



Content Repository - Databases for Content & Other Thoughts on Persistence

Jean Barmash
Director, Technical Services
jbarmash@alfresco.com

www.alfresco.com

DevNexus 2009, Atlanta, Mar 10, 2009



About Me / Alfresco

- Director, Technical Services
 - Blog – www.nywebguy.com
 - Live in New York
- Alfresco Software, www.alfresco.com
 - Open Source Enterprise Content Management
 - Document Management
 - Web Content Management
 - Collaboration – “Open Source SharePoint”
 - Social Computing

What are some ways we store information?

Systems for Storing Information

- Databases
- File Systems
- Directories
- **Content Repositories**
- Amazon's S3
- Amazon's SimpleDB
- Google's BigTable
- Document-Oriented Databases
- RDF Triple Stores
- XML Databases
- Object Oriented Databases
- Time-Based DBs
- Data Cache Grid

Agenda

- Content Repository Abstraction – Files + Data
- Content-Oriented Applications
- Content as a Service
- CMIS - Content Management Interoperability Services
- Other Persistence Ideas

What Capabilities do We Need to
Store Content files?

Content Oriented Apps Examples

- Manage Outsourcing Process Paperwork
- Email Archiving
- Contract Management
- Today's Websites
 - Content Pages
 - Video
 - Images
 - Audio
 - User Generated Content

Requirements

- Store Large Files
- Granular Access Control
- User Management
- Transactions
- Query / Search
- Store Multi-Value Properties
- Observation / Eventing
- Running Code
- Structured + Unstructured Data
- Versioning
- Exclusive Item Locking
- Content Lifecycle / Publishing

File Systems - Good

- Trees & Hierarchies
- Ability to Browse
- Storing Large Files
- Access Control

File Systems - Bad

- No Query / Search
- No Versioning
- No Observation of Files
- No Exclusive Locking
- Limited metadata – no modeling ability
- No Transactions
- No Ability to Run Code

Databases - Good

- Query / Search
- Observation / Eventing (Triggers)
- Transactions
- Ability to Model Data

Databases - Bad

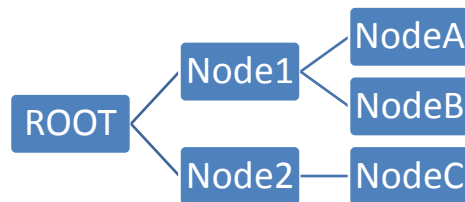
- Multi-Value Properties
- No Versioning
- BLOBs – for storing Large Files
- Trees & Hierarchies
- Access Control
- Only structured content
- Can't Browse

Content Repository Requirements

	File Systems	Databases
Hierarchies	Y	
Store Large Files	Y	
Granular Access Control	Y	
User Management		Y
Transactions		Y
Query / Search		Y / N
Observation / Eventing		Y
Running Code		Y
Data Modeling		Y
Store Structured Data		Y
Store Unstructured Data	Y	
Versioning		
Exclusive Item Locking	Y / N	
Content Lifecycle / Publishing		

Content Repository Abstraction

- One or more Workspaces



- Each workspace - Tree of Nodes
- Node has multiple Properties
 - Data is stored in properties

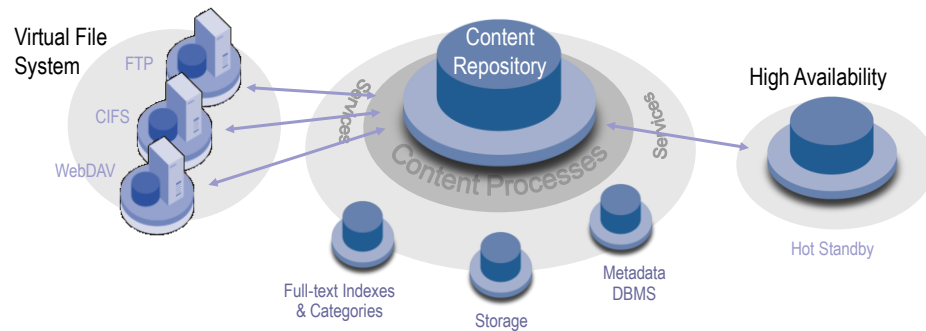
Content Modeling

- Content Types
 - Properties
 - Inheritance
- Mix-Ins / Aspects
- Associations
 - Child Associations
 - Peer Associations
- Data Integrity Checks

Review – Content Repository Features

- Store Large Files
- Granular Access Control
- User Management
- Transactions
- Query / Search
- Observation / Eventing
- Running Code
- Structured + Unstructured Data
- Versioning
- Exclusive Item Locking
- Content Lifecycle / Publishing

Content Repository



Content Repository is a great abstraction. But to enable content oriented applications, we want more.

Content as a Service

- Additional Services to act on Content
- Embed Easily into your app
- OR
- Interact with content over HTTP

Additional Features

- Content Services
- Business Processes
- Collaboration Features
- Social Networking Features
- Deployment

Content Services

- Content Transformations, i.e. doc to pdf
- Plain Text Searching
- Multi-Format Publishing / Renditioning
- Metadata Extraction
- Library Services
 - Taxonomy
 - Classification
- Image Manipulation / Thumbnails

Business Processes

- Task Management
- Workflow
- Scheduled Processes
- Auditing / Compliance
- Business Process
- Lifecycle Management
- Membership Management

Collaboration

- Pre-Built Functionality / Widgets
- Email Integration
- Social Networking
- Templated Spaces / Micro-Sites
- In-Context Editing

Social Computing Services

- Activities
- Tagging
- Site
- Thumbnailing

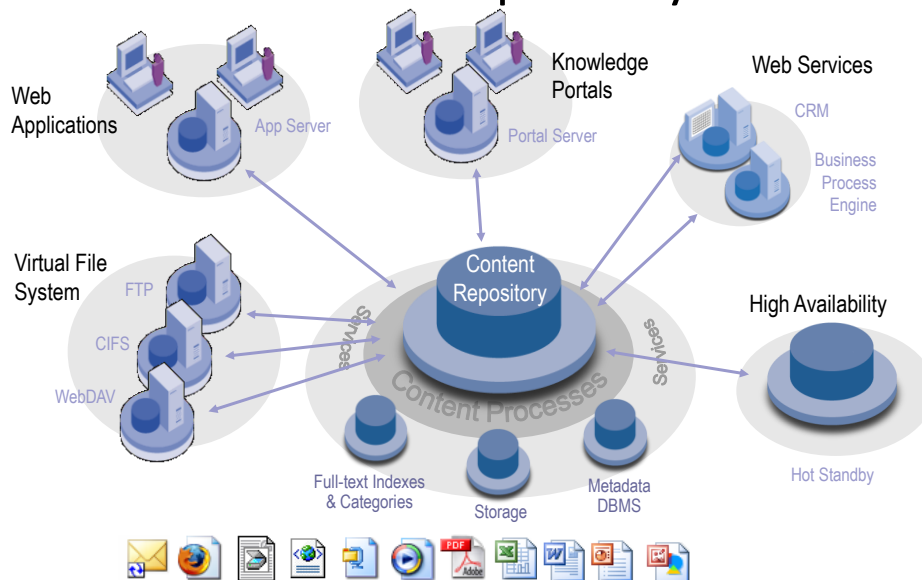
Content Repository Deployment

- Scalability
- High Availability
- User Directory Integration
- Backup / Recovery

Other Features

- Multiple environments (dev, staging, prod)
- Specialized Versioning (i.e. Subversion-like)
- Scheduled Publishing / Expiration
- Extensibility and Integration
- Internationalization / Localization
- Digital Rights Management
- Ability to Expose Data Model to other applications

Content Repository



What is CMIS?

- “The objective of the CMIS standard is to define a common content management web services interface that can be implemented by content repositories and enable interoperability across repositories.”
- A (draft) standard defining APIs to support interoperability with ECM systems
- CMIS defines:
 - Model e.g. Types, Relationships
 - Standardised Query Language
 - Protocol Bindings e.g. REST, Web Services
 - Services e.g. Check out/in, versioning

Why CMIS?

- Most large organisations have multiple ECM solutions
- No standard across ECM systems
 - Proprietary specific APIs
 - Proprietary Query interfaces
 - Language dependent Java vs .Net ...
- One-off integrations
 - No reuse
 - Expensive to implement, maintain

Target Use Cases

- | | |
|--|---|
| <ul style="list-style-type: none"> • Collaborative Content Creation <ul style="list-style-type: none"> – Authentication, Checkin/out, Version Control • Portals <ul style="list-style-type: none"> – Browsing, properties, indexing, search • Mashups <ul style="list-style-type: none"> – URL addressability, properties | <ul style="list-style-type: none"> • Archival Applications <ul style="list-style-type: none"> – Properties, indexing and search • Compound Documents <ul style="list-style-type: none"> – Relationships • Electronic Legal Discovery <ul style="list-style-type: none"> – Versioning, properties, indexing, search |
|--|---|

Non-Target Use Cases

- *Maybe* addressed in future CMIS versions
- Records Management & Compliance
 - Retention schedules, classification, legal holds
- Digital Asset Management
 - Renditions, streaming
- Web Content Management
 - Templates, staging, preview, deployment . . .
- Subscription/Notification Services
 - Event triggers

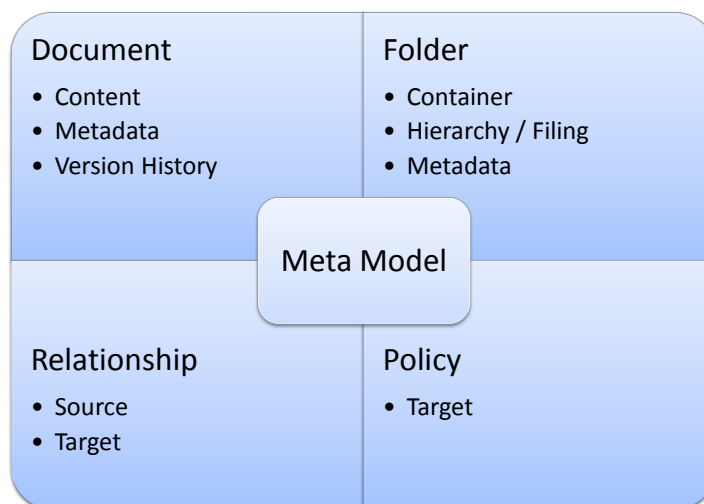
CMIS and Other Standards

- Why not using an existing standard?
- JCR-170/283
 - Java Only
 - Too prescriptive
 - Requires changes to core ECM capabilities to support specific features and models
 - Not service oriented
 - Requires persistent connections
 - Unsuitable to Mashups
- WebDAV
 - No types and properties
 - No Query
 - No relationships
 - Tied to HTTP
- Atom Publishing Protocol (APP)
 - HTTP and resource specific
 - Note: CMIS builds on APP conventions

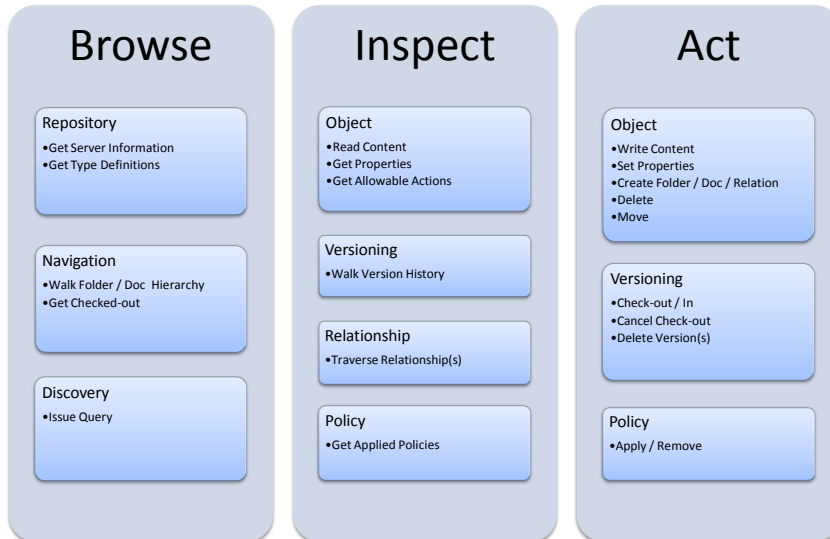
Specification Overview

- Part I - Encapsulates ECM experience
 - Defines Domain Model
 - Defines Services i.e. interaction with Model
 - Common to ECM repositories
- Part II – Map Part I to Protocol Bindings
 - SOAP / WSDL
 - Leverage years of investment in infrastructure/tools
 - Service-oriented
 - Content Repository orchestration
 - REST
 - “Web 2.0” stack
 - Resource-oriented
 - Content syndication / publishing

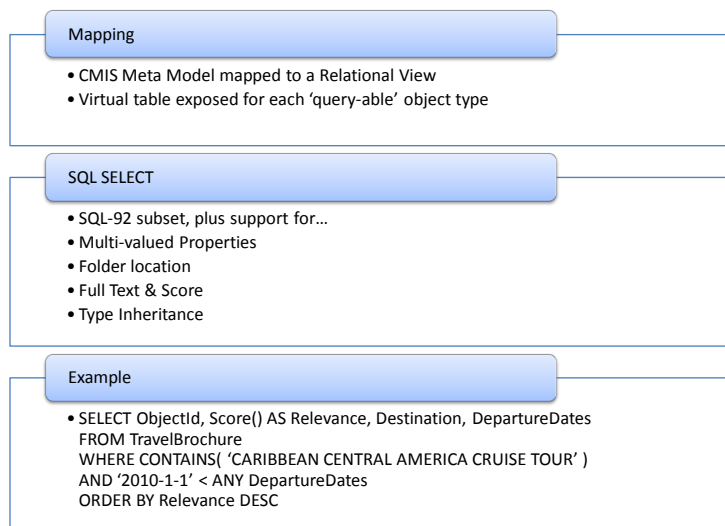
CMIS Domain Model



CMIS Services



CMIS Query



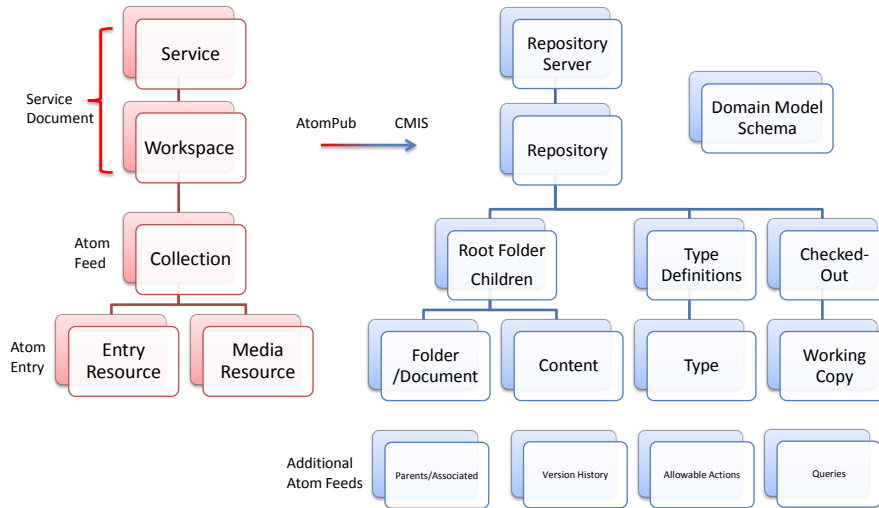
CMIS SOAP Binding

- WSDL definition...
 - XML schema for CMIS Domain Model
 - XML schema for Service messages
 - Direct exposure of CMIS (Part I) Services
 - Generate client API for almost all languages
- WS-Security & Username Token Profile (MUST)
- WS-I Basic Profile & Basic Security Profile
- MTOM content transfers

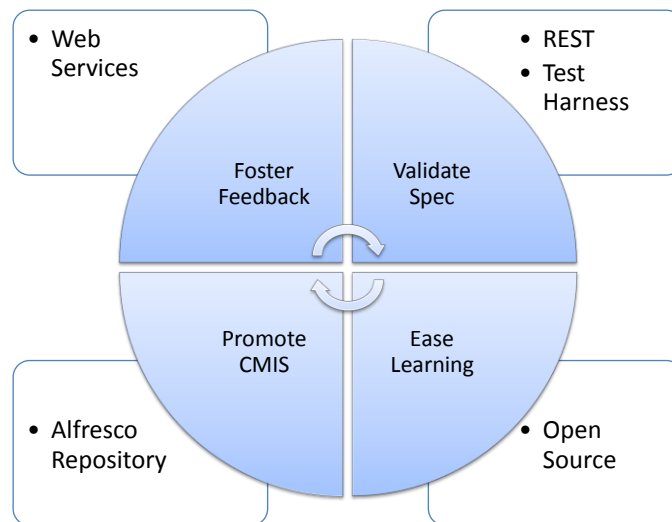
CMIS REST Binding

- ATOM Publishing Protocol
 - ATOM syndication format for web feeds (GET)
 - Create & update web resources (POST, PUT, DELETE)
 - Extension mechanism supported
- CMIS extension
 - XML Schema for CMIS Domain Model
 - As used in SOAP Binding
 - New Web Resources / Method mappings
- Use any existing HTTP or ATOM client API

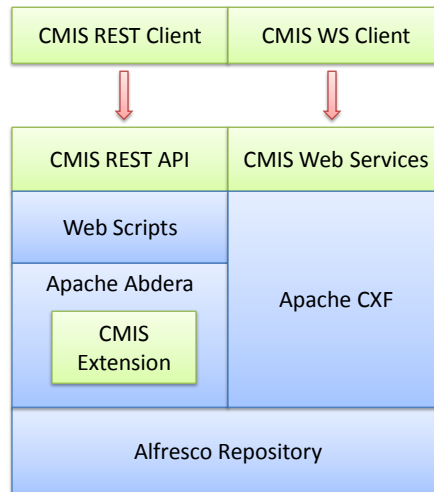
Atom Publishing Protocol to CMIS



Alfresco Draft CMIS Implementation



Alfresco Implementation Stack

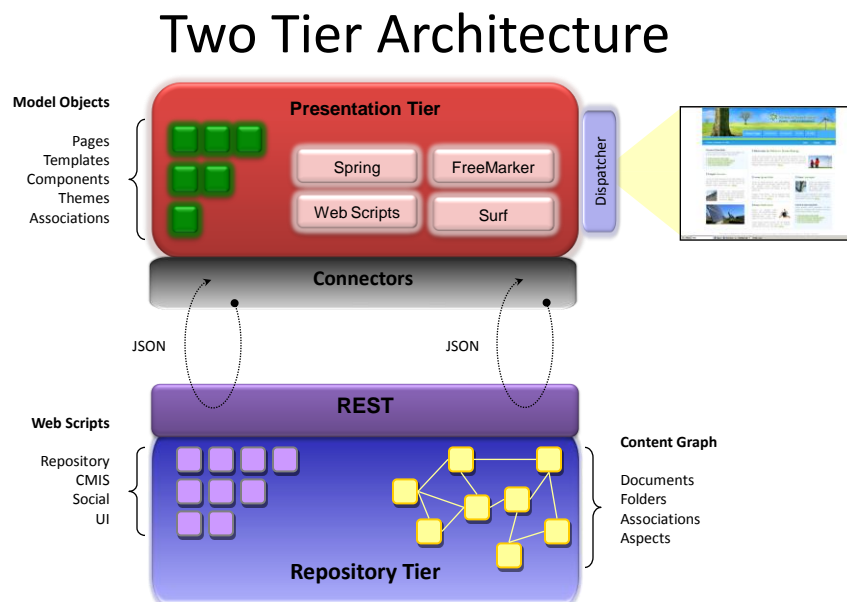


Alfresco CMIS Strategy

- Actively contribute to specification
- Continue to update draft implementation based up updates
 - Provide Open Source implementation as spec evolves
- Provide 100% compliance
- Productised CMIS Client Support and Tools

Two Tier Architecture

- Presentation Tier
 - Web Application
- Repository (Data) Tier – Content as a Service
 - Repository
 - Content Services
 - REST Interface to outside world



Applications of Enterprise Content Mgmt

- Web and Portal Content Management
- Collaborative Development
- On Demand Publishing
- Compliance
- Records Management
- Document Management
- Digital Asset Management
- Source Control

Other Implications

- Great integration platform for content
 - Wiki, Blogs, Documents, Web Content
- A content repository should client-runtime independent
 - Content as a Service in your SOA strategy
- Can expose Content Repository as Filesystem, FTP, Shared Drive

Conclusion - 1

- Databases and File Systems aren't well equipped to handle unstructured content
- Today's content explosion stretches the existing systems:
 - Content-oriented websites
 - User-generated Content
 - Scalability Requirements
 - Content Integration

Conclusion - 2

- Though the content Repository offers a great abstraction, additional services are typically needed
- Using Content Repository as a backend allows encapsulation of business logic and processes, while exposing content services to other applications.

But Wait, There is More!

Some Thoughts on Persistence

- Over the last 10 years, there have been an explosion of different ideas about how to store information
- What are some of the reasons?

Some Reasons

- Scalability Requirements Higher
 - RDBMS breaks down
 - Consistency Assumptions Revisited
- New Easy Ways to Integrate Data
 - SOAP / REST
- New Types of User Interfaces
- Cloud Computing
- Moore's Law (Always!)

No good programming model
exists yet

Some Interesting Ideas

- Distributed Key-Value Stores
 - MemcacheDB
 - HBase
- Document-Oriented DBs
 - CouchDB
- Drizzle – fork from MySQL
- Object DBs
 - Db4o, GemStone

More Interesting Ideas

- Distributed Caching
 - GigaSpaces, Oracle Coherence
- Cloud DBs
 - MongoDB
- How do you choose?

Questions?

- Alfresco, www.alfresco.com
- Open Source Enterprise Content Management
 - Content Repository
 - Document Management
 - Web Content Management
 - Collaboration
 - Enterprise 2.0 / Social Computing Platform



References

- [Content Repositories](http://www.gadgetopia.com/post/5940?rl)
- [Session on JCR – JSR 170](http://developers.sun.com/learning/javaoneonline/j1sessn.jsp?sessn=TS-4474&yr=2006&track=coreenterprise)
- <http://jcp.org/en/jsr/detail?id=170>
- <http://www.aiim.org/standards.asp?ID=29284>
- [http://en.wikipedia.org/wiki/Atom_\(standard\)](http://en.wikipedia.org/wiki/Atom_(standard))
- http://en.wikipedia.org/wiki/Content_management_system

References

- Alfresco wiki page on CMIS
 - <http://wiki.alfresco.com/wiki/CMIS>
- Download specification
 - <http://www.alfresco.com/about/cmisis/cmisis-draft-v0.5.zip>
- Try out Draft CMIS Implementation
 - Alfresco Labs 3b - <http://wiki.alfresco.com/wiki/CMIS>
- Subscribe to CMIS Blog
 - <http://blogs.alfresco.com/cmisis/>

Beyond Databases

- <http://www.metabrew.com/article/anti-rdbms-a-list-of-distributed-key-value-stores/>
- <http://www.webperformancematters.com/journal/2007/8/21/asynchronous-architectures-4.html>
- http://www.readwriteweb.com/archives/amazon_dynamo.php
- BigTable – Distributed Storage System for Structured Data - <http://labs.google.com/papers/bigtable.html>
- <http://martinfoowler.com/bliki/DatabaseThaw.html>