



# Beagle Desktop Search and Activity Based Metadata

By Raluca Paiu



# Outline

- Current Approaches to Desktop Search
- Beagle Desktop Search
- Extending Beagle with Metadata Desktop Search
- Project's Current Status



## Current Approaches to Desktop Search [1]

- **Google desktop search** [<http://desktop.google.com>]

Finds:

- Emails (Outlook / Outlook Express)
  - Files (Text, Word, Excel, PowerPoint)
  - Web History (Internet Explorer)
  - Chats (AOL Instant Messaging)
- **MSN desktop search application** [<http://beta.toolbar.msn.com>]



## Current Approaches to Desktop Search [2]

- **Spotlight Search**

[<http://www.apple.com/macosx/tiger/spotlight.html>]

- Supported File Formats:

- Mail, Address Book contacts
- Folders / directories
- Files (txt, rtf, pdf, doc, xls, ppt)
- Applications
- Photoshop images
- Video & audio files (MP3, MOV, AAC)
- Images (JPEG, GIF, TIFF, PNG, EXIF)

- *Incorporates semantics*: it uses explicit information, such as file size, creator, last modification date, metadata embedded into specific files.



## Current Approaches to Desktop Search [3]

- **Beagle desktop search [<http://gnome.org/projects/beagle>]**
  - Open source project for Linux
  - Searches for:
    - Documents (txt, rtf, pdf, doc, ppt, sxw, sxi, sxm)
    - Emails
    - Web History (html)
    - IM/IRC conversations
    - Source code (java, c, c++, c#, python)
    - Images (jpg, png)
    - Music files (mp3, ogg, flac)
    - Applications

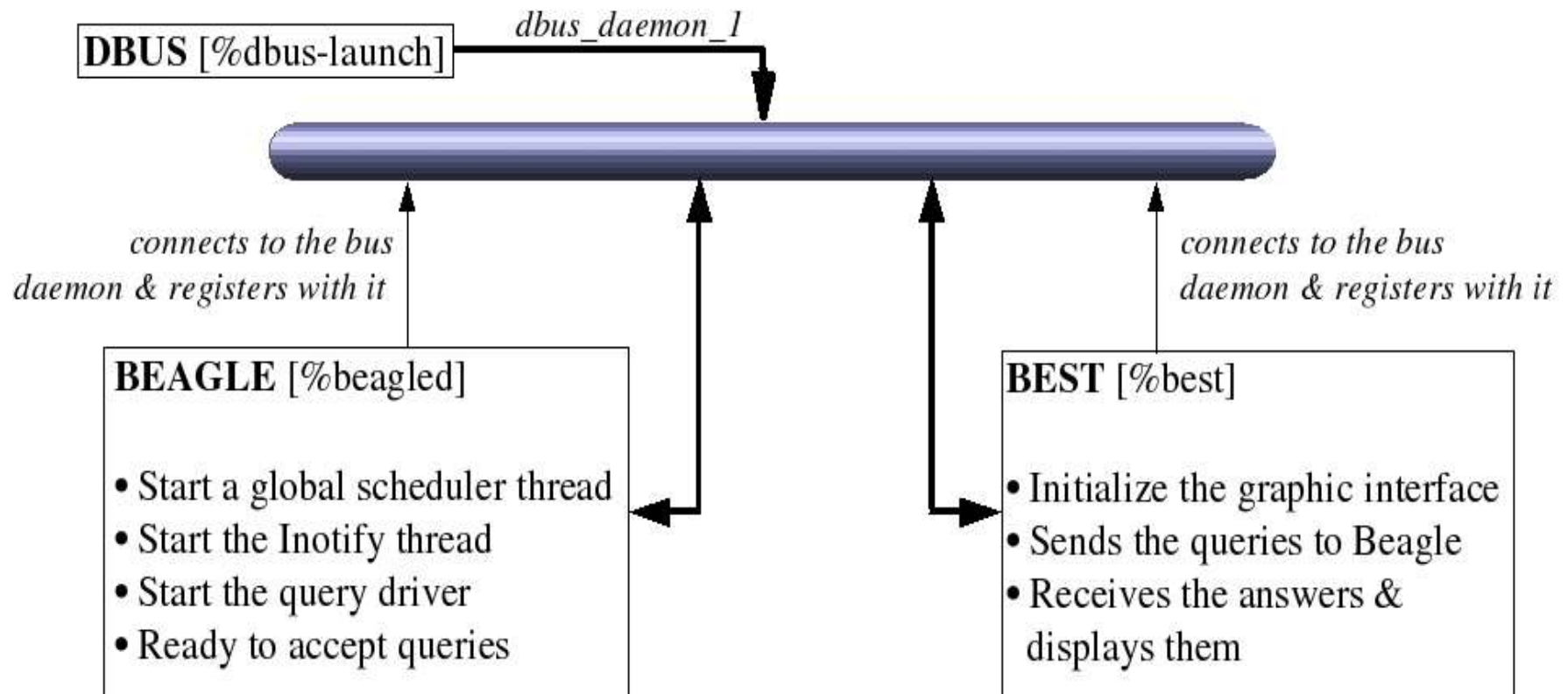


## Beagle Desktop Search [1]

- Search tool for Linux
- Written in C#
- Uses *Mono* and *Gtk#*
- Indexing is handled by *Lucene.Net*, a C# port of the *Lucene* indexer



## Beagle Desktop Search [2] - Architecture





## Beagle Desktop Search [3] - DBUS

- *D-Bus* – library that provides one-to-one communication between any two applications
- *dbus-daemon-1* – application that uses this library to implement a bus daemon
- 2 standard message bus instances:
  - *system-wide message bus*
    - usually installed as “messagebus” service
    - used to broadcast system events (adding/removing devices)
  - *per-user-login-session message bus*
    - started each time a user logs in
    - used for various inter process communication among desktop applications



## Beagle Desktop Search [4] – The Inotify Thread

- return a file descriptor for “/dev/inotify” in read-only mode
- create a thread, “snarf-thread”, which:
  - tries to read from the file descriptor pointing to “/dev/inotify” every 15 seconds, with a timeout of 1 second for reading
  - if some events occurred, stores them into a buffer
  - moves all the events from the buffer into a queue of events, “event\_queue”
- create a thread, “dispatch\_thread”, which:
  - waits at the “event\_queue”
  - dispatches the events to the appropriate “thread workers”

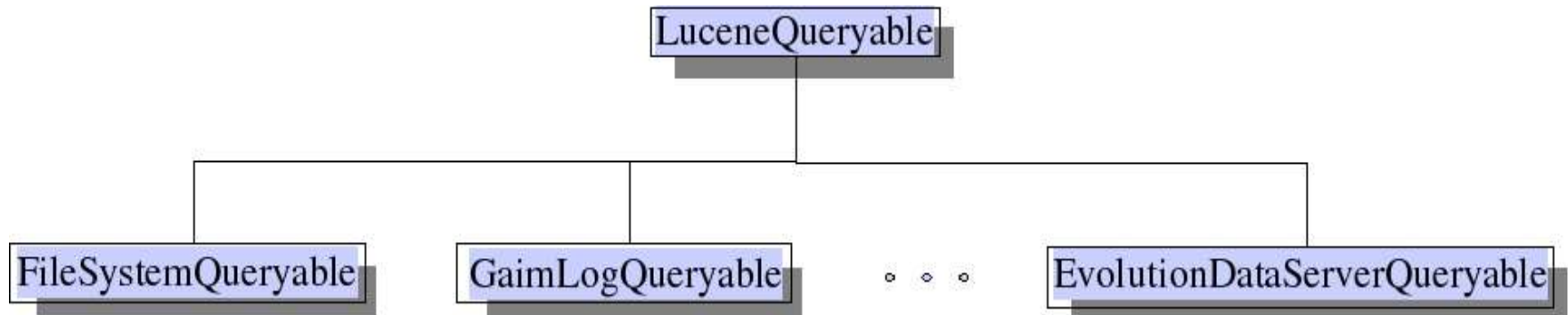


## Beagle Desktop Search [5] – The Query Driver (1)

- **Construct a QueryDriver**
  - assemble a Queryable object for each supported type
  - stick the newly created objects to the list of queryables (*Files, IMLog, Mail, WebHistory, Google, EvolutionDataServer*)
- **Start the QueryDriver**
  - every object from the list of queryables starts its own thread
  - every queryable object waits to receive notifications and treats the events accordingly



## Beagle Desktop Search [6] - The Query Driver (2)



- *FileSystemQueryable*, *GaimLogQueryable*, etc. are subclasses of the *LuceneQueryable* class
- **starting the QueryDriver marks the start of indexing (Lucene.Net)**



## Beagle Desktop Search [7] - Indexing

- *Advantages of Lucene* over other search engines:
  - ease of use
  - rapid implementation
  - flexibility
- Lucene takes a slightly different approach from other search engines: instead of maintaining a single index, *it builds multiple index segments and merges them periodically*



## Extending Beagle with Metadata Desktop Search [1]

- Documents on the desktop are not linked like in the web model →  
**The result ranking is poor or even inexistent**
- We can profit from the implicit and explicit semantic information available in:
  - emails
  - folder hierarchies
  - browser cache, etc.



## Extending Beagle with Metadata Desktop Search [2]

- Generate input metadata: *event triggered metadata generation*
- Main characteristics of our desktop search architecture:
  - metadata generation
  - indexing on-the-fly (based on events)
    - triggered by events generated upon occurrence of file system changes
    - notification functionality provided by the *inotify-enabled linux kernel*



## Extending Beagle with Metadata Desktop Search [3]

### Metadata generator applications

- Depending on the type and context of the events, metadata generation is performed by the appropriate metadata generator applications:
  - **Email Metadata Generator**
  - **File Metadata Generator**
  - **Web Cache Metadata Generator**



## Extending Beagle with Metadata Desktop Search [4] Email Metadata Generator

- built on top of the JavaMail API
- incoming mails are processed into a class derived from the *Message* class defined in JavaMail
- generated metadata for the incoming mails include informations like:
  - *Sender, Recipients*
  - *Subject and Body*
  - *Date when the email was sent*
  - *Attachments*
- metadata are stored as RDF using the Jena toolkit



## Extending Beagle with Metadata Desktop Search [5] File Metadata Generator

- Implemented in Java, and uses the JWNL API
- Generated metadata include informations about:
  - *Type of the file*
  - *Name*
  - *Date of creation*
  - *Date of last change*
  - *Location of file on the disk*
  - *WordNet additional metadata for the file name and the path to the file*
- Annotations are stored as RDF files

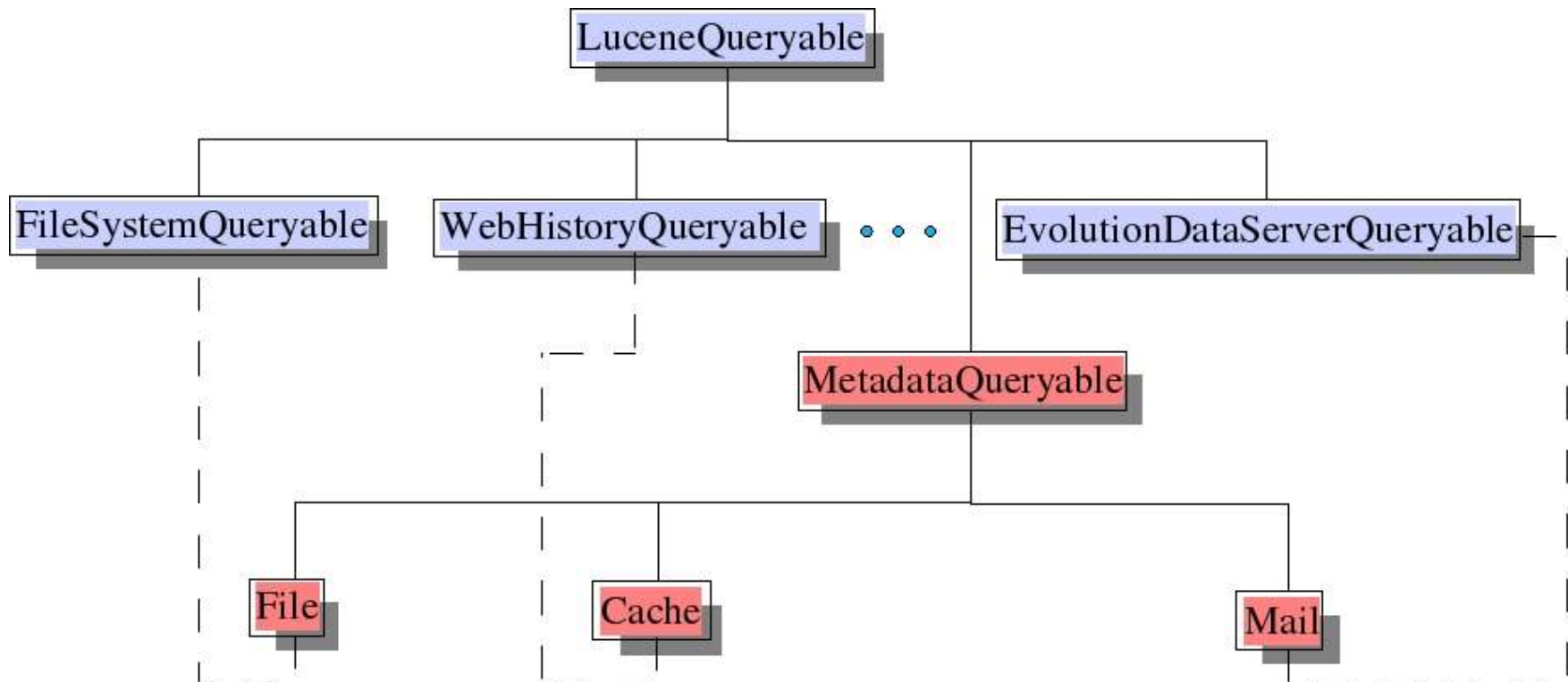


## Extending Beagle with Metadata Desktop Search [6] Web Cache Metadata Generator

- Indexing – triggered by browsing pages which are not in the cache
- Annotations include:
  - Access date
  - Connections between web pages
    - From which page did the user arrive at the current one
    - Which hyperlinks of the current page are traversed
- Generated metadata are stored as RDF files

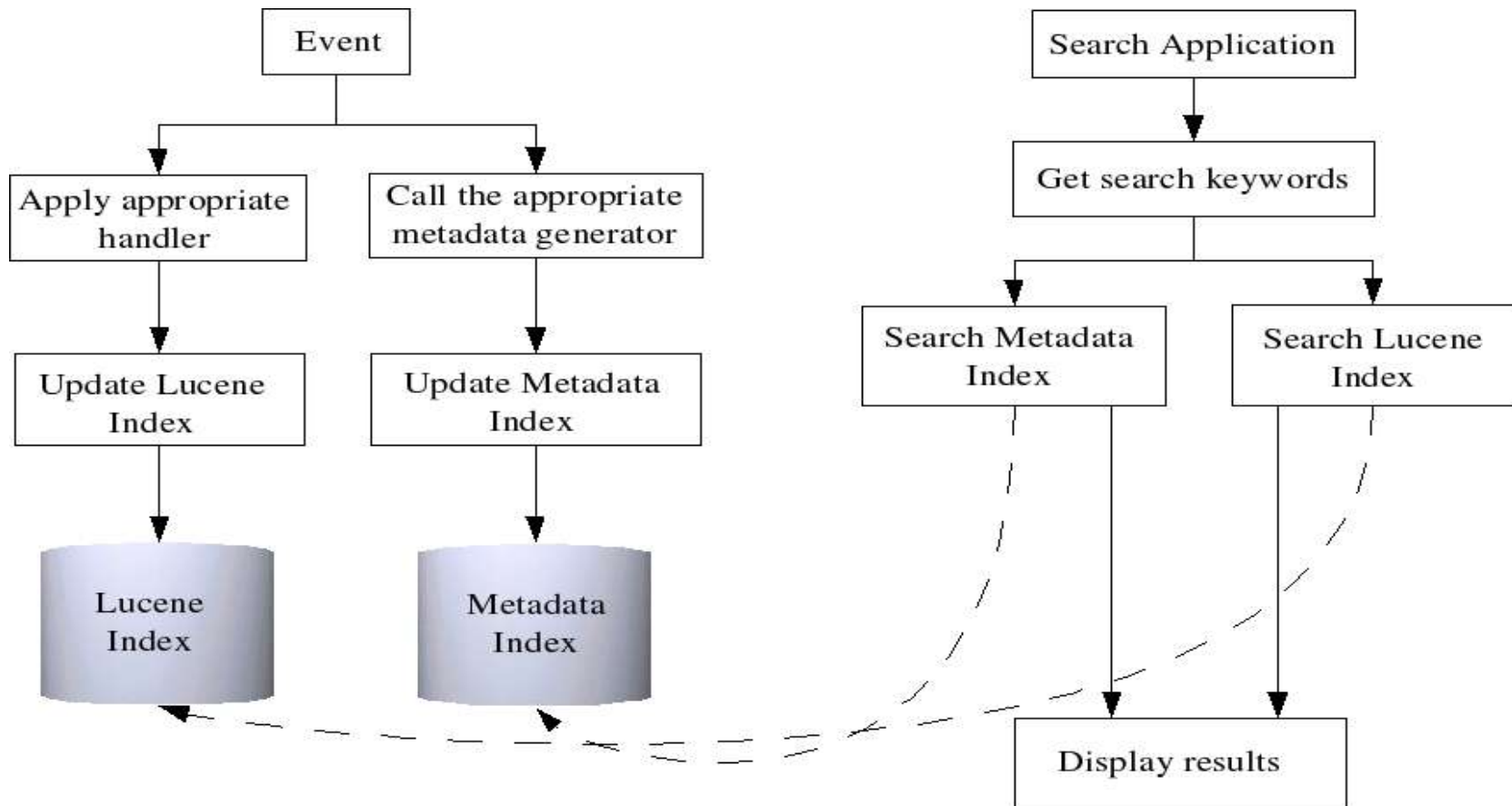


## Extending Beagle with Metadata Desktop Search [7]



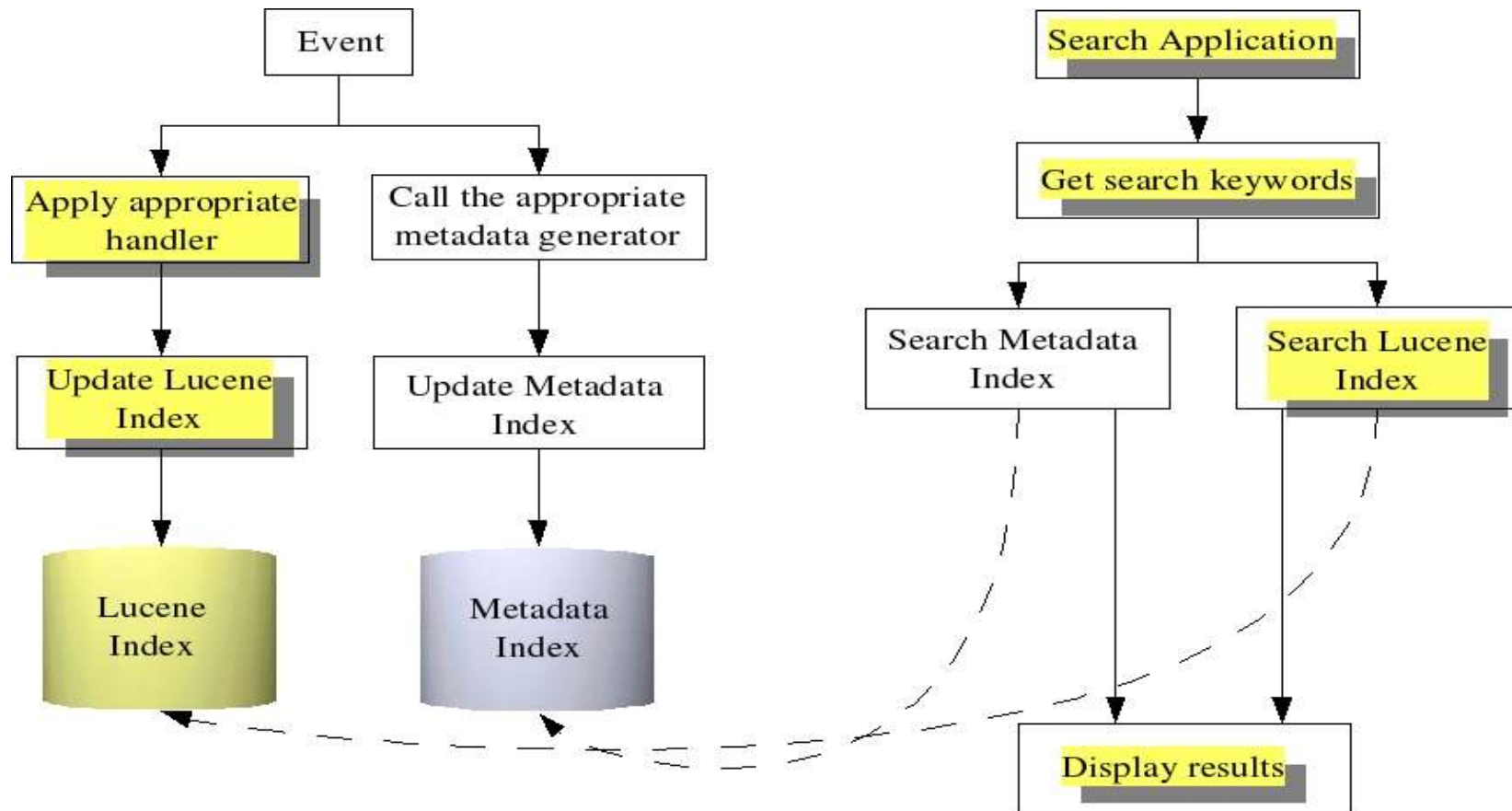


## Extending Beagle with Metadata Desktop Search [8]



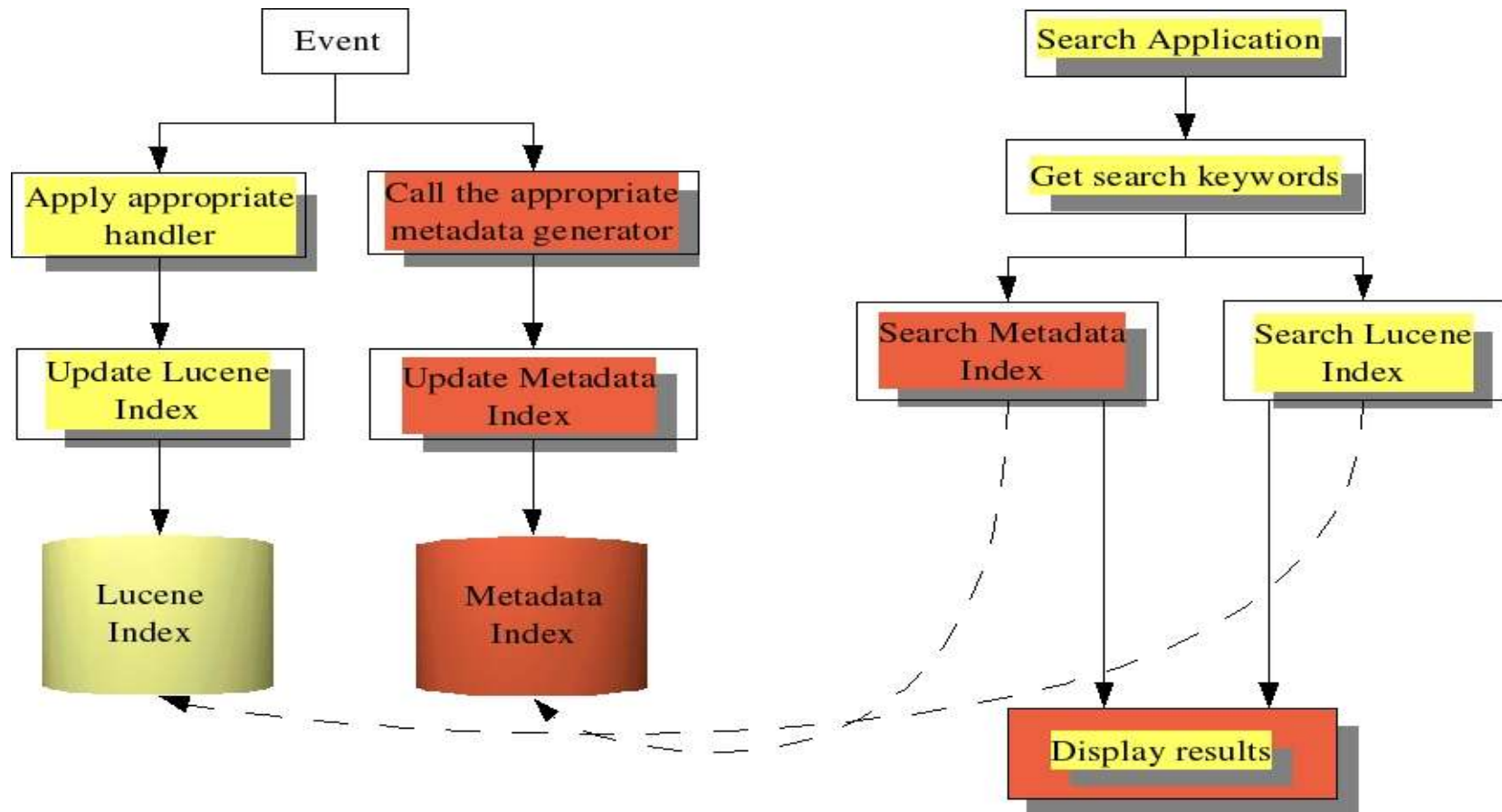


## Extending Beagle with Metadata Desktop Search [8]





## Extending Beagle with Metadata Desktop Search [8]





## Project's Current Status [1]

- **Beagle extended with metadata annotations for *.tex* files**
  - Java application which produces the annotations for every *.tex* file
    - takes as input the file
    - reads the first 100 lines
    - extracts the author and the title
    - creates an *.xml* file with the annotations
  - Upon receiving an inotify event regarding a *.tex* file, call the Java application, which creates the annotations or updates the old ones (if the “*.tex*” file is deleted, the annotation file is also deleted)
  - *.tex* files, as well as *.xml* files are included in the result sets



## Project's Current Status [2] - Implementation Details (1)

- **Create a new class, “MetadataQueryable.cs”, subclass of the “LuceneQueryable.cs”**
  - when a new thread is started, it composes a new delegate onto the “Event” field from the Inotify class
    - when an inotify event occurs the delegate is invoked
    - if the event is about a .tex file, call the Java application
- **Create two new filters:**
  - TeX Filter
  - XML Filter



## Project's Current Status [3] – Implementation Details (2)

- **Creating a new Filter type:**
  - Adding a new MIME type to the supported MIME types of Beagle (*text/xml*, *text/x-tex*)
  - XML and TeX Filters:
    - make use of the Text Filter (already implemented in Beagle)
    - create an external process which transforms the “.xml” and the “.tex” files into “.text” files with the aid of “cat” and “detex” respectively
    - redirect the output of the external process to a StreamReader
    - read the output as plain text



## Project's Current Status [4] – Displaying Results

The screenshot shows the 'Bleeding-Edge Search Tool' window. At the top, there is a search bar with the text 'Activity Based Metadata' and a 'Find' button. Below the search bar, there are five search results listed, each with an icon, a title, a 'Last modified' date, and three action buttons: 'Open', 'Send to..', and 'Reveal in file manager'.

Icon	File Name	Last modified	Actions
	<b>DesktopSearch.tex.beagle.rdf.xml, in folder v0.31</b>	Today, 16:46	Open, Send to.., Reveal in file manager
	<b>DesktopSearch-15-12.tex, in folder v034</b>	January 3, 2:13 PM	Open, Send to.., Reveal in file manager
	<b>DesktopSearch-15-12.pdf, in folder v034</b>	December 15 2004, 5:41 PM	Open, Send to.., Reveal in file manager
	<b>DesktopSearch.tex, in folder v0.30</b>	January 3, 5:25 PM	Open, Send to.., Reveal in file manager
	<b>DesktopSearch.tex, in folder v0.31</b>	January 3, 2:12 PM	Open, Send to.., Reveal in file manager

At the bottom of the window, there is a status bar that reads 'Results 16 through 20 of 22 are shown.' and two buttons: 'Show Previous Results' and 'Show More Results'.



Thank You !