Subversion (svn) Basics

Department of Computer Science
Kent State University
Prof. Jonathan I. Maletic

Version Control

- svn is a tool for version control of files
- Keeps a repository of what is under control at a central location
- Allows users to make local (aka working) copies of the repository on their machine or file system
- Saves each version a user commits
- Administrator sets up main repository.

Getting and Updating

- *Checkout* a repository. A local/working copy of a repository is created on a machine or file system. This links a given folder to the address of the repository.
  - `svn checkout` *address folder*
- *Update* a local/working copy. The contents of the repository are copied to the local folder. This updates the local copy.
  - `svn update`

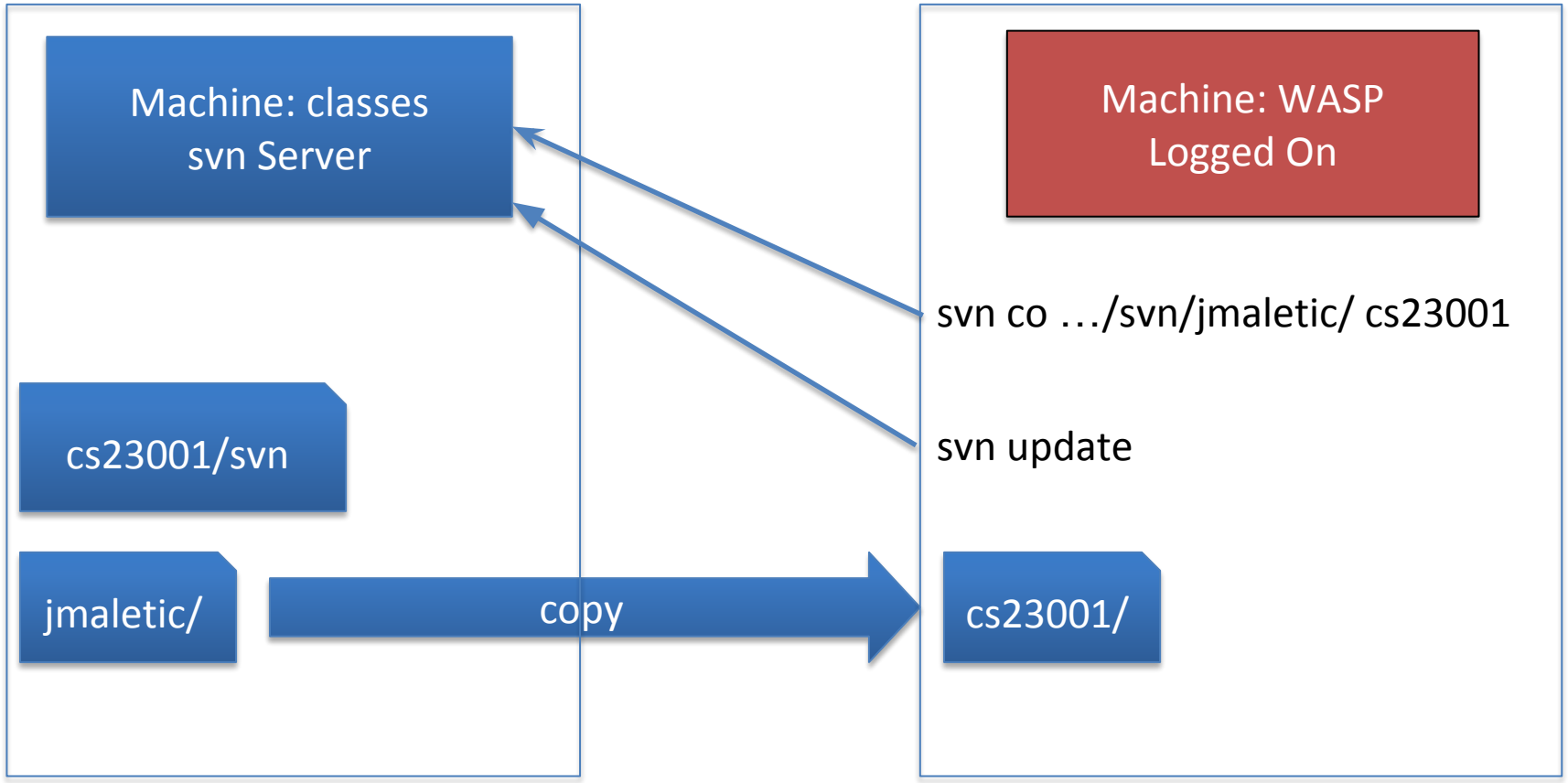
Committing Changes

- *Commit* a change. Copy a local change to repository. This commits a change to the repository. This is a new version of the item under version control. It is numbered and the old version is saved. Need to commit:
  - Modified files
  - Added or removed files or folders
  - `svn commit` *file* `-m` "*Commit note*"

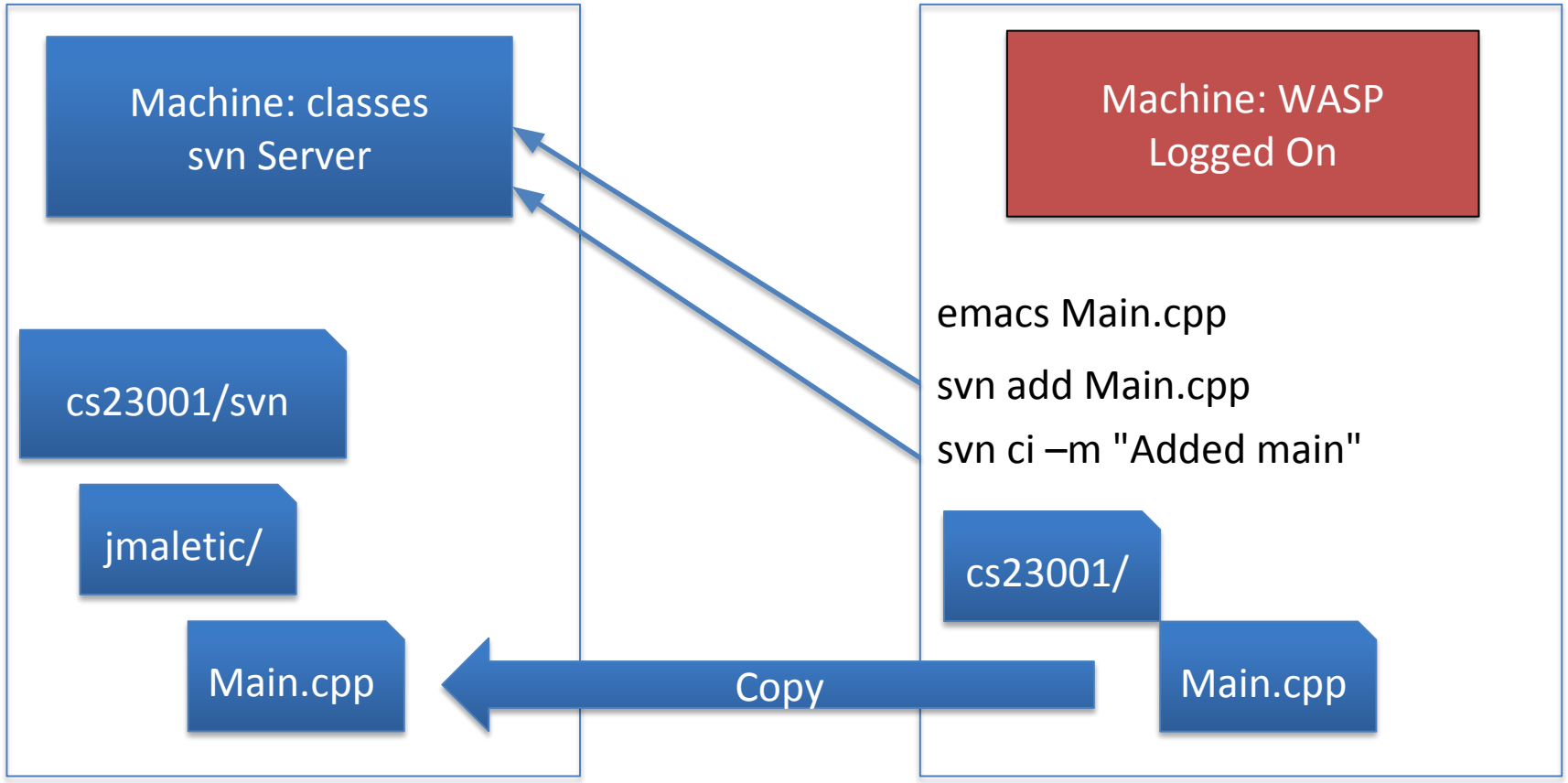
Other Essential Commands

- `svn add` – add a file. A commit must be done after the add to commit it to the repository.
- `svn delete` – delete a file, followed by a commit.
- `svn log` – get all the commit messages. Use this to roll back to an earlier version.

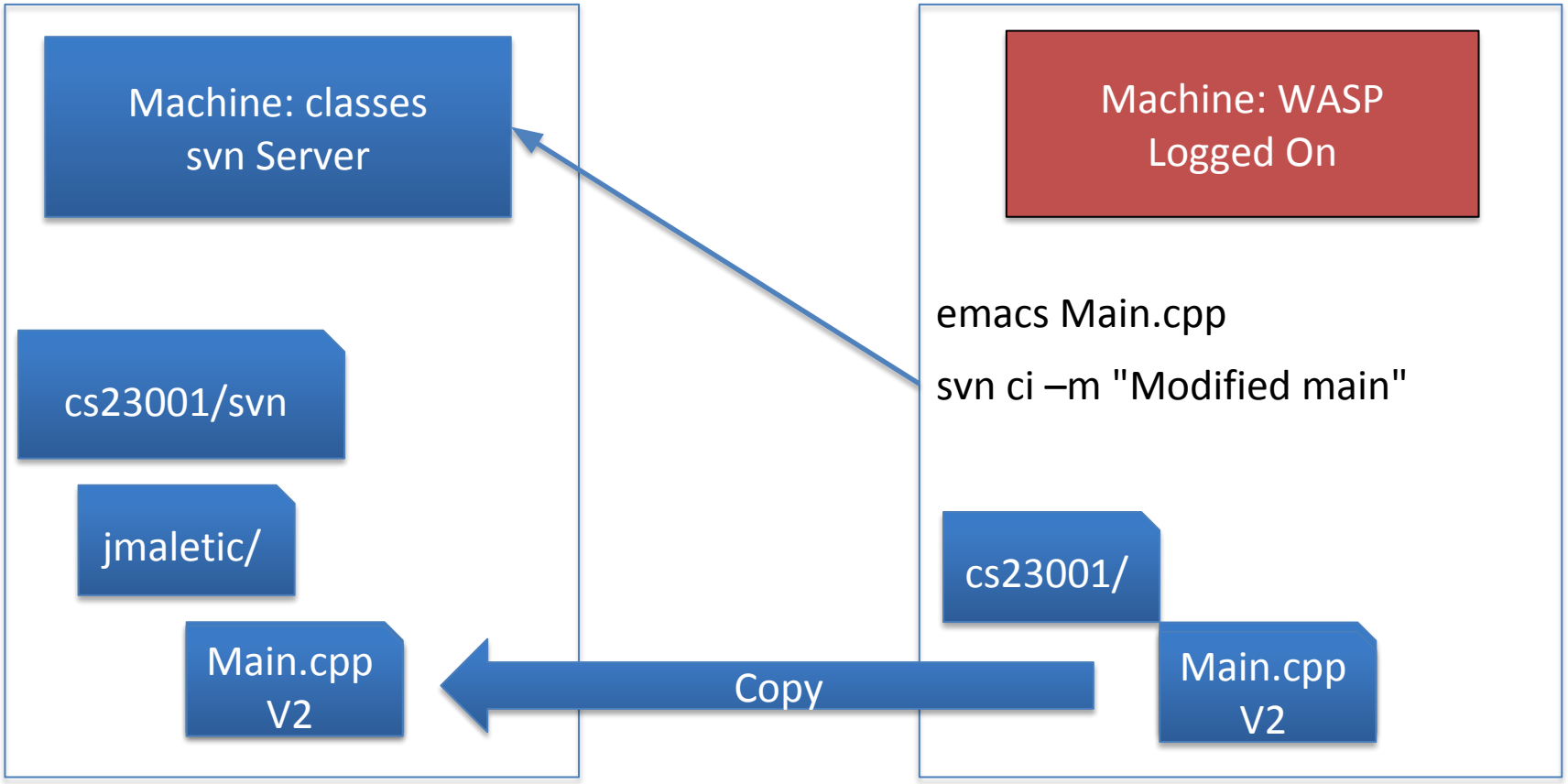
Check Out Working Copy



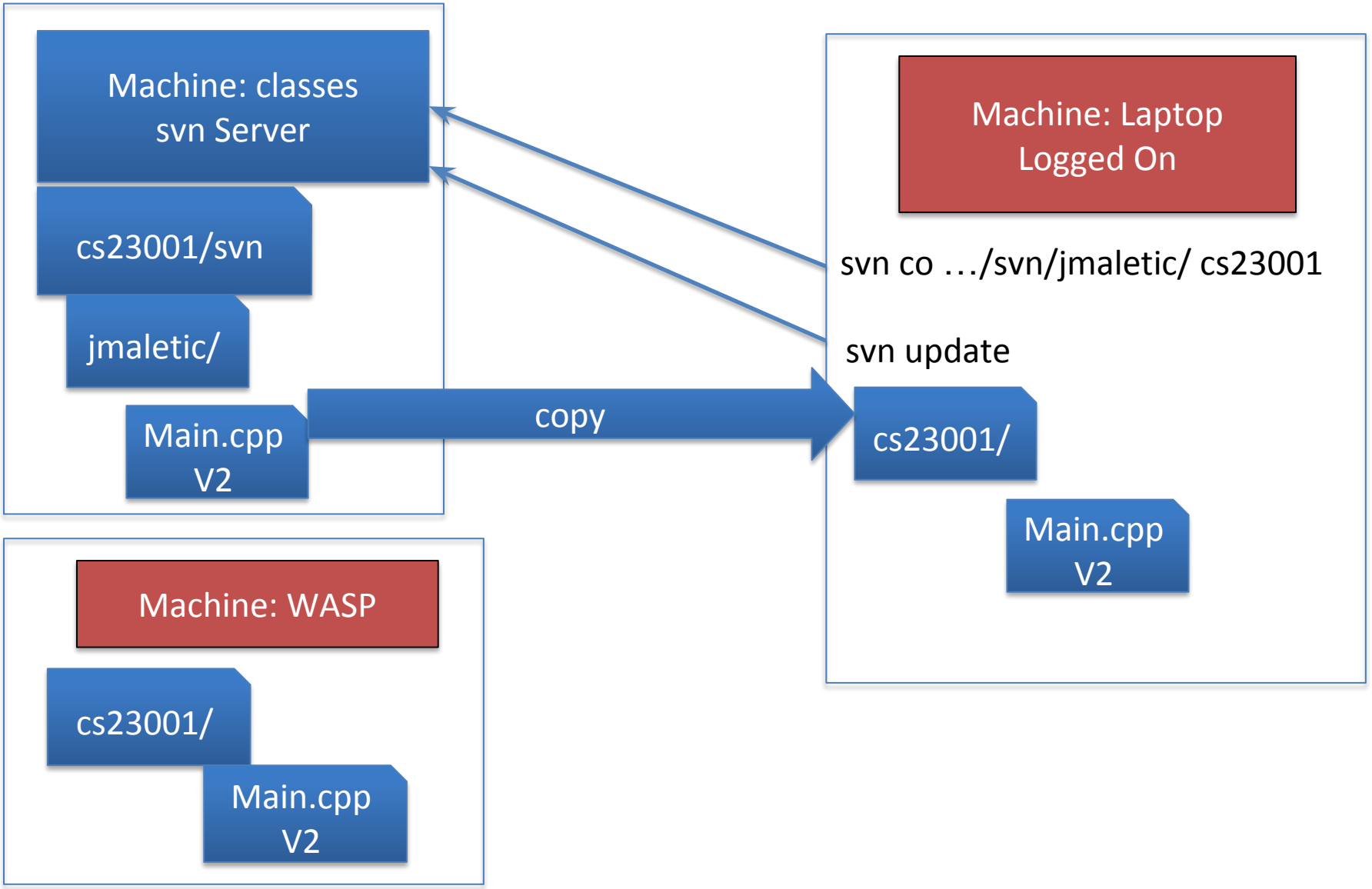
Add & Commit


Machine: classes
svn Server
Machine: WASP
Logged On
emacs Main.cpp
svn add Main.cpp
svn ci –m "Added main"
cs23001/svn
jmaletic/
Main.cpp
Copy
cs23001/
Main.cpp

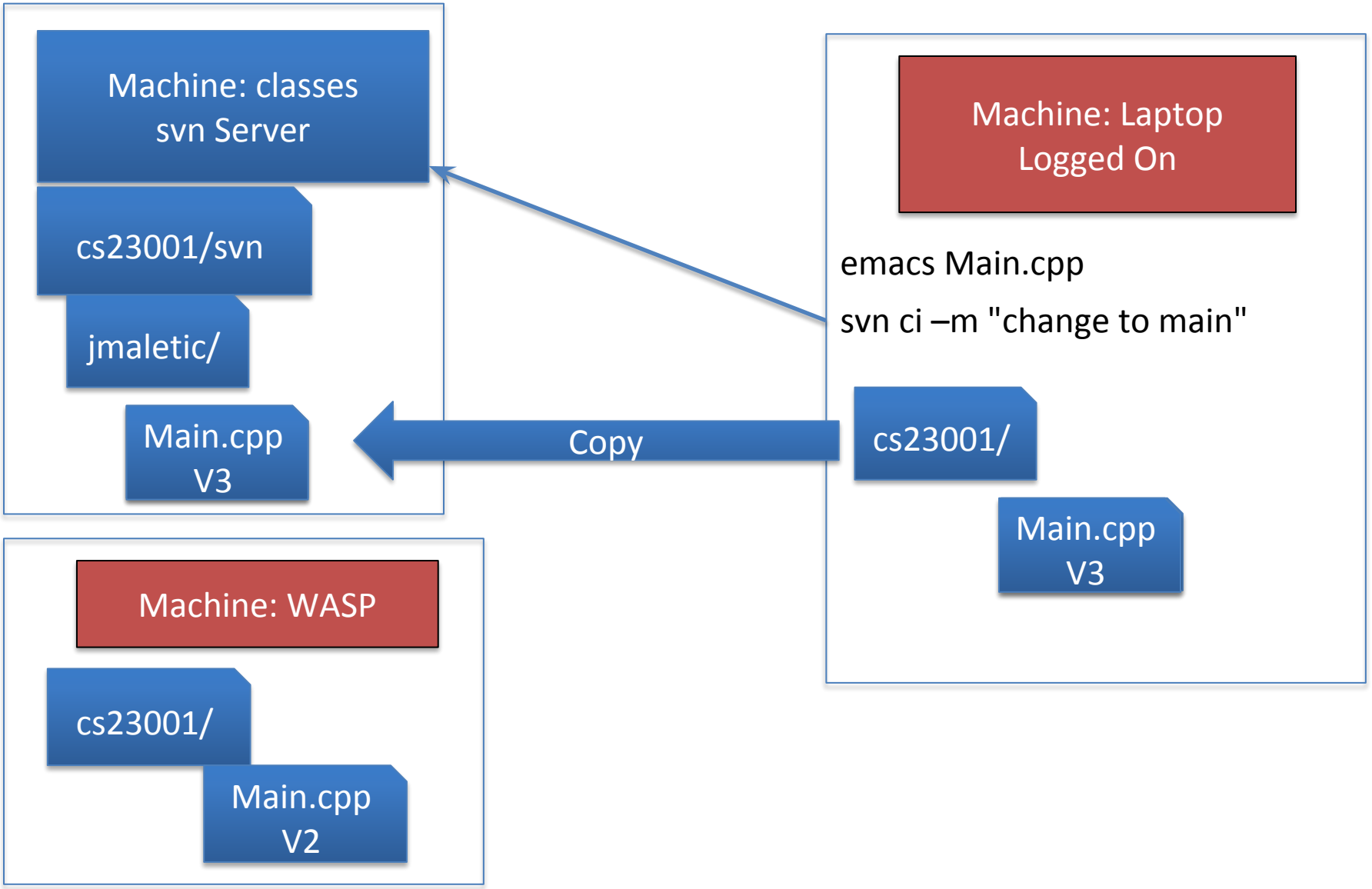
Commit a Change



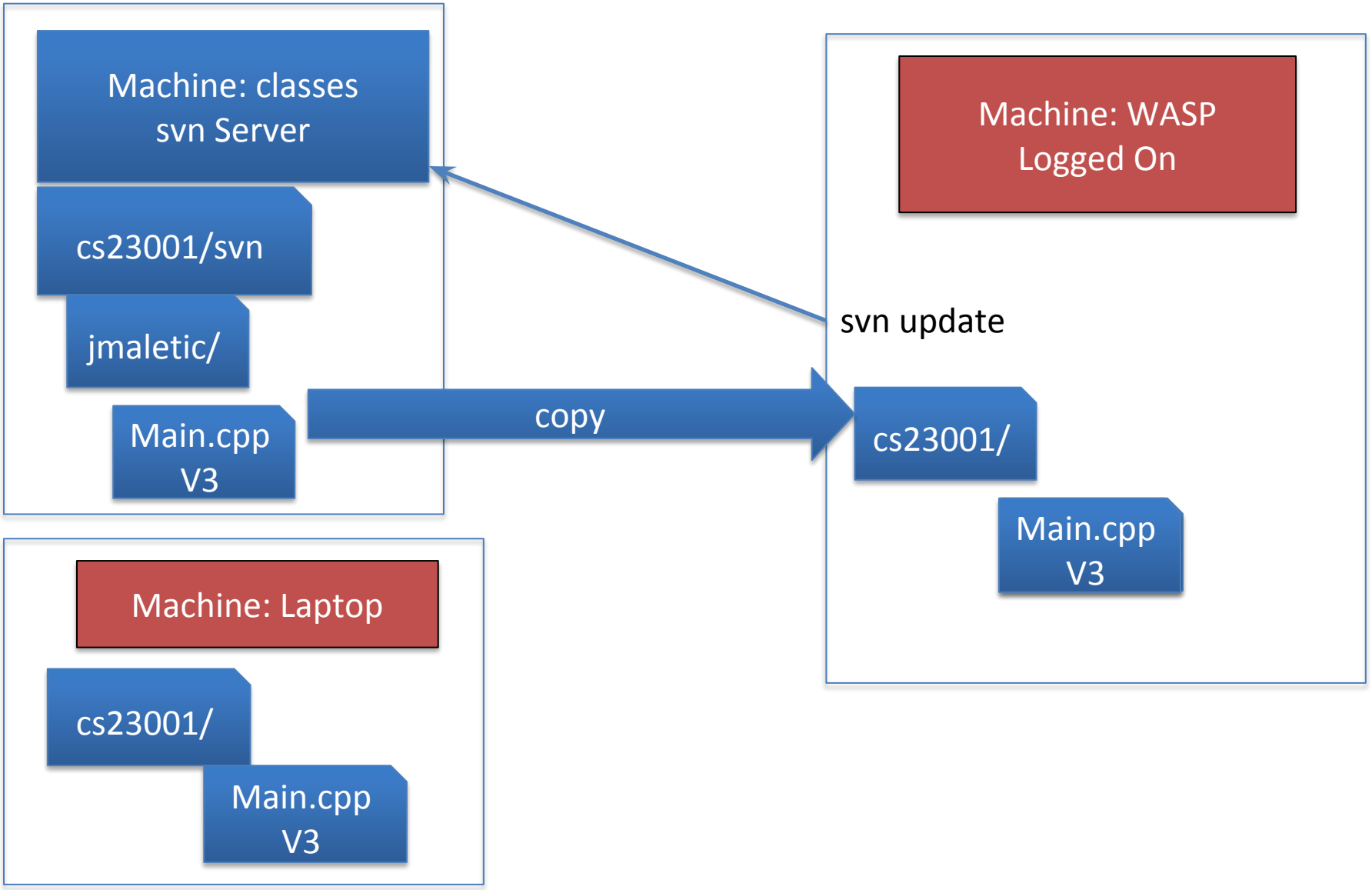
Multiple Working Copies



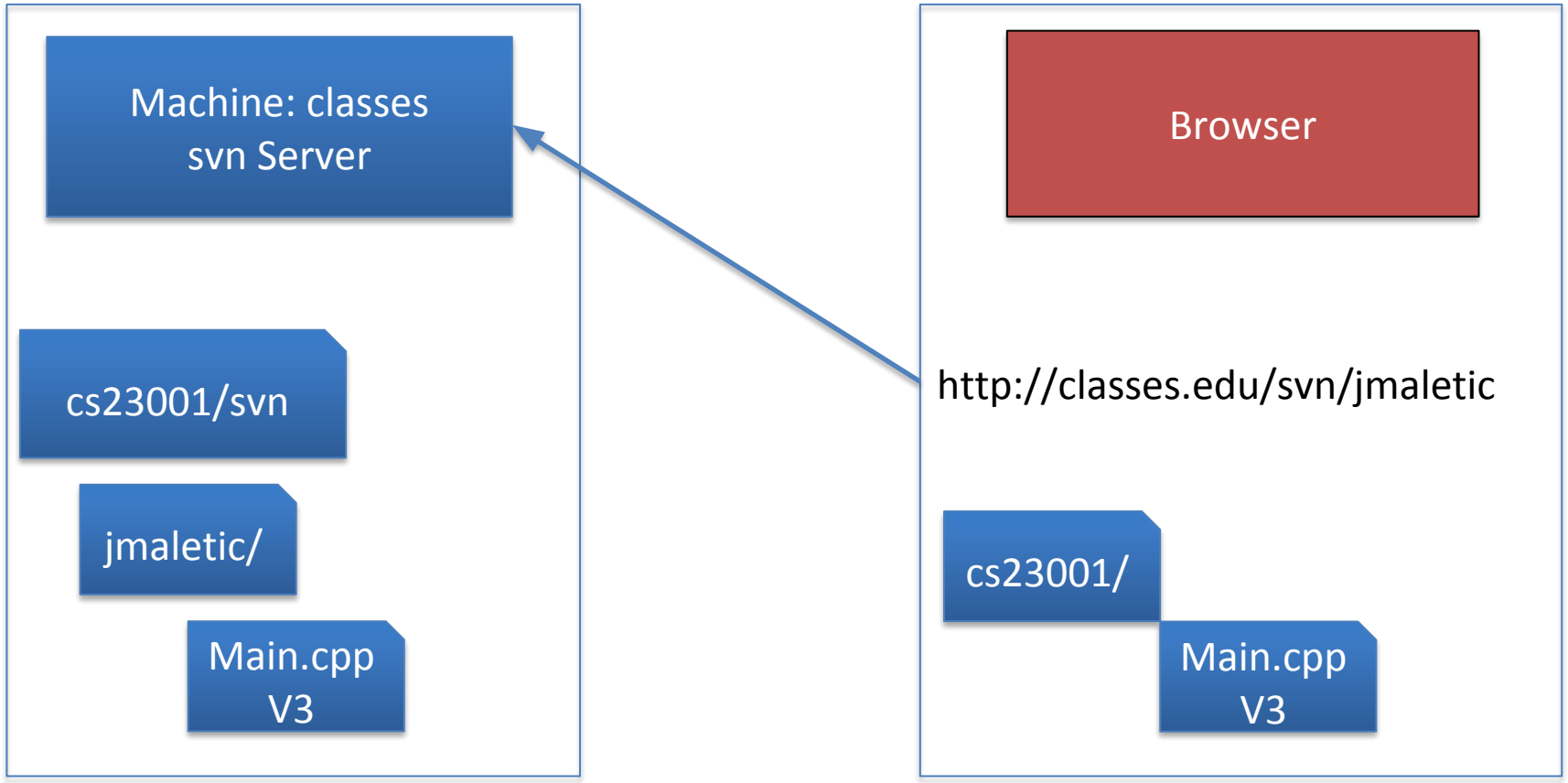
Commit a Change



Update WASP


Machine: classes
svn Server
cs23001/svn
jmaletic/
Main.cpp
V3
copy
Machine: WASP
Logged On
svn update
cs23001/
Main.cpp
V3
Machine: Laptop
cs23001/
Main.cpp
V3

Web Interface



Rules to Live By (svn)

- **Always** `update` before you start working
- **Always** `commit` after changes

- Commit often – after a major (or minor) change:
  - Add a method
  - Fix a bug
  - Change a method