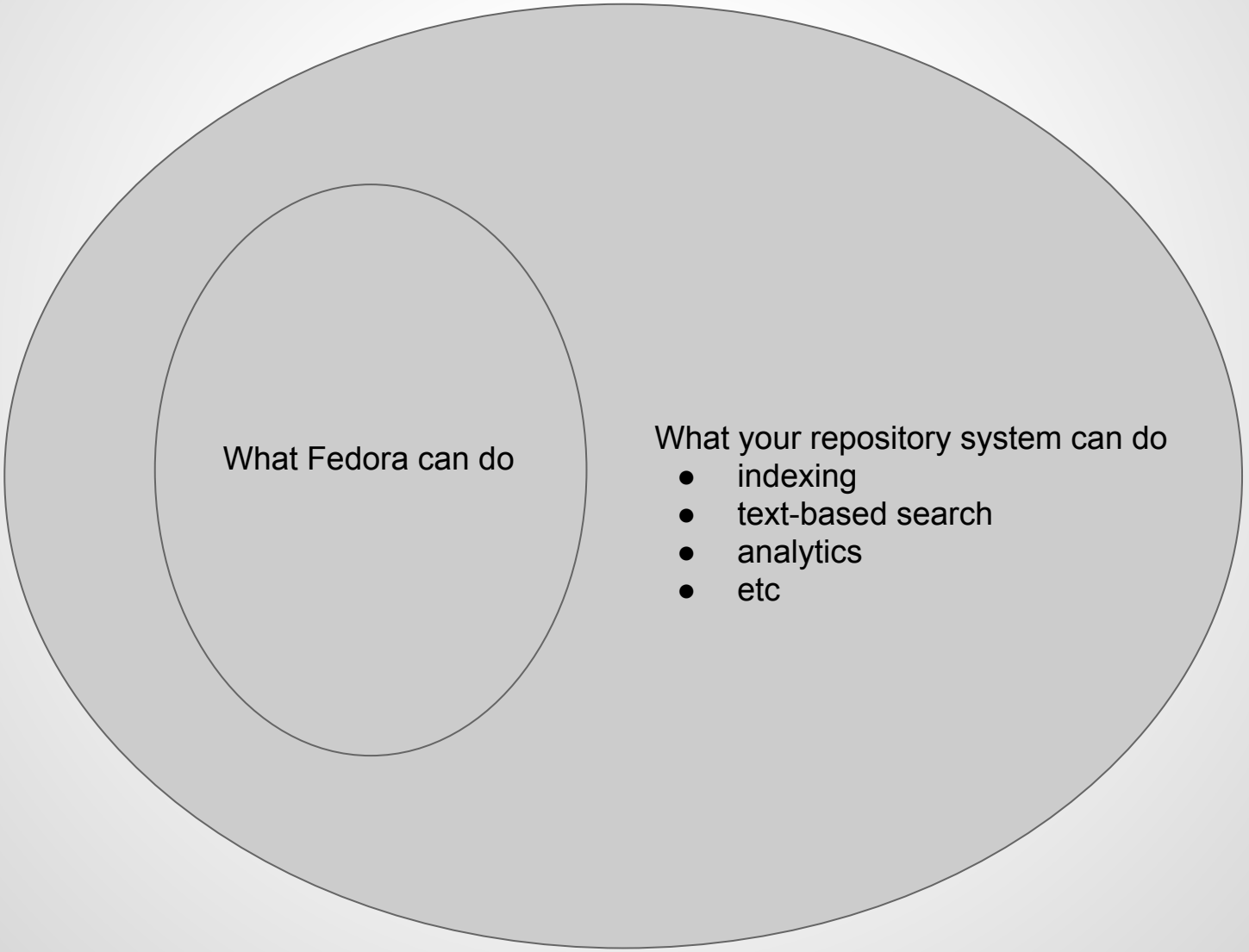
External Integrations

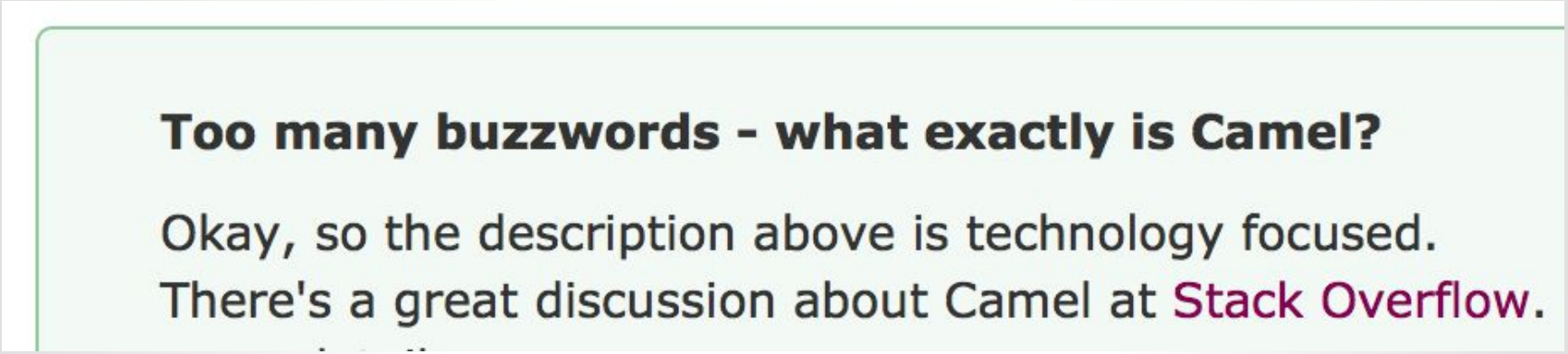
Introducing Camel

Repository Scope



What is Camel?

Good question. See: http://camel.apache.org/what-is-camel.html



So really you want see this: http://stackoverflow.com/questions/8845186/what-exactly-is-apache-camel

In short...

Camel is a framework for creating small message based applications… and then some.

Camel is middleware for integration with external systems

Camel is all the code you should not have to write in order to work with queues, files, databases, RESTful APIs, common data formats, command line utilities, etc… in a consistent and reliable manner.

Available Camel Components

http://camel.apache.org/components.html

- ActiveMQ
- AWS SQS
- DropBox
- System calls
- Local files
- FTP
- HTTP resources
- LDAP
- SMTP
- SQL
- Twitter
- etc, etc, etc

Camel can run...

As a stand-alone Java application

In a servlet container like Tomcat or Jetty

In an OSGi runtime such as Karaf

What is OSGi?

Open Service Gateway Initiative

Framework for modularizing and deploying Java applications

- Hot deployment
- Automatic reloading of configuration
- Sophisticated dependency resolution
- XML scripting for complex deployments (features)

Hot Deployment

Bundles can be started, stopped, updated, etc… at runtime!

In other words:

YOU DO NOT HAVE TO RESTART YOUR SERVER TO UPDATE CODE OR CONFIGURATION

Camel Examples

Routing

```
from("activemq:topic:fedora")
    .to("http4:localhost/api");
```

Transformation

```
from("activemq:topic:fedora")
    .process(new SparqlUpdater())
    .to("http4:localhost/api");
```

Filter

```
from("activemq:topic:fedora")
    .filter(header("org.fcrepo.jms.identifier").startsWith("/audit"))
    .to("http4:localhost/api");
```

Triplestore

Fedora 3: internal triplestore

Resource index (mulgara)

Very useful

Very slow

Very resource intensive

Fedora 4: no internal triplestore

This keeps fedora running faster

There is less code to maintain in the fedora core

The fedora API is simpler

Not everyone uses a triplestore

Decoupled components are easier to scale out

Mirror Fedora content in a triplestore

Two approaches:

1) Synchronous: client(s) operate on content in fedora and then do the same in a triplestore

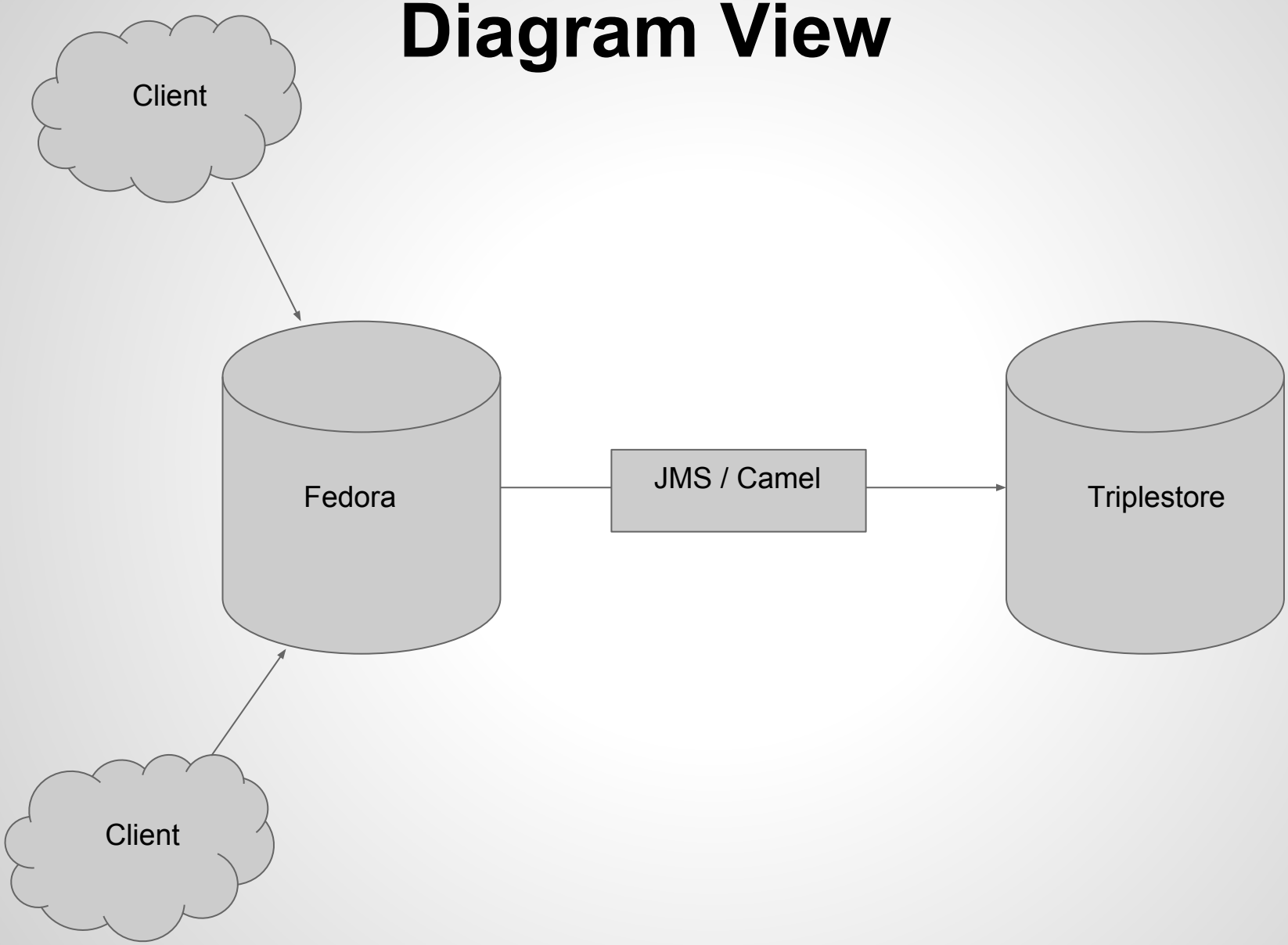
2) Asynchronous: client(s) operate on content only in fedora, a separate process synchronizes changes

Asynchronous Patterns

Camel recipe for this (DSL for integration patterns)

`fcrepo-indexing-triplestore` in github.com/fcrepo4-exts/fcrepo-camel-toolbox

Diagram View



Indexing to a Triplestore

```
from("activemq:topic:fedora")
     .to("fcrepo:{{fcrepo.baseUrl}}?accept=application/n-triples")
     .process(new SparqlUpdateProcessor())
     .to("http4://{{triplestore.baseUrl}}");
```

Choose your own triplestore

Fuseki comes with the fcrepo-vagrant VM

In production, any will work. These have been tested:

- Sesame/OpenRDF
- Jena/Fuseki
- Blazegraph

...just update the location of the triplestore in the camel configuration

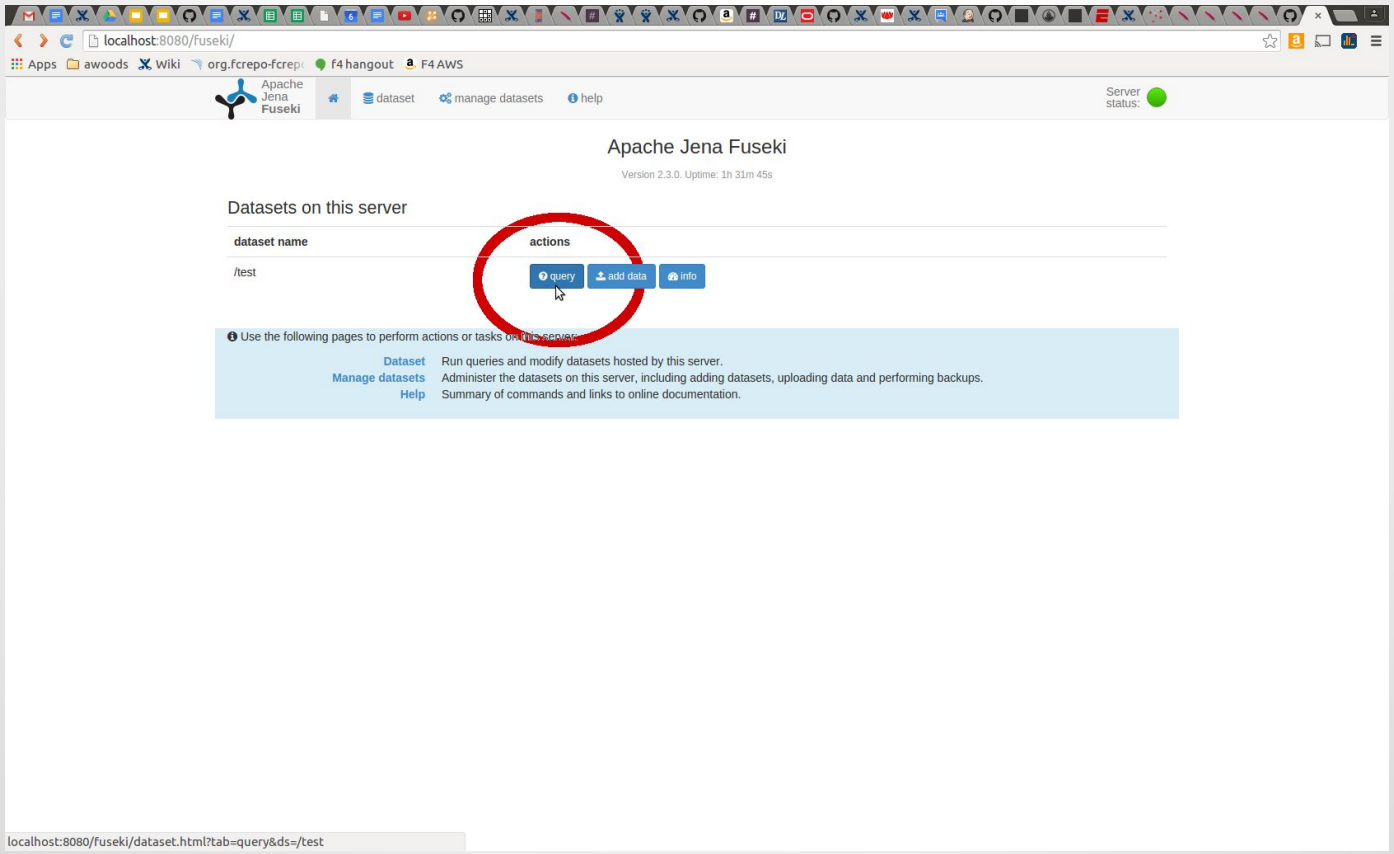
SPARQL & RDF Triples: S P O

Syntax like English (sort of):

- Statements end with a period
- Clauses end with a semicolon
- Lists are separated with a comma

Hands-On: Indexing in triplestore

http://localhost:8080/fuseki



Hands-On: Indexing in triplestore

```
select * where {
  <http://localhost:8080/fcrepo/rest/cover> ?p ?o
}
```

Hands-On: Indexing in triplestore

```
PREFIX ldp: <http://www.w3.org/ns/ldp#>
PREFIX ebucore: <http://www.ebu.
ch/metadata/ontologies/ebucore/ebucore#>

select * where {
   ?s ldp:contains ?o .
   ?o ebucore:hasMimeType ?m
}
```

Hands-On: Indexing in triplestore *Audit*

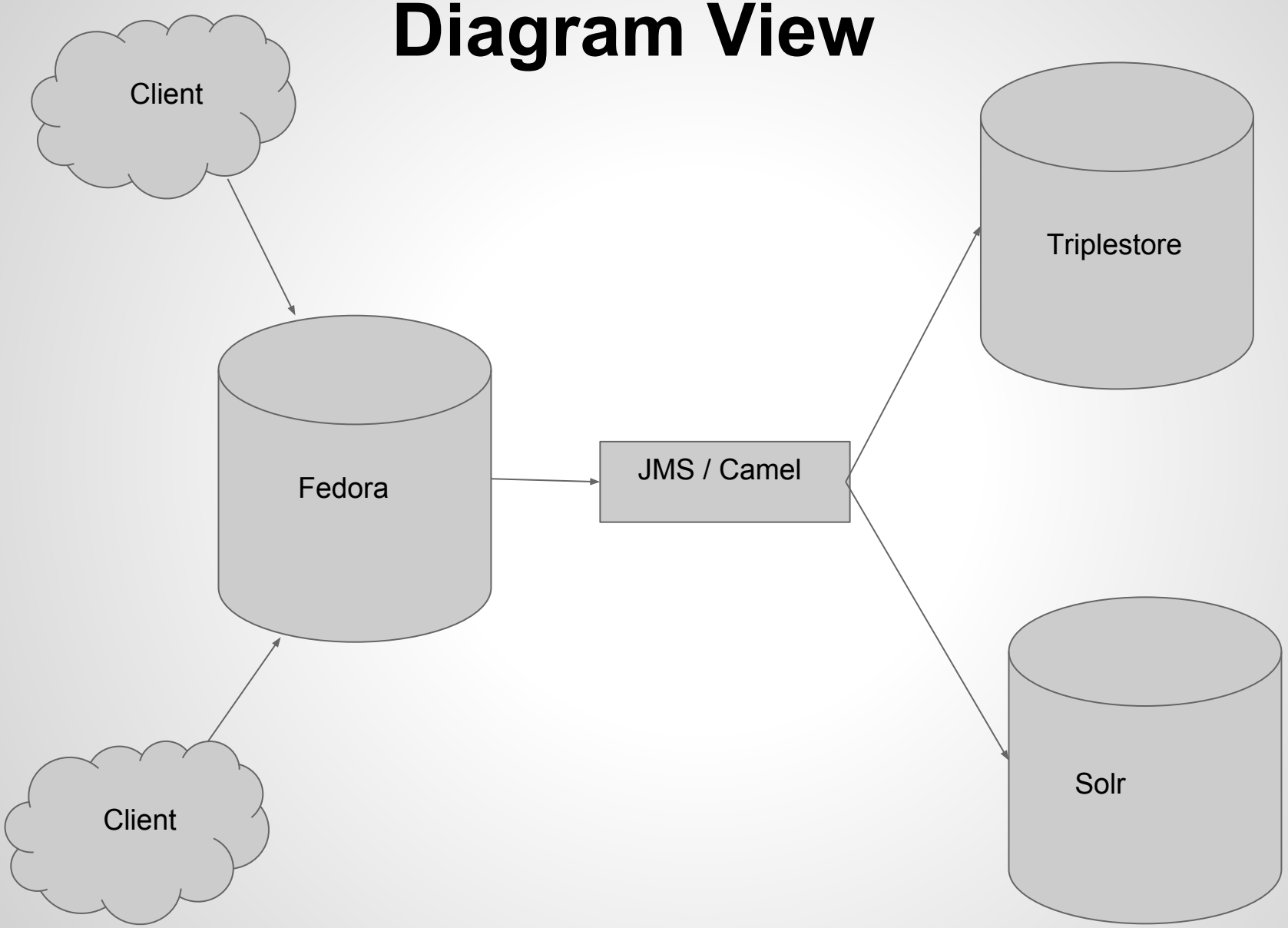
```
prefix premis: <http://www.loc.gov/premis/rdf/v1#>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>

select ?s ?d where {
   ?s ?p <http://fedora.info/definitions/v4/audit#InternalEvent> .
   ?s premis:hasEventRelatedObject <http://localhost:8080/fcrepo/rest/cover> .
   ?s premis:hasEventDateTime ?d .
   FILTER (?d > "2015-10-06T04:21:14Z"^^xsd:dateTime)
}
```

Solr

Indexing to Solr

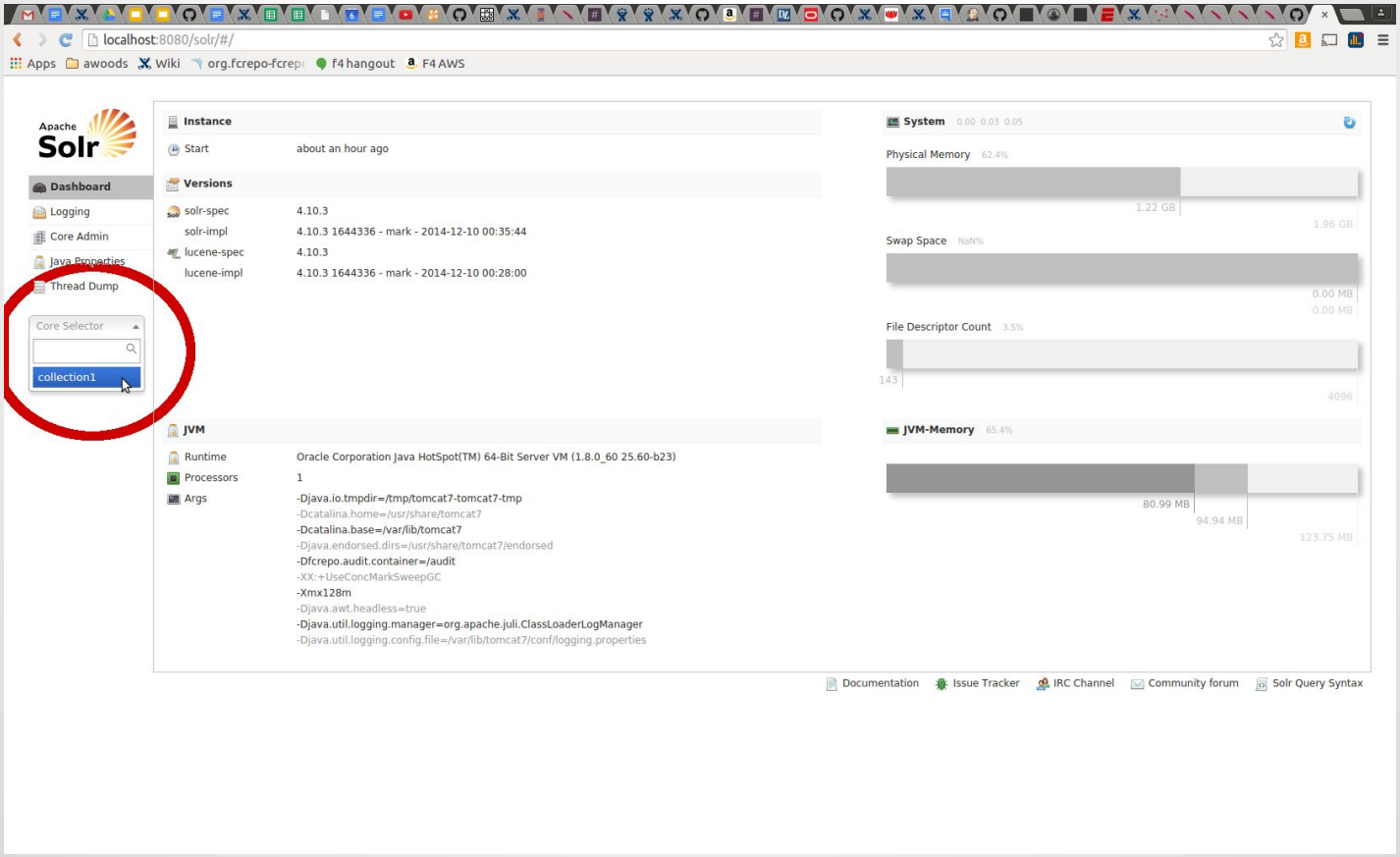
```
from("activemq:topic:fedora")
  .choice()
    .when(header("org.fcrepo.jms.eventType")
      .isEqualTo("http://fedora.info/definitions/v4/repository#NODE_REMO
        .to("direct:delete")
    .otherwise()
      .to("direct:index");

from("direct:delete")
  .process(new SolrDeleteProcessor())
  .to("http4://localhost:8983/solr/collection1/update");

from("direct:index")
  .to("fcrepo:localhost:8080/rest?transform=default")
  .to("http4://localhost:8983/solr/collection1/update");
```

Diagram View

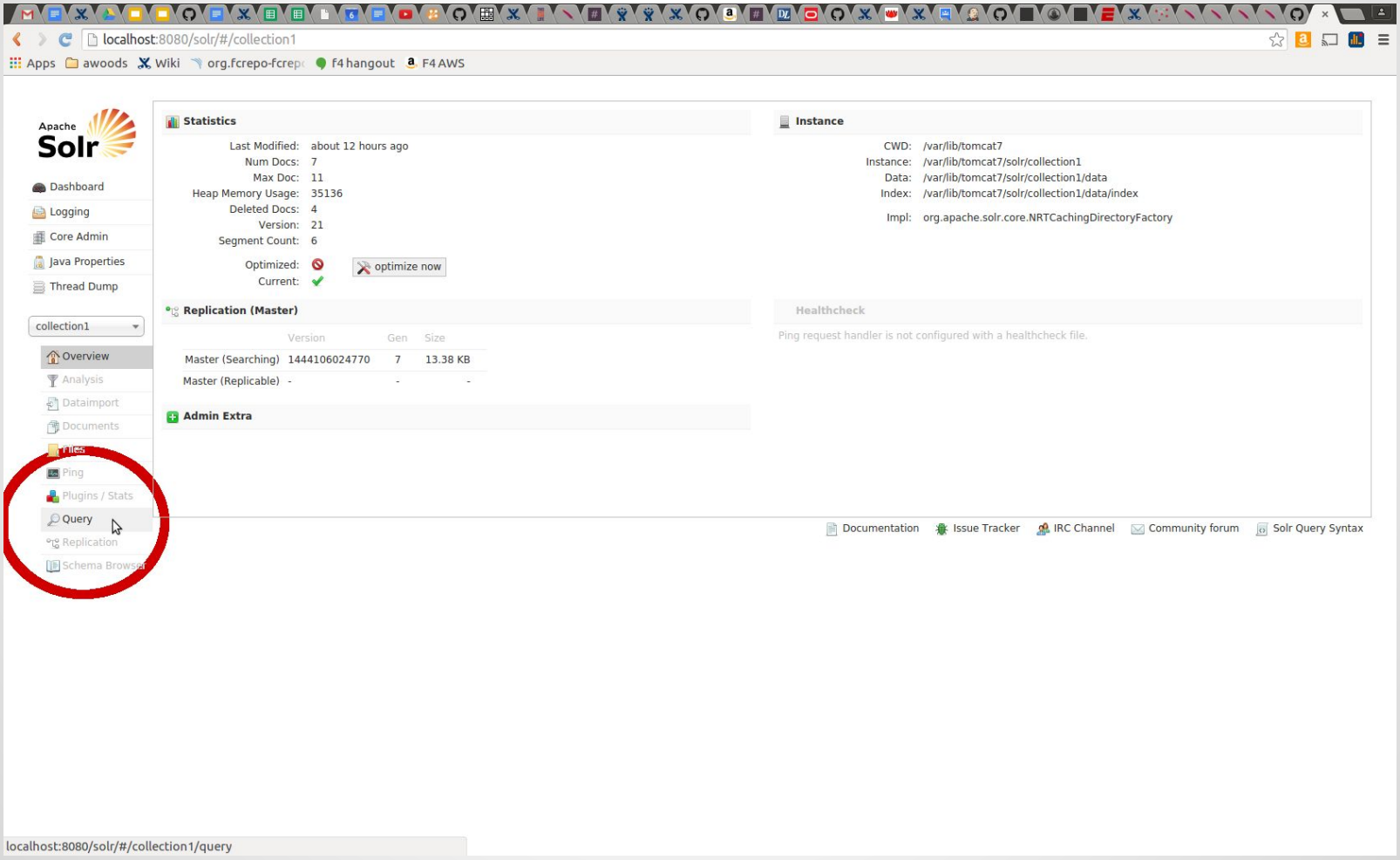


Hands-On: Indexing in Solr

http://localhost:8080/solr



Hands-On: Indexing in Solr



Success!