Software Engineering at SDE Group, GSI

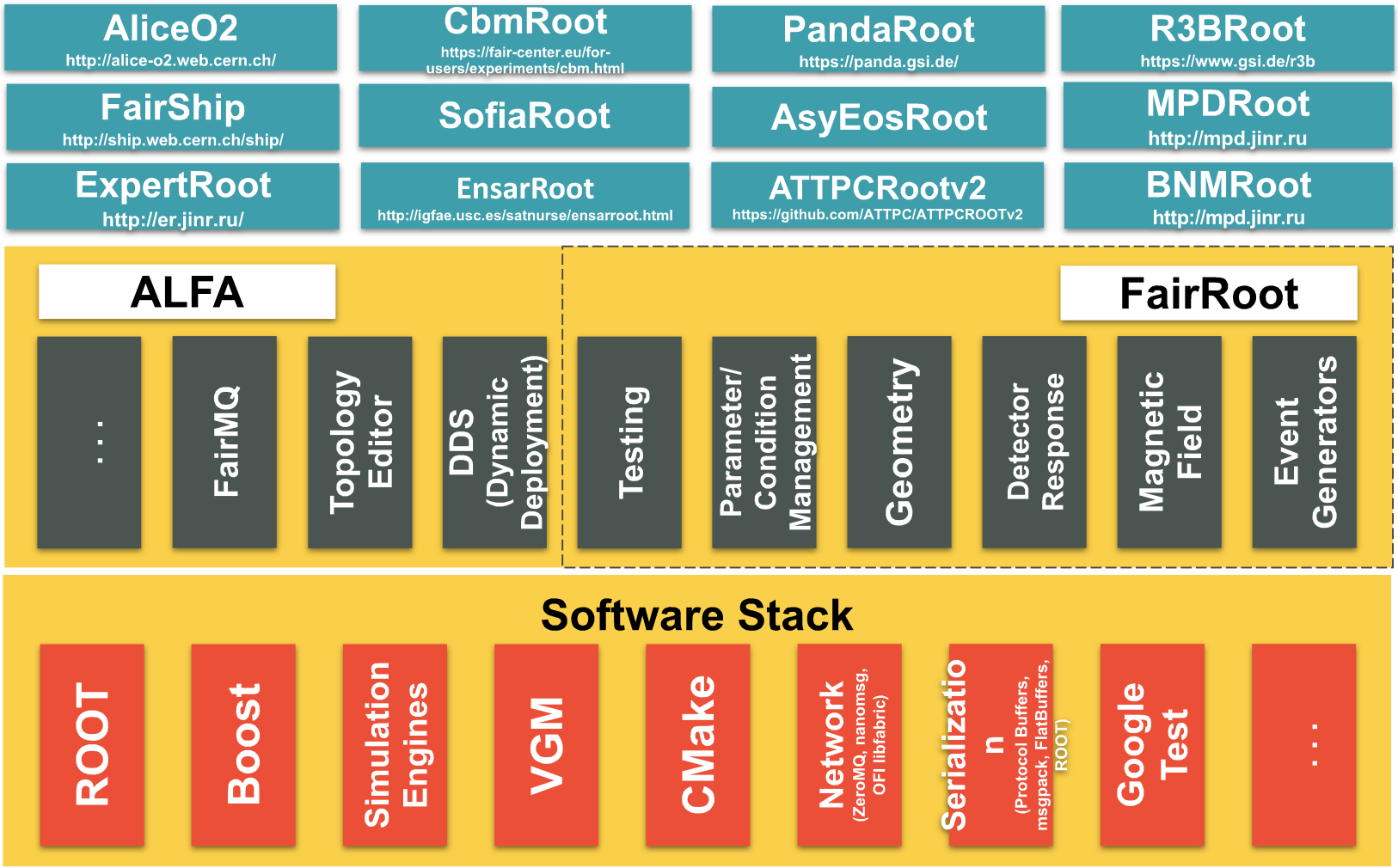
ESCAPE

Christian Tacke, GSI

Our Group

- “We”: Software Development for Experiments Group in the IT Department of GSI / FAIR.
- Most of our software can be considered library or a framework.
- Our users actually create simulation- and analysis-software using them.
- Also we‘re recently responsible for more selected infrastructure parts

Software Relations



Introduction

- Many aspects of software engineering have been shown in previous talks.

  → No “Why should you do X?”

- Workflows / tools used in our group with some notes

  https://github.com/FairRootGroup/GitWorkflow/blob/master/GitWorkflow.markdown

- Possibly recommendations (lessons learned)

Version Control

- All our stuff is in some Git Repository
- Github works very well for our public projects.
- Gitlab on a local instance also works fine for us.
- Everybody can have a private repo for whatever is needed.
  Group repos are also created on an as-needed basis.
- We mostly try to have granular commits distinct by topic and with a meaningful descriptive commit message (no squashing by pull request).

Testsuite

- All our big software projects have a testsuite, even our FairSoft “software distribution” has one.
- Our big projects use CMake / CTest + CDash, which works nicely.
- Our smaller projects use anything from home grown Makefiles to pytest.

Review : Merge / Pull Requests

- Development happens in distinct branches and then gets reviewed using a pull / merge request.
- We usually only allow changes to enter after a review (unless it‘s a hotfix)
- Accept pull / merge requests with reasonable test coverage!
- Github and Gitlab work fine for us.

CI | Continuous Integration

- Run the testsuite on each change!
  Let a machine do that for you.

- Some projects run a subset (to get quicker responses)
  on Merge / Pull Requests

- Only approving changes, if they pass CI
  (unless the CI failure can be explained)

Continuous Integration | Tools

- Jenkins
  - Extremely powerful and flexible
  - Definitely has a learning curve
  - Needs maintenance
- Gitlab CI
  - Easy entry
  - Very well integrated
  - Sometimes limited
- Github Actions
  - Very well integrated
  - Not much experience yet, work very well for Python project

CI | Using Super Computer

- A *full* testsuite run of FairSoft takes around 4 to 5 hours on a decent desktop machine
- We need to run this testsuite on around ten different OS
- Use HPC super computer
- Jenkins agent on job submit node
- Singularity containers to provide expected OS
- A full run still needs 3+ hours, still improving

- Waiting so long is more acceptable than we expected

CI | Avoid vendor lock in

- You can easily get into performing test orchestration with the dedicated features of your CI tool
  → Be careful!
- Vendor lock in: Not easy to switch your CI tools.
- Abstract things into small, dedicated utilities, that work outside of the CI tool. Use some nice scripting language.
- Developers can run this utility on their development machines!
- We‘re currently refactoring to do exactly that.

Project Management

- Structure our work using tickets / issues / projects / milestones
- Keep track of the status:
  What needs to be done, who is going to take care, what has been done.
- Ask people to file issues for their problems

- Github really has decent tools, which work very well for us.
- Gitlab works also.

Release Management

- This is really quite project specific.
- Many of our projects have stable (“master”) and development (“dev”, “develop”) branches
- Release frequently when development ongoing, release once/twice a year for stable projects in maintenance mode
- Frequent releases are only possible when proper exhaustive CI is in place, otherwise manual testing overhead too expensive

Hosting: Gitlab

- Gitlab (community edition) is hosted for GSI/FAIR
- Integrated platform to link things together, for example:
  - Which Merge Request fixes which bug
  - Which commit broke in CI
- We use it for internal / infrastructure projects.
- Sometimes feels a bit “cut off” from the rest of the world, because there is no federation
- Uses CD (Continuous Delivery).
- You don’t want to control your super computer from the cloud

Hosting: Github

- Github is free for Open Source projects.
- Also very well integrated platform
- We have a big bunch of our software there.
- It‘s all public, even the trackers, milestone planning, etc.

- “Integrating the public” is a topic in EOSC:
  Github makes this easy.
  Anyone can interact with us (file bugs, etc).
  Developers / Scientist from other facilities actually do that!

Lessons Learned / Recommendations

- Invest time in a good testsuite / test coverage.
  It IS worth it:
  – Confidence, that expected use cases are handled
  – Less time finding bugs

- Avoid CI vendor lock in (already mentioned)

- A good CI can handle multiple scenarios using containers

- It IS okay, that a CI run takes hours