

Let's Git – Versionsverwaltung und Open Source

open_source

beitragen

Motivation

Open Source

1. Issue

Organisation

1. Pull Request

Issues

Pull Request

Caterina Mandel

Sandro Speh

Der Kurs

- Ihr sollt lernen, wie man **Git als Versionsverwaltungstool** nutzt und wie Ihr **erste Schritte** beim Entwickeln von **Open Source Projekten** macht.
- **Keine** Programmier-Vorkenntnisse benötigt, Grundkenntnisse Englisch
- Kurs ist an Lernende gerichtet, die
 - das Versionsverwaltungstool **Git** lernen wollen
 - verstehen wollen, wie **Open Source Software** entwickelt wird
 - bei Open Source Projekten **beitragen** möchten

Was ist ein **VCS**?

fetch

clone

Wie kann ich etwas **speichern**?

init

commit

Was ist ein **Repository**?

Was ist **Git**?

pull

push

Wie komme ich auf den **neuesten Stand**?

checkout

Was ist ein **Branch**?

log

Wie kann ich etwas **rückgängig** machen?

reset

Was ist ein **merge Konflikt**?

Was ist das Git **Datenmodell**?

merge

push

Was ist **Open Source**?

Free Software

Github

Was ist Semantic Versioning?

Git Flow

Was ist ein **Pull Request**?

Feature

Was ist ein **Issue**?

Wie **organisiert** man sich in Open Source Projekten?

Wieso sollte ich **beitragen**?

Best Practices

Wie finde ich mein **erstes Issue**?

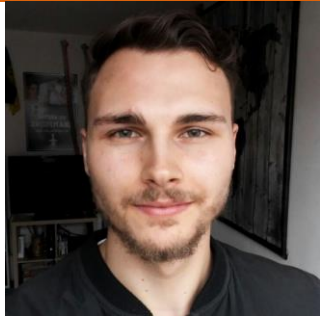
Continuous Integration

Wie stelle ich einen **Pull Request**?

Wo kann ich **anfangen**?

Was für **Tools** erleichtern mir die Arbeit?

Teaching Team



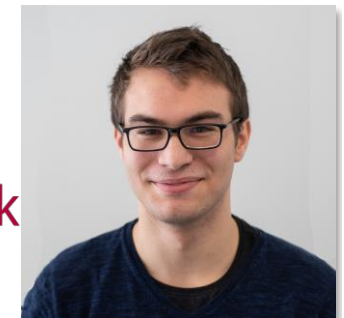
Sandro Speh



Caterina Mandel



Til Schniese



Leo Wendt

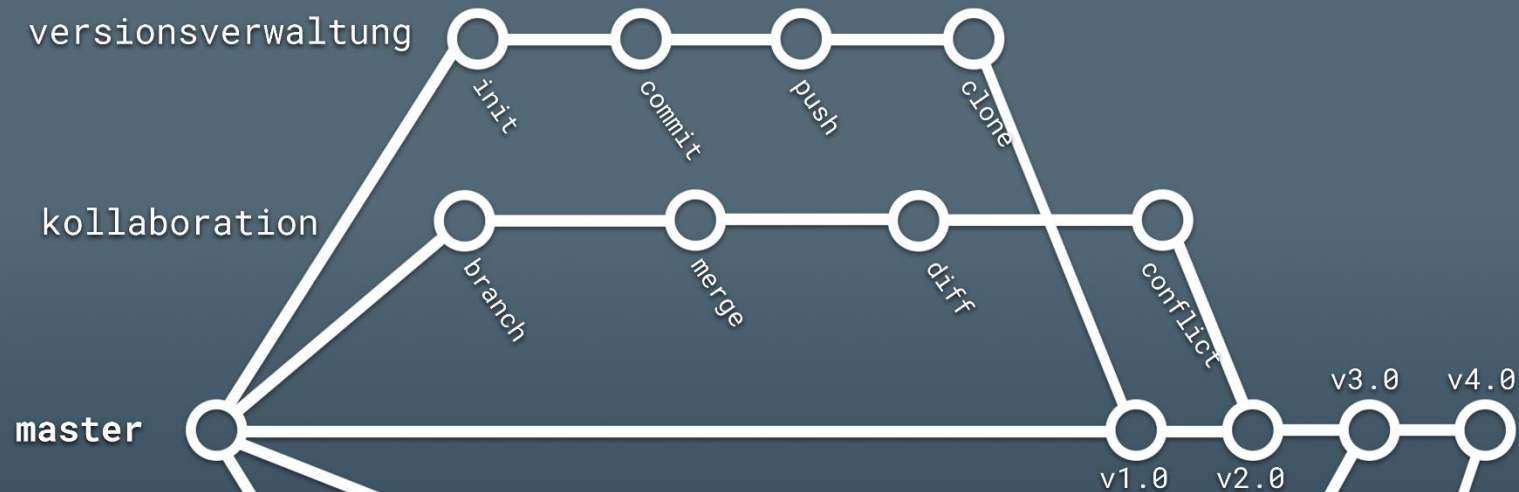
Competitive-Programming
DevOps
Modellierung
Front-End
Scripting
Projektmanagement
Mobile-Development
Softwaretechnik
Design-Thinking
Computer-Vision
Datenbanksysteme
Prozessanalyse
Open-Source
ITSE
Web-Technologien
Back-End
Theoretische-Informatik
Automatisierung
Git



Marc Rosenau



Udo Pigorsch



Let's Git – Versionsverwaltung und Open Source

open_source

beitragen

Motivation

Open Source

1. Issue

Organisation

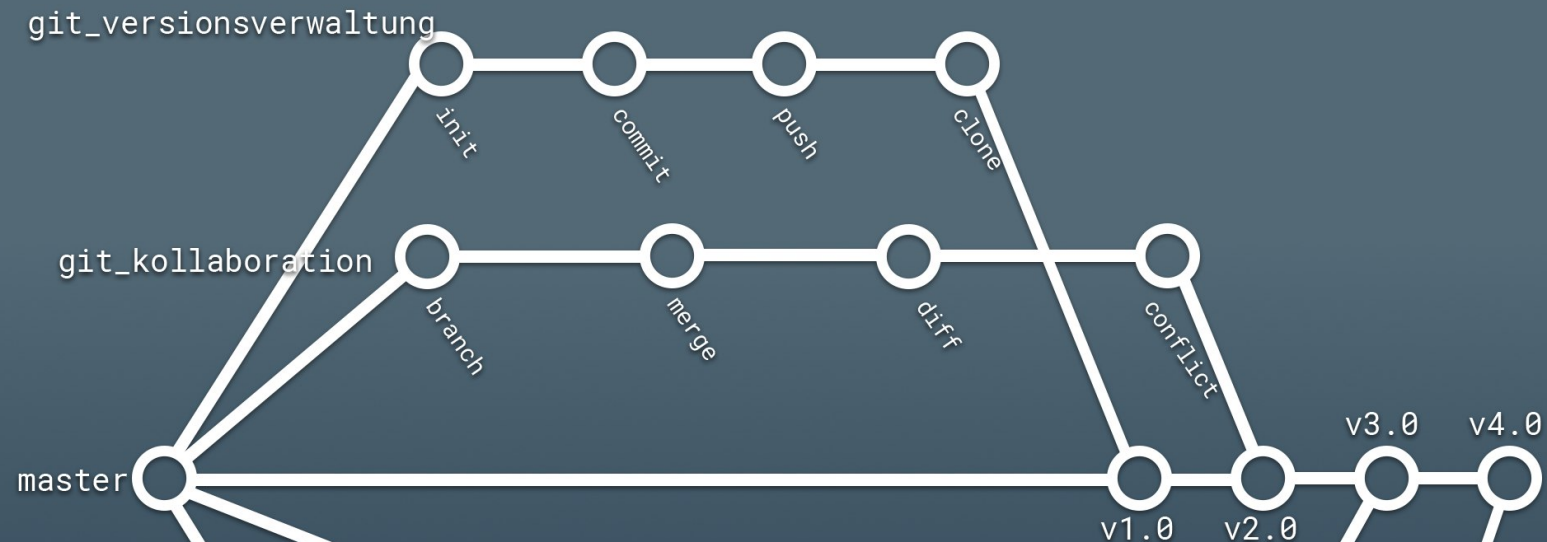
1. Pull Request

Issues

Pull Request

Caterina Mandel

Sandro Speth



Versionskontrolle

beitragen

Motivation

1. Issue

1. Issue

1. Pull Request

1. Issue

1. Pull Request

Tools

CI

Caterina Mandel
Udo Pigorsch

Begriffserklärung

Was ist eine Datei?

- z.B. Text (.txt), Bilder (.jpg), Musik (.mp3)

Begriffserklärung

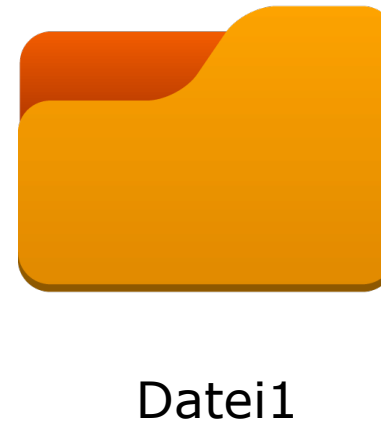
Was ist eine Datei?

- z.B. Text (.txt), Bilder (.jpg), Musik (.mp3)

Was ist eine Version?

- beschreibt den Inhalt einer Datei zu bestimmten Zeitpunkt
- jede Datei besitzt mindestens eine Version
- Änderungen der Datei erzeugen eine neue Version

Begriffserklärung



Begriffserklärung

Was ist eine Datei?

- z.B. Text (.txt), Bilder (.jpg), Musik (.mp3)

Was ist eine Version?

- beschreibt den Inhalt einer Datei zu bestimmten Zeitpunkt
- jede Datei besitzt mindestens eine Version
- Änderungen der Datei erzeugen eine neue Version

Was ist Versionskontrolle?

- Verwaltung verschiedener Versionen von Dateien

Begriffserklärung

Was ist eine Datei?

- z.B. Text (.txt), Bilder (.jpg), Musik (.mp3)

Was ist eine Version?

- beschreibt den Inhalt einer Datei zu bestimmten Zeitpunkt
- jede Datei besitzt mindestens eine Version
- Änderungen der Datei erzeugen eine neue Version

Was ist Versionskontrolle?

- Verwaltung verschiedener Versionen von Dateien

Warum Versionskontrolle interessant ist?

- ermöglicht digitale Zeitreisen durch die Versionen unserer Dateien

Versionskontrolle



Versionskontrolle



- Datei



- DateiKopie1Geschnitten



- DateiKopie2FarbeBearbeitet



- DateiKopie3Backup



Formen der Versionskontrolle

Lokale Versionskontrolle

- Version Control System - VCS

Zentralisierte Versionskontrolle

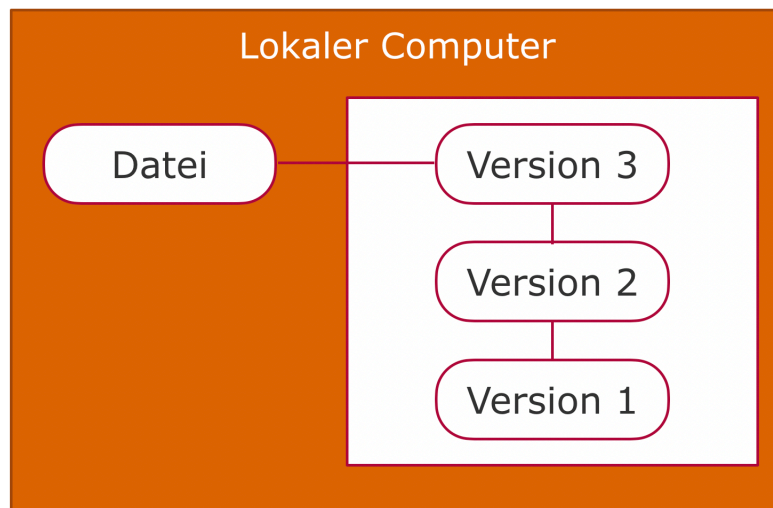
- Centralised Version Control System - CVCS

Verteilte Versionskontrolle

- Distributed Version Control System - DVCS

Formen der Versionskontrolle:

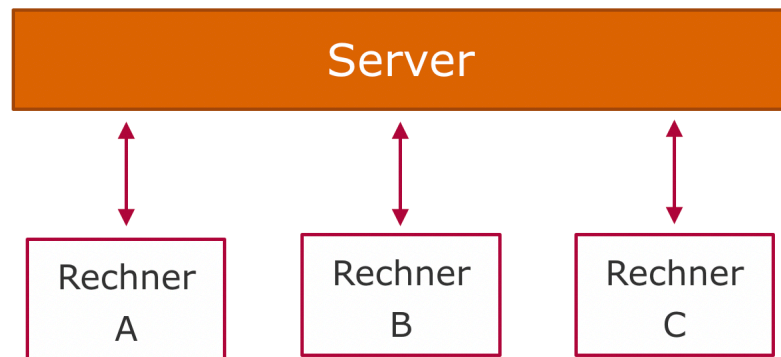
Lokale Versionskontrolle



- Unterschiede zwischen Versionen (Patch-Sets) werden gespeichert
- Patch-Sets können kombiniert werden, um die Datei eines Zeitpunktes wiederherzustellen

Formen der Versionskontrolle:

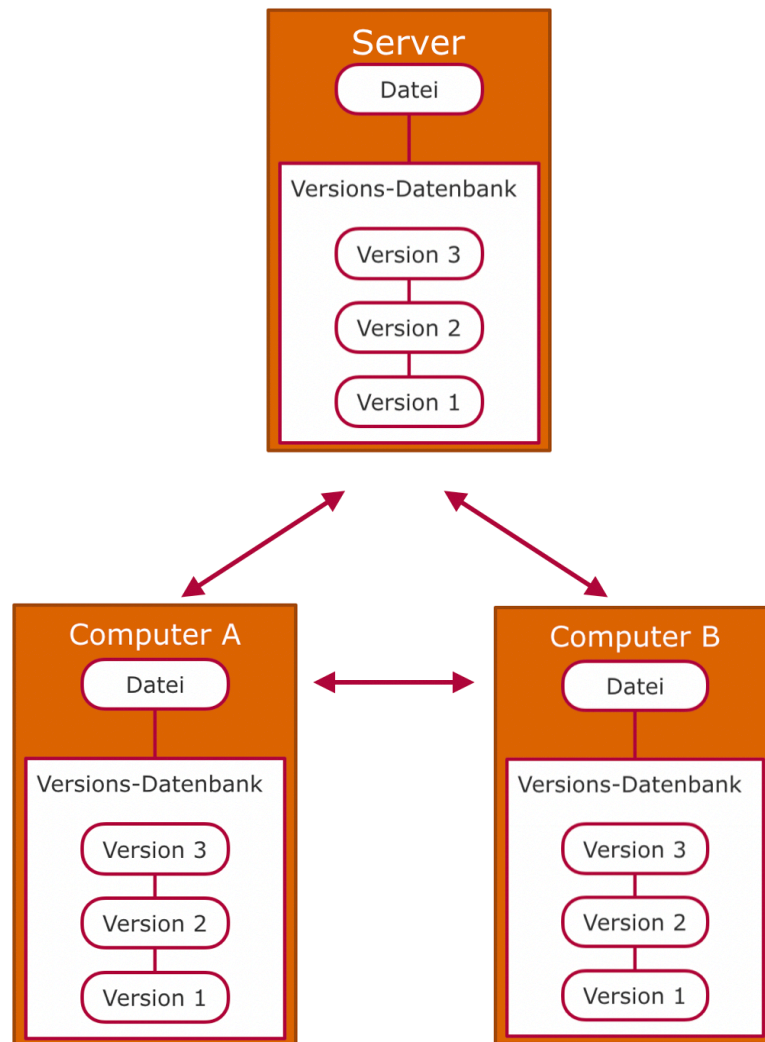
Zentralisierte Versionskontrolle



- Ein Server enthält alle versionierten Dateien, Klienten laden diese
- Verbessert die Verwaltung und ermöglicht Kontrolle „wer darf was“
- Single Point of Failure - Ausfall des Servers bedroht die Arbeit aller

Formen der Versionskontrolle:

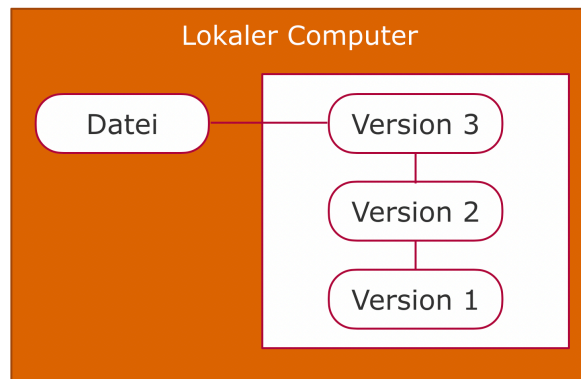
Verteilte Versionskontrolle



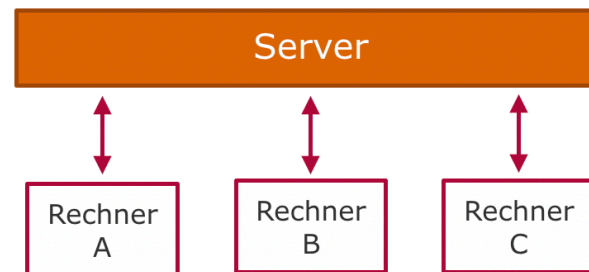
- Klienten spiegeln das Repository mit seiner vollständigen Historie
- Kein Single Point of Failure

Formen der Versionskontrolle

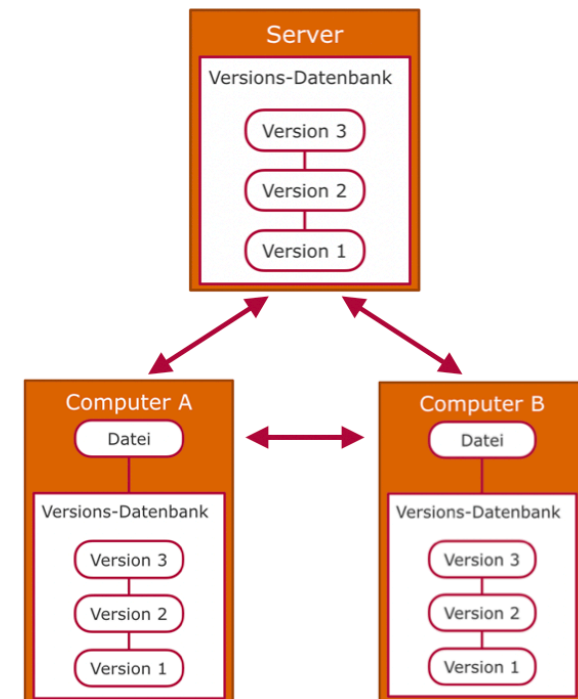
Lokale Versionskontrolle

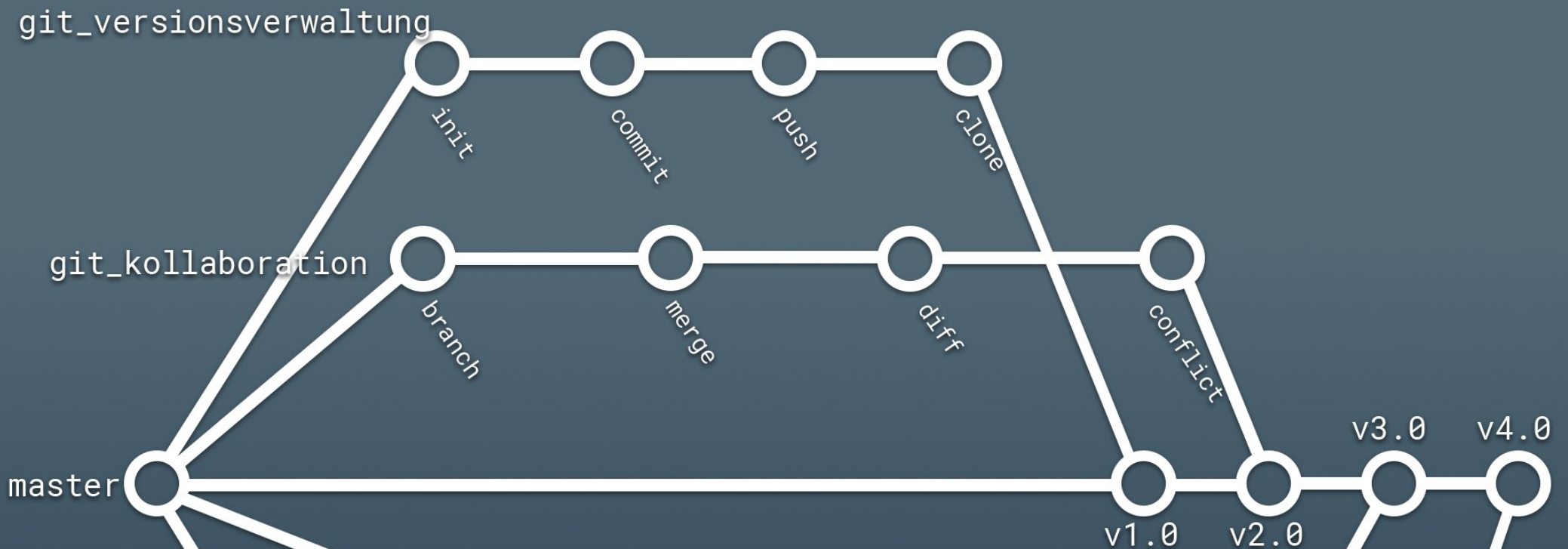


Zentralisierte Versionskontrolle



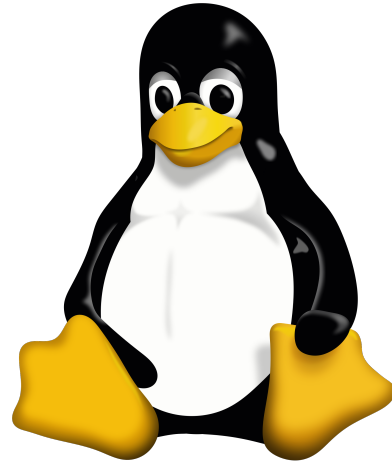
Verteilte Versionskontrolle

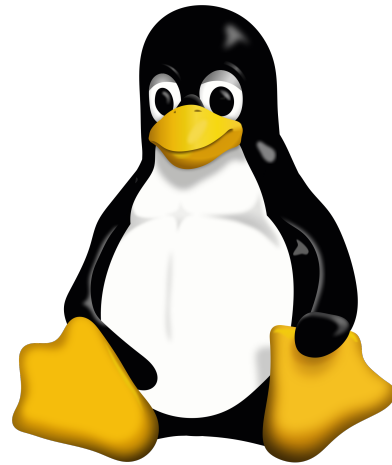




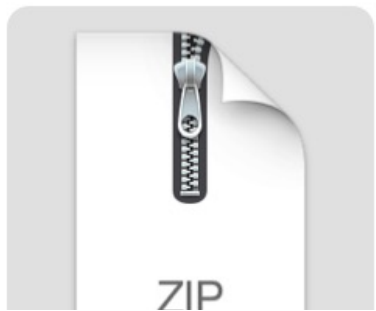
Entstehungsgeschichte von Git

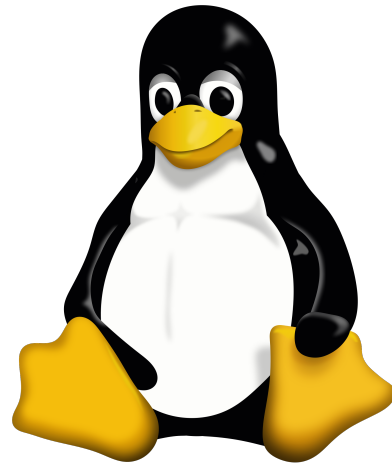






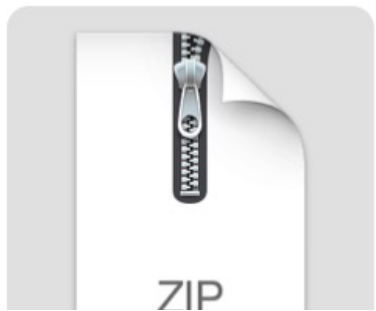
1991



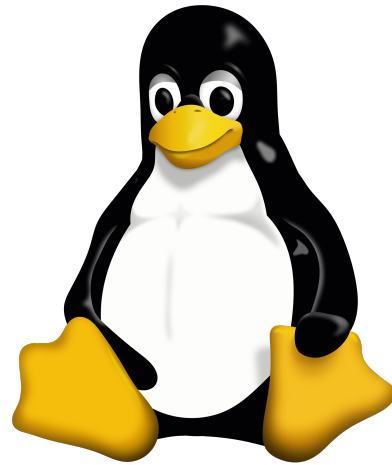


1991

2002



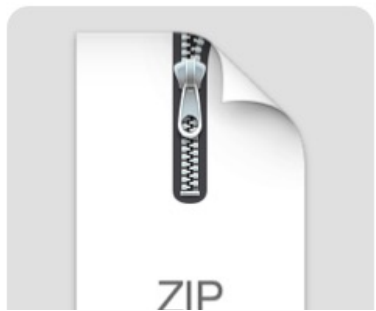
DVCS
BitKeeper



1991

2002

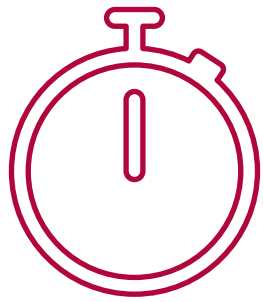
2005

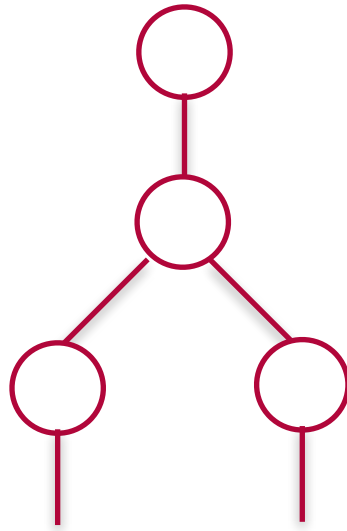
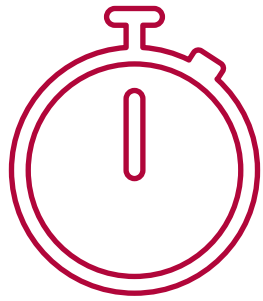


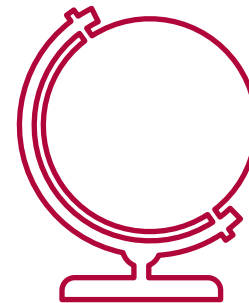
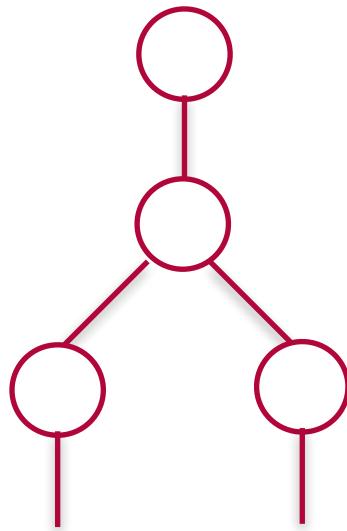
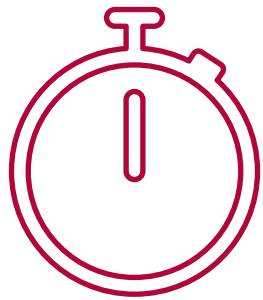
DVCS
BitKeeper

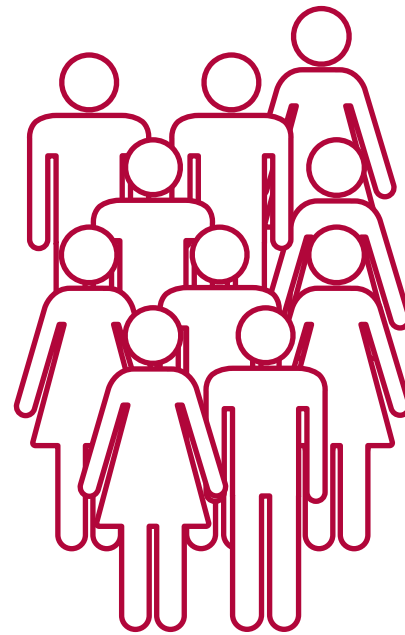
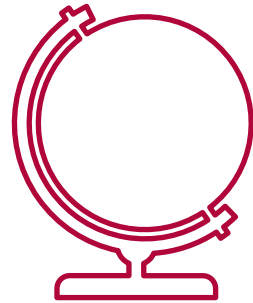
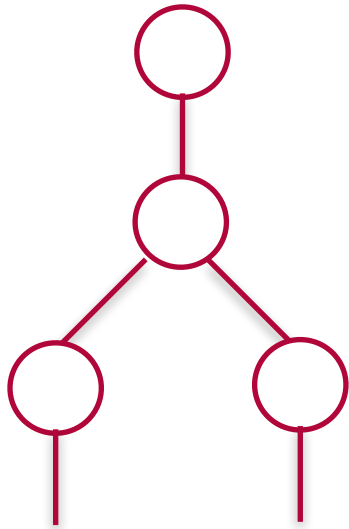
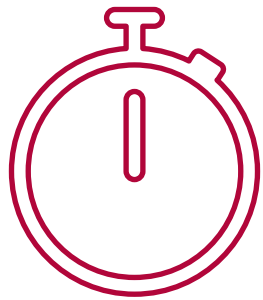


git









git_versionsverwaltung

init

commit

push

clone

git_kollaboration

branch

merge

diff

conflict

master

v1.0

v2.0

v3.0

v4.0

open_source

Open Source

Organisation

Issues

Pull Request

Grundprinzipien von Git

beitragen

Motivation

1. Issue

1. Pull Request

Tools

Caterina Ma

Snapshots, statt Unterschiede

Fast jede Operation ist lokal

Integrität

Git fügt Daten hinzu

Die drei Zustände

Check-Ins over Time

Version 1

Datei A

Datei B

Datei C

Check-Ins over Time

Version 1

Version 2

Datei A

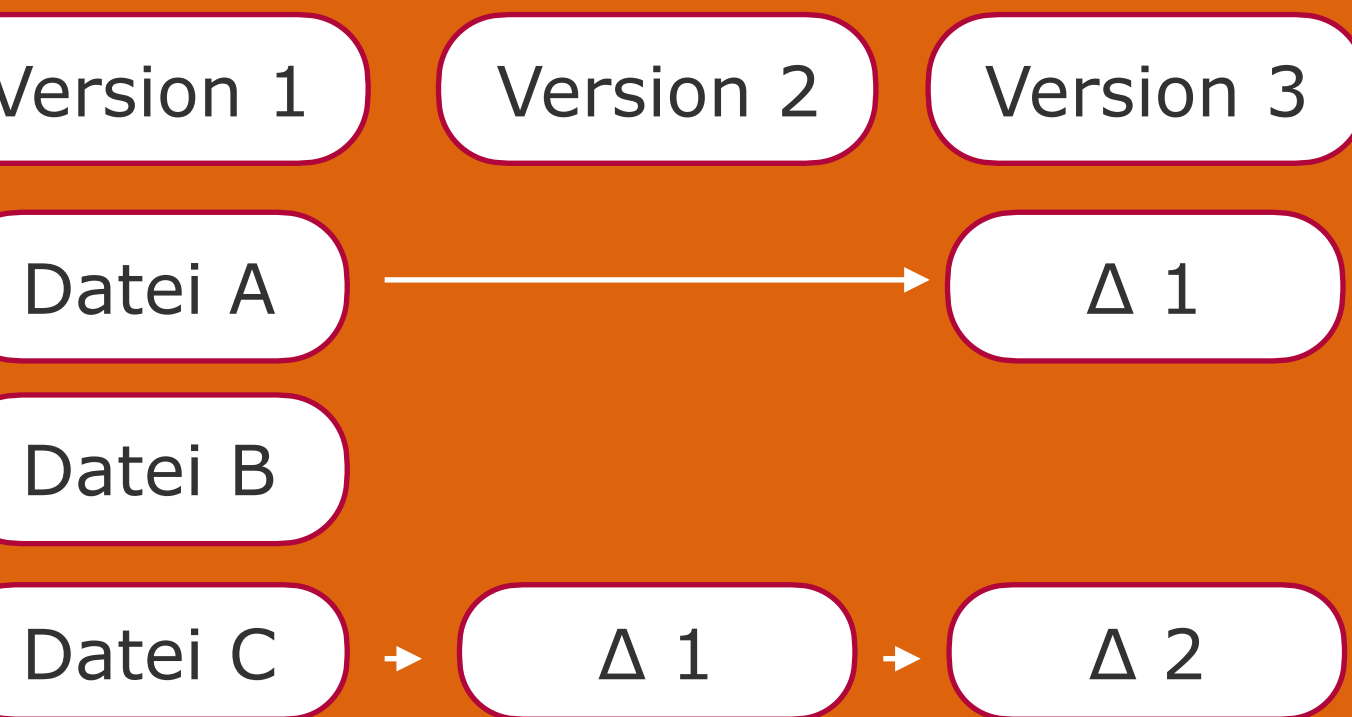
Datei B

Datei C

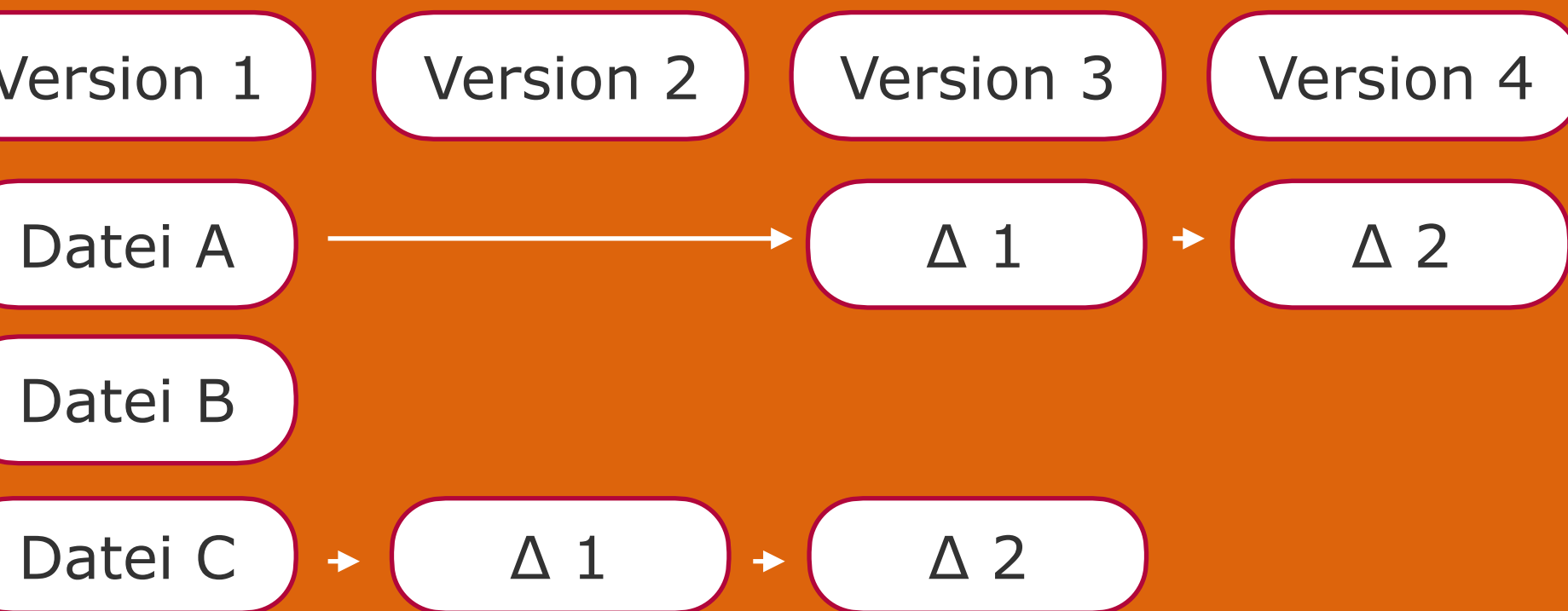


Δ 1

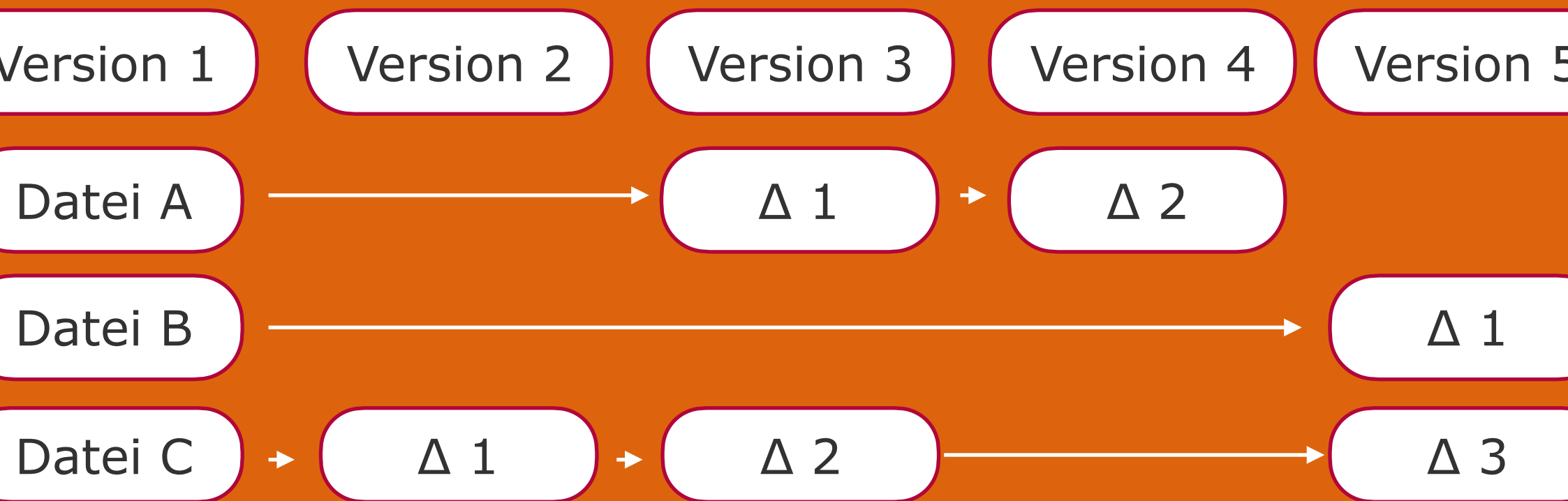
Check-Ins over Time



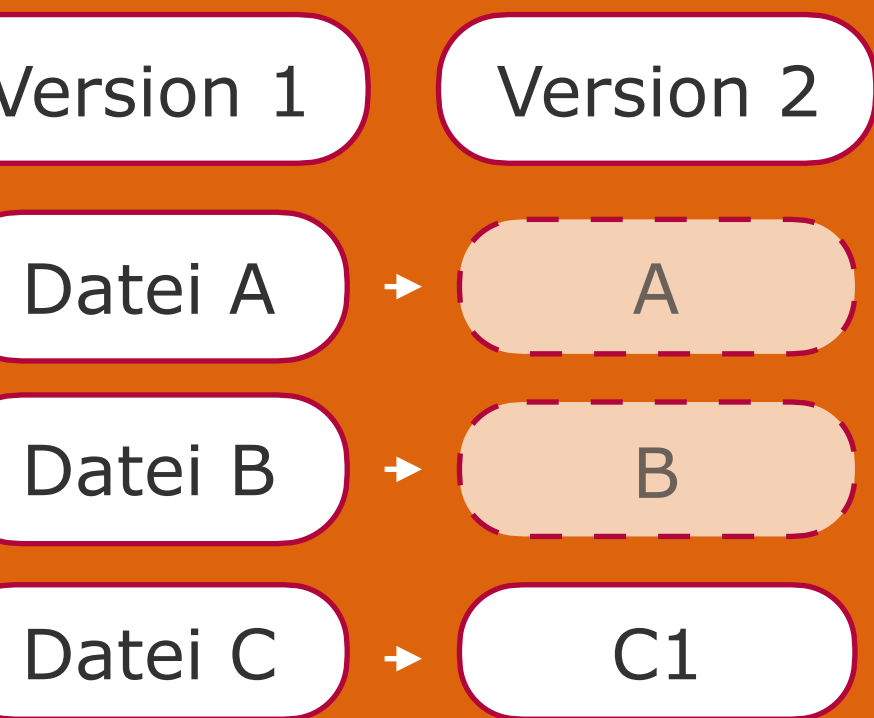
Check-Ins over Time



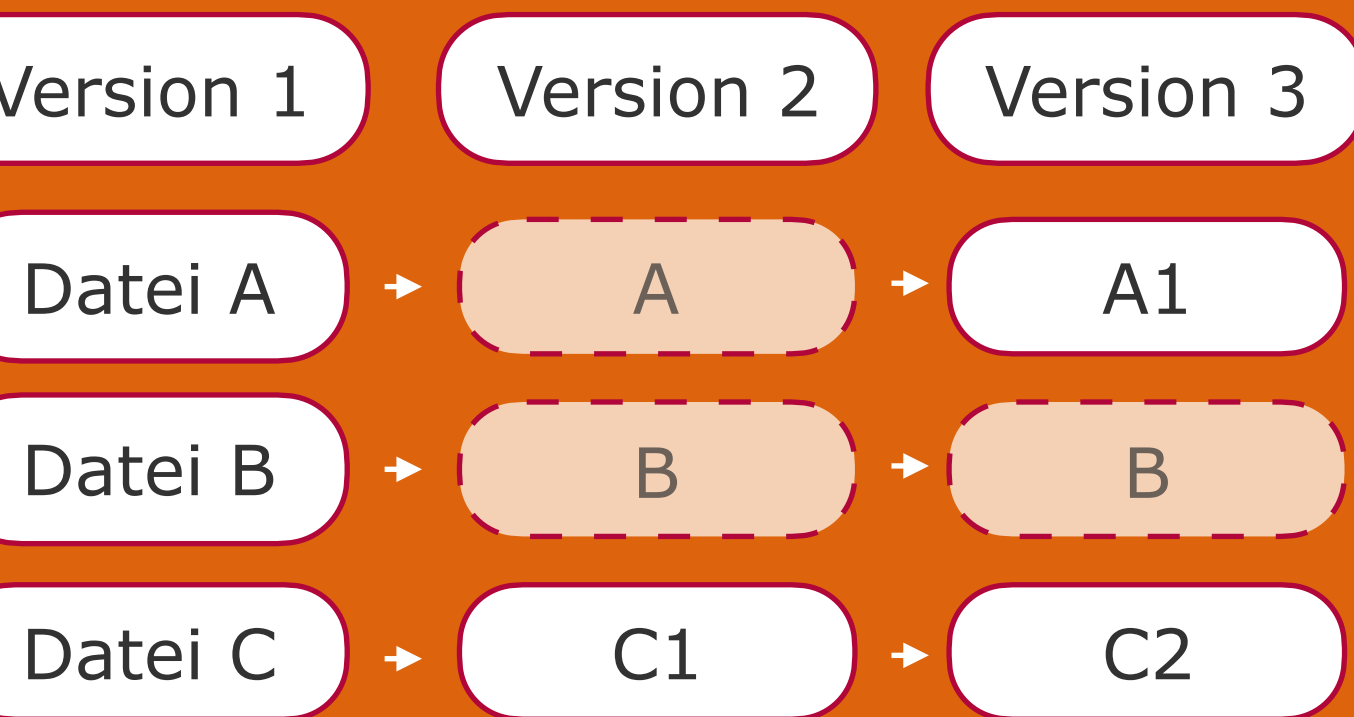
Check-Ins over Time



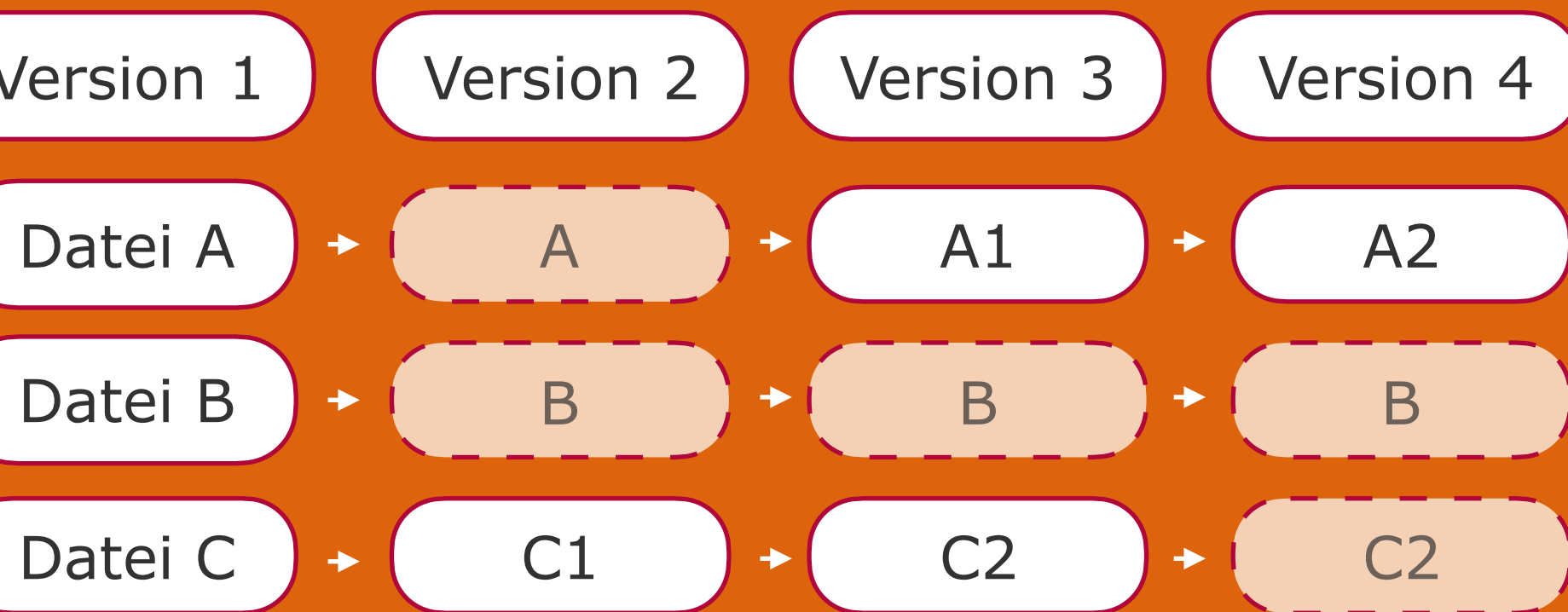
Check-Ins over Time



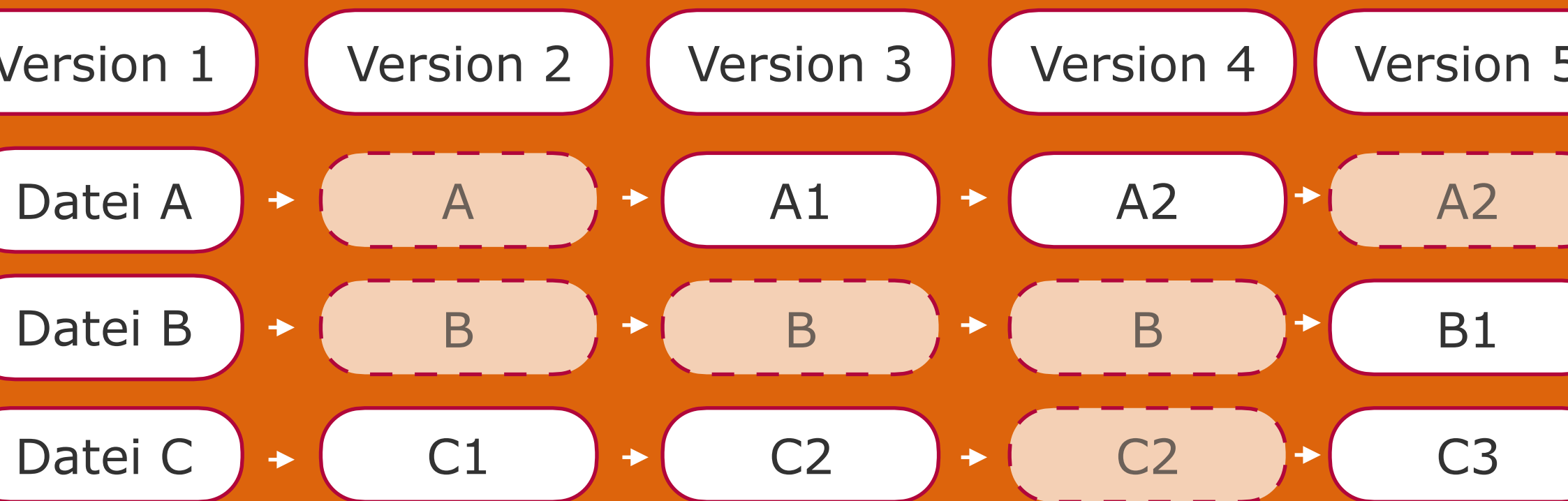
Check-Ins over Time



Check-Ins over Time



Check-Ins over Time



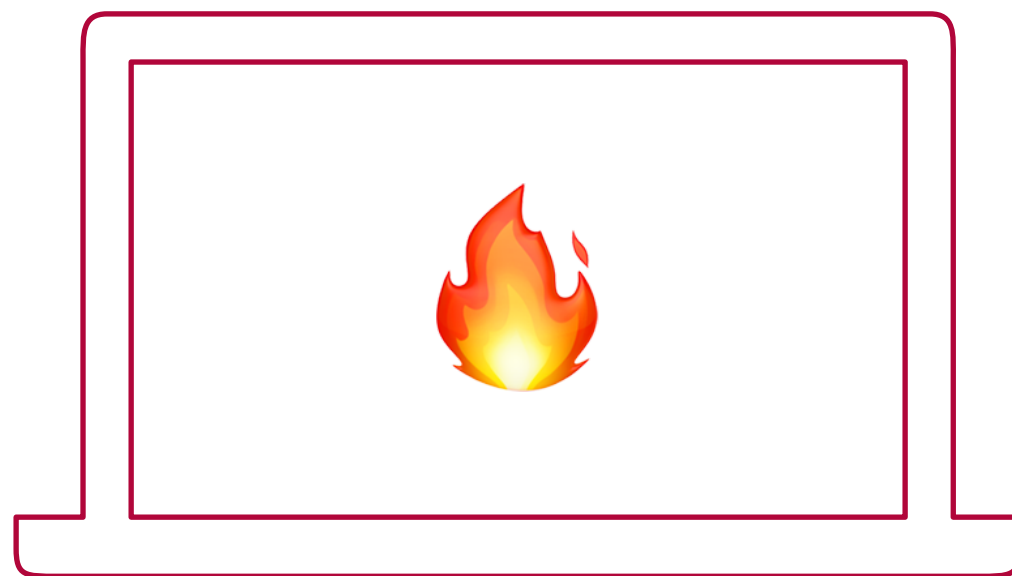
Snapshots, statt Unterschiede ✓

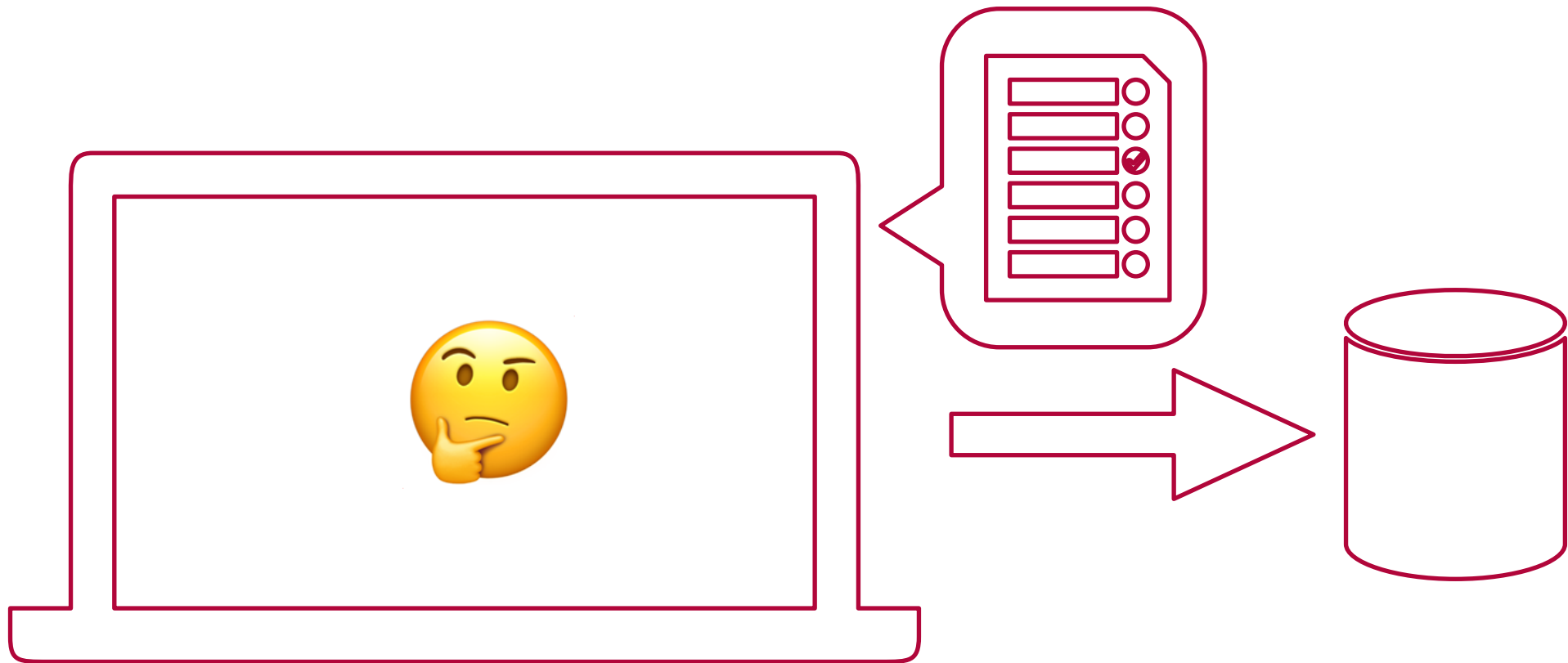
Fast jede Operation ist lokal

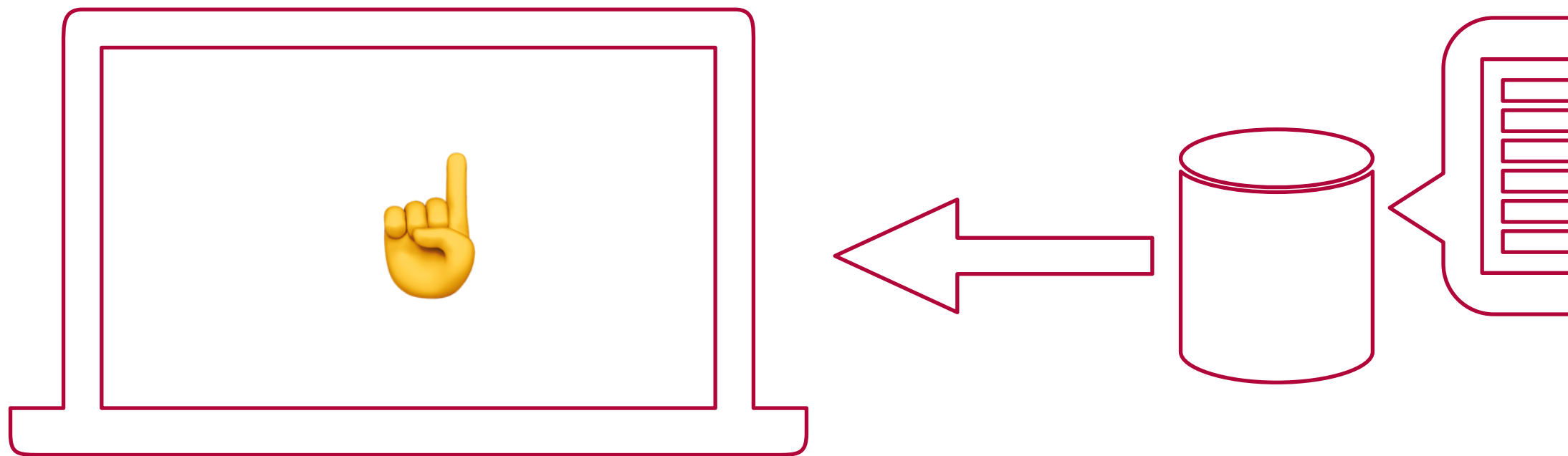
Integrität

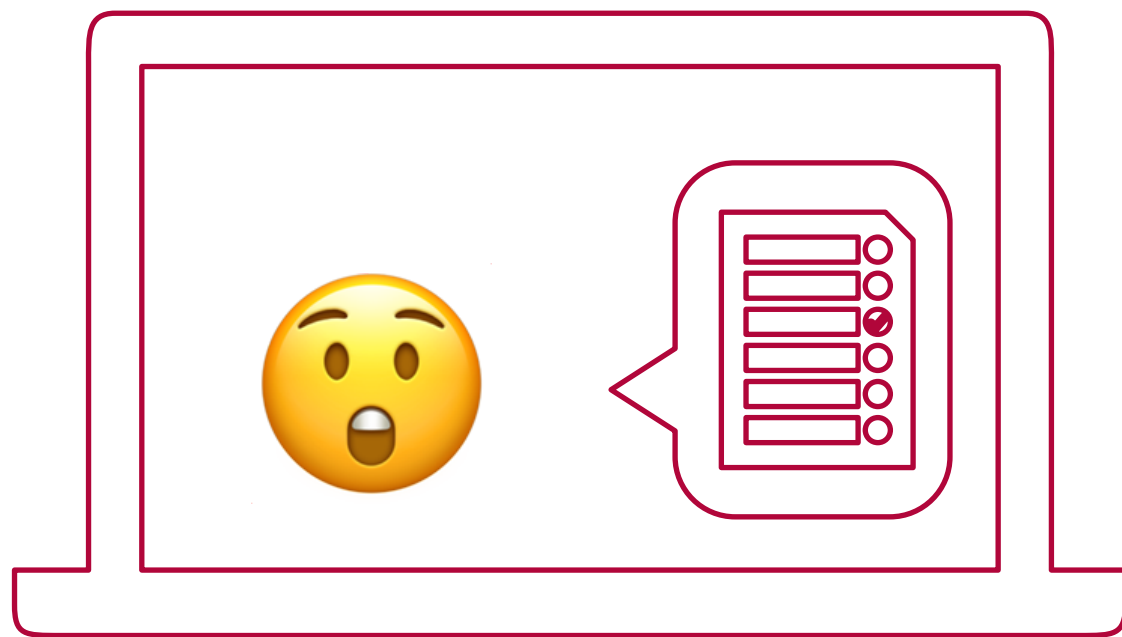
Git fügt Daten hinzu

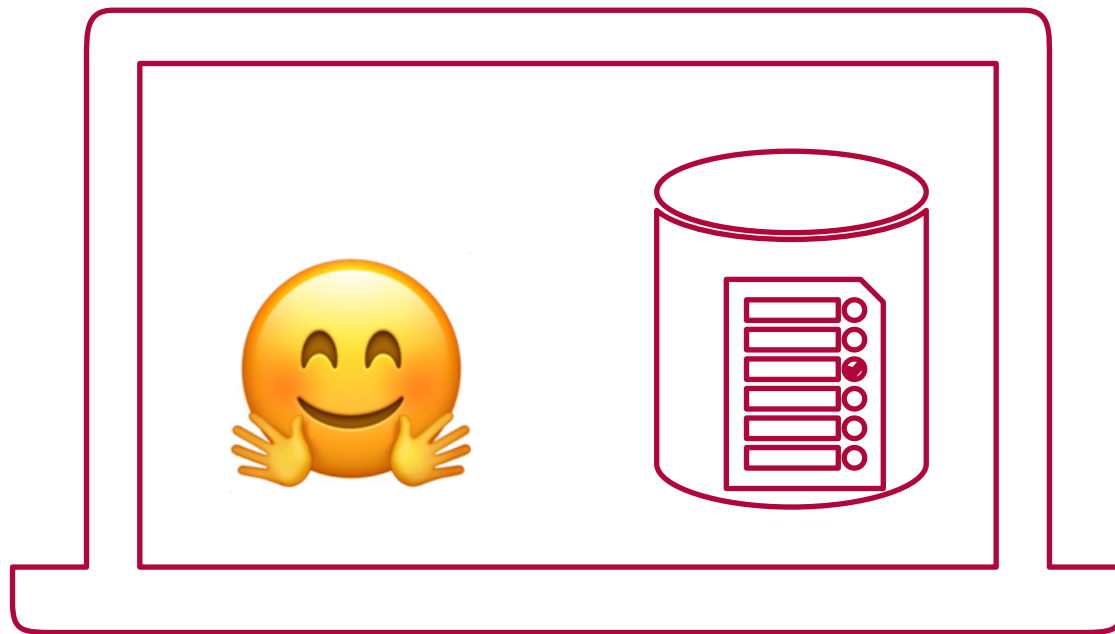
Die drei Zustände











Snapshots, statt Unterschiede ✓

Fast jede Operation ist lokal ✓

Integrität

Git fügt Daten hinzu

Die drei Zustände



24b9da6552252987



24b9da65522525as

Snapshots, statt Unterschiede ✓

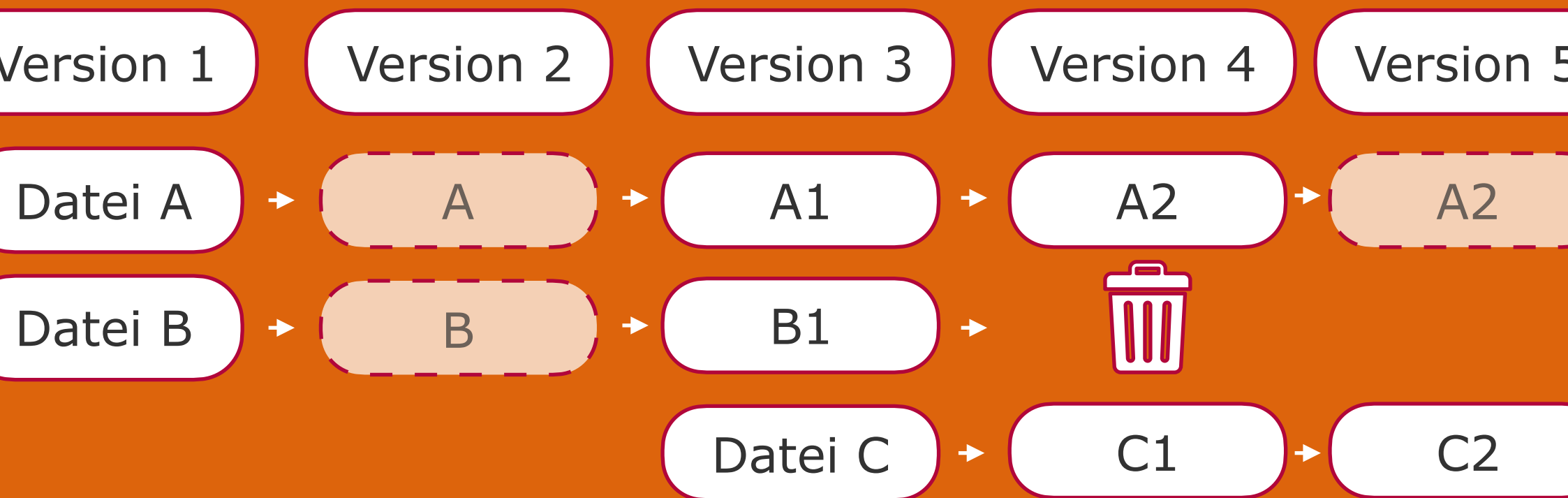
Fast jede Operation ist lokal ✓

Integrität ✓

Git fügt Daten hinzu

Die drei Zustände

Check-Ins over Time



Snapshots, statt Unterschiede ✓

Fast jede Operation ist lokal ✓

Integrität ✓

Git fügt Daten hinzu ✓

Die drei Zustände



Modified



Staged

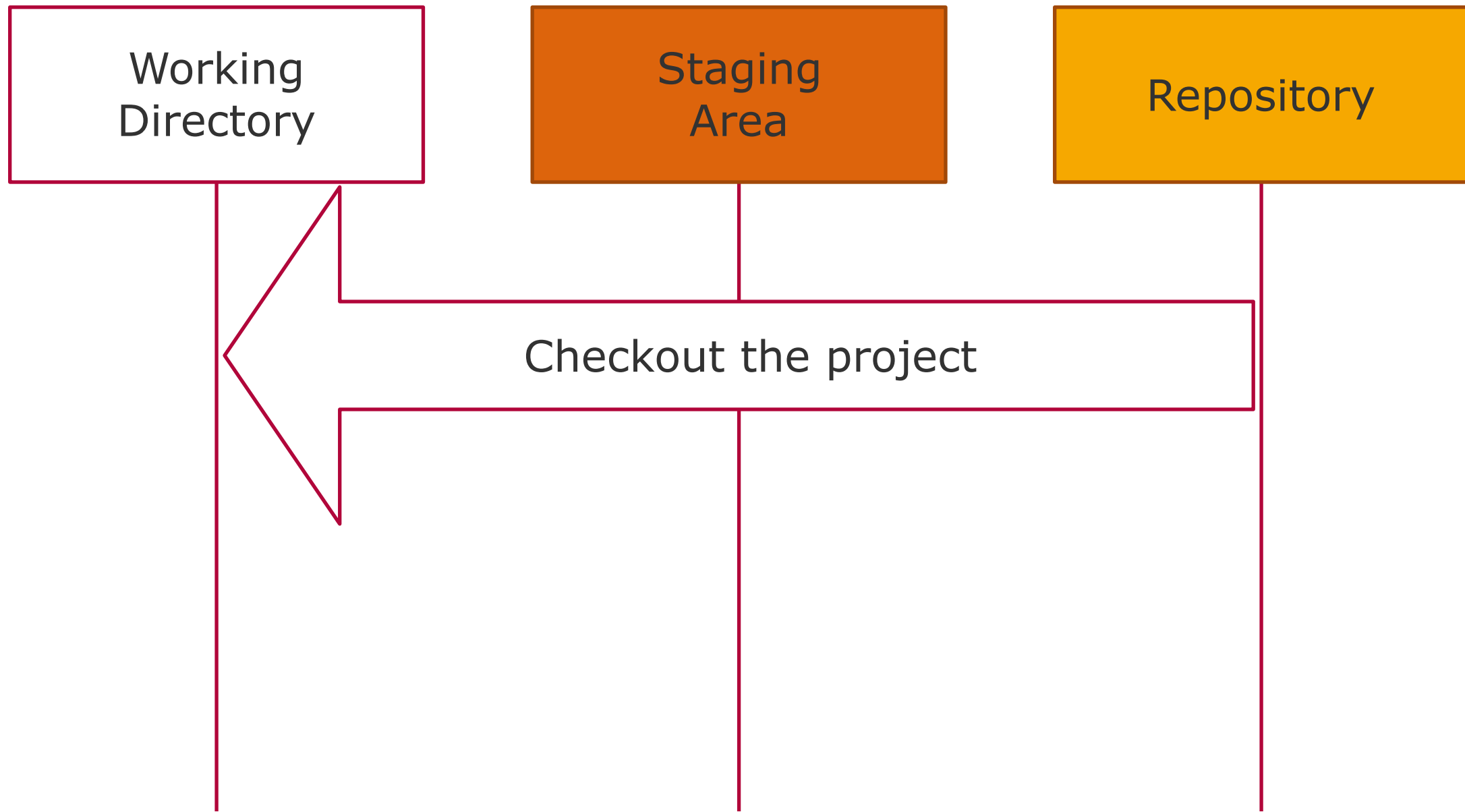


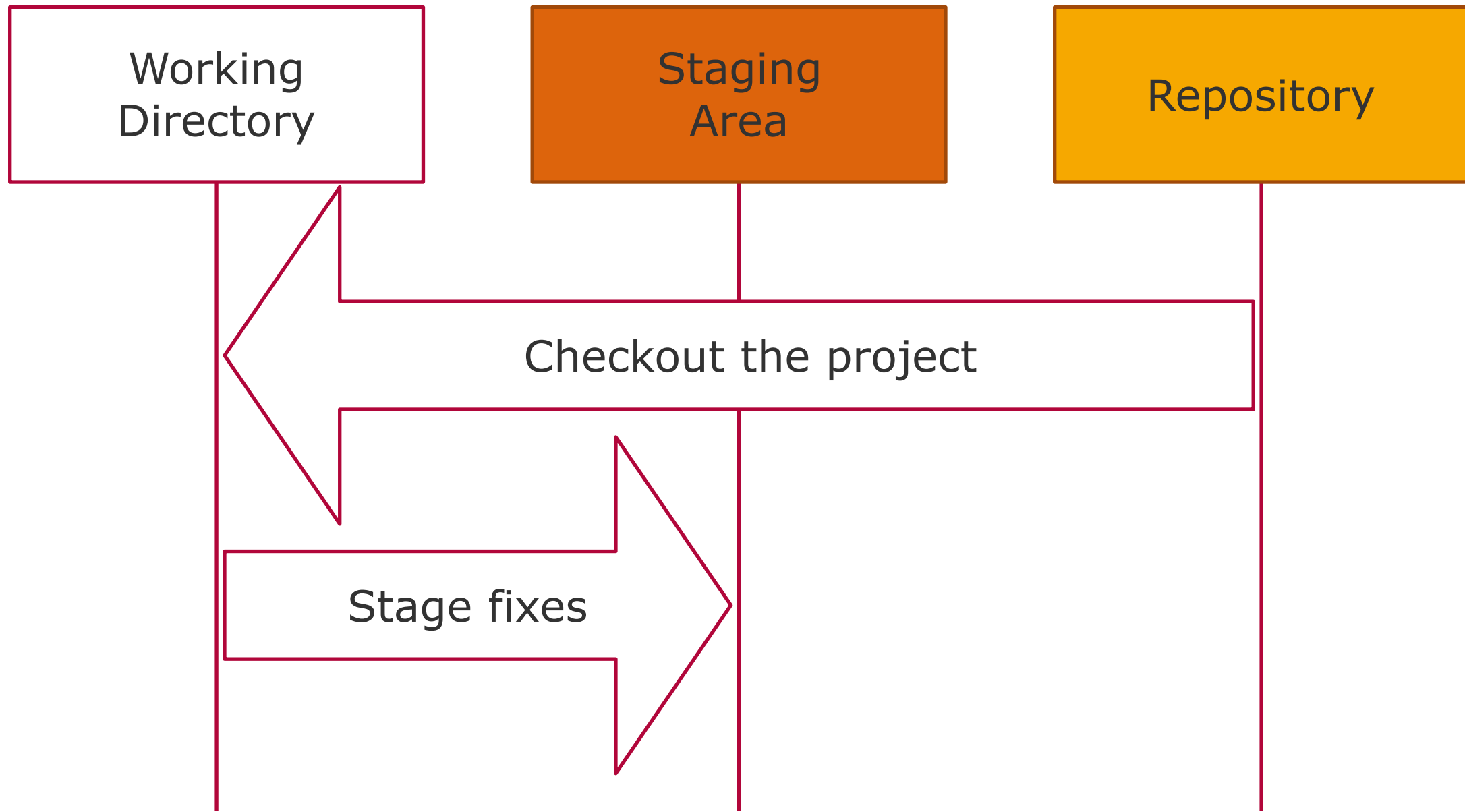
Committed

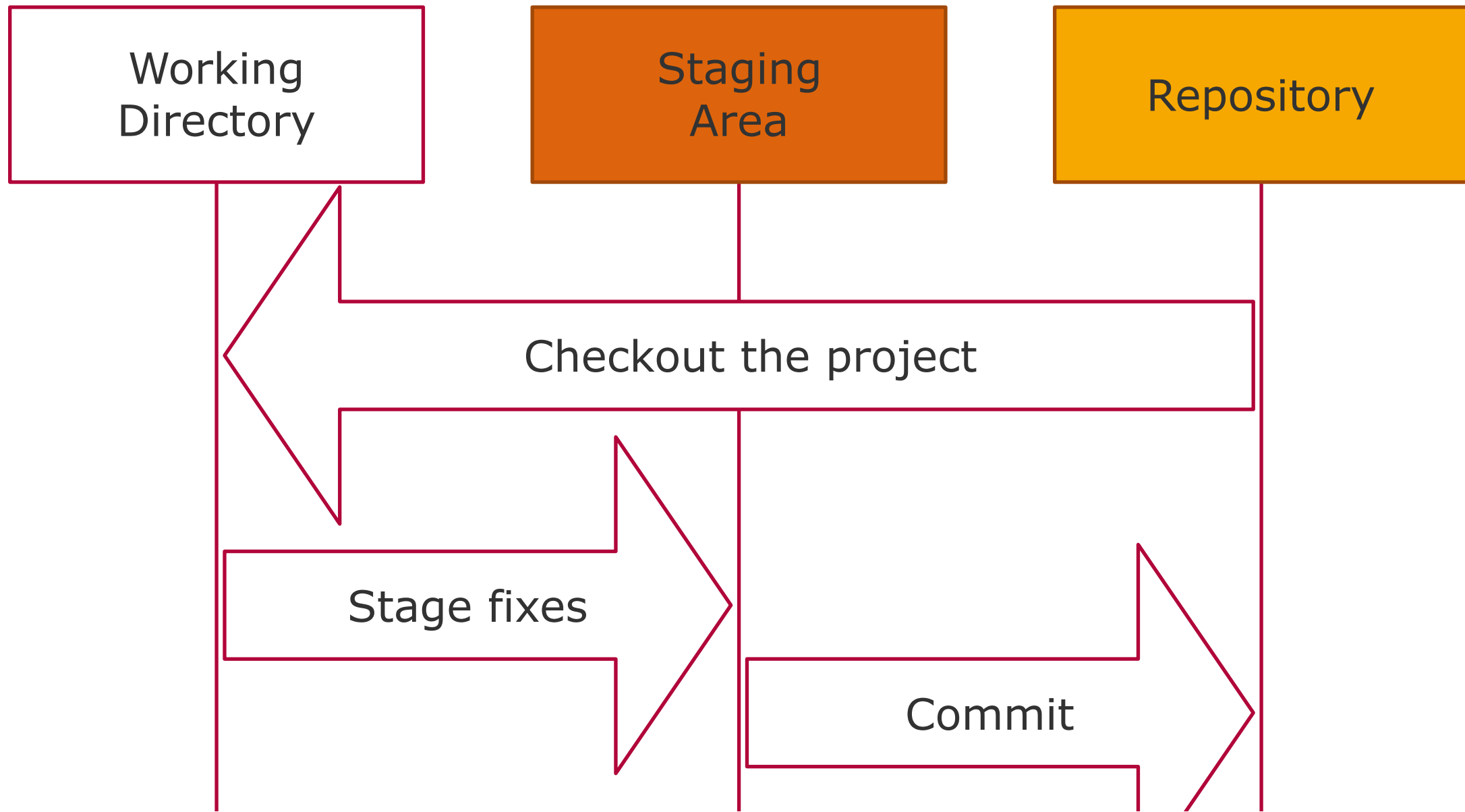
Working
Directory

Staging
Area

Repository







Verstehen

shots, lokale, Integrität, fügt Daten hinzu
ed ↔ Staged ↔ Committed

Installieren

manager bzw. <https://git-scm.com/download>

le ein leeres Git-Repository / Reinitialisiere ein
ndenes
nit

zum Index hinzufügen
dd

aus dem Arbeitsbaum und aus dem Index
nen
m

ungen am Repository aufzeichnen

Verwalte einen Satz von verfolgten Repositories
\$ git remote

Klone ein Repository in ein neues Verzeichnis
\$ git clone

Aktualisiere die Remote-Repositories zusammen mit d
zugehörigen Objekten
\$ git push

Objekte und Referenzen aus einem anderen Repository
herunterladen
\$ git fetch

Aus einem anderen Repository oder einem lokalen Zw
holen und in diesen integrieren
\$ git pull

git_versionsverwaltung

init

commit

push

clone

git_kollaboration

branch

merge

diff

conflict

master

v1.0

v2.0

v3.0

v4.0

open_source

Open Source

Organisation

Issues

Pull Request

installieren

beitragen

Motivation

1. Issue

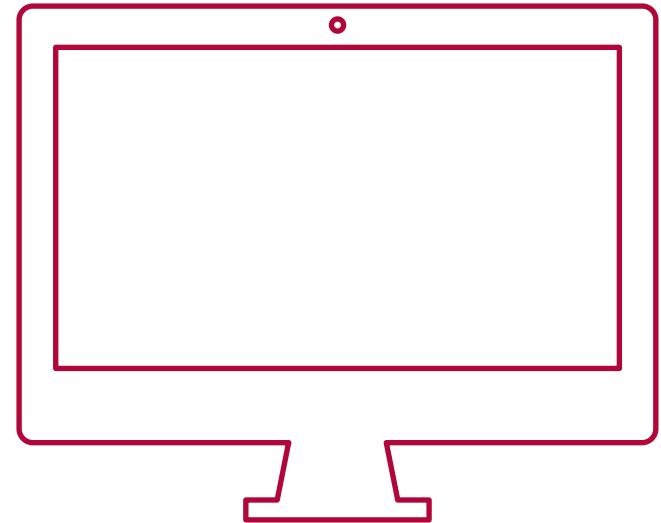
1. Pull Request

Tools

Caterina Ma

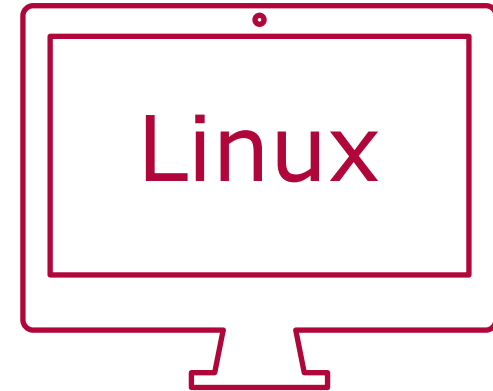
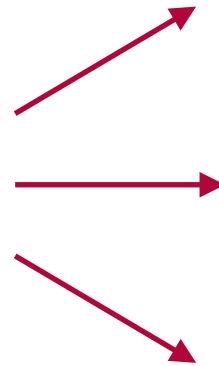


git





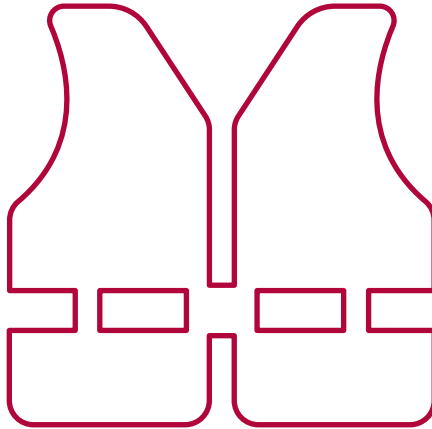
git



Paketmanager



Installationsassistent



Source Code
selbst kompilieren

01001110

tribution Fedora

Paketverwaltungstool: dnf

```
$ dnf install git
```

tribution Ubuntu

Paketverwaltungstool: apt

```
$ apt-get install git
```

de Kommandozeilen-Tools

```
$ git --version
```

der Installationsassistent

<https://git-scm.com/download/mac>

Vorsicht: Download startet automatisch

er Installationsassistent

<https://git-scm.com/download/win>

Vorsicht: Download startet automatisch

an die eigenen Bedürfnisse anpassen

`~/.gitconfig` bzw. `~/.config/git/config`

config Datei im aktuellen Repository

ellungen anzeigen

```
$ git config --list
```

username und E-Mail angeben

```
$ git config --global user.name "Git.Nutzer"
```

```
$ git config --global user.email "git.nutzer@examplemail.com"
```

```
1  [user]
2      name = Pavan Kumar Sunkara
3      email = pavan.sss1991@gmail.com
4      username = pksunkara
5  [core]
6      editor = vim
7      whitespace = fix,-indent-with-non-tab,trailing-space,cr-at-eol
8      excludesfile = ~/.gitignore
9  [sendemail]
10     smtpencryption = tls
11     smtpserver = smtp.gmail.com
12     smtpuser = pavan.sss1991@gmail.com
13     smtppass = password
14     smtpserverport = 587
15  [web]
16     browser = google-chrome
17  [instaweb]
18     httpd = apache2 -f
19  [rerere]
20     enabled = 1
21     autoupdate = 1
22  [push]
23     default = matching
24  [color]
25     ui = auto
26  [color "branch"]
27     current = yellow bold
28     local = green bold
29     remote = cyan bold
30  [color "diff"]
31     meta = yellow bold
```

Verstehen

shots, lokale, Integrität, fügt Daten hinzu
ed ↔ Staged ↔ Committed

Installieren

manager bzw. <https://git-scm.com/download>

le ein leeres Git-Repository / Reinitialisiere ein
ndenes
nit

zum Index hinzufügen
dd

aus dem Arbeitsbaum und aus dem Index
nen
m

ungen am Repository aufzeichnen

Verwalte einen Satz von verfolgten Repositories
\$ git remote

Klone ein Repository in ein neues Verzeichnis
\$ git clone

Aktualisiere die Remote-Repositories zusammen mit
zugehörigen Objekten
\$ git push

Objekte und Referenzen aus einem anderen Repository
herunterladen
\$ git fetch

Aus einem anderen Repository oder einem lokalen
holen und in diesen integrieren
\$ git pull

git_versionsverwaltung

init

commit

push

clone

git_kollaboration

branch

merge

diff

conflict

master

v1.0

v2.0

v3.0

v4.0

open_source

Open Source

Organisation

Issues

Pull Request

init, Git add, Git rm, Git commit

beitragen

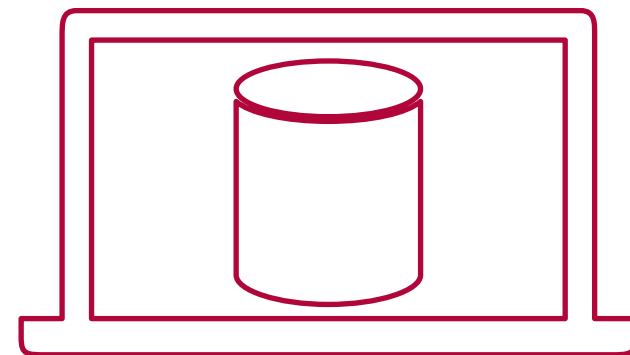
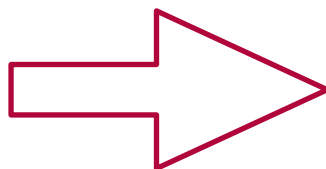
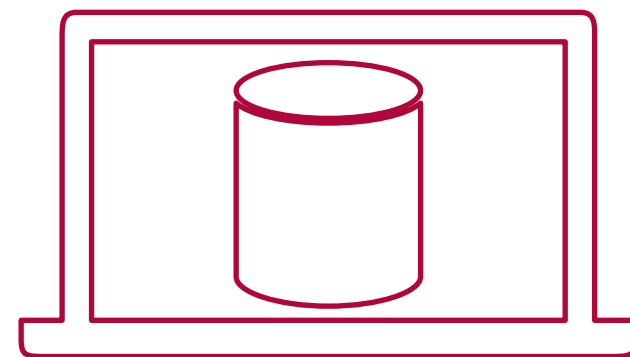
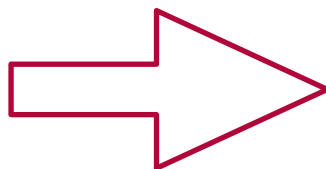
Motivation

1. Issue

1. Pull Request

Tools

Caterina Ma



```
$ cd /Pfad/Projekt
```

```
$ git init
```

```
$ git add beispielbild.jpg
```

```
$ git add *.jpg
```

```
$ git commit
```



```
$ git add beispielbild.jpg
```

```
$ git add *.jpg
```

```
$ git commit
```

oder

```
$ git commit -m 'initial project version'
```

```
$ git add beispielbild.jpg
```

```
$ git add *.jpg
```

```
$ git commit
```

```
$ git remove beispielbild.jpg
```

Untracked

Datei lokal,
nicht im
Repository

vorgemerkt

Staged

Datei markiert,
commit überträgt
ins Repository

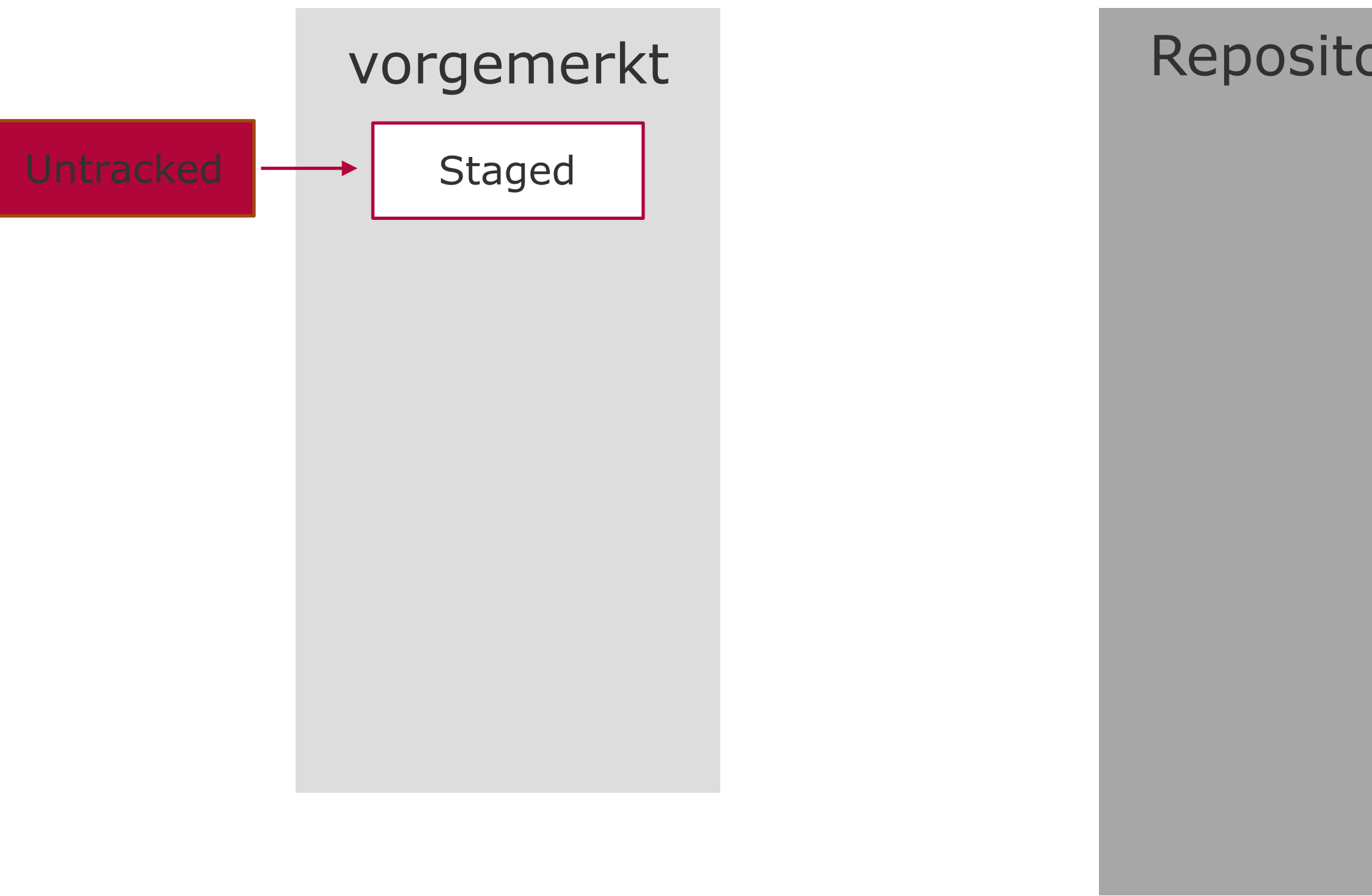
Modified

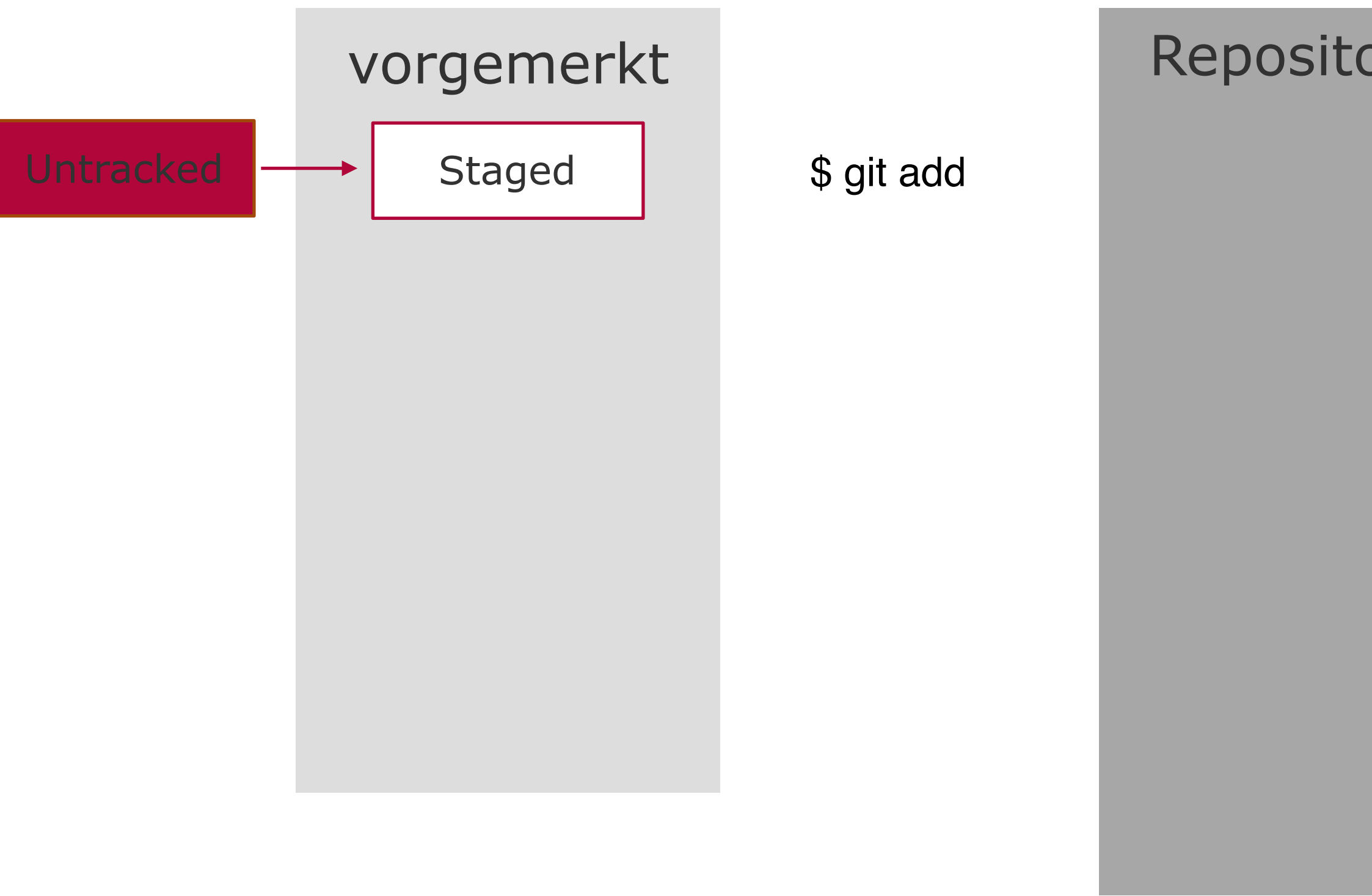
Datei in älterer
Version im
Repository, lokale
Änderung nicht

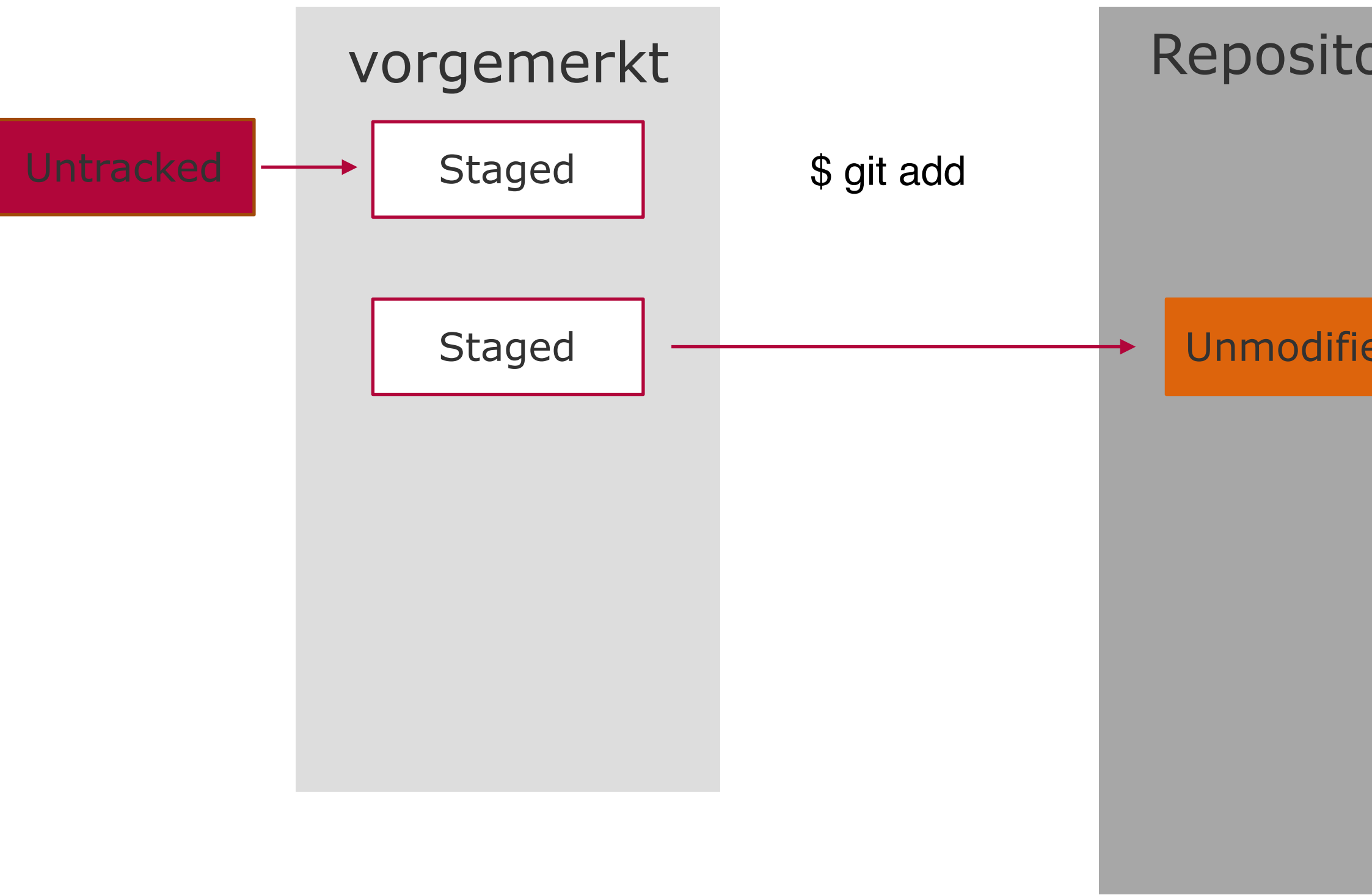
Repository

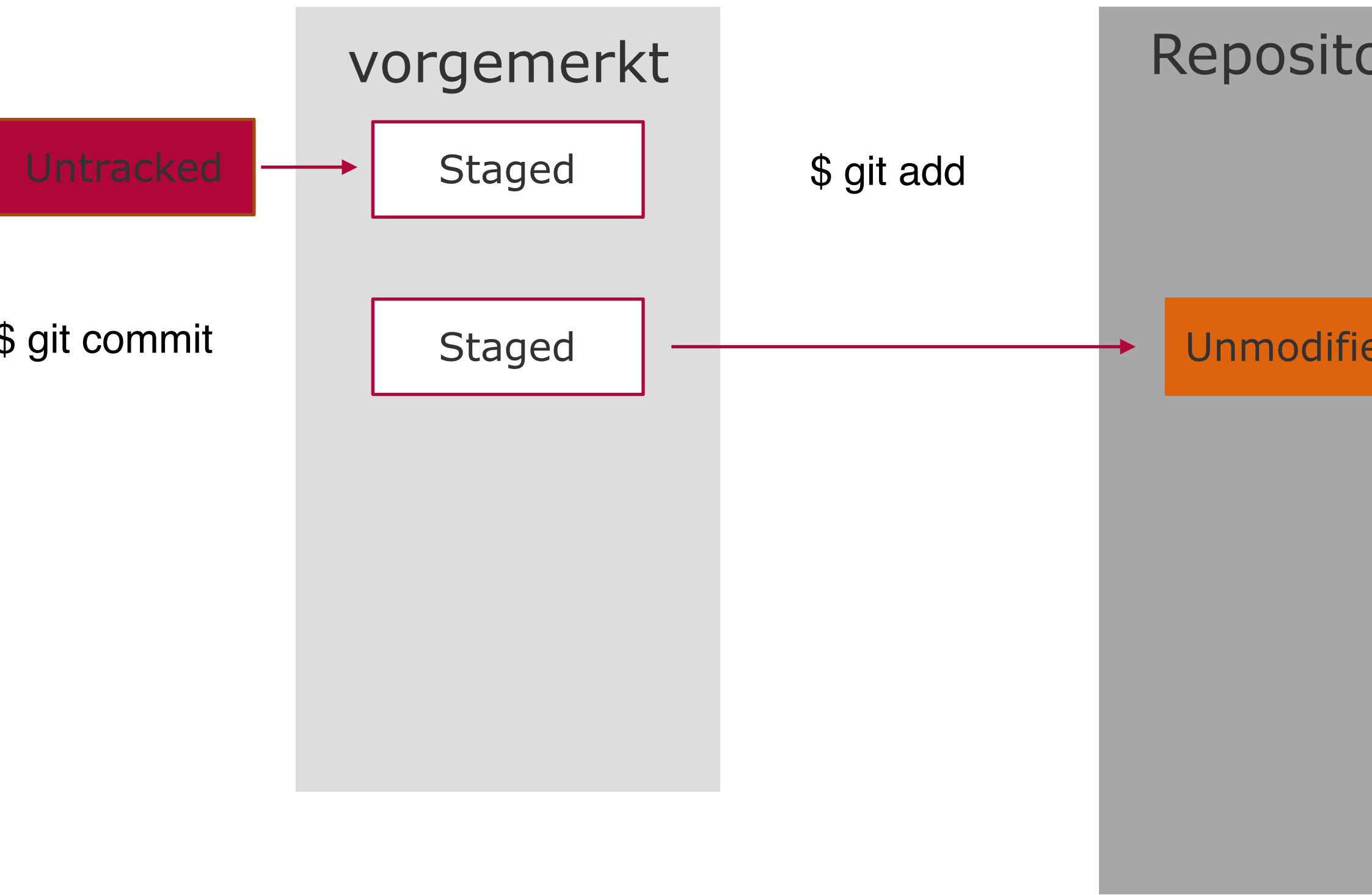
Unmodified

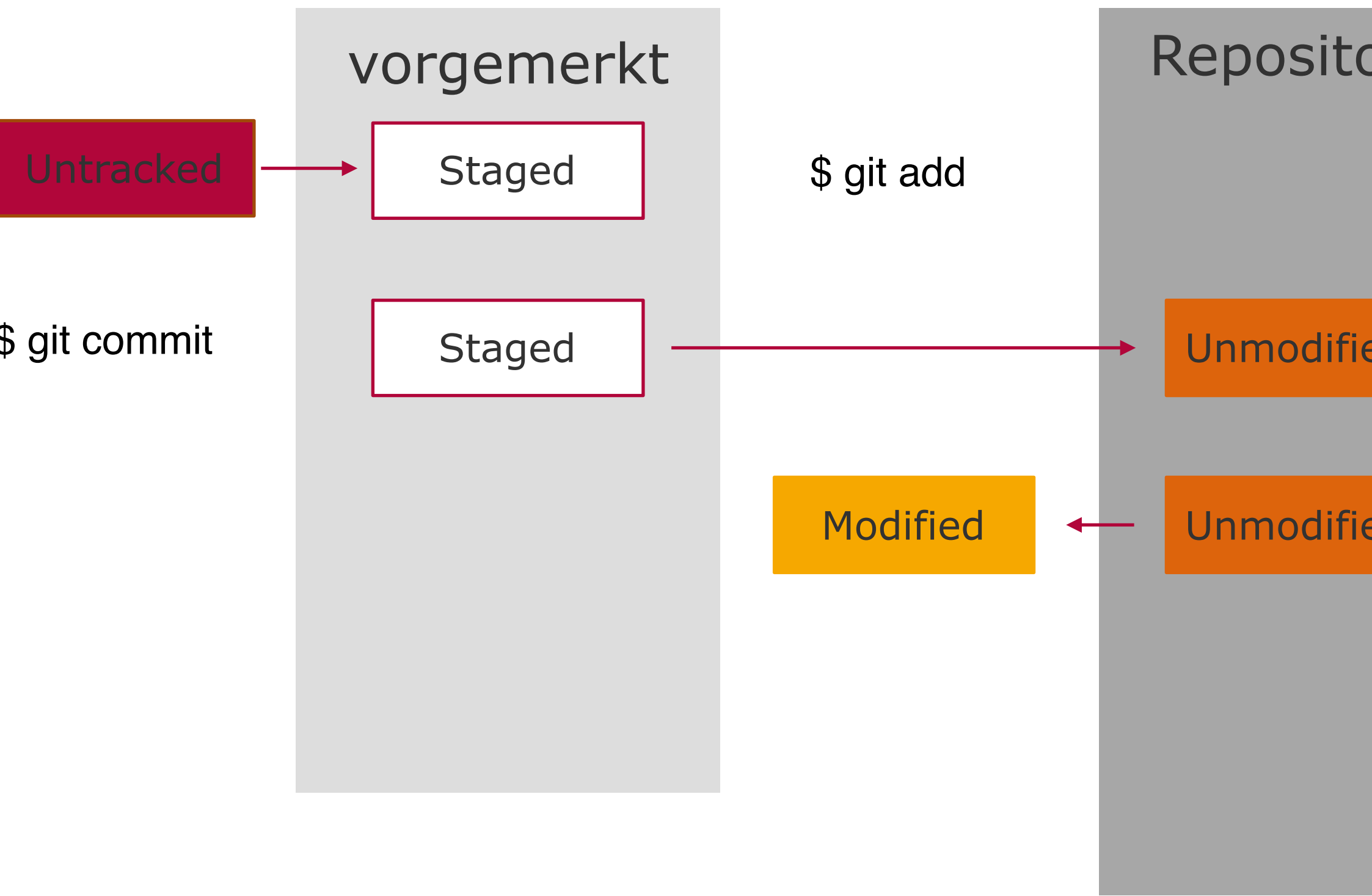
Datei im aktu-
Stand im
Repository

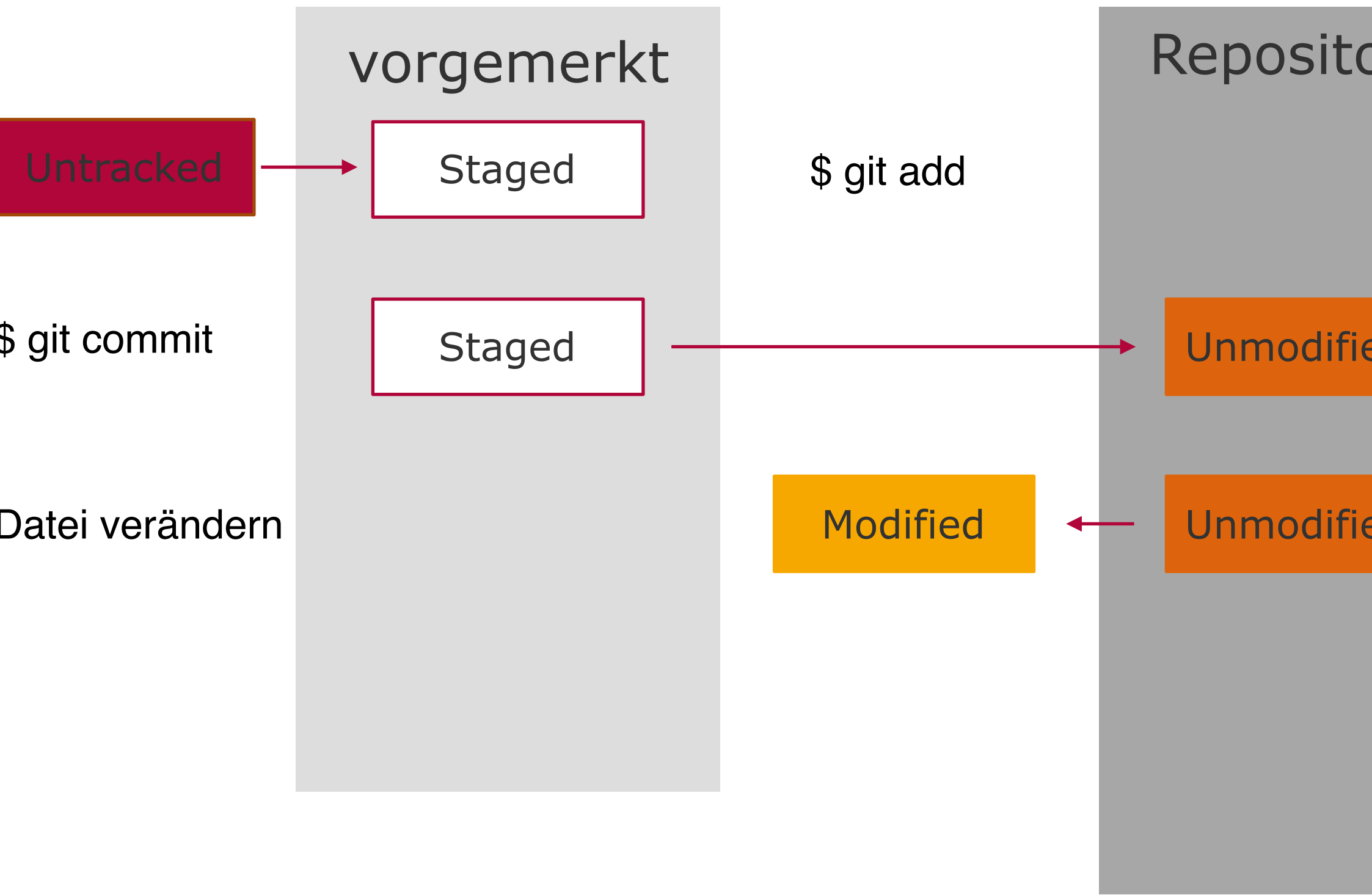


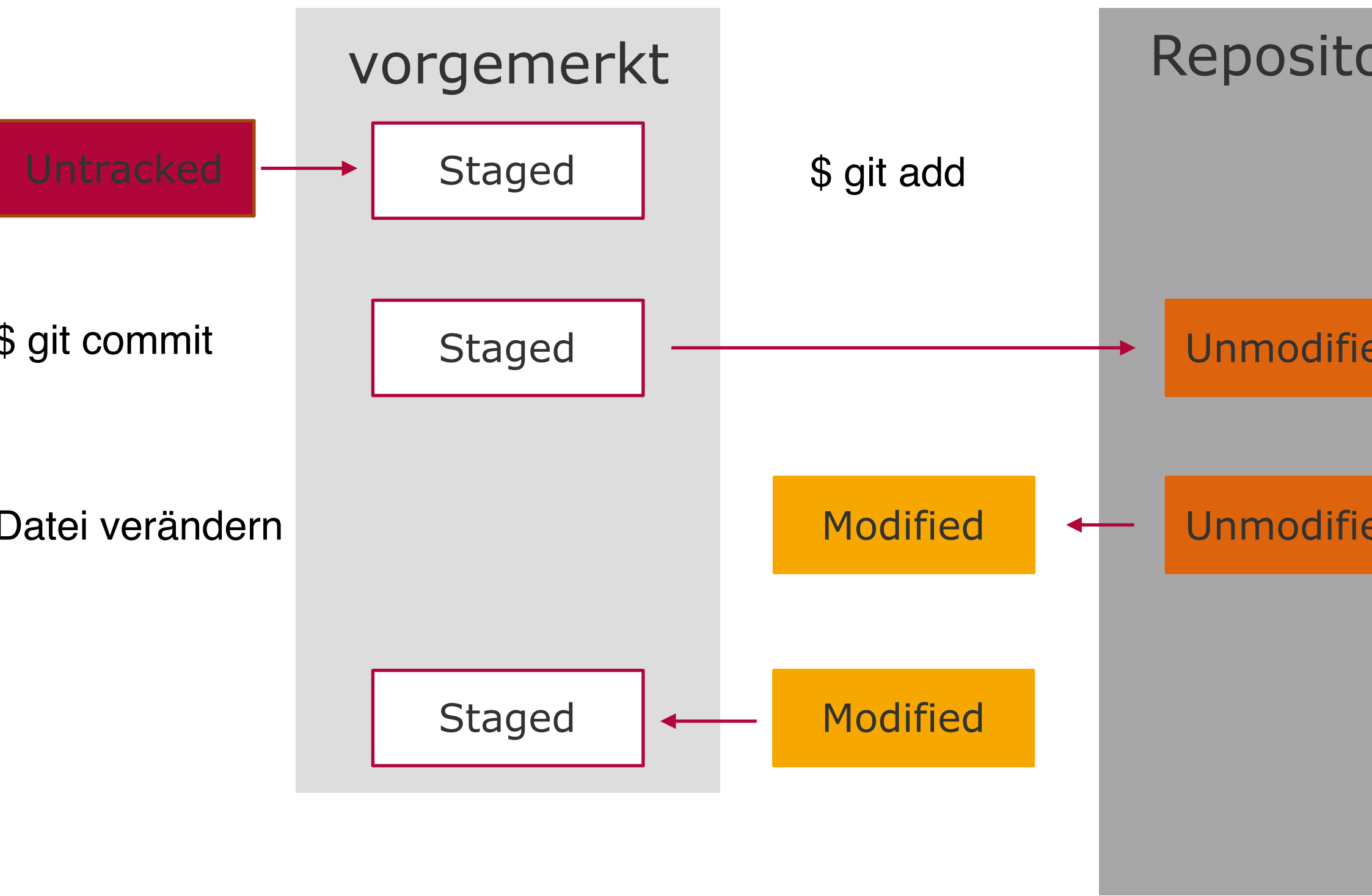


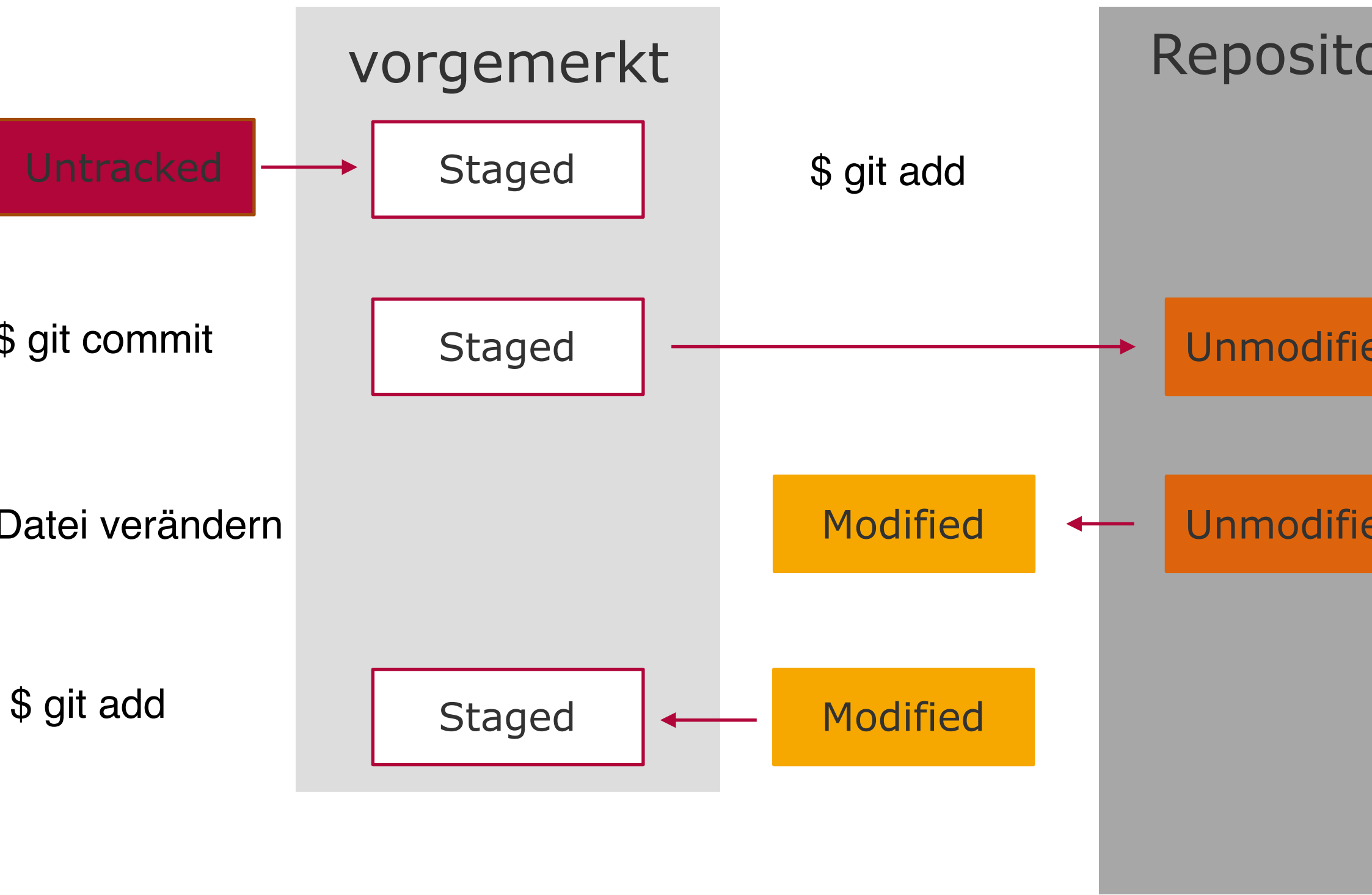


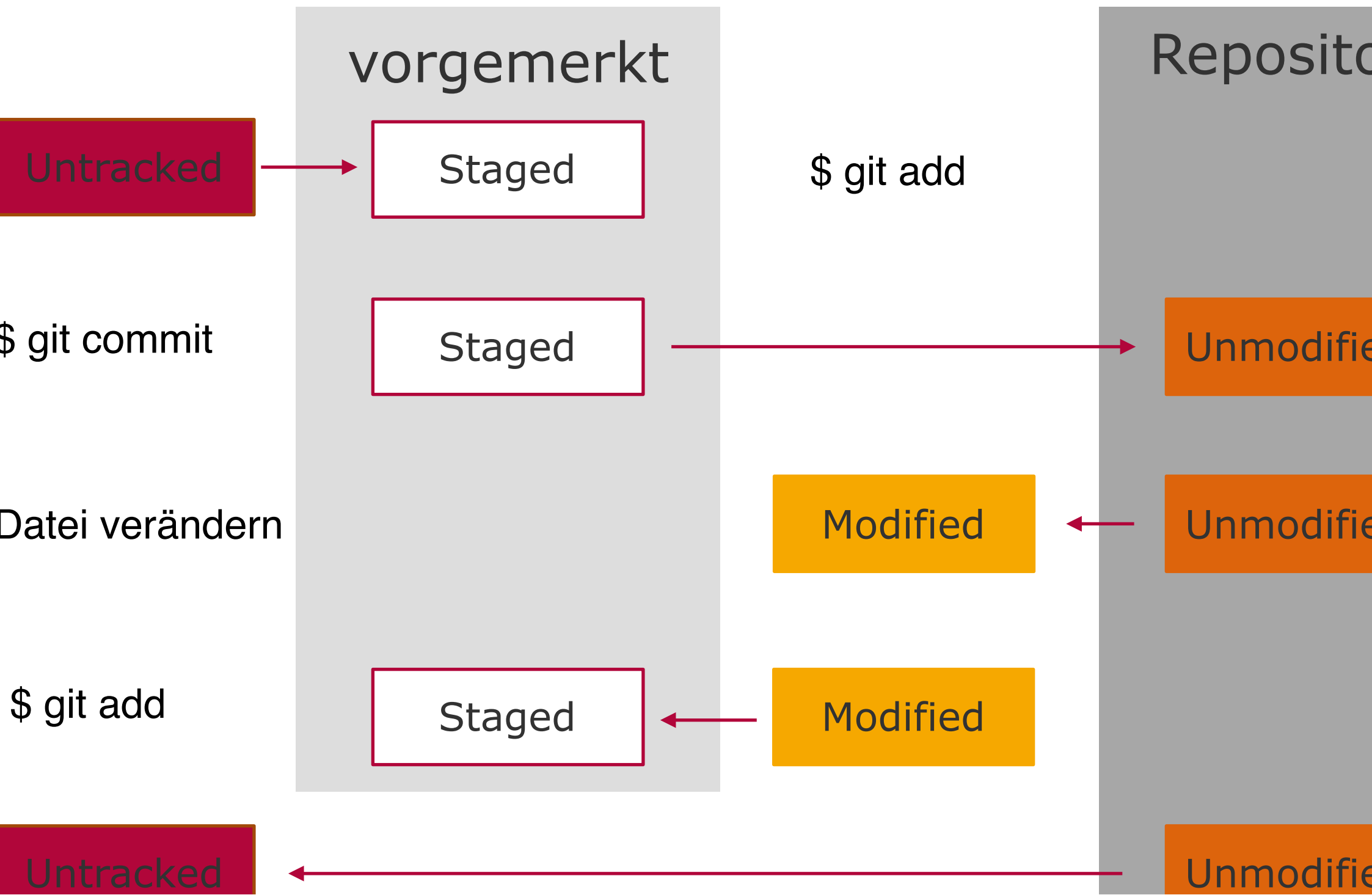


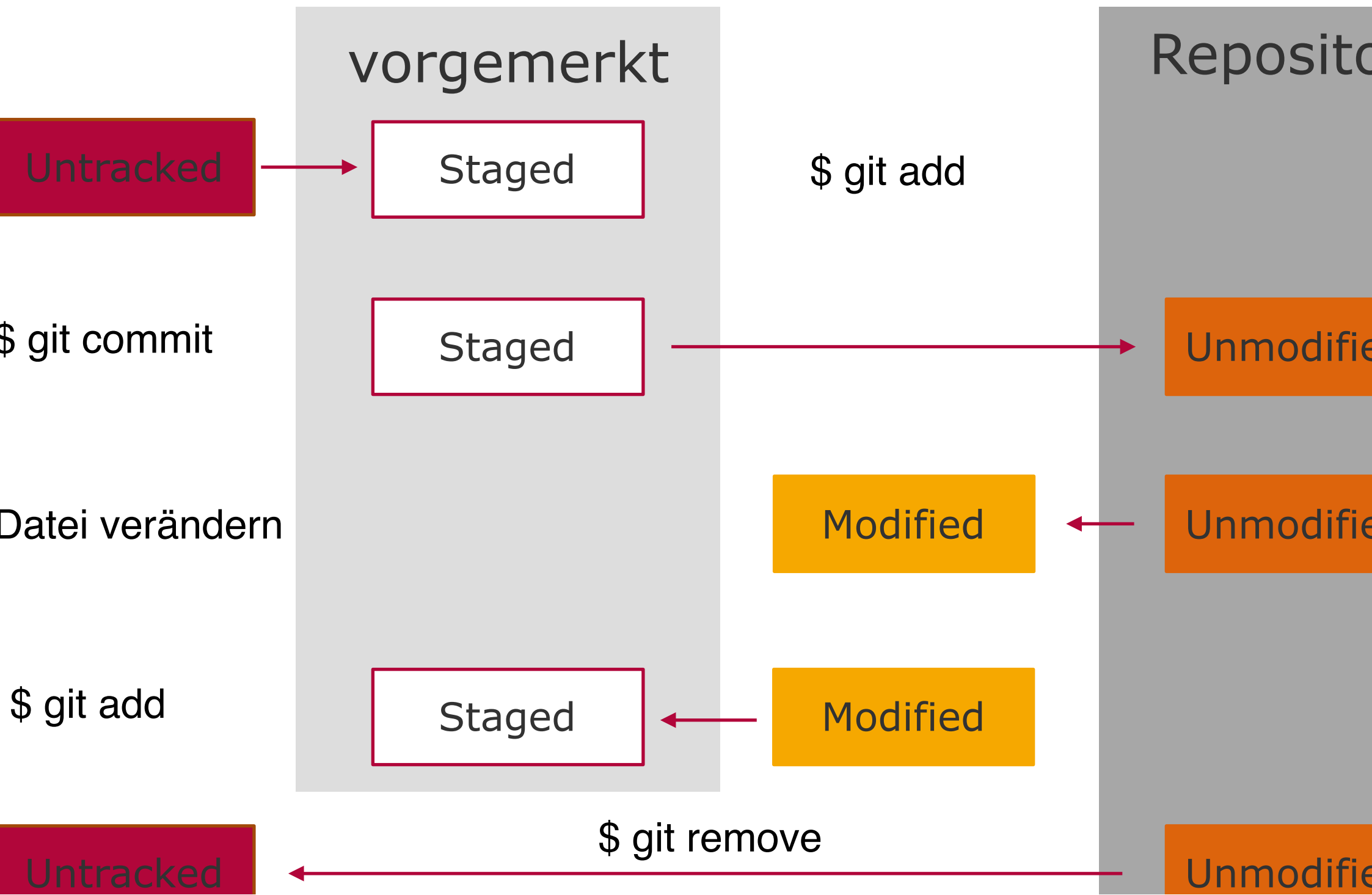












Verstehen

shots, lokale, Integrität, fügt Daten hinzu
ed ↔ Staged ↔ Committed

Installieren

manager bzw. <https://git-scm.com/download>

le ein leeres Git-Repository / Reinitialisiere ein
ndenes

nit

zum Index hinzufügen

dd

aus dem Arbeitsbaum und aus dem Index
nen

m

ungen am Repository aufzeichnen

Verwalte einen Satz von verfolgten Repositories

\$ git remote

Klone ein Repository in ein neues Verzeichnis

\$ git clone

Aktualisiere die Remote-Repositories zusammen mit
zugehörigen Objekten

\$ git push

Objekte und Referenzen aus einem anderen Repo
herunterladen

\$ git fetch

Aus einem anderen Repository oder einem lokalen
holen und in diesen integrieren

\$ git pull

git_versionsverwaltung

init

commit

push

clone

git_kollaboration

branch

merge

diff

conflict

master

v1.0

v2.0

v3.0

v4.0

open_source

Open Source

Organisation

Issues

Pull Request

remote, Git clone

beitragen

Motivation

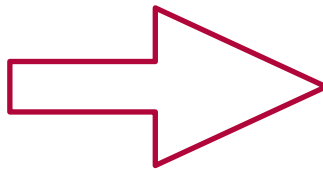
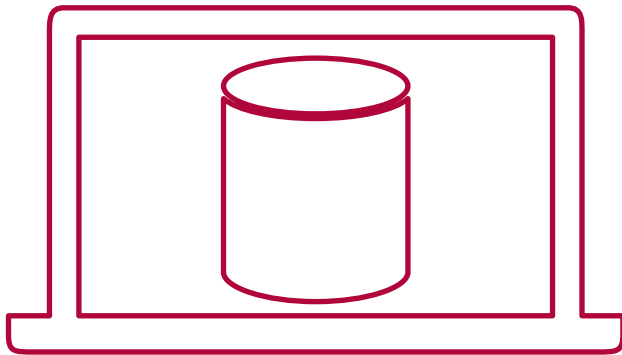
1. Issue

1. Pull Request

Tools

Caterina Ma

remote



remote add <alias><url>

it remote add repo <https://github.com/repo>

remote rename <alias><neues alias>

it remote rename repo projekt1

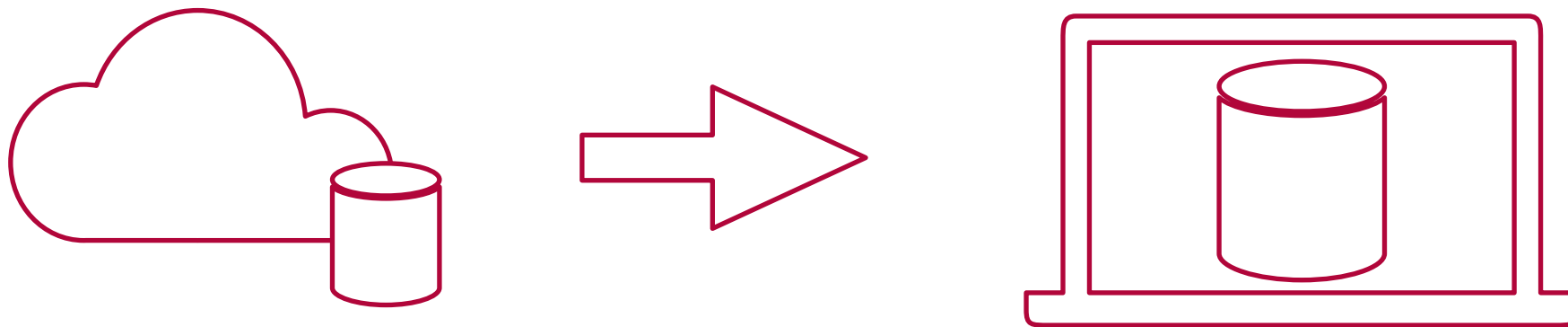
remote remove <alias>

: \$ git remote rm <alias>

it remote rm projekt1

clone <url> <alias>

it clone <https://github.com/repo> repo



- Vollständige Kopie fast aller Daten
- Versionierung jeder Datei

Verstehen:

shots, lokale, Integrität, fügt Daten hinzu
ed ↔ Staged ↔ Committed

Installieren:

manager bzw. <https://git-scm.com/download>

le ein leeres Git-Repository /
ualisiere ein vorhandenes
nit

zum Index hinzufügen
dd

aus dem Arbeitsbaum und aus dem Index
nen
m

ungen am Repository aufzeichnen

Verwalte einen Satz von verfolgten Repositories

\$ git remote

Klone ein Repository in ein neues Verzeichnis

\$ git clone

Aktualisiere die Remote-Repositories zusammen
zugehörigen Objekten

\$ git push

Objekte und Referenzen aus einem anderen Repo
herunterladen

\$ git fetch

Aus einem anderen Repository oder einem lokalen
holen und in diesen integrieren

\$ git pull

git_versionsverwaltung

init

commit

push

clone

git_kollaboration

branch

merge

diff

conflict

master

v1.0

v2.0

v3.0

v4.0

open_source

Open Source

Organisation

Issues

Pull Request

push, Git fetch, Git pull

beitragen

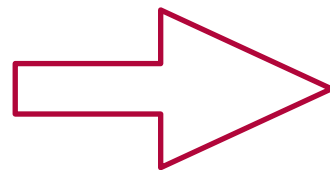
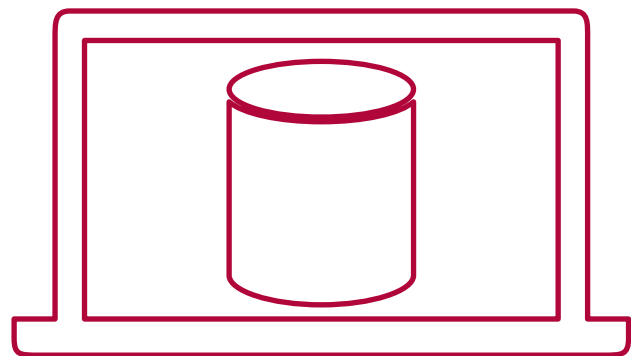
Motivation

1. Issue

1. Pull Request

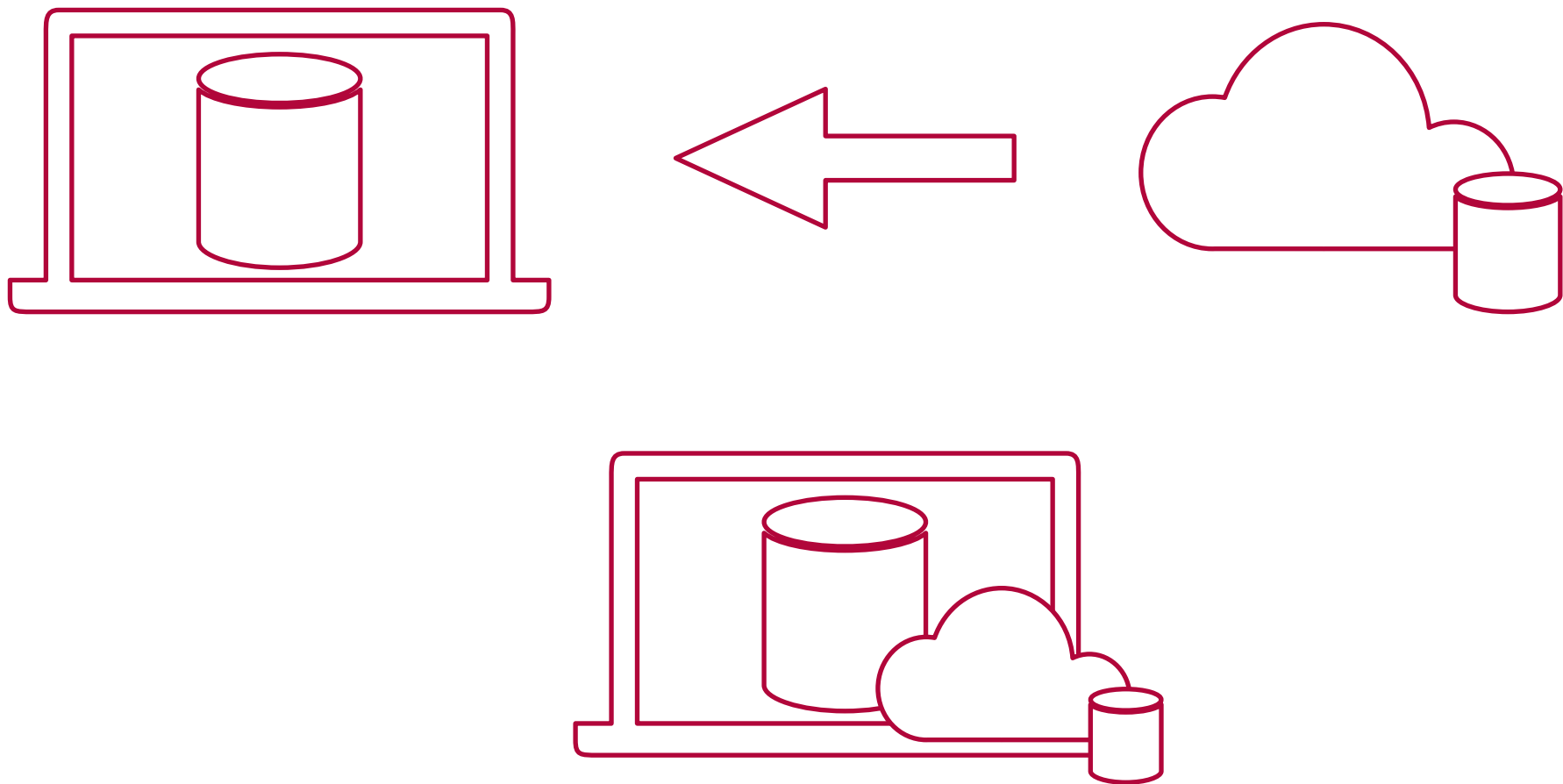
Tools

Caterina Ma



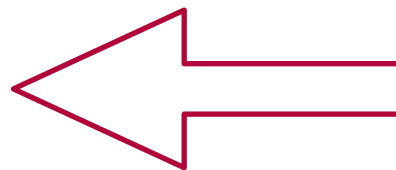
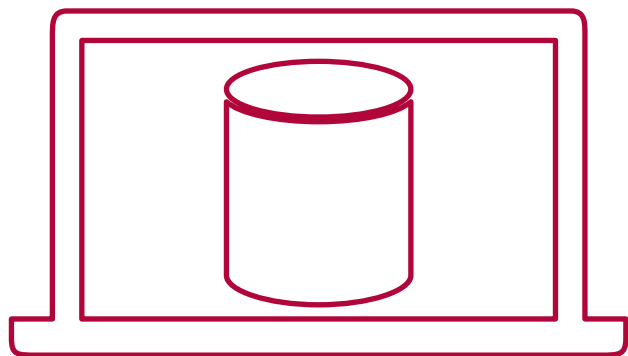
push origin/<branch>

it push origin/master



fetch origin

merge origin/<branch>



pull =
git fetch +
git merge

Verstehen:

shots, lokale, Integrität, fügt Daten hinzu
ed ↔ Staged ↔ Committed

Installieren:

manager bzw. <https://git-scm.com/download>

le ein leeres Git-Repository /
ualisiere ein vorhandenes
nit

zum Index hinzufügen
dd

aus dem Arbeitsbaum und aus dem Index
nen
m

ungen am Repository aufzeichnen

Verwalte einen Satz von verfolgten Repositories

\$ git remote

Klone ein Repository in ein neues Verzeichnis

\$ git clone

Aktualisiere die Remote-Repositories zusammen
zugehörigen Objekten

\$ git push

Objekte und Referenzen aus einem anderen Repo
herunterladen

\$ git fetch

Aus einem anderen Repository oder einem lokalen
holen und in diesen integrieren

\$ git pull