

1.– 4. September 2014
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Motorhaube auf, Kernkonzepte verstehen, sicher einsteigen

Git started für Fortgeschrittene

Benjamin Schmid

eXXcellent solutions gmbh

„GIT STARTED“

FÜR FORTGESCHRITTENE

*Konzepte, Funktionsweise
und Best Practices*

Wer ist **Benjamin**?



B.SCHMID@EXXCELLENT.DE



@BENTOLOR



[HTTP://WWW.EXXCELLENT.DE](http://www.excellent.de)



Warum?



git

VERTEILT

**LEICHTGEWICHTIGE
BRANCHES**

LEISTUNGSFÄHIG

KRYPTOGRAPHISCH
GESICHERT



Repository Format

+

bash/C **Skripte**

Junio C Hamano



BIT

ARCHITEKTUR



Git

ARCHITEKTUR

DATENHALTUNG

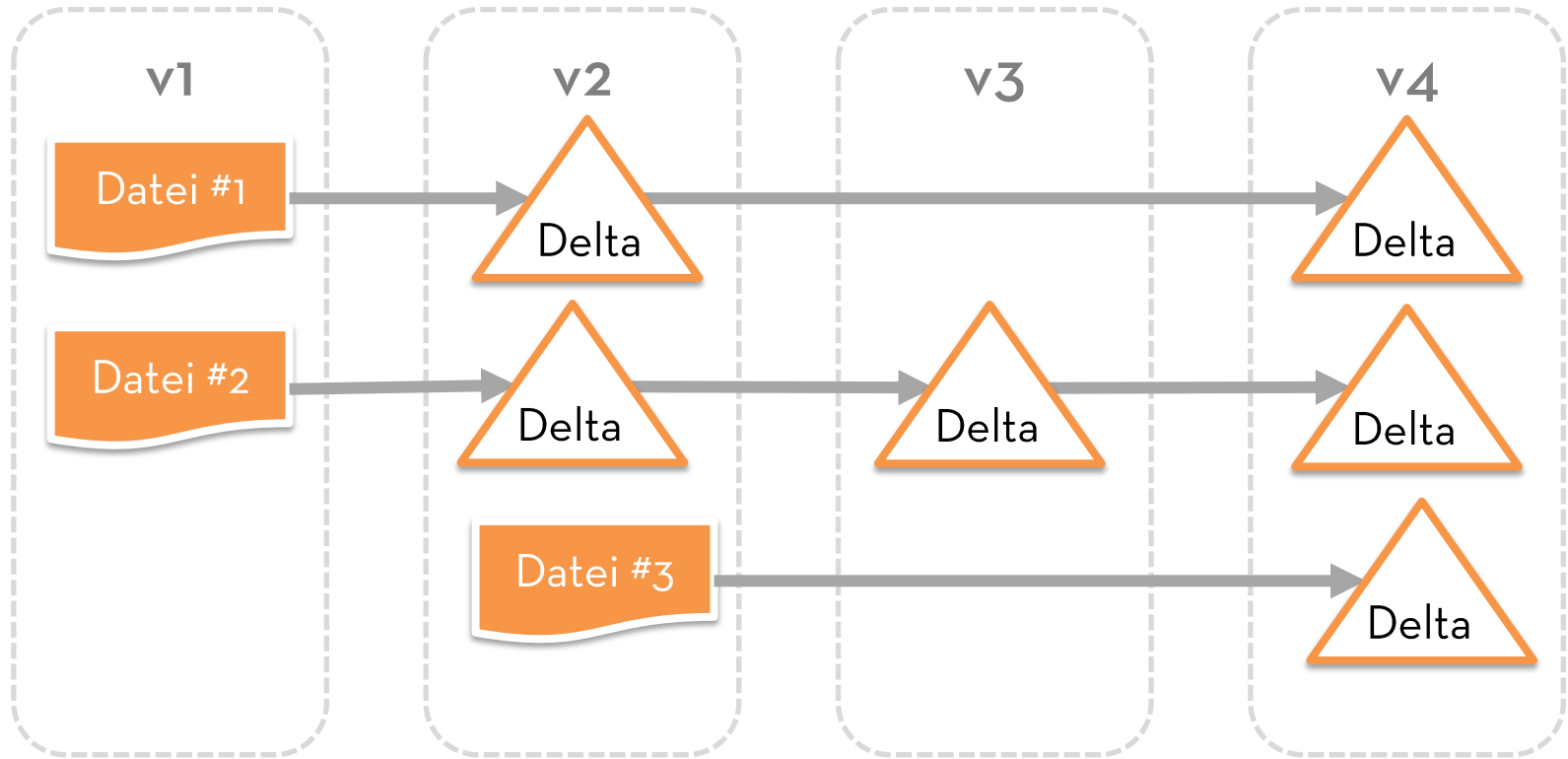
VERSIONS-HISTORIE

BRANCHES

MERGING



Exkurs: Subversion & Co

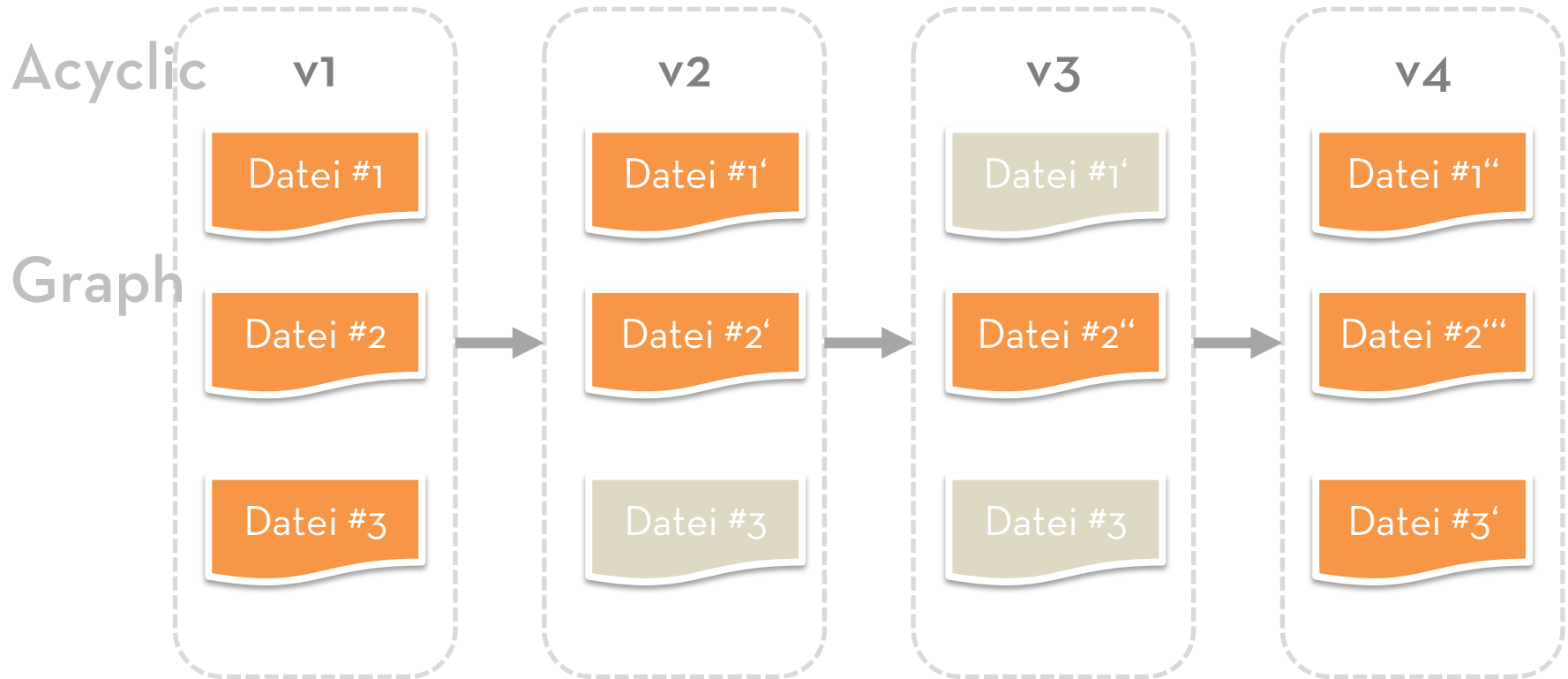


\$ cp -r projekt repository/projekt.v2

Jeder **Git Commit** =
vollständige 1:1 Kopie
der *gesamten* Dateibaums





DAG IN GIT

irect



Quasi: `$ cp -r projekt repository/projekt.v2`

GIT OBJECTS

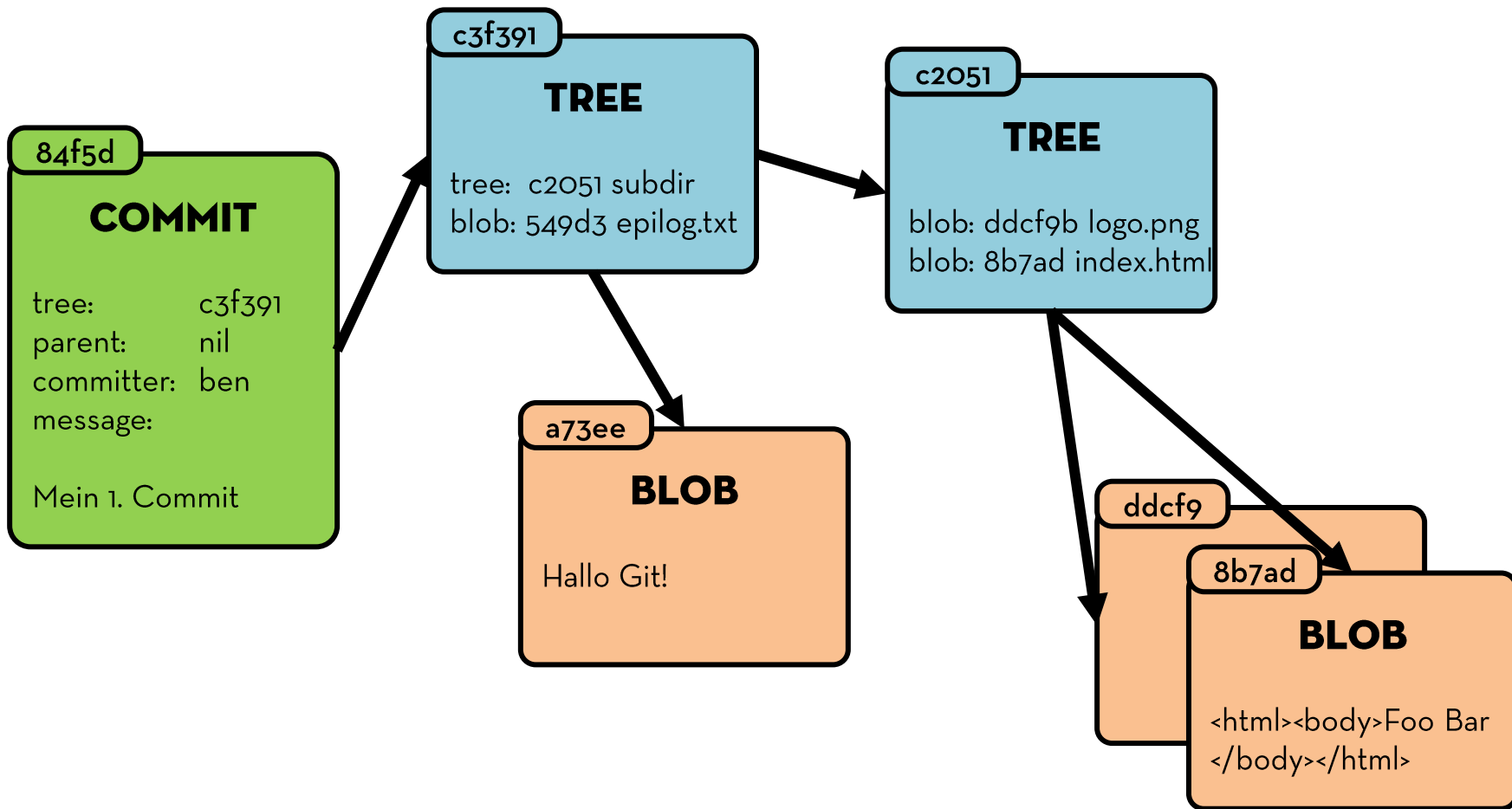
-  **blob** Dateiinhalt
-  **tree** Verzeichnis
-  **commit**
-  **tag** signierter Name



Objekte werden anhand
SHA1-Hash ihres Inhalts
identifiziert &
abgespeichert

a94d7c21aa85415a9a721033807ea5bd4b95e98d

Kurzform: a94d7c

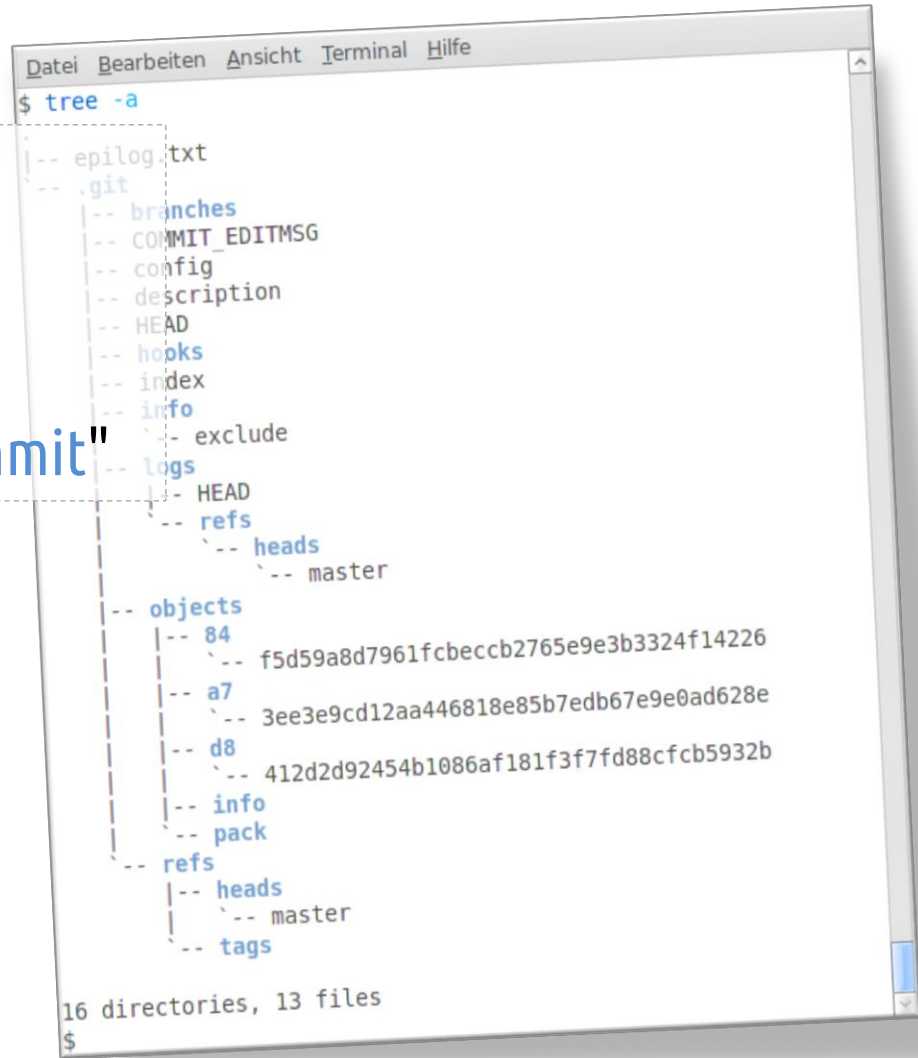


```
$ git init
```

```
$ echo "Hallo Git!" > epilog.txt
```

```
$ git add epilog.txt
```

```
$ git commit -m "Mein erster Commit"
```



```
Datei Bearbeiten Ansicht Terminal Hilfe
$ tree -a
|-- epilog.txt
|-- .git
|   |-- branches
|   |-- COMMIT_EDITMSG
|   |-- config
|   |-- description
|   |-- HEAD
|   |-- hooks
|   |-- index
|   |-- info
|   |   |-- exclude
|   |-- logs
|   |   |-- HEAD
|   |-- refs
|   |   |-- heads
|   |   |-- master
|   |-- objects
|   |   |-- 84
|   |   |   |-- f5d59a8d7961fcbeccb2765e9e3b3324f14226
|   |   |-- a7
|   |   |   |-- 3ee3e9cd12aa446818e85b7edb67e9e0ad628e
|   |   |-- d8
|   |   |   |-- 412d2d92454b1086af181f3f7fd88cfcb5932b
|   |   |-- info
|   |-- pack
|   |-- refs
|   |   |-- heads
|   |   |-- master
|   |   |-- tags
16 directories, 13 files
$
```



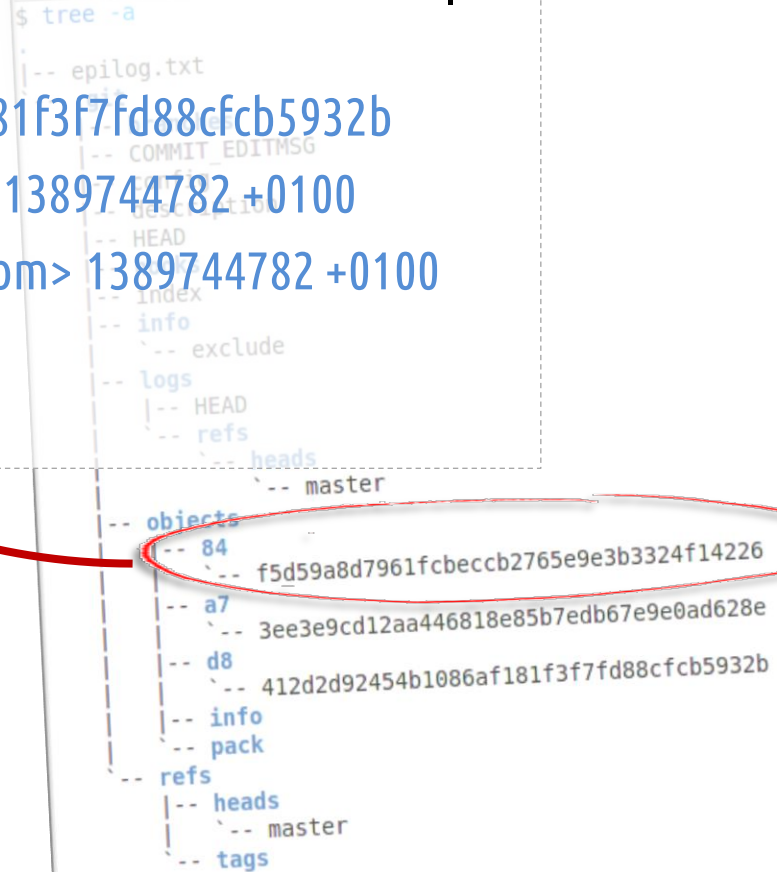
```
$ cat .git/objects/84/f5d59a8d7961fcbeccb2765e9e3b3324f14226 |  
zlib-flate -uncompress; echo
```

```
commit 181tree d8412d2d92454b1086af181f3f7fd88cfcb5932b
```

```
author Benjamin Schmid <foo@bar.com> 1389744782 +0100
```

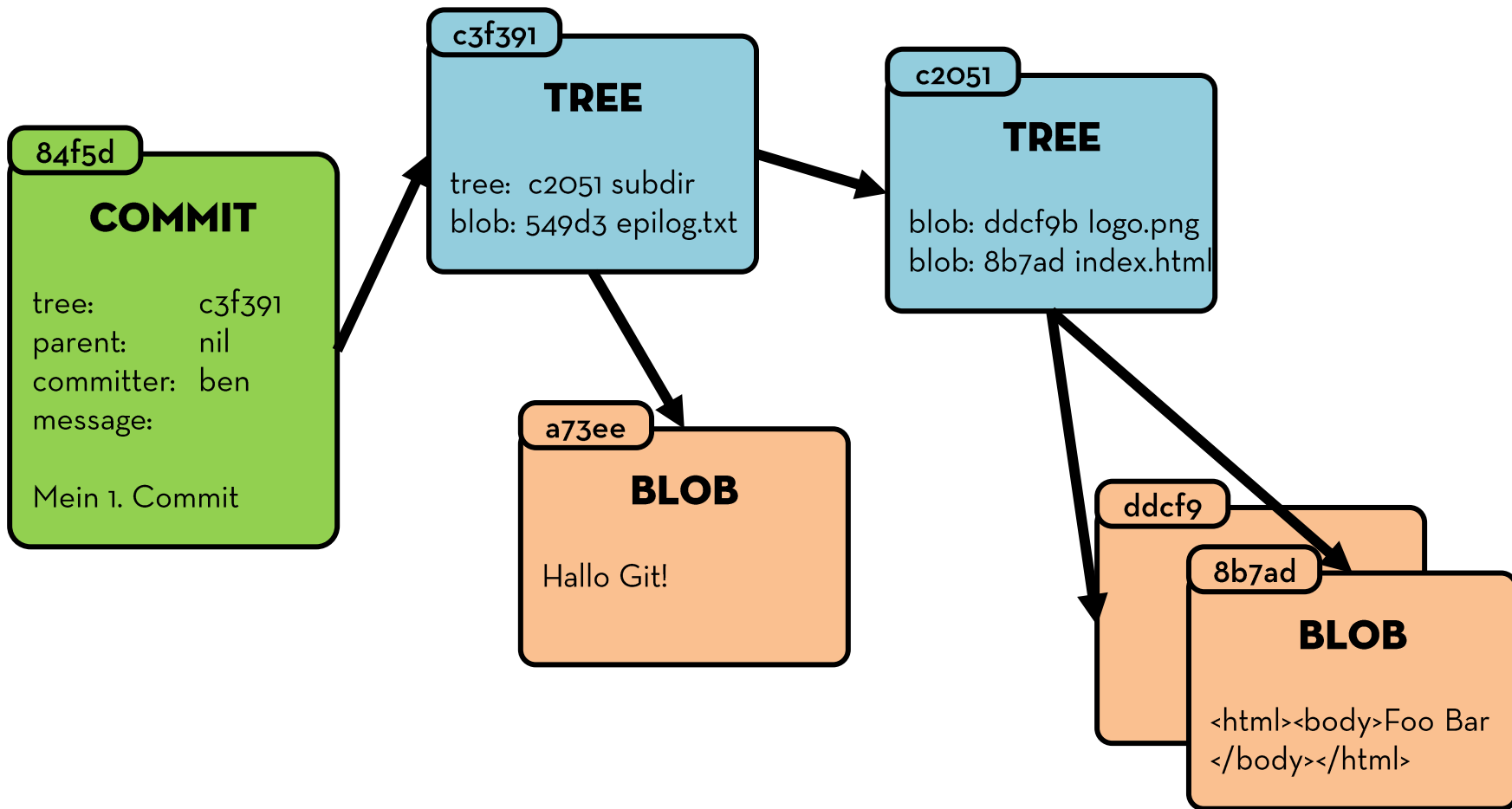
```
committer Benjamin Schmid <foo@bar.com> 1389744782 +0100
```

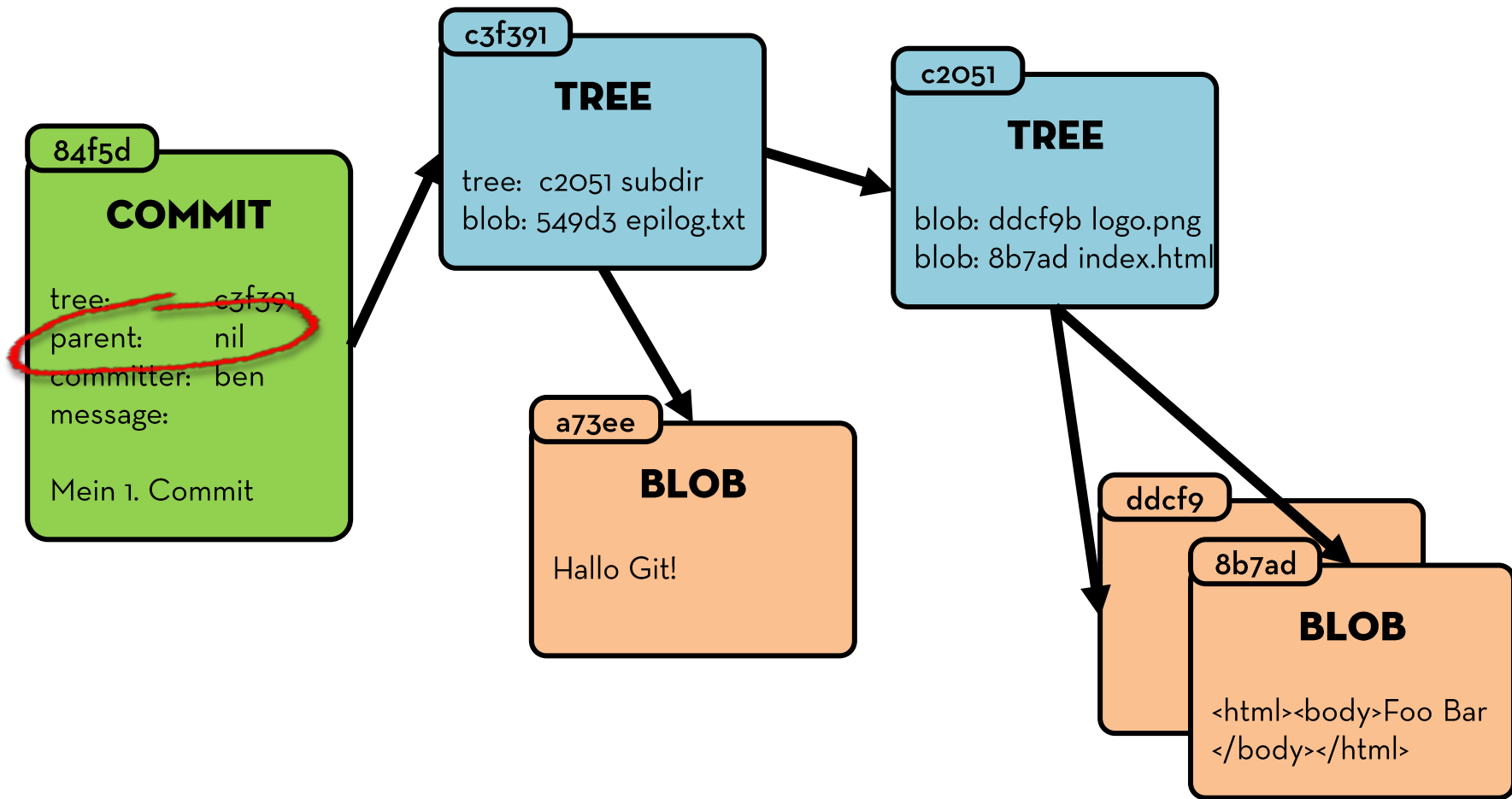
Mein erster Commit



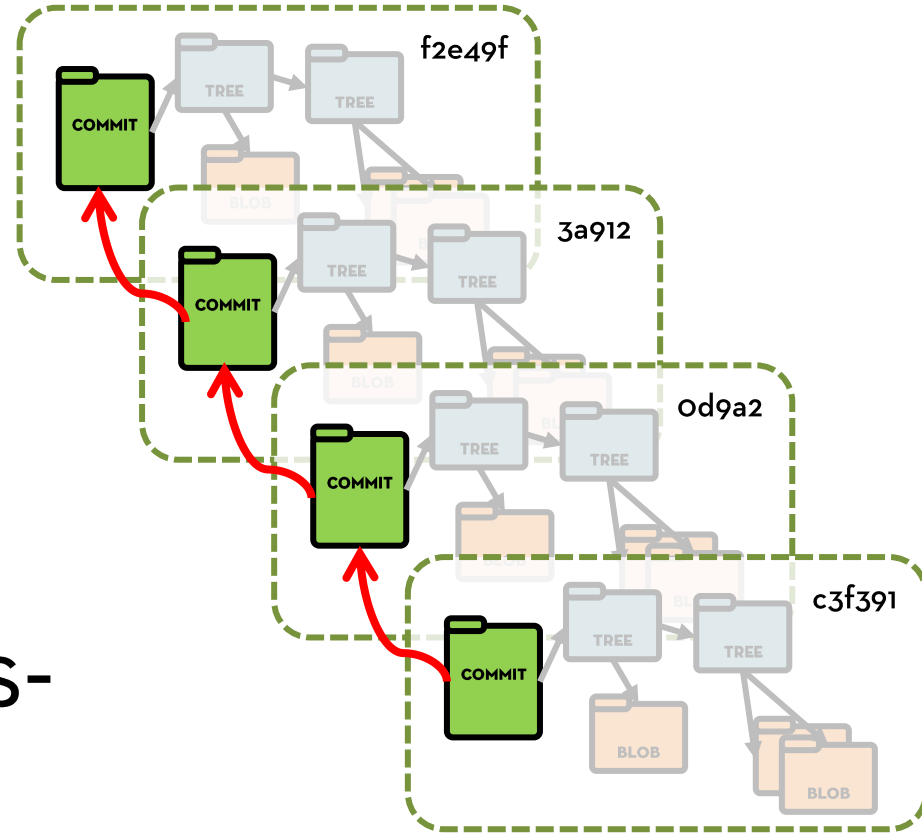
```
$ tree -a  
|-- epilog.txt  
|-- COMMIT_EDITMSG  
|-- HEAD  
|-- index  
|-- info  
|   |-- exclude  
|   |-- logs  
|   |-- HEAD  
|   |-- refs  
|   |-- heads  
|   |-- master  
-- objects  
   |-- 84  
   |   |-- f5d59a8d7961fcbeccb2765e9e3b3324f14226  
   |-- a7  
   |   |-- 3ee3e9cd12aa446818e85b7edb67e9e0ad628e  
   |-- d8  
   |   |-- 412d2d92454b1086af181f3f7fd88cfcb5932b  
   |-- info  
   |-- pack  
-- refs  
   |-- heads  
   |   |-- master  
   |-- tags
```

```
16 directories, 13 files  
$
```





Versions- Historie



Commits, Dateien, Ordner

1. Alles sind „Git Objekte“:

Einzelne Datei im .git-Folder

SHA1 des Inhalts als Name

2. Alles ist ein Baum

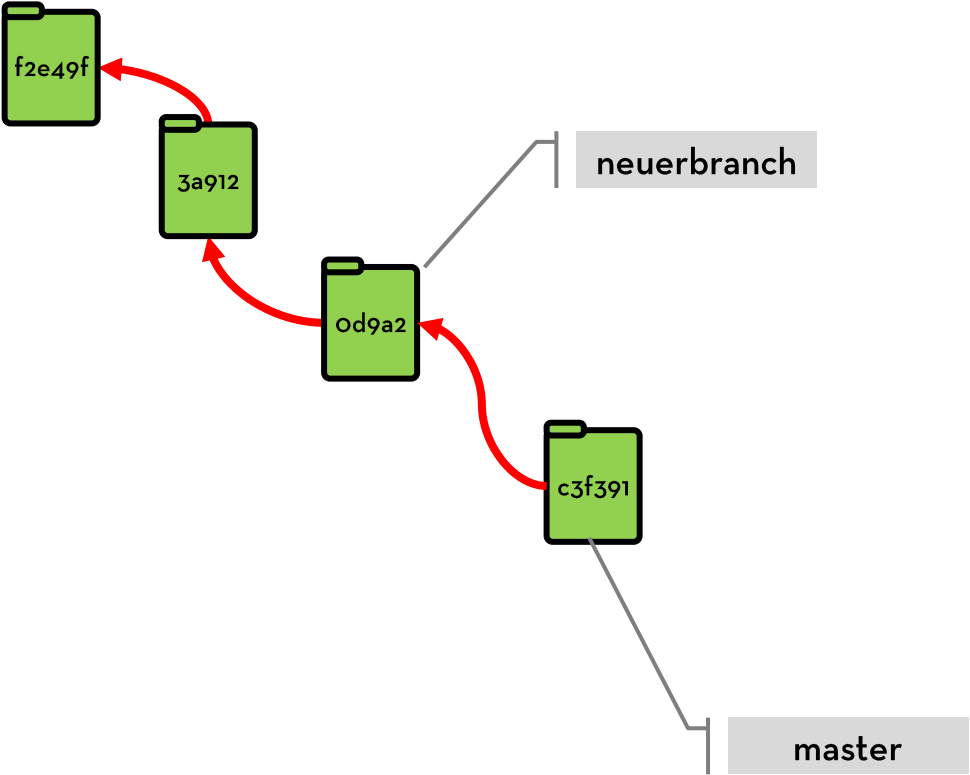
Repository = Baum sich
referenzierender Objekte

84f5d59a8d7961fcbeccb
2765e9e3b3324f14226

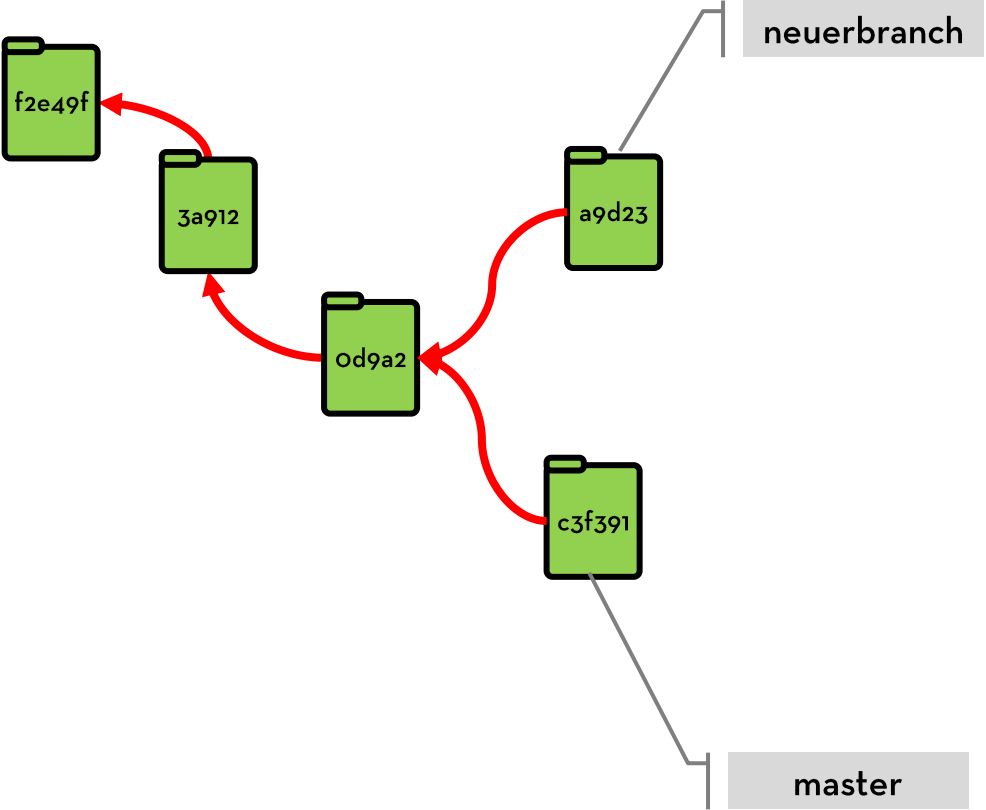


Branches

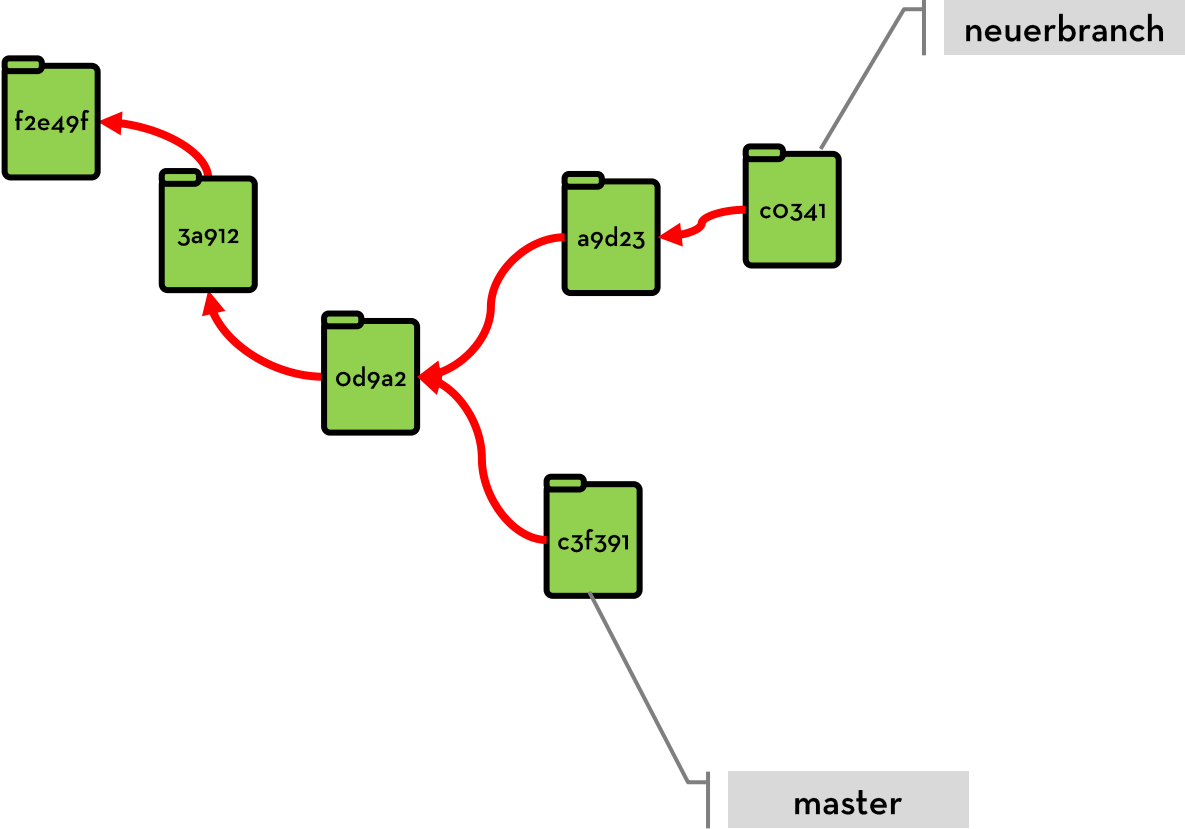
Branches & Commits



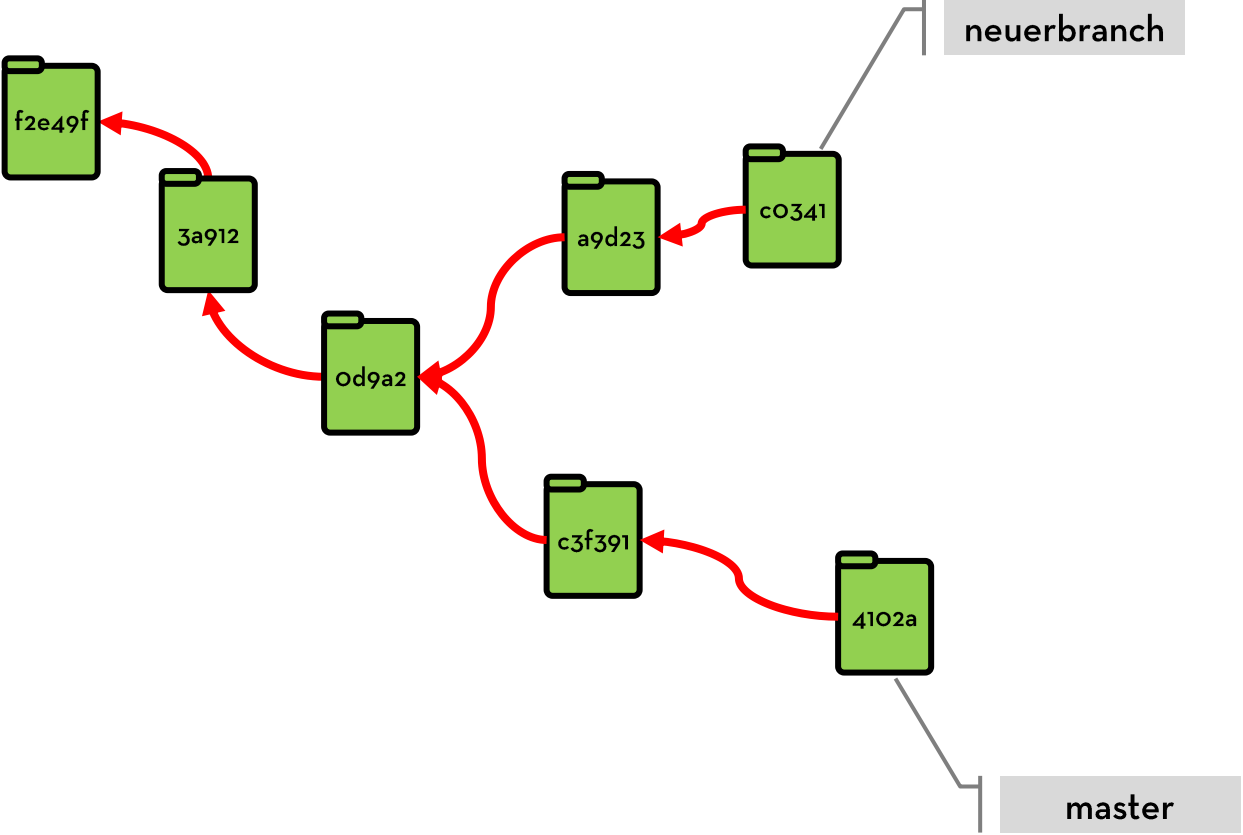
Branches & Commits



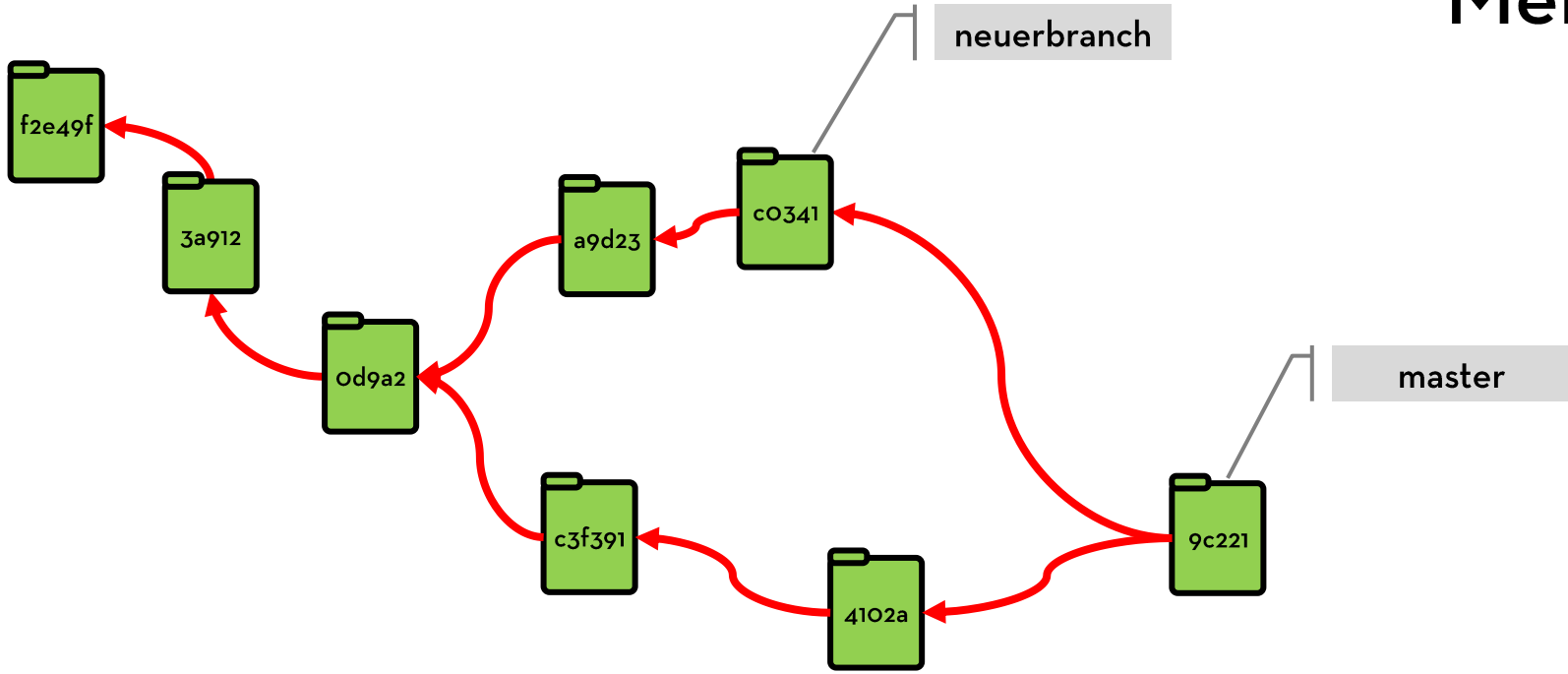
Branches & Commits



Branches & Commits

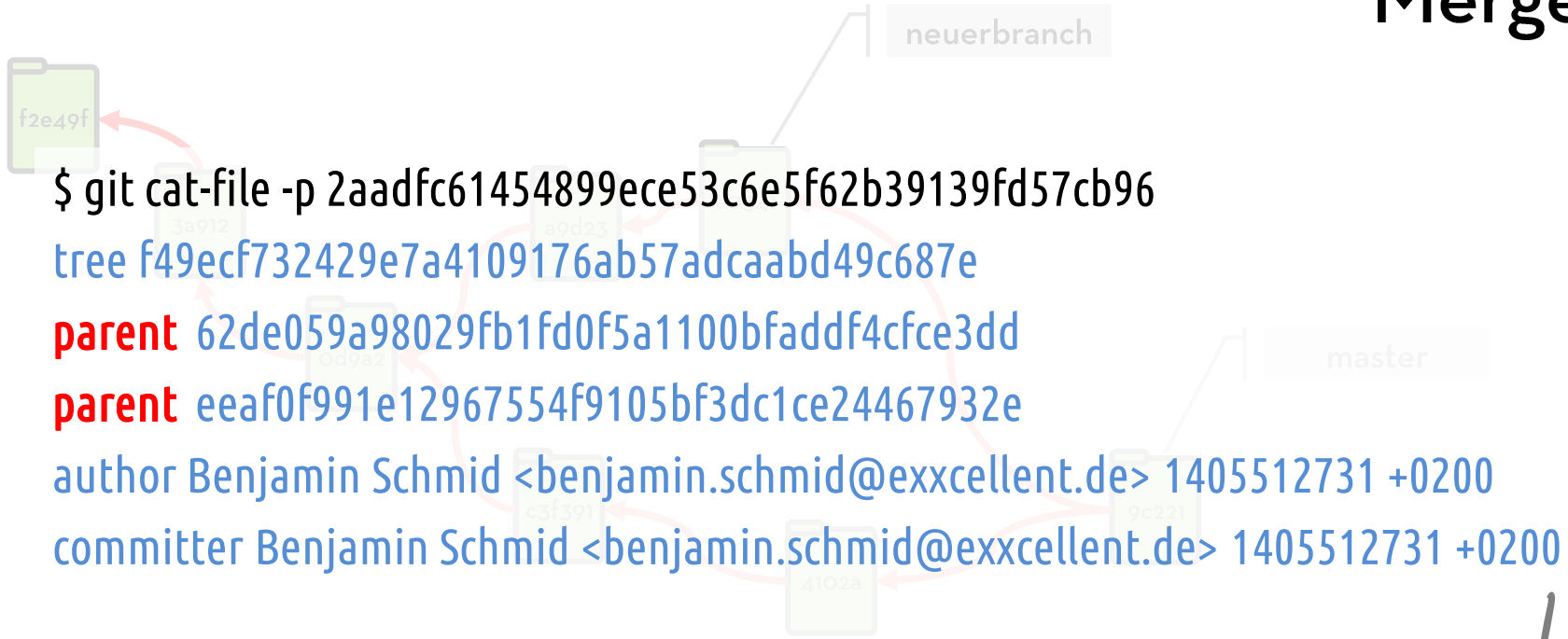


Merge



2 Parents!

Merge

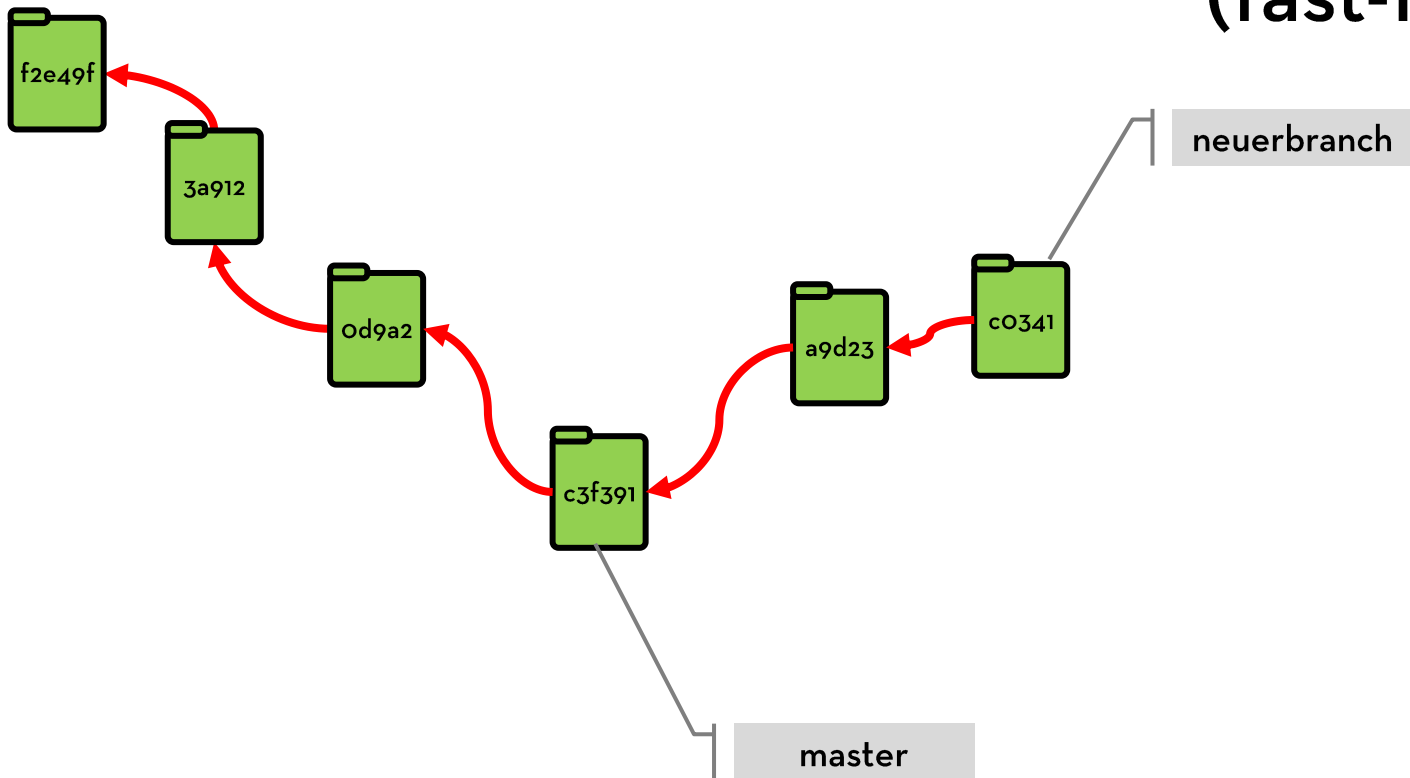


```
$ git cat-file -p 2aadfc61454899ece53c6e5f62b39139fd57cb96
tree f49ecf732429e7a4109176ab57adcaabd49c687e
parent 62de059a98029fb1fd0f5a1100bfaddf4cfce3dd
parent eeaf0f991e12967554f9105bf3dc1ce24467932e
author Benjamin Schmid <benjamin.schmid@excellent.de> 1405512731 +0200
committer Benjamin Schmid <benjamin.schmid@excellent.de> 1405512731 +0200
```

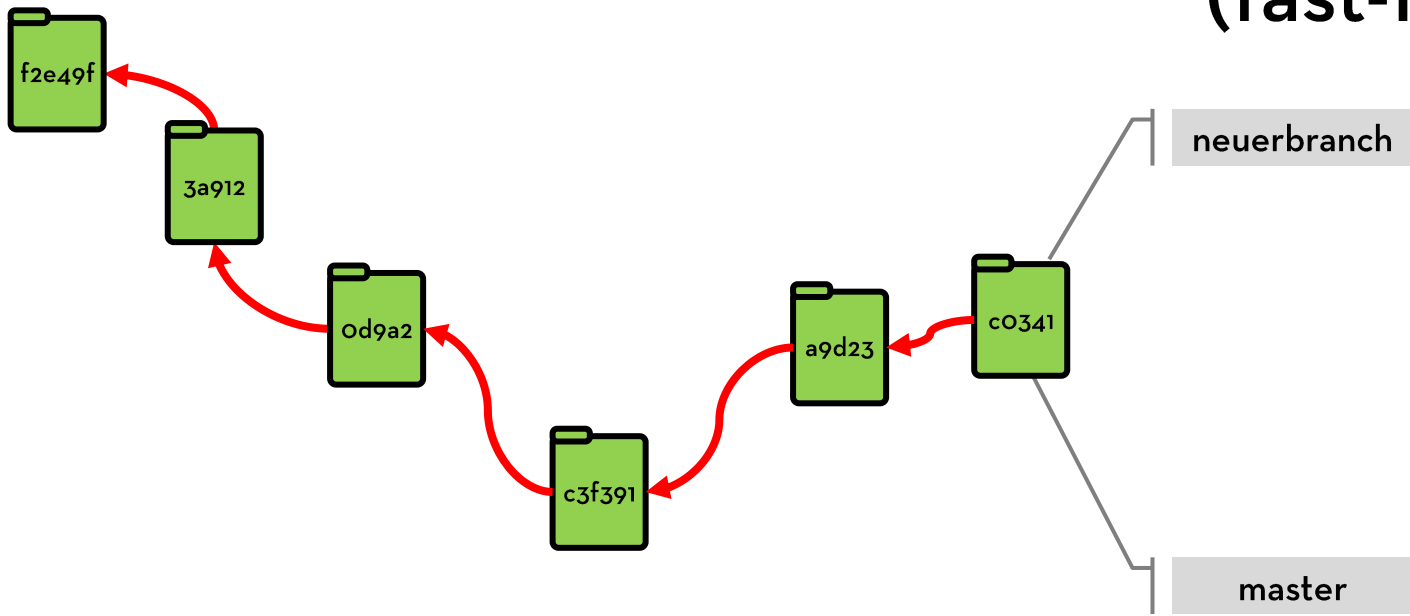
Merge branch 'meinbranch'

2 Parents!

Merge (fast-forward)



Merge (fast-forward)



Commitish in a nutshell

Eine Git-Version (Hash) umfasst stets

- ◆ alle Inhalte
- ◆ alle Datei(-ordner)
- ◆ alle Commit-Nachrichten

des **Commits** und **aller Vorgänger**

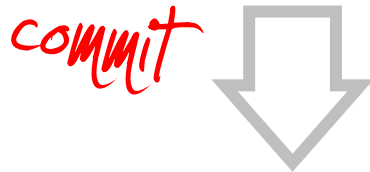
*Nagelprobe:
git fsck*

WORKSPACE



STAGING AREA

(auch: „Index“ oder „Cache“)



COMMIT

git

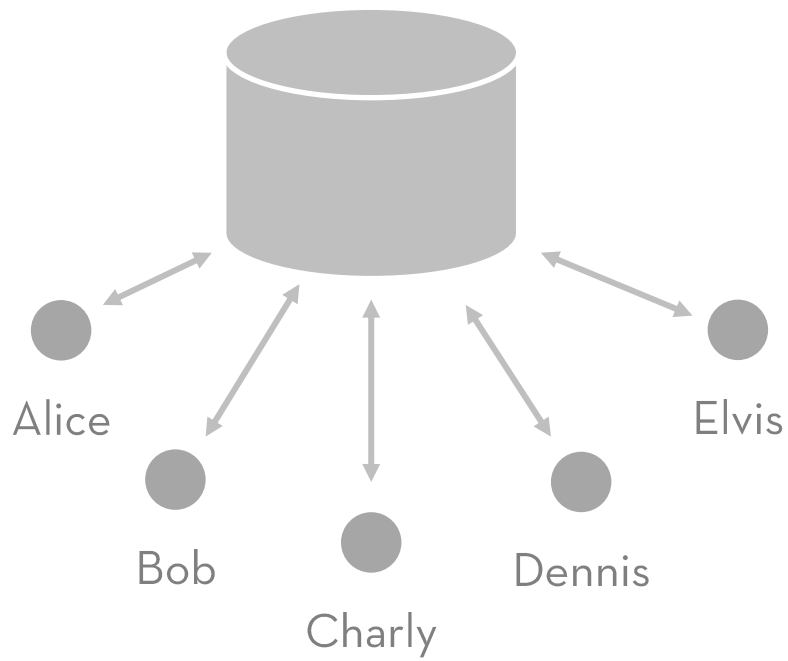
GIT IM TEAM

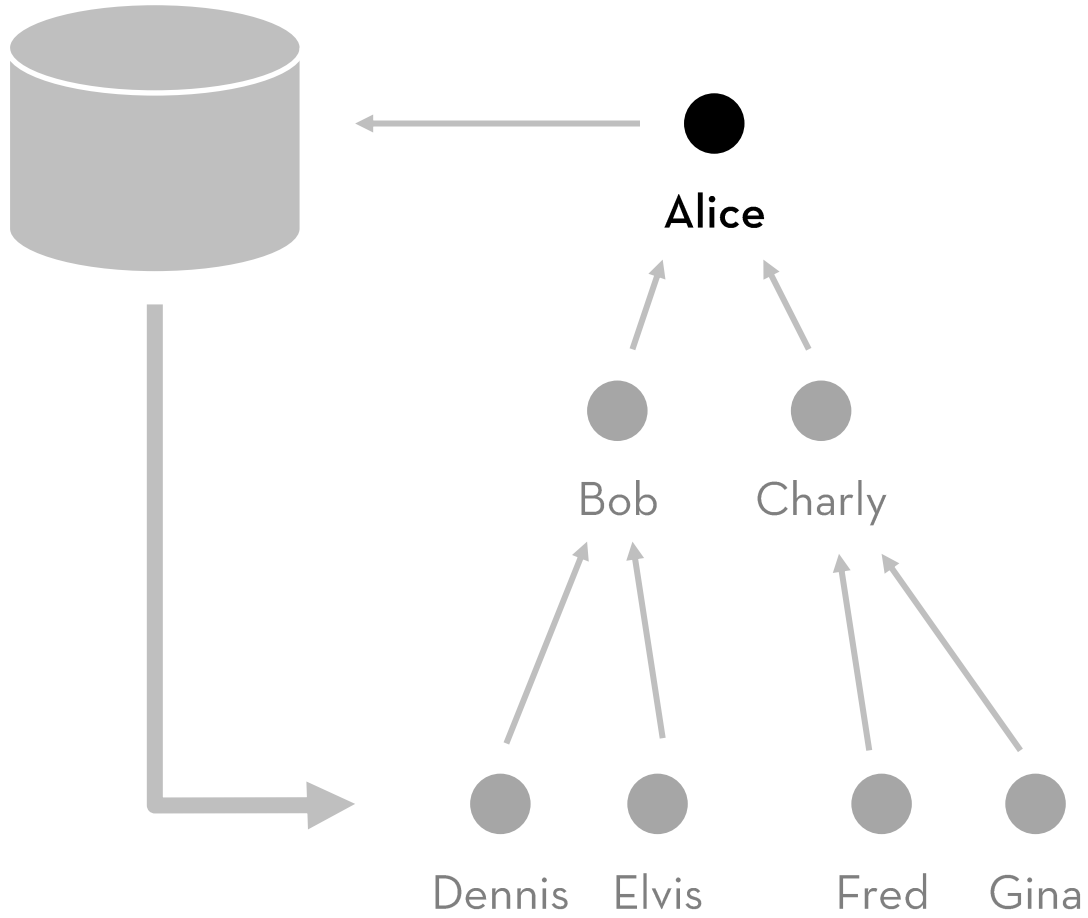


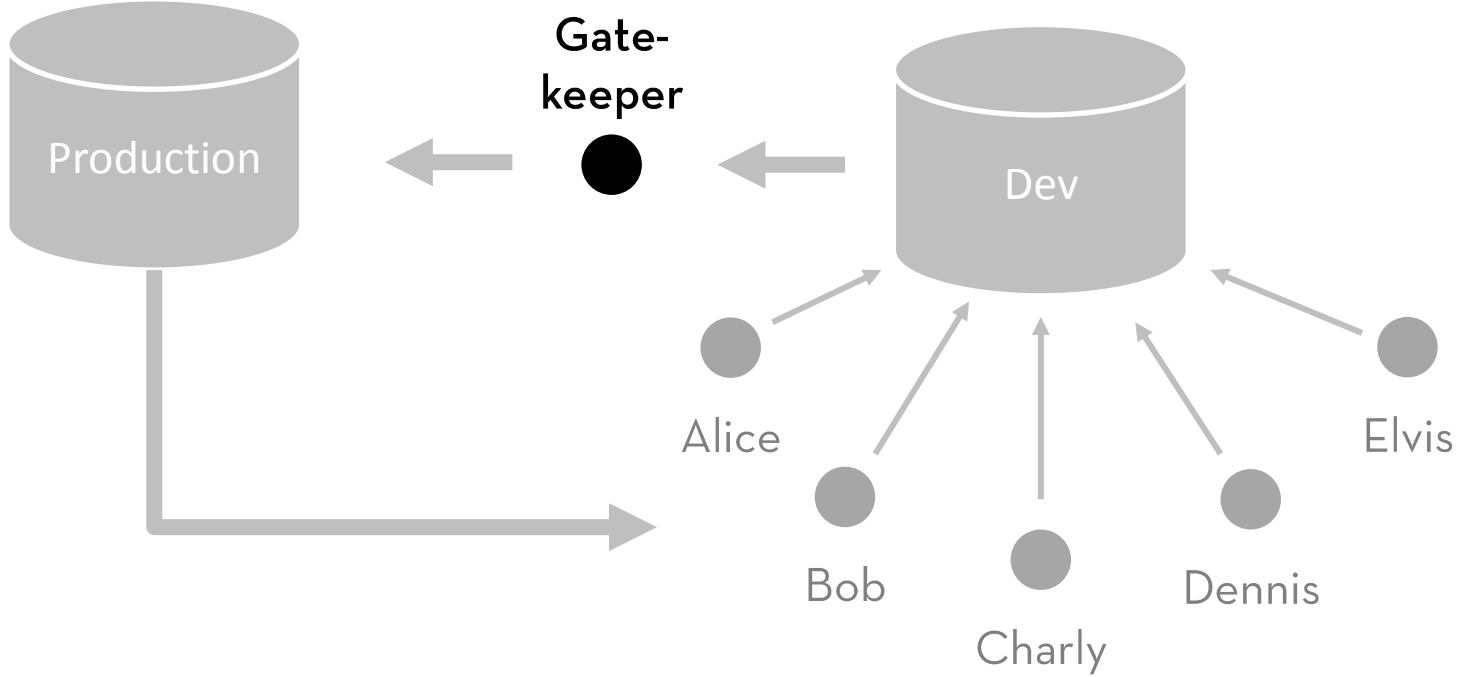
DATEN-AUSTAUSCH



N:M







INDIVIDUELLE PROZESSE

Branch

- ◆ Feature-basiert
- ◆ Trunk-basiert
- ◆ Basisprodukt +
Varianten
- ◆ ...

Repository

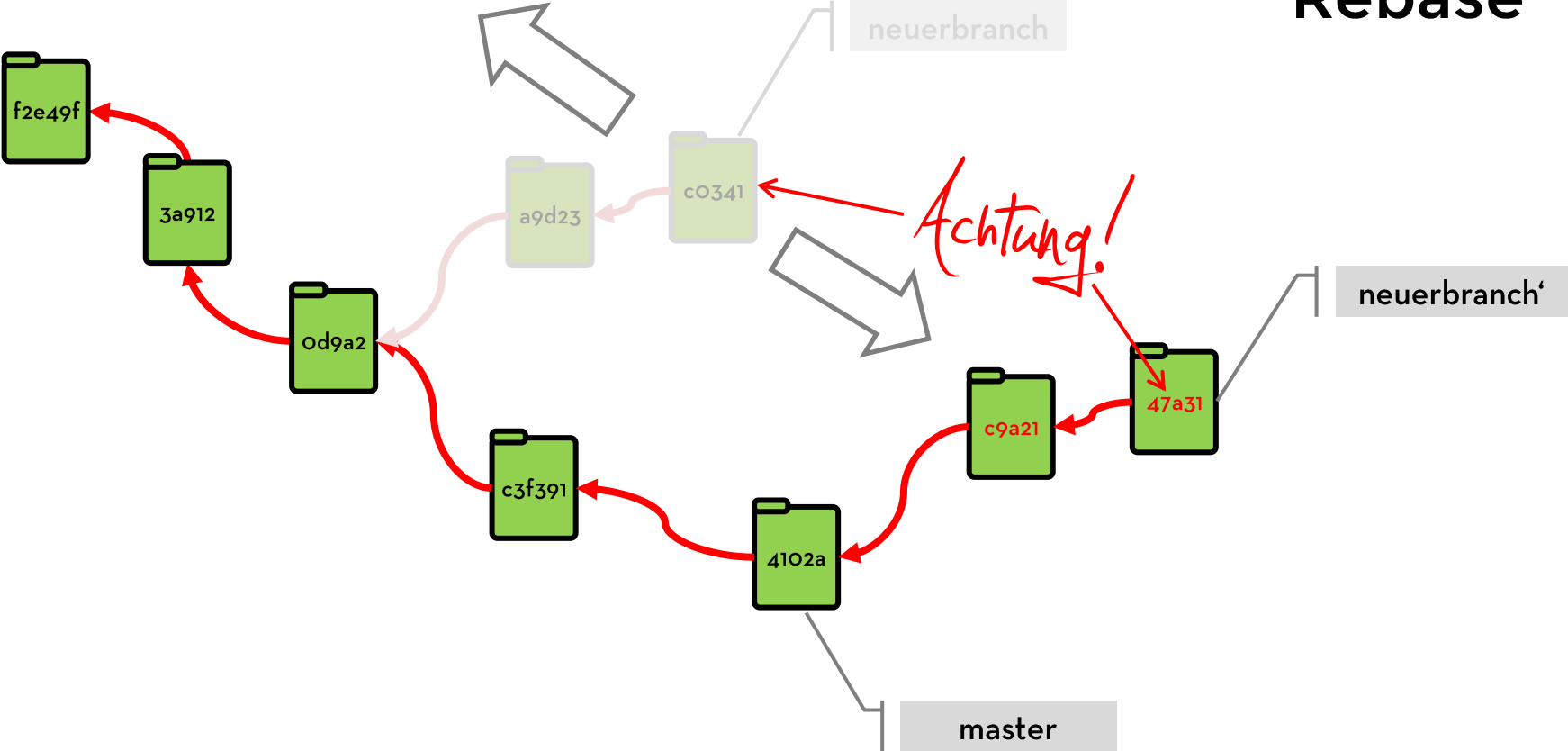
- ◆ Zentral
- ◆ Hierarchisch
- ◆ Gatekeeper
- ◆ ...

511

ADVANCED FEATURES



Rebase



interactive rebase

Commits

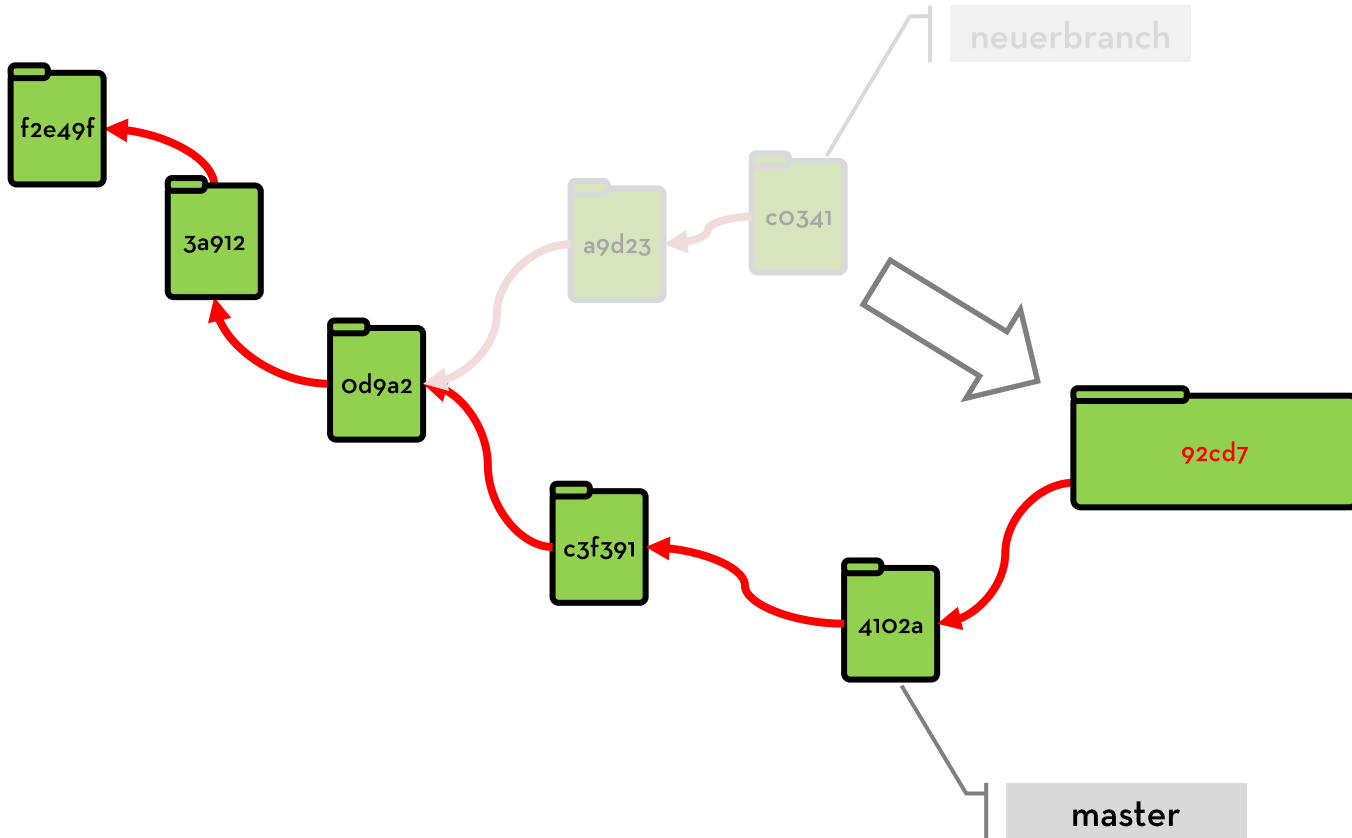
... zusammenfassen

... weglassen

... umsortieren

... ändern

Interactive Rebase



Commits
zusammenfassen
weglassen
umsortieren
ändern

bisect

Welcher Commit ist
Auslöser des Bugs?

git-svn | Arbeiten gegen Subversion-Repo

<http://subgit.com/> Kommerzielle, serverbasierte Alternative
mit weniger Einschränkungen (git/svn Mirroring)

Best

BEST PRACTICES



commit often
commit early

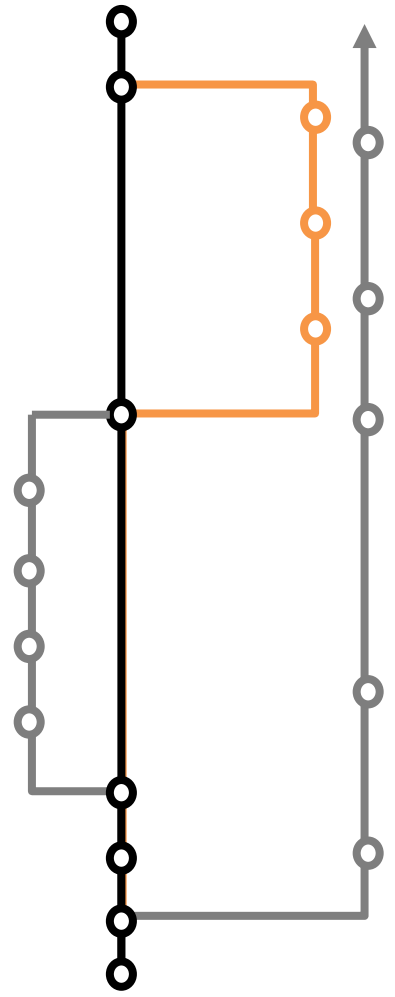
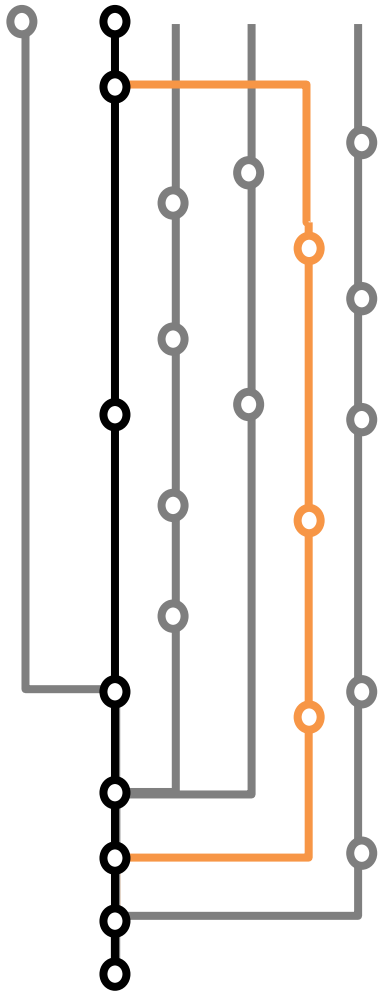
Kleine Änderungen
Isolierte Änderungen
Intensives Branching
Cleanup am Ende

Klares Branching-Modell

Angemessen
Verständlich
Dokumentiert

Lineare Historie
anstreben

Kurzlaufende Branches
Rebasing



Historie
intakt halten | Keine Änderungen
(z.B. Rebase) an
veröffentlichten Commits

Ordnung halten

Cherry-Picks
dokumentieren
Branch-Inhalte
Branch-Naming
Commit-Kommentare

811

FALLSTRICKE



Windows

Vieles besser geworden
dank ~~msysgit~~, GitHub & Co.

Windows

Lange Dateipfade

CRLF vs. LF

Performance

File Execution Bits

Umlaute

Σ = Plattform 2.te Klasse

```
git push --force
```

Falscher Workflow

BIT

CLIENTS



CLI (Shell)

```
fish /home/ben/projects/echo3.git
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
→ echo3.git git:(master) X git status
Auf Branch master
Ihr Branch ist auf dem selben Stand wie 'official/master'.

Änderungen, die nicht zum Commit vorgemerkt sind:
(benutzen Sie "git add <Datei>..." um die Änderungen zum Commit vorzumerken)
(benutzen Sie "git checkout -- <Datei>..." um die Änderungen im Arbeitsverzeichnis zu verwerfen)

    geändert:   .idea/runConfigurations/HelloWorld__JavaScript_version_.xml
    geändert:   .idea/runConfigurations/Interactive_Testapp__JavaScript_version_.xml
    geändert:   .idea/runConfigurations/NumberGuess__JavaScript_version_.xml
    geändert:   .idea/runConfigurations/TestApp__Java_.xml

Unbeobachtete Dateien:
(benutzen Sie "git add <Datei>..." um die Änderungen zum Commit vorzumerken)

    atlassian-ide-plugin.xml
    projectFilesBackup/

keine Änderungen zum Commit vorgemerkt (benutzen Sie "git add" und/oder "git commit -a")
→ echo3.git git:(master) X git log @~4... --oneline
5b1756a Merge tag 'v3.0.2' for easier merging.
5af0e5a Merge pull request #78 from ceari/feature/visual-hint-for-modal-windows
a97efc9 WindowPane: Fix NPE in modal overlay handling when component is unregistered
d9bf957 Merge branch 'hotfix/invalid-colors' of github.com:ceari/echo3
4d1e873 Fix color handling in Sync.Color.renderClear()
bc76577 Merge branch 'feature/invalid-response-handling' of github.com:ceari/echo3
f31c2b9 Merge branch 'feature/invalid-response-handling' of github.com:ceari/echo3 into stable-3.0.x
2193a58 Configurable handling of invalid Ajax responses
7a22a33 Merge branch 'feature/outdated-browser-noscript-hints' of github.com:ceari/echo3 into stable-3.0.x
28d1606 Release Echo v3.0.2
→ echo3.git git:(master) X
```

GitHub for Windows/Mac

The screenshot shows the GitHub web interface for a repository named 'octokit/octokit.net'. The page displays a merge pull request #253 from the branch 'feature/documentation' to the 'master' branch, submitted by user 'Brendan Forster' (commit 60def6e). The pull request title is 'Merge pull request #253 from pmacn/feature/documentation'. Below the title, the file 'Documentation' is expanded to show the file 'Octokit\Clients\ActivitiesClient.cs'. A diff view shows the changes to this file, with a green highlight on the new summary and parameter documentation. The left sidebar shows a list of recent activity, including merge pull requests and updates by other users like Phil Haack and Peter MacNaughton.

no uncommitted changes

octokit/octokit.net

sync master

Merge pull request #253 from pmacn/feature/documentation

Brendan Forster 60def6e expand all github revert roll back

Documentation

▼ Octokit\Clients\ActivitiesClient.cs

old	new
...	... @@ -2,6 +2,10 @@
2	2 {
3	3 public class ActivitiesClient : ApiClient, IActivitiesClient
4	4 {
	5 + /// <summary>
	6 + /// Instantiate a new GitHub Activities API client.
	7 + /// </summary>
	8 + /// <param name="apiConnection">An API connection</param>
5	9 public ActivitiesClient(IApiConnection apiConnection)
6	10 : base(apiConnection)
7	11 {

▶ Octokit\Clients\AssigneesClient.cs

▶ Octokit\Clients\BlobsClient.cs

▶ Octokit\Clients\CommitsClient.cs

▶ Octokit\Clients\EventsClient.cs

history

- Phil Haack 13 hours ago Merge pull request #260 from octokit/...
- Haacked 13 hours ago Add comments to concrete types
- Phil Haack 13 hours ago Merge pull request #257 from pmacn/...
- Peter MacNaughton 3 days ago Updating CustomBuildTask.dll. Change...
- Peter MacNaughton 3 days ago Modified Octokit project to run custo...
- Brendan Forster 3 days ago Merge pull request #253 from pmacn/...**
- Brendan Forster 3 days ago Merge pull request #256 from andrero...
- Andre Rodrigues 3 days ago Add null checks in constructor
- Andre Rodrigues 3 days ago Remove null check
- Andre Rodrigues 3 days ago Update regex expression

Source Tree (Win/Mac)

The screenshot displays the SourceTree application interface. The top menu bar includes File, Edit, View, Repository, Actions, Tools, and Help. Below the menu is a toolbar with icons for Clone/New, Commit, Checkout, Discard, Stash, Add, Remove, Add/Remove, Fetch, Pull, Push, Branch, Merge, Tag, and Terminal. The main workspace is divided into several sections:

- File Status:** Shows Working Copy.
- Branches:** Lists the current branch (master).
- Remotes:** Shows the origin remote.
- Stashes:** Shows no stashes.
- Commit History Table:** A table with columns for Graph, Description, Date, Author, and Commit. The current commit is highlighted in blue.
- Commit Details:** Shows the commit hash, parent hashes, author, date, and labels for the selected commit.
- Diff View:** Shows a diff for the file RestSharp/Deserializers/XMLAttributeDeserializer.cs, highlighting changes in green and red.

Graph	Description	Date	Author	Commit
	Merge pull request #259 from Dharun/patch-1	11 Jan 2013 19:18	Andrew Young <a>	cc283fb
	Update CONTRIBUTING.markdown	18 Sep 2012 0:26	John Sheehan <jol	054adce
	Merge pull request #324 from PedroLamas/master	17 Sep 2012 18:18	Pedro Lamas <pec	a529617
	Added CONTRIBUTING.markdown	17 Sep 2012 18:15	Pedro Lamas <pec	cfce270
	Merge pull request #321 from apodlaski/master	13 Sep 2012 5:56	Andrew Young <a>	eef0c86
	Another fix for ExecuteAsync	7 Sep 2012 14:16	Aleksander Podlas	2ce5204
	104.1 Version bump: 401.1	7 Sep 2012 6:39	Andrew Young <a>	196fa7e
	Merge pull request #320 from jasonmoore2k/ExecuteAsyncFix	5 Sep 2012 14:16	Pete Johanson <la	4531670

Commit: cc283fb0843e1141f717dcc6b3009206b6c95b62 [cc283fb]
Parents: 054adce80b, c68de9d856
Author: Andrew Young <andrewyoung@gmail.com>
Date: 11 January 2013 19:18:31
Labels: HEAD, origin/master, origin/HEAD, master

Merge pull request #259 from Dharun/patch-1

Filename	Path
XmlAttributeDeserializer.cs	RestSharp/Deserializers
.gitignore	
NuGet.Config	.nuget
NuGet.Exe	.nuget

```
RestSharp/Deserializers/XMLAttributeDeserializer.cs
Modified file, 1 lines added, 1 lines removed
Reverse File

Hunk 1 : Lines 109-115
109 109
110 110
111 111
112 -
112 +
113 113
114 114
115 115

var value = GetValueFromXml(root, name, is
}
if (value == null)
if (value == null || value == string.Empty)
{
// special case for inline list it
if (Type.IsGenericType)
```

Clean | master Atlassian

Weitere gute Clients

◆ IDE: IntelliJ

Nahtlos

◆ TortoiseGit

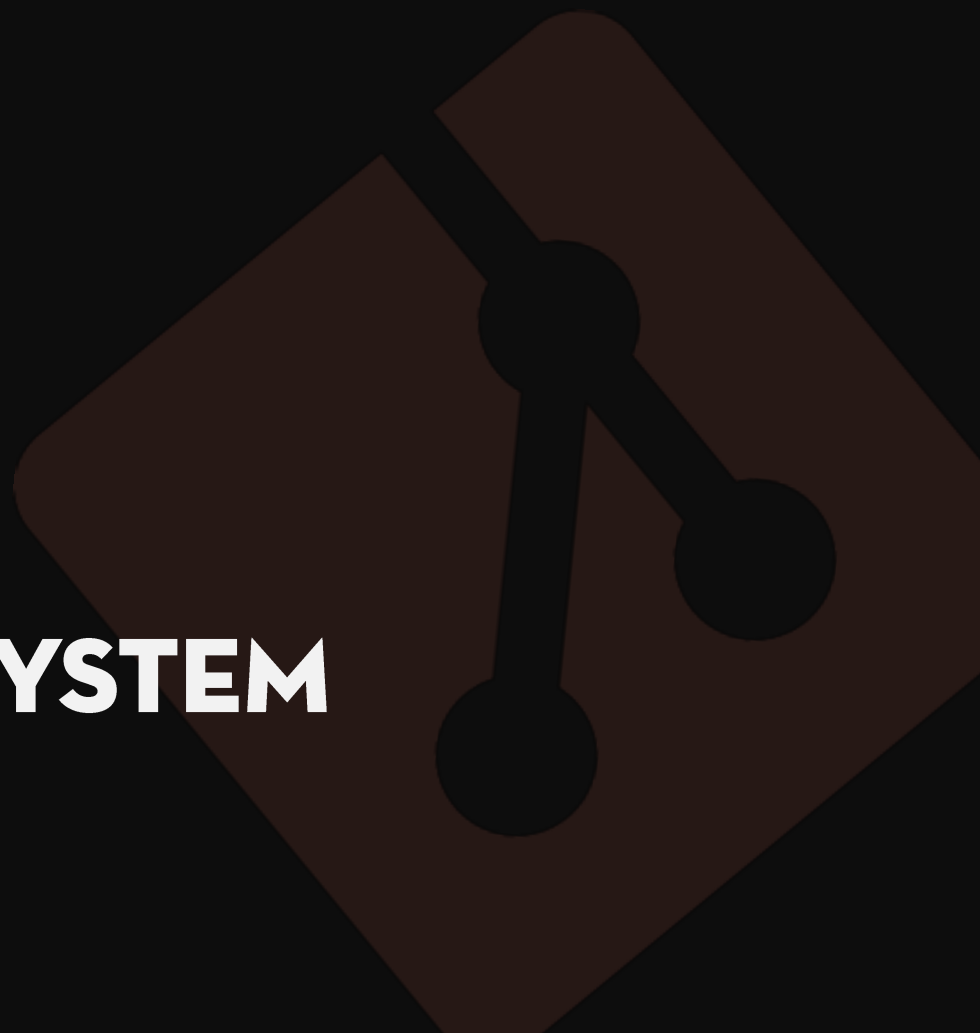
Explorer-Integration

◆ Tower (Mac, 49€)

Sehr leistungsfähig

BIT

TOOLS & ÖKOSYSTEM



GitHub Clones

„Pull Requests“
Issue Tracking
Code Browser & Review
Wiki
Security & Permissions

<https://github.com/takezoe/gitbucket>

<http://gitblit.com/>

<http://gitlabhq.com/>

Scala-basierter Clone

Java-basiert

Populär. Etwas komplexer Betrieb

Code Review Tools

<http://code.google.com/p/gerrit/>
<http://phabricator.org/>

Git-basiertes Code-Review System
Code Review & Repo-Browser

Continuous Integration Continuous Delivery

CI als Gatekeeper
Tests vor Code-
Integration

<https://travis-ci.org/>
<http://jenkins-ci.org/>

Hosted CI. Frei für FOSS-Projekte
Populärer, FOSS CI Server



file | 15 lines (12 sloc) | 0.614 kb

Edit

Raw

Blame

History

Delete

```
1 language: java
2
3 # Snapshot deployment to Sonatype Maven repository
4 # The password is encrypted using "travis encrypt -r echo3/echo3 SONATYPE_PASSWORD=password"
5 # and only decrypted and set as environment variable by travis-ci if the build runs in the context of
6 # the echo3/echo3 github repository.
7
8 env:
9   global:
10    - SONATYPE_USERNAME=bschmid
11    - secure: "UTEI77pRoUliei20VAnOTJ7C0SuR234w3lbdR3NqBxvItyMJBHpv+/1nxNLhtvbx0L7DX0vdqqUnAn2vFPXmUtJFUq+G038FjNKhHHVki9YKN9+"
12
13 after_success:
14 - python resource/travis-ci/addServer.py
15 - ant travis.mvn.snapshot
```

[My Repositories](#)
[Recent](#)

[echo3/echo3](#) 134

4 min 7 sec

Finished: a day ago

[excellent/echolot](#) 15

48 sec

Finished: 2 months ago

[excellent/echo3](#) 15

1 min 59 sec

Finished: 3 months ago

[echo3/echo3filetransfer](#) 17

54 sec

Finished: 3 months ago

[echo3/echo3extras](#) 38

52 sec

Finished: 4 months ago

[bschmid/echo3](#) 5

2 min 22 sec

Finished: 6 months ago

[excellent/echo3filetransfer](#) 3

31 sec

Finished: 8 months ago

[excellent/echo3extras](#) 4

echo3/echo3

build passing

Echo is a platform for building web-based applications that approach the capabilities of rich clients. The applications are developed using a component-oriented and event-driven API, eliminating the need to deal with the "page-based" nature of browsers. To the developer, Echo works just like a user interface toolkit. The latest iteration, Echo3, features the ability to write applications in either server-side Java or client-side JavaScript. Server-side applications run in any Java Servlet Container and do not require developer knowledge of HTML, HTTP, or JavaScript. Client-side JavaScript-based applications do not require any server at all, but can of course communicate with one via XML or JSON over HTTP. Cross-platform AJAX technology is used for rendering, and no browser plugins are required. Echo is open-source software, dual-licensed under the Mozilla Public License and LGPL.

[Current](#)
[Build History](#)
[Pull Requests](#)
[Branch Summary](#)


Build	134	Commit	7148107 (master)
State	Passed	Compare	3ad182b4219e...7148107b2a7c
Finished	a day ago	Author	Scott Knick
Duration	4 min 7 sec	Committer	Benjamin Schmid
Message	Scrub README.md from minor formatting & spelling issues		

Removed line breaks in middle of hyphenated words as this put a space in the word in the Markdown view. Made other minor grammar/capitalization changes.

Fixes [#84](#)

Signed-off-by: Benjamin Schmid <B.Schmid@excellent.de>

```

1 Using worker: worker-linux-9-2.bb.travis-ci.org:travis-linux-12
2
3 $ export SONATYPE_USERNAME=bschmid
4 $ export SONATYPE_PASSWORD=[secure]
5 $ git clone --depth=50 --branch=master git://github.com/echo3/echo3.git echo3/echo3
6 $ cd echo3/echo3
7
8 $ git checkout -qf 7148107b2a7c0018f38eb0e29c8d8591073a3d44
9 $ git submodule init
10 $ git submodule update
11 $ java -version
12 java version "1.7.0_45"
13 Java(TM) SE Runtime Environment (build 1.7.0_45-b18)
14 Java HotSpot(TM) 64-Bit Server VM (build 24.45-b08, mixed mode)
15 $ javac -version
16 javac 1.7.0_45
    
```

FAZIT

FAZIT



◆ Innovatives & lebendiges Ökosystem & Tools!

◆ „Distributed“

◆ Arbeiten mit leichtgewichtigen Branches

◆ ‚Einfache‘ & Effiziente Konzepte

◆ Große Freiheitsgrade (Workflow, Undo, ...)

- ◆ Teils **steile Lernkurve**
- ◆ Mächtige Mittel zur Manipulation der Historie
- ◆ Keine Teil-Checkout möglich
- ◆ Kein Rename-Tracking
- ◆ Kanten unter Windows

BIT

**WEITERFÜHRENDE
INFORMATIONEN**



LERNEN & DOKUMENTATION

„Git - Grundlagen & Workflows“ – Git in einfachen „Kochrezepten“

„Pro Git“ – Freies Online-Buch mit deutscher Übersetzung

<http://git-scm.com/book/de>

„Try Git“ – Interaktives Tutorial im Browser zum Einstieg

<http://try.github.io/>

„Learn Git Branching“ – Sehr gutes, Interaktives Tutorial welches Rebasing & Co. vermittelt

<http://pcottle.github.io/learnGitBranching/>

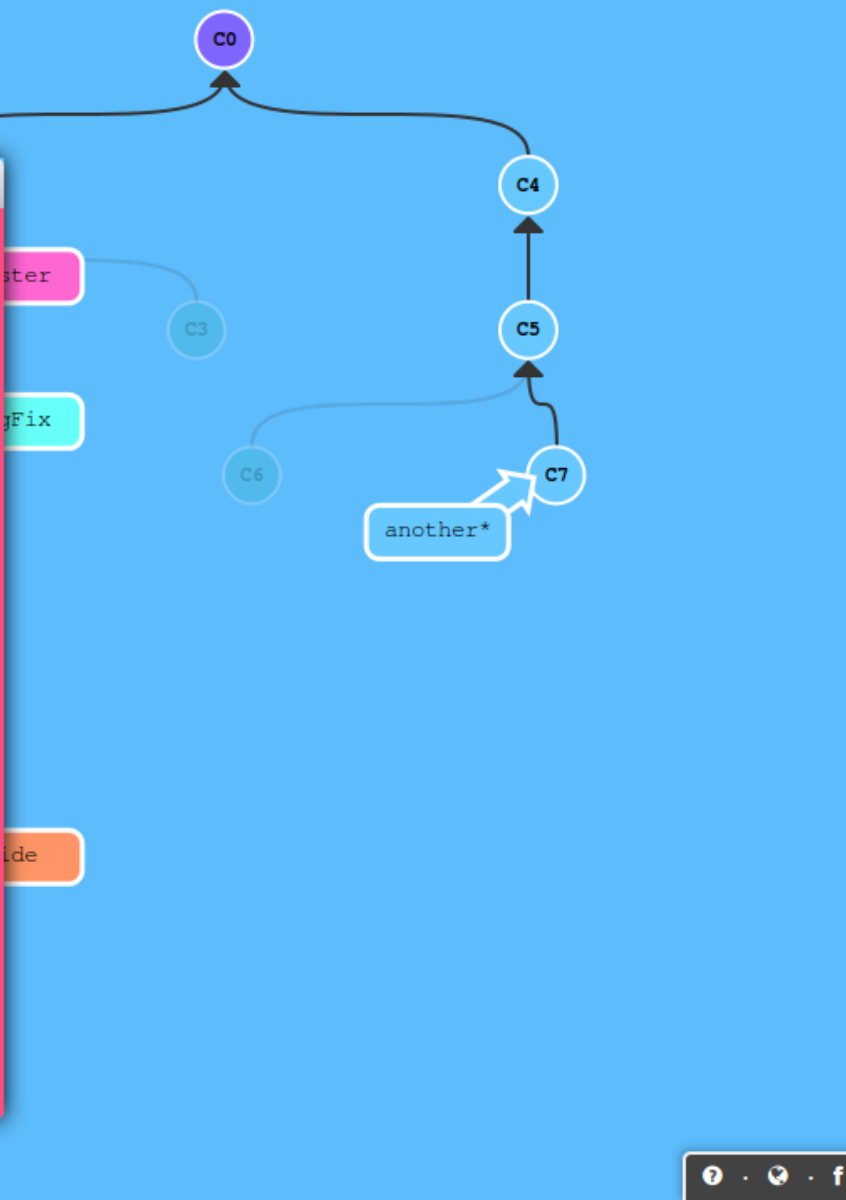
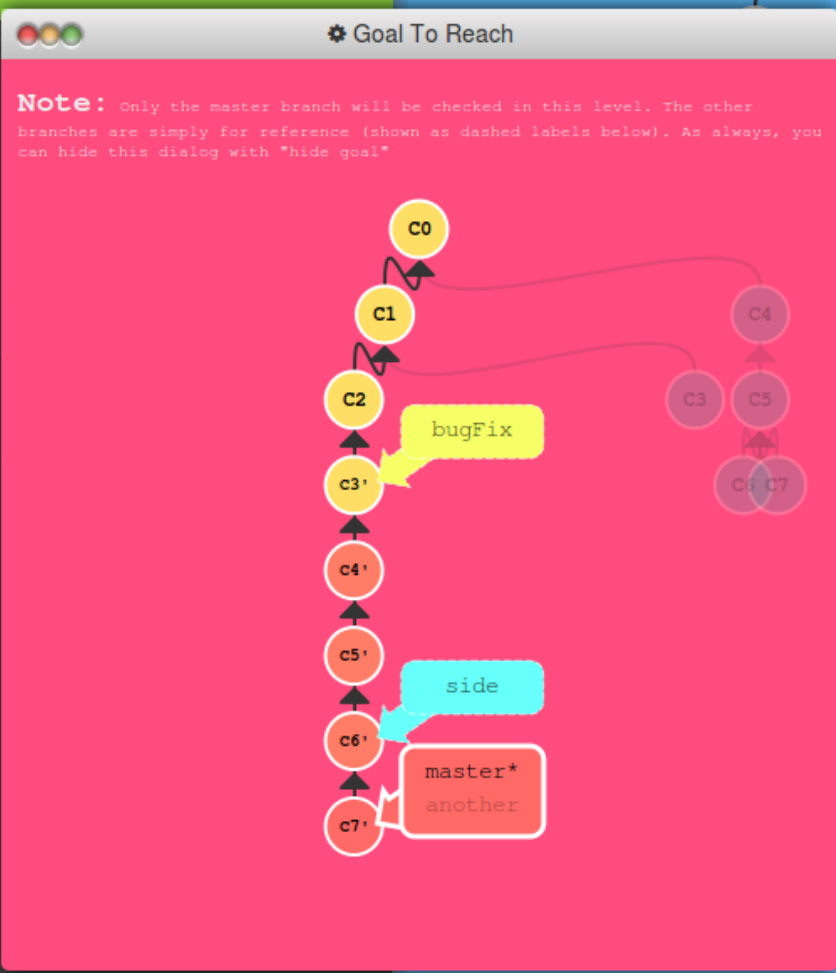


Empfehlung!

```

$ level advanced
--noStartCommand

$ delay 2000
$ show goal
$ delay 1000
$ hide goal
$ git checkout master
$ git rebase master
$ git checkout bugFix
$ git rebase bugFix
$ git checkout master
$ git rebase master
$ help
$ levels
    
```



WORKFLOW & BEST PRACTICES

„Simple Git workflow is simple“

<https://blogs.atlassian.com/2014/01/simple-git-workflow-simple/> &
<https://gist.github.com/jbenet/ee6c9ac48068889b0912>

agile Projekte

„Git Flow“ – Relativ populäres, aber auch komplexes Modell

<http://nvie.com/posts/a-successful-git-branching-model/>

*Produkt-É
Variantenpflege*

„Commit Often, Perfect Later, Publish Once: Git Best Practices“

<http://sethrobertson.github.io/GitBestPractices/>

FRAGEN & ANTWORTEN

