CLARIAH - SIC 4.2

2022-01-13

Jan Wijffels: jan.wijffels@vub.be

DIGI - VUB - Brussels

VRIJE UNIVERSITEIT BRUSSEL

Opzet

VRIJE
UNIVERSITEIT
BRUSSEL

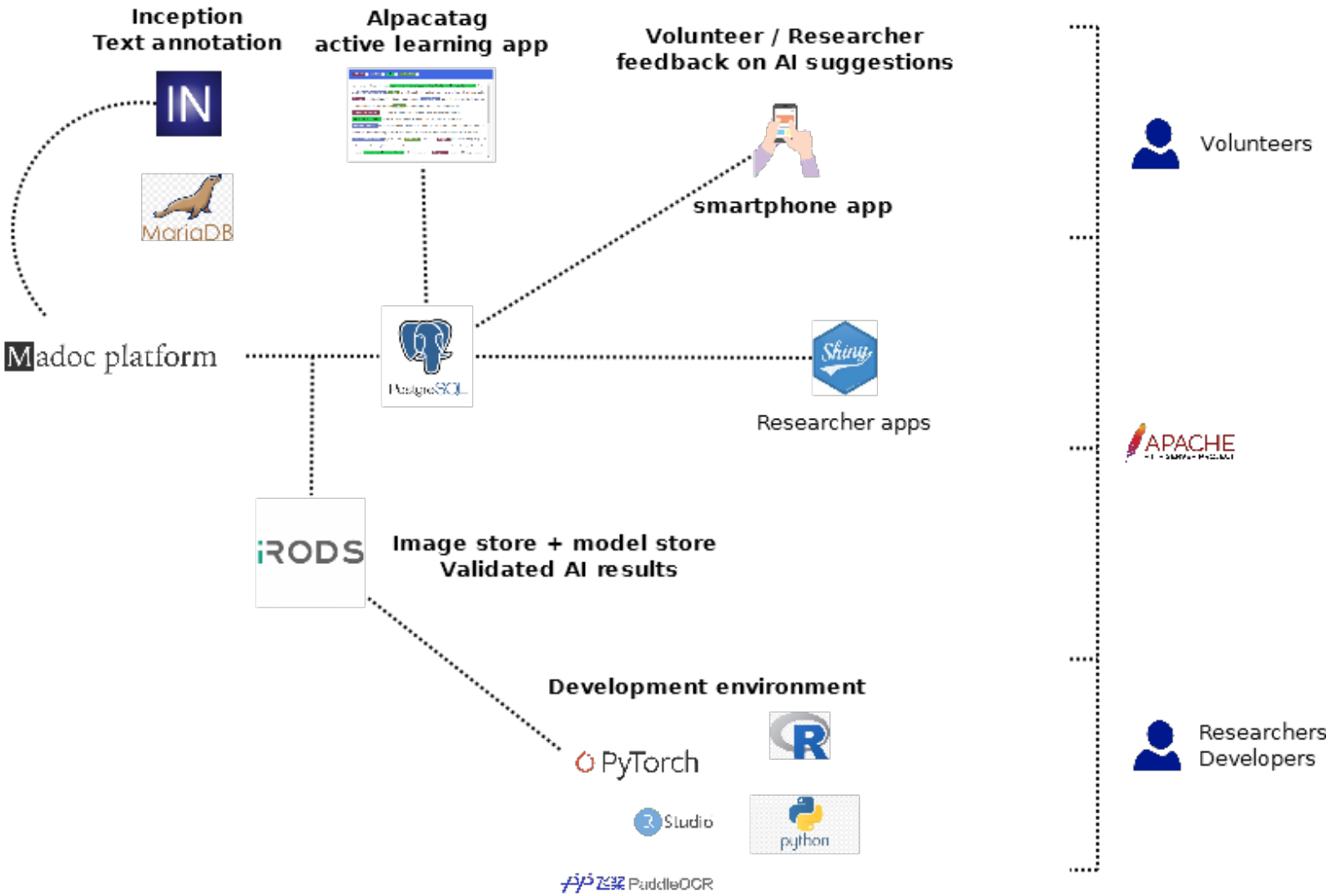
Scope SIC 4.2

AI-driven Participatory Digital Asset Enrichment

SIC 4.2 will provide a provide a set of scenarios, tool integration options, web and smartphone applications to allow researchers in digital humanities alongside volunteers to enrich their digital assets which either consists of images of text scans and raw transcribed texts with richer content through a combination of NLP technologies alongside human-in-the-loop interventions.

- Allow researchers/volunteers in digital humanities to enrich digital assets with content obtained through techniques such as NLP, HTR, OCR.
- Builds upon the CLARIAH-VL architecture, and further intends to connect to VSC Tier1/2 and DATA components.

How


Inception
Text annotation
Alpacatag
active learning app
Volunteer / Researcher
feedback on AI suggestions
smartphone app
MariaDB
Madoc platform
Researcher apps
iRODS
Image store + model store
Validated AI results
Development environment
PyTorch
Studio
python
PaddleOCR
Volunteers
APACHE
Researchers
Developers

- integrate with components of CLARIAH
  - use the crowd-sourcing toolkit Madoc as a backend platform
  - allow to connect to NLP models trained on SIC 5, use the Entity Referencing and the Linked Open Data toolkits (SIC 3.1 / SIC 3.2) on handwritten texts
  - allow to store data on the Vlaamse Supercomputer through iRODS and allows to build custom-made HTR and NER models on the Vlaamse Supercomputer
- integrate Madoc and SIC 5 models with the Inception human-in-the-loop annotation tool (`https://inception-project.github.io`)
- build smartphone applications allowing a researcher to collect feedback on straightforward tasks generated by AI toolkits
  - Named Entity Recognition / Text classification / Handwritten Text Recognition / Geospatial location identification
  - The app will allow validation, curation and improvement of model-generated output by researchers, volunteer civilians and students.

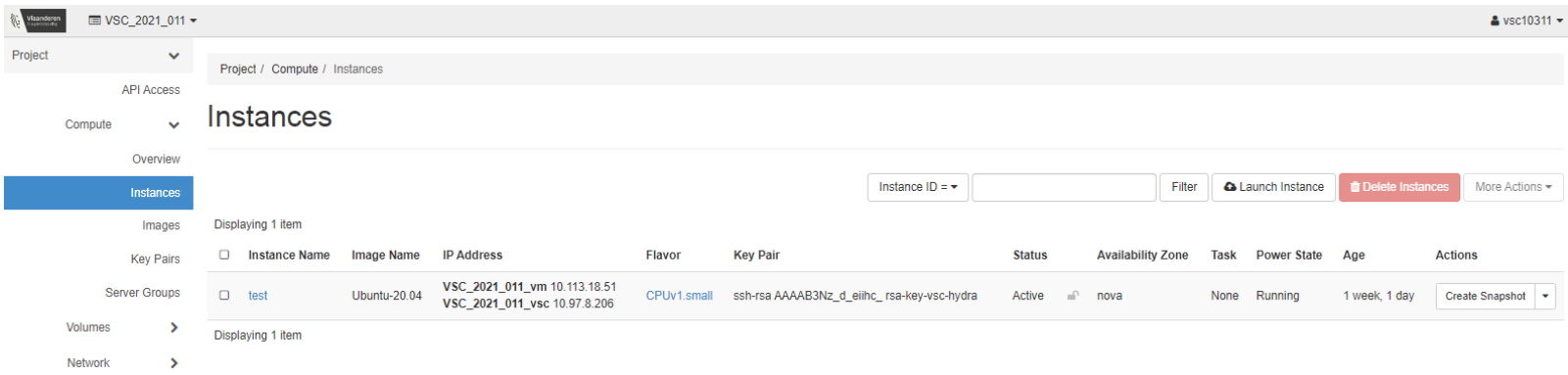
Vooruitgang

VRIJE
UNIVERSITEIT
BRUSSEL

Vooruitgang

Toegang tot infrastructuur op VSC

- Sinds eind december 2021 VSC_2021_011 project goedgekeurd:
- Toegang tot Tier 1 sinds 12/01/2022 (duration of 2 years)
- VSC Tier-1
  - 1 publiek IP adres
  - 300 GB op shared filesystem op VSC Cloud
  - 1 CPUv1 VM and 1 GPUv1 VM,
    - 4 vCPUs + 1 vGPUs
    - 128 GB of RAM + persistent local disk space of 1 TB



Connectie naar Madoc

- Samen met Tom test project opgezet op Madoc om scans uit Brugse Vrije te laten transkriberen door vrijwilligers
- R pakket opgezet om data uit Madoc te halen
  - voor o.a. transcripties van vrijwilligers
  - **`https://github.com/DIGI-VUB/madoc.utils`**

Example on Madoc

- Get transcriptions

```
library(madoc.utils)
site        <- "https://www.madoc.ugent.be/s/brugse-vrije"

## Get all projects on that madoc site
projects    <- madoc_projects(site)
projects    <- subset(projects, slug == "brugse-vrije-gebruikerstest")
projects
```

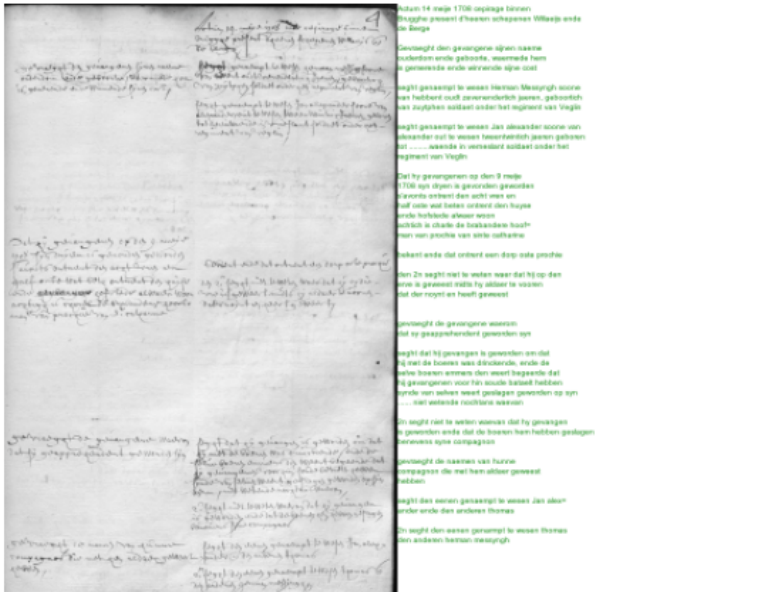
```
#>   project_id collection_id                      slug       label summary
#> 1         12          2746 brugse-vrije-gebruikerstest Brugse Vrije
```

```
## Get all manifests and canvasses of a collection
manifests <- list()
manifests <- madoc_collection(site = site, id = projects$collection_id, tidy_metadata = TRUE)
canvasses <- madoc_manifest(site = site,   id = manifests$manifest_id)

## Get annotations on a canvas or several canvasses
annotations  <- madoc_canvas_model(site = site, id = canvasses$canvas_id)
anno         <- subset(annotations, nchar(value) > 0)

## Get URL of canvas for which volunteers performed an annotation
images <- madoc_canvas_image(site = site, id = sort(unique(anno$canvas_id)))
images <- merge(images, canvasses, by = "canvas_id")
images <- images[, c("manifest_id", "canvas_id", "height", "width", "image_url")]
```

- Combine annotations with image url and manifest metadata

```
anno <- merge(images, anno, by = "canvas_id", all.x = TRUE, all.y = FALSE)
anno <- merge(anno, manifests, by = "manifest_id", all.x = TRUE, all.y = FALSE, suffixes = c("", "_manifest"))
anno <- subset(anno, !is.na(value) & nchar(value) > 0)
str(anno)
```

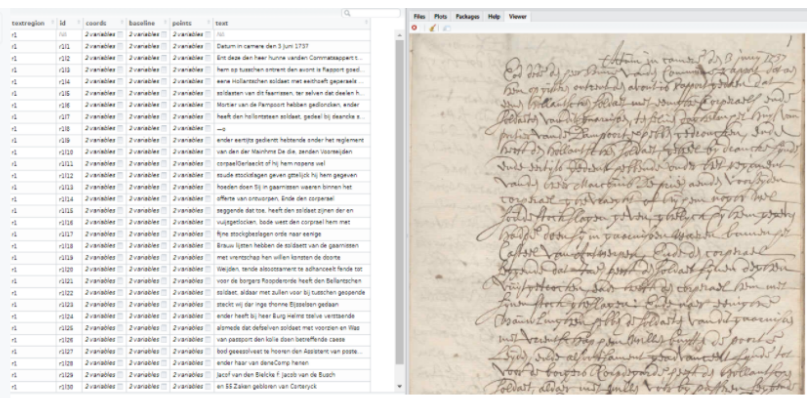
```
img  <- image_read(url)
img  <- image_resize(img, "x800")
txt  <- anno$value[[1]]

trans <- image_blank(width = image_info(img)$width, height = image_info(img)$height)
trans <- image_annotate(trans, txt, size = 10, color = "green")
image_append(c(img, trans))
```



Connectie naar Transkribus voor HTR

- R pakket opgezet om de Transkribus API op te roepen + verwerken Alto-XML / PageXML
- Functionaliteiten
  - Creatie van Transkribus projecten vanuit R
  - Uploaden van beelden naar Transkribus vanuit R
  - Layout analyse vanuit R met Transkribus tools
  - Alto-XML / PageXML downloaden + importeren in R
  - Transcripties uitvoeren vanuit R op Transkribus en Alto-XML / PageXML verwerken
  - **`https://github.com/DIGI-VUB/madoc.utils`** maar zal een eigen pakket worden

```
library(madoc.utils)
library(magick)
img <- c(system.file(package = "madoc.utils", "extdata", "example.png"),
         system.file(package = "madoc.utils", "extdata", "alto-example.jpg"))
api <- Transkribus$new(user = "jan.wijffels@vub.ac.be", password = Sys.getenv("TRANSKRIBUS_PWD"))

msg <- api$create_collection(label = "test-collection")
msg <- api$list_collections()
msg <- api$list_models(collection  = "test-collection")
msg <- api$upload(collection       = "test-collection", document = "Example document", data = img)
msg <- api$layout(collection       = "test-collection", document = "Example document")
msg <- api$transcribe(collection   = "test-collection", document = "Example document", page = 1,
                      model = "IJsberg", dictionary = "Combined_Dutch_Model_M1.dict")
api$list_job(job = msg)

## Get all documents in a collection, get pages of a document
## Once your job finished, import the PageXML file
msg   <- api$list_collection(collection = "test-collection")
msg   <- api$list_document(collection = "test-collection", document = "Example document")

pages <- head(msg, n = 1)
x     <- read_pagexml(pages$page_xml)
View(x)
image_read(pages$url)

## Delete the collection if no longer needed
msg <- api$delete_collection(collection = "test-collection")
```



Ongoing

Handgeschreven Tekst Herkenning modellering

- PyLaia HTR model gebouwd op Getuigenissen data op VSC
- Verbeteren lijn segmentatie om CER lager te krijgen

Voor Getuigenissen

- IIIF image server `https://sipi.io` opzetten op VSC Tier 1 server
- Python module aan het opzetten om Named Entity Recognition modelbouw te vergemakkelijken op Inception
- Spatial location tool developped

Volgende stappen

- Alto-XML met transcripties opzetten met project op Madoc
- Inception migrereren van eigen server naar VSC + integreren met Madoc + voorbeeld
- BERT model bouwen op 18e/19e eeuws corpus
- Entity Recognition model op Brugse Vrije + vrijwilligers correctie
- Beginnen aan smartphone app

Contact:

- jan.wijffels@vub.be
- https://github.com/DIGI-VUB