# 1. Git

1.1. What is a version control system?

A version control system (VCS) allows you to track the history of a collection of files. It supports creating different versions of this collection. Each version captures a snapshot of the files at a certain point in time and the VCS allows you to switch between these versions. These versions are stored in a specific place, typically called a *repository*.

You may, for example, revert the collection of files to a state from 2 days ago. Or you may switch between versions of your files for experimental features.

The process of creating different versions (snapshots) in the repository is depicted in the following graphic. Please note that this picture fits primarily to Git. Other version control systems like *Concurrent Versions System* (CVS) don't create snapshots but store file deltas.
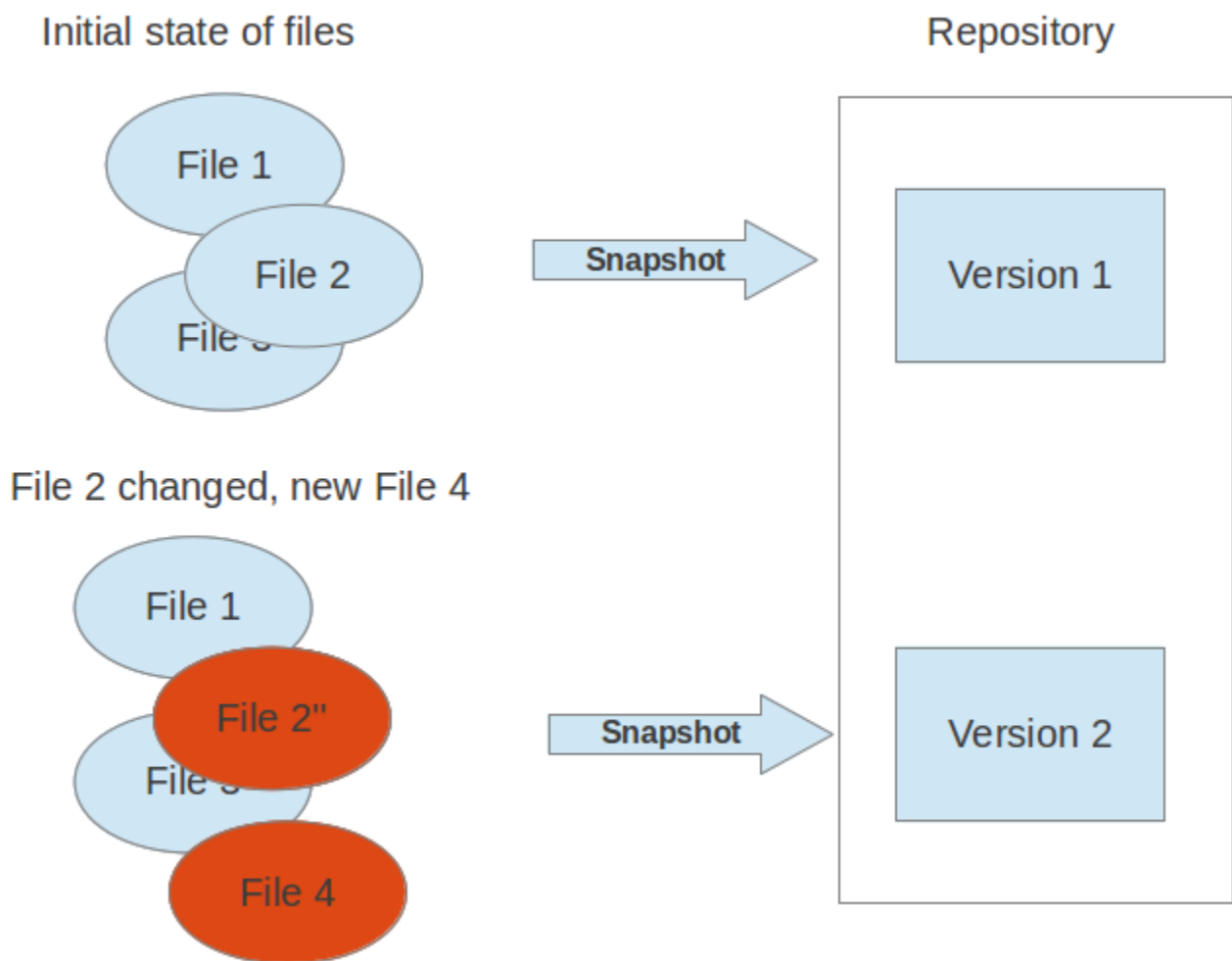
VCS are typically used to track changes in *source code* for a programming language or other text files, like HTML code or configuration files. But a typical version control system can put any type of file under version control, e.g., you may use a VCS to track the different versions of your company logo.

1.2. Localized and centralized version control systems

A localized version control system keeps local copies of the files. This approach can be as simple as creating a manual copy of the relevant files. A centralized version control system provides a server software component which stores and manages the different versions of the files and let developer copy (checkout) a certain version onto their individual computer.

Both approaches have the drawback that they have only one single point of failure, e.g., in localized version control systems the individual computer and in a centralized version control systems the server machine. Both system makes it also harder to work in parallel on different features.

1.3. What is a distributed version control system?

In a distributed version control system each user has a complete local copy of a repository on his individual computer. The user can copy an existing repository. This copying process is typically called *cloning* and the resulting repository can be referred to as a *clone*.

Every clone contains the full history of the collection of files and a cloned repository has the same functionality as the original repository.

Every repository can exchange versions of the files with other repositories by transporting these changes. This is typically done via a repository running on a server which is, unlike the local machine of a developer, always online. Typically there is a central server for keeping a repository but each cloned repository is a full copy of this repository. The decision which of the copies is considered to be the central server repository is pure convention and not tied to the capabilities of the distributed version control system itself.
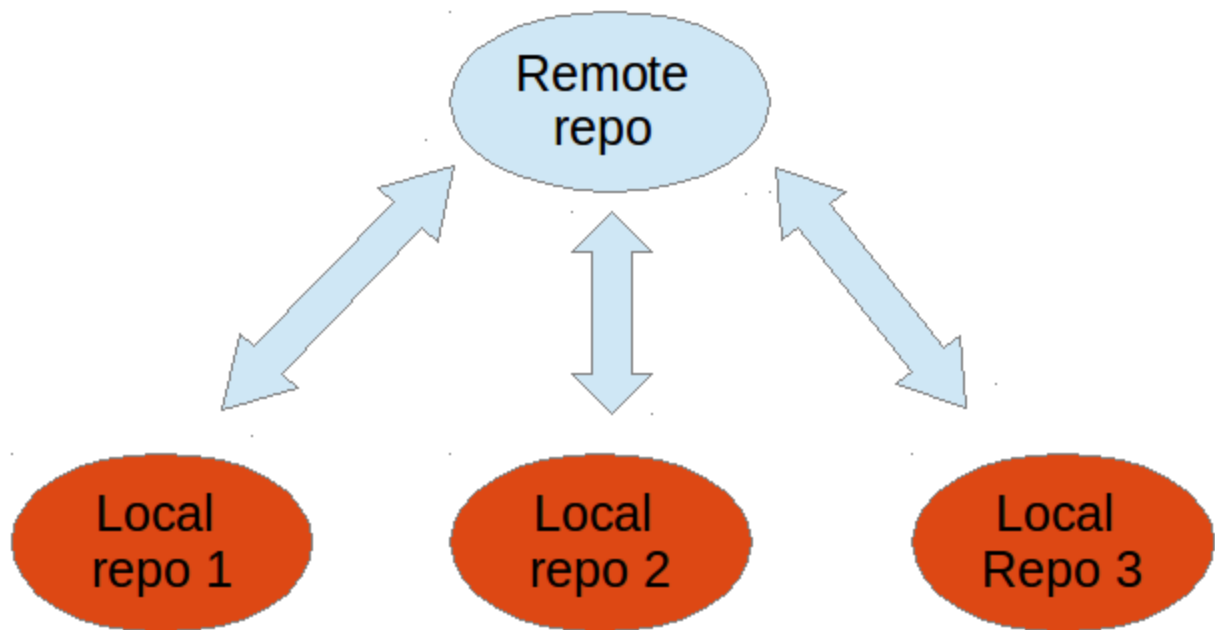
1.4. What is Git?

Git is currently the most popular implementation of a distributed version control system.

Git originates from the Linux kernel development and was founded in 2005 by Linus Torvalds. Nowadays it is used by many popular open source projects, e.g., the Android or the Eclipse developer teams, as well as many commercial organizations.

The core of Git was originally written in the programming language *C*, but Git has also been re-implemented in other languages, e.g., Java, Ruby and Python.