Getting Git

Nolan Chinn

What is Git?

- Version Control System:
  - Records changes to files over time
  - Captures snapshots of data
- Snapshots not Differences
  - Other version control systems mark differences
  - Saves the state of the entire project
- Store multiple versions simultaneously

Why use Git?

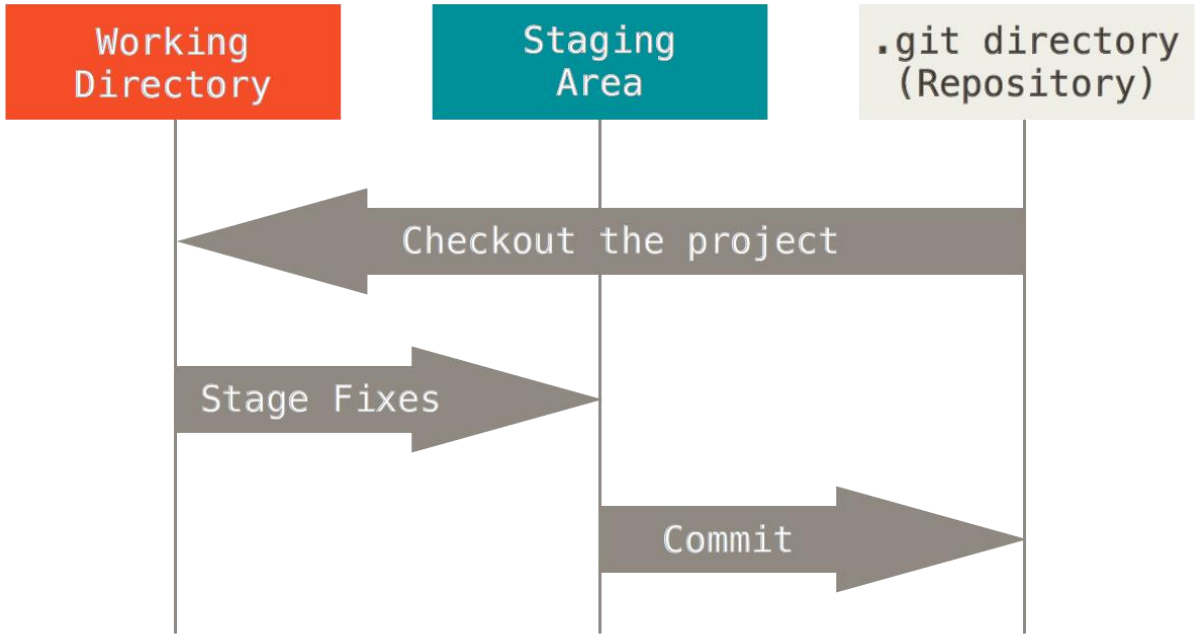
- Backup
- Collaboration
- Safe space to try stuff out
- Modification
- Customizable
- More than just programming
- Common tool

History

- Began during the development of the Linux kernel
- In 2005 the version control system used by the developers (BitKeeper) stopped offering their services for free
- Linus Torvalds and the Linux development team wrote their own solution

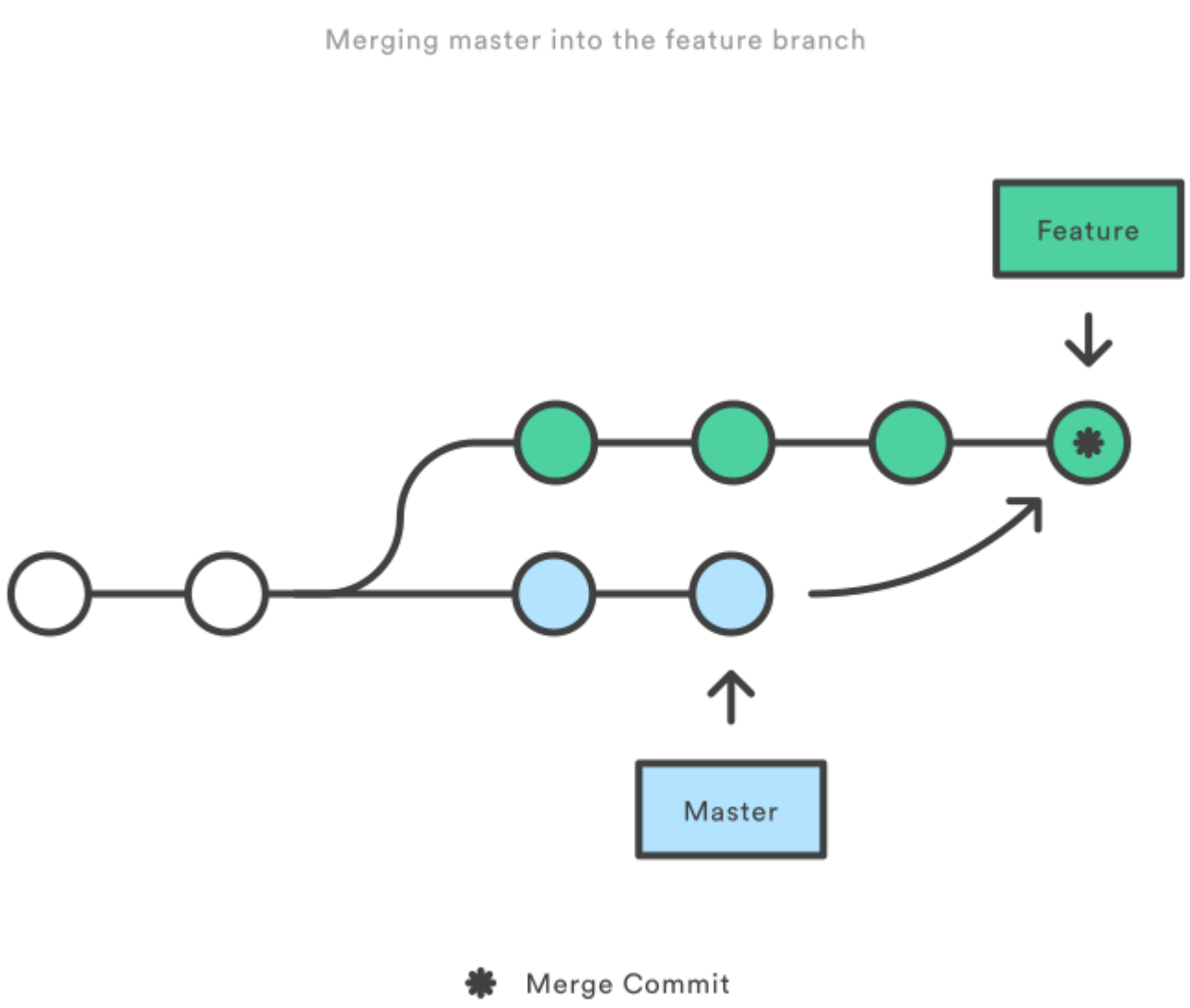
Components

- Working Directory (Modified)
  - Files in current snapshot
- Staging Area (Staged)
  - Modified files ready to be saved
- Repository (Committed)
  - All snapshots in the project
- Branches
  - Different chains of snapshots



Basics

- Initialize repository
- Add files
- Commit changes
- (Branch)
- Modify files
- Commit changes
- (Merge)

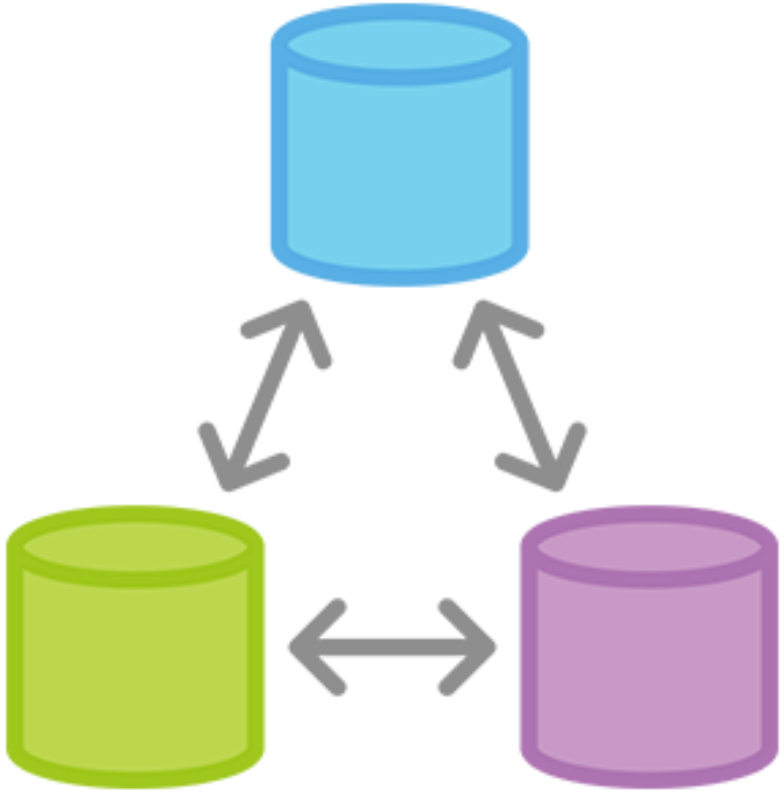


Core Concepts

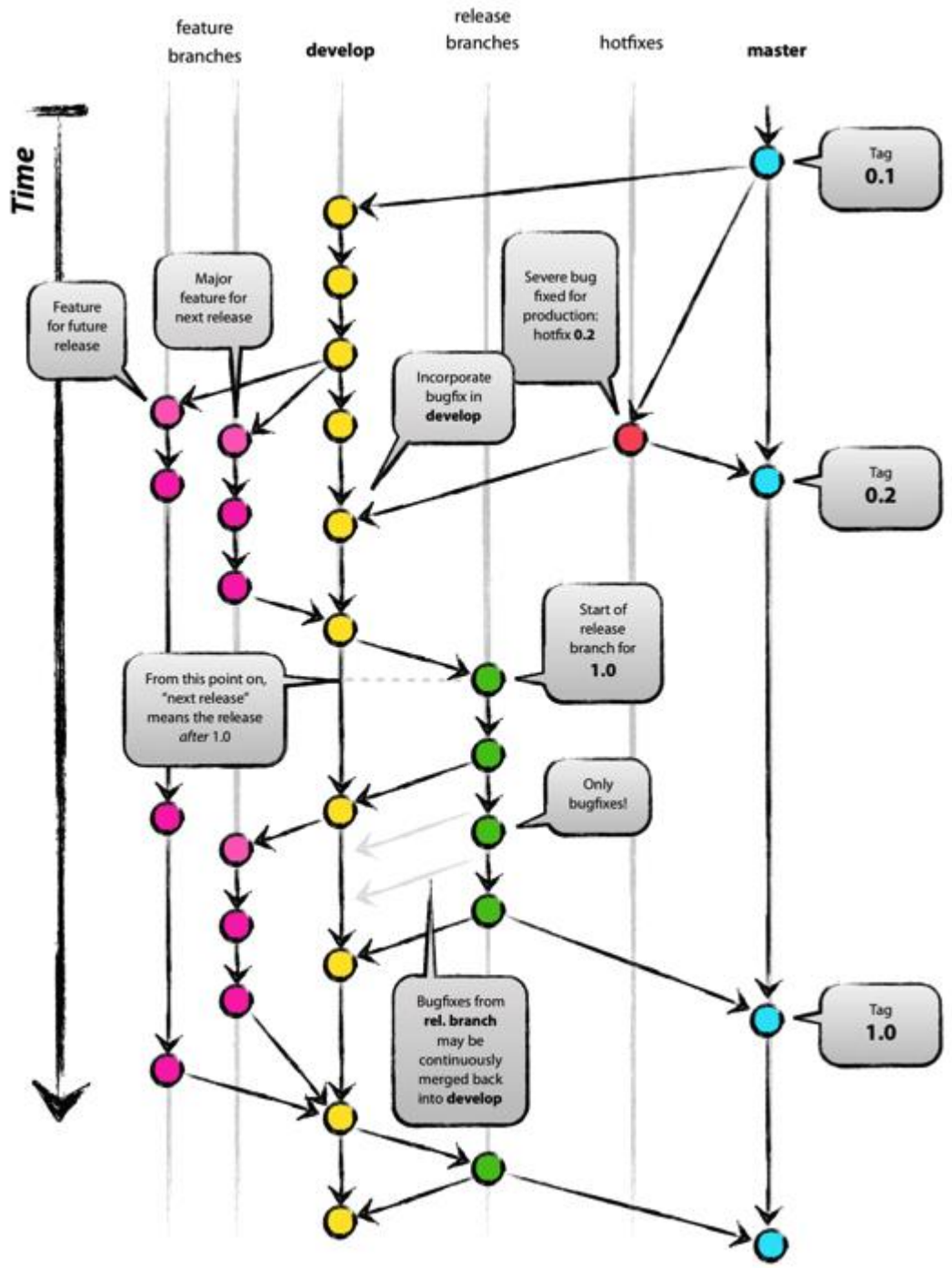
- Repository: The collection of all the project’s snapshots
- Master branch: The main branch all others stem from
- Commit: A snapshot of the files in the project at a certain point
- Head: The snapshot in the working directory

Remote Repositories

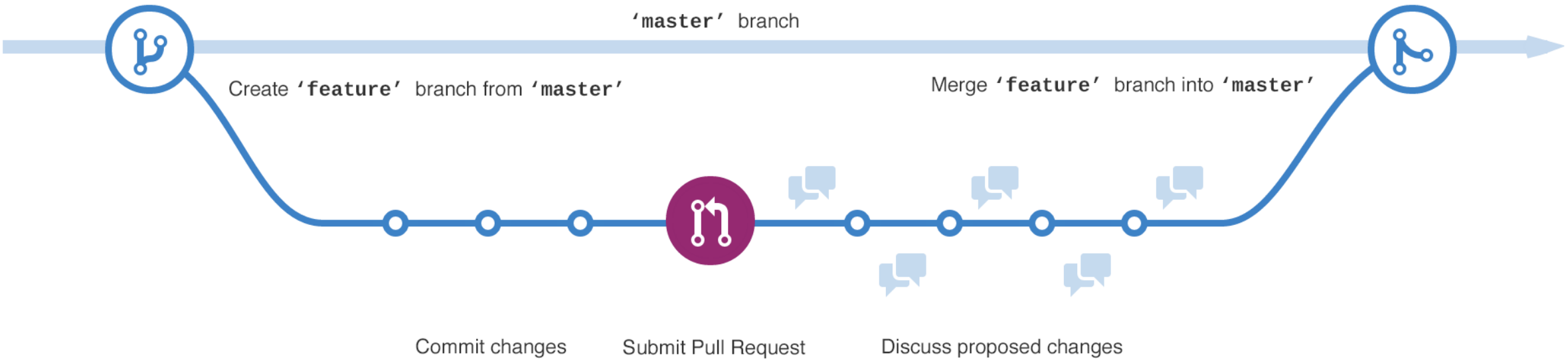
- Pushing/Pulling
- Common Solutions:
  - GitHub
  - GitLab
  - BitBucket
  - etc.



Git Flow


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

GitHub Flow


'master' branch
Create 'feature' branch from 'master'
Merge 'feature' branch into 'master'
Commit changes
Submit Pull Request
Discuss proposed changes

Common Commands

git init

git add <filename>

git commit -m “commit message”

git status

git checkout –b <branchname>

git merge <branchname>

git checkout <branchname>/<commit>

git pull

git push origin <branchname>