

GITHUB AND GITLAB

Combining the best of both worlds

06.03.2024 | JAKOB FRITZ (FZJ), THOMAS GRUBER (FAU)

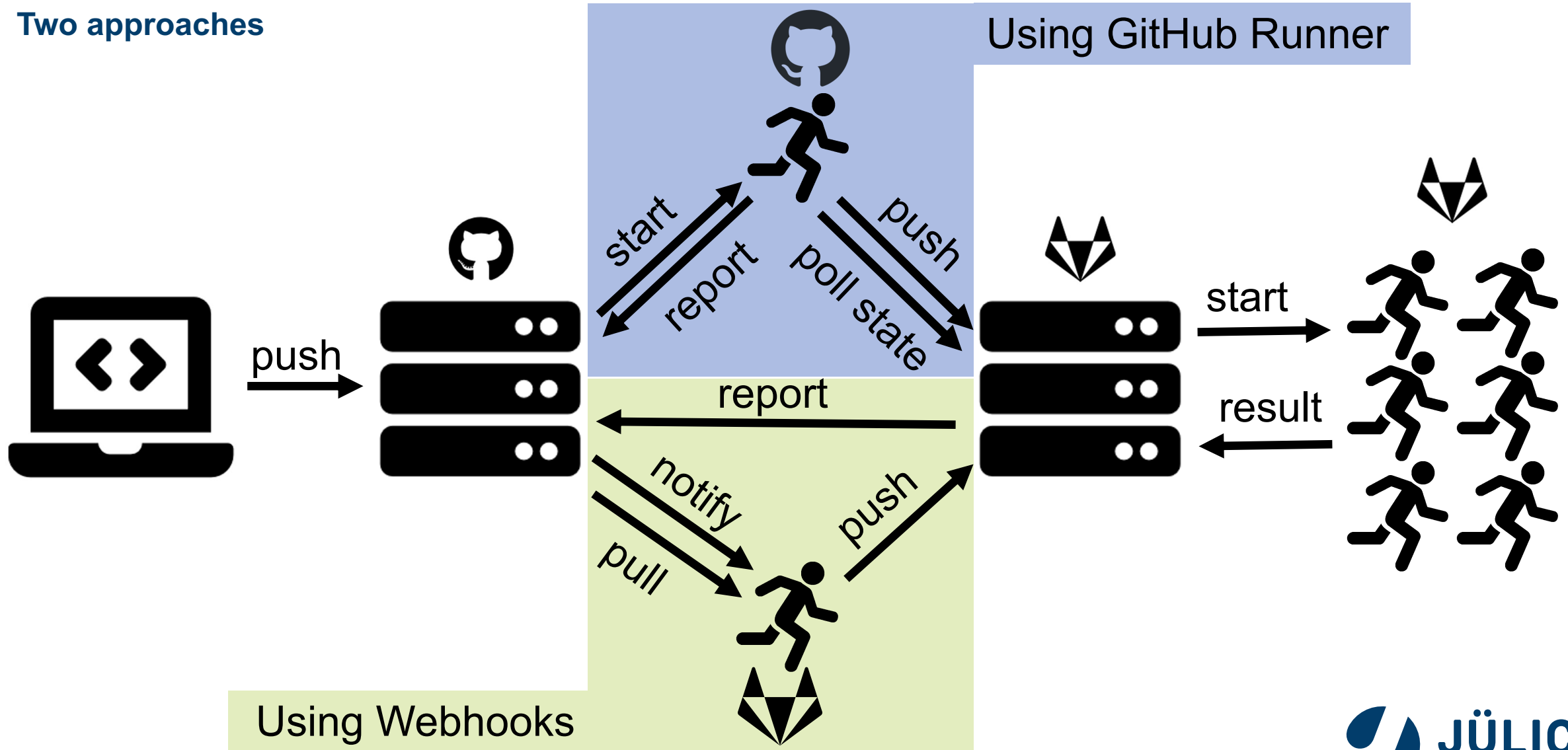
WHY COMBINE GITHUB AND GITLAB

Both Platforms have their advantages:

- 🕒 Large community
- 🕒 High visibility of software and contributors
- ♥️ Option for own Gitlab-Server on premise
- ♥️ Easy to couple with own Gitlab-Runners on own and/or special hardware

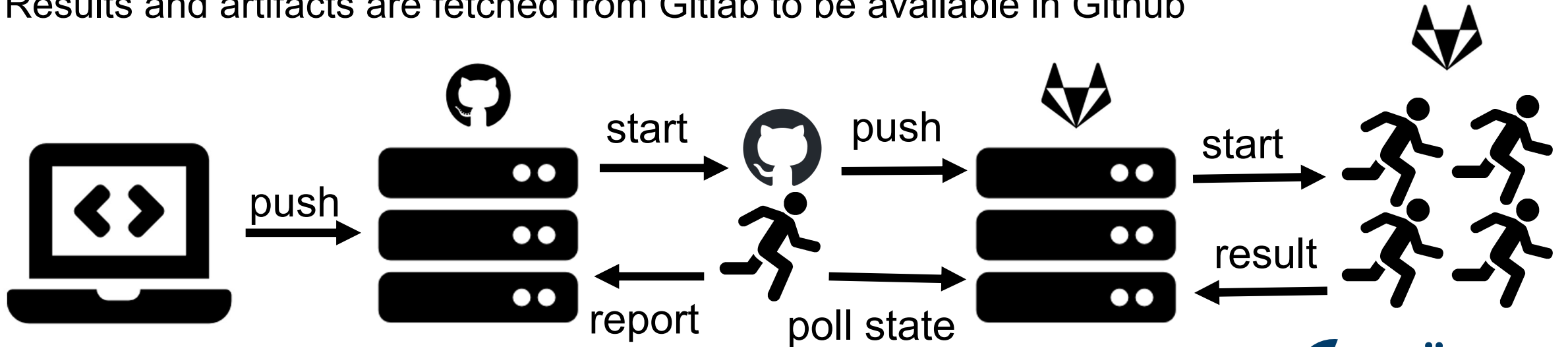
HOW TO COMBINE GITHUB AND GITLAB

Two approaches



USING GITHUB-ACTIONS

- When pushing new code (directly or via pull-request)
- A Github-Action runs mirroring the latest version to Gitlab (using pure git)
- As Gitlab detects a new version, Gitlab-CI starts to run
- Github queries Gitlab-API to get state of Gitlab-Pipeline
- Results and artifacts are fetched from Gitlab to be available in Github



USING GITHUB-ACTIONS

How to use

Github Action is used in Job

name: Mirror and get status

uses: jakob-fritz/github2lab_action@main

env:

MODE: 'all' # Either 'mirror', 'get_status', 'get_artifact', or 'all'

GITLAB_TOKEN: \${ secrets.GITLAB_TOKEN }

FORCE_PUSH: "true"

GITLAB_HOSTNAME: "codebase.helmholtz.cloud"

GITLAB_PROJECT_ID: "6627"

GITHUB_TOKEN: \${ secrets.GITHUB_TOKEN }

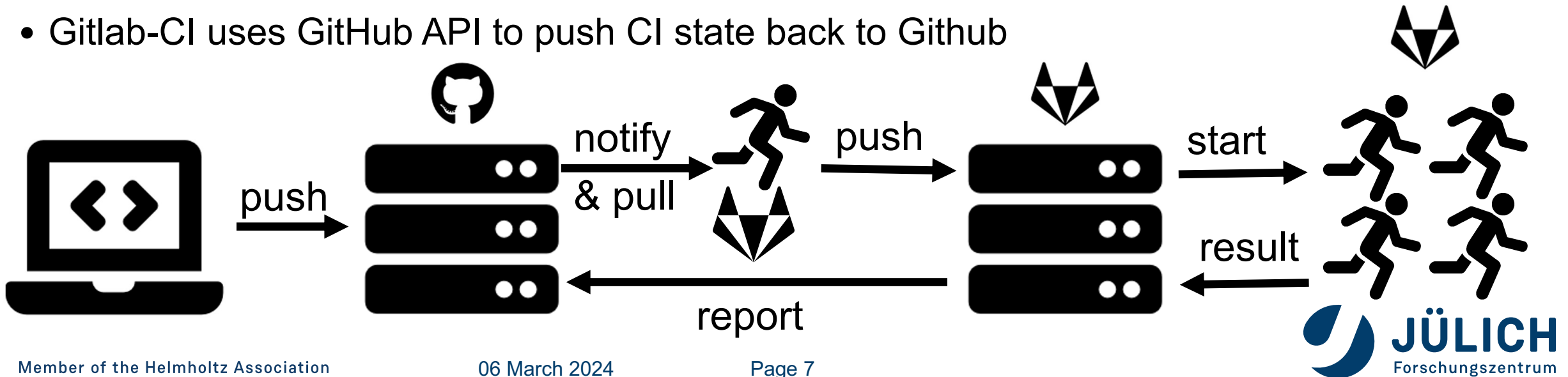
USING GITHUB-ACTIONS

Three steps

- Mirror as push from Github to Gitlab
 - Uses curl for API and GIT for pushing code
- Query state of Gitlab pipeline
 - Repeatedly polls Gitlab-API (until successful or failed), done in sh-file
- Fetch artifacts from Gitlab pipeline
 - Done in Python
 - Interacts with Gitlab-API

USING WEBHOOKS

- When pushing new code (directly or via pull-request) to project X,
- A webhook is used to trigger another Gitlab repo that takes care of syncing
- Gitlab runner in external repository pulls the current state and pushes it to the mirrored Gitlab repository of project X
- As Gitlab detects a new version, Gitlab-CI starts to run
- Gitlab-CI uses GitHub API to push CI state back to Github



USING WEBHOOKS

How to use

- Create access tokens for Gitlab sync repo and mirrored project repo
- Webhook at Github triggers remote sync pipeline
`https://gitlab.example.com/api/v4/projects/<projectId>/ref/main/trigger/pipeline?token=<Gitlab_sync_project_access_token>`

- CI sync job

```
$ git clone --mirror https://github.com/ExampleOrg/project.git
$ cd project
$ git remote remove origin
$ git push --mirror
"https://foo:${GIT_PUSH_TOKEN}@gitlab.example.com/ExampleOrg/project.git"
```

Gitlab_mirrored_project_access_token
as secret CI variable

USING WEBHOOKS

How to use

- Use Gitlab CI pipeline to send status updates via GitHub API

```
https://api.github.com/repos/${GITHUB_ORG}/${GITHUB_REPO}/statuses/${GITHUB_SHA}
```

```
notify-github-pending:
```

```
  stage: .pre
```

```
  when: always
```

```
  script:
```

```
    - ./notify_github.sh pending
```

```
notify-github-success:
```

```
  stage: .post
```

```
  when: on_success
```

```
  script:
```

```
    - ./notify_github.sh success
```

- Current implementation does **not** push artifacts back to GitHub
- Keeps authors etc. from GitHub, even if they have no account at Gitlab

ENOUGH OF PRESENTING IT

Try it out yourself

github.com/jakob-fritz/github2lab_action

doc.nhr.fau.de/sdt/ci-syncing

Feel free to send questions & feedback to both of us:

j.fritz@fz-juelich.de

Thomas.Gruber@fau.de