

Like a Can Opener for your Data Silo

simple access through AtomPub and Jangle

A short history of library APIs

Z39.50

- Amazingly, yes, this has to be considered an API
- Own network protocol
- `@or @and @attr 1=1003 "Hemingway, Ernest" @attr 1=4 "The Sun Also Rises" @attr 1=4 "A Farewell to Arms"`
- Unknown/unused outside of library domain
- Client support generally needs to be compiled in
- Incredibly high barrier to entry to create services

SRU

- Improvement: XML over HTTP
- CQL vastly more user & developer friendly than RPN/PQF
- Unknown/unused outside of library domain
- Uncommon *inside* the library domain
- Read only (currently)

OAI-PMH

- Simple and effective
- Provides unambiguous, reusable identifiers for records
- Read only
- No search capability
- Retrieve one or everything. No way to request multiple specific identifiers
- Can only transport XML

DLF ILS-DI API

- “Best of Breed” approach
 - OAI-PMH
 - SRU
 - NCIP, possibly

Proprietary APIs

- RDBMS Access
- SirsiDynix Unicorn/Symphony API
- Ex Libris X-Server

The net effect

- scattershot
- niche
- awkward
- limiting

Atom Publishing Protocol



AtomPub

- IETF Standard (RFC 5023) for publishing content on the web
- Atom Syndication Format + REST = AtomPub
- Workspaces, Collections, Entries, Categories
- Unambiguous identifiers (via URIs) for every resource
- Only two kinds of documents ever served: Atom feeds and service documents

AtomPub continued

- Used by Google, Microsoft, IBM
- Available in Wordpress, MovableType, Drupal, etc.
 - Broad client support
 - Broad awareness outside library domain
- No baked in search, but can easily use OpenSearch (which adds a third document type with the description document)

Jangle

- Applies a common data model to library services using AtomPub
- Four discrete collection types (*Entities*)
 - Resources, Items, Actors, Collections
- Two components
 - Core & Connectors
- OpenSearch + CQL

Jangle

- Applies a common data model to library services using AtomPub
- Four discrete collection types (*Entities*)
 - *Resources*, Items, Actors, Collections
- Two components
 - Core & Connectors
- OpenSearch + CQL

Resources

- The primary objects being exposed by this service
 - Bibliographic records
 - Reserve records
 - Archival collections
 - Electronic Journals

Jangle

- Applies a common data model to library services using AtomPub
- Four discrete collection types (*Entities*)
 - Resources, *Items*, Actors, Collections
- Two components
 - Core & Connectors
- OpenSearch + CQL

Items

- A specific physical representation of a Resource
 - A copy of a book
 - Serials holdings
 - An electronic representation (PDF, PS, JPG, etc.)

Jangle

- Applies a common data model to library services using AtomPub
- Four discrete collection types (*Entities*)
 - Resources, Items, *Actors*, Collections
- Two components
 - Core & Connectors
- OpenSearch + CQL

Actors

- The 'users' of a system
 - Borrowers
 - Submitters
 - Account holders
 - Content creators

Jangle

- Applies a common data model to library services using AtomPub
- Four discrete collection types (*Entities*)
 - Resources, Items, Actors, *Collections*
- Two components
 - Core & Connectors
- OpenSearch + CQL

Collections

- Any combination of the other entities
- Can be homogenous or heterogenous among entity types

Jangle

- Applies a common data model to library services using AtomPub
- Four discrete collection types (*Entities*)
 - Resources, Items, Actors, Collections
- Two components
 - *Connectors & Core*
- OpenSearch + CQL

Connectors

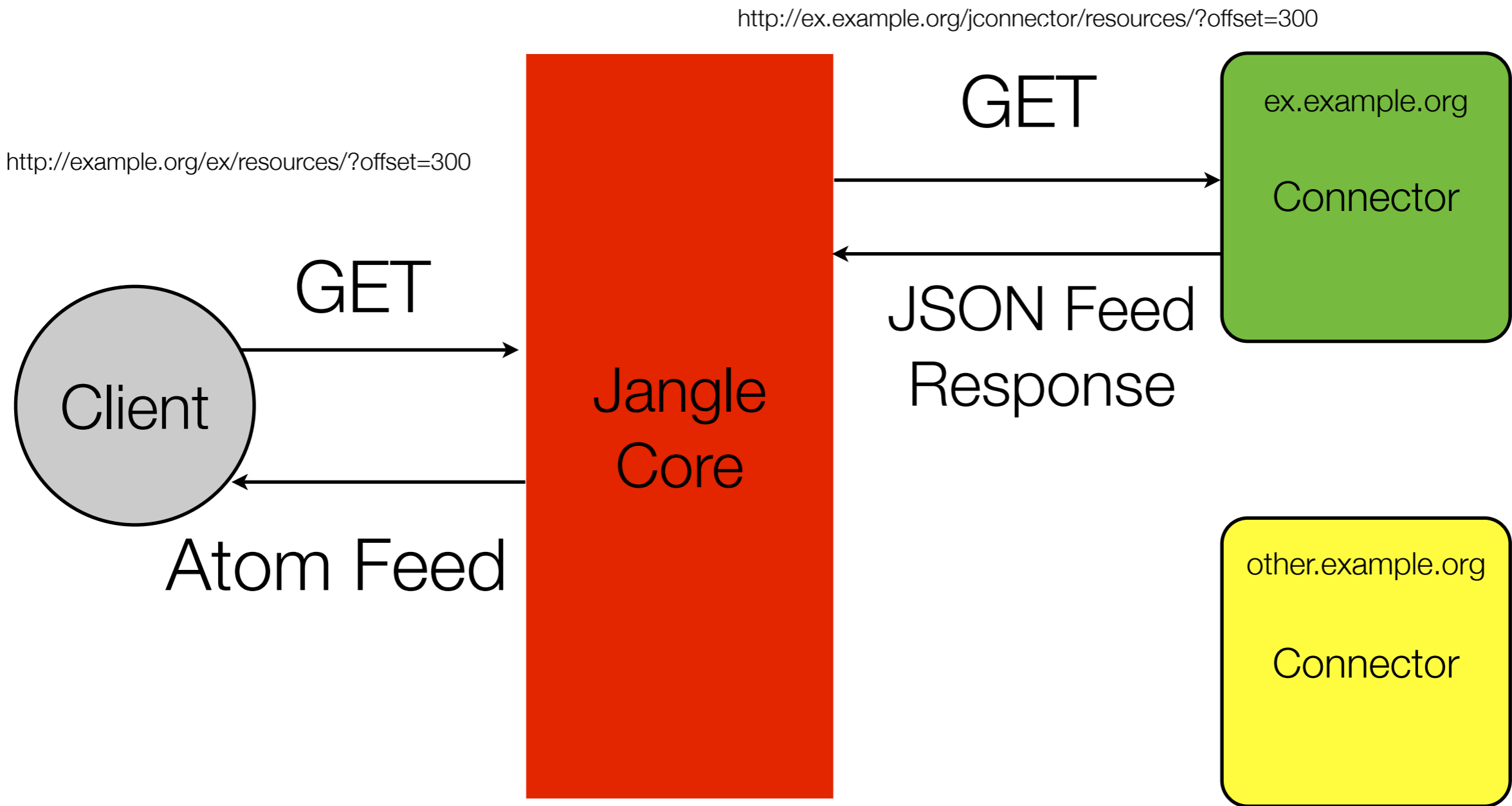
- Provide the business logic for specific systems
- Provide responses as JSON objects
- Four response type:
 - Service, Feed, Search, Explain
- Inspired by, but not identical to, AtomPub

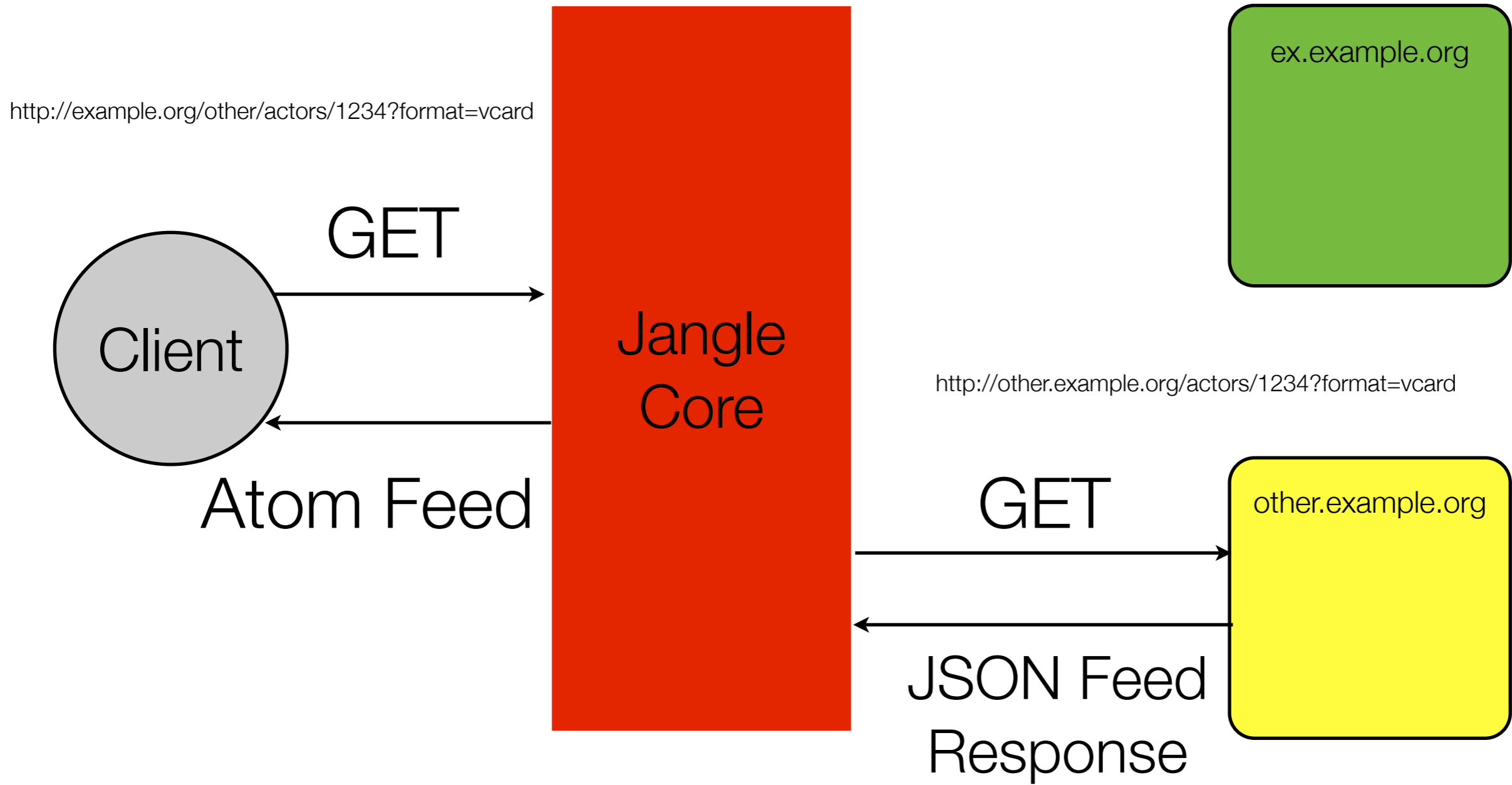
Jangle

- Applies a common data model to library services using AtomPub
- Four discrete collection types (*Entities*)
 - Resources, Items, Actors, Collections
- Two components
 - Connectors & *Core*
- OpenSearch + CQL

The Jangle Core

- The AtomPub public facing interface
- Proxies requests for one or many connectors
- Serializes connector responses into:
 - Atom service documents
 - Atom feeds
 - OpenSearch description documents





Services Connector Response

```
{ "request":"http://demo.jangle.org/openbiblio/services",
  "type":"services",
  "version":"1.0",
  "title":"openbiblio",
  "entities":
  { "Actor":{
    "title":"Borrowers",
    "path":"/vectors",
    "searchable":false
  }, "Resource":{
    "title":"Bibliographic records",
    "path":"/resources",
    "searchable":"/openbiblio/resources/search/description",
    "categories":
    ["opac"]
  }, "Item":{
    "title":"Holdings records",
    "path":"/items",
    "searchable":false,
    "categories":
    ["copy","hold"]
  },
  "Collection":
  {
    "title":"Categories",
    "path":"/collections",
    "searchable":false
  }
  },
  "categories":
  { "opac":{
    "scheme":"http://jangle.org/vocab/terms#dlf-ilsdi-resource"
  }, "hold":{
    "scheme":"http://jangle.org/vocab/terms#hold"
  }, "copy":{
    "scheme":"http://jangle.org/vocab/terms#copy"
  }
  }
}
```

Service Document

```
<workspace>
  <atom:title>openbiblio</atom:title>
  <collection href="http://demo.jangle.org/openbiblio/actors">
    <atom:title>Borrowers</atom:title>
  </collection>
  <collection href="http://demo.jangle.org/openbiblio/resources">
    <atom:title>Bibliographic records</atom:title>
    <atom:category term='opac' scheme='http://jangle.org/vocab/terms#dlf-ilsdi-
resource' />
  </collection>
  <collection href="http://demo.jangle.org/openbiblio/items">
    <atom:title>Holdings records</atom:title>
    <atom:category term='copy' scheme='http://jangle.org/vocab/terms#copy' />
    <atom:category term='hold' scheme='http://jangle.org/vocab/terms#hold' />
  </collection>
  <collection href="http://demo.jangle.org/openbiblio/collections">
    <atom:title>Categories</atom:title>
  </collection>
</workspace>
```

Service Document

```
<service xmlns="http://www.w3.org/2007/app" xmlns:atom="http://www.w3.org/2005/Atom">
  <workspace>
    <atom:title>openbiblio</atom:title>
    <collection href="http://demo.jangle.org/openbiblio/actors">
      <atom:title>Borrowers</atom:title>
    </collection>
    <collection href="http://demo.jangle.org/openbiblio/resources">
      <atom:title>Bibliographic records</atom:title>
      <atom:category term='opac' scheme='http://jangle.org/vocab/terms#dlf-ilsdi-resource' />
    </collection>
    <collection href="http://demo.jangle.org/openbiblio/items">
      <atom:title>Holdings records</atom:title>
      <atom:category term='copy' scheme='http://jangle.org/vocab/terms#copy' />
      <atom:category term='hold' scheme='http://jangle.org/vocab/terms#hold' />
    </collection>
    <collection href="http://demo.jangle.org/openbiblio/collections">
      <atom:title>Categories</atom:title>
    </collection>
  </workspace><workspace>
    <atom:title>alto</atom:title>
    <collection href="http://demo.jangle.org/alto/collections">
      <atom:title>Categories</atom:title>
    </collection>
    <collection href="http://demo.jangle.org/alto/items">
      <atom:title>Holdings records</atom:title>
      <atom:category term='copy' scheme='http://jangle.org/vocab/terms#copy' />
      <atom:category term='hold' scheme='http://jangle.org/vocab/terms#hold' />
    </collection>
    <collection href="http://demo.jangle.org/alto/resources">
      <atom:title>Bibliographic records</atom:title>
      <atom:category term='opac' scheme='http://jangle.org/vocab/terms#dlf-ilsdi-resource' />
    </collection>
    <collection href="http://demo.jangle.org/alto/actors">
      <atom:title>Borrowers</atom:title>
    </collection>
  </workspace>
</service>
```

Jangle

- Applies a common data model to library services using AtomPub
- Four discrete collection types (*Entities*)
 - Resources, Items, Actors, Collections
- Two components
 - Core & Connectors
- OpenSearch + CQL

Explain Document

```
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/" xmlns:jangle="http://jangle.org/opensearch/">
  <ShortName>Bibliographic records</ShortName>
  <LongName>Search Bibliographic records in OpenBiblio</LongName>
  <Description>Bibliographic records search. Defaults to keyword anywhere.</Description>
  <SyndicationRight>open</SyndicationRight>
  <Tags>catalog library</Tags>
  <Query role="example" searchTerms="dc.title=thomas">
    <zr:explain xmlns:zr="http://explain.z3950.org/dtd/2.1/">
      <zr:indexInfo>
        <zr:set name="dc" identifier="info:srw/cql-context-set/1/dc-v1.1"/>
          <zr:index><zr:map><zr:name set="dc">title</zr:name></zr:map></zr:index>
          <zr:index><zr:map><zr:name set="dc">creator</zr:name></zr:map></zr:index>
          <zr:index><zr:map><zr:name set="dc">subject</zr:name></zr:map></zr:index>
          <zr:index><zr:map><zr:name set="dc">identifier</zr:name></zr:map></zr:index>
        <zr:set name="rec" identifier="info:srw/cql-context-set/2/rec-1.1"/>
          <zr:index><zr:map><zr:name set="rec">identifier</zr:name></zr:map></zr:index>
          <zr:index><zr:map><zr:name set="rec">collectionName</zr:name></zr:map></zr:index>
          <zr:index><zr:map><zr:name set="rec">lastModificationDate</zr:name></zr:map></zr:index>
          <zr:index><zr:map><zr:name set="rec">creationDate</zr:name></zr:map></zr:index>
        <zr:set name="cql" identifier="info:srw/cql-context-set/1/cql-v1.2"/>
          <zr:index><zr:map><zr:name set="cql">allIndexes</zr:name></zr:map></zr:index>
          <zr:index><zr:map><zr:name set="cql">anyIndexes</zr:name></zr:map></zr:index>
          <zr:index><zr:map><zr:name set="cql">keywords</zr:name></zr:map></zr:index>
      </zr:indexInfo>
    </zr:explain>
  </Query>
</OpenSearchDescription>
```

Explain Document

<Query role="example" searchTerms="dc.title=thomas">

<zr:explain xmlns:zr="http://explain.z3950.org/dtd/2.1/">

<zr:indexInfo>

<zr:set name="dc" identifier="info:srw/cql-context-set/1/dc-v1.1"/>

<zr:index><zr:map><zr:name set="dc">title</zr:name></zr:map></zr:index>

<zr:index><zr:map><zr:name set="dc">creator</zr:name></zr:map></zr:index>

<zr:index><zr:map><zr:name set="dc">subject</zr:name></zr:map></zr:index>

<zr:index><zr:map><zr:name set="dc">identifier</zr:name></zr:map></zr:index>

<zr:set name="rec" identifier="info:srw/cql-context-set/2/rec-1.1"/>

<zr:index><zr:map><zr:name set="rec">identifier</zr:name></zr:map></zr:index>

<zr:index><zr:map><zr:name set="rec">collectionName</zr:name></zr:map></zr:index>

<zr:index><zr:map><zr:name set="rec">lastModificationDate</zr:name></zr:map></zr:index>

<zr:index><zr:map><zr:name set="rec">creationDate</zr:name></zr:map></zr:index>

<zr:set name="cql" identifier="info:srw/cql-context-set/1/cql-v1.2"/>

<zr:index><zr:map><zr:name set="cql">allIndexes</zr:name></zr:map></zr:index>

<zr:index><zr:map><zr:name set="cql">anyIndexes</zr:name></zr:map></zr:index>

<zr:index><zr:map><zr:name set="cql">keywords</zr:name></zr:map></zr:index>

</zr:indexInfo>

</zr:explain>

</Query>

Atom with extensions

- Jangle adds a few extensions & conventions to establish:
 - relationships between entities
 - alternate metadata formats for resources
 - indexes for OpenSearch queries

Jangle Vocabulary

- URIs to unambiguously define relationships, formats, categories
 - <http://jangle.org/vocab/formats#application/marc>
 - <http://jangle.org/vocab/Entity#Actor>
 - <http://jangle.org/vocab/terms#dlf-ilsdi-resource>
- Should eventually move to the NSDL MetadataRegistry or similar service

Feed Document

- <http://demo.jangle.org/openbiblio/resources/>
 - <http://connector.jangle.org/resources/>
- <http://demo.jangle.org/openbiblio/actors/1711/items>
- <http://demo.jangle.org/openbiblio/items/-/copy>

Current State of Jangle

- Version 1.0 of the specification approved in November
 - Currently compiling requirements for 1.1
- Connector & Core frameworks available in
 - PHP
 - Ruby
 - Groovy

Jangle enabled applications

- Connectors
 - OpenBiblio ILS - Reference ILS implementation
 - Talis Alto
- Helios/fac-back-opac/Kobold Chieftain
- Scriblio
- Blacklight

Adapters

- Convert Jangle's output to other formats
- DLF ILS-DI
 - OAI-PMH
 - Availability Lookup
- Google SiteMaps

The Future

- Need more connectors to begin establishing community profiles
- Begin experimenting with POST, PUT, DELETE
 - SWORD as template?
- Examples of non-OPAC based client support
 - Courseware
 - Reserves systems

The Community

- <http://jangle.org/>
 - spec, announcements, HOWTOs
- <http://groups.google.com/group/jangle-discuss>
 - Primary discussion forum
- <http://code.google.com/p/jangle>
 - Source, Issue tracking

Questions?

Thanks!

Ross Singer - Talis
rossfsinger@gmail.com