integrata cegos

Git

Scratch zum Webinar vom 9.-10.1.2020

Informationen

- Referent
  - Rainer Sawitzki
  - training@rainer-sawitzki.de
- 4 Sessions, jeweils 2,5 Stunden
  - Inklusive 15 Minuten Pause
- Ablauf
  - Vortrag
    - Etwa 45 Minuten
  - Präsentation
    - Etwa 45 Minuten
  - Zur Auflockerung kleinere Übungseinheiten
    - Jede 5 – 10 Minuten
  - Fragerunde
    - Nach Bedarf
    - Am Ende, etwa 15 Minuten

Ressourcen

- Dieser Scratch
- Die Präsentation
- Online
  - https://github.com/Javacream/org.javacream.training.gitscm
    - Ein paar Beispiele und Skripte auf GitHub
  - https://git-scm.com/download/win
    - Download Git Portable
  - https://git-scm.com/docs
    - Dokumentation

Git Repositories

- Normales Verzeichnis
- + .git => Git Project Directory
- .git
  - Das Git Repository
- Alles andere: Workspace
- Git Repository selbst:
  - Staging Area
  - Internes Repository
  - Stashes

Git Begriffe

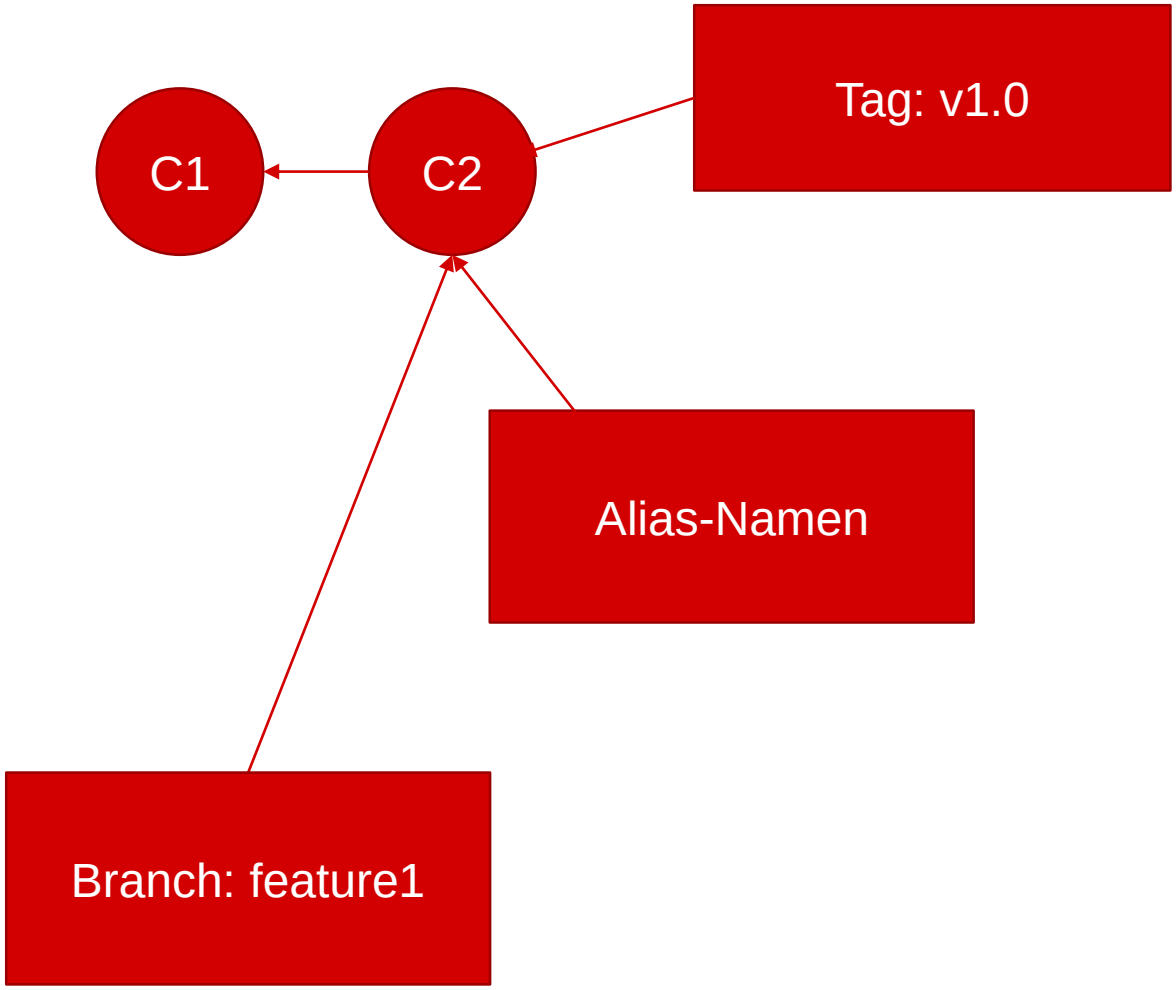
- Content-Objekte
  - Dateien
    - Tree, +++
- Commit-Objekte
  - Liste aller Content-Objekte des aktuellen Standes
  - Autor, Timestamp, Commit-Message

Befehle

- git
  - config –global
    - User, eMail
  - init
  - add
    - Workspace – Staging
  - Commit
    - -m <Commit-Message>
    - Commit-Objekt wird erstellt
    - Und über Hash identifizierbar gemacht
  - status
    - Aktuellen Stand und Diskrepanzen
  - log
    - Aktuelle Commit-Objekte
      - Hash
      - Autor
      - Date
      - Commit-Message

Commits

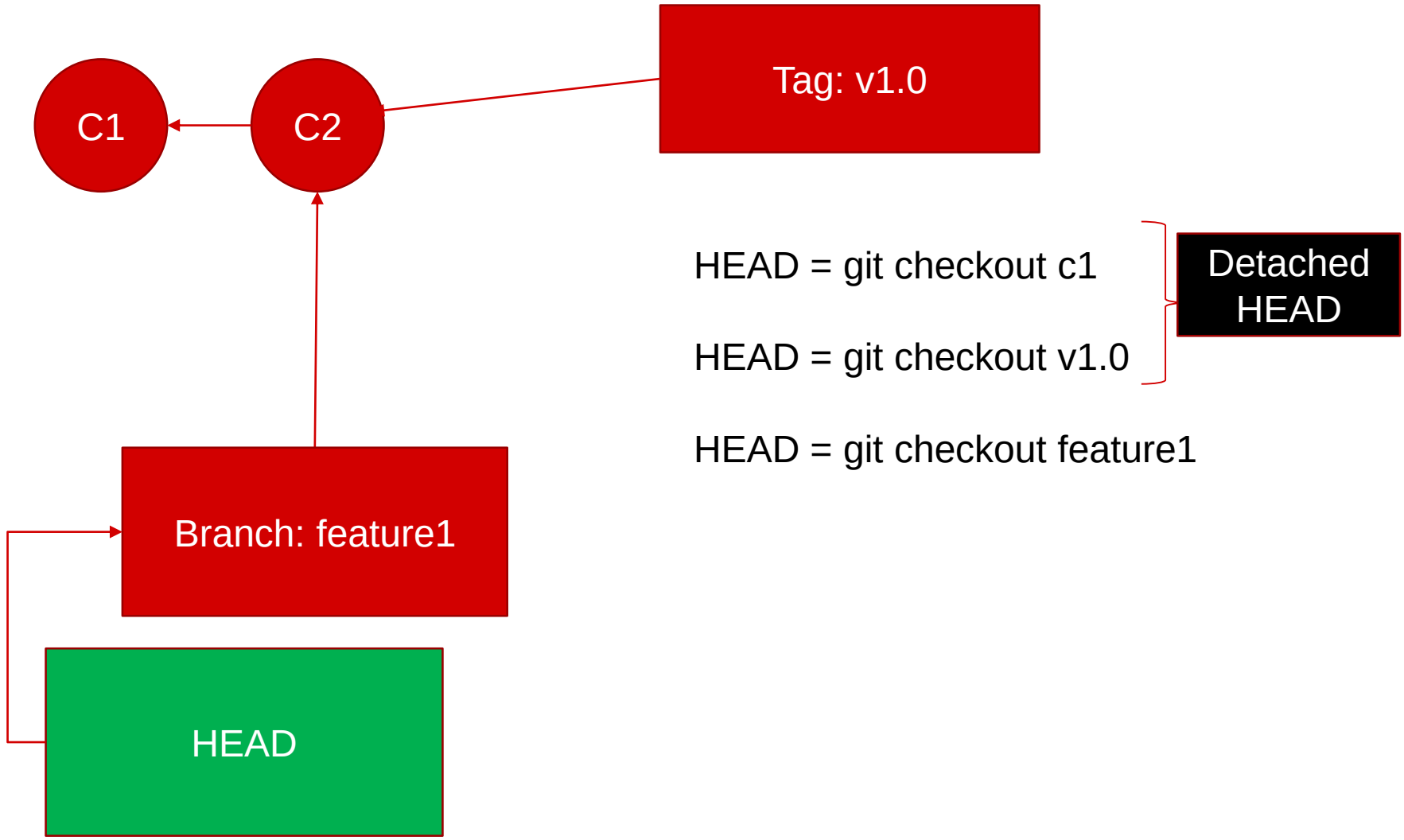
integrata cegos



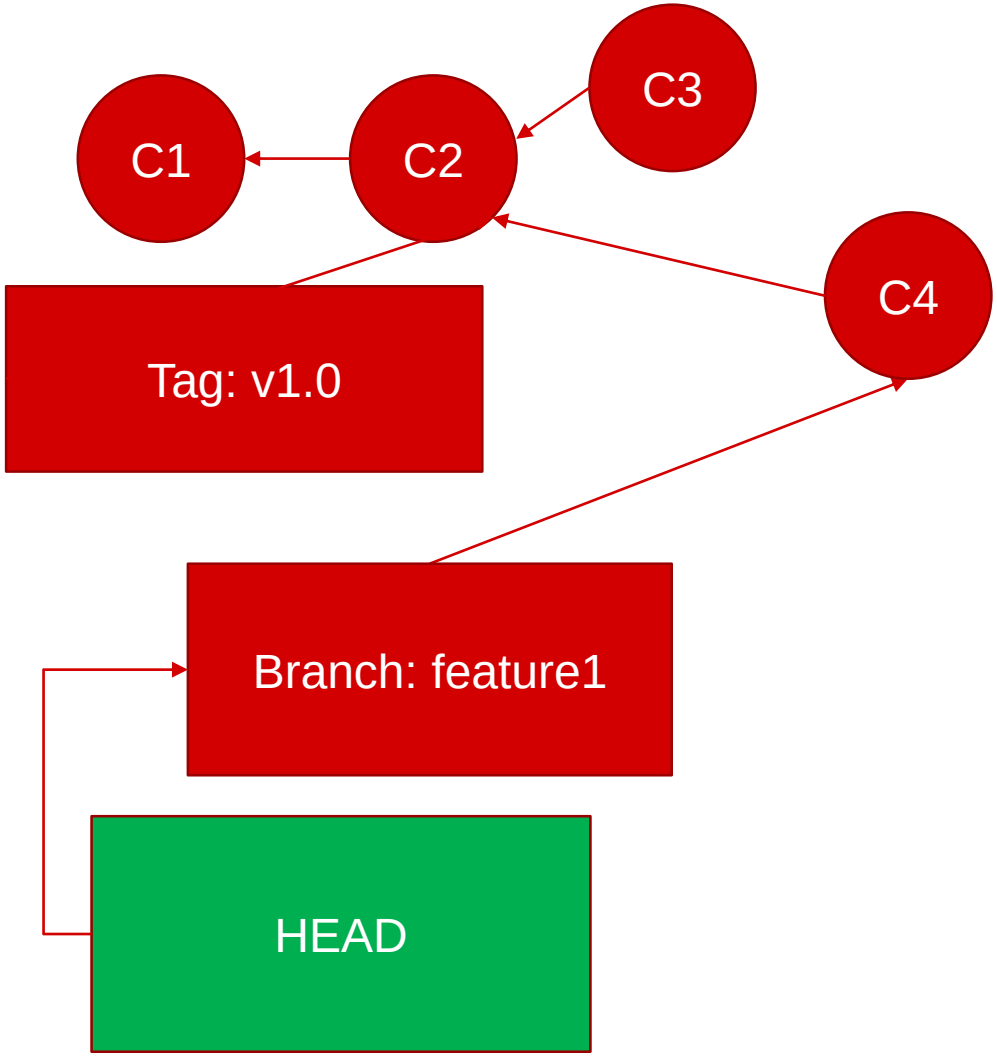
2 Arten von Alias

- Unverrückbare, feste Projektstände
  - Substantive
  - Versionsnamen, v.1.0
  - Tag
    - Git checkout c2
    - Git tag v1.0
    - Git tag –list
    - Git checkout v1.0
- Aktuell laufende Aktionen
  - Verb
  - ToDos, Jira-Ticket
  - Branch
    - Git checkout c1
    - Git branch feature1
    - Git checkout feature1
    - Git branch --list

Commits und checkout



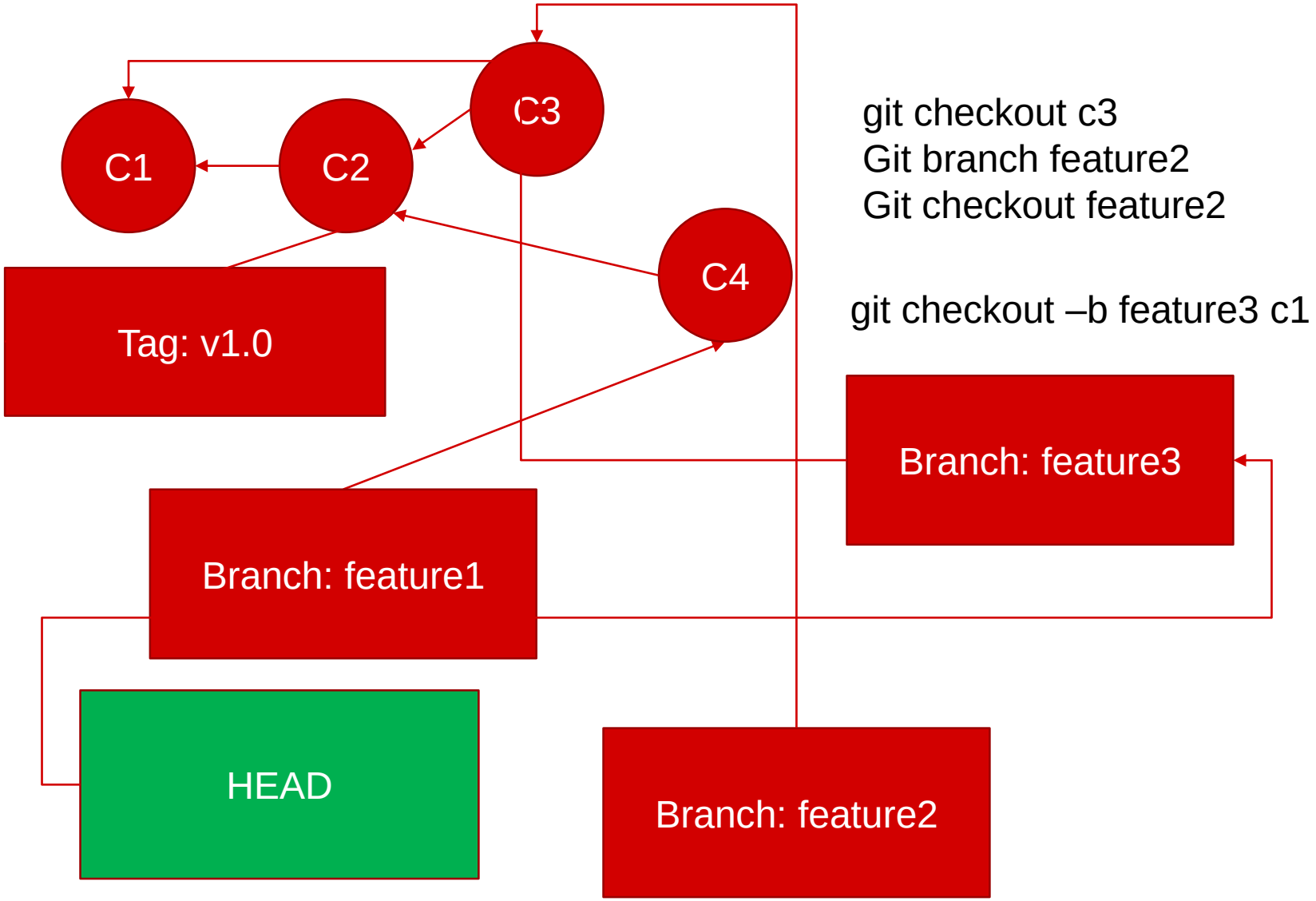
Commits und commit



git checkout v1.0
//Changes
Add + commit

git checkout feature1
//Changes
Add + commit

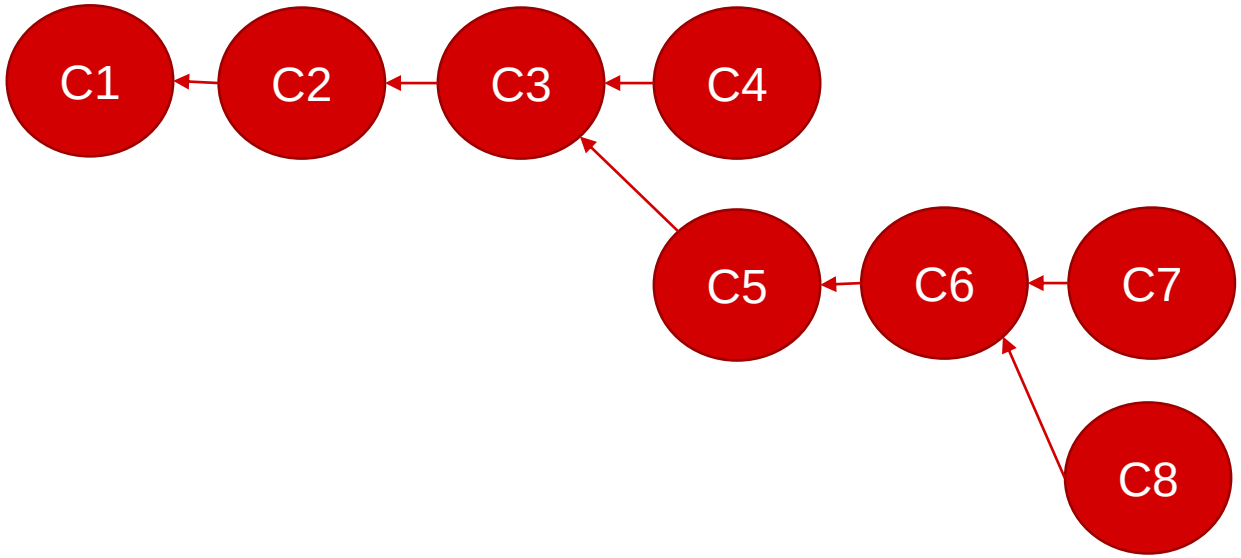
Commits und commit



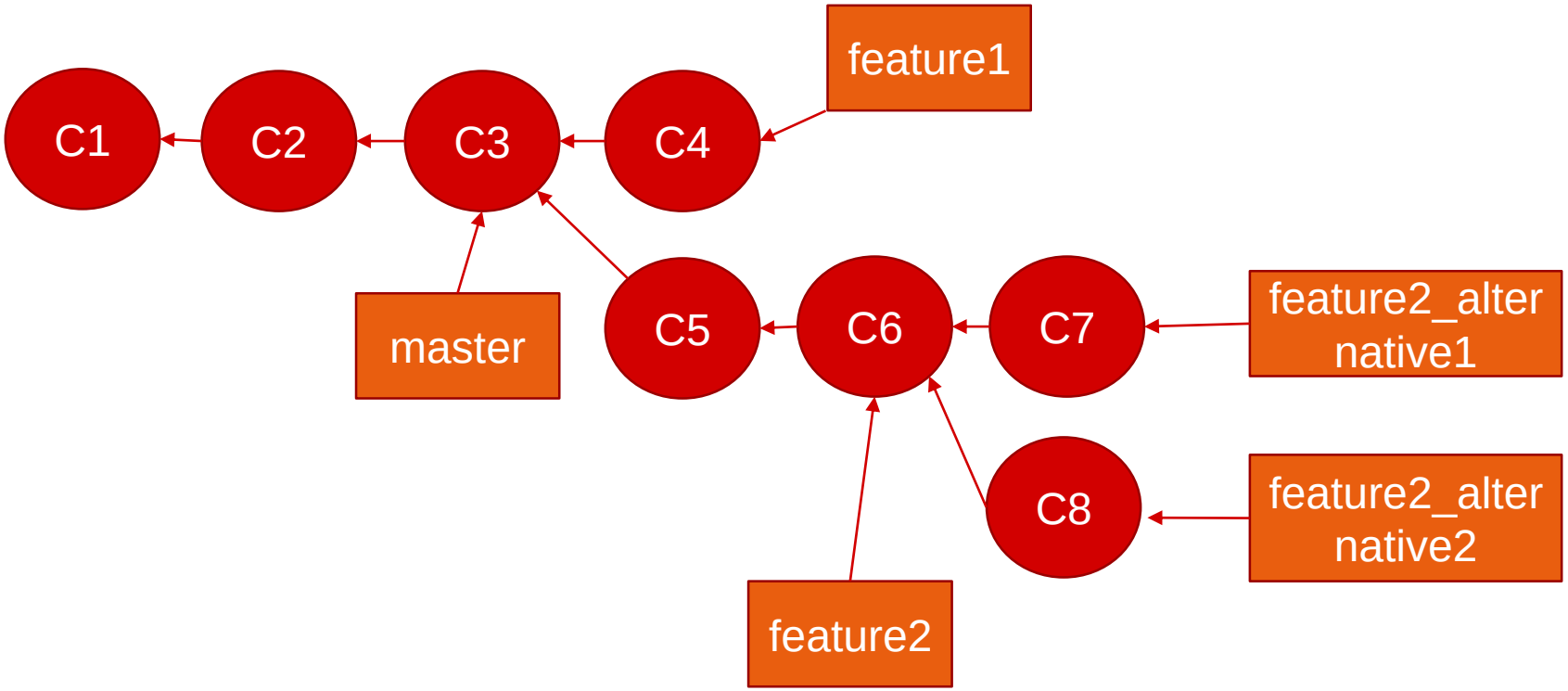
git checkout c3
Git branch feature2
Git checkout feature2

git checkout –b feature3 c1

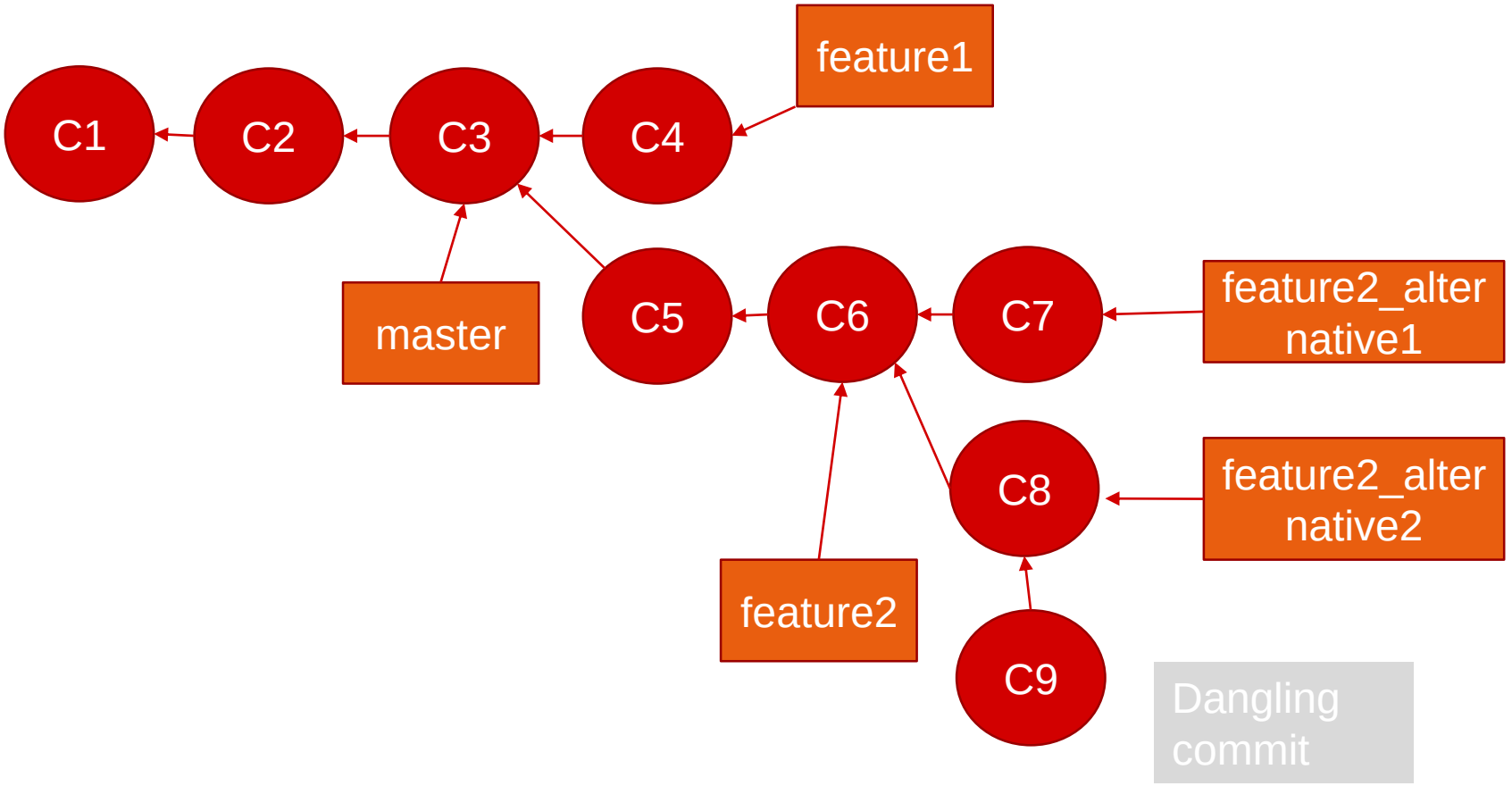
Template



Create-branches.bat



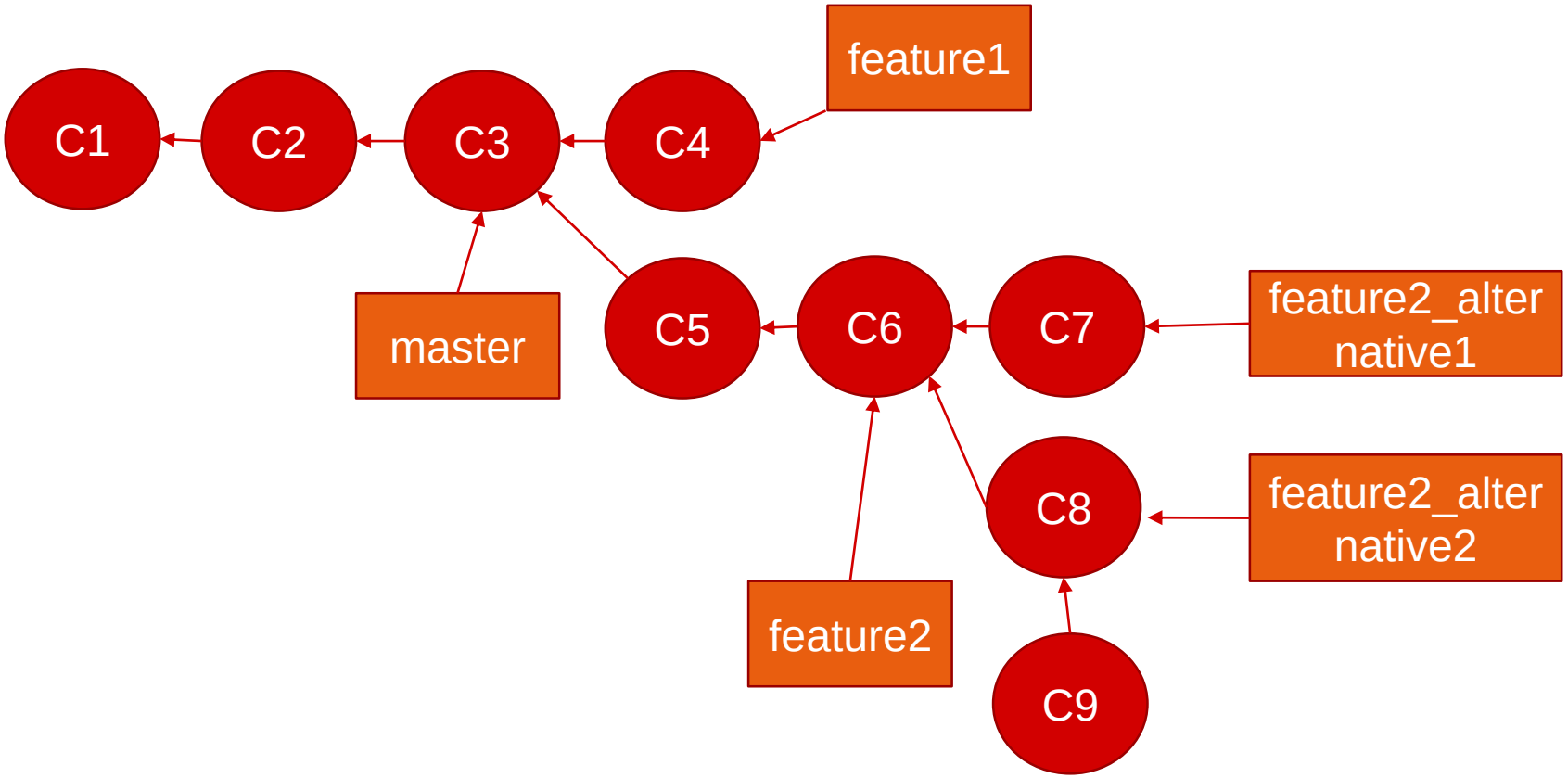
Dangling


feature1
C1
C2
C3
C4
master
C5
C6
C7
feature2_alter native1
C8
feature2_alter native2
feature2
C9
Dangling commit

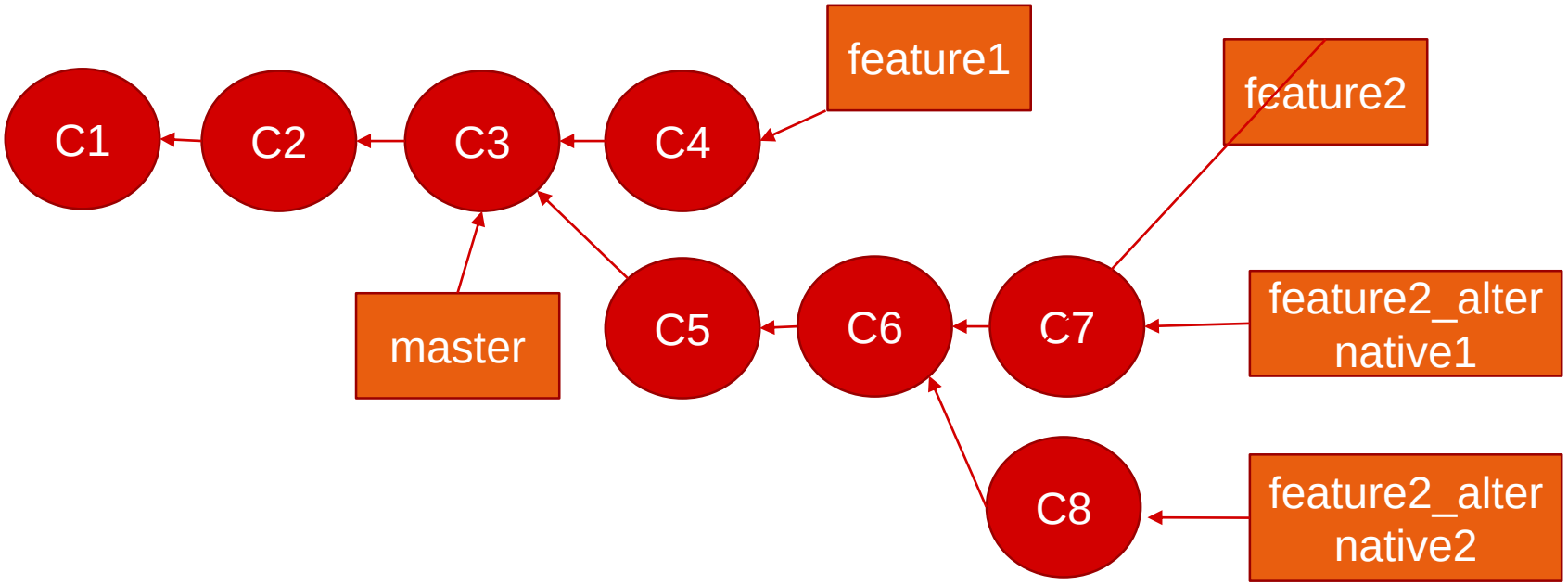
Zusammenführen von Projektständen

- Merge
- Rebase
- Cherry Picking
  - Aktuell in der Community eher als Anti Pattern diskutiert
- Interactive Rebasing

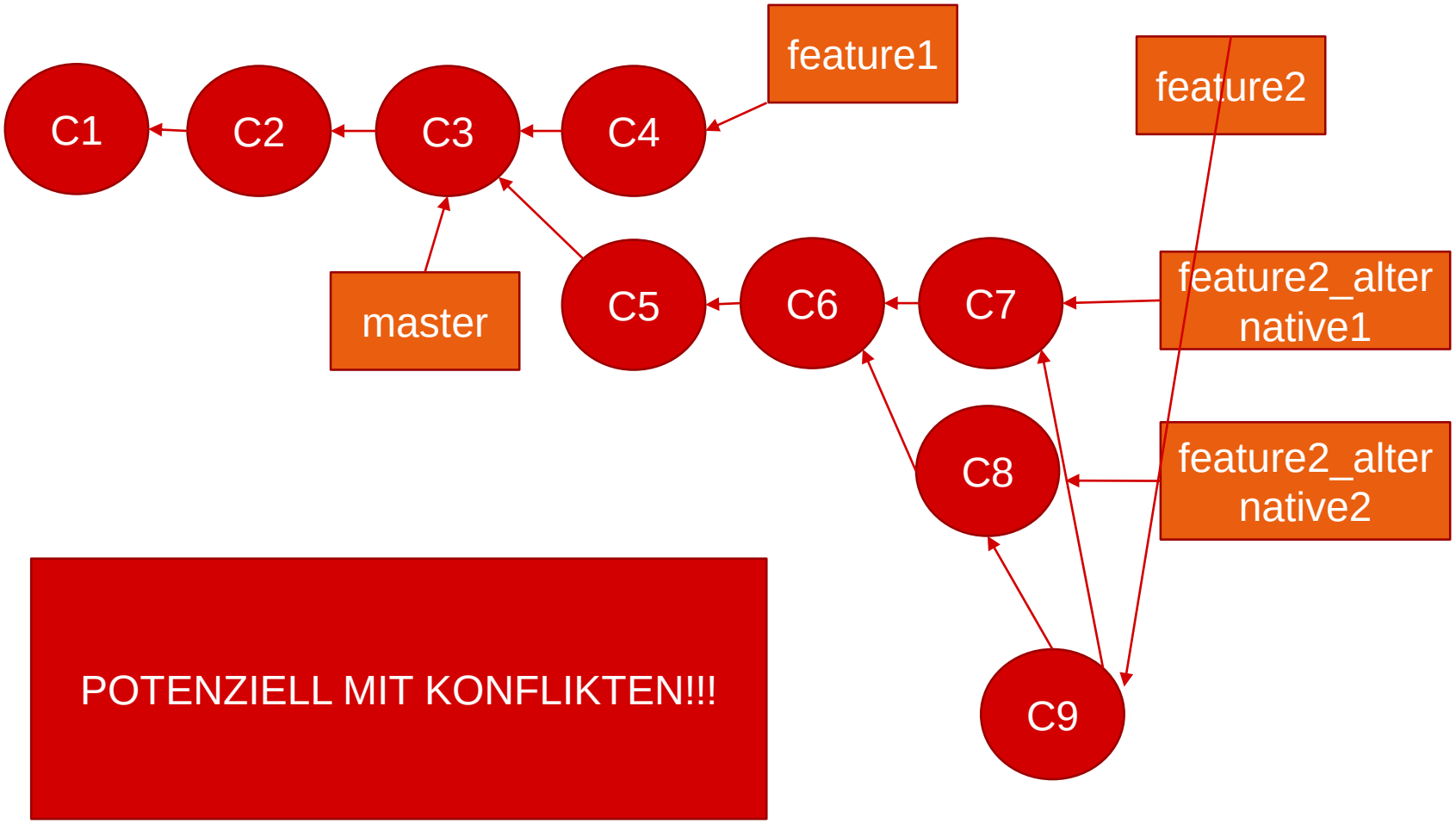
Merge


feature1
C1
C2
C3
C4
master
C5
C6
C7
feature2_alter
native1
C8
feature2_alter
native2
feature2
C9

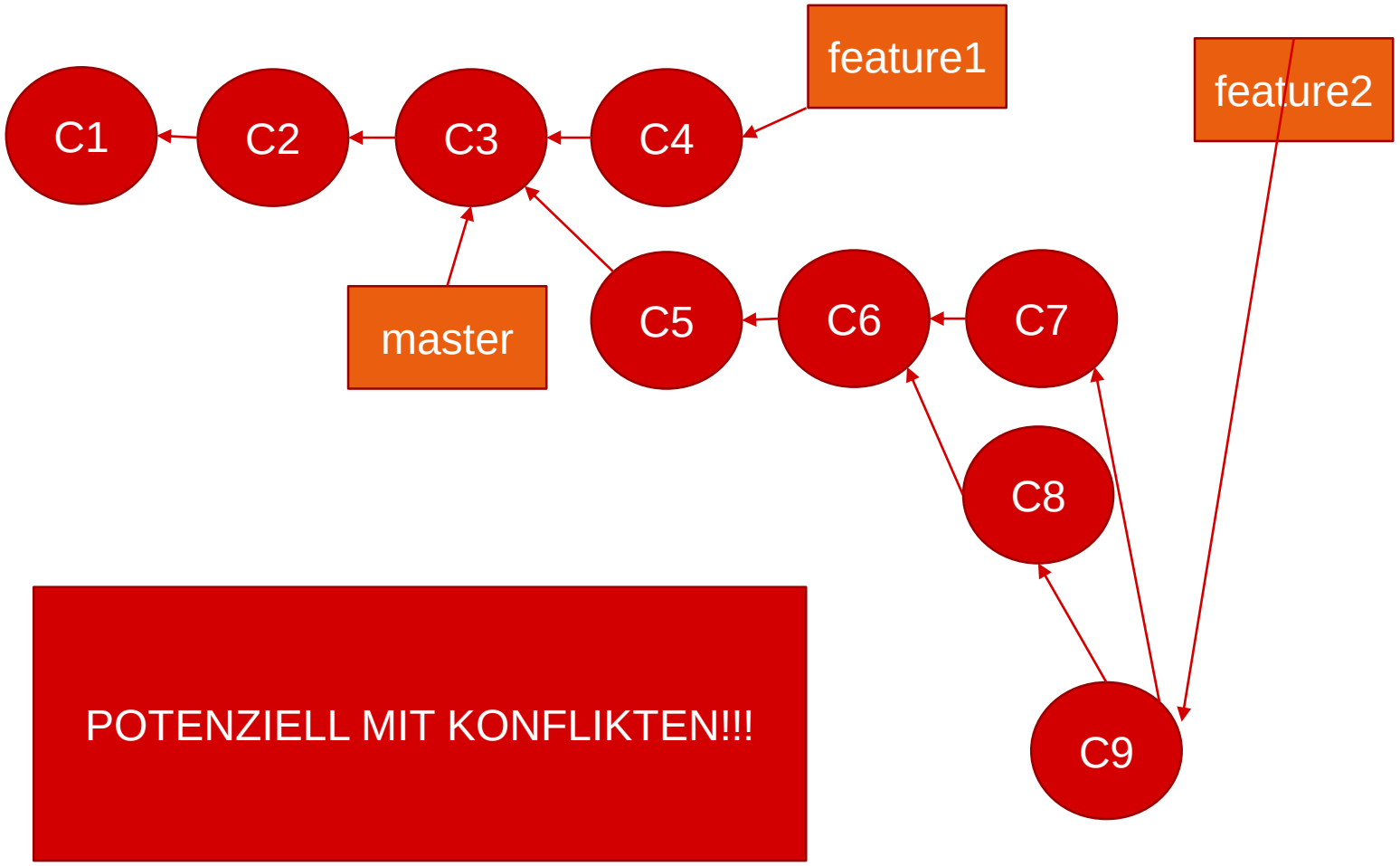
Merge feature 2 mit alt1: fast Forward



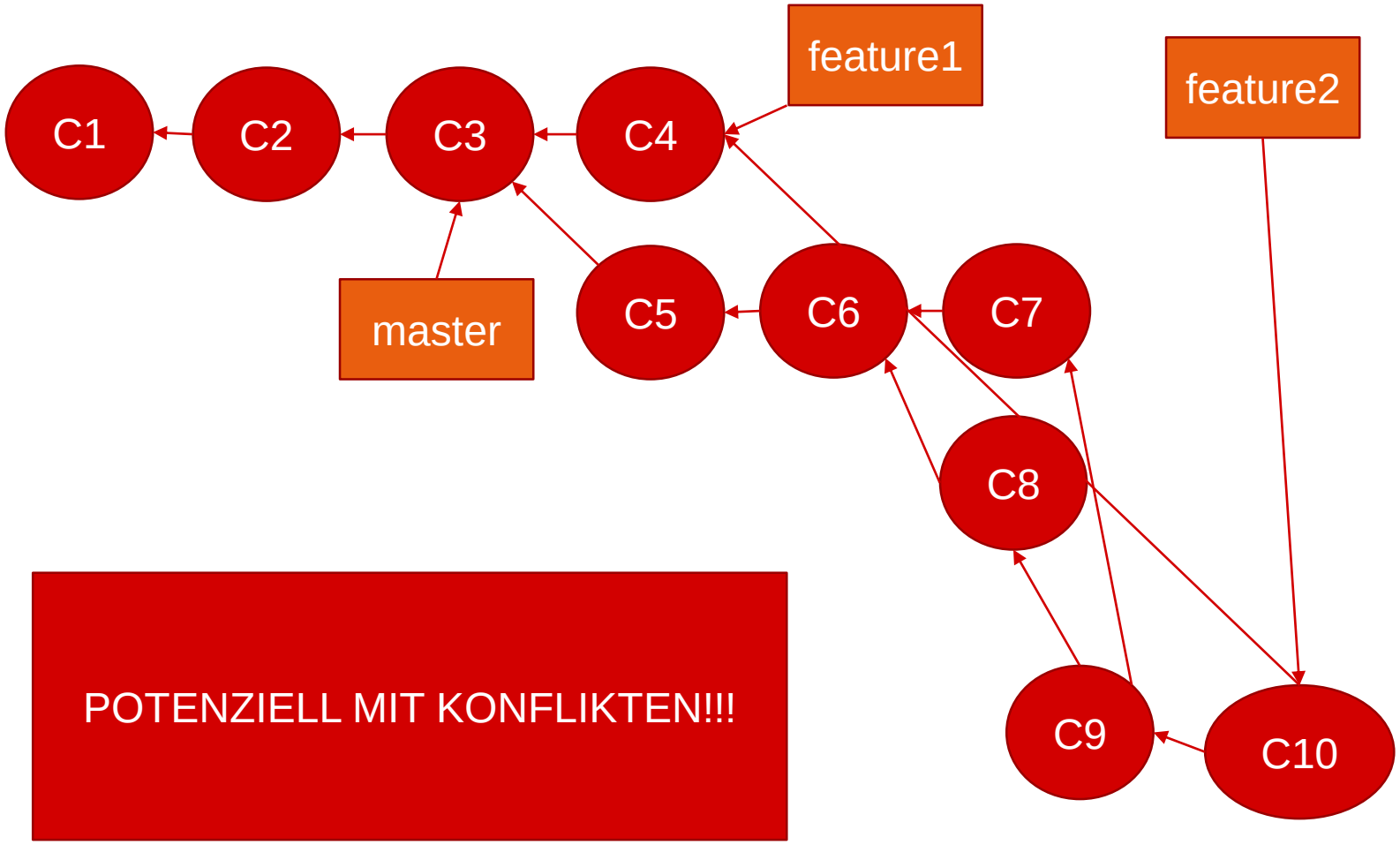
Merge feature 2 mit alt2: Recursive Merge



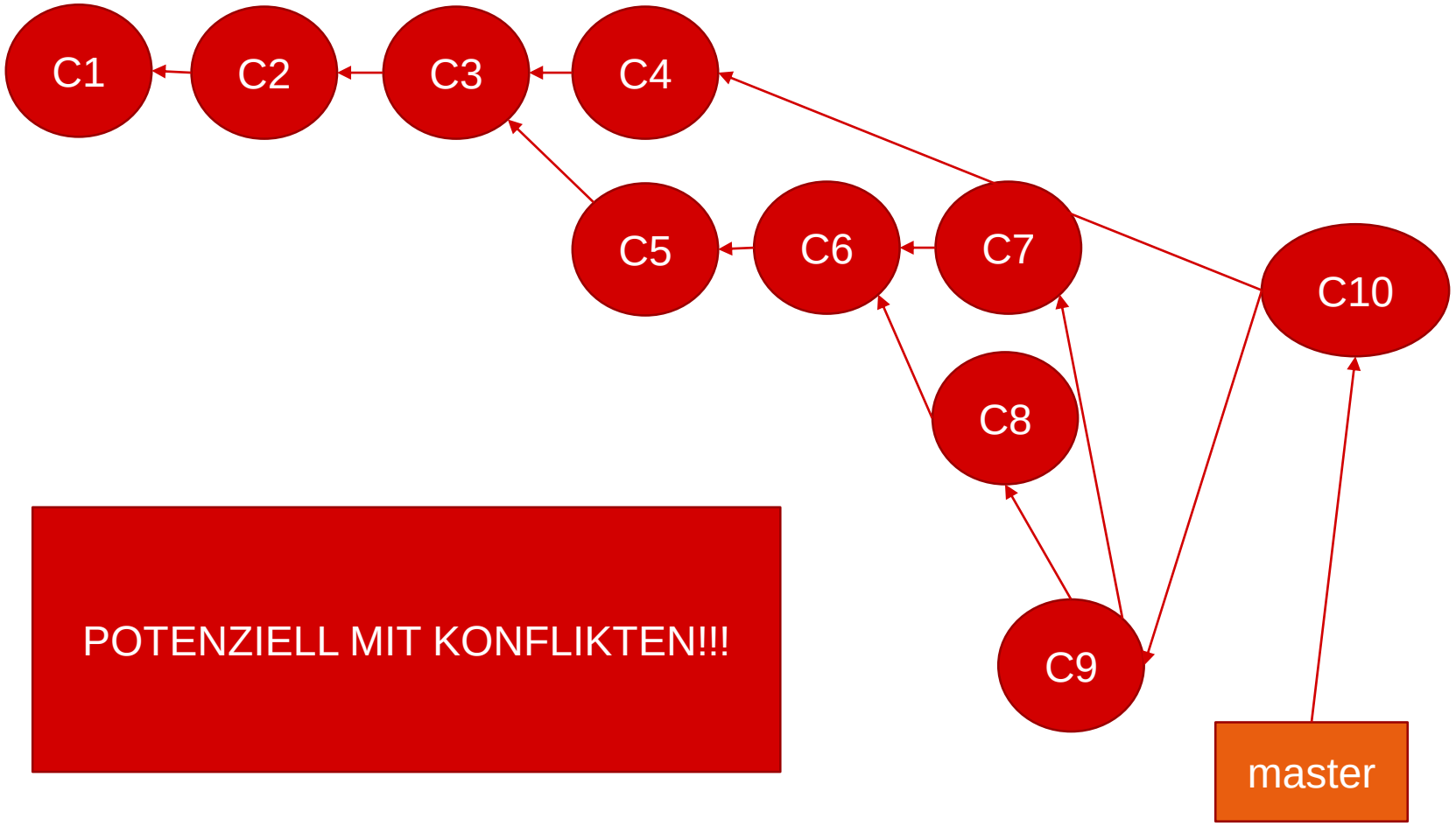
Merge feature 2 mit alt2:



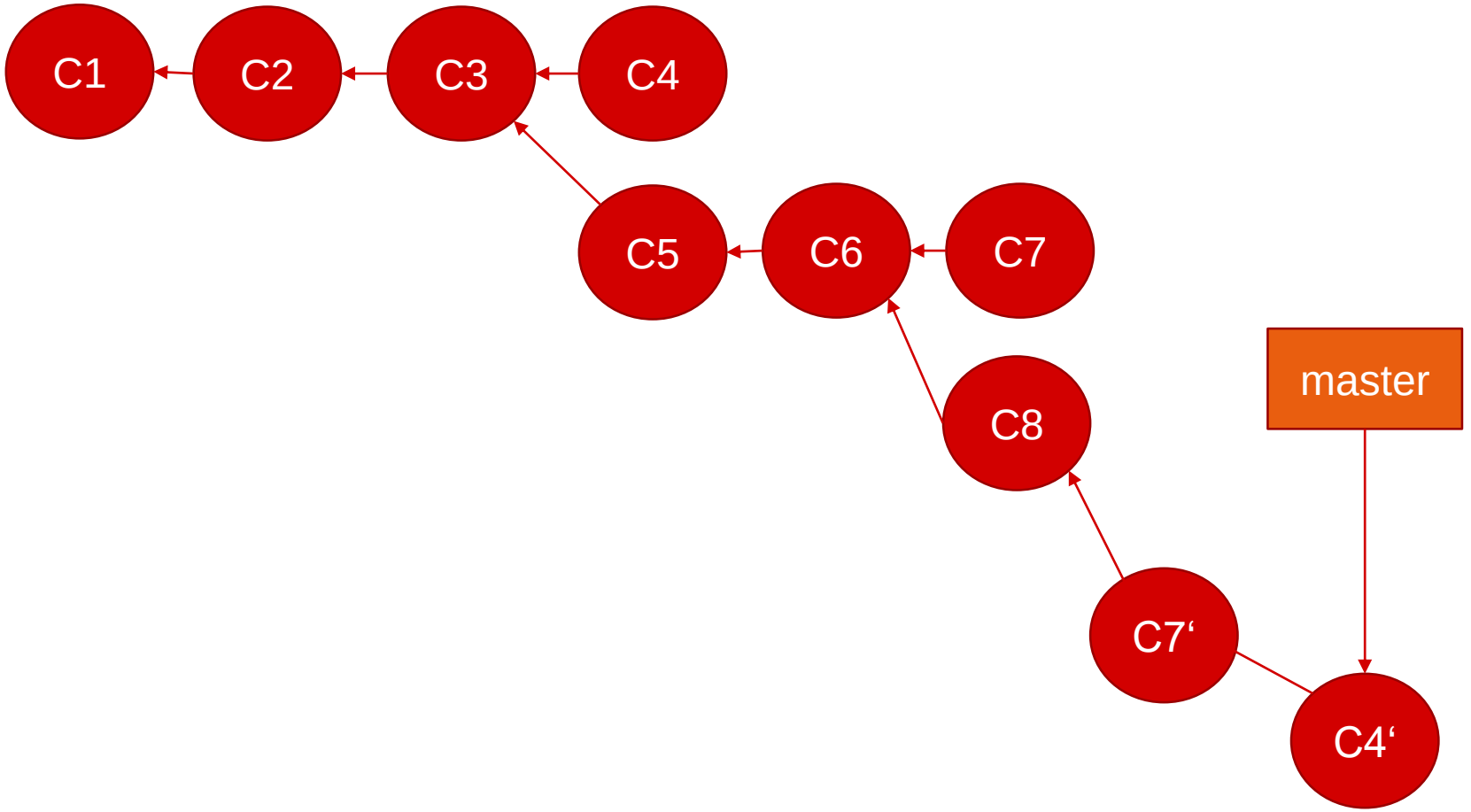
Merge feature2 mit feature1:



Merge master mit feature2:

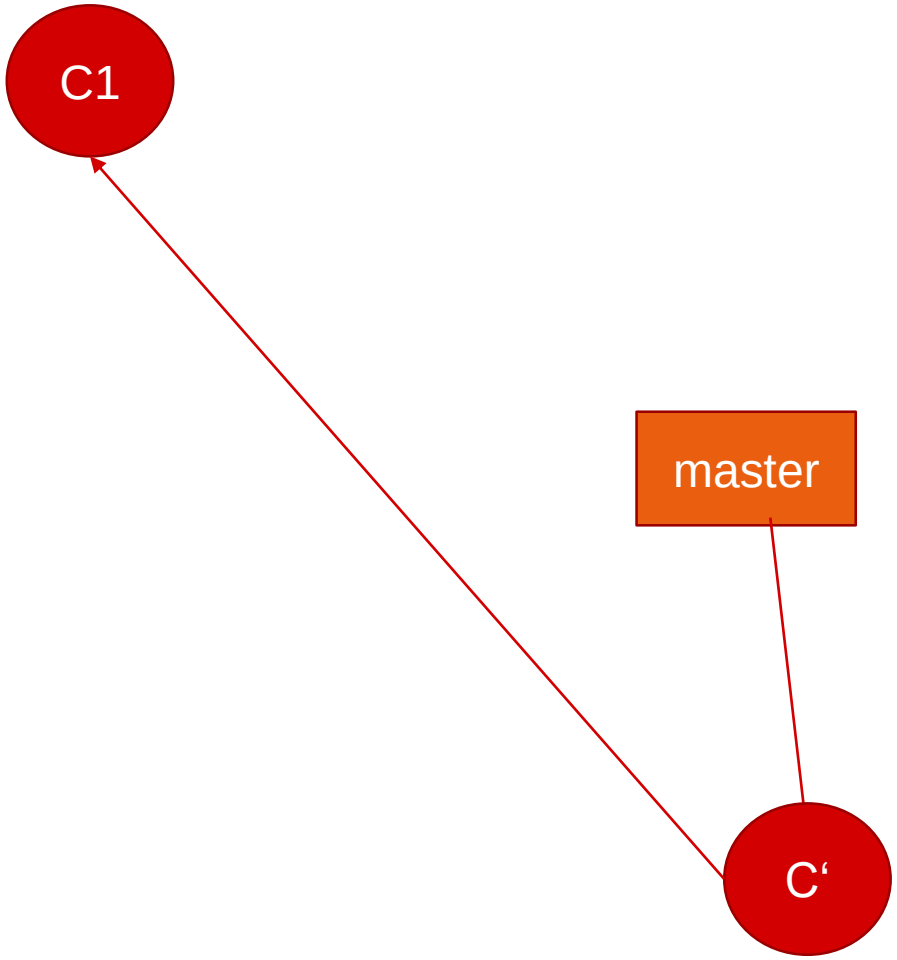


Rebasing

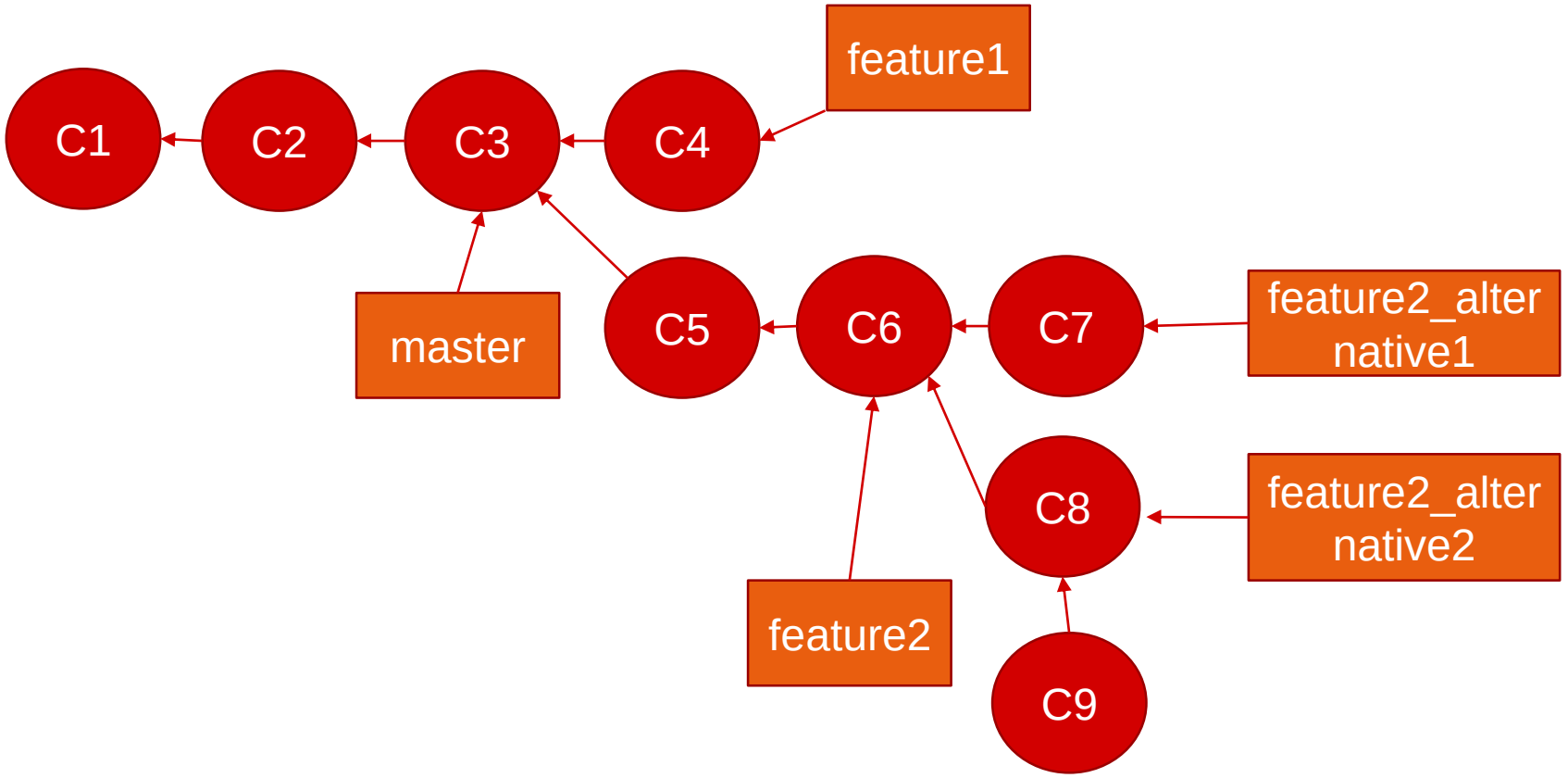


Interactive Rebasing

integrata cegos



Template


feature1
C1
C2
C3
C4
master
C5
C6
C7
feature2_alter
native1
C8
feature2_alter
native2
feature2
C9

Übung Branches

integrata cegos

- Create-branches.bat|sh
- git log --all --decorate --oneline –graph
- Garbage Collection
  - git fsck --unreachable --no-reflogs
  - git reflog expire --expire-unreachable=now --all
  - rem initialize garbage collection
  - git gc --prune=now
- Ideensammlung:
  - FRAGEN STELLEN!
  - Detached HEAD
  - Neue Branches erstellen
  - Löschen feature1-> Dangling commits