

# Alfresco Two-Way Sync with Apache Camel

Peter Lesty  
Technical Director - Parashift



# The Problem

Synchronisation Between Alfresco and External  
Systems

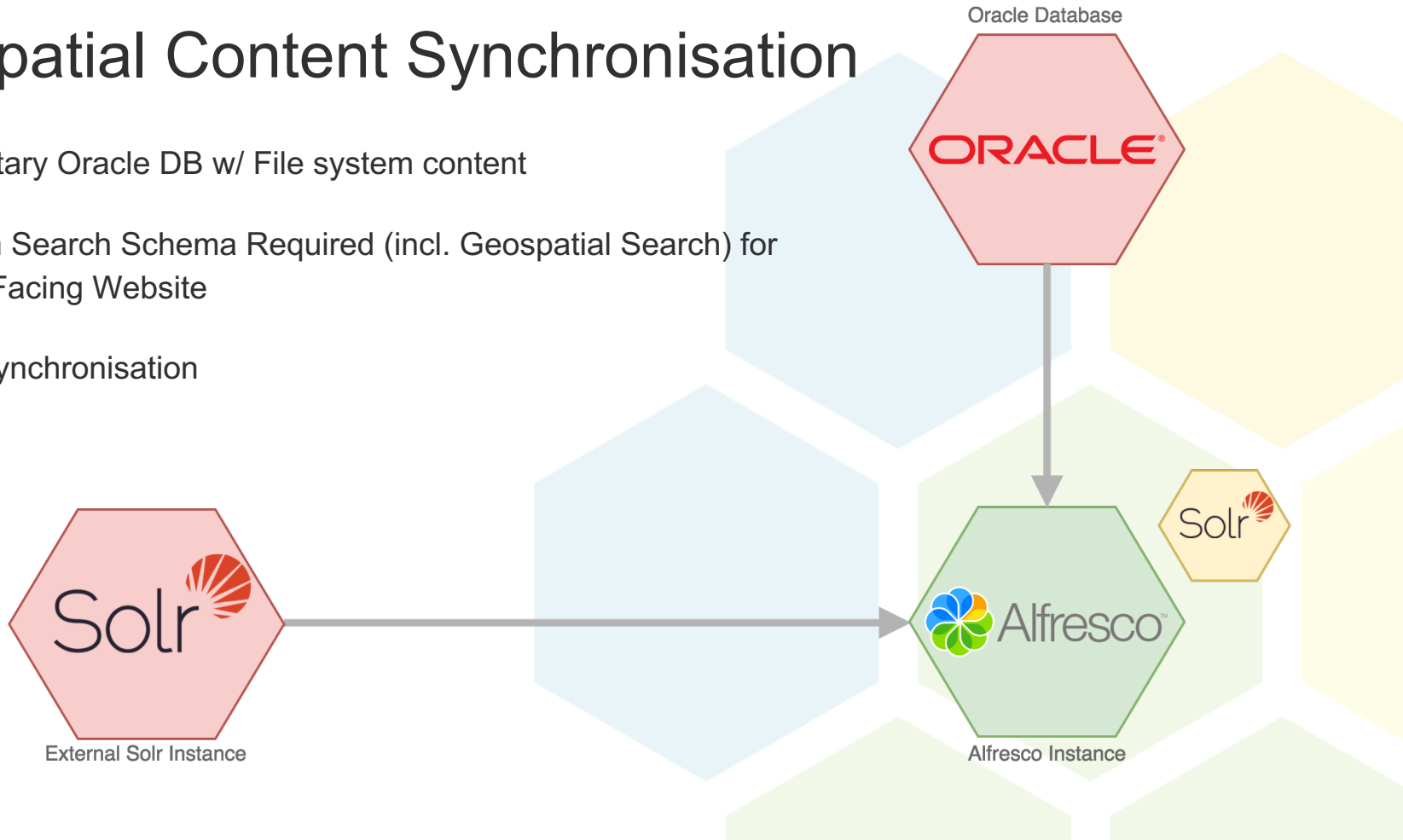
# Alfresco Two-Way Synchronisation

- Sync a selection of Nodes between Instances
- Not Limited to Folders and Files, should include Data Lists, Wikis and Forums
- Should Sync Document Locks and Permissions as well as Metadata Updates
- Network Partition Resilient: Aim for AP in CAP Theorem



# Geospatial Content Synchronisation

- Proprietary Oracle DB w/ File system content
- Custom Search Schema Required (incl. Geospatial Search) for Public Facing Website
- Daily Synchronisation



# Alfresco Sirsi Dynix Synchronisation

- Sync Nodes with Specific Aspects to Sirsi Dynix for Cataloguing
- Translate Alfresco Content Model into Marc21 Fields
- Report back any Sync-Related Errors and Update Reference



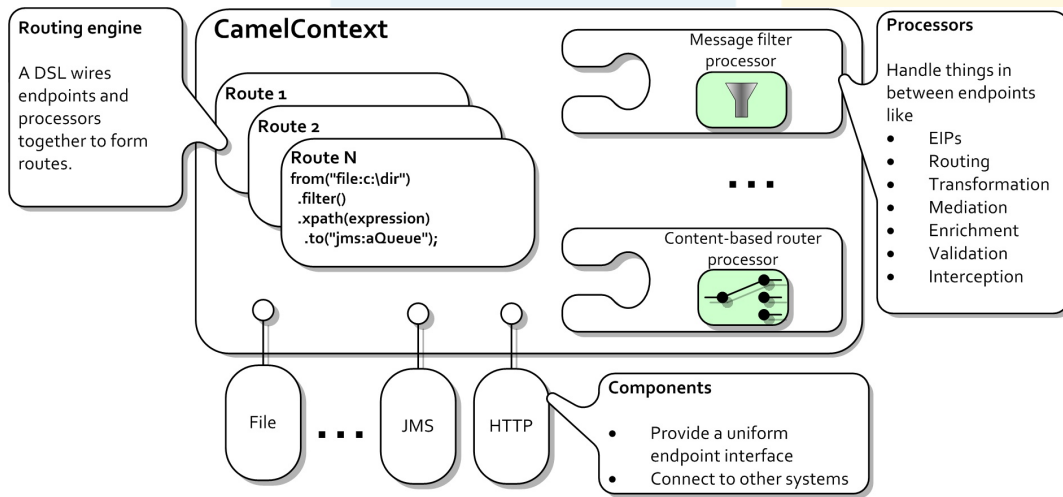
The background features a pattern of overlapping hexagons in light blue, light green, and light yellow. A thick green line is drawn across the center, forming a large, irregular hexagonal shape that frames the text.

# Apache Camel

Open Source EIP Framework

# Apache Camel

- Open Source Enterprise Integration Pattern Framework (Not an ESB)
- 100+ Components (File, JDBC, CMIS, REST, JMS, etc..)
- Multiple Route DSLs (XML, Java, Groovy, Kotlin)
- Custom Components + Beans
- Open Source (Apache 2.0 License)

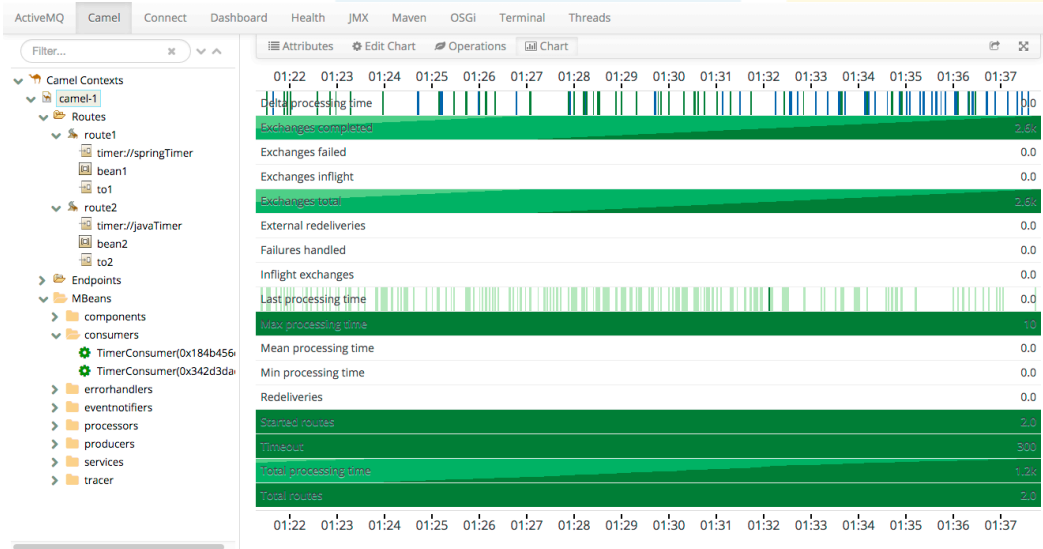


# Apache Camel – Recommended Stack

- Apache Karaf (OSGi Container)
- Hawtio (Web Console)
- Blueprint (OSGi DI Framework)
- Install Using Karaf CLI:

```
feature:repo-add camel  
feature:repo-add hawtio
```

```
feature:install camel  
feature:install camel-core  
feature:install camel-blueprint  
feature:install hawtio
```





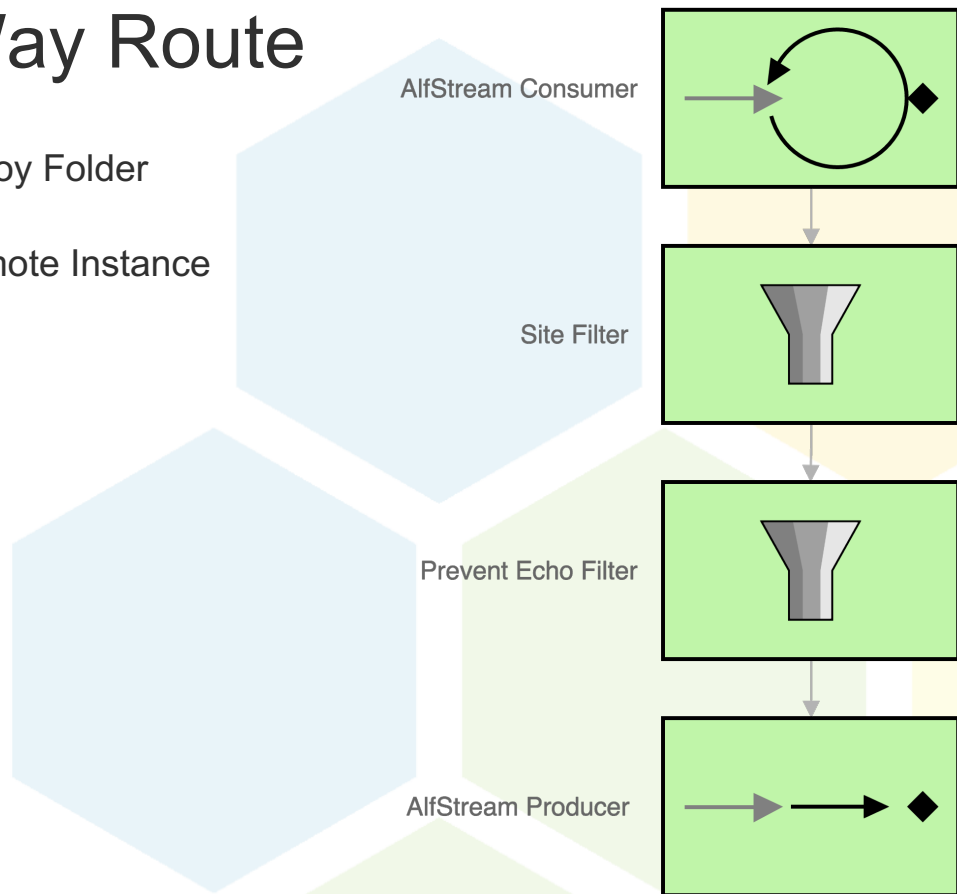


# Camel Routes

Route Configurations

# Apache Camel – Two Way Route

- Drop a Blueprint XML file into the Karaf Deploy Folder
- Poll and Consume Events from Alfresco Remote Instance
- Limit to specific Sites or Paths
- Prevent a Feedback Loop of Events
- Submit to Alfresco Local Instance
- Deployed to Both sides



The logo features a stylized orange camel silhouette positioned above three horizontal blue wavy lines, representing water. This entire graphic is enclosed within a green hexagonal border that is partially open on the left and right sides.

# AlfStream

Alfresco Camel Component

# AlfStream – Alfresco Camel Component

- Event Sourcing: Treats Alfresco as a Sequence of Events in an Event Log
- Use Transaction IDs for Tracking and Pagination – No ACL Check limitations and no reliance on time
- Retroactively applied – Does not rely on the Audit Service
- RESTful Endpoints - JSON for Consumer, Multipart for Producer
- Idempotent – Facilities for handling duplicate events
- Potential to expand to other frameworks such as Mule ESB or Standalone





# AlfStream Consumer – Alfresco Repo AMP

- RESTful Repo-End Webscript:

**maxResults:** max number of results to get back per call (500 by default)

**fromTxnId:** beginning transaction ID

**toTxnId:** ending transaction ID (uses last transaction ID from current time if not set)

**fromNodeId:** For pagination within a Transaction range if there are more than 500 entries

- Array of JSON NodeEvents (Using GSON):

```
[{
  "nodeRef": "91e4b557-20a9-4232-8ca3-285d31a323d8",
  "properties": {
    "cm_created": "2014-12-02T02:21:28.823Z",
    "cm_title": "Data Dictionary",
    "imap_maxUid": 0,
    "cm_description": "User managed definitions",
    "app_icon": "space-icon-default",
    "cm_creator": "System",
    "sys_node-uuid": "91e4b557-20a9-4232-8ca3-285d31a323d8",
    "cm_name": "Data Dictionary",
    "sys_store-protocol": "workspace",
    "sys_store-identifier": "SpacesStore",
    "sys_node-dbid": 14,
    "sys_locale": "en_US",
    "cm_modifier": "admin",
    "cm_modified": "2016-03-11T07:05:46.313Z",
    "imap_changeToken": "0a7a199a-2d1a-4fd1-b04c-7ef39fc9b35d"
  },
  "eventType": "UPSERT",
  "type": "cm_folder",
  "path": "/Company Home"
}]
```



# AlfStream Consumer – Camel Component

- Polls Repo Webscript
- Keeps Track of the current Transaction ID
- Converts NodeEvents into Camel Exchanges:
  - Exchange Headers include Node Metadata
  - Exchange Body is Content InputStream

```
app_icon = space-icon-default
Aspects = [cm_titled, cm_auditable, sys_referenceable, sys_localized,
app_uifacets]
Associations = []
AssocType = sys_children
breadcrumbId = ID-demo-53430-1492560010646-3-5
cm_created = 2017-02-14T07:49:30.593Z
cm_creator = System
cm_description = The company root space
cm_modified = 2017-02-14T07:49:38.096Z
cm_modifier = System
cm_name = Company Home
cm_title = Company Home
InheritPermissions = false
NodeEventType = UPSERT
NodeRef = 814a8066-6acd-44c8-a2e5-08ac7384798d
Path =
PermissionHash = ab54c3154b40bb5b741d4fd8ae0ca32370daf454
PropertyHash = 99872621d7152e8d2455a03a321ee45ee9dd2e0f
SecondaryParentAssociations = []
SetPermissions =
[{"permission": "Consumer", "accessStatus": "ALLOWED", "authority": "GROUP_EV
ERYONE", "authorityType": "EVERYONE", "position": 0}]
Site = null
sys_node-dbid = 13.0
sys_node-uuid = 814a8066-6acd-44c8-a2e5-08ac7384798d
sys_store-identifier = SpacesStore
sys_store-protocol = workspace
Type = cm_folder
```



# AlfStream Producer– Camel Component

- Converts Exchange to Multipart Form POST Submission
- (Optional) Checks to see whether Node exists first by using Property and Permission Checksum
- Uploads Exchange Body as Content Data if Present
- Not Limited to AlfStream Consumer – Can use any Camel Exchange Type (Such as the File Consumer)



# AlfStream Producer– Alfresco Repo AMP

- Multipart Form Data interface for submitting Nodes to Alfresco
- Ensures the Node's state is update as per the Request
- This includes changing (If necessary): Properties, Content, Permissions, Aspects, Peer and Parent Associations, Locks and Version Labels
- For Properties: Deserialise the the form request, converting into QName and Native Java Type based upon Content Model
- For Content: Update cm:content property based upon uploaded file





# Practice and Theory

Environmental Challenges

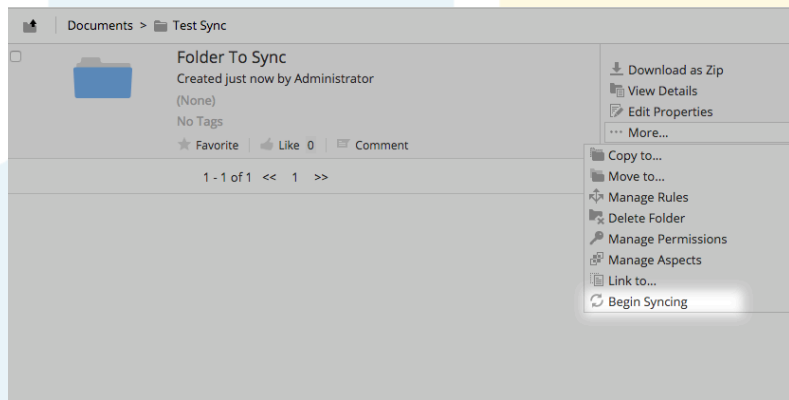
# User Configured Synchronisation

## Challenge

Users should be able to add and remove folders from sync easily, without having to readjust the Camel Route each time.

## Solution

Create an Aspect that cascades down to child nodes on application. Adjust the route to only listen for nodes with that aspect.



# Preventing a Feedback Loop

## Challenge

When one Alfresco Instance is Updated, it generates an Exchange that the originating instance receives. This can cause an Infinite Feedback Loop

## Solution

Skip Exchanges that have already been processed. Track equivalent Exchanges based upon Node UUID and Modification Time

# Updating Nodes

## Challenge

Modification Time is not always updated when changes are made (I.e, when a Node is Locked, or ACLs are Updated). This causes some Exchanges to be ignored when they should be processed

## Solution

Generate a Node SHA Hash for both Permissions and Properties for equivalence. As a default use Modification Date, Lock Type and Version Label as inputs for the Property Hash (converting them to their byte values)

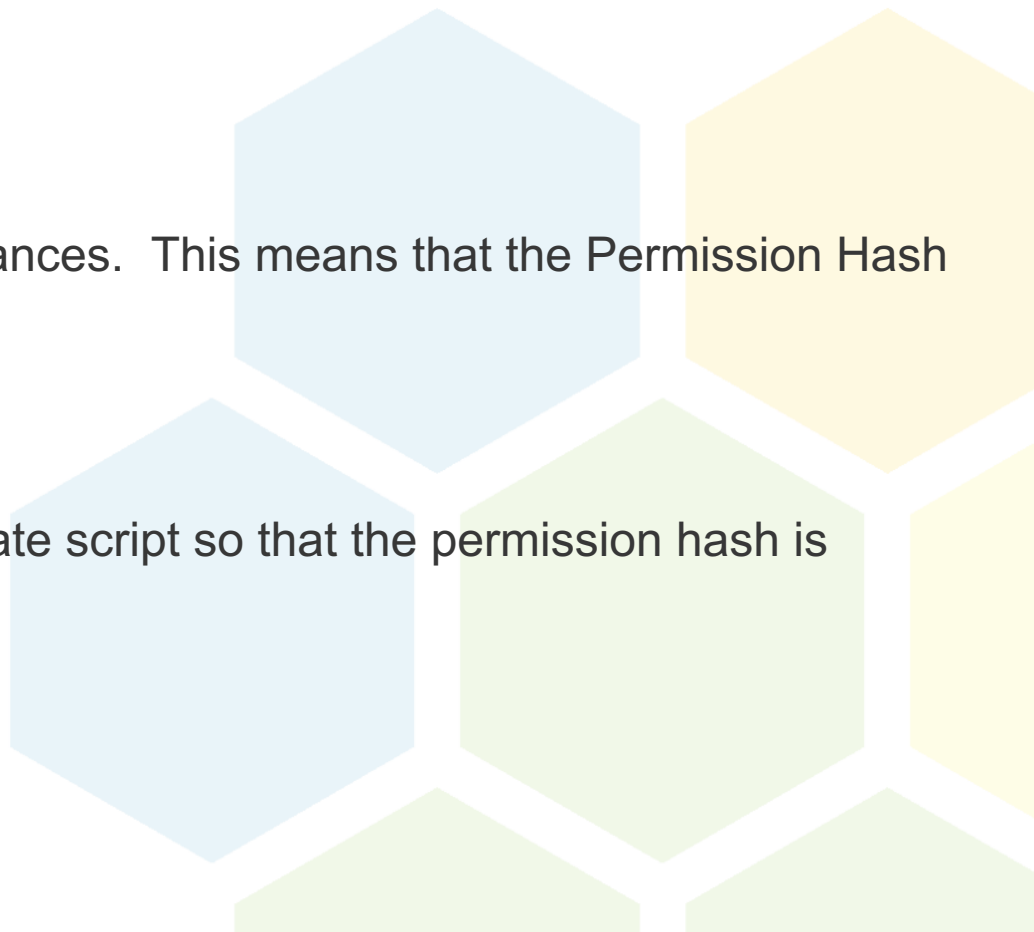
# Permission Authorities

## Challenge

Authorities may not exist on both instances. This means that the Permission Hash may not be equal on each instance

## Solution

Generate an Authority within the Update script so that the permission hash is always equal



# Permission Changes

## Challenge

When you update the Permissions of a Node, this is not done within a Transaction: It is done within an ACL Change Set. This means that Exchanges aren't generated when ACLs of a Node are changed.

## Solution

Track ACL Changesets as well as Node Transactions, generating events if either one changes.

# Version Numbers Sync

## Challenge

When you receive an Exchange and update a node, the version number may be different at the other end (I.e, Major Update instead of Minor).

## Solution

Adjust the Version Service to be able to Provide the correct Version Label

# Restarting the Route

## Challenge

When you Restart the Camel Route, the AlfStream consumer will begin from the beginning. This can take a long time if there are 1000s of Nodes to process.

## Solution

Allow the AlfStream producer to persist transaction ids and changesets to a file so it can pick up where it left off if it restarts



The background features a honeycomb pattern of hexagons in light blue, light green, and light yellow. A thick green line is drawn across the image, forming a large, irregular shape that frames the central text.

Quick Demo



# Looking Ahead

Changes and Updates to AlfStream

# Full Site Synchronisation

## Challenge

Sites are cached in Alfresco Share have cached configurations. This means that updating it within the Repo End does not reflect the changes from the Front End

## Solution

Force Share to reset its cache when changes to the dashboard configuration take place

# Transaction Level Exchanges

## Challenge

Groups of nodes need to be updated atomically within the same exchange. This prevents things like Folder Rules from Syncing correctly

## Solution

Allow the consumer and producer to handle and update multiple nodes within the same transaction block

# SaaS Storage Integrations



Google Drive

The box logo, featuring the word "box" in white lowercase letters on a blue rectangular background.

box

The background features a honeycomb pattern of hexagons in light blue, light green, and light yellow. A thick green line is drawn across the image, starting from the left, forming a partial hexagon, and then continuing as a horizontal line that passes behind the word 'Conclusion'.

Conclusion

# Conclusion

- Synchronisation between systems is a very common use case
- Apache Camel provides a platform for creating Routes and Integrations and abstracting away common integration paradigms
- Apache Karaf + Hawtio provides a base for managing Camel Routes and hot deploying changes
- Camel allowed us to create custom component to handle Consuming and Producing from Alfresco to handle our existing and future use cases
- Integration is always more challenging than you think!

# Speaker contacts

Website: <https://www.parashift.com.au>

Github: <https://github.com/cetra3/>

Email: [peter@parashift.com.au](mailto:peter@parashift.com.au)