

A Platform Approach to Enterprise Content Management with Eclipse Apricot, CMIS and Nuxeo Platform

Florent Guillaume, Director of R&D, Nuxeo
2011-09-23

Agenda

- What Is a Content Repository?
- Eclipse Apricot
- Nuxeo Enterprise Platform
- The Future

Who Am I?

- Florent Guillaume (twitter: @efge)
- Director of R&D at Nuxeo
- Architect of content repository technologies
- Apache committer, OASIS member
- Nuxeo: open-source content management
- Based in Paris / Boston / San Francisco

What Is a Content Repository?

What Is Content?



- **Everything** is content!
- **Unstructured**
 - Files, Images, Assets, XML, Binary streams
- **Structured**
 - Schema, Metadata, Business Data, Tables
- **Semi-Structured**
 - Files + Metadata, Web pages (assemblies, relations), Emails (attachments), Records Management

A Content Repository is Middleware

- **Between** Application and Storage Subsystem
 - Does not replace either
- **Persistence Service**
 - Stores structured and unstructured content
- **High-Level** Abstraction
 - Stop caring about storage details
 - Focus on your domain model and its objects

What a Content Repository is *Not*

- Not a **CMS** (WCM, ECM, ...)
 - A CMS is one application on top of a Content Repository
- Not a **filesystem**
 - If all you have is a filesystem, everything looks like a file
- Not an **ORM**
 - Not that granular, don't think in SQL terms
- Not just for **storage**
 - Provides Services, Domain Model / Business Model

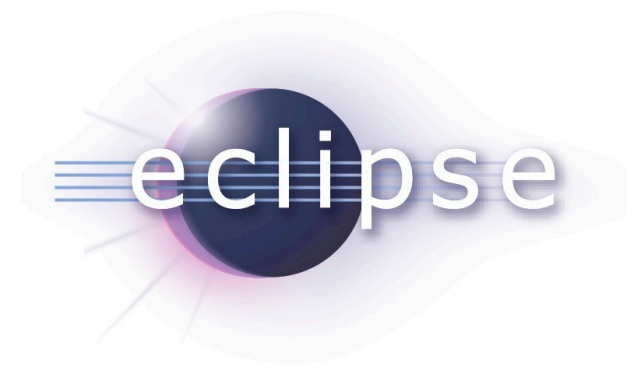
apricôt

Eclipse Apricot

Eclipse Apricot



- **OSGi** framework
- Under the **Eclipse Runtime** project
 - Currently in the *Incubation* phase
 - Mentored by Gary Xue (Actuate) and Cédric Brun (Obeo)
- Contributed by Nuxeo, from **Nuxeo Core**
- <http://www.eclipse.org/apricot>



What Is Apricot?

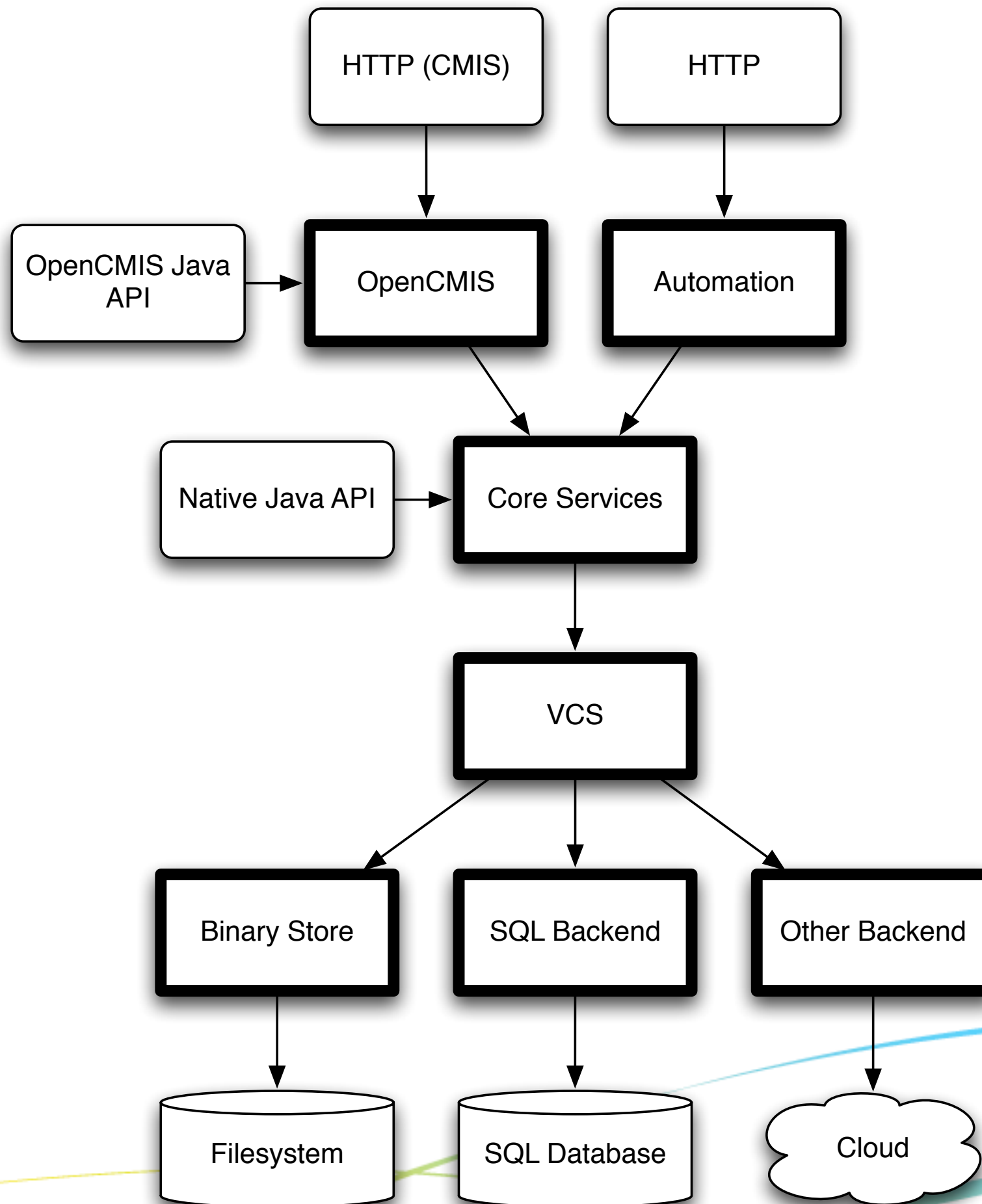
- **Content Services**
 - Types and schema definition, persistence, query, access control, locking, versioning
- **Web Support**
 - Expose content on the **web** through a fast **REST** framework extending JAX-RS and using FreeMarker templating
- **Content Automation**
 - Build **reusable** business logic based on **Commands/Operations**, accessible from UI frameworks and HTTP/REST

What Is Apricot?



- **CMIS**
 - OASIS Content Management Interoperability Services
- HTTP APIs (SOAP, AtomPub, Browser Bindings)
- OpenCMIS Java APIs
- <http://docs.oasis-open.org/cmisis/CMIS/v1.0/cmisis-spec-v1.0.html>





When to Use Apricot?

- Need to store Objects with **Properties**
 - And maybe **Files**
- Don't want to write SQL
 - But be able to fall back to SQL if really needed
- Need **Access Control**
- Need **Versioning, Queries, ...**
- Don't want to **reinvent the wheel**

Why Is Apricot Good?

- **Modular** — OSGi, extension points
- **Fast** — Efficient use of storage backends, caching
- **Safe** — Transaction-aware, hot backups
- **Scaling** — Clusterable, pluggable binary store (FS, SQL, AWS S3, ...), ready for NoSQL backends
- **Battle-tested** — In production for 4 years, thousands of deployments, originates from Nuxeo Core (used in Nuxeo DM, DAM, CMF)

nuxeo

Open Source ECM

Nuxeo Enterprise Platform

Platform Approach

- Platform to build **Content** applications
 - **Reusable** components and services
 - Comprises **UI** frameworks and building blocks
 - Ready to use **pre-built** applications (DM, DAM, CMF)
- Nuxeo platform is
 - **Open source**, 100% Java-based since 2006

Vertical Applications

Construction

Publishing

Government

Life Sciences

Correspondence Management

Contracts Management

Invoice Processing

Records Management

Packaged Products

Document Management

Digital Asset Management

Case Management Framework

Technical Product

Content Application Platform

A packaged development framework for building content-centric applications, is designed for java developers and architects.

Technology

Nuxeo Enterprise Platform

Complete set of components covering all aspects of ECM.
Extensible modular architecture designed for content

Marketplace
Solution Catalog

Studio
Customization &
Configuration

Support
with SLAs

Maintenance
Patches & upgrades

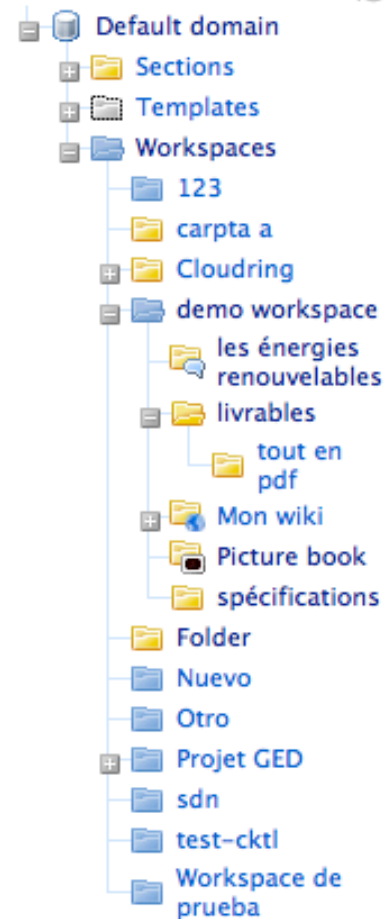
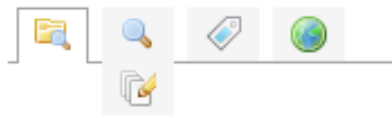
Nuxeo Connect
Subscription

Platform Modularity

- ECM projects usually require a lot of **customization**
 - Without compromising **upgrades** (we sell support!)
 - Without **forking** the source code
- For our Java platform it was a requirement to have
 - A **modular** framework
 - A **plugin** system

What Modularity Achieved

- Complex software platform
 - 200+ **bundles**, 210+ **extension points**, 490+ **contributions**
- Very flexible thanks to extension points
 - Easy to **contribute** new services and **customize** the default application (including UI and business logic)
- Running virtually everywhere
 - **JBoss**, as an EAR application
 - **Tomcat** and **Jetty**, as a WAR application
 - **Equinox**, as an OSGi application (and Virgo very soon)



WorkList

Clipboard

No document in clipboard

> Default domain > Workspaces > demo workspace > livrables

livrables

Content

Edit

Alerts

History

Manage

New

Import a file

Items/page 20

Filter

<input type="checkbox"/>	Title	Modified	Last contributor	Author	Version	State
<input type="checkbox"/>	cmis-spec-v1.0.pdf	9/22/11 6:55 PM	Administrator	Administrator	0.0	Project
<input type="checkbox"/>	tout en pdf	6/10/10 11:49 AM	Administrator	Administrator		Project
<input type="checkbox"/>	Webinar_Nuxeo_ReadSoft	6/10/10 11:48 AM	Administrator	Administrator	1.0+	Project

Paste

Add to worklist

Copy

Delete

RSS | ATOM

Modularity Framework

- ~~Java SE~~
 - No bundle life cycle, no modularity, no extension system
- ~~Java EE~~
 - All is packaged as one big application (EAR or WAR), cannot update or add a feature without recompiling the entire application
- **OSGi** — yes, but
 - Not compatible with Java EE
 - No real OSGi Enterprise Framework implementations yet

Additions to OSGi

- To achieve a **plugin** model
 - Eclipse already had the answer: extension points
- To integrate with **Java EE** application servers
 - Configuration fragments, pre-startup assembly of *web.xml* and *application.xml*
 - Specific OSGi runtime that can load bundles in an EAR

Extension Points

- Eclipse used EXSD, which is very tied to PDE
 - *(remember it was 2006 and we had to use maven)*
- Nuxeo redefined something similar but more flexible
 - **No intermediate** object model between services and contributed extensions
 - Write an extension class and map it to XML using Java **annotations**
 - **Easy** to write for developers, no specific IDE needed
- Available in **Apricot**

Extension Point Example

```
<extension point="servlets"
  target="org.sample.ServletRegistry">
  <servlet name="jaxrs" path="/ecr"
    class="org.sample.MyServlet">
    <filters>
      <filter class="org.sample.AuthFilter" />
      <filter class="org.sample.TxFilter" />
    </filters>
  </servlet>
</extension>
```

```
@XObject("servlet")
public class ServletDescriptor {

    @XNode("@name")
    protected String name;

    @XNode("@class")
    protected String classRef;

    @XNode("@path")
    protected String path;

    @XNodeList(value="filters/class",
      type=ArrayList.class,
      componentType=Class.class)
    protected List<Class<Filter>> filters;

}
```

Configuration Fragments

- Java EE configuration is **monolithic**
 - To declare servlets (*web.xml*) one must know in advance the servlets provided by all the different bundles, same for *application.xml*
- Using templates for Java EE configuration files
 - Dynamically **generate** these files at **application startup** from the configuration contributed by each bundle
- Ability to package EAR applications that can **adapt** themselves to the configuration provided by new application bundles
 - No need to have different product packagings for different configurations
- Needs a **server-specific hook** to do this processing

Configuration Fragment Example

```
<?xml version="1.0"?>
<fragment version="1">
  <extension target="web#SERVLET">
    <servlet>
      <servlet-name>ResourceServlet</servlet-name>
      <servlet-class>
        org.nuxeo.ecm.web.servlet.ResourceServlet
      </servlet-class>
      <init-param>
        <param-name>prefix</param-name>
        <param-value>/nuxeo/site/skin</param-value>
      </init-param>
      <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
      <servlet-name>ResourceServlet</servlet-name>
      <url-pattern>/site/skin/*</url-pattern>
    </servlet-mapping>
  </extension>
</fragment>
```

```
<?xml version="1.0" ?>
<web-app version="2.4">

  <listener>
    <listener-class>
      com.sun.faces.config.ConfigureListener
    </listener-class>
  </listener>

  %{LISTENER}%

  %{FILTER}%

  %{SERVLET}%

</web-app>
```

Java EE Features Added to OSGi

- Full OSGi integration of **JAAS** (authentication system)
- Full **JTA** support through Apache Geronimo (transactions)
- Full **JCA** support through Apache Geronimo (resource adapters and pooling)
- In-memory **JNDI** server
- Future plans to integrate the work done in the **Gemini** project (and also support **Virgo**)

The Future

Where Are We Going?

- Finish Apricot first release
 - Cleanup, testing framework
- Replace Nuxeo Core with Apricot
 - Core running under a full OSGi container
- Update JSF and Seam to Java EE 6
- Use CDI for easier access to services
- Better IDE support

Lots of New Features

- HTML5
- Storage improvements
- VCS improvements
- OpenSocial
- Activity Stream
- Collaboration
- Search
- Workflow
- Document Routing
- Filesystem sync
- Static WAR packaging
- Video and Rich Media
- Modular installer
- Cloud dev and prod deployment
- Mobile devices connectors
- Nuxeo IDE
- Nuxeo Studio

Thank You!

nuxeo

Open Source ECM

Additional Information & Links

- <http://www.nuxeo.com>
- <http://community.nuxeo.com>
- <http://www.eclipse.org/apricot>
- <http://www.slideshare.net/efge/cmis-and-apache-chemistry-apachecon-2010>
- <http://docs.oasis-open.org/cmis/CMIS/v1.0/cmis-spec-v1.0.html>