Git Version Control System

Petr Baudiš ⟨pasky@suse.cz⟩

Motivation

- Version control
- Distributed nature

Motivation

- Version control
- Distributed nature
- Scales extremely well
- Powerful interface

Motivation

- Version control
- Distributed nature
- Scales extremely well
- Powerful interface
- Git history

What will we talk about

- The Git Theory
- Basic Usage Tutorial
- Git and the World
- The Wizardry

Resources

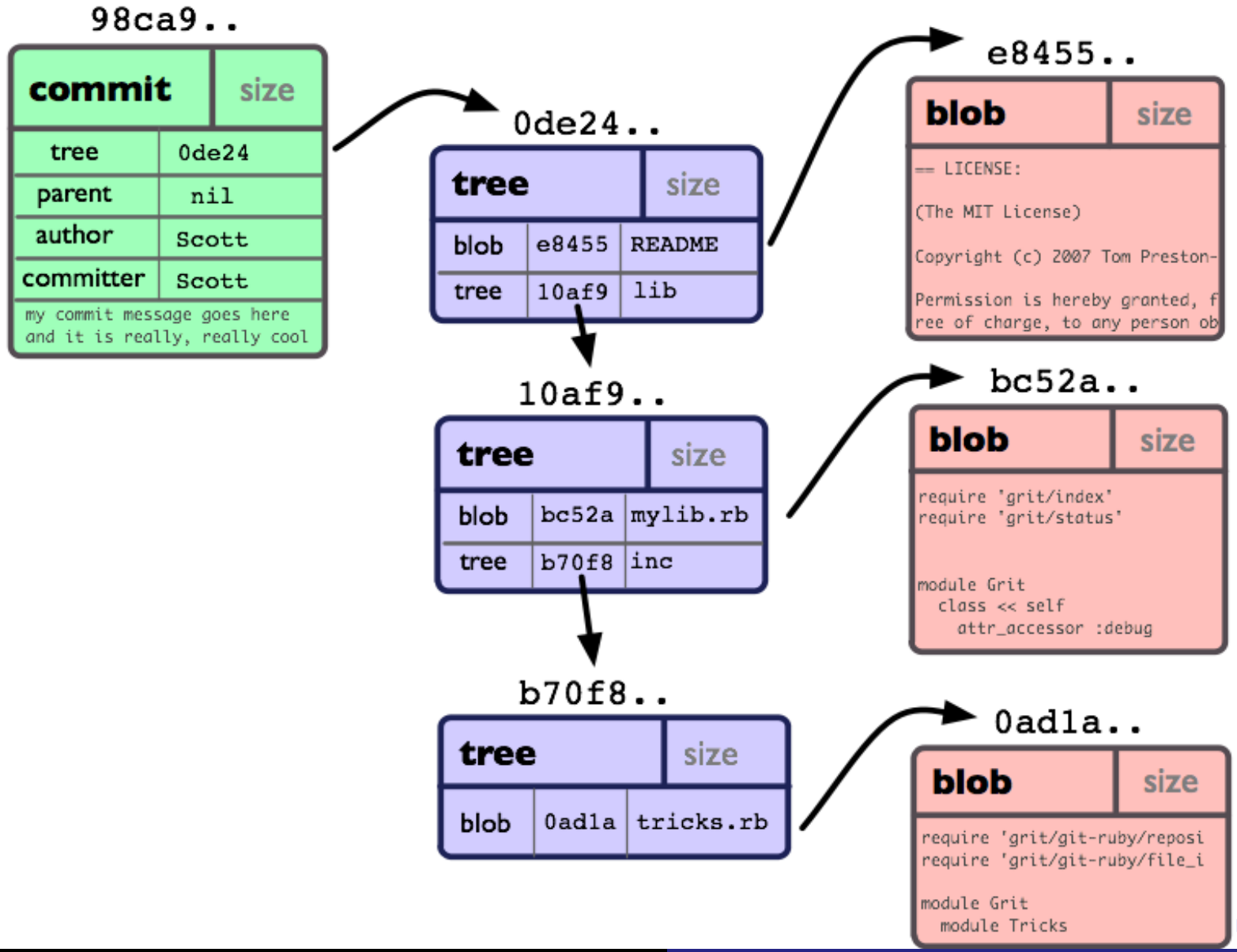
- Pro Git, Git Magic, Git Book
- Git User Manual
- Manual pages, cheat sheets

Outline

Git Places

- **Repository:** Special directory; object database, branches information, configuration
- **Working copy:** Part of non-bare repository - checked out copy where you hack away
- **Index:** Staging area where next commit is built

Git Object Model



Commit Graph

- Commit object contains list of parent pointers
- Commits form directed acyclic graph (tree + merges)
- **Ref:** Named commit (object) pointer
- **Tag:** Static ref (possibly with extra metadata)
- **Head:** Forward-moving ref
- **Branch:** Slight abstraction of "head"

More about refs

- **HEAD** is a "meta-ref" representing your current branch or commit
- **master** is the default name of the default branch

More about refs

- **HEAD** is a "meta-ref" representing your current branch or commit
- **master** is the default name of the default branch
- Refs are cheap! Branches are not "special"!
- $\Rightarrow$ topic branches, throwaway "work-in-progress" branches and stashes

Usage Philosophy

- Porcelain vs. plumbing
- Porcelain: User-friendly high-level commands
- Plumbing: Scriptable low-level commands
- Somewhat intermixed

Outline

Basic Usage Tutorial

- `git config --global user.name "Vladivoj Vomacka"`
- `git config --global user.email vomacka@example.com`

Basic Usage Tutorial

- `git config --global user.name "Vladivoj Vomacka"`
- `git config --global user.email vomacka@example.com`
- `git init`, `git add`, `git commit`, `git clone`
- `gitk`, `git-gui`, `tig`
- `git status`, `git diff`, `git rm`, `git mv`
- `git log`, `git show`
- `git checkout`, `git branch`, `git merge`
- `git checkout`, `git reset`

Basic Recipe — Tracking Your Sources

- `git init`
- `git add .`
- `git commit -a -m"..."`

Basic Recipe — Tracking Their Sources

- `git clone git://git.kernel.org/pub/scm/linux/hotplug/udev.git`
- `git pull`

Outline

Commit-hauling

- `git clone`, `git pull`
- **origin**, remotes
- `git remote`, `git push`
- Hosting: repo.or.cz, GitHub, Gitorious

Cooperating

- Pulling vs. rebasing
- Submitting patches: `git format-patch`, `git send-email`, `git request-pull`, StGit, TopGit
- Accepting patches: `git am`
- Bare repositories

Outline

Random Tidbits

- `git bisect`
- Cherry-pick, revert
- Pickaxe (`git log -S`), blame, grep, show again
- `git add -p`
- `git rebase -i`
- `git archive`
- A cursory look: Stashes, reflog, refspecs, branch filtering, submodules, notes, bundles, grafts, hooks

Thank you!

Thank you! Final questions?