An online search tool for the UFO research community

1 Target

Offer to the UFO research community a mean to 'full text search' in a large amount of documents. Essentially books, magazines and any other types of documents

Yet reduce copyright infringement risks to acceptable levels. A tradeoff has to be found.

This document explains how it is possible to have the files remain in the servers of their respective owners, with no public access, yet allow a public full text search into them.

This document is a high/medium level specification.

This is not in order to produce a fancy website with adverts and facebooks links. The interface will be pretty dry.

This document is tentative, explains the mandatory features as well as some nice to have (nth) things.
nth features should not be developed first, but the architecture should ensure we can do it easily later.

Nothing in this document has been approved by any party.
It is only a proposal. All is here to be discussed. It is not yet fully consistent.

Here, could would will should shall is ill defined.

2 The current situation

A fair amount of relevant documents exist in digital format.
They are spread among several collections worldwide.

- Many are in restricted access. Typically just accessible to the members of an association. Typically, limited access is secretly granted to motivated researchers, or just a few files at a time are shared.
- Some are publicly available on websites and freely downloadable.

When a researcher is looking for something, the best he can do is to post his question to a forum, then get the advice from someone that knows better for where to look for.

This, while search engines have been around for more than 20 years.

3 Motivation

For the existence of the tool itself, the document is also here in order to express a clear intent.

Anybody can make any constructive remark.

Review, analysis, remarks, critics, ideas, advices, encouragement, help, approval, disapproval, know better. Comes into conflict with any other project ? Could instead merge into an already existing project ?

I am convinced we will find something. I already want to make many targeted searches.

4 Input data

I would like this : granted access to crawl by SFTP, pdf sets from as many associations and private collections as possible.
All should be OCRed.
All should be in pdf format.
In many languages

Expected amount of pages is in the several million range

Part of them are in the public domain.
Part of them are still under copyright, so that they cannot be shared freely.

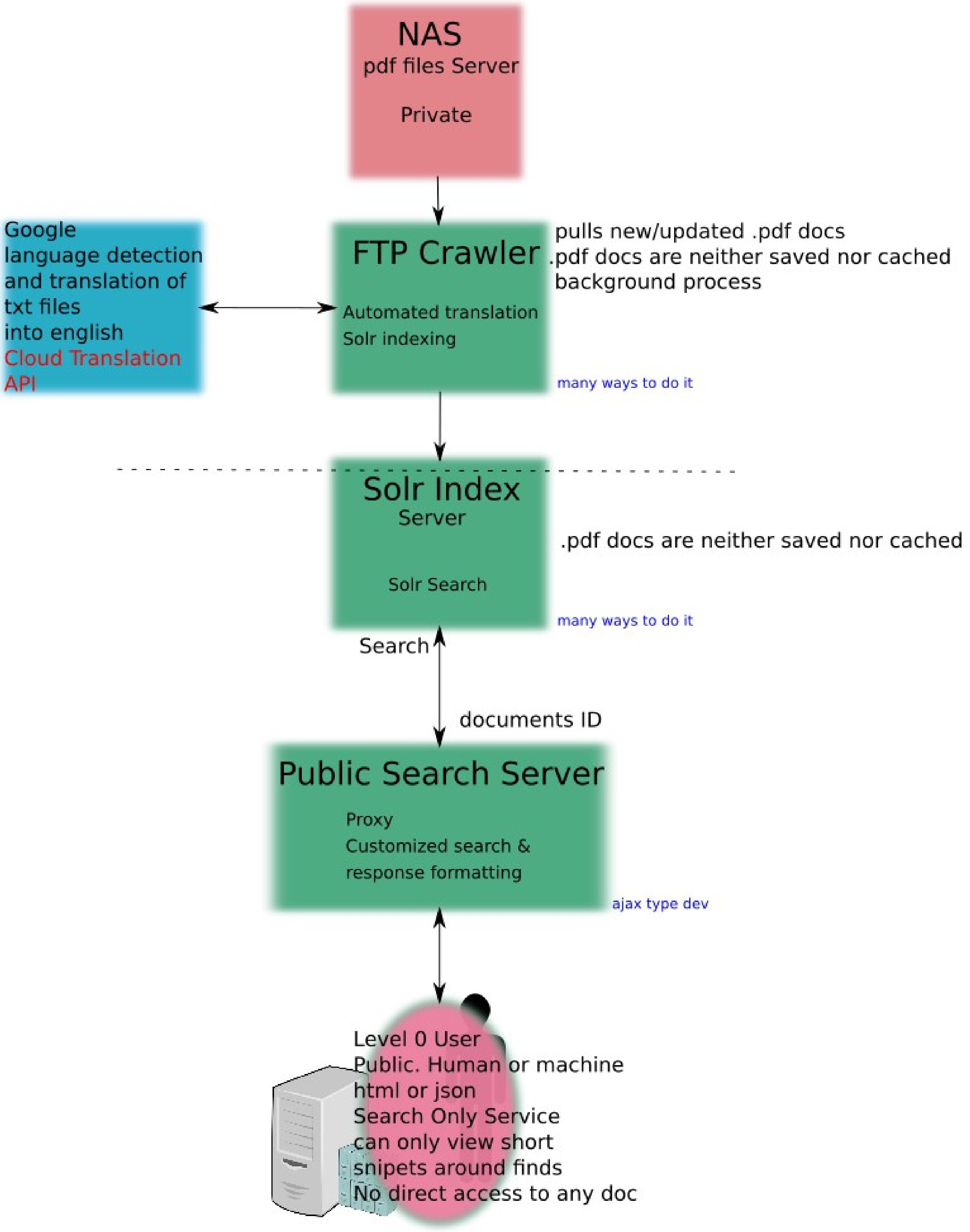
The system would behave like a web crawler

4.1 Data ownership

Of course, it would be best and easier to manage if all the pdf files could exist centralized in at least one server.

However, it remains possible to architecture the system so that the pdf files remain only in the servers of the source associations. More on this below.

5 Architecture



5.1 FTP Crawler

5.1.1 Principles

Using SFTP (final choice remains TBD)

This implies that the Collections owners grant a ftp account to the crawler.

The code of the ftp crawler should be open source (github) (as long as it does not put data protection at risk) so that the Collection owners know what is happening with their data.

The crawler builds the ID : CollectionID/full_directory _path/filename.pdf

The crawler identifies the language of each doc using Google Cloud Translate API
https://cloud.google.com/translate/docs/detecting-language

The crawler also saves a hash of the full .pdf file and a hash of the text contents.

As much as possible, for performance reasons, the crawler should not use/rely on Solr for the crawl process. Only for pushing files into the index.

The crawler detects efficiently potential file changes by the combination of : size, date, hashes

The crawler should smoothly manage moved files (same hash but location → delete doc by old ID, add doc with new ID).

5.1.2 Hardware

Many web servers can do the thing.
It could also be done on a Windows PC, using python. The crawl process being started on demand or on a schedule.

5.2 User Interface

Essentially: a web page on a public server offering human (and nth machine readable) search results
Any web server can do that

5.3 Search engine/server

5.3.1 Principles

For its power, efficiency, speed and simplicity, Google Custom Search under the hood was a target. However, the impossibility to prevent Google from caching the pages forbids its use for the private part of the files.

Alternative search methods where studied :

- DSM on Synology servers offers a 'universal search' service; however, it cannot work if download is disabled + It seems to be very inefficient at indexing big amounts of files.
- One can index pdf files using Acrobat DC Pro. It works rather fine, even for big amounts of files. Very practical. However, this works practically only if you have the docs on your PC. We don't want that.
- Solr / apache is very powerful, open source but really complex to setup. Still, seems to be the best remaining choice.

Solr is right now the only remaining solution I have found.

The public search server could run on any machine. It has to manage much less disk space (down to 1/100 approx) than the documents. If we intend to develop our own search engine or use Solr, then it must be quite powerful, probably with 6/8GB ram.

The public search server does not contain at all neither the txt files nor the pdf files.
It contains only the freely accessible search-able index.

The pdf search server is fed by a ftp crawler with new pdf files for indexation.

When one feed a new pdf to Solr, one can provide additional *fields* that will be saved. As a bare minimum, a unique ID is one of these fields.

My proposal is to use as an ID the full path to the file in the form

CollectionID/full_directory_path/filename.pdf

Where CollectionID uniquely identifies the server where the original pdf is located. It is a simple name.

The ID creation will be automated and done by the crawler.

When this document will be found as a search result, the ID will be also retrieved.

The ID should be enough (with optionally some additional secret information held by the file server owner) to find the exact location of the file.

That does not mean that these files are freely accessible !

In the case of private docs, the search tool returns the snippets and just the ID. With that, a researcher will be able to locate the document. Contact information might be part of the result displayed.

The search tool should give the option to limit the search to the public part.
In the case of the public docs, the search server can directly provide the URL of the file.

The indexation is fully automatic. The search server crawls often the collections, searching for updated or new docs.

The crawler is autonomous, that is a huge advantage over any manual method.

5.3.2 Hardware

Solr needs a java jre which means you need quite a lot of processing power.
+ that will require some work and maintenance.
If on a Synology NAS, requires DS218+(for java) and even probably DS718+ for the Virtual Machine service that make it possible to run Docker-Solr. But here we don't have a need for a huge storage. The specificity of our need is more in the processing power and RAM size.

6 Nice to have

6.1.1 Admin command line interface

Direct indexation
For the case of a very big bunch of files that can't be pulled from the web, a tool should be able to index files directly from a HD connected to the Solr Index machine. Then the crawler must be updated with the hashes/dates/sizes/IDs.
The tool builds the ID based on the file tree, file name. The user should provide the CollectionID as a parameter

6.1.2 The problem of the languages

I would like to be able to search all docs, including non English documents in one single search in English.

Here is the trick.

Two additional fields will be added.
One contains the name of the original language of the doc. (ISO)
One contains the name of the language of the doc. (ISO)
All non English docs will be indexed twice: in their original language, and in English.
All English docs will be indexed once; in English.

Because the ID must be unique, the translated version should use a slightly (TBD) modified ID that makes it easy to reconstruct the ID of the original file. Since there could be one ID for each language, there is no other logic choice than to include the language name into the ID.
It could be something like :
CollectionID/full_directory_path/En-en/filename.pdf

Automated translation would use the Google translation api (fully automated, not free)
In view of how poor some language translations are, some translations won't be done.

Because the translation service is not free, the translated .txt file shall be cached somewhere (private/hidden).

It will be found again later because the .txt tree mirrors the .pdf tree. The ID of the original file can be used to reconstruct the path of the translated file into the hidden cache.

Translation should be done again if the hash of the file AND the hash of the text contents did change.

We could also translate all documents in all languages and offer search in all the languages. (could be one Solr core per language) But that would be quite heavy.

6.1.3 Crawler reports

The FTP crawler may send a report by email after each crawl to the file server admin.
May report the list of pdfs that are not OCRed.

7 Data protection

7.1 Against Thieves

Looking carefully at the index files generated by Solr, they are pretty unreadable. Yet one could reconstruct the documents from them by a very repetitive automated process.

As an additional protection the search server should forbid repeated 'scanning' queries and many other types of tricks.

Having the public search server physically separated from the indexing machine adds some protection (proxy) as shown in the architecture drawing.

By all means it should be impossible to directly pull search results directly from the Solr index. Only the search server should be allowed to do it. One way to do it : have the index reply only to the IP address of the public server. This should avoid data leaks because the protocols above IP won't establish a link in case of a thieve using the same IP address as the public server. + we can add an authentication layer.

Services/Advices from a pro is wanted.

7.2 Index corruption recovery

Absolutely required because full re-indexation is expected to be very very long.
TBD

8 Steps

8.1 Step 0: Feedback from the community

For validation of the intent.

8.2 Step 1: demo

Provided that step 0 validates the project, a demo will be made in order to motivate collection owners.
This using public documents.
Then we will see if we go further.

9 Annex

9.1 Google Custom Search

Why not using it ?

With Google, all the txt documents get cached by Google.
Not acceptable for the private part of the index, but perfectly ok for the public part.

Custom Search Engine is now integrating with Google For Nonprofits,
https://www.google.com/nonprofits/

Now, since Solr can also do it, there is no real need for Google custom search.

laurent chabin (2018/07/11) v2.01

*** end of document ***