There are three primary types of Version Control Systems (VCS):

**Local Version Control Systems (LVCS):**

These systems track changes to files directly on a single local machine.

Each version of a file is typically stored as a patch, which represents the changes made since the previous version.

An example is RCS (Revision Control System).

LVCS are generally not suitable for collaborative environments as they lack features for multiple users to work on the same codebase simultaneously.

**Centralized Version Control Systems (CVCS):**

These systems utilize a single central server that stores all file versions and the complete history of changes.

Users check out files from this central repository, make modifications, and then check them back in.

Examples include SVN (Subversion) and CVS.

CVCS enable collaboration among developers but have a single point of failure: if the central server becomes unavailable, no one can commit changes or access the repository.

**Distributed Version Control Systems (DVCS):**

In a DVCS, each user maintains a complete copy of the entire repository, including its full history, on their local machine.

Users can commit changes locally and work offline, then synchronize their changes with remote repositories as needed.

Examples include Git and Mercurial.

DVCS offer greater flexibility, resilience (no single point of failure), and often better performance for distributed teams due to local operations and efficient merging capabilities.