ADVANCED SYSTEMS FOR DOCUMENT MANAGEMENT

CURS 2

Lect. Univ. Dr. Mihai Stancu

DEPARTAMENTUL DE INFORMATICĂ

ECM Architecture – Case study: Alfresco

- Repository on server persisting: content, metadata, associations, and full text indexes
- JEE application with Spring modularize functionalities such as: versioning, security, and rules.
- Client/Server architecture:
  - Web-based client: Alfresco Share application exposing main ECM functionalities
    - Can be deployed and customized on different application servers
    - Clients for portals (JSR-168), mobile, MS Office, desktop
  - content application server provides the following categories of services built upon the content repository
    - Content services (transformation, tagging, metadata extraction)
    - Control services (workflow, records management, change sets)
    - Collaboration services (social graph, activities, wiki)
    - Communication protocols: HTTP, SOAP, CIFS, FTP, WebDAV, IMAP, and Microsoft SharePoint

Alfresco architecture overview

- Supports the requirements of ECM applications, such as:
  - Document Management (DM)
  - Web Content Management (WCM)
  - Records Management (RM)
  - Digital Asset Management (DAM)
  - Search
- Guiding design principles
  - Simple, simple, simple
  - Scaling to the enterprise
  - Modular approach
  - Incorporating best-of-breed libraries
  - Environment independence
  - Solid core
  - Scriptable extensions
  - Standards-based approach
  - Architecture of participation

Alfresco Web tier and Surf

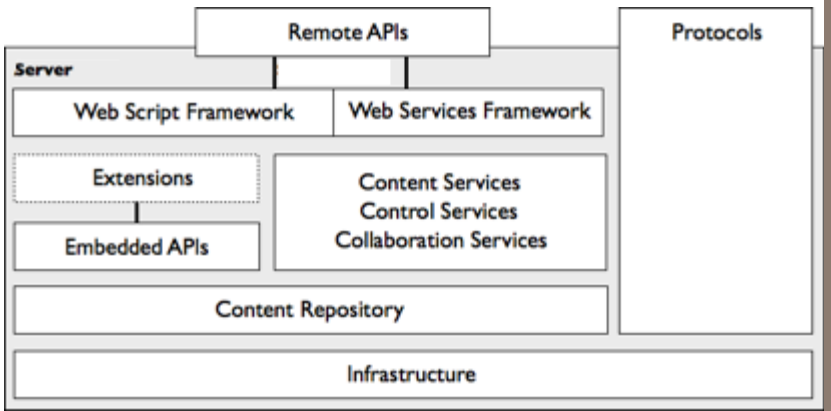
- As ECM capabilities:
  - data services
  - **user interfaces**
  - user applications
- Surf framework: a faster way to develop content applications using scripting and REST architecture
  - web scripts allows the creation of a REST-based API
  - easily extend the Alfresco standard services
  - build user interface components and deploy them as Alfresco Share components, portlets, or other web platforms such as Google Gadgets.

Alfresco Share client application

- Web-based client application that provide means of accessing the repository
  - content management capabilities (create, upload, and manage content)
  - tools to search and browse the repository
  - content such as thumbnails and associated metadata
  - renditions of content
  - a set of collaboration tools such as wikis, discussions, and blogs
- can be run on a different server (than the repository server), providing opportunities to increase scale and performance.

Alfresco Application server

- Manages and maintains the repository
- Provide services for use in building ECM solutions
  - Remote public interfaces - the only part of the server visible to the client
    - Remote APIs - for interacting with services of the server programmatically
    - Protocol bindings - for mapping services for use by a protocol-compliant client
- comprises several layers
  - cutting across all capabilities
    - configuration
    - authentication
    - permissions
    - transactions

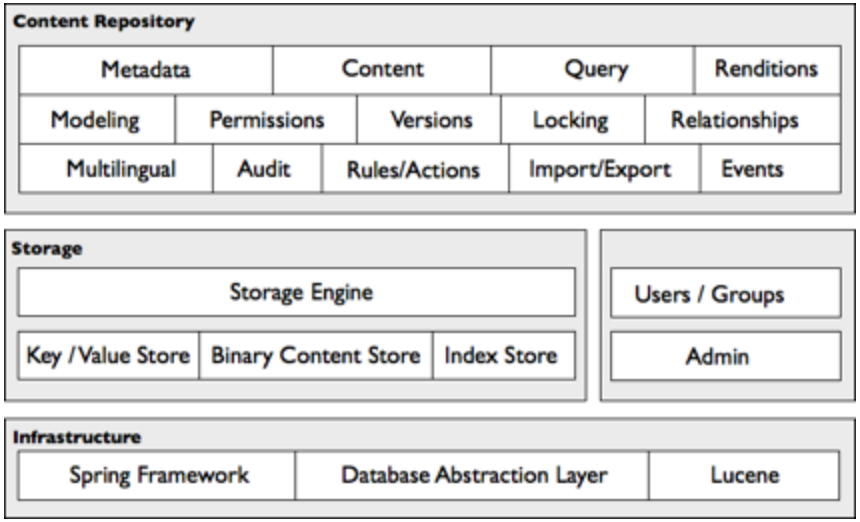


Alfresco Repository

- The repository is comparable to a database, except that it holds more than data. The binary streams of content are stored in the repository and the associated full-text indexes are maintained by SOLR indexes.
- The repository also holds the associations among content items, classifications, and the folder/file structure. The folder/file structure is maintained in the database and is not reflected in the internal file storage structure.

Alfresco Repository – 2

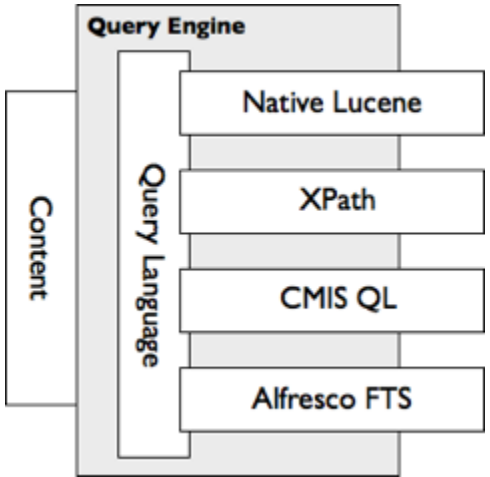
- The repository implements services including:
  - Definition of content structure (modeling)
  - Creation, modification, and deletion of content, associated metadata, and relationships
  - Query of content
  - Access control on content (permissions)
  - Versioning of content
  - Content renditions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



- The repository implements and exposes these services through an API, CMIS protocol bindings, and the JSR-170 Java API

Alfresco Repository – 3

- The storage engine of the repository stores and retrieves content, metadata, and relationships, and operates on the following constructs:
  - Nodes - provide metadata and structure to content. A node can support properties, such as author, and relate to other nodes such as folder hierarchy and annotations. Parent to child relationships are treated specially.
  - Content - the content to record, such as a Microsoft Word document or an XML fragment.
- The storage engine also exposes query capabilities provided by a custom query engine built on Apache Lucene that supports the following search constructs:
  - Metadata filtering
  - Path matching
  - Full text search
  - Any combination of these search constructs



Alfresco Content Services

- Address the core use cases for content management apps
  - logical organization of content
  - file management
  - version control
  - security
  - workflow and process management
  - social and collaborative applications
- Alfresco exposes services at various levels including:
  - Java
  - Scripting
  - REST/Web services
  - Client interfaces, such as Alfresco Share
- accessible through other public interfaces including the public APIs, client applications, and CMIS

Alfresco Programming models

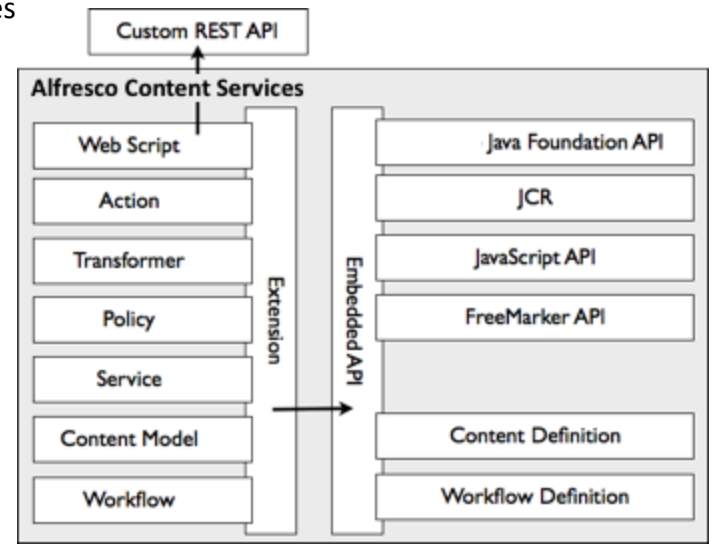
- Out-of-the-box components of the Alfresco Share
  - OOTB components and rules actions defined by a wizard to perform actions such: converting content, moving content, or executing a simple JavaScript snippet
- Web scripts
  - Using JavaScript to build new services or extensions to Share
  - Can use FreeMarker template language to define these scripts
- Extending Share using Java
  - Trough Surf, the Web Runtime Framework (also include web scripts)
  - Directly on Spring platform (open source code as support for your custom solutions)
- Content Management Interoperability Services (CMIS), the OASIS standard for accessing content repositories

Alfresco – API's

- Embedded APIs: used for developing extensions to the application server
- Remote APIs: primarily used to build ECM solutions against the content application server

Alfresco – Embedded APIs

- Extensions depend on existing services therefore, we must use the Embedded API to gain access to those service
- Embedded API comes in several forms
  - **Alfresco Public Java API** - a set of public Java interfaces exposed by services built into the content application server
  - **JavaScript API** - an object-oriented view of the Java Foundation API specifically tailored for use in JavaScript. There is a JavaScript API for the repository tier and a JavaScript API for the Share tier.
  - **FreeMarker API** - an object-oriented view of the Java Foundation API specifically tailored for use in FreeMarker templates
  - **Content Definition** - an API for creating and editing content models
  - **Workflow Definition** - an API for defining business processes



- The JavaScript and Template APIs are the key building blocks for web scripts to develop the RESTful APIs.

Alfresco – Remote APIs

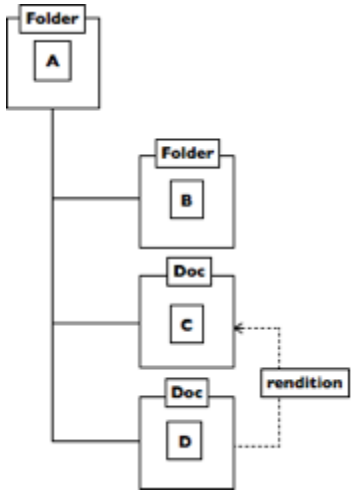
- used to build ECM solutions against the content application server
- There are three main remote APIs:
  - Alfresco Community Edition API
    - the main remote API, the recommended API for developing remote client applications to work across cloud, on-premise and hybrid deployments
    - the Mobile SDK for Android and the Mobile SDK for iOS both use the services of the Alfresco Community Edition API.
  - CMIS API
    - provides a standardized set of common services for working with content repositories
    - is not language-specific, it does not dictate how a repository works, and it does not seek to incorporate every feature of every repository
  - Repository REST API (Deprecated)
    - provides access to the core repository functionality using a RESTful approach

Alfresco – Content modeling

- Content modeling specifies how nodes stored in the repository are constrained, imposing a formal structure on nodes that an application can understand and enforce. Nodes can represent anything stored in the repository, such as folders, documents, XML fragments, renditions, collaboration sites, and people
- Each node has a unique ID and is a container for any number of named properties, where property values can be of any data type, single or multi-valued.
- Nodes are related to each other through relationships. A parent/child relationship represents a hierarchy of nodes where child nodes cannot outlive their parent. You can also create arbitrary relationships between nodes and define different types of nodes and relationships.
- A content model defines how a node in the repository is constrained. Each model defines one or more types, where a type enumerates the properties and relationships that a node of that type can support. Often, concepts that cross multiple types of node must be modeled, which the repository supports through aspects. Although a node can only be of a single type, you can apply any number of aspects to a node. An aspect can encapsulate both data and process, providing a flexible tool for modeling content.

Alfresco – Content modeling 2

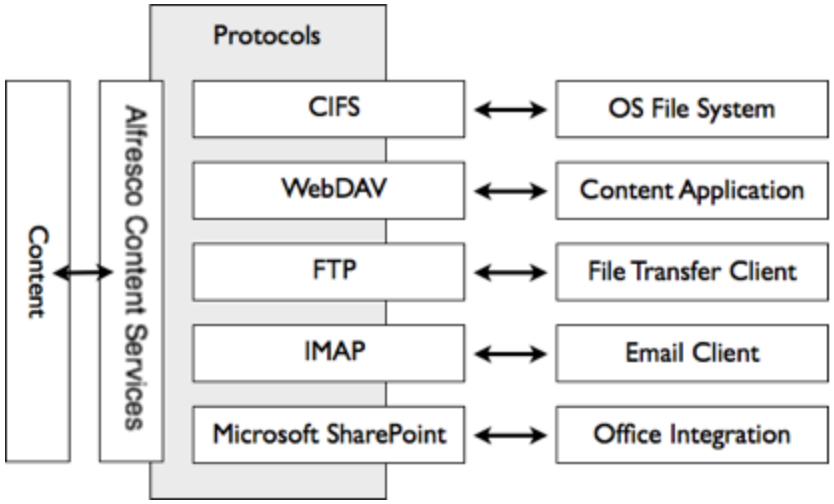
- Content modeling puts the following constraints on the data structure:
  - A node must be of a given kind.
  - A node must carry an enumerated set of properties.
  - A property must be of a given data type.
  - A value must be within a defined set of values.
  - A node must be related to other nodes in a particular way.



- These constraints allow the definition (or modeling) of entities within the domain. For example, many applications are built around the notion of folders and documents. It is content modeling that adds meaning to the node data structure.
- The repository provides services for reading, querying, and maintaining nodes. Events are fired on changes, allowing for processes to be triggered. In particular, the repository provides the following capabilities based on events:
  - Policies: event handlers registered for specific kinds of node events for either all nodes or nodes of a specific type
  - Rules: declarative definition of processes based on addition, update, or removal of nodes (for example, the equivalent of email rules)
- A special data type called content allows a property to hold arbitrary length binary data.

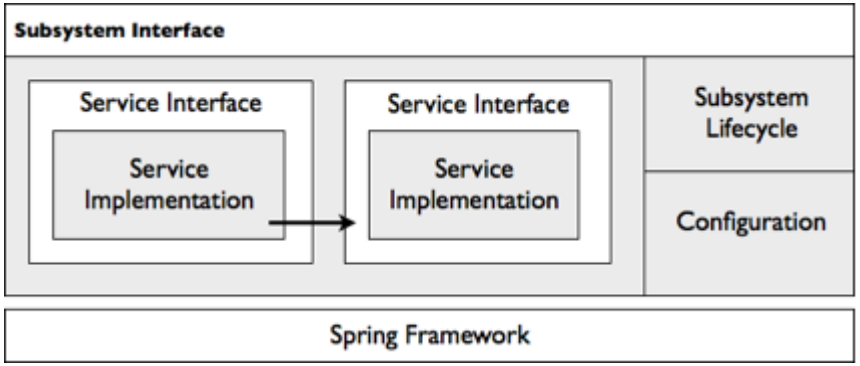
Alfresco – Protocols

- CIFS (Common Internet File System)
- WebDAV (Web-based Distributed Authoring and Versioning)
- FTP (File Transfer Protocol)
- IMAP (Internet Message Access Protocol)
- Microsoft SharePoint Protocols



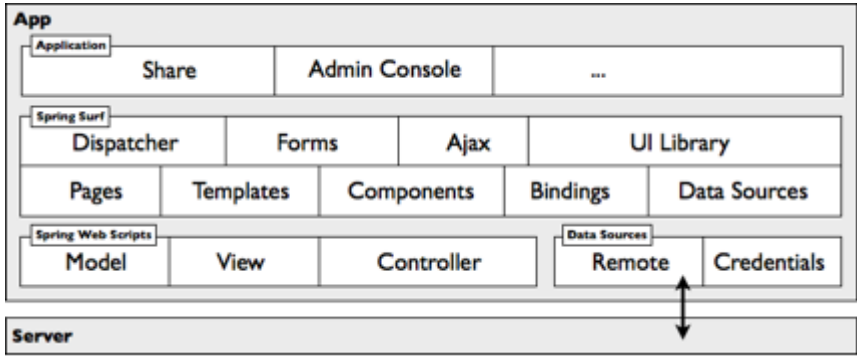
Alfresco – Modularity

- Pick and mix of services for building an ECM solution
- Reimplementation of individual services
- Multiple implementations of a service, where the appropriate implementation is chosen based on the context within which the solution is executed
- A pattern for extending Alfresco Community Edition (at design and runtime)
- Easier testing of services



Alfresco – Web application framework

- Surf – provides the typical features of this kind of framework and supports web content management needs
- At the heart of Surf is a site assembly framework that bundles a full site construction object model and toolkit for building websites and applications
  - A Site Dispatcher to create pages easily, link them to the overall navigation of a website, and build pages in a way that promotes reusability.
  - Templates for defining a page layout once and then reusing it across a large set of pages. You can develop pages using FreeMarker, JSP, HTML, or Java.
  - A UI Library containing reusable UI components comprising back-end application logic and front-end presentation code that can be bound into regions (or slots) within a page or template.
  - Pages that you can render in multiple formats, such as print, PDF, or mobile device.
  - AJAX support for integration with the Yahoo! User Interface (YUI) library.
  - Forms in a rich forms engine for rendering and collecting data.



  - Surf provides the following integration services:
    - Remote: encapsulates any number of data sources with out-of-the-box support for the content application server
    - Credentials: manages user authentication with out-of-the-box support for the content application server

Alfresco – Deployment options

- Embedded Alfresco Community Edition
- Content application server
- Multi-tenancy