Strigi in KDE4

the power of indices

Jos van den Oever

History of free desktop search

Age of Free Computing

1985

GNU project

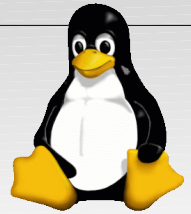
GPL

grep

find

locate

1990



1995

Age of Internet Search

kfind

2000

Age of Desktop Search

libferris

2005

History of search and semantics in KDE

1996: KFind

2001: KFileMetaInfo

2005: start of Kat

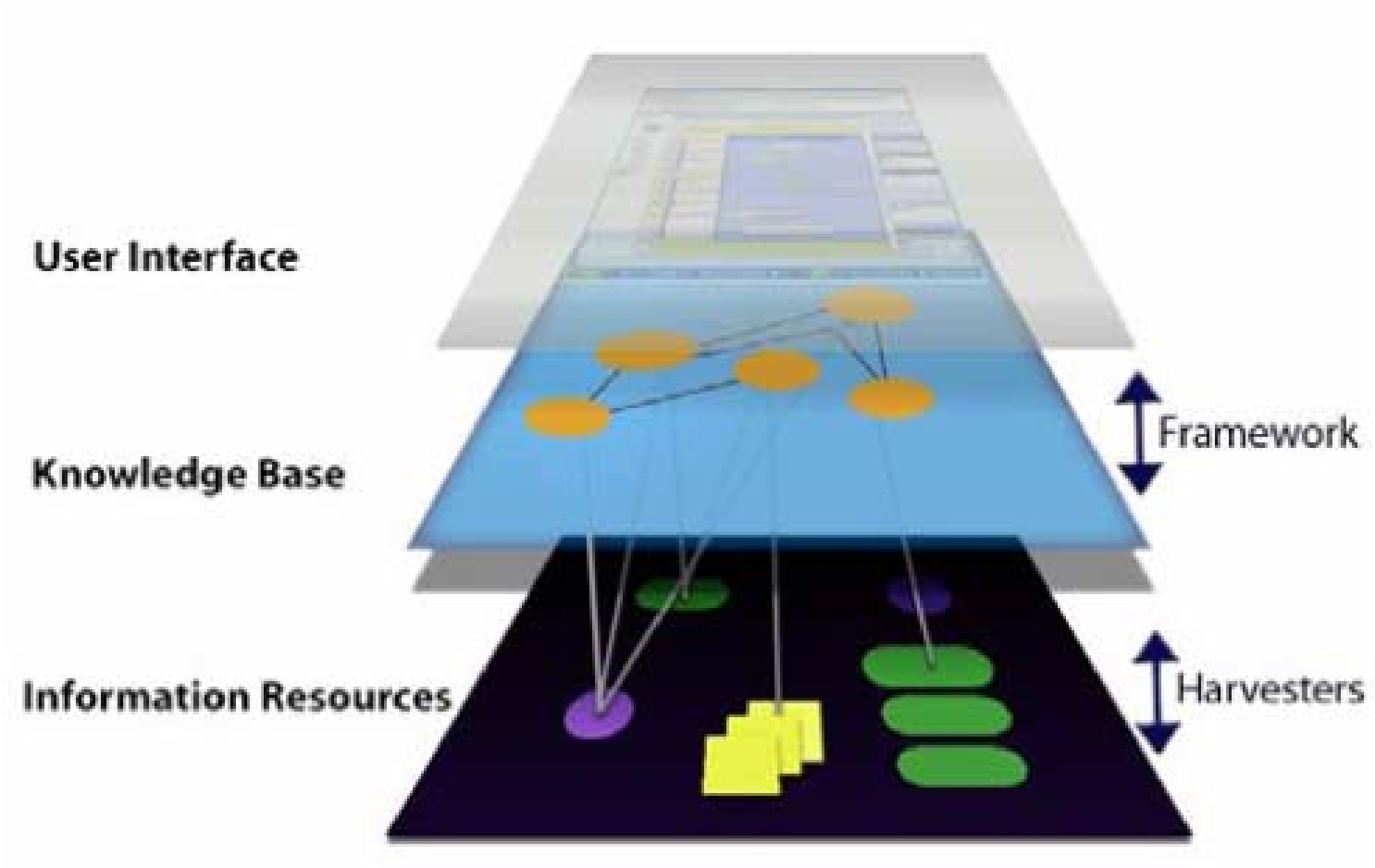
aKademy 2005: Kat and Tenor hype

aKademy 2006: Nepomuk and Strigi are presented

Now

Nepomuk	Strigi	Xesam
semantic storage and standards	data extraction, indexing, search	freedesktop.org search standard

The Semantic Desktop



Strigi libraries

libstreams

- efficient streaming access to file contents
- universal API to different formats

libstreamanalyzer

- analysis of libstreams streams with many parallel analyzers
- storage and retrieval over abstract interface

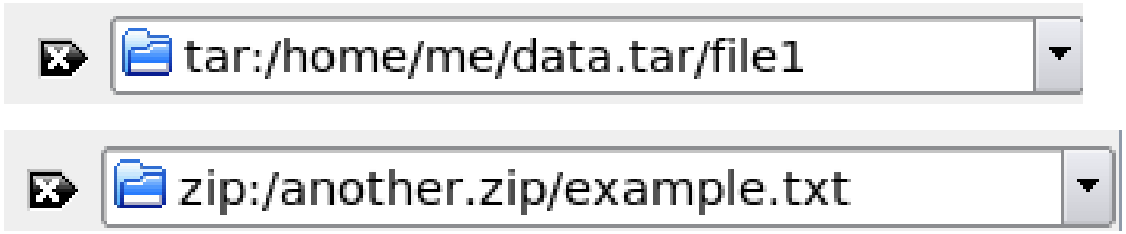
Reading nested files

*.gz	zcat
*.bz2	bzcat
*.tar	tar
*.zip, *.[jwe]ar, openoffice files	unzip
email	mail client
email attachment	mail client
*.pdf (?)	?
*.deb, *.ar, static libs	ar
*.cpio	cpio
*.rpm	rpm2cpio + cpio

many formats, many tools, many interfaces

Common API for nested files

Can we use kio or vfs?



zip:/
tar:/
gz:/
rpm:/
deb:/

commonapi:/

disadvantages:
- user has to figure out what kio or vfs is required

solution:
- make a clever kio/vfs that understands all

alternative: fuse

Files nested in nested files

tar:/home/me/data.tar/file1.zip#zip:example.txt

“None of the chained uri stuff (tar/zip/etc)

really work, and never did.”

Alexander Larsson,

Oct 2005 to gnome-vfs-list@gnome.org

“Bug 73821: Please "unchain" kioslaves.

Browsing a zip inside a zip should work.”

KDE bug since Jan 2004

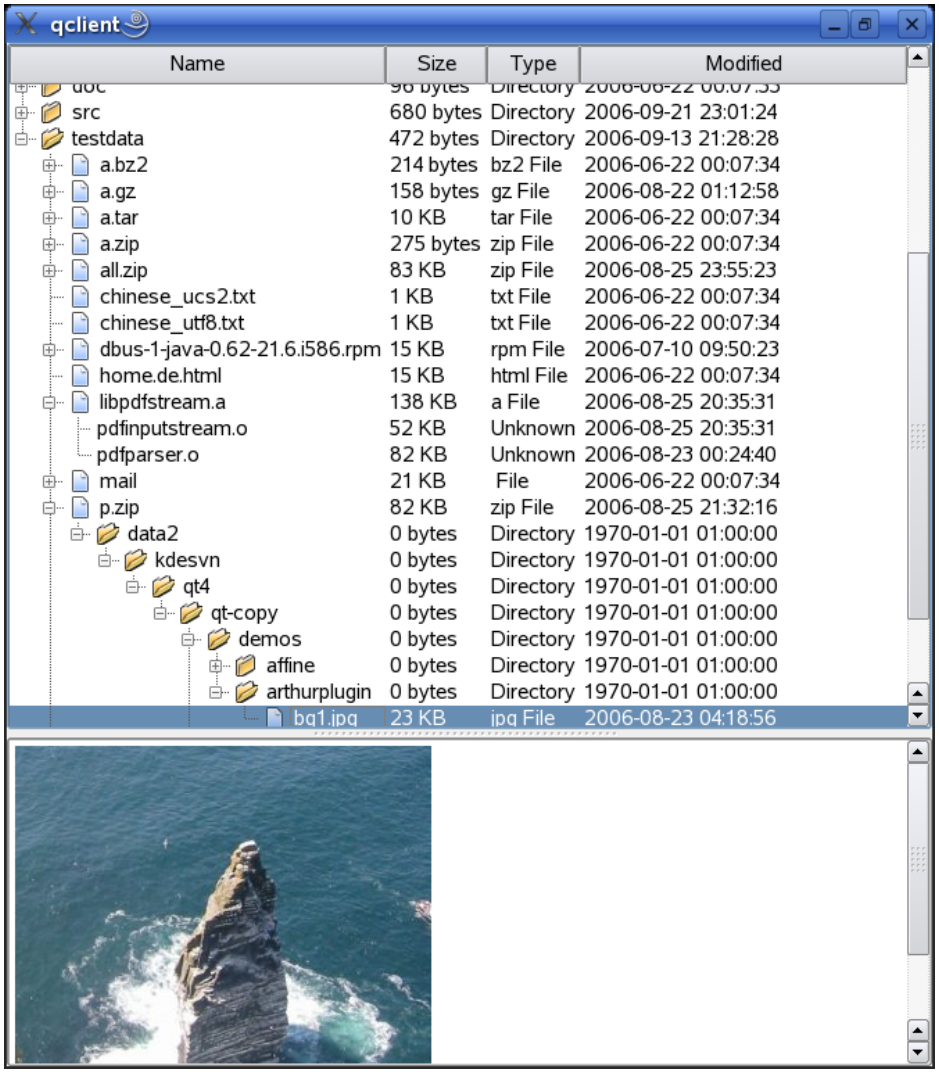
StreamBase and SubStreamProvider

```
class StreamBase {
    virtual int32_t read(const char** data, int32_t min, int32_t max) = 0;
    int64_t reset(int64_t newpos) = 0;
};
```

```
void
readdemo() {
    int32_t nread;
    const char* data;
    nread = stream->read(data,  1,  0); // read at least 1 byte
    stream->reset(0); // reset to start of stream
    nread = stream->read(data,  3,  3); // read exactly 3 bytes
}
```

```
class SubStreamProvider {
    virtual int32_t read(const char** data, int32_t min, int32_t max) = 0;
    virtual int64_t reset(int64_t newpos) = 0;
};
```

More powerful Qt

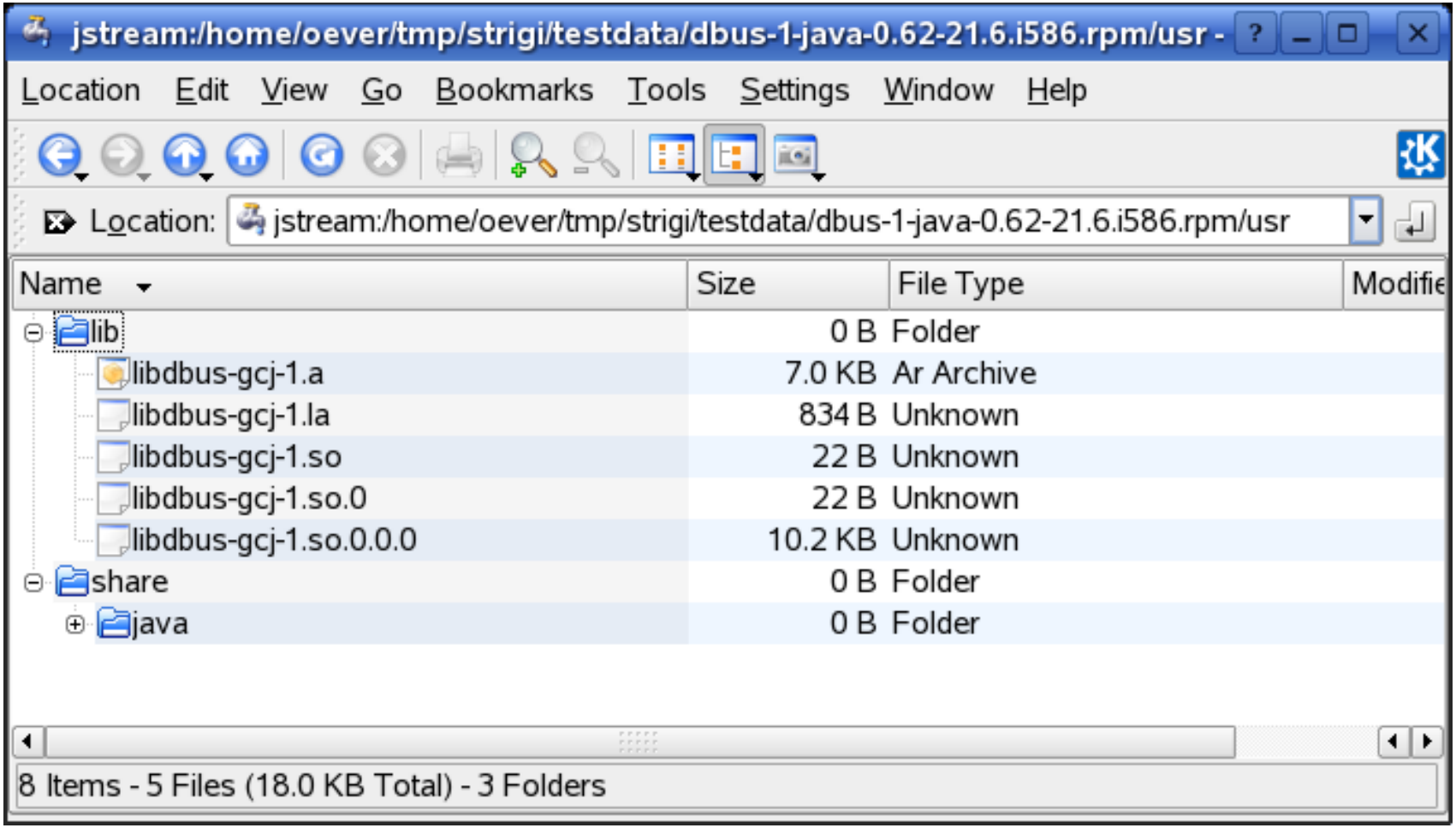


add read access to archive formats by adding only one line of code:

ArchiveEngineHandler engine;

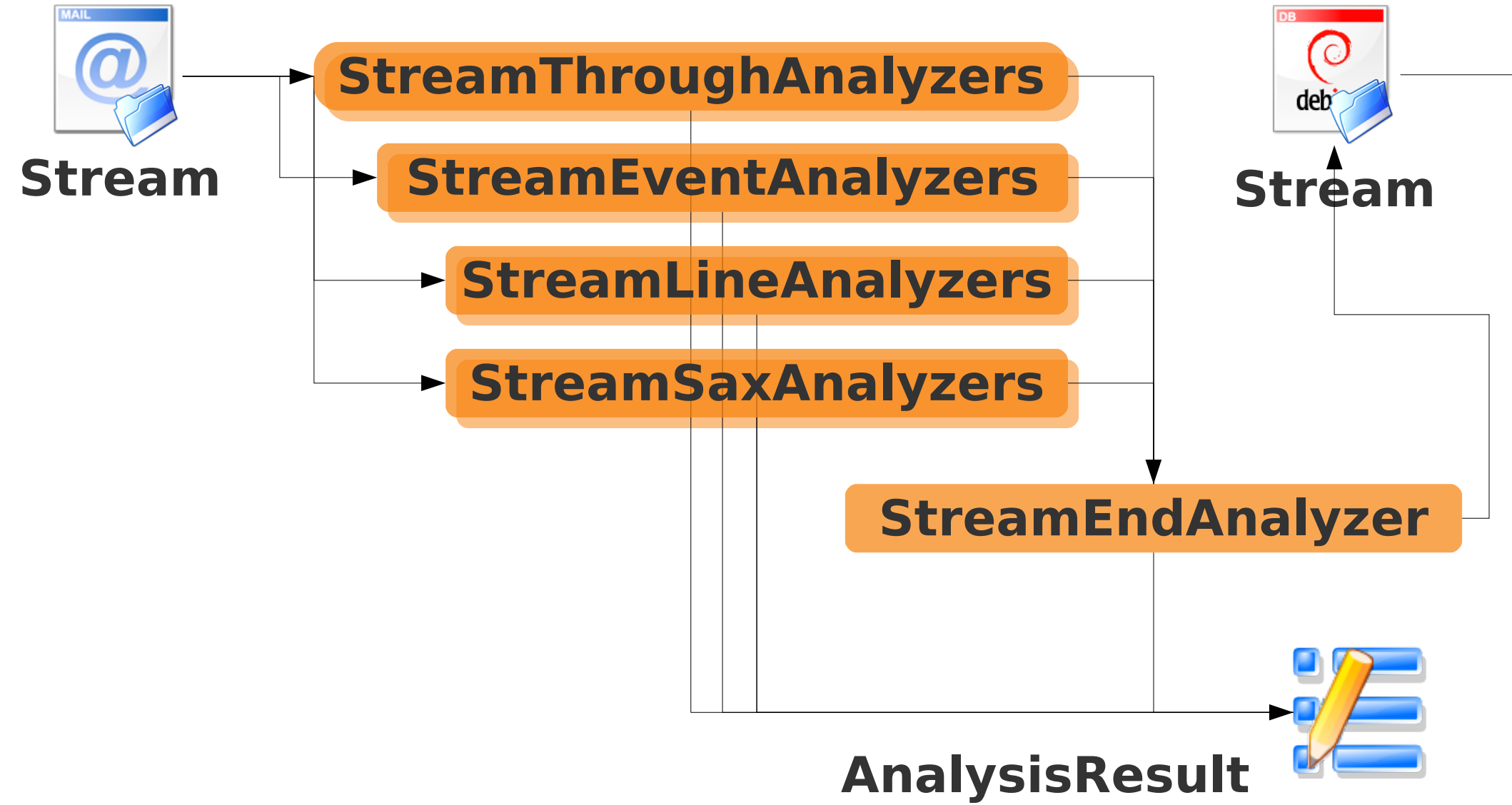
Class that comes with Strigi that uses QabstractFileEngine to give Qt applications transparent access to a custom filesystem.

More powerful kioslave



directory | file

Analyzing streams

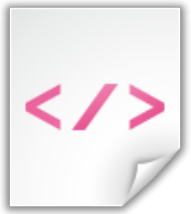


Simple RegEx Analyzer

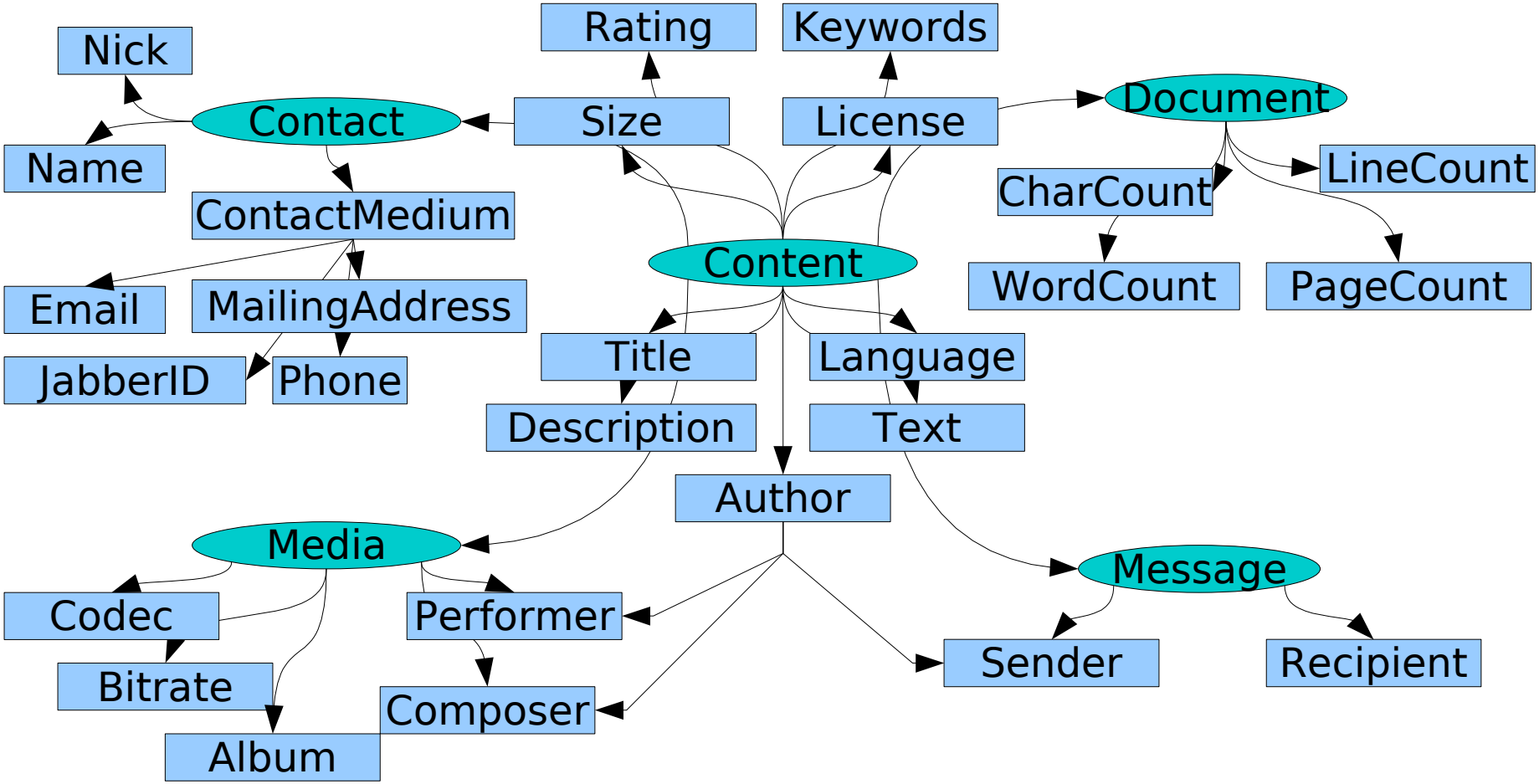
```
class RegExLineAnalyzerFactory : public LineAnalyzerFactory {
  StreamLineAnalyzer* newInstance() const;
};
```

```
class RegExLineAnalyzer : public StreamLineAnalzer {
public:
    void startAnalysis(Strigi::AnalysisResult*);
    void handleLine(const char* data, uint32_t length);
    void endAnalysis();
    bool isReadyWithStream();
};
```

Selection of file formats



Ontology overview



Evgeny Egorochkin

Indexes and Index Management

IndexManager

IndexReader IndexWriter

Indexes

Clucene

Soprano

SQLite

HyperEstraier

Xapian

semi-Indexes

KFileMetaInfo

CombinedIndexReader

GrepIndex

`xmlindexer`

`deepfind`

`deepgrep`

strigicmd and strigidaemon

strigicmd

create, query, inspect indexes from the command line

libstreams **libxml** **libbz2**

libdbus-1 **libclucene**

libz **libstreamanalyzer**

3 MB resident memory

strigidaemon

connection protocols

dbus **unix socket** **web service**

interfaces

Xesam Live Query **Strigi**

implementation

multithreaded queue

configuration **indices**

Speed Comparison

Indexing 10 000 text files (168 MB)		
Beagle	2h18	12m
Jindex	3h02	9m
Tracker	3h03	142m
Strigi	0h04	>4m

Source: Comparison of indexers
November, 2006
Michal Pryc, Xusheng Hui
Sun Microsystems

new KFileMetaInfo

API changed to fit to common ontology

mostly implementation changes

- KFilePlugin changed
  - Strigi<X>Analyzer for reading
  - KFileWritePlugin for writing
- libstreamanalyzer calls many analyzers on each file
- fieldnames changed: ontology is used

Social Semantic Desktop

http://nepomuk.semanticdesktop.org

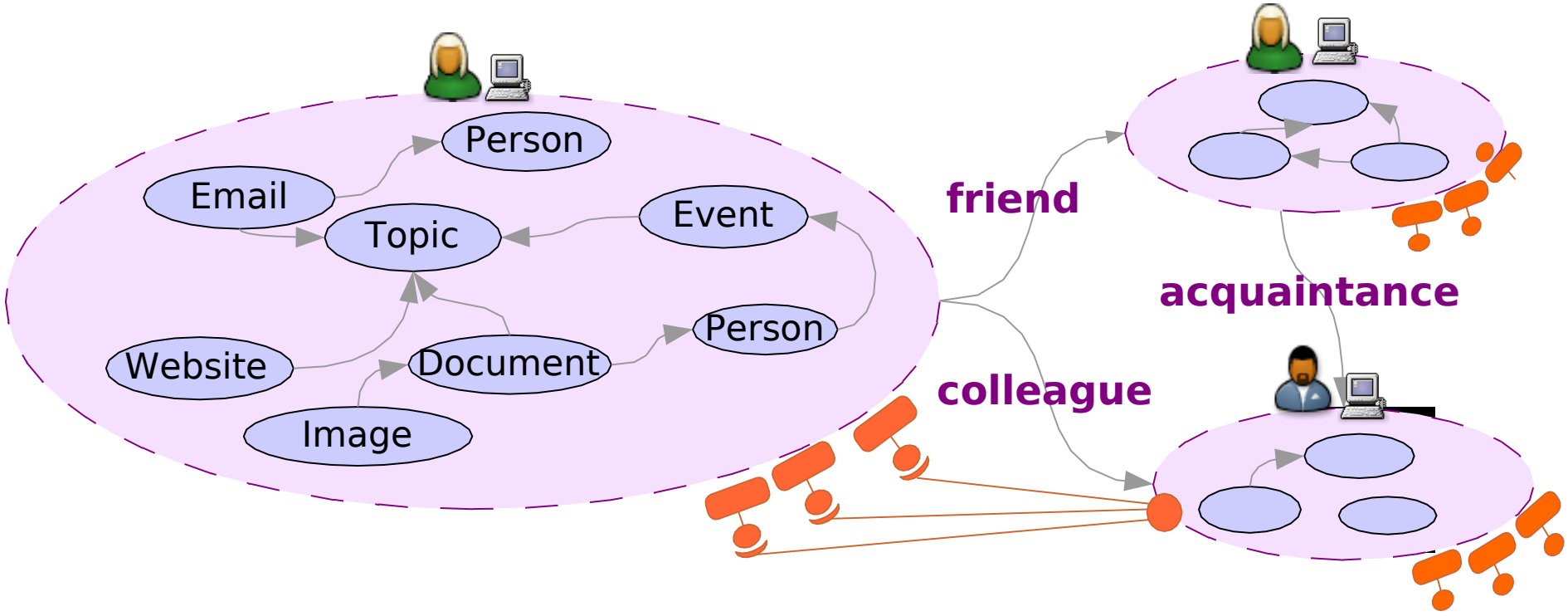
The Social Semantic Desktop

The desktop is a privileged adoption channel for the Semantic Web

Desktop: Help individuals in managing information on the Web/their PC

Semantic: Make content available to automated processing

Social: Enable exchange across individual boundaries



Personal Semantic Web — ***semantically enlarged intimate supplement to memory***

Social protocols and distributed search

Social semantic peers

Xesam: a common search API

eXtEnsible Search And Metadata specification

- DBus API for searching
- fieldnames for standardization

http://freedesktop.org/wiki/XesamAbout


Pinot


Nepomuk


Recoll


Strigi


Beagle


Tracker

+ Mikkel Kamstrup Erlandsen

Xesam: a common search API

DBus interfaces

- GetHits (in s search, in i num, out aav hits)
- GetHitData (in s search, in ai hit_ids, in as properties,
  out aav hit_data)

User Query Language

- type:music hendrix

XML Query Language

- <query><contains><field name=”dc:title”>
  <string>Gödel</string></contains></query>

Core Ontology

Strigi-chemical Analyzers

18 chemical formats:

(xyz, vmd, shelx, pdb, mol2, mdl, gaussian, cif, alchemy, cml, ...)

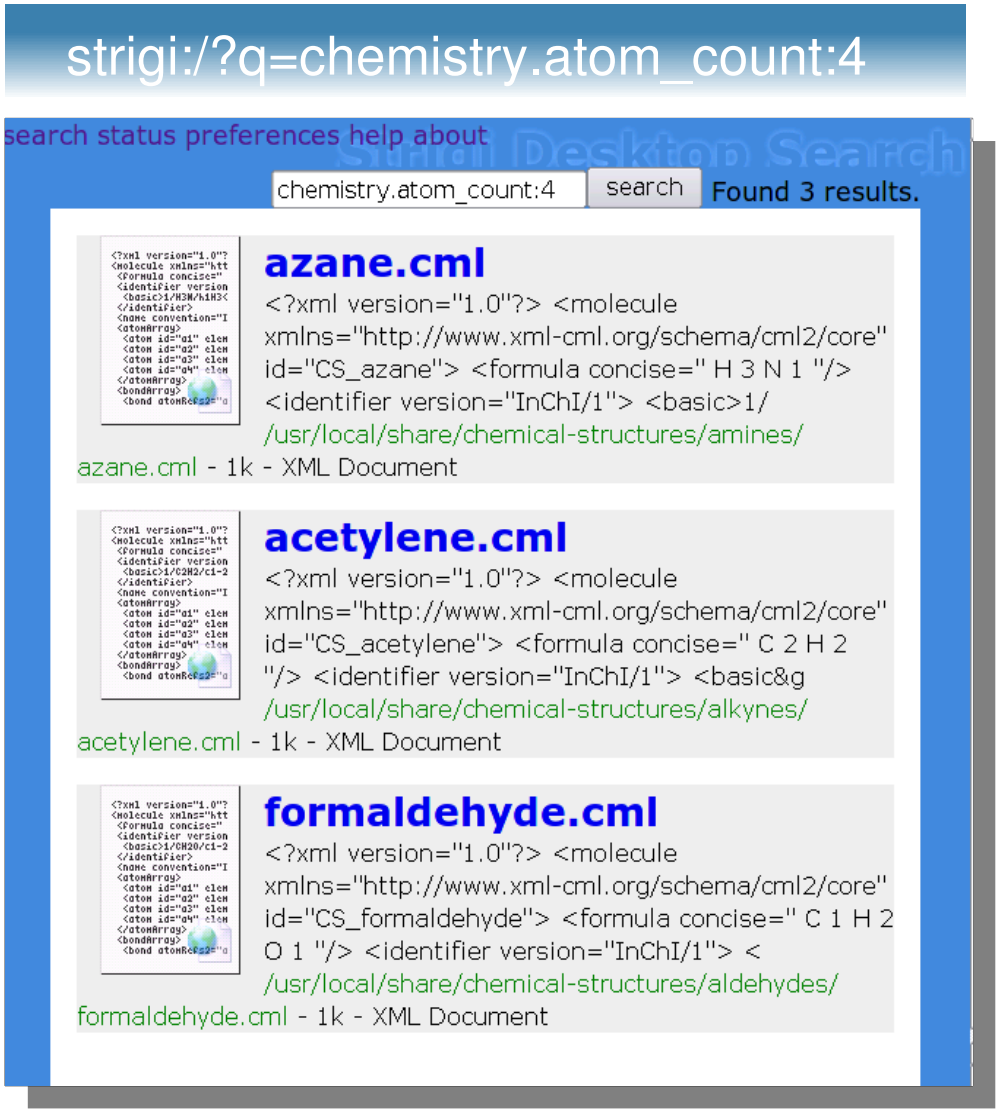
3 streamanalyzers:

(lineanalyzer, saxanalyzer, eventanalyzer)

19 fieldproperties:

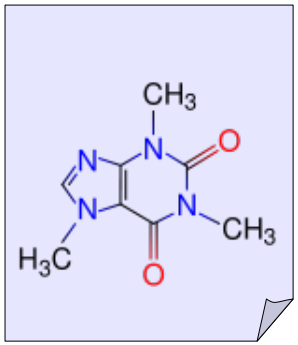
(chemistry.inchi, chemistry.molecular_weight, chemistry.molecular_formula, ...)

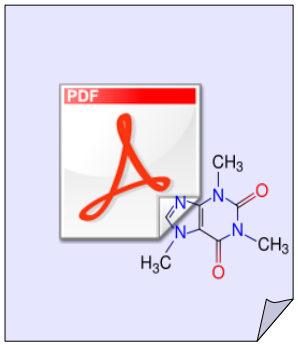
libOpenBabel to generate InChI

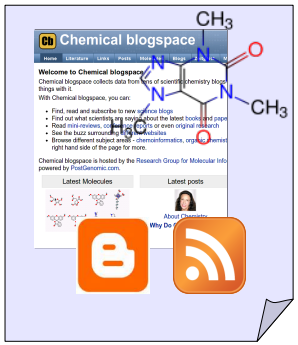


Alexandr Goncearenco, Egon Willighagen

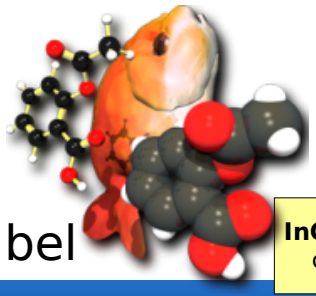
Strigi-chemical Workflow





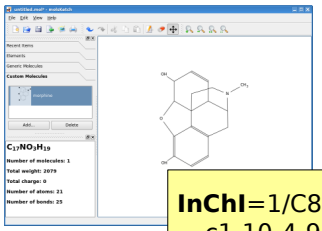


Chemical MIME



libOpenBabel

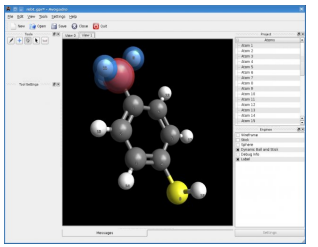
InChI=1/C8H10N4O2/c1-10-4-9-6-5(10)



molsKetch

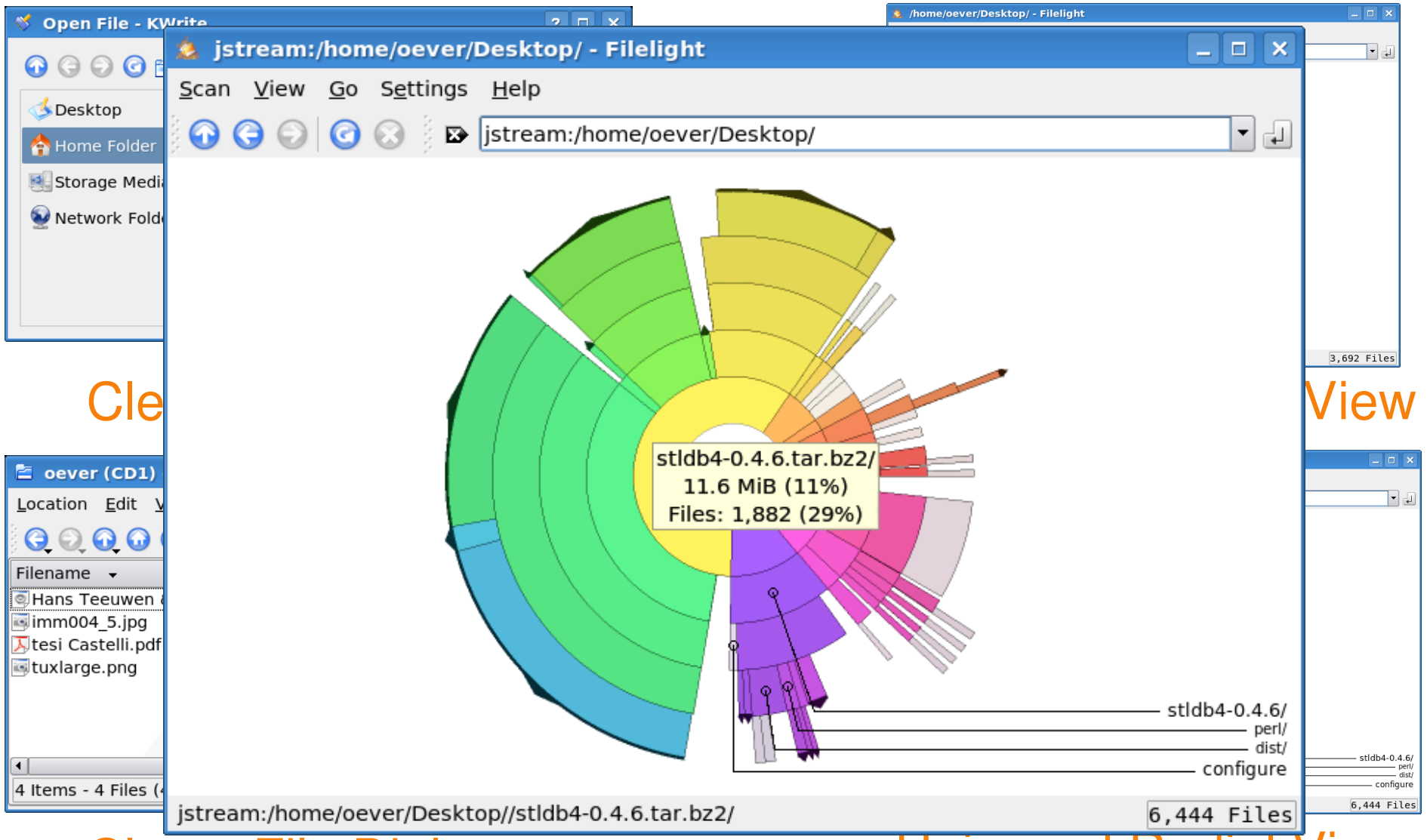
InChI=1/C8H10N4O2/c1-10-4-9-6-5(10)

List of search results



Kalzium/Avogadro

File Manager improvements



Clever File Dialog

Universal Radial View

Strigi for KDE4

fast stream libraries for reading and analyzing streams

use of modern technologies with a wide consensus

power of a indices to make your applications fast and clever

KDE 4

Nepomuk	Strigi	Xesam
semantic storage and standards	data extraction, indexing, search	freedesktop.org search standard

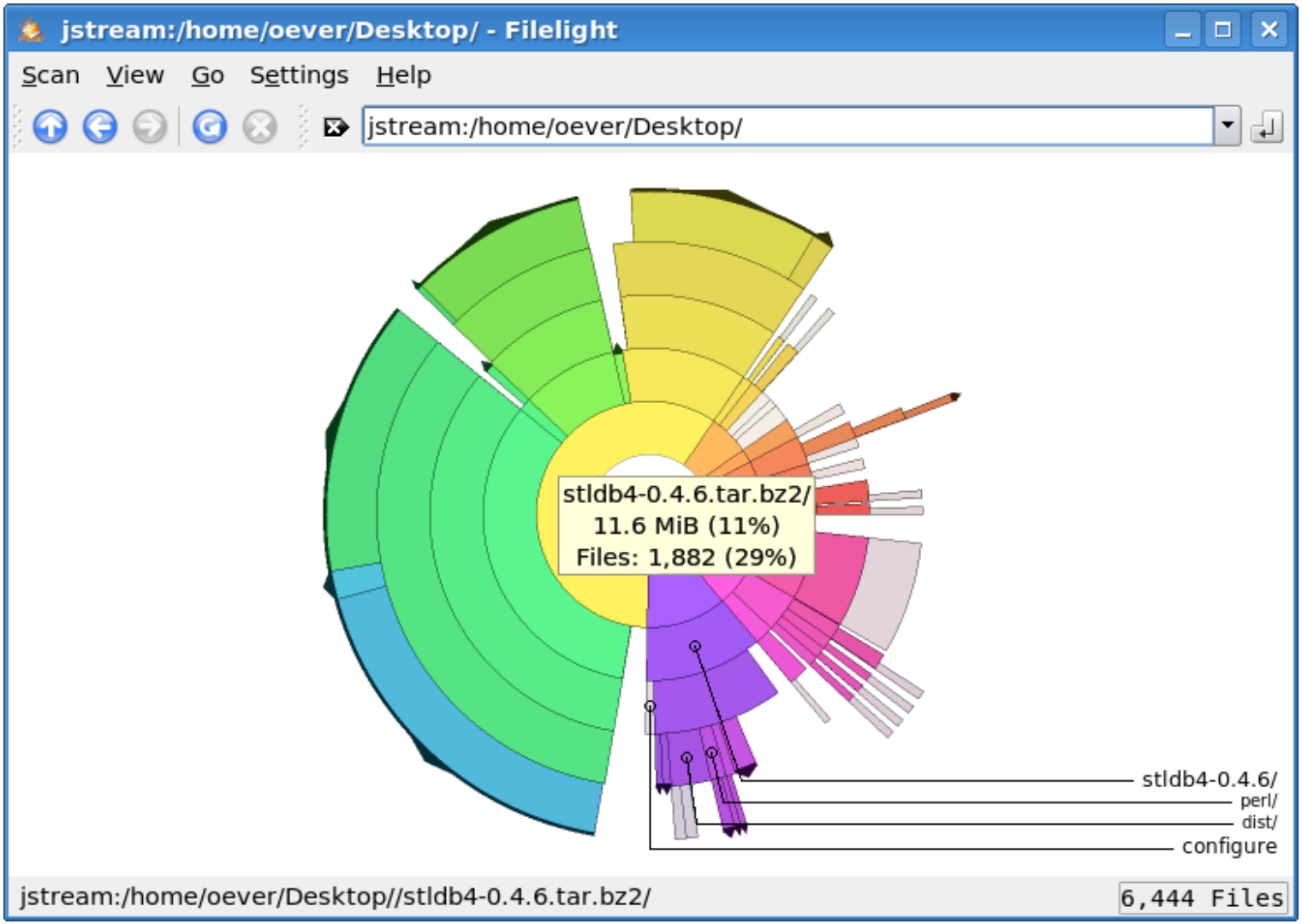
Google Desktop Search

\+ is widely deployed and tested on other platforms
\+ has a stable well documented API
\+ has a documented API for querying the search daemon
\- is closed source software
\- uses a proprietary index format
\- uses COM for communication
\- has a large brand recognition and there will a demand for it
\- calls analyzer plugins based on file extension
\- has a limited, unexpandable list of categories for files
\- identifies files by mtime + uri
\- uses wchar_t internally
\- is file based
\- has no command-line tools

Google Indexing plugins

Audio: 3
Chats: 4
Email: 4
Files: 36
Images: 2
Remote: 2
Source Included: dead link
Video: 3
Web History: 3
Other: 19

Browsing your files



Browsing your files