROSERVICES
ARE SYNONYMOUS
WITH HYPER AGILITY
IN CONTENT
MANAGEMENT*

TechTarget® Custom Media

FREE RESOURCES FOR TECHNOLOGY PROFESSIONALS
TechTarget publishes targeted technology media that address your need for information and resources for researching products, developing strategy and making cost-effective purchase decisions. Our network of technology-specific websites gives you access to industry experts, independent content and analysis and the Web's largest library of vendor-provided white papers, webcasts, podcasts, videos, virtual trade shows, research reports and more—drawing on the rich R&D resources of technology providers to address market trends, challenges and solutions. Our live events and virtual seminars give you access to vendor neutral, expert commentary and advice on the issues and challenges you face daily.
Our social community IT Knowledge Exchange allows you to share real world information in real time with peers and experts.

WHAT MAKES TECHTARGET UNIQUE?
TechTarget is squarely focused on the enterprise IT space. Our team of editors and network of industry experts provide the richest, most relevant content to technology professionals and management. We leverage the immediacy of the Web, the networking and face-to-face opportunities of events and virtual events, and the ability to interact with peers—all to create compelling and actionable information for enterprise technology professionals across all industries and markets.