DVCS and Git

Samuel Tesla
Automation Developer
Engine Yard
stesla@engineyard.com

What is Git?

Version Control System

- RCS
- CVS
- Subversion
- Perforce
- VSS
- Vault

Distributed Version Control System

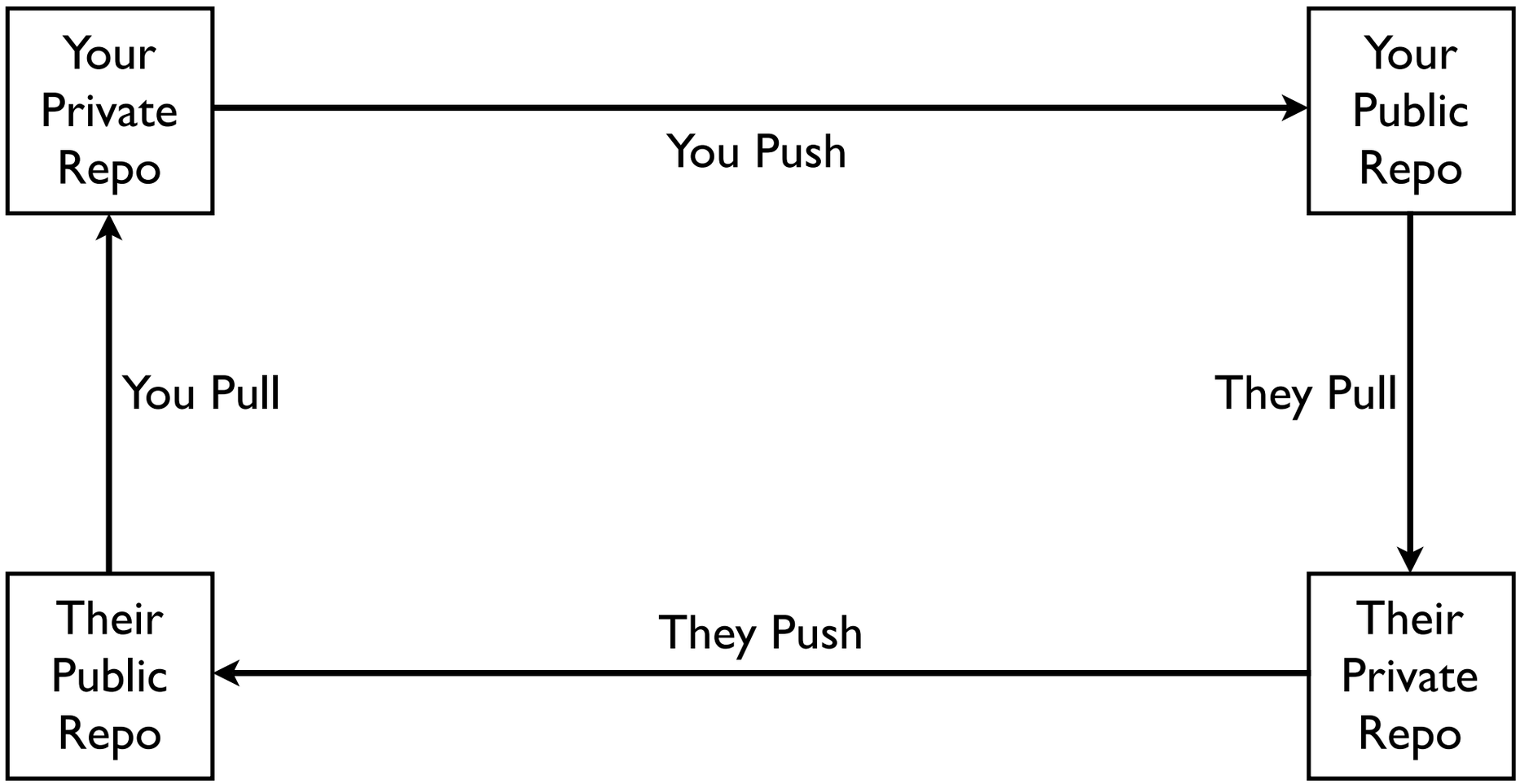
- Mercurial (hg)
- Bazaar (bzr)
- Git
- Monticello
- Darcs

Subversion

- Centralized repository is a single point of failure.
- The repository is the only place where history is stored.
- Working copies are just a snapshot.
- Cannot commit offline.

Git

- There are only repositories.
- Your “working copy” is a complete repository.
- Commits are made locally.
- Changes are pushed to or pulled from other repositories.
- Commits can be made offline.



Two Person Example

Why use Git?

Peer Pressure

- Linux
- KDE
- Wine
- Ruby on Rails
- Merb
- Rubinius

Scalability

- No central repository means no single bottleneck.
- The pull model eliminates the need for a committers list, and all the politics that go with it.
- The trust model is much more scalable.

Safety

- Every copy of the repository has a complete copy of the history.
- Git uses cryptographic hashes to prove that the content you get out is the same content you put in.
- You do not have to grant access to your repository to anybody.

Convenience

- Git is ultimately configurable.
- Git can be used offline.
- Branches can be made locally to manage changes.
- Git is fast.
- Git merges very well.

Demonstration

Resources

- http://git.or.cz/
- http://github.com/
- http://www.selenic.com/mercurial/wiki/
- http://bazaar-vcs.org/
- http://darcs.net/

Questions?