Anhang A: Versionsverwaltung mit Git

Gliederung

Motivation

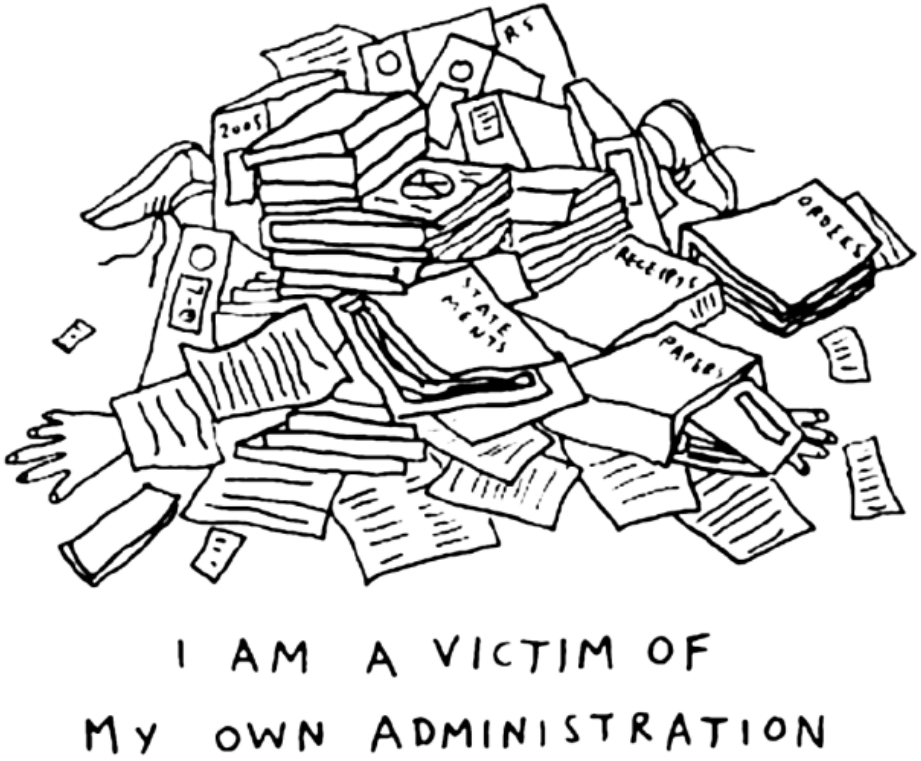
Konzepte

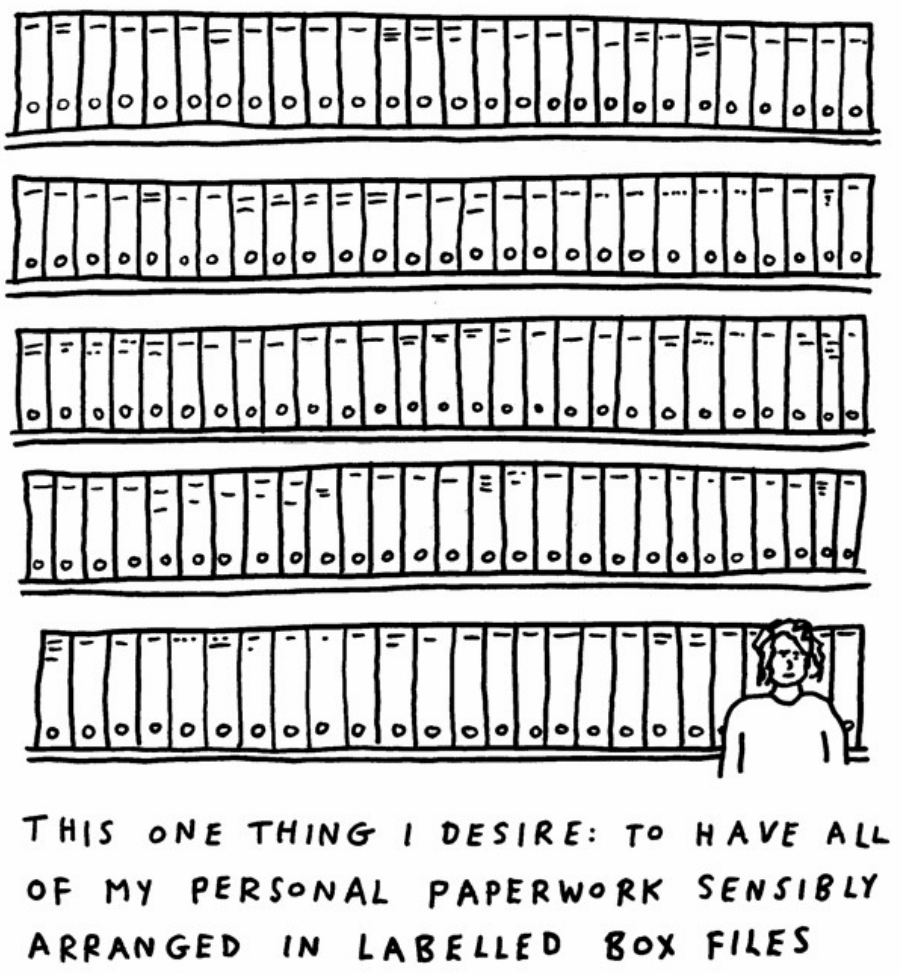
Git vs. GitHub

Git Kommandos

Motivation

Versionsverwaltung: Motivation

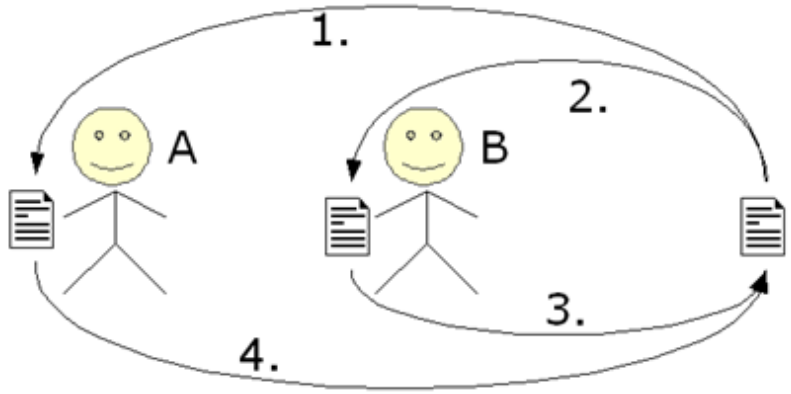




Bildquellen: http://weblogcartoons.com

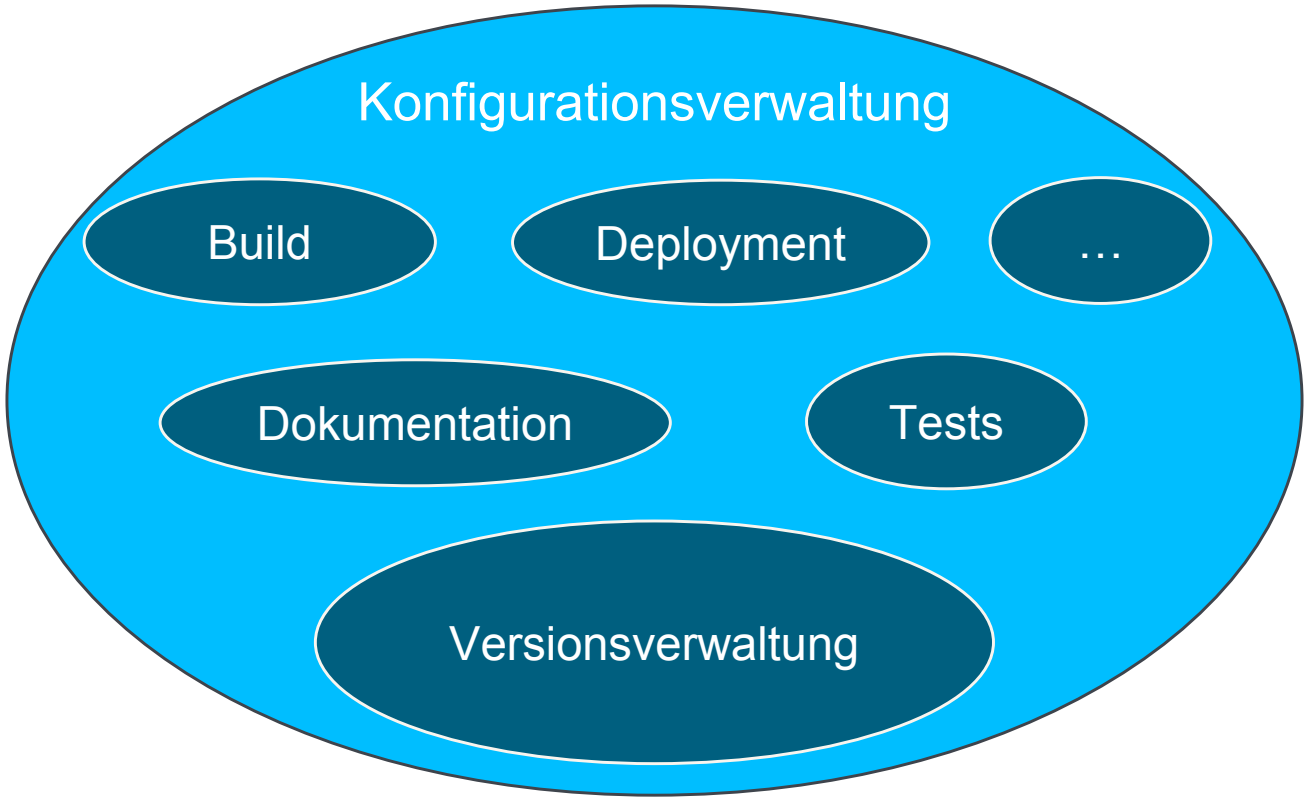
Versionsverwaltung: Motivation

- Code und andere Dokumente entstehen im Laufe der Zeit in verschiedenen Fassungen
- Entwicklungsgeschichte sollte immer aufbewahrt werden, um bei Bedarf alte Fassungen rekonstruieren zu können
- Andernfalls verliert man den Überblick darüber, **wer was wann** geändert hat
- Änderungen verschiedener Entwickler können durch gegenseitiges Überschreiben verloren gehen



- Lösung: Maßnahmen, die eine geordnete Entwicklung auch mit vielen Dateien, Versionen und Entwicklern ermöglichen:
  Versionsverwaltungssystem (Version Control System, VCS)

Versionsverwaltung: Teil der Konfigurationsverwaltung



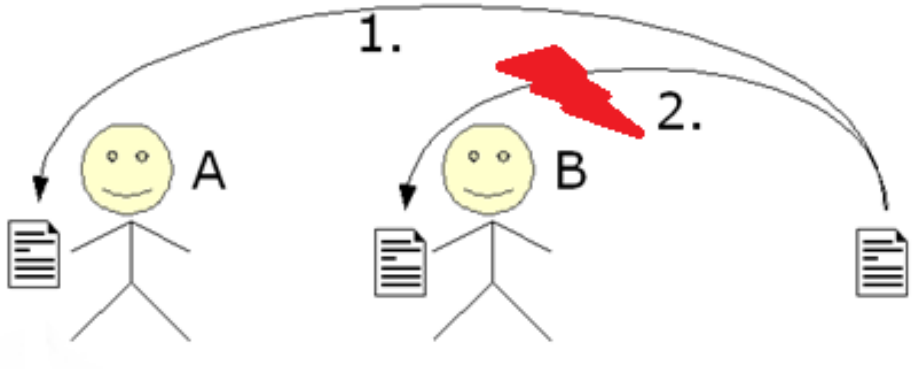
"[Konfigurationsverwaltung] stellt für die Software-Projekte eine ebenso wichtige Infrastruktur dar wie die Stromversorgung für einen produzierenden Betrieb: Wenn sie vorhanden ist und funktioniert, bemerkt man sie nicht, aber wenn sie unterbrochen ist, geht nichts mehr."

Ludewig & Lichter (2010), *Software Engineering*

Konzepte

Versionsverwaltung: Konzepte

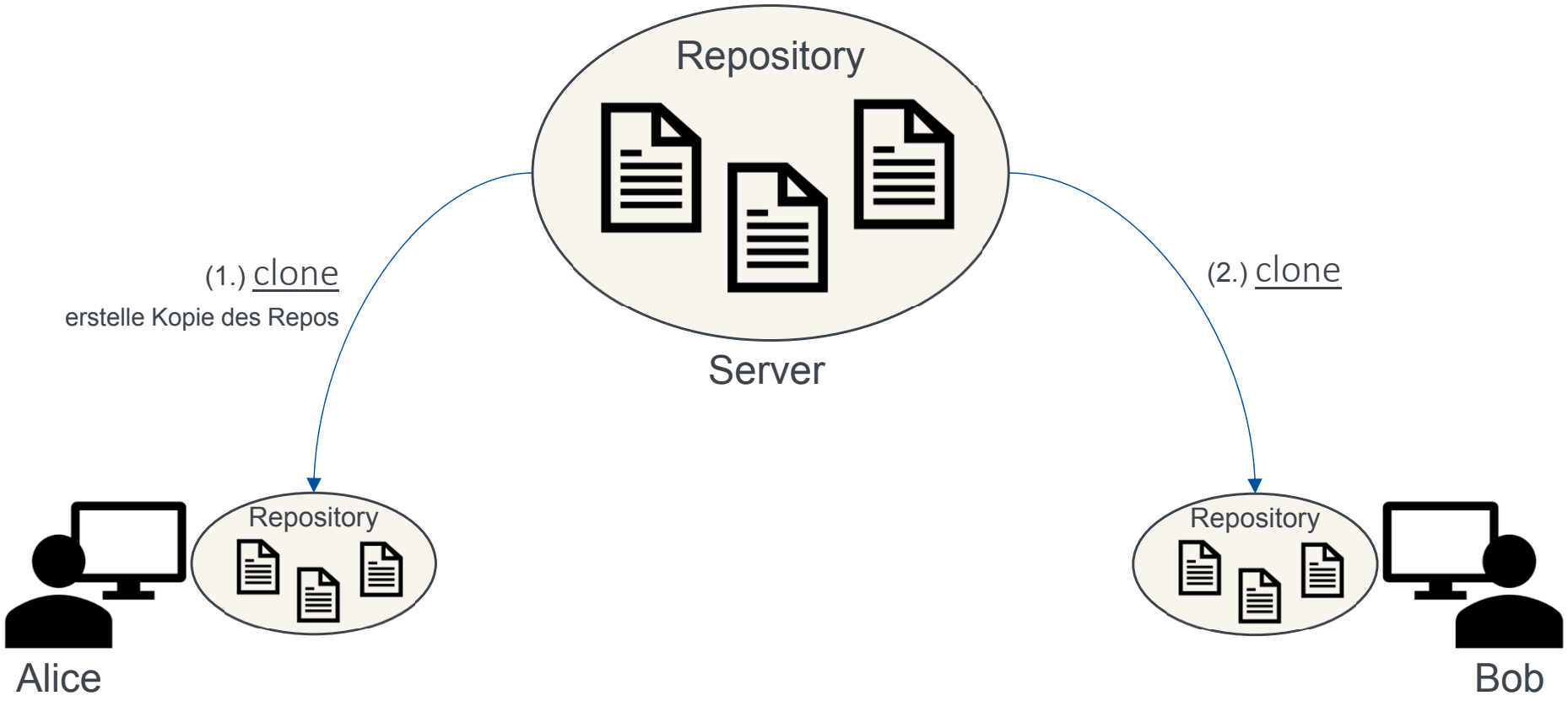
- **Konzept 1**: **Lock-Modify-Unlock**
  - Sperren einer Datei am zentralen Ort zur lokalen Bearbeitung



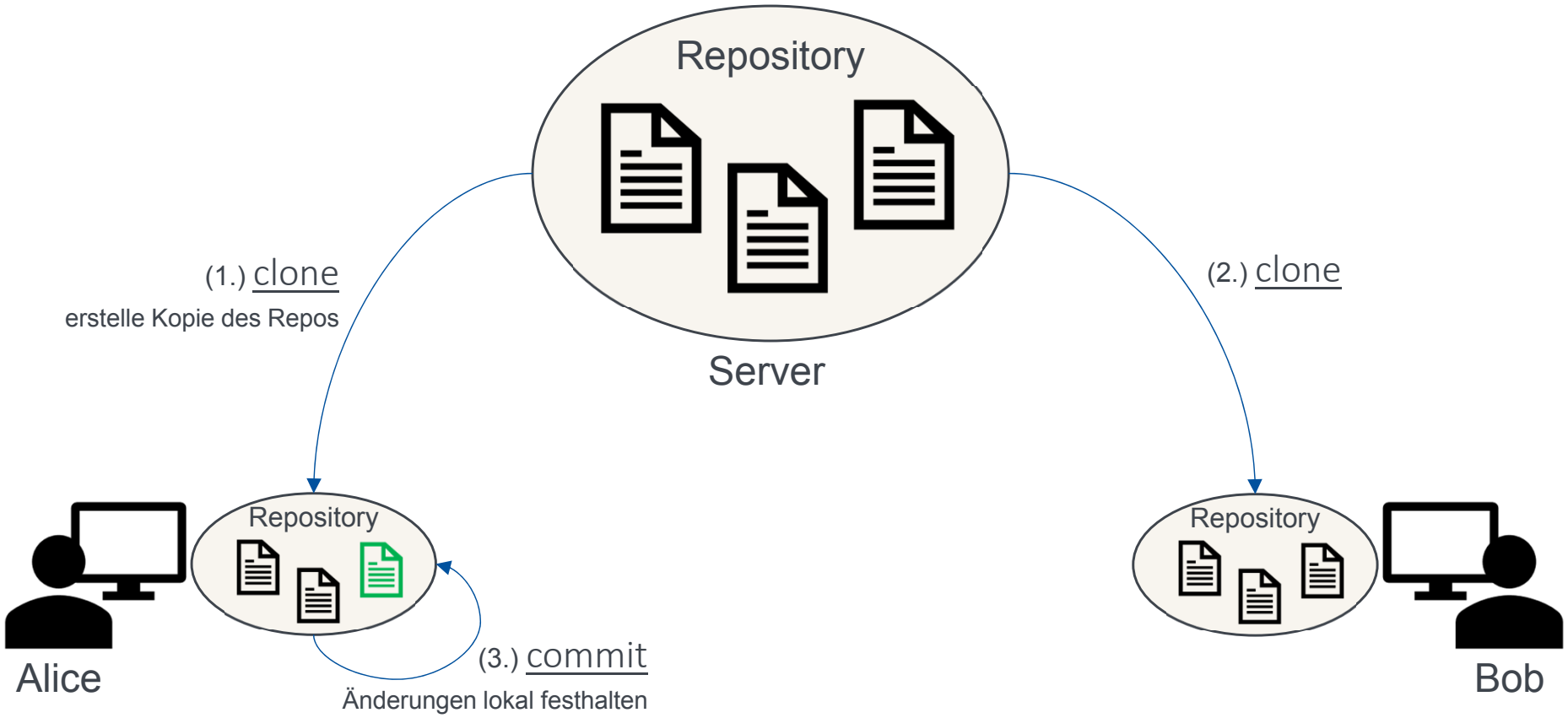
  - Problem: Gleichzeitige Bearbeitung einer Datei ist nicht möglich

- **Konzept 2: Copy-Modify-Merge**
  - Jeder Entwickler kopiert die zentral gelegenen Dateien an einen lokalen Ort (**copy**).
  - Änderungen führt jeder Entwickler lokal durch (**modify**)
  - Anschließend werden diese wieder an den zentralen Ort übertragen und dabei zusammengeführt (**merge**).

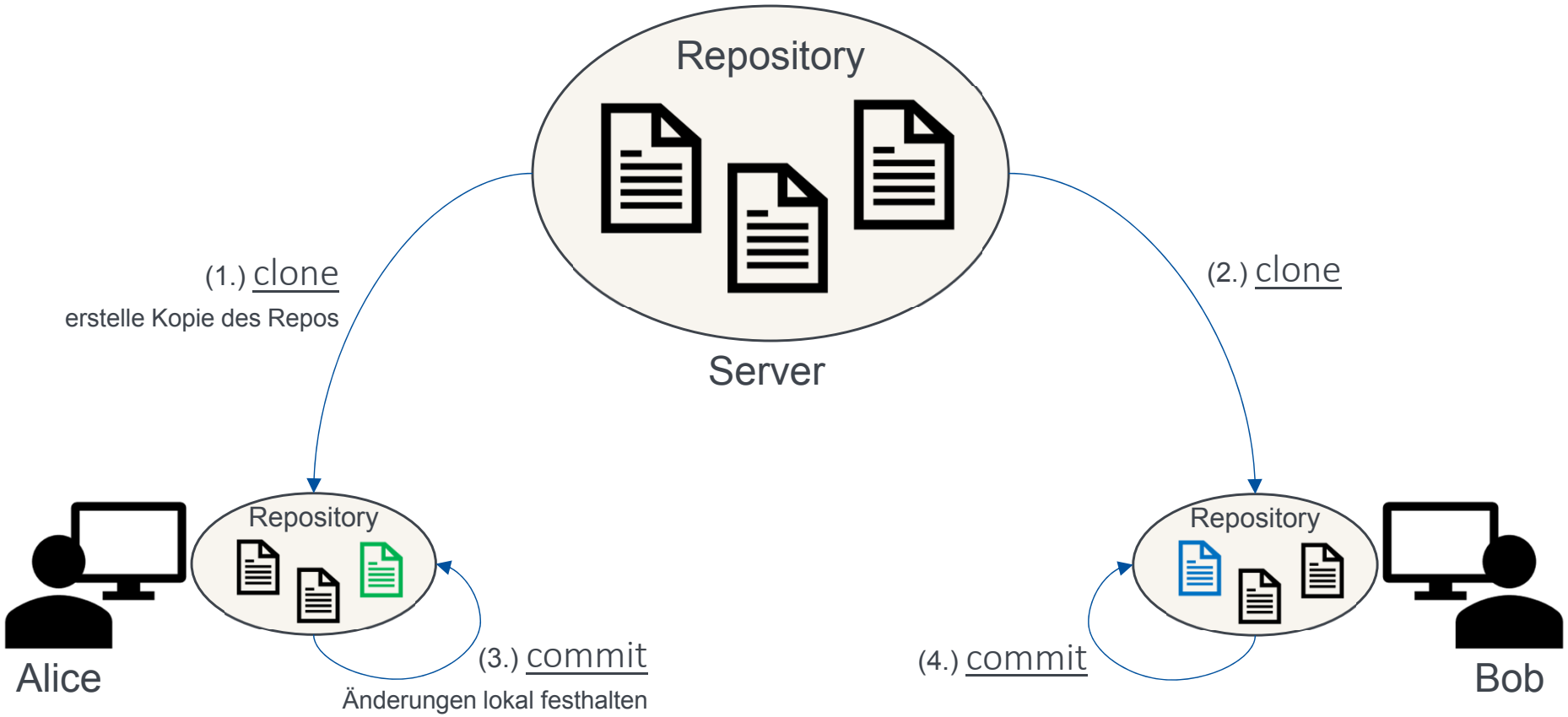
Copy-Modify-Merge



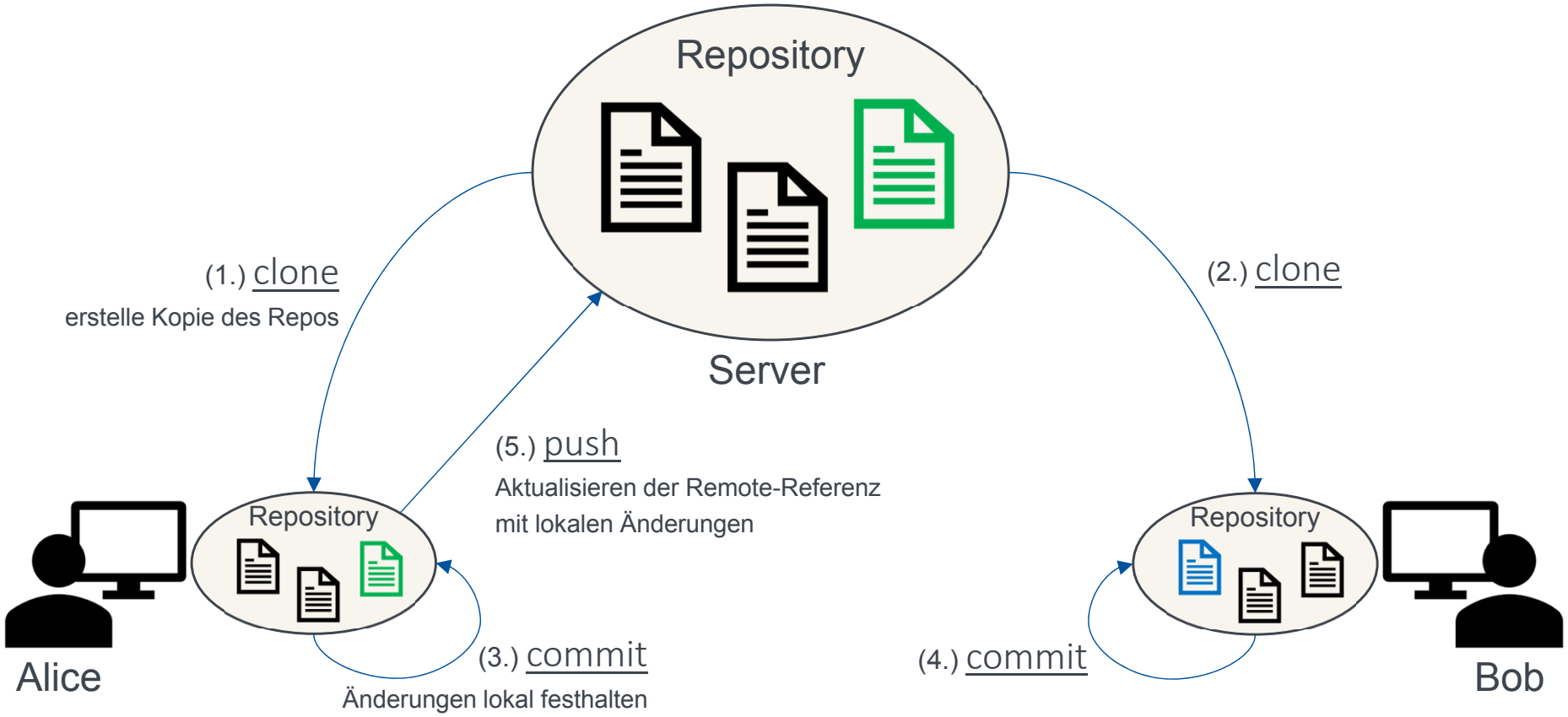
Copy-Modify-Merge



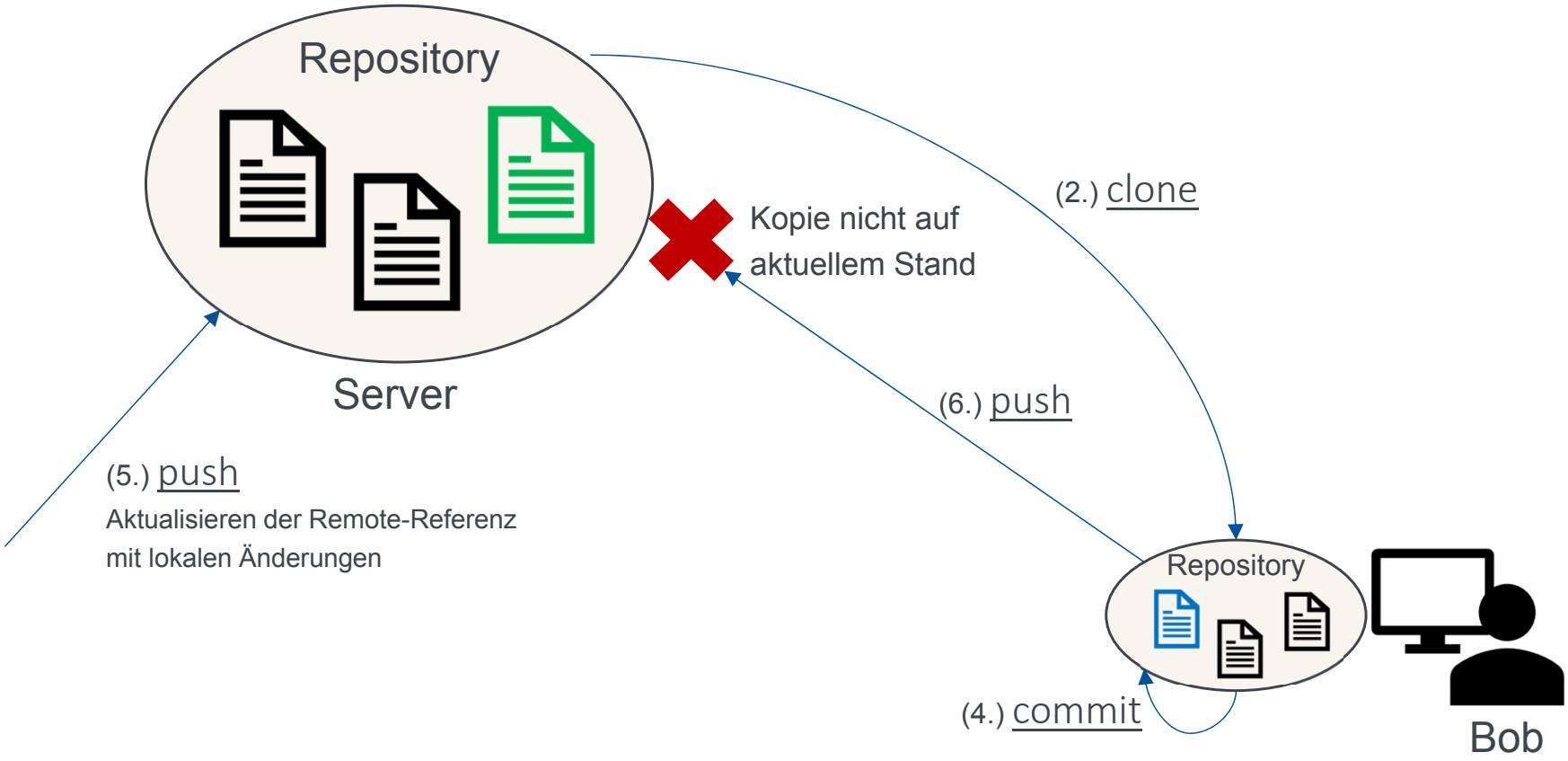
Copy-Modify-Merge



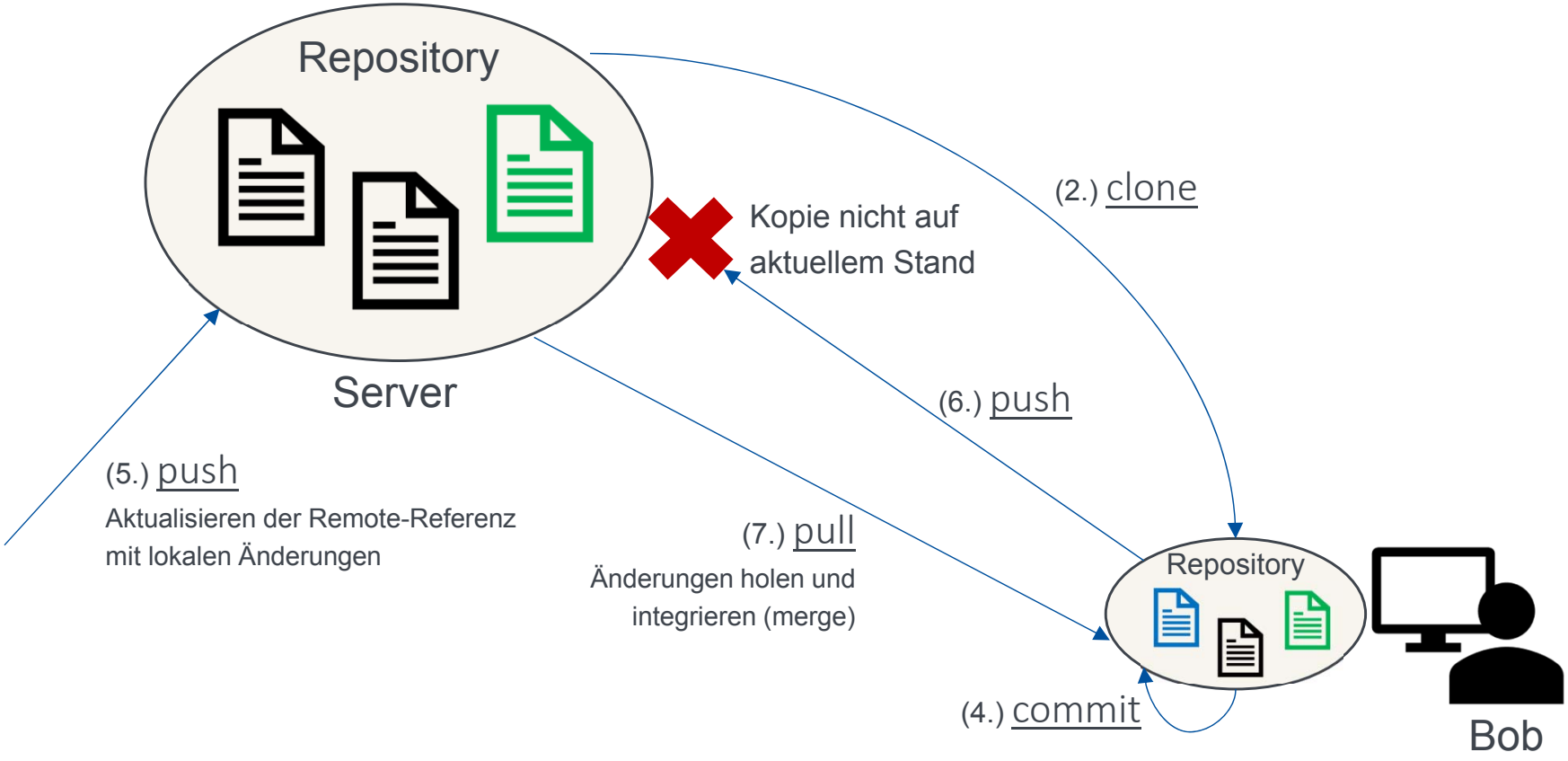
Copy-Modify-Merge



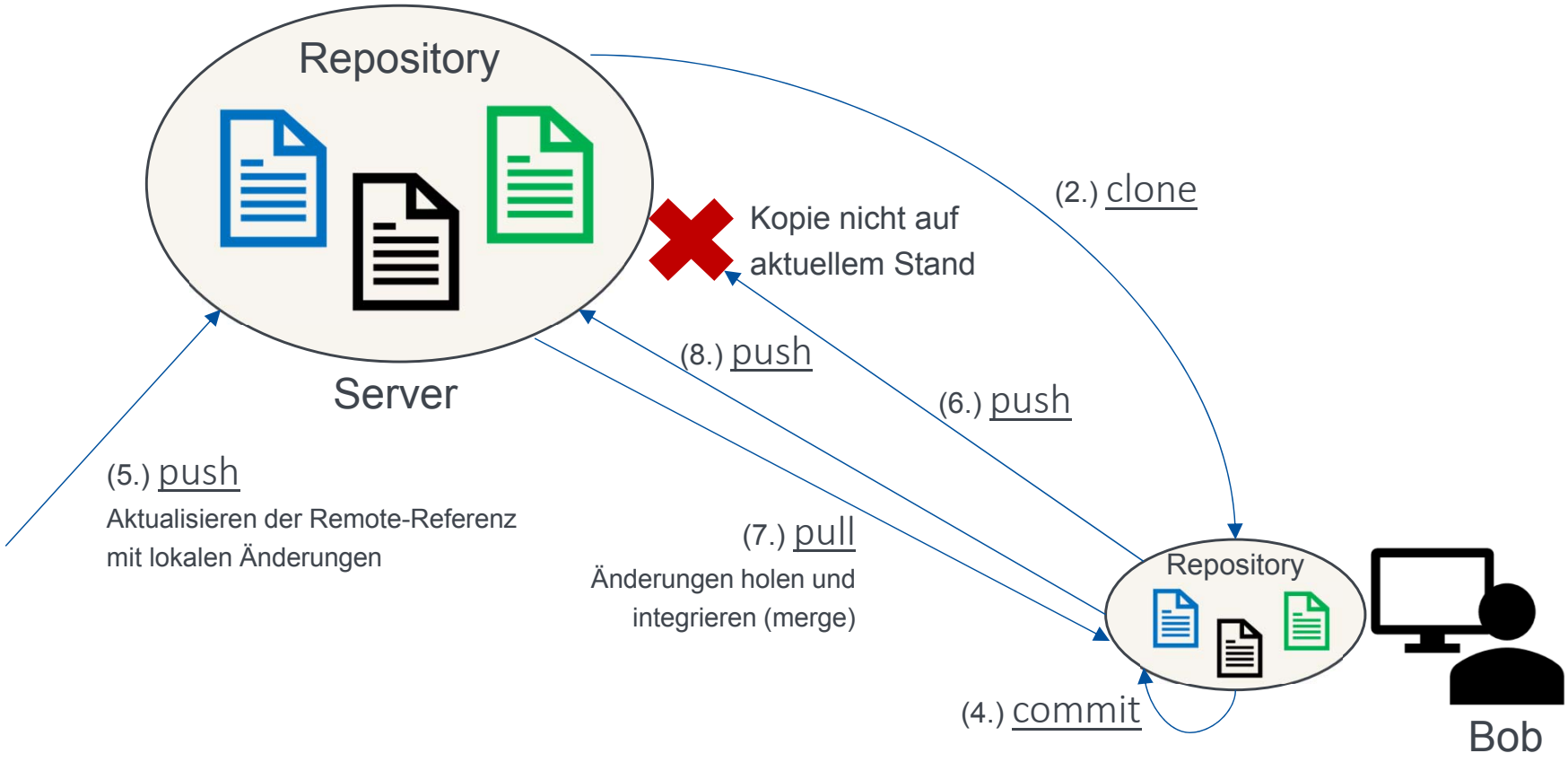
Copy-Modify-Merge



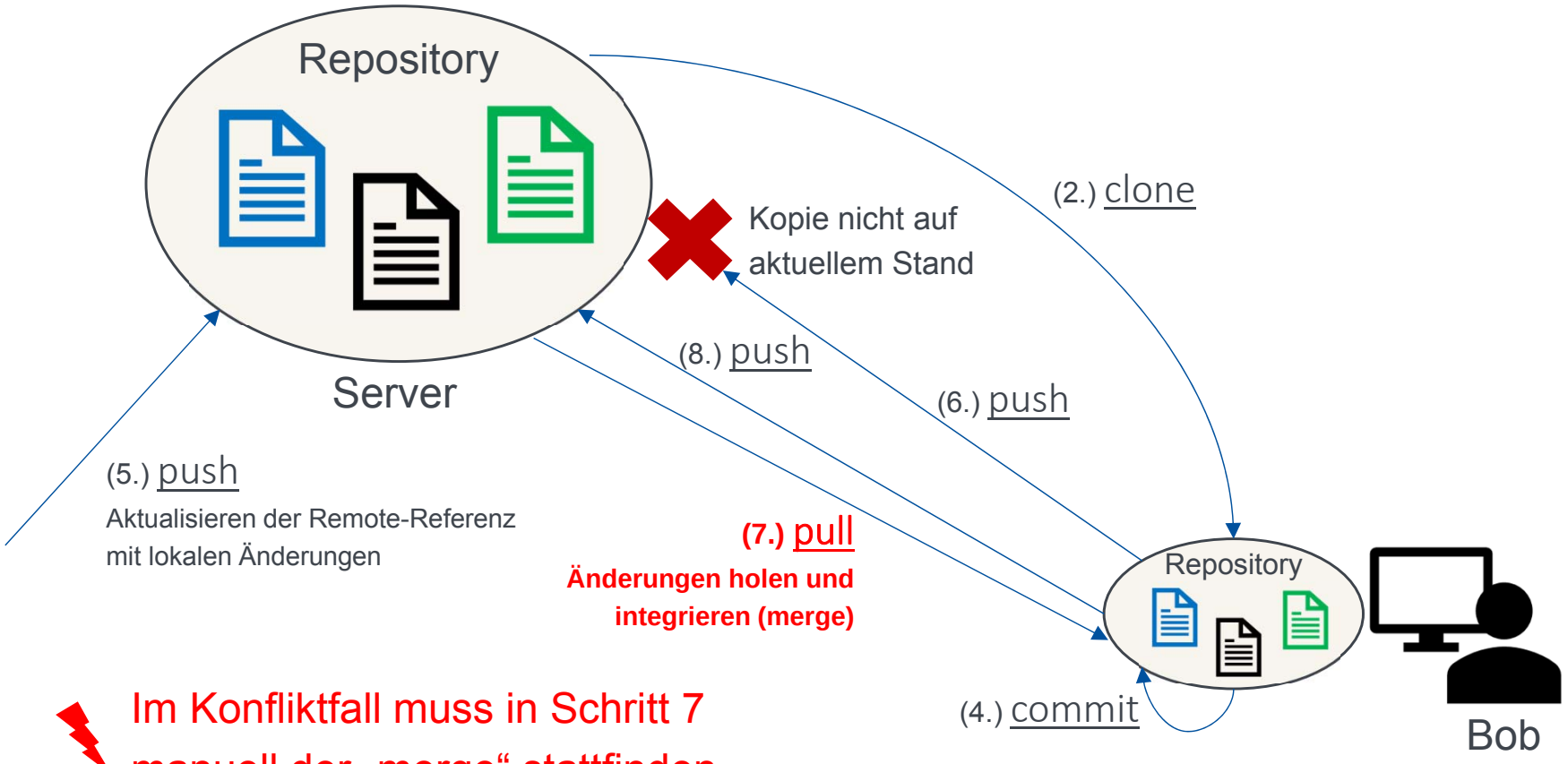
Copy-Modify-Merge



Copy-Modify-Merge



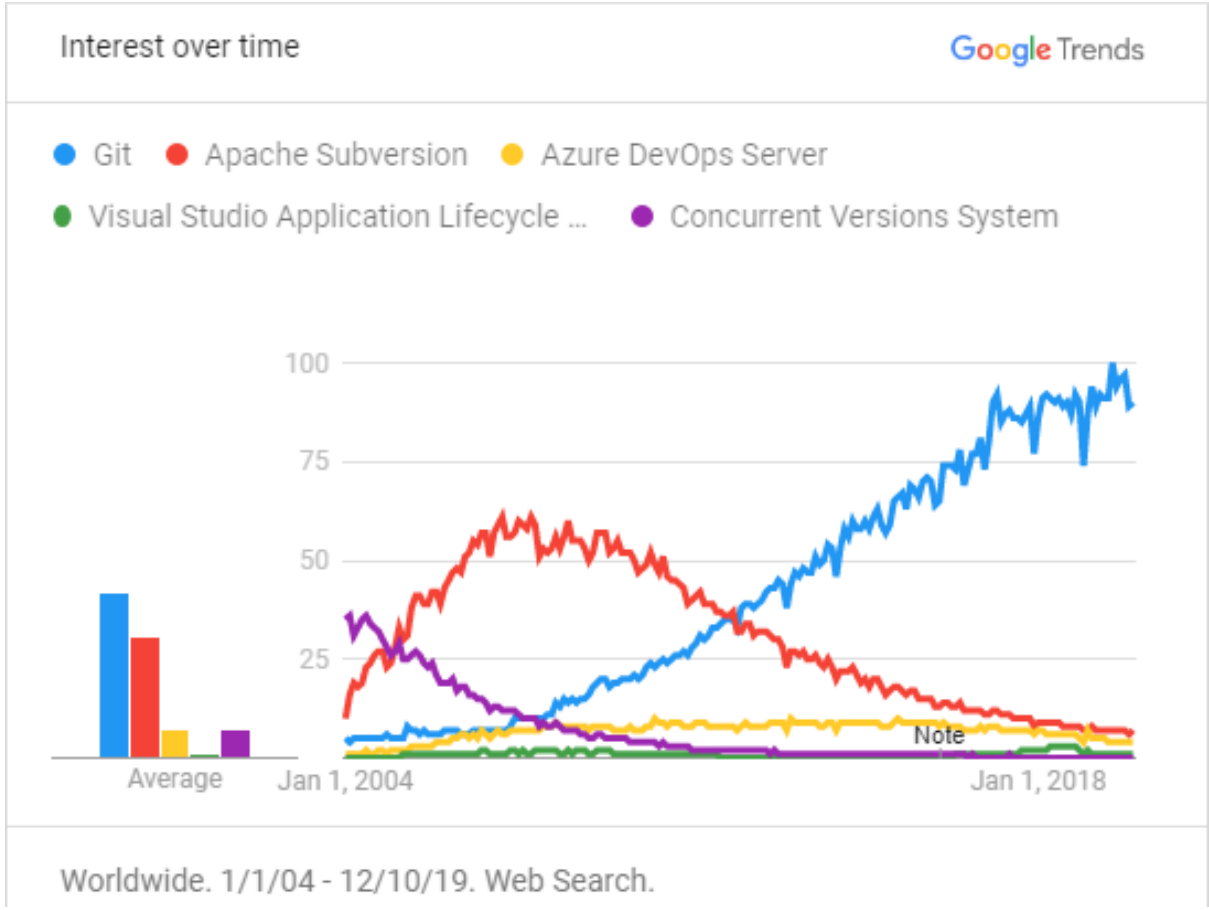
Copy-Modify-Merge



Im Konfliktfall muss in Schritt 7
manuell der „merge“ stattfinden

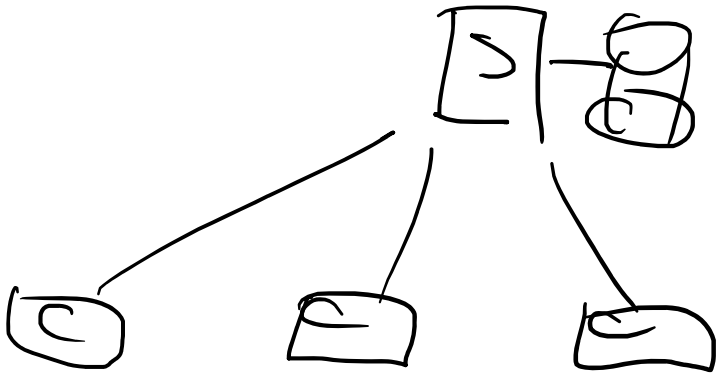
Git

Versionsverwaltungssysteme im Vergleich



Git

- Git ist ein (verteiltes) Versionsverwaltungssystem
- basiert auf dem Konzept **Copy-Modify-Merge**
- Kostenlos, Open-Source
- Git Clients: https://git-scm.com/downloads
- GUI Clients: ermöglichen es, in den meisten Fällen auf Konsoleneingaben zu verzichten und bieten mehr Komfort (dafür evtl. langsamer)
- GitHub Desktop: https://desktop.github.com/



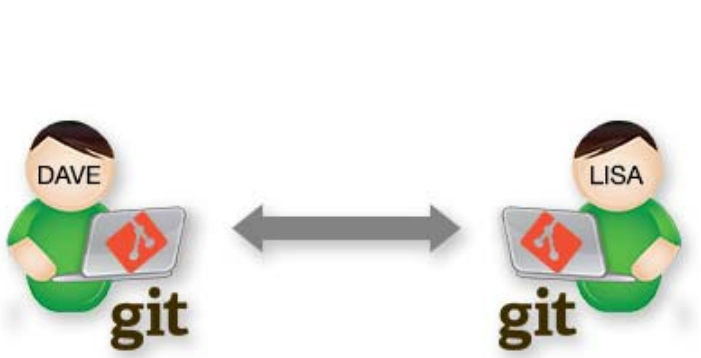
GitHub

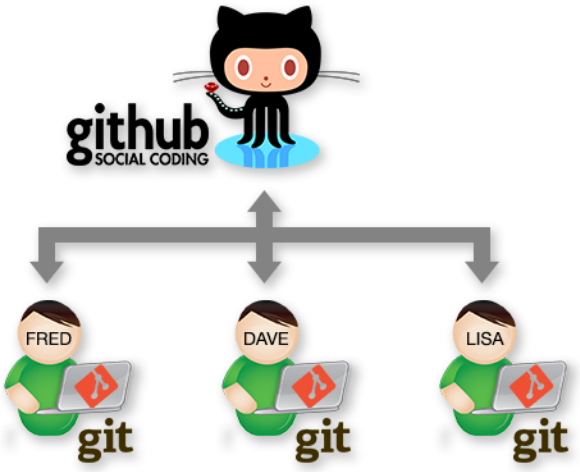
- www.github.com
- Onlinedienst und Marktplatz zur Verwaltung von Projekten
- basiert auf einem integrierten Git VCS
- Tracking sämtlicher Änderungen aller Autoren eines Projekts
- Wiki-System, Issue Tracking, Code Reviews, etc.
- kostenlos für Open Source Projekte
- große Community mit 100 Mio. Entwicklern und 420 Mio. Projekte [1]

[1] www.github.com, Jan. 2024

GitHub

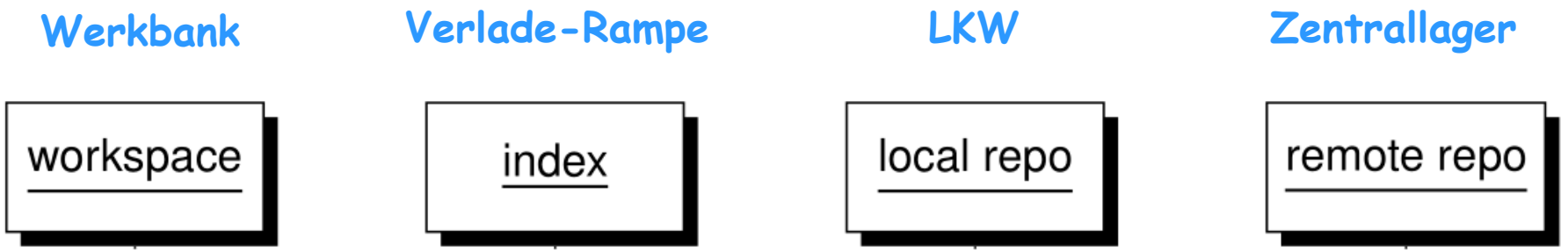

git
≠
github
SOCIAL CODING


DAVE
LISA
git
git

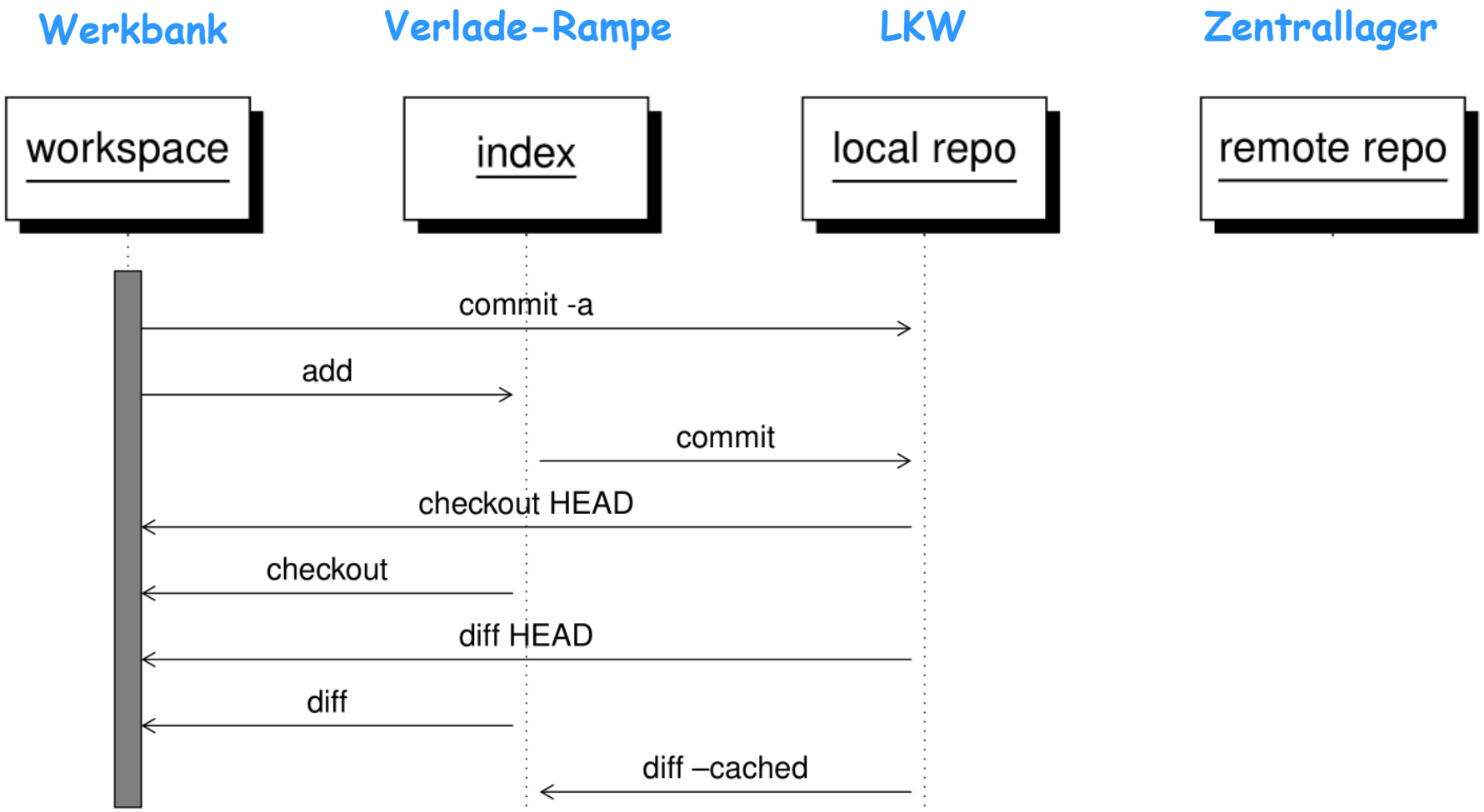

github
SOCIAL CODING
FRED
DAVE
LISA
git
git
git

Git Kommandos

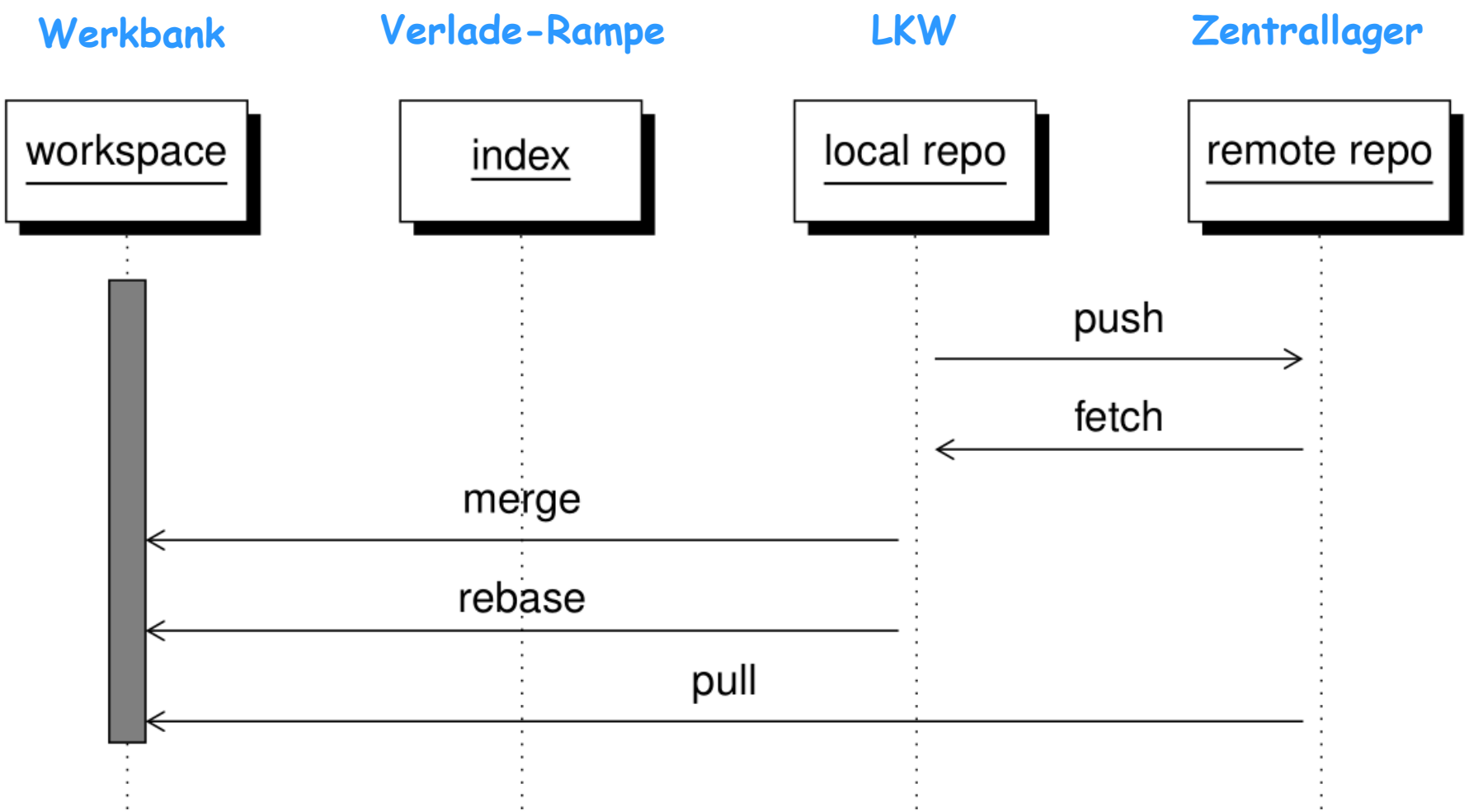
Git Kommandos im Überblick



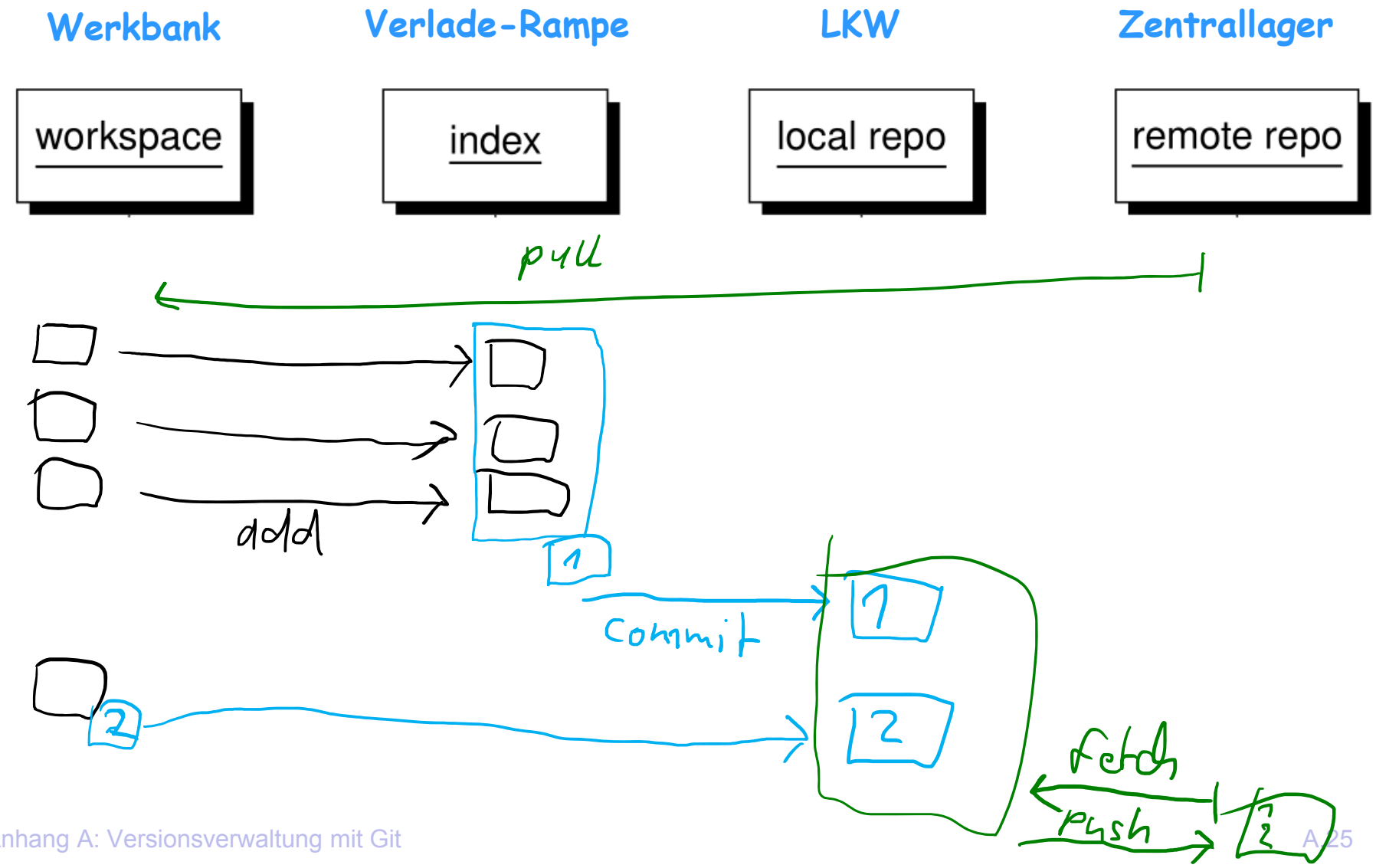
Git Kommandos – Lokal



Git Kommandos – Remote



Git Kommandos im Überblick



Git Kommandos – Lokal

- Dokumentation

  https://git-scm.com

  https://git-scm.com/doc

  https://git-scm.com/book/de/v2

- Tutorials/Referenzen

  http://try.github.io/ → Git Handbook

  http://marklodato.github.io/visual-git-guide/index-en.html

https://xkcd.com/1597/