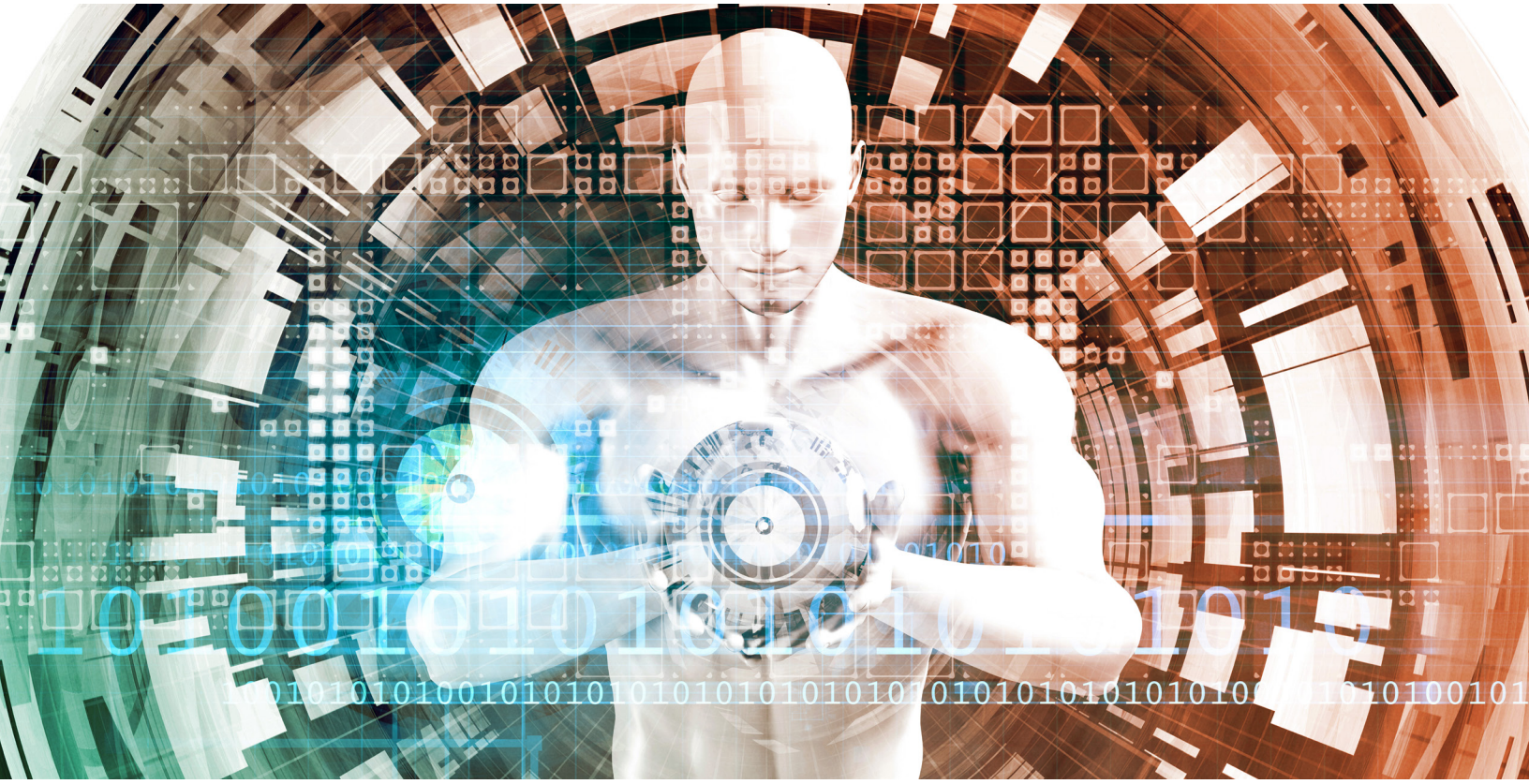
INGENIUX



The New Content Architecture for Modern Web Content Management

Contents

All software goes through an evolution that is dictated by both the needs of its users and advances in technology. The web content management system is a great example of how technology can evolve to help organizations work more efficiently and better meet the demands of customers.

We've talked about the three waves of changes in content management systems that have happened over the past ten or more years. From the start of the Internet and basic brochure-ware style websites to building personalized digital experiences across multiple channels and devices, the CMS has changed how it stores, manages and delivers content. (Read about these waves in more detail by downloading The Third Wave of Content Management)

Technical Challenges to Older CMS Architectures

There were a few important challenges to the traditional technical architectures of content and web experience management systems:

- **Content mixed with HTML:** Older CMS architectures mixed content and HTML (presentation code). Editing of content in this model was difficult. Non-technical writers and editors couldn't change the content because they didn't understand the HTML code that wrapped the content. To make updates, they either employed a web developer or engaged a technical web manager to make the changes. The same was true for changing how the content looked on a web page – developers had to make the changes.
- **Scaling web servers:** With a Web Experience Management suite, scaling the architecture to support higher traffic required adding more front-end web servers, which is typical. However, in some instances both the delivery web server, the database and the backend content management application were housed on the same server, resulting in bigger challenges related to synchronizing content across servers.
- **Database location and security issues:** Older CMS architectures often required the database to reside on the same server as the web server. In architectures where it was separated, it was still connected to the web server in a two-tier architecture model that exposed it to hackers.

A New Architecture for Web Content Management

It's important to look at the CMS as a foundational, or core, component of a customer experience technology framework. For example, an organization may have a website, a customer support portal, a mobile application and a few web applications. A lot of the content within these solutions is the same, just presented differently. The Web CMS is the place where you curate or create and manage this content for delivery to different channels.

Another example involves creating new front-end experiences using newer technologies such as AngularJS, Ember and React. These client-side JavaScript frameworks give organizations the ability to create highly customized, interactive web experiences, including Single Page Applications (SPAs) and mobile applications. In this example, the CMS serves only as the place to curate or create and manage content, the delivery tier is separate.

These examples are only two of many that explain the need for a new WCM content architecture. Others include providing help documentation to a web application, creating a search experience that searches across repositories and content types, and a highly personalized web experience that adapts content according to the visitor.

Elements of a Modern CMS Architecture

Many CMS architecture elements are required to support the new ways customers want to interact with organizations.

Open standards

A CMS built on open standards offers many benefits. First, it increases and improves interoperability with other applications. To create engaging experiences for customers and prospects, you need to bring together information from other systems. Likewise, much of the content in a CMS is delivered to other systems. In both instances, open standards provide a common language upon which these systems can speak to each other and share information.

When a CMS leverages open standards, it is also increasing its longevity. Open standards that are commonly adopted tend to last longer than proprietary standards because many developers and organizations have a vested, shared interest in maintaining and improving the standards over time.

Clear separation of content management from delivery

Another key requirement for today's content management systems is the clear separation of content management from content delivery. Non-technical users (content authors, editors) need a way to create and edit content without having to know code like HTML or scripting.

Just as important, by separating the management of content from its presentation, you can now use that content across many delivery channels.

Multi-channel publishing model

An organization can support the information needs of customers and prospects in a few ways, including digital and print. For example, content can be published to InDesign for printing brochures or datasheets, or it can be published to the brand website, an application or a customer support portal.

In all cases, you can modify the content to suit the publishing environment, changing how much content is published and in what format.

The CMS must support the ability to add new channels as they are needed without major development or customization.

Loosely coupled vs headless vs hybrid

There are different approaches for how a CMS delivers content; these include loosely coupled, headless and hybrid.

Loosely coupled means that the CMS management software is separate from content delivery but provides the option of a content delivery framework or application for rendering dynamic content and other services. A headless model is a CMS that provides only the back-end content management services and an API that other applications can access to retrieve content from the CMS. Any formatting of the presentation of that content takes place on the front-end delivery tier and is not tied to the CMS in any way.

The best of both worlds is the hybrid model, which provides both options. An organization may want their website hosted by the CMS using the loosely coupled model, but also have the headless option to support content delivered to a website hosted somewhere else, a business application or even a customer support portal.

Whether it's a loosely coupled or headless model, or some type of hybrid approach, the CMS must maintain the content separately to support its use across different channels (website, mobile, apps, portals).

Integration of third party content via REST/ API

Along with the ability to deliver CMS managed content to multiple channels, you may also want to integrate content from third parties. Most third-party applications provide a RESTful API that the CMS can use to pull content out of the system.

The option may be to import content directly into the CMS for ongoing management or it may be to connect to the content in its native repository to display it on the website. When the integration doesn't import the content into the CMS, it can be read only, or the API may provide the ability to send updated content back to the third-party system.

Whether content is imported into the CMS or linked, you can include it in searches, mix it with other content in the CMS and display it to other channels (assuming you have the permissions to do so).

A New Content Model

To support this modern content management architecture, the way content is stored and managed must change as well. It's no longer sufficient to think about content in the context of a website or an application. You must approach content from a higher level, above its use in any one channel if you are to create a content model that can be used across multiple publishing channels.

When you define the structure, ask yourself:

- Does the structure help authors create better content?
- Does the structure make the content reusable? Do you need the structure for processing?
- Do you need to filter the content (hide and show different elements of the content depending on the situation)?
- Do you want to display content differently for different channels or audiences?
- Do you need to deliver targeted content?

Once you understand how you need to structure your content, you can create the model in your CMS using taxonomies, schemas and metadata.

Content is intelligent and leverages taxonomies

An intelligent (structured) content model is a way to create and manage content separate from how it is presented in any application or website, in a way that is both human and machine-readable. You store structured content in a format that both defines it using content types and relationships and describes it using metadata. This semantic definition enables the CMS to adapt the content for multiple outputs and formats.

It's important to bring the modeling of content a step up, outside of its delivery to any channel. Take some time to understand all the content your organization creates and manages. Define your organization's taxonomy including content types, their

relationships, and associated metadata. Defining and managing content in this manner ensures that it can be reused across all your channels, both offline and digital. It also supports federated and faceted search.

Your CMS must support the ability to create and deliver structured content. Structured content includes XML, JSON and other formats that provide a rich content definition.

Digital assets are managed content

Most CMS systems manage digital assets such as images, videos, and documents separately. But this content is increasingly important to the digital experience and a new approach is required to ensure it is easily leveraged in UX designs.

In the new content architecture model you can manage your digital assets and their associated metadata without having to go to another digital asset management tool. The same taxonomy and metadata you apply to text-based content is also used for digital assets. This enables you to include digital assets in search queries and to pull them into web pages in the same way you pull in text-based content. To ensure digital assets support multi-channel use, the CMS must also provide features such as renditions, content-aware image resizing, as well as the ability to edit metadata.

Summary

When you think about web content management as a foundational element for your customer experience architecture you see why it's important for the CMS to provide capabilities to enable its use for multiple channels and devices, both digital and print.

A CMS that tightly couples content with presentation will not support a "create once, publish many" content strategy. It will not support the ability to build unique experiences using new web technologies such as SPA and mobile apps.

The new WCM content architecture requires major changes in how a CMS manages and delivers content. It is this new architecture that is set to deliver the next generation of engaging customer experiences.

About Ingeniux

Ingeniux is the leading provider of web content management and digital experience software. We enable organizations to orchestrate the entire customer experience from acquisition through to sales to support and service, across any device, application, or website.

We build content management software with an unparalleled focus on the content itself. The Ingeniux CMS is designed to manage and deliver modern websites, customer support portals, online communities, and other customer touchpoints.

We believe in intelligent "structured" content. We design our software to enable content reuse, enable true mobile and multi-channel content delivery, and insightful content discovery. Our unique content-as-a-service capabilities deliver content into web and mobile applications, and other key channels.

Ingeniux software is available as a fully managed software service or an on-premise application. Ingeniux delivers unparalleled service and support to customers worldwide.

To learn more, visit us at http://www.ingeniux.com.

INGENIUX

PO Box 21466
Seattle, WA 98111

info@ingeniux.com
877 445 8228