

# Principles of Software Construction: Objects, Design, and Concurrency

## Version Control

**Jonathan Aldrich**   **Charlie Garrod**

Based on slides by Christian Kästner

# Administrivia

- Extra office hours for HW6 and exam
  - See online schedule
- Sample final exam – coming soon
- Review session Wednesday 12/16
  - 2-4pm in DH 1112
- Final exam Thursday 12/17
  - 8:30-11:30 in MM 103 & MM A14

# Learning Goals

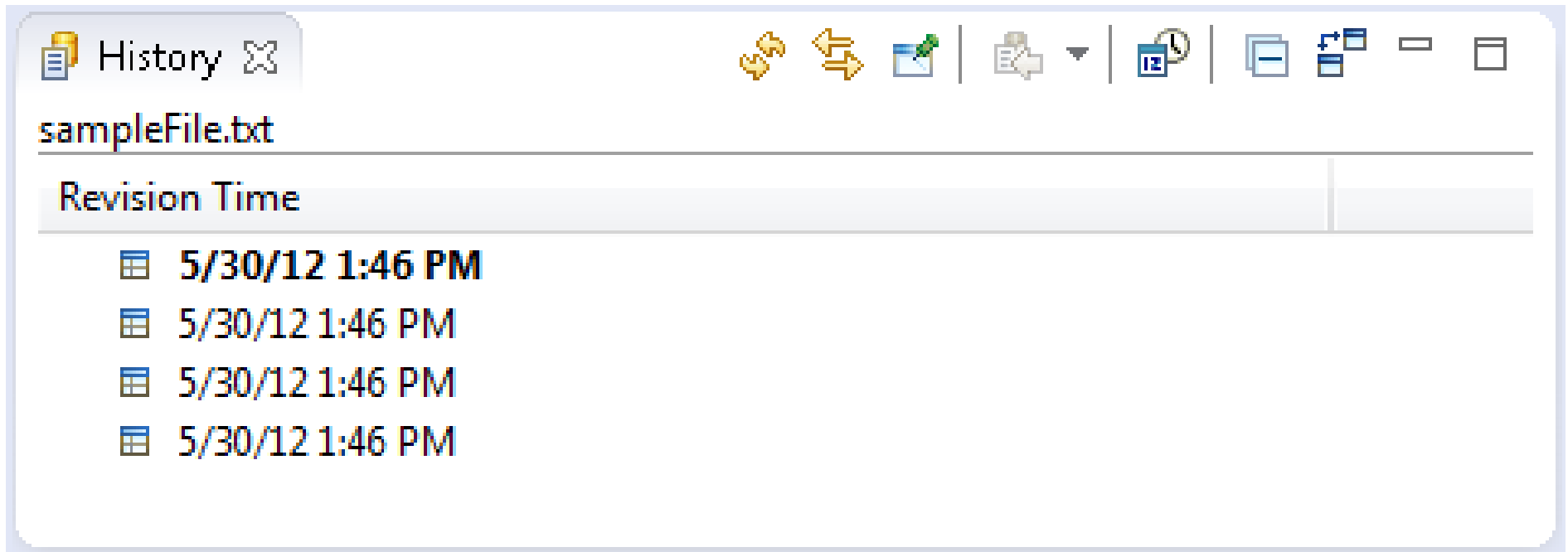
- Understand the benefits and limitations of version control (history, parallel development, branching, etc)
- Ability to cooperate on a code base using branching and merging; to resolve merge conflicts
- Distinguish classes of version control systems (local, central, distributed)

# HISTORY

# History

- Record relevant steps in the history of the project
- Supports partial undo
- Label cohesive development steps
  - Allows understanding changes
  - Explains rationale
  - Traceability to other development artifacts (bug trackers, CVEs, requirements, etc)
- Record who performed changes
- Implies a backup
- Early local systems: SCCS and RCS
- -> Version control systems useful even when used entirely locally

# Eclipse's Local History



The screenshot shows the Eclipse IDE's Local History window for a file named `sampleFile.txt`. The window title is "History" with a close button. The toolbar contains icons for undo, redo, compare, and other actions. The main area displays a table with a header "Revision Time" and four rows of revision information, all dated "5/30/12 1:46 PM".


Revision Time
5/30/12 1:46 PM
5/30/12 1:46 PM
5/30/12 1:46 PM
5/30/12 1:46 PM

# Revision History


Revision	Tags	Revision Time	Author	Comment
*1.64	v20100831-0800, r361_v20100...	11.09.08 13:59	dme...	Removed trailing whitespace and organized the imports
1.63	r342_v20091014, r342_v20090716, ...	28.01.08 10:51	bbau...	216167 [performance][breadcrumb] Java editor leak after closing editor when in ..
1.62	v20080122-0800a, v20080122-0800	21.01.08 13:04	bbau...	215965 [breadcrumb] Activating editor should set focus into breadcrumb if it was...
1.61	v20080115-0800, v20080108-0800, ...	21.06.07 14:44	dmegert	Fixed bug 182108: [api] Add action constants for IDEActionFactory and ActionF...
1.60	R3_3_2, r332_20080128, r332_2008...	12.04.07 12:33	dmegert	Fixed bug 178744: [misc] Editor opened upon an out of sync resource should allo...
1.59	v20070410-0800, v20070403-0800, ...	28.03.06 18:53	mkeller	Updated copyright.
1.58	v20060327-1600, v20060327-0010, ...	10.02.06 12:27	mkeller	Renamed BasicEditorActionContributor
1.57	v20060207-0800, v20060131-0800, ...	18.08.05 15:44	mkeller	removed unnecessary \$NON-NLS-*\$ tags
1.56	R3_1_2, r312_v20060104-0800, r31...	17.06.05 17:51	dmegert	Updated copyright date to 2005
1.55	v20050617-preCopyrightPass, v200...	19.04.05 14:47	dmegert	Removed trailing whitespace.
1.54	v20050419-0800	19.04.05 11:02	dmegert	Converted to Eclipse NLS.
1.53	v20050412-0800, v20050405-0800, ...	25.02.05 16:34	dbae...	Converted from CPL to EPL
1.52	R3_0_2, r302_v20050311-0800, v20...	23.06.04 21:33	dmegert	Copyright fix - automated changes with releng tool
1.51	v20040522-1600, pre-copyright, v2...	02.05.04 17:10	dmegert	Batch from Christof Marti: some modifications...

Removed trailing whitespace and organized the imports


Pass, buy special tile GUI impl [ci skip]

 master


fixes

 master


completed Exercise class

 master


Checkpoint

 master


finish boom

 master

GUI Finished, Test Cases Not Finised

 master

new environment

 master

 Showing 1,862 changed files with 0 additions and 0 deletions.

Sorry, we could not display the entire diff because too many files (1,862) changed.



final

authored 11 days ago



final

authored 11 days ago



final

authored 11 days ago

a

authored 11 days ago



final

authored 11 days ago

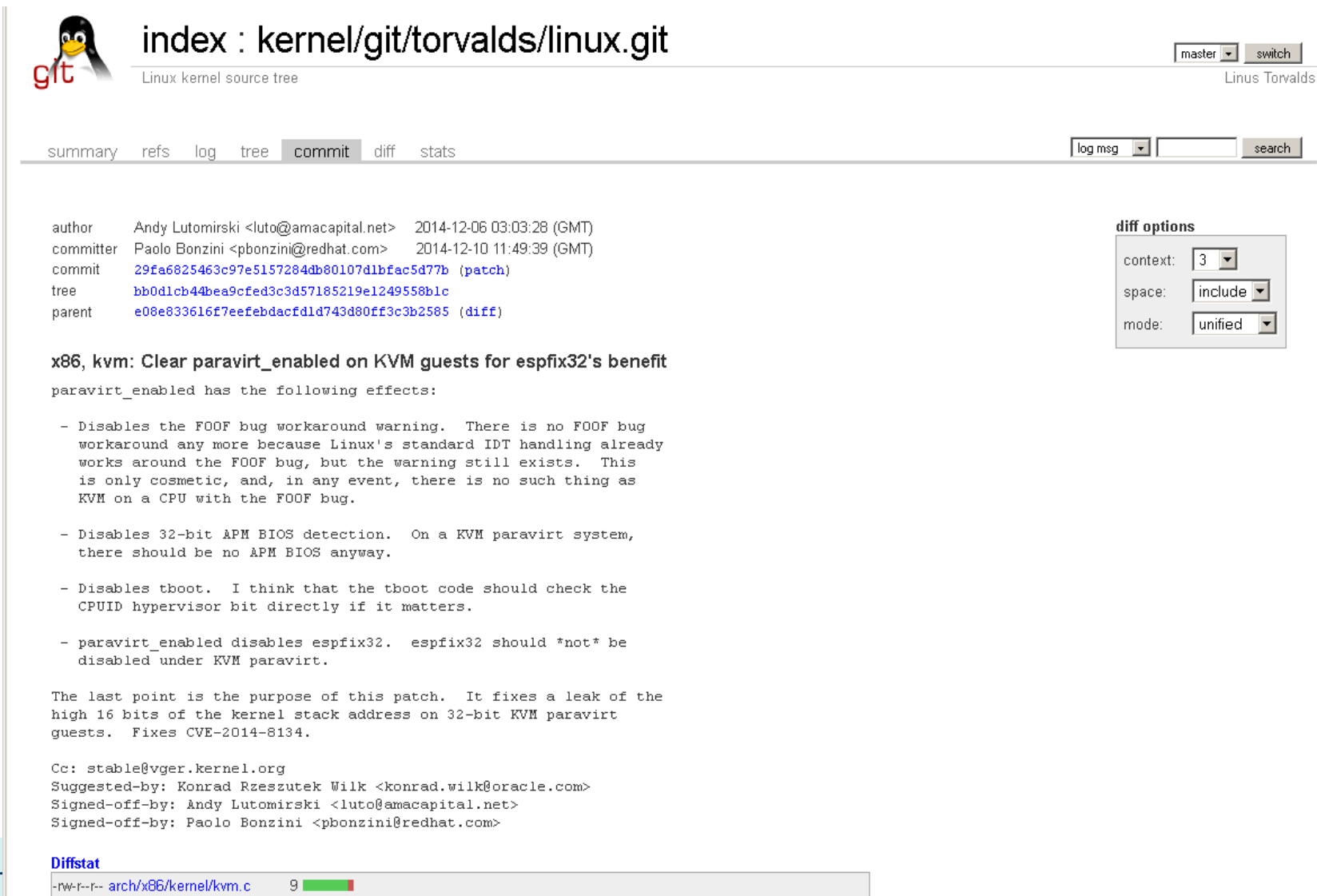


final

authored 11 days ago



# Recording Rationale and Traceability



 index : kernel/git/torvalds/linux.git master switch  
Linux kernel source tree Linus Torvalds

summary refs log tree **commit** diff stats log msg search

author Andy Lutomirski <luto@amacapital.net> 2014-12-06 03:03:28 (GMT)  
committer Paolo Bonzini <pbonzini@redhat.com> 2014-12-10 11:49:39 (GMT)  
commit 29fa6825463c97e5157284db80107d1bfac5d77b (patch)  
tree bb0d1cb44bea9cfed3c3d57185219e1249558b1c  
parent e08e833616f7eefebdacfd1d743d80ff3c3b2585 (diff)

**x86, kvm: Clear paravirt\_enabled on KVM guests for espfix32's benefit**

paravirt\_enabled has the following effects:

- Disables the FOOF bug workaround warning. There is no FOOF bug workaround any more because Linux's standard IDT handling already works around the FOOF bug, but the warning still exists. This is only cosmetic, and, in any event, there is no such thing as KVM on a CPU with the FOOF bug.
- Disables 32-bit APM BIOS detection. On a KVM paravirt system, there should be no APM BIOS anyway.
- Disables tboot. I think that the tboot code should check the CPUID hypervisor bit directly if it matters.
- paravirt\_enabled disables espfix32. espfix32 should \*not\* be disabled under KVM paravirt.

The last point is the purpose of this patch. It fixes a leak of the high 16 bits of the kernel stack address on 32-bit KVM paravirt guests. Fixes CVE-2014-8134.

Cc: stable@vger.kernel.org  
Suggested-by: Konrad Rzeszutek Wilk <konrad.wilk@oracle.com>  
Signed-off-by: Andy Lutomirski <luto@amacapital.net>  
Signed-off-by: Paolo Bonzini <pbonzini@redhat.com>

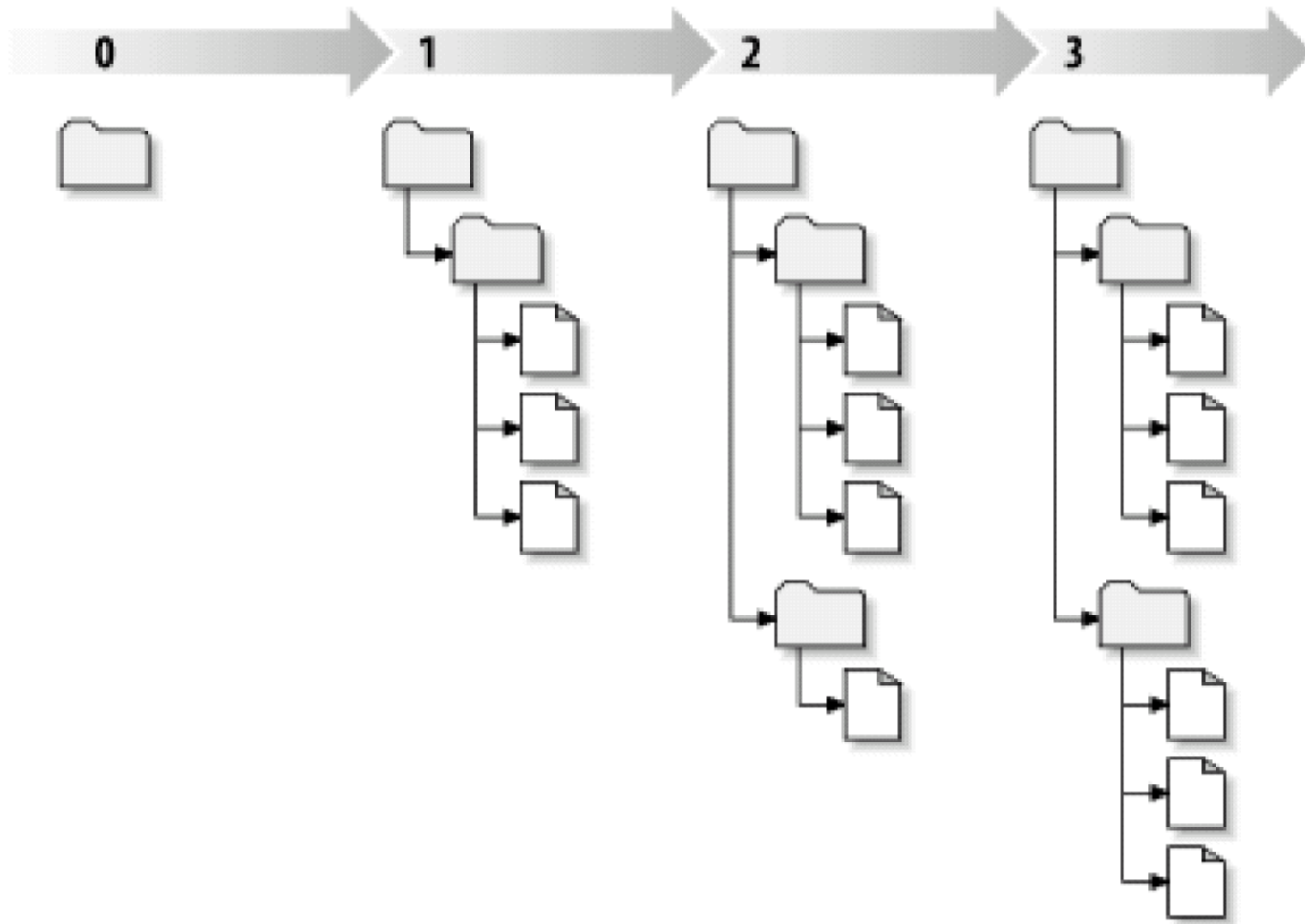
**diff options**

context: 3  
space: include  
mode: unified

**Diffstat**

-rw-r--r-- arch/x86/kernel/kvm.c	9
----------------------------------	---

# Versioning entire projects



# Versioning and Tags

- Version numbers for files individually vs. per project
- Version numbers vs. hashes
- Tags to name specific states

# While files to manage

- All code and noncode files
  - Java code
  - Build scripts
  - Documentation
- Exclude generated files (.class, ...)
- Most version control systems have a mechanism to exclude files (e.g., .gitignore)

# COLLABORATION

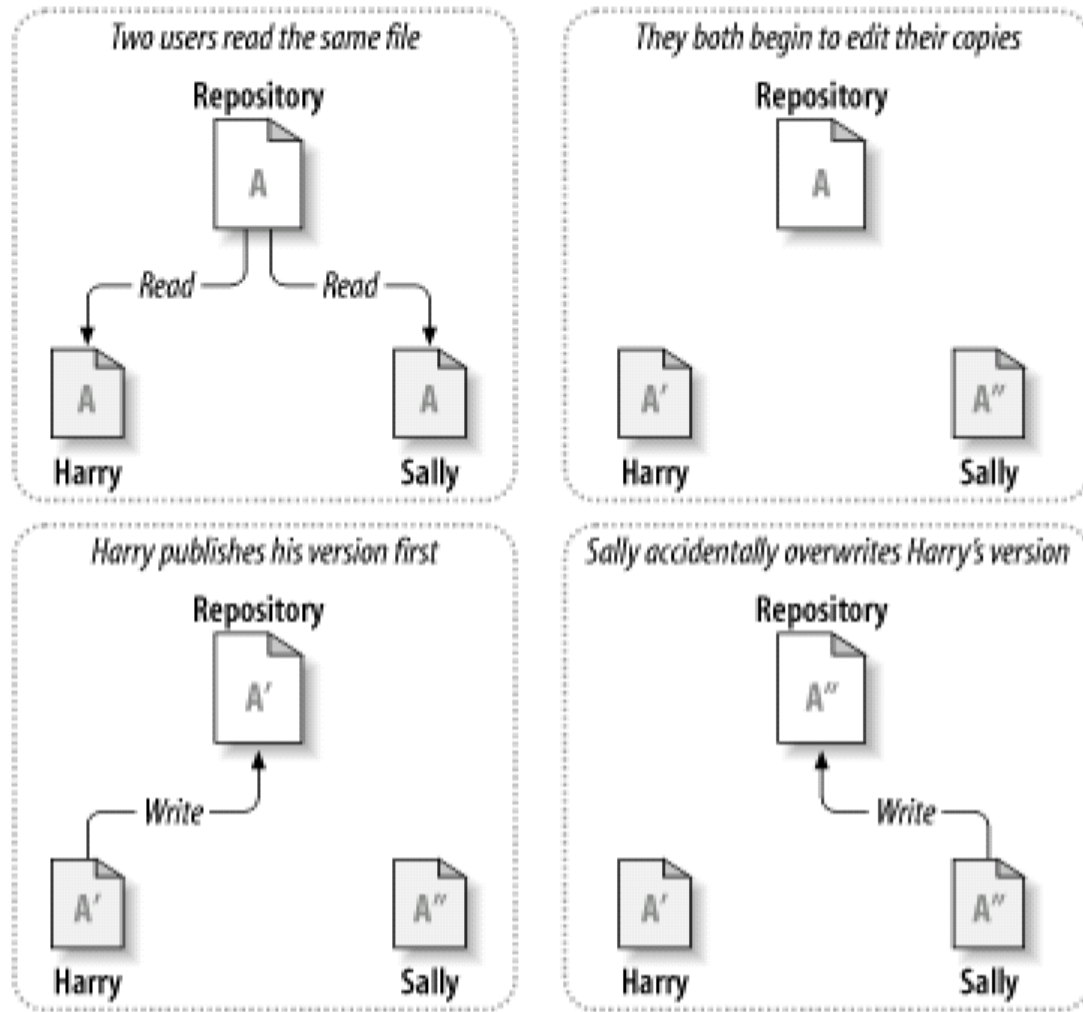
# Collaborating on Files

- How to exchange files
  - Send changes by email
  - Manual synchronization at project meeting
  - All files on shared network directory
  
- Permission models
  - Each file has an owner; only person allowed to change it
  - Everybody may change all files (collective ownership)

# Concurrent Modifications

- Allowing concurrent modifications is challenging
- Conflicts (accidental overwriting) may occur
- Common strategies
  - Locking to change
  - Detecting conflicts (optimistic model)

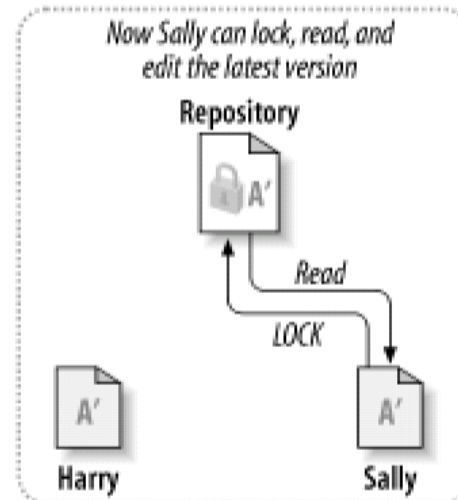
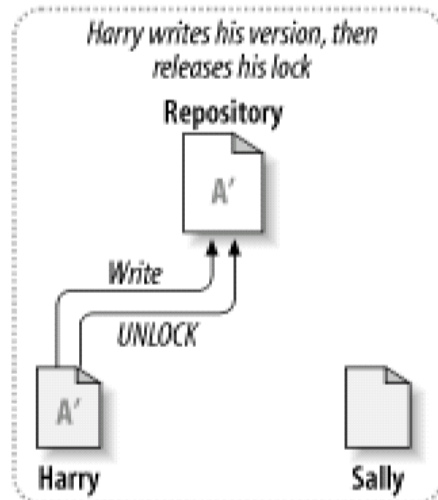
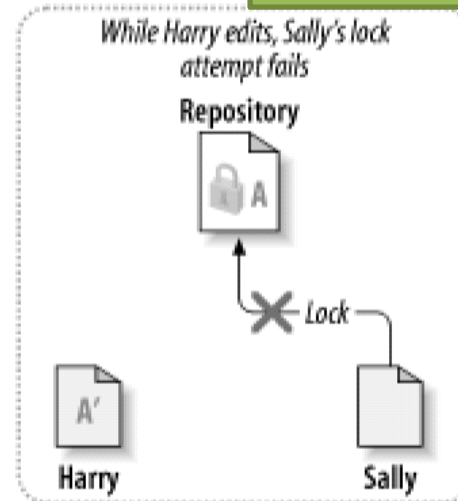
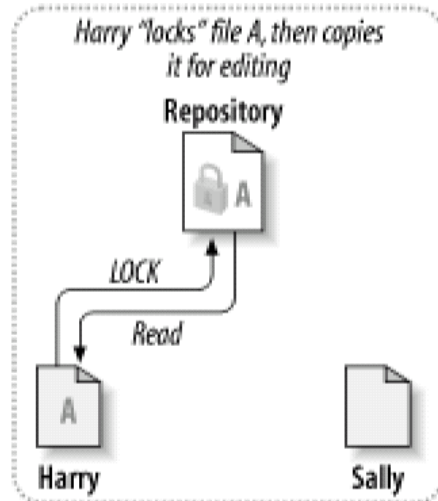
# Change Conflicts





# Locking Files

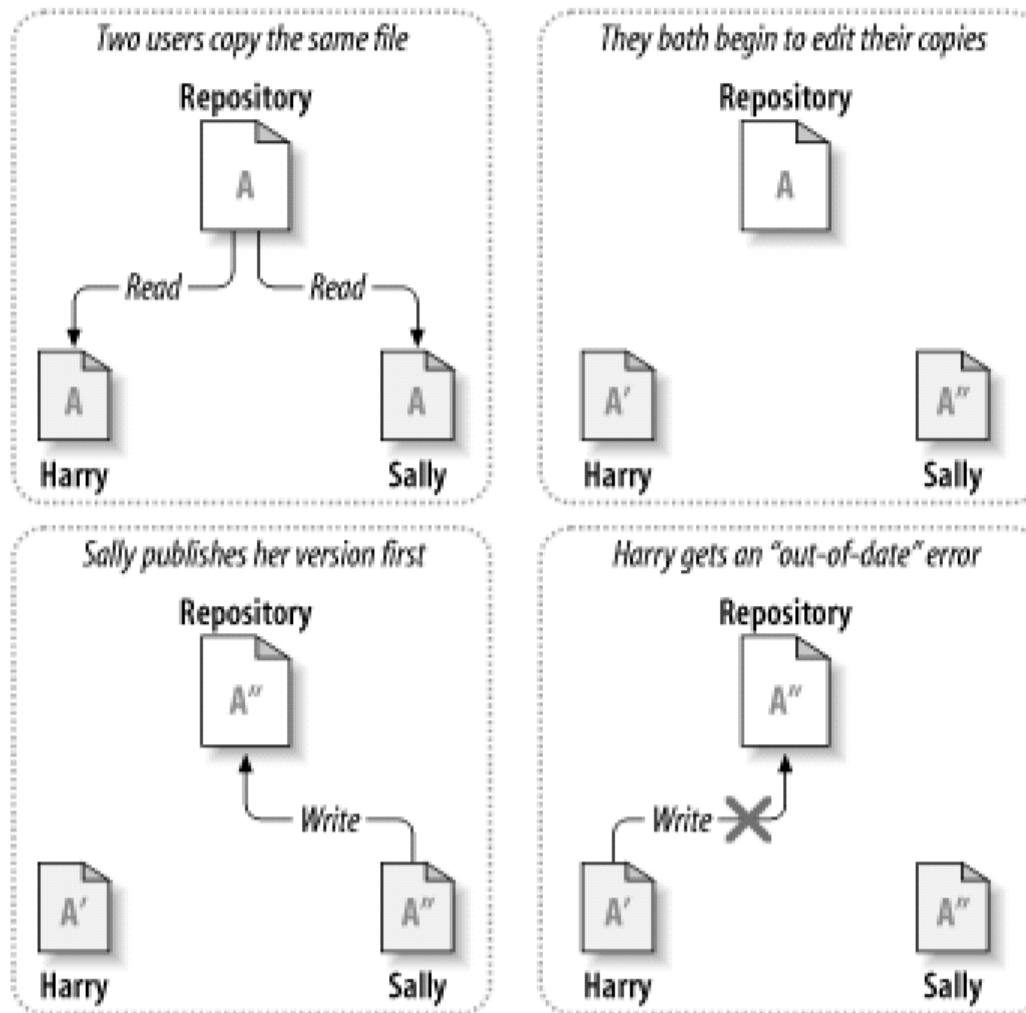
Practical problems of locking model?



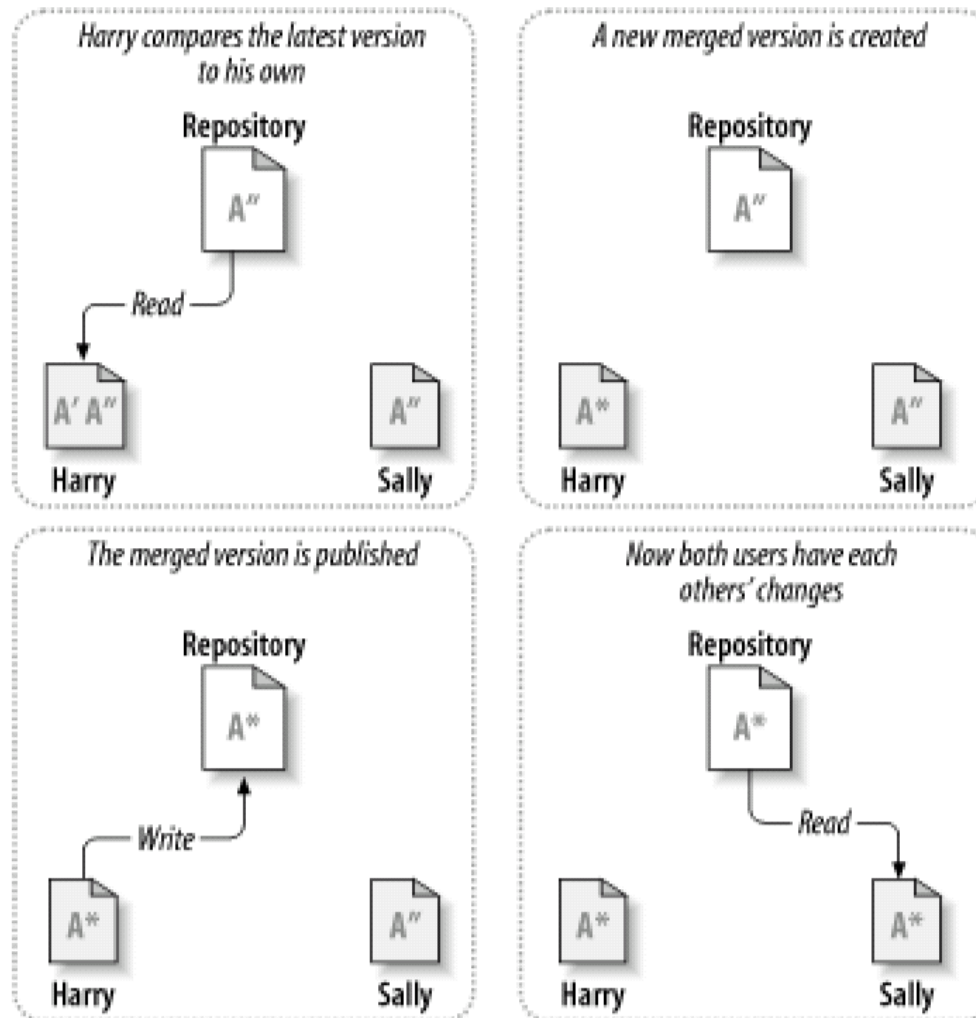
# Locking Problems

- How to lock?
  - Central system vs announcement on mailing list
  - Forgetting to unlock common
- Unnecessary sequentializing
  - Cannot work on different concepts in same file
- False sense of security
  - Changing dependant files can cause conflicts not prevented by locking

# Merging (1/2)



# Merging (2/2)



# Example

```
import java.util.LinkedList;
public class Stack<T> implements Cloneable {
    private LinkedList<T> items = new LinkedList<T>();
    public void push(T item) {
        items.addFirst(item);
    }
    public T pop() {
        if(items.size() > 0) return items.removeFirst();
        else return null;
    }
}
```

# Example

```
import java.util.LinkedList;
public class Stack<T> implements Cloneable {
    private LinkedList<T> items = new LinkedList<T>();
    public void push(T item) {
        items.addFirst(item);
    }
}
```

Edit 1

```
import java.util.LinkedList;
public class Stack<T>
    implements Cloneable {
    private LinkedList<T> items =
        new LinkedList<T>();
    public void push(T item) {
        items.addFirst(item);
    }
    public int size() {
        return items.size();
    }
    public T pop() {
        if(items.size() > 0) return
            items.removeFirst();
        else return null;
    }
}
```

Edit 2

```
import java.util.LinkedList;
public class Stack<T>
    implements Cloneable {
    private LinkedList<T> items =
        new LinkedList<T>();
    public void push(T item) {
        items.addFirst(item);
    }
    public T top() {
        return items.getFirst();
    }
    public T pop() {
        if(items.size() > 0) return
            items.removeFirst();
        else return null;
    }
}
```

# Example

```
import java.util.LinkedList;
public class Stack<T>
    implements Cloneable {
    private LinkedList<T> items =
        new LinkedList<T>();
    public void push(T item) {
        items.addFirst(item);
    }
    public int size() {
        return items.size();
    }
    public T pop() {
        if(items.size() > 0)
            return items.removeFirst();
        else return null;
    }
}
```



```
import java.util.LinkedList;
public class Stack<T>
    implements Cloneable {
    private LinkedList<T> items =
        new LinkedList<T>();
    public void push(T item) {
        items.addFirst(item);
    }
    public T pop() {
        if(items.size() > 0)
            return items.removeFirst();
        else return null;
    }
}
```



```
import java.util.LinkedList;
public class Stack<T> implements Cloneable {
    private LinkedList<T> items = new LinkedList<T>();
    public void push(T item) {
        items.addFirst(item);
    }
    public T top() {
        return items.getFirst();
    }
    =====
    public int size() {
        return items.size();
    }
    >>>>>> Size/Stack.java
    public T pop() {
        if(items.size() > 0) return items.removeFirst();
        else return null;
    }
}
```

<<<<<< Top/Stack.java

>>>>>> Size/Stack.java

System cannot decide order

# 3-way merge

- File changed in two ways
  - Overlapping changes -> conflicts
  - Merge combines non-conflicting changes from both
- Merging not always automatic
  - diff tool to show changes
  - Manual resolution of conflicts during merge (potentially requires additional communication)
- Automatic merge potentially dangerous
  - > syntactic notion of conflicts
- Merging of binary files difficult
- In practice: most merges are conflict free

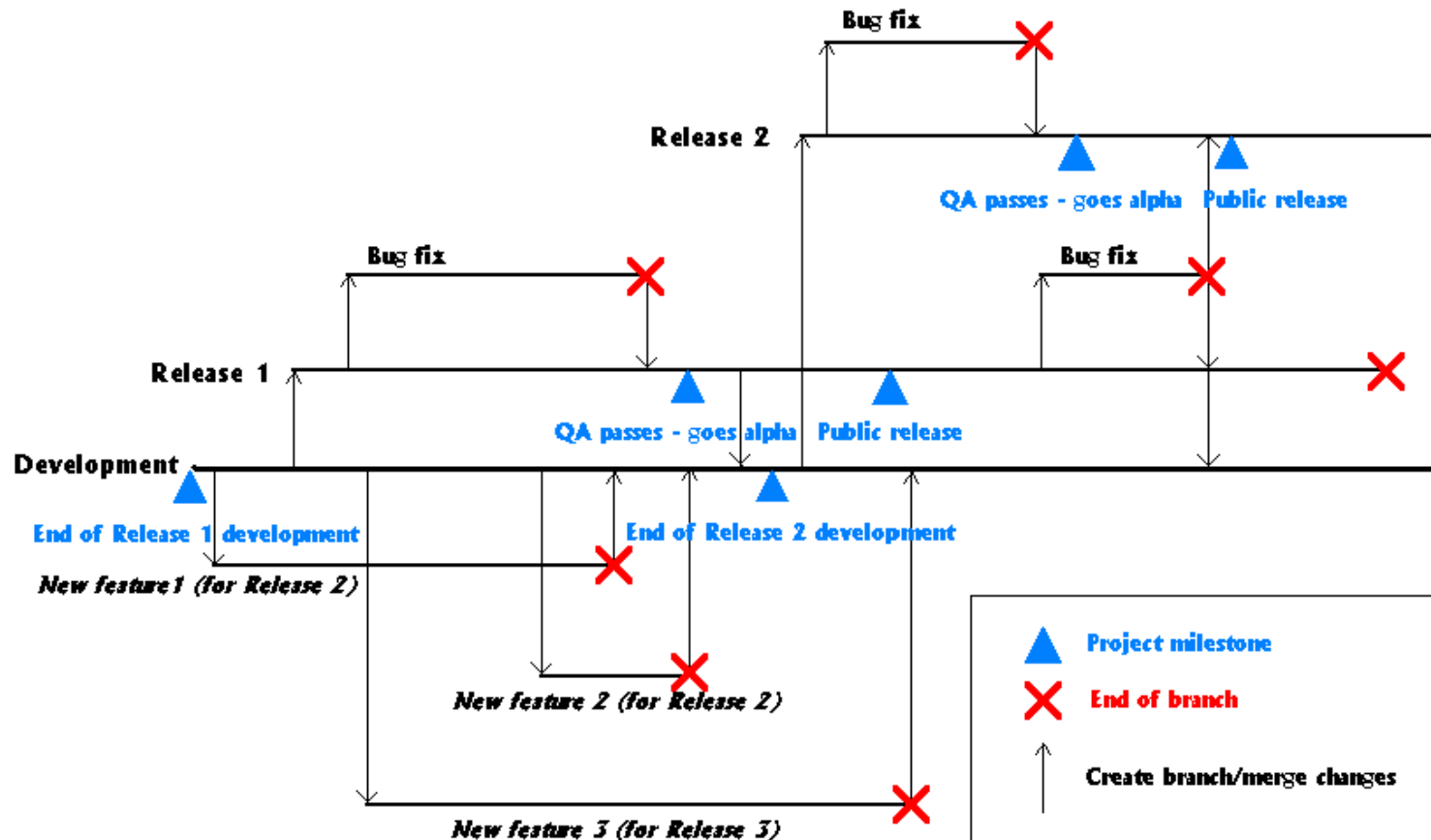


# BRANCHING

# Branching

- Parallel copies of the source tree
- Can be changed independently, versioned separately, and merged later (or left separate)
- Often used for exploratory changes or to isolate development activities
- Many usage patterns, common:
  - Main branch for maintenance OR main development
  - New branches for experimental features; merge when successful
  - New branches for nontrivial maintenance work
  - Branches for maintenance of old versions

# Release management with branches



# Variants and Revisions

- **Revision** replaces prior revision (temporal)
- **Variant** coexists with other variants
- **Version** describes both
- **Release**: Published and named version

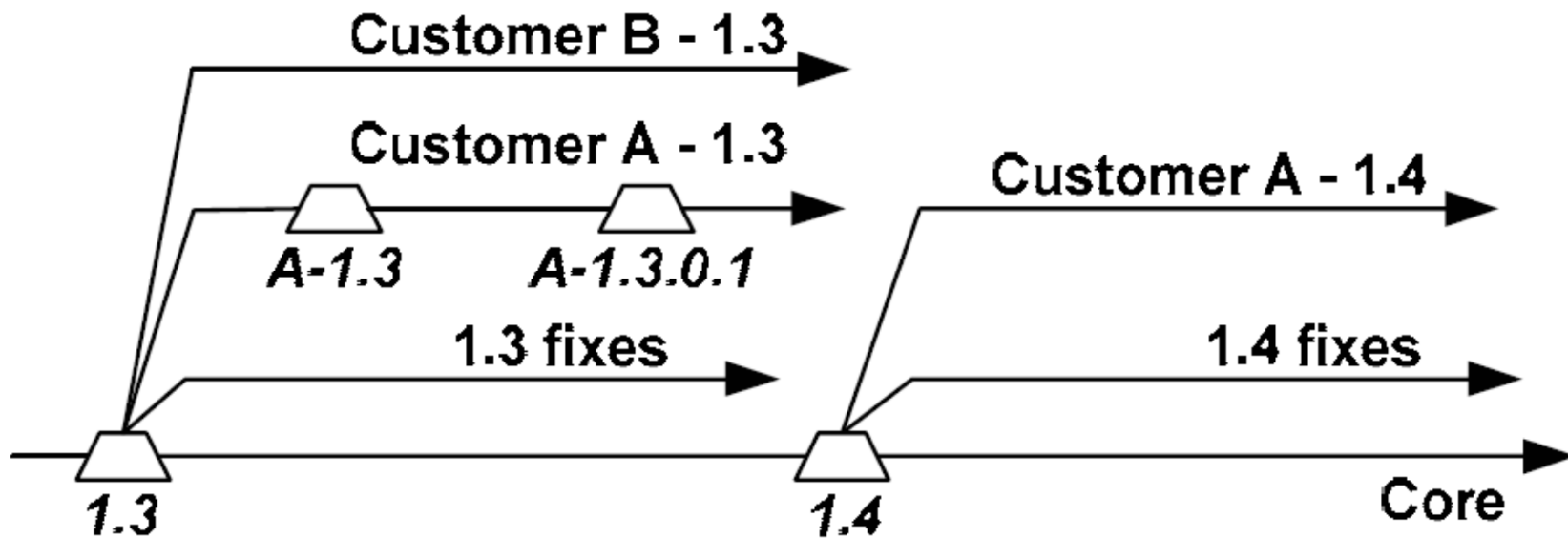
	V1.0	V1.1	V2.0	V3.0
Base system (Windows)	X	X	X	X
Linux variant		X	X	
Server variant			X	X
Extension for customer A		X	X	X
Extension for customer B				X

# Semantic Versioning for Releases

- Given a version number MAJOR.MINOR.PATCH, increment the:
  - MAJOR version when you make incompatible API changes,
  - MINOR version when you add functionality in a backwards-compatible manner, and
  - PATCH version when you make backwards-compatible bug fixes.
- Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

**<http://semver.org/>**

# Variants and Revisions



[Staples&Hill, APSEC'04]

# Managing variants

- Branching for variants does not scale well
- Requires special planning or tooling
- Many solutions
  - Configuration files
  - Preprocessors
  - Build systems
  - DSLs
  - Software product lines
  - ...

```
/* common parts */  
...  
/* dependent on operating system */  
#if (OS == Unix)  
...  
#elif (OS == VMS)  
...  
#else  
...  
#endif  
...
```

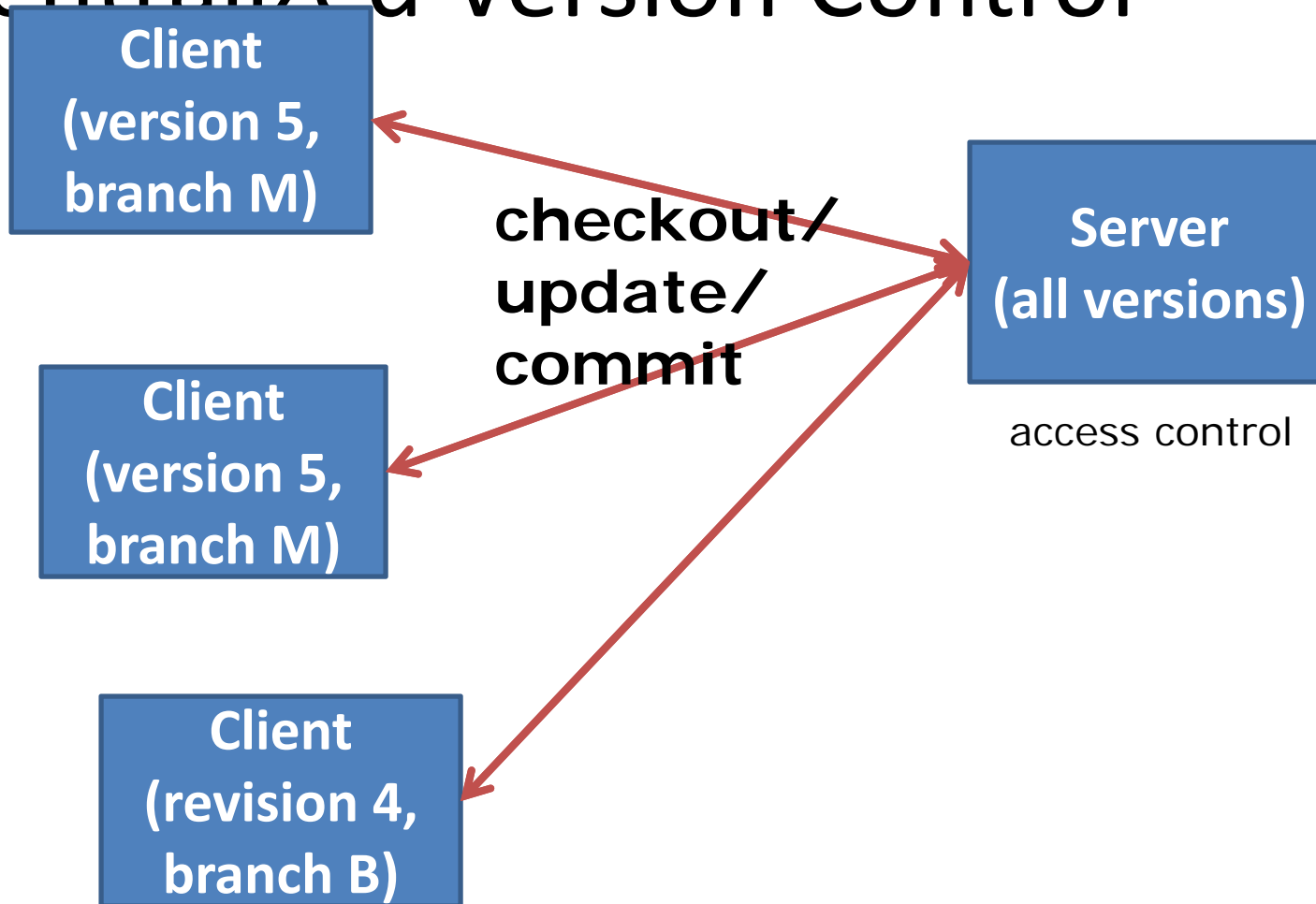
# **CENTRALIZED VERSION CONTROL (E.G., SVN)**



# Classes of version control systems

- Systems supporting merging and/or locking
- Local version control
  - Local history of files: SCCS (1970s), RCS (1982)
- Central version control
  - Versions stored on central master server
  - Clients synchronize with server (update, commit)
  - CVS (1990), SVN (2004), Perforce, Visual SourceSafe
- Distributed version control
  - Many repositories; synchronization among repositories (push, pull)
  - Git (2005), Mercurial, Bitkeeper, ClearCase

# Centralized Version Control



# Typical work cycle

- Once: Create local workspace
  - svn checkout

---

- Update workspace:
  - svn update
- Perform changes in workspace:
  - svn add
  - svn delete
  - svn copy
  - svn move
- Show workspace changes:
  - svn status
  - svn diff
- Revert changes in workspace:
  - svn revert
- Update and merge conflicts:
  - svn update
  - svn resolved
- Push workspace changes to server:
  - svn commit

# CVS vs. SVN

## CVS

- Improvement over RCS in tracking entire directories
- Revision number per file
- Text files (binary files possible)

## SVN

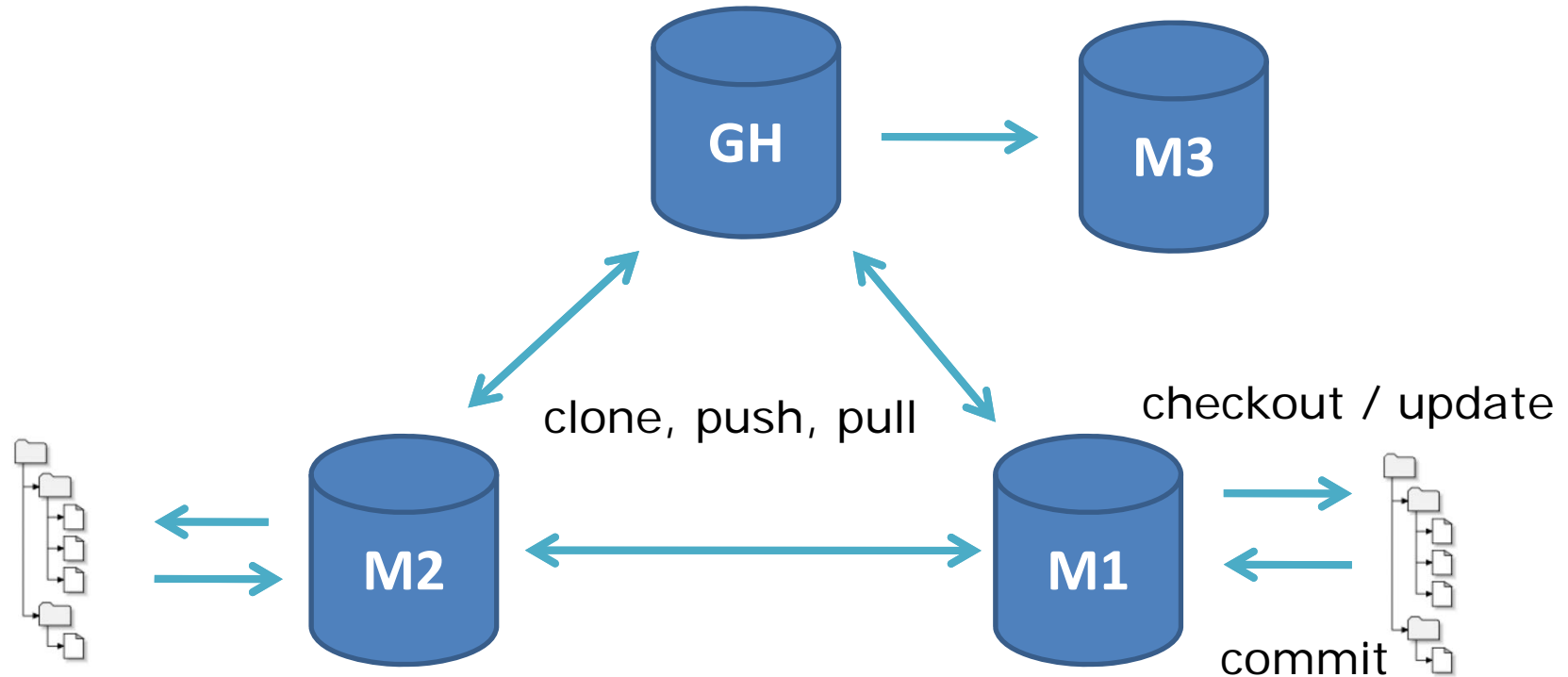
- Revision numbers for project
- Atomic commits (committing multiple files at once)
- Tracking files and directories
- Support renaming
- Tracking of Metadata

# **DISTRIBUTED VERSION CONTROL (E.G., GIT)**

# Git

- Distributed version control
- No central server necessary (but possible)
- Local copies of repositories (containing all history)
  - Locally SVN like functionality: checkout, update, commit, branch, diff
- Nonlinear development: each local copy can evolve independently
- Synchronization among repositories (push/fetch/pull)
- Fast local operations (branch, commit, diff, ...)

# Overview

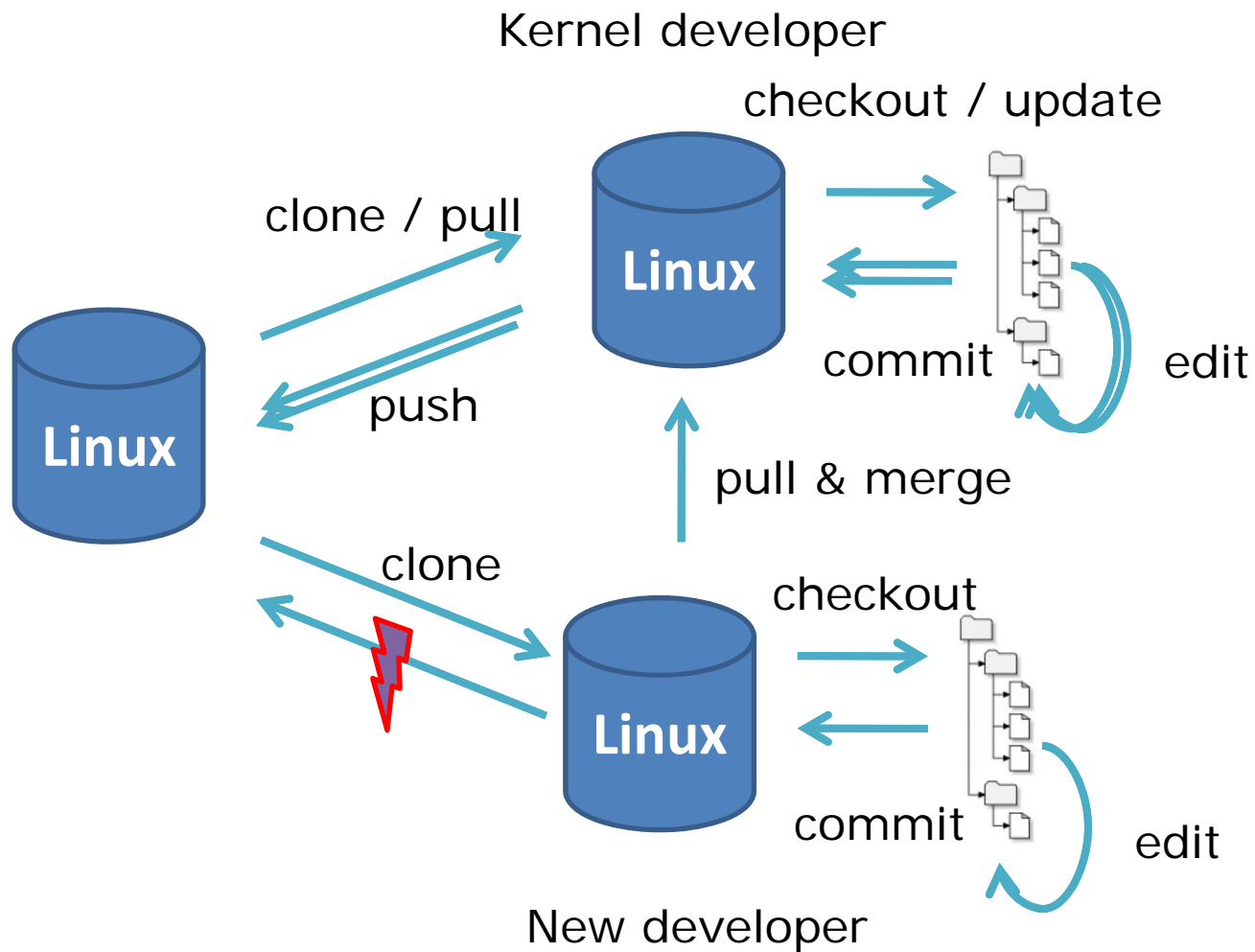


# Distributed Versions

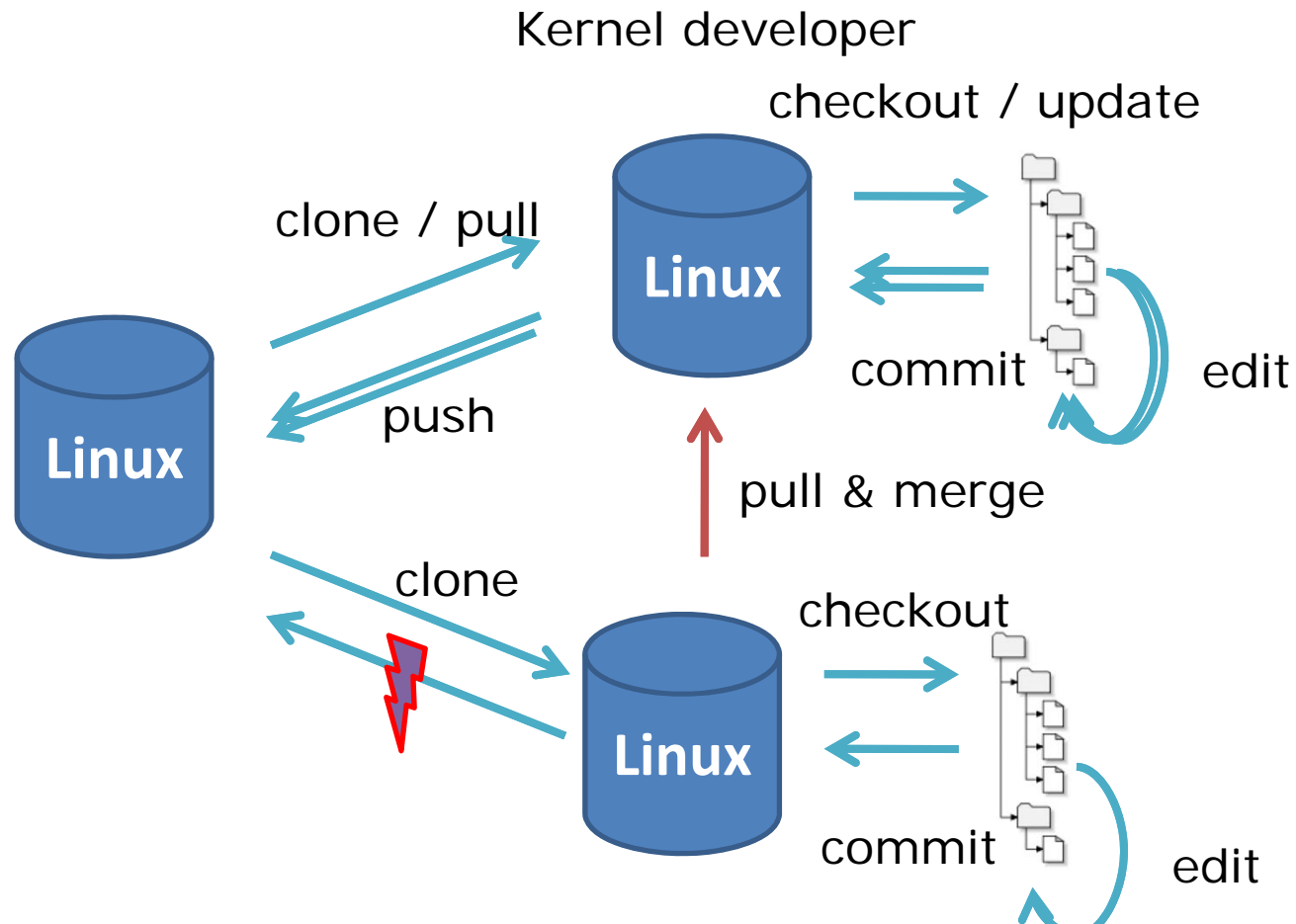
- Versions not globally coordinated/sorted
- Unique IDs through hashes, relationships tracked in successor graph
  - e.g., 52a0ff44aba8599f43a5d821c421af316cb7305
  - Possible to merge select changes (cherry picking)
  - Possible to rewrite the history as long as not shared remotely (git rebase etc)
- Cloning creates copy of repository (including all versions)
  - Tracks latest state when cloned, relevant for updating and merging
  - Normal checkout and commit operations locally
  - Commits don't change original repository
- Fetch and pull get missing versions from remote repository (one or more)
- Push operations sends local changes to remote repository (one or more), given access rights



# Example workflow

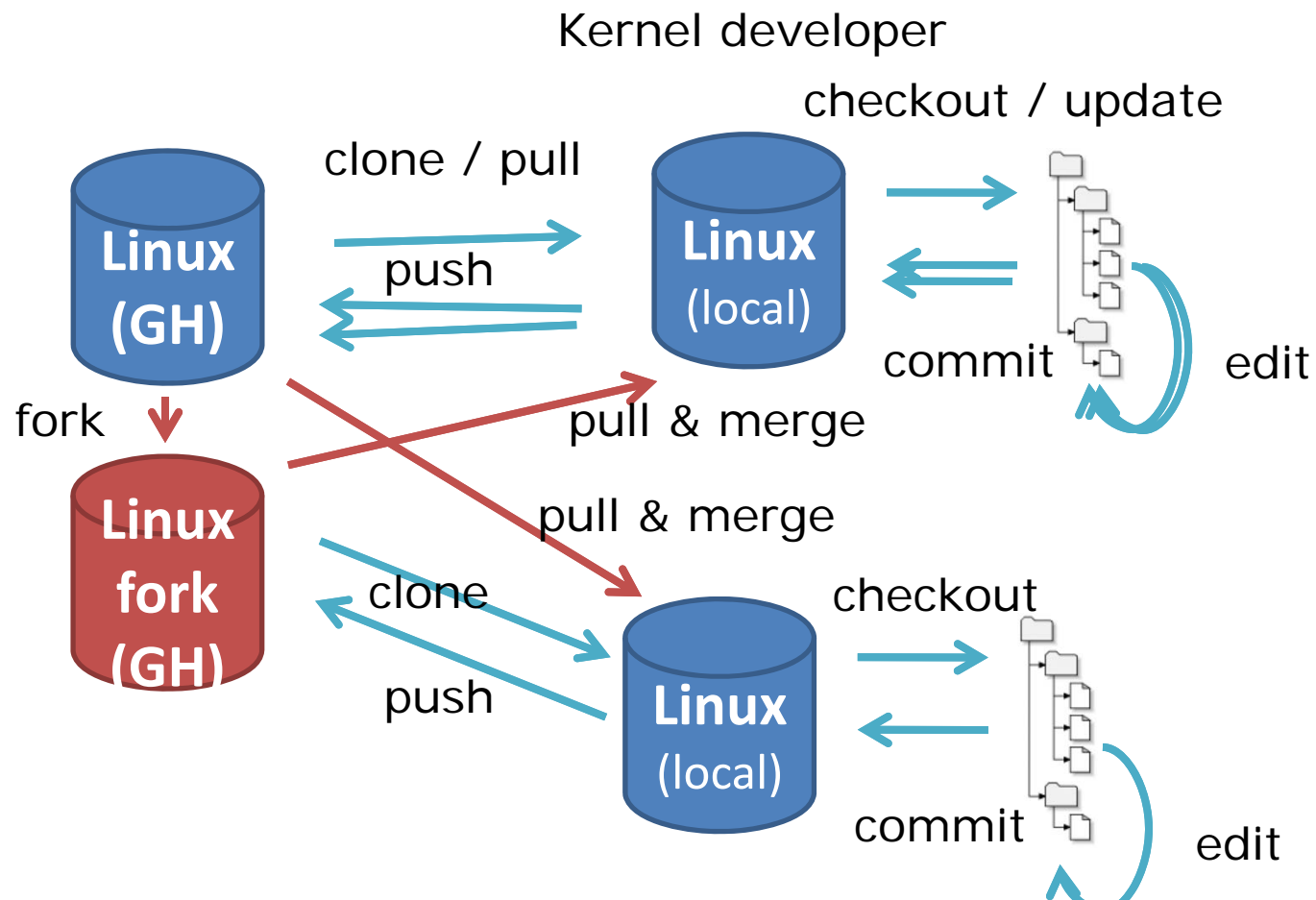


# Pull Requests



Pull request: Github feature to ask developer to pull a specific change (alternative to sending email); integration with Travis CI

# Forks



Fork: Github feature to clone repository on Github (own copy with full rights)

# Forks

Kernel developer

checkout / update

**Caution:**  
**Please to not fork 214 repositories.**

214 Collaboration Policy: "Here are some examples of behavior that are inappropriate: Making your work publicly available in a way that other students (current or future) can access your solutions, even if others' access is accidental or incidental to your goals."

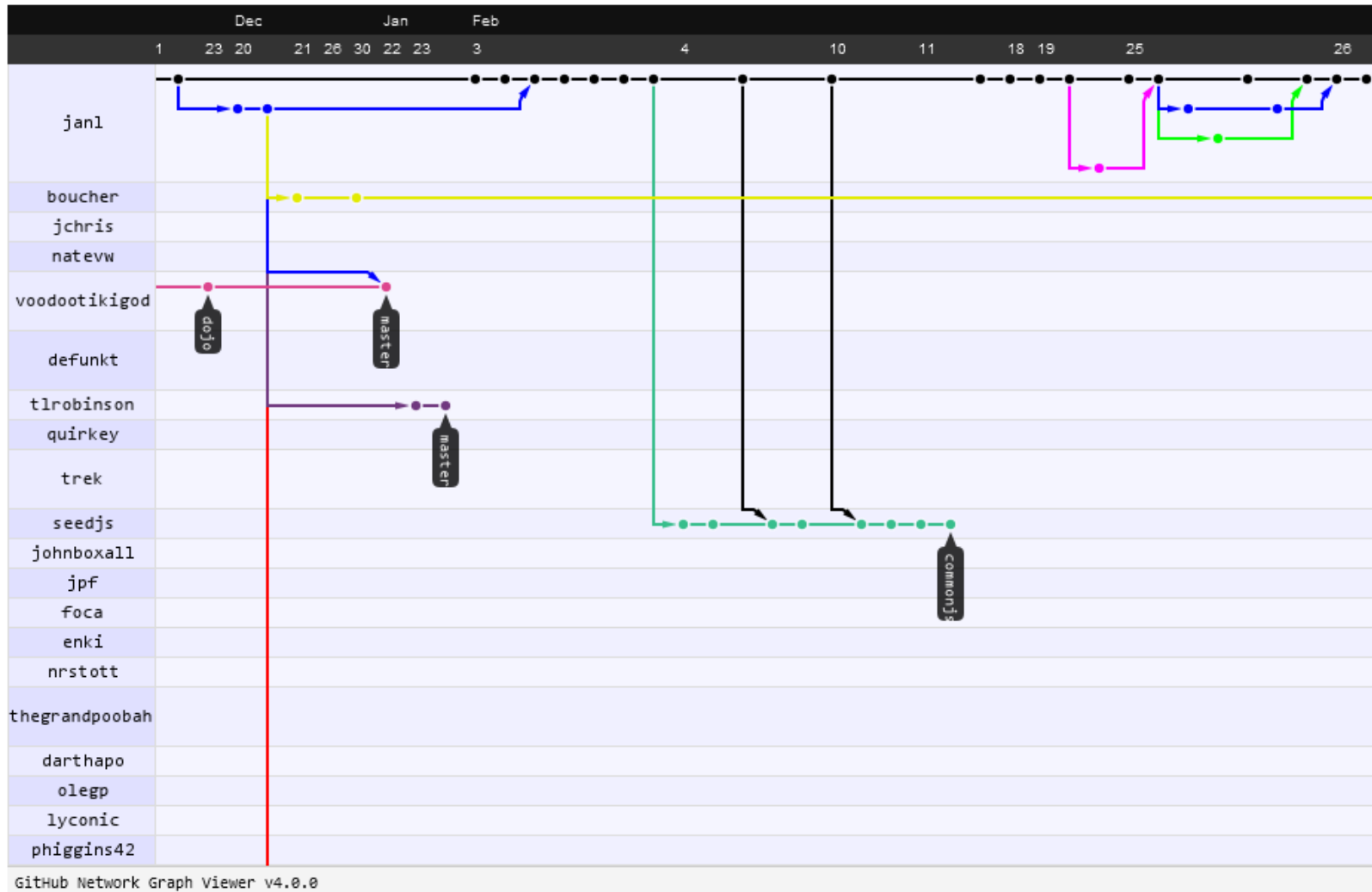
(local)

commit

edit

Fork: Github feature to clone repository on Github (own copy with full rights)

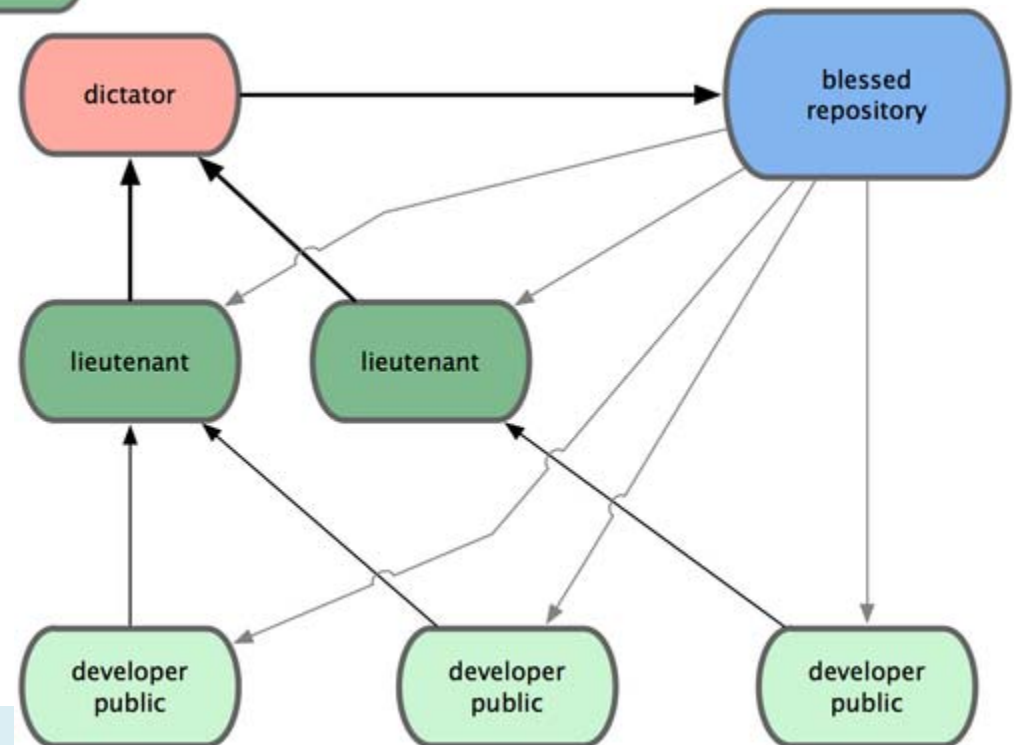
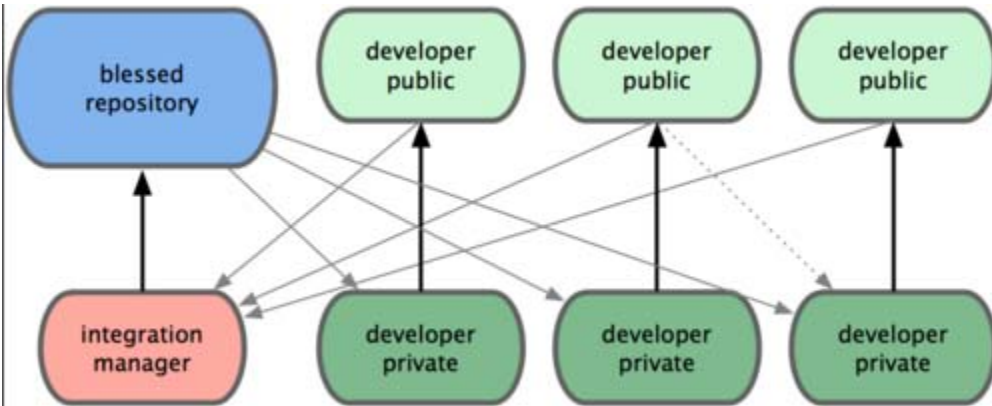
# Repositories in mustache.js



# Git History

```
Terminal — less — 156x42
sh  bash  less
* 2009-12-08 575c2d8 bchesneau@gmail.com (bchesneau@gmail.com) Merge remote branch 'thomo/master' into mthomo
* 2009-12-07 a315955 Thomas.Mohaupt@gmail.com (Thomas.Mohaupt@gmail.com) Merge branch 'master' of git://github.com/couchapp/couchapp
* 2009-12-01 cbd2bc8 Thomas.Mohaupt@gmail.com (Thomas.Mohaupt@gmail.com) Bug: Trailing carriage returns are not stripped.
* 2009-11-28 c649863 Thomas.Mohaupt@gmail.com (Thomas.Mohaupt@gmail.com) Merge branch 'master' of git://github.com/couchapp/couchapp
* 2009-11-27 1418fa1 Thomas.Mohaupt@gmail.com (Thomas.Mohaupt@gmail.com) Fix wrong couchapp path
* 2009-11-27 1944cc0 Thomas.Mohaupt@gmail.com (Thomas.Mohaupt@gmail.com) Merge branch 'master' of git://github.com/couchapp/couchapp
* 2009-11-27 ae97905 Thomas.Mohaupt@gmail.com (Thomas.Mohaupt@gmail.com) Fix UTF-8 issue on WinXP
* 2009-11-26 97b5026 Thomas.Mohaupt@gmail.com (Thomas.Mohaupt@gmail.com) Merge branch 'master' of git://github.com/couchapp/couchapp
* 2009-11-19 00ed4d9 Thomas.Mohaupt@gmail.com (Thomas.Mohaupt@gmail.com) Fix WinXP problem: Attachment name is not UTF-8 encoded
* 2009-11-17 6a52137 mot@sq.s.int (mot@sq.s.int) Fix: USERPROFILE handling
* 2009-12-08 34828d0 bchesneau@gmail.com (bchesneau@gmail.com) fix hooks and compress hook
* 2009-12-08 60dec82 bchesneau@gmail.com (bchesneau@gmail.com) fix compress options search path
* 2009-12-07 3c89e38 jasondavies@apache.org (jasondavies@apache.org) Generate attachment signatures before processing macros.
* 2009-12-07 54c3e83 jasondavies@apache.org (jasondavies@apache.org) Merge branch 'master' of git://github.com/couchapp/couchapp
* 2009-12-04 9b97679 bchesneau@gmail.com (bchesneau@gmail.com) fix __iter__
* 2009-12-04 4b8c9e7 bchesneau@gmail.com (bchesneau@gmail.com) fix typo. thanks to markh.
* 2009-12-04 b37230e bchesneau@gmail.com (bchesneau@gmail.com) patch from @rmg. thanks!
* 2009-11-29 233238a bchesneau@gmail.com (bchesneau@gmail.com) bump version number to 0.5.1.
* 2009-11-29 d8c9709 bchesneau@gmail.com (bchesneau@gmail.com) add update template to generators
* 2009-11-28 0964b63 bchesneau@gmail.com (bchesneau@gmail.com) couchapp standalone for macosx via py2app
* 2009-11-28 4def0a6 bchesneau@gmail.com (bchesneau@gmail.com) make sure ocntent type is defined
* 2009-11-28 10b53e3 benoitc@.(none) (benoitc@.(none)) new fixes while testing
* 2009-11-28 051e778 bchesneau@gmail.com (bchesneau@gmail.com) rethink hook system. final version. Now like extensions each hooktype is a list of
* 2009-11-28 c47718f bchesneau@gmail.com (bchesneau@gmail.com) rethink extensions. So now extensions are a list of key=value pair, where key is t
* 2009-11-27 53a2040 bchesneau@gmail.com (bchesneau@gmail.com) fix compress extension & load extensions like we load hooks
* 2009-11-27 343862d bchesneau@gmail.com (bchesneau@gmail.com) fix template paths with windows
* 2009-11-27 bfa2f8c bchesneau@gmail.com (bchesneau@gmail.com) backport import_module from python 2.7
* 2009-11-27 3380a4c benoitc@.(none) (benoitc@.(none)) more windows fix
* 2009-11-27 a049b41 bchesneau@gmail.com (bchesneau@gmail.com) more fixes
```

# Git and Central Repositories



© Scott Chacon "Pro Git"

# Social Coding

The screenshot shows a web browser window displaying the GitHub profile of 'janl' (Jan Lehnardt). The browser's address bar shows 'http://github.com/janl'. The profile header includes a profile picture, the name 'janl (Jan Lehnardt)', and buttons for 'Message' and 'Unfollow'. Below the header, the user's name 'Jan Lehnardt' and email 'jan@apache.org' are listed. Statistics show 92 public repositories and 229 followers. The 'Public Repositories' section lists three repositories: 'mustache.js' (JavaScript, 619 stars, 65 forks), 'talks' (3 stars, 1 fork), and 'konamibacon.js' (JavaScript, 2 stars, 1 fork). The 'Public Activity' section shows recent pushes and issue closures.

**Profile Information:**

- Name: Jan Lehnardt
- Email: jan@apache.org
- Website/Blog: http://jan.prima.de/plok/
- Location: Berlin, Germany
- Member Since: May 23, 2008

**Statistics:**

- 92 public repos
- 229 followers

**Public Repositories (92):**

- mustache.js** (JavaScript, 619 stars, 65 forks)  
Minimal templating with {{mustaches}} in JavaScript  
Last updated 2 days ago
- talks** (3 stars, 1 fork)  
my talks  
Last updated 3 days ago
- konamibacon.js** (JavaScript, 2 stars, 1 fork)  
what it says on the tin  
Last updated 5 days ago

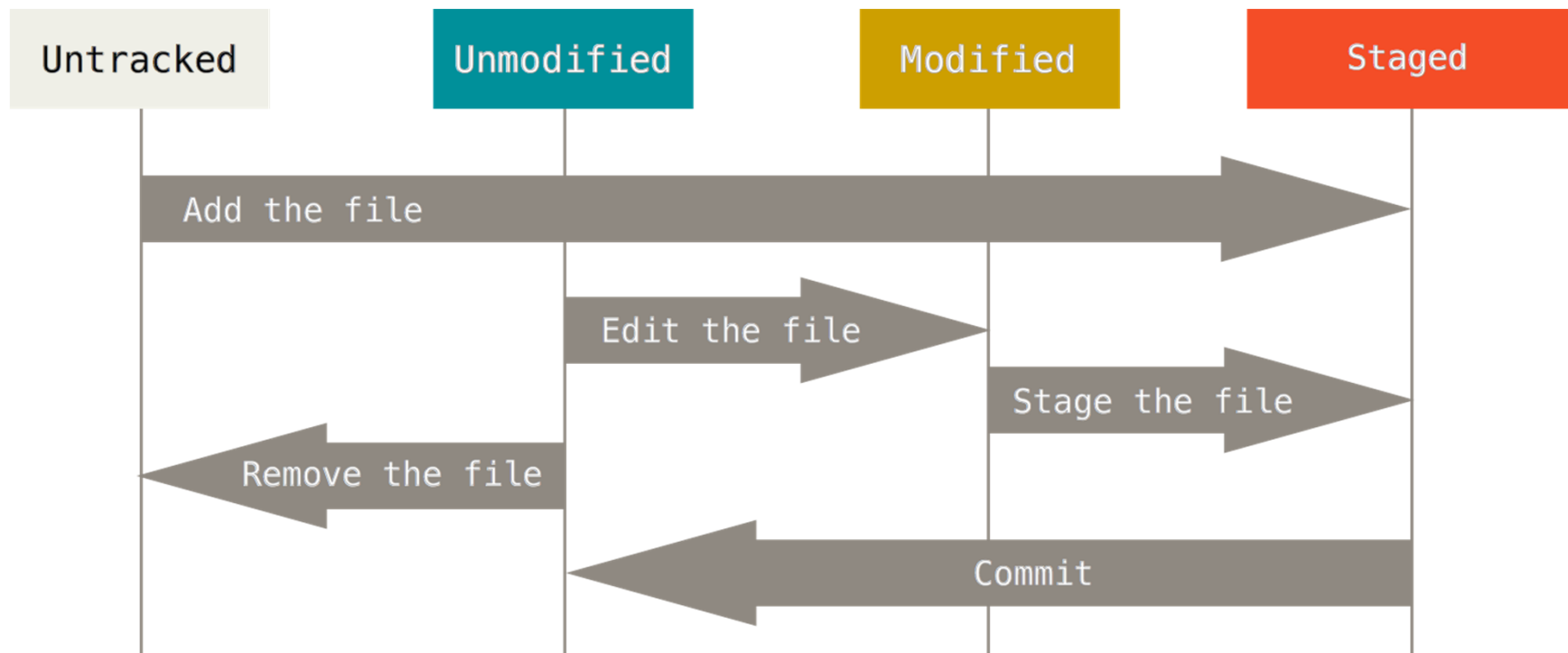
**Public Activity:**

- janl pushed to gh-pages at oreilly/couchdb-guide about 13 hours ago  
5e3505e fix listing, thanks to damjan
- janl started following daleharvey 1 day ago  
daleharvey has 12 public repos and 29 followers.
- janl closed issue 294 on oreilly/couchdb-guide 2 days ago  
Confusion in tour.html
- janl pushed to gh-pages at oreilly/couchdb-guide 2 days ago  
d209771 Clarify note, fix markup. Closes #294.
- janl pushed to master at janl/mustache.js 2 days ago  
e4a7966 More FAQurate  
c6524cc spelling

Awareness/  
News Feeds

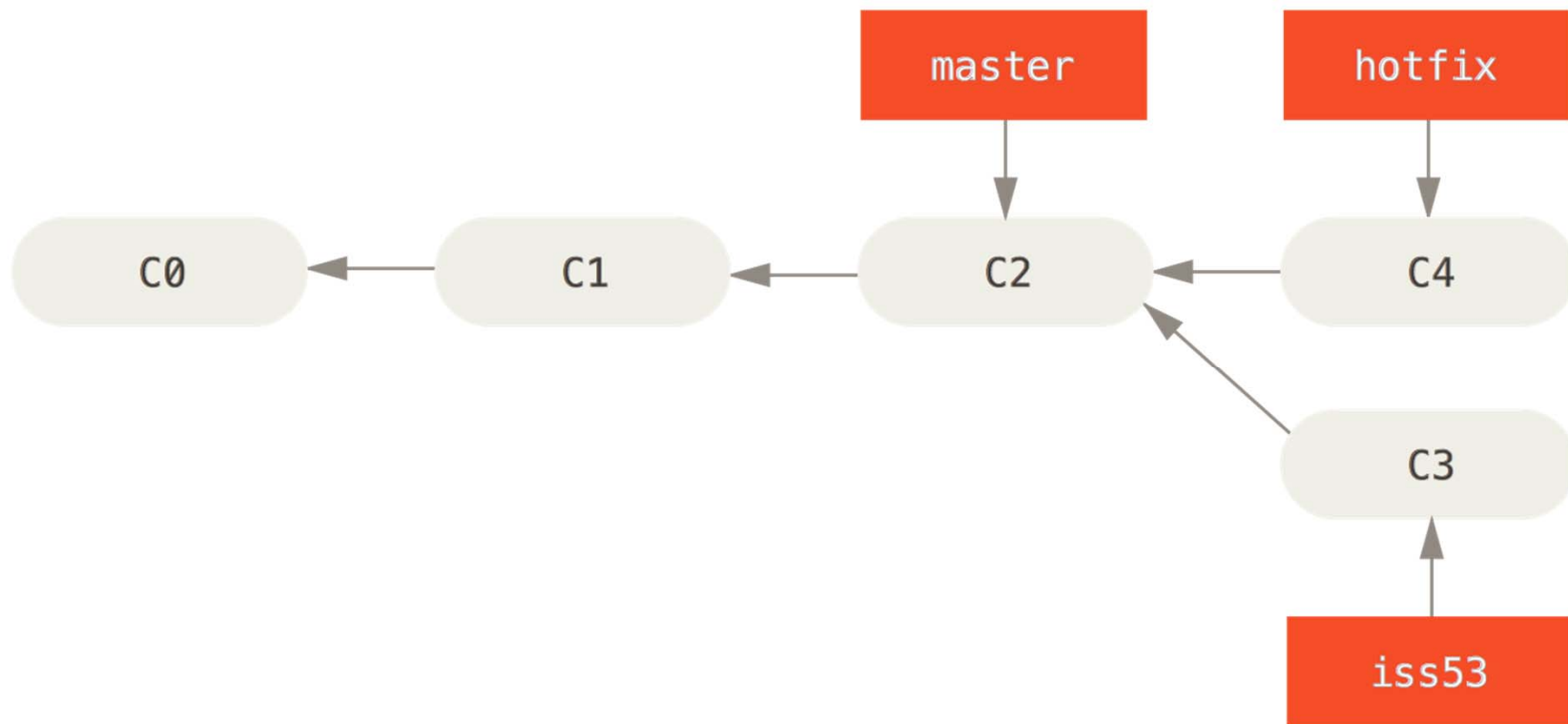


# Git Internals



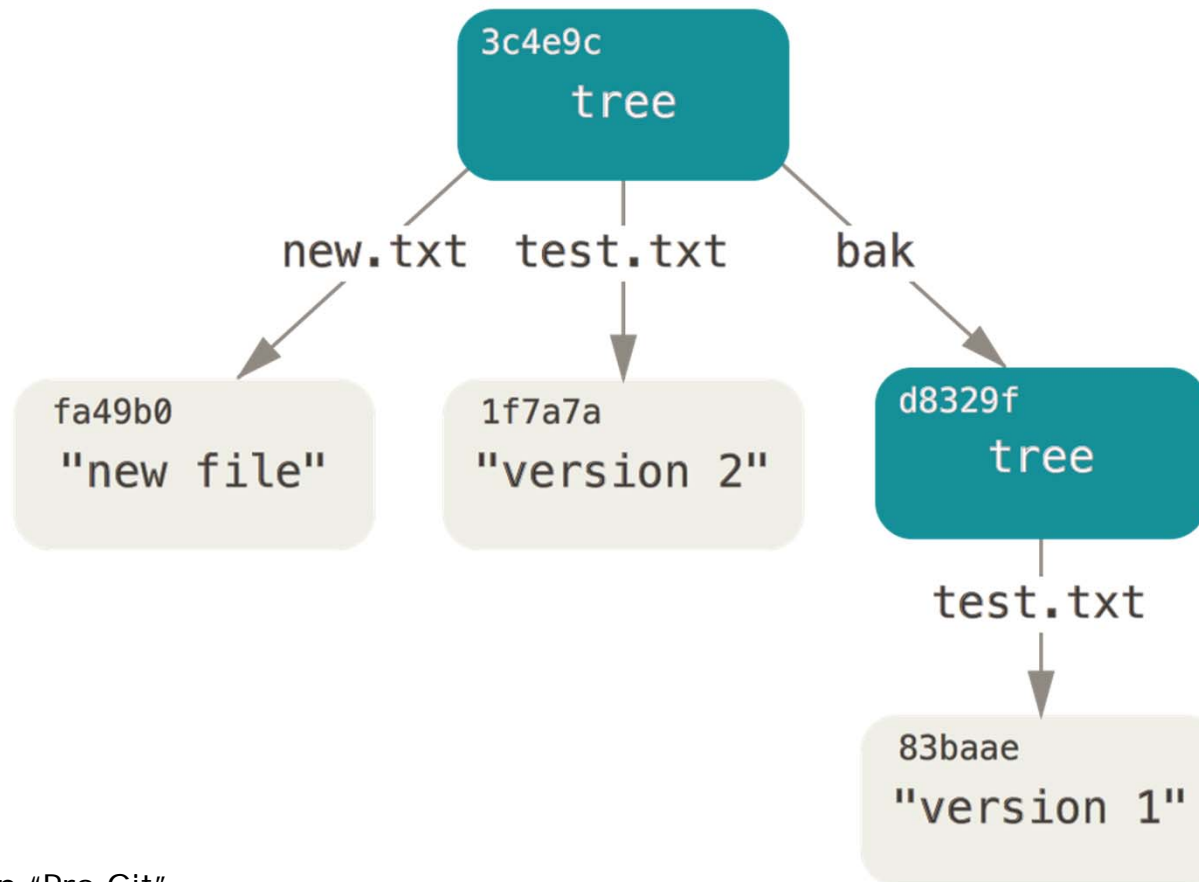
© Scott Chacon "Pro Git"

# Git Internals



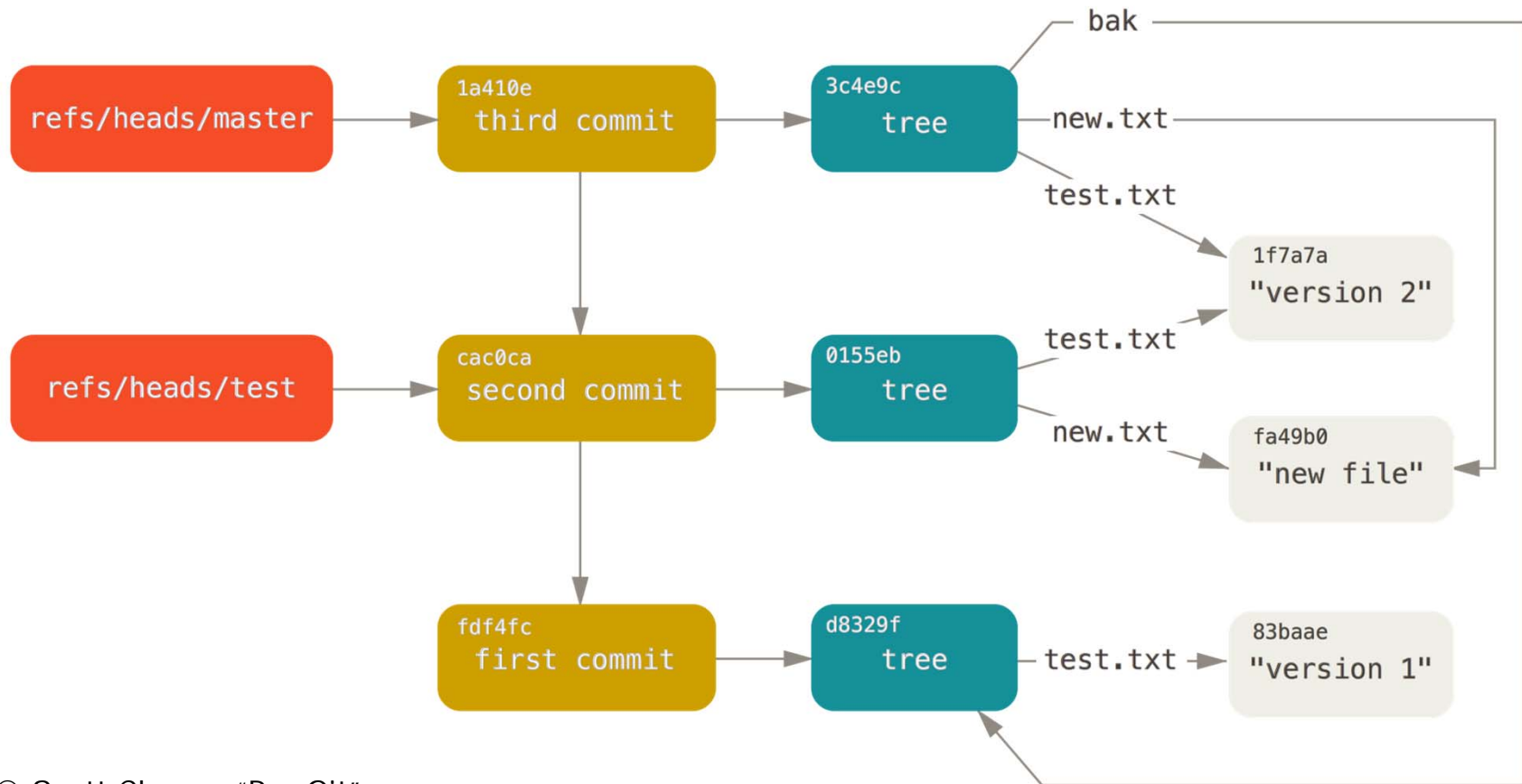
© Scott Chacon "Pro Git"

# Git Internals



© Scott Chacon "Pro Git"

# Git Internals



© Scott Chacon "Pro Git"

# Summary

- Version control has many advantages
  - History, traceability, versioning
  - Collaborative and parallel development
- Locking vs. merging and merge conflicts
- Collaboration with branches
- From local to central to distributed version control