MAXIMIZING VERSION CONTROL

Manage your Drupal project with Git submodules

Colan Schwartz

DrupalCamp Montréal, October 26th 2013

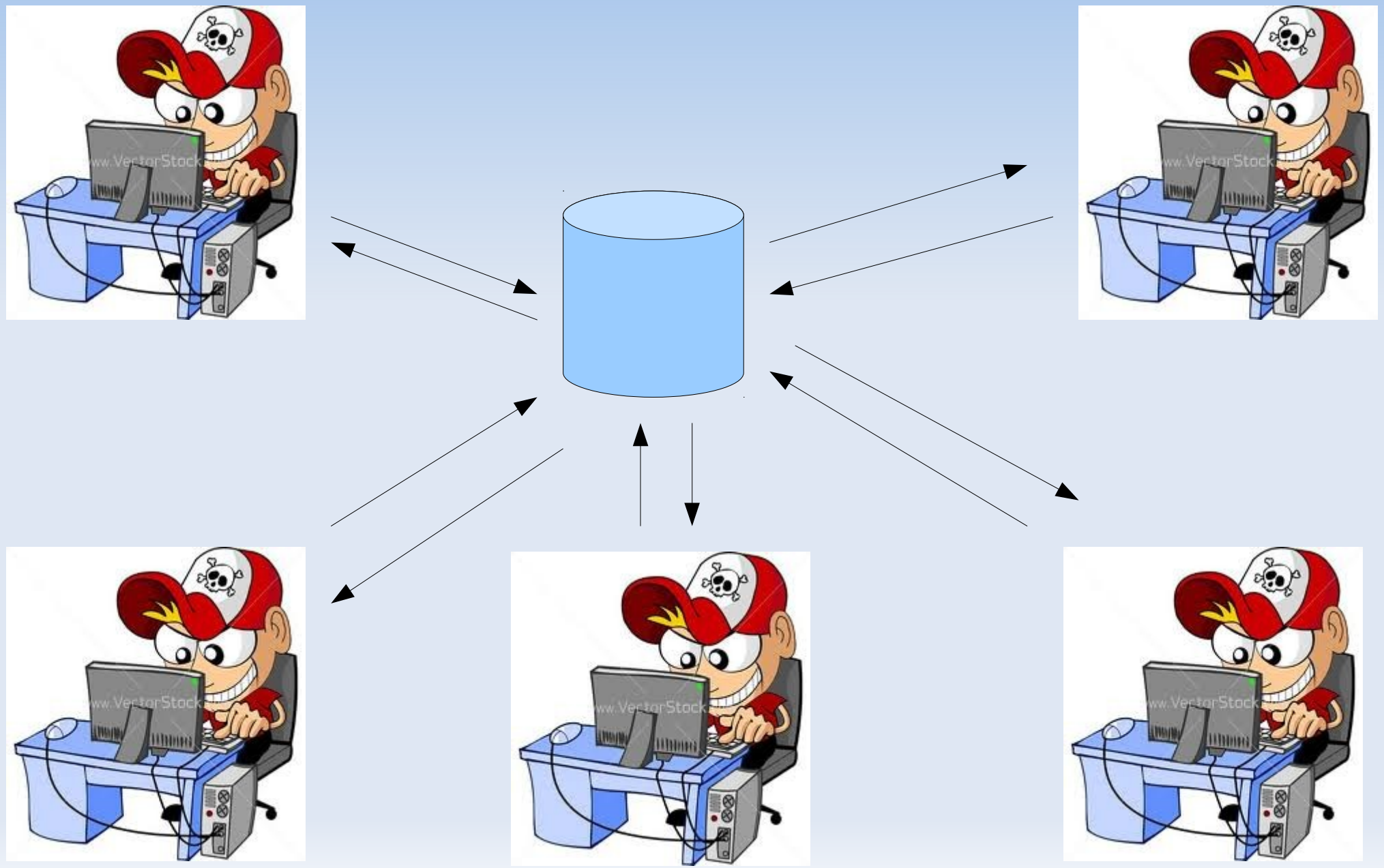
Who Am I?

- Hello, I'm Colan Schwartz, colans.net
- Username *"colan"* (http://drupal.org/user/58704)
- Doing Drupal since 2006 (4.6!)
- Maintain a lot of modules (e.g. Computed Field)
- Senior Freelance Web Technical Architect
- IM / IA / Tech Arch / DevOps / Back-End
- Been using Git before d.o's Great Git Migration

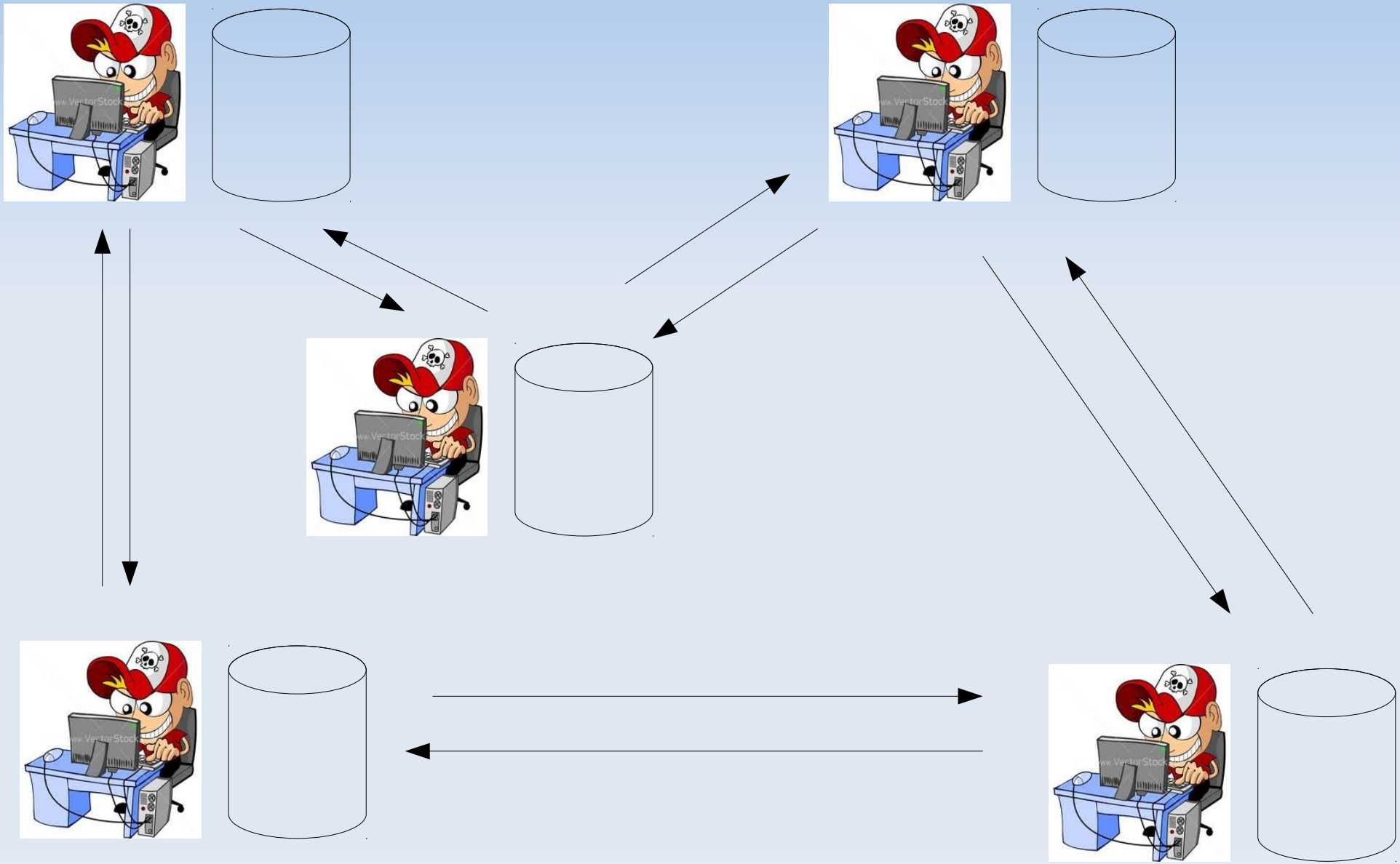
History

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

The Old Way



The New Way



Git's Popularity

- Git is slowly becoming the *de facto* VCS
- According to Ohloh, Git has almost 3K users, ~2K more than Bazaar or Mercurial
- Drupal.org recently switched to Git from CVS
- Now you're using it.
- But *how* are you using it?

Hosting Options

- GitHub: pay for private repos
- Bitbucket: pay for > 5 users, but works with Jira
- GitLab is the open-source alternative
- Self-hosting (without GitLab)
  - Git repos consume few resources on a VM
  - Give devs git-shell access only
  - Use Gitolite for granular permissions without SSH
  - Integrates well with Redmine

One Git Repository

1. Download the latest Drupal tarball & git init.
2. Configure your settings.php.
3. Download your modules/themes to sites/all.
4. To upgrade anything:
    1. Download latest tarball / drush dl.
    2. Commit it.
    3. Cherry-pick custom changes to reapply*.

** Easy to forget & can be difficult to find.*

Poll?

Alternative: Drush Makefiles

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

Problems with single Git repo

- Can't track upstream history (~~git log~~)
- Can't slurp in module changes without downloading entire modules
- Necessary to reapply custom changes.

Git Submodules

- Git can track a commit in an external repository as a ”submodule” in a subdirectory.
- Pros
  - External projects are recogized 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!**

Setting up the repo

- git clone -b 7.x http://git.drupal.org/project/drupal.git myproj
- cd myproj
- git remote rename origin drupal
- git checkout -b develop drupal/7.x
- git merge 7.23
- ssh git.example.com git init –bare /var/local/git/myproj.git
- git remote add origin ssh://git.example.com/var/local/git/myproj.git
- git push origin develop

Adding a contrib module

- git submodule add -b 7.x-2.x http://git.drupal.org/project/git_deploy.git sites/all/modules/contrib/git_deploy
- git submodule init sites/all/modules/contrib/git_deploy
- cd sites/all/modules/contrib/git_deploy
- git checkout 7.x-2.1; cd ..
- *Or replace all of the above with a single Drush command*
  - You need the Drush patch from https://drupal.org/node/1372442#comment-6099598
  - drush dl –select --package-handler=git_drupalorg –gitsubmodule git_deploy
- Then add, commit & push as usual.

Upgrading core

- cd myproj
- git checkout develop
- git fetch drupal
- git merge 7.24
- git push origin develop

Upgrading a contrib module

- cd sites/all/modules/contrib/git_deploy
- git fetch
- git checkout 7.x-2.2
- cd ..
- git add git_deploy
- git commit -m "Updated Git Deploy 2.1 → 2.2"
- git push origin develop

Custom changes to contribs

- ssh git.example.com git init –bare /var/local/git/myviews.git
- cd sites/all/modules/contrib/views
- git remote rename origin drupal
- git remote add origin ssh://git.example.com/var/local/git/myviews.git
- git checkout -b develop drupal/7.x-3.x
- *Make changes & commit.*
- git push origin develop
- cd ..
- git add views; git commit; git push origin develop
- When no longer needed, remove origin & set back to d.o.

Deploying Your Project Code

- First time:
  - git clone ssh://git.example.com/myproj.git
  - cd myproj
  - git checkout 1.0
  - git submodule update –init --recursive
- Every other time:
  - cd myproj
  - git fetch
  - git checkout 2.3
  - git submodule update –init --recursive

What's new: running the git submodule command

References & Useful Links

- Links
  - Randy Fay's
    Drupal Deployment with Git Submodules
  - http://git-scm.com/book/en/Git-Tools-Submodules
  - Drupal.org's Building a Drupal site with Git (Help!)
- Books
  - Pragmatic Version Control Using Git
    - by Travis Swicegood
- Help on IRC
  - #drupal-gitsupport on irc.freenode.net

Thank you!

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