Watson Content Hub
Content Management in the Cloud

Dr. Thomas Stober
STSM, IBM
Please Note:

- IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM’s sole discretion.

- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.
Key Concepts

Authoring and Publishing
**Key Concept: Content and Asset Publishing**

- **Authoring**
  - Information Architects
  - Content Authors
  - Designers
  - Business Owners
  - Templates/Layouts
  - Content Types
  - Authoring UI

- **Publishing**
  - Pre-rendering based on popular HTML templating engines
  - URL Rewrite

- **Delivery**
  - Users
  - CDN
  - Replication
  - Caching
  - Availability
  - Optimization for delivery

---

**Authoring System**

**Delivery System**

**Data Model**

© 2017 IBM
Key Concept: Content and Asset Publishing

Flexible, powerful and content managed micro services allow content authoring, publishing and run time capabilities to be easily consumed, maintained, scaled, updated and perform to expectations.
Defining the Content Model

- The content model consists of:
  - Content types
    - For example article, blog post, product, event, hero banner
    - This is where you define the elements (fields) for a type of content
  - Taxonomies and categories
    - Pre-defined choices for identifying content or assets
  - Image profiles
    - Image sizes for different “renditions” – thumbnail/full size, smartphone/tablet/desktop, etc.
Content types

- A content type defines the elements (fields) for a specific type of content
- Article, product, blog post, hero banner, etc.
- Element types include: Text, Image, Video, File, Category, Number, Link, Toggle
- Editing behavior is customizable
Taxonomies and categories

- Each taxonomy defines a set of categories, including nested child categories
- A category can be associated with a category UI element for a content type
Image profiles

- Add a “rendition” for each size you want to have for an image
- Image elements in a content type can specify an image profile to use
Renditions are automatically generated when the author creates content.

Cropped images are automatically generated.

Simple editing tool to modify crop position.
Search-centric interface

Filtering and dynamically generated facets help you find your stuff without folders

Search-centric UI with combined type-ahead and faceting
Uploading images and cognitive tagging

- Images and documents are automatically tagged by Watson when uploaded
- Applies to images and documents (PDF, DOC)
- Saves you time organizing and finding content

Add your own tags

10,000s of pre-defined tags
Creating and editing content

Choose a content type from “Compose new content” menu

Element editing behavior is defined by content type – required, date style, text field size, etc.
Content lifecycle model

- Content status values:
  - Draft: your content is ready to be composed or edited
  - Ready: your content is published and available on the content delivery network
  - Retired: your content is no longer required, but has not been deleted

Content Items are either DRAFT, READY or RETIRED
Multi-Lingual Aspects

- Content Items can specify their language
- Search support content-specific queries
Publishing transforms content into a consumable representation for delivery

- While the authoring environment is optimized for content management, the delivery system offers a representation which is highly optimized for consumption by arbitrary applications, e.g. a single page app or a mobile app

- Publishing services aggregate and transform content from authoring to delivery. This transformation includes:
  - Push content and assets to content delivery networks (Akamai)
  - Pre-render content based on popular HTML templating engines and store the resulting markup as static artefacts on the delivery system
  - Resolve dependencies between artefacts and rewrite URLs
  - Manage schedule and life-cycle of the published artefacts (*)

(*) these capabilities are intended to be delivered in future updates of IBM Watson Content Hub
Publishing: Dealing with Resources and Assets

**Assets**
- Digital Assets
- Referenced by managed content
- Maintained e.g. by image contributors

**Resources**
- Opaque Resources
- Nested folder structure
- Maintained by Front-end-developer
- Implement visual content design (HTML, CSS, JS, Icons, ...)

**Content**
- Structured & Managed
- Maintained by authors
- Typed & indexed
- Plain Data (no design)
- Embedded images, ...

---

**Publishing Components**
- Delivery Resource Service
- DAM Resources (managed)
- Delivery Resource Service
- Resources (unmanaged)
- Delivery Content Service
- Content (managed)
- Search Service

**Delivery Runtime (local or CDN)**
- Resources + Assets + Generated HTML
- HTTP Caching Policies
- Condensed Content
- Search Index

---

**Copy Resources**
- Copy referenced Assets
- Copy Resources
- Select & Transform Asset DAM
- Generate URLs for DAM

**Update Index**
- Update Index
Publishing: Dealing with Templates and Markup (*)

- Apply (Handlebar) template to content
  - Server-Side transformation into static (Cacheable) artifacts

MTL

<table>
<thead>
<tr>
<th>Heading</th>
<th>Heading</th>
<th>Heading</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</td>
<td>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</td>
<td>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</td>
</tr>
</tbody>
</table>

Hello, world!

This is a template for a simple marketing or informational website. It includes a large callout called the hero unit and three supporting pieces of content. Use it as a starting point to create something more unique.

Spacedrive

Hello, world!

This is a template for a simple marketing or informational website. It includes a large callout called the hero unit and three supporting pieces of content. Use it as a starting point to create something more unique.

MTL

JSON Context

```
{“title”:”Sleekfit”}
```

(Static) HTML Document or fragment

```
<h1>Sleekfit</h1>
```

(*) these capabilities are intended to be delivered in future updates of IBM Watson Content Hub
Key Concepts

Infrastructure
Integrating Akamai Content Delivery Network

- Delivery of published content and assets via CDN is tightly integrated into the IBM Watson Content Hub architecture
  - Worldwide distribution of content and assets for fast access by end users
  - Managing caching policies (e.g. expiration)
  - Configuration of Akamai services (Edge Configuration Service)
  - Manage Custom Domain Names for individual tenants (*)
  - Metering and Billing of Watson Content Hub usage
  - Security gateway at the “edge” (e.g. to prevent Denial of Service Attacks)
    - Akamai Edge is the main entry point/proxy into the system and dispatches between cached resources stored by the CDN Netstorage and dynamic API calls to Content Hub microservices

(*) these capabilities are intended to be delivered in future updates of IBM Watson Content Hub
Exploiting Web Server Proximity

- Content and Assets are distributed from the Origin-Servers to Edge Servers in proximity to the end-users to accelerate response times.
SOLR Search Services

All features are available via the message bus and HTTP.

Public Authoring and Delivery APIs use facades on top of the same core search service.

Potentially multiple Solr Cloud setups and multiple collections per tenant.
Key Concept: Modern Microservices Architecture

Authoring
Content, Types, Assets, Categories, Designs
Shell UI
Search
Sites
Page Assemblies

Publishing
Publishing Manager
Core Publishing
Rendering
Preview

Delivery
File Storage Proxy
File Access
Search
Content Access
Targeted Content

Infrastructure Services
Service Registry, Message Bus, Cloudant, Monitoring, Logging, Tenant & User Services, Configuration Service

Authoring UI
Command Line Dev Tools
End User UI
Content Services API

Akamai, Gateway, Dispatcher, Login
Key Concept: Resilient and Scalable Infrastructure

- SaaS, cloud only and multi-tenant
- Multi-active across multiple datacenters
  - worldwide distribution, disaster recovery and fail-over
  - Integration with content delivery networks
- Stateless, session-less micro-services
- Dynamic Scaling
- Secure platform
  - Datacenters comply with SSAE-16 and are ISO27001 certified
  - Secured by IBM Blue ID
- Full set of managed public APIs (JSON@REST)
  - Deliver content to any kind of consuming client
IBM Watson Content Hub Microservices are …

- independently developed, deployed and managed
  - one service can be deployed without affecting others
  - multiple versions of one service can be active at a given time
  - each service runs in its own docker runtime
- independently scaled, fully resilient and self healing
- independent of their deployment location, state-less and cache-less
- using their own persistence layer, which is completely independent from other services
- implemented in the best suited programming language for their purpose
  - Java, NodeJS, …
- exposed via public REST interfaces through an API gateway
  - dependency between micro services is limited to usage of exposed APIs
Example Docker Layer for each Service

Docker Layers for Java Services

- API
- Service A
- Java
- New Relic Agent
- Logging Java
- Consul Registration
- Ubuntu

Docker Layers for NodeJS Services

- API
- Service B
- NodeJS
- New Relic Agent
- Logging NodeJS
- Consul Registration
- Ubuntu

Shared infrastructure

- Marathon & Mesos
- MirrorMaker for Kafka
- Zookeeper for Configuration
- Vault for Protected data
- Consul for Service Registry
All Datacenters can handle all service requests around the world in multi-active topology.
Resilient and Scalable Infrastructure

- **Datacenters:** Softlayer
- **Orchestration:** Marathon / Mesos
- **Containerization:** Docker
- **Storage:** Cloudant (JSON) & Akamai NetStorage (Assets)
- **API Gateway/Management:** Papillion, IBM API Connect
- **Service Registry:** Consul
- **Message Bus:** Kafka
- **Search:** Solr
- **Configuration:** Zookeeper
- **HTML templating:** Handlebars
- **Logging:** Kafka, Graylog2
- **Monitoring:** New Relic
Authentication & User Management

For access to Authoring environment:
- Authentication via IBM Blue ID
- Users can be assigned to an individual tenant
- Roles can be Admin, Editor, Viewer, Manager

For Delivery Environment
- Public access to published artefacts
- Protection of content on delivery system will be added to the offering at a later point
Security and Compliance

- Data-in-transit protection
  - TLS (Version 1.2 or above)
  - IPsec or TLS VPN gateway

- Asset Protection
  - Watson Content Hub is EU-US Privacy Shield certified
  - Datacenter and Cloud Platform Operation are ISO27K certified and comply with SSAE-16

- Operational security
  - ISO27001 and SSAE-16 compliant operation in SoftLayer Datacenters.
  - Regular automated vulnerability scans during development, test and production to assess potential security impacts.
  - IBM internal security policies cover SaaS operation, secure engineering, and incident response (CSIRT).
  - Regular reporting of the status and approvals go up to the executive management level.
  - Management responsibility, annual training for all employees, internal and external audits ensure that policies are followed.

- Secure development
  - The Secure Engineering Framework at IBM includes education and awareness, project planning, risk assessment and threat modelling, security requirements, secure coding, test and vulnerability assessment, documentation, and incident response.
  - Product Security Incident Response teams ensure the timely identification, analysis, resolution and reporting of security vulnerabilities in IBM products. More information at https://www-03.ibm.com/security/secure-engineering/
Monitoring and Alerting: New Relic
Why should a partner or customer care about details of IBM Watson Content Hub infrastructure?

- **Simplicity**
  - Get started quickly using public APIs which are based on common patterns

- **Speed**
  - Frequent (daily/weekly) feature roll outs vs quarterly/yearly

- **Quality**
  - Single code stream per service version improves maintenance and test

- **Secure**
  - Datacenters comply with SSAE-16 and are ISO27001 certified

- **Performance and Resilience**
  - Auto-Scalability and Zero Downtime

- **Costs**
  - Operational efficiency reduces costs and leads towards an economy of scale
  - → true Software as a Service
Consuming Watson Content Hub
Using IBM Watson Content Hub

Fully-documented APIs on developer.ibm.com/api

Swagger API documentation

Sample API usage

Samples
Accessing Tenant Information (for your tenant)
Accessing Path Information (for a content item)
Accessing Content from authoring environment (Authentication required)

- Access a resource delivered from Content Hub using API layer https://www.digitalexperience.ibm.com/api/authoring/v1/resources/049440bd2c2b25b9a22e14387cd76af

- Access a content Item as JSON data from Content Hub API https://www.digitalexperience.ibm.com/api/authoring/v1/content/fadf705f-6e44-4918-91f0-d5573dd1a389

- Access selected content items as JSON data from Content Hub based on a search query API https://my.digitalexperience.ibm.com/api/552fd080-e4c6-4aca-9f5b-b2d618c0d617/authoring/v1/search?q=*&wt=json&fq=type%3A%22Product%22&fq=classification:%28%22content%22%29&fq=tags:%28%22kitchen%22%29&sort=lastModified%20desc
Accessing published Content from delivery environment (Public)

- Access a HTML page hosted and delivered from Content Hub via Akamai CDN
  https://my.digitalexperience.ibm.com/552fd080-e4c6-4aca-9f5b-b2d618c0d617/tstober/index.html

- Access a resource delivered from Content Hub via Akamai CDN
  https://my.digitalexperience.ibm.com/552fd080-e4c6-4aca-9f5b-b2d618c0d617/dxdam/ad/adb020c9-4140-4301-9435-15e449e87108/2013_53_D012_100.jpg

- Access a resource from Content Hub API
  https://www.digitalexperience.ibm.com/api/delivery/v1/resources/049440bd2c2b25b9a22e14387cd76afb

- Access a content Item as JSON data from Content Hub API
  https://www.digitalexperience.ibm.com/api/delivery/v1/content/fadf705f-6e44-4918-91f0-d5573dd1a389

- Access selected content items as JSON data from Content Hub based on a search query API
  https://my.digitalexperience.ibm.com/api/552fd080-e4c6-4aca-9f5b-b2d618c0d617/delivery/v1/search?q=**:*&wt=json&fq=type%3A%22Product%22
Use the powerful SOLR-based search API

- With one simple REST call you can retrieve a set of documents from search
- Search options include:
  - Filtering: by type, category, text match, status, and more
  - Ordering: order by any field
  - Number of rows and start row
- Search for content, assets, types, or taxonomies
Consuming Content using a JavaScript SPA

- Article Carousel:
  List all content in WCH built by using the Article Content Type
Consuming Content using a JavaScript SPA

// Base URL for APIs - replace {Host} and {Tenant ID} using the values available from the "i" information icon at the top left of the WCH screen

const baseTenantUrl = "https://{Host}/api/{Tenant ID}"
const serverBaseUrl = "https://{Host}

// Services used for this sample
const searchService = "/delivery/v1/search"

// search parameters for retrieving all content items of content type "Article"

const searchParams = 
"q=":"*&fl=id,document&wt=json&fq-type%3A%22Article%22&fq=classification:(content)&sort=lastModified%20desc"

function showContent() {

var searchURL = baseTenantUrl + searchService + "?" + searchParams;

var reqOptions = {
xhrFields: { withCredentials: true },
dataType: "json",
url: searchURL,
};

$.ajax(reqOptions).done(function(json) {

// generate HTML for carousel indicators and items from the search result list

var indicators = []; // carousel indicators
var innerItems = []; // the carousel items

$(json.documents).each(function(index, item) {

// the entire content item is available in the "document" field as a JSON string

var docJson = $.parseJSON(item.document);

// console.log('docJson: ', JSON.stringify(docJson));

var elements = docJson.elements;

// for the first indicator and the first item, set the "active" class

var activeClass = indicators.length == 0 ? ' class="active"' : ' class="item active"';
var activeItemClass = indicators.length == 0 ? ' class="item active"' : ' class="item"';


});
Using Watson Content Hub to host web applications and resources with CDN delivery

• Any HTML, CSS, JS, images, and other assets can be uploaded and then served from the integrated CDN
• You can use this to host Single Page Applications (SPA’s) or any web resources
• Example – Angular sample SPA
• Use the following command to push web resources to Content Hub for CDN delivery:
  • wchtools push -w
WCHtools: Command line tooling for Watson Content Hub

- Use this simple tool to upload (push) and download (pull) content, assets, and content model artifacts from Watson Content Hub
- Use it to install sample packages or pull authoring artifacts for archiving locally
- Use it for bulk upload of assets such as images, with automatic Watson cognitive tagging
Retrieve and Render Content using Handlebar Templates

- Define a static HTML page
- Retrieve selected content items or even the result of a search query
- Convert the content JSON-representation to a to a "templated" page using Handlebars references to dynamic content

Hello, world!
This is a template for a simple marketing or informational website. It includes a large callout called the hero unit and three supporting pieces of content. Use it as a starting point to create something more unique.

Welcome To WCH
Manage your content on the cloud with simplicity and flexibility. Customise the content model and compose rich content, that is hosted in our Content Delivery Network for optimal performance and scalability. There is no restriction to where and how you deliver content.

Search for latest three Article content items with tag “top-articles”

The Latest VR Headsets Will Take You on an Incredible Ride
Learn about how the new batch of headsets that will let you move around in virtual surroundings and give you experiences that are unforgettable.

Rock on With the Summer Concert Line up Review
Our rock captain has been on a summer swing and ready to deliver you the tips on where to catch the best tunes.

The Most Amazing Places to Catch a Wave in the Entire World
Travel to the best surf spots to enjoy long sunny days and endless wave rides. Find out where it is best for both high-tide and low-tide without thinking twice about timing.
Consuming content from web application and mobile app

Web application using “articles” content

iOS native app using same content
More Information for Developers

- Code samples: https://github.com/ibm-wch
Summary
For more information on Watson Content Hub

30-day free trial
Thank you!