Staff/PE machine coding round

Design and build an application that can search documents from a cloud storage service like Dropbox or Google Drive on the content inside the document.

Programming languages to be used: Java /C++

Please keep these considerations in mind as you go about solving the problem:

1. Implement the solution as described below. Do your best to reflect your design and coding ability.
2. Implement clean modular code respecting OO paradigms.
3. The application will be demoed during the interview.
4. Write a simple 1 pager PRD / RFC for the above problem. You can use any format you are familiar with.
5. You can refer to code samples on the web, but you should be able to explain your solution.

In this coding assessment you will develop a basic search service for data stored in online storage services like Google Drive, or DropBox. You can upload the code to github(preferred) or upload the project as a zip/tarball and share by email/file sharing service.

Requirements:

1. Connect to an online data storage service of your choice (Google Drive, Dropbox, S3, or any service of your choice) and use the provided APIs to fetch the files stored by the service, e.g.
    a. https://dropbox.github.io/dropbox-api-v2-explorer/.
    b. https://developers.google.com/drive/api/guides/search-files

2. You can have files that are in either .csv, .txt, .pdf or .docx format. (Bonus : For the files which are not utf-8 encoded, you may choose to extract the text content from them using a library of your choice like Apache Tika(available for Java/python): https://tika.apache.org/. Or tesseract (https://github.com/tesseract-ocr/tesseract)

3. Index the content within the files to provide maximum-text search capabilities on either the data of the file or the meta information of the file. You can use Elasticsearch / Postgres or any storage you think is suitable for this:
    a. https://github.com/elastic/elasticsearch#readme.
    b. https://github.com/postgres/postgres.

4. You will provide an API that takes a search term/token as input and returns a list of files and their HTTP URLs that contain the term in their content, e.g.

```
curl https://<search-service-host>/search?q=”@mail.com”
```

5. You will provide a basic command line or web interface that consumes the above API and displays the files matching the given query.

```
E.g
files in storage with content below
FilePath          Content
X../File1         a,b,c,d,e
Y../File2         c,d,e
X../File3         g,h

input 1 -> curl https://<search-service-host>/search?q=”c”
output -> X../File1
          Y../File2
```

```
input2 -> curl https://<search-service-host>/search?q=”notfound-term”
output -> Empty
```

Delete file1 at source(cloud storage)

```
input 3 -> curl https://<search-service-host>/search?q=”c”
output -> Y../File2
```