

Version Control Software and why it is useful

Rob's Technical Seminar 13.07.2012
Jonathan Mackey

What are we talking about???

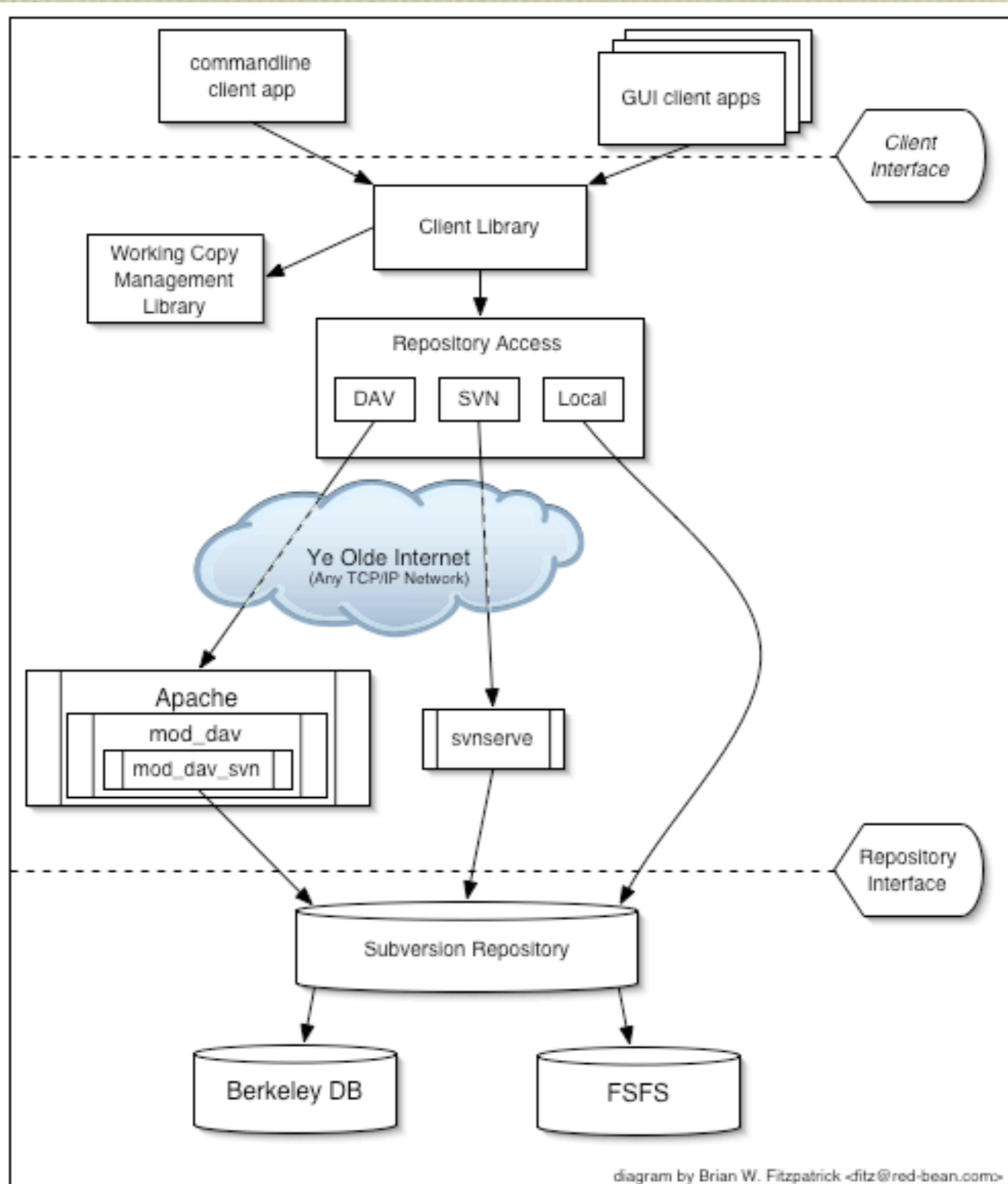
- Software that tracks changes in files (revisions).
- Stores all files and their history in an archive.
- Archive can be centralised (CVS, Subversion), or distributed (Git... I won't talk about this).
- Multiple users can work with the archive.
- Access control (e.g. username/password).
- Conflict resolution (if two people edit one file).

Why use version control?

- Sometimes people get very drunk and have a “great idea” for improving their code...
- Sometimes people accidentally delete files...
- If someone asks about reproducing a plot from your paper from ten years ago.
- If you work with a code that has more than one user (e.g. Norbert’s stellar evolution code).
- If you find a bug in your code and need to know if your paper from 2008 is affected.
- If you use more than one computer for work.
- It is really, really, easy!
- Why not??? It has no disadvantages.

CVS and Subversion (SVN)

- CVS was (>5 or 10 years ago) the default standard for version control, but it had problems...
- SVN was designed to be a better CVS, and has replaced it. CVS is no longer developed or maintained.
- Open source, available on most platforms.
- GUI interfaces exist, already included in IDEs.
- Central server has master copy of archive, “Repository”.
- Users “check out” local copies from repository (repo).
- Can then edit files, and “commit” changes to repo, which adds a new revision to the master copy in repo.
- Or, can revert local copy to master version.
- User can delete local copy with no effect on repo.

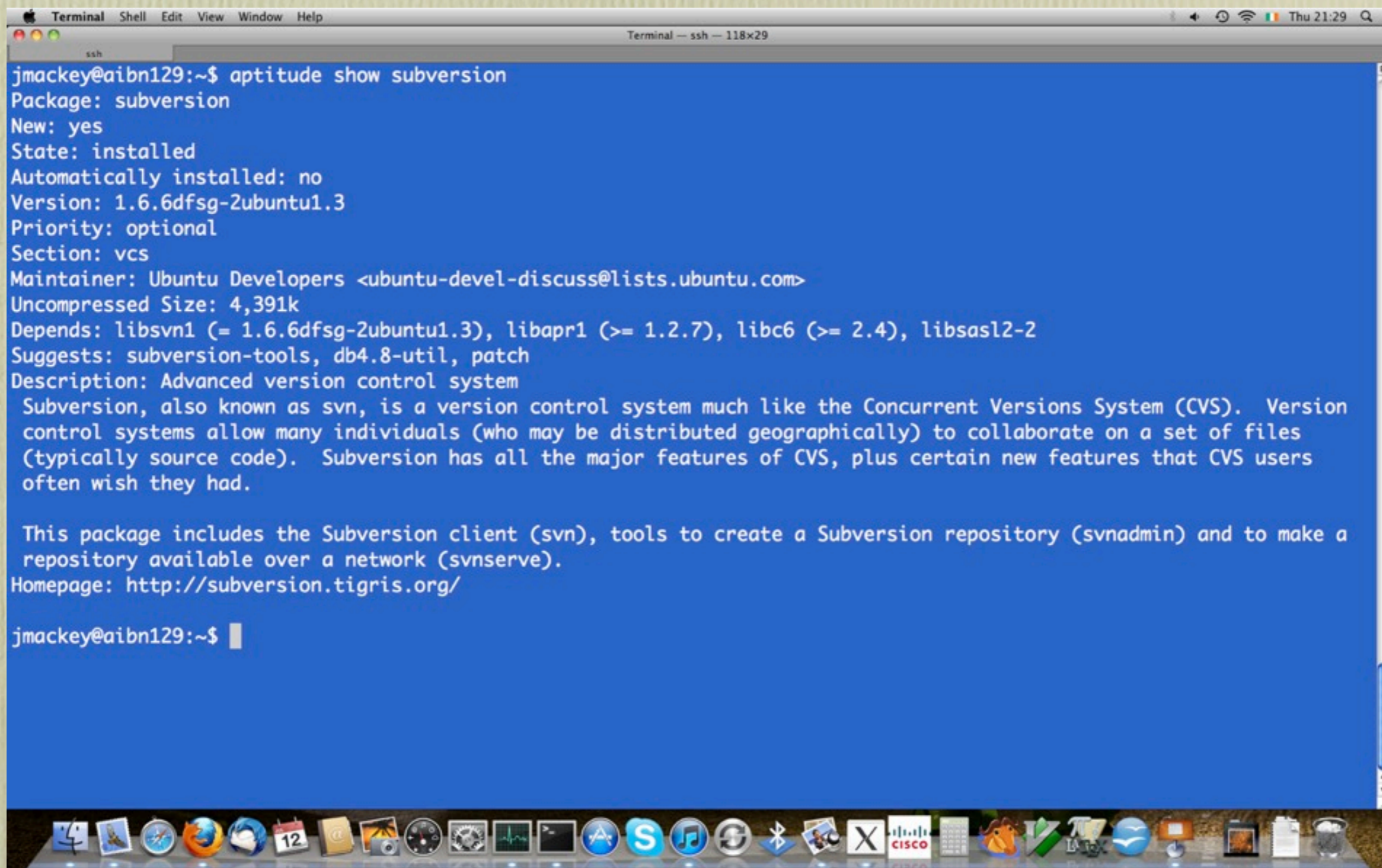


From the
SVN book

How to get SVN

- Already there on AIfA computers.
- debian/ubuntu: `aptitude install subversion`
- OS X: pre-installed I think.
- Other OS? I don't know, can't be that hard.
- Homepage: <http://subversion.apache.org/>
- Free book: <http://svnbook.org/>

See, it really is installed...



```
Terminal — ssh — 118x29
ssh
jmackey@aibn129:~$ aptitude show subversion
Package: subversion
New: yes
State: installed
Automatically installed: no
Version: 1.6.6dfsg-2ubuntu1.3
Priority: optional
Section: vcs
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Uncompressed Size: 4,391k
Depends: libsvn1 (= 1.6.6dfsg-2ubuntu1.3), libapr1 (>= 1.2.7), libc6 (>= 2.4), libsasl2-2
Suggests: subversion-tools, db4.8-util, patch
Description: Advanced version control system
 Subversion, also known as svn, is a version control system much like the Concurrent Versions System (CVS). Version control systems allow many individuals (who may be distributed geographically) to collaborate on a set of files (typically source code). Subversion has all the major features of CVS, plus certain new features that CVS users often wish they had.

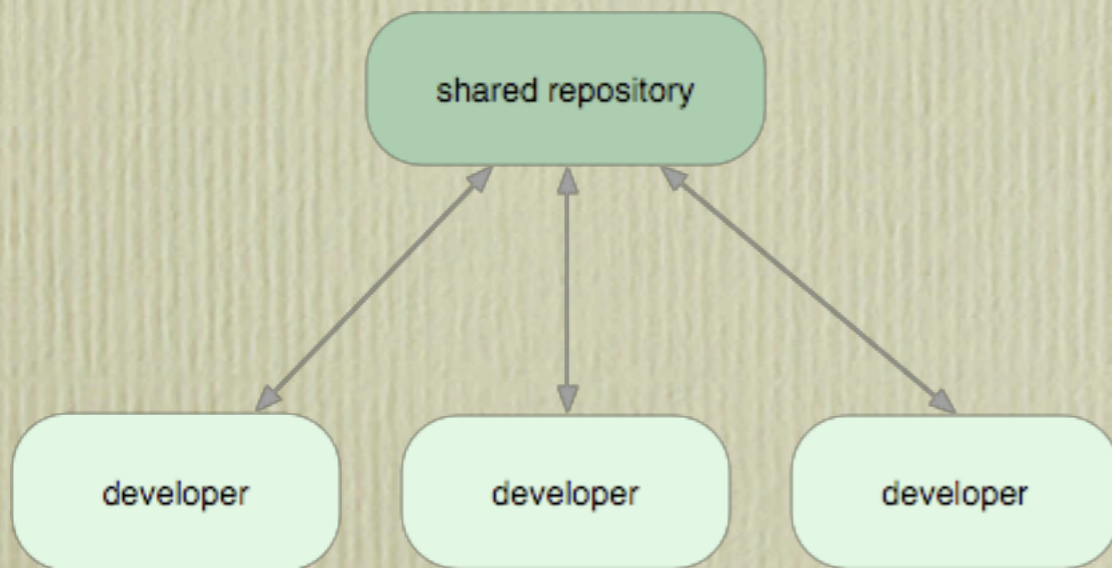
This package includes the Subversion client (svn), tools to create a Subversion repository (svnadmin) and to make a repository available over a network (svnserve).
Homepage: http://subversion.tigris.org/

jmackey@aibn129:~$ █
```

SVN weaknesses

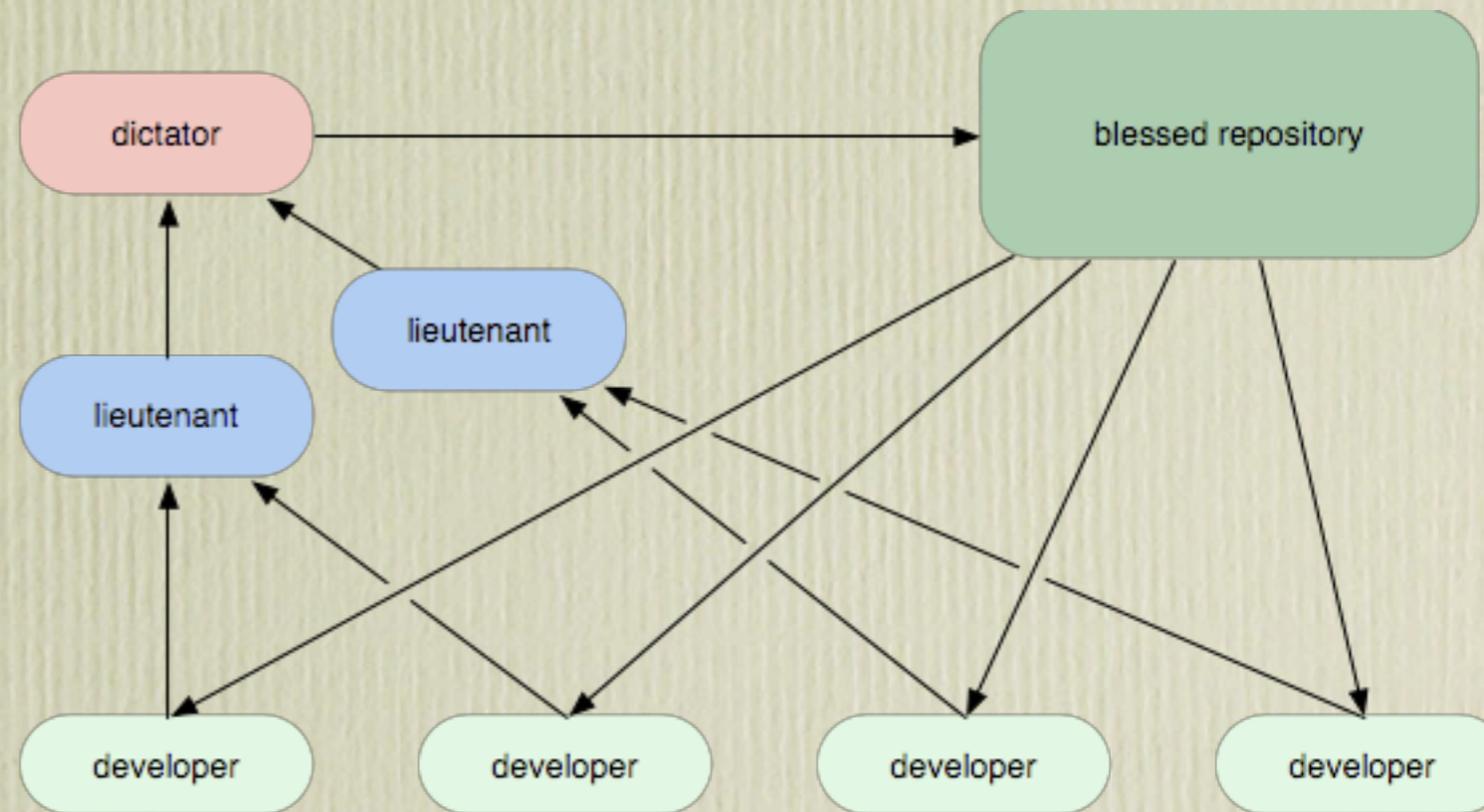
- Working copy only contains current version, so most operations require internet access (if repository is remote).
- “Git” users claim SVN is slow and bloated (and old).
- Some people like git better. It has more workflow support.
- It seems like overkill for my work.

Typical SVN structure:



<http://github.com/schacon/whygitisbetter>

Git can do stuff like this:



What you need to know:

- How to set up/start a subversion repository.
- How to import your project into the repo.
- How to check out a working copy.
- How to edit and add files/directories.
- How to commit changes back to the repo.
- Comparing revisions of a file, reverting to older copies, that sort of thing.
- What if it all goes wrong (argh! Tree Conflicts!)

Setting up a SVN Repo

```
jmackey@aibn129:~$  
jmackey@aibn129:~$ mkdir temp_svn_repo_store  
jmackey@aibn129:~$ svnadmin create /users/jmackey/temp_svn_repo_store/RobTechnicalSeminar  
jmackey@aibn129:~$ █
```

- Where to store the repository?
 - AIfA has a SVN server with http access.
IT support staff can set up a repository for you.
 - Or in your home directory (ssh remote access).
You can set this up yourself.
- What to call it?
- That's it.
- Command: `svnadmin create /path/to/repo/NAME`

Starting a project

- Probably have an existing project. If so, make a clean copy of it, with no binary files (usually) or temporary files.
- Or start a new project: create a directory and a file or two.
- Then import the entire project directory to your new SVN repo.

```
Terminal Shell Edit View Window Help
Terminal — ssh — 103x25
ssh
jmackey@aibn129:~$ cd
jmackey@aibn129:~$ mkdir temp_svn_repo_store
jmackey@aibn129:~$ svnadmin create /users/jmackey/temp_svn_repo_store/RobstechnicalSeminar
jmackey@aibn129:~$ # now create the project
jmackey@aibn129:~$ mkdir temp
jmackey@aibn129:~$ mkdir temp/tech_seminar
jmackey@aibn129:~$ echo "List of speakers/topics" > temp/tech_seminar/speakers.txt
jmackey@aibn129:~$
jmackey@aibn129:~$ # Good enough, now import the project into the SVN repo.
jmackey@aibn129:~$ svn import ~/temp/ \
> file:///users/jmackey/temp_svn_repo_store/RobstechnicalSeminar \
> -m "initial import of new project"
Adding          /users/jmackey/temp/tech_seminar
Adding          /users/jmackey/temp/tech_seminar/speakers.txt

Committed revision 1.
jmackey@aibn129:~$
jmackey@aibn129:~$ # "-m" is a flag to specify the log-message to write with the import.
jmackey@aibn129:~$
jmackey@aibn129:~$ # You can now delete the project (maybe check it imported ok first...).
```

Get a working copy and edit stuff

- `svn co file:///path/to/repo/NAME /path/to/working/copy/`

```
Terminal Shell Edit View Window Help
Terminal - ssh - 127x30
ssh
jmackey@aibn129:~$ # Check out a working copy of the project:
jmackey@aibn129:~$ mkdir working_copy
jmackey@aibn129:~$ svn co file:///users/jmackey/temp_svn_repo_store/RobTechnicalSeminar working_copy/
A    working_copy/tech_seminar
A    working_copy/tech_seminar/speakers.txt
Checked out revision 1.
jmackey@aibn129:~$
jmackey@aibn129:~$ # Edit a file:
jmackey@aibn129:~$ cd working_copy/tech_seminar/
jmackey@aibn129:~/working_copy/tech_seminar$ echo "Rob: Intro to seminar, other stuff, all very interesting" >> speakers.txt
jmackey@aibn129:~/working_copy/tech_seminar$ echo "Mackey: boring talk about SVN, won't invite him back." >> speakers.txt
jmackey@aibn129:~/working_copy/tech_seminar$
jmackey@aibn129:~/working_copy/tech_seminar$ # Create a new file and edit it:
jmackey@aibn129:~/working_copy/tech_seminar$ echo "List of Attendees:" > attendance.txt
jmackey@aibn129:~/working_copy/tech_seminar$ echo "-----" >> attendance.txt
jmackey@aibn129:~/working_copy/tech_seminar$ echo "Rob: attends every week. punctual. contributes questions. A" >> attendance.txt
jmackey@aibn129:~/working_copy/tech_seminar$ echo "J.Mackey: usually attends. always late (even for own talk). C-" >> attendance.txt
jmackey@aibn129:~/working_copy/tech_seminar$ svn add attendance.txt
A    attendance.txt
jmackey@aibn129:~/working_copy/tech_seminar$
jmackey@aibn129:~/working_copy/tech_seminar$ # Commit changes
jmackey@aibn129:~/working_copy/tech_seminar$ svn commit -m "Added Rob and J.Mackey to list of speakers. Added Rob and J.Mackey to new file keeping track of attendance."
Adding      tech_seminar/attendance.txt
Sending     tech_seminar/speakers.txt
Transmitting file data ..
Committed revision 2.
jmackey@aibn129:~/working_copy/tech_seminar$
```

Getting info about Files

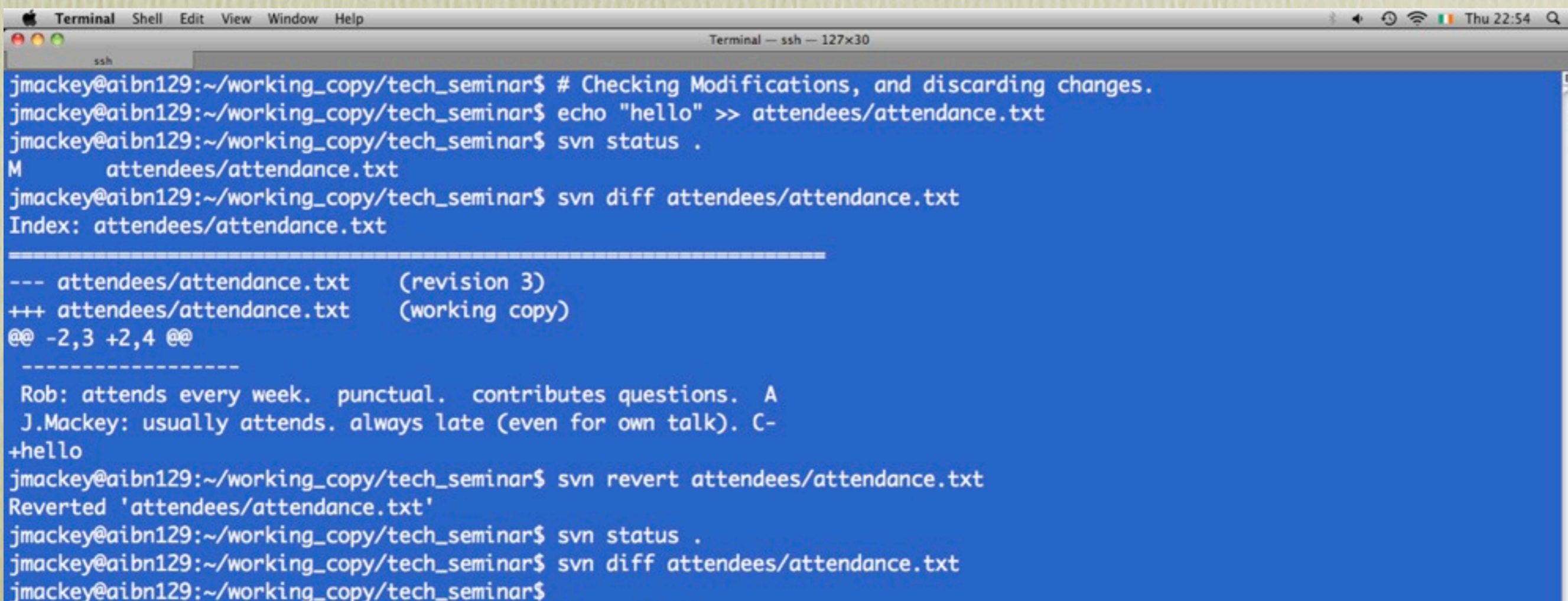
- All svn commands have this help syntax:
`svn help [command]`
- Adding a file: first create the file, add some text, then:
`svn add filename`
- Adding a directory: DO NOT create it first!
`svn mkdir directory_name`
- Getting history of a file:
`svn log path/to/file.txt`
- Diff-ing a file compared to revision:
`svn diff file.txt`
`svn diff -r1:3 file.txt`
`svn diff -rHEAD file.txt`
- Reverting a file:
`svn revert filename`

Using DIFF and LOG

```
Terminal Shell Edit View Window Help
Terminal - ssh - 127x30
ssh
jmackey@aibn129:~/working_copy/tech_seminar$ # Check the history of files.
jmackey@aibn129:~/working_copy/tech_seminar$ svn log topics_speakers/speakers.txt
-----
r3 | jmackey | 2012-07-12 22:42:43 +0200 (Thu, 12 Jul 2012) | 1 line
Reorganised directory structure; moved attendance.txt and speakers.txt
-----
r2 | jmackey | 2012-07-12 22:22:56 +0200 (Thu, 12 Jul 2012) | 1 line
Added Rob and J.Mackey to list of speakers. Added Rob and J.Mackey to new file keeping track of attendance.
-----
r1 | jmackey | 2012-07-12 22:20:48 +0200 (Thu, 12 Jul 2012) | 1 line
initial import of new project
-----
jmackey@aibn129:~/working_copy/tech_seminar$ svn log -rHEAD attendees/attendance.txt
-----
r3 | jmackey | 2012-07-12 22:42:43 +0200 (Thu, 12 Jul 2012) | 1 line
Reorganised directory structure; moved attendance.txt and speakers.txt
-----
jmackey@aibn129:~/working_copy/tech_seminar$ svn log -r1 attendees/attendance.txt
svn: Unable to find repository location for 'file:///users/jmackey/temp_svn_repo_store/RobsTechnicalSeminar/tech_seminar/attendees/attendance.txt' in revision 1
jmackey@aibn129:~/working_copy/tech_seminar$ svn log -r1 topics_speakers/speakers.txt
-----
r1 | jmackey | 2012-07-12 22:20:48 +0200 (Thu, 12 Jul 2012) | 1 line
initial import of new project
-----
```

Discarding changes

- Use “svn status [filename]” to query if a file has been changed.
- Use “svn diff [filename]” to show the changes.
- Use “svn revert filename” to discard changes and go back to last saved revision.



```
Terminal Shell Edit View Window Help
Terminal — ssh — 127x30
ssh
jmackey@aibn129:~/working_copy/tech_seminar$ # Checking Modifications, and discarding changes.
jmackey@aibn129:~/working_copy/tech_seminar$ echo "hello" >> attendees/attendance.txt
jmackey@aibn129:~/working_copy/tech_seminar$ svn status .
M      attendees/attendance.txt
jmackey@aibn129:~/working_copy/tech_seminar$ svn diff attendees/attendance.txt
Index: attendees/attendance.txt
-----
--- attendees/attendance.txt      (revision 3)
+++ attendees/attendance.txt      (working copy)
@@ -2,3 +2,4 @@
-----
Rob: attends every week.  punctual.  contributes questions.  A
J.Mackey: usually attends.  always late (even for own talk).  C-
+hello
jmackey@aibn129:~/working_copy/tech_seminar$ svn revert attendees/attendance.txt
Reverted 'attendees/attendance.txt'
jmackey@aibn129:~/working_copy/tech_seminar$ svn status .
jmackey@aibn129:~/working_copy/tech_seminar$ svn diff attendees/attendance.txt
jmackey@aibn129:~/working_copy/tech_seminar$
```

Checking out a project remotely

- `svn co file:///path/to/repo/NAME /path/to/working/copy/`
- `svn commit -m "Message for log file"`
- Alternative ways to do these things:
 - Check out remotely with `svn+ssh`:
Set the variable `SVN_SSH` (e.g. in `.bashrc`), or on command line:

```
export SVN_SSH="ssh -p1234"  
svn co svn+ssh://jmackey@gate1.astro.uni-bonn.de/  
users/jmackey/temp_svn_repo_store/  
RobsTechnicalSeminar working_copy
```
 - Notes: you need the full path following the URL, and all one line.
 - Commit using a text editor:
Set the variable `SVN_EDITOR` in e.g. `~/.bashrc`

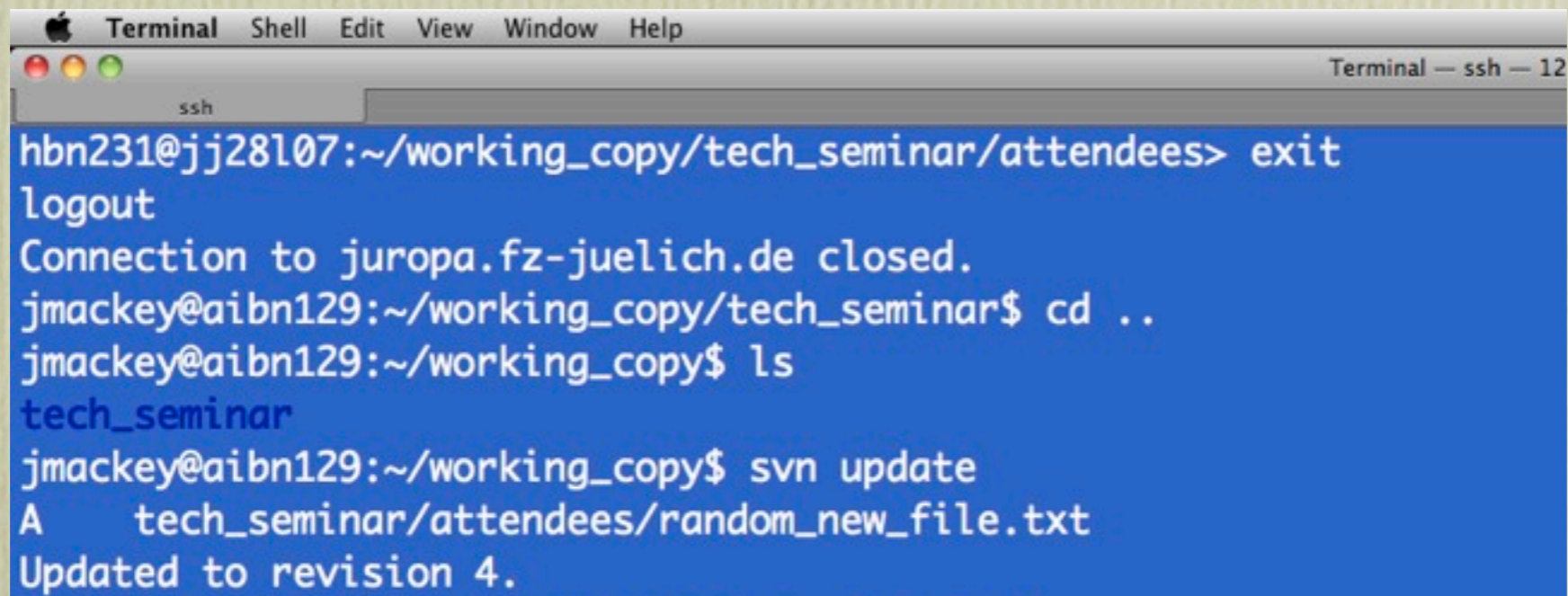
```
export SVN_EDITOR=/usr/bin/vim  
svn commit
```


Working with a second copy

```
Terminal Shell Edit View Window Help
Terminal — ssh — 127x30
ssh
hbn231@jj28l07:~> SVN_SSH="ssh -p1234" svn co svn+ssh://jmackey@gate1.astro.uni-bonn.de/users/jmackey/temp_svn_repo_store/Robst
echnicalSeminar working_copy
jmackey@gate1.astro.uni-bonn.de's password:
jmackey@gate1.astro.uni-bonn.de's password:
A    working_copy/tech_seminar
A    working_copy/tech_seminar/attendees
A    working_copy/tech_seminar/attendees/attendance.txt
A    working_copy/tech_seminar/topics_speakers
A    working_copy/tech_seminar/topics_speakers/speakers.txt
Checked out revision 3.
hbn231@jj28l07:~> cd working_copy/tech_seminar/attendees
hbn231@jj28l07:~/working_copy/tech_seminar/attendees> echo "random text" >> random_new_file.txt
hbn231@jj28l07:~/working_copy/tech_seminar/attendees> svn status .
?    random_new_file.txt
hbn231@jj28l07:~/working_copy/tech_seminar/attendees> svn commit
hbn231@jj28l07:~/working_copy/tech_seminar/attendees> svn add random_new_file.txt
A    random_new_file.txt
hbn231@jj28l07:~/working_copy/tech_seminar/attendees> svn status .
A    random_new_file.txt
hbn231@jj28l07:~/working_copy/tech_seminar/attendees> svn status -u
jmackey@gate1.astro.uni-bonn.de's password:
jmackey@gate1.astro.uni-bonn.de's password:
A    0    random_new_file.txt
Status against revision:    3
hbn231@jj28l07:~/working_copy/tech_seminar/attendees> svn commit
jmackey@gate1.astro.uni-bonn.de's password:
Adding    attendees/random_new_file.txt
Transmitting file data .
Committed revision 4.
hbn231@jj28l07:~/working_copy/tech_seminar/attendees> exit
```

Now update first working copy

- just do: `svn update`
- That's it!

A screenshot of a macOS Terminal window. The window title is "Terminal" and it has standard macOS window controls (red, yellow, green buttons). The terminal shows a sequence of commands and outputs. It starts with a user logging out of an SSH session on a remote host. Then, the user returns to a local host and runs 'cd ..' to move to the parent directory. Next, 'ls' is run, showing a directory named 'tech_seminar'. Finally, 'svn update' is run, which outputs 'A tech_seminar/attendees/random_new_file.txt' and 'Updated to revision 4.'

```
Terminal Shell Edit View Window Help
Terminal - ssh - 12
ssh
hbn231@jj28107:~/working_copy/tech_seminar/attendees> exit
logout
Connection to juropa.fz-juelich.de closed.
jmackey@aibn129:~/working_copy/tech_seminar$ cd ..
jmackey@aibn129:~/working_copy$ ls
tech_seminar
jmackey@aibn129:~/working_copy$ svn update
A    tech_seminar/attendees/random_new_file.txt
Updated to revision 4.
```

Stuff can go wrong!

- If you have more than one active copy, and make lots of changes to both, and commit one, then sometimes SVN will get confused with the other.
- When you try to commit the second active copy, it will complain about conflicts, or even worse, “tree conflicts”.
- Usually only happens if you rename or move files/directories, and then edit them before committing.
- In this case, either just give up on one copy, or try to resolve the conflicts, or cut and paste the changes into the active copy which is not “in conflict”.
- Doesn't happen very often... if it does, look in the SVN book under “resolving conflicts”. I found it difficult to work with.

Summary

- I find SVN very useful both for code and organising documents relating to a project.
 - I have between 5 and 10 svn repositories for projects.
- You get to keep the full history of the project.
- You have a backup (and it is easy to backup the repository).
- Easy to keep code versions consistent on many machines.
- It's not that hard! And it is worth the effort.
- With about 10 commands you can already do all that you are likely to need in everyday use.
- Talk to IT staff about hosting projects on AlFA's http server (if you want to share it you can't put it on your home dir)