Version Control

- Introduction / Taster
- Use on-line resources, books or colleagues for more info

Solution Perspective Media

solution
perspective
media

Version Control

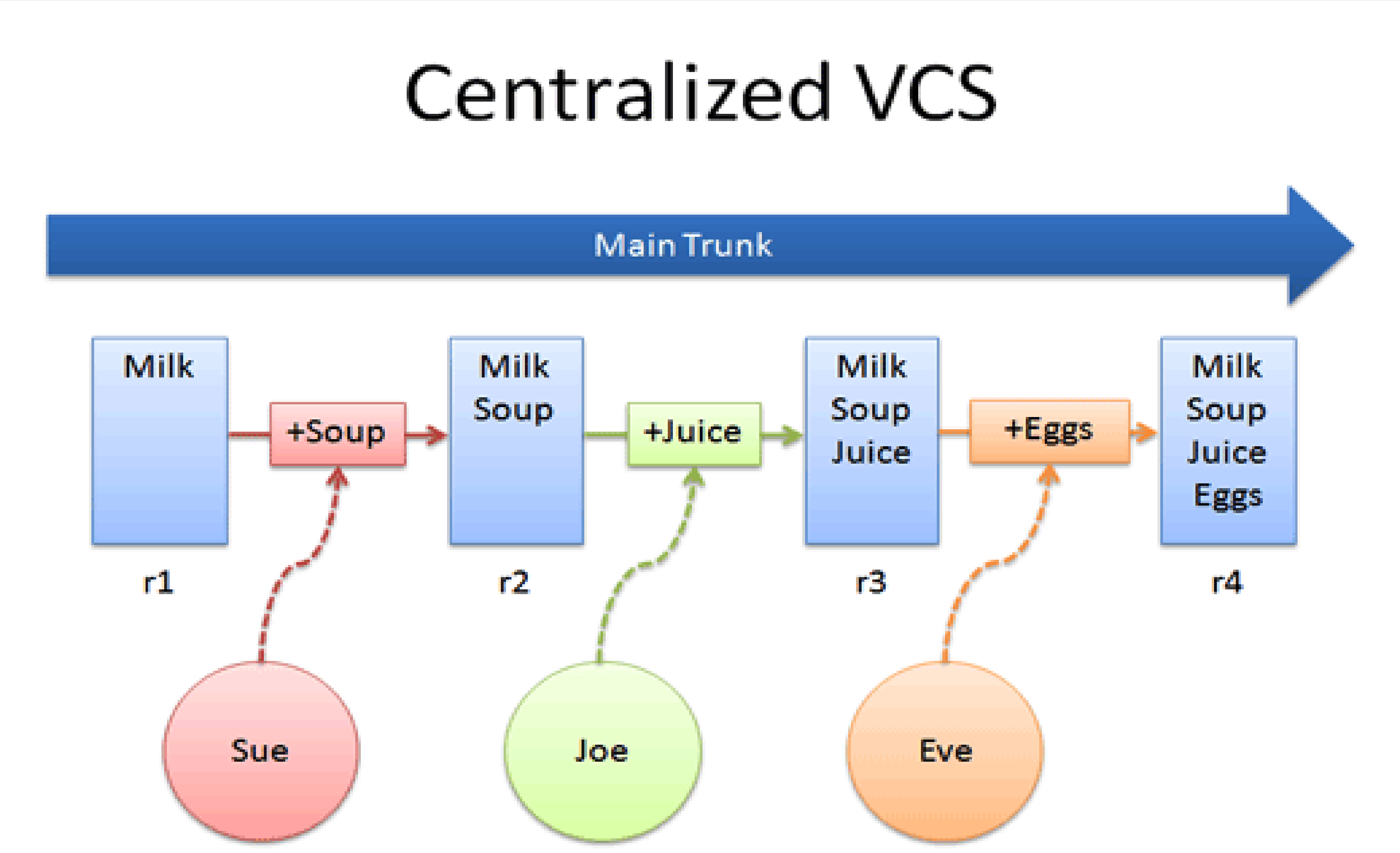
- What is it
  - A method for centrally storing files
  - Keeping a record of changes
  - Who did what, when in the system
  - Covering yourself when things inevitably go wrong

Version Control: Why?

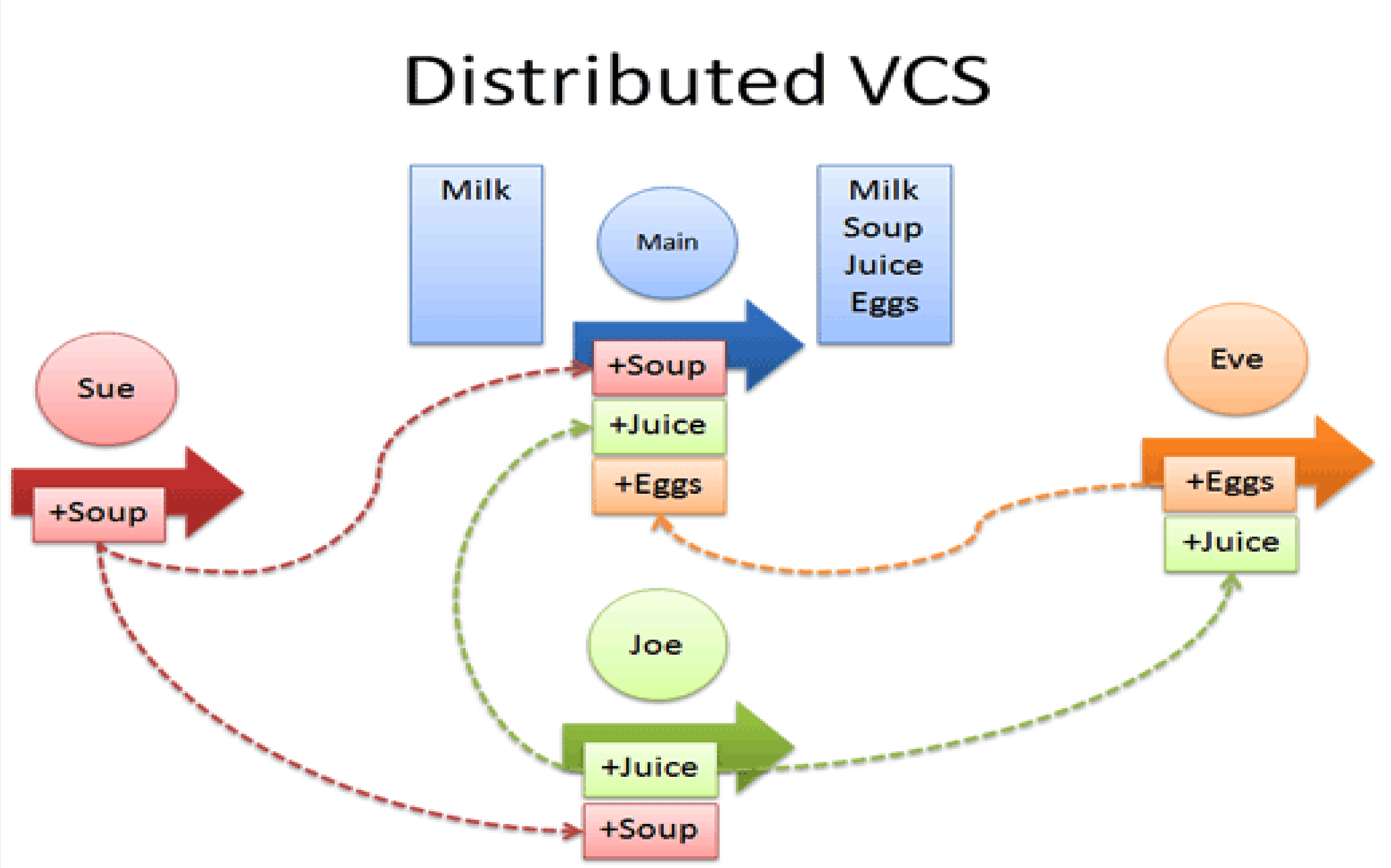
- Individual
  - Back-up methodology
  - Increments – know which version is live
  - Point in time marking aka. Tagging
  - Branching – release versions maintained & main development can continue
  - Change history – when features were added or amended

- Team
  - As Individual plus:
  - Allow multiple developers (in remote locations) to work on same code base
  - Merge changes across same files – handle collisions
  - Answer who did what – blame / praise

Version Control: Centralised



Version Control: Distributed



Version Control: Types

- CVS – Concurrent Version System
  - http://www.nongnu.org/cvs/
- SVN – Subversionhttp://subversion.tigris.org/
- Git* - http://git.or.cz/
- Bazaar* - http://bazaar-vcs.org/
- Mercurial* - http://www.selenic.com/mercurial/
- Monotone* - http://www.monotone.ca/
- VSS – Visual Source Safe – Microsoft visual tool
  - * Distributed version control

Subversion

- Code centralised in a repository
- Check out a working copy into a development area on local machine
- Make changes, test etc.
- Changes committed back to the central repository – usually with a useful comment
- Each work session an update is performed to get changes from other team members

Subversion

- Change log – who, when, what
- Check for differences between current version and any version in repository
- Create and apply patches between tags, branches and trunk
- Recover old versions of files, roll-back when it goes wrong
- Recover old versions of project – single version number for all files (unlike e.g. CVS)

Subversion Clients

- SVN command-line tool
- HTTP (WebDav)
- SVN+SSH
- Dedicated client tools for all major platforms
- Plug-ins for IDEs e.g. Zend Studio, Eclipse
- Automatic integration as a network drive (user doesn't need to know it's a version control system)

Subversion Resources

- Documentation, links to clients etc. at http://subversion.tigris.org/
- Read Bean Book (Open Source) http://svnbook.red-bean.com/
- Pragmatic Version Control Using Subversion – Mike Mason, Pragmatic Bookshelf
- Version Control with Subversion – Collins-Sussman, O'Reilly (Turtles)