



# AEM Solr integration

This document, its contents, and any attachments are confidential and copyrighted information intended only for the use of the designated recipient. Any copy or record of it or its contents may not be made, forwarded or disclosed without the consent of Axamit Inc.

# Solr

---



Solr is a standalone enterprise search server with a REST-like API. You put documents in it (called "indexing") via JSON, XML, CSV or binary over HTTP. You query it via HTTP GET and receive JSON, XML, CSV or binary results.

# Query and Performance

---

- REST interfaces provide for easy integration with any language.
- Multiple scoring (similarity) approaches allow for easy experimentation and relevance tuning.
- Near to Real Time (NRT) search and indexing allow to update and receive data immediately.
- A multitude of smart caching options enable exacting control over repetitive results a breeze.

# Scaling

---

- Add and remove compute capacity as needed, while letting Solr do the dirty work of load balancing and more.
- Advanced transaction log, replication and failover capabilities help minimize data loss while still serving results.

# Plugins and Extensions

---

- Language detection
- Indexing rich content (e.g. PDFs, Word)
- Clustering
- Data-import
- UIMA
- Apache Velocity

# User-experience features

---

- Pagination and sorting
- Faceting
- Autosuggest
- Spell-checking
- Hit highlighting
- Geospatial search

# Tested and proven

TESTED AND PROVEN

## Solr is trusted.

Solr powers some of the most heavily-trafficked websites and applications in the world.

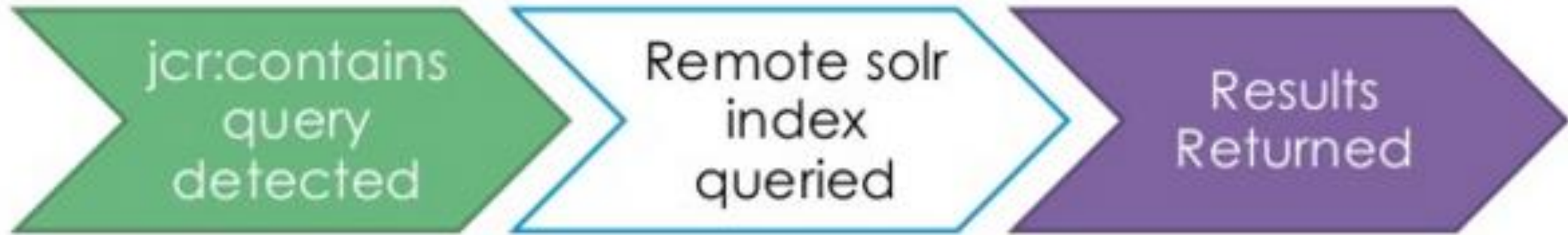


AT&T  
Ticketmaster  
Chegg  
eBay  
Magento  
Comcast

Other Notable Users  
Instagram  
Netflix  
Disney  
Internet Archive  
IBM Websphere Commerce  
MTV Networks

Buy.com  
The Echo Nest  
Adobe  
SAP Hybris  
Bloomberg  
Travelocity

# Flow



- *In oak-solr-core 1.0.1+ (AEM 6 SP1) you can add property, path & primary*
- *type restrictions to your query*



# Types of Solr that Oak uses

---

## **Embedded Solr**

Primarily used for development work. The solr instance runs within AEM and can be configured similar to remote instance.

## **Remote Solr**

Used for non-development, production level environments. Typically these instances take advantage of fault tolerant features of the Solr cloud. In many cases, existing solr instances are used.

# Solr Configuration in AEM

## Oak Solr indexing / search

path restrictions	<input checked="" type="checkbox"/>
	Des
property restrictions	<input checked="" type="checkbox"/>
	Des
primary types restrictions	<input checked="" type="checkbox"/>

## Oak Solr remote server

Description for org.apache.jackrabbit.oak.plugins.index.solr.osg

Solr HTTP URL	<input type="text" value="http://127.0.0.1:8983/solr/oak"/>
	Description for solr.http.url (sol
ZooKeeper host	<input type="text" value="127.0.0.1:9983"/>
	Description for solr.zk.host (sol
Solr collection	<input type="text" value="oak"/>
	Description for solr.collection (s
No. of collection shards	<input type="text" value="2"/>
	Description for solr.shards.no (:
Replication factor	<input type="text" value="2"/>
	Description for solr.replication.f

## Oak Solr server provider

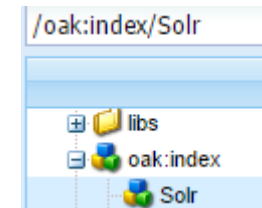
Description for org.apache.jackrabbit.oak.plugins.in

Property server.type	<input type="text" value="None"/>
	Des

**Configuration Information**

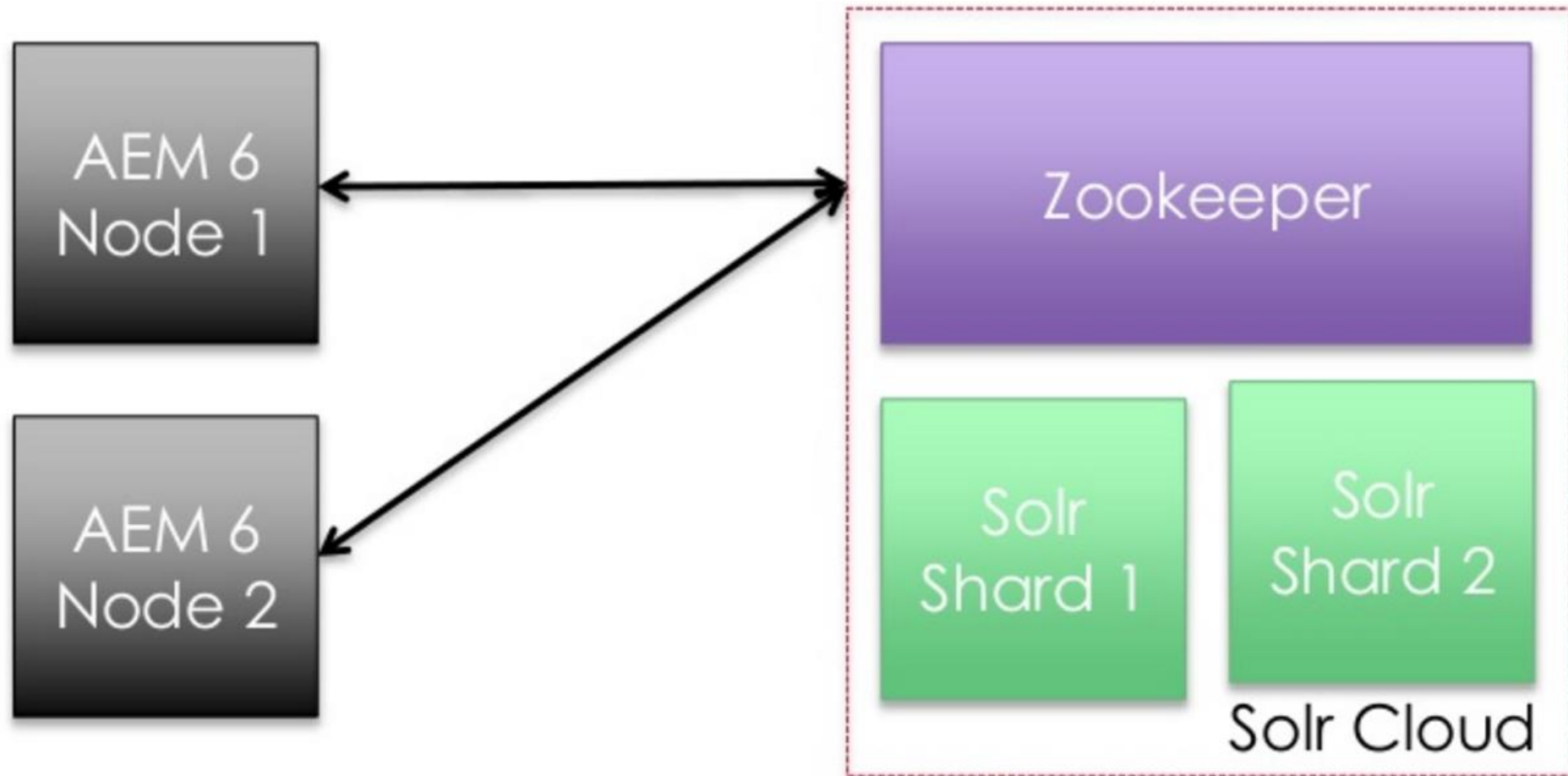
Persistent Identity (PID)

- None
- None
- Embedded Solr
- Remote Solr



Properties				Access Control	Replication	Console
	Name ^	Type	Value			
1	async	String	async			
2	jcr:primaryType	Name	oak:QueryIndexDefinition			
3	reindex	Boolean	true			
4	type	String	solr			

# Solr Conceptual Architecture with SolrCloud



# External Search Platform

---

- Supports more than one content sources
- Provide more advanced search features (faceted search, geospatial search, autosuggest, hit highlighting)

# Solr Update Handler Pattern

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).

```
{
  description: "",
  id: "/content/axamit/en/news-list/english-school",
  title: "English School",
  body: "Working with the clients from all over the world, we
international clients, we decided to introduce new language
levels intensive business English courses classes with native
Movie Nights, English Cafe, English socialising games. Yes,
last_modified: "2016-03-15T14:08:00.00Z",
sling_resource_type: "training/components/pages/oneNews",
url: "/content/axamit/en/news-list/english-school.html",
- tags: [
  "ENERGY"
]
},
r
```

# Solr Update Handler Pattern

# Request from CQ a dump of the content in the Solr JSON update handler

```
curl -s -u admin:admin -o news.json
```

```
http://localhost:4502/apps/axamit/solr/updatehandler?type=/training/components/pages/oneNews
```

# Post the local JSON file that was saved in step 1 to Solr and commit

```
curl http://localhost:8888/solr/axamitCollection/update?commit=true -H "Content-Type: application/json" --data-binary @news.json
```

**Good for initial indexing of content (high index latency)**

# Event Listener Pattern

---

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).

**Good for real-time content indexing (low index latency)**

# Resources

---

<http://www.aemsolrsearch.com/#/>

<http://lucene.apache.org/solr/>

[https://docs.adobe.com/docs/en/aem/6-1/deploy/platform/queries-and-indexing.html#par\\_title\\_5](https://docs.adobe.com/docs/en/aem/6-1/deploy/platform/queries-and-indexing.html#par_title_5)

<https://docs.adobe.com/content/docs/en/aem/6-1/deploy/best-practices/best-practices-for-queries-and-indexing.html#Solr%20Indexing>