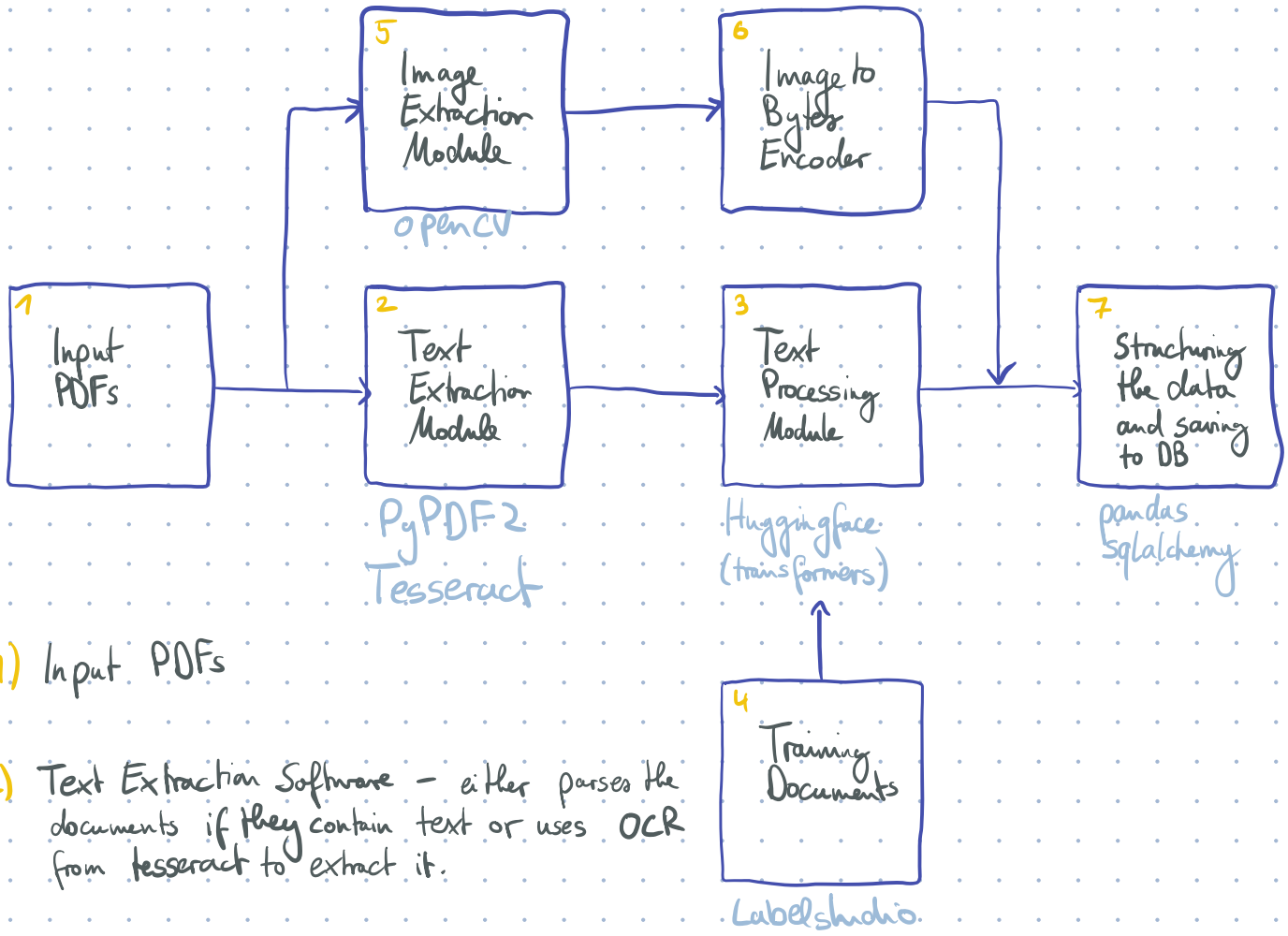
System design



1) Input PDFs

2) Text Extraction Software – either parses the documents if they contain text or uses OCR from tesseract to extract it.

3) Text Processing Module – a transformer from Huggingface that is trained on similar files and outputs the relevant information such as name, description etc.

4) Training Documents – documents that are extracted the same way as we use it later (for consistency), labeled using Labelstudio. Putting a gold standard aside.

5) Image Extraction Module – using OpenCV to extract images from the documents

6) Image-to-byte-Encoder – in order to save the images as strings in our database, we convert them to bytes.

7) Structuring the data – Finally, we put everything together in a pandas dataframe and saving it into our data warehouse

Potential Challenges

1) The training data might not be a good representation of the kind of documents that are being processed in production.

2) We might have a high variety in input documents that are all to be processed differently.

3) A large number of input samples is needed to have a good model. They also might take a lot of time to label.

4) Multilingual support can be tricky, especially with a large number of languages

5) Scalability is a problem when many documents are being processed at the same time, especially the Text Extraction.

6) Tesseract is not perfect, therefore we might encounter issues with documents that are not being processed correctly.

7) Image and text quality of input documents needs to be good so that the Text Extraction Software is more reliable.

Possible Solutions

1) Constant testing and monitoring over time.

2) We need to make sure the training data covers all different types of potential inputs.

3) Ask stakeholders to provide a large number of documents and working students to help labelling. Alternatively, start a group competition. Whichever team labels the most documents, wins a small price.

4) Get documents in multiple languages. Alternatively, use Chat GPT's API to translate.

5) Create an asynchronous process, extracting the text from the documents seperately or use cloud infrastructure

6) Use parameters on Tesseract that tested to be helping or involve humans in the loop.

7) Use pre-processing steps like filters and noise reduction.