

Java User Group München

Subversion, Mercurial & Co. -
Zentrale und verteilte Versionsverwaltungen
im Vergleich

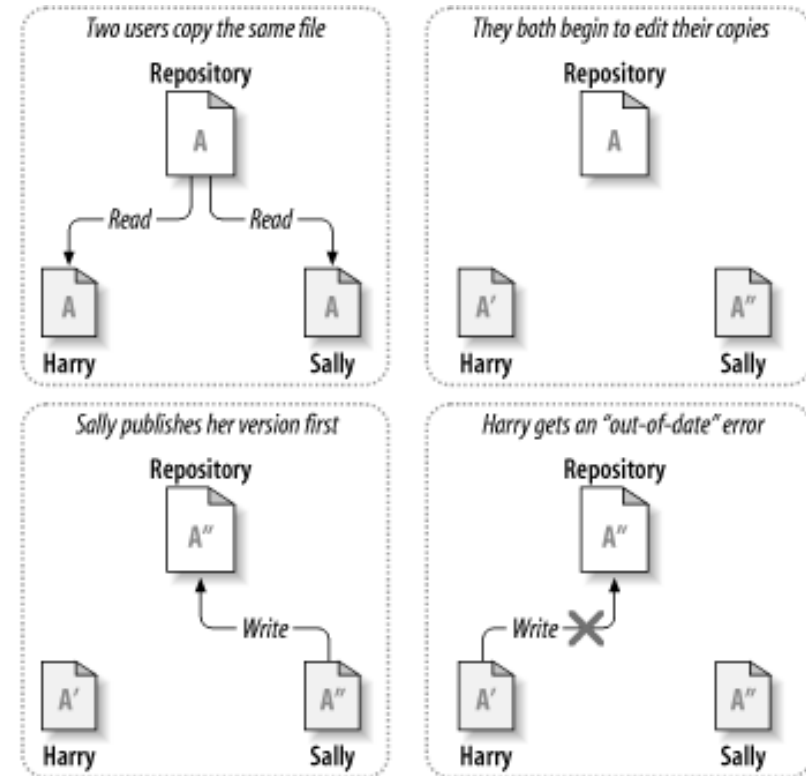
Christine Koppelt
Stefan Seelmann

- Zentral vs. Verteilt
- Funktionen von Subversion, Mercurial, Git
- Tools und Entwicklungswerkzeuge
- Fazit

Zentral vs. Verteilt

Aufgaben einer Versionsverwaltung

- Koordinierung gleichzeitiger Änderungen an Dateien (Konfliktbehandlung)
- Nachvollziehbarkeit von Änderungen
- Transaktionale Änderungen
- Kennzeichnung und Wiederherstellung von Versionen
- Zugriffskontrolle

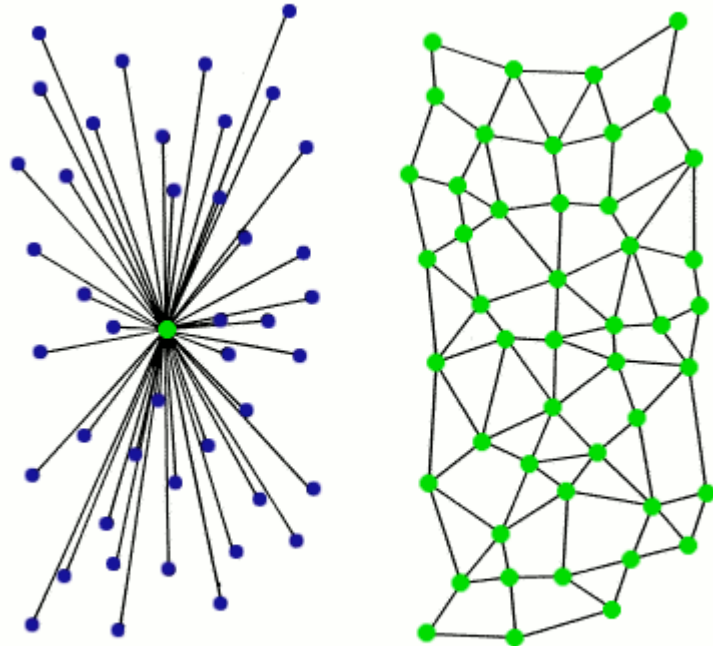


Quelle: <http://svnbook.red-bean.com/en/1.5/svn-book.html>

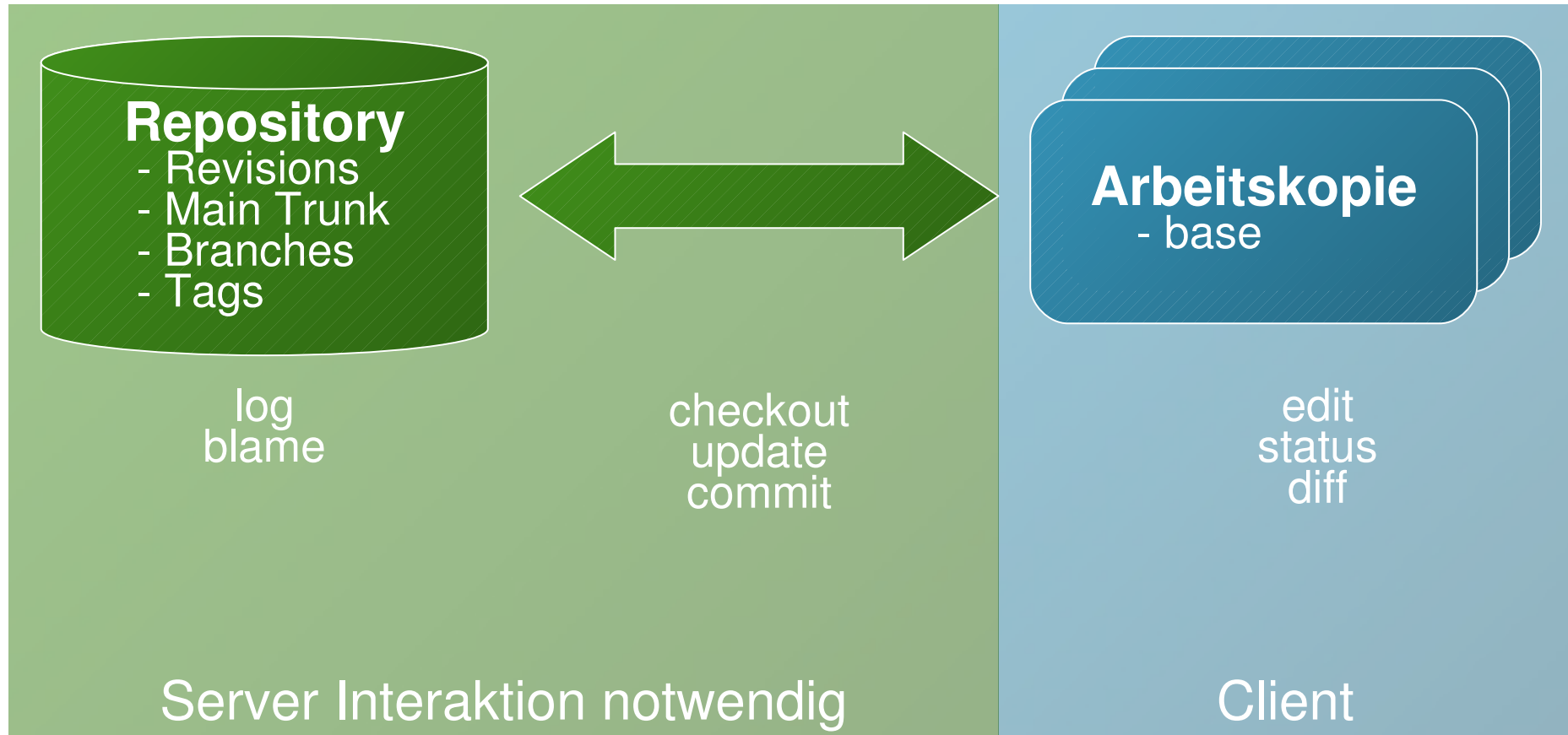
Gemeinsamkeiten zentral - verteilt

- Konzept von Repository + Arbeitskopie
- Konzept von Checkout + Checkin
- Konzept von Trunk + Branches + Tags
- Gemeinsamer Zugriff
 - Konfliktbehandlung mittels Copy – Modify – Merge Strategie
- Zugriffskontrolle (lesen / schreiben)
- Protokollierung von Änderungen / Versionierung
 - Wiederherstellung eines älteren Zustandes
 - Navigation in der Historie

Unterschied



Topologie und Arbeitsweise zentral



Zusammenfassung zentral

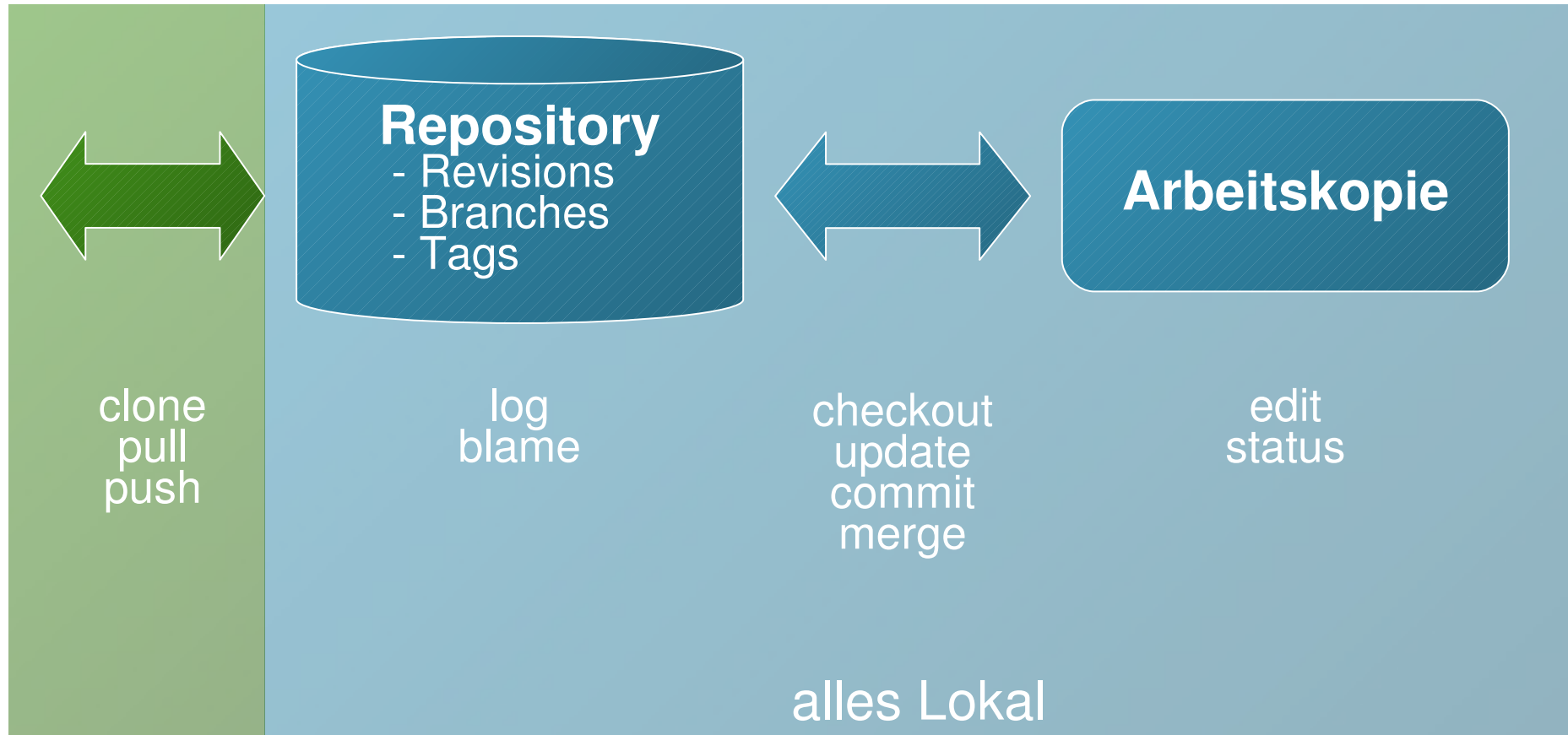
- Pros

- Fortlaufende Nummerierung der Revisionen
 - Aktuelle Version klar erkennbar
 - Backups
- Synchronisationspunkt für Entwickler
 - Weniger merges
 - Es entsteht kein Wildwuchs
- Etablierte Entwicklungsprozesse, Einfache Konzepte

- Cons

- Serverzugriff für die meisten Operationen notwendig
 - Single Point of Failure
 - Geschwindigkeit
 - Kein Offline Arbeiten
- Gewichtige Angelegenheit
- Sandboxes entweder zentral oder nicht versioniert

Topologie und Arbeitsweise verteilt



Zusammenfassung verteilt

- Pros

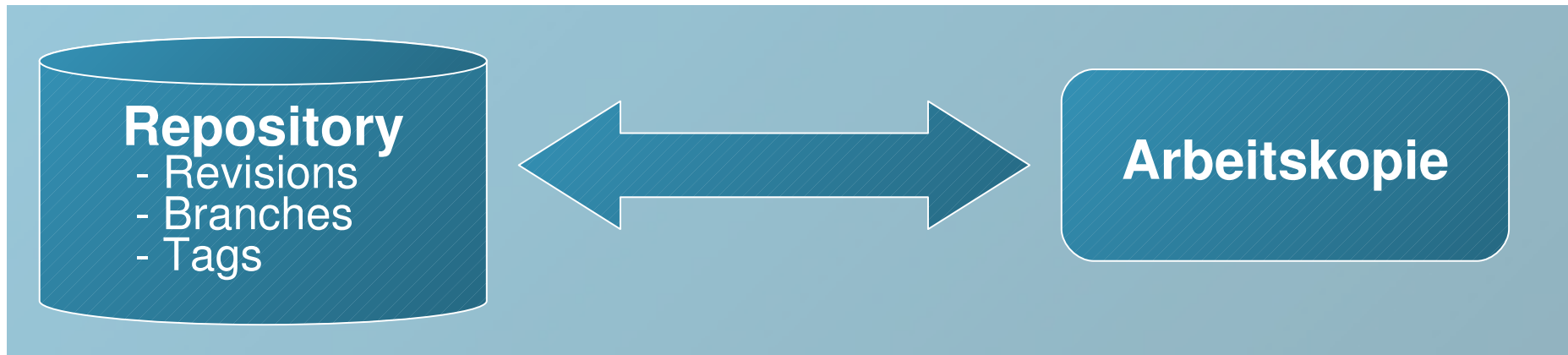
- Gesamte Historie lokal verfügbar
 - Lokales Repository ist jedoch nicht unbedingt größer
- Ermöglicht offline arbeiten, kein Netzzugriff notwendig
 - Server-, Netzausfall, beim Kunden, auf Reisen, Homeoffice
- Höhere Performance, da Repository lokal liegt
- Keine „Störung“ durch andere Entwickler

- Cons

- Kein führendes System
- Zusätzlicher Schritt für Synchronisation notwendig
- Evtl. mehr mergen notwendig
- Unabhängige Branches / Forks

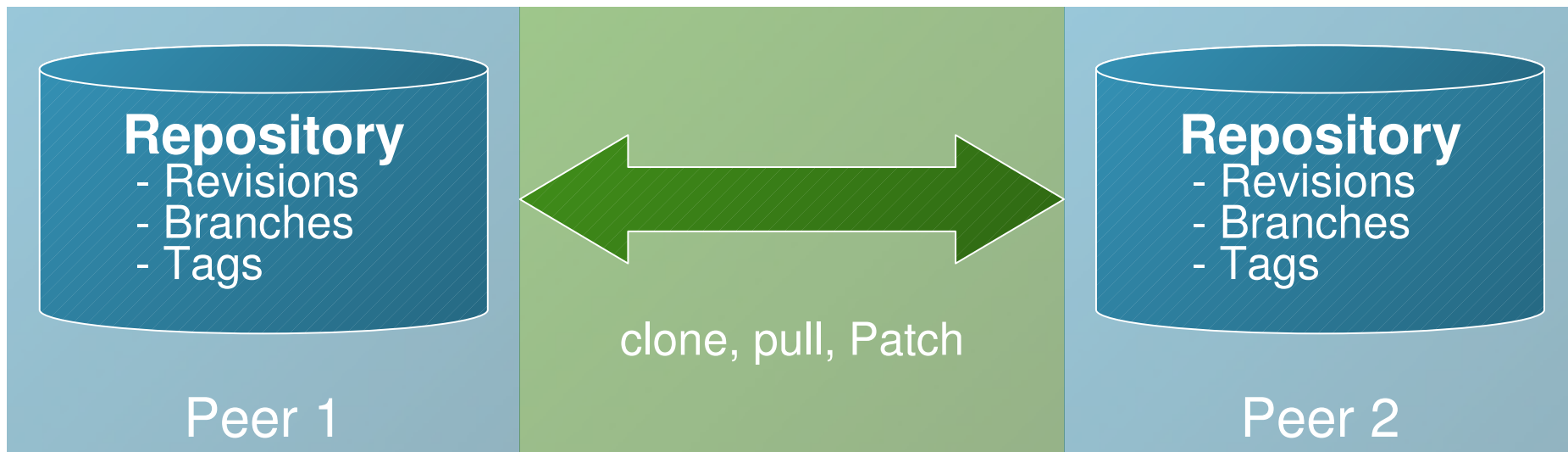
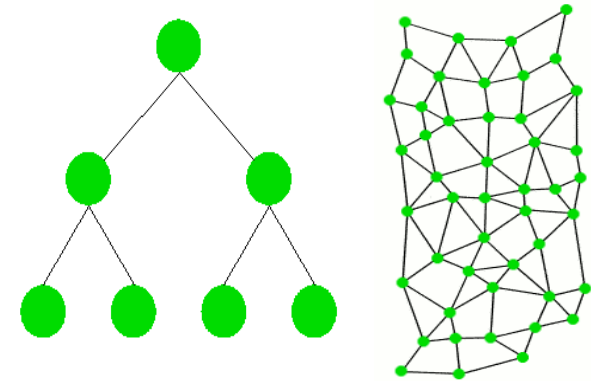
Beispiele verteilt: Standalone

- Anwendungsbeispiele
 - „Eigene Dateien bzw home-Laufwerk
 - Eigenes Java Projekt
 - /etc oder /www auf meinem Webserver

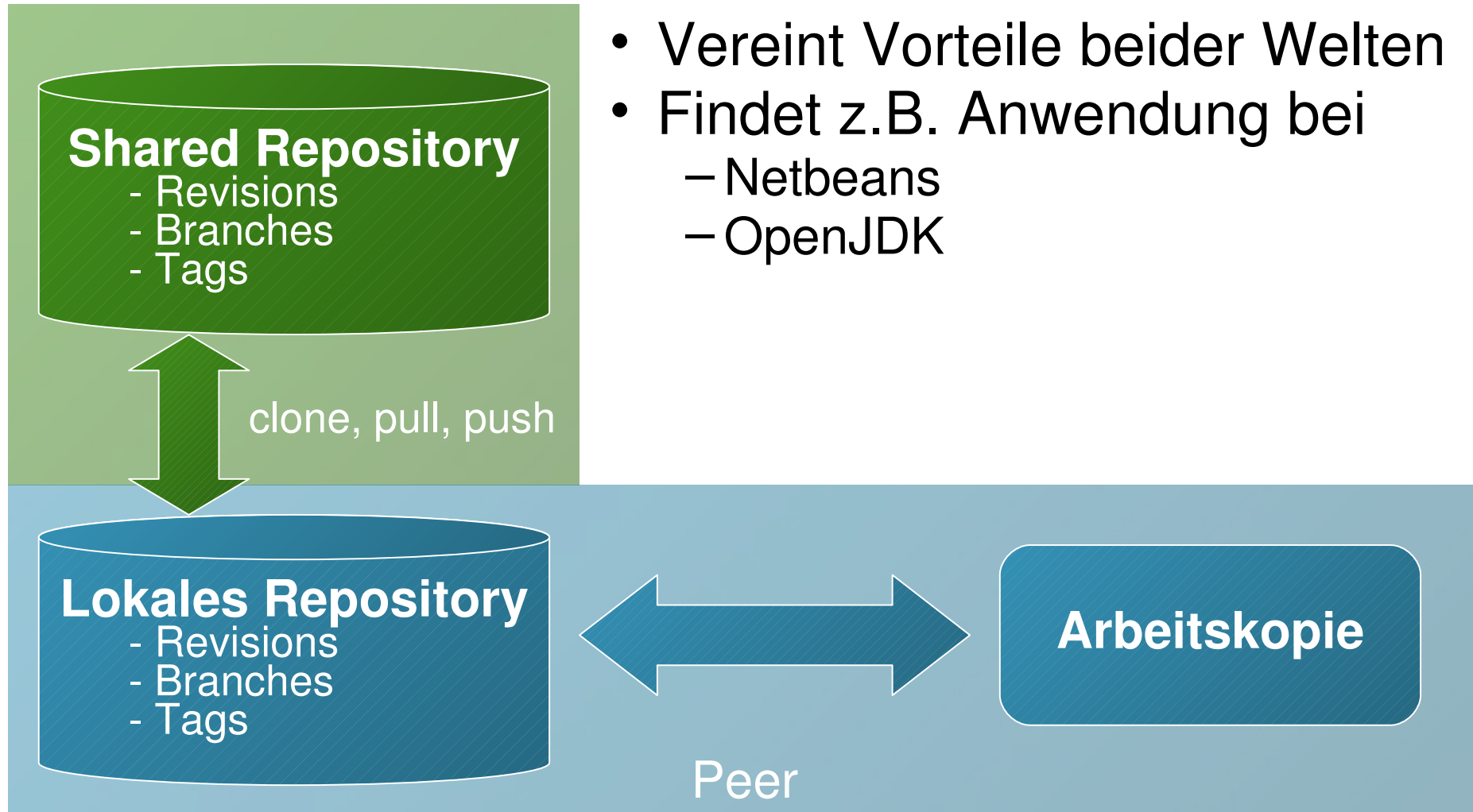


Beispiele verteilt: Peer-to-Peer

- Anwendungsbeispiele
 - Kleine Arbeitsgruppe (pull-only)
 - Maintainer Model, Linux-Kernel
 - Reviewer (Studenten/Junior)
 - Datensicherung



Beispiele verteilt: Doch zentral



Funktionsweise von Subversion, Mercurial, Git

Subversion

- Facts

- Entstanden als Alternative zu CVS
- Aktuelle Version 1.5.2
- Lizenz: Subversion Licence (BSD)
- Wo eingesetzt?
 - Apache Software Foundation
 - Sourceforge



- Neue Features in 1.4/1.5

- svnsync für Read-Only-Kopien
- WebDAV Write Through Proxy
- Merge Tracking
 - manueller Abgleich von Revisionsnummern entfällt



- Facts
 - Aktuelle Version: 1.0.1
 - Lizenz: GPL
 - Wo eingesetzt?
 - Mozilla
 - NetBeans
 - OpenSolaris
 - Entstanden 2005 (als Alternative zu Bitkeeper)
 - Implementierung in Python (und ein bisschen C)
- Eigenheiten
 - Ähnliche Syntax wie Subversion
 - Erweiterung durch Extensions
 - Klein & schnell (Komprimierung)

- Facts

- Aktuelle Version: 1.6.0
- Lizenz: GPL
- Wo eingesetzt?
 - Linux Kernel
 - X.Org
 - OLPC
 - Ruby on Rails
- Entstanden 2005 (als Alternative zu Bitkeeper)
- Implementierung in C und Perl

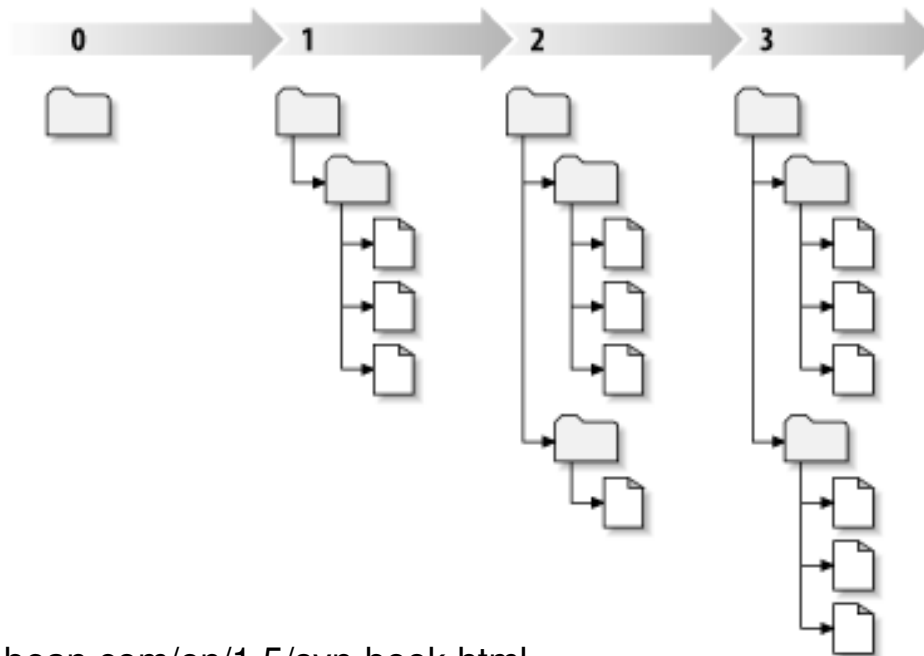
- Eigenheiten

- Von Kernelhackern für Kernelhacker
- Klein & sehr schnell (Komprimierung)



Revisionen in Subversion

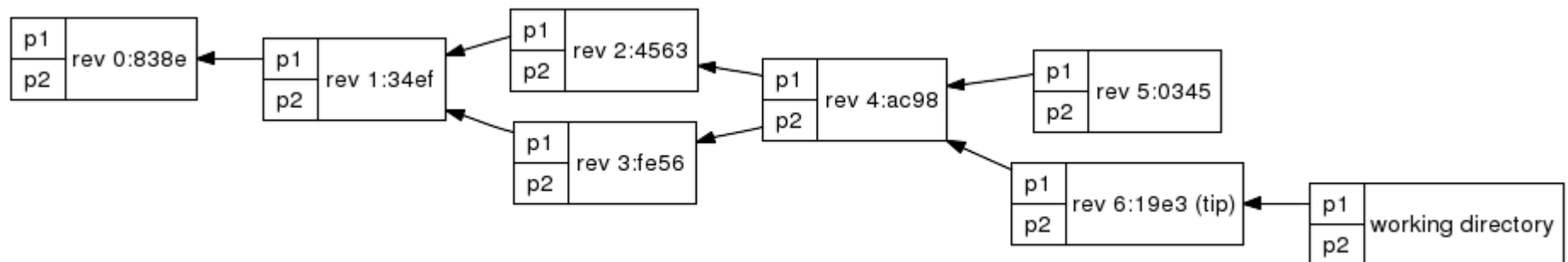
- Kette von Versionen: „Schnappschuss“
- Fortlaufende Nummerierung: 0 – 1 – 2 – 3 –
- Globale Revisionsnummern fürs komplette Repo.



Quelle: <http://svnbook.red-bean.com/en/1.5/svn-book.html>

Revisionen in Mercurial und Git

- Azyklischer gerichteter Graph von Changesets
- Revisionsnummern müssen eindeutig sein
 - Auch über Repository-Grenzen hinweg
 - Nutzung von SHA-1 Hashes (160 Bit, 40 Hex Zeichen)
 - Mercurial: zusätzlich lokale fortlaufende Nummern



example repository

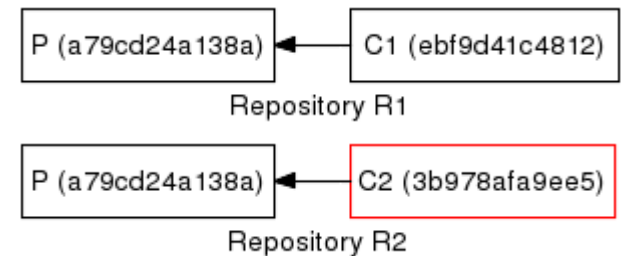
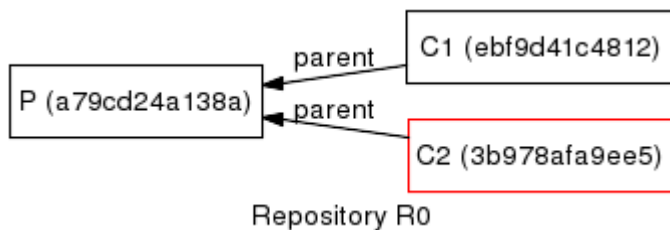
Quelle: <http://www.selenic.com/mercurial/wiki/index.cgi/UnderstandingMercurial>

Branches in Subversion

- Keine spezielles Branch Konzept
- Ein Branch ist eine Kopie eines Verzeichnisses
 - trunk -> branches/bugfixes-v1
 - Leichtgewichtig, nur eine Referenz
- Branches sind immer auf dem Server und damit global sichtbar

Branches in Mercurial

- Ein Branch entsteht, wenn ein Changeset mehr als ein Nachfolge-Changeset hat
 - Innerhalb eines Repositories
 - Eigenes Repository (clone)
 - Andere lokale bzw. zentrale Repositories
- Branches sind erstmal „anonym“, können aber einen Namen bekommen
- Hauptzweig default



Quelle: <http://www.selenic.com/mercurial/wiki/index.cgi/UnderstandingMercurial>

Branches in Git

- Git kennt lokale Branches
 - Ad hoc anlegen und löschen
- Haupt-Zweig `master`
- Bei `clone` eines Repositories gibt es einen Zweig `origin`, dieser enthält alle Zweige des Original-Repositories als Unterzweige

- Subversion
 - Änderungshistorie des Branches verloren
 - Änderungen des Branches werden im Arbeitsverzeichnis angewandt, dann committet
 - Kein Merge Tracking bis 1.4
- Mercurial und Git
 - Intelligenteres Mergen
 - Original-Commits bleiben erhalten (wer, wann, was)
 - Zusätzliche Merge-Änderungen durch Merger
 - Merge Tracking:
 - Startpunkt eines Branches
 - Was wurde bereits gemerged? (Revisionen)
 - Nächster Merge beginnt beim letzten durchgeführten

Workflow Subversion

Sourcecode vom Repository ins Arbeitsverzeichnis
`checkout`

Arbeiten

Dateien ändern

`add, delete, rename, status, diff, revert`

Nochmal updaten und Konflikte beseitigen (mergen)

`update, resolved`

Veröffentlichen

`commit`

Workflow Mercurial „Zentral“

Sourcecode vom zentralen Repository holen

`clone`

Arbeiten, lokal committen

Dateien ändern

`add, delete, rename, status, diff`

`commit, branch, merge`

Synchronisierten mit zentralem Repository

`fetch`

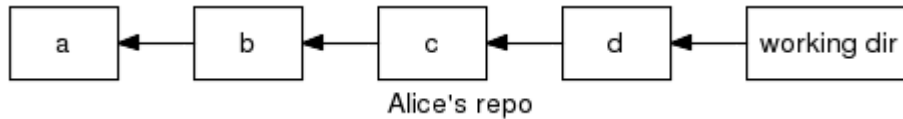
`pull`: remote Änderungen ins lokale Repo

`update`: Änderungen ins Arbeitsverzeichnis übernehmen

`merge + commit`: bei einem Konflikt

`push`

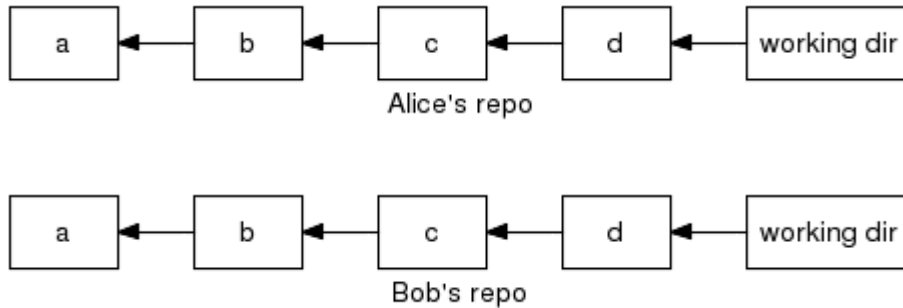
Workflow Mercurial „Peer-to-Peer“



Alice hat ein Repository

Quelle: <http://www.selenic.com/mercurial/wiki/index.cgi/UnderstandingMercurial>

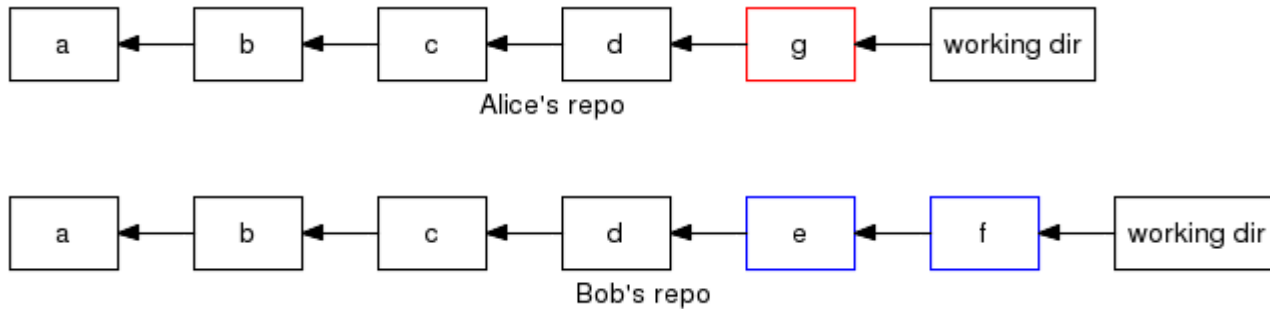
Workflow Mercurial „Peer-to-Peer“



- Bob kopiert Alice' Repository (`hg clone`)

Quelle: <http://www.selenic.com/mercurial/wiki/index.cgi/UnderstandingMercurial>

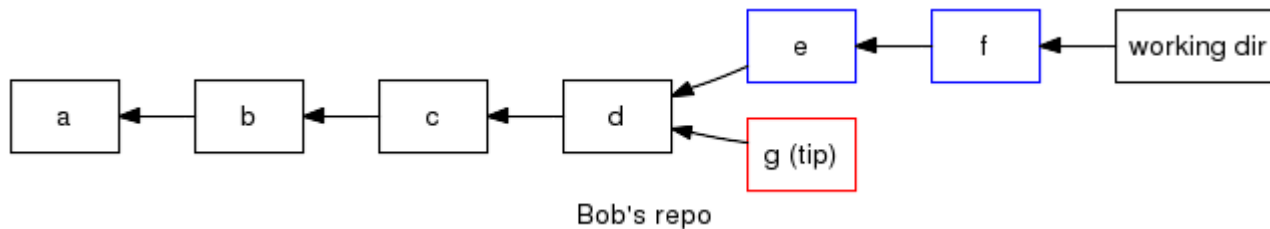
Workflow Mercurial „Peer-to-Peer“



- Alice und Bob machen unabhängig Änderungen an ihren Repositories (`hg commit`)
- Die Repositories divergieren!

Quelle: <http://www.selenic.com/mercurial/wiki/index.cgi/UnderstandingMercurial>

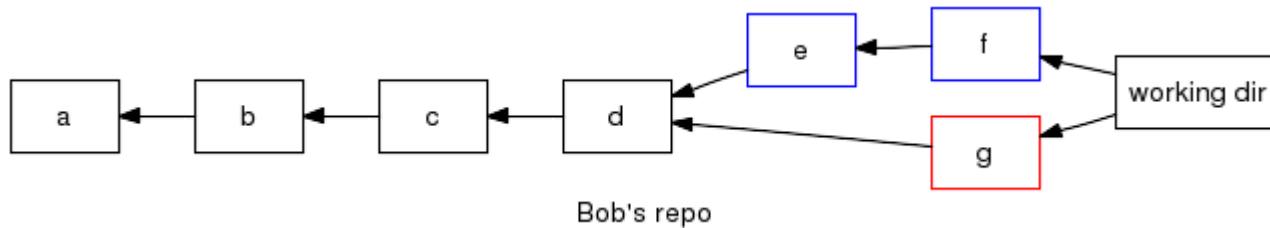
Workflow Mercurial „Peer-to-Peer“



- Bob holt Änderungen von Alice (`hg pull`) -> g
- Bob's Arbeitsverzeichnis bleibt unverändert
- Mercurial erkennt, dass d der Vorgänger von g ist **und** d noch weitere Kindknoten hat:
 - Ein (anonymer) Branch entsteht
 - Es muss gemergt werden!

Quelle: <http://www.selenic.com/mercurial/wiki/index.cgi/UnderstandingMercurial>

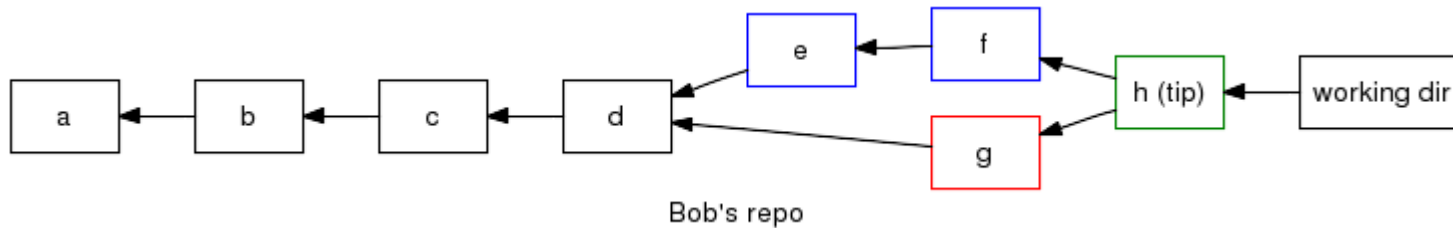
Workflow Mercurial „Peer-to-Peer“



- Bob merged Alice' Änderungen in sein Arbeitsverzeichnis (hg merge)
- Bob's Arbeitsverzeichnis muss sauber sein!
- Bob's Arbeitsverzeichnis zeigt nun auf zwei Revisionen
- Evtl. Konflikte muss Bob nun manuell lösen

Quelle: <http://www.selenic.com/mercurial/wiki/index.cgi/UnderstandingMercurial>

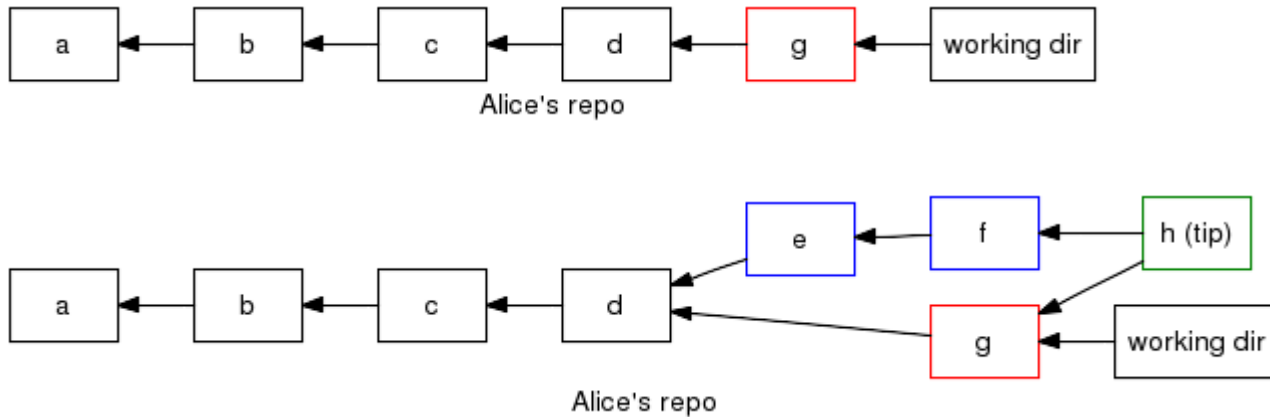
Workflow Mercurial „Peer-to-Peer“



- Bob erzeugt ein „Merge-Changeset“ h (`hg commit`)
- Alle originalen Änderungen von Alice (g) und Bob (e und f) bleiben erhalten!
- Changeset h beinhaltet
 - Die Information, dass f und g gemergt wurden
 - Evtl. Korrekturen von Konflikten

Quelle: <http://www.selenic.com/mercurial/wiki/index.cgi/UnderstandingMercurial>

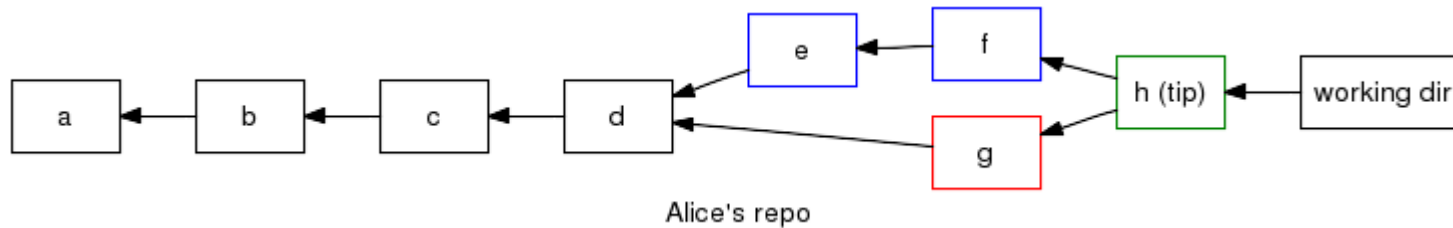
Workflow Mercurial „Peer-to-Peer“



- Alice holt Änderungen von Bob (`hg pull`) -> e, f und h
- Alice' Arbeitsverzeichnis bleibt noch unverändert!
- Mercurial erkennt, dass durch h bereits f und g gemergt wurden
 - Kein weiterer Merge notwendig! (Git: Fast-Forward)

Quelle: <http://www.selenic.com/mercurial/wiki/index.cgi/UnderstandingMercurial>

Workflow Mercurial „Peer-to-Peer“



- Mit `hg update` wird Alice' Arbeitsverzeichnis nach h aktualisiert

- Locks sind hilfreich beim bearbeiten von binären Dateien, da meist keine diff/merge möglich ist
- Subversion (und andere zentrale System) unterstützt Locks
- Mercurial und Git (und auch andere dezentrale Systeme) unterstützen keine Locks

Löschen aus Historie

- Hintergrund:
 - Versehentlich eingecheckte Passwörter
 - Vertrauliche Dokumente
- Subversion
 - Nur über externes Tool (svn dumpfilter) möglich
 - dump-filter-load des gesamten Repositories notwendig
- Mercurial
 - hg qdel aus MqExtension
- Git
 - cg-admin-rewritehist aus Cogito

Dateien umbennennen / verschieben

- Subversion
 - `svn move` bzw. `svn rename`
 - Problem: C2 ändert A, C1 verschiebt A->B
- Mercurial
 - `hg rename`
 - `hg log --follow`
 - Problem: C1 verschiebt A->B, C2 verschiebt A->X
- Git
 - Git verwaltet keine Datei-/Verzeichnisnamen
 - Heuristik erkennt, wenn Inhalt „wandert“
 - `git mv` oder per Dateisystem Operationen
 - `git log --follow`
 - Problem: Umbennennen A->B + neu anlegen von A

Zugriffsmöglichkeiten

- Subversion
 - http/https (WebDAV): Apache Modul
 - ssh
 - svn-Protokoll (svnserve), Lokales Filesystem
- Git
 - http/https (WebDAV) (pull+push)
 - ssh (git-shell) (pull+push)
 - Git-Protokoll (git daemon), rsync, FTP, E-Mail, Lokales Filesystem
- Mercurial
 - http/https über hg serve (nur pull) oder über cgi (pull+push)
 - ssh (pull+push)
 - E-Mail bundles (Extension), Lokales Filesystem

Authentifizierung

- Subversion
 - http/https: Auth-Module des Apache HTTP-Server
 - ssh: Authentifizierung gegen SSH-Server (Shellzugriff!)
 - svn: Username/Passwort wird über config-file definiert; optional SASL
- Git
 - http/https: Auth-Verfahren des HTTP-Servers
 - ssh: Authentifizierung gegen SSH-Server
 - git daemon: Bietet keine Authentifizierungsmöglichkeit
- Mercurial
 - http/https: Auth-Verfahren des HTTP-Servers; hg serve bietet read Access für jedermann
 - ssh: Authentifizierung gegen SSH-Server

Weitere Eigenschaften

- Keywords
 - Subversion: Über Properties
 - \$Rev\$, \$Author\$, \$Date\$
 - Git: Keine Unterstützung
 - Mercurial: Unterstützung über Extension
- File Ignores
 - Werden von allen 3 Versionsverwaltungen unterstützt
- Hooks
 - Werden von allen 3 Versionsverwaltungen unterstützt
- Daten der Base bzw. des Repository
 - Subversion: .svn Verzeichnis in jedem Source Verzeichnis
 - Mercurial/Git: Ein .hg bzw. .git Verzeichnis im Wurzelverzeichnis

Tools und Entwicklungswerkzeuge

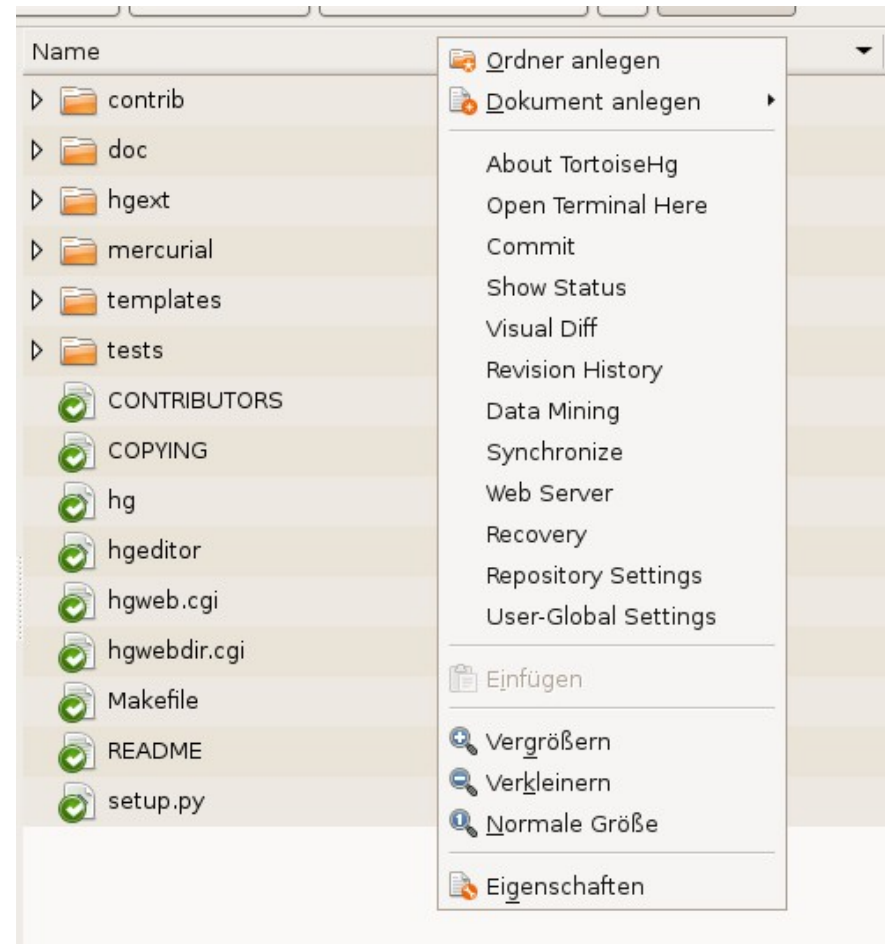
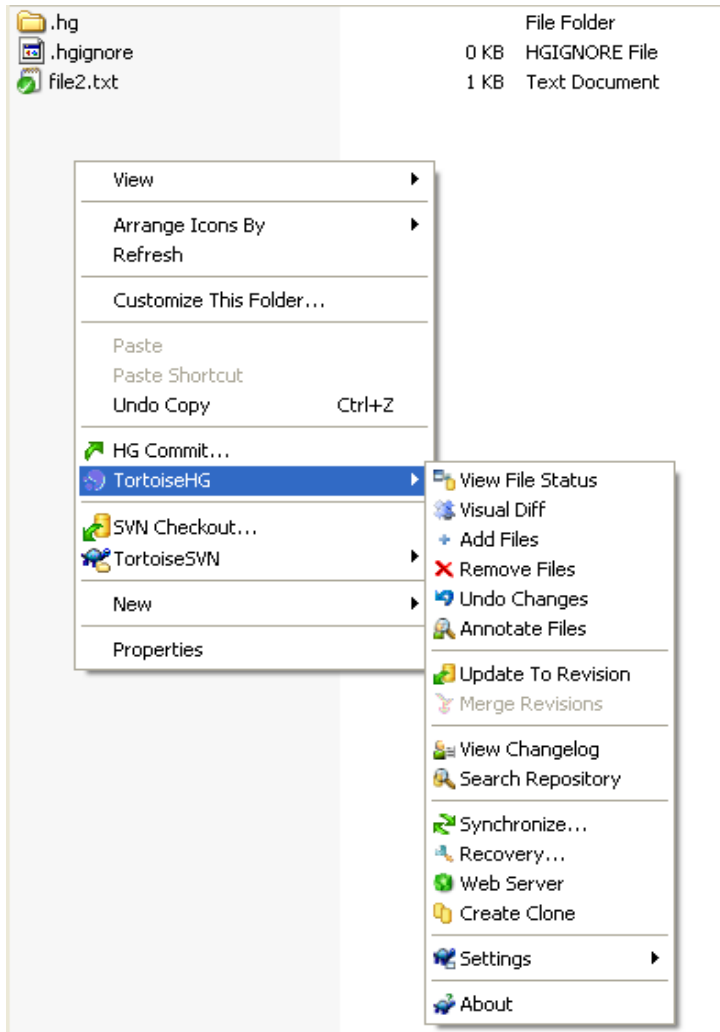
Plattformen

	Subversion	Git	Mercurial
Linux	++	++	++
Mac	++	+	++
Windows	++	+	++

- Subversion
 - Windows: TortoiseSVN
 - Linux: Gnome und KDE GUIs (KDESVN, RapidSVN, eSVN)
- Mercurial
 - Webseite listet 13 verschiedene Tools
 - TortoiseHG (Windows/Linux), hgk (Windows/Linux → Tcl/Tk)
 - Export des Graphen als Bild
- Git
 - Windows: msysgit beinhaltet gitk, git-gui
 - Linux: gitk, git-gui, Zusatztools



TortoiseHG Windows und Linux



- Subversion
 - Einfacher Repository Browser bei Installation des Apache Moduls bereits enthalten
 - Daneben existieren ViewVC und ein halbes Dutzend anderer Tools
- Mercurial
 - hg serve bietet auch eine Weboberfläche
 - Auch das CGI Skript beinhaltet diese Oberfläche
- Git
 - gitweb inklusive, erfordert lighttpd oder Apache
 - git-instaweb für schnelles Setup
 - weitere Tools

summary | [shortlog](#) | [changelog](#) | [tags](#) | [manifest](#) | [bz2](#) | [zip](#) | [gz](#)

description JDK 7 Master
owner jdk7-dev@openjdk.java.net
last change Thu, 23 Oct 2008 10:12:57 -0700

changes

3 days ago	xdono	Added tag jdk7-b38 for changeset cc47a76899ed	default	tip	changeset manifest
2 weeks ago	xdono	Added tag jdk7-b37 for changeset 744554f5a329	jdk7-b38		changeset manifest
3 weeks ago	xdono	6754988: Update copyright year	jdk7-b37		changeset manifest
4 weeks ago	xdono	Added tag jdk7-b36 for changeset 4b4f5fea8d7d			changeset manifest
5 weeks ago	xdono	Merge	jdk7-b36		changeset manifest
5 weeks ago	ohair	6724787: OpenJDK README-builds.html suggested changes			changeset manifest
6 weeks ago	ohair	Merge			changeset manifest
6 weeks ago	ohair	6746424: Remove build dependency on findbugs and FINDBUGS_HOME			changeset manifest
6 weeks ago	tbell	Merge			changeset manifest
6 weeks ago	martin	6746055: Spelling error in README-builds.html			changeset manifest

...

tags

2 weeks ago	jdk7-b38	changeset changelog manifest
3 weeks ago	jdk7-b37	changeset changelog manifest
5 weeks ago	jdk7-b36	changeset changelog manifest
8 weeks ago	jdk7-b35	changeset changelog manifest
2 months ago	jdk7-b34	changeset changelog manifest
2 months ago	jdk7-b33	changeset changelog manifest
3 months ago	jdk7-b32	changeset changelog manifest
3 months ago	jdk7-b31	changeset changelog manifest
3 months ago	jdk7-b30	changeset changelog manifest
4 months ago	jdk7-b29	changeset changelog manifest

...

branches

3 days ago	ab523b49de1f	default	changeset changelog manifest
------------	------------------------------	---------	--

...

jdk7/jdk7

RSS

summary | [shortlog](#) | [log](#) | [commit](#) | [commitdiff](#) | [tree](#)

commit



?

search:

description Linus' kernel tree
owner Linus Torvalds
last change Sun, 26 Oct 2008 23:42:18 +0000
URL [git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux-2.6.git](http://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux-2.6.git)
<http://www.kernel.org/pub/scm/linux/kernel/git/torvalds/linux-2.6.git>

shortlog

8 hours ago	Linus Torvalds	Merge branch 'for-linus' of git://neil.brown.name/md	master	commit	commitdiff	tree	snapshot
8 hours ago	Linus Torvalds	Merge branch 'for-linus' of git://git./linux/kernel ...		commit	commitdiff	tree	snapshot
8 hours ago	Linus Torvalds	Merge branch 'for-linus' of git://git.o-hand.com/linux-mfd		commit	commitdiff	tree	snapshot
8 hours ago	Linus Torvalds	Merge branch 'for-upstream' of git://git./linux/kernel ...		commit	commitdiff	tree	snapshot
8 hours ago	Linus Torvalds	Merge branch 'for-linus' of git://git./linux/kernel ...		commit	commitdiff	tree	snapshot
8 hours ago	Randy Dunlap	docbooks: fix fatal filename errors		commit	commitdiff	tree	snapshot
8 hours ago	Randy Dunlap	kernel-doc: allow more whitespace in macros		commit	commitdiff	tree	snapshot
8 hours ago	Linus Torvalds	Merge branch 'release' of git://git./linux/kernel/git ...		commit	commitdiff	tree	snapshot
13 hours ago	Linus Torvalds	Linux 2.6.28-rc2	v2.6.28-rc2	commit	commitdiff	tree	snapshot
13 hours ago	Geert Uytterhoeven	m68k: Disable Amiga serial console support if modular		commit	commitdiff	tree	snapshot
13 hours ago	Davide Libenzi	epoll: avoid double-inserts in case of EFAULT		commit	commitdiff	tree	snapshot
13 hours ago	Arjan van de Ven	select: deal with math overflow from borderline valid ...		commit	commitdiff	tree	snapshot
14 hours ago	Arjan van de Ven	wireless: fix regression caused by regulatory config ...		commit	commitdiff	tree	snapshot
15 hours ago	Stephen Rothwell	cgroup: remove unused variable		commit	commitdiff	tree	snapshot
15 hours ago	Linus Torvalds	Merge branch 'hwmon-for-linus' of git://jdelvare.pck ...		commit	commitdiff	tree	snapshot
15 hours ago	Francois Romieu	r8169: revert "read MAC address from EEPROM on init"		commit	commitdiff	tree	snapshot

[...](#)

tags

13 hours ago	v2.6.28-rc2	Linux 2.6.28-rc2	tag	commit	shortlog	log
3 days ago	v2.6.28-rc1	Linux 2.6.28-rc1	tag	commit	shortlog	log
2 weeks ago	v2.6.27	Linux 2.6.27	tag	commit	shortlog	log
2 weeks ago	v2.6.27-rc9	Linux 2.6.27-rc9	tag	commit	shortlog	log
3 weeks ago	v2.6.27-rc8	Linux 2.6.27-rc8	tag	commit	shortlog	log
5 weeks ago	v2.6.27-rc7	Linux 2.6.27-rc7	tag	commit	shortlog	log
6 weeks ago	v2.6.27-rc6	Linux 2.6.27-rc6	tag	commit	shortlog	log
8 weeks ago	v2.6.27-rc5	Linux 2.6.27-rc5	tag	commit	shortlog	log
2 months ago	v2.6.27-rc4	Linux 2.6.27-rc4	tag	commit	shortlog	log
2 months ago	v2.6.27-rc3	Linux 2.6.27-rc3	tag	commit	shortlog	log
2 months ago	v2.6.27-rc2	Linux 2.6.27-rc2	tag	commit	shortlog	log
2 months ago	v2.6.27-rc1	Linux 2.6.27-rc1	tag	commit	shortlog	log
2 months ago	v2.6.26	Linux 2.6.26	tag	commit	shortlog	log

Migration

- Wichtig: Historie bleibt erhalten

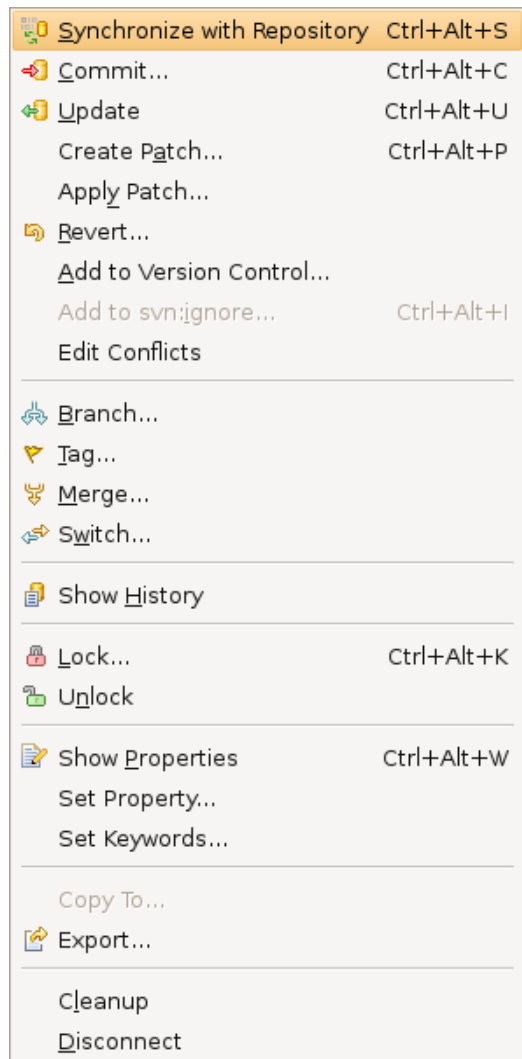
	Subversion	Mercurial	Git
CVS	cvs2svn	hg convert	cvs2git git-cvsmimport
Subversion	X	hg convert	git-svn
Mercurial	X	X	hg-fast-export
Git	X	hg convert	X

Subversion Bridges

- Funktionsweise
 - Import eines svn-Repository (inkl. History) in lokales Repository
 - lokale commits und Änderungen
 - Push-back in das Subversion Repository
- Subversion ↔ Git
 - git-svn ermöglicht bidirektionalen Abgleich mit Subversion
- Subversion ↔ Mercurial
 - Push-back zu Subversion problematisch
 - Erstellen und Einspielen von Patches
 - Hgsubversion noch nicht für Produktiveinsatz geeignet

	Subversion	Git	Mercurial
Eclipse	Subversive und Subclipse	Eclipse GIT (nutzt Java GIT)	Mercurial Eclipse und merclipse
Netbeans	integriert, benötigt Kommandozeilentool	nbgit (nutzt Java GIT)	seit Version 6.1 integriert, benötigt Kommandozeilentool
IntelliJ IDEA	bereits enthalten	Git4Idea: benötigt Kommandozeilen-tool, momentan werden nicht alle Funktionen unterstützt	mercurialidea: benötigt Kommandozeilentool, momentan werden nicht alle Funktionen unterstützt

Subversive



Mercurial Eclipse

- Apply Patch...
- Commit...**
- Push...
- Pull...
- Update
- Import patch...

- Tag as Version...
- Merge...
- Add Branch...
- Update to another changeset...

- Show History
- Synchronize...

- Revert...
- Undo
- mqqueue

- Sign...
- Transplant...

- Serve repository...

- Remove from Version Control
- Add to .hgignore...

- Refresh status
- Disconnect

/mercurial						
Graph	Changeset	Tag	Branch	User	Date	Summary
	6642:c8c7248f6172			Adrian Buehlma	2008-05-31 11:37 +0200	resolve: add file codes to doc string
	6641:1603bba96411			Dirkjan Ochtma	2008-05-29 22:23 +0200	merge from crew-stable
	6640:d6b53b48943f			Christian Ebert	2008-05-29 16:29 +0200	Make test-bundle-type work on MacOS
	6639:6334569c8caa			Patrick Mezard	2008-05-28 22:45 +0200	localrepo: fix partial merge test (issue 1
	6638:5868d0b8509f			Lee Cantey	2008-05-25 13:39 -0700	Fix bad span tag in Readme
	6637:79434e68512a			Patrick Mezard	2008-05-24 18:23 +0200	Merge with crew-stable
	6636:0397f592372f			Dirkjan Ochtma	2008-05-23 14:02 +0200	merge from crew-stable
	6635:d90d83ebea9e			Dirkjan Ochtma	2008-06-03 12:10 +0200	mq: don't update the working copy on s
	6634:d6caebe9c293			Adrian Buehlma	2008-06-03 09:31 +0200	mq: qimport: explain insertion point in d
	6633:67a2d2d9bf21			Patrick Mezard	2008-06-02 21:38 +0200	test-convert-mtn: test descriptions with
	6632:aa3f61884a48			David Reiss	2008-05-16 00:48 -0700	convert.monotone: fix quotes and back
	6631:a2b13cac0922			Stefano Tortaro	2008-05-29 22:21 +0200	Active branches fix (issue1104)
	6630:8542fac26f63			Benoit Boissino	2008-05-26 14:20 +0200	hgweb: correctly validate permissions w

Continuous Integration Tools

	Subversion	Git	Mercurial
CruiseControl	Ja	Ja	Ja
Hudson	Ja	Ja	Ja
Apache Continuum	Ja	Nein	Ja
Atlassian Bamboo	Ja	Nein	Nein
JetBrains TeamCity	Ja	Nein	Nein

Fazit

- Die „Verteilten“ bringen frischen Wind
- Zentral vs. Verteilt
 - Firmenpolitik und Projektvorgaben
 - Stabilität und Zukunftssicherheit?
 - Plattformen und Tool-Unterstützung sehr volatil
- Git vs. Mercurial (vs. Bazaar)
 - Kein klarer Gewinner bei den verteilten
 - Windows: Mercurial
 - SVN Mischbetrieb: Git mit git-svn

- Vergleich der Features verschiedener Versionsverwaltungen
 - <http://better-scm.berlios.de/comparison/comparison.html>
 - <http://www.infoq.com/articles/dvcs-guide>
 - http://en.wikipedia.org/wiki/Comparison_of_revision_control_softwareGit
- Git
 - <http://git.or.cz/>
 - <http://book.git-scm.com/>
 - Java GIT/Eclipse GIT
 - <http://git.or.cz/gitwiki/EclipsePlugin>
- Mercurial
 - <http://www.selenic.com/mercurial>
 - Mercurial Eclipse
 - <http://www.vectrace.com/mercurialeclipse/>