

14.–17. 09. 2009
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Igittigitt

Versionskontrolle mit Git

Andreas Schubert

MATHEMA Software GmbH

Git?

- git (Brit.) [coll.]

Blödmann, Depp, Idiot

- "I'm an egotistical bastard, and I name all my projects after myself. First Linux, now git."
(Linus Torvalds)

Git?

- Ist gerade „hipp“, in, ein „Hype“...
- <http://www.git-scm.com>
- Open Source
- Linus Torvalds
- Dezentrales/Verteiltes SourceCodeManagement-System
- große Projekte
 - Linux Kernel (2.6.30: 11.637.173 LOC)

Designprinzipien

- CVS als Beispiel, wie es nicht gemacht werden sollte (im Zweifel genau das Gegenteil)
 - SVN: „CVS done right“ - „There is no way to do CVS right“
- Unterstützung für verteilte Arbeitsabläufe, ähnlich wie BitKeeper
- Sehr starke Absicherung gegen Datenverlust, egal ob versehentlich oder absichtlich herbeigeführt
- Sehr hohe Geschwindigkeit

Charakteristika

- **Non-linear Development**
 - Schnelles Branchen und Mergen
 - Tools zum Visualisieren und Navigieren in der nicht-linearen Historie
 - Es wird häufiger „gemerged“ als geändert
- **Distributed Development**
 - jeder Entwickler hat komplettes Repository mit der gesamten Historie
 - Änderungen werden als Branch weitergegeben
- **Repositories können publiziert werden**
 - per HTTP(S)/FTP/rsync/SSH/...

Charakteristika

- Authentizität der Historie
 - Cryptographie
 - der Name (SHA1) einer bestimmten Revision (commit) ist von der kompletten Historie, die zu dieser Version geführt hat, abhängig
 - dadurch ist es unmöglich, die Historie zu manipulieren, ohne den Namen der Revision zu ändern
- Toolkit Design
 - „Plumping“ und „Porcelain“
 - Plumping: „core“ git Kommandos
 - Porcelain: Wrapper um die „core“ Kommandos
 - eigenes Porzellan ist möglich

Warum Git?

- Leichtgewichtige lokale Branches
- Alles ist lokal
- Git ist schnell
- Git ist klein
- Die Staging Area
- Verteilte Entwicklung
- Beliebige Arbeitsabläufe
- GitHub
- relativ einfach zu lernen

Warum nicht?

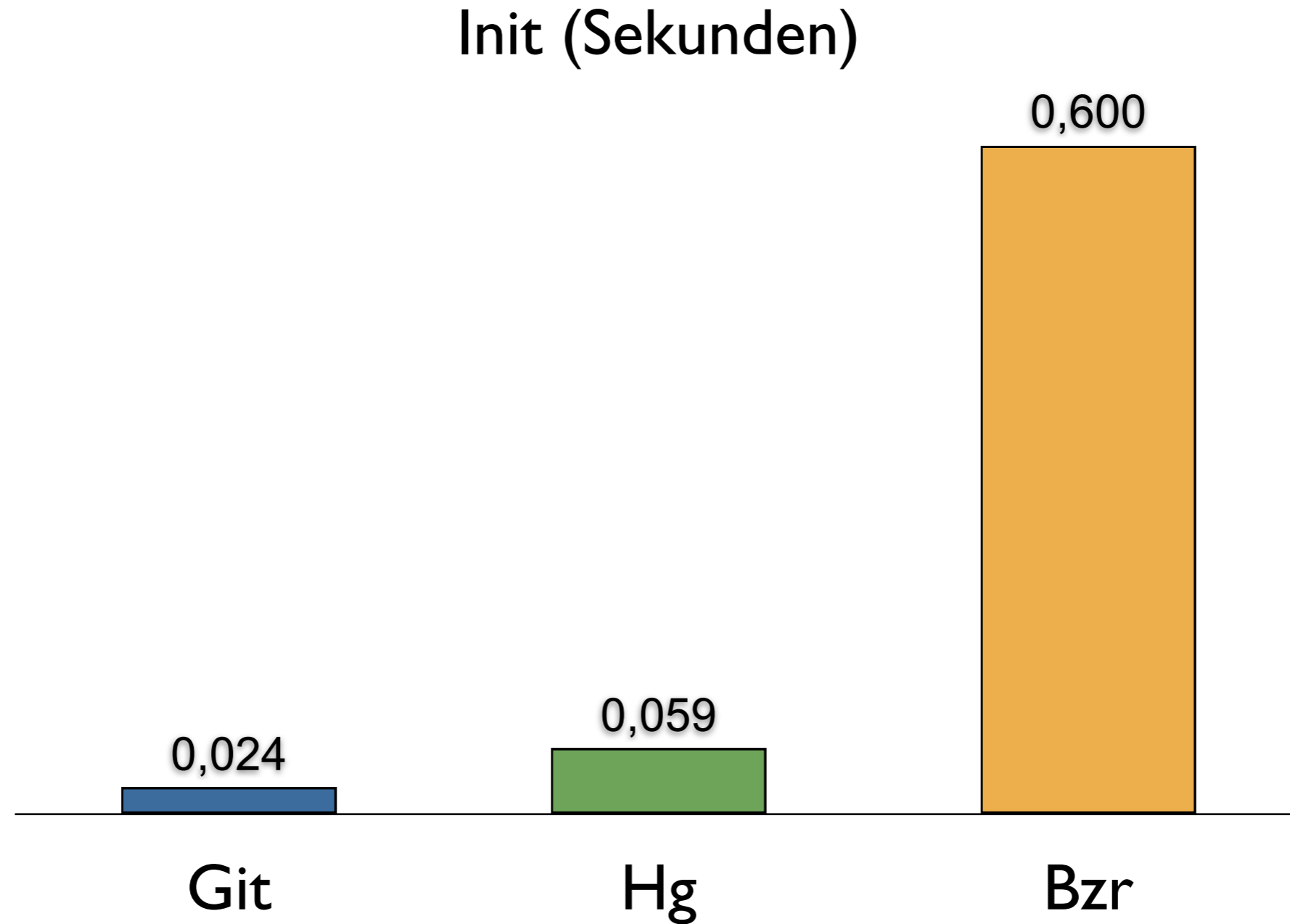
- kein explizites Tracking von Renames
 - wobei: „I do not see it as a tradeoff not to record renames. It **is** a feature.“ (Linus Torvalds)
- Unterstützung in Eclipse (noch) eher rudimentär
- Misstrauen gegenüber Heuristiken
- Man ist mit dem zufrieden, was man hat

Leichtgewichtige lokale Branches

- Andere SCM: kompletter Klon des Repositories in eigenes Verzeichnis beste Strategie
- Erzeugen, wechseln, zusammenführen und löschen ist schnell und einfach
- Im gleichen Working Directory
- Branches müssen nicht veröffentlicht werden
- Änderung der Arbeitsweise
 - für jedes Feature ein eigener Branch
 - für Ideen und Experimente ein eigener Branch
 - für ... ein eigener Branch

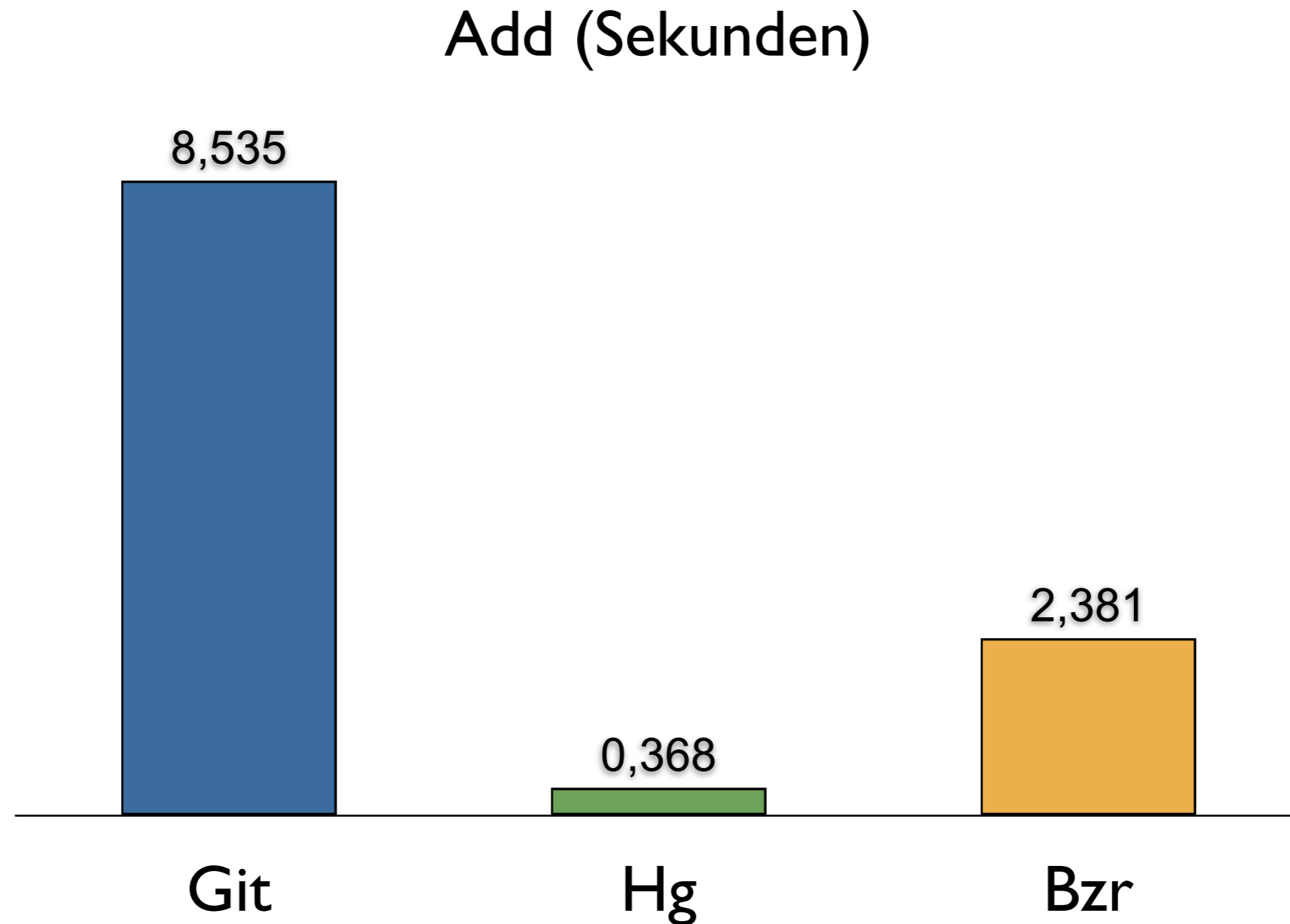
Git ist schnell

(Quelle: Scott Chacon, whygitisbetterthanx.com)



Git ist schnell

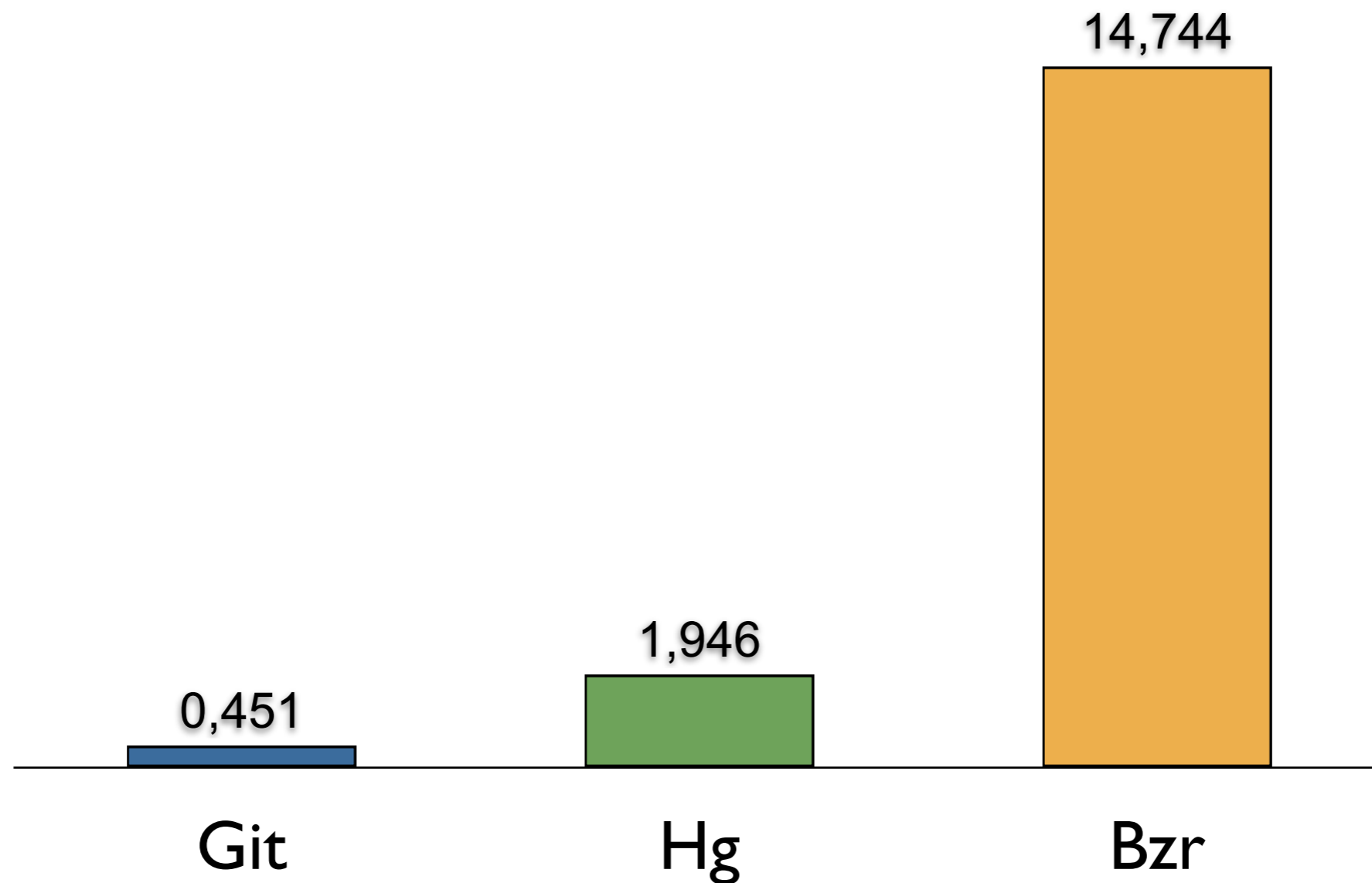
(Quelle: Scott Chacon, whygitisbetterthanx.com)



Git ist schnell

(Quelle: Scott Chacon, whygitisbetterthanx.com)

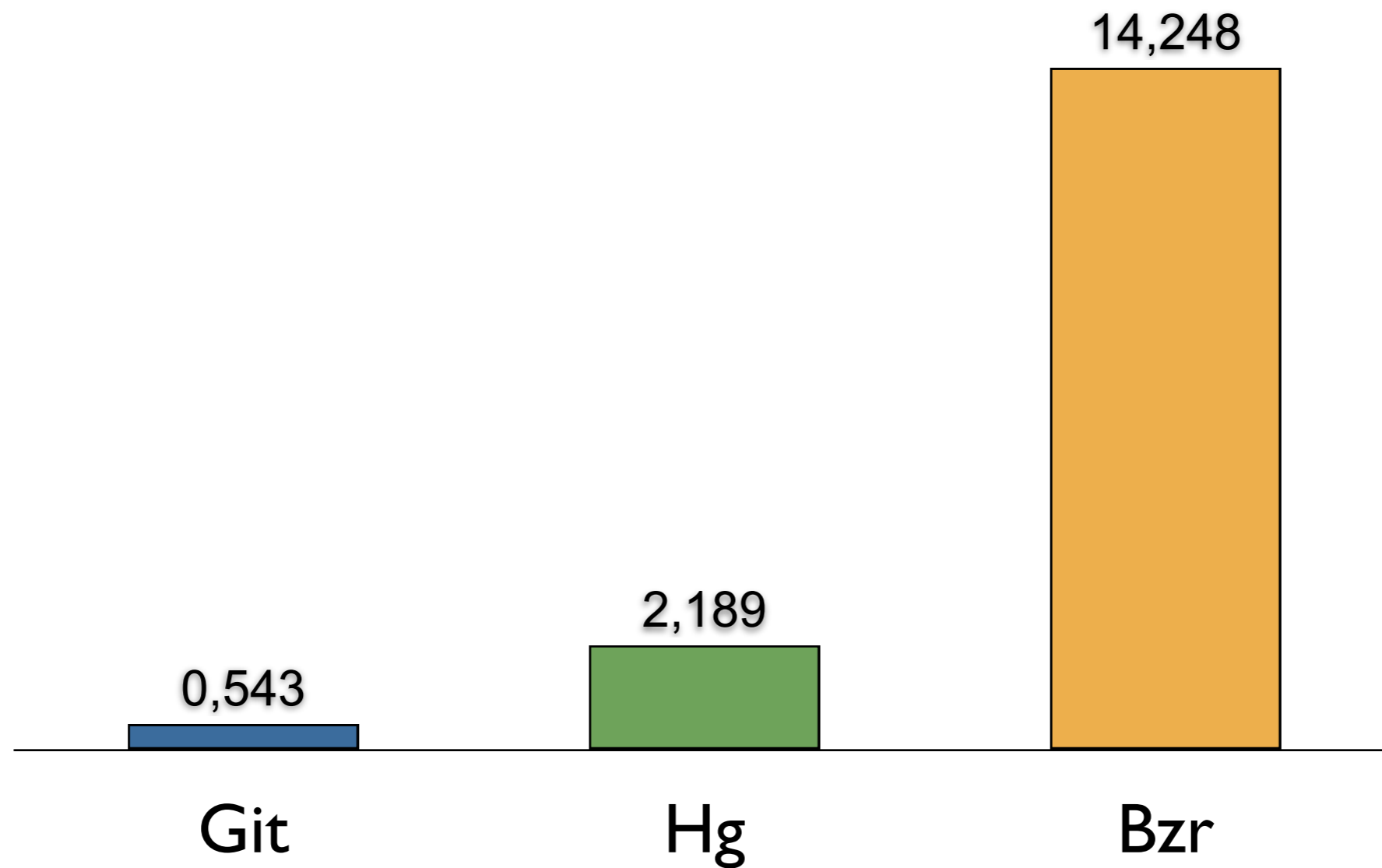
Status (Sekunden)



Git ist schnell

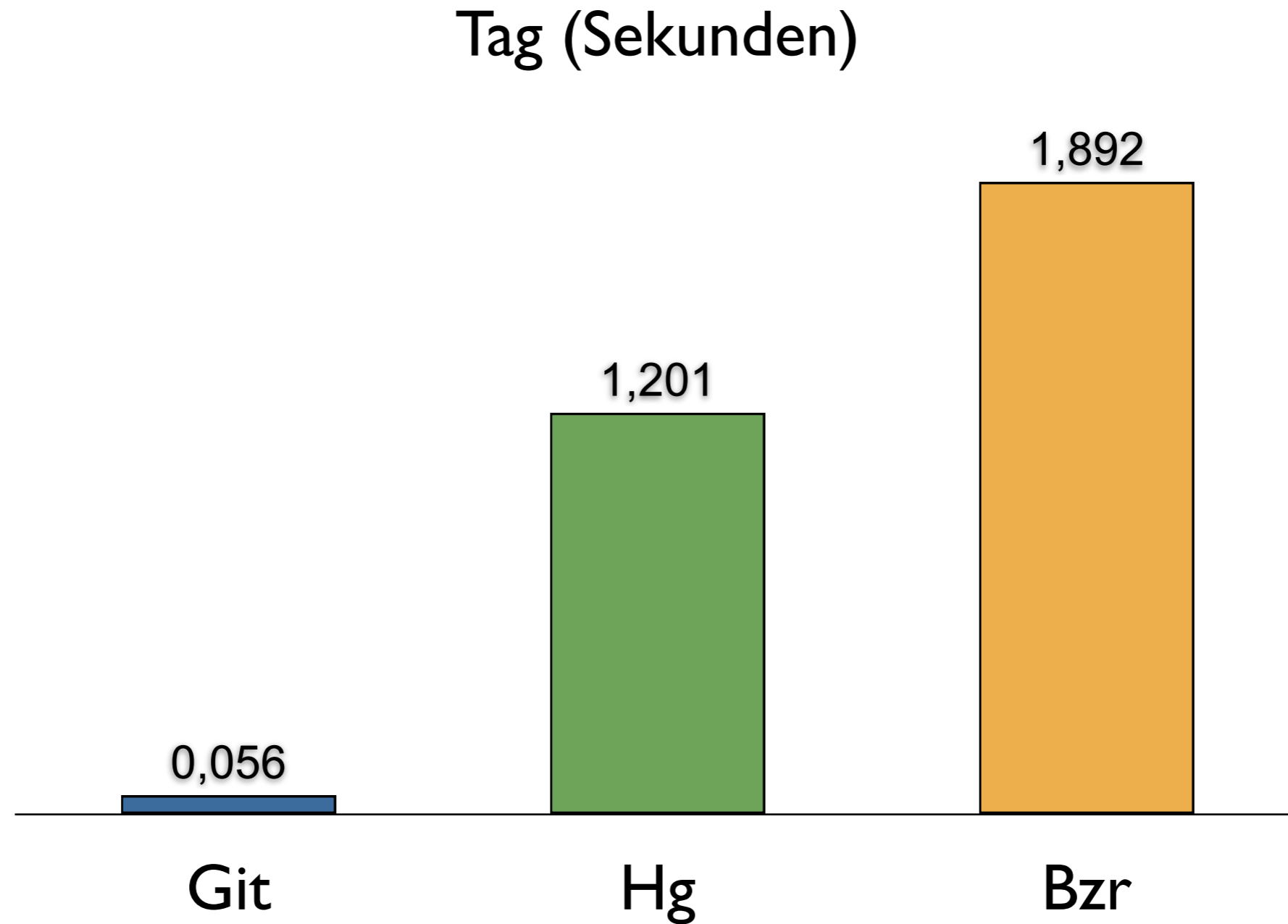
(Quelle: Scott Chacon, whygitisbetterthanx.com)

Diff (Sekunden)



Git ist schnell

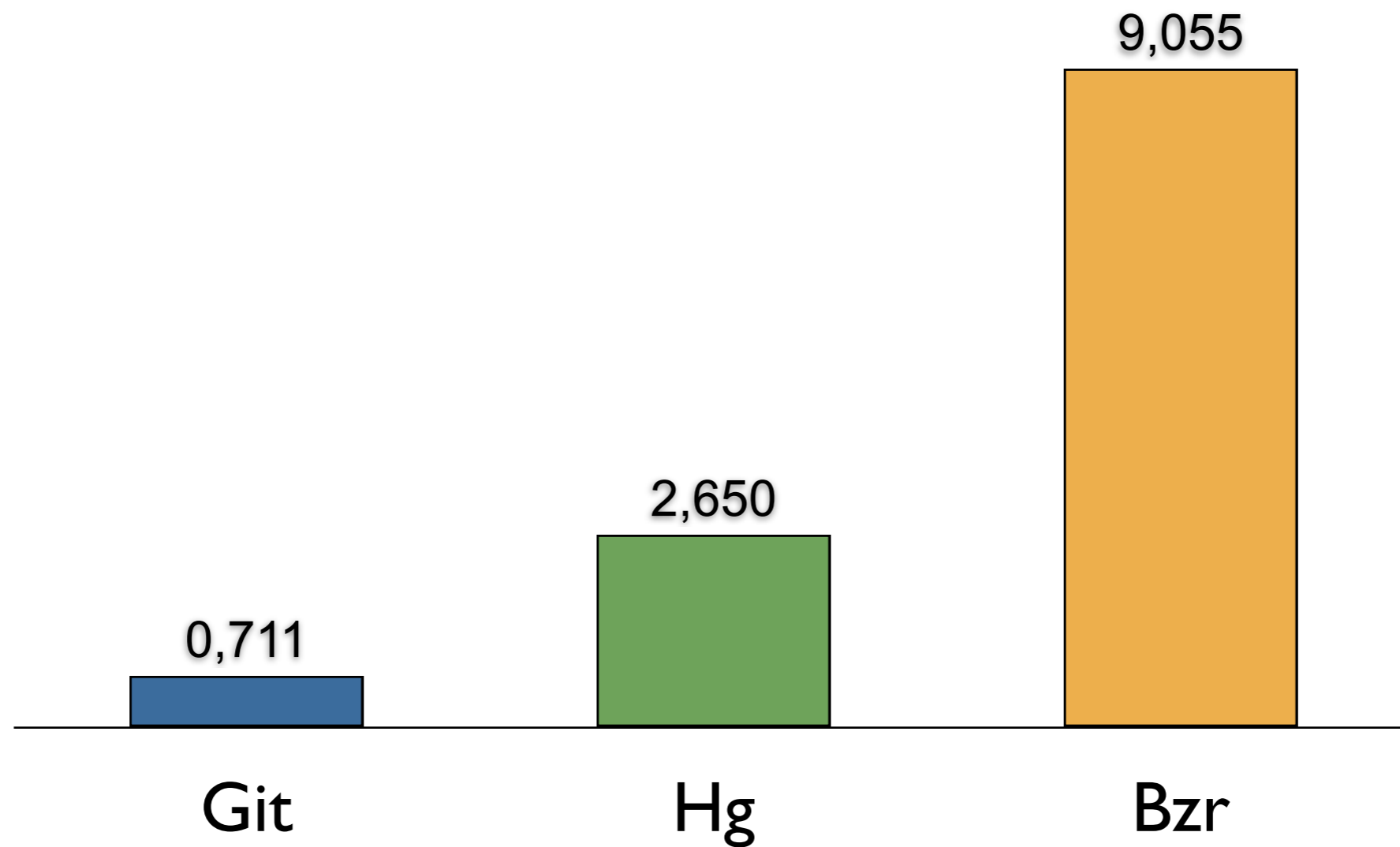
(Quelle: Scott Chacon, whygitisbetterthanx.com)



Git ist schnell

(Quelle: Scott Chacon, whygitisbetterthanx.com)

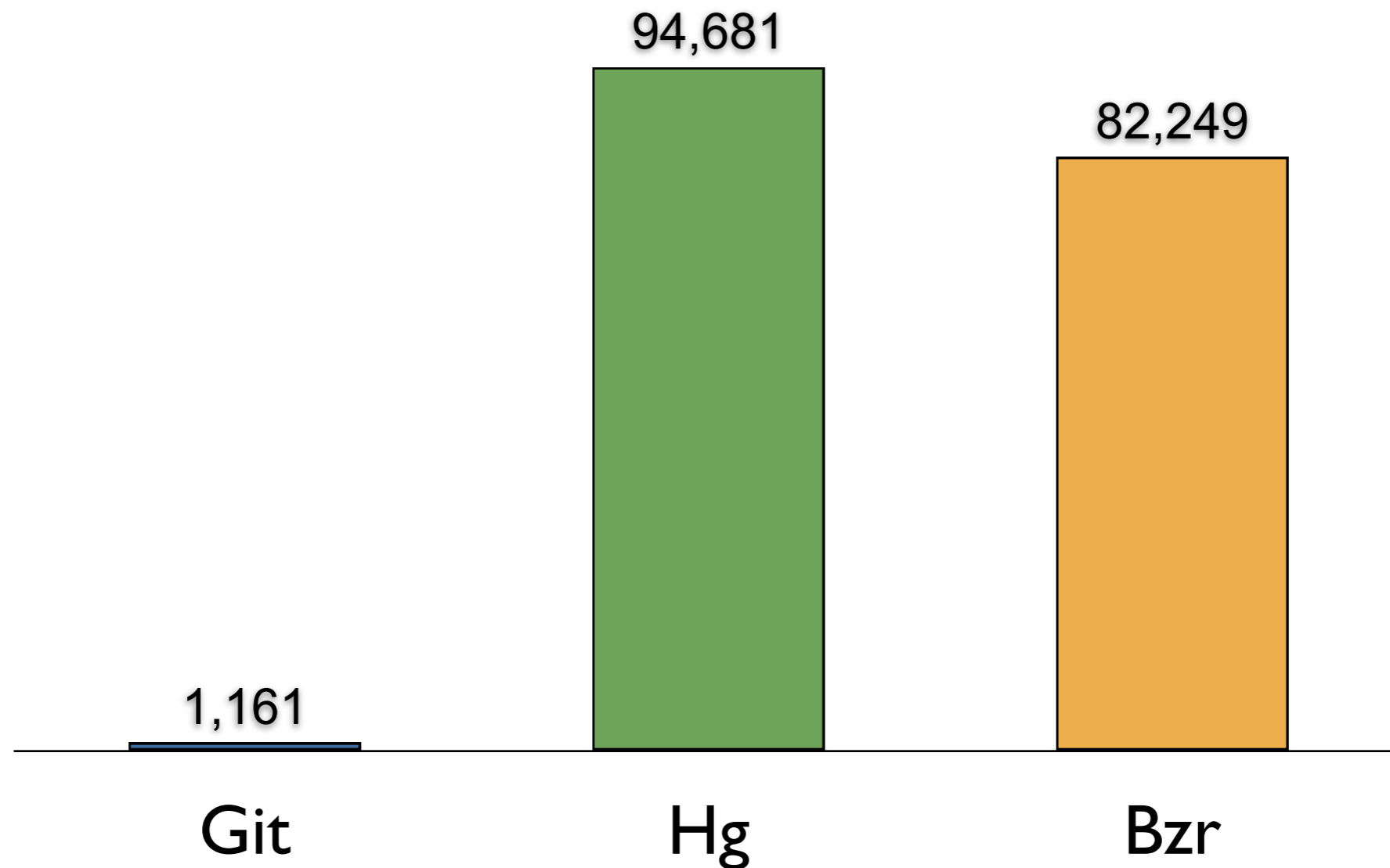
Log (Sekunden)



Git ist schnell

(Quelle: Scott Chacon, whygitisbetterthanx.com)

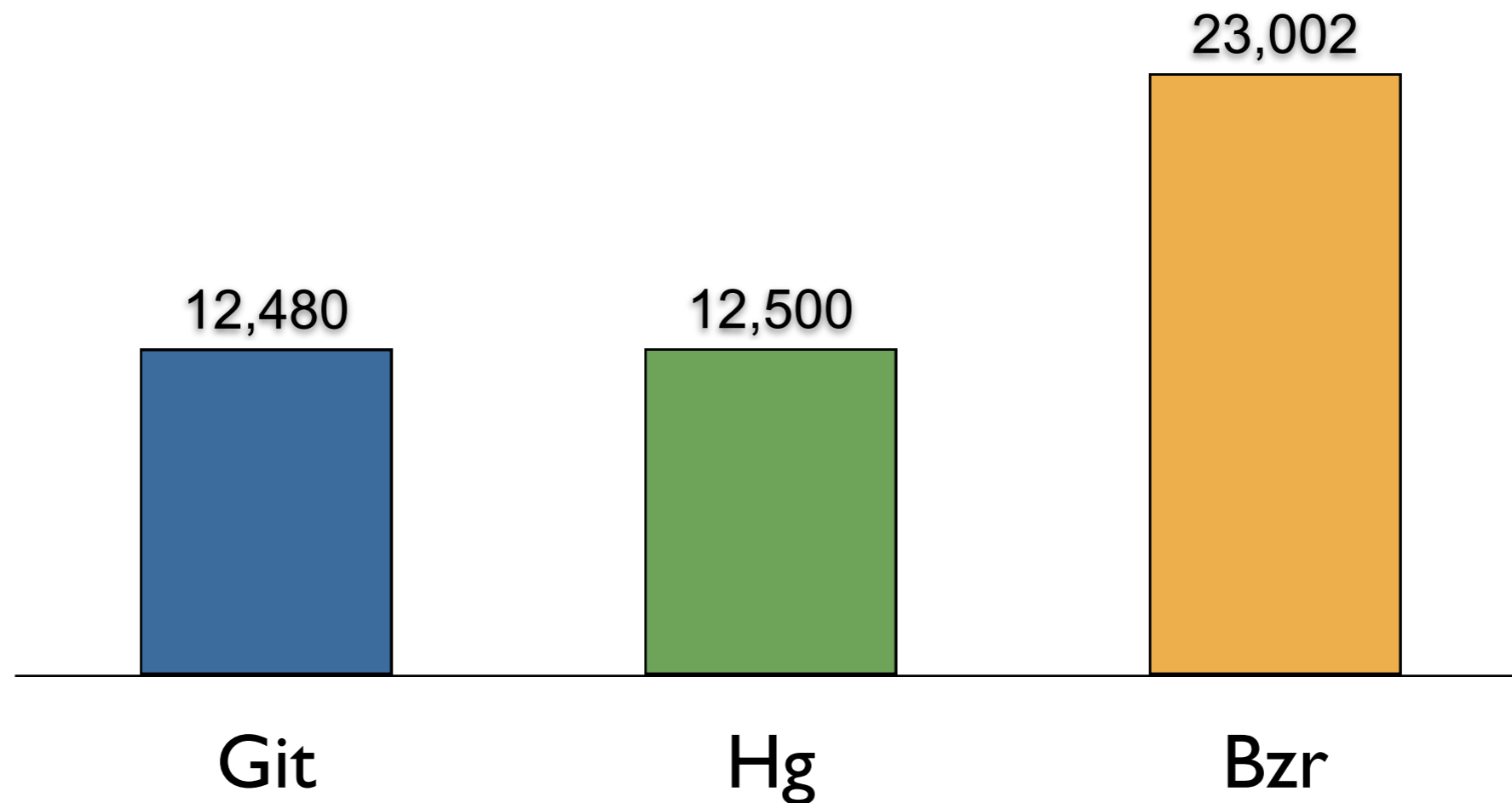
Branch, kalt (Sekunden)



Git ist schnell

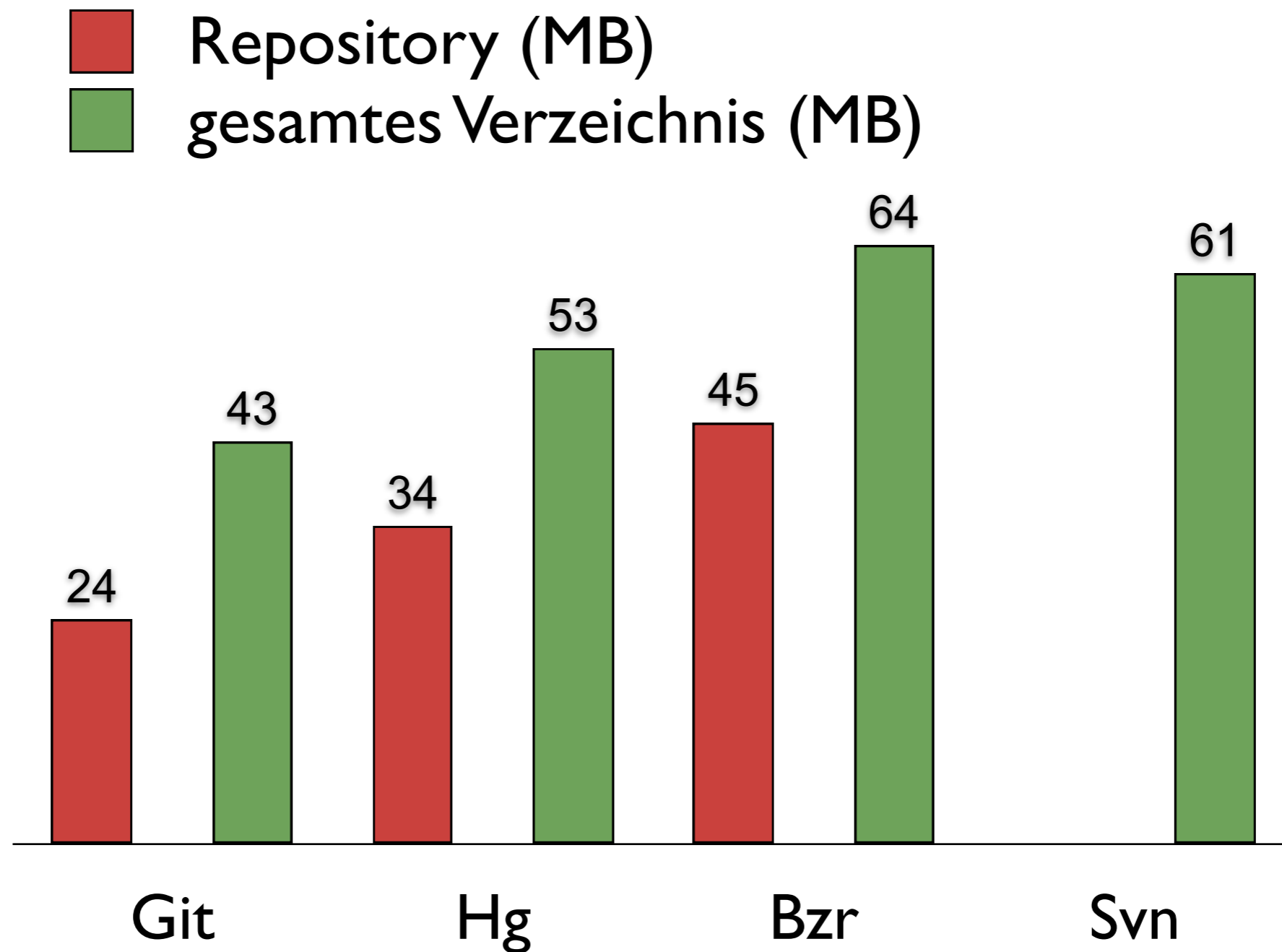
(Quelle: Scott Chacon, whygitisbetterthanx.com)

Commit, groß (Sekunden)

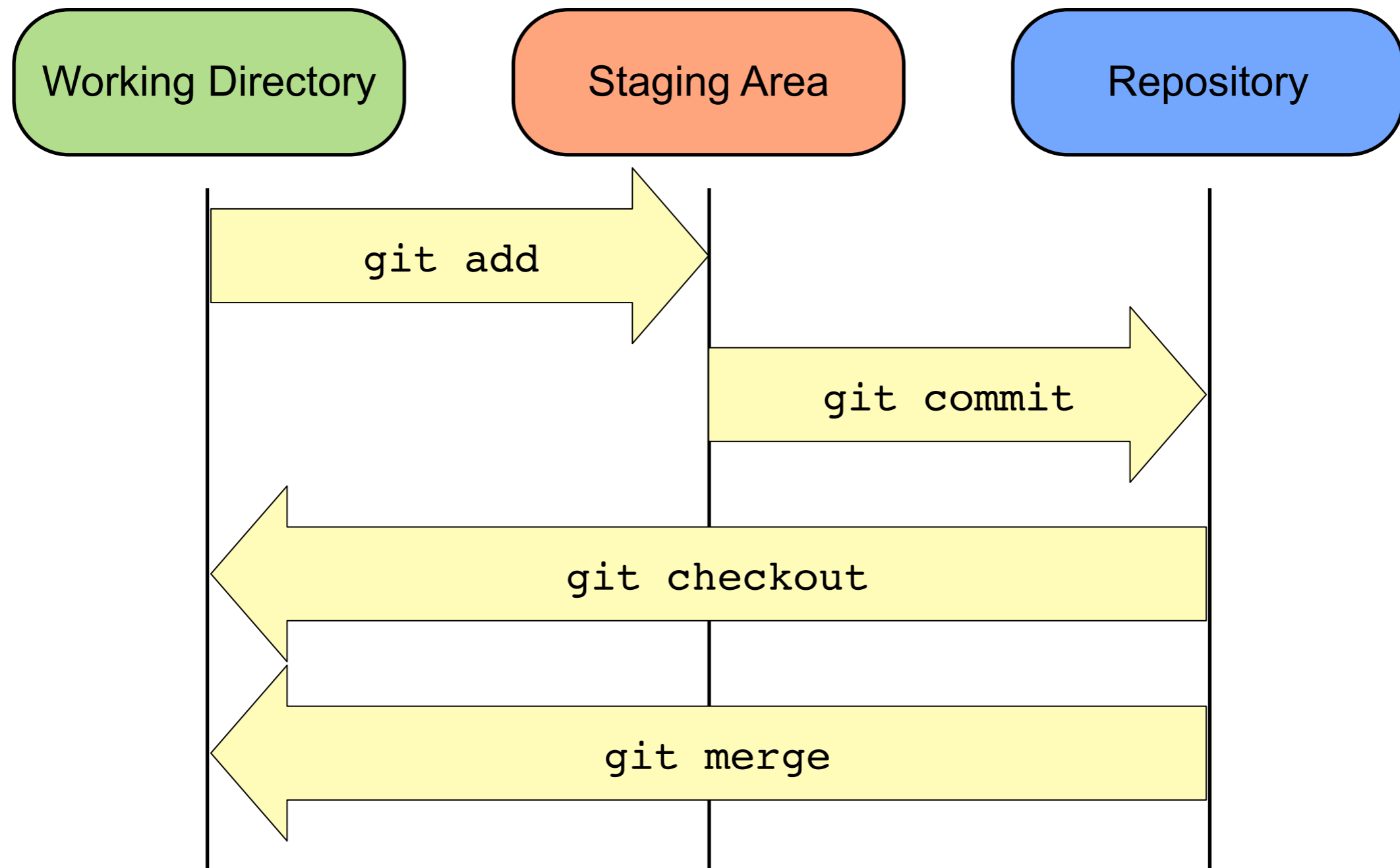


Git ist klein

(Quelle: Scott Chacon, whygitisbetterthanx.com)



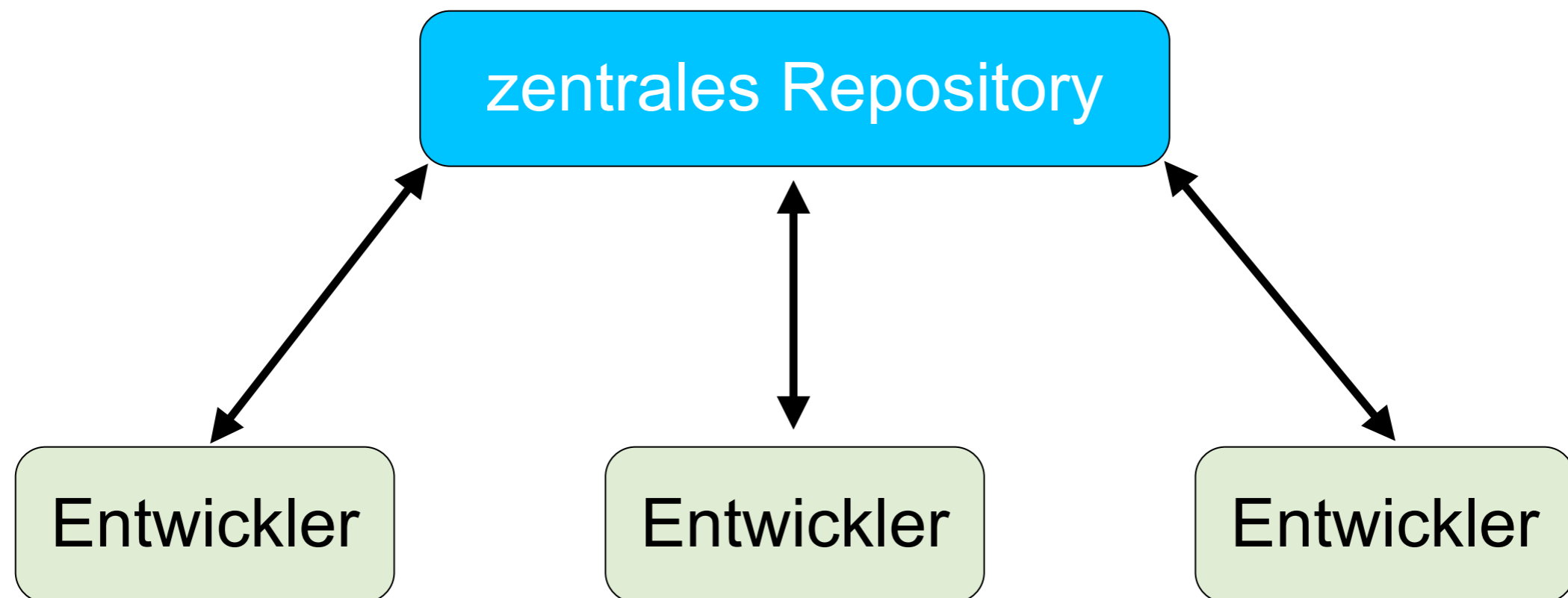
Die Staging Area



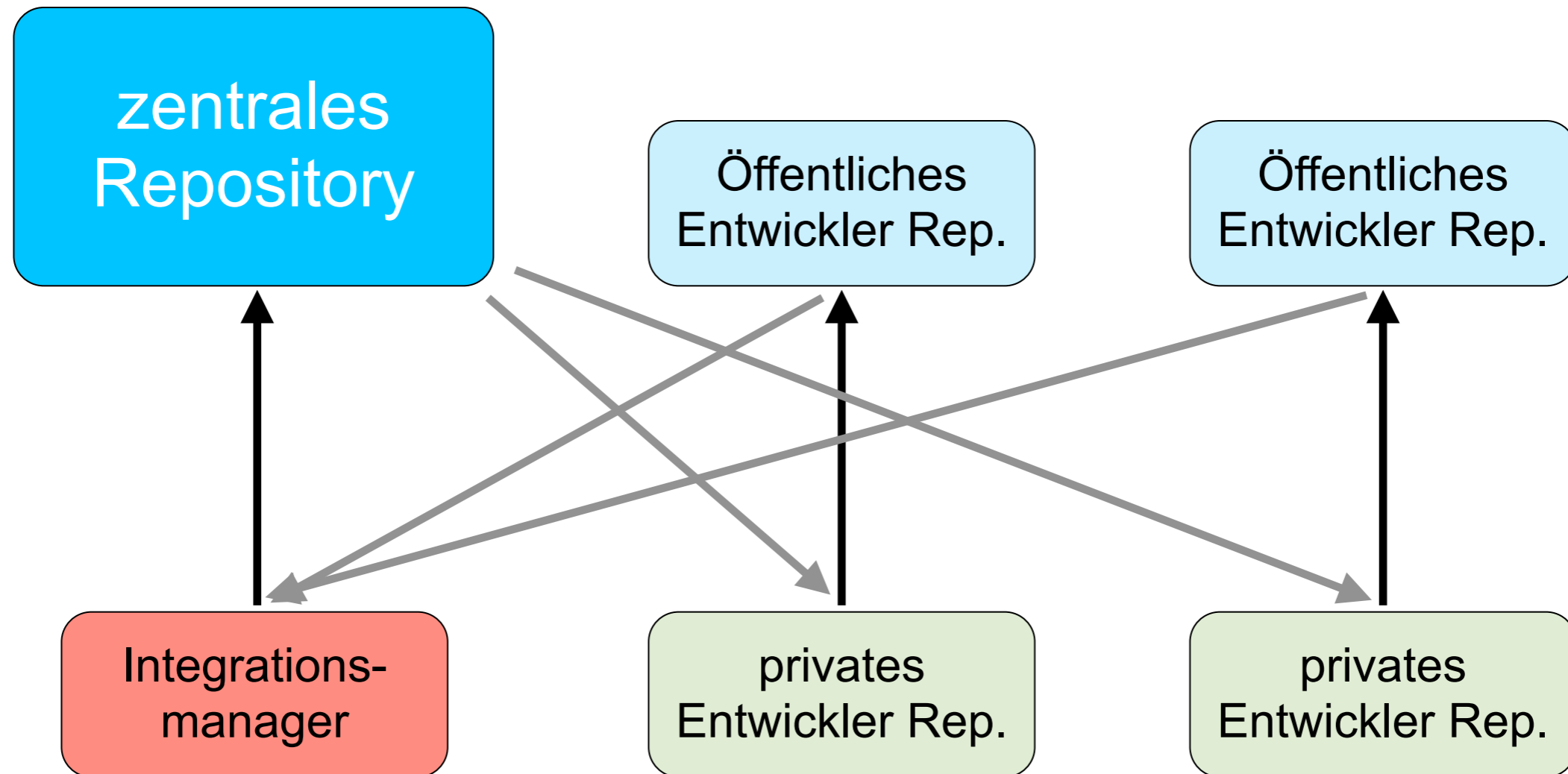
Beliebige Arbeitsabläufe (eine Auswahl)

- Zentralisiert
(Subversion-Stil)
- Integrationsmanager
- Diktator-Leutnant
- ...

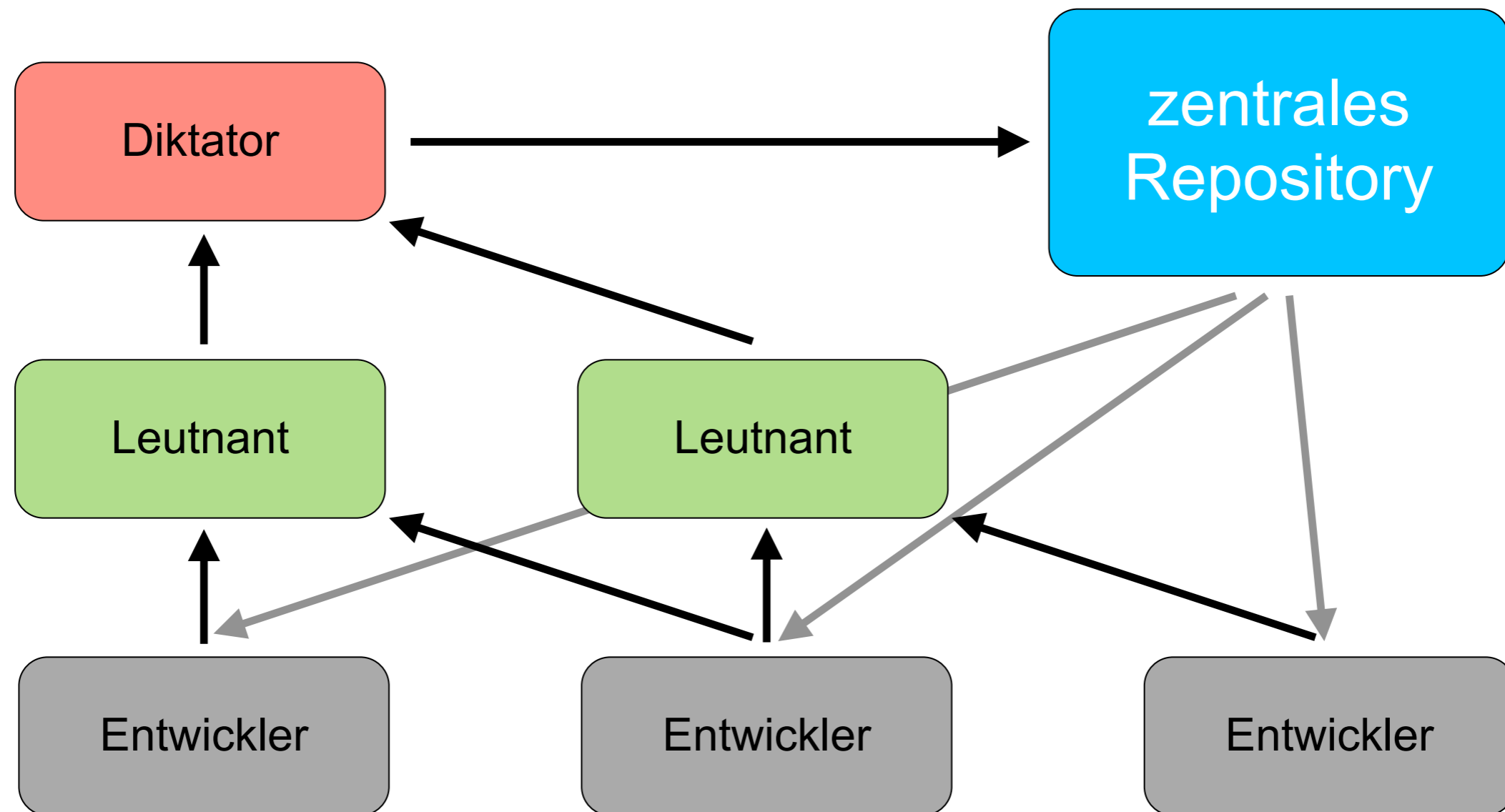
Zentralisiert



Integrationsmanager



Diktator - Leutnant



Kommandos

```

git — 128x33
> git
Display all 140 possibilities? (y or n)
git
git-add
git-add --interactive
git-am
git-annotate
git-apply
git-archimport
git-archive
git-bisect
git-blame
git-branch
git-bundle
git-cat-file
git-check-attr
git-check-ref-format
git-checkout
git-checkout-index
git-cherry
git-cherry-pick
git-citool
git-clean
git-clone
git-commit
git-commit-tree
git-config
git-count-objects
git-cvsexportcommit
git-cvsexport
git-cvsserver
git-daemon
git-describe
git-diff
git-diff-files
git-diff-index
git-diff-tree
git-fast-export
git-fast-import
git-fetch
git-fetch-tool
git-fetch-pack
git-filter-branch
git-fmt-merge-msg
git-for-each-ref
git-format-patch
git-fsck
git-fsck-objects
git-gc
git-get-commit-id
git-grep
git-gui
git-hash-object
git-http-fetch
git-http-push
git-imap-send
git-index-pack
git-init
git-init-db
git-instaweb
git-log
git-lost-found
git-ls-files
git-ls-remote
git-ls-tree
git-mailinfo
git-maildir
git-merge
git-merge-base
git-merge-file
git-merge-index
git-merge-octopus
git-merge-one-file
git-merge-ours
git-merge-recursive
git-merge-resolve
git-merge-stupid
git-merge-subtree
git-merge-tree
git-mergetool
git-mktag
git-mktag
git-mv
git-name-rev
git-pack-objects
git-pack-redundant
git-pack-refs
git-parse-remote
git-patch-id
git-peek-remote
git-prune
git-prune-packed
git-pull
git-push
git-quiltimport
git-read-tree
git-rebase
git-rebase --interactive
git-receive-pack
git-reflog
git-reinit
git-remote
git-repack
git-repo-config
git-request-pull
git-revert
git-revert
git-revert
git-rev-parse
git-revert
git-rm
git-send-email
git-send-pack
git-sh-setup
git-shell
git-shortlog
git-show
git-show-branch
git-show-index
git-show-ref
git-stash
git-status
git-strip-space
git-submodule
git-svn
git-symbolic-ref
git-tag
git-tag-tree
git-unpack-file
git-unpack-objects
git-update-index
git-update-ref
git-update-server-info
git-upload-archive
git-upload-pack
git-var
git-verify-pack
git-verify-tag
git-web --browse
git-whatchanged
git-write-tree
gitk
  
```


Kommandos

```
git — 128x33
> git
Display all 140 possibilities? (y or n)
git
git-add
git-add--interactive
git-am
git-annotate
git-apply
git-archimport
git-archive
git-bisect
git-blame
git-branch
git-bundle
git-cat-file
git-check-attr
git-check-ref-format
git-checkout
git-checkout-index
git-cherry
git-cherry-pick
git-citool
git-clean
git-clone
git-commit
git-commit-tree
git-config
git-count-objects
git-cvsexportcommit
git-cvsiimport
git-cvsserver
git-daemon
git-describe
git-diff
git-diff-files
git-diff-index
git-diff-tree
git-fast-export
git-fast-import
git-fetch
git-fetch--tool
git-fetch-pack
git-filter-branch
git-fmt-merge-msg
git-for-each-ref
git-format-patch
git-fsck
git-fsck-objects
git-gc
git-get-tar-commit-id
git-grep
git-graph-objects
git-http-fetch
git-http-push
git-imap-send
git-index-pack
git-init
git-init-db
git-instaweb
git-log
git-lost-found
git-ls-files
git-ls-remote
git-ls-tree
git-mailinfo
git-mailsplit
git-merge
git-merge-base
git-merge-file
git-merge-index
git-merge-octopus
git-merge-one-file
git-merge-ours
git-merge-recursive
git-merge-resolve
git-merge-stupid
git-merge-subtree
git-merge-tree
git-mergetool
git-mktree
git-mv
git-name-rev
git-pack-objects
git-pack-redundant
git-pack-refs
git-parse-remote
git-patch-id
git-peek-remote
git-prune
git-prune-packed
git-pull
git-push
git-quiltimport
git-read-tree
git-rebase
git-rebase--interactive
git-receive-pack
git-reflog
git-relink
git-remote
git-repack
git-repo-config
git-request-pull
git-rerere
git-reset
git-rev-list
git-rev-parse
git-revert
git-rm
git-send-email
git-send-pack
git-sh-setup
git-shell
git-shortlog
git-show
git-show-branch
git-show-index
git-show-ref
git-stash
git-status
git-stripspace
git-submodule
git-svn
git-symbolic-ref
git-tag
git-tar-tree
git-unpack-file
git-unpack-objects
git-update-index
git-update-ref
git-update-server-info
git-upload-archive
git-upload-pack
git-var
git-verify-pack
git-verify-tag
git-web--browse
git-whatchanged
git-write-tree
gitk
```

git help <kommando>

git config

git init

git status

git add

git commit

git log

git branch

git checkout

git merge

```
git diff  
git diff --cached  
git diff HEAD  
git diff <com1> <com2>
```

git tag

git commit --amend
git reset
git blame
git mergetool

git bisect



Objekte in Git

- Adressiert über SHA1
 - 2 Objekte sind gleich, wenn der „Name“ (SHA1) gleich ist
 - in unterschiedlichen Repositories wird der gleiche Inhalt über die gleiche SHA1 adressiert
 - Fehlerhafte Objekte können über die SHA1 erkannt werden
- Blob
 - Inhalt - eine Datei
- Tree
 - Im Prinzip ein Verzeichnis - Referenzen auf andere Trees und Blobs
- Commit
 - Zeiger auf einen Tree und Meta-Informationen
- Tag
 - Zeiger auf einen Commit

Objekte in Git

Unterschied zu CVS, Subversion, Mercurial, ...

- Git nutzt kein „Delta Storage“
- nicht der Unterschied von einem Commit zum nächsten wird gespeichert
- Git speichert einen Snapshot aller Dateien zum Zeitpunkt des Commit

Blob-Objekt

- enthält den Inhalt einer Datei
- `git show 6a3f5d...`
- definiert sich nur durch Daten
- keine Referenz auf irgendetwas
- keine Attribute/Metadaten
- noch nicht mal einen Dateinamen
- zwei im Working Directory identische Dateien teilen sich den gleichen Blob

6a3f5d...

Blob	Größe
Zeile 1	
Zeile 2	
Zeile 3	

Tree-Objekt

- Verzeichnis, Zeiger auf andere Trees oder Blobs
- `git ls-tree ca3f5b...`
- Zwei identische Trees teilen sich das gleiche Tree Objekt
- Enthält Dateiattribute, Objekt-Typ, SHA1 und Name
- Nur das Executable-Bit wird berücksichtigt

ca3f5b...

Tree		Größe
blob	5a3f4e...	File1
tree	4f5a3b...	Dir1
tree	2aafb...	Dir2

Commit-Objekt

- Zeiger auf einen Tree
 - Zeiger auf einen Parent (oder keinen beim „root commit“)
 - bei einem „Merge“-Commit können es auch mehrere Parents sein
 - einen Autor
 - einen „Committer“
 - einen Kommentar
-
- enthält keine Informationen über die Veränderung selbst
 - wird durch Vergleich zwischen Tree und Parent(s) berechnet
 - Umbenennungen von Dateien werden nicht explizit gespeichert

cf32ad...

Commit	Größe
tree	5a3f4e...
parent	4f5a3b...
author	John Doe
committer	Jane Doe
Commit Message	

Tag-Objekt

- Zeiger auf einen Commit
- hat einen Namen,
- einen „Autor“
- und eine Beschreibung

e34afc...

Tag	Größe
object	5a3f4e...
type	commit
tagger	Jane Doe
Tag Message	

...und zum Schluss...

Links

- <http://www.git-scm.com>
 - Homepage
 - Sourcecode
 - Dokumentation
- <http://www.gitcasts.com>
 - Screencasts zum Thema
- <http://www.github.com>
 - Git Hoster
 - „Nette“ AJAX Oberfläche
 - Community
- <http://jan-krueger.net/git>
 - Git cheat sheet
- <http://code.google.com/p/tortoisegit/>
 - Windows GUI für Git, ähnlich TortoiseSVN

14.–17. 09. 2009
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Vielen Dank!

Andreas Schubert

MATHEMA Software GmbH