Source code management

Alexandre Bergel
abergel@dcc.uchile.cl
20/10/2011

Outline

1. Source code management
2. Subversion: centralized version control
3. GIT: decentralized version control

Outline

1. **Source code management**
2. Subversion: centralized version control
3. GIT: decentralized version control

Source code management

A version control system is a system that *tracks incremental versions* of files and directories

What makes a version control system interesting is exploring changes and facilitates the *arbitrary recall of the same*

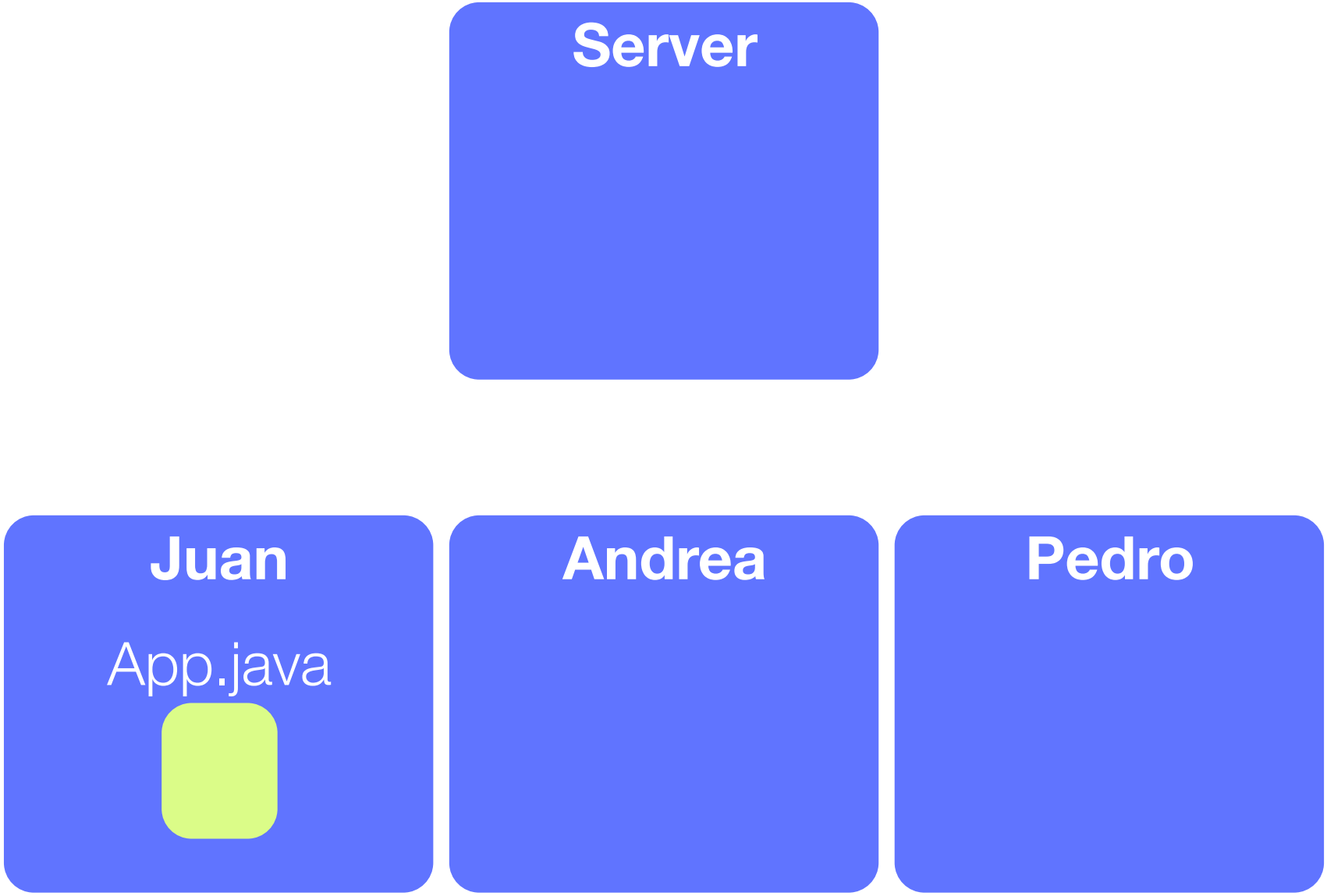
Source code management

Life lesson: *never* plan collaborative work *without* a source code management system

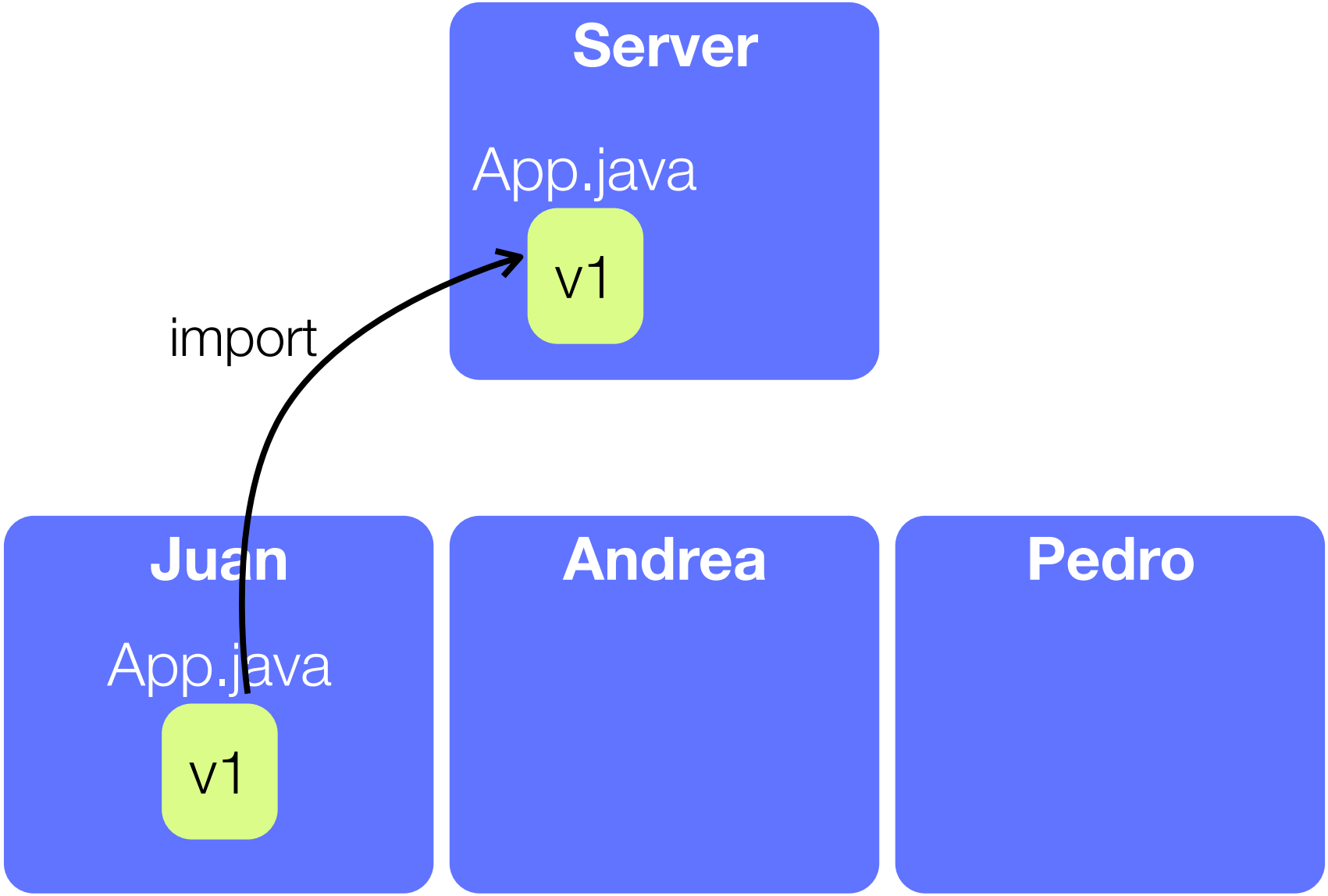
You are doomed to fail if you do so

Essential no only for source code, but for papers and document writing

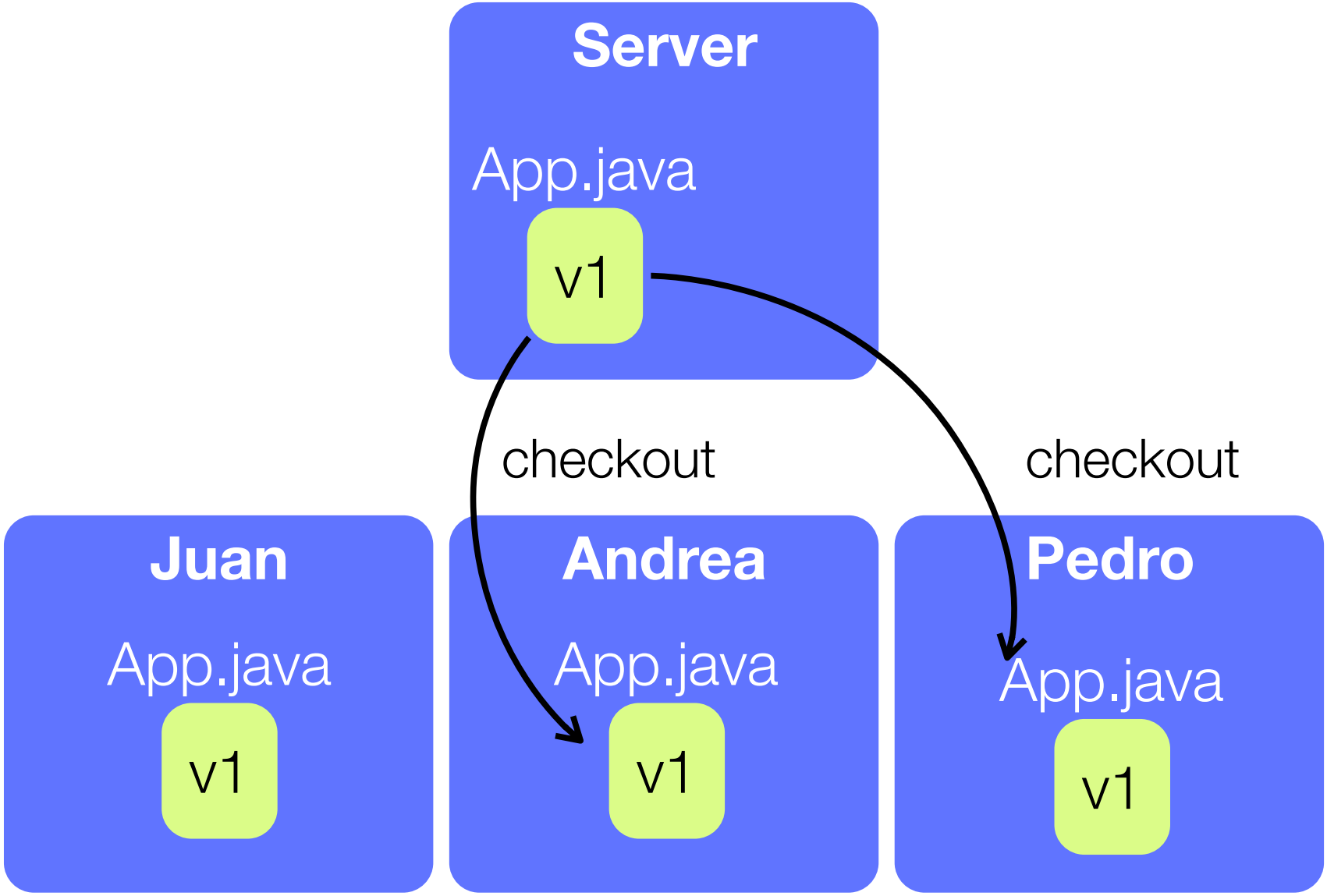
Illustration



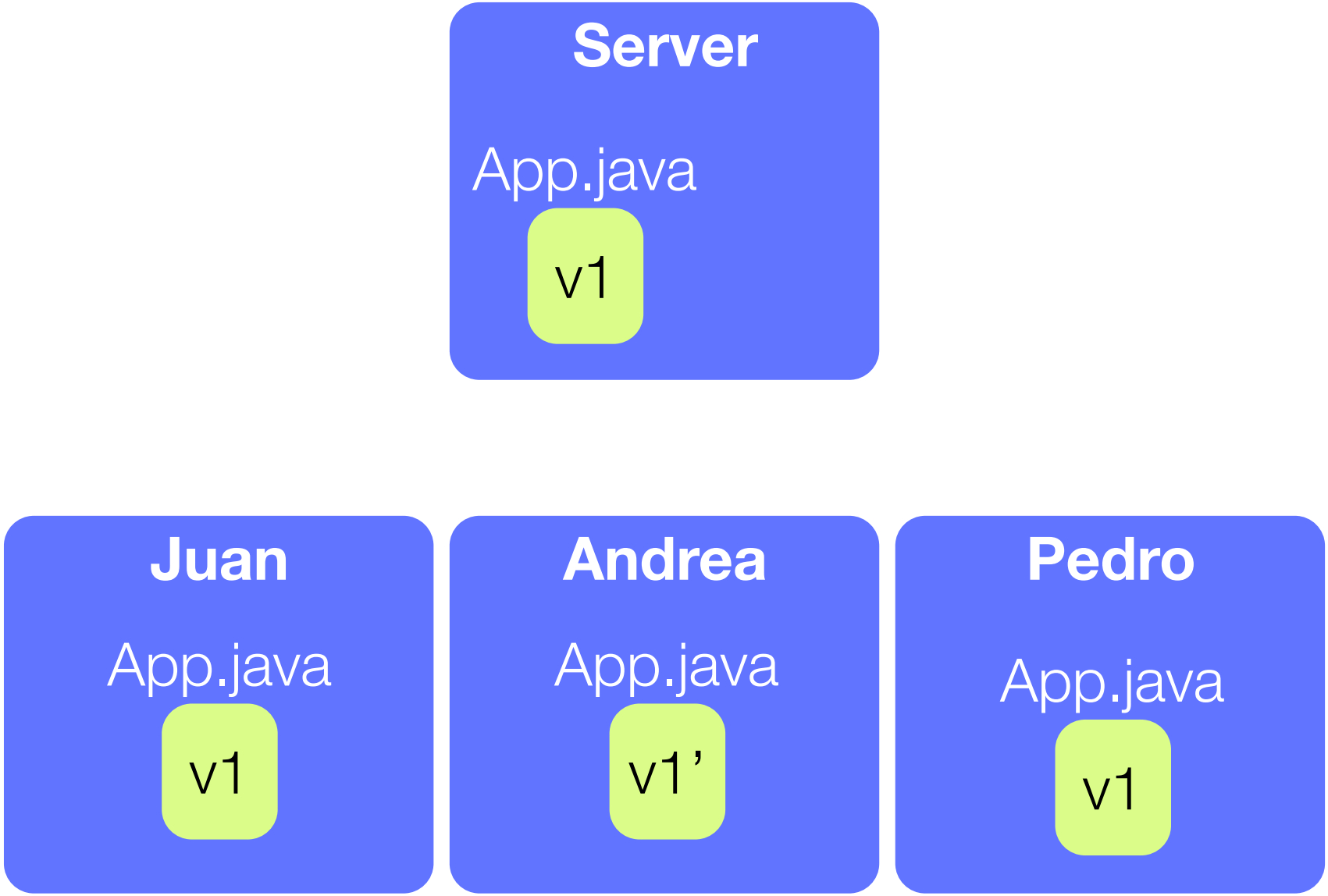
Illustration



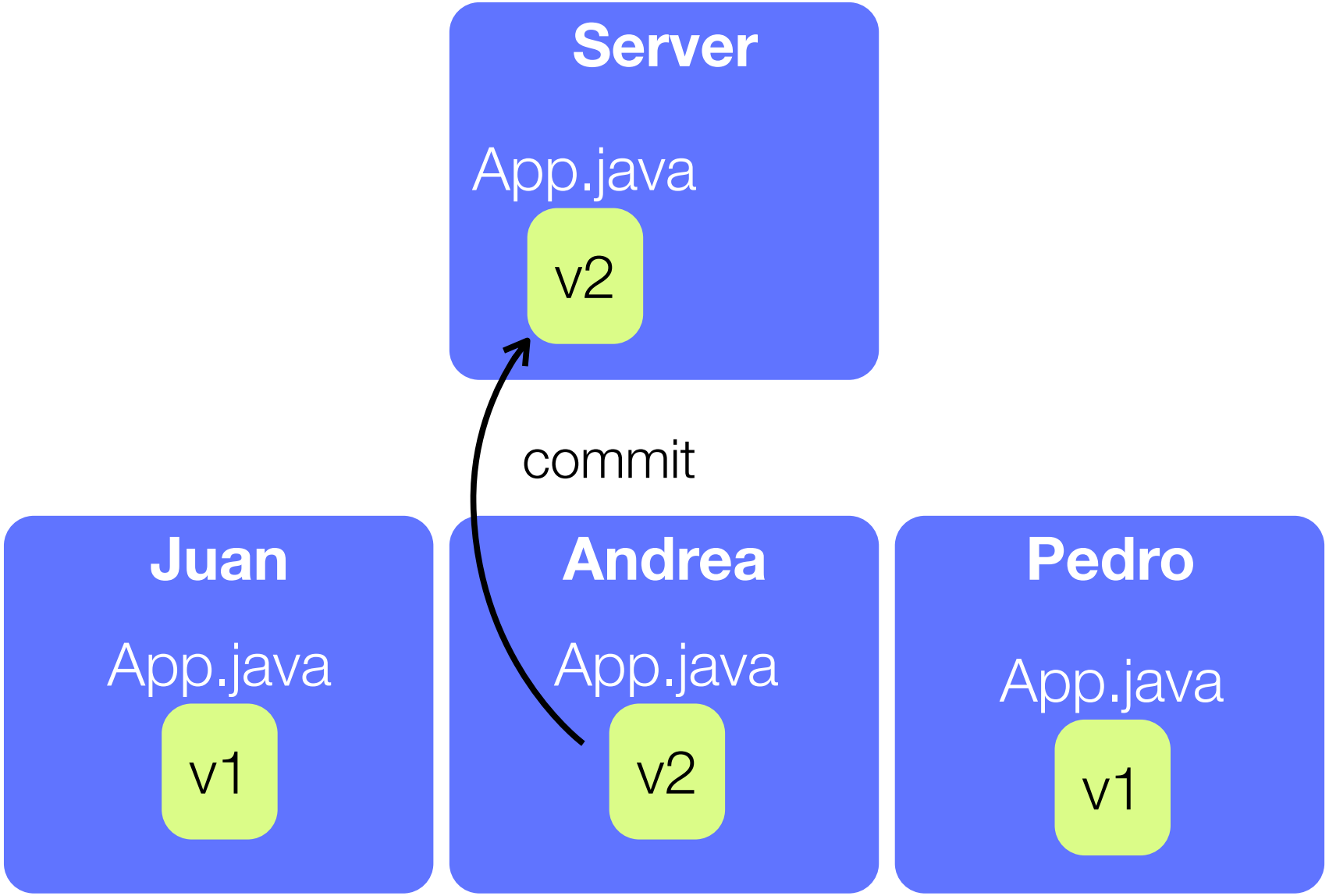
Illustration



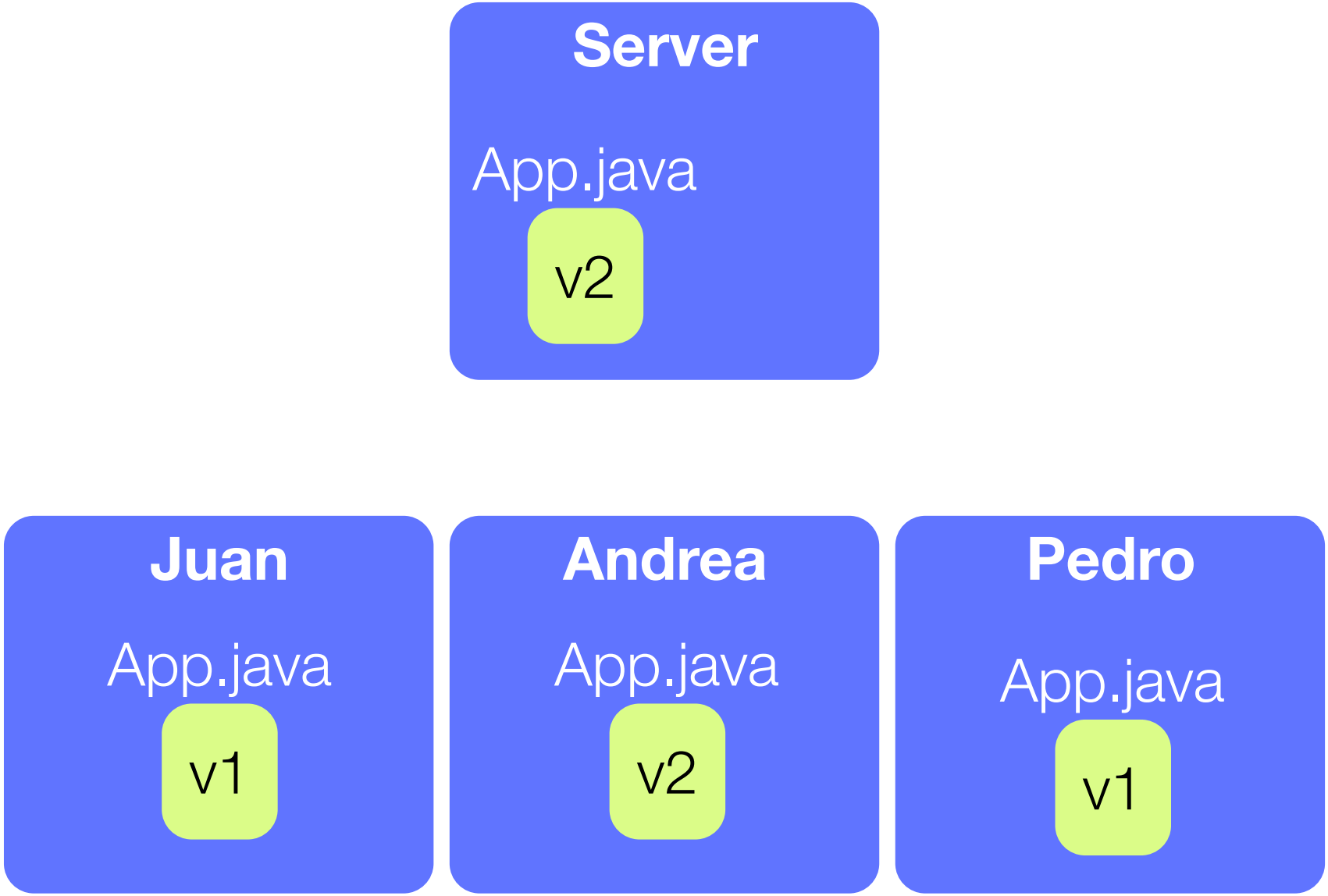
Illustration



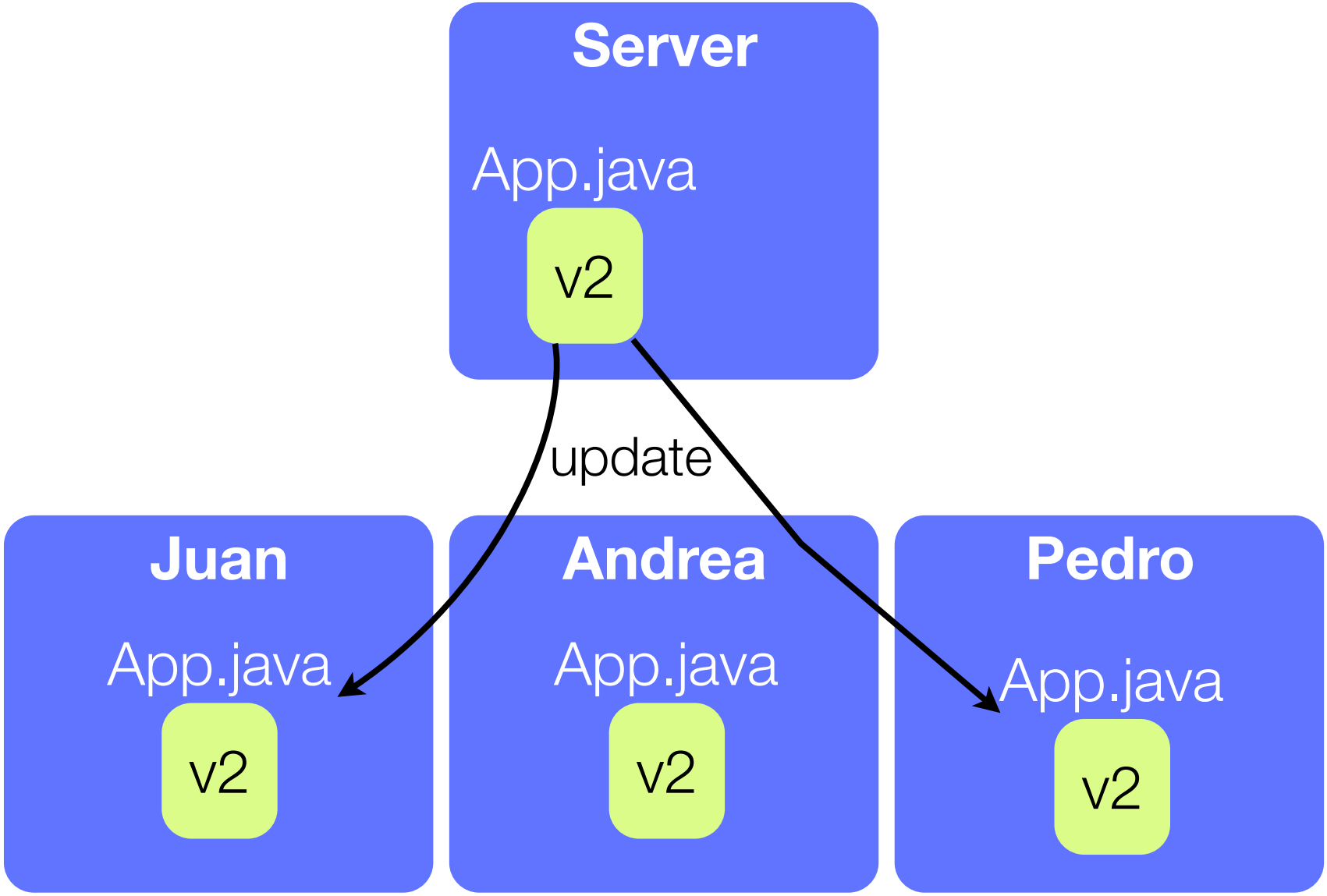
Illustration



Illustration



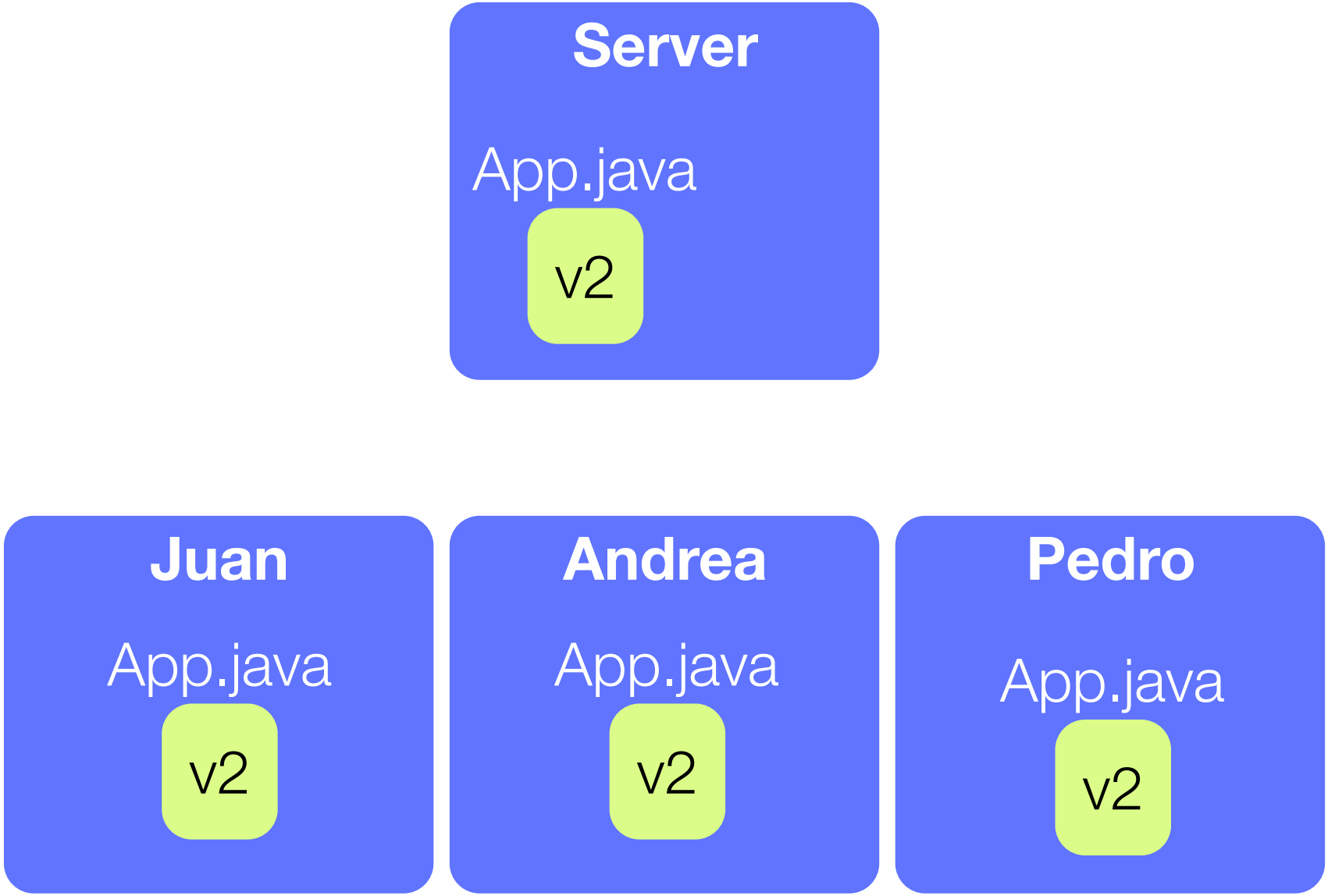
Illustration



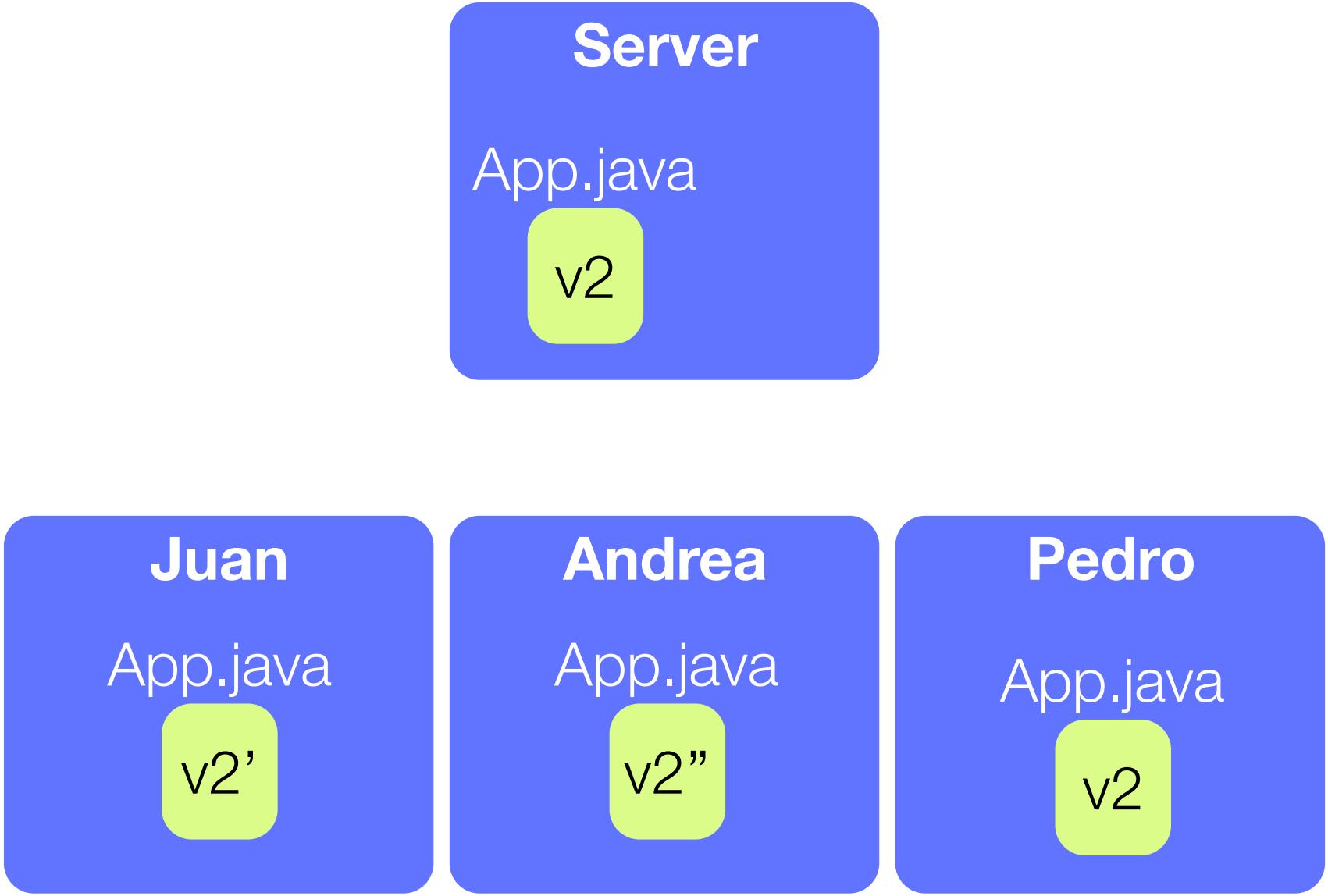
The fundamental problem

How to allow users to share information, but prevent them from accidentally stepping on each other's feet?

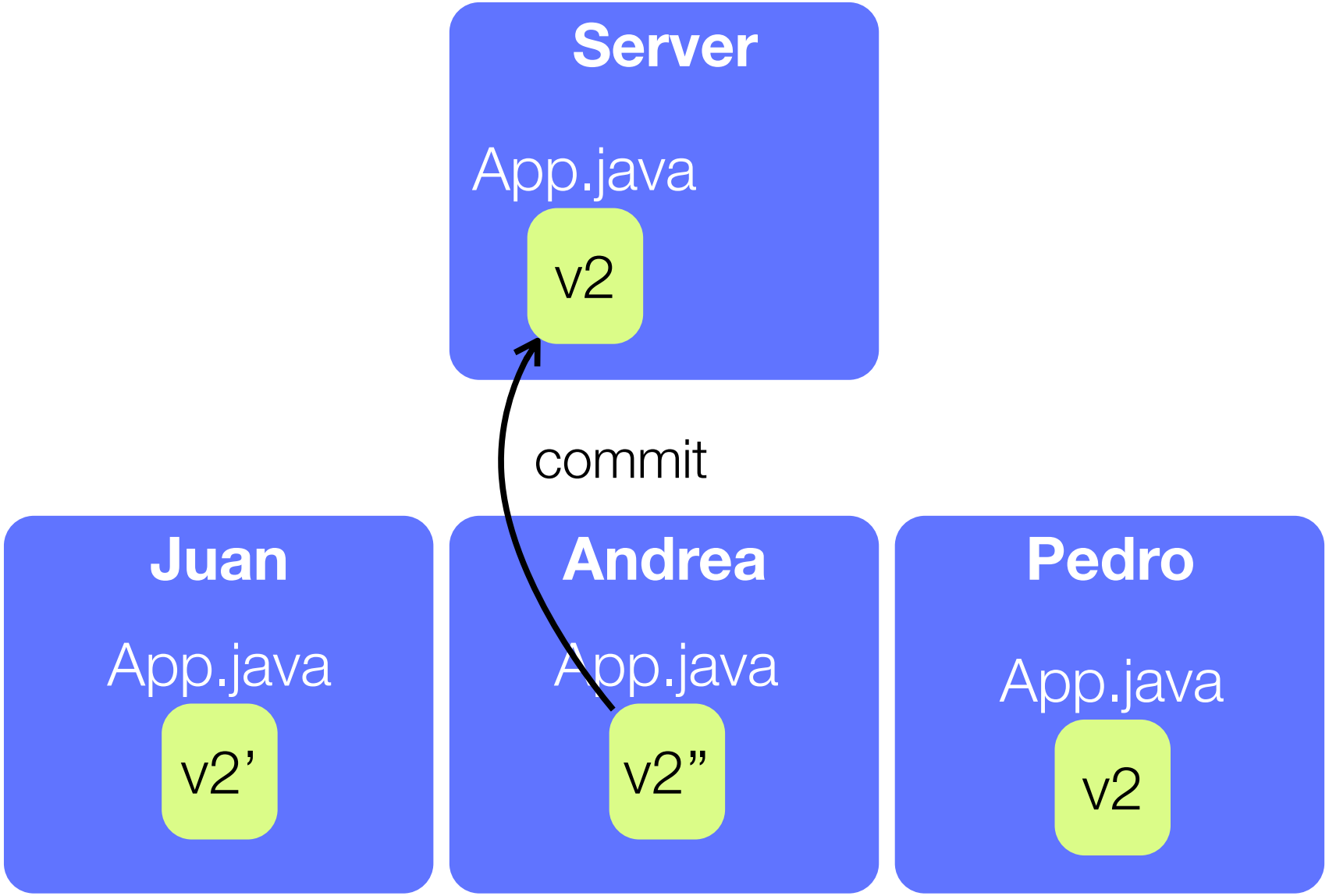
Illustration



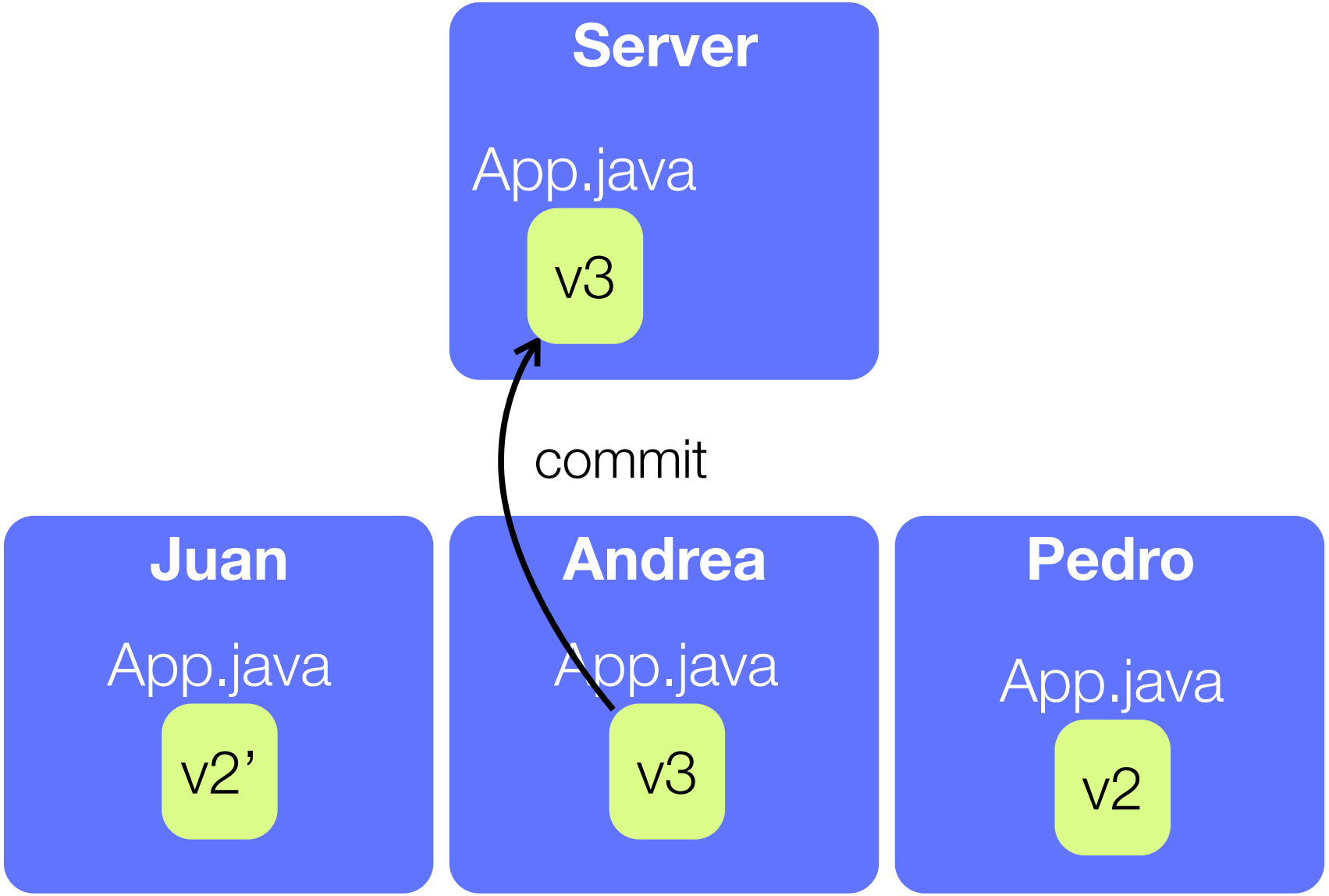
Illustration



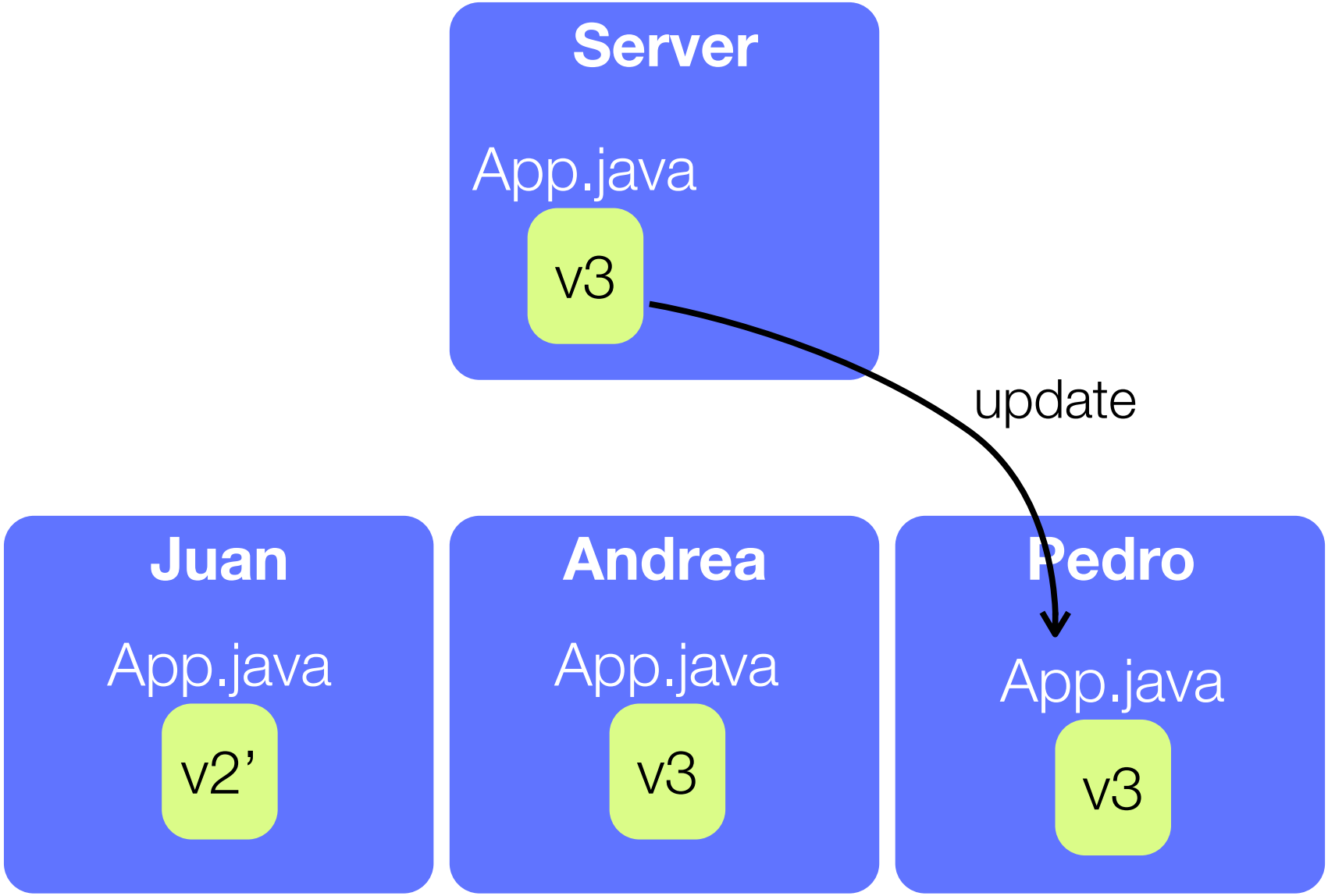
Illustration



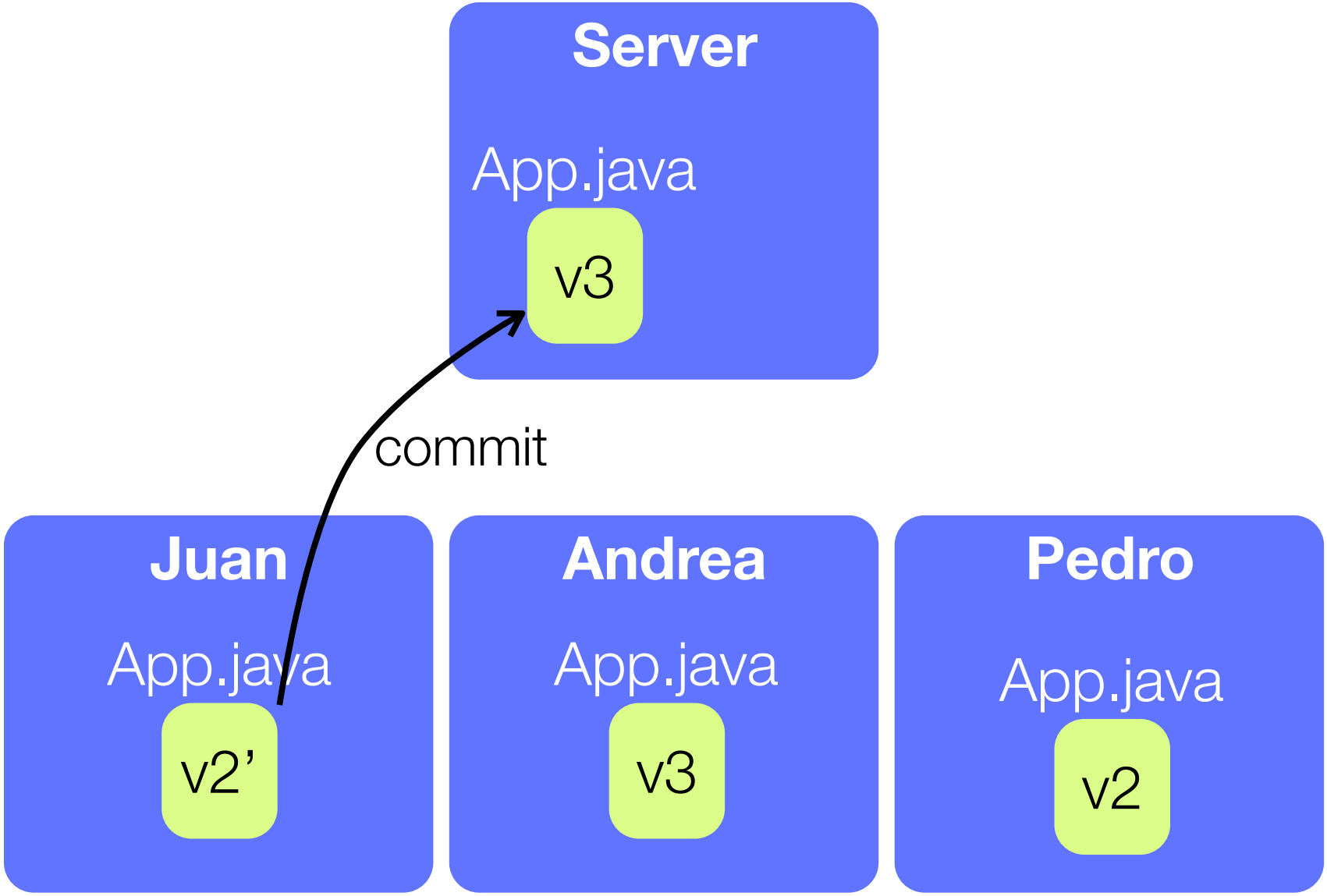
Illustration



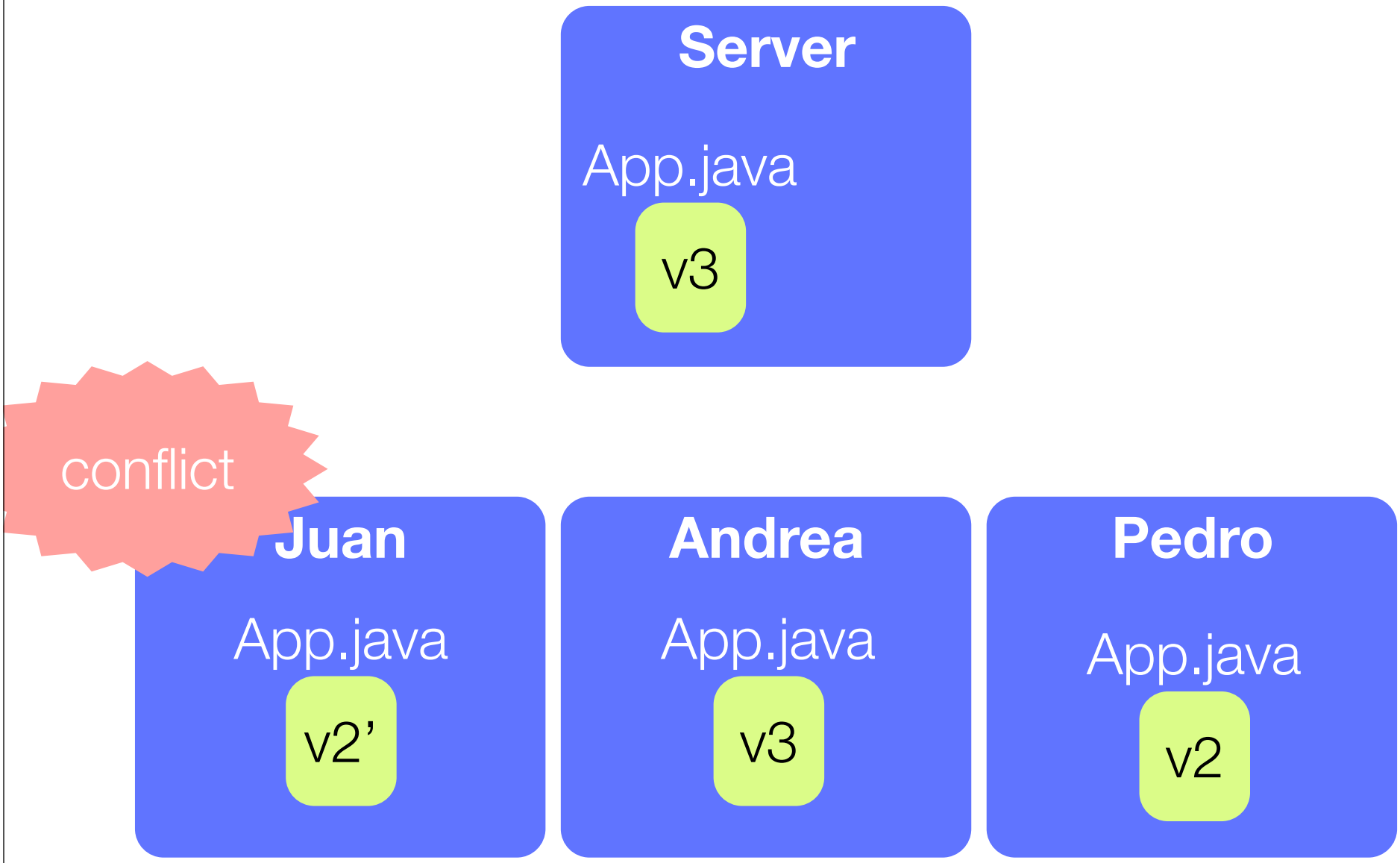
Illustration



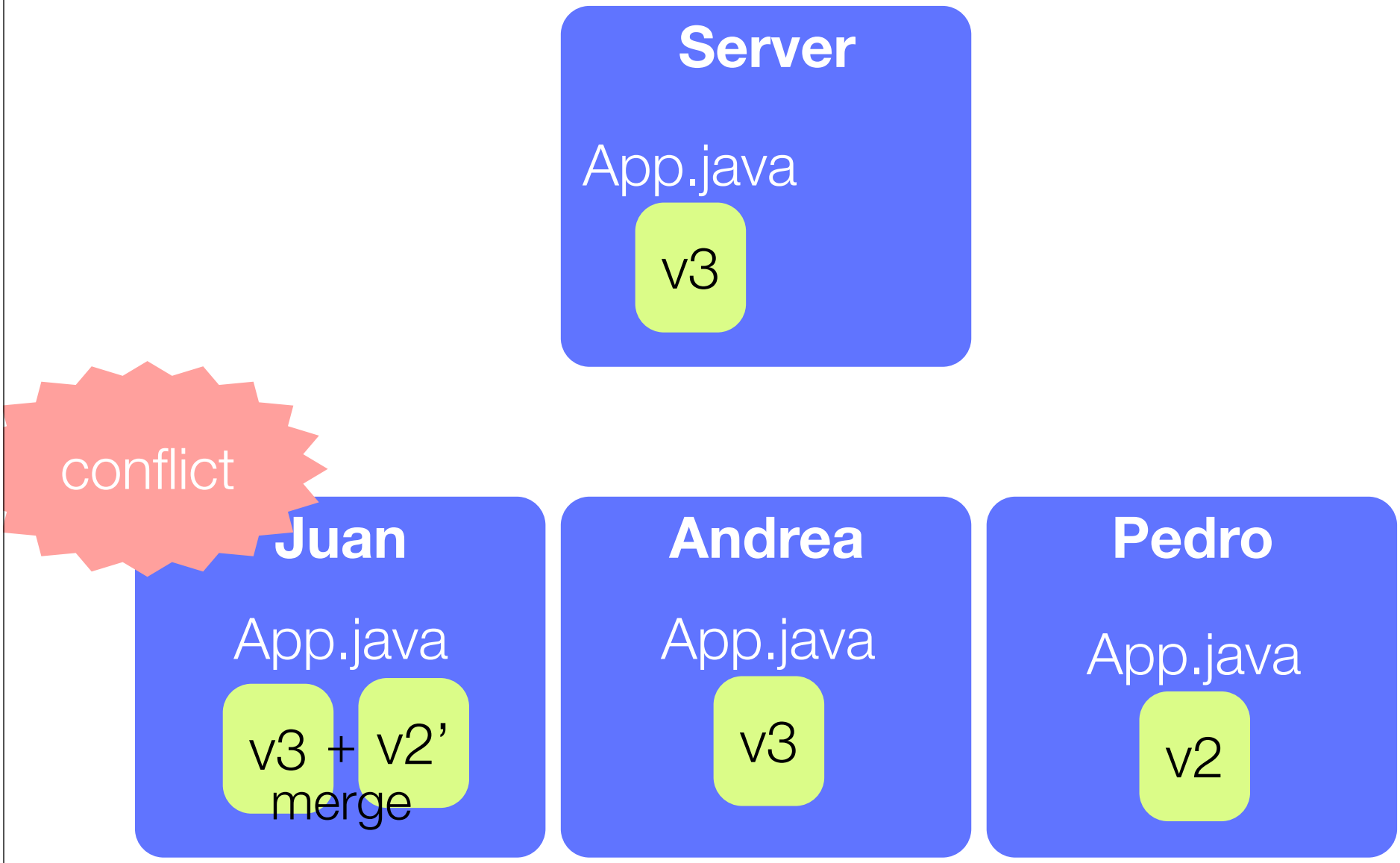
Illustration



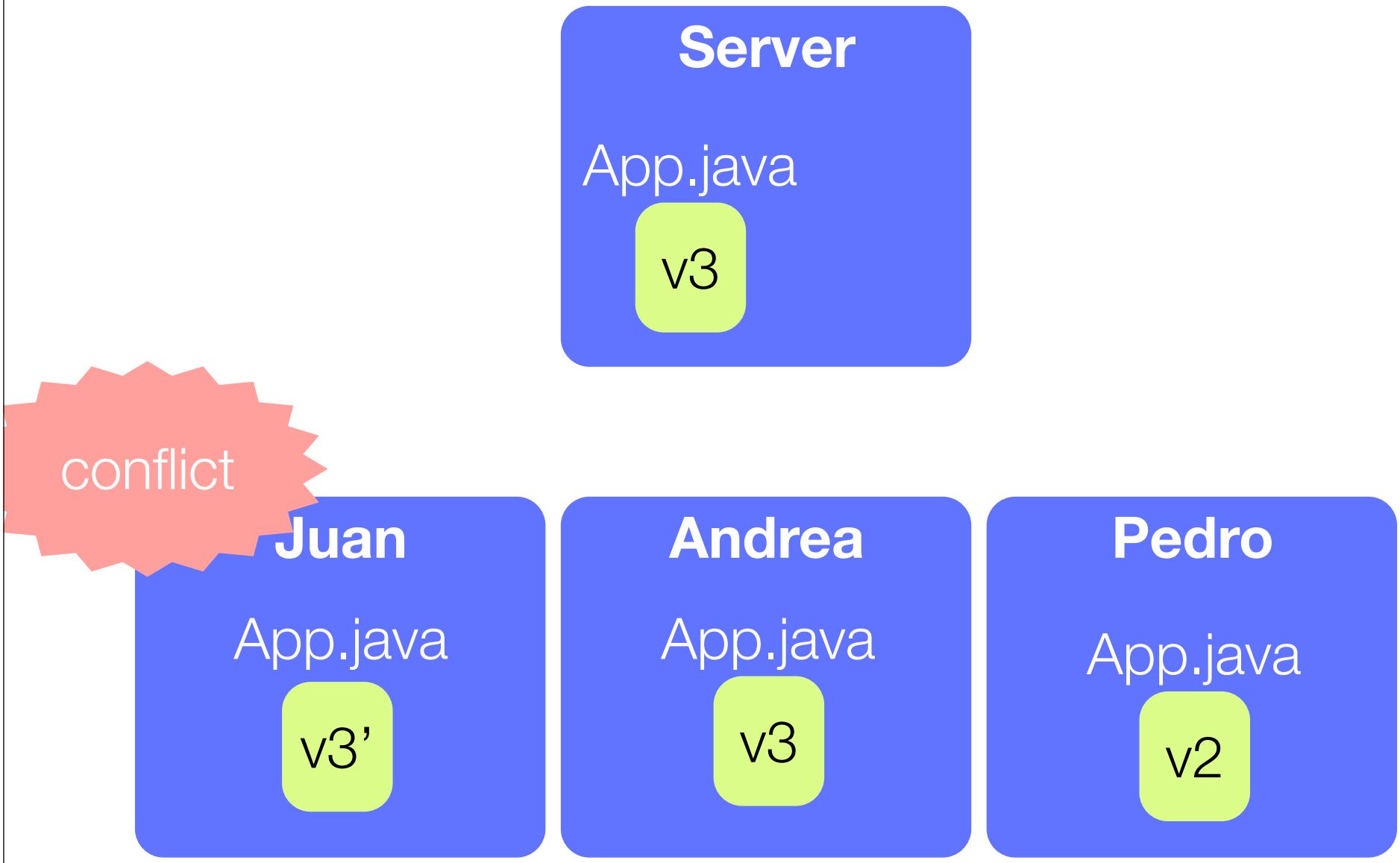
Illustration



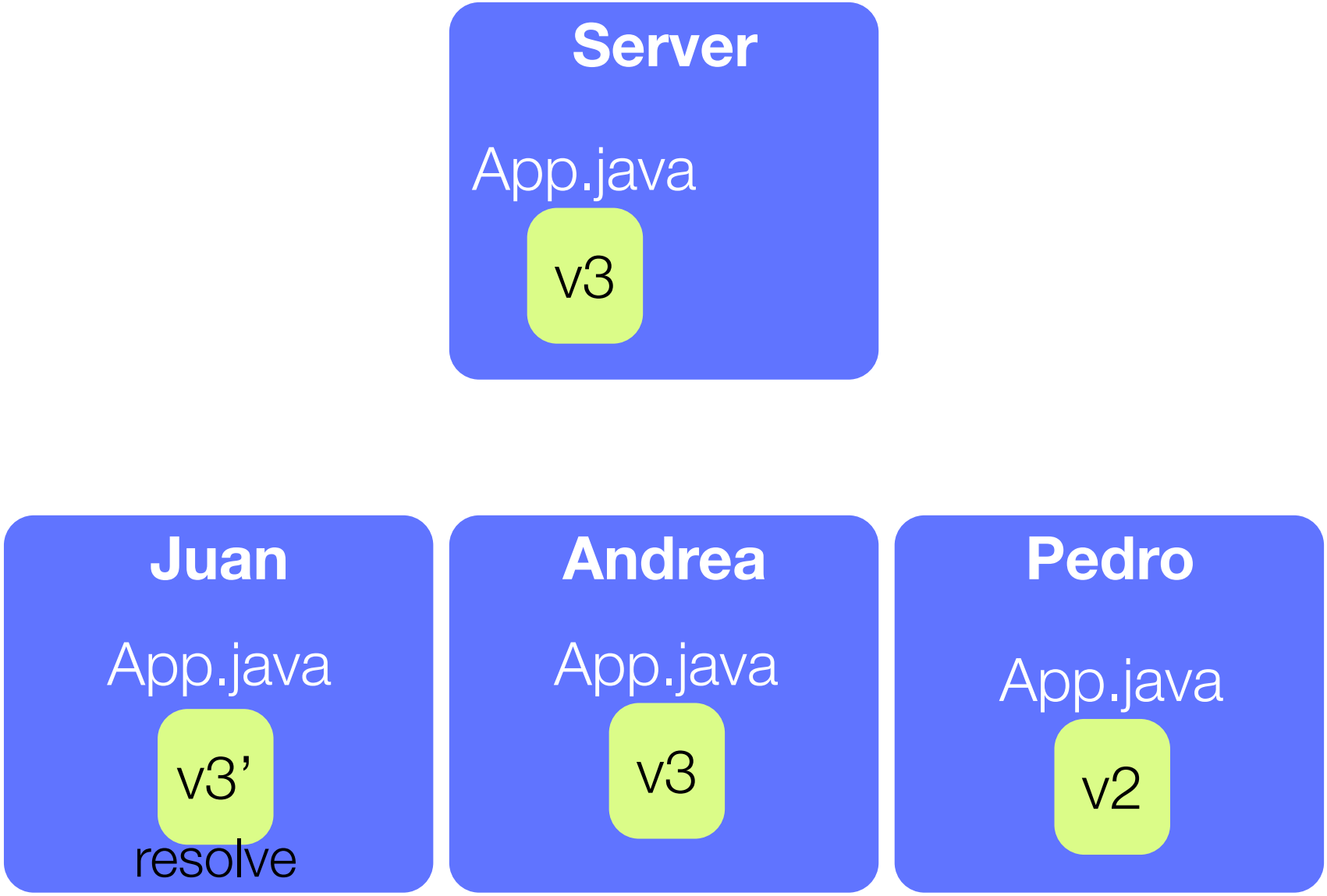
Illustration



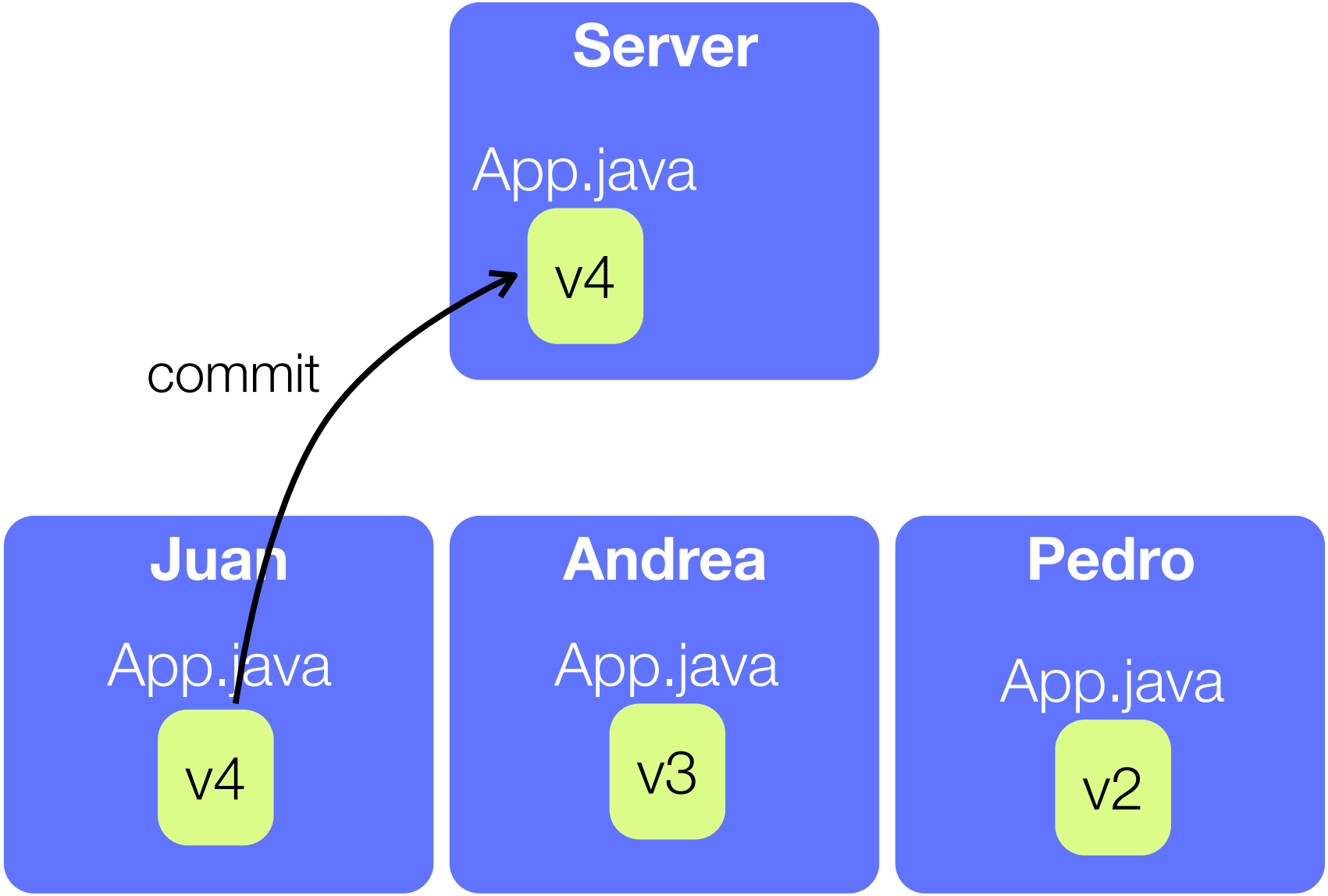
Illustration



Illustration



Illustration



Outline

1. Source code management
2. **Subversion: centralized version control**
3. GIT: decentralized version control

Subversion

Excellent book

http://svnbook.red-bean.com/

Successor of CVS

Widely used in open source project and industry

Accessing the repository

Example to type in your xterm or terminal

svn checkout http://svn.example.com:9834/repos

Repository access URLs

Direct repository access (on local disk): files:///

WebDAV protocol to Subversion-aware apache: http://

SSL encryption: https://

Custom protocol: svn://

SSH tunnel: svn+ssh://

Help!

```
$ svn help
Subversion command-line client, version 1.6.13.
Type 'svn help <subcommand>' for help on a specific subcommand.
Type 'svn --version' to see the program version and RA modules
  or 'svn --version --quiet' to see just the version number.

Most subcommands take file and/or directory arguments, recursing
on the directories. If no arguments are supplied to such a
command, it recurses on the current directory (inclusive) by
default.

Available subcommands:
 add
 blame (praise, annotate, ann)
 cat
...
```

Import

Getting data into your repository

svn import /path/to/mytree http://host.com/svn -m “my comment”

Immediately after the commit, you can see your data

svn list http://host.com/svn

Important: after the import, your original local directly is *not* converted into a working copy!

Checkout

Getting a copy on your local machine

```
$ svn checkout http://host.example.com/svn/repo/trunk
A trunk/README
A trunk/INSTALL
A trunk/src/main.c
A trunk/src/header.h
...
Checked out revision 8810. $
```

Add

Adding file into the repository

```
svn add FOO
```

Remember to do a commit afterward

FOO can be a textual or binary file

Log

See the history

```
~/Workspace/translucent-objects> svn log main.tex | head
------------------------------------------------------------------------
r29 | ab | 2011-03-28 10:29:29 -0400 (Mon, 28 Mar 2011) | 2 lines
started to work on the introduction
------------------------------------------------------------------------
r28 | vranyj1 | 2011-03-23 15:16:56 -0400 (Wed, 23 Mar 2011) | 1 line
temporary commit - minor improvements in Implementation
------------------------------------------------------------------------
~/Workspace/translucent-objects>
```

This is what we will use to check your code

Commit

Push your change toward the server

```
svn commit
```

You may want to provide a file name to push a particular file

Update

Update your local copy

$ svn update

U INSTALL

G README

Conflict discovered in 'bar.c'.

Select: (p) postpone, (df) diff-full, (e) edit, ...

Update

An easy way to solve conflicts, is to postpone the resolution

Press “p” when you have a conflict upon an update

Open the file in conflict

Example of a conflicted file

```
$ cat sandwich.txt
Top piece of bread
Mayonnaise
Lettuce
Tomato
Provolone
<<<<<<< .mine
Salami
Mortadella
Prosciutto
=======
Sauerkraut
Grilled Chicken
>>>>>>> .r2
Creole Mustard
Bottom piece of bread
```

Example of a conflicted file

```
$ cat sandwich.txt
Top piece of bread
Mayonnaise
Lettuce
Tomato
Provolone
<<<<<<< .mine
Salami
Mortadella
Prosciutto
=======
Sauerkraut
Grilled Chicken
>>>>>>> .r2
Creole Mustard
Bottom piece of bread
```

What you have entered

What is on the server

Solving the conflict

Edit the file and merge manually the contents on the two parts of ===

Remove the <<<<<, >>>>>>, and ====

You need to tell SVN that the conflict is solved:

```
svn resolve sandwich.txt
```

Specifying username and password

Simply use --username and --password

svn co --username pedro --password wna http://host.com

Operation on files

you should not directly perform “mv” for moving or renaming files

Instead, use “svn mv file1 file2”

Else, it will get messy

Survival to Vl

Vl may be one of the oldest programs running on your machine

It is often used as the default text editor by SVN

Simple text editor

For editing a file, press “i”, then enter your text

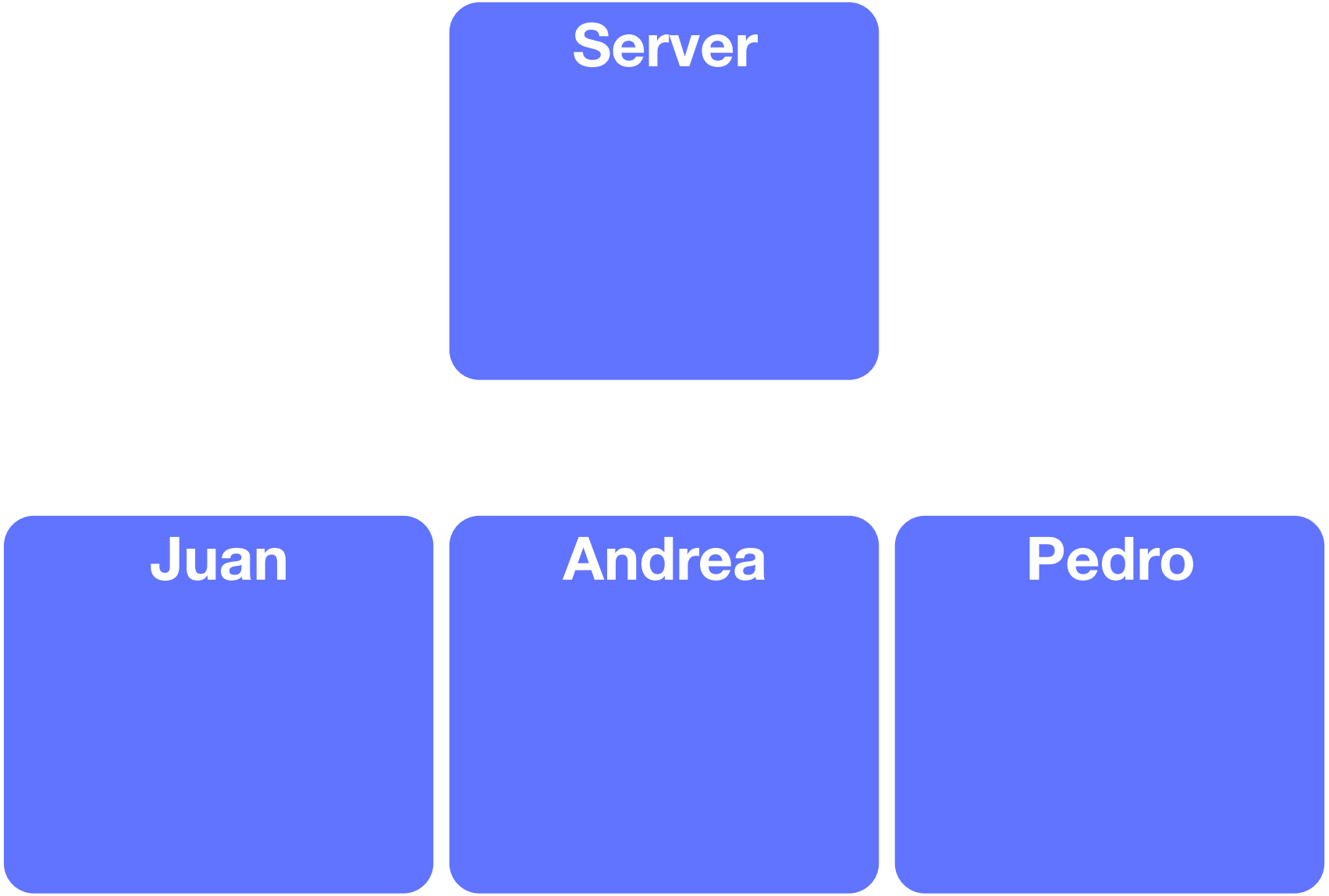
For saving and quitting

Press the escape key, then type :wq

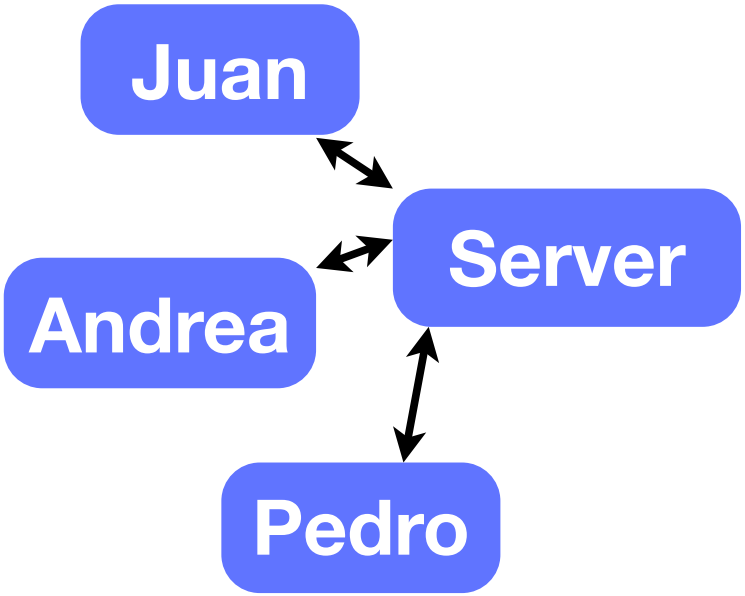
Outline

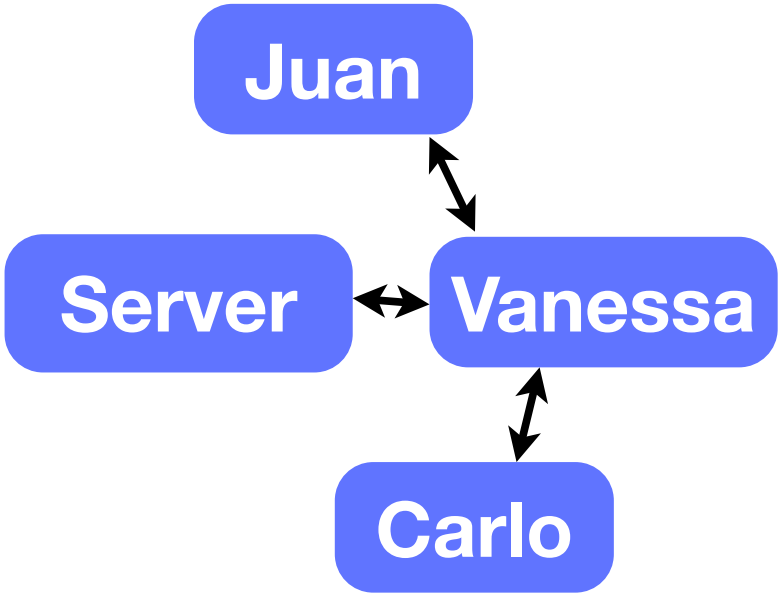
1. Source code management
2. Subversion: centralized version control
3. **GIT: decentralized version control**

Typical small project

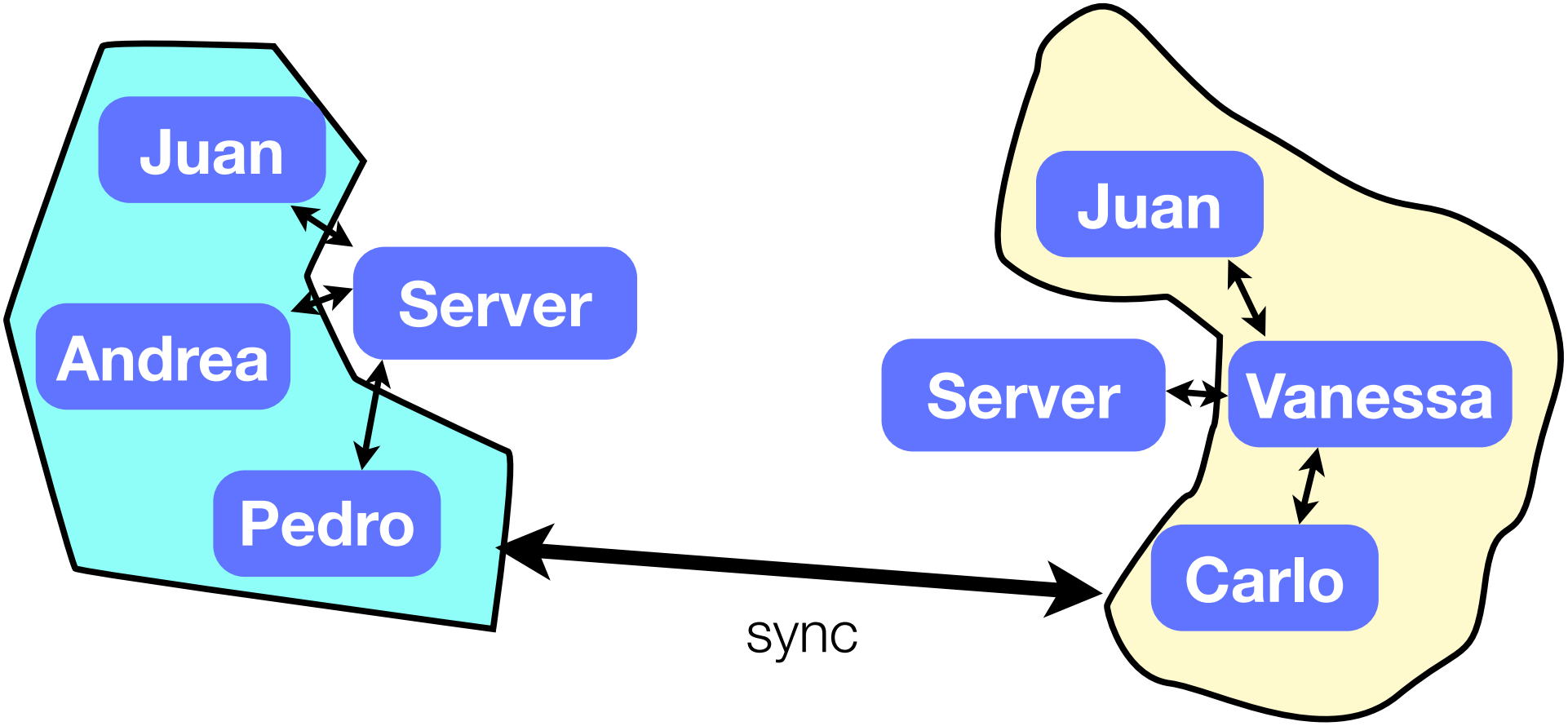


What may happen on a large project





What may happen on a large project



Multi server project

GIT emerges from the need expressed by the Linux community

Often, part of a software ecosystem has to live in a separate repository

In Pharo, there is squeaksource, gemstone.source and many private repository servers

GIT in brief

GIT is similar to SVN

The main difference is about server synchronization

You can explicitly pull and push your commit on a designated server

github.com is a popular public GIT server

GIT survival commands

Create a local copy (on your computer) of a repository

```
git clone git@github.com:bergel/2011-RefactoringWithAOP.git
```

Updating your local copy

```
git pull
```

Committing your change (local commit)

```
git add main.tex MyClass.java ...
```

```
git commit
```

Pushing your change to the server

```
git push
```

Resolving conflict

```
git mergetool
```

Conclusion

Although SVN and GIT offer a poor way to version source code, it is reliable and works well

Do not hesitate to use a fancy graphical tool to manage the version control system