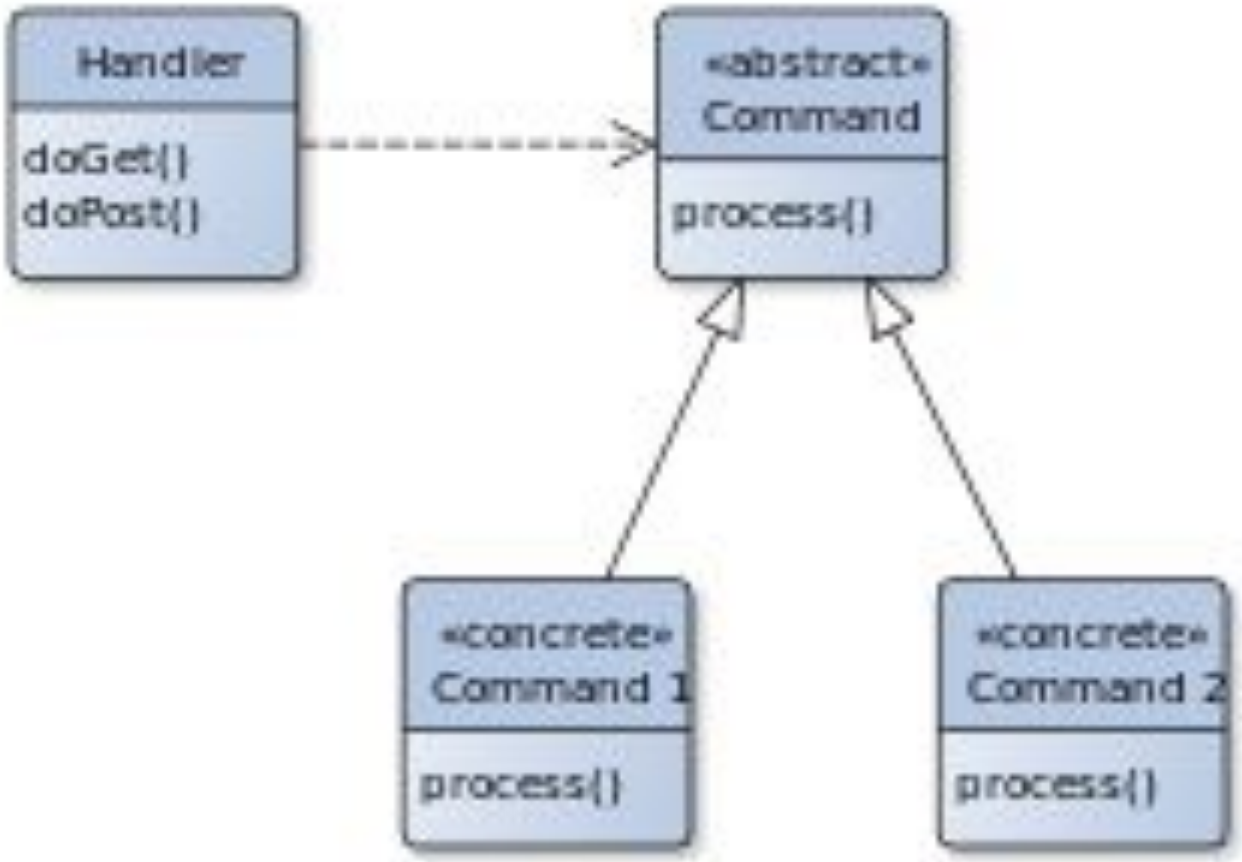
Front Controller Pattern

UNIT III

Front Controller

- A front controller is a software design pattern that handles all requests for a website.
- It is a useful structure for web application developers to achieve flexibility and reuse without code redundancy.



Front Controller

- Front controllers are often used in web applications to implement workflows.
- While not strictly required, it is much easier to control navigation across a set of related pages (for instance, multiple pages used in an online purchase) from a front controller than it is to assign individual pages responsibility for navigation.
- Front controller may be implemented as a Java object or as a script in a scripting language such as PHP, Raku, Python or Ruby
- It is called for every request of a web session
- The script handles the tasks that are common to the application or the framework – session handling, caching and input filtering
- Based on the specific request, it would then instantiate further objects and call methods to handle the required tasks.

Alternate for front controller

- The alternative to a front controller is the usage of page controllers mapped to each site page or path. Although this may cause each individual controller to contain duplicate code, the page-controller approach delivers a high degree of specialization.

Application frameworks that implement the front controller pattern

- Apache Struts
- ASP.NET MVC
- Cairngorm framework in Adobe Flex
- Bailador framework in Raku
- Drupal
- MVC framework in PHP
- Spring framework
- Yesod in Haskell

Components of FCs

- Front controllers is divided into three components:
  - XML mapping: files that map requests to the class that will handle the request processing.
  - Request processor: used for request processing and modifying or retrieving the appropriate model.
  - Flow manager: determines what will be shown on the next page.

Participants

- Controller
- Dispatcher
- Helper
- View

Controller

- The controller is an entrance for users to handle requests in the system.
- It realizes authentication by playing the role of delegating helper or initiating contact retrieval.

Dispatcher

- Dispatchers can be used for navigation and managing the view output.
- Users will receive the next view that is determined by the dispatcher.
- Dispatchers are also flexible; they can be encapsulated within the controller directly or separated into another component.
- The dispatcher provides a static view along with the dynamic mechanism

Helper

- Helpers assist in the processing of views or controllers.
- On the view side, the helper collects data and sometimes stores data as an intermediate station.
- Helpers do certain preprocess such as formatting of the data to web content or providing direct access to the raw data.
- Multiple helpers can collaborate with one view for most conditions.
- Additionally, a helper works as a transformer that adapts and converts the model into a suitable format.

View

- With the collaboration of helpers, views display information to the client by processing data from a model.
- The view will display if the processing succeeds, and vice versa

Java Implementation - FC

```
1  private void doProcess(HttpServletRequest request,
2                         HttpServletResponse response)
3      throws IOException, ServletException {
4      ...
5      try {
6          getRequestProcessor().processRequest(request);
7          getScreenFlowManager().forwardToNextScreen(request, response);
8      } catch (Throwable ex) {
9          String className = ex.getClass().getName();
10         nextScreen = getScreenFlowManager().getExceptionScreen(ex);
11         // Put the exception in the request
12         request.setAttribute("javax.servlet.jsp.jspException", ex);
13         if (nextScreen == null) {
14             // Send to general error screen
15             ex.printStackTrace();
16             throw new ServletException("MainServlet: unknown exception: "
   +
17                 className);
18         }
19     }
```

Benefits of FC

- **Centralized control**
  - The front controller handles all the requests to the web application.
  - This implementation of centralized control that avoids using multiple controllers is desirable for enforcing application-wide policies such as user tracking and security.
- **Thread safety**
  - A new command object arises when receiving a new request, and the command objects are not meant to be thread-safe.
  - Thus, it will be safe in the command classes.
  - Though safety is not guaranteed when threading issues are gathered, code that interacts with commands is still thread-safe.
- **Configurability**
  - As only one front controller is employed in a web application, the application configuration may be greatly simplified.
  - Because the handler shares the responsibility of dispatching, new commands may be added without changes needed to the code.

Drawback of FC

- The front controller pattern may incur performance issues because the single controller is performing a great deal of work, and handlers may introduce bottlenecks if they involve database or document queries.
- The front controller approach is also more complex than that of page controllers.

Relationship with MVC (Model-View-Controller)

- In order to improve system reliability and maintainability, duplicate code should be avoided and centralized when it involves common logic used throughout the system.

- The data for the application is best handled in a single location, obviating the need for duplicate data-retrieval code.

- Different roles in the MVC pattern should be separated to increase testability, which is also true for the controller part in the MVC pattern.

Features	Page Controller	Front Controller
Base Class	A base class is needed and will grow simultaneously with the development of the application.	The centralization of requests is easier to modify than is a base class.
Security	Low security because various objects react differently without consistency.	High, because the controller is implemented in a coordinated fashion.
Logical Page	Single object on each logical page.	Only one controller handles all requests.
Complexity	Low	High

CONTENT MODELING

What is content model?

- A content model documents all the different kinds of content you have on your website.
- It breaks content types down into **their component parts**, describes them in detail, and maps out how they relate to one another.
- A content model is an important step in working out the finer details and practicalities of how you write and manage your content, and how you will present it on the page.

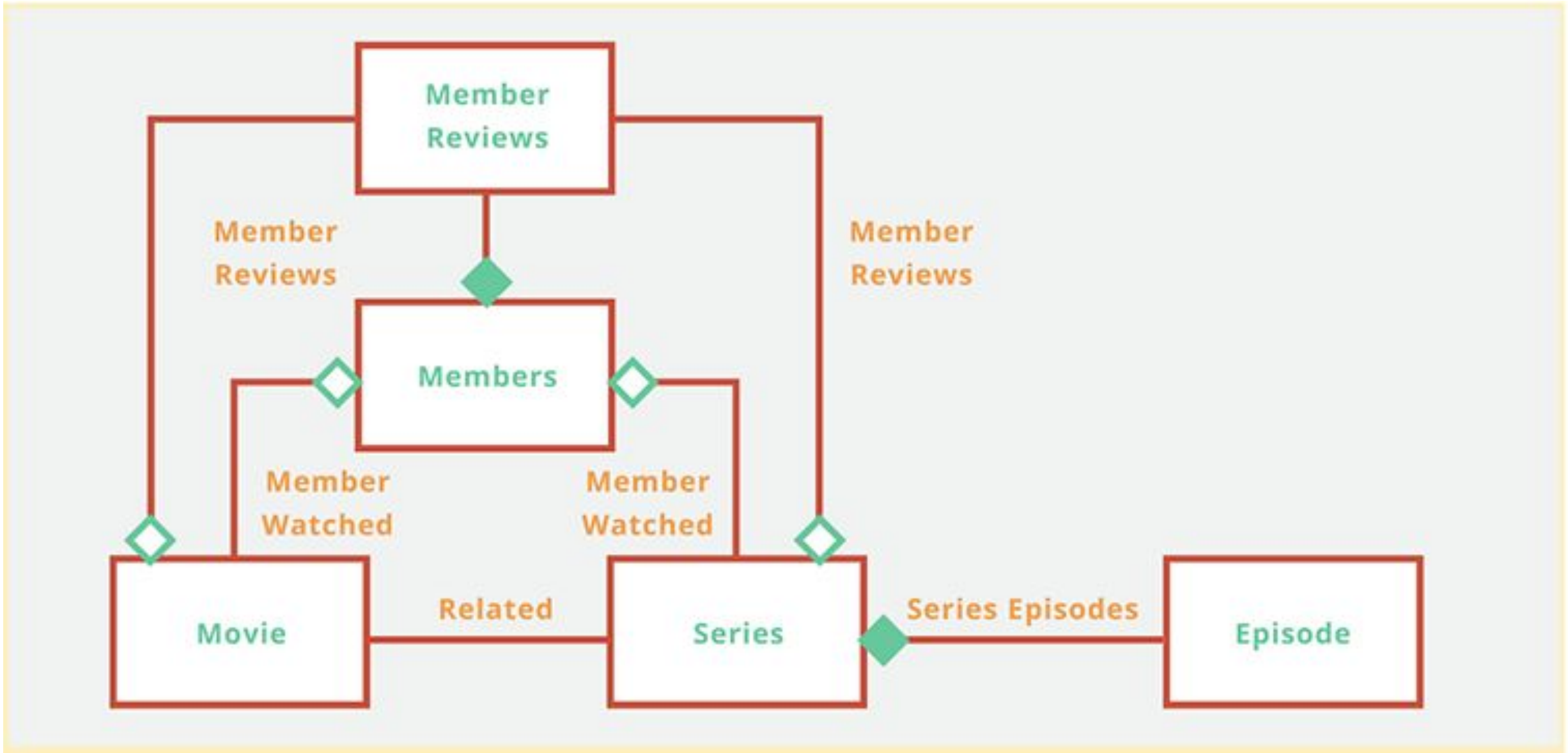
How to create a content model?

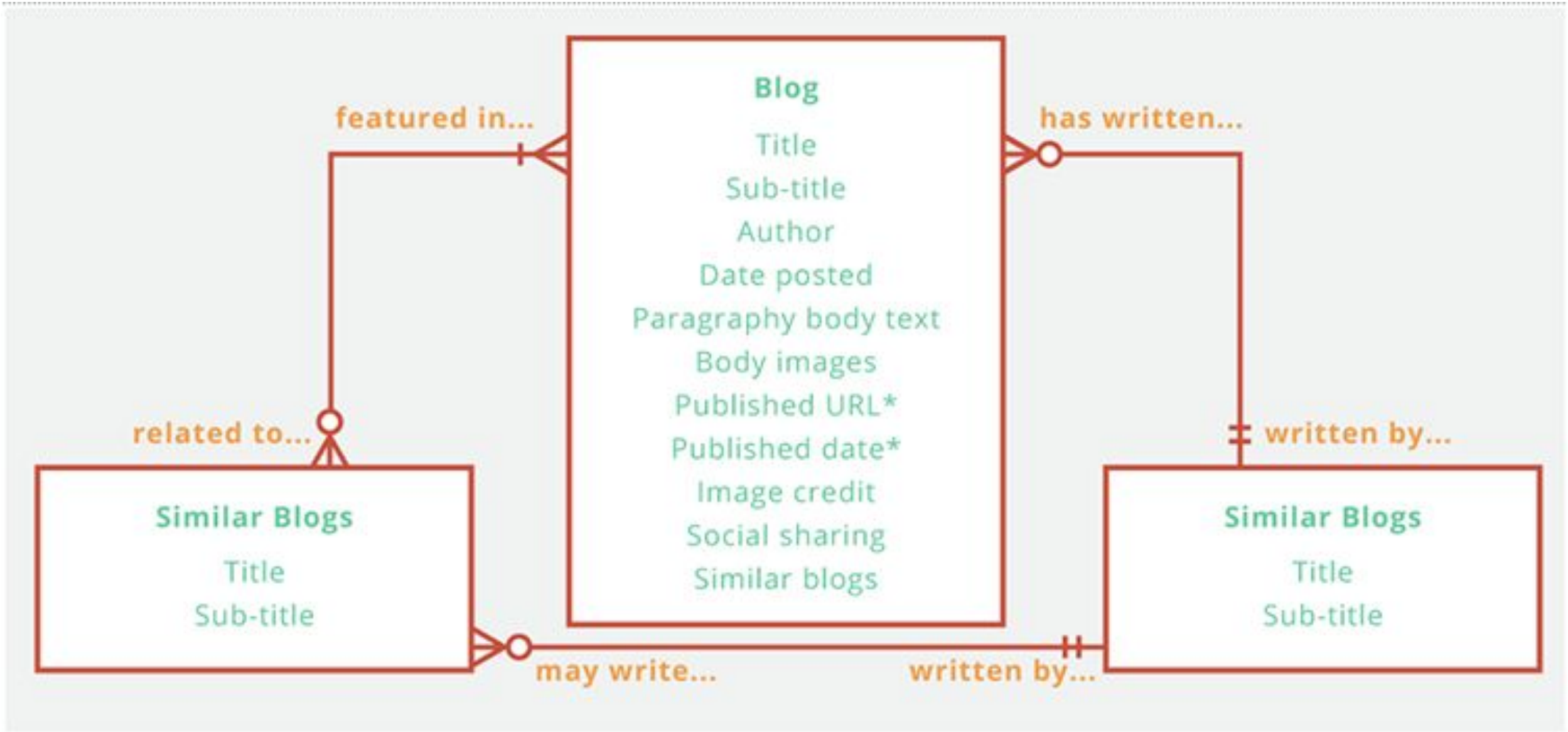
- **Content types**
  - A content type is like a template you follow to make multiple pieces of content in the same vein.
  - For example, a university might have a content type ‘template’ for course pages, subject areas, and academic bios.
  - Reuse the ‘template’ to create multiple pieces of content in the same format.

- **Content attributes**
  - Content attributes are the different elements that come together to make up the content type ‘template’.
  - For example, your course page content type might consist of attributes like course name, description, modules, fees, etc.

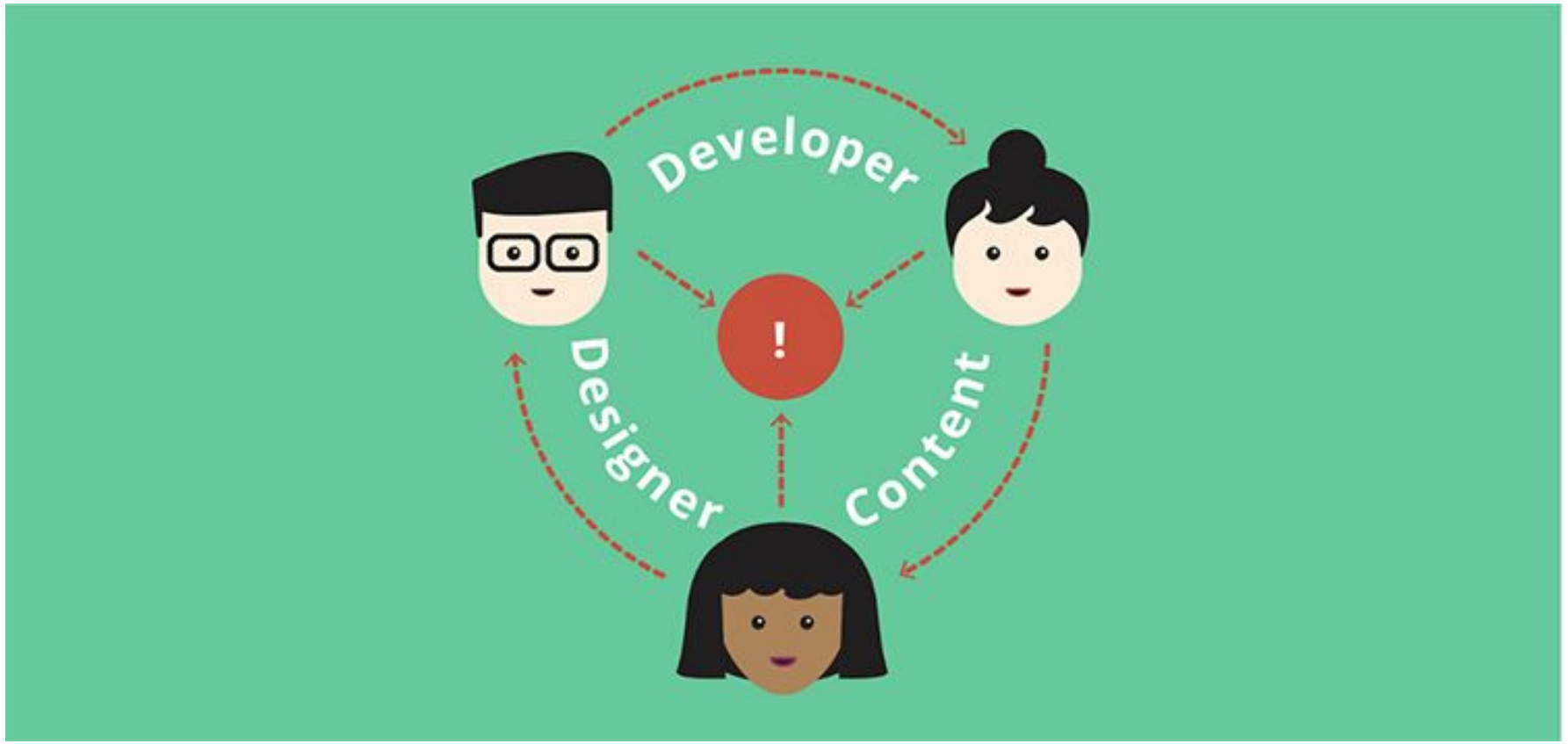
Content Model

- A content model should contain:
  - detailed definitions of each type of content (blog, web page, draft, etc.),
  - the components needed for each of these content types (fields like H1, meta description, body text, etc.),
  - and the relationship (hierarchy, internal linking, etc.) between all the different content types.


Member
Reviews
Member
Reviews
Member
Reviews
Members
Member
Watched
Member
Watched
Movie
Related
Series
Series Episodes
Episode


Blog
Title
Sub-title
Author
Date posted
Paragraphy body text
Body images
Published URL*
Published date*
Image credit
Social sharing
Similar blogs
featured in...
has written...
related to...
written by...
Similar Blogs
Title
Sub-title
may write...
written by...
Similar Blogs
Title
Sub-title
Object
Mandatory one
Mandatory many
Optional one
Optional many
None

How content is modeled?



How content is modeled?

- An effective content model encourages collaboration among designers, developers, and content producers while definitively conveying project requirements and goals.
- **Information Architects and Designers**
  - A content model tells information architects and designers what kinds of content and how much of it each section needs to display.
  - With it, they may also be able to establish module “templates” to ensure a consistent user experience throughout the site.
- **CMS Developers**
  - Development team to have a content model that carefully details your content needs and goals.
- **Content Producers**
  - A detailed content model is a helpful outline for the authors and producers who will eventually create and upload content into the CMS..

3 Steps to Creating Your Own Content Model

1. Create a Content Map

2. Take Stock of What You Have (Or Need)

3. Determine Your Content Types and Components

4. Define Relationships to Bring the Whole Thing Together

Create a Content Map

- List out the content types and break them into attributes
- Add lines and labels to show relationship between them

Content attributes

Course name

Course description

Modules

Fees

Content type

Course

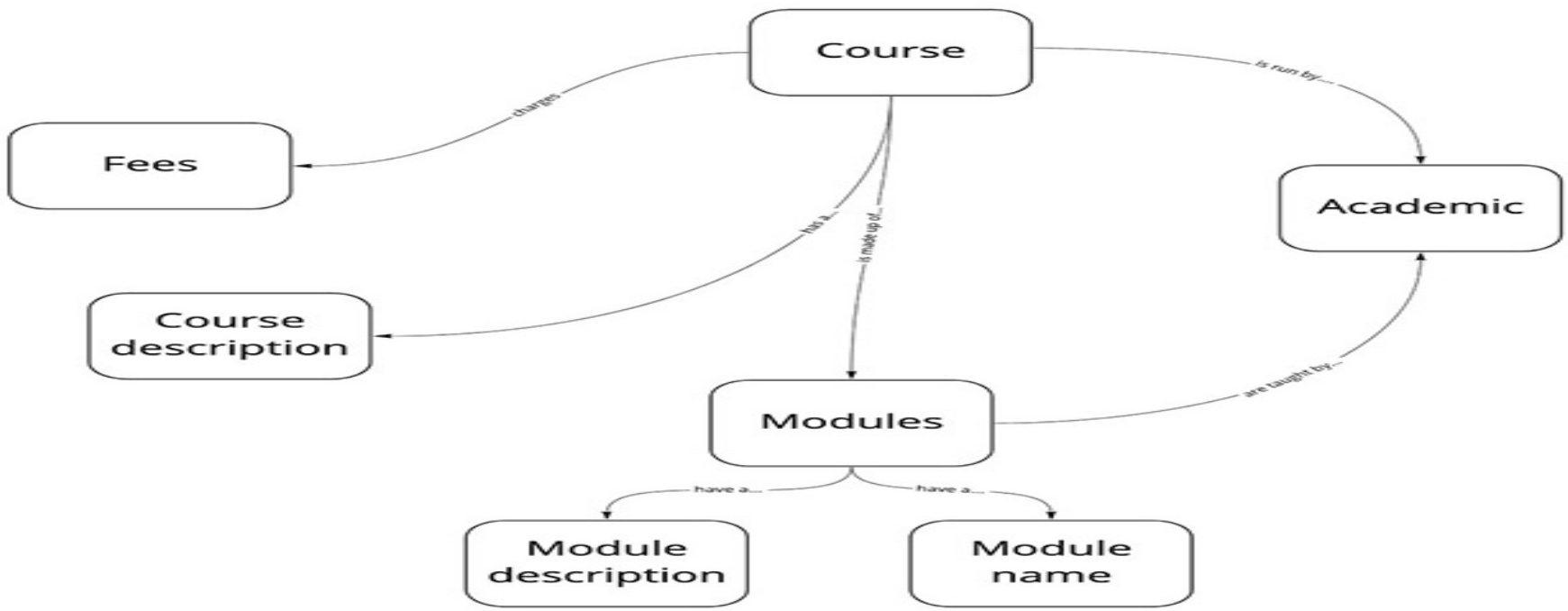
Course name

Course descripti

Modules

Fees

Content map



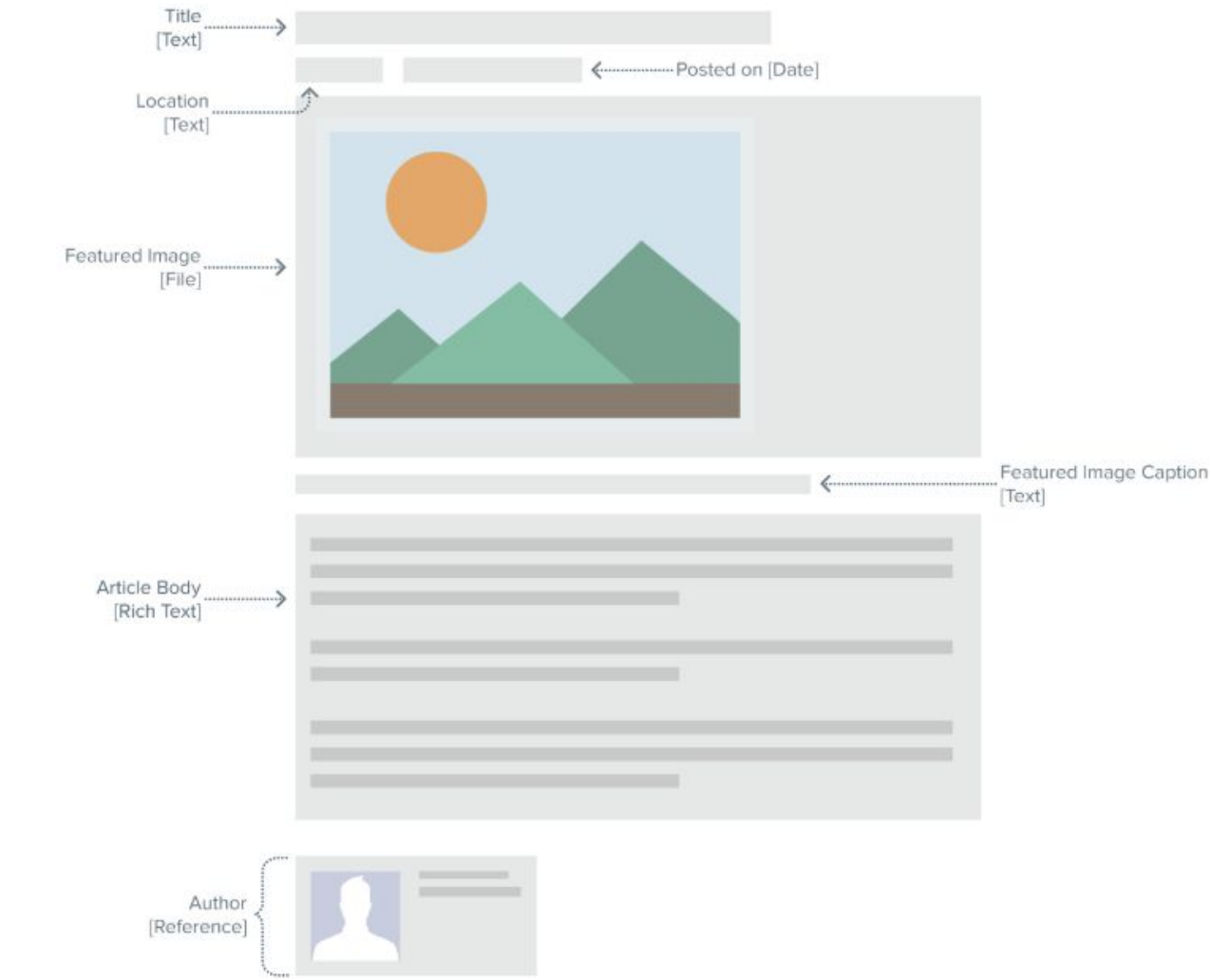
Label	H1, H2, H3	Content type	Compulsory	Owner
Give the content attribute a descriptive name to help you identify it quickly.	Is this content attribute an H1, H2, or H3?	What content type(s) is this part of?	Is this content attribute compulsory or optional in the content type?	Who is responsible fo this content attribute
Title tag	N/A	All	Yes	Author
Meta description	N/A	All	Yes	Author
Title	H1	All	Yes	Author
Standfirst		Article	Yes	Author

Take Stock of What You Have (Or Need)

- Redefine the infrastructure of your content, first you have to know what content you have.
- After gathering all your content in one place, review it to decide what to keep and what to eliminate.
- Next, develop a taxonomy that will inform the content model and help you port existing content into the final CMS.
- If you're starting from scratch, during this first phase of content modeling you will want to create a master outline, complete with taxonomy, for content to be created in the future.

Determine Your Content Types and Components

- With your outline in place, it's time to determine what kinds of content you'll eventually need.
- Types of content could include an author bio, a blog post, a call to action, an image gallery, a testimonial showcase, a navigational menu, and so on.
- Within each of these types are the components (which may also be called "fields," "elements," etc.) which a content creator or manager will eventually fill in with live content.
- These component fields are usually labeled with things like "title," "date," "body," etc.
- Don't forget invisible components like metadata and tags that consumers may not see but that play a vital role in your content's scalability and search engine ranking.

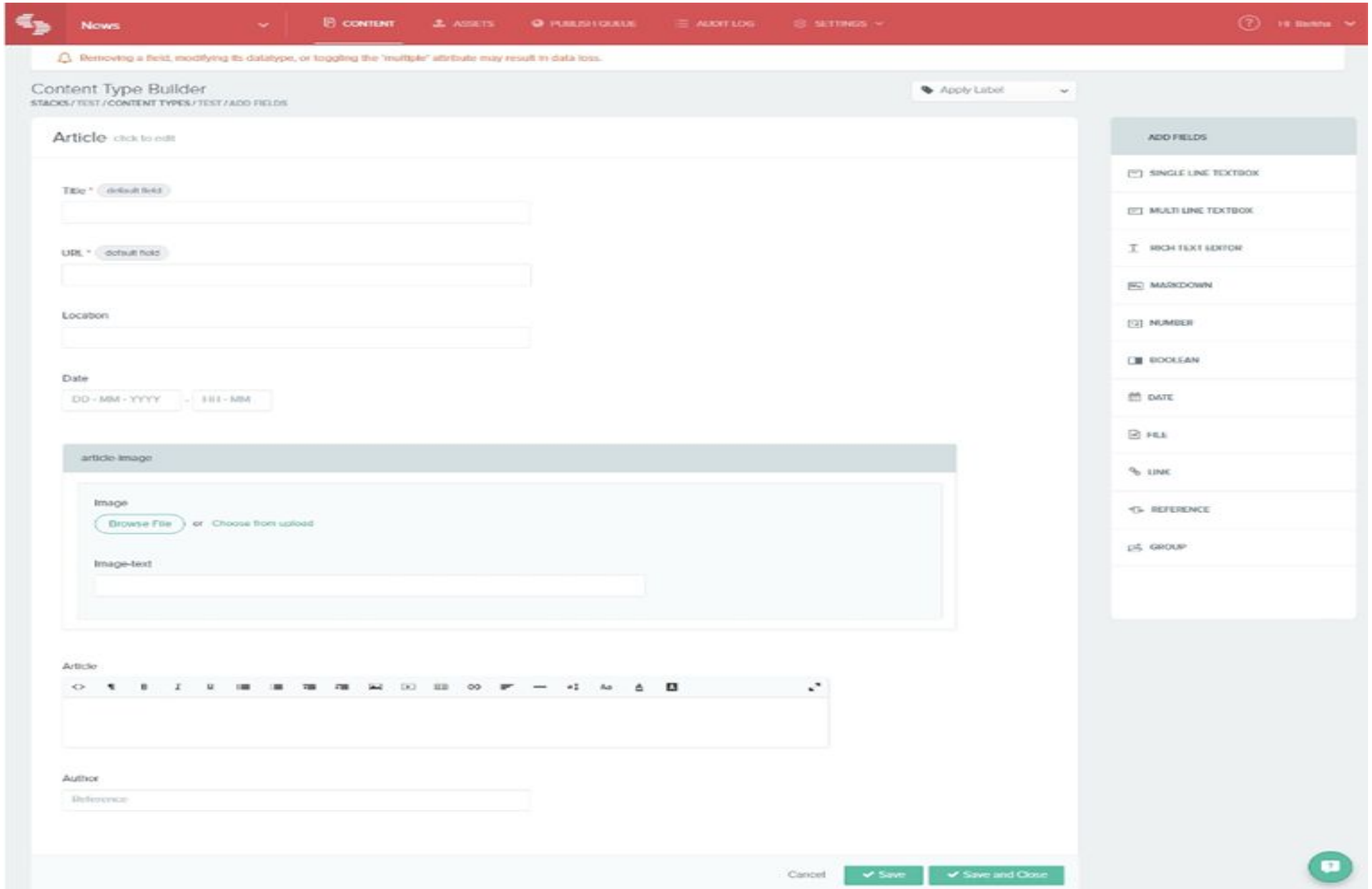

Title
[Text]
Posted on [Date]
Location
[Text]
Featured Image
[File]
Featured Image Caption
[Text]
Article Body
[Rich Text]
Author
[Reference]

Determine Your Content Types and Components

- The point of this step is to create a blueprint for the reusable, customizable content modules that designers and developers will bring to life in the CMS.
- Here, you may design a rough sketch or wireframe so content managers can get an idea of how the content will be organized, developers can clarify questions about functionality, and you can gather feedback on how the content plan will be implemented.

Define Relationships to Bring the Whole Thing Together

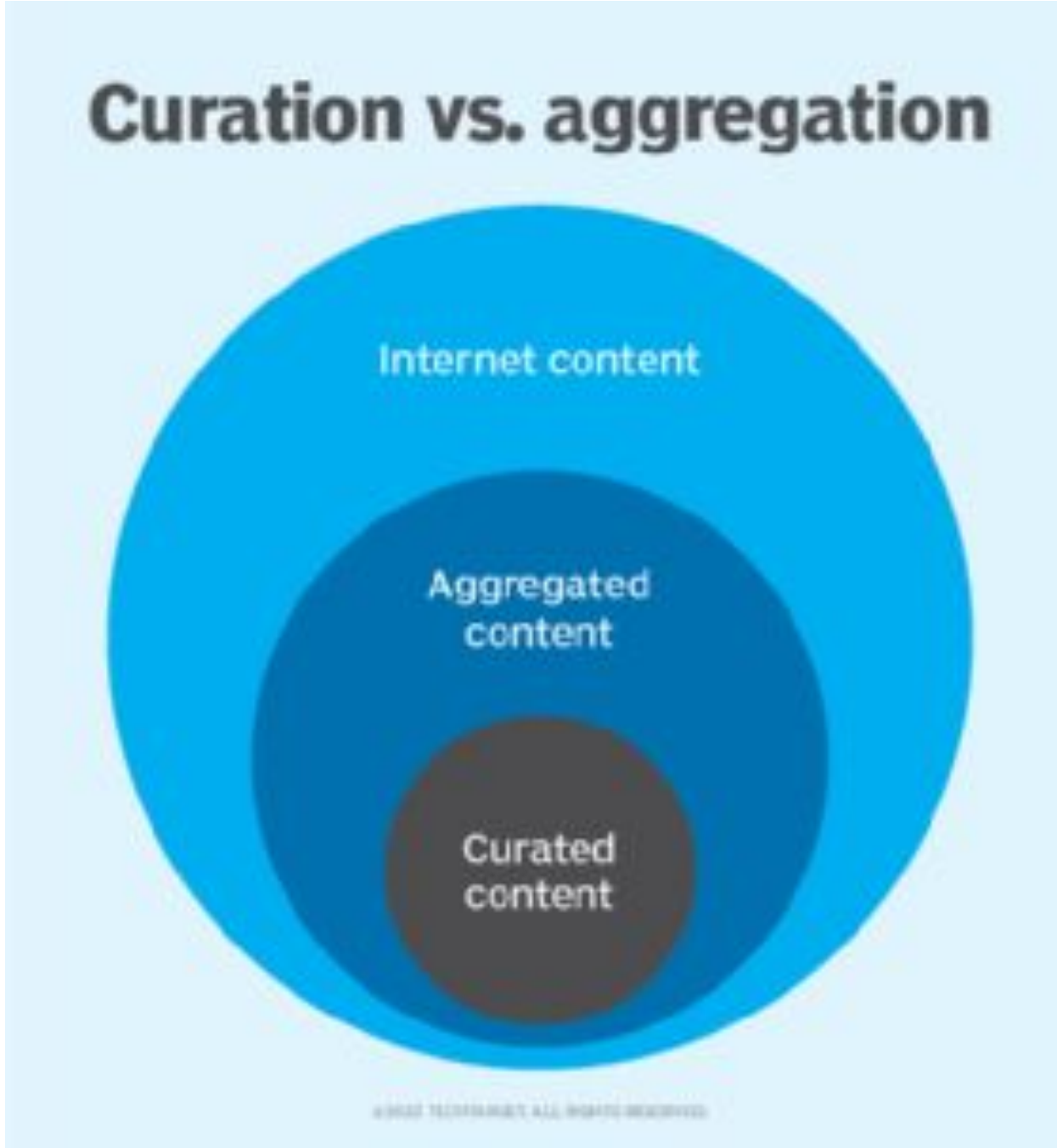
- You know what content you want and how you want to display it.
- Now, it is time to flesh out the final content model by defining how all of these elements function in relation to each other.
- This is when you'll finally draw out a content model to help designers create consistent templates and developers ensure they're building out connections and functionality correctly.
- The relationships you define in the content model will determine the workflow of the final CMS.


News
CONTENT
ASSETS
PUBLISH QUEUE
AUDIT LOG
SETTINGS
Removing a field, modifying its datatype, or toggling the "multiple" attribute may result in data loss.
Content Type Builder
STACKS / TEST / CONTENT TYPES / TEST / ADD FIELDS
Apply Label
Article click to edit
Title *
URL *
Location
Date
DD - MM - YYYY
article-image
Image
Browse File
or
Choose from upload
Image-text
Article
Author
Reference
Cancel
Save
Save and Close
ADD FIELDS
SINGLE LINE TEXTBOX
MULTI LINE TEXTBOX
RICH TEXT EDITOR
MARKDOWN
NUMBER
BOOLEAN
DATE
FILE
LINK
REFERENCE
GROUP

CONTENT AGGREGATION

What is content aggregation?

- A content aggregator is an entity that pulls together web or media content, applications or both from online sources for reuse or resale. It's a means of curating content.
- Two types of content aggregators exist:
  - those who gather news and other materials from various sources for publication on their own Web sites
  - those who syndicate content, gathering and distributing material that suits their customers' needs


Curation vs. aggregation
Internet content
Aggregated content
Curated content

What is content aggregation?

- Content aggregation involves collecting materials in one place, such as:
  - blogs
  - newsletters
  - newspaper and magazine articles
  - social media posts

Why use a content aggregator?

- Content aggregator tool can:
  - Enhance marketing efforts as part of a content marketing strategy
  - power your business applications
  - drive your corporate intranet
  - empower you to add value to customers
  - keep you up to date on industry information

Types of content aggregators

- Aggregators differ based on the type of content they work with and the sources they gather content from.
- Some of the types of content where an aggregator may be used include the following six:

1. **Blogs.** Blog aggregators gather niche blog posts from multiple sources and present them on a central site. Blog Engage is an example of a blog aggregator.
2. **News.** These aggregators gather news from multiple sources. Examples include Google News and Apple News.
3. **Social media.** Social media aggregators, such as Curator, take information from various social sites like Facebook and Twitter and display that information as a live feed.

Types of content aggregators

4. Research Research aggregators gather information from research journals to answer questions from specialists or to keep up with trends in various industries. Feedly can be used to aggregate research articles.

5. Services. Service aggregators gather multiple service providers and categorize them to make it easier for users to browse through the choices and select one. For example, Airbnb presents all the possible places a user could stay in a particular location.

6. Video. Video aggregators bring together recently published videos on specific topics from a variety of sites. YouTube is an example of a video aggregator.

Google News

Search for topics, locations & sources

- Top stories
- For you
- Following
- Saved searches
- COVID-19
- U.S.
- World
- Your local news
- Business
- Technology
- Entertainment

Headlines

More Headlines

COVID-19 news: See the latest coverage of the coronavirus

Arkansas governor says large businesses in state should not comply with Biden administration's 'oppressive vaccine mandate'

CNN · 3 hours ago

- **Walensky says Sotomayor's pediatric COVID hospitalization number was off dramatically**
  Fox News · 3 hours ago
- **Vaccine Mandates Have a Bad Day at the Supreme Court**
  The New Yorker · 21 hours ago
- **Despite continued idiocy, COVID sense is finally winning out**
  New York Post · 17 hours ago · Opinion
- **Federal vaccine mandate enters 'major question' land | The Hill**
  The Hill · Yesterday · Opinion

Examples of content aggregators

- Content aggregators are usually web-based tools or applications.
- Some tools can aggregate different content types and are often customizable to enable users to focus on specific types of content.
  - Google News aggregates news content
  - Apple Podcasts aggregates podcasts.
  - Rotten Tomatoes aggregates movie reviews.
  - Reddit is a news and social content aggregator.

How to start with content aggregation?

- Sites – WordPress
- Requirements:
  - A domain name
  - Web hosting service
  - Access to RSS feed plugin or feature

Best Practices

1. **Add value.** Posting content from another source should be part of a broader context that enhances the user experience. There should be a good reason why the user is getting the content from the aggregator as opposed to using the original source.
2. **Diversity is important.** If the goal of the aggregator is to present users with a selection of content from the web, make sure all of the content doesn't come from the same source. Having different content sources adds value.

Best Practices

3. Link to the original. Give users the option to access the original source of the aggregated content, especially in the case of news stories and blogs. This benefits both the creator and the user.

4. Quality is more important than quantity. It is important to present only valuable information to readers because the point of aggregation is to organize information and do some of the screening for the user. This is why curation to complement aggregation is often a good idea.

Benefits

- Faster learning
- Automation opportunities
- Increased traffic
- Low cost
- Customization
- Improved user engagement
- Trend identification
- Diversified information
- Search Engine Optimization

Tools

- Octoparse
- Google News
- Taggbox
- Castbox
- Alltop
- Flockler
- Feedly

PLUG-IN

What are Plug-ins?

- plug-in, also called add-on or extension, computer software that adds new functions to a host program without altering the host program itself.
- Widely used in digital audio, video, and Web browsing, plug-ins enable programmers to update a host program while keeping the user within the program's environment.

History of Plugins

- Plug-ins first gained popularity in the 1990s as software and microprocessors became more powerful.
- One of the first programs to make extensive use of plug-ins was Adobe Photoshop, an image-processing and editing program.
- Early plug-ins provided enhanced functions such as special effects, filters, and other options for manipulating images within Photoshop.

Why are plugins used?

- Plugins are software additions that are used to add or extend functionality to your website.
- For instance, you have required a plugin to handle everything if you are going to take donations or sell products on our site.
- There are also some other plugins that are mainly useful for WordPress websites, such as a WordPress SEO plugin, a WordPress forms plugin, a WordPress security plugin, a WordPress backup plugin.

Types

- Custom language support.
- Framework integration.
- Tool integration.
- User interface add-ons.
- Themes

Plugins

- Plugins are also known as ‘add-ons’ or extensions
- **WordPress Plugins**
- **Adobe Acrobat Reader**
- **Bukkit Plugins**
- **HP Print Service**