

# Rechteverwaltung in Git innerhalb eines Repository

## Abschlussprojekt zum Fachinformatiker für Systemintegration

Martin Heckel

GK Software SE

2019-07-04

# Gliederung

## Ausgangssituation

- Ist-Zustand

- Soll-Zustand

## Umsetzung

- Ansätze

- Synchronisation

- Zusammenführen der Ordnerstrukturen

## Auswertung

# Ist-Zustand

# Ist-Zustand

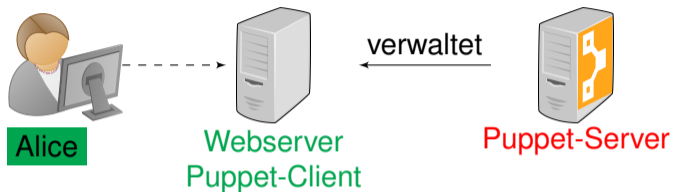


Alice

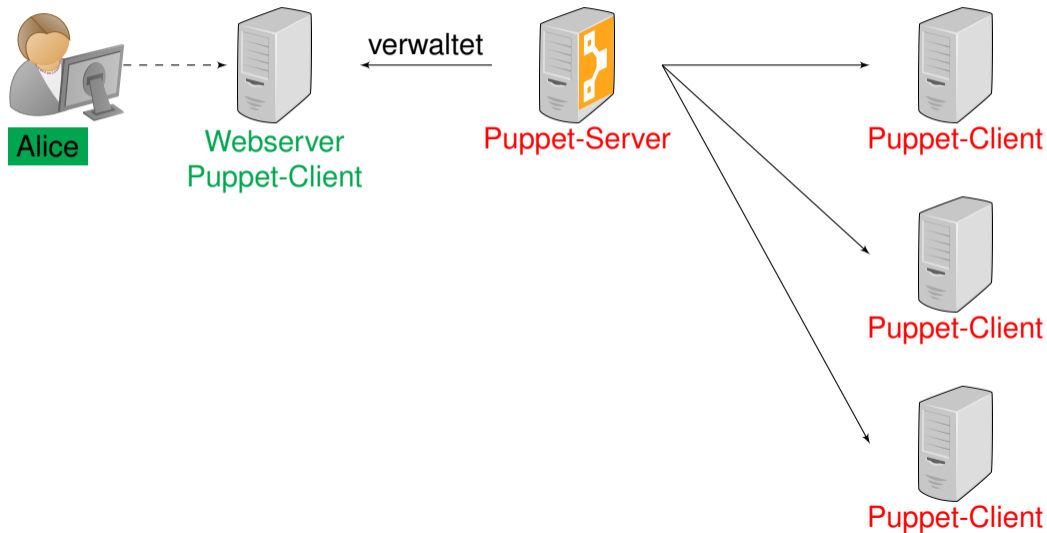
## Ist-Zustand



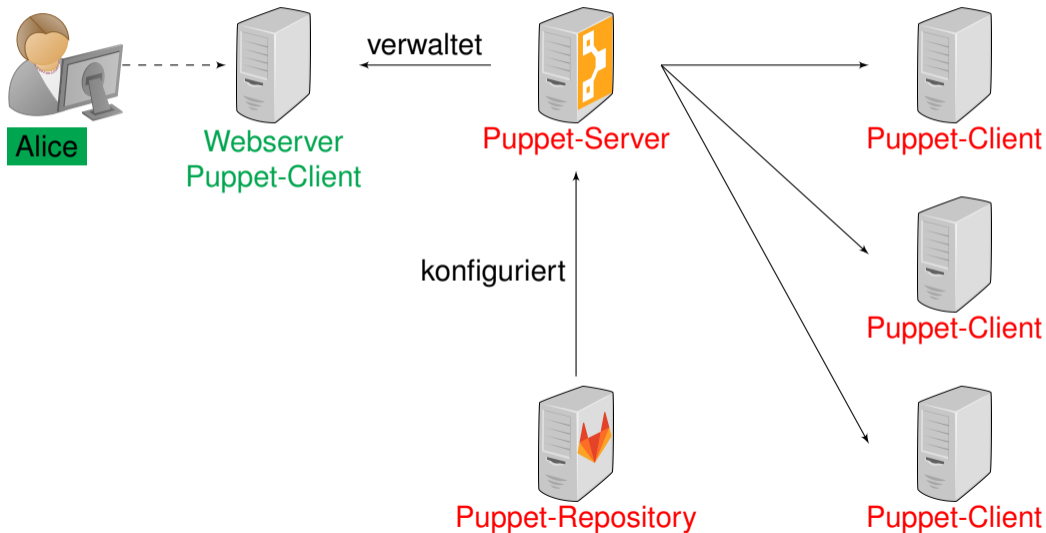
## Ist-Zustand



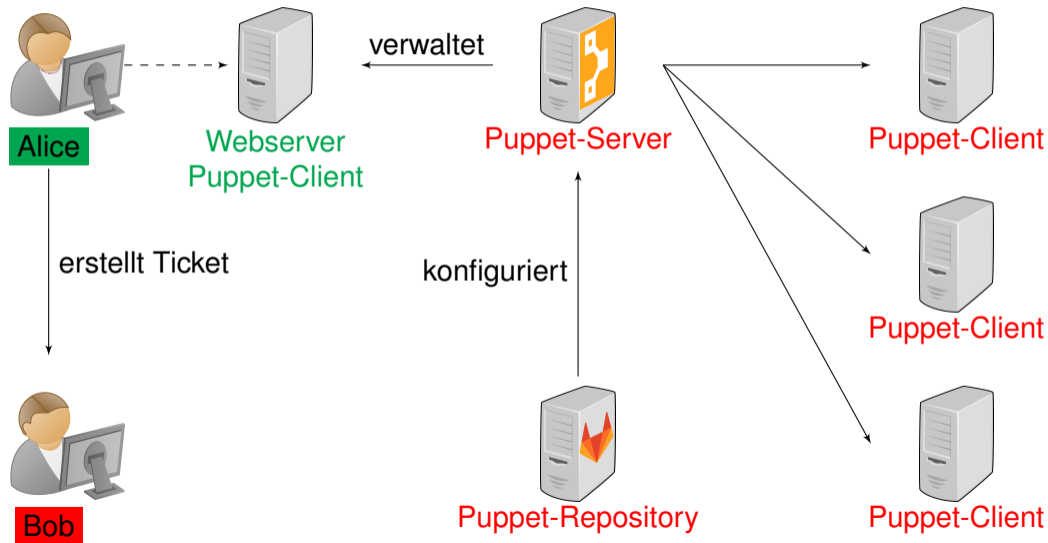
## Ist-Zustand



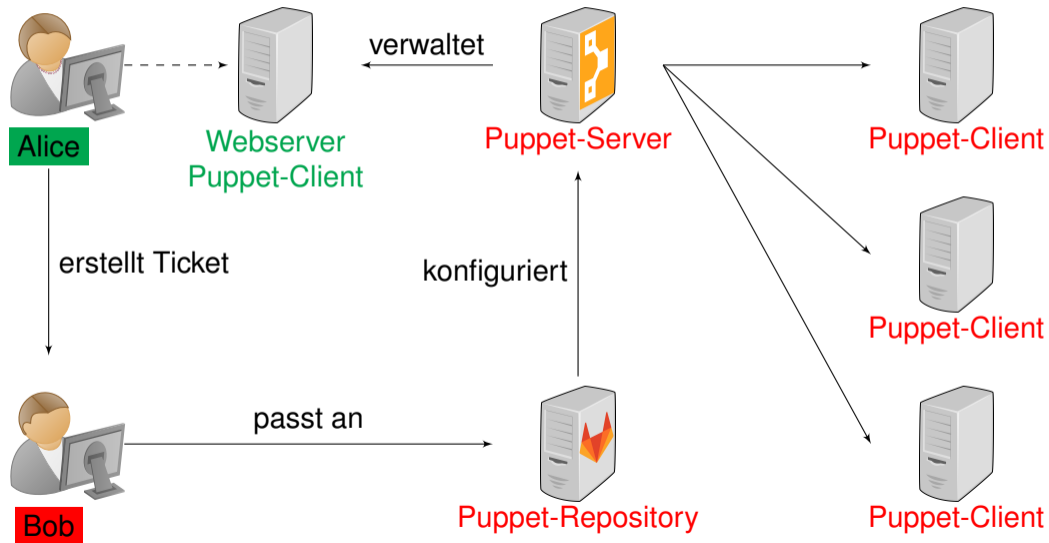
## Ist-Zustand



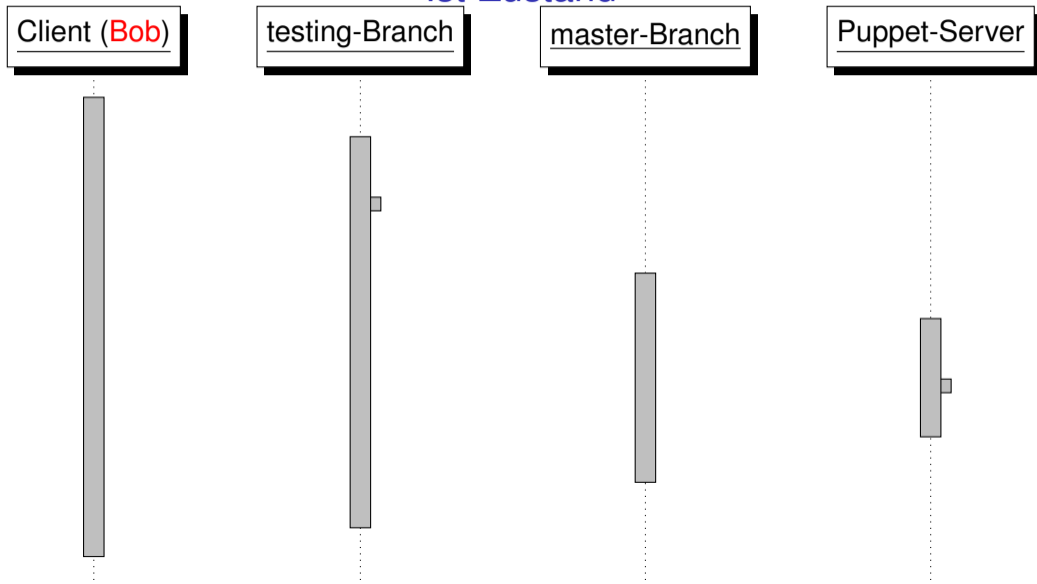
## Ist-Zustand



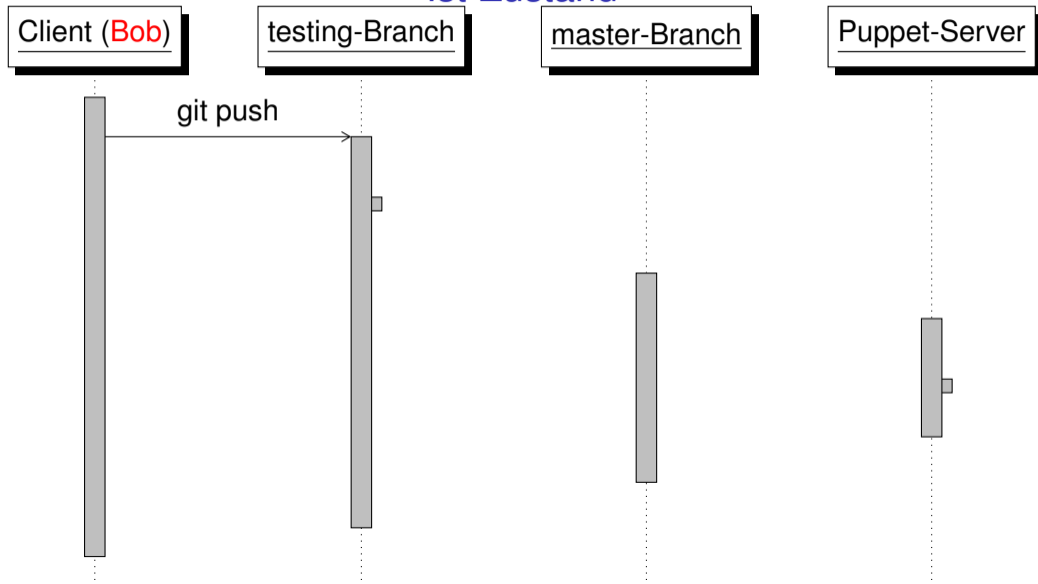
## Ist-Zustand



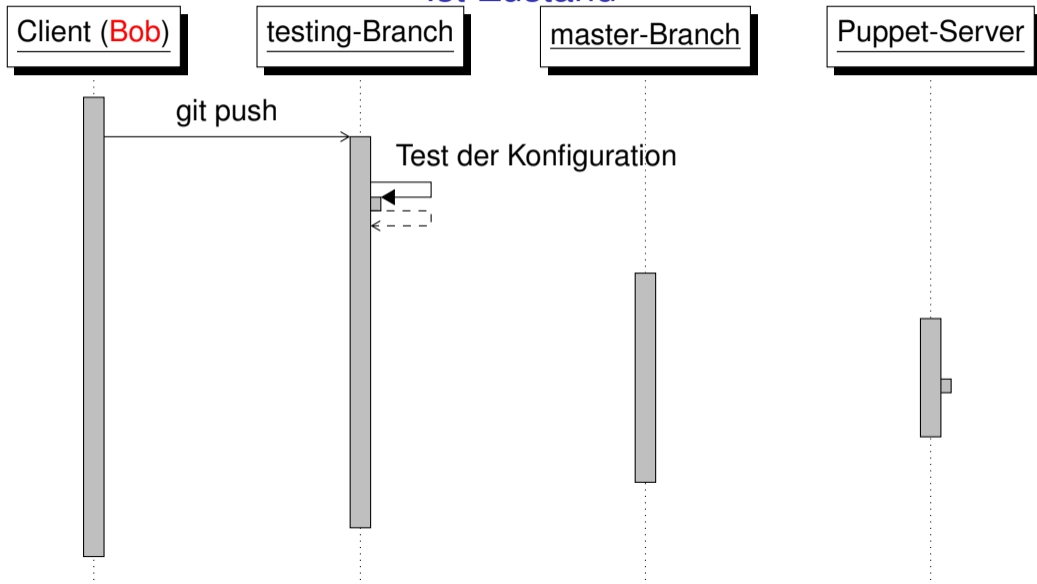
## Ist-Zustand



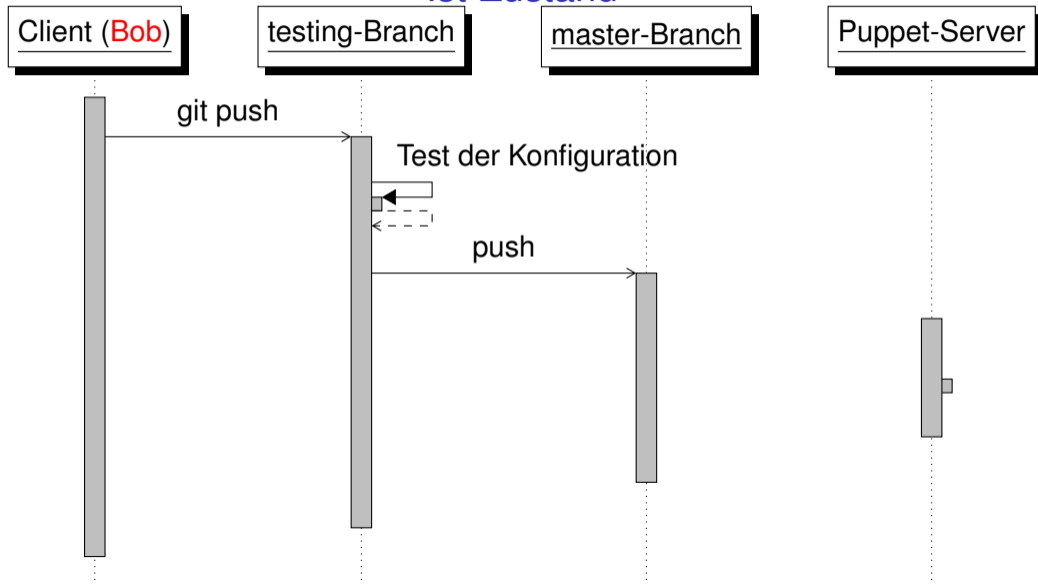
## Ist-Zustand



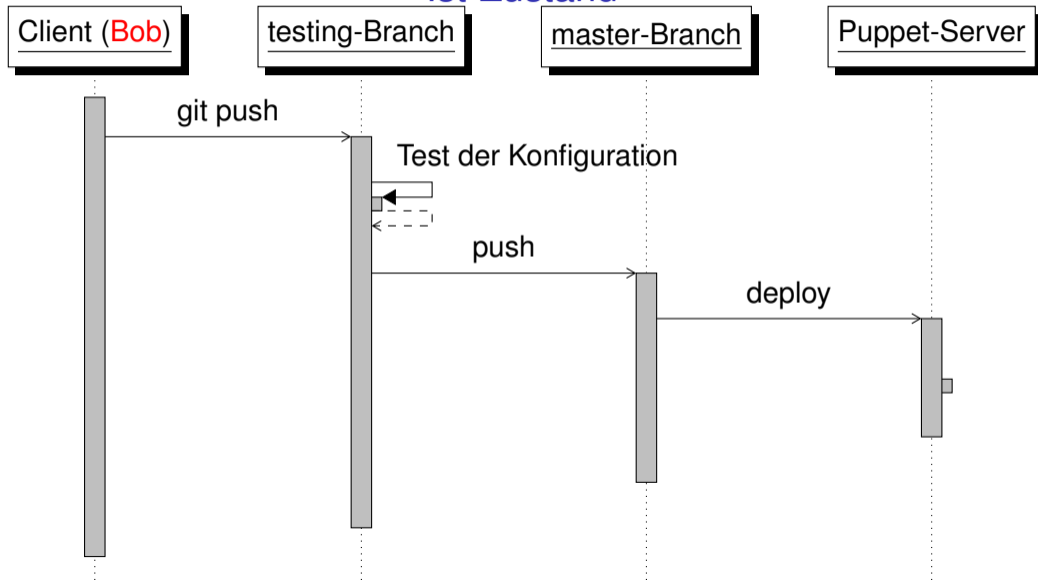
## Ist-Zustand



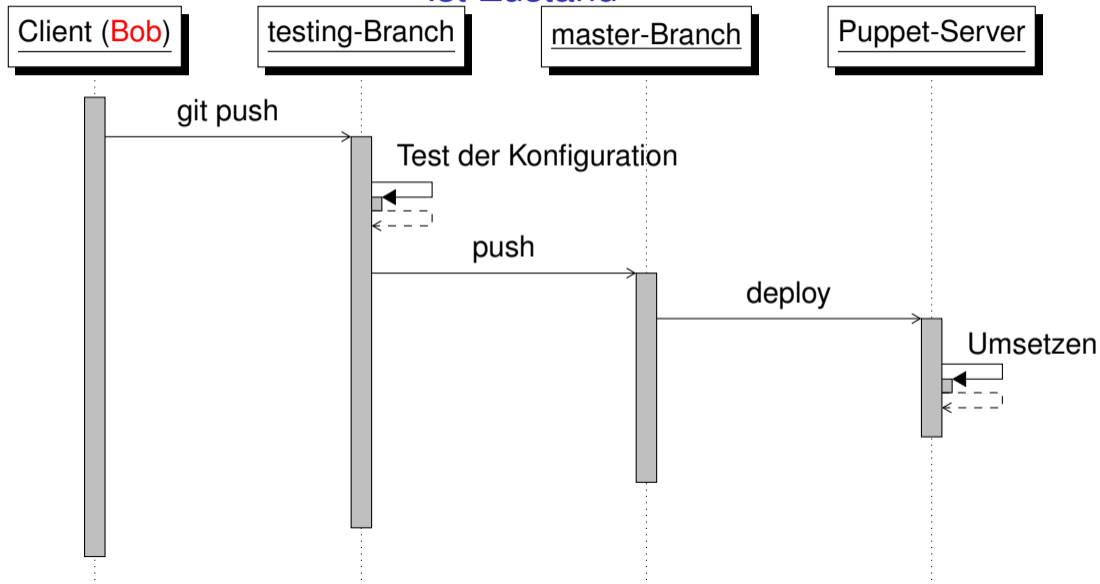
## Ist-Zustand



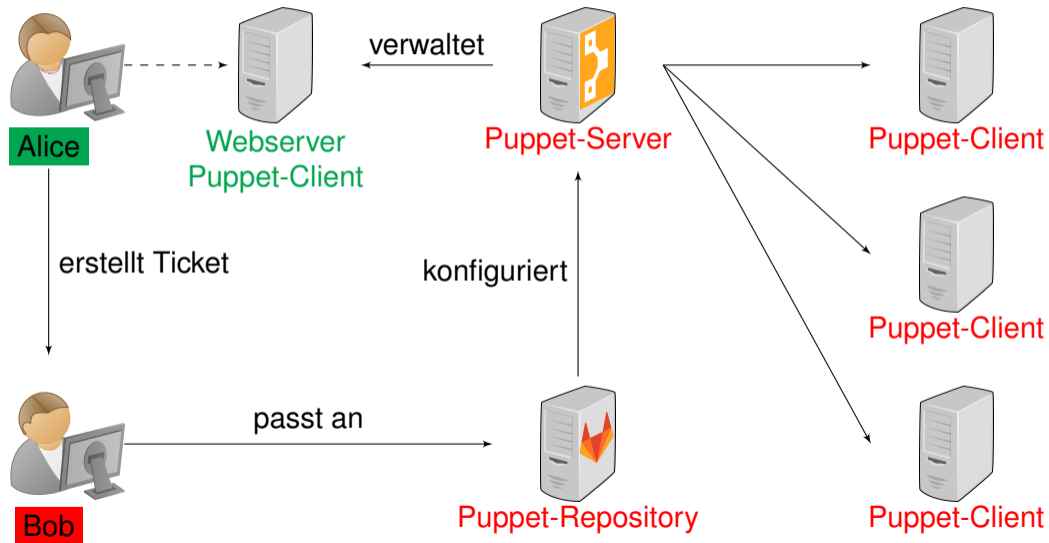
## Ist-Zustand



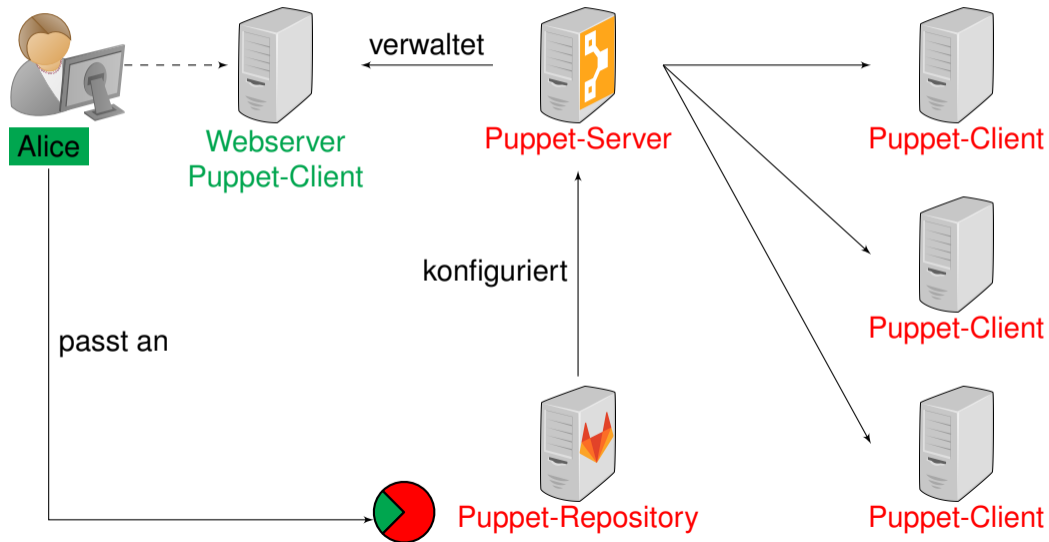
## Ist-Zustand



## Ist-Zustand



## Soll-Zustand



# Ansätze

# Ansätze

- Berechtigungen auf Dateiebene setzen

## Ansätze

- Berechtigungen auf Dateiebene setzen
- Verwaltung in mehreren Repositories und Synchronisation

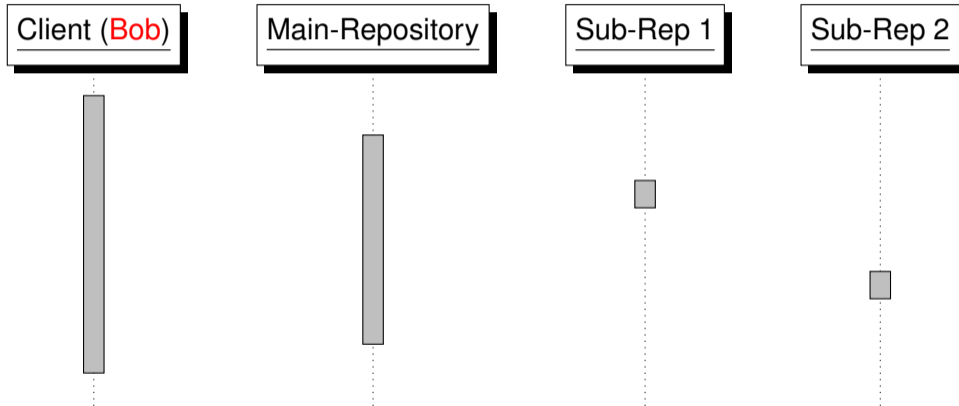
## Ansätze

- Berechtigungen auf Dateiebene setzen
- Verwaltung in mehreren Repositories und Synchronisation
  - Submodules

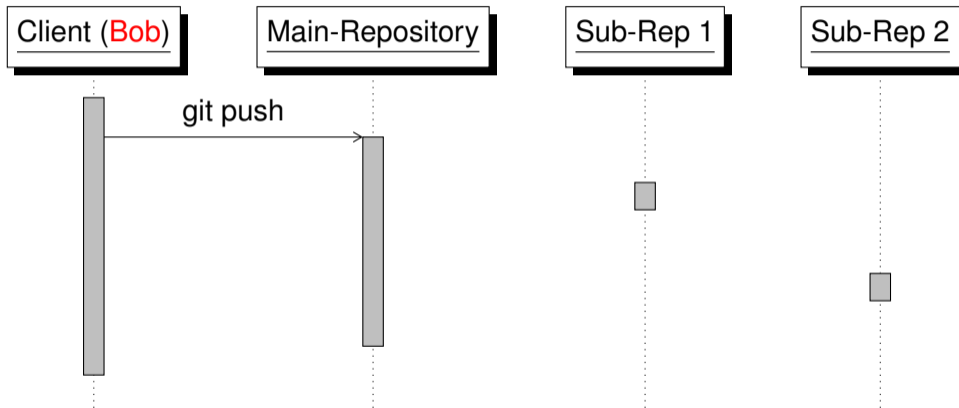
# Ansätze

- Berechtigungen auf Dateiebene setzen
- Verwaltung in mehreren Repositories und Synchronisation
  - Submodules
  - Subtrees

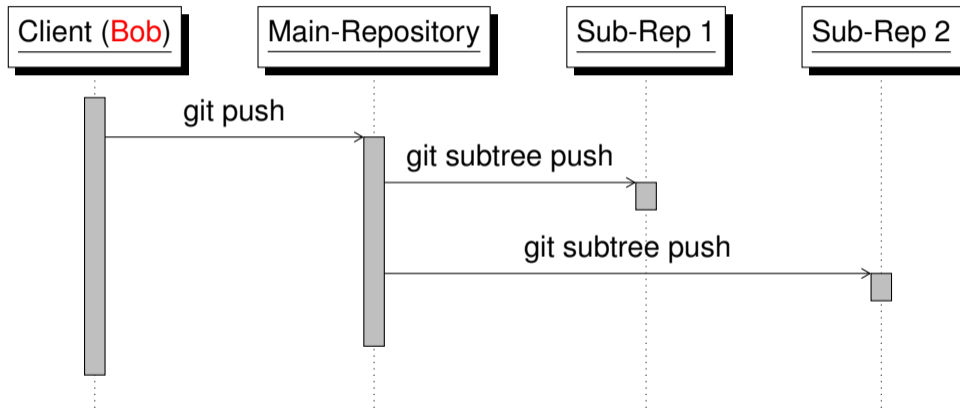
# Synchronisation vom übergeordneten Repository



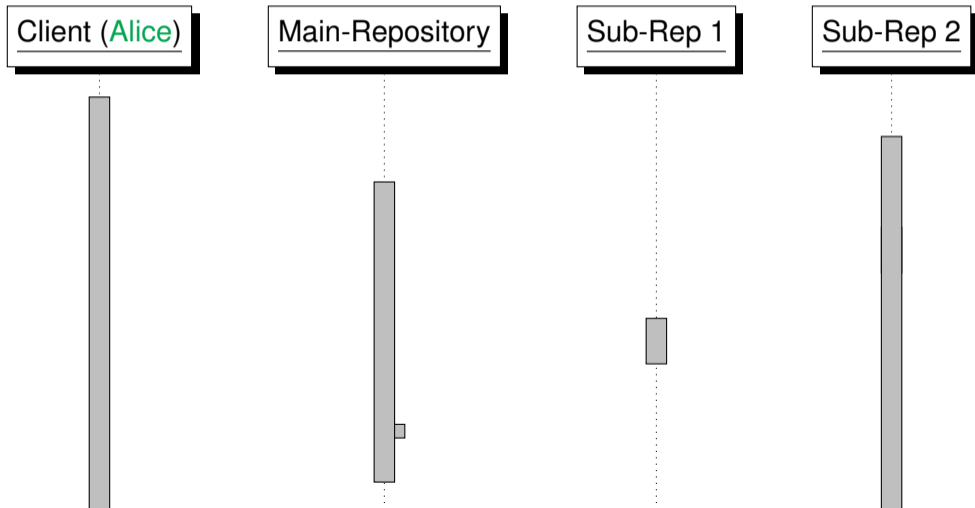
## Synchronisation vom übergeordneten Repository



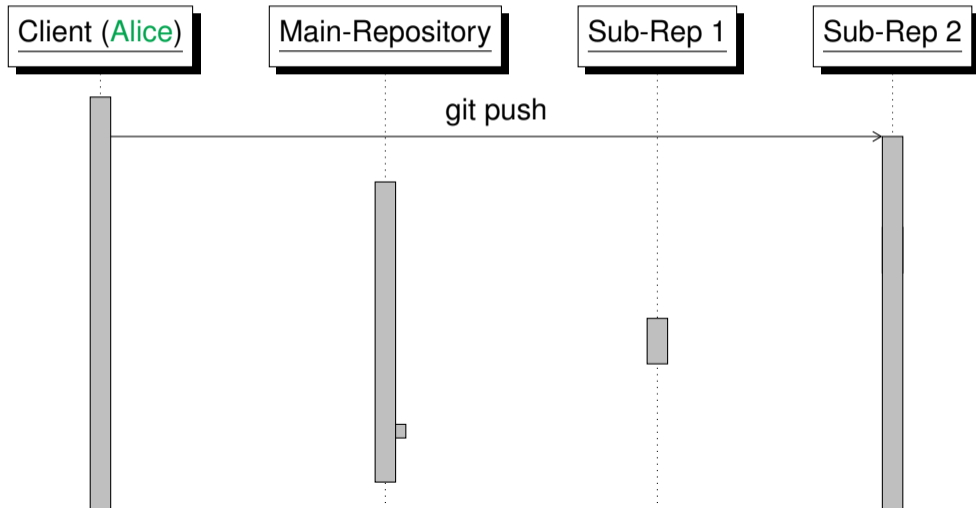
## Synchronisation vom übergeordneten Repository



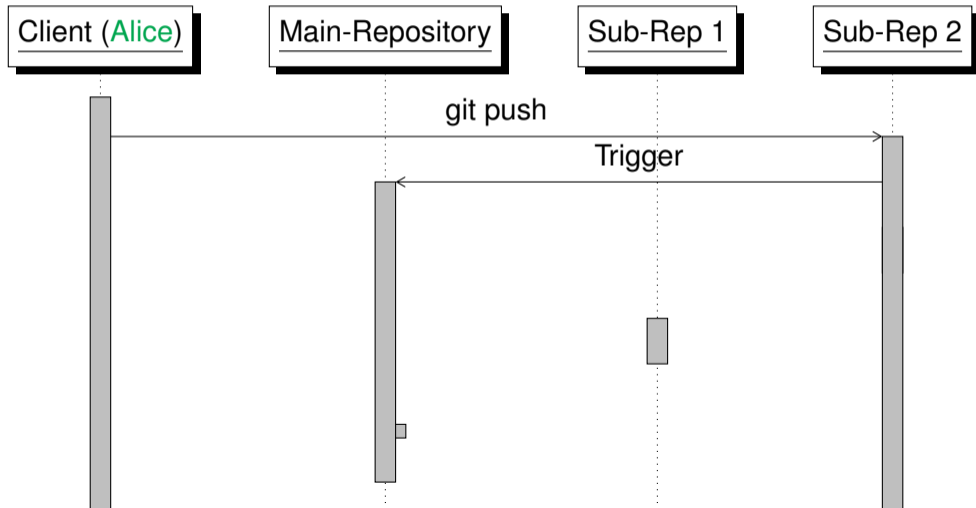
## Synchronisation vom untergeordneten Repository



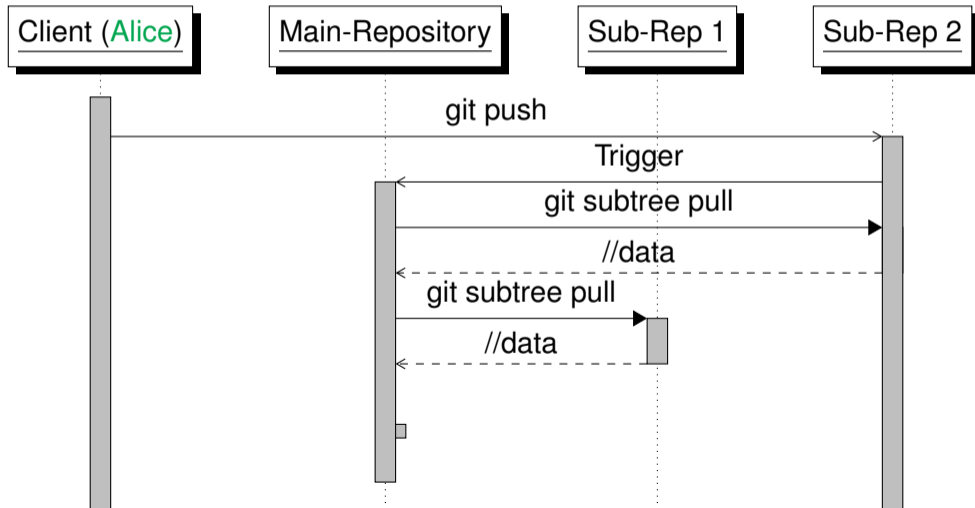
## Synchronisation vom untergeordneten Repository



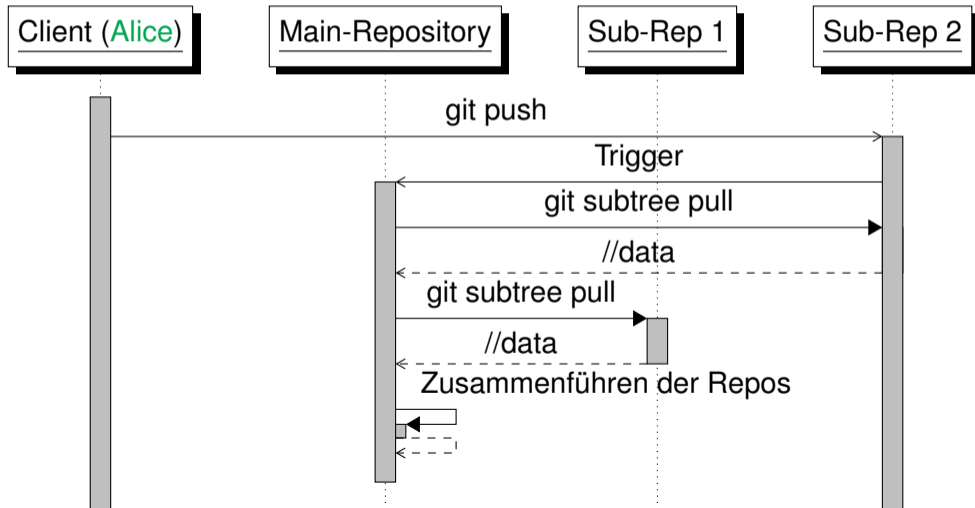
## Synchronisation vom untergeordneten Repository



## Synchronisation vom untergeordneten Repository



## Synchronisation vom untergeordneten Repository



# Zusammenführen der Ordnerstrukturen

## Zusammenführen der Ordnerstrukturen

- Dateien/Verzeichnisse müssen in bestimmten Struktur liegen

## Zusammenführen der Ordnerstrukturen

- Dateien/Verzeichnisse müssen in bestimmten Struktur liegen
- Subtrees können nur in eigenständigen Ordnern verwaltet werden

## Zusammenführen der Ordnerstrukturen

- Dateien/Verzeichnisse müssen in bestimmten Struktur liegen
- Subtrees können nur in eigenständigen Ordnern verwaltet werden
- Dateien der Subrepositories werden durch Symlinks an den entsprechenden Stellen im Puppet-Verzeichnis repräsentiert

# Auswertung

## Auswertung

- Alice kann ihre Anpassungen direkt auf den Server anwenden und muss nicht warten

## Auswertung

- Alice kann ihre Anpassungen direkt auf den Server anwenden und muss nicht warten
- Sollte Alice ein fehlerhaftes Script auf den Server geladen haben, kann sie es schnell selbst aktualisieren

## Auswertung

- Alice kann ihre Anpassungen direkt auf den Server anwenden und muss nicht warten
- Sollte Alice ein fehlerhaftes Script auf den Server geladen haben, kann sie es schnell selbst aktualisieren
- Bob muss sich nicht um die von Alice gewünschten Anpassungen kümmern

## Auswertung

- Alice kann ihre Anpassungen direkt auf den Server anwenden und muss nicht warten
- Sollte Alice ein fehlerhaftes Script auf den Server geladen haben, kann sie es schnell selbst aktualisieren
- Bob muss sich nicht um die von Alice gewünschten Anpassungen kümmern
- Die Wahrscheinlichkeit von Fehlern ist geringer, da weniger Mitarbeiter am Prozess beteiligt sind

# Auswertung



Trent



Alice



Bob

# Gitlab-Pipelines

```
stages :  
  - pull_subtrees  
  - push_subtrees  
  - test  
  - maintain_branches  
  - deploy  
  
pull_subtrees :  
  script :  
    - ./scripts/pull_subtrees.sh  
  tags :  
    - puppet-merge  
  only :  
    variables :  
      - $PULL == "true"  
  
push_subtrees :  
  script :  
    - ./scripts/push_subtrees.sh  
  tags :  
    - puppet-merge  
  only :  
    - testing  
  except :  
    variables :  
      - $PULL == "true"
```

# Gitlab-Pipelines

```
trigger_build:
  script:
    - "curl -X POST -F token=XXXXXXXXXXXXXXXXXXXXXXXXXXXX -F ref=testing -F 'variables[PULL]=true' \
      <git-resource>"
  except:
    variables:
      - $ERR == "true"
notify_error:
  script:
    - "bash sendMails.sh \"$ERRMSG\""
    - "exit 1"
  only:
    variables:
      - $ERR == "true"
```