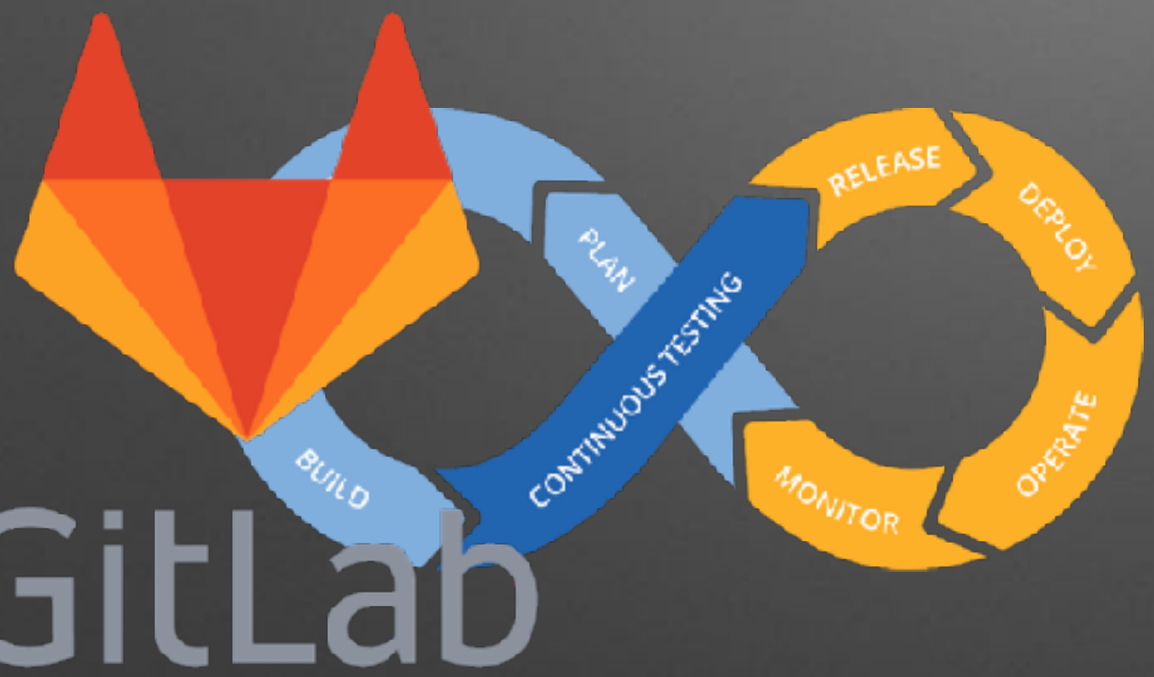
CI/CD with Gitlab



MPIWG
Max Planck Institute
for the History of Science

contents:

- motivation
- background
- TEI in Django
- Python workflow
- Gitlab ci/cd
- tl;dr

Malte Vogl
mvogl@mpiwg-berlin.mpg.de
Max-Planck-Institute for History of Science

Motivation

- Make software reusable
- Start development with sustainability in mind
- Maintenance should be simple
- Tests are a form of documentation
- docker for testing in well-defined environment
- docker-compose for simple deployment

Gitlab CI/CD: gitlab.com/users

- Runner interface to test in docker on your own servers
- Pipelines to define what happens when
- Environments to deploy to different servers

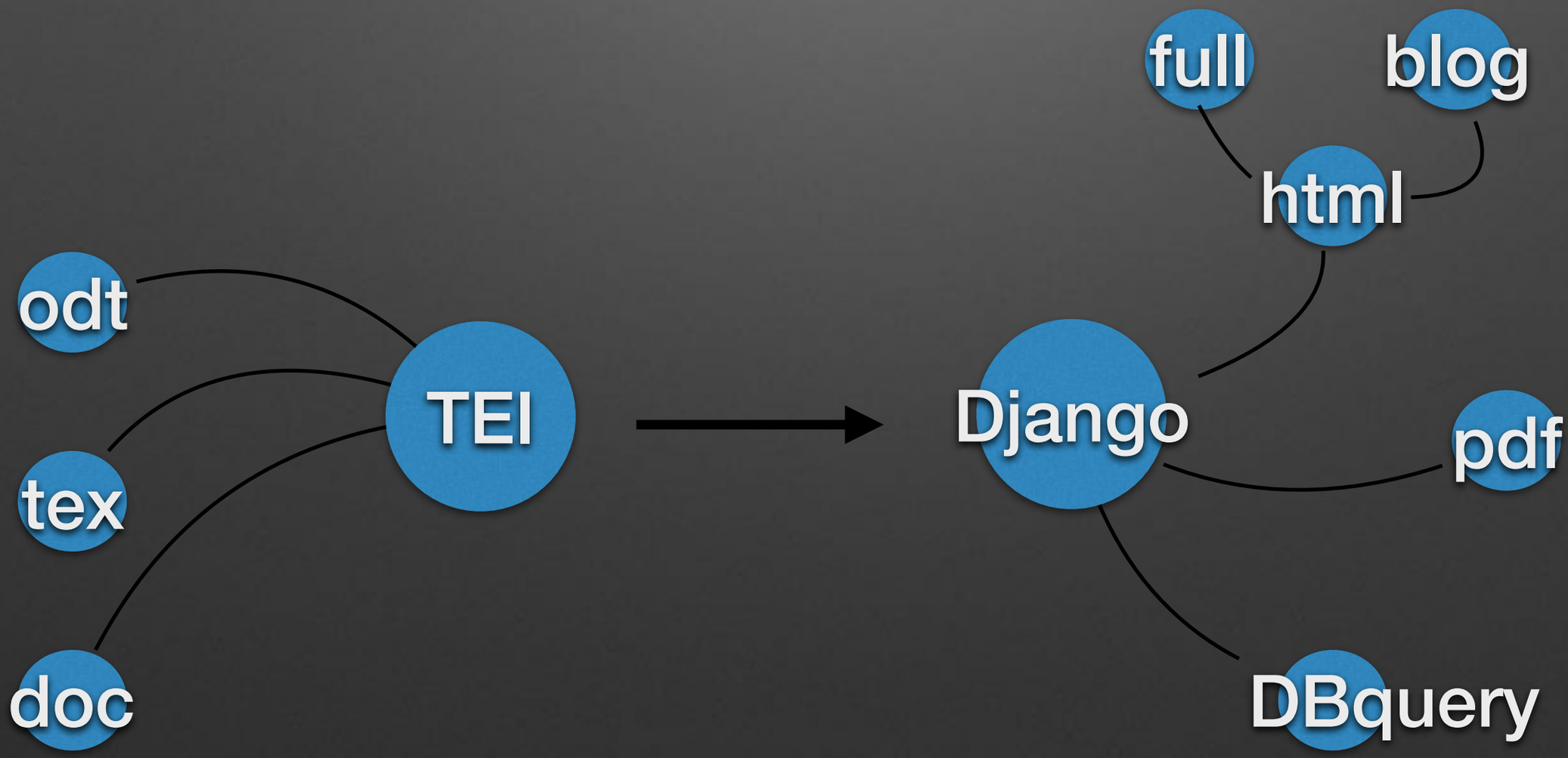
Project Background

Idea:

Implement the integration of TEI documents as fundamental unit in publication workflow

Here:

- Use existing TEI
- Import into Django DB
- Display as simple html site



TEI in Django

TEI documents: www.tei-c.org

Markup is used to define metadata and text elements semantically

```
<teiHeader>
      <author> M Vogl </author>
</teiHeader>
```

```
<body>
  <chapter>
      <p>
```

Steps in Django: www.djangoproject.com

- Define necessary models (e.g. book, author, chapter, etc)
- Write test for each model
- Write importer from TEI to models
- Write Views to display DB entry

Local Python workflow

1. Local workflow (Python):
    - Make virtual env
    - install requirements
    - run scripts/run_local_tests.sh
        - for local Django tests
    - run scripts/run_local.sh
        - for testing website locally

2. git add / commit / push
   to dev branch of gitlab repository

Gitlab CI / CD

3 .gitlab-ci.yml defines what happens next:

```
image: django:latest
```

uses Django docker image

```
.shared_hidden_key: &test
    services:
      - postgres:latest
```

```
stages:
  - test
  - deploy
```

two stages: test & deploy

```
variables:
  POSTGRES_DB: postgres
  POSTGRES_USER: postgres
  POSTGRES_PASSWORD: secret
```

Gitlab CI / CD

3.1

all commits tested

```
all_tests:
  <<: *test
  stage: test
  script:
    - pip install -r config/requirements.txt
    - PGPASSWORD=secret psql -h postgres -U postgres -d template1 -c 'create
extension hstore;'
    - sh scripts/run_tests.sh

deploy:
    <<: *deploy
    stage: deploy
    script:
        - ssh -t cloud@$DEPLOYHOST "cd django/eoa-django-test && git pull && sh
scripts/start_deploy.sh "
    environment:
        name: production
        url: https://c105-187.cloud.gwdg.de/publications/studies/312/index.html
    only:
        - master
    when: manual
```

deploy only from master

tl;dr

- develop locally
- test locally
- push to dev branch
  - code is tested

errors?

- open pull request to master branch
- automated tests again

errors?

- deploy manually
  - possibly several servers