An Introduction to Subversion

Stéphane Marchesin

stephane.marchesin@gmail.com

20 août 2009

What is SVN ?

- Subversion (also known as SVN) is a revision control software
- Stores your code in a safe place (a code repository) on a server
- The code on the server is only handled through the SVN tool (no direct modifications on the server)
- Keeps old versions
- Not the only versioning tool available (CVS, git, mercurial, bazaar...)
- More information at `http://svnbook.red-bean.com/`

Why do you need SVN ?

```
Student: "Paul, can we share the code for our project?"

Paul: "Yes, I have a plan for that. We will do regular
       copies from your machine to mine and back using
       an USB key"
```

Why do you need SVN ?

SVN lets you share code between multiple people.

Why do you need SVN ?

```
Student: "Paul, can you get last week's advection.c file?"

Paul: "Yes, I think I still have it,
       let me look at my folder"

 $ ls
 advection.c          advection.c.new   advection.c.old2
 advection.c.bak      advection.c.old   advection.c.tmp
 advection.c.broken   advection.c.OLD   advection.c.works-ok
```

Why do you need SVN ?

SVN keeps all the previous versions of your file and allows you to find them easily.

Why do you need SVN ?

```
Paul: "My hard drive crashed. I'll have to redo all my work
       since the last backup I did (about a month ago)."

Student: "You don't do regular backups?"

Paul: "..."
```

Why do you need SVN ?

SVN also acts as a backup (yes, disasters happen.)

Why do you need SVN ?

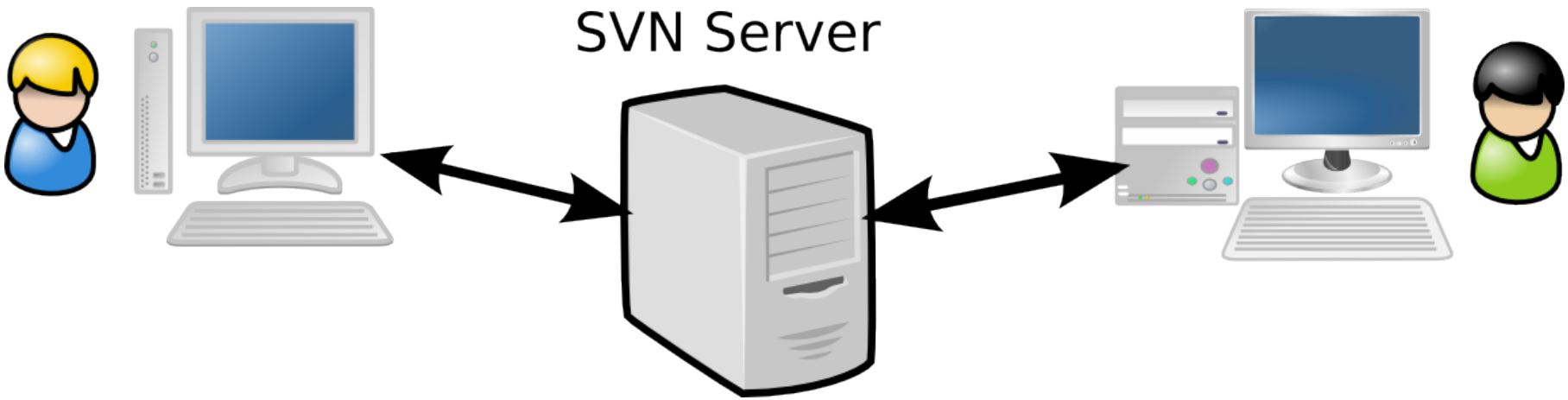
```
Student: "Do you know who added this bug on line 154 of
          advection.c?"

Paul: "Yes, I have a great technique to know
       who did changes and what they did"

/* Function compute(float* t) Added by Brian on 04/05/2007 */
void compute(double*t)
// Switched from float to double for accuracy
// D. on 25/06/2007
{
// commented out since it does not work 29/04/2008
// float f[10];
// 0.003 is better than 0.005 - T. on 24/08/2008
double d = 0.003;
d += exp(t[3]);    <- line 154
[...]

Paul: "Oops, line 154 does not have a comment"
```

Why do you need SVN ?

SVN lets you know who did changes to the code, when and why.

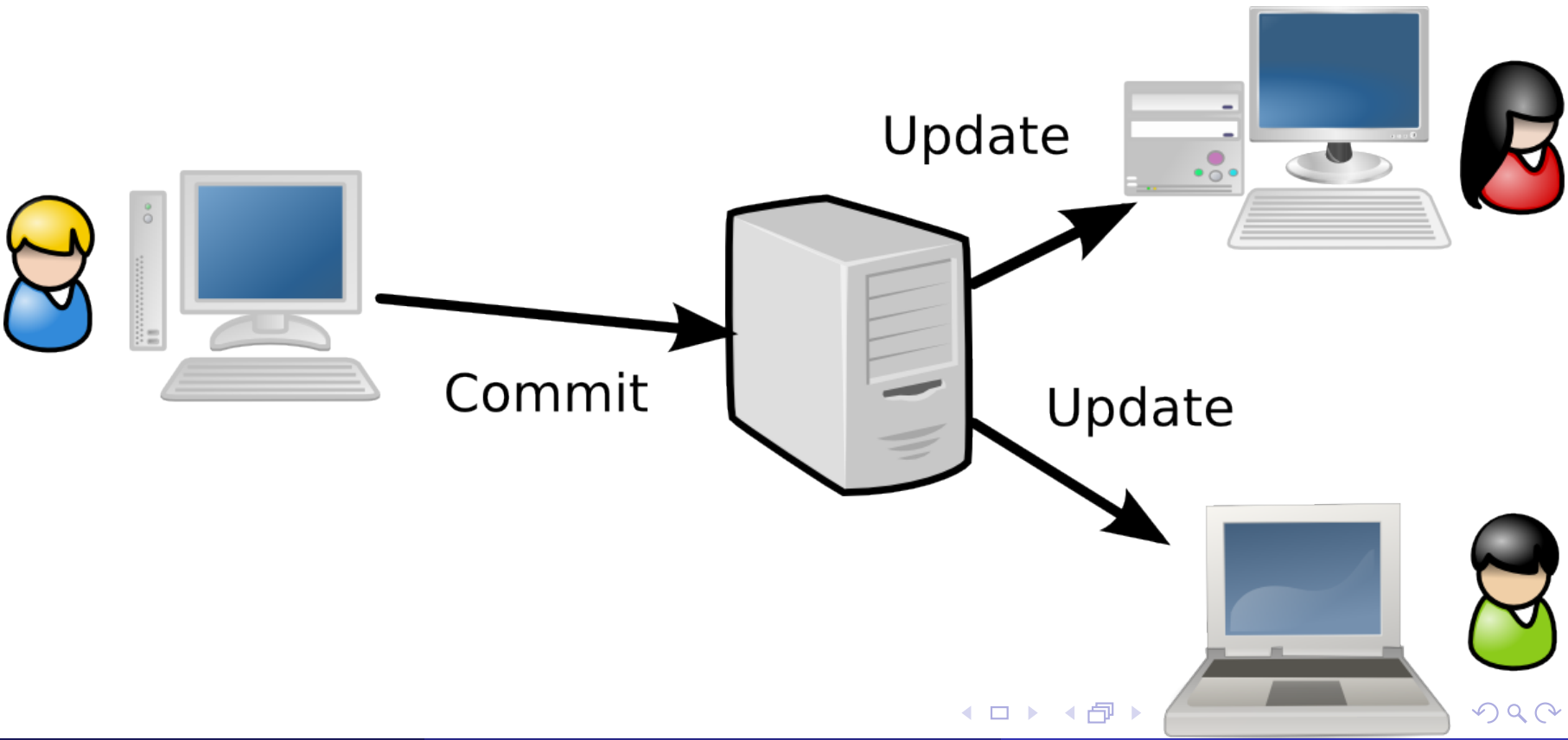
How do you work with SVN ?

- There is a central code repository on a server
- Each developer makes a copy of this repository on his computer
- Each time a developer wants to do a change, he has to send it to the server
- Developers can update their local copy with respect to the server's copy
- Everything goes through the server, no direct interaction between developers



How do you work with SVN ?

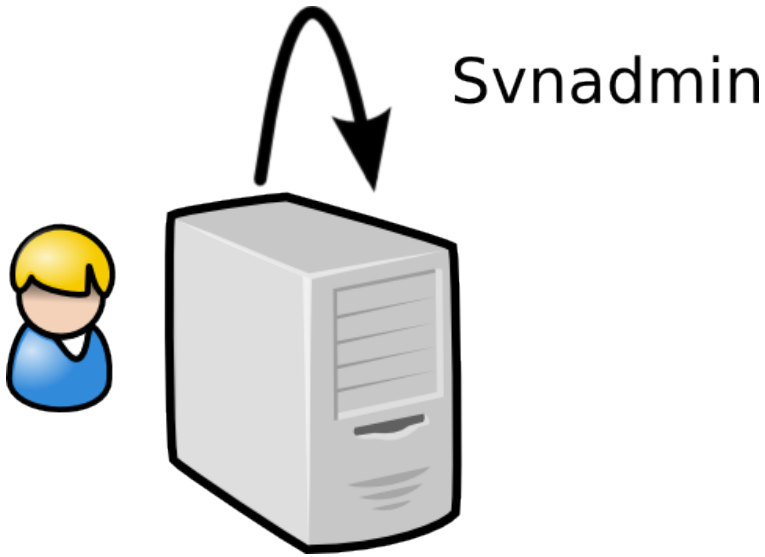
- Workflow with SVN
  - A developer makes a change to the code
  - The developer pushes the change to the server ("commit")
  - Other developers can update their copy of the code to get the latest change ("update")



Basic SVN commands : svnadmin

svnadmin create

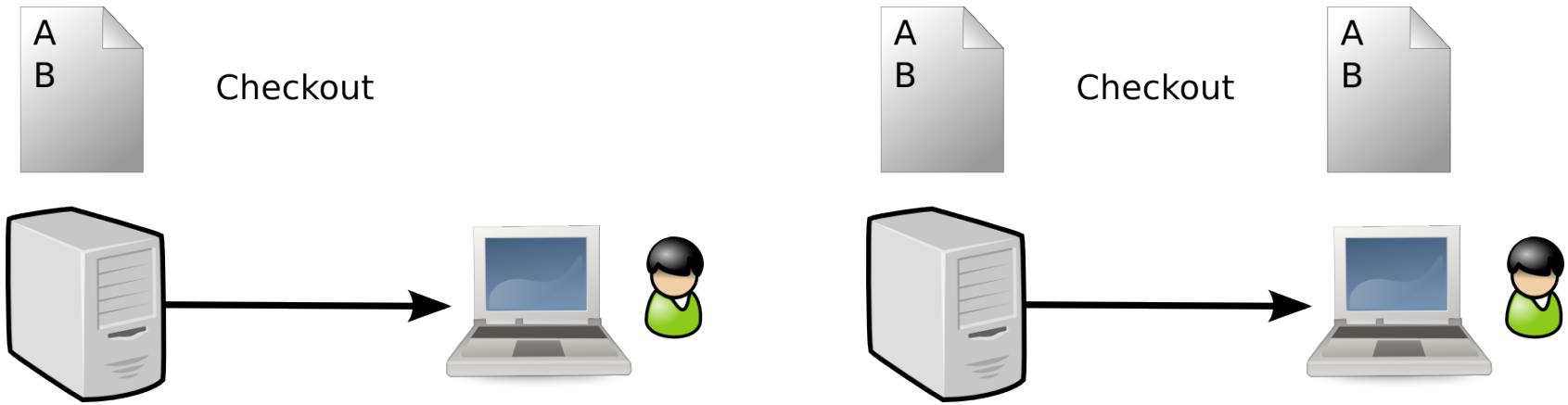
- Creates a new SVN repository on the server
- Usually you only use this once
- svnadmin create [repository name]



Basic SVN commands : svn checkout

svn checkout

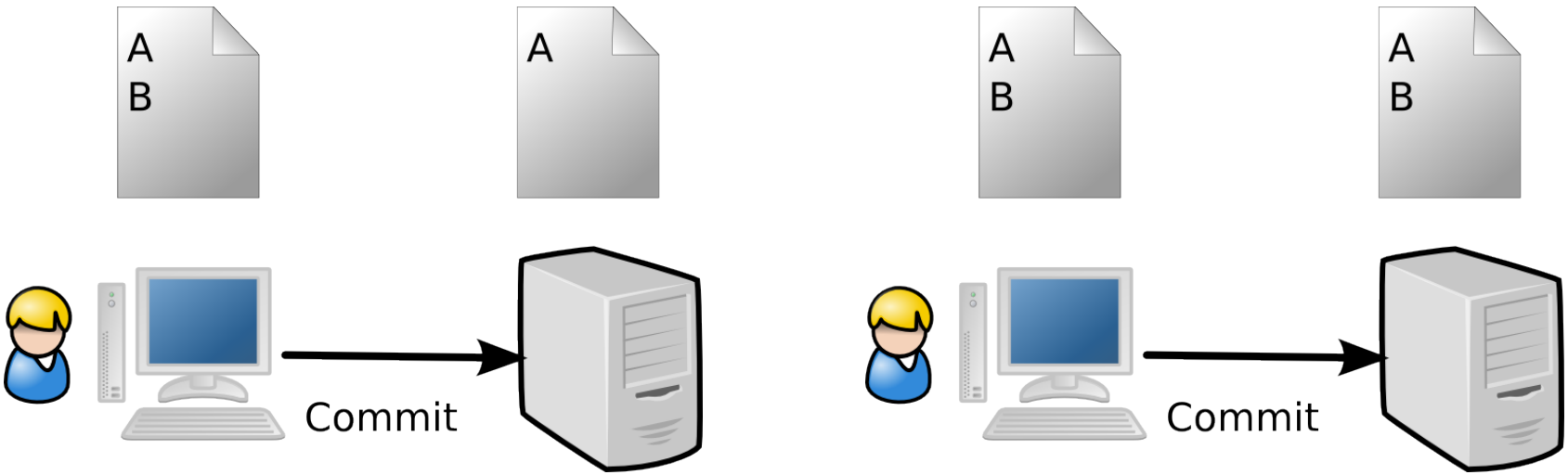
- Creates a new local copy from the server
- This copy can be subsequently worked on
- svn checkout [repository URL]



Basic SVN commands : svn commit

svn commit

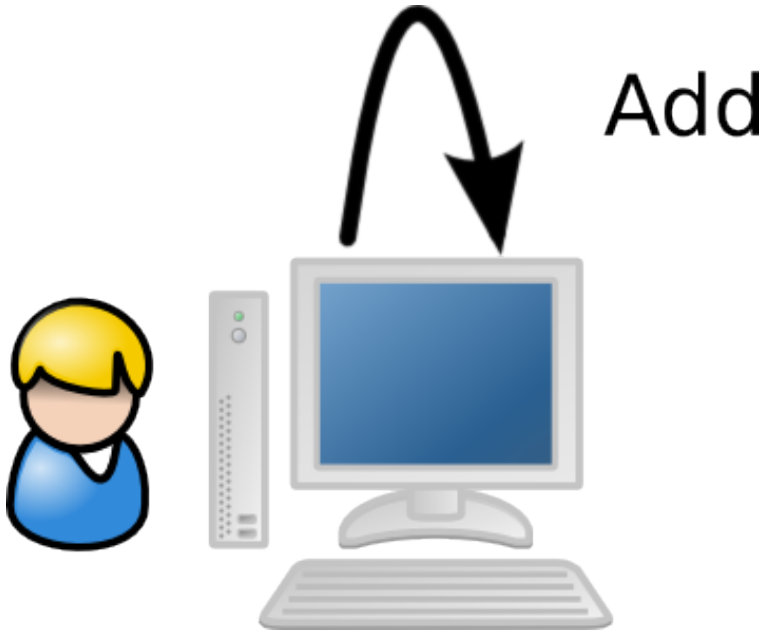
- Takes the current changes, and puts them on the server
- Asks for a message, which describes the change
- Creates a new revision on the server with the changes



Basic SVN commands : svn add

svn add

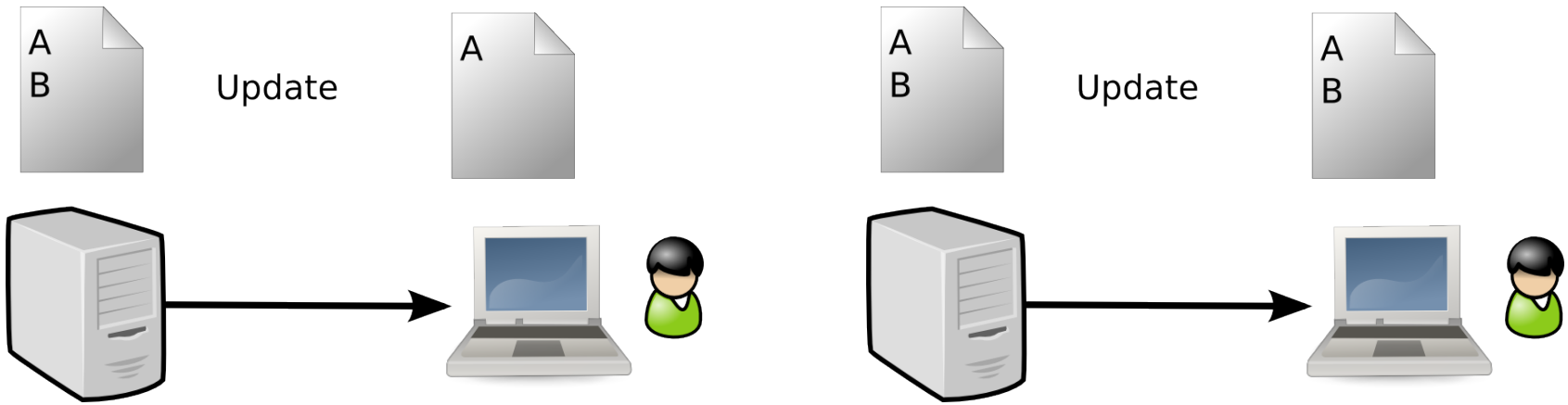
- svn add [filename]
- Adds a file to the files that SVN tracks
- Local operation
- No effect until the next "commit"
- After the next "commit", the file will be copied to the server



Basic SVN commands : svn update

svn update

- Updates the local copy from the server
- The fetches changes from other developers
- svn update -r [version] to fetch an old version



Basic SVN commands : svn resolved

svn resolved

- svn update fails, and tells you there was a conflict
- Because local changes conflict with the latest changes on the server
- The offending file is modified as follows :

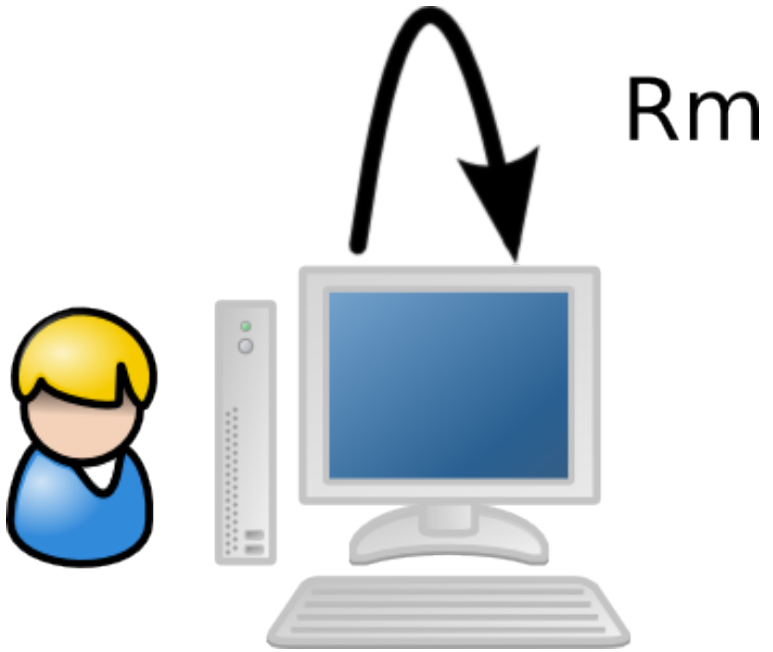
```
<<<<<<< .mine
What I added
=======
What was added on the server
>>>>>>> .r4
```

- Edit the file to resolve the conflict
- svn resolved [filename]
- svn commit

Basic SVN commands : svn rm

svn rm

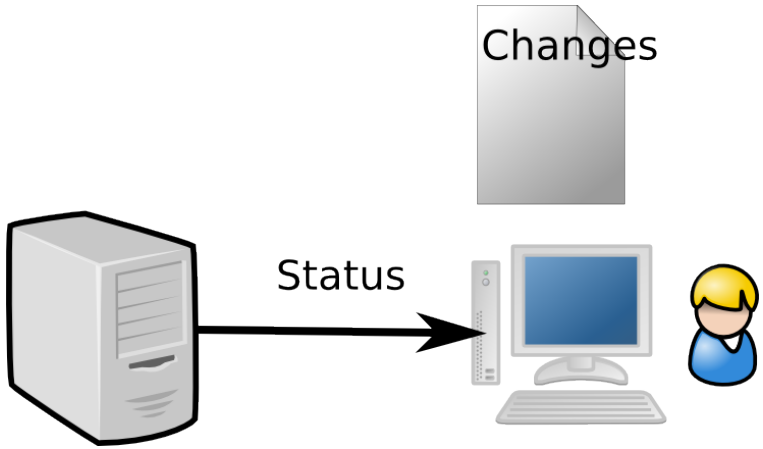
- svn rm [filename]
- Removes a file from the files that SVN tracks
- Local operation
- No effect until the next "commit"
- After the next "commit", the file will be removed from the server
- Of course, old versions of the files are still reachable



Basic SVN commands : svn status

svn status

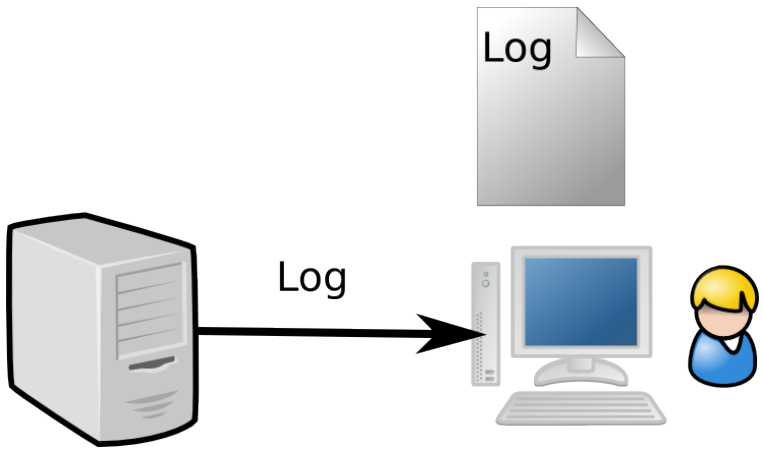
- Checks the status of your local copy
- Compares it to the server's version
- Tells you what changes you have made



Basic SVN commands : svn log

svn log

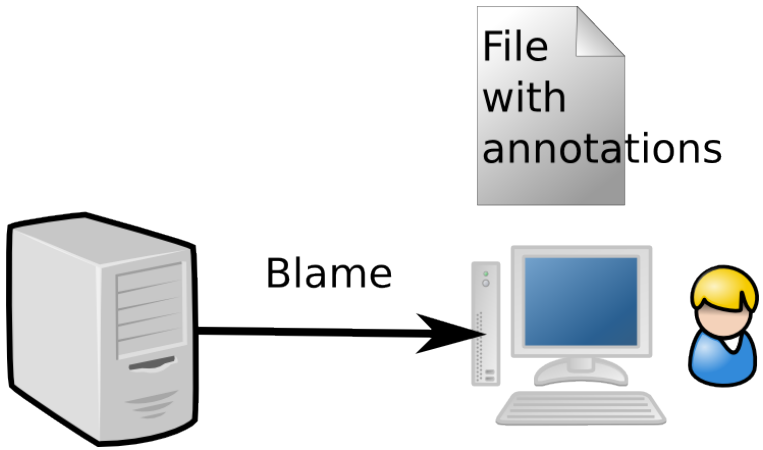
- Ask the server for a log of the changes
- svn log -r 0 :HEAD to see all the changes



Basic SVN commands : svn blame

svn blame

- svn blame [filename]
- For each line of the file, tells you who changed it last
- Useful to blame someone for a bug
- Also works as "svn praise"



Installing and using SVN

- svn (command line tool) : linux, windows, OSX
- TortoiseSVN (extension for windows explorer)
- SCplugin (extension for OSX finder)
- See `http://subversion.tigris.org/links.html#clients` for other clients

Installing and using SVN

- We already have a SVN server !
- `http://grille.u-strasbg.fr:8000/svn/`
- `http://cemr20/svn` (faster, but only available internally)
- Can host your projects
- Viewable with your web browser
- Login and password same as the other cemracs machines

Installing and using SVN

- Install a SVN client
- Checkout a copy of `http://grille.u-strasbg.fr:8000/svn/test1`
- Make a change to the file
- Add your change to the server
- List all previous changes
- Add a new file with your name, and put it on the server
- Update your copy to get your neighbour's file
- Now import your project into the SVN server !