Version control

Michael Tsai

2012/4/10

Reference

- http://betterexplained.com/articles/a-visual-guide-to-version-control/
- http://www.ericsink.com/scm/source_control.html
- http://betterexplained.com/articles/intro-to-distributed-version-control-illustrated/

Version control

- Also called “source control”
- Other alias
  - source configuration management
  - source code management

You have been doing it

- Your own version control system:
  - KalidAzadResumeOct2006.doc
  - KalidAzadResumeMar2007.doc
  - instacalc-logo3.png
  - instacalc-logo4.png
  - logo-old.png
- "Save as": leave the old version intact.
  - Single backup file: **Document.old.doc**
  - Version number or date: **Document_V1.doc**
  - **Share folder** for other people to access/modify the file(s)

Still better than nothing!

But it doesn’t scale

- Imagine putting all the related files of a gigantic software project (e.g. WINDOWS 8) in a single shared folder, and have thousands of developers accessing it.
- NO WAY.

What does a version control system do?

- **Backup and Restore.** Files are saved as they are edited, and you can jump to any moment in time. Need that file as it was on Feb 23, 2007? No problem.
- **Synchronization.** Lets people share files and stay up-to-date with the latest version.
- **Short-term undo.** Monkeying with a file and messed it up? (That's just like you, isn't it?). Throw away your changes and go back to the "last known good" version in the database.

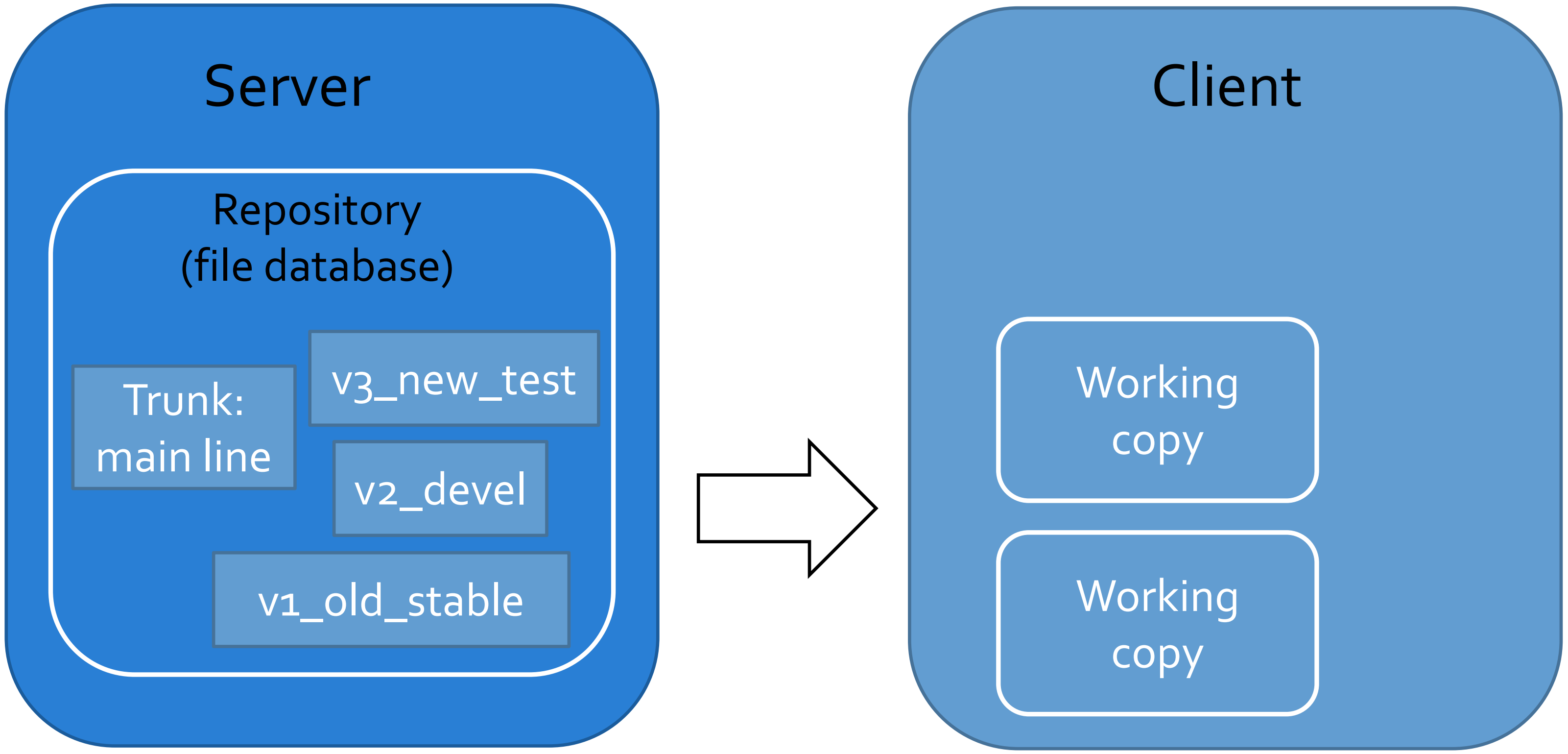
What does a version control system do?

- **Long-term undo.** Sometimes we mess up bad. Suppose you made a change a year ago, and it had a bug. Jump back to the old version, and see what change was made that day.
- **Track Changes**. As files are updated, you can leave messages explaining why the change happened (stored in the VCS, not the file). This makes it easy to see how a file is evolving over time, and why.
- **Track Ownership.** A VCS tags every change with the name of the person who made it. Helpful for ~~blamestorming~~ giving credit.

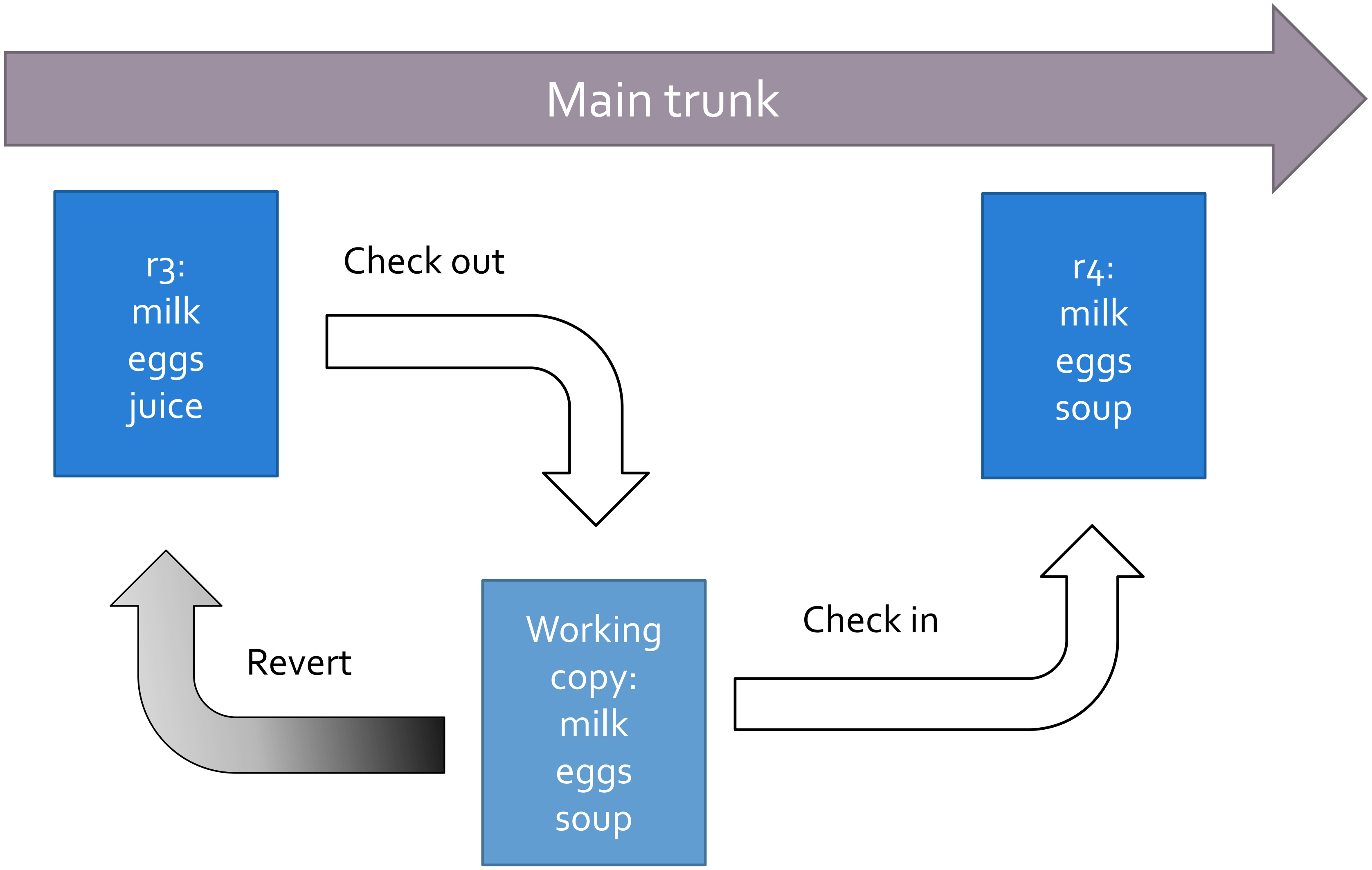
What does a version control system do?

- **Sandboxing**, or insurance against yourself. Making a big change? You can make temporary changes in an isolated area, test and work out the kinks before “checking in” your changes.
- **Branching and merging**. A larger sandbox. You can **branch** a copy of your code into a separate area and modify it in isolation (tracking changes separately). Later, you can **merge** your work back into the common area.

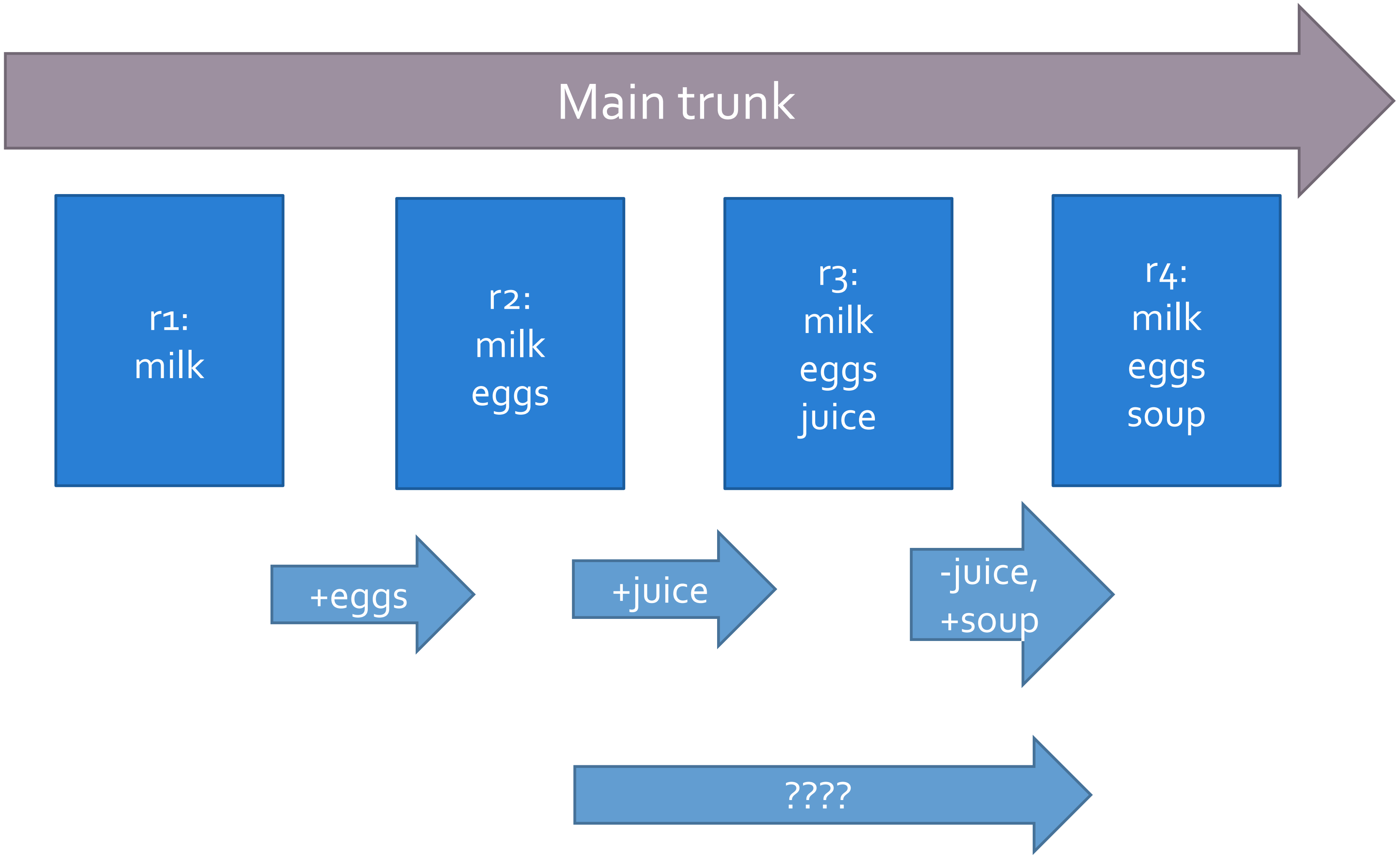
Learn some terms



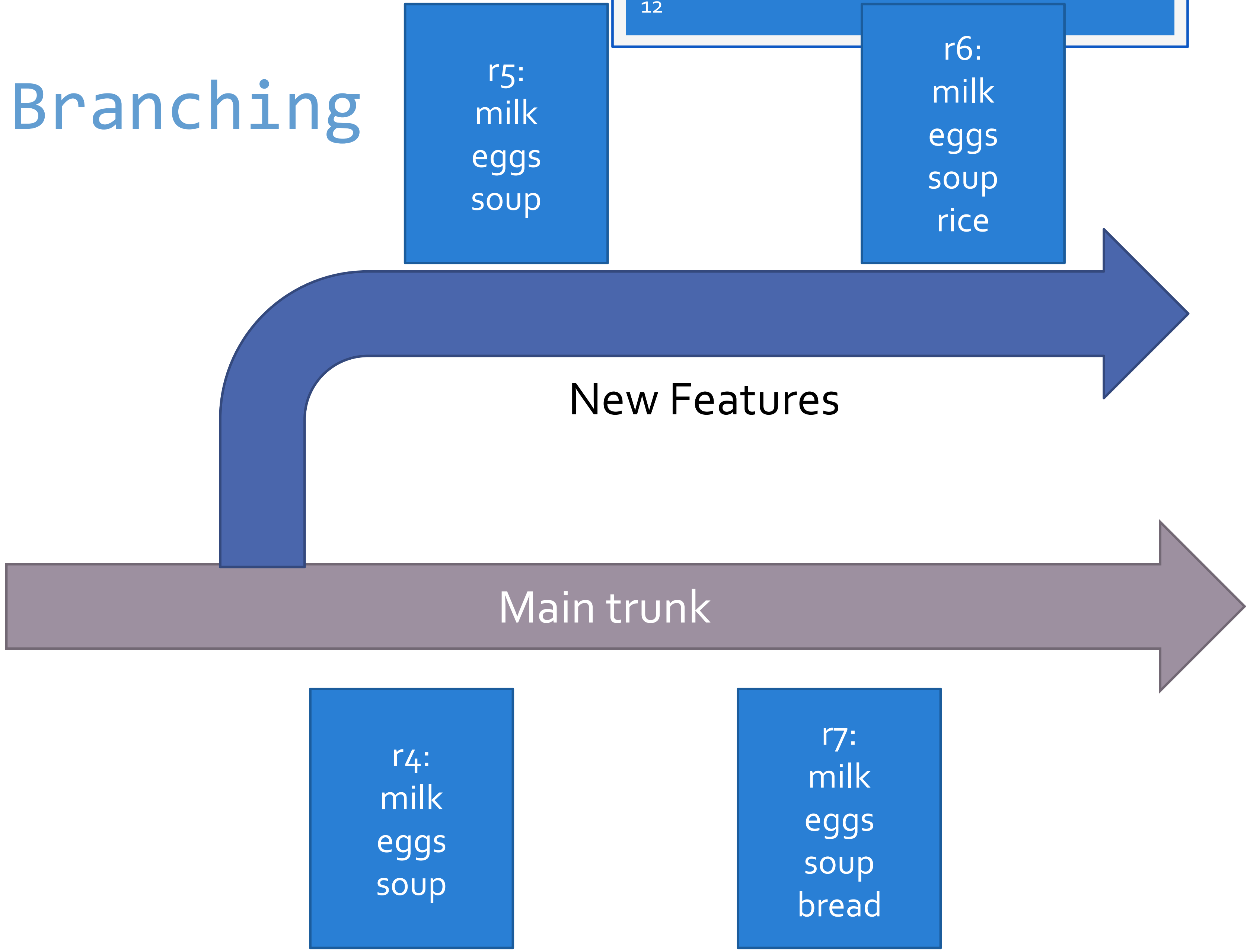
Checkout and Edit



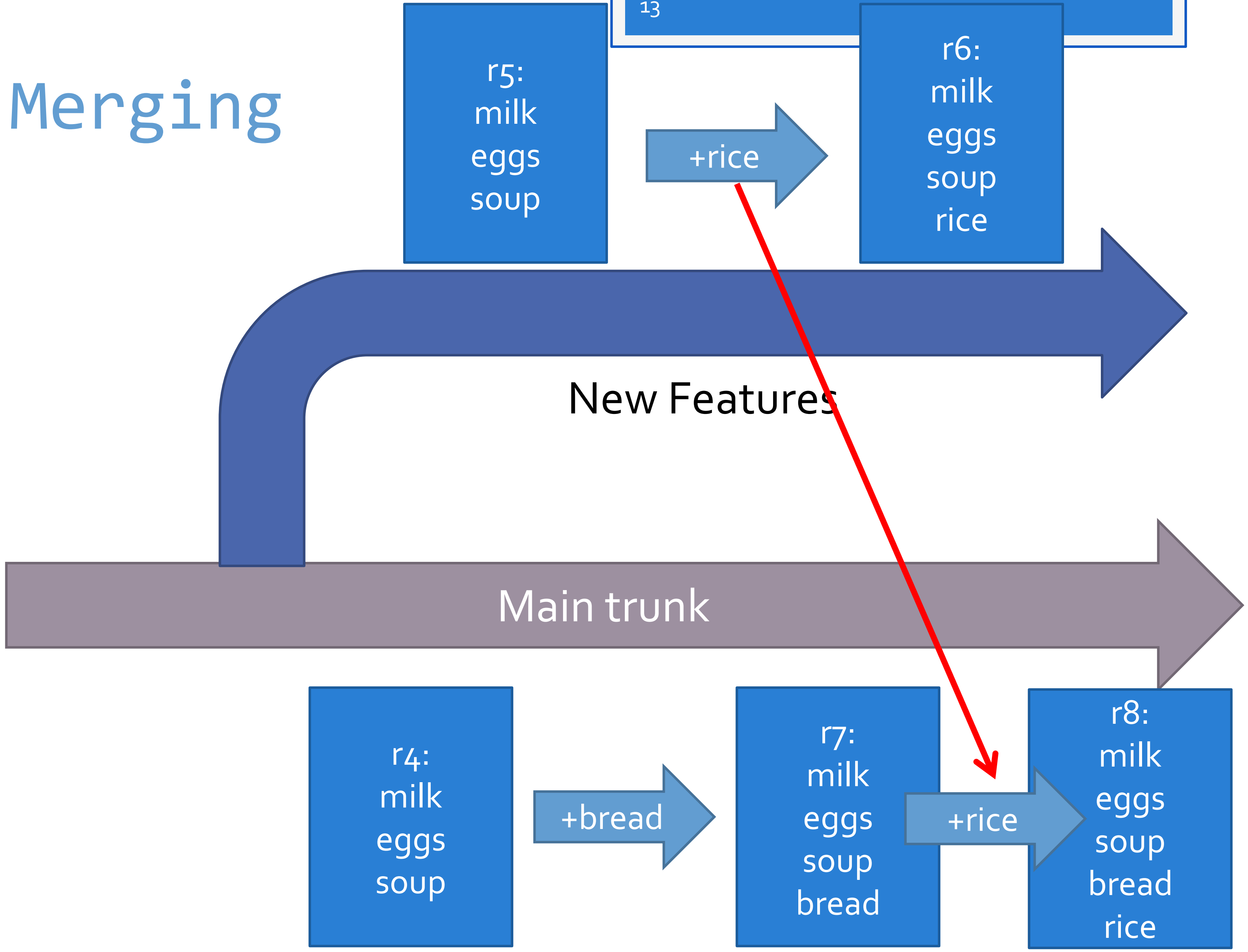
Basic Diffs


Main trunk
r1:
milk
r2:
milk
eggs
r3:
milk
eggs
juice
r4:
milk
eggs
soup
+eggs
+juice
-juice,
+soup
????

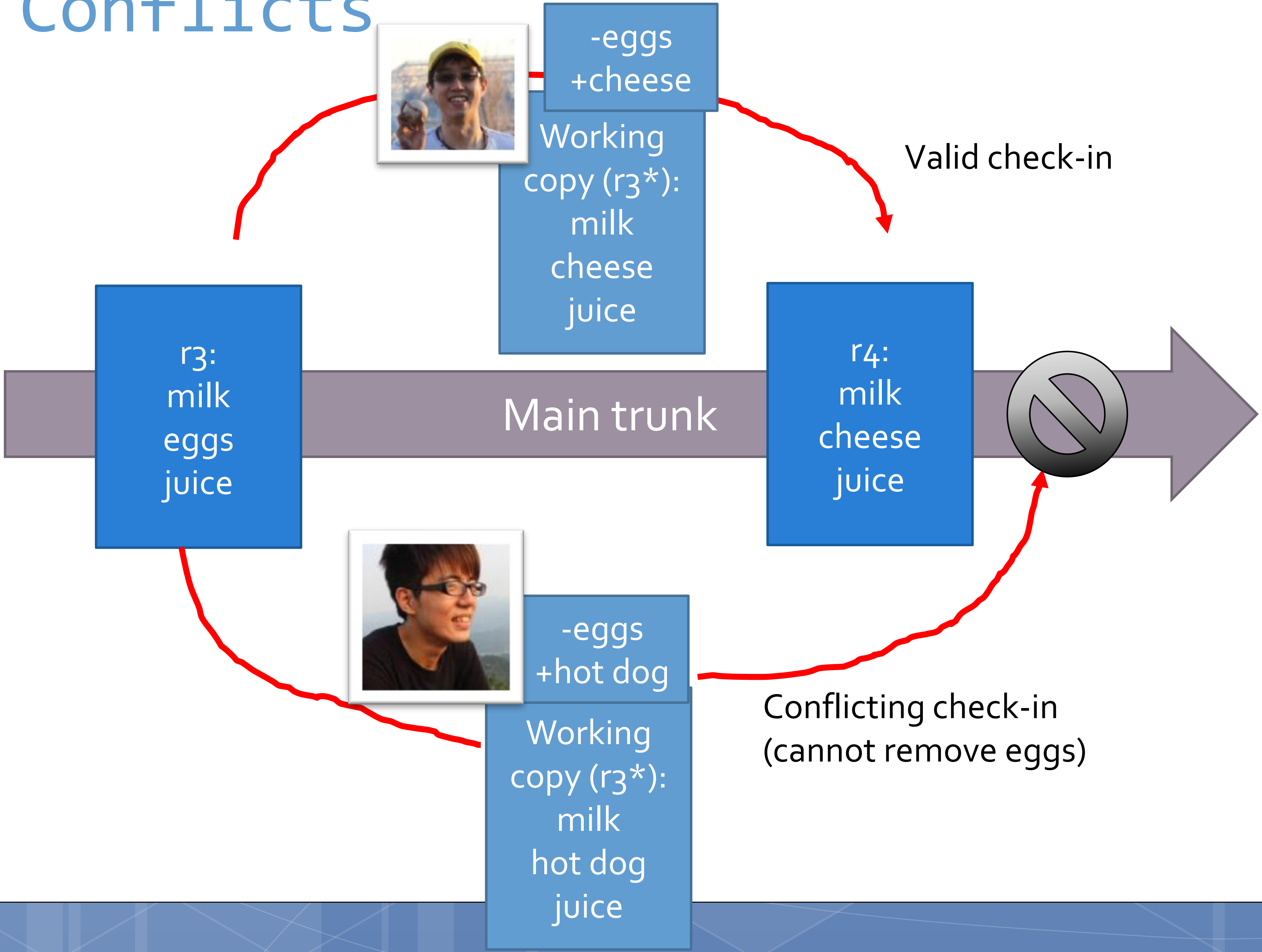
Branching


r5:
milk
eggs
soup
r6:
milk
eggs
soup
rice
New Features
Main trunk
r4:
milk
eggs
soup
r7:
milk
eggs
soup
bread

Merging


r5:
milk
eggs
soup
+rice
r6:
milk
eggs
soup
rice
New Features
Main trunk
r4:
milk
eggs
soup
+bread
r7:
milk
eggs
soup
bread
+rice
r8:
milk
eggs
soup
bread
rice

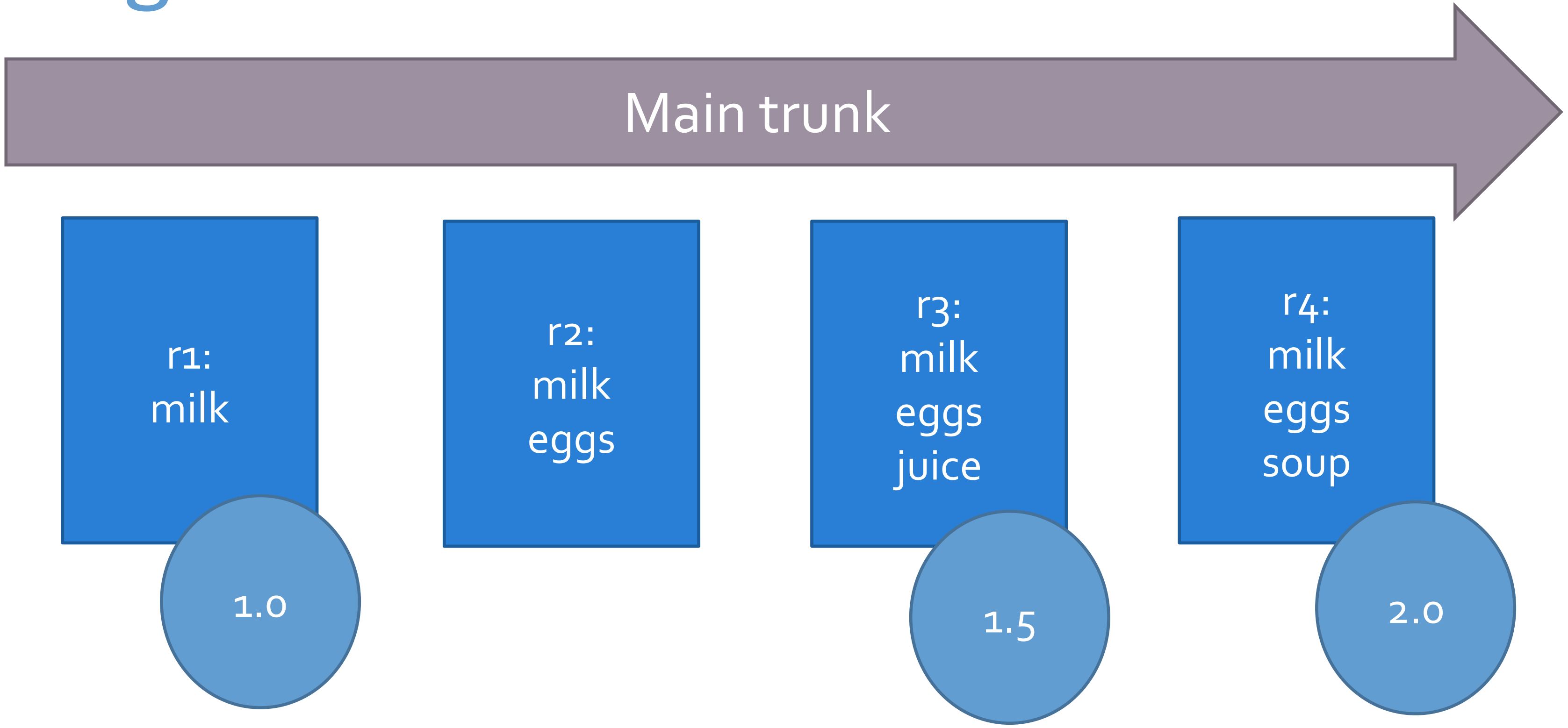
Conflicts


-eggs
+cheese
Working copy (r3*):
milk
cheese
juice
Valid check-in
r3:
milk
eggs
juice
Main trunk
r4:
milk
cheese
juice
-eggs
+hot dog
Working copy (r3*):
milk
hot dog
juice
Conflicting check-in
(cannot remove eggs)

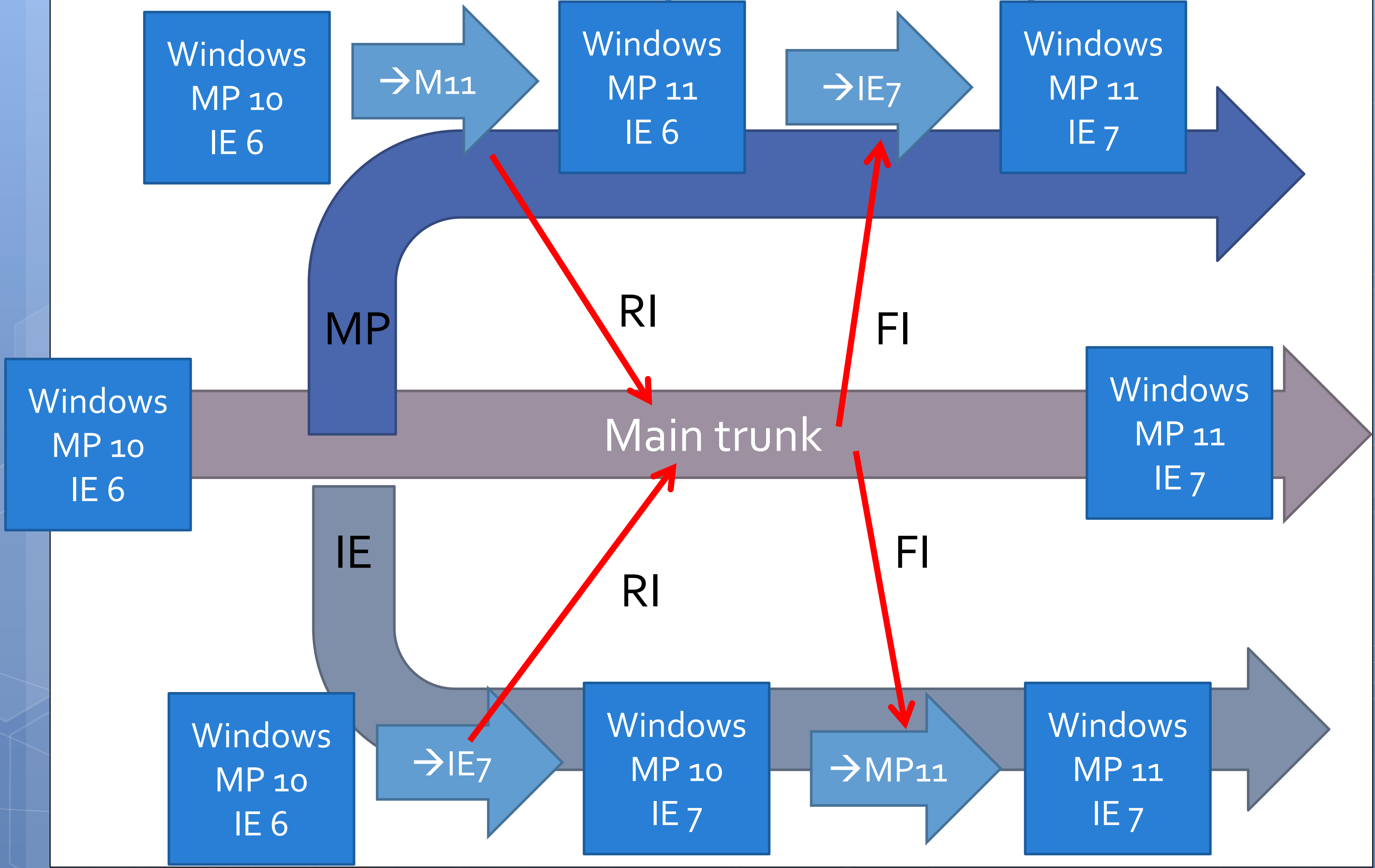
How to resolve the conflict?

- **Re-apply your changes**. Sync to the latest version (r4) and re-apply your changes to this file: Add hot dog to the list that already has cheese.
- **Override their changes with yours**. Check out the latest version (r4), copy over your version, and check your version in. In effect, this removes cheese and replaces it with hot dog.

Tag


Main trunk
r1:
milk
r2:
milk
eggs
r3:
milk
eggs
juice
r4:
milk
eggs
soup
1.0
1.5
2.0

Real world example - simplified



How do I get started?

- Get yourself more familiar with subversion
  (not the fastest/fanciest/most powerful on the market, but it's good enough for most projects)
- Windows GUI for subversion: TortoiseSVN
  http://tortoisesvn.tigris.org/
  - Very easy to use!
- Free book about subversion:
  http://svnbook.red-bean.com/
- Once you know how to use it, make it a habit (not hard at all).

Further reading

- Distributed Version Control: http://betterexplained.com/articles/intro-to-distributed-version-control-illustrated/
- (Probably will not cover it this semester)