Dataweavers

Headless CMS Architecture Map

A practical guide to your day-one architecture

Getting started with headless can feel overwhelming, but you don’t need the full enterprise stack on day one. This map shows the minimum viable architecture for different types of sites, from simple brochureware to complex interactive applications. It highlights what is essential now and what can wait until later. As your requirements grow, you can add API layers and strengthen CI/CD, security, and observability step by step... without over-investing too early.

Understanding the factors in your Architecture Stack: Site Categories

There are many types of headless websites. From an architectural requirements perspective, they fall into the following categories:

Static brochureware — Slow-changing, static content. Example: A small business website with fixed service descriptions and contact details.

Content-led brochureware — A mix of slow-changing evergreen pages and regularly published articles, with no interactivity. Example: A professional services company site featuring detailed pages about products and services alongside a library of thought-leadership articles.

Minimally interactive sites — Limited interactivity such as enquiry forms, registration links, and complaint submissions, combined with informational content. Example: A public sector website providing information on government services with options to submit enquiries, register for programs, or file complaints.

Enterprise content-led — Uses an enterprise-class CMS for governance, compliance and scale (multi-site, multi-language), with limited interactivity and some API integrations for content and basic services rather than full transactional flows. Example: A global corporate website with strict brand governance and multi-region publishing workflows.

Extensively interactive sites — Authenticated experiences where users access personalised data or complete transactions, combined with publicly available product information, product disclosure statements, and supporting content such as articles and industry news. Example: An insurance business platform allowing customers to log in, submit online claims, obtain quotes in real time, and access related product documentation and news.

Line-of-business applications — Purpose-built applications that do not follow CMS patterns, where most processing occurs on the server. Example: An inventory management tool integrated with ERP systems.

Understanding the factors in your Architecture Stack: Layers

Choosing a headless CMS is only part of the journey. To run effectively from day one, you also need to think about the layers that support delivery, security, and operations.

CMS tier	The level of CMS capability required, ranging from none/basic tools through midmarket headless to full enterprise headless platforms. (For more read our article on choosing the right CMS.)
API layers (min)	The minimum set of APIs your stack needs to function, such as BEFE (backend-for-frontend), Experience APIs, or deeper integration/data layers.
Rendering & caching	How your content is delivered and optimised for users, from static site generation (SSG) to selective server-side rendering (SSR) and CDN/edge caching.
CI/CD	The level of automation for builds and deployments, from simple pipelines to enterprise-grade blue/green, canary, and rollback capabilities.
Security essentials	The minimum security controls needed at each stage, covering basics like TLS and WAF through to governance, audit, and advanced data protections.
Observability	How you monitor and measure the health of your digital platform, from simple uptime checks to full telemetry, tracing, and business-level KPIs.
Hosting patterns	The typical hosting approach for this category, including SaaS CMS, in-tenant APIs, front-end hosting with CDN, and in more complex cases, data stores and integration buses. (For more read a comparison of the 3 primary options).

Category	CMS tier	API layers (min)	Rendering & caching	CI/CD	Security essentials	Observability	Hosting patterns
Static brochureware	None or basic	None	SSG, CDN cache	Simple build + deploy	WAF, TLS, basic headers	Static logs, uptime ping	Static site hosting through SaaS or In-tenant + CDN
Content-led brochureware	Basic or mid-tier headless	BEFE	SSG/ISR, image optimisation, CDN	Standard pipelines	WAF, secrets mgmt, least privilege	Page perf metrics, publish webhooks	SaaS headless CMS SaaS or In-tenant front end host + CDN
Minimally interactive sites	Mid-tier or enterprise	BEFE, Experience	SSG + selective SSR/ISR	Zero-downtime deploys	OIDC if needed, input validation, rate limiting	Structured logs, tracing BEFE→Experience	SaaS headless CMS In tenant APIs SaaS or In-tenant front end host + CDN
Enterprise content-led	Enterprise headless	BEFE, Optional Experience	SSG/ISR at scale selective SSR	Zero-downtime, env promotion	Governance, WAF, secrets, role-based authoring	Content ops, KPIs, cache hit ratio, CDN logs	SaaS headless CMS In-tenant APIs SaaS or In-tenant front end host + CDN
Extensively interactive sites	Enterprise headless	BEFE, Experience, Integration/Data	SSR/ISR, edge caching, queue-backed writes, reference storage	Enterprise pipelines with Blue/green, canary, rollback	OIDC, PII controls, audit, WAF, DDoS	Full telemetry, SLOs, synthetics, traces end-to-end and audit trails	SaaS headless CMS In-tenant front end host + CDN In-tenant APIs, message bus, data stores + APIs/EAI, to systems of record
Line-of-business applications	None or Basic CMS	Experience, Integration/Data	SPA/SSR as needed, CDN	Enterprise pipelines with Blue/green, canary, rollback	Fine-grained authZ, secrets, network policies	Deep APM, domain events, audit trails	CMS not mandatory. In-tenant front end host In-tenant APIs, message bus, data stores + APIs/EAI, to systems of record

Dataweavers

What next?

Start by confirming your category today and the next one you are likely to reach. Close any gaps in the “minimum” stack first, then add layers only when requirements demand them.

Revisit the CMS Capability and API Maturity Quadrant, to check that API maturity and CMS capability are advancing together. Keep your non-functional guardrails visible: security, performance, reliability and cost.

For more information refer to our Headless CMS Reference Architecture Article