Agile Webentwicklung mit Ruby on Rails

Prof. Dr. Oliver Vornberger
Nils Haldenwang, B.Sc.

UNIVERSITÄT OSNABRÜCK

Institut für Informatik
AG Medieninformatik

Institut für Informatik
Prof. Dr. Oliver Vornberger
Nils Haldenwang, B.Sc.

Universität Osnabrück
http://www-lehre.inf.uos.de/~ror
10.05.2012

Agile Webentwicklung mit Ruby on Rails

Sommersemester 2012

Blatt 3

Aufgabe 1: Vector Refactoring

Betrachten Sie die vom letzten Übungsblatt bekannte (und hier beiliegende) Datei `vector.rb`. Führen Sie ein geeignetes Refactoring der Klasse durch.

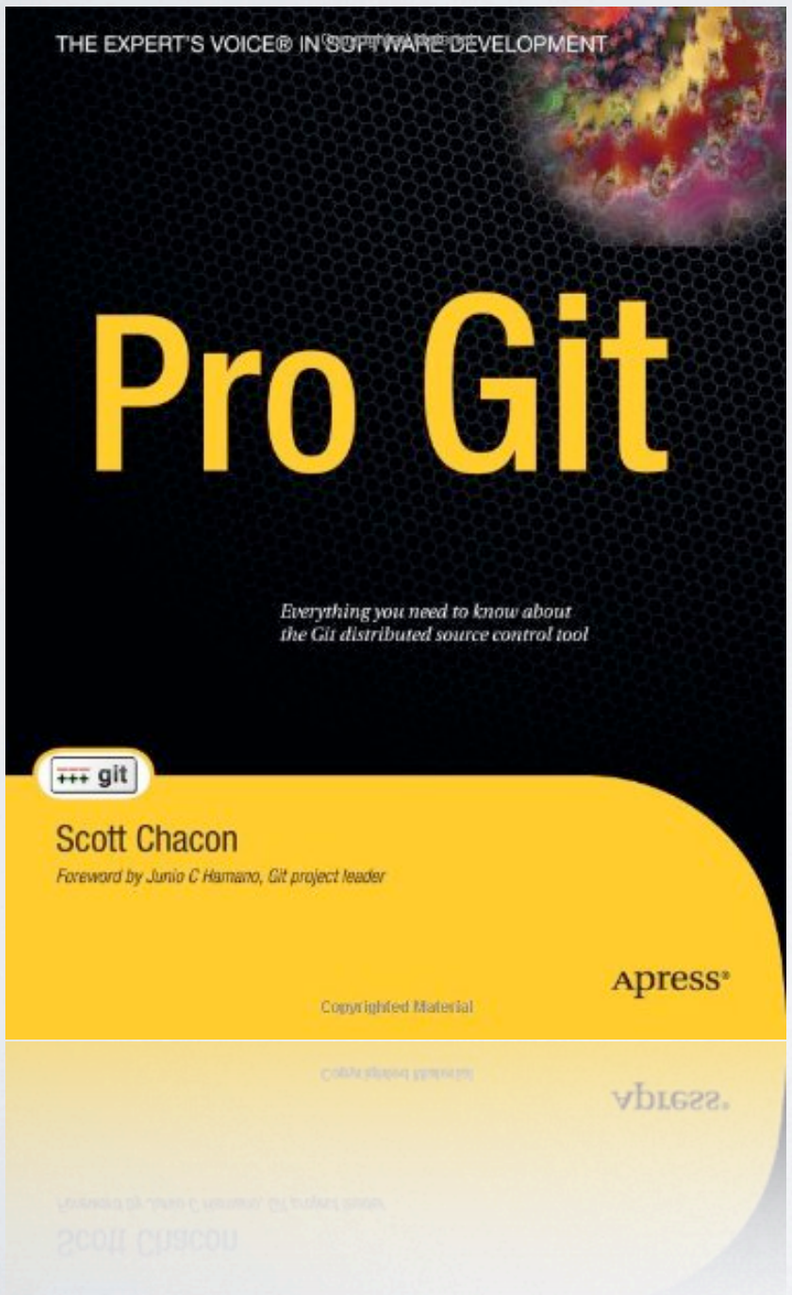

git

Historie

- erste Version veröffentlicht am 7. April 2005 von Linus Torvalds
- aktuelle Version 1.7.10.1 veröffentlicht am 2. Mai 2012
- entwickelt zur Verwaltung des Linux Kernels mit den Zielen:
  - Geschwindigkeit
  - einfaches Design
  - verteiltes System
  - Unterstützung für nicht-lineare Entwicklung
  - effiziente Verwaltung großer Projekte

Firmen und Projekte, die Git einsetzen

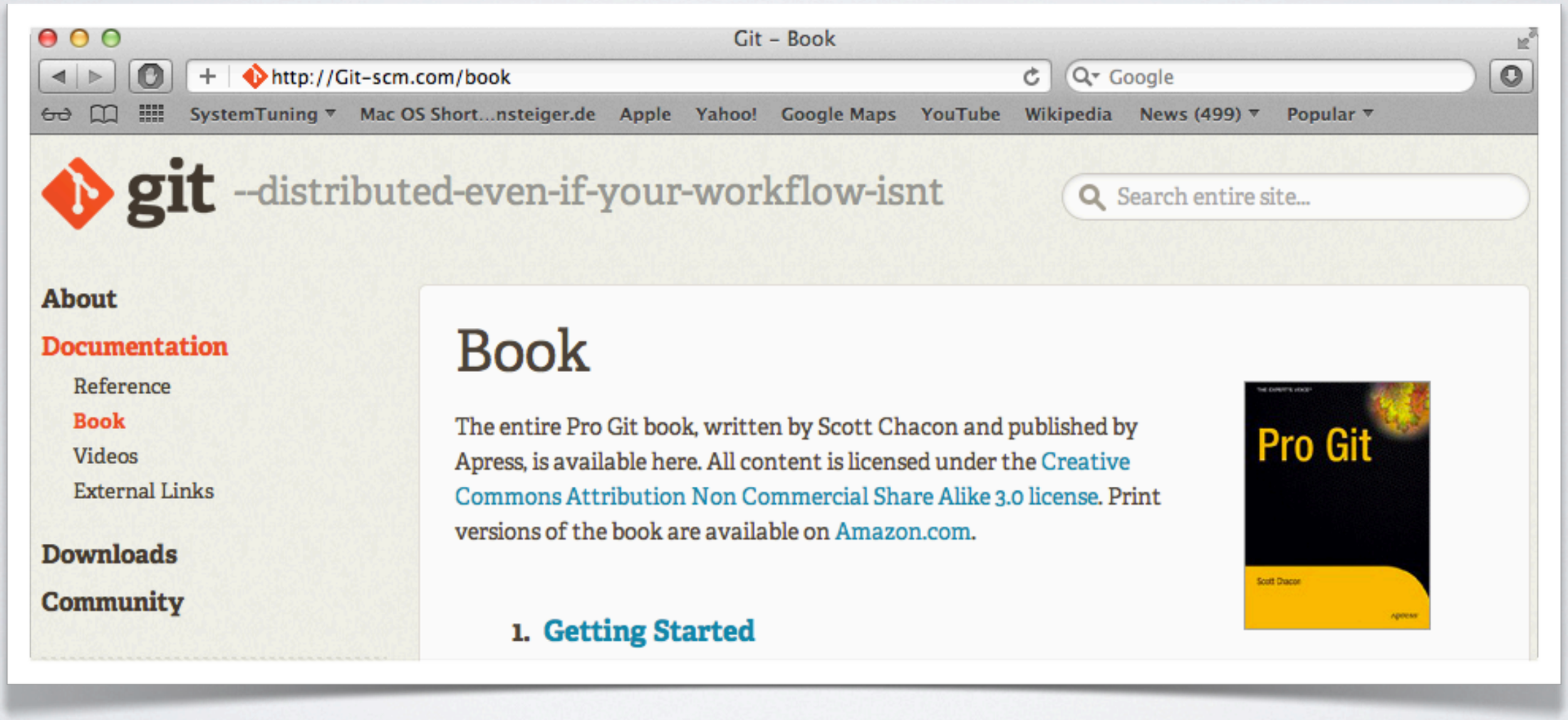
Quellen / Literatur



S. Chacon,
Pro Git,
Apress, 2009

Quellen / Literatur

http://www.git-scm.com



Versionsverwaltung

Was ist Versionsverwaltung?

- Protokollierung von Änderungen an Dokumenten oder Dateien mit Zeitstempel und Autor
- Archivierung von Zwischenständen der Arbeit
- Wiederherstellung alter Zustände
- Koordinierung des Zugriffs mehrerer Entwickler
- Entwicklung mehrerer Zweige eines Projektes

Vorteile von Versionskontrolle

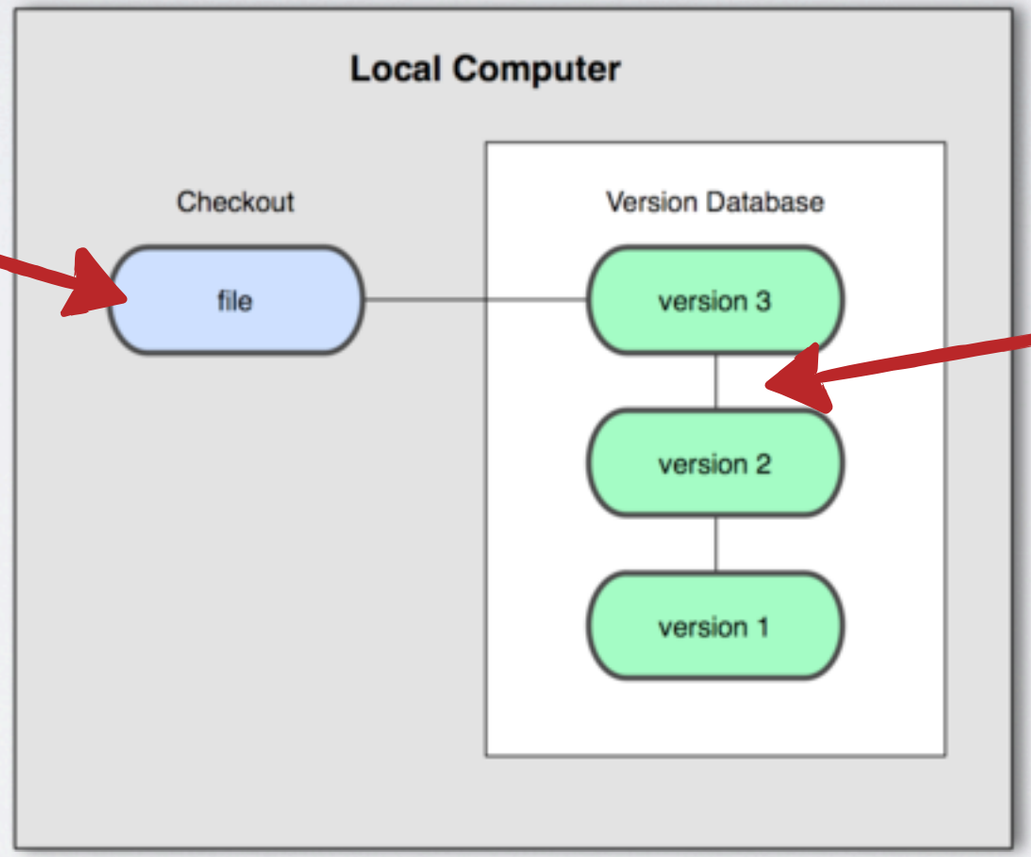
- Wiederherstellung einzelner Dateien oder des ganzen Projektes zum Stand eines bekannten Zeitpunktes
- Analyse der Änderungen eines Systems über die Zeit
- Auffinden des Verantwortlichen bei Fehlern oder Problemen
- Erhöhte Sicherheit, da bei Datenverlust oder groben Fehlern in der aktuellen Änderung ein funktionierender Stand des Systems wieder hergestellt werden kann

Version Control Systems

Lokales VCS

Version Control System

Arbeitskopie

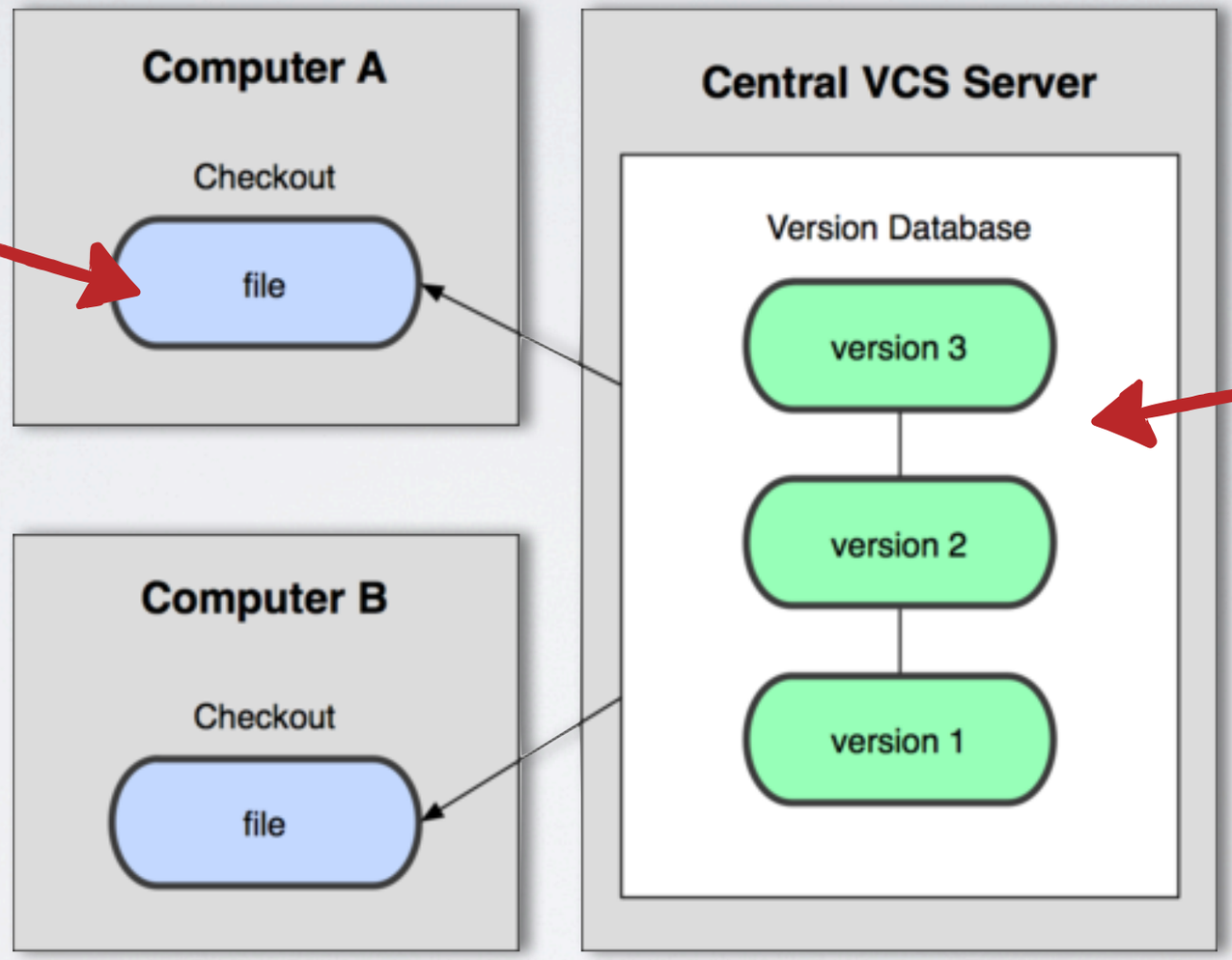


Speichern der Änderungen

git

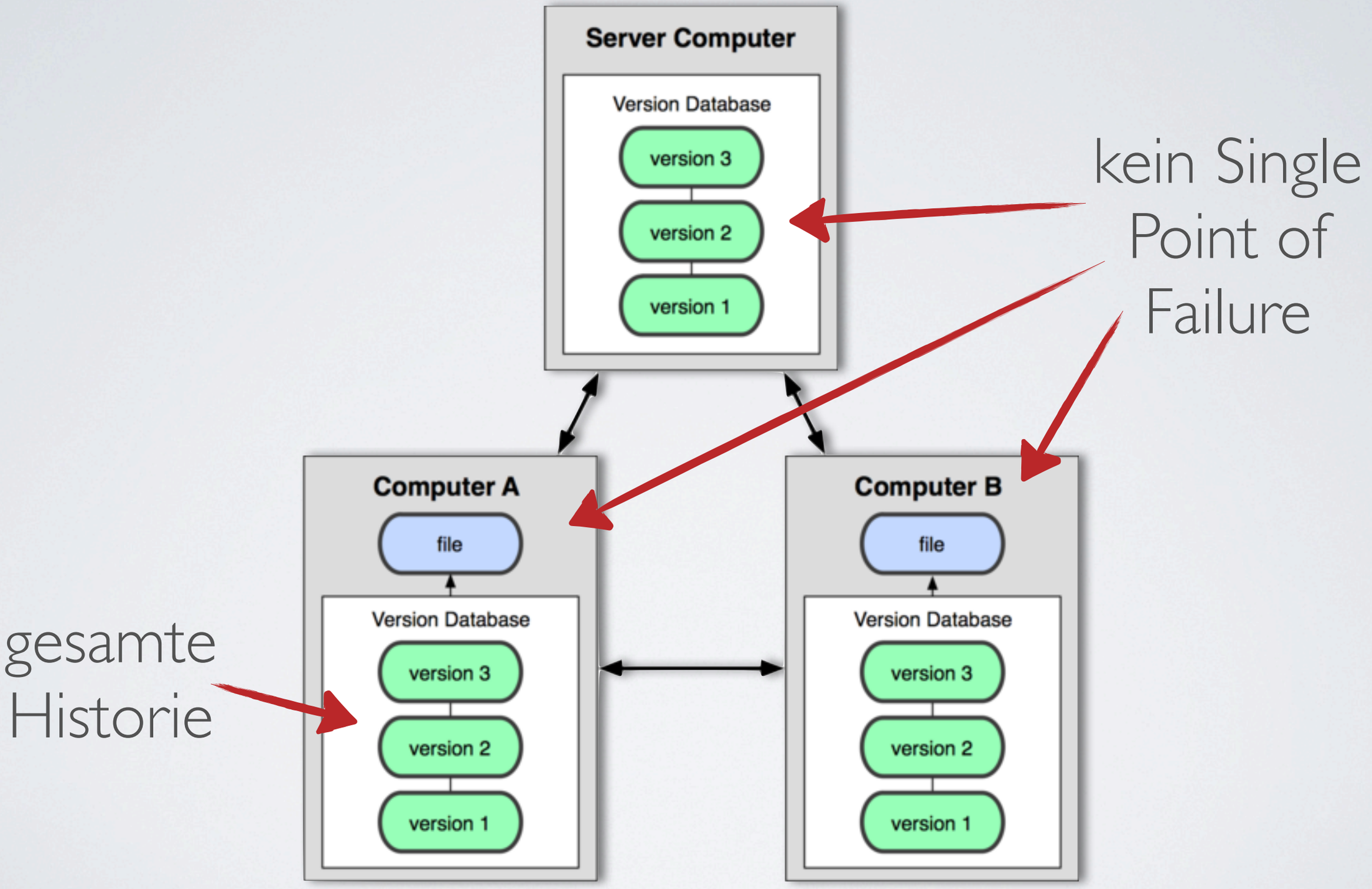
Zentralisiertes VCS

Keine Historie



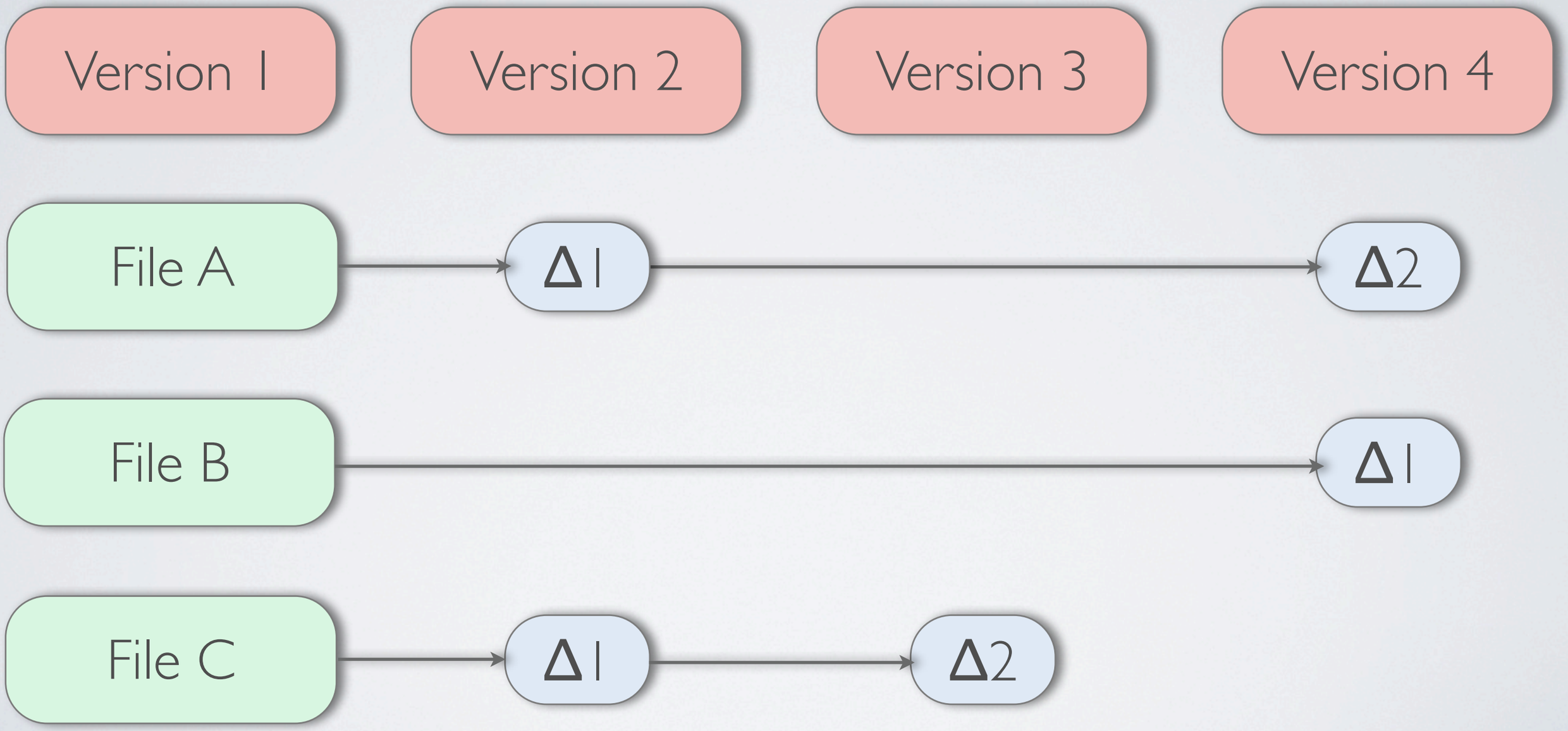
Single Point of Failure

Verteiltes VCS

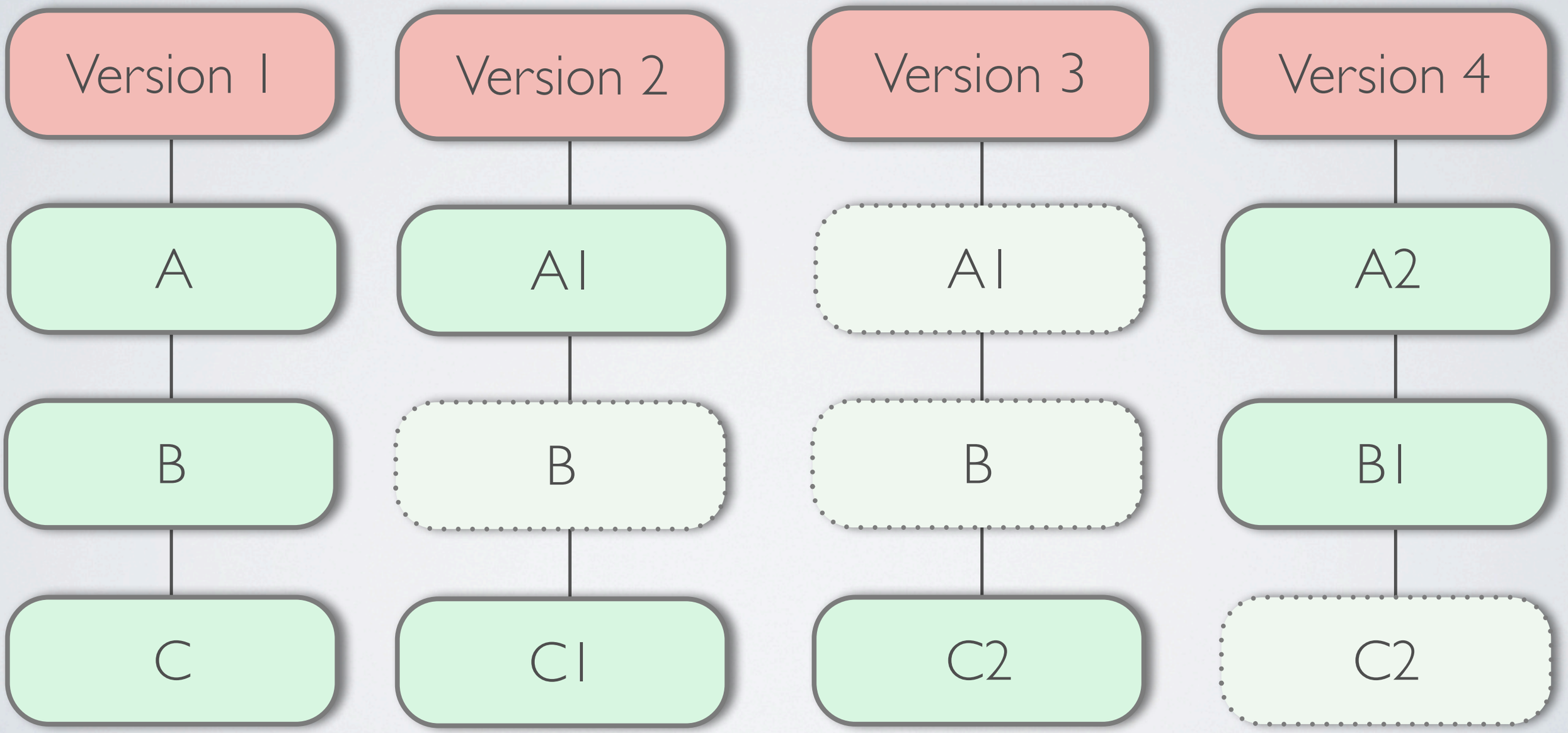


Datenspeicherung

Delta-Mechanismus



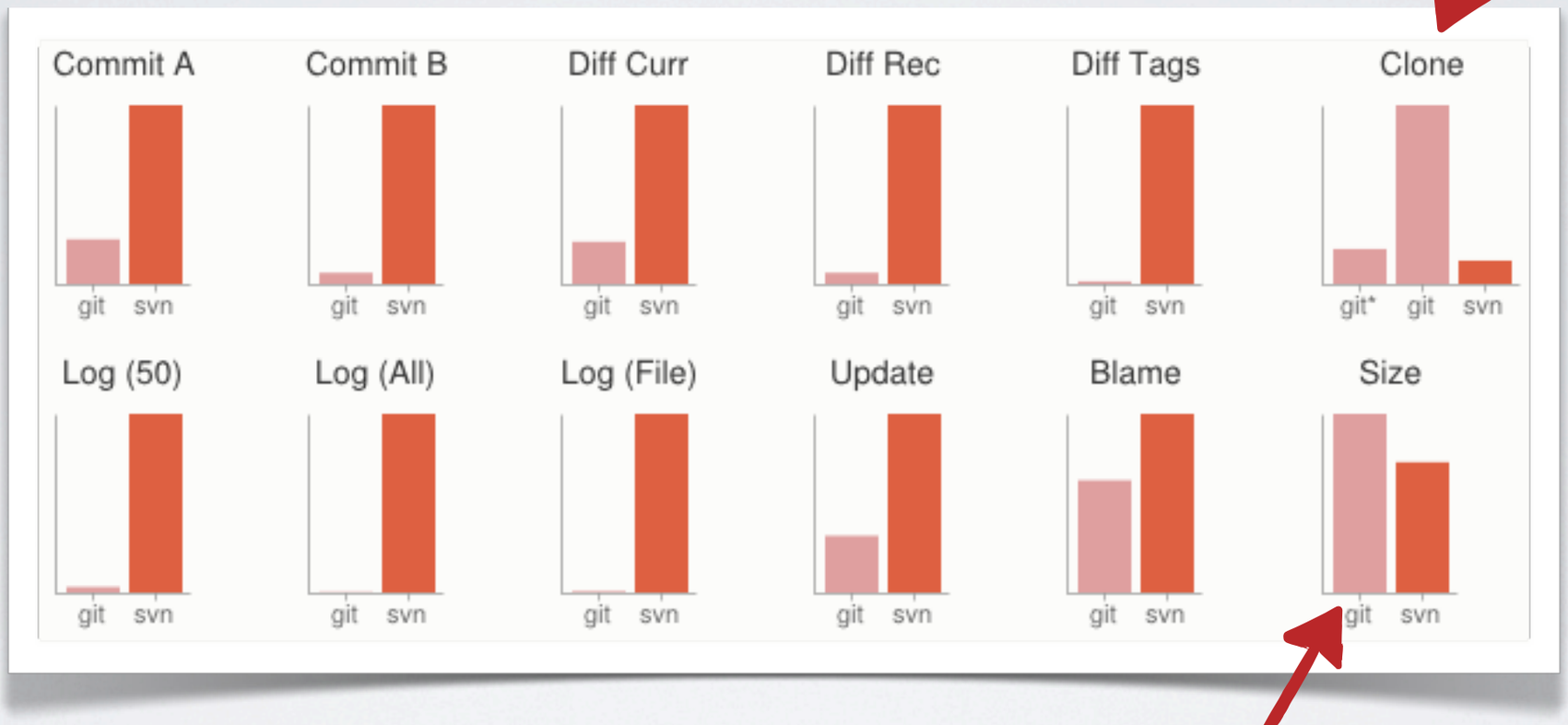
Snapshots



git

Geschwindigkeit: Git vs. SVN

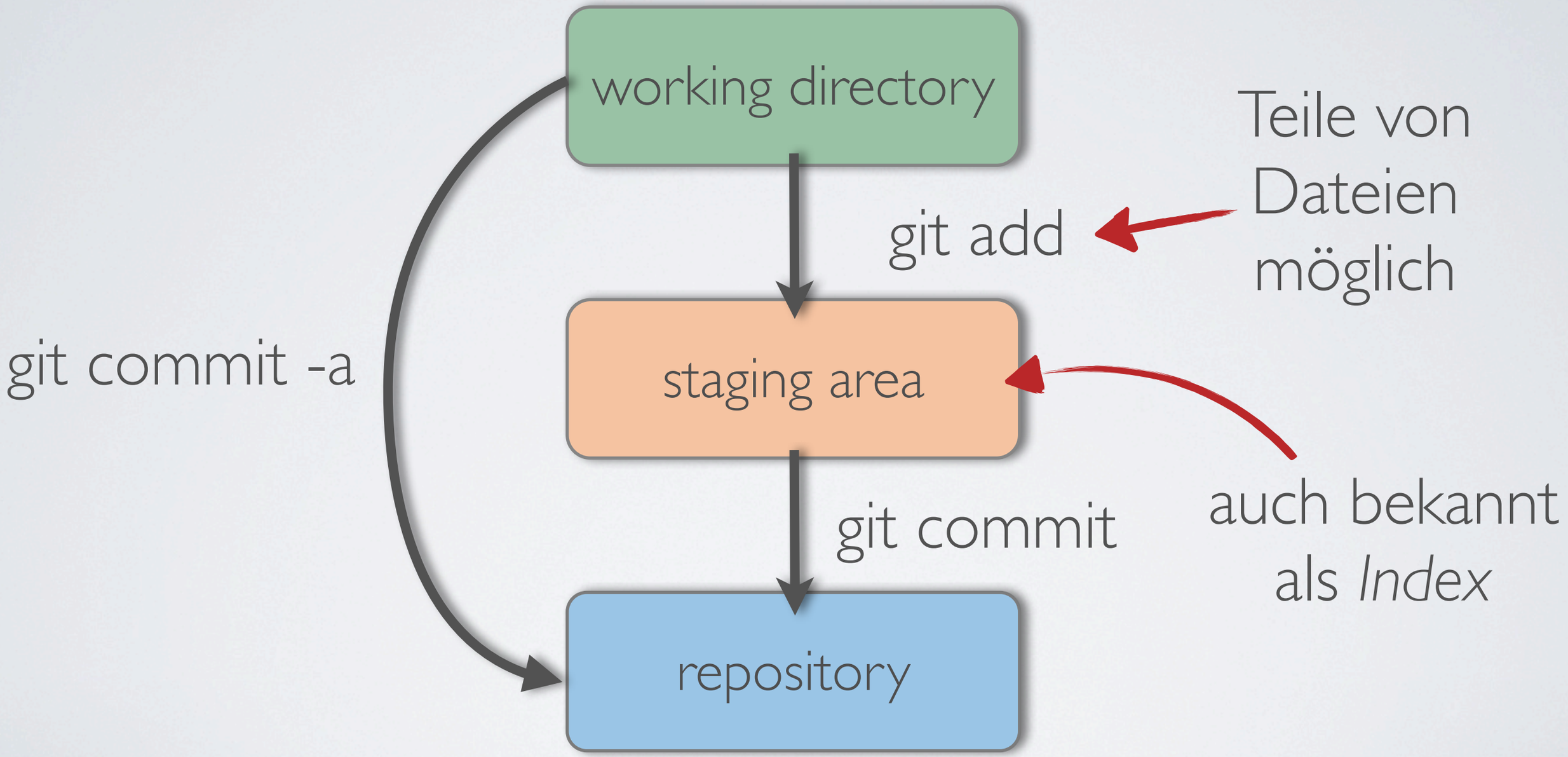
gesamte Historie



181 vs. 132 MB

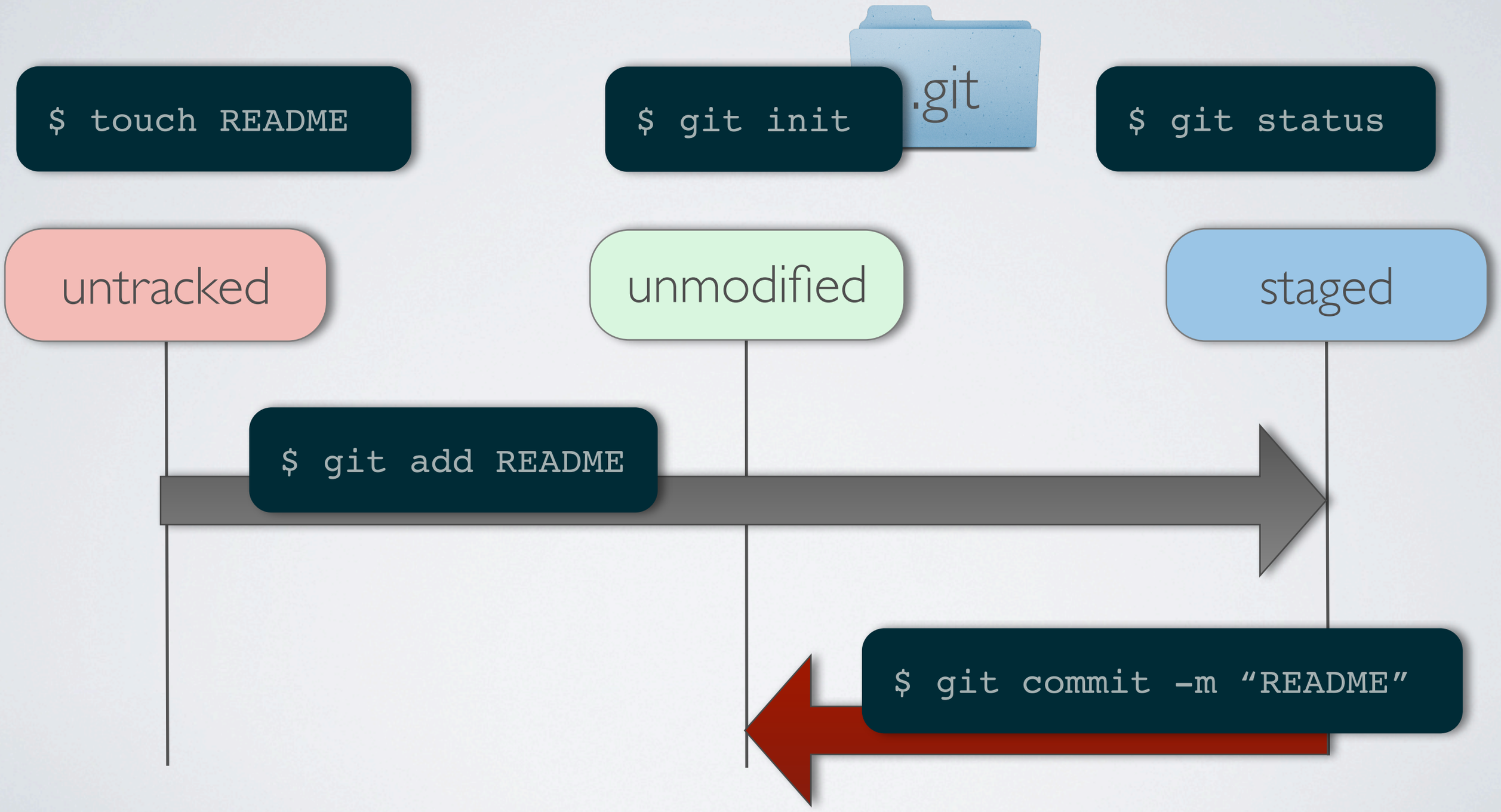
Grundlagen

Bereiche

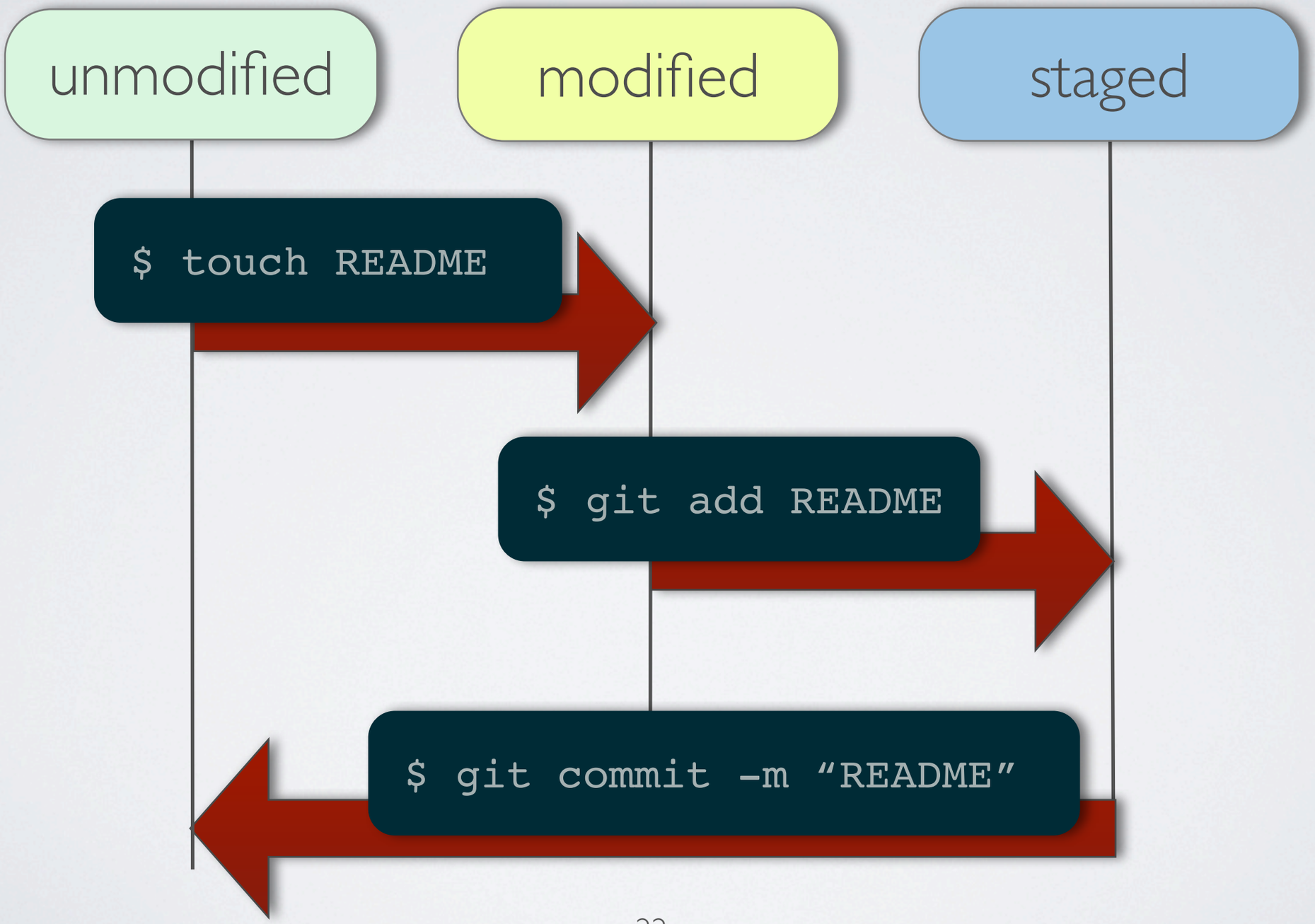


git

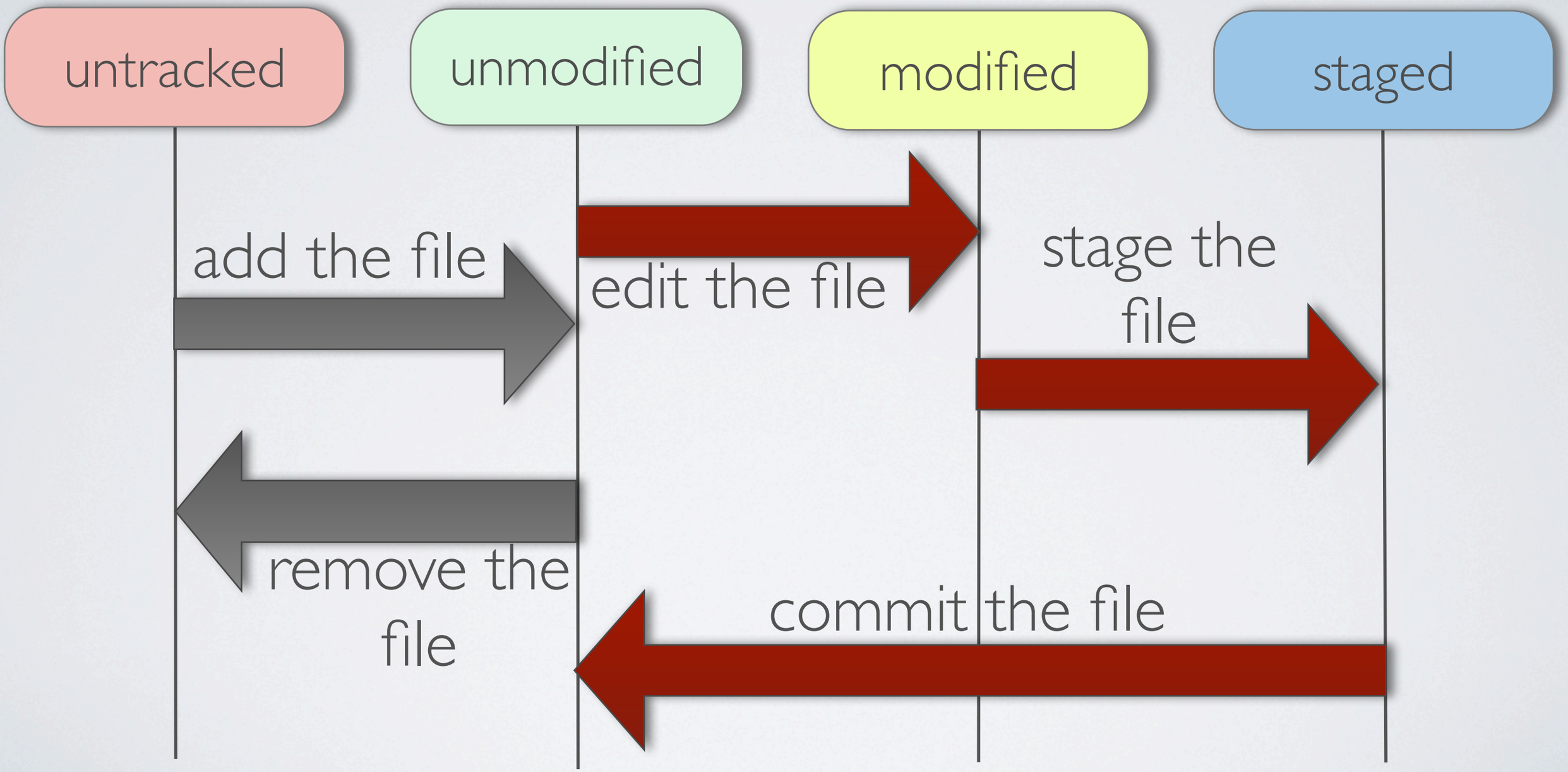
Repository anlegen und Dateien hinzufügen



Dateien bearbeiten



Übersicht

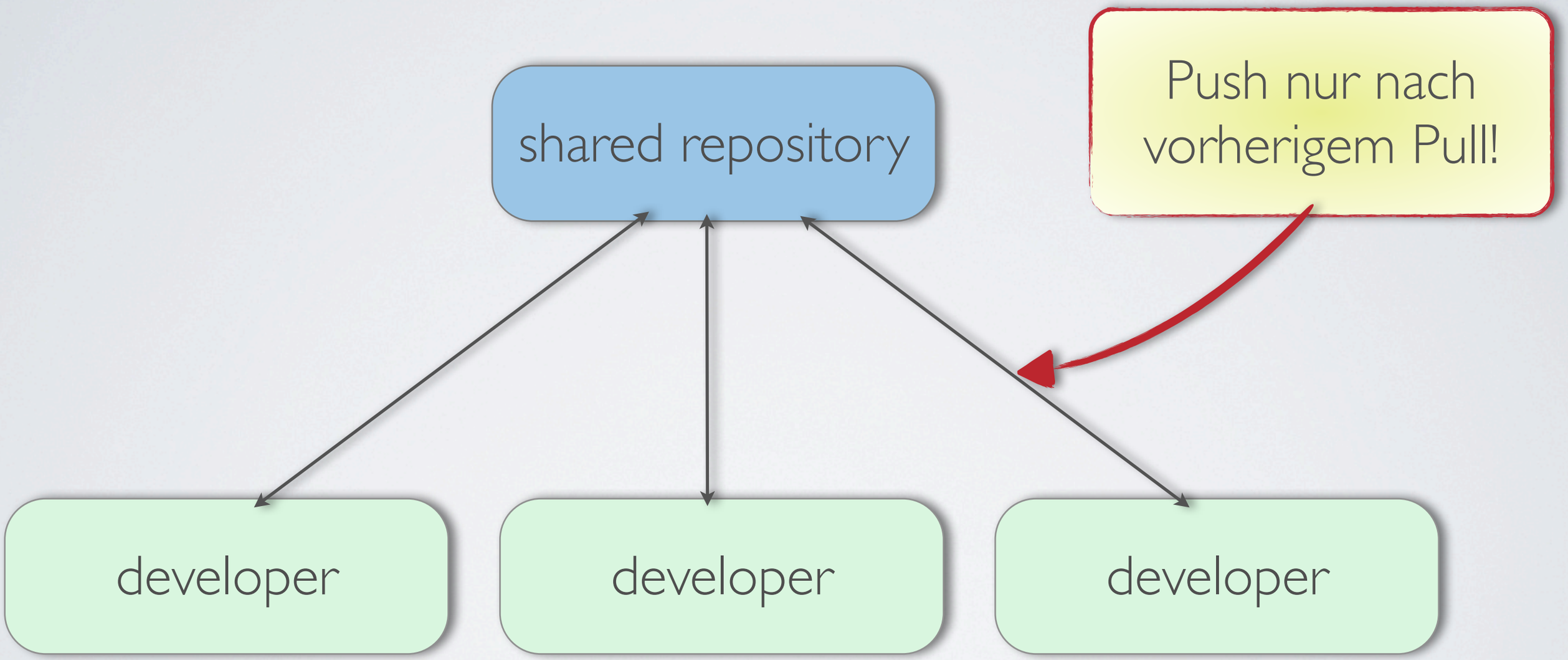

untracked
unmodified
modified
staged
add the file
edit the file
stage the file
remove the file
commit the file

Demo


Ruby

Collaboration

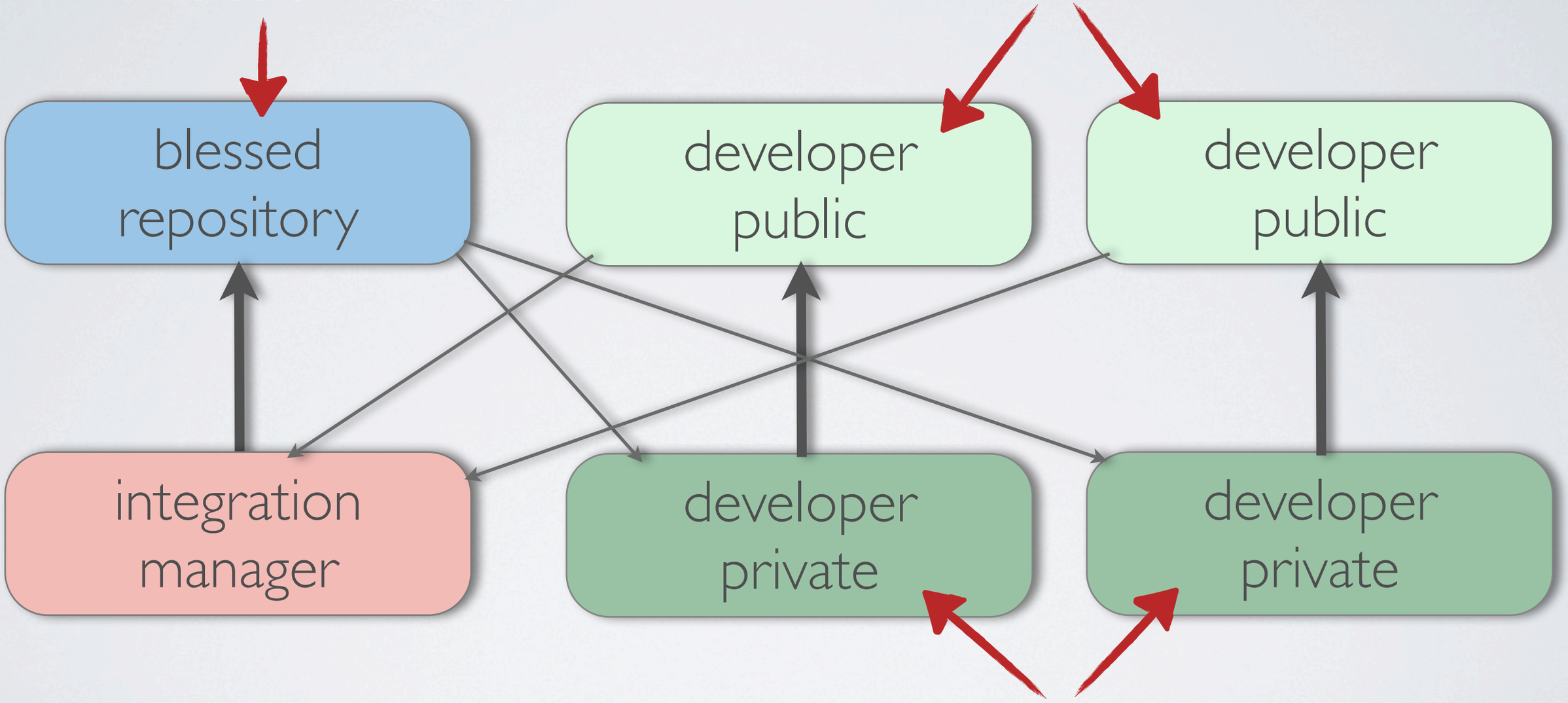
Centralized Workflow



Integration Manager Workflow

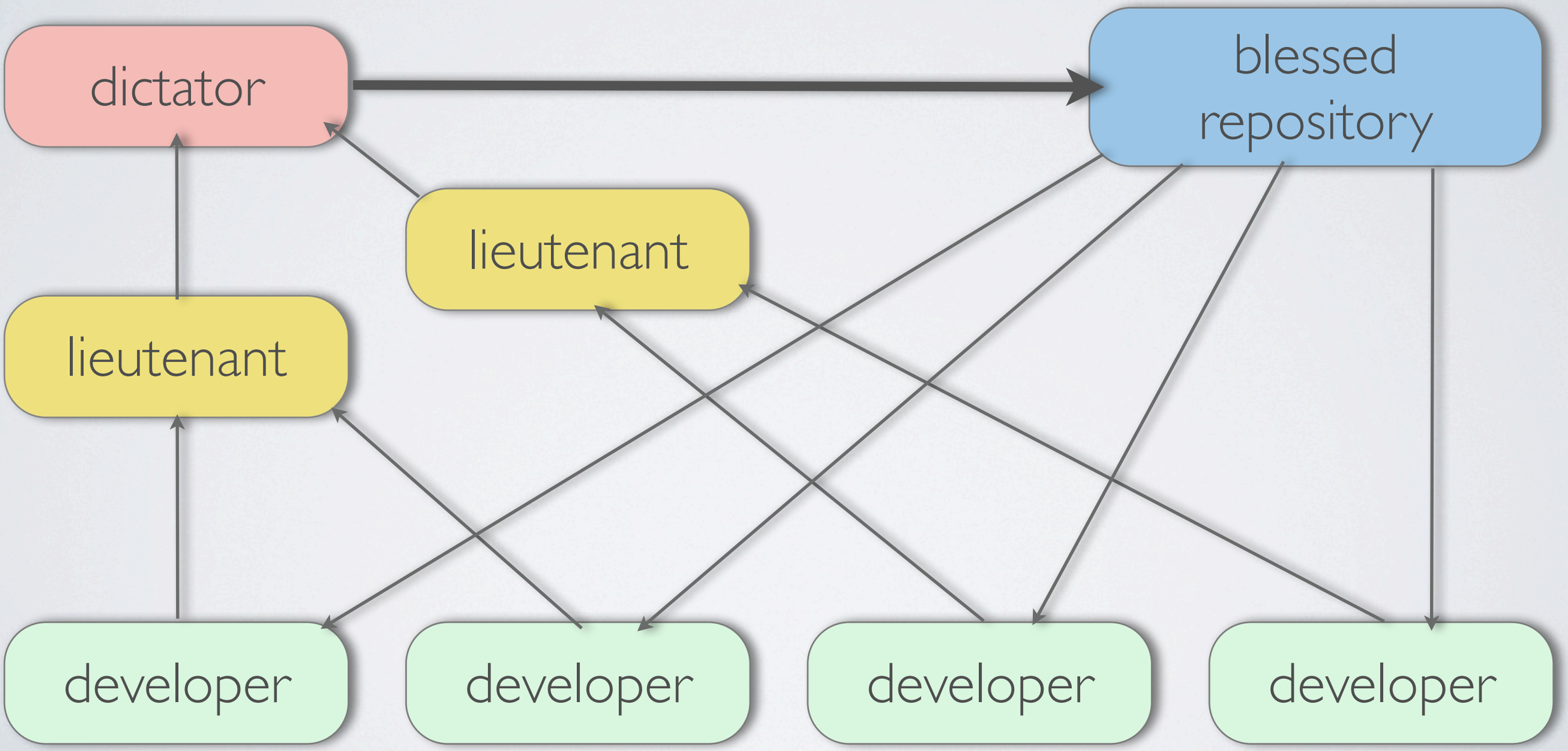
Production Code

öffentlich, z.B. Github



lokal auf dem Rechner

Dictator and Lieutenants Workflow



Demo


Ruby

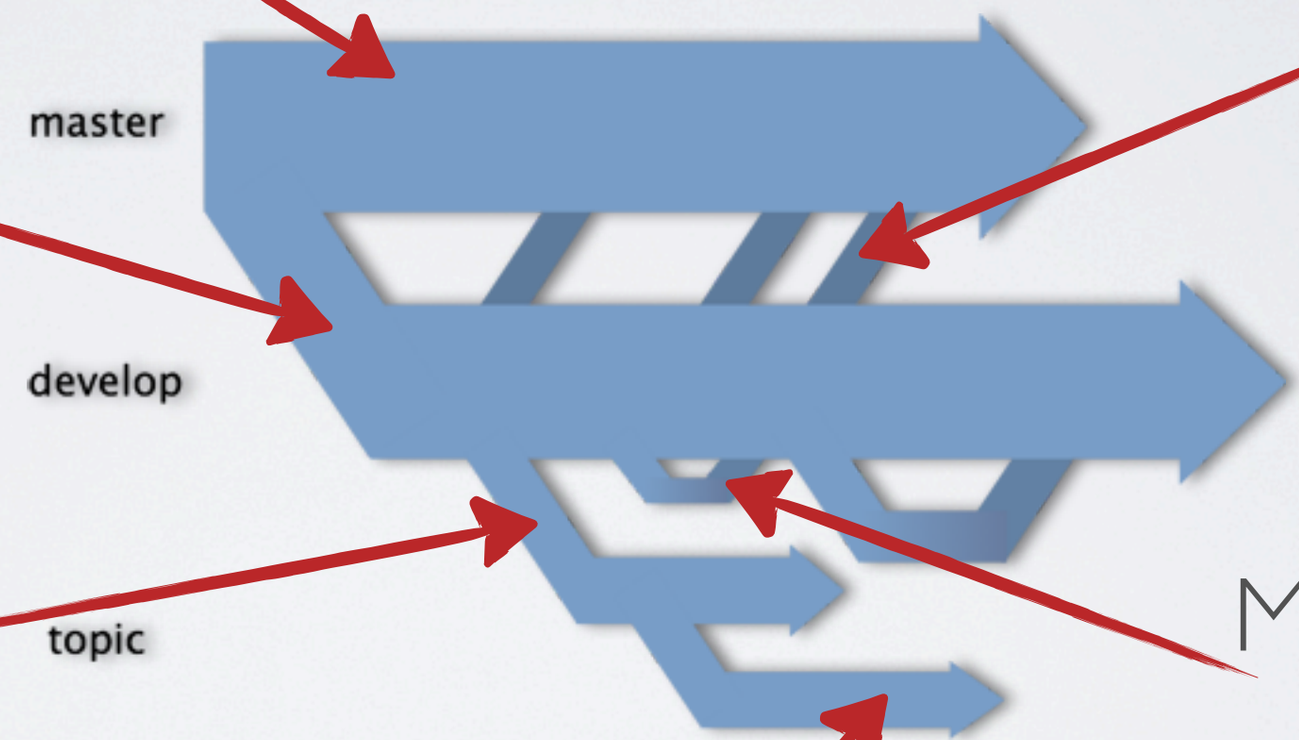
Branching & Merging

Branching & Merging

Production Code

Merge zurück in den Master

Tests und Integration

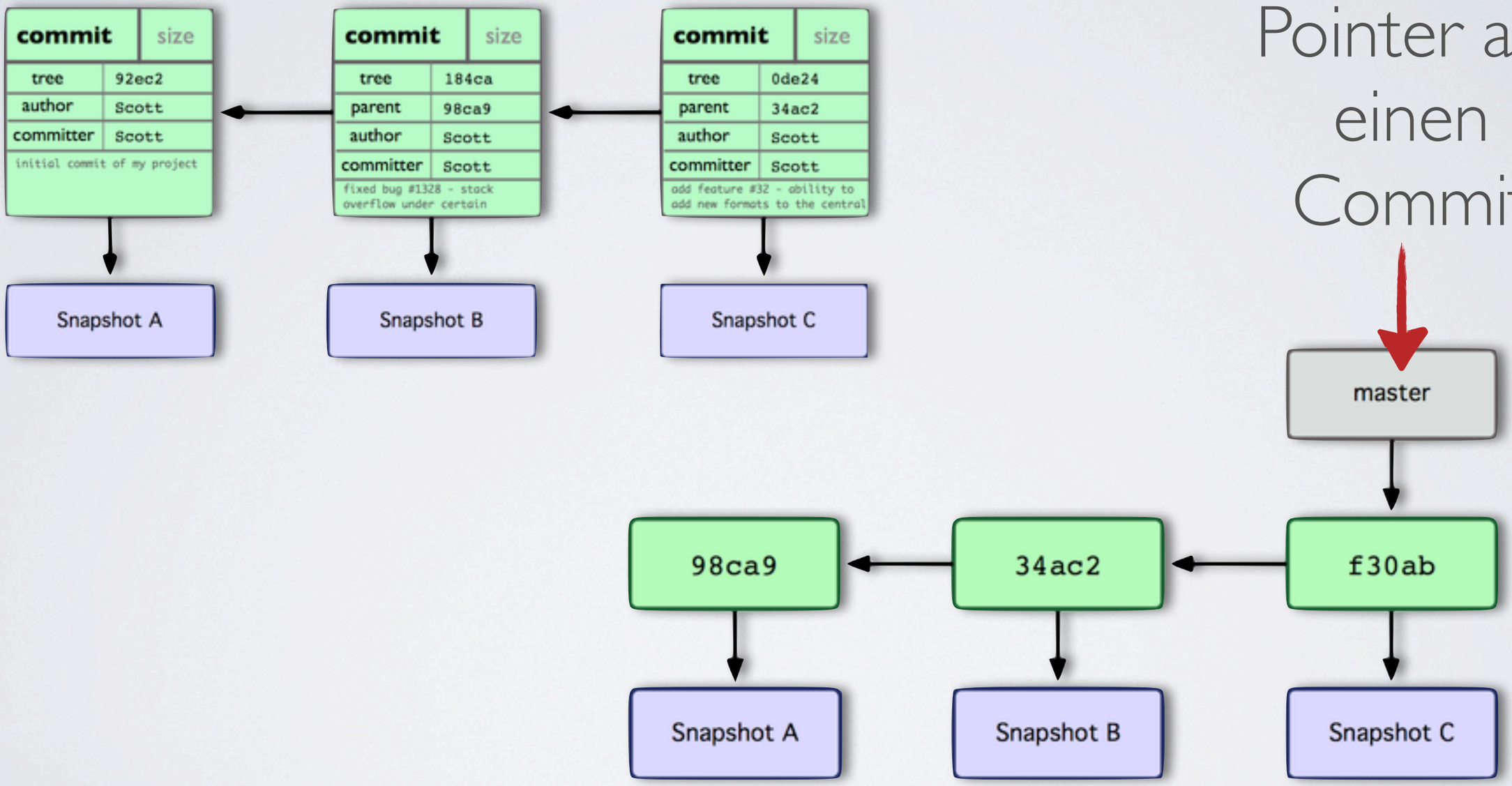


Features und Experimente

Merge bei Erfolg

Verwerfen bei Misserfolg

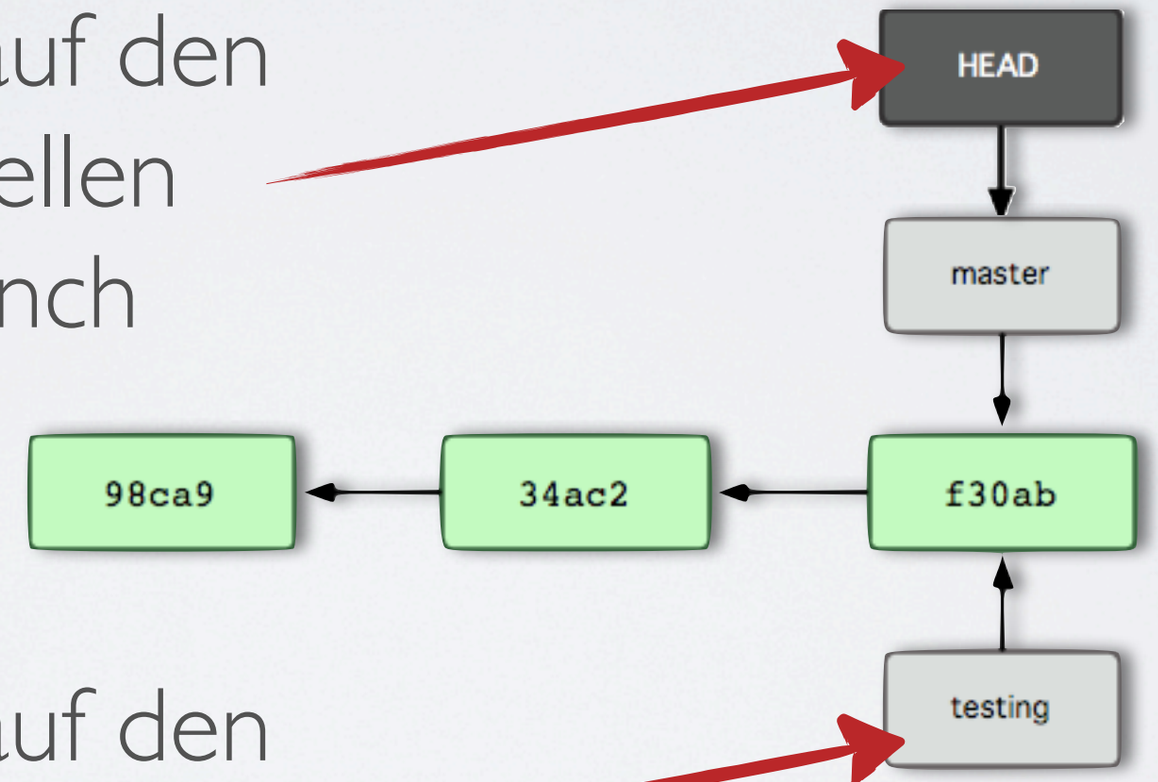
Was ist ein Branch?



Neuen Branch anlegen

```
$ git branch testing
```

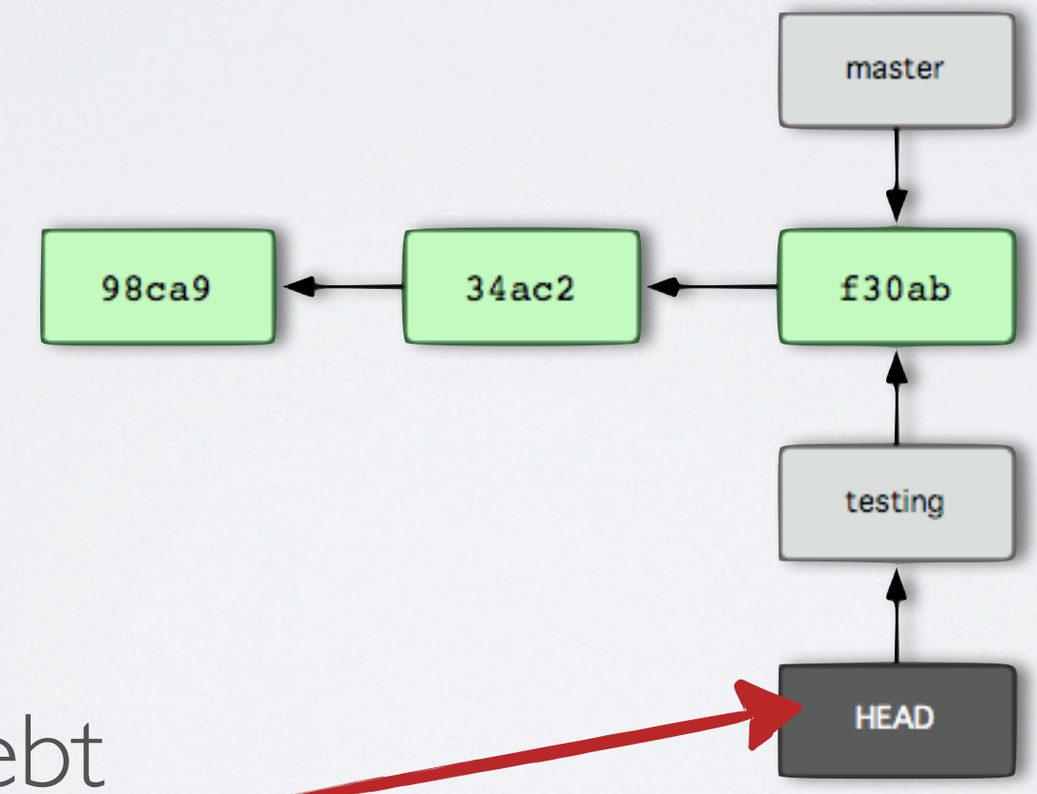
Zeigt auf den aktuellen Branch



Zeigt auf den aktuellen Commit

Branch wechseln

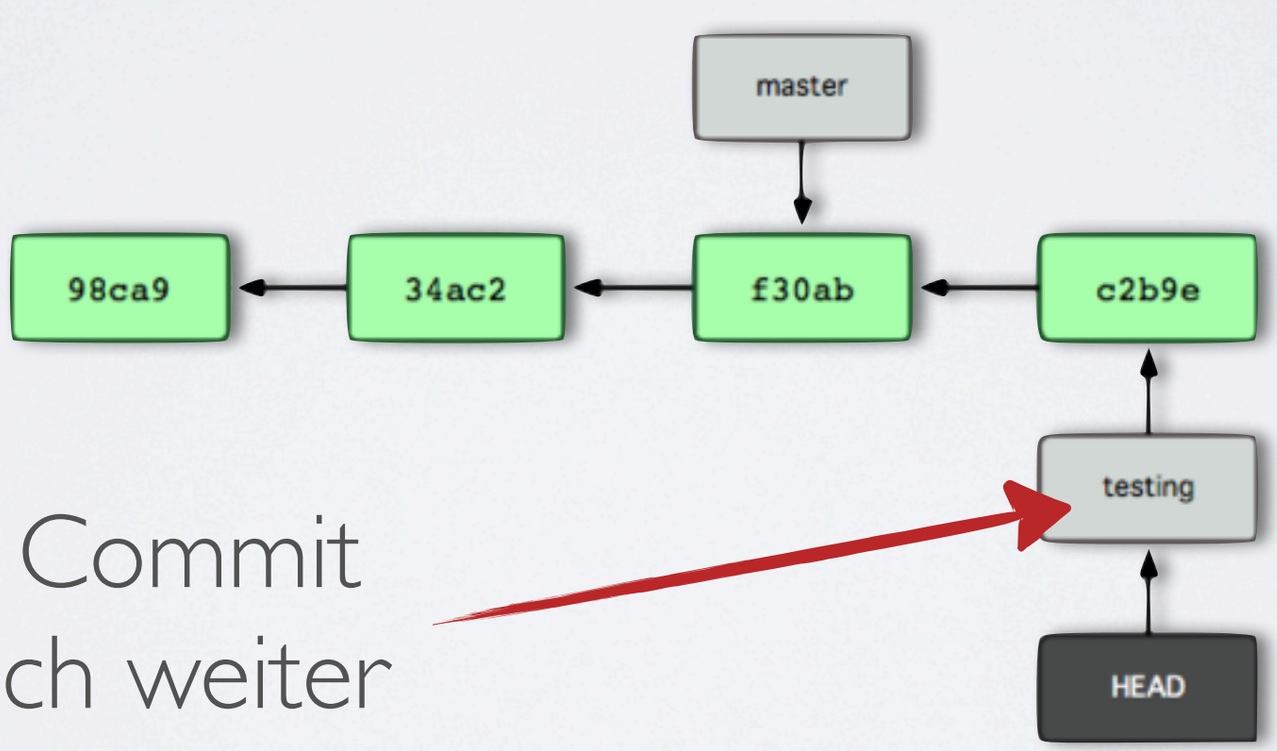
```
$ git checkout testing
```



verschiebt HEAD

Änderungen in einem Branch

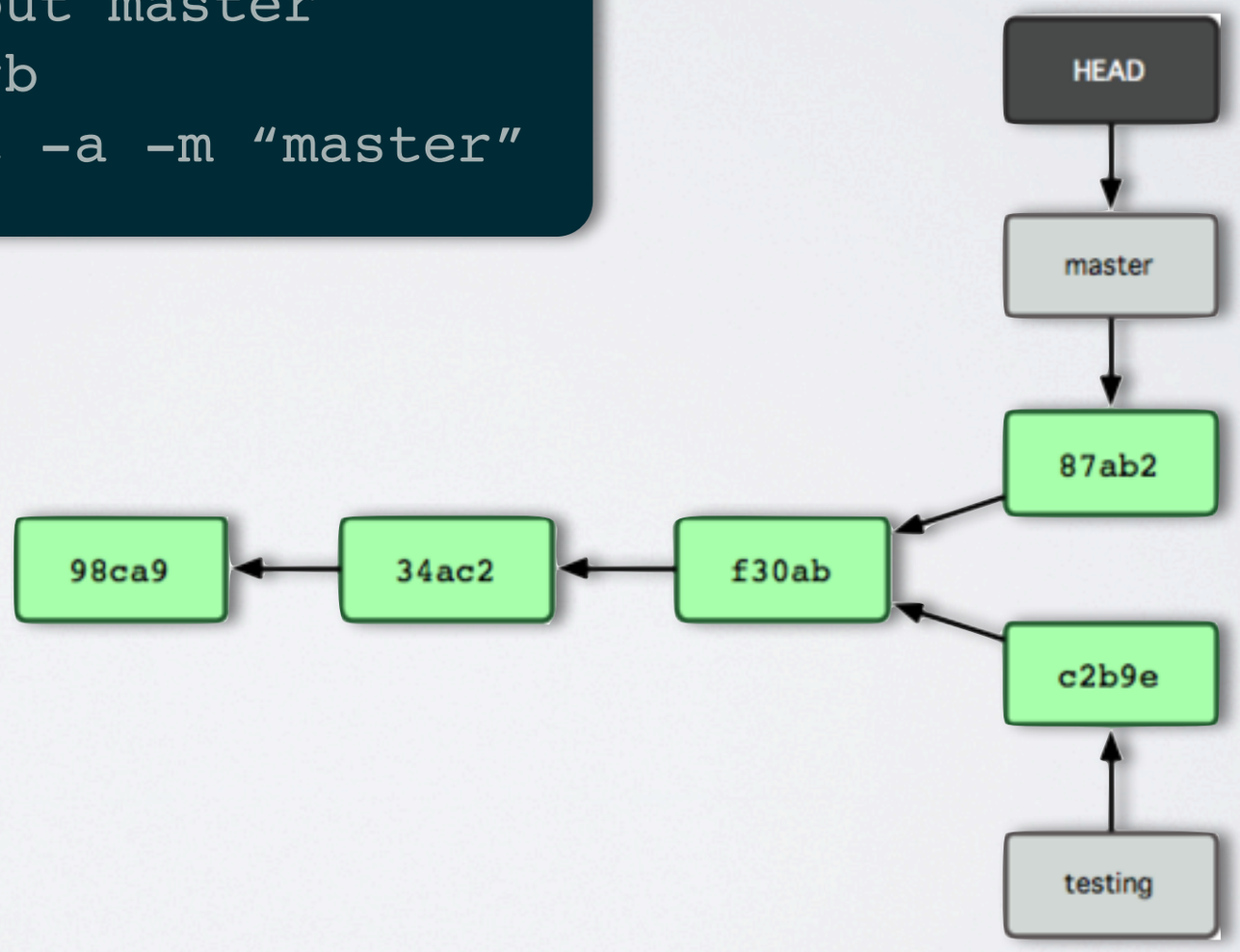
```
$ git checkout testing
$ vim test.rb
$ git commit -a -m “test”
```



rückt bei Commit
automatisch weiter

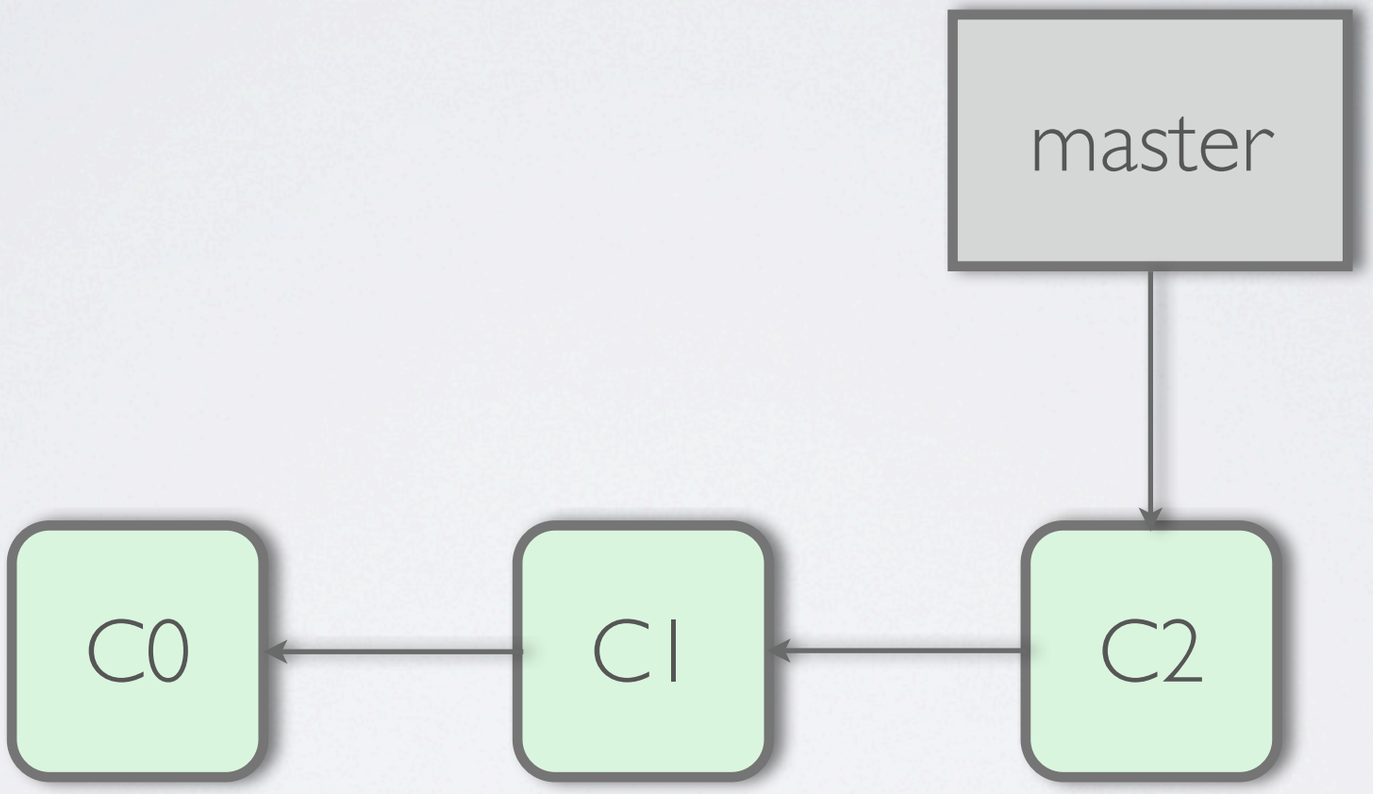
Änderungen in mehreren Branches

```
$ git checkout master
$ vim test.rb
$ git commit -a -m “master”
```



git

Beispiel: Website

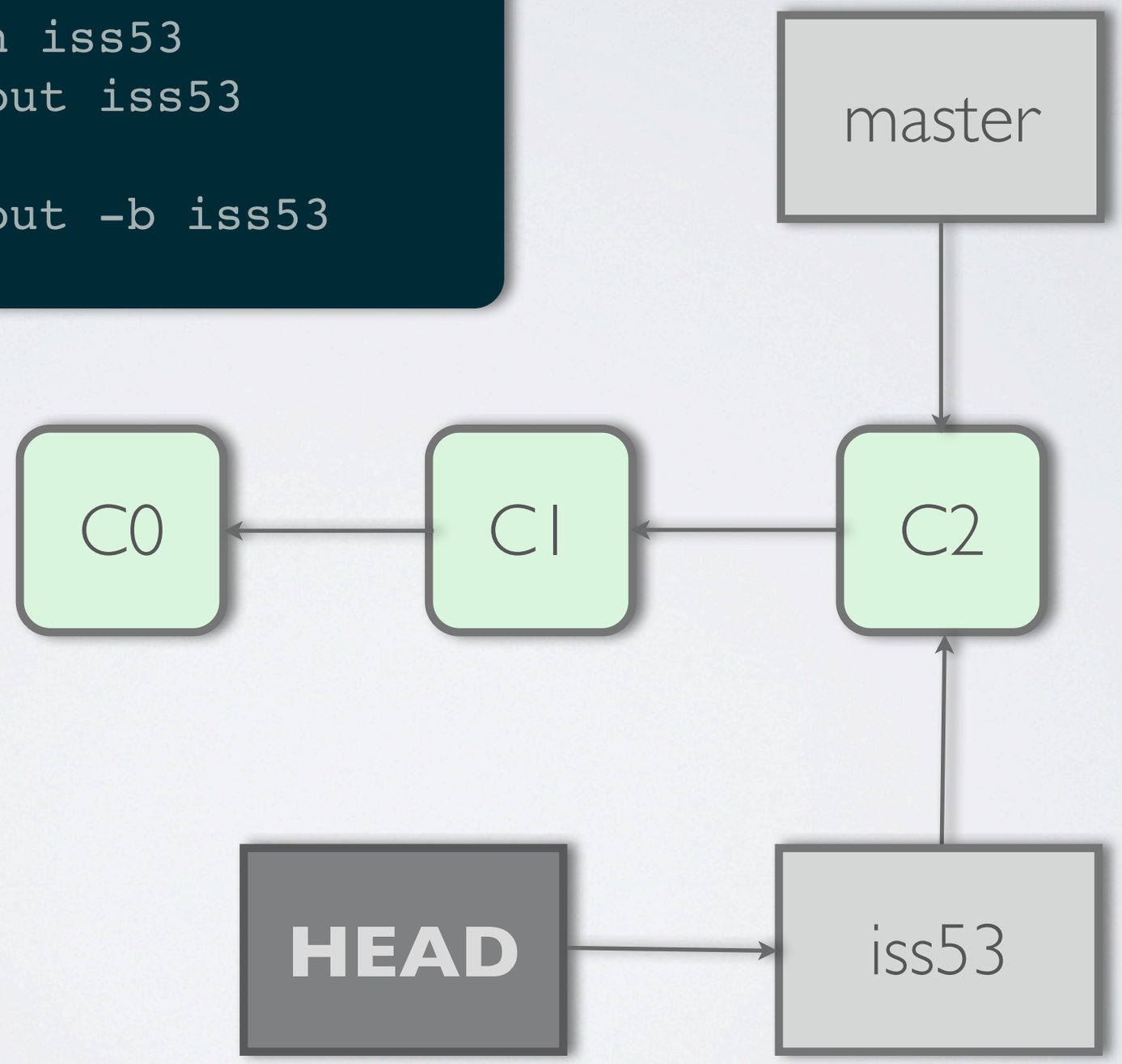


Branch für #53 anlegen

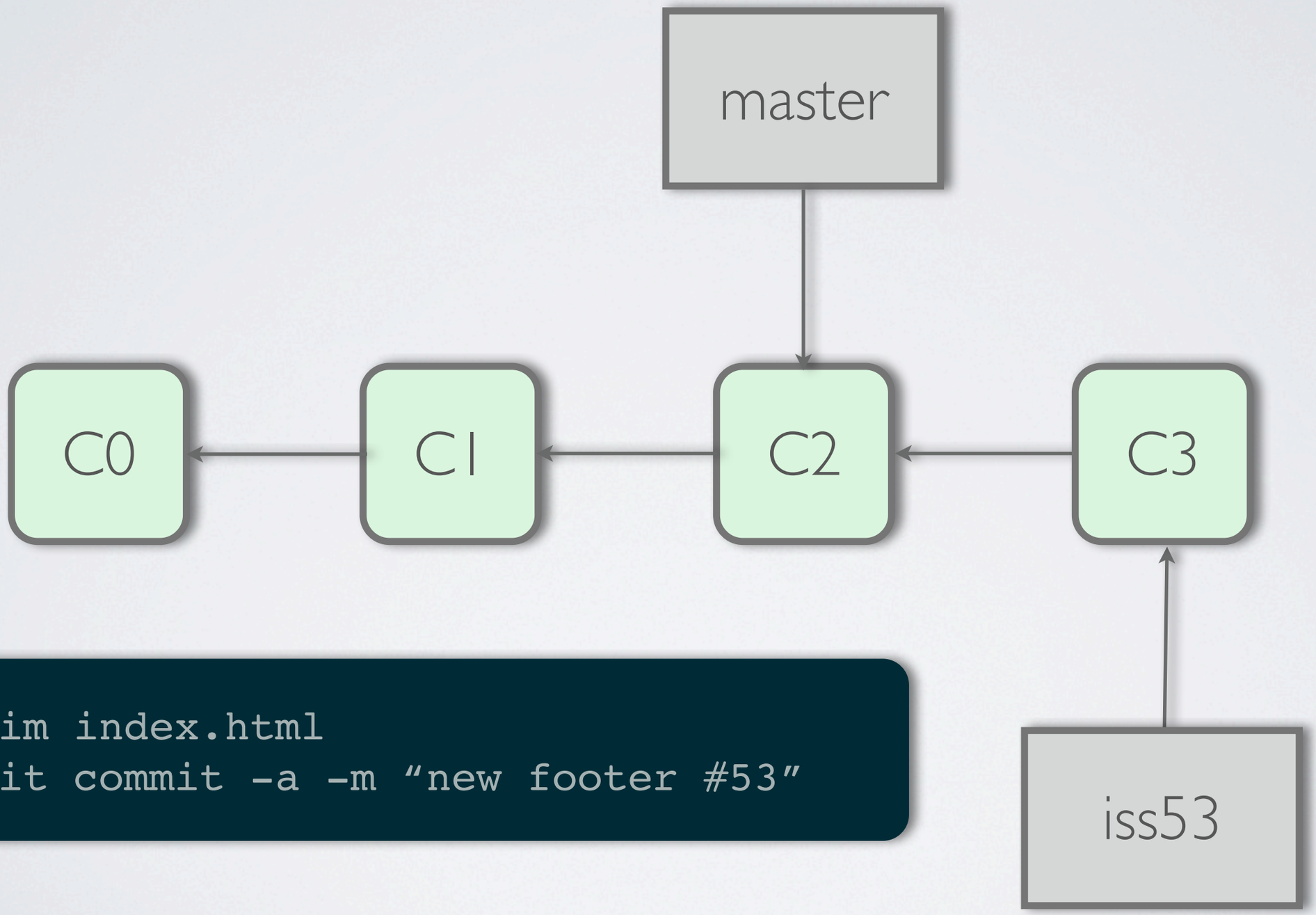
```
$ git branch iss53
$ git checkout iss53

$ git checkout -b iss53
```

Shortcut

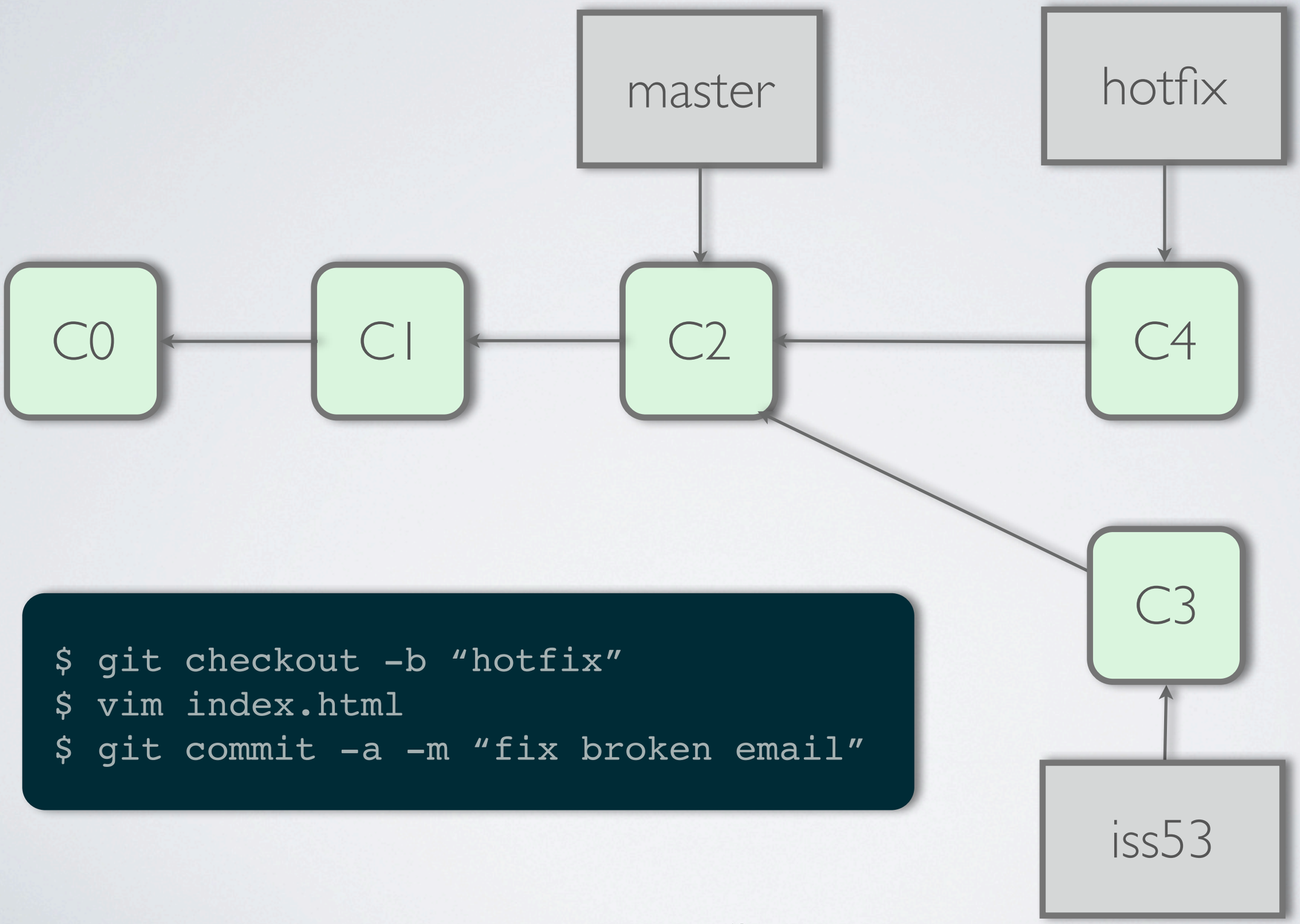


Änderungen durchführen



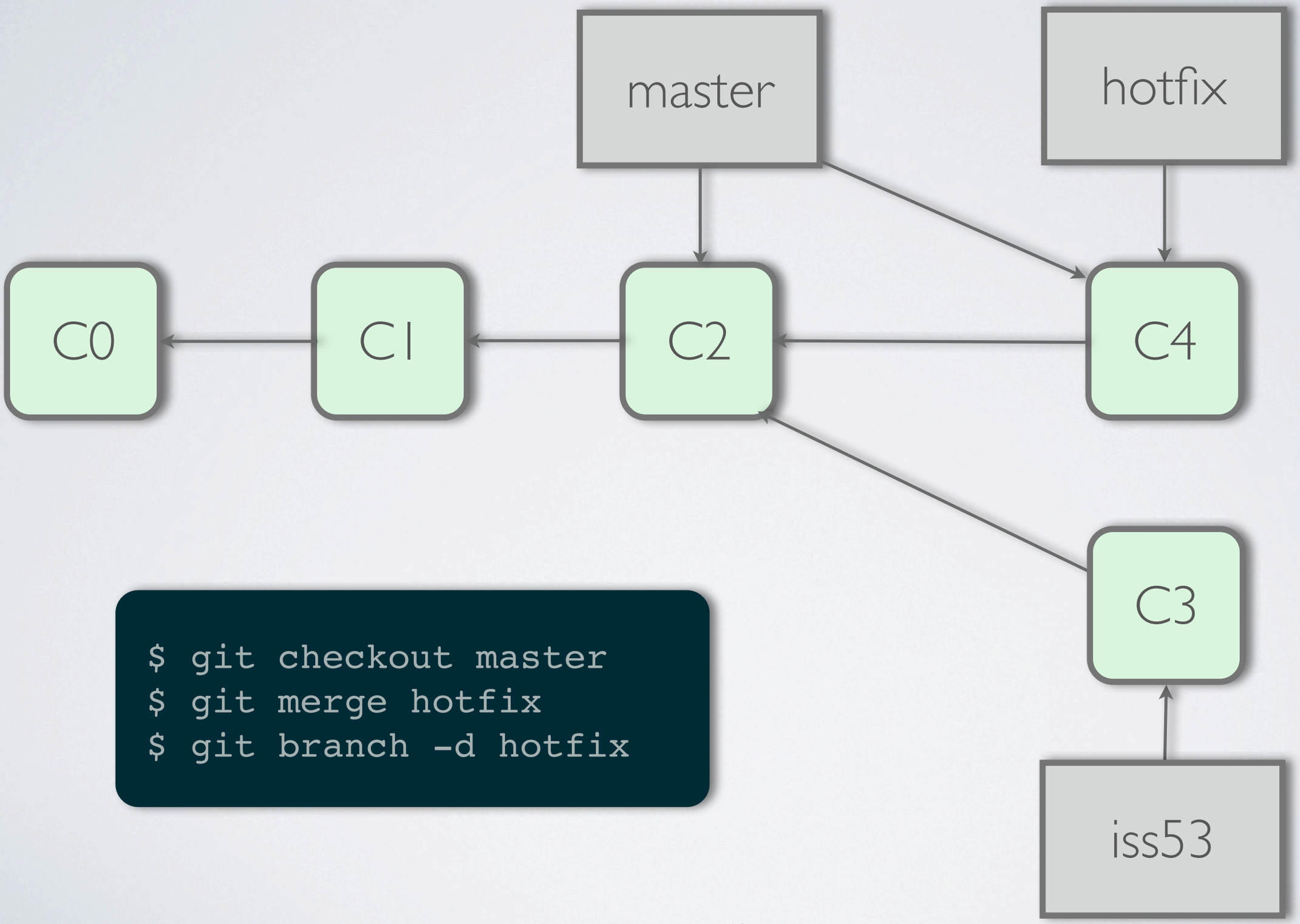

```
$ vim index.html
$ git commit -a -m “new footer #53”
```

Dringender Hotfix




```
$ git checkout -b “hotfix”
$ vim index.html
$ git commit -a -m “fix broken email”
```

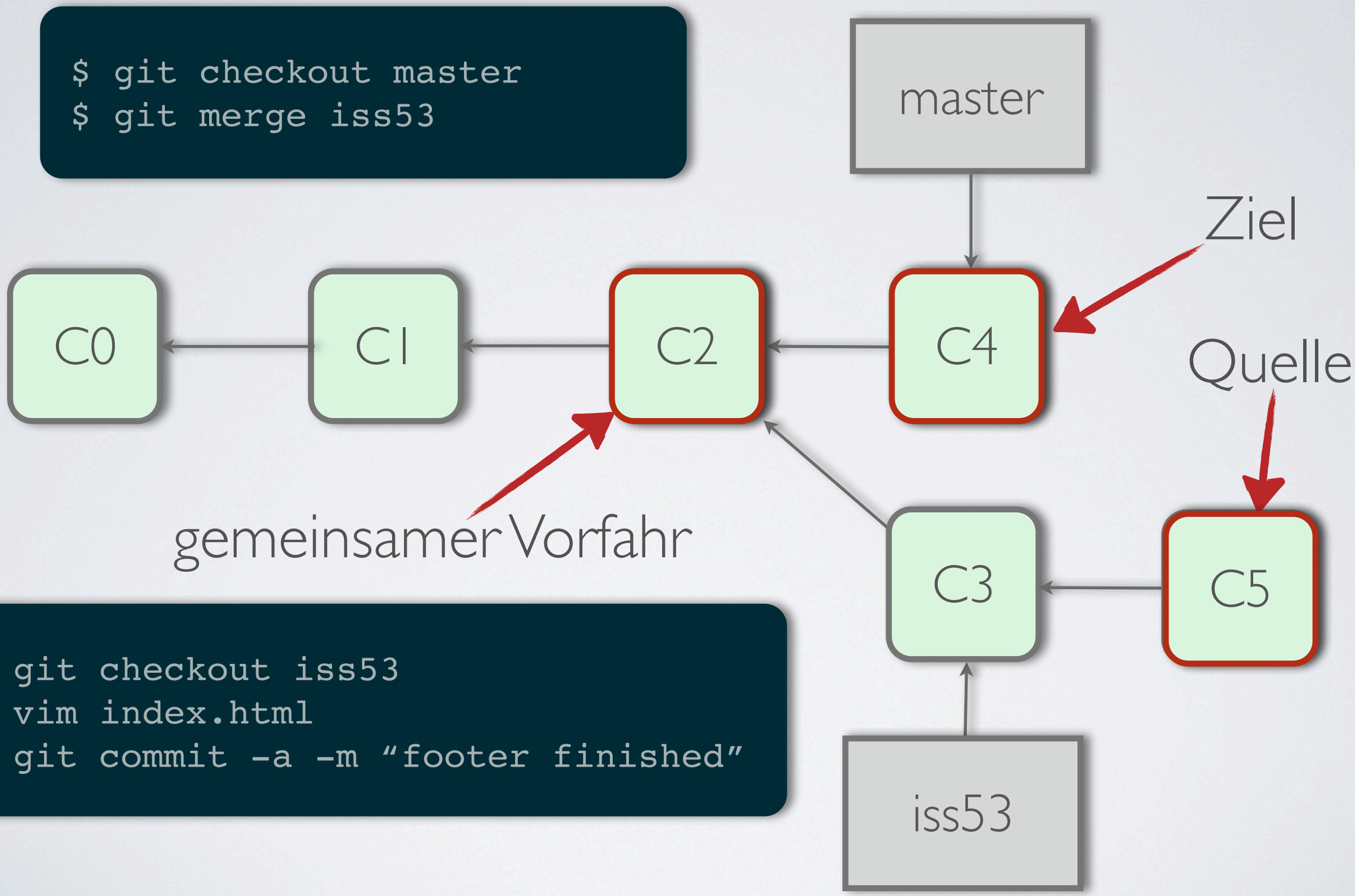
Fast-Forward Merge




```
$ git checkout master
$ git merge hotfix
$ git branch -d hotfix
```

Three-Way Merge

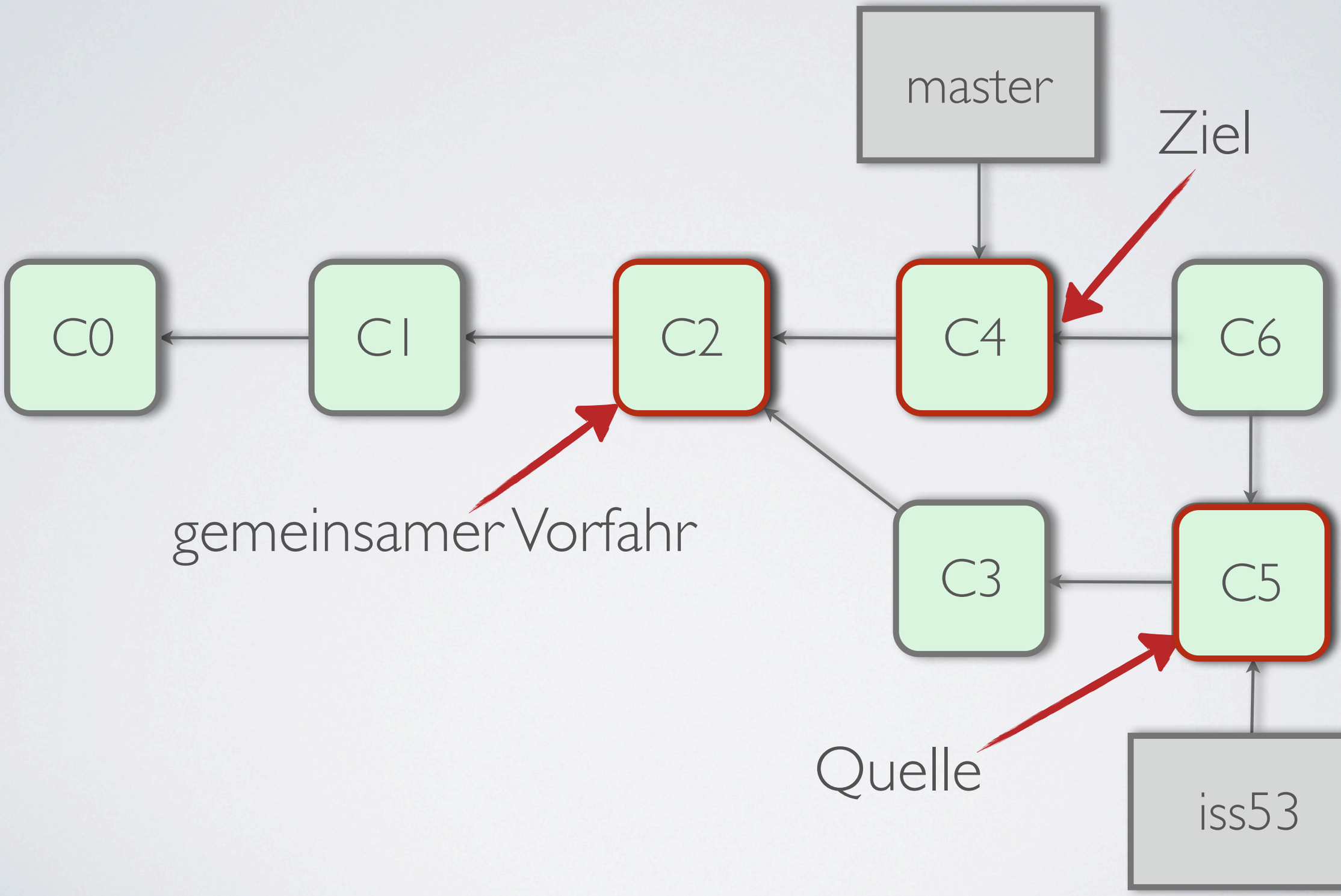
```
$ git checkout master
$ git merge iss53
```




```
$ git checkout iss53
$ vim index.html
$ git commit -a -m “footer finished”
```


git

Three-Way Merge



Git und Vim

https://github.com/tpope/vim-fugitive

http://vimcasts.org/