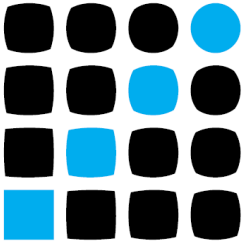


adaptTo()

APACHE SLING & FRIENDS TECH MEETUP

BERLIN, 22-24 SEPTEMBER 2014

# Integrating Open Source Search with AEM

Gaston Gonzalez


# Agenda

- Apache Solr
- Data Ingestion
  - Drivers
  - Implementation Patterns
- Searching Data


# Data Ingestion: Drivers

# Choosing a Data Ingestion Approach

- Where is your content stored?
  - AEM/CQ
  - External databases and data stores
  - External feeds (RSS)
- How many documents?
  - Small – Tens of Thousands
  - Medium – Hundreds of Thousands
  - Large – Millions+

adaptTo()

# Choosing a Data Ingestion Approach (2/3)

- Index latency
  - Near real-time indexing (low latency)
  - Hourly, daily, weekly (high latency)
- Do you need document enrichment?
  - Add additional metadata
  - Sanitize and transform data
  - Merge/join documents/records

# Choosing a Data Ingestion Approach (3/3)

- Implementation Complexity
  - Implementation effort
  - Time to market
- Supporting infrastructure
  - Crawlers
  - Document Processing Platform
  - Messaging Queues


# Data Ingestion: Implementation Patterns

# Solr Update Handler Pattern

**Method:** Pull

**Content Location:** CQ, CQ + External

**Document Corpus:** Small

**Index Latency:** High

**Document Enrichment:** Simple

**Implementation Complexity:** Low

**Infrastructure:** None

**Recipe**

1. Implement servlet to generate same output as a Solr's update handler (i.e., JSON).
2. Create script to invoke servlet (curl) and save response to file system.
3. Update index by posting (curl) documents to Solr request handler (i.e., JsonUpdateRequestHandler).
4. Call script from scheduler (cron).

# Solr Update Handler Pattern (2/3)

## Summer Blockbuster Hits and Misses

by Iris Mccoy
08/21/2012

This year's summer movie season is coming to a close, and the box office total is actually down about 9% last years record-breaking year. But while the combined numbers were off, there were still a few blockbusters that brought fans out in droves. See the biggest hits and misses from May through August.

— most popular —


Raw | Parsed

```
{
  "id": "/content/geometrixx-media/en/entertainment/summer-blockbuster-hits-and-misses",
  "author": "Iris Mccoy",
  "body": "This year's summer movie season is coming to a close, and the box office total is actually down about 9% last years record-breaking year. But while the combined numbers were off, there were still a few blockbusters that brought fans out in droves. See the biggest hits and misses from May through August. Double Identity Summer started with the superhero team smashing the record for biggest opening ever with a $207 million debut. \"Double Identity\" is now only $40M away from surpassing \"Tablo\" to be the #2 highest-grossing film of all time domestically. And with the movie coming back to theaters for Labor Day weekend, there is a chance it could take it. Warship \"Warship\" was an enormously expensive misfire (reportedly costing over $200 million). The board-game adaptation was more successful internationally than in the U.S. -- it actually made less here than its star Kyle Kitchon's other high-profile bomb “Mr. President\" -- but not enough to justify a sequel. Rock and Roll After the success of last year's \"Impossible Mission,\" it looked like Tim Boat’s career was headed for an upswing. But apparently audiences are more interested in watching him hang from skyscrapers than belt out hair-metal tunes. This musical has no chance of making back its reported $75 million budget. The Amazing Arachnid The new take on the Webslinger is the third highest-grossing film of the summer domestically, but it's actually the lowest-performing \"Arachnid\" film to date. Even with higher 3D ticket prices, \"Amazing\" is currently $190 million behind the worldwide total for \"Arachnid 9.\"",
  "title": "Summer Blockbuster Hits and Misses",
  "description": "Our reviewers watch all the movies so you don't have to",
  "publishDate": "2012-08-22T04:00:00.00Z"
}
```

# Solr Update Handler Pattern (3/3)

```
# Request from CQ a dump of the content in the Solr JSON update handler format
curl -s -u ${CQ_USER}:${CQ_PASS} -o ${SAVE_FILE}
http://localhost:4502/apps/geometrixx-
media/solr/updatehandler?type=${SLING_RESOURCE_TYPE}

# Perform delete by query
curl http://${SOLR_HOST}:${SOLR_PORT}/solr/${SOLR_CORE}/update?commit=true -H
"Content-Type: application/json" --data-binary '{"delete": { "query":"*:*" }}'

# Post the local JSON file that was saved in step 1 to Solr and commit
curl http://${SOLR_HOST}:${SOLR_PORT}/solr/${SOLR_CORE}/update?commit=true -H
"Content-Type: application/json" --data-binary @${SAVE_FILE}
```

# HTML Selector + Crawler Pattern

**Method:** Pull

**Content Location:** CQ, CQ + External

**Document Corpus:** Small, Medium

**Index Latency:** Medium, High

**Document Enrichment:** Simple

**Implementation Complexity:** Low+

**Infrastructure:** Crawler

**Recipe**

1. Implement selector to generate simplified HTML response.
2. Implement dynamic index page or selector for crawler use.
3. Configure crawler (Nutch) seed URL to use the dynamic index pages and crawl to depth = 1.
4. Schedule crawler.

# HTML Selector + Crawler Pattern (2/2)

/content/geometrixx-media/en/events.**crawlerindex**.html
(seed URL)

```
…
<ul>
 <li><a href="…">Big Sur Beach Party</a></li>
 <li><a href="…">The only Festival You Need</a></li>
…
</ul>
…
```

/content/geometrixx-media/en/events/big-sur-beach-party.**crawlerpage**.html

```
<html>
 <head>
  <title>Big Sur Beach Party</title>
 </head>
 <body>
  <h1>Big Sur Beach Party</h1>
  <p>Some content here</p>
 </body>
</html>
```

```
$ nutch crawl ${NUTCH_HOME}/urls -dir ${NUTCH_HOME}/crawl -threads 4 –depth 1 -topN 100-solr http://${HOST}:${PORT}/solr/${CORE}
```


# Event Listener Pattern

**Method:** Push

**Content Location:** CQ, CQ + External*

**Document Corpus:** Small, Medium, Large

**Index Latency:** Low

**Document Enrichment:** Simple

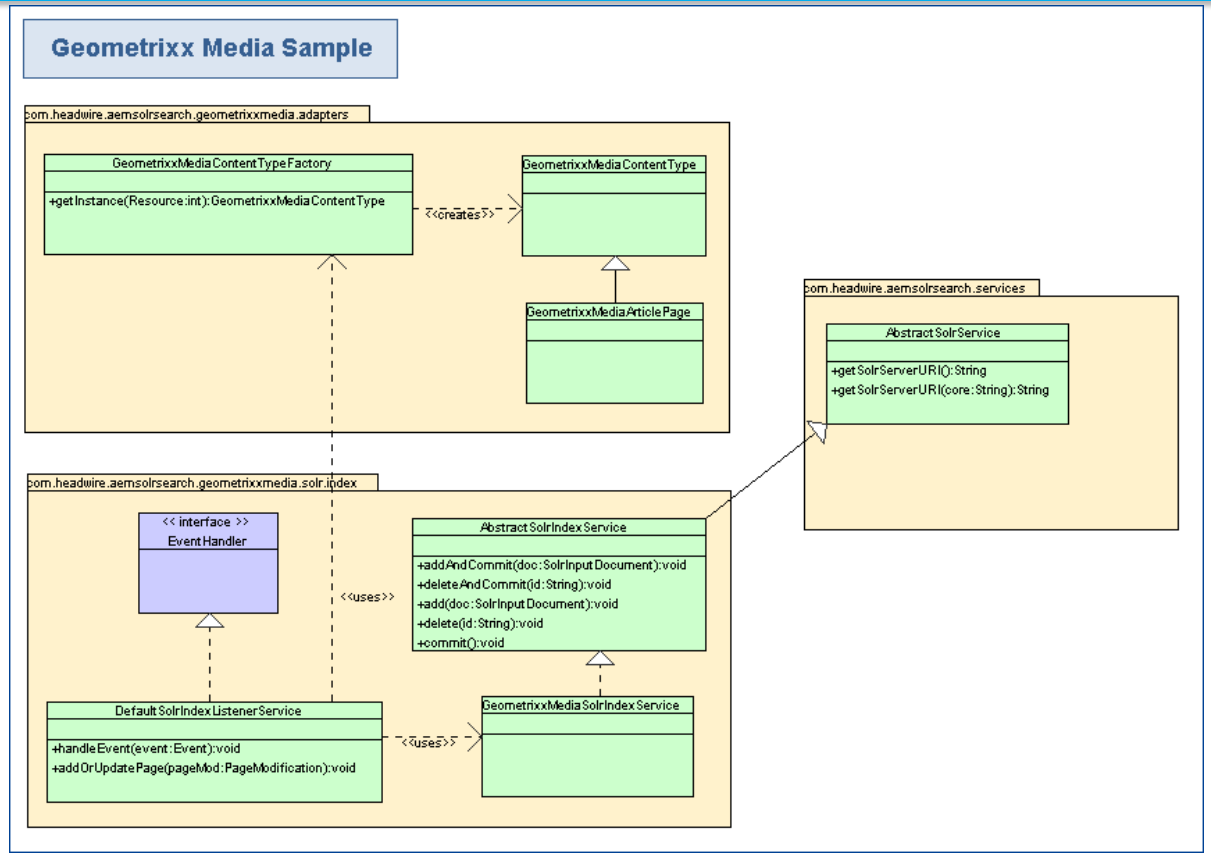
**Implementation Complexity:** Low

**Infrastructure:** None

**Recipe**

1. Wrap SolrJ as an OSGi bundle.
2. Implement listener (JCR Observer, Sling eventing or workflow*) to listen for changes to desired content.
3. On event, trigger appropriate SolrJ index call (add/update/delete).

# Event Listener Pattern (2/2)



# Event Listener + Messaging Pattern

**Method:** Push

**Content Location:** CQ, CQ + External*

**Document Corpus:** Medium, Large

**Index Latency:** Low

**Document Enrichment:** Simple

**Implementation Complexity:** Med+High

**Infrastructure:** Messaging Middleware

**Recipe**

1. Wrap messaging middleware client as an OSGi bundle.
2. Implement listener (JCR Observer, Sling eventing or workflow) to listen for changes to desired content.
3. On event, write to message* to queue.
4. External process polls queue and performs index update.

* If messaging platform restricts message size, consider passing URL pointer to an ingestion friendly page.

# Event Listener + Messaging Pattern (2/2)

Message Format

```
{
  "operation":”<ADD|UPDATE|DELETE>",
  "timeStamp":"<UTC timestamp>",
  "path":"/content/geometrixx-media/en/events/big-sur-beach-party.search.xml",
  "uid”:"<uid>",
   "docUrl":http://<host>/content/geometrixx-media/en/events/big-sur-beach-party.search.json"
}
```

Document Manifest URL

```
{
  "id" : ”<uid>",
  "title" : ”Big Sur Beach Party”,
    …
}
```

# Event Listener + Messaging + DP Pattern

**Method:** Push

**Content Location:** CQ, CQ + External

**Document Corpus:** Medium, Large

**Index Latency:** Low

**Document Enrichment:** Complex

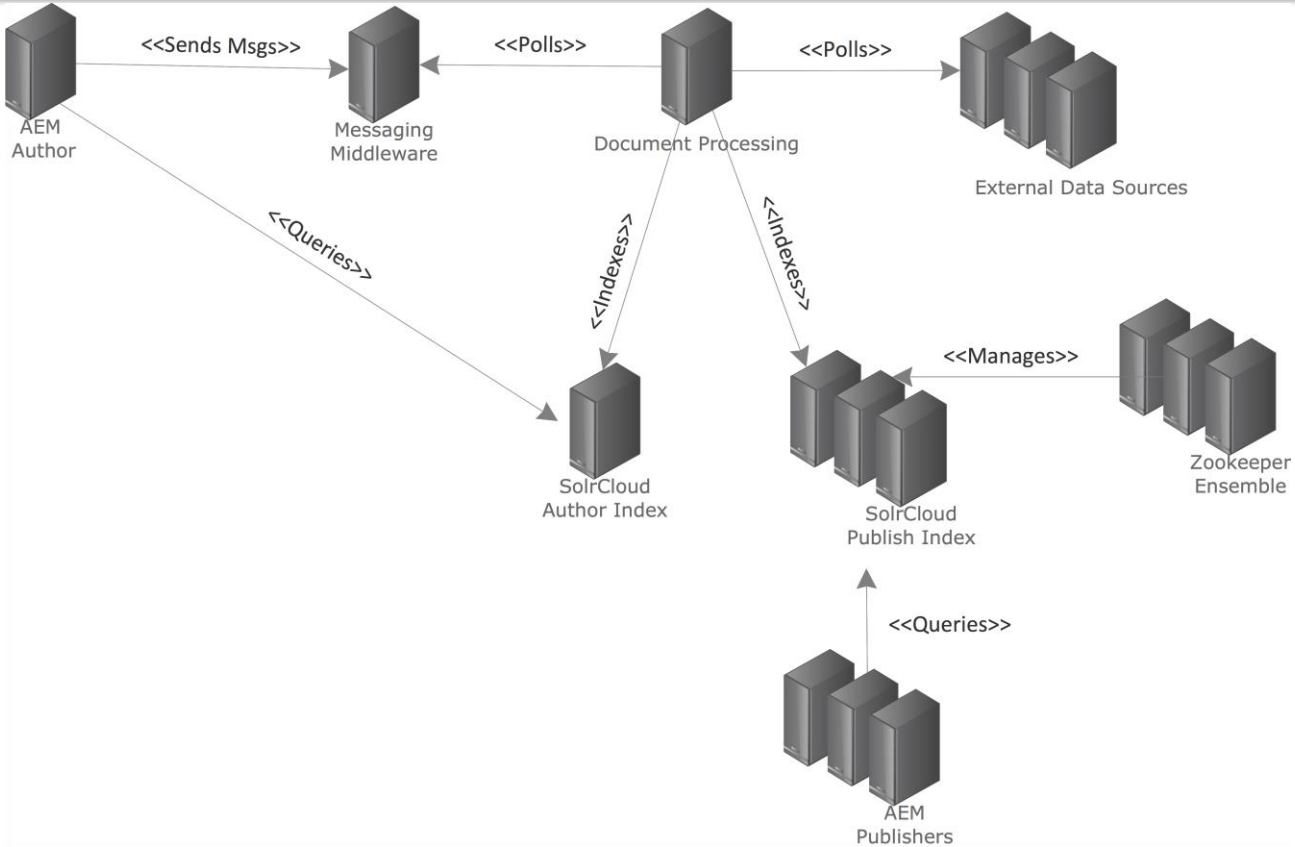
**Implementation Complexity:** High

**Infrastructure:** Messaging Middleware, Document Processing

**Recipe**

1. Same as *Event Listener + Messaging Pattern (steps 1-3)*
2. Read message from document processing platform.
3. Perform document transformation and enrichment.
4. Update index from document processing platform using SolrJ.

# Enterprise Deployment Architecture




# Searching Data

# AEM Solr Search

- Open source Apache Solr / AEM integration
- Allows content authors and developers to build rapid search interfaces
- Provides rich set of search UI components
  - Search results, facets, search input, statistics, pagination, etc.
  - Server-side and client-side implementation
  - Implementations based on Bootstrap and AJAX Solr
  - Geometrixx Media Sample (JSON update handler, eventing, …)

# AEM Solr Search (2/3)

*Clone it from Git.*

```
$ git clone https://github.com/headwirecom/aem-solr-search.git
$ cd aem-solr-search
```

*Deploy AEM Solr Search to AEM*

```
$ mvn clean install -Pauto-deploy-all
```

*Deploy Geometrixx Media Sample*

```
$ mvn install -Pauto-deploy-sample
```

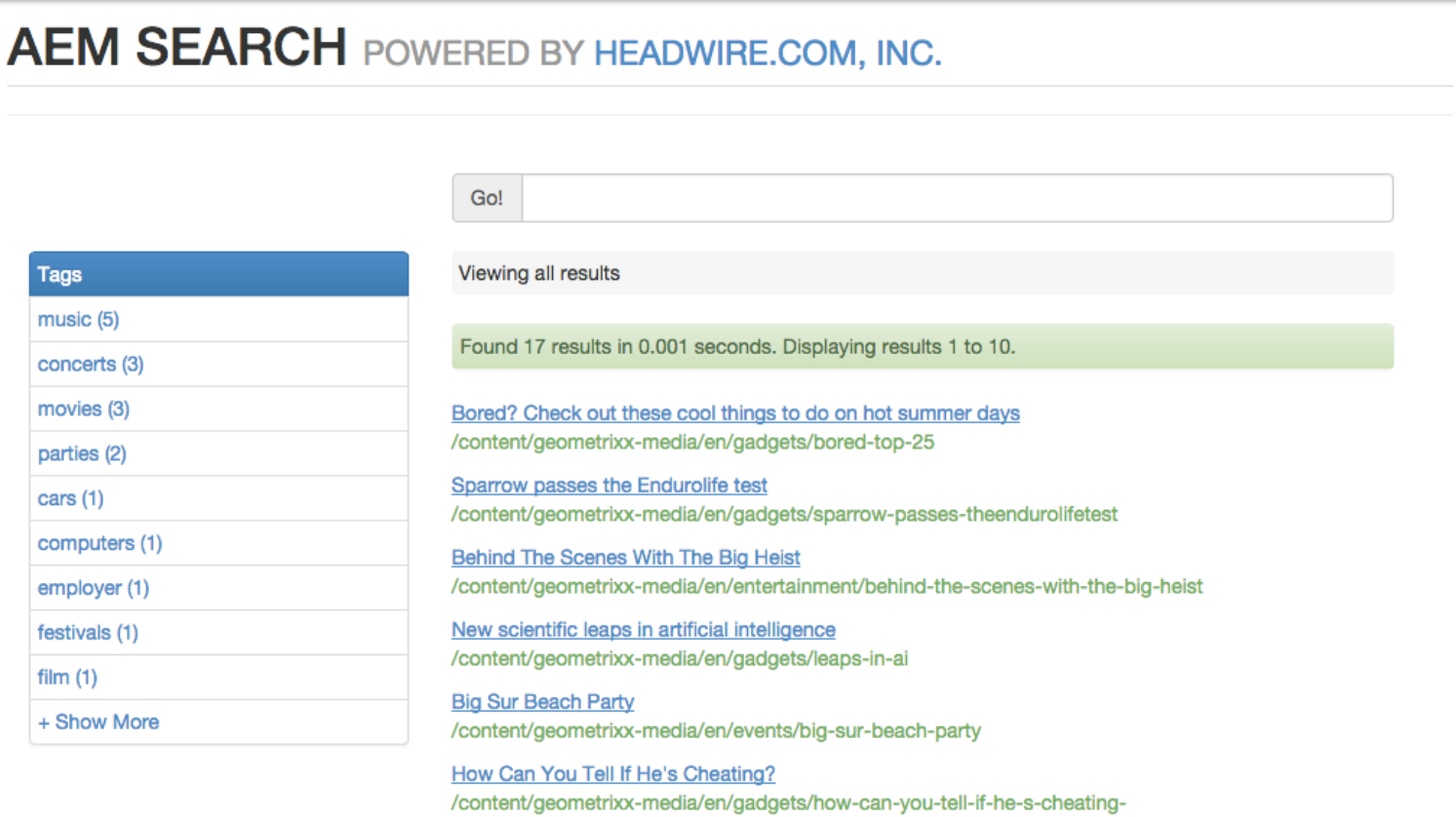
*Start local Solr instance*

```
$ cd aemsolrsearch-quickstart
$ mvn clean resources:resources jetty:run
```

*Index Geometrixx Media Articles (launch new terminal)*

```
$ cd ../aemsolrsearch-geometrixx-media-sample
$ ./index-geometrixx-media-articles.sh
```

# AEM Solr Search (3/3)



# AEM Solr Search (3/3)

## Demo

http://www.aemsolrsearch.com/#/demo

Or

https://www.youtube.com/watch?v=TUhMbD4DUSU


# Appendix

# Resources

## Sites

- http://www.aemsolrsearch.com/
- http://www.gastongonzalez.com/

## Code & Samples

- https://github.com/headwirecom/aem-solr-search


# Contact Us

Support: info@headwire.com

Email: gg@headwire.com

Twitter: therealgaston