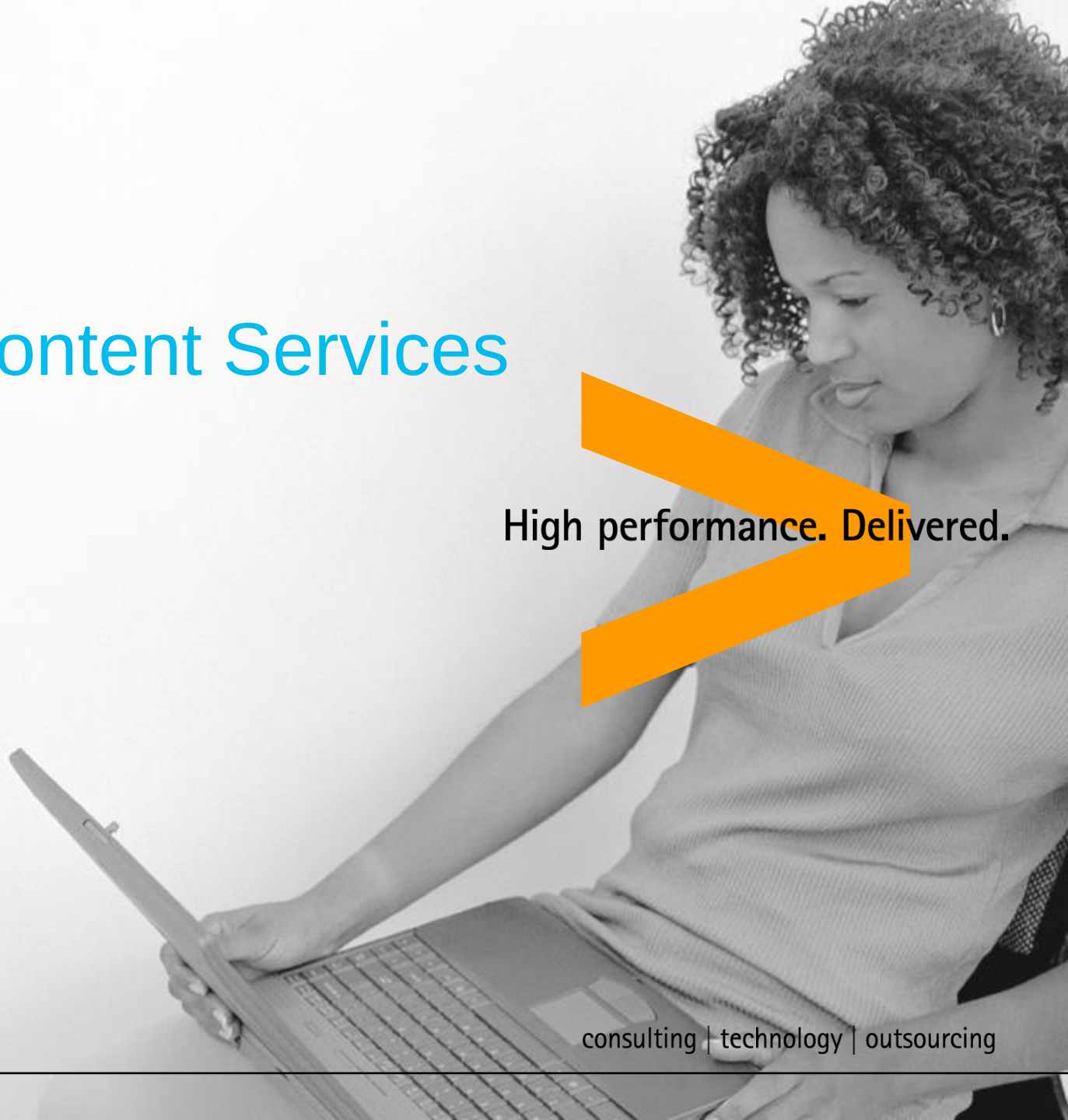
Alfresco Content Services

Parameswari



High performance. Delivered.

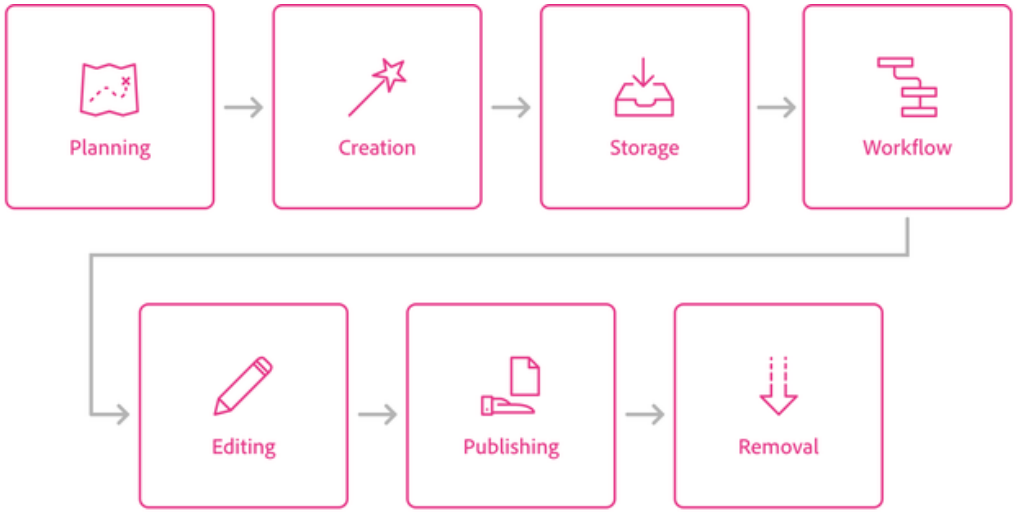
consulting | technology | outsourcing

Goals

- Alfresco Architecture
- Installation
- Content Model
- Configuration and Layouts
- Alfresco Share Components Customization
- Working with the new Aikau framework in Alfresco Share
- Workflows

Content Management Story So Far 1980s

- Content management, often abbreviated as CM, refers to the process of generating, storing, accessing, delivering, and refining digital media through a unified platform.
- 40 years ago, Enterprise needed way to manage all the electronic contents they were creating.



Content Management Types

- Enterprise content management system (ECM)
- Digital asset management system (DAM)
- Social media content management
- Mobile content management (MCM)

Enterprise content management system (ECM)

- An enterprise content management system (ECM) operates much like a CMS.
- But it's designed to handle the vast content needs of large-scale organizations.
- It's a powerhouse that manages, stores, and delivers a wide array of content types, from documents and images to videos and more.
- But the true strength of an ECM lies in its ability to streamline content workflows across different departments, fostering collaboration and ensuring consistency in content delivery.
- With an ECM, enterprises can effectively manage their content lifecycle, so their digital experiences remain fresh, relevant, and aligned with their brand's core values and unique attributes.

Content Management Story So Far 1990s

Alfresco Born in 2005

Alfresco™

A fresh approach to Enterprise Content Management

Open Source entered the ECM scene ...

ECM Solutions in 2021

Alfresco Story To Date

Alfresco is now the **largest open source content management company** in the world:

- 400+ employees
- 2 main products - Alfresco One (ECM) and Alfresco Activiti (BPM)
- Multiple modules, such as Records Management
- Millions of downloads
- Hundreds of global partners
- Thousands of customers in many different countries
- Thousands of community members
- 100K+ active installations

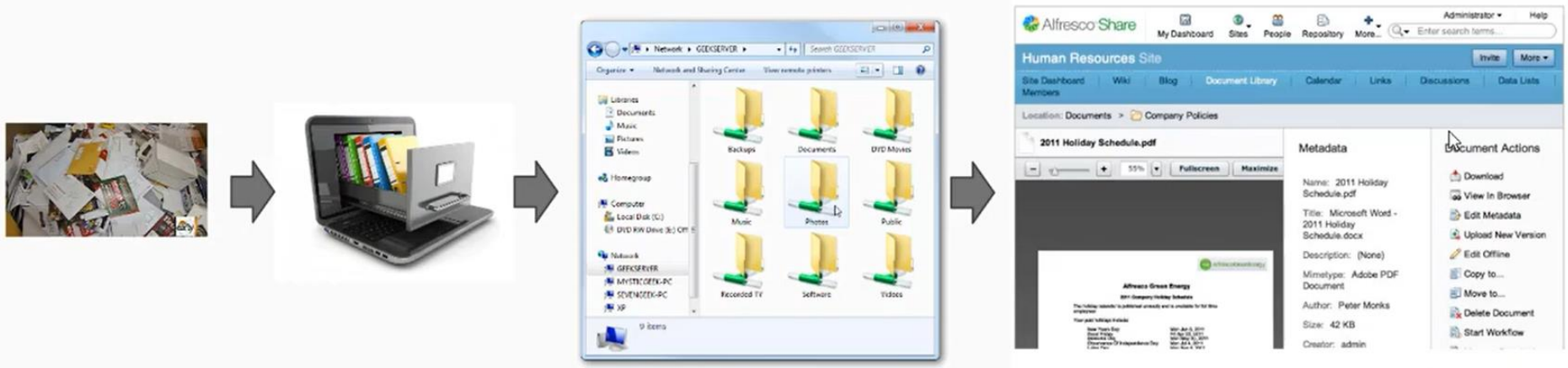
Alfresco ECM Customers

Alfresco Products

- Alfresco community edition: LGPL licensed suitable to developers and technical enthusiasts.
- Alfresco One-Enterprise edition: Overcomes limitation of community edition by being highly modular and scalable.
- Alfresco cloud Edition: a SaaS version of Alfresco.
- Alfresco mobile: A free mobile document management app for iOS and Android tablets and smartphones.
- Alfresco Activiti: Provides a highly scalable, Java-based workflow and Business Process Management (BPM) platform.

ECM Product Capabilities

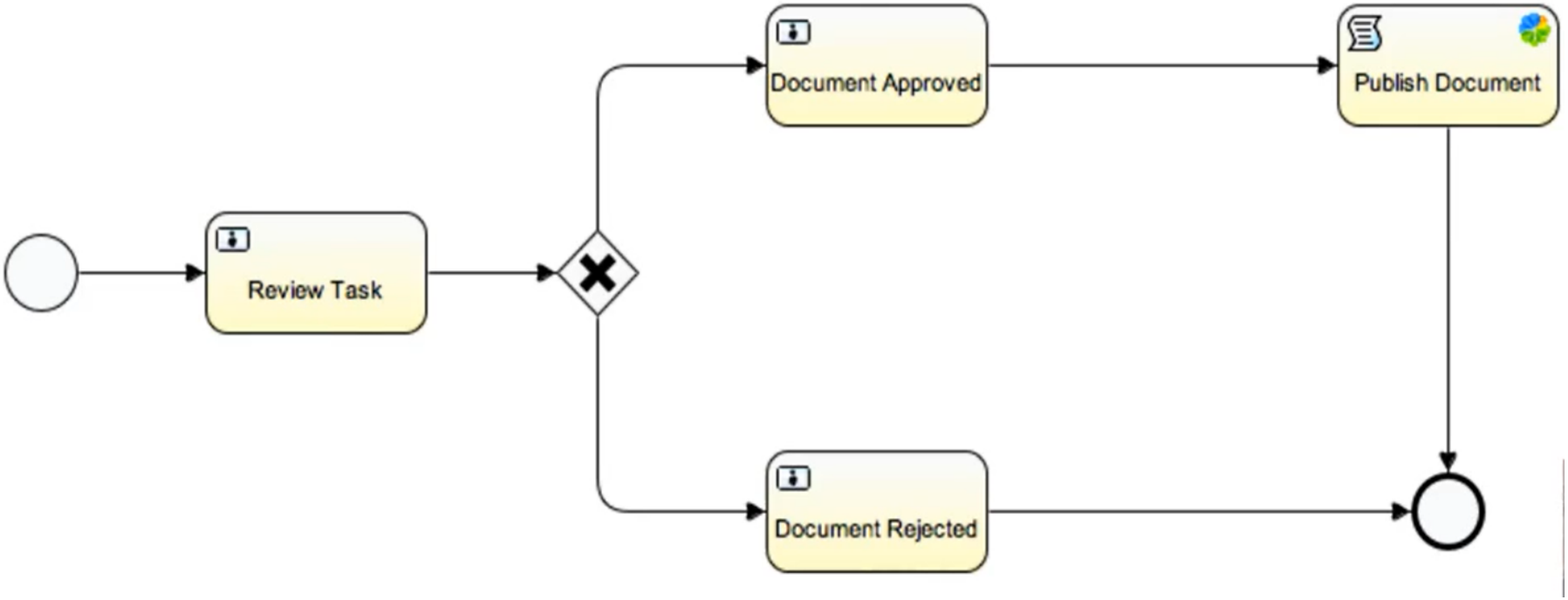
Document Management



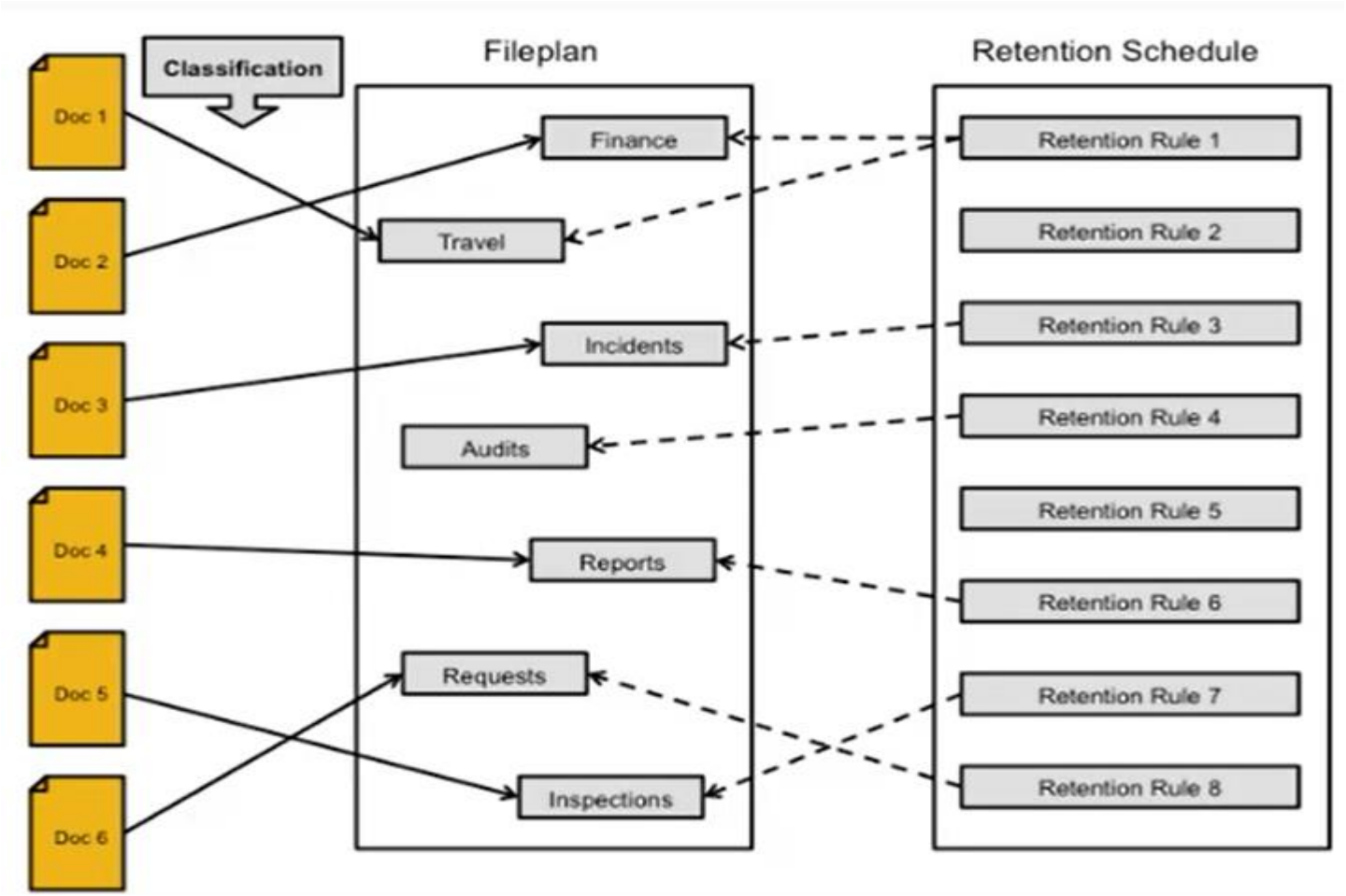
Can store and track all type of documents.

Workflow Management

- **In Alfresco a workflow is a sequence of connected tasks applied to a document or other item of content.**
- **Each task can be performed by a person, a group, or automatically.**
- **There are number of tools you get to design, execute, and monitor your workflows.**

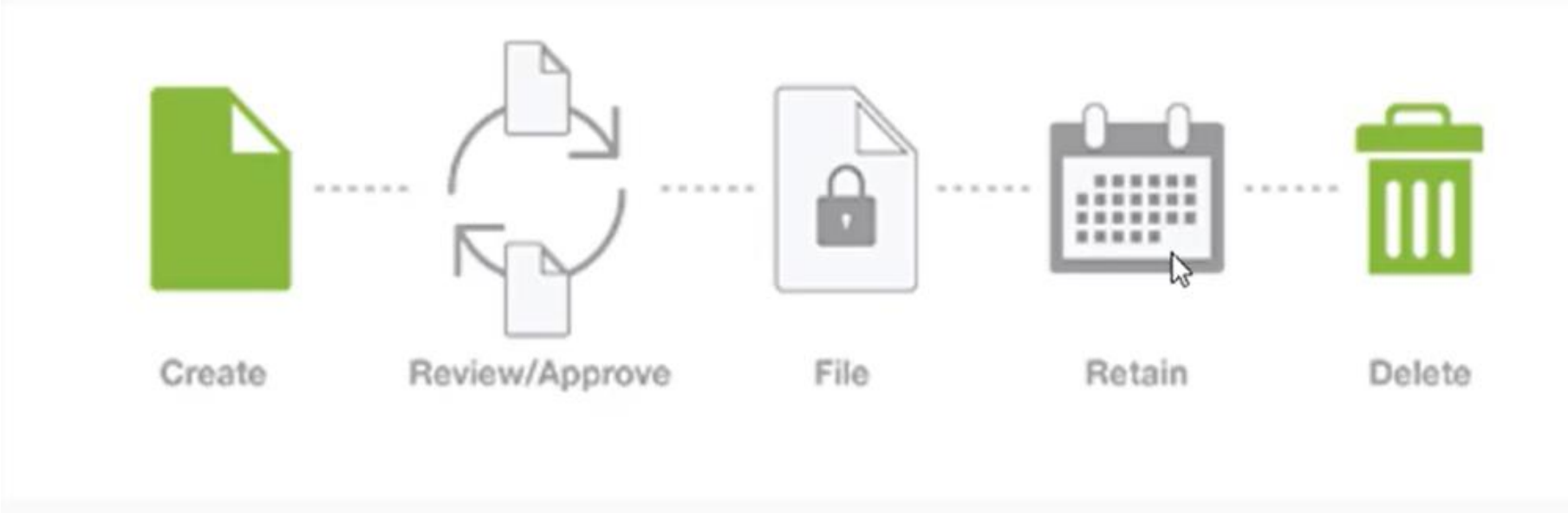


Electronics Record Management



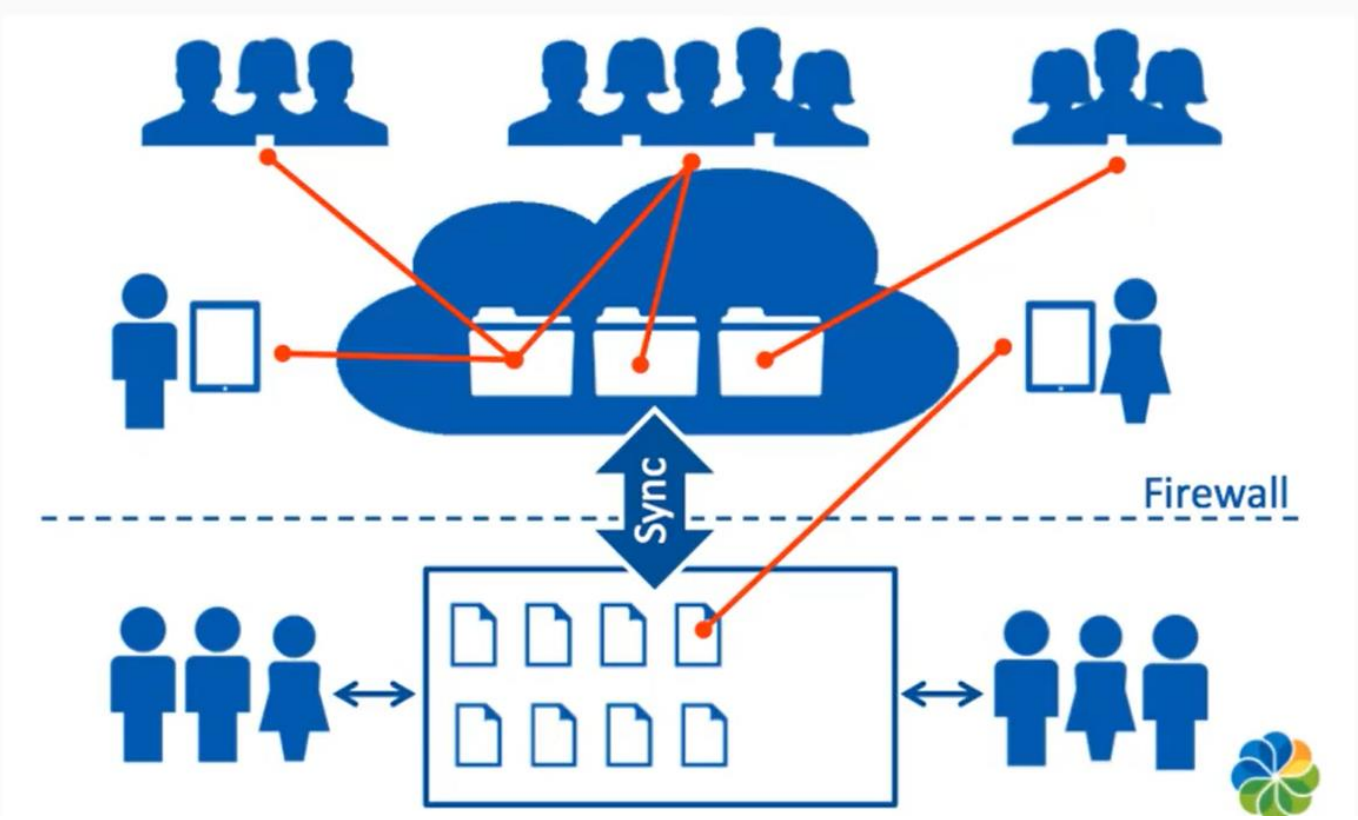
Electronics Record Management

- **Can manage records which can be either tangible objects, such as paper documents like birth certificates, driver's licenses, and physical medical x-rays, or digital information, such as electronic office documents, data in application databases, web site content, and electronic mail (email).**



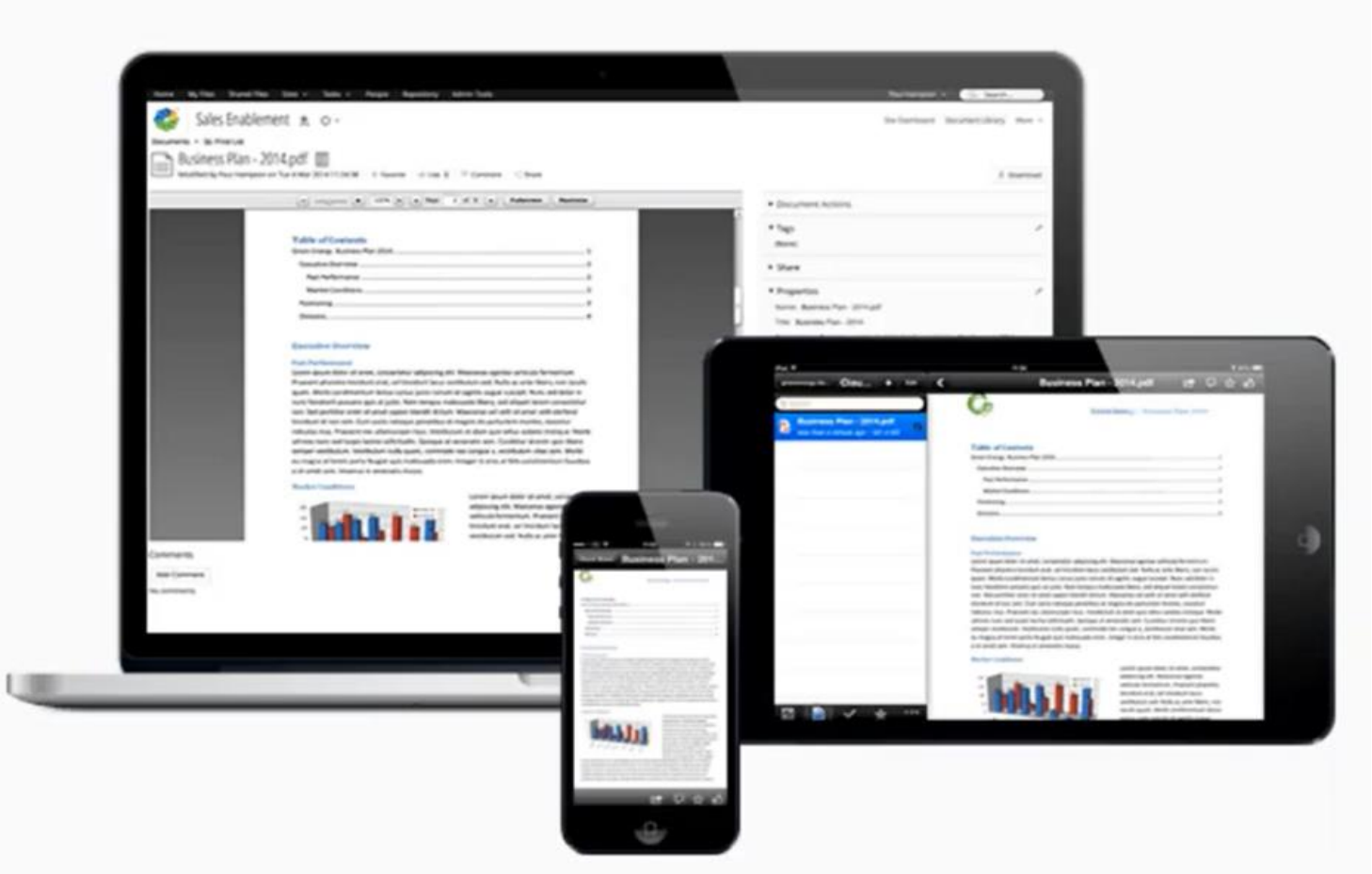
Enterprise Collaboration

- **For effective Collaboration Alfresco provides tools to facilitate communication, share information, and run projects within your team as well as to include partners, consultants, external agencies, and customers.**



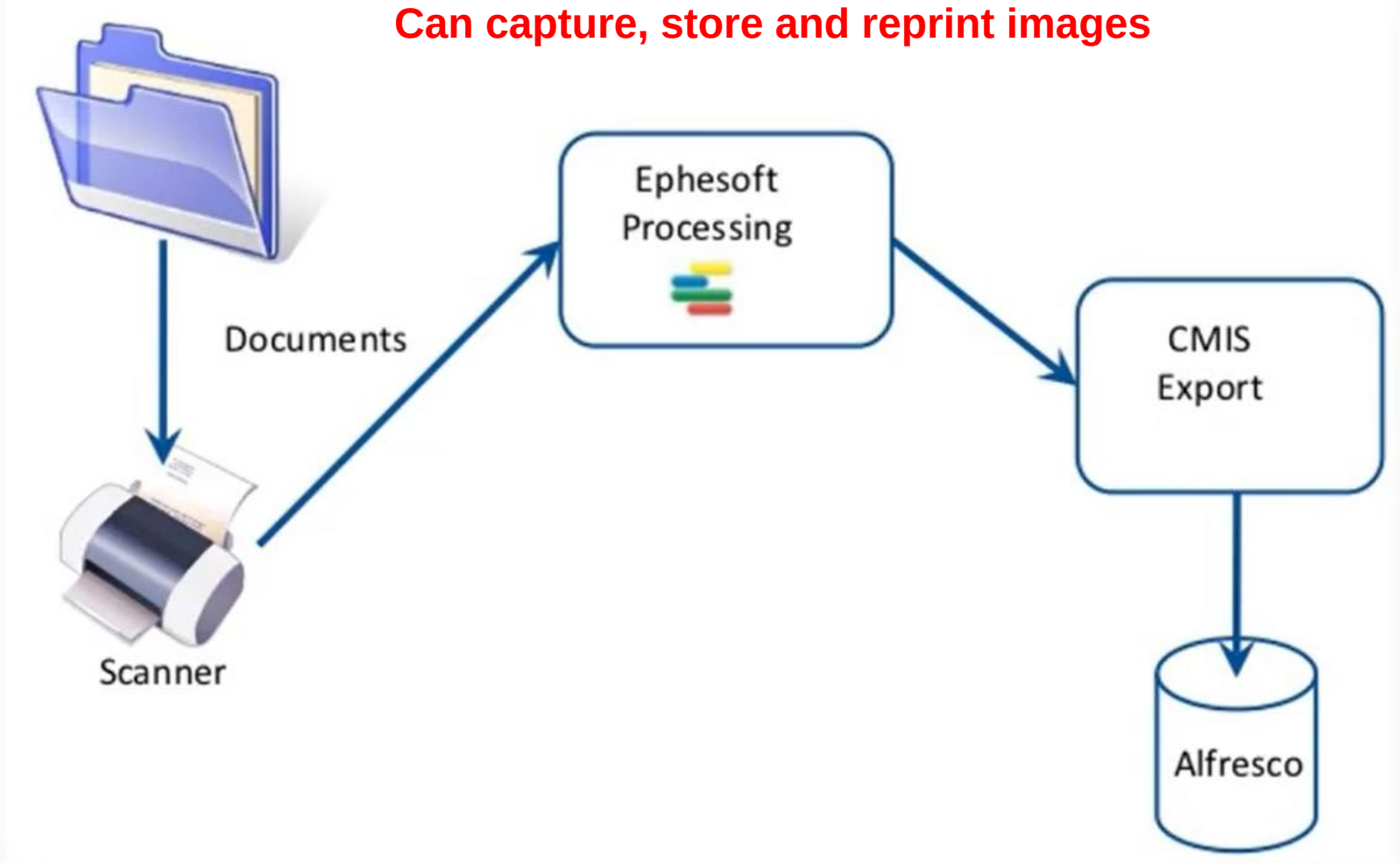
Mobile Content Management

- **Allow users with little knowledge of web / Mobile programming languages or markup languages to create and manage website content with relative ease**



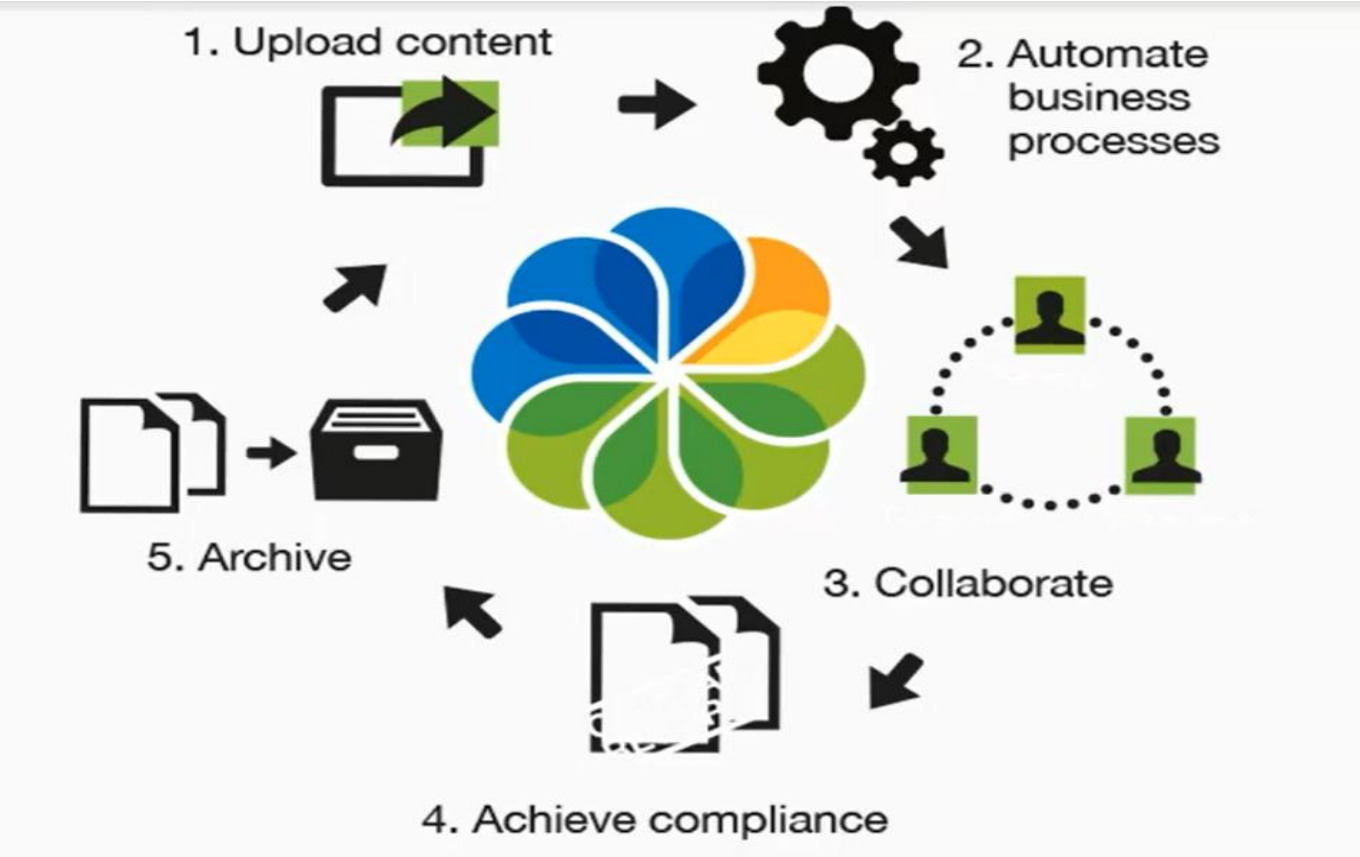
Scanning

Can capture, store and reprint images



Content Management

- **Allow users with little knowledge of web programming languages or markup languages to create and manage website content with relative ease**



Why Alfresco?

- Provides a clean and secure web interface for document management.
- Provides all document management features like metadata, versioning, search, audit, workflow.
- Different programmatic ways are available to integrate your application with Alfresco.
- Some of the interfaces are CMIS, Web scripts, File based access, Open search and feeds, JAVA, PHP and .Net applications.
- Is one of the most portable solutions available having cloud support. You can use it on web, mobile, tablet.
- One of the best features of Alfresco is that it can be easily integrated with Google docs, Office, SAP, AWS and other content tools.

Why Alfresco

Content Management (i.e. Document Management, Image Management etc)

Plus a number of modules for:

- Records Management (RM)
- Media Management & Scanning
- Content Encryption
- Analytics
- Transformation Server
- Outlook Client

And integrations with:

- SAP
- Salesforce
- Ephesoft
- Outlook
- Google Docs

Case Study

- Requirements
  - End user should be able to upload different types of documents on website in support of the application.
  - The document is further validated by Admin.
  - All users of the portal have secure inbox access wherein they are notified about account activities/admin change actions/application statues.
  - These emails should be saved as PDF documents.
  - Authorized persons should be able to view/edit the static web content of the application as and when required.

Case Study

- Challenges
  - We needed one solution to fit all above requirements.
  - We knew the content size would grow with time.
- The key points which we considered:
  - A system with strong security.
  - A scalable solution to meet future requirements.
  - A system which can be easily integrated with existing Java application.

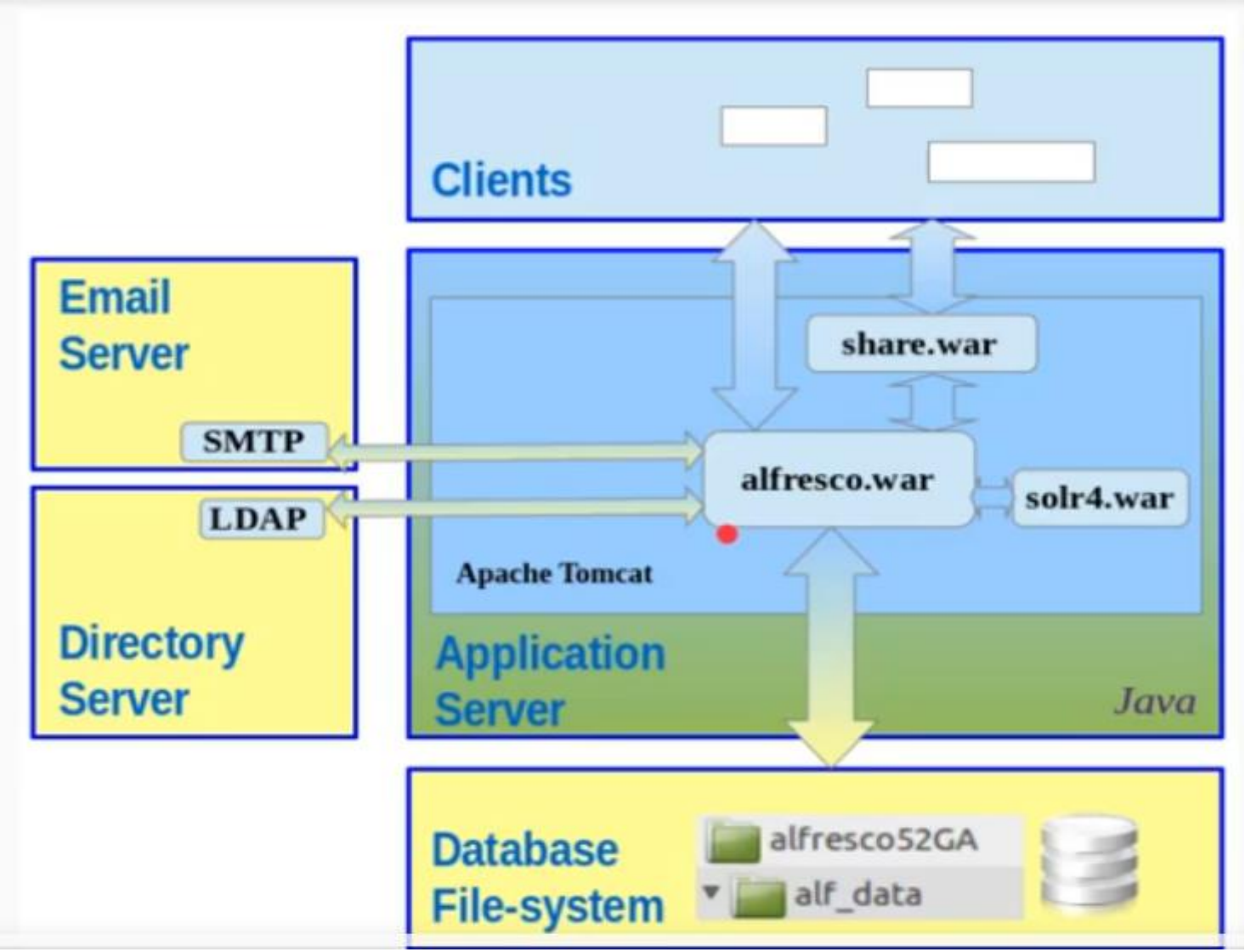
Case Study

- Solution:
  - After analyzing available ECMs we figured out Alfresco was able to cater to all these requirements and was a cost-effective solution.
  - With following features, Alfresco stood as a solution encompassing all our requirements.
    - Alfresco guarantees content safety through its security model which is comprised of authentication and authorization.
    - Alfresco One-Enterprise edition is highly modular and scalable.
    - We could easily integrate Java application with Alfresco using its OpenCMIS extension which depends on Apache Chemistry OpenCMIS libraries.

Alfresco Limitations:

- Though alfresco meets most of the requirements there are few limitations of this ECM as per different users.
  - The system lacks backup capabilities.
  - Its support system needs improvement in Live chat and FAQ sections.
  - Few of the UI features can be automated and provide better user experience.
  - Enrichment of current content is not so easy, have to reload the content.
  - No support for XQuery.
  - Content transformation is absent.

Alfresco Architecture

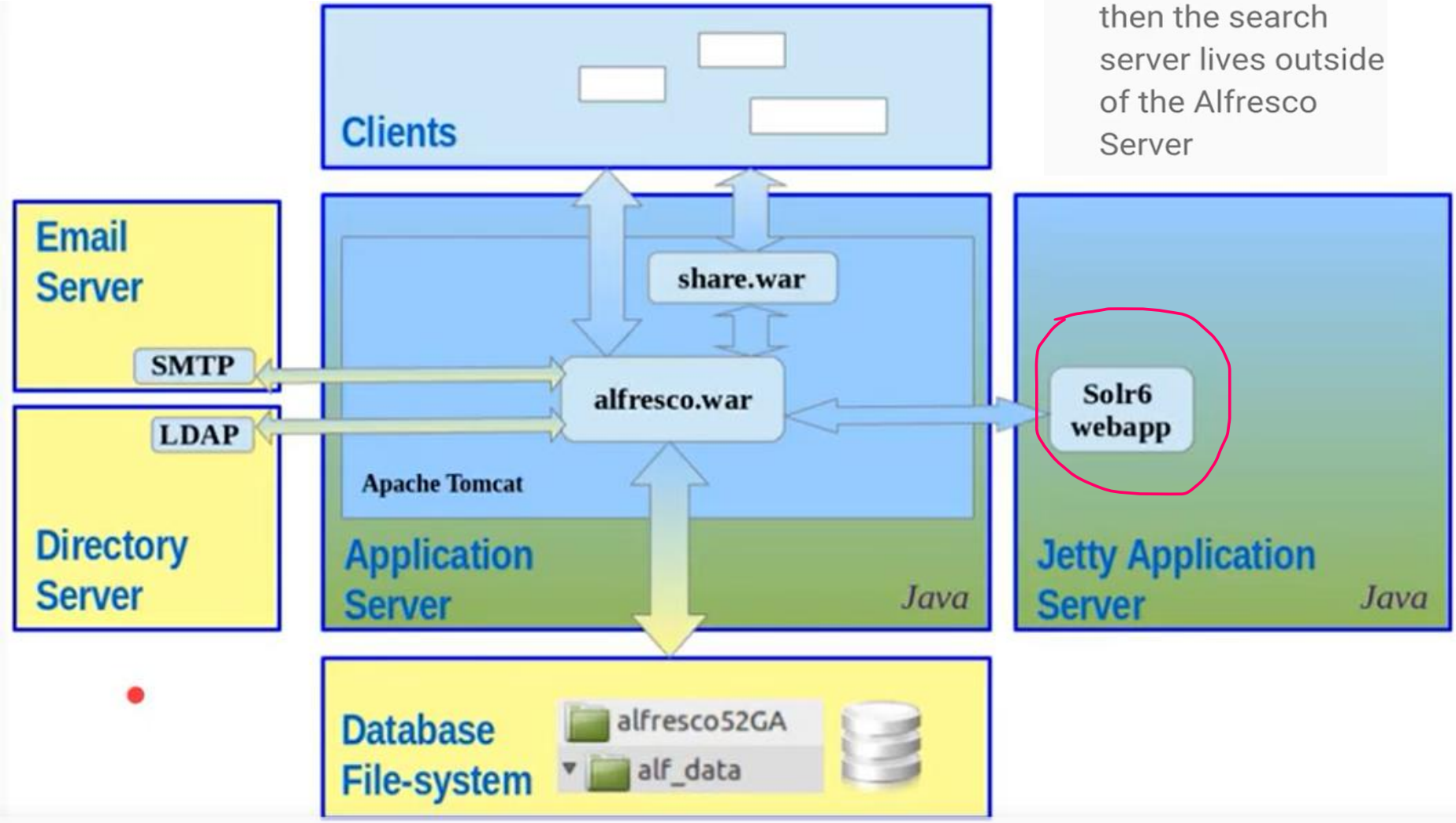


Alfresco Architecture

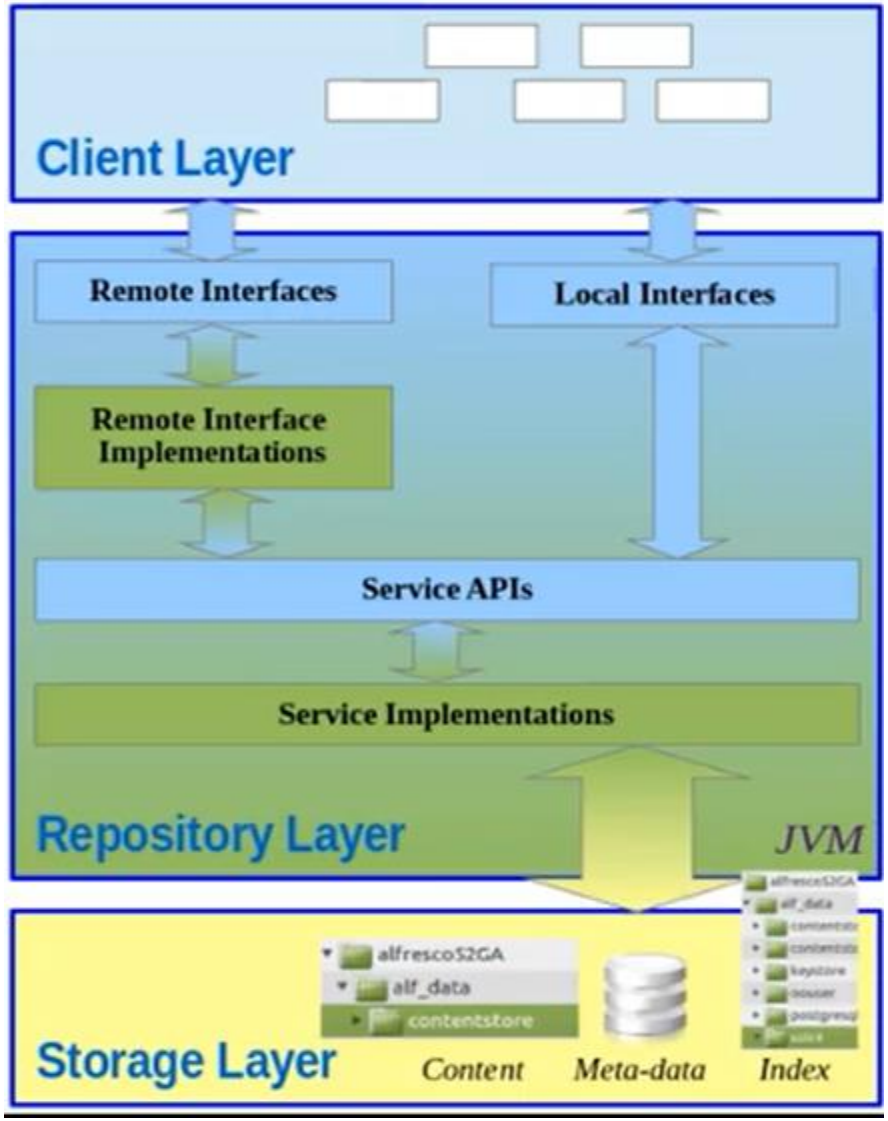
- The Alfresco Platform is built up of a number of web applications
- These web applications runs in an application server such as Apache Tomcat, which runs in a Java Environment
- Alfresco Web Applications:
  - **alfresco.war** – the repository
  - **solr4.war** – the search application
  - **share.war** – the main user interface
- Enterprise Application Integration:
  - The Alfresco application talks to other enterprise servers such as Directory and Email servers

Alfresco Architecture

- If Solr 6 is used, then the search server lives outside of the Alfresco Server



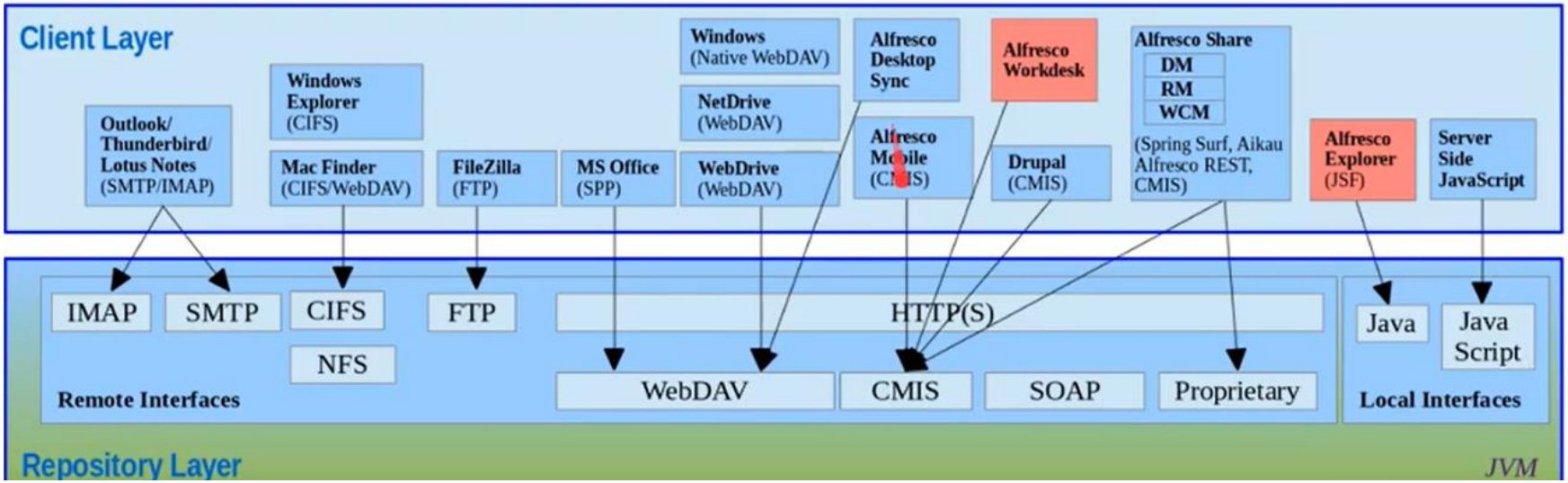
Alfresco Layered Architecture



Alfresco Layered Architecture

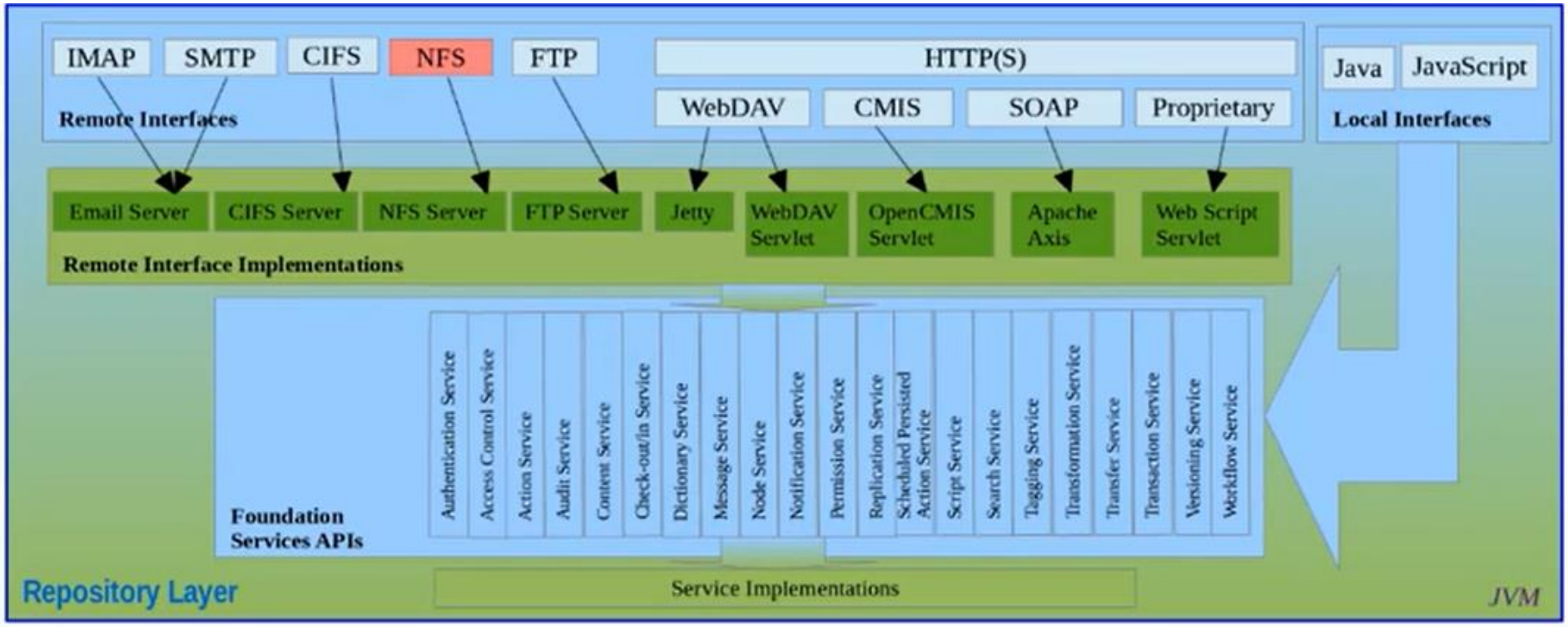
- Client layer
  - UI Clients (share.war etc), Tools, Applications
- Repository layer
  - alfresco.war, solr4.war
  - The Repository layer provides a multitude of Interfaces and Service APIs through which clients can connect
  - It also provides all the service Implementations for the service APIs
- Storage Layer
  - As the name suggests this would be where all the data will be stored

Alfresco Client Layer



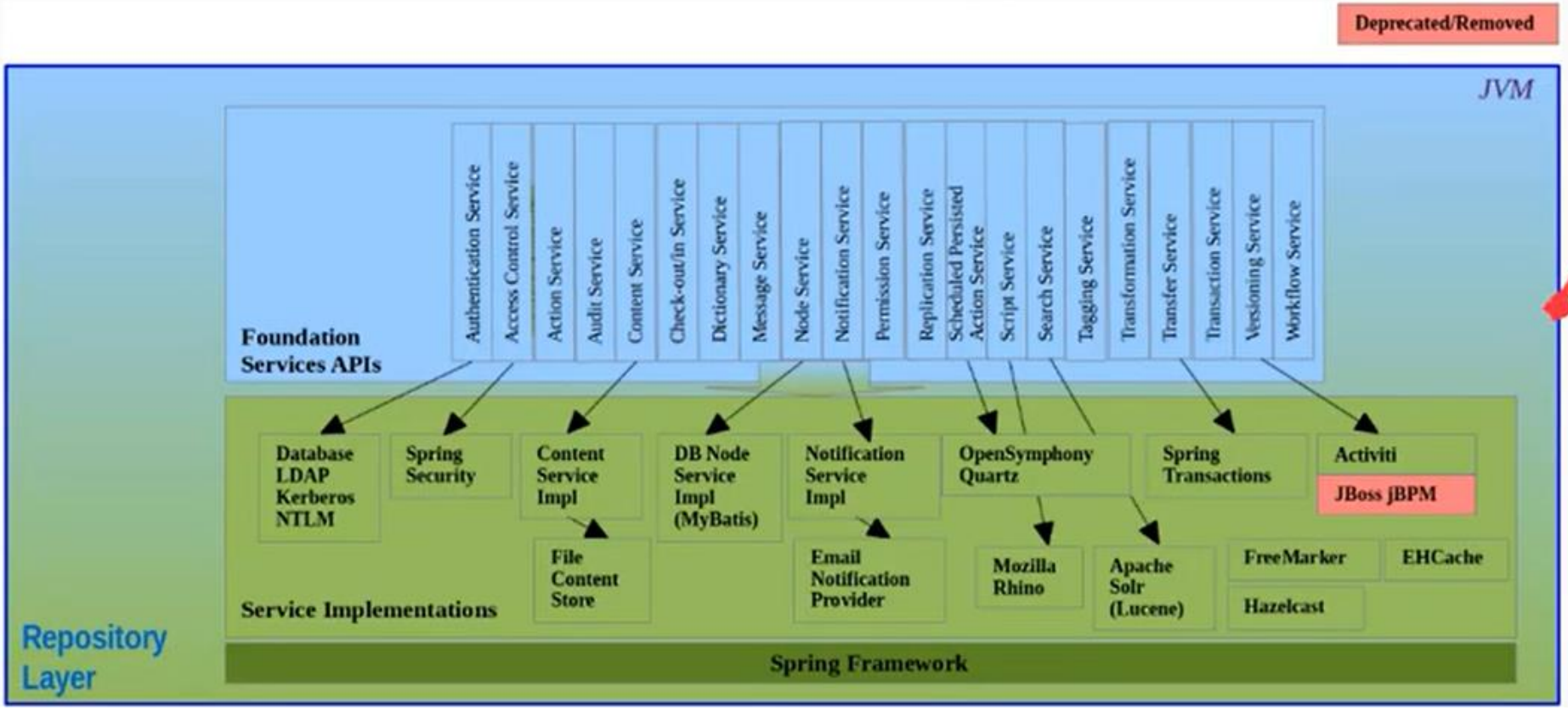
The client layer consists of all the client applications (UI) that talks to Alfresco via the Remote Interfaces and Service APIs:

Alfresco Repository Layer



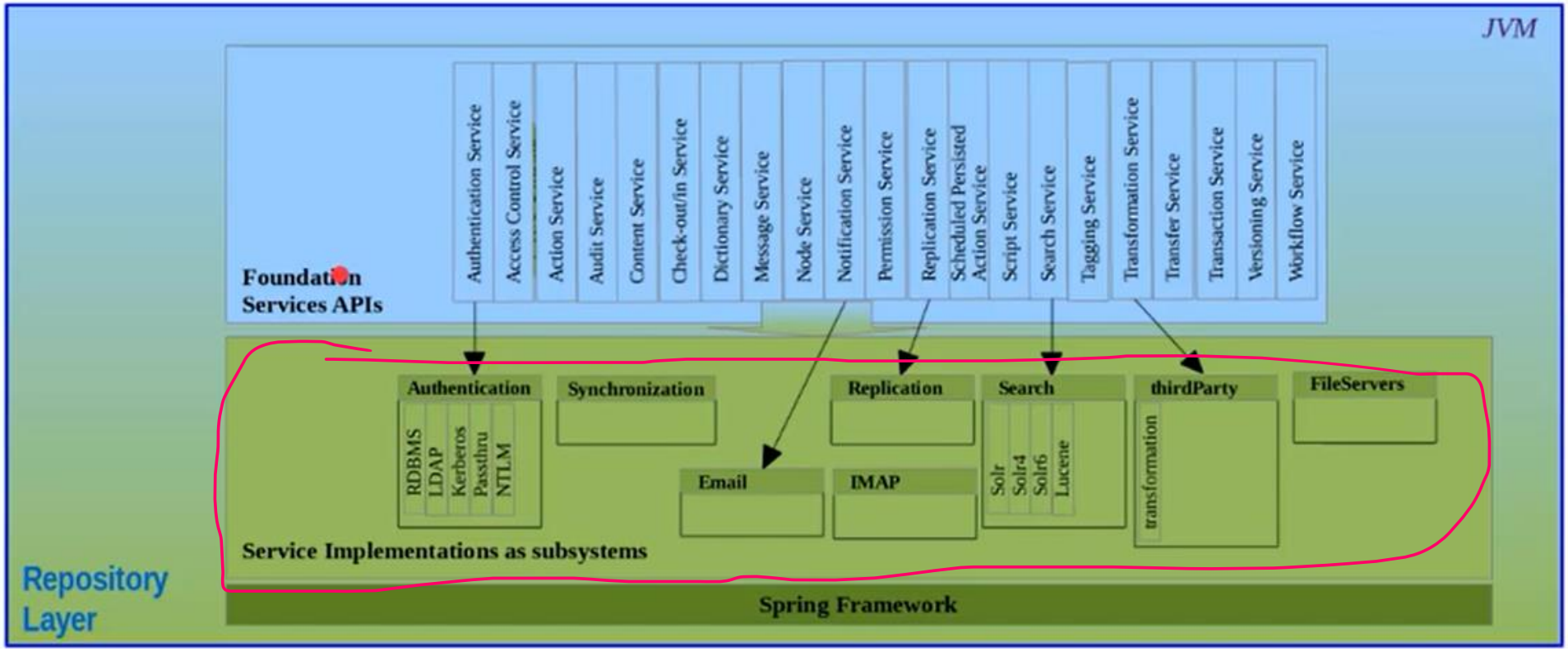
Each one of the internal Servers or Servlets that the client applications talk to uses the Public Service interface to manage content:

Alfresco Repository Layer



The public service APIs are implemented with different open source libraries and custom component implementations:

Alfresco Repository Subsystems

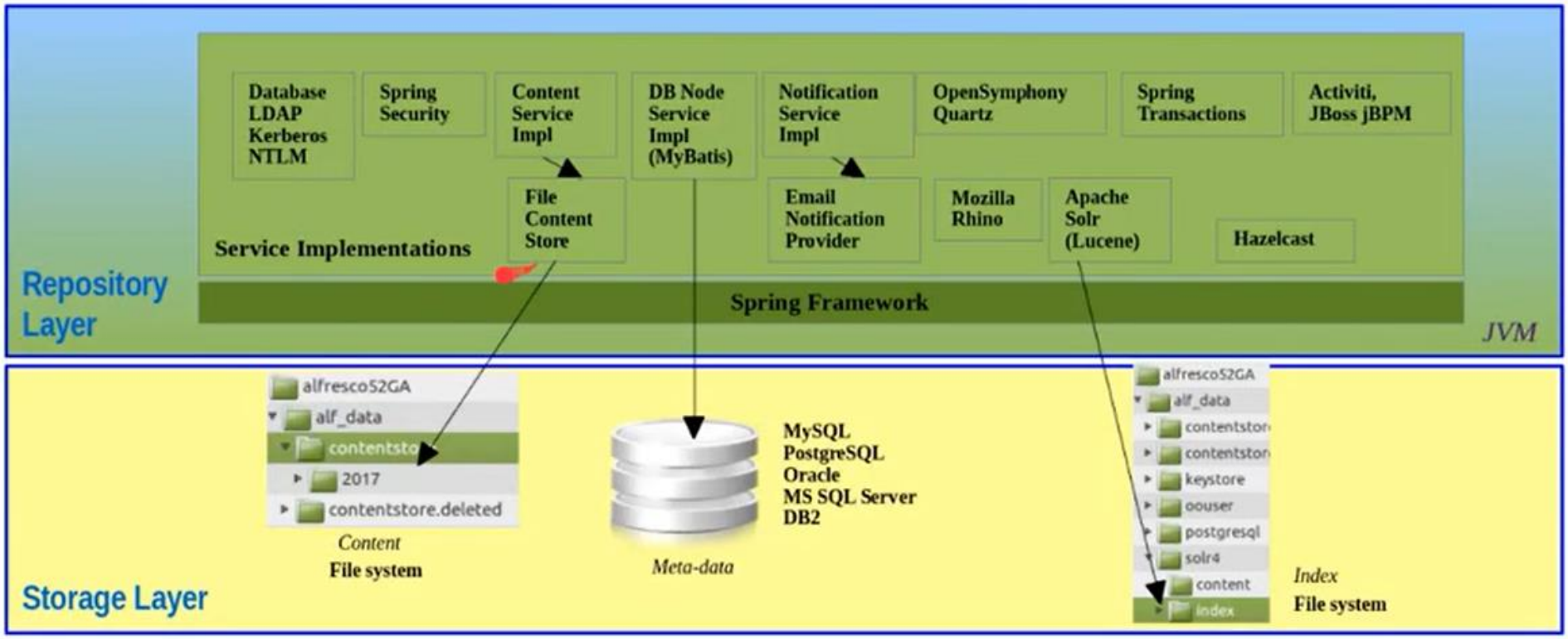


Some of the services are implemented as subsystems, which gives greater control over them:

Alfresco Repository Subsystems

- Subsystems are configurable modules responsible for a piece of functionality in the Alfresco content management system
- A subsystem can be thought of as a mini-server that runs embedded within the main Alfresco server
- A subsystem has the following characteristics:
- It can be started, stopped, and configured independent of the Alfresco server
- It has its own isolated Spring application context and configuration

Alfresco Storage Layer



The Public Service Implementations talk to the database and the file system in the Storage Layer:

How does the integration work

The easiest way to put it is as follows:

- Alfresco uses HTTP GETs to talk to Solr and search for content.
- Solr updates the indexes by talking to Alfresco to get the number of transactions that have been committed since it last talked to Alfresco.
- The custom components in Solr that talks to Alfresco are referred to as trackers.

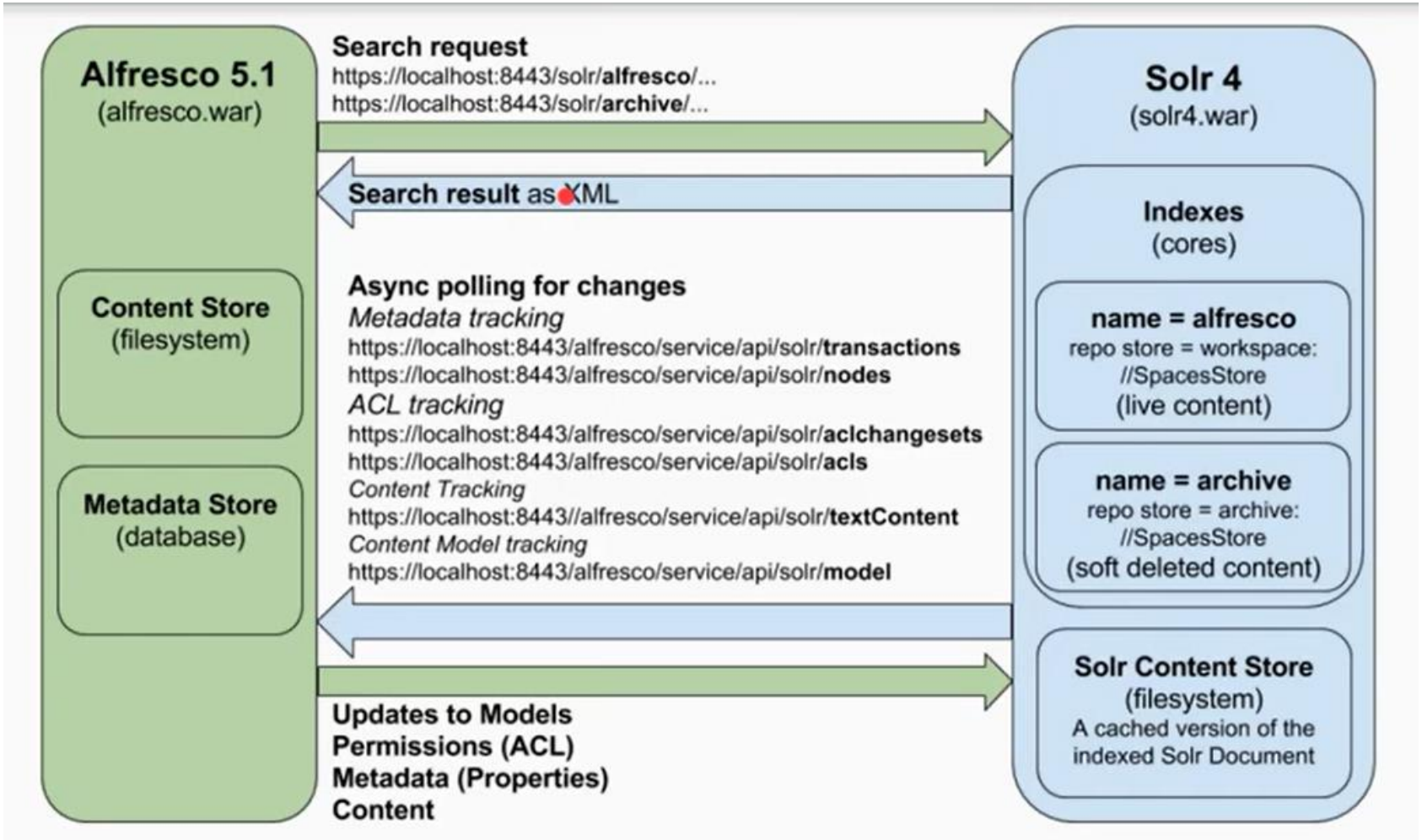
Solr talks to Alfresco every 15 seconds to see if any new transactions have been committed. So the index will eventually be up to date with the database.

- This means that if you upload a document to Alfresco and then immediately execute a full text search, it's not 100% sure that you will hit the document just uploaded. An important thing to know is that unlike with Solr 1, Solr 4 indexing is split between metadata tracking, ACL tracking, and content tracking.

How does the integration work

- Solr will also keep track of new custom content models that have been deployed and download them to be able to index the properties in these models.
- Permission settings will also be downloaded by Solr so it can do query time permission filtering.
- There are 2 cores (i.e. indexes):
  - **WorkspaceStore**: for searching all live content (i.e. stuff in `alf_data/contentstore`)
  - **ArchiveStore**: for searching soft deleted content (i.e. content in `alf_data/contentstore` that has been marked as deleted)

Big Picture


Alfresco 5.1
(alfresco.war)
Content Store
(filesystem)
Metadata Store
(database)
Search request
https://localhost:8443/solr/alfresco/...
https://localhost:8443/solr/archive/...
Search result as XML
Async polling for changes
Metadata tracking
https://localhost:8443/alfresco/service/api/solr/transactions
https://localhost:8443/alfresco/service/api/solr/nodes
ACL tracking
https://localhost:8443/alfresco/service/api/solr/aclchangesets
https://localhost:8443/alfresco/service/api/solr/acls
Content Tracking
https://localhost:8443//alfresco/service/api/solr/textContent
Content Model tracking
https://localhost:8443/alfresco/service/api/solr/model
Updates to Models
Permissions (ACL)
Metadata (Properties)
Content
Solr 4
(solr4.war)
Indexes
(cores)
name = alfresco
repo store = workspace:
//SpacesStore
(live content)
name = archive
repo store = archive:
//SpacesStore
(soft deleted content)
Solr Content Store
(filesystem)
A cached version of the
indexed Solr Document

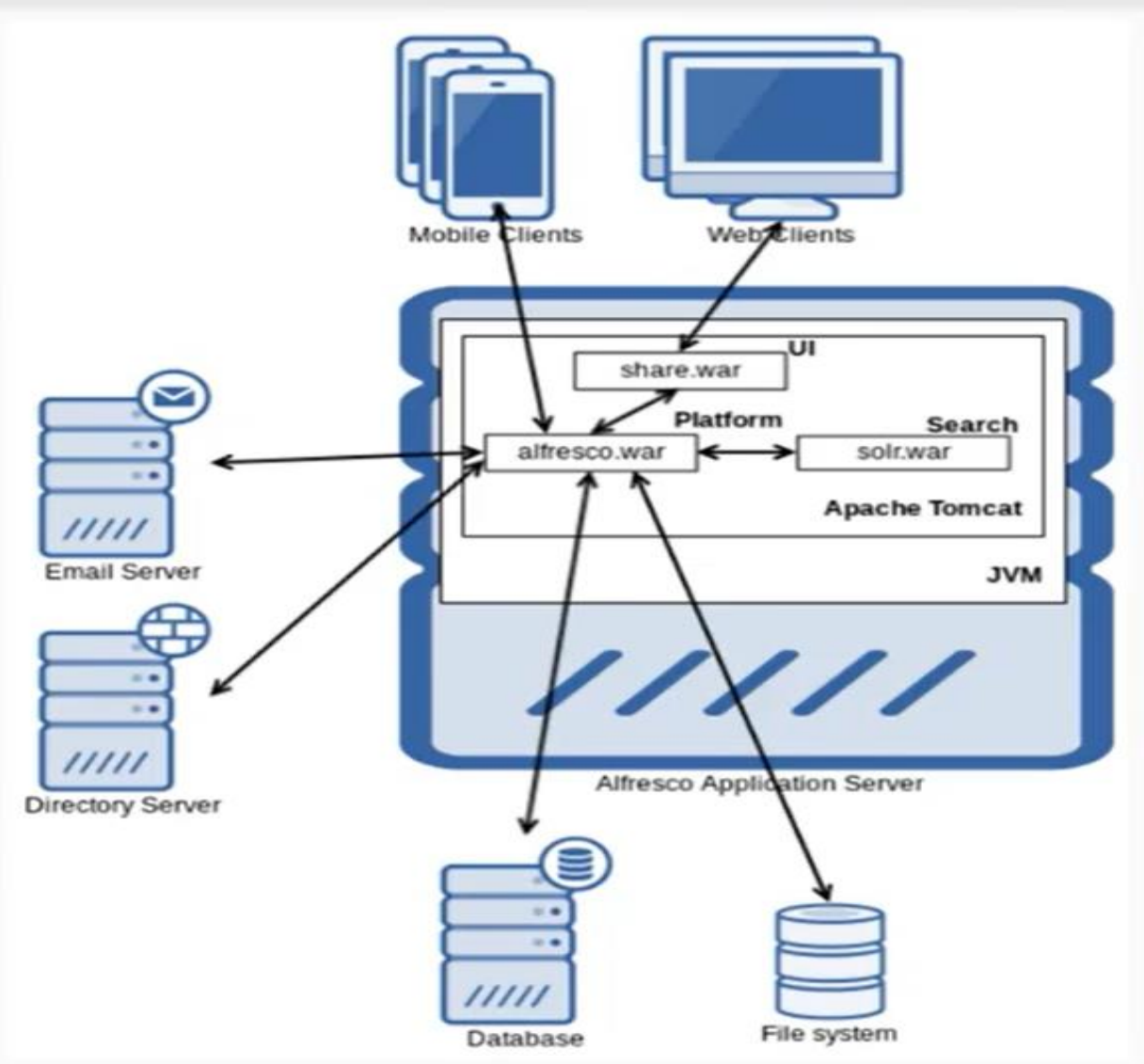
Big Picture

- As the picture showed, Solr version 4 (or 6) is used and has been extended with code to talk to Alfresco.
  - So if you are going to install Solr on a separate dedicated search box, then you need to download it from Alfresco download site and not from Apache...
- If you wanted to use a newer version of Solr such as for example 6.5, then that would be a bit of a problem. It would not be a simple task to upgrade and patch it with Alfresco specific code for trackers etc, and Alfresco might not support you...
- Note the new Solr Content Store that has been added to store every indexed Solr Document, including metadata and content text (new for the Alfresco 5 - Solr 4 integration).
  - This way things like re-indexing and updates can be done in a quicker way.

When/Where Solr Used

- Instant Search (search field in top right corner)
- Advanced Search
- Filters
- Tags
- Categories (implemented as facets)
- Dashlets such as the Recently Modified Documents, Site Content
- People Finder and Site Finder pages
- Wildcard searches for People, Groups, Sites (uses database search if not wildcard)

Product Deployment Architecture Single Server

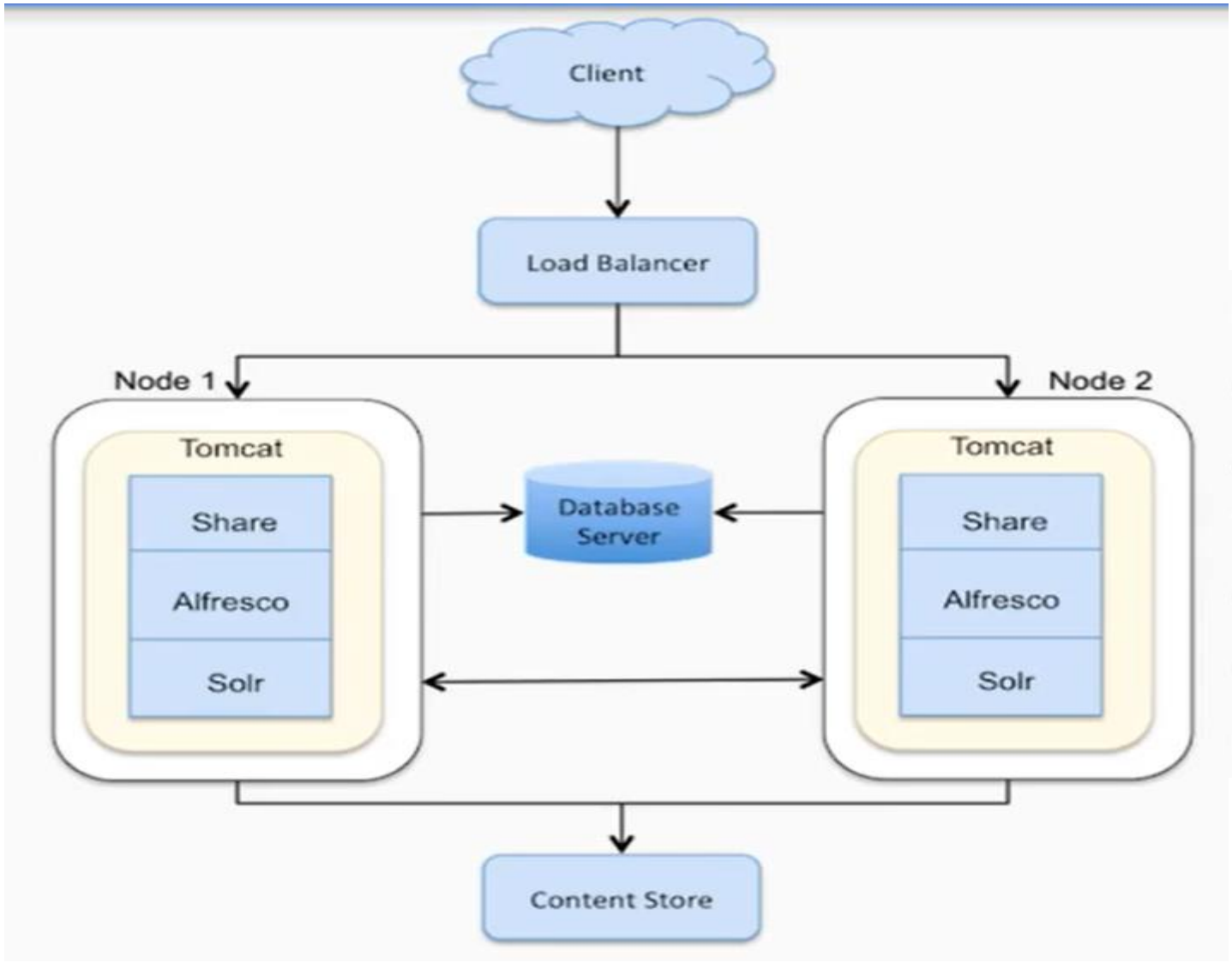


- Easy install:
  - All on one server
  - (single CPU)
- But…
  - Single point of failure (app server, content storage, database)
  - Limited performance

Product Deployment Architecture Single Server

- Components:
  - UI
  - Repository/Platform
  - Search/Index
  - Application Server (e.g. Tomcat)
  - Database (RDBMS)
  - Filesystem
- Common Integrations:
  - Email Server
  - Directory Server

Product Deployment Architecture Cluster

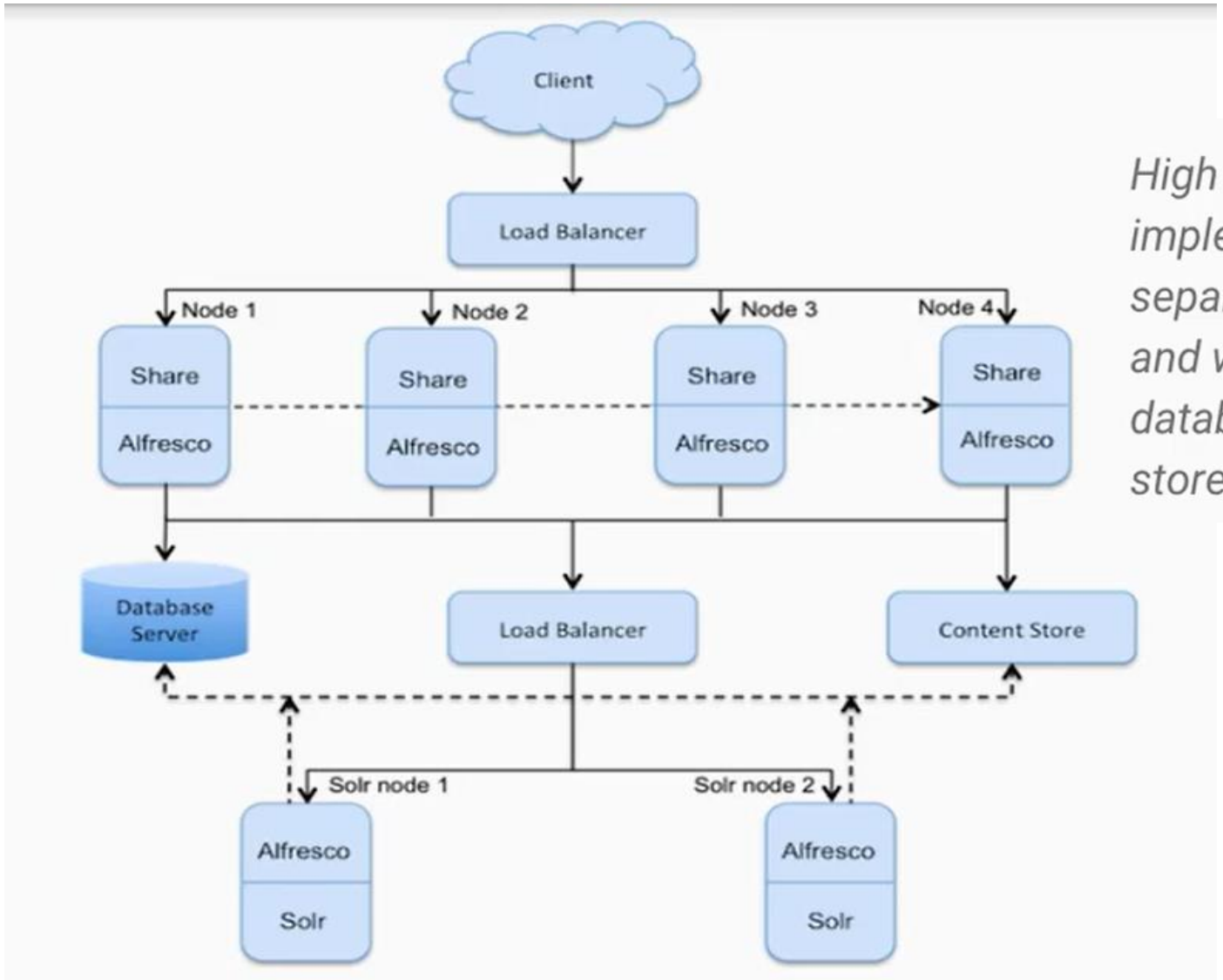


Common cluster implementation with shared database and content store (filesystem)

Product Deployment Architecture Cluster

- Alfresco Edition supports clustering the Alfresco Repository and Alfresco share
- Alfresco is three separate applications
  - Alfresco
  - Share
  - Solr
- These can be on separate tiers

Product Deployment Architecture High Throughput

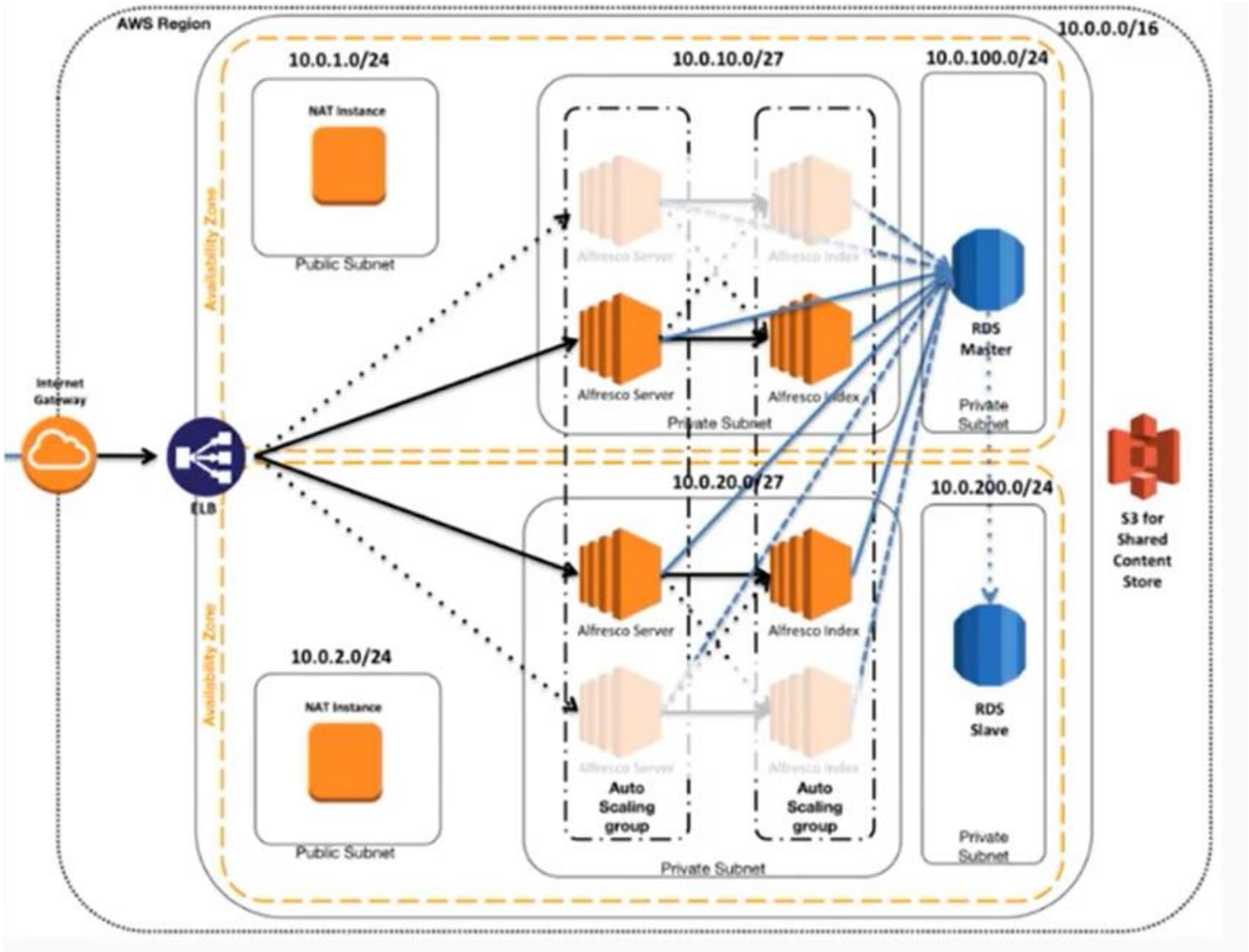


High throughput cluster implementation using separate search tier and with shared database and content store (filesystem)

Product Deployment Architecture High Throughput

- Alfresco Enterprise edition supports clustering the Alfresco index server.
- Supports redundancy and High throughput.
- But…
  - Single point of failure in database layer
  - Single point of failure in file system layer

Product Deployment Enterprise High Throughput, Redundancy and Fail-Over

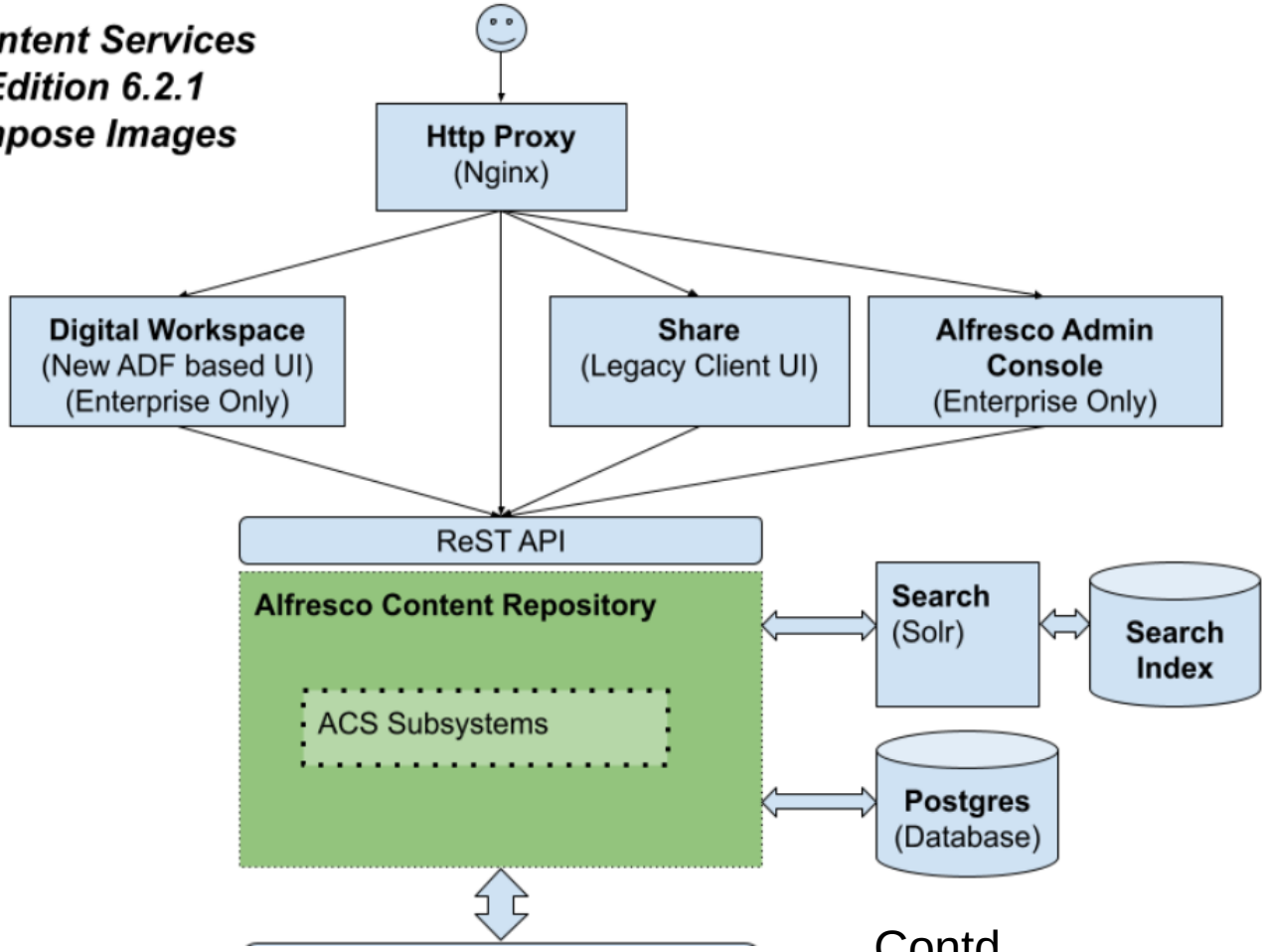


Product Deployment Enterprise High Throughput, Redundancy and Fail-Over

- Alfresco supports quick and easy installation via AWS cloud formation template.

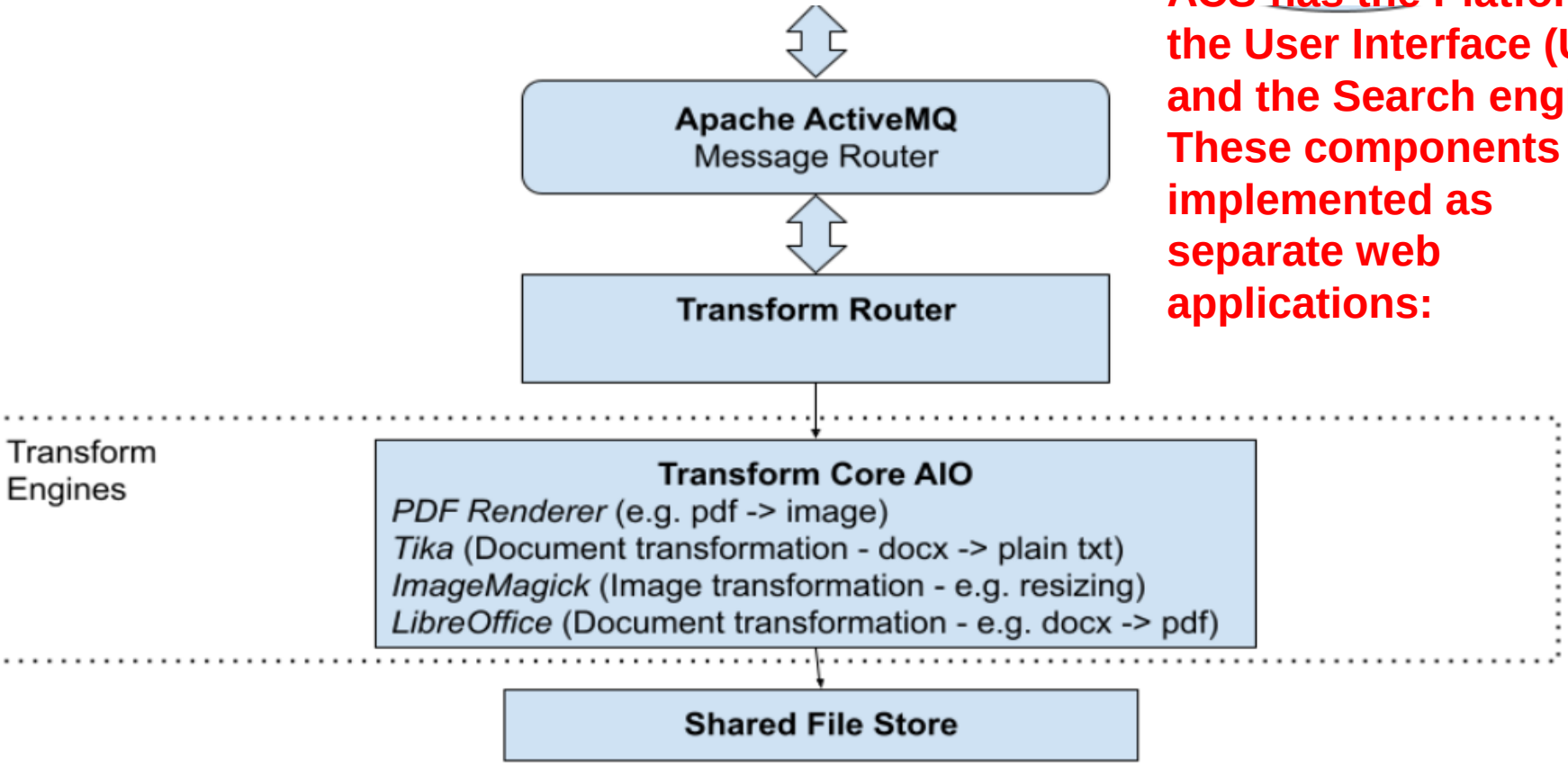
Alfresco Content Services Docker Images

Alfresco Content Services Enterprise Edition 6.2.1 Docker Compose Images



Contd….

Alfresco Content Services Docker Images



ACS has the Platform, the User Interface (UI), and the Search engine. These components are implemented as separate web applications:

Alfresco Content Services Docker Images

- The main component is called the Platform and is implemented in the alfresco.war web application.
- It provides the repository where content is stored plus all the associated content services.
- Alfresco Share provides a web client interface (that is a User Interface, UI) for the repository and is implemented as the share.war web application.
- Share makes it easy for users to manage their sites, documents, users and so on.
- The search functionality is implemented on top of Apache Solr 6 and provides the indexing of all content, which enables powerful search functionality.

Alfresco Content Services Docker Images

- There are also mobile clients that will access the content via ReST APIs provided by the platform.
- Most Content Services projects will implement a domain specific web client based on the Alfresco Application Development Framework (ADF).
- It gives you full freedom to design a content and process web client supporting exactly the use cases needed by the domain.
- Giving the end-users the best possible experience.

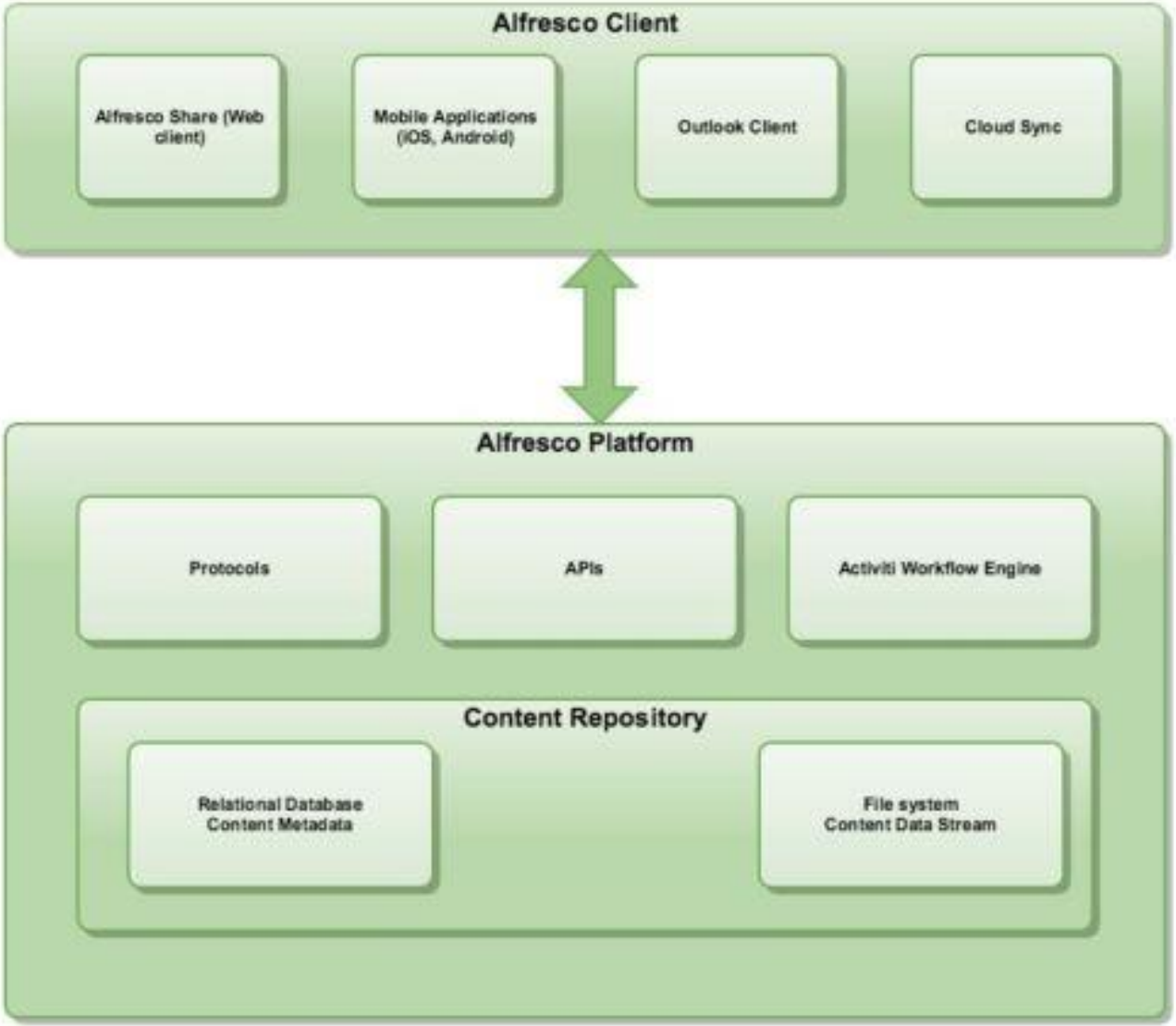
Alfresco Content Services Docker Images

- The Platform and UI components run in the same Apache Tomcat web application server.
- The Search component runs in its own Jetty web application server.
- The Platform is usually also integrated with a Directory Server (LDAP) to be able to sync users and groups with Content Services.
- And most installations also integrates with an SMTP server so the Platform can send emails, such as site invitations.
- The Platform also contains numerous APIs, such as the Java foundation API and the ReST API.

Alfresco Content Services Docker Images

- Alfresco provides several extension points to allow you to customize Content Services.
- These extensions points have various formats, but include:
  - Platform extension points and detailed architecture
  - Share extension points and detailed architecture
  - REST API.
  - Public Java API

Alfresco Platform Architecture



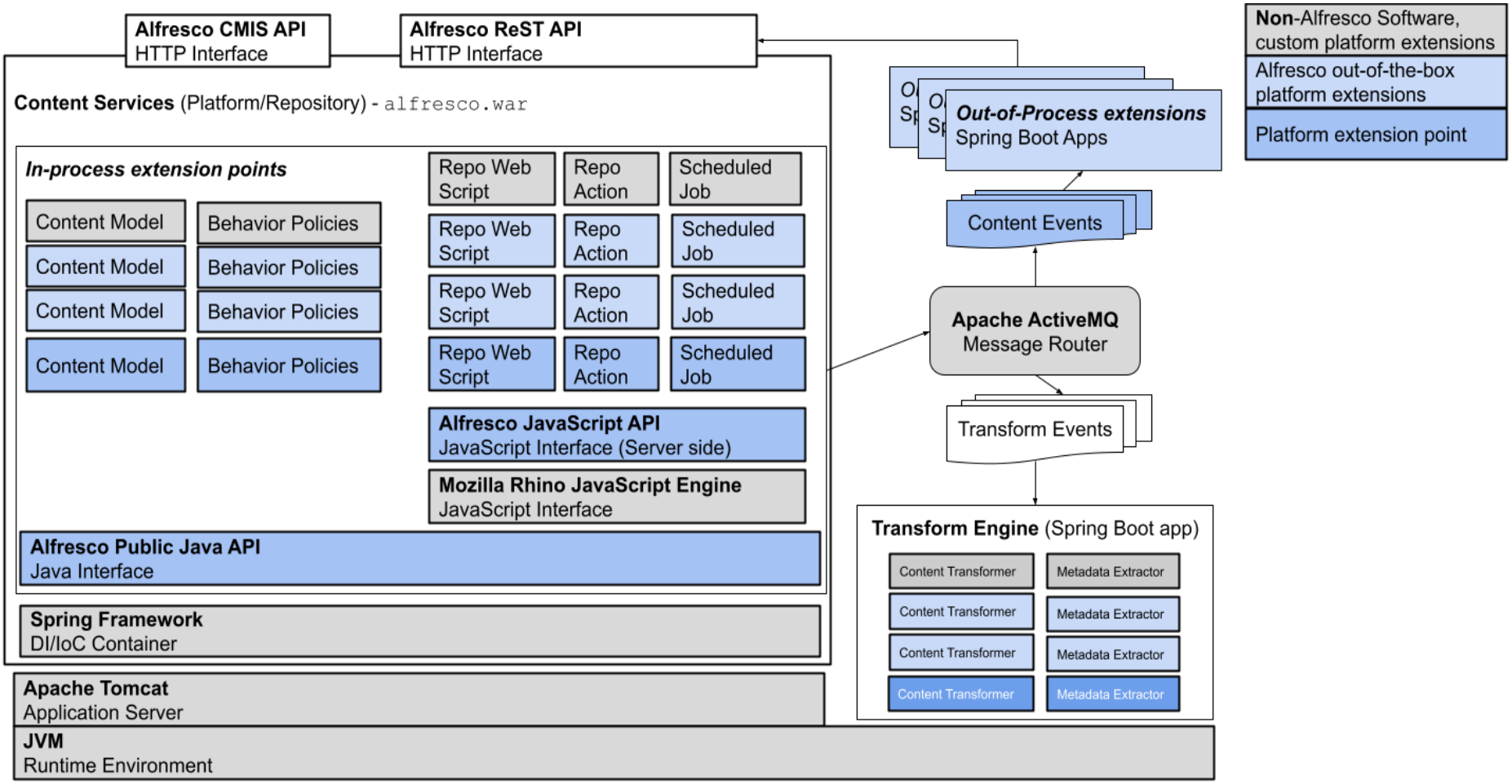
Alfresco Platform Architecture

- Layered architecture
- Components/Services
- Spring based
- Java based

Alfresco Platform Architecture

- The platform consists of the repository and all services, developer extension points, and APIs, such as the REST API.
- The repository provides storage for documents and other content.
- The content metadata is stored in a relational database, while the content itself is stored directly on the file system.
- The relationships between content items, and their various properties (metadata) are defined in one or more content models.

Alfresco Platform Architecture



Alfresco Platform Architecture

- Content models can be thought of as describing types of content and the relationships between pieces of content.
- For example, there is a relationship between a content that has a container functionality (that is, folder), and the piece of content contained within it (that is, sub-folders and files).
- There might be constraints defined in the content model, such as a content type cannot contain other content unless it is a container type.

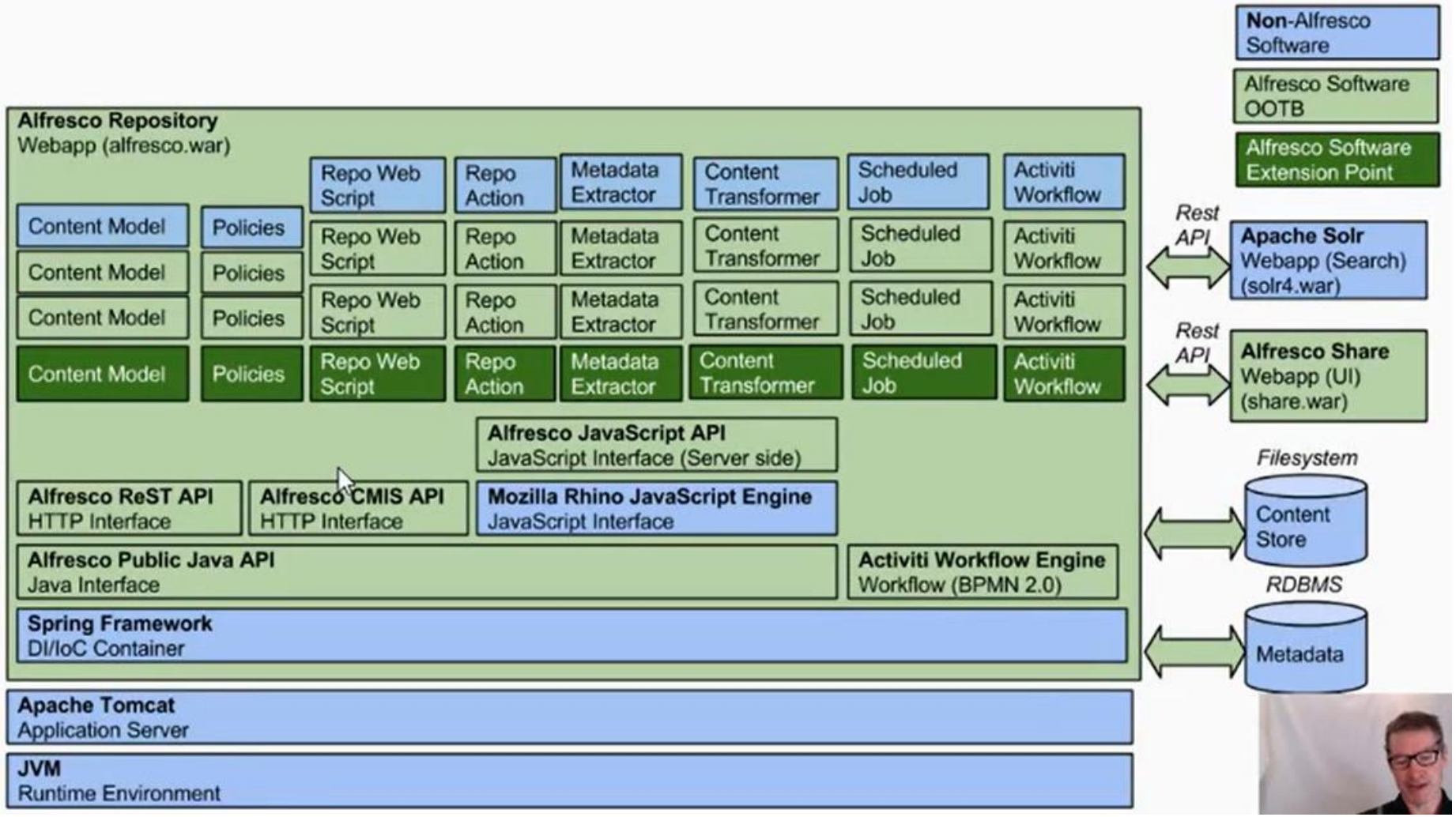
Alfresco Platform Architecture

- The platform provides a wide range of content-related services.
- These include core services such as the Node Service, and the Version Service.
- Thumbnail Service (for creating thumbnail images and renditions of documents).
- The Site Service used for creating and managing sites in the Share application.
- Tagging Service, which provides the ability to tag content with keywords.

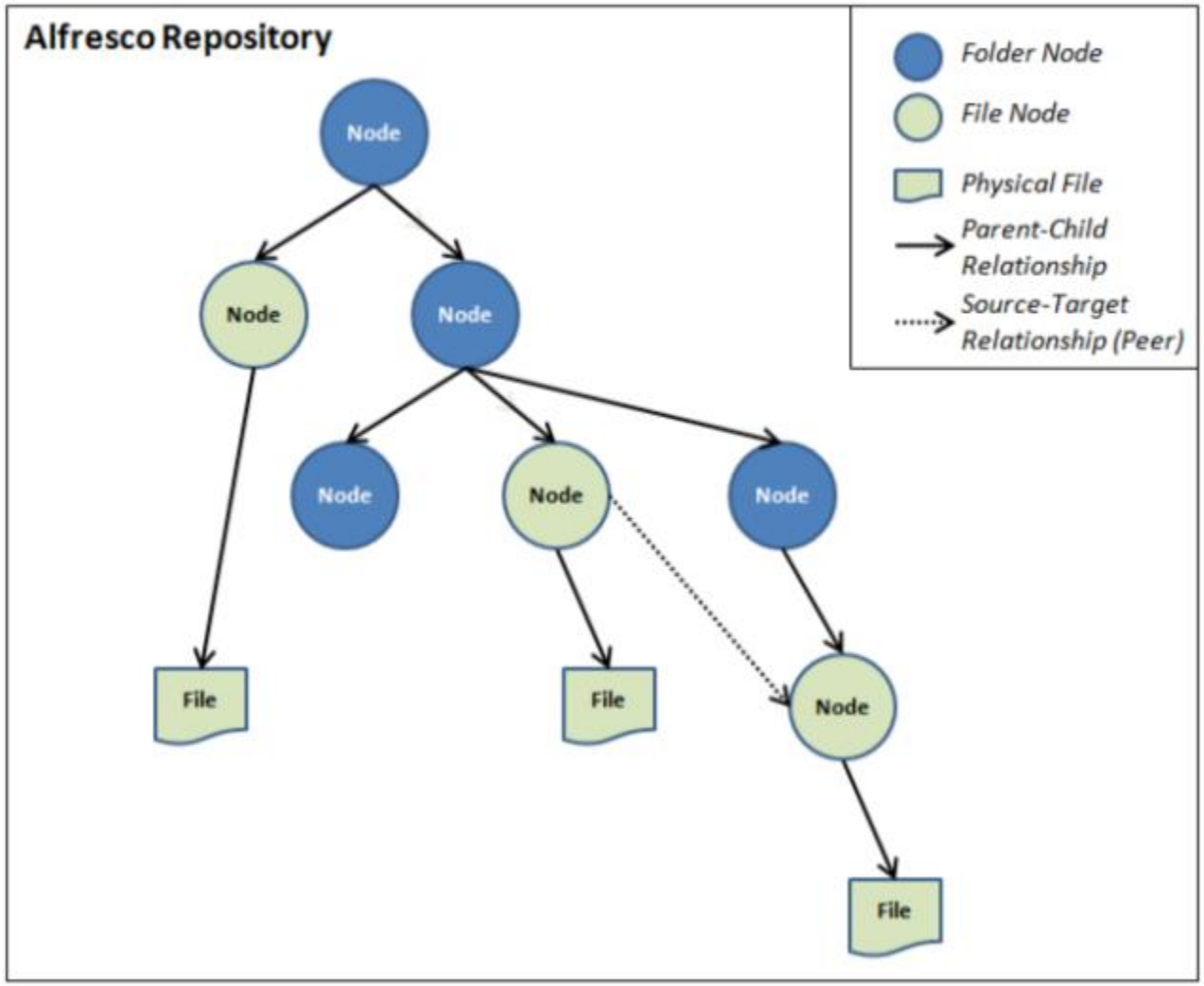
Alfresco Platform Architecture

- These services are implemented in Java, and expose an API described by the Public Java API.
- The platform is highly extensible.
- We can write extensions in Java, JavaScript, and FreeMarker.
- We can write client applications in any language using the ReST API.
- We can create new content models that define new content types, metadata, and relationships.
- We can define custom actions that the repository will carry out when certain events happen (such as when new content is added to the repository).
- We can even create entirely new services, if required.

Alfresco Repository Architecture



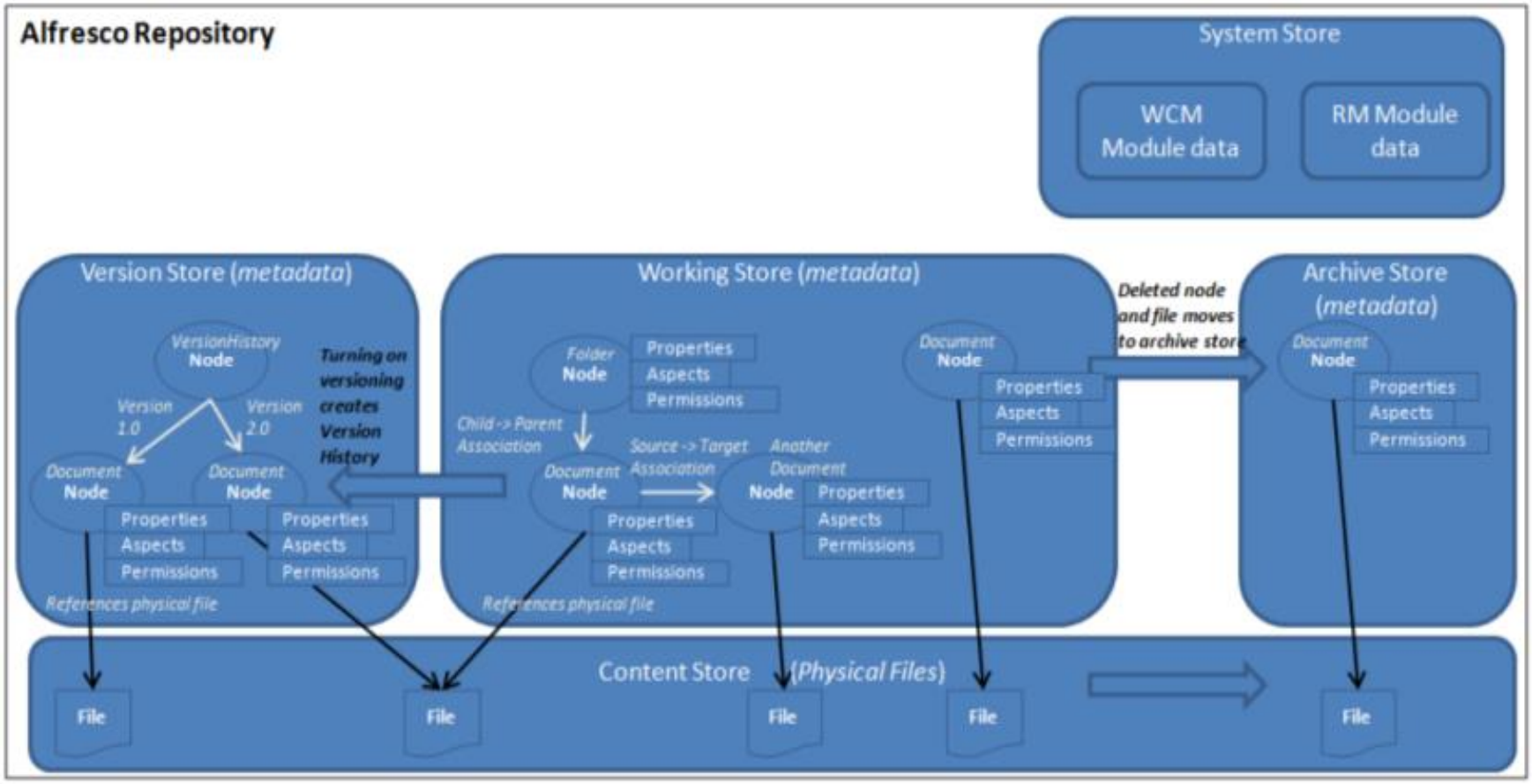
Alfresco Repository Concepts



Alfresco Repository Concepts

- The repository implements services including:
  - Definition of content structure (modeling)
  - Creation, modification, and deletion of content, associated metadata, and relationships
  - Query of content
  - Access control on content (permissions)
  - Versioning of content
  - Content renditions
  - Locking
  - Events
  - Audits
  - Import/Export
  - Multilingual
  - Rules/Actions

Alfresco Repository Store View



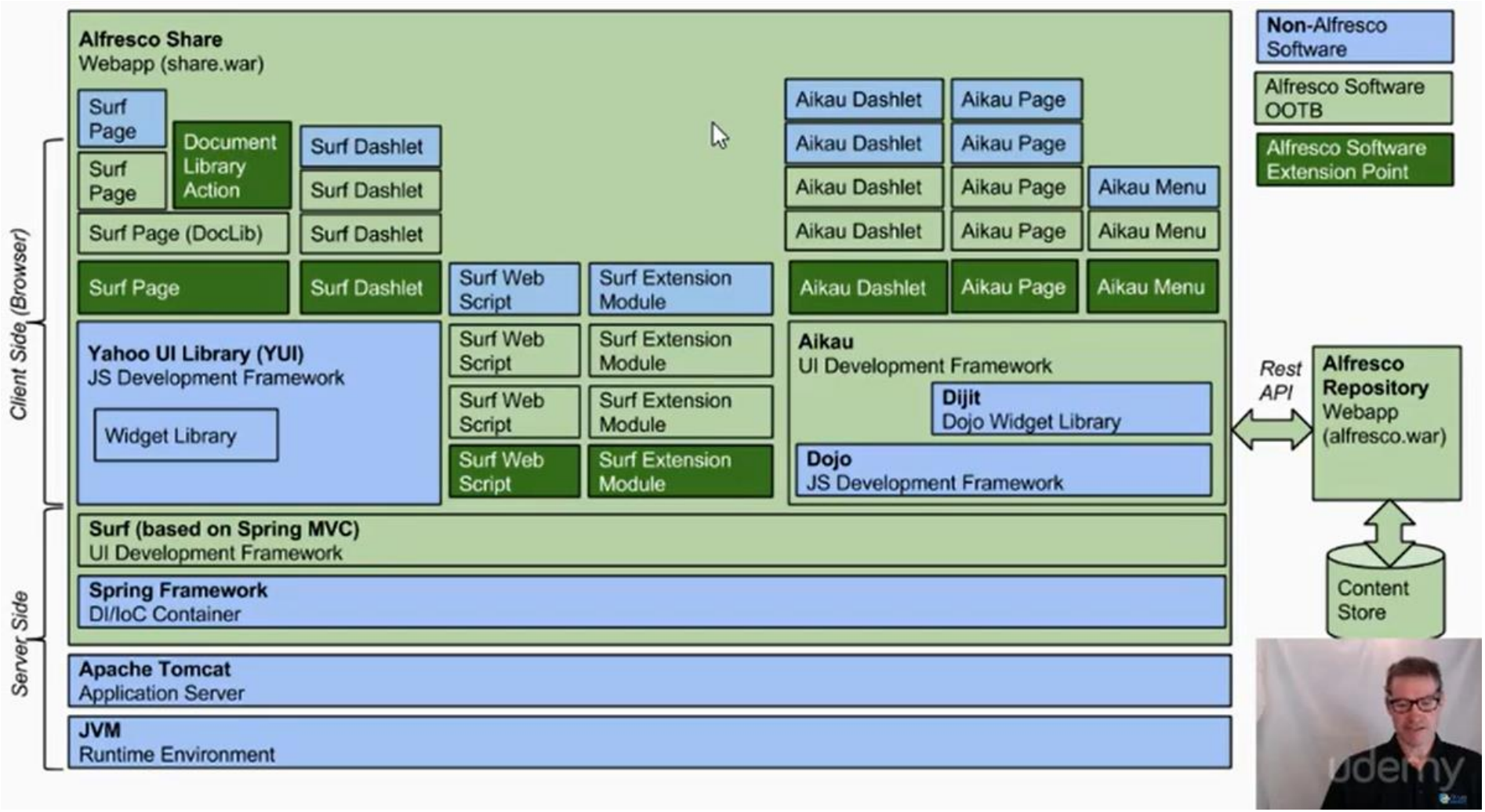
Alfresco Repository Store View

- The Working Store (workspace://SpacesStore) contains the metadata for all active/live nodes in the Repository.
- This store is implemented using a database (RDBMS).
- The Content Store contains the physical files uploaded to the Repository.
- It is located in the {Alfresco install dir}/alf_data/contentstore directory on the filesystem by default.
- But can also be configured to use other storage systems, for example, Amazon S3.
- It is also possible to define content store policies for storing files on different storage systems, effectively defining more than one physical content store.

Alfresco Repository Store View

- Whenever a node is deleted, the metadata for the node is moved to the Archive Store (archive://SpacesStore), which uses the configured database.
- The physical file for a deleted node is moved (by default after 14 days) to the {Alfresco install dir}/alf_data/contentstore.deleted directory, where it stays indefinitely.
- A clean-up job can be configured to remove the file at a certain point in time (referred to as eager clean-up).
- When the versionable aspect is applied to a node, a version history is created in the Version Store (workspace://version2Store).
- Versioned node metadata is stored in the database and files remain in the {Alfresco install dir}/alf_data/contentstore directory.
- Versioning is not applicable to folder nodes.
- The System Store is used to save information about installed Alfresco Content Services extension modules.

Alfresco Share UI Architecture



Alfresco Share UI Architecture

- Alfresco Share (share.war) is a web application that runs on the Java Platform.
- In a development environment it is usually deployed and run on top of Apache Tomcat.
- Share is built up of a main menu that leads to pages, like other web applications.
- There is one special page type called Dashboard that contains dashlets.
- A Dashboard page can be configured by the end-user, who can add, remove, and organize the dashlets on the page.

Alfresco Share UI Architecture

- Share pages and dashlets are implemented with something called web scripts.
- It is basically REST-based APIs.
- These APIs are called Surf web scripts when we are dealing with Alfresco Share.
- There is also repository web scripts that are used to extend the repository web application (alfresco.war) with REST-based APIs.
- Surf Web Scripts are referred to as Presentation Web Scripts and the repository web scripts as Data Web Scripts.

Alfresco Share UI Architecture

- Share web scripts, pages, and dashlets are implemented with a user interface (UI) development framework called Surf.
- This framework was originally developed by Alfresco, then donated to the Spring Source foundation, and finally brought back into Alfresco products.
- It provides a way of breaking a HTML page into re-usable component parts.
- Surf is built on top of the Spring Web MVC technology, which in turn uses the Spring Framework.

Alfresco Share UI Architecture

- Developers can also add completely new pages and dashlets to the Share UI when content should be viewed or handled in a specific way.
- Sometimes it is also required to modify existing pages.
- To customize the Share UI developers, use so called Extension Points.
- It is the way to inject new custom code that should alter the functionality of the Share web application.

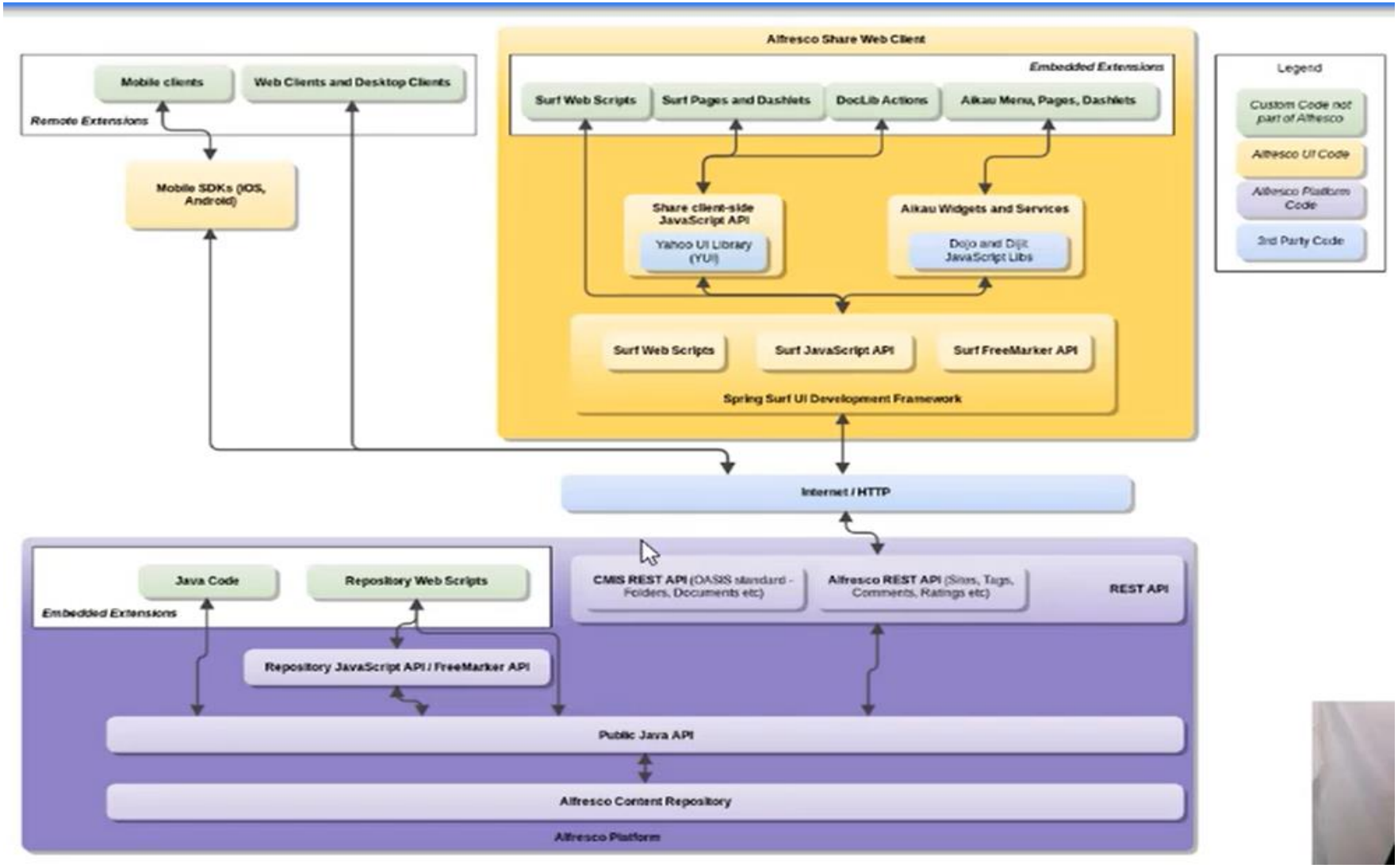
Alfresco Share UI Architecture

- Share gets the content that it should display in pages and dashlets by calling repository web scripts.
- It returns JSON or XML that can be incorporated into the presentation.
- The presentation is put together with two different kinds of JavaScript frameworks, Yahoo UI library (YUI) and Aikau, which is based on Dojo.
- An Aikau page is based on Surf, but it makes page composition much easier than with pure Surf pages.

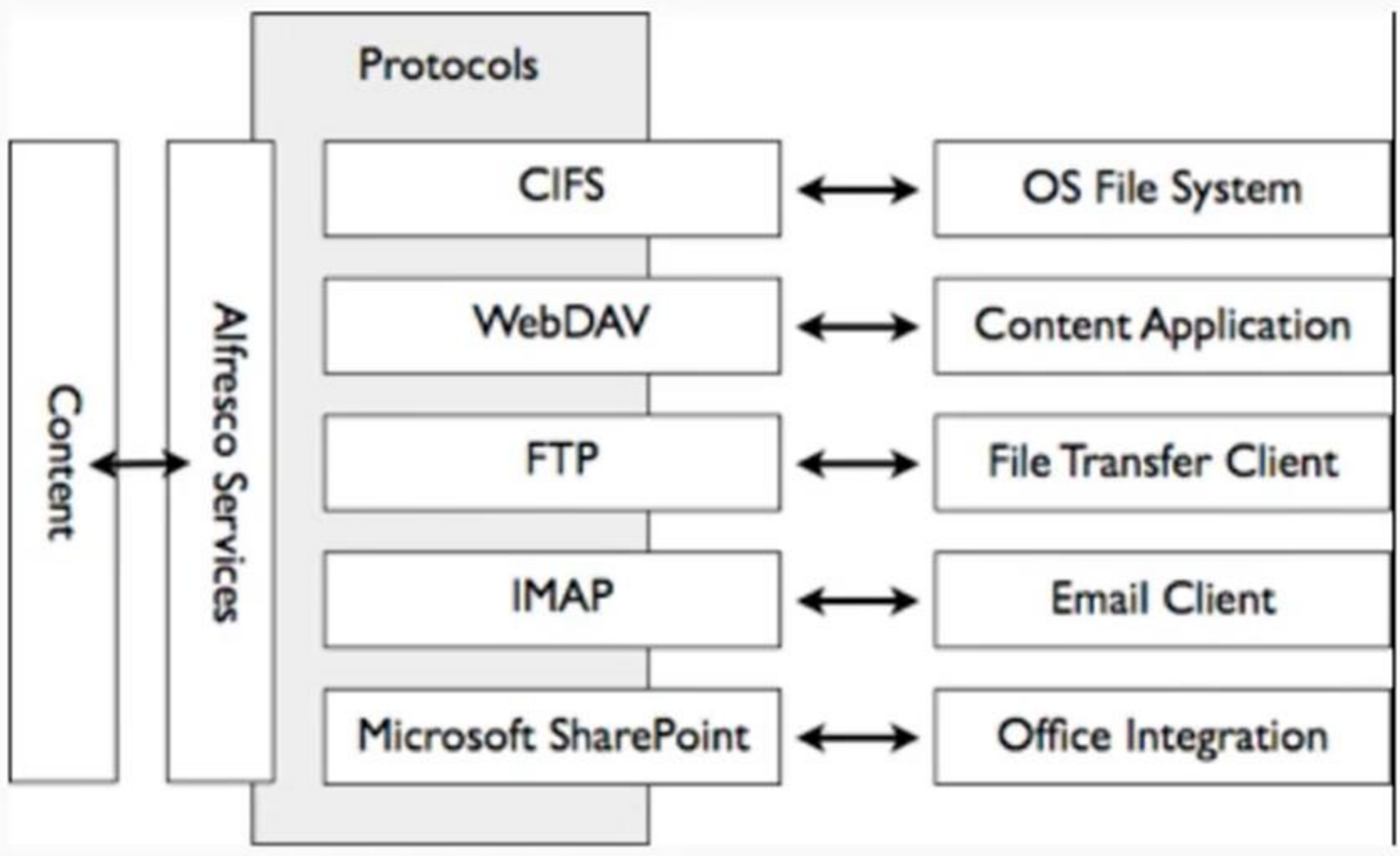
Alfresco Share UI Architecture

- We can focus solely on Aikau if the only thing we are going to do is add new stuff to the Share UI.
- Following can be converted to Aikau
  - Share Header Menu and Title (4.2)
  - Live Search (5.0)
  - Filtered Search Page (5.0)
  - Search Management Page (5.0)
  - Site Management Page (5.0)
  - Analytics and Reporting Widgets (5.0)
  - Document List prototype (5.0)

Alfresco APIs Architecture



Alfresco Protocols (Addition to HTTP)



Alfresco Content Services

- Alfresco Content Services 6.0 (or ACS) offers full-featured Enterprise Content Management (ECM).
- It supports enterprise-grade scalability, performance, and 24x7 support for business-critical content and compliance.
- It delivers a wide range of use cases such as content and governance services, contextual search and insight.
- It has ability to easily integrate with other applications.

Getting started with Alfresco SDK

- Prerequisites
- There are a number of software requirements for using Alfresco SDK 4.1:
  - Java Development Kit (JDK) - Version 11
  - Maven - Version 3.3
  - Docker - Latest stable version
  - JRebel (optional) for hot reloading of web resources, configuration, and classes

Generate your project from the archetypes

- After we've successfully configured Java and Maven, it's time to generate your project.
- **mvn archetype:generate -Dfilter=org.alfresco:**
- We'll be prompted to select the archetype we want.
- The previously available archetypes, alfresco-amp-archetype and share-amp-archetype will still show up as an option, however these archetypes are not part of Alfresco SDK 4.1.

Generate your project from the archetypes

The output looks something like this:

```
[INFO] Generating project in Interactive mode
[INFO] No archetype defined. Using maven-archetype-quickstart (org.apache.maven.archetypes:maven-archetype-quickstart:1.0)
Choose archetype:
1: remote -> org.alfresco.maven.archetype:alfresco-platform-jar-archetype (Sample project with full support for lifecycle an
2: remote -> org.alfresco.maven.archetype:alfresco-share-jar-archetype (Share project with full support for lifecycle and ra
        Packages))
3: remote -> org.alfresco.maven.archetype:alfresco-allinone-archetype (Sample multi-module project for All-in-One developmen
...
```

2. Select one of the following archetypes:

- `org.alfresco.maven.archetype:alfresco-allinone-archetype`
- `org.alfresco.maven.archetype:alfresco-platform-jar-archetype`
- `org.alfresco.maven.archetype:alfresco-share-jar-archetype`

Generate your project from the archetypes

3. Choose the latest version, such as 4.1.0.

```
Choose org.alfresco.maven.archetype:alfresco-allinone-archetype version:
1: 2.0.0-beta-1
2: 2.0.0-beta-2
3: 2.0.0-beta-3
4: 2.0.0-beta-4
5: 2.0.0
6: 2.1.0
7: 2.1.1
8: 2.2.0
9: 3.0.0
10: 3.0.1
11: 4.0.0
12: 4.1.0
```

Generate your project from the archetypes

4. Next you will be prompted for additional values, like `groupId`, `artifactId`, and `package`, as shown below:

```
Define value for property 'groupId':
Define value for property 'artifactId':
[INFO] Using property: version = 1.0-SNAPSHOT
Define value for property 'package':
```

5. After you have specified the information according to your project, a final confirmation will appear.

```
Confirm properties configuration:
groupId: com.acme
artifactId: my-all-in-one
version: 1.0-SNAPSHOT
package: com.acme
Y: :
```

6. Press **Y** and then press **Enter**.

Generate your project from the archetypes

If everything has been configured correctly, you should see something similar to this:

```
[INFO] ----------------------------------------------------------------------------
[INFO] Using following parameters for creating project from Archetype: alfresco-allinone-archetype:4.1.0-SNAPSHOT
[INFO] ----------------------------------------------------------------------------
[INFO] Parameter: groupId, Value: com.acme
[INFO] Parameter: artifactId, Value: my-all-in-one
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.acme
[INFO] Parameter: packageInPathFormat, Value: com/acme
[INFO] Parameter: package, Value: com.acme
[INFO] Parameter: groupId, Value: com.acme
[INFO] Parameter: artifactId, Value: my-all-in-one
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parent element not overwritten in /Users/Alfresco/my-all-in-one/my-all-in-one-platform/pom.xml
[INFO] Parent element not overwritten in /Users/Alfresco/my-all-in-one/my-all-in-one-share/pom.xml
[INFO] Parent element not overwritten in /Users/Alfresco/my-all-in-one/my-all-in-one-integration-tests/pom.xml
[INFO] Parent element not overwritten in /Users/Alfresco/my-all-in-one/my-all-in-one-platform-docker/pom.xml
[INFO] Parent element not overwritten in /Users/Alfresco/my-all-in-one/my-all-in-one-share-docker/pom.xml
[INFO] Executing META-INF/archetype-post-generate.groovy post-generation script
[INFO] Project created from Archetype in dir: /Users/Alfresco/my-all-in-one
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 04:11 min
[INFO] Finished at: 2019-01-10T16:21:46+01:00
[INFO] Final Memory: 17M/1024M
[INFO] ------------------------------------------------------------------------
```

Generate your project from the archetypes

7. You have successfully generated your first SDK 4.1 project.

Inside the project, you will find the `run.bat` and `run.sh` scripts. These are convenience scripts for you to quickly compile / test / run your project.

In the terminal window, use:

- `./run.sh build_start` for Mac OS X or Linux.
- `run.bat build_start` for Windows.

If this is the first time you are doing this, it will take a while for Maven to download all the required dependencies and for Docker to download all the required images.

Configuring Alfresco

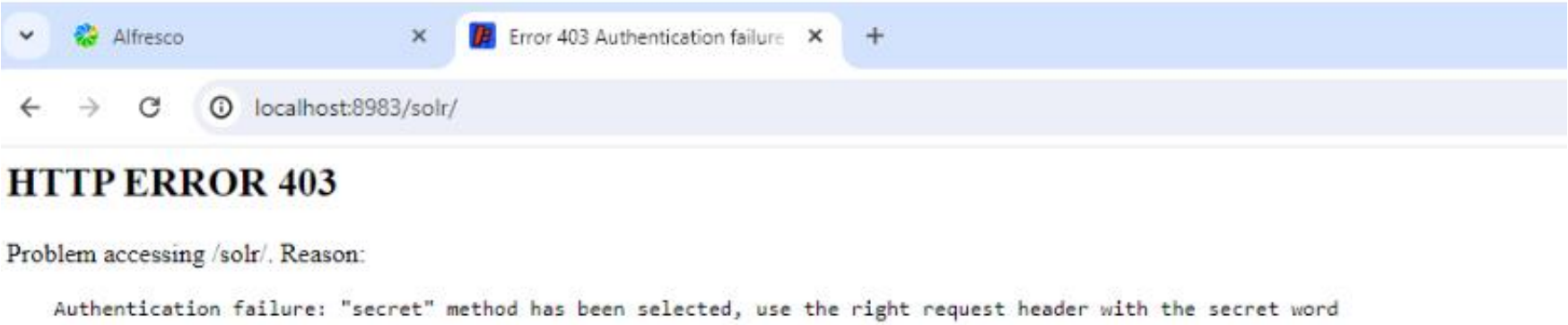
- Important Directories
- Alfresco Repository Configuration Files
- Alfresco Repository Configuration Files Order of Importance
- Alfresco Share Configuration Files
- Alfresco Repository Share Files Order of Importance
- Solr Configuration Files

Configuration Directories

<classpathRoot> directory (configuring **alfresco.war**)

- The <classpathRoot> denotes a directory whose contents are automatically added to the start of your application server's classpath
- This directory contains the main configuration file for the Alfresco Repository: **alfresco-global.properties**
- The location of this directory varies depending on your application server:
- *Windows*:
  - (Tomcat) C:\Alfresco\tomcat\shared\classes
  - (JBoss) C:\jboss\server\default\conf
  - (WebLogic) C:\bea\user_projects\domains\alf_domain
- *Linux*:
  - (Tomcat) /opt/alfresco/tomcat/shared/classes/
  - (JBoss) /jboss/server/default/conf
  - (WebLogic) <bea>/user_projects/domains/alf_domain

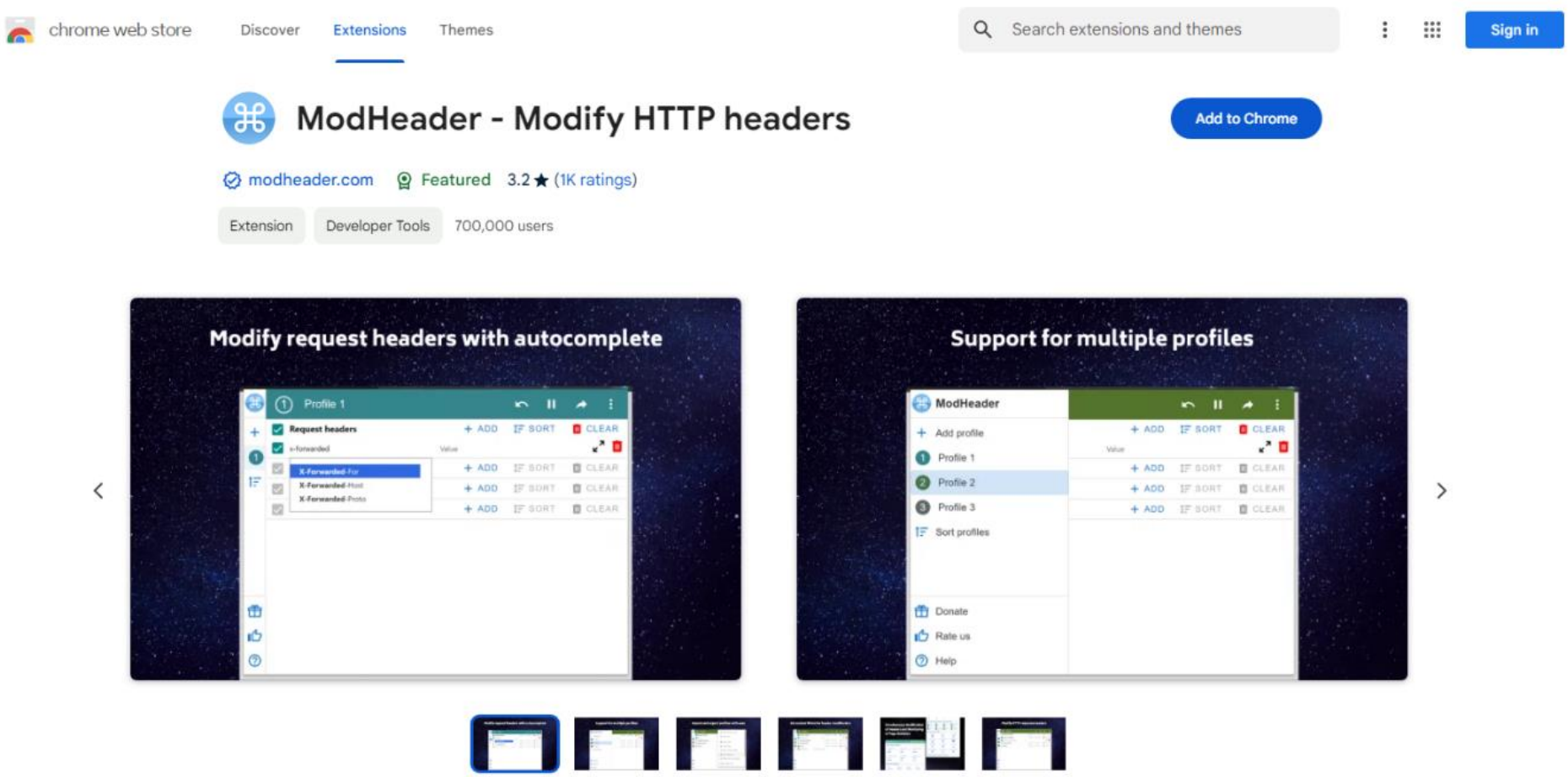
Solr 6


Alfresco
Error 403 Authentication failure
localhost:8983/solr/
HTTP ERROR 403
Problem accessing /solr/. Reason:
Authentication failure: "secret" method has been selected, use the right request header with the secret word

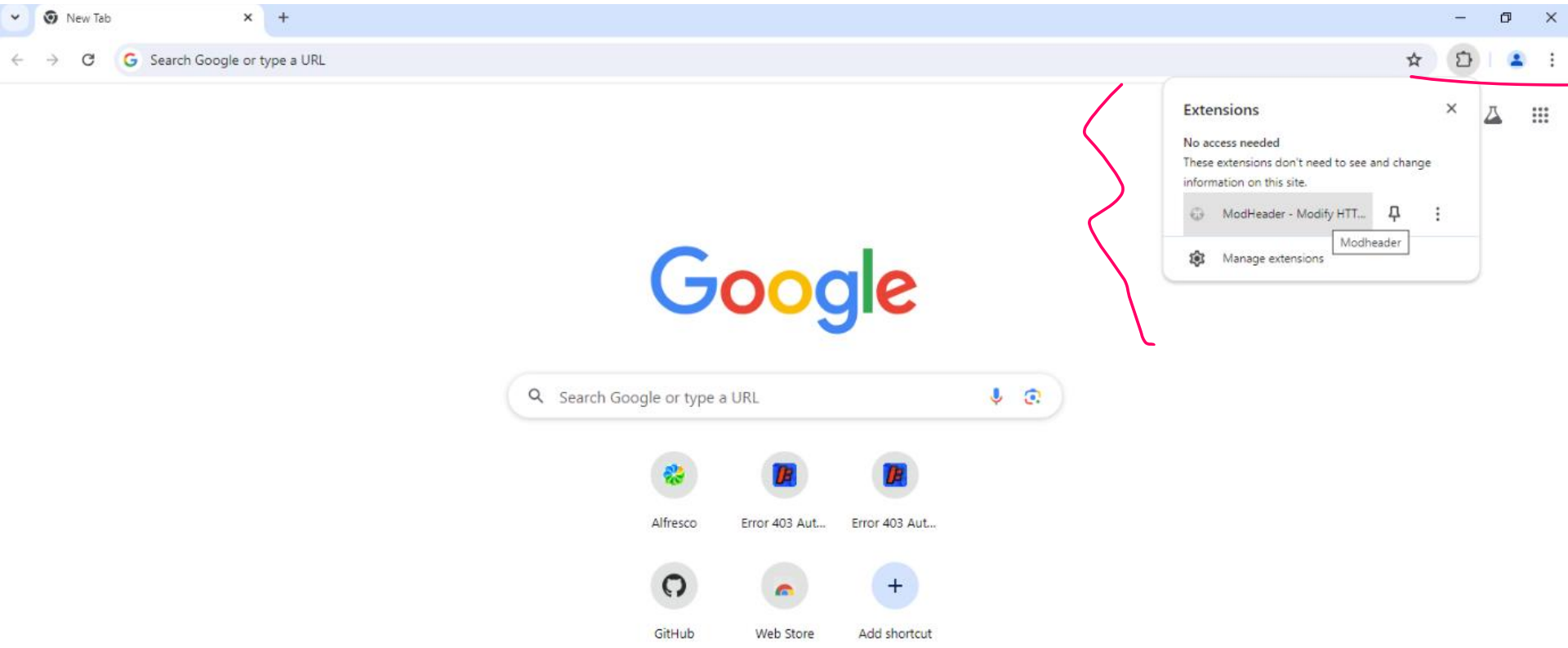
Solr 6

- Search Services administration http://localhost:8083/solr
- For solr
- For Chrome, FireFox, Opera and Edge:
  - * Install ModHeader extension.
  - * Add the X-Alfresco-Search-Secret header with the secret value.

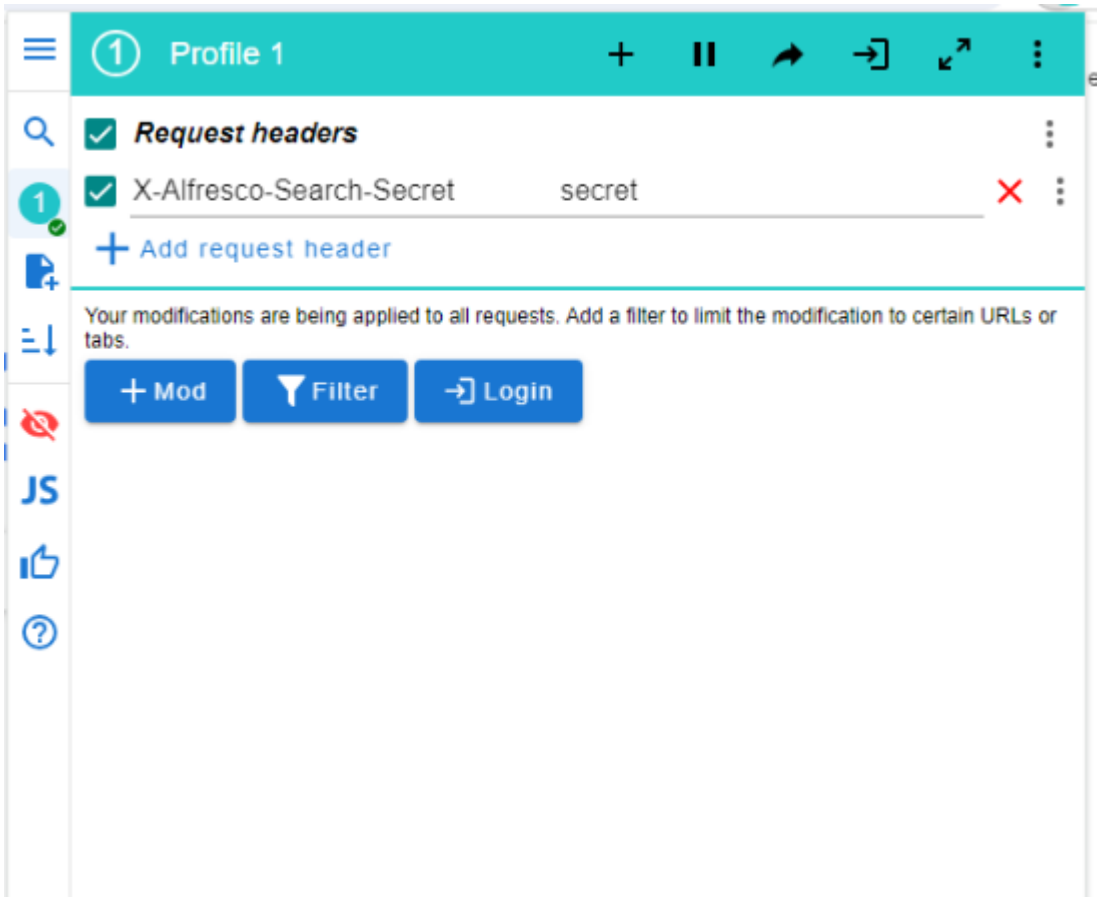
Solr 6


chrome web store
Discover
Extensions
Themes
Search extensions and themes
Sign in
ModHeader - Modify HTTP headers
Add to Chrome
modheader.com
Featured
3.2 ★ (1K ratings)
Extension
Developer Tools
700,000 users
Modify request headers with autocomplete
Support for multiple profiles
Profile 1
Request headers
ADD
SORT
CLEAR
ModHeader
Add profile
Profile 1
Profile 2
Profile 3
Sort profiles
Donate
Rate us
Help
Activate Windows
Go to Settings to activate Windows.

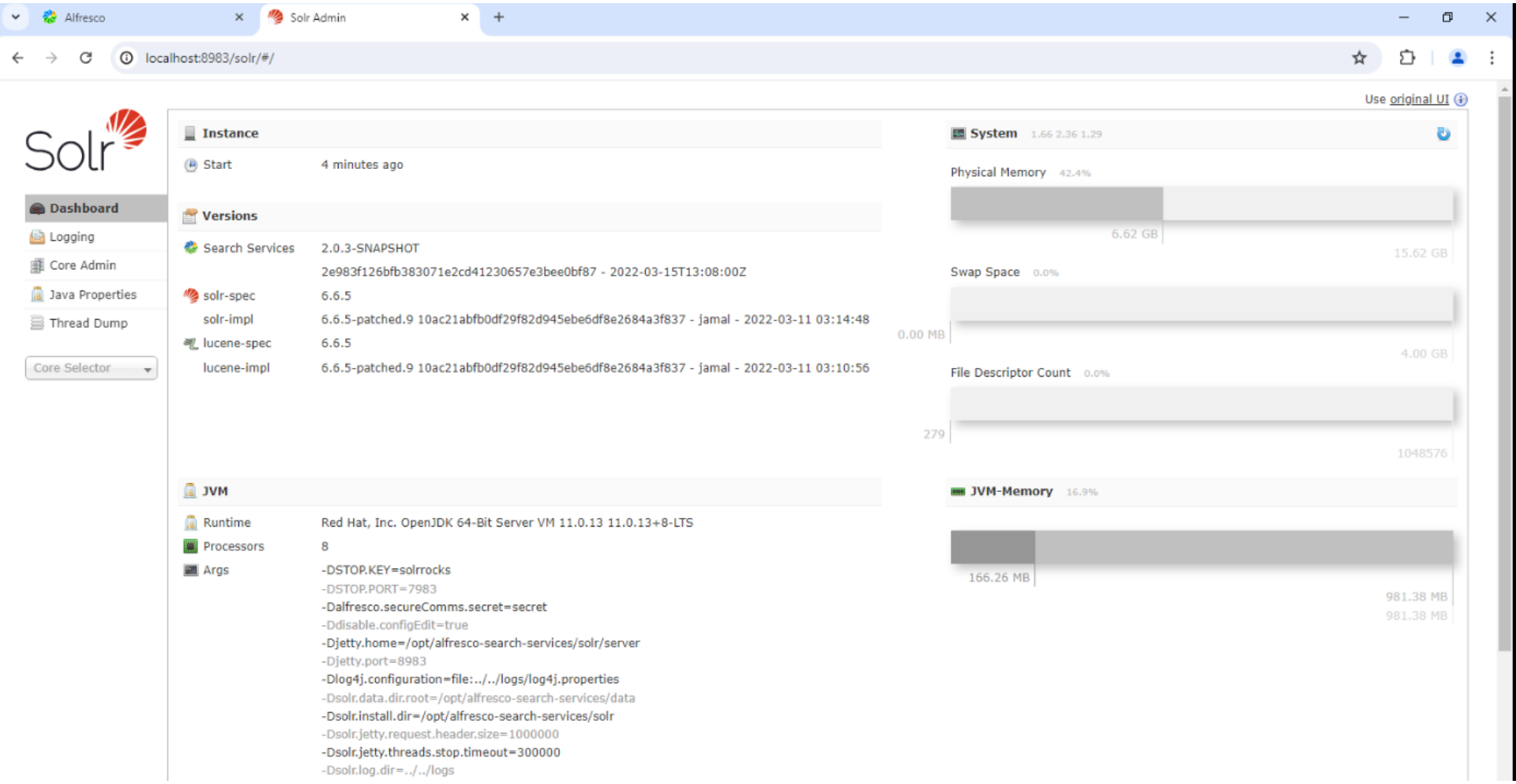
Solr 6


New Tab
Search Google or type a URL
Extensions
No access needed
These extensions don't need to see and change information on this site.
ModHeader - Modify HTT...
Modheader
Manage extensions
Google
Search Google or type a URL
Alfresco
Error 403 Aut...
Error 403 Aut...
GitHub
Web Store
Add shortcut

Solr 6


Profile 1
Request headers
X-Alfresco-Search-Secret
secret
Add request header
Your modifications are being applied to all requests. Add a filter to limit the modification to certain URLs or tabs.
Mod
Filter
Login
JS

Solr 6


Alfresco
Solr Admin
localhost:8983/solr/#/
Use original UI
Solr
Dashboard
Logging
Core Admin
Java Properties
Thread Dump
Core Selector
Instance
Start
4 minutes ago
Versions
Search Services
2.0.3-SNAPSHOT
2e983f126bfb383071e2cd41230657e3bee0bf87 - 2022-03-15T13:08:00Z
solr-spec
6.6.5
solr-impl
6.6.5-patched.9 10ac21abfb0df29f82d945ebe6df8e2684a3f837 - jamal - 2022-03-11 03:14:48
lucene-spec
6.6.5
lucene-impl
6.6.5-patched.9 10ac21abfb0df29f82d945ebe6df8e2684a3f837 - jamal - 2022-03-11 03:10:56
JVM
Runtime
Red Hat, Inc. OpenJDK 64-Bit Server VM 11.0.13 11.0.13+8-LTS
Processors
8
Args
-DSTOP.KEY=solrrocks
-DSTOP.PORT=7983
-Dalfresco.secureComms.secret=secret
-Ddisable.configEdit=true
-Djetty.home=/opt/alfresco-search-services/solr/server
-Djetty.port=8983
-Dlog4j.configuration=file:../../logs/log4j.properties
-Dsolr.data.dir.root=/opt/alfresco-search-services/data
-Dsolr.install.dir=/opt/alfresco-search-services/solr
-Dsolr.jetty.request.header.size=1000000
-Dsolr.jetty.threads.stop.timeout=300000
-Dsolr.log.dir=../../logs
System
1.66 2.36 1.29
Physical Memory
42.4%
6.62 GB
15.62 GB
Swap Space
0.0%
0.00 MB
4.00 GB
File Descriptor Count
0.0%
279
1048576
JVM-Memory
16.9%
166.26 MB
981.38 MB
981.38 MB

Configuration Directories

```
@3baa3b4ca64c:/usr/local/tomcat/webapps
C:\Windows\System32>docker container ls -a
CONTAINER ID   IMAGE                                                   COMMAND                  CREATED       STATUS       PORTS
                                                                                   NAMES
c22b47626eee   alfresco-content-services-alfrescoall:development   "catalina.sh run -se…"   11 days ago   Up 3 hours   0.0.0.0:8080->80
80/tcp, 8000/tcp, 10001/tcp, 0.0.0.0:8888->8888/tcp                                docker-alfrescoall-acs-1
cf7080207f19   alfresco/alfresco-search-services:2.0.3             "/bin/sh -c '$DIST_D…"   11 days ago   Up 3 hours   0.0.0.0:8983->89
83/tcp, 10001/tcp                                                                  docker-alfrescoall-ass-1
3baa3b4ca64c   alfresco-share-alfrescoall:development              "/usr/local/tomcat/s…"   11 days ago   Up 3 hours   8000/tcp, 0.0.0.
0:8180->8080/tcp, 0.0.0.0:9898->8888/tcp                                           docker-alfrescoall-share-1
1108ab7141fe   alfresco/alfresco-activemq:5.16.1                   "/bin/sh -c './init.…"   11 days ago   Up 3 hours   0.0.0.0:5672->56
72/tcp, 0.0.0.0:8161->8161/tcp, 0.0.0.0:61613->61613/tcp, 0.0.0.0:61616->61616/tcp   docker-alfrescoall-activemq-1
88ad16c788a3   postgres:9.6                                        "docker-entrypoint.s…"   11 days ago   Up 3 hours   0.0.0.0:5555->54
32/tcp                                                                             docker-alfrescoall-postgres-1

C:\Windows\System32>docker exec -it 3baa3b4ca64c  bash
[root@3baa3b4ca64c tomcat]# ls
BUILDING.txt     NOTICE         RUNNING.txt   alfresco.log.2024-06-30  alfresco.log.2024-07-04  bin   logs            temp
CONTRIBUTING.md  README.md      alfresco-mmt  alfresco.log.2024-07-02  alfresco.log.2024-07-05  conf  native-jni-lib  webapps
LICENSE          RELEASE-NOTES  alfresco.log  alfresco.log.2024-07-03  amps_share               lib   shared          work
[root@3baa3b4ca64c tomcat]# cd webapps
[root@3baa3b4ca64c webapps]# yum install nano
Rocky Linux 8 - AppStream                                                  4.8 kB/s | 4.8 kB     00:00
Rocky Linux 8 - AppStream                                                  3.7 MB/s |  11 MB     00:02
Rocky Linux 8 - BaseOS                                                     5.2 kB/s | 4.3 kB     00:00
Rocky Linux 8 - BaseOS                                                     1.9 MB/s | 5.0 MB     00:02
Rocky Linux 8 - Extras                                                     4.2 kB/s | 3.1 kB     00:00
Rocky Linux 8 - Extras                                                      16 kB/s |  14 kB     00:00
Dependencies resolved.
================================================================================================================
 Package                    Architecture                Version                          Repository                    Size
================================================================================================================
Installing:
 nano                       x86_64                      2.9.8-1.el8                      baseos                       579 k
```

Configuration Directories

@c22b47626eee:/usr/local/tomcat/shared/classes

```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>docker container ls -a
CONTAINER ID   IMAGE                                                   COMMAND                  CREATED       STATUS       PORTS
                                                                                    NAMES
c22b47626eee   alfresco-content-services-alfrescoall:development   "catalina.sh run -se…"   11 days ago   Up 3 hours   0.0.0.0:8080->80
80/tcp, 8000/tcp, 10001/tcp, 0.0.0.0:8888->8888/tcp                                 docker-alfrescoall-acs-1
cf7080207f19   alfresco/alfresco-search-services:2.0.3             "/bin/sh -c '$DIST_D…"   11 days ago   Up 3 hours   0.0.0.0:8983->89
83/tcp, 10001/tcp                                                                   docker-alfrescoall-ass-1
3baa3b4ca64c   alfresco-share-alfrescoall:development              "/usr/local/tomcat/s…"   11 days ago   Up 3 hours   8000/tcp, 0.0.0.
0:8180->8080/tcp, 0.0.0.0:9898->8888/tcp                                            docker-alfrescoall-share-1
1108ab7141fe   alfresco/alfresco-activemq:5.16.1                   "/bin/sh -c './init.…"   11 days ago   Up 3 hours   0.0.0.0:5672->56
72/tcp, 0.0.0.0:8161->8161/tcp, 0.0.0.0:61613->61613/tcp, 0.0.0.0:61616->61616/tcp   docker-alfrescoall-activemq-1
88ad16c788a3   postgres:9.6                                        "docker-entrypoint.s…"   11 days ago   Up 3 hours   0.0.0.0:5555->54
32/tcp                                                                              docker-alfrescoall-postgres-1

C:\Windows\System32>docker exec -it c22b47626eee  bash
[alfresco@c22b47626eee tomcat]$ ls
BUILDING.txt     LICENSE  README.md      RUNNING.txt  alfresco-mmt  bin   lib   native-jni-lib  temp     work
CONTRIBUTING.md  NOTICE   RELEASE-NOTES  alf_data     amps          conf  logs  shared          webapps
[alfresco@c22b47626eee tomcat]$ cd shared
[alfresco@c22b47626eee shared]$ ls
classes
[alfresco@c22b47626eee shared]$ cd classes
[alfresco@c22b47626eee classes]$ ls
alfresco  alfresco-global.properties
[alfresco@c22b47626eee classes]$
```

31°C Mostly cloudy Search ENG IN 20:35 11-07-2024

Configuration Directories

@c22b47626eee:/usr/local/tomcat/shared/classes

```
# alfresco/tomcat/shared/classes/alfresco-global.properties.
#
index.subsystem.name=solr6
solr.host=alfrescoall-ass
solr.port=8983
solr.secureComms=secret
solr.sharedSecret=secret

db.username=alfresco
db.password=alfresco
db.pool.initial=10
db.pool.max=100

db.url=jdbc:postgresql://alfrescoall-postgres:5432/alfresco

# File servers related properties
# For local runs we disable CIFS and FTP
cifs.enabled=false
ftp.enabled=false

csrf.filter.enabled=false

# Embedded broker without persistence
messaging.broker.url=vm://localhost?broker.persistent=false

# Disable ATS
transform.service.enabled=false
local.transform.service.enabled=false
legacy.transform.service.enabled=false
[alfresco@c22b47626eee classes]$
```

Alfresco Global Properties

- The global properties file (alfresco-global.properties) is used to detect extended properties.
- For example, when you install Content Services, many of the installation settings are saved in the global properties file.
- The global properties file is used to detect the extended properties.
- We can use the global properties to set all your property settings; whenever we make a change, we must restart the server to apply those changes.

Configuration Directories

<extension> directory (configuring **alfresco.war**)

- The <extension> directory is where you store Spring configuration files that extend and override the system configuration:
  - Content Model definitions
  - Workflow process definitions
- By default this directory will only contain sample config files that are not active
- This directory can be found at:
  - **<classpathRoot>\alfresco\extension**
  - (/opt/alfresco/tomcat/shared/classes/alfresco/extension)

Configuration Directories

@c22b47626eee:/usr/local/tomcat/shared/classes/alfresco/extension

```
C:\Windows\System32>docker exec -it c22b47626eee /bin/bash
[alfresco@c22b47626eee tomcat]$ ls
BUILDING.txt     LICENSE  README.md      RUNNING.txt  alfresco-mmt  bin   lib   native-jni-lib  temp     work
CONTRIBUTING.md  NOTICE   RELEASE-NOTES  alf_data     amps          conf  logs  shared          webapps
[alfresco@c22b47626eee tomcat]$ cd shared
[alfresco@c22b47626eee shared]$ ls
classes
[alfresco@c22b47626eee shared]$ cd classes
[alfresco@c22b47626eee classes]$ ls
alfresco  alfresco-global.properties
[alfresco@c22b47626eee classes]$ cd alfresco
[alfresco@c22b47626eee alfresco]$ ls
extension
[alfresco@c22b47626eee alfresco]$ cd extension
[alfresco@c22b47626eee extension]$ ls
dev-log4j2.properties  disable-webscript-caching-context.xml  keystore  mimetypes  transform
[alfresco@c22b47626eee extension]$
```

Configuration Directories

<web-extension> directory (configuring **share.war**)

- The <web-extension> directory is where you store Spring configurations that extend and override the Alfresco Share UI configuration
- This directory contains the main configuration file for Alfresco Share: **share-config-custom.xml**
- This directory can be found at:
  - **<classpathRoot>/alfresco/web-extension**
  - (/opt/alfresco/tomcat/shared/classes/alfresco/web-extension)

Configuration Directories

```
@3baa3b4ca64c:/usr/local/tomcat/shared/classes/alfresco/web-extension
C:\Windows\System32>docker exec -it 3baa3b4ca64c  bash
[root@3baa3b4ca64c tomcat]# ls
BUILDING.txt     NOTICE         RUNNING.txt   alfresco.log.2024-06-30  alfresco.log.2024-07-04  bin   logs            temp
CONTRIBUTING.md  README.md      alfresco-mmt  alfresco.log.2024-07-02  alfresco.log.2024-07-05  conf  native-jni-lib  webapps
LICENSE          RELEASE-NOTES  alfresco.log  alfresco.log.2024-07-03  amps_share               lib   shared          work
[root@3baa3b4ca64c tomcat]# cd shared
[root@3baa3b4ca64c shared]# ls
classes
[root@3baa3b4ca64c shared]# cd classes
[root@3baa3b4ca64c classes]# ls
alfresco
[root@3baa3b4ca64c classes]# cd alfresco
[root@3baa3b4ca64c alfresco]# ls
substituter.sh  web-extension
[root@3baa3b4ca64c alfresco]# cd web-extension
[root@3baa3b4ca64c web-extension]# ls
custom-slingshot-application-context.xml.sample  share-config-custom.xml  smartfolders-amp-actions-config.xml
[root@3baa3b4ca64c web-extension]#
```

Configuration Directories

<configRoot> directory (configuring **alfresco.war**)

- The <configRoot> directory is where the default configuration files are stored for the Alfresco Repository
- For Tomcat App Server <configRoot> is **<TOMCAT_HOME>/webapps/alfresco/WEB-INF** (/opt/alfresco/tomcat/webapps/alfresco/WEB-INF)
- **<configRoot>/classes/alfresco** contains the default configuration file for the Alfresco Repository: **repository.properties**
  - **/enterprise/repository.properties** (Enterprise edition only)
- And all default sub-system configuration files under **<configRoot>/classes/alfresco/subsystems**

Configuration Directories

```
@c22b47626eee:/usr/local/tomcat/webapps/alfresco/WEB-INF
exit

C:\Windows\System32>docker exec -it c22b47626eee /bin/bash
[alfresco@c22b47626eee tomcat]$ ls
BUILDING.txt     LICENSE  README.md      RUNNING.txt  alfresco-mmt  bin   lib   native-jni-lib  temp     work
CONTRIBUTING.md  NOTICE   RELEASE-NOTES  alf_data     amps          conf  logs  shared          webapps
[alfresco@c22b47626eee tomcat]$ cd webapps
[alfresco@c22b47626eee webapps]$ ls
ROOT  _vti_bin  alfresco  api-explorer
[alfresco@c22b47626eee webapps]$ cd alfresco
[alfresco@c22b47626eee alfresco]$ ls
META-INF  WEB-INF  admin  cmis-browser-app  cmisfs  css  error.jsp  favicon.ico  images  index.jsp  scripts
[alfresco@c22b47626eee alfresco]$ cd WEB-INF
[alfresco@c22b47626eee WEB-INF]$ ls
classes  cmis10  cmis11  lib  licenses  web-application-context.xml  web.xml
[alfresco@c22b47626eee WEB-INF]$ cd ..
[alfresco@c22b47626eee alfresco]$ cd ..
[alfresco@c22b47626eee webapps]$ cd ..
[alfresco@c22b47626eee tomcat]$ ls
BUILDING.txt     LICENSE  README.md      RUNNING.txt  alfresco-mmt  bin   lib   native-jni-lib  temp     work
CONTRIBUTING.md  NOTICE   RELEASE-NOTES  alf_data     amps          conf  logs  shared          webapps
[alfresco@c22b47626eee tomcat]$ cd webapps
[alfresco@c22b47626eee webapps]$ ls
ROOT  _vti_bin  alfresco  api-explorer
[alfresco@c22b47626eee webapps]$ cd alfesco
bash: cd: alfesco: No such file or directory
[alfresco@c22b47626eee webapps]$ cd alfresco
[alfresco@c22b47626eee alfresco]$ ls
META-INF  WEB-INF  admin  cmis-browser-app  cmisfs  css  error.jsp  favicon.ico  images  index.jsp  scripts
[alfresco@c22b47626eee alfresco]$ cd WEB-INF
[alfresco@c22b47626eee WEB-INF]$ ls
classes  cmis10  cmis11  lib  licenses  web-application-context.xml  web.xml
[alfresco@c22b47626eee WEB-INF]$
```

Configuration Directories

@c22b47626eee:/usr/local/tomcat/webapps/alfresco/WEB-INF/classes/alfresco

```
C:\Windows\System32>docker exec -it c22b47626eee /bin/bash
[alfresco@c22b47626eee tomcat]$ ls
BUILDING.txt     LICENSE  README.md      RUNNING.txt  alfresco-mmt  bin   lib   native-jni-lib  temp     work
CONTRIBUTING.md  NOTICE   RELEASE-NOTES  alf_data     amps          conf  logs  shared          webapps
[alfresco@c22b47626eee tomcat]$ cd webapps
[alfresco@c22b47626eee webapps]$ ls
ROOT  _vti_bin  alfresco  api-explorer
[alfresco@c22b47626eee webapps]$ cd alfresco
[alfresco@c22b47626eee alfresco]$ ls
META-INF  WEB-INF  admin  cmis-browser-app  cmisfs  css  error.jsp  favicon.ico  images  index.jsp  scripts
[alfresco@c22b47626eee alfresco]$ cd WEB-INF
[alfresco@c22b47626eee WEB-INF]$ ls
classes  cmis10  cmis11  lib  licenses  web-application-context.xml  web.xml
[alfresco@c22b47626eee WEB-INF]$ cd classes
[alfresco@c22b47626eee classes]$ ls
THIRD-PARTY.txt  alfresco  log4j2.properties  logging.properties  org
[alfresco@c22b47626eee classes]$ cd alfresco
[alfresco@c22b47626eee alfresco]$ ls
alfresco-jmxrmi.access    extension  subsystems  version.properties                 web-client-security-config.xml
alfresco-jmxrmi.password  module     templates   web-client-application-context.xml
[alfresco@c22b47626eee alfresco]$
```

Configuration Directories

1. (**jar**)/META-INF/**share-config-custom.xml** (EXTENSION MODULE CONFIGURATION)
2. <web-extension>/**share-config-custom.xml** (MAIN CONFIGURATION FILE)
3. <configRootShare>/classes/alfresco contains the default configuration files for the Alfresco Share:
   a. **share-config.xml** – main configuration
   b. **share-documentlibrary-config.xml** – specific configuration for the Document Library
   c. **share-form-config.xml** – UI forms configuration
   d. **share-workflow-form-config.xml** – Workflow task UI forms configuration
4. DEFAULT ALFRESCO CONFIGURATION - DO NOT CHANGE THESE FILES – COPY PROPERTY TO <web-extension>/share-config-custom.xml INSTEAD!

Activate Windows

Configuration Directories

@3baa3b4ca64c:/usr/local/tomcat/webapps/share/WEB-INF/classes/alfresco

```
C:\Windows\System32>docker exec -it 3baa3b4ca64c  bash
[root@3baa3b4ca64c tomcat]# ls
BUILDING.txt     NOTICE         RUNNING.txt   alfresco.log.2024-06-30  alfresco.log.2024-07-04  bin   logs            temp
CONTRIBUTING.md  README.md      alfresco-mmt  alfresco.log.2024-07-02  alfresco.log.2024-07-05  conf  native-jni-lib  webapps
LICENSE          RELEASE-NOTES  alfresco.log  alfresco.log.2024-07-03  amps_share               lib   shared          work
[root@3baa3b4ca64c tomcat]# cd webapps
[root@3baa3b4ca64c webapps]# ls
share
[root@3baa3b4ca64c webapps]# ls
share
[root@3baa3b4ca64c webapps]# cd share
[root@3baa3b4ca64c share]# ls
META-INF  cmm         css     error500.jsp  index.jsp  js       site-index.jsp  themes
WEB-INF   components  ctools  favicon.ico   jquery     modules  templates       yui
[root@3baa3b4ca64c share]# cd META-INF
[root@3baa3b4ca64c META-INF]# ls
MANIFEST.MF  maven
[root@3baa3b4ca64c META-INF]# cd WEB-INF
bash: cd: WEB-INF: No such file or directory
[root@3baa3b4ca64c META-INF]# cd ..
[root@3baa3b4ca64c share]# cd WEB-INF
[root@3baa3b4ca64c WEB-INF]# ls
classes  lib  surf.xml  urlrewrite.xml  web.xml
[root@3baa3b4ca64c WEB-INF]# cd classes
[root@3baa3b4ca64c classes]# ls
THIRD-PARTY.txt  hotswap-agent.properties  org                      surf-config.xml
alfresco         log4j2.properties         share-config.properties  web-application-config.xml
[root@3baa3b4ca64c classes]# cd alfresco
[root@3baa3b4ca64c alfresco]# ls
enterprise-config.xml  share-cmm-config.xml              share-security-config.xml          slingshot-documentlibrary-context.xml
form-config.xml        share-config.xml                  share-workflow-form-config.xml     team-config.xml
messages               share-datalist-form-config.xml    site-data                          templates
module                 share-documentlibrary-config.xml  site-webscripts                    web-extension
```

Configuration Directories

- /opt/alfresco52GA/solr4/**workspace-SpacesStore**/conf/**solrcore.properties** = Live content index config
- /opt/alfresco52GA/solr4/**archive-SpacesStore**/conf/**solrcore.properties** = Archived/Soft deleted content index config

- *These are the main config files for search, there is no hierarchy of config files*

Configuration Directories

Administrator: Command Prompt - docker exec -it cf7080207f19 bash

```
C:\Windows\System32>docker exec -it cf7080207f19 bash
bash-4.2$ ls
contentstore  data  keystores  logs  README.MD  solr  solrhome  solr.in.cmd  solr.in.sh
bash-4.2$ cd solr
bash-4.2$ ls
bin  CHANGES.txt  docs  licenses  LICENSE.txt  LUCENE_CHANGES.txt  NOTICE.txt  README.txt  server
bash-4.2$ cd ..
bash-4.2$ ls
contentstore  data  keystores  logs  README.MD  solr  solrhome  solr.in.cmd  solr.in.sh
bash-4.2$ cd solrhome
bash-4.2$ ls
alfresco  archive  conf  security.json  solr.xml  templates
bash-4.2$ cd archive
bash-4.2$ ls
conf  core.properties
bash-4.2$ cd conf
bash-4.2$ ls
admin-extra.html              admin-extra.menu-top.html  generated_copy_fields.xml  protwords.txt  solrconfig.xml       spellings.txt
admin-extra.menu-bottom.html  elevate.xml                lang                       schema.xml     solrcore.properties  synonyms.txt
bash-4.2$ cat solrcore.properties
#Sun Jun 30 14:51:37 UTC 2024
solr.authorityCache.initialSize=64
alfresco.encryption.ssl.truststore.location=ssl.repo.client.truststore
solr.suggester.minSecsBetweenBuilds=3600
solr.filterCache.size=256
alfresco.batch.count=5000
solr.initial.transaction.range=0-2000
alfresco.cascadeNodeBatchSize=10
alfresco.contentReadBatchSize=100
alfresco.corePoolSize=8
alfresco.metadata.getPathsInNodeBatches=true
data.dir.root=/opt/alfresco-search-services/data
```

Alfresco Platform Configuration – Authentication sync with LDAP server

@c22b47626eee:/usr/local/tomcat/webapps/alfresco/WEB-INF/lib

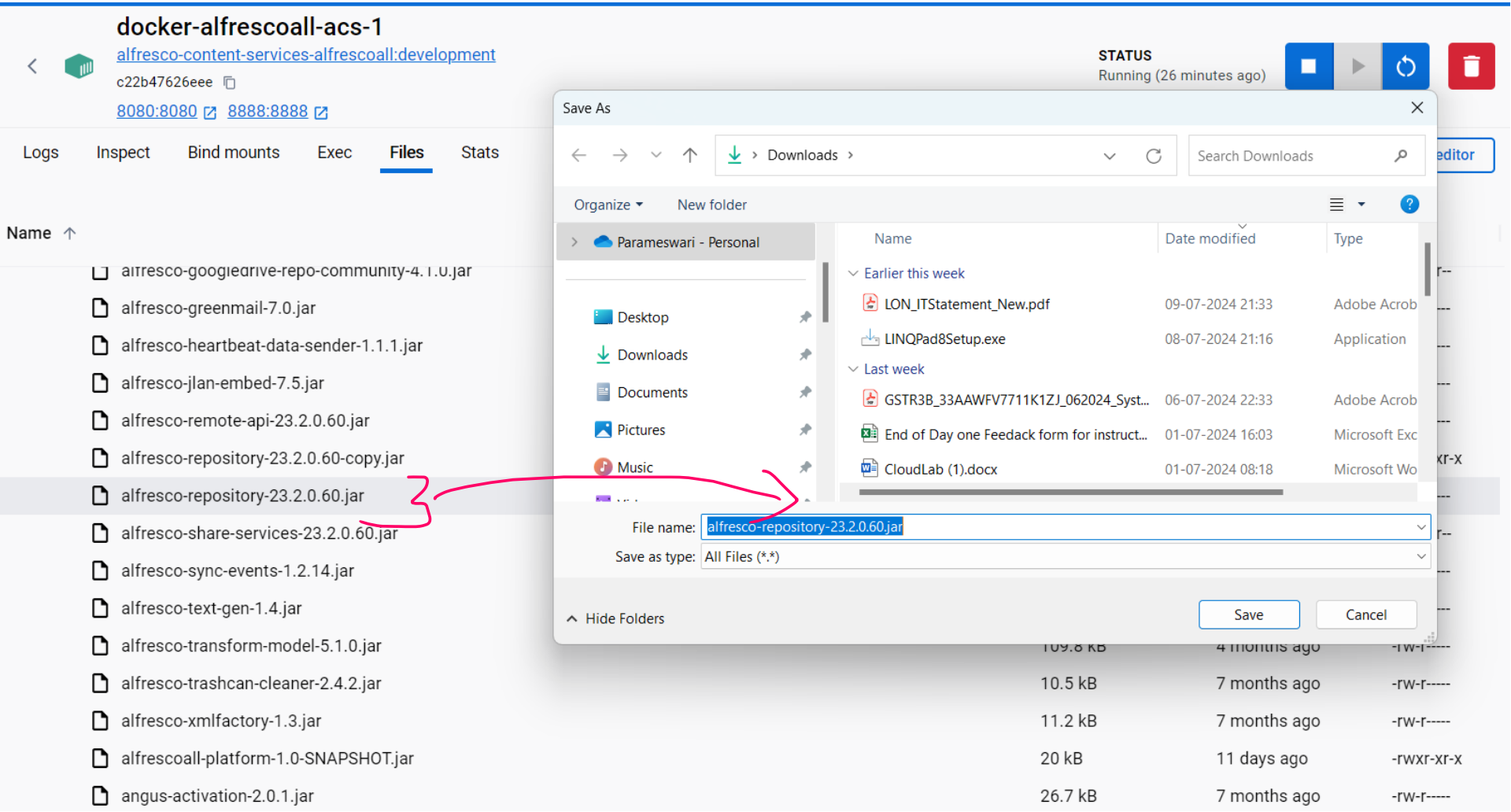
```
[alfresco@c22b47626eee WEB-INF]$ ls
classes  cmis10  cmis11  lib  licenses  web-application-context.xml  web.xml
[alfresco@c22b47626eee WEB-INF]$ cd lib
[alfresco@c22b47626eee lib]$ ls
Dockerfile                                          htmlparser-2.1.jar
FastInfoset-2.1.0.jar                               httpclient-4.5.14.jar
HikariCP-java7-2.4.13.jar                           httpclient5-5.2.1.jar
ST4-4.3.1.jar                                       httpcore-4.4.16.jar
SparseBitSet-1.3.jar                                httpcore5-5.2.3.jar
accessors-smart-2.5.0.jar                           httpcore5-h2-5.2.jar
acegi-security-0.8.2_patched.jar                    httpmime-4.5.14.jar
acs-event-model-0.0.27.jar                          icu4j-73.2.jar
activemq-client-jakarta-5.18.3.jar                  istack-commons-runtime-4.1.2.jar
activiti-bpmn-converter-5.23.0.jar                  j2objc-annotations-2.8.jar
activiti-bpmn-model-5.23.0.jar                      jackcess-4.0.5.jar
activiti-engine-5.23.0.jar                          jackcess-encrypt-4.0.2.jar
activiti-image-generator-5.23.0.jar                 jackson-annotations-2.15.2.jar
activiti-process-validation-5.23.0.jar              jackson-core-2.15.2.jar
activiti-spring-5.23.0.jar                          jackson-databind-2.15.2.jar
alfresco-aos-module-3.0.0.jar                       jackson-dataformat-yaml-2.15.2.jar
alfresco-aos-repo-binding-3.0.0.jar                 jai-imageio-core-1.4.0.jar
alfresco-core-23.2.0.60.jar                         jakarta.activation-2.0.1.jar
alfresco-data-model-23.2.0.60.jar                   jakarta.annotation-api-2.1.1.jar
alfresco-googledocs-community-config-1.0.1.jar      jakarta.jms-api-3.0.0.jar
alfresco-googledrive-repo-base-4.1.0.jar            jakarta.jws-api-3.0.0.jar
alfresco-googledrive-repo-community-4.1.0.jar       jakarta.mail-2.0.1.jar
alfresco-greenmail-7.0.jar                          jakarta.transaction-api-2.0.1.jar
alfresco-heartbeat-data-sender-1.1.1.jar            jakarta.xml.bind-api-4.0.0.jar
alfresco-jlan-embed-7.5.jar                         jakarta.xml.soap-api-2.0.1.jar
alfresco-remote-api-23.2.0.60.jar                   jakarta.xml.ws-api-3.0.1.jar
alfresco-repository-23.2.0.60.jar                   java-libpst-0.9.3.jar
alfresco-share-services-23.2.0.60.jar               jaxb-core-4.0.3.jar
alfresco-sync-events-1.2.14.jar                     jaxb-runtime-4.0.3.jar
```

29°C Mostly sunny Search ENG IN 07:51 12-07-2024

Alfresco Platform Configuration – Authentication sync with LDAP server

```
@c22b47626eee:/usr/local/tomcat/webapps/alfresco/WEB-INF/lib
activiti-spring-5.23.0.jar
alfresco-aos-module-3.0.0.jar
alfresco-aos-repo-binding-3.0.0.jar
alfresco-core-23.2.0.60.jar
alfresco-data-model-23.2.0.60.jar
alfresco-googledocs-community-config-1.0.1.jar
alfresco-googledrive-repo-base-4.1.0.jar
alfresco-googledrive-repo-community-4.1.0.jar
alfresco-greenmail-7.0.jar
alfresco-heartbeat-data-sender-1.1.1.jar
alfresco-jlan-embed-7.5.jar
alfresco-remote-api-23.2.0.60.jar
alfresco-repository-23.2.0.60.jar
alfresco-share-services-23.2.0.60.jar
alfresco-sync-events-1.2.14.jar
alfresco-text-gen-1.4.jar
alfresco-transform-model-5.1.0.jar
alfresco-trashcan-cleaner-2.4.2.jar
alfresco-xmlfactory-1.3.jar
alfrescoall-platform-1.0-SNAPSHOT.jar
angus-activation-2.0.1.jar
antlr-3.5.3.jar
antlr-runtime-3.5.3.jar
aoservices-4.0.11-A1.jar
apache-mime4j-core-0.8.9.jar
apache-mime4j-dom-0.8.9.jar
asm-9.5.jar
aspectjrt-1.9.20.1.jar
avalon-framework-api-4.3.1.jar
avalon-framework-impl-4.3.1.jar
bcmail-jdk18on-1.77.jar
bcpkix-jdk18on-1.77.jar
bcprov-jdk18on-1.77.jar
jackson-databind-2.15.2.jar
jackson-dataformat-yaml-2.15.2.jar
jai-imageio-core-1.4.0.jar
jakarta.activation-2.0.1.jar
jakarta.annotation-api-2.1.1.jar
jakarta.jms-api-3.0.0.jar
jakarta.jws-api-3.0.0.jar
jakarta.mail-2.0.1.jar
jakarta.transaction-api-2.0.1.jar
jakarta.xml.bind-api-4.0.0.jar
jakarta.xml.soap-api-2.0.1.jar
jakarta.xml.ws-api-3.0.1.jar
java-libpst-0.9.3.jar
jaxb-core-4.0.3.jar
jaxb-runtime-4.0.3.jar
jaxen-1.2.0.jar
jcip-annotations-1.0-1.jar
jcl-over-slf4j-2.0.9.jar
jdom2-2.0.6.1.jar
jhighlight-1.1.0.jar
jibx-run-1.4.2.jar
jmatio-1.5.jar
joda-time-2.12.5.jar
json-20231013.jar
json-path-2.9.0.jar
json-simple-1.1.1.jar
json-smart-2.5.0.jar
jsr107cache-1.1.jar
jsr305-3.0.2.jar
jug-2.0.0-asl.jar
juniversalchardet-2.4.0.jar
junrar-7.5.5.jar
jutf7-1.0.0.jar
```

Alfresco Platform Configuration – Authentication sync with LDAP server



Alfresco Platform Configuration – Authentication sync with LDAP server

- Make copy of alfresco repository.jar
- Convert jar extension to .zip
- Extract zip files
- Copy folder to /usr/local/tomcat/webapps/alfresco/WEB-INF/lib

Alfresco Platform Configuration – Authentication sync with LDAP server

folder

@c22b47626eee:/usr/local/tomcat/webapps/alfresco/WEB-INF/lib/alfresco-repository-23.2.0.60-copy

```
C:\Users\Dell\Downloads>docker cp alfresco-repository-23.2.0.60-copy c22b47626eee:/usr/local/tomcat/webapps/alfresco/WEB-INF/lib
Successfully copied 34.1MB to c22b47626eee:/usr/local/tomcat/webapps/alfresco/WEB-INF/lib

C:\Users\Dell\Downloads>docker exec -it c22b47626eee bash
[alfresco@c22b47626eee tomcat]$ cd webapps
[alfresco@c22b47626eee webapps]$ ls
ROOT  _vti_bin  alfresco  api-explorer
[alfresco@c22b47626eee webapps]$ cd alfresco
[alfresco@c22b47626eee alfresco]$ ls
META-INF  WEB-INF  admin  cmis-browser-app  cmisfs  css  error.jsp  favicon.ico  images  index.jsp  scripts
[alfresco@c22b47626eee alfresco]$ cd WEB-INF
[alfresco@c22b47626eee WEB-INF]$ cd lib
[alfresco@c22b47626eee lib]$ ls
Dockerfile                                          htmllexer-2.1.jar
FastInfoset-2.1.0.jar                               htmlparser-2.1.jar
HikariCP-java7-2.4.13.jar                           httpclient-4.5.14.jar
ST4-4.3.1.jar                                       httpclient5-5.2.1.jar
SparseBitSet-1.3.jar                                httpcore-4.4.16.jar
accessors-smart-2.5.0.jar                           httpcore5-5.2.3.jar
acegi-security-0.8.2_patched.jar                    httpcore5-h2-5.2.jar
acs-event-model-0.0.27.jar                          httpmime-4.5.14.jar
activemq-client-jakarta-5.18.3.jar                  icu4j-73.2.jar
activiti-bpmn-converter-5.23.0.jar                  istack-commons-runtime-4.1.2.jar
activiti-bpmn-model-5.23.0.jar                      j2objc-annotations-2.8.jar
activiti-engine-5.23.0.jar                          jackcess-4.0.5.jar
activiti-image-generator-5.23.0.jar                 jackcess-encrypt-4.0.2.jar
activiti-process-validation-5.23.0.jar              jackson-annotations-2.15.2.jar
activiti-spring-5.23.0.jar                          jackson-core-2.15.2.jar
alfresco-aos-module-3.0.0.jar                       jackson-databind-2.15.2.jar
alfresco-aos-repo-binding-3.0.0.jar                 jackson-dataformat-yaml-2.15.2.jar
alfresco-core-23.2.0.60.jar                         jai-imageio-core-1.4.0.jar
alfresco-data-model-23.2.0.60.jar                   jakarta.activation-2.0.1.jar
```

Alfresco Platform Configuration – Authentication sync with LDAP server

```
@c22b47626eee:/usr/local/tomcat/webapps/alfresco/WEB-INF/lib/alfresco-repository-23.2.0.60-copy
alfresco-googledocs-community-config-1.0.1.jar
alfresco-googledrive-repo-base-4.1.0.jar
alfresco-googledrive-repo-community-4.1.0.jar
alfresco-greenmail-7.0.jar
alfresco-heartbeat-data-sender-1.1.1.jar
alfresco-jlan-embed-7.5.jar
alfresco-remote-api-23.2.0.60.jar
alfresco-repository-23.2.0.60-copy
alfresco-repository-23.2.0.60-copy.jar
alfresco-repository-23.2.0.60.jar
alfresco-share-services-23.2.0.60.jar
alfresco-sync-events-1.2.14.jar
alfresco-text-gen-1.4.jar
alfresco-transform-model-5.1.0.jar
alfresco-trashcan-cleaner-2.4.2.jar
alfresco-xmlfactory-1.3.jar
alfrescoall-platform-1.0-SNAPSHOT.jar
angus-activation-2.0.1.jar
antlr-3.5.3.jar
antlr-runtime-3.5.3.jar
aoservices-4.0.11-A1.jar
apache-mime4j-core-0.8.9.jar
apache-mime4j-dom-0.8.9.jar
asm-9.5.jar
aspectjrt-1.9.20.1.jar
avalon-framework-api-4.3.1.jar
avalon-framework-impl-4.3.1.jar
bcmail-jdk18on-1.77.jar
bcpkix-jdk18on-1.77.jar
bcprov-jdk18on-1.77.jar
bcutil-jdk18on-1.77.jar
bliki-3.0.2.jar
bsf-2.4.0.jar
jakarta.annotation-api-2.1.1.jar
jakarta.jms-api-3.0.0.jar
jakarta.jws-api-3.0.0.jar
jakarta.mail-2.0.1.jar
jakarta.transaction-api-2.0.1.jar
jakarta.xml.bind-api-4.0.0.jar
jakarta.xml.soap-api-2.0.1.jar
jakarta.xml.ws-api-3.0.1.jar
java-libpst-0.9.3.jar
jaxb-core-4.0.3.jar
jaxb-runtime-4.0.3.jar
jaxen-1.2.0.jar
jcip-annotations-1.0-1.jar
jcl-over-slf4j-2.0.9.jar
jdom2-2.0.6.1.jar
jhighlight-1.1.0.jar
jibx-run-1.4.2.jar
jmatio-1.5.jar
joda-time-2.12.5.jar
json-20231013.jar
json-path-2.9.0.jar
json-simple-1.1.1.jar
json-smart-2.5.0.jar
jsr107cache-1.1.jar
jsr305-3.0.2.jar
jug-2.0.0-asl.jar
juniversalchardet-2.4.0.jar
junrar-7.5.5.jar
jutf7-1.0.0.jar
jwarc-0.28.3.jar
lang-tag-1.7.jar
listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar
log4j-api-2.20.0.jar
```

Alfresco Platform Configuration – Authentication sync with LDAP server

@c22b47626eee:/usr/local/tomcat/webapps/alfresco/WEB-INF/lib/alfresco-repository-23.2.0.60-copy

```
google-api-services-drive-v3-rev20220709-1.32.1.jar     vorbis-java-tika-0.8.jar
google-api-services-oauth2-v2-rev20200213-1.32.1.jar    woodstox-core-6.5.1.jar
google-http-client-1.42.2.jar                           wsdl4j-1.6.3.jar
google-http-client-apache-v2-1.39.2.jar                 xalan-2.7.2-alfresco.jar
google-http-client-gson-1.39.2.jar                      xercesImpl-2.12.2.jar
google-http-client-jackson2-1.42.2.jar                  xml-resolver-1.2.jar
google-oauth-client-1.33.3.jar                          xmlbeans-5.2.0.jar
grpc-context-1.27.2.jar                                 xmlschema-core-2.3.0.jar
gson-2.10.1.jar                                         xmlsec-3.0.3.jar
guava-32.1.2-jre.jar                                    xmpbox-2.0.29.jar
guava-mini-0.1.6.jar                                    xmpcore-6.1.11.jar
guessencoding-1.4.jar                                   xpp3-1.1.3.4.O.jar
hawtbuf-1.11.jar                                        xz-1.9.jar
[alfresco@c22b47626eee lib]$
[alfresco@c22b47626eee lib]$ cd alfresco-repository-23.2.0.60-copy
[alfresco@c22b47626eee alfresco-repository-23.2.0.60-copy]$ ls
META-INF  THIRD-PARTY.txt  alfresco  alfresco-global.properties.sample  builddef.lst  lib3party  org  queryRegister.dtd
[alfresco@c22b47626eee alfresco-repository-23.2.0.60-copy]$
```

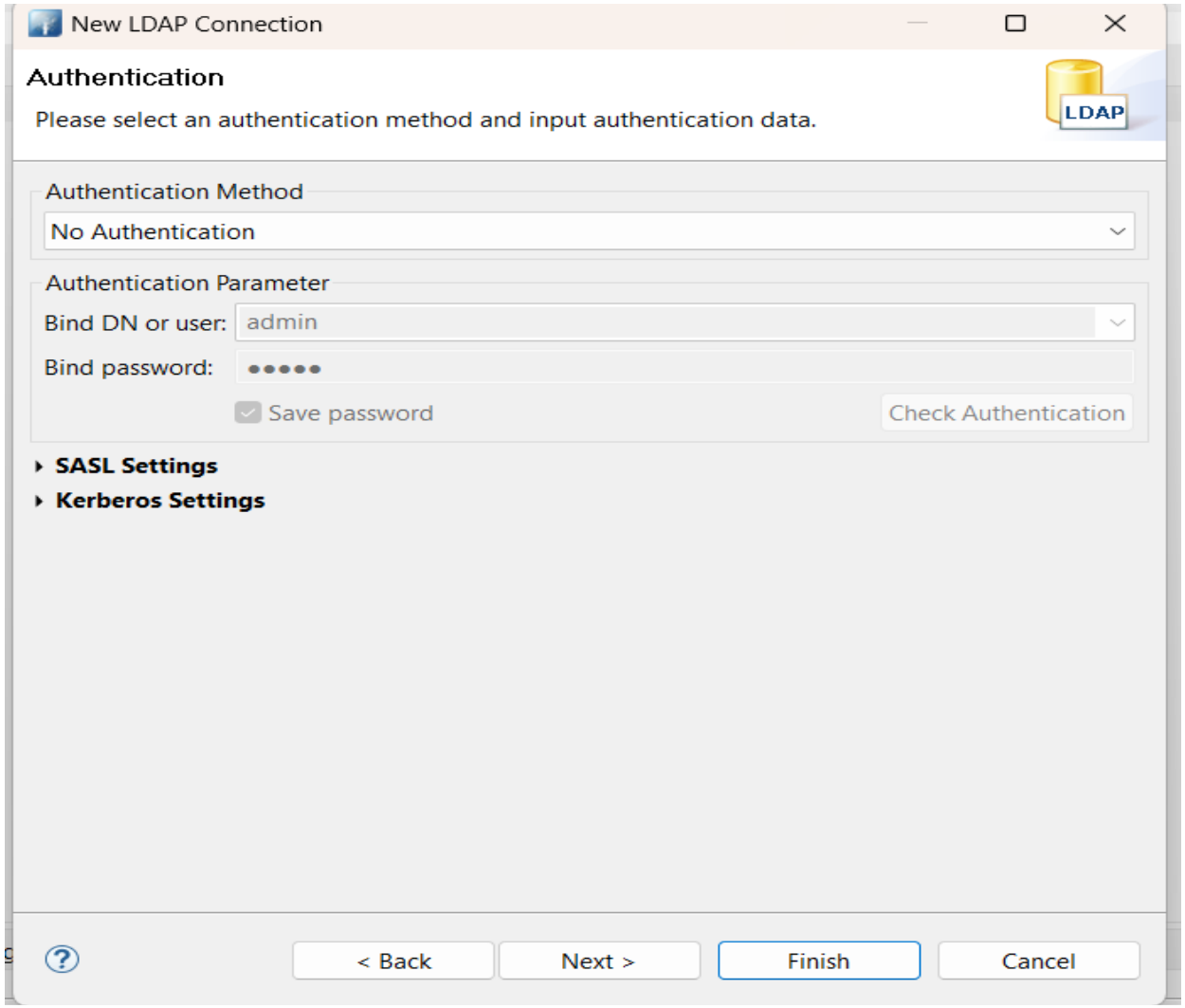
Alfresco Platform Configuration – Authentication sync with LDAP server

```
@c22b47626eee:/usr/local/tomcat/webapps/alfresco/WEB-INF/lib/alfresco-repository-23.2.0.60-copy/alfresco/subsystems/Authentication/ldap
calendar-services-context.xml                 module                                          smartfolder-context.xml
client                                        module-context.xml                              social-publishing-context.xml
comment-services-context.xml                  mt                                              solr-facets-context.xml
content-services-context.xml                  network-authentication-context.xml              spring
copy-services-context.xml                     node-locator-context.xml                        st-context.xml
core-services-context.xml                     node-services-context.xml                       subsystems
custom-model-management-services-context.xml  notification-services-context.xml              tagging-services-context.xml
dao                                           oauth1-credentials-store-services-context.xml   template-services-context.xml
dbscripts                                     oauth2-credentials-store-services-context.xml   templates
deprecated-context.xml                        opencmis-context.xml                            thumbnail
deprecated-model-bootstrap-context.xml        opencmis-qnamefilter-context.xml                thumbnail-service-context.xml
desktop                                       ownable-services-context.xml                    tika
discussions-services-context.xml              patch                                           transfer-service-context.xml
distributionpolicies-context.xml              period-type-context.xml                         transforms
domain                                        policy-context.xml                              tx-cache-context.xml
download-services-context.xml                 preference-service-context.xml                  usage-services-context.xml
encryption-context.xml                        public-services-context.xml                     webdav-context.xml
events-context.xml                            public-services-security-context.xml            workflow
events2-context.xml                           quickshare-services-context.xml                 workflow-context.xml
[alfresco@c22b47626eee alfresco]$ cd subsystems
[alfresco@c22b47626eee subsystems]$ ls
ActivitiesFeed  ContentStore  Messaging    Search         Synchronization  email        imap
Authentication  Events        Replication  Subscriptions  Transformers     fileServers  sysAdmin
[alfresco@c22b47626eee subsystems]$ cd Authentication
[alfresco@c22b47626eee Authentication]$ ls
alfrescoNtlm  common-ldap-context.xml  external  identity-service  kerberos  ldap  ldap-ad
[alfresco@c22b47626eee Authentication]$ cd ldap
[alfresco@c22b47626eee ldap]$ pwd
/usr/local/tomcat/webapps/alfresco/WEB-INF/lib/alfresco-repository-23.2.0.60-copy/alfresco/subsystems/Authentication/ldap
[alfresco@c22b47626eee ldap]$
```

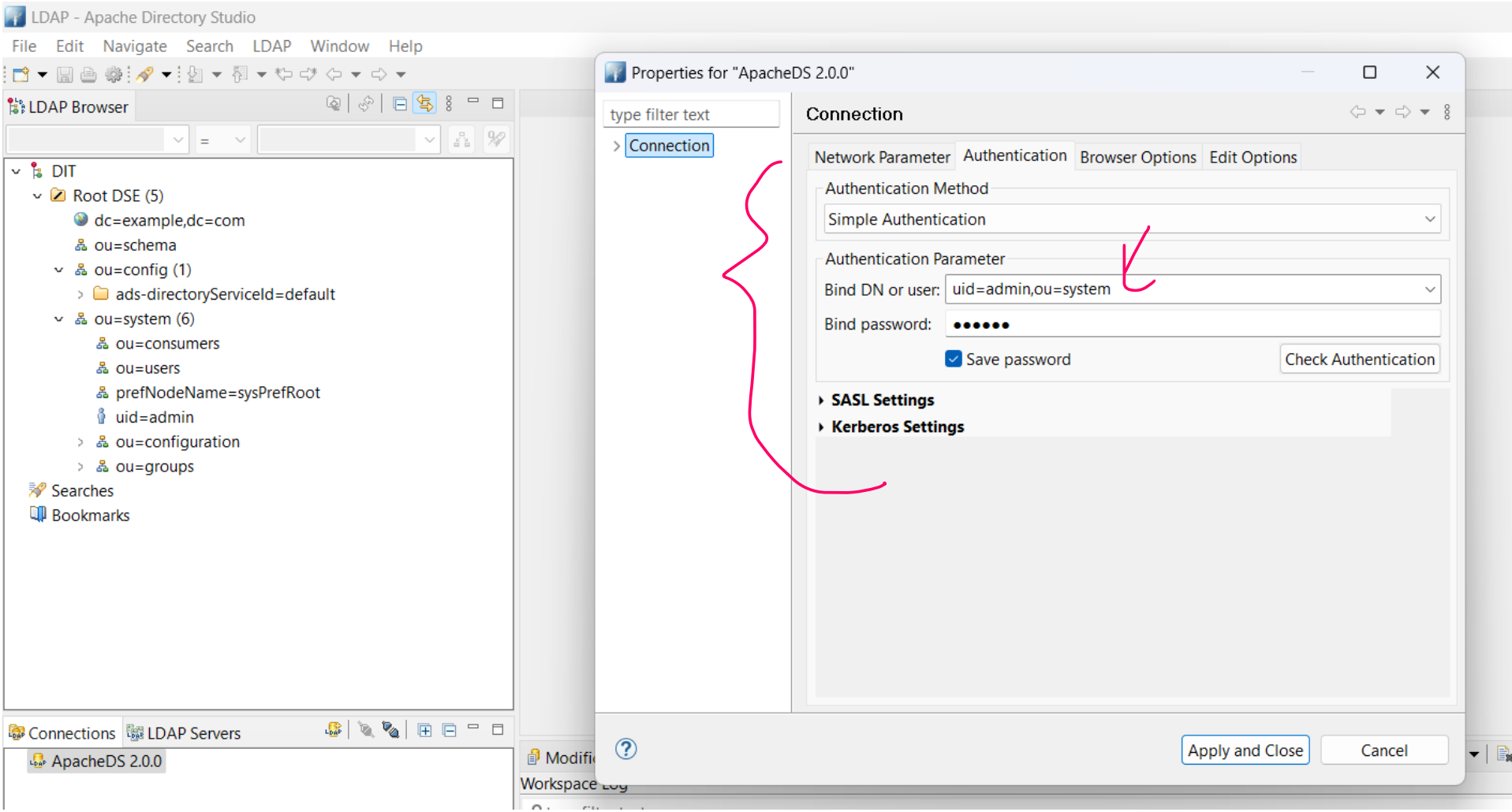
Apache Directory Studio

- Initialize and start gerrit.
- docker run -d -p 389:10389 openfrontier/apacheds
- Open your Apache Directory Studio and create a new ldap connection to ldap://:389 with
- **uid=admin,ou=system as the BindDN**
- **secret as the bind BindPassword.**
- Use a docker named volume as the storage.
- DO NOT mount host volumes in particular directories under the home directory like ~/apacheds as a volume!!! Use named volume instead!!!
- Create a named volume for the ApacheDS.
- docker volume create apacheds_data
- Initialize and start ApacheDS using the volume created above.
- **docker run -d -p 389:10389 -v apacheds_data:/var/lib/apacheds openfrontier/apacheds**

Apache Directory Studio



Apache Directory Studio



Apache Directory Studio

Restart LDAP server after creating the partition

LDAP - C:Users\Dell\.ApacheDirectoryStudio\.metadata\.plugins\org.apache.directory.studio.ldapservers\servers\2759e939-ddcb-41d1-80ab-c767581f82b9\conf\ou=config.ldif - Apache Directory Studio

File Edit Navigate Search LDAP Window Help

LDAP Browser

- DIT
  - Root DSE (5)
    - dc=example,dc=com
    - ou=schema
    - ou=config (1)
      - ads-directoryServiceId=default
    - ou=system (6)
      - ou=consumers
      - ou=users
      - prefNodeName=sysPrefRoot
      - uid=admin
      - ou=configuration
      - ou=groups
- Searches
- Bookmarks

No entry selected | ou=config.ldif

LDAP/LDAPS Servers

LDAP/LDAPS Servers

- Enable LDAP Server
  - Port: 10389 (Default: 10389)
  - Address: 0.0.0.0 (Default: 0.0.0.0)
  - NbThreads: 8 (Default: 4)
  - BackLog Size: 50 (Default: 50)
- Enable LDAPS Server
  - Port: 10636 (Default: 10636)
  - Address: 0.0.0.0 (Default: 0.0.0.0)
  - NbThreads: 3 (Default: 4)
  - BackLog Size: 50 (Default: 50)

Limits

SSL/Start TLS Keystore

SSL Advanced Settings

Supported Authentication Mechanisms

- Simple
- GSSAPI
- CRAM-MD5
- DIGEST-MD5
- NTLM
  - Provider: com.foo.Bar
- GSS-SPNEGO
  - Provider: com.foo.Bar

SASL Settings

Advanced

Apache Directory Studio

Initially LDIF details won't be there

File Edit Navigate Search LDAP Window Help

LDAP Browser

- DIT
  - Root DSE (7)
    - dc=example,dc=com
    - dc=vheb,dc=com
    - ou=schema
    - o=sevenseas (2)
      - ou=groups (2)
        - ou=ranks
        - ou=crews (3)
          - cn=HMS Bounty
          - cn=HMS Lydia
          - cn=HMS Victory
      - ou=people (11)
        - cn=Cornelius Buckley
        - cn=Fletcher Christian
        - cn=Horatio Hornblower
        - cn=Horatio Nelson
        - cn=John Fryer
        - cn=John Hallett
        - cn=Moultrie Crystal
        - cn=Thomas Masterman Hardy
        - cn=Thomas Quist

ou=ranks,ou=groups,o=sevenseas | ou=config.ldif

Partitions

All Partitions

- example (dc=example,dc=com) [JDBM]
- sevenses (o=sevenseas) [JDBM]
- system (ou=system) [JDBM]
- vheb (dc=vheb,dc=com) [JDBM]

Add | Delete

Partition General Details

Set the properties of the partition.

Partition Type: JDBM

ID: sevenses

Suffix: o=sevenseas

☑ Synchronization On Write

Context Entry

Set the attribute/value pairs for the Context Entry of the partition.

☑ Auto-generate context entry from suffix DN

Attribute	Value
objectclass	domain
objectclass	top

Add... | Edit... | Delete

Partition Specific Settings

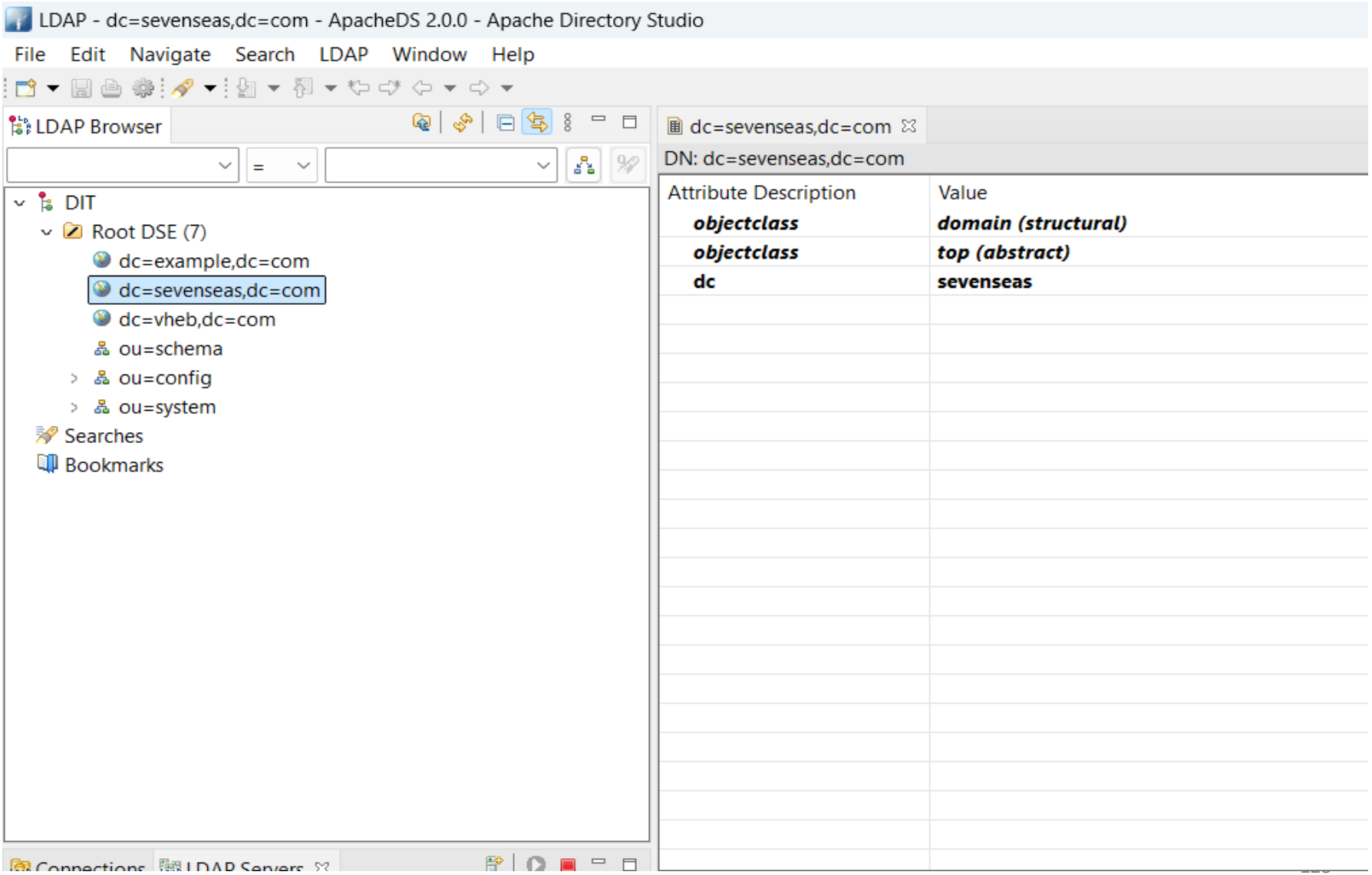
Cache Size: 100

☑ Enable Optimizer

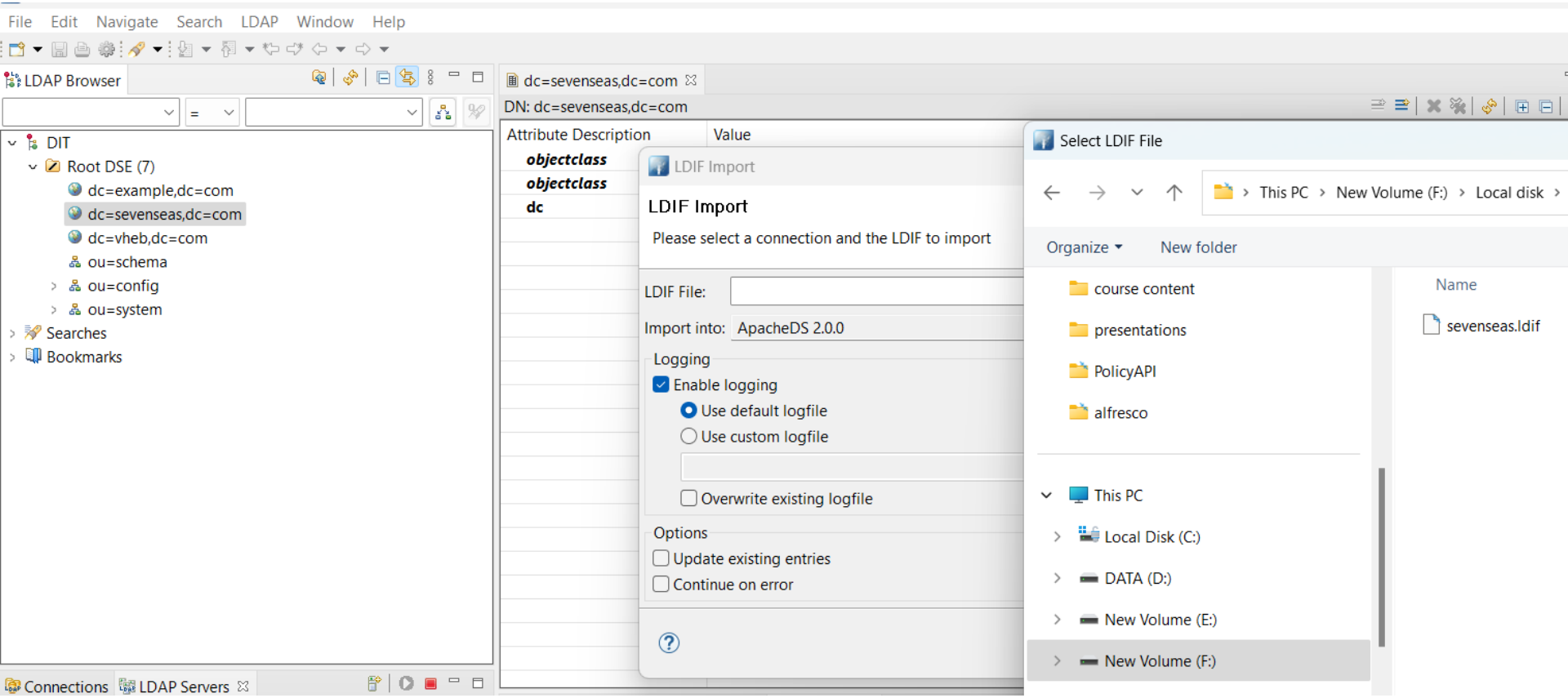
Indexed Attributes

Set the indexed attributes of the partition.

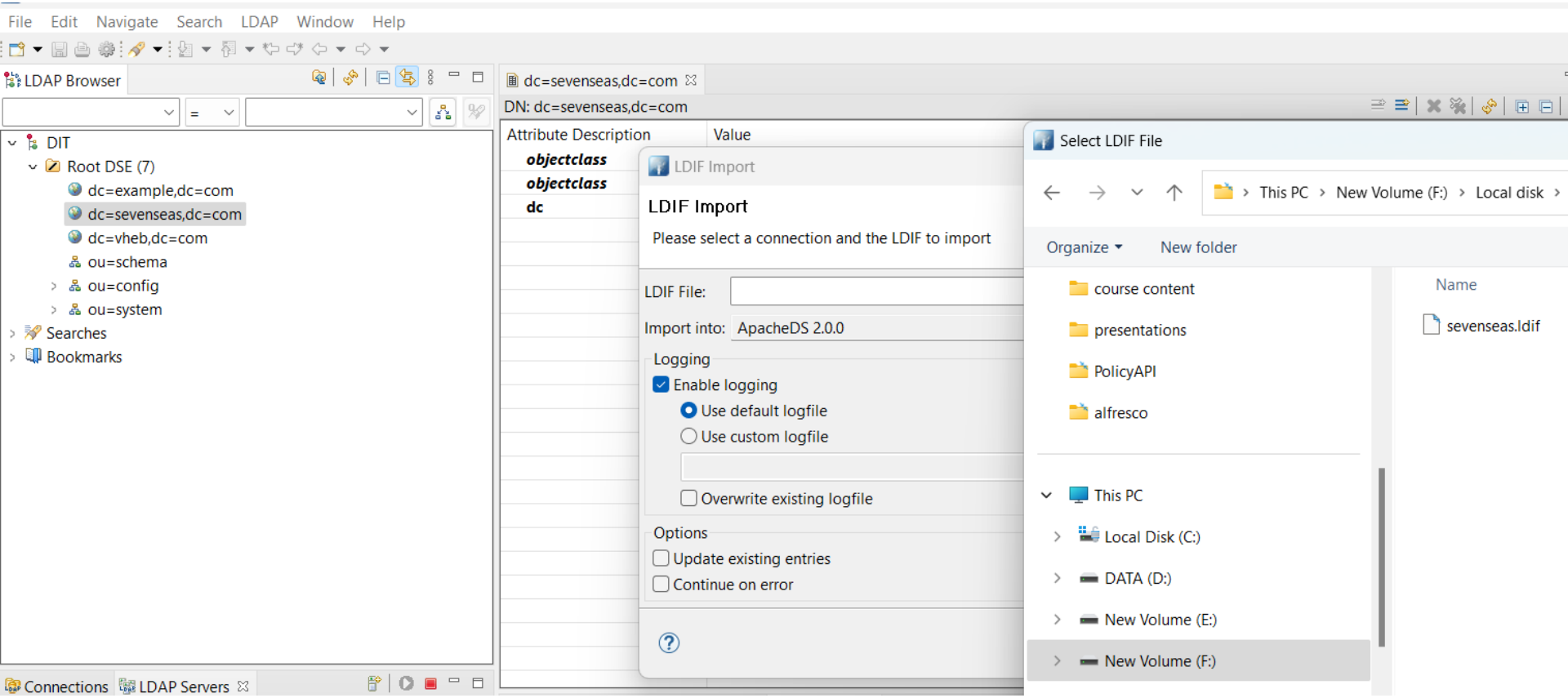
Apache Directory Studio



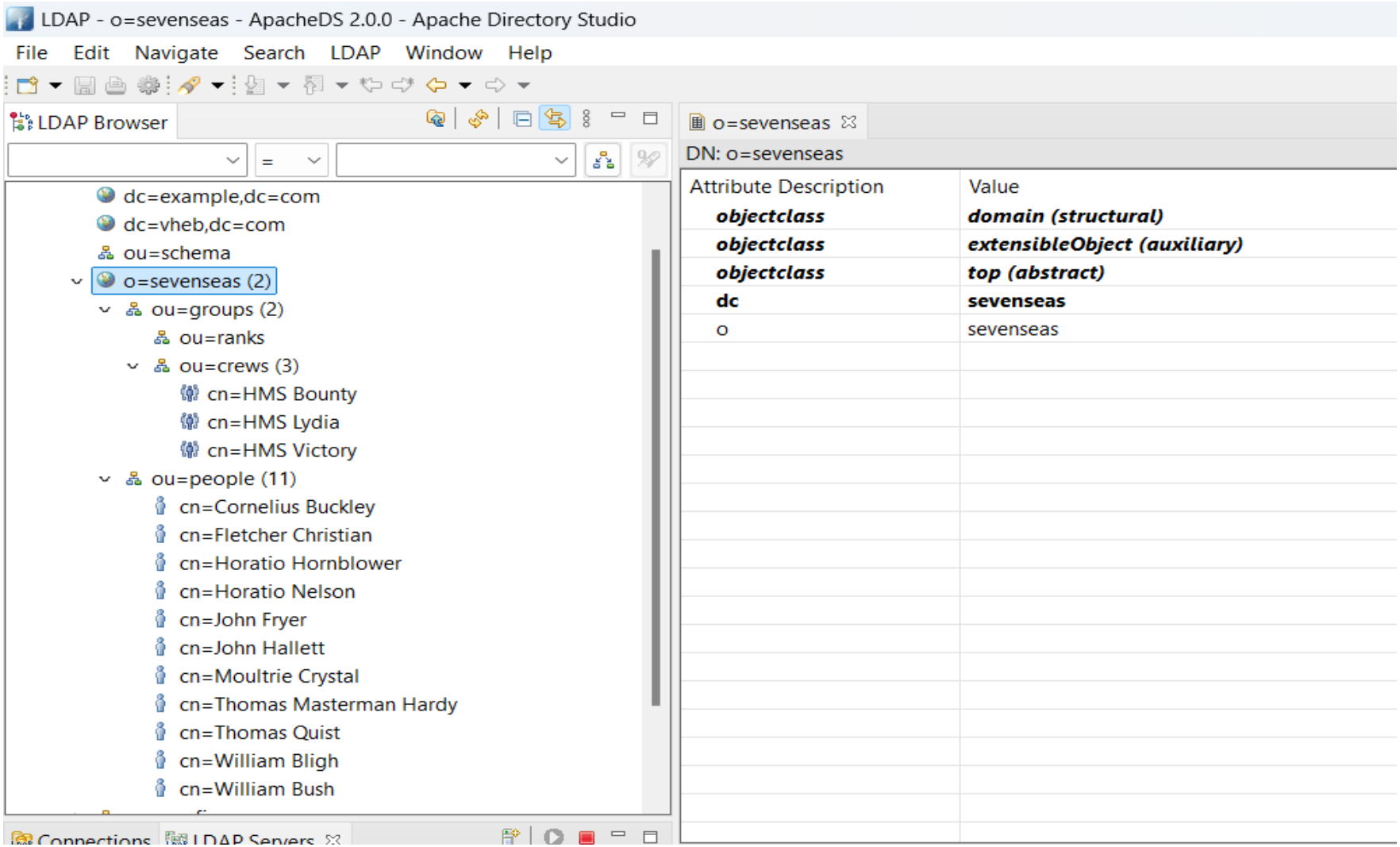
Apache Directory Studio



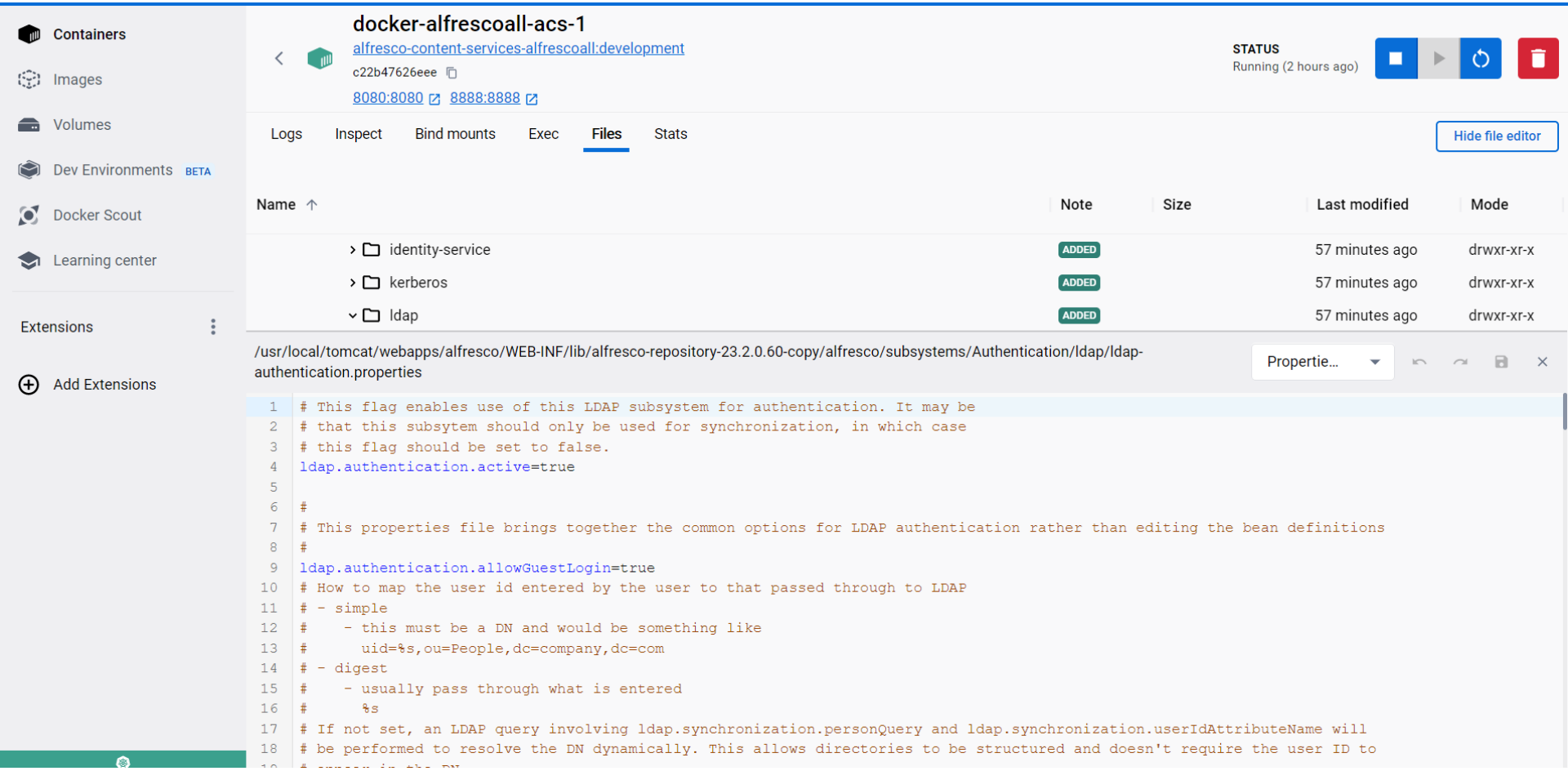
Apache Directory Studio



Apache Directory Studio After LDIF Import



Apache Directory Studio Edit LDAP properties

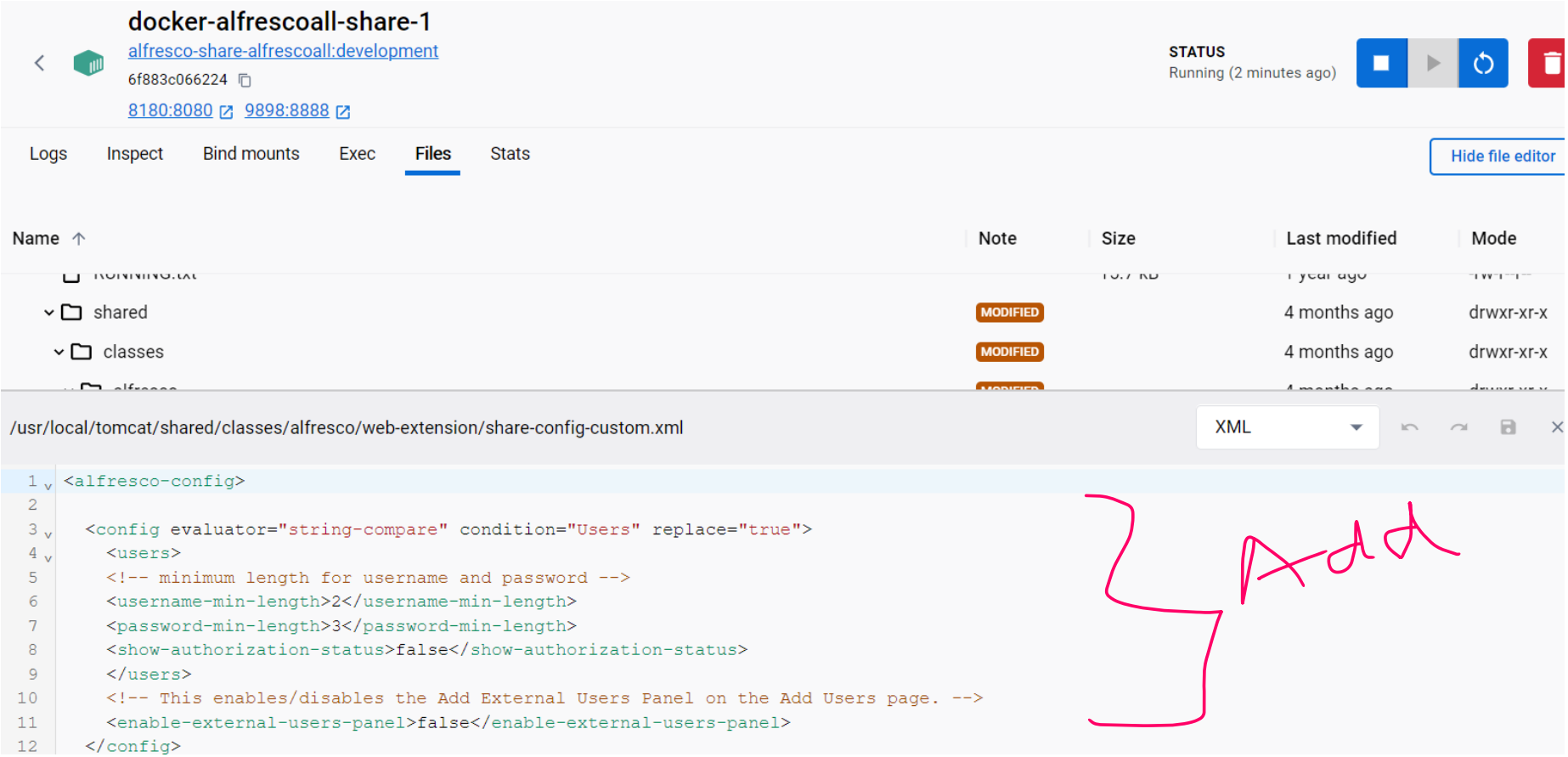


Apache Directory Studio Edit LDAP properties

/usr/local/tomcat/shared/classes/alfresco-global.properties

```
## Authentication with local Directory server
ldap.authentication.active=true
authentication.chain=localLDAP:ldap,alfrescoNtlm1:alfrescoNtlm
ldap.authentication.java.naming.provider.url=ldap://host.docker.internal:389
ldap.authentication.userNameFormat=cn=%s,ou=people,o=sevenseas

## Synchronization of user and group info from directory server
ldap.synchronization.active=true
ldap.synchronization.java.naming.security.principal=uid\=admin,ou\=system
ldap.synchronization.java.naming.security.credentials=secret
ldap.synchronization.groupQuery=(objectclass\=groupOfUniqueNames)
ldap.synchronization.personQuery=(objectclass\=person)
ldap.synchronization.groupSearchBase=ou\=groups,o\=sevenseas
ldap.synchronization.userSearchBase=ou\=people,o\=sevenseas
ldap.synchronization.groupMemberAttributeName=uniquemember
```

Apache Directory Studio Edit share-config-custom

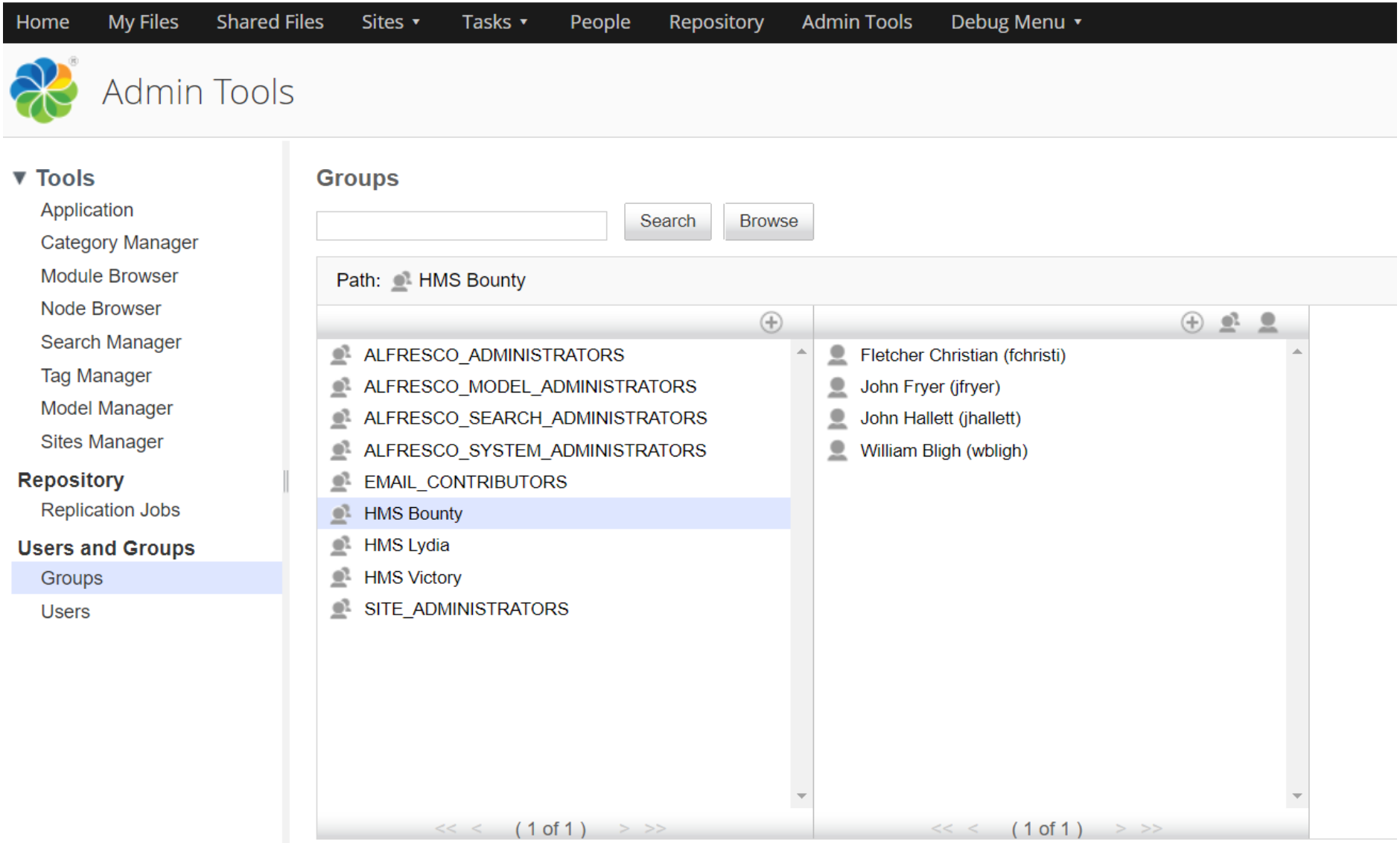


https://docs.alfresco.com/content-services/7.2/develop/share-ext-points/share-config/

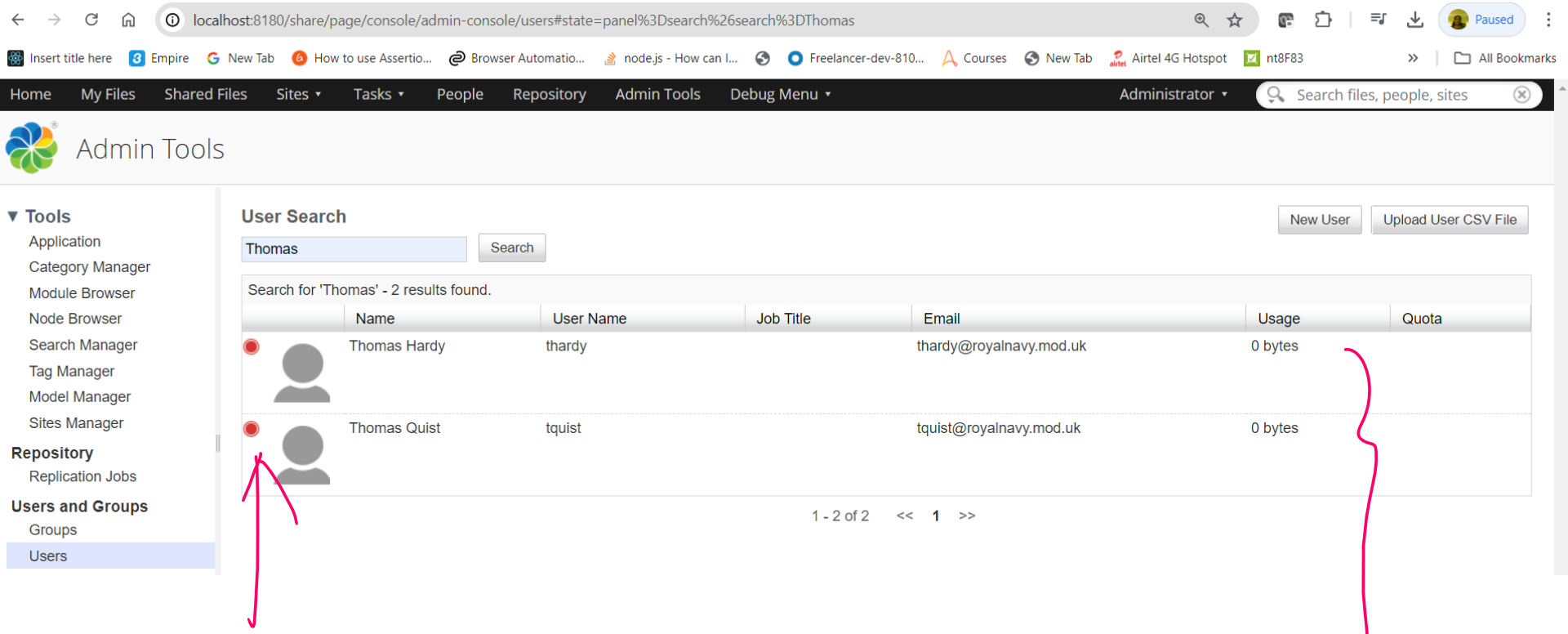
Apache Directory Studio Edit share-config-custom

- <config evaluator="string-compare" condition="Users" replace="true">
- <users>
- <!-- minimum length for username and password -->
- <username-min-length>2</username-min-length>
- <password-min-length>3</password-min-length>
- <show-authorization-status>false</show-authorization-status>
- </users>
- <enable-external-users-panel>true</enable-external-users-panel>
- </config>

Apache Directory Studio Edit LDAP properties

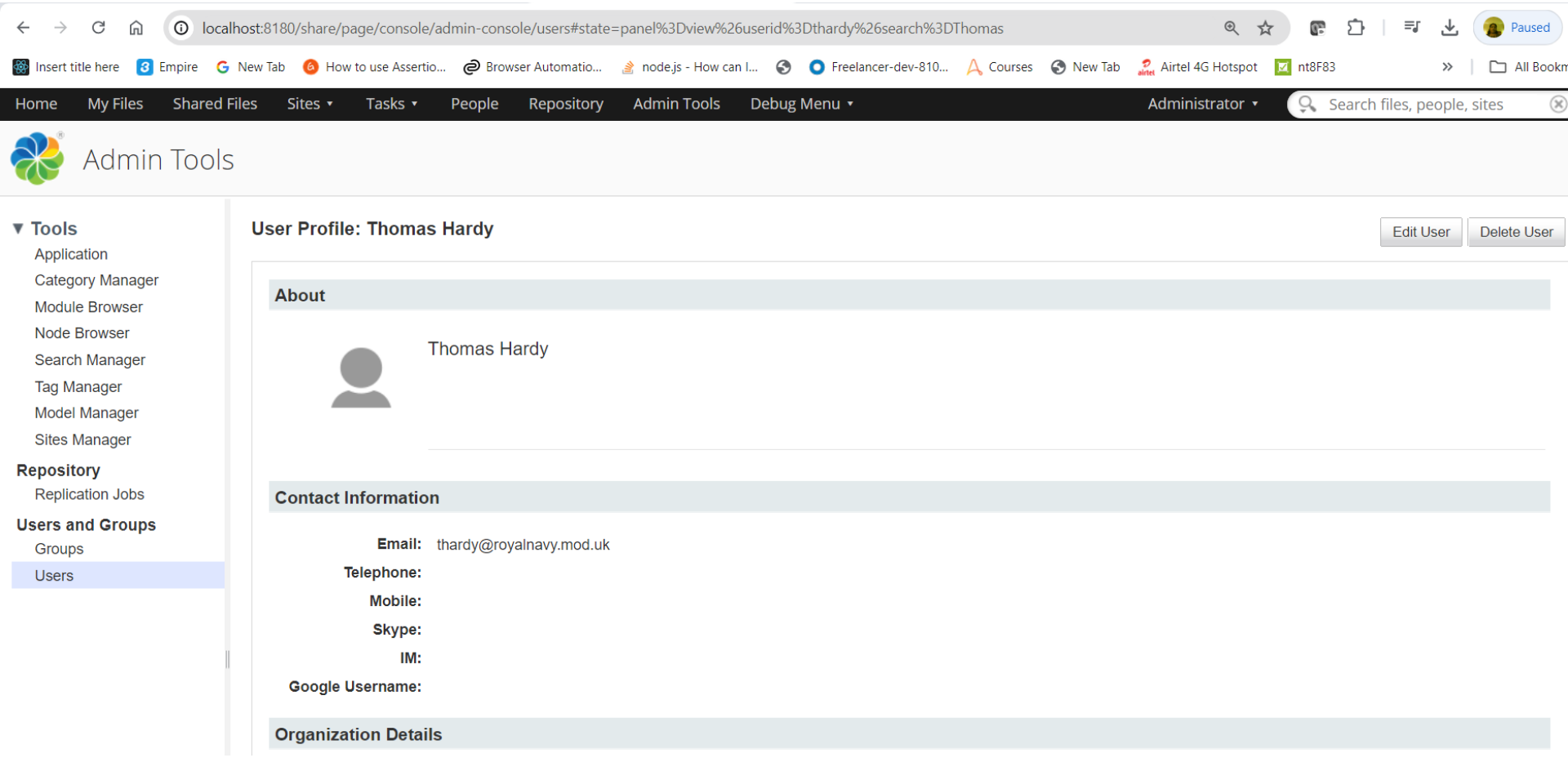


Apache Directory Studio Edit LDAP properties



Disable status. Need to Enable

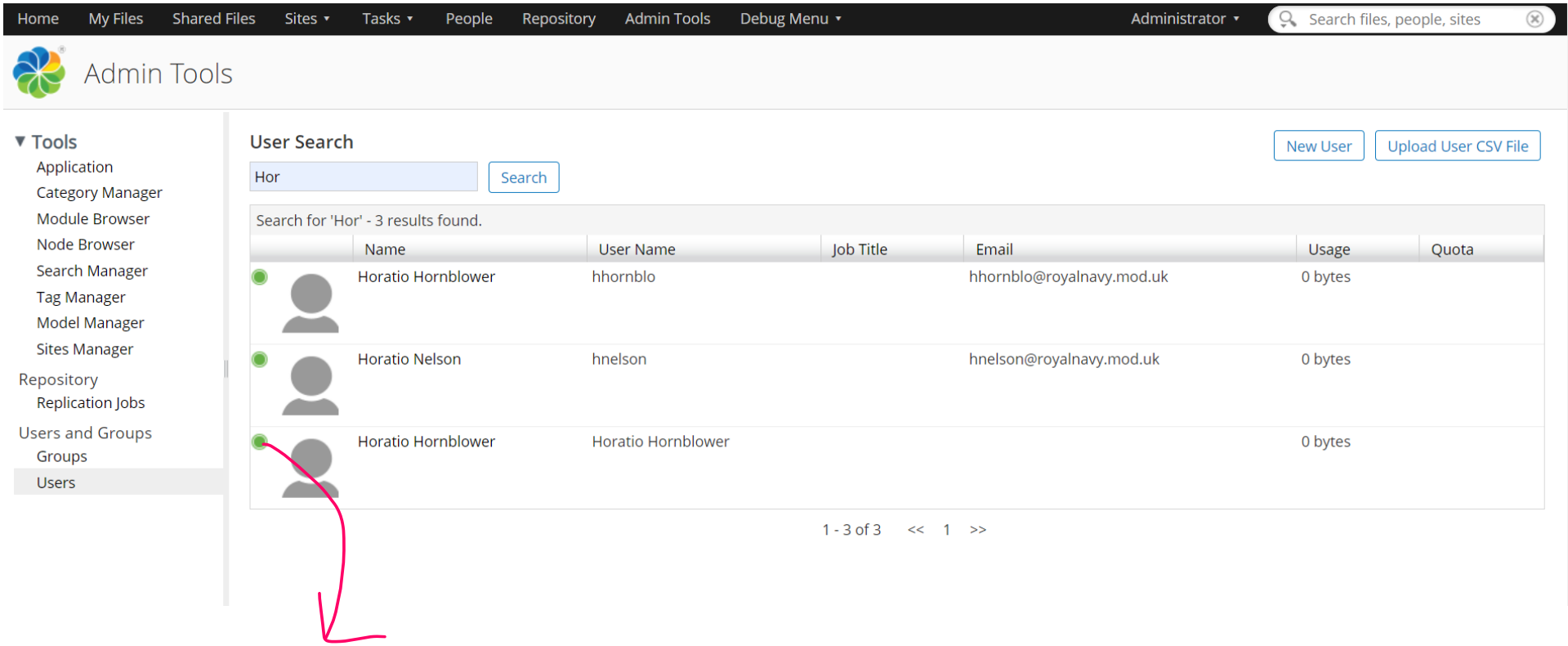
Apache Directory Studio Edit LDAP properties



Enable Account Status

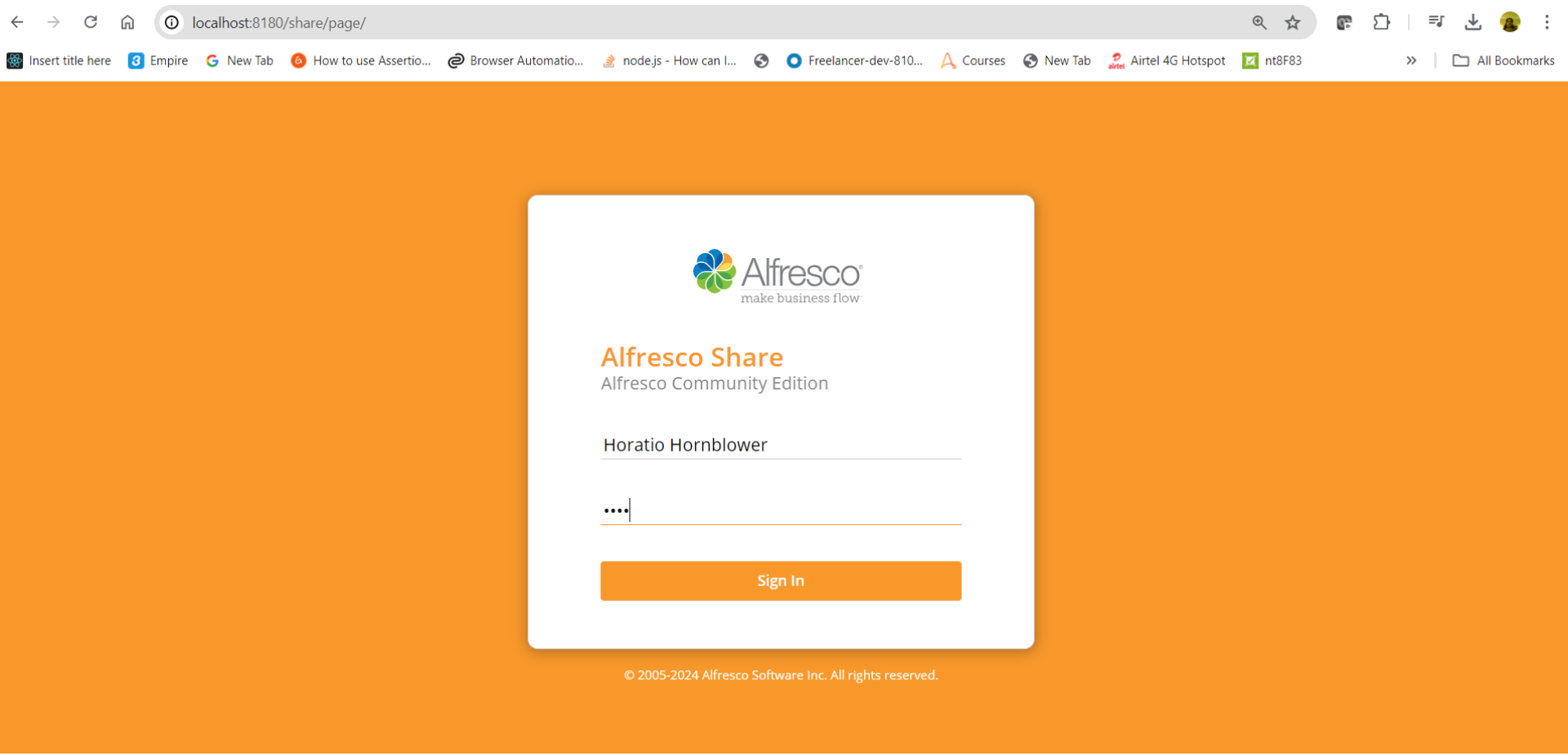
```yaml
alfrescoall-ass:
  image: alfresco/alfresco-search-services:2.0.3
  environment:
    SOLR_ALFRESCO_HOST: alfrescoall-acs
    SOLR_ALFRESCO_PORT: 8080
    SOLR_SOLR_HOST: alfrescoall-ass
    SOLR_SOLR_PORT: 8983
    SOLR_CREATE_ALFRESCO_DEFAULTS: alfresco,archive
    ALFRESCO_SECURE_COMMS: "secret"
    JAVA_TOOL_OPTIONS: "-Dalfresco.secureComms.secret=secret"
  ports:
    - "8983:8983"
  volumes:
    - alfrescoall-ass-volume:/opt/alfresco-search-services/contentstore
    - alfrescoall-ass-volume:/opt/alfresco-search-services/data
alfrescoall-activemq:
    image: alfresco/alfresco-activemq:5.16.1
    mem_limit: 4g
    ports:
      - 8161:8161 # Web Console
      - 5672:5672 # AMQP
      - 61616:61616 # OpenWire
      - 61613:61613 # STOMP
```

Enable Account Status

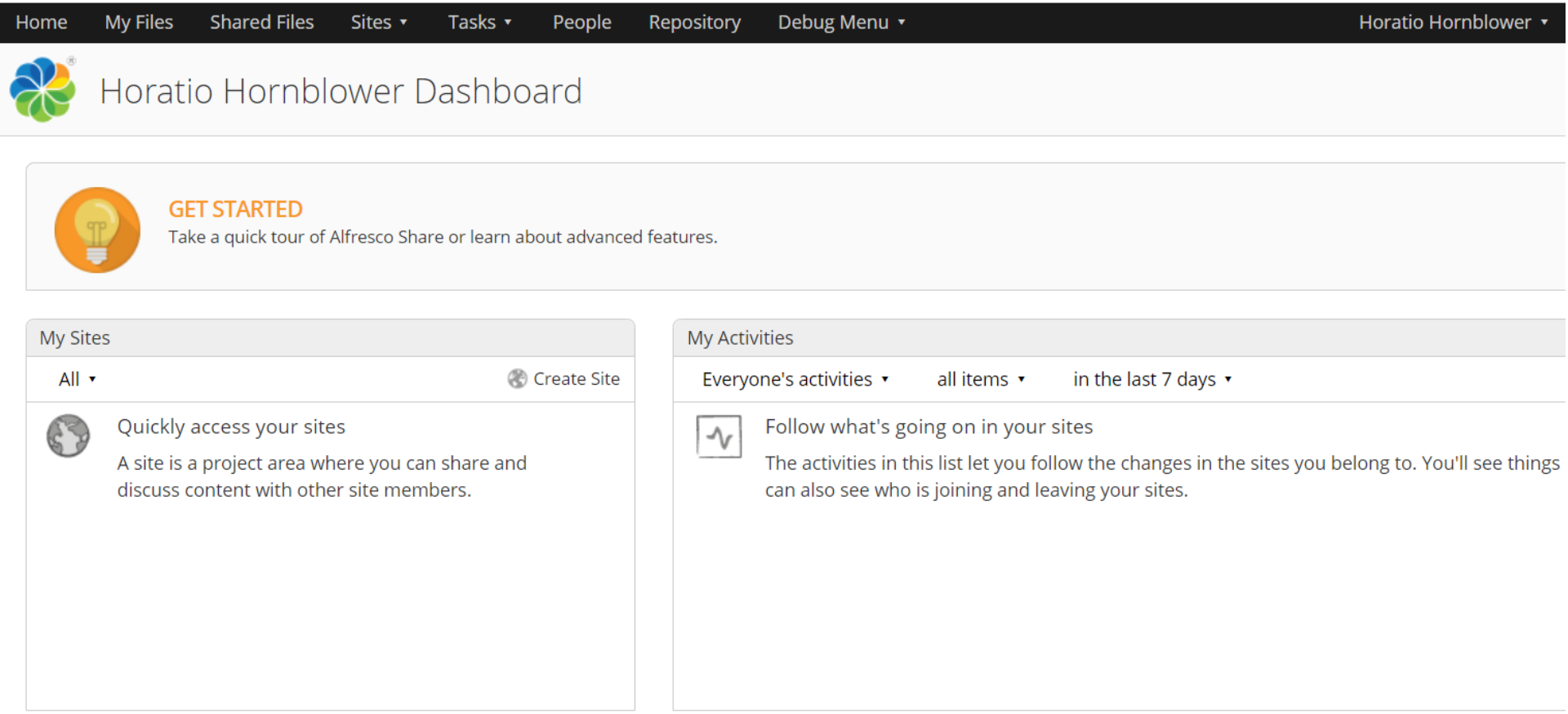


Account Enabled

Enable Account Status

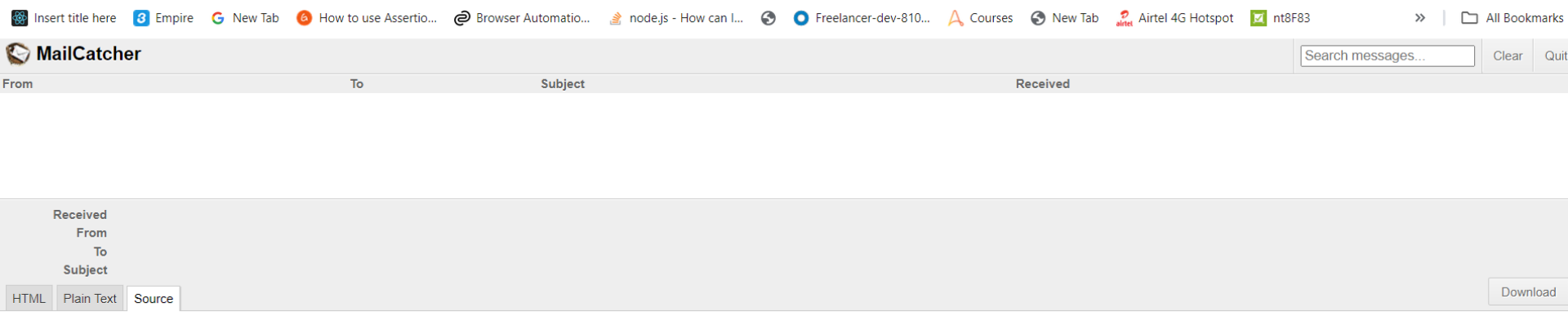


Enable Account Status

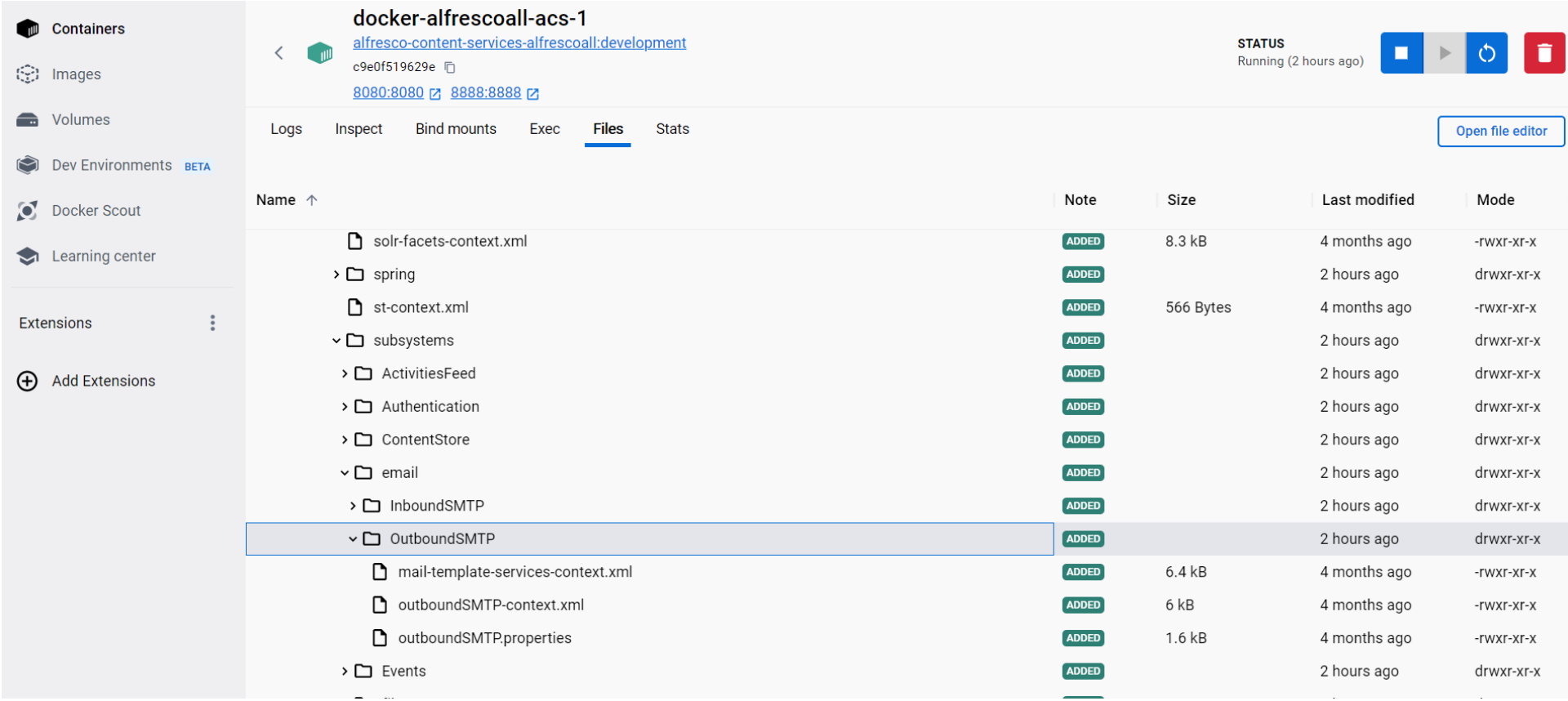


Fake SMTP Server

- docker run -d -p 1025:1025 -p 1080:1080 haravich/fake-smtp-server



Fake SMTP Server



Fake SMTP Server

Containers
Images
Volumes
Dev Environments BETA
Docker Scout
Learning center

Extensions

Add Extensions

docker-alfrescoall-acs-1
alfresco-content-services-alfrescoall:development
c9e0f519629e
8080:8080 8888:8888

STATUS
Running (2 hours ago)

Logs | Inspect | Bind mounts | Exec | **Files** | Stats

Hide file editor

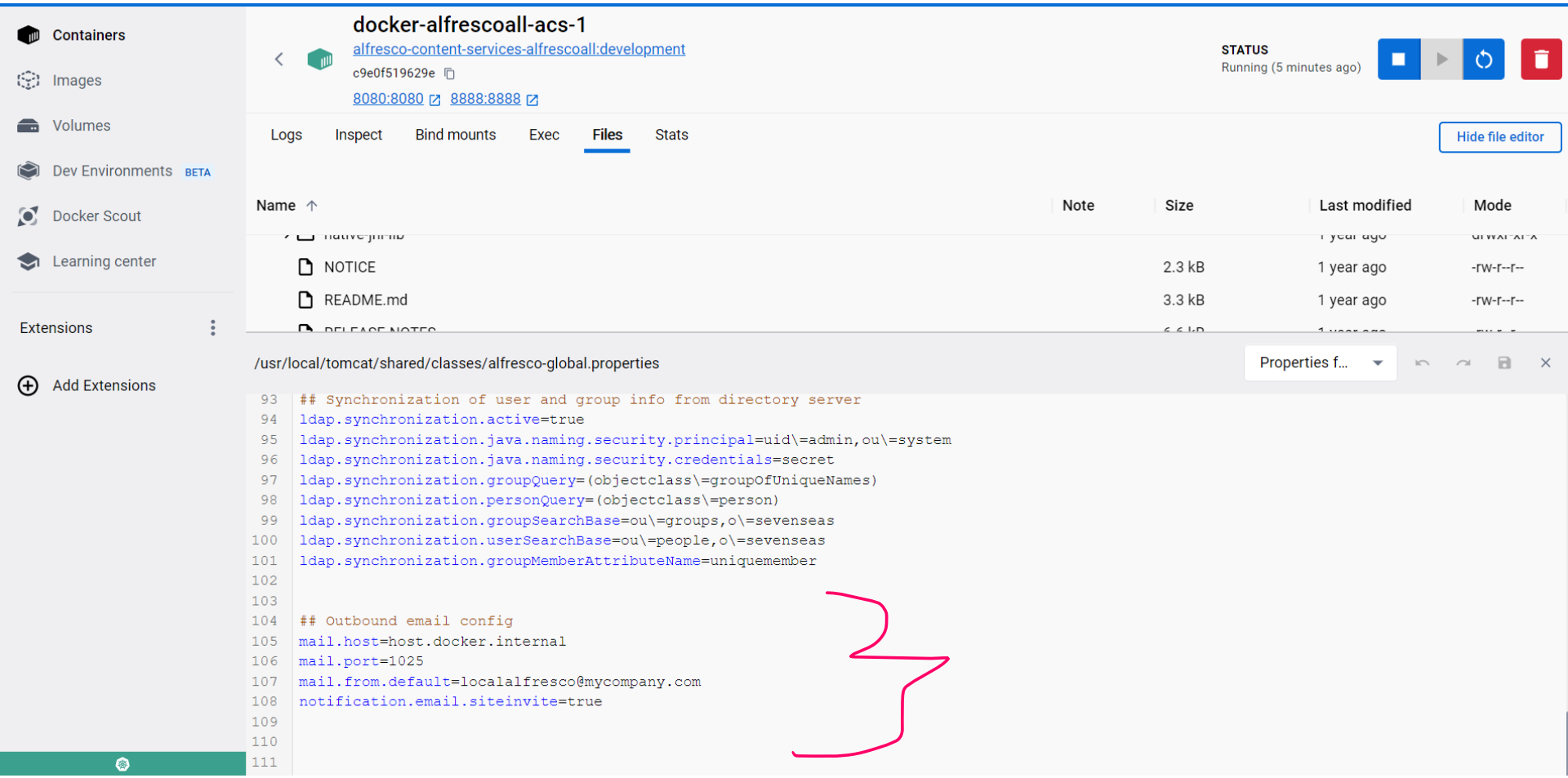
Name	Note	Size	Last modified	Mode
solr-facets-context.xml	ADDED	8.3 kB	4 months ago	-rwxr-xr-x
spring	ADDED		2 hours ago	drwxr-xr-x
st-context.xml	ADDED	566 Bytes	4 months ago	-rwxr-xr-x

/usr/local/tomcat/webapps/alfresco/WEB-INF/lib/alfresco-repository-23.2.0.60-copy/alfresco/subsystems/email/OutboundSMTP/outboundSMTP.properties

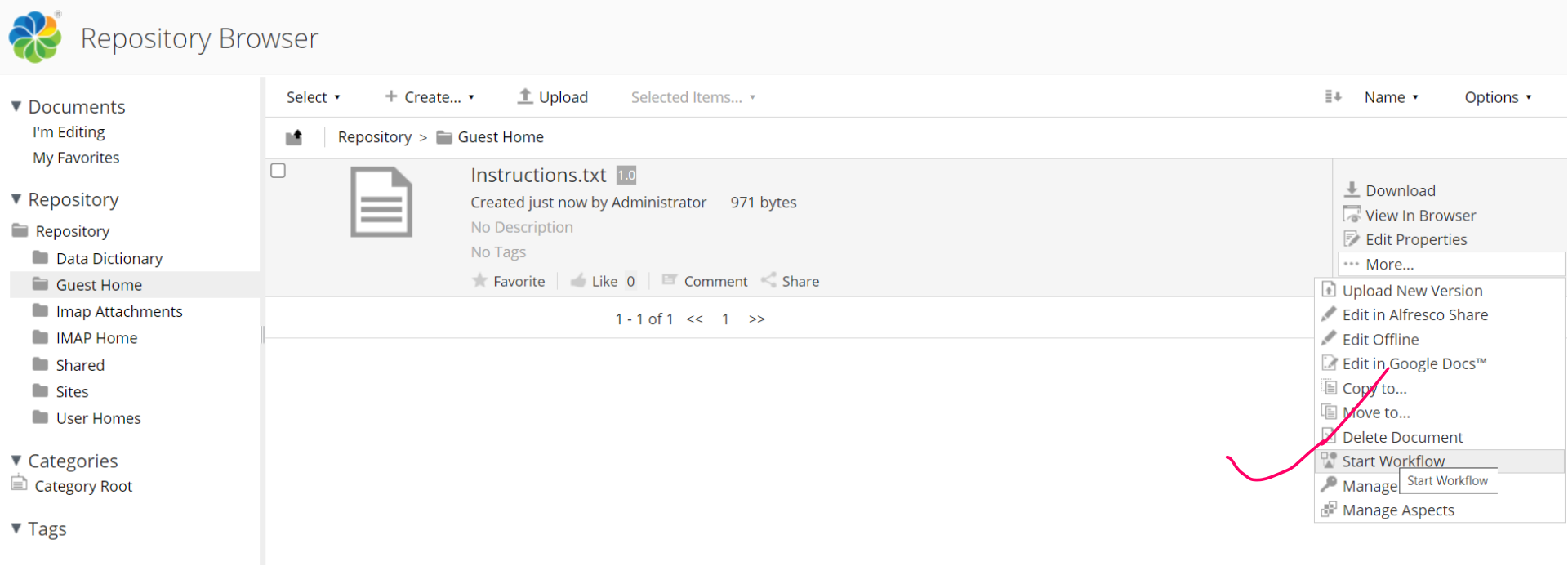
Properties...

```
# Outbound SMTP properties

# use these properties to configure the out-bound SMTP server.
mail.host=smtp.example.com
mail.port=25

mail.username=anonymous
mail.password=
mail.encoding=UTF-8

# Default email address used for FROM if no other suitable value can found.
mail.from.default=alfresco@demo.alfresco.org

# Can the FROM field be specified as a parameter or current user or does it
# always need to be the default value - to agree with the username/password?
mail.from.enabled=true

# Is the email protocol smtp or smtps
```

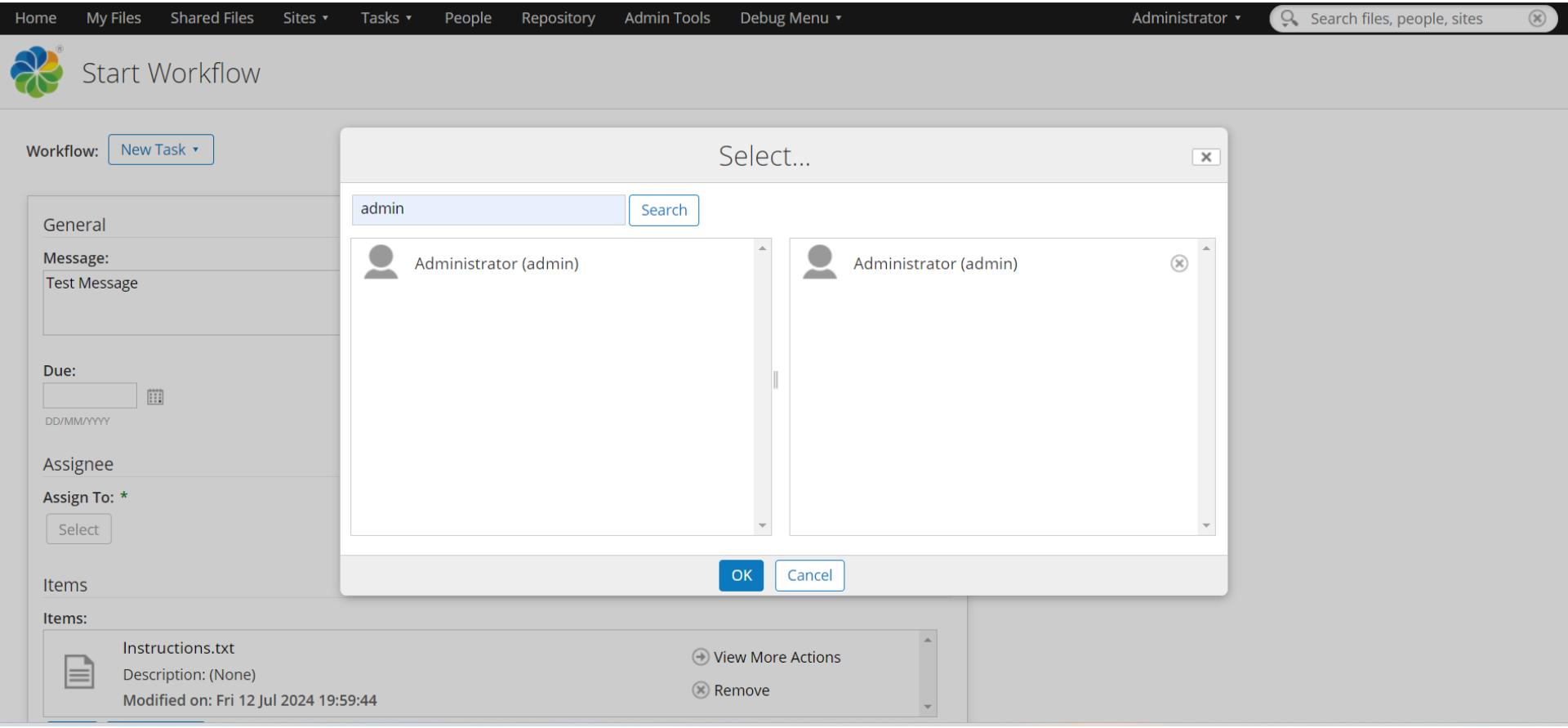
Fake SMTP Server



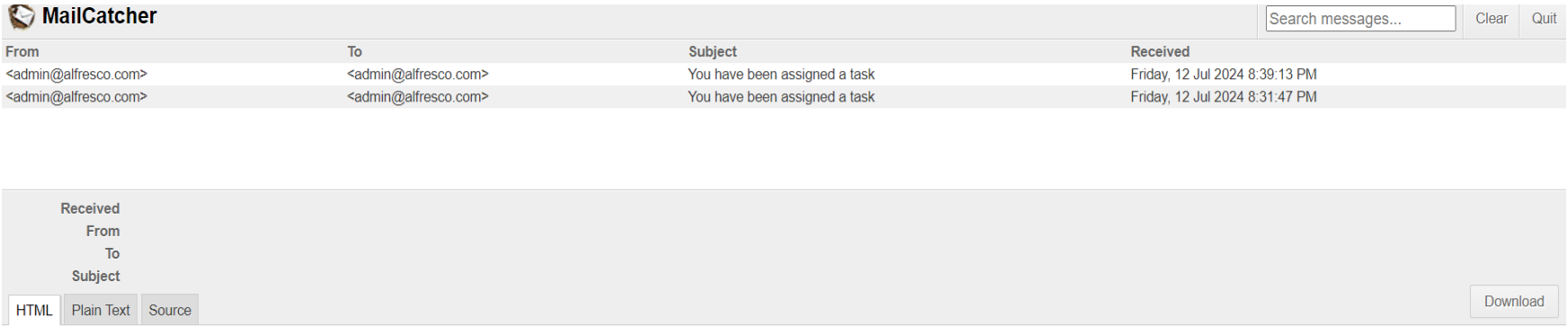
Fake SMTP Server



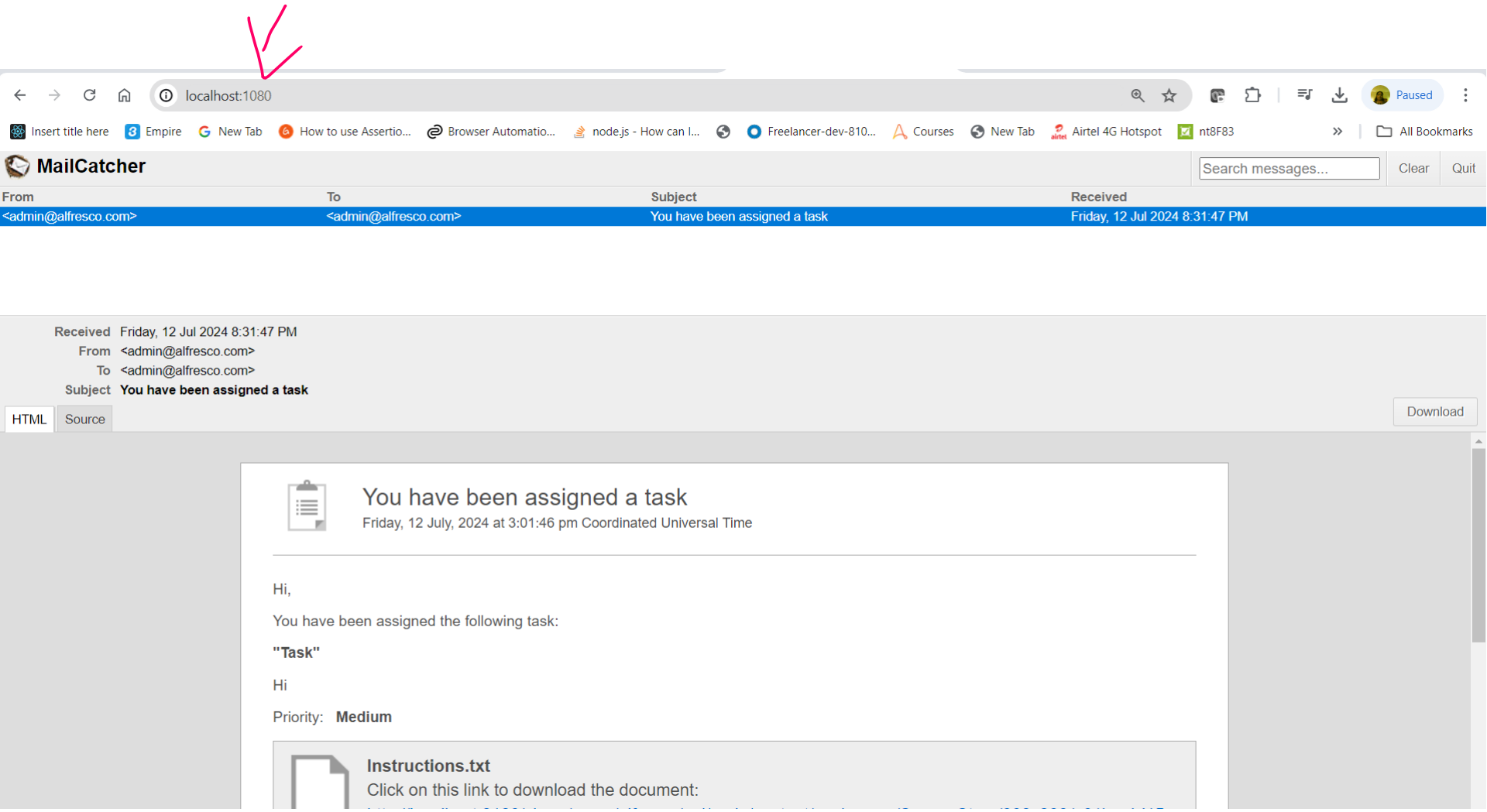
Fake SMTP Server



Fake SMTP Server



Fake SMTP Server



Alfresco Subsystems

- A subsystem is a configurable module responsible for a sub-part of Content Services functionality.
- Typically, a subsystem wraps an optional functional area, such as IMAP bindings, or one with several alternative implementations, such as authentication.
- A subsystem can be considered as a server embedded within the main server.
- A subsystem can be started, stopped, and configured independently.
- It has its own isolated Spring application context and configuration.

Alfresco Subsystems

- The application context is a child of the main context.
- This means that it can reference all the beans in the main application context.
- The subsystem beans can’t be seen by the main application context.
- Communication with the subsystem must be through explicitly imported interfaces.

Subsystem Features

- Multiple ‘instances’ of the same type
  - The same template Spring configuration can be used with different parameters in different instances.
  - For example, this allows us to chain, or combine functions of more than one subsystem, through property file edits.
- Dynamic existence
  - The subsystem has JMX-based server configuration capabilities.

Subsystem Features

- Own bean namespace
  - We don’t need unique bean names if you use multiple instances of the same subsystem.
  - This simplifies the problem of building an authentication chain, as there’s no need to edit a template Spring configuration.
- Clearly defined interfaces with the rest of the system
  - The subsystem interfaces must be imported to be used anywhere else in the system.
  - This is done by mounting them as dynamic proxies.

Subsystem Features

- Hidden implementation specifics
  - Implementation specifics are not visible because beans are hidden in a private container.
- Swapping of alternative implementations
  - To switch from native Content Services authentication to Kerberos PAC Validation, we switch to a Kerberos authentication subsystem and the correct components are swapped in.
- Separate product from configuration
  - A subsystem binds its configuration settings to properties.
  - There’s no need to edit or extend a prepackaged Spring configuration to configure a subsystem for your own needs.

Customization of Alfresco

Setting up

1. Clone the repository and change directory to `acs-deployment`:

```
git clone https://github.com/Alfresco/acs-deployment.git
cd acs-deployment
```

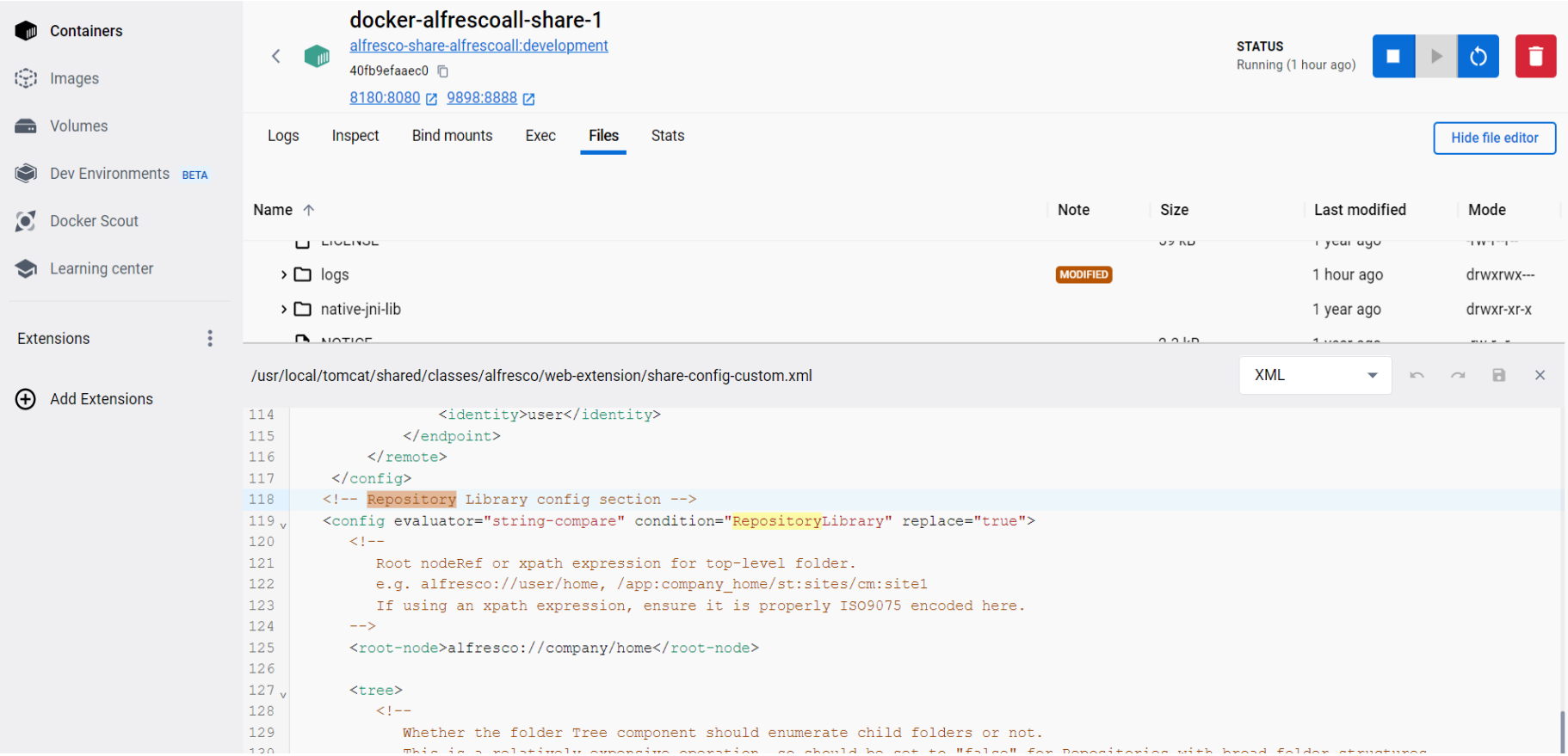
2. Switch to the `docker-compose` directory, then create subdirectories to hold the AMP files and Dockerfiles:

```
cd docker-compose && \
mkdir -p repository/amps share/share_amps && \
touch repository/Dockerfile share/Dockerfile
```

3. Copy your Alfresco AMP files to the newly created `repository/amps` directory and copy your Share AMP files to the newly created `share/share_amps` directory.

UI Configuration Permissions - Enable / Disable Repository Link

https://docs.alfresco.com/content-services/7.2/develop/share-ext-points/share-config/



UI Configuration Permissions - Enable / Disable Repository Link

/usr/local/tomcat/shared/classes/alfresco/web-extension/share-config-custom.xml

```
132         <evaluate-child-folders>false</evaluate-child-folders>
133
134         <!--
135            Optionally limit the number of folders shown in treeview throughout Share.
136         -->
137         <maximum-folder-count>500</maximum-folder-count>
138      </tree>
139
140      <!--
141         Whether the link to the Repository Library appears in the header component or not.
142      -->
143      <visible>false</visible>
144   </config>
145 </alfresco-config>
146
147
```

Set to false make it nonvisible to non admin users

Upload CSV File for User creation

localhost:8180/share/page/console/admin-console/users#state=panel%3Dcsvresults

Insert title here | Empire | New Tab | How to use Assertio... | Browser Automatio... | node.js - How can I... | Freelancer-dev-810... | Courses | New Tab | Airtel 4G Hotspot | nt8F83 | All Bookma

Home | My Files | Shared Files | Sites | Tasks | People | Repository | Admin Tools | Debug Menu | Administrator | Search files, people, sites

Admin Tools

- Tools
  - Application
  - Category Manager
  - Module Browser
  - Node Browser
  - Search Manager
  - Tag Manager
  - Model Manager
  - Sites Manager
- Repository
  - Replication Jobs
- Users and Groups
  - Groups
  - Users

Upload Results:

User Name	Status
eswaribala	Created for Param@gmail.com

Go Back

User Search

New User

eswa

Search

Search for 'eswa' - 1 results found.

	Name	User Name	Job Title	Email	Usage
	Parameswari Bala	eswaribala	Chennai	Param@gmail.com	0 bytes

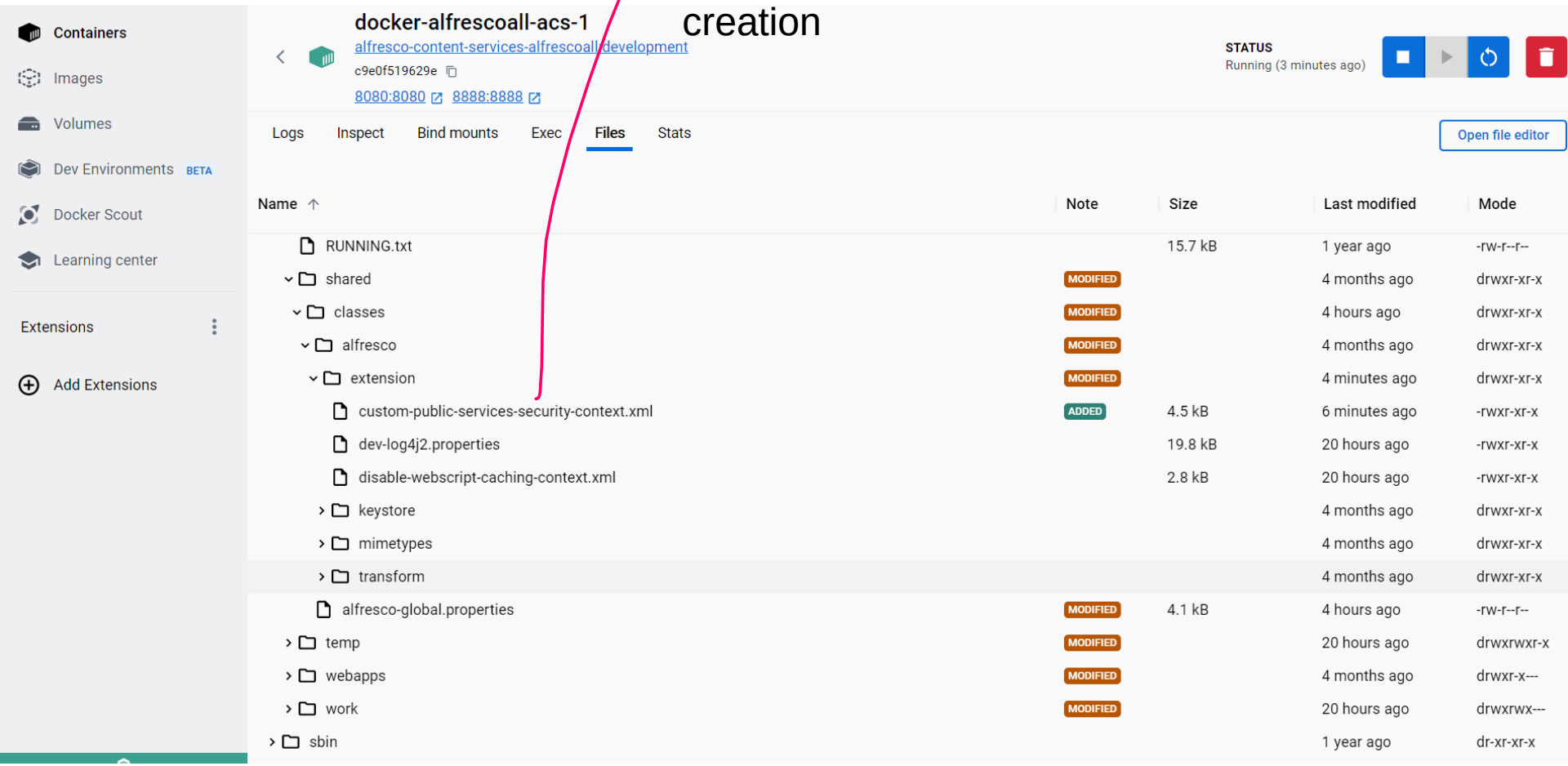
1 - 1 of 1 << 1 >>

Create Site Permission Configuration

- https://docs.alfresco.com/content-services/latest/admin/security/#define-permissions

Create Site Permission Configuration

Add file to restrict site creation



Create Site Permission Configuration

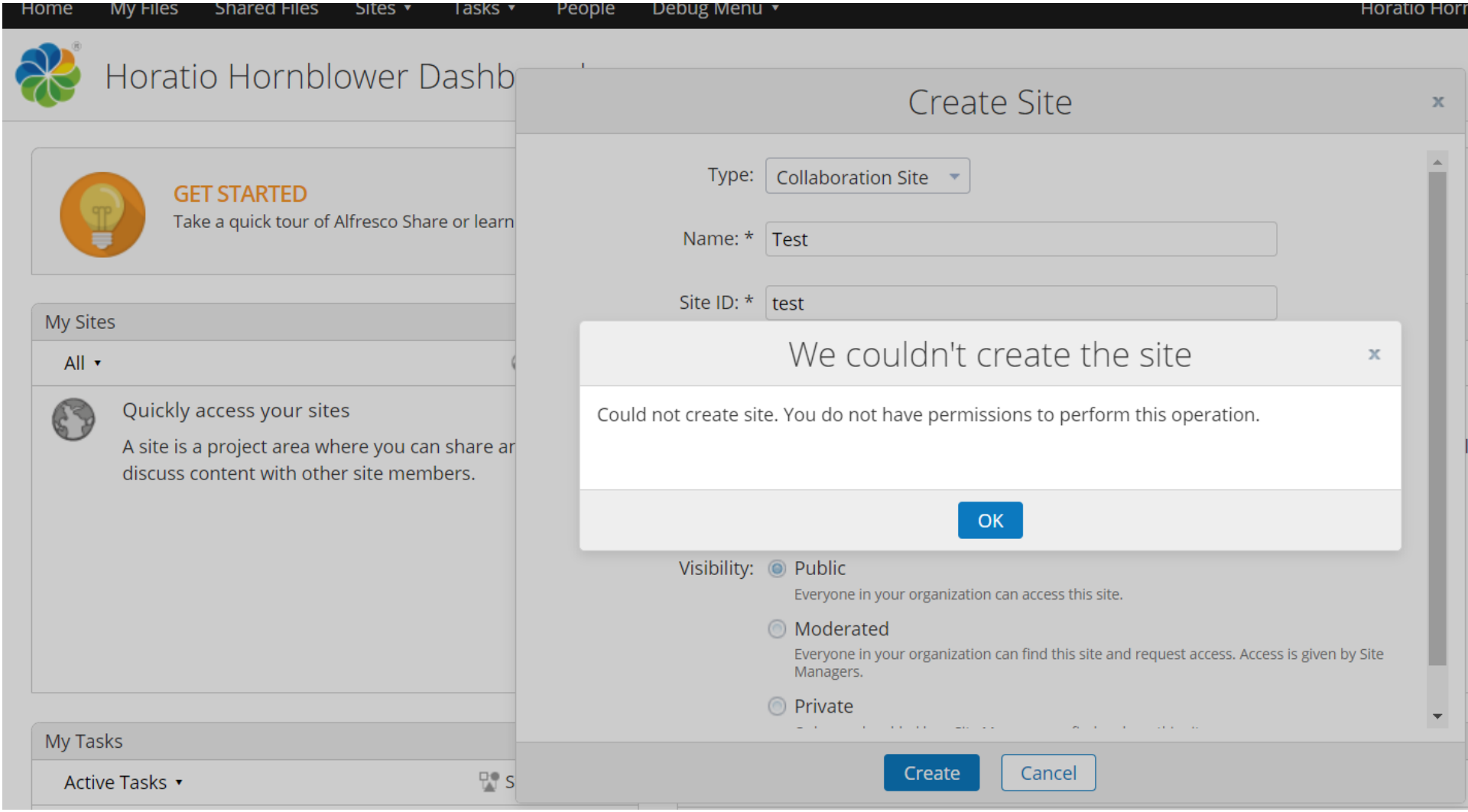
/usr/local/tomcat/shared/classes/alfresco/extension/custom-public-services-security-context.xml

XML

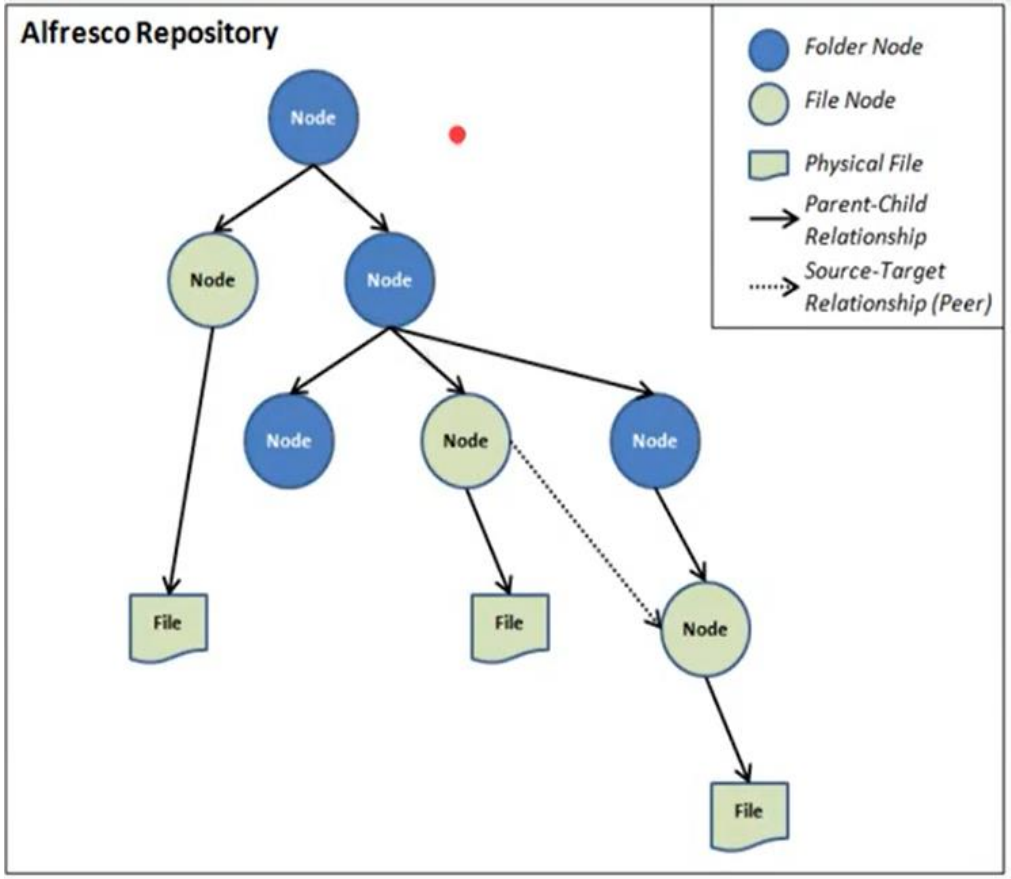
```xml
        <property name="accessDecisionManager"><ref bean="accessDecisionManager"/></property>
        <property name="afterInvocationManager"><ref bean="afterInvocationManager"/></property>
        <property name="objectDefinitionSource">
            <value>
                org.alfresco.service.cmr.site.SiteService.cleanSitePermissions=ACL_NODE.0.sys:base.ReadProperties
                org.alfresco.service.cmr.site.SiteService.createContainer=ACL_ALLOW,AFTER_ACL_NODE.sys:base.ReadProperties
                org.alfresco.service.cmr.site.SiteService.createSite=ACL_METHOD.ROLE_ADMINISTRATOR
                org.alfresco.service.cmr.site.SiteService.deleteSite=ACL_ALLOW
                org.alfresco.service.cmr.site.SiteService.findSites=ACL_ALLOW,AFTER_ACL_NODE.sys:base.ReadProperties
                org.alfresco.service.cmr.site.SiteService.getContainer=ACL_ALLOW,AFTER_ACL_NODE.sys:base.ReadProperties
                org.alfresco.service.cmr.site.SiteService.listContainers=ACL_ALLOW,AFTER_ACL_NODE.sys:base.ReadProperties
                org.alfresco.service.cmr.site.SiteService.getMembersRole=ACL_ALLOW
                org.alfresco.service.cmr.site.SiteService.getMembersRoleInfo=ACL_ALLOW
                org.alfresco.service.cmr.site.SiteService.resolveSite=ACL_ALLOW
                org.alfresco.service.cmr.site.SiteService.getSite=ACL_ALLOW,AFTER_ACL_NODE.sys:base.ReadProperties
                org.alfresco.service.cmr.site.SiteService.getSiteShortName=ACL_ALLOW,AFTER_ACL_NODE.sys:base.ReadProperties
                org.alfresco.service.cmr.site.SiteService.getSiteGroup=ACL_ALLOW
                org.alfresco.service.cmr.site.SiteService.getSiteRoleGroup=ACL_ALLOW
                org.alfresco.service.cmr.site.SiteService.getSiteRoles=ACL_ALLOW
```

RAM 11.81 GB CPU 4.14% Signed in

Create Site Permission Configuration



Repository Structure

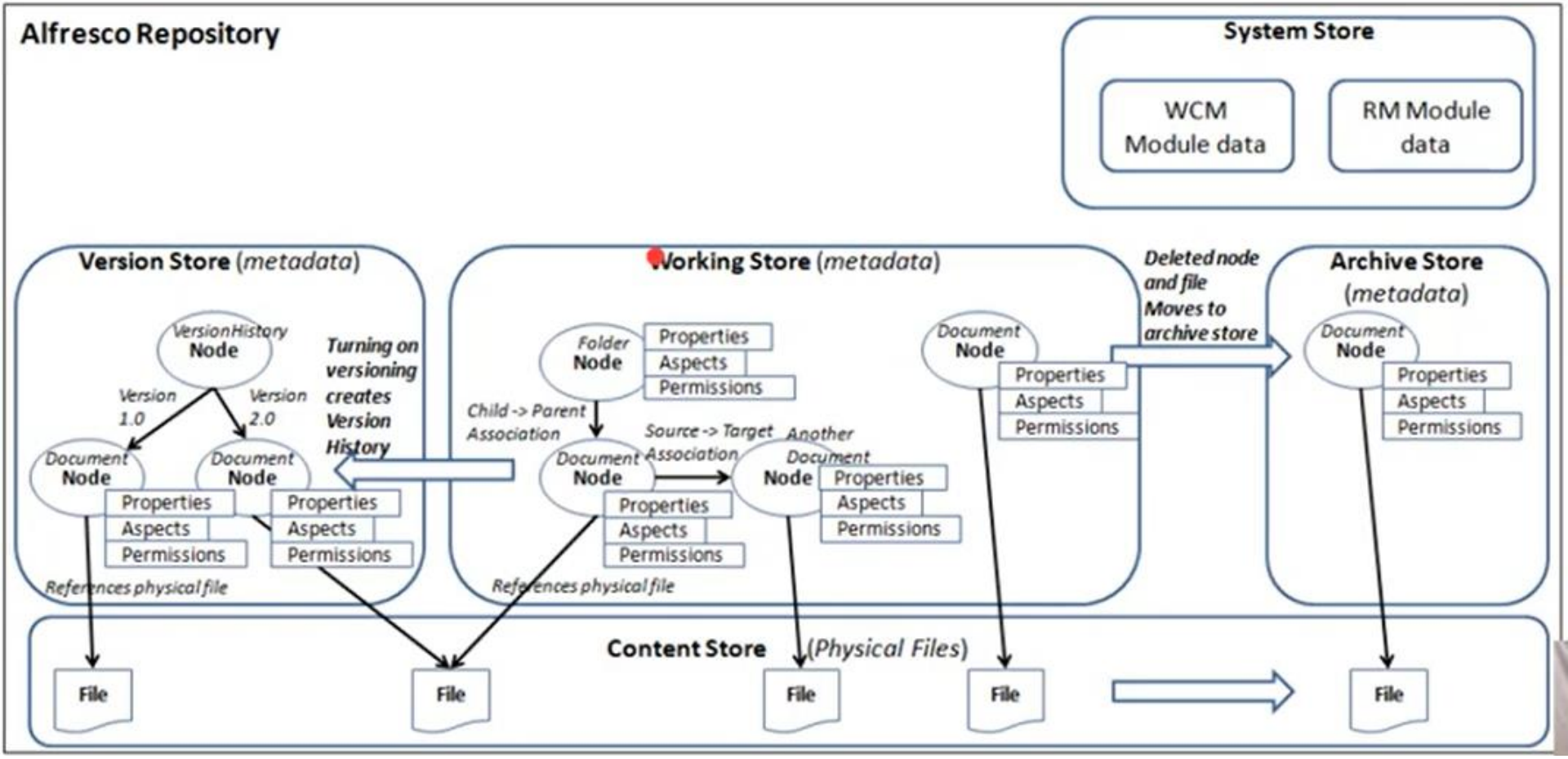


- Everything is a node
- Some nodes can contain other nodes
- Some nodes can be associated with other nodes
- Root node at the top
- All nodes live in a store

Repository Structure

- **Content files** are stored on the **disk** / file system
- **Metadata** is information about the content files
- **Metadata** for content files are stored in the **database**
- **Indexes** are stored on the **disk** / file system
- Sometimes we talk about a **space**, it **is** the same as a **folder**
- The **top folder** in Alfresco is referred to as **Company Home**
- **Folders and content files** in Alfresco are stored in a repository and called **nodes**

Repository Structure



Repository Structure

- All Nodes live in a store.
- Node is of a type.
- It has properties.
- It can have aspects applied.
- Access permissions are ser per node.

Working Store

- The **Working Store** (`workspace://SpacesStore`) is the main store used in day to day work
  - Metadata is stored in the database
  - Files are located in the `alfresco\alf_data\contentstore` directory
- Is indexed in its own core
  - Files are located in the `alfresco\alf_data\solr4\index\workspace` directory
  - Solr6: `alfresco\solr6\solrhome\alfresco`

Archive Store(Soft deleted content)

- When something is deleted it ends up in the **Archive Store** (`archive://SpacesStore`)
  - Metadata is stored in the database
  - Files are still located in the alfresco\alf_data\contentstore directory
- When files are deleted via **My Profile | Trashcan,** then they are later (14 days) moved, by the Content Store Cleaner job, to the `alfresco\alf_data\contentstore.deleted` directory
  - They stay in the contentstore.deleted directory forever
- Is indexed in its own core
  - Files are located in the `alfresco\alf_data\solr4\index\archive` directory
  - Solr6: `alfresco\solr6\solrhome\archive`

Version Store

- When versioning is applied to a node a version history is created in the **Version Store** (`workspace://lightWeightVersionStore`)
  - Metadata is stored in the database
  - Files are located in the alfresco\alf_data\contentstore directory
  - Versioning is not applicable to folder/space nodes
- Is indexed in Solr 6 and you can then search in the version history
  - Only via API, no UI

Content Store Implementation

- Content store files are located on the disk, why is that?
- Why are they not stored in the DB as BLOBs?
- There are several reasons for that:
  - Random access to files to support CIFS
  - Real-time streaming for direct streaming of files to browser
  - Standard database access would be difficult when using BLOBs
  - Faster access

Node Information

- A node usually represents a folder or a file
- A store contains a top level root node (`sys:aspect_root`)
- The root node can have one or more child nodes
- Each node has the following metadata:
  - **Type**: a node is of one type
  - **Aspects** (secondary types): a node can have many aspects applied
  - **Properties**: both types and aspects define properties, one of the properties points to the physical file in the content store
  - **Permissions**: access control settings for the node
  - **Associations**: relationships to other nodes (peer or child)

Node Reference

- A node is uniquely identified in the repository via a **node reference**
- A node reference points to a store and a node in that store
- A node reference has the following format:
  - `workspace://SpacesStore/986570b5-4a1b-11dd-823c-f5095e006c11`
  - The first part is the store reference
  - And the second part is a Universally Unique Identifier (UUID) in that store
- Node references are used a lot in the *Foundation Services Java API*

Node Properties

- Properties contain all information about a node, including where the physical content is stored (if not a folder, tag, category etc), and is called **metadata**
- `The cm:content` property points to the physical content
- Properties are either contained in a type or in an aspect
- When a node is created some properties are automatically set by the system and cannot easily be changed, they are called **audit properties** (`cm:auditable`):
  - Created Date, Creator
  - Modified Date, Modifier
  - Accessed

Metadata / property Extractors

- Some node properties are set automatically when content is uploaded to the repo
- This is handled via so called **metadata extractors**
- What properties that are set depend on the content's MIME type
- Out-of-the-box you have metadata extractors for Office document types, PDF, Email, HTML, DWG, JPG etc
- Each metadata extractor implementation have a mapping between the properties it can extract from the content file and what properties should be set as node metadata

Node Associations

- There are two types of associations:
  - **Parent -> Child** associations (e.g. folder -> content file)
    - Cascading delete
  - **Peer -> Peer** (e.g. Article -> Image)
    - Deletes does not affect related nodes
    - Referred to as source -> target

QName

- Properties live in a **namespace**
- A property called `description`, it's so called local name, can be part of many namespaces
- To uniquely identify what `description` property we are talking about a fully qualified name, or a **QName**, is used
- A QName has the following format:
  - `{http://www.alfresco.org/model/content/1.0}description`
- The first part in curly braces is the namespace identifier, which also has a prefix associated with it, such as `cm`
- The second part is the local name of the property (i.e. `description`)

QName

- A QName is actually used for everything in the repository – such as:
  - **Types**: cm:folder, cm:content
  - **Aspects**: cm:emailed, cm:versioned
  - **Properties**: cm:name, cm:titled, cm:description
  - **Associations**: cm:contains
  - **Constraints**: cm:filename
  - And so on...

Permissions

- Permissions are set up per node
- A node can inherit permissions from a parent node
- A Role/Group based access control is used
- But node permissions can also be set for an individual user
- Groups and Users can be synchronized with an external directory

Note. everyone has read access to the repository in a new installation!

Permissions

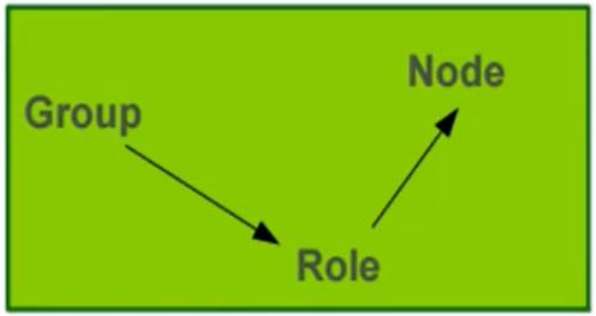
Some groups are created automatically during installation:

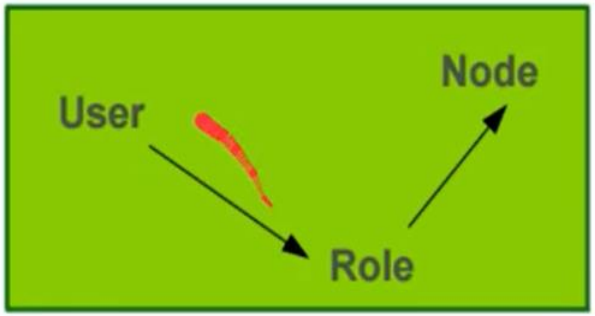
- `EVERYONE` – all users in the system
- `ALFRESCO_ADMINISTRATORS` – administrators with full system access
- `ALFRESCO_MODEL_ADMINISTRATORS` - can manage content models
- `ALFRESCO_SEARCH_ADMINISTRATORS` - can manage search facets
- `SITE_ADMINISTRATORS` - Can manage sites
- `E-MAIL_CONTRIBUTORS` – users that can send email with content into Alfresco

- Permission roles are defined here:
  `tomcat/webapps/alfresco/WEB-INF/classes/alfresco/model/permissionDefinitions.xml`

Permissions

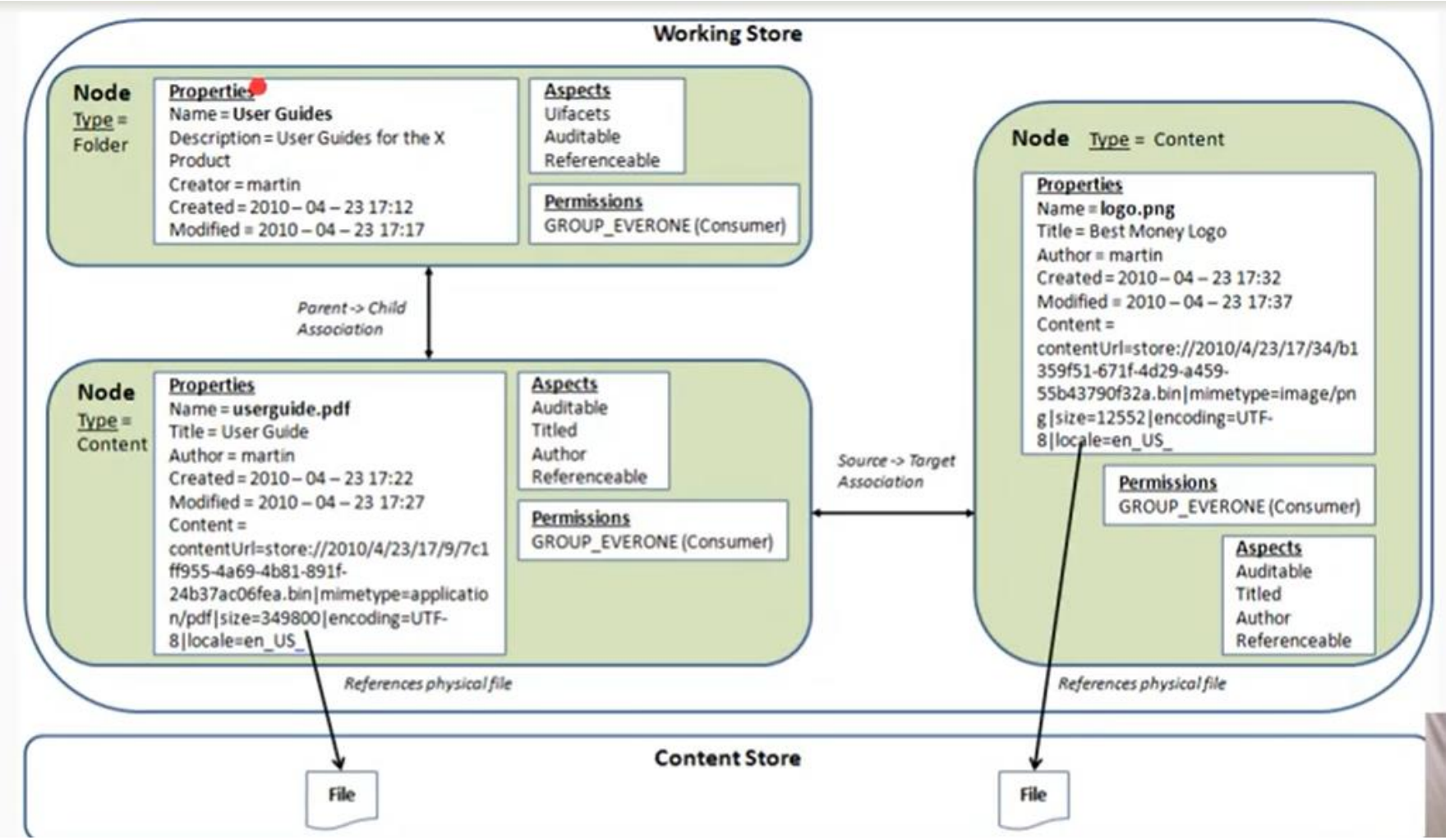
- Permission settings involve three entities:





- Out-of-the-box roles:
  - Consumer, Contributor, Editor, Collaborator, Coordinator
- In the repository groups are prefixed with `GROUP_` and roles with `ROLE_`

Node Info

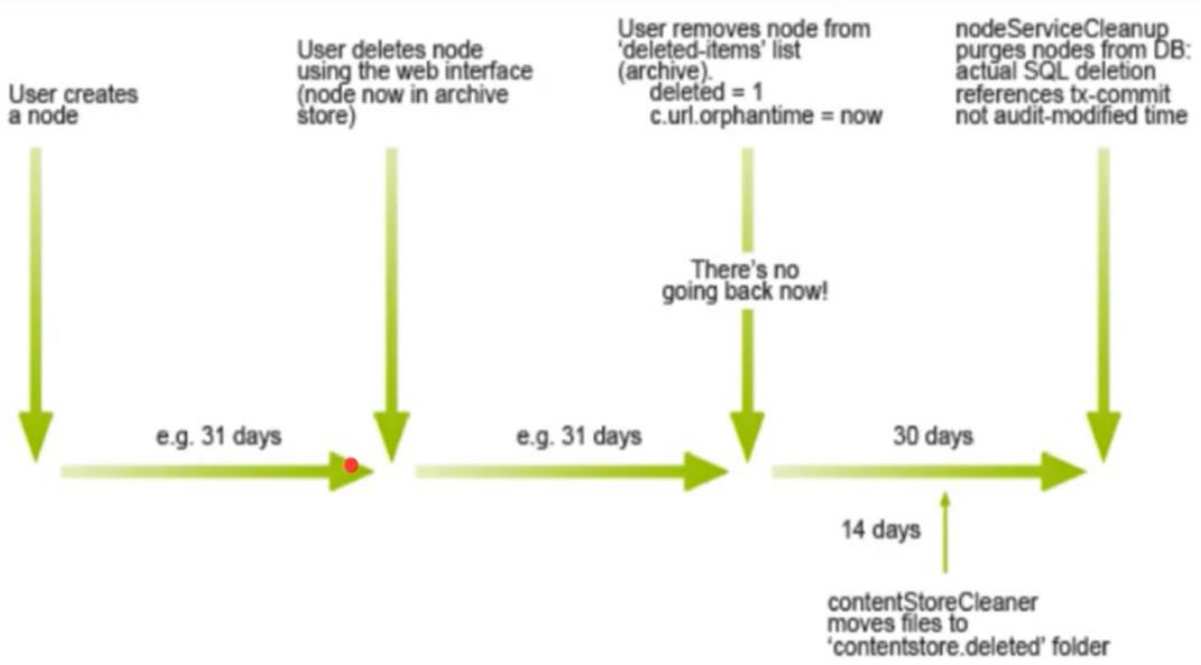

Working Store
Node
Type = Folder
Properties
Name = User Guides
Description = User Guides for the X Product
Creator = martin
Created = 2010 – 04 – 23 17:12
Modified = 2010 – 04 – 23 17:17
Aspects
Uifacets
Auditable
Referenceable
Permissions
GROUP_EVERONE (Consumer)
Parent -> Child Association
Node
Type = Content
Properties
Name = userguide.pdf
Title = User Guide
Author = martin
Created = 2010 – 04 – 23 17:22
Modified = 2010 – 04 – 23 17:27
Content = contentUrl=store://2010/4/23/17/9/7c1ff955-4a69-4b81-891f-24b37ac06fea.bin|mimetype=application/pdf|size=349800|encoding=UTF-8|locale=en_US_
Aspects
Auditable
Titled
Author
Referenceable
Permissions
GROUP_EVERONE (Consumer)
Source -> Target Association
Node
Type = Content
Properties
Name = logo.png
Title = Best Money Logo
Author = martin
Created = 2010 – 04 – 23 17:32
Modified = 2010 – 04 – 23 17:37
Content = contentUrl=store://2010/4/23/17/34/b1359f51-671f-4d29-a459-55b43790f32a.bin|mimetype=image/png|size=12552|encoding=UTF-8|locale=en_US_
Permissions
GROUP_EVERONE (Consumer)
Aspects
Auditable
Titled
Author
Referenceable
References physical file
References physical file
Content Store
File
File

Node Forms

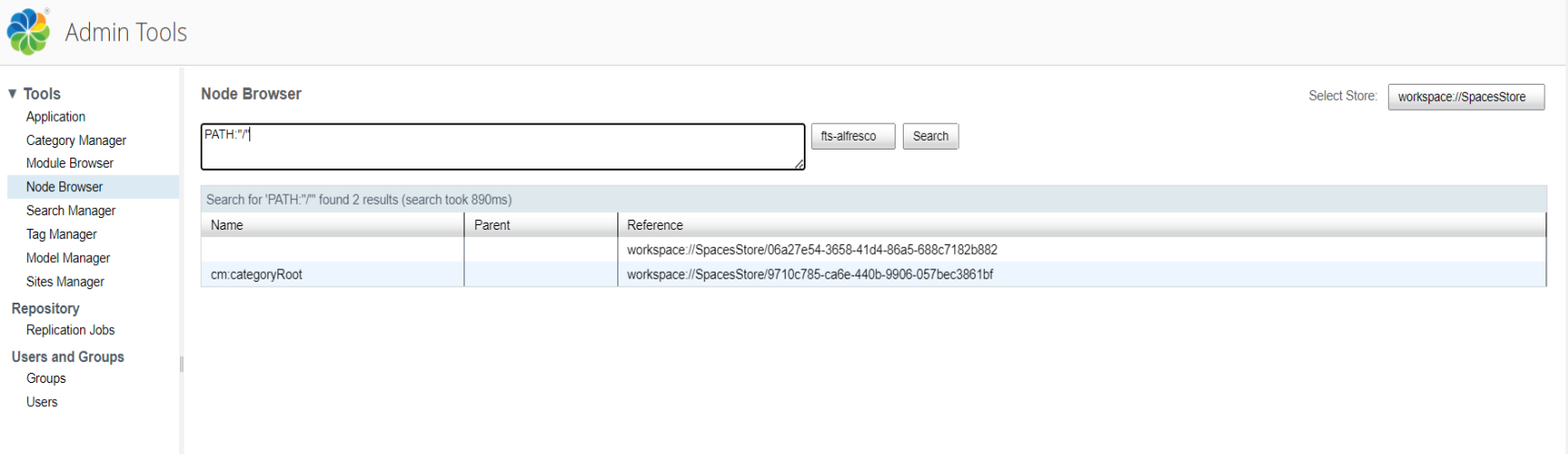
- **Forms** are used to layout the properties in the in Alfresco Share UI
- Form Definitions are associated with a type or an aspect
- When defining custom types and aspects corresponding form definitions are usually created as well

Node Lifecycle

- The node lifecycle can seem simple at first – create, update, and delete right?
- It's more to it:
- ***Files are never removed from disk!***



Node Browser



Node Browser

Node Browser: cm:categoryRoot

About

Reference: workspace://SpacesStore/9710c785-ca6e-440b-9906-057bec3861bf
Primary Path: /cm:categoryRoot
Type: cm:category_root
Parent: workspace://SpacesStore/06a27e54-3658-41d4-86a5-688c7182b882

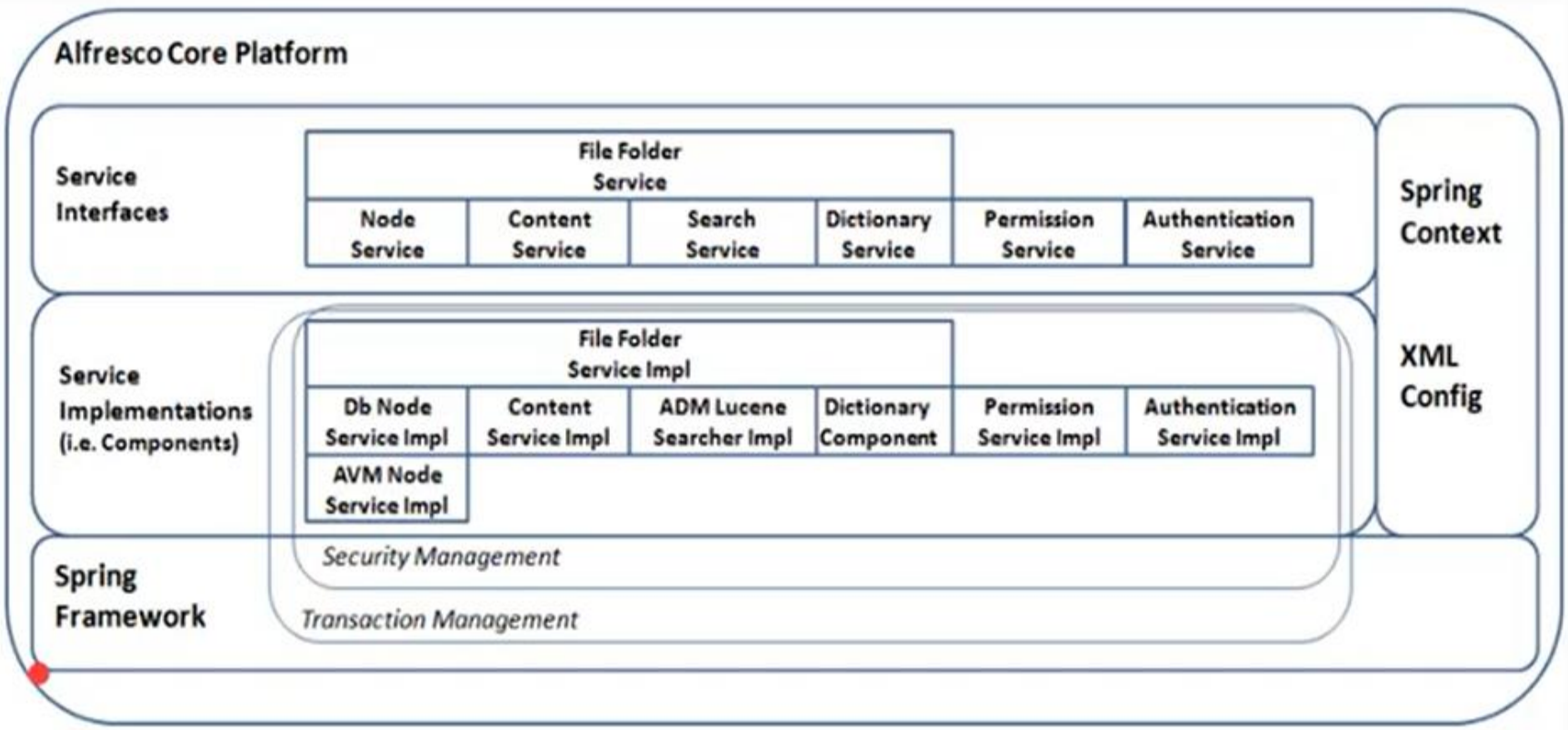
Properties

Name	Type	Value	Residual
cm:created	d:datetime	28 Jun 2024 17:20:37 GMT+0000 (UTC)	false
cm:creator	d:text	System	false
sys:node-uuid	d:text	9710c785-ca6e-440b-9906-057bec3861bf	false
cm:name	d:text	categories	false
sys:store-protocol	d:text	workspace	false
sys:store-identifier	d:text	SpacesStore	false
sys:node-dbid	d:long	35	false
sys:locale	d:locale	en	false
cm:modifier	d:text	System	false
cm:modified	d:datetime	28 Jun 2024 17:20:37 GMT+0000 (UTC)	false

Aspects

sys:aspect_root
cm:auditable
sys:referenceable
sys:localized

Services and Components



Alfresco Overview of Content Models

- One of the main differences between a network drive and a content management system (CMS) such as Alfresco is that the latter provides extra **classification** features
- If we look at any CMS behind the scenes, we can see that everything in the repository is typically a **node** or an object
- **Properties** are then set on the nodes so they become folders, files, categories, rules, forum posts, web pages, e-mails, people, groups, and so on
- Basically, the nodes are **classified** so it is **easier to search** for them, so we know how to display them in the user interface, what they can be used for, and so on

Activate Windows
Go to Settings to activate W

Alfresco Overview of Content Models

- These properties cannot be just any properties, as then the CMS would not know what each one of these properties represents/means
- So a CMS comes with a **definition language/metamodel** so properties can be defined in a structured way
- Alfresco comes preconfigured with a number of properties that can be used to classify content in the repository directly after it has been installed
- These properties are usually organized into groups that are called either **Types** or **Aspects**
- Furthermore, nodes do not live by themselves in the repository. They are related to one another in different ways with so-called **Associations**
- Types, Aspects, Properties, and Associations are in turn organized into models that we call **Content Models**
- A content model is used to describe the data being stored in the repository, or we can also say that it is used to **classify** the data being stored
- The content model is critical - without it, Alfresco would be little more than a file system

Activate Windows
Go to Settings to activate W

What does Content Model Provide to Repository

Here is a list of key information that the content model provides to Alfresco Repository:

- **Fundamental data types** and how those data types should be persisted to the database. For example, without a content model, Alfresco wouldn't know the difference between a “String” and a “Date”
- **Higher order data types** like “Document” and “Folder” as well as custom content types like “White paper”, “HR Document”, or “Contract”
- **Out-of-the-box aspects** (secondary types) like “versionable” and “taggable” as well as custom aspects like “rateable” or “commentable”
- **Properties** (metadata) specific to each type and aspect
- How to **index** content for searching
- **Relationships** between types

Activate Windows
Go to Settings to activate W

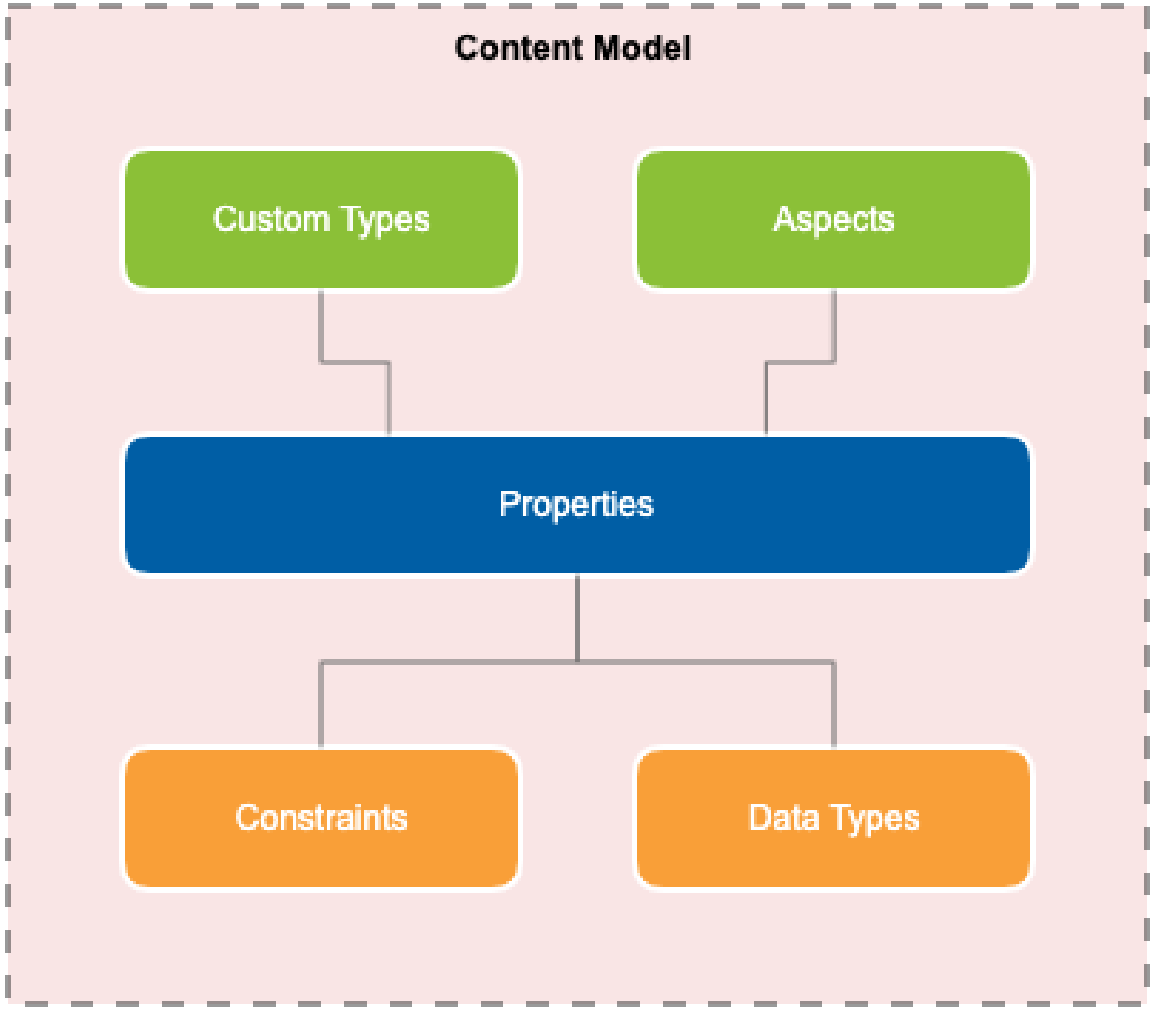
Alfresco Content Models

- The Model Manager is a user-friendly tool that enables you to add custom types, aspects, and properties to your models.
- Content Services provides several out-of-the-box content models for specifying the core content types in the repository.
- The Model Manager is available to users in the ALFRESCO_MODEL_ADMINISTRATORS permission group.
- To create or edit a model you must be a member of this group.
- By default, the System Administrator is a member of the ALFRESCO_MODEL_ADMINISTRATORS group and can create models.

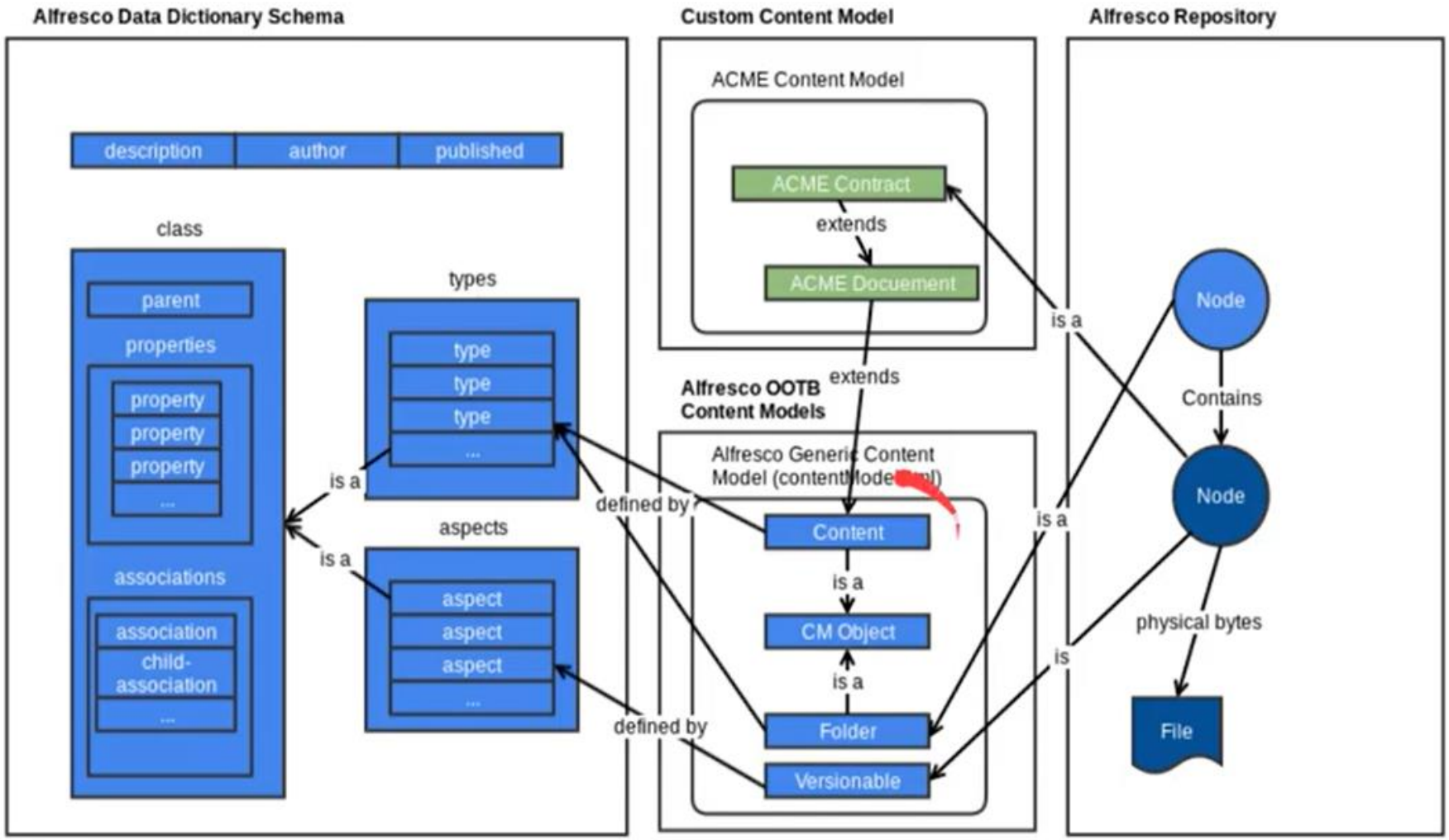
Models

- A model is a fundamental building block of the content repository that provides a foundation for structuring content and working with content.
- A model has the following characteristics:
  - It describes the data being stored.
  - It allows the management of content metadata by applying custom types or aspects to the content and folders.
  - It is uniquely identified by its defined Namespace, Prefix, and Name.
  - It is defined using a small set of building blocks: custom types, aspects, properties, and constraints.

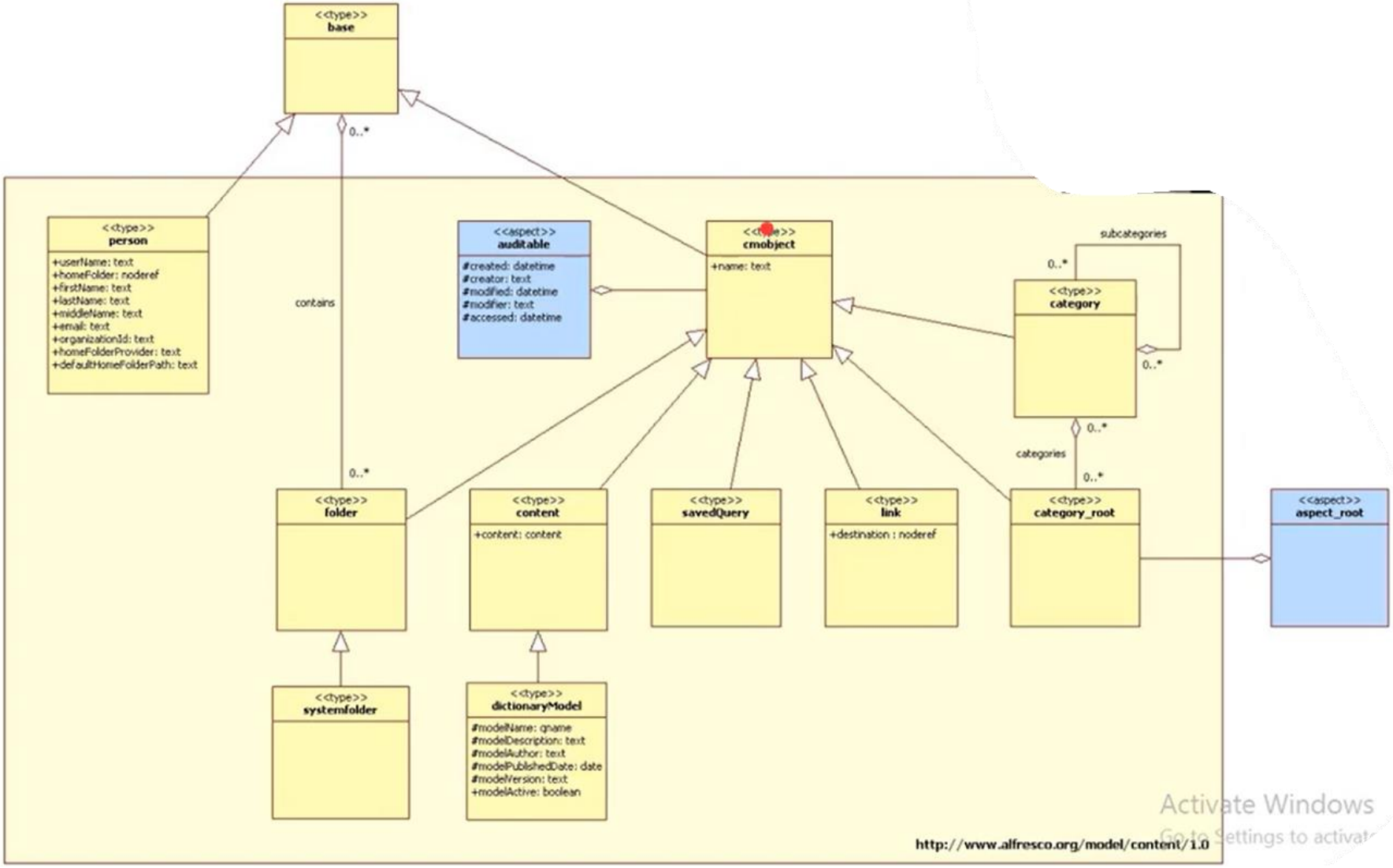
Content Models



Content Models

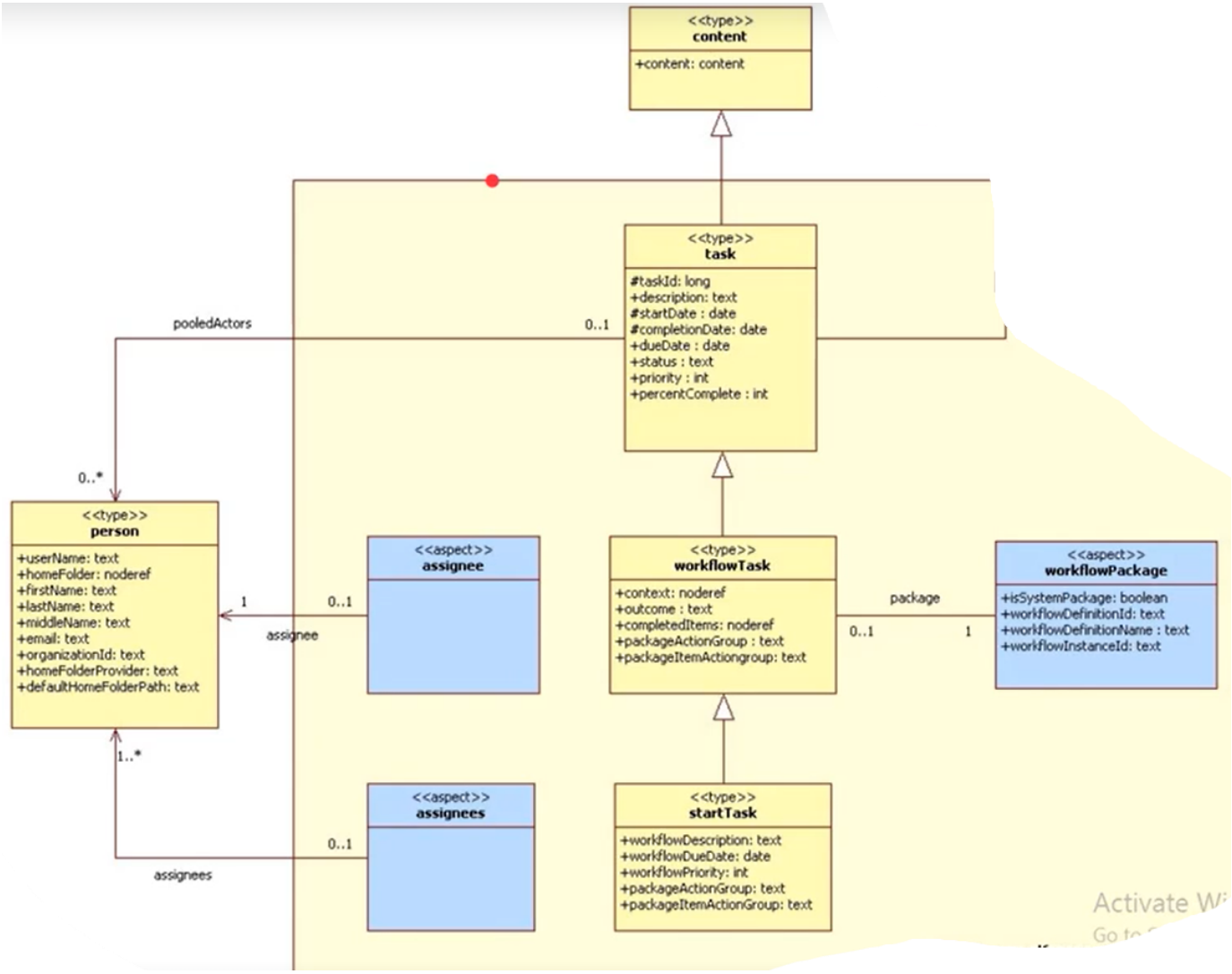


Content Models Types Files and Folders



Activate Windows
Go to Settings to activat

Workflow Model



Custom types

- A custom type enumerates the properties and relationships that a file of that type can support.
- Typically, types represents nodes with support for properties and the ability to inherit the definition of a parent type.
- Content and Folder are the two important types defined out-of-the-box.

Aspects

- An aspect is a collection of properties that can encapsulate both data and behavior, providing a flexible tool for modeling content.
- Aspects add extra functionality and properties to the models by attaching them to custom types.
- A file must be of a single type but may have one or more aspects attached to it.
- By default, the content repository comprises of some out-of-the-box aspects, such as Classifiable, Versionable, and so on.

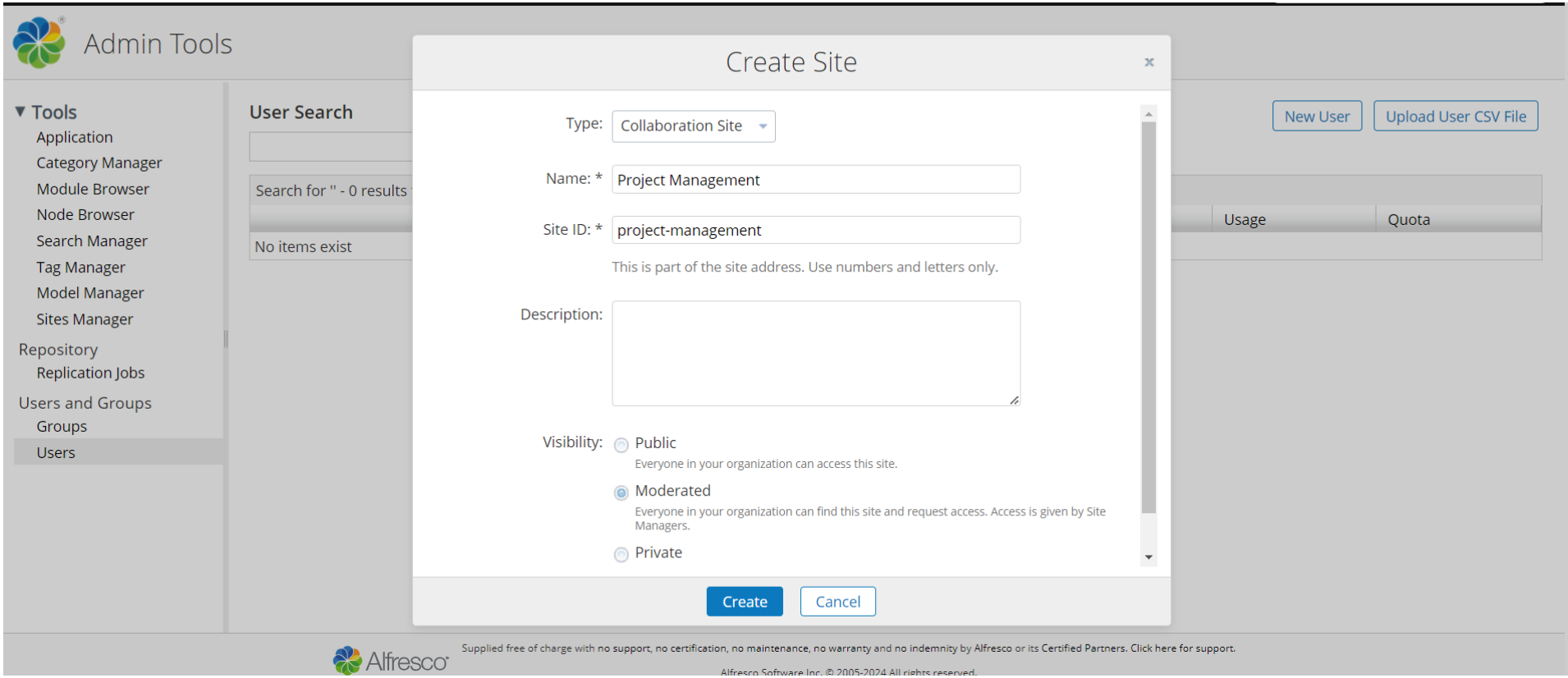
Properties

- Properties are metadata which describes the content.
- For example, Author is a property which specifies the person who wrote the content.

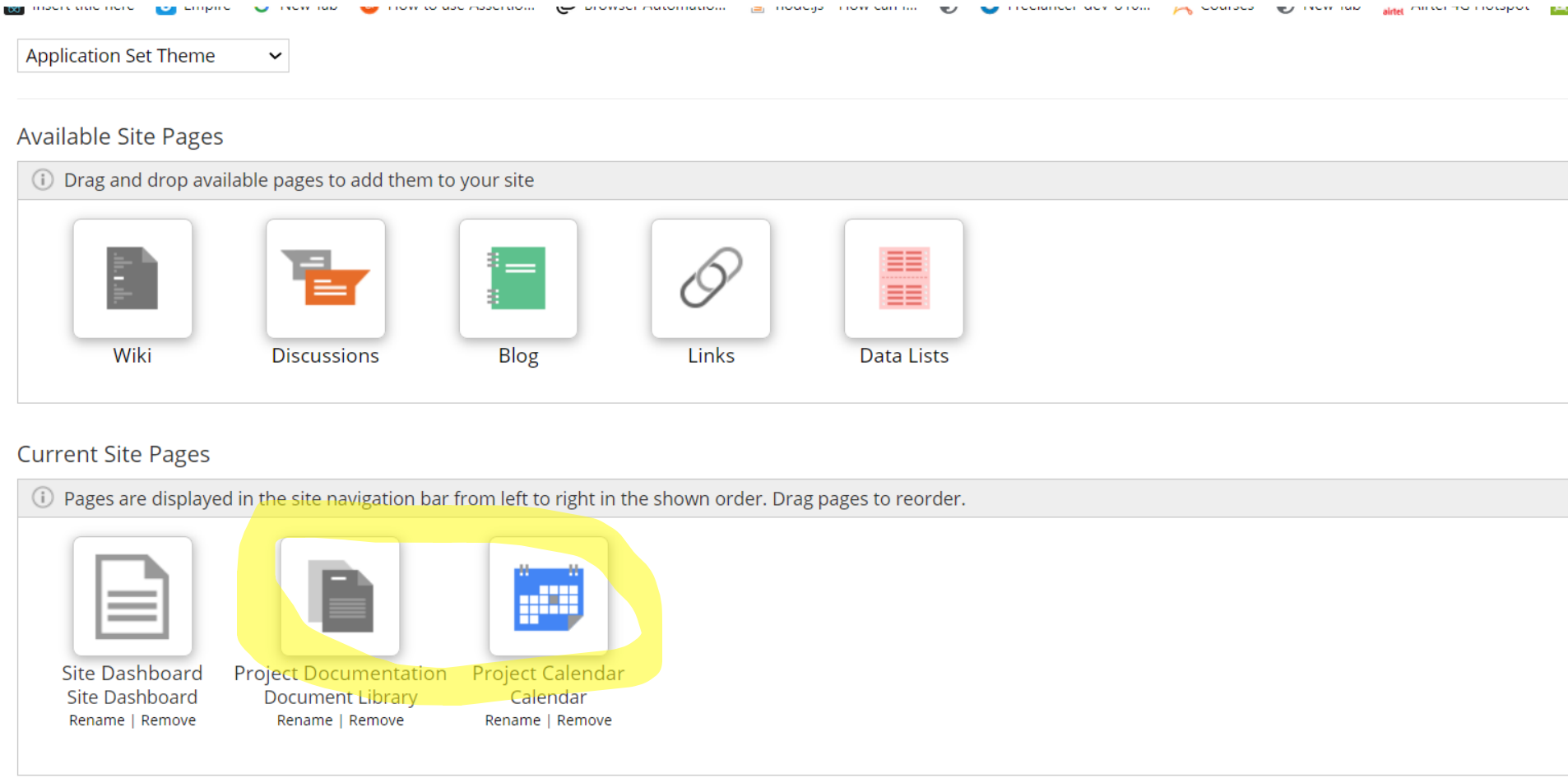
Constraints

- Constraints control the input property values.
- For example, you can specify that the author's name must not be more than 40 characters in length.

Create Site


Admin Tools
Tools
Application
Category Manager
Module Browser
Node Browser
Search Manager
Tag Manager
Model Manager
Sites Manager
Repository
Replication Jobs
Users and Groups
Groups
Users
User Search
Search for '' - 0 results
No items exist
New User
Upload User CSV File
Usage
Quota
Create Site
Type:
Collaboration Site
Name: *
Project Management
Site ID: *
project-management
This is part of the site address. Use numbers and letters only.
Description:
Visibility:
Public
Everyone in your organization can access this site.
Moderated
Everyone in your organization can find this site and request access. Access is given by Site Managers.
Private
Create
Cancel
Alfresco
Supplied free of charge with no support, no certification, no maintenance, no warranty and no indemnity by Alfresco or its Certified Partners. Click here for support.
Alfresco Software Inc. © 2005-2024 All rights reserved.

Customize Site



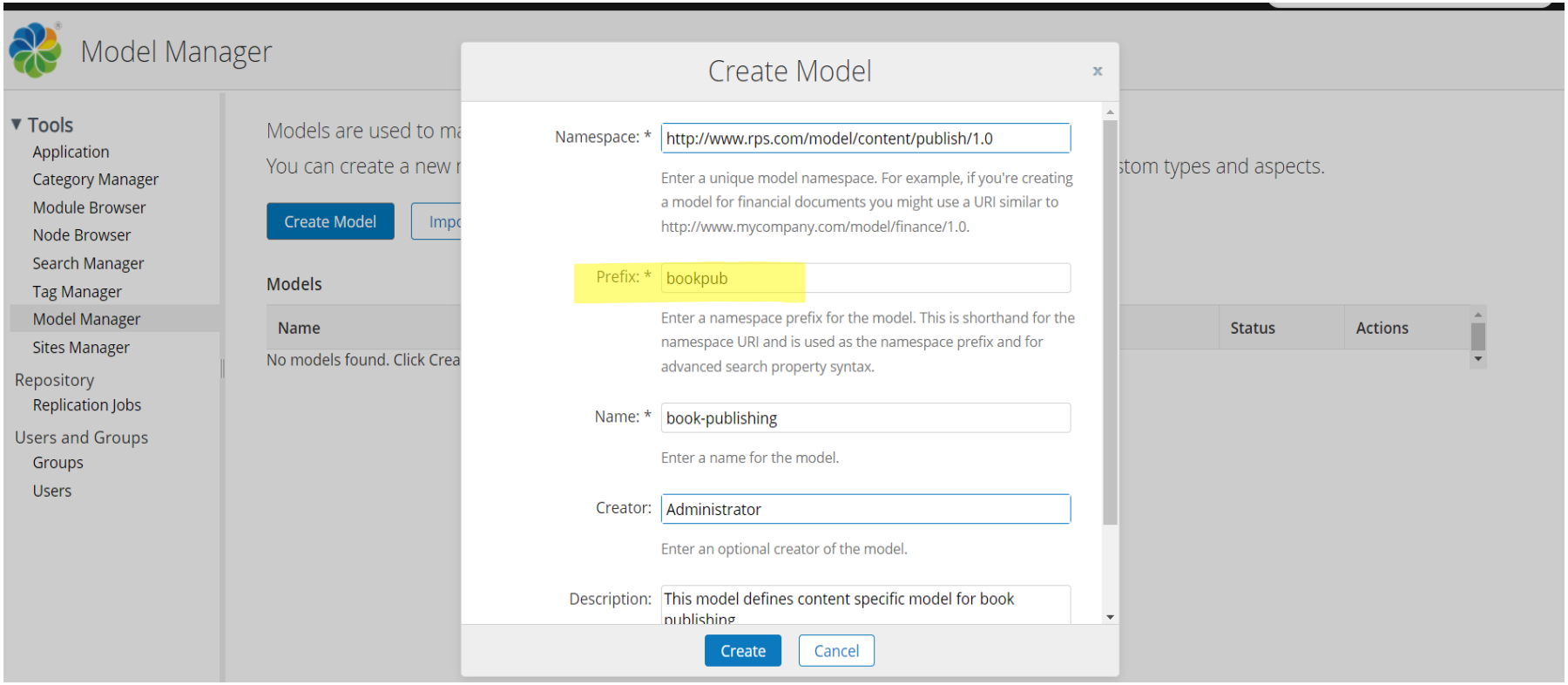
Create Model

Namespace bookpub

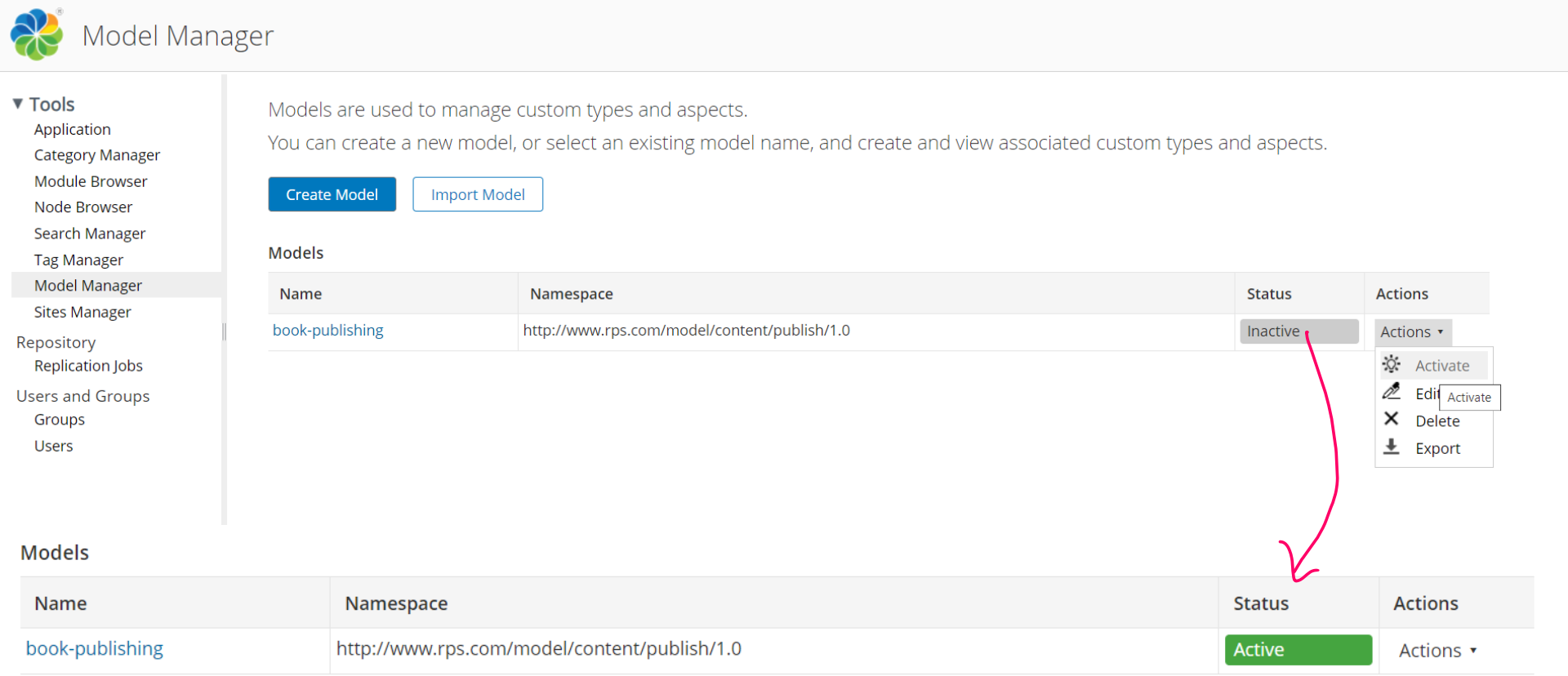
Types and Aspects

Name	Class (Type, Aspect)	Extends	Description	Notes	Advanced Search?
bookFolder	Type	cm:folder	Folder that contains all media related to a book	This type should be applied to new ISBN folders	YES
chapterFolder	Type	cm:folder	Contains all media related to one chapter of a book	This type should be applied to new chapter folders	
mediaFile	Type	cm:content	A generic base type for book related files, such as a chapter, artwork, or the complete book.	This type should not be applied directly to a file	
chapterFile	Type	mediaFile	A file with this type contains a chapter of a book.	This type should be applied to all files containing chapter content, such as PDFs, XML, Word	
bookFile	Type	mediaFile	A file with this type contains a complete book.	This type should be applied to files containing the completed book, such as PDFs, XML, Word files with complete books	YES
supplementaryFile	Type	mediaFile	A file with this type is a supplementary file to a book, such as the index.	This type should be applied to files containing supplementary content to the book, such as index and table of contents.	
artworkFile	Type	mediaFile	A file with this type contains artwork for a chapter, such as a diagram.	This type should be applied to files containing artwork, such as a diagram.	
bookInfo	Aspect		Contains the main metadata/properties for a book, such as ISBN.	Applied to all files and folders related to a book	
chapterInfo	Aspect		Metadata/properties for a specific chapter of the book	Applied to chapter folders and chapter files.	
webPublishingInfo	Aspect		Information about when the book was published on the web	Applied to book folders when a book is published on the Web	

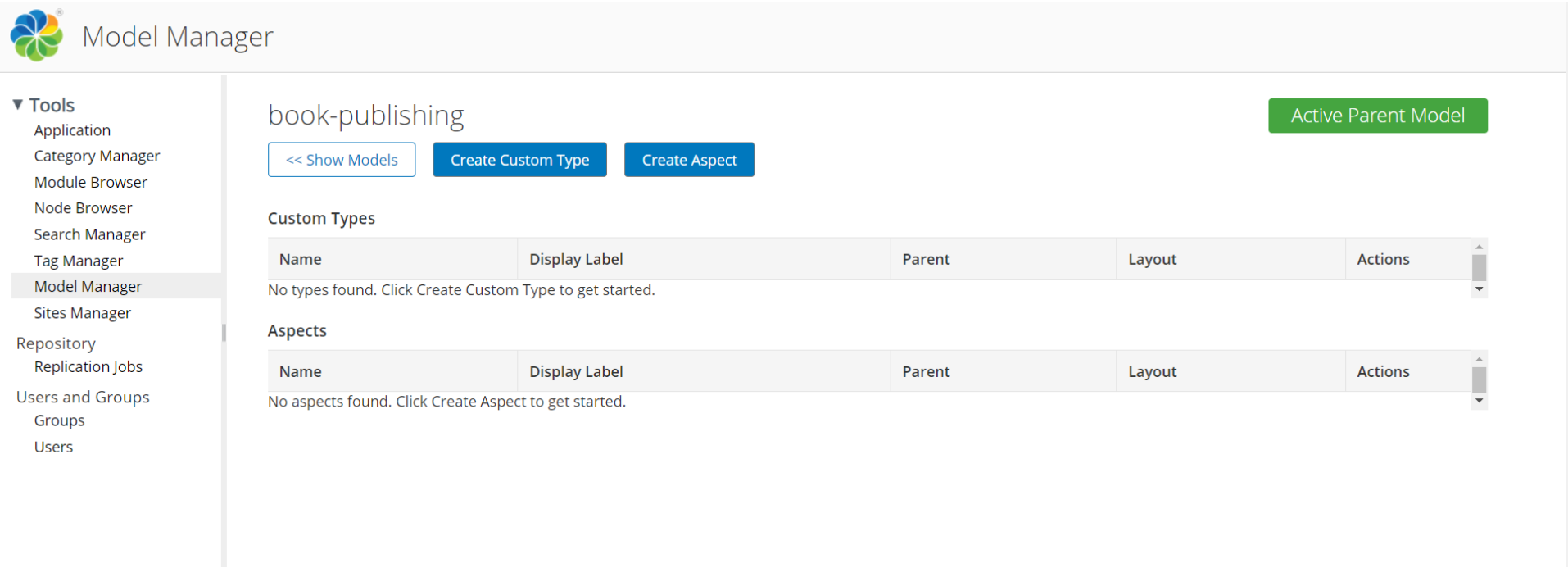
Create Model



Create Model



Create Model



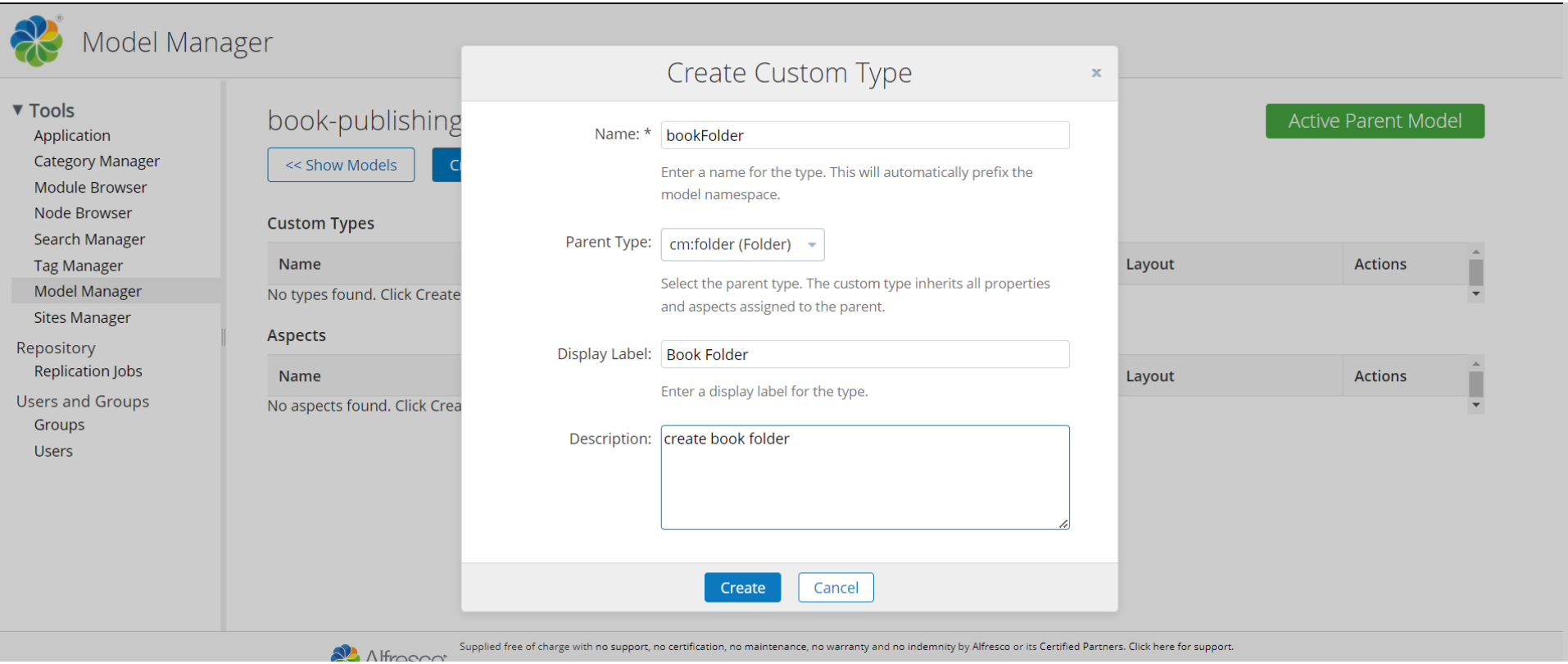
Create Model

Namespace	bookpub

Types and Aspects

Name	Class (Type, Aspect)	Extends	Description	Notes	Advanced Search?
bookFolder	Type	cm:folder	Folder that contains all media related to a book	This type should be applied to new ISBN folders	YES
chapterFolder	Type	cm:folder	Contains all media related to one chapter of a book	This type should be applied to new chapter folders	
mediaFile	Type	cm:content	A generic base type for book related files, such as a chapter, artwork, or the complete book.	This type should not be applied directly to a file	
chapterFile	Type	mediaFile	A file with this type contains a chapter of a book.	This type should be applied to all files containing chapter content, such as PDFs, XML, Word	
bookFile	Type	mediaFile	A file with this type contains a complete book.	This type should be applied to files containing the completed book, such as PDFs, XML, Word files with complete books	YES
supplementaryFile	Type	mediaFile	A file with this type is a supplementary file to a book, such as the index.	This type should be applied to files containing supplementary content to the book, such as index and table of contents.	
artworkFile	Type	mediaFile	A file with this type contains artwork for a chapter, such as a diagram.	This type should be applied to files containing artwork, such as a diagram.	
bookInfo	Aspect		Contains the main metadata/properties for a book, such as ISBN.	Applied to all files and folders related to a book	
chapterInfo	Aspect		Metadata/properties for a specific chapter of the book	Applied to chapter folders and chapter files.	
webPublishingInfo	Aspect		Information about when the book was published on the web	Applied to book folders when a book is published on the Web	

Create Custom Type



Create Custom Type

book-publishing

Active Parent Model

<< Show Models | Create Custom Type | Create Aspect

Custom Types

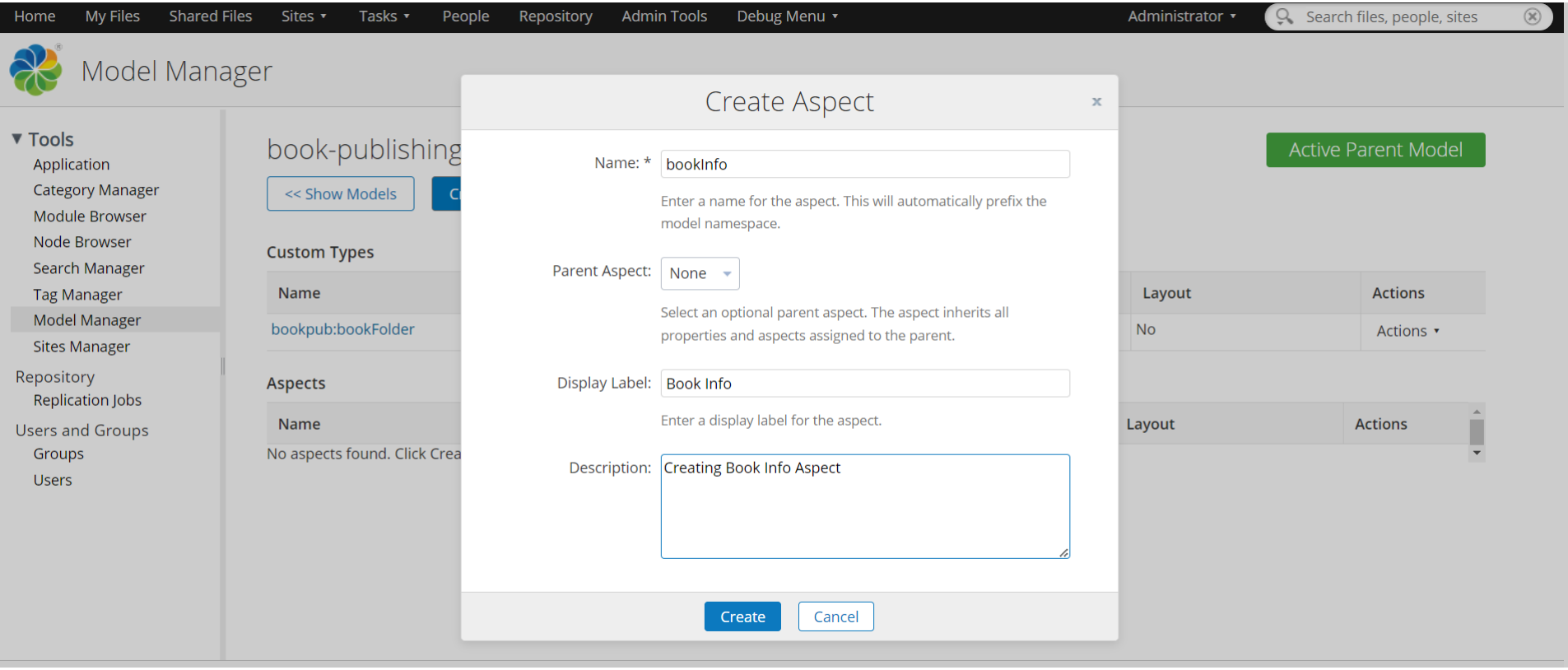
Name	Display Label	Parent	Layout	Actions
bookpub:bookFolder	Book Folder	cm:folder	No	Actions ▾

Aspects

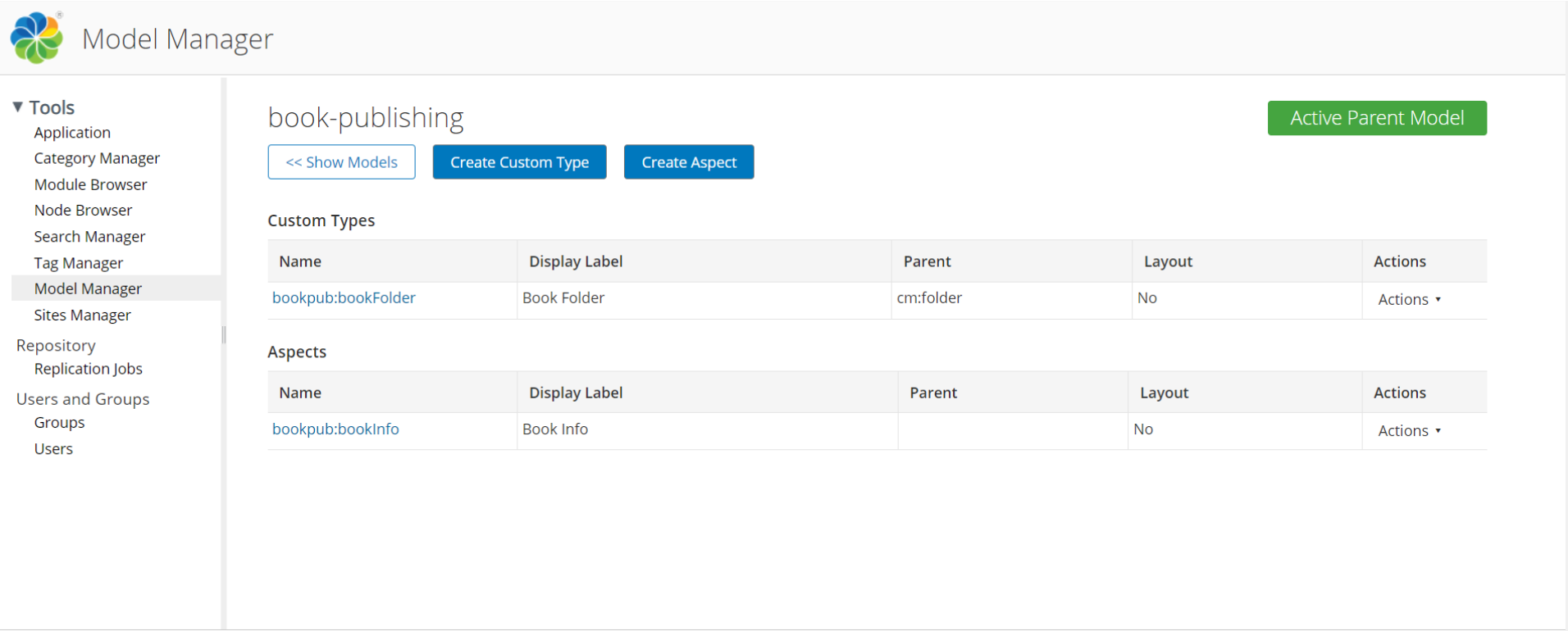
Name	Display Label	Parent	Layout	Actions

No aspects found. Click Create Aspect to get started.

Create Custom Type



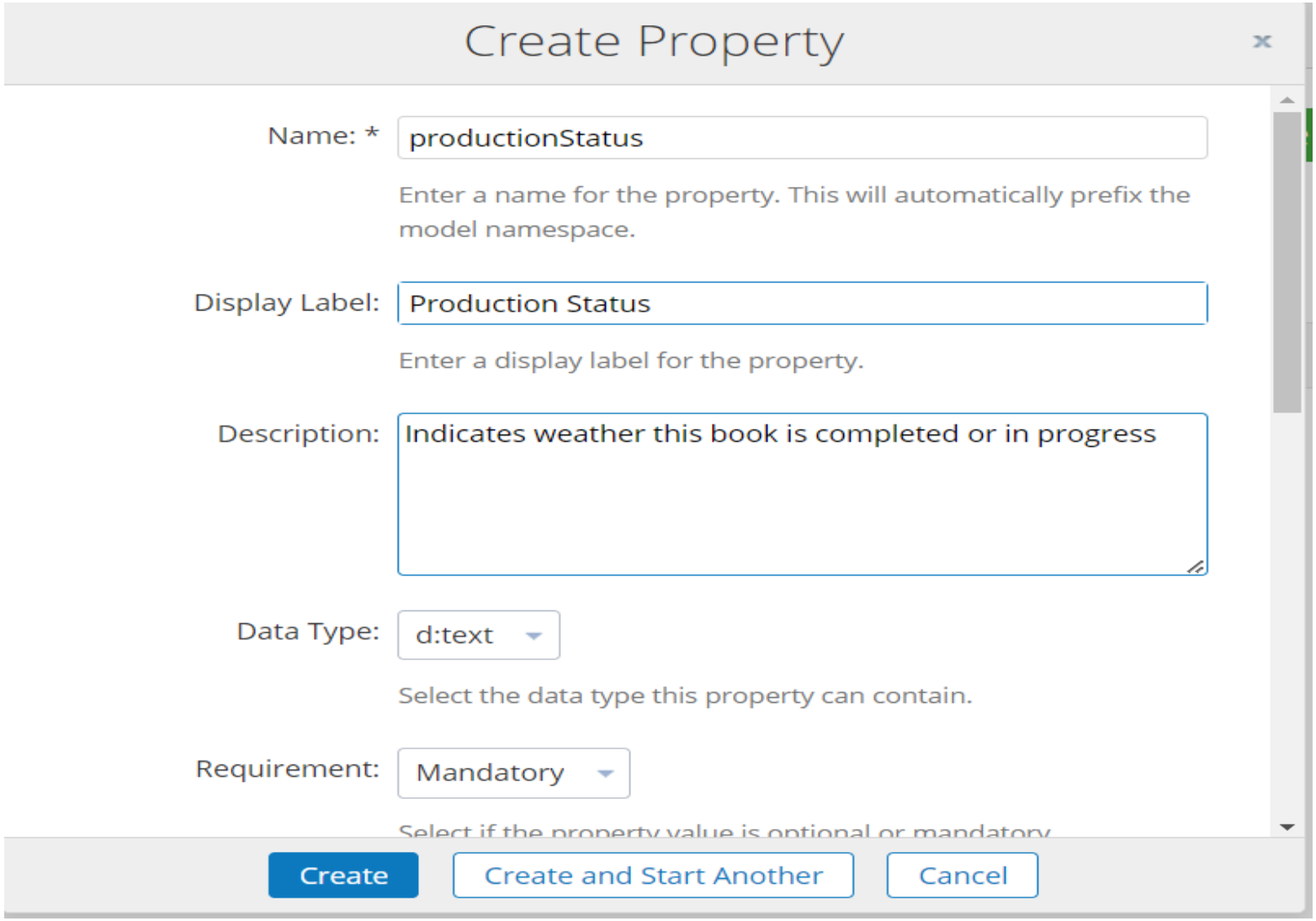
Create Custom Type



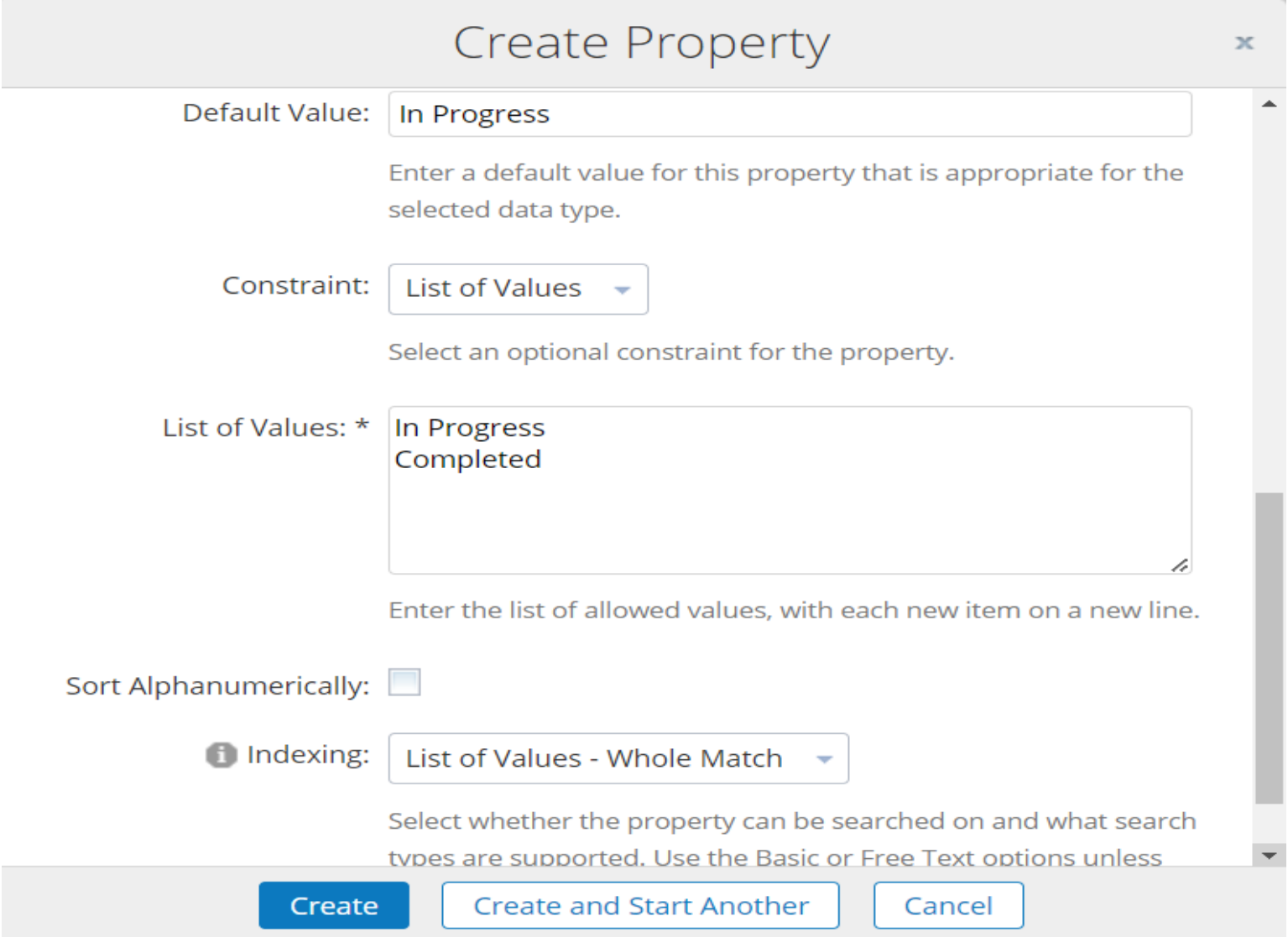
Create Properties

Name	Title	Description	Data type	Constraint	Mandatory?	Default Value	Multi?	Example	Included In Type/Aspect	Notes
productionStatus	Production Status	Indicates weather this book is completed or in progress	d:text	In Progress, Completed	Y	In Progress	N	Completed	bookFolder	
nrOfChapters	Number of Chapters	The total number of chapters that the book has	d:int	MINMAX(1,200)	N		N	15	bookInfo	
nrOfPages	Number of Pages	The total number of pages	d:int	MINMAX(10,2000)	N		N	502	bookInfo	
ISBN	ISBN	The main ISBN-13 number for the book	d:text	REGEX(^(97(8|9))\d{9}(\d|X)$)	Y		N	9781849513340'	bookInfo	
bookSubjectName	Book Subject Name	The subject of the book	d:text	"Business and Management","Communication, Journalism, Media and Culture","Criminology","Economics and Finance","Education","Environment and Sustainability","Geography","Information Technology"	Y		N	Business and Management	bookInfo	
bookTitle	Book Title	The title of the book	d:text		Y		N	Alfresco CMIS	bookInfo	
bookAuthors	Authors	The authors of this book	d:text		Y		Y	Martin Bergljung	bookInfo	
chapterAuthor	Author	The author of this chapter	d:text		Y		N	Martin Bergljung	chapterInfo	
chapterNumber	Chapter Number	The chapter number/order in the book	d:int	MINMAX(1,200)	Y		N	3	chapterInfo	
chapterTitle	Chapter Title	The title of the book chapter	d:text		Y		N		chapterInfo	
webPublishedDate	Web Published Date	The date the book was last published on the web electronically	d:datetime		N		N		webPublishingInfo	
webPublishedVersion	Web Published Version	Last published version to the web	d:text		N		N		webPublishingInfo	

Create Properties



Create Properties



Create Properties

Properties

Name	Display Label	Data Type	Requirement	Default Value	Multivalued	Actions
bookpub:productionStatus	Production Status	d:text	Optional	In Progress	No	Actions ▾

Create Aspect Properties

Name	Title	Description	Data type	Constraint	Mandatory?	Default Value	Multi?	Example	Included In Type/Aspect	Notes
productionStatus	Production Status	Indicates weather this book is completed or in progress	d:text	In Progress, Completed	Y	In Progress	N	Completed	bookFolder	
nrOfChapters	Number of Chapters	The total number of chapters that the book has	d:int	MINMAX(1,200)	N		N	15	bookInfo	
nrOfPages	Number of Pages	The total number of pages	d:int	MINMAX(10,2000)	N		N	502	bookInfo	
ISBN	ISBN	The main ISBN-13 number for the book	d:text	REGEX(^(97(8|9))\d{9}(\d|X)$)	Y		N	9781849513340'	bookInfo	
bookSubjectName	Book Subject Name	The subject of the book	d:text	"Business and Management","Communication, Journalism, Media and Culture","Criminology","Economics and Finance","Education","Environment and Sustainability","Geography","Information Technology"	Y		N	Business and Management	bookInfo	
bookTitle	Book Title	The title of the book	d:text		Y		N	Alfresco CMIS	bookInfo	
bookAuthors	Authors	The authors of this book	d:text		Y		Y	Martin Bergljung	bookInfo	
chapterAuthor	Author	The author of this chapter	d:text		Y		N	Martin Bergljung	chapterInfo	
chapterNumber	Chapter Number	The chapter number/order in the book	d:int	MINMAX(1,200)	Y		N	3	chapterInfo	
chapterTitle	Chapter Title	The title of the book chapter	d:text		Y		N		chapterInfo	
webPublishedDate	Web Published Date	The date the book was last published on the web electronically	d:datetime		N		N		webPublishingInfo	
webPublishedVersion	Web Published Version	Last published version to the web	d:text		N		N		webPublishingInfo	

Create Aspect Properties

book-publishing - bookInfo

Active Parent Model

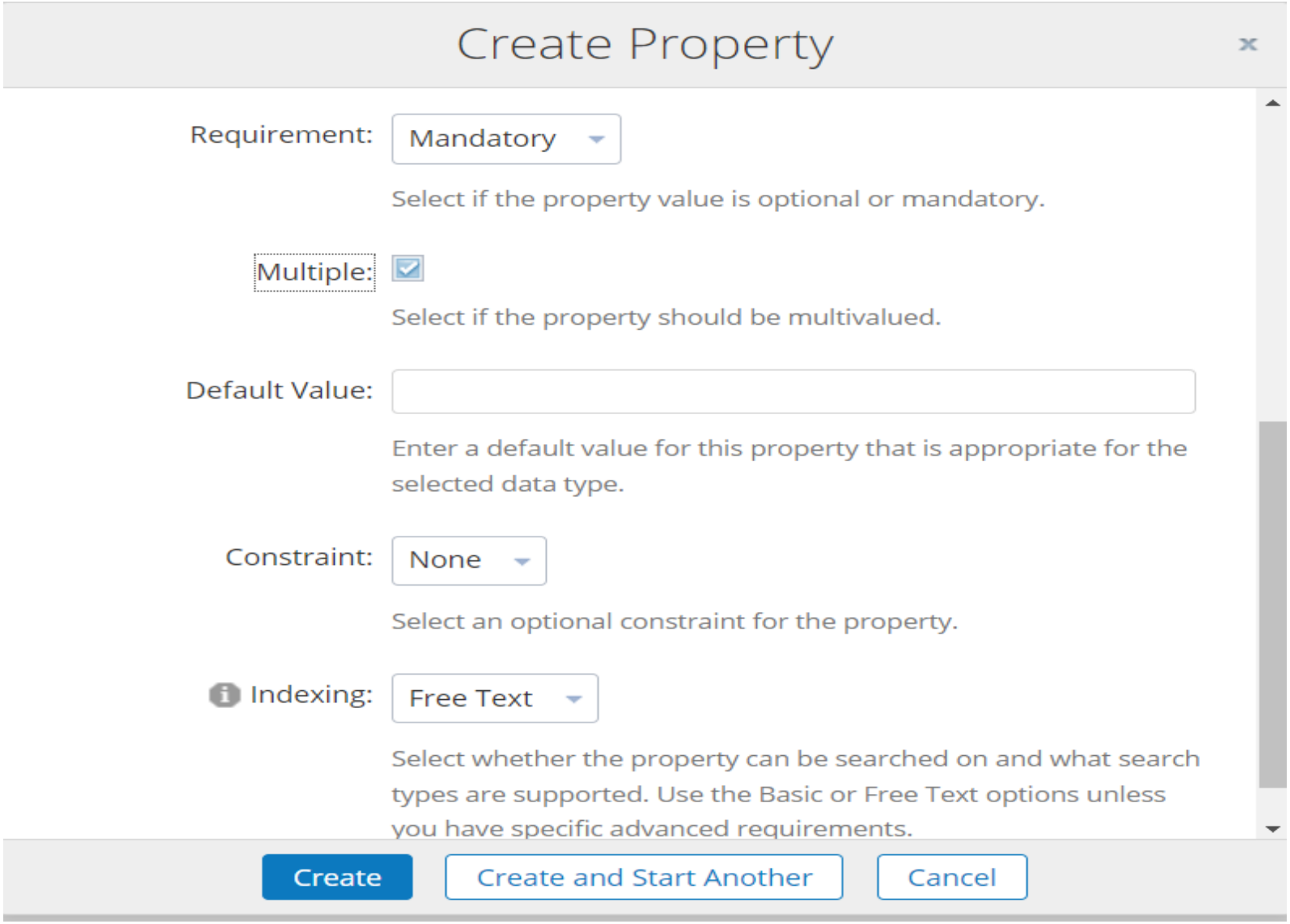
<< Show Types and Aspects

Create Property

Properties

Name	Display Label	Data Type	Requirement	Default Value	Multivalued	Actions
bookpub:bookTitle	Book Title	d:text	Optional		No	Actions ▾

Create Aspect Properties



Create Aspect Properties

book-publishing - bookInfo

Active Parent Model

<< Show Types and Aspects

Create Property

Properties

Name	Display Label	Data Type	Requirement	Default Value	Multivalued	Actions
bookpub:bookAuthors	Book Authors	d:text	Mandatory		Yes	Actions
bookpub:bookTitle	Book Title	d:text	Optional		No	Actions

Import the model

Models are used to manage custom types and aspects.

You can create a new model, or select an existing model name, and create and view associated custom types and aspects.

Create Model | Import Model

Models

Name	Namespace	Status	Actions
book-publishing	http://www.rps.com/model/content/publish/1.0	Active	Actions

Deactivate

Export

Deactivate

Models are used to manage custom types and aspects.

You can create a new model, or select an existing model name, and create and view associated custom types and aspects.

Create Model | Import Model

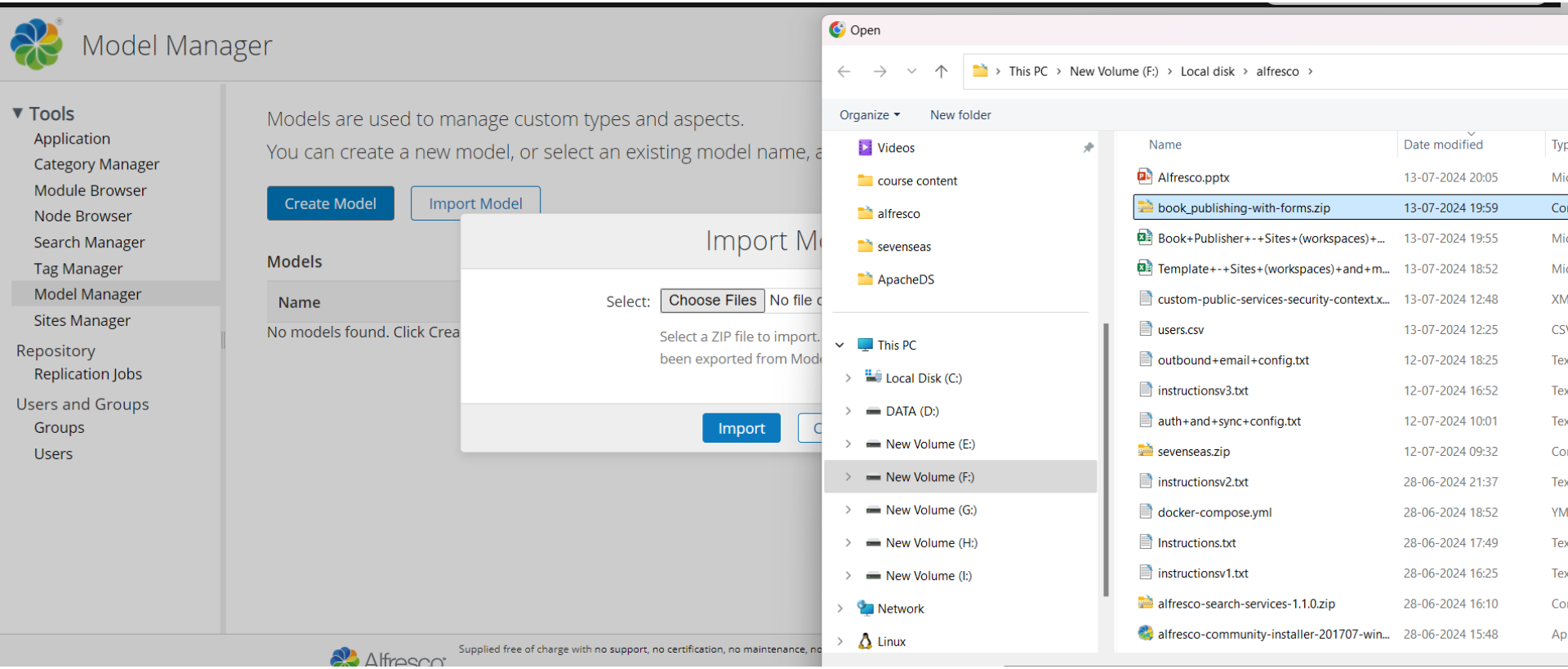
Models

Delete Model

Are you sure you want to delete model "book-publishing"? All custom types, aspects and properties in the model will also be deleted.

Delete | Cancel

Import the model



Import the model

Models are used to manage custom types and aspects.

You can create a new model, or select an existing model name, and create and view associated custom types and aspects.

Create Model | Import Model

Models

Name	Namespace	Status	Actions
book_publishing	http://www.acme.com/model/content/publishing/1.0	Active	Actions ▾

Import the model

book_publishing

Inactive Parent Model

<< Show Models | Create Custom Type | Create Aspect

Custom Types

Name	Display Label	Parent	Layout	Actions
bookpub:artworkFile	Artwork File	bookpub:mediaFile	Yes	Actions
bookpub:bookFile	Book File	bookpub:mediaFile	Yes	Actions
bookpub:bookFolder	Book Folder	cm:folder	Yes	Actions
bookpub:chapterFile	Chapter File	bookpub:mediaFile	Yes	Actions
bookpub:chapterFolder	Chapter Folder	cm:folder	Yes	Actions
bookpub:mediaFile	Media File	cm:content	No	Actions
bookpub:supplementaryFile	Supplementary File	bookpub:mediaFile	Yes	Actions

Aspects

Name	Display Label	Parent	Layout	Actions
bookpub:bookInfo	Book Info		Yes	Actions
bookpub:chapterInfo	Chapter Info		Yes	Actions
bookpub:webPublishingInfo	Web Publishing Info		Yes	Actions

Layout the UI for custom content Model

book_publishing

Active Parent Model

<< Show Models | Create Custom Type | Create Aspect

Custom Types

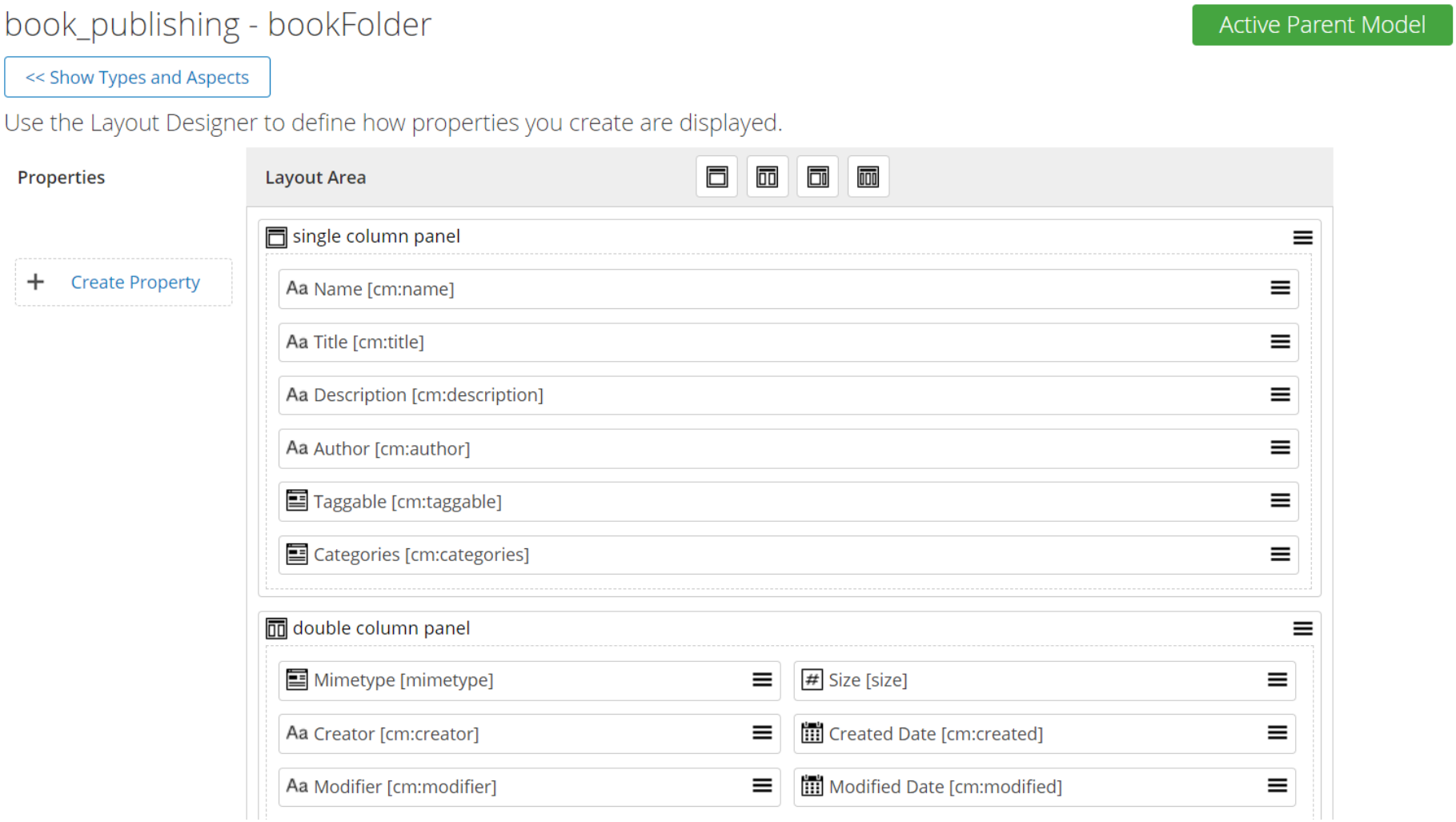
Name	Display Label	Parent	Layout	Actions
bookpub:artworkFile	Artwork File	bookpub:mediaFile	Yes	Actions
bookpub:bookFile	Book File	bookpub:mediaFile	Yes	Actions
bookpub:bookFolder	Book Folder	cm:folder	Yes	Actions
bookpub:chapterFile	Chapter File	bookpub:mediaFile	Yes	
bookpub:chapterFolder	Chapter Folder	cm:folder	Yes	
bookpub:mediaFile	Media File	cm:content	No	
bookpub:supplementaryFile	Supplementary File	bookpub:mediaFile	Yes	Actions

- Layout Designer
- Edit
- Delete
- Find Where Used

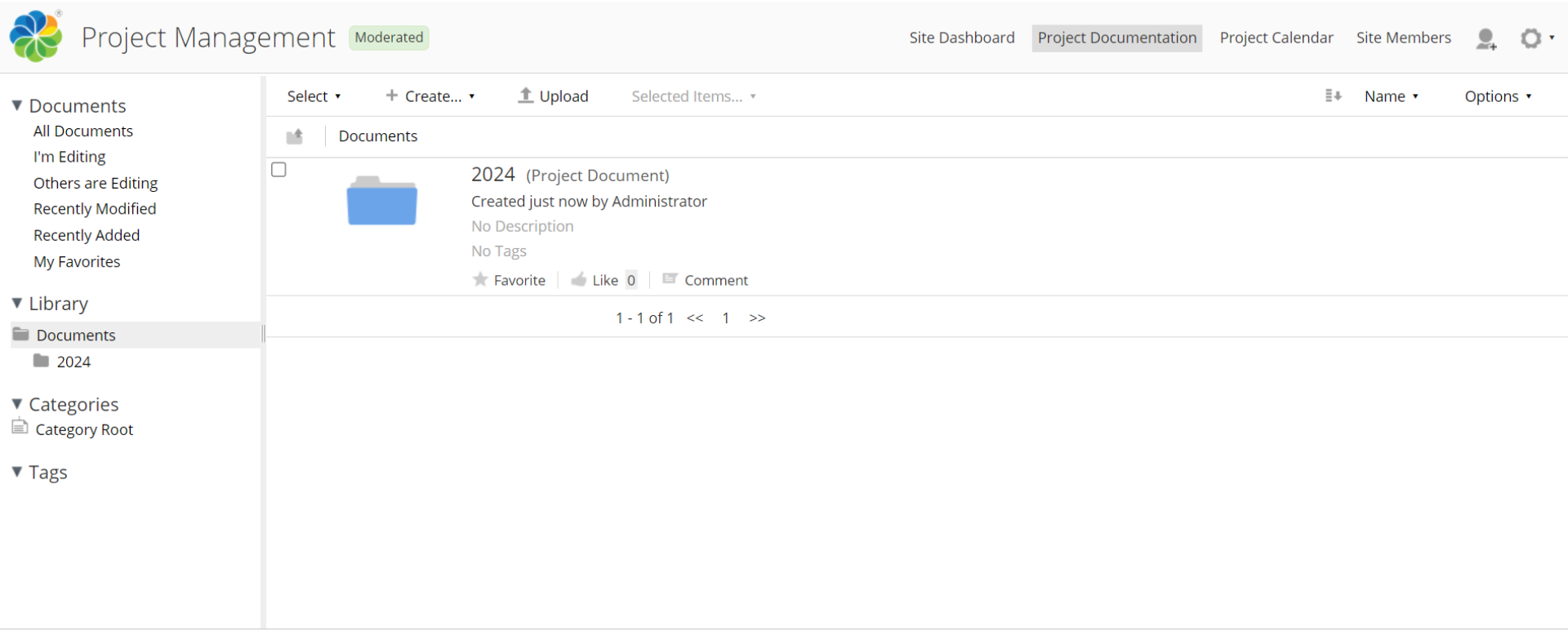
Aspects

Name	Display Label	Parent	Layout	Actions
bookpub:bookInfo	Book Info		Yes	Actions

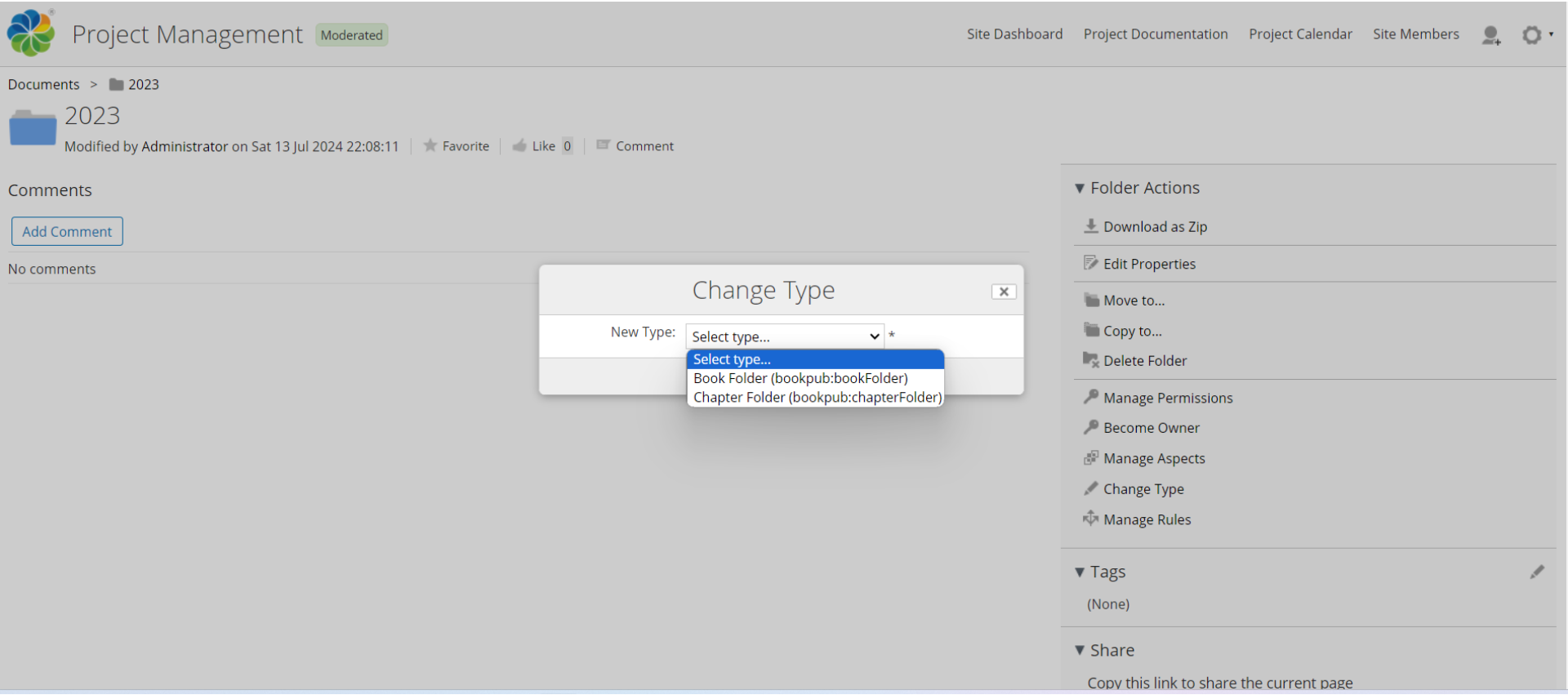
Layout the UI for custom content Model



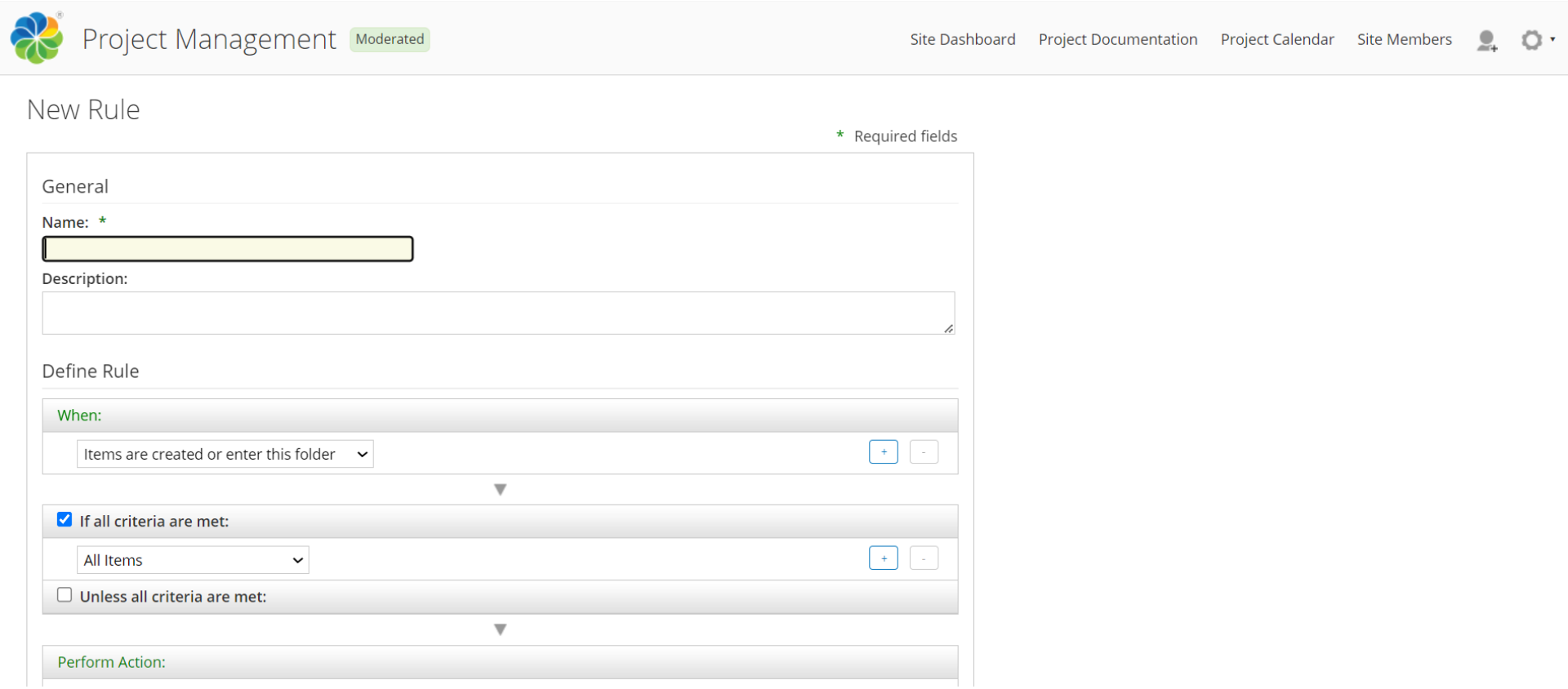
Add Document



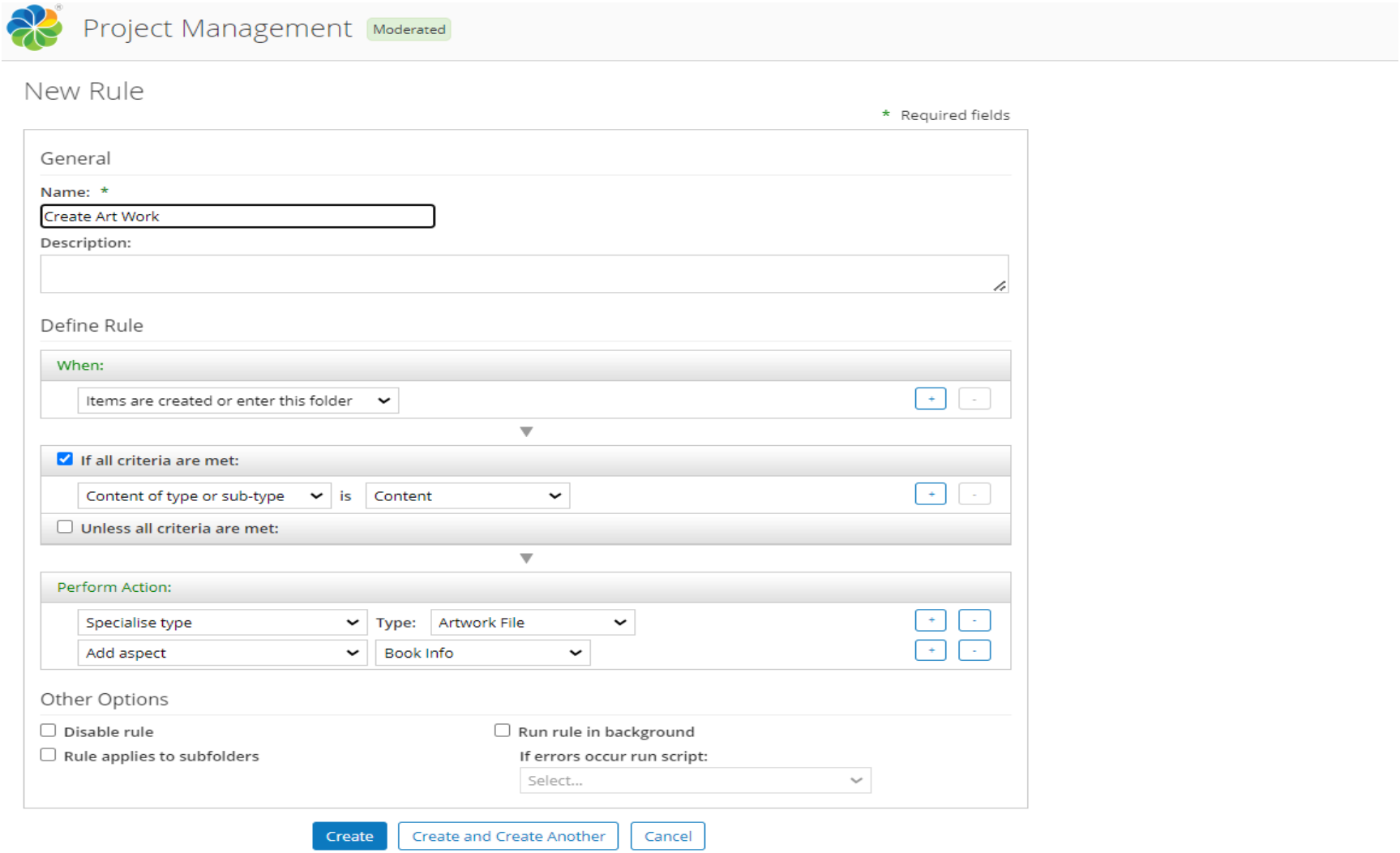
Add Document



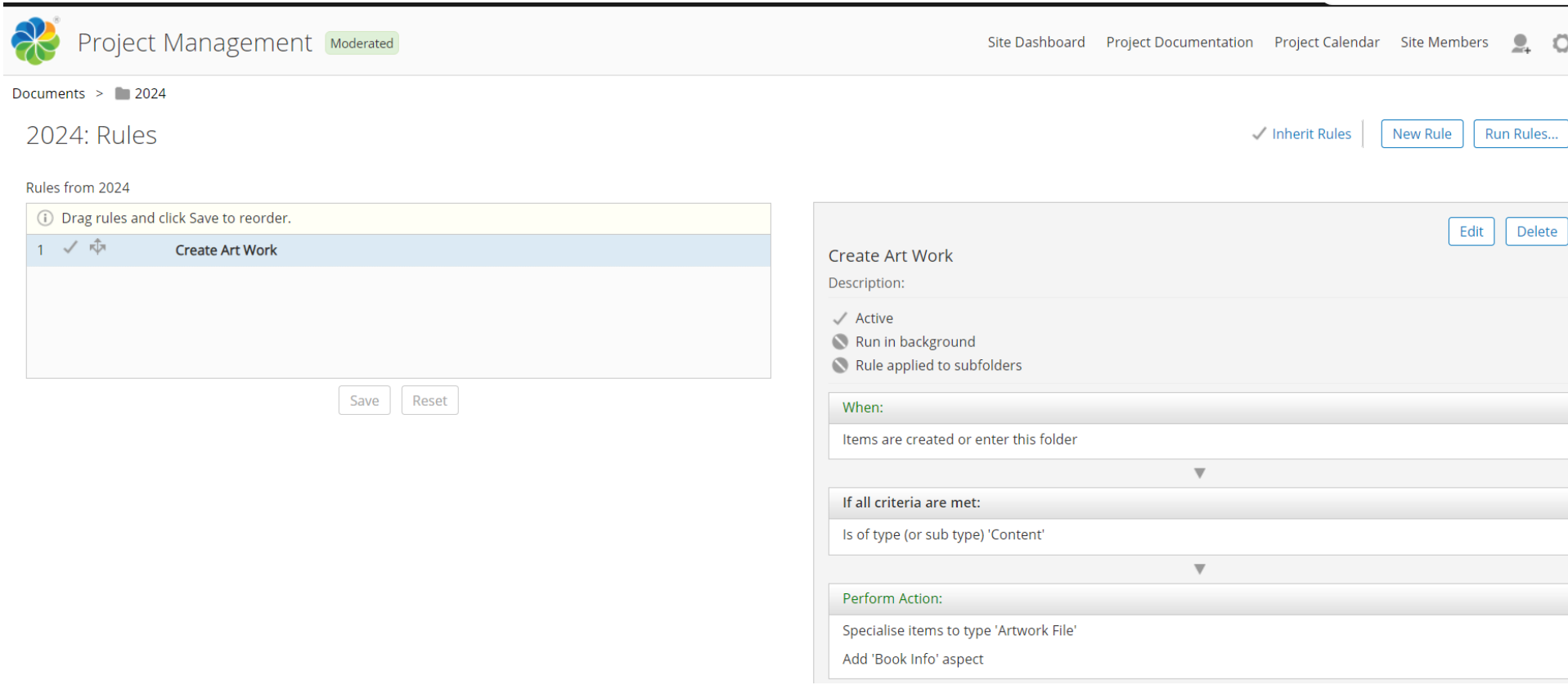
Add Document



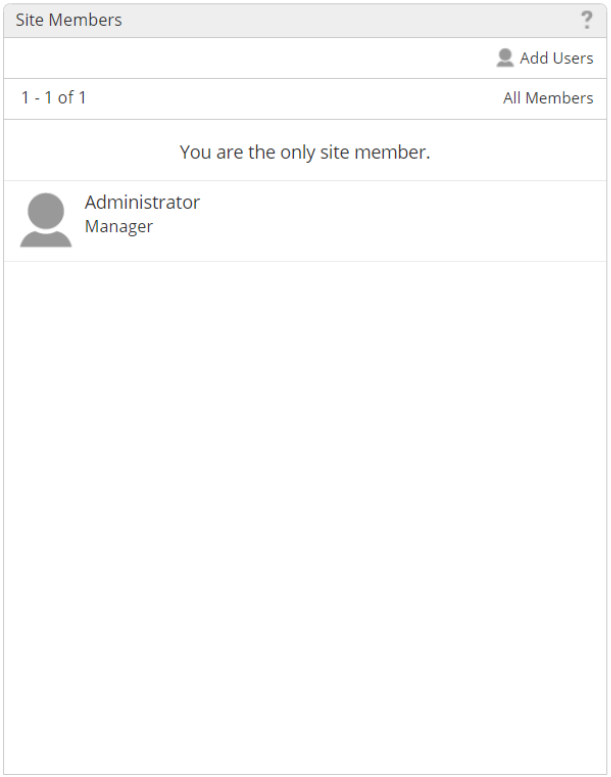
Add Document

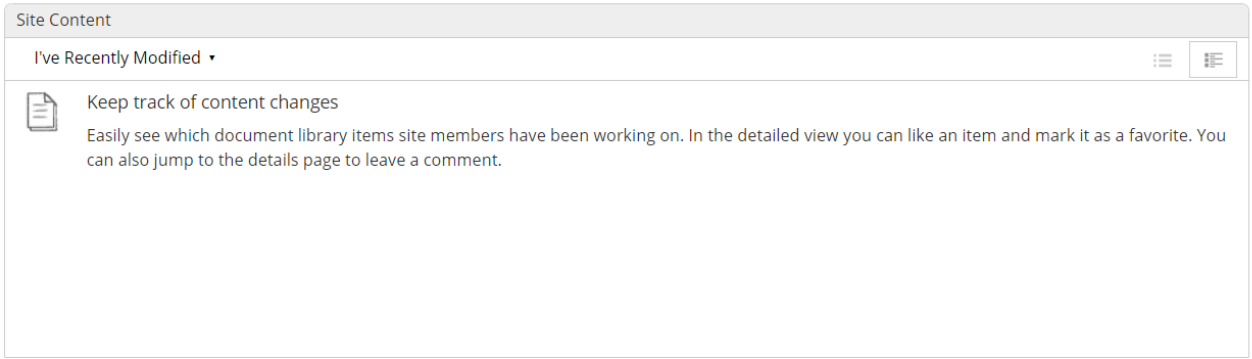


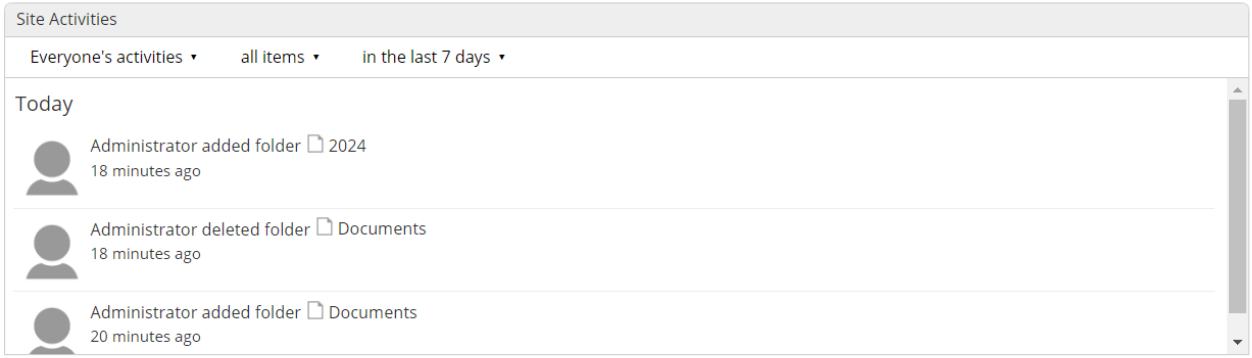
Add Document



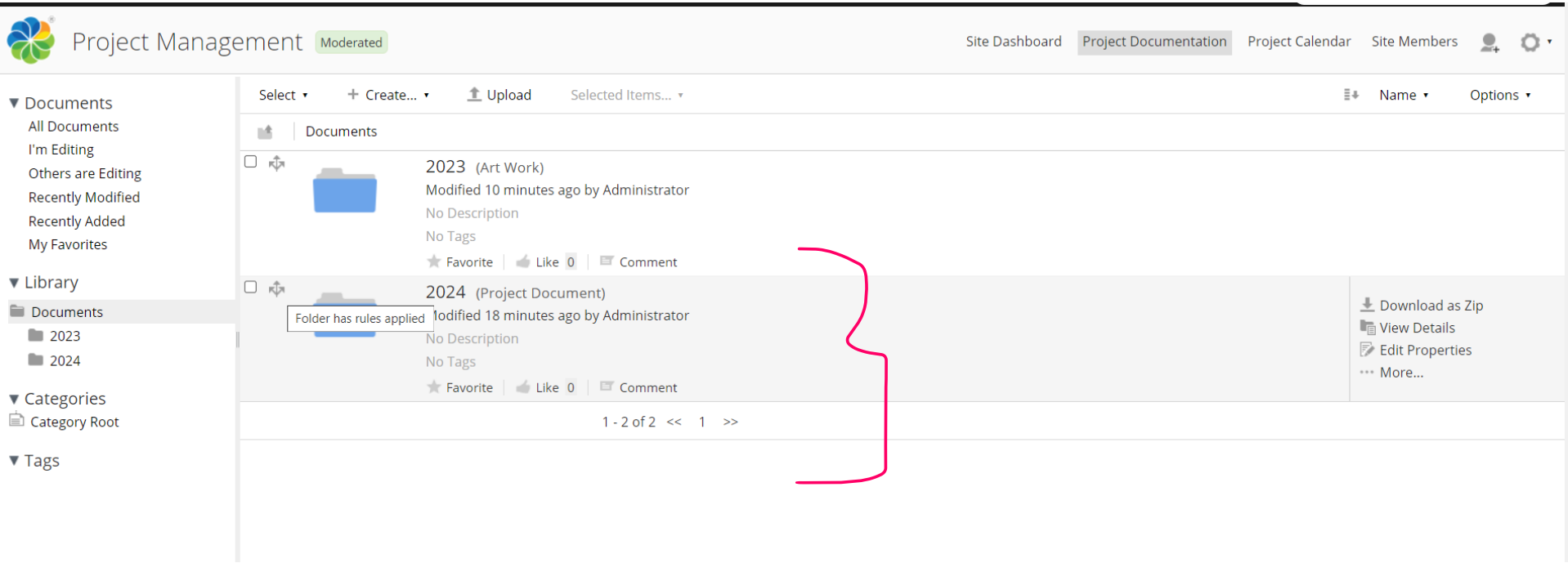
Add Document



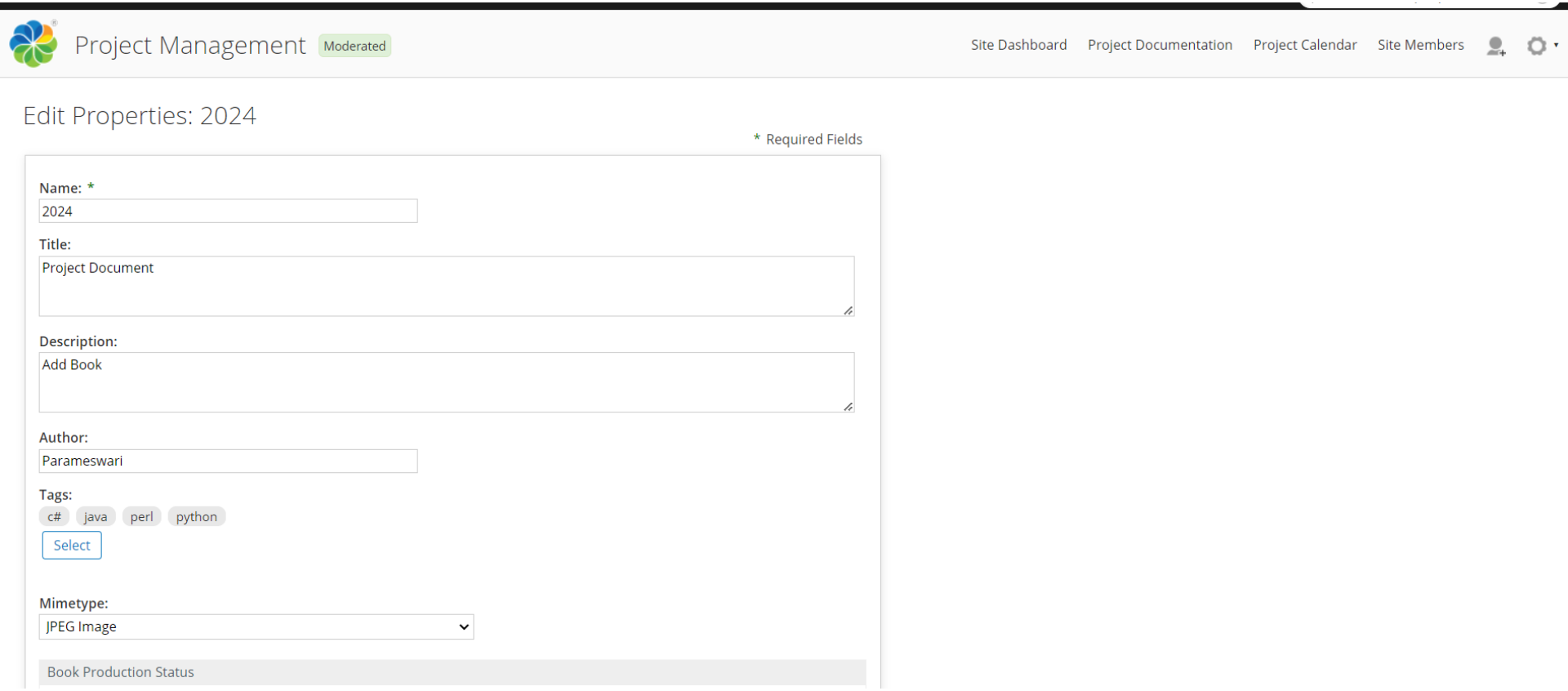




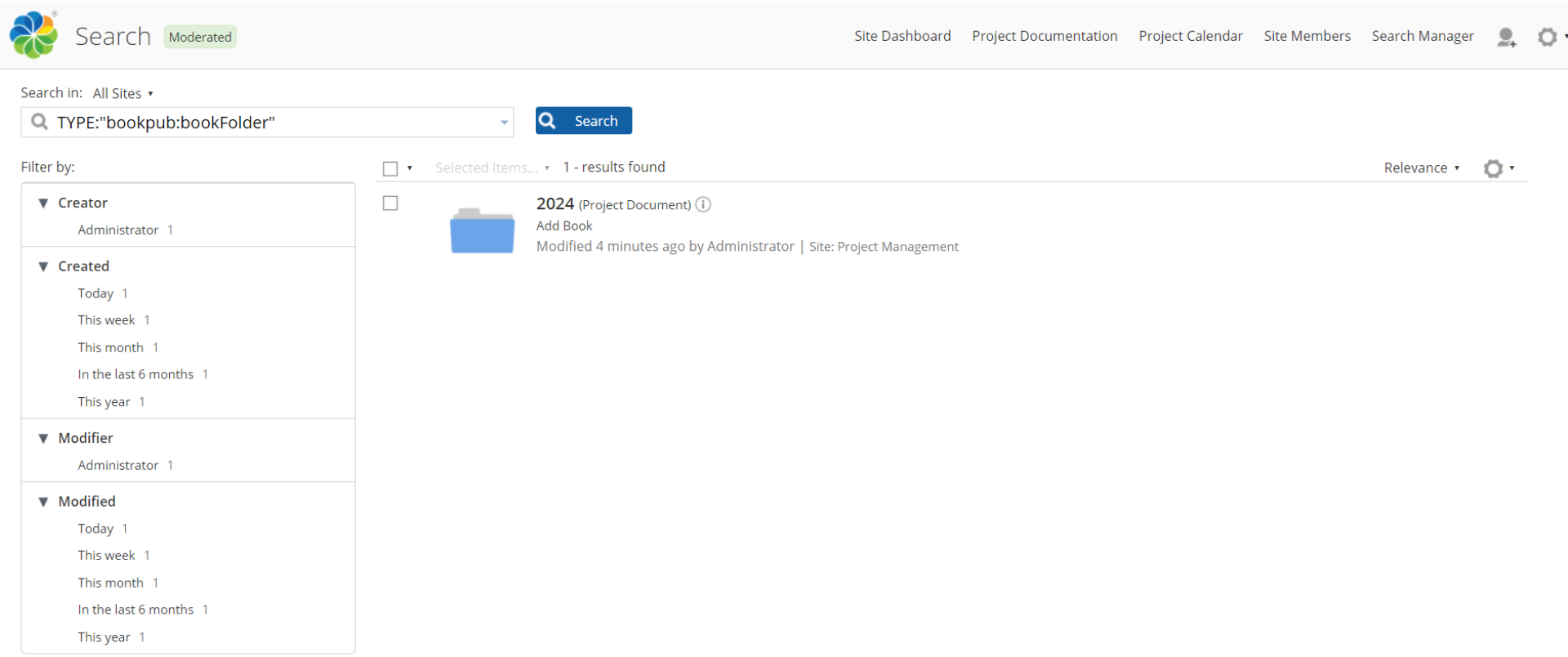
Add Document



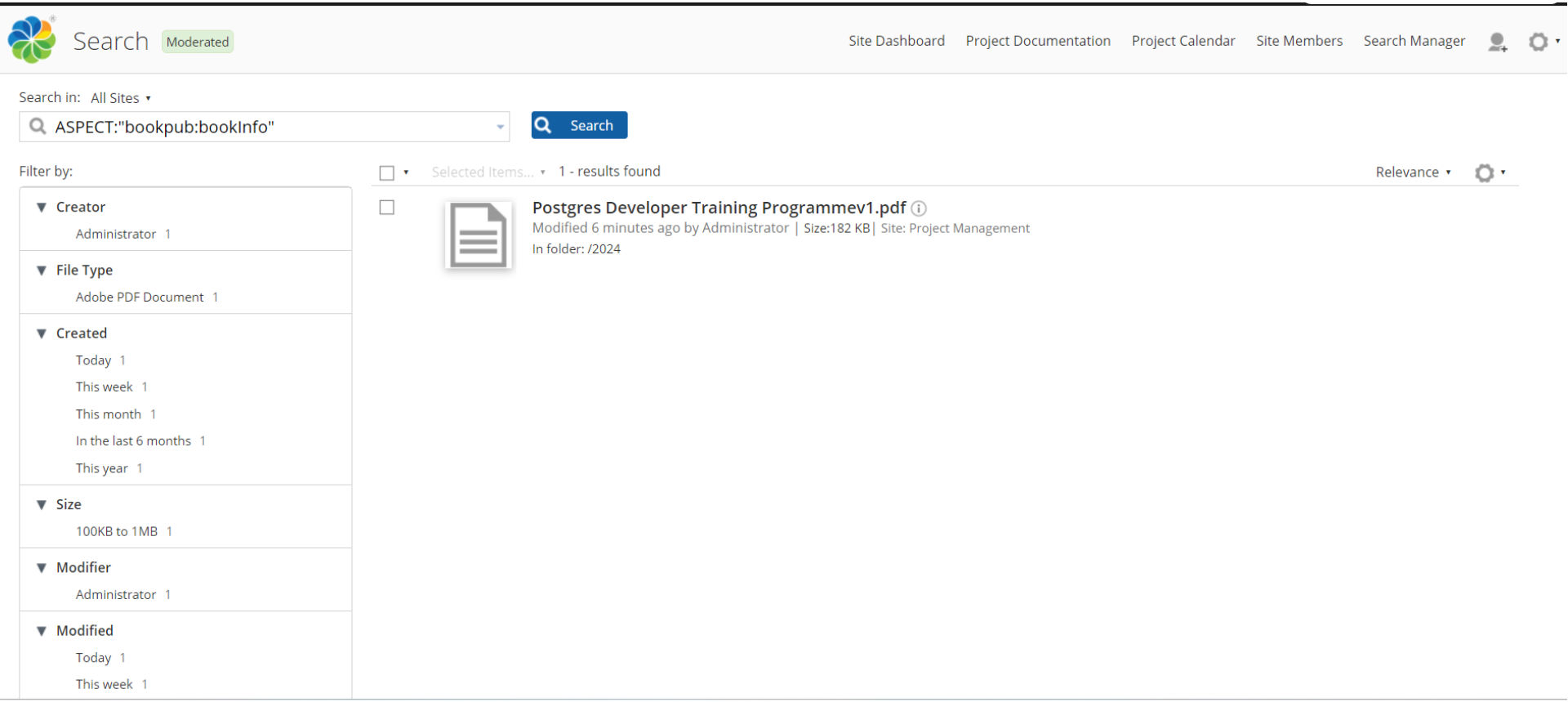
Add Document



Search Document



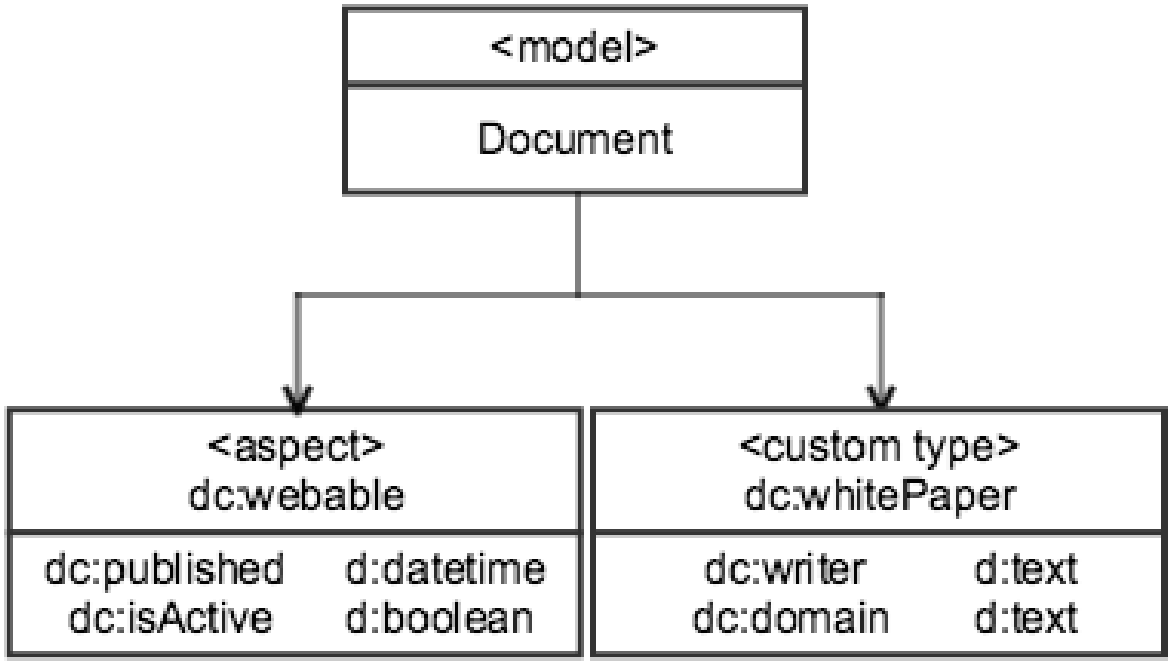
Search Document



Alfresco Models

DemoCo's model

dcDocument.xml
uri=http://www.democo.com/model/document/1.0
prefix="dc"



Alfresco Models

- Step 1: Navigate to Model Manager
  - Log into Alfresco Share as a user that is a member of the ALFRESCO_MODEL_ADMINISTRATORS permission group and click Admin Tools > Model Manager.

Alfresco Models

- Step 2: Create a new model
- Click Create Custom Model.
- Set the following properties:
  - Namespace: http://www.democo.com/model/document/1.0
  - Prefix: doc
  - Name: Document
- Click Create.

Alfresco Models

- Step 3: Create a new type
- Click the content model, Document.
- Click Create Custom Type.
- Set at least the following properties:
  - Name: whitePaper
  - Display label: White Paper
- Click Create.

Alfresco Models

- Step 4: Create new custom type properties
- Under the Custom Types list, click doc:whitePaper.
- Click Create Property.
- Set at least the following properties:
  - Name: writer
  - Display label: Writer
  - Select d:text for the data type
  - Requirement: Mandatory

Alfresco Models

- Click Create and Start Another and set at least the following properties:
  - Name: domain
  - Display label: Domain
  - Select d:text for the data type
  - Select List of Values for the constraint and enter Engineering, Marketing, HR, Sales, Finance and Operations.
  - Select Sort Alphanumerically for the order of values.

Alfresco Models

- Step 5: Add new type properties to the Layout Designer
  - Navigate to the page displaying the custom types list.
  - Click Layout Designer from the Actions drop-down list for the type, dc:whitePaper.
  - Drag the double column panel layout from top onto the layout area.
  - To specify the panel label, click anywhere on the element's top panel displaying the label, double column panel.
  - In the Label field, enter Details.
  - Configure other optional fields, if required.
  - Drag the properties, writer and domain onto the Details element.
  - To configure the properties, click anywhere on the property.
  - On the Layout Designer, click Save.

Alfresco Models

- Step 6: Create new aspect
  - Navigate to the Model Manager page.
  - Click the model, Document.
  - Click Create Aspect.
  - Set at least the following properties:
    - Name: webable
    - Display label: Webable
  - Click Create.

Alfresco Models

- Step 7: Create new aspect properties
  - Under the Aspects list, click dc:webable.
  - Click Create Property.
  - Set at least the following properties:
    - Name: published
    - Display label: Published
    - Select d:datetime for the data type
  - Click Create and Start Another.
  - Set at least the following properties for the new property:
    - Name: isActive
    - Display Label: Is Active
    - Select d:boolean for the data type
    - Default value: false
- Click Create.

Alfresco Models

- Step 8: Add new aspect properties to the Layout Designer
  - Navigate to the page displaying the aspects list.
  - Click Layout Designer from the Actions drop-down list for the aspect dc:webable.
  - Drag the single column panel layout from top onto the layout area.
  - To specify the panel label, click anywhere on the element's top panel displaying the label, single column panel.
  - In the Label field, enter Publication details and configure the optional fields if required.
  - Drag the properties, published and isActive onto the Publication details element.
  - To configure the properties, click anywhere on the property.
  - On the Layout Designer, click Save.

Alfresco Models

- Step 9: Activate the model
  - Navigate to the Model Manager page.
  - Click the Actions drop-down list for the model, Document.
- Click Activate.

Alfresco Models

- Step 10: Apply custom type and aspect to a file in Alfresco Share
  - In your site, click Document Library to access the library.
  - Click on a file in the library to view it in the file preview screen.
  - To apply the type, perform the following steps:
  - Under Document Actions, click Change Type.
  - In the Change Type window, select whitePaper (dc:whitePaper) from the New Type drop-down list.
- Click OK.

Alfresco Models

- In the Writer field, enter User1.
- In the domain field, select Engineering.

- Click Save.

Alfresco Models

- To apply the aspect, perform the following steps:
- Under Document Actions, click Manage Aspects.
  - In the Available to Add list, click the + next to Webable.
  - Click the trashcan to remove any existing aspects from the Currently Selected list.
- Click Apply changes.
- Select Is Active.
- In the Published field, enter the date and time as 31/07/2020 and 10:00, respectively.
- Click Save.

Alfresco Workflow

- A workflow is a sequence of connected tasks applied to a document or other item of content.
- Each task can be performed by a person, a group, or automatically.
- A workflow is not for external integration nor collecting, disseminating, or coordinating information not directly tied to a document or other item of content in Alfresco Content Services.
- References in this overall documentation to terms like process, tasks, or other tooling are in the context of Alfresco Content Services and as such capabilities are considered workflow.

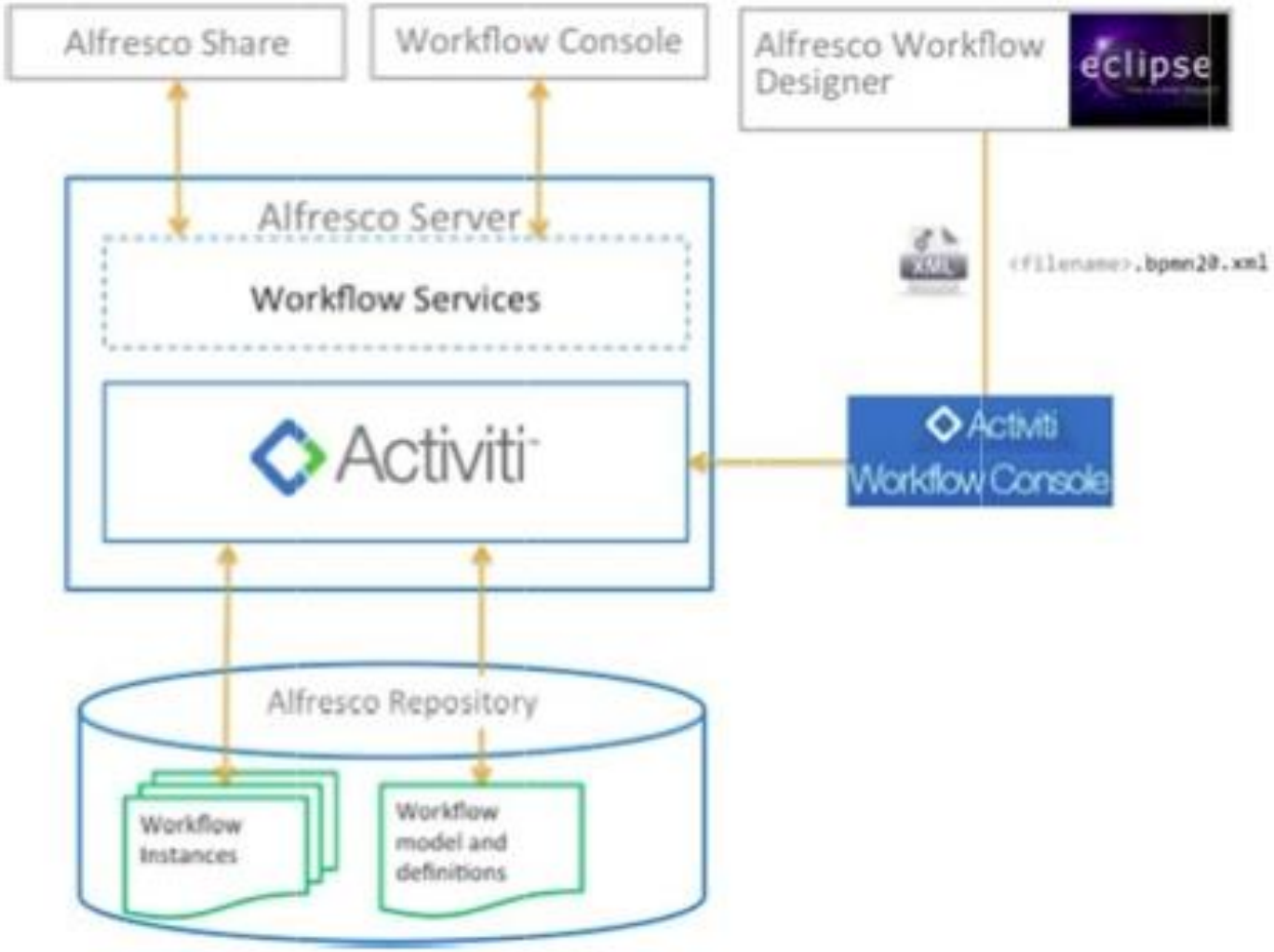
Alfresco Workflow

- For example, we might have a document that you needed reviewing and approving by several people.
  - The sequence of connected tasks would be:
  - Send an email to each reviewer asking them to review the document within a certain time
  - Each reviewer reviews the document
  - Each reviewer approves or rejects the document
  - If enough reviewers approve, the task is completed successfully.

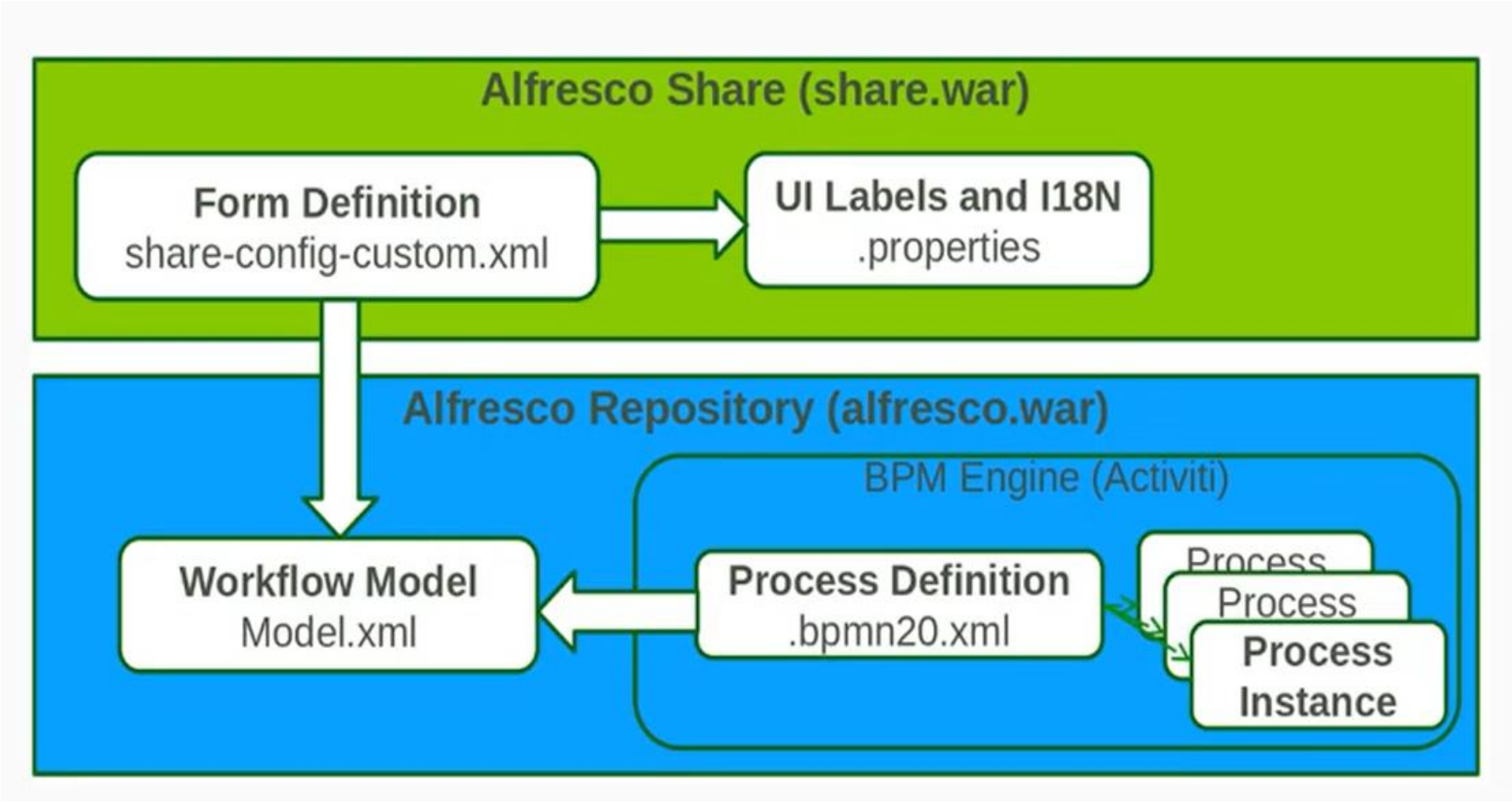
Alfresco Workflow Definitions

- Adhoc
  - Enables you to assign a task to a single user
- Group Review & Approve
  - Enables you to set up review and approval of content, assigning the workflow task to a single group.
- Parallel Review & Approve
  - Enables you to set up review and approval of content, assigning the workflow task to multiple users.
- Pooled Review & Approve
  - Enables you to set up review and approval of content, assigning the workflow task to multiple users.
  - One user can take ownership of the task at a time, completing it or returning it to the pool to be claimed by another user associated with the task.
- Review & Approve
  - Enables you to set up review and approval of content, assigning the workflow task to a single user.

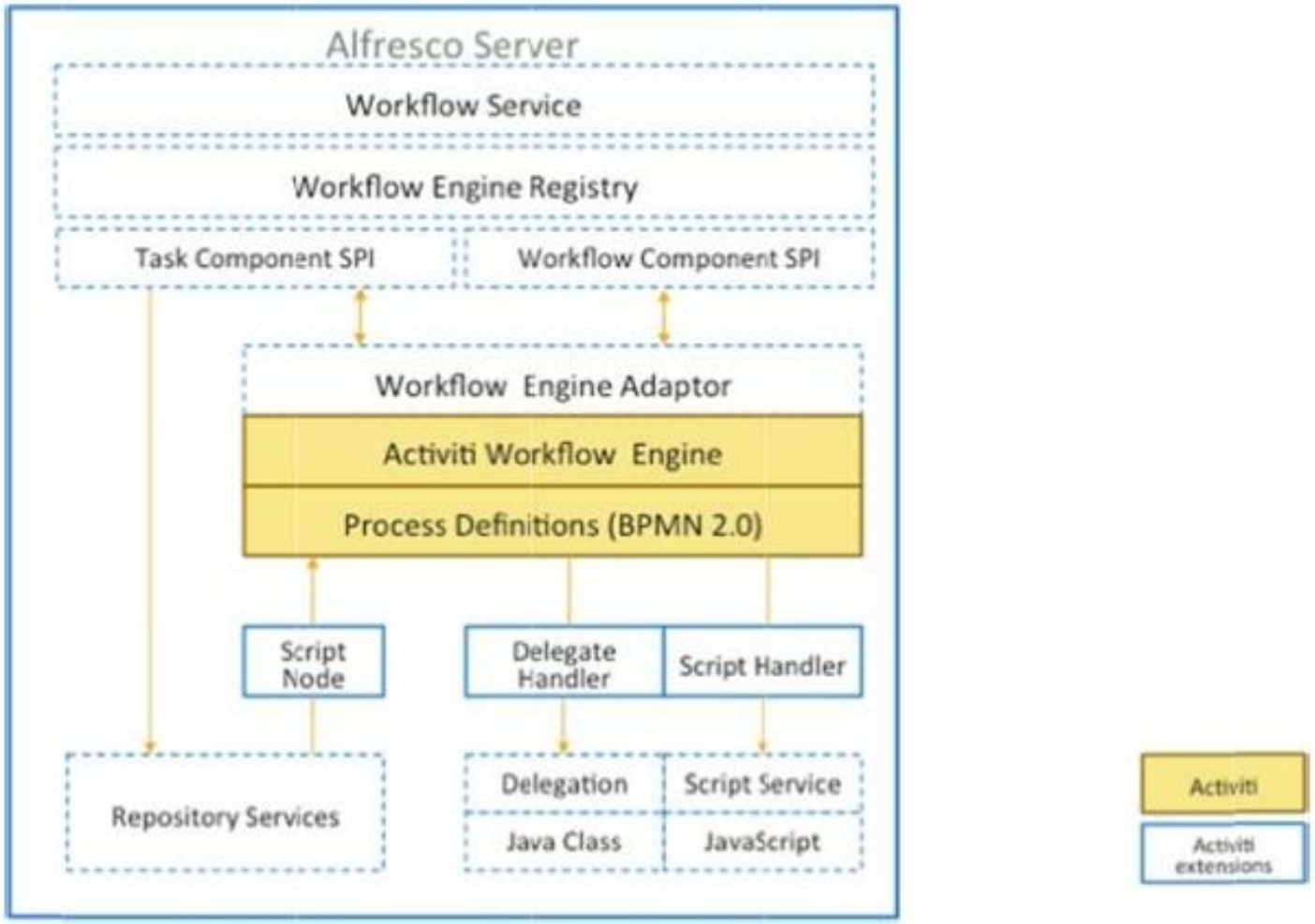
Workflow Architecture



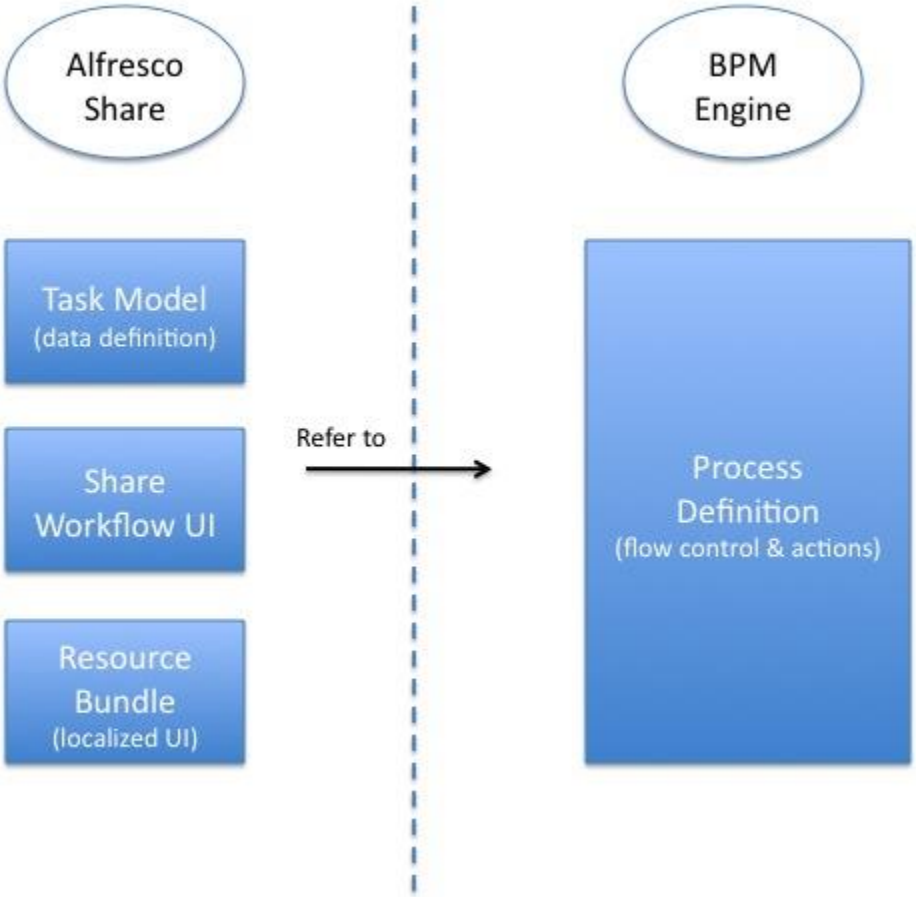
Workflow Architecture



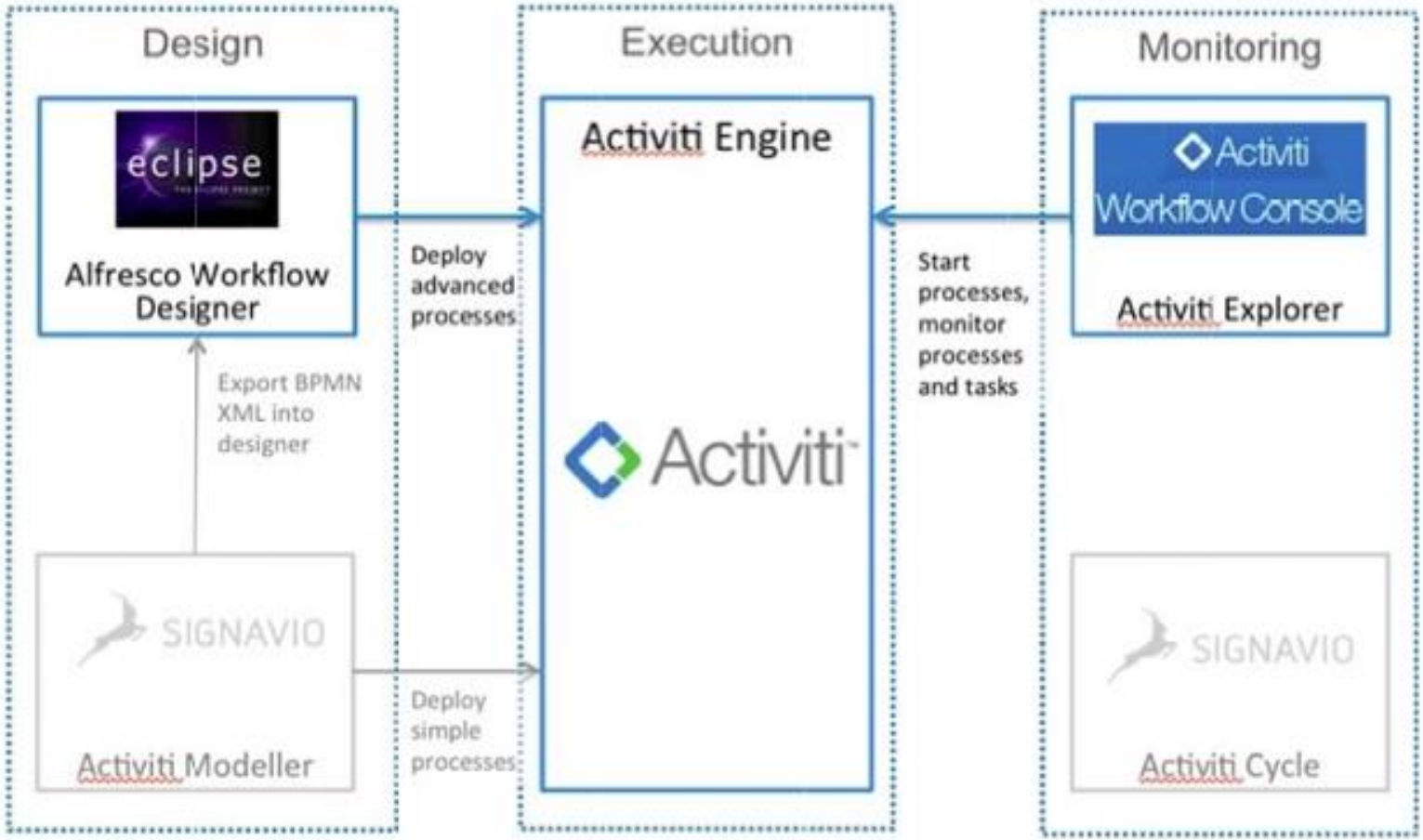
Workflow Architecture



Workflow Artifacts



Workflow Tools



Setting up Activiti designer

- Installing Eclipse
- Installing Activiti designer

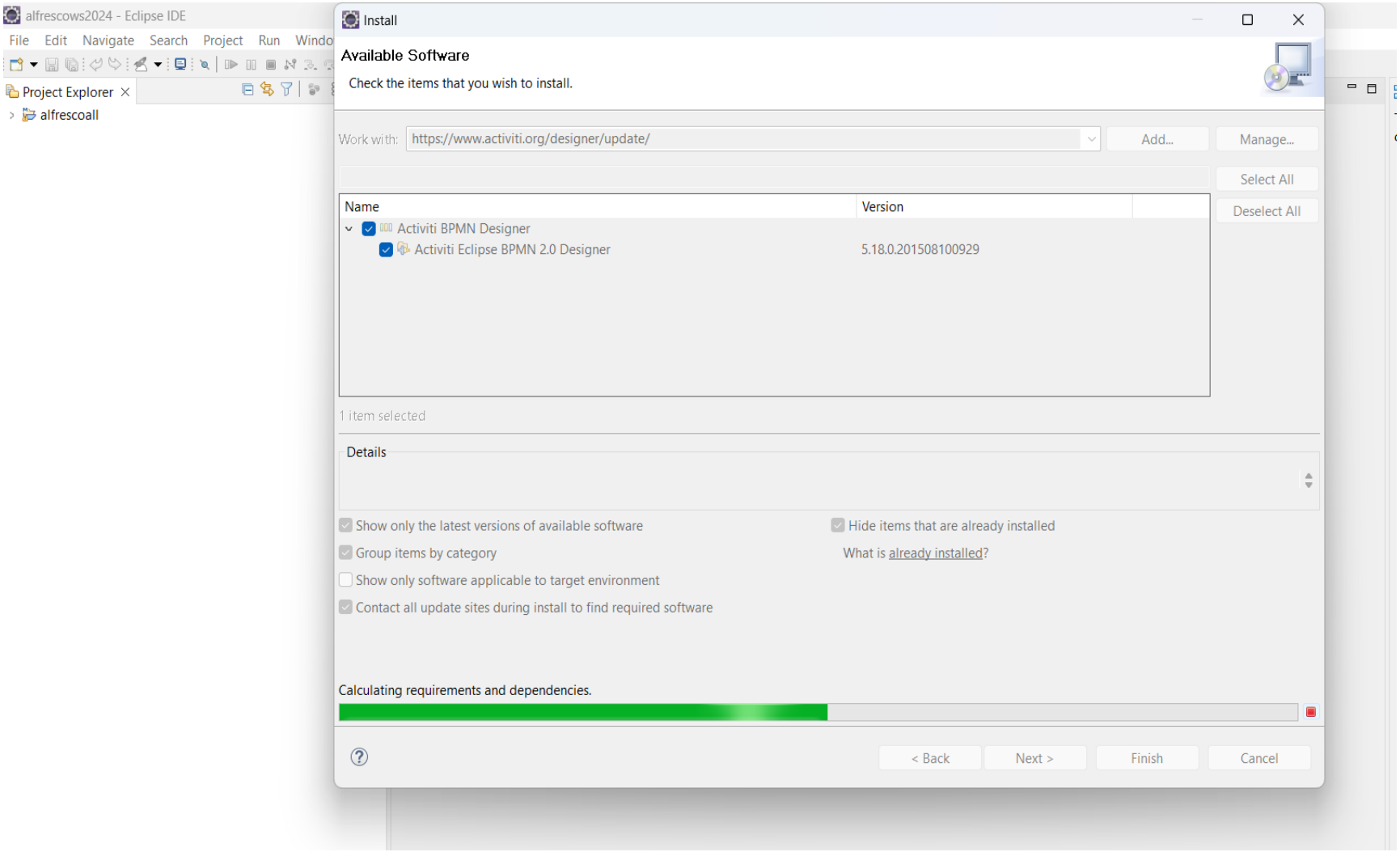
1. Start Eclipse.
2. In the eclipse menu bar, click **Help** > **Install New Software**.
3. Click **Add**.

   The Add Repository dialog is displayed.
4. Fill in the name field with ActivitiBPMN 2.0 designer, and fill in the location field with the following URL:

   ```
   https://www.activiti.org/designer/update/
   ```
5. Click **OK**.
6. Click **Next** and accept any license agreement check boxes, and then click **Finish**.

   Eclipse will install the latest version of the Activiti designer eclipse plugin.

Setting up Activiti designer



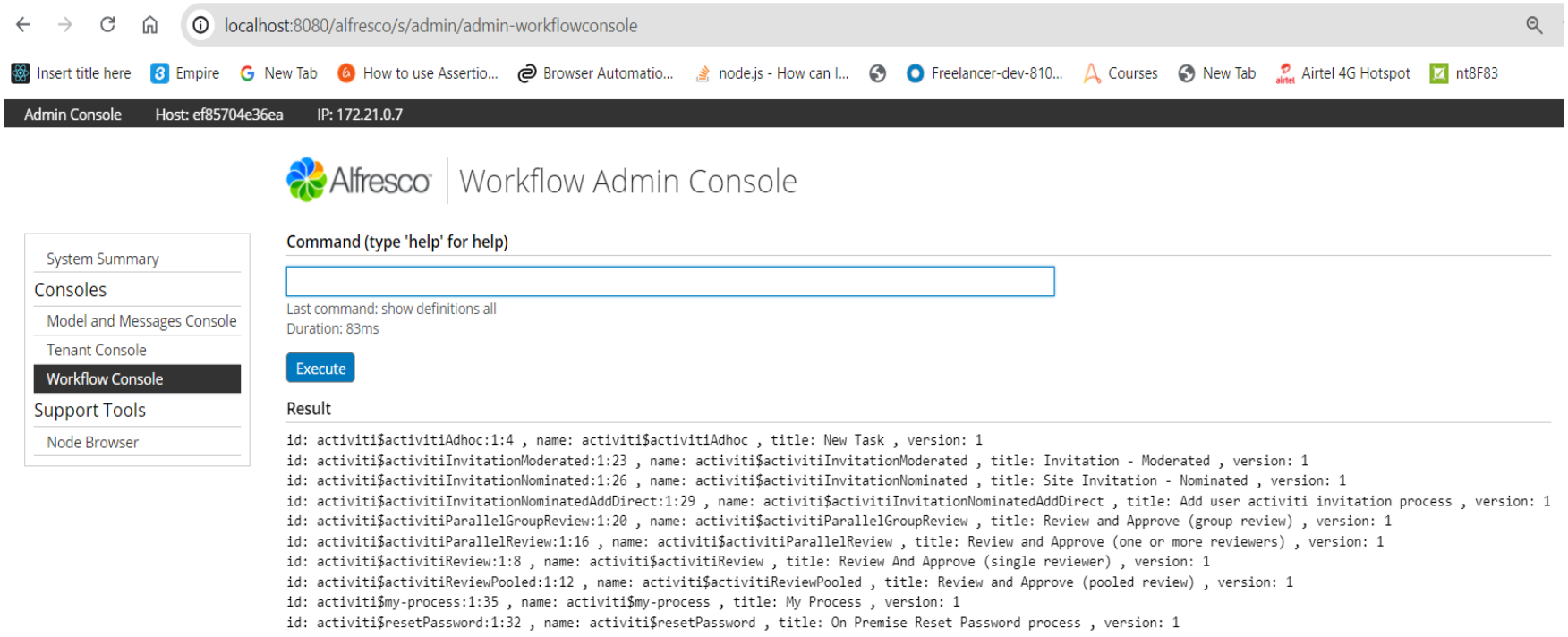
Alfresco Workflow Document

- https://ecmarchitect.com/alfresco-developer-series-tutorials/workflow/tutorial/tutorial.html#hello-world-examples

Activiti using Alfresco Web console

http://localhost:8080/alfresco/s/admin/admin-workflowconsole

Command
show definitions all



Activiti using Alfresco Web console

Alfresco | Workflow Admin Console

Command (type 'help' for help)

use definition activiti$helloworld:1:2804

Last command: start
Duration: 739ms

Execute

Result

```
started workflow id: activiti$2901 , def: helloworld
no further transitions
```

Activiti using Alfresco Web console

Alfresco | Workflow Admin Console

Command (type 'help' for help)

use definition activiti$helloworld:1:2804

Last command: start
Duration: 739ms

Execute

Result

```
started workflow id: activiti$2901 , def: helloworld
no further transitions
```

Activiti using Alfresco Web console

Alfresco | Workflow Admin Console

Command (type 'help' for help)

```
start
```

Last command: use definition activiti$helloworld:1:2804
Duration: 1ms

Execute

Result

```
definition: activiti$helloworld:1:2804 , name: helloworld , version: 1
workflow: None
path: None
```

Activiti using Alfresco Web console

Alfresco | Workflow Admin Console

Command (type 'help' for help)

Last command: start
Duration: 374ms

Execute

Result

```
started workflow id: activiti$2970 , def: helloworld
no further transitions
```

Activiti using Alfresco Web console Commands

Command	What it does
show workflows all	Shows all running workflows.
use workflow <workflow id> where <workflow id> is something like activiti$6300	Makes all subsequent commands happen in the context of the specified workflow.
show transitions	Shows all leaving transitions.
signal <path id> <transition> where <path id> is something like activiti$6334 and transition is the name of the leaving transition you want to take. Leave off the transition to take the default.	Signals the token. Good when your workflow is stuck on a node or when you want to take a transition without fooling with the task management UI.
desc path <path id> where <path id> is something like activiti$6334	Dumps the current context. Great for debugging process variables.
end workflow <workflow id> where <workflow id> is something like activiti$6300	Cancels the specified workflow.
show definitions all	Shows the current deployed workflows.
undeploy definition <workflow id> or undeploy definition name <workflow name>	Undeploys the specified workflow and stops any workflows running with that definition. The <workflow id> variant undeploys a specific version of a workflow.

Activiti Goals

- Activiti Introduction
- BPMN 2.0
- Activiti Modeler
- Activiti Explorer
- The Activiti API
- Deployment
- Forms
- REST API
- Cluster Deployment

Activiti

- Activiti Cloud is the first Cloud Native BPM framework built to provide a scalable and transparent solution for BPM implementations in cloud environments.

- The BPM discipline was created to provide a better understanding of how organizations do their work and how this work can be improved in an iterative fashion.

Activiti

- The Activiti project was started in 2010 by Tom Baeyens and Joram Barrez, the former founder and the core developer of jBPM (JBoss BPM), respectively.
- Activiti project is to build a rock-solid open source BPMN 2.0 process engine.
- Activiti is funded by Alfresco.
- But Activiti acts as an independent, open-source project.

Activiti

- The Activiti project was started in 2010 by Tom Baeyens and Joram Barrez, the former founder and the core developer of jBPM (JBoss BPM), respectively.
- Activiti project is to build a rock-solid open source BPMN 2.0 process engine.
- Activiti is funded by Alfresco.
- But Activiti acts as an independent, open-source project.

Activiti

- Activiti is a BPMN 2.0 process-engine framework that implements the BPMN 2.0 specification.
- It’s able to deploy process definitions, start new process instances, execute user tasks, and perform other BPMN 2.0 functions.
- But at its core, the Activiti Engine is a state machine.
- A BPMN 2.0 process definition consists of elements like events, tasks, and gateways that are wired together via sequence flows (think of arrows).
- When such a process definition is deployed on the process engine and a new process instance is started, the BPMN 2.0 elements are executed one by one.

Previous Versions

- Activiti Cloud uses at it's core Activiti 7.x which is an evolution from Activiti 6.x.
- Most of these changes will be simplifications to make sure that the framework itself doesn’t promote practices.
- It caused huge pains in real life implementations.
- If we are using Activiti 6.x or 5.x and it is working fine, no need to worry Activiti 7.x will.
- But if we are looking into microservices and cloud environments and we are suffering to adopt BPM tools and frameworks

Activiti 7

- Activiti 7 at is core will maintain the process engine, but related services such as:
  - IDM (User / Groups / Memberships)
  - Forms
  - History Service
  - Job executor
  - Timers
  - Email/Notification Services etc.,
- will be deprecated and refactored out in favor of integrations with third party components already provided by most cloud infrastructures.
- All these refactoring's have the primary goal of making sure that the Process Engine bootstrap and footprint is minimal.
- It doesn't overlap with functionalities and behavior which is expected to be provided by the infrastructure.

Activiti 7

- With the recent releases of Activiti 8.x, adopts Java 17 and Spring Boot 3.x.

Cloud Native BPMN support

- The BPMN specification describe many constructs (BPMN elements) that are allowed inside a Business Process definition.
- While supporting this in Activiti 5.x and 6.x was all done by the Process Engine.
- In Activiti Cloud (Activiti Core 7.x) less elements are supported due the fact that we are now dealing with a distributed and highly scalable infrastructure.
- Elements such as timers, signals and messages now involve interaction with the infrastructure and other services to work correctly.

The list of supported BPMN elements in the 7.1.x Release

- Start / End Events
- SequenceFlows (conditional, default)
- Service Task
- User Task (assignee, candidateUsers, candidateGroups)
- Gateways: Parallel, Exclusive, Inclusive
- Call Activity
- Signal Intermediate Catch Event
- Signal Intermediate Throw Event, Signal Boundary Event
- Embedded Subprocess

Getting started with Activiti Cloud

- It has three options

Getting started with Activiti Cloud : here are 3 options.

- Docker Compose (local install)
- Automated CI/CD with Jenkins-X
  - on AWS EKS
  - on GCP GKE
- Using Helm
  - on AWS EKS
  - on GCP GKE

Getting Started with Activiti Cloud

- Activiti Cloud is a set of Cloud Native components designed from the ground up to work in distributed environments.
- Activiti chosen Kubernetes as our main deployment infrastructure.
- They are using Spring Cloud / Spring Boot along with Docker for containerization of these components.

Getting Started with Activiti Cloud

- Activiti Cloud includes 5 foundational building blocks:
  - Activiti Cloud Runtime Bundle
  - Activiti Cloud Query
  - Activiti Cloud Audit
  - Activiti Cloud Connectors
  - Activiti Cloud Notifications Service (GraphQL)

Getting Started with Activiti Cloud

- These building blocks are Spring Boot Starters that can be attached to any Spring Boot (2.x) application.
- These building blocks are enhanced with Spring Cloud functionalities which provide the Cloud Native capabilities.

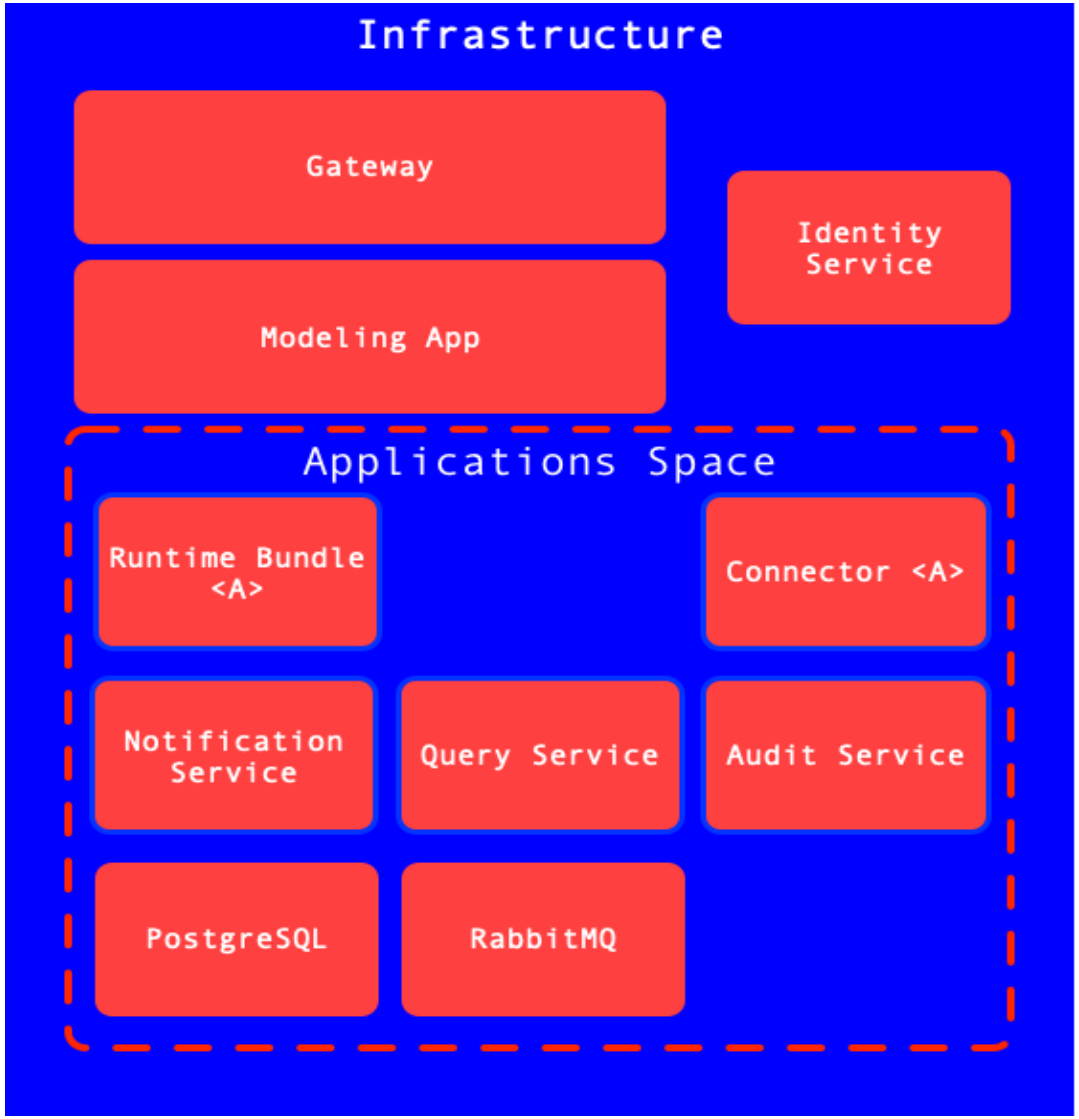
Getting Started with Activiti Cloud

- By using these components, we can create Activiti Cloud applications that
- can be scaled independently based on demand
- can be managed in completely isolated cycles
- can be upgraded and maintained independently
- can provide domain specific features using the right tool for the job

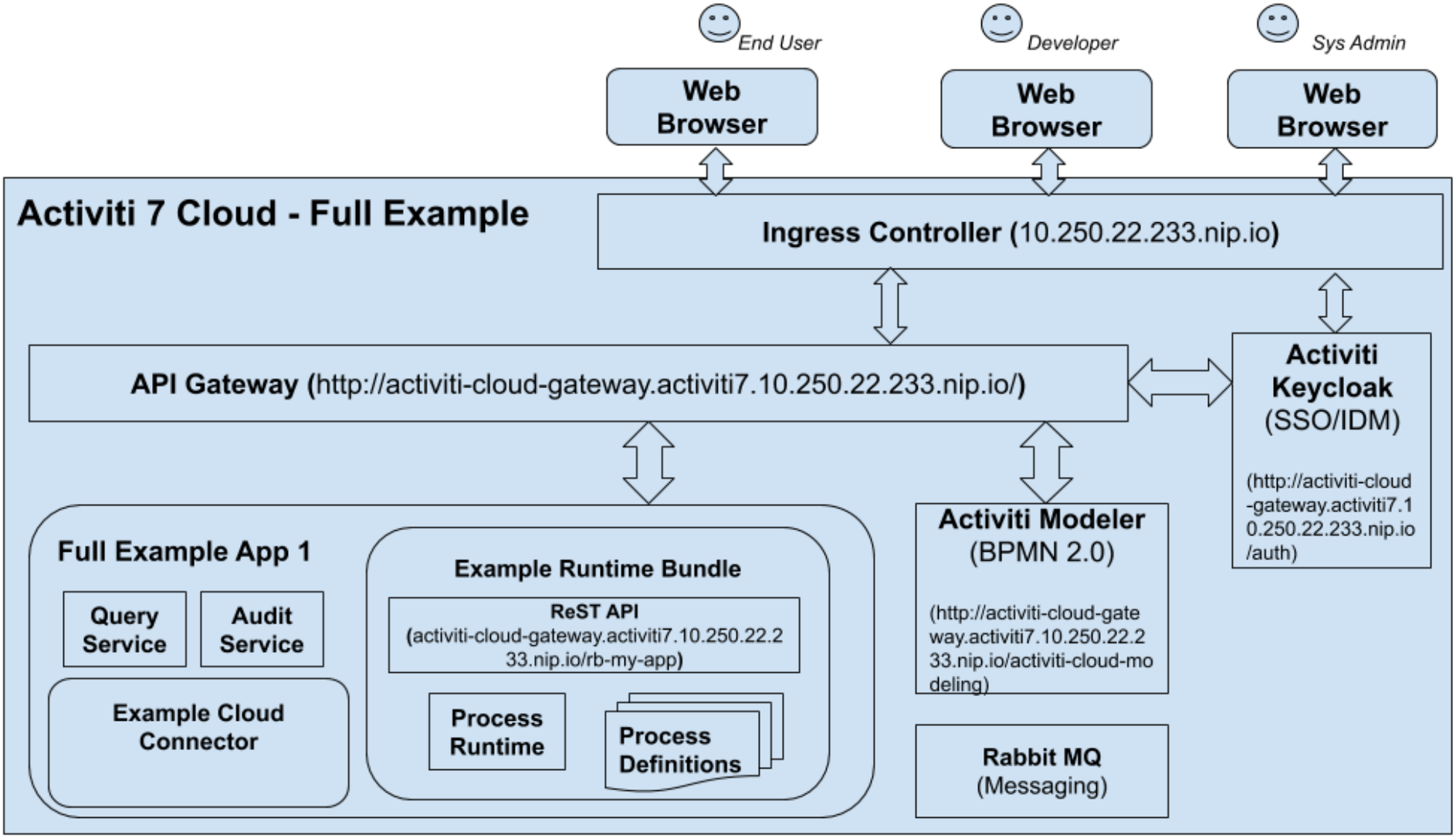
Prerequisite

- Docker Desktop with Kubernetes Enabled
- Helm
- (or)
- AWS / GCP Account

Activiti Cloud Full Example



Activiti Cloud Full Example



Activiti Cloud Full Example

- Activiti Keycloak - SSO - Single Sign On to all services
- Activiti Keycloak - IDM - Identity Management so Activiti knows what the organisational structure looks like and who the groups and users are, and who are allowed to do what.
- Activiti Modeler - BPMN 2.0 modelling application where developers can define new process applications with one or more process definitions.

Activiti Cloud Full Example

- API Gateway - Gives user interfaces and other systems access to the process applications
- Service Registry - a service registry that allows all the services and applications to register with it and be discovered in a dynamic way.
- Config Server - centralized and distributed configuration service that can be used by all other services.
- Zipkin - is a distributed tracing system. It helps gather timing data needed to troubleshoot latency problems in microservice architectures.

Activiti Cloud Full Example

- Activiti Applications - each application represents one or more business process implementations.
- So, one application could be an HR process for job applications while another application could be used for loan applications.
- Cloud connectors are used to make calls outside of the business process..

Activiti Cloud Full Example

- Runtime Bundles - will provide different runtimes for different business models.
- The first available runtime is for business processes.
- The Process Runtime would be comparable to what was previously referred to as the Activiti Process Engine.
- A runtime will be as small as possible and efficient as possible.
- A Runtime Bundle contains several process definitions and is immutable.
- This means that it will run just several immutable process definitions (instances).

Activiti Cloud Full Example

- Query Service - is used to aggregate data and makes it possible to consume the data in an efficient way.
- There might be multiple Runtime Bundles inside an Activiti application and the Query Service will aggregate data from each of those.
- Notification Service - can provide notifications about what is happening in the different Runtime Bundles.

Activiti Cloud Full Example

- Audit Service - this is the standard audit log that you have in BPM systems, it provides a log of what exactly happened when a process was executed.
- Cloud Connectors - are about System - to - System interaction.
- Instead of having all code that talks to external systems in the process definition implementation and the process runtime, it’s now decoupled and implemented as separate services with separate SLAs.
- A Service Task would typically be implemented as a Cloud Connector.

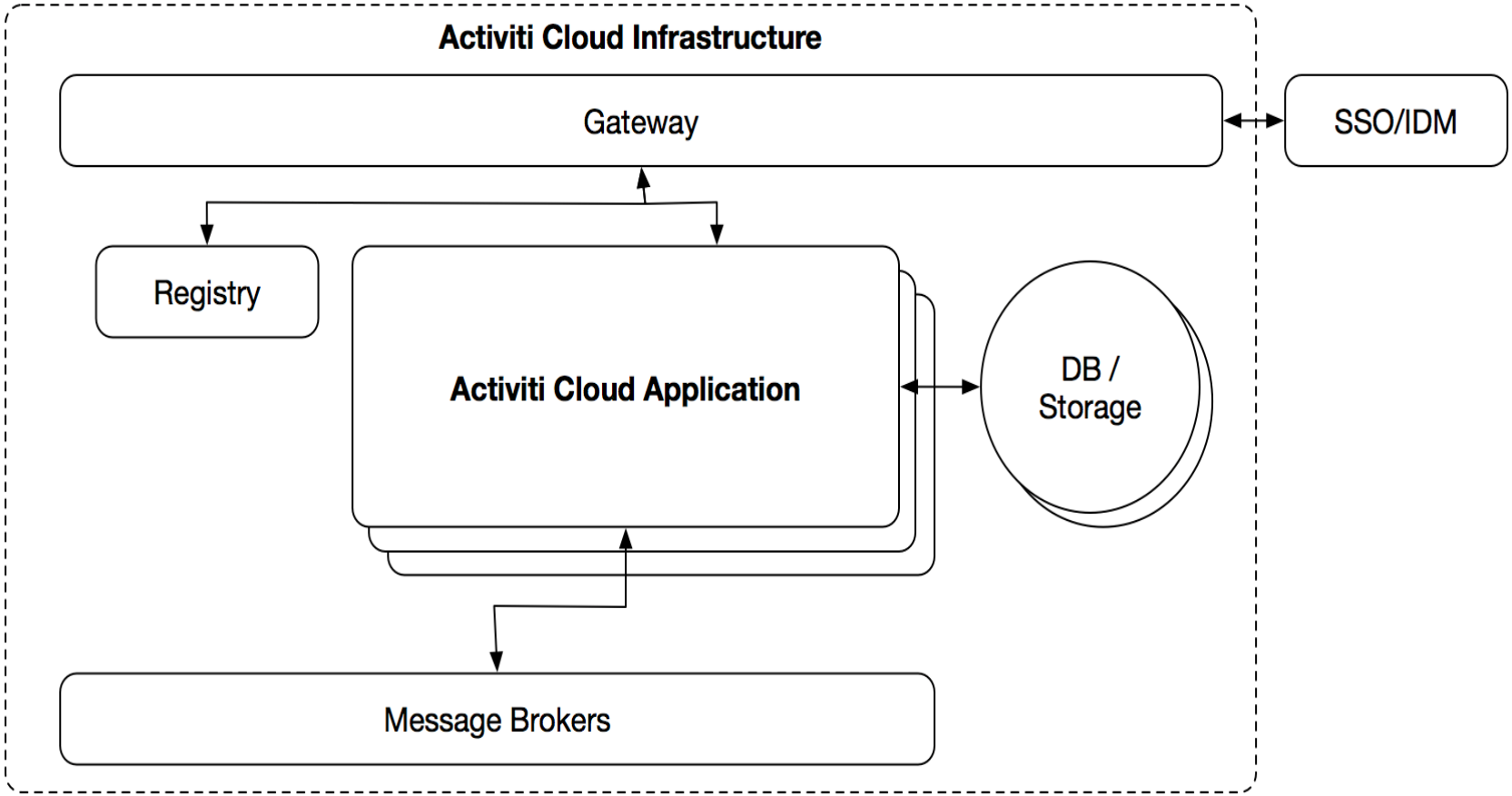
Activiti 7 Full Example

- API Gateway (Spring Cloud)
- SSO/IDM (Keycloak)
- Activiti 7 Runtime Bundle (Example)
- Activiti 7 Cloud Connector (Example)
- Activiti 7 Query Service
- Activiti 7 Audit Service

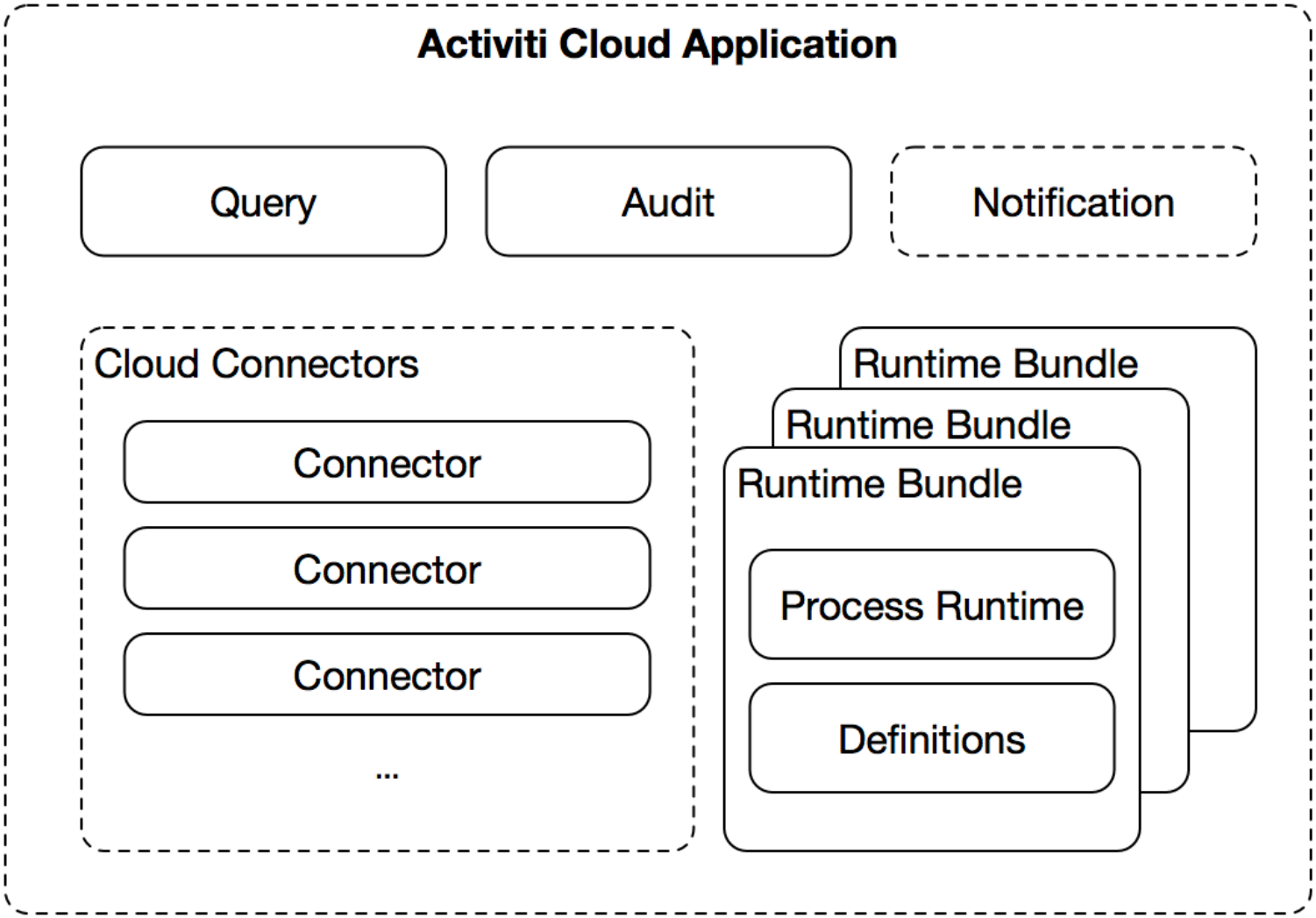
Activiti Components

- Activiti Cloud provides a set of basic building blocks which can be divided into 3 separate groups:
- Activiti Cloud Infrastructure
- Activiti Cloud Application
- Activiti Cloud Modeler

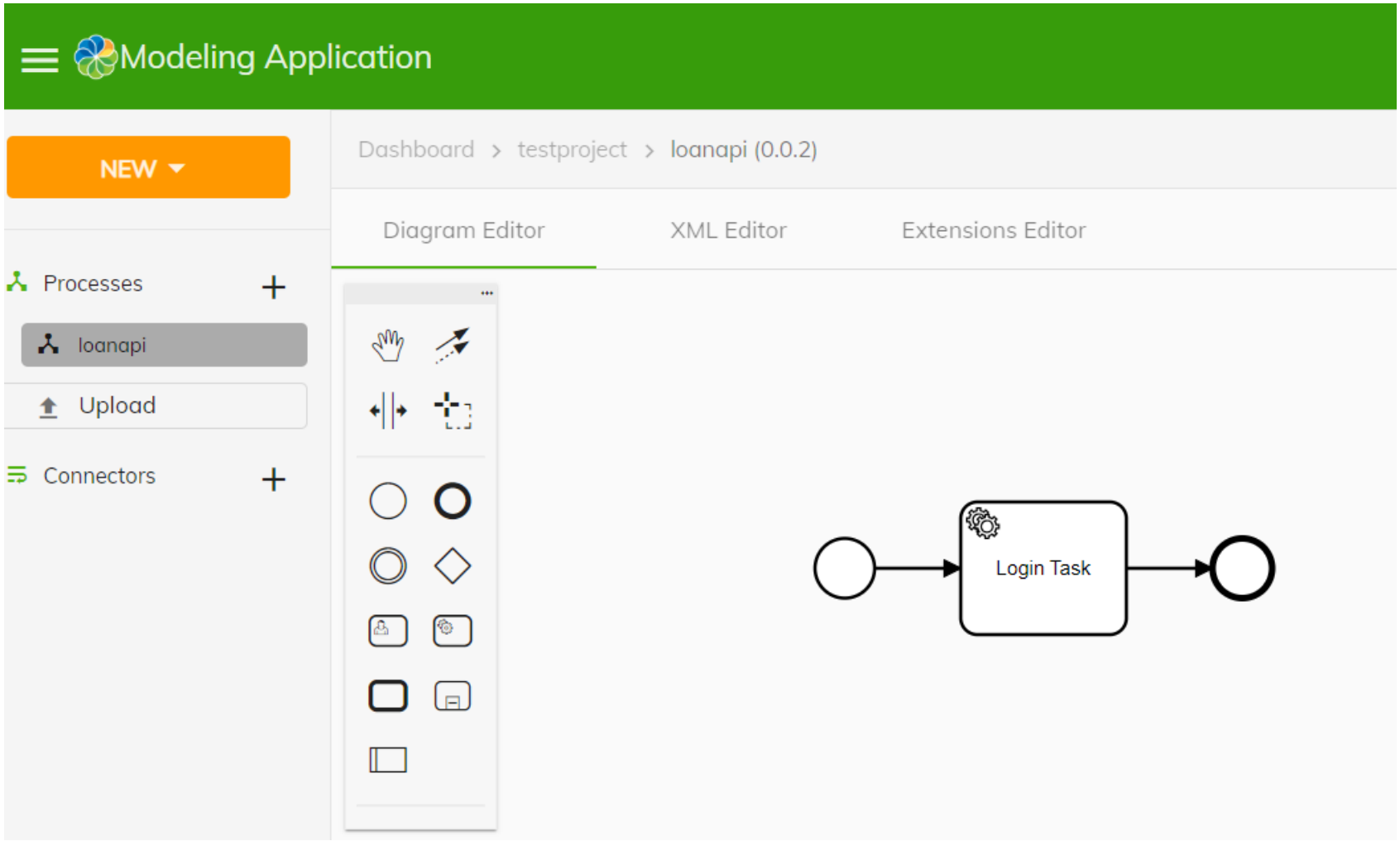
Activiti Components



Activiti Components Large Scale Deployments



Activiti Components Large Scale Deployments



Activiti Cloud

Runtime Bundle

Inmutable, scalable & pain free Process & Decision Runtimes designed to integrate with your cloud native infrastructure.

Audit Service

Scalable, storage independent and extensible audit service.

Query Service

Scalable, storage independent and extensible query service.

Cloud Connectors

Simplified system to system interactions that can scale in distributed environments.

Application Service

Distributed & Scalable application aggregation layer.

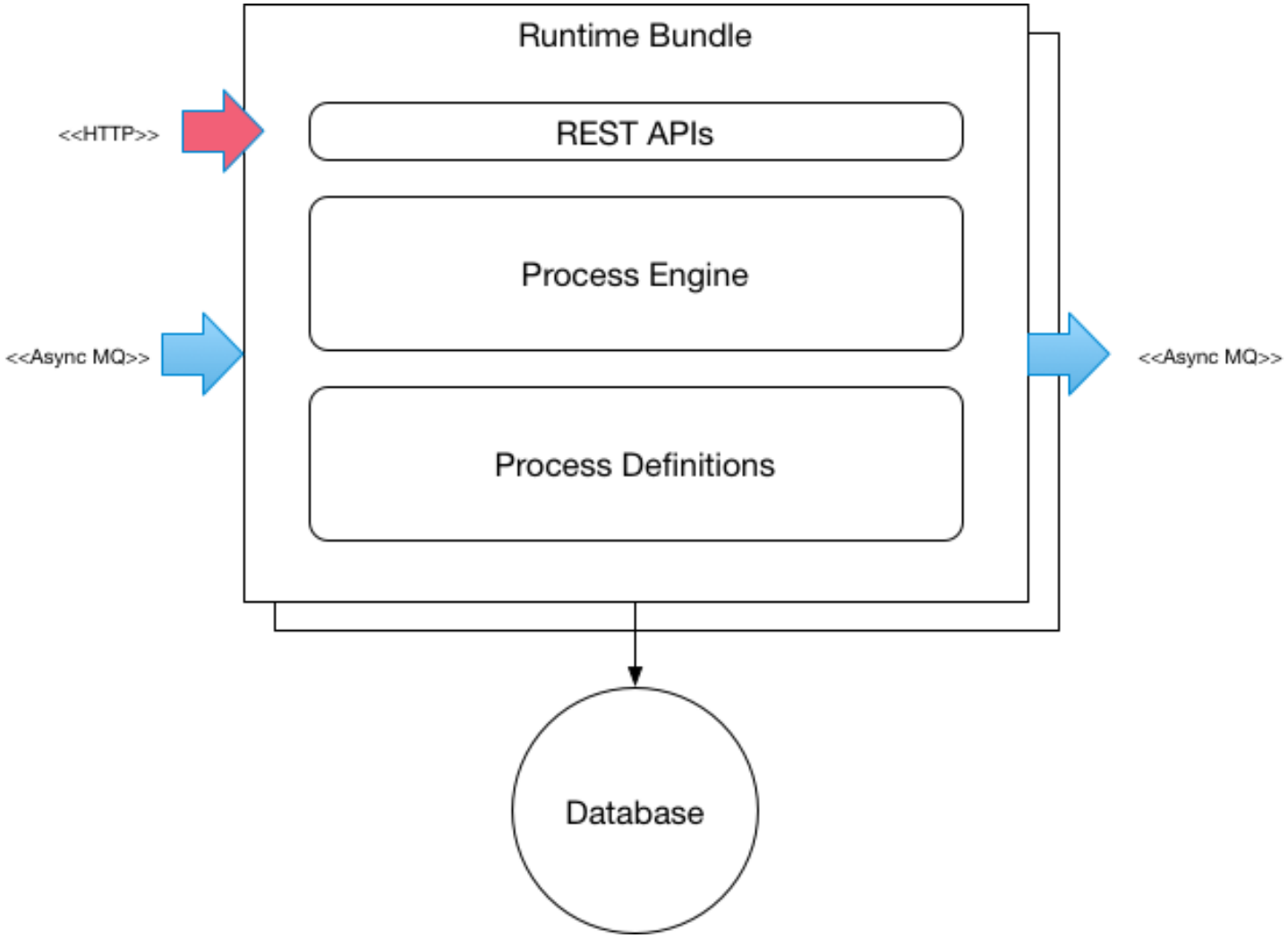
Notification Service

Cloud ready secure WebSocket and subscription handling as part of GraphQL integration.

Activiti Cloud Runtime Bundle

- Runtime Bundles in the context of Activiti Cloud represent a stateless instance of the process engine to execute an immutable set of process definitions.
- We cannot deploy new process definitions to a Runtime Bundle.
- Instead, we will create a new immutable version of your Runtime Bundle if we want to update your process definitions.
- Runtime Bundles expose a (Sync) REST and (Async) Message Based API to interact with them.

Activiti Cloud Runtime Bundle



Activiti 8 Postman Environment Settings

activitienv Edit

Variable	Initial value	Current value
gateway	gateway-default.abc.vhebconsulting.com	gateway-default.abc.vhebconsulting.com
idm	identity-default.abc.vhebconsulting.com	identity-default.abc.vhebconsulting.com
realm	activiti	activiti

Postman Collection Execution

activitienv | POST getKeycloakToken testu | activitienv

Activiti v7 REST API / keycloak / **getKeycloakToken testuser** | Save

POST | {{idm}}/auth/realms/{{realm}}/protocol/openid-connect/token | Send

Params | Authorization | Headers (11) | Body | Pre-request Script | Tests | Settings | Cookies

none | form-data | x-www-form-urlencoded | raw | binary | GraphQL

	Key	Value	Description
☑	client_id	activiti	
☑	grant_type	password	
☑	username	testuser	
☑	password	password	
	Key	Value	Description

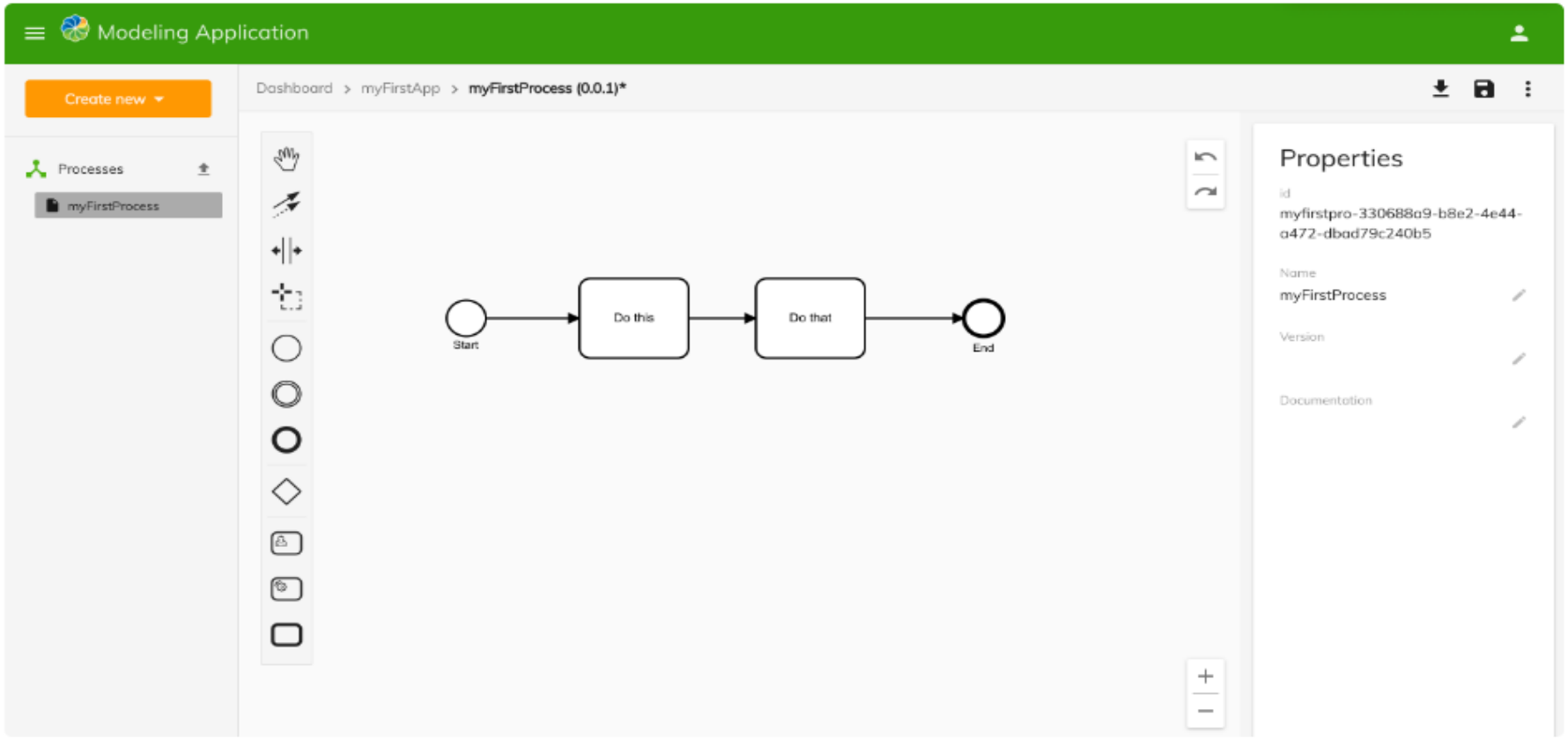
Postman Collection Execution


```
"access_token":
    "eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJCQl9VV2FVeTI0VjE2TFpxOEM1V3NScUZxUldUODlyajlNVHVMMFRGMU13In0.
    eyJleHAiOjE3MDI5MTc1NDcsImlhdCI6MTcwMjkxNzI0NywianRpIjoiZGI3YjhlNjEtNmQ4OC00MDgyLWExOTEtMzJiOGI4ZjU0ZTQyIiwiaXNzIjoiaHR0c
    DovL2lkZW50aXR5LWRlZmF1bHQuYWJjLnZoZWJjb25zdWx0aW5nLmNvbS9hdXRoL3JlYWxtcy9hY3Rpdml0aSIsImF1ZCI6WyJyZWFsbS1tYW5hZ2VtZW50Ii
    wiYWNjb3VudCJdLCJzdWIiOiI4MjI4ZjAwZC1lNWY3LTRiMjQtYjEzOS1mMzM4NTM3NzRhOGEiLCJ0eXAiOiJCZWFyZXIiLCJhenAiOiJhY3Rpdml0aSIsInN
    lc3Npb25fc3RhdGUiOiI0MTk1ZjU3My01ODBjLTQzMTktODBiMy04NWI1MTdlZGY0NGUiLCJhY3IiOiIxIiwiYWxsb3dlZC1vcmlnaW5zIjpbIioiXSwicmVh
    bG1fYWNjZXNzIjp7InJvbGVzIjpbIm9mZmxpbmVfYWNjZXNzIiwiQUNUSVZJVElfVVNFUiIsInVtYV9hdXRob3JpemF0aW9uIl19LCJyZXNvdXJjZV9hY2Nlc
    3MiOnsicmVhbG0tbWFuYWdlbWVudCI6eyJyb2xlcyI6WyJ2aWV3LXVzZXJzIiwidmlldy1jbGllbnRzIiwicXVlcnktY2xpZW50cyIsInF1ZXJ5LWdyb3Vwcy
    IsInF1ZXJ5LXVzZXJzIl19LCJhY2NvdW50Ijp7InJvbGVzIjpbIm1hbmFnZS1hY2NvdW50IiwibWFuYWdlLWFjY291bnQtbGlua3MiLCJ2aWV3LXByb2ZpbGU
    iXX19LCJzY29wZSI6ImVtYWlsIHByb2ZpbGUiLCJzaWQiOiI0MTk1ZjU3My01ODBjLTQzMTktODBiMy04NWI1MTdlZGY0NGUiLCJlbWFpbF92ZXJpZmllZCI6
    ZmFsc2UsIm5hbWUiOiJUZXN0IFVzZXIiLCJncm91cHMiOlsidGVzdGdyb3VwIl0sInByZWZlcnJlZF91c2VybmFtZSI6InRlc3R1c2VyIiwiZ2l2ZW5fbmFtZ
    SI6IlRlc3QiLCJmYW1pbHlfbmFtZSI6IlVzZXIiLCJlbWFpbCI6InRlc3R1c2VyQGV4YW1wbGUuY29tIn0.
    LqoNMutfP3RkC_pm30fSNi8KAOB8izmJUZ8MIV7WVKYJ4aXVRCkeTBrtXjFRsd9JA32wxUWZQdKeJXJR-mPCn5bnA7O8I9lf0vXYljHN6MsIf1T2q-6QkuRMX
    aX0VcBmo7oHLr-ks0tvuft2Dl9F4v2dyCf62JGPi7YhR1WZA7YP-GTB5ZgO4oJCjKD54VW5KNlCA9RCEYs0SJGwjDI8rHLFaTUaWMAnpXA6GW47CWCmT9xP3z
```

Activiti Cloud Modelling(Angular)

The User Interface built with Angular can be found here

This application uses BPMN.io for the BPMN process standard xml generation.



New process modeling experience

Activiti Cloud Runtime Bundle

REST APIs (HAL)

Runtime Bundles expose a REST API with the following **user** (**ACTIVITI_USER** role) endpoints:

- v1/process-definitions
- v1/process-definition/{id}/meta
- v1/process-instances/
- v1/process-instances/{id}/tasks
- v1/process-instances/{id}/variables
- v1/tasks
- v1/tasks/{id}/variables

Activiti Cloud Runtime Bundle

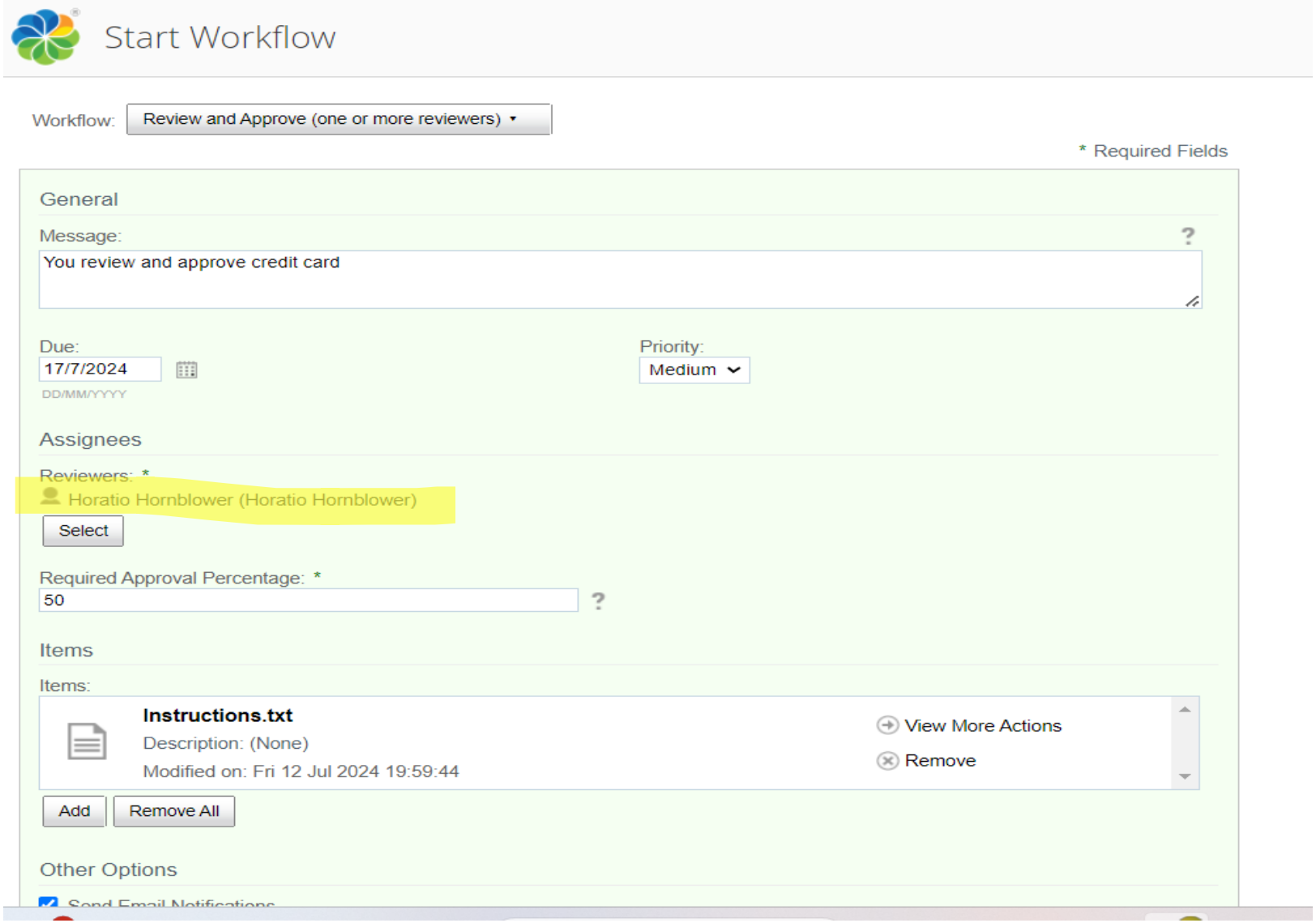
Also the following **admin** (**ACTIVITI_ADMIN** role) endpoints:

- admin/v1/process-definitions/
- admin/v1/process-instances
- admin/v1/tasks

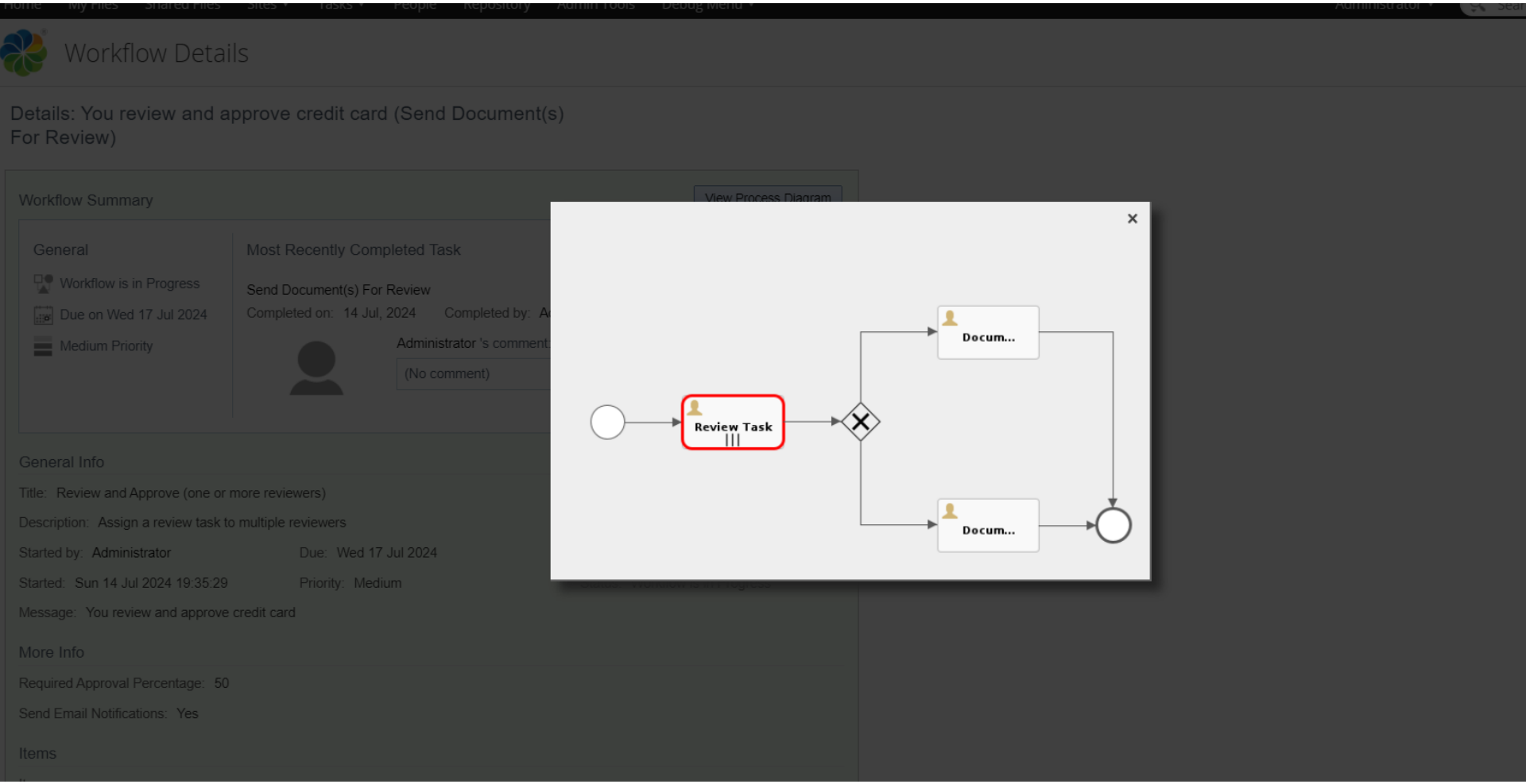
Activiti Cloud Connectors

- Activiti Cloud Connectors provide scalable way to perform bi-directional system to system interactions.
- It enables business processes to interact with external systems in a safe way without compromising the process runtime.

Create Workflow in Tasks



Create Workflow in Tasks



Create Workflow in Tasks

General Info

Title: Review and Approve (one or more reviewers)

Description: Assign a review task to multiple reviewers

Started by: Administrator | Due: Wed 17 Jul 2024 | Completed: <in progress>

Started: Sun 14 Jul 2024 19:35:29 | Priority: Medium | Status: Workflow is in Progress

Message: You review and approve credit card

More Info

Required Approval Percentage: 50

Send Email Notifications: Yes

Items

Items:

Instructions.txt

Description: (None)

Modified on: Fri 12 Jul 2024 19:59:44

Current Tasks

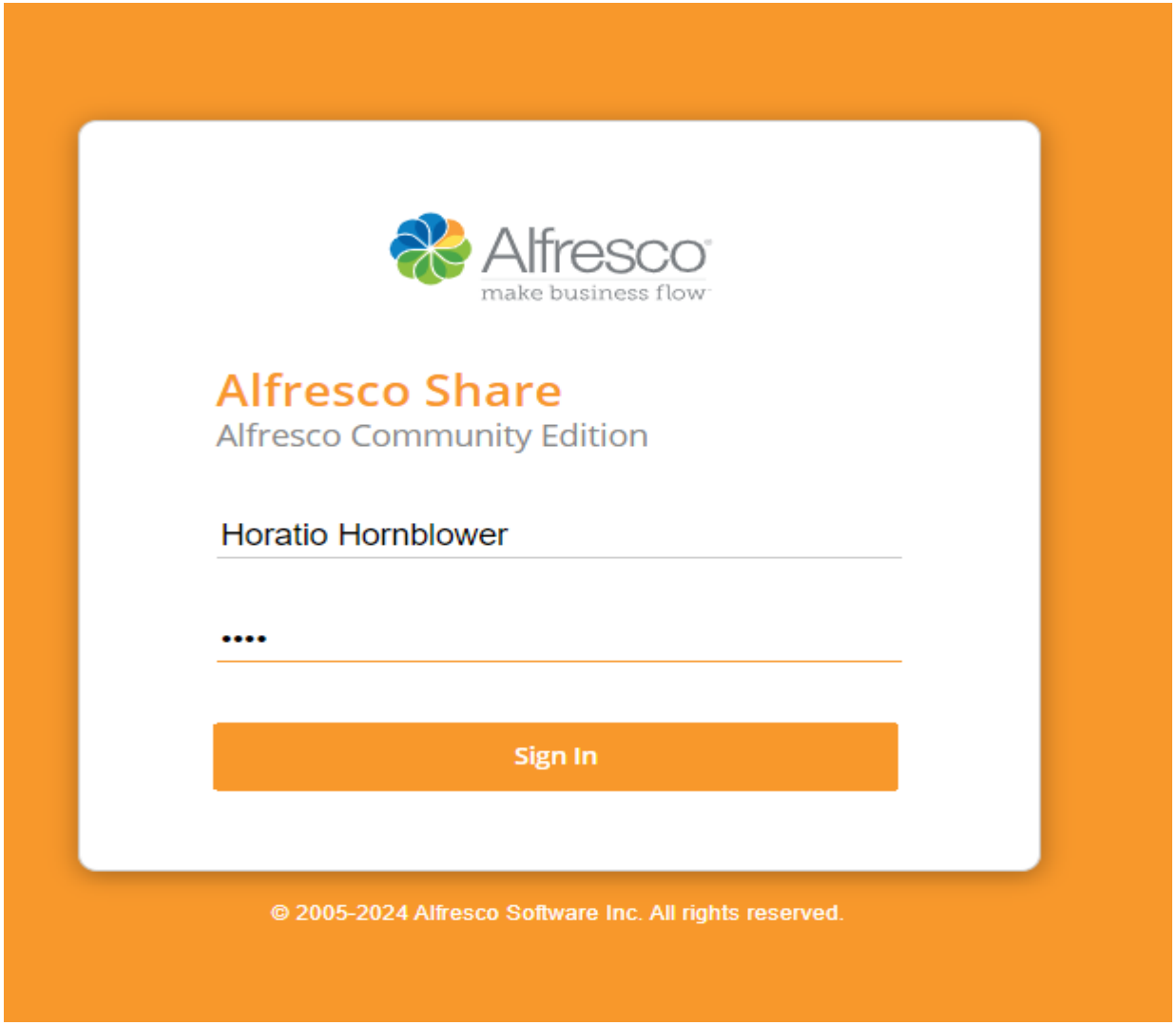
Type	Assigned To	Due Date	Status	Actions
Review	Horatio Hornblower	Wed 17 Jul 2024	Not Yet Started	

History

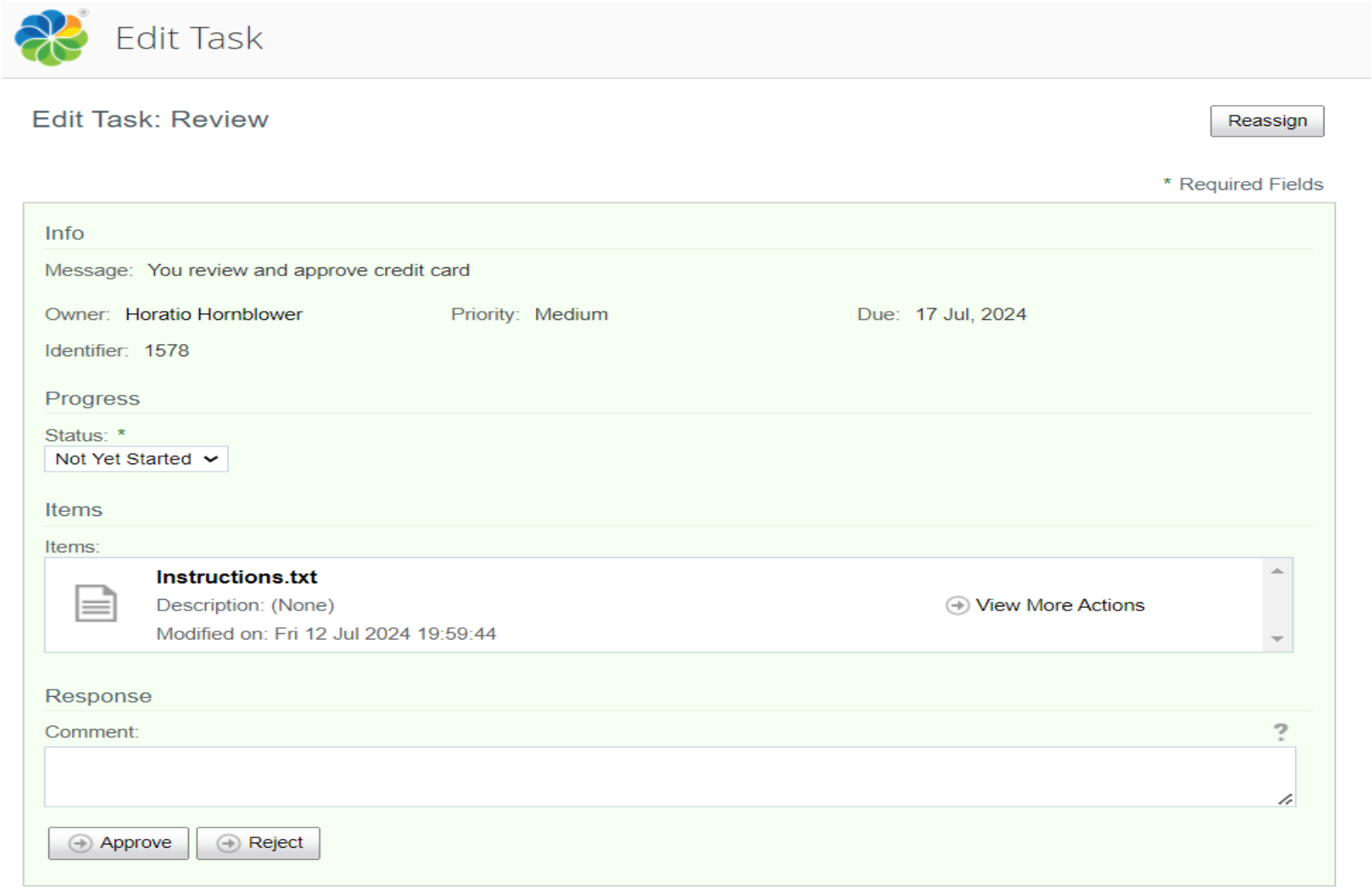
Type	Completed By	Date Completed	Outcome	Comment
Send Document(s) For Review	admin	Sun 14 Jul 2024 19:35:29	Task Done	

Cancel Workflow

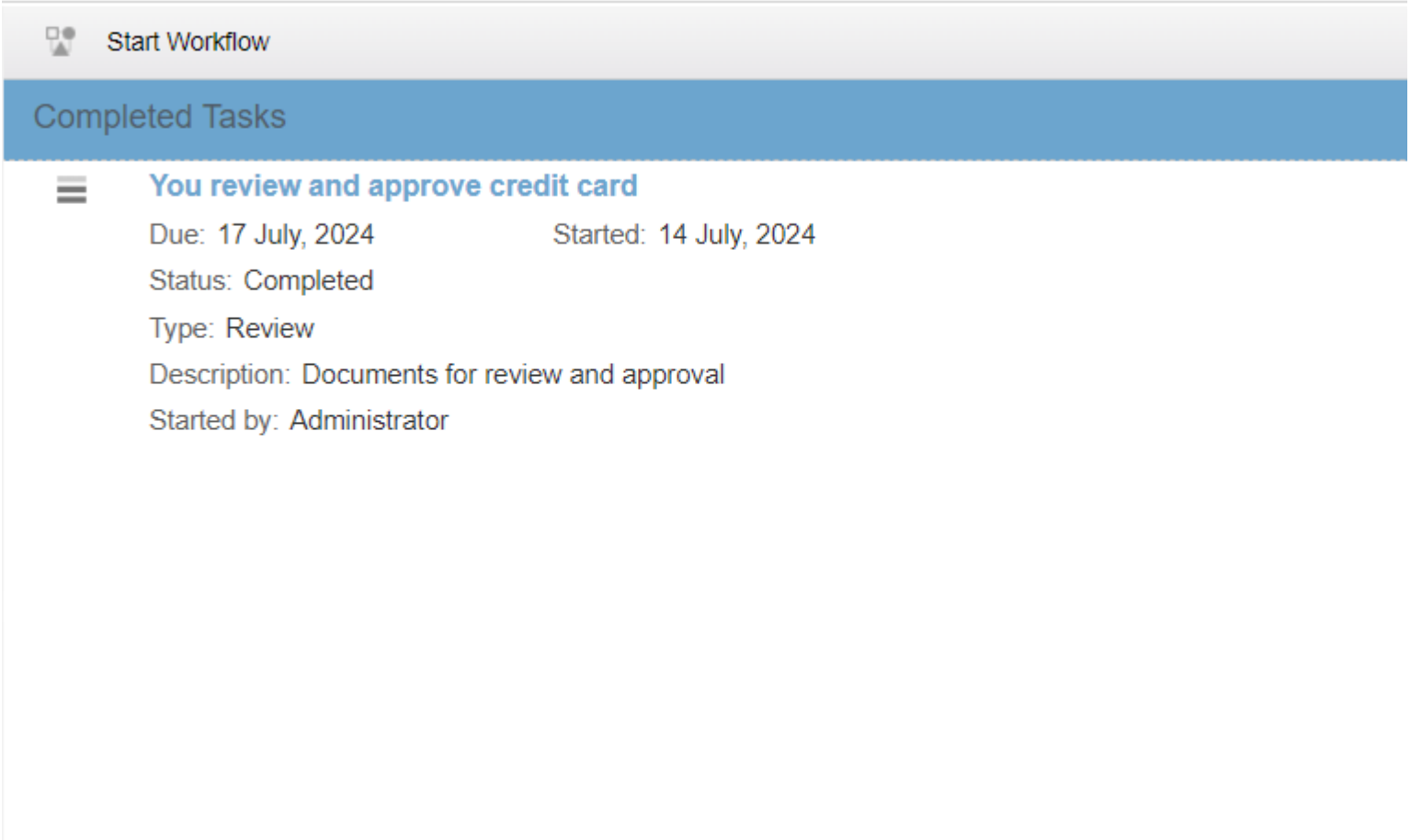
Create Workflow in Tasks



Create Workflow in Tasks



Create Workflow in Tasks



Create Workflow in Tasks

Task Details

Details: You review and approve credit card (Review)

Task Details | Workflow Details

Info

Message: You review and approve credit card

Owner: Horatio Hornblower Priority: Medium Due: 17 Jul, 2024

Identifier: 1578

Progress

Status: Completed

Items

Items:

Instructions.txt

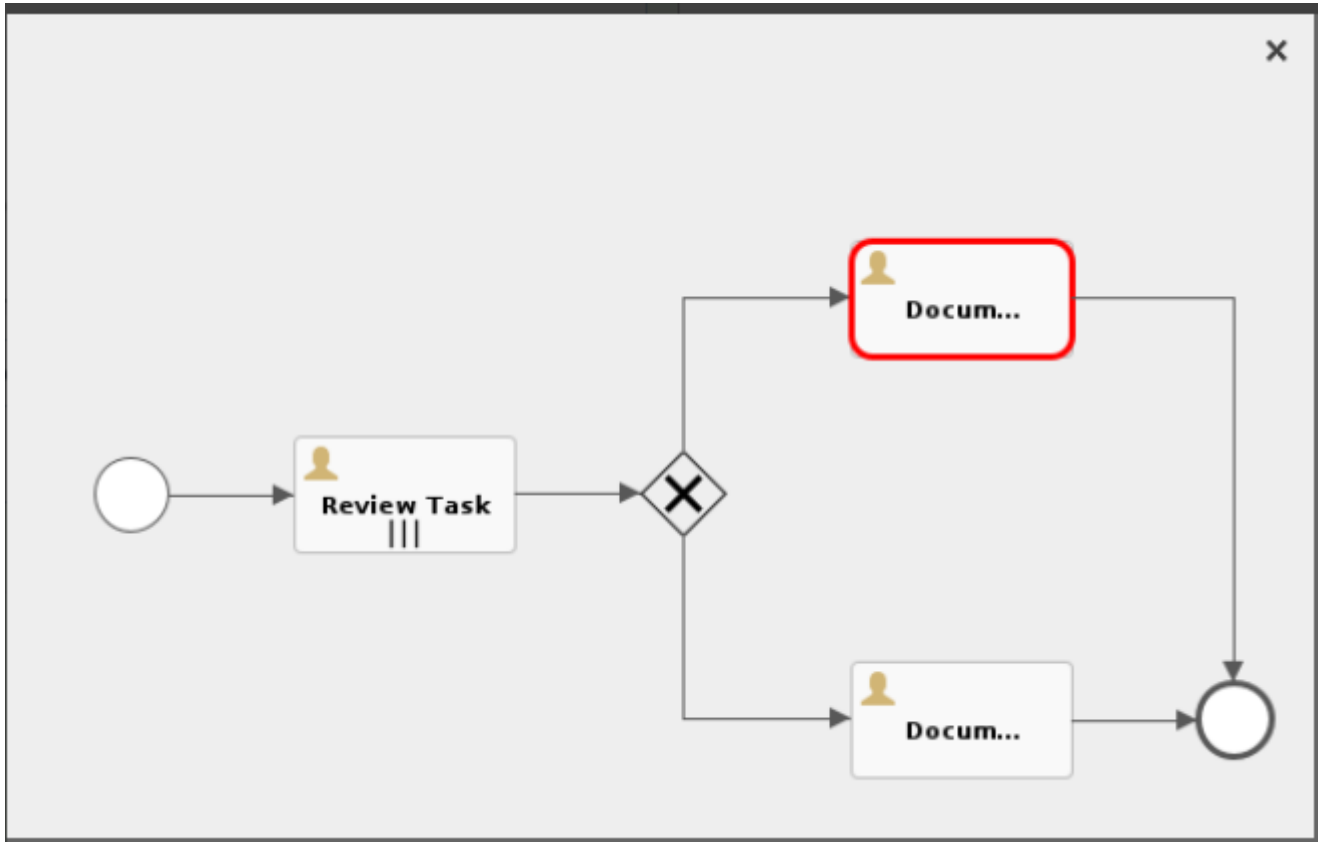
Description: (None)

Modified on: Fri 12 Jul 2024 19:59:44

Response

Comment: Proceed to issue the card

Create Workflow in Tasks



Create Workflow in Tasks

Task Details

Details: The document was reviewed and approved. (Document Approved)

Task Details | Workflow Details

Info

Message: The document was reviewed and approved.

Owner: Administrator
Priority: Medium
Due: 17 Jul, 2024
Identifier: 1624

Progress

Status: Completed

Outcome

Number of Reviewers: 1
Reviewers Who Approved: 1
Required Approval Percentage: 50
Actual Approval Percentage: 100

Items

Items:

Instructions.txt
Description: (None)
Modified on: Fri 12 Jul 2024 19:59:44

Response

Comment: Card Issued

Create Workflow in Tasks

General Info

Title: Review and Approve (one or more reviewers)

Description: Assign a review task to multiple reviewers

Started by: Administrator | Due: Wed 17 Jul 2024 | Completed: Sun 14 Jul 2024 19:46:11

Started: Sun 14 Jul 2024 19:35:29 | Priority: Medium | Status: Workflow is Complete

Message: You review and approve credit card

More Info

Required Approval Percentage: 50

Send Email Notifications: Yes

Items

Items:

Instructions.txt

Description: (None)

Modified on: Fri 12 Jul 2024 19:59:44

Current Tasks

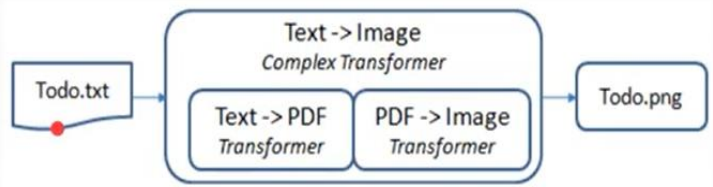
Type	Assigned To	Due Date	Status	Actions
No tasks				

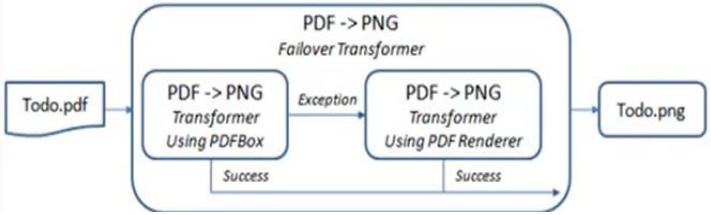
History

Type	Completed By	Date Completed	Outcome	Comment
Document Approved	admin	Sun 14 Jul 2024 19:46:11	Task Done	Card Issued
Review	Horatio Hornblower	Sun 14 Jul 2024 19:40:57	Approved	Proceed to issue the card
Send Document(s) For Review	admin	Sun 14 Jul 2024 19:35:29	Task Done	

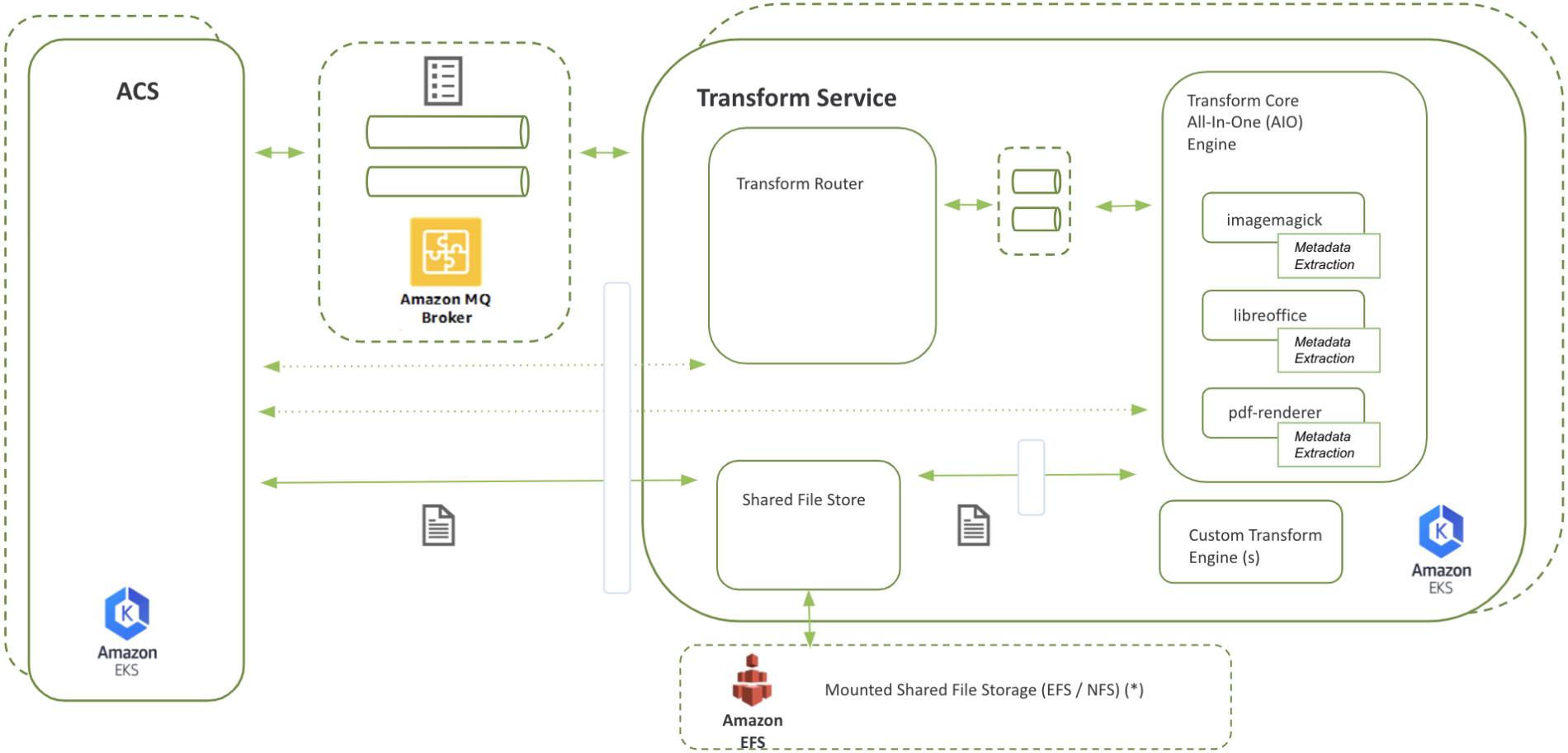
Transformers

- In many cases content should be transformed to another format when uploaded, part of a workflow, as response to an action etc
- Many transformers comes out-of-the-box:
  - PDF -> Text, Office types -> Text
  - HTML -> Text, Email -> Text
  - PDF -> Image, Text -> PDF
  - Text -> Image
- There are three types of transformers:
  - Format X -> Format Y
  - Format X -> Format Y -> Format Z (Complex)
  - Format X -> Format Y, if fail try different Format





Administer Transform Service



ReST API

- The Alfresco ReST API version 1.0 is a complete application interface that gives you access to all the features of the Alfresco Repository.
- When building remote extensions, the Alfresco ReST API is the preferred interface.
- The endpoint to access the API has the following format:
- http://localhost:8080/api-explorer/

Web scripts

- Web scripts provide a unique way to programmatically interact with the Content Services server.
- The REST (Representational State Transfer) web architecture is based on HTTP requests and responses, URIs (Uniform Resource Identifiers), and document types.
- Web scripts let you implement your own RESTful API without tooling or Java knowledge, requiring only a text editor.
- This approach to developing Content Services API means that web scripts offer many advantages over existing technologies, including ease and speed of development, and flexibility in API design.

Web scripts

- By focusing on the RESTful architectural style, web scripts let you build custom URI-identified and HTTP accessible content management web services backed by the Content Services server.
- Web scripts provide RESTful access to content held within your repository.
- You can place controls on your content to manage it and provide uniform access for a wide variety of client applications and services, such as a browser, portal, search engine, or custom application.
- It can apply web technologies such as caching, authentication, proxies, and negotiation to your repository resources.

Web scripts

- The Content Services CMIS (Content Management Interoperability Services) AtomPub binding is implemented as a series of web scripts.
- You can use web scripts for various solutions, such as:
  - Integrating Content Services with third party systems
  - Providing feeds
  - Developing data services
  - Developing UI services such as portlets
  - Customizing search
  - Acting as a back-end to client tools, such as Orbeon Forms
  - Integrating with Microsoft Office
  - Developing Facebook applications
  - Building UI components in Surf

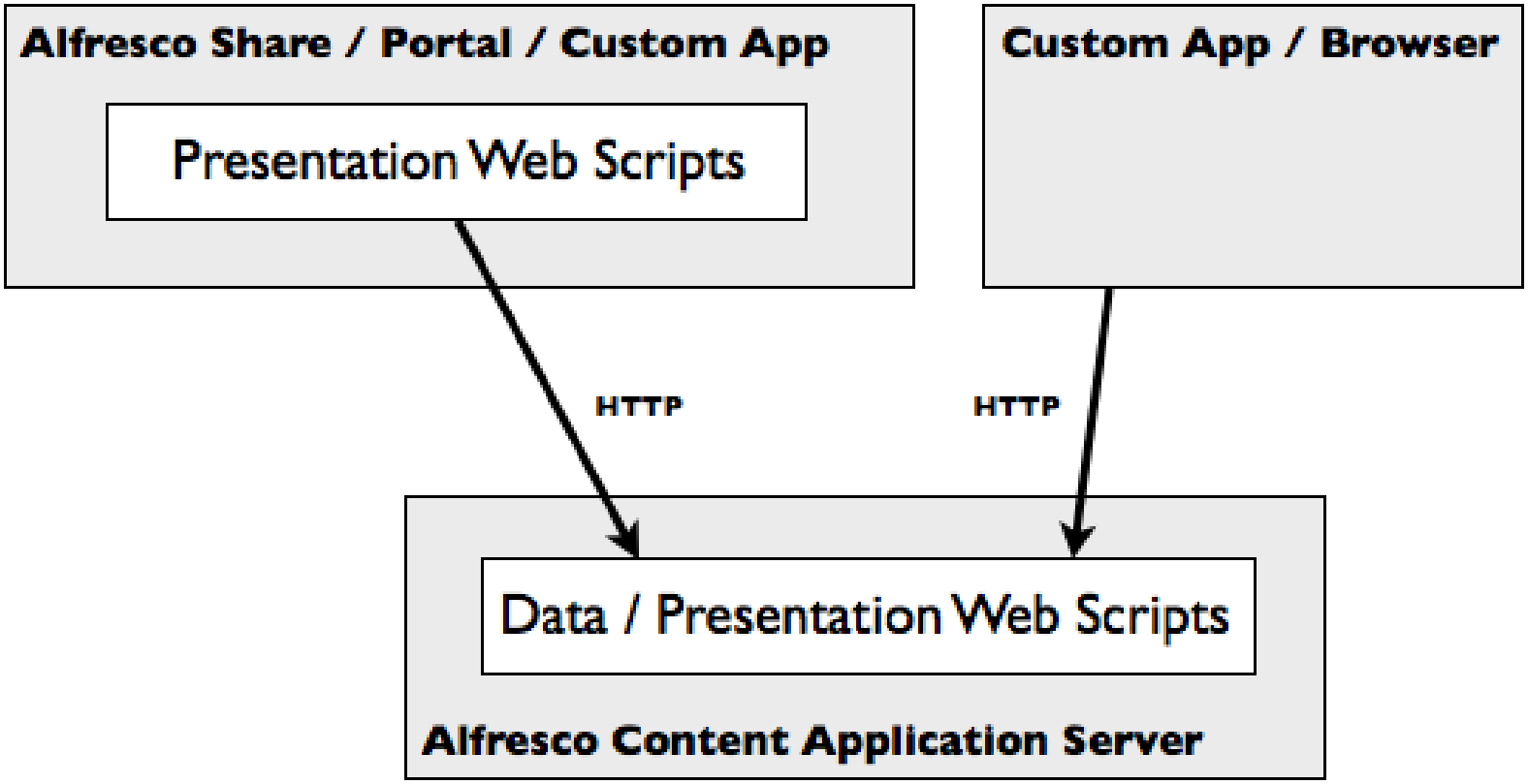
Web scripts

- Web Scripts are a way of implementing REST-based API.
- They could also be referred to as Web Services.
- They are stateless and scale extremely well.
- Repository Web Scripts are defined in XML, JavaScript, and Free Marker files and stored under alfresco/extension/templates/webscripts.
- Repository Web Scripts are referred to as Data Web Scripts as they usually return JSON or XML.

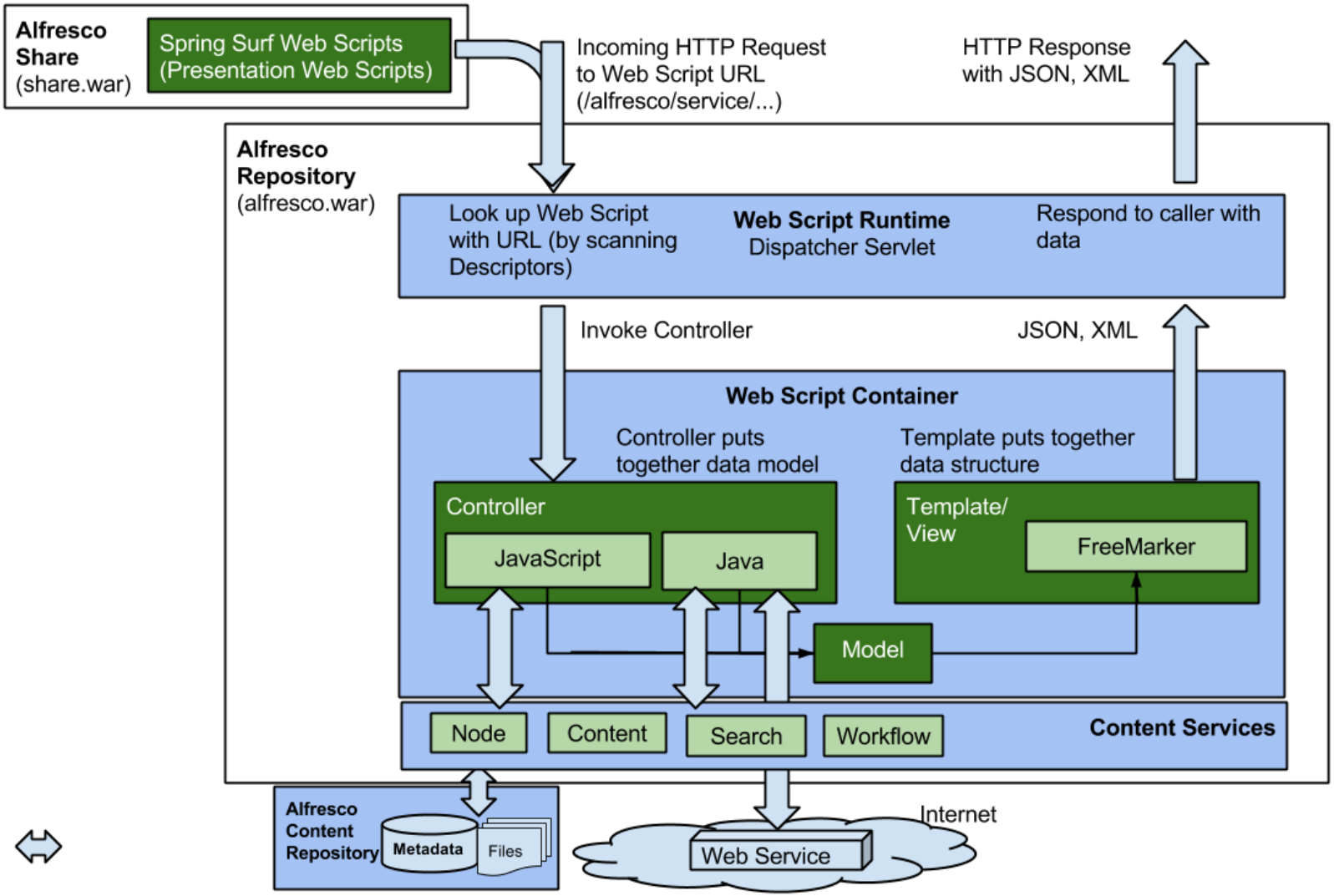
Web scripts Types

- A web script is a service bound to a URI that responds to HTTP methods such as GET, POST, PUT, and DELETE.
- There are two kinds of web scripts that use the same underlying code:
  - Data web scripts
  - Presentation web scripts

Web scripts Types



Web scripts


Alfresco Share (share.war)
Spring Surf Web Scripts (Presentation Web Scripts)
Incoming HTTP Request to Web Script URL (/alfresco/service/...)
HTTP Response with JSON, XML
Alfresco Repository (alfresco.war)
Look up Web Script with URL (by scanning Descriptors)
Web Script Runtime
Dispatcher Servlet
Respond to caller with data
Invoke Controller
JSON, XML
Web Script Container
Controller puts together data model
Template puts together data structure
Controller
JavaScript
Java
Template/ View
FreeMarker
Model
Node
Content
Search
Workflow
Content Services
Alfresco Content Repository
Metadata
Files
Internet
Web Service

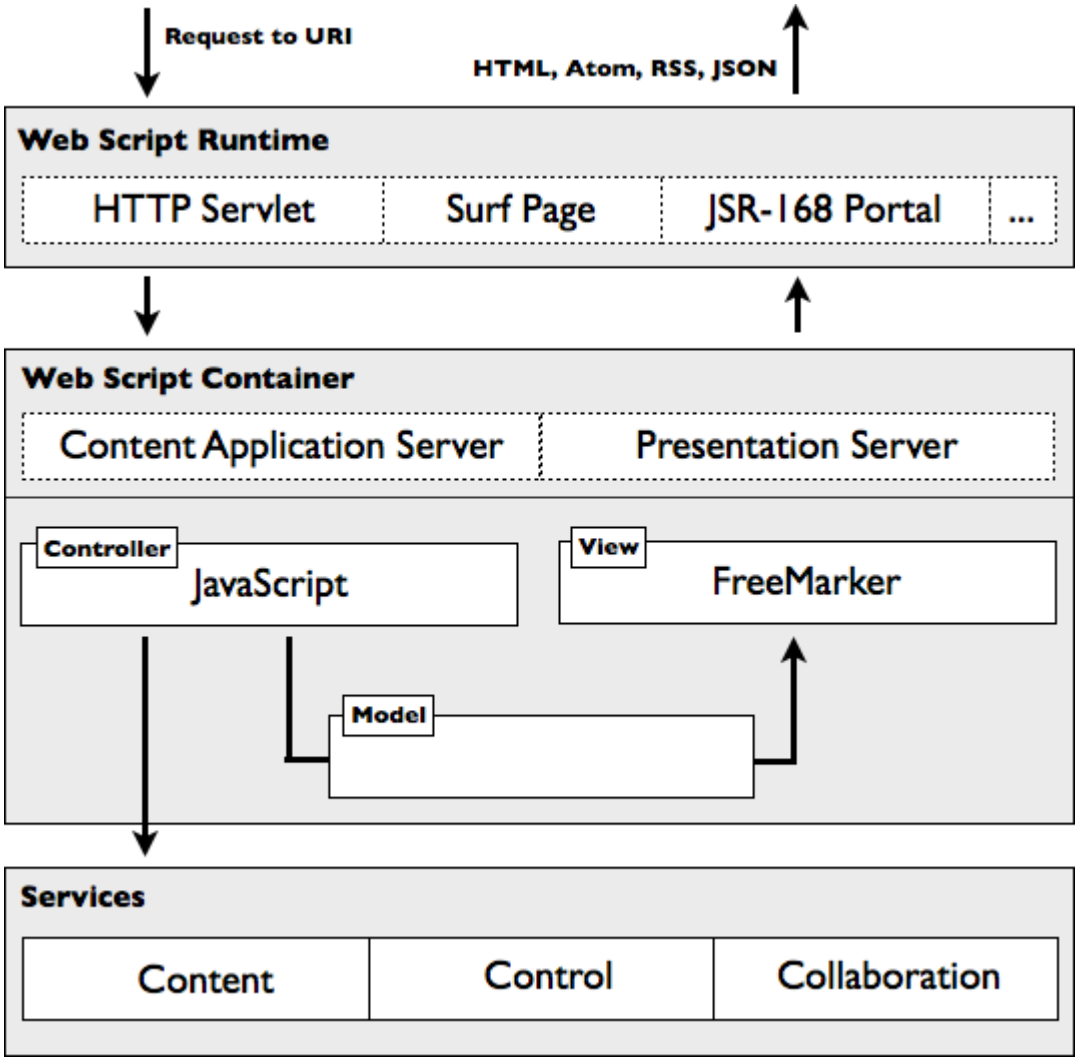
Web scripts

- Repository web scripts
  - Repository web scripts encapsulate access and modification of content/data held in the content repository; therefore, they are provided and exposed only by the Content Services server.
  - Data web scripts provide a server interface for client applications to query, retrieve, update, and perform processes, typically using request and response formats such as XML and JSON.

Web scripts

- Presentation web scripts
  - Presentation web scripts let you customize and extend the web UI.
  - They typically render HTML and can include browser-hosted JavaScript.
  - Unlike data web scripts, presentation web scripts can be hosted in the Content Services server or in a separate presentation server.
  - When hosted separately, presentation web scripts in the presentation server interact with data web scripts in the Alfresco content application server by using the Repository REST API.

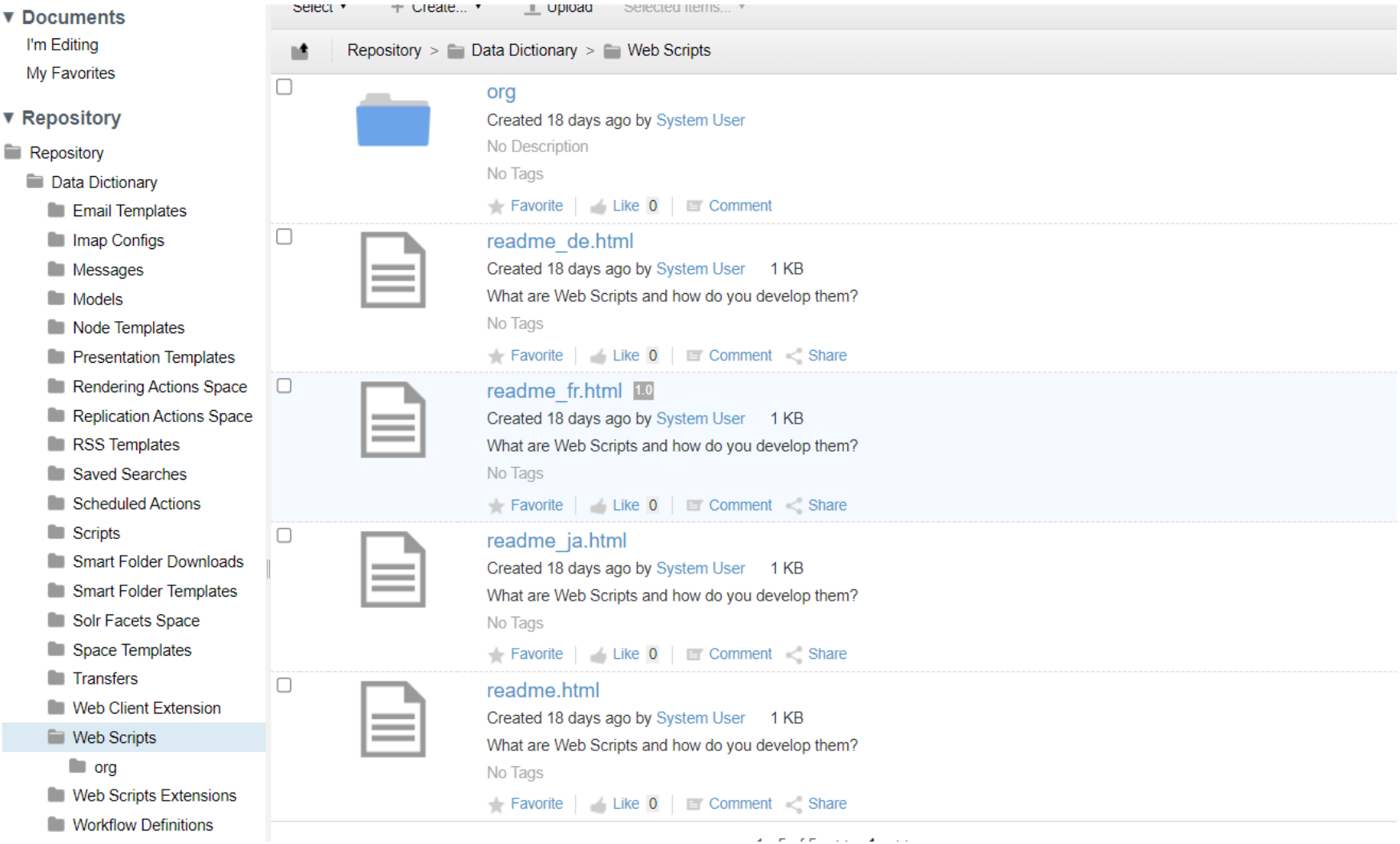
Web scripts


Request to URI
HTML, Atom, RSS, JSON
Web Script Runtime
HTTP Servlet
Surf Page
JSR-168 Portal
...
Web Script Container
Content Application Server
Presentation Server
Controller
JavaScript
View
FreeMarker
Model
Services
Content
Control
Collaboration

Webscripts

- http://localhost:8080/alfresco/service/cognizant/home
- https://docs.alfresco.com/content-services/6.0/tutorial/platform/web-scripts/

Web scripts


Select
Create...
Upload
Selected Items...
Documents
I'm Editing
My Favorites
Repository
Repository
Data Dictionary
Email Templates
Imap Configs
Messages
Models
Node Templates
Presentation Templates
Rendering Actions Space
Replication Actions Space
RSS Templates
Saved Searches
Scheduled Actions
Scripts
Smart Folder Downloads
Smart Folder Templates
Solr Facets Space
Space Templates
Transfers
Web Client Extension
Web Scripts
org
Web Scripts Extensions
Workflow Definitions
Repository > Data Dictionary > Web Scripts
org
Created 18 days ago by System User
No Description
No Tags
Favorite
Like 0
Comment
readme_de.html
Created 18 days ago by System User 1 KB
What are Web Scripts and how do you develop them?
No Tags
Favorite
Like 0
Comment
Share
readme_fr.html 1.0
Created 18 days ago by System User 1 KB
What are Web Scripts and how do you develop them?
No Tags
Favorite
Like 0
Comment
Share
readme_ja.html
Created 18 days ago by System User 1 KB
What are Web Scripts and how do you develop them?
No Tags
Favorite
Like 0
Comment
Share
readme.html
Created 18 days ago by System User 1 KB
What are Web Scripts and how do you develop them?
No Tags
Favorite
Like 0
Comment
Share

Creating a web script using cache controls

- Log in to Alfresco Share:
- Open a web browser and enter the URL: http://localhost:8080/share
- If prompted, log in with the user name admin and password admin.
- Navigate to Data Dictionary > Web Scripts Extensions > org > example.
  - Create a web script description document for your cache sample:
  - In the Create menu, select XML.
  - Enter the name for the web script in the Name field: cache.get.desc.xml

Creating a web script using cache controls

3. Type the following in the content box:

```
<webscript>
  <shortname>Cache example</shortname>
  <description>Demonstrate cache controls</description>
  <url>/cache</url>
  <authentication>user</authentication>
  <cache>
    <never>false</never>
    <mustrevalidate/>
  </cache>
</webscript>
```

4. Click **Create**.

5. Navigate back to the `org/example` folder using the breadcrumb trail.

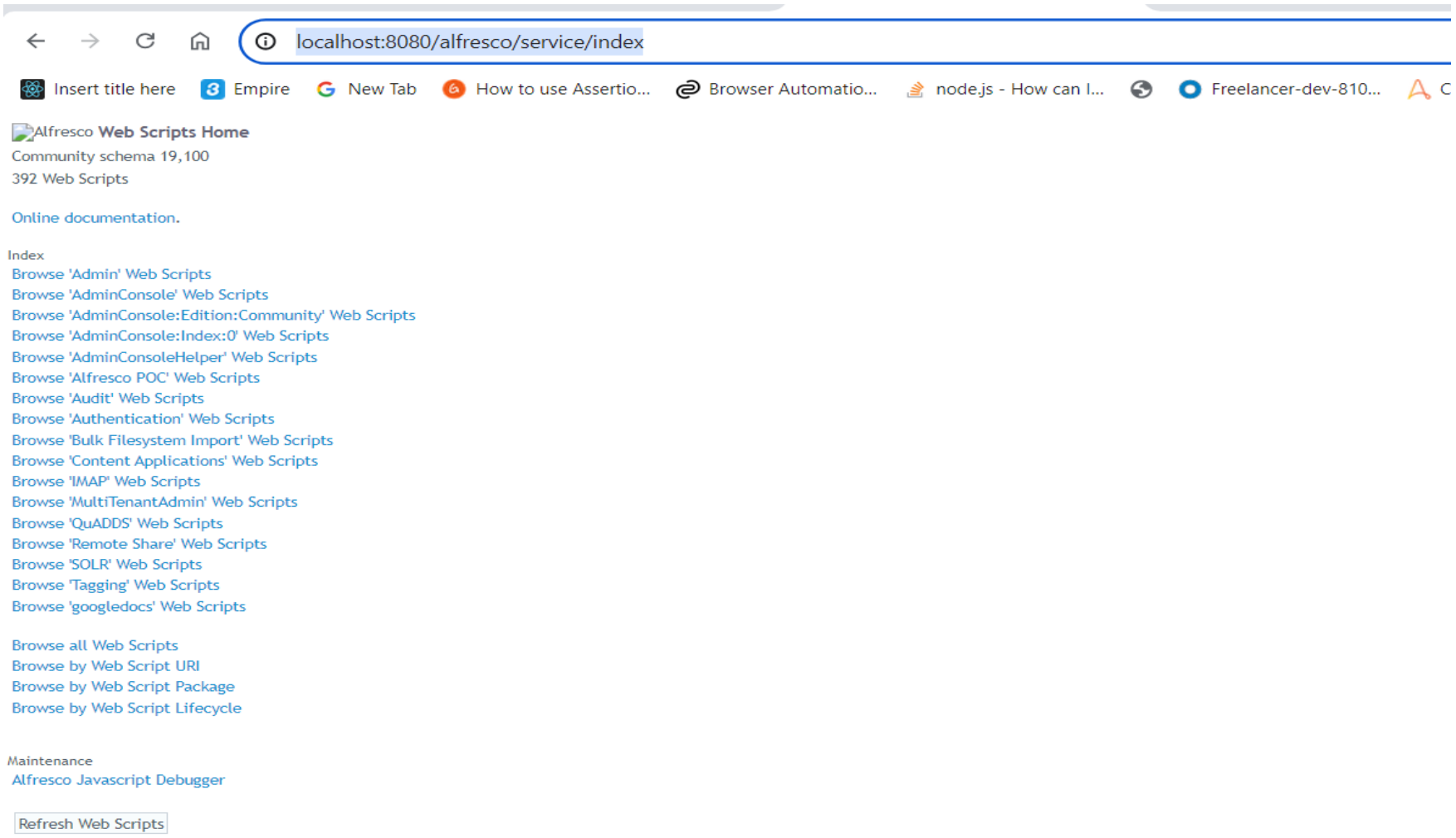
Creating a web script using cache controls

4. Create a controller script for your cache sample:
    1. In the Create menu, select **Plain Text**.
    2. Enter the name in the Name field: `cache.get.html.ftl`
    3. Type the following in the content box:

    ```
    Cached response
    ```

    4. Click **Create**.
    5. Navigate back to the org/example folder using the breadcrumb trail.
5. Register the web script with Content Services.
    1. Create a new browser tab.
    2. In the browser tab, enter the URL: `http://localhost:8080/alfresco/service/index`
    3. If prompted, log in with the user name `admin` and password `admin`.
    4. Click **Refresh Web Scripts**.

Creating a web script using cache controls



Creating a web script using cache controls

Alfresco **Index of All Web Scripts**
Community schema 19,100
393 Web Scripts

Back to Web Scripts Home

Package: /alfresco/demo

Home
GET /alfresco/service/cognizant/home

Launching Product Home

Authentication: user
Transaction: required
Format Style: any
Default Format: html
Id: alfresco/demo/home.get
Descriptor: classpath:alfresco/extension/templates/webscripts/alfresco/demo/home.get.desc.xml

Package: /alfresco/tutorials

Hello World Sample Webscript
GET /alfresco/service/sample/helloworld

Hands back a greeting

Authentication: user
Transaction: required
Format Style: any
Default Format: html
Lifecycle: sample
Id: alfresco/tutorials/helloworld.get
Descriptor: classpath:alfresco/extension/templates/webscripts/alfresco/tutorials/helloworld.get.desc.xml

Creating a web script using cache controls

http://localhost:8080/alfresco/service/cache

Package: /org/example

Cache example
GET /alfresco/service/cache

Demonstrate cache controls

Authentication: user
Transaction: required
Format Style: any
Default Format: html
Id: org/example/cache.get
Descriptor: workspace://SpacesStore/app:company_home/app:dictionary/cm:extensionwebscripts/org/example/cache.get.desc.xml

Creating a web script using cache controls

Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>curl -uadmin:admin -v "http://localhost:8080/alfresco/service/cache"
* Host localhost:8080 was resolved.
* IPv6: ::1
* IPv4: 127.0.0.1
*   Trying [::1]:8080...
* Connected to localhost (::1) port 8080
* Server auth using Basic with user 'admin'
> GET /alfresco/service/cache HTTP/1.1
> Host: localhost:8080
> Authorization: Basic YWRtaW46YWRtaW4=
> User-Agent: curl/8.7.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200
< Cache-Control:  must-revalidate
< Content-Type: text/html;charset=UTF-8
< Content-Length: 35
< Date: Wed, 17 Jul 2024 01:12:06 GMT
< Server:
<
<p><span>Cached response</span></p>* Connection #0 to host localhost left intact

C:\Windows\System32>
```

Processing Multi Part Forms

- Log in to Alfresco Share
- Open a web browser and enter the following URL: http://localhost:8080/share
- If prompted, log in with the user name admin and password admin.
- Click the Repository link in the Share header.
- Navigate to Data Dictionary > Web Scripts Extensions > org > example.
- Create a web script description document for your form:
- In the Create menu, select XML.
- Enter the name for the web script in the Name field: multipart.get.desc.xml

Processing Multi Part Forms

3. Type the following in the content box:

```
<webscript>
  <shortname>File Upload Sample</shortname>
  <description>Form to upload file.</description>
  <url>/multipart</url>
  <authentication>user</authentication>
</webscript>
```

4. Click **Create**.

5. Navigate back to the `org/example` folder using the breadcrumb trail.

Processing Multi Part Forms

5. Create an HTML response template for your form:
    1. In the Create menu, select **Plain Text**.
    2. Enter the name in the Name field: `multipart.get.html.ftl`
    3. Type the following in the content box:

```
<html>
  <body>
    <form action="${url.service}" method="post" enctype="multipart/form-data">
      File: <input type="file" name="file"><br>
      Title: <input name="title"><br>
      Description: <input name="description"><br>
      <input type="submit" name="submit" value="Upload">
    </form>
  </body>
</html>
```

    4. Click **Create**.
    5. Navigate back to the `org/example` folder using the breadcrumb trail.
6. Create a web script description document for your upload web script:
    1. In the Create menu, select **XML**.
    2. Enter the name in the Name field: `multipart.post.desc.xml`

Processing Multi Part Forms

3. Type the following in the content box:

```
<webscript>
  <shortname>File Upload Sample</shortname>
  <description>Handling of multipart/form-data requests.</description>
  <url>/multipart</url>
  <authentication>user</authentication>
</webscript>
```

4. Click **Create**.

5. Navigate back to the org/example folder using the breadcrumb trail.

7. Create a controller script for your upload web script:

   1. In the Create menu, select **Plain Text**.

   2. Enter the name in the Name field: `multipart.post.js`

Processing Multi Part Forms

3. Type the following in the content box:

```
// extract file attributes
var title = args.title;
var description = args.description;

// extract file
var file = null;
for each (field in formdata.fields)
{
  if (field.name == "file" && field.isFile)
  {
    file = field;
  }
}

// ensure file has been uploaded
if (file.filename == "")
{
  status.code = 400;
  status.message = "Uploaded file cannot be located";
  status.redirect = true;
}
else
{
  // create document in company home from uploaded file
  upload = companyhome.createFile(file.filename) ;
  upload.properties.content.guessMimetype(file.filename);
  upload.properties.content.write(file.content);
  upload.properties.title = title;
  upload.properties.description = description;
  upload.save();
  // setup model for response template
  model.upload = upload;
}
```

4. Click **Create**.

Processing Multi Part Forms

8. Create a response template for your upload web script:
    1. In the Create menu, select **plain Text**.
    2. Enter the name in the Name field: `multipart.post.html.ftl`
    3. Type the following in the content box:

```
<html>
  <body>
    Uploaded ${upload.name} of size ${upload.properties.content.size}.
  </body>
</html>
```

    4. Click **Create**.
    5. Navigate back to the `org/example` folder using the breadcrumb trail.
9. Register the web scripts with Alfresco.
    1. Switch to a new browser tab, enter the URL: `http://localhost:8080/alfresco/service/index`
    2. If prompted, log in with the user name `admin` and password `admin`.
    3. Click **Refresh Web Scripts**.

A message indicates there are two additional web scripts.

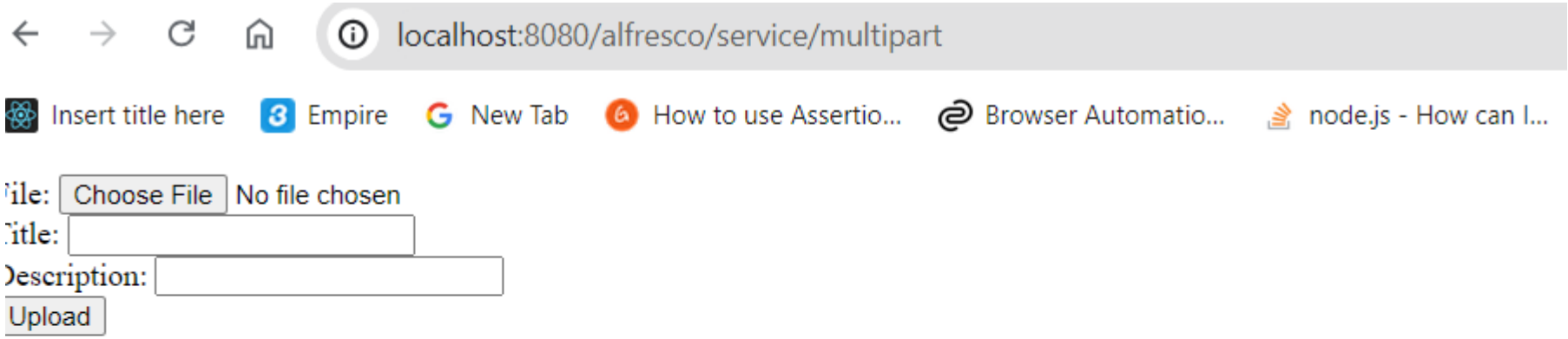
Processing Multi Part Forms

This task demonstrates how to test an upload web script.

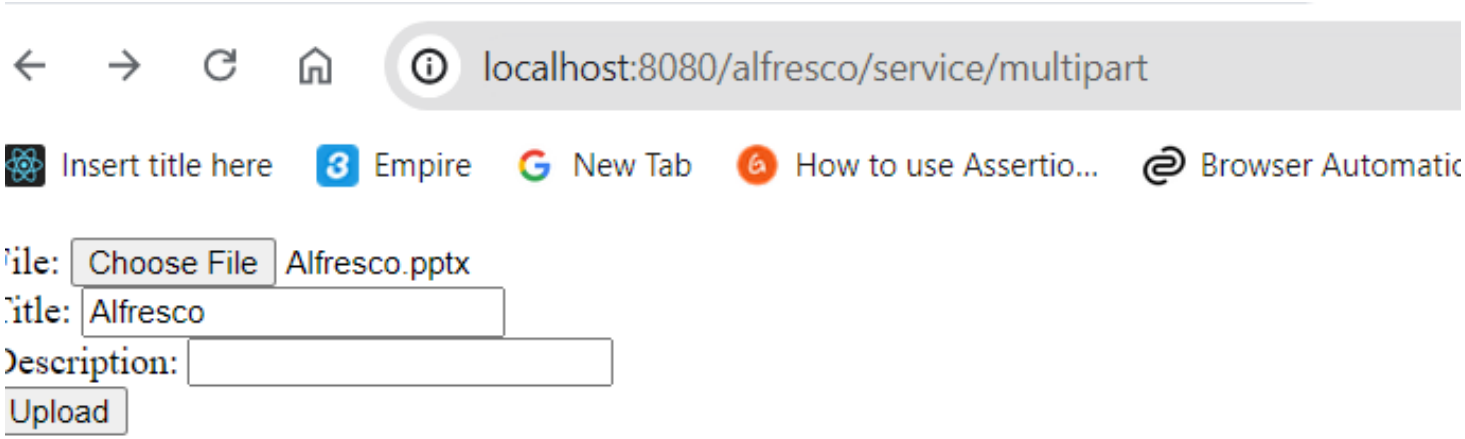
1. Launch the upload form:
    1. Open a web browser and enter the following URL: `http://localhost:8080/alfresco/service/multipart`
    2. If prompted, log in with the user name `admin` and password `admin`.
    3. Fill in the file, title, and description fields of the form.
    4. Click **Upload**.

    If you see a confirmation message detailing the name and size of the uploaded file, your web script is working.
2. Locate the created document in the repository.
    1. Open a web browser and enter the following URL: `http://localhost:8080/share`
    2. If prompted, log in with the user name `admin` and password `admin`.
    3. Click the Repository link in the Share header and locate the document whose name matches the uploaded file name.
    4. Examine the properties and content of the created document.

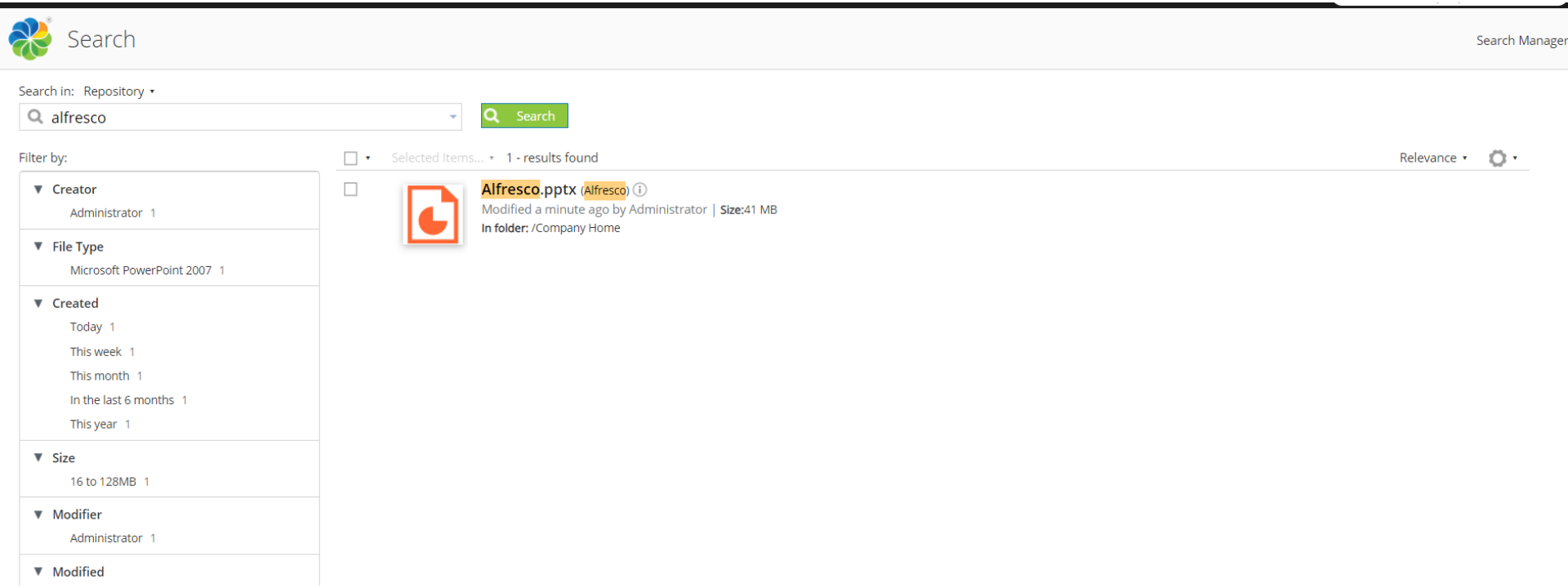
Processing Multi Part Forms



Processing Multi Part Forms



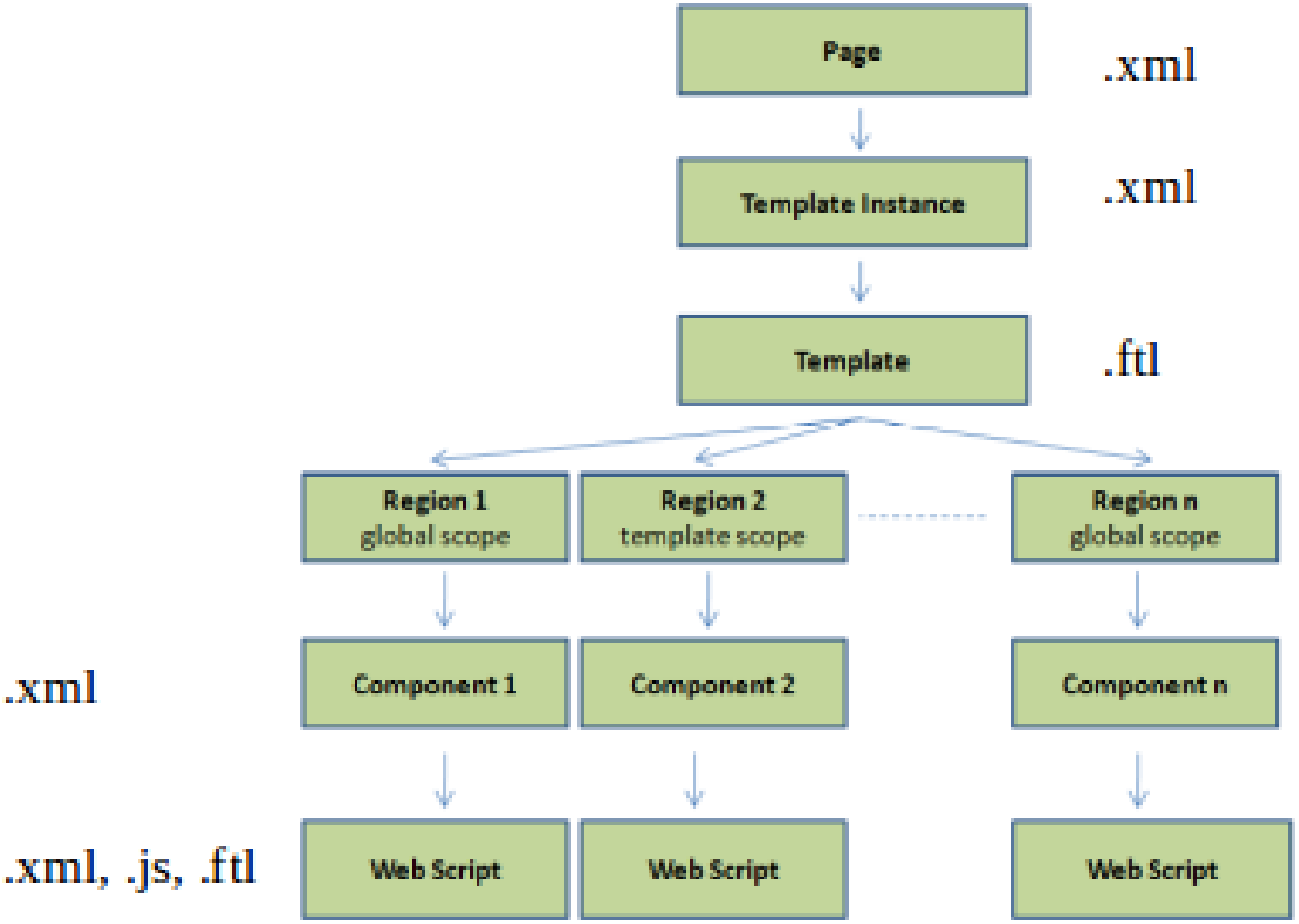
Processing Multi Part Forms



Surf Pages Extension Point

- The Alfresco Share web application is built up of a main menu from which you can navigate to several pages.
- These pages are implemented with the Surf development framework.

Surf Description



Surf Description

The definition of a Surf page↓ is done in XML and looks like this in a Hello World example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<page>
    <title>Hello World</title>
    <title-id>page.helloworld.title</title-id>
    <description>Hello World Description</description>
    <description-id>page.helloworld.description</description-id>
    <template-instance>helloworldhome-three-column</template-instance>
    <authentication>none</authentication>
</page>
```

Page definition file names follow a naming convention: `<page-id>.xml`, the above page definition could be stored in a file called `helloworld.xml` under `site-data/pages`.

The page definition refers to a template instance↓ that links to the physical template, it is defined in XML:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<template-instance>
    <template-type>org/alfresco/training/helloworld-1-column</template-type>
</template-instance>
```

Surf Description

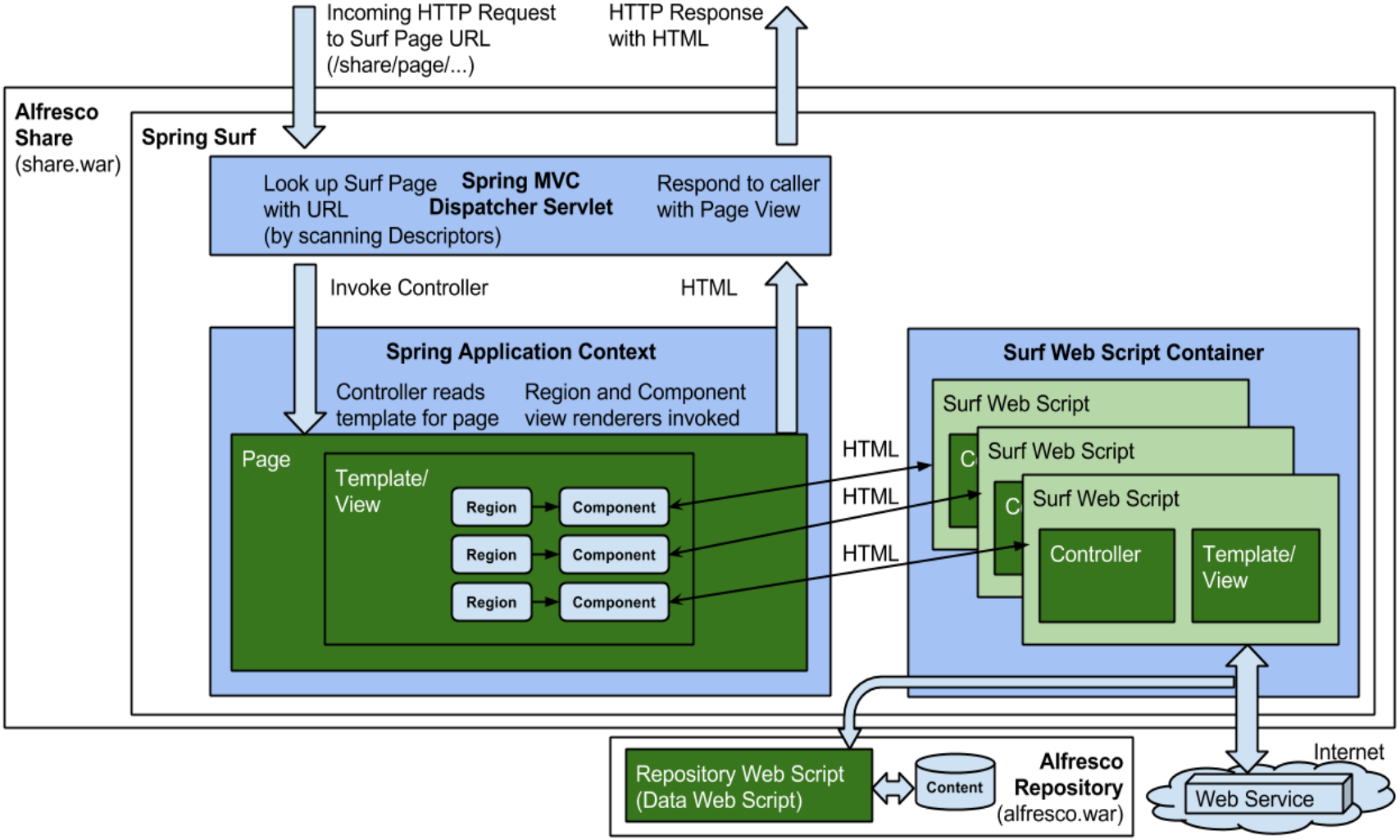
Template instance definition file names follow a naming convention: `<template-instance-id>.xml`, the above template instance definition could be stored in a file called `helloworld-1-column.xml` under `site-data/template-instances`.

The physical template is defined in FreeMarker and looks something like this:

```
<#include "/org/alfresco/include/alfresco-template.ftl" />
<@templateHeader>
</@>

<@templateBody>
   <@markup id="alf-hd">
   <div id="alf-hd">
      <@region scope="global" id="share-header" chromeless="true"/>
   </div>
   </@>
   <@markup id="bd">
    <div id="bd">
        <@region id="body" scope="page" />
    </div>
   </@>
</@>

<@templateFooter>
   <@markup id="alf-ft">
   <div id="alf-ft">
      <@region id="footer" scope="global" />
   </div>
   </@>
</@>
```

Template files are stored under `/templates`.

Surf Description



Surf Description

- https://docs.alfresco.com/content-services/6.2/tutorial/share/pages/

Surf Description

- A simple and effective way to add content to a Surf page is to follow these steps:
  - Create a Web Script that returns the content to be displayed.
  - Find a page component that is located where you want to add new content, such as the footer in our case.
  - Add a new sub-component to the page component, referencing the new Web Script.
  - Use a Surf Extension module to deploy the new sub-component.

Questions

Module Summary

- Alfresco Architecture
- Installation
- Content Model
- Configuration and Layouts
- Alfresco Share Components Customization
- Working with the new Aikau framework in Alfresco Share
- Workflows