

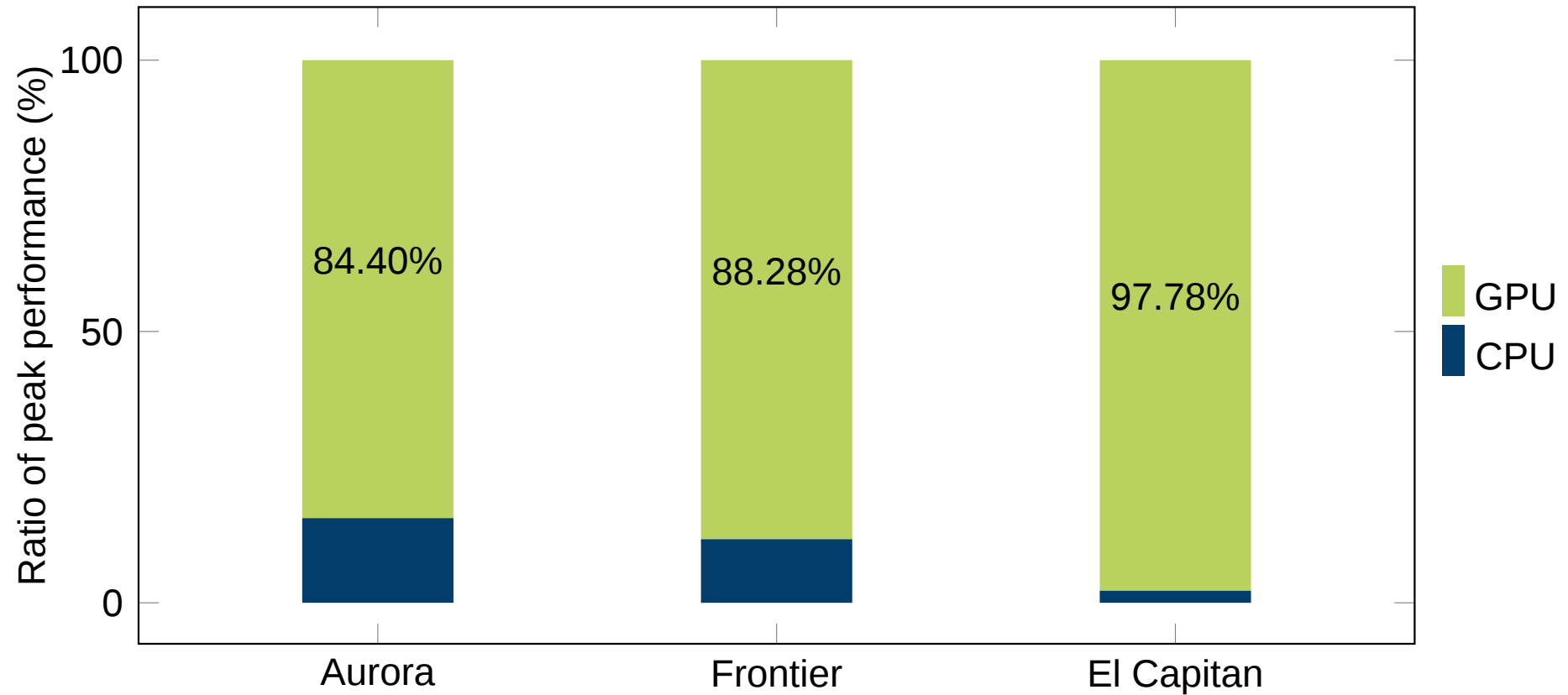


Research Software Engineering Practices

CI/CD, CB, OMG

February 13, 2025 | Robert Speck | Jülich Supercomputing Centre at Forschungszentrum Jülich

GPU-first Approach Required For Exascale



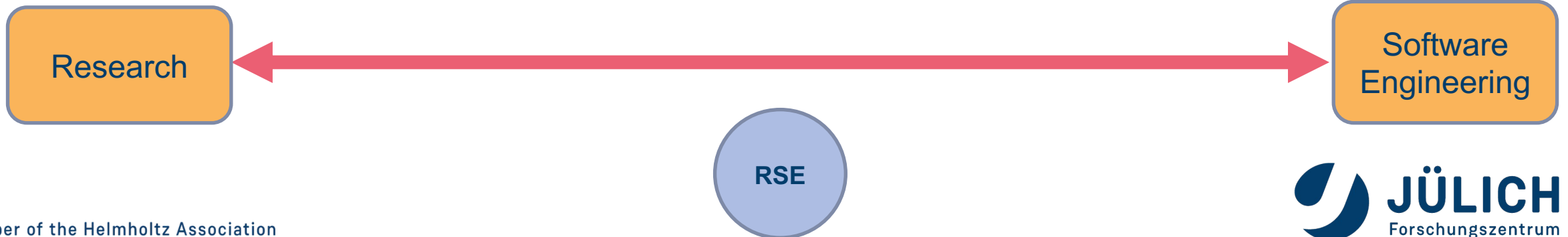
WHAT IS RESEARCH SOFTWARE ENGINEERING?

Research Software Engineering is the use of software engineering practices in research applications.

It's an intricate understanding of research with software engineering expertise.

Focuses on making research software

Reproducible Sustainable Reusable



WHEN CODE GOES ROGUE...

illegal Conversion between data types caused Ariane 5 rocket to explode 430 seconds after lift off.



Has a software bug really called decades of brain imaging research into question?

The Guardian, 2016

Science
A Scientist's Nightmare: Software Problem Leads to Five Retractions

illegal mixing of different units of measurement (SI, Imperial, and US) caused the Mars Climate Orbiter to be lost on entering orbit around Mars in 1999.

LET'S NOT GO ROGUE

A first wish list for exaNLA

- Open, joint, collaborative development in a central place
- Following established standards for code publication (license, citation.CFF, DOI, ...)
- Usage of established tools (linting, testing, ...)
- Testing, testing, testing from the very beginning (but not necessarily TDD)
- A common CI pipeline...
 - covering as much as possible,
 - as fast as possible,
 - as robust as possible
- Continuous benchmarking

CHALLENGES FOR CONTINUOUS BENCHMARKING

obvious

- Define reliable and expressive benchmarks

Avoid transient problems; not too synthetic instead have real use-cases; may require domain knowledge

- In case of manual benchmarks: manpower and rigour

Automation is key at some point

- Compute time and access to hardware

Availability and use of; different architectures; permissions/authentication

- Evaluate (plethora of?) benchmarks to come to informed decision

Conflicting or unclear goals; identifying root cause; failure condition

- Reproducibility

Statistics may be an issue; use of shared resources

subtle

CONTINUOUS BENCHMARKING

Tracking Performance Evolution over Time and Systems

- HPC all about performance, monitor during **development**, on different **systems**, with different **toolchains**
- Showcase expected performance and correct setup to potential users

Explore / CI/CD Catalog

rg-rse/continuous-benchmarking/cb-pipeline
CB Pipeline 0.0.4

A component to run continuous benchmarking via JACAMAR and JUBE

Components Readme

store-benchmark-data
plot-benchmark-data

```
include:  
- component: $CI_SERVER_FQDN/rg-rse/continuous-benchmarking/cb-pipeline/plot-benchmark-data@0.0.4
```

Inputs

| Name | Mandatory | Type | Description | Default |
|-------|-----------|--------|---|-------------------------|
| stage | false | string | Name of the continuous benchmarking stages. [continuous_benchmarking] | continuous_benchmarking |

jube-benchmark

```
include:  
- component: $CI_SERVER_FQDN/rg-rse/continuous-benchmarking/cb-pipeline/jube-benchmark@0.0.4
```

Inputs

| Name | Mandatory | Type | Description | Default |
|------|-----------|------|-------------|---------|
|------|-----------|------|-------------|---------|



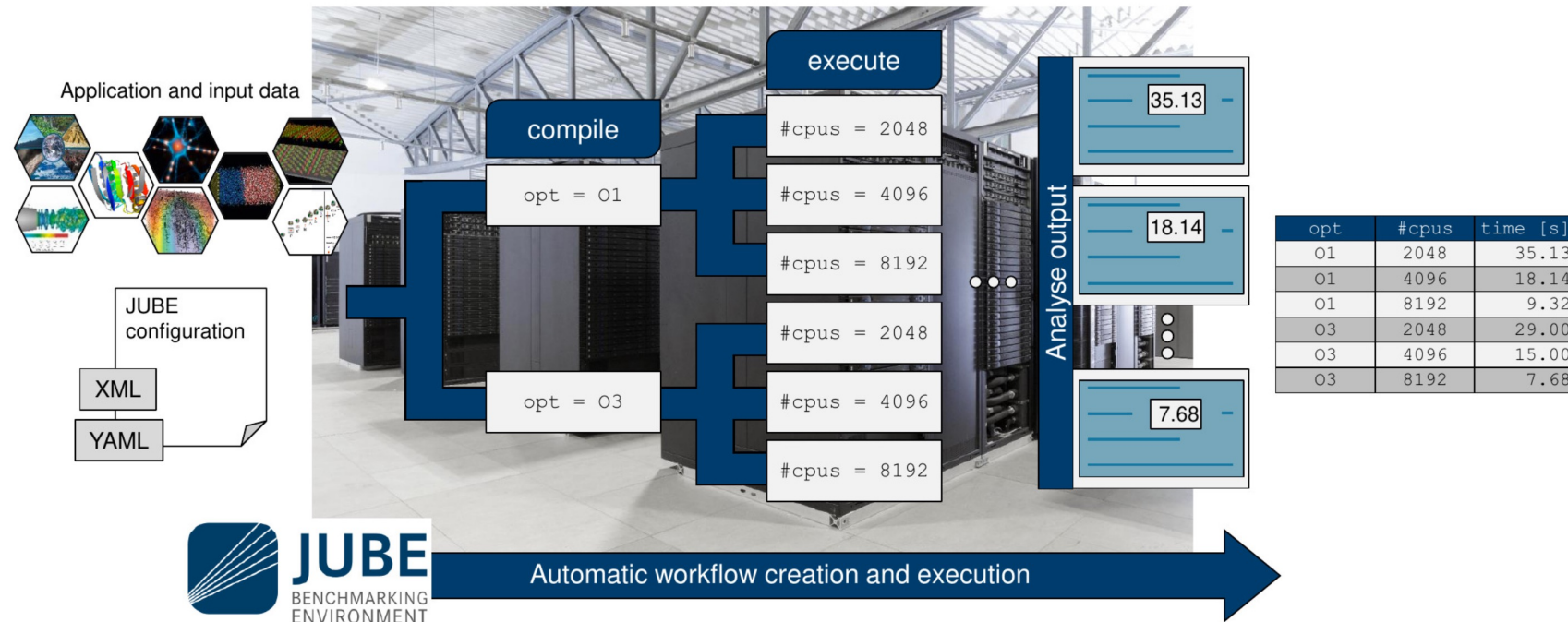
- Setup by domain specialist, **avoid additional dependencies**, especially **additional servers**
- Access HPC systems at JSC
- Ensure easy reuse via **CI/CD** components in GitLab

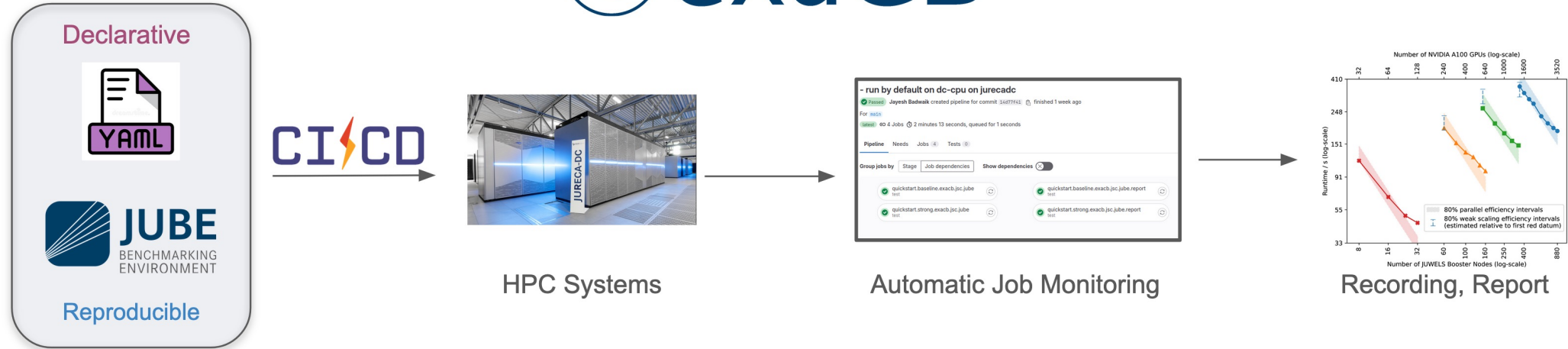
JUBE

The Jülich Benchmarking Environment

www.fz-juelich.de/jsc/jube

- Makes it easy to spawn parameter ranges (think: benchmarks, toolchains, scaling, optimisations, ...)
- Compiles and executes the code, generates job-scripts, performs analysis (patterns, statistics)
- Stores results also in `sqlite` format





Features

- Continuous Benchmarking on HPC Systems
- Template-based CI/CB (declarative syntax)
- Reduced barrier to entry for CI/CB
- Click-and-run reproducible benchmarks
- Ease of sharing workflows with community

Usecase

- >70 applications in JUREAP
- Automated report generation for JUREAP
→ Compute time review
- Energy measurements
- Performance evaluation after system update

NOW IT'S YOUR TURN

How does your wish list look like?

What worked well for you?

Do you have dos and don'ts?