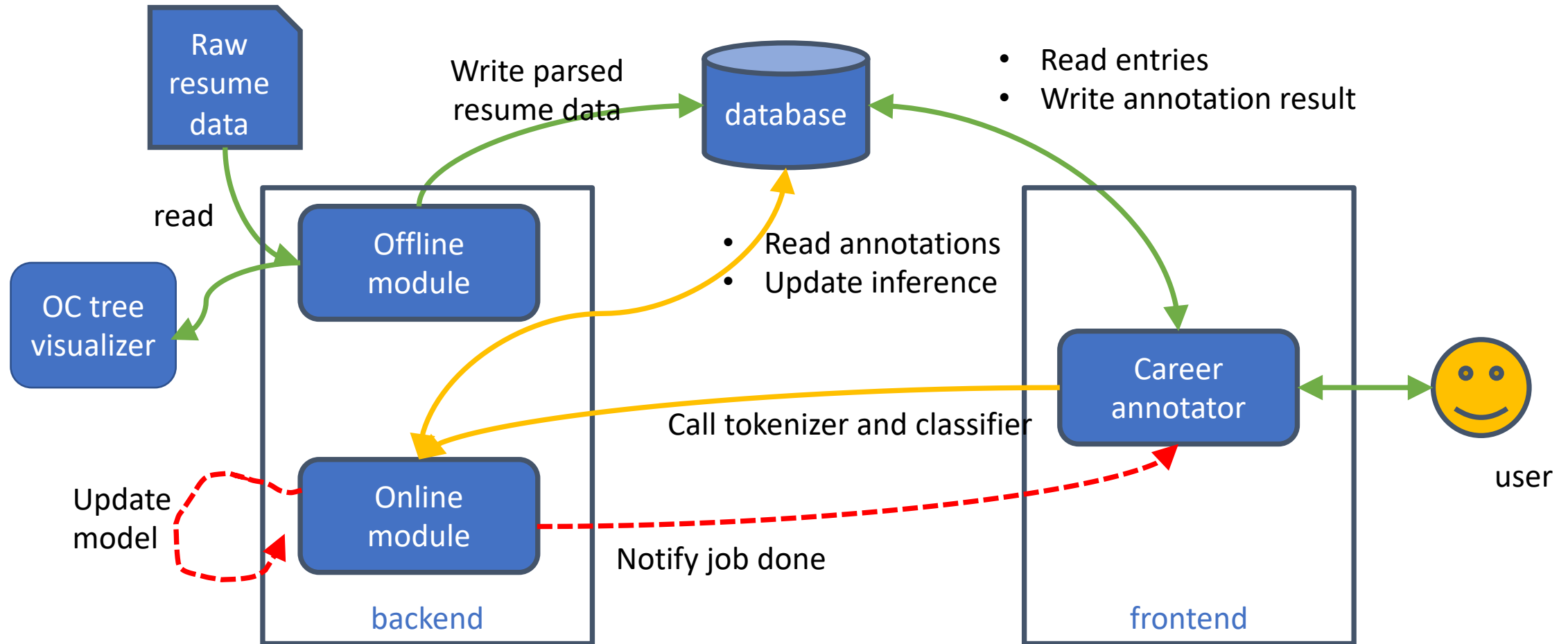
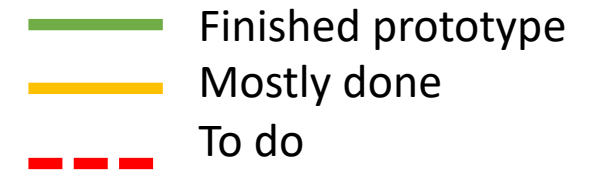


Meeting Agenda

2020.5.5

Current Progress

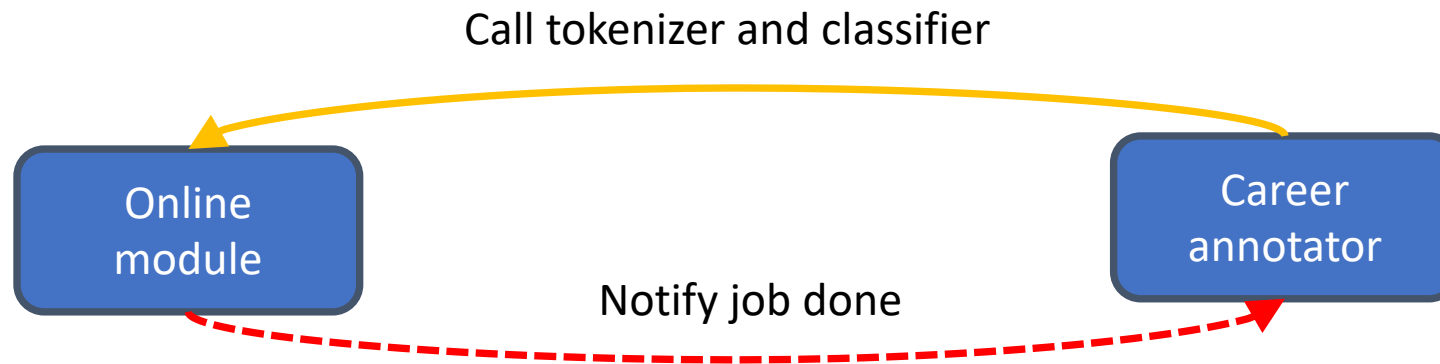


Demos

- Yang/Calvin: Back-end processing
- Zhiwei: Front-end and database
- Jiashuo: OC tree visualization

To-do: Communication Method

- Synchronous
 - Long response time
- Asynchronous
 - Invoke backend as a sub-process (job queue) *Load/store updated model from file or database*
 - Run backend as a service (API) best? *A persistent instance*



Proposed workflow (asynchronous)

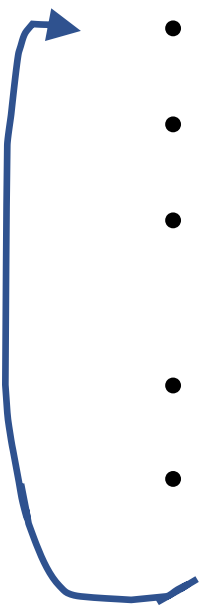
front-end

- Select **marked** entries from database
- Display entry and predicted annotation
- Get user's annotation and store to database
- Invoke backend update
- Check job.is_done ?
 - If yes, show message ...

back-end

- Get manual annotations from database
- Update model
- Rerun inference on all entries
- Mark entries to be labeled next
- Set job.is_done()

Start job



To-do: Label Classification

- Get initial training data for label classification
- Classifier choice
 - Pre-trained word/sentence embedding model
 - Chinese BERT model *(can also work without initial training data)*
 - FastText: <https://fasttext.cc>
 - Train from scratch
 - e.g. Naïve Bayes classifier, other models

Next week

- communication with online-module (Zhiwei & Jiashuo)
- classification demo (Calvin)
- front-end improvement (Zhiwei)
- OC tree visualizer improvement (Jiashuo)