

Version Control

The Plan

- Why version control?
- Concepts/Goals of VCS
- Systems

The Plan

- **Why version control?**
- **Concepts/Goals of VCS**
- **Systems**

Why Version Control?

- Sanity!
- Allow many people to collaboratively work
- Roll back changes if they don't work out
 - Encourage experimentation
- “What changed?!”
- “When did this break?” (blame)
- Keep code elsewhere (in case of disaster)

The Plan

- Why version control?
- **Concepts/Goals of VCS**
- Systems

Goals

- Required:
 - Keep all previous copies of code
 - Give different people different places to work
 - Connect these back together later
- Nice:
 - Credit contributors
 - Mark “release” versions

Concepts

- Versions
- Branches
- Tagging
- Merging

Concepts

- Versions
 - Snapshot of a point in time
- Branches
- Tagging
- Merging

Concepts

- Versions
- Branches
 - Personal space to work in
- Tagging
- Merging

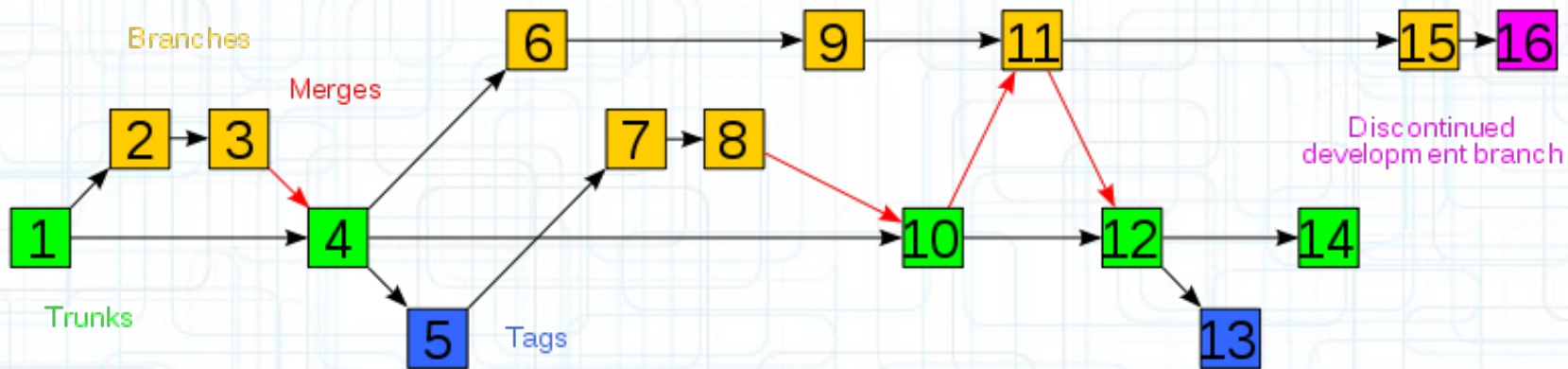
Concepts

- Versions
- Branches
- Tagging
 - Mark a version (ex: Release-1.0)
- Merging

Concepts

- Versions
- Branches
- Tagging
- Merging
 - Mush two branches together

Visualization



Systems

- Terminology
- Git
- Subversion

Terminology

- Repository (repo)
- Centralized version control
- Distributed version control

Terminology

- Repository (repo)
- Centralized version control
 - All operations hit a central server
- Distributed version control

Terminology

- Repository (repo)
- Centralized version control
- Distributed version control
 - Keep a local copy of the repository
 - Operations work against this

Warning – Religious Wars Ahead!

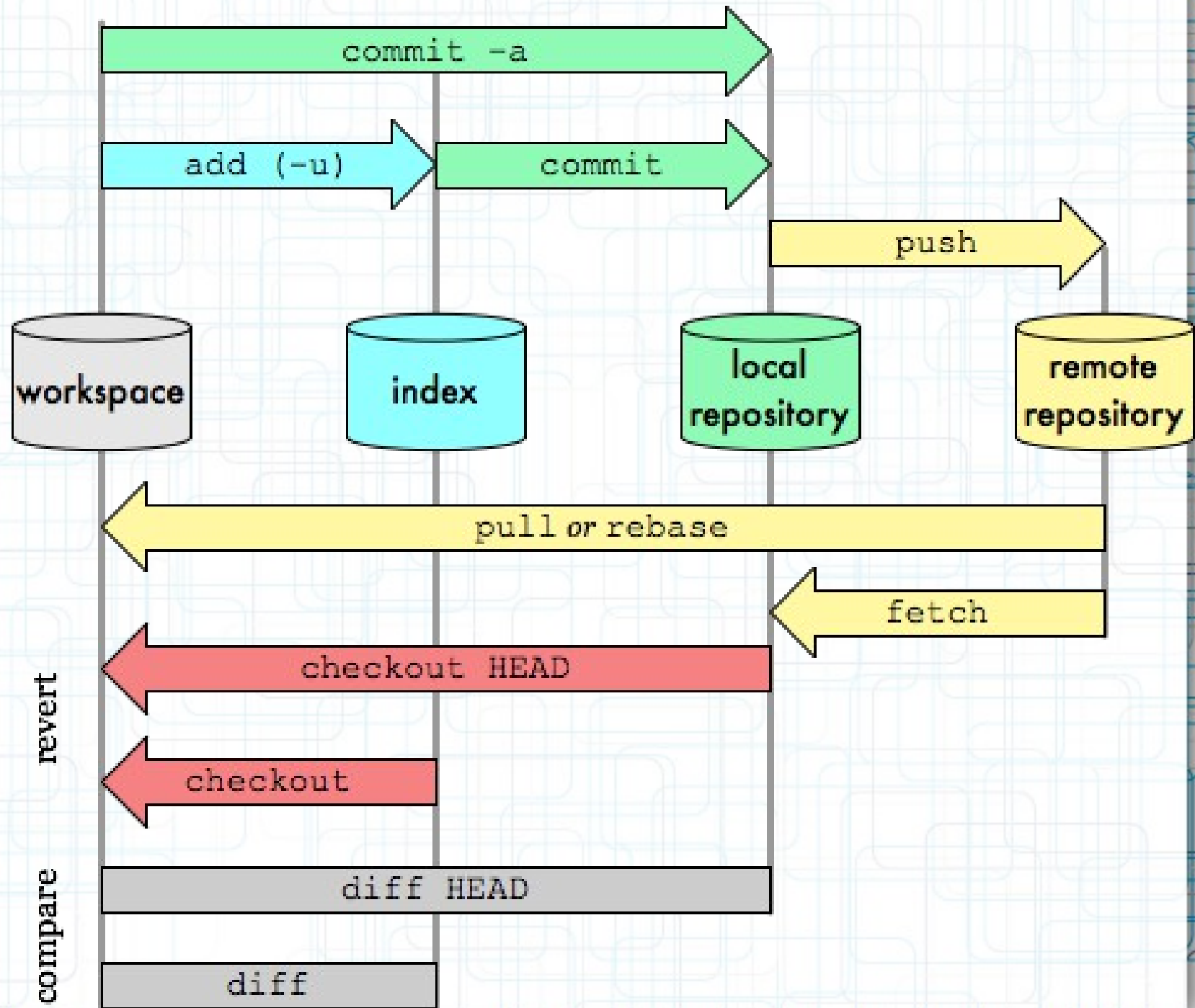


Git – The “Stupid Content Tracker”

- "I'm an egotistical bastard, and I name all my projects after myself. First 'Linux', now 'git'."
 - Linus Torvalds
- Decentralized VCS
- Tracks by folder

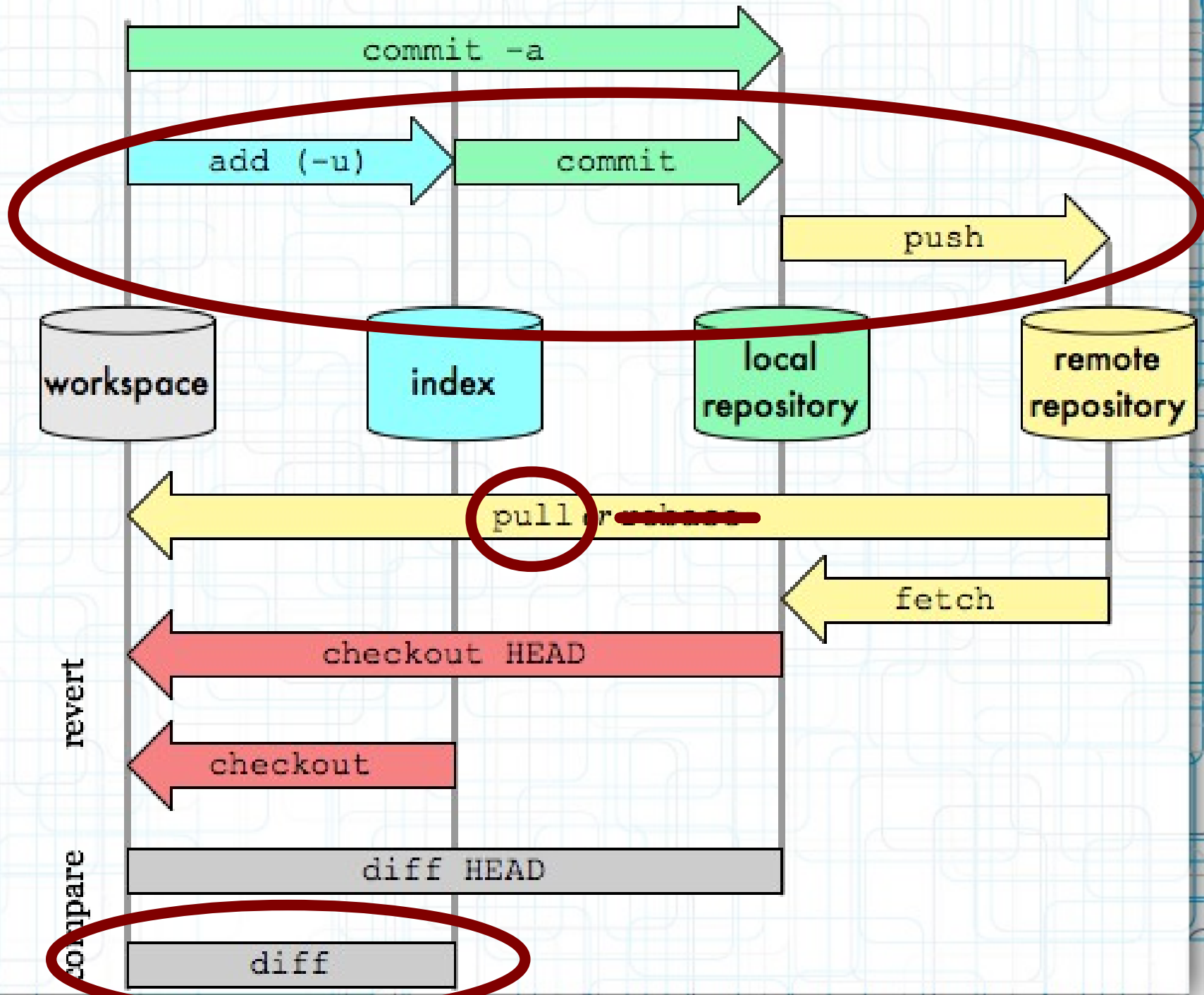
Git Data Transport Commands

<http://osteele.com>



Git Data Transport Commands

<http://osteele.com>



Subversion

- Centralized VCS
- Simpler (most of the time)
- Tracks by file

Ignoring files

- `.gitignore`
 - Put in a line for each file to ignore (you can use regexps)
 - <https://github.com/github/gitignore>
- `svn:ignore`
 - `svn propset svn:ignore secret.txt .`
 - (this is complicated...)

Some Dos

- Check in/commit early and often
- Use descriptive commit messages

Some Donts

- Check in binary executeables
 - VCSs in general don't know how to handle it
- Check in/push broken code to the master (you'll owe your team booze)
 - Bad etiquette
 - Exception: on your own branch
- Check in passwords, local configs, keys
- **USE DROPBOX/EMAIL AS A VCS**

Getting Repos

- SVN: Talk to CS
- Git: Run locally
- Git: github.com
- Both: ACM!

Comments/Questions/Stories?

Slides online at

[http://pressers.name/static/slidesets/ACMVC
S-Oct12.pdf](http://pressers.name/static/slidesets/ACMVC
S-Oct12.pdf)

[http://pressers.name/static/slidesets/ACMVC
S-Oct12.odp](http://pressers.name/static/slidesets/ACMVC
S-Oct12.odp)

