Source code management

And how SCM enables collaboration

Modelon

About this session

- Target audience: have never seen git live
- After this session, you have learned:
  - …what version control can do for you
  - …basic commands and GUI
  - …about ecosystem and integration
  - …how to use version control on a simple task

Modelon

About Me

- Matthis Thorade
- Senior Simulation Engineer
- Library Developer for Thermal Management Libraries

- Studied mechanical engineering
- PhD in thermodynamics
- Using Modelica since 2009

Modelon

The Problem

What problems is source code management trying to solve?

Can you send me your newest version?

Here it is: ProjectDeliverable_v02_finalfinal_v1_20241021-MT.zip

Thanks for your quick fix on modelYZ, but now modelXY seems to be broken!?

… but it worked yesterday!

What did you change?

I got it to work now, please change ZZ. Can you integrate that back into your version?

Modelon

Motivation: what can SCM do for you?

- Just one contributor:
  - Unlimited undo & history
  - Reproducibility by SHA hash
  - Diff: compare, see changes
  - Testing

- Multiple contributors:
  - All of the above, plus
  - Collaboration: Transfer, Review, Merge
  - Requires common remote repository

Modelon

Git introduction

Git basics, tools and ecosystem

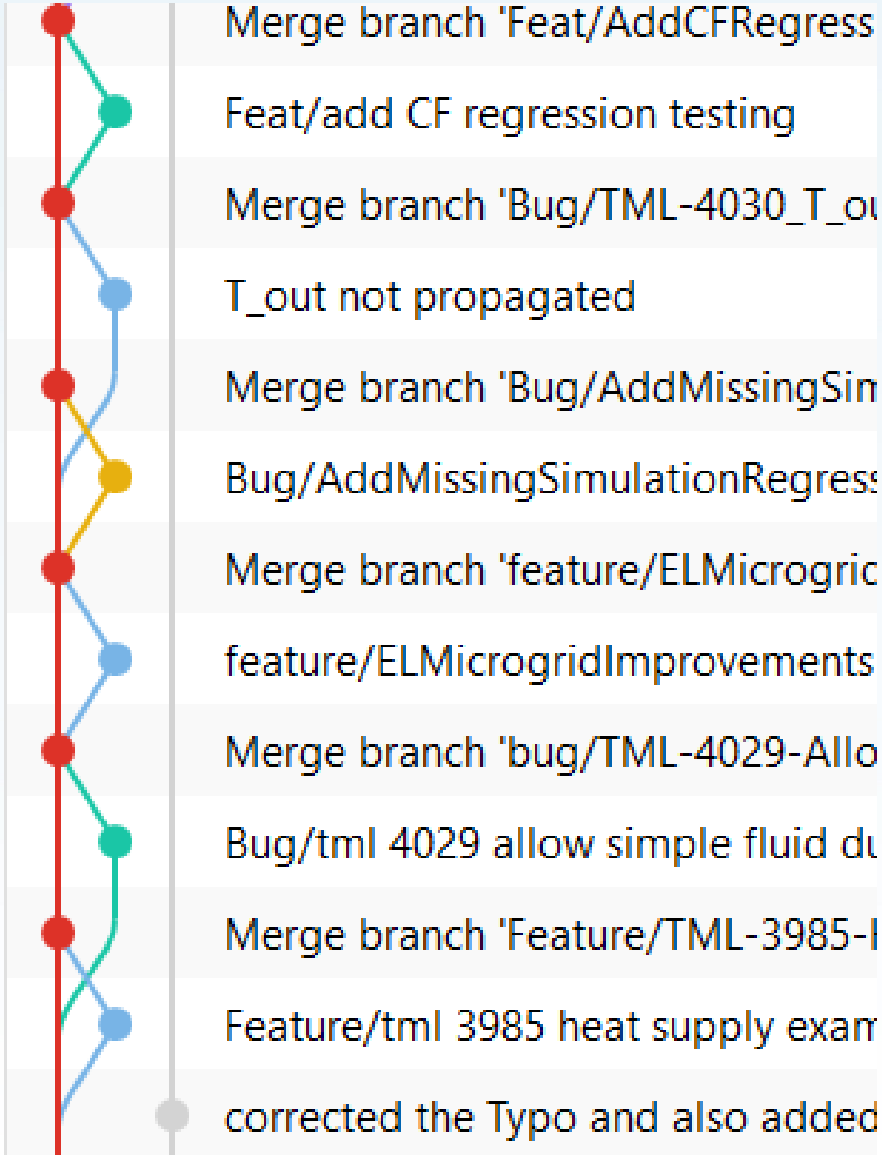
Modelon

Glossary

- Commit graph: visualization of code evolution over time, like subway map or family tree
- Commit: A single point in the git history (a ”dot” in the graph), same as revision or version in other SCM / VCS
- Branch: A line of development

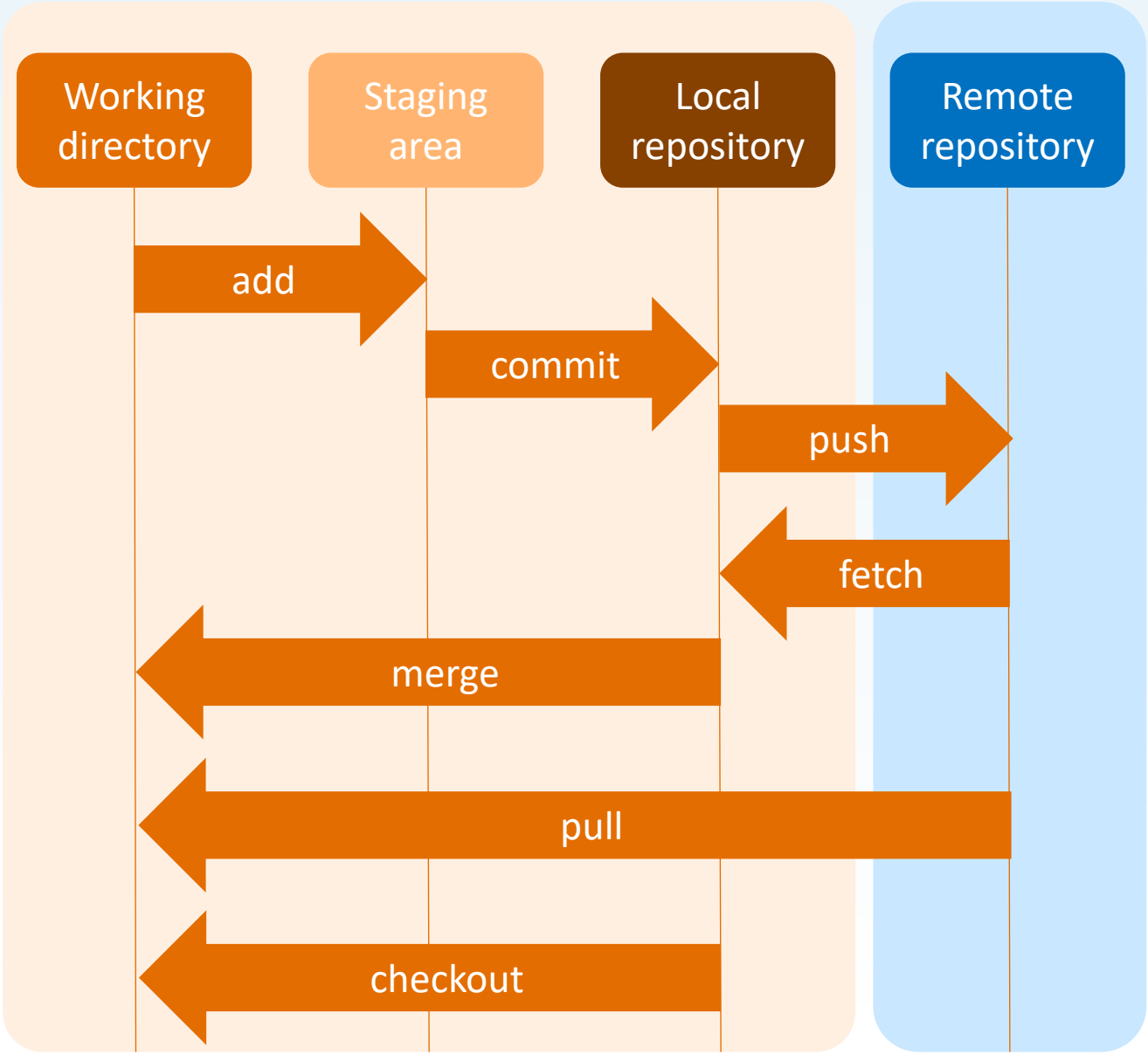
- No commits directly to ”main” branch, all work happens on branches

Demo



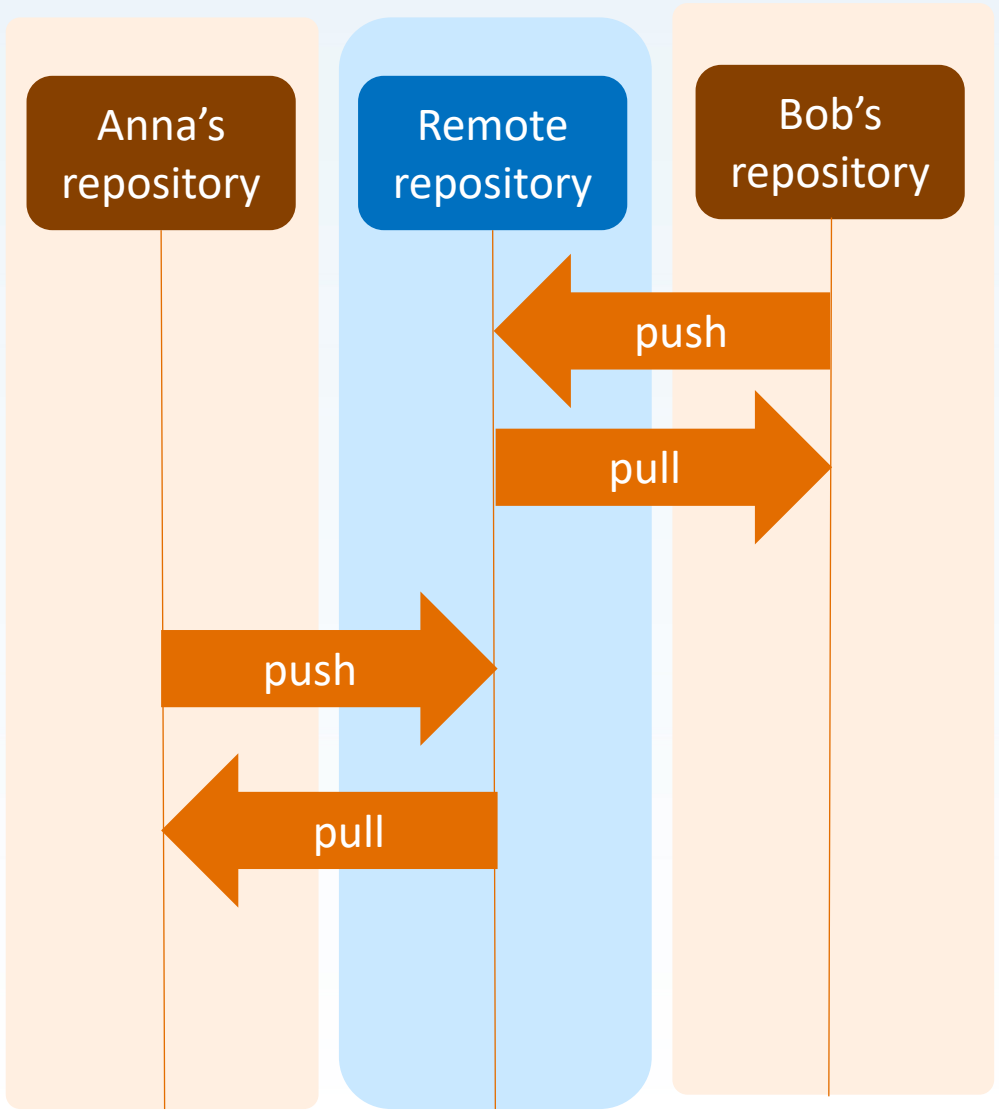
Glossary

- Fetch: Fetch information about commits in current branch from remote repository to local repository
- Checkout: Switch the local working tree to a selected commit/revision
- Push: Push the commits in the current branch from local repository to the remote repository
- Merge: To bring the contents of another branch into the current branch.
- Pull: pull = fetch + merge
- https://git-scm.com/docs/gitglossary



Collaboration

- All collaborators have a local copy of the repo, and commit their contributions to a branch
- Remote repository for syncing between multiple collaborators
- Compare and review branches, merge changes

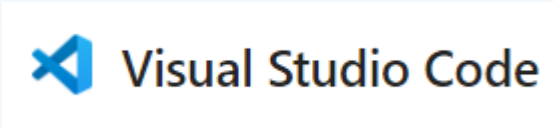


Ecosystem: git GUI

- Built into tool of your choice?

- Visual Studio Code + gitlens
- Sourcetree by Atlassian
- GitExtensions
- GitHub Desktop

- https://git-scm.com/downloads/guis



Modelon

Ecosystem: web GUI for the remote repository

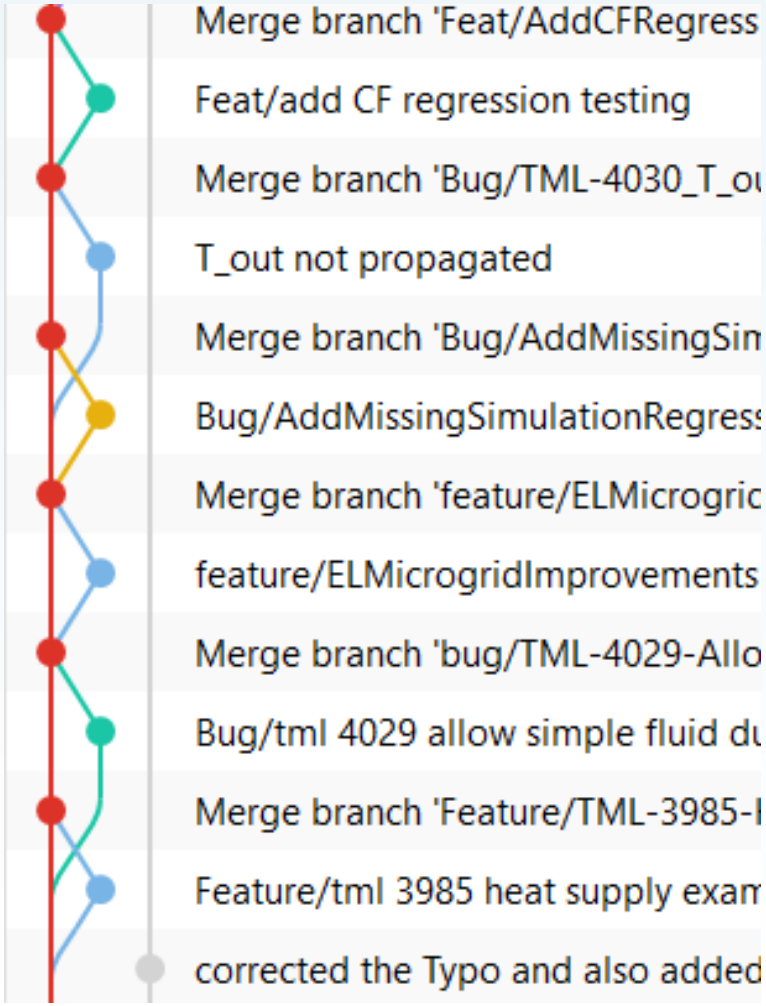
- GitHub
- GitLab
- Gitea
- BitBucket

- Typical features:
  - Browse source code with history and blame
  - Diff between two commits
  - Commit graph
  - Reviewing Pull Requests / Merge Requests
  - Issues / Tickets

Modelon

Testing and Automation

- Whenever there is a new commit to the remote repo, run some automated actions:
- Syntax check
- Translate, compile and simulate a model
- Compare results against reference results
- Benchmark performance or statistics
- Build HTML documentation
- Encrypt & release
- Report generation
- …what else?



Demo and Hands-on

Demo or example to illustrate how the problem is solved by applying the methodology or workflow

Demo 1

- Open Impact, clone ExampleTestLibrary
  - https://github.com/modelon-community/ExampleTestLibrary
- Change 1 parameter
- Commit to branch and push
- Create a Merge Request
- Check Jenkins with test selection
  - https://jenkins.modelon.com/example-external/job/ExampleTestLibrary_verify_Linux/

Modelon

Demo 2

- Share workspace configuration

Hands on

- Fork and clone of these repos:
  - https://github.com/modelica-3rdparty/
  - https://github.com/modelon-community/ExampleTestLibrary
- Create a branch
- Change a file (from Impact or from VScode), commit the change
- Push your branch to remote
- Open a Pull Request
- Have a colleague review the change

Conclusion and Q&A

- Use SCM to reproducibly transfer code to multiple computers
- Follow software development best practice: SCM with branches, testing, reviews
- Impact has git support built-in
- Sharing of workspaces

- Questions and Discussion

Modelon

Accurate Simulations. Better Decisions.