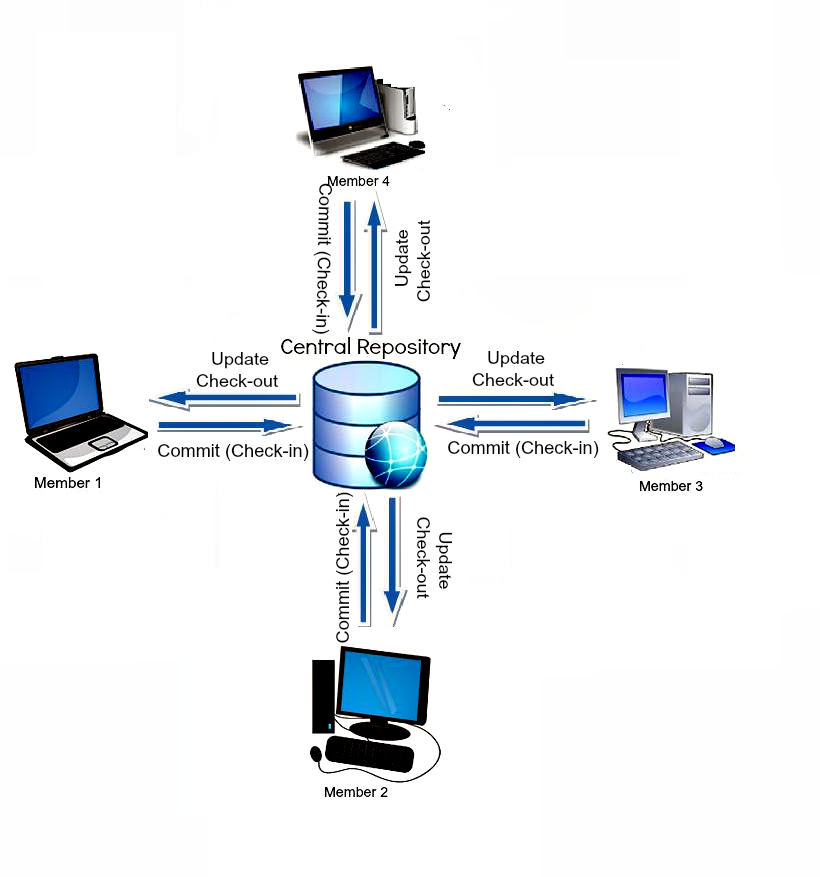
**Centralized vs Distributed Version Control**

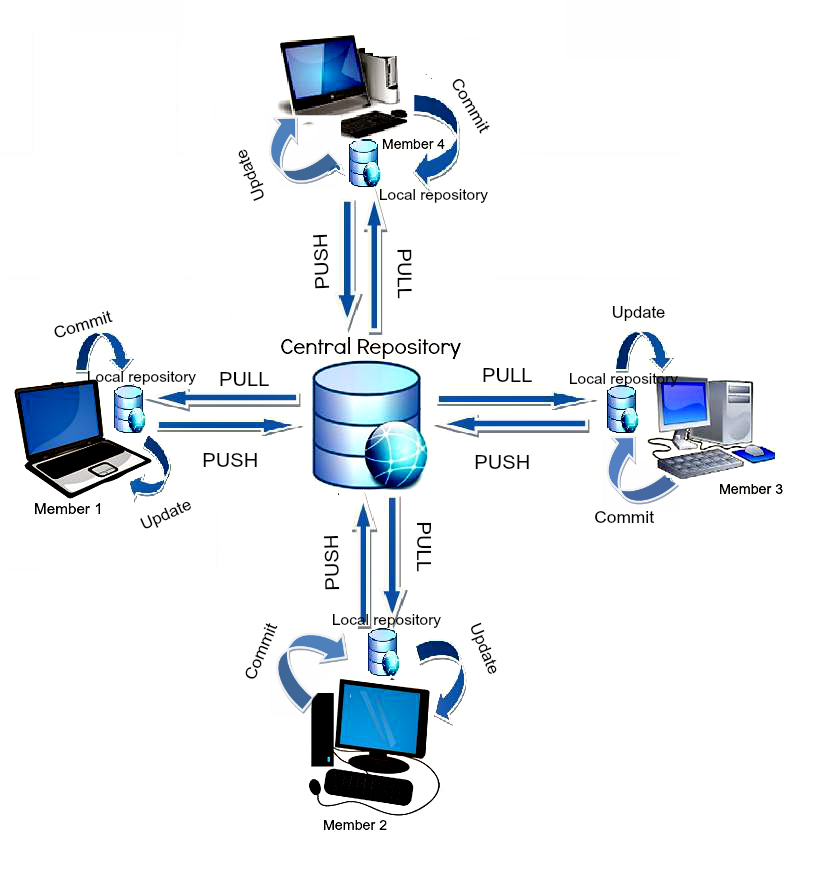
# [CVCS vs DVCS]

I am going to shed some light on the**differences between Centralized Version Control systems and Distributed Version Control Systems.**

The concept of a **centralized system** is that it works on a Client-Server relationship. The repository is located at one place and provides access to many clients.

[](https://scmquest.com/wp-content/uploads/2014/05/central_vcs.jpg)

Whereas, in a **Distributed System**, every user has a local copy of the repository in addition to the central repo on the server side.

[](https://scmquest.com/wp-content/uploads/2014/05/distributed_vcs.jpg)Centralized Version Control is the simplest system with the concept of 1 central repository which servers provides latest code to the all the clients across the globe

Distributed Version Control provides flexibility and has emerged with the concept that everyone has their own repository, they don’t just check out the snapshot of the code – they fully mirror the central repository.

CVCS is easy to understand whereas DVCS has some complex process for beginners.

CVCS is dependent on the access to the server whereas DVCS provides the benefits to work offline. Everything except push and pull the code can be done without an internet connection.

CVCS is easy to administrate and has more control over users and access as it is server from one place.

DVCS is comparatively fast comparing to CVCS as you don’t have to contact the central server for every command. DVCS just takes much time on the first check-out as its mirroring the central repository on your local.

If your project has a very long history and change-sets then downloading the entire history can take an unreasonable amount of time and disk space in DVCS whereas CVCS allows you to checkout only few lines of code if you just need to work on few modules.

DVCS provides a powerful and detailed change tracking, which means fewer conflicts at the time of merge.

DVCS gives an ability that developers can share changes with one or two other members of team at a time if they want to get some feedback before showing the changes to everyone.

The revisions in DVCS are typical big guids (like fa333b7rer96cd6d3b0037d660) – it’s not incremental numbers (which is provided by CVCS) which make them harder to reference and remember.

DVCS provides an advantage wherein if the main server’s repository crashes, you still have a local repository in every developer’s local space from which you can create the main repository.

SVN and CVS are the popular tools of CVCS

GIT and Mercurial are the popular tools of DVCS

I hope now you have a good idea of differences between the different version control systems – Centralized and Distributed.