

Microservice Architecture

Code & Context, WS 2024 / 25

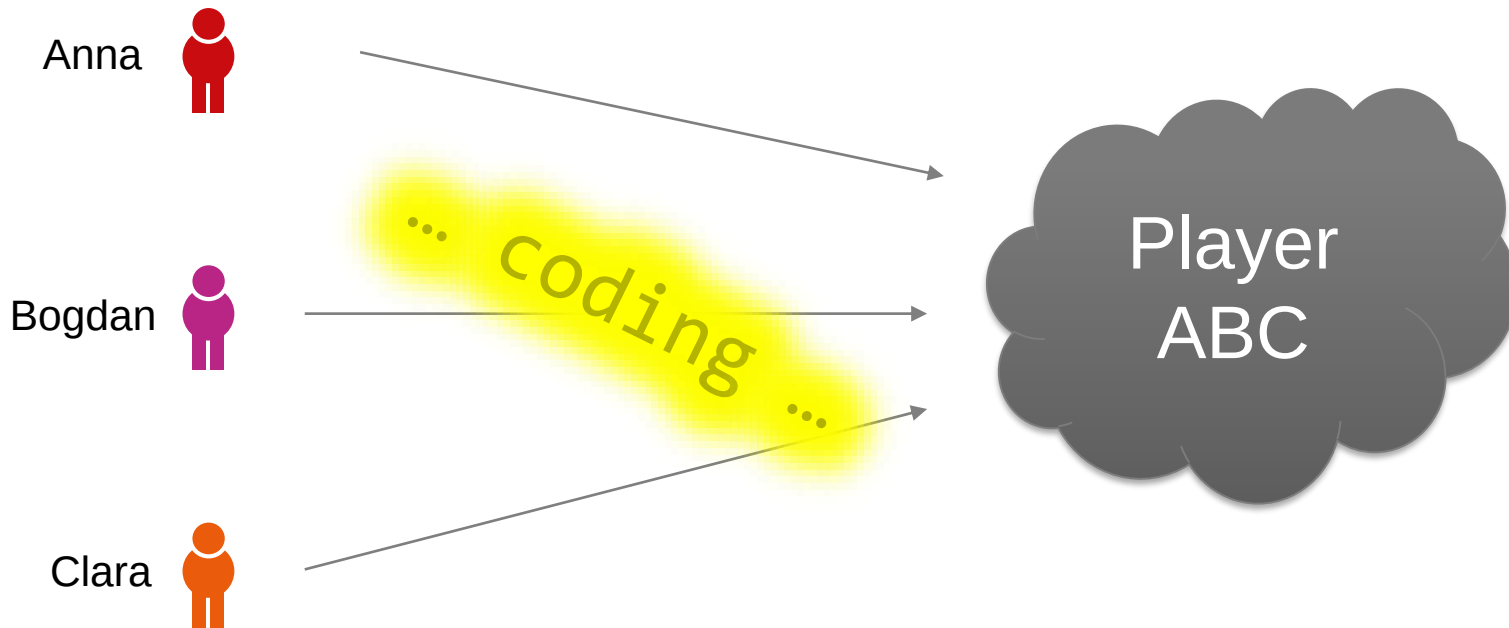
Prof. Dr.-Ing. Stefan Bente

Entwickeln im Team

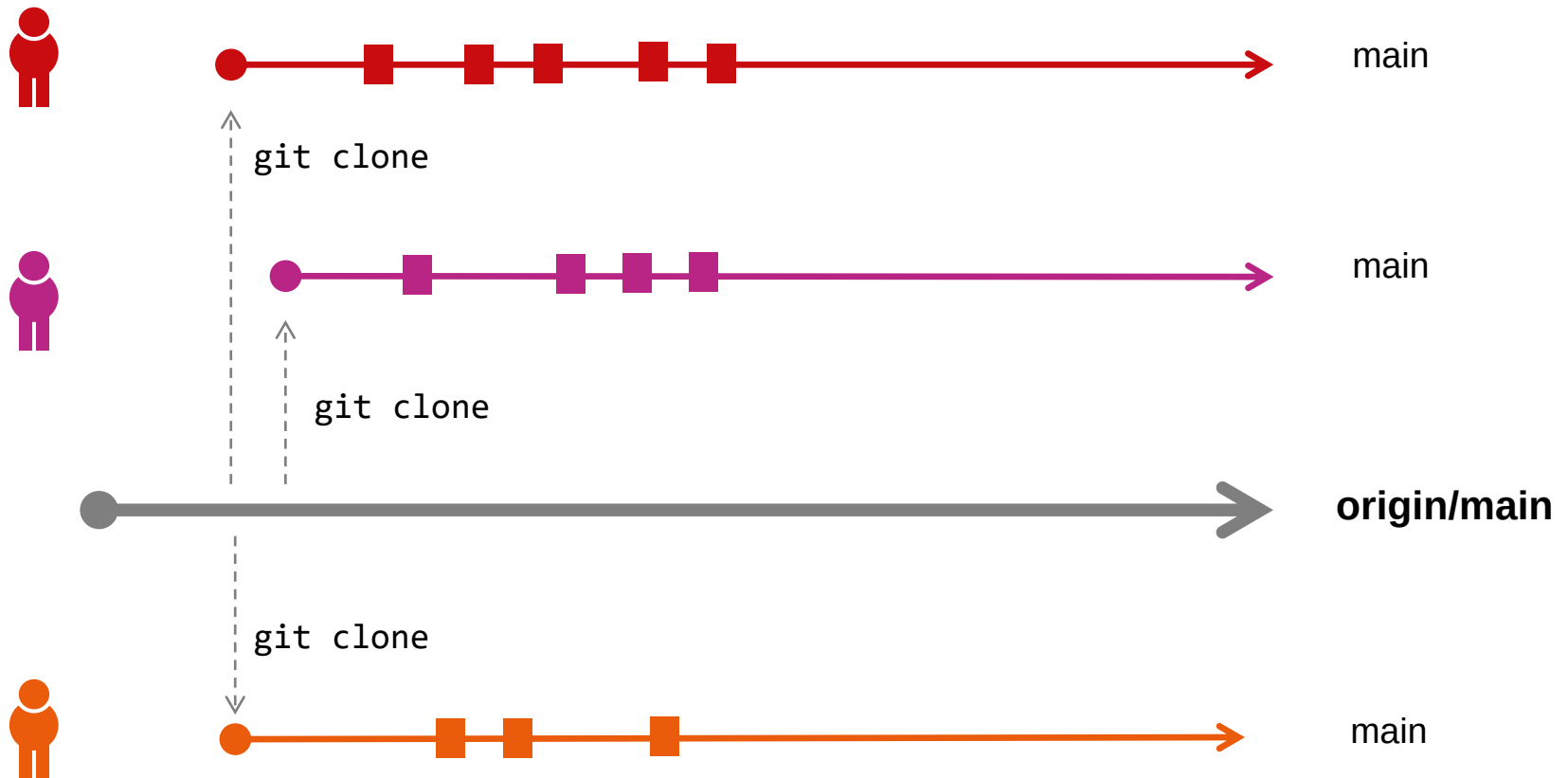
18.10.2024

Technology
Arts Sciences
TH Köln

Arbeiten im Team

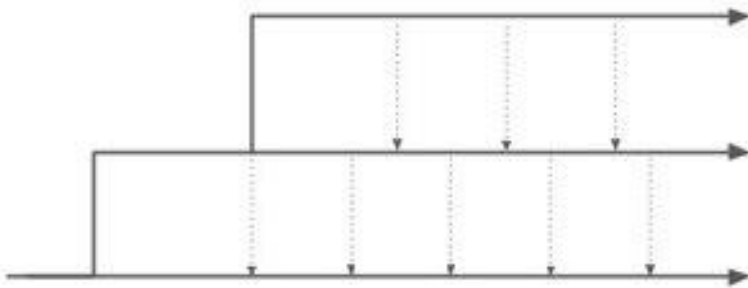


1) Gemeinsame Entwicklung auf dem Main Branch ...

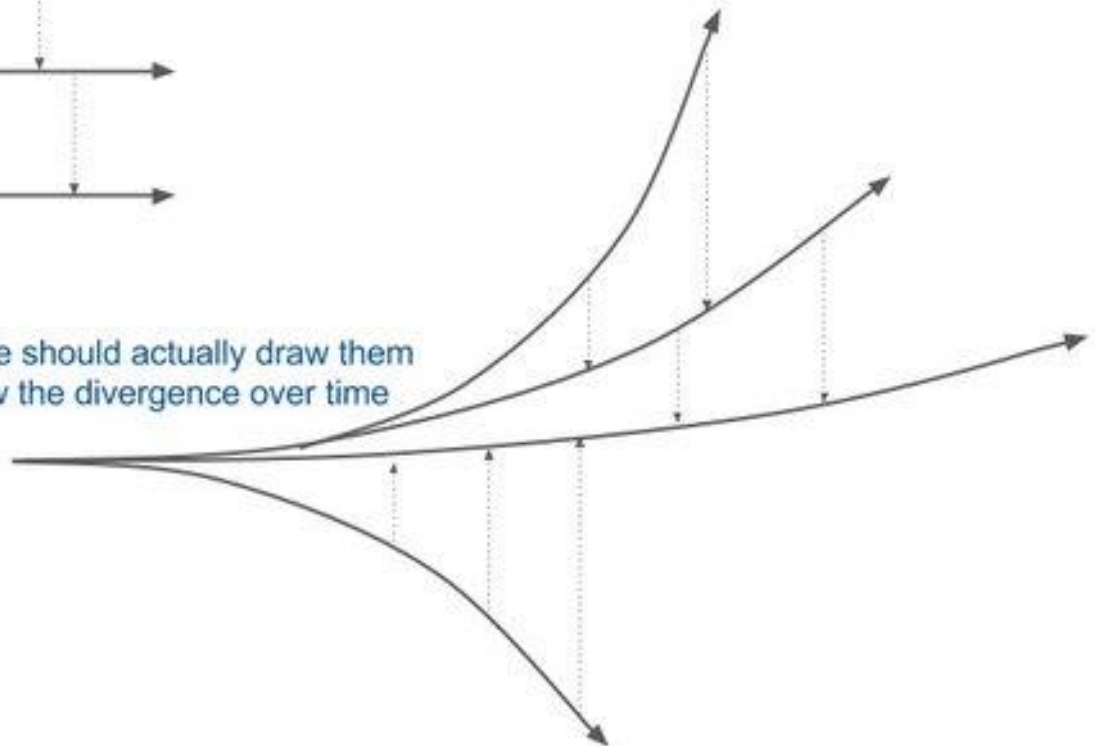


Der Effekt von „ungeregelter Zusammenarbeit“

How most people draw branching diagrams



How we should actually draw them to show the divergence over time

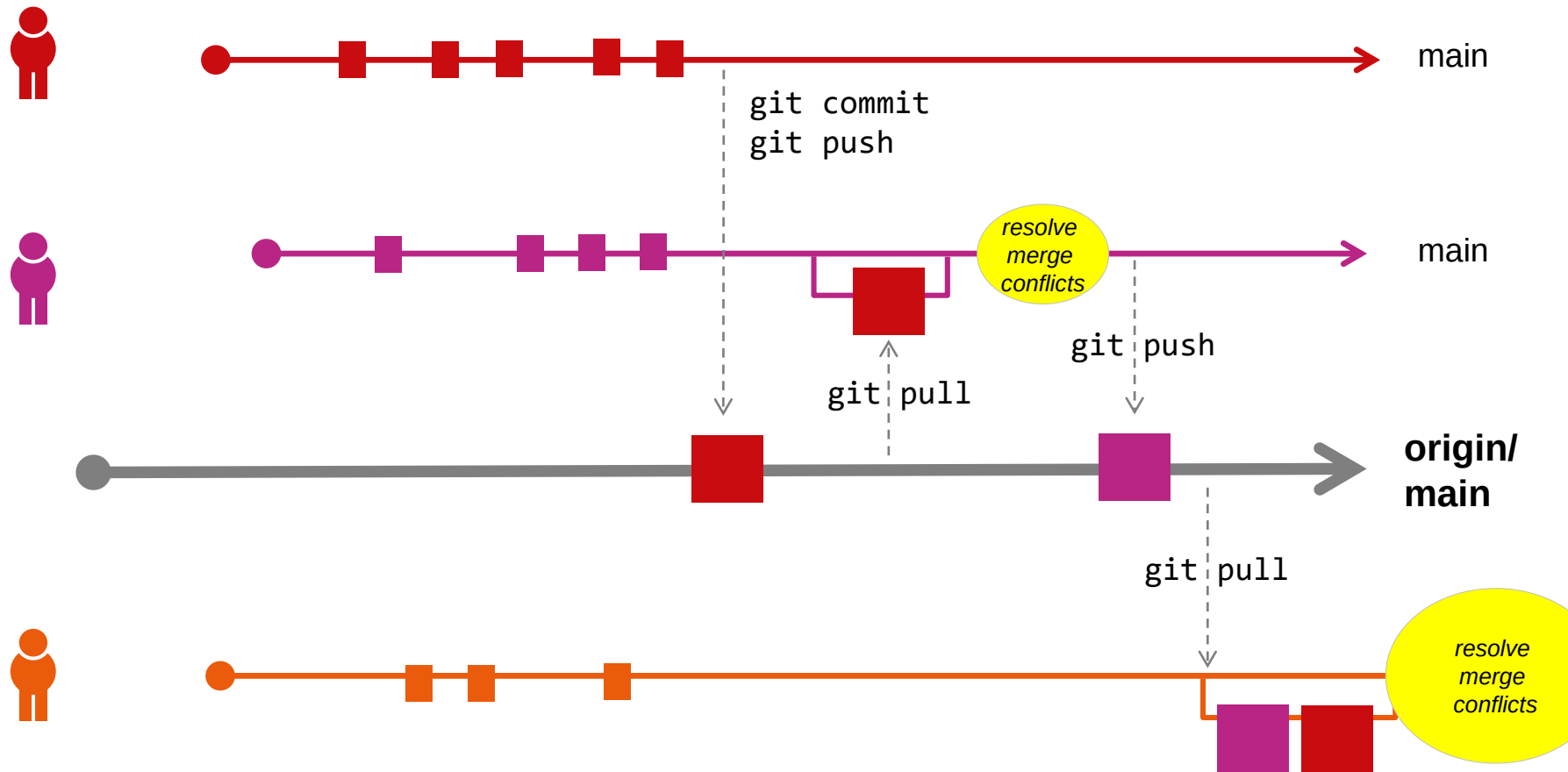


Quelle:

Martin Fowler, Patterns for Managing Source Code Branches

<https://martinfowler.com/articles/branching-patterns.html>

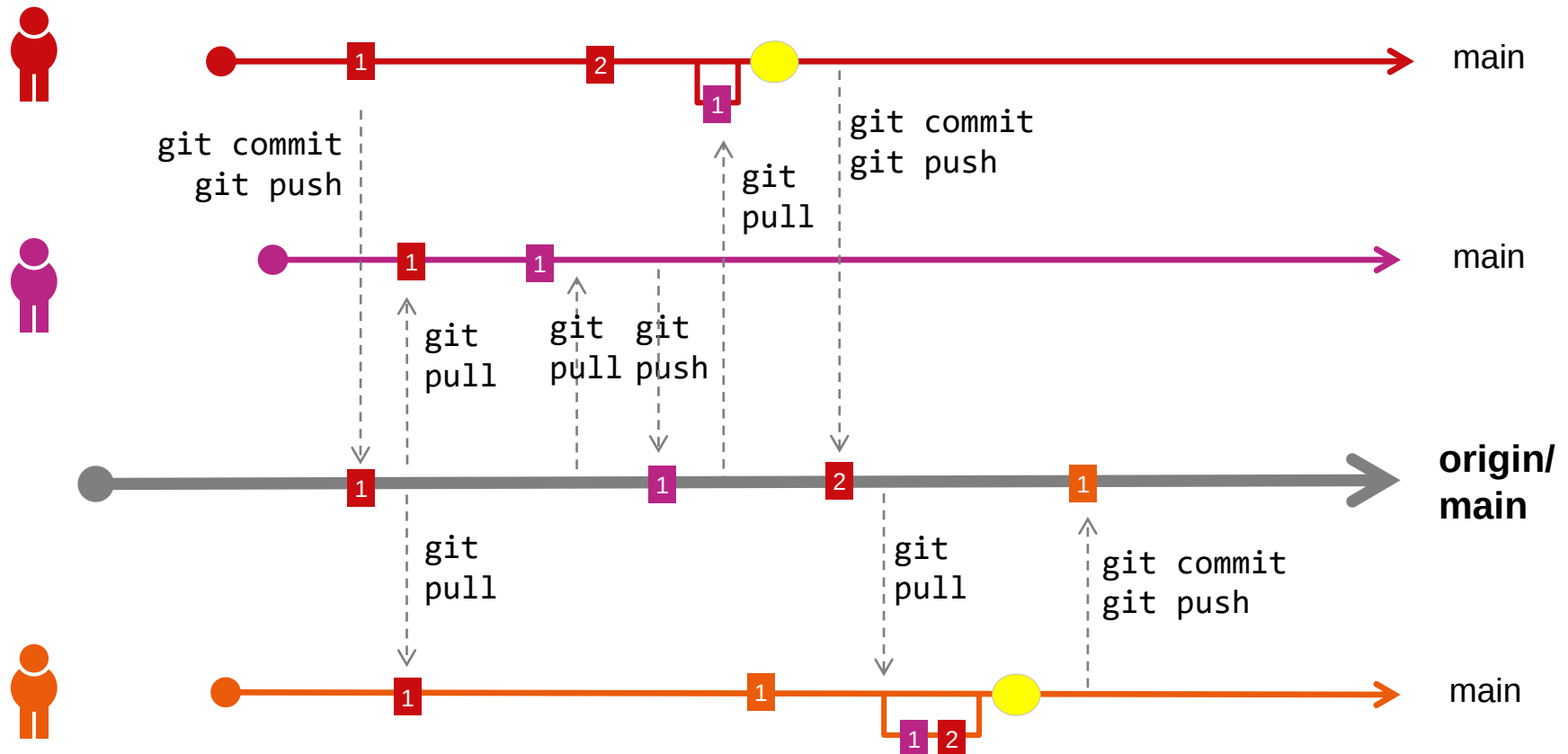
1) Gemeinsame Entwicklung auf dem Main Branch ...



(Darstellungsweise angelehnt an:

Martin Fowler, Patterns for Managing Source Code Branches, <https://martinfowler.com/articles/branching-patterns.html>

1) gemeinsamer Main Branch: **Wie vermeidet man Merge-Hölle?**

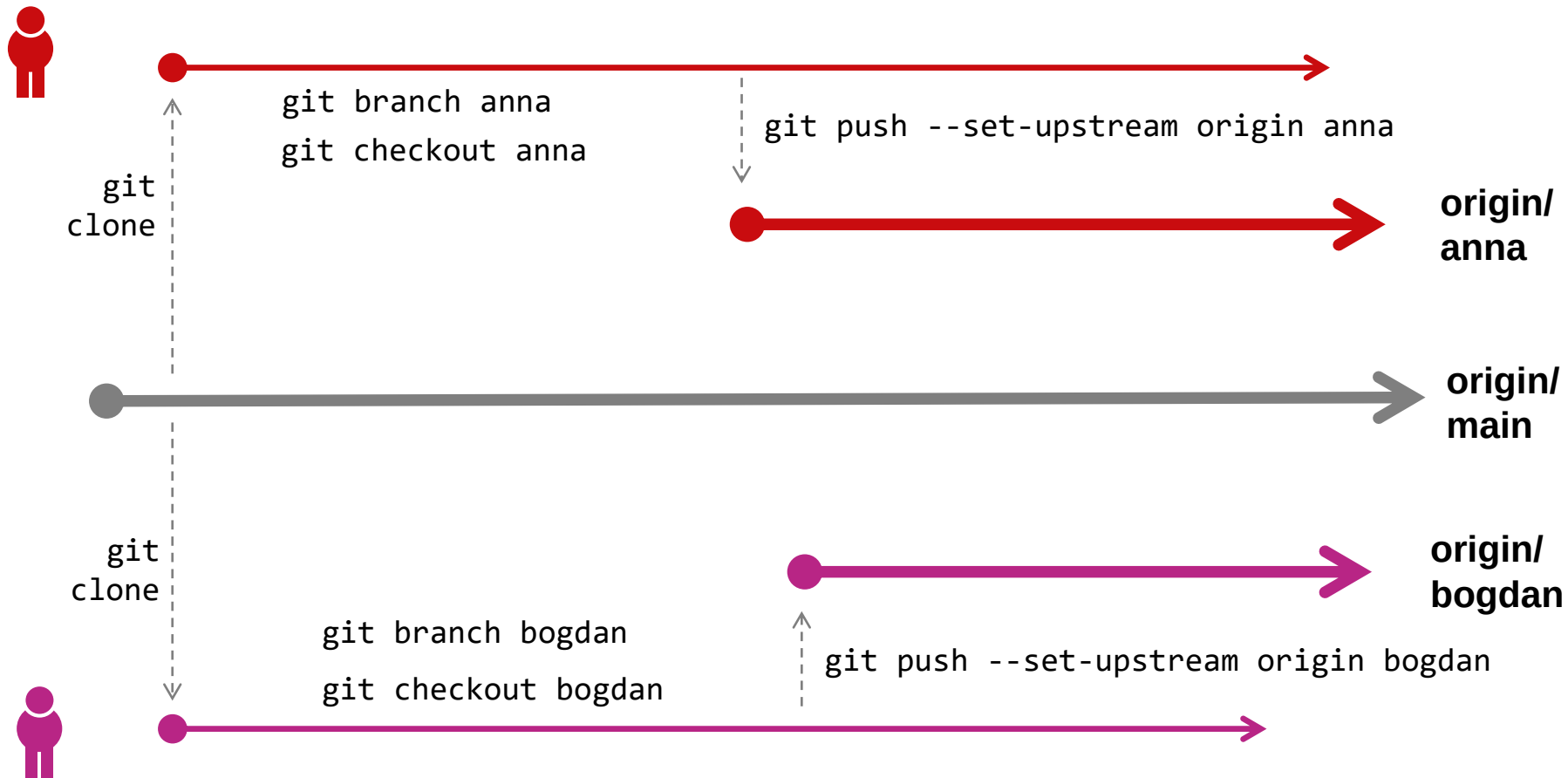


1) *gemeinsamer Main Branch*: **Wie vermeidet man Merge-Hölle?**

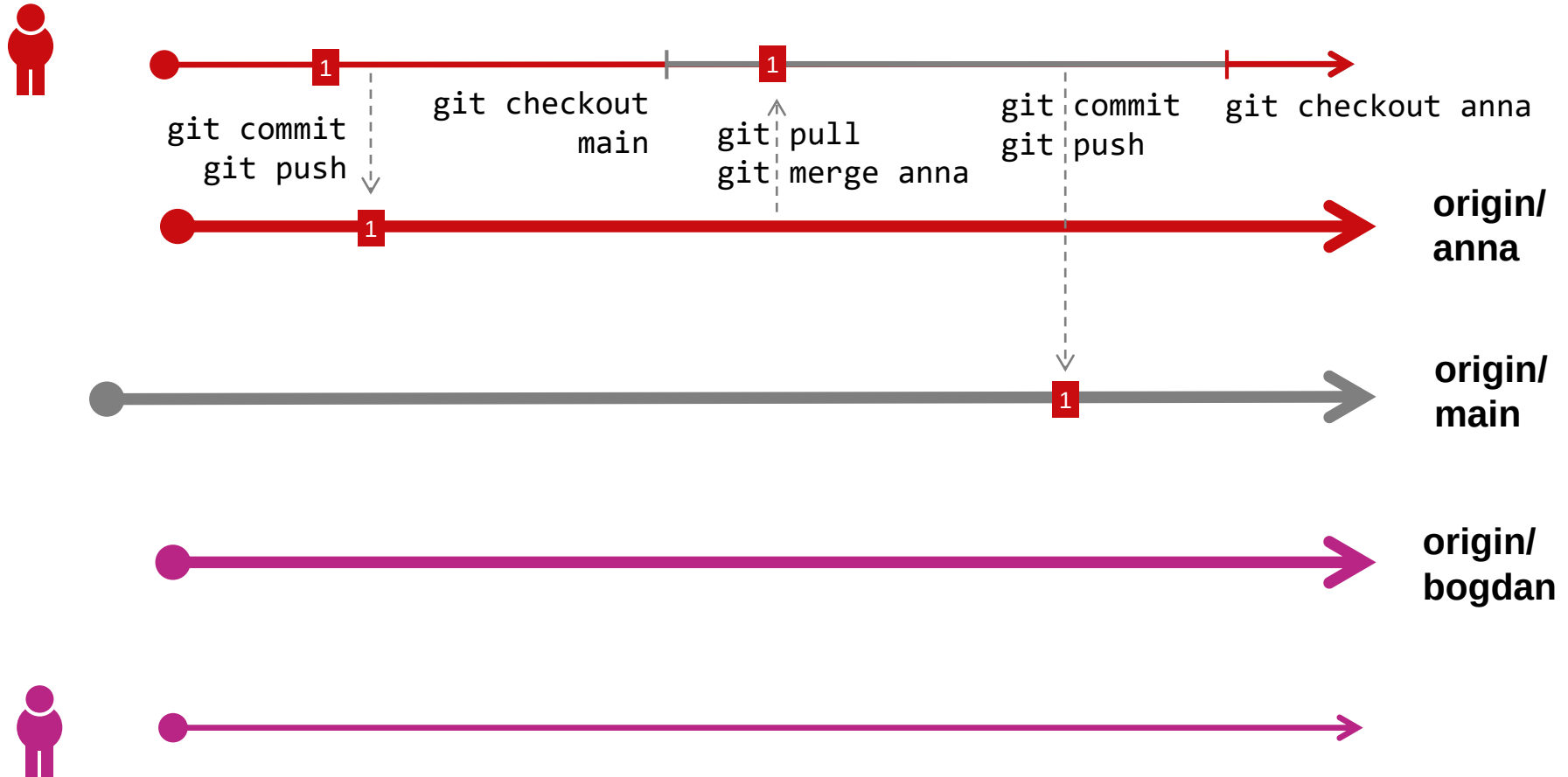
- Merke: Jeden Tag mindestens 1x ...
- `git commit`
- `git pull`
- `(resolve merge conflicts)`
- `git push`
- Ansonsten ist das Projekt schon nach 1-2 Tagen völlig zerschossen 😞
- Bei drei Team-Mitgliedern: Ggfs. besser in Branches arbeiten.
 - Gemeinsamer Main Branch ist immer dann geeignet, wenn man an verschiedenen Dateien arbeitet
 - Dann funktioniert der „automatische Merge“

2) Arbeiten mit Feature Branches

(der Einfachheit halber nur mit 2 Teammitgliedern gezeigt, funktioniert aber natürlich auch mit >2)



2) Arbeiten mit Feature Branches (1)



2) Arbeiten mit Feature Branches (2)

