

Securing Solr Documents with ManifoldCF

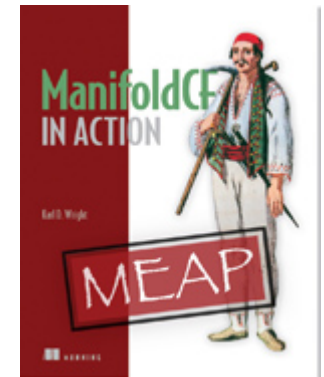
How to Enforce Repository
Authorization with Solr

What I Will Cover

- What ManifoldCF does and the problem it is designed to solve
- ManifoldCF's way of mapping repository security to documents indexed by Solr/Lucene
- A Q&A panel session describing real-world usage of the ManifoldCF security projection model

Who am I?

- I am:
 - Karl Wright (kwright@apache.org)
 - Principal Software Engineer at Nokia, Inc.
 - Formerly Principal Software Engineer at MetaCarta, Inc.
- What I do:
 - Work at Nokia on making location search better
 - Designer and original implementer of ManifoldCF
 - Author of 'ManifoldCF in Action'
 - Committer for ManifoldCF
 - Other interests include musical composition, quantum mechanics, and evolutionary biology



Let's search our repository using Solr!

- But first, we have to get our repository documents indexed by Solr
- And then... there's another obstacle... VINNY



Who is this Vinny guy??

- Chances are, you already know him
- “Vinny” protects your organization’s content
- “Vinny” prevents unauthorized users from seeing what they aren’t supposed to see
- “Vinny” isn’t going to let you index his content unless you can control access in the same way

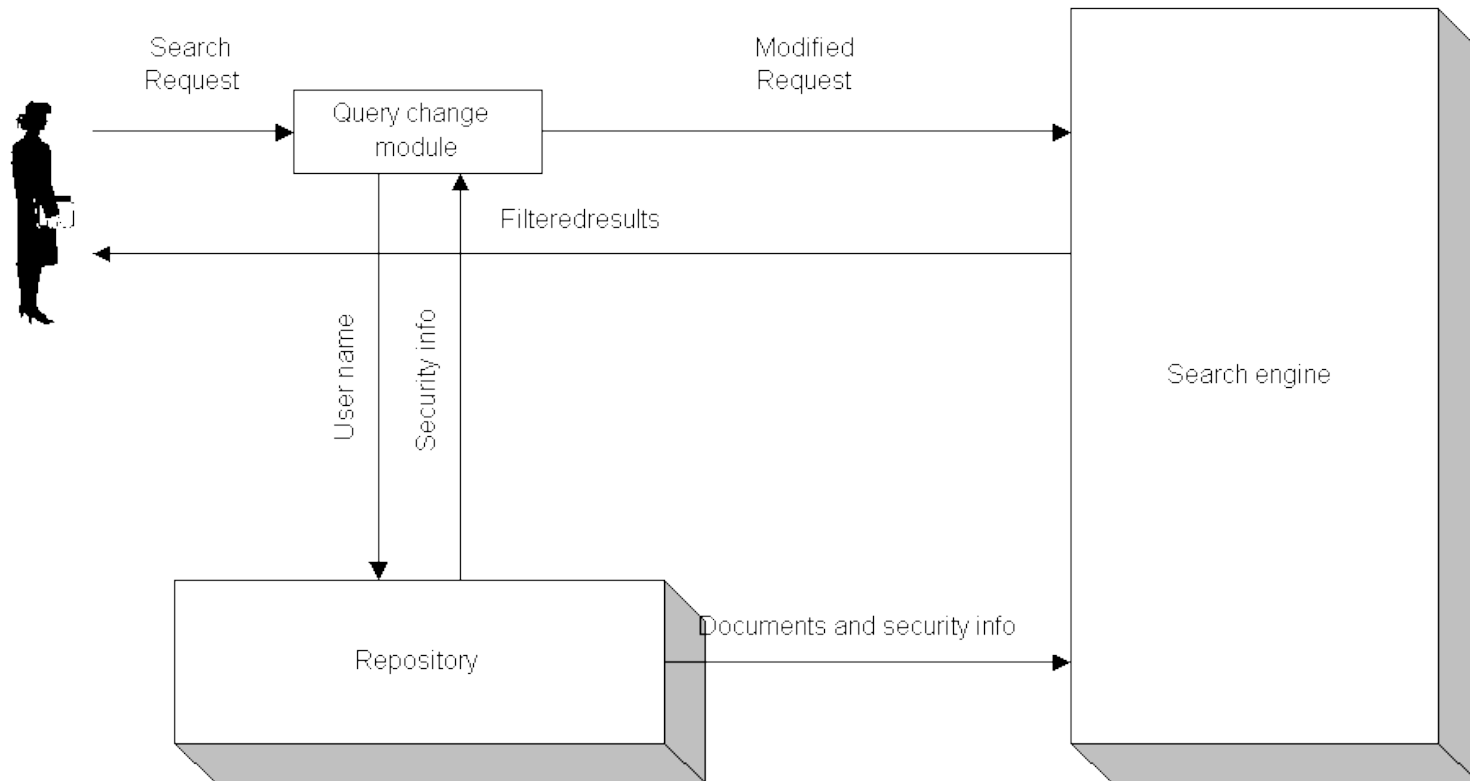


ManifoldCF to the Rescue!

- Plug-in architecture allows connectors to easily be written, if they don't exist already
- Existing repository connectors for web, RSS, JDBC, CIFS (shared file system), SharePoint, Meridio, FileNet, LiveLink, Documentum, CMIS
- Existing output connectors for Solr, GTS, and OpenSearchServer
- Includes a user-facing UI, an API, and an Authorization Service



Query Restriction Model

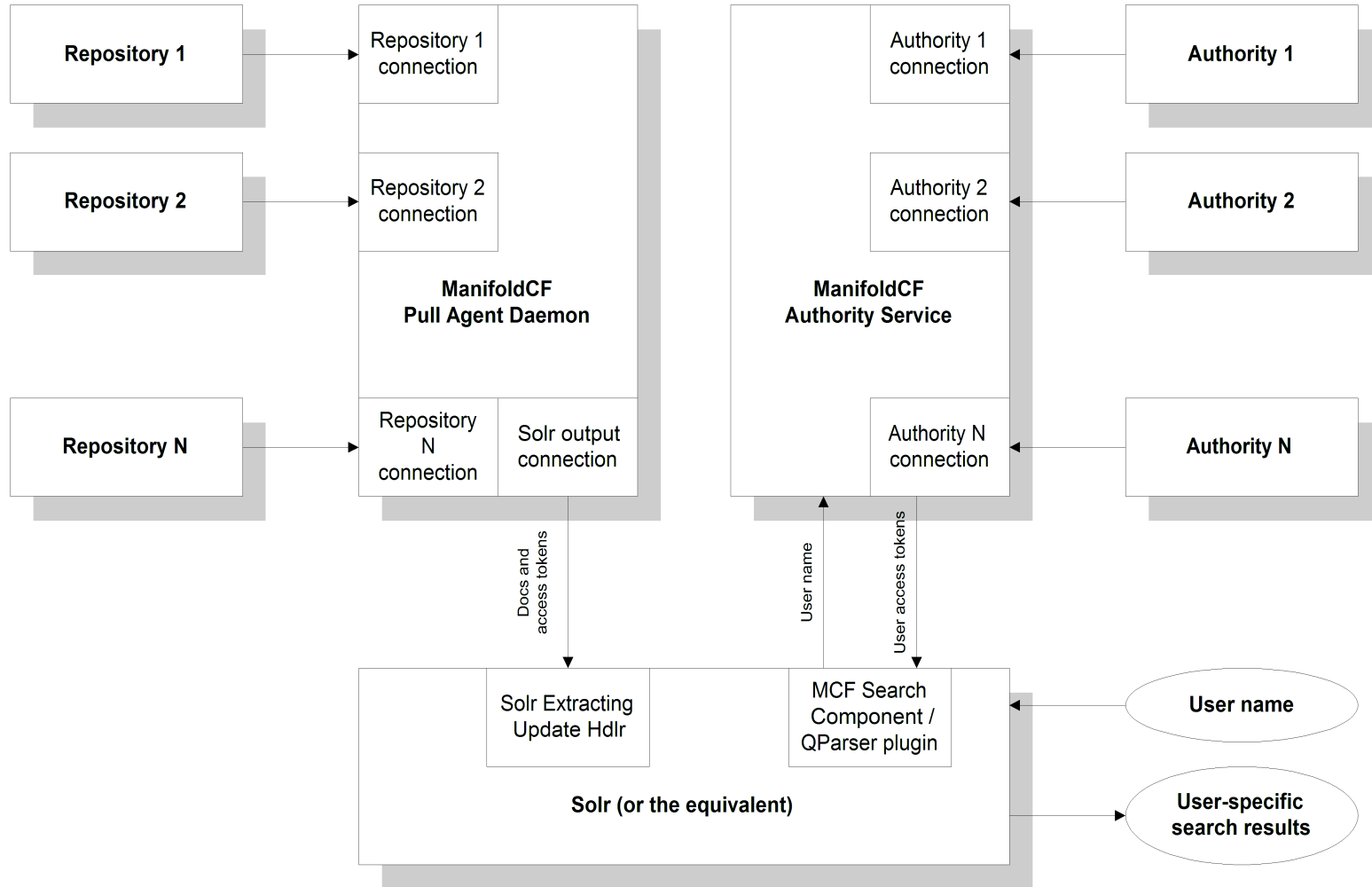


(From ManifoldCF in Action, Chapter 4. Reprinted with permission.)

How ManifoldCF Implements Query Restriction

- Document access tokens are sent to the search index along with the document content
- Separate bins for “allow” tokens, “deny” tokens – for “file”, multiple “folder”, and “share” levels
- In practice, only “file” and “share” levels are needed
- ManifoldCF Authority Service maps user names to a user’s access tokens
- Solr SearchComponent or QParserPlugin communicates with the MCF Authority Service and performs the query modification

ManifoldCF Architecture



What does the Pull-Agent daemon do?

- Pulls documents from various repositories, continuously or on a schedule, and hands them to the output search engine
- Incremental – does as little work as possible
- Also fetches and indexes each document's access tokens



What does the Authority Service do?



Ok, what does the Authority Service REALLY do?

- User names go in (user@domain)
- Access tokens come out – for all active authority connections currently defined in that ManifoldCF instance
- HTTP based, line-by-line output, with helpful hints:

```
curl http://localhost:8345/mcf-authority-service/  
UserACLs?username=foo@bar.com
```

```
UNREACHABLEAUTHORITY:The+Spanish+Inquisition
```

```
TOKEN:My+Authority:DEAD_AUTHORITY
```

```
AUTHORIZED:Null+authority
```

```
TOKEN:Null:foo%40bar.com
```

What do you have to do to Solr to make this all work?

- Add fields to the schema to contain document access tokens
 - A field for document-level “allow” tokens
 - A field for document-level “deny” tokens
 - A field for share-level “allow” tokens
 - A field for share-level “deny” tokens

Add something that authenticates a user and obtains a user name

- Add a SearchComponent or Query Parser to restrict incoming query



The Solr component is NOT where the magic is...

- Each access token returned by the Authority Service adds a clause to a BooleanQuery
- It is rare for a user to have more than one hundred access tokens – except for Documentum!!
- ManifoldCF in Action provides an example Solr SearchComponent
- `dist/solr-integration` provides a Solr SearchComponent and QParserPlugin (MCF trunk)



How are the four token types related?

- Share and document levels computed independently; an included document must pass both
- For each level, DENY tokens exclude and ALLOW tokens permit, but DENY tokens always win over ALLOW
- Special meaning for no tokens at all at a level – no ALLOW nor DENY tokens means “public” – handled by a default token in Solr
- Active Directory does it exactly the same way, oddly enough, using SIDs for tokens

Example

Document	Share allow	Share deny	Doc allow	Doc deny
Look_at_me	(empty)	(empty)	(empty)	(empty)
Very_secret	(empty)	(empty)	(empty)	T1
Not_picky	(empty)	(empty)	T1, T2, T3	T4
Really_picky	(empty)	(empty)	T1	(empty)
Insane	T1, T2	T3	T3, T2	T1
Share_ctrl'd	T1, T2, T3	T4	(empty)	(empty)

- “Not_picky” and “Share_ctrl'd” seen by the same people
- “Very_secret” seen by nobody
- “Insane” seen by people with T2 only

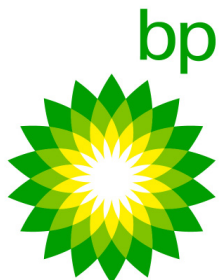
What is still missing from the picture?

- Well, getting documents and authorization info into Solr is covered...
- Getting authorization information for a user is covered...
- Modifying the search to enforce authorization is covered...
- Authentication is NOT covered!
 - **ManifoldCF does not help you with this problem – yet**
 - **Consider JAAS in Tomcat**
 - **Apache web server's mod-auth-kerb also works**

Do you think these people care about security?



U.S. AIR FORCE



StatoilHydro



Statoil



Wrap Up

- ManifoldCF provides a great way to project repository security into Solr
- ManifoldCF effectively converts repository security into an AD-like token model
- As long as you can provide the authentication, MCF and Solr can provide the rest
- Nobody ever expects the Spanish Inquisition

Our Panel Today

- Karl Wright
- Eric Pugh
- Shinichiro Abe

Sources

- ManifoldCF in Action
 - <http://www.manning.com/wright>
 - http://manifoldcfinaction.googlecode.com/svn/trunk/edition_1/security_example

Contacts

- Shinichiro Abe
 - shinichiro@apache.org
 - <http://www.rondhuit.com/apache-manifoldcf.html> (In Japanese)
- Eric Pugh
 - epugh@opensourceconnections.com
- Karl Wright
 - kwright@apache.org
 - <http://manifoldcfinaction.blogspot.com>