

Adobe Experience Manager Notes

Introduction

Adobe Experience Manager (AEM) provides a complete suite of applications for the Web Experience Management (WEM) of organizations.

3 modes of working:

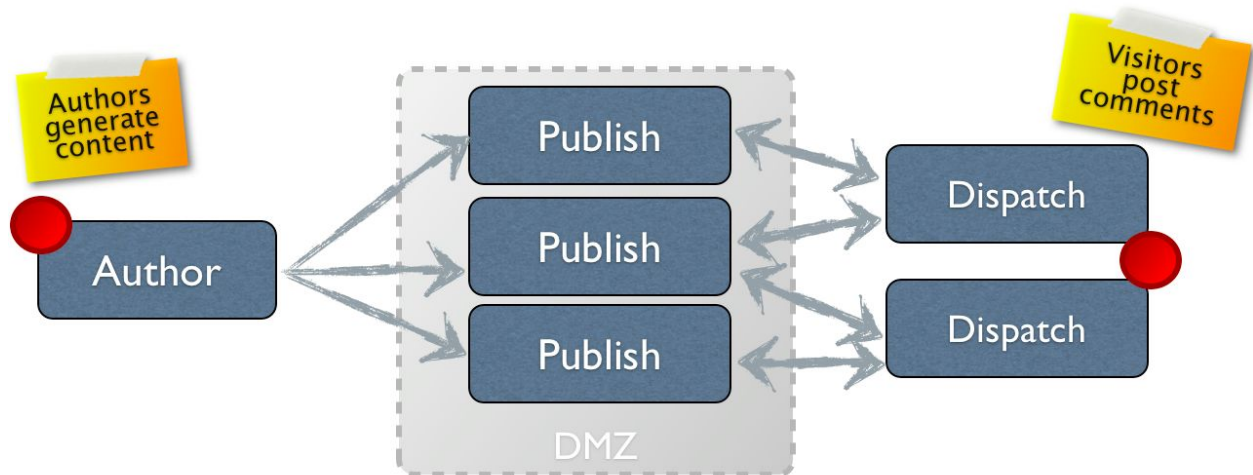
- author
- publish
- dispatcher

Is customer using the ecommerce module? If yes, with what hardware/software? IBM Websphere commerce server?

Author

The author environment provides the mechanisms for creating, updating and reviewing this content before actually publishing it:

An author creates and reviews the content (this can be of several types; for example, pages, assets, publications, etc) which will, at some point, be published to your website.



On the author environment the functionality of AEM is made available through two UIs. For the publish environment you design the entire look-and-feel of the interface made available to your users.

Publish Environment

When ready, the content is published to the publish environment. Here your pages are made available to your intended audience, according to the entire look-and-feel of the interface that you have designed. For a normal internet site, the publish environment is located inside the demilitarized zone; in other words, available to the internet, but no longer under the full protection of your internal network. On the published website visitors can also post comments on the individual pages or interact with the forums.

Dispatcher

<https://docs.adobe.com/docs/en/dispatcher.html>

Dispatcher is Adobe Experience Manager's caching and/or load balancing tool. Using AEM's Dispatcher also helps to protect your AEM server from attack.

The most common use of a Dispatcher is to cache responses from an AEM publish instance, to increase the responsiveness and security of your externally facing published website.

Why use dispatcher?

There are two basic approaches to web publishing:

- Static Web Servers: such as Apache or IIS, are very simple, but fast.
- Content Management Servers: which provide dynamic, real-time, intelligent content, but require much more computation time and other resources.

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:

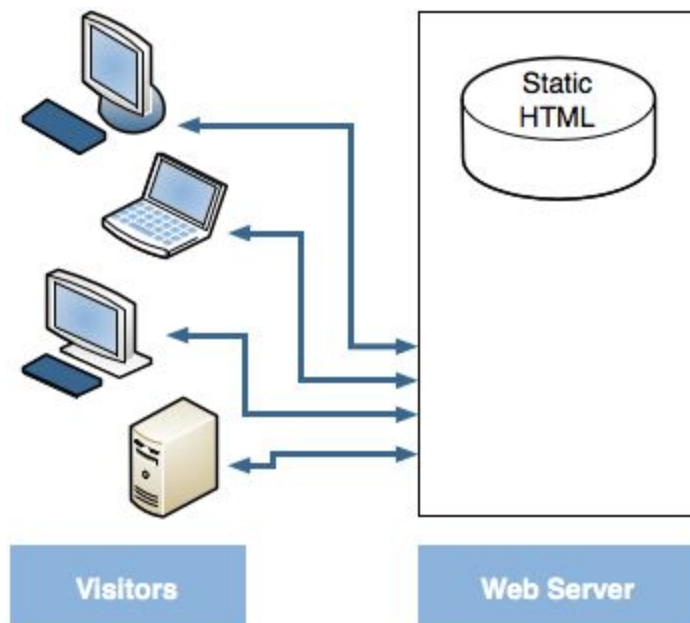
- storing (or "caching") as much of the site content as is possible, in the form of a static website
- accessing the layout engine as little as possible.

Which means that:

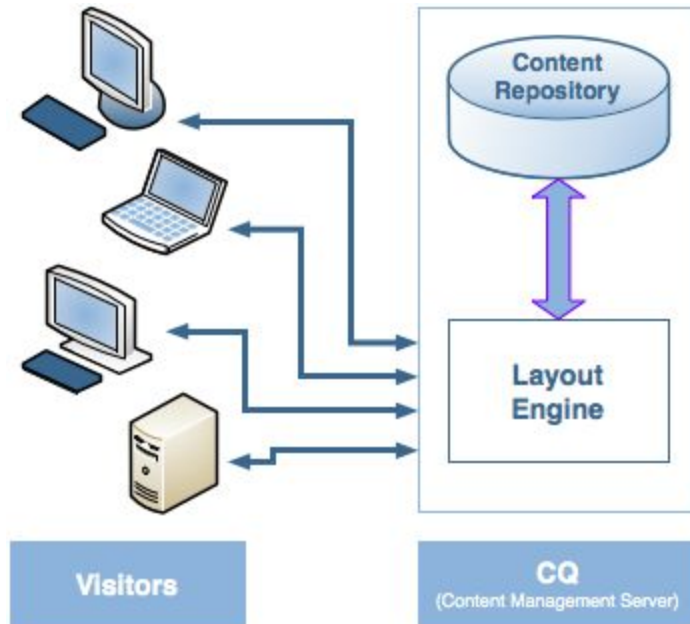
- static content is handled with exactly the same speed and ease as on a static web server; additionally you can use the administration and security tools available for your static web server(s).
- dynamic content is generated as needed, without slowing the system down any more than absolutely necessary.

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.

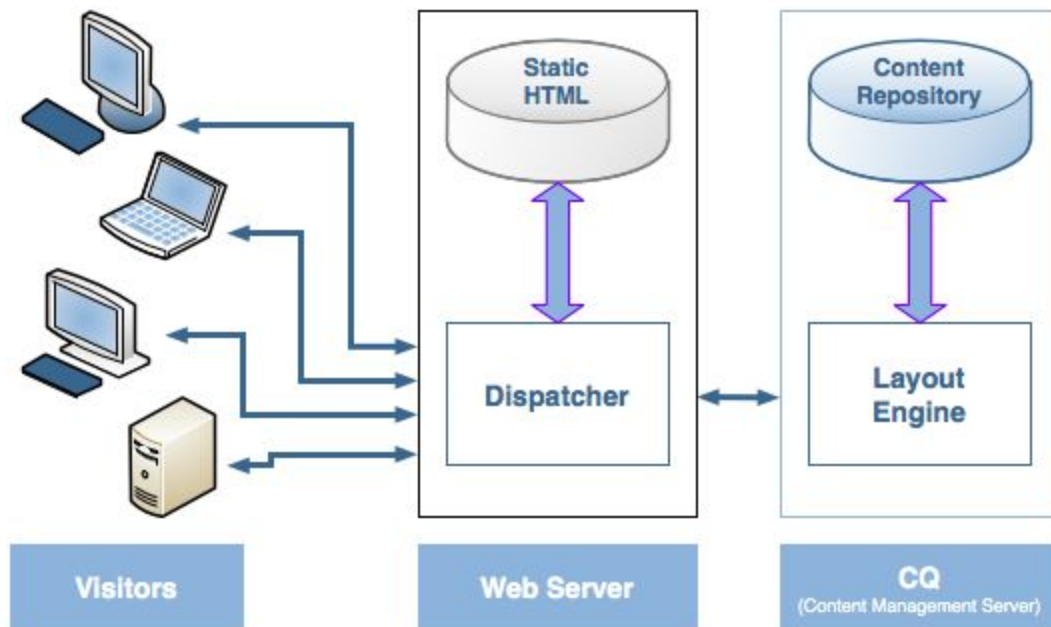
Static Webserver



CMS



CMS With dispatcher

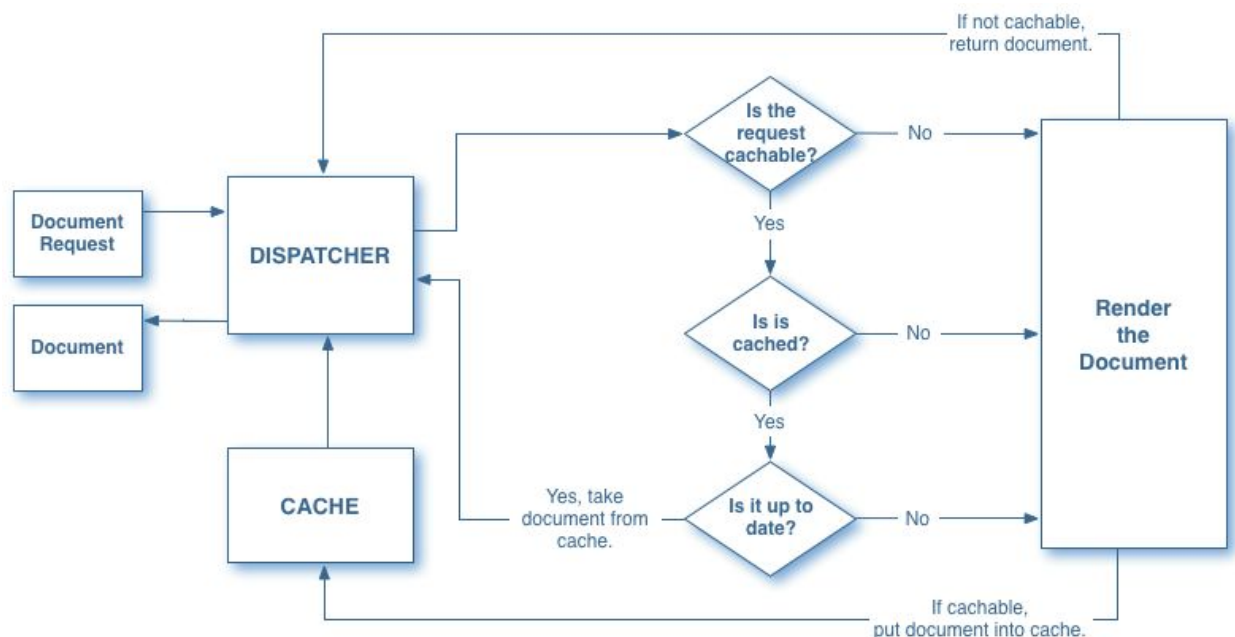


For caching, the Dispatcher module uses the web server's ability to serve static content. The Dispatcher places the cached documents in the document root of the web server.

Methods for caching {same as akamai cache purge v/s invalidation}

The Dispatcher has two primary methods for updating the cache content when changes are made to the website. Dispatcher has no time out. So no concept of TTL.

- Content Updates remove the pages that have changed, as well as files that are directly associated with them.
 - AEM sends a syndication request to the Dispatcher, which updates the cache accordingly.
 - Operates just like purge. Object is evicted but not replaced. New version is pulled only upon the next end user request.
- Auto-Invalidation automatically invalidates those parts of the cache that may be out of date after an update. i.e. it effectively flags relevant pages as being out of date, without deleting anything.
 - It works by playing with the statfile.
 - The Dispatcher has a list of files that are subject to auto-invalidation. When a document from that list is requested, the Dispatcher compares the date of the cached document with the timestamp of the statfile:
 - if the cached document is newer, the Dispatcher returns it.
 - if it is older, the Dispatcher retrieves the current version from the AEM instance.



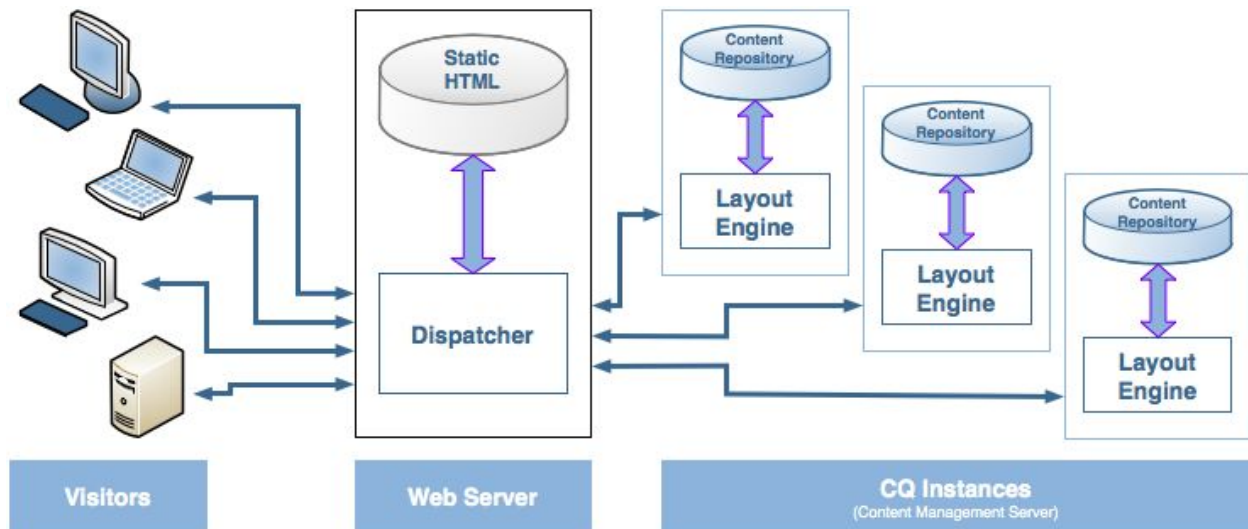
Default Caching rules for Dispatcher

The Dispatcher always requests the document directly from the AEM instance in the following cases:

- If the HTTP method is not GET. Other common methods are POST for form data and HEAD for the HTTP header.
- If the request URI contains a question mark "?". This usually indicates a dynamic page, such as a search result, which does not need to be cached.
- The file extension is missing. The web server needs the extension to determine the document type (the MIME-type).

- The authentication header is set (this can be configured)

Dispatcher with Load Balancing



Dispatcher performs load balancing in 2 modes:

- **Performance statistics:** Based on server response times. The computation can be controlled - eg: image display is different from HTML page generation
- **Personalized content (sticky connections):** When sticky connection is used, caching degrades significantly. So use with care. Session data is maintained only by the target CMS (CQ instance).

Dispatcher with Akamai

[ContentBuilder](#) API can be used as a REST API to control cache invalidation on Akamai.

Supported Webservers for the dispatcher

https://docs.adobe.com/docs/en/aem/6-0/deploy/technical-requirements.html#par_title_15

Troubleshooting info for Dispatcher with SSL

When Dispatcher receives an HTTPS request, Dispatcher includes the following headers in the subsequent request that it sends to AEM or CQ:

- X-Forwarded-SSL
- X-Forwarded-SSL-Cipher
- X-Forwarded-SSL-Keysize
- X-Forwarded-SSL-Session-ID

This could be something that customers can log to help in troubleshooting.

https://blogs.adobe.com/cqpost/files/2012/10/CQDispatcher_workflow_v2.png

Serve stale

<https://helpx.adobe.com/experience-manager/kb/ServeStaleContentOnError.html>

302s are not cached by dispatcher

<https://helpx.adobe.com/experience-manager/kb/RedirectPageNotCached.html>

Configuring dispatcher

<https://docs.adobe.com/docs/en/dispatcher/disp-config.html>

Akamai and CQ5 best practices:

<https://forums.adobe.com/thread/1088527>