



**OST**

Eastern Switzerland  
University of Applied Sciences

# Distributed Systems & Blockchain (DS1)

## Distributed Version Control - git

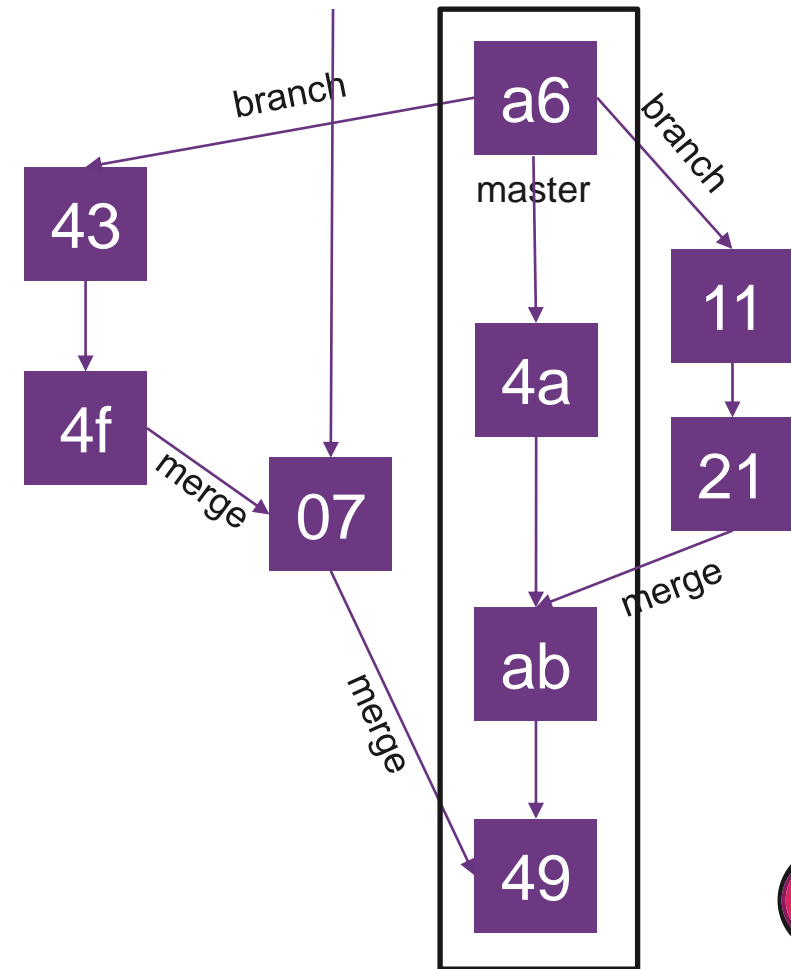
Thomas Bocek

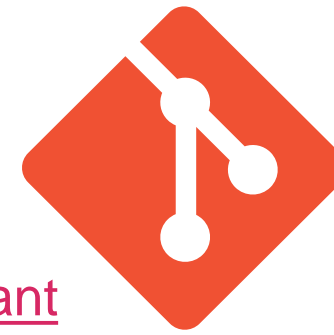
11 April 2021

# Distributed Version Control

- What is a version control system
  - Management of changes to documents/source code
  - Revisions
    - Compared, Restored, Source code: merged
- Why version control
  - Control over changes
  - Teams work on source code, features, bugs, different deployment
- Why distributed version control system
  - Simplest method: locks
  - Distributed revision control - complete codebase, including its full history, is mirrored on every developer's computer

- No single master copy, working copies
- P2P, every peer has full repo. Commit to your local repo → push/pull to interact with others





# git

## Distributed Version Control

- [Git](#) – distributed version control system for source code
- Created in 2005 by Linus Torvalds, creator of Linux
  - Before 2005 Linux used BitKeeper, a proprietary VCS
  - Some hackers started to reverse engineer the protocol of BitKeeper → they withdrew the free license. → [git](#), [mercurial](#)
  - Good [article](#) about history of git
- Distributed development
  - Local copy of the full development history

- [Git is dominant](#)
  - Git 87.2%
  - Subversion 16.1%
  - MS Team Foundation Server 10.9%
  - Zip files: 7.9%
- Merge: [3-way merge](#)
  - 2-way merge with diff (apply diffs sequentially)

Source File	Base File	Destination File
1 2 c class HttpServerDemo { 3 lic static void main(String[] args) thr 4 netSocketAddress addr = new InetSocketAddress 5 ttpServer server = HttpServer.create(ad 6 7 f( args.length == 0 ) 8	1 2 c class HttpServerDemo { 3 lic static void main(String[] args) t 4 netSocketAddress addr = new InetSocke 5 ttpServer server = HttpServer.create( 6 7 f( args.length == 0 ) 8 9 System.out.println("aborting..."); 10 return; 11 12	1 2 c class HttpServerDemo { 3 lic static void main(String[] args) thr 4 netSocketAddress addr = new InetSocketAddress 5 ttpServer server = HttpServer.create(ad 6 7 f( args.length == 0 ) 8 9 System.out.println("the operation ca 10 return; 11 12
Result File		
1 2 c class HttpServerDemo { 3 lic static void main(String[] args) throws IOException { 4 netSocketAddress addr = new InetSocketAddress(8080); 5 ttpServer server = HttpServer.create(addr, 0); 6 7 f( args.length == 0 ) 8 9 System.out.println("cancelling..."); 10 System.out.println("aborting..."); 11 System.out.println("the operation can't run..."); 12 return;		



# Distributed Version Control

- Simple commands (create repo with github)
  - mkdir test-git; cd test-git
  - git init
    - Creates .git with meta information
  - create file...
  - git add file.txt
  - git commit
    - Everything still local
  - Connect with remote git repo
    - Github (selfhost with [gitea](#), [gogs](#)), you need to have account, and created the repo
  - git remote add origin [git@github.com:tbocek/test-git.git](#)
  - git push -u origin master
- Now you are ready, check github website

- Clone a repository
  - git clone [git@github.com:tbocek/test-git.git](#)
  - Or https also for readonly
- Workflow
  - Local



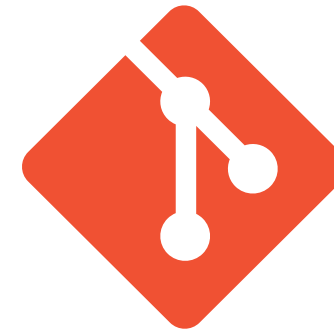
- Remote: pull/push



# Distributed Version Control

- Branching
  - `git checkout -b feature`
  - `git checkout **branch-name**`
    - Switch between branches
  - `git branch -a`
    - Show branches
  - `git branch -d **branch-name**`
    - Remove local branch
  - `git push -d origin feature`
    - Remove remote branch
  - `git push origin feature`
    - Push it remote, before, everything was local
- Merging
  - `git merge feature`
- Get changes from remote
  - [git pull / git pull --rebase](#)
- Go back
  - `git log`
  - `git reset --hard cc5507b071`
  - `git revert cc5507b071..HEAD`
    - Revert last commits
  - Remove passwords, data: [git-filter-branch](#), [bfg](#)
- Tags (release in github specific)
  - `git tag tagname cc5507b071`
  - `git checkout tagname`
  - `git show tagname`
  - `git push origin tagname`
    - To sync it with remote

# Distributed Version Control



# git

- Merge conflict
  - Edit file and fix it
  - `git add *`
  - `git commit`

- Squash
  - `git checkout master`
  - `git merge --squash feature`
  - `git commit`
- Cherry picking
  - `git cherry-pick cc5507b071`