

Beyond git

# Hosting services ("forges")

These web-based services provide:

- Repository hosting (e.g. a remote everyone can use)
- Code browsing
- Merge requests / code review
- Issue tracking / social features
- Access control
- Release management

Centrally hosted (free or paid):

- **GitHub**
- **GitLab**
- **Savannah**
- **sourcehut**

Self hosted:

- **Gitea**
- **Gogs**
- **gitweb** (code browsing only)
- **cgit** (code browsing only)

*demo time!*

# A history of version control

- SCCS (1977)
- RCS (1982)
  - Single files
- CVS (1986)
- *Perforce* (1995)
  - Proprietary
  - Centralized
  - Good at tracking binary files
  - Used by many game developers
- *Subversion* (2000)
- BitKeeper (2000)
  - Proprietary
  - Used by Linux for a while
- Darcs (2003)
- Git (2005)
  - Built by Linus for Linux
- Mercurial (2005)
- Fossil (2006)
- Pijul (2020)
  - Based on a sound theory of patches

(Systems in *italics* rely on a centralized server.)

# Case study: Git vs Mercurial

- Mercurial does not have an equivalent to Git branches
  - Each commit is individually marked as shown or hidden in the DAG
  - No need for branches to dictate which commits "matter"
  - Lines of development can be tracked using *bookmarks*, which serve a similar purpose
- Rebasing and rewriting history are more acceptable
  - In part due to the lack of branches, Mercurial's tooling for handling rewritten history is better
  - Very easy to move bookmarks around
- Mercurial has no staging area
  - A file's contents are not remembered at all until it's committed
  - Simpler to reason about, but results in more amends
  - Better UI for selecting changes to go into a specific commit (`hg commit -i`)

# Source control at Big Tech

## Challenges:

- Huge numbers of commits and committers: log/blame still has to be fast
- Tightly integrated code: need to change multiple codebases atomically
- Total size of code is very big: cannot all fit on a single workstation

## Solutions:

- Extend existing DVCS to have centralized storage and tight tooling integration
  - Facebook: [EdenSCM](#)
  - Microsoft: [VFS for Git](#)
- Build custom solution on top of existing Big Data infrastructure
  - Google: Piper

# Git GUI wrappers

- **gitk**
  - Open-source, distributed alongside Git
- **GitKraken, SourceTree**
  - Closed-source, free GUIs
- **GitHub Desktop**
  - Closed-source GUI
  - Integrates tightly with GitHub remotes
  - Supports GitHub Pull Requests, Issues

# Using Git from your IDE/editor

Some editors allow you to see and commit changes inside their environment:

- **VS Code**: built-in (see demo)
- **Sublime Text**: built-in (Merge)
- **Emacs**: Magit
- **Vim**: fugitive

These integrations are usually not nearly as fully-featured as the CLI or a dedicated GUI, however.



# Separate code review tools

- Gerrit
- Phabricator

What questions do  
you have?