Introduction to Subversion

- Source-code control: motivation
- Basic Concepts: Repository
- Basic Concepts: Working Copy
- Various operations
- Other commands (including locks)
- Branches and merges

What is Subversion?

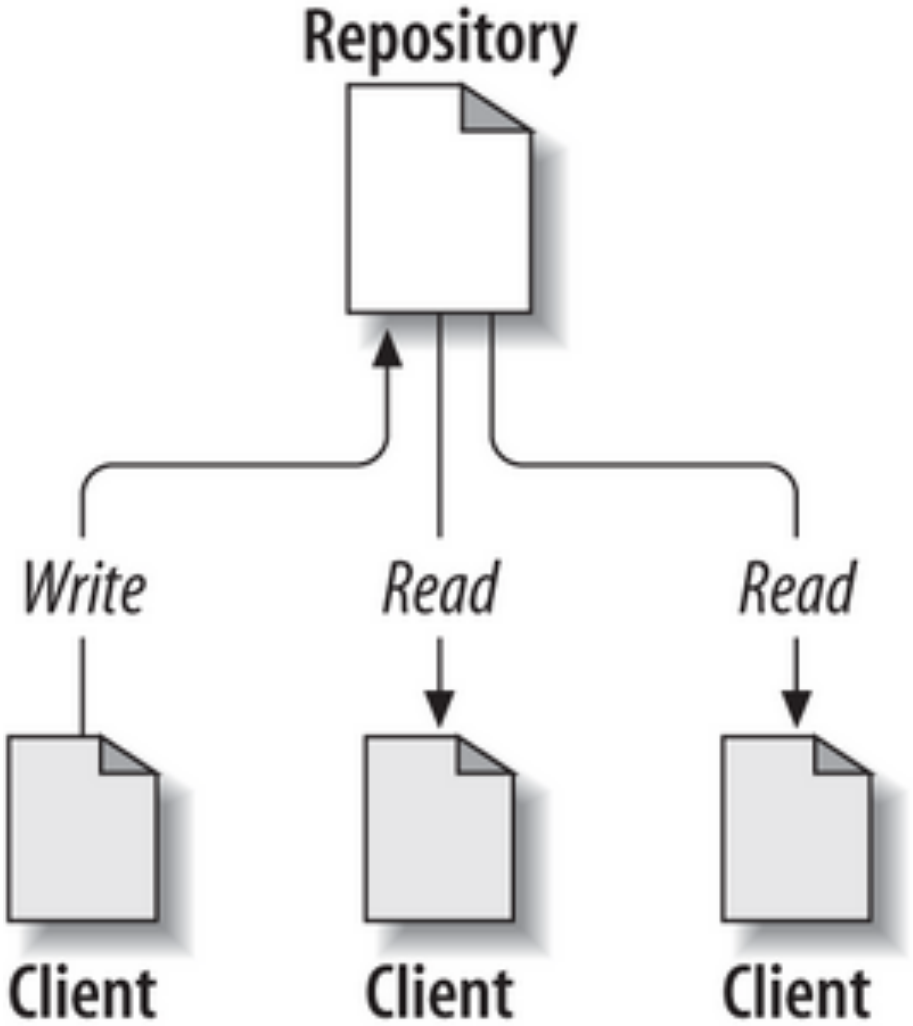
- Open-source version control system
- Helps manage files and directories as they change over time
  - These files/directories are usually associated with some software development project
  - However, svn is extensively used in the HTML world
  - (git is also used more and more now, but we'll use svn this term)
- Resembles some features of a file system yet:
  - remembers changes to all files managed by the repository
  - can recover older versions of softwrae systems (i.e., like a time machine)
- Also permits concurrent access of its repository over a network
  - This facilitates work on shared projects
  - Therefore also enhances collaboration (we hope!!)
  - Some programmers use svn as a one-stop data repository

Some history

- First version start in 2000 by CollabNet
  - Ongoing development is still funded, in part, by CollabNet
- Their goal:
  - Make a clean version of CVS…
  - … Thereby fixing shortcomings of CVS
- Source and precompiled binaries available for a large number of systems
- **http://subversion.apache.org**

Basic Concepts: Repository

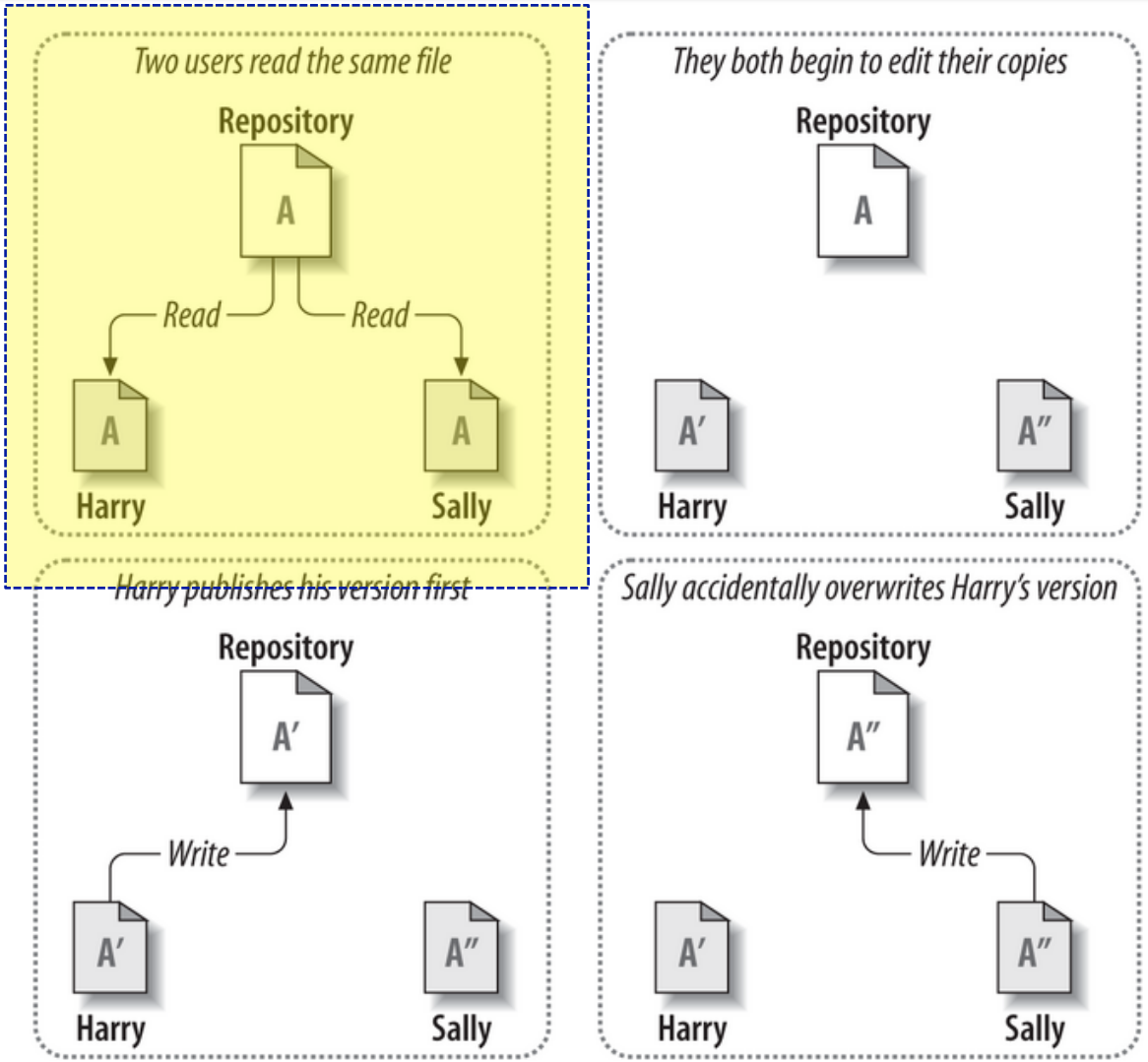
- A Subversion **repository** is a central store of data
- Repositories are usually stored on an svn **server**
- Data is stored in a manner that **resembles** a filesystem tree
- Any number of **clients** can connect to the repository
  - These clients can then read and write files in the repository
- By writing (e.g., committing) files, client makes files available to other clients
- By reading (e.g., updating) files, client is receiving information from other clients



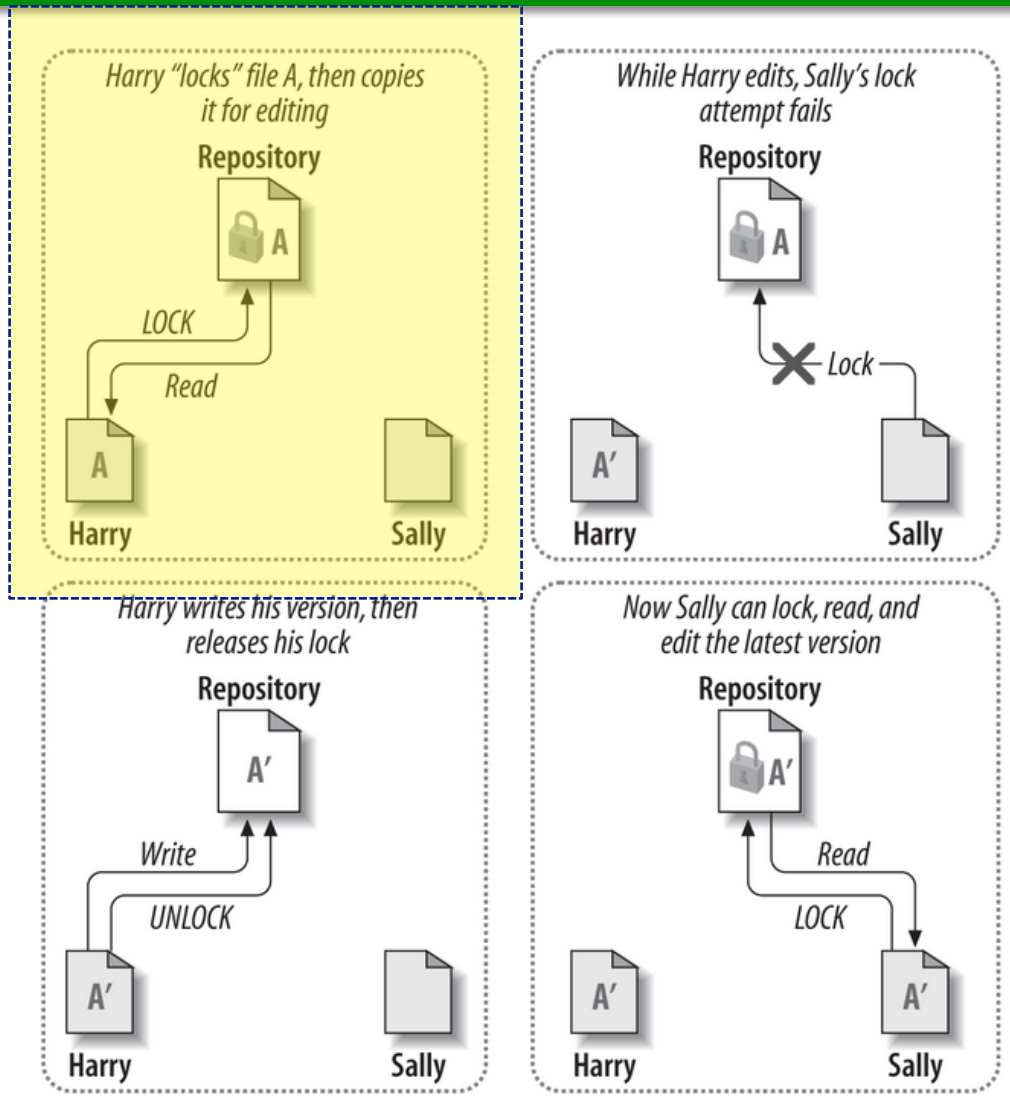
Basic Concepts: Repository

- A repository remembers every **committed** change to every **controlled** file
  - Even remembers additions and deletions to directory trees
- Clients reading from the repository normally sees latest version of file structure
  - Clients, however, can also choose to view previous states of the file structure
- Examples of previous states:
  - "What did this directory contain last Wednesday?"
  - "Who was the last person to change this file?"
  - "What changes did Pat make to this file?"
  - "Give me the version of my code that worked yesterday."

What we do not expect/want…



One approach around problem…

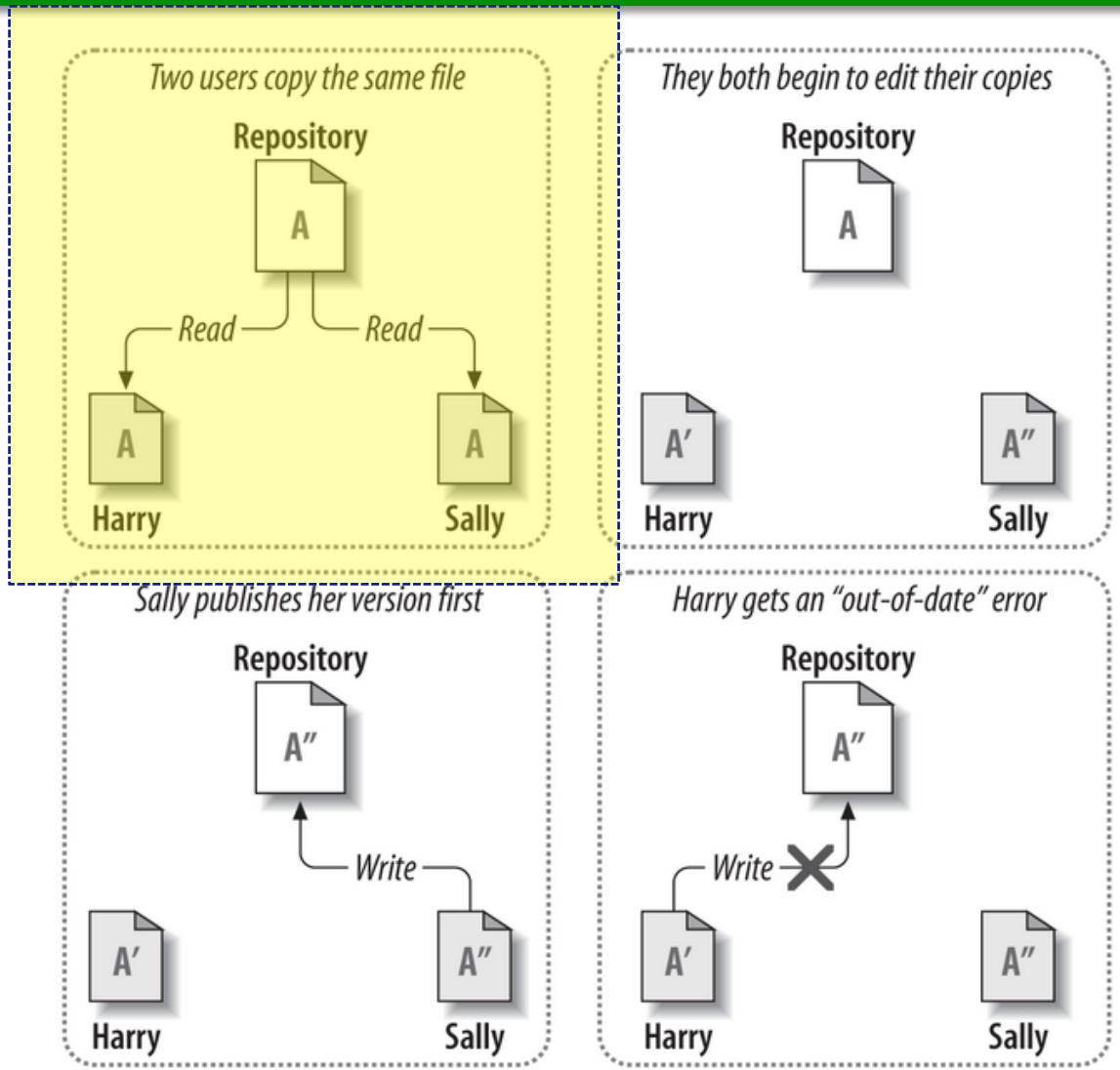


Some downsides to "locking"

- Can cause some administration problems
  - User with lock on holidays? sick?
  - Is it appropriate for all changes?
- Awkward in distributed environments
  - CVS and its lack of locking was attractive for early open-source projects
  - Such projects had developers located around the globe
  - Locked files would be very inconvenient (if not disastrous)
- Locking gives a false sense of security.

"Copy-modify-merge" solution

scenario: conflict is introduced



"Copy-modify-merge" solution

scenario: conflict is resolved