

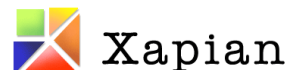
# Access control in an open source search solution



Tom Mortimer, Flax  
London Intranet show & tell: Intranet Search, 2010

# Flax

- ◆ Search engine specialists
- ◆ Formed in 2001 from the ashes of Muscat Ltd and Webtop as Lemur Consulting Ltd
- ◆ Based in Cambridge UK
- ◆ Contributors to and users of Xapian
- ◆ Recently selected as UK Authorized Partner by Lucid Imagination
- ◆ Customers include Mydeco, NLA, Durrants Ltd, Financial Times, MediaMiser, MySkreen



*Apache Lucene and Solr are trademarks of The Apache Software Foundation*

# The customer



## Tait Electronics Ltd.

- ♦ A global leader in designing and delivering radio solutions
- ♦ Customers include public safety agencies, government services, urban transport providers etc.
- ♦ Corporate services based in New Zealand - network of worldwide offices and distributors





# The job

- ◆ 12 million documents
- ◆ Various formats (MS Office, OpenOffice, PDF etc.)
- ◆ 99% English language
- ◆ On three Sun Thumpers running Solaris/ZFS
- ◆ Exported via CIFS to end users on Windows
- ◆ User access using Unix permissions and ACLs
- ◆ User authentication with LDAP
- ◆ Available globally, but less than 1000 regular users





# Customer requirements

- ◆ Search results in under 1s
- ◆ Facets
- ◆ User tagging
- ◆ Results filtered by file permissions
- ◆ Index kept up to date daily

Customer had considered a variety of commercial search engines including search appliances, but rejected these in favour of an open source solution due to cost and flexibility



# Basics: Search engine

- ◆ We chose **Xapian**
- ◆ Open source (GPL)
- ◆ Probabilistic ranking
- ◆ Fast
- ◆ Highly customisable with C++ API
- ◆ We have over a decade of experience with it



Xapian

[www.xapian.org](http://www.xapian.org)



# Basics: Indexing



- ◆ Implemented in **Python**
- ◆ Rapid development
- ◆ Readability and maintainability
- ◆ Extractors use 'headless' **OpenOffice.org** processes
- ◆ PDFs handled by **pdftotext**
- ◆ Can scan and update entire corpus in 1 day



# Basics: Front end



- ◆ Web app implemented in **Python**
- ◆ **WSGI: mod\_wsgi** on **Apache 2**
- ◆ User authentication via **mod\_authnz\_external / pam**
- ◆ Not very fast! but Xapian does all of the heavy lifting



# User tagging

- ◆ Web app writes temporary file containing the tag info
- ◆ Indexer watches directory with **inotify**
- ◆ Indexer updates document terms immediately
- ◆ Changes visible to search within seconds



# Access Control: Plan A

- ◆ Store ACLs and Unix permissions with each document in the index
- ◆ Use a Xapian MatchDecider to filter search results by evaluating permissions for each user/document
- ◆ (evaluation at search time)



# Access Control: Plan A

- ◆ Store ACLs and Unix permissions with each document in the index
- ◆ Use a Xapian MatchDecider to filter search results by evaluating permissions for each user/document
- ◆ (evaluation at search time)

## ***BUT:***

- ◆ This does not take account of permissions of parent directories
- ◆ Noticeable overhead



# Access Control: Plan B

- ◆ Check whether current user can read each file directly from the file server
- ◆ This has the advantage of behaving *exactly* like the file system
- ◆ No indexing lag
- ◆ (evaluation at search time)



# Access Control: Plan B

- ◆ Check whether current user can read each file directly from the file server
- ◆ This has the advantage of behaving *exactly* like the file system
- ◆ No indexing lag
- ◆ (evaluation at search time)

***BUT:***

- ◆ **Very slow!**



# Access Control: Plan C

- ◆ At indexing time, iterate user list for each document, and check readability with OS
- ◆ Store a term for each user with each readable document
- ◆ At search time, use this term as a Boolean filter
- ◆ (evaluation at index time)



# Access Control: Plan C

- ◆ At indexing time, iterate user list for each document, and check readability with OS
- ◆ Store a term for each user with each readable document
- ◆ At search time, use this term as a Boolean filter
- ◆ (evaluation at index time)
- ◆ Very fast! Customer was happy with this solution.

## ***BUT:***

- ◆ Would be impractical for large user lists
- ◆ Indexer lag of up to 1 day (in this installation)



# Hardware

- ◆ 1 x Dell™ PowerEdge™ R710 Rack Mount Server
- ◆ 2 x QuadCore E2550 Intel processors @ 2.26GHz
- ◆ 6 x 300GB disks
- ◆ Runs all indexing and search processes



# Result



Xapian



# What did we learn?

- ◆ Access Control is not trivial
- ◆ The first approach isn't always (usually) the best
- ◆ Compromise is essential unless the budget is infinite
- ◆ There are many other possibilities we have not explored
- ◆ Open Source can make a happy customer (but we knew that already!)

Thank you!

tom@flax.co.uk  
www.flax.co.uk  
@FlaxSearch

