IBM DocumentHub

Fast and lightweight content management system, which stores documents in GitHub

- Why DocumentHub?
- Platform Architecture
- Microservices
- Web Components

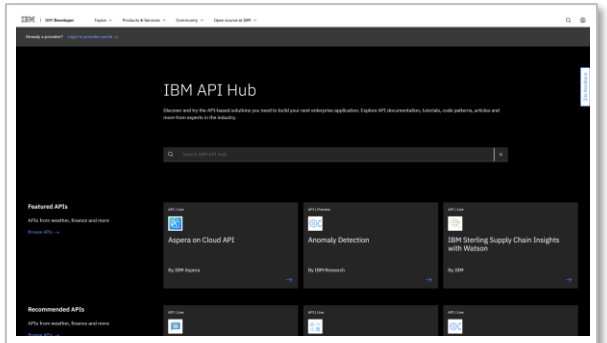
IBM

Who uses DocumentHub?

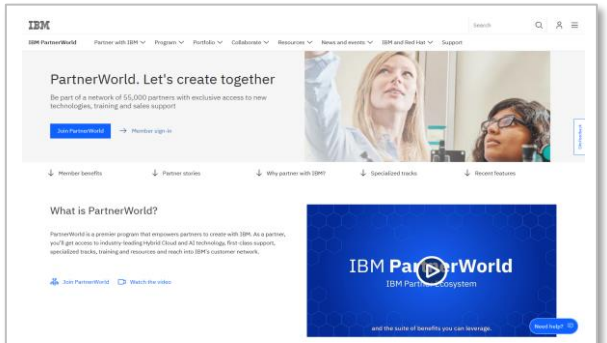
100,000+
requests/day

1 millisecond
execution/request

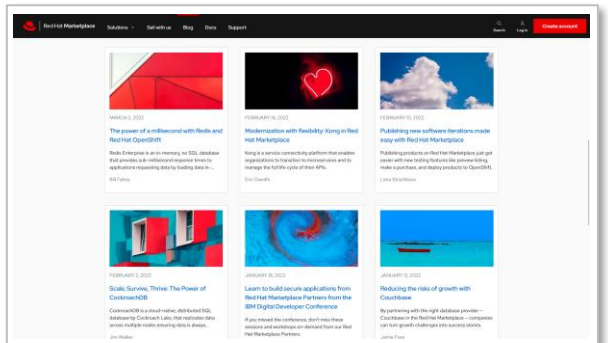
200,000+
documents



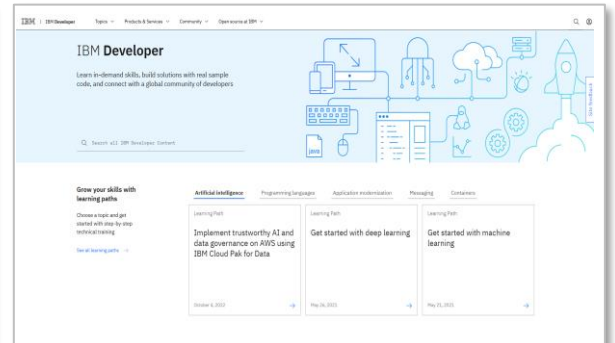
API Economy developer.ibm.com/apis



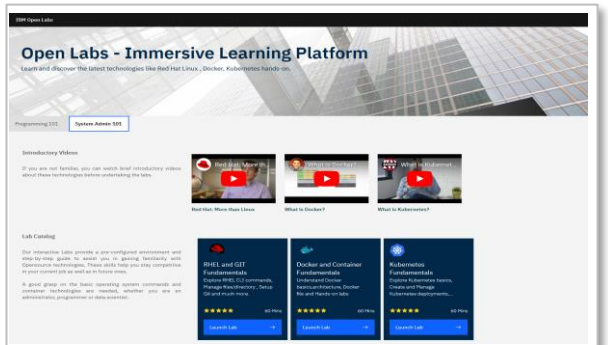
PartnerWorld ibm.com/partnerworld



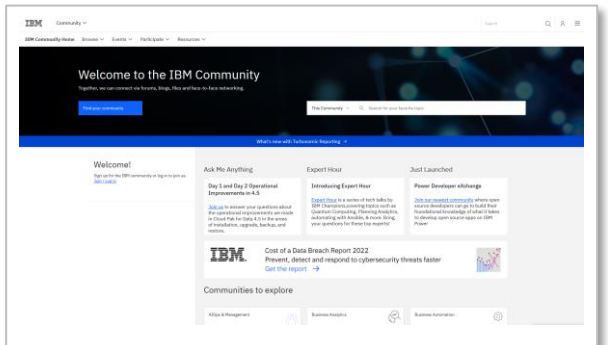
RedHat marketplace.redhat.com/en-us/blog



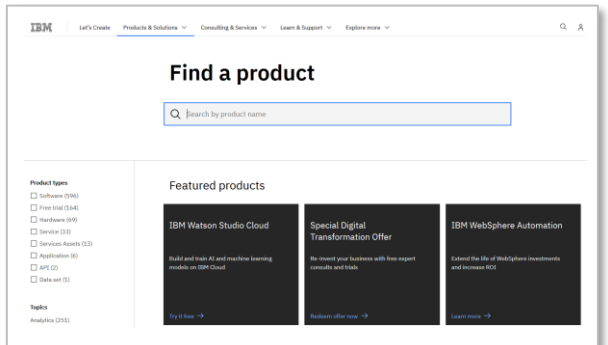
IBM Developer developer.ibm.com



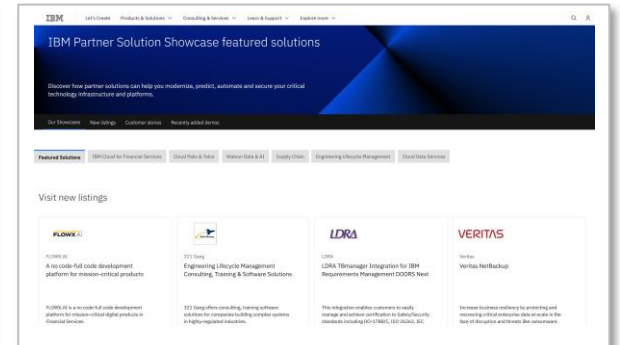
Open Labs developer.ibm.com/openlabs



IBM Community community.ibm.com



IBM Products ibm.com/products



Partners Showcase ibm.com/partners/showcase

Why DocumentHub?

Scenario 1/3: Choose technology stack for a new application

An unpopular technology stack means:

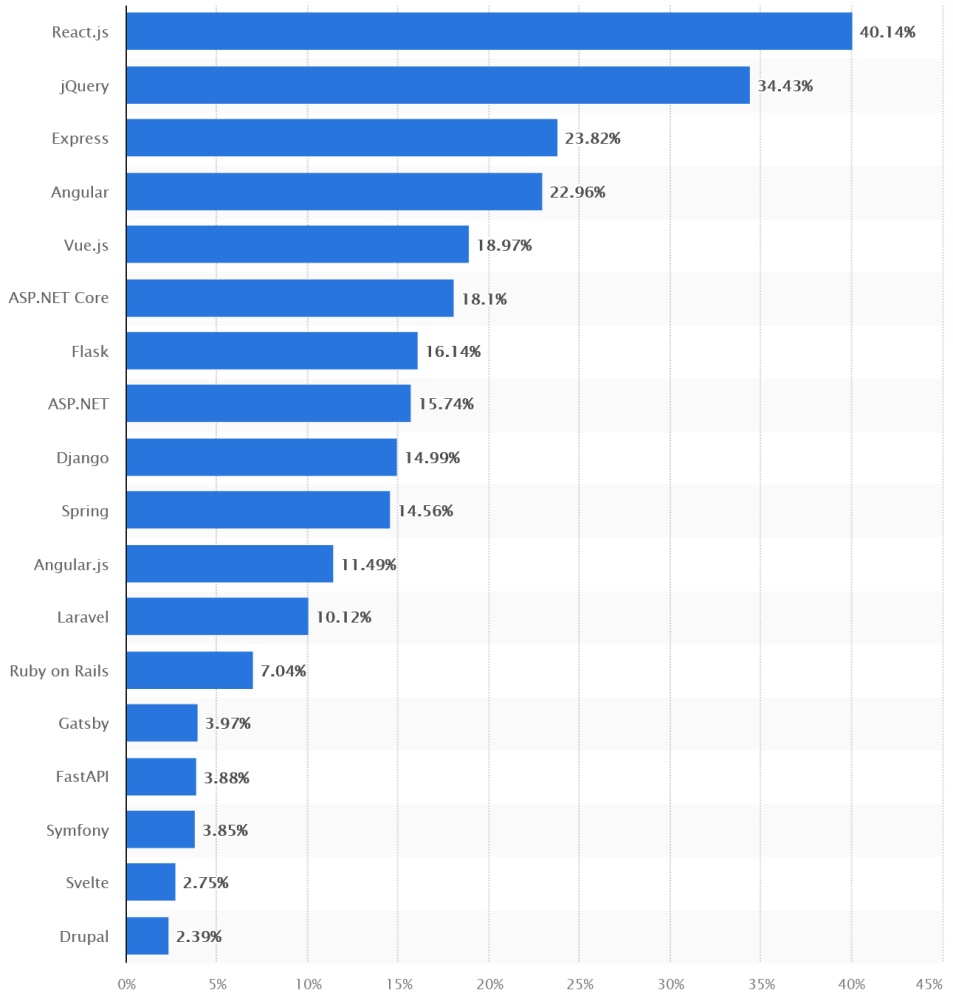
- fewer developers available
- less interest in the project
- higher learning curve
- higher costs

Content Management System	Compatibility with web frameworks
DocumentHub	**100%** (compatible with all top frameworks)
Adobe Experience Manager	< 10%
Salesforce	< 10%
WordPress/Drupal	< 20%

DocumentHub

- ✓ **developer friendly** – compatible with all top frameworks
- ✓ **increased productivity**
  - popular technologies
  - simplifies code and architecture

Most used web frameworks among developers worldwide, as of 2021



Why DocumentHub?

Scenario 2/3: A user uploads an infected file

*"It is estimated that, worldwide, cyber crimes will cost **$10.5 trillion annually** by 2025" – Cybercrime Magazine*

Other users download the infected file

The manager starts receiving complains that the application is spreading malware

Most **CMS platforms** don't have virus scan

- Infected files are uploaded without scanning
- Infected files are downloaded by other users without scanning

DocumentHub has antivirus scanning for all uploaded and downloaded files.

Why DocumentHub?

Scenario 3/3: A user discovers that his data suddenly disappeared

User blames the application for the missing data

The manager checks with the team to figure out if it's a user mistake or an application bug, before it escalates

The team's classic technology stacks can be of little help:

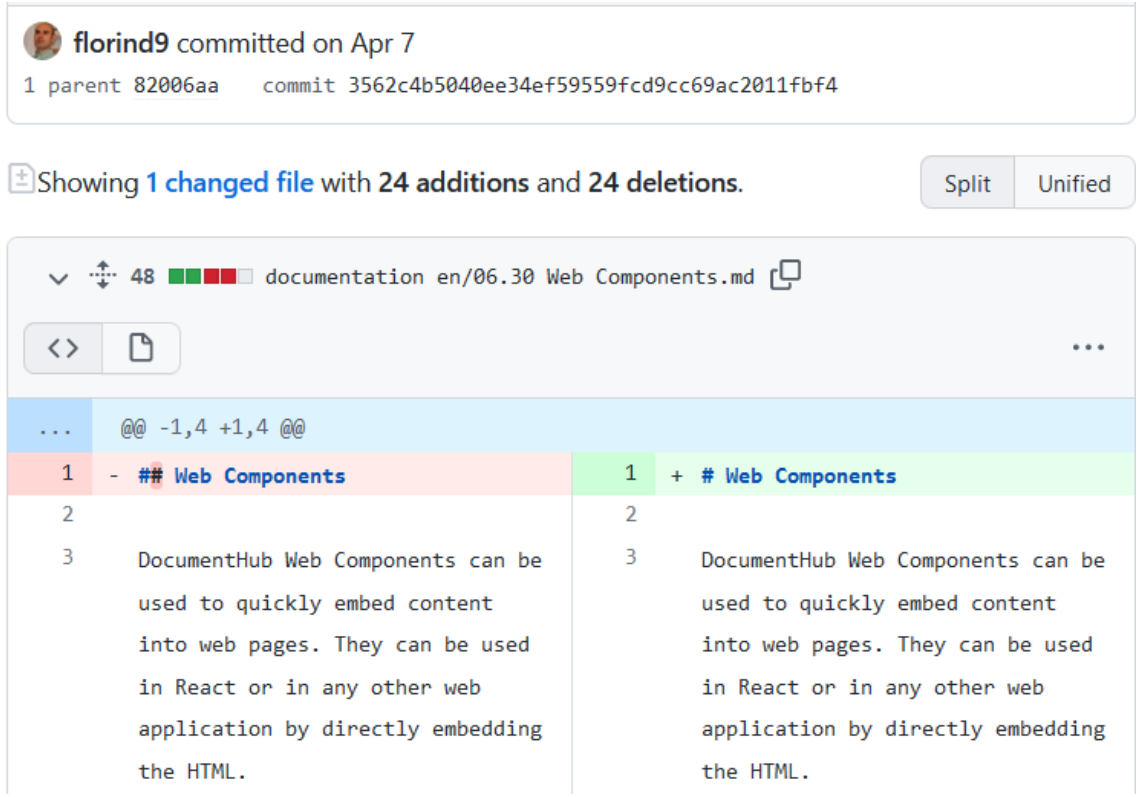
- Databases don't have a tracking system for the changes
- Databases don't correlate changes with the users
- Comparing the current database with a backup is difficult and time consuming
- Even so, the comparison will show a lot of irrelevant changes since the last backup
- Writing a complex change tracking system can be time consuming and expensive
- Even with a tracking system, only the user actions are logged but not the application bugs

How do you solve this, before it escalates?

Why DocumentHub?

DocumentHub stores the data and the content in GitHub

- GitHub is the best content versioning system
- Change log, history and differences for all the changes
- Changes are correlated with users
- Easy to identify a change and revert it
- Content is stored in a friendly, human readable format
- Friendly UI to browse and edit the content
- No database admins to run SQL queries. Anyone can manage the content
- Edit the content in GitHub UI, your application, or any external editor
- You own the content repository. DocumentHub only syncs the content with your application
- IBM Enterprise (github.ibm.com) or public (github.com)
- No costs



Why DocumentHub?

Developer friendly

- Compatible with all top frameworks, which developers prefer
- Simplifies code and architecture. Reduces complexity and risks

GitHub

- Content is stored in GitHub which is the best content versioning system
- Content is stored in a friendly, human readable format

Two-way sync

- Content can be changed in GitHub, and it will be synchronized in application, or in can be changed in application and it will be synchronized in GitHub

Search

- The fastest search engine, with accurate search for precise results and fuzzy search for finding matches even for typos

Queries

- Query the content like any other NoSQL database, at the same speed as in a database

Cloud Object Storage

- Media files are automatically uploaded to IBM Cloud Object Storage

Akamai CDN

- Media content is automatically cached around the globe and optimized for slow connections

Cache

- Two levels of cache: memory cache and database cache
- Access to content in GitHub at the same speed as in a database

Security

- Access control lists and access control rules for content
- Automatic filtering of the content based on the access for each user

Antivirus

- Virus scanning on demand for specific files or automatically for all uploaded and downloaded files

Web components

- Available web component like a full search page, a course or a quiz
- Add IBM SSO to your application with just one line of code

Web templates

- Speed up development with web templates
- App starter templates for React/Angular/Vue

Multi language

- Multi language support for documents

JavaScript module

- JavaScript module for a quick and simple access to content
- Automatic handling of content types and errors at client level

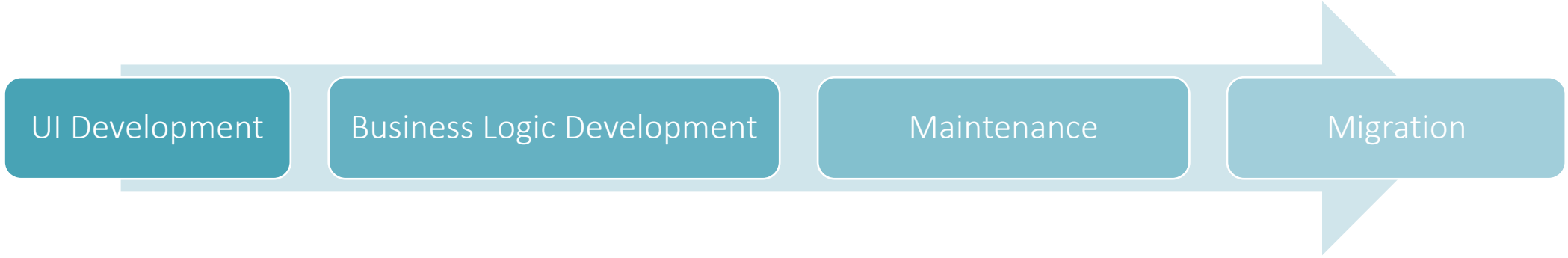
API

- Content can be accessed from any language or frameworks using API

Content architecture

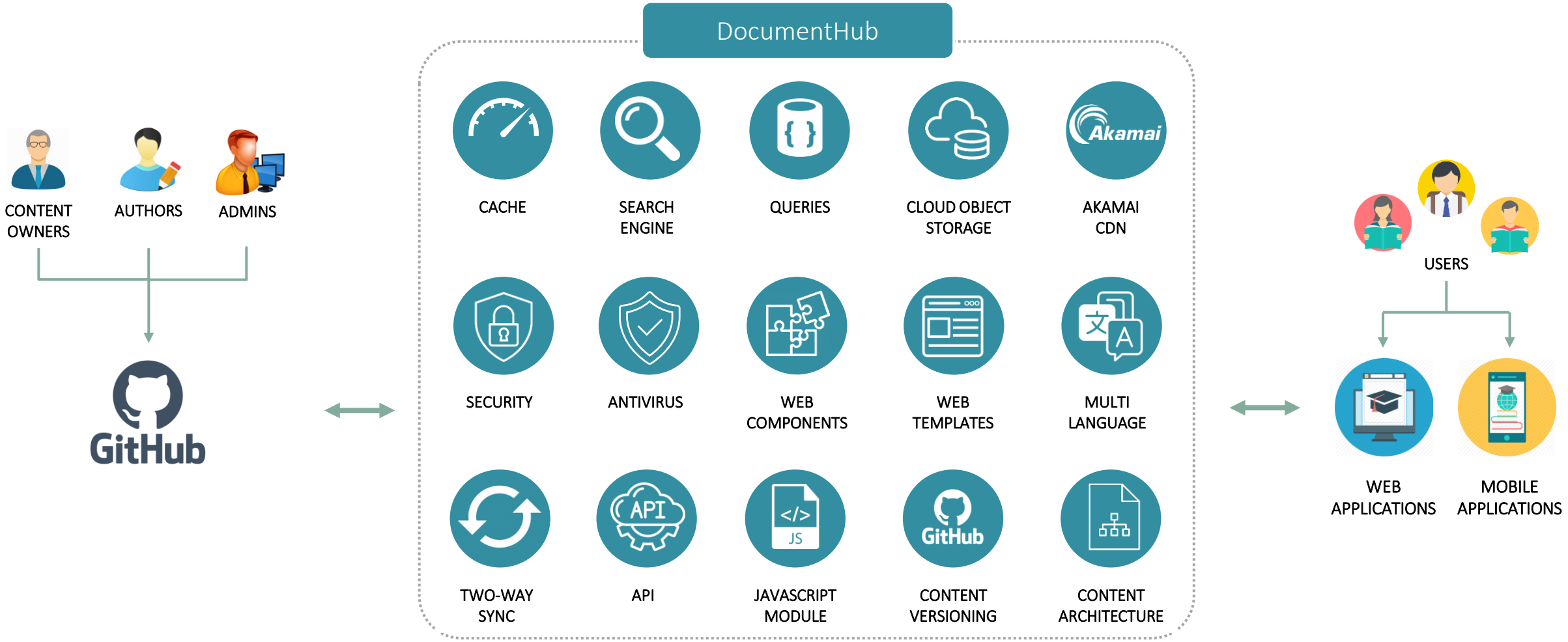
- Flexible content architecture supporting flat or nested folder structures
- Content is stored in a human friendly way

CMS Comparison - Lifecycle



Content Management System	UI Development	Business Logic Development	Maintenance	Migration
DocumentHub	Very Good Developers can choose any framework they prefer	Very Good Developers can choose any framework they prefer	Very Good + Use any editor + Cheap cloud instance	Very Good Code is developer friendly; content is in GitHub in a human friendly format
Adobe Experience Manager	Good Limited choices for language and frameworks, difficult for custom components	Poor Very limited choices, hard to find developers	Good + WYSIWYG editor - Expensive	Poor A monolith with a proprietary data format
Salesforce	Good Limited choices for language and frameworks, difficult for custom components	Poor Proprietary language, hard to find developers	Good + WYSIWYG editor - Expensive	Poor A monolith with a proprietary code and data format
WordPress / Drupal	Good Limited to PHP, difficult for custom components	Poor Limited to PHP, hard to find developers	Good + WYSIWYG editor - Not cloud friendly	Poor A monolith with a proprietary data format

How does it work?



Platform Architecture – decoupled CMS architecture

Content Creation

Use **any tool or platform** to edit the content

Content Repository

GitHub
public or enterprise

Content Delivery

Content as a Service

Content Presentation

Develop web or mobile apps on **any platform or framework**

Edit content with any tool or platform

- GitHub UI
- Your own content editor
- Any external content editor

GitHub - the best content repository

- Content is stored in a human readable format
- Content versioning
- Revision history, changes diffs, revert changes, activity log
- Dev, staging and production branches
- Users access control
- Enterprise and public GitHub

Content as a Service – very fast content delivery

- **Very fast microservices** with memory cache
- **Search and filter** the content by any metadata field
- **Content is reusable** in any other web or mobile application
- **Access control** for content and users
- **Cloud object storage** with **Akamai CDN**

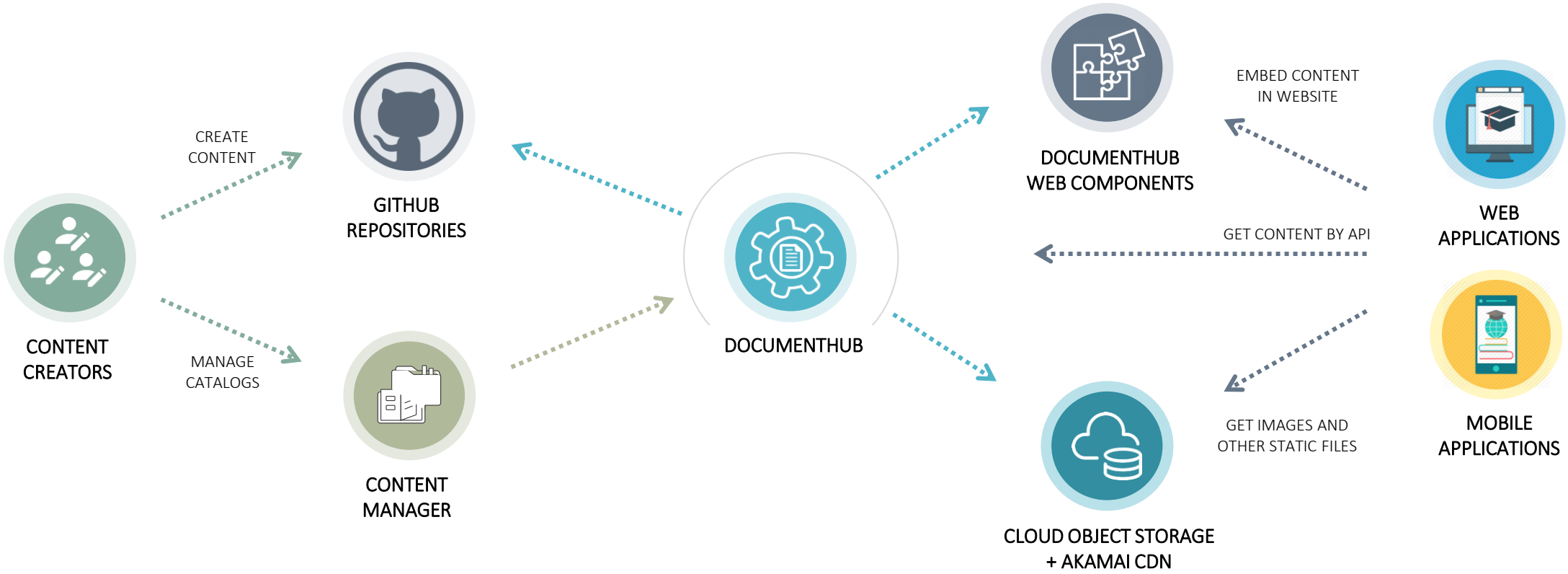
Web or mobile apps on any platform or framework

- **CaaS** delivers the content to any web or mobile applications on platform or framework
- **Web Components** are helping to quickly embed content in websites
- **MyLearning** can be used to quickly share a library with other people
- **Web/Mobile Templates** can be used for a faster development

Platform Architecture

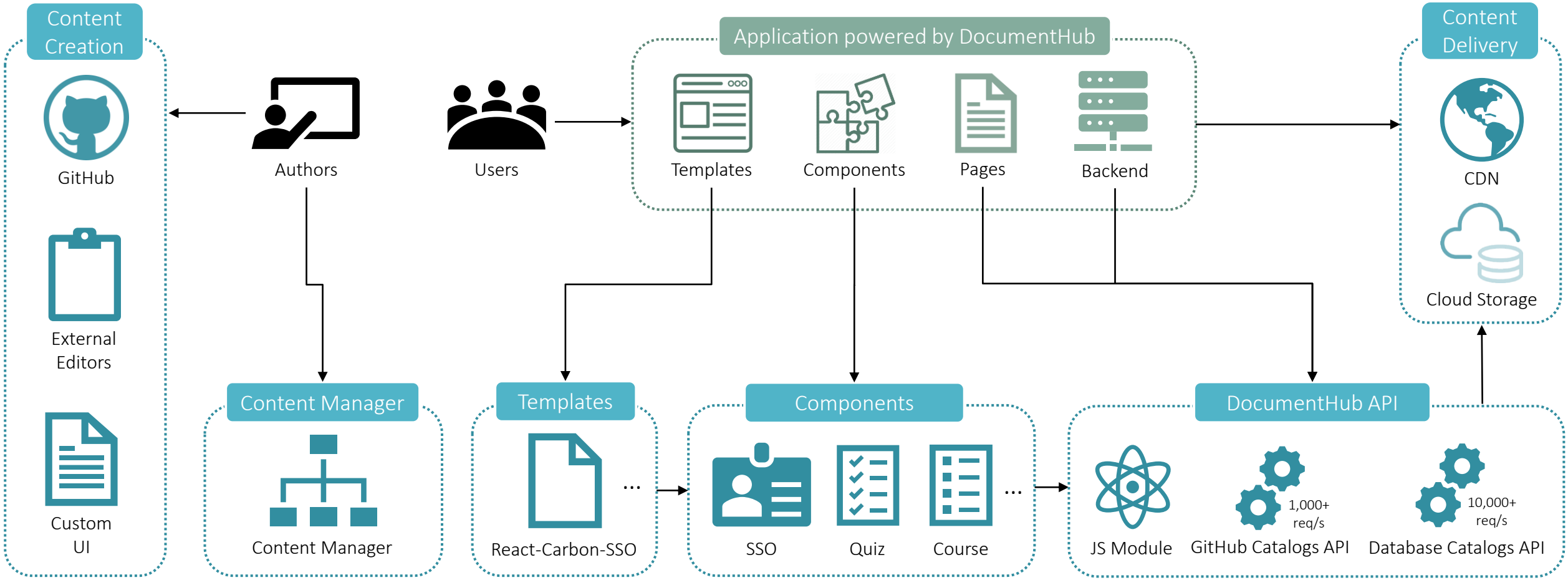
1 Manage the content in GitHub or application

2 Show the content on web or mobile



Content Creation | Content Repository | Content Delivery | Content Presentation

Platform Architecture



Content as a Service API

Very fast access to content

Edge CaaS microservices are using **memory cache** to quickly respond to any content request. API execution time is usually around only a **few milliseconds**.

Reuse the content on multiple websites

You can reuse the same physical document in multiple logical catalogs. In this way you can show documents on multiple websites **without cloning them physically**.

Search and filter the content

Edge CaaS uses the **fastest search engine** to **search** the text content and **filter** the content by any metadata field like title, tags, keywords, type or categories.

Access control on catalogs and documents

Access permissions can be set at **user level** or at **application level** for the entire catalog or for individual documents.

CDN and Cloud Object Storage

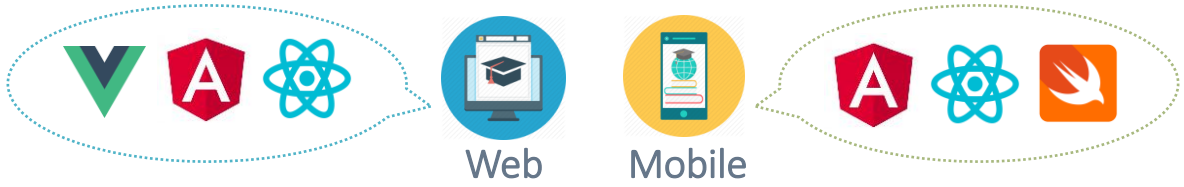
You can benefit of **Akamai CDN** to get the content cached in the nearest location. Attachments are **automatically uploaded to Cloud Storage.**

99.9% uptime

Edge CaaS is running in **IBM Cloud** and is using the **blue-green deployment** model to deploy updates with **no downtime**.

Web Components

Light and flexible applications with reusable components on any platform or framework.



API

Fast access to content
Search and filtering
Access control

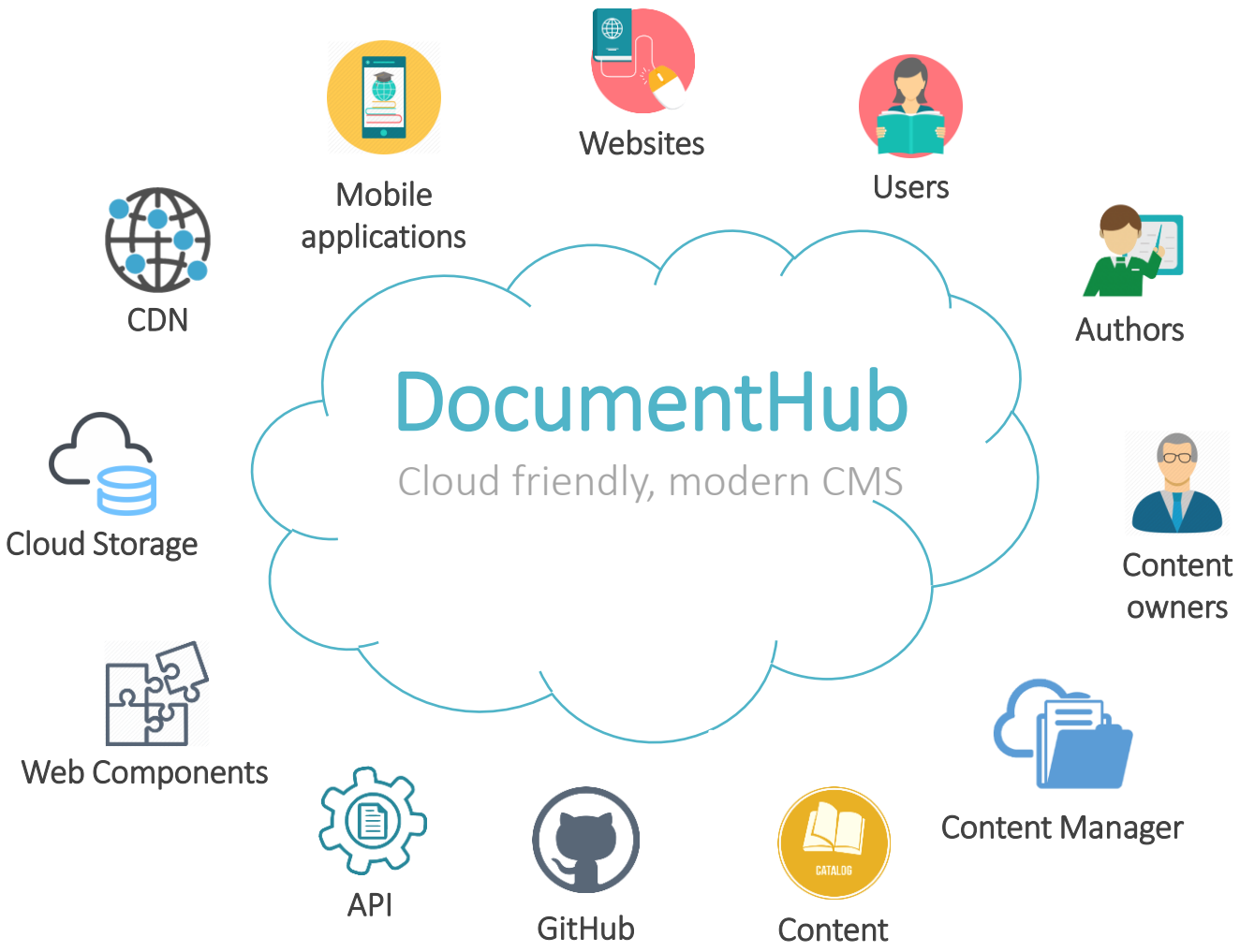
Web Components

Easy embed courses, quizzes, articles and other components

CDN and Cloud Object Storage

Public media files are stored in Cloud Object Storage and delivered by Akamai CDN

Summary


Mobile applications
Websites
Users
CDN
Authors
DocumentHub
Cloud friendly, modern CMS
Cloud Storage
Content owners
Web Components
Content Manager
API
GitHub
CATALOG
Content