Continuous Integration (CI) for Research Data

Experiments with adapting concepts from software development to research data management — an inhouse data repository.

Daniel Mohr, Björn Brötz

DLR German Aerospace Center
Institute of Atmospheric Physics
Weßling

April 20, 2021

Table of Contents

Overview: Typical

Your data is nothing more than a data bubble, until it is:

- described
- shared
- published

Typical steps:

1. create data (e. g. measurement)
2. analyze data
   - e. g.: using/writing Python scripts
   - sharing part of data with colleagues to get help (coworker)
3. create derivative work
4. preparation of publication of derivative work (e. g. paper)
5. create publication of derivative work

But only the publication is somehow archived.

Overview: Idea

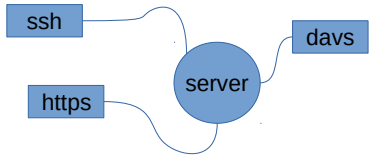
1. create data (e. g. measurement)
2. store data in a good way: – **here we create our environment** –
   - data integrity
   - data security
   - version control to track history
   - access by coworkers
   - access for analysis tools
3. analyze data
   - using data directly from data storage
   - share access with coworkers to the data storage
4. create derivative work
5. preparation of publication of derivative work (e. g. paper)
6. create publication of derivative work
7. archive: data, analyzed data and publication
8. reuse of data and/or analyzed data

Environment

- centralized server instance
  - easily cooperate
  - data integrity
- back end: git repositories
  - common tool
  - options for continuous integration (CI) – automatic processes
  - advanced stage of development
  - decentralized use possible
  - extension for large binary data: git-annex
- common access
  - ssh
  - http over SSL (https)
  - webdav over SSL (davs)
- check/use metadata by CI (e. g. git hooks on centralized server)
  - json
  - json schema

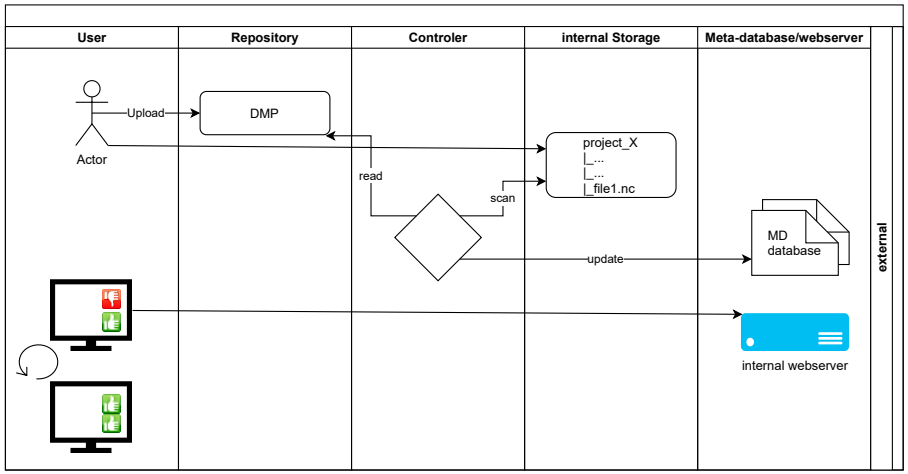
use of sustainable technologies and tools

Access Ports and Utilization: Overview



- git (uses https or ssh):
  - read write access
  - commit with message (metadata of changes, awareness of tracking)
  - GUIs available
- webdav (uses davs):
  - read only access
  - user can mount content
  - content available for scripts in file system
  - GUIs available
- server:
  - cooperated access for allowed users
  - professional storage
- git back end (on server):
  - history of changes
  - automatic processes (CI)
- git-annex (uses https or ssh):
  - large binary files (data)
  - can upload references to data
  - important for data larger than available memory

Access Ports and Utilization: Example Use Case



Access Ports and Utilization: Continuous Integration

Typical Repository:

- .dabu.json or .dmp.json
- .dabu.schema or .dmp.schema
- data/
- LICENSE.txt
- README.md

Automatic build processes or continuous integration (CI):

- validate json document '.dabu.json' with json schema '.dabu.schema'
  - inform user/supervisor by email if not OK
  - reject commit
- capture metadata from '.dabu.json' for search index
- suggest further metadata based on provided data files
- suggest further metadata based on commit messages

Prepare and collect metadata for publication in early step.

Thank you for your attention.