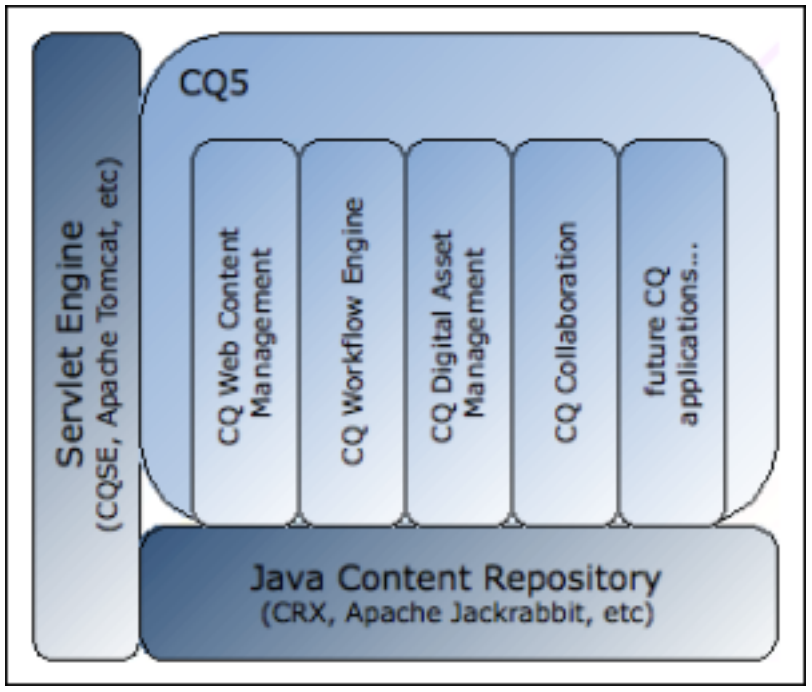
ADOBE CQ5 TUTORIAL

Interview Material with Evolution in Web Content World

CQ5 Architecture

An Overview of CQ5

The following diagram illustrates the interrelationship between CQ and other operational elements; which may be products from Day Management AG, or their third-party equivalents:



Servlet Engine

The Servlet Engine acts as the server within which each CQ (and CRX if used) instance runs as a web application.Any Servlet Engine supporting the Servlet API 2.4 (or higher) can be used.Although you can run CQ WCM without an application server, a Servlet Engine is needed. Both CRX, and therefore CQ WCM, ship with Day's CQSE (CQ Servlet Engine), which you can use freely and which is fully supported.

ABOUT ME

Anav Mahajan

Java Content Repository (JCR)

A Java Content Repository uses the JSR-170 API to access the content repository using Java, independent of the physical implementation. **JCR** is the **J**ava **C**ontent **R**epository standard, also known as **JSR-170** after its Java Specification Request.A repository effectively consists of two parts:

- A Web application that offers the JSR-170 compliant API and temporary data storage (in the form of the session).
- A Persistence Manager with persistent data storage, such as the file system or a database.

Content Repository Extreme (**CRX**) is Day Management AG's own repository product. See the CRX documentation for more details; including direct access using WebDAV, CIFS, File Vault etc.

3.2 years of IT experience in development, migration, integration and maintenance projects primarily using Migration and Content Management technologies, tools, frameworks and libraries including Adobe Cq5,Java,J2ee,Sharepoint,Word etc

Personal Links

Linkedin Profile
View Full Profile →

BLOG STATS

120,368 hits

CQ5

The common foundation of the CQ5 platform provides a basis for the interoperability and seamless integration of all CQ applications. This is available to both:

- the applications that are integral to CQ itself
- any customized applications developed for the CQ5 platform.

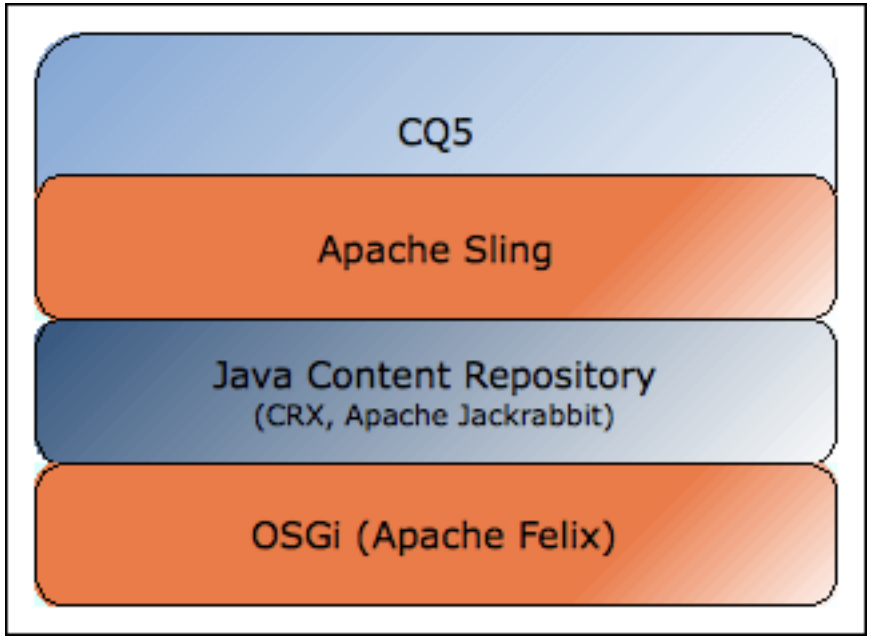
CQ WCM (Web Content Management) and the CQ Workflow Engine were the first applications developed to exploit the advantages of CQ5. CQ DAM and CQ Social Collaboration are now available and other Day products will be developed in the near future

The Technology Stack that CQ5 is

based on

CQ5 is based on new technologies including:

CQ5 Technology Stack



Apache Sling

Apache Sling is a web application framework for content-centric applications, using a Java Content Repository, such as Apache Jackrabbit or CRX, to

store and manage content.Sling:

- is based on **REST** principles to provide easy development of content-oriented applications.
- is embedded within CQ5.
- is used to process HTTP rendering and data-storage requests which assemble, render and send the content to a client (i.e. the *new delivery*).
- maps Content objects to Components (which render them and process incoming data).
- comes with both server-side and AJAX scripting support.
- can be used with a range of scripting languages, including JSP, ESP and Ruby.
- started as an internal project of Day Management AG .

Interview Questions

User Survey

Website Feedback

TOP CQ5 Web Links

- has been contributed to the Apache Software Foundation.

> **Note**
>
> See http://incubator.apache.org/projects/sling.html for more information.

OSGi (Apache Felix)

CQ5 is built within an application framework which is based on the OSGi Service Platform Release 4.OSGi technology

- "is the dynamic module system for Java™."
- comes under the classification Universal Middleware.
- "provides the standardized primitives that allow applications to be constructed from small, reusable and collaborative components. These components can be

RECENT POSTS: ADOBE CQ5 TUTORIAL

(No Title)

Introducing CQ In-Depth JSR-170 and the JCR API JSR 170 is the Java Specification Request for the Content Repository for JavaTM

composed into an application and deployed."

- OSGi bundles can contain compiled Java code, scripts, content that is to be loaded in the repository, and configuration or additional files, as needed.
- allows the bundles to be loaded, and installed, during normal operations. In the case of CQ5, this is managed by the Sling Management Console.

Apache Felix has been used to implement this framework.

- "Apache Felix is a open-source project to implement the OSGi R4 Service Platform, which includes the OSGi framework and standard services, as well as providing and

technology API. Specification lead is held by Day Software AG.The JCR API package, javax.jcr.* is used for the direct access and manipulation of repository content. CRX is Day's proprietary implementation of the JCR and […]

(No Title)

Adobe CQ5 Installation Generally, when you set up Adobe CQ, you need to set up an Author and a Publish instance – see Author and Publish Environments for further details on the two types of environment. Installation

supporting other interesting OSGi-related technologies.”

Java Content Repository (JSR-170 API)

A JCR uses the JSR-170 API to access the content repository using Java, independent of the physical implementation.

Inside CQ5

CQ5 forms a stable platform for content-centric applications such as CQ WCM:

CQ5 Internal Layers

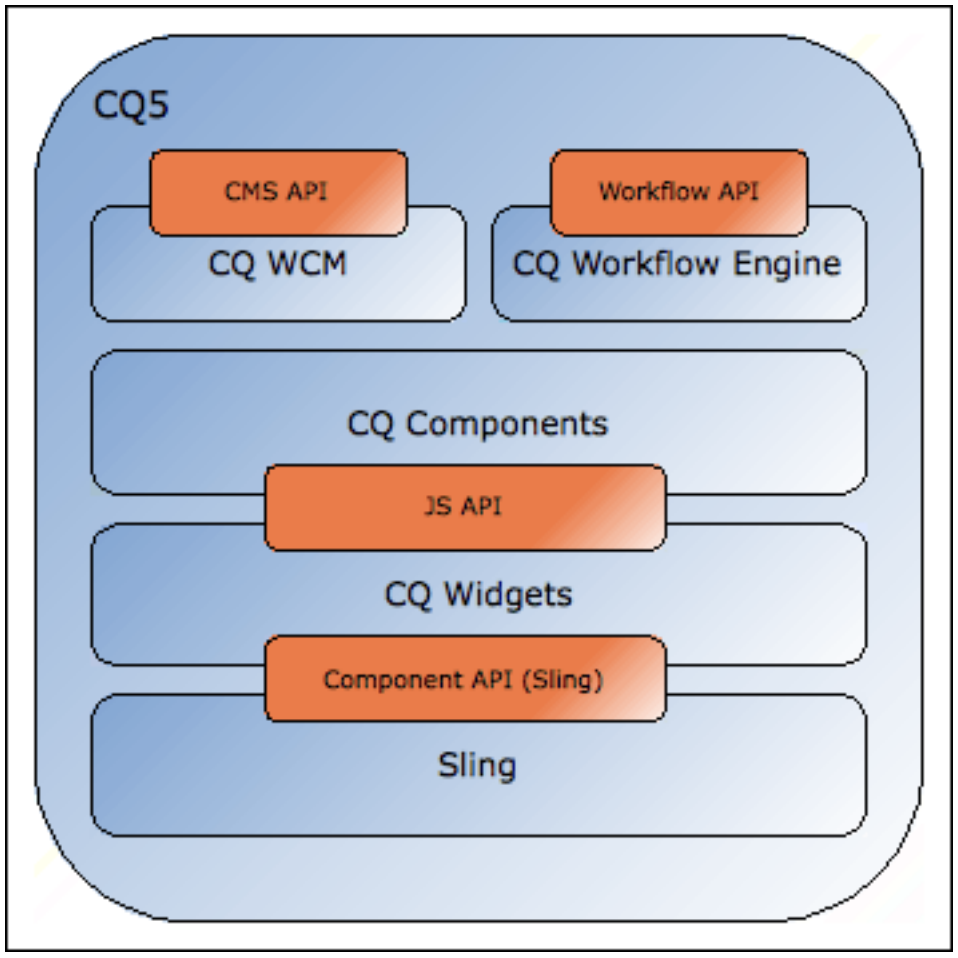
procedures for these are described in Installing an Author instance and Installing a Publish instance. If you need […]

Introducing CQ5 Video School

Now https://adobecq5interviewquestio comes up with CQ5 Video School .Go and Start your CQ5 Video Tutzzz.This Video Tuto. includes CQ5 Introduction,installation,Compone and other Related Videos. Click Here For CQ5 Video School

Introducing CQ5 Downloads

This CQ5 Downloads provide information about the architecture ,Development,Usage and Management of CQ5, together with the concepts and technologies used.Start Your Download Now!! Click Here For



CQ WCM

Web Content Management within the CQ5 platform allows you to generate and publish pages to your website..

CQ Workflow Engine

The CQ Workflow Engine is a powerful and

(No Title)

(No Title)

Manage and deliver digital content

The CQ Workflow Engine is a powerful and easy to use process engine that can be used by all applications running on the CQ5 platform. A Java API and RESTful HTTP interface is also provided for access by applications outside CQ5.Within CQ WCM workflows can be used to control the process of generating and publishing content, which are often subject to organizational processes, including steps such as approval and sign-off by various participants.

CQ Components

Components provide the logic (code) to render content. They include both templates and specific components such as Text with Image, Column Control and Subtitle amongst others. Components are based on a

with ease, across channels Adobe® CQ is the foundation of the Adobe Experience Manager solution. It provides digital marketers with easy-to-use, web-based applications for creating, managing, and delivering personalized online experiences. Adobe CQ provides out-of-the-box integration with other products in Adobe Marketing Cloud. Web Content Management An easy-to-use application […]

What is CQ5 and will Adobe kill its Golden Goose?

What is CQ5? CQ5 is Adobe's flagship Content Management System and the leading CMS in the MARKET. Like any CMS you

combination of widgets, replacing the CFC from Communiqué 4.

CQ Widgets

Widgets are the basic elements used to implement a specific user function, often the editing of a piece of content; they include buttons, radio-boxes, dialogs, etc.

Apache Sling

The Component Framework (Sling) provides the underlying mechanisms for rendering content.

Server Startup Sequence

CQ5 has drastically reduced startup time as compared to Communiqué 4. For CQ WCM

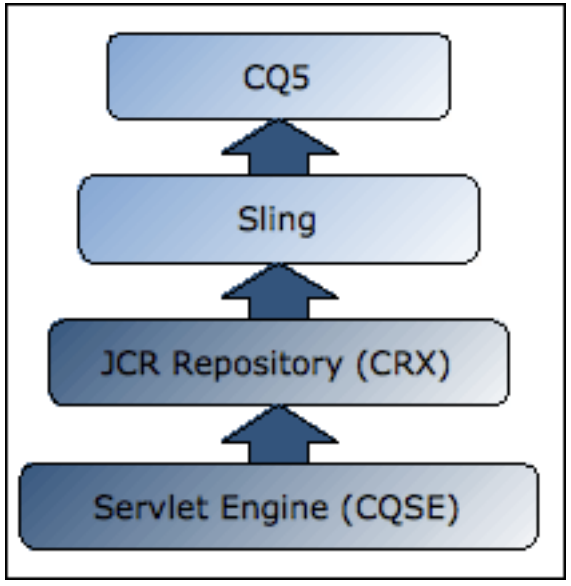
can use it to build and maintain your web presence, but more importantly you can use it to capture your online visitor behaviour and convert customer interest into sales. It works on multiple platforms […]

CQ5 Polling

Video helps you to Poll

(straight out-of-the-box) the startup time is now in the order of 5 seconds.The elements required for CQ WCM are started in the following sequence (bottom-up):

Server Startup Sequence



CQ WCM Environments and how they interact

Author and Publish Environments

A CQ WCM installation usually comprises of multiple instances, used for different purposes. Content, including code and other resources held in the repository, can be replicated (copied from destination to source) using a HTTP, or HTTPS, connection.

A production environment often consists of two different types of instances:

author

This is the environment where you, and your colleagues, will input your

content and administrate the system.

publish

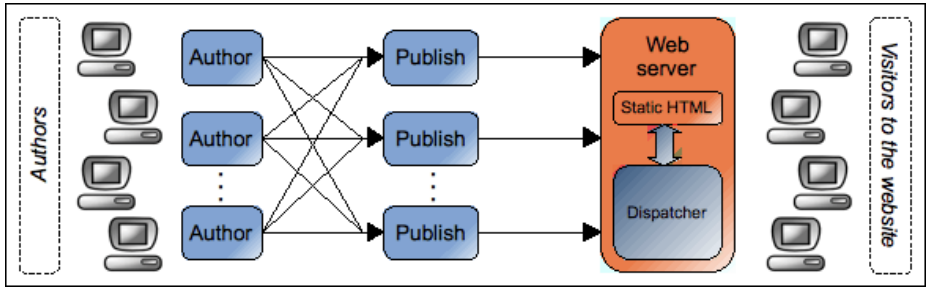
This environment holds the content you have made available to visitors to your website; be it public, or within your intranet.

Defining a CQ instance (or environment) as “publish” or “author” depends primarily on where in the overall system structure the environment is located and what its tasks are.

The following diagram gives an overview of typical configurations possible for the various CQ WCM environment. It shows how content flows from the authoring environments until it is available to be

accessed by visitors to your website. It also highlights the fact that to increase performance and availability it is common to combine several authoring and publishing environments to service a website.

CQ WCM Environments



Note

This diagram covers a range of possible configurations, with multiple environments of either sort. Depending on your configuration, each author environment can propagate content to one, or more, publish environments.

Author

Author is usually located behind the internal firewall as it is the environment where you and your colleagues will:

- administrate the entire system
- input your content
- configure the layout and design of your content
- activate your content to the publish environment

The author environment of CQ WCM is accessed using the **siteadmin**. Access to the content and functionality is controlled by authorization permissions assigned

to your user account.

Replication agents in the author environment(s) are used to publish (activate) content and functionality from the author to the publish instance:

- content to be published is packaged and placed in the **replication queue** (in the author environment)
- the content is transported to the publish environment
- the content is received and published

Publish

This holds the content which you

have made available to visitors to your website and is usually located in the Demilitarized Zone (DMZ).The content is dynamic, real-time and can be personalized for each individual user.**Reverse replication** is necessary from the publish environment, to return user input from a publish instance to the author, and then to any other publish, instance(s). A reverse replication agent in the publish environment places the input into an outbox, which is matched with replication listeners in the author environment. The listeners poll the outboxes to collect any input made and then distribute it as necessary. This ensures that any traffic from the publish to the author

environment is strictly controlled. Reverse replication is of particular significance for CQ Social Collaboration.

Static Web Server

For performance optimization it is possible to convert your dynamically published content (excluding any personalized parts) to static HTML, serviced by a static web server. Static web servers are very simple, but fast. Examples include Apache, and IIS.The Dispatcher can then be used in conjunction with the web server to realize an environment that is both fast and dynamic and with moderate hardware requirements.

Dispatcher

The Dispatcher helps realize an environment that is both fast and dynamic. It works as part of a static HTML server, such as Apache, with the aim of:

- storing (or "caching") as much of the site content as is possible, in the form of a static website.
- accessing the layout engine to retrieve dynamic content as and when necessary, but as little as possible.

Which means that:

- **static content** is handled with

exactly the same speed and ease as on a static web server; additionally you can use the administration and security tools available for your static web server(s).

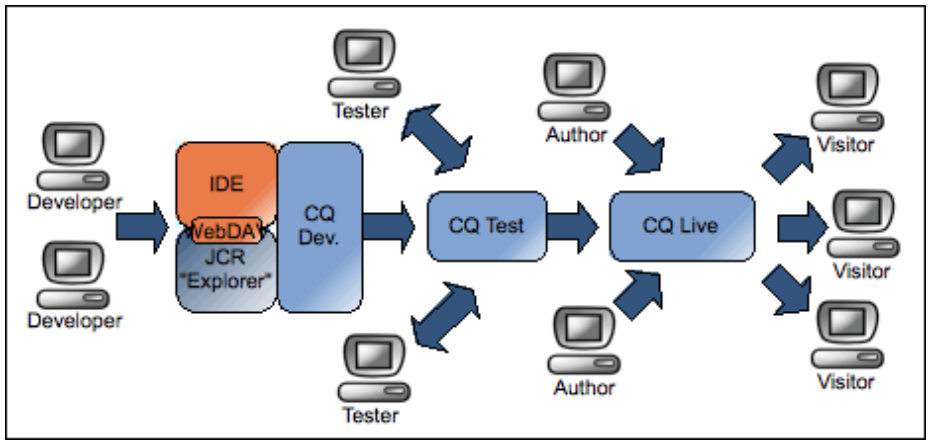
- **dynamic content** is generated as needed, without slowing the system down any more than absolutely necessary.

The Dispatcher contains mechanisms to generate, and update, static HTML based on the content of the dynamic site. You can specify in detail which documents are stored as static files and which are always generated dynamically.

Environment levels used within the full development cycle

As with any software, the projects developed using CQ WCM are subject to the development cycle. Therefore various environment levels are required to cover the development and testing phases. Depending on your requirements, each level may be configured for either only an author, or an author and publish environment.

Development and Test Environment Levels



Note

Multiple instances of each environment level can exist.

Development

Prior to authors registering their content in CQ WCM, the developers are responsible for developing and customizing the proposed website. They:

- develop and customize components

- realize the design within the website
- develop the necessary scripts to implement the required functionality of the website

The major development tools used are:

- an Integrated Development Environment.Day provides an Eclipse-based development environment, CQDE (the Communiqué Development Environment). It is also possible to use other IDEs, such as Eclipse or IntelliJ, for which plug-ins have developed to simplify their use for CQ

and to integrate them with the repository.

- a method of direct access to the Java Content Repository.In the case of CRX, the Content Explorer, Content Loader and other in-built tools are used.
- WebDAV or CIFS, which simplify access to the repository.

Depending on the scale of your system, the development environment can have both **author** and **publish** environments, or the test environment will be used for such functionality.

Test

After development, it is usual to have a Testing environment where you can access the new system, to test both design and functionality.This will often comprise of both an **author** and **publish** environment.Day provides a basis for automated GUI tests, together with some reference test scripts.

Live / Production

As discussed previously, the Live (or Production) environment comprises both:

- an **author**ing environment for the input of content
- a **publish** environment for

content made available to visitors to the website

Performance and Availability

Caching

There are two basic approaches to web publishing:

Static Web Servers

Are very simple, but fast.A static web server, such as Apache or IIS, serves static HTML files to visitors of your website. Static pages are

created once, so the same content will be delivered for each request. If a visitor requests a file (e.g. a HTML page), the file is usually taken directly from memory, at worst it is read from the local drive.This process is very simple, and thus extremely efficient, but does not cater for personalization or dynamic content.

Content Management Servers

Provide dynamic, real-time, intelligent content, but require much more computation time and other resources.CQ is such a server,

whereby an advanced layout engine processes the request from a visitor. The engine reads content from a repository which, combined with styles, formats and access rights, transforms the content into a document that is tailored to the visitor's needs and rights. This allows you to create richer, dynamic content, which increases the flexibility and functionality of your website.However, the layout engine requires more processing power than a static server, so this setup may be prone to slowdown if many visitors use the system.

The Dispatcher is used to combine the 2 philosophies, on the principle of:

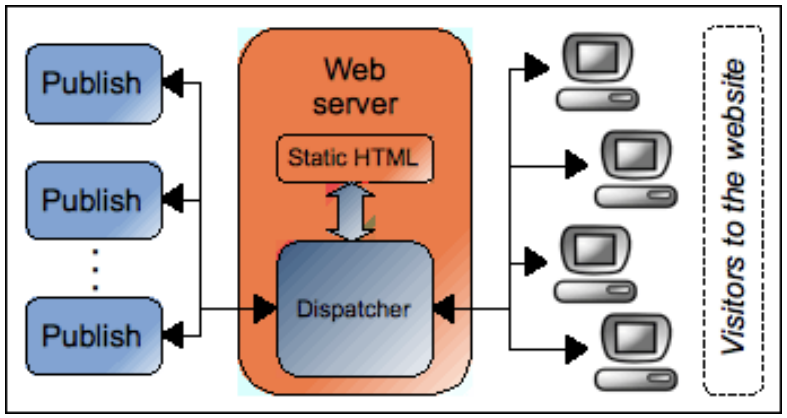
- if the cached document is newer, the Dispatcher returns it
- if older, the Dispatcher retrieves the current version from the appropriate CQ WCM instance (see also Load Balancing)

This allows you to take full advantage of a static web server, while retaining the capability to process dynamic content as and when necessary.

Load Balancing

The Dispatcher keeps internal statistics about how fast the web servers are processing documents. Based on this data, the Dispatcher estimates which server will provide the quickest response time when answering a request, and so it assigns the necessary request, and computational time, to that server.

Load Balancing



You gain:

Increased processing power

In practice this means that the Dispatcher shares document requests between several web servers. Because each server now has fewer documents to process, you have faster response times. The Dispatcher keeps internal statistics for each document category, so it can estimate

the server load and distribute the queries efficiently.

Increased fail-safe coverage

If the Dispatcher does not receive responses from a web server, it will automatically relay requests to the other server(s). Thus, if a server becomes unavailable, the only effect is a slowdown of the site, proportionate to the computational power lost. However, all services will continue.

Increased flexibility

You can also manage different websites on the same static

web server.

High availability CQ5

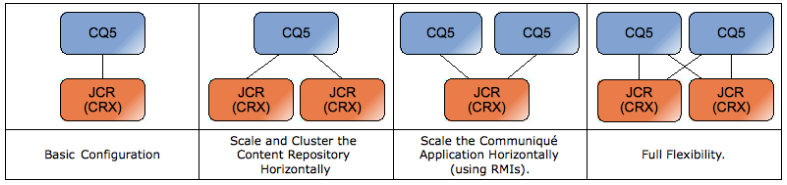
To increase availability, and performance, of your Production environment, it is common to combine multiple instances of the CQ WCM author and publish environments.

This concept can be extended to include multiple instances of your content repository. After you have installed a high-availability JCR solution, you can integrate it to ensure that your CQ WCM solution is protected against hardware and software failure. CRX Clusters are

supported for various persistence managers. See the CRX Clustering documentation and How To Cluster for further information.

There are various high-availability configurations possible (dependent on whether it is a publish or author environment) the basic principles of which include:

High-availability CQ WCM and JCR (CRX)



Authentication and

Authorization

Authentication

Authentication is the process of identifying, and verifying, a user.

The process of authentication and login can be broken down as follows:

1. Authentication information is extracted from the request. In CQ this is done by an *authentication handler*.
2. The authentication information is then checked to determine whether it is sufficient and/or correct. In

CQ this is performed by the *login modules*.

3. The appropriate response is initiated.

For CQ, initial authentication uses a standard HTML-login form in conjunction with the *Authorization Header Authentication Handler*. The HTML-form must have fields for the user name and password (the same field names must then be used by the Authorization Header Authentication Handler).

You can also use a similar form for controlled access to various areas of your website.

Authorization

Authorization determines whether a user is allowed to take action on specific areas within the system. For example, a user can be authorized to read or update a specific page.

Authorization is managed using a series of entities:

User

A user accesses a system using their user account. A user models either a human user or an external system connected to the system. The user account holds the details needed for accessing CQ; a key

purpose of an account is to provide the information for the authentication and login processes – allowing a user to log in.

Groups

A group is a collection of users and/or other groups. A change in the permissions/privileges assigned to a group is automatically applied to all users in that group. A user does not have to belong to any group, but often belongs to several.

Action

Actions are performed on a

resource. For example, a user can read, edit or delete a page, amongst other actions.

Permissions

A permission allows a user to perform an action on a given resource within the repository. Permissions are stored, and can be seen, at resource level within the repository.

Privileges

Privileges allow access to functionality available within the application; for example, replication of a specific path, or the ability to update the

page hierarchy (including creating new pages).

Resources

Resources define the functionality to be accessed.

Rate this:

 30 Votes

---

Share this:

Print Google Facebook Twitter 1 LinkedIn Email

Tumblr Reddit

Loading...

3 thoughts on “CQ5 Architecture”

Akhil Raj

FEBRUARY 18, 2015 AT 10:12 PM

Very helpful

5 1 Rate This

REPLY

chanduwala

APRIL 13, 2015 AT 1:17 PM

Awesome! Very Helpful!

3 0 Rate This

REPLY

Meena

JUNE 2, 2015 AT 6:39 AM

awesome dude great job..

2 1 Rate This

REPLY

Leave a Reply

Enter your comment here...

TOP RATED POSTS

Posts | Pages | Comments

All | Today | This Week | This Month

5/5 (3 votes)

What is CQ5 and will Adobe

TOP POSTS & PAGES

Interview Questions

CQ5 Architecture

CQ Introduction

Downloads

FOLLOW BLOG VIA EMAIL

Enter your email address to follow this blog and receive notifications of new posts by email.

Join 853 other followers

Enter your email address

kill its Golden Goose?

5/5 (2 votes)

Getting started with Adobe CQ5

5/5 (1 vote)

5/5 (1 vote)

Introducing CQ5 Job Portal.Apply Now

5/5 (1 vote)

ARCHIVES

Select Month

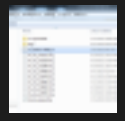
FLICKR PHOTOS

Adobe CQ5 Installation

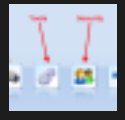
CQ5 Video School

CQ in-depth

CQ5 Template



CQ5 Application



CQ User Guide

Follow

BLOGS I FOLLOW

More Photos

CREATE A FREE WEBSITE OR BLOG AT WORDPRESS.COM. | THE HEMINGWAY REWRITTEN THEME.

Follow

Follow “Adobe