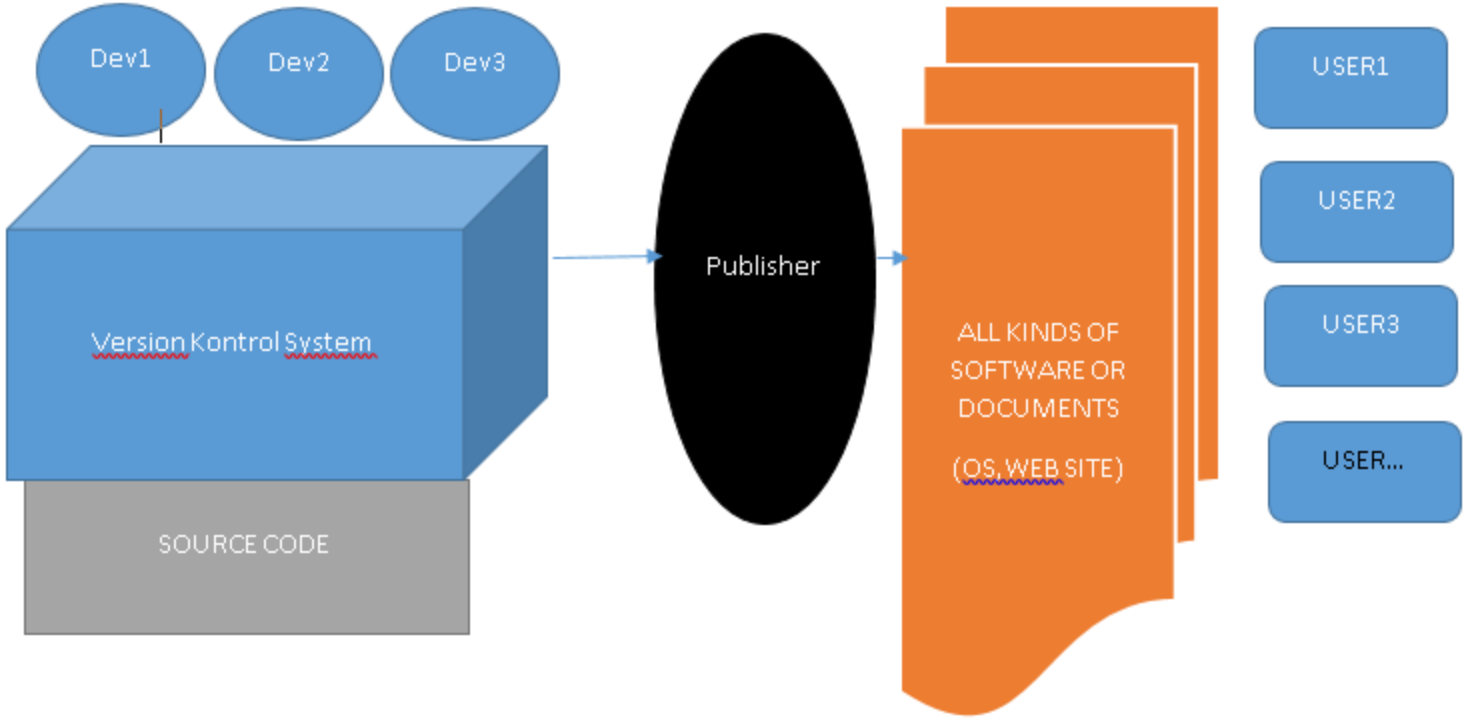
Version Control Protocols: GIT vs. SVN

Version Control Protocols

- What Is Version Control and Why Is it Important? Version control is important **to keep track of changes — and keep every team member working on the right version**.



Definitions

- **Repository,** a data structure which stores metadata for a set of files or directory structure
- Software repository, a storage location for software packages
- Contrubutor
- **Get,Clone,Pull**
- **Checkout,Undo Checkout**
- **Check-in, Commit,Stage, Push**

Version Control Protocols

In software engineering, **version control**

also known as

- **revision control**
- **source control**
- **source code management**

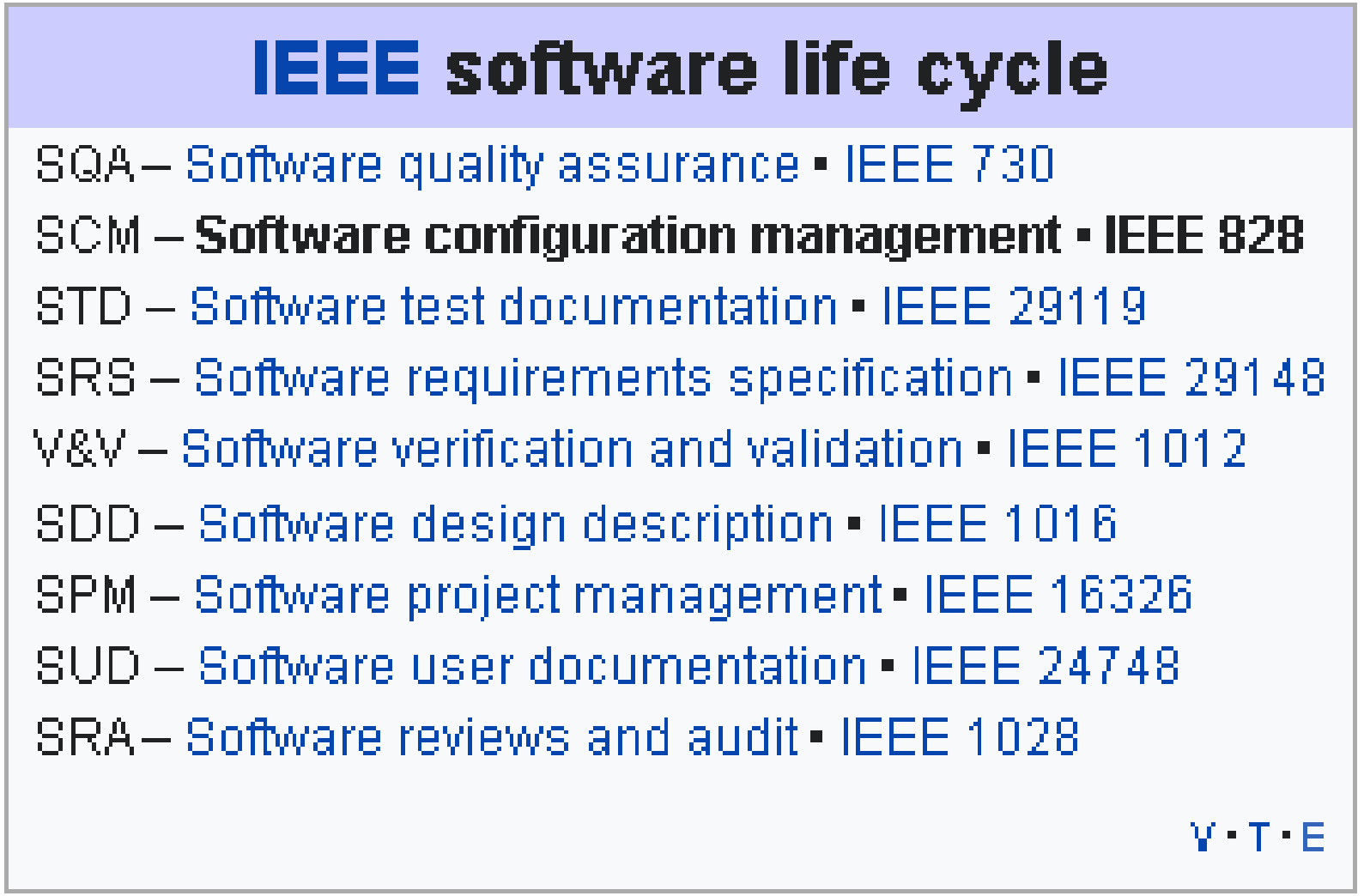
computer programs,
documents,
large web sites,
or other collections of information

Version control is a component of software configuration management.
SCM can determine the "what, when, why and who" of the change

Position of Protocol

IEEE Software life cycle→SCM→Version Control

SCM



Version Control

- In computer software engineering,

➢ practice that tracks and provides control over changes to source code, document, configuretion files…

➢ Contrubutor designer, developer,tester and deployer software

➢ Multiple copies of the different versions of the program,

Version Control Software Model

- Local data model

In the local-only approach, all developers must use the same file system.(Revision Control System (RCS), Source Code Control System (SCCS) – part of UNIX)

- Client-server model

In the client-server model, developers use a shared single repository.

IBM Rational ClearCase, Visual SourceSafe –by Microsoft; oriented toward small teams

Team Foundation Version Control - by Microsoft for Team Foundation Server, now Azure DevOps Server

- Concurrent Versions System (CVS) – originally built on RCS, licensed under the GPL.
- Subversion (SVN) – versioning control system inspired by CVS
- **TortoiseSVN** Windows client for the *Apache™ Subversion®* version control system

Version Control Software Model

➢Distributed model

In the distributed approach, each developer works directly with their own local repository,

- BitKeeper – was used in Linux kernel development (2002 – April 2005)
- Git –, designed by Linus Torvalds based on the needs of the Linux kernel project; decentralized,

aims to be

✓fast,

✓flexible

✓robust

What is Distributed?

In software development,

Distributed version control (also known as **distributed revision control**)

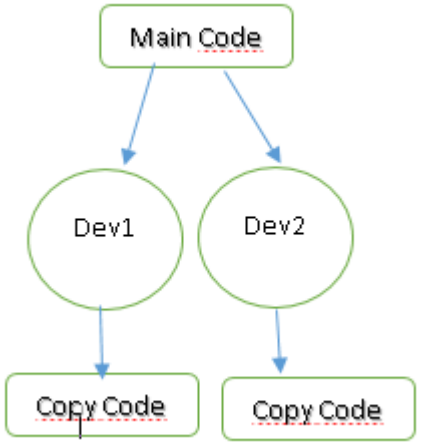
Compared to centralized version control, this enables

- ✓Its full history, is mirrored on every developer's computer.
- ✓Automatic management branching and merging, speeds up most operations
- ✓Ability to work offline, and does not rely on a single location for backups.
- ✓Allows private work
- ✓Avoids relying on one physical machine as a single point of failure

Distributed vs. centralized

DisAdvantages of DVCS

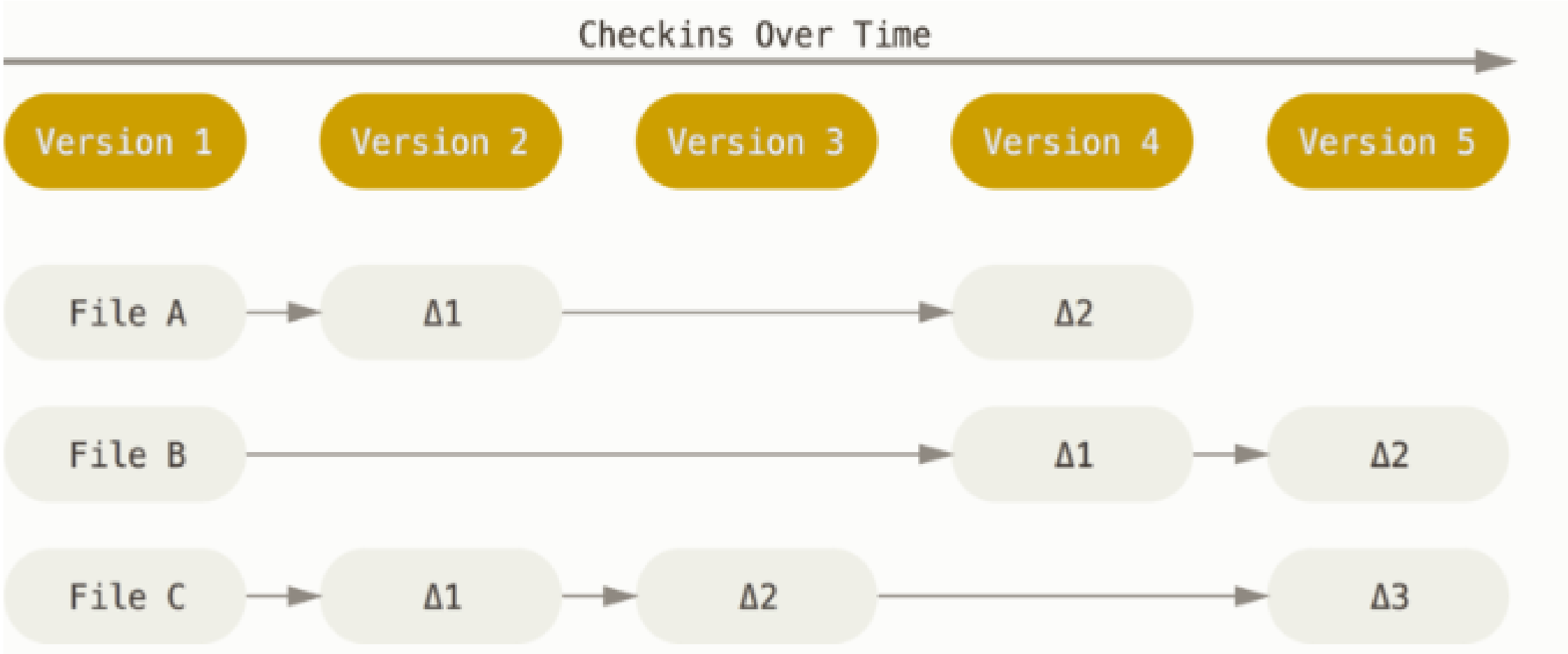
- Initial checkout of a repository is slower as compared to checkout in a centralized version control system, because all branches and revision history are copied to the local machine by default.
- Additional storage required for every user to have a complete copy of the complete codebase history.



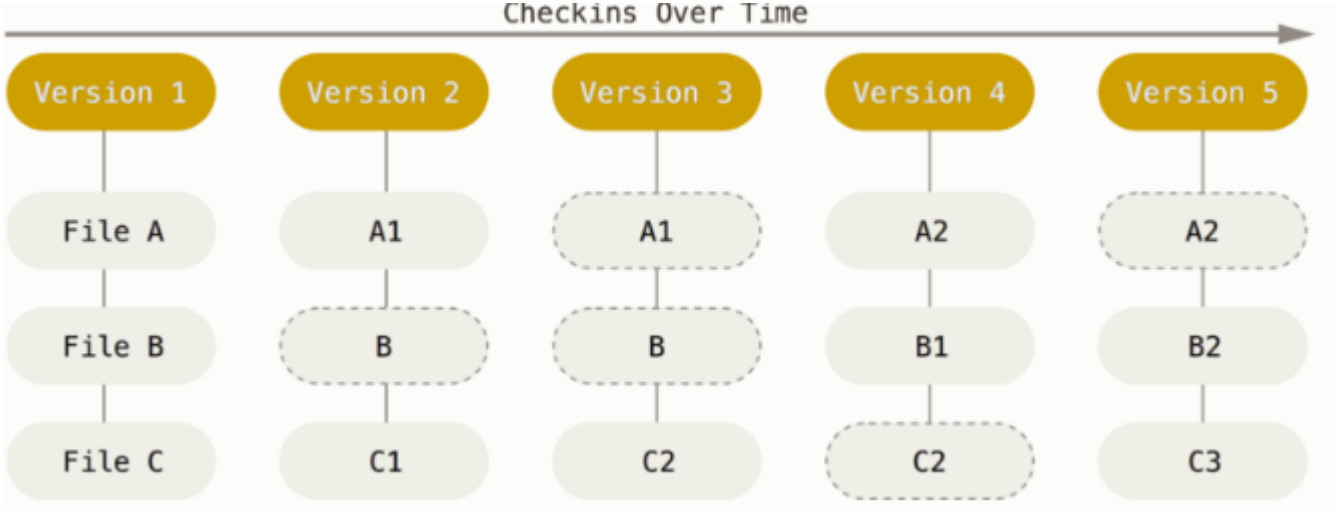
What is Git?

- Git is a free and open source

The major difference between Git andAny other VCS use **delta-based** method



Git thinks about its data more like a **stream of snapshots**.



This makes Git more like a mini filesystem with some incredibly powerful tools built on top of it, rather than simply a VCS.

What is Git?

- **Nearly Every Operation Is Local**

For example, to browse the history of the project, Git doesn't need to go out to the server to get the history and display it for you

- **Git Has Integrity (**SHA-1 hash**)**

Everything in Git is checksummed before it is stored and is then referred to by that checksum. This means it's impossible to change the contents of any file or directory without Git knowing about it.

- **Git Generally Only Adds Data**

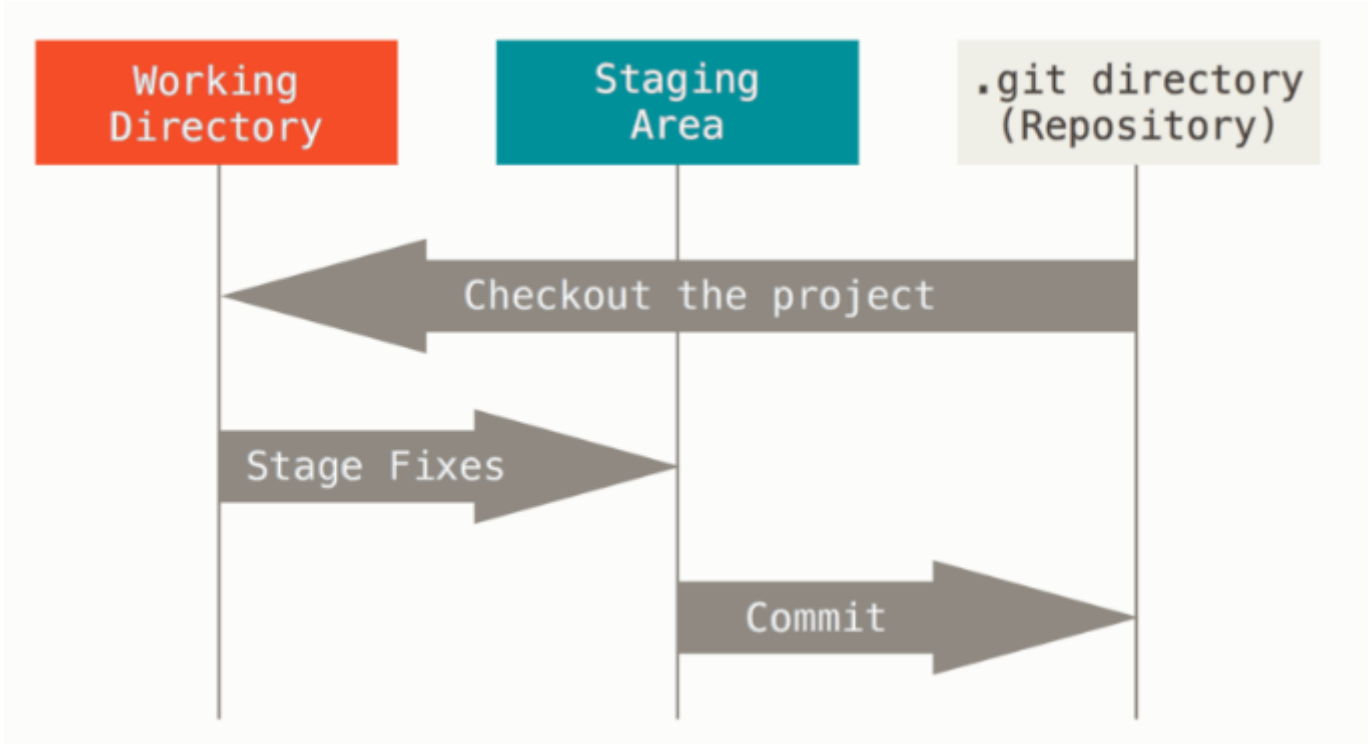
You can lose or mess up changes you haven't committed yet, but after you commit a snapshot into Git, it is very difficult to lose, especially if you regularly push your database to another repository.

Operations of Contrubutor

Centralized (SVN)	Distributed (Git)
GET LATEST VERSIYON	FORK, CLONE OR COPY
GET	PULL
MERGE	MERGE
CHECK-IN = COMMİT	STAGE-COMMIT-PUSH
CHECK-OUT	PULL REQUEST –MERGE REQUEST (FORK)

Git -The Three States

- **modified**, **staged**, and **committed**:
- Modified means that you have changed the file but have not committed it to your database yet.
- Staged means that you have marked a modified file in its current version to go into your next commit snapshot.
- Committed means that the data is safely stored in your local database.



Git Protocols to transfer data

Git uses Local, HTTP, Secure Shell (SSH) and Git

- **LOCAL**→remote repository is in another directory on the same host

The most basic is the **Local protocol**, in which the remote repository is in another directory on the same host.

$ git clone /srv/git/project.git

File-based repositories are that they're simple and they use existing file permissions and network access. If you already have a shared filesystem to which your whole team has access, setting up a repository is very easy.

Git Protocols to transfer data

- **HTTP**→ Dumb HTTP Smart HTTP ,

1.Dumb HTTP

The Dumb protocol expects the bare Git repository to be served like normal files from the web server

2.Smart HTTP

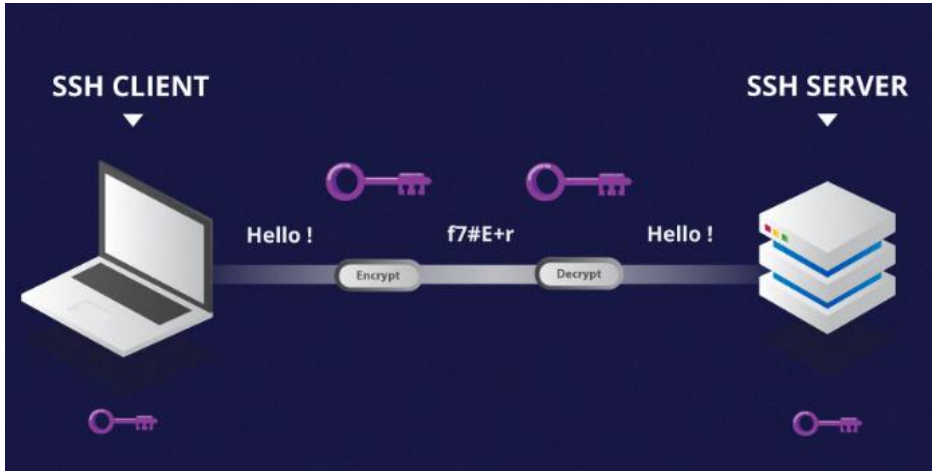
Use standard HTTPS ports and can use various HTTP authentication mechanisms

- **SSH**

A common transport protocol for

Git when self-hosting is over SSH.

$ git clone ssh://[user@]server/project.git



Git Protocols to transfer data

-Finally, we have the Git protocol it listens on a dedicated port (9418) service similar to the SSH protocol, but with absolutely no authentication.

-*The Git protocol is*

- *fastest network transfer protocol available.*
- *same data-transfer mechanism as the SSH protocol but without the encryption and authentication overhead.*

-*The downside of the Git*

- *protocol is the lack of authentication*
- *difficult protocol to set up. It also requires firewall access to port 9418,*

What is SVN

- SVN stands for **Subversion**. It is a **centralized version control system**. It is an **open-source** tool for version control.

- SVN acts as the time machine for the developers and allows them to go back and browse the history of the project.

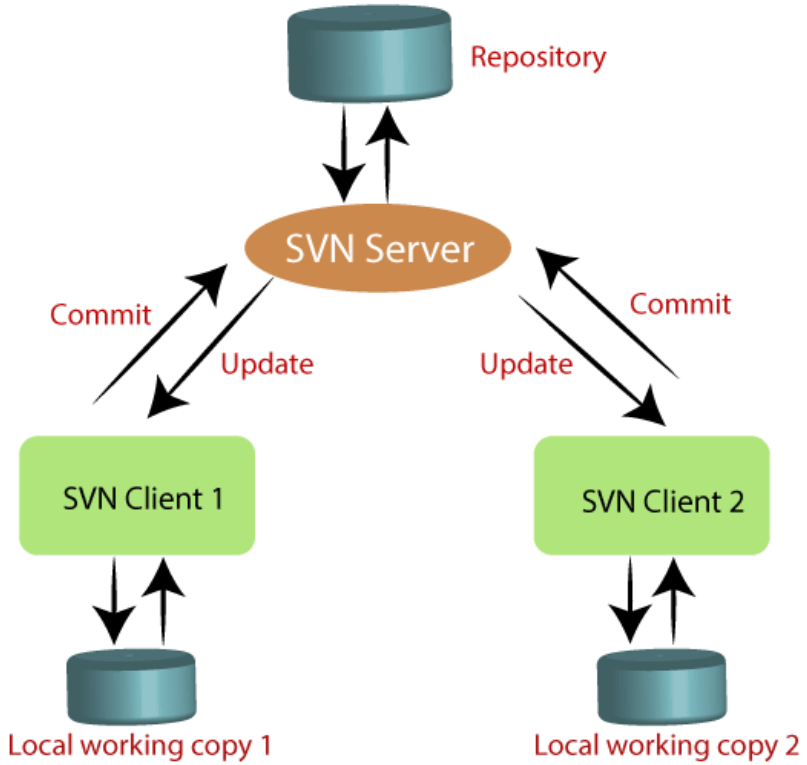
- There are a large number of projects that are still running on the Subversion.

- **Subversion** is **open-source** and comes under the **Apache License**,

SVN

The features of SVN are as follows:

- It supports atomic commits
- Less storage area
- It keeps a full revision history on server.
- It provides file locking for the files that cannot be merged.



Svn Protocols to transfer data

- http://repos
- https://repos

preferably https because of the encryption-and-authentication layer

- svn://repos

faster than HTTP(S)

- svn+ssh://repos

svn+ssh is the svn protocol run inside a SSH tunnel.

Teşekkürler