Source Control for People Who Don't Like Source Control

Jim Weirich
Chief Scientist / EdgeCase
jim@edgecase.com
@jimweirich

Source Control for People

Jim Weirich
Chief Scientist / EdgeCase
jim@edgecase.com
@jimweirich

Your Mission

Your Mission

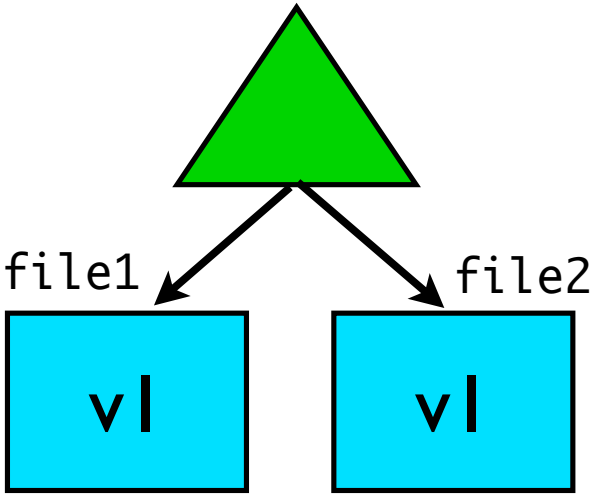
Design and Build A Source Code Control System

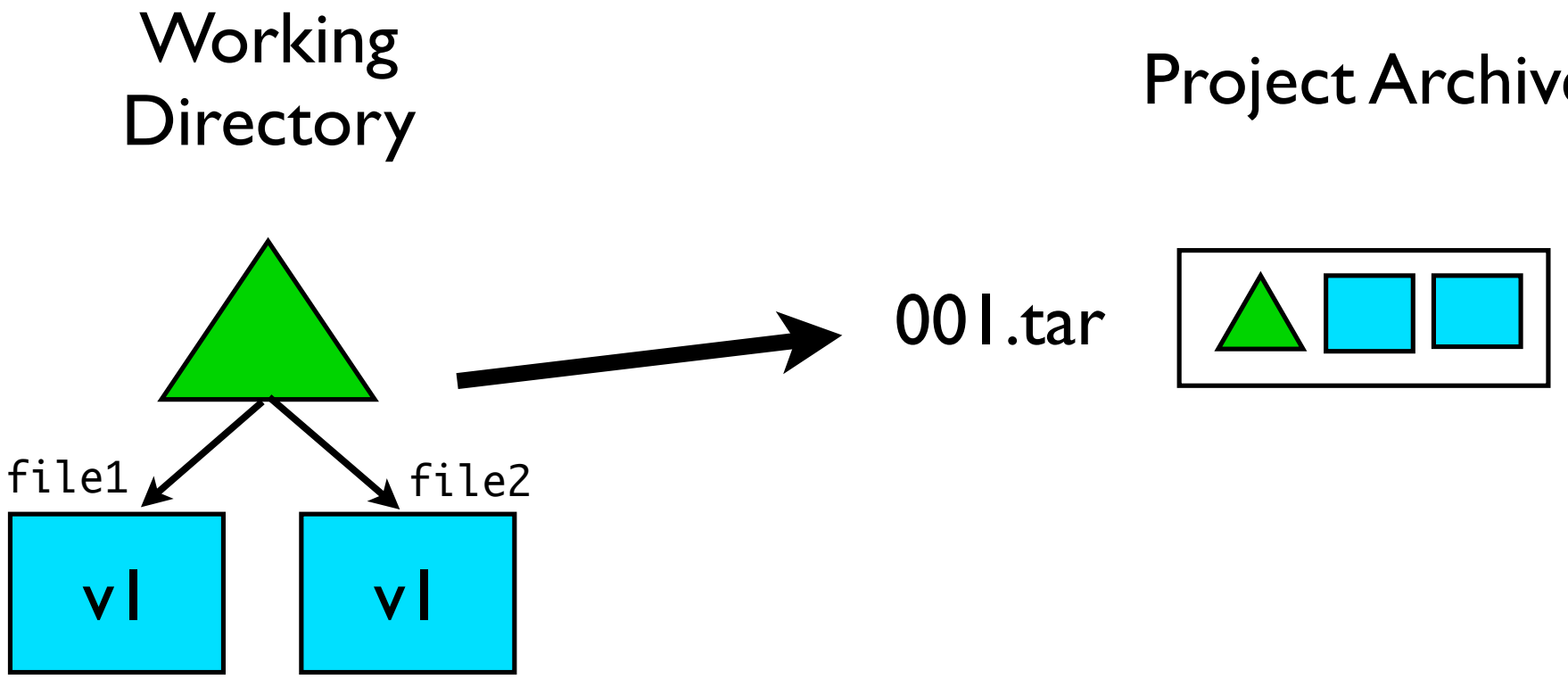
Custom Source Control

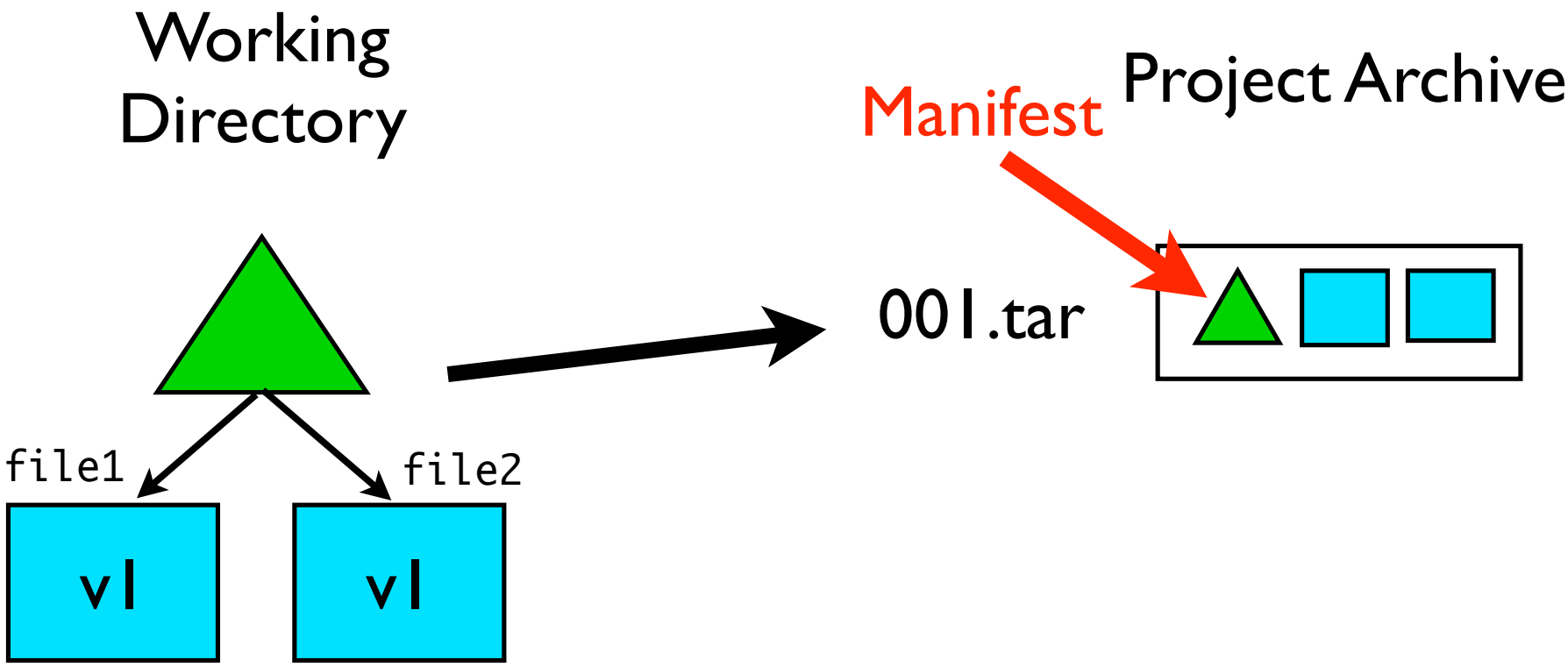
Custom
Source
Control

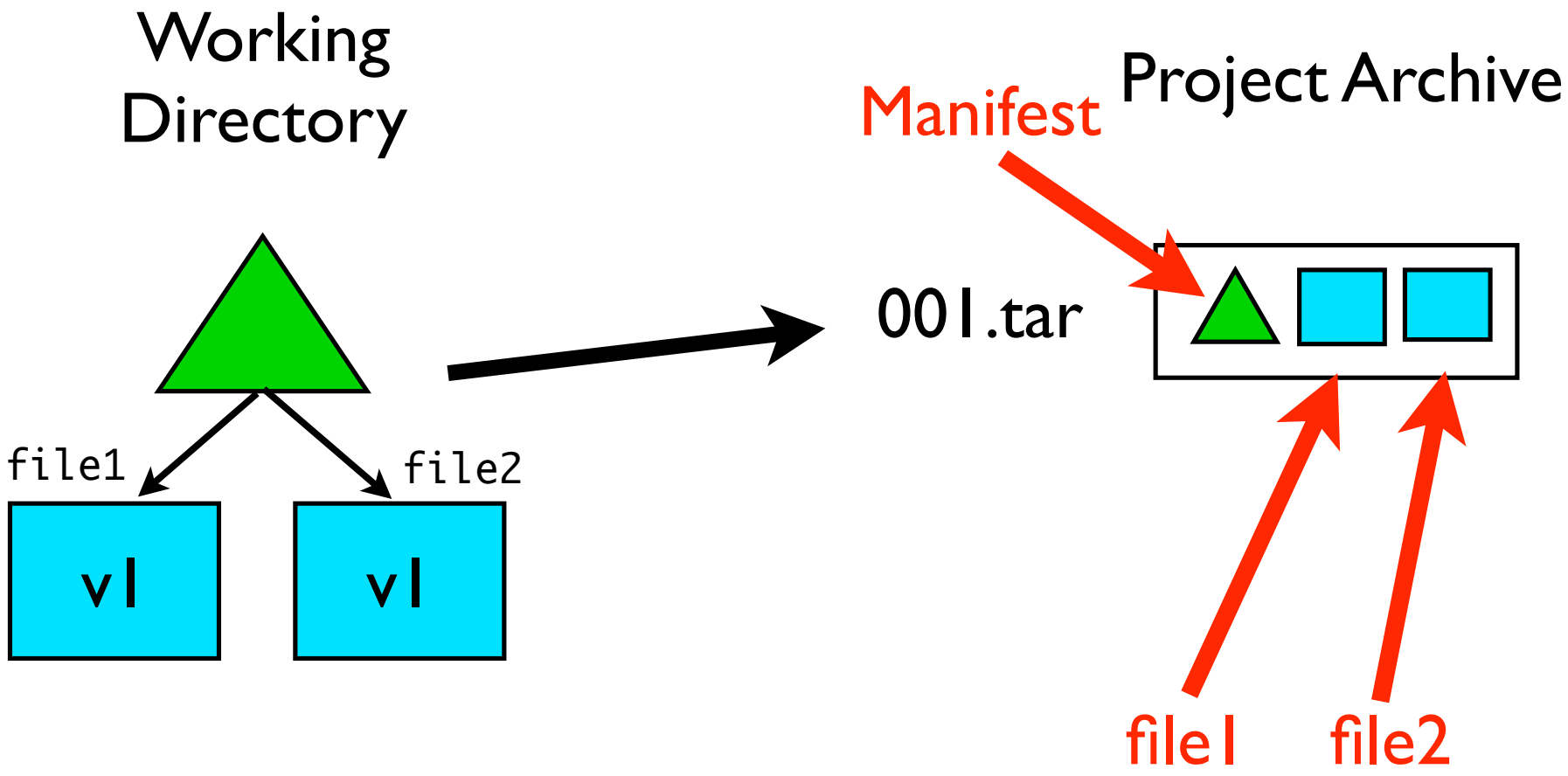
In the *Old Days* ...

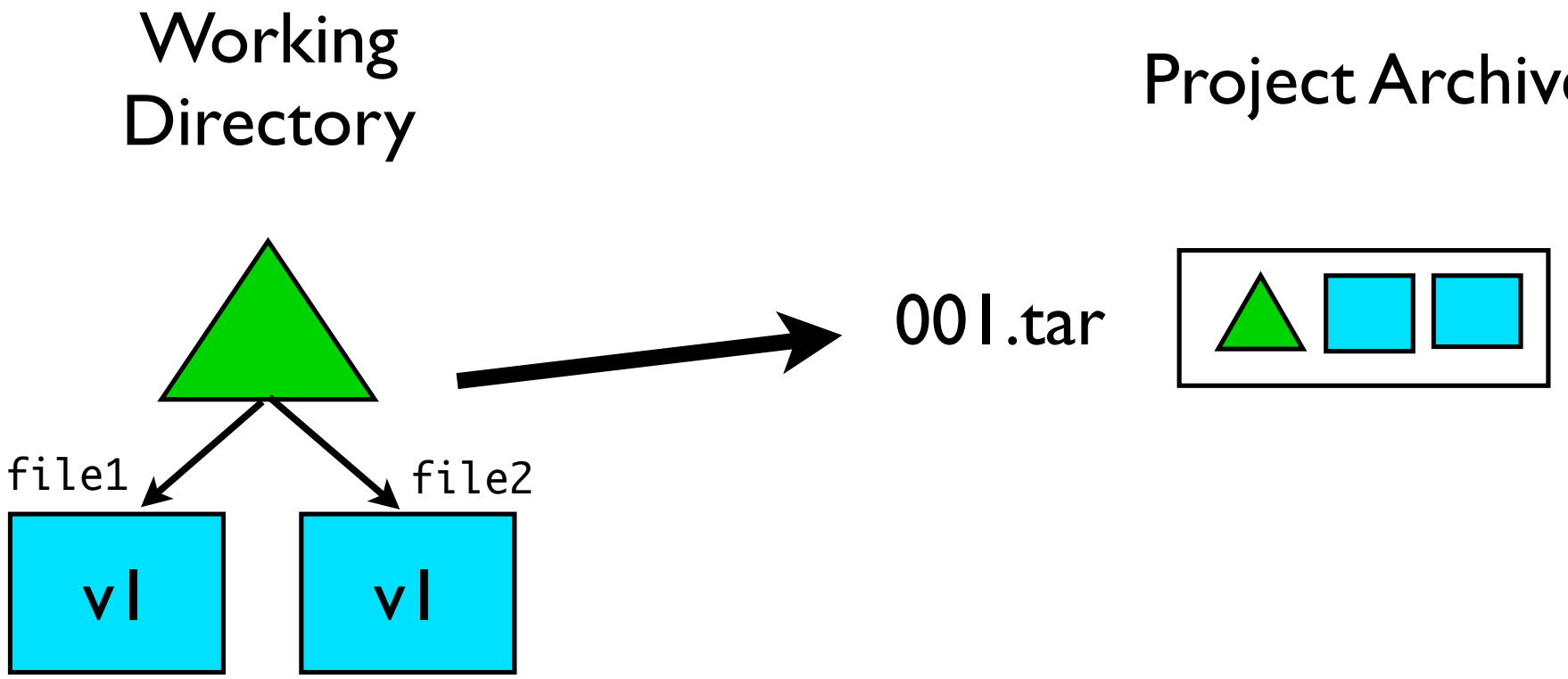
Working Directory

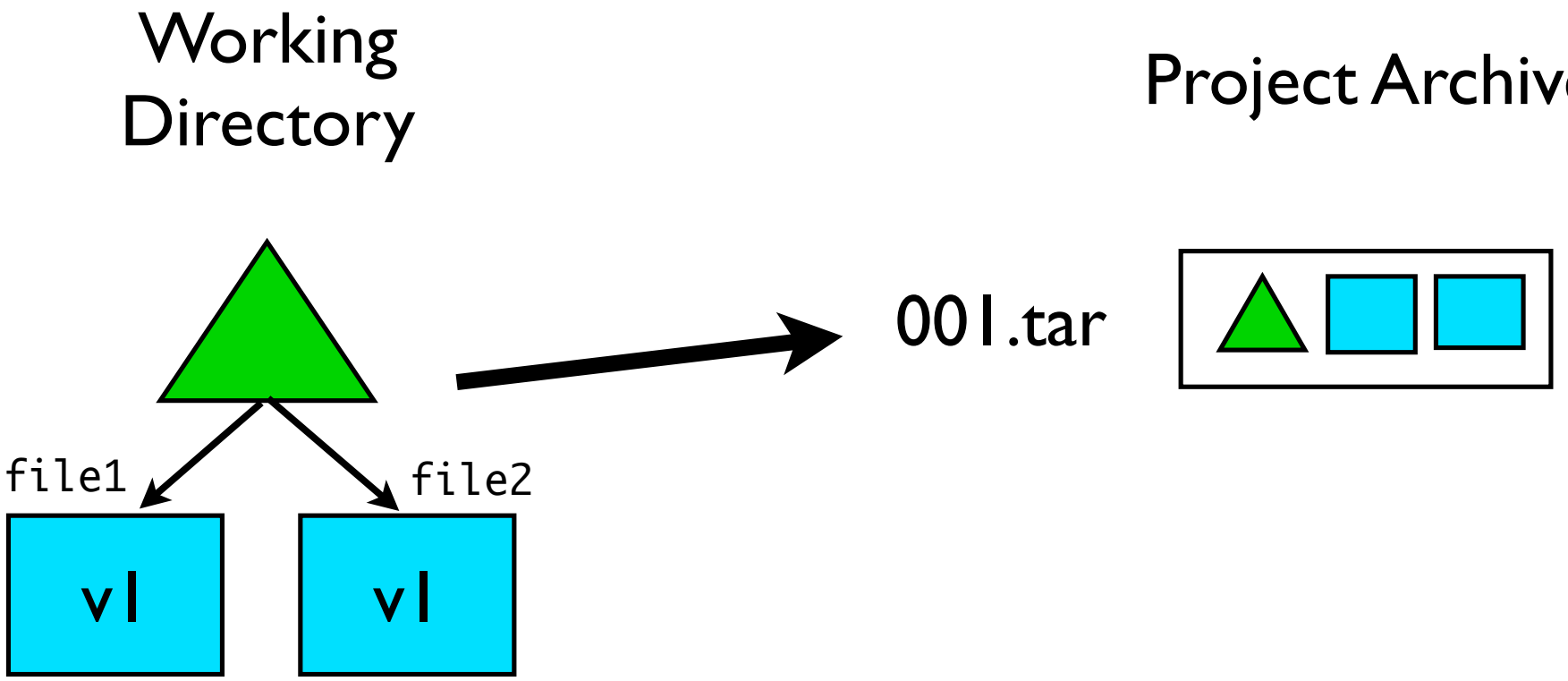



Working
Directory
Project Archive
001.tar
file1
file2
v1
v1

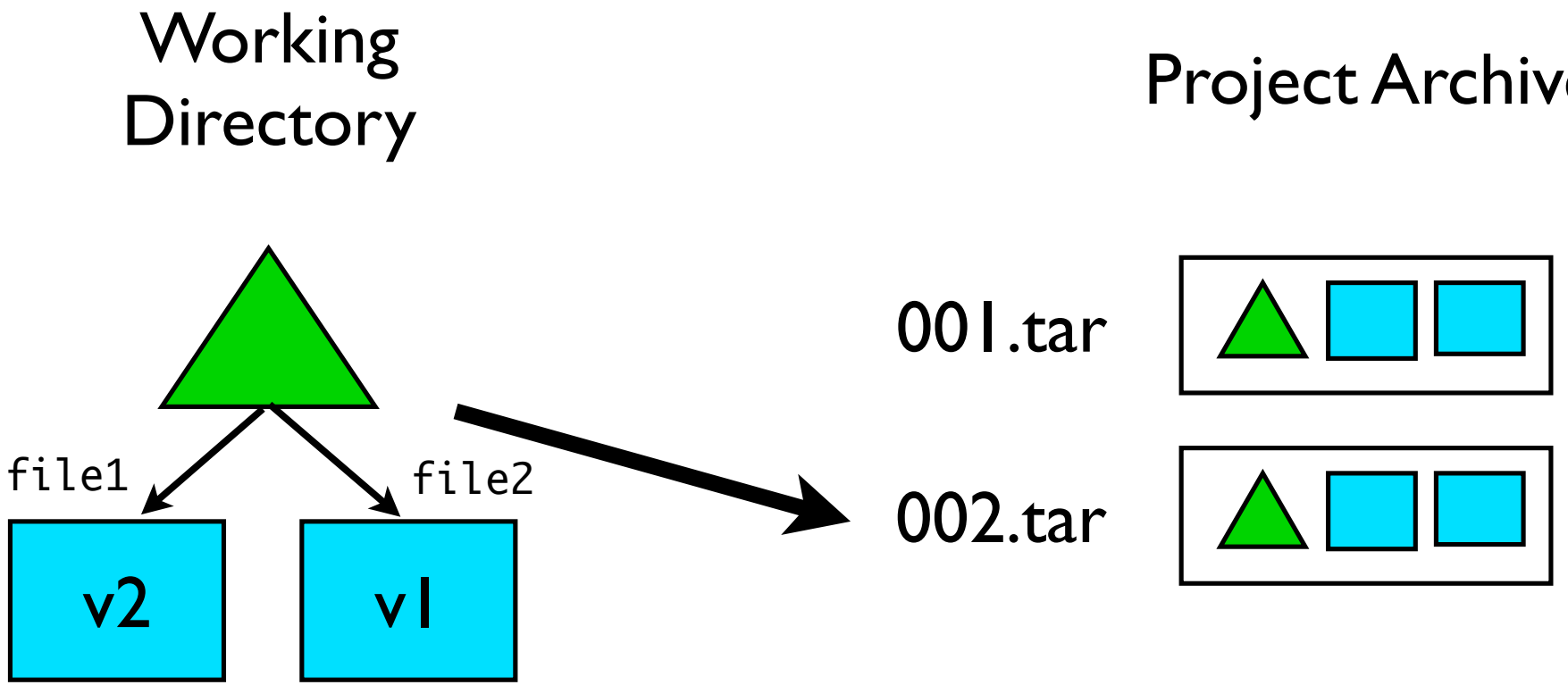

Working
Directory
Manifest
Project Archive
001.tar
file1
file2
v1
v1


Working
Directory
file1
file2
v1
v1
Manifest
Project Archive
001.tar
file1
file2


Working
Directory
Project Archive
001.tar
file1
file2
v1
v1

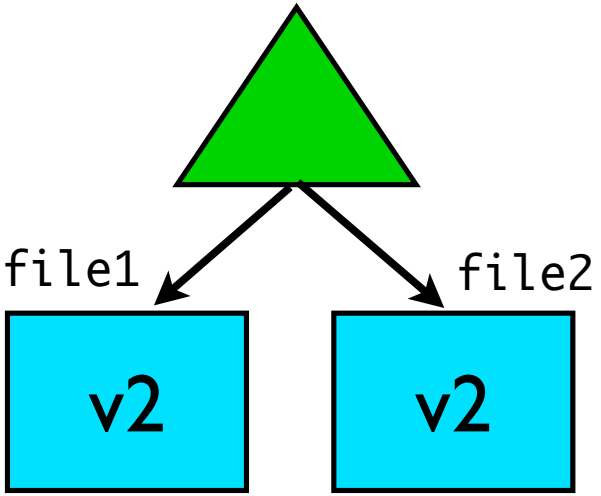

Working
Directory
Project Archive
001.tar
file1
file2
v1
v1

```
$ csc snap
```

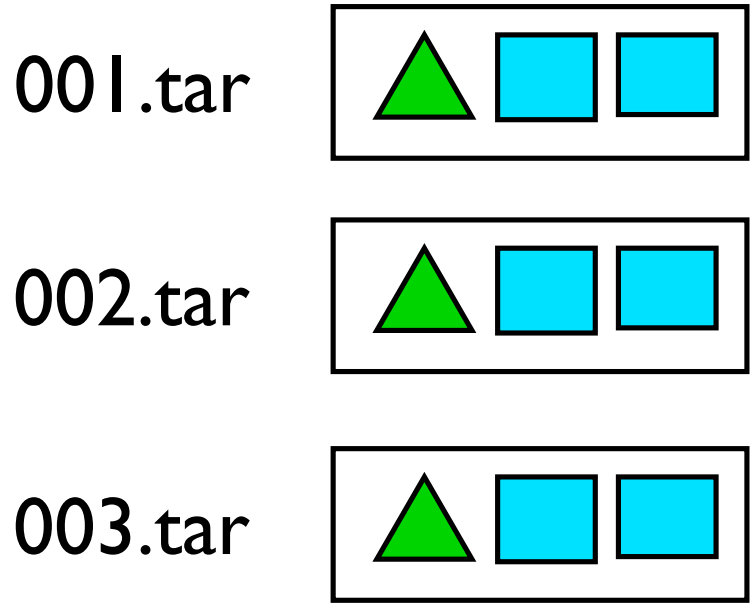

Working
Directory
Project Archive
file1
file2
v2
v1
001.tar
002.tar

```
$ csc snap
```

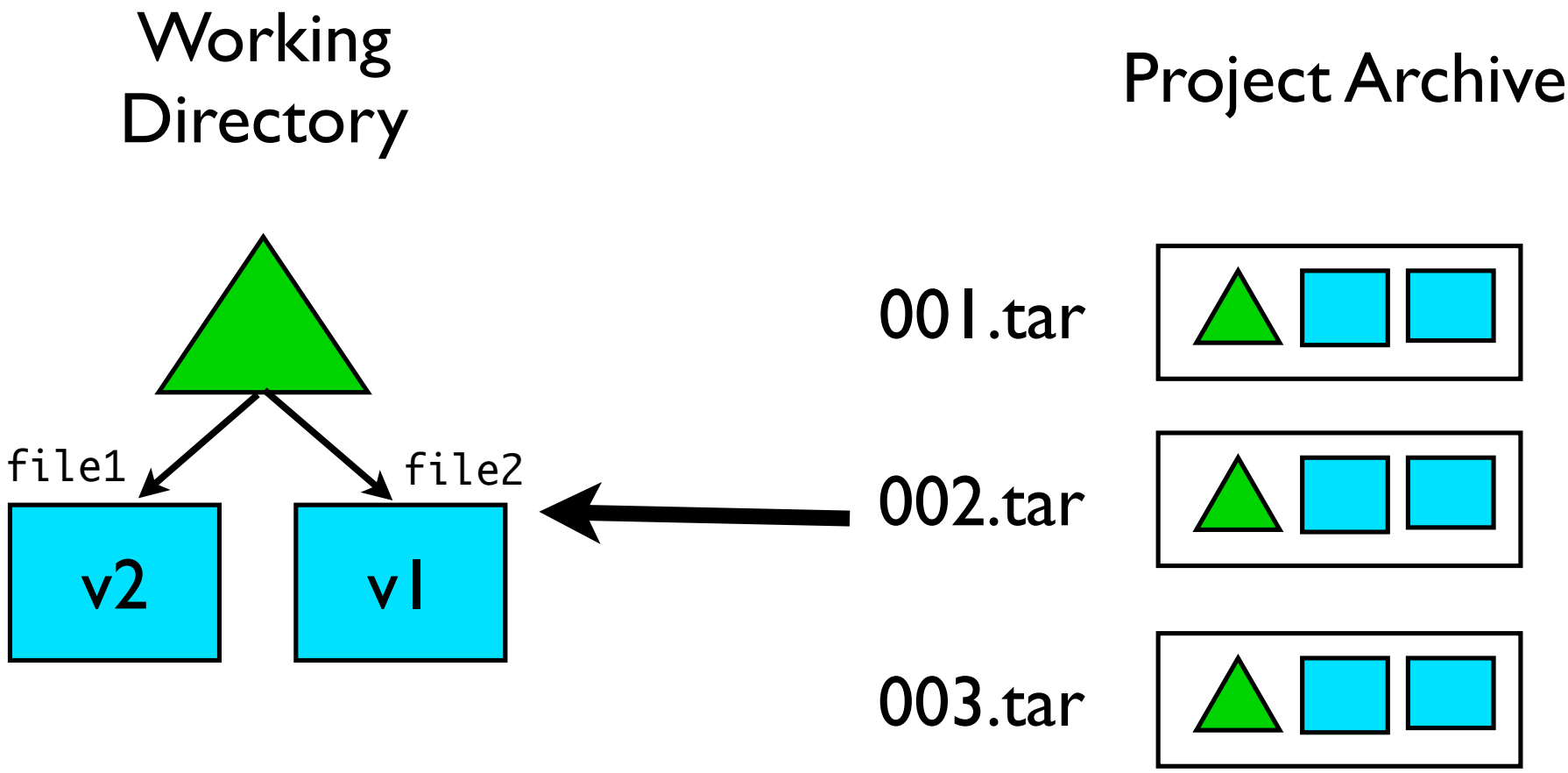
Working Directory



Project Archive




```
$ csc snap
```



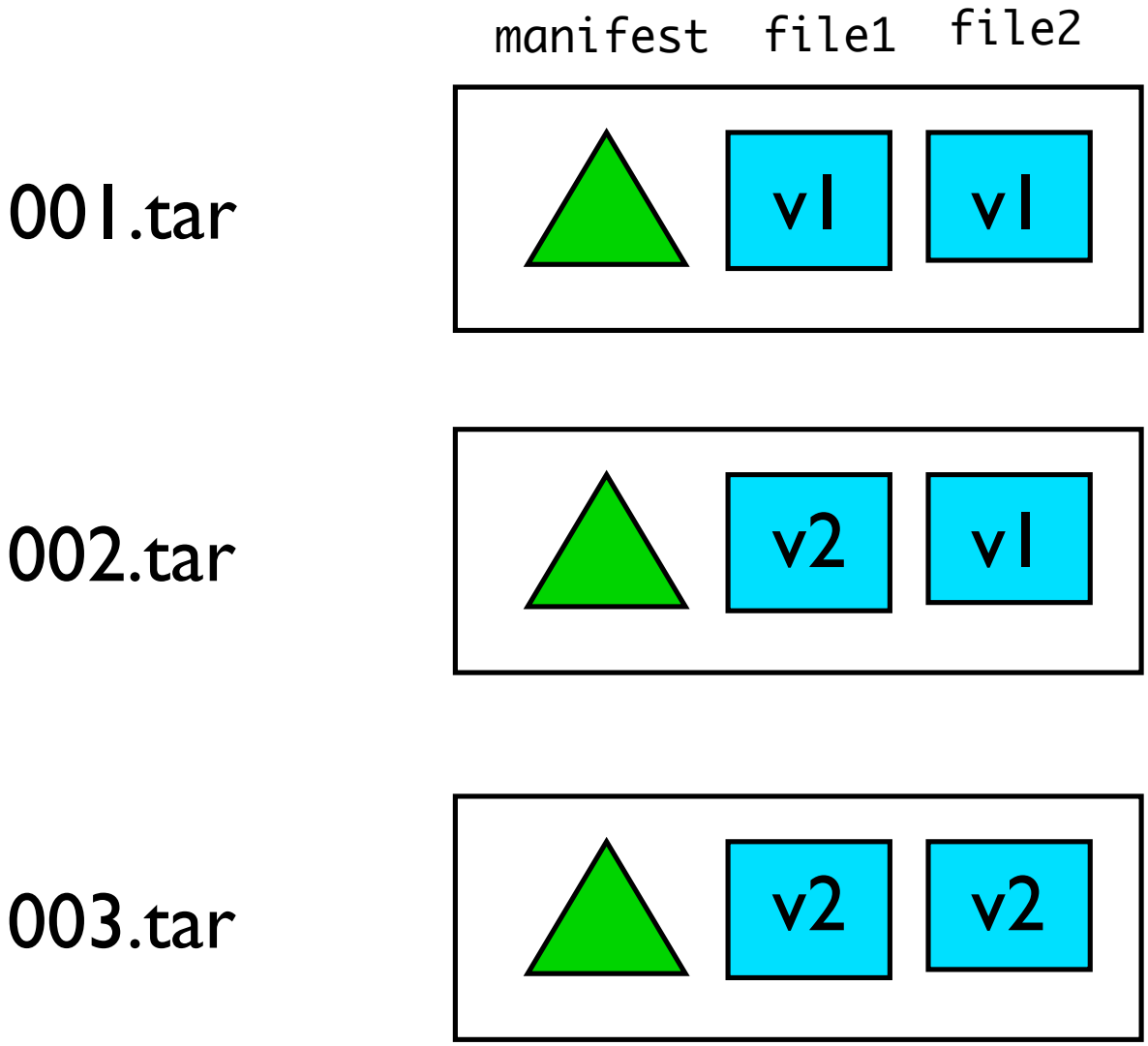

```
$ csc checkout 002
```

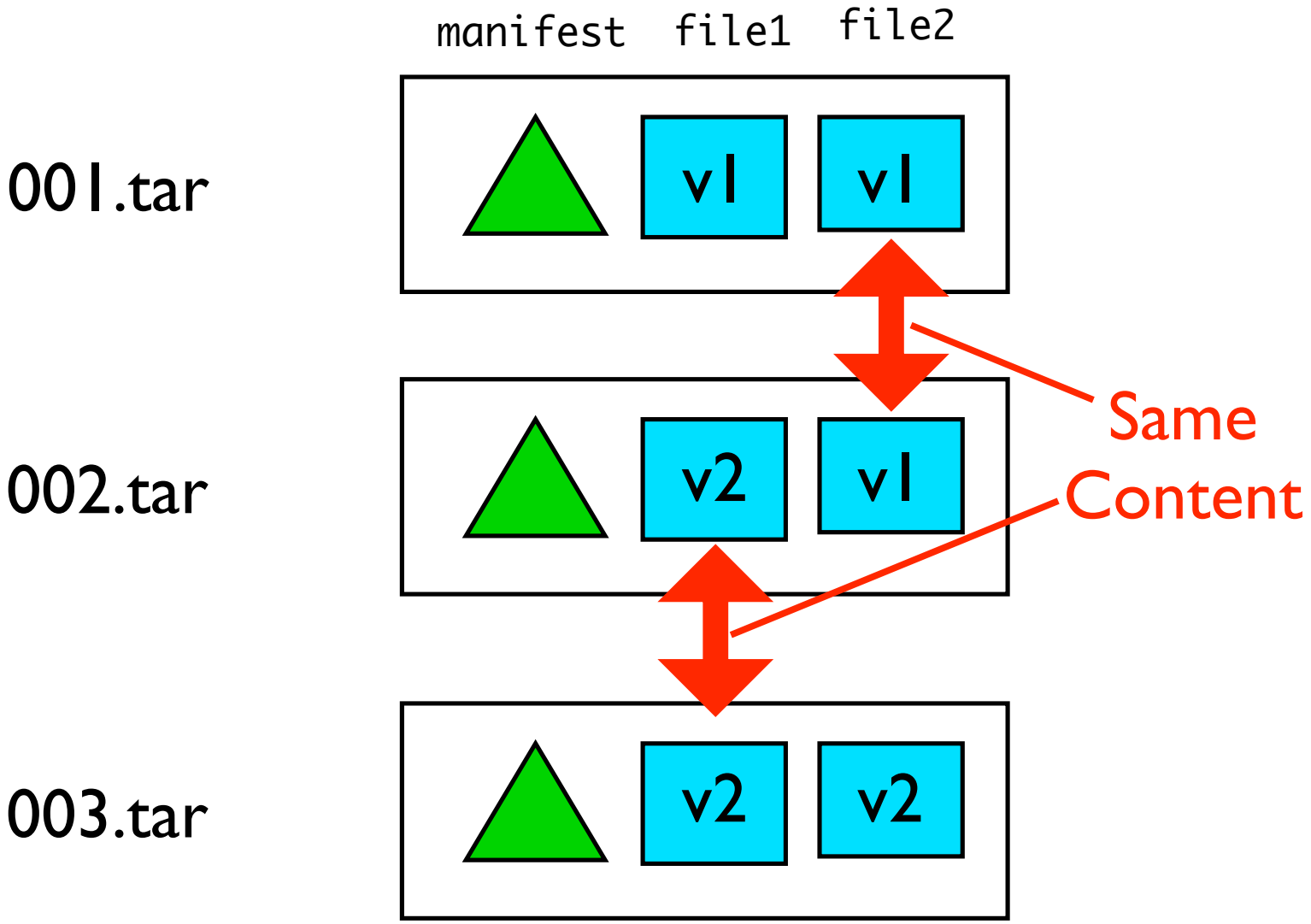
Two Basic Commands

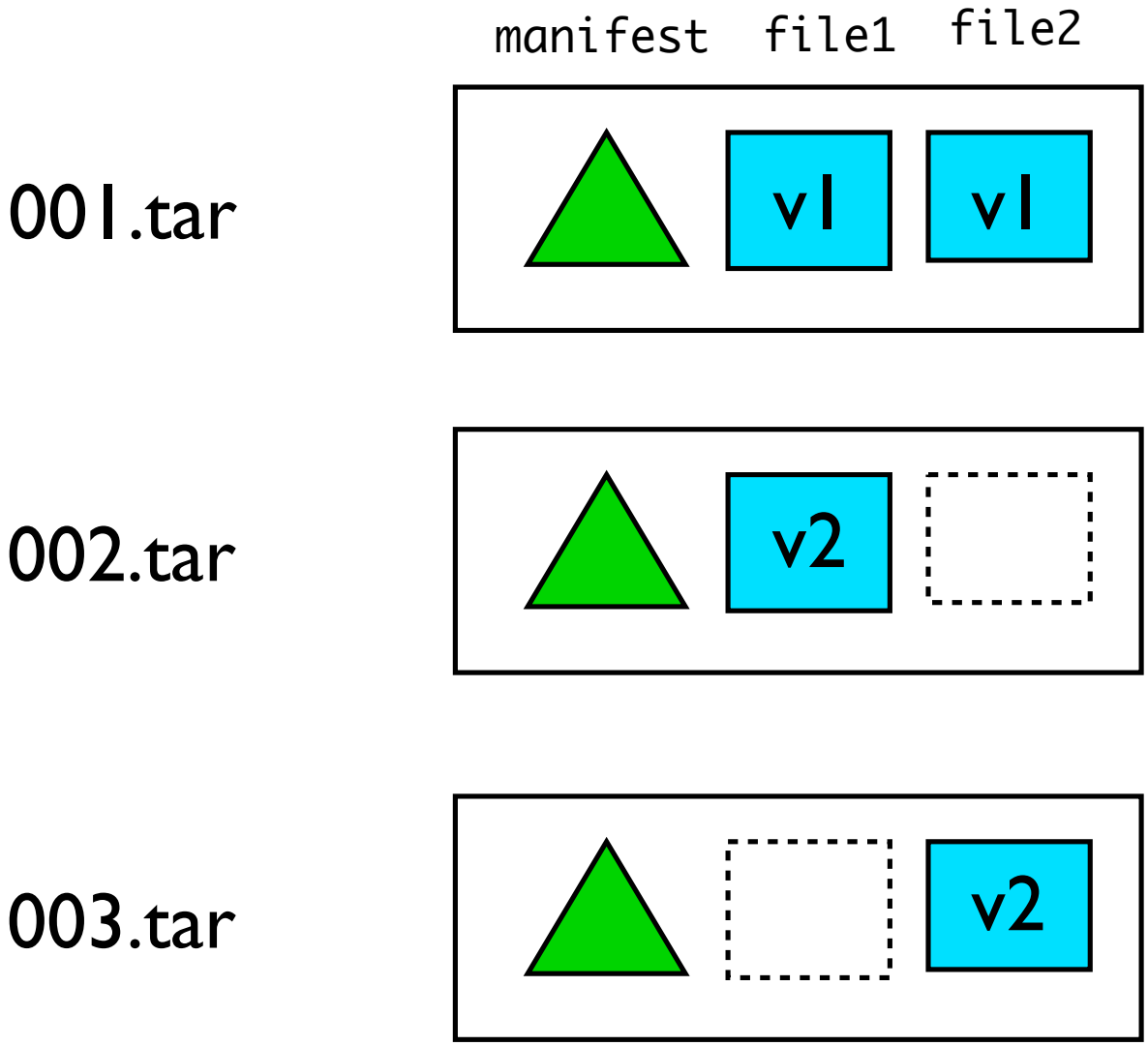
- Snap
- Checkout

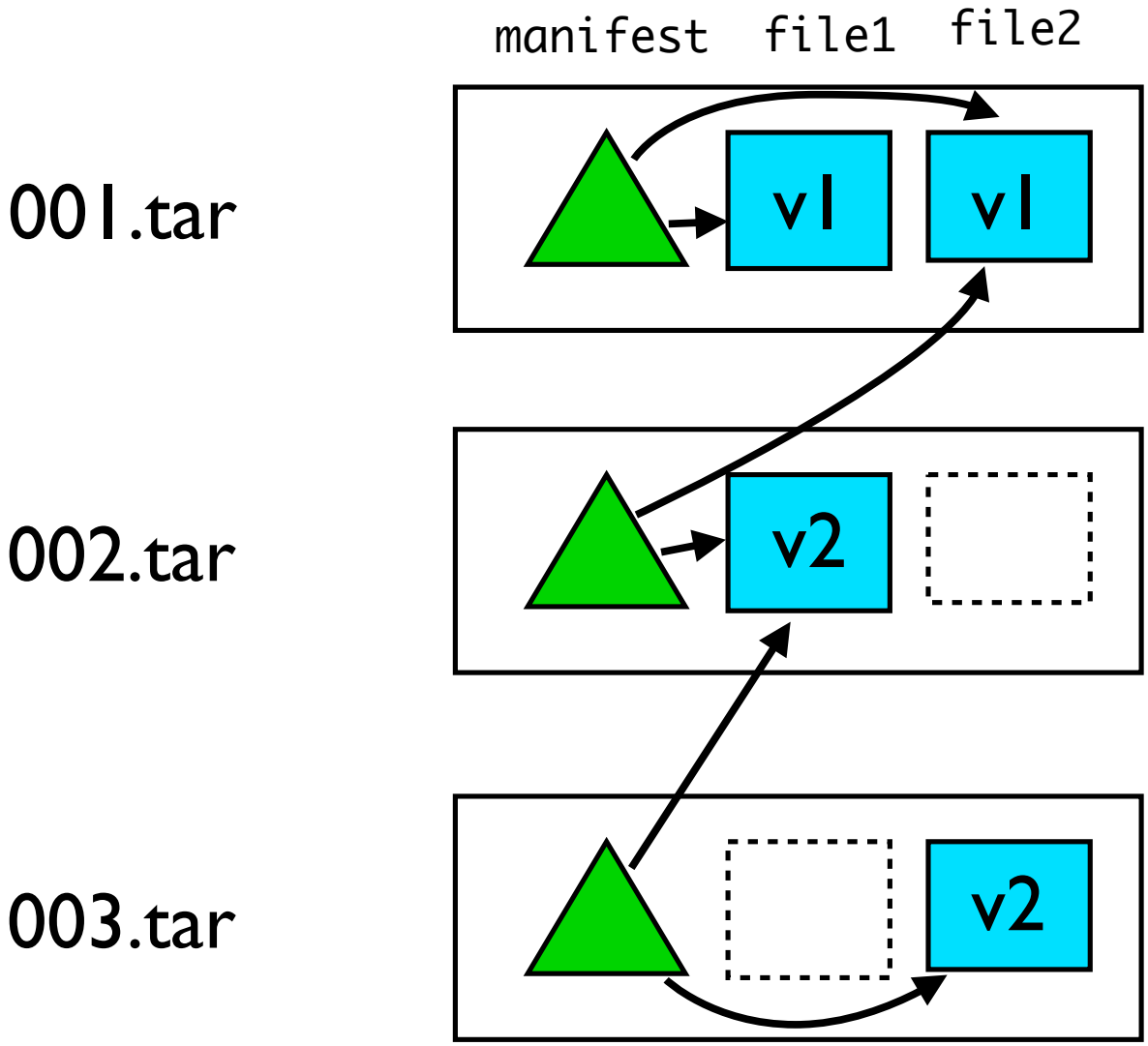
Very Easy /
Very Simple

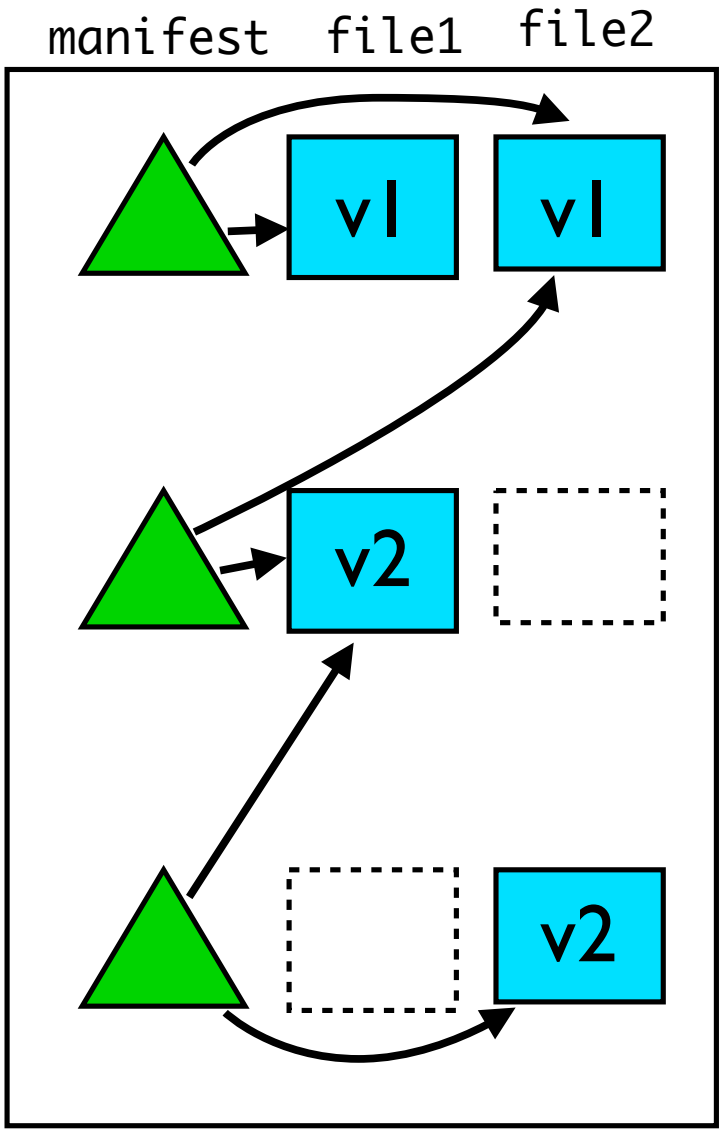
Rather Wasteful

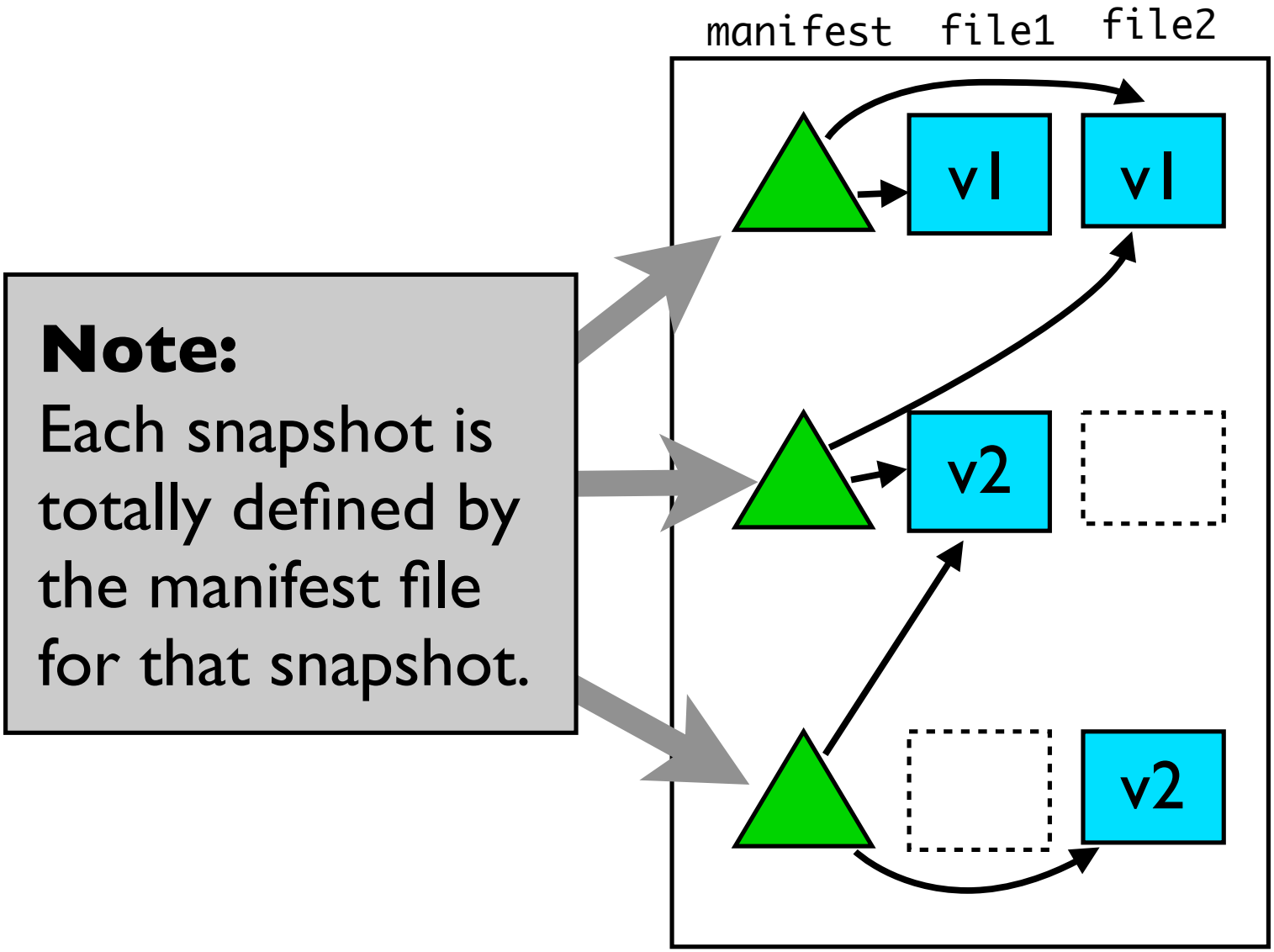

manifest
file1
file2
001.tar
v1
v1
002.tar
v2
v1
003.tar
v2
v2

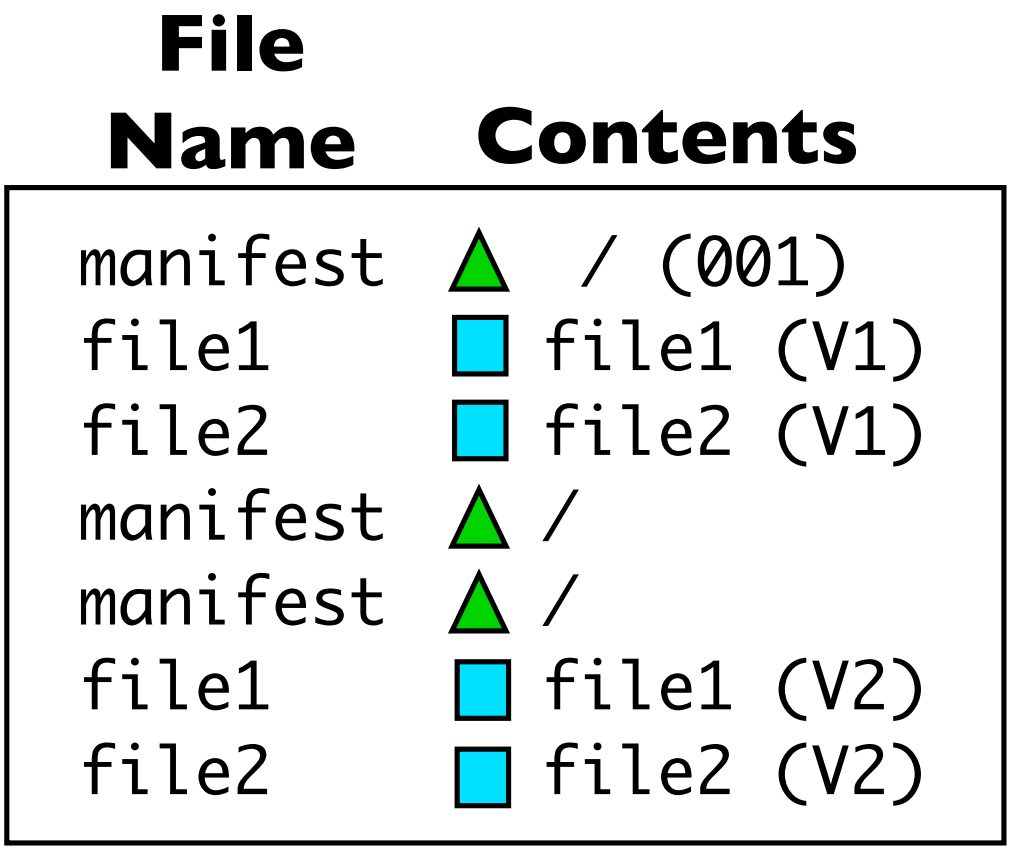

manifest
file1
file2
001.tar
v1
v1
002.tar
v2
v1
003.tar
v2
v2
Same
Content

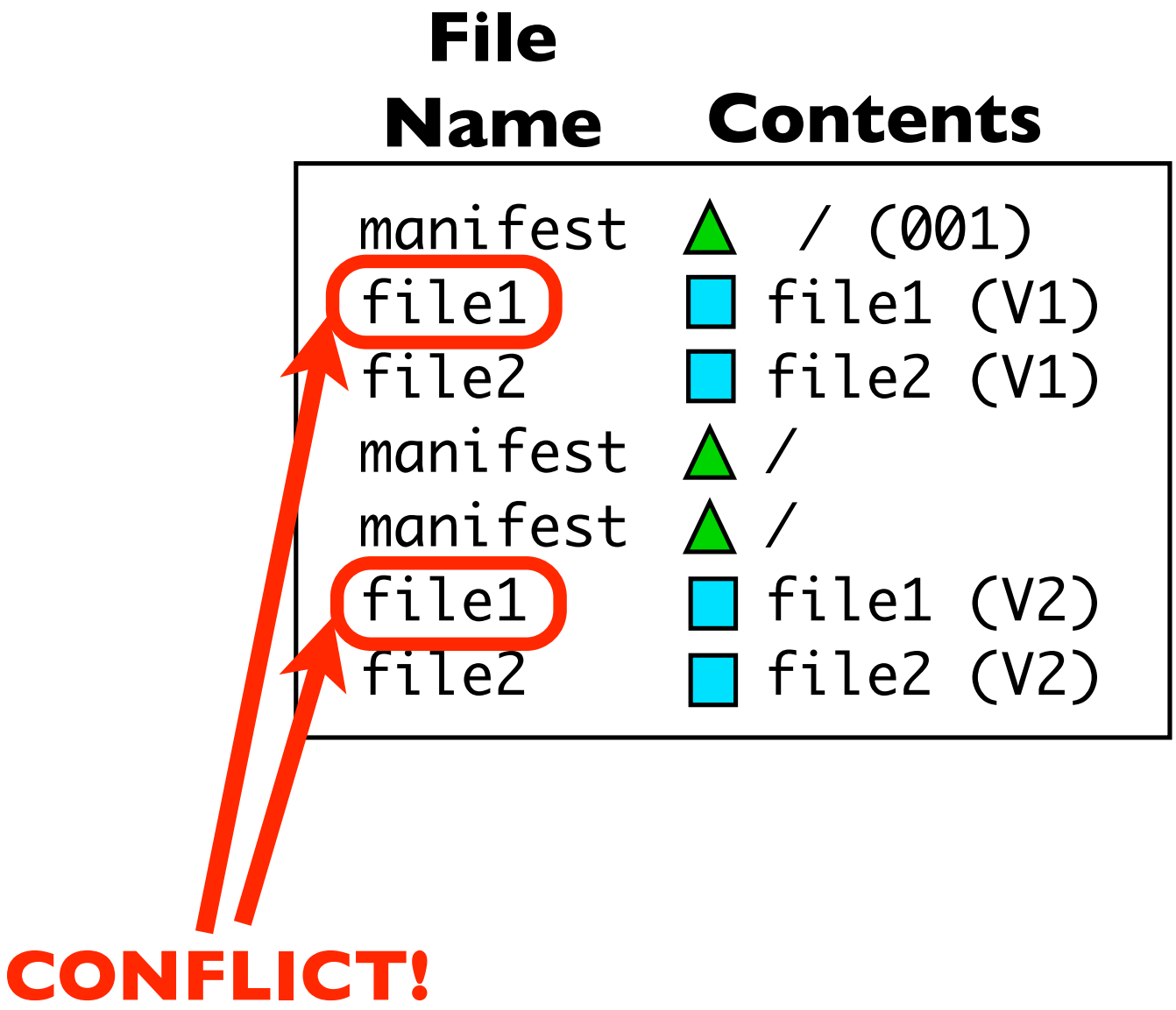

manifest
file1
file2
001.tar
v1
v1
002.tar
v2
003.tar
v2

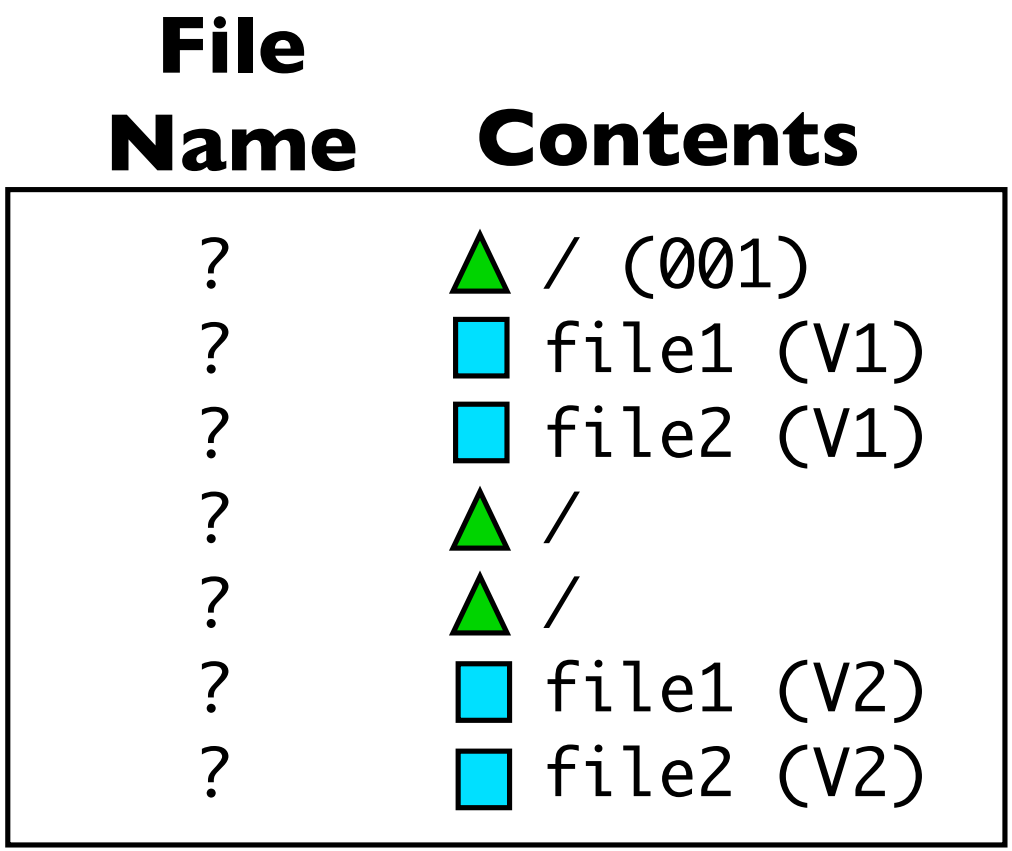

manifest
file1
file2
001.tar
v1
v1
002.tar
v2
003.tar
v2


manifest
file1
file2
v1
v1
v2
v2

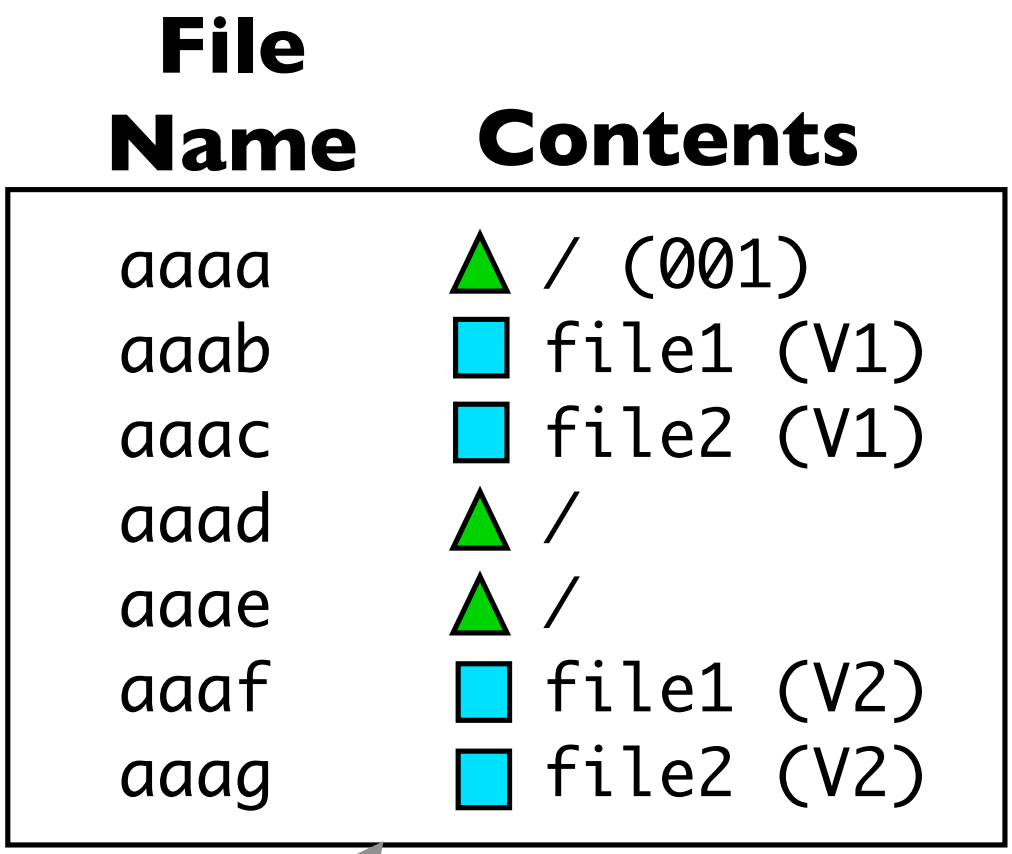

manifest
file1
file2
v1
v1
v2
v2
Note:
Each snapshot is totally defined by the manifest file for that snapshot.


File Name
Contents
manifest
/ (001)
file1
file1 (V1)
file2
file2 (V1)
manifest
/
manifest
/
file1
file1 (V2)
file2
file2 (V2)

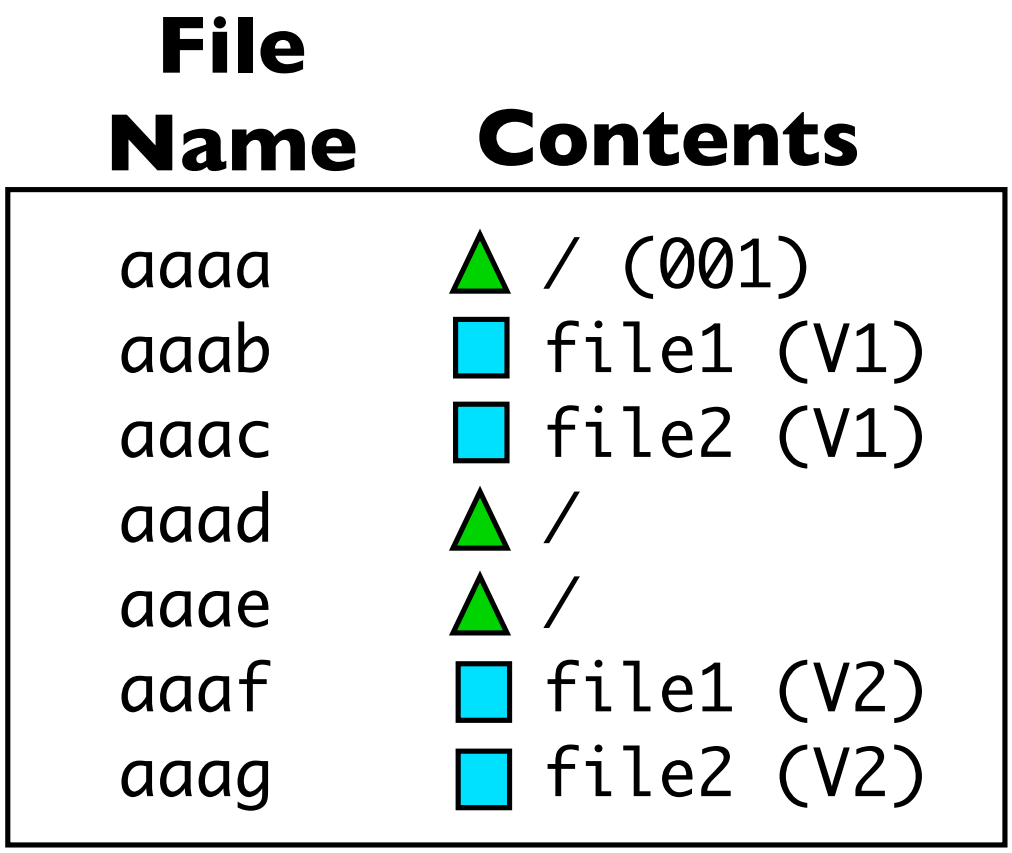

File Name
Contents
manifest
/ (001)
file1
file1 (V1)
file2
file2 (V1)
manifest
/
manifest
/
file1
file1 (V2)
file2
file2 (V2)
CONFLICT!


File Name
Contents
? / (001)
? file1 (V1)
? file2 (V1)
? /
? /
? file1 (V2)
? file2 (V2)

File Name	Contents
aaaa	/ (001)
aaab	file1 (V1)
aaac	file2 (V1)
aaad	/
aaae	/
aaaf	file1 (V2)
aaag	file2 (V2)



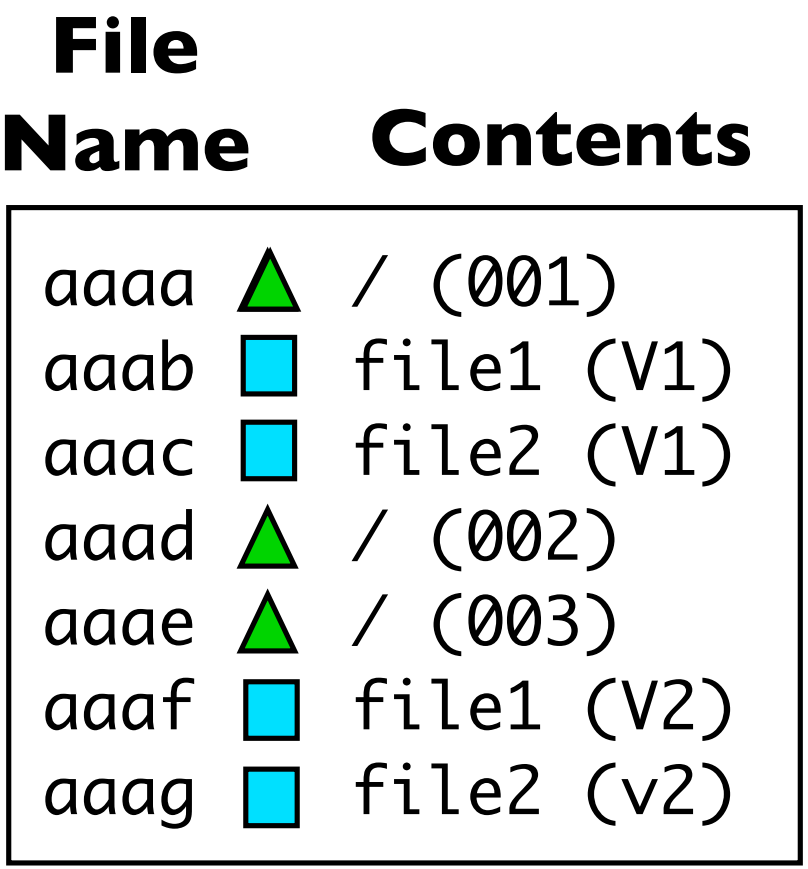
Note:
The Project Archive is just a directory with files in it.

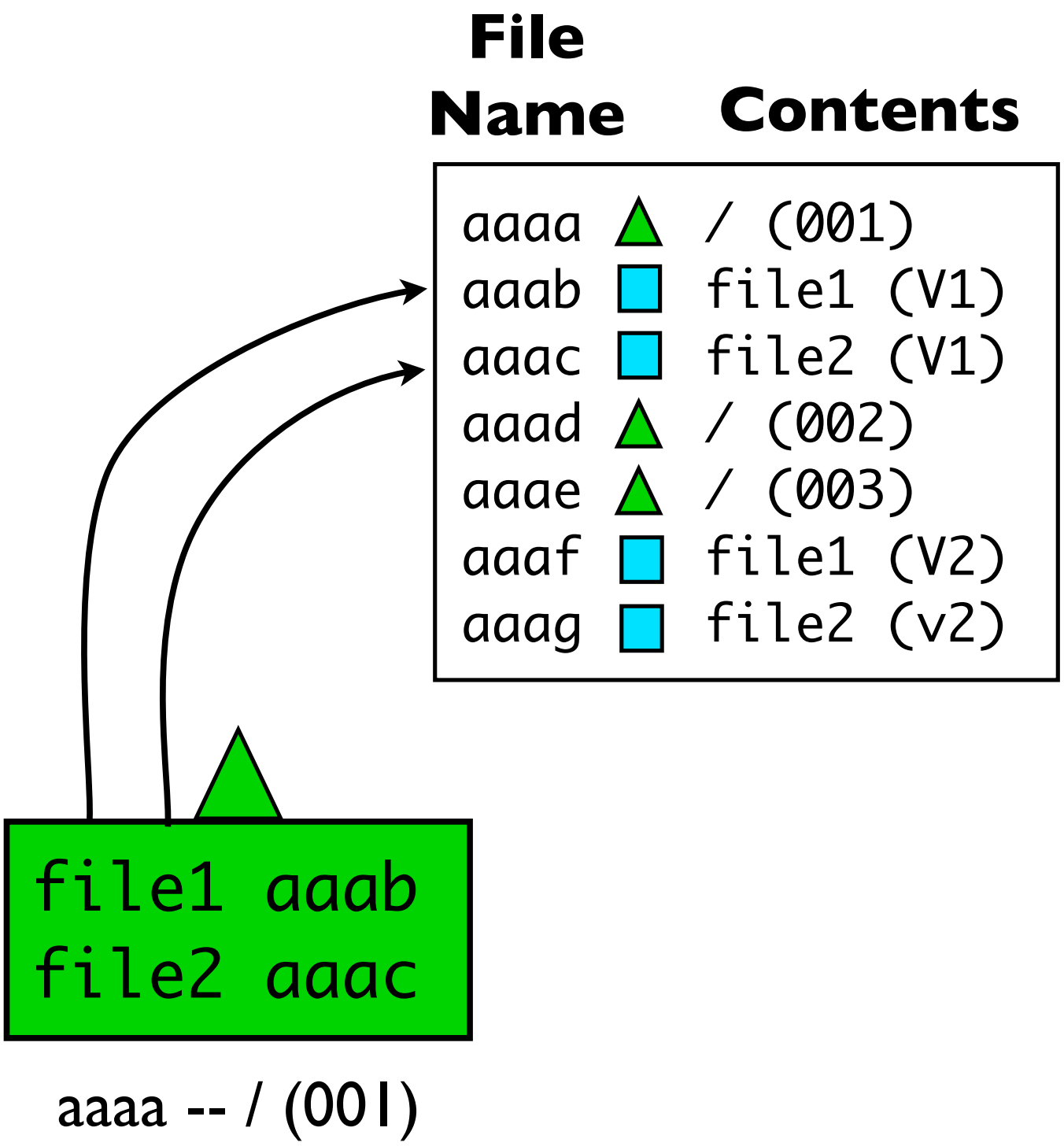



```
$ ls
aaaa  aaab  aaac  aaad  aaae  aaaf  aaag
```

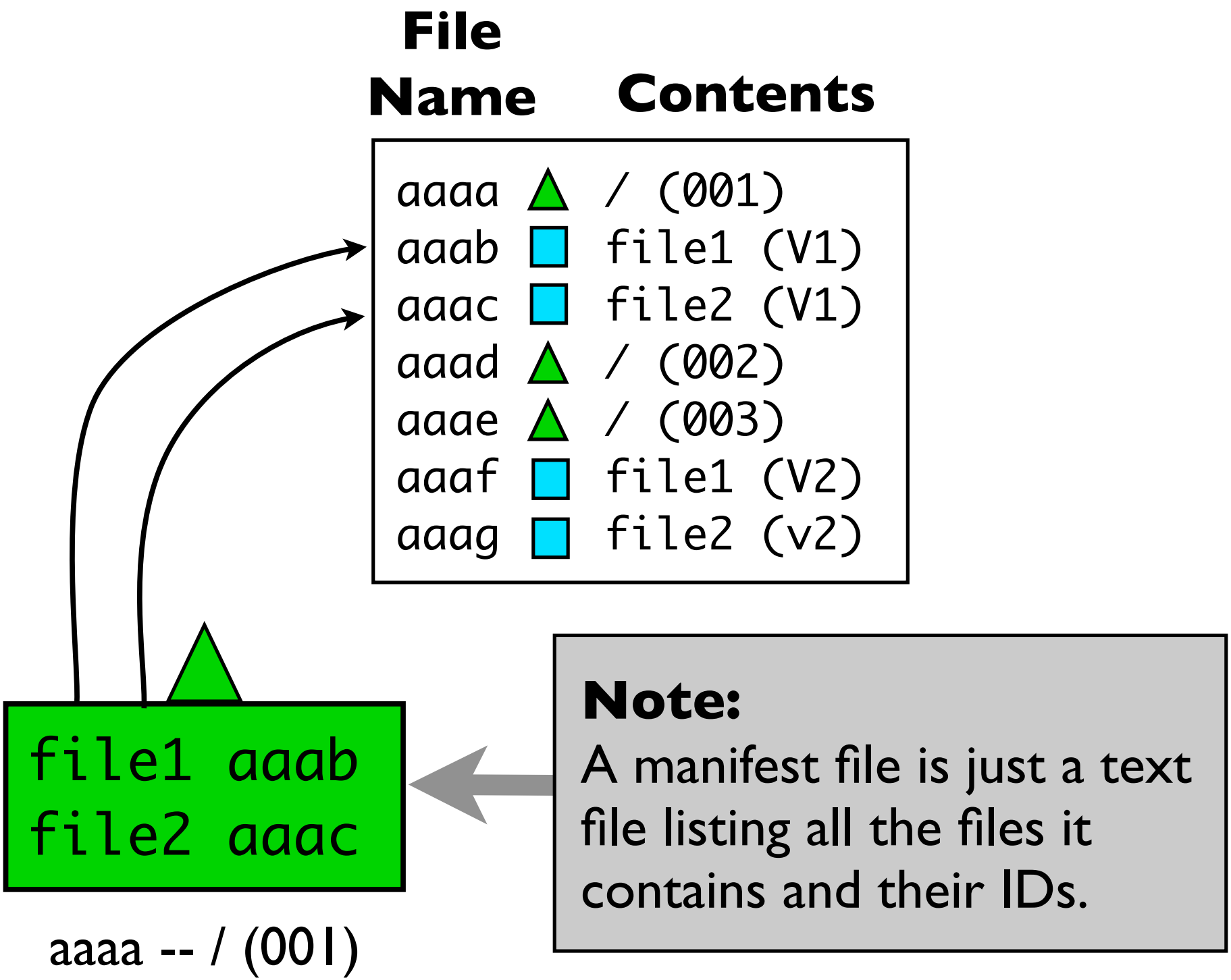
Project Archive

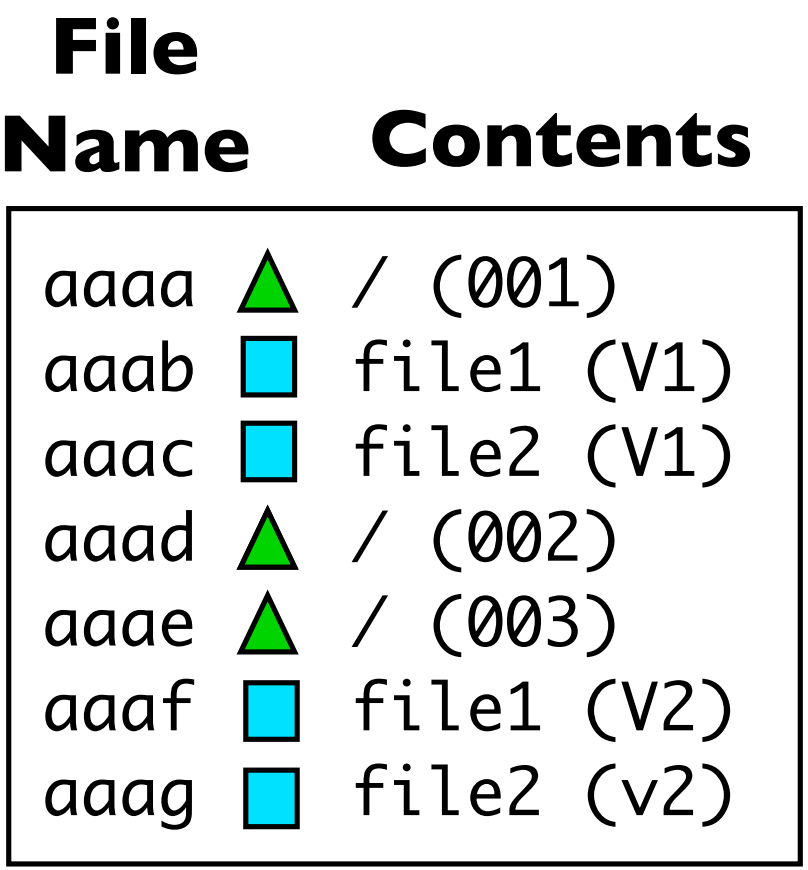
- Is just a **directory with files** in it
- The archive files have **arbitrary** names.
  - The file names can be anything as long as they are **unique**
- We track the **real file** names in the manifest files.

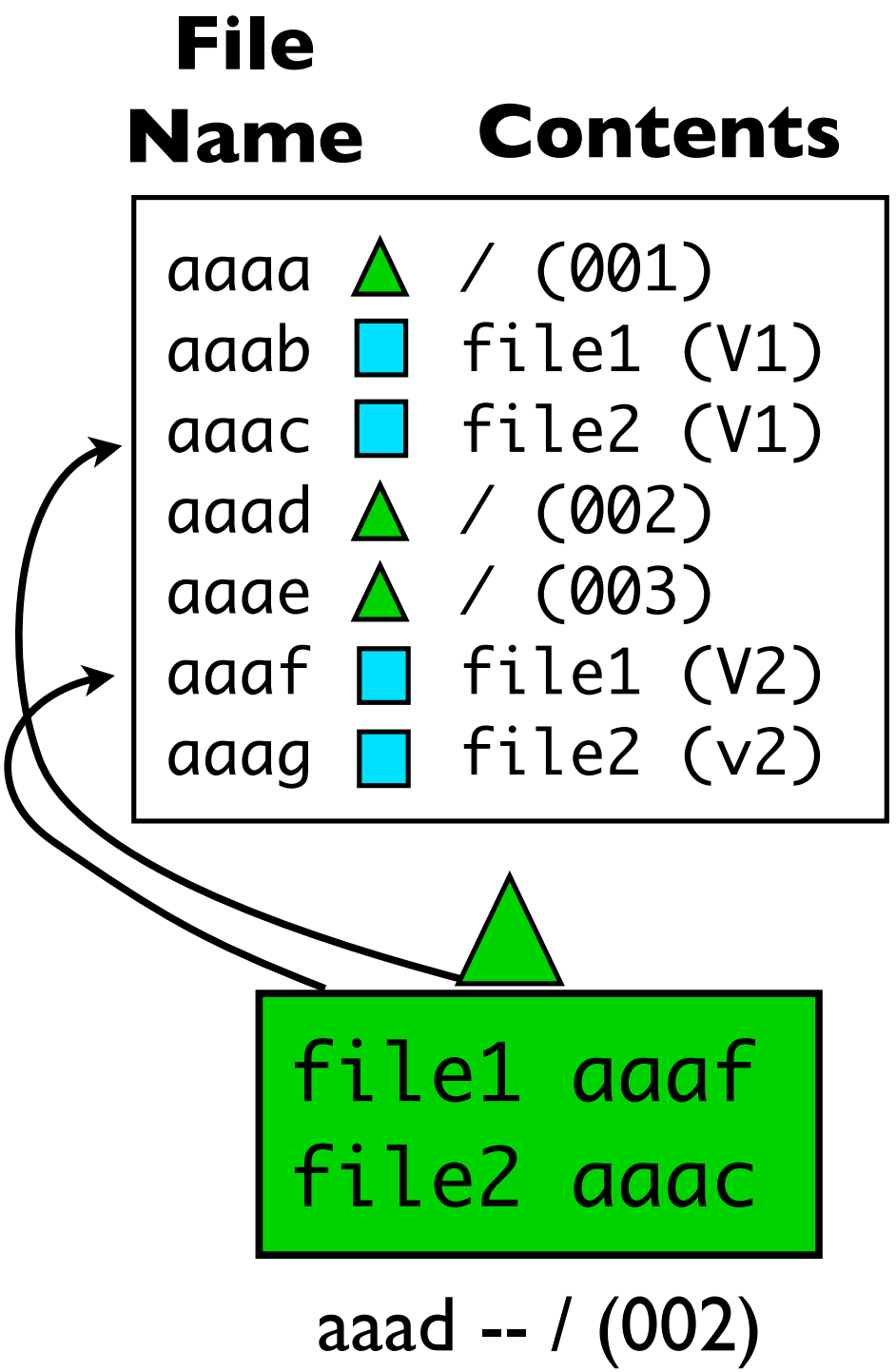

File Name
Contents
aaaa
/ (001)
aaab
file1 (V1)
aaac
file2 (V1)
aaad
/ (002)
aaae
/ (003)
aaaf
file1 (V2)
aaag
file2 (v2)


File Name
Contents
aaaa / (001)
aaab file1 (V1)
aaac file2 (V1)
aaad / (002)
aaae / (003)
aaaf file1 (V2)
aaag file2 (v2)
file1 aaab
file2 aaac

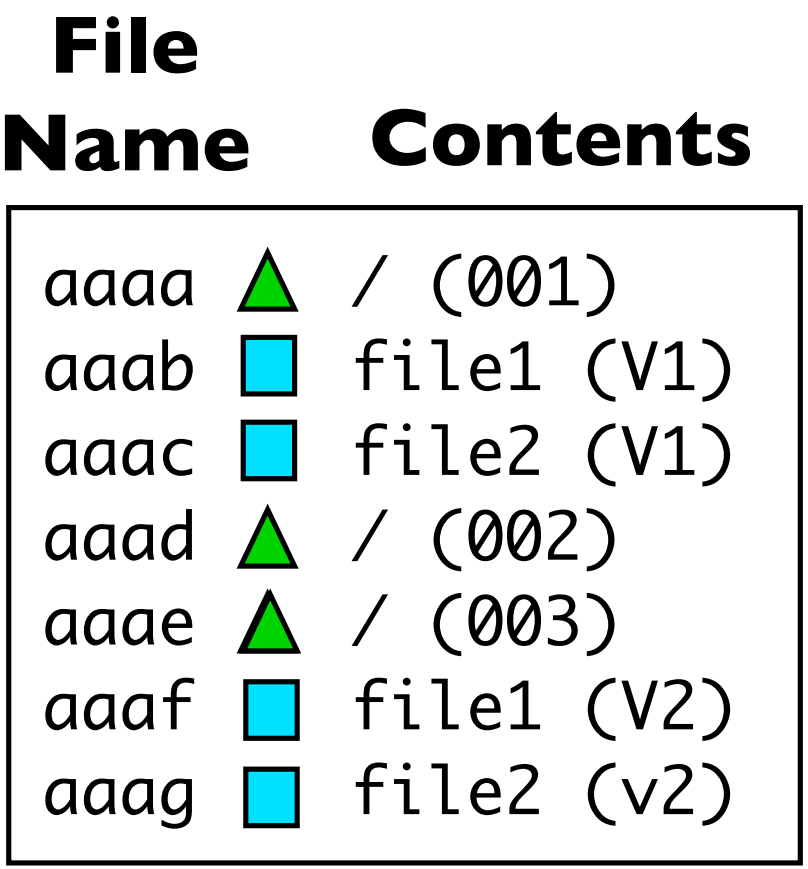
aaaa -- / (001)

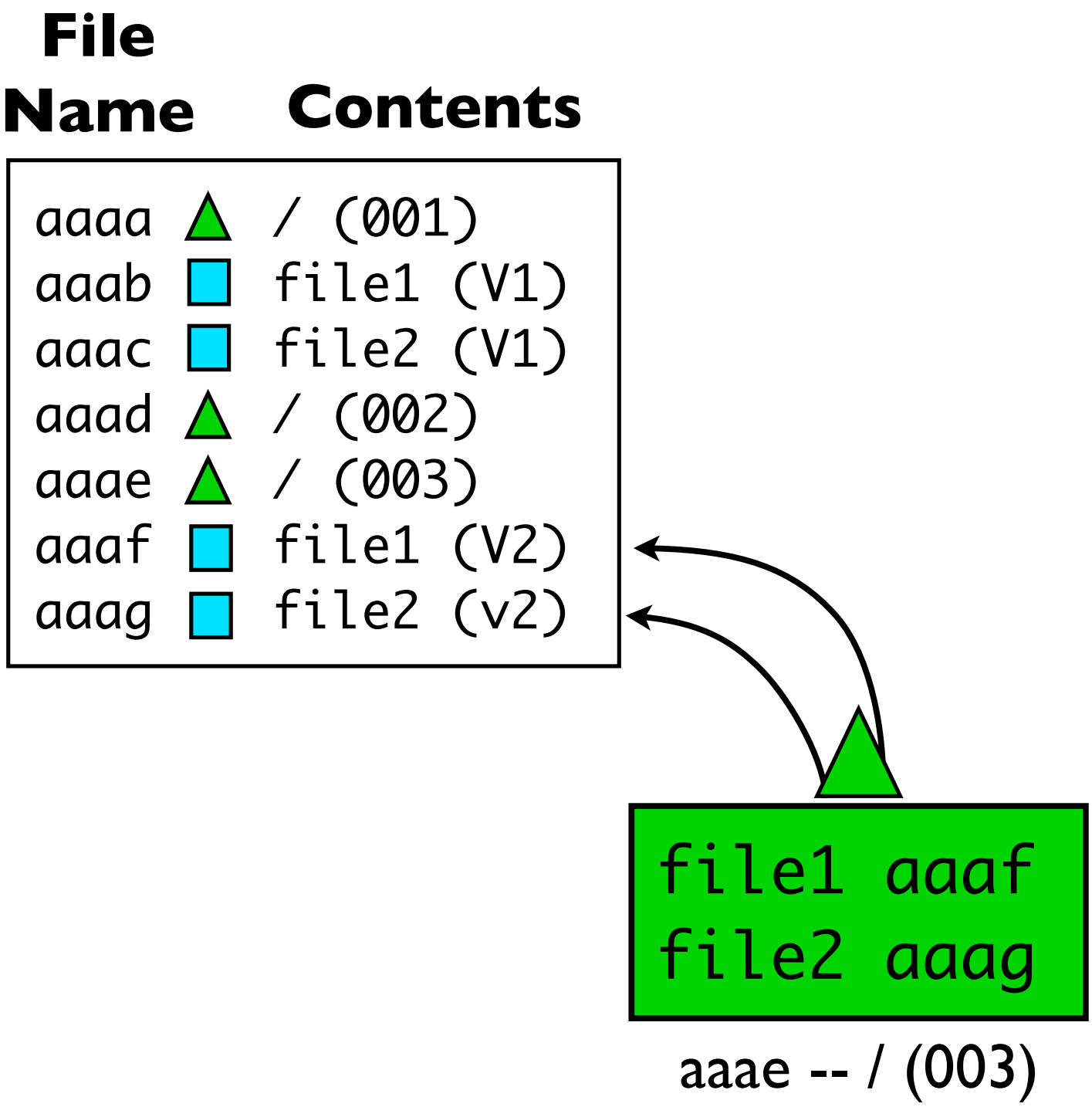

File Name
Contents
aaaa / (001)
aaab file1 (V1)
aaac file2 (V1)
aaad / (002)
aaae / (003)
aaaf file1 (V2)
aaag file2 (v2)
file1 aaab
file2 aaac
aaaa -- / (001)
Note:
A manifest file is just a text file listing all the files it contains and their IDs.

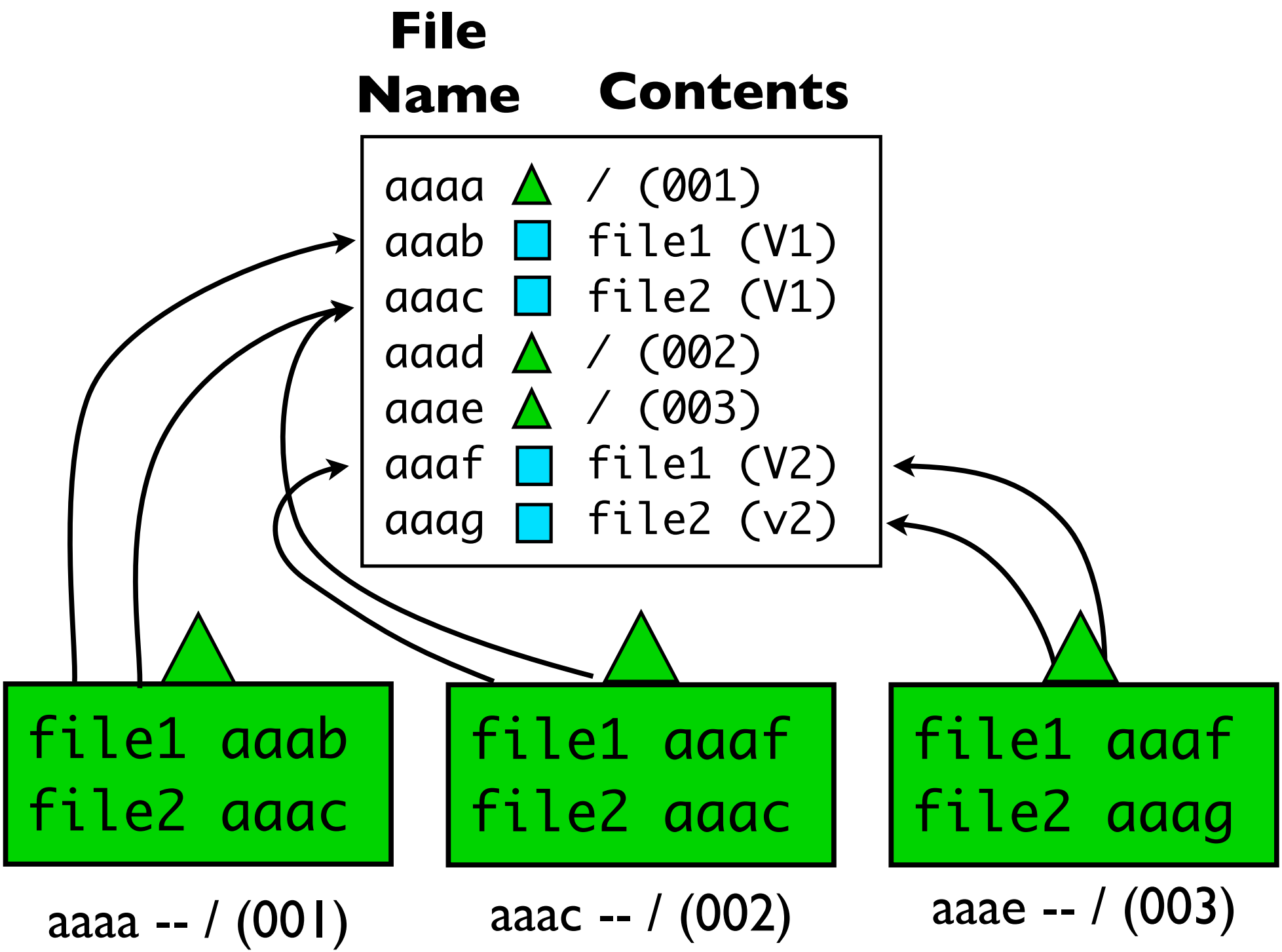

File Name
Contents
aaaa
/ (001)
aaab
file1 (V1)
aaac
file2 (V1)
aaad
/ (002)
aaae
/ (003)
aaaf
file1 (V2)
aaag
file2 (v2)

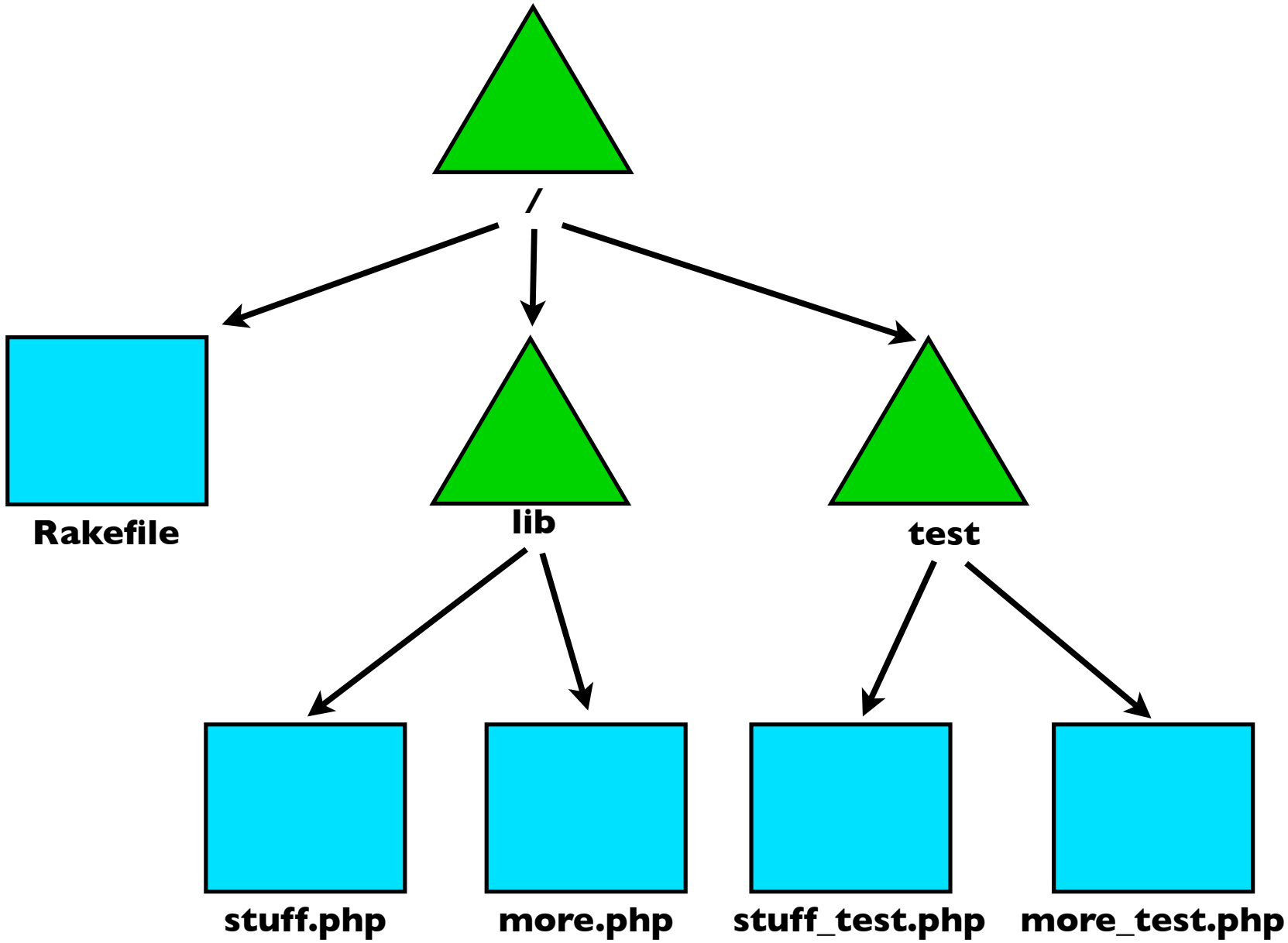

File Name
Contents
aaaa / (001)
aaab file1 (V1)
aaac file2 (V1)
aaad / (002)
aaae / (003)
aaaf file1 (V2)
aaag file2 (v2)
file1 aaaf
file2 aaac

aaad -- / (002)

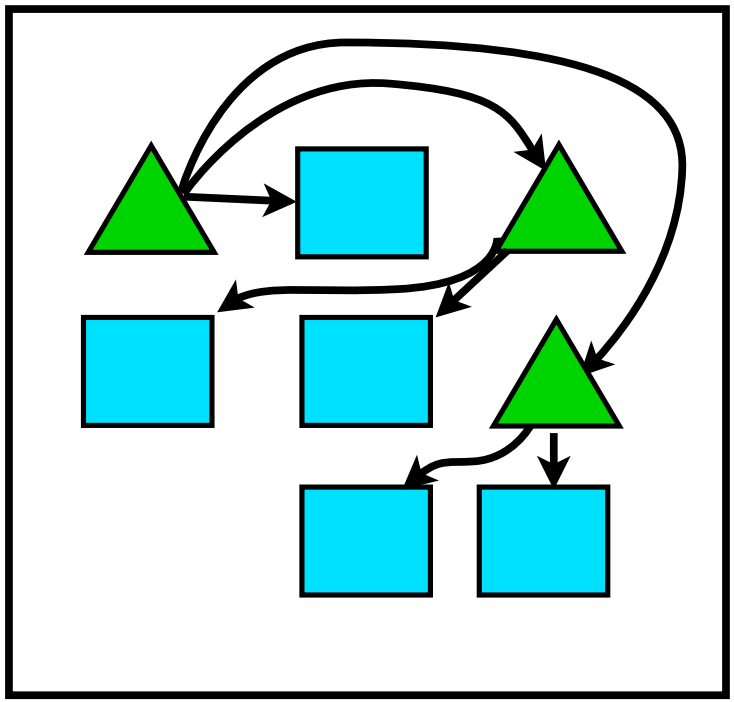

File Name
Contents
aaaa
/ (001)
aaab
file1 (V1)
aaac
file2 (V1)
aaad
/ (002)
aaae
/ (003)
aaaf
file1 (V2)
aaag
file2 (v2)


File Name
Contents
aaaa / (001)
aaab file1 (V1)
aaac file2 (V1)
aaad / (002)
aaae / (003)
aaaf file1 (V2)
aaag file2 (v2)
file1 aaaf
file2 aaag
aaae -- / (003)


File Name
Contents
aaaa / (001)
aaab file1 (V1)
aaac file2 (V1)
aaad / (002)
aaae / (003)
aaaf file1 (V2)
aaag file2 (v2)
file1 aaab
file2 aaac
file1 aaaf
file2 aaac
file1 aaaf
file2 aaag
aaaa -- / (001)
aaac -- / (002)
aaae -- / (003)


/
Rakefile
lib
test
stuff.php
more.php
stuff_test.php
more_test.php

Project Archive



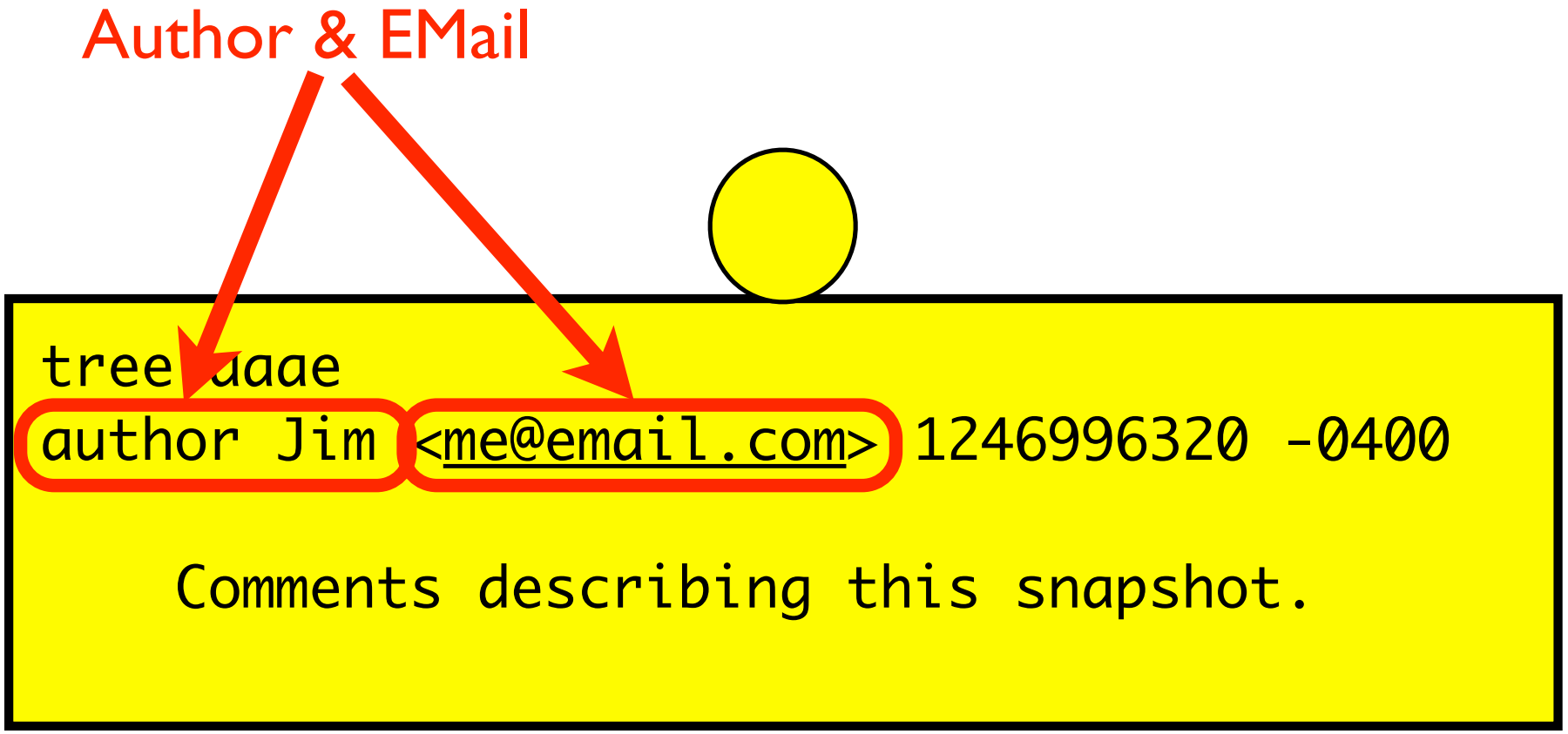
Metadata would be nice

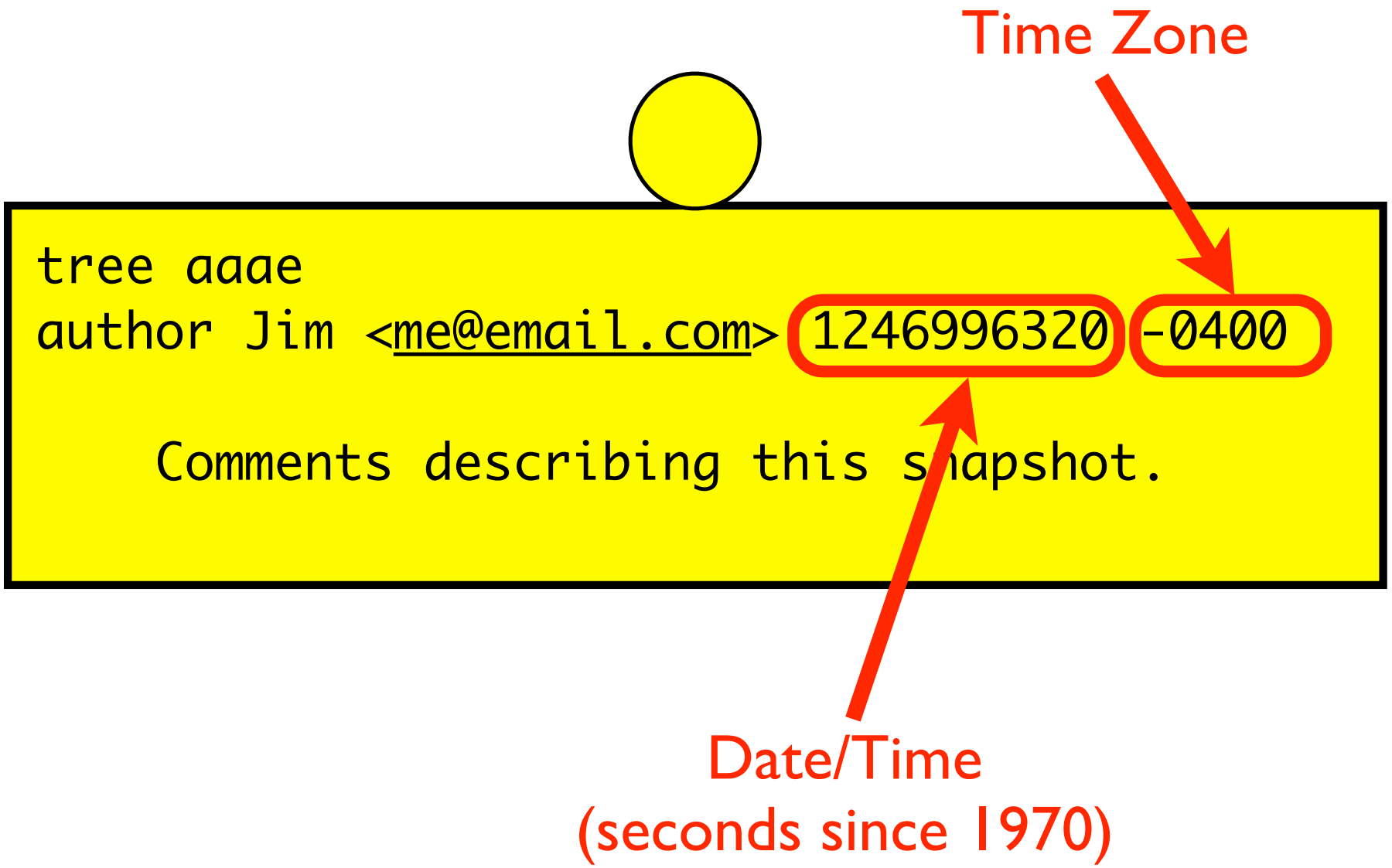
What Metadata?

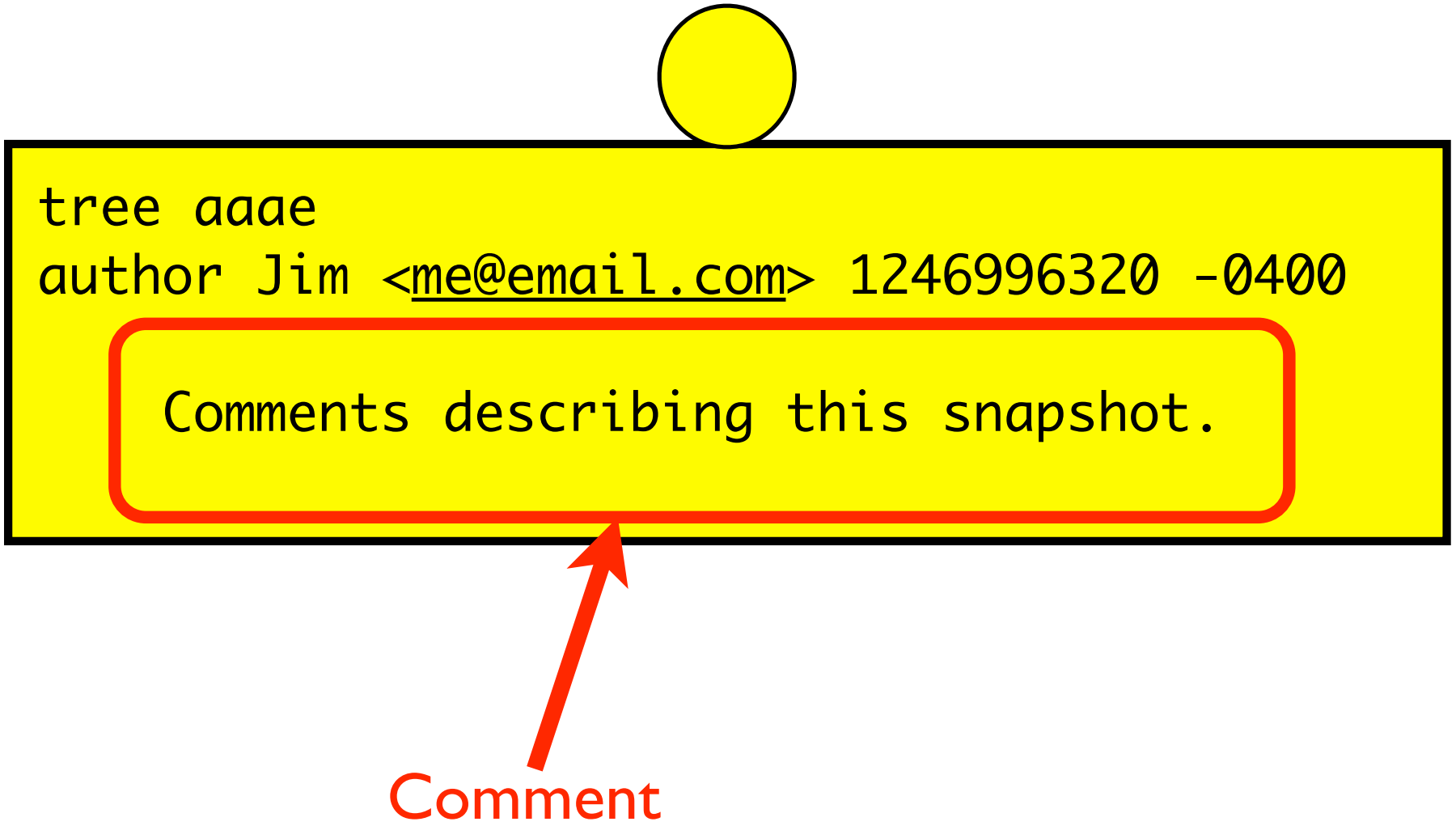
- The Root tree of the snapshot
- Name/EMail of the person commiting the snapshot
- Date/Time of the snapshot
- Some comments describing the snapshot

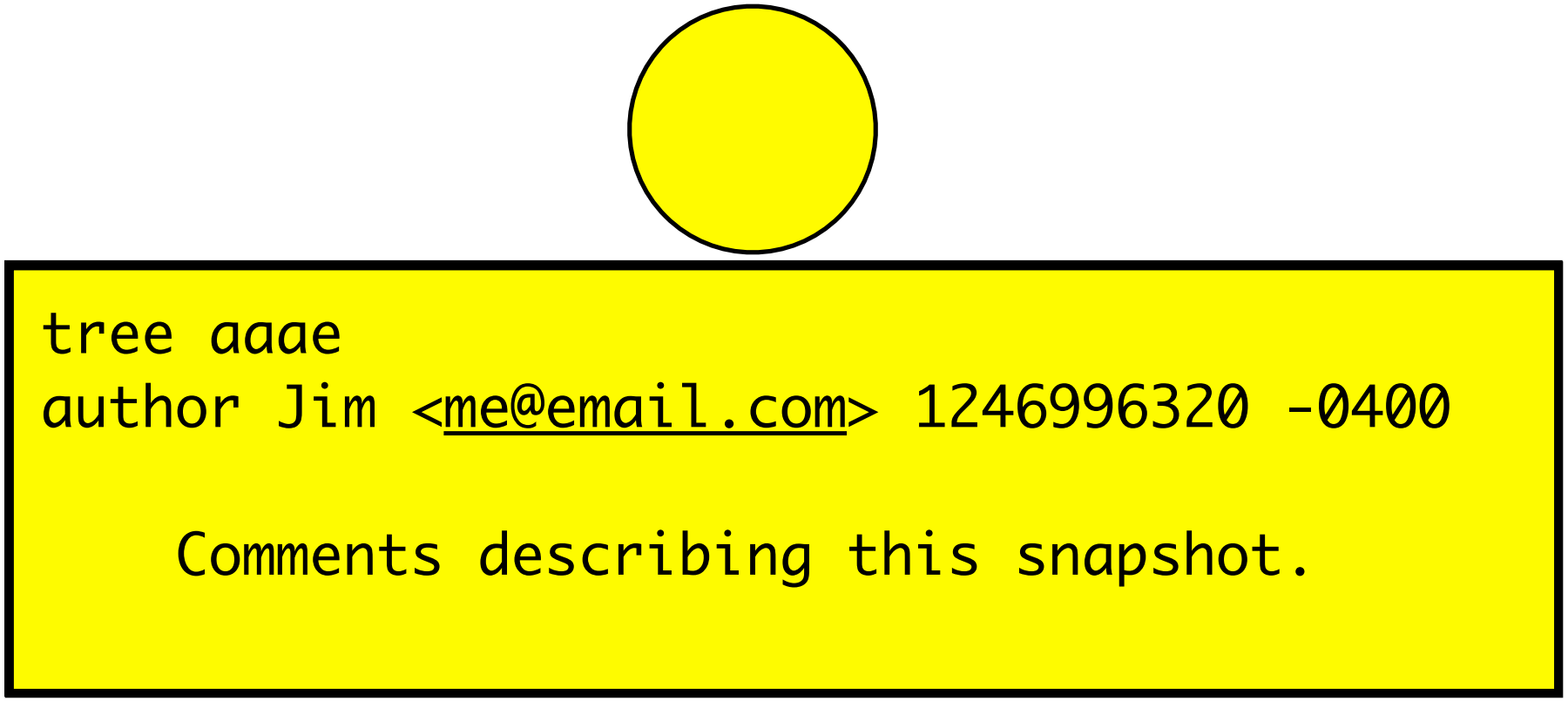

tree aaae
author Jim <me@email.com> 1246996320 -0400
Comments describing this snapshot.

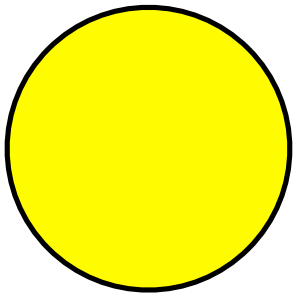

Root Manifest
for Snapshot
tree aaae
author Jim <me@email.com> 1246996320 -0400
Comments describing this snapshot.

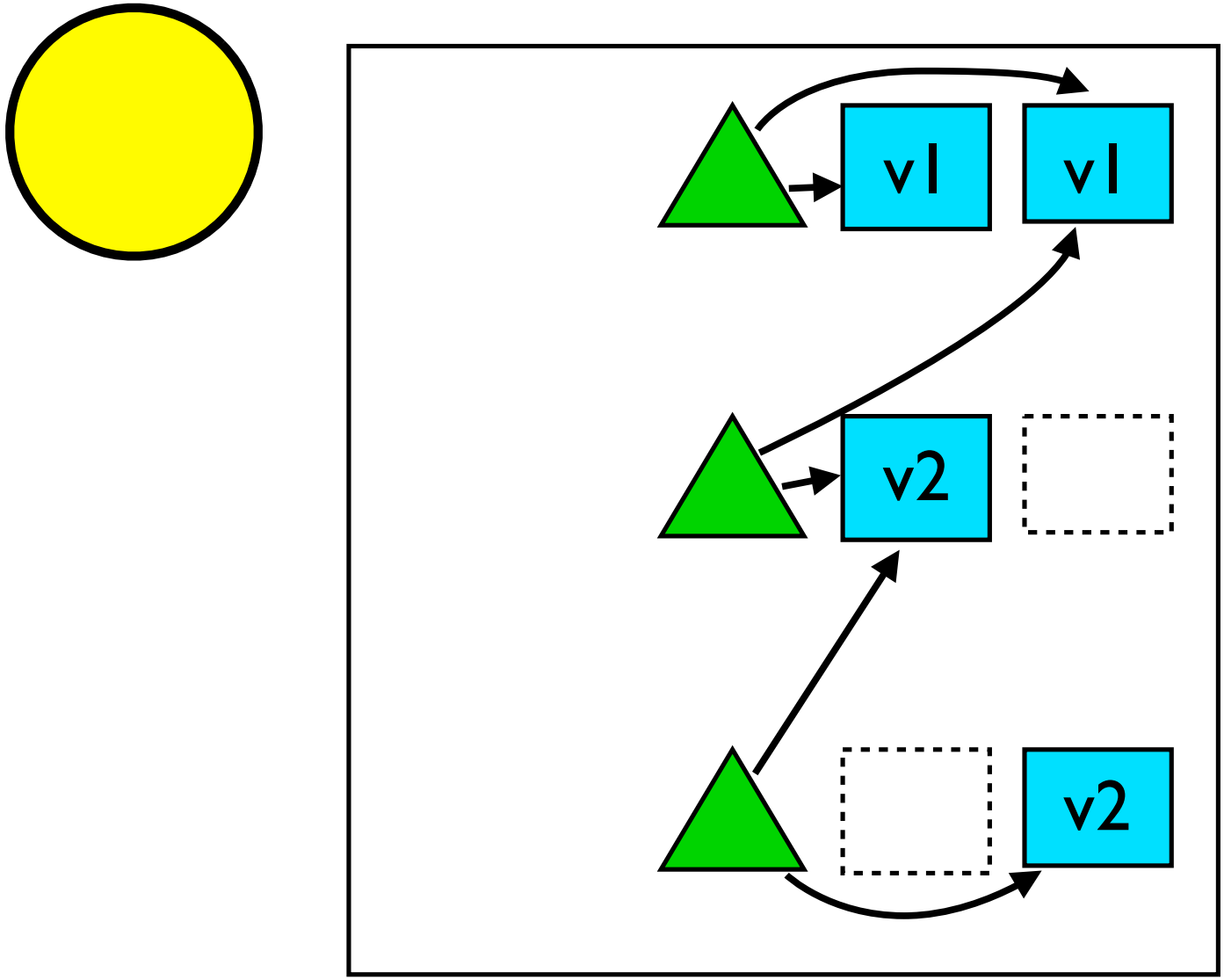

Author & EMail
tree aaae
author Jim <me@email.com> 1246996320 -0400
Comments describing this snapshot.

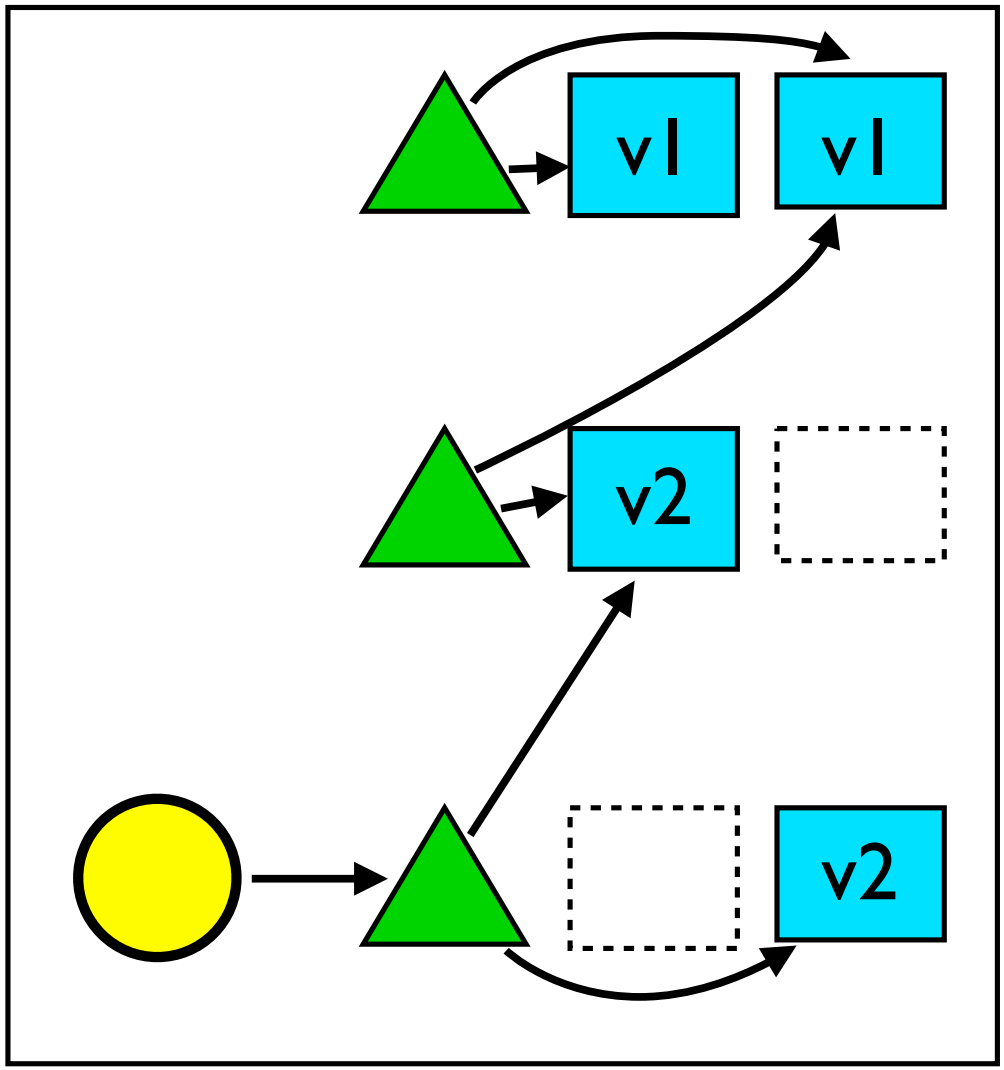

Time Zone
tree aaae
author Jim <me@email.com> 1246996320 -0400
Comments describing this snapshot.
Date/Time
(seconds since 1970)

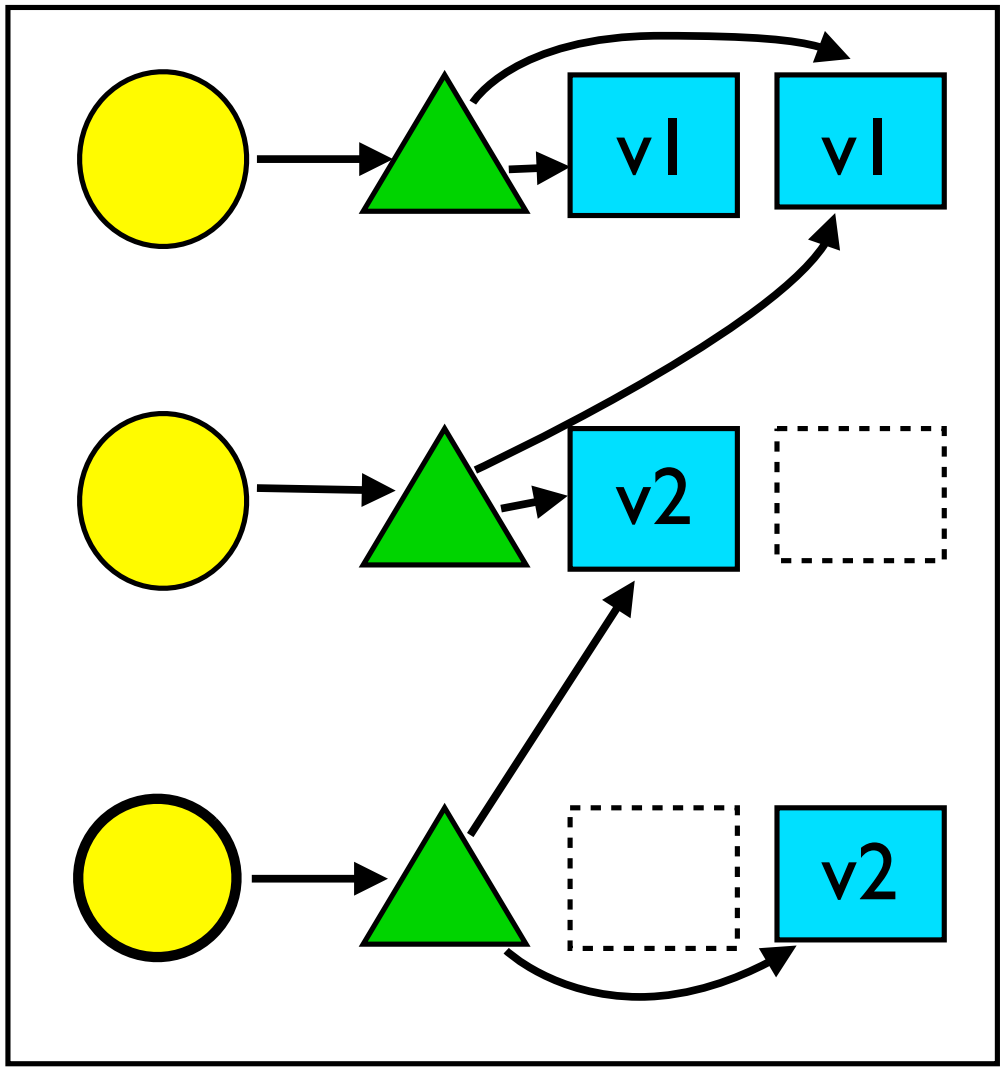

tree aaae
author Jim <me@email.com> 1246996320 -0400
Comments describing this snapshot.
Comment

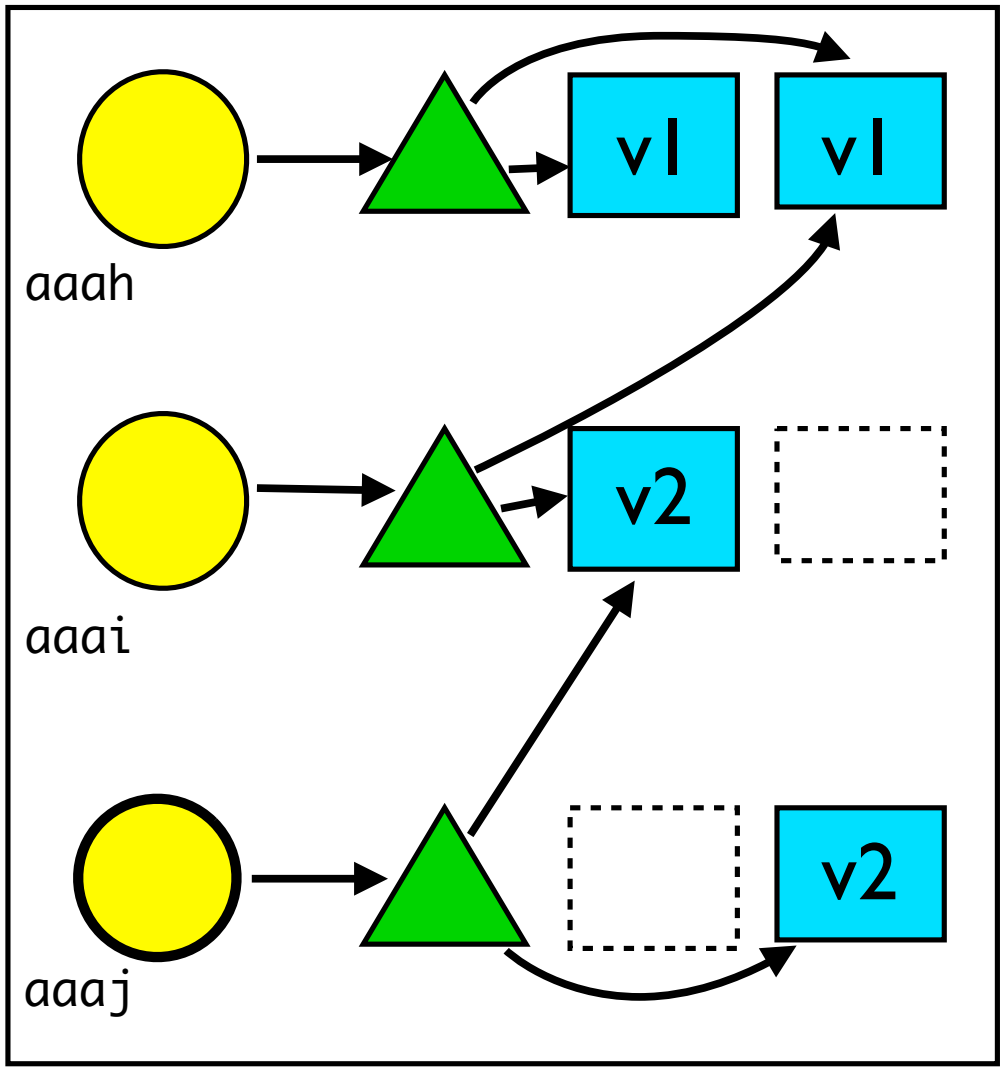

tree aaae
author Jim <me@email.com> 1246996320 -0400
Comments describing this snapshot.




v1
v1
v2
v2


v1
v1
v2
v2


v1
v1
v2
v2


v1
v1
aaah
v2
aaai
v2
aaaj

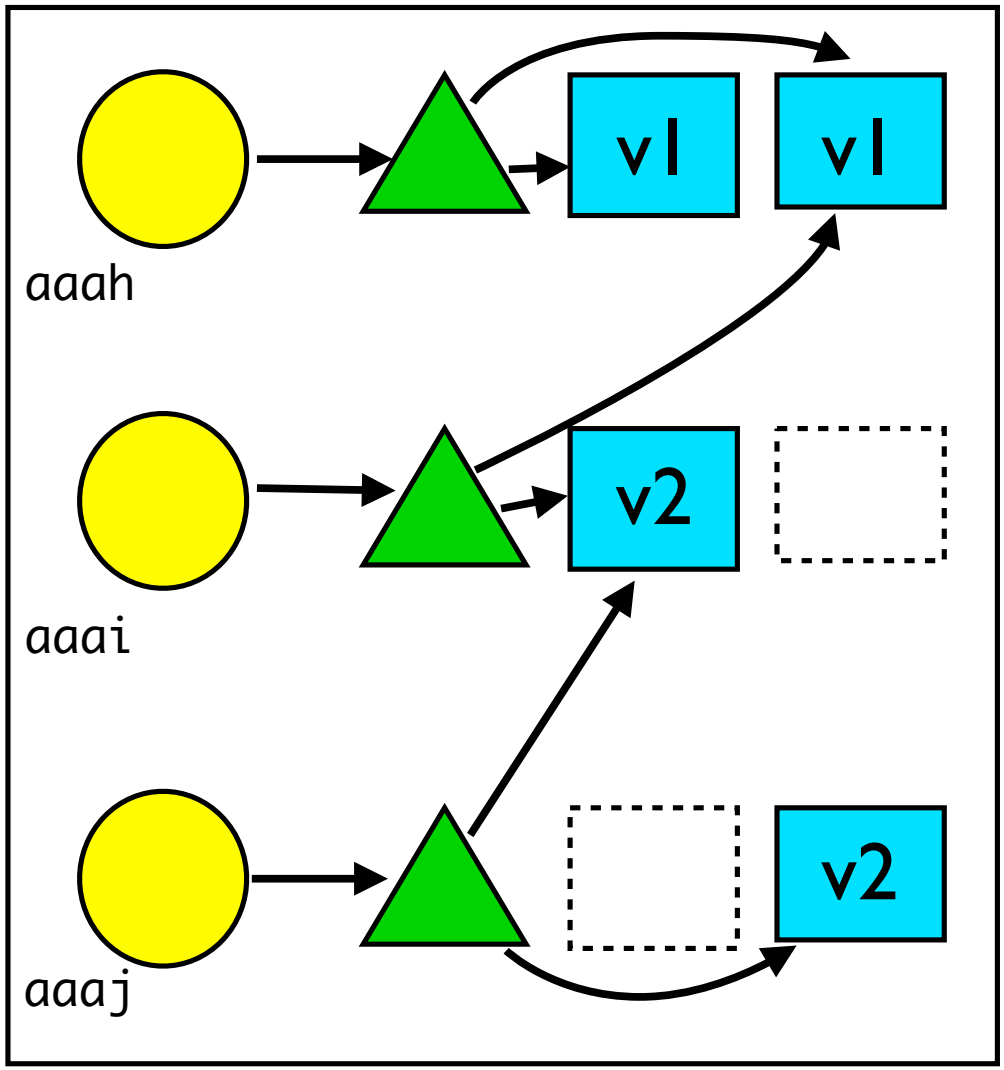
```
tree aaae
author Jim <me@email.com> 1246996320 -0400

    Comments describing this snapshot.
```

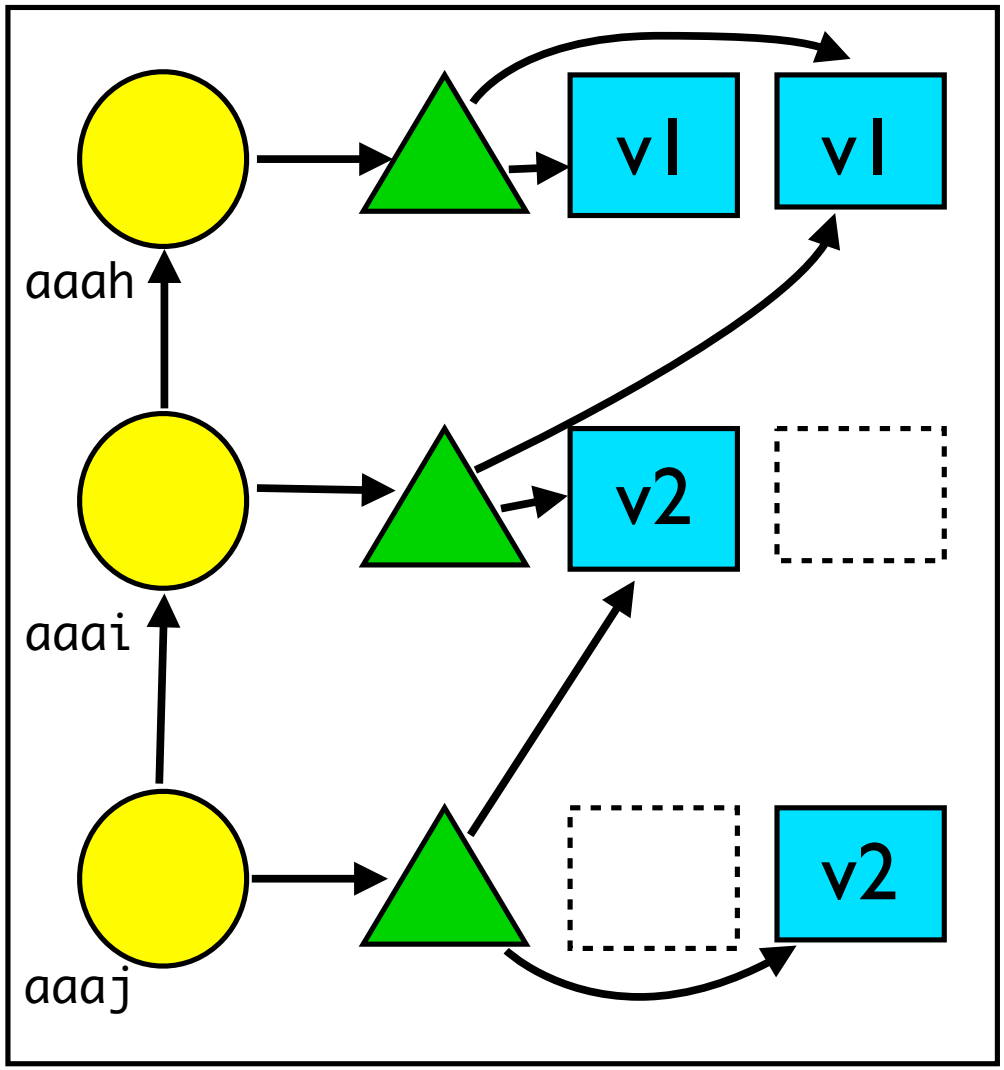
```
tree aaae
parent aaai
author Jim <me@email.com> 1246996320 -0400

    Comments describing this snapshot.
```

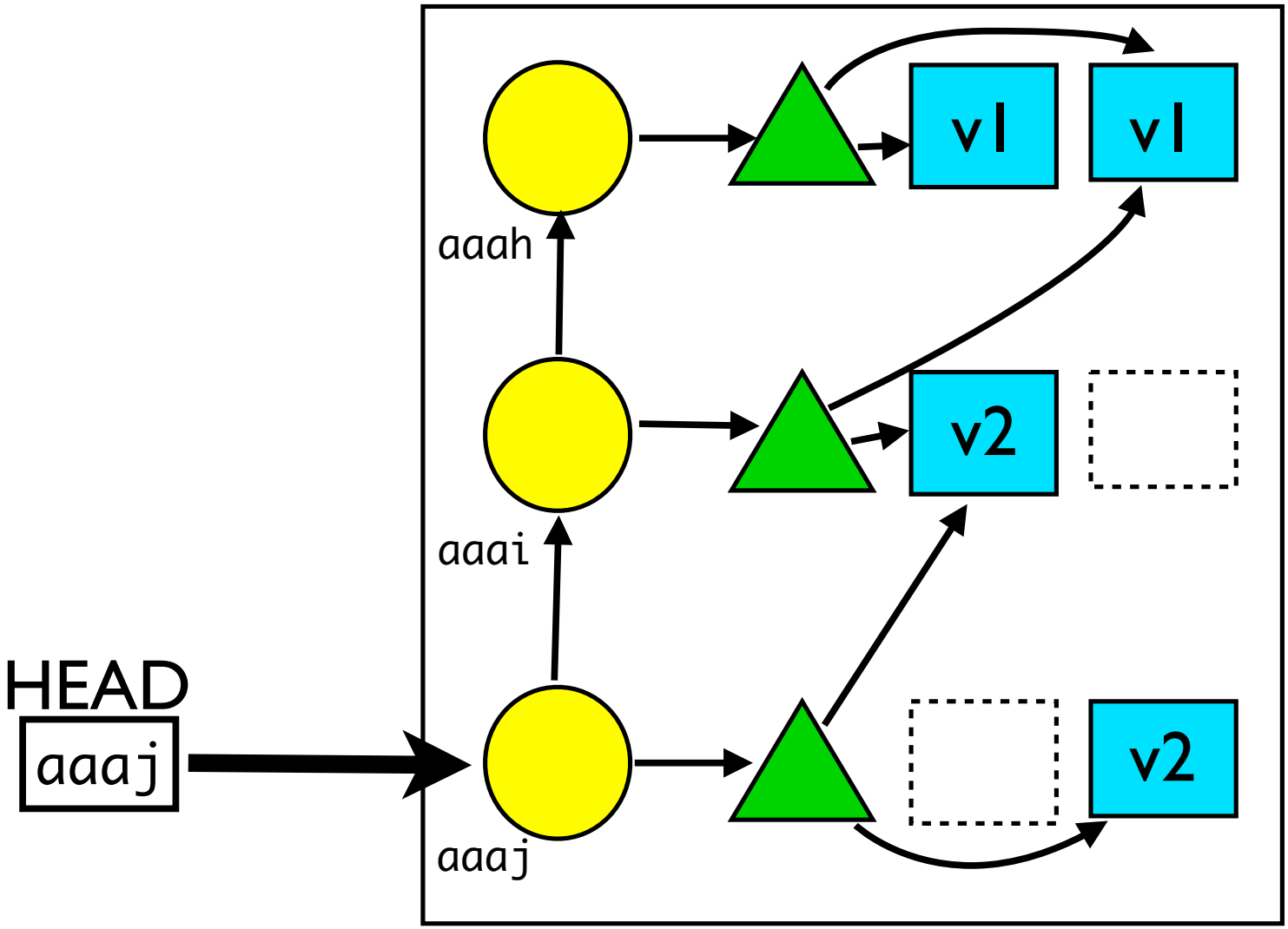
Project Archive



Project Archive

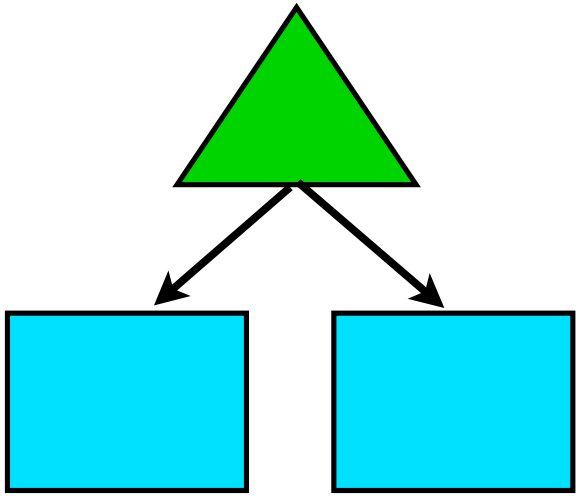


Project Archive



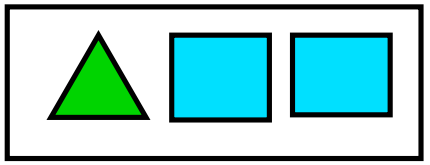
Summary So Far ...

Working Directory

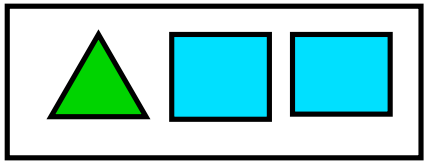


Project Archive

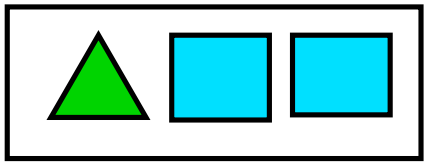
001.tar



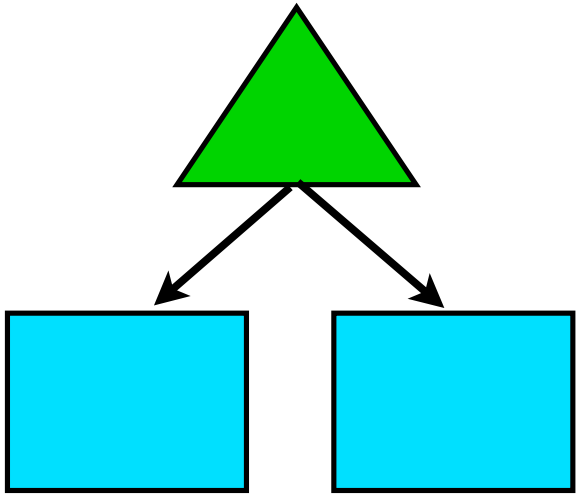
002.tar



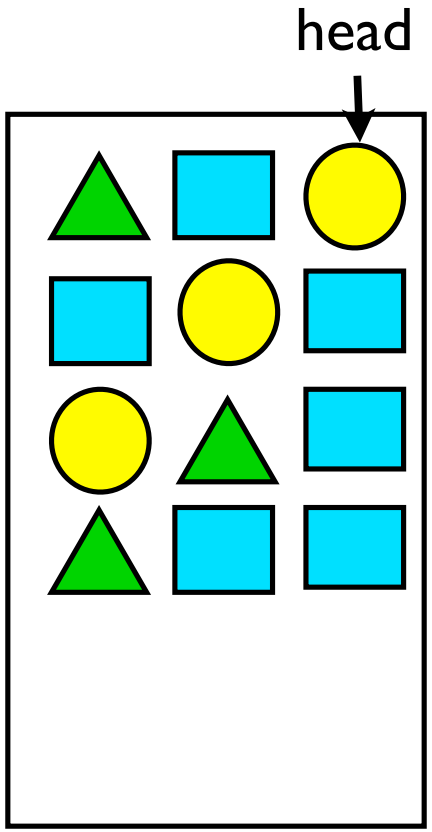
003.tar



Working Directory

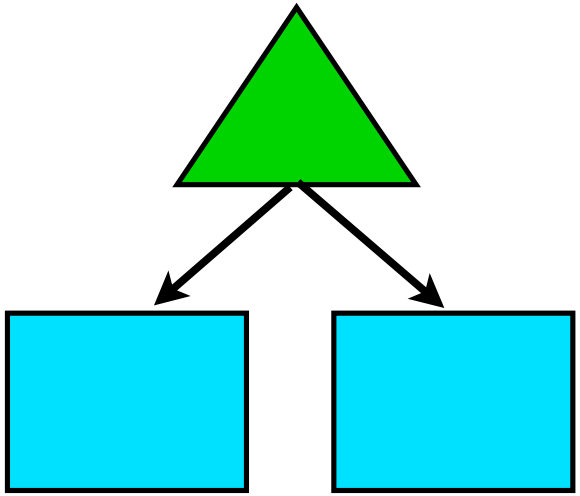


Project Archive

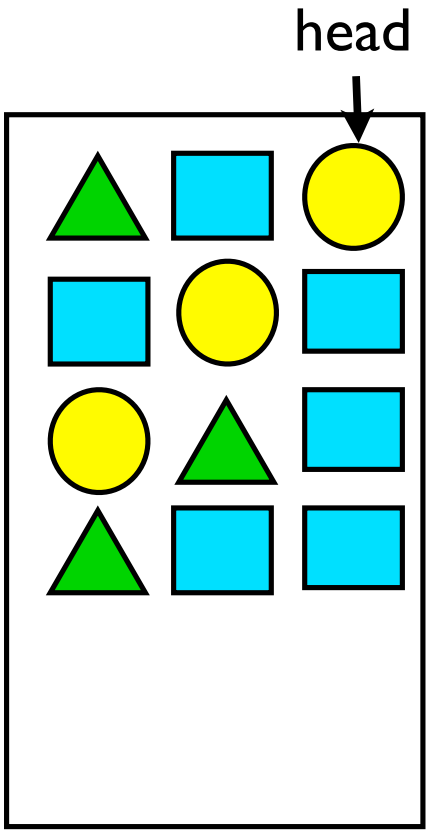


How do we create a snapshot now?

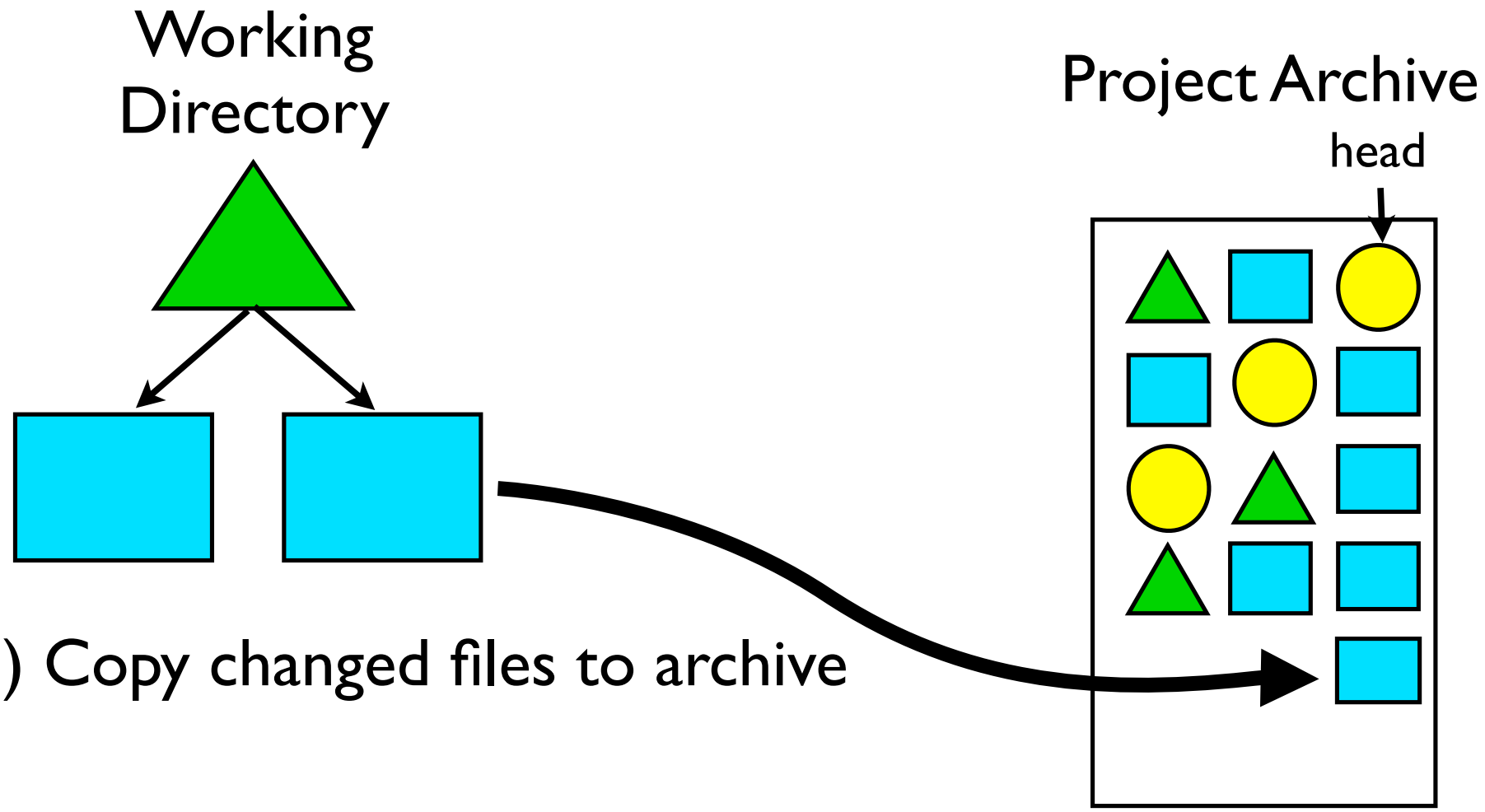
Working Directory

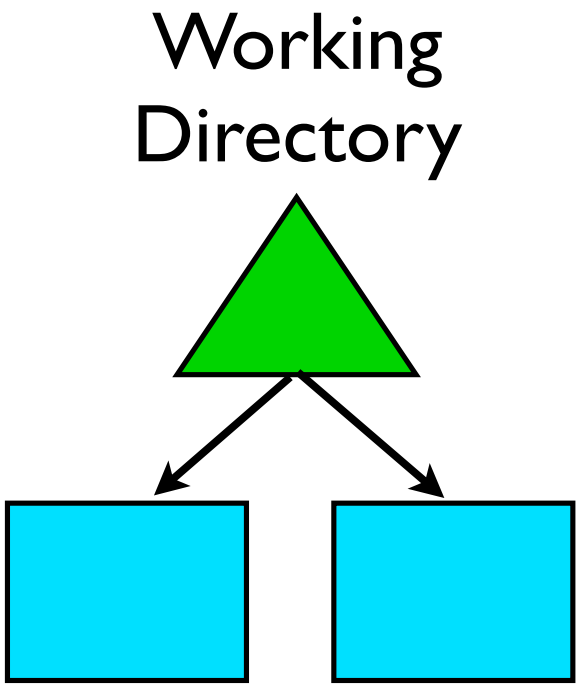


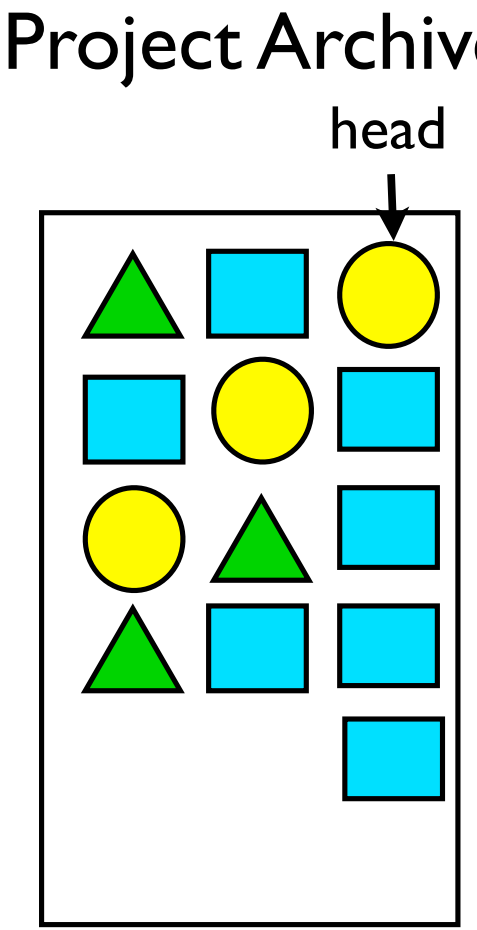
Project Archive



(1) Copy changed files to archive

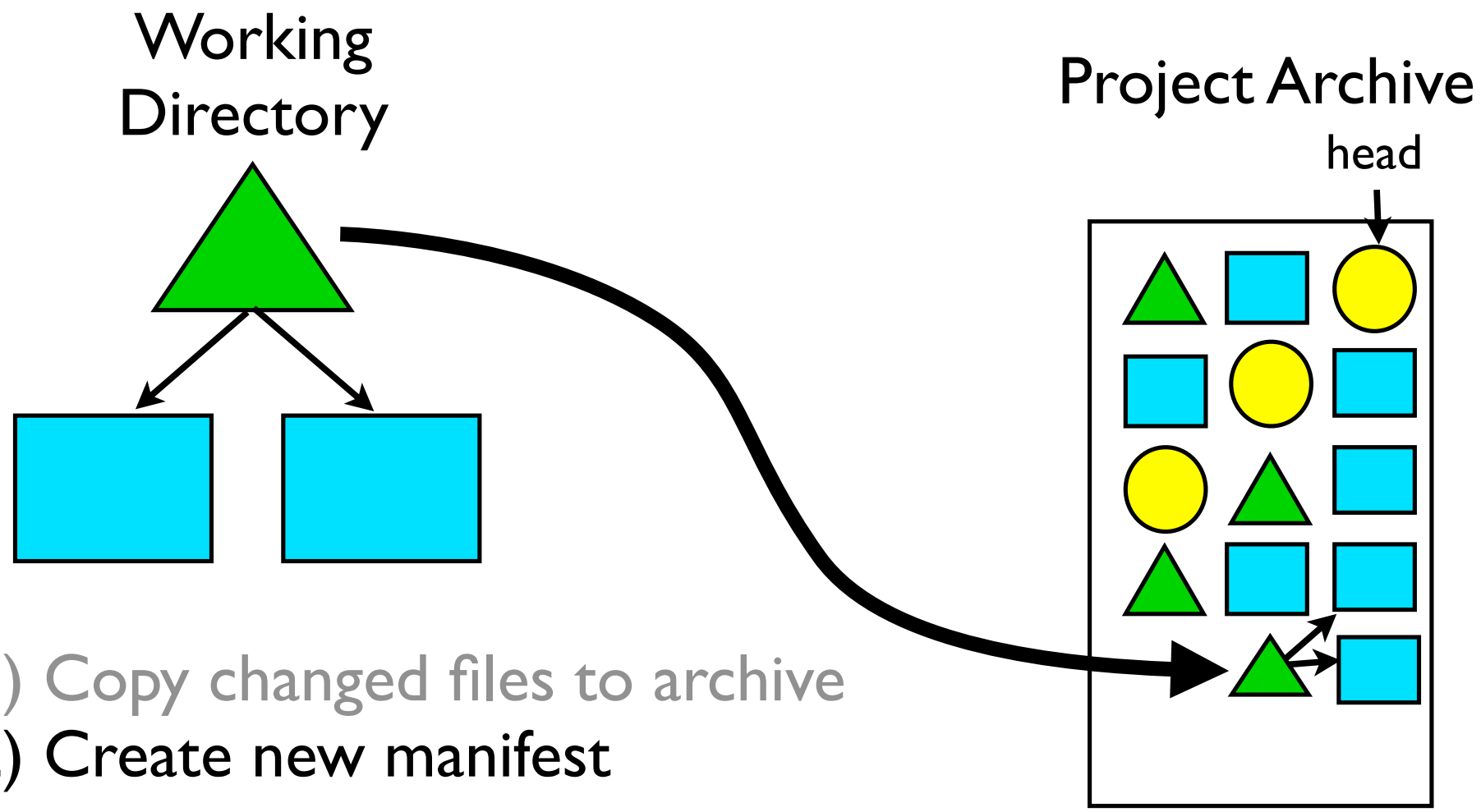

Working
Directory
Project Archive
head
(1) Copy changed files to archive





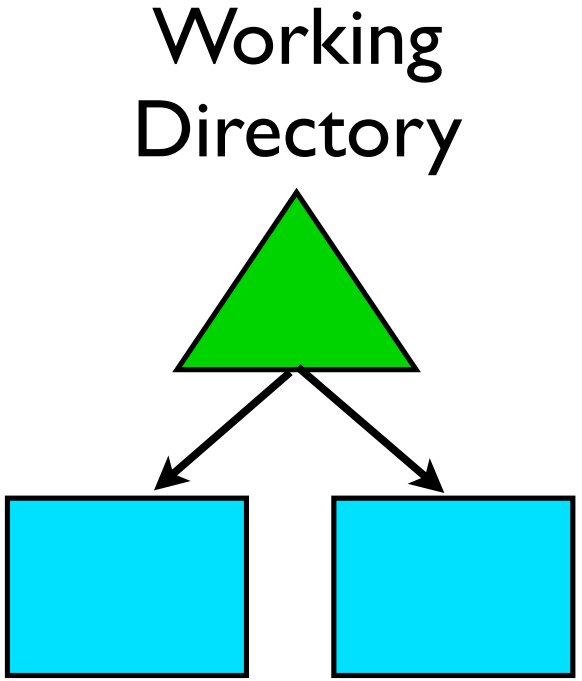
(1) Copy changed files to archive

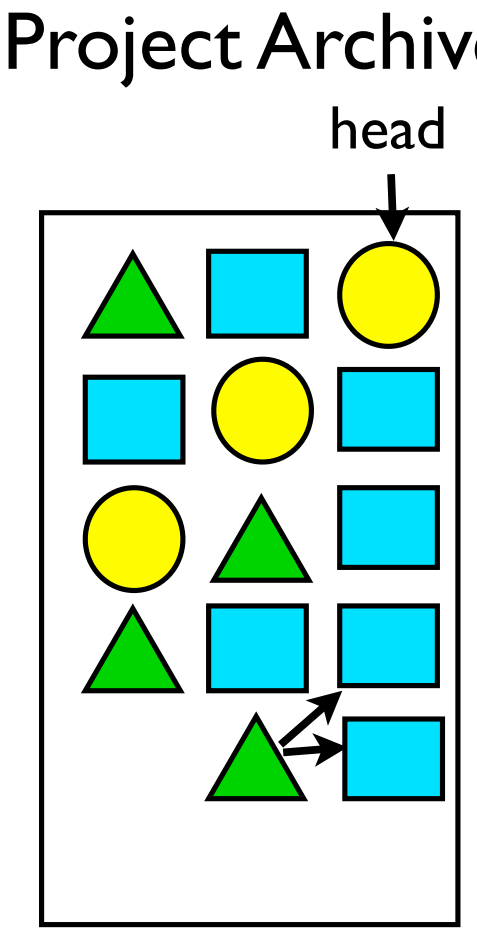
(2) Create new manifest



(1) Copy changed files to archive

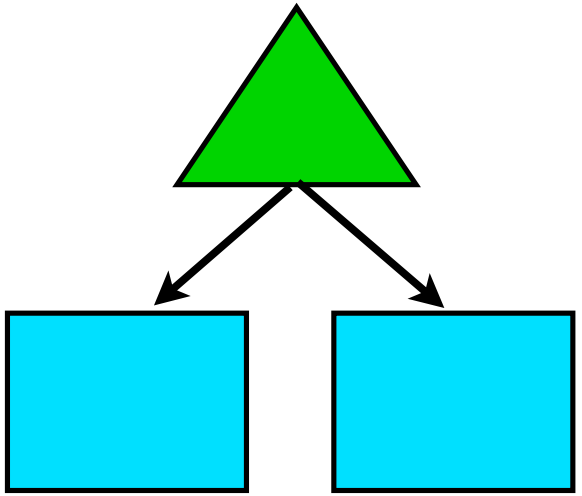
(2) Create new manifest



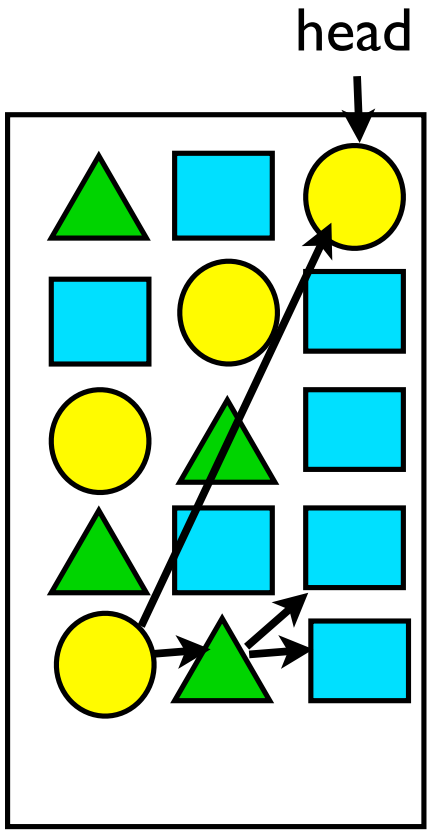


(1) Copy changed files to archive
(2) Create new manifest
(3) Create new snapshot record

Working Directory

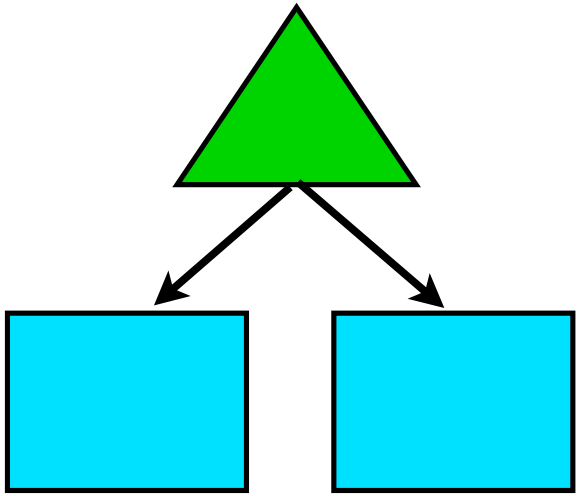


Project Archive

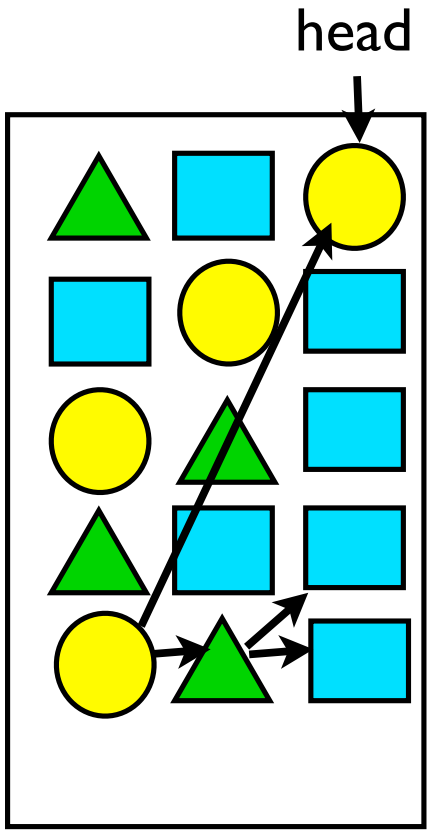


(1) Copy changed files to archive
(2) Create new manifest
(3) Create new snapshot record

Working Directory

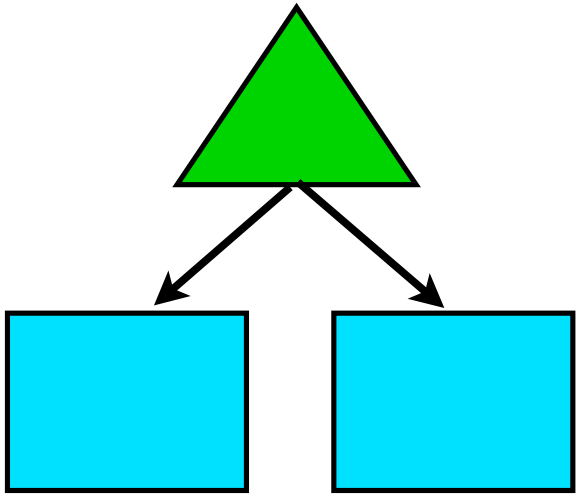


Project Archive

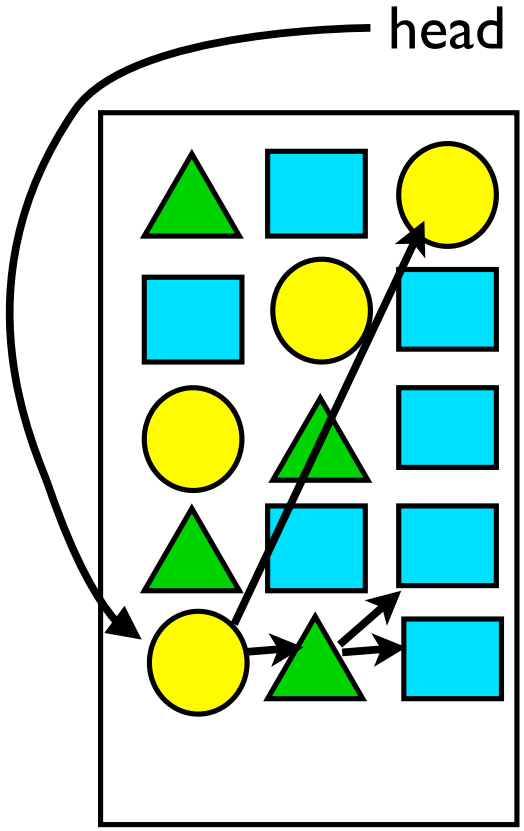


(1) Copy changed files to archive
(2) Create new manifest
(3) Create new snapshot record
(4) Readjust HEAD

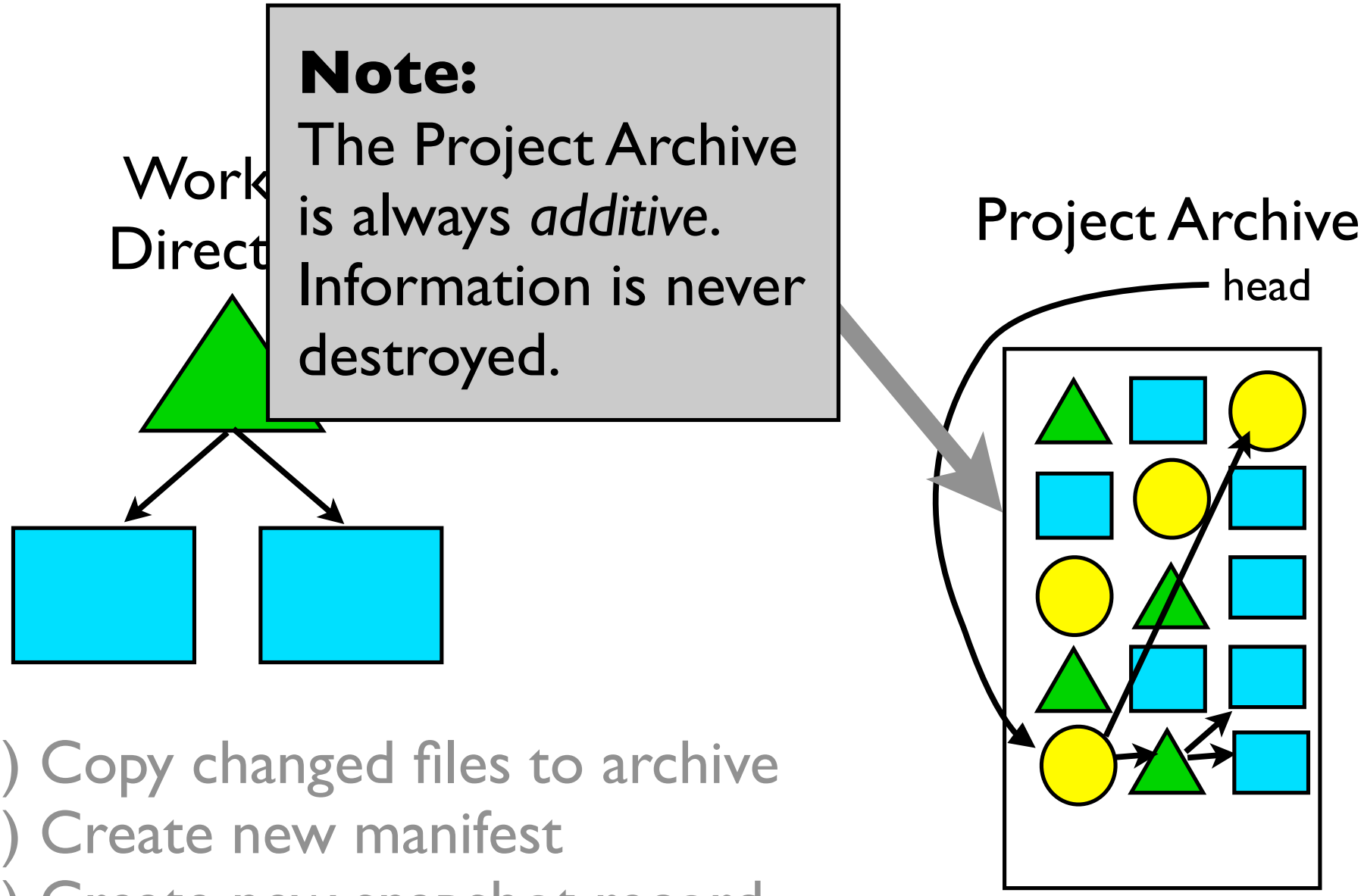
Working Directory



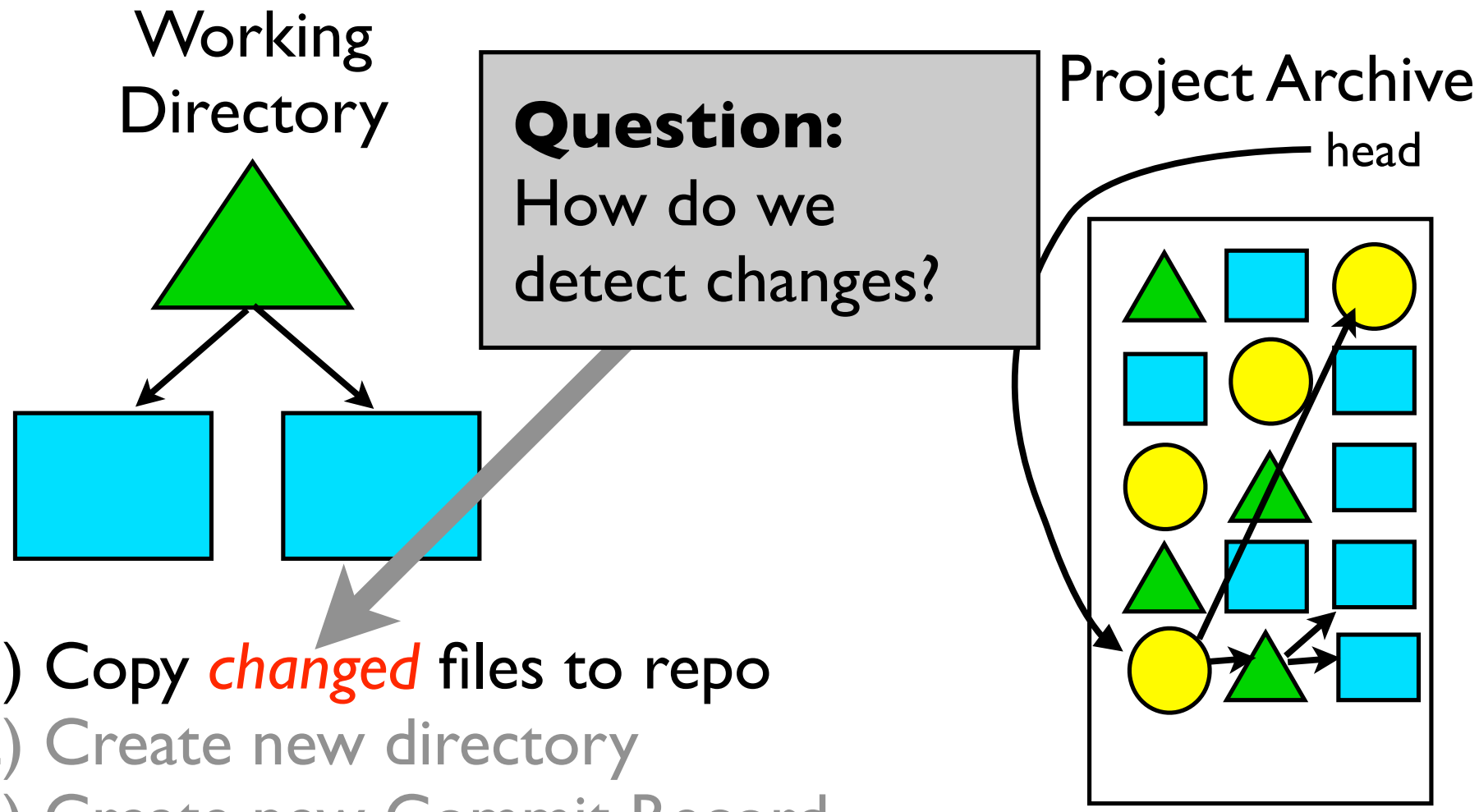
Project Archive



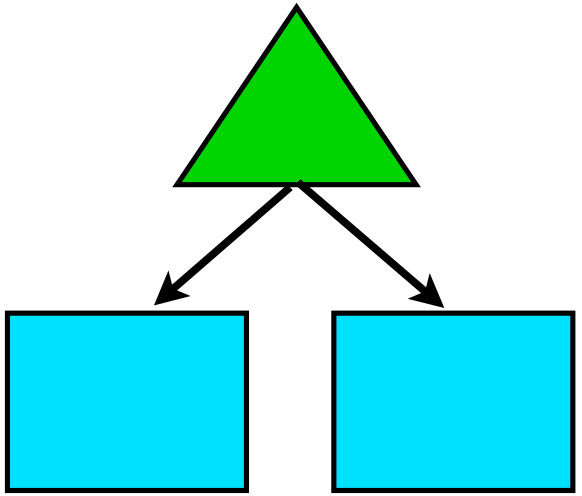
(1) Copy changed files to archive
(2) Create new manifest
(3) Create new snapshot record
(4) Readjust HEAD



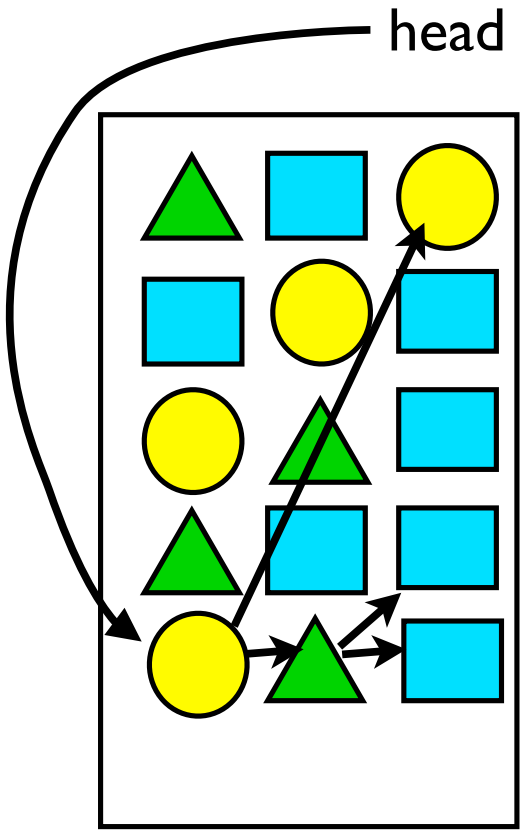
(1) Copy changed files to archive
(2) Create new manifest
(3) Create new snapshot record
(4) Readjust HEAD


Working Directory
Question:
How do we detect changes?
Project Archive
head
(1) Copy changed files to repo
(2) Create new directory
(3) Create new Commit Record
(4) Readjust HEAD

Working Directory

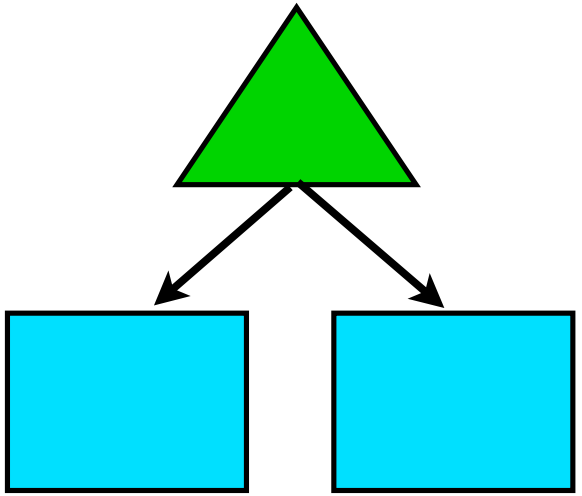


Project Archive

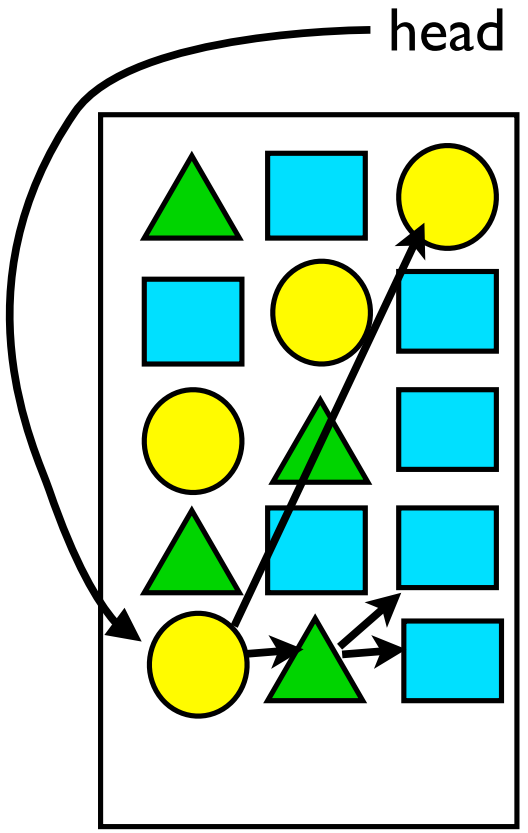


(1) Copy *changed* files to repo
(2) Create new directory
(3) Create new Commit Record
(4) Readjust HEAD

Working Directory

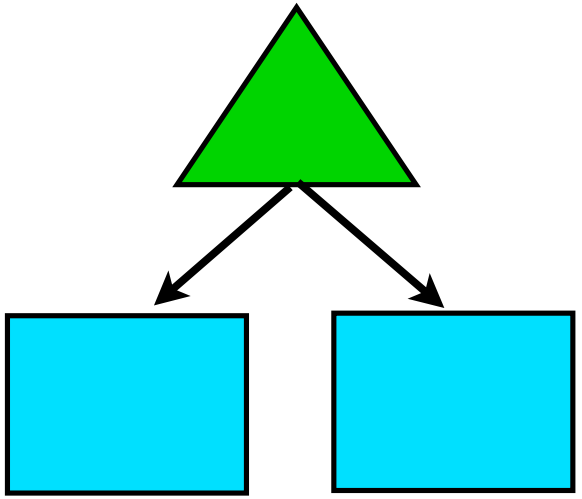


Project Archive

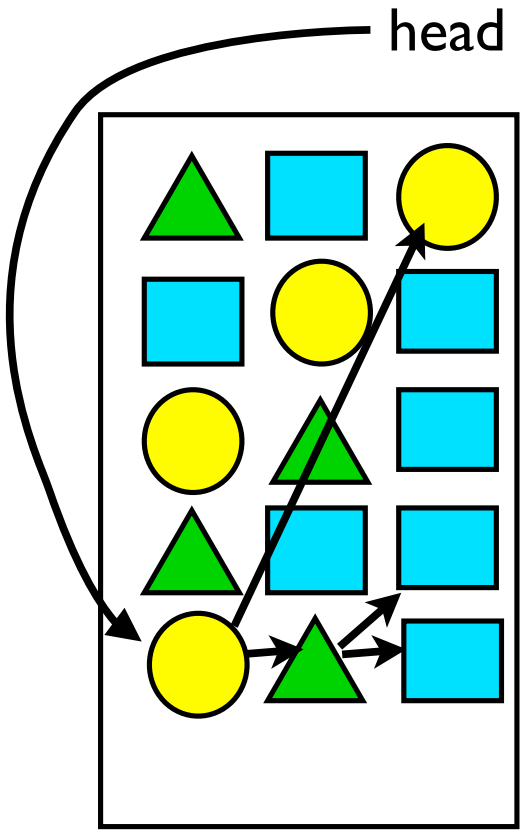


(1) Copy *changed* files to repo
(2) Create new directory
(3) Create new Commit Record
(4) Readjust HEAD

Working Directory

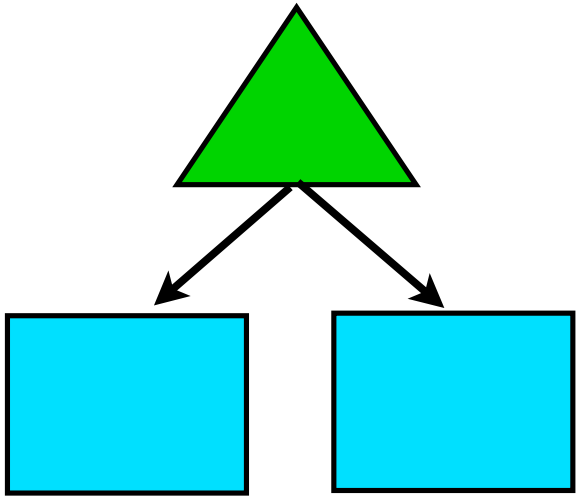


Project Archive

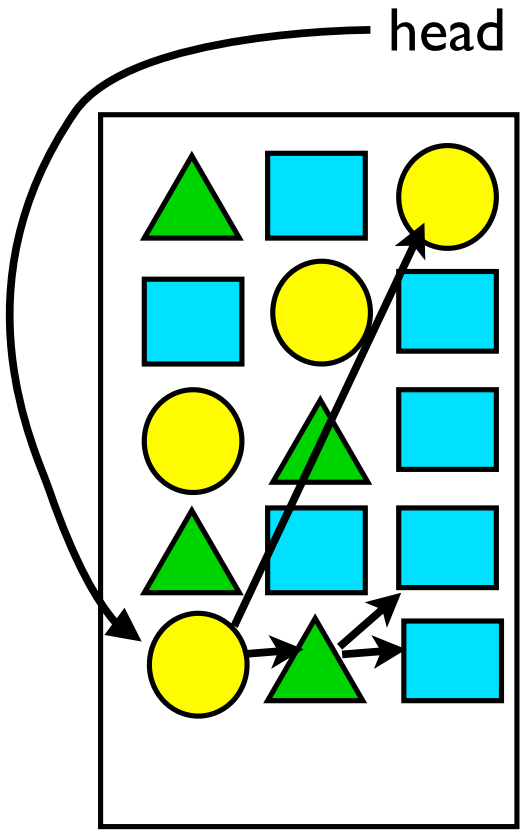


(1) Copy *changed* files to repo
(2) Create new directory
(3) Create new Commit Record
(4) Readjust HEAD

Working Directory



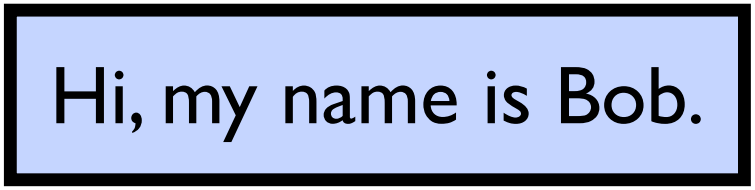
Project Archive



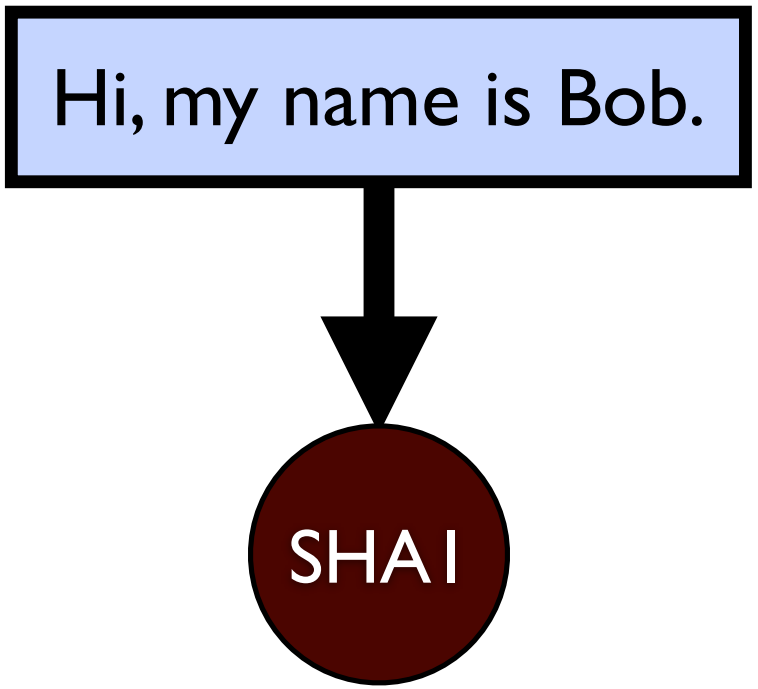
(1) Copy *changed* files to repo
(2) Create new directory
(3) Create new Commit Record
(4) Readjust HEAD

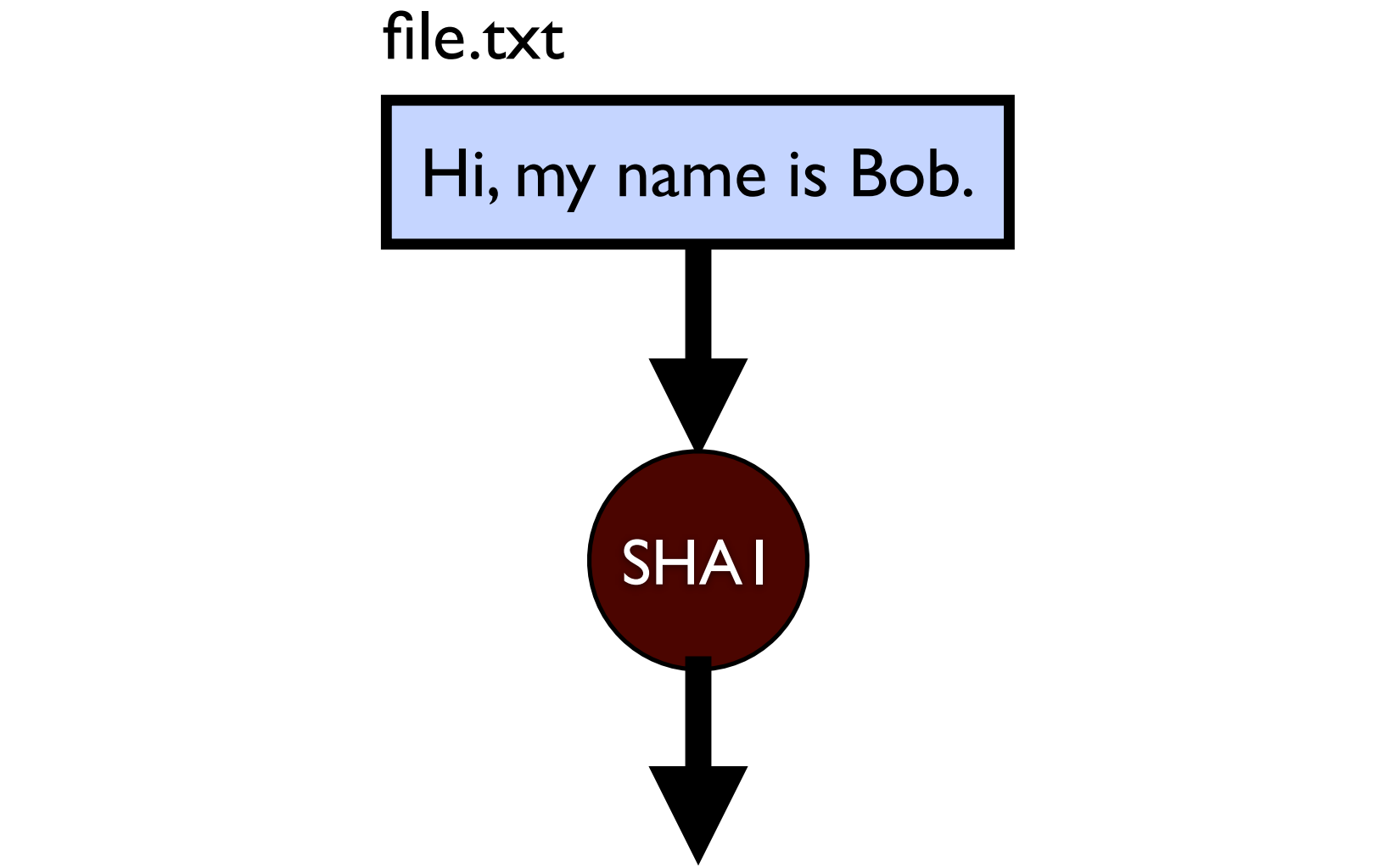
Detecting Duplication

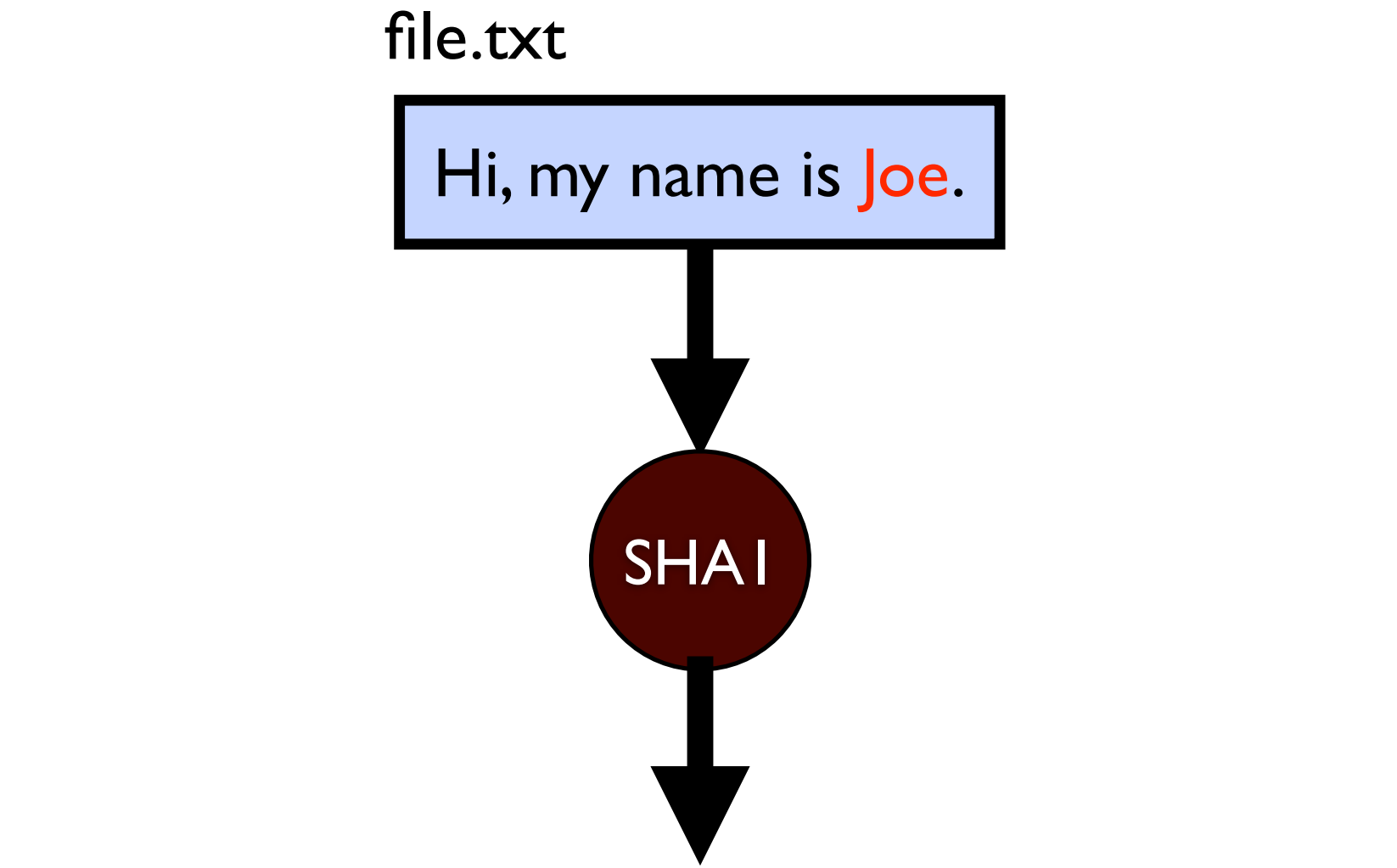
file.txt

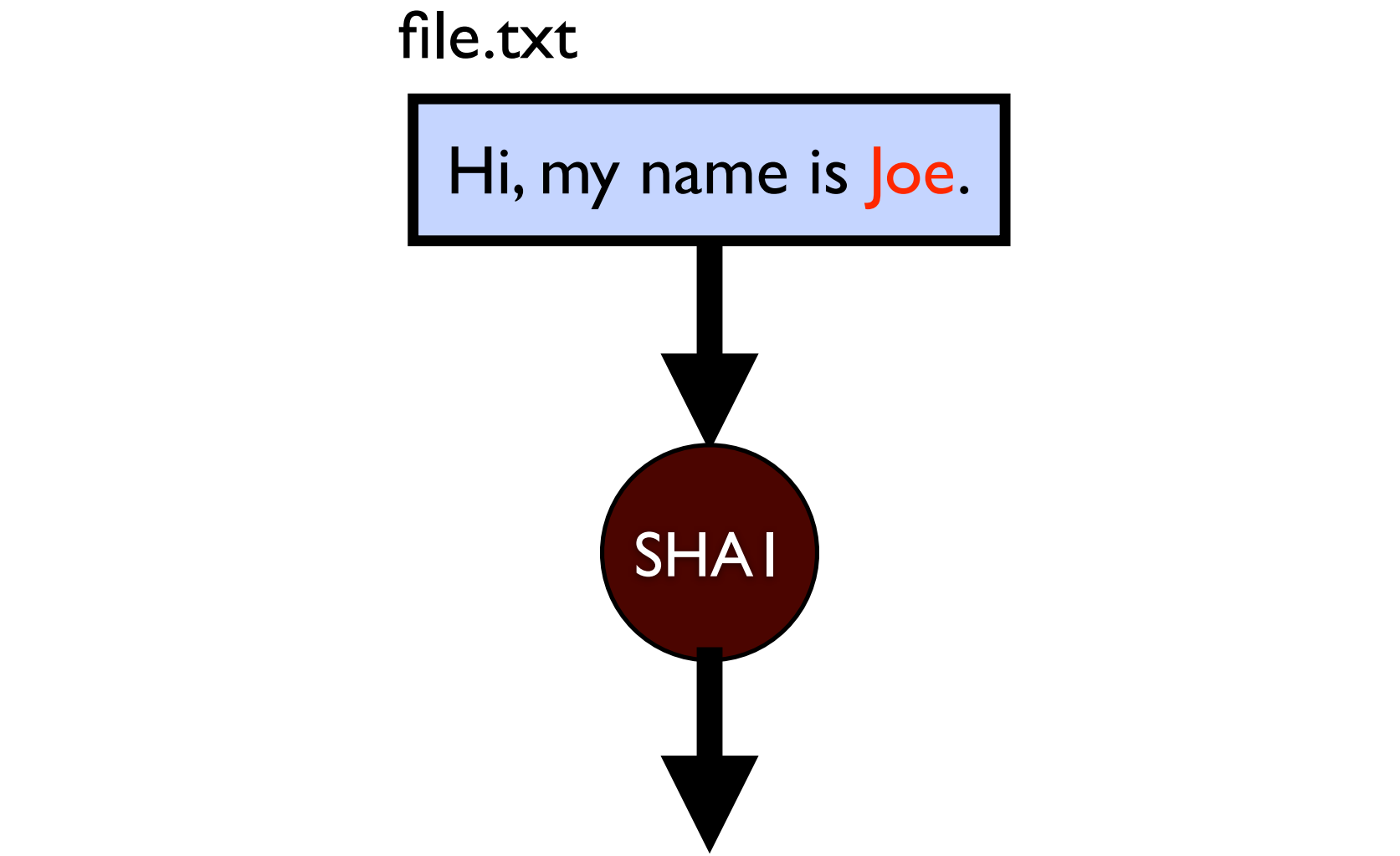


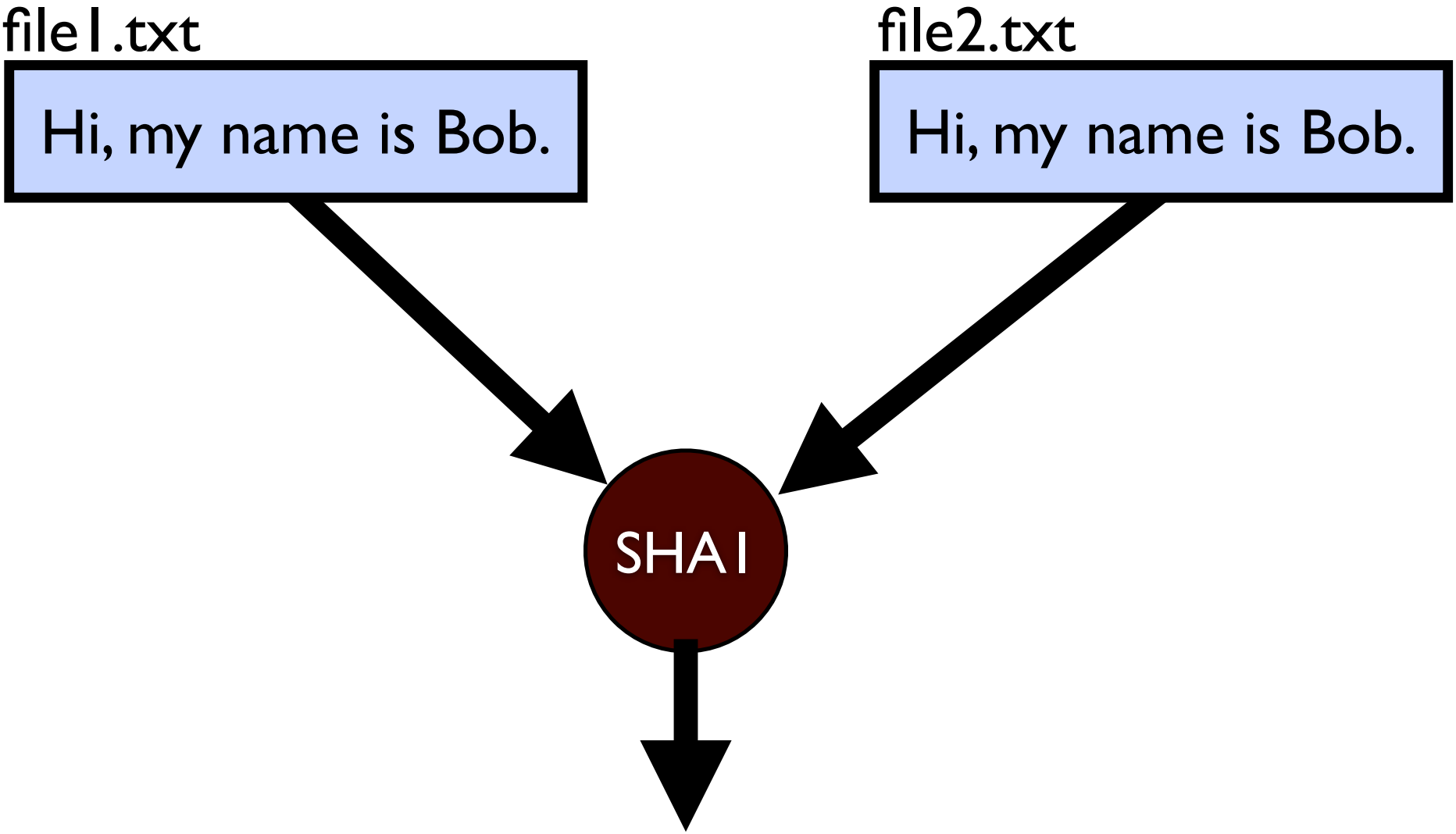
file.txt

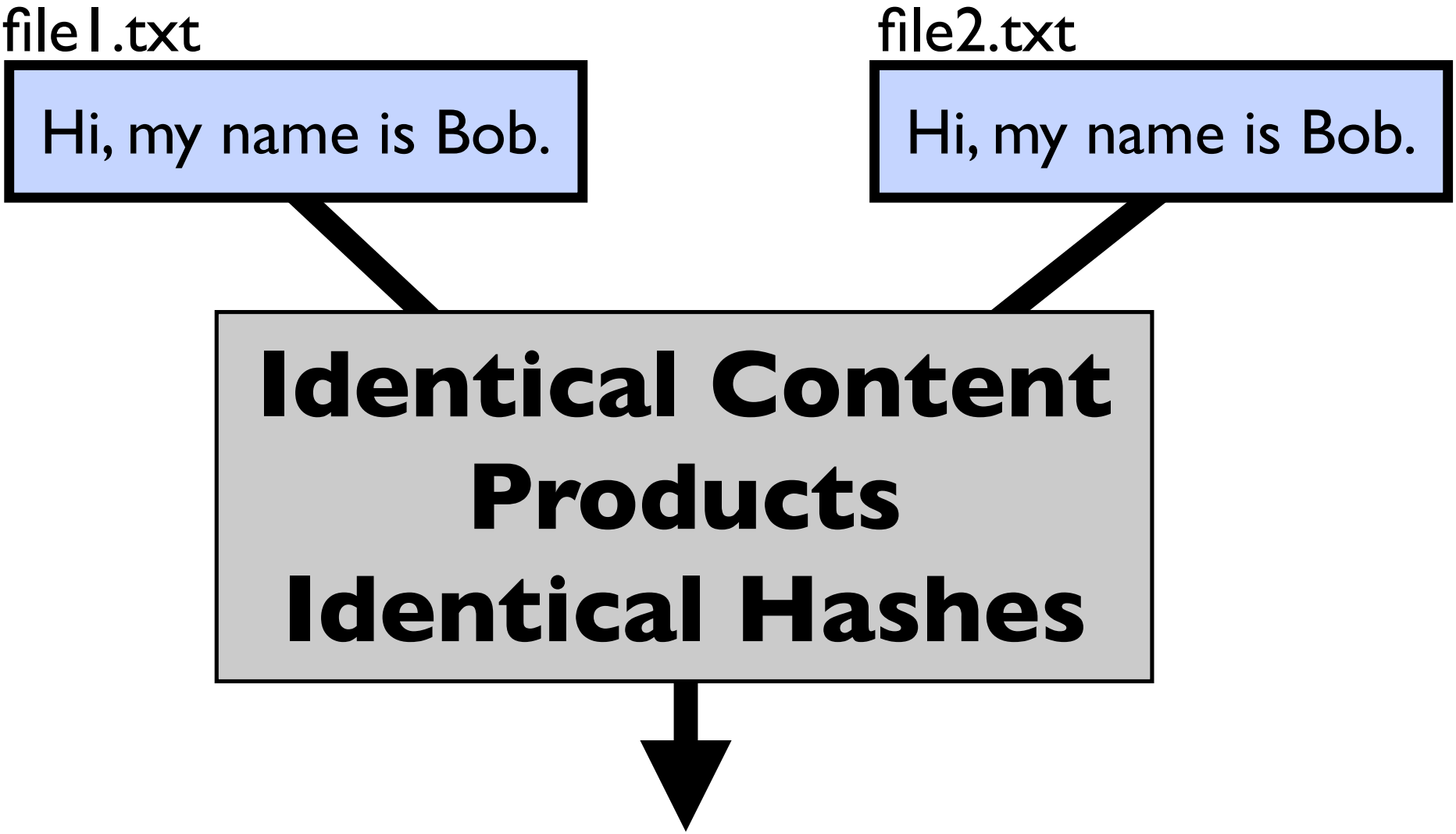



file.txt
Hi, my name is Bob.
SHA1
"55dc96ede241511e35f1973957bbd51f37f3762b"

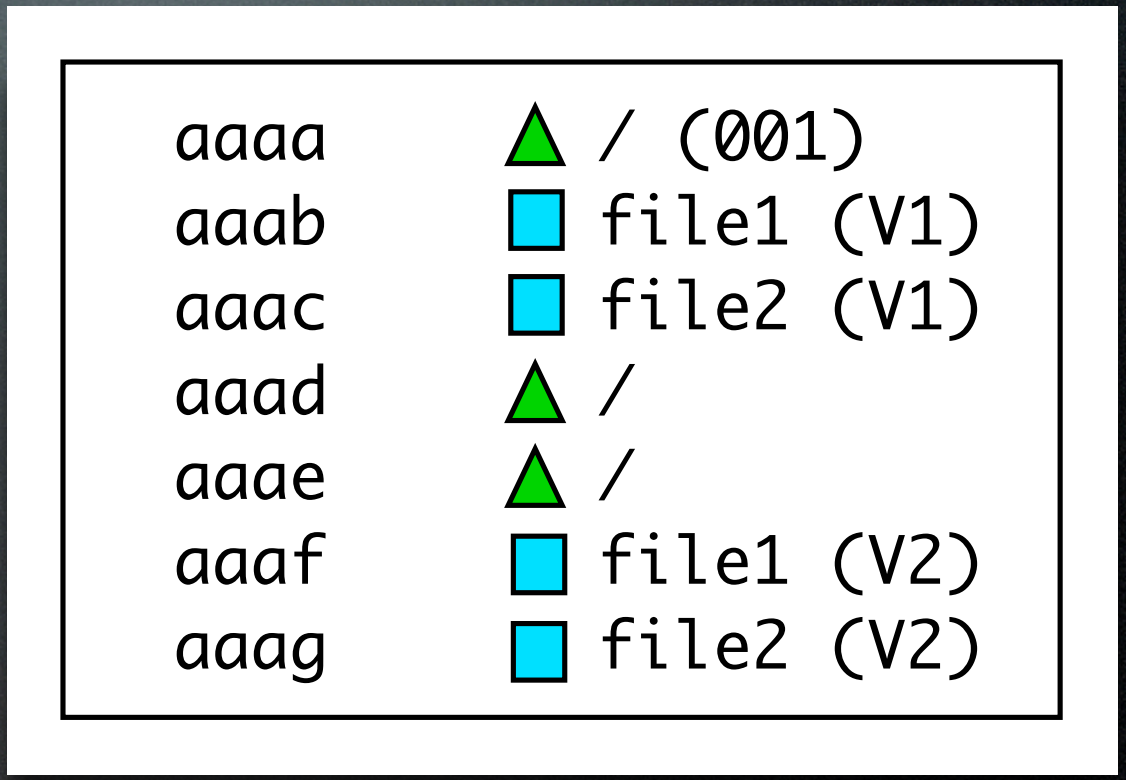

file.txt
Hi, my name is Joe.
SHA1
"55dc96ede24151 1e35f1973957bbd51f37f3762b"


file.txt
Hi, my name is Joe.
SHA1
"55dc96ede2415116e35f1973957bbd51f37f3762b"
"53f2509e877de3cad85c547a0c2a22b43c5d81eb"


file1.txt
Hi, my name is Bob.
file2.txt
Hi, my name is Bob.
SHA1
"55dc96ede241511e35f1973957bbd51f37f3762b"

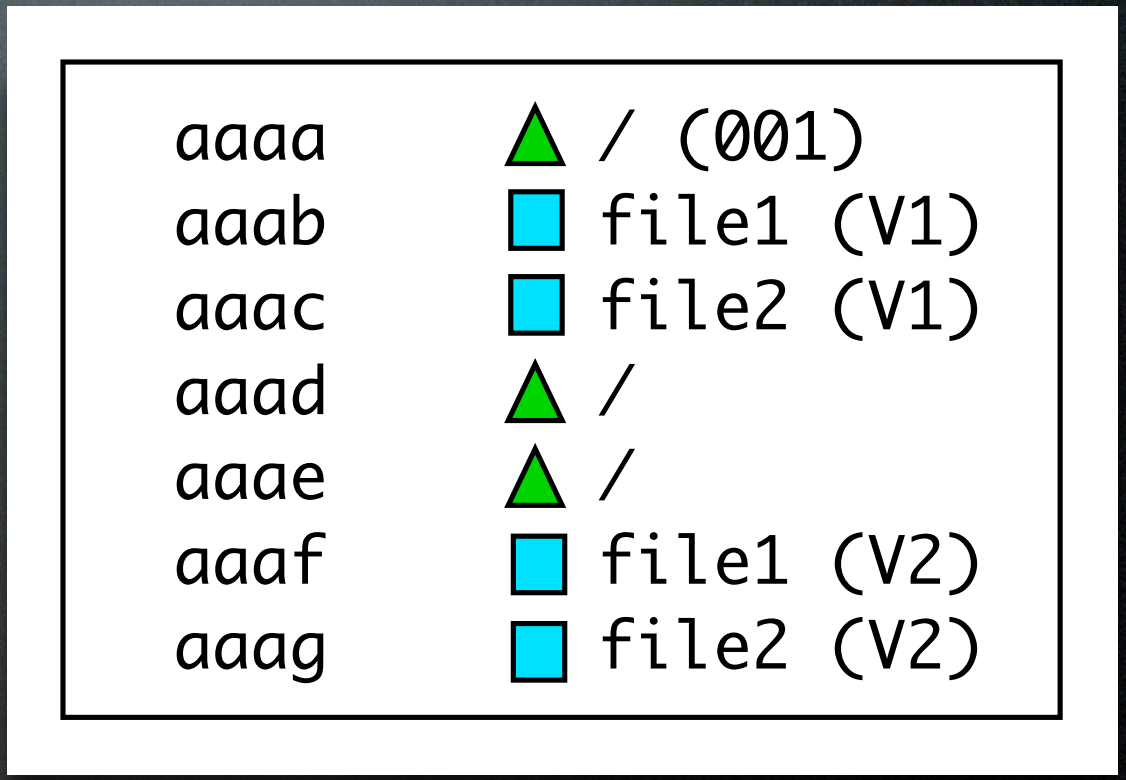

file1.txt
Hi, my name is Bob.
file2.txt
Hi, my name is Bob.
Identical Content
Products
Identical Hashes
"55dc96ede24151 1e35f1973957bbd51f37f3762b"

File Names in the Archive



File Names in the Archive

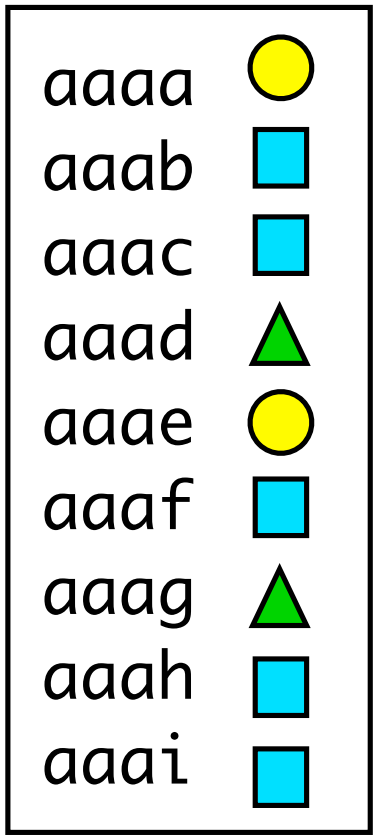
- Arbitrary
- Unique



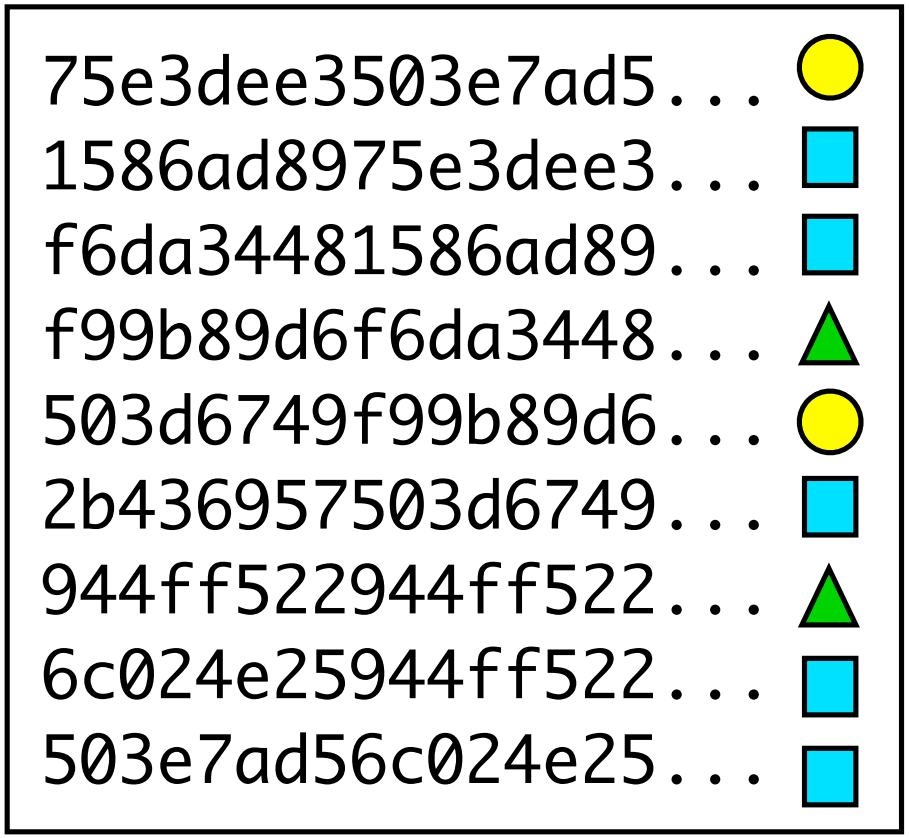
Idea:

Use the SHA1 Hash for the File Name

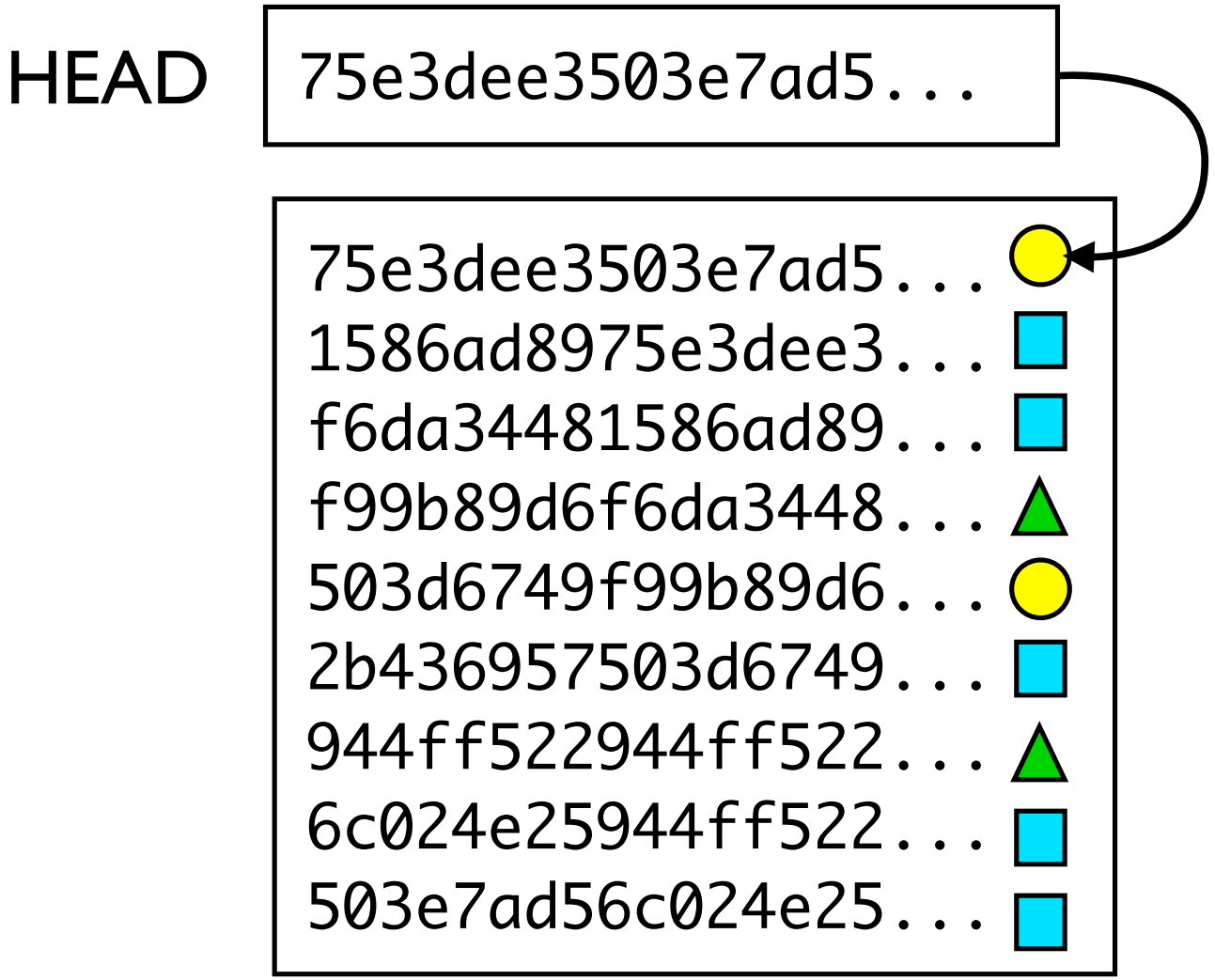
Project Archive

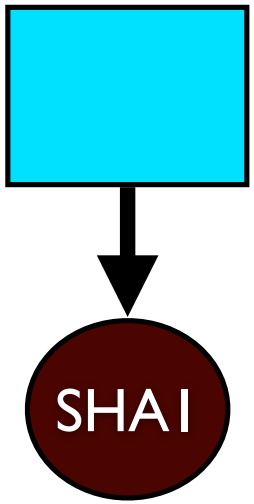


Project Archive



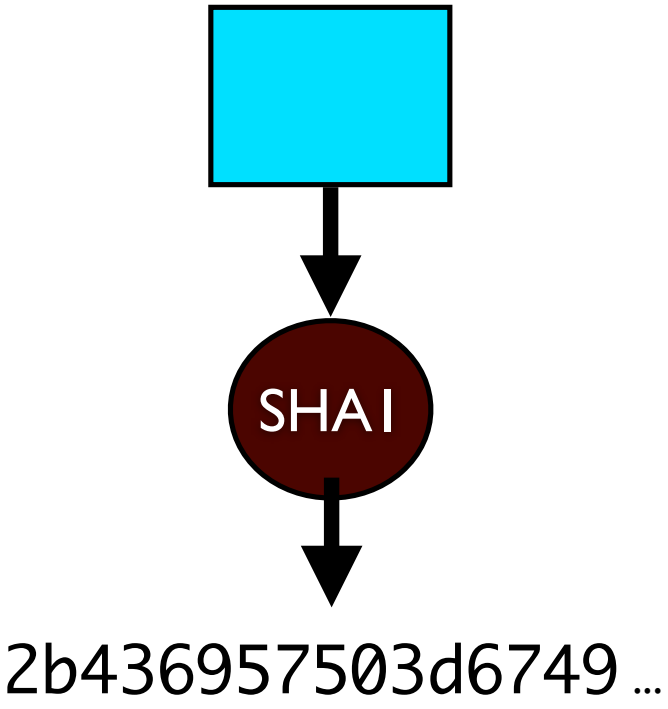
Project Archive





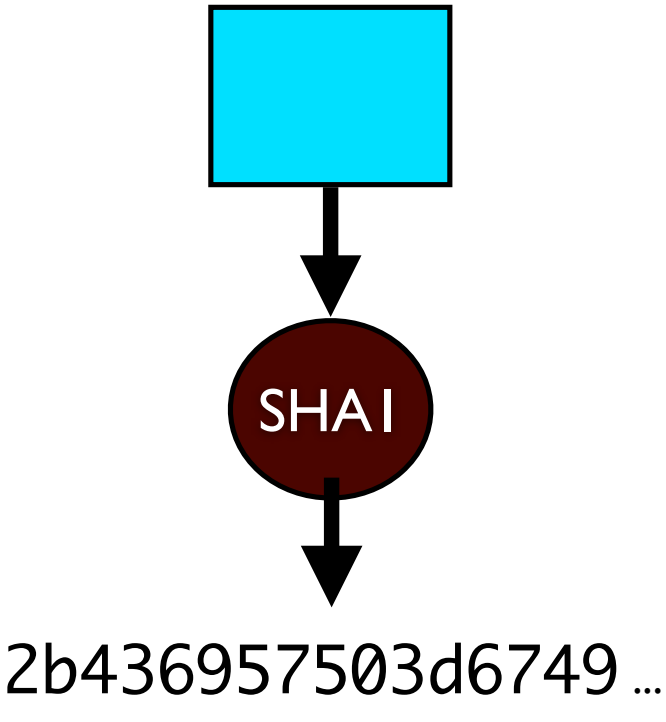
Project Archive

```
75e3dee3503e7ad5...
1586ad8975e3dee3...
f6da34481586ad89...
f99b89d6f6da3448...
503d6749f99b89d6...
2b436957503d6749...
944ff522944ff522...
6c024e25944ff522...
503e7ad56c024e25...
```



Project Archive

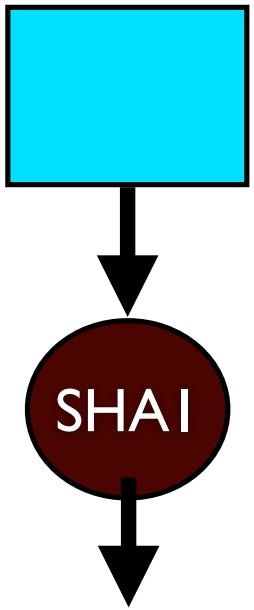
```
75e3dee3503e7ad5...
1586ad8975e3dee3...
f6da34481586ad89...
f99b89d6f6da3448...
503d6749f99b89d6...
2b436957503d6749...
944ff522944ff522...
6c024e25944ff522...
503e7ad56c024e25...
```



Match!

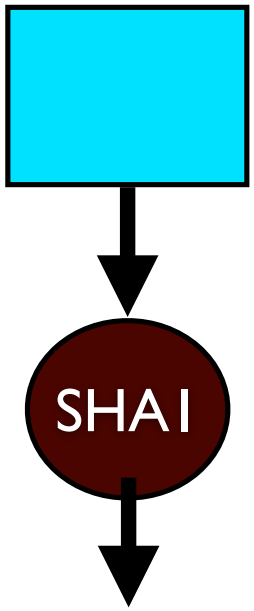
Project Archive

75e3dee3503e7ad5...
1586ad8975e3dee3...
f6da34481586ad89...
f99b89d6f6da3448...
503d6749f99b89d6...
2b436957503d6749...
944ff522944ff522...
6c024e25944ff522...
503e7ad56c024e25...



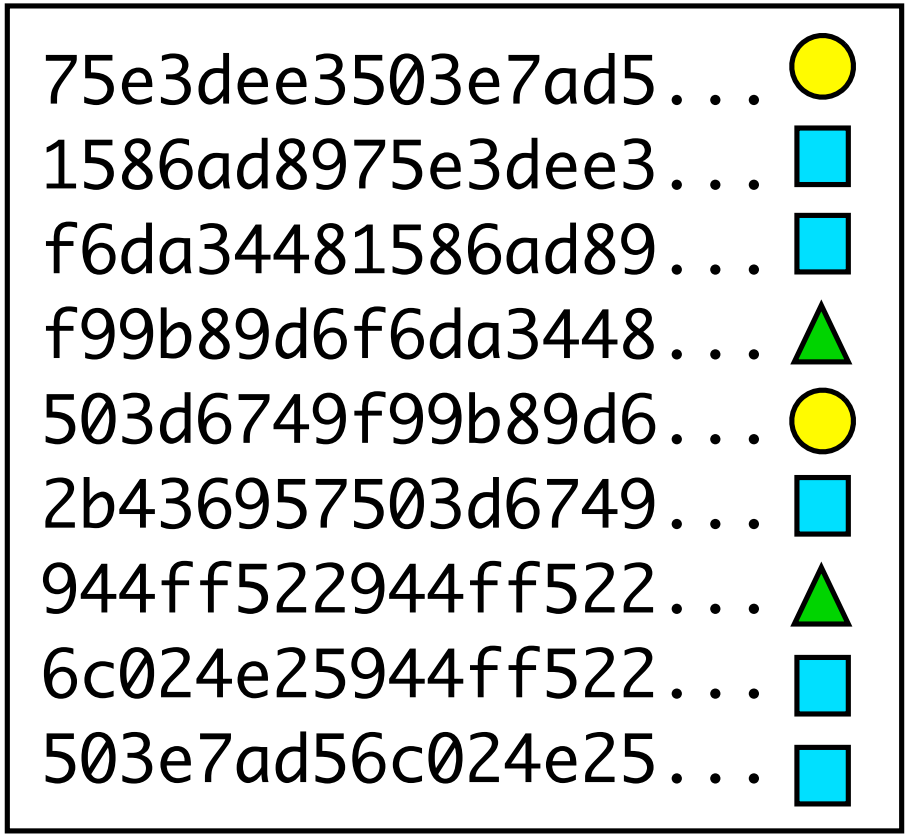
Project Archive

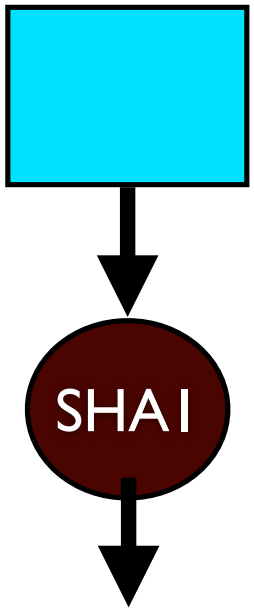
```
75e3dee3503e7ad5...
1586ad8975e3dee3...
f6da34481586ad89...
f99b89d6f6da3448...
503d6749f99b89d6...
2b436957503d6749...
944ff522944ff522...
6c024e25944ff522...
503e7ad56c024e25...
```



9d6f503d24e2d56c ...

Project Archive



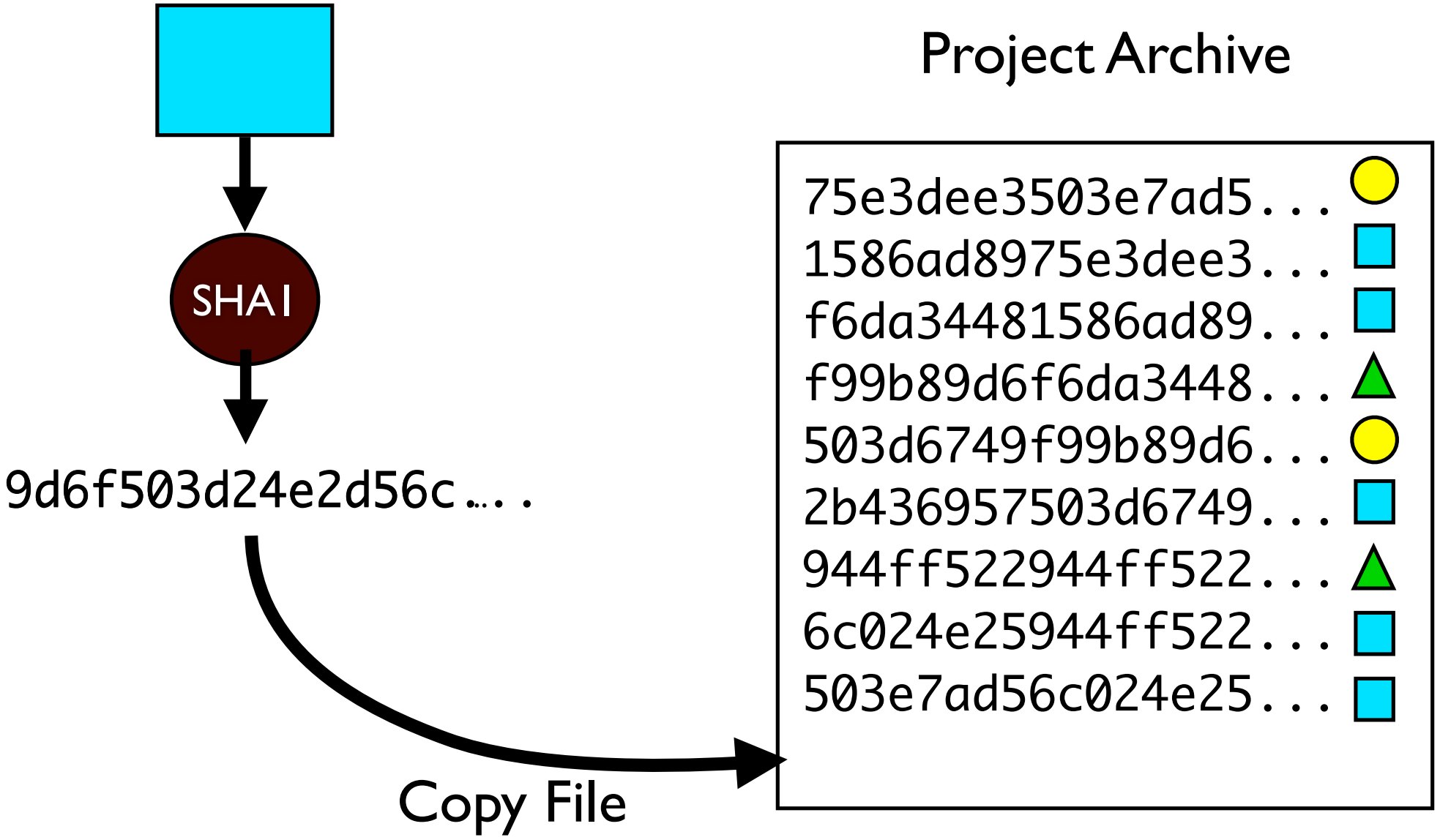


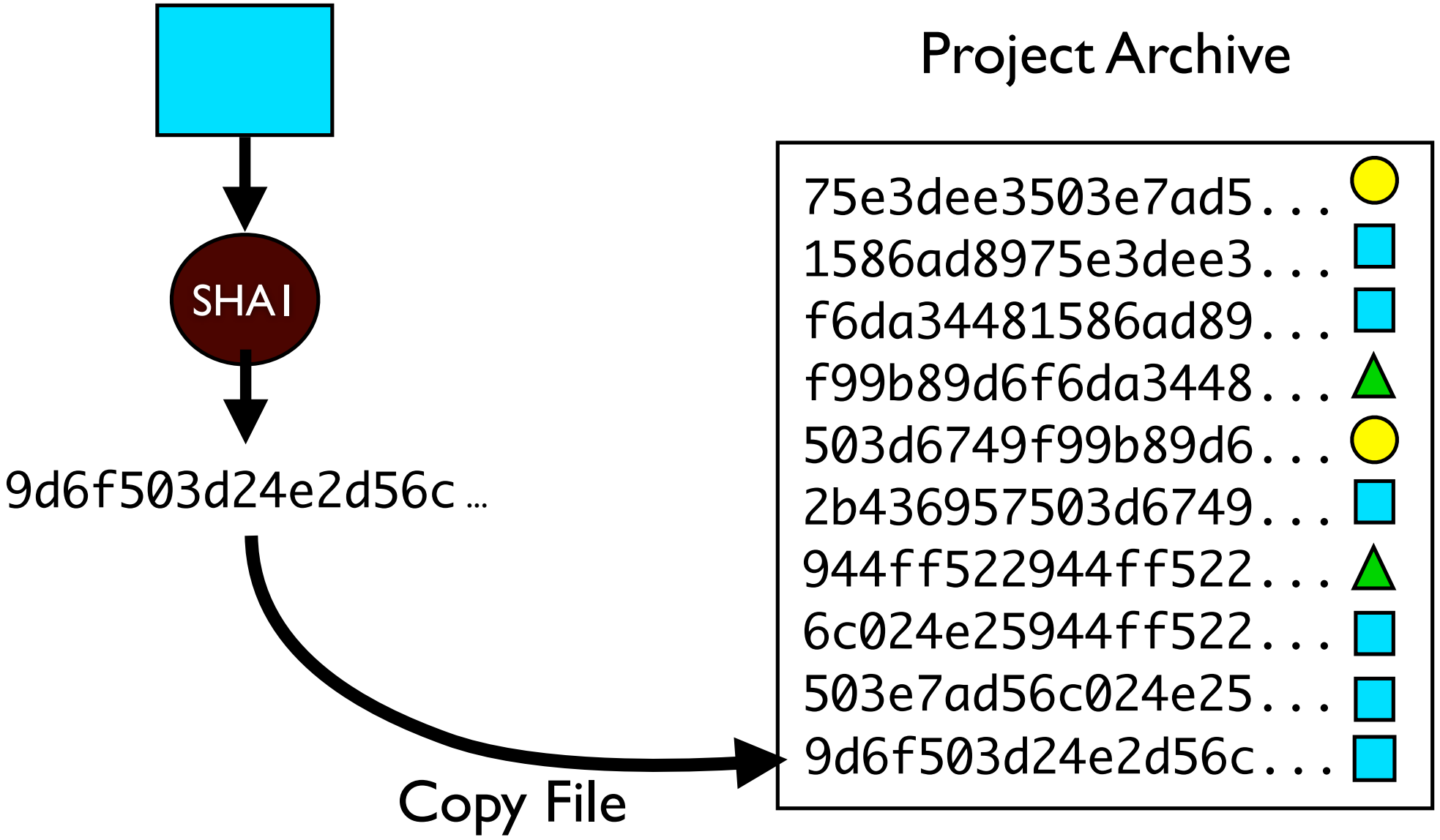
9d6f503d24e2d56c ...

No Match!

Project Archive

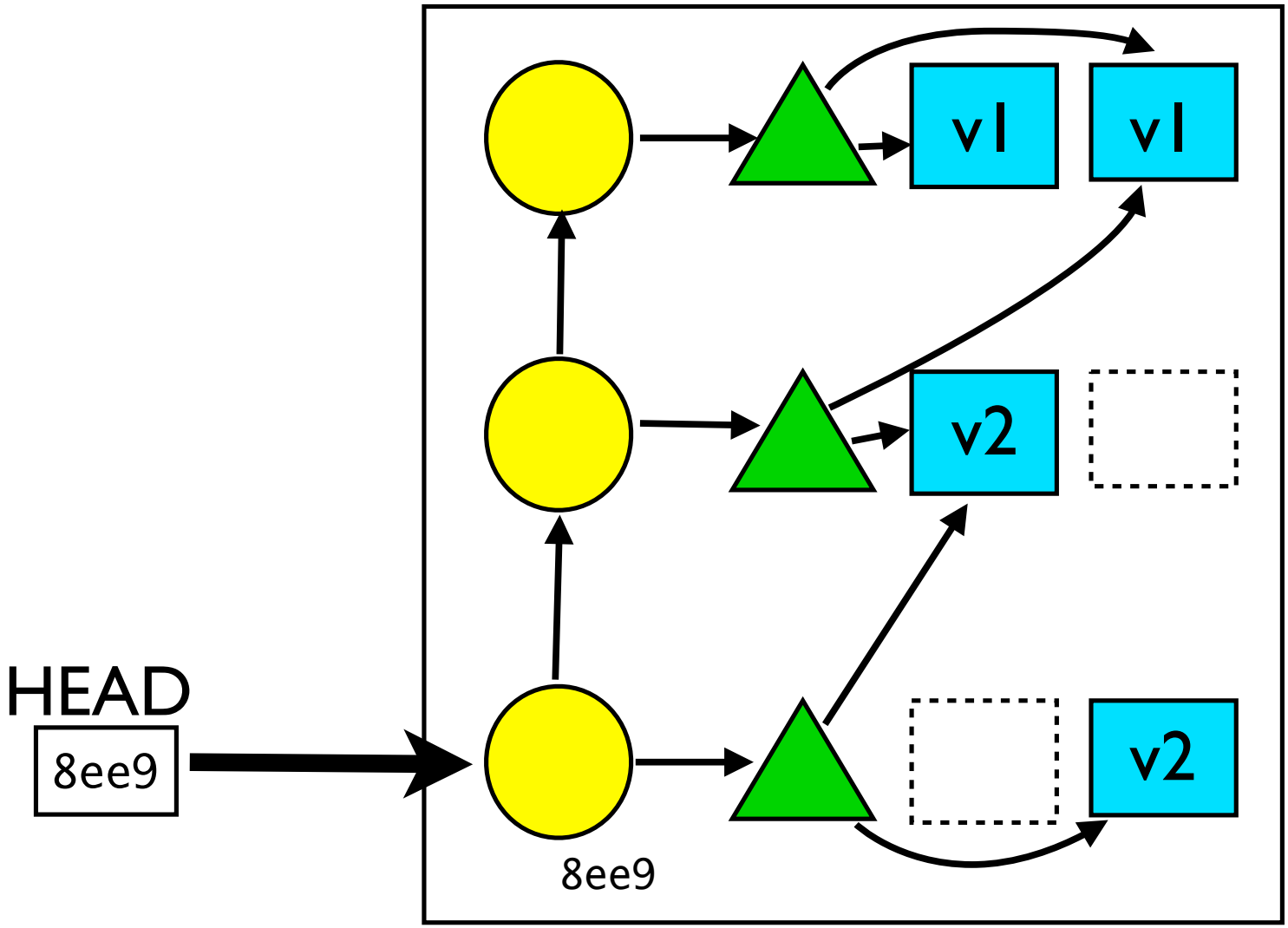
75e3dee3503e7ad5...
1586ad8975e3dee3...
f6da34481586ad89...
f99b89d6f6da3448...
503d6749f99b89d6...
2b436957503d6749...
944ff522944ff522...
6c024e25944ff522...
503e7ad56c024e25...


SHA1
9d6f503d24e2d56c….
Copy File
Project Archive
75e3dee3503e7ad5...
1586ad8975e3dee3...
f6da34481586ad89...
f99b89d6f6da3448...
503d6749f99b89d6...
2b436957503d6749...
944ff522944ff522...
6c024e25944ff522...
503e7ad56c024e25...

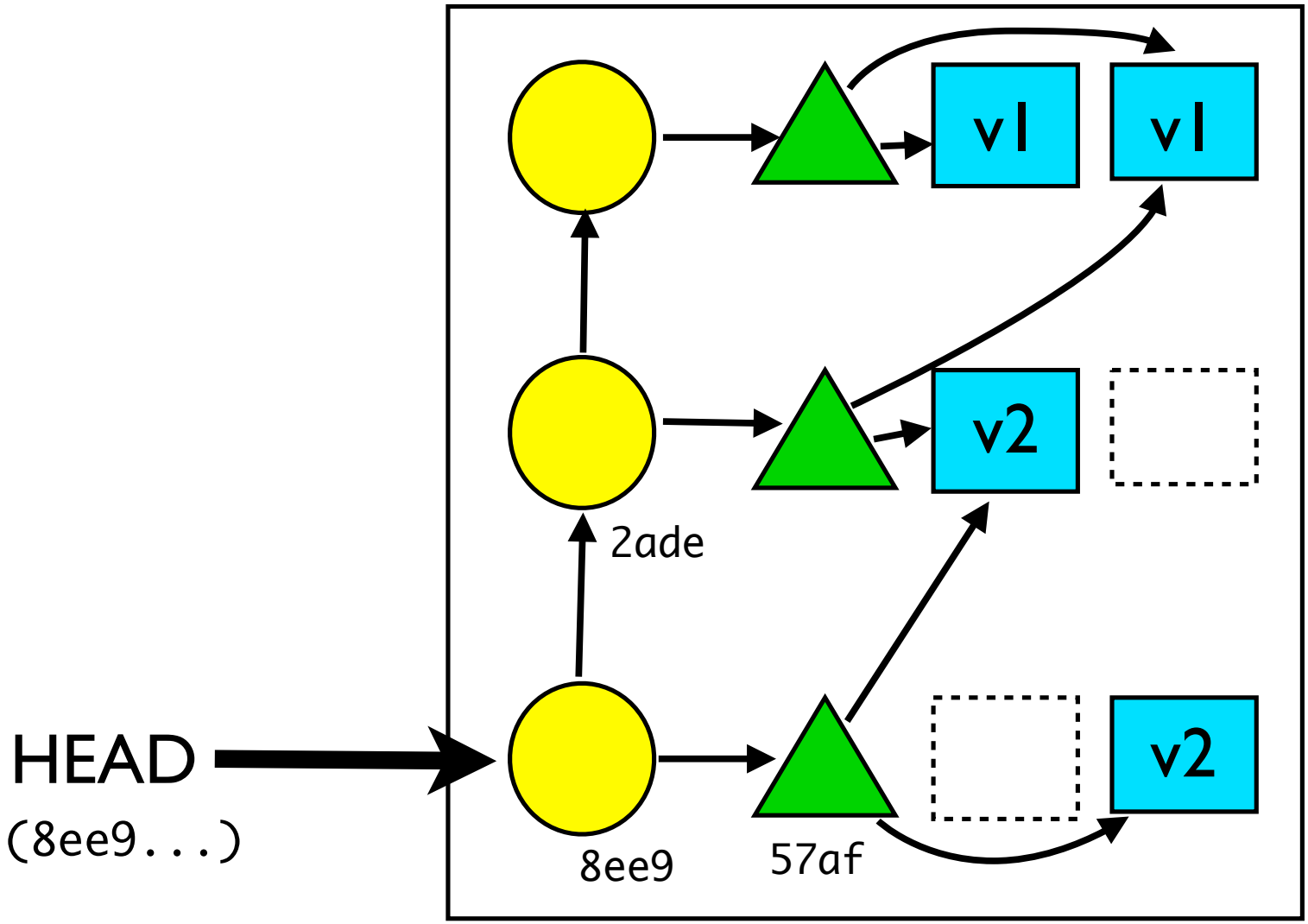

SHA1
9d6f503d24e2d56c ...
Copy File
Project Archive
75e3dee3503e7ad5...
1586ad8975e3dee3...
f6da34481586ad89...
f99b89d6f6da3448...
503d6749f99b89d6...
2b436957503d6749...
944ff522944ff522...
6c024e25944ff522...
503e7ad56c024e25...
9d6f503d24e2d56c...

Project Archive Integrity

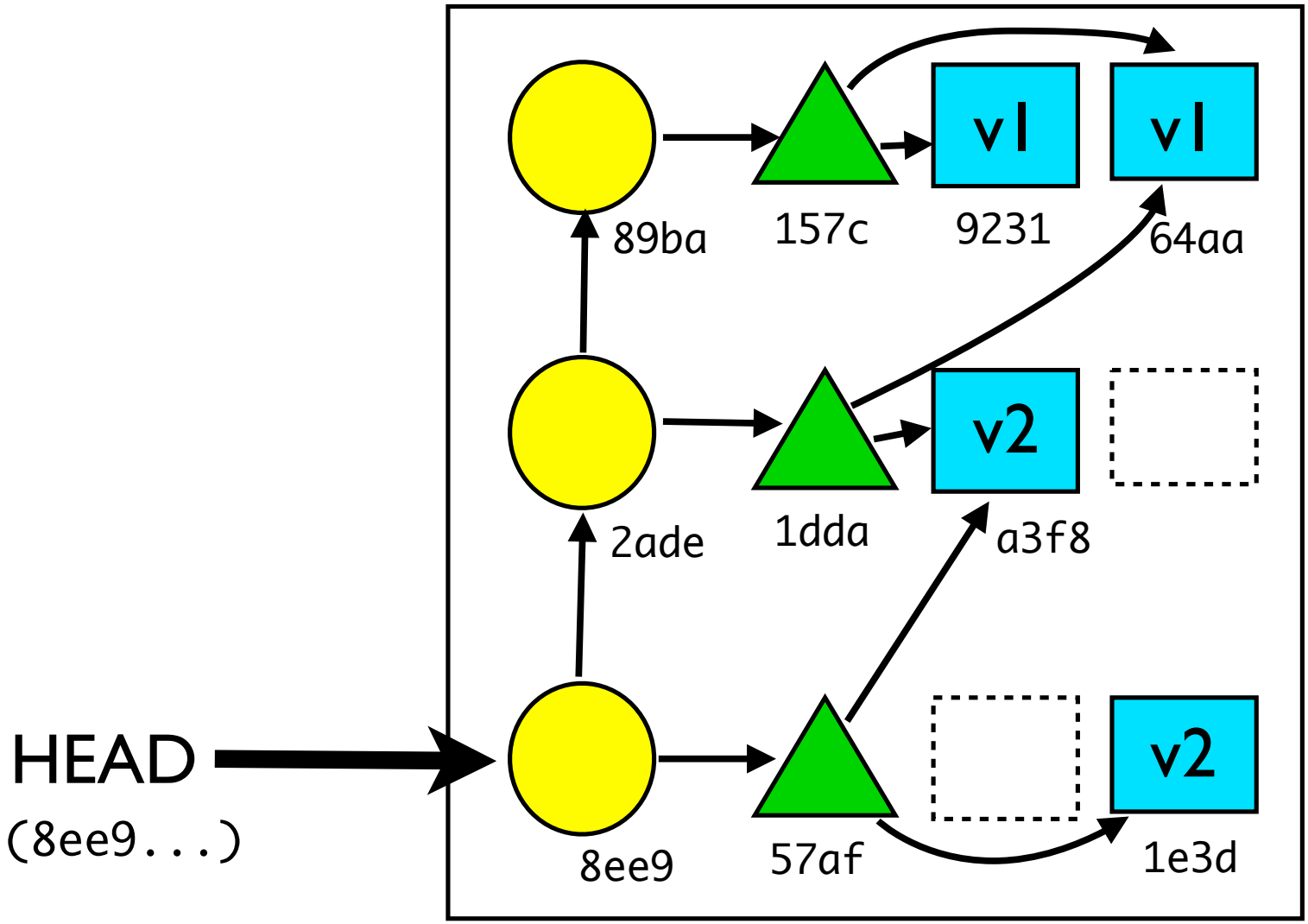
Project Archive

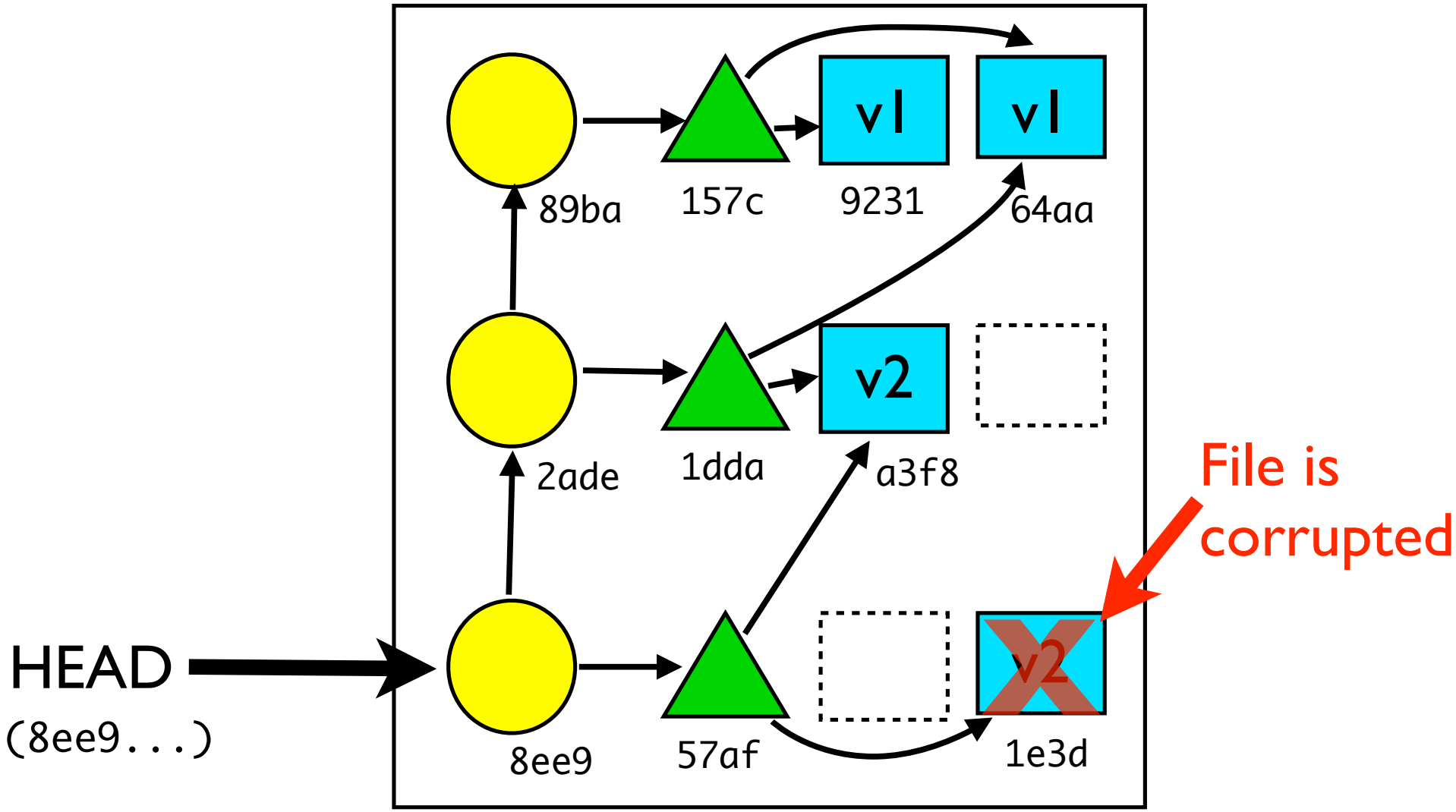


Project Archive

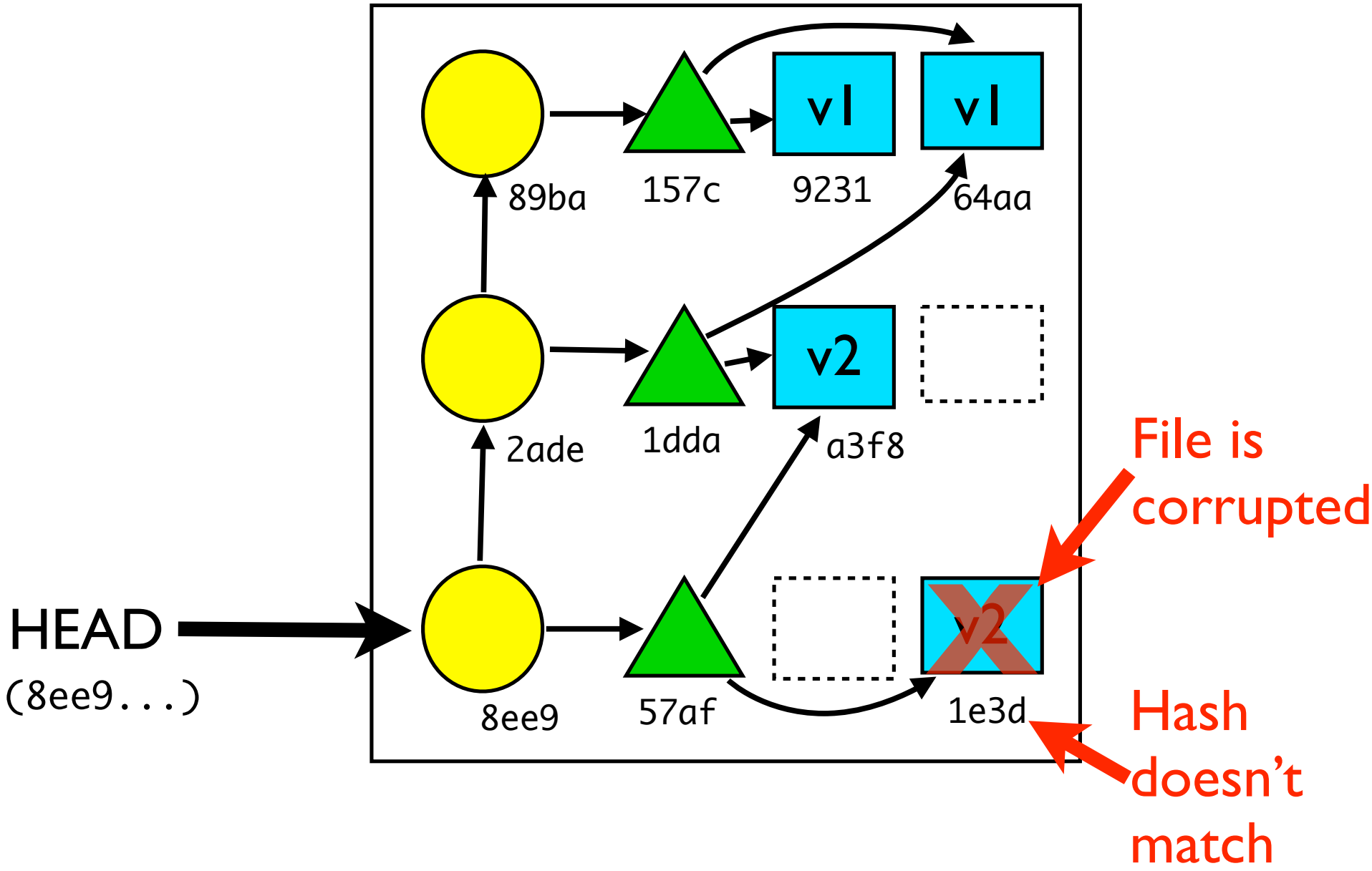


Project Archive

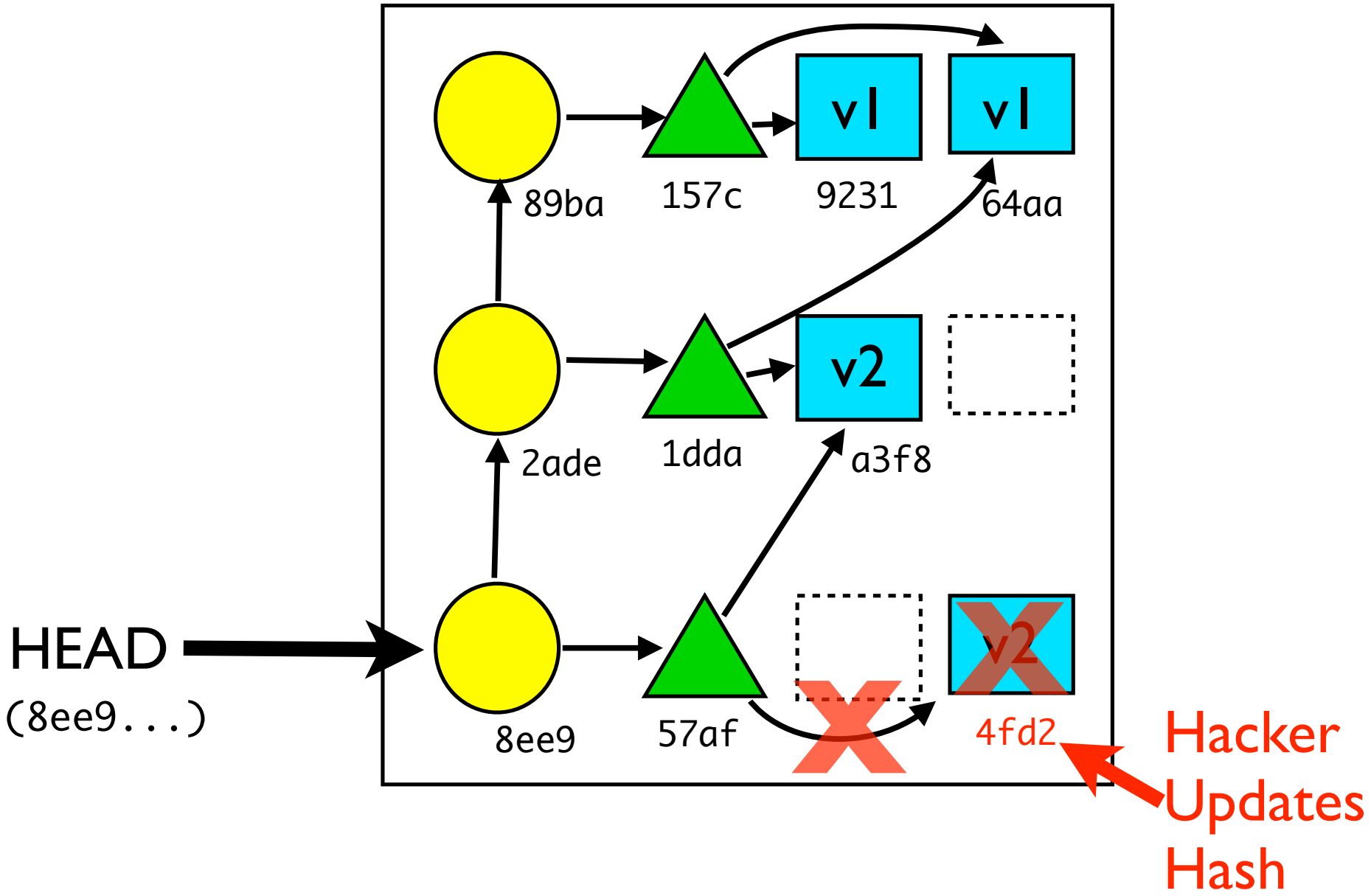



v1
v1
89ba
157c
9231
64aa
v2
2ade
1dda
a3f8
File is
corrupted
HEAD
(8ee9...)
v2
8ee9
57af
1e3d

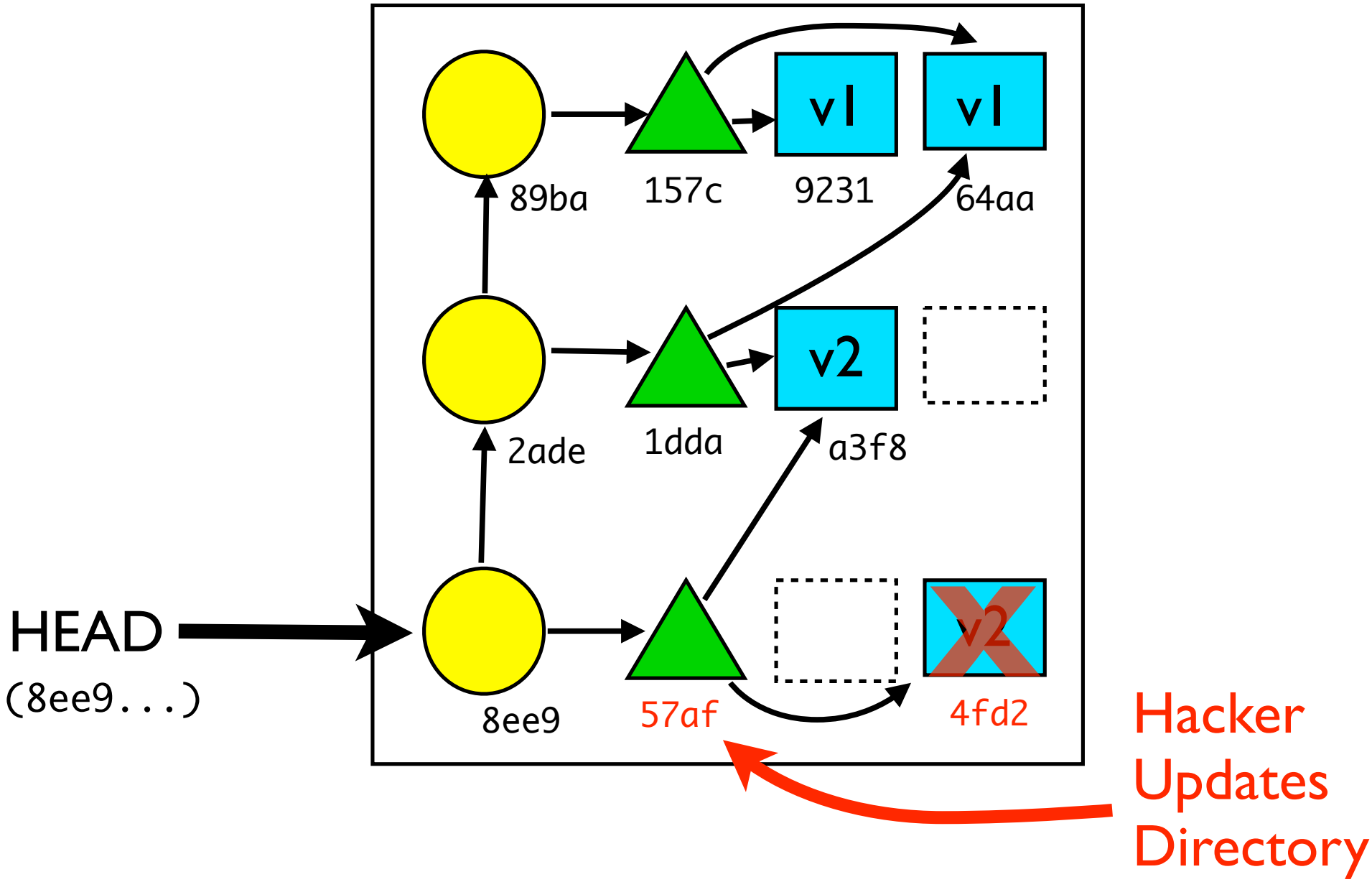
Project Archive



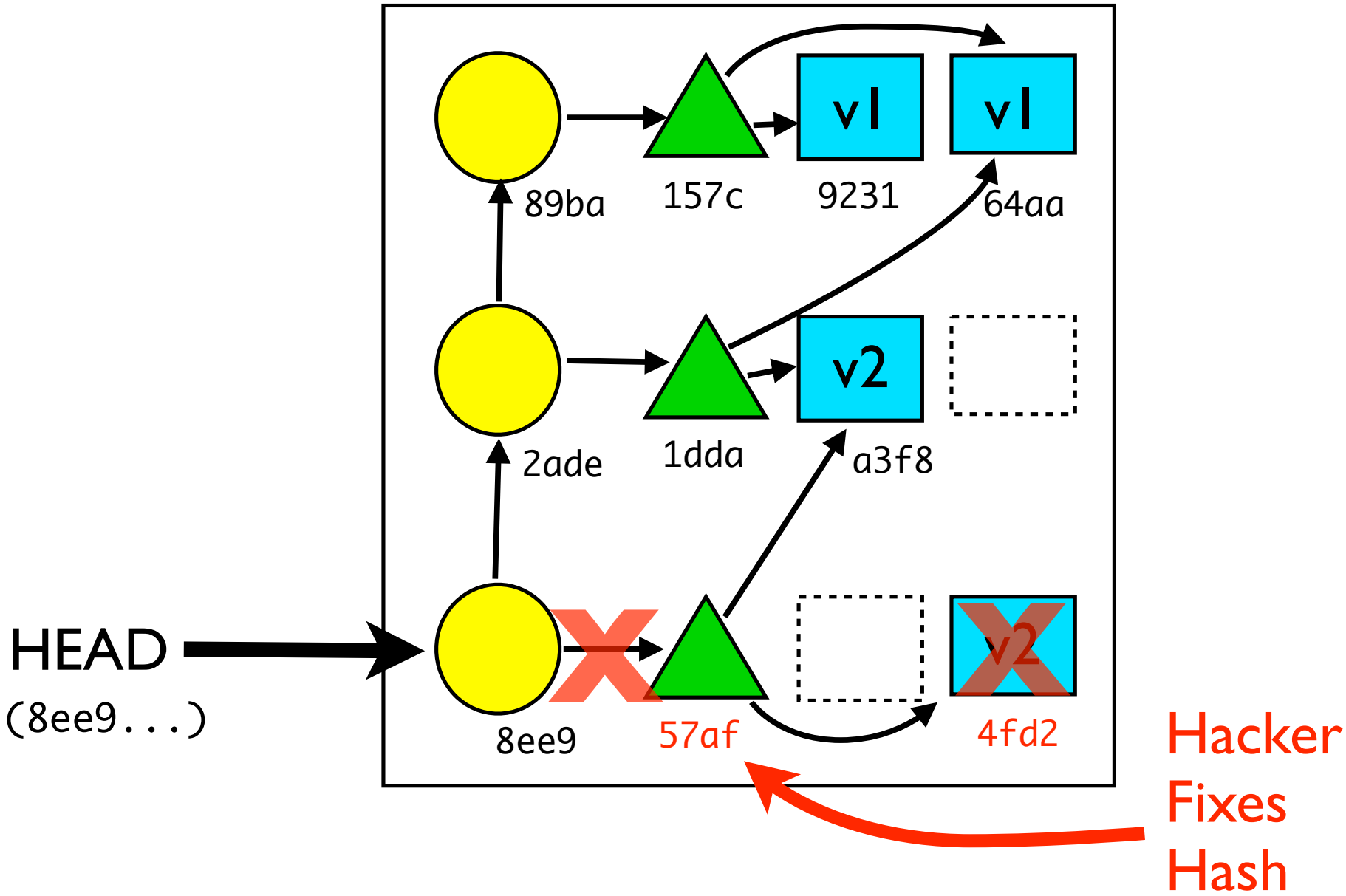
Project Archive



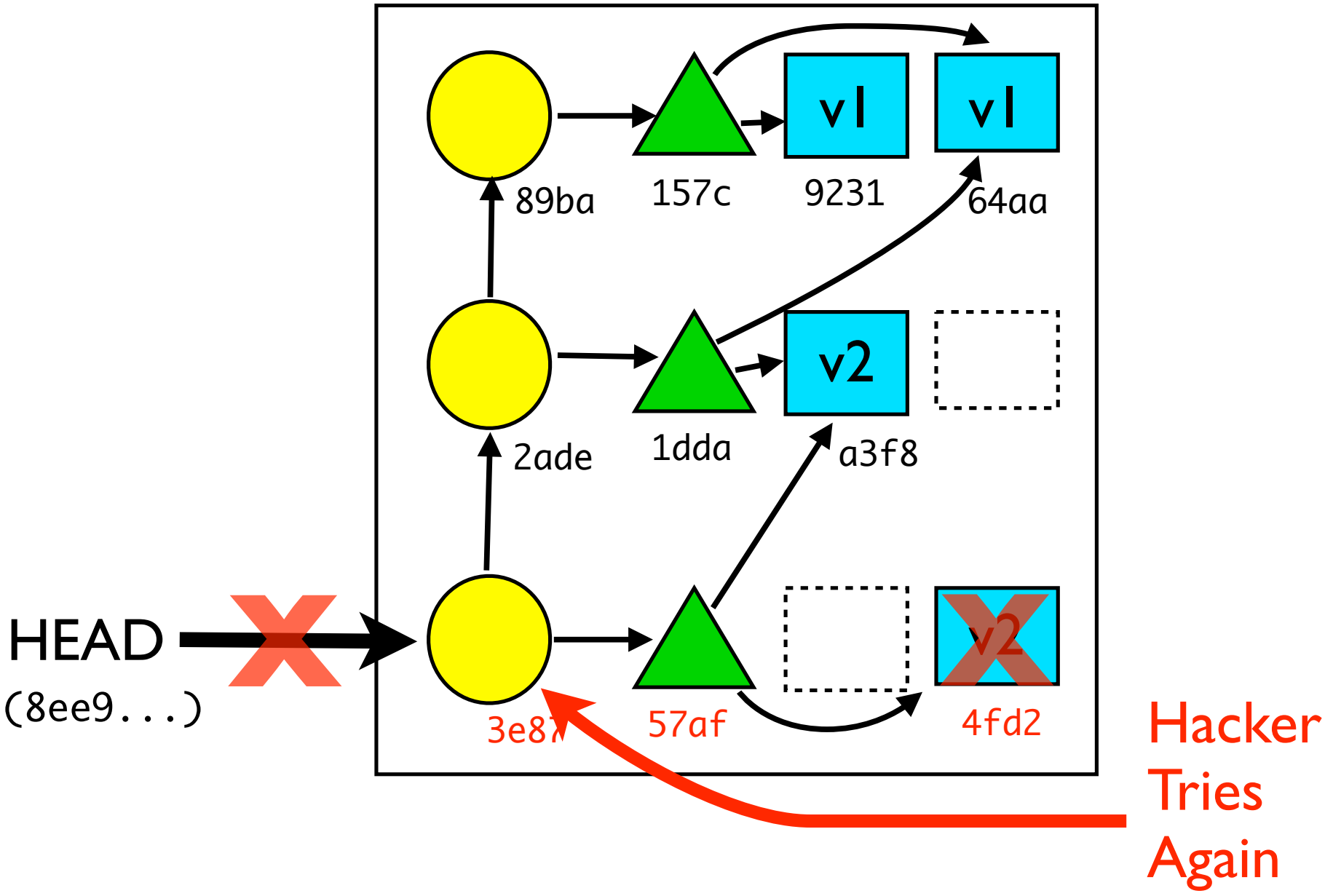
Project Archive



Project Archive



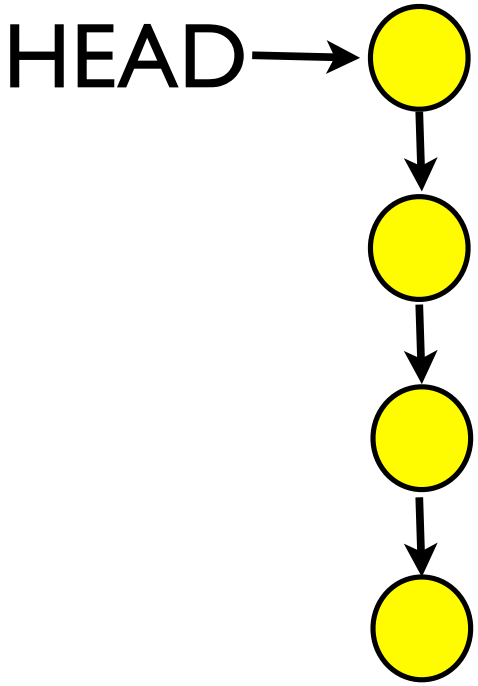
Project Archive



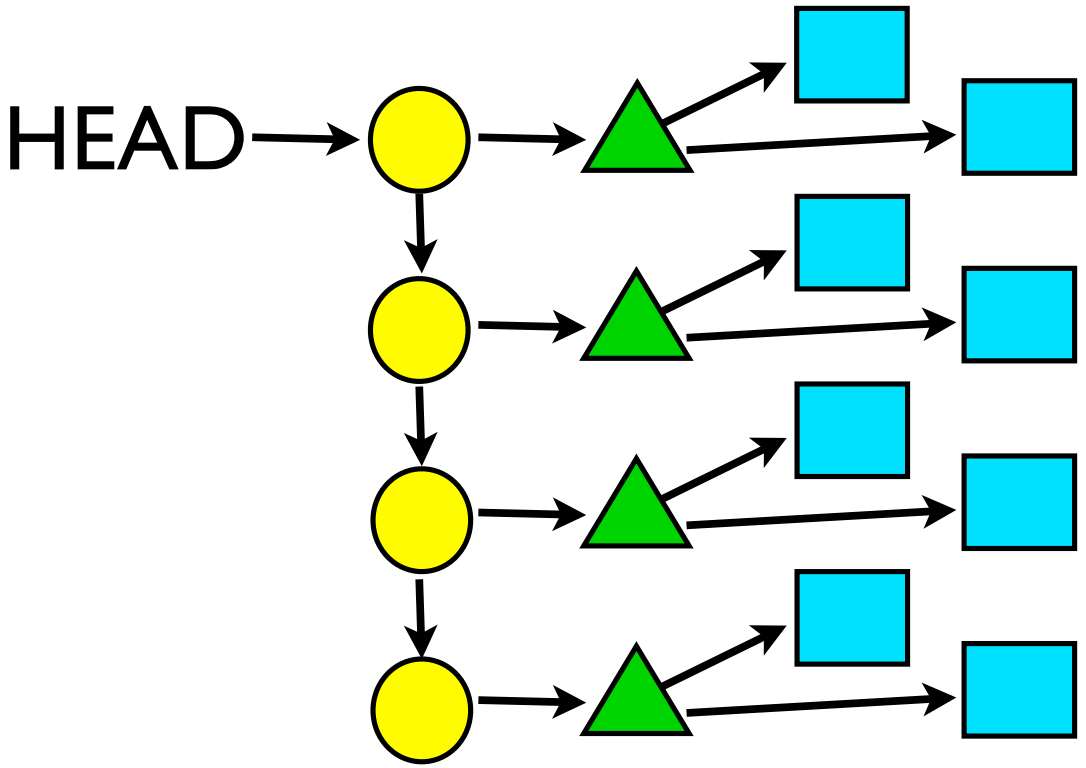
Tracking Snapshots

Need to keep track of versions by name.

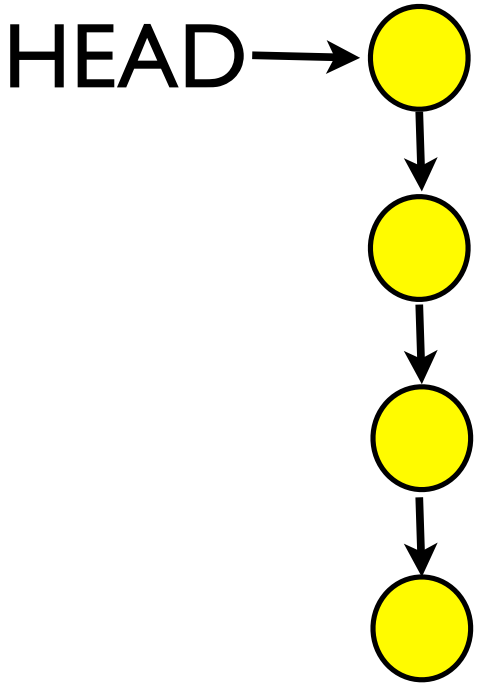
Project Archive
(snapshots only)



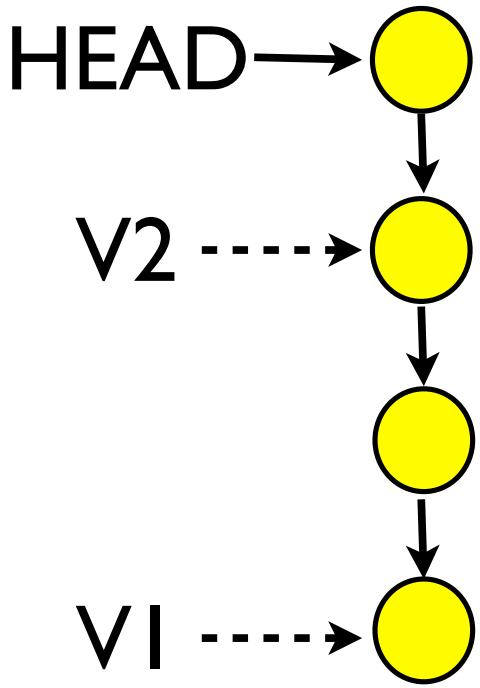
Project Archive



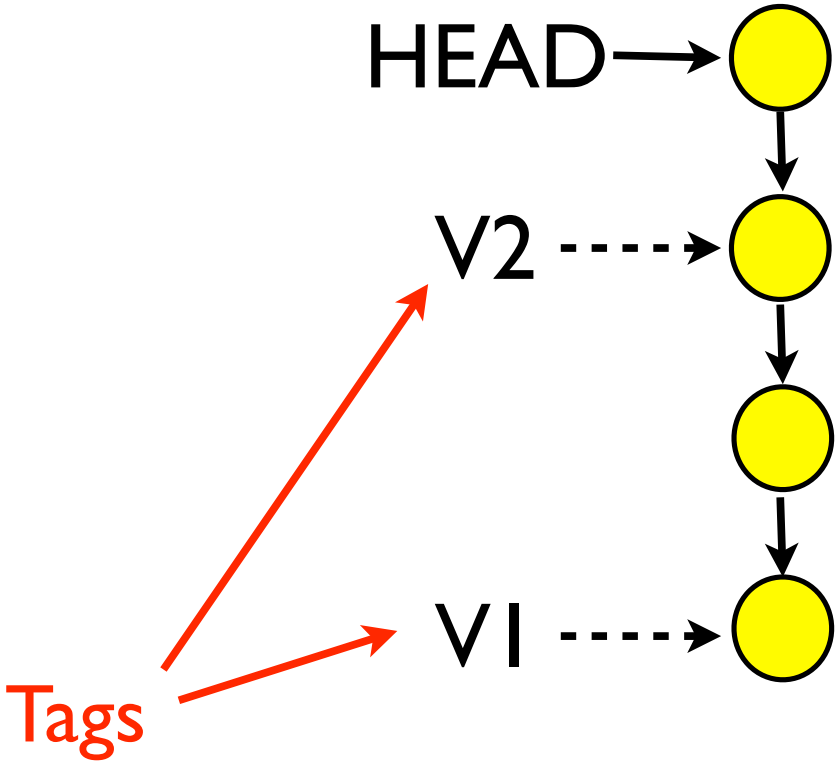
Project Archive
(snapshots only)



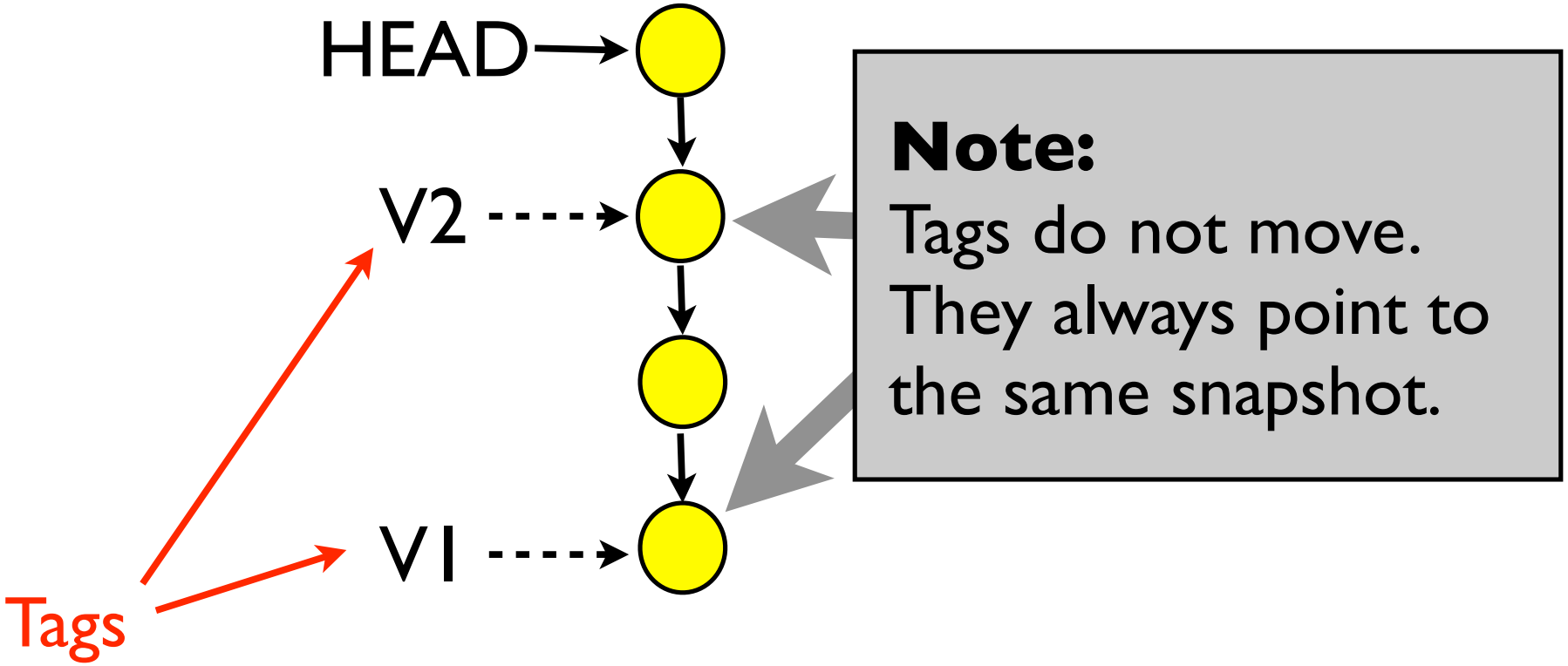
Project Archive
(snapshots only)



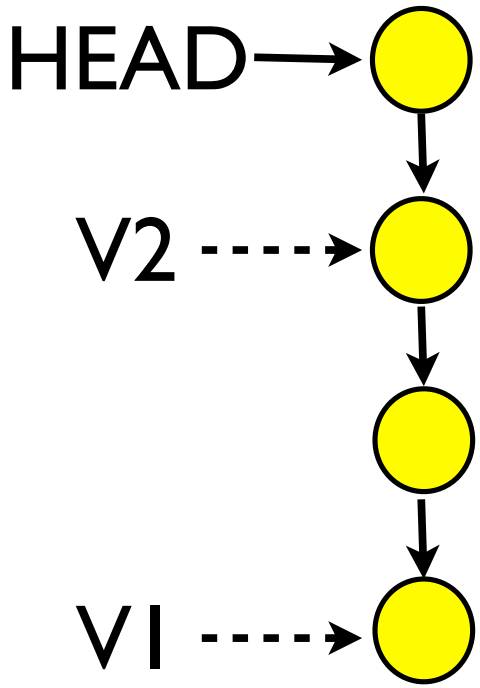
Project Archive
(snapshots only)



Project Archive
(snapshots only)

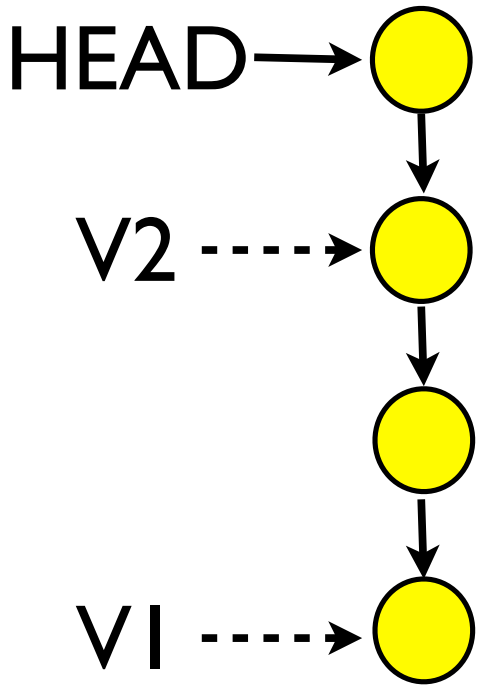


Project Archive
(snapshots only)

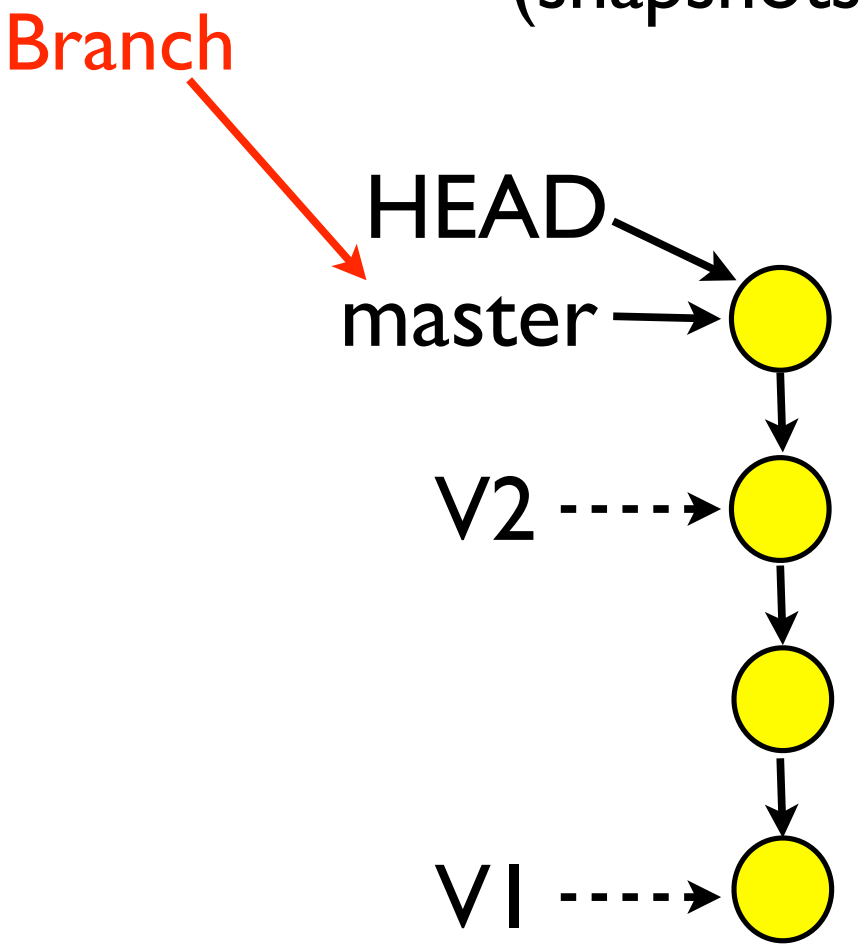


Need to handle changes to older versions while allowing development on the latest version.

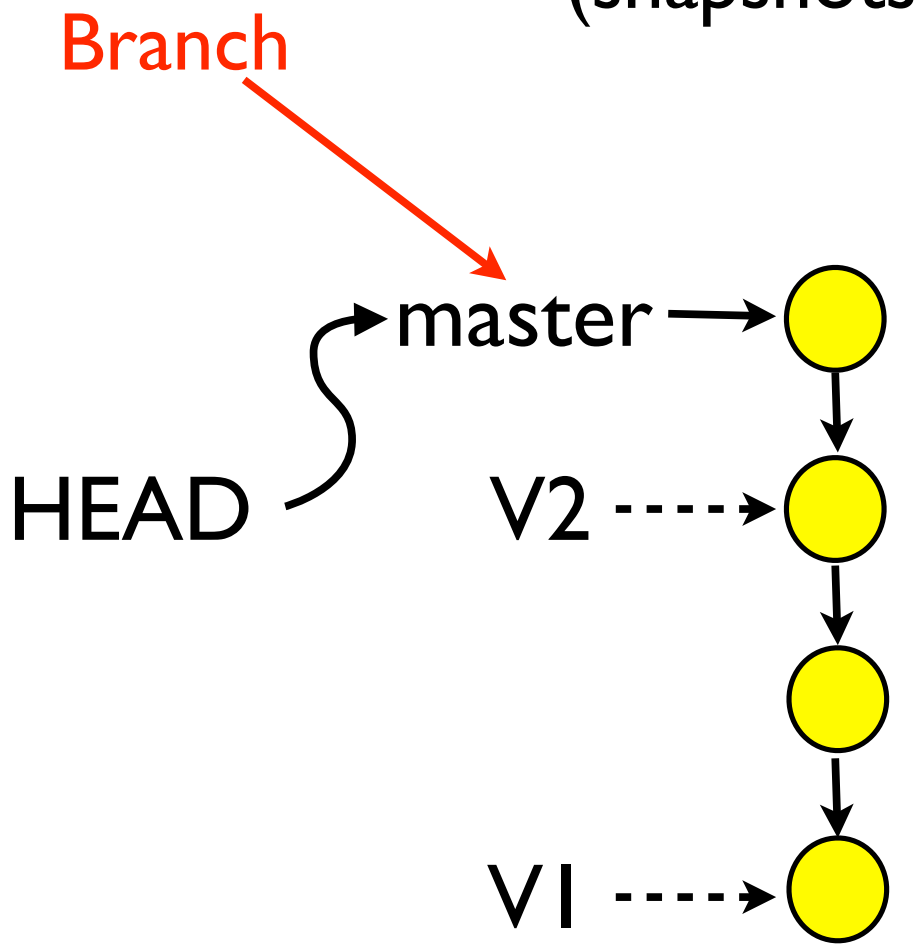
Project Archive
(snapshots only)



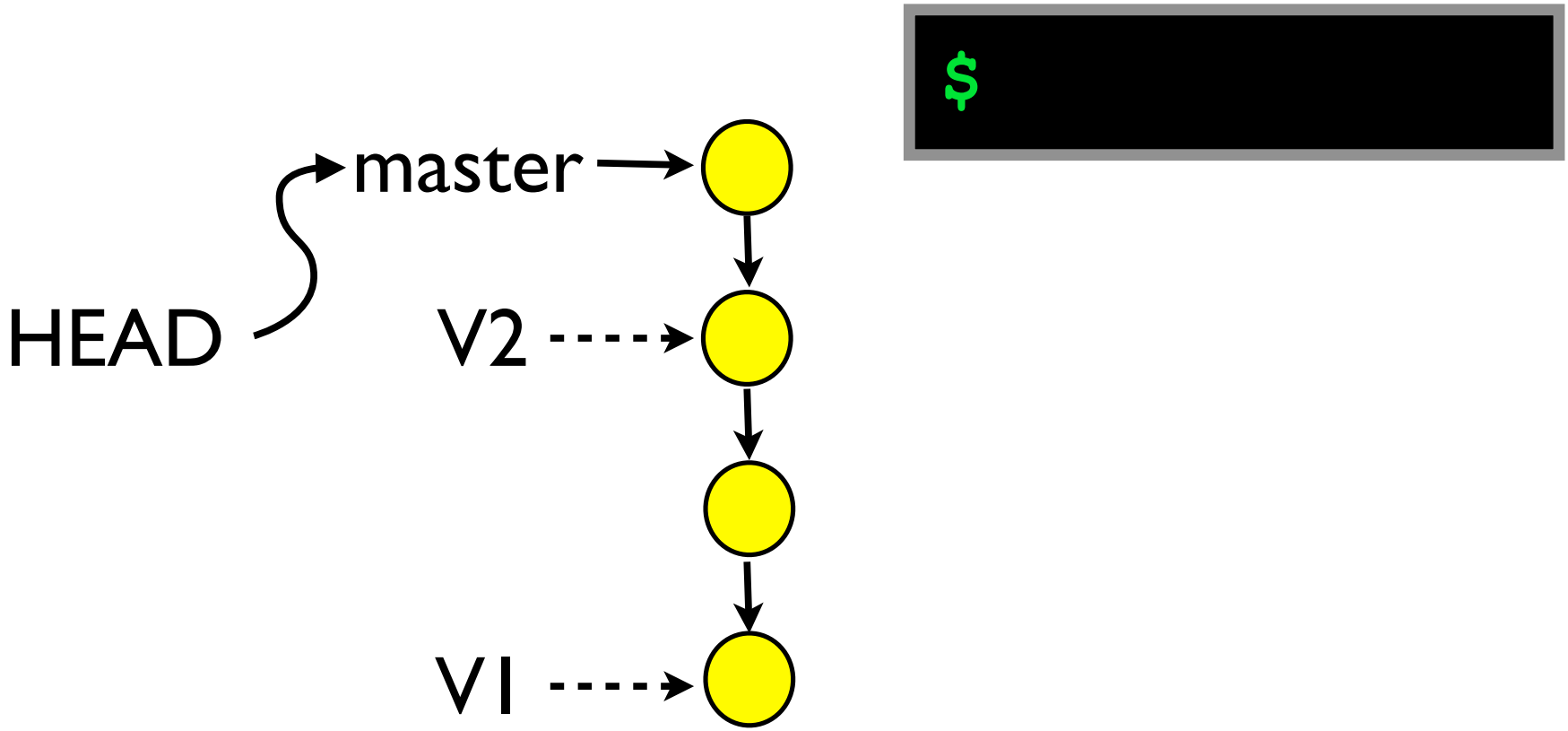
Project Archive (snapshots only)



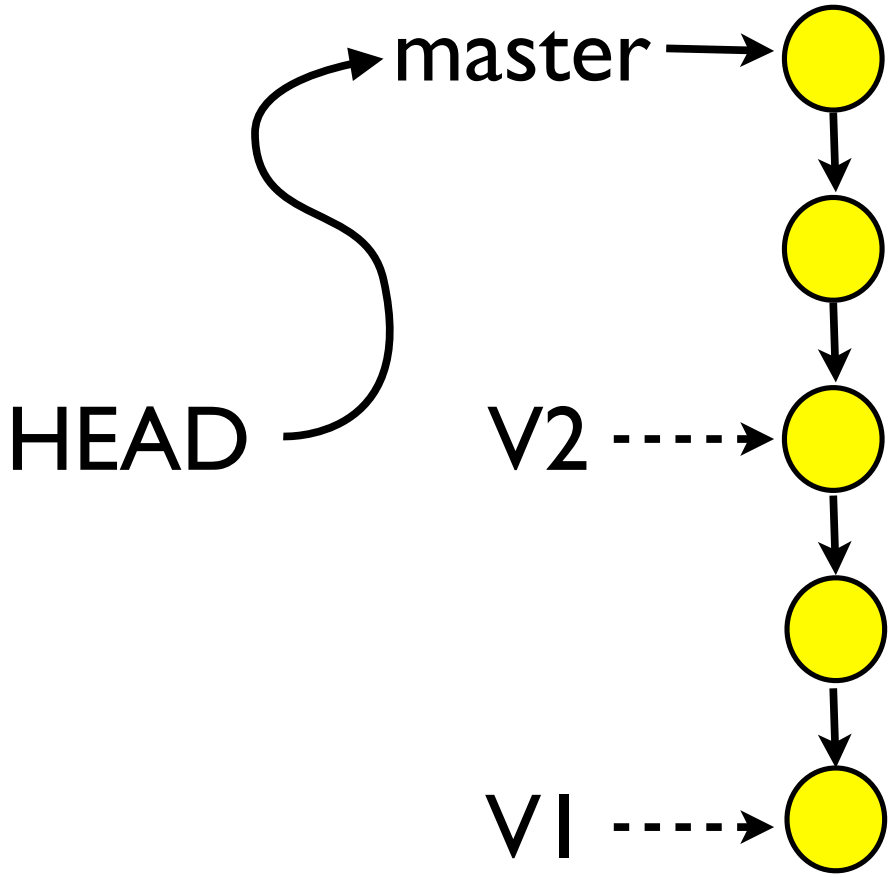
Project Archive (snapshots only)



Project Archive
(snapshots only)

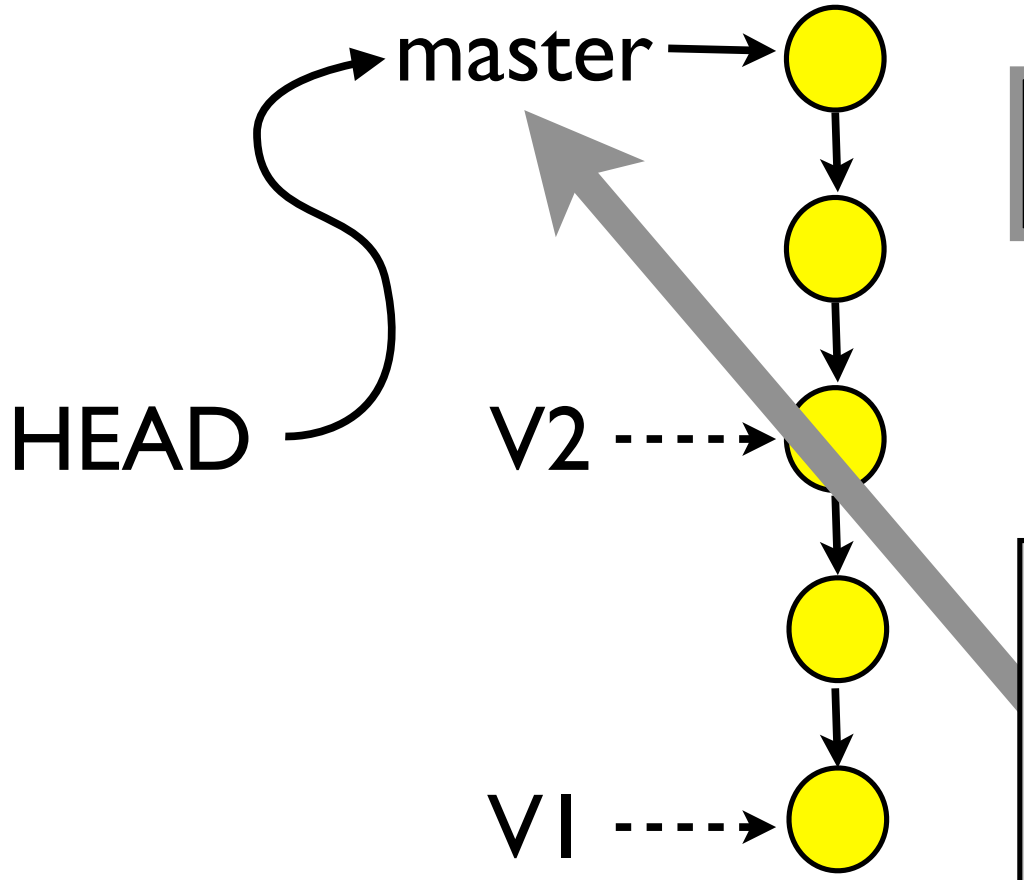


Project Archive (snapshots only)




```
$ csc snap
```

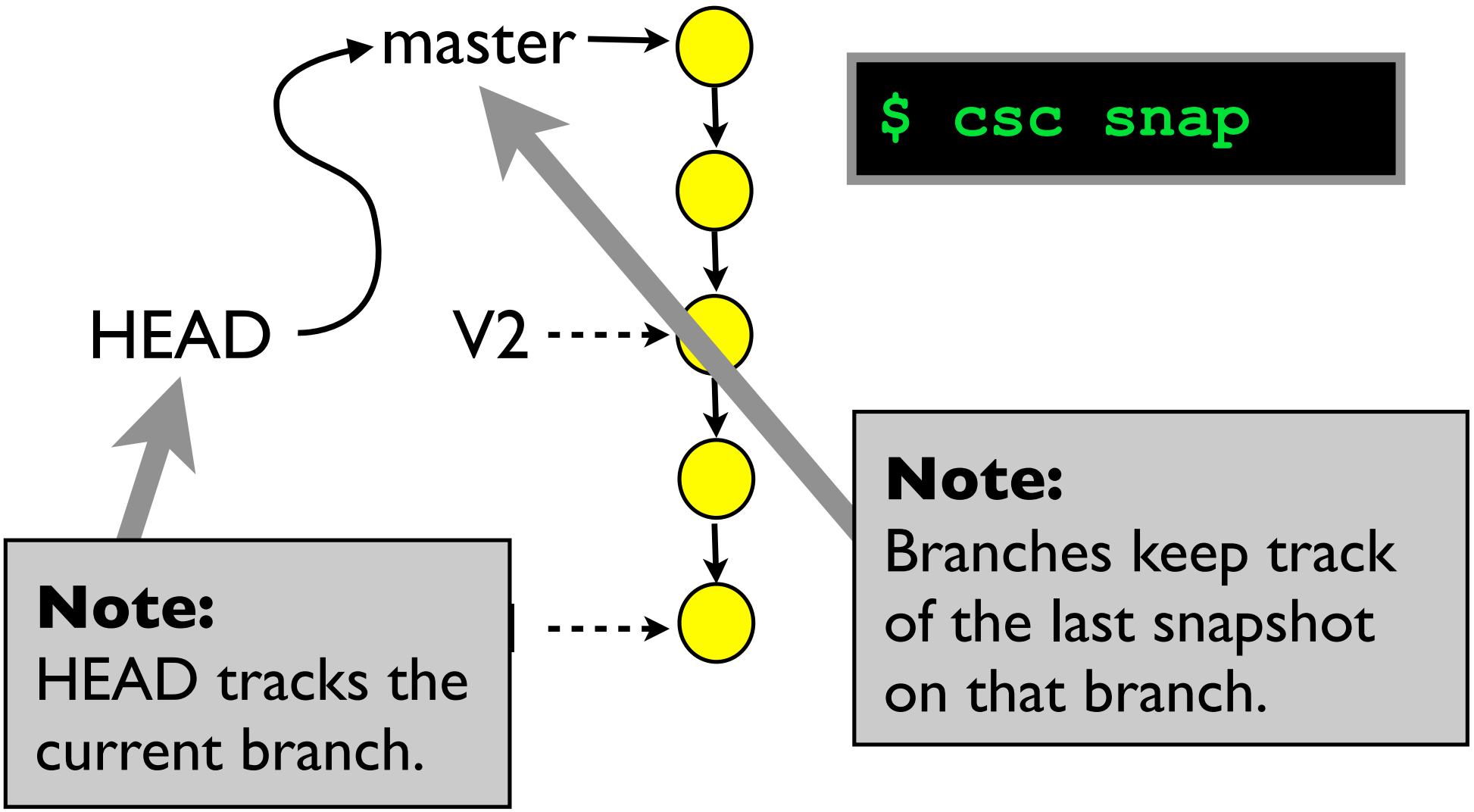
Project Archive
(snapshots only)




```
$ csc snap
```

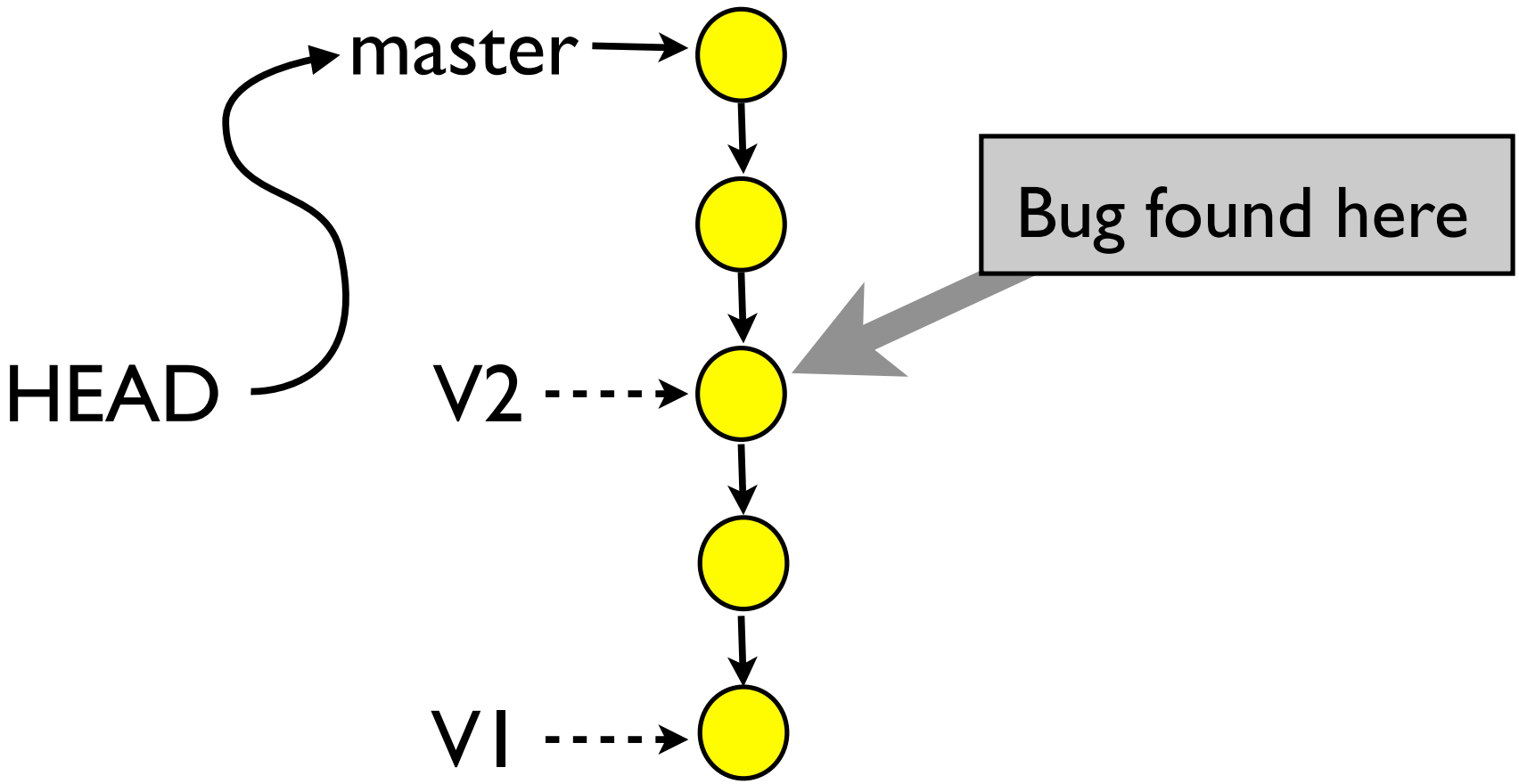
Note:
Branches keep track of the last snapshot on that branch.

Project Archive (snapshots only)

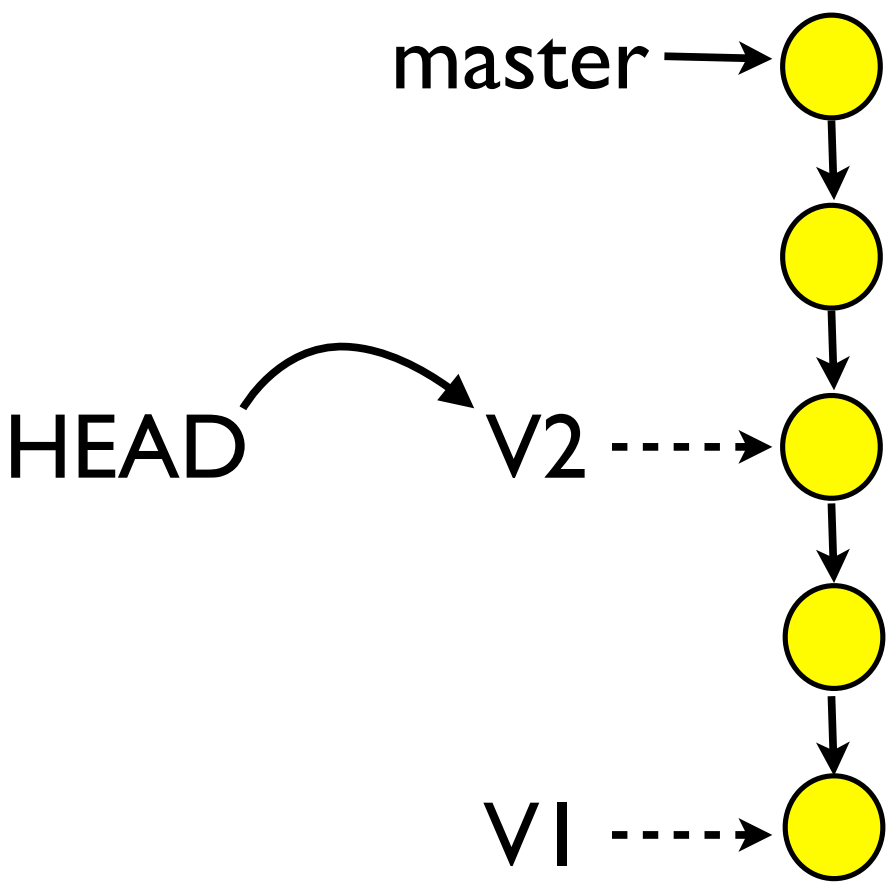


Oops! We found a bug

Project Archive

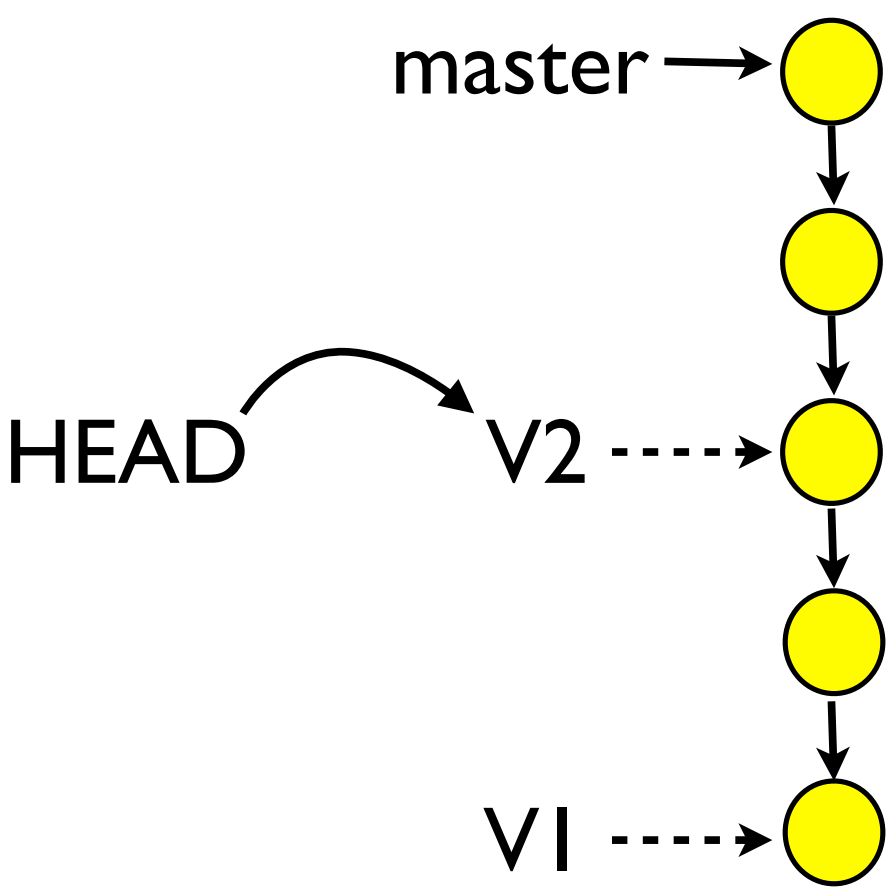


Project Archive



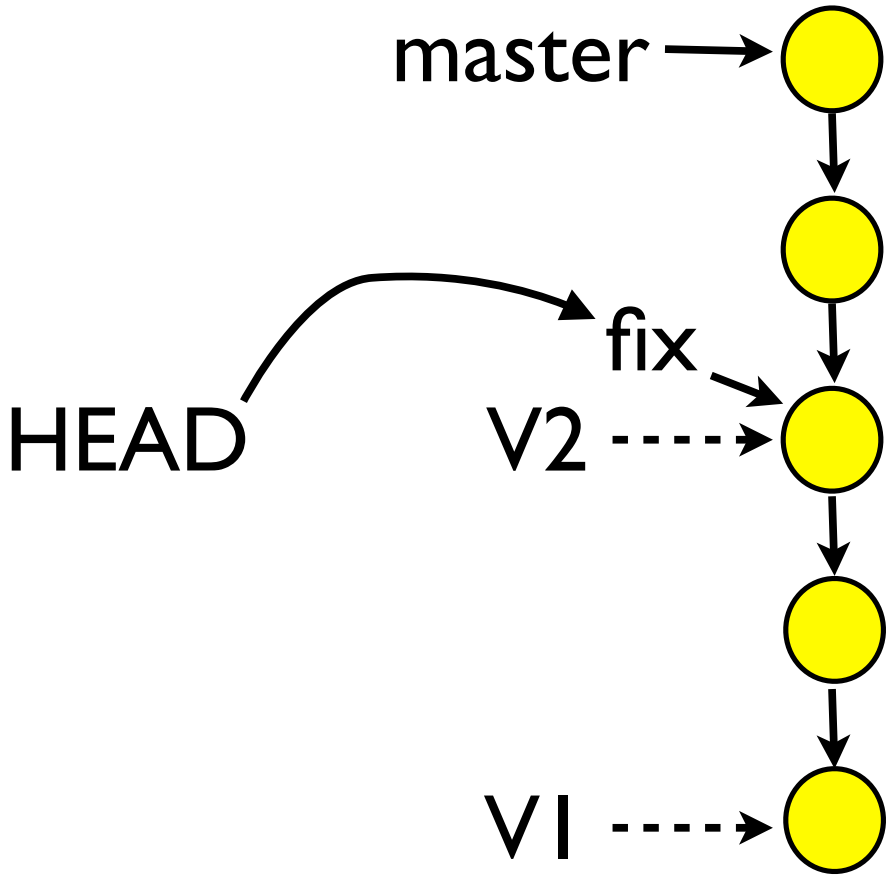

```
$ csc checkout V2
```

Project Archive



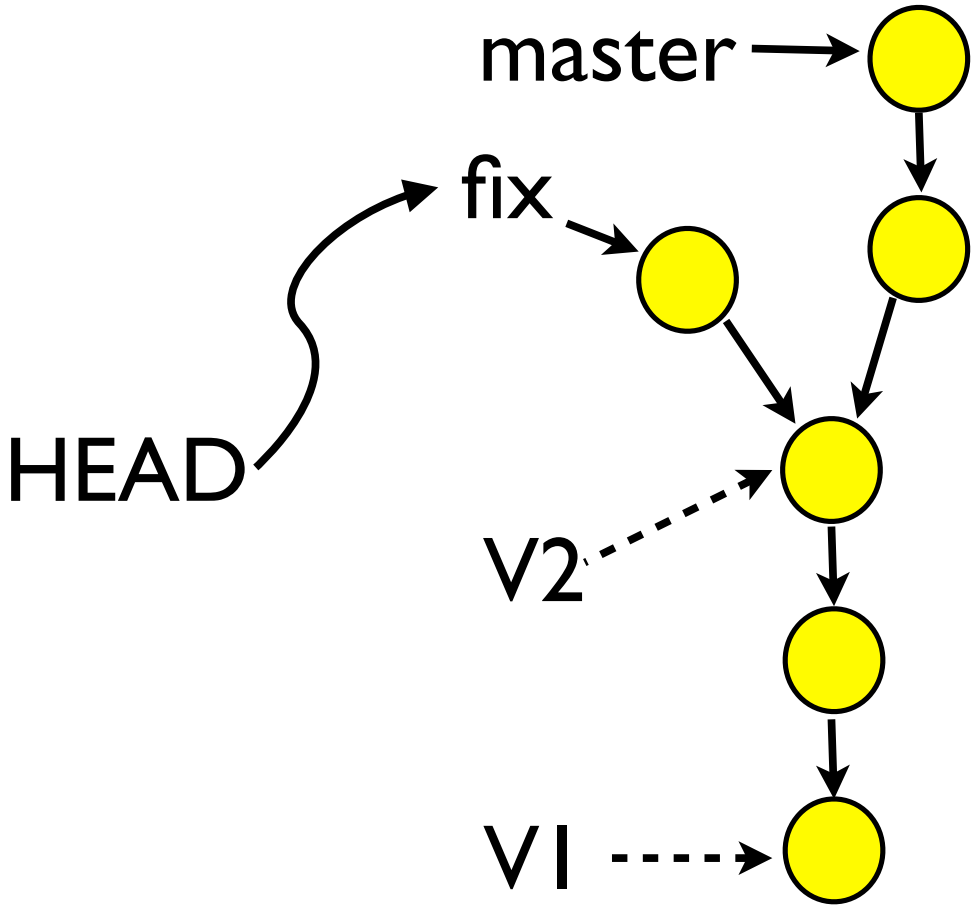

```
$ csc checkout V2
$ edit files
```

Project Archive



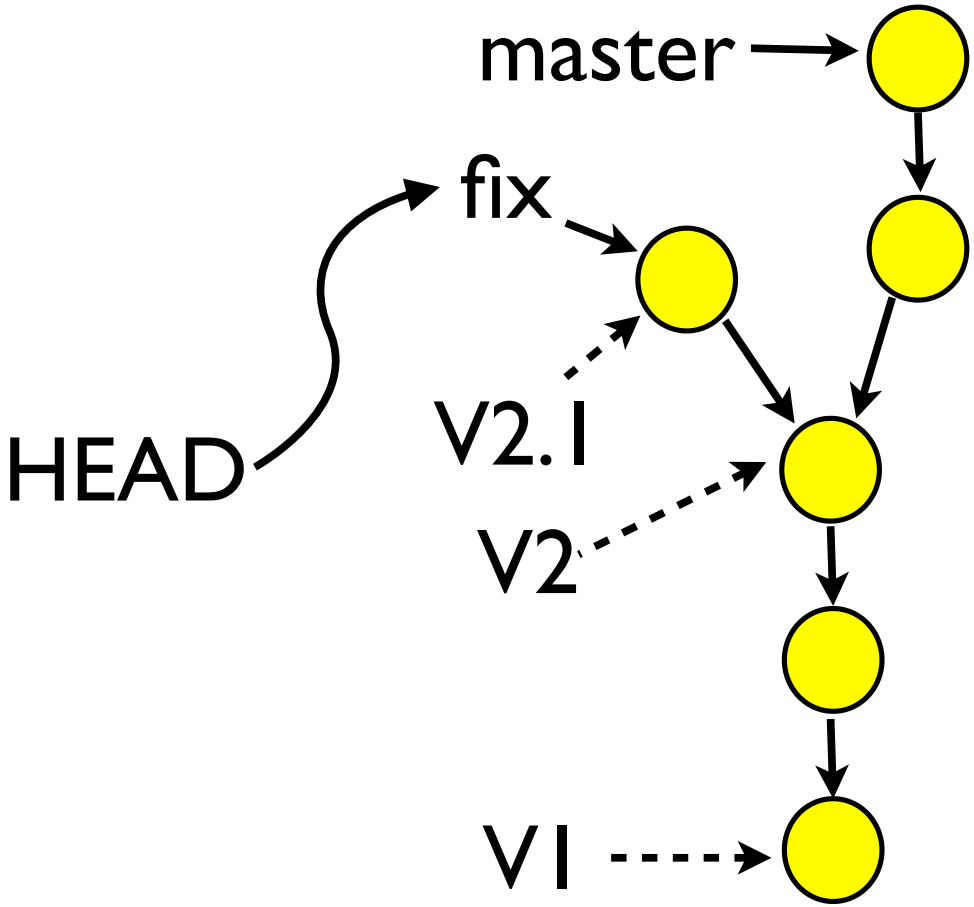

```
$ csc checkout V2
$ edit files
$ csc branch fix
```

Project Archive



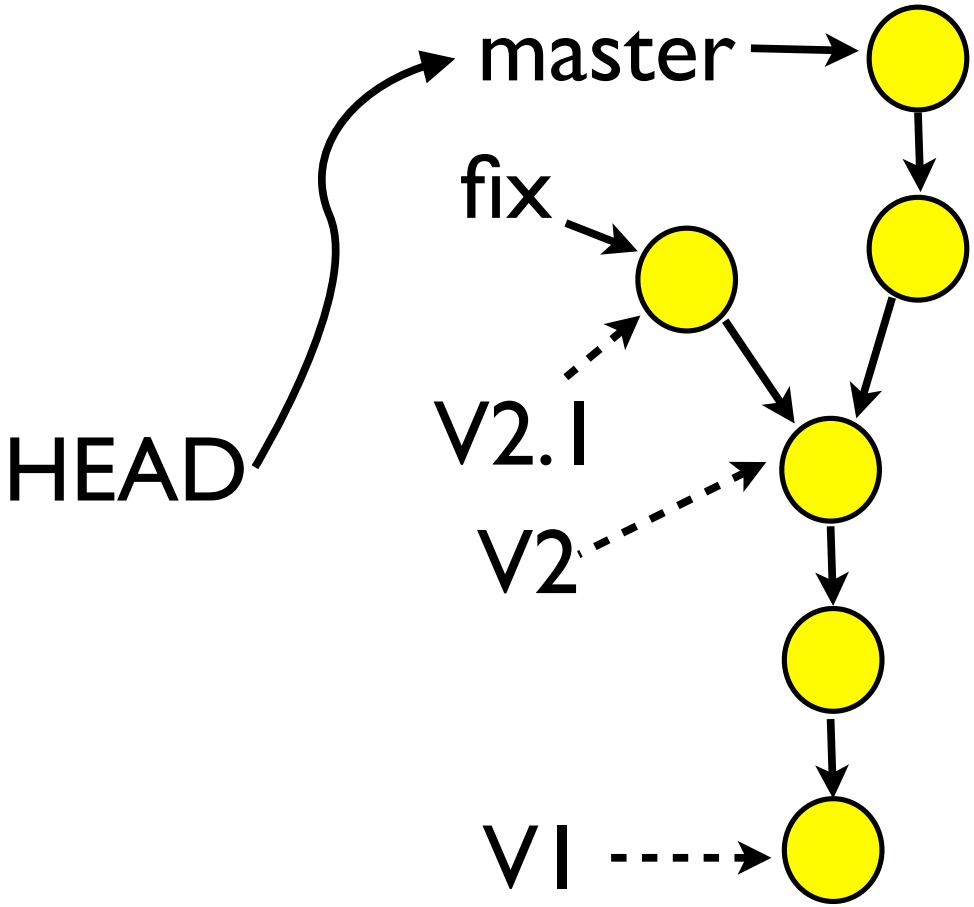

```
$ csc checkout V2
$ edit files
$ csc branch fix
$ csc snap
```

Project Archive



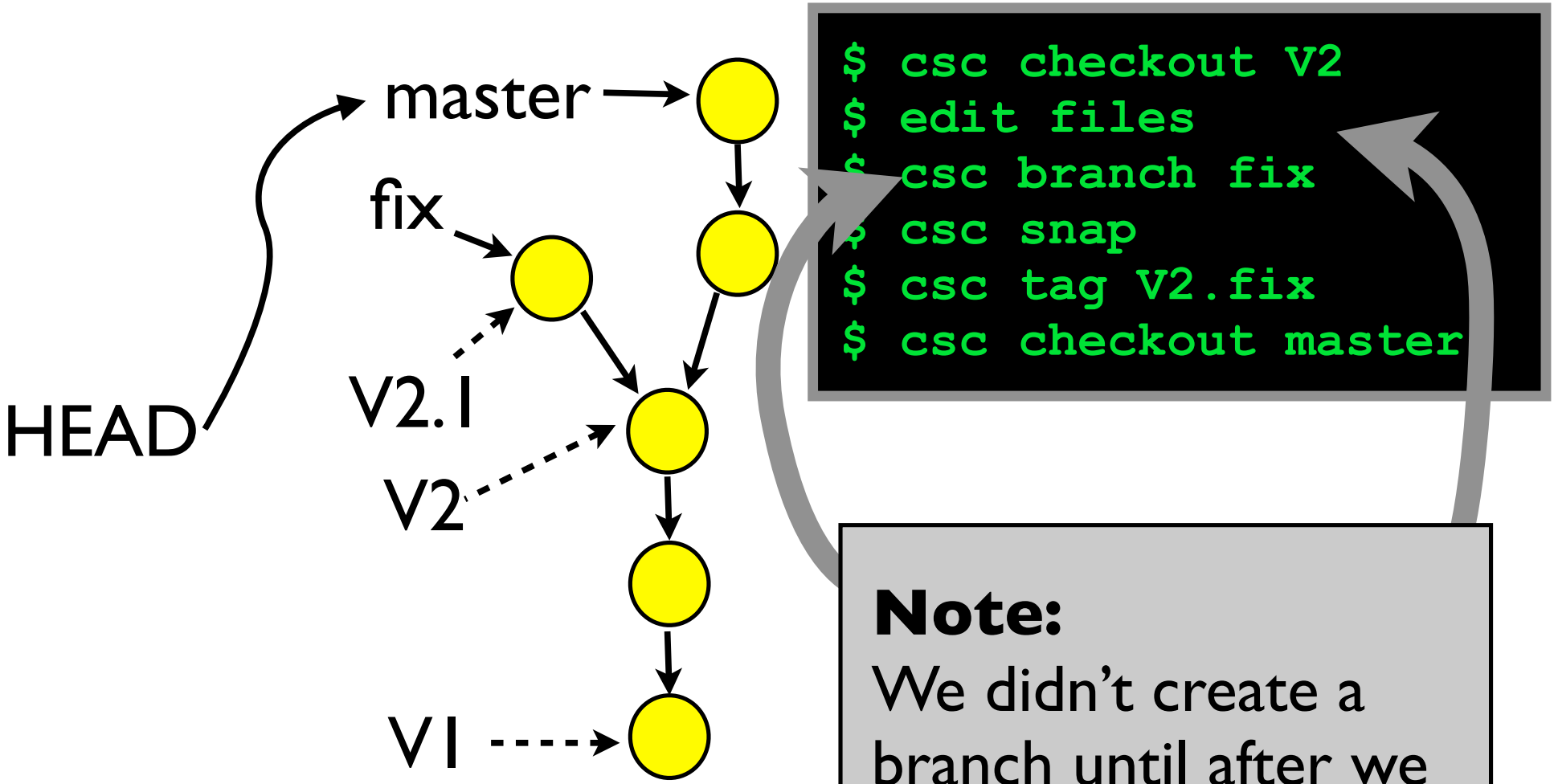

```
$ csc checkout V2
$ edit files
$ csc branch fix
$ csc snap
$ csc tag V2.1
```

Project Archive




```
$ csc checkout V2
$ edit files
$ csc branch fix
$ csc snap
$ csc tag V2.fix
$ csc checkout master
```

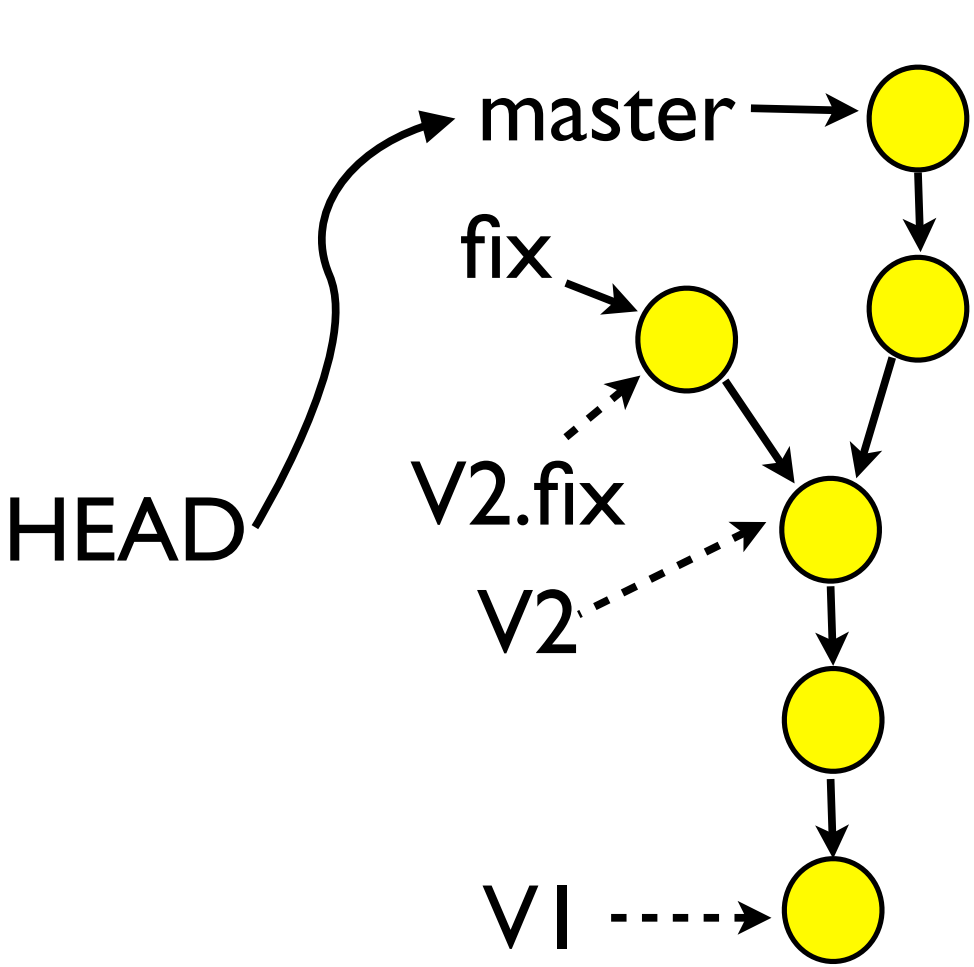
Project Archive



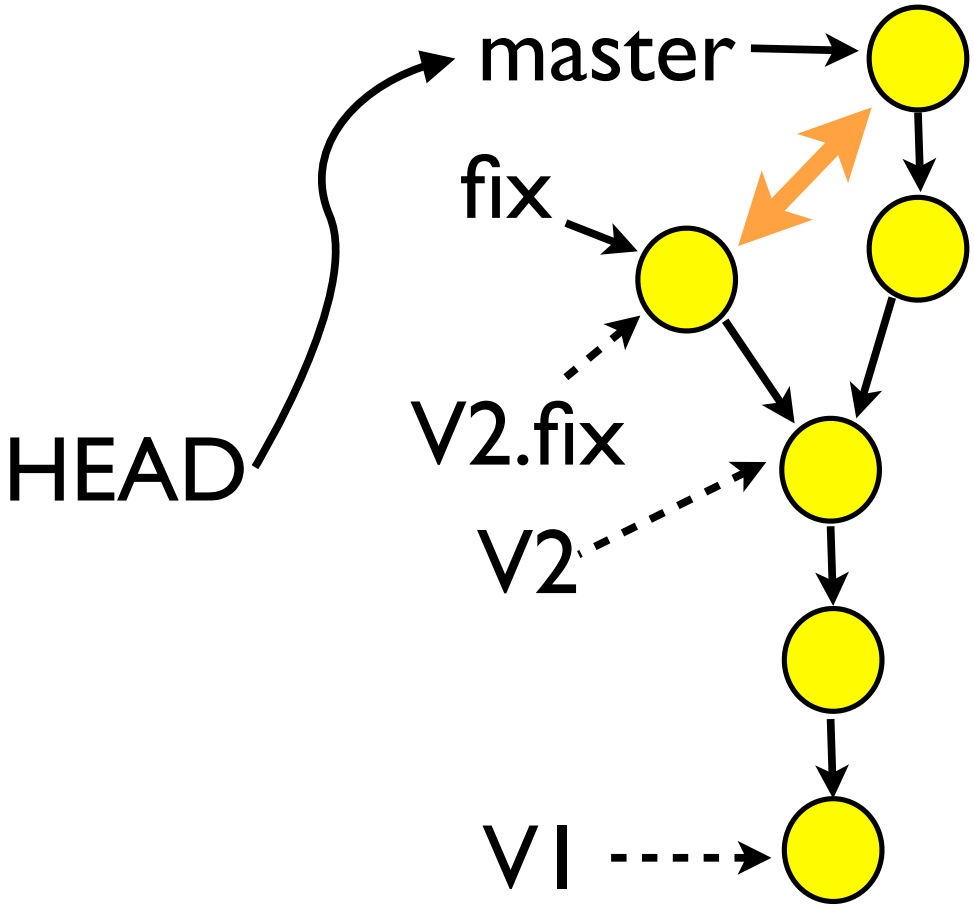
Note:
We didn't create a branch until after we edited files.

Merge Bug Fixes to Master

Project Archive

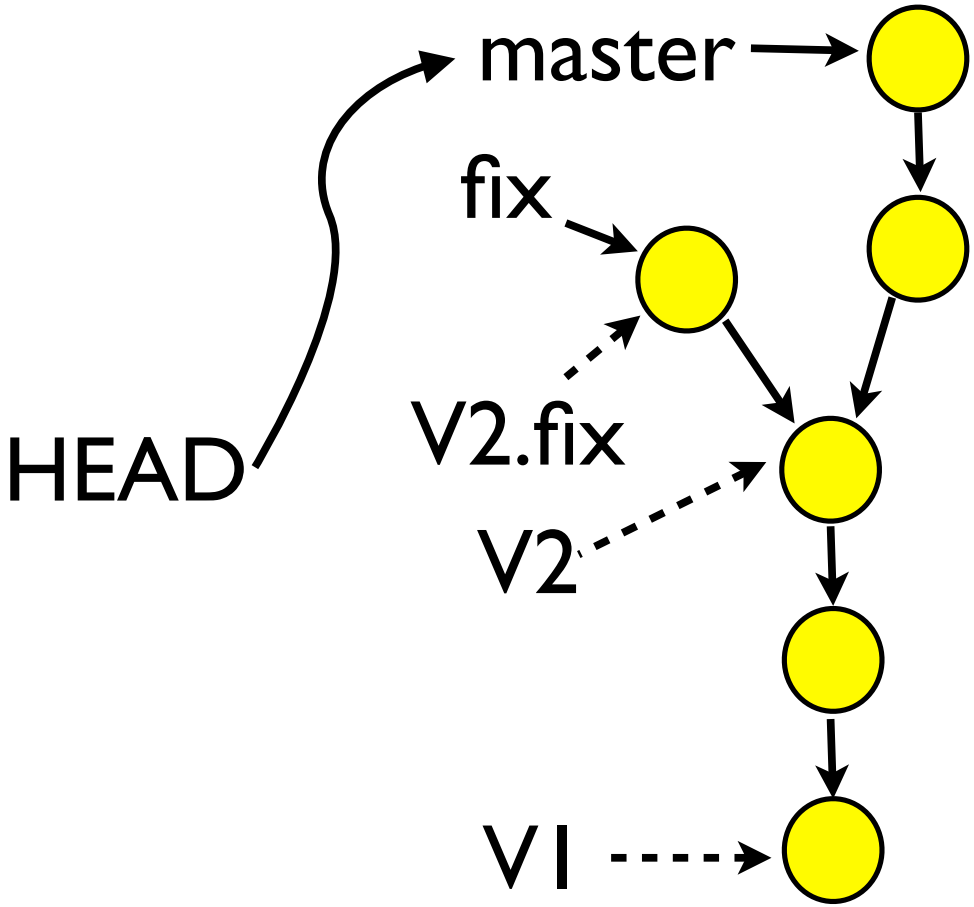


Project Archive

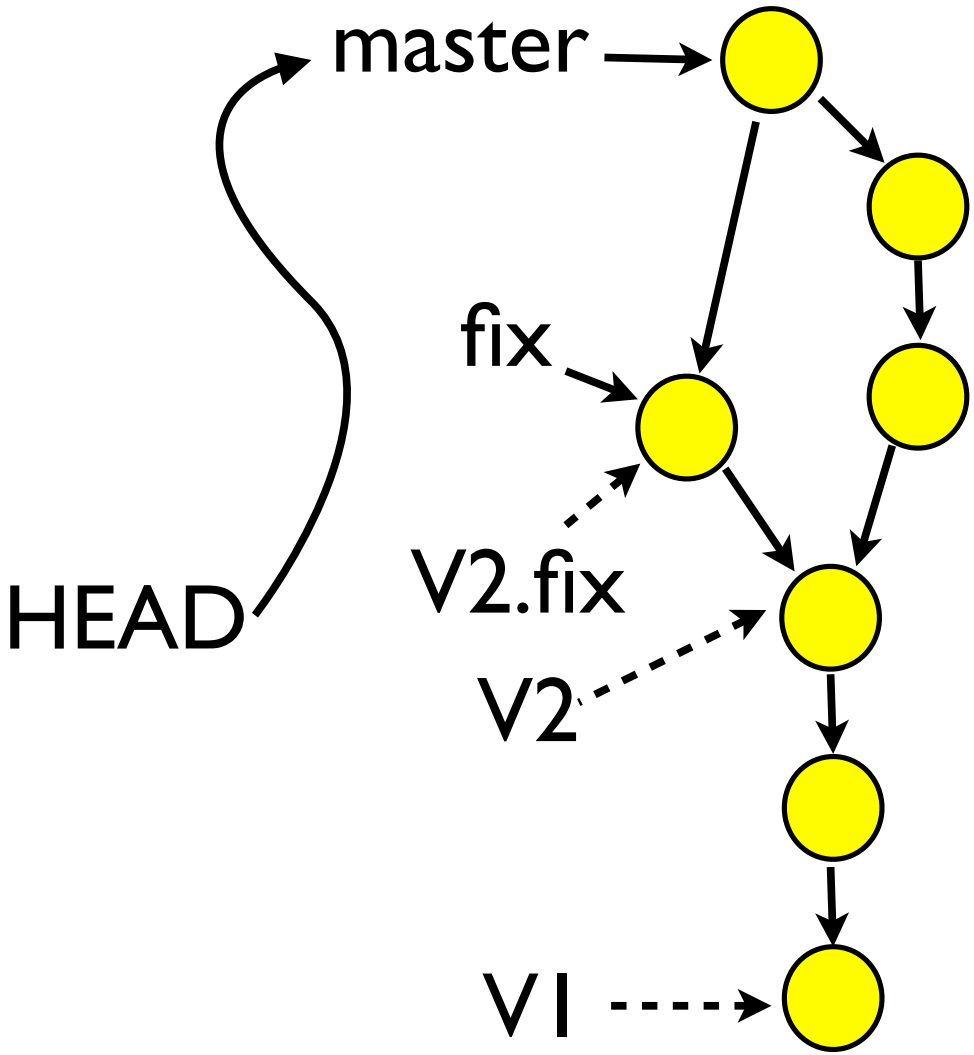



```
$ csc merge fix
```

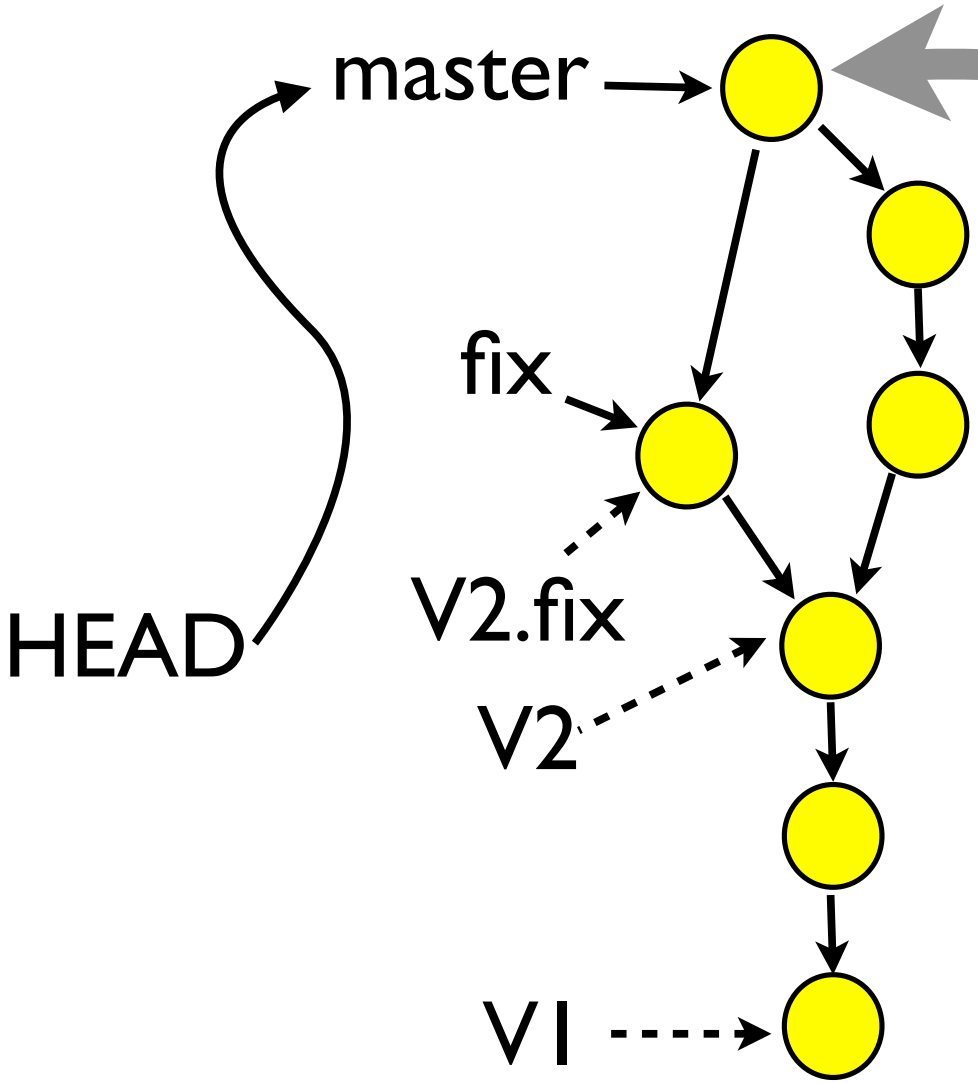
Project Archive




```
$ csc merge fix
(resolve conflict)
```

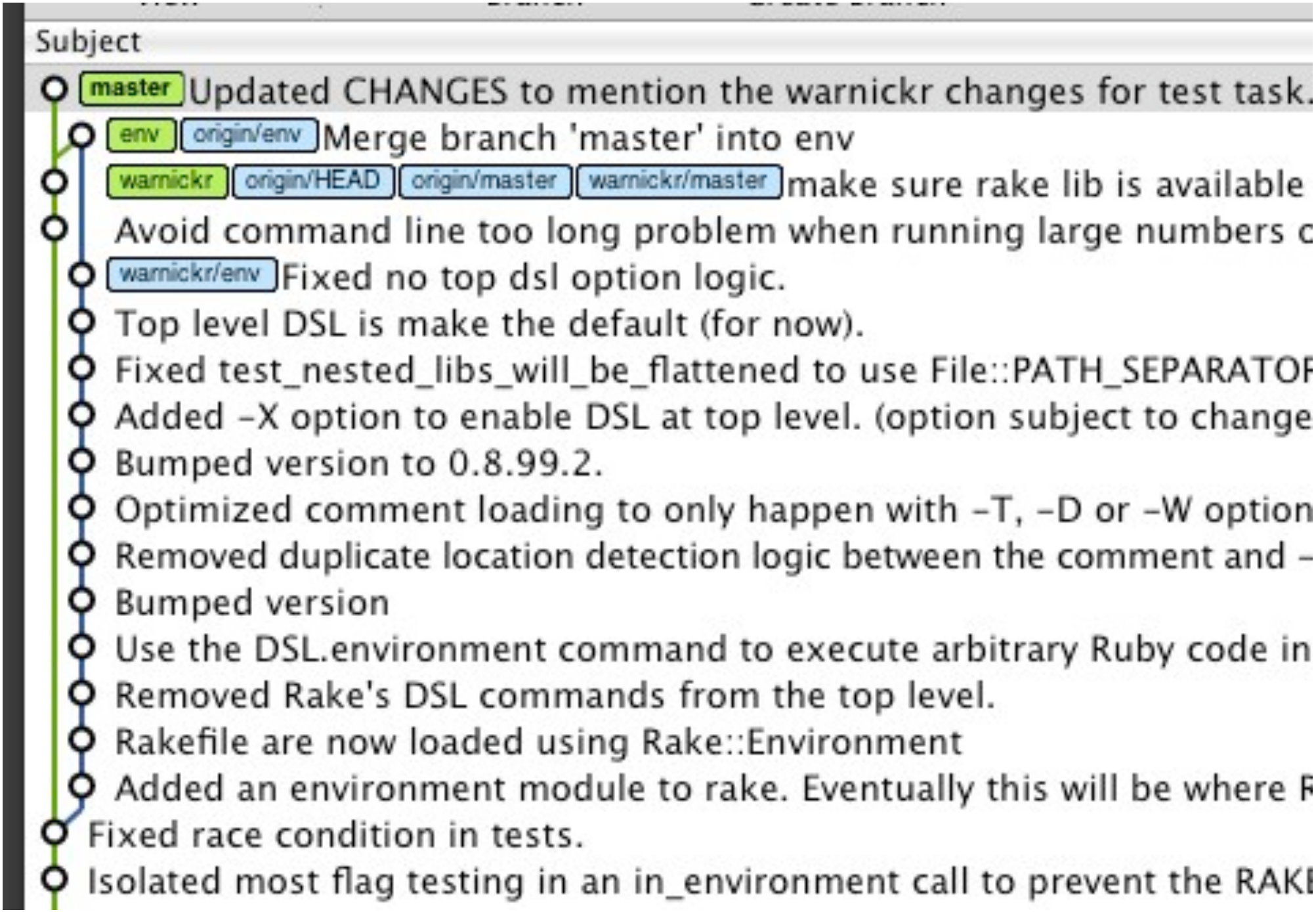


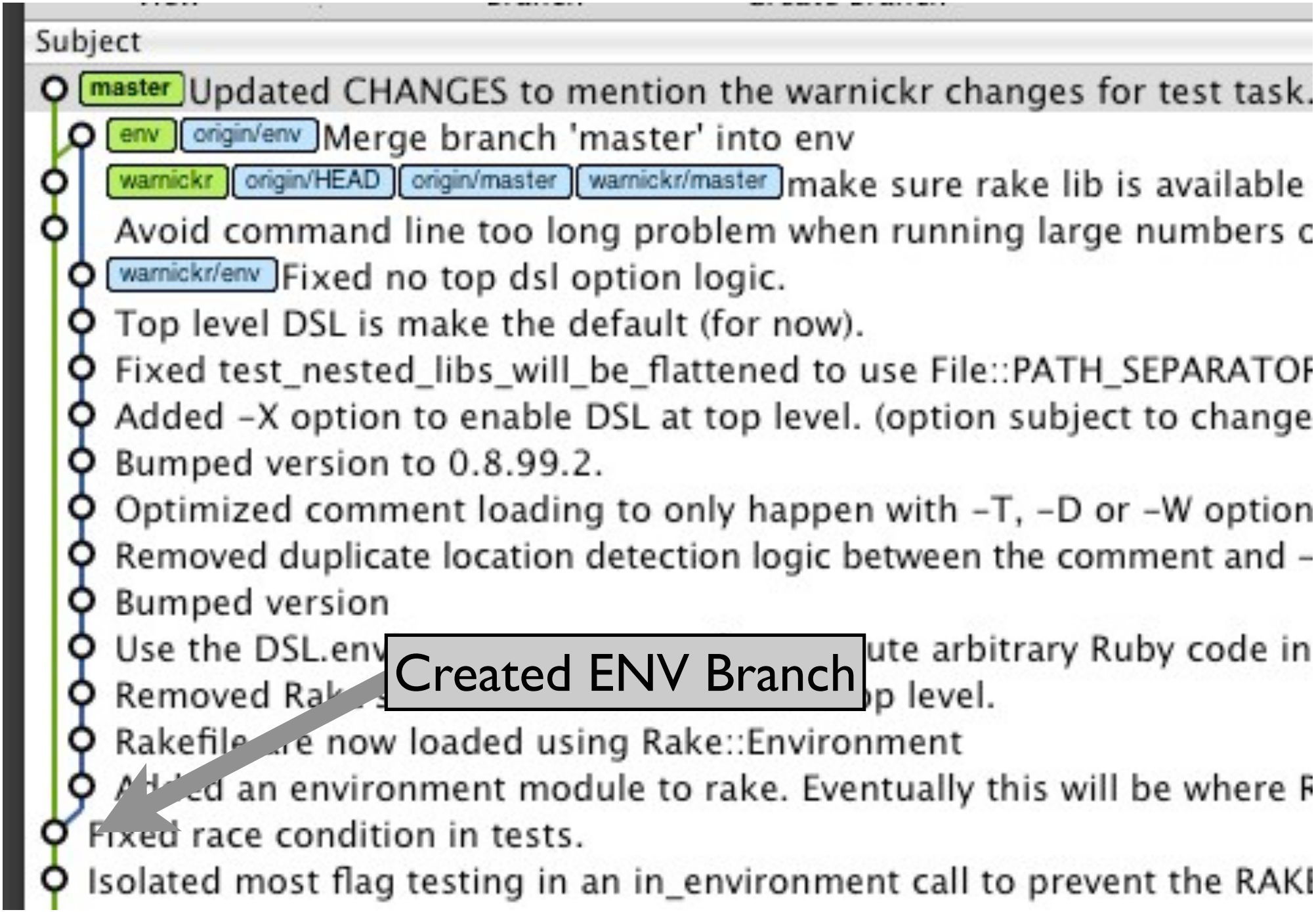

```
$ csc merge fix
(resolve conflict)
$ csc snap
```

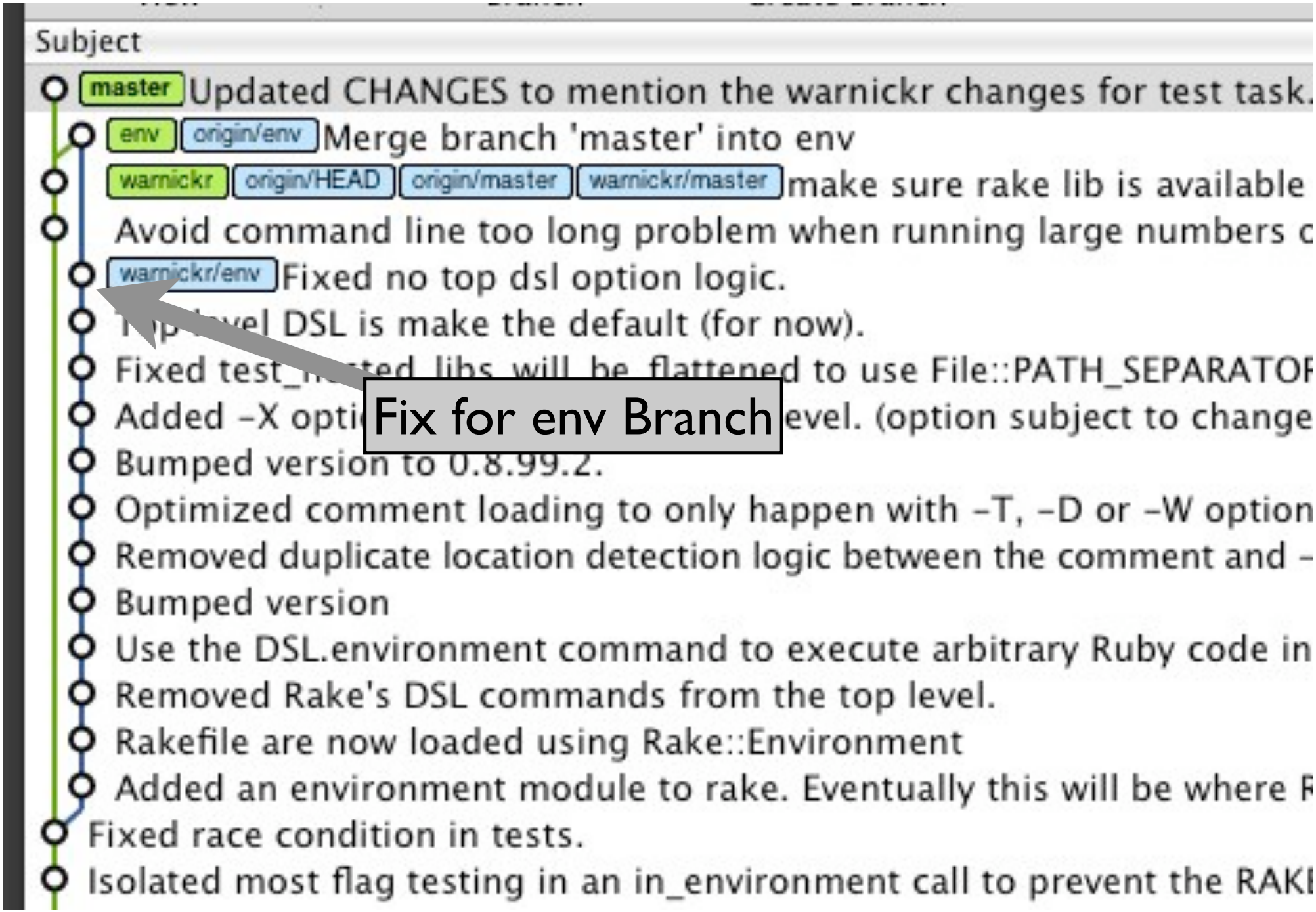


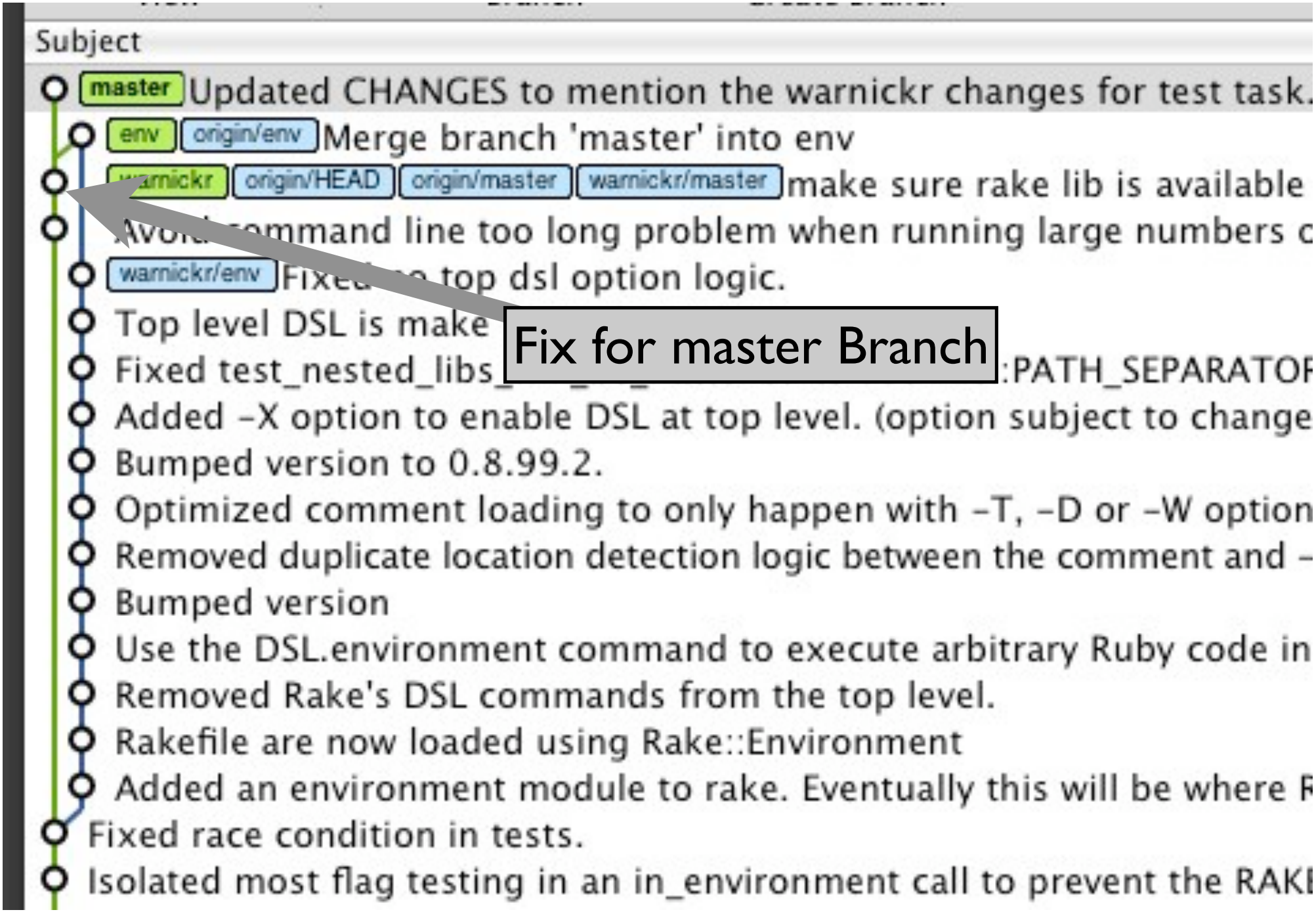

```
$ csc merge fix
(resolve conflict)
$ csc snap
```

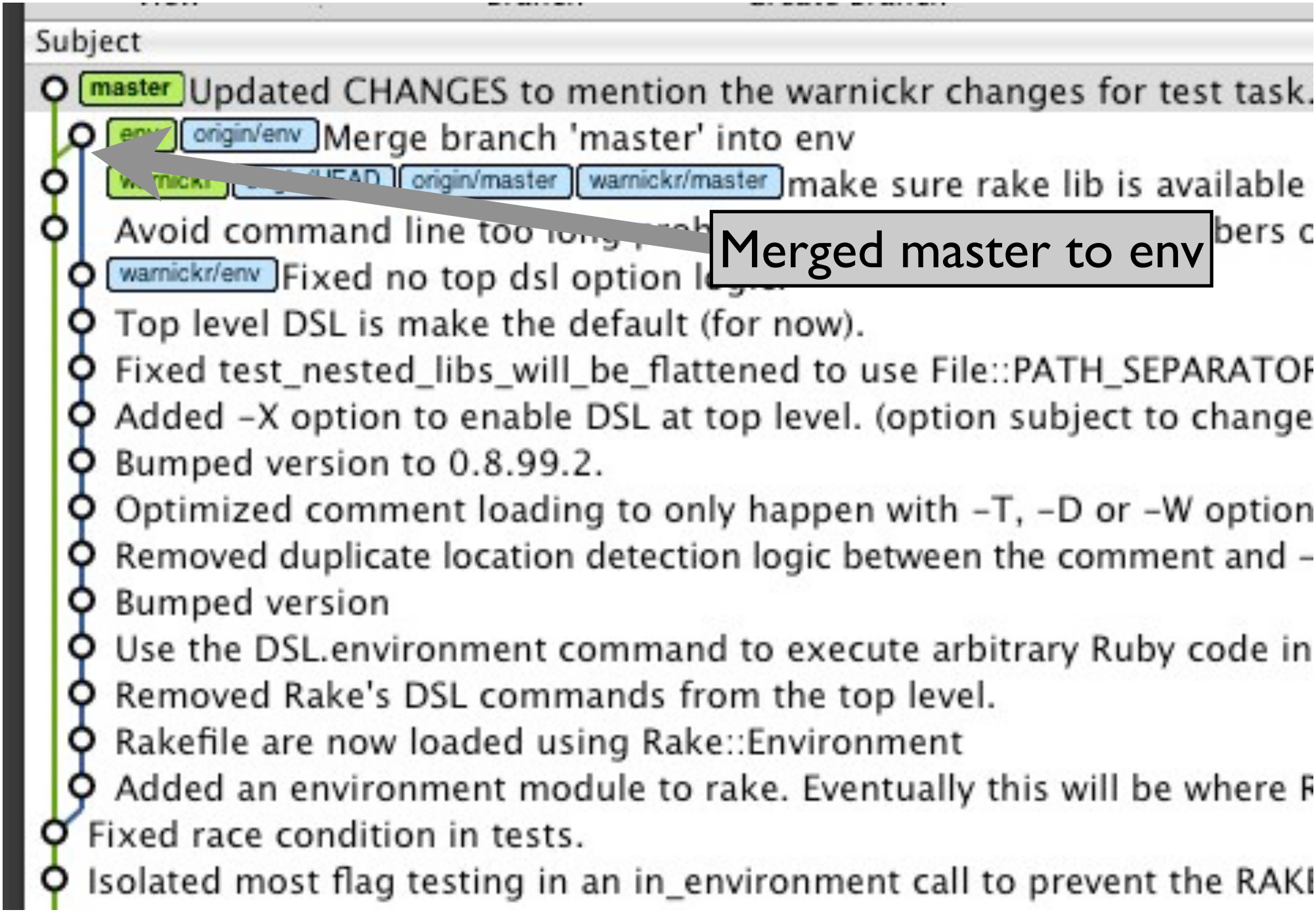
Note:
This snapshot has two parents. The changes from both parents have been merged together.

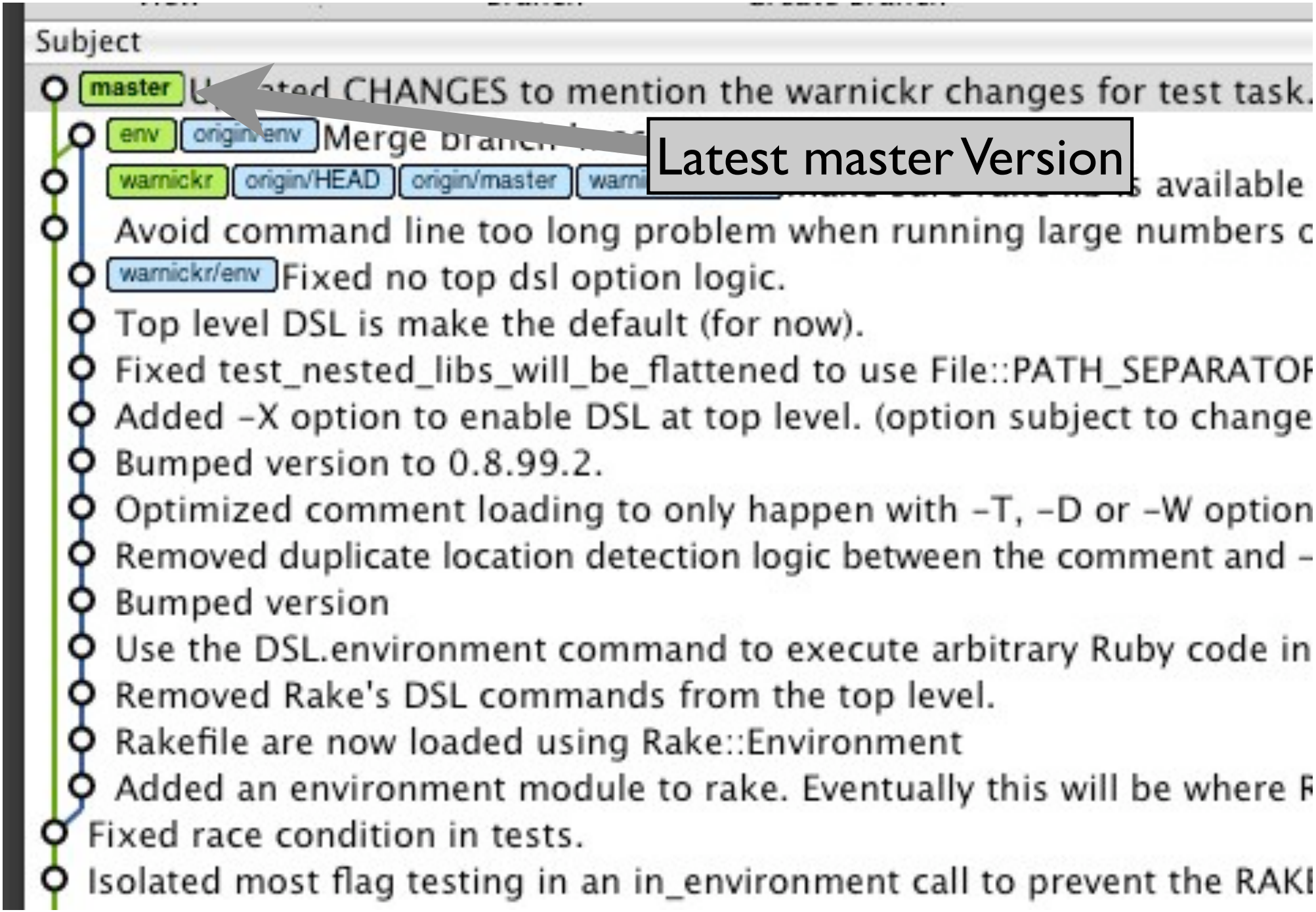

Subject
master
Updated CHANGES to mention the warnickr changes for test task.
env
origin/env
Merge branch 'master' into env
warnickr
origin/HEAD
origin/master
warnickr/master
make sure rake lib is available
Avoid command line too long problem when running large numbers c
warnickr/env
Fixed no top dsl option logic.
Top level DSL is make the default (for now).
Fixed test_nested_libs_will_be_flattened to use File::PATH_SEPARATOR
Added -X option to enable DSL at top level. (option subject to change
Bumped version to 0.8.99.2.
Optimized comment loading to only happen with -T, -D or -W option
Removed duplicate location detection logic between the comment and -
Bumped version
Use the DSL.environment command to execute arbitrary Ruby code in
Removed Rake's DSL commands from the top level.
Rakefile are now loaded using Rake::Environment
Added an environment module to rake. Eventually this will be where R
Fixed race condition in tests.
Isolated most flag testing in an in_environment call to prevent the RAKE

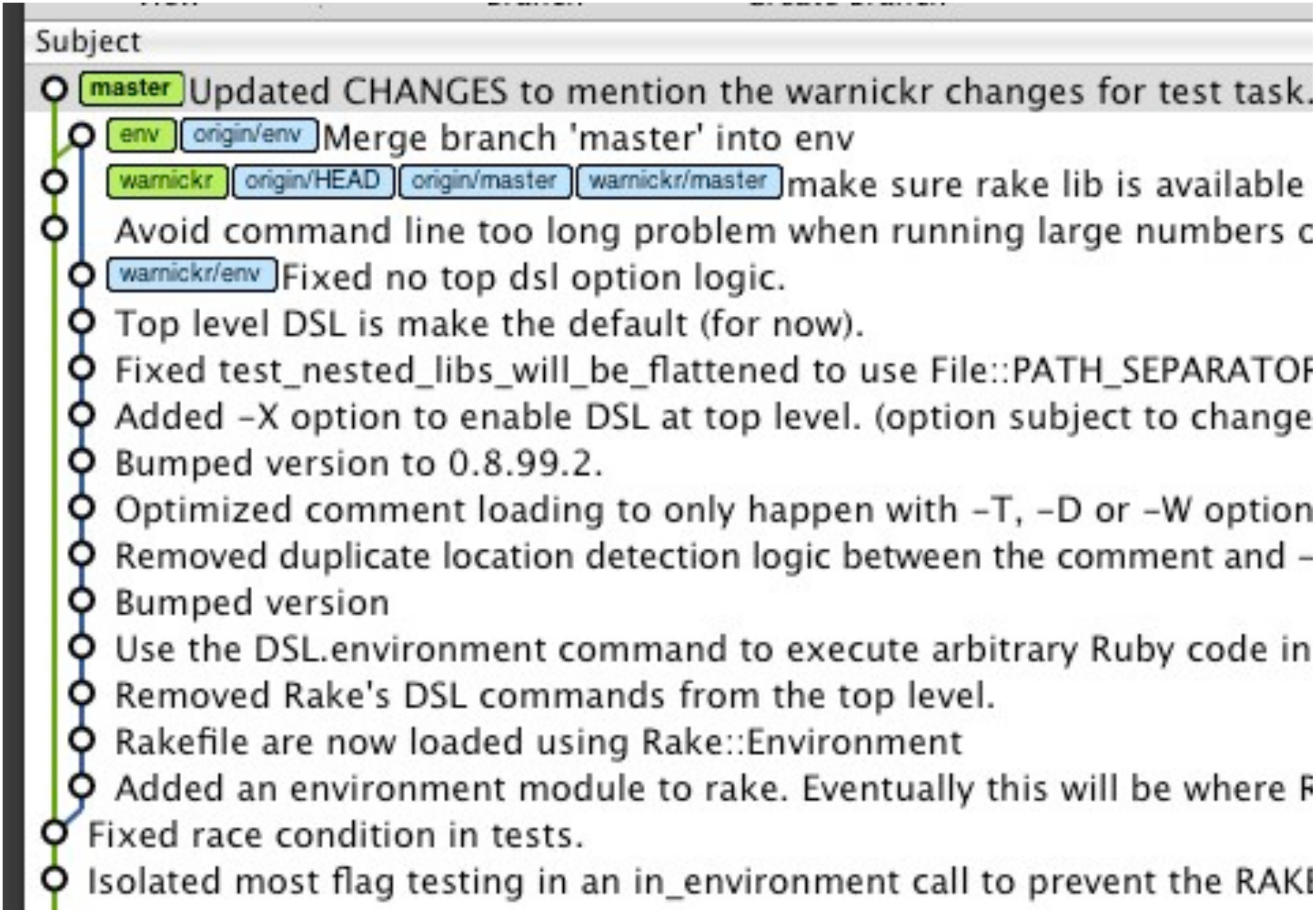

Subject
master Updated CHANGES to mention the warnickr changes for test task.
env origin/env Merge branch 'master' into env
warnickr origin/HEAD origin/master warnickr/master make sure rake lib is available
Avoid command line too long problem when running large numbers c
warnickr/env Fixed no top dsl option logic.
Top level DSL is make the default (for now).
Fixed test_nested_libs_will_be_flattened to use File::PATH_SEPARATOR
Added -X option to enable DSL at top level. (option subject to change
Bumped version to 0.8.99.2.
Optimized comment loading to only happen with -T, -D or -W option
Removed duplicate location detection logic between the comment and -
Bumped version
Use the DSL.env ute arbitrary Ruby code in
Removed Rak op level.
Rakefile re now loaded using Rake::Environment
Added an environment module to rake. Eventually this will be where R
Fixed race condition in tests.
Isolated most flag testing in an in_environment call to prevent the RAKE
Created ENV Branch


Subject
master Updated CHANGES to mention the warnickr changes for test task.
env origin/env Merge branch 'master' into env
warnickr origin/HEAD origin/master warnickr/master make sure rake lib is available
Avoid command line too long problem when running large numbers c
warnickr/env Fixed no top dsl option logic.
Fix for env Branch


Subject
master
Updated CHANGES to mention the warnickr changes for test task.
env
origin/env
Merge branch 'master' into env
origin/HEAD
origin/master
warnickr/master
make sure rake lib is available
warnickr/env
Top level DSL is make
Fix for master Branch
Fixed test_nested_libs
:PATH_SEPARATOR
Added -X option to enable DSL at top level. (option subject to change
Bumped version to 0.8.99.2.
Optimized comment loading to only happen with -T, -D or -W option
Removed duplicate location detection logic between the comment and -
Bumped version
Use the DSL.environment command to execute arbitrary Ruby code in
Removed Rake's DSL commands from the top level.
Rakefile are now loaded using Rake::Environment
Added an environment module to rake. Eventually this will be where R
Fixed race condition in tests.
Isolated most flag testing in an in_environment call to prevent the RAKE

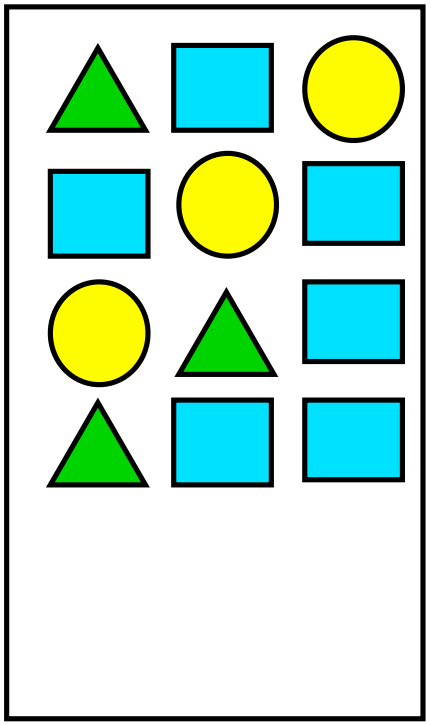

Subject
master
Updated CHANGES to mention the warnickr changes for test task.
origin/env
Merge branch 'master' into env
origin/master
warnickr/master
make sure rake lib is available
Merged master to env
warnickr/env
Top level DSL is make the default (for now).
Fixed test_nested_libs_will_be_flattened to use File::PATH_SEPARATOR
Added -X option to enable DSL at top level. (option subject to change
Bumped version to 0.8.99.2.
Optimized comment loading to only happen with -T, -D or -W option
Removed duplicate location detection logic between the comment and -
Bumped version
Use the DSL.environment command to execute arbitrary Ruby code in
Removed Rake's DSL commands from the top level.
Rakefile are now loaded using Rake::Environment
Added an environment module to rake. Eventually this will be where R
Fixed race condition in tests.
Isolated most flag testing in an in_environment call to prevent the RAKE


Subject
master
CHANGES to mention the warnickr changes for test task.
env
origin/env
Merge
Latest master Version
warnickr
origin/HEAD
origin/master
available
Avoid command line too long problem when running large numbers c
warnickr/env
Fixed no top dsl option logic.
Top level DSL is make the default (for now).
Fixed test_nested_libs_will_be_flattened to use File::PATH_SEPARATOR
Added -X option to enable DSL at top level. (option subject to change
Bumped version to 0.8.99.2.
Optimized comment loading to only happen with -T, -D or -W option
Removed duplicate location detection logic between the comment and -
Bumped version
Use the DSL.environment command to execute arbitrary Ruby code in
Removed Rake's DSL commands from the top level.
Rakefile are now loaded using Rake::Environment
Added an environment module to rake. Eventually this will be where R
Fixed race condition in tests.
Isolated most flag testing in an in_environment call to prevent the RAKE


Subject
master
Updated CHANGES to mention the warnickr changes for test task.
env
origin/env
Merge branch 'master' into env
warnickr
origin/HEAD
origin/master
warnickr/master
make sure rake lib is available
Avoid command line too long problem when running large numbers c
warnickr/env
Fixed no top dsl option logic.
Top level DSL is make the default (for now).
Fixed test_nested_libs_will_be_flattened to use File::PATH_SEPARATOR
Added -X option to enable DSL at top level. (option subject to change
Bumped version to 0.8.99.2.
Optimized comment loading to only happen with -T, -D or -W option
Removed duplicate location detection logic between the comment and -
Bumped version
Use the DSL.environment command to execute arbitrary Ruby code in
Removed Rake's DSL commands from the top level.
Rakefile are now loaded using Rake::Environment
Added an environment module to rake. Eventually this will be where R
Fixed race condition in tests.
Isolated most flag testing in an in_environment call to prevent the RAKE

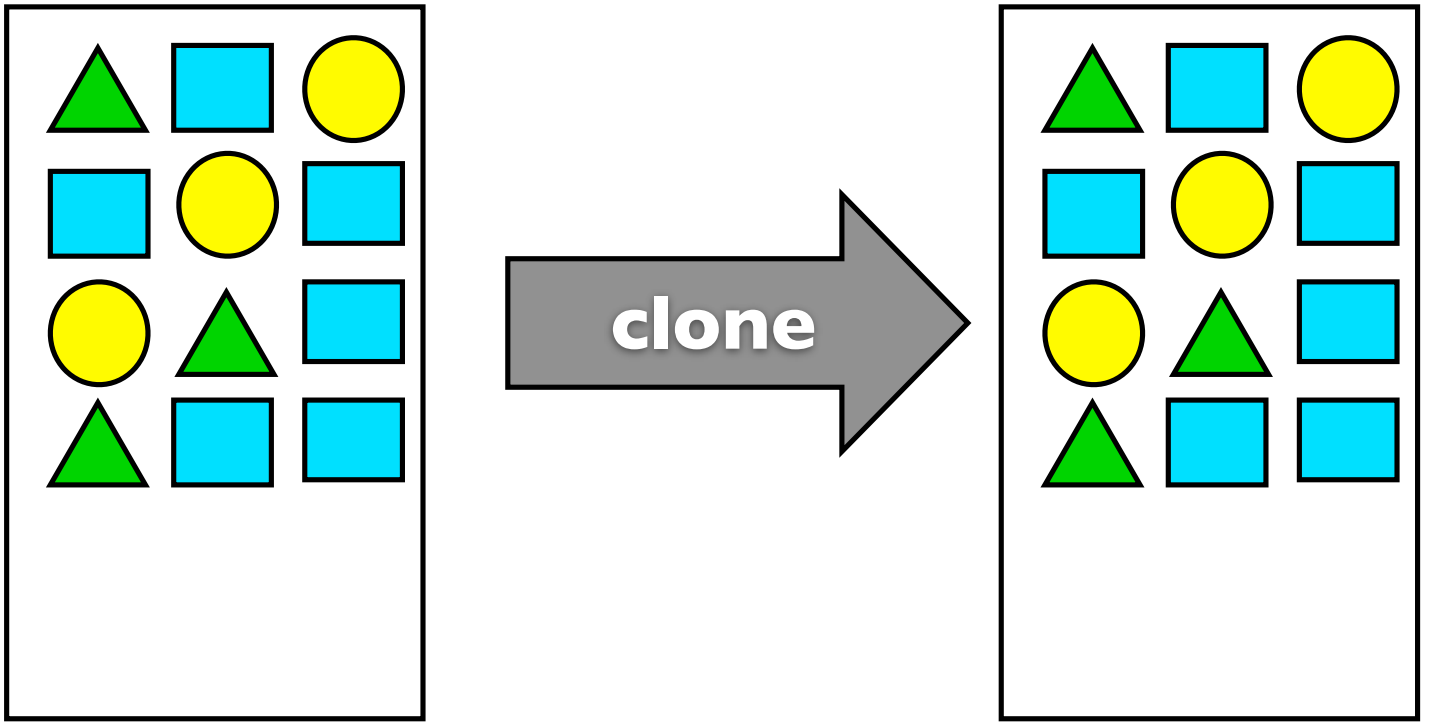
Sharing Code

My Archive



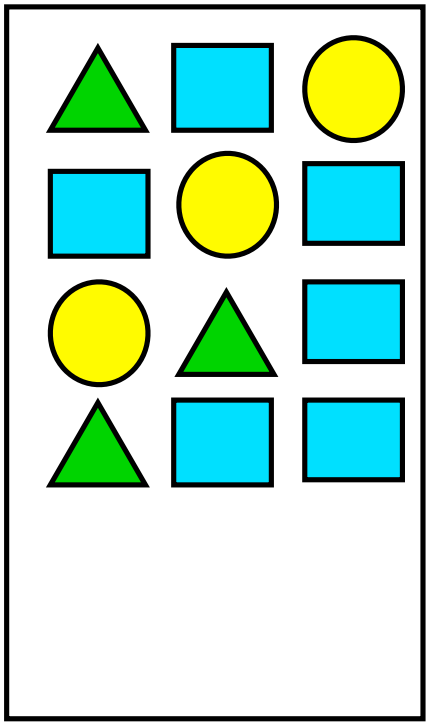
My Archive

Your Archive

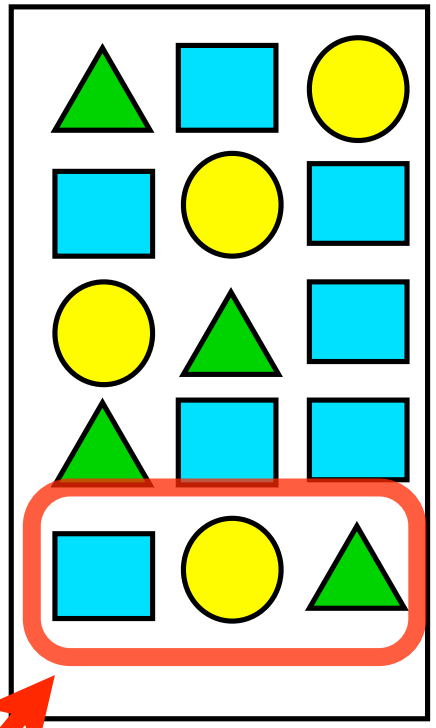



```
you$ csc clone url_for_my_repo
```

My Archive



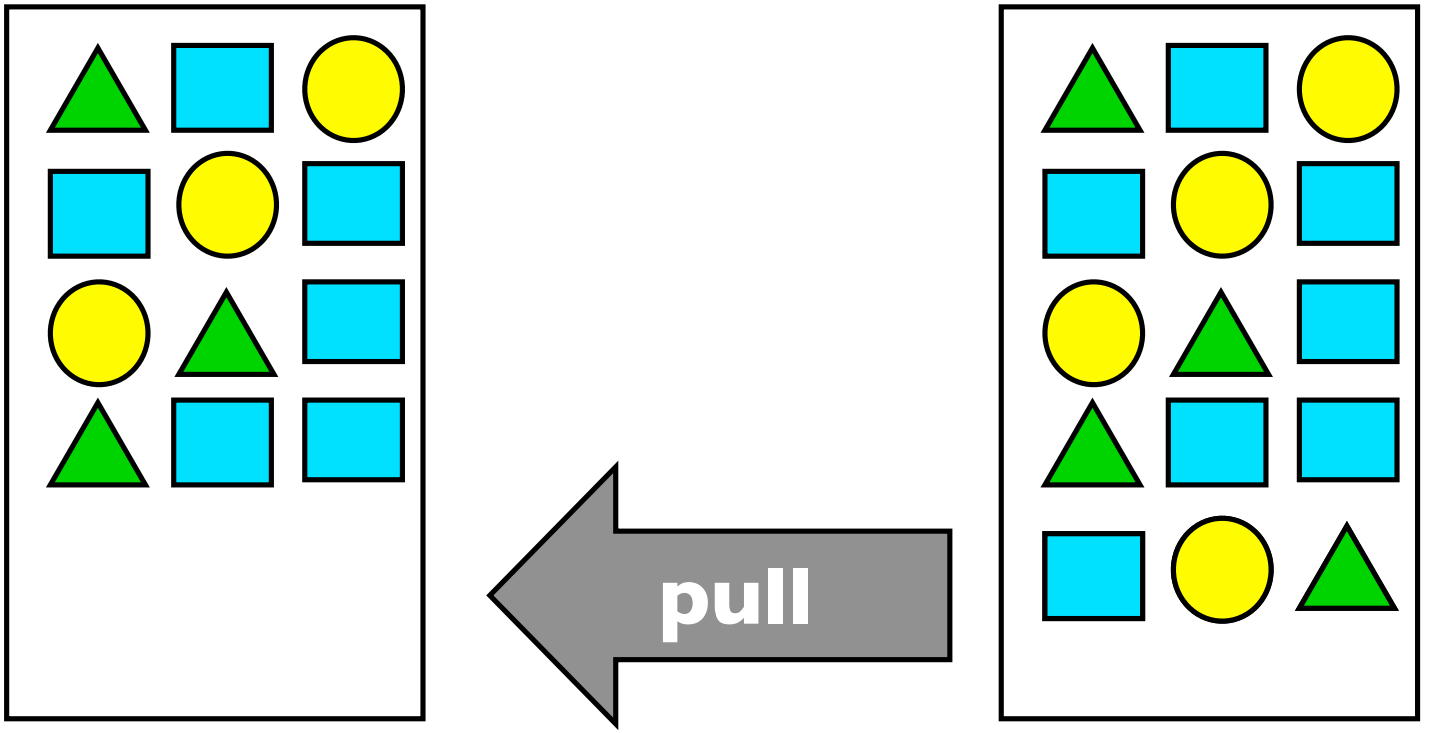
Your Archive



You make changes

My Archive

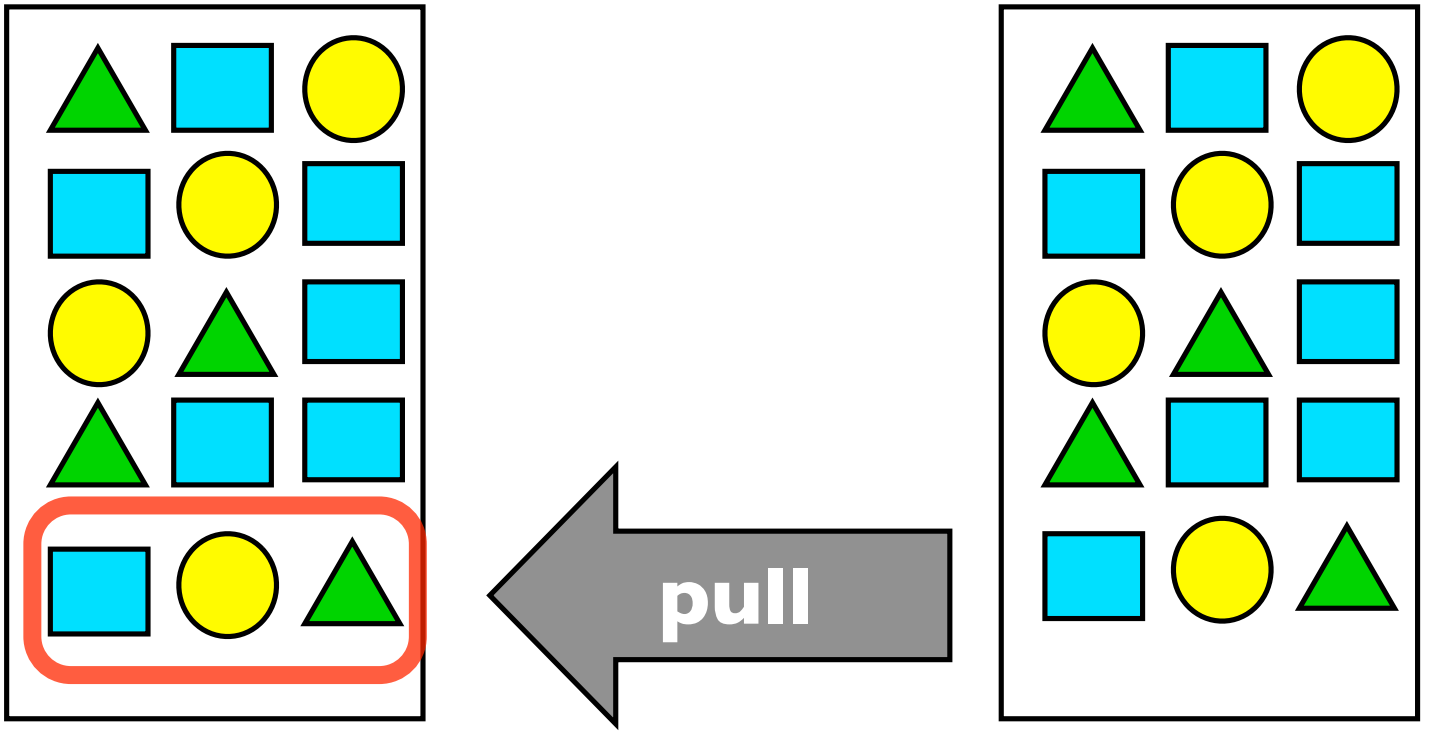
Your Archive




```
me$ csc pull url_for_my_repo
```

My Archive

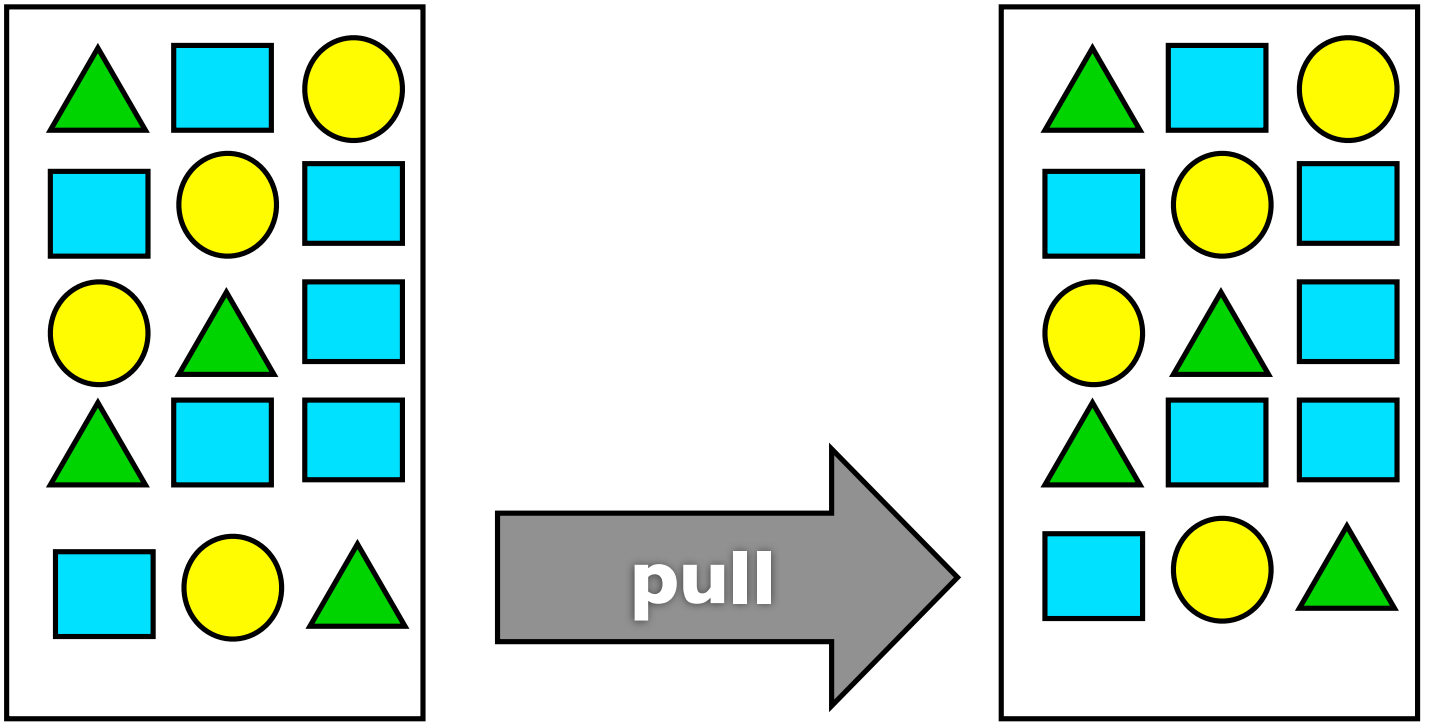
Your Archive




```
me$ csc pull url_for_my_repo
```

My Archive

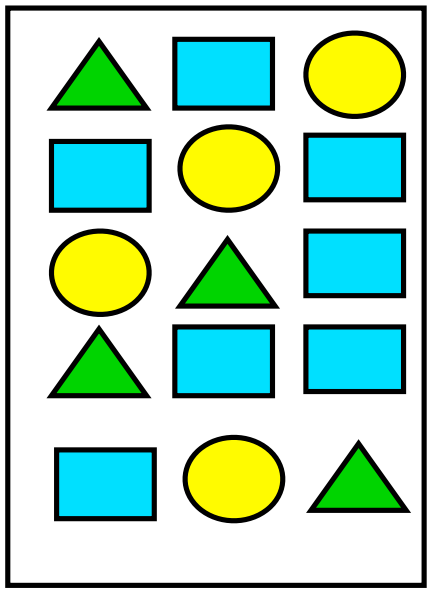
Your Archive

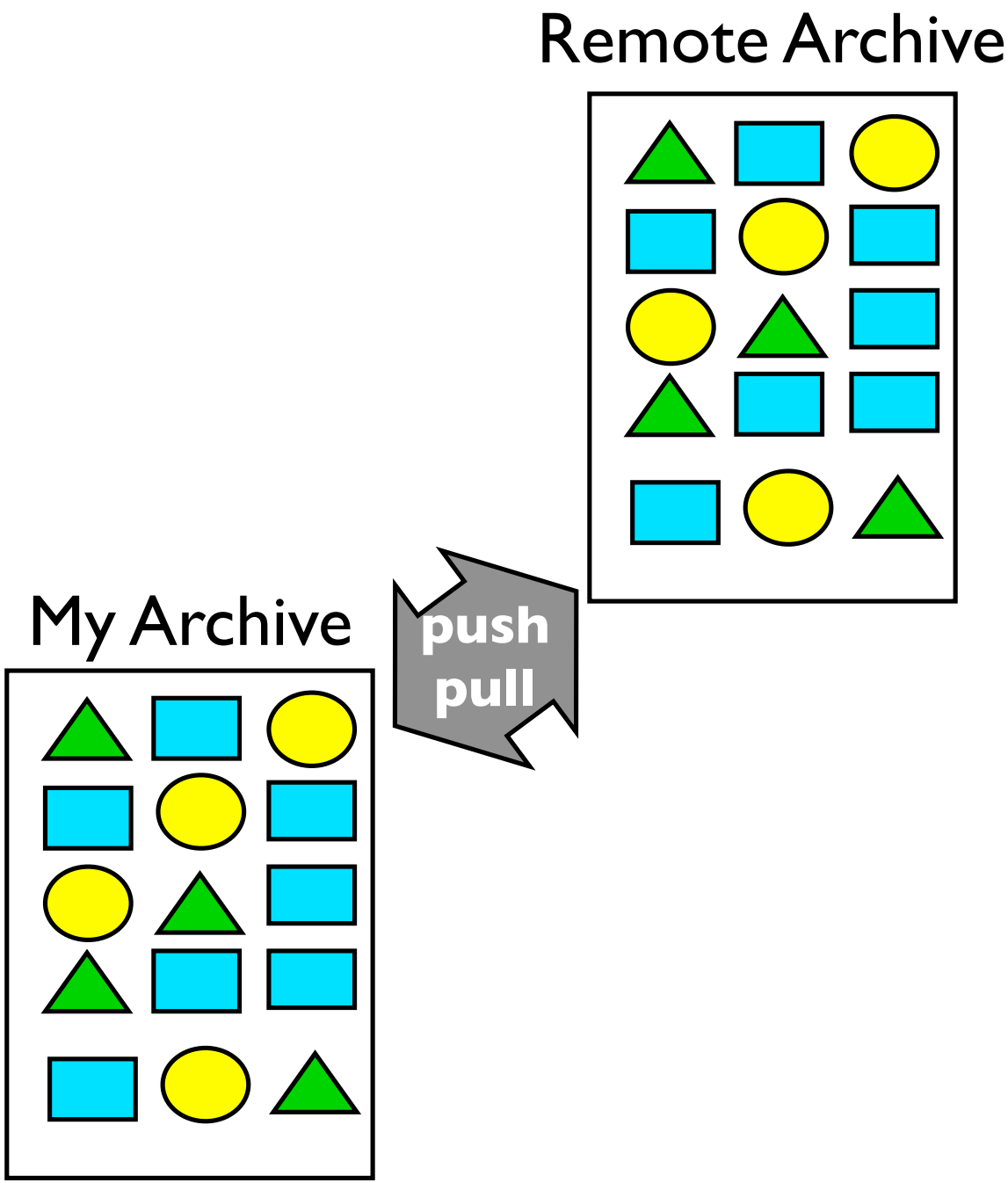


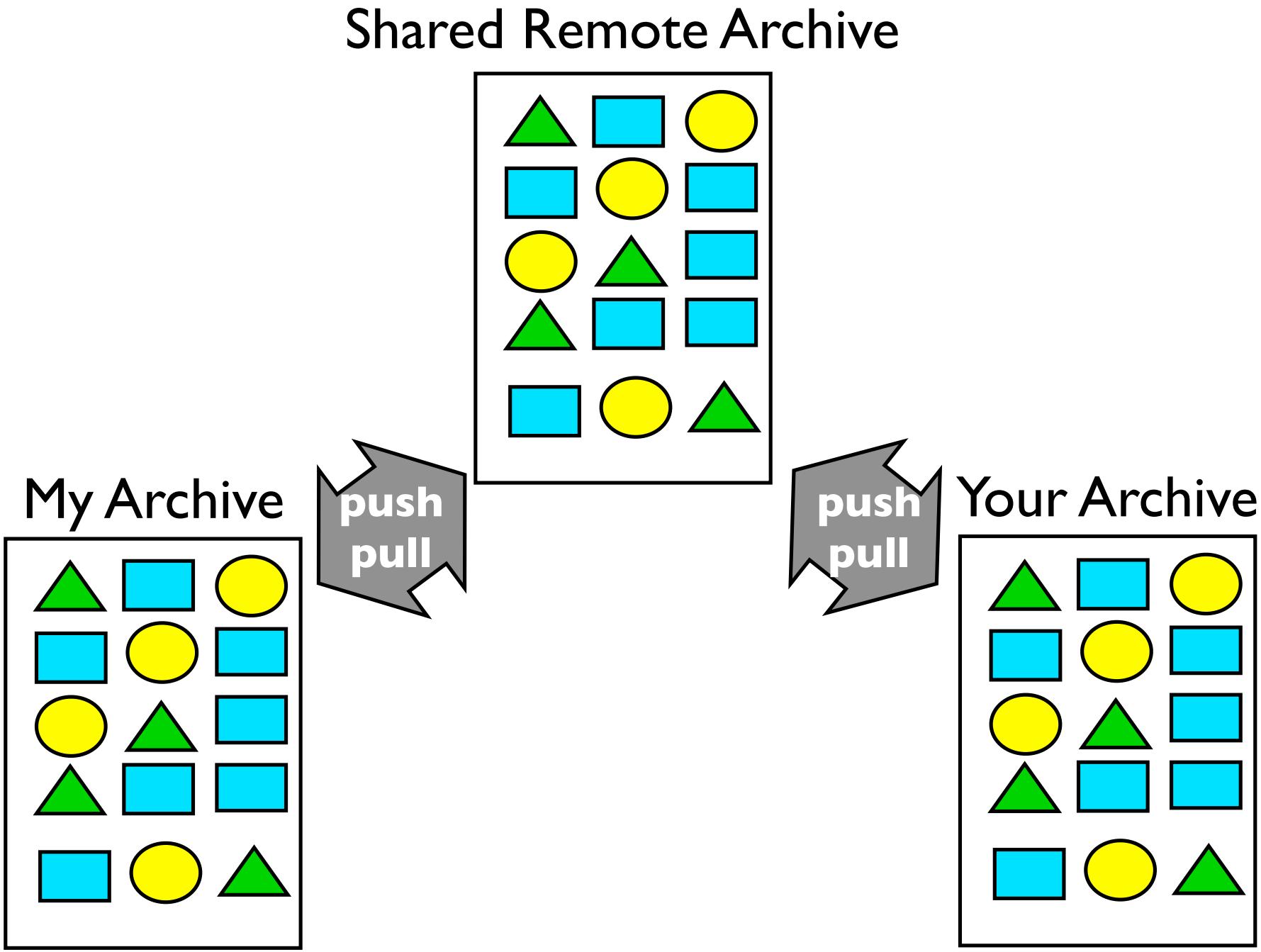

```
you$ csc pull url_for_my_repo
```

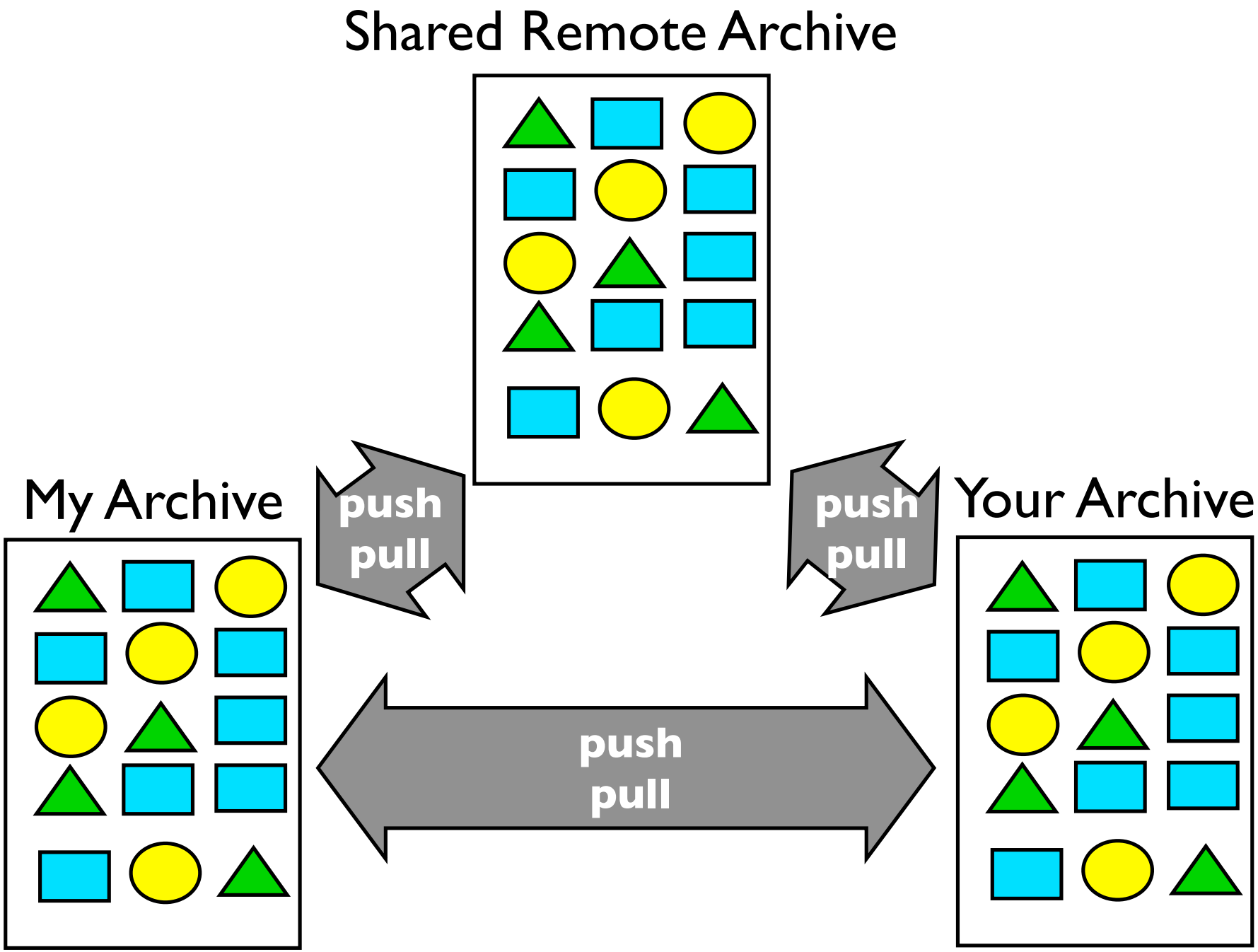
Common Remote Uses

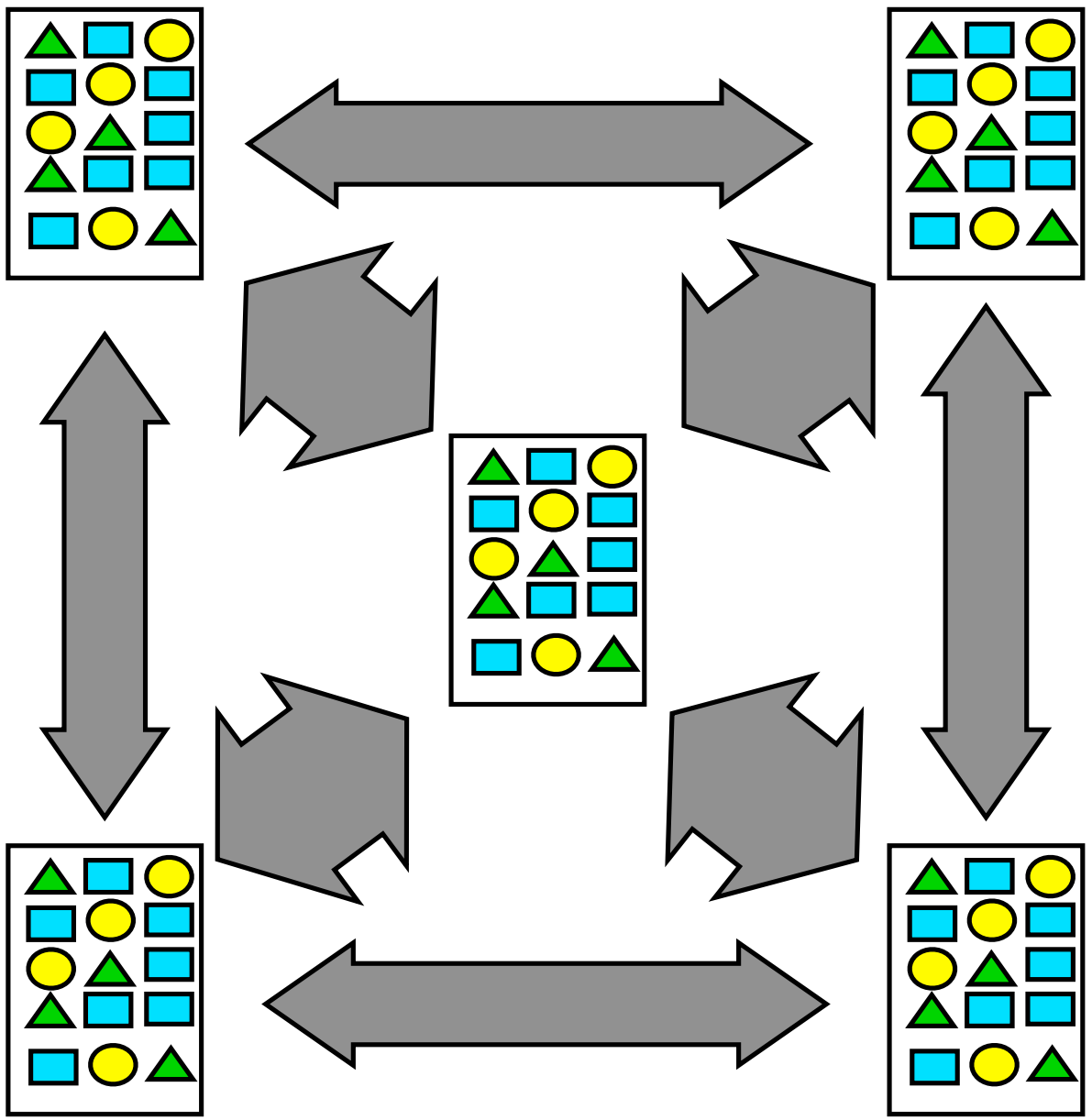
My Archive

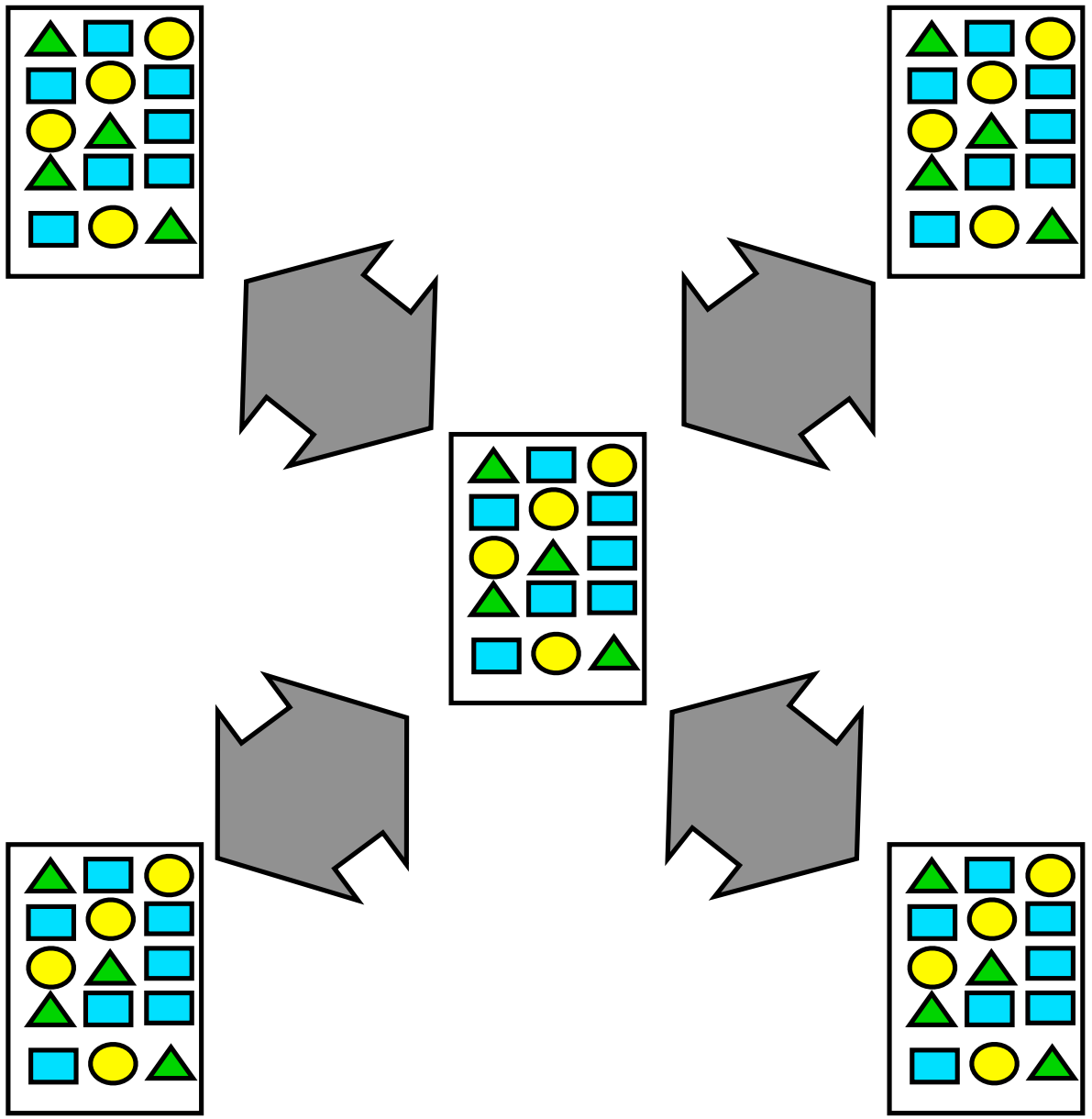


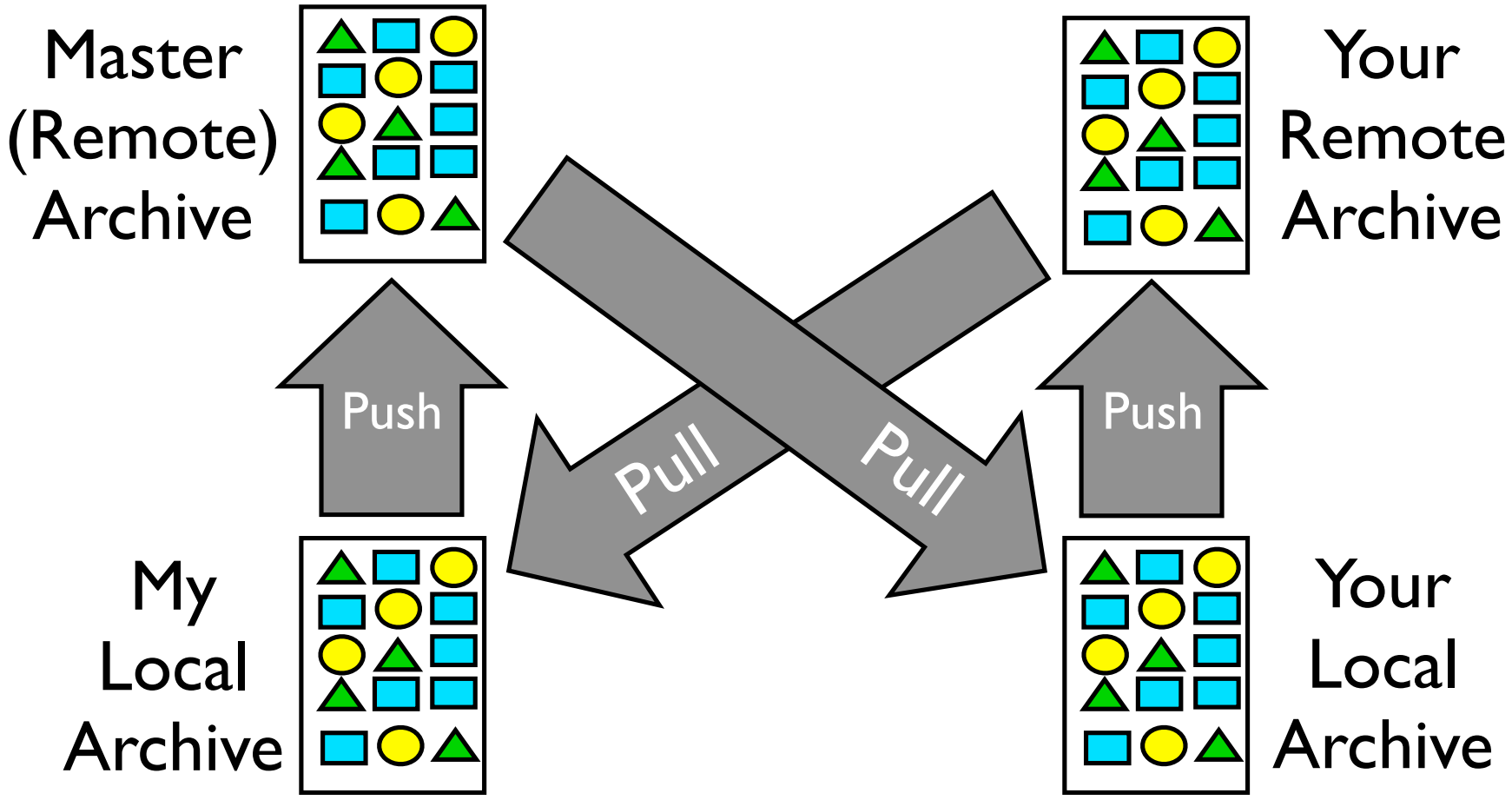

Remote Archive
My Archive
push
pull


Shared Remote Archive
My Archive
push
pull
push
pull
Your Archive

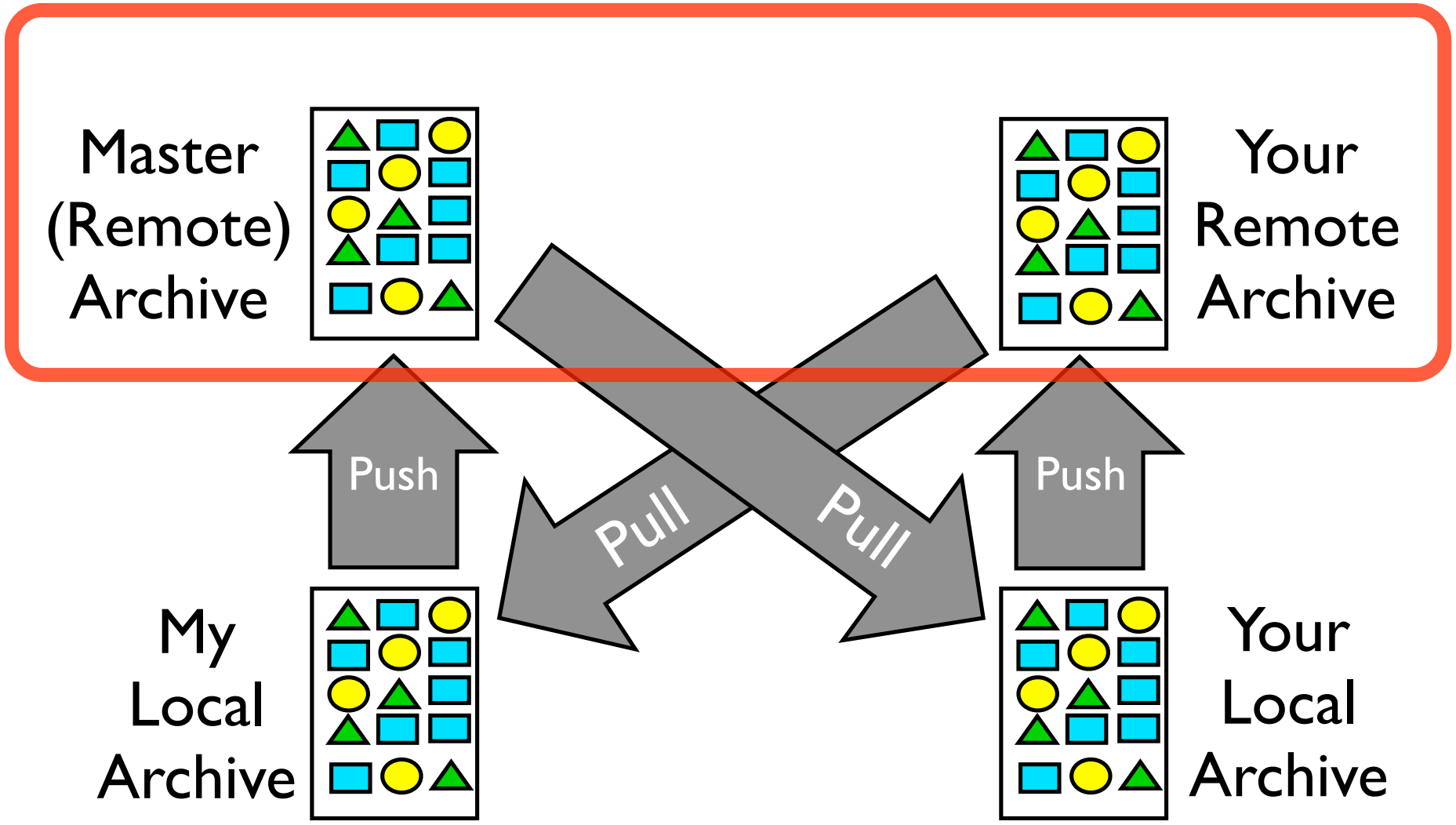

Shared Remote Archive
My Archive
Your Archive
push
pull
push
pull
push
pull

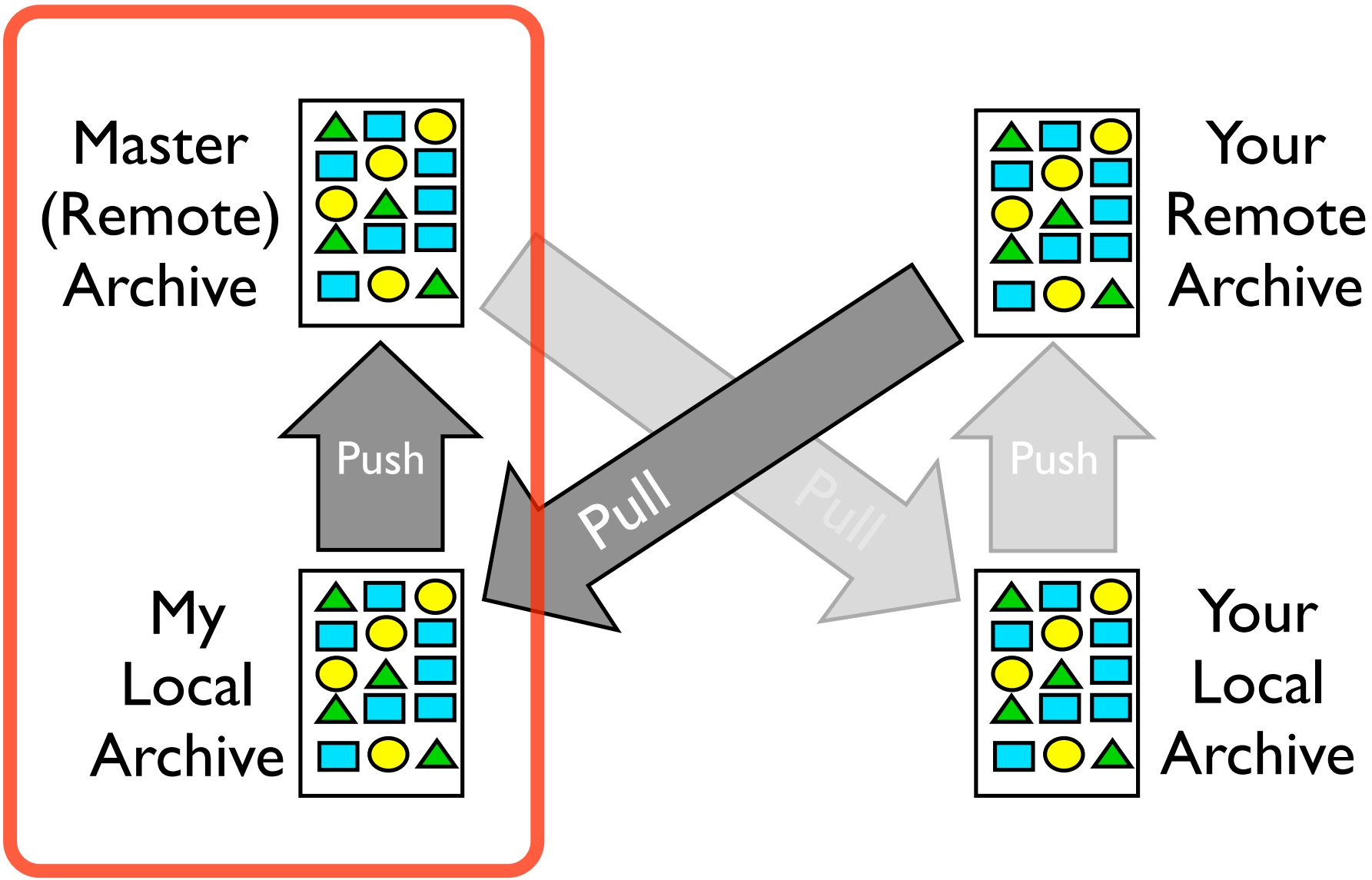


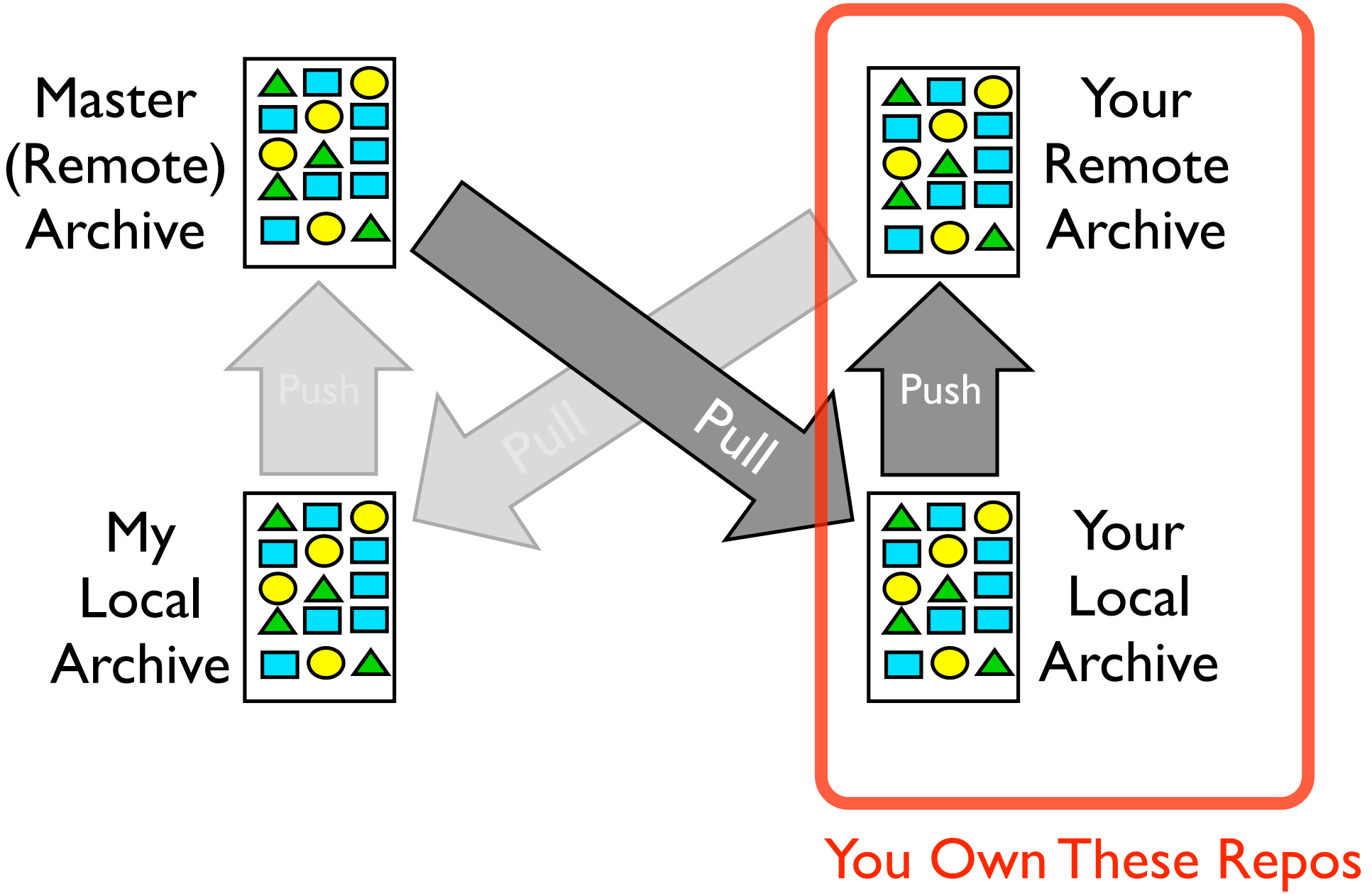


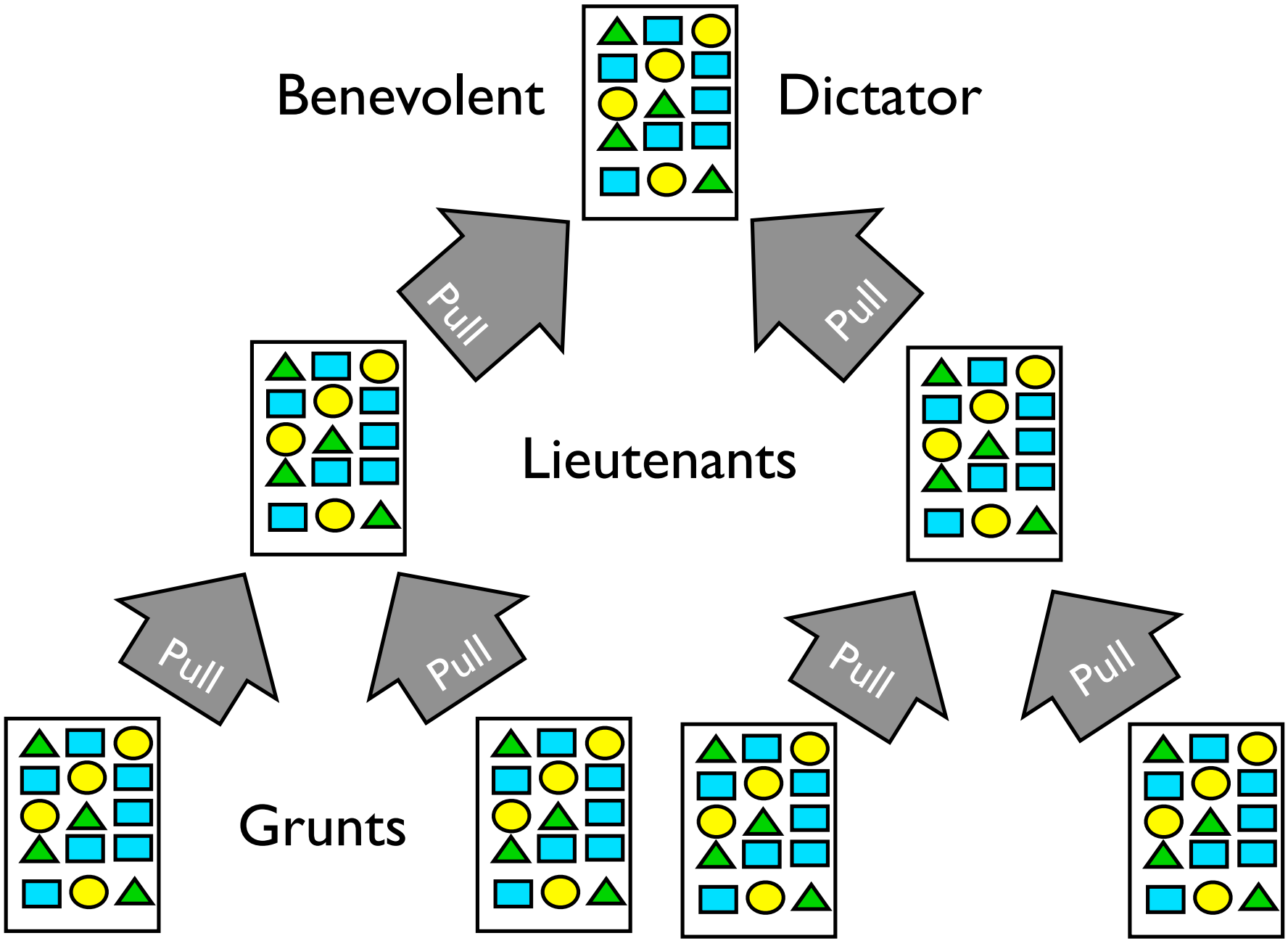

Master
(Remote)
Archive
Your
Remote
Archive
Push
Pull
Pull
Push
My
Local
Archive
Your
Local
Archive

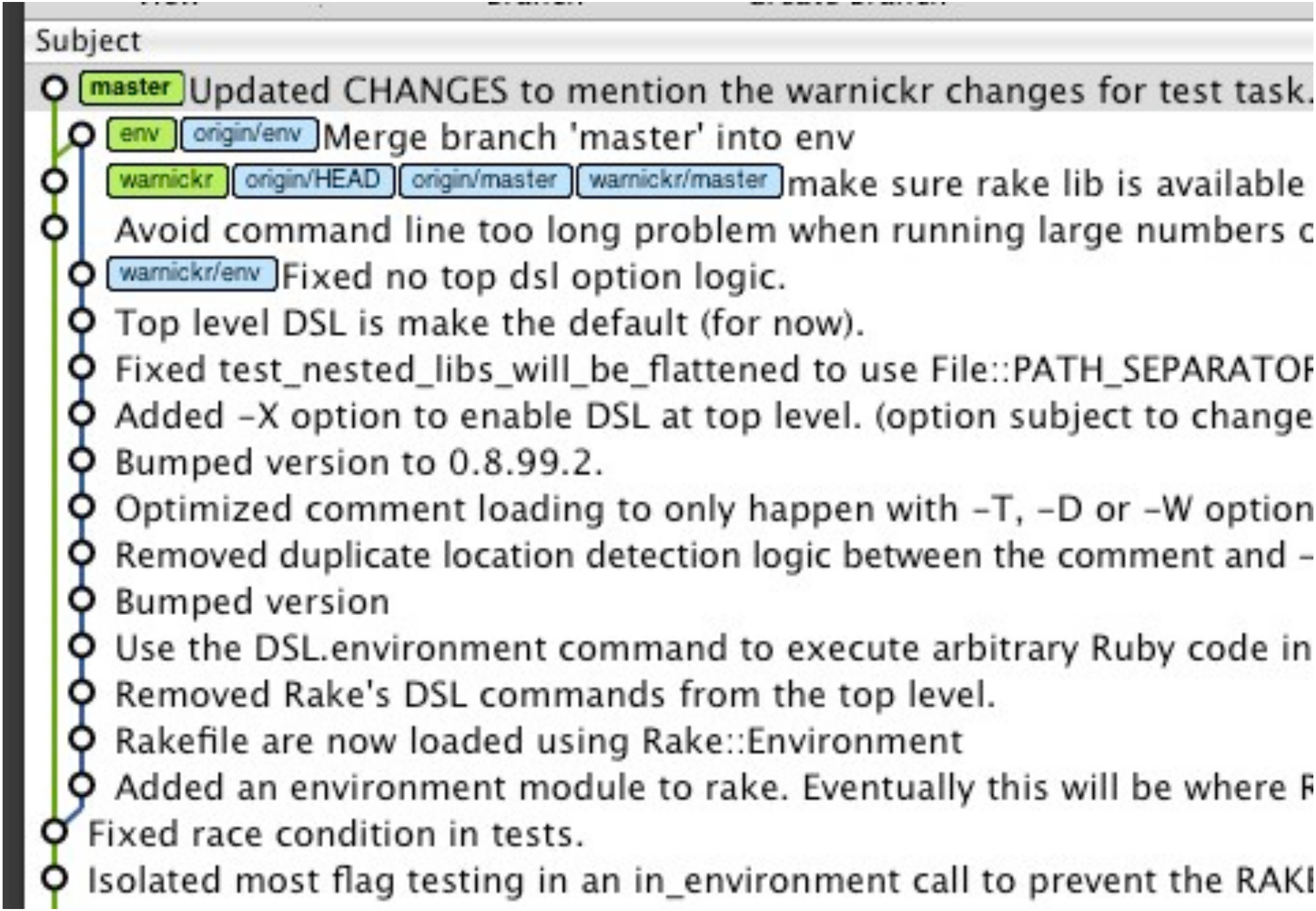
These Repos are Public

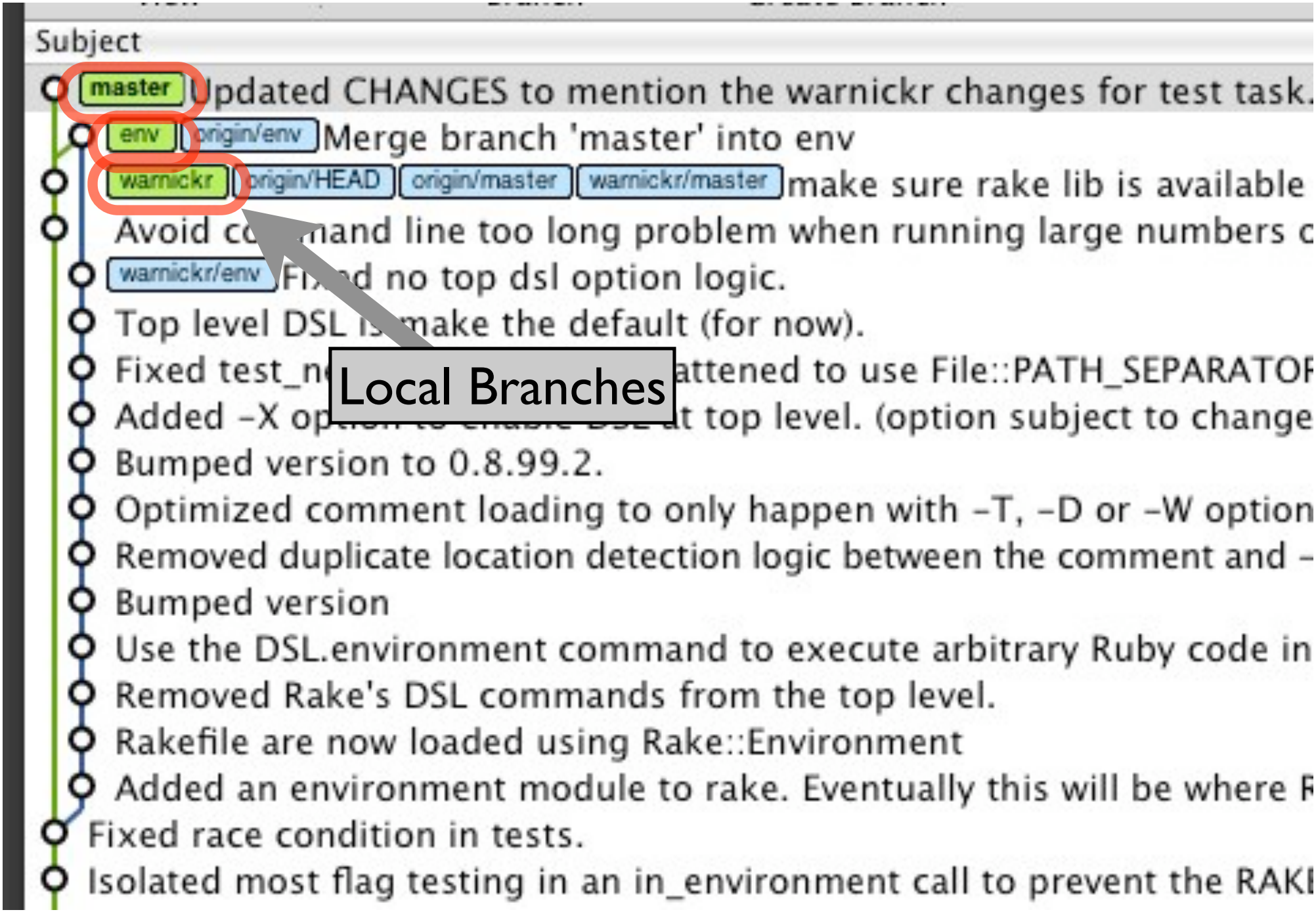


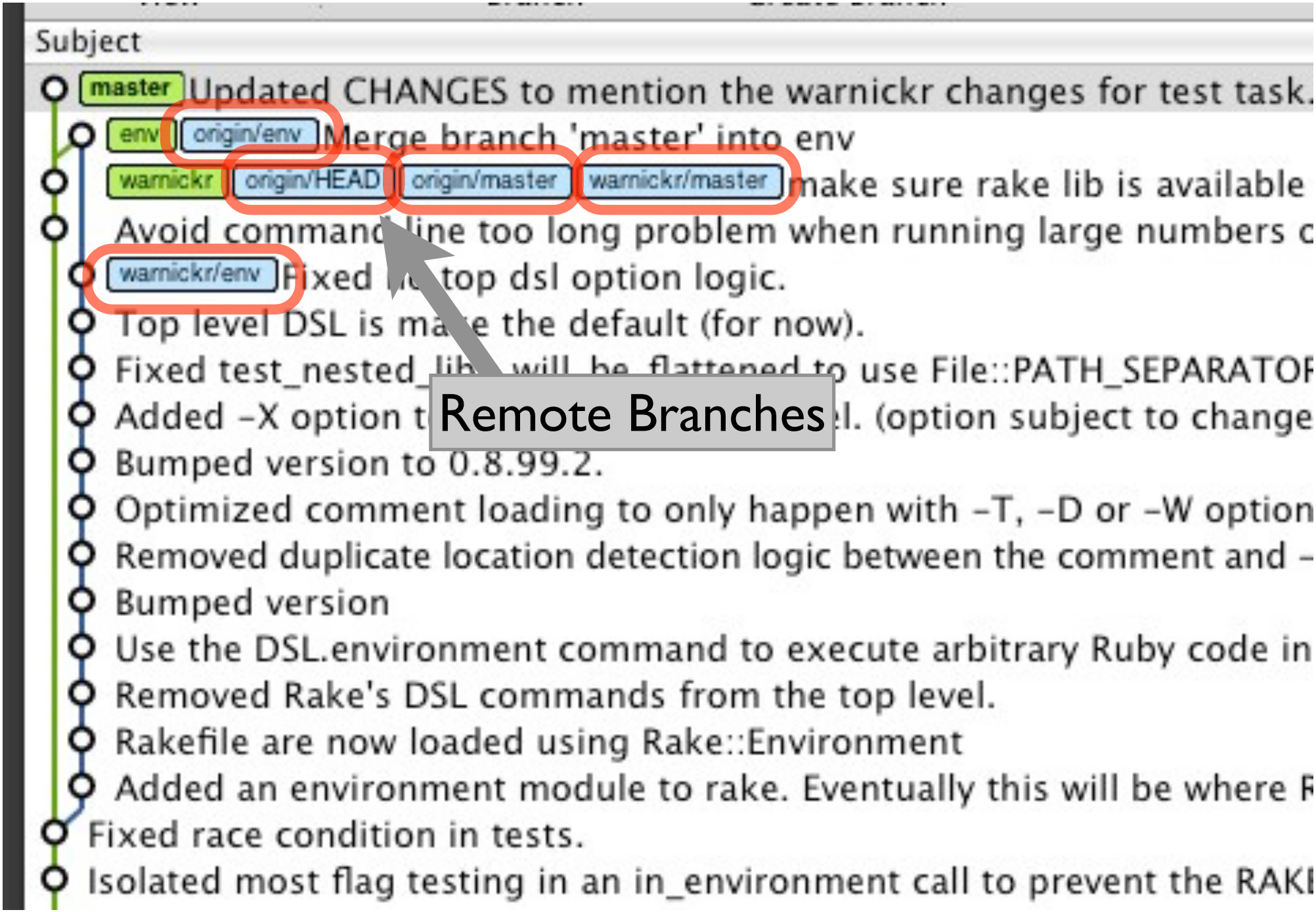

Master
(Remote)
Archive
Your
Remote
Archive
Push
Pull
Pull
Push
My
Local
Archive
Your
Local
Archive
I Own These Repos


Master
(Remote)
Archive
Your
Remote
Archive
Push
Pull
Pull
Push
My
Local
Archive
Your
Local
Archive
You Own These Repos


Benevolent
Dictator
Pull
Pull
Lieutenants
Pull
Pull
Pull
Pull
Grunts


Subject
master
Updated CHANGES to mention the warnickr changes for test task.
env
origin/env
Merge branch 'master' into env
warnickr
origin/HEAD
origin/master
warnickr/master
make sure rake lib is available
Avoid command line too long problem when running large numbers c
warnickr/env
Fixed no top dsl option logic.
Top level DSL is make the default (for now).
Fixed test_nested_libs_will_be_flattened to use File::PATH_SEPARATOR
Added -X option to enable DSL at top level. (option subject to change
Bumped version to 0.8.99.2.
Optimized comment loading to only happen with -T, -D or -W option
Removed duplicate location detection logic between the comment and -
Bumped version
Use the DSL.environment command to execute arbitrary Ruby code in
Removed Rake's DSL commands from the top level.
Rakefile are now loaded using Rake::Environment
Added an environment module to rake. Eventually this will be where R
Fixed race condition in tests.
Isolated most flag testing in an in_environment call to prevent the RAKE


Subject
master
Updated CHANGES to mention the warnickr changes for test task.
env
origin/env
Merge branch 'master' into env
warnickr
origin/HEAD
origin/master
warnickr/master
make sure rake lib is available
Avoid co
nand line too long problem when running large numbers c
warnickr/env
Fix
d no top dsl option logic.
Top level DSL is
make the default (for now).
Fixed test_n
attened to use File::PATH_SEPARATOR
Added -X op
t top level. (option subject to change
Bumped version to 0.8.99.2.
Optimized comment loading to only happen with -T, -D or -W option
Removed duplicate location detection logic between the comment and -
Bumped version
Use the DSL.environment command to execute arbitrary Ruby code in
Removed Rake's DSL commands from the top level.
Rakefile are now loaded using Rake::Environment
Added an environment module to rake. Eventually this will be where R
Fixed race condition in tests.
Isolated most flag testing in an in_environment call to prevent the RAKE
Local Branches


Subject
master Updated CHANGES to mention the warnickr changes for test task.
env origin/env Merge branch 'master' into env
warnickr origin/HEAD origin/master warnickr/master make sure rake lib is available
Avoid command line too long problem when running large numbers c
warnickr/env Fixed top dsl option logic.
Top level DSL is made the default (for now).
Fixed test_nested_lib will be flattened to use File::PATH_SEPARATOR
Added -X option to el. (option subject to change
Bumped version to 0.8.99.2.
Optimized comment loading to only happen with -T, -D or -W option
Removed duplicate location detection logic between the comment and -
Bumped version
Use the DSL.environment command to execute arbitrary Ruby code in
Removed Rake's DSL commands from the top level.
Rakefile are now loaded using Rake::Environment
Added an environment module to rake. Eventually this will be where R
Fixed race condition in tests.
Isolated most flag testing in an in_environment call to prevent the RAKE
Remote Branches

Wrap Up

CSC Features

- Complete revision history
- Efficient Storage
  - very lightweight project archives
- Guaranteed integrity
- Flexible Sharing
- Cheap Branching
- Easy Tagging

CSC Features

- Complete revision history
- Efficient Storage
  - very lightweight project archives
- Guaranteed integrity
- Flexible Sharing
- **Cheap Branching**
- Easy Tagging

Sounds Like ...

csc

GIT

csc == git
snapshot == commit
project archive == repository
manifest == tree (or directory)

csc snap == git commit -a

csc checkout *name* == git checkout *name*

csc branch *name* == git checkout -b *name*

csc tag *tagname* == git tag *tagname*

Differences from Git

Differences

Csc

- We have only:
  - Working Directory
  - Repository (project archive)

Git

- Git Uses
  - Working Directory
  - Staging Area
  - Repository

Differences

Csc

- We have only:
  - Working Directory
  - Repository (project archive)
- File Based

Git

- Git Uses
  - Working Directory
  - Staging Area
  - Repository
- Content Based

Differences

Csc

- One level object file storage

Git

- Files stored in .git/objects
  - Uses a 2-level directory
  - SHA1 hash is the directory/file name

Differences

Csc

- One level object file storage
- Plain Files

Git

- Files stored in .git/objects
  - Uses a 2-level directory
  - SHA1 hash is the directory/file name
- Encoded/Compressed files

Differences

Csc

- Just the basic features

Git

- init/log/status/diff
- reset/rebase
- cherry-pick
- bisect
- ... lots of others

Demo

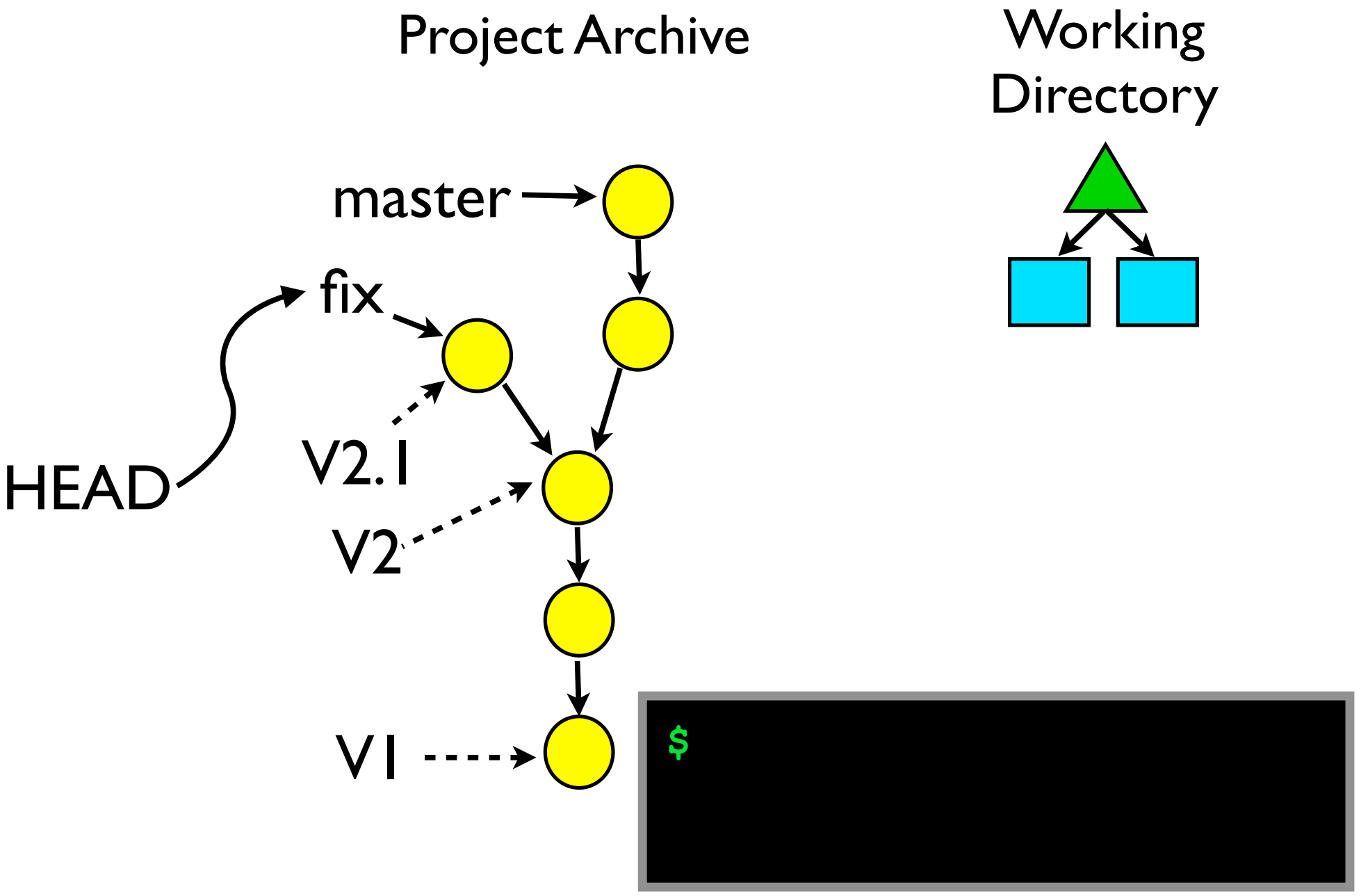
References

- Git from the Bottom Up (John Wiegley)
  - http://ftp.newartisans.com/pub/git.from.bottom.up.pdf
- Git Internals -- PeepCode PDF (Scott Chacon)
  - http://peepcode.com/products/git-internals-pdf
- Any presentation by Scott Chacon
  - http://github.com/schacon/git-presentations

git://github.com/jimweirich/presentation_source_control.git

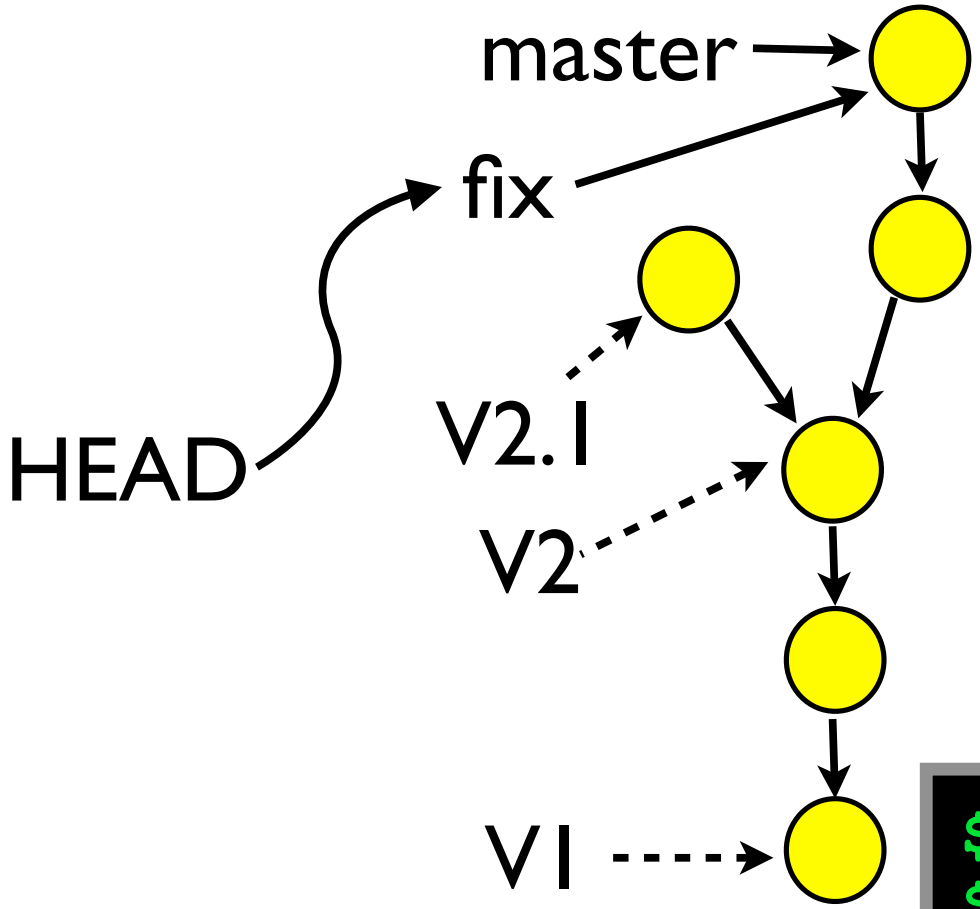
Questions?

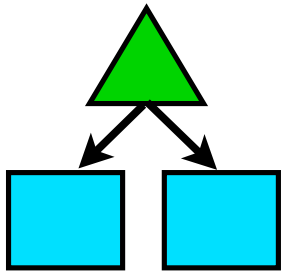
Appendix A -- Reset


Project Archive
Working Directory
master
fix
HEAD
V2.1
V2
V1
$

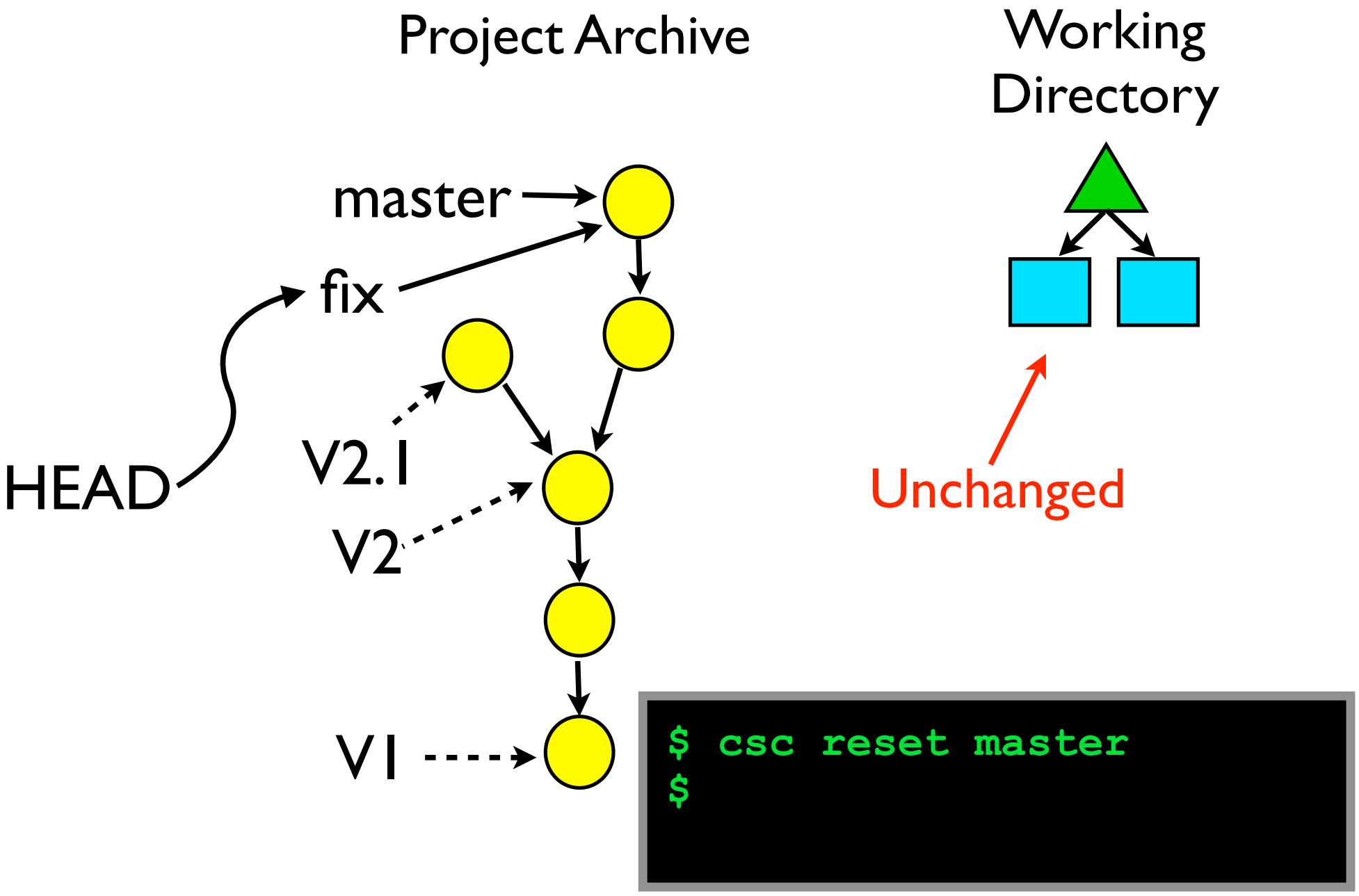
Project Archive

Working Directory

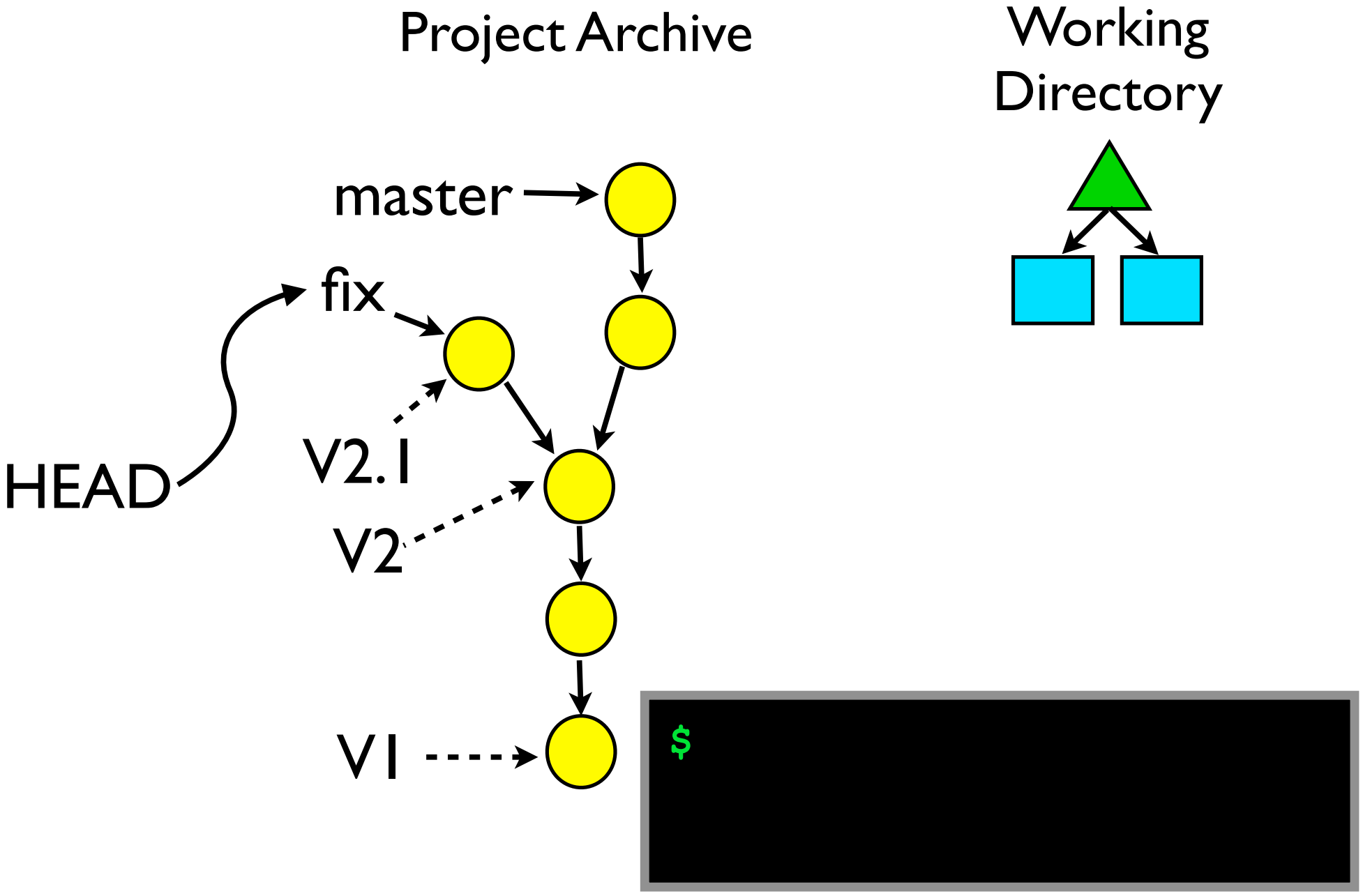




```
$ csc reset master
$
```

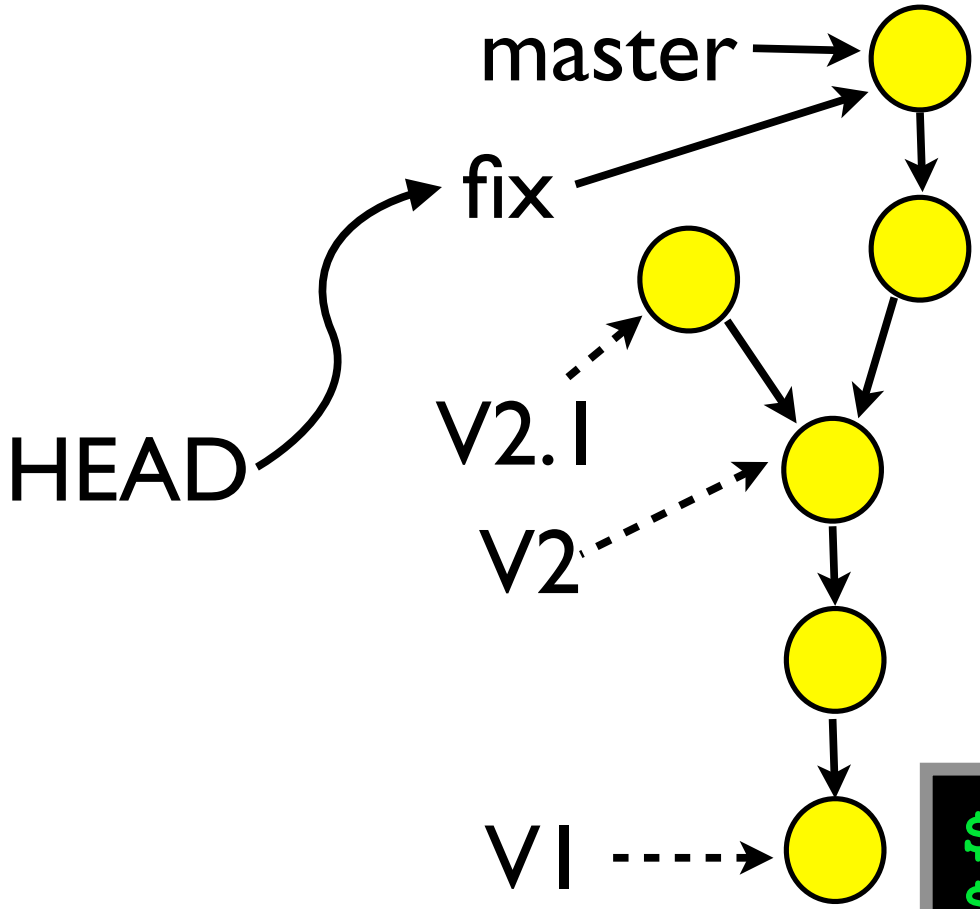

Project Archive
Working Directory
master
fix
HEAD
V2.1
V2
V1
Unchanged
$ csc reset master
$

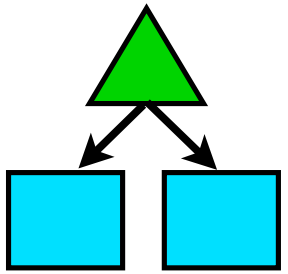
Now the --hard way


Project Archive
Working Directory
master
fix
HEAD
V2.1
V2
V1
$

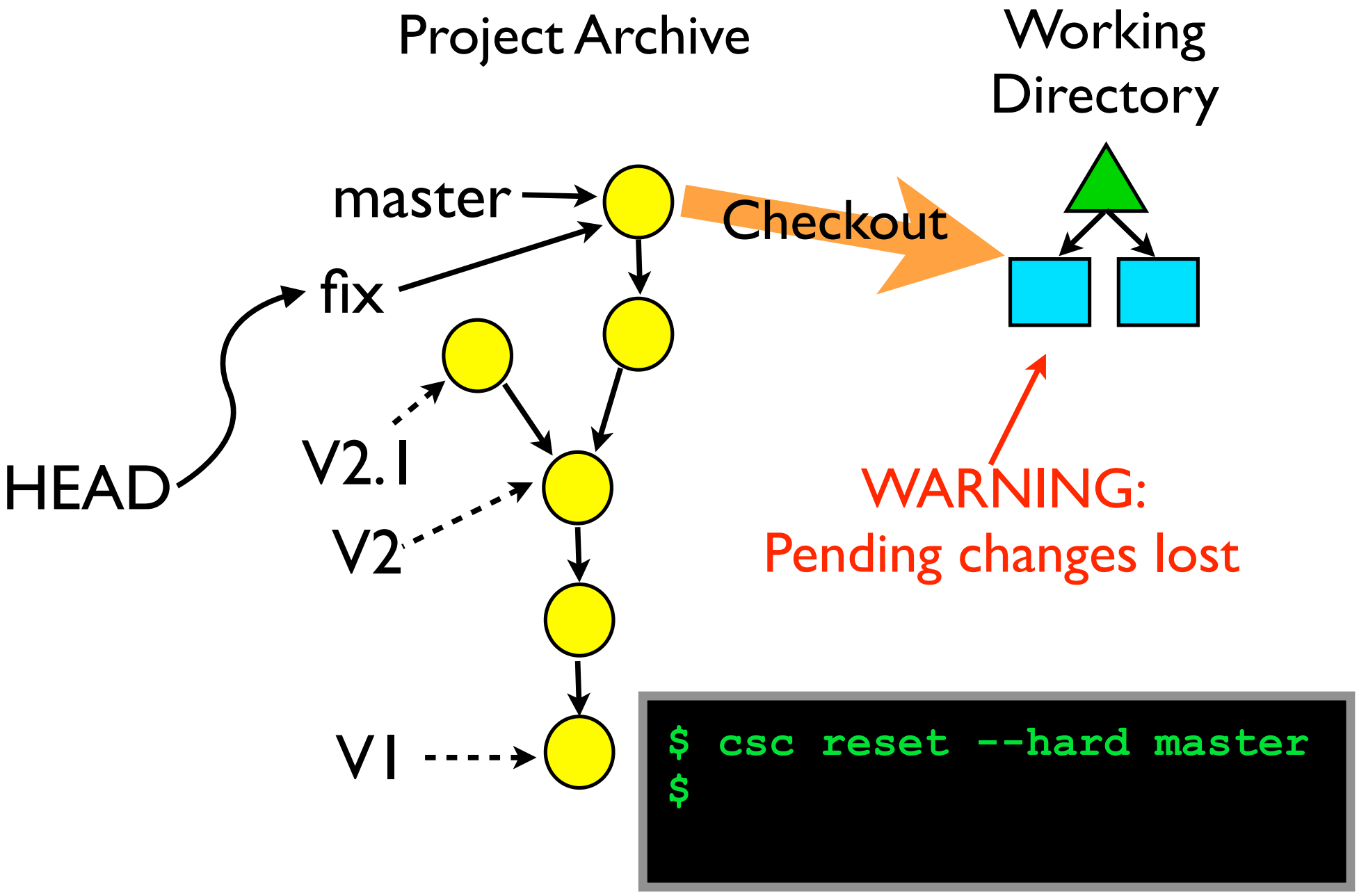
Project Archive

Working Directory



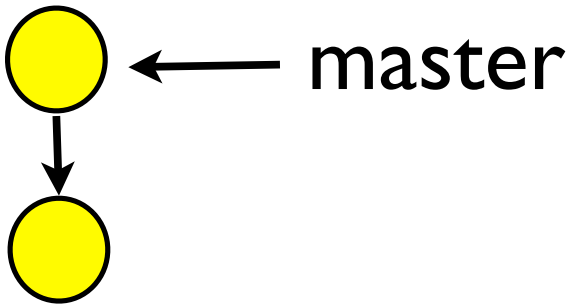


```
$ csc reset --hard master
$
```

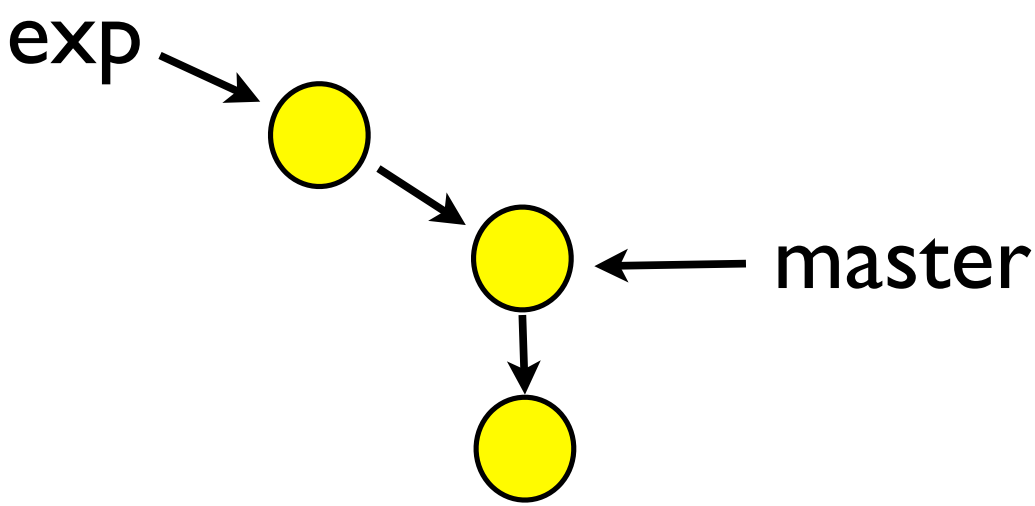

Project Archive
Working Directory
master
fix
HEAD
V2.1
V2
V1
Checkout
WARNING:
Pending changes lost
$ csc reset --hard master
$

Appendix B -- Rebase

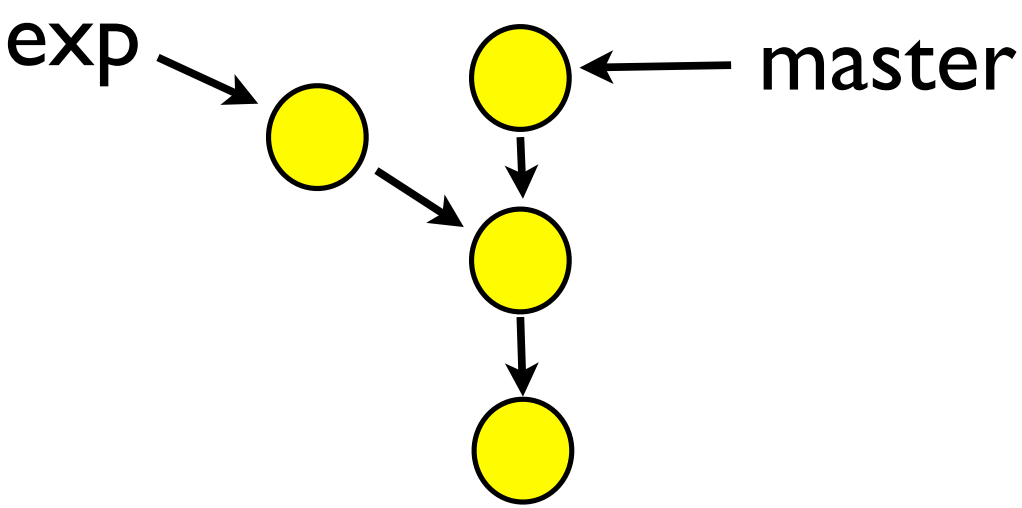
Rebase (setup)



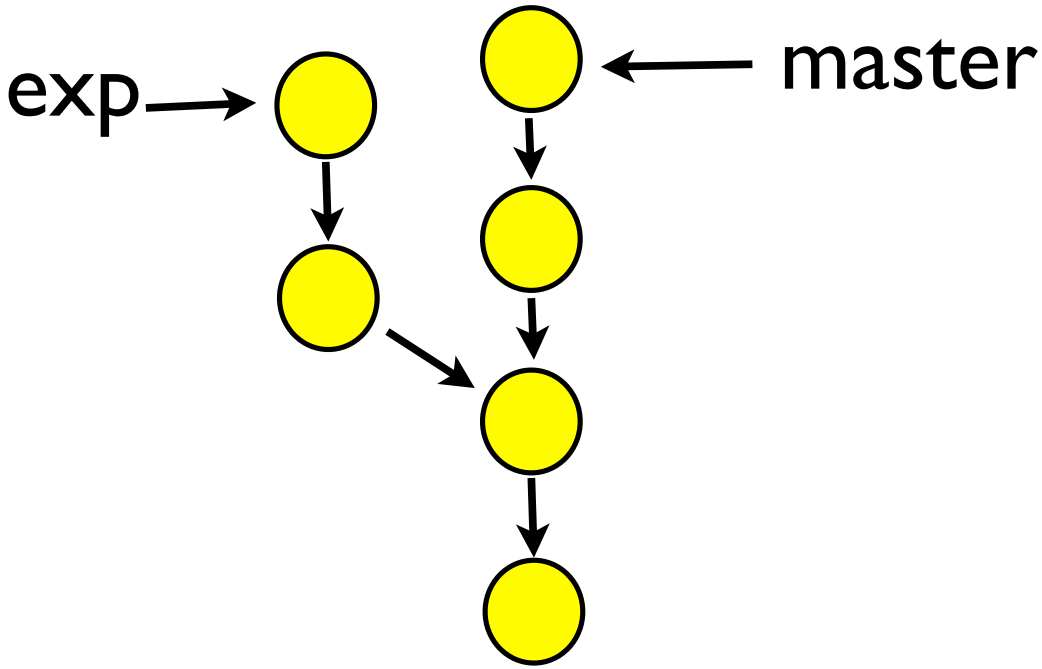
Rebase (setup)



Rebase (setup)

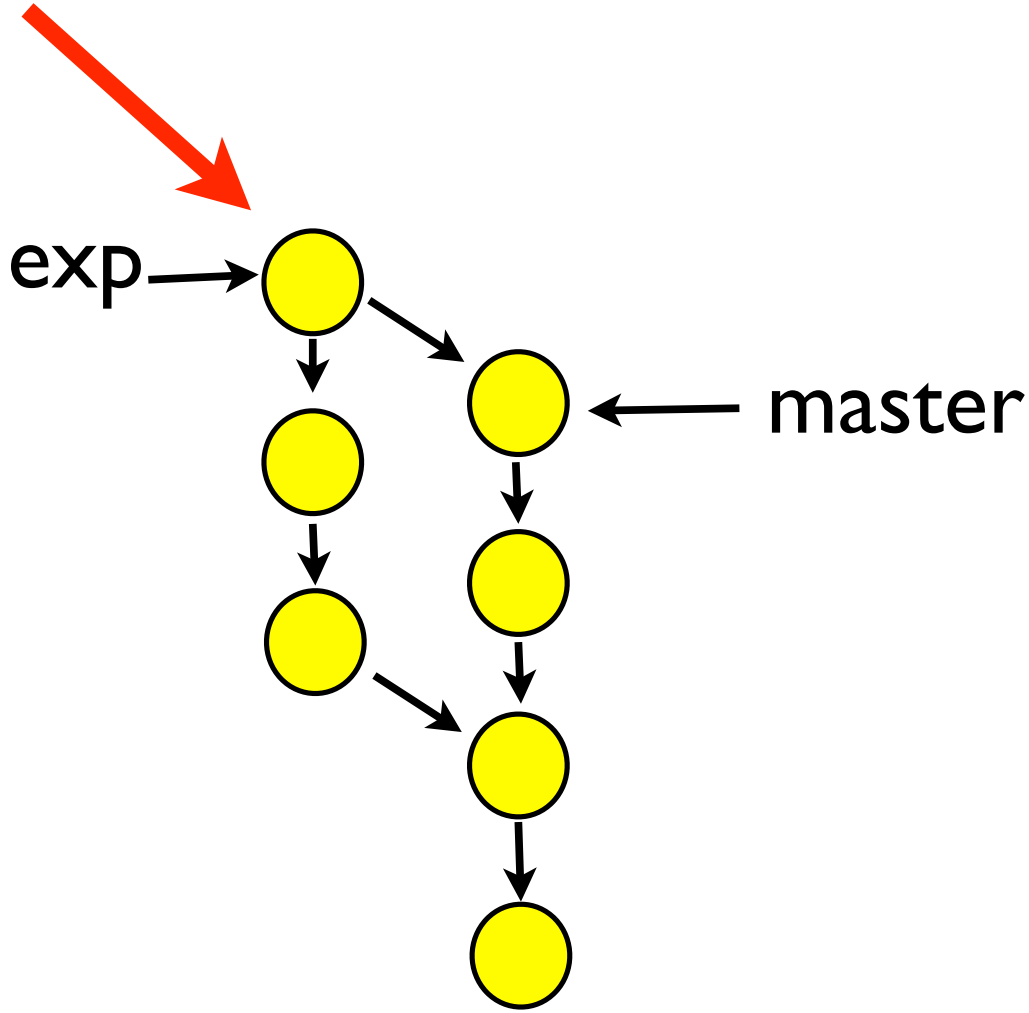


Rebase (setup)

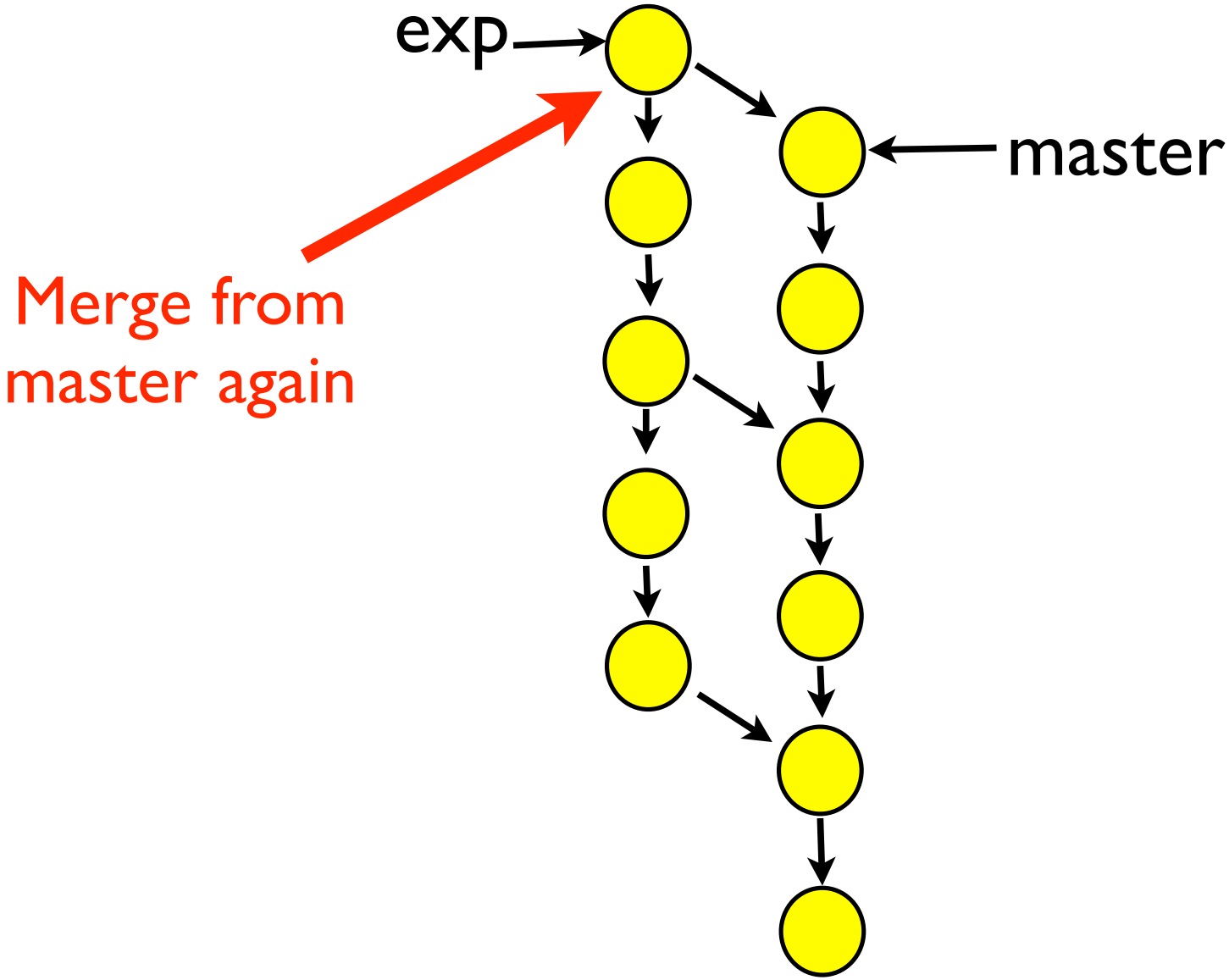


Rebase (merging option)

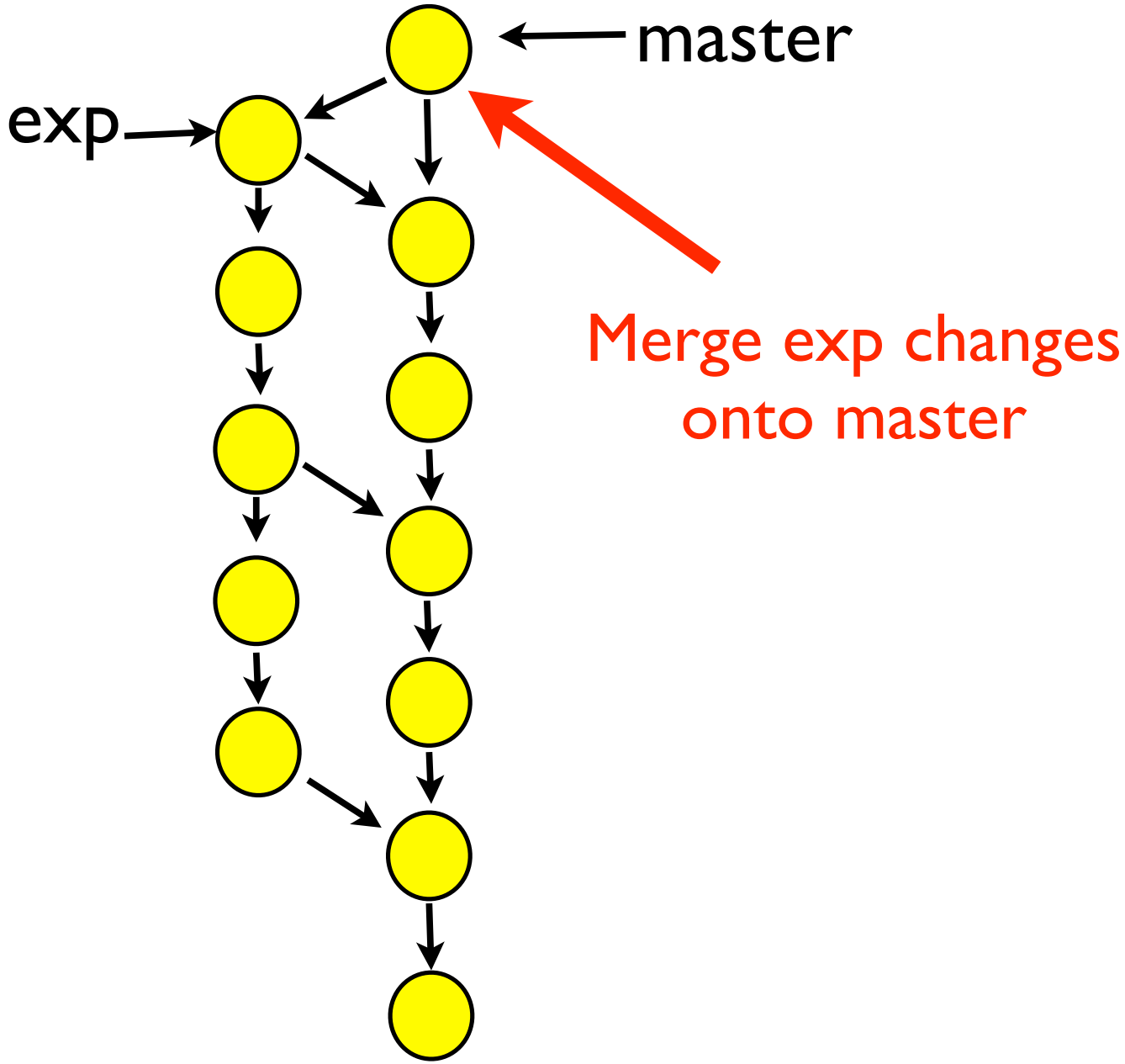
Merge changes on master to exp



Rebase (merging option)

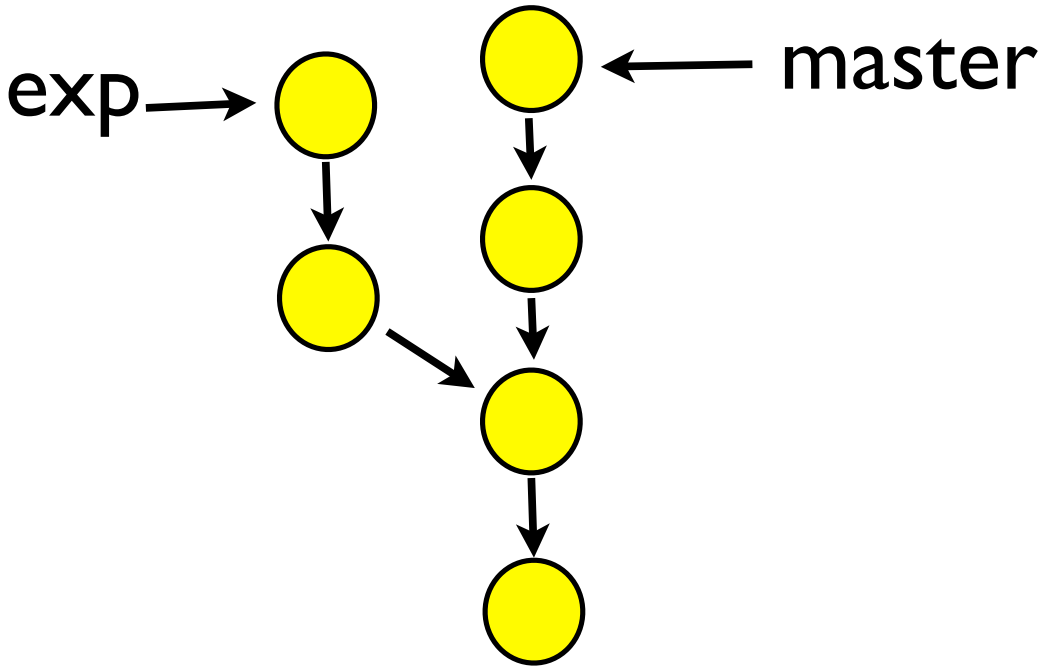


Rebase (merging option)

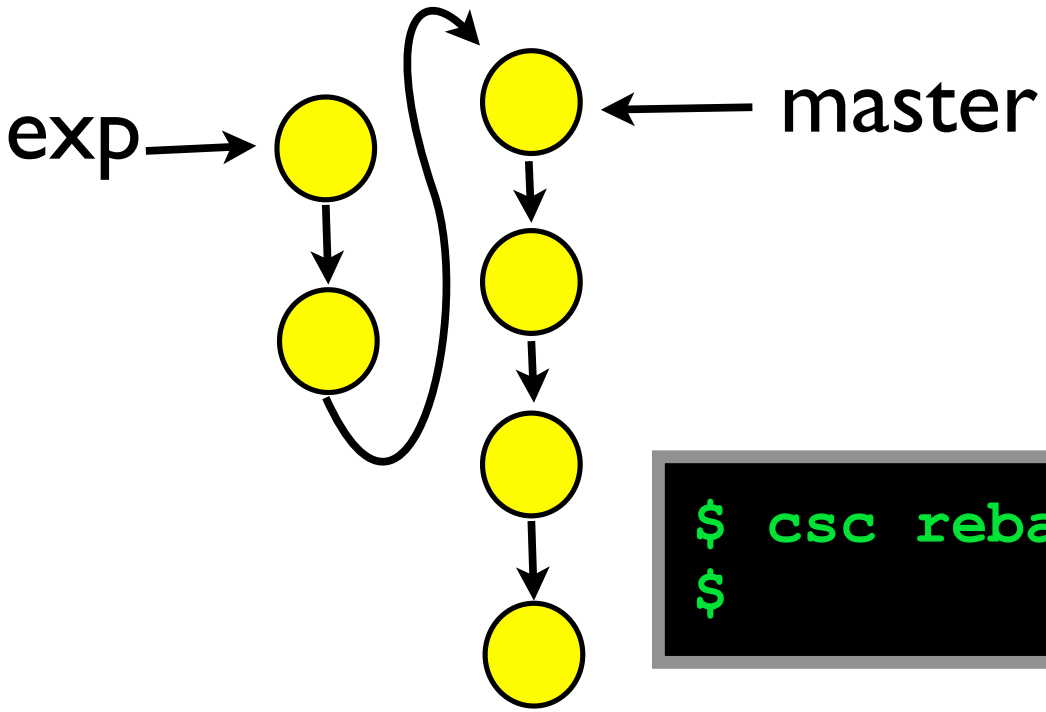


Rewind !

Rebase

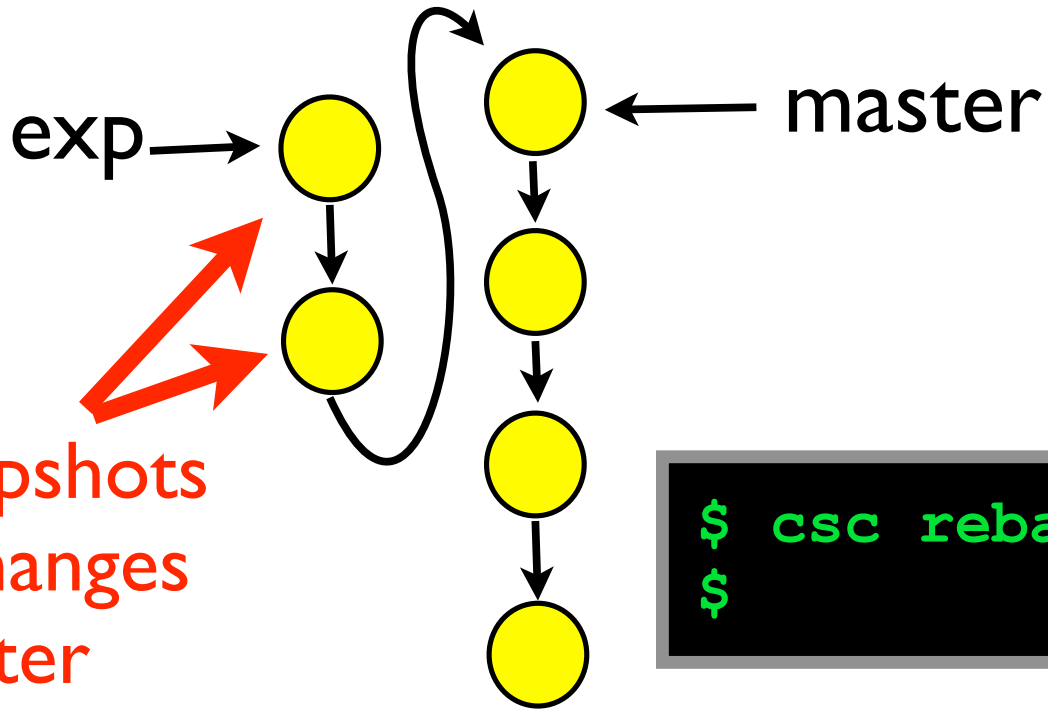


Rebase




```
$ csc rebase master
$
```

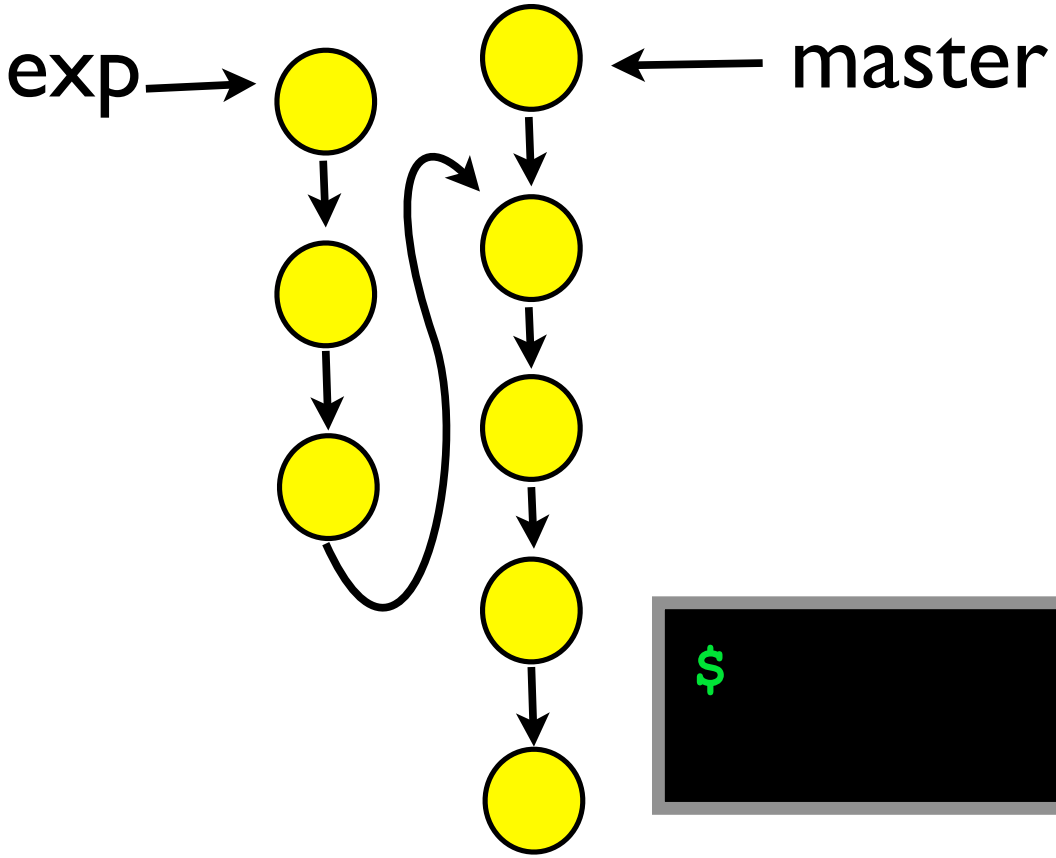
Rebase



Rewrites Snapshots to include changes from master

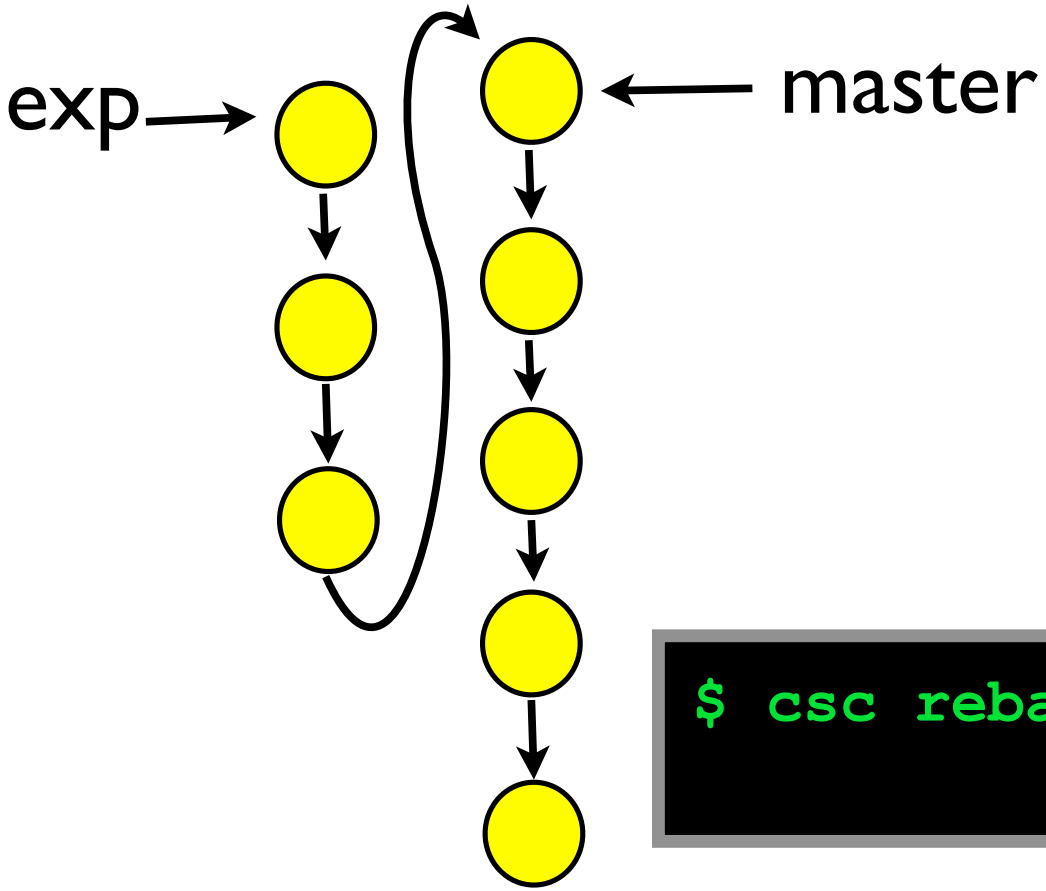
```
$ csc rebase master
$
```

Rebase



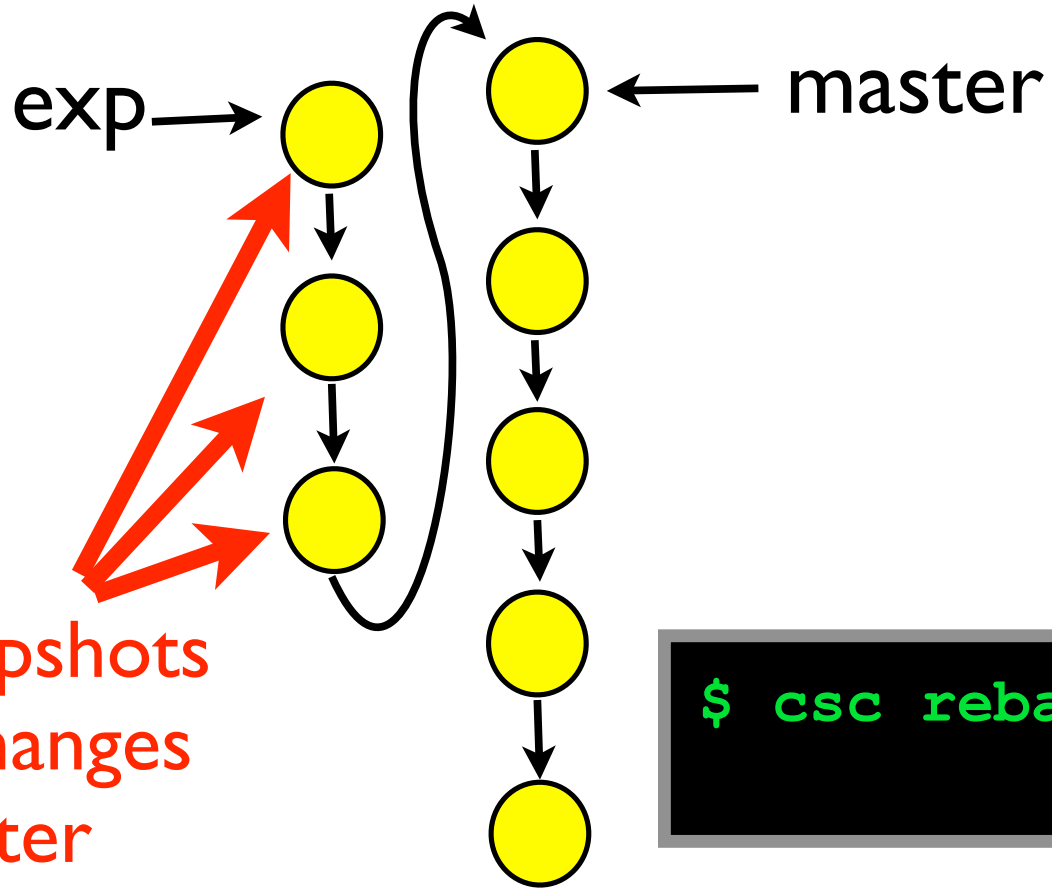

```
$
```

Rebase




```
$ csc rebase master
```

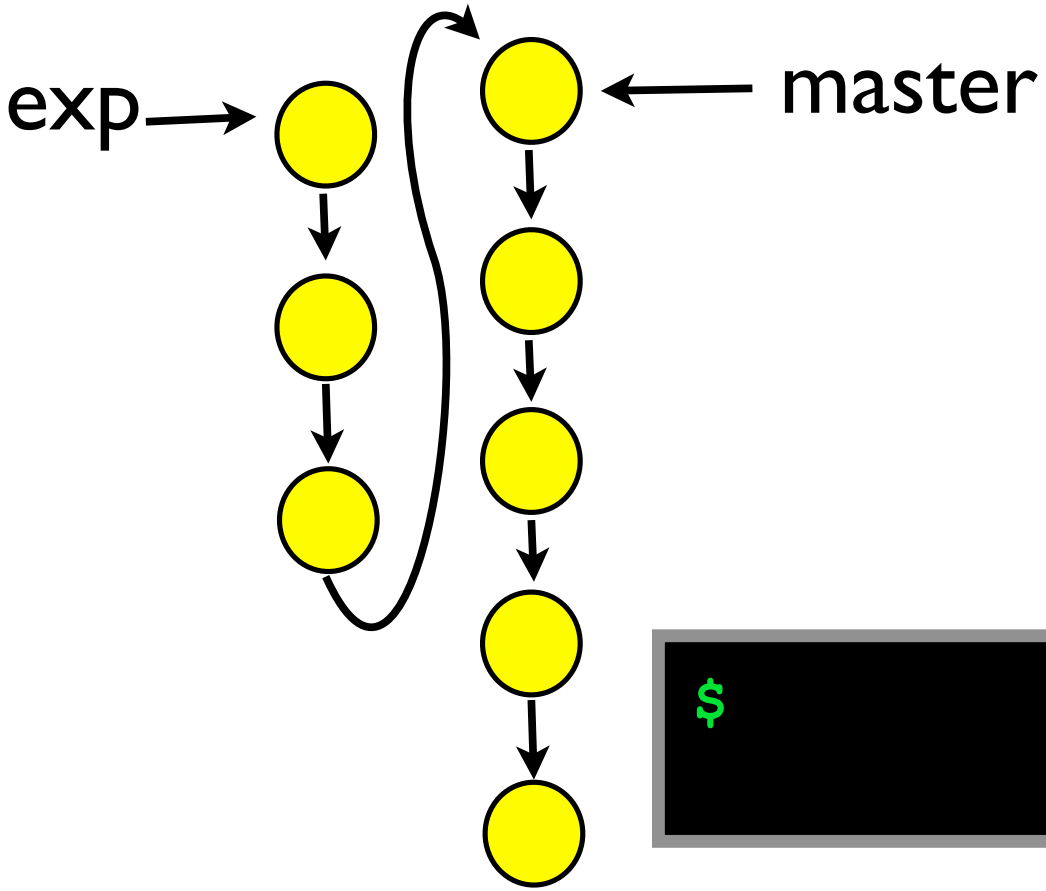
Rebase



Rewrites Snapshots to include changes from master

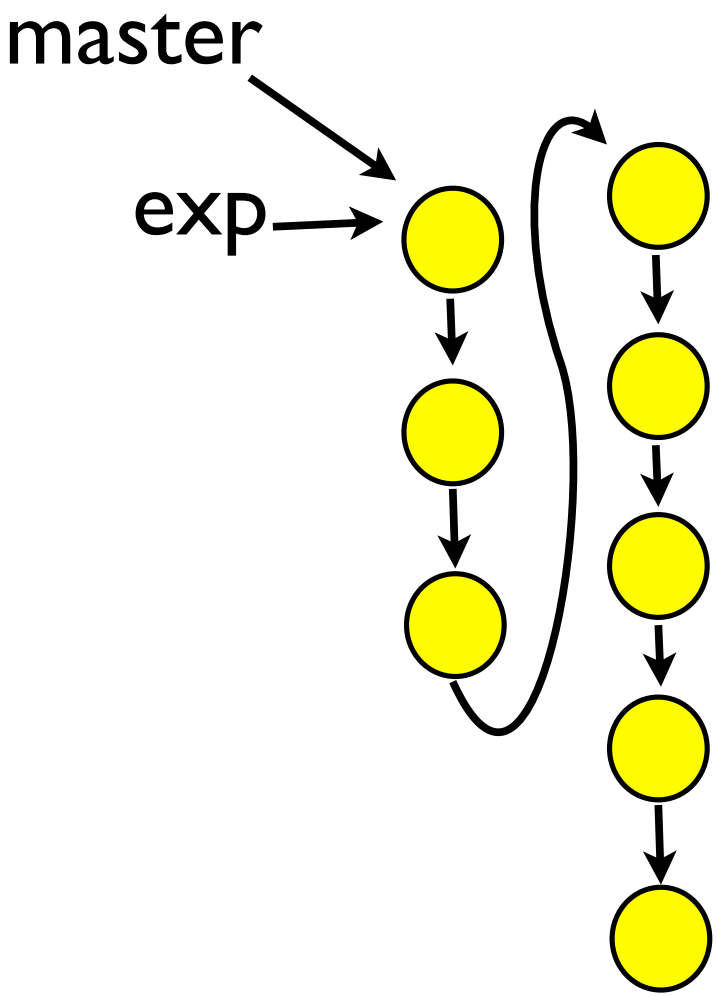
```
$ csc rebase master
```

Rebase




```
$
```

Rebase




```
$ csc checkout master
$ csc merge exp
```

WARNING:

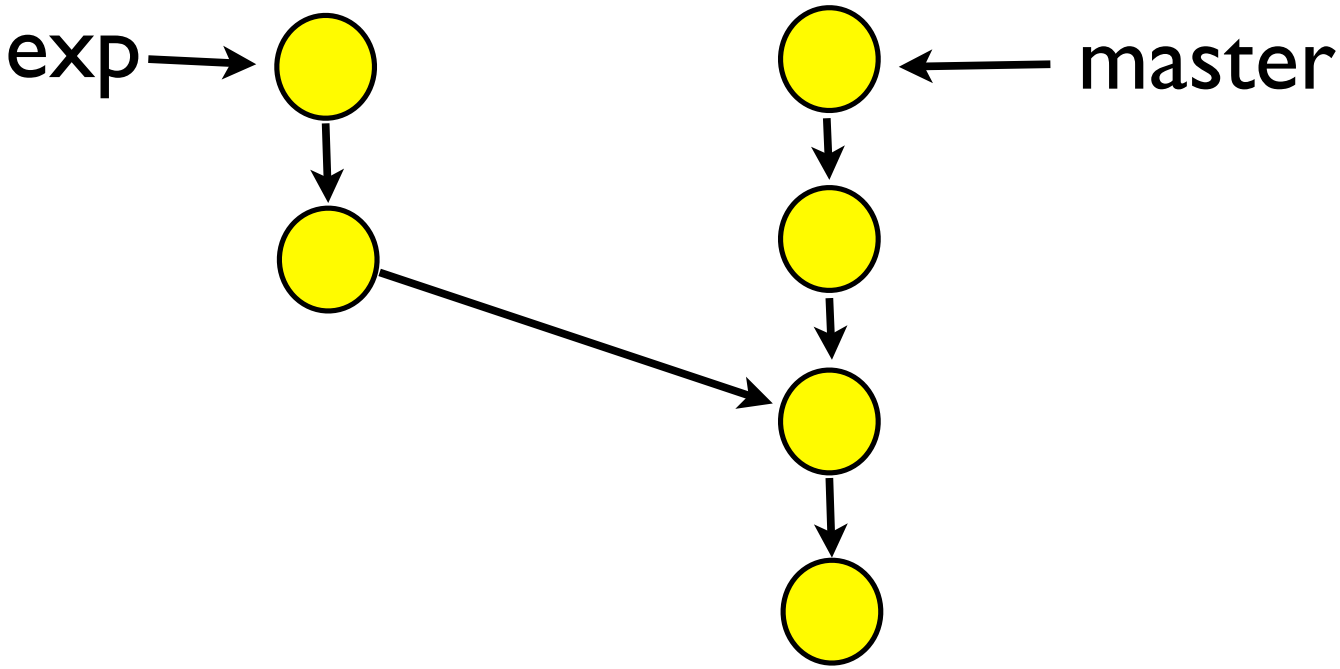
Conflicts may occur during rebase

WAIT!

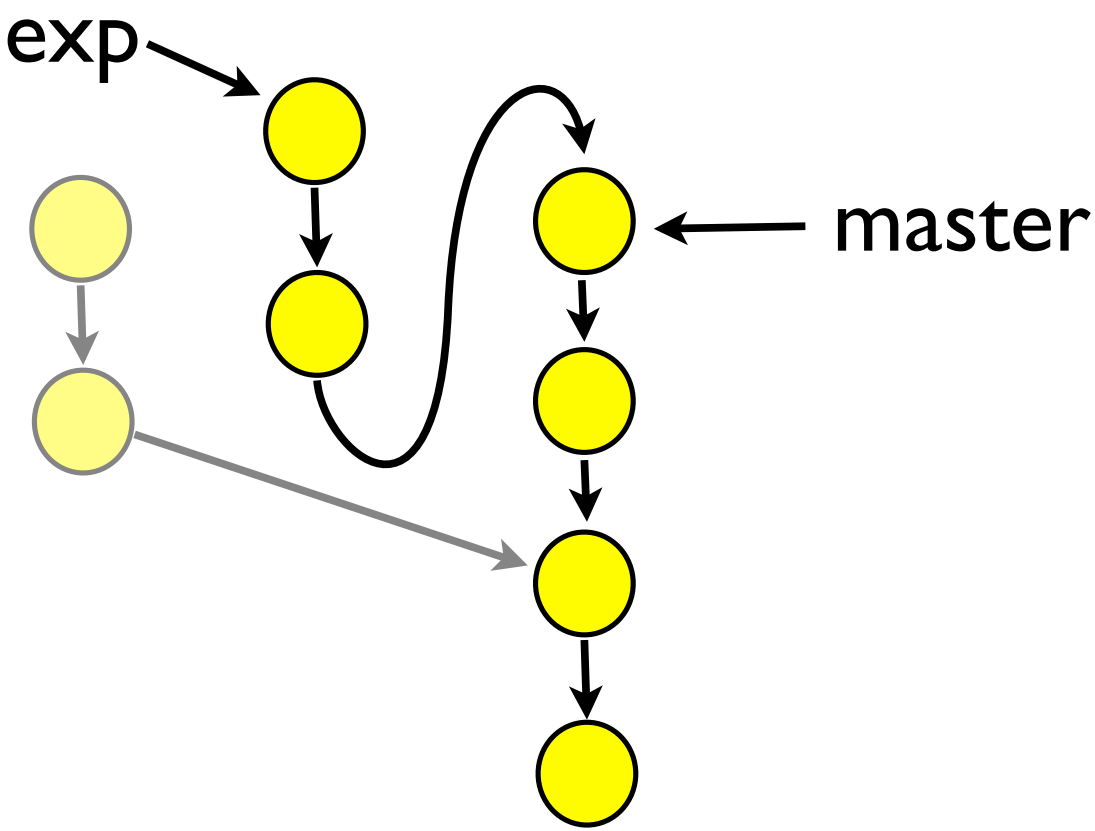
Rebase rewrites history?
Isn't that dangerous?

Yes and No

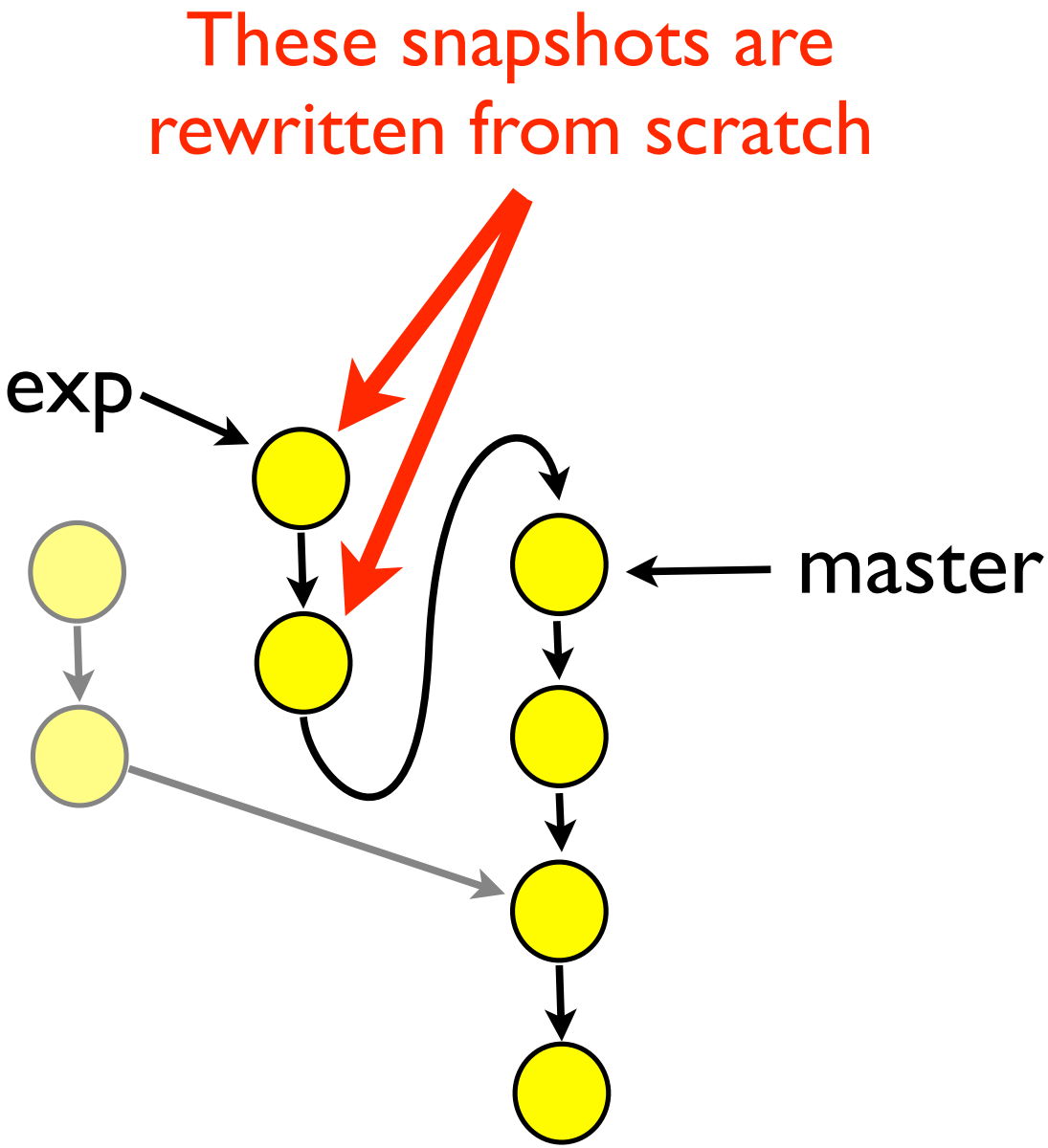
Rebase



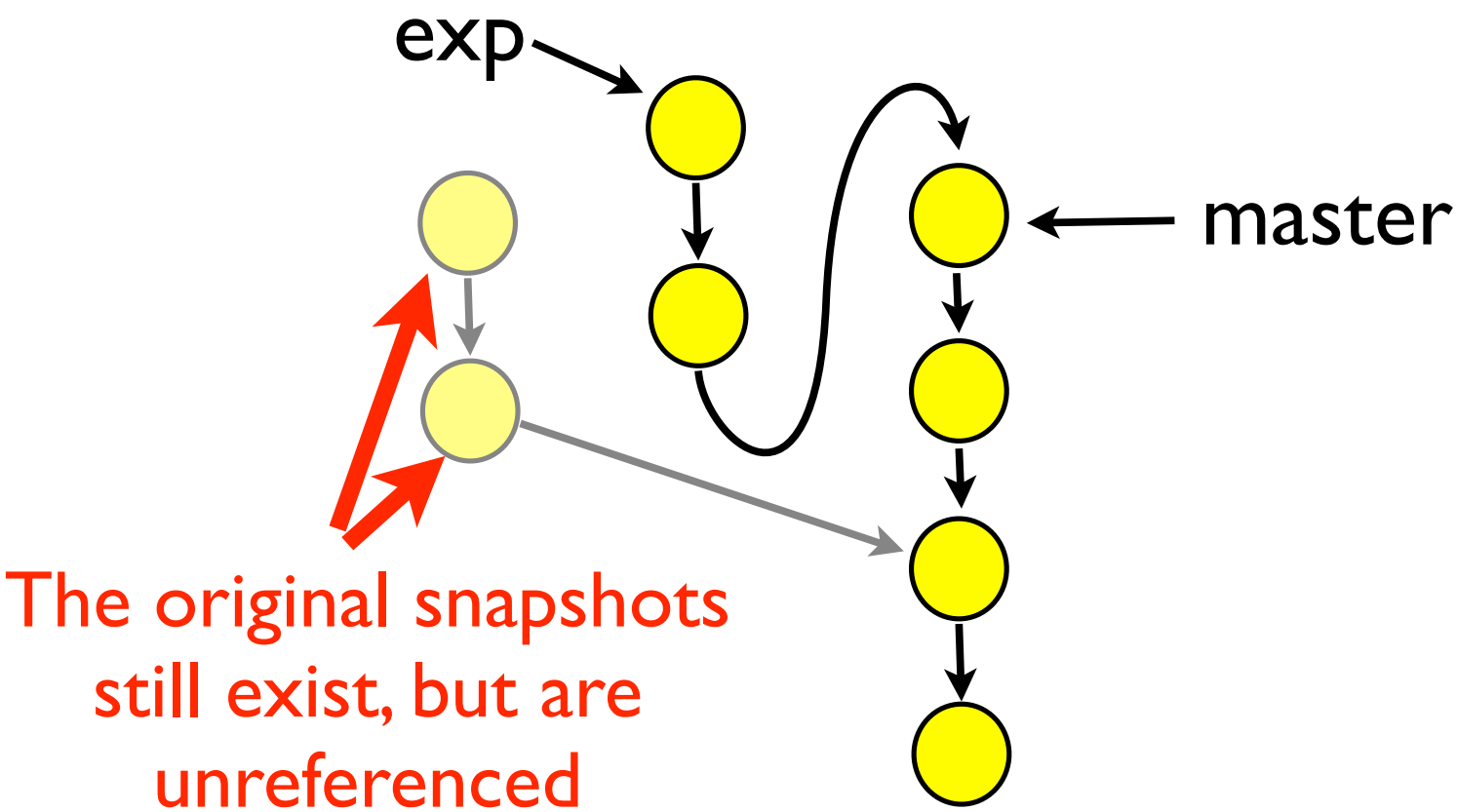
Rebase



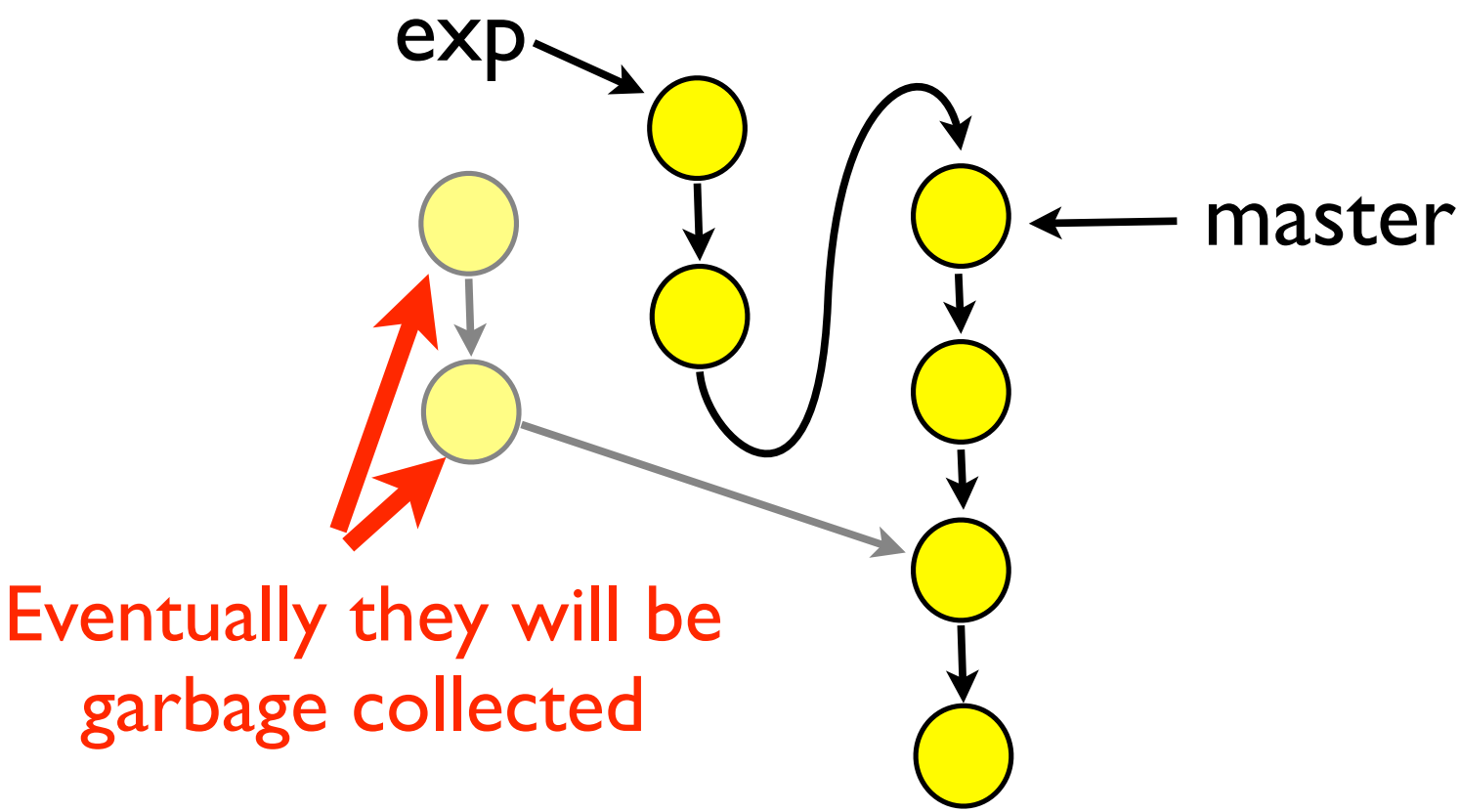
Rebase



Rebase


exp
master
The original snapshots still exist, but are unreferenced

Rebase



WARNING:

Do **not** rebase shared branches!