

# EINFÜHRUNG IN GIT

Linux statt Windows - 03.03.2016  
von Niklas Heer



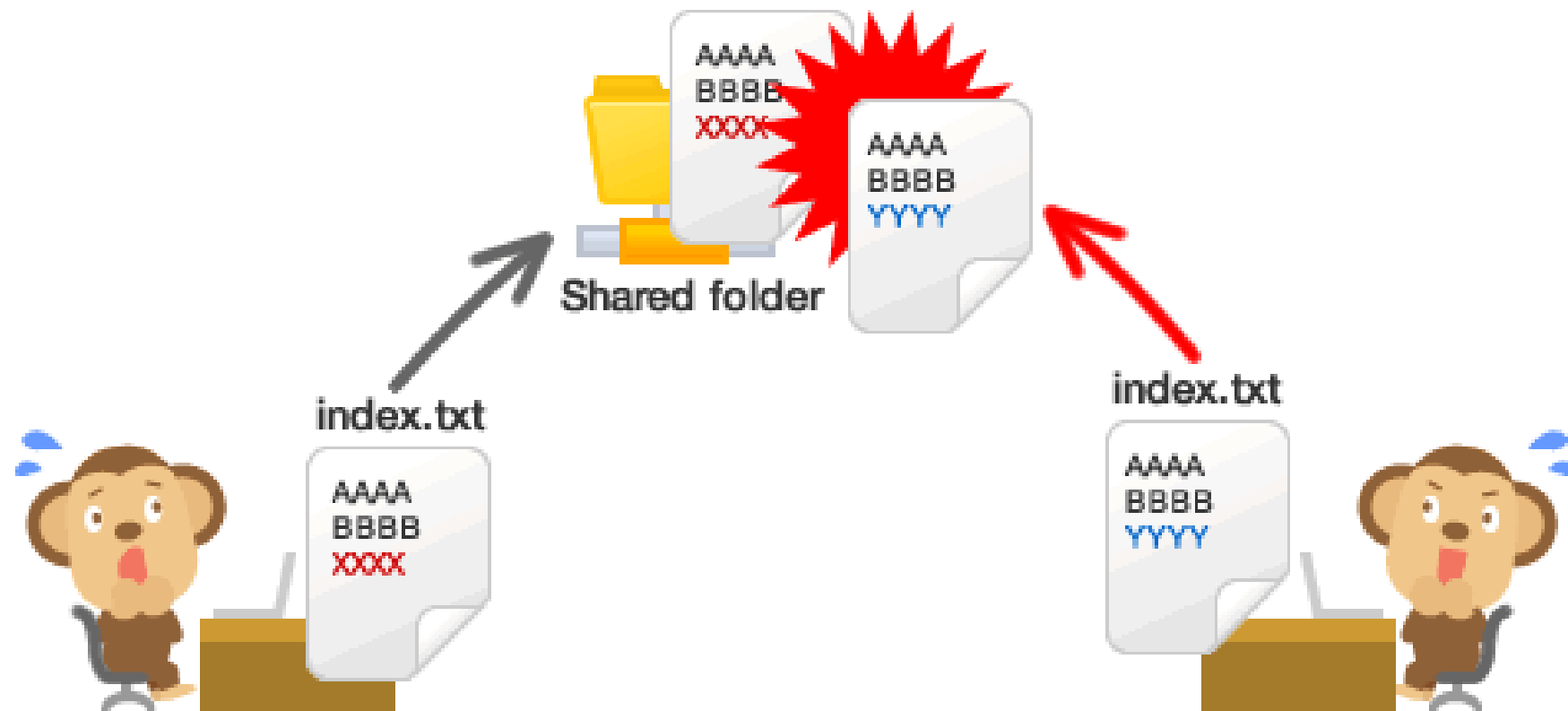
# GLIEDERUNG

1. Was ist Git?
2. Git benutzen
3. Git Tools
4. Github
5. Zusammenfassung
6. Fragen

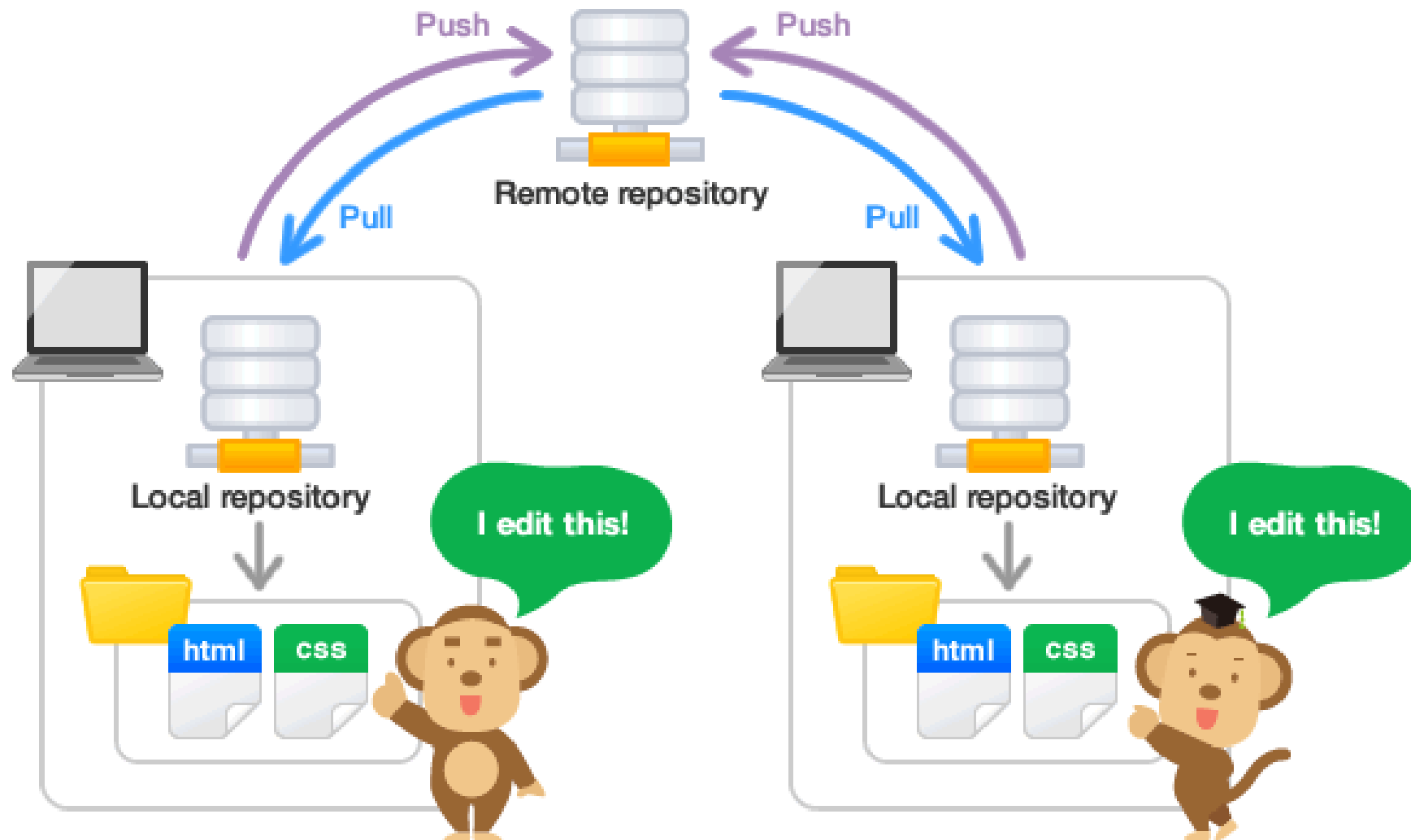
**WAS IST GIT?**

# **VERSIONSVERWALTUNG**

version control systems (VCS)



Quelle: [https://backlogtool.com/git-guide/en/intro/intro1\\_1.html](https://backlogtool.com/git-guide/en/intro/intro1_1.html)



Quelle: [https://backlogtool.com/git-guide/en/intro/intro1\\_2.html](https://backlogtool.com/git-guide/en/intro/intro1_2.html)

# DEFINITIONEN

Begriffe, die man kennen sollte

# **WORKSPACE**

Das Verzeichnis, das ihr für euer Git- Repository ausgewählt habt.



# REPOSITORY

Eine Sammlung an Commits & Branches, welche im .git Verzeichnis eures Workspaces liegen.

# **LOCAL REPOSITORY**

Dies sind eure lokalen Commits & Branches.  
(im Prinzip euer Workspace)

# REMOTE REPOSITORY

Dies ist ein Repository, welches nicht auf eurem Rechner liegt, sondern auf einem entfernten Rechner/Server.

(z.B. Github)

# BRANCH

Eine parallele Version eures ***Repository***s. Änderungen in einem Branch betreffen andere Branches nicht. Euer „Standard-Branch“ ist der **master** Branch.

# COMMIT

Eine Momentaufnahme eures *Workspaces* zu einem bestimmten Zeitpunkt. Dieser wird anhand einer *SHA-1 Prüfsumme* identifiziert.

# SHA-1 PRÜFSUMMEN

```
powerlevel9k : git — Konsole
commit e7e11d5ec87f30323d065100c1e8fdb4a98c3aba
Author: Niklas Heer <niklas.heer@gmail.com>
Date: Mon Jan 4 19:26:25 2016 +0100

    fixed documentation

commit a7bedd5320f0d6fba5d4179aace5ead1f42c459d
Author: Niklas Heer <niklas.heer@gmail.com>
Date: Mon Jan 4 19:00:02 2016 +0100

    added documentation

commit 3441f750bf114bc68e519d9ab351373b54e83d7c
Author: Niklas Heer <niklas.heer@gmail.com>
Date: Thu Dec 31 19:04:26 2015 +0100

    used set_default-function instead

commit f472ba7341615f17b279cc6859535e6e7668f637
Author: Niklas Heer <niklas.heer@gmail.com>
Date: Thu Dec 31 15:57:56 2015 +0100

    added a new variable POWERLEVEL9K_SHORTEN_DELIMITER to specifiy the delimiter

commit 2fdbd6158c78a0b1786d6a030629fe3a930fc46d
Author: Niklas Heer <niklas.heer@gmail.com>
Date: Wed Dec 30 15:49:23 2015 +0100

: _
```

```
powerlevel9k : git — Konsole
commit e7e11d5ec87f30323d065100c1e8fdb4a98c3aba
Author: Niklas Heer <niklas.heer@gmail.com>
Date: Mon Jan 4 19:26:25 2016 +0100

    fixed documentation

commit a7bedd5320f0d6fba5d4179aace5ead1f42c459d
Author: Niklas Heer <niklas.heer@gmail.com>
Date: Mon Jan 4 19:00:02 2016 +0100

    added documentation

commit 3441f750bf114bc68e519d9ab351373b54e83d7c
Author: Niklas Heer <niklas.heer@gmail.com>
Date: Thu Dec 31 19:04:26 2015 +0100

    used set_default-function instead

commit f472ba7341615f17b279cc6859535e6e7668f637
Author: Niklas Heer <niklas.heer@gmail.com>
Date: Thu Dec 31 15:57:56 2015 +0100

    added a new variable POWERLEVEL9K_SHORTEN_DELIMITER to specifiy the delimiter

commit 2fdbd6158c78a0b1786d6a030629fe3a930fc46d
Author: Niklas Heer <niklas.heer@gmail.com>
Date: Wed Dec 30 15:49:23 2015 +0100

: _
```



# HEAD

Name des Commits der momentan ausgechecked ist.

# **FORK**

Eine Kopie eines kompletten ***Repositorys*** mit allen Commits und Branches.

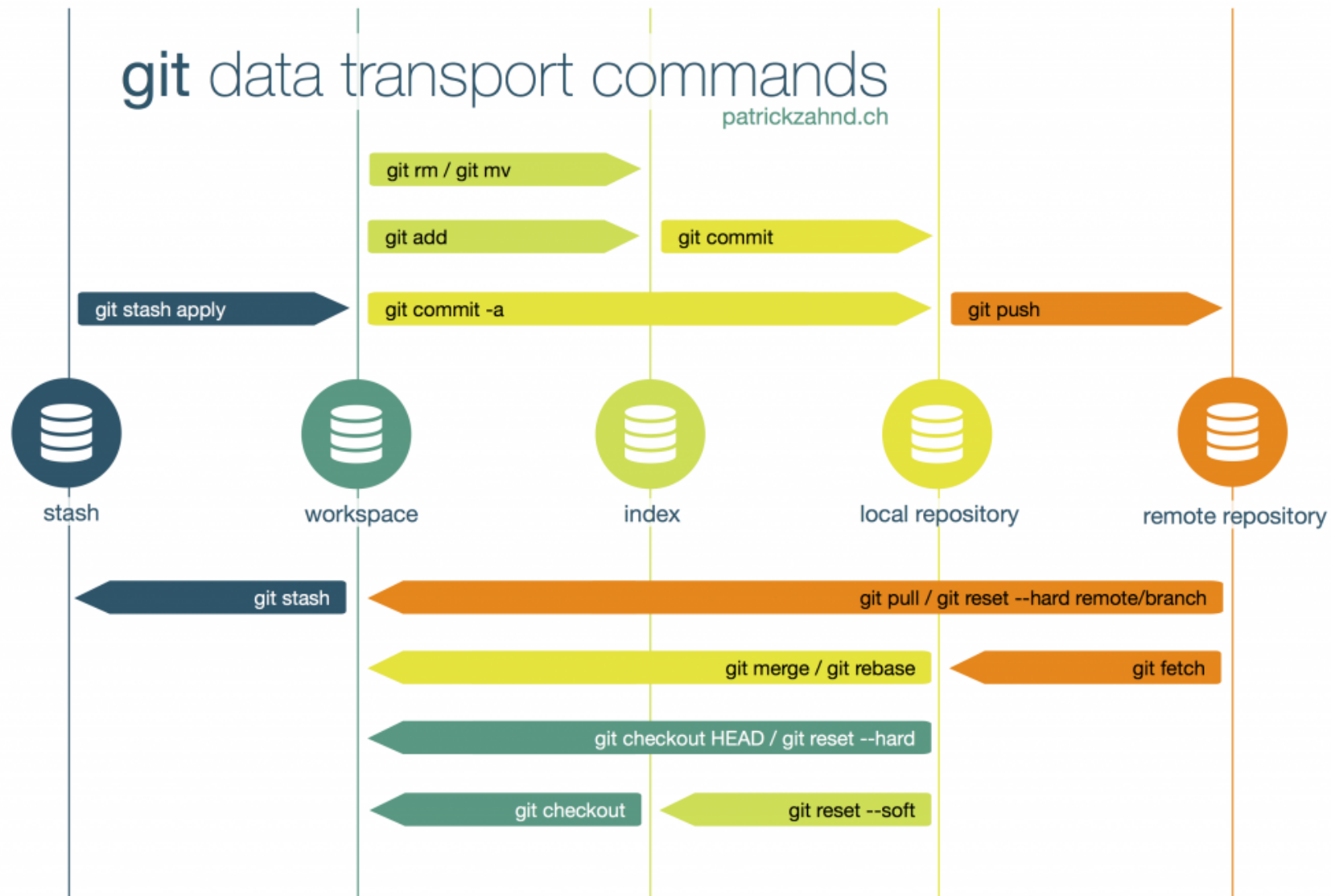
# UPSTREAM

Bei *Forks/Branches* bezeichnet dies das original *Repository*.

# **GIT BENUTZEN**

# git data transport commands

patrickzahnd.ch





***DEMO***

# **SELBST NACHLESEN**

**<https://rogerdudler.github.io/git-guide/index.de.html>**



# **GIT TOOLS**

# **GIT GUI**

Mit diesem Tool kann man in einer GUI nochmals seinen Commit checken, bevor man ihn dann committed.

# **BASH ALIASES**

Wir haben Shortcuts für die wichtigsten git-Befehle angelegt.  
(gPush, gPull & gReset)

# TIG, GITK & GITG

Sind alles Programme, die es euch erleichtern können alte Commits anzusehen.  
(es gibt auch *git log*)

***DEMO***

**GITHUB**

***DEMO***

# FAZIT



# BEGRIFFE

- Workspace
- Repository
- Local Repository
- Remote Repository
- Branch
- Commit
- HEAD

# TOOLS

- git gui
- Bash Aliases
- tig, gitk & gitg
- *uvm...*

# GITHUB

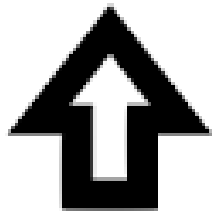
Tool um noch komfortabler gemeinsam an Projekten zu arbeiten.

**FRAGEN?**

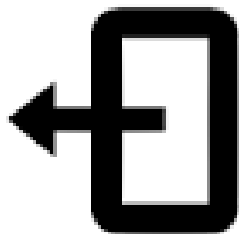
# In case of fire



1. `git commit`



2. `git push`



3. leave building

