

# **The pain of versions**

The  
Subversion Repository Search Engine  
(SupoSE)  
[www.supose.org](http://www.supose.org)

Web Site:

[www.soebes.com](http://www.soebes.com)

Blog:

[blog.soebes.com](http://blog.soebes.com)

Email:

[info@soebes.com](mailto:info@soebes.com)

Dipl.Ing.(FH) Karl Heinz Marbaise

# Agenda

1. The Fundamental Idea
  2. The Requirements
  3. Ideas
  4. Basic Concepts
  5. Basic Architecture
  6. The components
  7. Open Questions
  8. Roadmap
  9. Current State
- A. Examples
  - B. Performance

# 1. The Fundamental Idea

- We would like to search for different items within Subversion repositories.

– Why and How?

# 1. The Fundamental Idea

- We don't know the particular revision number
- We don't know the range of time
- We don't know which file etc.
- We don't know in which file in which revision etc.
- ...

# 1. The Fundamental Idea

- But what we know..
  - It must be in the Repository

Somewhere ;-)

# 1. SupoSE was Born...

- The **S**ubversion **R**epository **S**earch **E**ngine....
  - SupoSE for short....

## 2. The Requirements

- In which Revision the Ticket #76 has been solved ?
  - You have to search within the log messages of all revisions.

Note: This only works if you put in the needed information into the log message.

## 2. The Requirements

- Which Tags or Branches did or do exist within the current project?
  - Search for directories in all folders and revisions.
    - This needed to find deleted folders (e.g. Tags or Branches) as well.

## 2. The Requirements

- In which documents did we used the term(s) “...” ?
  - Search within the contents of the versioned items in all revisions and all folders (branches/tags/trunk).

## 2. The Requirements

- In which file did we used the method “executeTestXYZ” ?
  - Search within the contents based on context sensitive informations (parsed files of particular type).
  - For example Java, Perl, Python, Ruby ....files.

## 2. The Requirements

- Where do we use the property name „xyz...“?
  - Search for property names
- Which files/revisions etc. do have the property “xyz...” with the particular value “content”?
  - Search for particular property values

## 2. The Requirements

- The search process shouldn't be limited to a single Repository.
  - In usual industrial setup's you will find multiple Subversion Repositories.

# 3. Ideas

- If we would scan the whole Repository every time we do a query it would be:
  - to slow....
  - it will produce a high load on the repository server.

So this is no option.

# 3. Ideas

- We have basically two phases:
  - Initial Phase
    - Reading the content from the Repository and indexing it.
  - Update Phase
    - Read the changed/added contents of the Repository and indexing it.

# 3. Ideas

- We need to do a full-text search:
  - Many search engines working this way.
  - e.g. The Eclipse Search works the same way...
  - And many others too...

# 3. Ideas

- How could we update the index?
  - Using Hook scripts to update the indexed informations
    - Pro:
      - Only if something changes

# 3. Ideas

- How could we update the index?
  - Using Hook scripts to update the indexed informations
    - Con:
      - Slow down commit performance
      - Need to change the Repositories
      - May be we don't have access to repository server.

# 3. Ideas

- Indexing the Repositories based on the existing access permission of SVN users.
  - Pro:
    - No need to change the repositories.
  - Con:
    - Not everything can be indexed.
    - Performance

# 3. Ideas

- Scan the repositories based on file:/// access.
  - Pro:
    - Very fast
    - No need for authorization
      - We can scan everything

# 3. Ideas

- Scan the repositories based on file:/// access.
  - Con:
    - Installation on the SVN Repository server
    - Load of the SVN Server (peek load for the initial phases).
    - Permissions on items in repository.

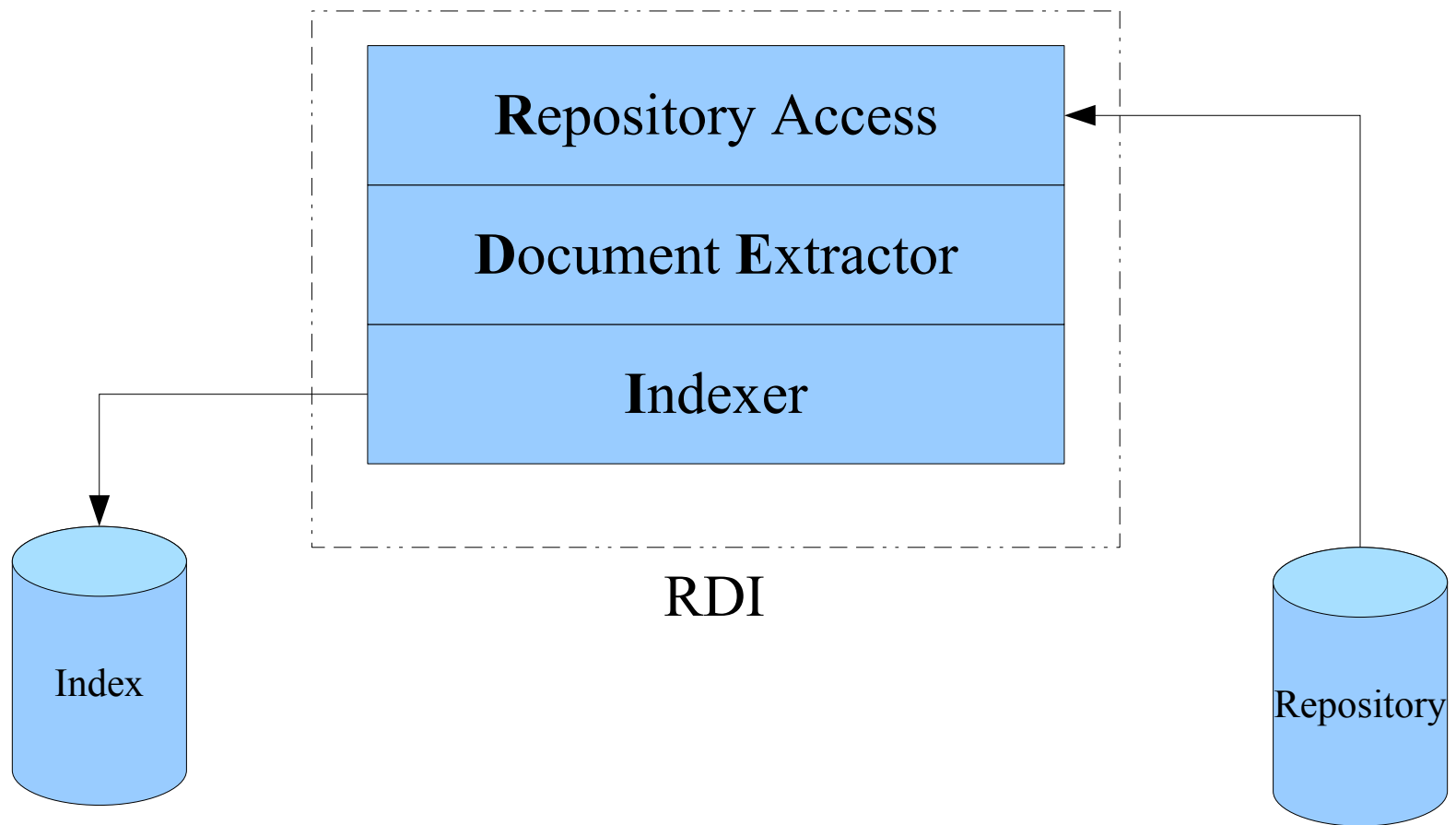
# 4. Basic Concepts

- Scan the repositories and indexing the information we need.
  - Use the file:/// protocol to access the Repository as preferable method.
  - Use other protocols (http, https or svn) if needed.

# 4. Basic Concepts

- Scan on a scheduled base for example daily or hourly etc.
- Should be made configurable.

# 5. Basic Architecture



# 5. Basic Architecture

- The Repository Access
  - Read information from the Repository.
    - Revisions
    - log messages
    - filenames, folders
    - properties
    - contents of files

# 5. Basic Architecture

- The Document Extractor
  - Extract information from different kind of document types which can easily be indexed.
    - PDF,
    - HTML, XML,
    - OpenOffice, MS Office
    - etc.

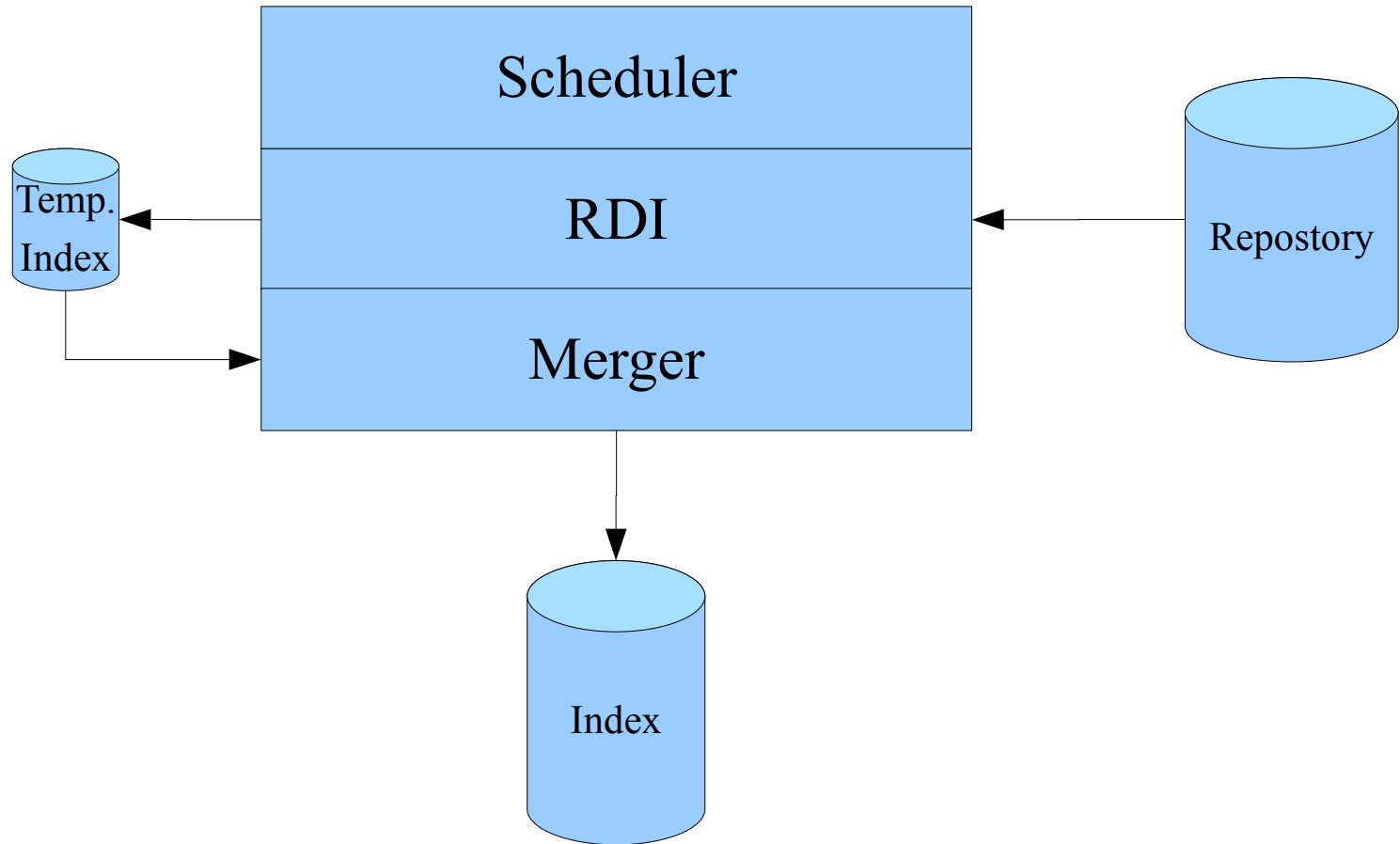
# 5. Basic Architecture

- The Document Extractor
  - Archive types like .tar.gz, zip etc. are supported in two ways.
    - Just use the filename
    - Extract the contents of the archives and extract the information as well.

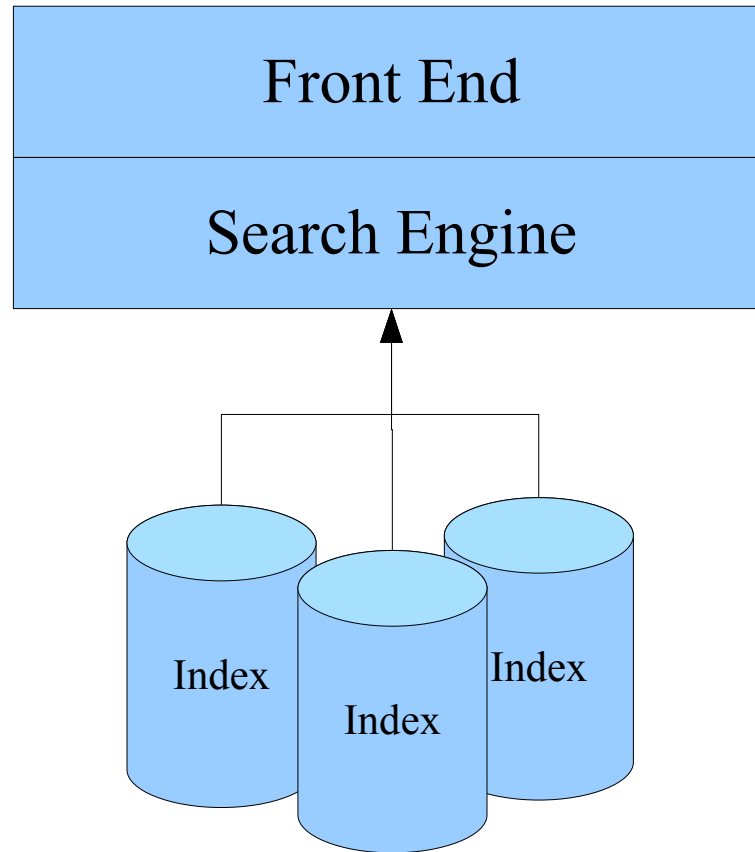
# 5. Basic Architecture

- The Indexer
  - File name, folder name
  - Contents as the extractor has delivered it.
  - Log message
  - Revision number
  - Meta Information like properties etc.

# 5. Basic Architecture



# 5. Basic Architecture



# 6. The components

- Accessing the Subversion Repository via Java only:
  - SVNKit Library
    - No need to install Subversion client
    - No external Working Copy etc. needed.

# 6. The components

- Full Text searching capabilities:
  - Apache Lucene
    - Searching capabilities
    - Indexing capabilities
    - Query language

# 6. The components

- Scheduled running of Jobs:
  - Quartz Framework
    - Cron like execution of Jobs etc.
    - No external Configuration needed.

# 6. The components

- Access and extract information from different file types:
  - Tika Framework
    - Word, Excel, PowerPoint, PDF, OpenOffice, Archives like zip, tar.gz, tar.bz2 etc.

# 6. The components

- Parsing of different Languages:
  - Java
    - via existing Java Grammar based on ANTLR 3.0

# 7. Open Questions

- Security for the indexed results
  - Authorization of the Search Engine
- What about restrictions for the search results ?
- What about property changes?
  - How do we get informed about them? Hook Scripts ?

# 7. Open Questions

- What if a repository has path-based authorization and what will happen if this has been changed?
  - What about the already indexed informations?
  - What about the search result?

# 8. Roadmap

- Releases
  - 0.7.0 (Branch)
    - Refactoring for SOAP/RESTlet and Web-Front-End
  - 0.6.2 Calypso (Currently working on it).
    - Bug Fixing some parts, Update to Tika 0.5 / SVNKit 1.3.1 / Java 1.6

# 8. Roadmap

- Recognize renaming of files/folders.
- Improve/simplify search for daily usage
- Improve/simplify configuration of the indexing processes

# 8. Roadmap

- GUI
  - Web based
- May be PlugIn's
  - trac, Eclipse, Redmine etc.
- Enhance documentation (DocBook Maven?)
- Enhance Command line interface
  - Better output etc.

# 8. Roadmap

- Make the whole part part runnable in Tomcat/JBoss/Glassfish etc.
- Performance ?
- Clustering?

# 9. Current State

- Currently Command Line Based only.
- Indexing of a single or multiple (scheduled) repositories working
  - Results can be stored into different destination indexes
- Searching currently only via command line or via Luke (Swing)

# A. Examples

- Scanning of a single Repository

suppose

scan

--url URL\_to\_Repository

--create

--index index.Repos

# A. Examples

- Which tags existing in SupoSE Repository?

```
supose  
  search  
  --index index.Supose  
  --query "+tag:*
```

# A. Examples

- Do exist Word files in this repository?

suppose

search

--index index.Supose

--query "+filename:\*.doc"

# A. Examples

- What is part of revision 100 of the particular repository?

suppose

search

--index index.Su

--query "+revision:100"

# B. Performance

- Currently the scan of the SupoSE repository itself (with 112 Revisions) via http:// (Internet)
  - This has taken ca. 25 Minutes ;-(
- A scan of Repository (2.8 GiBi) with 12.168 Revisions via file:/// protocol took ca. 37 minutes.
- 40222 Repos via file protocol took 52 minutes.

# On-line Sources I

- [1] Homepage SupoSE
  - <http://www.supose.org>
- [2] SVNKit pure Java Subversion Library
  - <http://www.svnkit.com>
- [3] Quartz Framework
  - <http://www.opensymphony.com/quartz/>
- [4] ANTLR
  - <http://www.antlr.org>

# On-line Sources II

- [5] Lucene Framework
  - <http://lucene.apache.org>
- [6] Tike Framework
  - <http://lucene.apache.org/tika/>

# Questions?

SubConf2009@soebes.com

- Thank you for your attention.