



Software and tools for MUonE

FLAVIO PISANI

UNIVERSITY AND INFN OF BOLOGNA, CERN

Software Needed by the experiment

- ▶ Monte Carlo simulations of the detector
- ▶ Online/Offline alignment and reconstruction algorithms
- ▶ Data analysis software

Monte Carlo simulations of the detector

- ▶ Requirements:
 - ▶ Accurate modelling of MS effects
 - ▶ Precise detector and setup geometry description
 - ▶ Truthful detector response modelling

- ▶ What is already done:
 - ▶ Preliminary setup geometry description in Geant IV
 - ▶ Preliminary study on the MS effects

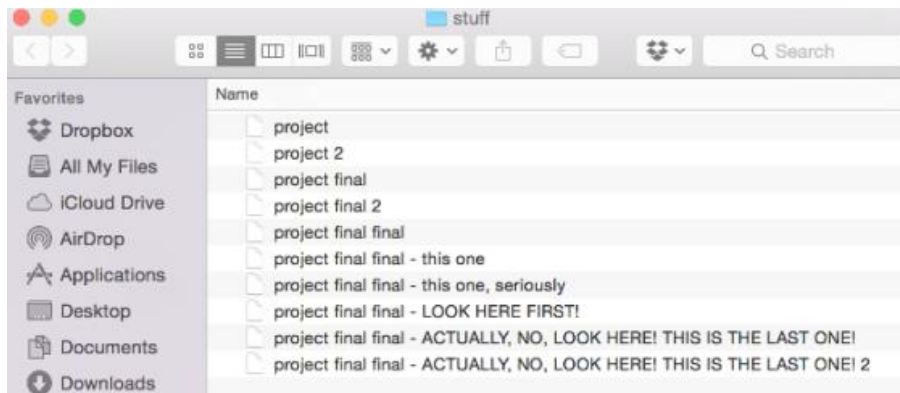
There is a lot of work to be done by different people

Software tools for developing

- ▶ Versioning tools (CVS, SVN, Git)
 - ▶ The code grows and keep track of changes is complex
 - ▶ The team grows and coordinating people is difficult
- ▶ Building testing and packaging tools (CMake, CMT)
 - ▶ Compile the code on different platforms is challenging
 - ▶ Handling all the dependencies can be a nightmare

Don't let your project grow before start using the tools

Things to avoid



- How do I build your project?
- It is very easy, just run this 2000 lines bash script that I wrote.

Versioning tools

A versioning tool is not:

- ▶ Necessarily a backup
- ▶ Usually meant to be used with binary files
- ▶ A magic wand that protects you and your colleagues from bad practices

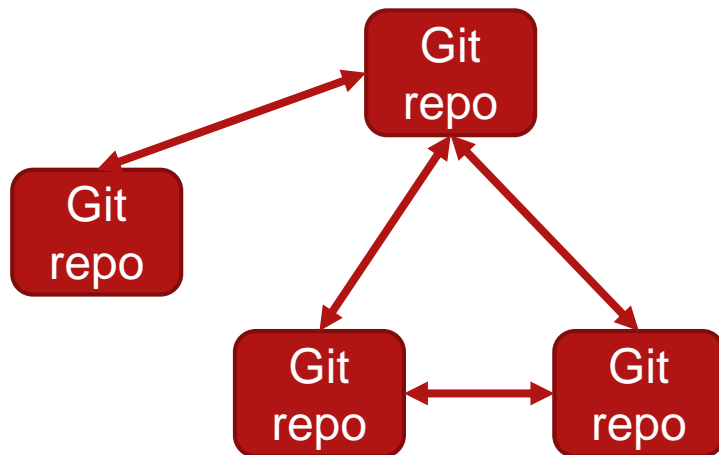
A versioning tool is:

- ▶ A powerful tool that helps you keep track of your code
- ▶ An easier way of integrating contributions to a project
- ▶ A helping hand for tracking down bugs and issues

Versioning tools

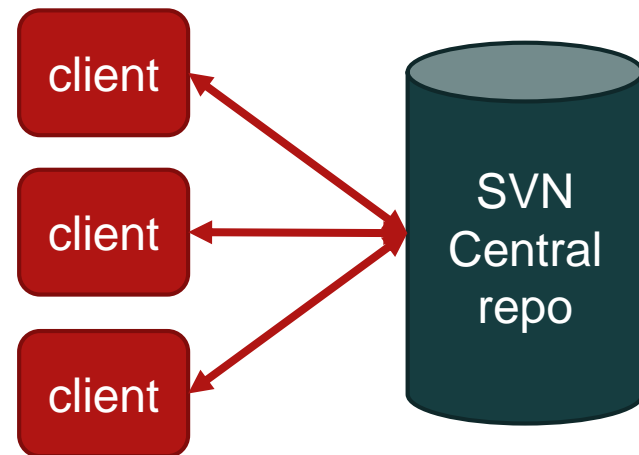
Git

Distributed versioning



SVN

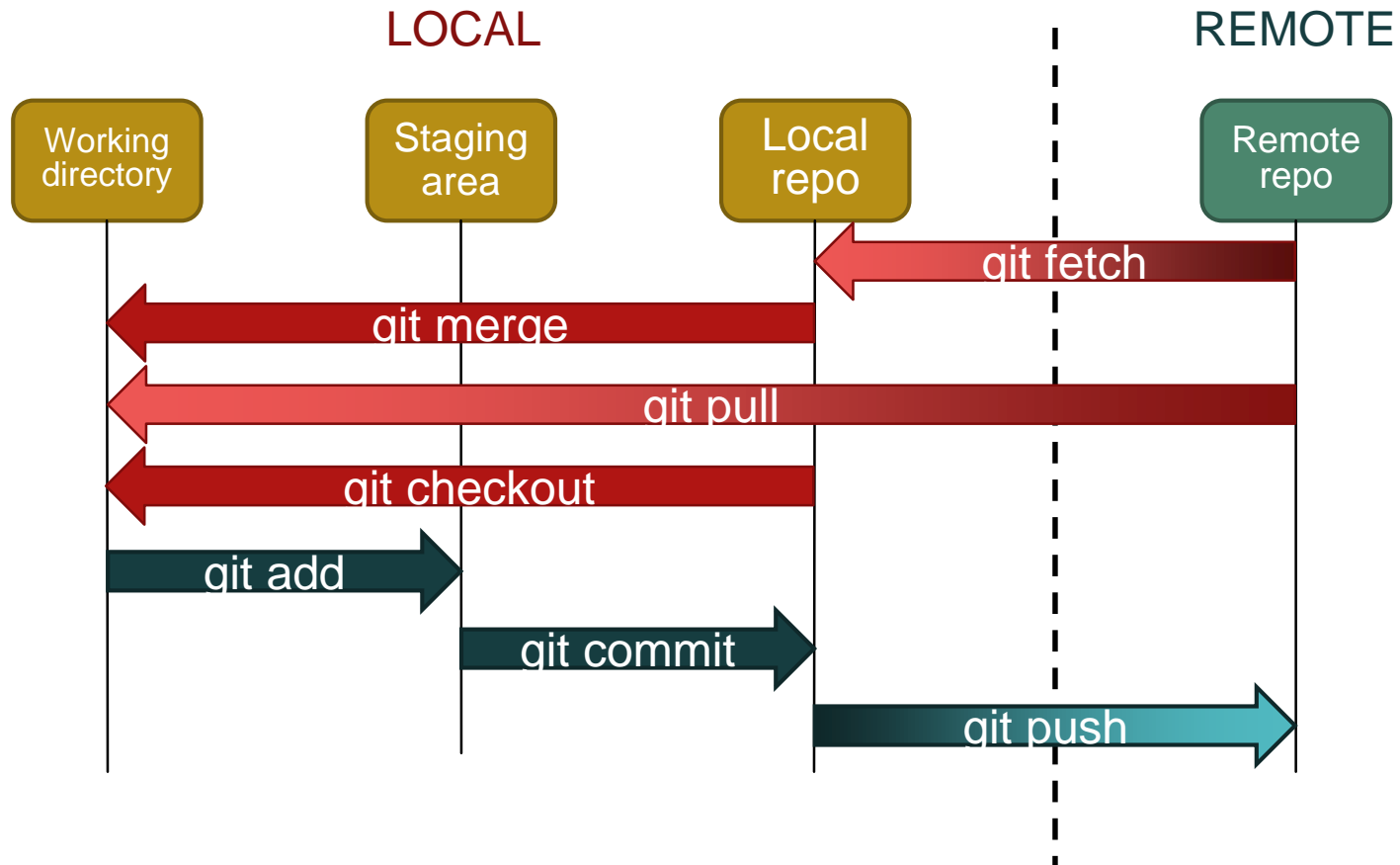
Centralized versioning



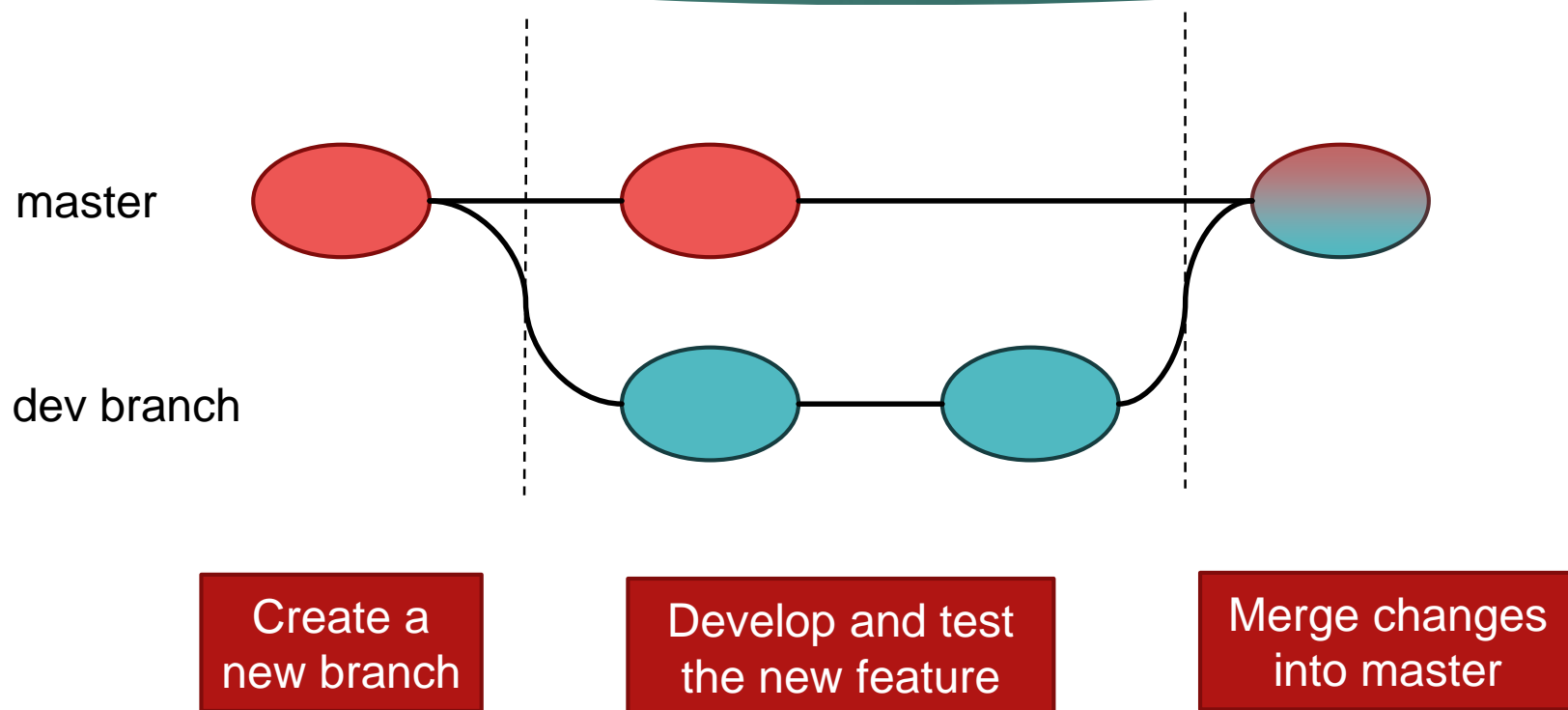
Git basic concepts

- ▶ Repository:
 - ▶ Stored history of the changes to tracked files
 - ▶ Can be local or remote
- ▶ Commit:
 - ▶ A single point in the Git history, it Defines a specific version for all the files in the repo.
- ▶ Branch:
 - ▶ An active line of development.
 - ▶ Multiple branches can exists in parallel but only one is active on the working tree

Git operations



Git workflow example



Remote repo hosting

- ▶ Usually a web based service is available for managing repos ACLs and requests
- ▶ Typical workflow
 - ▶ Copy the repo on your computer (git clone)
 - ▶ Update and commit the code (keep the master branch clean)
 - ▶ Make a request to merge you changes into master (pull/merge request)
- ▶ Widely used services: GitHub and GitLab

Few words on CMake

What CMake is?

- ▶ CMake is a tool for building and testing software packages

Why CMake?

- ▶ It is system and compiler independent
- ▶ It requires minimal configuration (compared to other build systems)
- ▶ It is easy to manage and maintain

CMake basic workflow

- ▶ Write CMake configuration file(s) CMakeLists.txt
- ▶ Run CMake and generate all the build folder structure and makefiles
- ▶ Make your project

```
$>mkdir build  
$>cd build  
$>cmake ..  
$>make
```

Summary

- ▶ There is a significant and non trivial amount of software to write
- ▶ This works will be shared among different people
- ▶ To ease this task we should use the proper development tools
- ▶ Because this is a new project we should start using the tools as early as possible

Thank you for your attention

In case of fire



1. git commit



2. git push



3. leave building