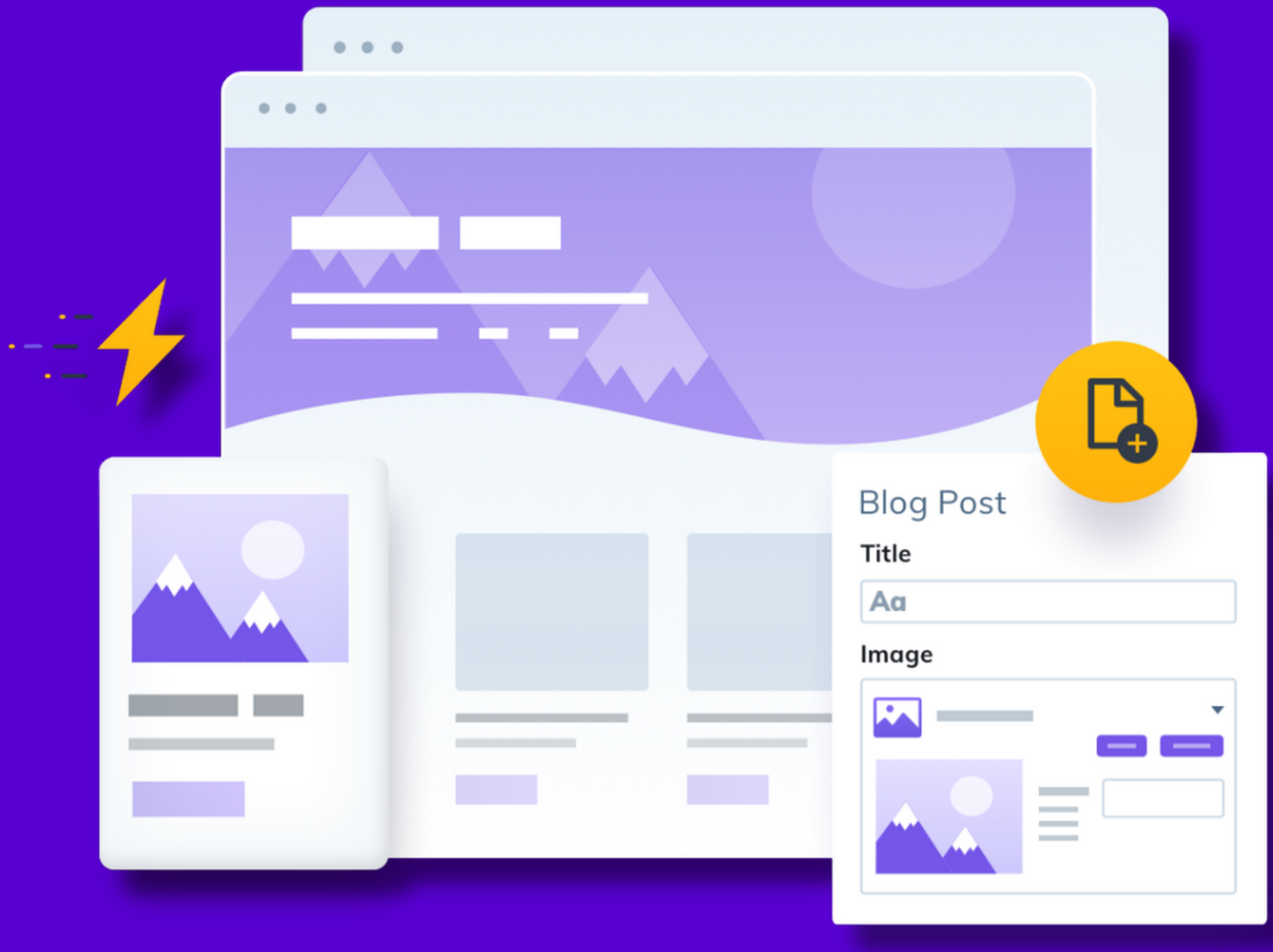


An Introduction to Content Architecture

agility

Welcome to "Introduction to Content Architecture," an ebook designed to provide you with a comprehensive understanding of the architecture behind effective and scalable digital experiences.

In today's digital age, content is king, and a website or application's success largely depends on its content's quality and organization.

A well-designed content architecture is critical to ensuring that all your digital experiences are correctly structured; that means in such a way that they are intuitive and easy to navigate for users who are consuming the content, as well as those composing and moderating content built within those structures.
In this ebook, we will explore the fundamentals of content architecture and the key components, such as content modelling, information architecture, and metadata.

We will provide you with practical guidance on how to plan and implement a content architecture that meets the needs of your organization and users so you’re ready to tackle your next digital endeavour with confidence.

Whether you are a content creator, developer, or project manager, this eBook will provide the tools and knowledge you need to create compelling digital experiences that deliver value to your users. Let's get started on our journey into the world of content architecture!

What is Content Architecture?

Content Architecture refers to how your content is organized, structured, labelled, and interconnected.

Having well-designed architecture is key for two things:

- The content management experience for your content editors
- The experience for the visitors of your digital solution

Your content editors will be able to build, edit, and manage content more intuitively and as a result, your visitors will have a better user experience. This is the best practice for the **Content Before Design** approach, or **Content First**.

Having a smooth and intuitive content architecture maximizes your content editors' experience by giving them the tools to have full control over their content. This makes things like building new pages and editing content easier for your editors and avoids putting them in situations where they might have to hand-code HTML content. All in all, it allows your editors to rely less on developers when a change is needed, leading to faster and more efficient updates.

Solid content architecture is also extremely beneficial when it comes time to expand your digital stack. By future-proofing content ahead of time to ensure it's reusable, flexible, and easy to manage, you can be sure that your composable stack will be easier to enhance and maintain.

That means you don't have to redo your entire website whenever you need to change something or add a new integration to your stack. It should be like adding a new Lego block onto a set.

Content Infrastructure

Content infrastructure is a collection of tools, systems, and processes that manage the creation, organization, storage, and delivery of digital content.

It encompasses everything from content management systems (CMS) and digital asset management (DAM) systems to application programming interfaces (APIs) and content delivery networks (CDNs).

Robust infrastructure ensures your content creators can reliably store, secure, and deliver content to end-users across various digital channels. This involves implementing standardized content formats, metadata structures, and workflows to streamline the content creation process and improve collaboration between content creators.

In addition, content infrastructure provides mechanisms for controlling access to content, such as roles and permissions. Implementing permissions like this makes it easier for team members to focus on what matters most to their job ensuring compliance with regulatory requirements and optimizing content for performance and search engine visibility.

Architecture and Infrastructure: Side by Side Comparison

Since we'll be talking infrastructure and architecture a lot, let's compare them real quick before we carry on to avoid confusion.

Content infrastructure and content architecture are related but distinct concepts. Content architecture refers to how content is organized, structured, and presented to users. This includes elements such as information architecture, content types, content models, and content hierarchy. Content architecture is focused on the user experience and how content is presented and consumed.

Content infrastructure, on the other hand, refers to the underlying technology and processes that enable the creation, management, and delivery of digital content. This includes elements such as content management systems, workflows, APIs, and integrations with other marketing and customer engagement tools. Content infrastructure is focused on the operational aspects of content management and delivery.

While content architecture and content infrastructure are different concepts, they are complementary and interdependent. A well-designed content architecture needs a robust content infrastructure to effectively manage and deliver content. In turn, a well-designed content infrastructure needs a clear architecture to ensure that content is organized, structured, and presented to meet user needs.

For example, a content architecture might define different content types, such as articles, videos, and infographics, and how they relate. A content infrastructure might then provide tools and workflows to create, manage, and deliver these different content types across multiple channels.

Without a clear content architecture, the content infrastructure may become disorganized and difficult to manage. And with a robust content infrastructure, the content architecture may be effectively executed and delivered to users.

Aspect	Content Architecture	Content Infrastructure
Definition	The strategic organization and structuring of content to ensure it is intuitive, user-friendly, and easily navigable.	The underlying systems, technologies, and processes that support content creation, management, and distribution.
Focus	The logical layout of content, including categorization, tagging, and interlinking.	Technical aspects of content management and delivery, such as storage, management, and accessibility across platforms.
Components	Taxonomy, metadata, information hierarchy, and user interfaces.	Content Management System (CMS), Digital Asset Management (DAM), APIs, Content Delivery Networks (CDN), and content workflows.
Role in Content Strategy	Organizing and structuring content for seamless discovery and consumption.	Providing the necessary tools and systems to execute content organization and structure plans.
Interaction with Each Other	Content architecture informs the setup and configuration of content infrastructure components.	Content infrastructure supports the implementation of content architecture and ensures seamless integration across platforms.

How Content Infrastructure Benefits Your Digital Strategy

- Streamlining content creation: A content infrastructure can provide tools and workflows to streamline content creation, enabling teams to produce high-quality content more efficiently. This can include content templates, collaboration tools, content libraries, and automated workflows.

- Organizing content: A content infrastructure can help organize and categorize content, making finding and reusing content across different channels and platforms easier. This can include taxonomy development, content modelling, and metadata tagging.

- Managing content: A content infrastructure can provide tools for content governance, including content approval workflows, version control, and content lifecycle management. This helps ensure that content is accurate, up-to-date, and consistent across channels.

- Delivering content: A content infrastructure can deliver content across multiple channels, such as websites, mobile apps, social media, and email. This can include content management systems, APIs, and integrations with other marketing and customer engagement tools.

Content Infrastructure and Composability

Composability refers to a software design approach that emphasizes modularity, reusability, and flexibility. In a composable architecture, applications are built using smaller, self-contained components (or modules) that can be easily combined, replaced, or modified without affecting the overall system.

These components typically have well-defined interfaces, allowing them to interact with each other in a standardized manner. By leveraging APIs, developers ensure flexibility, modularity, and efficiency in building and maintaining complex applications. Thanks to the composable architecture and the use of APIs, developers can build adaptable digital experiences.

In a composable infrastructure, compute, storage and network resources are treated as services that can be dynamically provisioned, configured, and reallocated on demand to meet the changing needs of applications and workloads.

Composable infrastructure is built on the principles of disaggregation and orchestration:

- Disaggregation: The physical resources, such as computing, storage, and networking, are separated from their traditional siloed configurations and pooled together as shared, flexible resources. Each resource type can be independently scaled and upgraded, which simplifies management and reduces overprovisioning.

- Orchestration: A centralized management platform, often referred to as an "infrastructure manager," oversees the provisioning and configuration of resources. This software layer uses APIs to communicate with the disaggregated resources, allowing administrators to compose and decompose infrastructure resources dynamically based on application requirements.

Content infrastructure is the backbone of every great digital experience. It's like the scaffolding of a high-rise building – you don't necessarily see it, but everything would come crashing down without it. It's the glue that holds all your content together, the engine that drives your website, or even the wizard behind the curtain, making sure your content is delivered seamlessly to your audience.

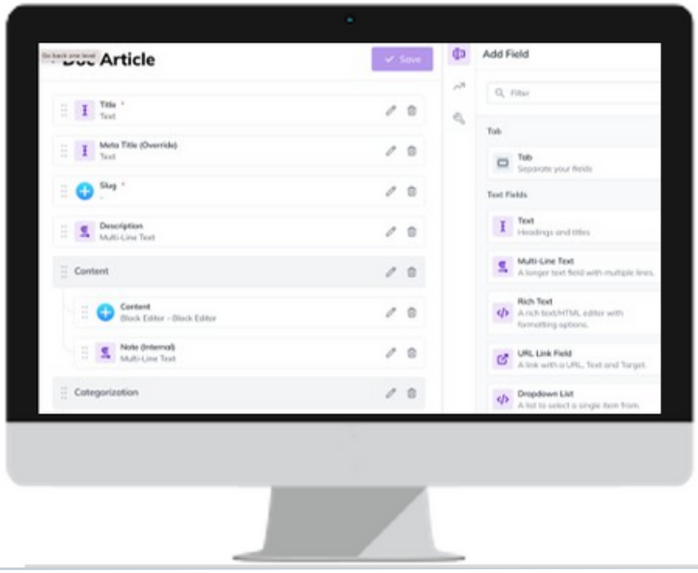


Content Models

Content modeling is the process of defining and structuring content types, attributes, and relationships within a content platform to ensure that content is organized, reusable, and easily maintainable.

A content model is a structured representation of the various types of content. It serves as a blueprint for organizing, creating, and managing content in a consistent, reusable, and easily maintainable manner. Content models play an essential role in content strategy, information architecture, and user experience design.

Content modelling is like assembling a puzzle of your digital content. Only that instead of legos, your blog posts, web pages, white papers, images, and videos are the pieces; content modeling combines them into a neat and organized structure.



agility

A content model typically includes:

- Content types: The different types of content that are required, such as articles, blog posts, videos, images, and so on.
- Fields: The attributes and metadata associated with each content type, such as the title, author, date, description, and tags.
- Templates: Predefined formats for each content type that incorporate the defined attributes and relationships.
- Relationships: The connections and dependencies between different content types, such as how articles are related to blog posts or how images are related to products.

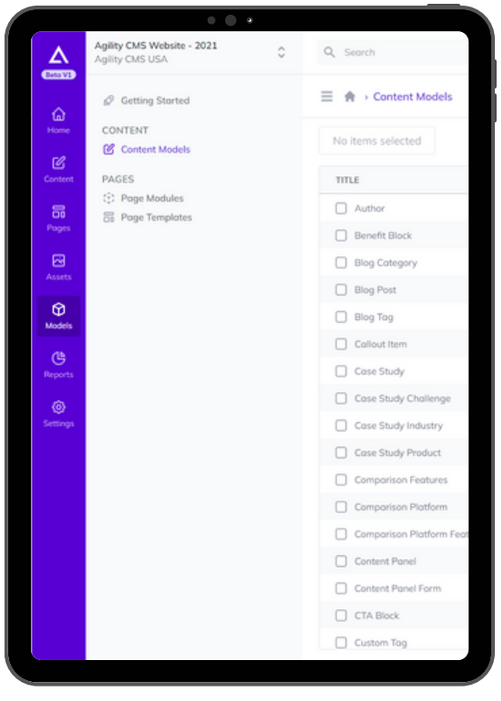
Content models ensure consistency and quality in content creation and management, and they provide a framework for organizing and delivering content across different channels and platforms.

They are often used in content management systems and content platforms to guide content creation and optimization.

With a structured approach to content modeling, your digital presence will expand beyond just words and pictures. It'll be a cohesive masterpiece that tells a story, captures attention, and keeps your audience coming back for more.

Benefits of Content Models

- **Not Just for Developers:** Customize your content models right in the platform using an easy-to-use interface that doesn't require code.
- **Optimize Your Editor's Experience:** Content models improve the editor experience by providing a clear and intuitive framework for content creation. Group fields together, set required fields, default values, and more.
- **Future Proof Your Content:** Content models facilitate future-proofing by providing a flexible and adaptable framework for content creation that can evolve to meet changing needs and requirements.
- **Evolve Your Content Model Over Time:** Seamlessly switch from editing content to your content models to create quick updates.
- **Flexible:** A content model can be re-used for various types of content.



External Content

Agility can easily work with content in external systems.

The are several effective ways to link Agility Content with data from other systems. We normally group these into 3 categories:

- External Linking
- Content Import and Sync
- Content Orchestration

External Linking

You can use an ID from an external system, such as an eCommerce or DAM, or Forms platform, and link that into Agility. Oftentimes a custom field app is used to provide a rich selection UI to allow the editor to choose the external resource, and often the ID is stored with some other metadata as JSON, too.

Content Import and Sync

If you have data in an external system that doesn't allow for workflow and wish to have that shown on your website or in an app, you could run an import to sync that into an Agility list. From there, your content team can editor, moderate, and publish the content to be available on your front-ends.

Content Orchestration

Agility content can also be used as part of a larger orchestration effort as you move data around the different parts of your overall organization. These kinds of network infrastructures are becoming more popular as businesses more tightly integrate their tools to reduce redundancy and improve consistency across their messaging. This is often used with personalization systems and customer data platforms, as well, so to create consistent, personalized messaging and communication across many digital channels, such as websites, email, apps, and instant messaging.

Agility's webhook architecture can be used to drive events and actions in your orchestration layer. This serves as a means of notifying other pieces of the system that Agility content has changed.

Webhook Use Cases

- Clearing and managing custom cache in an external system such as a website, mobile application, or database
- Providing a way to sync content/data for integrations with third-party systems
- Building custom content workflows
- You can kick off a CI/D pipeline to redeploy your website when content has been updated - this is often used for Jamstack sites that are built using static site generators.

Editing Experience

A well-designed content architecture provides content editors with a streamlined and intuitive editing experience to create high-quality content without being burdened by technical or design-related issues.

An effective editing experience should include features such as a user-friendly content platform, easy-to-use editing tools, and customizable workflows. It should also support different content formats, such as text, images, videos, and audio, and provide tools for organizing and categorizing content.

An editor should be able to add a Page Module to a page, and it should render some UI and display content based on the fields on the module or its related content.

As a developer, you define what Page Modules are available in the CMS and what fields they consist of. Each Module defined within Agility should have a corresponding code file within your site or application.

Editor Workflow

Whether you want to integrate your content with other services or automate approvals, workflows empower your team, increase the quality of content, save time, and make you more productive.

Workflows foster collaboration with team members while maintaining the quality of your content.

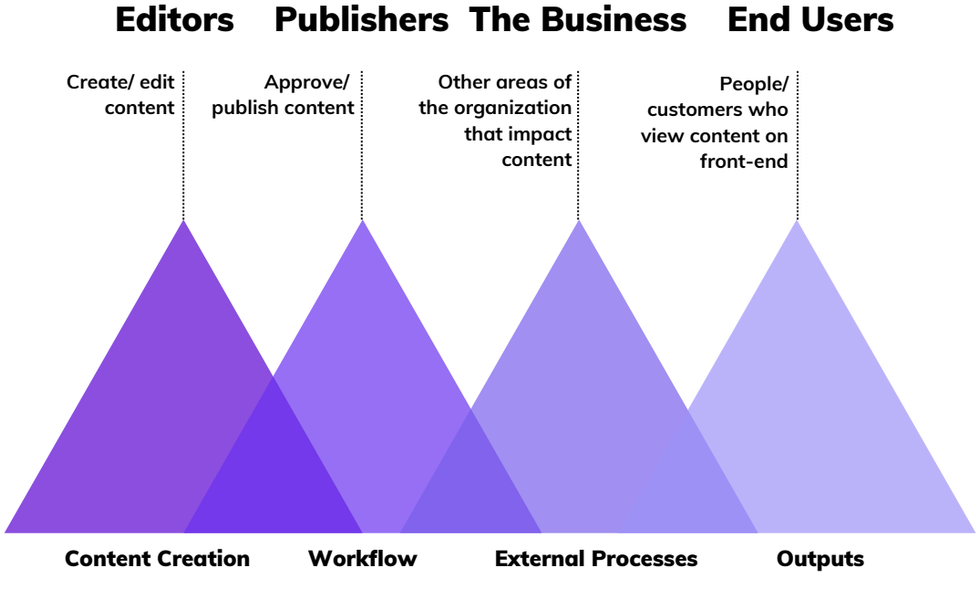
Imagine that you have multiple editors managing content. Often, you'll need to review their work for quality assurance. With workflows, you can customize how your team reviews and publishes content.

An example of a typical content approval workflow looks like this:

- A content editor has set up a new landing page for a campaign on their website.
- Then, when the editor feels the page is ready, they need to notify their approver to review.
- Next, the approver would review the content and decide whether this page should be published or not.
- Lastly, editors publish the content.

With workflows, you can also:

- Send a message to a Slack channel whenever content is updated in your platform
- Post articles to social media channels
- Check content for spelling and grammar errors by using a service such as Grammarly, flag it and send an email to the person that last modified it
- Create a version of the content in another language by using the Google Translate API
- Detect usage of <script> tags in content or pages, flag them for manual review and send an email
- Run automated, browser-based unit tests in preview mode to ensure a content update will not break functionality on a page



Page Modules

Page Modules are reusable, self-contained components or blocks of content within a web page or application. Page Modules are designed to encapsulate specific types of content, functionality, or design elements, making it easier to build, manage, and maintain web pages by assembling these modular components. Where a component is an element of a design system, a Page Module defines the “properties” of those components. It allows editors to place them on pages anywhere in the sitemap.

Page modules can include various elements, such as text blocks, images, videos, call-to-action buttons, forms, or navigation menus. They can be easily added, removed, or rearranged within a page, providing flexibility and consistency in layout and design.

The main benefits of page modules are:

- **Reusability:** Modules can be reused across multiple pages, promoting consistency in design and user experience while reducing the effort required to create new pages or update existing ones.

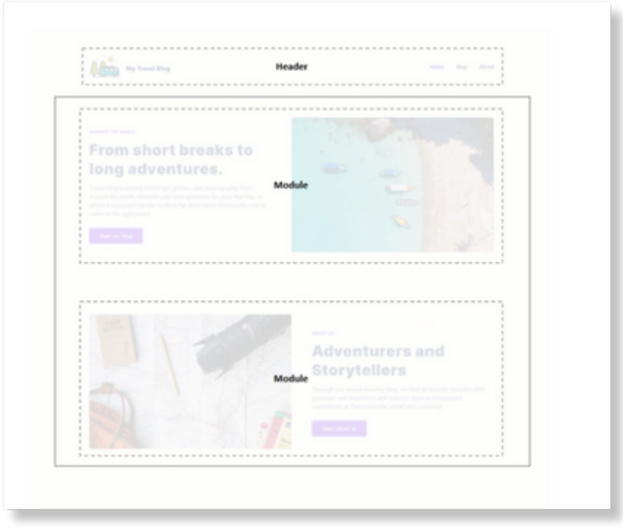
- **Flexibility:** Page modules can be easily rearranged or customized on a per-page basis, allowing for rapid adaptation to changing content or design requirements.

- **Managing Content:** A content infrastructure can provide tools for content governance, including content approval workflows, version control, and content lifecycle management. This helps ensure that content is accurate, up-to-date, and consistent across channels.

- **Scalability**: As modules are designed to be reusable and easily integrated, they support the efficient scaling of a website or application, accommodating growth in content and features.

- **Streamlined Content Management:** Many content platforms support modular page building, enabling content creators and editors to easily add, update, or rearrange page modules without needing in-depth technical knowledge or coding skills.

In a content platform, an editor can add a Module to a page and render content based on the fields on the module. Let’s see an example:



In Agility, Page Modules are the individual functional components that populate a page layout. Developers create a toolbox of modules (often based on their design system) that editors can use to compose what type of content is on each page and in what order they appear by placing them within a Content Zone.

Before editors access and apply Page Modules, developers define which modules are available in the platform and what fields they consist of. Each Module defined within Agility should have a corresponding code file within your site or application.

Content Integrations

Content Platforms, like Agility, provide extensibility layers to allow developers to build custom integrations with any system.

Integration Types

Depending on the type of integration you want for your solution, it may be split into two parts:

- **Platform Integration:** This includes developing custom fields where editors can interact with an external system and executing event-driven business logic.
- **Website/App Integration:** For interfacing with your external system from your website/app directly.

On the other hand, some integrations do not require a CMS integration or may not require any Website/App integration. It all depends on what you are trying to do.



Website/App Integrations

Agility is a Content Platform with Headless capabilities, meaning it does not enforce any rules or restrictions on how you build your website or app. The way your integration works in your solution is entirely up to you.

Use Case: Custom Field for Selecting Database Records
Your developers created a module that displays a listing of featured items that is sourced from an external database. The editor would like to be able to control exactly which of those items are displayed on the website.

A solution to this could be to create a Custom Field Type via JS that renders a Searchlistbox (search based on input and select) that uses AJAX and CORS to return a list of results from that external database. Now editors are actually selecting real records from their database, saving the ID of the record on save and then in the module code, and a database query is executed to retrieve those details.

Use Case: Validating Content

A list of "Products" is maintained with Agility. Each product has an SKU field. The SKU field is used later to map and synchronize data with Microsoft Dynamics. The editor must set a valid SKU when they save a product in Agility.

A solution to this was to validate the SKU on the onBeforeSave event using the Custom Events JS API. A controller action result was set up on the website to accept an SKU parameter and query Dynamics to see if that SKU exists and is active.

Then, in Agility, every time a Product is saved, the onBeforeSave event is called, which executes a JS function. Code was implemented in the function to call the Dynamics SKU validation controller action via AJAX and CORS and await a response. If the SKU was valid, Save was allowed to continue. If it wasn't valid, the save would be cancelled, and the user was prompted to enter a valid SKU.

Why is Content Architecture Important?

Usability

Clear and intuitive navigation is critical for users (editors) to easily move between different pages and sections of a website or app.

Content architecture helps to establish a clear and intuitive navigation structure that guides users. A well-designed content architecture creates a clear and logical hierarchy of information, making it easier for users and search engines to navigate and understand the site's content. Proper use of categories, tags, and internal linking helps search engines crawl and index the site efficiently, improving its SEO.

Content architecture plays a crucial role in the usability of SEO, as it determines how easily search engines can crawl and understand the content on a website. Plus, a well-organized website with a clear hierarchy and meaningful content can lead to higher search rankings and better visibility for the site.

Flexibility

Content architecture allows for flexibility by organizing and structuring content in a way that accommodates future changes and additions, customizes the content to meet user needs, repurposes content for different formats and audiences, and provides clear navigation that allows users to access information in a way that works best for them.

Future-Proofing

As organizations continue to face increasingly hectic schedules, inefficient use of systems can impede our ability to fulfill necessary tasks. In the enterprise world, it’s common to encounter scenarios where agencies or in-house employers build a product or service only for the stakeholders to desire modifications months later. The resulting effort and expense can be substantial when such constructions are developed with a rigid foundation.

Content architecture provides a flexible and scalable framework that can adapt to changing business needs and growing amounts of content.

Doing proper Content Architecture ensures you're not just building for today but also for tomorrow and for changes. For example, hardcoding a dropdown of categories in an article versus building a list of categories and linking to that list allows you to modify that list easily but also call that list for the values to be shown through the API easily.

Content architecture is a critical element of creating digital content that can withstand the test of time. By organizing content logically and systematically, content architecture enables future-proofing. A well-designed content architecture allows for scalability, flexibility, and adaptability, enabling content creators to update, modify or add new content without disrupting the structure of the entire system.

This ability to make changes and updates seamlessly ensures that content remains relevant and up-to-date for longer periods, reducing the need for frequent overhauls or complete redesigns. Additionally, effective content architecture ensures that content is easy to navigate and understand, which is particularly important as new users access the content over time. By providing a stable foundation for the content, content architecture enables future-proofing, ensuring that the content remains usable and effective over time.

How do You Get Started?

Step One: Understand

The first step in content architecture is understanding your existing content. This involves conducting a content audit to identify what types of content you have, where it's currently stored, and how it's being used.

You can also review analytics data to see how users engage with your content and identify any gaps or opportunities for improvement. It's also important to talk to stakeholders, such as content creators, editors, and marketers, to get their perspectives on what's working and what's not.

Step Two: Define

Start by defining a logical hierarchy and organization of content and the relationships between different pieces of content. This includes considerations such as:

- Information architecture: Create a clear and intuitive navigation structure for the website or application, allowing users to find and access content easily.

- Content types and templates: Define the different types of content that will be included on the site or app, as well as the templates or layouts that will be used to present that content.

Metadata and tagging: Establishing a system for metadata and tagging, which provides additional information about the content and makes your content easily searchable and discoverable.

Step Three: Plan

Map out how your content will be organized, structured, and presented across different channels and touchpoints.

For example, you might:

- Develop a content taxonomy: Develop a system for categorizing and organizing your content by topic, audience, or content type. This will help users find and navigate your content more easily.
- Identify your target audience: Identity who your target audience is, their needs, and how they will be using your website or application. This will help you determine what content you need to create and how to organize it in a user-friendly and relevant way for your audience.
- Map out the different touchpoints where users will interact with your content: Create a map of where your visitors start their journey and when they finish. Plan how your content will be structured and optimized for each touchpoint.

Step Four: Implement

With your plan in place, you can begin to implement your content architecture. This involves selecting and configuring the tools and systems you'll need to manage your content, such as a Content Platform, digital asset management (DAM) system, or content delivery network (CDN).

You'll also need to

- Define workflows: Establish clear editorial guidelines to manage your content effectively.
- Develop policies and procedures for managing and updating the content: Include how new content will be added, how old content will be archived or removed, and who will be responsible for managing the content.

Step Five: Make it composable

- **Define the building blocks:** Identify the building blocks of your website or app, such as modules, components, and services. These building blocks should be designed to be reusable, composable, and independent of one another.

- **Create a modular design:** Develop a modular design for your software system, where each building block can be composed to create larger systems and applications.

- **Use standard protocols and formats:** Use standard protocols and formats for communication between building blocks, such as RESTful APIs, JSON, and XML. This ensures that different building blocks can communicate with each other in a consistent and standardized way.

- **Implement separation of concerns:** Ensure each building block is responsible for a single function or feature. This makes testing, updates, and maintenance easier.

- **Implement versioning and dependency management:** To ensure that different building blocks can evolve independently, implement versioning and dependency management. This involves managing dependencies between building blocks and ensuring that each building block uses the correct versions of its dependencies.

- **Use automation and testing:** To ensure that building blocks function correctly and can be easily integrated into larger systems. This involves implementing automated testing, continuous integration, and deployment processes to ensure that changes to building blocks can be quickly tested and deployed.

Step Six: Monitor and iterate

Finally, it's important to monitor and iterate your content architecture over time continually. This involves reviewing analytics data, conducting user research, and gathering stakeholder feedback.

For example, you might:

- Review website analytics: Use website analytics tools to monitor user behaviour and engagement, such as page views, bounce rates, and time on the page.

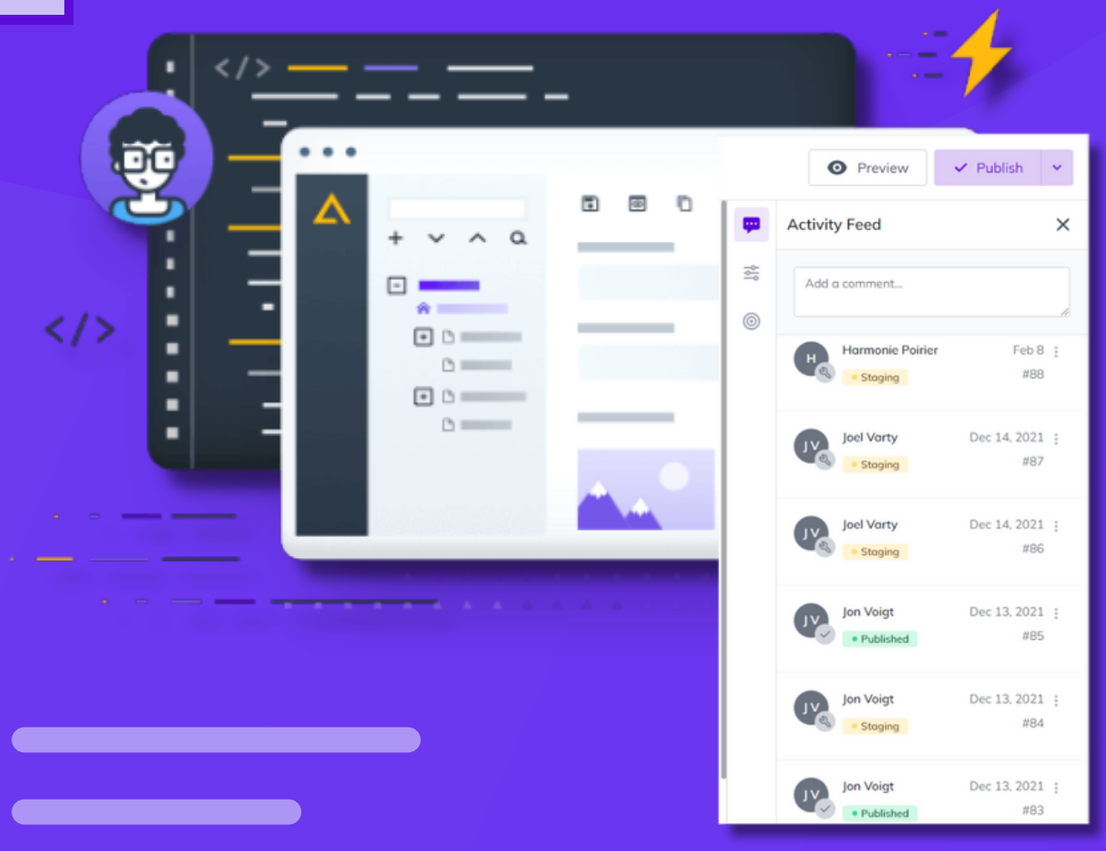
- Monitor user feedback: Solicit user feedback through surveys, focus groups, or other methods to determine whether they find the website or application easy to navigate.

- Review search queries: Monitor search queries on the website or application to identify common search terms and whether the search functionality effectively returns relevant results.

- Monitor content updates: Monitor updates to the website or application content to ensure that new content is being added in a way that is consistent with the established content architecture.

Supercharge Your Content Architecture With Agility

Get in Touch >



agility