Collaborating Using Git

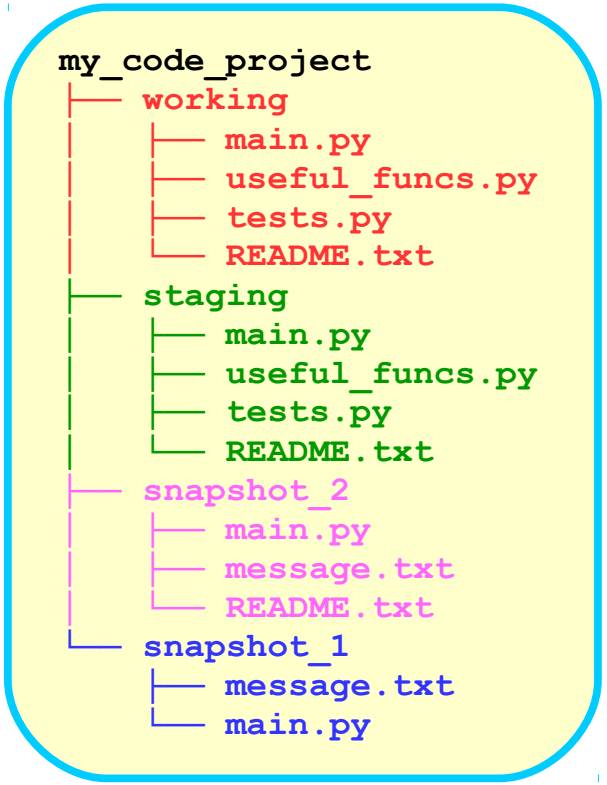
Mark Slater

mslater<at>cern.ch, Physics West 317

UNIVERSITY OF BIRMINGHAM

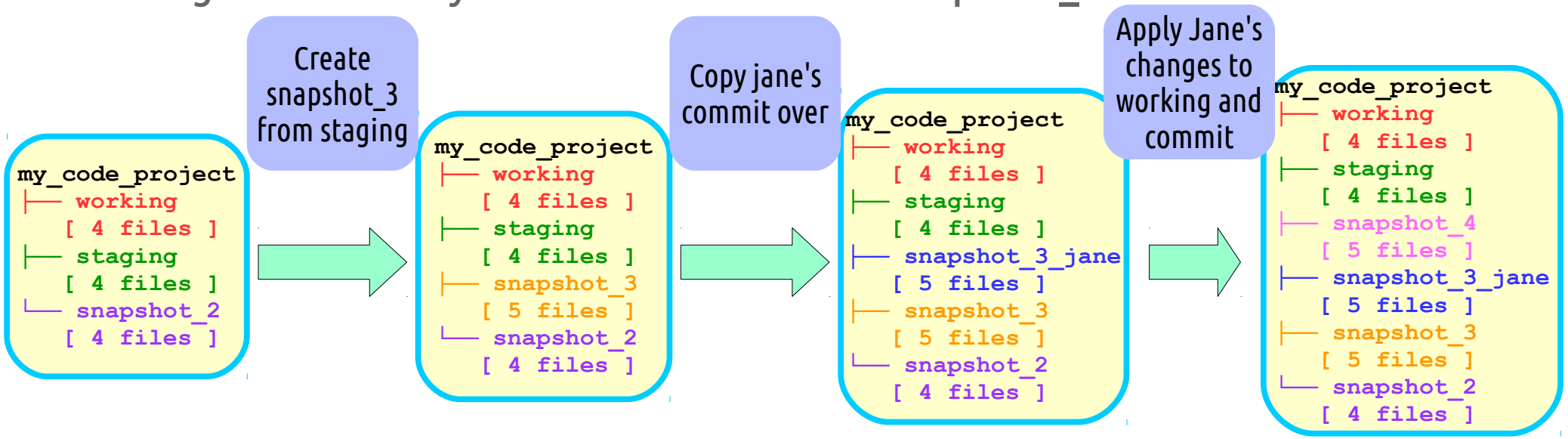
Collaborating

- Another of the powerful features of Git is how easy it is to work with others on material
- To give an idea of how it works, let's return to developing our own VCS:



Developing a VCS: Playing nicely with Others

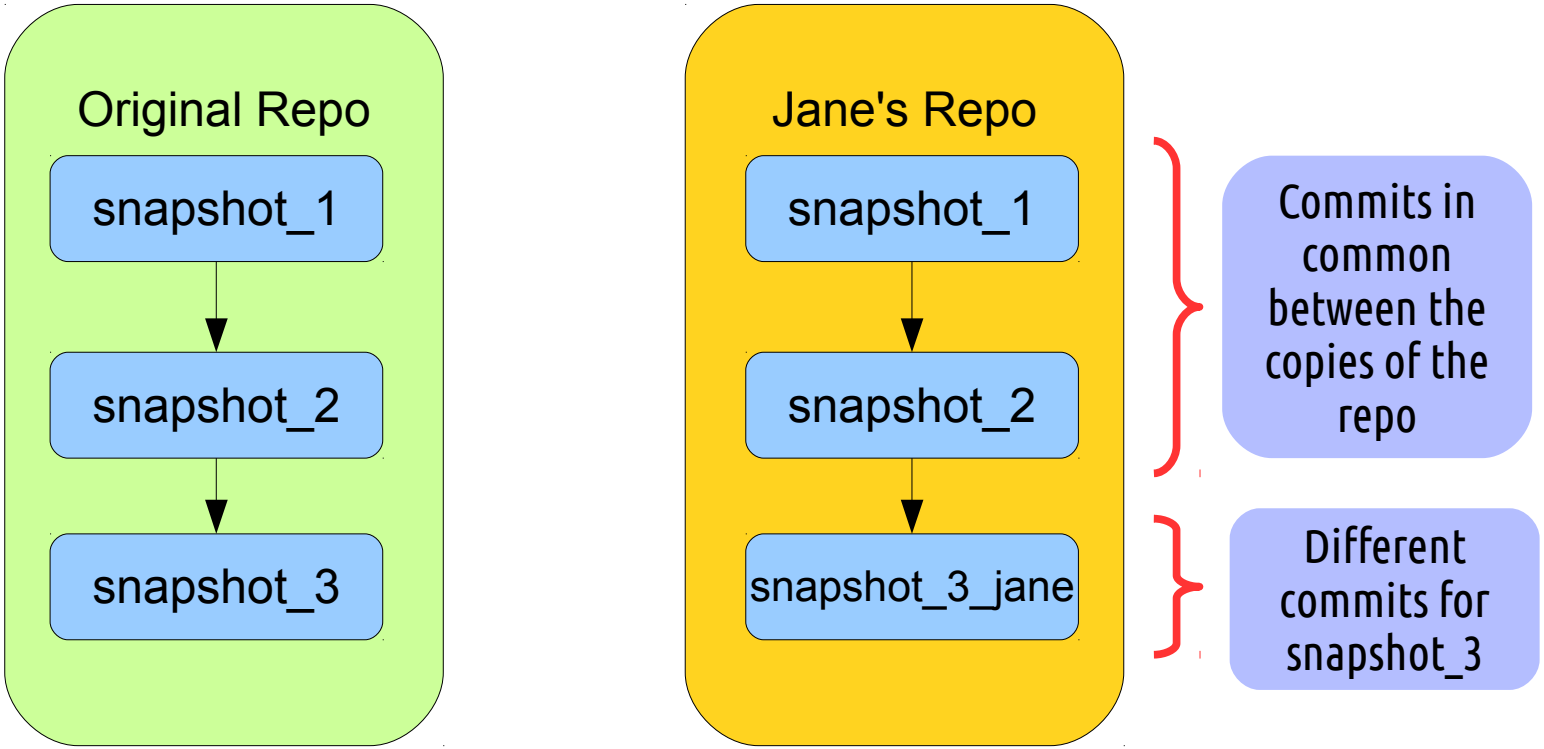
- Let's say you share your repository with someone ('Jane') and in parallel both develop a 'snapshot_3' commit – what happens?
- After committing your version, you copy Jane's commit directory and call it 'snapshot_3_jane'
- Then you can change your working version (i.e. 'snapshot_3'), apply Jane's changes and finally make the commit as 'snapshot_4'



- Because you are merging two sets of changes, this final commit is called a 'Merge Commit'

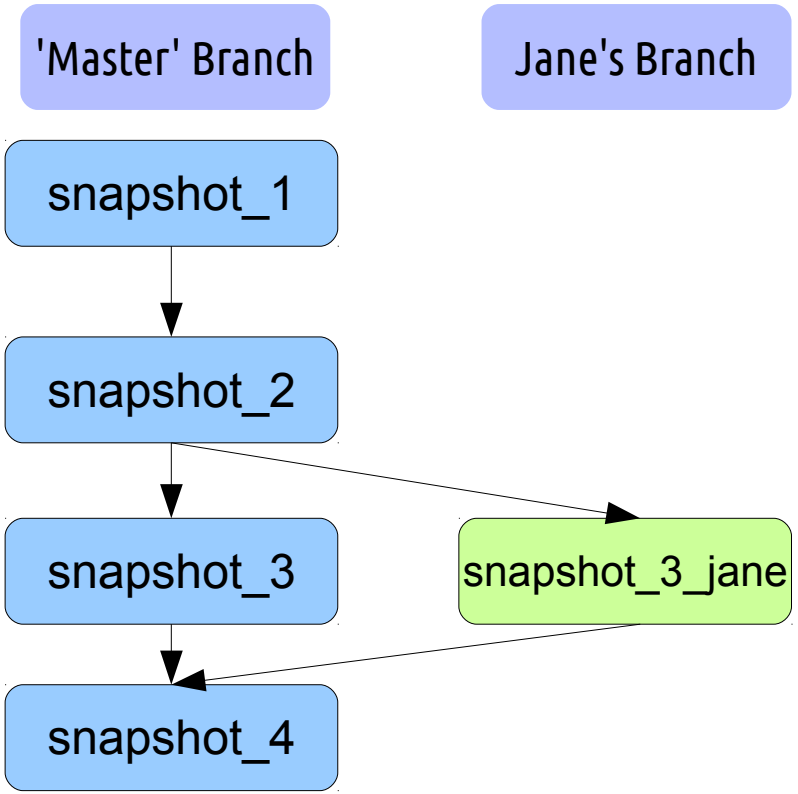
Remotes

- This has introduced the idea of 'Remote Repositories'
- As far as Git is concerned, neither your's or Jane's copy of the repository is special and each can have different sets of commits
- To link them, you can set up each as a 'Remote' of the other and then share/merge commits between them
- Quite often, Github or similar will hold the 'main' repository but again, this is an arbitrary designation



Branches

- This also demonstrates how branches work
- In git, branches are simply labels on particular commits that track any additional commits in that sequence
- This can be very useful to do parallel development on different features or between collaborators without directly affecting the 'master' version of the code
- Branches can then be merged together in to add their code to a particular branch, e.g. add a feature to the 'master'



Collaboration Tools

- We have now learnt about more of the common concepts in Git:
  - Repository – The folder with all the files associated with the project and git are located
  - Index – What git calls the 'staging area'
  - Commit – creating a copy of the index, adding a message and updating the hash pointers
  - Hash – Used to create unique filenames based on the file contents
  - HEAD – the hash that points to the last commit of the current branch you're working on, used to compare the index with when committing.
  - Branch – Refers to a particular development path, e.g. Jane's changes above
  - Remote – This is a remote copy of the repository that may have different commits to yours, e.g. Jane's copy of the directory