

A horizontal search bar with a rounded right end. On the right side of the bar is a small green magnifying glass icon.

An Open Decentralized Platform for  
Collaborative Search, Filtering and Content  
Curation

# True Problems with Search ?

- ▶ Privacy issue: engines collect  $\{IP, query\}$ 
  - ▶ not a problem: single hop on the network hides the IP (and location).
  - ▶ other signals remain that can be tweaked (HTTP headers, ...).
- ▶ Capture of all searches (aka “they know everything”).
  - ▶ love secret ? use custom crawler or Yacy.
  - ▶ love sharing & datalove ? share with everyone (including engines), use Seeks.
- ▶ **User isolation (search is a solitary place).**
- ▶ **Lack of user control** (aka “filter bubble”).
- ▶ **Proprietary solutions** (prevents innovation).

# User Control over Search

- ▶ no engine is an oracle (and can't answer all questions),
- ▶ personalization with no control (deactivate it and meet the 'oracle'),
- ▶ statistical Machine Learning algorithms make rational understanding difficult.

# The Sharing of Queries

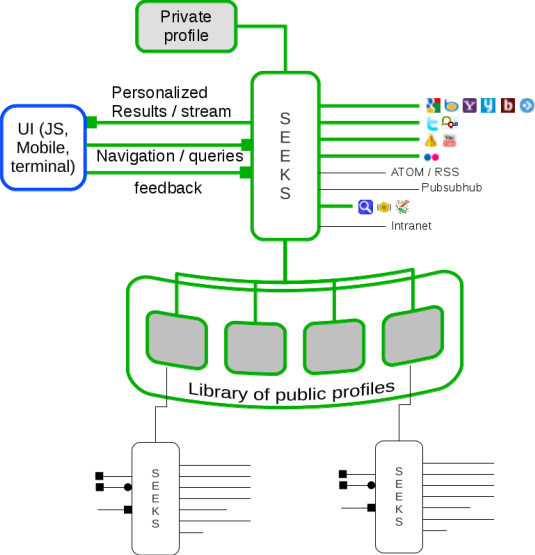
Fear to share ?

- ▶ the search engine *knows*: you can let others *know*,
- ▶ you're not alone: if you're looking for *it*, someone must have written on *it*,
- ▶ sharing with people with similar interests: fear that they know you know they know ?
- ▶ possibility to share while partially obfuscating the queries: not impossible to *know*, but more difficult.

# Seeks Overview

- ▶ Free Software (AGPLv3),
- ▶ Installs on your machine (laptop, server, sheevaplug, neufbox).
- ▶ Written mostly in C++.
- ▶ Acts as a proxy and / or Web server.

# Seeks Overview



# Seeks Overview

## Technical Information:

- ▶ originally based on Privoxy code,
- ▶ plugin based, basic platform is a proxy with some extra lib, everything else are plugins,
- ▶ Profile DB uses Tokyo Cabinet (low level, tiny & fast),
- ▶ Protocol Buffers serialization for minimal storage and network transmissions,
- ▶ libevent for web service,
- ▶ low memory trace (though configurable),
- ▶ DB size ( 1M records for 500MB) on large instances.
  - ▶ Google indexes 50 billion documents.
  - ▶ if we were to index those same URLs with title and summary,  
 $50 \times 1000 \times 500MB = 50.500GB = 25TB$ .
  - ▶ in practice, doesn't scale linearly.



# Seeks Overview

Why don't we crawl and index ?

- ▶ we can when needed (Nutch + Solr): do not reinvent the wheel,
- ▶ others do it faster, deeper, better: too costly to duplicate,
- ▶ others crawl and index (almost) useless pages (at a slow rate)
  - ▶ we store what users have found, somewhere, somehow.
- ▶ crawling / indexing is insane: shall we copy everything ? in 20 years, 50 years, 100 years ? (ecologically insane).

# Seeks Overview

What distinguishes Seeks from other metasearchers

- ▶ Every query is turned into a *halo* of similar queries and used to retrieve a larger set of results automatically,
- ▶ brings together a community of searchers whose positive (not negative) feedback is shared to improve search results within the community.

How sharing works:

- ▶ Positive feedback improves search for all,
- ▶ Negative feedback is local to every Seeks instance (acts as a “spam” filter).

# Seeks Overview

## Roadmap:

- ▶ 0.3.4 stable / 0.4 experimental: P2P ring of selected peers (list).
  - ▶ all queries go to all peers.
- ▶ 0.5: will run a DHT to build a per-query ring of peers.

# API

## Overview

- ▶ Input: HTTP GET/POST or proxy interception,
- ▶ Output: JSON or static HTML or dynamic HTML + JS.
- ▶ POST with body serialized as Protocol Buffers (with / without compression).

New API coming up (no details here).

# API

Get search results (text & images)

- ▶ /search and /search\_img resources,
- ▶ parameters: expansion (more results), page, prs (personalisation), engines (selection), lang, ...
- ▶ example: `http://www.seeks-project.info/search.php/search?q=rml1&lang=fr&prs=on&output=json`

# API

## Advanced functionalities

- ▶ Clustering

- ▶ by types (wiki, documents, files, pdf, forums, ...)

- `http://www.seeks-project.info/search.php/search?q=rml1&lang=fr&action=types`

- ▶ automatic `http://www.seeks-project.info/search.php/search?q=rml1&lang=fr&action=clusterize&clusters=5`

- ▶ `/search` returns most discriminant words per result snippet,

- ▶ similarity analysis ranks all results in decreasing similarity order from the selected result

- `http://www.seeks-project.info/search.php/search?q=rml1&action=similarity&id=13413418`

# API

## User Control

- ▶ Recommend result: impacts the ranking of the result for the current and similar queries `http://www.seeks-project.info/search.php/qc_redir?q=rml1&id=1313434`
- ▶ Reject result: impacts the ranking of the result for the current query `http://www.seeks-project.info/search.php/tbd?q=rml1&id=13498734`

# API

P2P search results (aka “recommendations”) & query suggestions

- ▶ get Seeks only results (no calls to other engines), from the P2P ring.
  - ▶ use the engine=seeks parameter,
  - ▶ new API will bring more control and resources.
- ▶ get similar queries, shared by other users.

Coming up: push new results yourself!

- ▶ associate a URL with a query, then push it to the P2P ring,
- ▶ it will be recommend to users issuing similar queries to the one it was pushed to.



## Check it out

`http://www.seeks-project.info/`

- ▶ Keep it in mind: it is always best to install Seeks on your machine(s) (let your machine work for you).
- ▶ Though there are public nodes:
  - ▶ `http://www.seeks.fr`
  - ▶ `http://www.seeks-project.info/search.php`
- ▶ Configurable Android app (F-Droid, Market, `https://dl.sileht.net/seeks/`), Mehdi Abaakouk.
- ▶ Contact:
  - ▶ `contact@seeks-project.info`
  - ▶ IRC `#seeks` and `#seeks-fr` on Freenode.