OSCI Exercises

Git Crash Course

Winter term 2026/27

Manuel Vögele & Benedict Herzog

Bochum Operating Systems and System Software Group (BOSS)

RUHR UNIVERSITÄT BOCHUM

RUB

BOCHUM OPERATING SYSTEMS & SYSTEM SOFTWARE

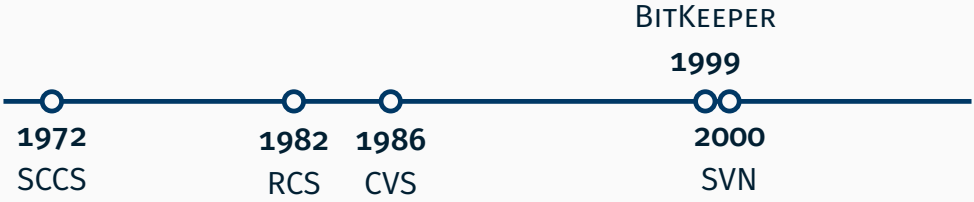
A small history of version control systems

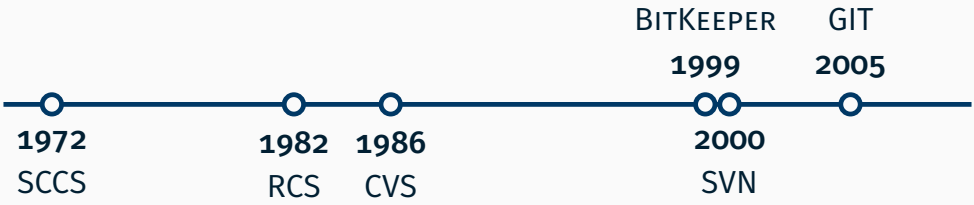
1972
SCCS

1972
SCCS
1982
RCS

1972
SCCS
1982
RCS
1986
CVS

1972
SCCS
1982
RCS
1986
CVS
2000
SVN


BitKeeper
1999
1972
SCCS
1982
RCS
1986
CVS
2000
SVN


BitKeeper
GIT
1999
2005
1972
SCCS
1982
RCS
1986
CVS
2000
SVN

Key concepts of GIT

- non-linear development (*branch*)
- integrity via SHA-1 hash
- files stored in entirety (*snapshot* instead of diffs)
- decentral (*clone*)

Initializing a new GIT repository

```
~$ mkdir example
~$ cd example
```

Initializing a new GIT repository

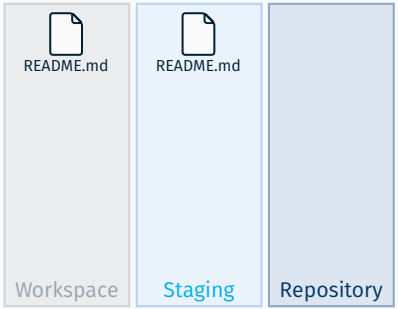
```
~$ mkdir example
~$ cd example
~/example$ git init
Initialized empty Git repository in example/.git/
```

Managing files with GIT

```
~/example$ touch README.md
```

Managing files with GIT

```
~/example$ touch README.md
~/example$ git add README.md
```

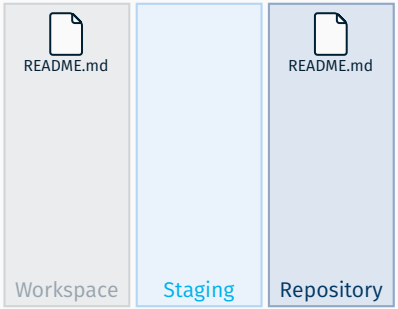


Managing files with GIT

bd2de5c

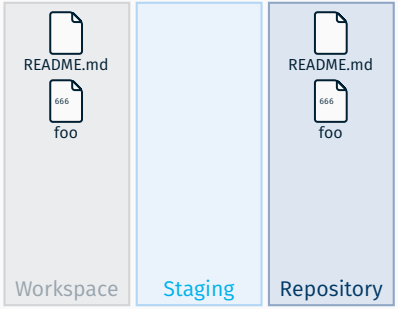
1

```
~/example$ touch README.md
~/example$ git add README.md
~/example$ git commit -m "Add readme"
[master (Root-Commit) bd2de5c] Add readme
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md
```



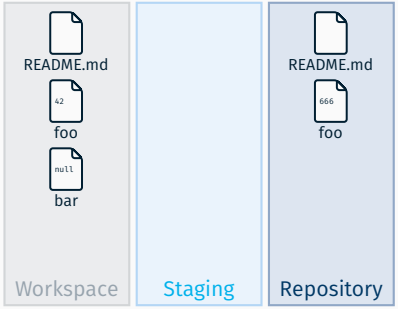
Managing files with GIT

```
~/example$ echo "666" > foo
~/example$ git add foo
~/example$ git commit -m "Create file foo"
[master df7aa5a] Create file foo
 1 file changed, 1 insertion(+)
 create mode 100644 foo
```



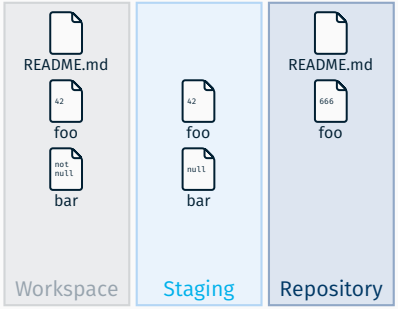
Managing files with GIT

```
~/example$ echo "42" > foo
~/example$ echo "null" > bar
~/example$ git status
On branch master
Changes not staged for commit:
    modified: foo

Untracked files:
    bar

no changes added to commit
```



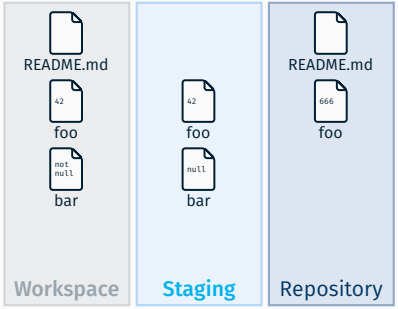
Managing files with GIT

```
~/example$ git add foo bar
~/example$ echo "not null" > bar
~/example$ git status
On branch master
Changes to be committed:
    new file: bar
    modified: foo

Changes not staged for commit:
    modified: bar
```



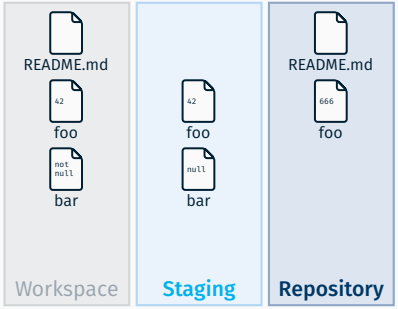
Managing files with GIT

```
~/example$ git diff
diff --git a/bar b/bar
index 19765bd..b263a85 100644
--- a/bar
+++ b/bar
@@ -1 +1 @@
-null
+not null
```



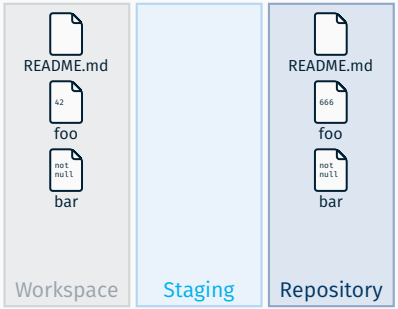
Managing files with GIT

```
~/example$ git diff --staged
diff --git a/bar b/bar
new file mode 100644
index 0000000..19765bd
--- a/bar
+++ b/bar
@@ -0,0 +1 @@
+null
diff --git a/foo b/foo
index 7cc86ad..d81cc07 100644
[...]
```



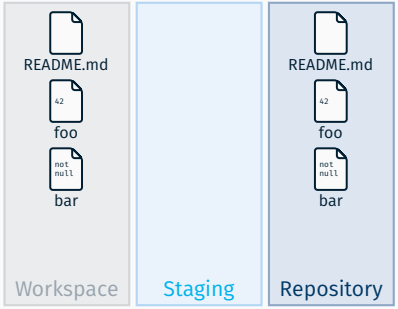
Managing files with GIT

```
~/example$ git add bar
~/example$ git commit -m \
           "Corrected foo and added bar"
[master 90f7cfe] Correct foo and add bar
 2 files changed, 2 insertions(+), 1 deletion(-)
 create mode 100644 bar
```

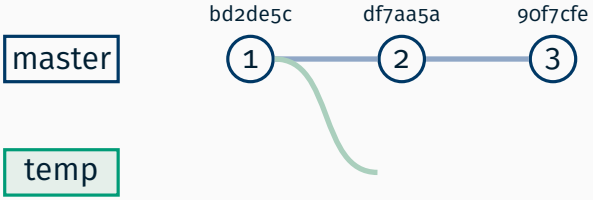


Managing files with GIT

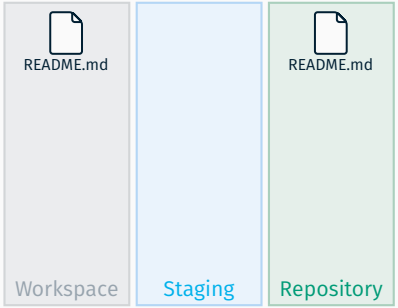
```
~/example$ git shortlog
Manuel Vögele (3):
    Add readme
    Create file foo
    Correct foo and add bar
```



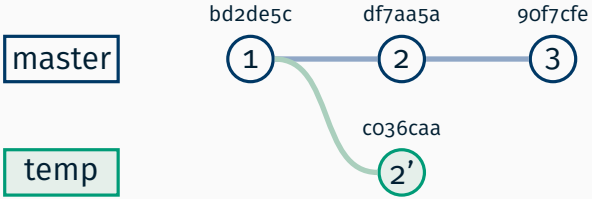
GIT branches



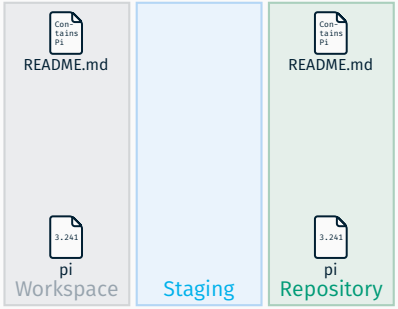

```
~/example$ git branch temp bd2de5c
~/example$ git switch temp
Switched to branch 'temp'
~/example$ git shortlog
Manuel Vögele (1):
    Add readme
```



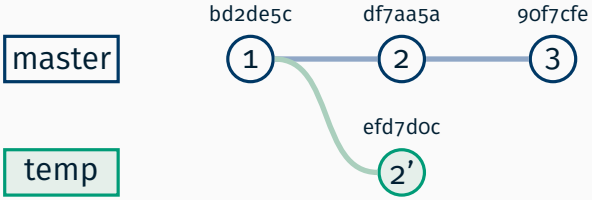
GIT branches



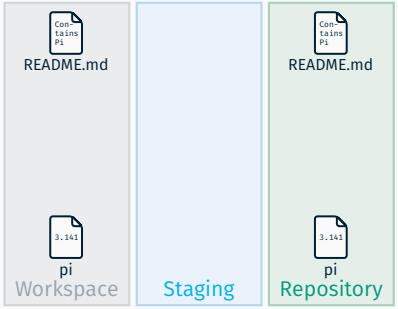

```
~/example$ echo "3.241" > pi
~/example$ echo "Contains Pi" >> README.md
~/example$ git add .
~/example$ git commit -m "Insert circle constant"
[temp c036caa] Insert circle constant
 2 files changed, 2 insertions(+)
 create mode 100644 pi
```



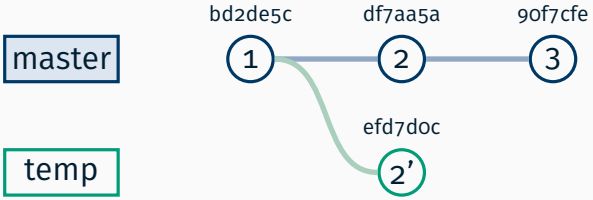
GIT branches



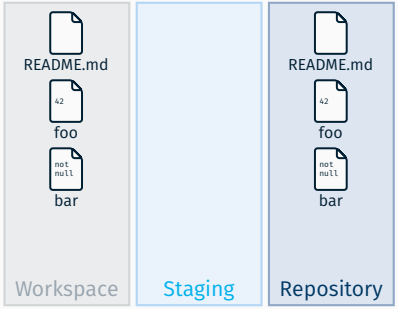

```
~/example$ echo "3.141" > pi
~/example$ git commit -a --amend -m \
                    "Add circle constant"
[temp efd7d0c] Add circle constant
 Date: Mon Oct 5 12:50:08 2020 +0200
 2 files changed, 2 insertions(+)
 create mode 100644 pi
```



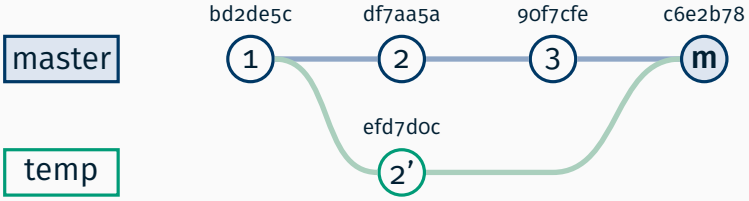
GIT branches



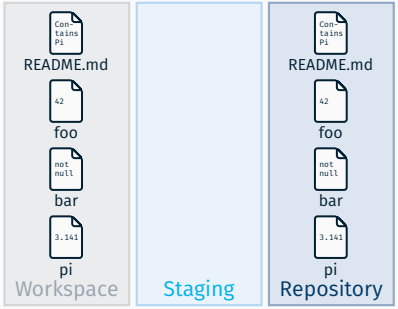

```
~/example$ git branch
  master
* temp
~/example$ git switch master
~/example$ git branch
* master
  temp
```



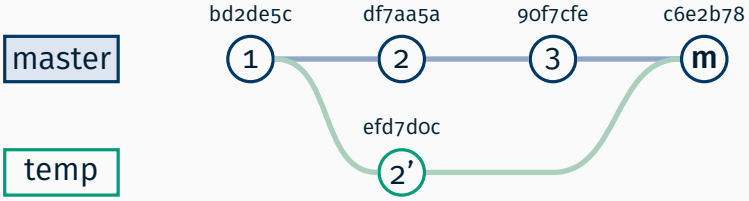
Merging GIT branches



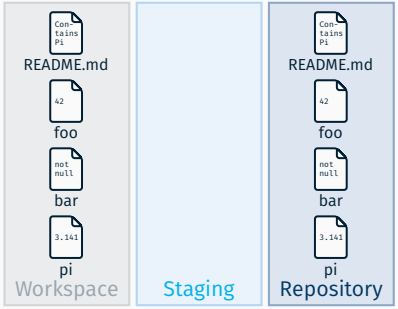

```
~/example$ git merge temp
Merge made by the 'recursive' strategy.
 README.md | 1 +
 pi        | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 pi
~/example$ git log
commit c6e2b781a60785fa2fac0d7467b5b0e7c9a7fb8c
Merge: 90f7cfe efd7d0c
Author: Manuel Vögele <manuel.voegele@rub.de>
[...]
```



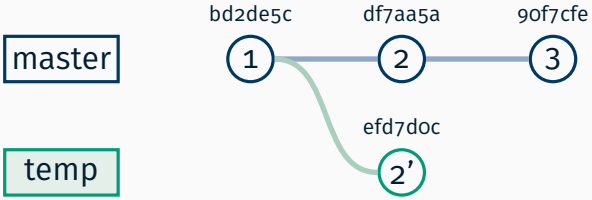
Merging GIT branches



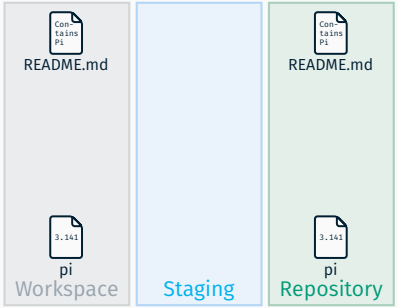

```
~/example$ git shortlog
Manuel Vögele (5):
    Add readme
    Create file foo
    Correct foo and add bar
    Add circle constant
    Merge branch 'temp'
```



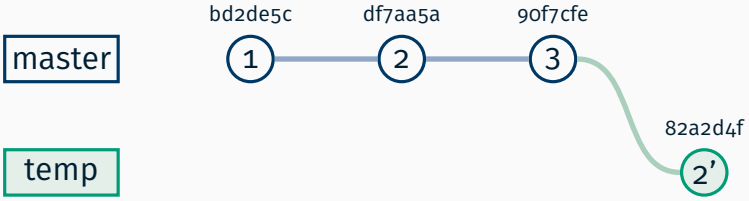
Alternatively: rewriting the GIT history



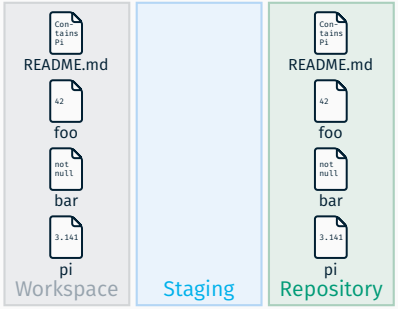

```
~/example$ git branch
  master
* temp
```



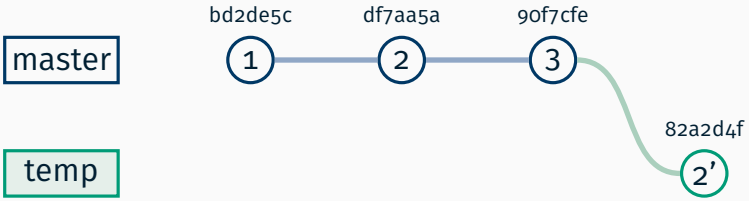
Alternatively: rewriting the GIT history



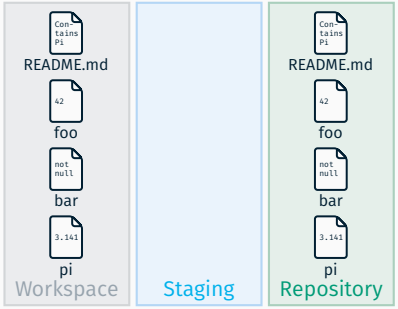

```
~/example$ git rebase master
Successfully rebased and updated refs/heads/temp.
~/example$ git log
commit 82a2d4f28d9986560aa75ea429ac5f510eebfd99
Merge: 90f7cfe efd7d0c
Author: Manuel Vögele <manuel.voegele@rub.de>
Date: Mon Oct 5 12:50:08 2020 +0200

    Add circle constant
```



Alternatively: rewriting the GIT history

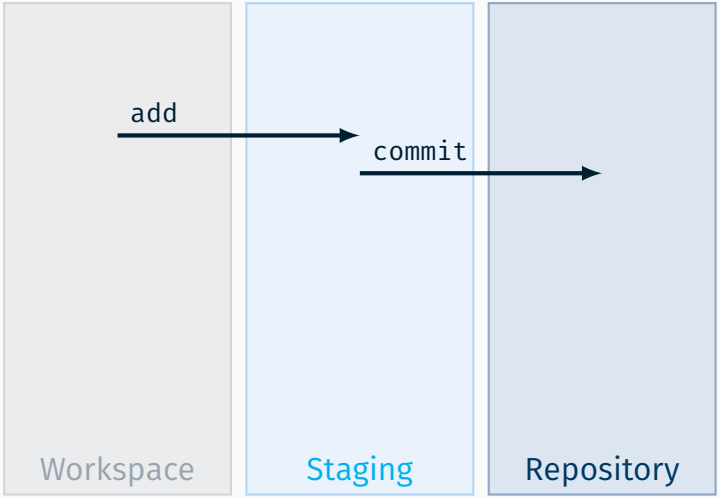



```
~/beispiel$ git shortlog
Manuel Vögele (4):
    Add readme
    Create file foo
    Correct foo and add bar
    Add circle constant
```

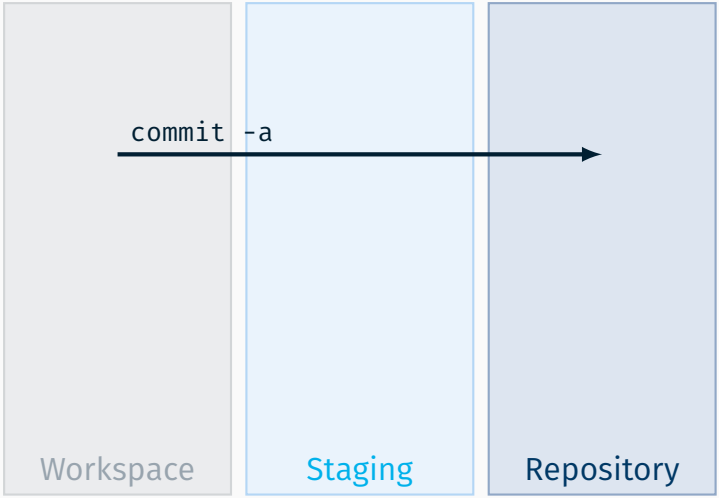


Overview: Checking files in and out in GIT

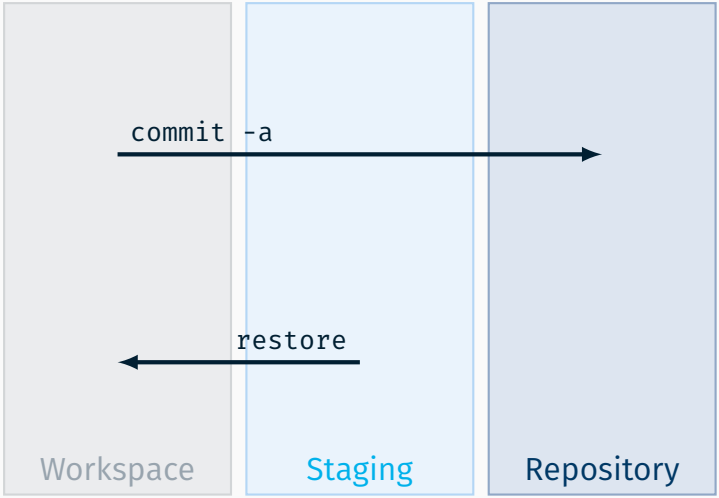
Overview: Checking files in and out in GIT



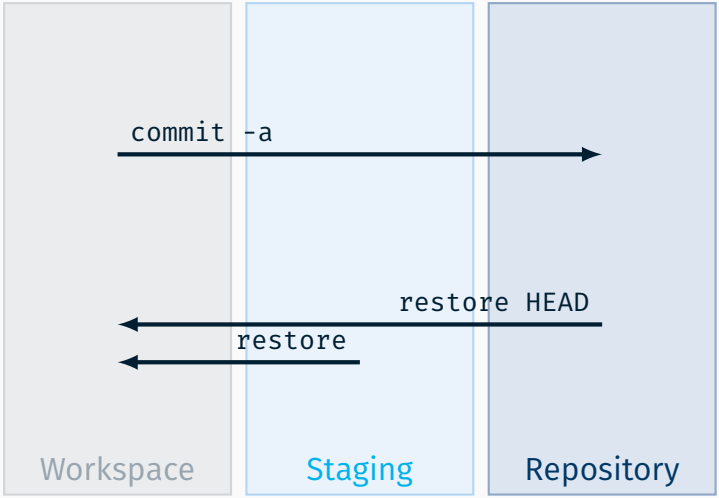
Overview: Checking files in and out in GIT



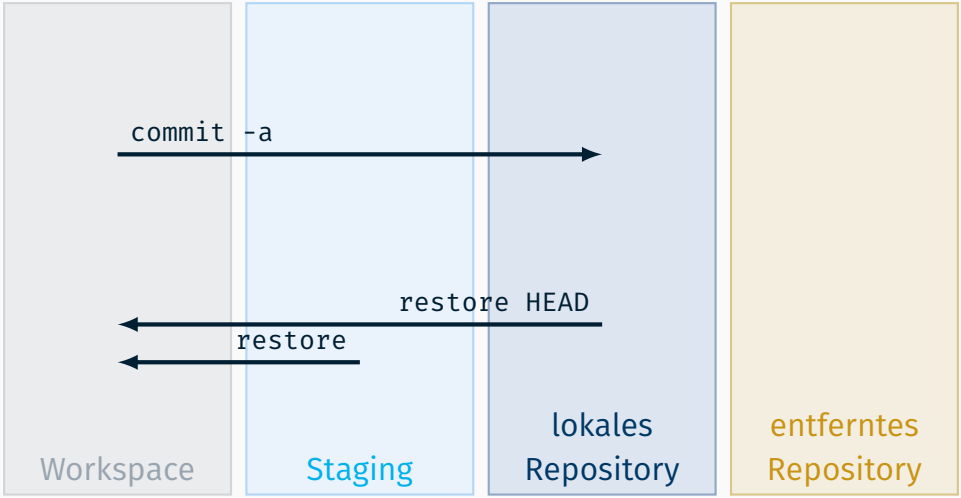
Overview: Checking files in and out in GIT



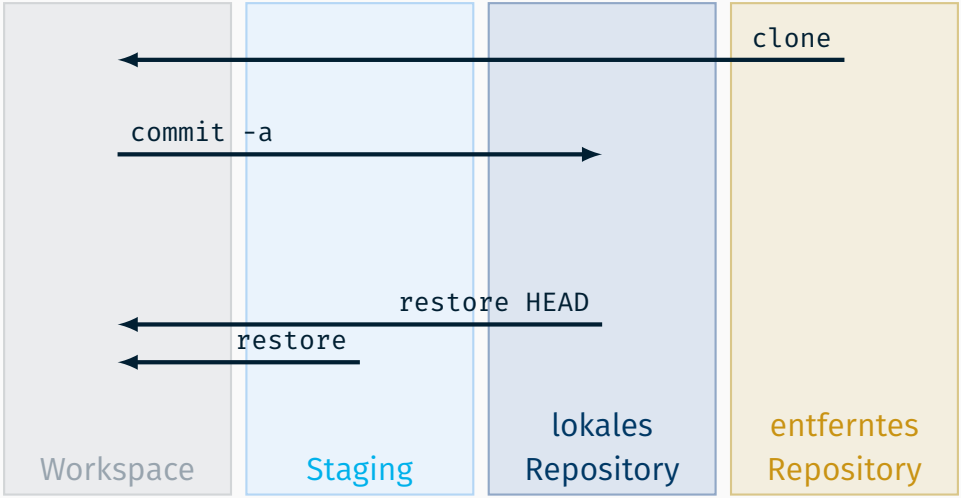
Overview: Checking files in and out in GIT



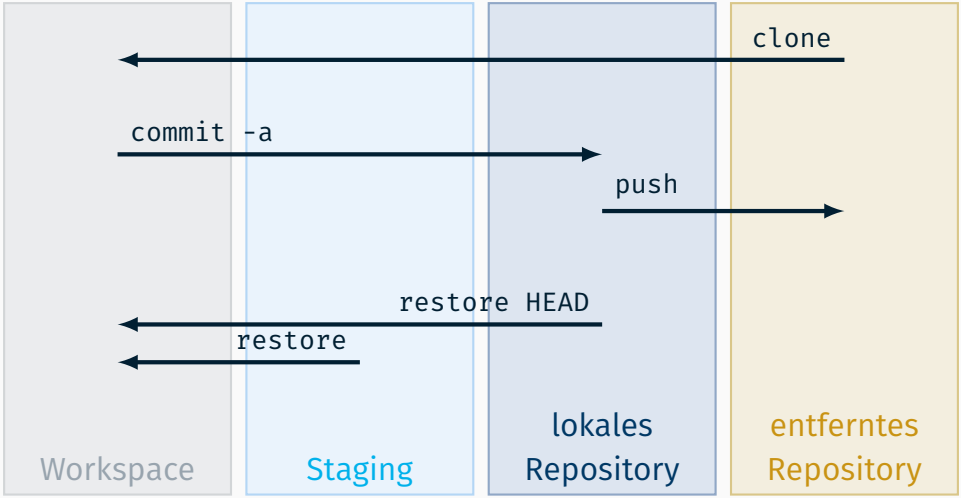
Overview: Checking files in and out in GIT



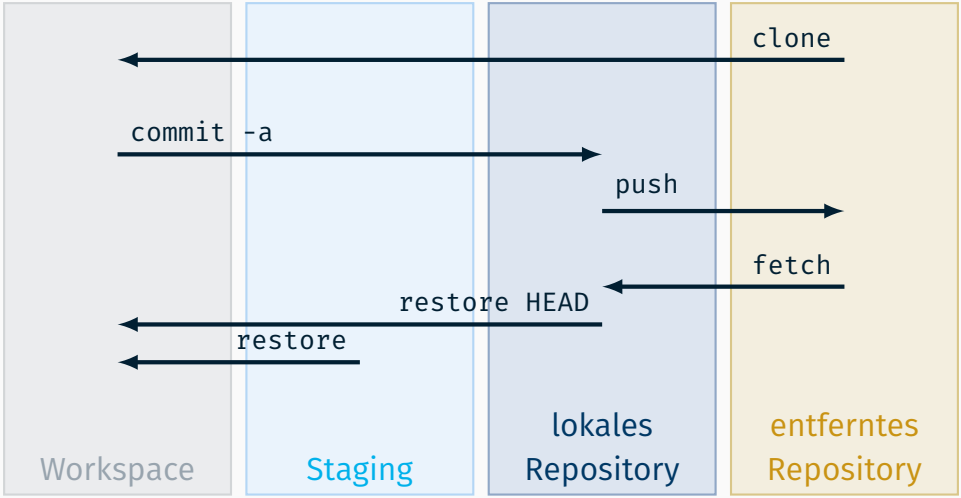
Overview: Checking files in and out in GIT



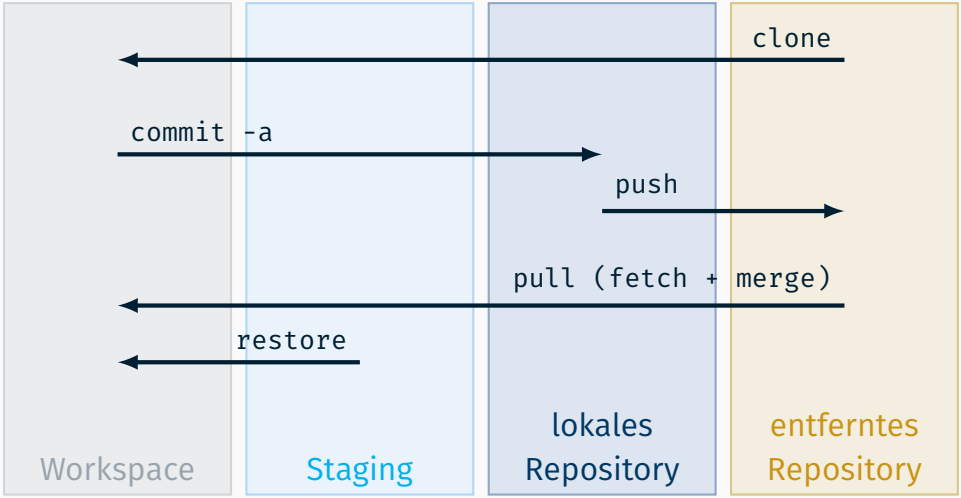
Overview: Checking files in and out in GIT



Overview: Checking files in and out in GIT



Overview: Checking files in and out in GIT



GIT Repository Hosting

- Many (commercial) providers, usually offering a web interface for repository management

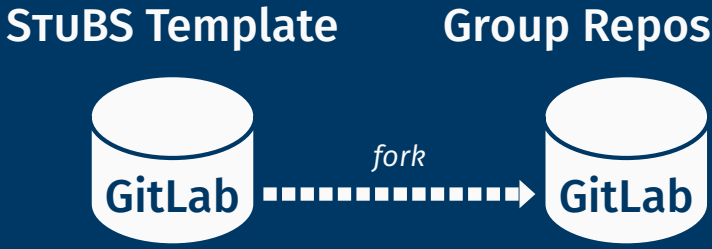
GitHub Atlassian Bitbucket

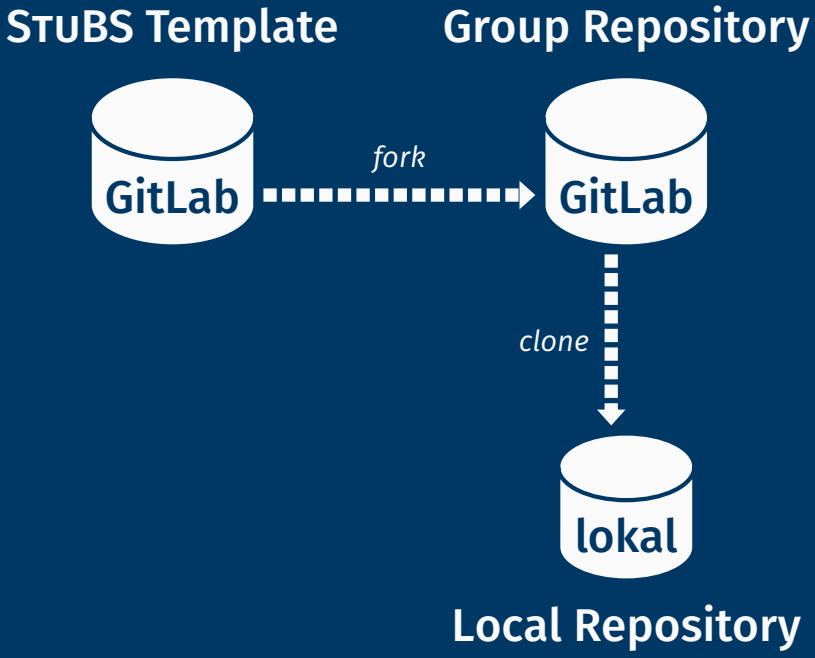
- RUB has its own GitLab installation at `gitlab.rub.de`
  - free private repositories
  - supports *Continuous Integration* (CI)
  - no registration necessary; uses your RUB credentials for login

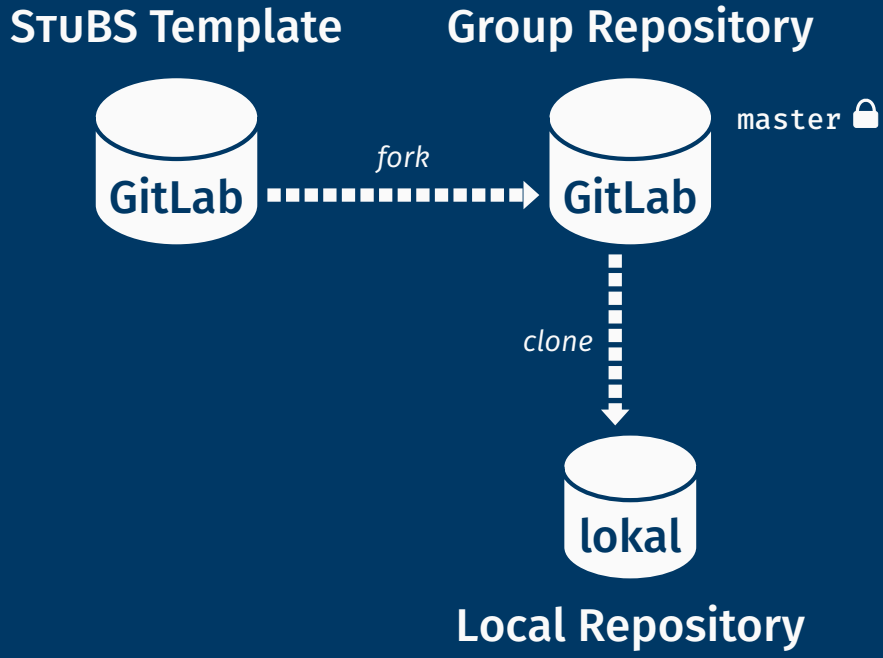
→ GitLab execrise repository will be created for you after you register your group

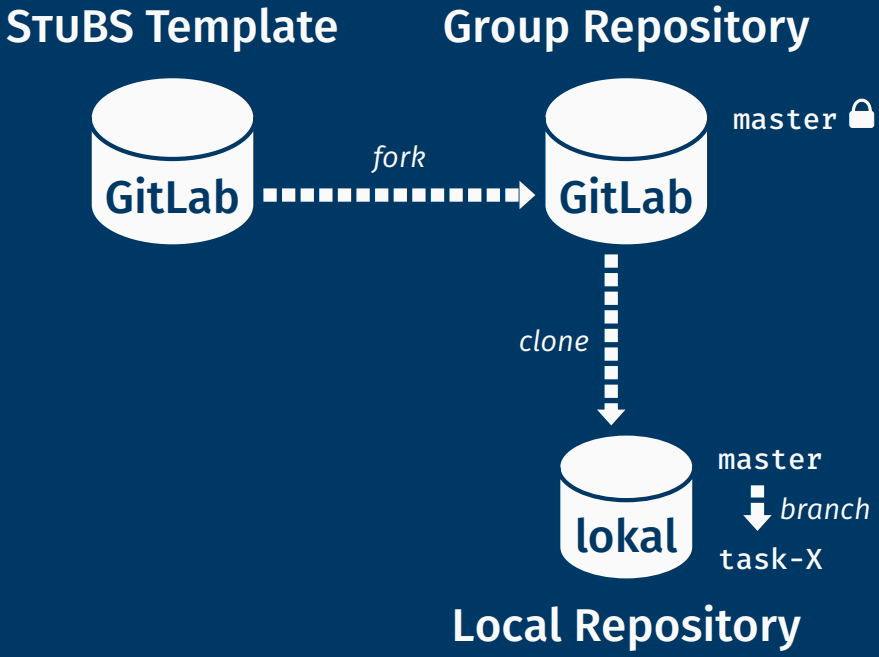
Workflow

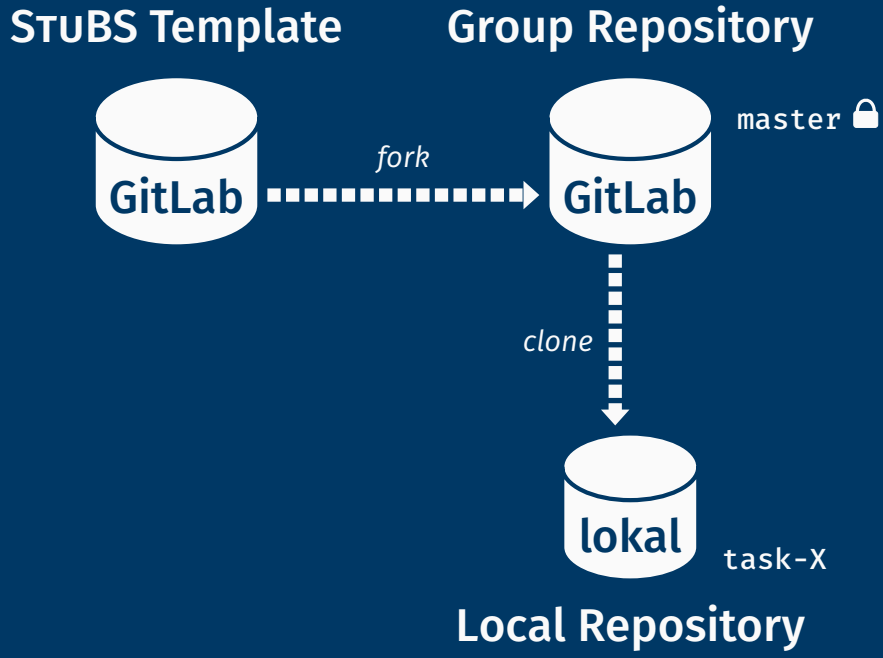
StuBS Template

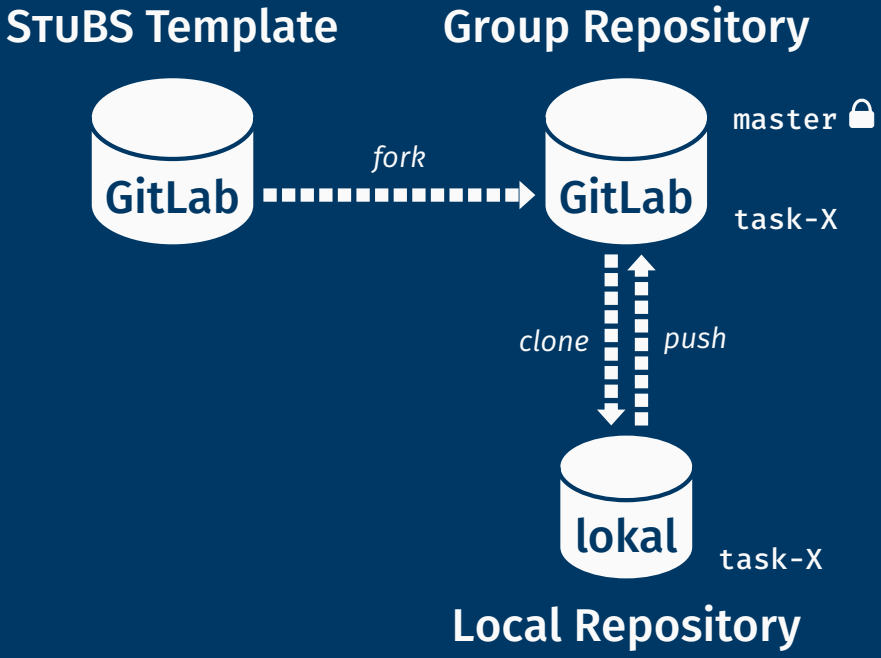

StuBS Template
Group Repository
GitLab
fork
GitLab

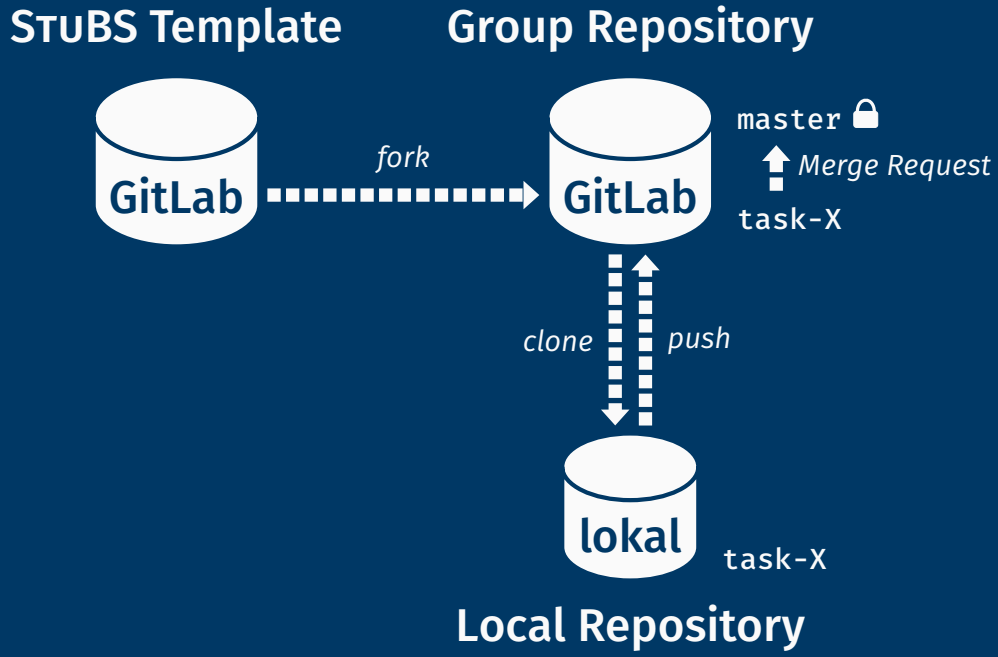

StuBS Template
Group Repository
GitLab
fork
GitLab
clone
lokal
Local Repository

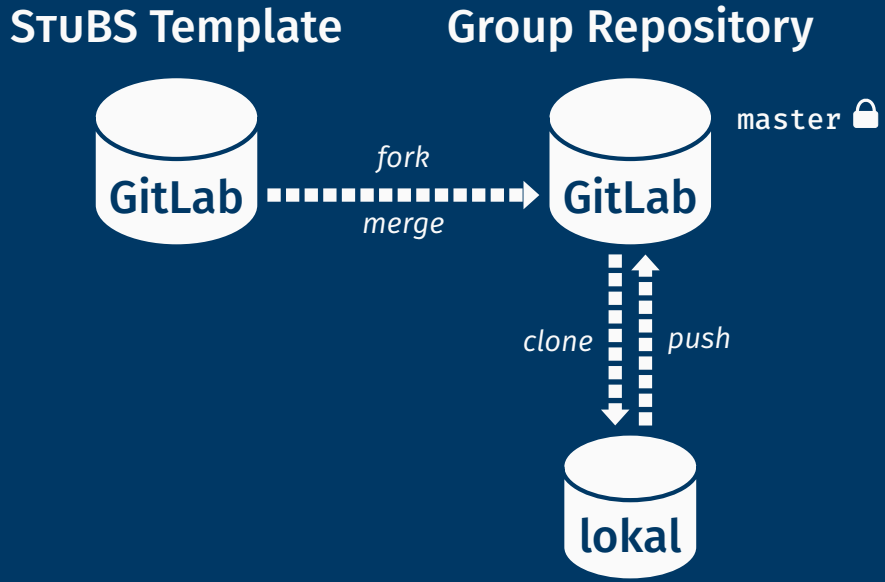

StuBS Template
Group Repository
GitLab
fork
GitLab
master
clone
lokal
Local Repository

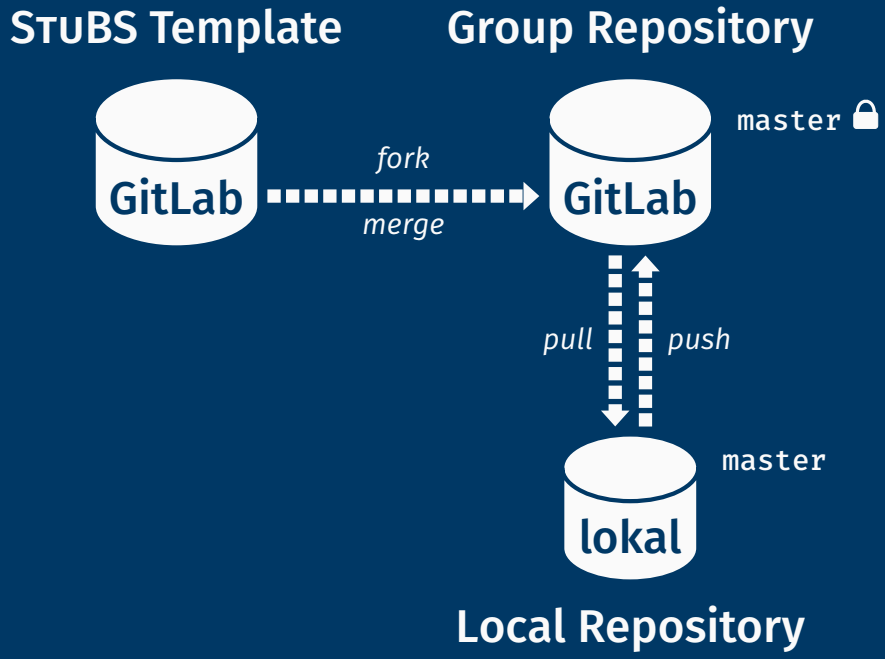

StuBS Template
Group Repository
GitLab
fork
GitLab
master
clone
lokal
master
branch
task-X
Local Repository

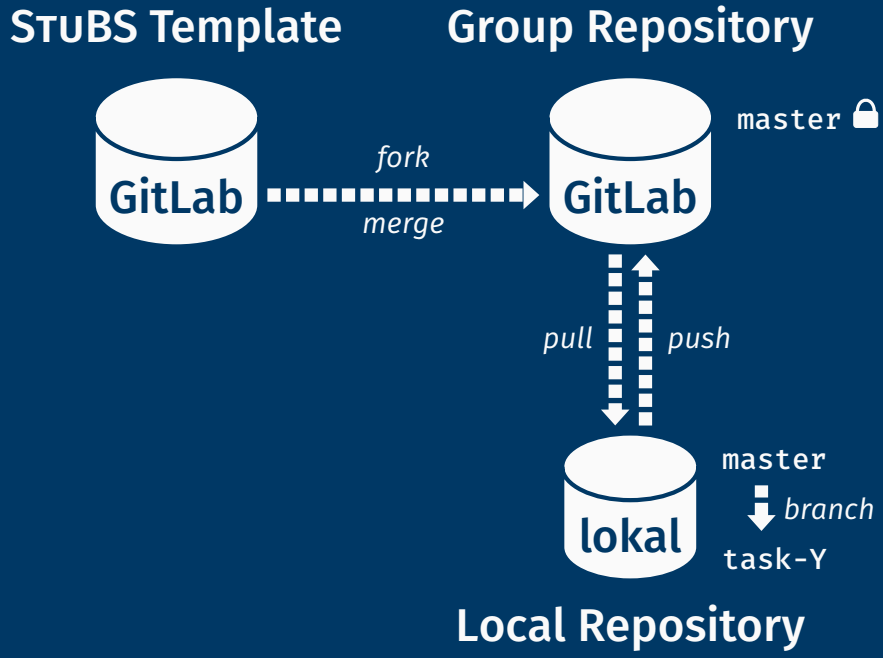

StuBS Template
Group Repository
GitLab
fork
GitLab
master
clone
lokal
task-X
Local Repository


StuBS Template
Group Repository
GitLab
fork
GitLab
master
task-X
clone
push
lokal
task-X
Local Repository


StuBS Template
Group Repository
GitLab
fork
GitLab
master
Merge Request
task-X
clone
push
lokal
task-X
Local Repository


StuBS Template
Group Repository
GitLab
fork
merge
GitLab
master
clone
push
lokal
Local Repository


StuBS Template
Group Repository
GitLab
fork
merge
GitLab
master
pull
push
master
lokal
Local Repository


STUBS Template
Group Repository
GitLab
fork
merge
GitLab
master
pull
push
lokal
master
branch
task-Y
Local Repository

Preparation

- Set name and mail address

```
$ git config --global user.email max.mustermann@fau.de
$ git config --global user.name "Max Mustermann"
```

- Customize GIT tools

```
$ git config --global core.editor nano
$ git config --global merge.tool meld
$ git config --list
```

- *Optional:* Define aliases

```
$ git config --global alias.word-diff=diff --word-diff=color -b
$ git word-diff
```

- *Optional:* Create an SSH key and register the public key in GitLab

Work with a remote GitLab repository

- Clone the remote GitLab exercise repository of group XX

```
$ git clone git@gitlab.ruhr-uni-bochum.de:boss/teaching/osci/submissions-ws2026/gXX.git
```

- Store new changes (*commits*) in GitLab

```
$ git push
```

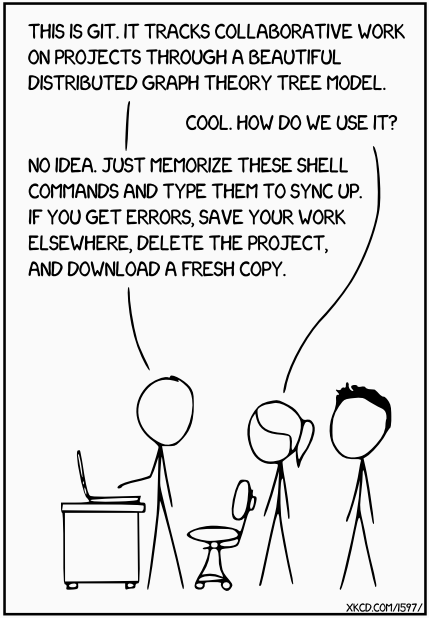
Possybly create branch `task-X` in GitLab

```
$ git push --set-upstream origin task-X
```

- Load changes from GitLab

```
$ git pull
```

Resolve *merge conflicts* using a mergetool (here: Meld)


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.
XKCD.COM/1597/

Cheat Sheet

git init	initialize a new repository in the current directory
git add *file*	mark file as candidate for next *commit*
git commit	create a new revision
git restore	reset a file to its commited state
git diff	show (uncommited) changes
git show	show changes made by (latest) commit
git status	display the list of files with uncommited changes
git branch	show and create branches
git switch	switch branches
git log	show the commit history
man git-*option*	view help, e.g. `man git-add`

Cheat Sheet (remote sources)

git clone *URL*	create a local copy of a remote repository
git fetch *name*	receive commits from a remote repository
git pull *name*	shorthand for git fetch + git merge
git push *name*	tansmit commits to a remote repository