CVS is dead ...

... and centralised revision control
is *so* second millennium

Bryan O'Sullivan
bos@serpentine.com

Why leave CVS behind?

- Commits are not atomic
- Disconnected operation is impossible
- Branching and tagging are conflated and expensive
- File renaming, copying and removal are wrong

- Divides world into “haves” (committers) and “have nots” (everyone else)

Subversion, the heir apparent

- Intended to be a better CVS
- Fixes many of CVS's glaring flaws
- Still centralised
- Preserves the committer/everyone-else distinction
- Trivial operations when disconnected (edit, diff)
- Modular, but complex
  - Two back ends, two wire protocols

Centralised SCM is bad

- Non-committers are second-class citizens
- Can’t create history
- Tools won’t help with merges after renames
- Stuck with ad-hoc tools like quilt
- Share changes in form of GNU patches

Why go distributed?

- Everyone gets to be a committer
- P2P fu: share changes with anyone you want
  - Core / non-core committer distinction becomes a matter of convention
- Completely disconnected operation: branch, commit, merge on your laptop
- Massive replication buys reliability: no single point of failure

- CVS/SVN-like workflow is trivial to implement

What are your choices?

- Bazaar-NG
- Codeville
- Cogito/git
- Darcs
- GNU Arch
- Mercurial
- Monotone
- SVK

Five systems to watch

- Bazaar-NG: has Ubuntu fu
- Codeville: source of everyone's merge tech, not yet cooked
- Mercurial: fast, scalable, used by kernel hackers, quilt-like features
- Monotone: influential, several interesting ideas
- SVK: built atop Subversion, quite popular

Upgrading, stability, usage

- Some systems have native CVS import support
- Mercurial imports from Cogito/git, will export soon
- Tailor provides almost arbitrary movement of changes between SCM tools
- All systems self-host
- Nobody loses changes or corrupts history
- A few are regularly used on big, dynamic trees with complex histories

Workflow example

- Handful of anointed central repositories
- Maintain a pristine clone of each central repository you care about
  - Sporadically pull changes in from master
- Create an ephemeral branch for each logically distinct piece of work
- Push commits to a personal staging repo
- Other people pull your changes to test them
- Later, someone migrates changes into central repo

Things to look out for

- Nobody gets every one of these right
  - History-sensitive merge
  - Modifications follow renames
  - Rename and create conflicts on files and directories
  - Binary file management
  - Windows line endings
  - Symbolic links
  - Executable files
  - Cheap cloning
  - Comparing changes on branches

Outstanding problems

- How do I find other people's work, and publish mine?
  - Many tools have web/RSS interfaces, but these don't tell you enough
- UI tools are primitive to non-existent
- Write up "good" workflow practices for newcomers
- Everyone uses flawed merge algorithms
  - Bram is working on precise merge, which everyone will immediately adopt

Useful third-party tools

- Tailor – move changes between SCMs
- kdiff3 – excellent conflict resolution tool