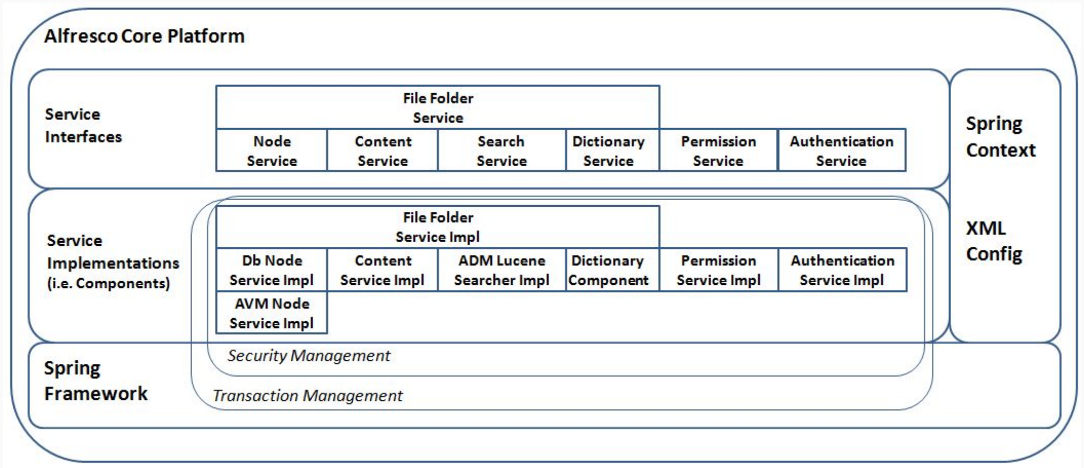
Alfresco Repository

Components and Services

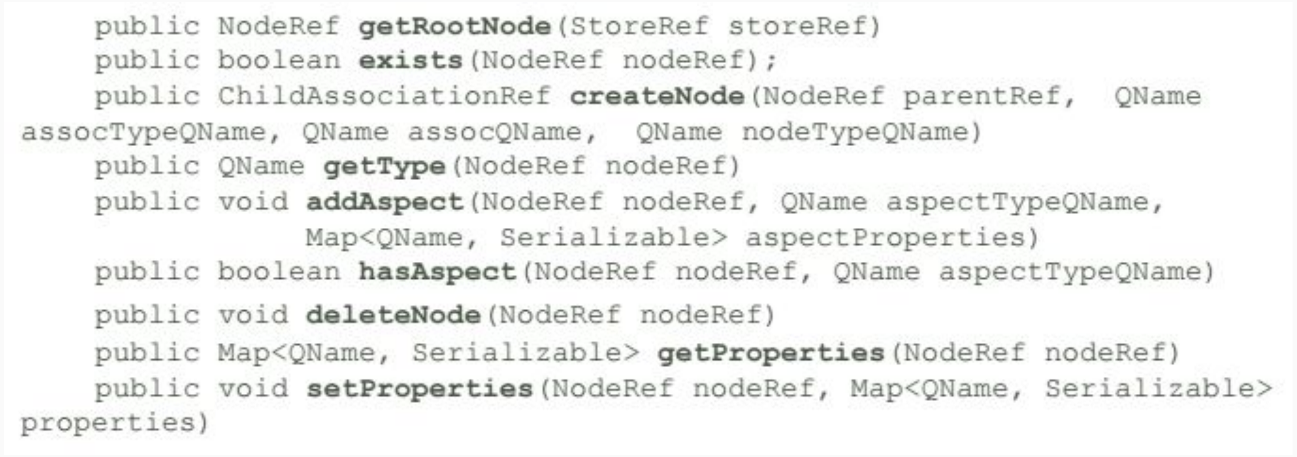
Overview

- Java
- Dependency Injection (DI) architecture using the Spring Framework
- Design by interfaces
- Implementations called components or *Impl
- Aspects to support transactions and security



Node Service

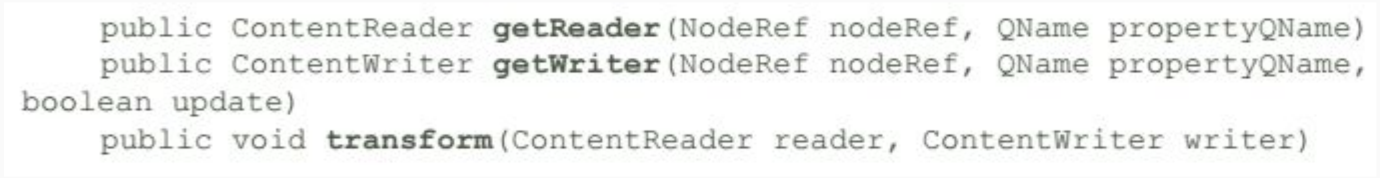
- The main foundation service for manipulating a node
- Interface has methods as follows:



```
    public NodeRef getRootNode(StoreRef storeRef)
    public boolean exists(NodeRef nodeRef);
    public ChildAssociationRef createNode(NodeRef parentRef,  QName
assocTypeQName, QName assocQName,  QName nodeTypeQName)
    public QName getType(NodeRef nodeRef)
    public void addAspect(NodeRef nodeRef, QName aspectTypeQName,
                Map<QName, Serializable> aspectProperties)
    public boolean hasAspect(NodeRef nodeRef, QName aspectTypeQName)
    public void deleteNode(NodeRef nodeRef)
    public Map<QName, Serializable> getProperties(NodeRef nodeRef)
    public void setProperties(NodeRef nodeRef, Map<QName, Serializable>
properties)
```

Content Service

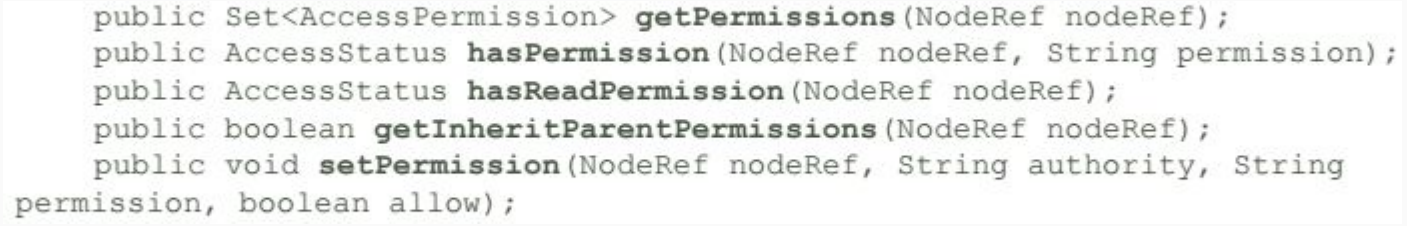
- The service to use when managing physical content for a node
- Interface has methods as follows:



```
    public ContentReader getReader(NodeRef nodeRef, QName propertyQName)
    public ContentWriter getWriter(NodeRef nodeRef, QName propertyQName,
boolean update)
    public void transform(ContentReader reader, ContentWriter writer)
```

Permission Service

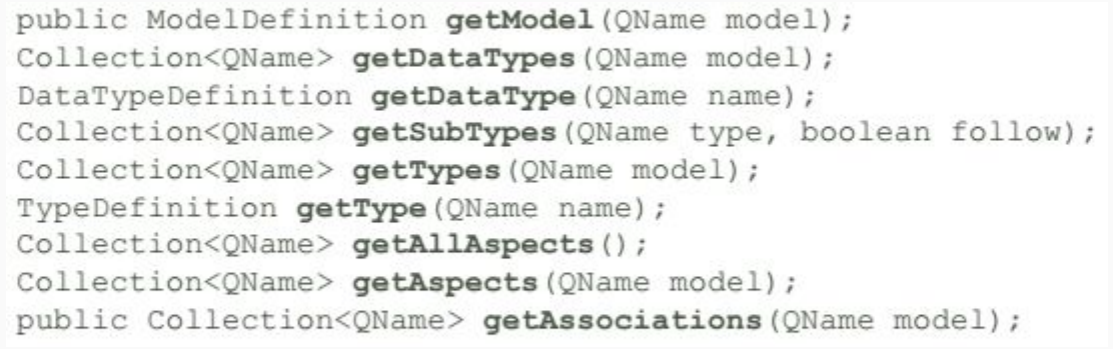
- The service to use when managing permissions for a node
- Interface has methods as follows:



```
      public Set<AccessPermission> getPermissions(NodeRef nodeRef);
      public AccessStatus hasPermission(NodeRef nodeRef, String permission);
      public AccessStatus hasReadPermission(NodeRef nodeRef);
      public boolean getInheritParentPermissions(NodeRef nodeRef);
      public void setPermission(NodeRef nodeRef, String authority, String
permission, boolean allow);
```

Dictionary Service

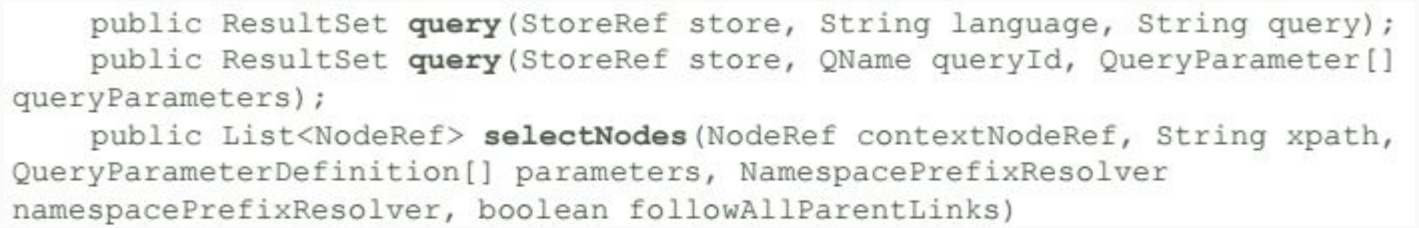
- The service to use when you want to find out stuff about the installed content models
- Interface has methods as follows:



```
public ModelDefinition getModel(QName model);
Collection<QName> getDataTypes(QName model);
DataTypeDefinition getDataType(QName name);
Collection<QName> getSubTypes(QName type, boolean follow);
Collection<QName> getTypes(QName model);
TypeDefinition getType(QName name);
Collection<QName> getAllAspects();
Collection<QName> getAspects(QName model);
public Collection<QName> getAssociations(QName model);
```

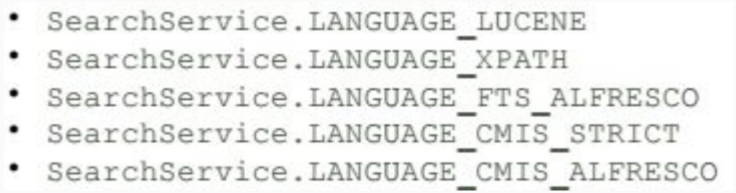
Search Service

- This service is used when searching the repository
- Interface has methods as follows:



```
    public ResultSet query(StoreRef store, String language, String query);
    public ResultSet query(StoreRef store, QName queryId, QueryParameter[]
queryParameters);
    public List<NodeRef> selectNodes(NodeRef contextNodeRef, String xpath,
QueryParameterDefinition[] parameters, NamespacePrefixResolver
namespacePrefixResolver, boolean followAllParentLinks)
```

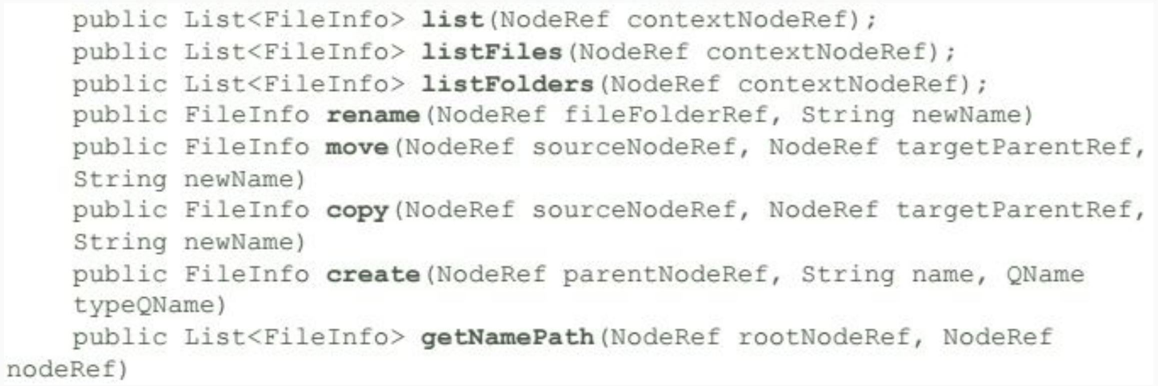
- Languages:



- `SearchService.LANGUAGE_LUCENE`
- `SearchService.LANGUAGE_XPATH`
- `SearchService.LANGUAGE_FTS_ALFRESCO`
- `SearchService.LANGUAGE_CMIS_STRICT`
- `SearchService.LANGUAGE_CMIS_ALFRESCO`

FileFolder Service

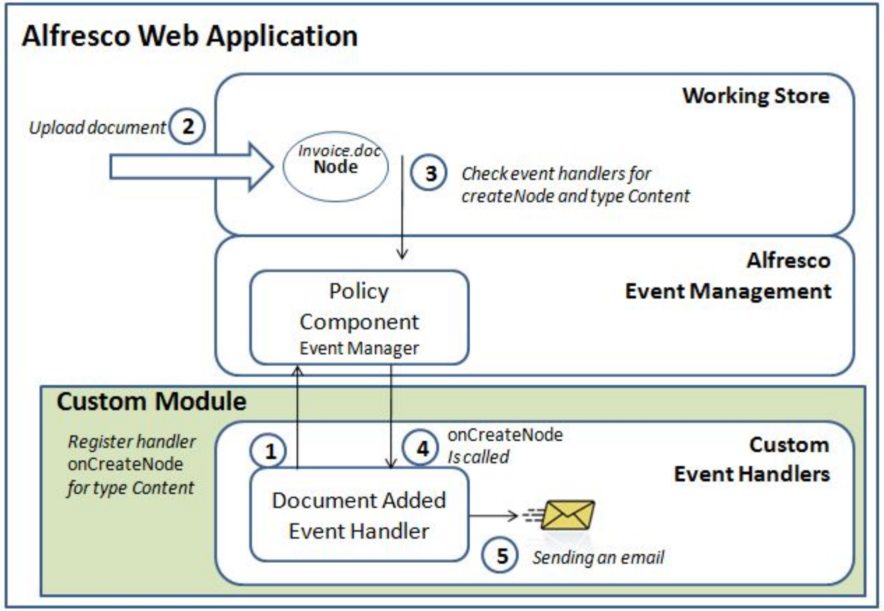
- This service is a simplification of some of the other services for working with folders and files
- Interface has methods as follows:



```
    public List<FileInfo> list(NodeRef contextNodeRef);
    public List<FileInfo> listFiles(NodeRef contextNodeRef);
    public List<FileInfo> listFolders(NodeRef contextNodeRef);
    public FileInfo rename(NodeRef fileFolderRef, String newName)
    public FileInfo move(NodeRef sourceNodeRef, NodeRef targetParentRef,
    String newName)
    public FileInfo copy(NodeRef sourceNodeRef, NodeRef targetParentRef,
    String newName)
    public FileInfo create(NodeRef parentNodeRef, String name, QName
    typeQName)
    public List<FileInfo> getNamePath(NodeRef rootNodeRef, NodeRef
nodeRef)
```

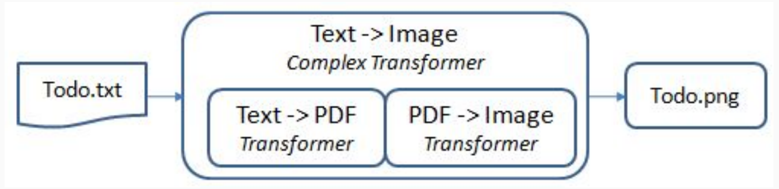
Event Listeners

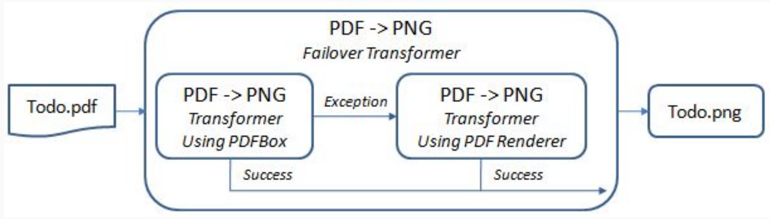
- Sometimes using rules might not cut it
- More fine grain control might be needed
- Then you can use the provided Event Model
- Event listeners can be plugged into any of the foundation services to listen to CRUD operations and call your code



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





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
    - Subsystem configuration is located in the tomcat\webapps\alfresco\WEB-INF\classes\alfresco\subsystems directory (you have to open up the tomcat\webapps\alfresco\WEB-INF\lib\alfresco-repository-<version>.jar to see this)

System Bootstrap

- If new content needs to be added to the system in a bootstrap procedure, then the following two ways can be used:
    - **Patches**, executed only once and are used for things like database upgrades, template installations, folder creations, permission updates, group imports.
        - Written in Java
        - Outcome logged in database
    - **Importers,** which are mostly used to bootstrap content.
        - Related to an Alfresco Module
        - Run once when module is first started
        - Outcome of running an importer is not logged in the database and an importer is not written in Java like a patch but XML configuration
        - Importers require special XML format

Source Code

See: https://community.alfresco.com/docs/DOC-4874-source-code