Git Kung Fu

Bartosz Majsak, Thomas Hug

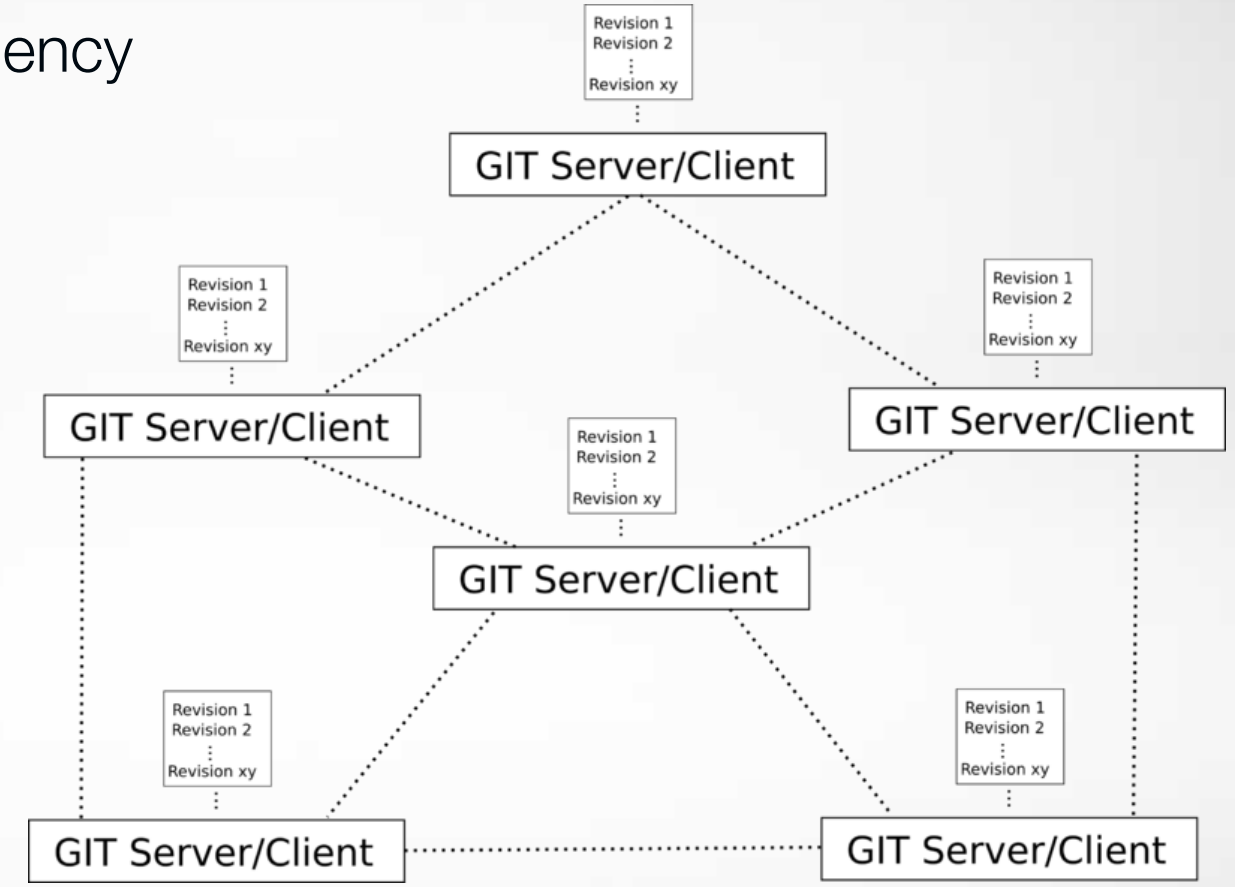
About Us

- Bartosz Majsak
  - Java Developer by day
  - Open source junkie by night (Arquillian core team member)
  - Conference speaker by passion (Devoxx, Jazoon ...)

- Thomas Hug
  - With Cambridge Technology Partners since 2002
  - Java Developer, TTL, Solution Architect
  - Apache Committer, OSS contributor and aficionado

Git Concepts

- No Central Server – Distributed VCS
- Performance and Efficiency
- Robustness

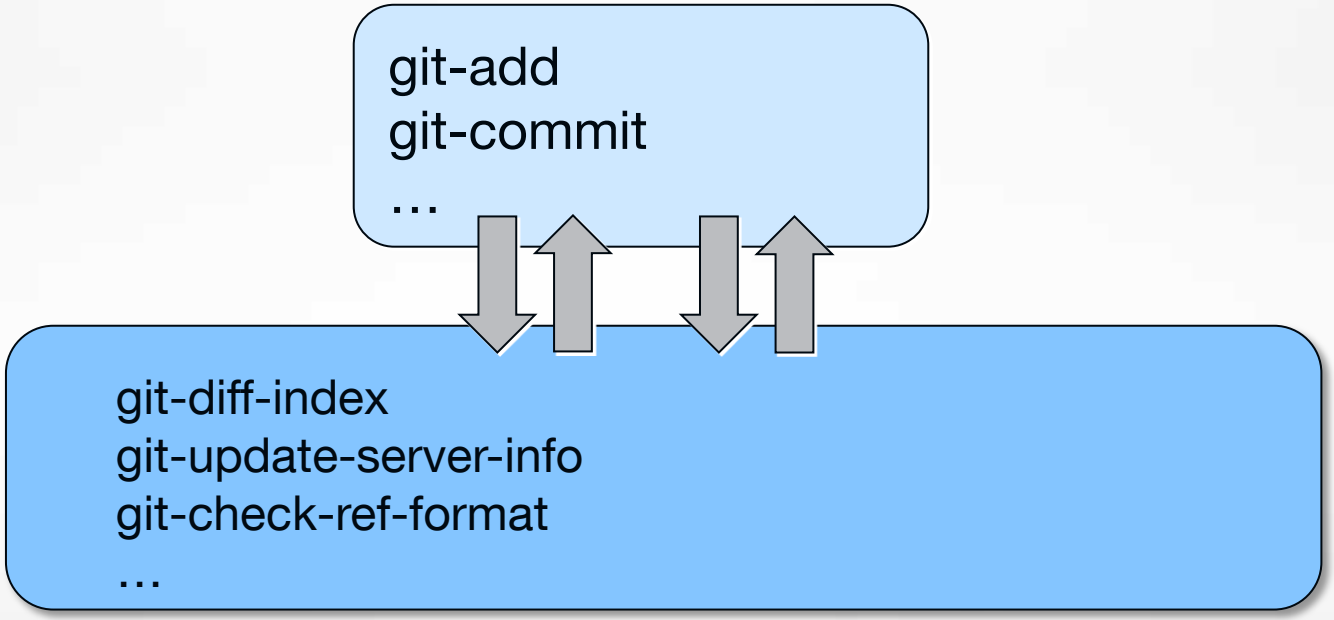


Cambridge Technology Partners

Git Internals

Git Architecture

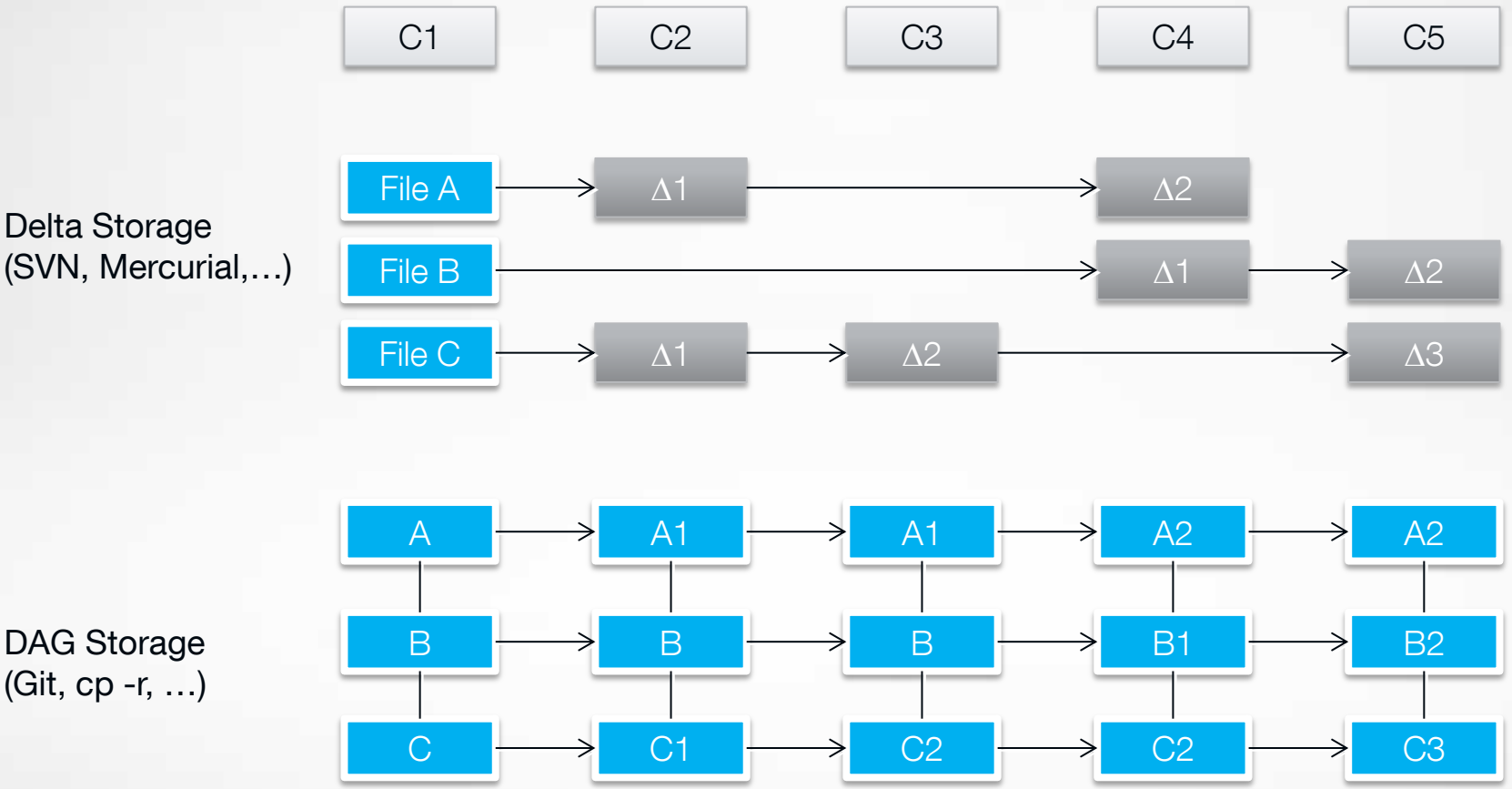
- Plumbing and Porcelain
  - Composition of low-level commands into high-level ones
  - Unix design principles
- Local as much as possible



Git Storage

Cambridge Technology Partners

- Delta storage vs. Directed Acyclic Graph (DAG)



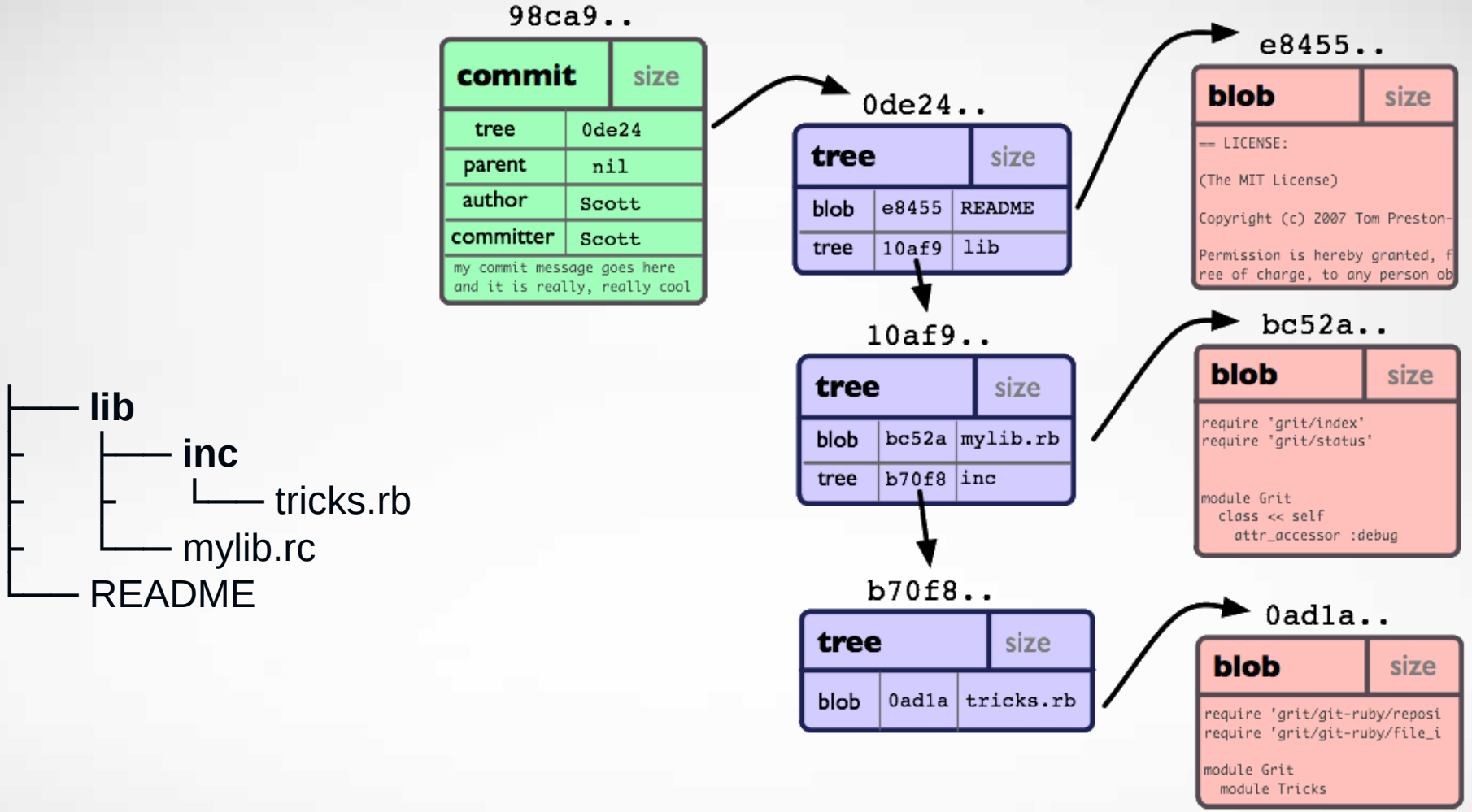
Git Storage – Object Model (1)

Cambridge Technology Partners

- Git tracks content, not files
- Content identified by 40 character SHA1 hash
  - Modified content easily identifiable
  - Immutable in the object database
- Objects: Blob, Tree, Commit, Tag
- References: HEAD, Tags, Remotes
  - Mutable, pointers to commits

Empty directories are not considered as content.
Add an empty `.gitignore` if you need a folder tracked.

Git Storage – Object Model (2)



Git Storage – Local Repository

Cambridge Technology Partners

- The repository .git directory

```
$ cd .git
$ tree -L 1
├── branches        # Pointers to branches
├── config          # Repository local configuration
├── description     # Repository description
├── HEAD            # Pointer to HEAD in current branch
├── hooks           # Pre- and post action hooks
├── info            # Additional information about the repository
├── objects         # Object database
└── refs            # Pointers to branches
```

Cambridge Technology Partners

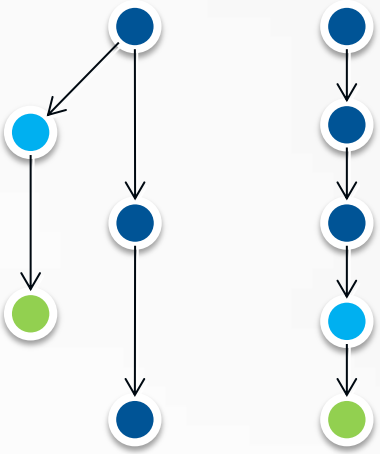
Rewriting History

Rebasing

Cambridge Technology Partners

```
$ git checkout master
$ git rebase mybranch
```

git rebase



Rewriting history: Interactive rebase last four commits

```
$ git rebase --i HEAD~4
```

All your base are belong to us

Objectives: Learn how interactive rebasing works.

- Experiment with interactive rebase on selected branch
  - Reword commit messages
  - Combine commits into one

```
$ git rebase –i [commits range]
```

Cherry-pick

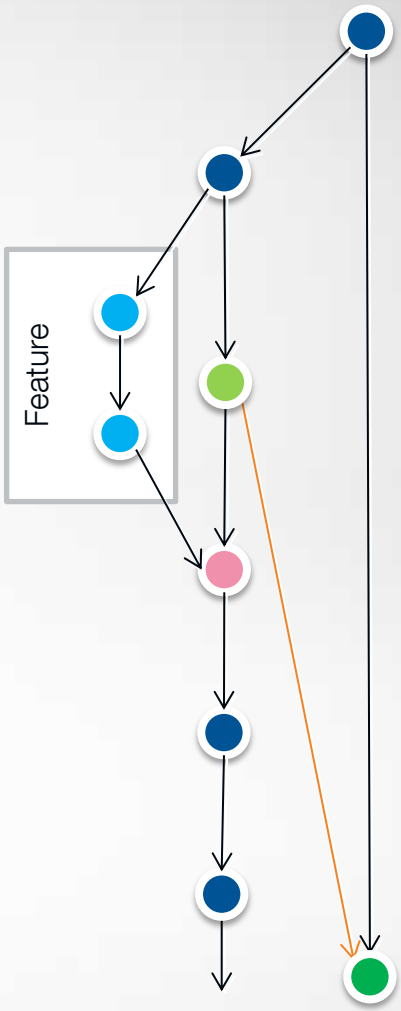
Cambridge Technology Partners

```
$ git cherry-pick [-x]
```

Cherry-pick „replays” arbitrary commits onto your current branch.

```
$ git cherry -v <other_branch>
```

Lets you check if given commit from other branch has been already applied on the current branch



Filter branch

Cambridge Technology Partners

```
$ git filter-branch
```

- Removing a file from every commit
- Changing commit metadata globally

Cambridge Technology Partners

Recovering from Mistakes

Typos

Is there a way to fix poor commit messages?

```
$ git commit --amend
$ git rebase --i HEAD~X
```

Fixing Commits and Staging Area

Cambridge Technology Partners

For not yet pushed commits:

```
$ git commit --amend
```

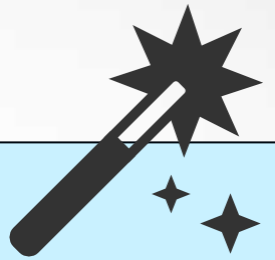
git reset

Unstage a file:

```
$ git reset HEAD file.txt
```

Discard local changes:

```
$ git checkout -- file.txt
```



Fully revert to a previous commit:

```
$ git reset --hard HEAD
```

Revert

```
$ git revert HEAD|hash|parent
```

git revert

One bridge too far

Disaster recovery.

What if...

```
$ git reset --hard HEAD^

$ git reflog
$ git reset --hard HEAD@{X}
```

Who broke the build?!

Cambridge Technology Partners

```
$ git blame FILE
```

git blame

```
$ git bisect start
$ git bisect bad
$ git bisect good <HASH>
```

git bisect

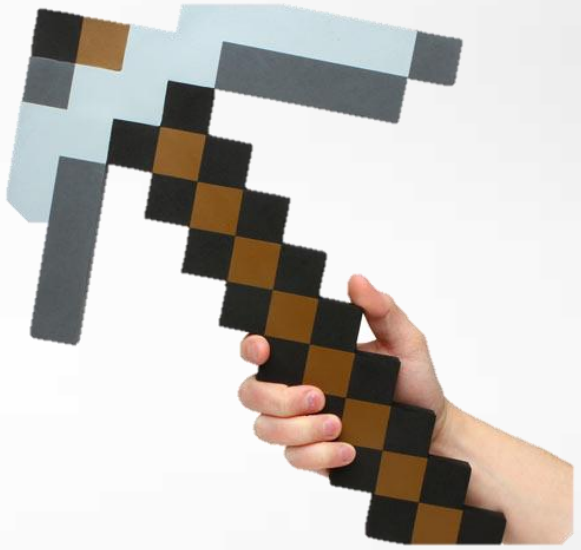
Who broke the build?! - Pickaxe

Cambridge Technology Partners

```
$ git log –S'search_term' [-p] <resource>
$ git log –G'regex'
```

git log –S|G

```
--name-status
--pickaxe-all
```



Sharing without network

Archives

```
$ git archive --format zip --output "repo.zip" master
```

git archive

Bundles

Cambridge Technology Partners

git bundle

```
$ git bundle create \
  ../jazoon.bundle master HEAD
$ git bundle create <filename> <refs ...>
$ git bundle create ../jazoon.bundle \
  --since="one week ago" master

$ git bundle verify /tmp/jazoon.bundle
$ git pull /tmp/ejmr.bundle master:master
```

You can also add a bundle as a remote:

```
$ git remote add bundle /path/to/bundle.bundle
```

Cambridge Technology Partners

Repeat Yourself

Repeat Yourself

Repeat Yourself

Reuse Recorded Resolution (ReReRe)

```
$ git config rerere.enabled true
… # create a merge conflict
$ git rerere status
$ git rerere diff
… # resolve conflict
$ git rerere diff
… # commit, reset hard HEAD^1, redo merge
```

git rerere

Evict old recorded resolutions from repository:

```
$ git rerere gc
```

Cambridge Technology Partners

Other tricks

Sparse Checkouts

Cambridge Technology Partners

```
$ git init
$ git config core.sparsecheckout true
$ echo path/to/checkout \
      >> .git/info/sparse-checkout
$ git remote add -f origin ...
$ git pull origin ...
```

You can also push to sparse checkout repositories

Orphan Branches


```
$ git checkout --orphan [branch]
$ git rm -rf *
```

Use for:

- Documentation (combine with hooks or CI server!)
- Resources

Have a look at the GitHub **`gh-pages`** for some ideas what orphan branches can do.

Notes

Cambridge Technology Partners

Annotates existing commits.

```
$ git notes add HEAD

$ git notes add –m’Fixes everything’ HEAD

$ git notes --ref=jazoon edit master~1

$ git push origin refs/notes/jazoon
```

Other goodies

Cambridge Technology Partners

```
$ git diff --word-diff --color-words
$ git config --global help.autocorrect 1
$ git rebase HEAD~3 -i --autosquash
```

Cambridge Technology Partners

Hooks

Git hooks

Cambridge Technology Partners

```
$ cd .git/hooks
```

Client-side

- pre-commit
- prepare-commit-msg
- commit-msg
- post-commit

Server-side

- pre-receive
- post-receive
- update

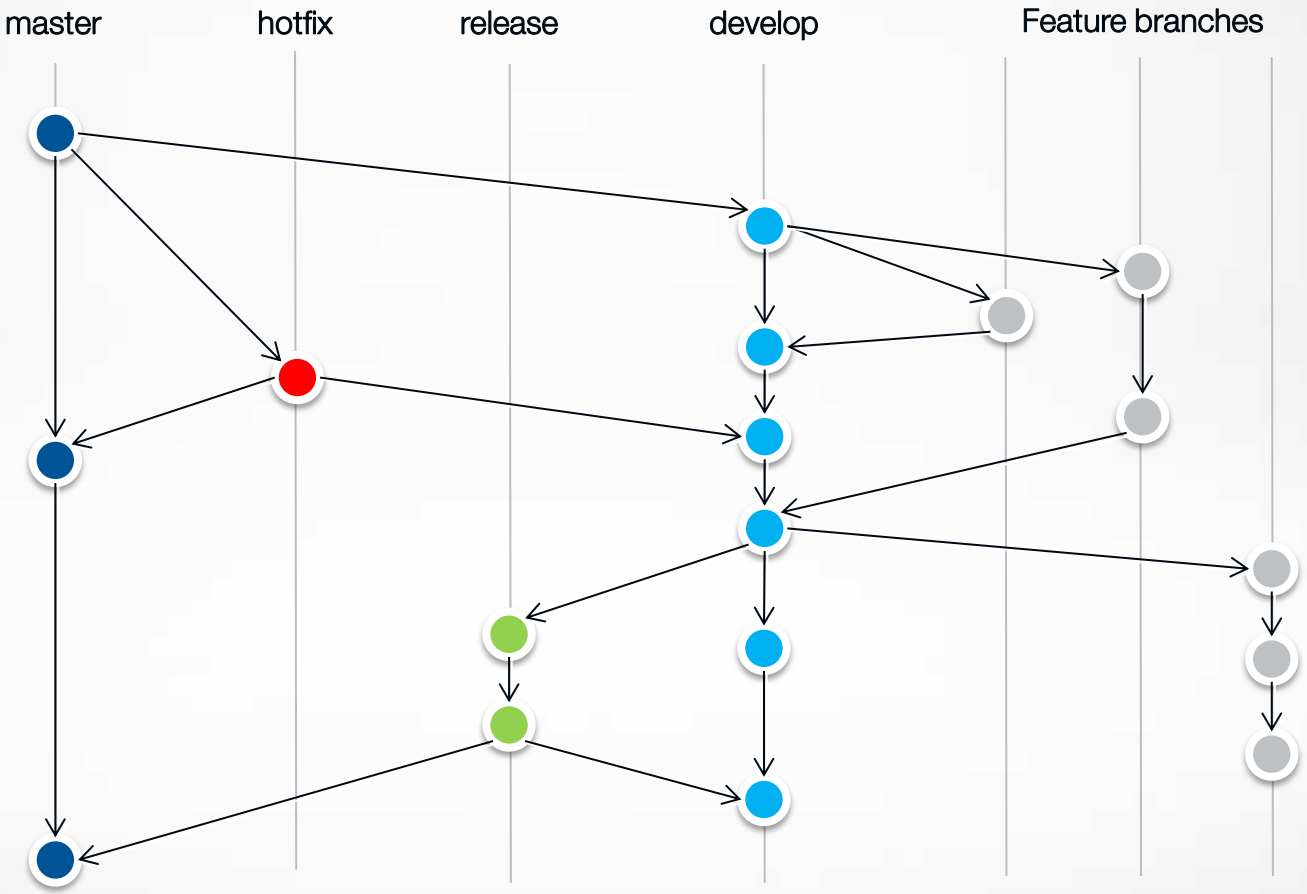
Cambridge Technology Partners

Workflows

Git Flow

```
$ git flow init
```

git flow

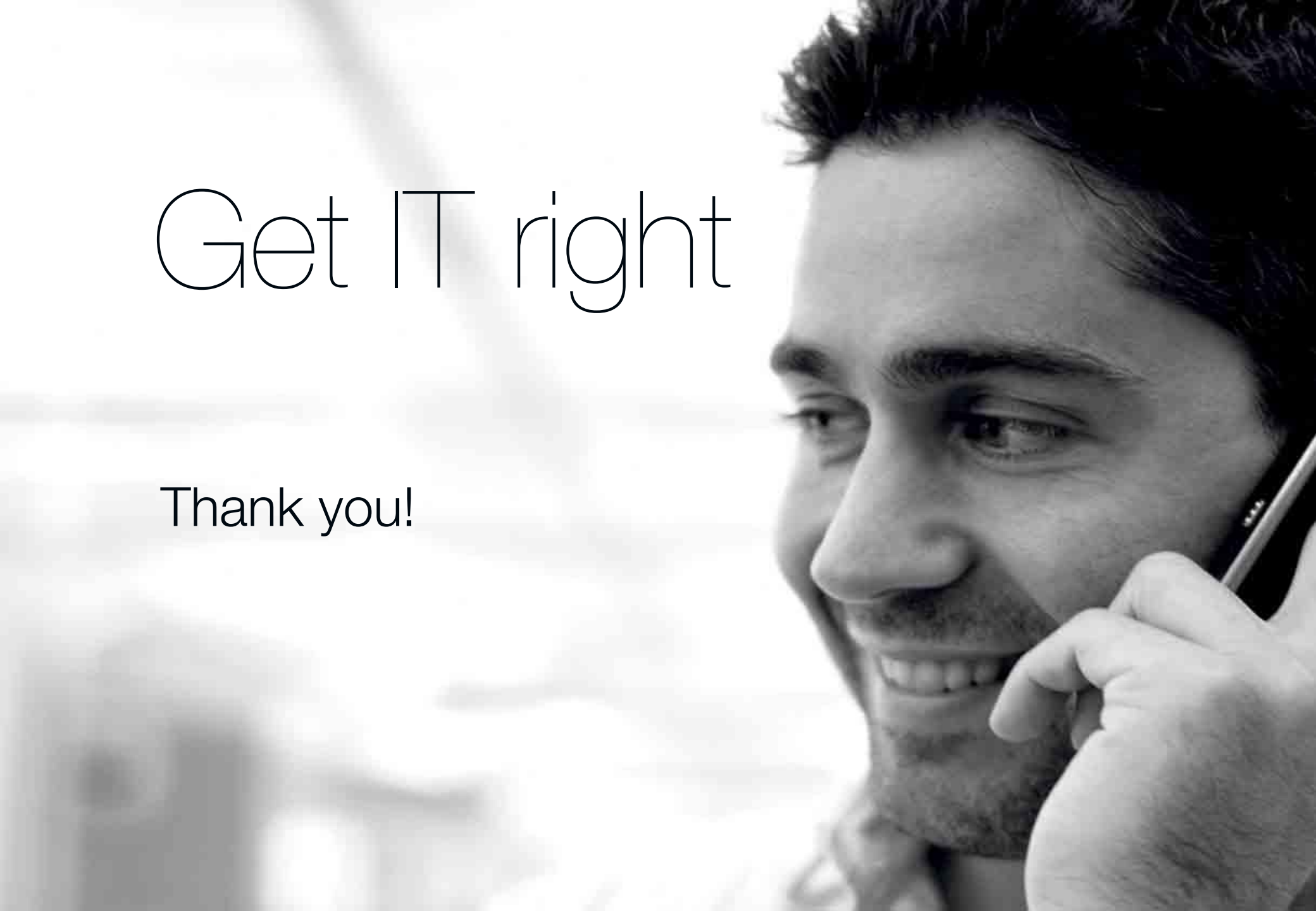


Cambridge Technology Partners

Git in the Enterprise

Get IT right

Thank you!



Credits

- Icons provided by Icons8: http://icons8.com/