

# Content modeling and Content Management on the JAMstack



February 4th 2020



# Hello!



Amelia Winger-Bearskin  
Developer Evangelist



**Amelia:** I'm Amelia Winger-Bearskin, Developer Evangelist at Contentful.

## Dream Stack



3



[Gatsby](#) (powered by GraphQL): Blazing fast modern site generator for React.

[Contentful](#): The content layer for modern digital experiences

Stop using old tools to solve a modern problem: delivering great digital experiences. Web CMSes create digital sprawl — content silos that slow innovation. But a unified content layer enables your team to scale and iterate faster. See ya, CMS.

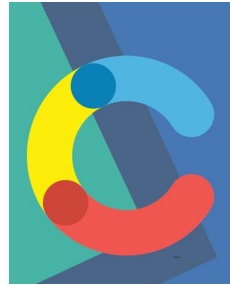
[Netlify](#): One workflow.

From local development  
to global deployment.

Netlify is everything you need to build fast, modern websites: continuous deployment, serverless functions, and so much more.

# Agenda

1. Content Strategy
2. Content Modeling and Management
3. How to align a Content Model that amplifies the benefits of a modern JAMStack architecture.



# What is Content Strategy?

it is an enterprise solution that:



- ⬡ **DEFINES** how content aligns to business goals.
- ⬡ **GUIDES** decisions about content through the delivery lifecycle.
- ⬡ **SETS** metrics & KPIs to help determine content performance and value.
- ⬡ **INFORMS** content platform development and user experience design.



A good content strategy answers any and all content usage questions such as:

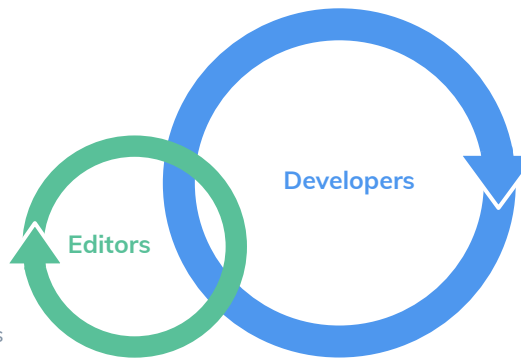
- Why should the content be published?
- Where are we going to publish this content?
- When should we roll out this content to viewers?
- Who do we want to see it?
- What reaction are we hoping to receive from the content?
- What in the world do we do with the content after we've published it?

# Digital teams (devs and editors) want to move fast

Both with and without each other

## Create & Manage Content

- Create landing pages
- Localize a microsite
- Launch a campaign
- Run A/B tests
- Update homepage promos
- Tweak signup page copy



## Build & Deliver Features

- Launch a new app
- Integrate with a DAM
- Iterate a feature
- Update a design system
- Update a content model
- Test/QA new features
- Import legacy content
- Ship bug fixes



Digital capital—are the resources to developing new products, services or experiences for the digital economy.

Whenever a product, service or process is captured in software and digitized, they become *digital capital* and the economics of *abundance* take over.

- Building digital products (like a website) takes cross functional teams:.
- Content marketers creating & managing content; they build campaigns. Optimize conversion on landing pages; are mindful of SEO in the content and order of their pages.

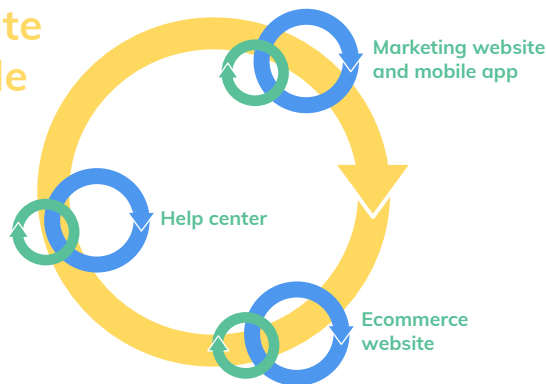
Developers, like us --focus on building & delivering with content models that meet the needs of editors and use integrations and diverse stacks to work flawlessly for cross-functional teams.

Fact of life: Teams have different priorities and pressures -  
Communication is key.

# Content platforms create agility at scale

Reusable architecture & processes allow teams to move fast across an organization

Operate  
& Scale



7



A:

- For teams to scale with reusable content ---**workflows** need to happen in parallel.
- To do this effectively, developers need to build content models that are optimized for those writing and updating the content - the marketers, editors, copywriters, whoever is non technical but needs to update the website as well with content.
- We should be doing what we do best - delivering amazing solutions with code, so that our marketing/editorial teams can do what they do best, getting our company's story out into the world in the most effective way possible.
- in order to build an elegant solution you'll need empathy. [Feel the pains of your marketing counterpart! It will help with content modeling, I promise.](#)
- And listening building and iterating helps you move faster!



The goal of content modeling is to  
redefine obscure content into  
manageable and measurable business  
assets in order to drive growth &  
reduces costs.

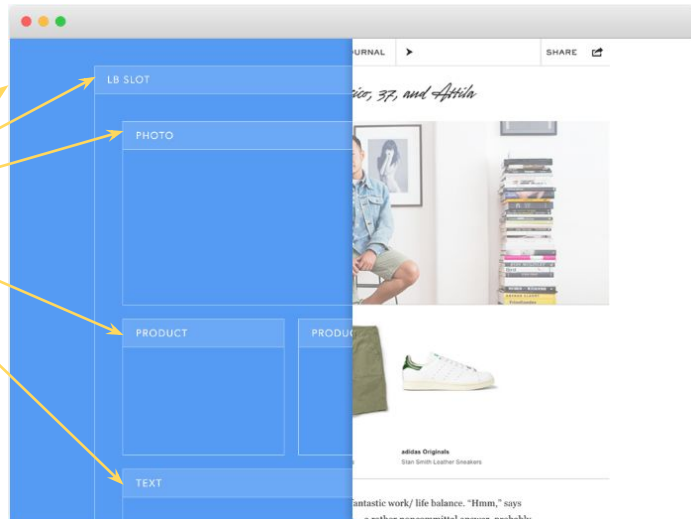


Create once publish everywhere!



# Reusable content means “structured” content

Content Types



9



Content Modeling: you have to think about content a bit differently than you may have in the past.

A web page is made out of building blocks. Breaking a “blob” of content into smaller components is what makes it possible to reuse and share content easily. Contentful calls these building blocks “**content types**.”

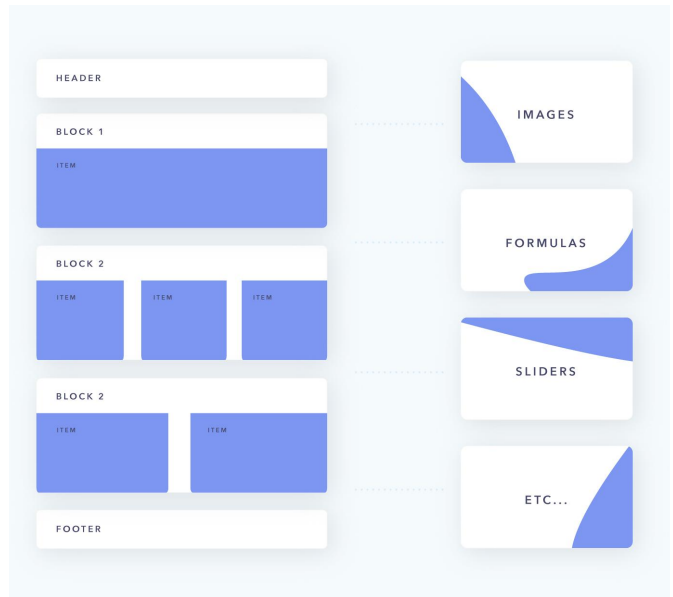
Just like structured data, Content modeling is helping you structure content. Identify chunks of content that could be used in multiple places - store it in one structured infrastructure, it is easier to deploy it again and again across different apps and outputs (like big screens, in store screens or pocket sizes screens).

It is also important to identify reusable visual components so that the code for displaying that visual representation (think: left aligned, or hero image) is maintained in one place.

Model once and publish everywhere is a common phrase, however sometimes you don't get it right the first time, which is why practice makes perfect! Sometimes that practice is also learning about automating changes as well (with migration scripts!).

# Topics and Assemblies

More information [here](#)



10



- There are two primary objectives for using structured content. One is to identify chunks of content that could be reused in multiple places so that the content only has to be maintained in one place. The other is to identify reusable visual components so that the code for displaying that visual representation can be maintained in one place.
- Content types intended primarily to control how content is displayed as called: assemblies, modules or components.

<https://www.contentful.com/r/knowledgebase/topics-and-assemblies/>

# Topics and Assemblies

More information [here](#)

Image Carousel



11



- One common example of an assembly is an image carousel. If you create an image carousel assembly, content authors can create multiple instances of it using different photos for each instance and your front-end code can render the carousel appropriately for each device or context. Note that assemblies are frequently nested. Assemblies can contain other assemblies, topics, and reference fields.
- Typically, one of the first steps in content modeling is identifying reusable content and assemblies. There are two strategies for this:
- Modeling content by focusing on the relationships inherent in the content itself
- Starting with a display mockup and identifying the content model required to build that display (while also thinking ahead to future needs)
- Either of these approaches can lead to a good content model, depending on the skills and preferences of those involved. In practice, content modeling often involves bouncing back and forth between these two perspectives.



# Colorful Demo 2020



That is really where JAMstack and API first CMS strategy comes in, you have separated the content infrastructure / content model from how it will be delivered via your code, so that your editors can say ok i'm uploading these images of products and entering my micro copy and that doesn't really have to determine how the images are delivered since that is part of the delivery strategy for the development team to decide and make accommodations for instance with a series of images that will change based on mobile view vs web view.

# Content Model



- A **content model** is a representation of content by a set of structured content types and relationships.
- Derived by a **cross-functional team** of content leads and technical stakeholders.
- Informed by the **content strategy** which encapsulates both business goals & technical requirements.



## Structured Content

How do we capture and organize our content so that it is meaningful, searchable, reusable? Structure!

- **Structured Content** Adheres to a predefined model with discrete fields and relationships. Easy to analyze, index or search.
- **Unstructured Content** Does not have a predefined model or organization. Ambiguous, difficult to analyze or collect.



- A Content Strategy is an essential first step that helps to define the process and purpose for content in an enterprise.
- How to align a Content Model that amplifies the benefits of a modern JAMStack architecture.

## Structured Content

- **Elements of a Structured Content type:**
  - Attributes: *properties can have validations*
  - Relationships: *one-to-one or one-to-many*
  -
- **Considerations for defining Structured content:**
  - How will we organize content for search and SEO?
  - What tags are most intuitive for users?
  - How will we categorize content for efficient management?
  - How will we structure our content for future reuse?
  - What are the requirements for personalization?

16



## Core Concept: Structured Content

- ⬡ **Elements of a Structured Content type:**
  - Attributes: *properties can have validations*
  - Relationships: *one-to-one or one-to-many*
- ⬡ **Considerations for defining Structured content:**
  - How will we organize content for browse-and-find?
  - What tags are most intuitive for users?
  - How will we categorize content for efficient management?
  - How will we structure our content for future reuse?
  - What are the requirements for personalization, dynamic delivery?



## Modern Architecture > Modern Content Model

“A system is only as fast as its slowest component.”

- JAMStack applications are by definition, Serverless, Dynamic and API-first.
- **Client-Side Logic** is used to dynamically render views (i.e. React, Next.js) implemented as decoupled and distributed modules.
- **Microservices & Serverless Functions** are where the main business logic is stored and invoked, highly specialized and orchestrated.
- **APIs Access Patterns** such as RESTful vs GraphQL can be used to retrieve or modify data resources.

17



JAMStack applications are by definition, Serverless, Dynamic and API-first.

- ⬡ **Client-Side Logic** is used to dynamically render views (i.e. React, Next.js) implemented as decoupled and distributed modules.
- ⬡ **Microservices & Serverless Functions** are where the main business logic is stored and invoked, highly specialized and orchestrated.
- ⬡ **APIs Access Patterns** such as RESTful vs GraphQL can be used to retrieve or modify data resources.



## JAMstack

- To fully realize the benefits of JAMStack we want to design a content model that adheres to the same set of quality attributes:
- **Flexible:** Able to pre-build and render app views dynamically.
- **Interplatform:** Able to support to multi-channel / multi-platform delivery.
- **Reusable:** Decoupled objects allow for reusable modules across projects.
- **Scalable:** API-first allows for easy scale via CDNs / caching strategies.
- **MUTUAL GOAL: Future-Proof Design**

19



To fully realize the benefits of JAMStack we want to design a content model that adheres to the same set of quality attributes:

- ⬡ **Flexible:** Able to pre-build and render app views dynamically.
- ⬡ **Interplatform:** Able to support to multi-channel / multi-platform delivery.
- ⬡ **Reusable:** Decoupled objects allow for reusable modules across projects.
- ⬡ **Scalable:** API-first allows for easy scale via CDNs / caching strategies.
- ⬡ **MUTUAL GOAL: Future-Proof Design**



## Domain Driven Design

How to apply Domain Driven Design for domain model & content model.

1. Identify domain experts and other business stakeholders to participate in domain driven design process.
2. Identify the entities that are core to your business “domain”.
3. Outline the properties or attributes that are required for meaningful representation of those entities.
4. Define the relationships between entities.



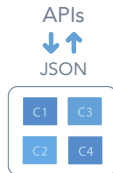
## Core Concept: Domain Driven Design

How to apply Domain Driven Design for domain model & content model.

1. Identify domain experts and other business stakeholders to participate in domain driven design process.
2. Identify real entities that are core to your business “domain”.
3. Outline the properties or attributes that are required for meaningful representation of the entities in the domain.
4. Define relationships between entities.

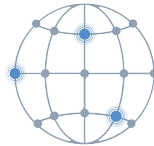
# JAMstack Content Modeling goals!

## Structured content



Reuse and repurpose content, independent of channel or language

## Decoupled architecture



Composable stack, connected to other platforms, at cloud scale

## Agile workflows



Integrate with digital delivery workflows for launch & optimization

22



## Structured content

- You get a lot more flexibility by creating your content as components, so it can be reused and repurposed across products, channels, and teams
- It's flexible enough for existing apps & and future-proof for those yet to come
- It's context-agnostic, independent from mediums, platforms, programming languages, etc.
- It's also not tied to opinionated data models like a traditional CMS
- **For example, The British Museum powers their interactive websites, on-site displays, and audio guides, all from a single content hub**

## Decoupled architecture

- The content is decoupled from the presentation layer, so your team can build unique customer experiences
- By being born in the cloud, it was built from day one to connect with other platforms and services
- It's composable, allowing you to use only what you need, and adopt new features as you grow
- Scales confidently on an enterprise-tested platform
- **For example, TELUS's Marketing team creates and manages both static content and personalized variations in the same place, integrated with customer data.**

## Agile workflows

- It was born agile, purpose-built to integrate with software delivery pipelines, developer environments, and automation
- It changes with your software and workflows
- It lets you iterate and experiment fast, inside your software delivery pipeline
- **For example, TUI can launch new digital touchpoints 5x faster with their new**

- **architecture**

# DreamStacks Meetup

January 20th

Brunch at Algolia

Next Tuesday 2-11-2020

Webinar w/ Netlify



Algolia and Contentful are gathering to bring you a new kind of events on the developer scene: DreamStacks! This event series will include an immersive set of experiences including healthy brunch menus, right-brain workshops, left-brain panel discussions, and of course quality networking. You'll also be able to create your own smoothie and eat some delicious pancakes.

The very first edition of DreamStacks will take place on Thursday, Feb 20 from 10am-12pm and we would be happy to see you there.



# Thank you!

Please make sure you put your name in the raffle box!



Extra slides for  
questions:



# Basic JAMstack CMS Architecture

Developers work in code...



Authors work in the web app...



All content goes to  
the same place...



...and is delivered  
to any platform!



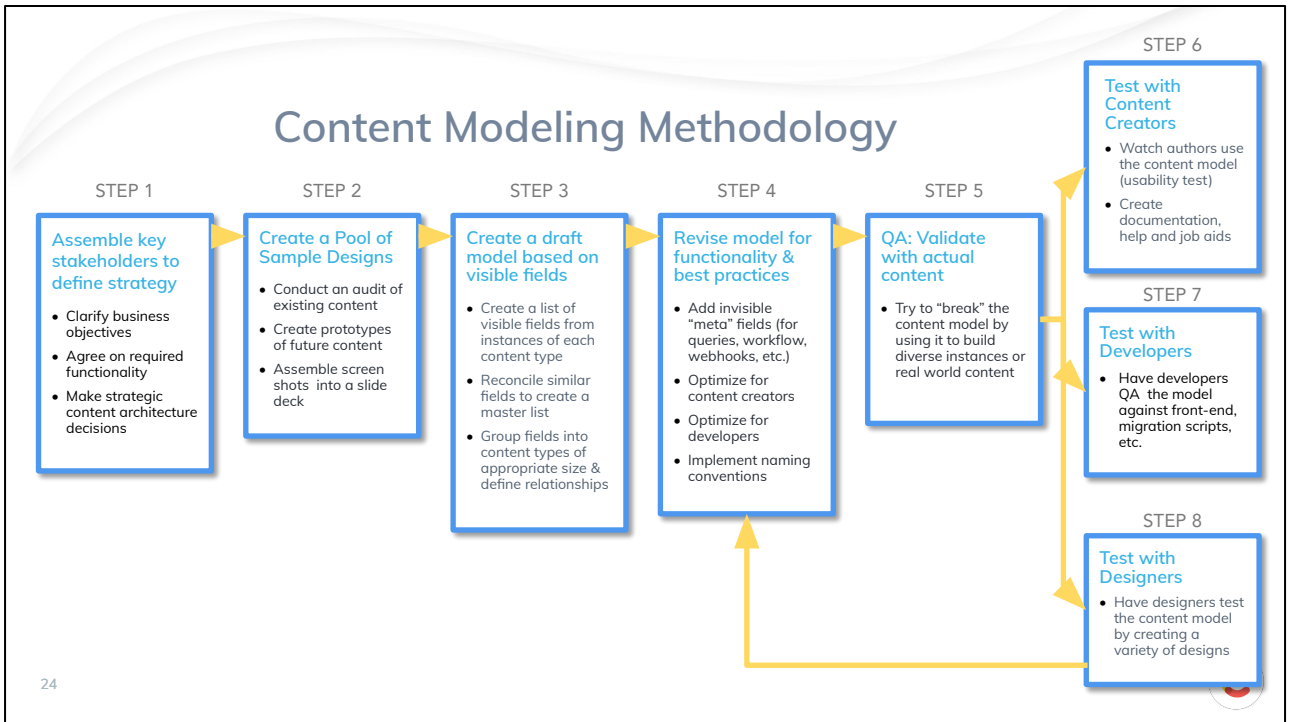
A: content infrastructure provides structured content that can be reusable - enabling devs and editors to work in parallel and move faster - this diagram gives a zoomed out, simplified view of how a CMS can make this possible.

Basic Contentful Architecture

Developers work in code

Authors work in the CMS Web app

all content goes to the same place, and is deployed where the user is and meets them where your story needs to be taken. It could be a website, mobile app, smart watch, voice activated navigation in a smart home, a jumbotron!



A:

This is a slide we have developed to help your team's strategy with Content Modeling.

I like to do this as a workshop to start this process, and like anything, practice makes perfect!

It is important to perfect your practice and make it work for you! A good planning strategy around your content will help your team, but so will a flexible and open attitude.

The short story: feedback is key; important for devs not to make assumptions on behalf of your content marketers. Test, revise, validate, test again! ----