intro to git

李希鹏

lixipeng@software.ict.ac.cn

lisnb.h@hotmail.com

git

Distributed Version Control System

eclipse
GNOME
Microsoft
Linked in
facebook
Google
NETFLIX
android
PostgreSQL
RAILS
Qt
K

git

≠

GitHub

git ≠

GitHub

Google code

GitCafé

Share a cup of Open Source.

source forge

Atlassian Bitbucket

git

≠

GitHub

SUBVERSION

Google code

GitCafé
Share a cup of Open Source.

mercurial

source
forge

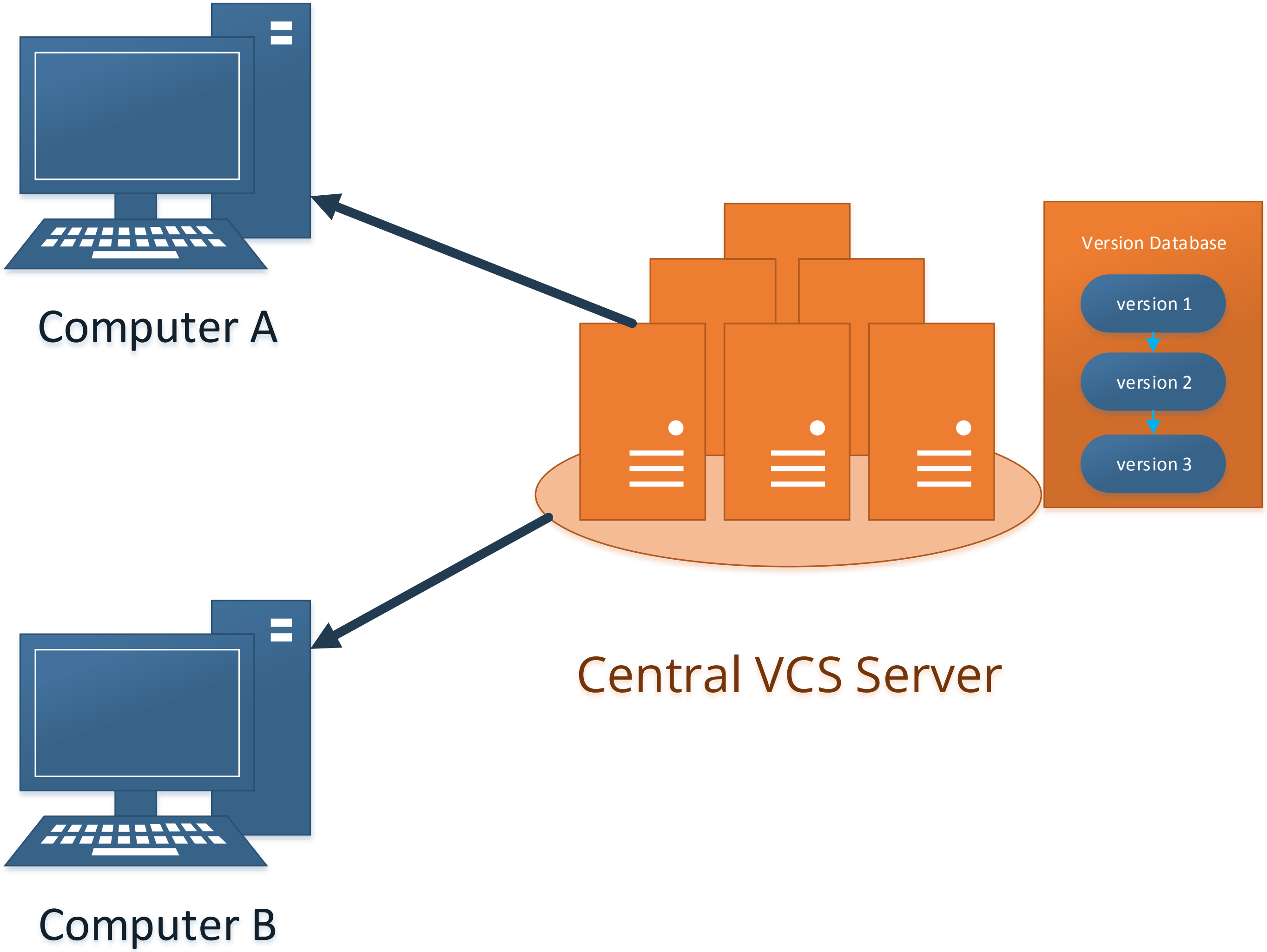
BITKEEPER

Atlassian
Bitbucket

git
GitHub
SUBVERSION
Google code
GitCafé
Share a cup of Open Source.
mercurial
source
forge
BitKeeper
Atlassian
Bitbucket

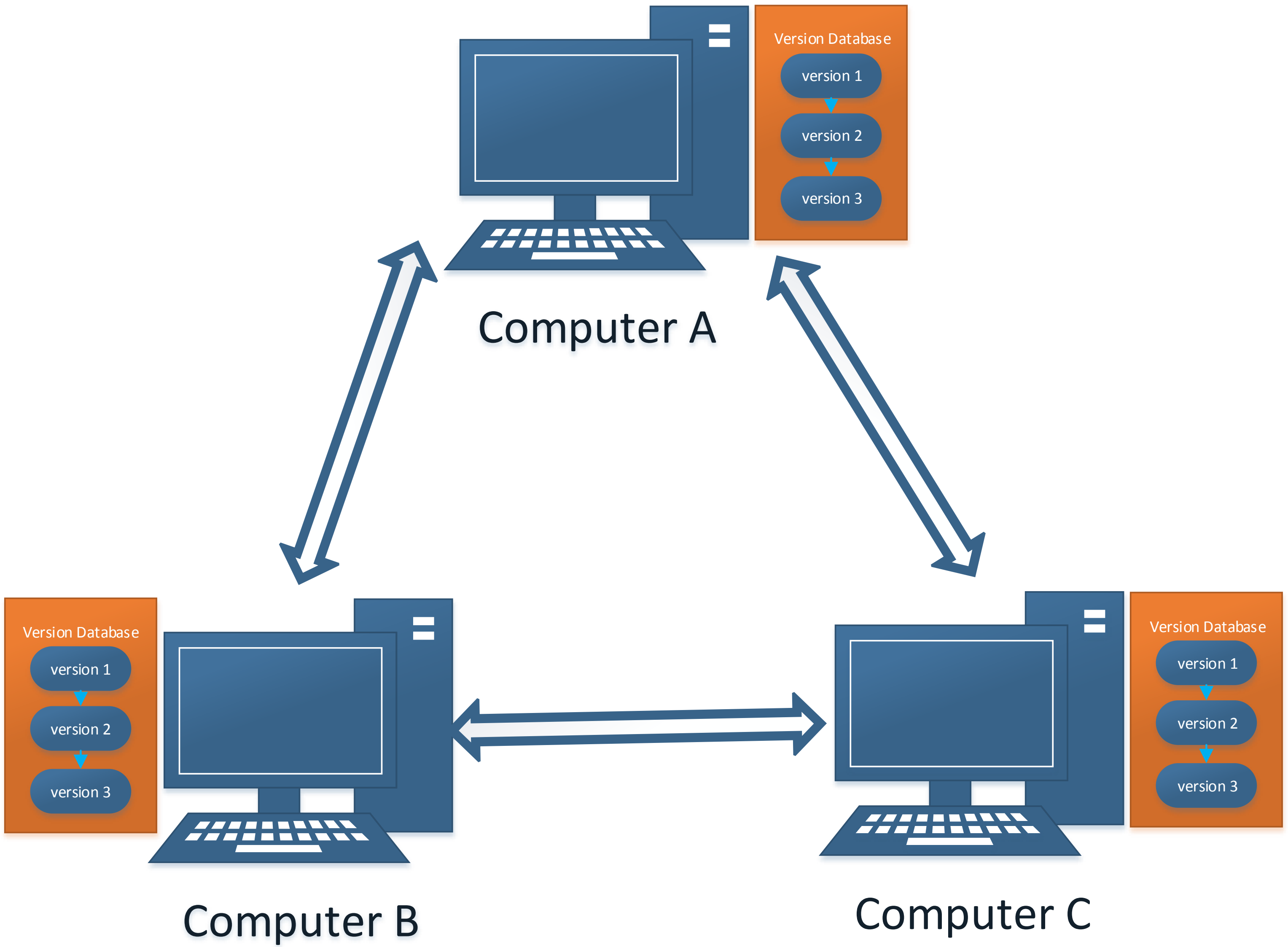
集中化的版本控制系统

Centralized Version Control System



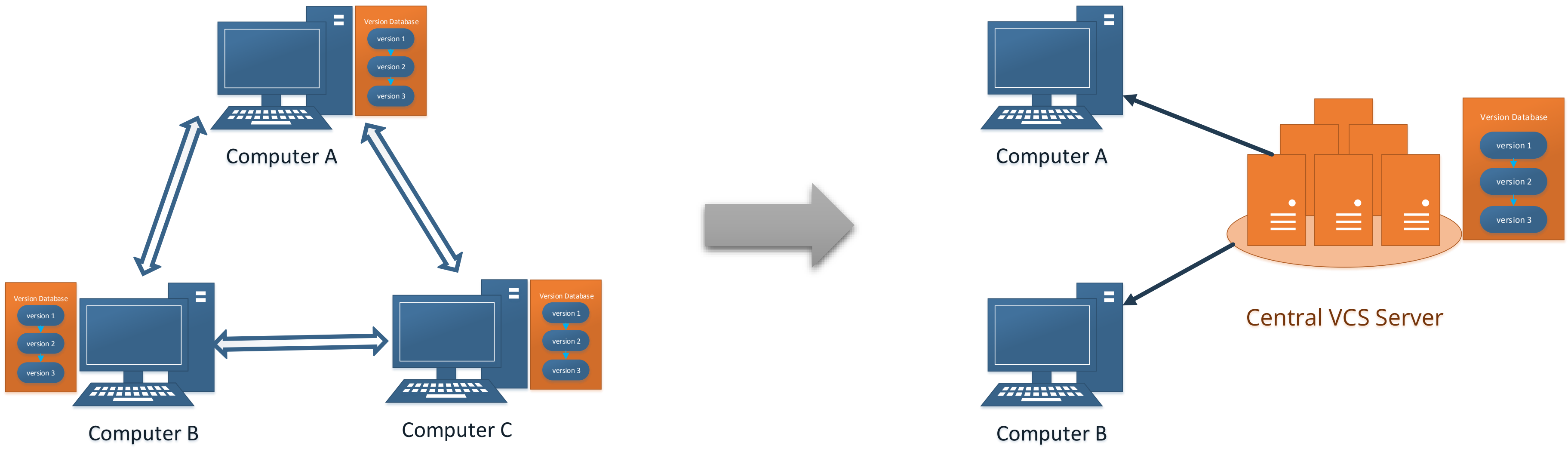
分布式的版本控制系统

Distributed Version Control System



git

SUBVERSION

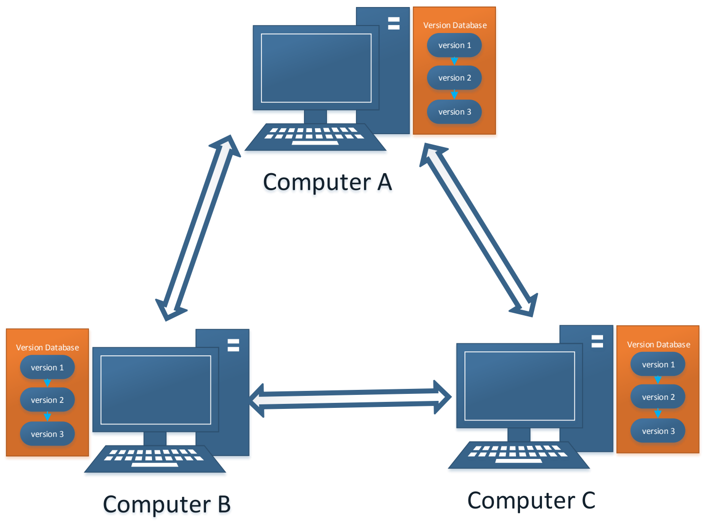


1. 完整镜像，clone，backup

2. 自由提交

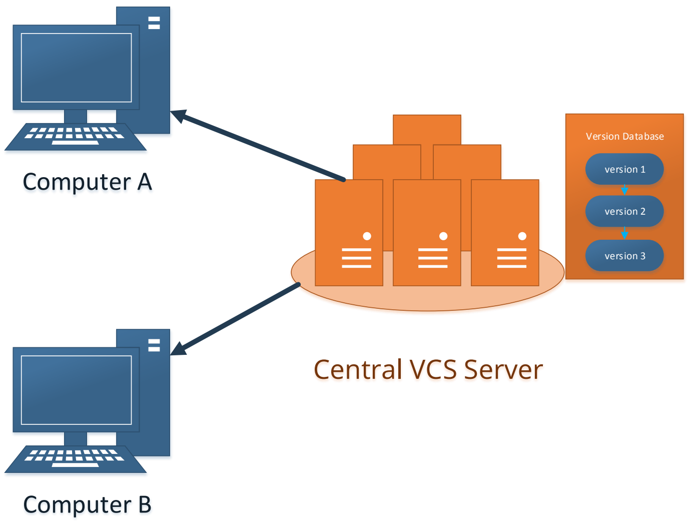
3. 层次开发

git



1. 完整镜像，clone，backup
2. 自由提交
3. 层次开发

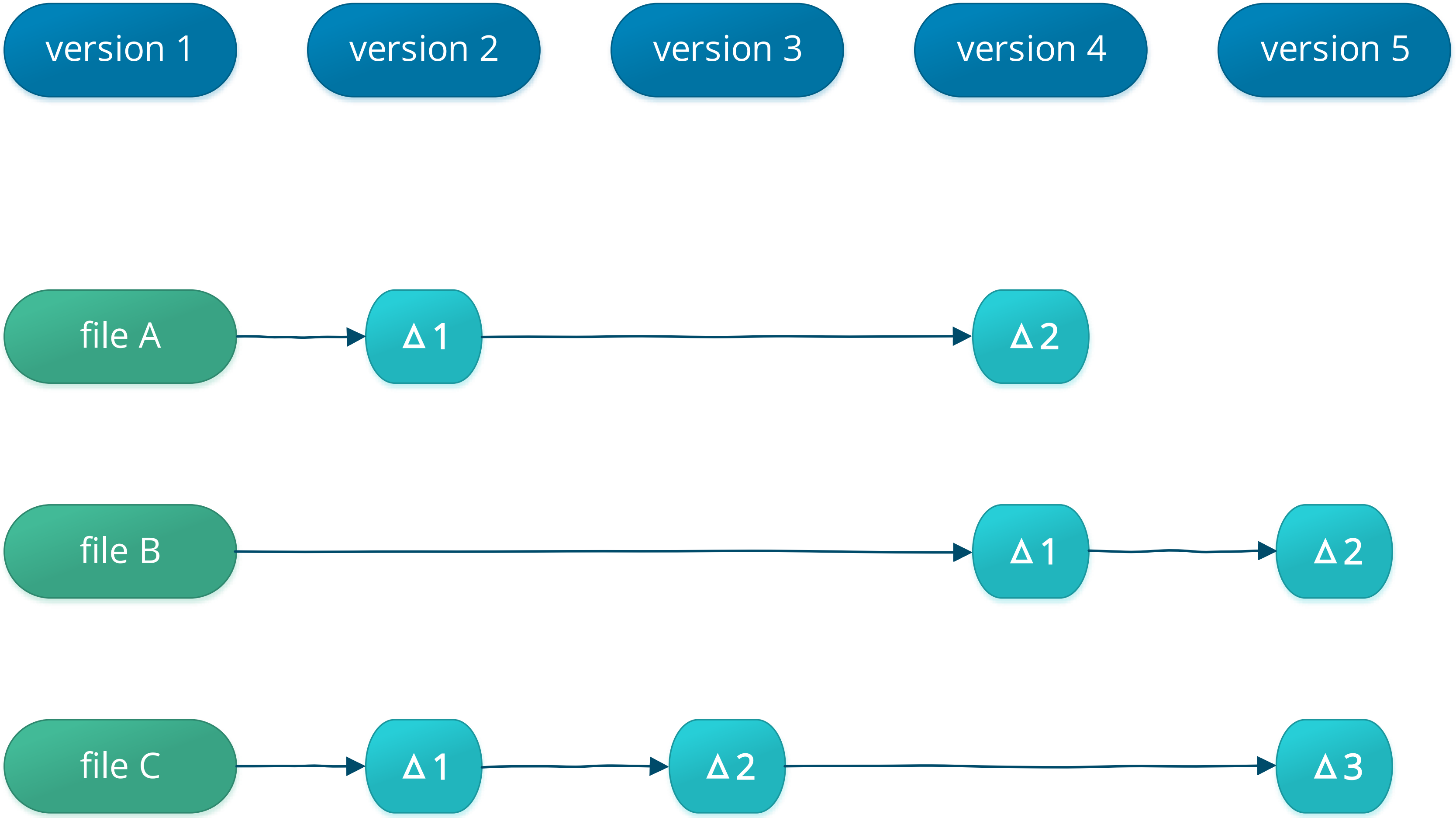
SUBVERSION



1. 进度报告
2. 权限控制，悲观锁
3. 维护简单

集中化的**版本控制**系统

Centralized **Version Control** System



version 1

version 2

version 3

version 4

version 5

file A

Δ 1

Δ 2

file B

Δ 1

Δ 2

file C

Δ 1

Δ 2

Δ 3

data as changes

README.txt

```
this is lisnb’s demo
to show how the
svn works

Licensed under GPL.
```

hello.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
    f.write(‘hello\n’)
```

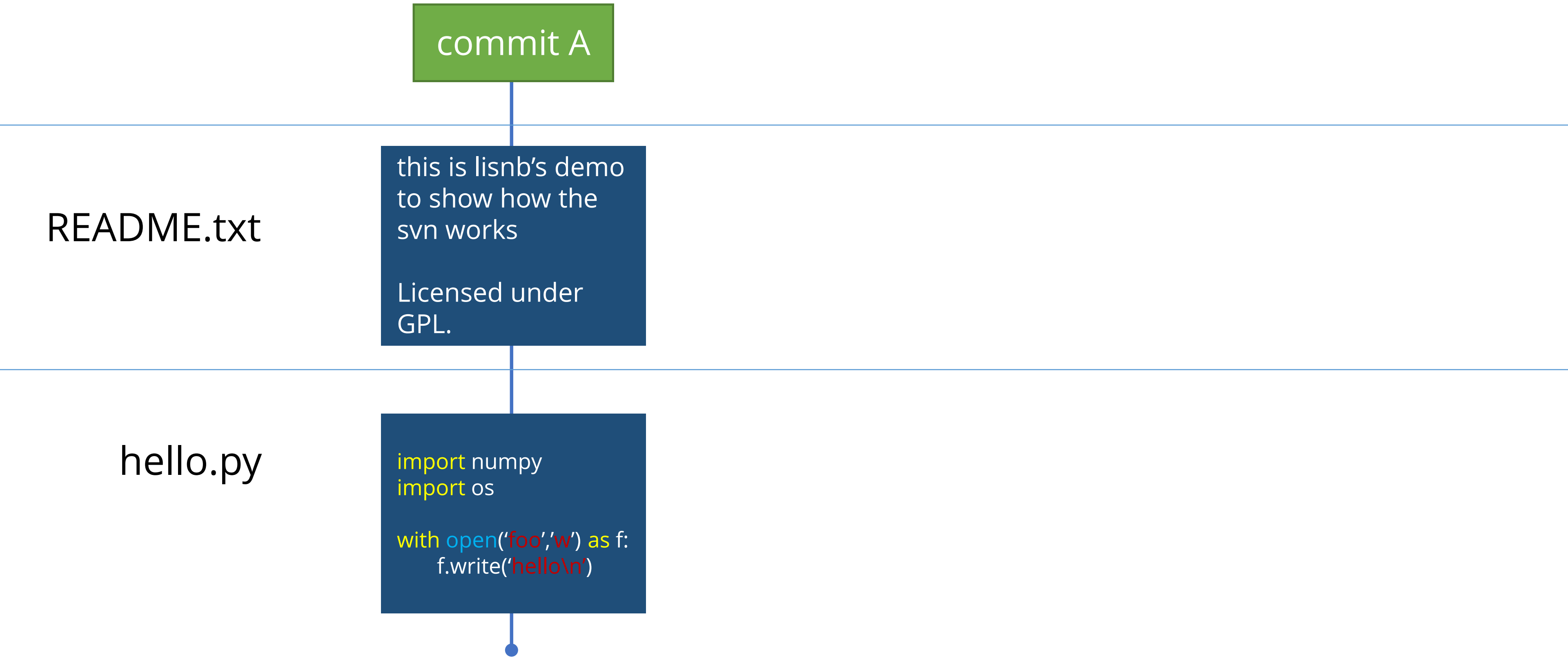
README.txt

this is lisnb’s demo
to show how the
svn works

Licensed under GPL.

hello.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
    f.write(‘hello\n’)
```



commit A

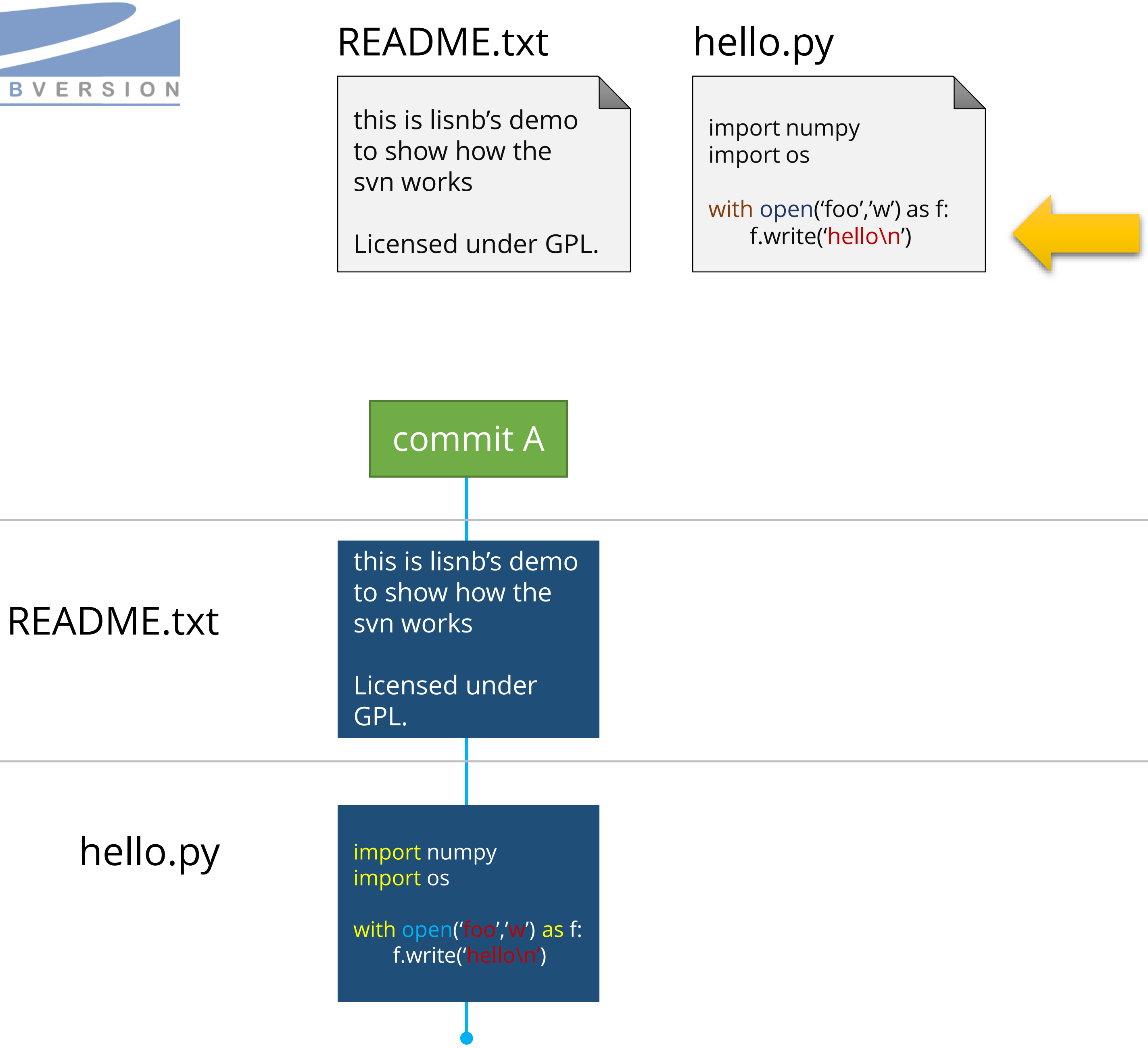
README.txt

this is lisnb’s demo
to show how the
svn works

Licensed under
GPL.

hello.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
    f.write(‘hello\n’)
```



README.txt

this is lisnb's demo
to show how the
svn works

Licensed under GPL.

hello.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('hello\n')
```

commit A

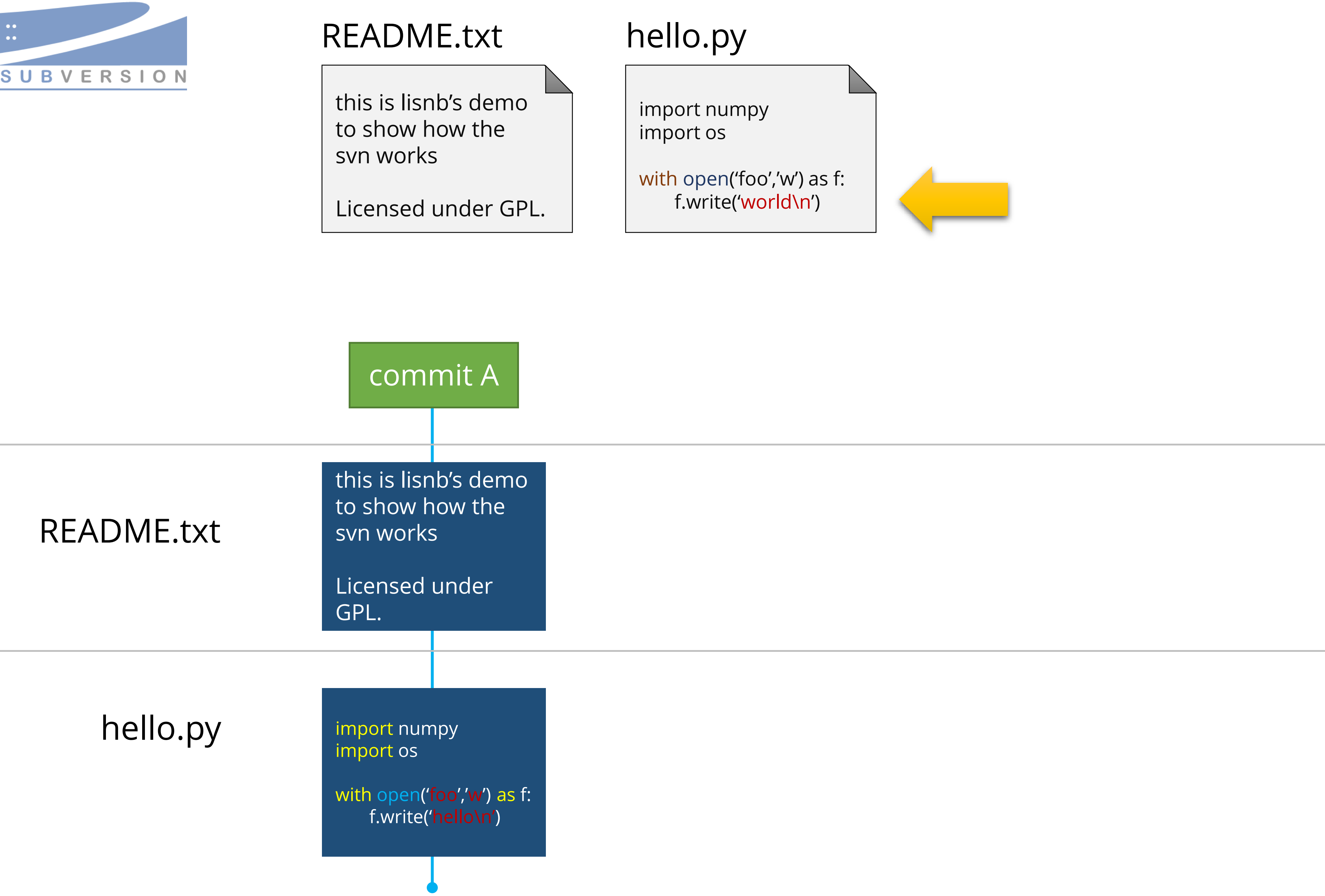
README.txt

this is lisnb's demo
to show how the
svn works

Licensed under
GPL.

hello.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('hello\n')
```



README.txt

```
this is lisnb’s demo
to show how the
svn works

Licensed under GPL.
```

hello.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
     f.write(‘world\n’)
```

commit A

README.txt

```
this is lisnb’s demo
to show how the
svn works

Licensed under
GPL.
```

hello.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
     f.write(‘hello\n’)
```

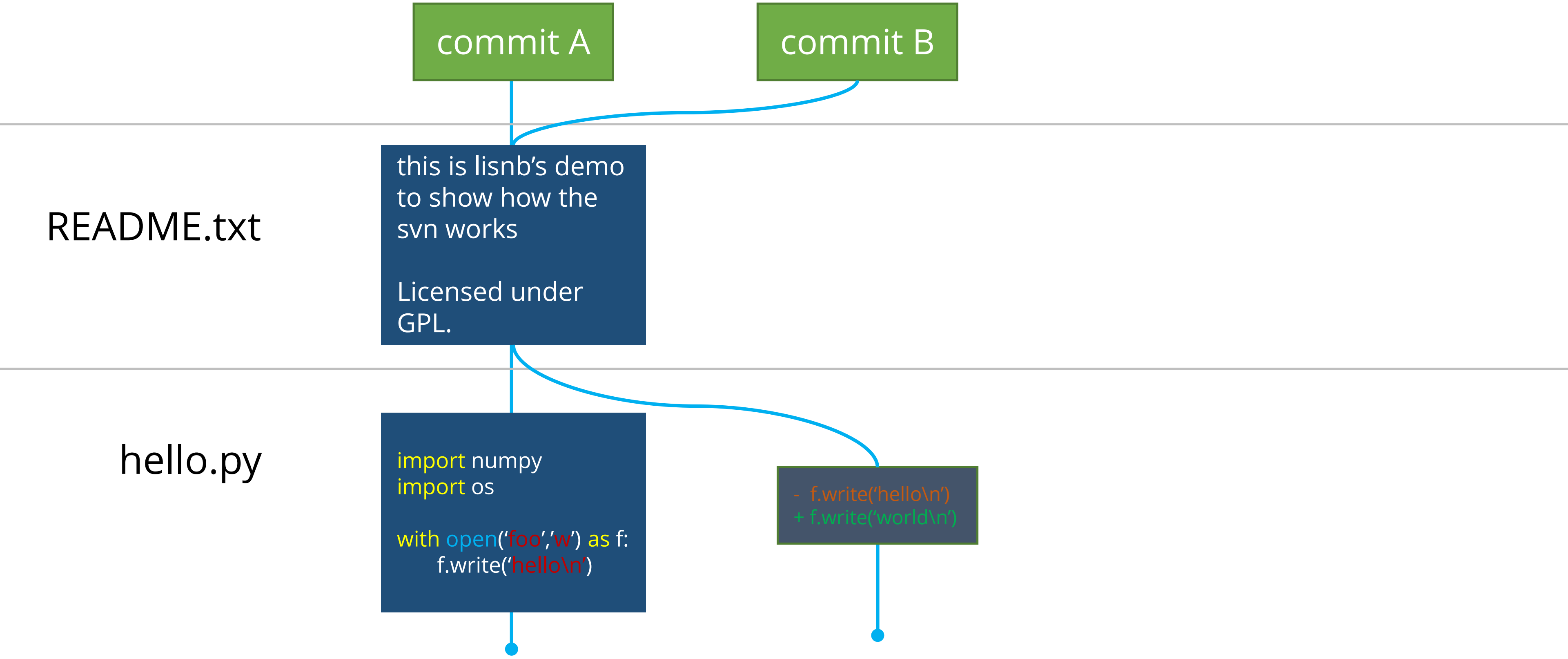
README.txt

this is lisnb’s demo
to show how the
svn works

Licensed under GPL.

hello.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
     f.write(‘world\n’)
```



commit A

commit B

README.txt

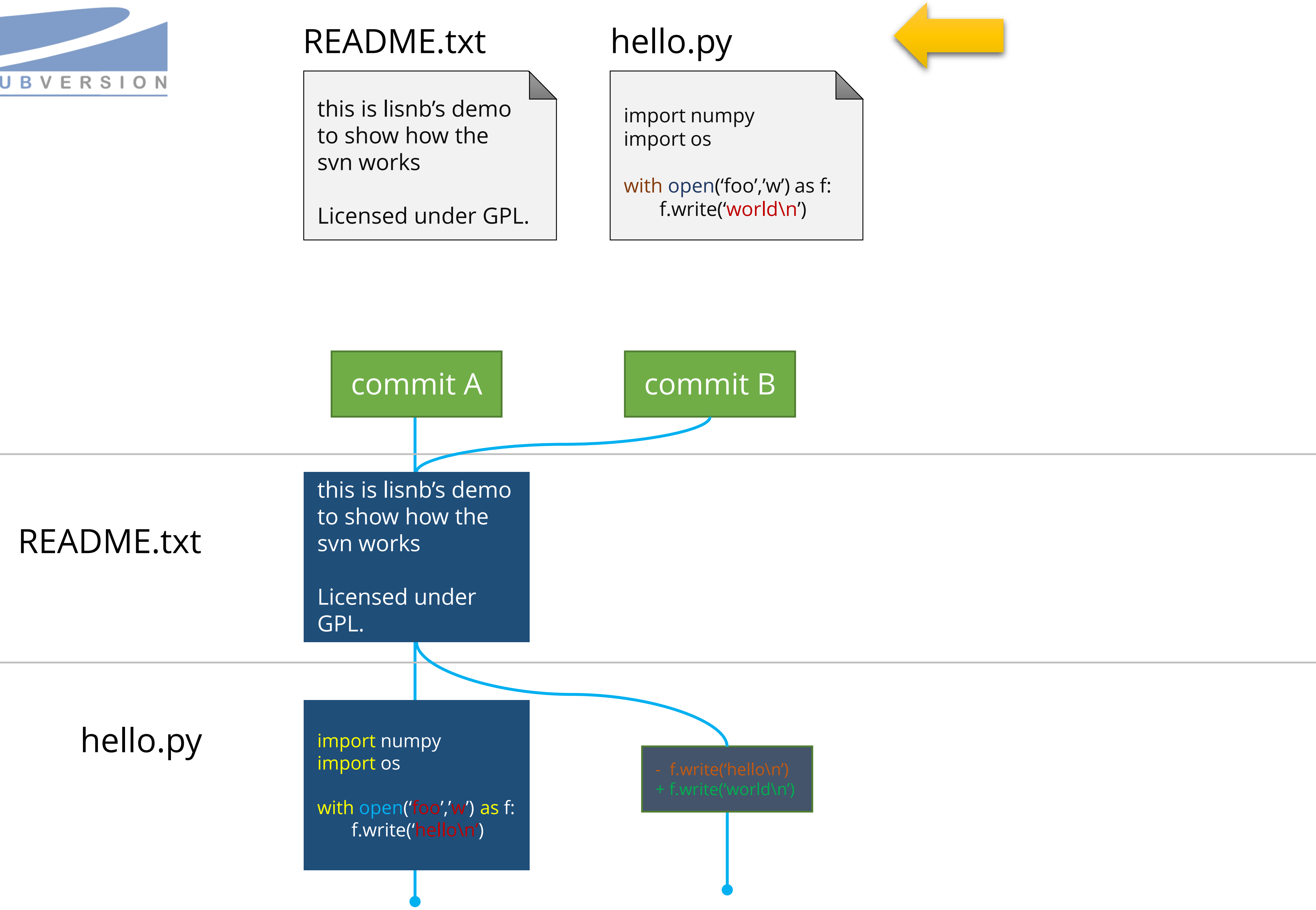
this is lisnb’s demo
to show how the
svn works

Licensed under
GPL.

hello.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
     f.write(‘hello\n’)
```

```
-  f.write(‘hello\n’)
+ f.write(‘world\n’)
```



README.txt

this is lisnb’s demo to show how the svn works

Licensed under GPL.

hello.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
    f.write(‘world\n’)
```

commit A

commit B

README.txt

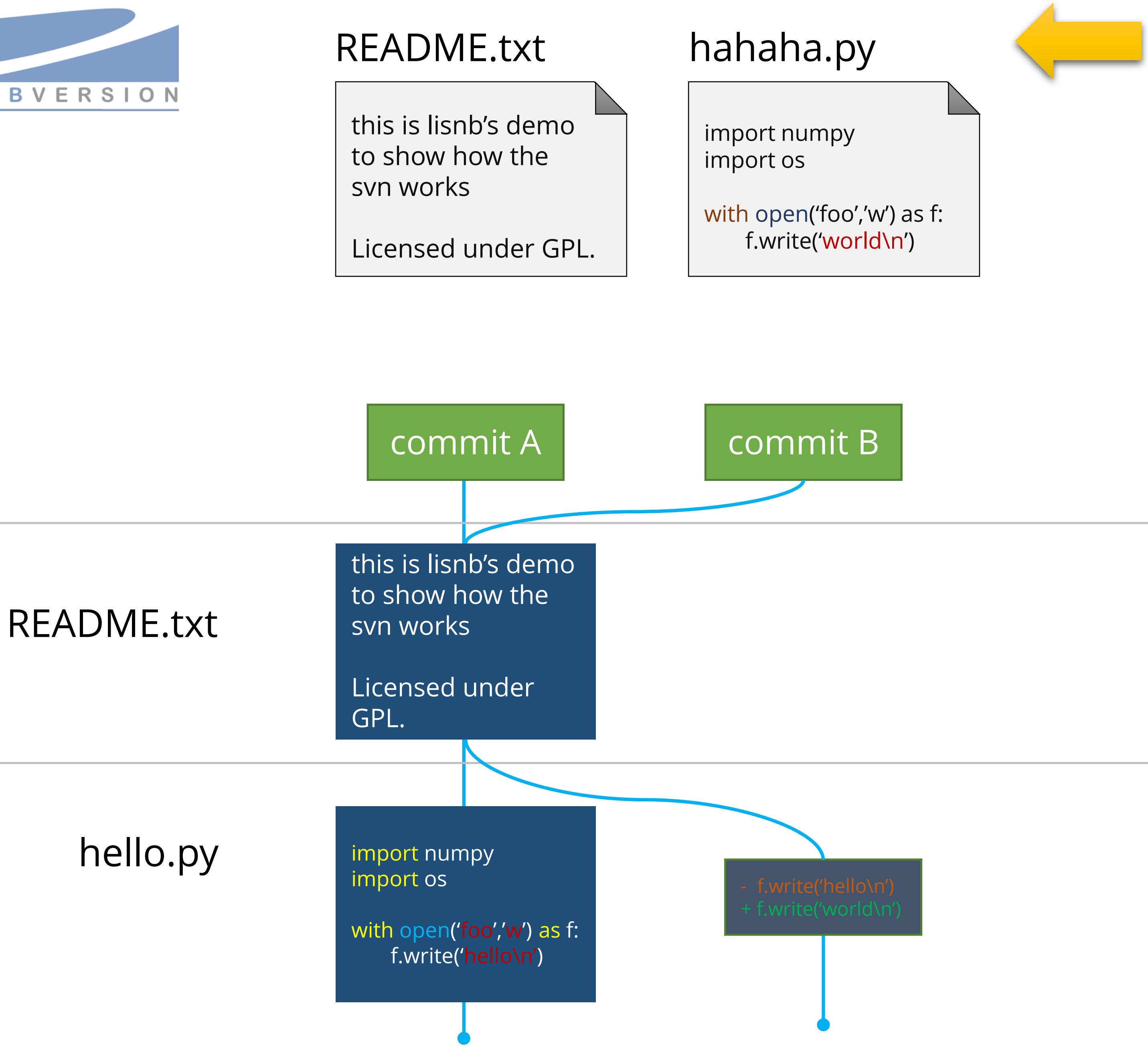
this is lisnb’s demo to show how the svn works

Licensed under GPL.

hello.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
    f.write(‘hello\n’)
```

```
-  f.write(‘hello\n’)
+ f.write(‘world\n’)
```



README.txt

```
this is lisnb’s demo
to show how the
svn works

Licensed under GPL.
```

hahaha.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
      f.write(‘world\n’)
```

commit A

commit B

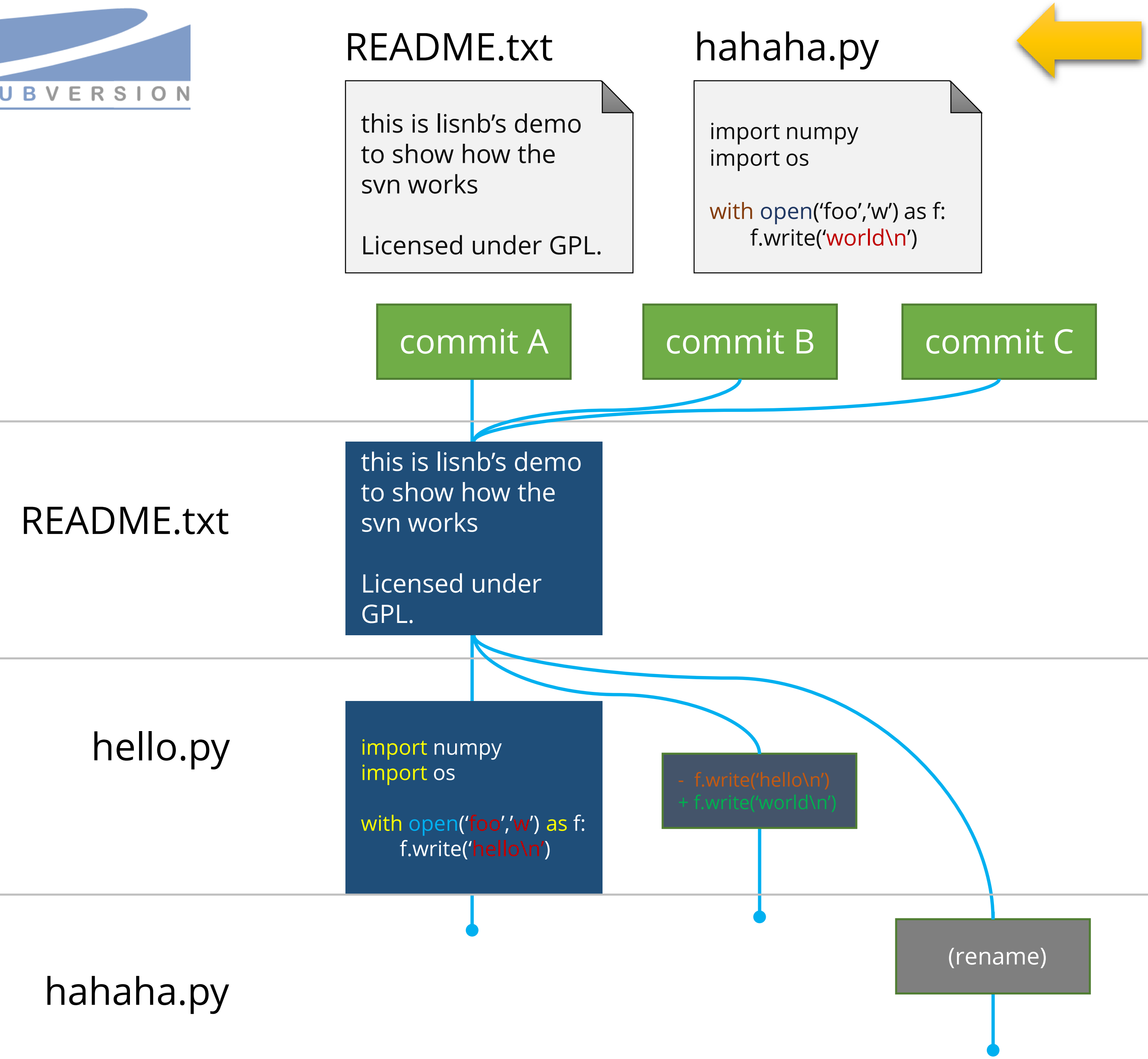
README.txt

```
this is lisnb’s demo
to show how the
svn works

Licensed under
GPL.
```

hello.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
      f.write(‘hello\n’)
```

```
-  f.write(‘hello\n’)
+ f.write(‘world\n’)
```



README.txt

```
this is lisnb's demo
to show how the
svn works

Licensed under GPL.
```

hahaha.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('world\n')
```

commit A

commit B

commit C

README.txt

```
this is lisnb's demo
to show how the
svn works

Licensed under
GPL.
```

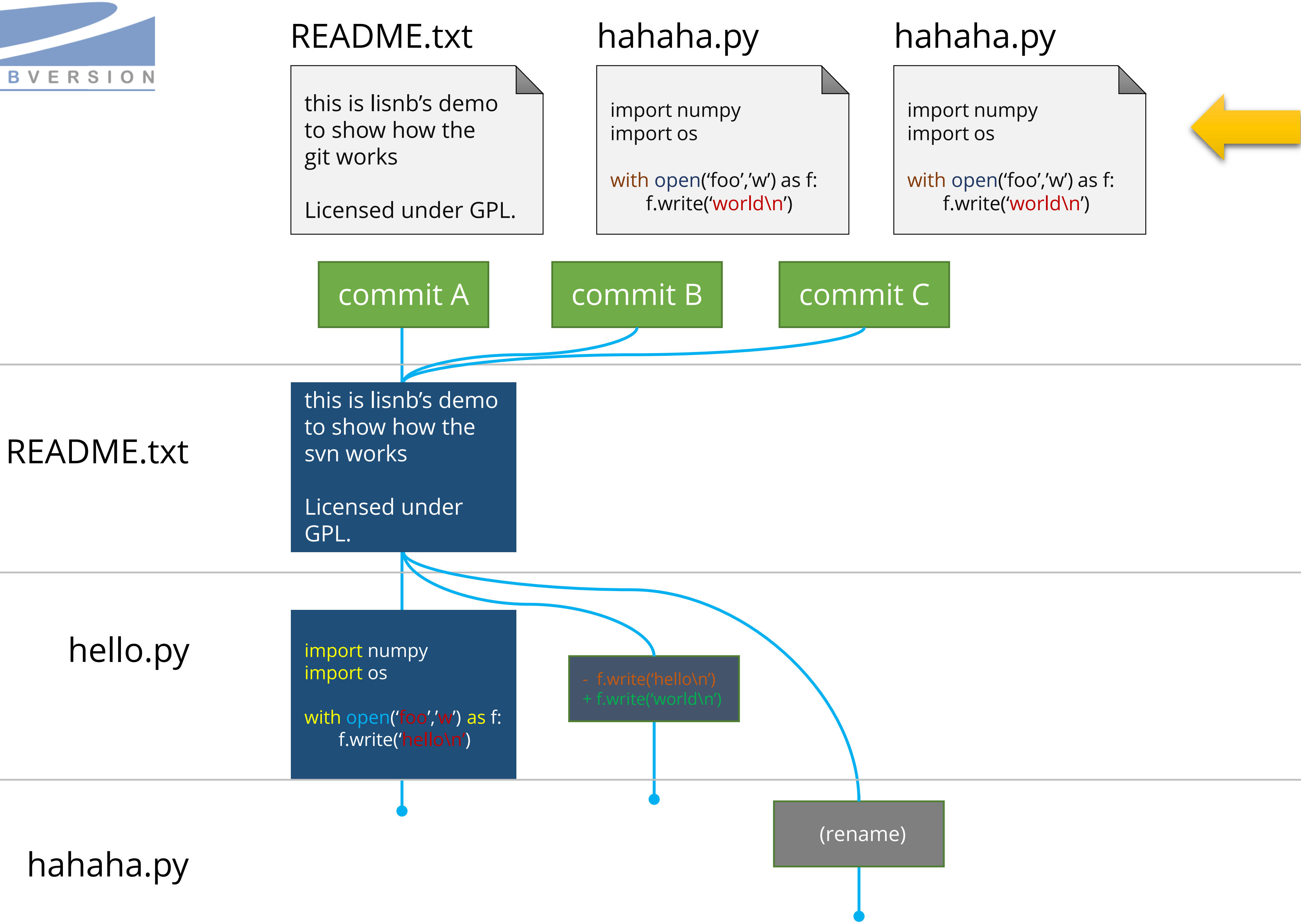
hello.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('hello\n')
```

```
-  f.write('hello\n')
+ f.write('world\n')
```

(rename)

hahaha.py



README.txt

```
this is lisnb's demo
to show how the
git works

Licensed under GPL.
```

hahaha.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('world\n')
```

hahaha.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('world\n')
```

commit A

commit B

commit C

README.txt

```
this is lisnb's demo
to show how the
svn works

Licensed under
GPL.
```

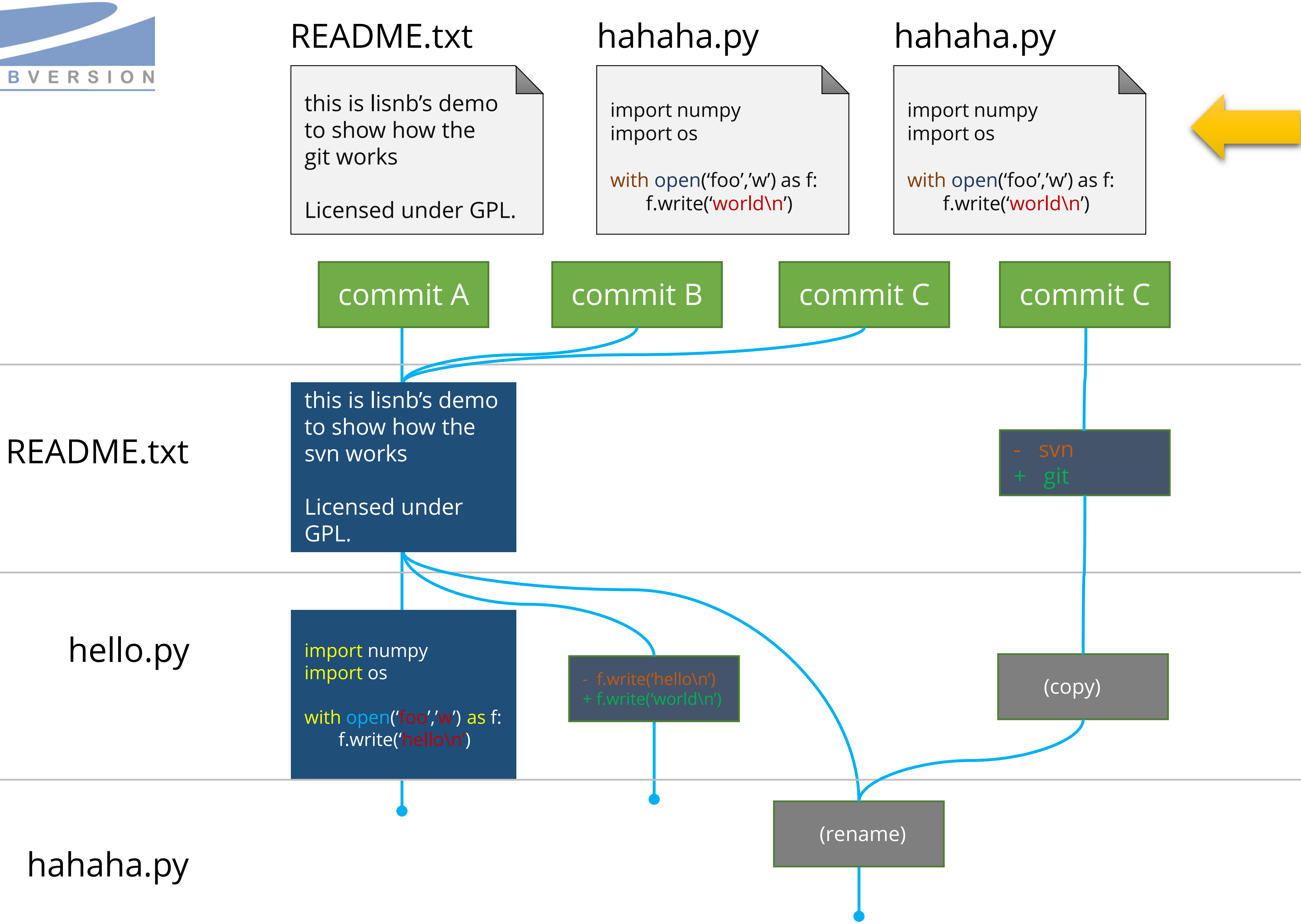
hello.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('hello\n')
```

```
- f.write('hello\n')
+ f.write('world\n')
```

(rename)

hahaha.py



README.txt

```
this is lisnb's demo
to show how the
git works

Licensed under GPL.
```

hahaha.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('world\n')
```

hahaha.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('world\n')
```

```
this is lisnb's demo
to show how the
svn works

Licensed under
GPL.
```

```
- svn
+ git
```

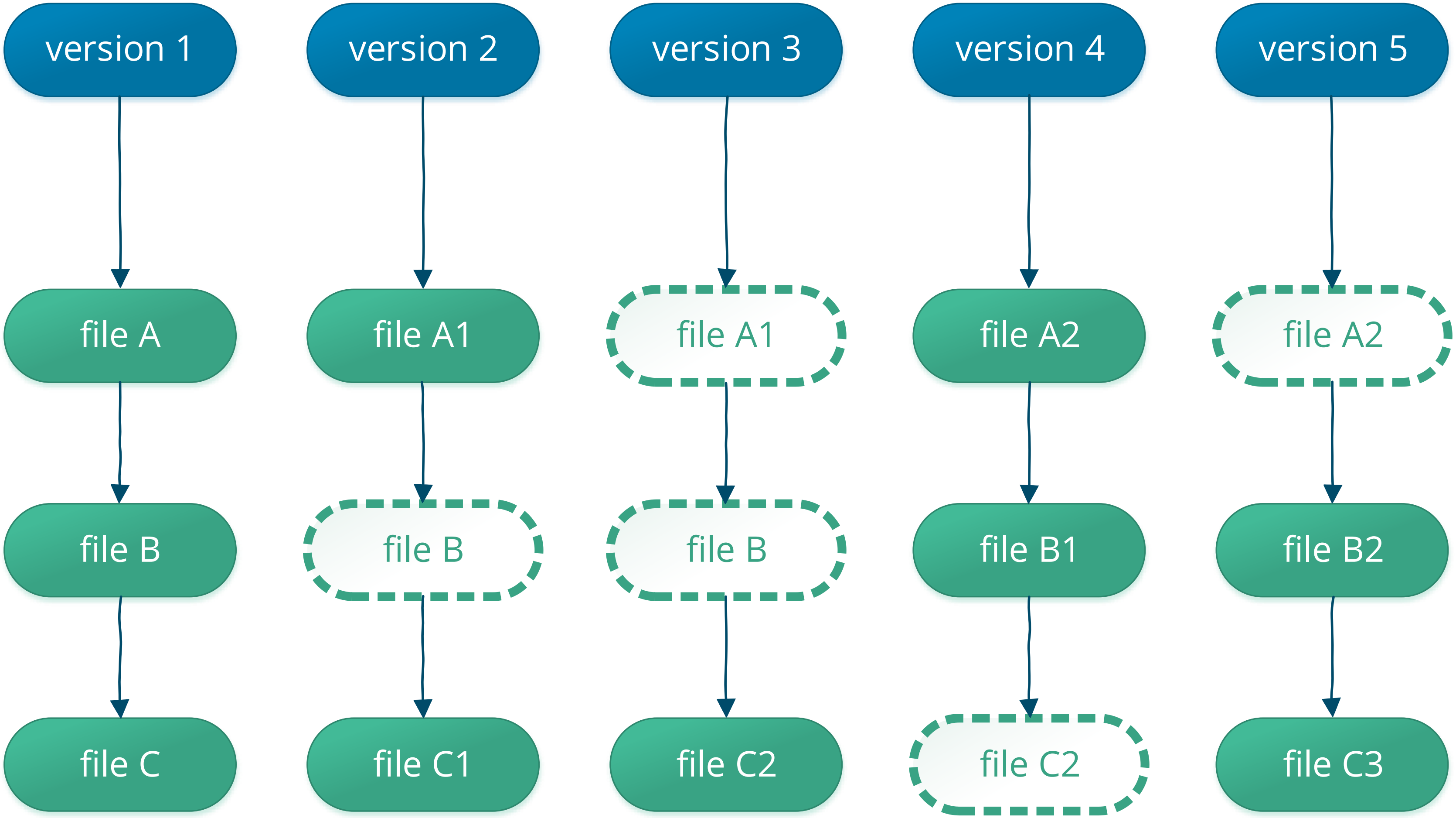
```
import numpy
import os

with open('foo','w') as f:
    f.write('hello\n')
```

```
- f.write('hello\n')
+ f.write('world\n')
```

git

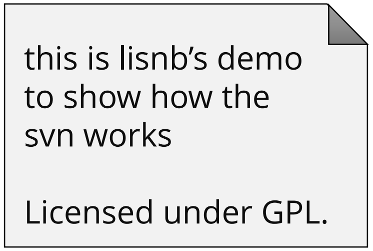
分布式的版本控制系统

Distributed Version Control System



data as snapshots

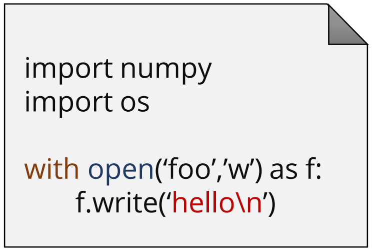
README.txt



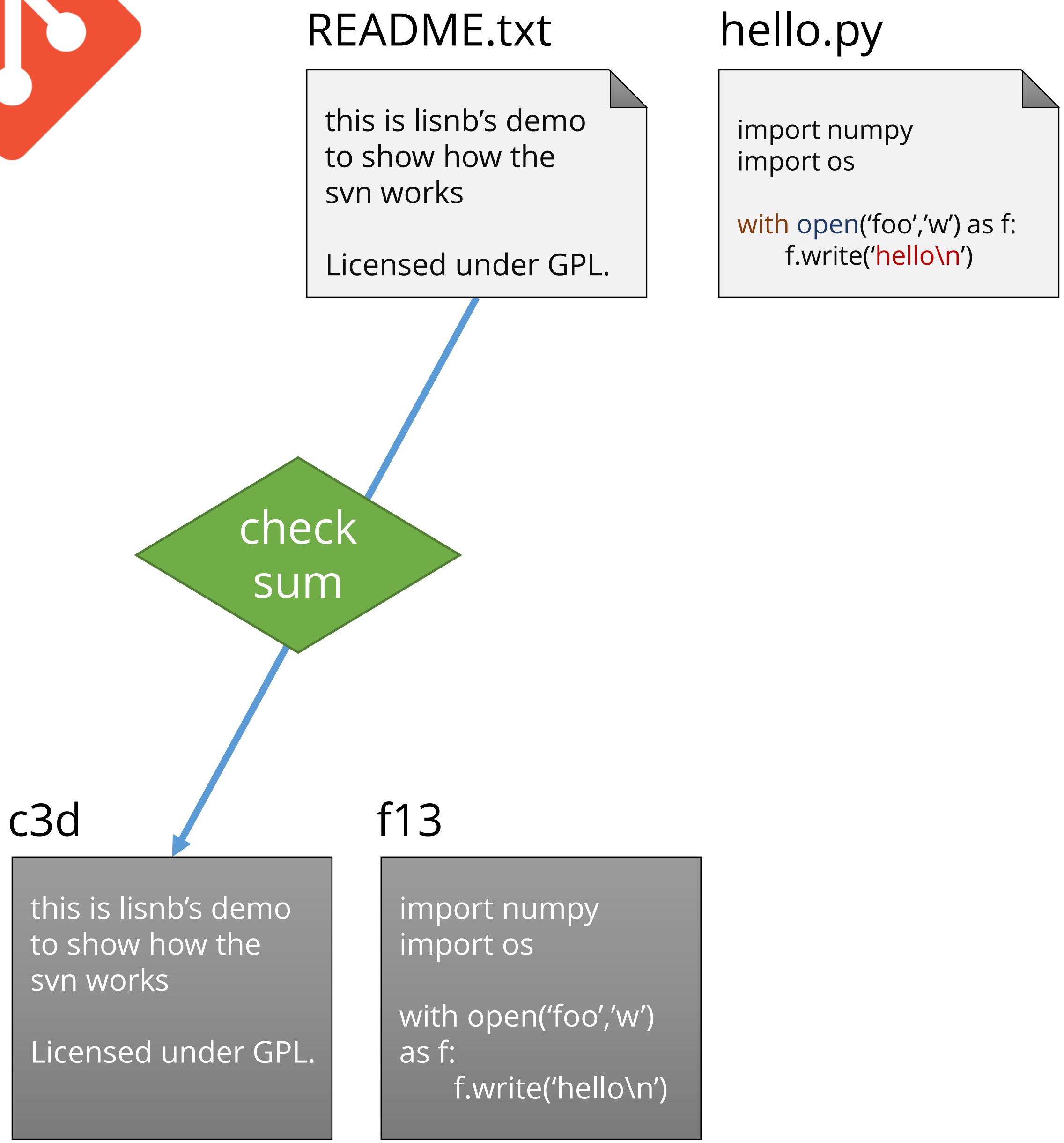
this is lisnb's demo
to show how the
svn works

Licensed under GPL.

hello.py



```
import numpy
import os

with open('foo','w') as f:
     f.write('hello\n')
```



README.txt

this is lisnb’s demo
to show how the
svn works

Licensed under GPL.

hello.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
      f.write(‘hello\n’)
```

check
sum

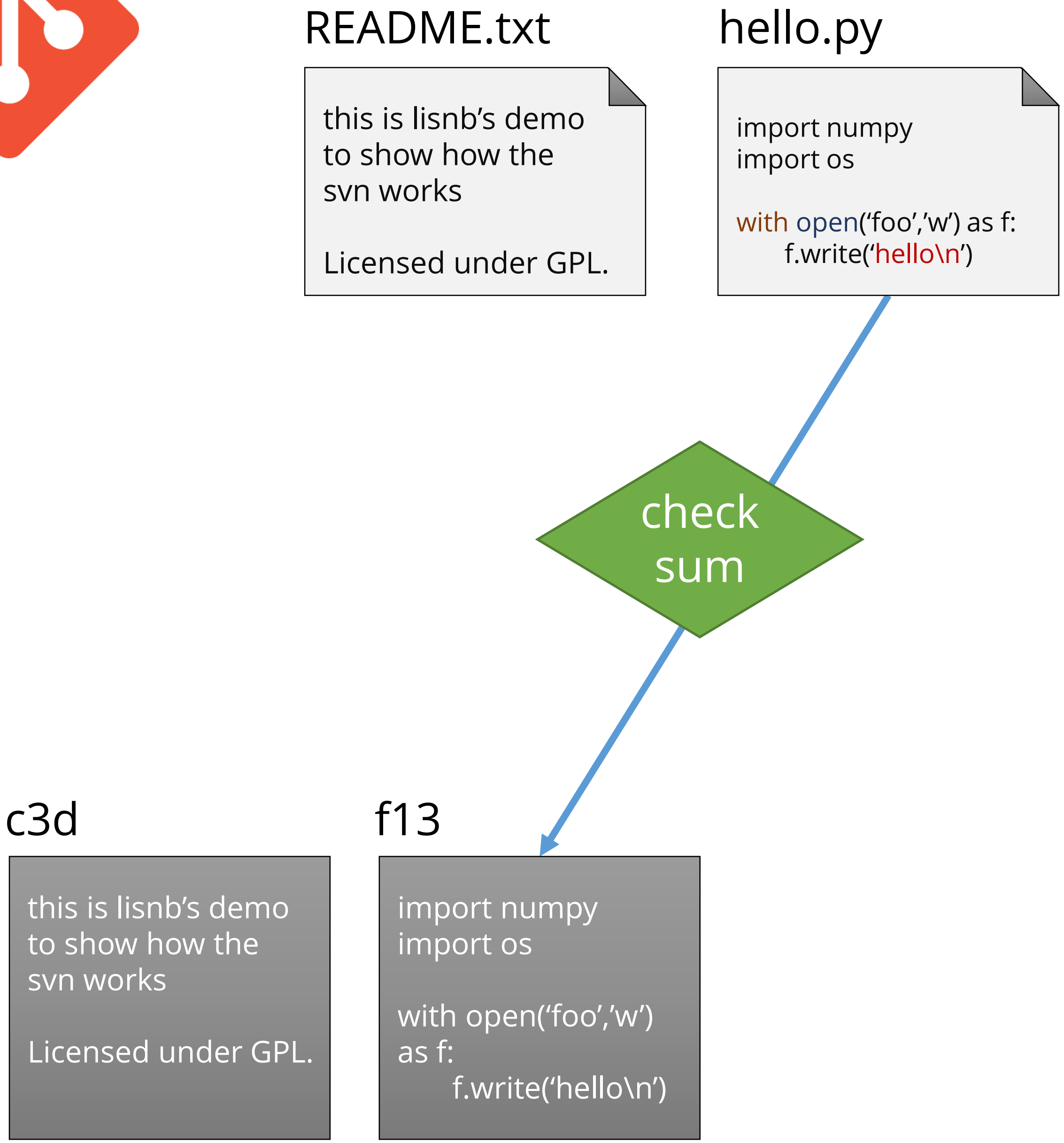
c3d

this is lisnb’s demo
to show how the
svn works

Licensed under GPL.

f13

```
import numpy
import os

with open(‘foo’,’w’)
as f:
      f.write(‘hello\n’)
```



README.txt

this is lisnb’s demo
to show how the
svn works

Licensed under GPL.

hello.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
     f.write(‘hello\n’)
```

check sum

c3d

this is lisnb’s demo
to show how the
svn works

Licensed under GPL.

f13

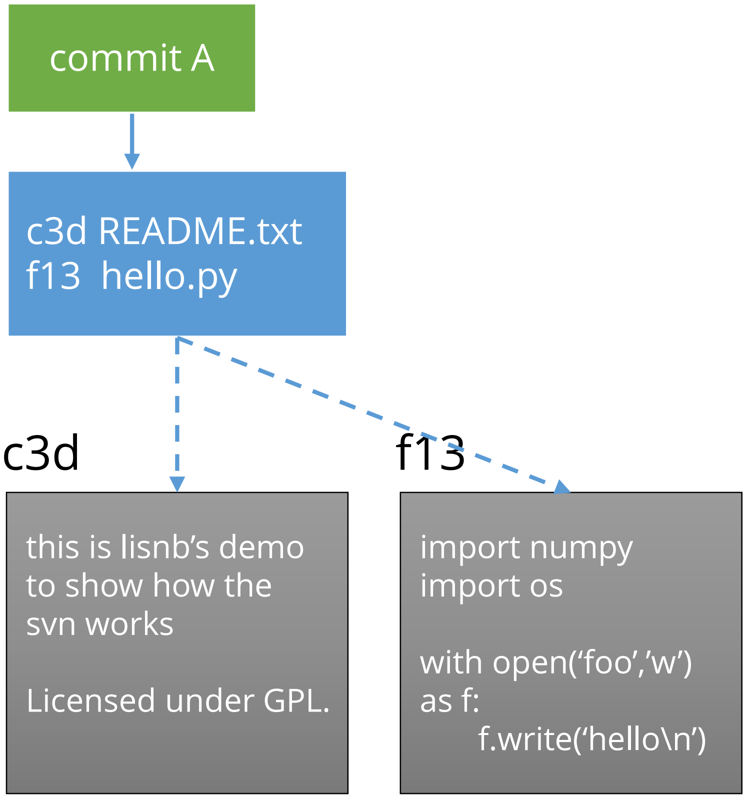
```
import numpy
import os

with open(‘foo’,’w’)
as f:
     f.write(‘hello\n’)
```

README.txt

```
this is lisnb’s demo
to show how the
svn works

Licensed under GPL.
```

hello.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
     f.write(‘hello\n’)
```



commit A

c3d README.txt
f13 hello.py

c3d

```
this is lisnb’s demo
to show how the
svn works

Licensed under GPL.
```

f13

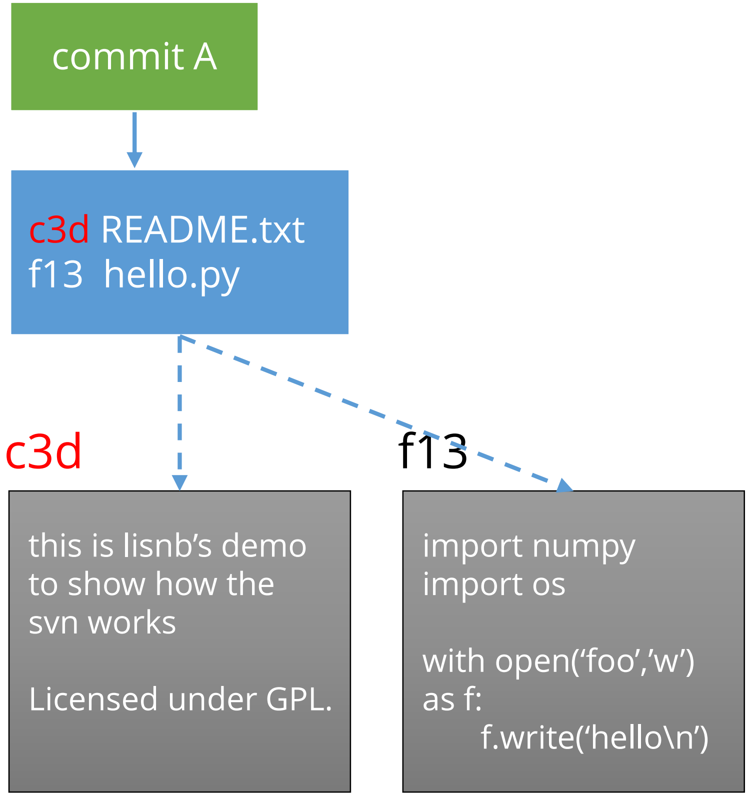
```
import numpy
import os

with open(‘foo’,’w’)
as f:
     f.write(‘hello\n’)
```

README.txt

```
this is lisnb’s demo
to show how the
svn works

Licensed under GPL.
```

hello.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
     f.write(‘hello\n’)
```



commit A

c3d README.txt
f13 hello.py

c3d

```
this is lisnb’s demo
to show how the
svn works

Licensed under GPL.
```

f13

```
import numpy
import os

with open(‘foo’,’w’)
as f:
     f.write(‘hello\n’)
```

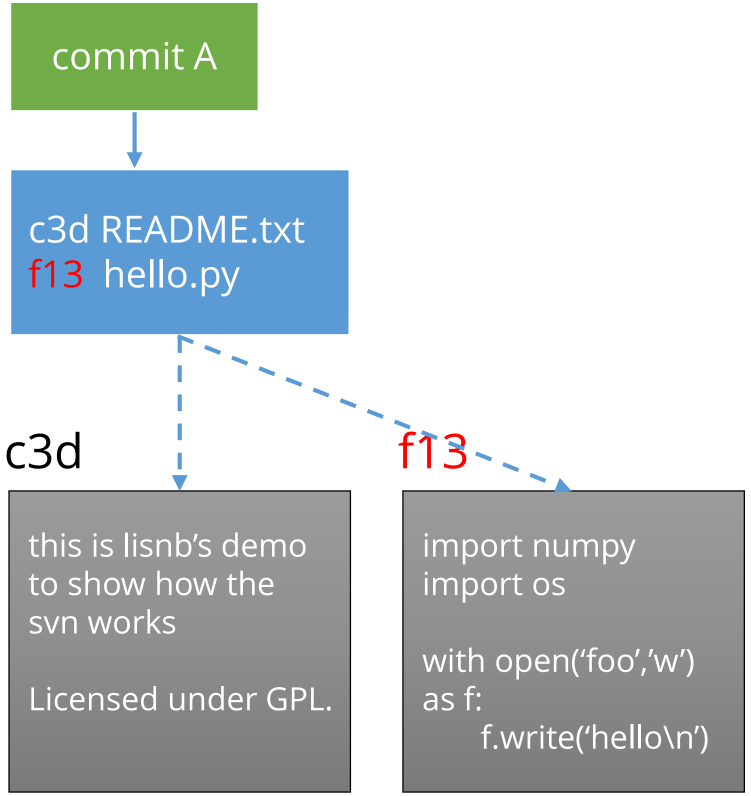
README.txt

this is lisnb's demo
to show how the
svn works

Licensed under GPL.

hello.py

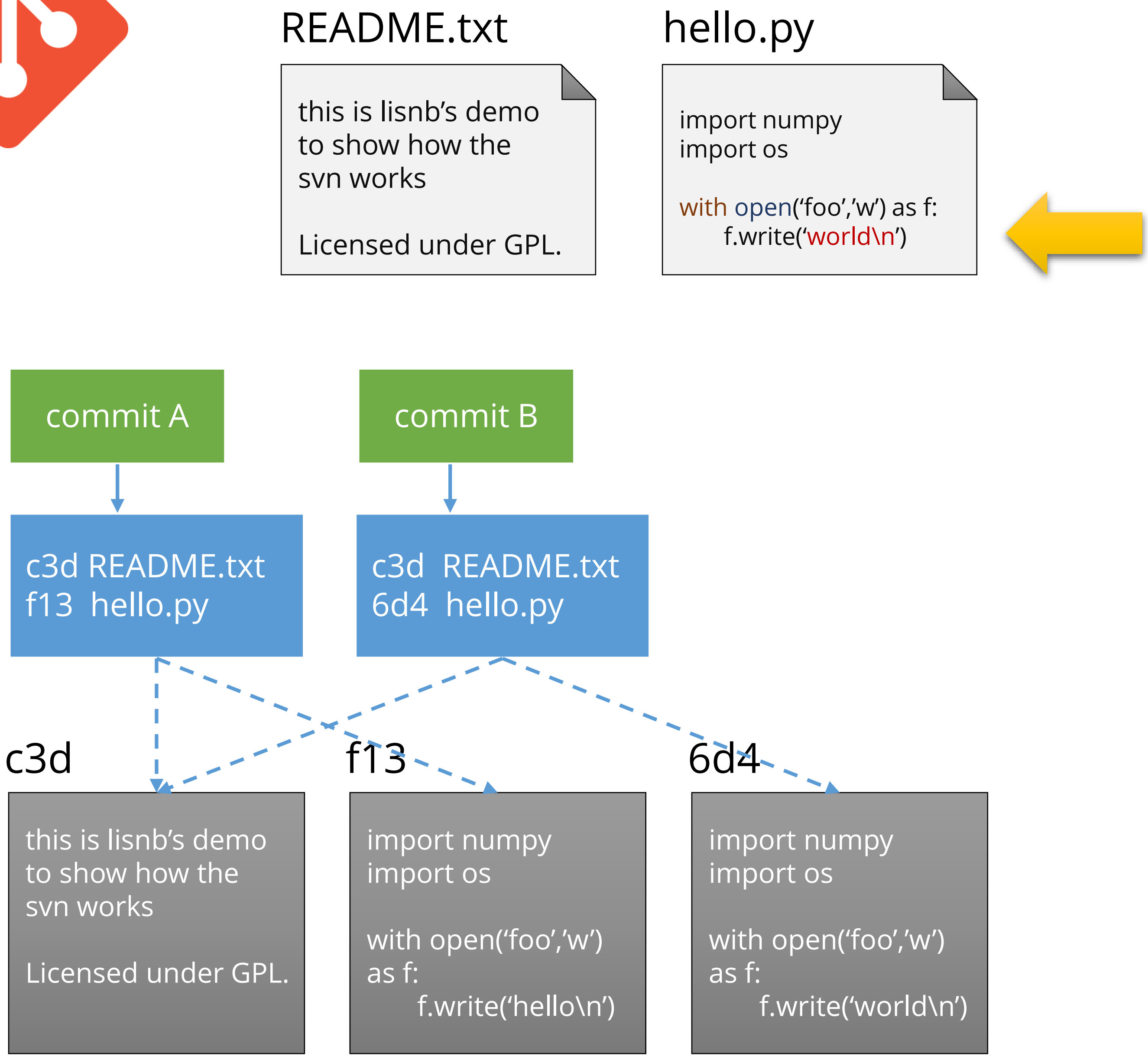
```
import numpy
import os

with open('foo','w') as f:
    f.write('hello\n')
```



this is lisnb's demo
to show how the
svn works

Licensed under GPL.

```
import numpy
import os

with open('foo','w')
as f:
    f.write('hello\n')
```



README.txt

```
this is lisnb's demo
to show how the
svn works

Licensed under GPL.
```

hello.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('world\n')
```

```
c3d README.txt
f13  hello.py
```

```
c3d  README.txt
6d4  hello.py
```

c3d

```
this is lisnb's demo
to show how the
svn works

Licensed under GPL.
```

f13

```
import numpy
import os

with open('foo','w')
as f:
    f.write('hello\n')
```

6d4

```
import numpy
import os

with open('foo','w')
as f:
    f.write('world\n')
```

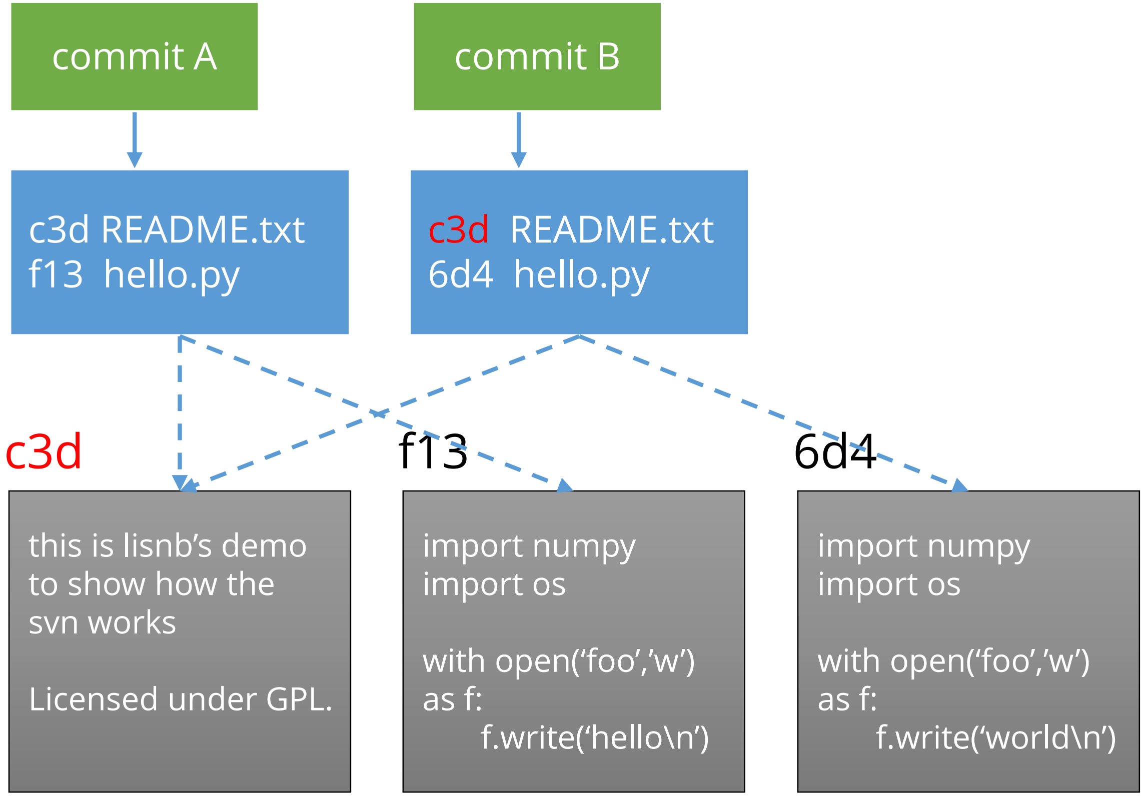
README.txt

this is lisnb's demo
to show how the
svn works

Licensed under GPL.

hello.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('world\n')
```



commit A

commit B

c3d README.txt
f13 hello.py

c3d README.txt
6d4 hello.py

c3d

this is lisnb's demo
to show how the
svn works

Licensed under GPL.

f13

```
import numpy
import os

with open('foo','w')
as f:
     f.write('hello\n')
```

6d4

```
import numpy
import os

with open('foo','w')
as f:
     f.write('world\n')
```

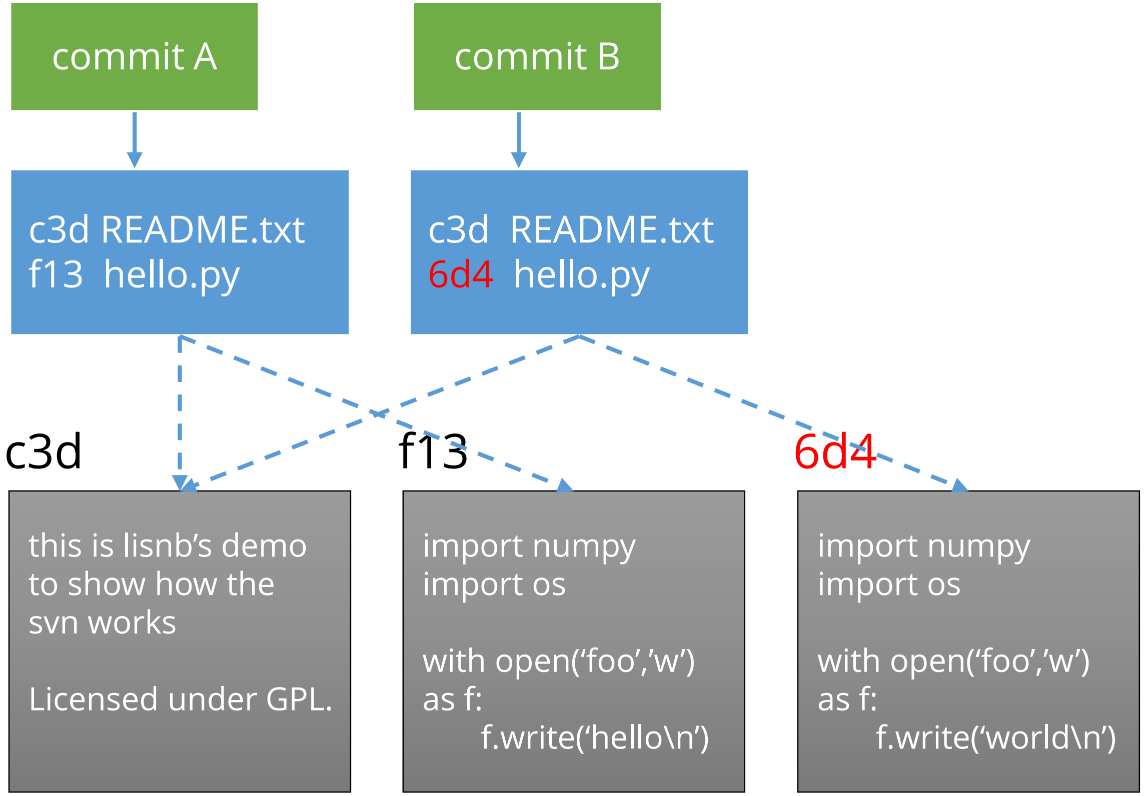
README.txt

this is lisnb's demo
to show how the
svn works

Licensed under GPL.

hello.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('world\n')
```



commit A

c3d README.txt
f13 hello.py

commit B

c3d README.txt
6d4 hello.py

c3d

this is lisnb's demo
to show how the
svn works

Licensed under GPL.

f13

```
import numpy
import os

with open('foo','w')
as f:
    f.write('hello\n')
```

6d4

```
import numpy
import os

with open('foo','w')
as f:
    f.write('world\n')
```

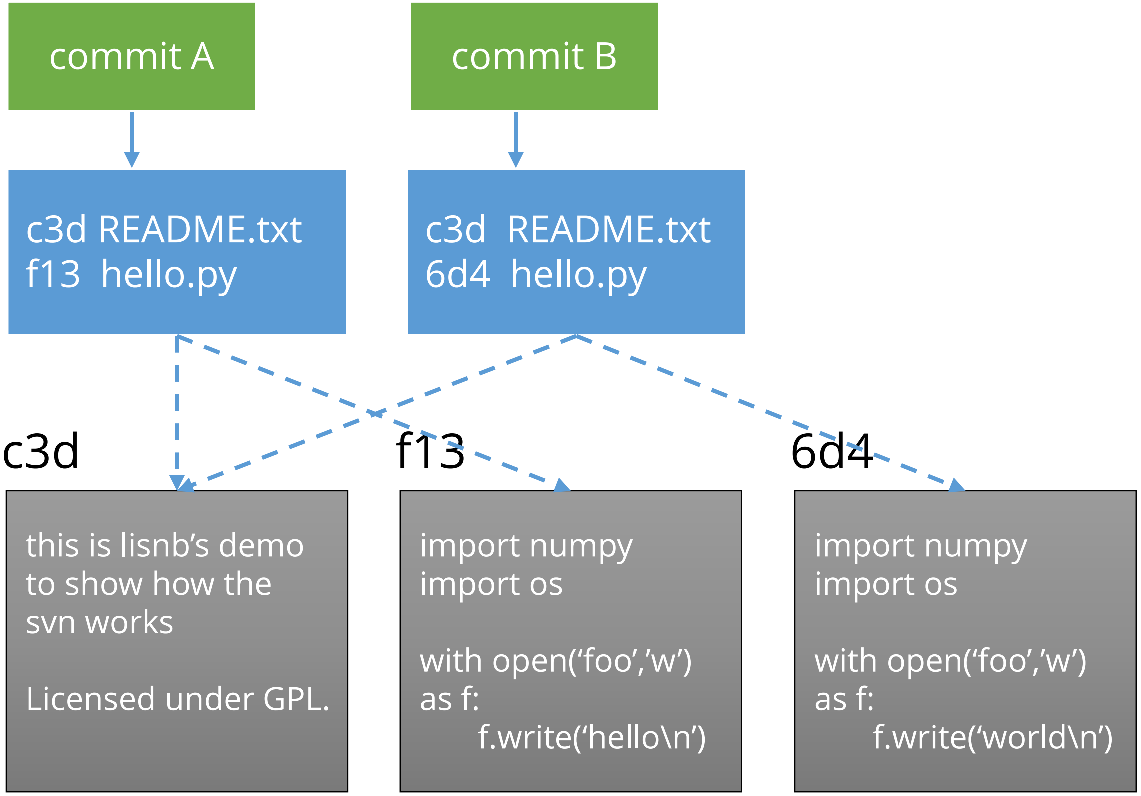
README.txt

this is lisnb’s demo
to show how the
svn works

Licensed under GPL.

hahaha.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
    f.write(‘world\n’)
```



commit A

c3d README.txt
f13 hello.py

commit B

c3d README.txt
6d4 hello.py

c3d

this is lisnb’s demo
to show how the
svn works

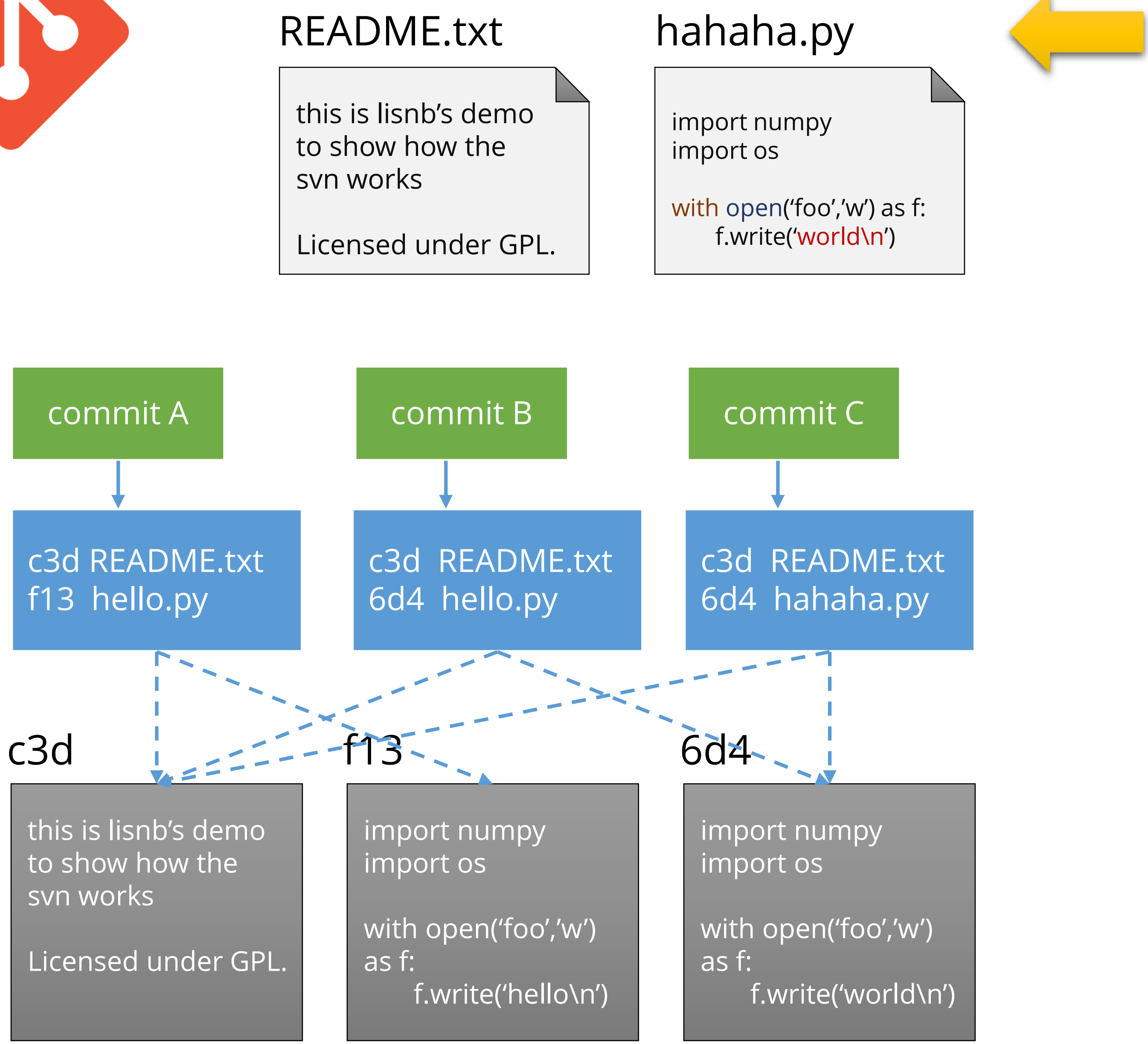
Licensed under GPL.

f13

```
import numpy
import os

with open(‘foo’,’w’)
as f:
    f.write(‘hello\n’)
```

6d4

```
import numpy
import os

with open(‘foo’,’w’)
as f:
    f.write(‘world\n’)
```



README.txt

this is lisnb’s demo
to show how the
svn works

Licensed under GPL.

hahaha.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
     f.write(‘world\n’)
```

commit A

commit B

commit C

c3d README.txt
f13 hello.py

c3d README.txt
6d4 hello.py

c3d README.txt
6d4 hahaha.py

c3d

this is lisnb’s demo
to show how the
svn works

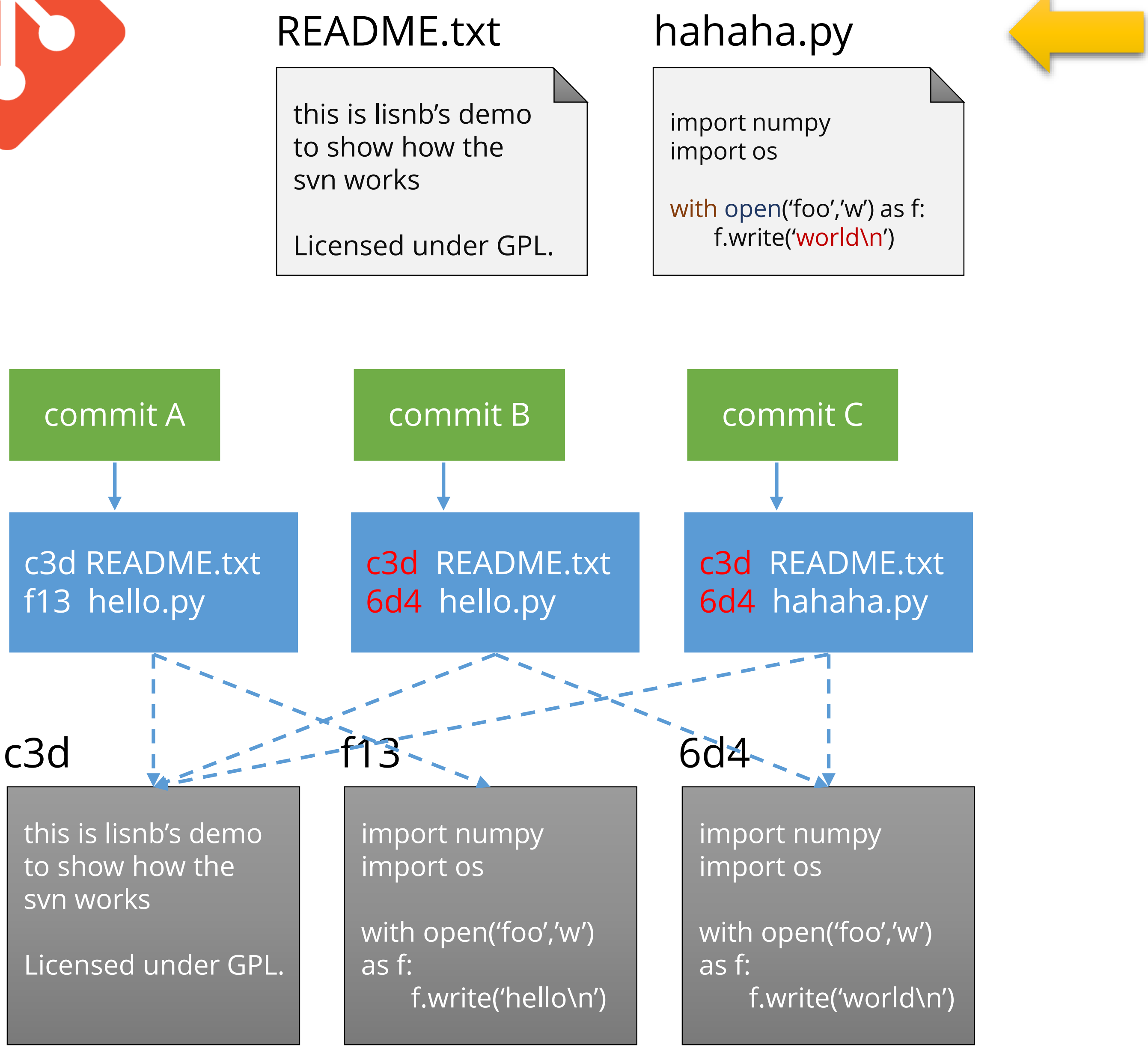
Licensed under GPL.

f13

```
import numpy
import os

with open(‘foo’,’w’)
as f:
     f.write(‘hello\n’)
```

6d4

```
import numpy
import os

with open(‘foo’,’w’)
as f:
     f.write(‘world\n’)
```



README.txt

this is lisnb's demo
to show how the
svn works

Licensed under GPL.

hahaha.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('world\n')
```

commit A

c3d README.txt
f13 hello.py

commit B

c3d README.txt
6d4 hello.py

commit C

c3d README.txt
6d4 hahaha.py

c3d

this is lisnb's demo
to show how the
svn works

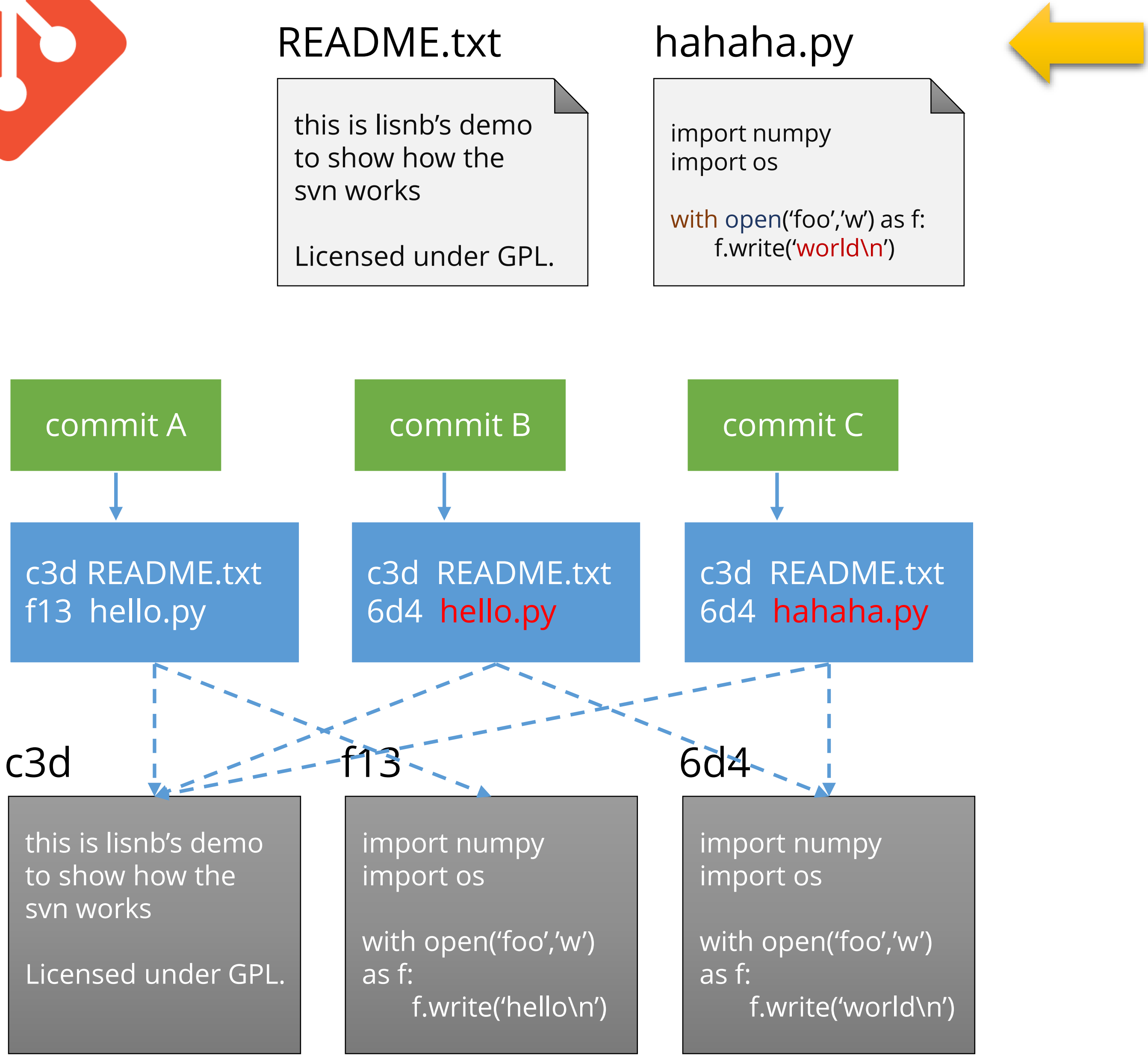
Licensed under GPL.

f13

```
import numpy
import os

with open('foo','w')
as f:
    f.write('hello\n')
```

6d4

```
import numpy
import os

with open('foo','w')
as f:
    f.write('world\n')
```



README.txt

```
this is lisnb’s demo
to show how the
svn works

Licensed under GPL.
```

hahaha.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
    f.write(‘world\n’)
```

commit A

```
c3d README.txt
f13  hello.py
```

commit B

```
c3d  README.txt
6d4  hello.py
```

commit C

```
c3d  README.txt
6d4  hahaha.py
```

c3d

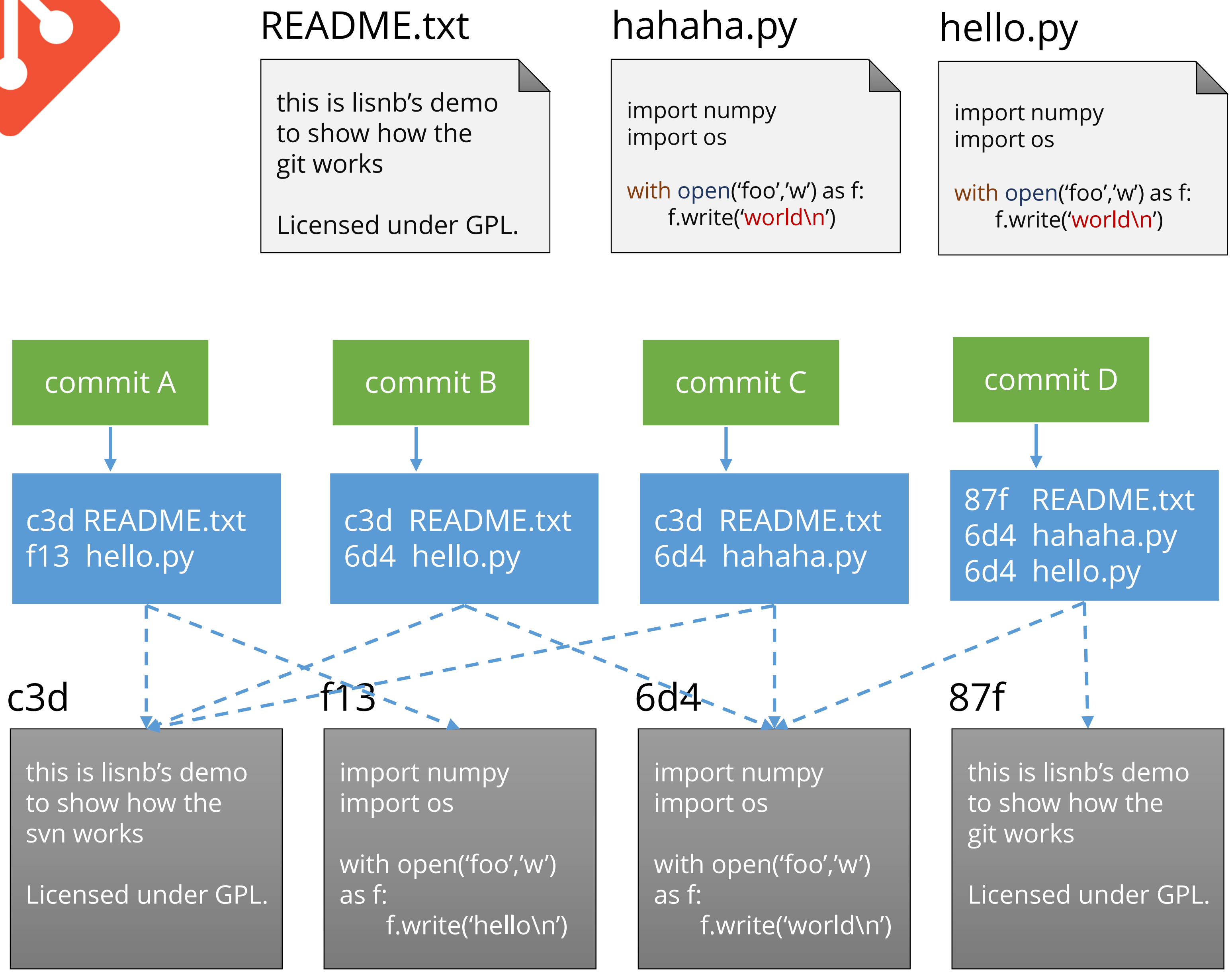
```
this is lisnb’s demo
to show how the
svn works

Licensed under GPL.
```

f13

```
import numpy
import os

with open(‘foo’,’w’)
as f:
    f.write(‘hello\n’)
```

6d4

```
import numpy
import os

with open(‘foo’,’w’)
as f:
    f.write(‘world\n’)
```



README.txt

```
this is lisnb's demo
to show how the
git works

Licensed under GPL.
```

hahaha.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('world\n')
```

hello.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('world\n')
```

commit A

c3d README.txt
f13 hello.py

commit B

c3d README.txt
6d4 hello.py

commit C

c3d README.txt
6d4 hahaha.py

commit D

87f README.txt
6d4 hahaha.py
6d4 hello.py

c3d

```
this is lisnb's demo
to show how the
svn works

Licensed under GPL.
```

f13

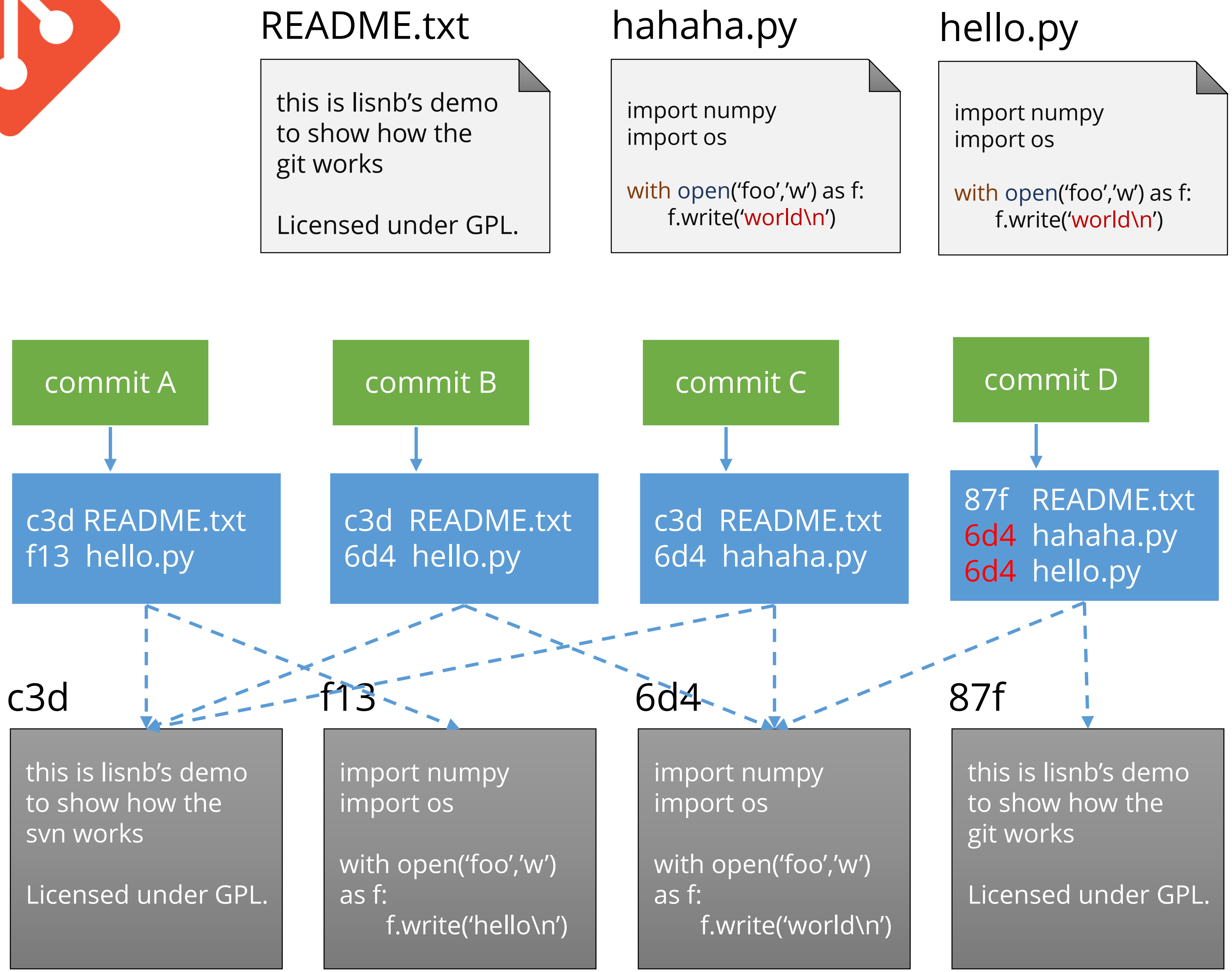
```
import numpy
import os

with open('foo','w')
as f:
    f.write('hello\n')
```

6d4

```
import numpy
import os

with open('foo','w')
as f:
    f.write('world\n')
```

87f

```
this is lisnb's demo
to show how the
git works

Licensed under GPL.
```



README.txt

```
this is lisnb's demo
to show how the
git works

Licensed under GPL.
```

hahaha.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('world\n')
```

hello.py

```
import numpy
import os

with open('foo','w') as f:
    f.write('world\n')
```

commit A

```
c3d README.txt
f13  hello.py
```

commit B

```
c3d  README.txt
6d4  hello.py
```

commit C

```
c3d  README.txt
6d4  hahaha.py
```

commit D

```
87f   README.txt
6d4  hahaha.py
6d4  hello.py
```

c3d

```
this is lisnb's demo
to show how the
svn works

Licensed under GPL.
```

f13

```
import numpy
import os

with open('foo','w')
as f:
      f.write('hello\n')
```

6d4

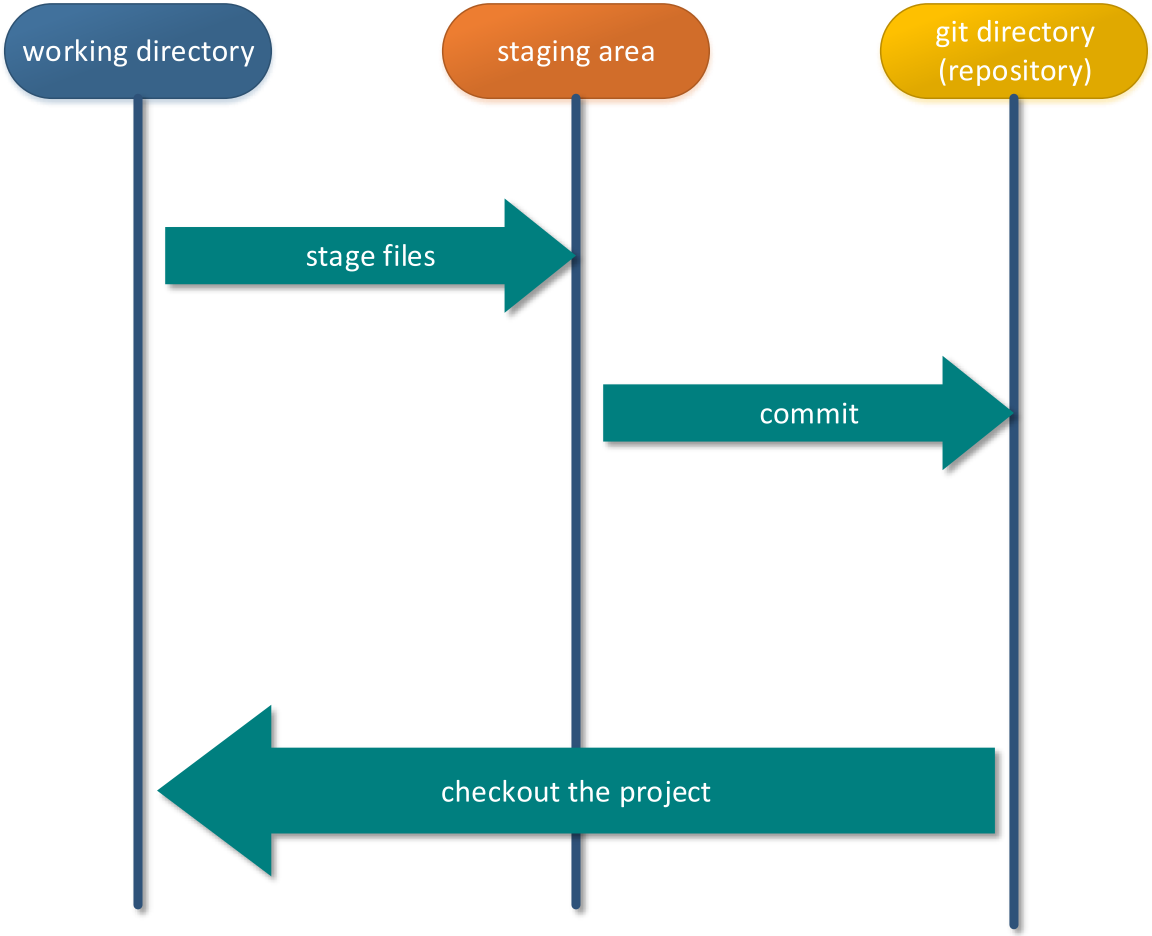
```
import numpy
import os

with open('foo','w')
as f:
      f.write('world\n')
```

87f

```
this is lisnb's demo
to show how the
git works

Licensed under GPL.
```

git 的使用实例

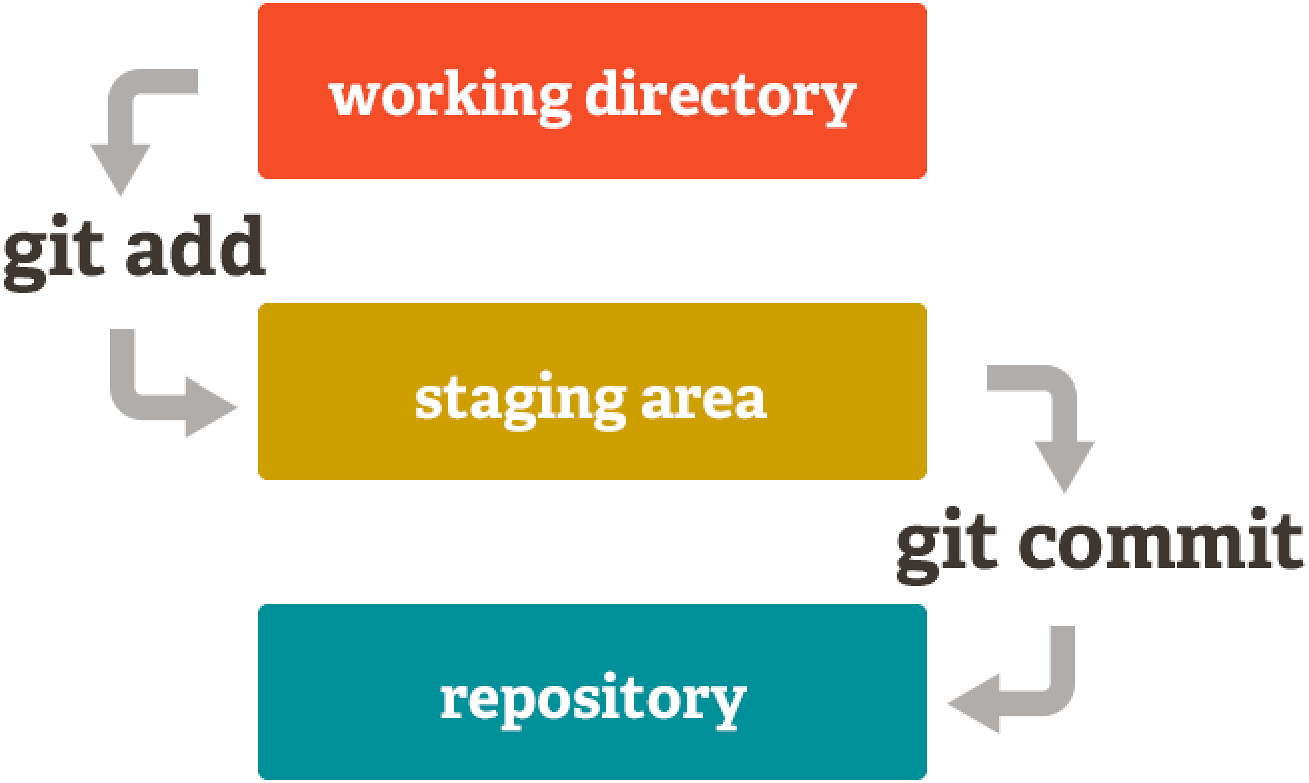
三个重要目录


working directory
staging area
git directory
(repository)
stage files
commit
checkout the project

本地操作 最基本的命令

init	初始化本地仓库，确定工作目录
add	确定添加哪些文件到缓存区中
commit	提交本次修改

git



working directory

git add

staging area

git commit

repository

git 的简单设置

git 三个配置文件

/etc/gitconfig	对所有用户都有用的最普遍的设置	git config -- system
~/.gitconfig	对当前用户有用的设置	git config -- global
.git/config	当前工作目录的设置	git config

```
git config --global  user.name   “lisnb”
git config --global  user.email   “lisnb.h@hotmail.com”

git config --global  core.autocrlf
git config --global  core.editor
git config --global  core.diff
```

…

如何获得一个代码仓库

git init （--bare)

如何获得一个代码仓库

git clone

HTTPS ：https://github.com/lisnb/intro_to_git.git

SSH ：(ssh://) git@github.com:lisnb/intro_to_git.git

GIT ：git://github.com/lisnb/intro_to_git.git

本地协议: (file://) ~/.gitrepos/intro_to_git.git

eg:

```
git clone git@github.com:lisnb/intro_to_git.git
```

git 的基本工作流

编辑文件 vim / sublime text / visual studio

git 的基本工作流

编辑文件	vim / sublime text / visual studio
保存变更	git add (file)

git 的基本工作流

编辑文件	vim / sublime text / visual studio
保存变更	git add (file)
浏览变更	git status / git diff

git 的基本工作流

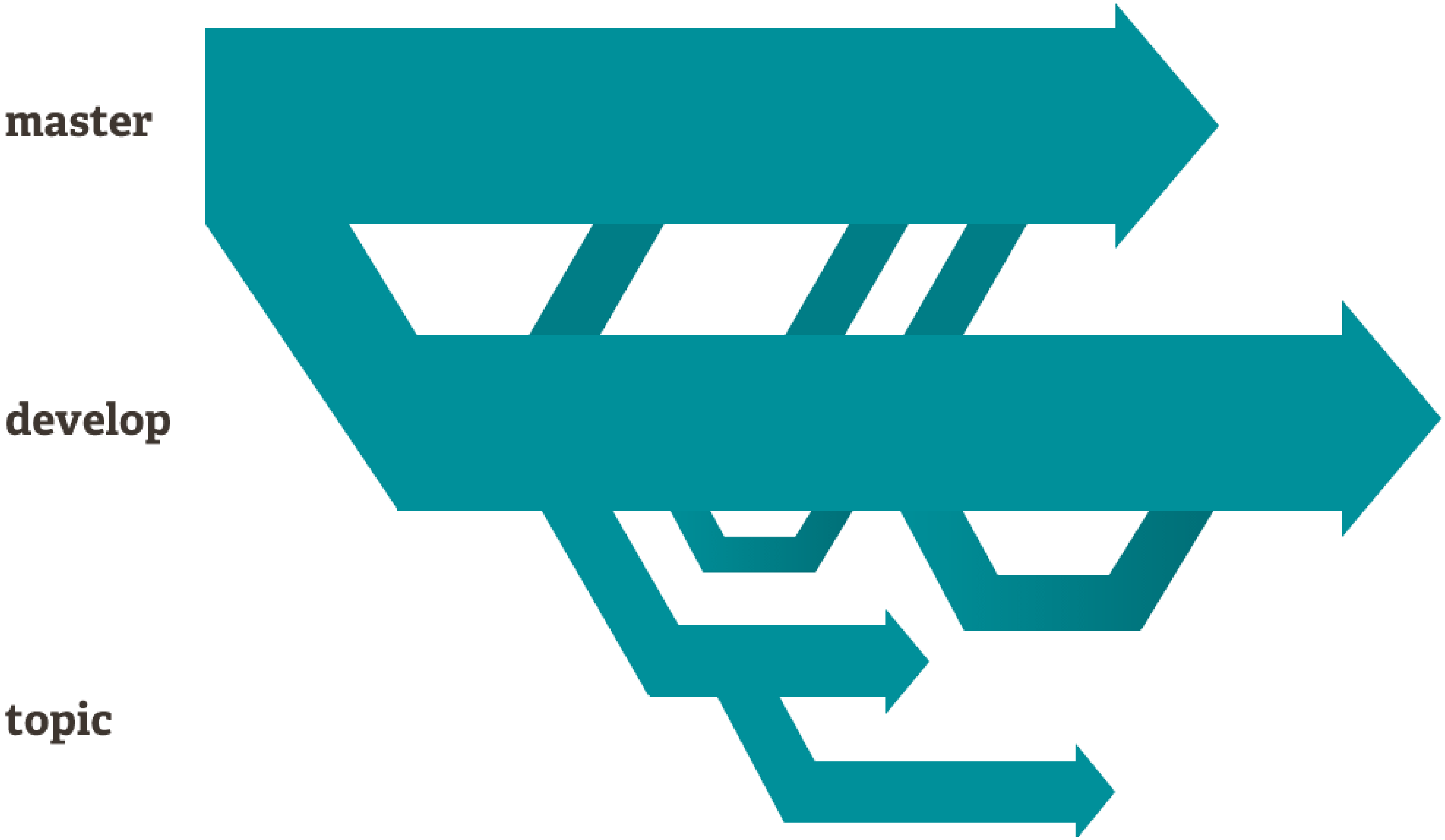
编辑文件	vim / sublime text / visual studio
保存变更	git add (file)
浏览变更	git status / git diff
提交变更	git commit

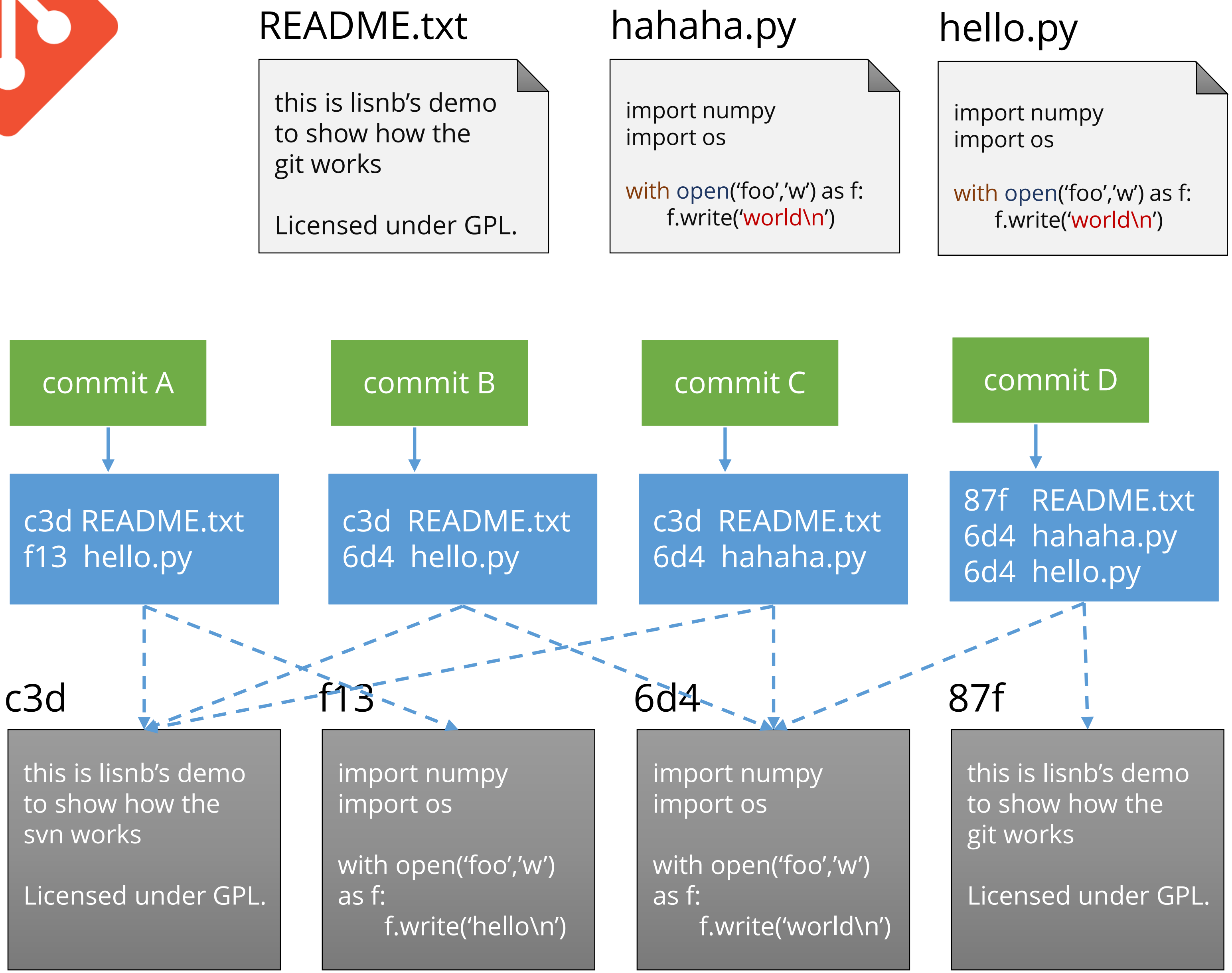
git 的更基本的工作流

编辑文件 vim / sublime text / visual studio

保存并提交变更 git commit –a

git branch





README.txt

this is lisnb’s demo
to show how the
git works

Licensed under GPL.

hahaha.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
    f.write(‘world\n’)
```

hello.py

```
import numpy
import os

with open(‘foo’,’w’) as f:
    f.write(‘world\n’)
```

commit A

c3d README.txt
f13 hello.py

commit B

c3d README.txt
6d4 hello.py

commit C

c3d README.txt
6d4 hahaha.py

commit D

87f README.txt
6d4 hahaha.py
6d4 hello.py

c3d

this is lisnb’s demo
to show how the
svn works

Licensed under GPL.

f13

```
import numpy
import os

with open(‘foo’,’w’)
as f:
    f.write(‘hello\n’)
```

6d4

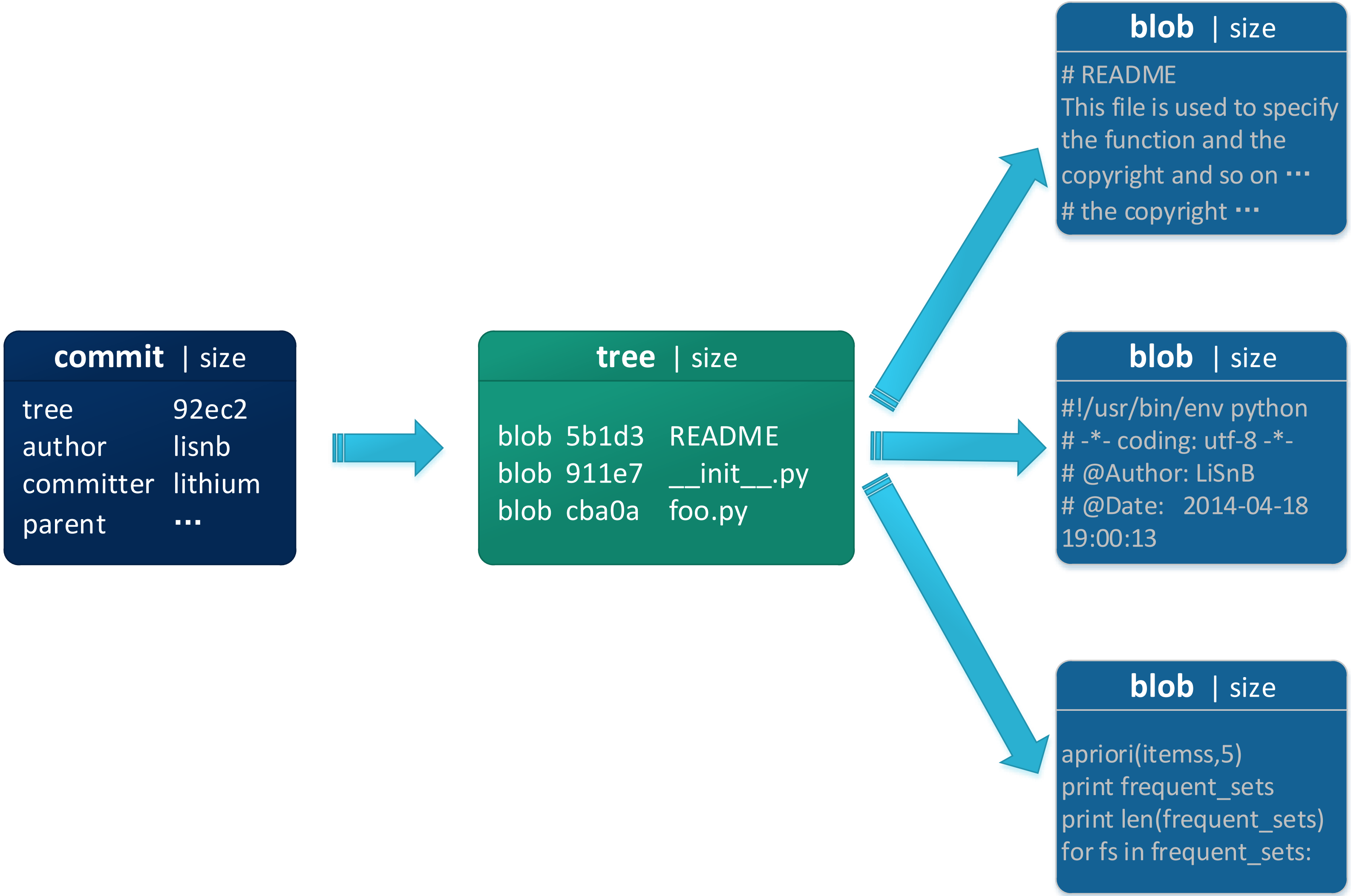
```
import numpy
import os

with open(‘foo’,’w’)
as f:
    f.write(‘world\n’)
```

87f

this is lisnb’s demo
to show how the
git works

Licensed under GPL.

blob, tree, commit



commit | size

tree	92ec2
author	lisnb
committer	lithium
parent	···

tree | size

```
blob  5b1d3   README
blob  911e7   __init__.py
blob  cba0a   foo.py
```

blob | size

```
# README
This file is used to specify
the function and the
copyright and so on ···
# the copyright ···
```

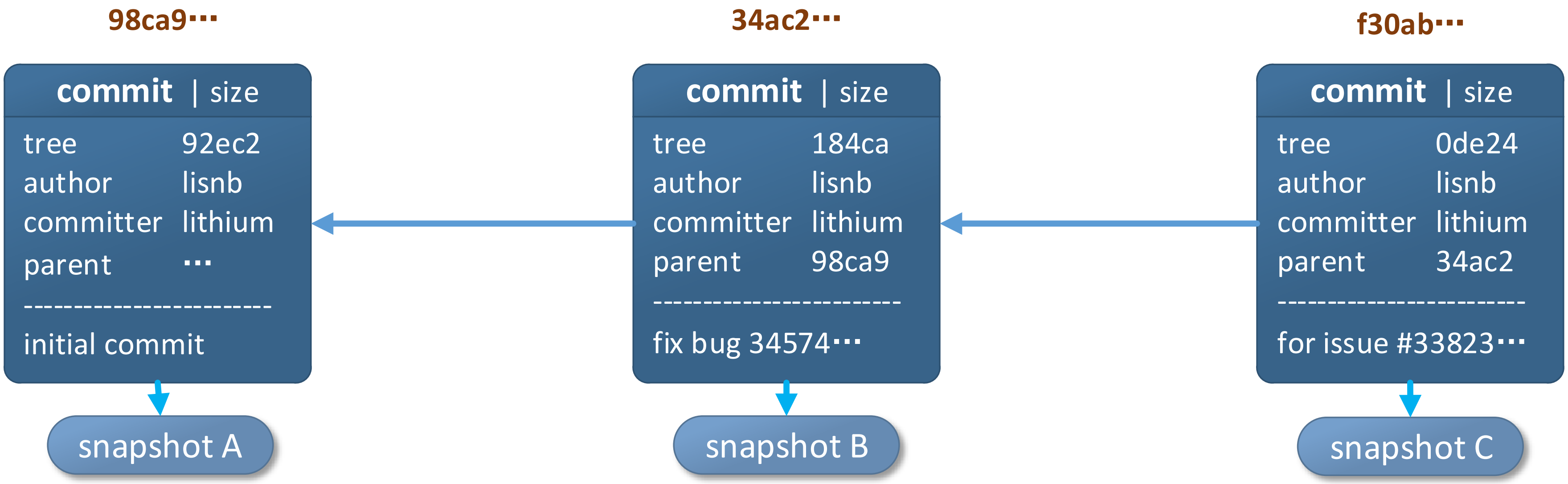
blob | size

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# @Author: LiSnB
# @Date:   2014-04-18
19:00:13
```

blob | size

```
apriori(itemss,5)
print frequent_sets
print len(frequent_sets)
for fs in frequent_sets:
```

git commits



98ca9…

commit | size

tree 92ec2
author lisnb
committer lithium
parent …

----

initial commit

snapshot A

34ac2…

commit | size

tree 184ca
author lisnb
committer lithium
parent 98ca9

----

fix bug 34574…

snapshot B

f30ab…

commit | size

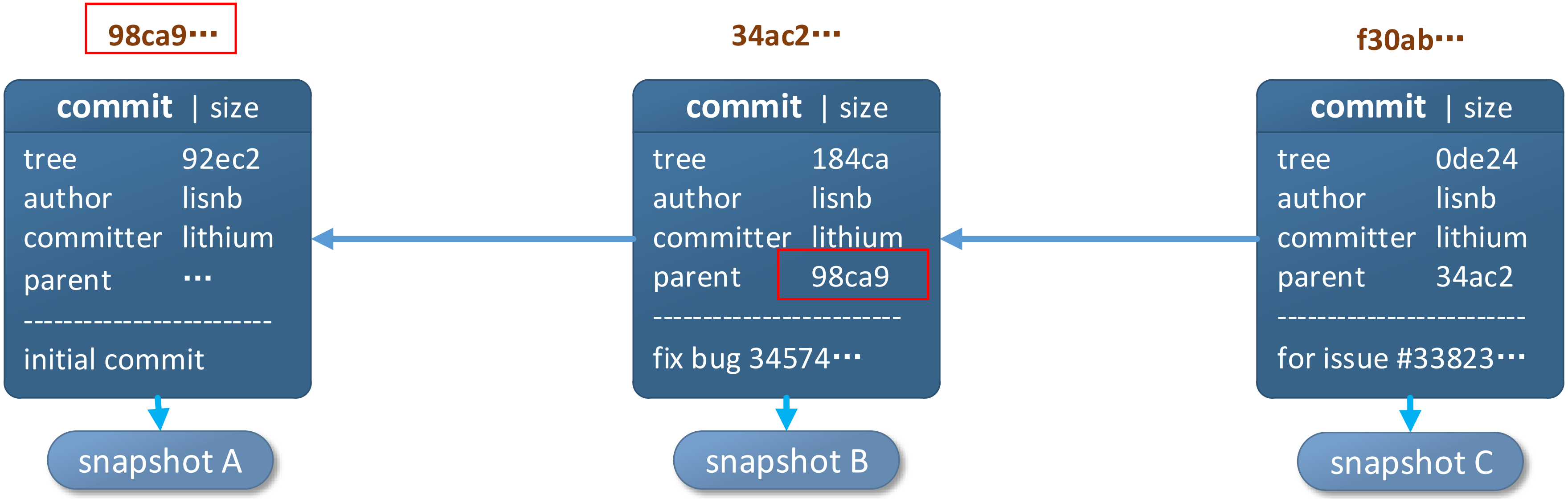
tree 0de24
author lisnb
committer lithium
parent 34ac2

----

for issue #33823…

snapshot C

git commits



98ca9···

commit | size

tree 92ec2
author lisnb
committer lithium
parent ···
----
initial commit

snapshot A

34ac2···

commit | size

tree 184ca
author lisnb
committer lithium
parent 98ca9
----
fix bug 34574···

snapshot B

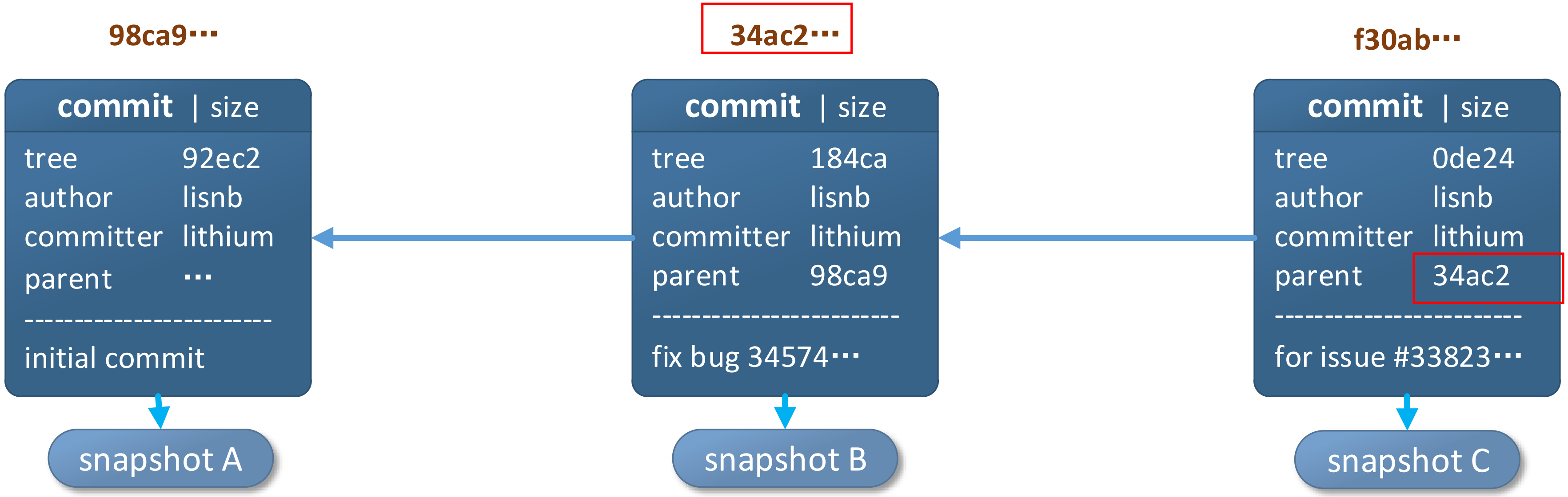
f30ab···

commit | size

tree 0de24
author lisnb
committer lithium
parent 34ac2
----
for issue #33823···

snapshot C

git commits



98ca9···

commit | size

tree 92ec2
author lisnb
committer lithium
parent ···
----
initial commit

snapshot A

34ac2···

commit | size

tree 184ca
author lisnb
committer lithium
parent 98ca9
----
fix bug 34574···

snapshot B

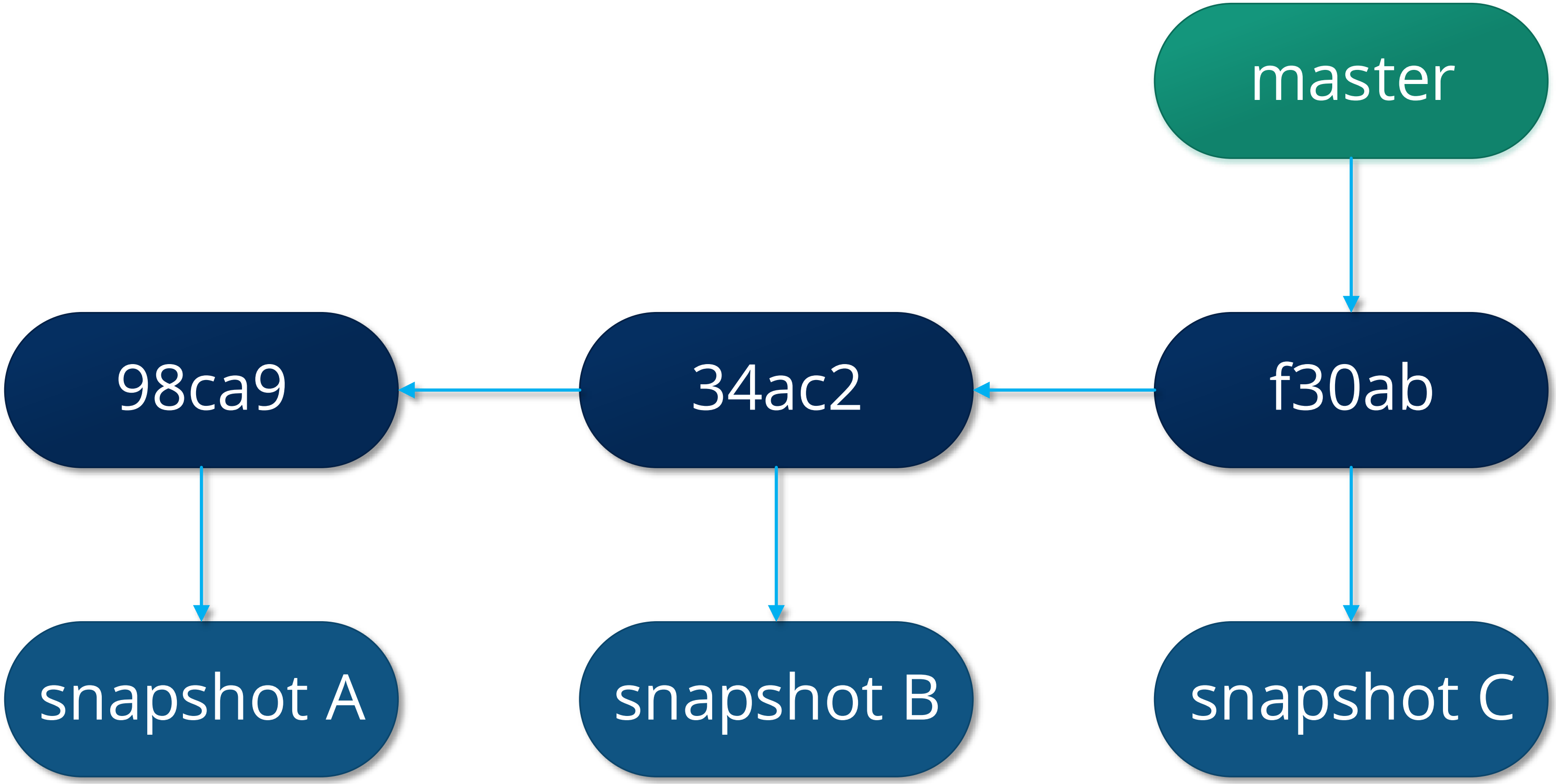
f30ab···

commit | size

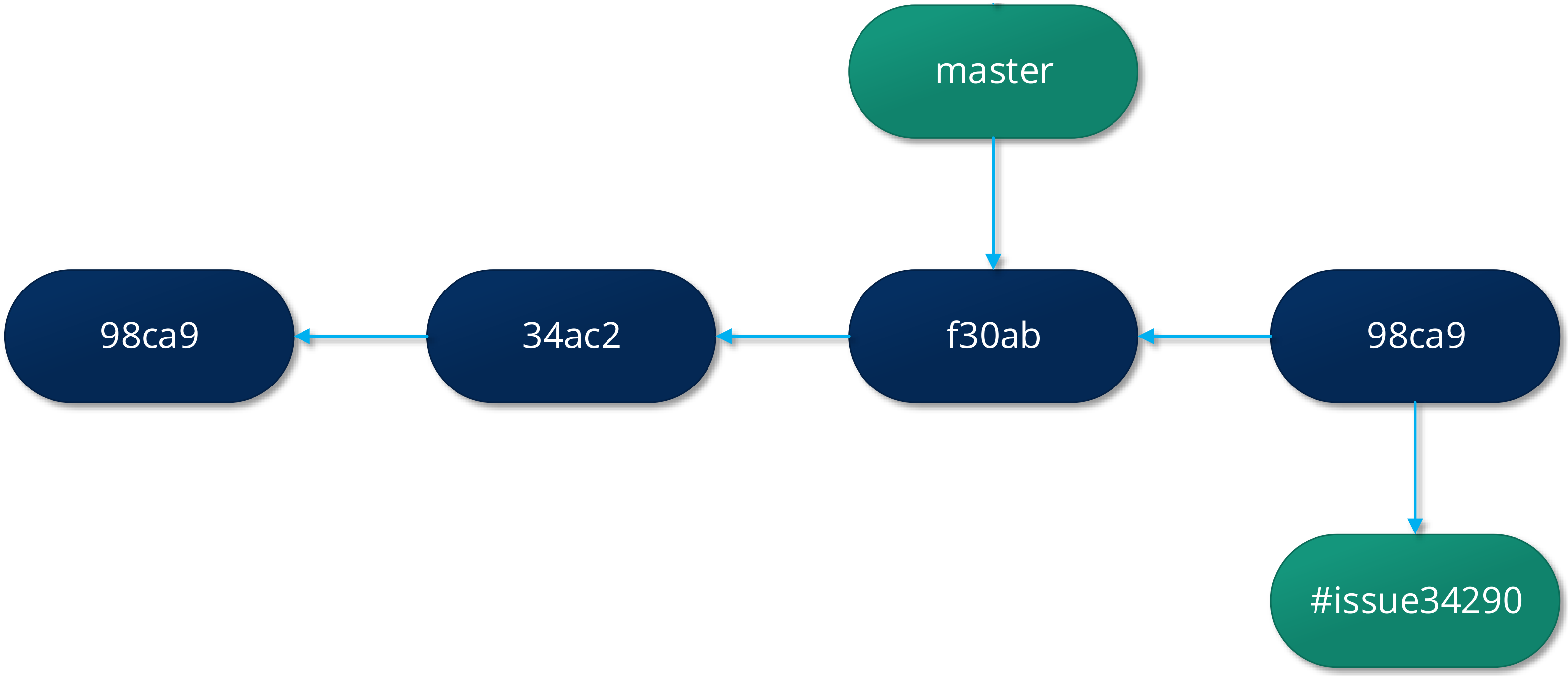
tree 0de24
author lisnb
committer lithium
parent 34ac2
----
for issue #33823···

snapshot C

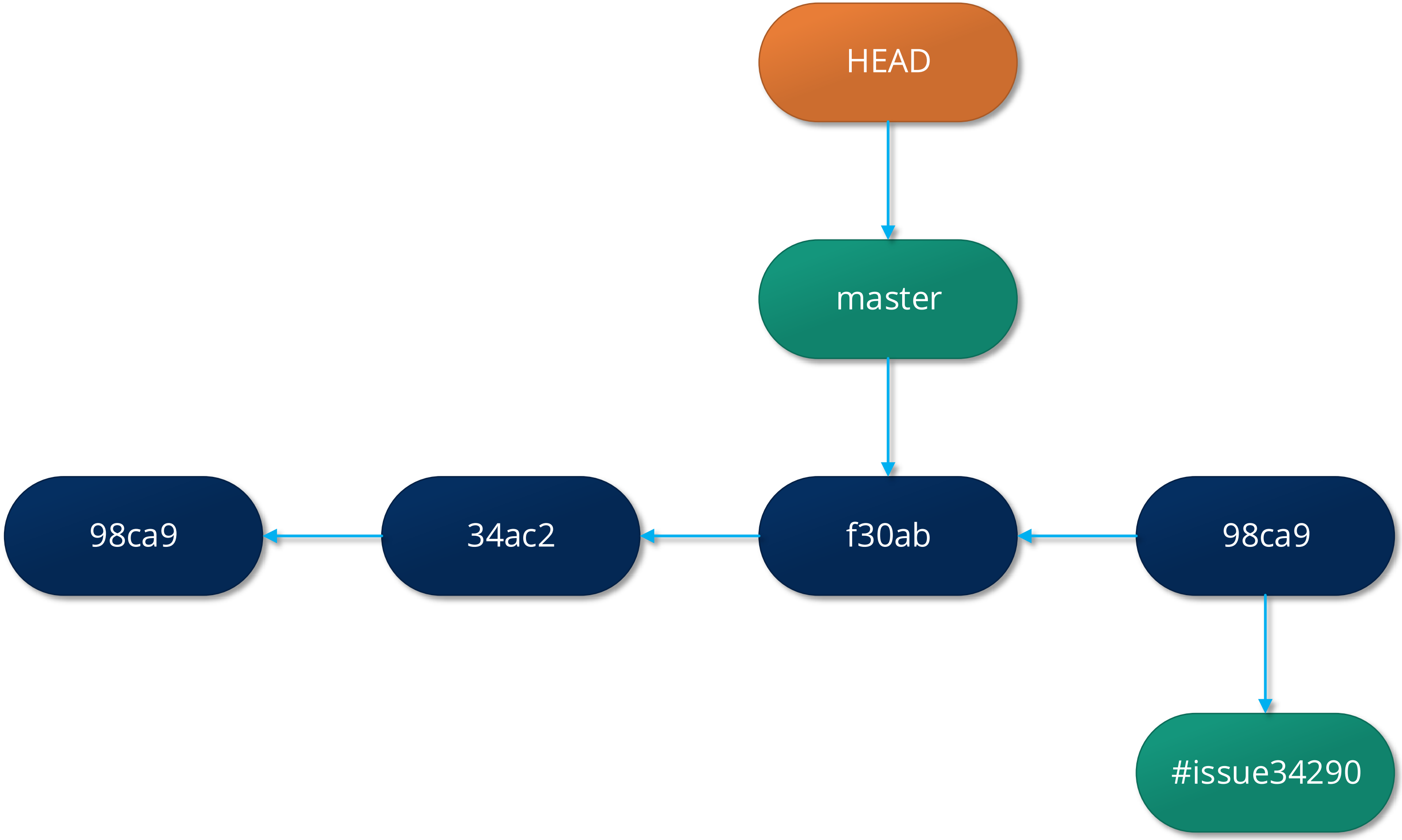
git branch



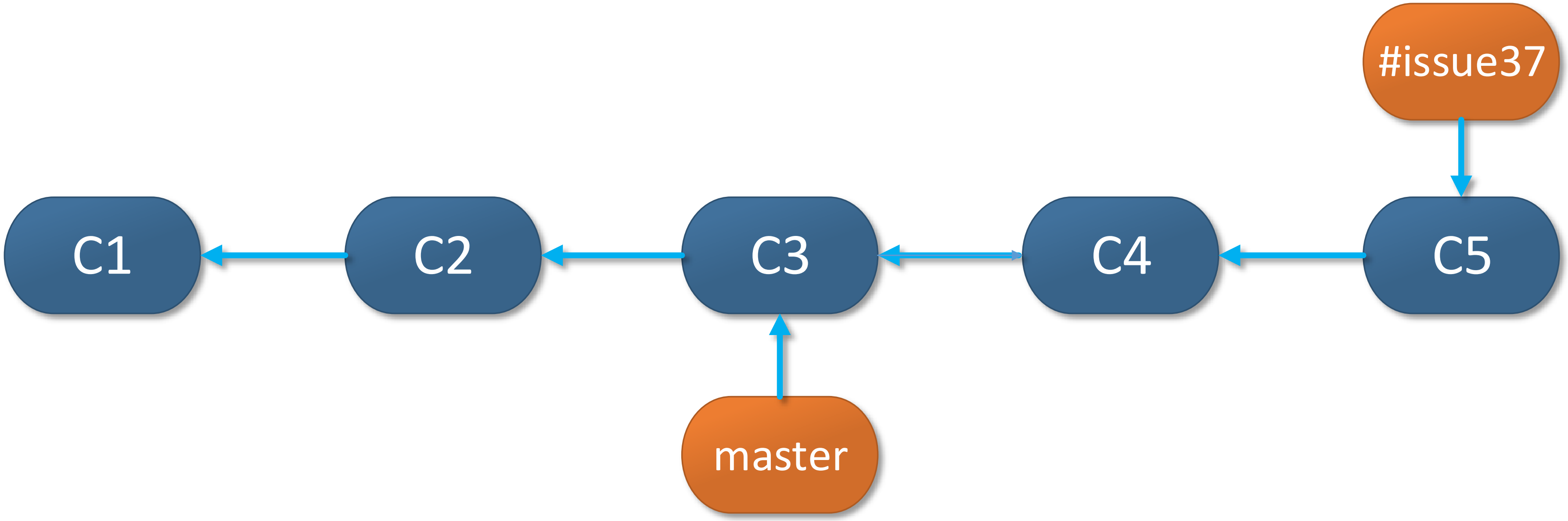
git branches



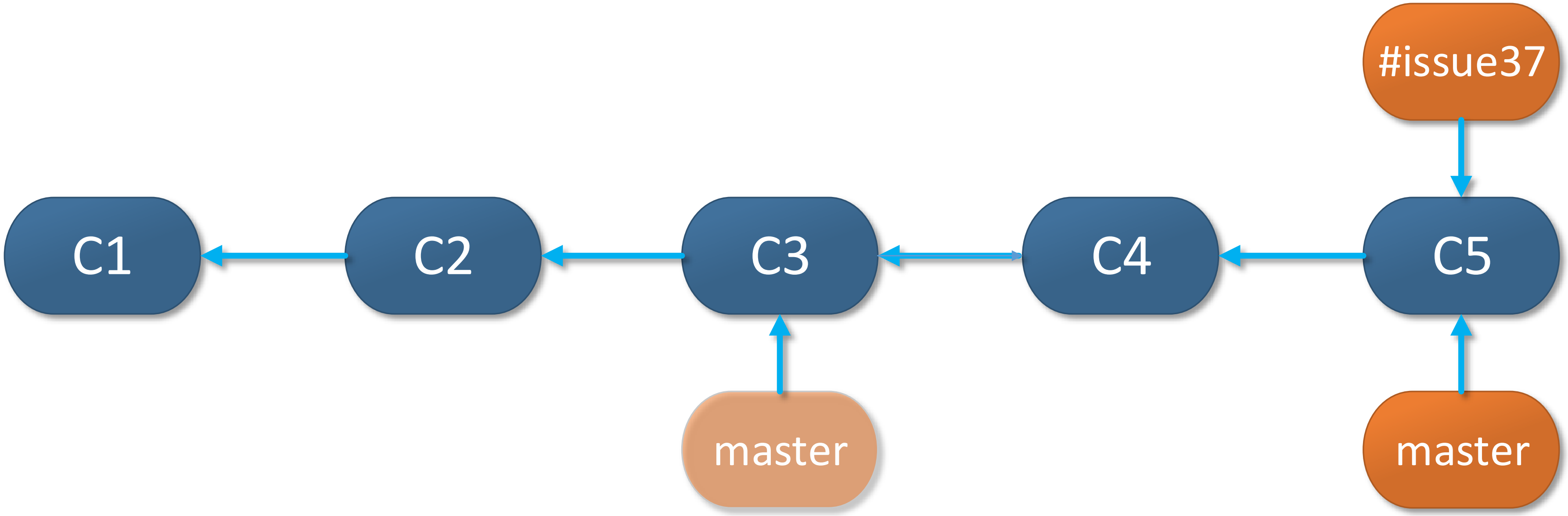
HEAD



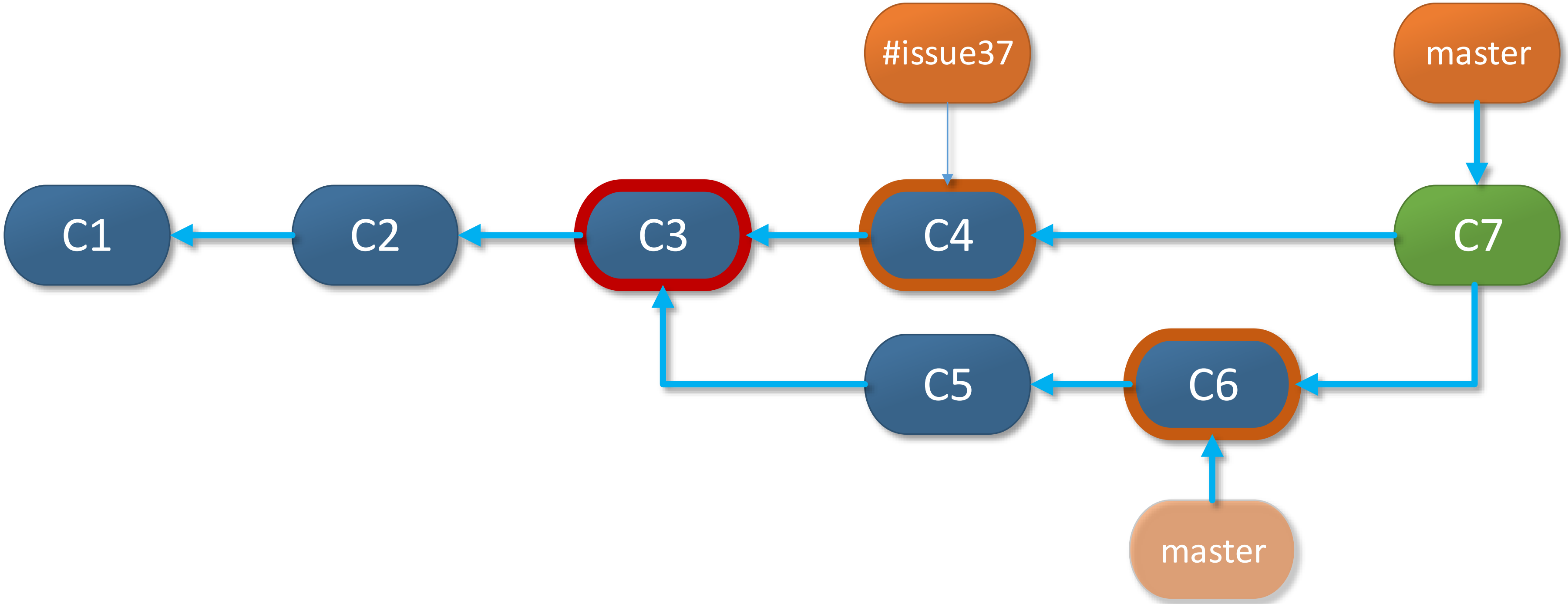
git merge – fast forward



git merge – fast forward



git merge



git 的分支操作

查看分支 git branch

git 的分支操作

查看分支	git branch
新建分支	git branch <branchname>

git 的分支操作

查看分支	git branch
新建分支	git branch <branchname>
删除分支	git branch (-d | -D) <branchname>

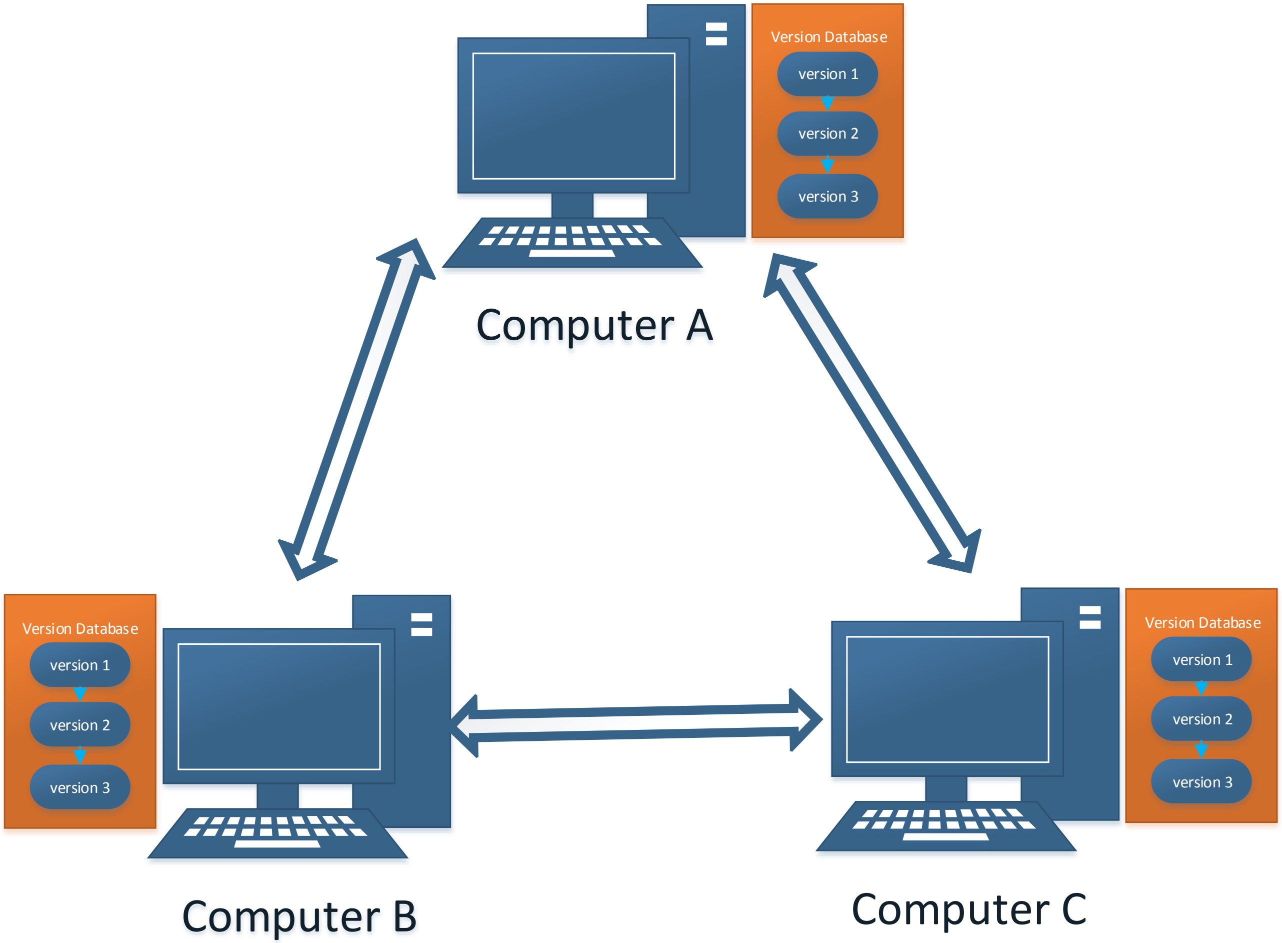
git 的分支操作

查看分支	git branch
新建分支	git branch <branchname>
删除分支	git branch (-d | -D) <branchname>
分支合并	git merge <branchname> (--ff-only)

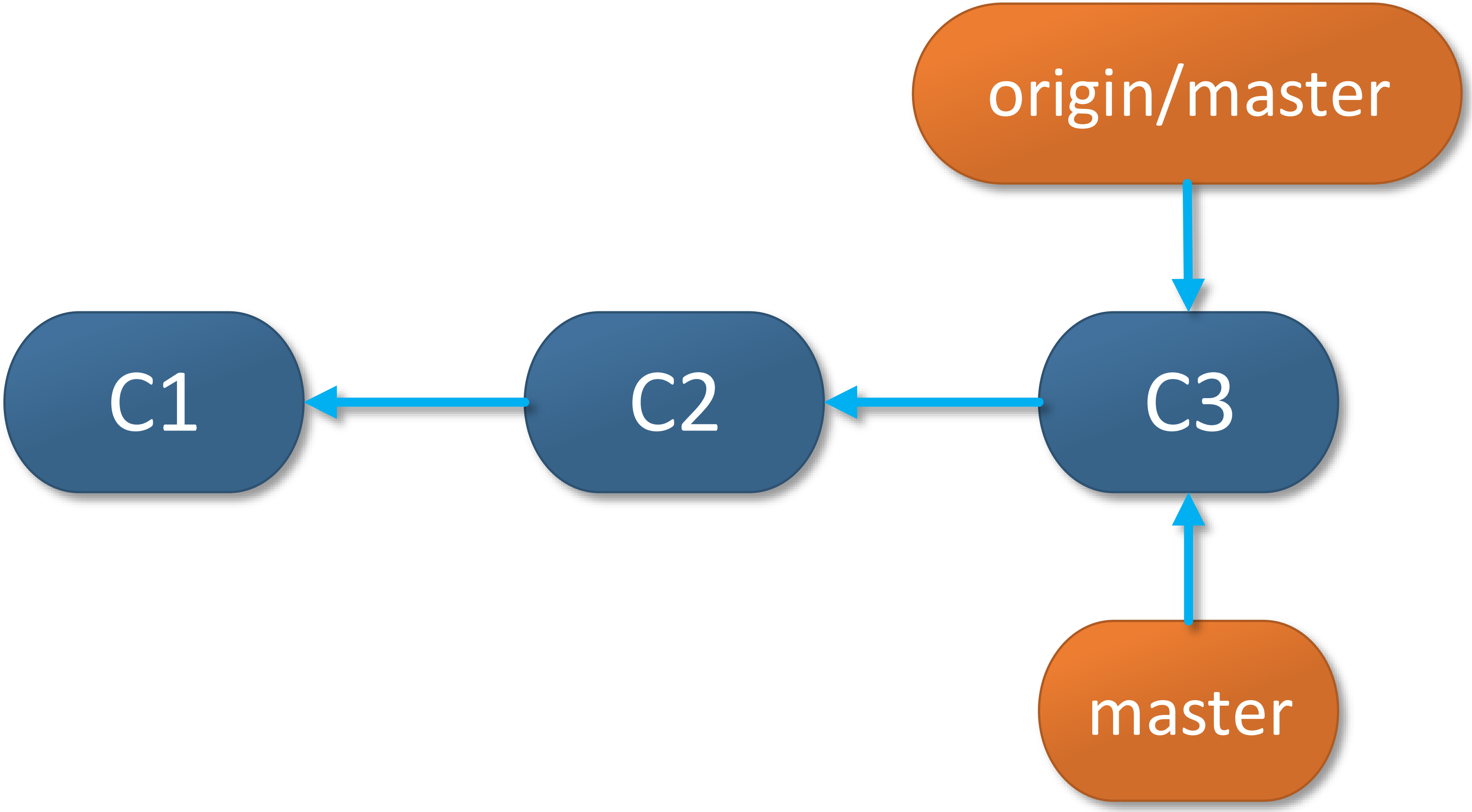
git 的分支操作

查看分支	git branch
新建分支	git branch <branchname>
删除分支	git branch (-d | -D) <branchname>
分支合并	git merge <branchname> (--ff-only)
检出分支	git checkout (-b) <branchname>

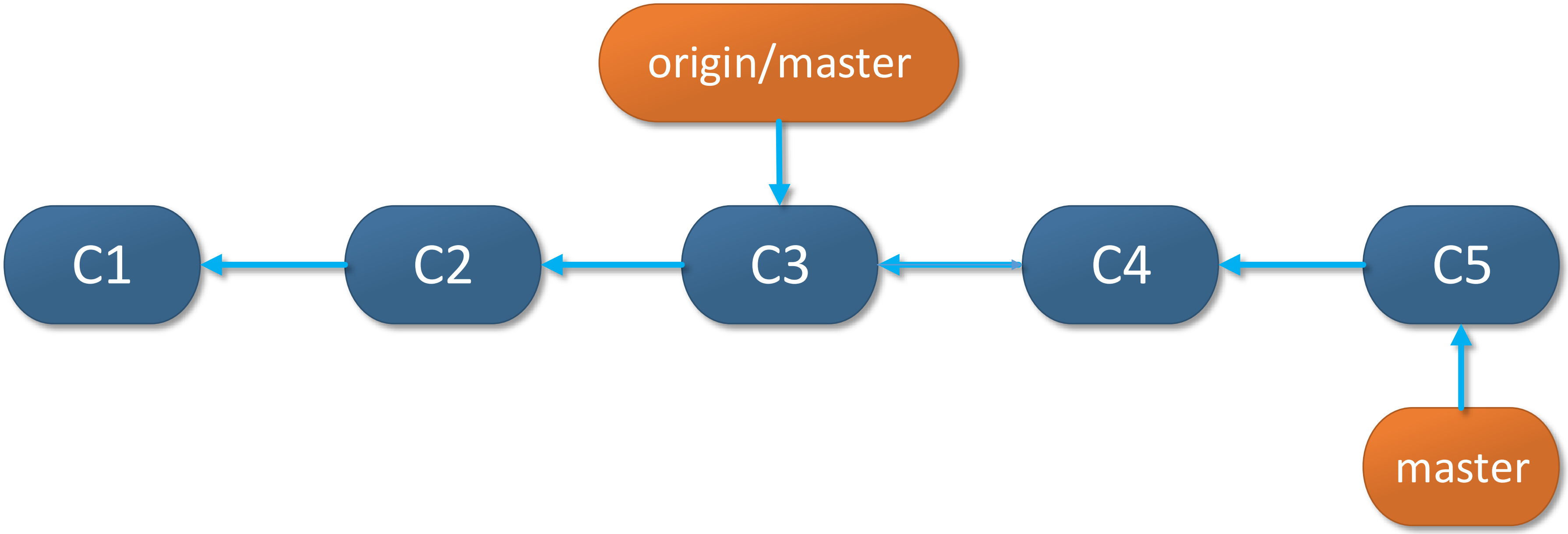
git remote

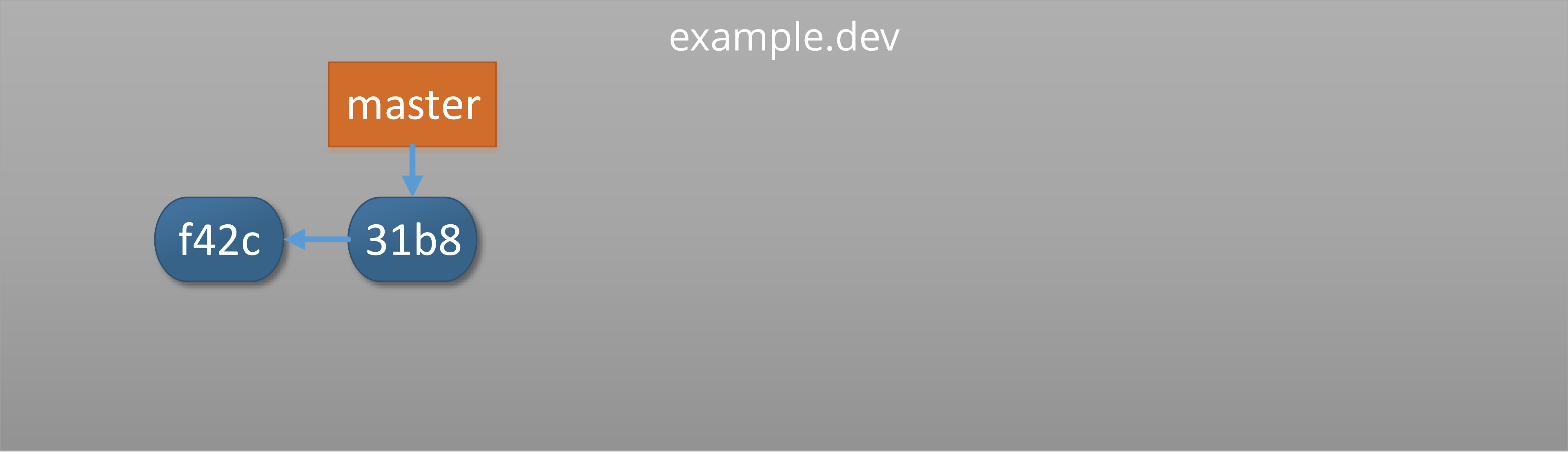

Version Database
version 1
version 2
version 3
Computer A
Version Database
version 1
version 2
version 3
Computer B
Version Database
version 1
version 2
version 3
Computer C

git remote



git remote



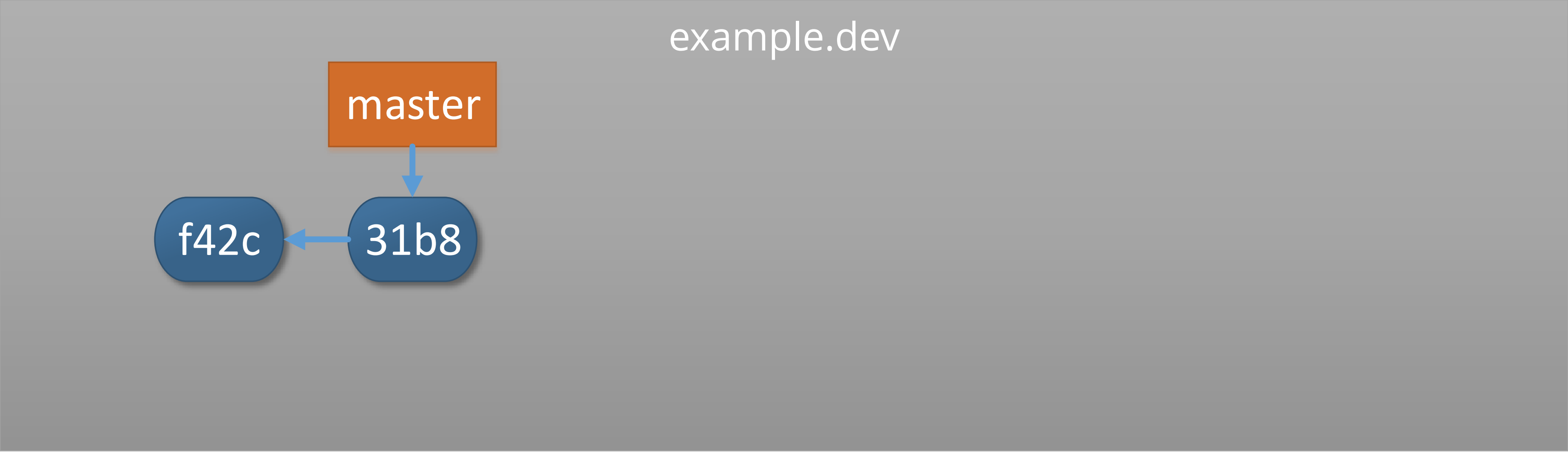


example.dev

master

f42c

31b8



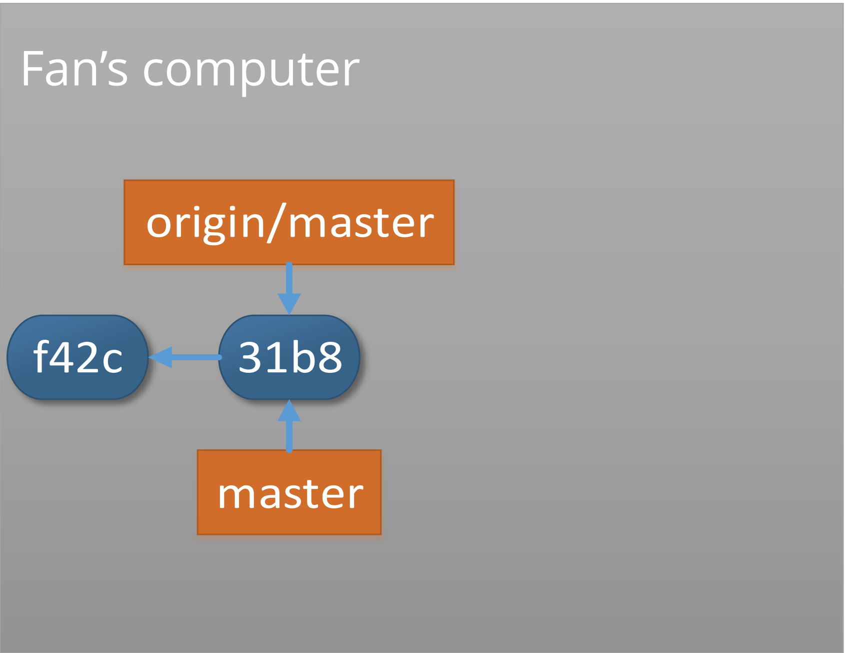
example.dev

master

f42c

31b8

git clone git@example.dev:project.git



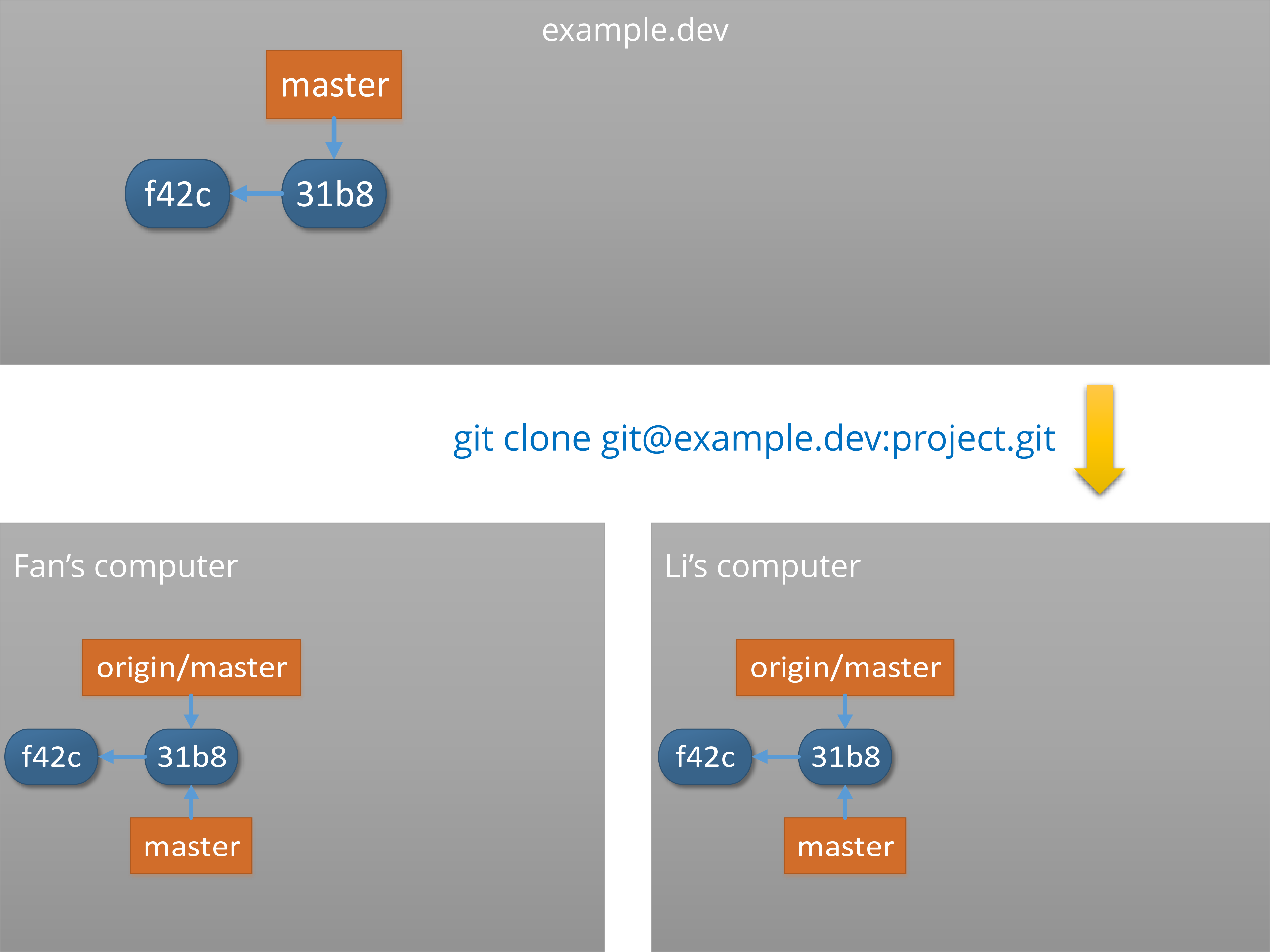
Fan’s computer

origin/master

f42c

31b8

master



example.dev

master

f42c

31b8

```
git clone git@example.dev:project.git
```

Fan’s computer

origin/master

f42c

31b8

master

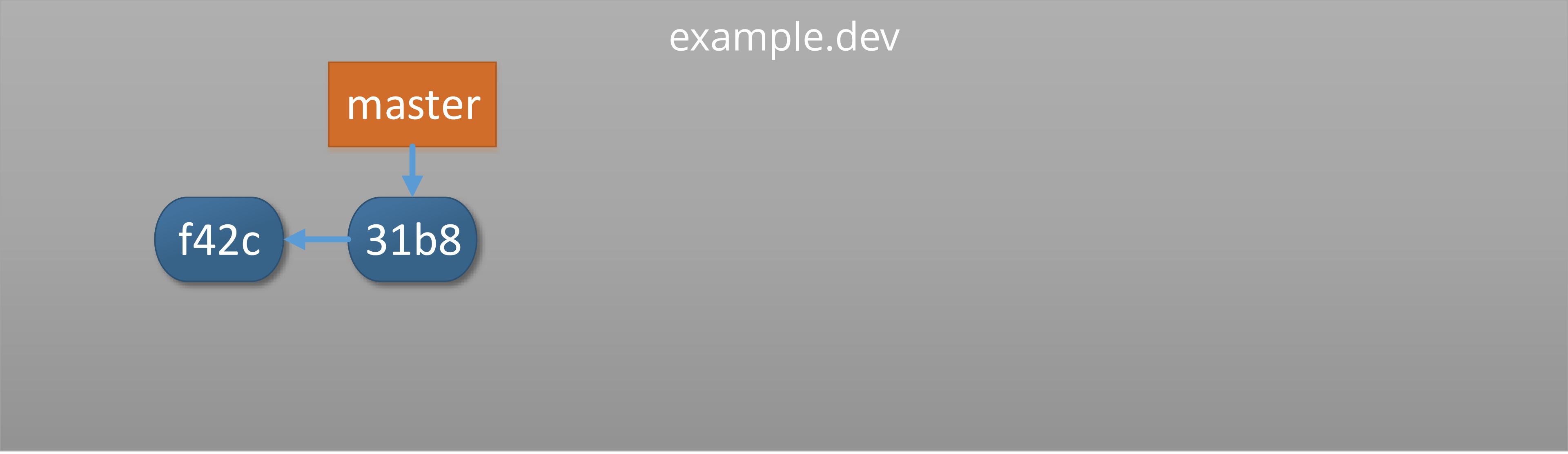
Li’s computer

origin/master

f42c

31b8

master



example.dev

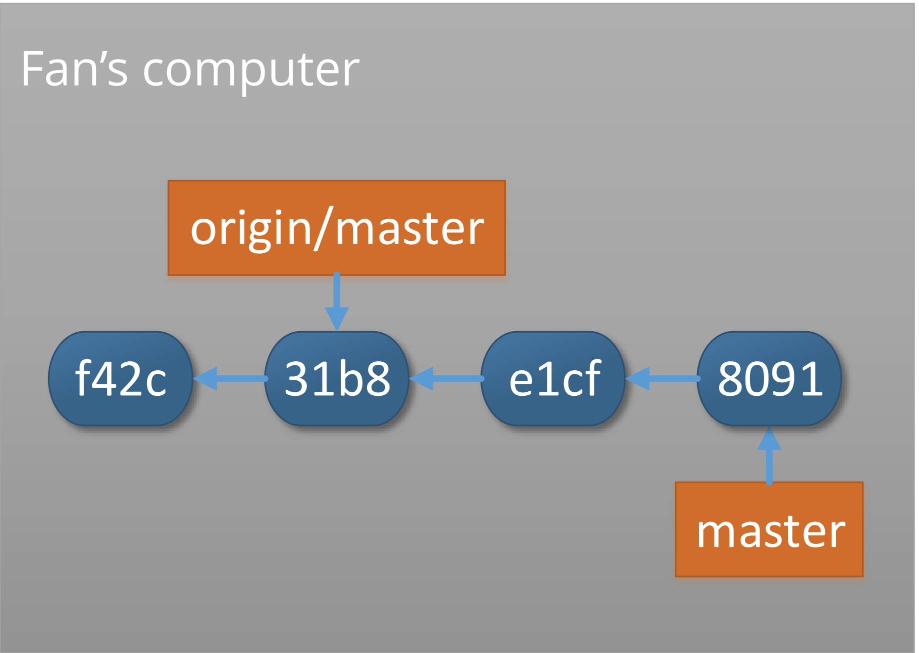
master

f42c

31b8

git commit

git commit



Fan's computer

origin/master

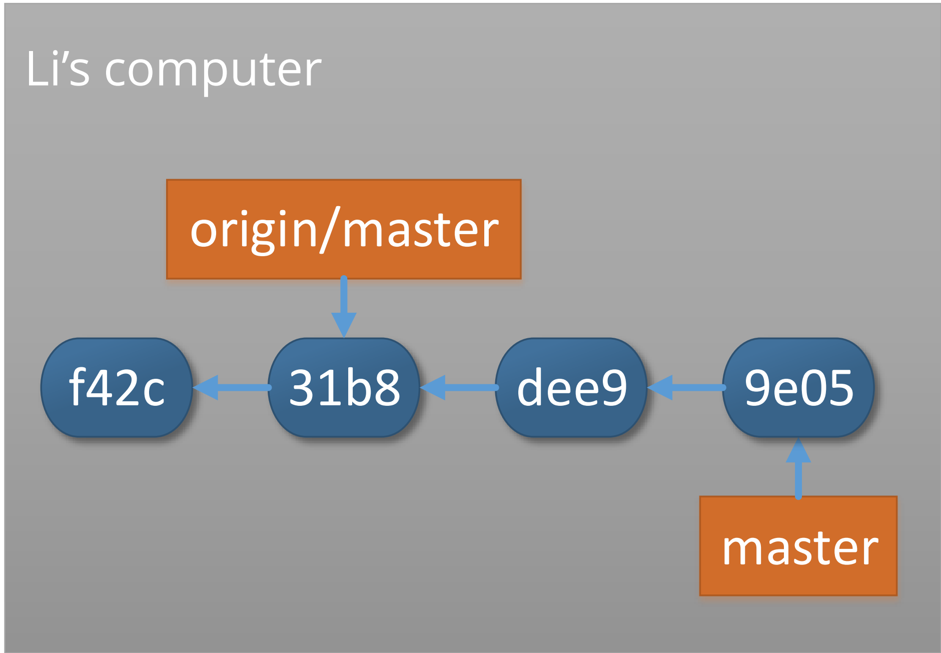
f42c

31b8

e1cf

8091

master



Li's computer

origin/master

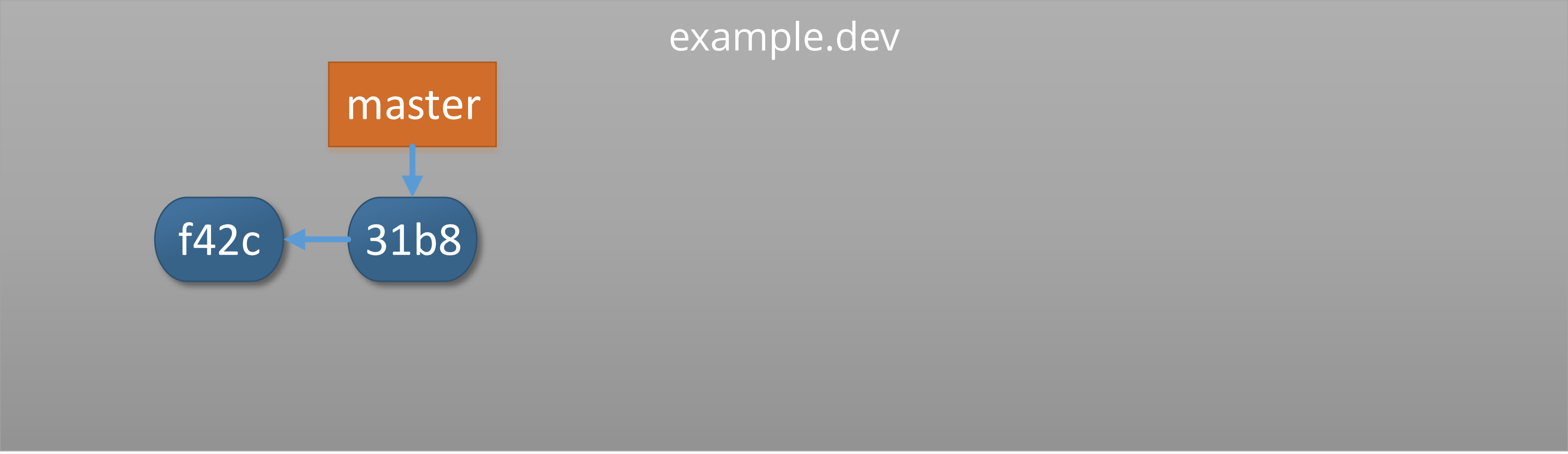
f42c

31b8

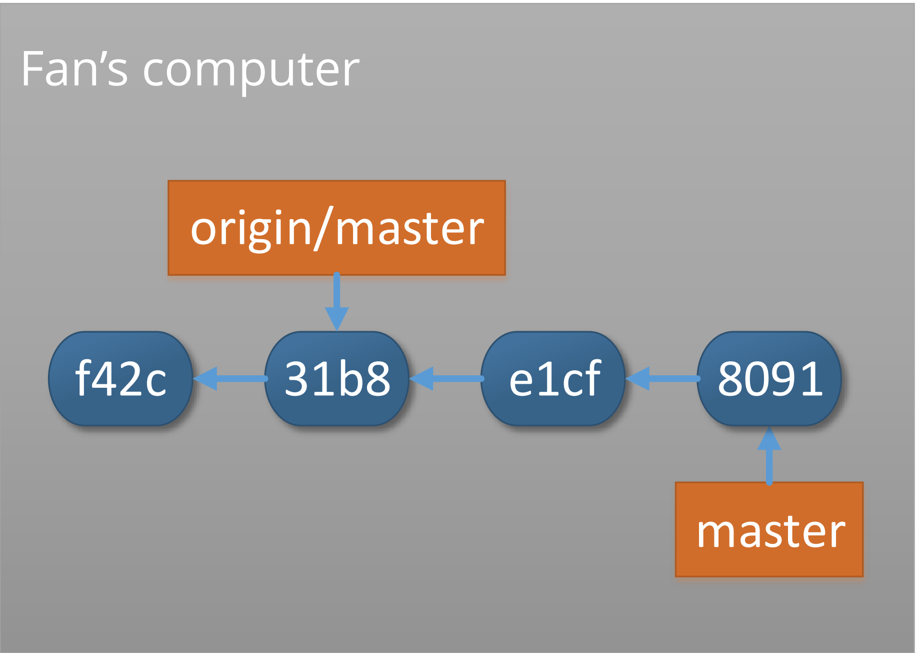
dee9

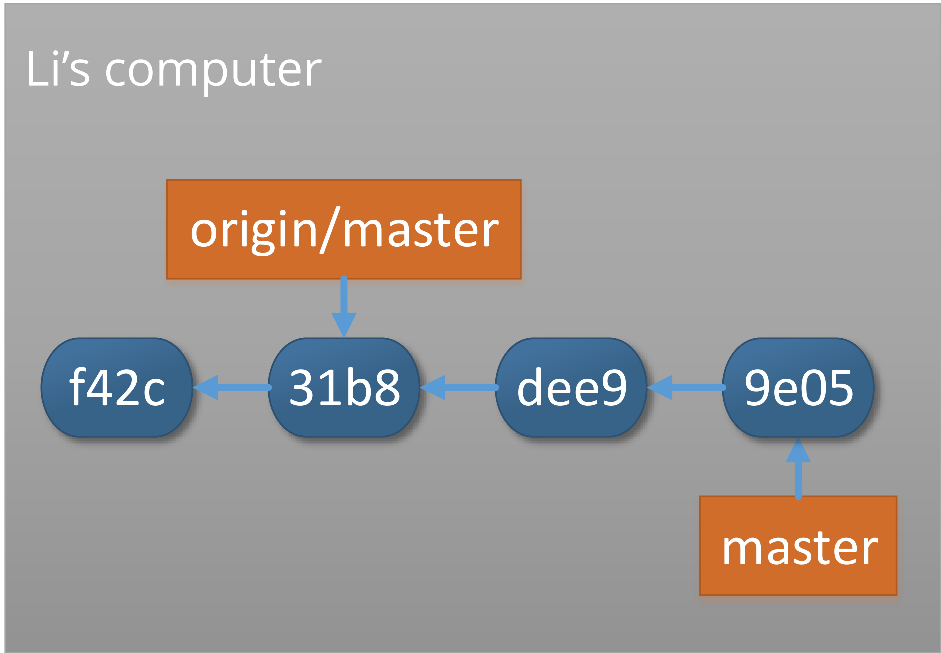
9e05

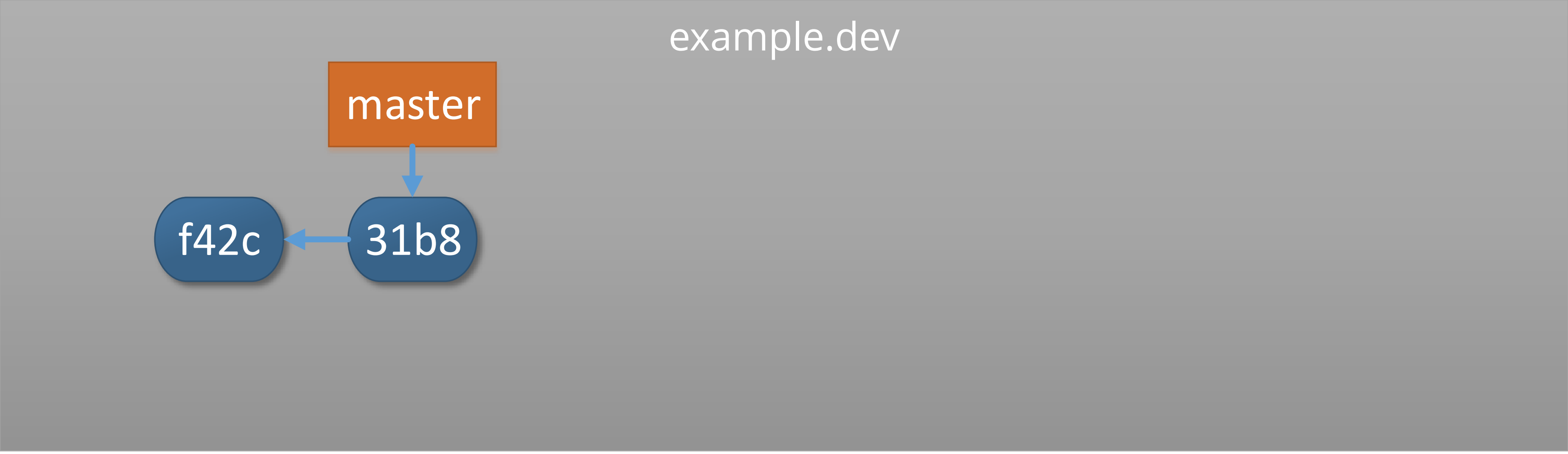
master



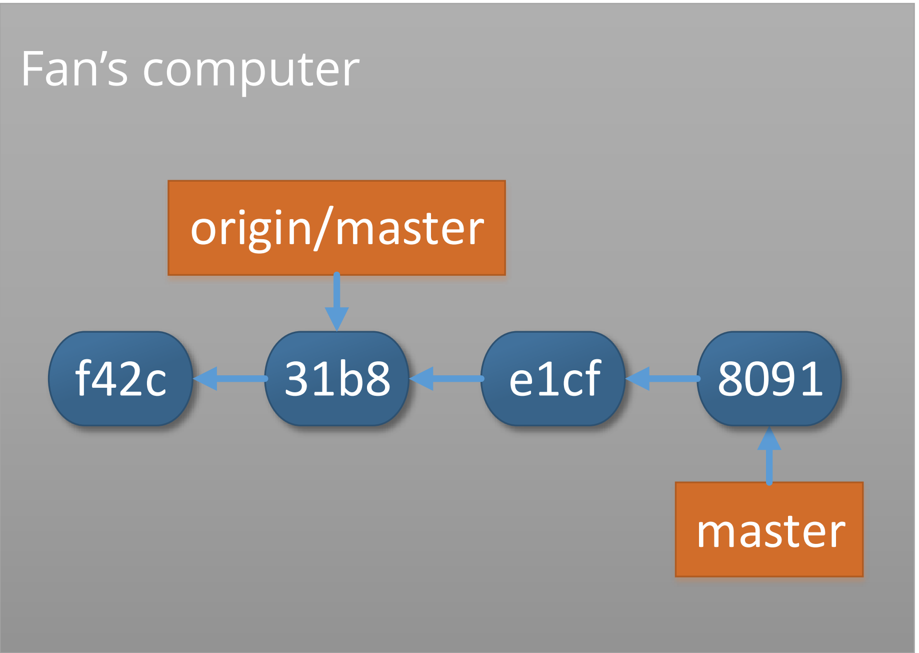
```
git push origin master
```

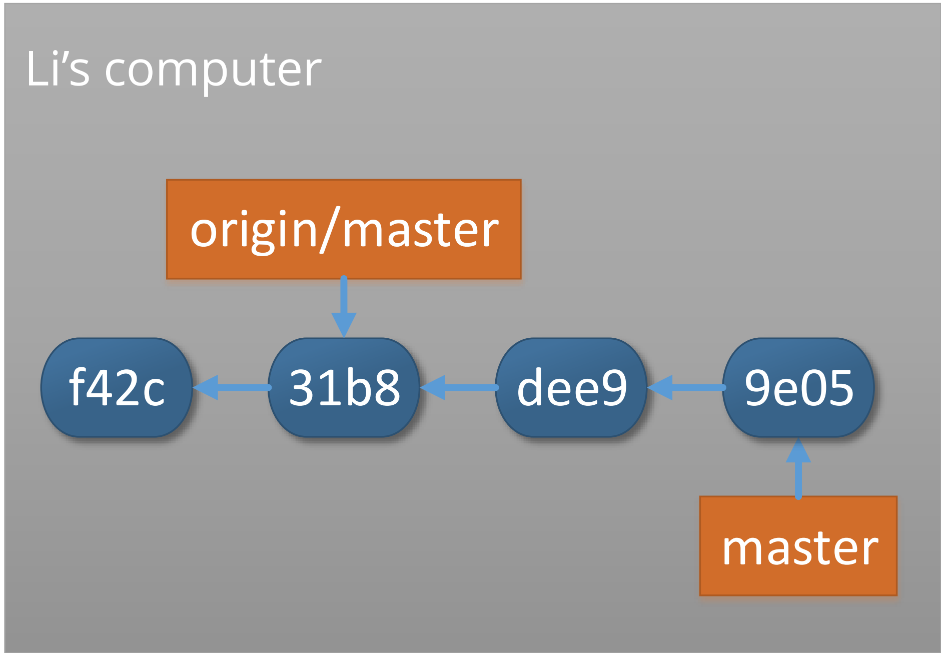


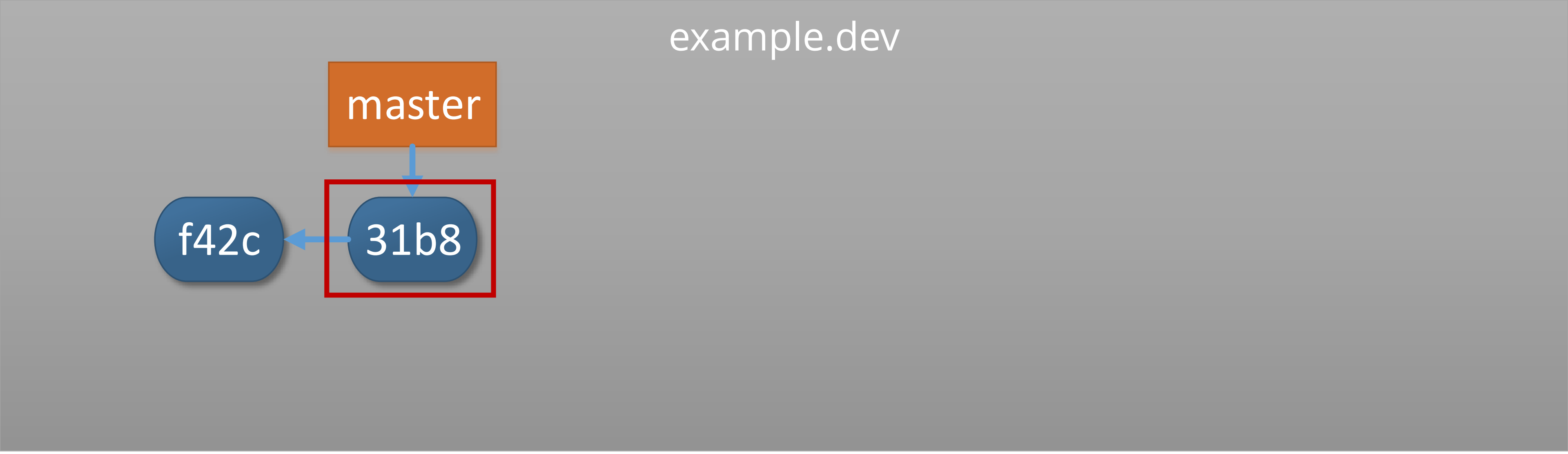




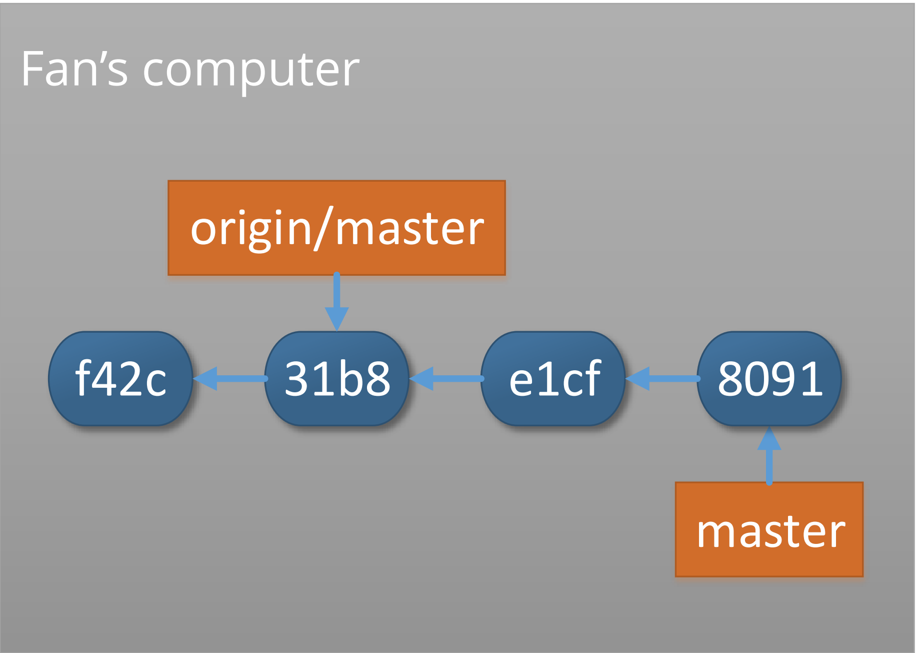
我希望example.dev上的master分支和我本地的一模一样

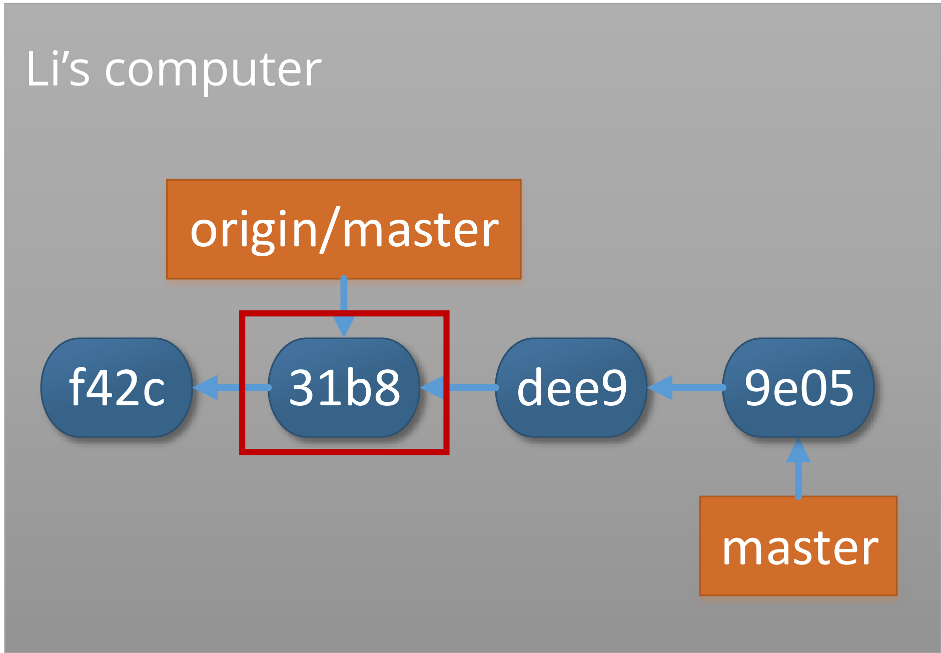


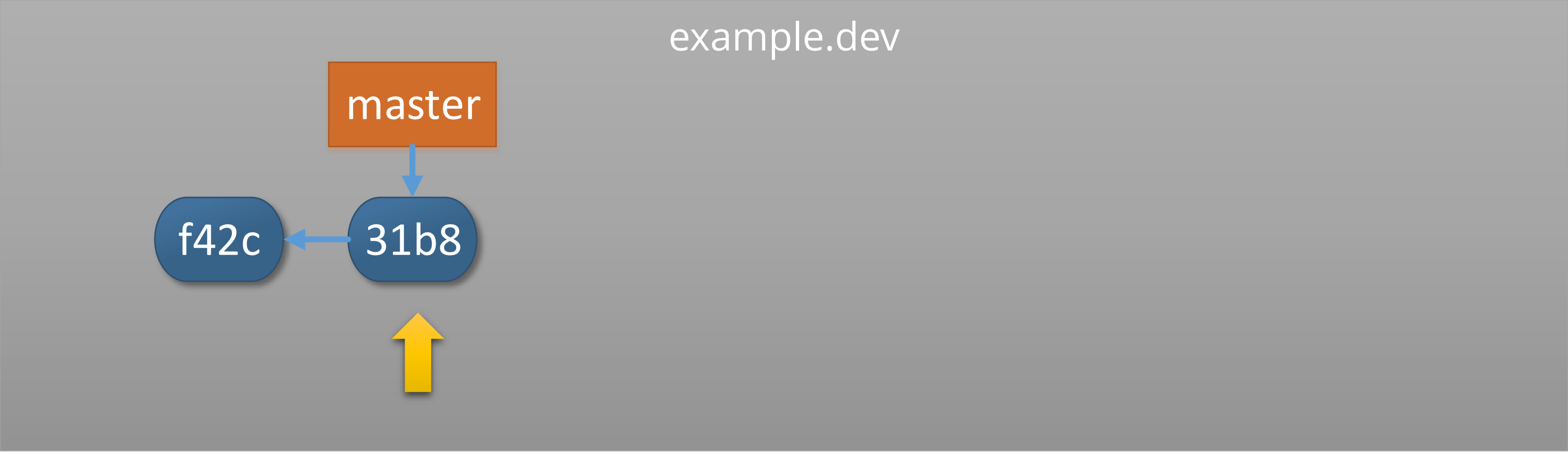




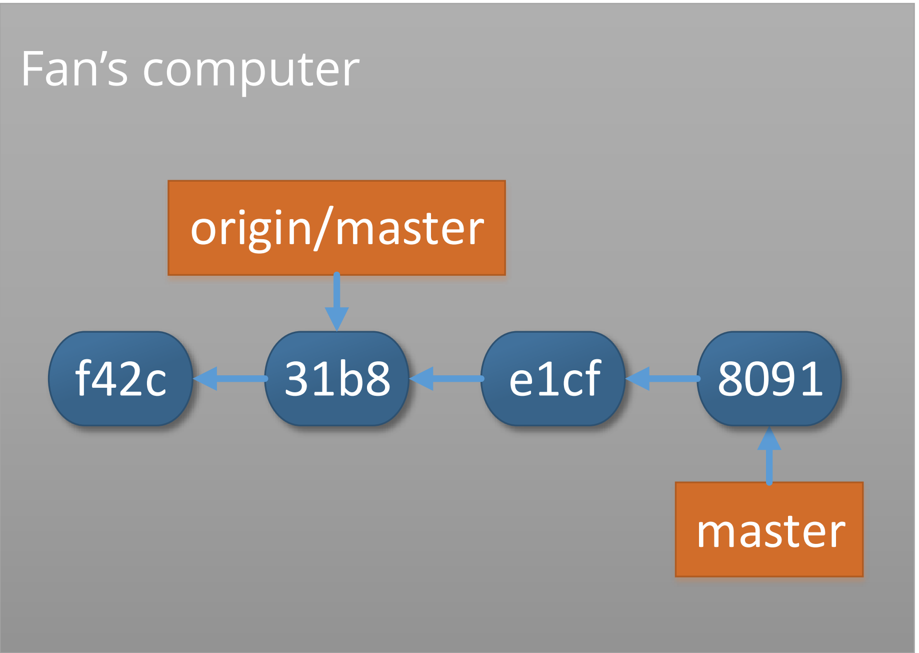
```
git push origin master
```

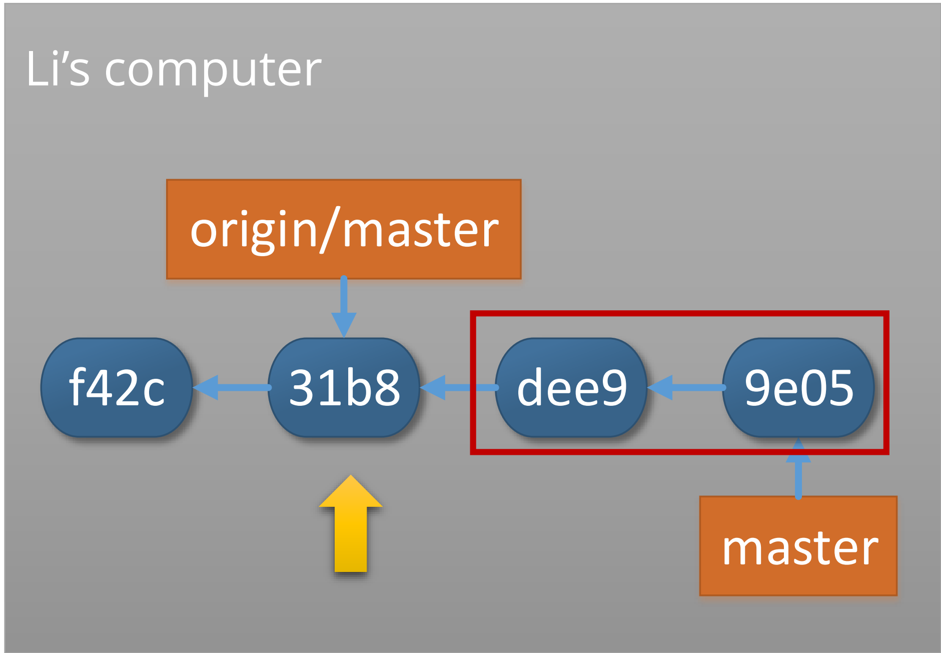


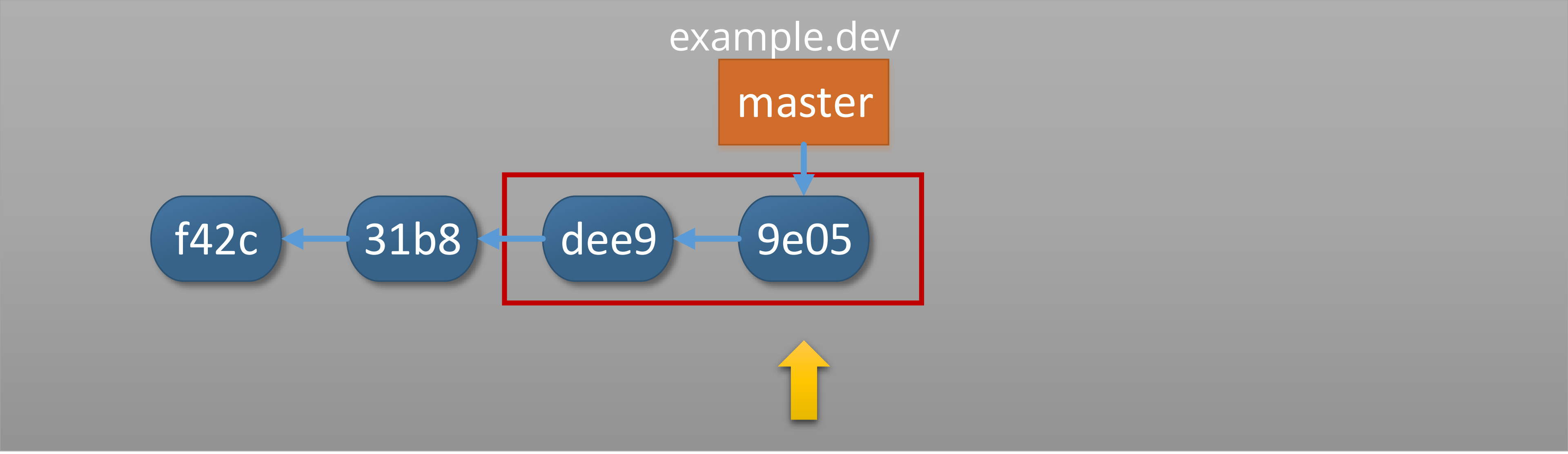




git push origin master







example.dev

master

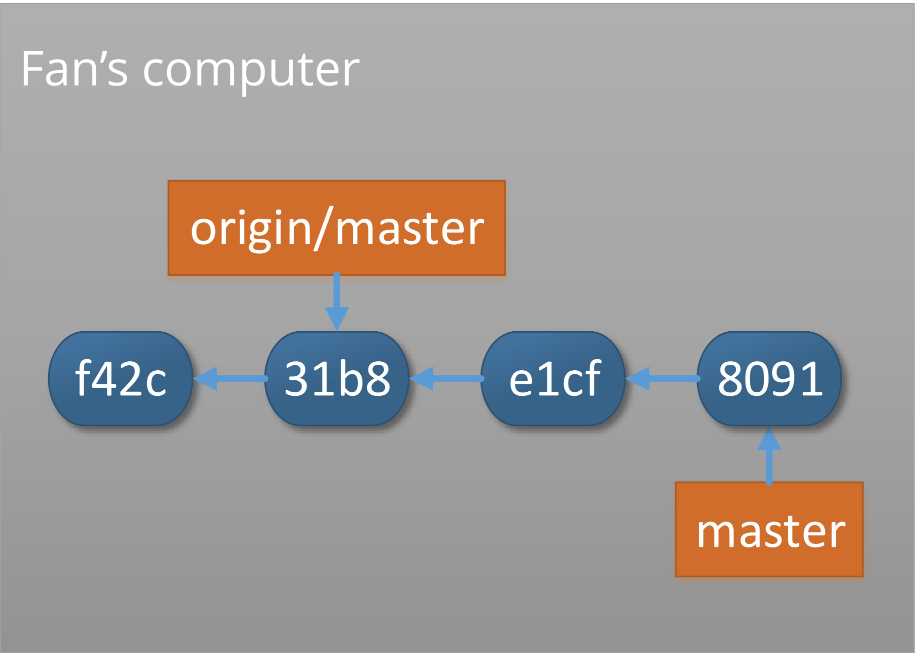
f42c

31b8

dee9

9e05

```
git push origin master
```



Fan's computer

origin/master

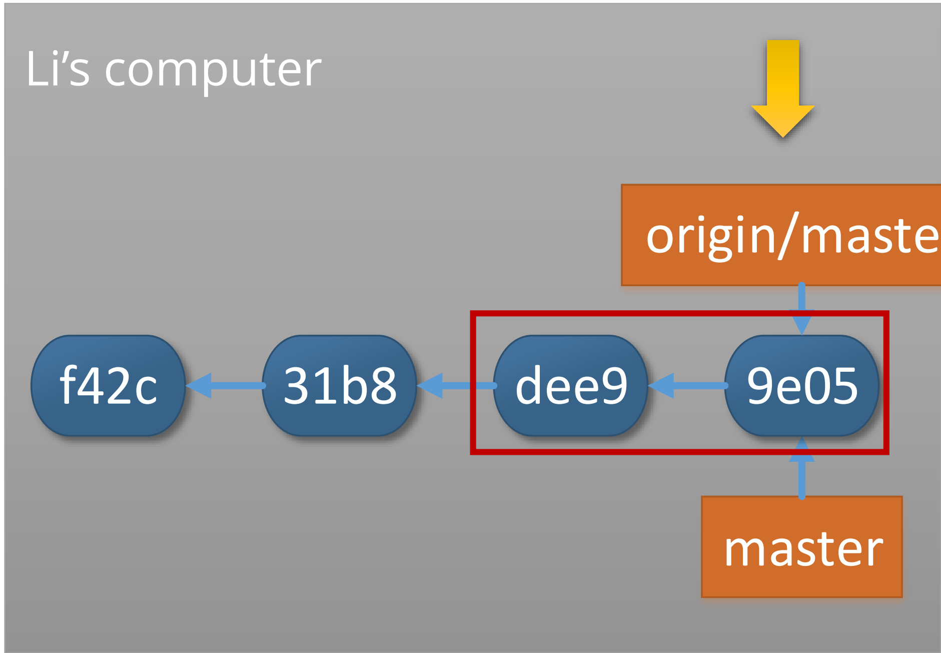
f42c

31b8

e1cf

8091

master



Li's computer

origin/maste

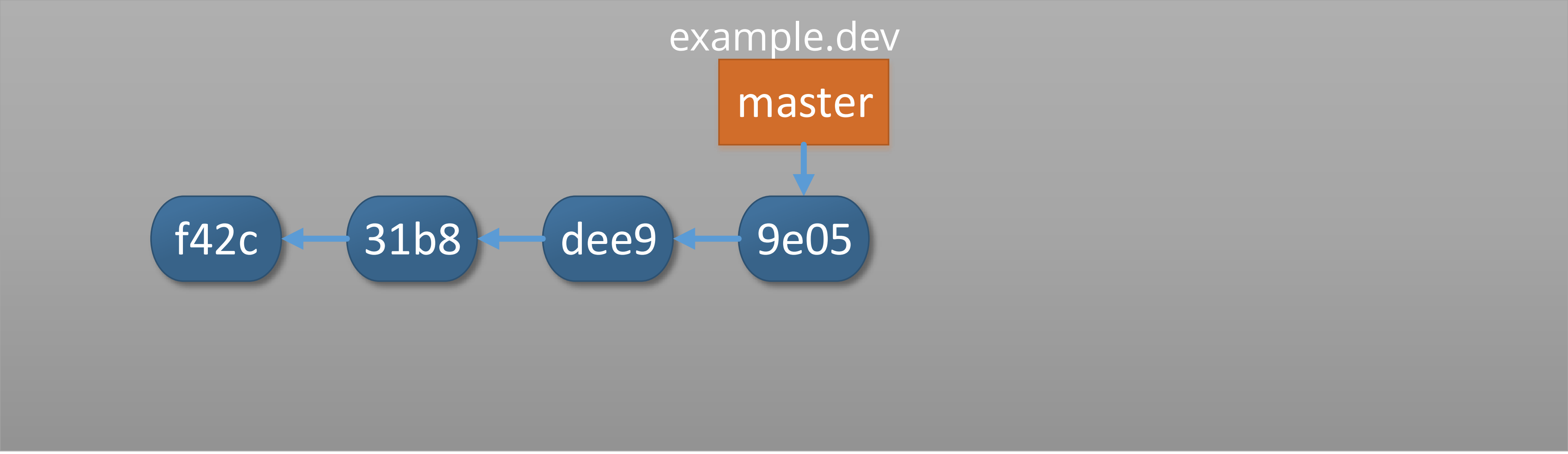
f42c

31b8

dee9

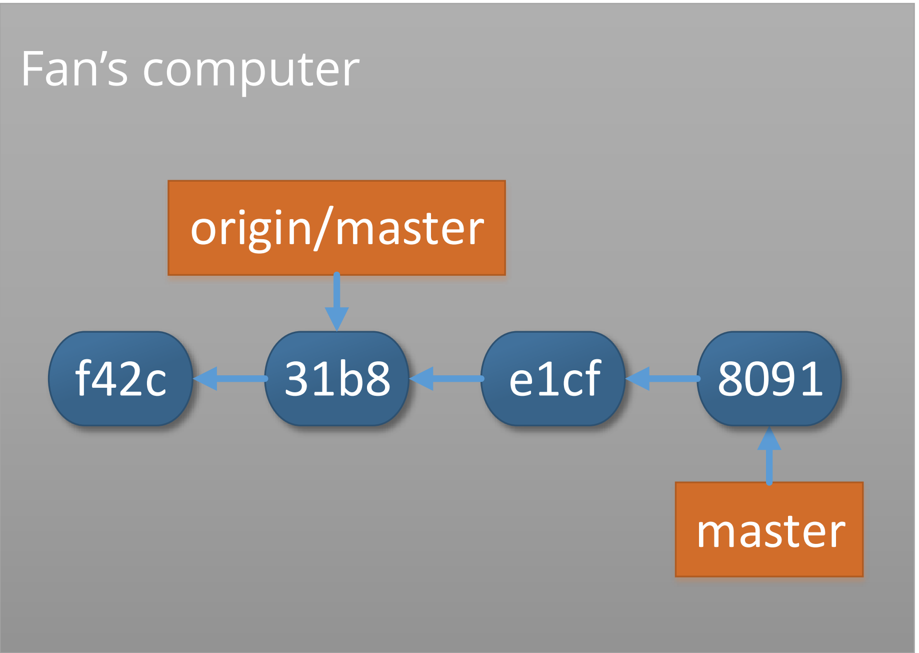
9e05

master

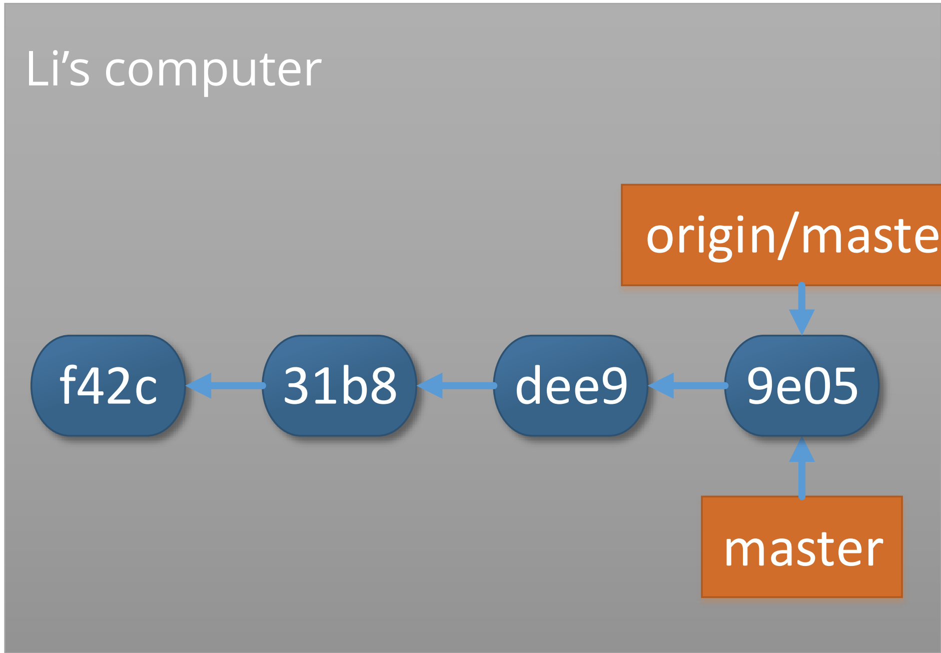


example.dev

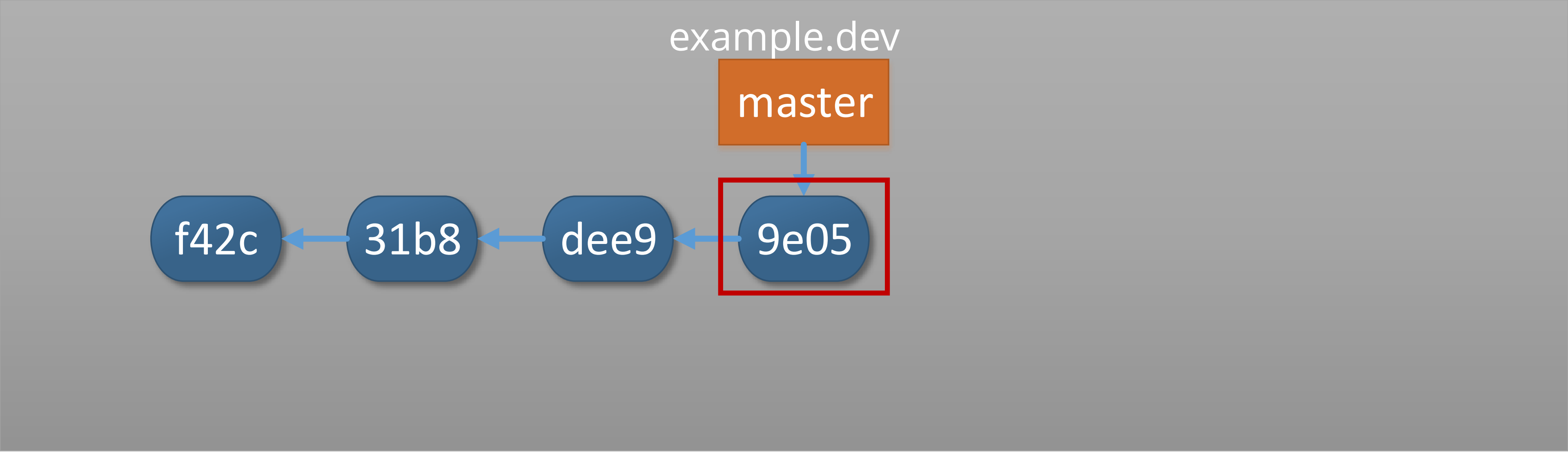
git push origin master



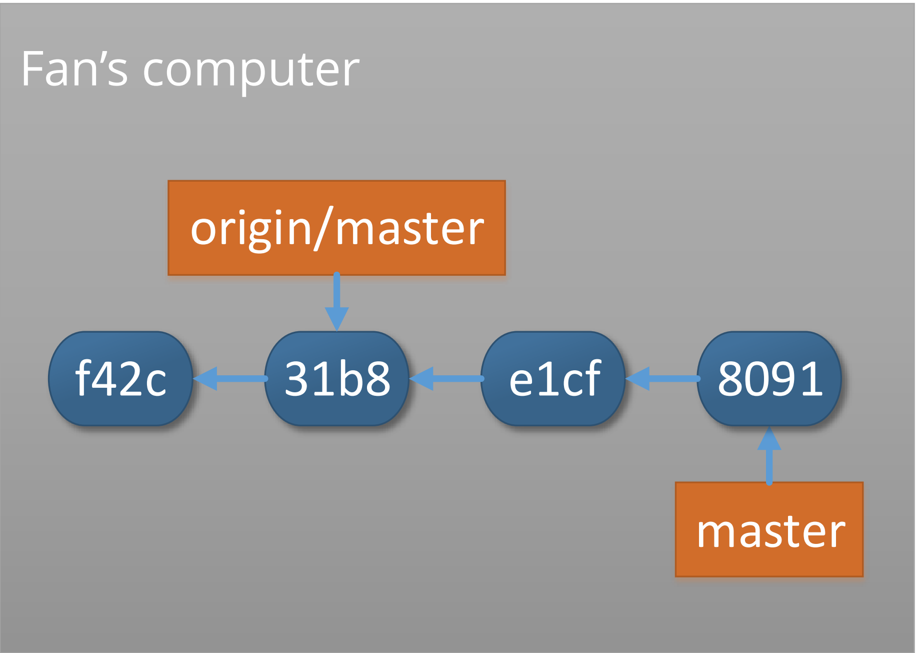
Fan's computer

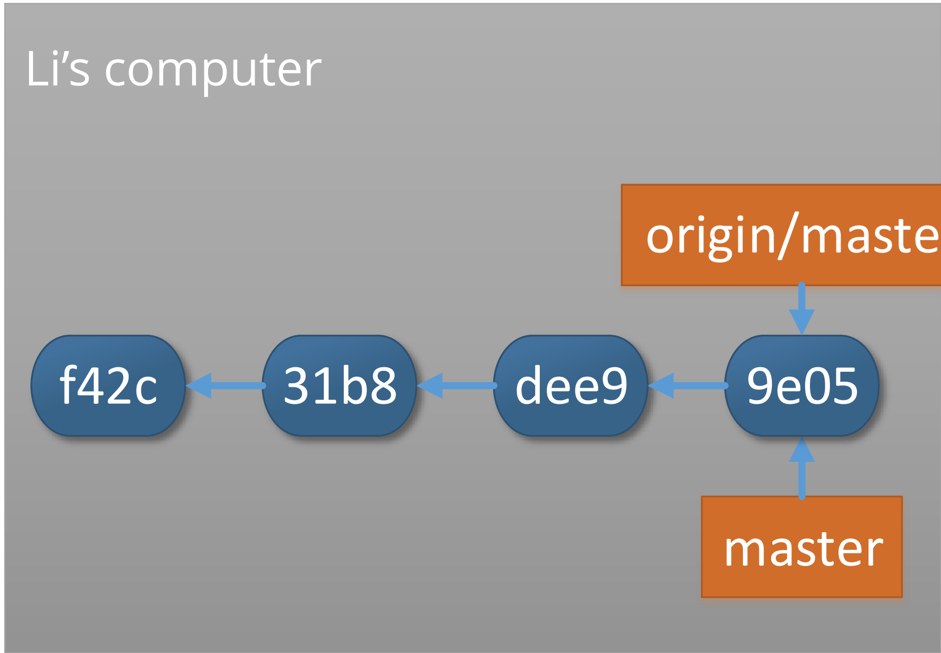


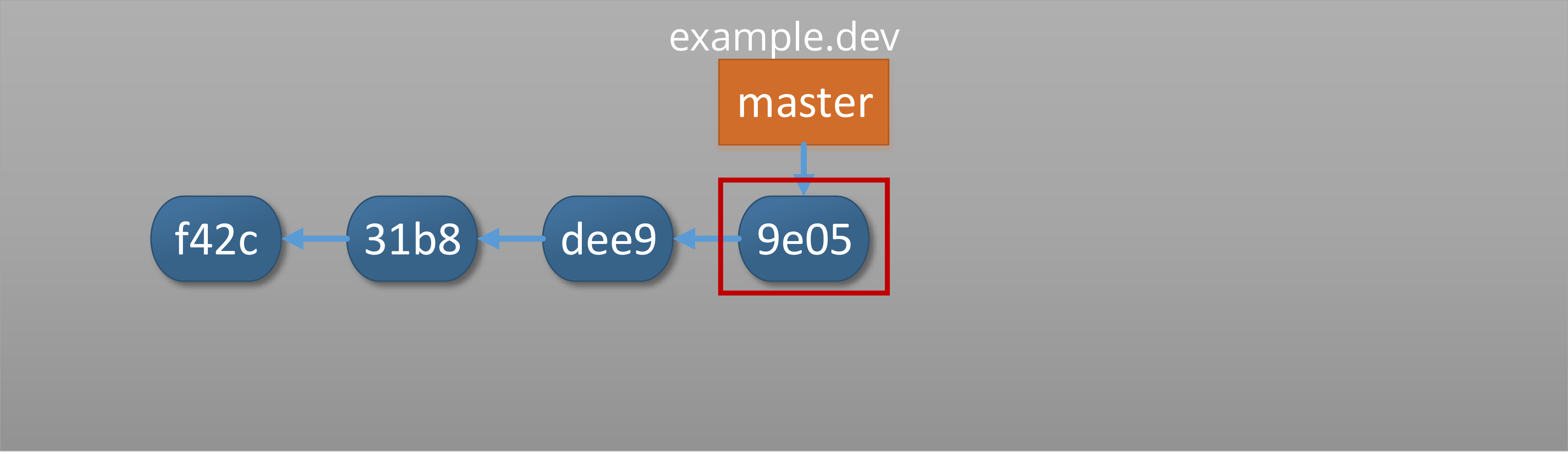
Li's computer



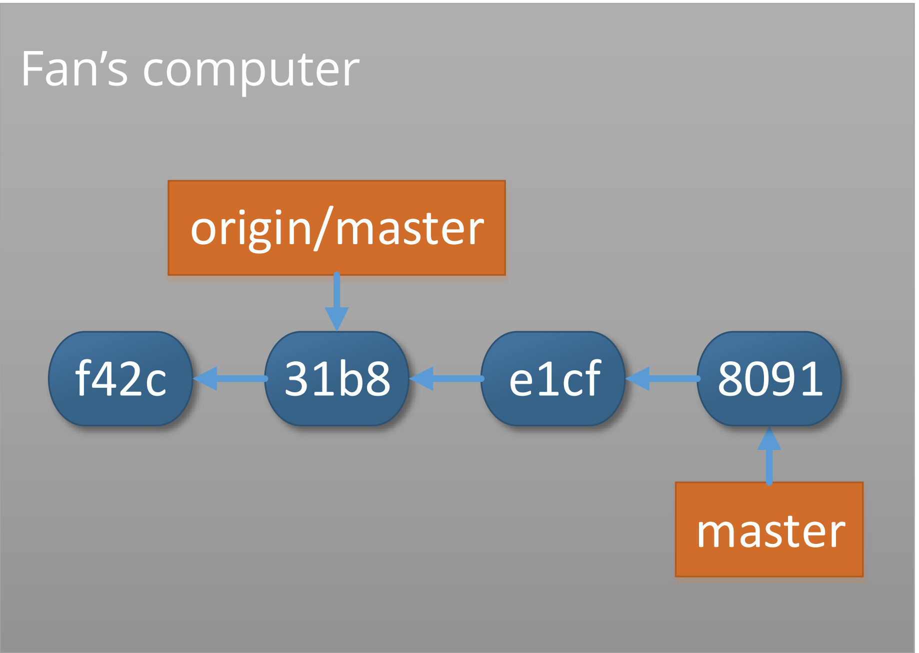
```
git push origin master
```

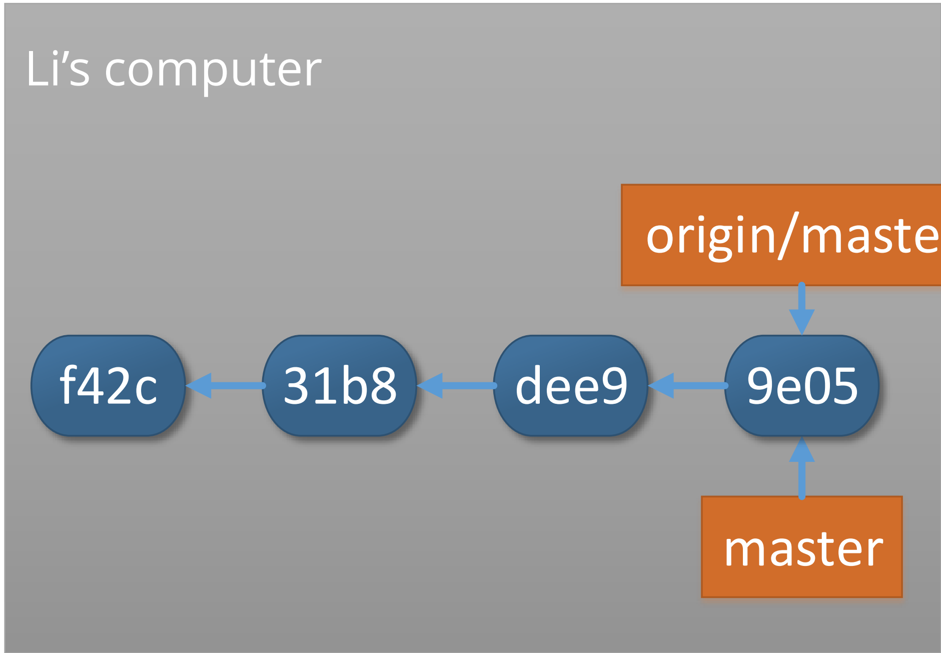


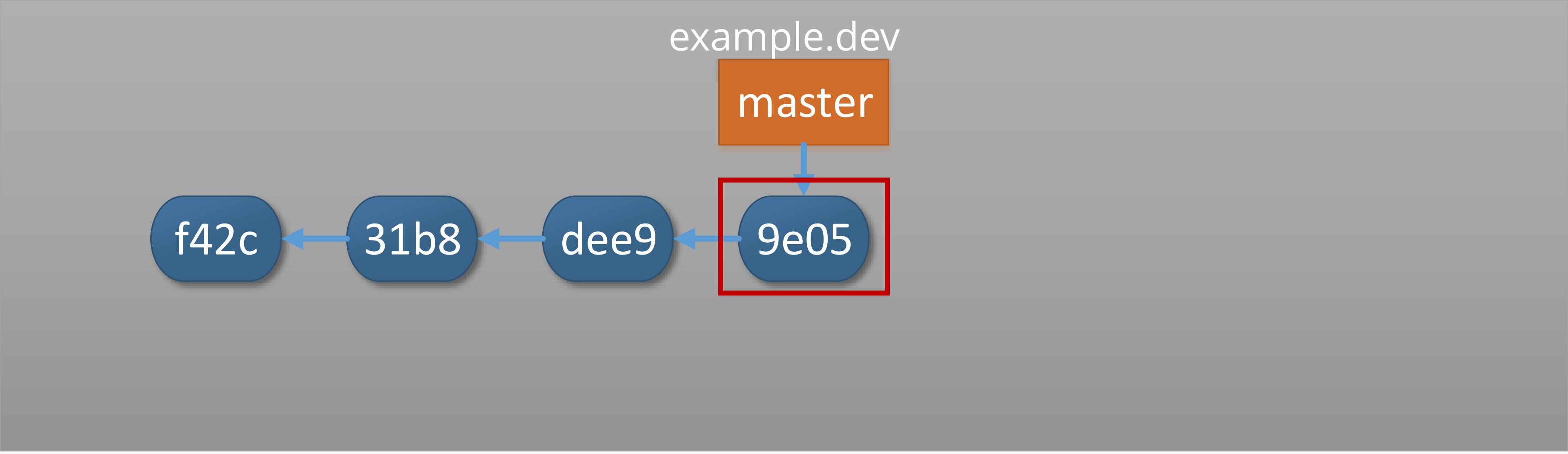




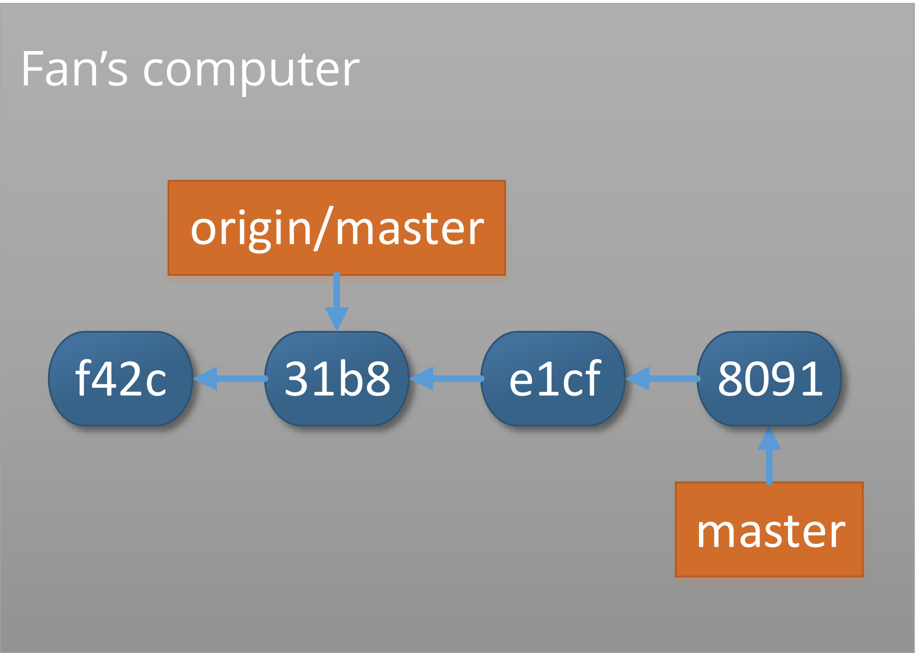
不在“server”上解决冲突

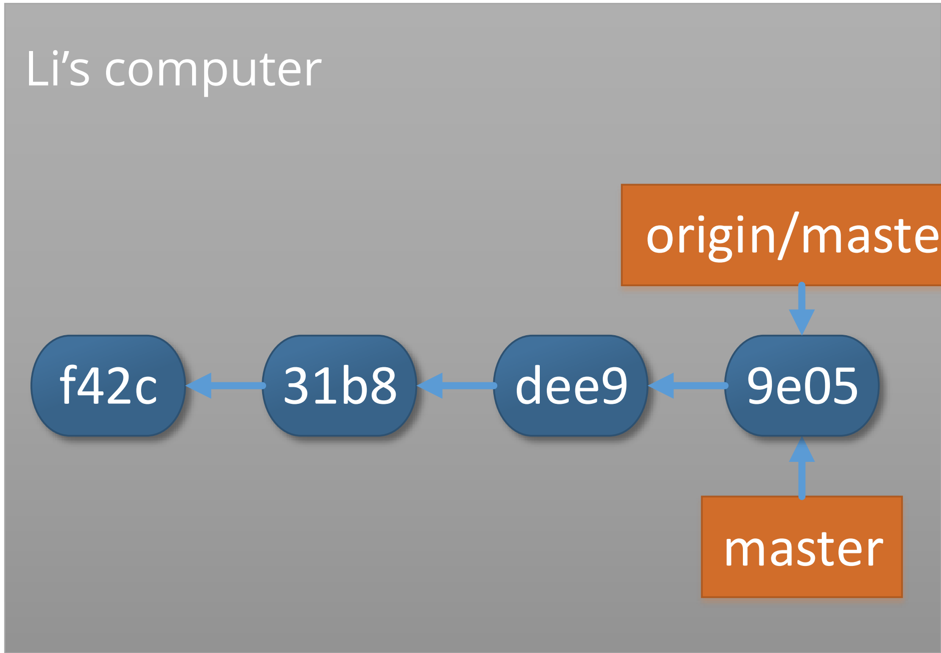


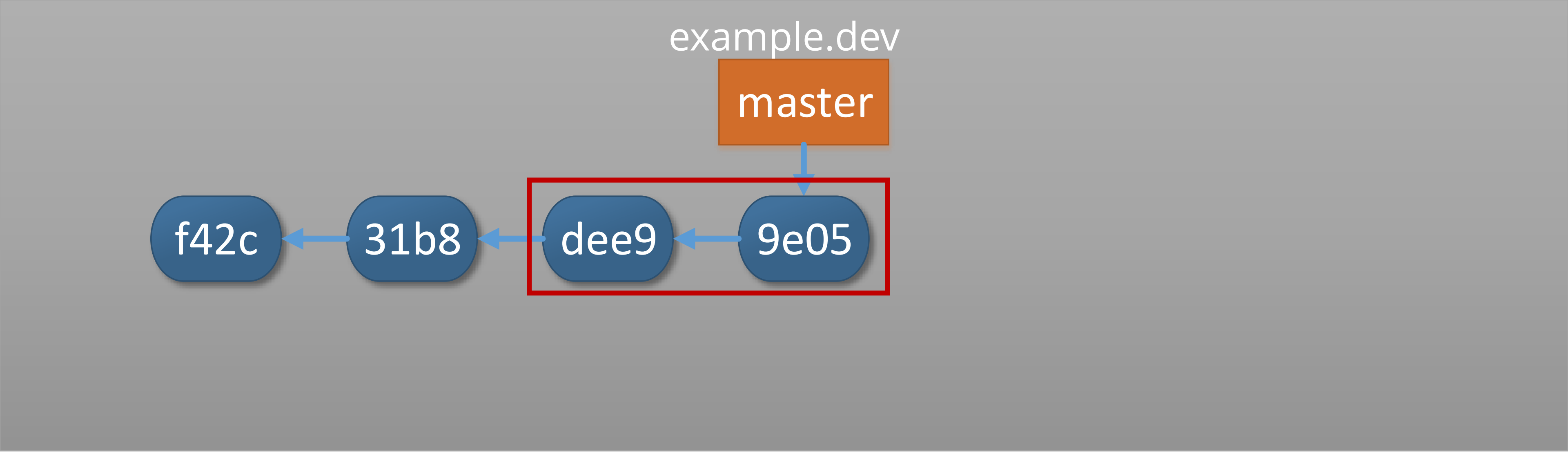



example.dev
master
f42c
31b8
dee9
9e05

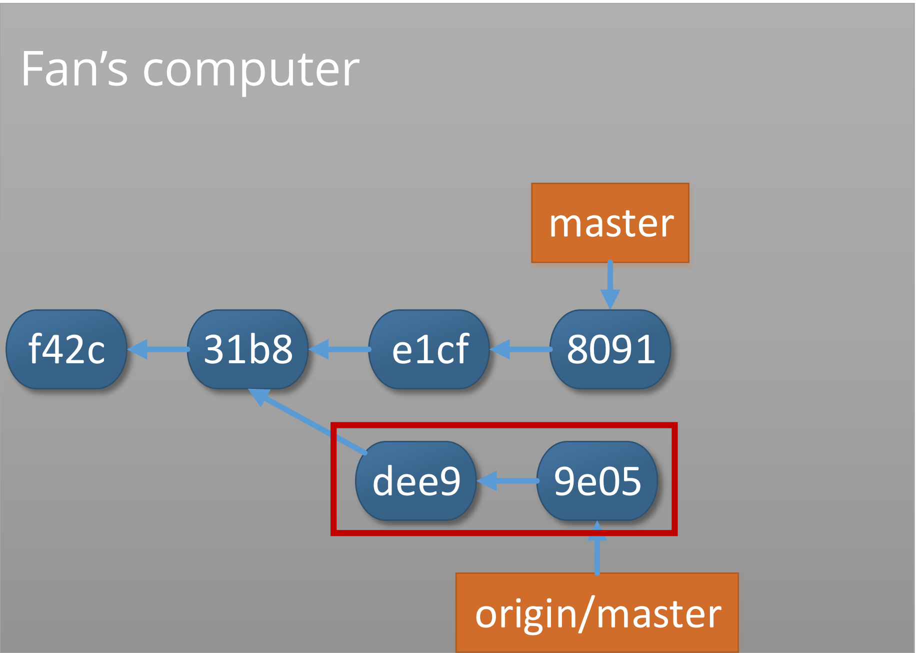
git fetch

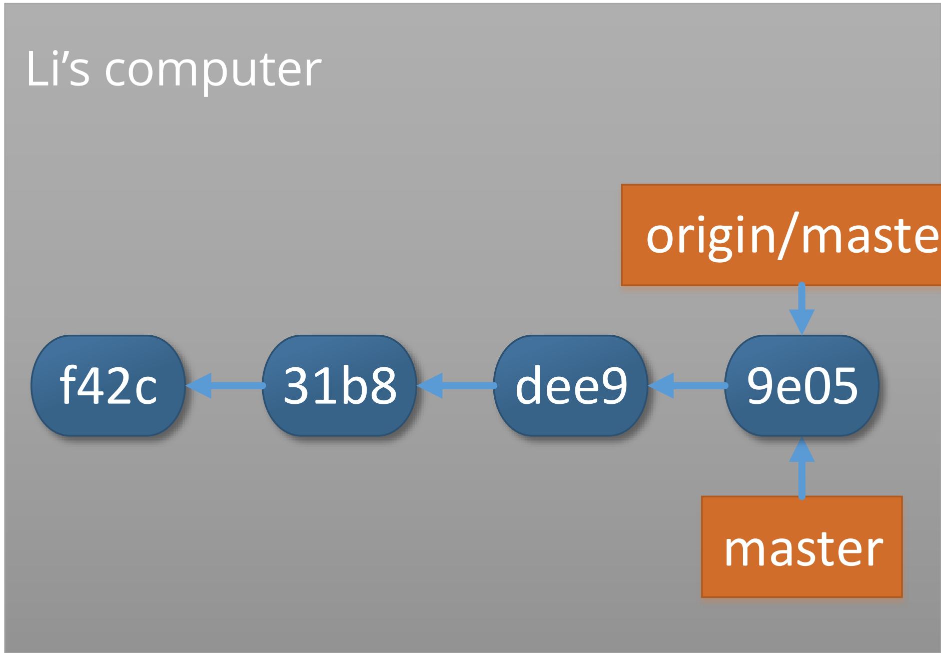

Fan's computer
origin/master
f42c
31b8
e1cf
8091
master

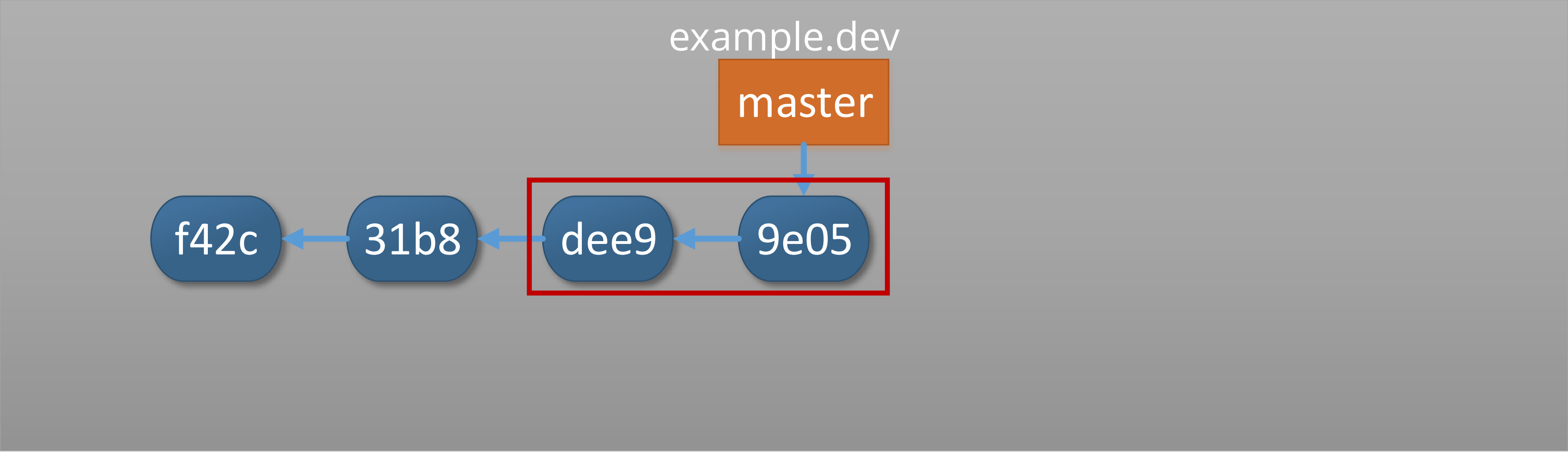

Li's computer
origin/maste
f42c
31b8
dee9
9e05
master


example.dev
master
f42c
31b8
dee9
9e05

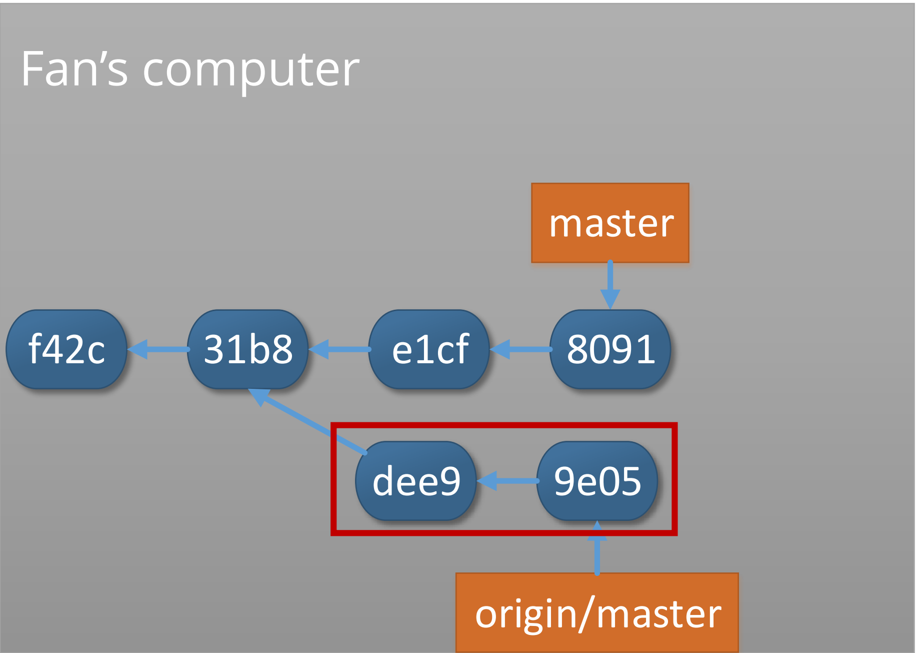
git fetch

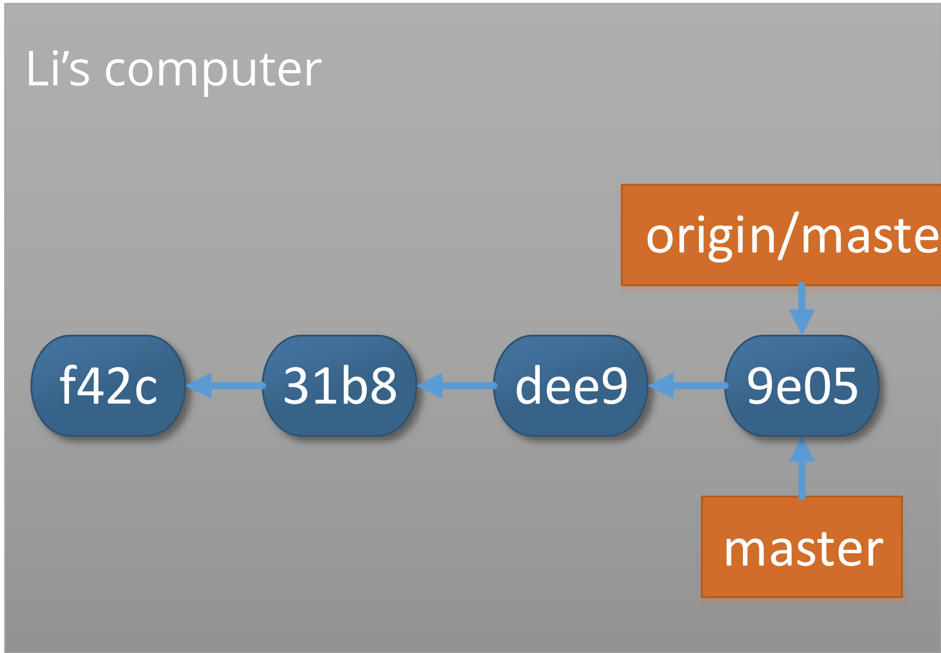

Fan's computer
master
f42c
31b8
e1cf
8091
dee9
9e05
origin/master

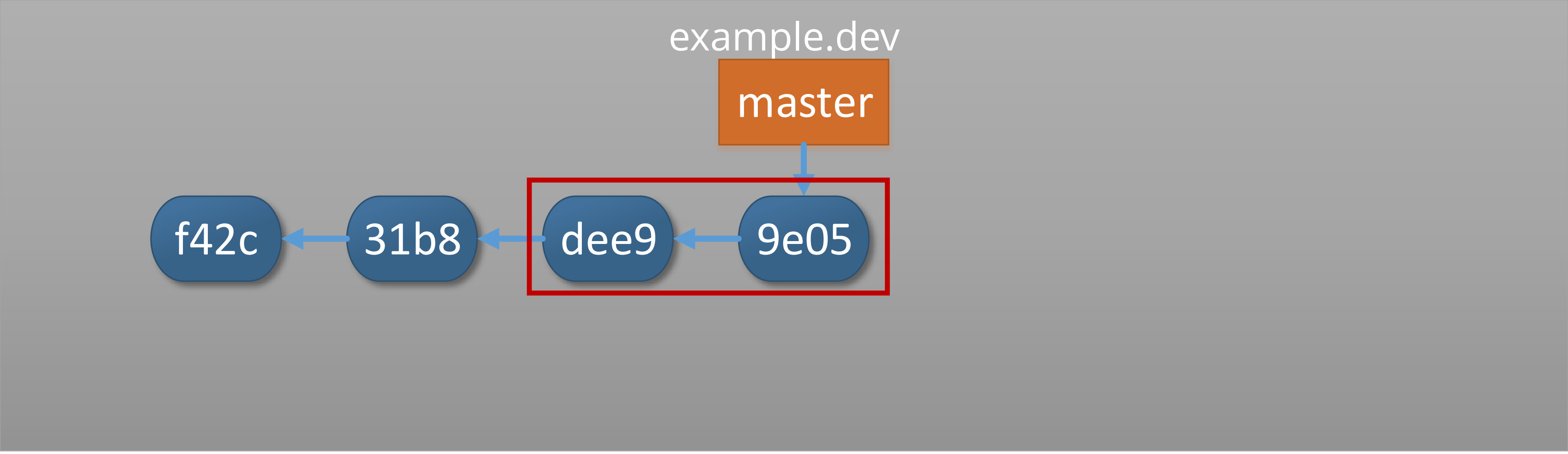

Li's computer
origin/maste
f42c
31b8
dee9
9e05
master


example.dev
master
f42c
31b8
dee9
9e05

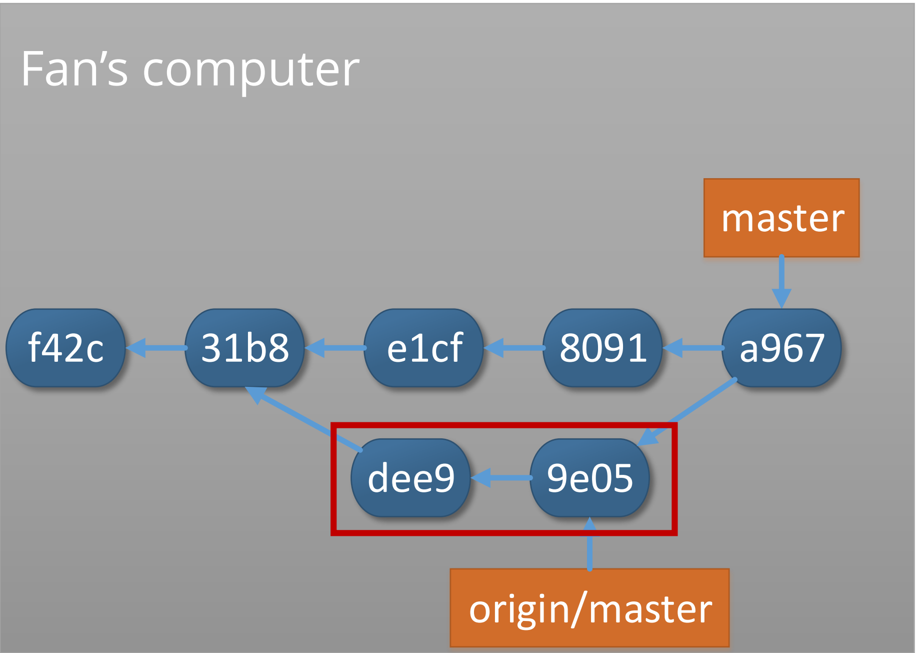
git merge origin/master

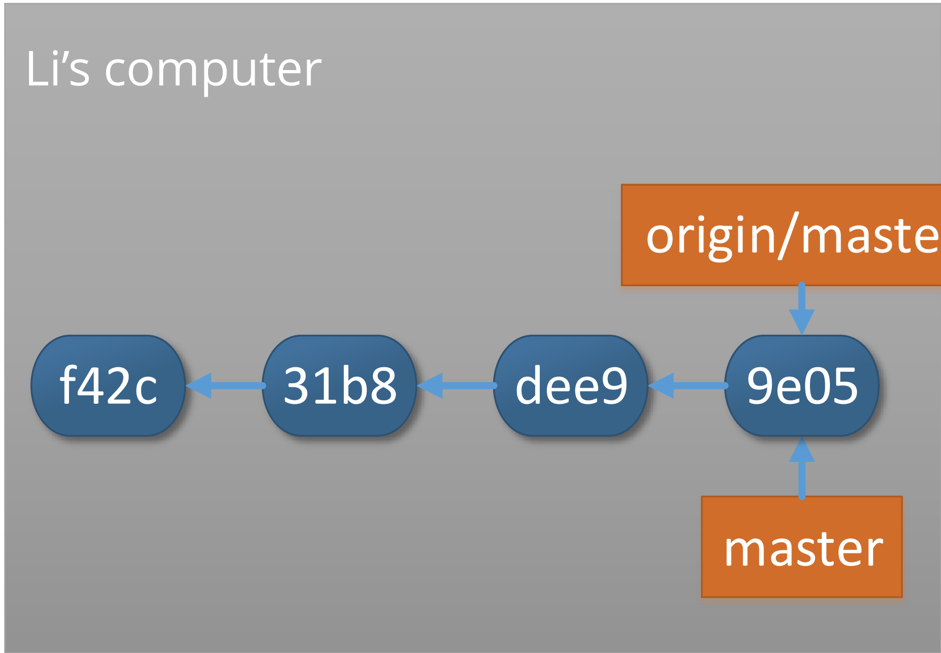

Fan's computer
master
f42c
31b8
e1cf
8091
dee9
9e05
origin/master

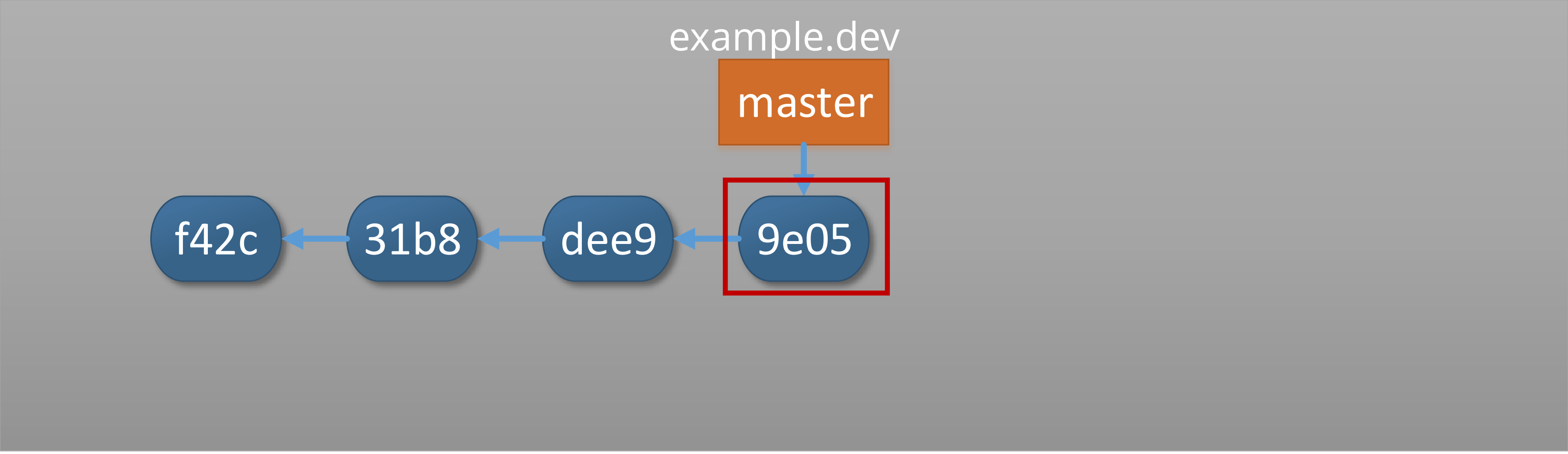

Li's computer
origin/maste
f42c
31b8
dee9
9e05
master



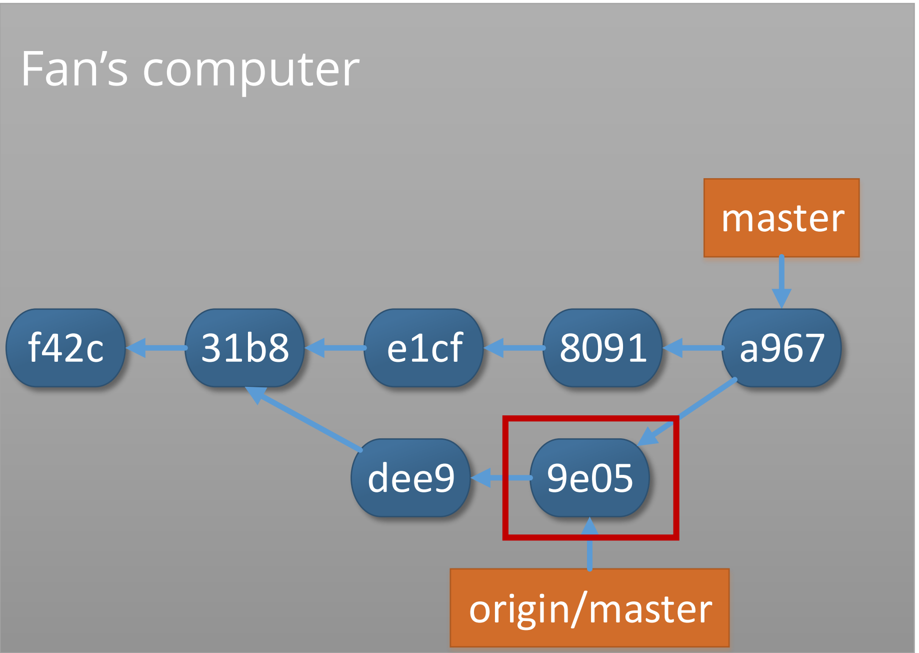
git merge origin/master

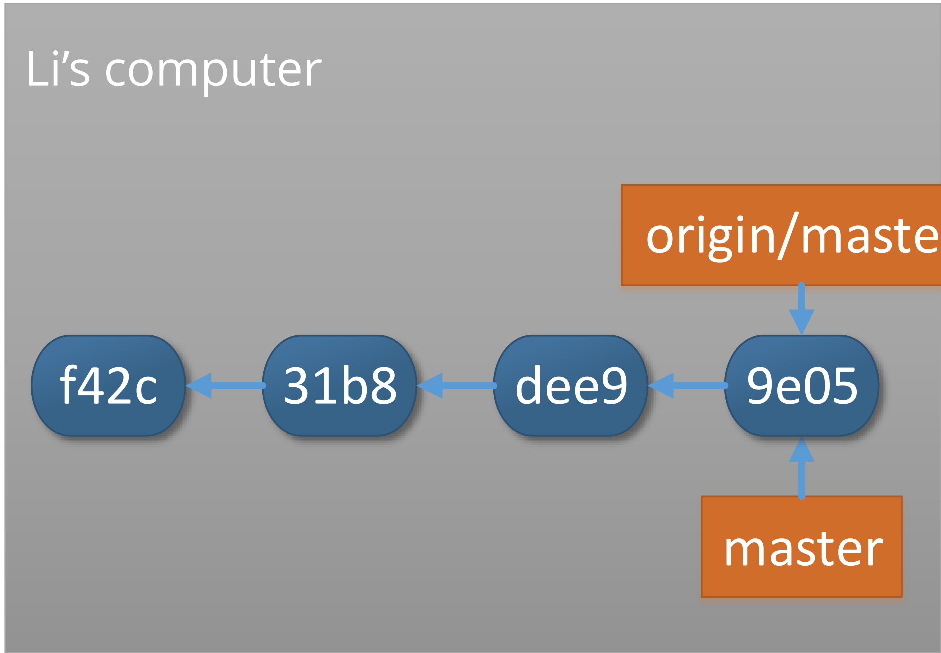


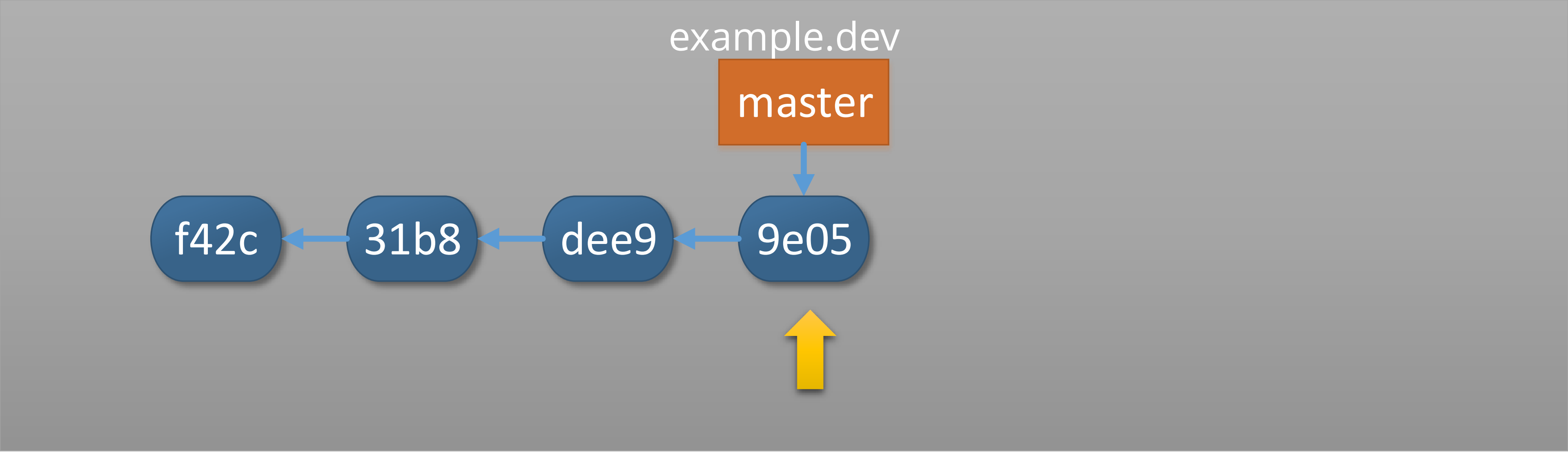




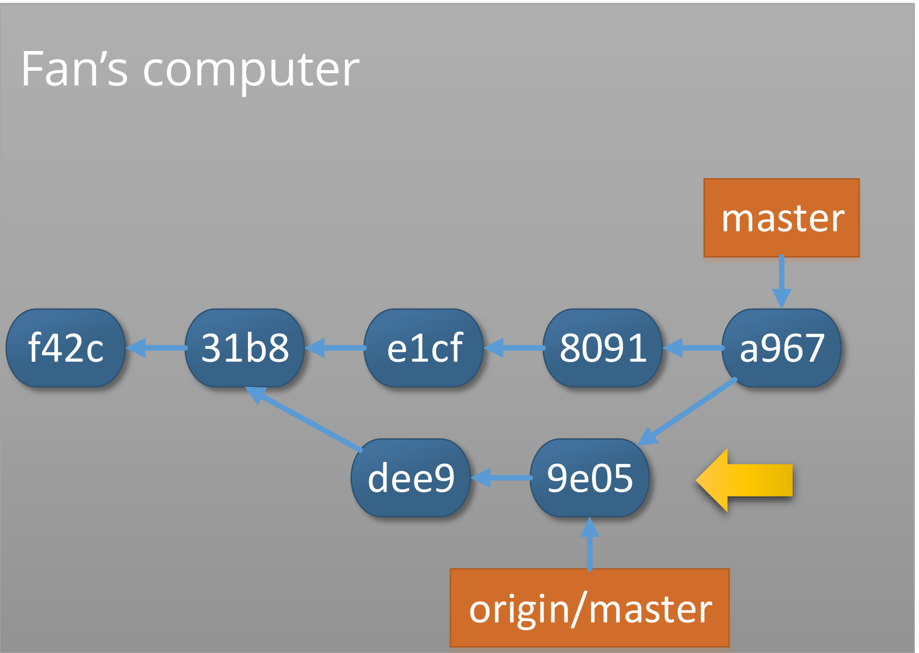
```
git push origin master
```

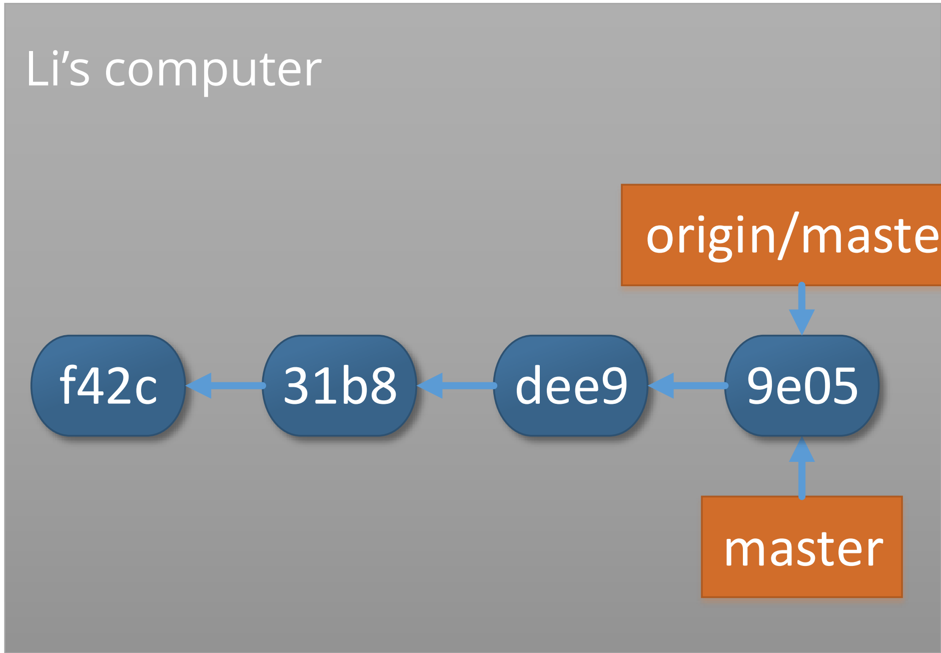


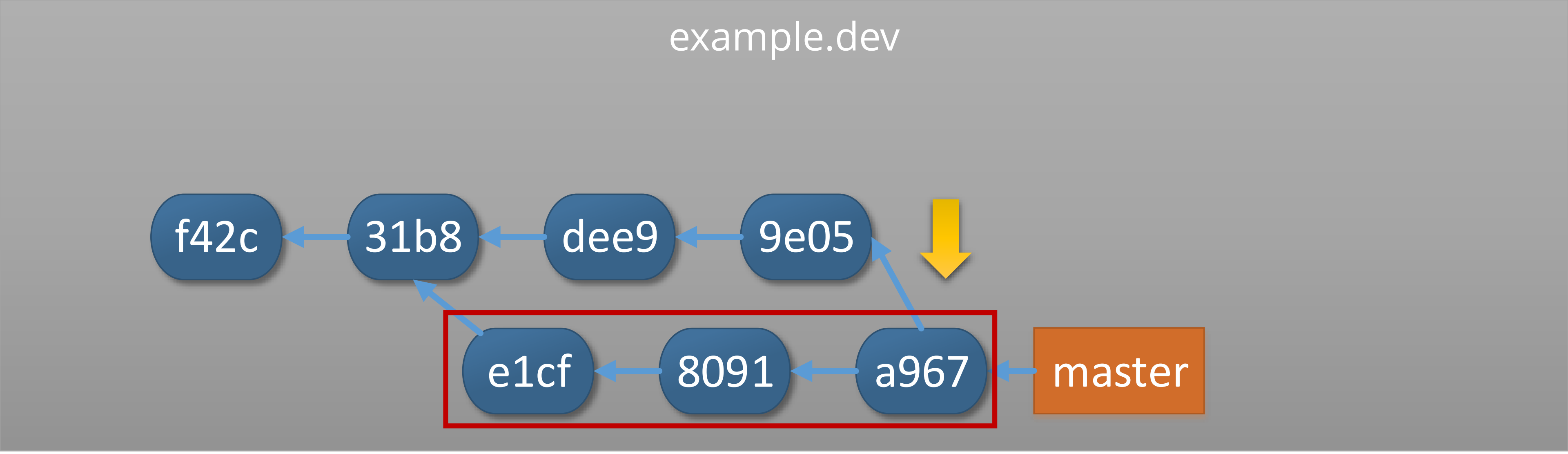




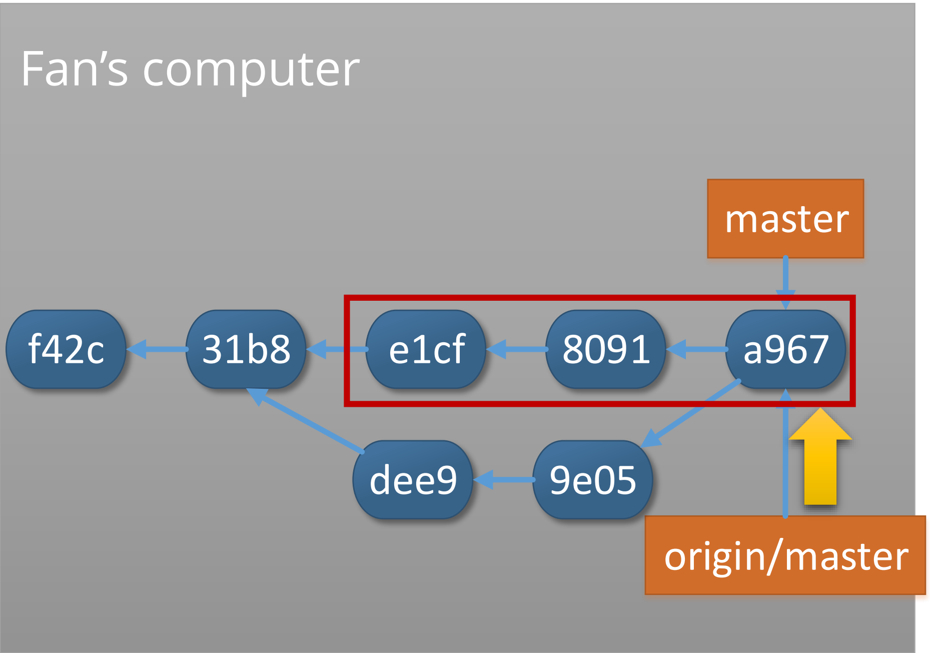
```
git push origin master
```

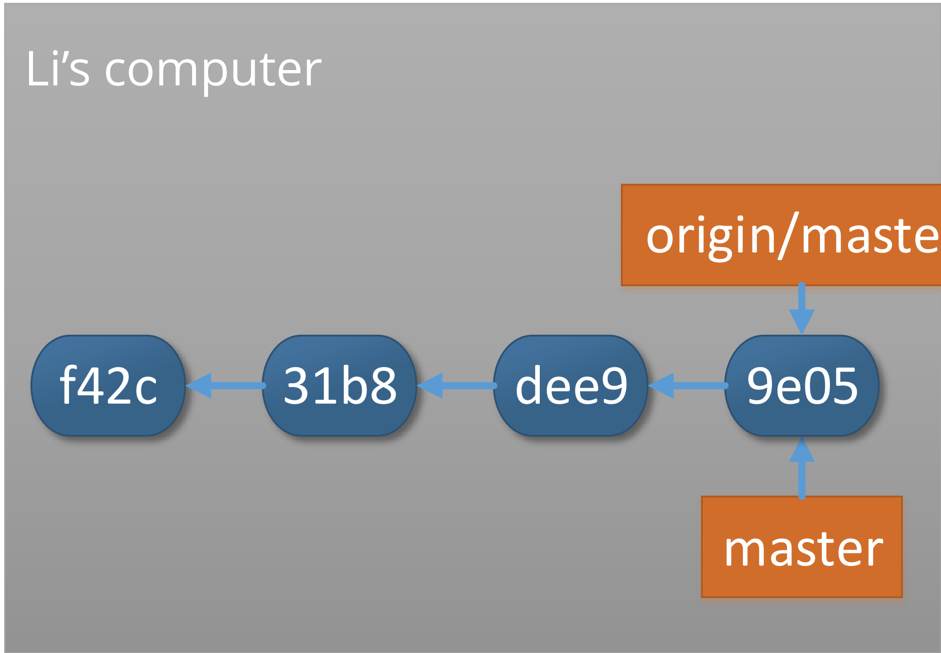


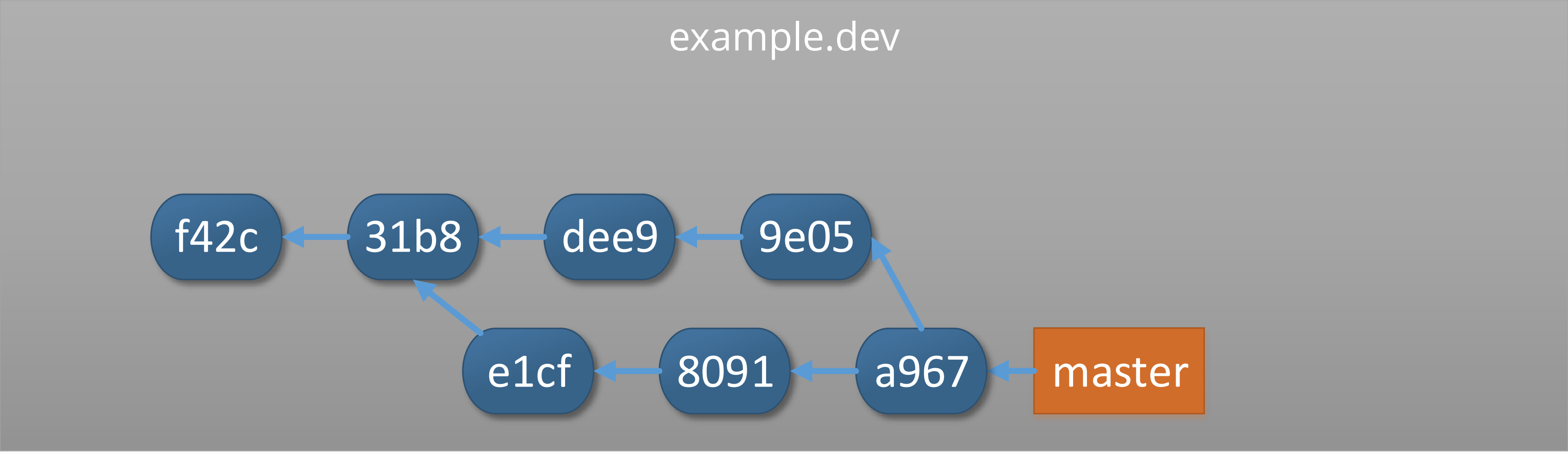




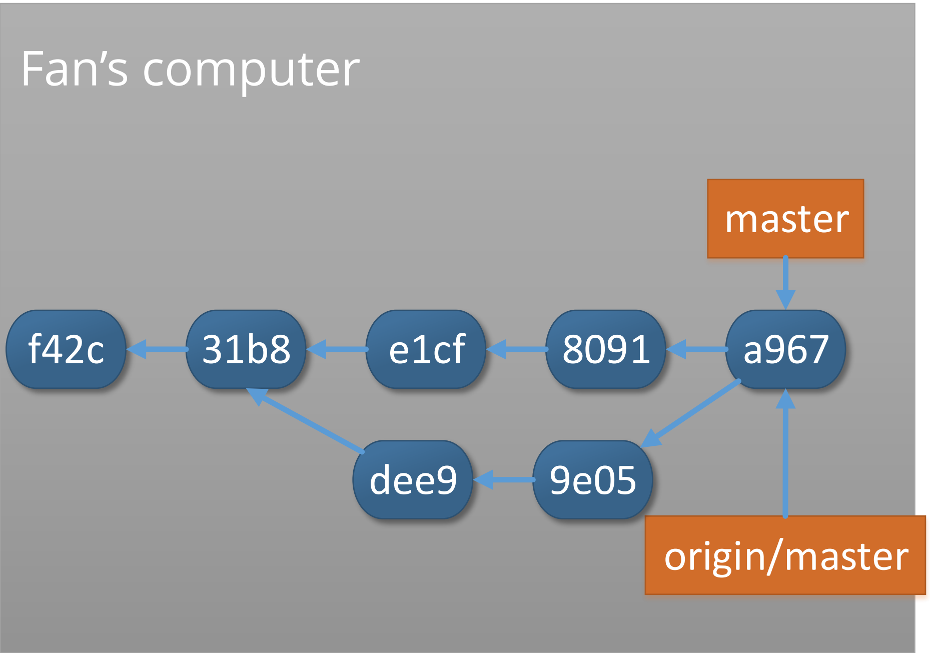
```
git push origin master
```

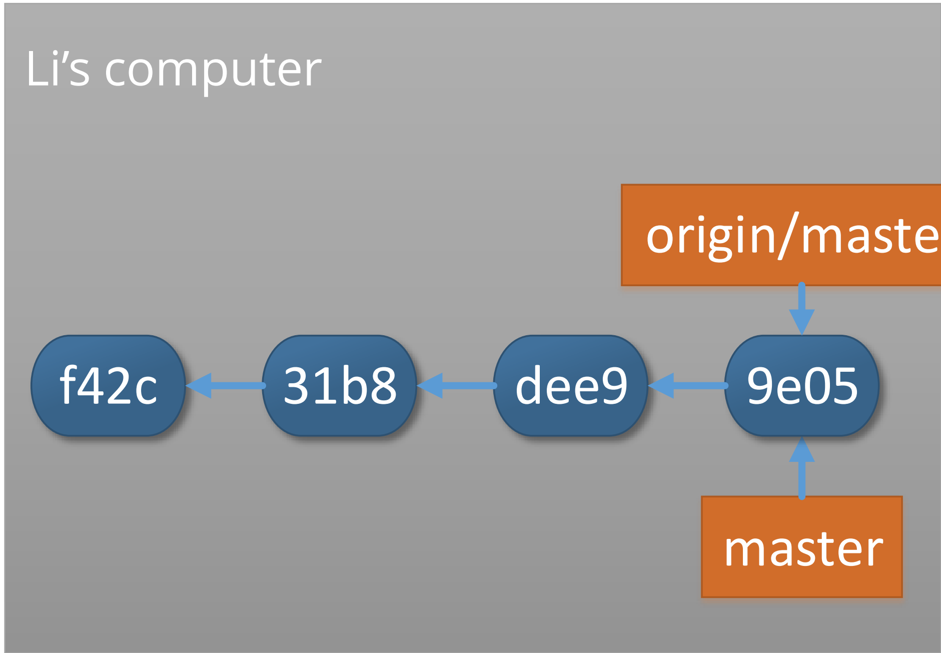




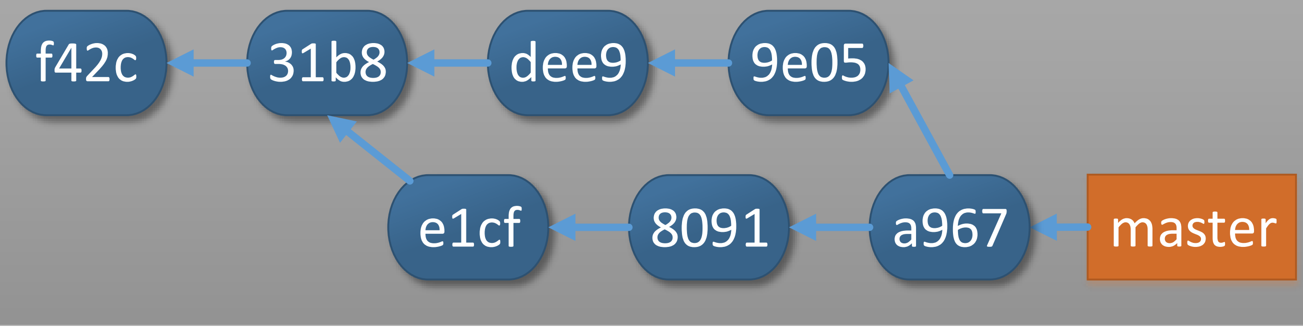


git checkout –b tang 31b8 ; git commit ;git commit



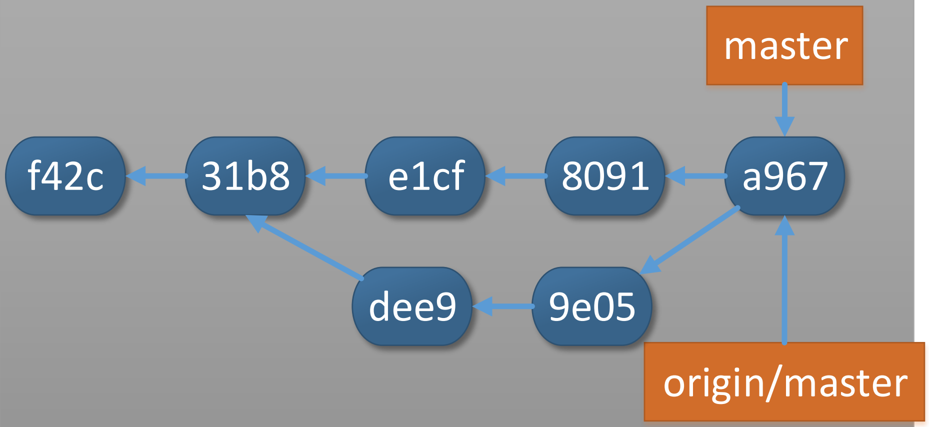


example.dev

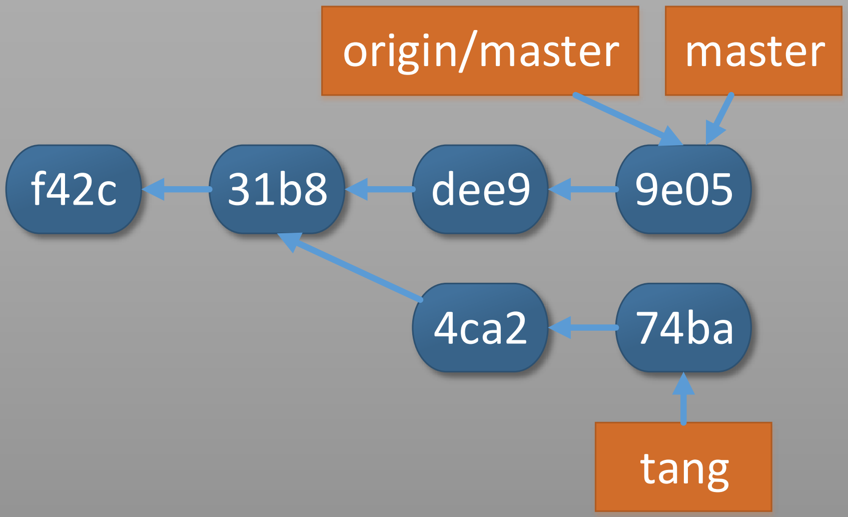


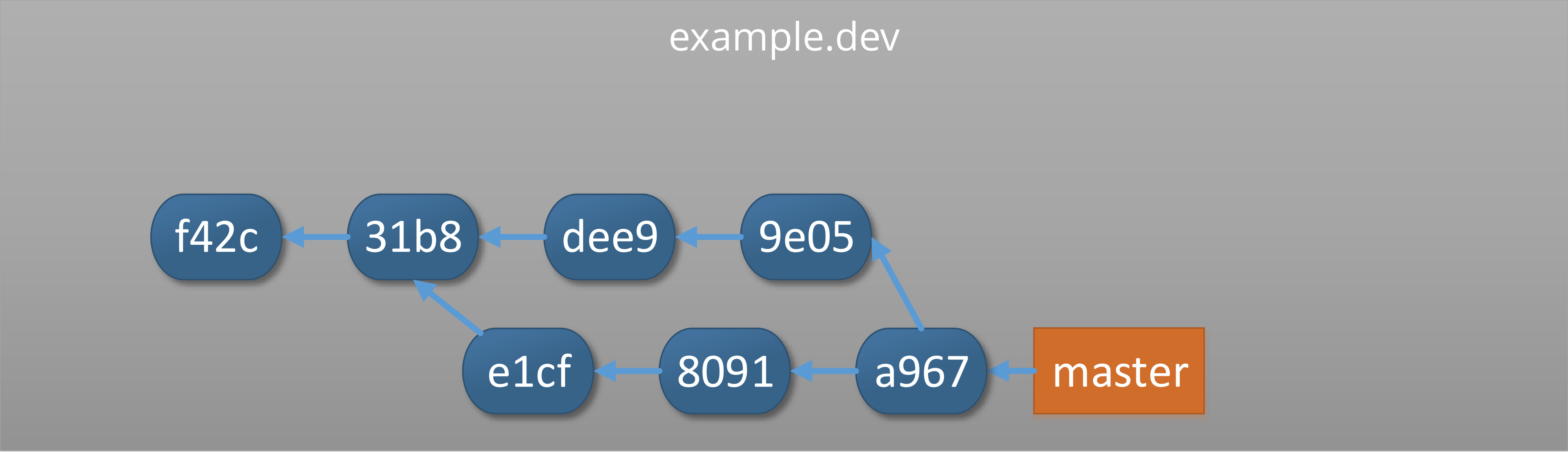
git checkout –b iss37 31b8 ; git commit ;git commit

Fan’s computer

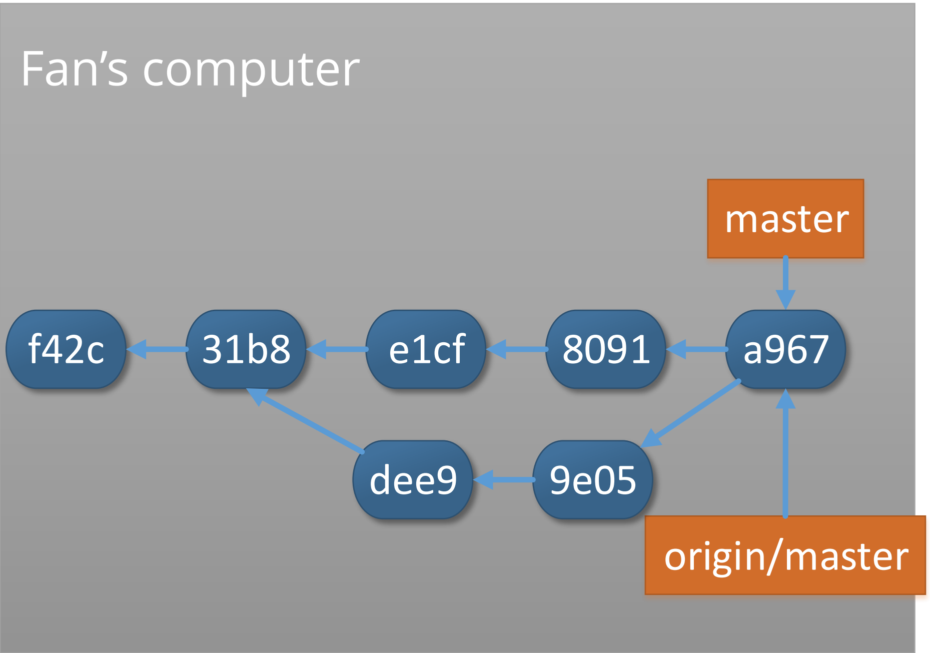


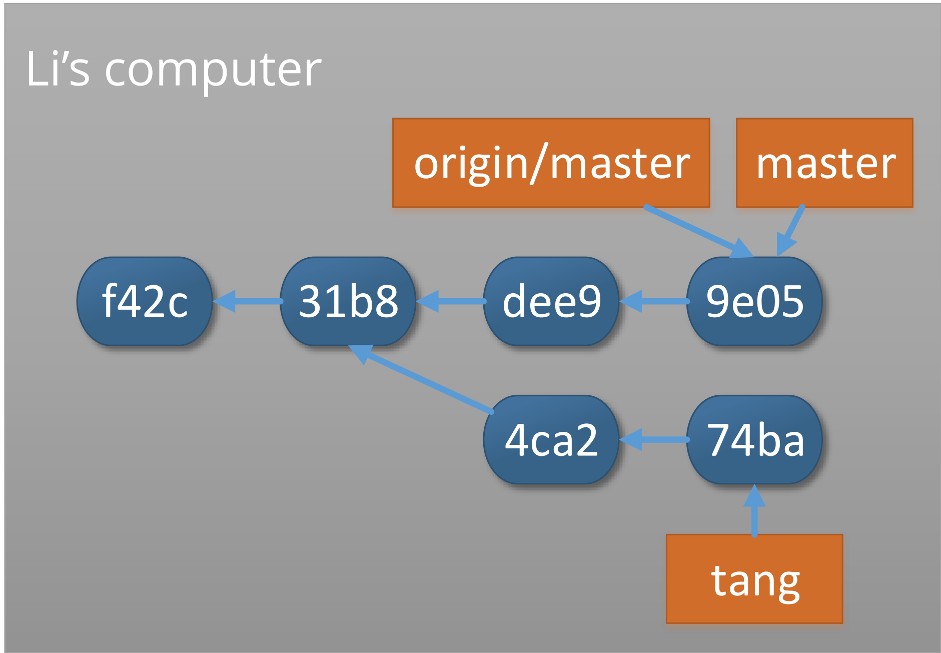
Li’s computer

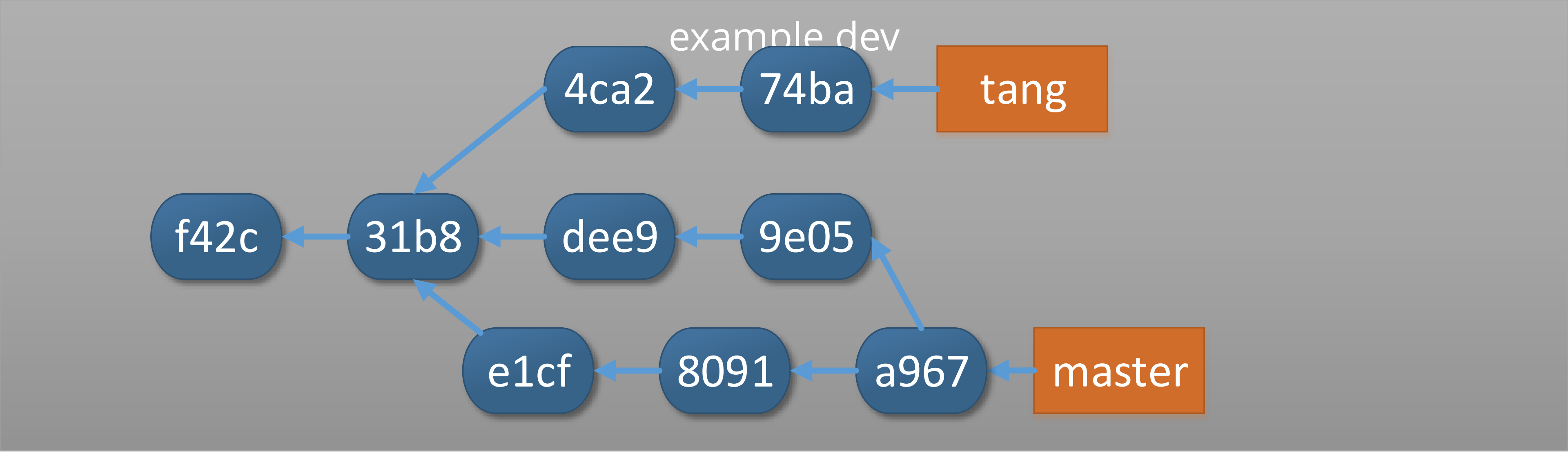




```
git push origin tang
```

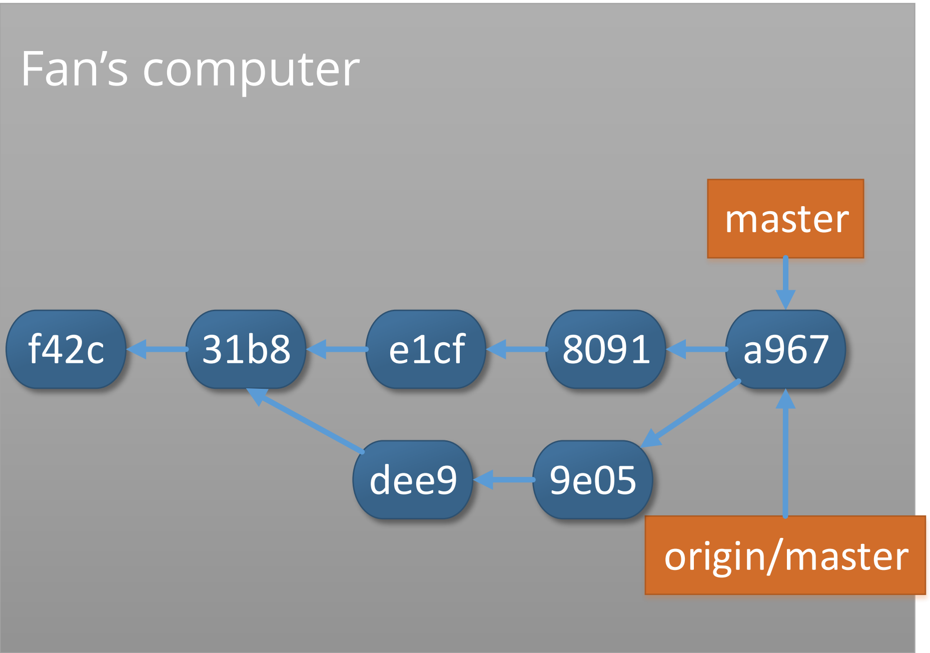




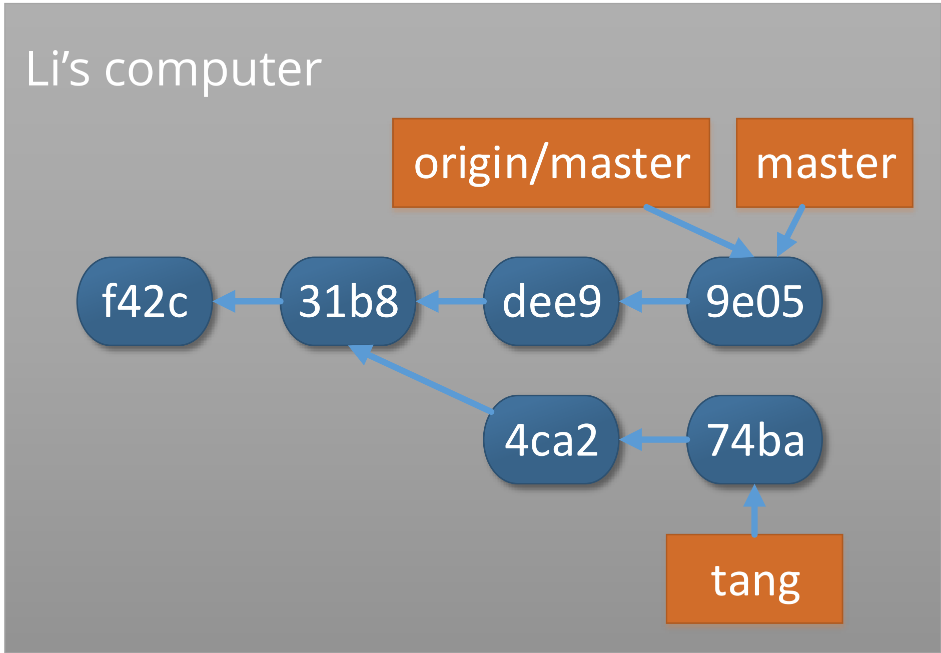


example dev

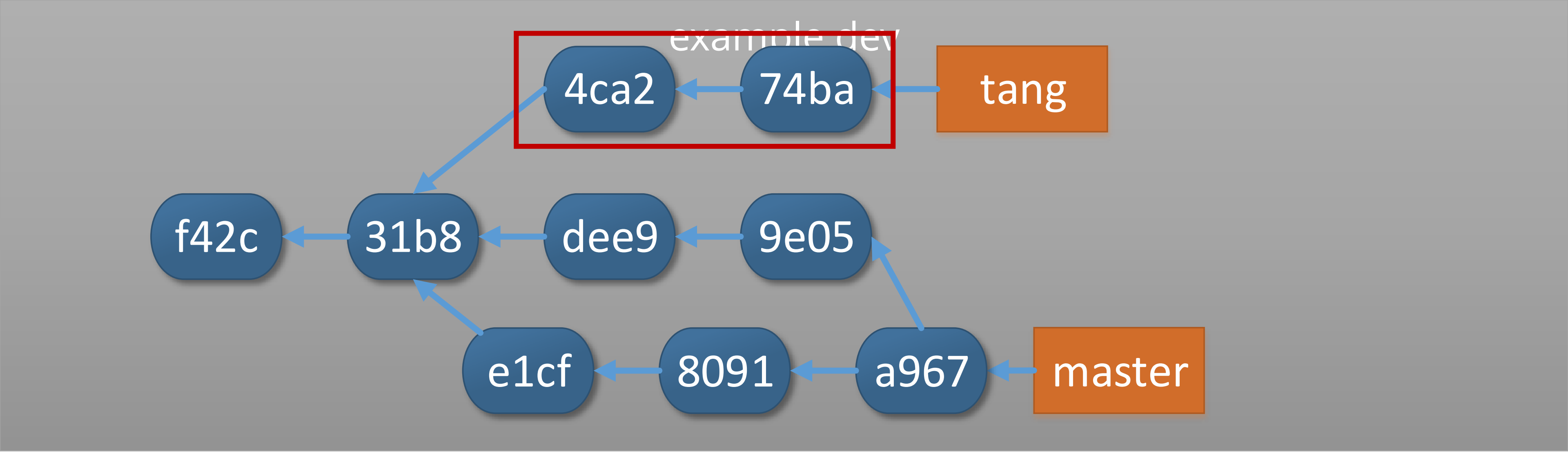
git push origin tang



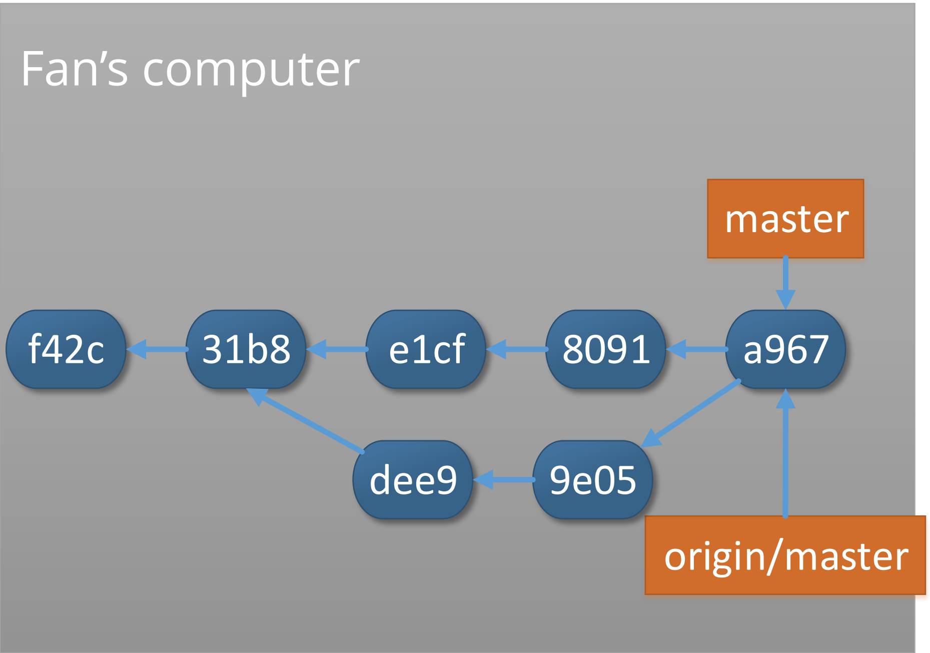
Fan's computer

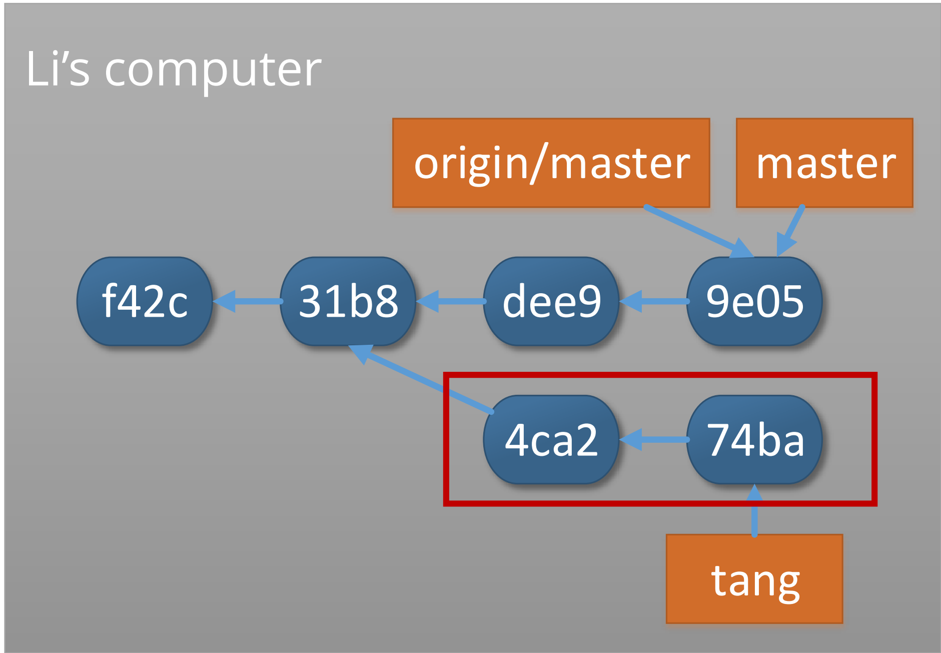


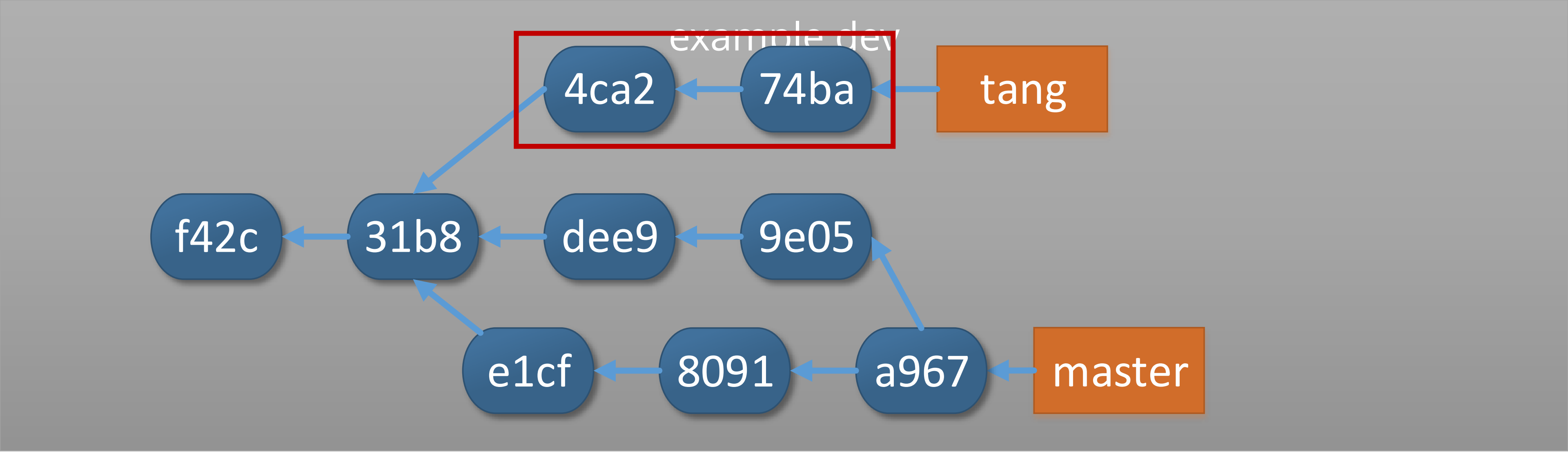
Li's computer



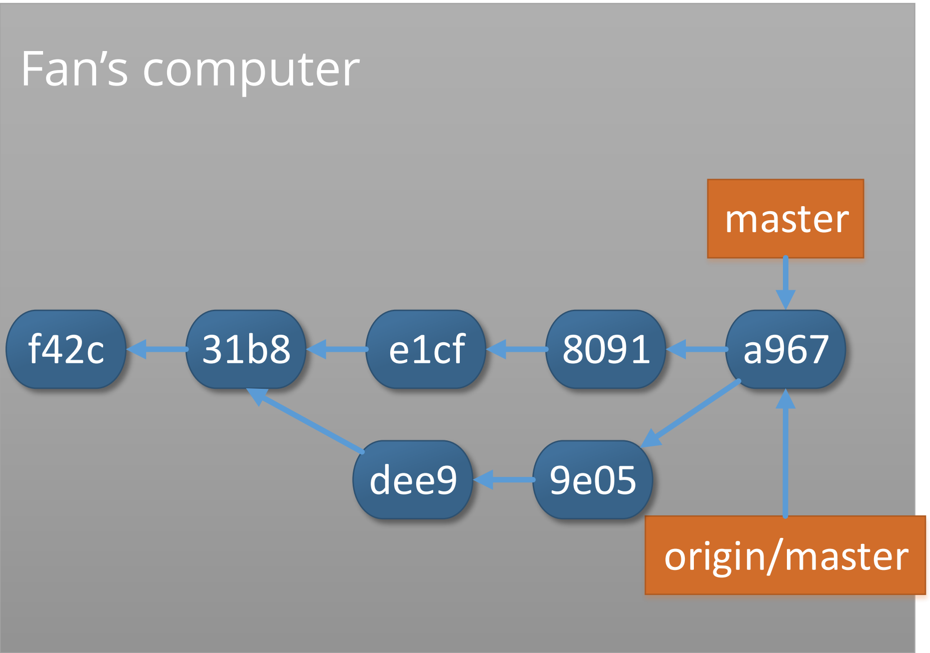
git push origin tang

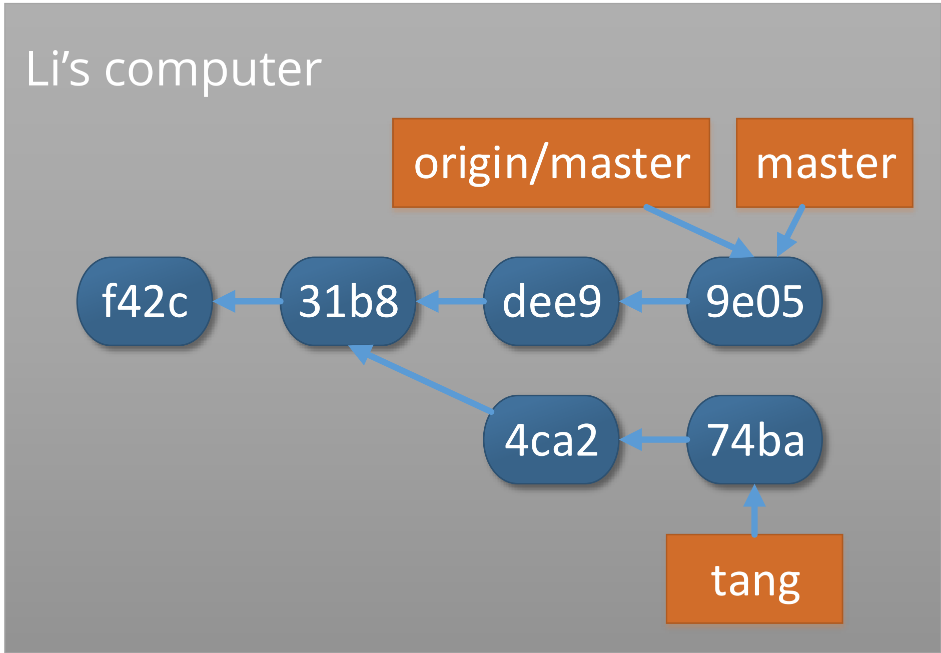




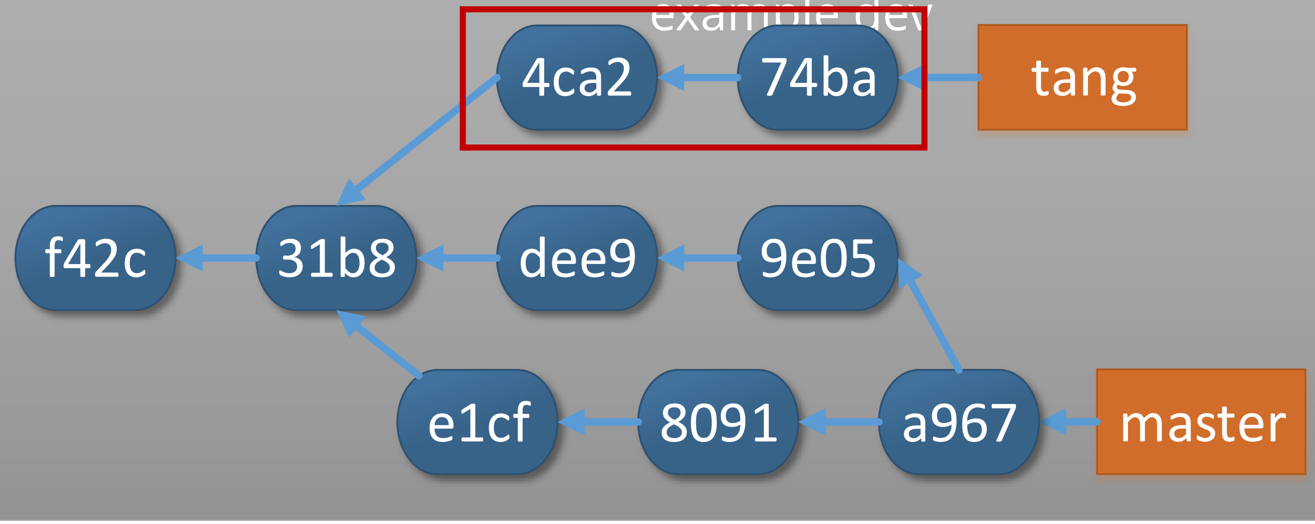


git fetch


Fan’s computer

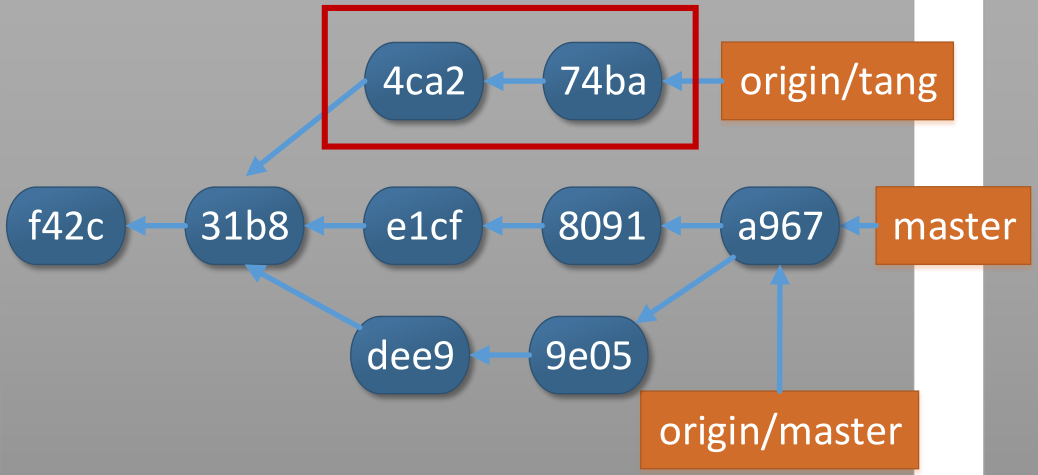

Li’s computer

example dev

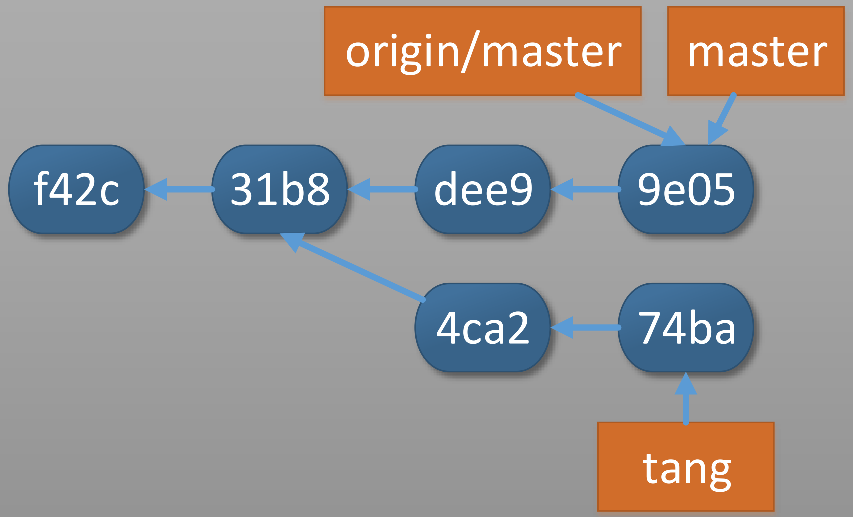


git fetch

Fan's computer



Li's computer



git 的远程操作

推送数据	git push
抓取并合并	git pull
remote	git remote
抓取数据	git fetch

pull = fetch + merge

git push origin <local>:<server>

FAQ: 如何安装 git

Windows:

msysGit + (Posh Git via Powershell)

Github for Windows

Atlassian SourceTree

Linux 发布版一般自带

FAQ：如何查看差异

git diff	没有缓存的变化
git diff –cached	缓存了的变化
git diff HEAD	与上次提交比的变化
git diff <branchname>	与分支的差异
git diff <branchname>...<branchname>	将会引入的变化

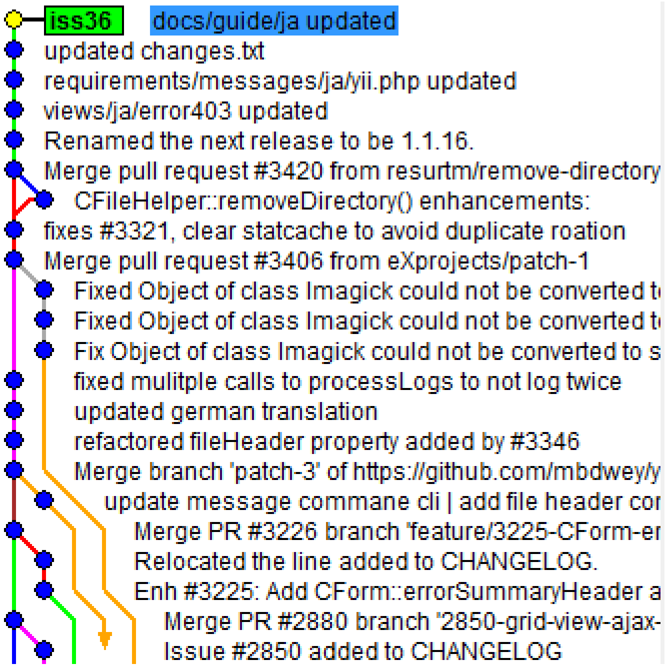
FAQ：如何查看提交历史

git log

eg: git log --pretty=oneline (--graph)

FAQ：查看分支合并情况

gitk



```
iss36  docs/guide/ja updated
updated changes.txt
requirements/messages/ja/yii.php updated
views/ja/error403 updated
Renamed the next release to be 1.1.16.
Merge pull request #3420 from resurtm/remove-directory
  CFileHelper::removeDirectory() enhancements:
fixes #3321, clear statcache to avoid duplicate roation
Merge pull request #3406 from eXprojects/patch-1
  Fixed Object of class Imagick could not be converted t
  Fixed Object of class Imagick could not be converted t
  Fix Object of class Imagick could not be converted to s
  fixed mulitple calls to processLogs to not log twice
  updated german translation
  refactored fileHeader property added by #3346
  Merge branch 'patch-3' of https://github.com/mbdwey/y
    update message commane cli | add file header cor
      Merge PR #3226 branch 'feature/3225-CForm-er
      Relocated the line added to CHANGELOG.
      Enh #3225: Add CForm::errorSummaryHeader a
        Merge PR #2880 branch '2850-grid-view-ajax-
        Issue #2850 added to CHANGELOG
```

git log --pretty=oneline (--graph)

FAQ：如何撤销更改

第一种情况：

修改最后一次提交

```
git commit --amend
```

FAQ：如何撤销更改

第二种情况：

撤销已经暂存了的文件

```
git reset HEAD <filename>
```

FAQ: 如何撤销更改

第三种情况：

取消对文件的修改

```
git checkout -- < file (.) >
```

FAQ：如何撤销更改

第四种情况：

取消提交（本地仓库回退）

```
git reset –(soft|mixed|hard)
```

```
git revert
```

谨慎使用！

resources

- 《Pro Git》
  - Scott Chacon （在GitHub工作）写的一本对Git的详细介绍，从使用到底层原理，这个intro大部分来自《Pro Git》,但书中对命令的介绍不是特别详细
  - 这本书使用了 CC BY-NC-SA 3.0 协议发布，所以是免费的
  - 而且被翻译成了多种语言，包括中文，还有epub格式

  - 官方网址： http://git-scm.com/book
  - GitHub地址： https://github.com/progit/progit

resources

- 《Introduction to Git with Scott Chacon of GitHub》
  - 仍然是来自 Scott Chacon，是一个公开课的视频，详细介绍了Git的实现原理，和基本使用，这个intro基本是把他的视频重新做了一遍，人很帅，有点像Eminem…
  - 大概一个半小时
  - 我这里有从YouTube上下载下来的，没有中文字幕

  - 官方网址：

  https://www.youtube.com/watch?v=ZDR433b0HJY

resources

- Git Reference
  - 对Git一个快速的了解，包括一些命令和Git的一些原理等等，还介绍了怎么用Git的思维进行思考
  - 官方网址： http://gitref.org/

resources

- GitHub help
  - GitHub 的官方帮助文档，非常棒
  - 官方网址：https://help.github.com/

resources

- GitHub .gitignore
    - GitHub上的各种.gitignore文件的实例，实际上在GitHub上新建一个repo的时候就会让你选择是否要添加一个这样的文件，并且可以根据你的repo的类型选择不同场景下合适的.gitignore文件
    - 官方网址：https://github.com/github/gitignore

Download Git

Windows:

msysGit 原生Git
http://msysgit.github.io/

Github for Windows GitHub 的GUI和Shell

http://windows.github.com/

Atlassian SourceTree 界面比较好看

http://www.sourcetreeapp.com/

Linux 发布版一般自带

在http://git-scm.com/ 上可找到更多可用的客户端

msysGit + PowerShell + Posh Git

Posh Git

因为Git的操作和命令都比较简单，所以使用图形界面反而显得麻烦。msysGit 作为原生的Git发布版，除了具有Git的基本功能之外，还融合了Git BASH，提供了一些简单的bash命令，如ls、touch、等等并且提供了一个简单的GUI界面（gitk），除此之外还能和Shell进行集成，配合Posh Git 和 Powershell，能够打造一个很舒服的使用环境

msysGit + PowerShell + Posh Git

从官网下载安装即可。

开始菜单中就有

msysGit + PowerShell + Posh Git

Posh Git

GitHub： https://github.com/dahlbyk/posh-git

是一系列的PowerShell 脚本，用来实现Git 和 PowerShell的集成

尤其是能够自动加载公钥，在需要远程操作的时候非常方便

根据文档安装即可。

注意：PowerShell默认情况是禁止执行脚本的，需要更改执行策略

在PowerShell中执行 Get-ExecutionPolicy 查看当前策略，应该是

RemoteSinged 或者Unrestricted

执行 Set-ExecutionPolicy RemoteSined 修改即可

生成 RSA 密钥

安装msysGit之后，应该就已经顺带安装了 ssh 客户端

在命令行中输入

```
ssh-keygen
```

根据提示操作即可。

（可能需要添加环境变量等。）

在GitHub上添加的是公钥，如 “id_rsa.pub”

生成时如果没有指定地址，则位于用户根目录下的.ssh文件夹中。

Windows用户，在C:\Users\<username>\.ssh

Linux用户，在~/.ssh下

若希望通过ssh协议和服务器进行通信，将本机的id_ras.pub添加到服务器的~/.ssh/authorized_keys中，有多台设备则一行一个。