

DEVELOPMENT BEST PRACTICES

Managing Your Drupal Project with

Git

Colan Schwartz

DrupalCamp Toronto, July 14th 2013

Licensed under Creative Commons Attribution 2.0 Generic

Who Am I?

- Hello, I'm Colan Schwartz, colans.net
- Username "*colan*" (<http://drupal.org/user/58704>)
- Doing Drupal since 2005 (4.6!)
- Maintain a lot of modules (e.g. Computed Field)
- Freelance Technical Architect
- IM / IA / Tech Arch / DevOps / Back-End
- Been using Git before d.o's Great Git Migration

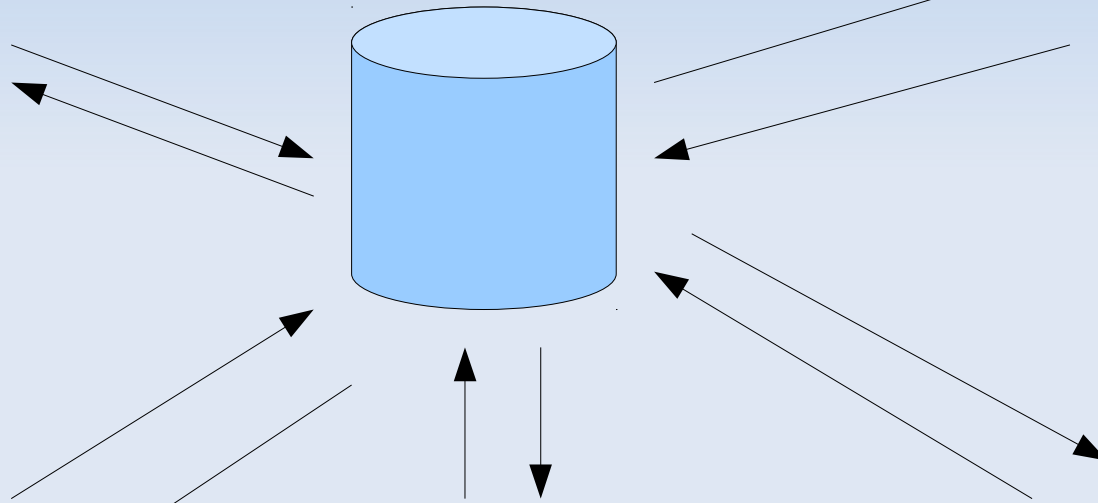
History

- Gen 0: Diff & Patch
- Gen 1: RCS
- Gen 2: CVS
- Gen 3: SVN
- Gen 4: Git, Bzr, Mercurial
 - Distributed => more flexibility

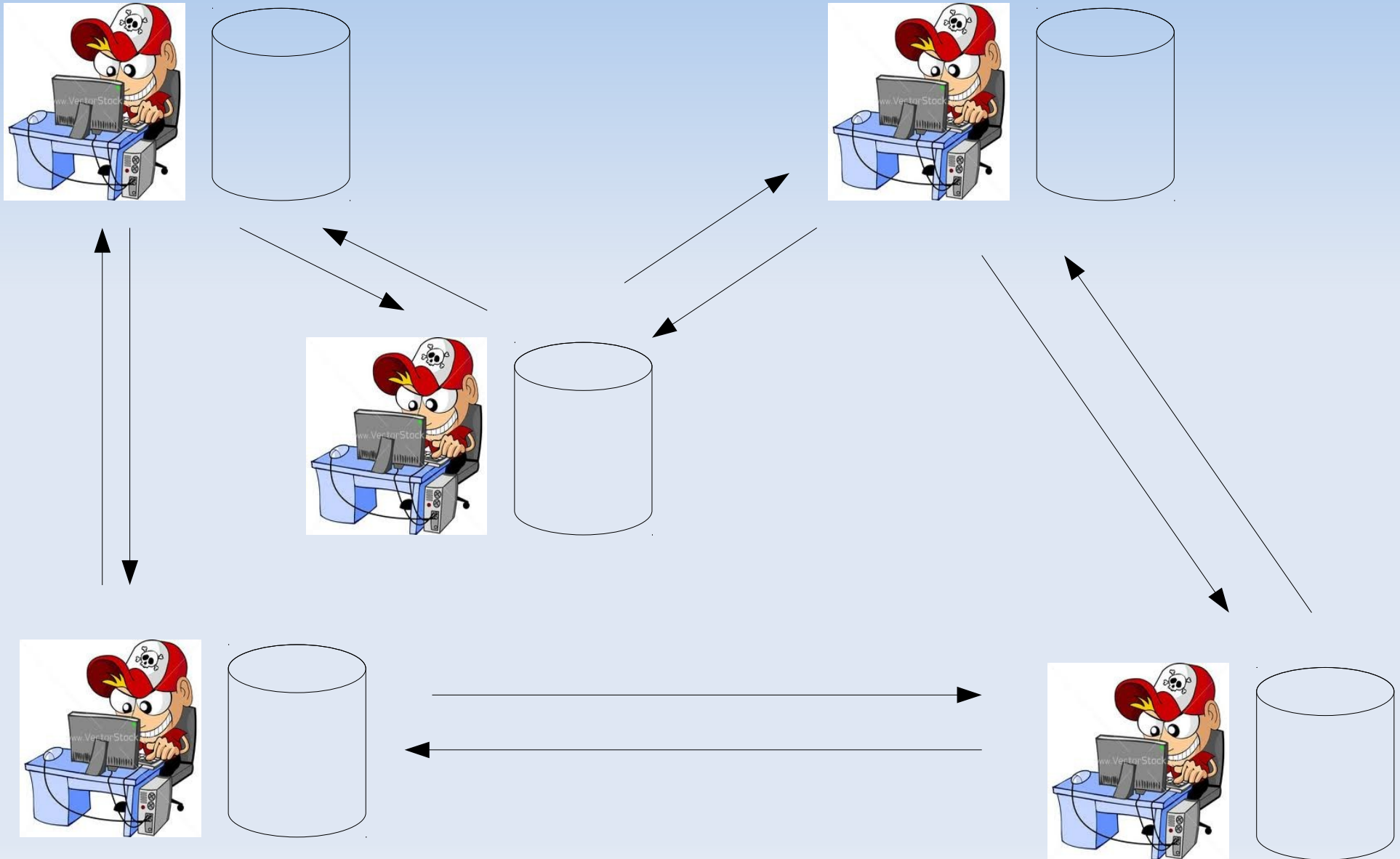
Git

- Git is slowly becoming the *de facto* VCS
- Drupal.org recently switched to Git from CVS
- Handles branching & tagging more cheaply
- Can handle renames efficiently, unlike Subversion
- Always does things correctly, unlike BZR & Mercurial *citation needed (à la Wikipedia)*
- Works with the (best) issue tracker Redmine/Chiliproject
- Drawback: Doesn't have many front-end tools

The Old Way



The New Way



Hosting Options

- Bitbucket, Gitorius, etc.
- GitHub is the most popular
- GitLab is the open-source alternative
- Self-hosting
 - Consumes few resources on a VM
 - Give devs git-shell access only
 - Use Gitolite for granular permissions
- Poll!
- Others?

Problems

- Conventional approach has some issues
- Can't track upstream history (~~git-log~~)
- No explicit portable recipe for your site
- Can't audit your developers' changes without reviewing all commits
 - Too easy to commit where they shouldn't

Solutions?

- Drupal-specific solutions
 - Drush makefiles
 - ?
- Non-Drupal-specific solutions
 - Git submodules
 - MR with Drupal plug-in:
wimleers.com/article/mr-drupal
 - ?

Drush Makefiles

- What?
- Pros
 - Explicitly state how your site is built
 - Prevents your devs from committing willy-nilly
 - Provides a portable recipe
- Cons
 - Takes time to build sites
 - Usually dependent on network
- Poll!

Git Submodules

- Git can track a commit in an external repository as a "submodule" in a subdirectory.
- Pros
 - External projects are recognized as such
 - Fits well with Drupal's contributed modules
 - Easy to track where you are relative to upstream
- Cons
 - Adds complexity
 - More Git craziness for your devs to understand
- Poll!

References & Useful Links

- Official documentation
 - <http://git-scm.com/documentation>
- Drupal.org's Git documentation
 - <http://drupal.org/documentation/git>
- Books:
 - Pragmatic Version Control Using Git
 - Travis Swicegood
 - Many others!
- Help on IRC
 - [#drupal-gitsupport](#) on [irc.freenode.net](#)

Thank you!

- Questions?
- Comments?
- Feedback?
- Other approaches?
- Discussion?