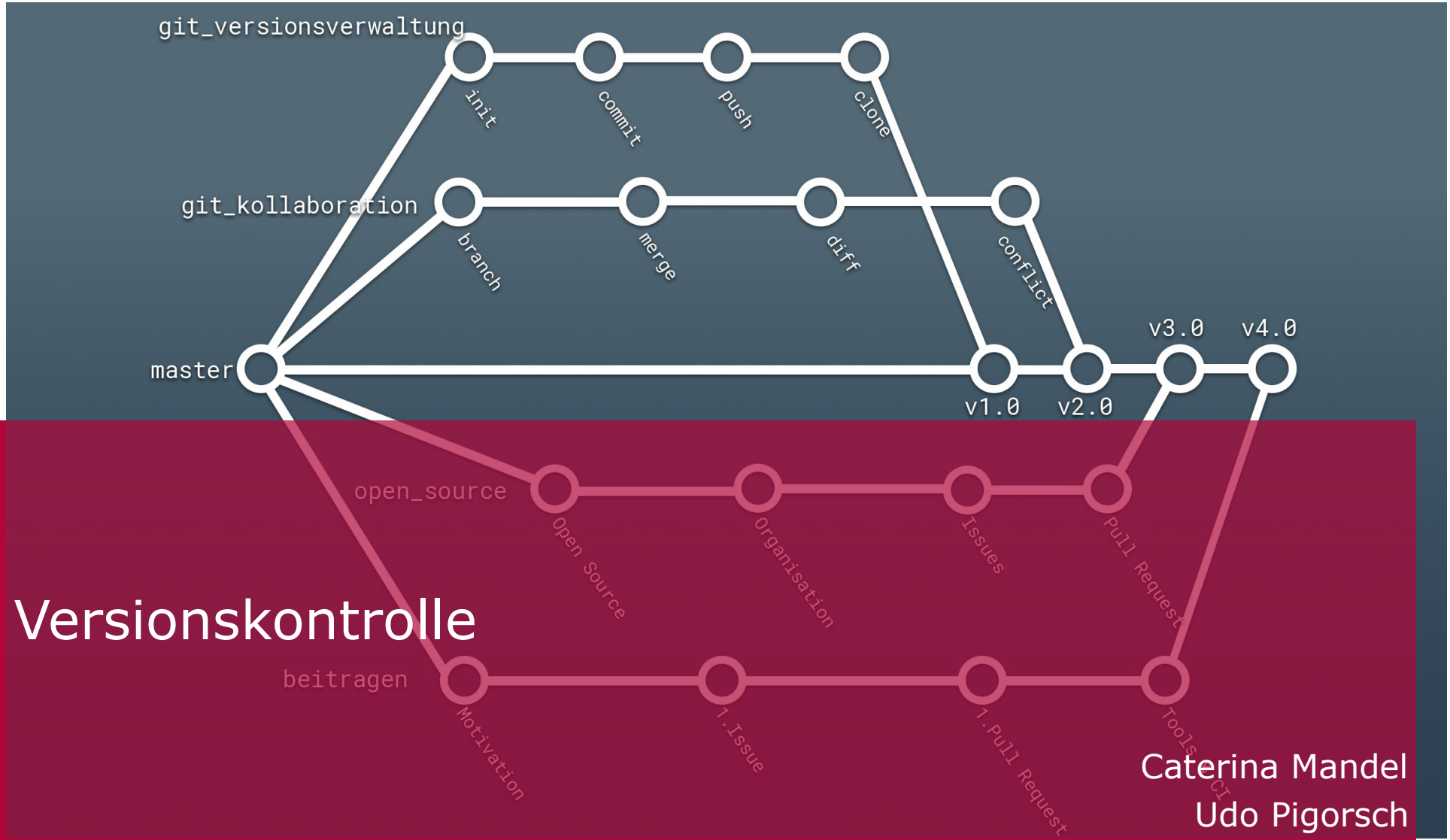


Versionskontrolle

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

Datei1

Datei1

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

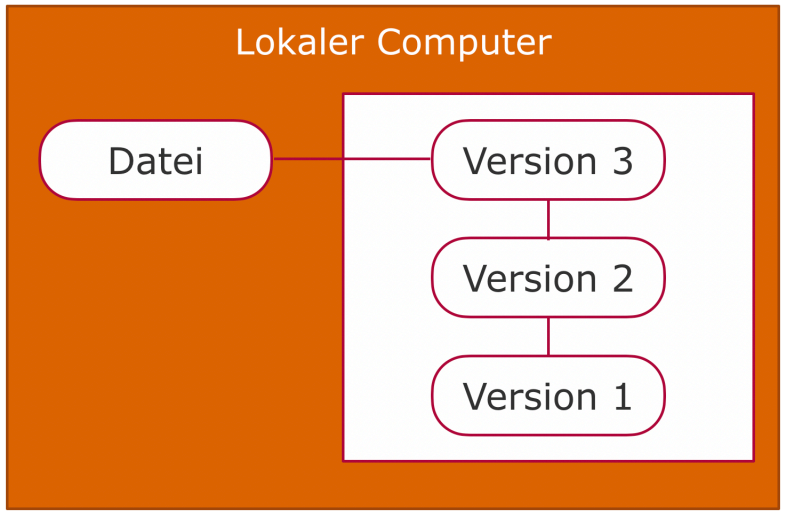
Zentralisierte Versionskontrolle

- Centralised Version Control System - CVCS

Verteilte Versionskontrolle

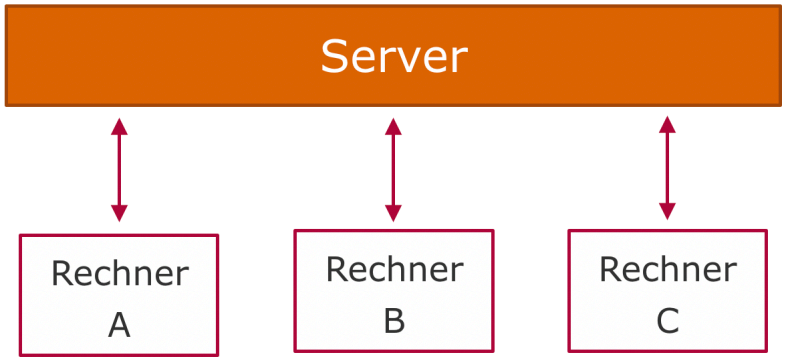
- Distributed Version Control System - DVCS

Formen der Versionskontrolle: Lokale Versionskontrolle



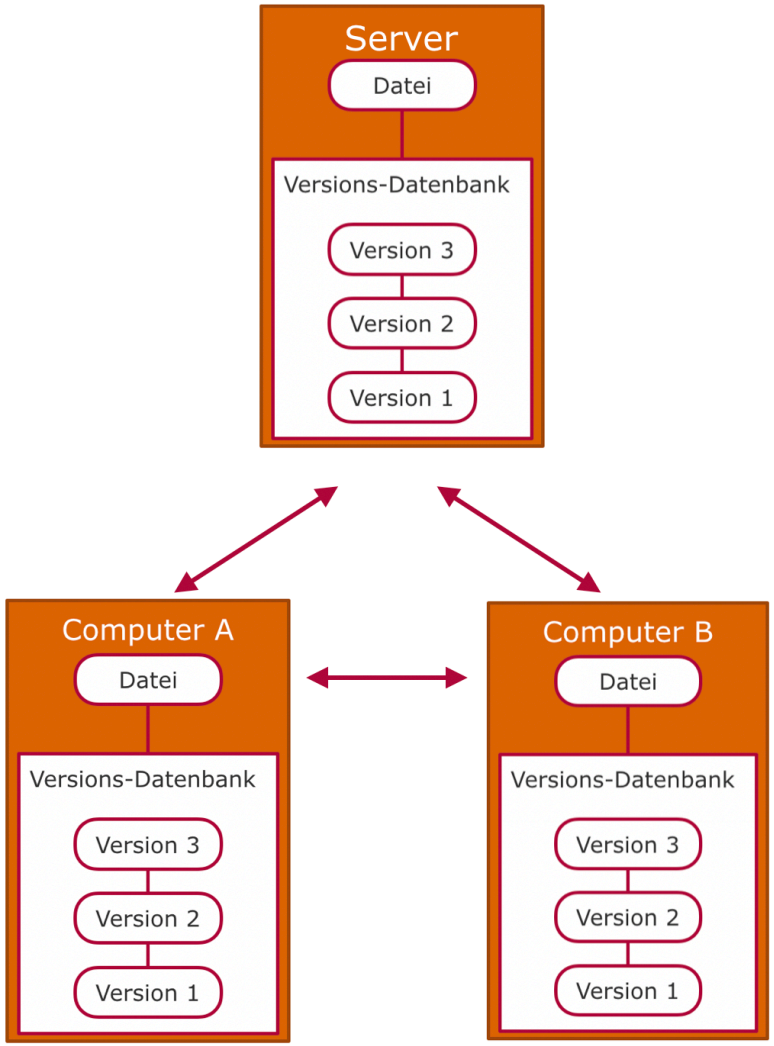
- Unterschiede zwischen Versionen (Patch-Sets) werden gespeichert
- Patch-Sets können kombiniert werden, um die Datei eines Zeitpunktes wiederherzustellen

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

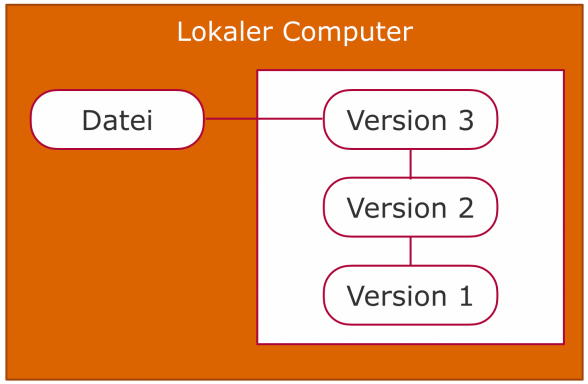
Formen der Versionskontrolle: Verteilte Versionskontrolle



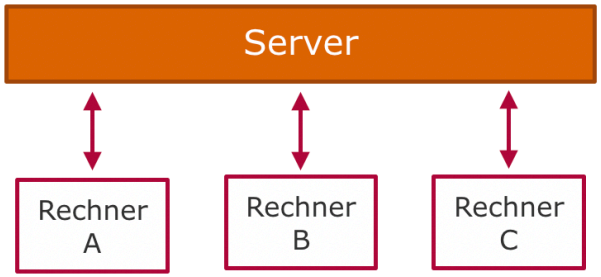
- Klienten spiegeln das Repository mit seiner vollständigen Historie
- Kein Single Point of Failure

Formen der Versionskontrolle

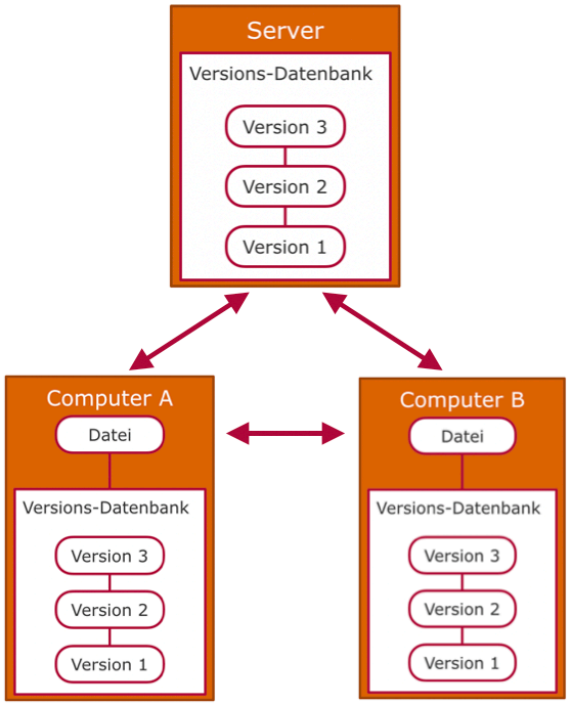
Lokale Versionskontrolle



Zentralisierte Versionskontrolle



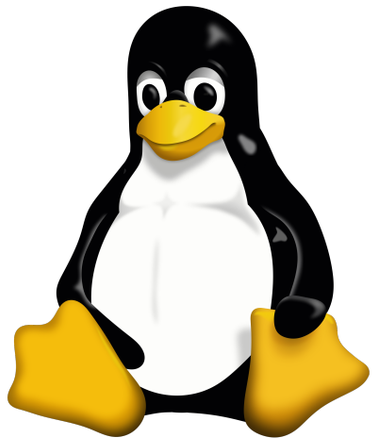
Verteilte Versionskontrolle

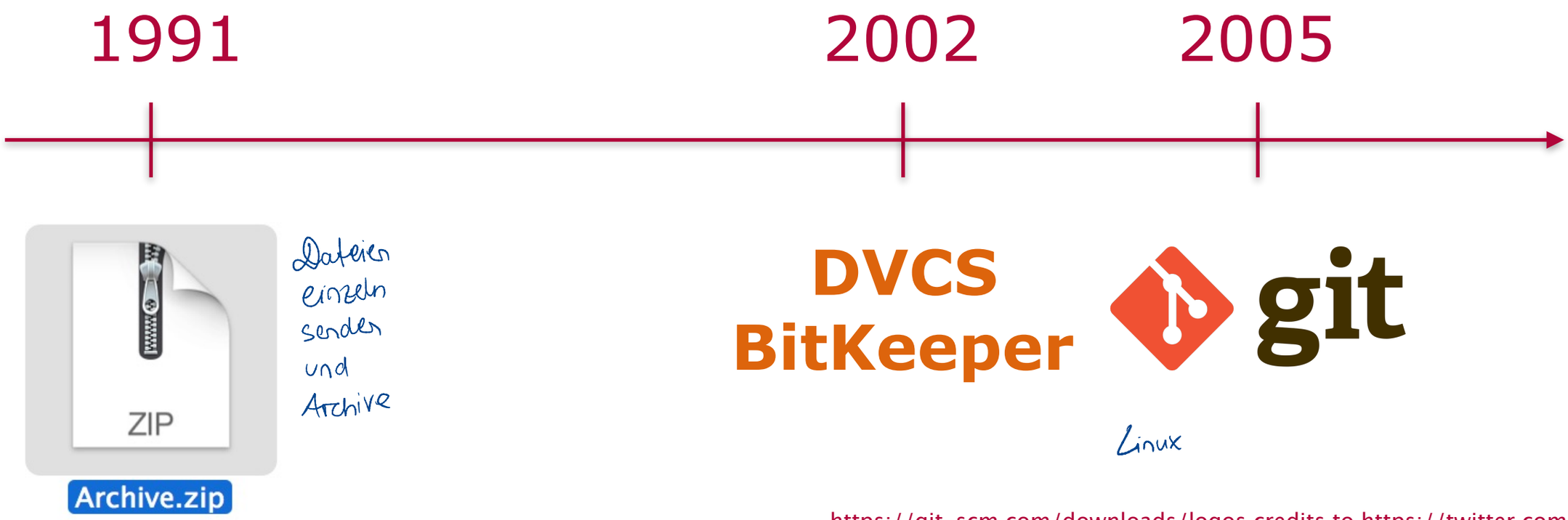


Die Entstehungsgeschichte von Git

Caterina Mandel
Udo Pigorsch

Die Entstehungsgeschichte von Git



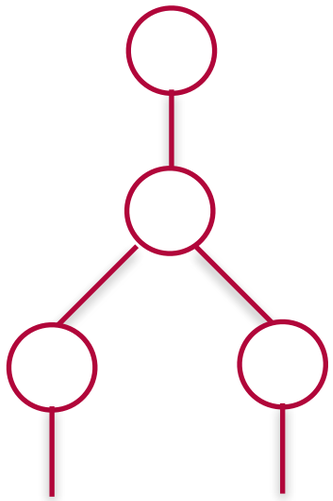


https://git-scm.com/downloads/logos credits to https://twitter.com/jasonlong

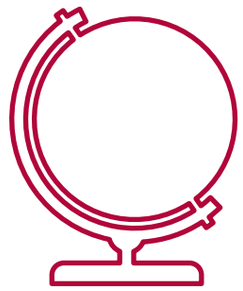
https://isc.tamu.edu/~lewing/linux/

Anforderungen der Entwickler an Git

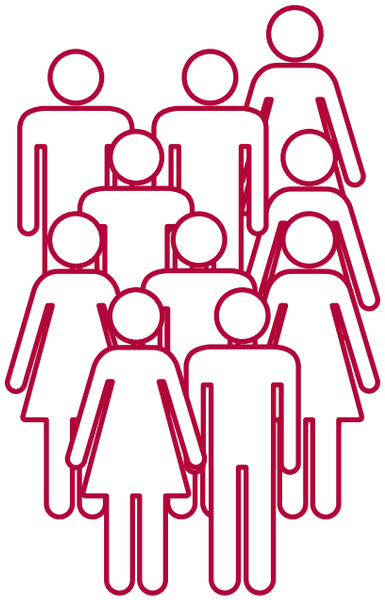
Geschwindigkeit
u. intuitive Bedienung



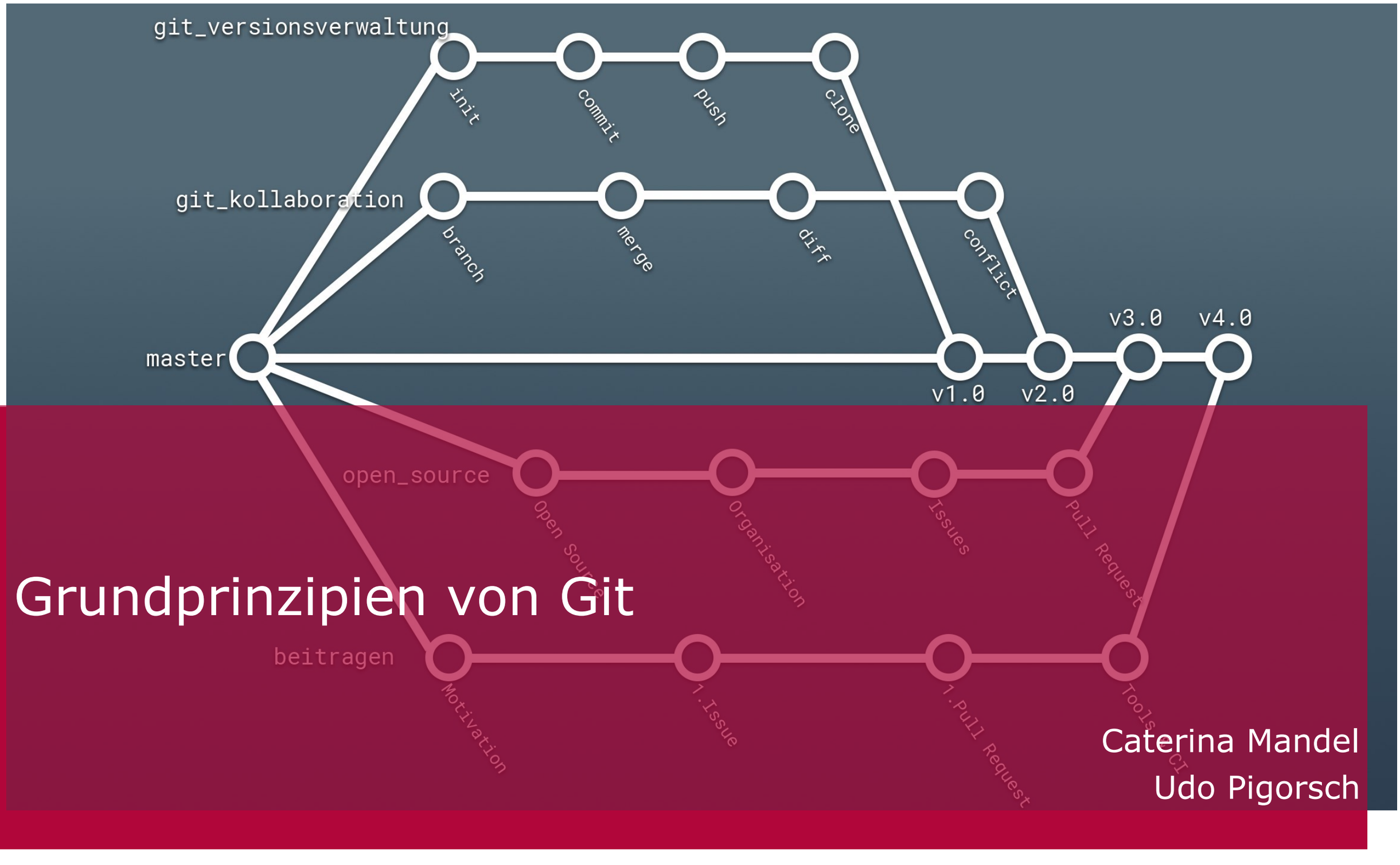
Nicht lineare
Entwicklung



Global



Große Teams



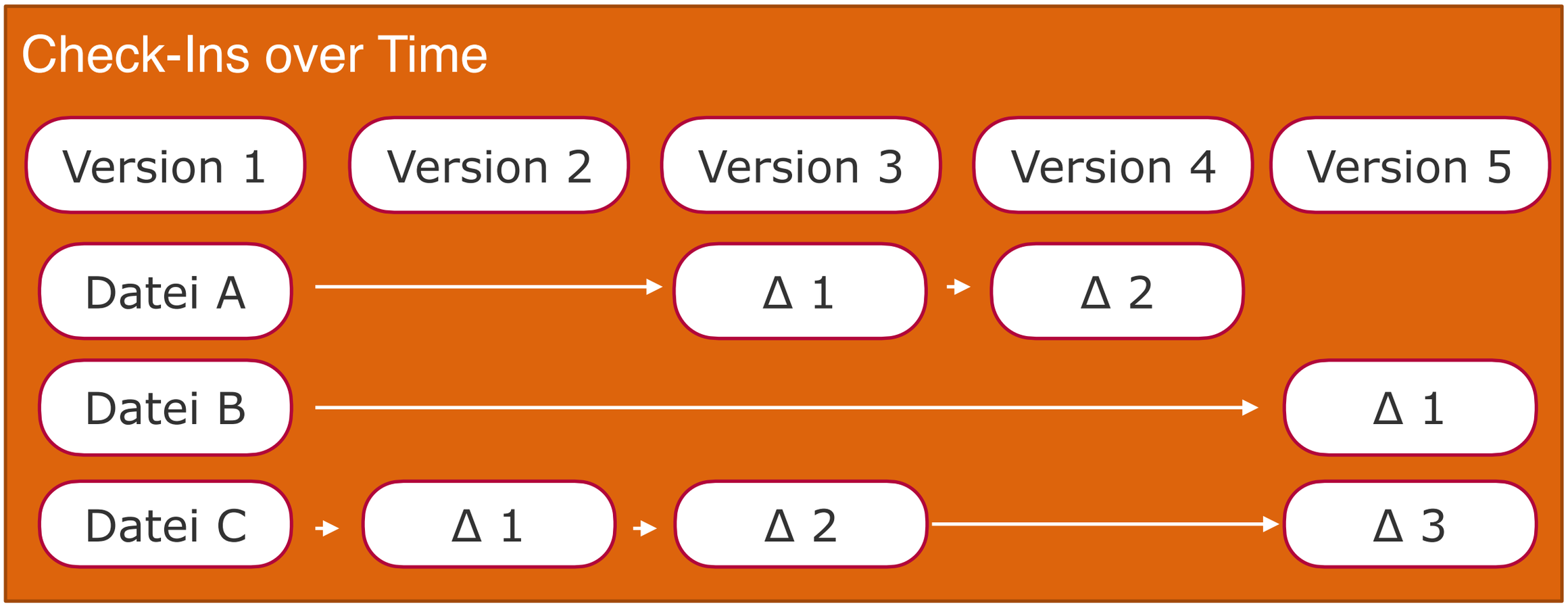
Grundprinzipien von Git

Caterina Mandel
Udo Pigorsch

Grundprinzipien von Git

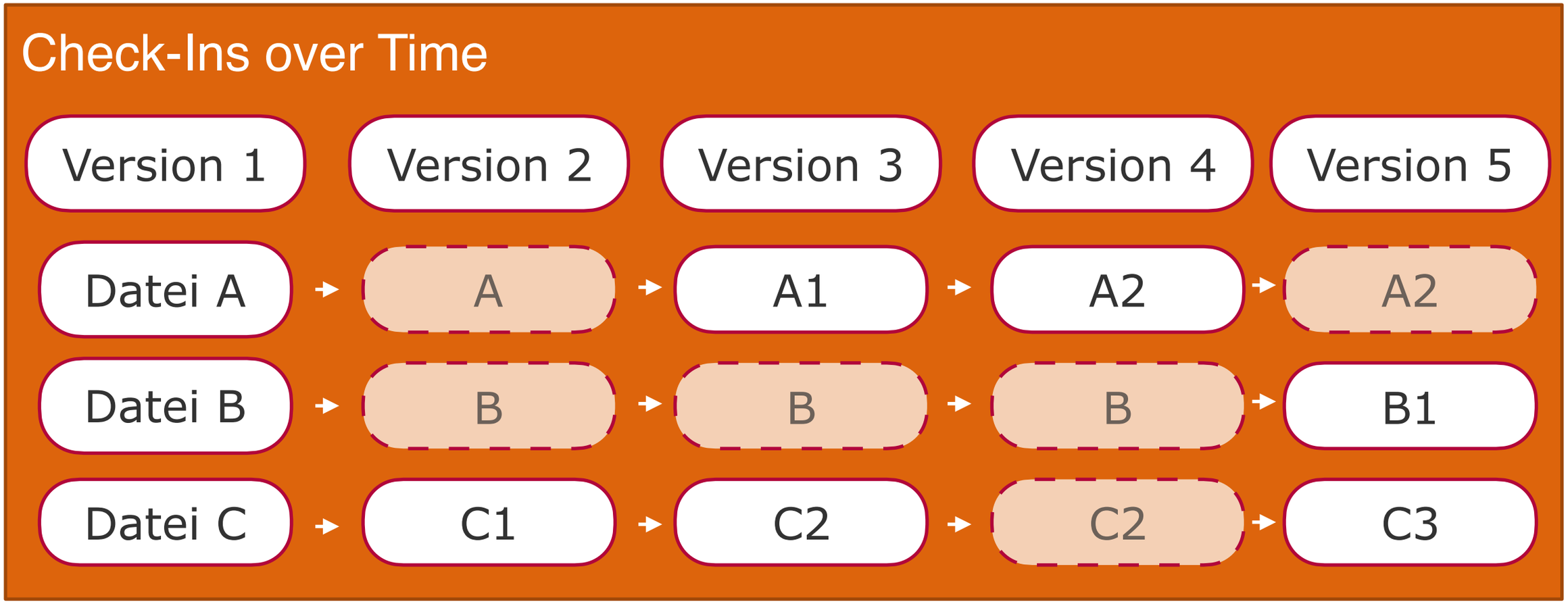
1. Snapshots, statt Unterschiede
2. Fast jede Operation ist lokal
3. Integrität
4. Git fügt Daten hinzu
5. Die drei Zustände

Snapshots vs. Unterschiede



Snapshots vs. Unterschiede

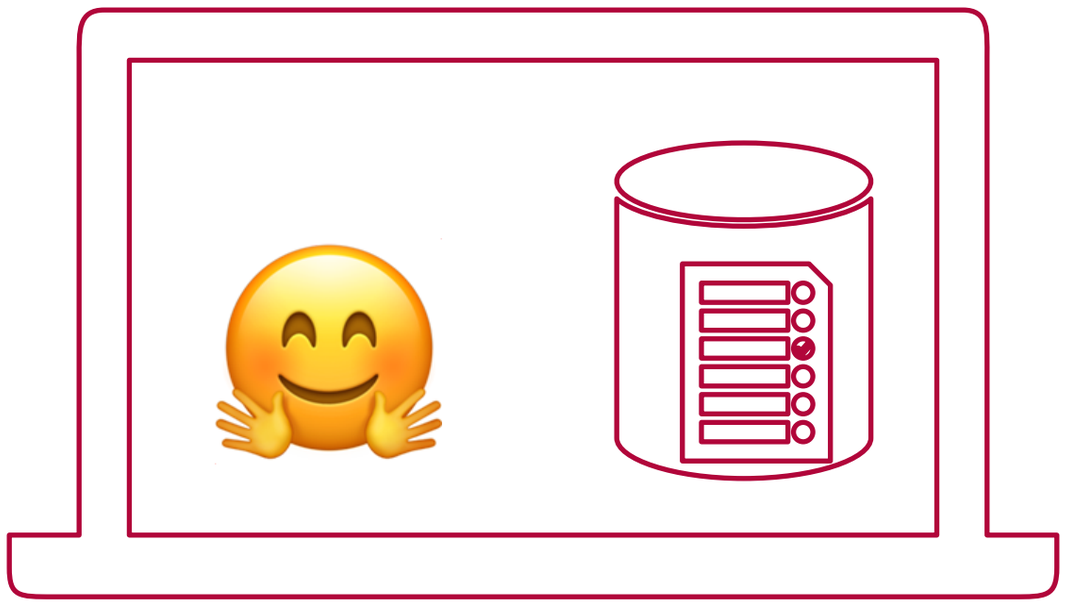
Snapshots vs. Unterschiede



Grundprinzipien von Git

1. Snapshots, statt Unterschiede ✓
2. Fast jede Operation ist lokal
3. Integrität
4. Git fügt Daten hinzu
5. Die drei Zustände

Fast jede Operation ist lokal



Grundprinzipien von Git

1. Snapshots, statt Unterschiede ✓
2. Fast jede Operation ist lokal ✓
3. Integrität
4. Git fügt Daten hinzu
5. Die drei Zustände

Integrität

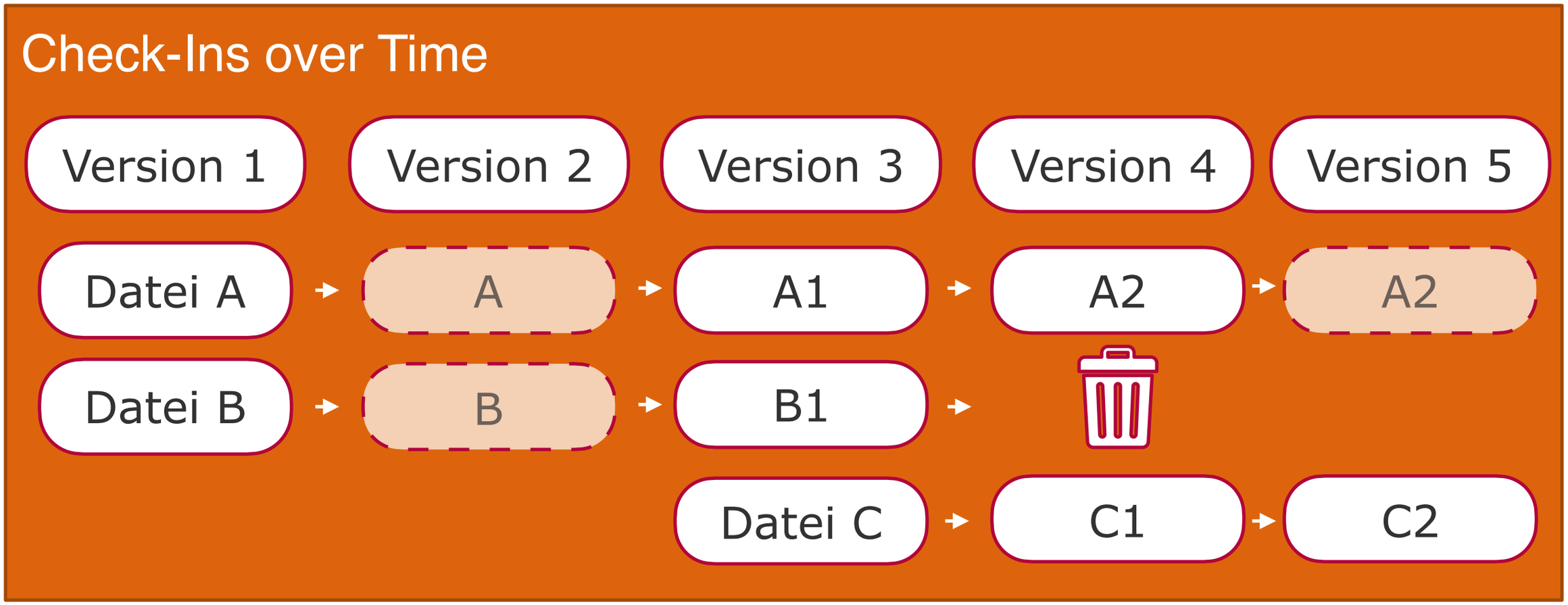
24b9da6552252987

24b9da65522525as

Grundprinzipien von Git

1. Snapshots, statt Unterschiede ✓
2. Fast jede Operation ist lokal ✓
3. Integrität ✓
4. Git fügt Daten hinzu
5. Die drei Zustände

Snapshots vs. Unterschiede



Grundprinzipien von Git

1. Snapshots, statt Unterschiede ✓
2. Fast jede Operation ist lokal ✓
3. Integrität ✓
4. Git fügt Daten hinzu ✓
5. Die drei Zustände

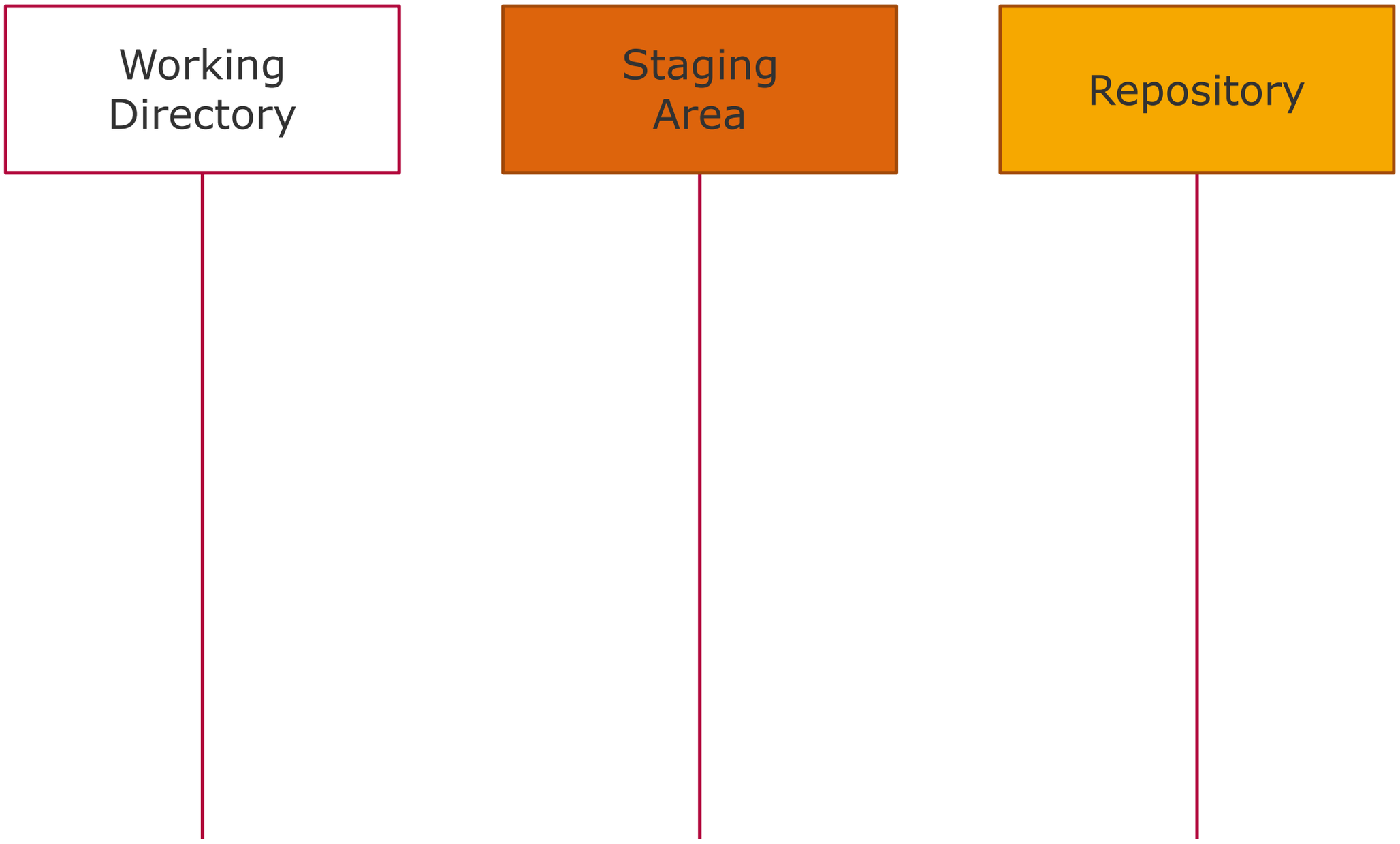
Die drei Zustände

Modified

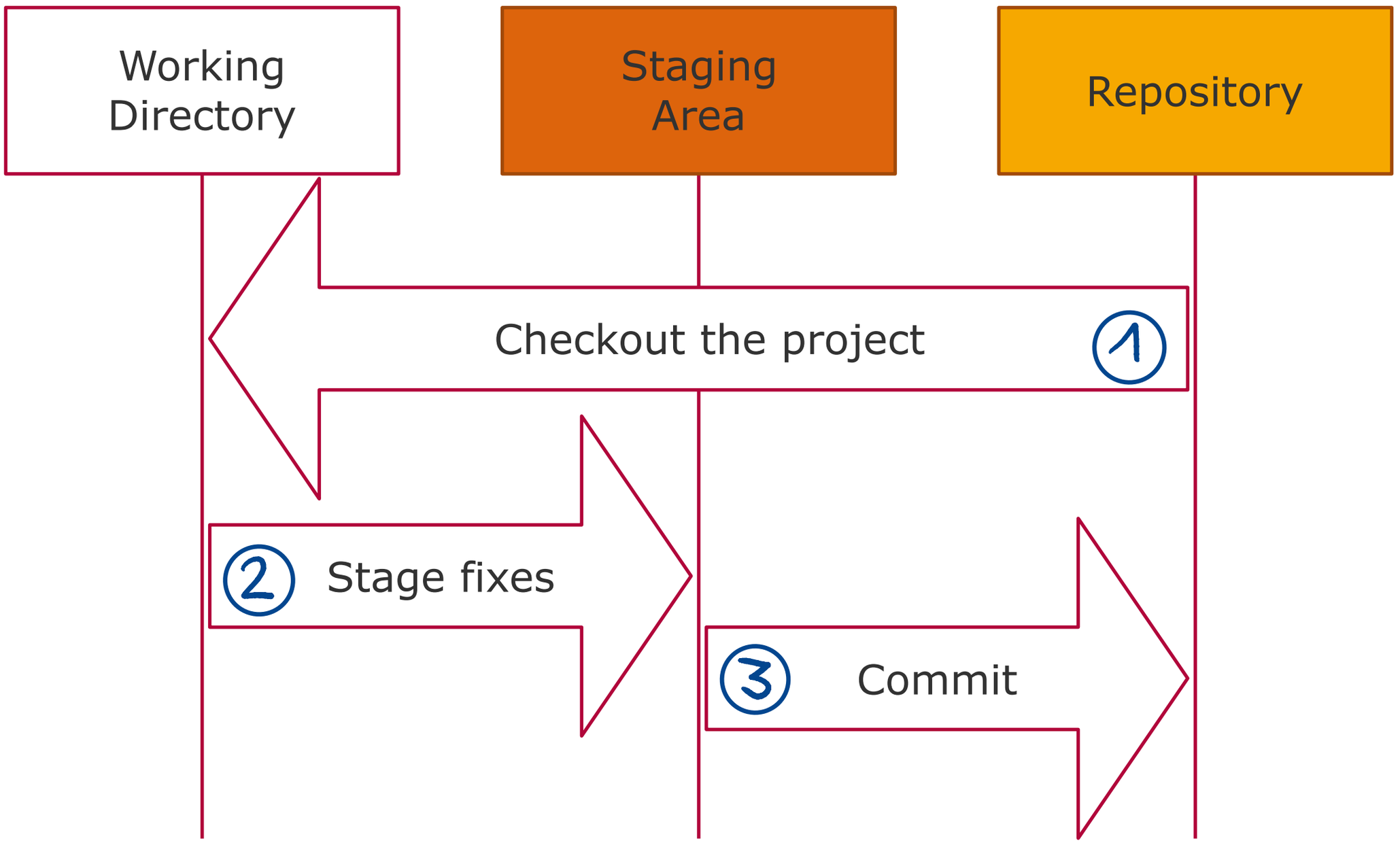
Staged

Committed

Die drei States



Die drei States



Git cheat sheet

Git verstehen

Snapshots, lokale, Integrität, fügt Daten hinzu

Modified ↔ Staged ↔ Commited

Git installieren

Paketmanager bzw. https://git-scm.com/download

Erstelle ein leeres Git-Repository / Reinitialisiere ein vorhandenes

$ git init

Datei zum Index hinzufügen

$ git add

Datei aus dem Arbeitsbaum und aus dem Index entfernen

$ git rm

Änderungen am Repository aufzeichnen

$ git commit

Verwalte einen Satz von verfolgten Repositories

$ git remote

Klone ein Repository in ein neues Verzeichnis

$ git clone

Aktualisiere die Remote-Repositories zusammen mit den zugehörigen Objekten

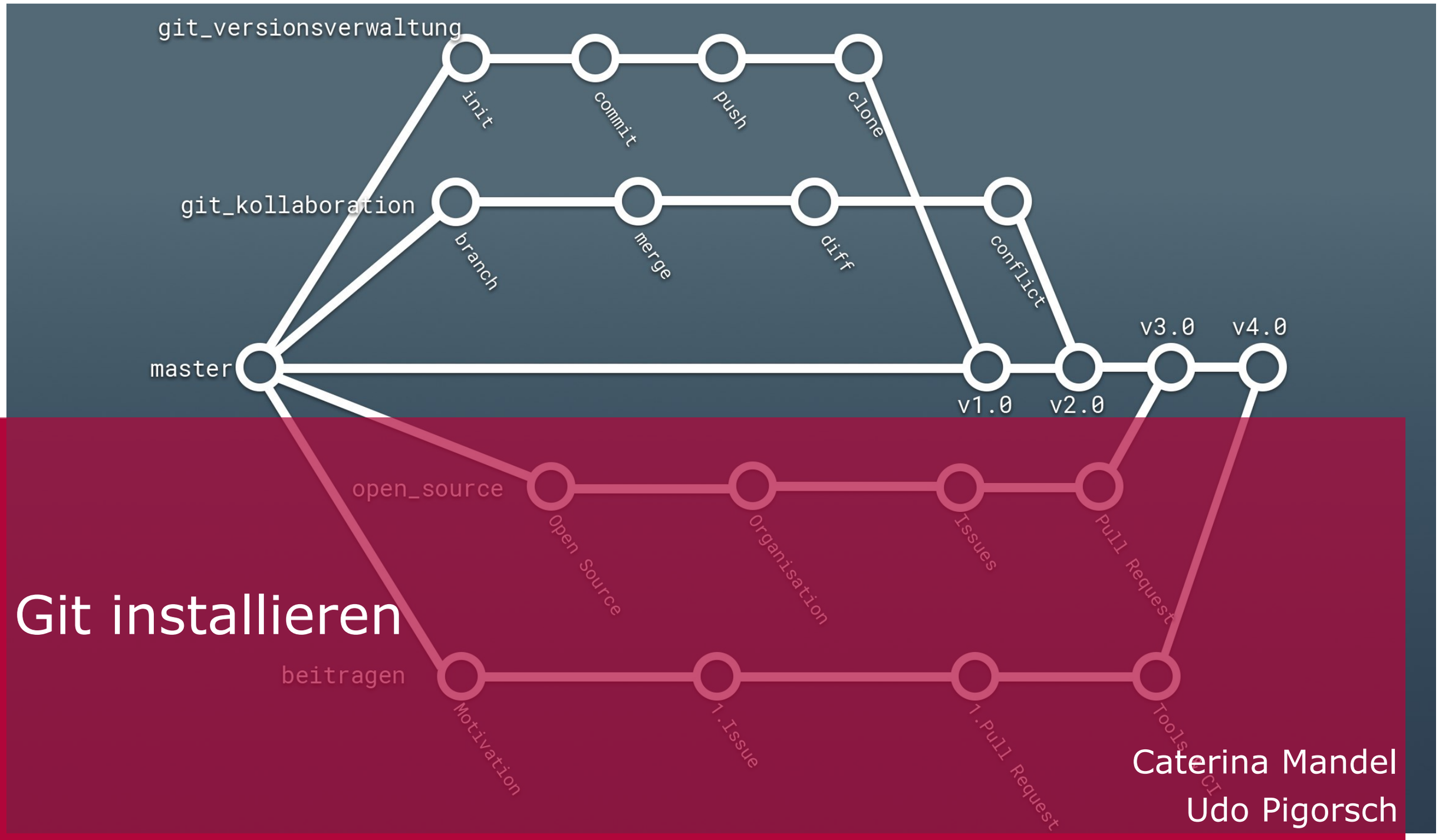
$ git push

Objekte und Referenzen aus einem anderen Repository herunterladen

$ git fetch

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

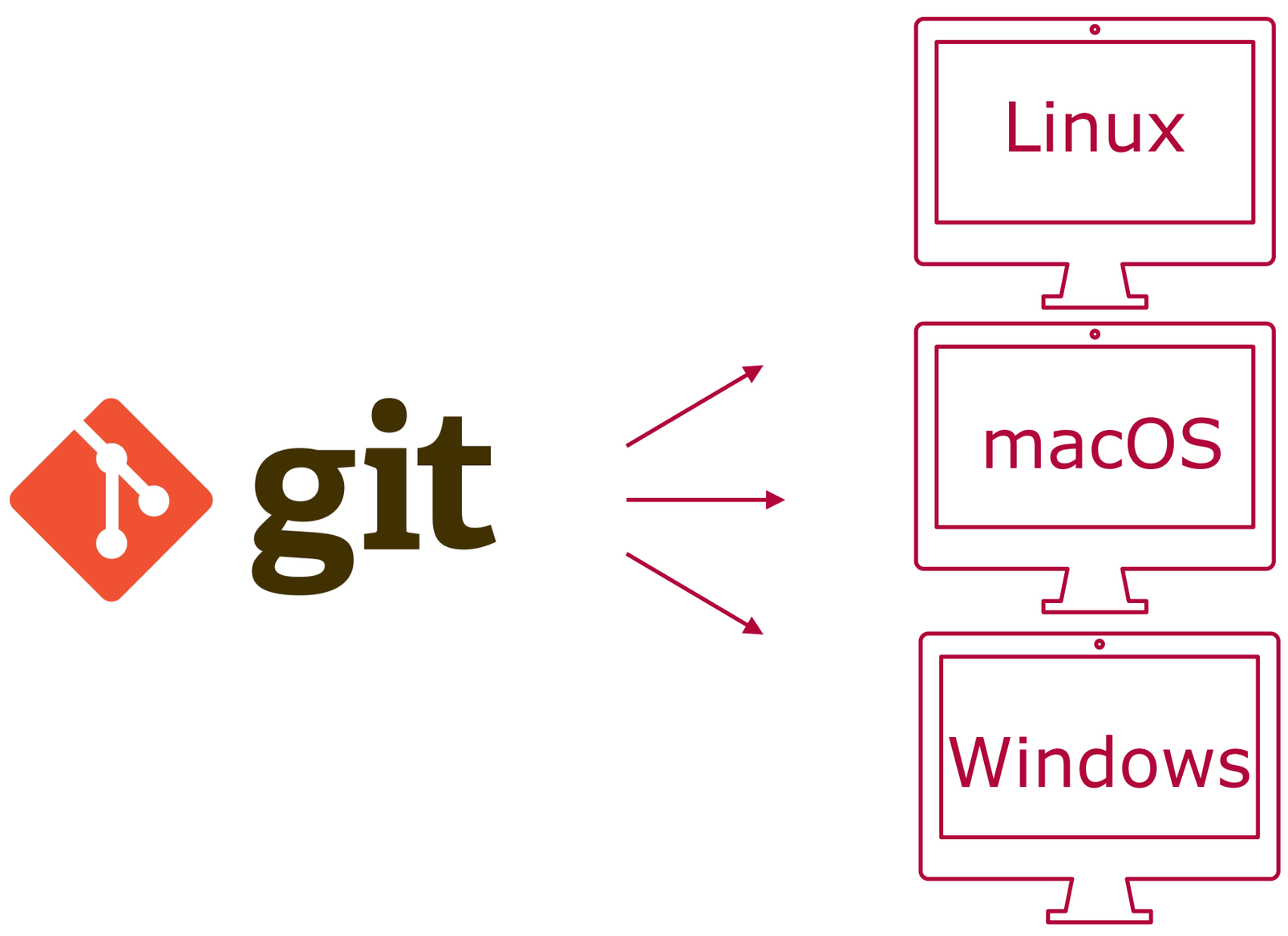
$ git pull



Git installieren

Caterina Mandel
Udo Pigorsch

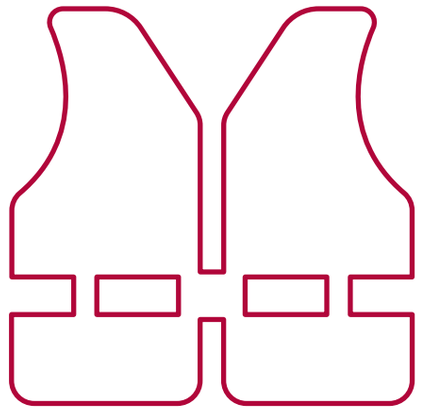
Git installieren



Git installieren

Paketmanager

Installationsassistent



Source Code

selbst kompilieren

01001110

Git für Windows installieren

Binärer Installationsassistent

- https://git-scm.com/download/win
- Vorsicht: Download startet automatisch

Git config

Git an die eigenen Bedürfnisse anpassen

- ~/.gitconfig bzw. ~/.config/git/config
- config Datei im aktuellen Repository

Einstellungen anzeigen

- $ git config --list

Nutzername und E-Mail angeben

- $ git config --global user.name ”Git.Nutzer”
- $ git config --global user.email ”git.nutzer@examplemail.com”

Git config

```
[user]
        name = Pavan Kumar Sunkara
        email = pavan.sss1991@gmail.com
        username = pksunkara
[core]
        editor = vim
        whitespace = fix,-indent-with-non-tab,trailing-space,cr-at-eol
        excludesfile = ~/.gitignore
[sendemail]
        smtpencryption = tls
        smtpserver = smtp.gmail.com
        smtpuser = pavan.sss1991@gmail.com
        smtppass = password
        smtpserverport = 587
[web]
        browser = google-chrome
[instaweb]
        httpd = apache2 -f
[rerere]
        enabled = 1
        autoupdate = 1
[push]
        default = matching
[color]
        ui = auto
[color "branch"]
        current = yellow bold
        local = green bold
        remote = cyan bold
[color "diff"]
        meta = yellow bold
        frag = magenta bold
        old = red bold
        new = green bold
        whitespace = red reverse
```

Git cheat sheet

Git verstehen

Snapshots, lokale, Integrität, fügt Daten hinzu
Modified ↔ Staged ↔ Commited

Git installieren

Paketmanager bzw. https://git-scm.com/download

Erstelle ein leeres Git-Repository / Reinitialisiere ein vorhandenes
$ git init

Datei zum Index hinzufügen
$ git add

Datei aus dem Arbeitsbaum und aus dem Index entfernen
$ git rm

Änderungen am Repository aufzeichnen
$ git commit

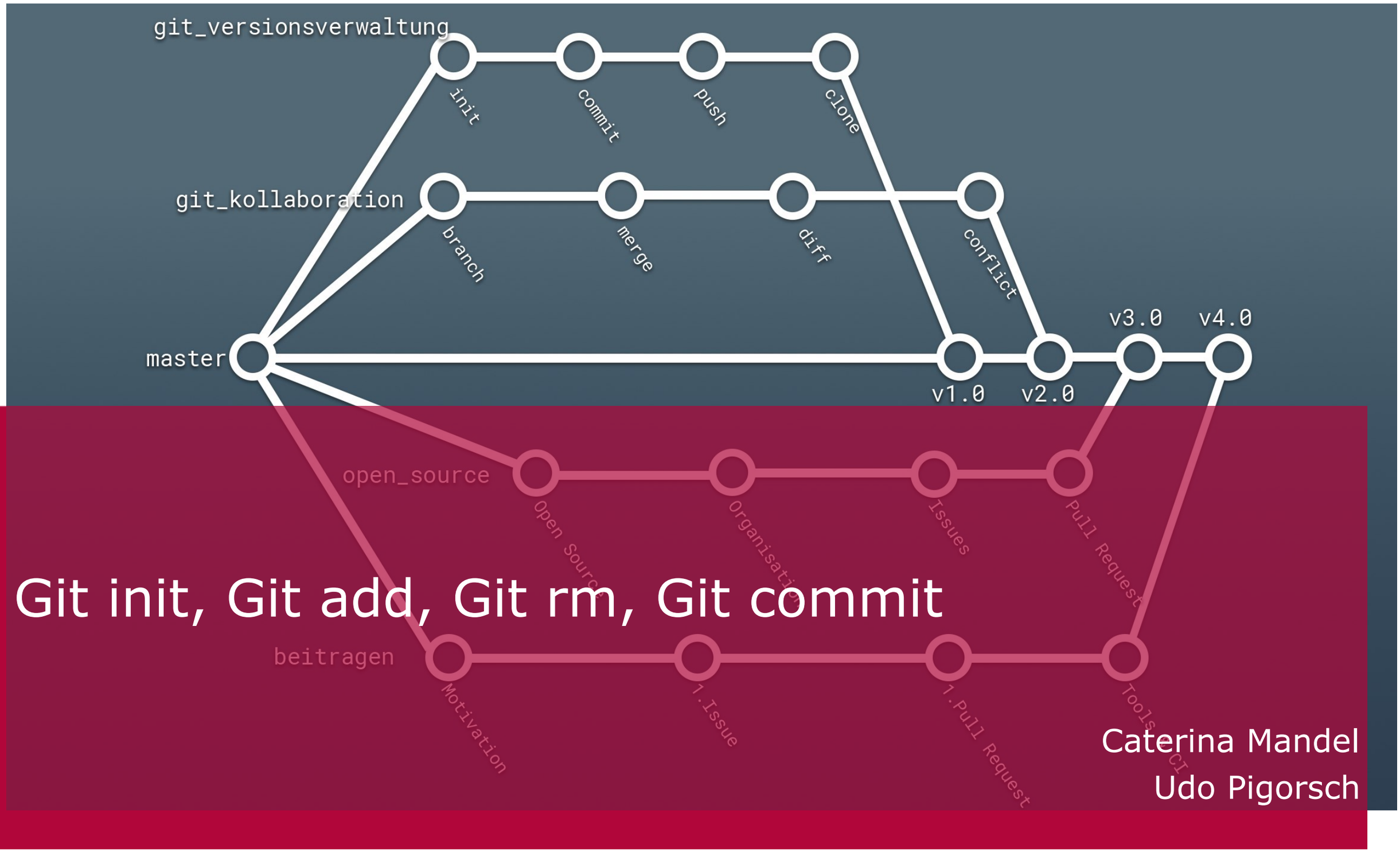
Verwalte einen Satz von verfolgten Repositories
$ git remote

Klone ein Repository in ein neues Verzeichnis
$ git clone

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

Objekte und Referenzen aus einem anderen Repository herunterladen
$ git fetch

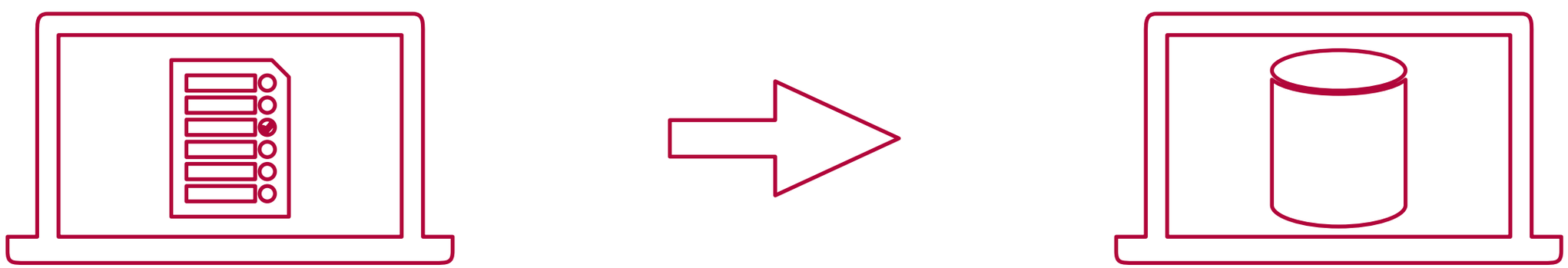
Aus einem anderen Repository oder einem lokalen Zweig holen und in diesen integrieren
$ git pull

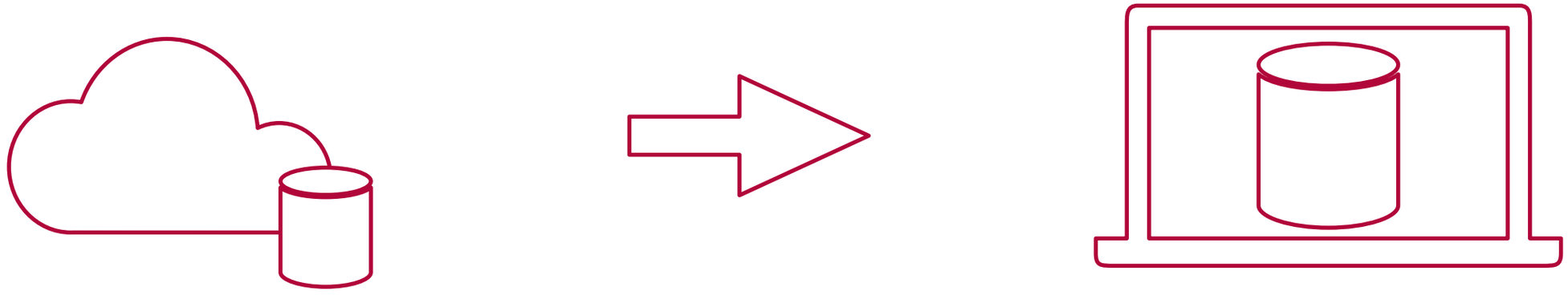


Git init, Git add, Git rm, Git commit

Caterina Mandel
Udo Pigorsch

Git init





Git init

$ cd /Pfad/Projekt

$ git init

Git add

$ git add beispielbild.jpg

$ git add *.jpg

$ git commit

Git commit

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

a) `$ git commit`

oder

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

Bemerkung ergänzt

Git add

$ git add beispielbild.jpg

$ git add *.jpg

$ git commit

$ git remove beispielbild.jpg

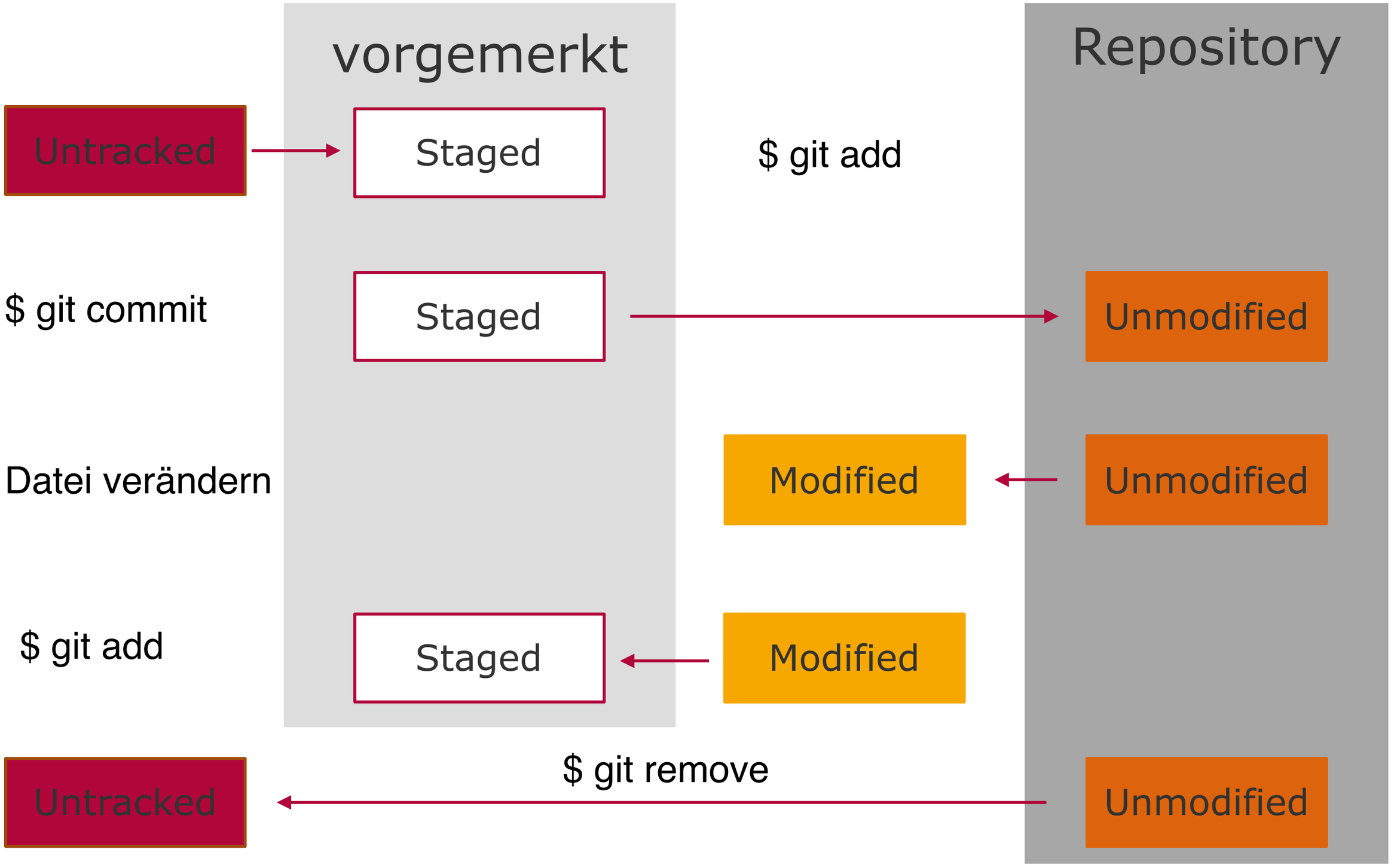
Git commit

vorgemerkt

Repository

Untracked	Staged	Modified	Unmodified
Datei lokal, nicht im Repository	Datei markiert, commit überträgt ins Repository	Datei in älterer Version im Repository, lokale Änderung nicht	Datei im aktuellen Stand im Repository

Git commit


vorgemerkt
Repository
Untracked
Staged
$ git add
$ git commit
Staged
Unmodified
Datei verändern
Modified
Unmodified
$ git add
Staged
Modified
$ git remove
Untracked
Unmodified

Git cheat sheet

Git verstehen

Snapshots, lokale, Integrität, fügt Daten hinzu
Modified ↔ Staged ↔ Commited

Git installieren

Paketmanager bzw. https://git-scm.com/download

Erstelle ein leeres Git-Repository / Reinitialisiere ein vorhandenes

$ git init

Datei zum Index hinzufügen

$ git add

Datei aus dem Arbeitsbaum und aus dem Index entfernen

$ git rm

Änderungen am Repository aufzeichnen

$ git commit

Verwalte einen Satz von verfolgten Repositories

$ git remote

Klone ein Repository in ein neues Verzeichnis

$ git clone

Aktualisiere die Remote-Repositories zusammen mit den zugehörigen Objekten

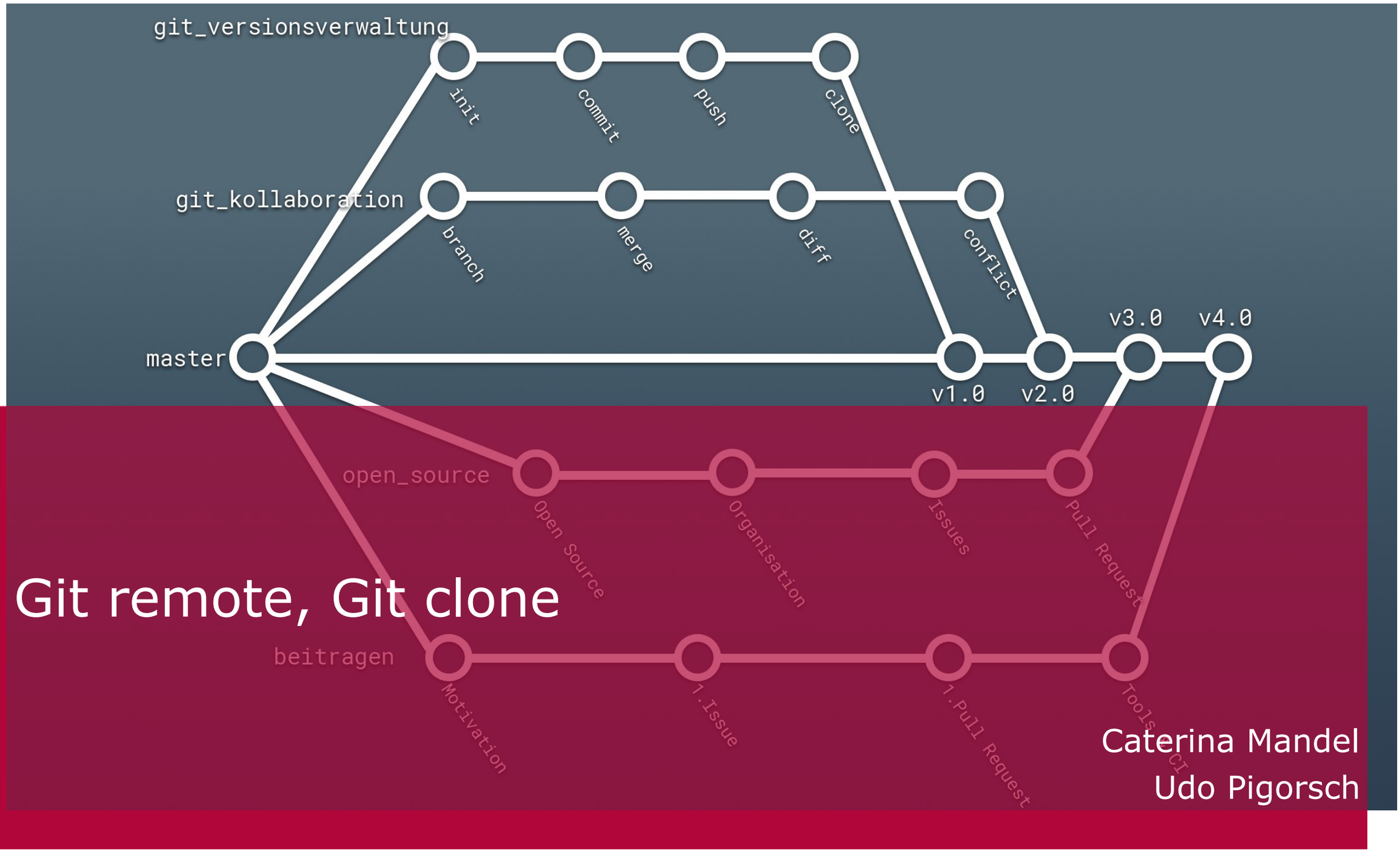
$ git push

Objekte und Referenzen aus einem anderen Repository herunterladen

$ git fetch

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

$ git pull

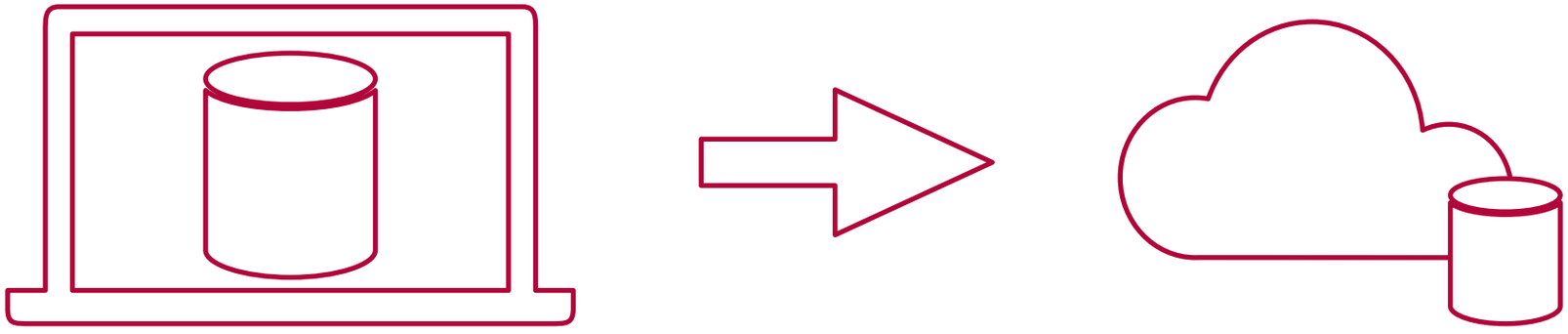


Git remote, Git clone

Caterina Mandel
Udo Pigorsch

Git remote

$ git remote



Git remote

$ git remote add <alias><url>

$ git remote add repo https://github.com/repo

$ git remote rename <alias><neues alias>

$ git remote rename repo projekt1

$ git remote remove <alias>

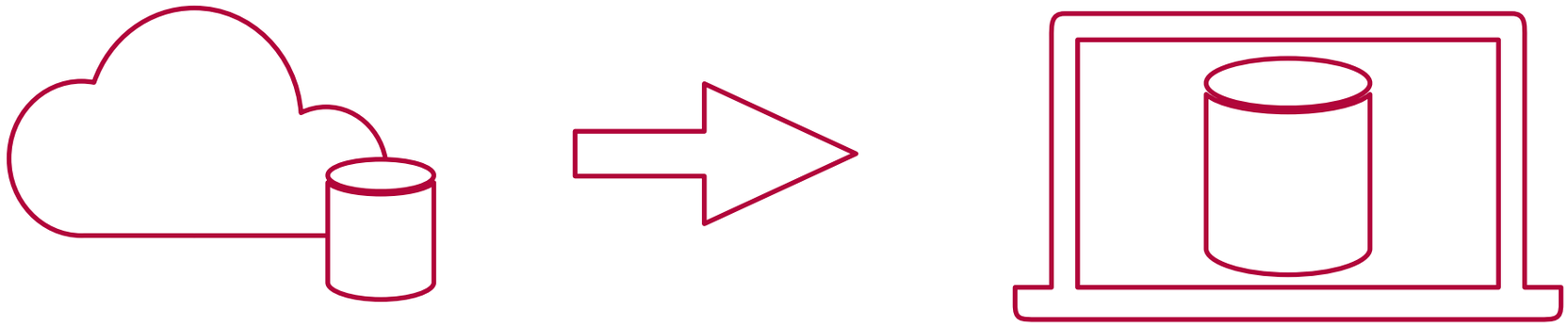
oder: $ git remote rm <alias>

$ git remote rm projekt1

Git clone

$ git clone <url> <alias>

$ git clone https://github.com/repo repo



- Vollständige Kopie fast aller Daten
- Versionierung jeder Datei

Git cheat sheet

Git verstehen:

Snapshots, lokale, Integrität, fügt Daten hinzu
Modified ↔ Staged ↔ Commited

Git installieren:

Paketmanager bzw. https://git-scm.com/download

Erstelle ein leeres Git-Repository / Reinitialisiere ein vorhandenes

$ git init

Datei zum Index hinzufügen

$ git add

Datei aus dem Arbeitsbaum und aus dem Index entfernen

$ git rm

Änderungen am Repository aufzeichnen

$ git commit

Verwalte einen Satz von verfolgten Repositories

$ git remote

Klone ein Repository in ein neues Verzeichnis

$ git clone

Aktualisiere die Remote-Repositories zusammen mit den zugehörigen Objekten

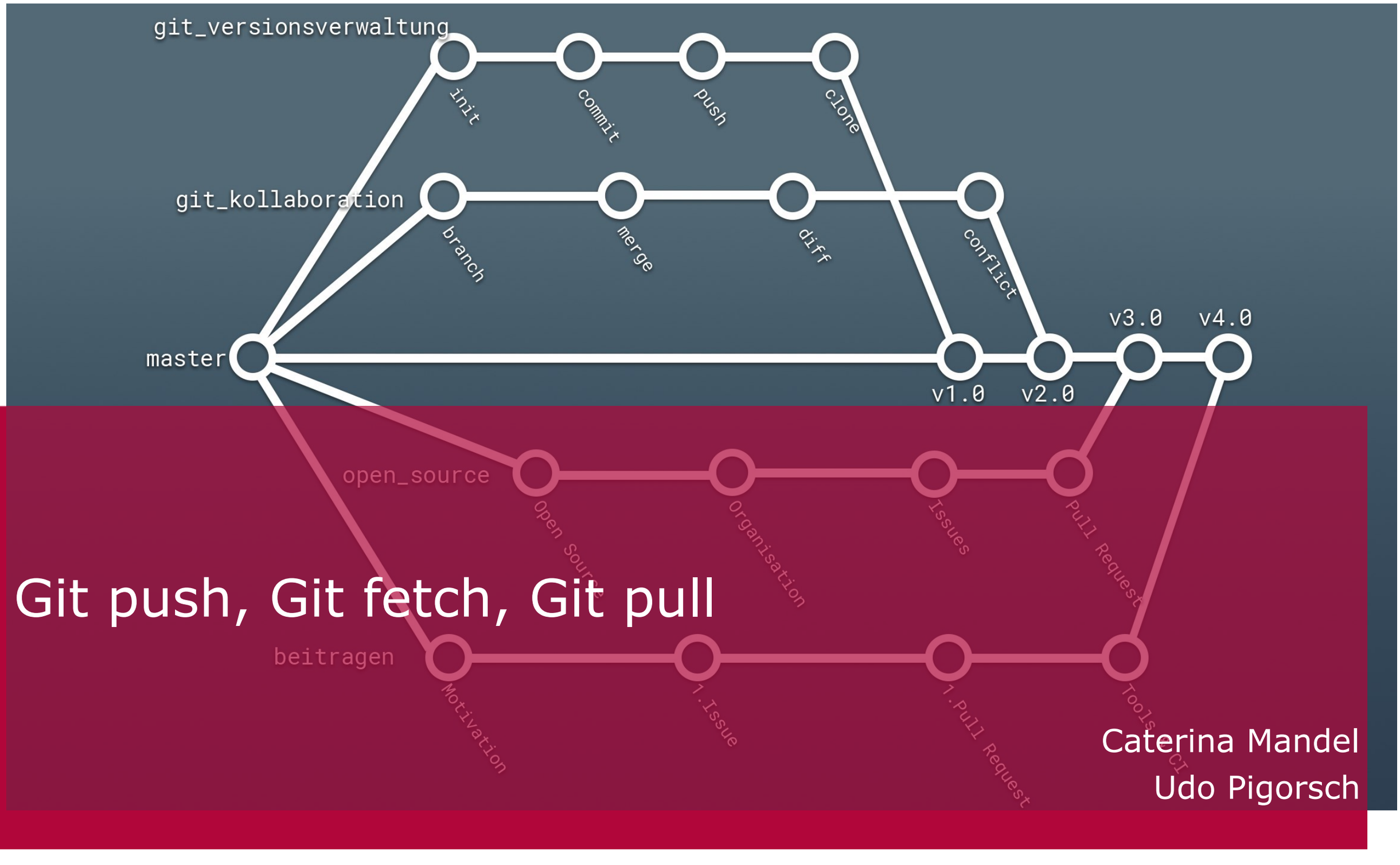
$ git push

Objekte und Referenzen aus einem anderen Repository herunterladen

$ git fetch

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

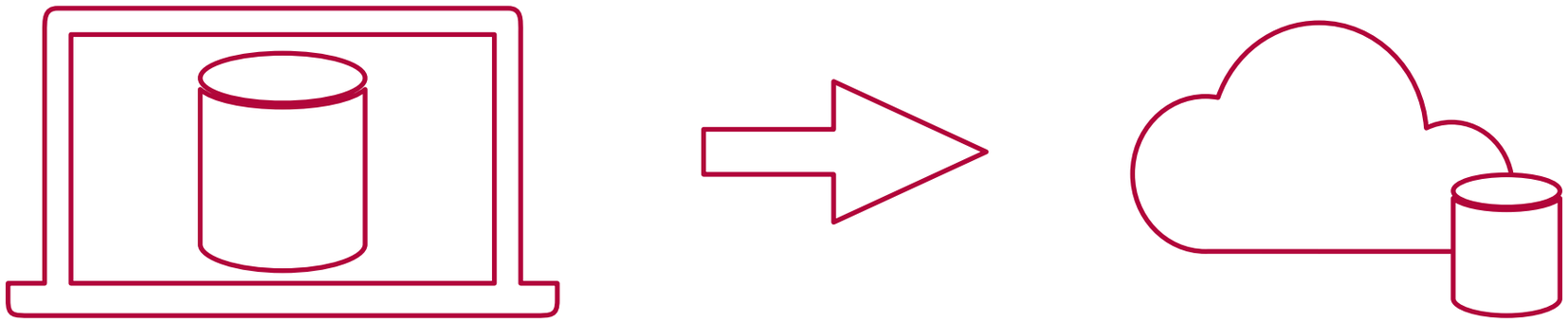
$ git pull



Git push, Git fetch, Git pull

Caterina Mandel
Udo Pigorsch

Git push

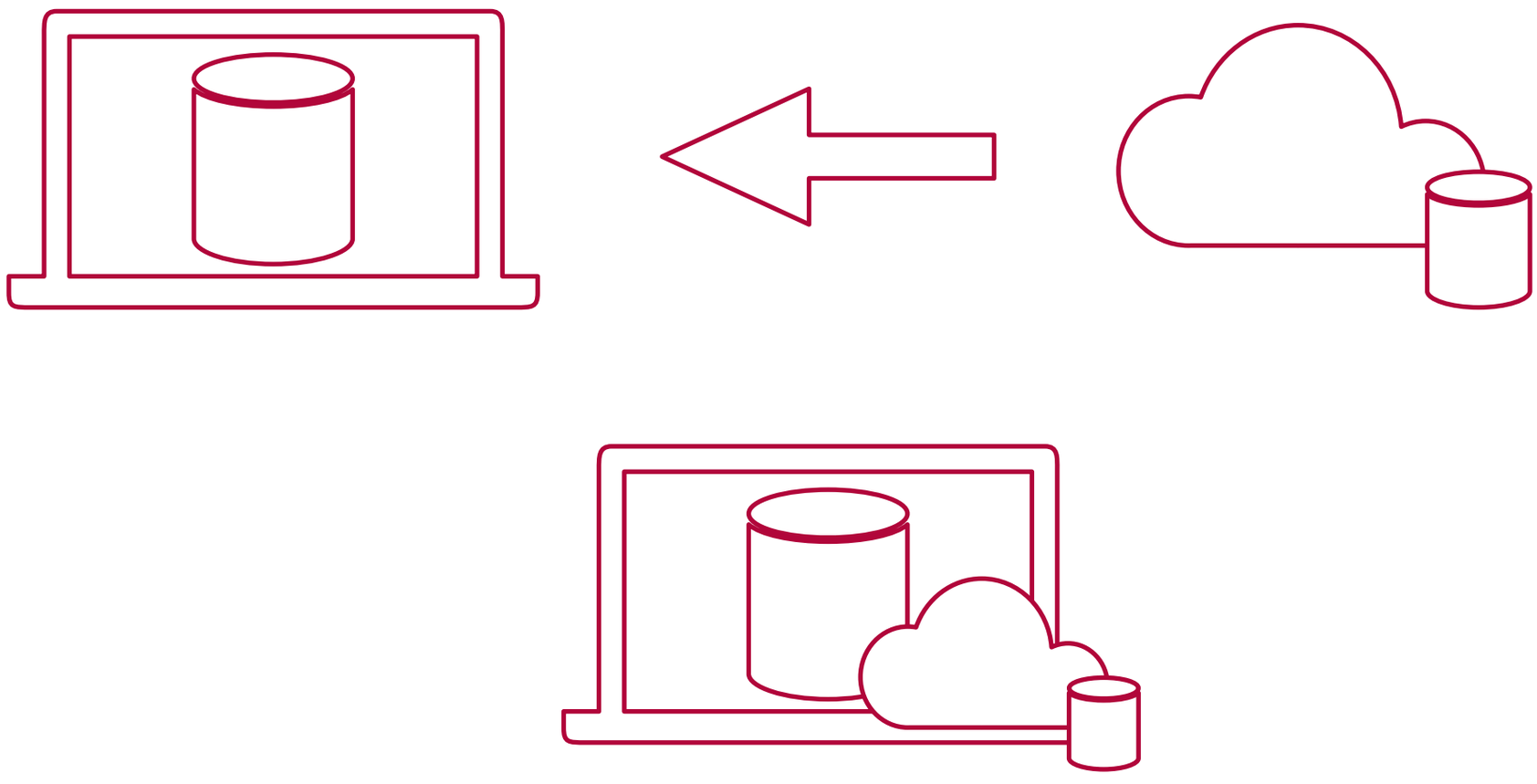


Git push

$ git push origin/<branch>

$ git push origin/master

Git fetch

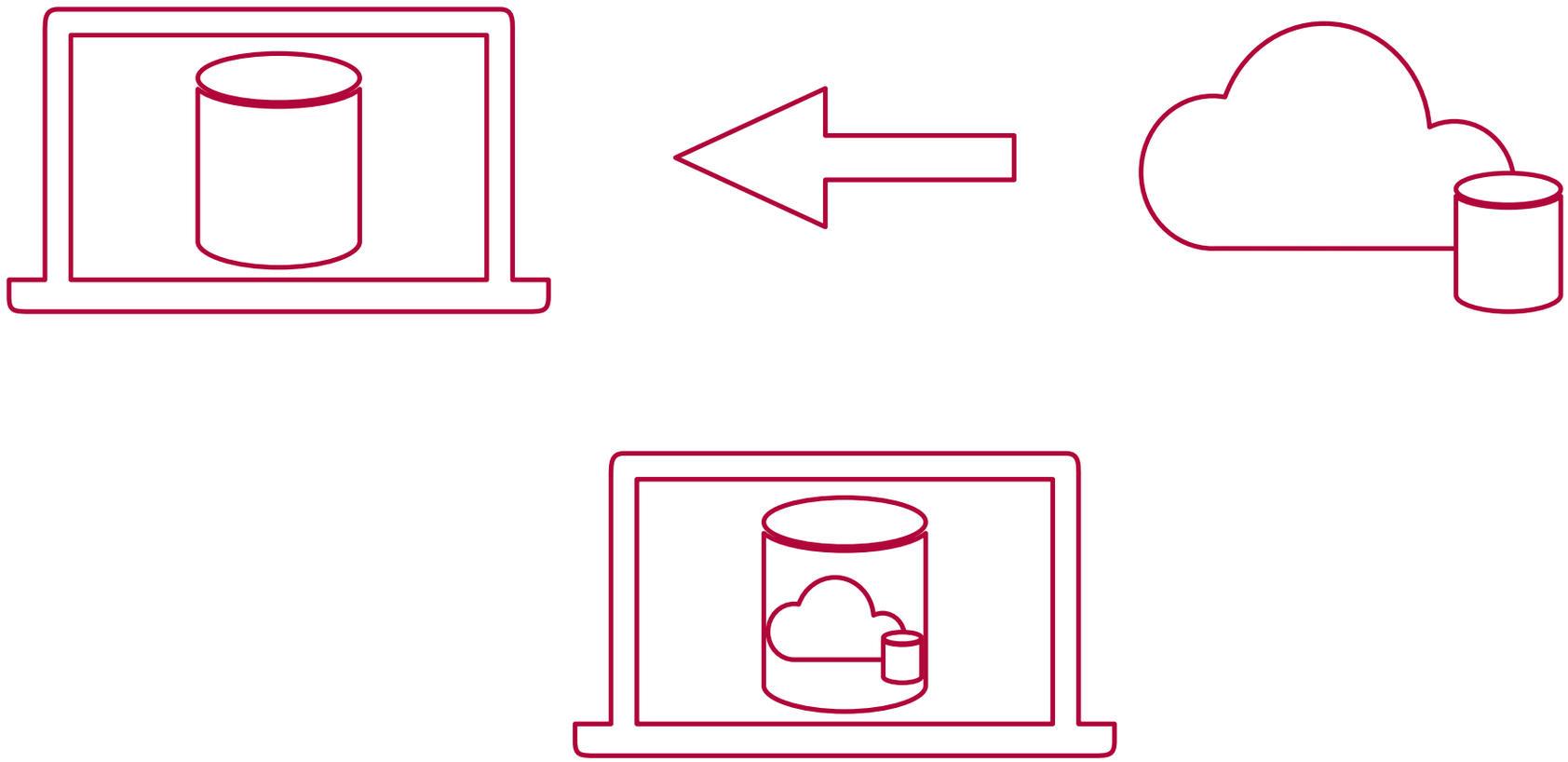


Git fetch

$ git fetch origin

$ git merge origin/<branch>

Git pull



Git pull

$ git pull =

$ git fetch +

$ git merge

Git cheat sheet

Hasso Plattner Institut

Git verstehen:

Snapshots, lokale, Integrität, fügt Daten hinzu
Modified ↔ Staged ↔ Commited

Git installieren:

Paketmanager bzw. https://git-scm.com/download

Erstelle ein leeres Git-Repository / Reinitialisiere ein vorhandenes

$ git init

Datei zum Index hinzufügen

$ git add

Datei aus dem Arbeitsbaum und aus dem Index entfernen

$ git rm

Änderungen am Repository aufzeichnen

$ git commit

Verwalte einen Satz von verfolgten Repositories

$ git remote

Klone ein Repository in ein neues Verzeichnis

$ git clone

Aktualisiere die Remote-Repositories zusammen mit den zugehörigen Objekten

$ git push

Objekte und Referenzen aus einem anderen Repository herunterladen

$ git fetch

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

$ git pull