

Git & L^AT_EX-Einführung

Noah Kälin¹, Simon Walker¹, Naoki Pross¹

¹Hochschule für Technik Rapperswil

30. Oktober 2019

Inhaltsverzeichnis

Warum Git
Warum
Vorteile

Git lernen

Git benutzen

Was möchten wir?

Das Problem

Dateien zwischen Machine synchronisieren und sie mit mehr Leute bearbeiten. Spezifischer Textdateien bearbeiten (z.B. Quellcode)

Was möchten wir?

Das Problem

Dateien zwischen Machine synchronisieren und sie mit mehr Leute bearbeiten. Spezifischer Textdateien bearbeiten (z.B. Quellcode)

Eine Lösung



(Engl. für Blödmann) Entwickelt von Linus Torvalds

Was ist git?

Was wir brauchen

- ▶ Synchronisation
- ▶ Team Datei Bearbeitung
- ▶ Problemlose Offline-Nutzung

Was ist git?

Was wir brauchen

- ▶ Synchronisation
- ▶ Team Datei Bearbeitung
- ▶ Problemlose Offline-Nutzung

Was kriegen wir extra

- ▶ Geschichte jedes Dokuments im Projekt
- ▶ Kryptographische Sicherheit der Projektgeschichte
- ▶ Gemeinsamer Dateizugriff ohne zentraler Server

Bemerkung

Git löst ein komplex Problem
daher ist Git auch komplex
But don't worry it's not *too* hard

Inhaltsverzeichnis

Warum Git

Git lernen

Repository

Commit

Remote

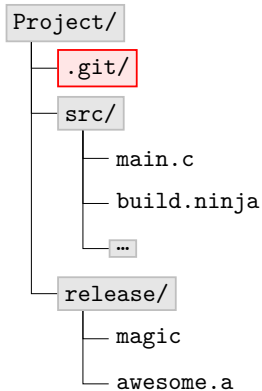
Merge

Branch

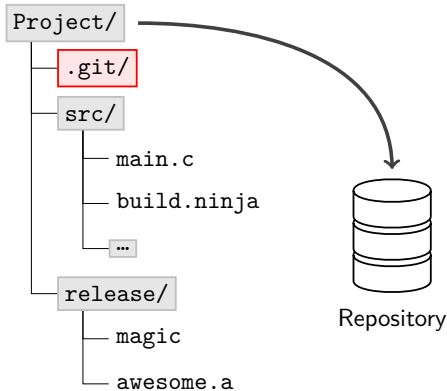
Fork

Git benutzen

Begriff: Repository



Begriff: Repository



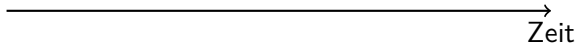
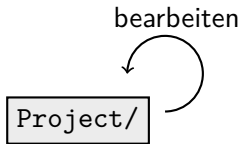
Für Git ein *Repo* ist eine Verzeichnis mit einer spezieller (unsichtbar) Unterverzeichnis **.git**

Man soll **nie** .git löschen.

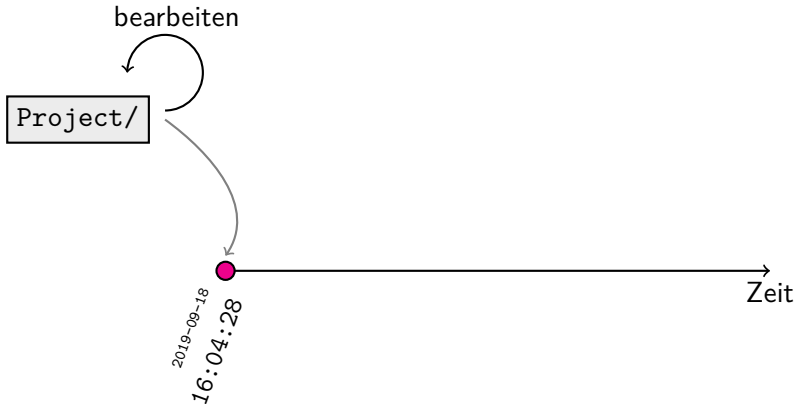
Snapshots

Project/

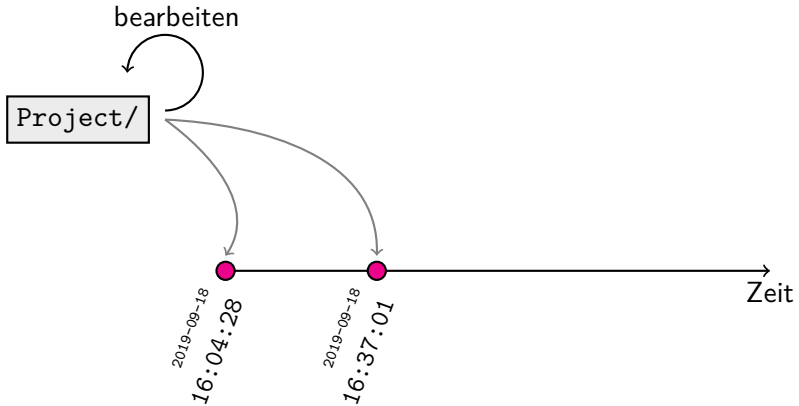
Snapshots



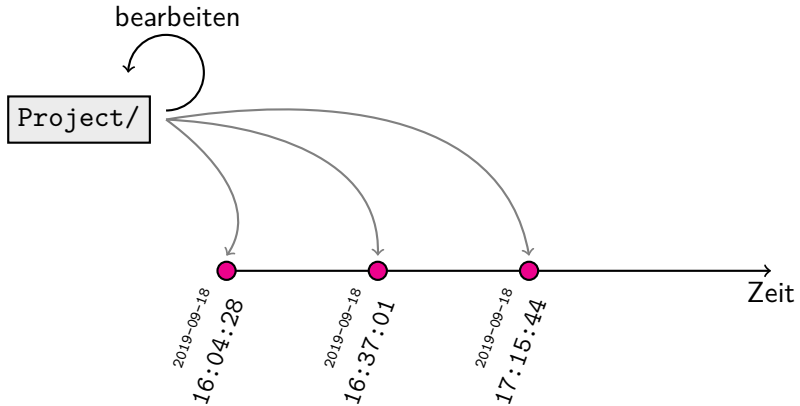
Snapshots



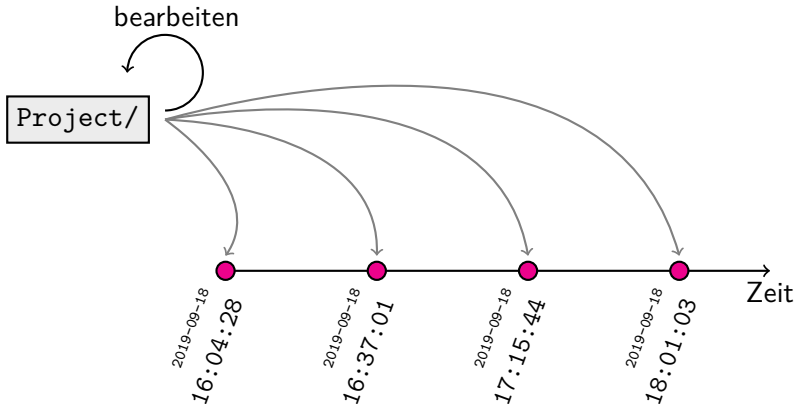
Snapshots



Snapshots



Snapshots



Begriff: Commit

Ein Commit enthält

- ▶ Die Änderungen der Dateien **“snapshot”**

Begriff: Commit

Ein Commit enthält

- ▶ Die Änderungen der Dateien **“snapshot”**
- ▶ Autor Name + Email

Begriff: Commit

Ein Commit enthält

- ▶ Die Änderungen der Dateien **“snapshot”**
- ▶ Autor Name + Email
- ▶ Zeitstempel

Begriff: Commit

Ein Commit enthält

- ▶ Die Änderungen der Dateien **“snapshot”**
- ▶ Autor Name + Email
- ▶ Zeitstempel
- ▶ **Eine Beschreibung der Änderungen**

Begriff: Commit

Ein Commit enthält

- ▶ Die Änderungen der Dateien **“snapshot”**
- ▶ Autor Name + Email
- ▶ Zeitstempel
- ▶ **Eine Beschreibung der Änderungen**
- ▶ kryptographisches Hash der Dateien

Begriff: Commit

Ein Commit enthält

- ▶ Die Änderungen der Dateien **“snapshot”**
- ▶ Autor Name + Email
- ▶ Zeitstempel
- ▶ **Eine Beschreibung der Änderungen**
- ▶ kryptographisches Hash der Dateien

Commits können nicht ändert werden,
weil sie **“Geschichte”** des Projekts sind.

Commit Beispiel

```
commit e584e04c5f8ffb14e50c701c1fd8178457a51743
Author: Nao Pross <naopross@thearcway.org>
Date:   Tue Jan 22 04:31:30 2019 +0100
```

Add test for task and job, fix bug in job

By being a `std::set` job did not allow to add duplicate elements, changing it to a `std::multiset` fixes the issue.

Aber was ist ein Commit?

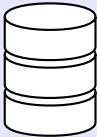
Commit = “Logical Unit of Work”

Ein Commit kann auch sehr klein sein.
Generell kann man sagen:

je mehr Commits, desto besser

Begriff: Remote

Laptop
(Sie)

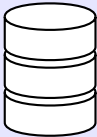


Server
(z.B. Github)



Begriff: Remote

Laptop
(Sie)



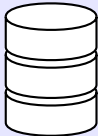
Server
(z.B. Github)



Ein *Remote* ist ein Clone (Kopie) des Repos auf eine andere Maschine

Begriff: Remote

Laptop
(Sie)



origin

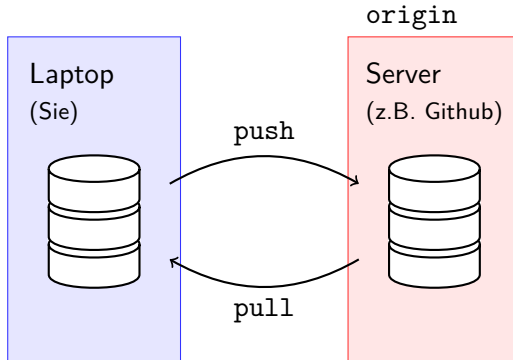
Server
(z.B. Github)



Ein *Remote* ist ein Clone (Kopie) des Repos auf eine andere Maschine

Remote können ein name haben, z.B. `origin`

Begriff: Remote



Ein *Remote* ist ein Clone (Kopie) des Repos auf eine andere Maschine

Remote können ein name haben, z.B. origin

Zu einem Remote synchronisieren

origin

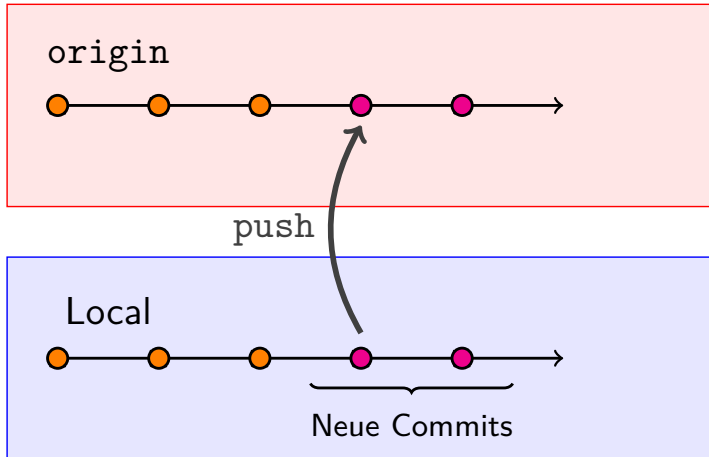


Local

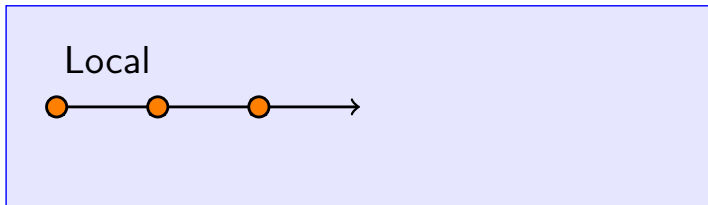
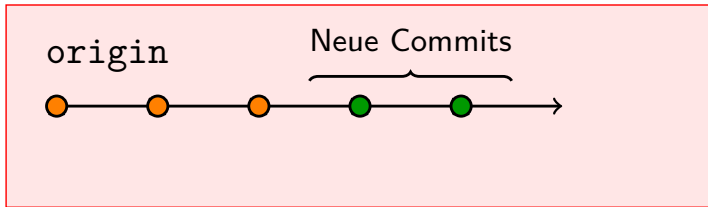


Neue Commits

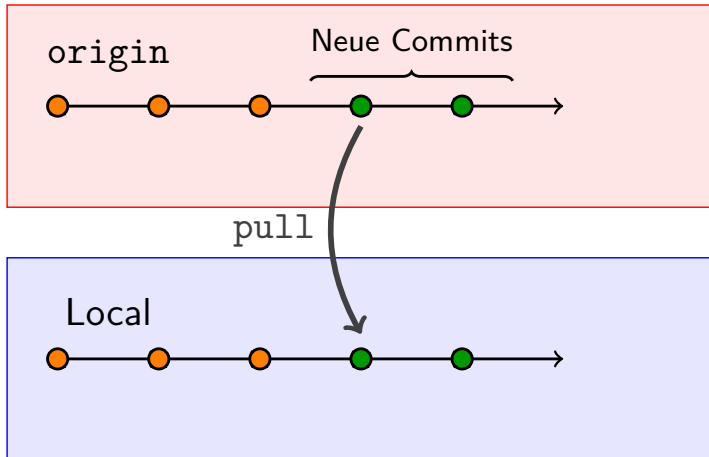
Zu einem Remote synchronisieren



Von einem Remote synchronisieren



Von einem Remote synchronisieren



Merge

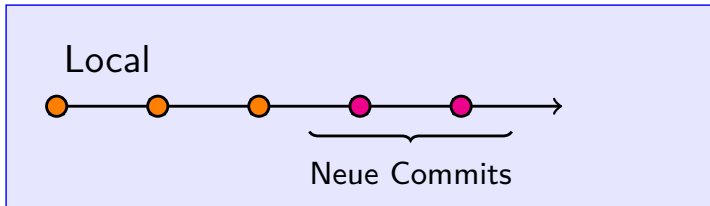
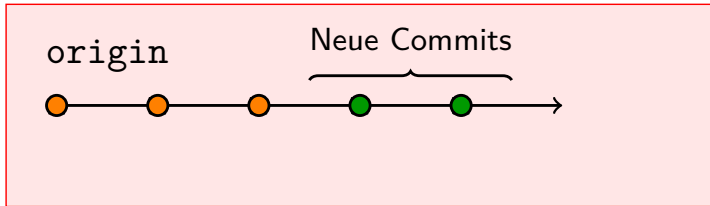
origin



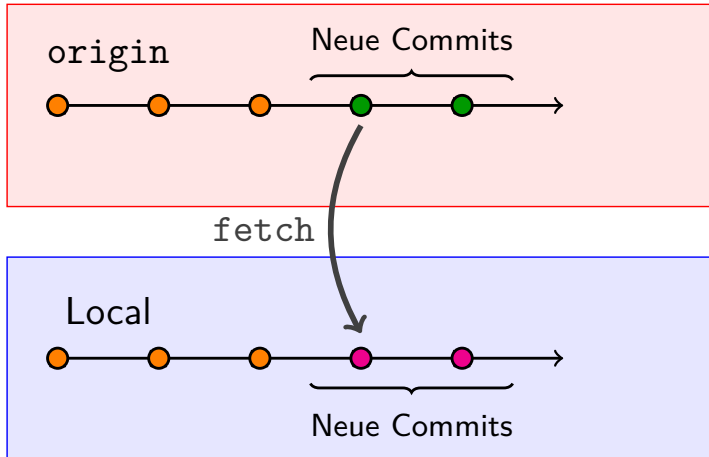
Local



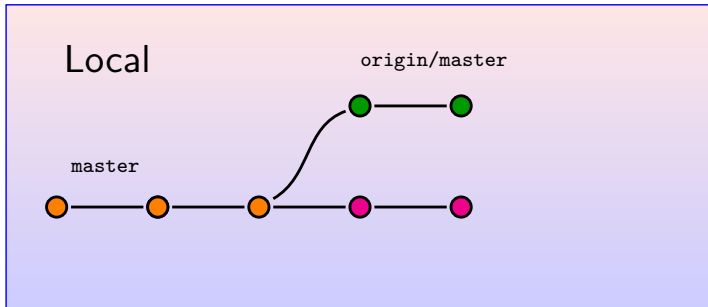
Merge



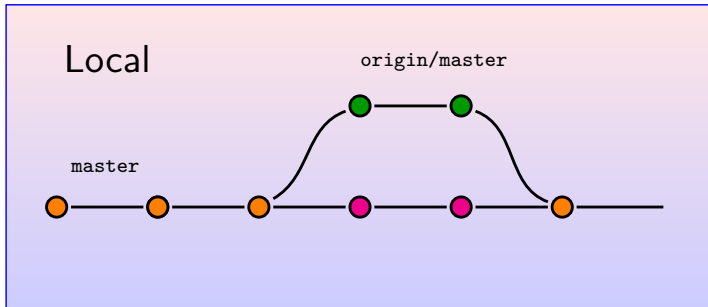
Merge



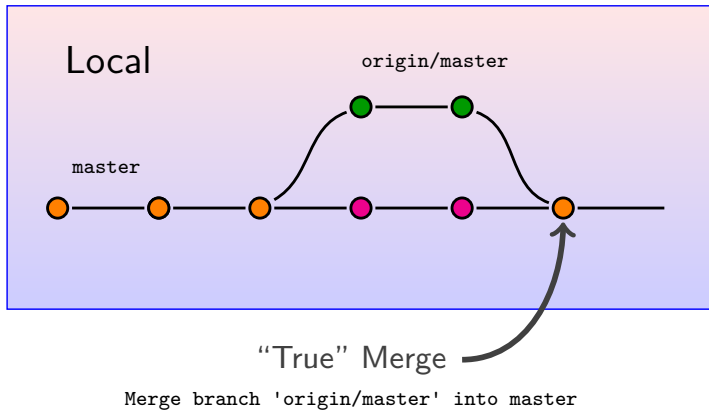
Begriff: 3-Way Merge



Begriff: 3-Way Merge



Begriff: 3-Way Merge



Konflikte

D.h. Was passiert wenn 2 Leute die gleiche Linie in einem Dokument bearbeiten?

Konflikte

D.h. Was passiert wenn 2 Leute die gleiche Linie in einem Dokument bearbeiten?

Git kann nicht wissen welche version besser ist, und so fragt dir.

```
...
Here is a line that nobody touched
<<<<<<< HEAD
I have edited this line
=====
Someone else has also edited this line
>>>>>>> origin/master
...
```


Warum “3-Way”?

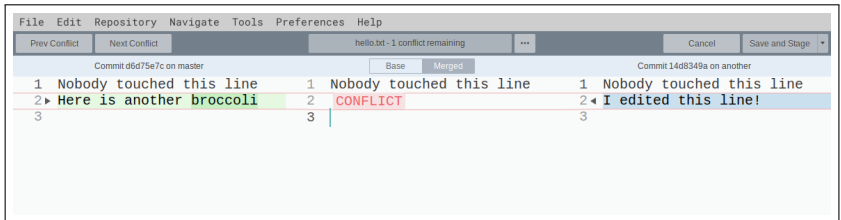


Abbildung: Sublime Merge

Warum “3-Way”?

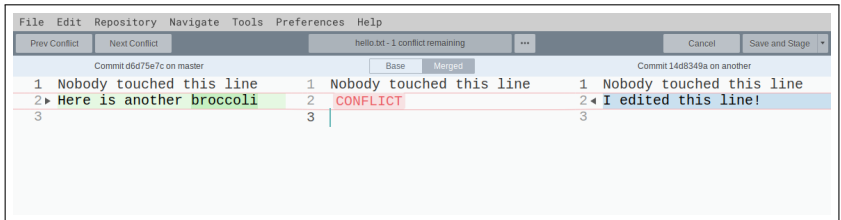
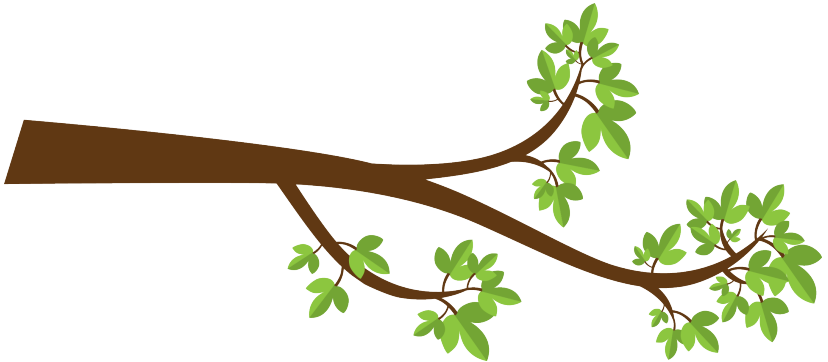


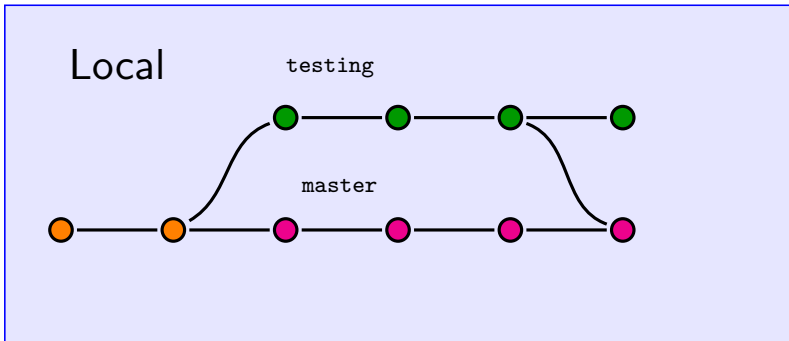
Abbildung: Sublime Merge

pull ist ein Alias für fetch + merge

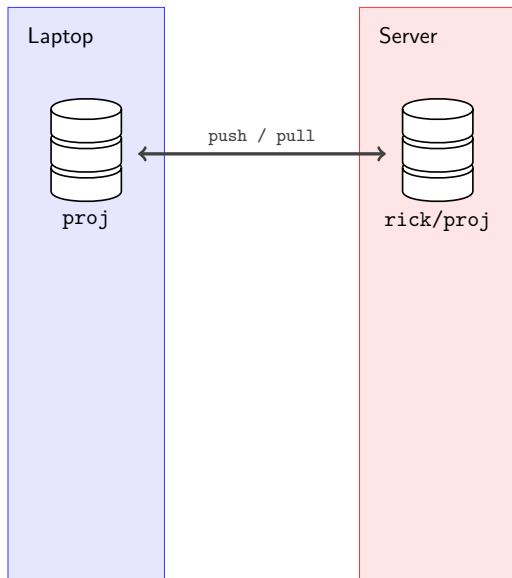
Branch



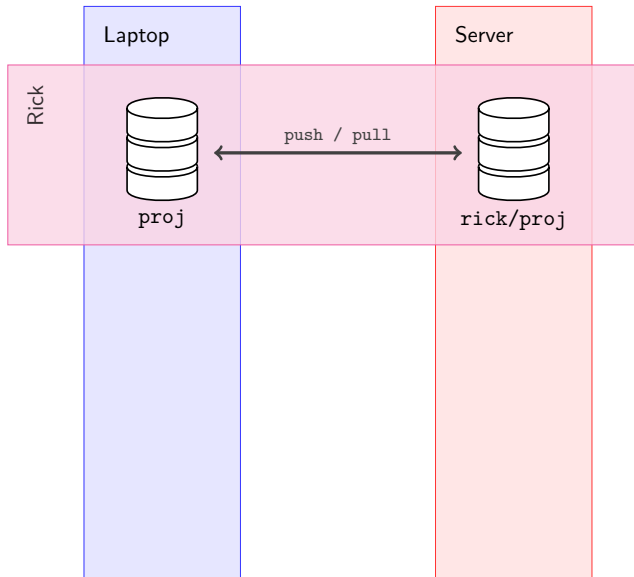
Begriff: Branch



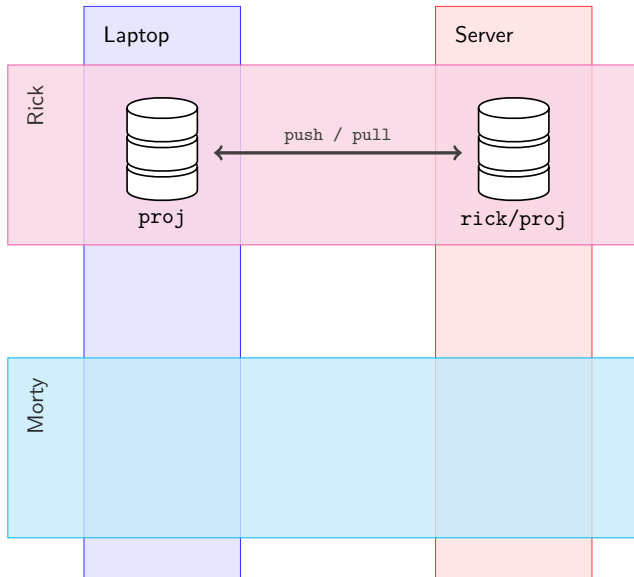
Begriff: Fork



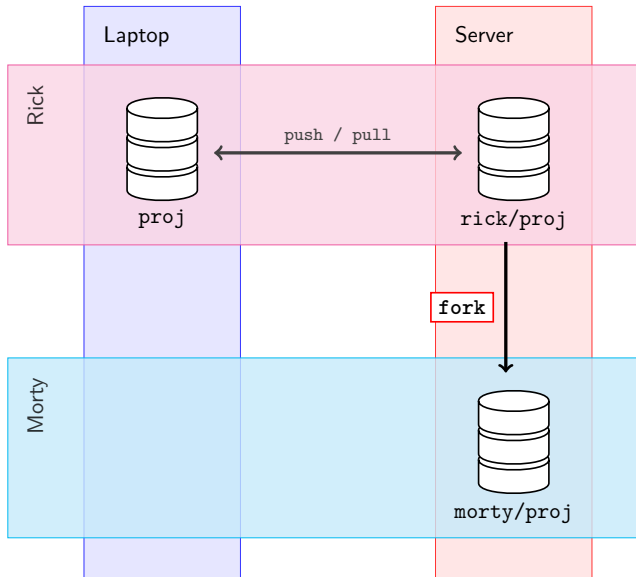
Begriff: Fork



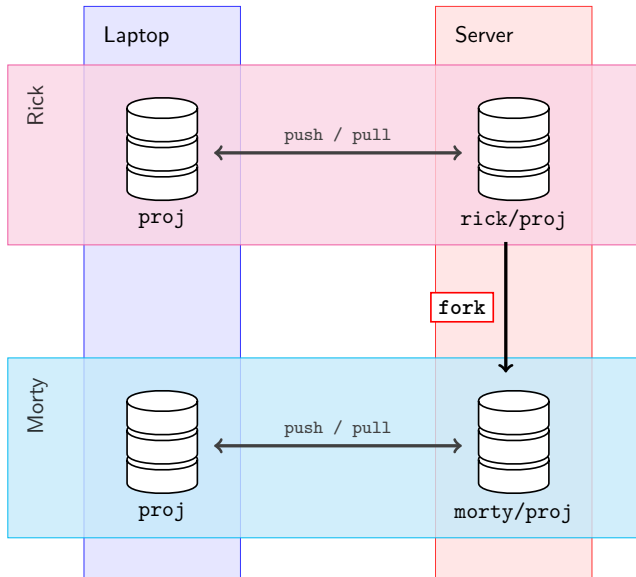
Begriff: Fork



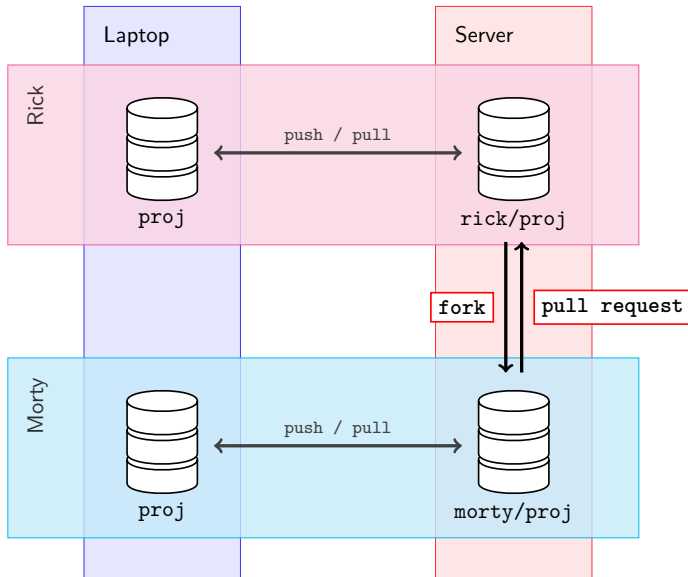
Begriff: Fork



Begriff: Fork



Begriff: Fork



Inhaltsverzeichnis

Warum Git

Git lernen

Git benutzen

Tips