# Introduction

## Problem/goals

* Data sovereignty & access
* Storage & backup
* Tagging system?
* Multilingual search
* Offline use

## Stakeholders

* Researchers – illustrator, animator, linguists, …?
* Josh lives in the middle of nowhere and goes to the city every now and then to upload everything at once
* Consider file size & internet access – proposed solution will vary between users

# Requirements

## Functional

* Back up all files every so often to archive
* Cleanup when files are unlikely to be accessed for a while?
  + ‘Recycle bin’ type situation maybe
* Retrieve files from archive
* Depends on access rules
* Multilingual search
  + Display name even if not accessible?

## Non-functional

* Easy to use with minimal tech knowledge
* Offline solution for users in remote areas

# Proposed solution

## Storage

### Options

* AWS – shitty documentation, slightly cheaper
* GCloud – easier to use, Harry’s used it before maybe?

### Concerns

* Kathrin mentioned issues with ethics etc
* (main options being Amazon and Google)
* Access to data not guaranteed (funding and whatnot)
* Unsure if any better options exist
* Better than bitrot I guess

### Conclusions

* Probably going to go with GCS but hoping to run into better options

### Backup

* GCS can do versioning – keeps X previous versions of files that have been overwritten
* Every month or so upload snapshot of files to archive?
* Can set as cron job for those with decent internet access
* Option to run manually

### Retrieval

* Probably fine not to check local files as they’re already there
* Check dropbox and archive both
* Multilingual search (for later)
* Allow for download if access permits

## Design

### Upload

* Can be just a script that gets run every now and then
* Requirements:
  + Root folder to upload to archive
  + Login details? (for both archive and Dropbox maybe?)

### Download

* May need GUI
* Login page / keep details in env?
* Figure out instructions to get accounts set up from non-owner

|  |  |
| --- | --- |
|  | * Search multilingual by default   (can add settings to change?)   * Download only available if access permits   Things to decide   * Folder or file results * Display unaccessible? |

* Run from a script?
* Figure out how to do that

## Implementation

### GUI framework

<https://www.bairesdev.com/blog/best-python-gui-libraries/>

<https://www.dice.com/career-advice/7-top-python-gui-frameworks>

* tkinter: the Classic™, lightweight, but looks dated
* PyQT5: supports sql, web toolkits, xml processing; lacks Python-specific class documentation
* PyGUI: similar to tkinter
* Kivy: fast but bulky, poorly documented
* **PySimpleGUI**: only supports Python 3 but well-documented
* Libavg: great for mobile dev
* PyForms: desktop, web, and terminal applications
* wxPython
* PyQt
* PyGObject (PyGi)

Update: using React to make an Electron app

## Logic

### Args/env

* Login credentials
* Root folder location (to start uploading from)
* Ignore list?
* Cleanup folder location
* Date of last upload?

### Upload

* [start process]
* crontab -e (edit file) crontab -l (to check)
* sudo service cron start/stop
* alternatively, run script when needed
* [refresh credentials]
* (read env file?)
* (set up account login stuff?)
* Upload files to archive
* (opt. upload to dropbox?)

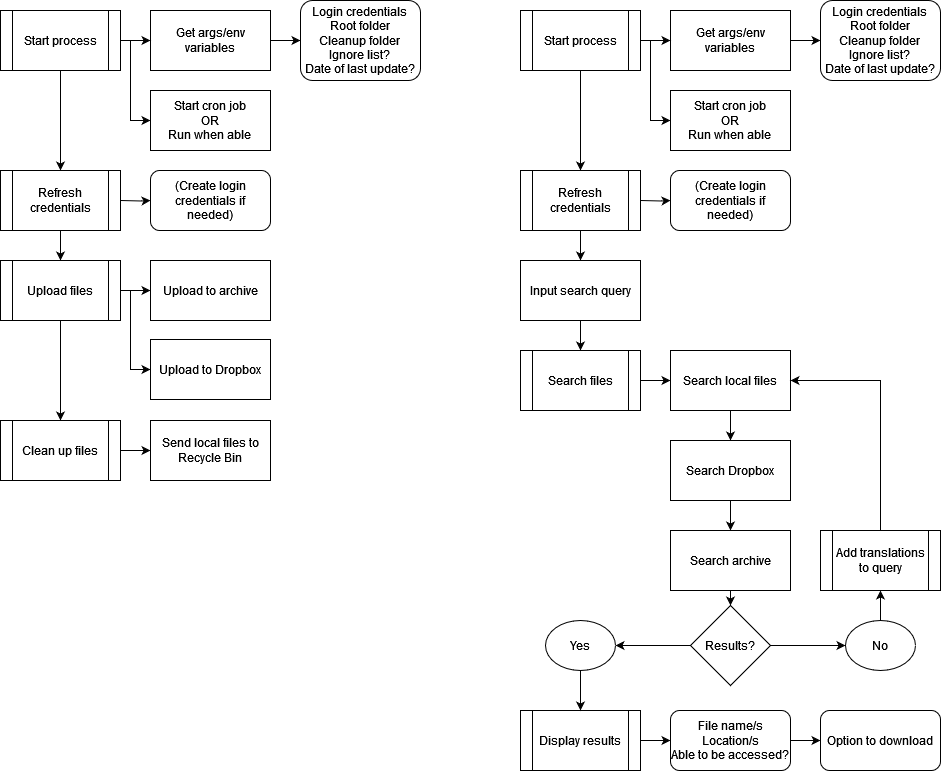
### Download

* Enter search term/s
* (search local files?)
* Search Dropbox files
* Use standard search (black box, up to implementation)
* Get all filenames, iterate through (no idea of efficiency)
* Search archive
* Get names, filter as necessary
* Multi-language search
* Search original string
* Dictionary of lists?

<https://www.emerald.com/insight/content/doi/10.1108/14684520710780458/full/html>

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5430129>

<http://epa.niif.hu/02400/02461/00064/pdf/EPA02461_acta_polytechnica_hungarica_2016_02_195-207.pdf>



# Considerations

## Storage platform

## Existing workflow

## Internet connection

## Tech experience & running

# Implementation

## Dropbox

<https://www.geeksforgeeks.org/how-to-automate-the-storage-using-dropbox-api-in-python/>

## Google Cloud IAM

* Sign into Google Cloud account
* Install CLI
* Run `gcloud init`
* Create or set a project:
* Create: `gcloud projects create PROJECT\_ID`
* Select: `gcloud config set project PROJECT\_ID`
* Enable resource manager API:

`gcloud services enable cloudresourcemanager.googleapis.com`

* Create local auth credentials: `gcloud auth application-default login`
* Grant roles:

gcloud projects add-iam-policy-binding PROJECT\_ID

--member=”user:EMAIL\_ADDRESS”

--role=ROLE

### Other stuff

#### Install client library (Python)

pip install --upgrade google-api-python-client google-auth google-auth-httplib2

#### Modify roles

<https://cloud.google.com/iam/docs/write-policy-client-libraries>

#### Google Cloud login

`gcloud auth login`

#### Enable IAP (Identity-Aware Proxy)

gcloud iap web enable --resource-type=app-engine --versions=*version*

## Traversal & upload

### Requirements

* Archive folder location
* Authentication completed

### Steps

* os.walk() – get list of all files under archive root
* for all files, upload to archive
* file versioning enabled, so it should keep a certain number before old versions are lost

## Download

### Requirements

* Authentication completed
* GUI

### Steps

* user enters search query
* (maybe swap with next step: determine other language versions of query)
* search local files, Dropbox, and archive
* display results
* if download button pressed, download file/folder (????)

## Cleanup

### Requirements

* Cleanup folder location

### Steps

* for all files in folder, send to recycle bin?

## Search

### Requirements

* Index of translations?

### Steps

* for all search terms, [???]
* word vectors to find similar topics?
* using metadata files to search by more than just name?
* include – add translations to optional
* exclude – leave as-is? extend list?
* Scaling??

## GUI

* Electron
* <https://wiki.ubuntu.com/WSL#Running_Graphical_Applications>
* React?

### Troubleshooting

#### Getting it to run in WSL

* Download VcXsrv
* Run xlaunch
* 
* .env 🡪 WATCHPACK\_POLLING=true
* "dev": "concurrently -k \"BROWSER=none npm start\" \"npm run electron\"",
* "electron": "wait-on tcp:3000 && electron ."
* (make sure concurrently, wait-on, and electron are installed)

### Running Python

#### Flask

#### PyScript

#### let python = spawn(‘python’,[…])

* "build": {

"extraResources": "python\_scripts",

...

* let python = spawn('python', [path.join(app.getAppPath(), '..', 'python\_scripts/my\_script.py'])

## Testing

## Monitoring

## Timelines

# Further investigations

## Queensland State Archives

### Leads

* Mukurtu CMS
* Digital Preservation Coalition
* Australian Society of Archivists
* Australasia Preserves
* <https://australasiapreserves.blogspot.com/p/digital-preservation-essentials.html>
* State Library of Queensland
* Jake Carroll & Fryer Library, UQ
* Email library with questions / make an appointment

### Mukurtu

* Managed by the Center for Digital Scholarship and Curation at Washington State University
* Showcase > Sustainable Heritage Network (has many resources on digital stewardship)

### CDSC/Sustainable Heritage Network

* Organisation, technology, resources
* Identify, Select, Store, Protect, Provide, Manage
* 3 copies, 2 types of storage, 1 different location (disaster zone)
* National Digital Stewardship Alliance’s matrix for levels of digital preservation:
* Storage
  + 3 copies, different threats
  + Plan & actions for obsolescence
* Integrity
  + Verify and replace/repair regularly
* Control
  + Determine, track, and review actions & access logs
* Metadata
  + Determine & maintain standards
  + Record preservation actions
* Content
  + Migrate, emulate etc. to ensure content is able to be accessed

## Cross-language information retrieval

<https://www.linkedin.com/advice/1/what-best-practices-cross-language-information-retrieval>

* Identify query language
* Translate both query and documents (into either common or multiple language/s)
* Rank and filter search results
* Relevance, usefulness, keywords, topics, categories, ratings, reviews, popularity, freshness, location

<https://nios.ac.in/media/documents/SrSecLibrary/LCh-016B.pdf>

<https://egyankosh.ac.in/bitstream/123456789/33102/1/Unit-4.pdf>

* Catalogues: author, title, both, subject
* Dictionary catalogue: includes subjects, titles, authors etc all in alphabetical list
* Easy to search, but can be clunky to cross-reference
* (less than ideal if the goal is MLIR)
* Classified catalogue: basically the LCCL system, topics have a number (requires alph index)
* Similar content grouped together
* Have to search topic index to find number to find resources

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2232200/>

<https://arxiv.org/abs/2210.06633>

## Chat with Gianluca Demartini

* Low-resource neural machine translation for query stage current standard
* Huggingface
* Elasticsearch & Apache lucene core currently being used for search stage
* <https://lucene.apache.org/core/>
* CLEF conference notes 5-10 years ago
* Older methods obsolete in some contexts but might be useful

Low-resource Neural Machine Translation: Methods and Trends <https://dl.acm.org/doi/full/10.1145/3524300>

* Back-translation: target-source, then use that to source-target for dummy parallel dataset
* Transfer learning – train on high-resource dataset, then init a child model based on weights
* Use paraphrasing for more sentence pairs
* Fuzzy matching
* Pivot-based methods – translating into a different intermediary language (e.g. Fr-Sp-En)
* Syntax-enhanced methods – probably not as useful here as we’ve just got search queries
* Source-ordered target sentences?

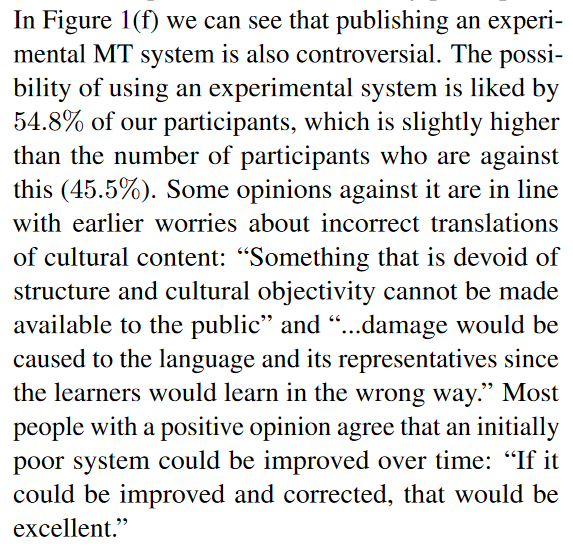
CLEF 2016 <https://link.springer.com/chapter/10.1007/978-3-319-44564-9_5>

* Typically producing single best-quality translation of query but aiming to investigate using multiple translations for better retrieval results

Document Translation vs. Query Translation for Cross-Lingual Information Retrieval in the Medical Domain, 2020 <https://aclanthology.org/2020.acl-main.613/>

* Statistical machine translation of query outperforms document translation but can be improved by NMT

Ethical Considerations for Machine Translation of Indigenous Languages: Giving a Voice to the Speakers, 2023 <https://arxiv.org/pdf/2305.19474.pdf>

* For later: deontology, normative ethics
* “For example, one comment talks about previous problems of their tribe, as recordings and other material taken by linguists is not accessible to them: “Wary of academic institutions since we currently have issues accessing recordings that belong to academics and libraries and are not publicly accessible.””
* 

Statistical Machine Translation of Australian Aboriginal Languages: Morphological Analysis with Languages of Differing Morphological Richness, 2007 <https://aclanthology.org/U07-1019.pdf>

* Separating out suffixes – helps in cases where it’s a significant feature of the language, less when it isn’t