

# IBM DocumentHub

Lightweight content management system, which stores documents in GitHub, in a human friendly way

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

# Why DocumentHub?

## Scenario: Choose technology stack for a new application

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

### An unpopular technology means:

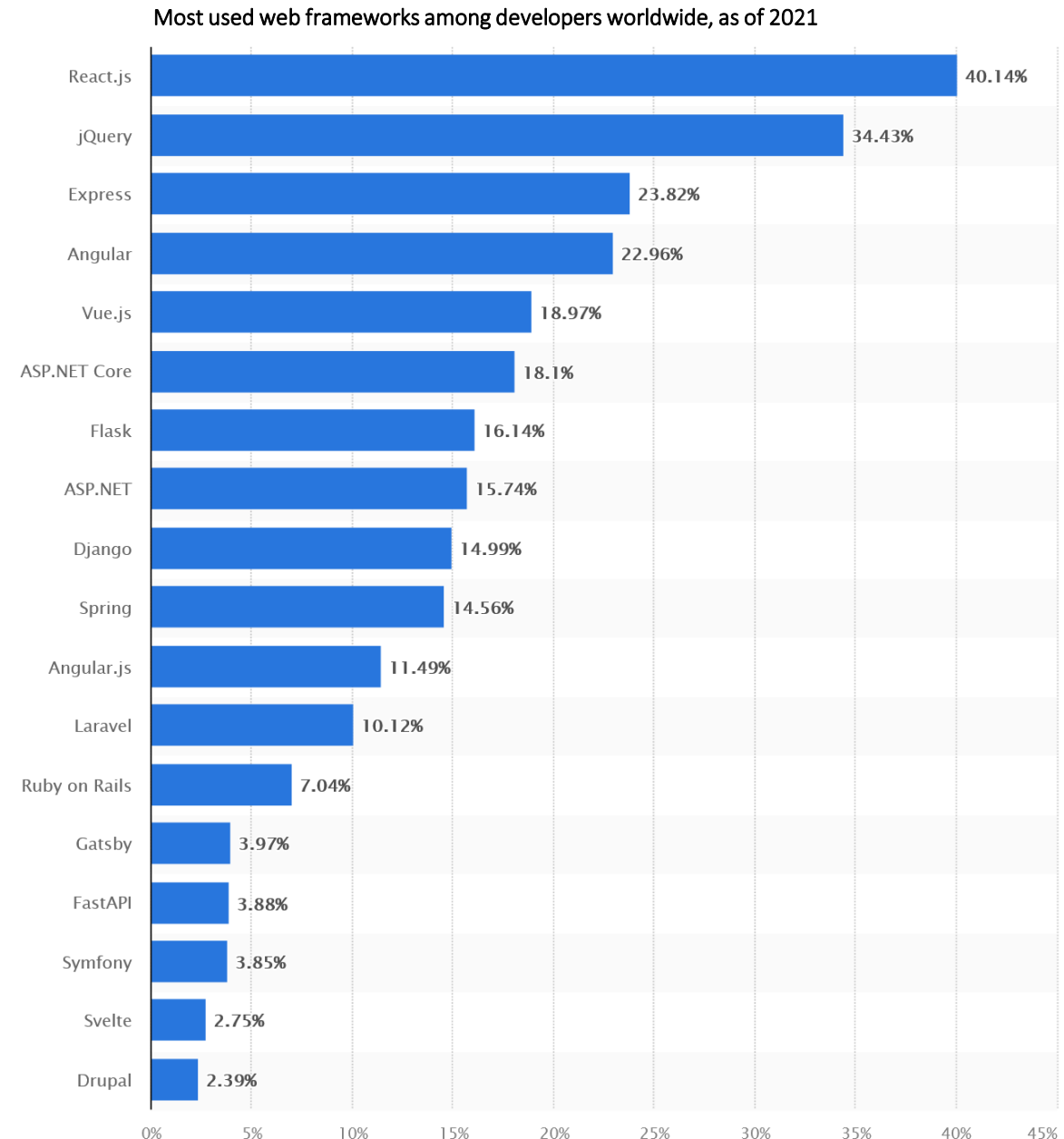
- fewer developers available
- less interest
- higher learning curve
- higher costs
- instability



**JavaScript** is the most preferred language by developers, constantly for the last 10 years

**Top 5** frameworks are JavaScript: React.js, jQuery, Express, Angular and Vue.js

**DocumentHub** is compatible with the latest technologies, which developers are enthusiastic to use.



# Why DocumentHub?

Scenario: A user discovers that some of his data is missing



**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:



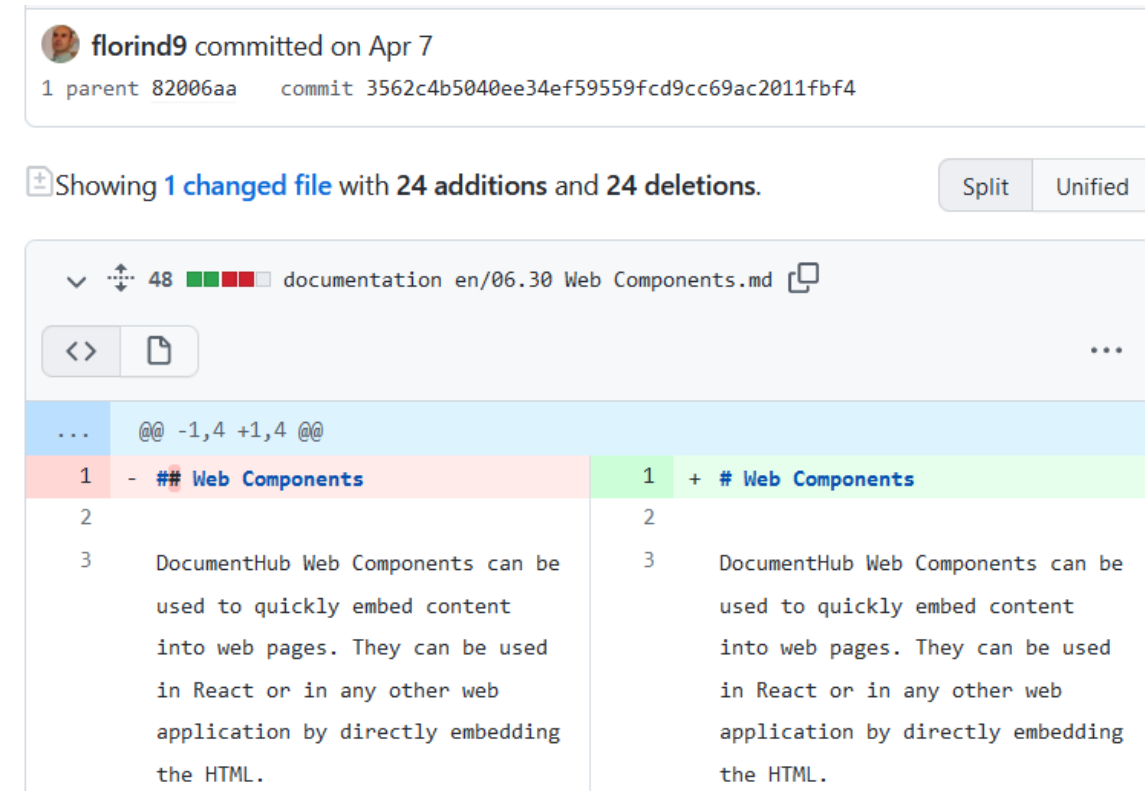
- **Databases** don't have a tracking system for the changes
- Databases don't correlate changes with the users
- Comparing the current database with a backup is difficult and time consuming
- Even so the comparison will show a lot of irrelevant changes since the last backup
- Writing a complex change 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?

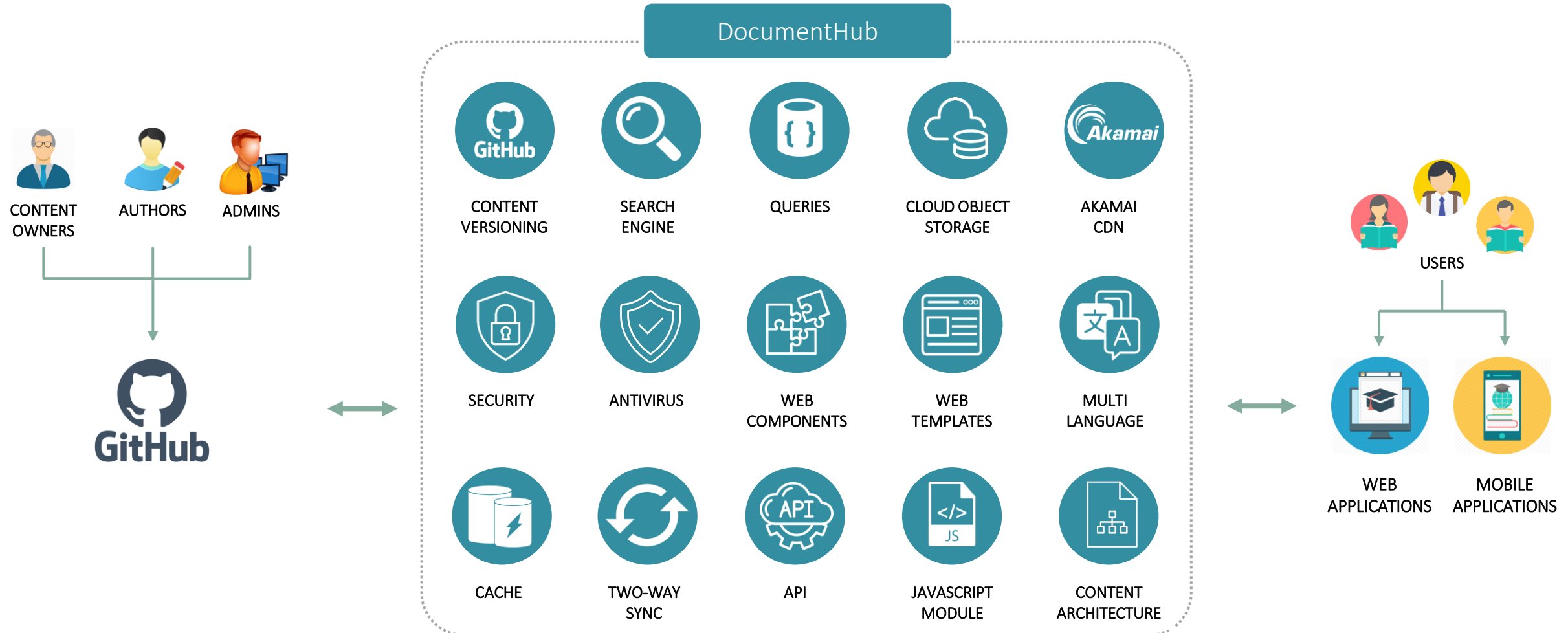
# Why DocumentHub?

## 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



# How it works?



# Why DocumentHub?

## Resources / Developers

- Compatible with the latest technologies, which developers are enthusiastic to use

## Time, Costs and Risks

- Faster development by using DocumentHub components and services
- Reduced costs with development, test, maintenance and hardware
- Reduced risks by using components and services proved to be reliable and secure

## GitHub

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

## Security

- Verified security with internal and external PEN tests
- Access control lists and access control rules for content
- Automatic filtering of the content based on the access for each user
- Virus scanning for files

## 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

## JavaScript Module / API

- JavaScript module gives a quick and simple access to content
- From any other language you can access the content with API

## Web Components / Web Templates

- Embed web component in your application like a full search page, a course or a quiz
- Add IBM SSO to your application with just one line of code
- Choose an app starter template for your application to speed up development

## Search

- The fastest search engine, 5x faster than IBM Enterprise Search
- Accurate search for precise results and fuzzy search for finding matches even for a typo

## Cache

- Two levels of cache: memory cache for JSON and text content and database cache for other files
- Access to content stored in GitHub to the same speed as a database

## Sync

- Two-way (bi-directional) synchronization between application and GitHub
- Update the content in application and the changes will be stored in GitHub
- Update the content in GitHub and the changes will be available in application in 2 seconds

# Who uses DocumentHub?



300,000+  
total users



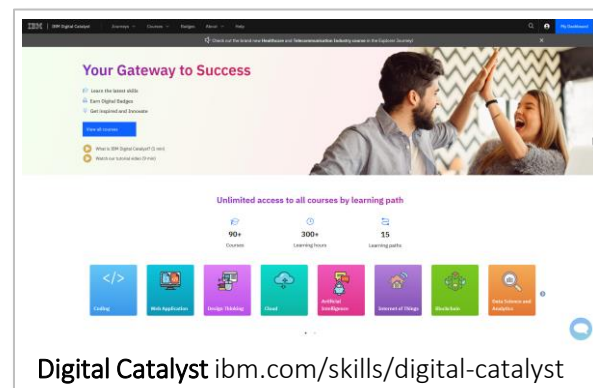
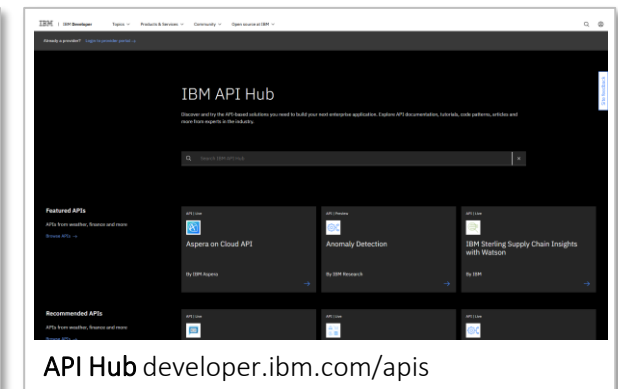
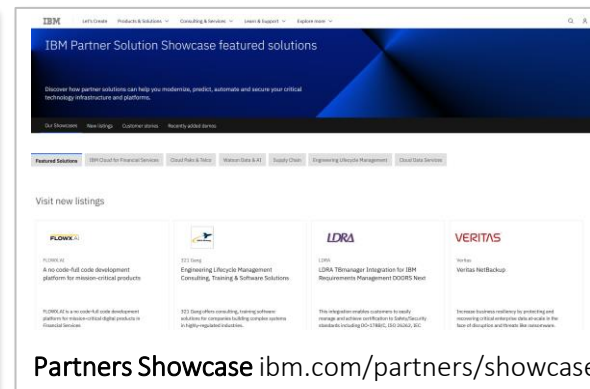
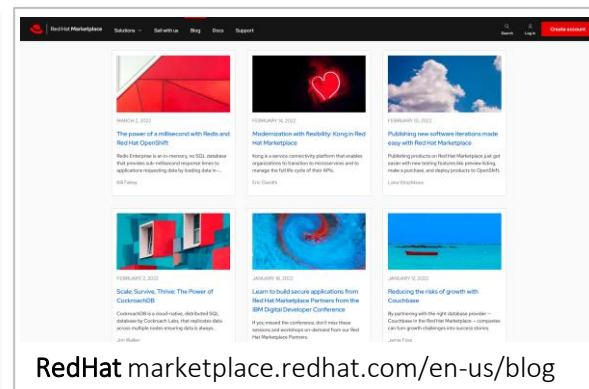
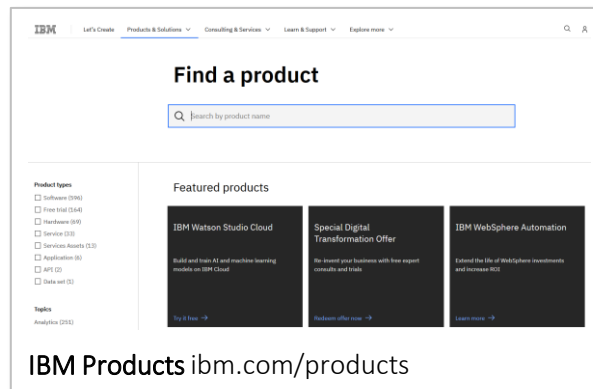
100,000+  
requests/day



1 millisecond  
execution time



1,000+  
GitHub branches



and others

# CMS Comparison - Lifecycle

UI Development

Business Logic Development

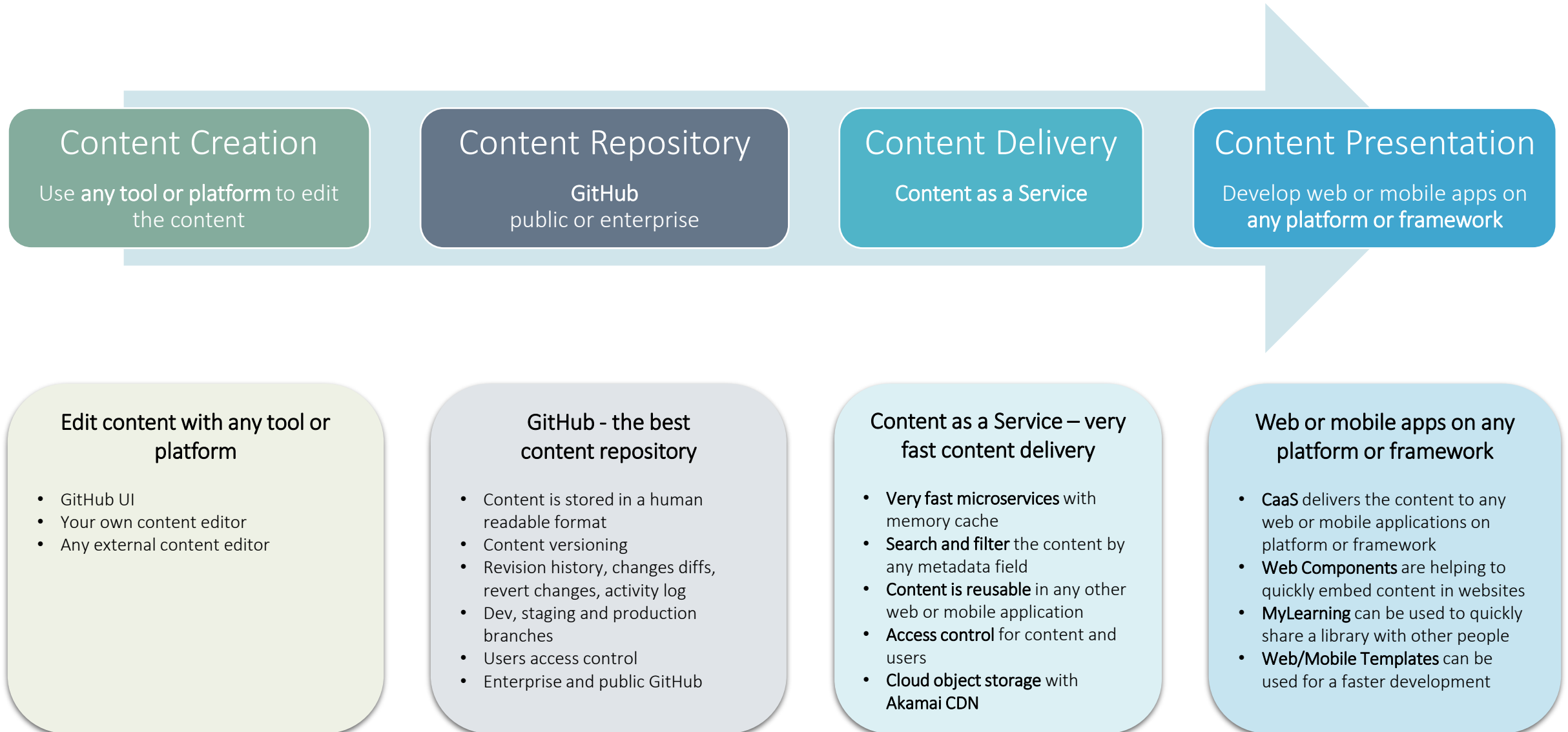
Maintenance

Migration

Content Management System	UI Development	Business Logic Development	Maintenance (content and config changes)		Migration (code or content)	
			Costs	Resources	Code	Content
<b>DocumentHub</b> * recommended for applications with custom development	<b>Very Good</b> Developers can choose the latest languages and frameworks they prefer	<b>Very Good</b> Developers can choose the latest languages and frameworks they prefer	<b>Very Good</b> Cheap cloud instance	<b>Good</b> No developers needed Has change history	<b>Very Good</b> Code is developer friendly	<b>Very Good</b> Content is in GitHub in a human friendly format
<b>Adobe Experience Manager</b> * recommended for large static websites	<b>Good</b> Limited choices, difficult for custom components	<b>Poor</b> Very limited choices, hard to find developers	<b>Poor</b> Pay per usage, expensive	<b>Very Good</b> Drag and drop, WYSIWYG editor	<b>Poor</b> Code is written in a proprietary format	<b>Poor</b> A monolith with a proprietary format
<b>Salesforce</b> * recommended for applications with a common UI and logic	<b>Good</b> Proprietary language, difficult for custom components	<b>Poor</b> Proprietary language, hard to find developers	<b>Poor</b> Pay per usage, expensive	<b>Very Good</b> Drag and drop, WYSIWYG editor	<b>Poor</b> A monolith with its own language	<b>Poor</b> A monolith with a proprietary format
<b>WordPress</b> * recommended for blogs or presentation websites	<b>Poor</b> Limited to PHP, difficult for custom components	<b>Poor</b> Limited to PHP, hard to find developers	<b>Good</b> Dedicated VM	<b>Good</b> WYSIWYG editor	<b>Poor</b> Code is written in PHP	<b>Poor</b> A monolith with a proprietary format
<b>Drupal</b>	<b>Poor</b> Limited to PHP, difficult for custom components	<b>Poor</b> Limited to PHP, hard to find developers	<b>Good</b> Dedicated VM	<b>Good</b> WYSIWYG editor	<b>Poor</b> Code is written in PHP	<b>Poor</b> A monolith with a proprietary format



# Platform Architecture – decoupled CMS architecture



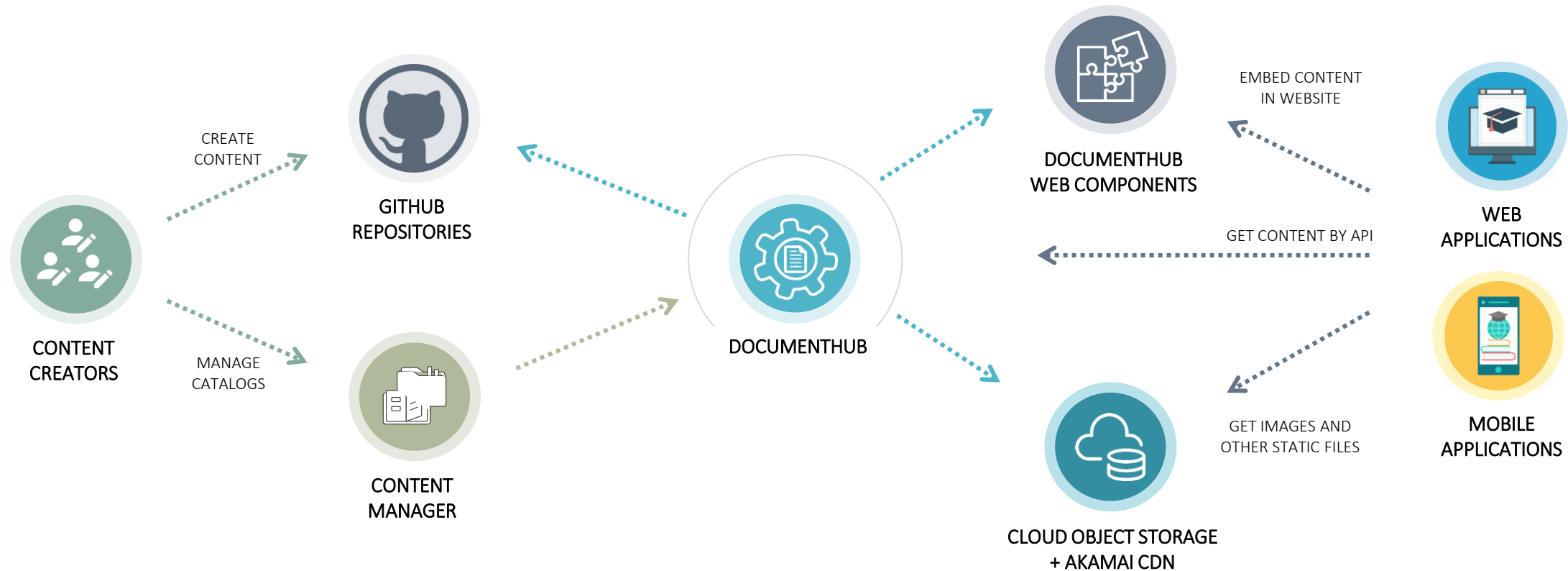
# Platform Architecture

1

Manage the content in GitHub or application

2

Show the content on web or mobile



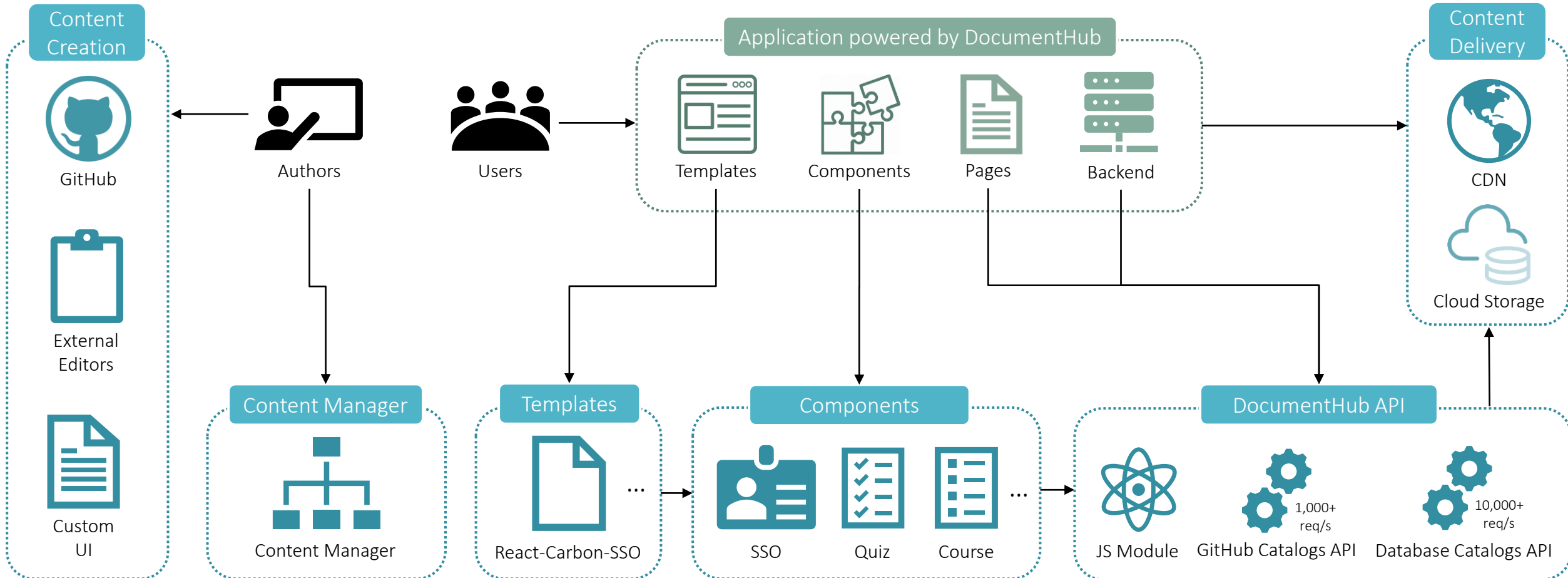
Content Creation

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**.



## 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.



## 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**.



## 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**.



## 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.

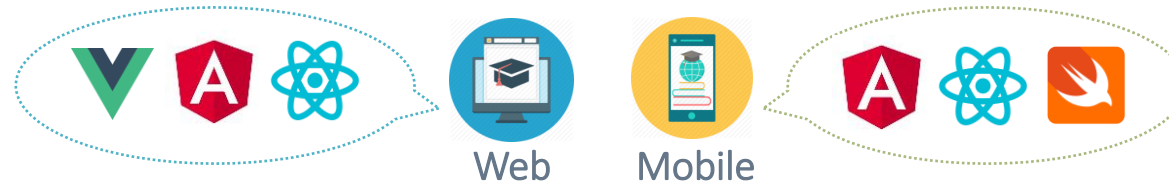


## 99.9% uptime

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

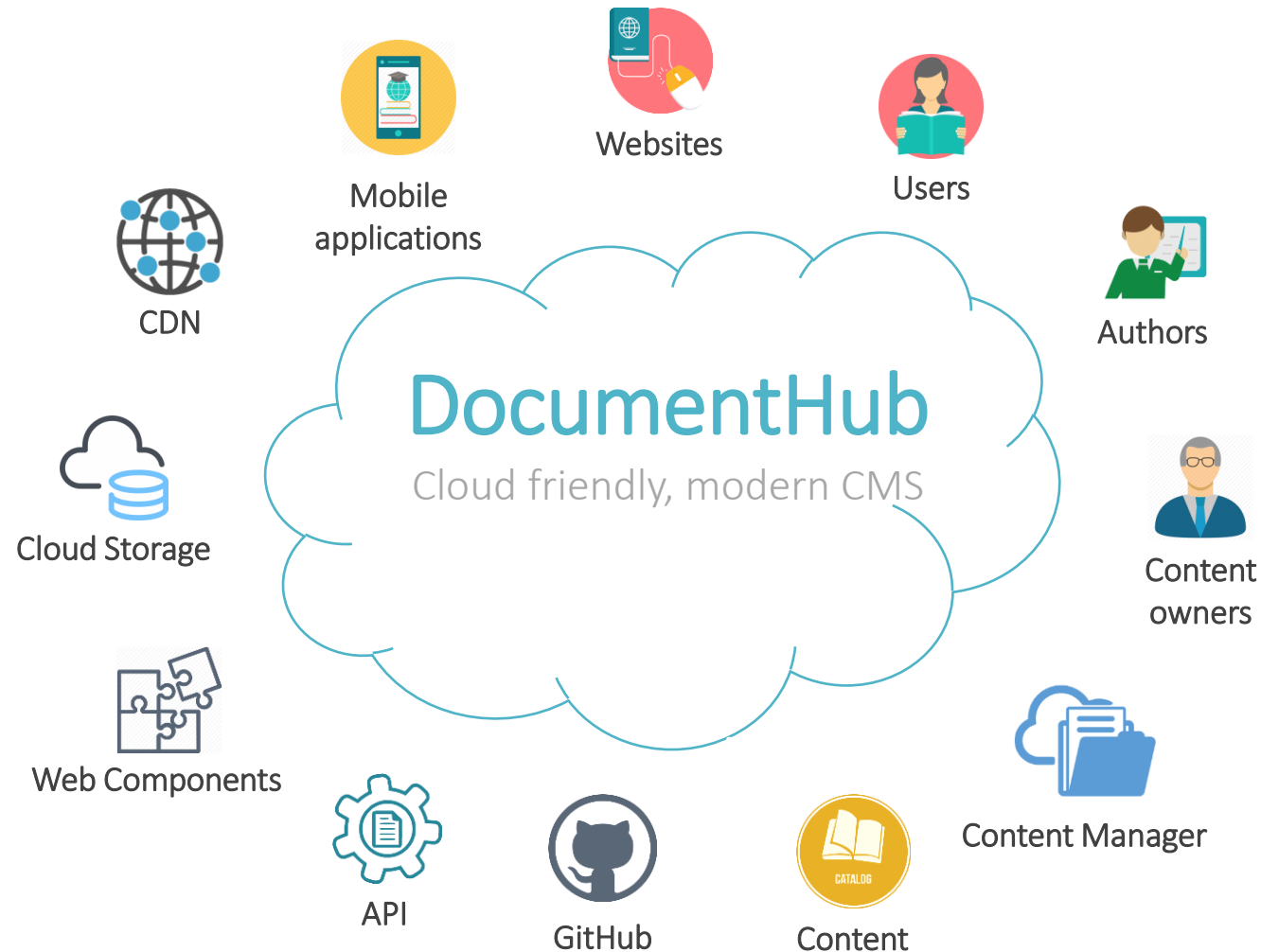
Easy embed courses,  
quizzes, articles and  
other components



CDN and Cloud Object Storage

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

# Summary



Business contact: [laksri@us.ibm.com](mailto:laksri@us.ibm.com)

Technical contact: [florin.dumitrescu@ro.ibm.com](mailto:florin.dumitrescu@ro.ibm.com)