

Source Control for People Who Don't Like Source Control

Jim Weirich
Chief Scientist / EdgeCase
jim@edgecase.com
[@jimweirich](https://twitter.com/jimweirich)



Your Mission

Your Mission

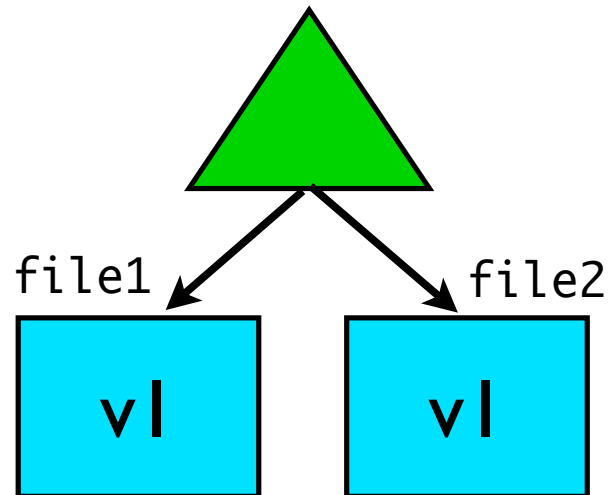
Design and Build
A Source Code
Control System

Custom Source Control

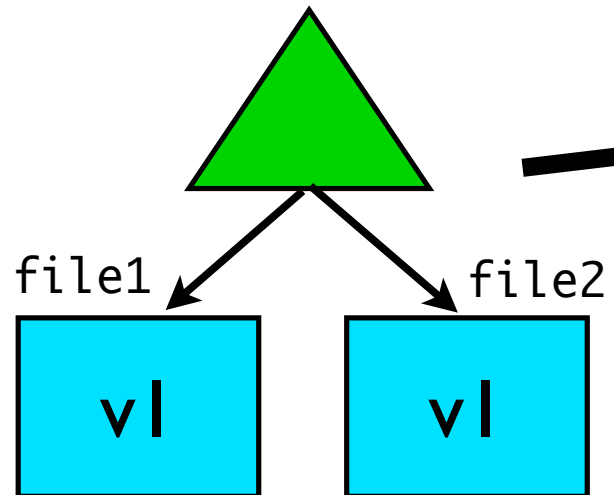
Custom Source Control

In the *Old Days* ...

Working Directory

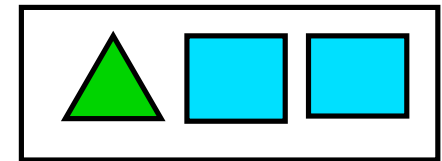


Working
Directory

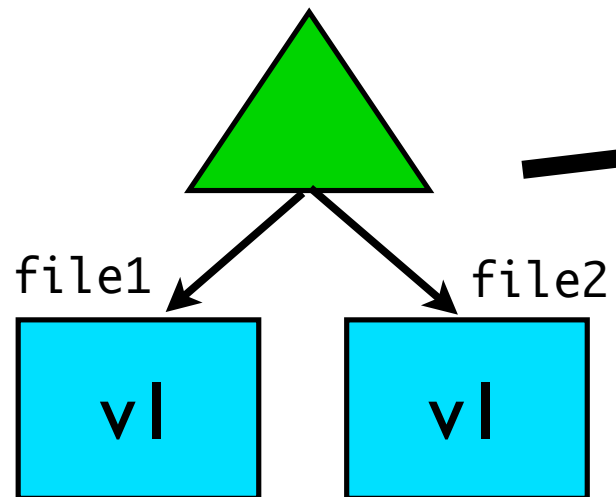


001.tar

Project Archive

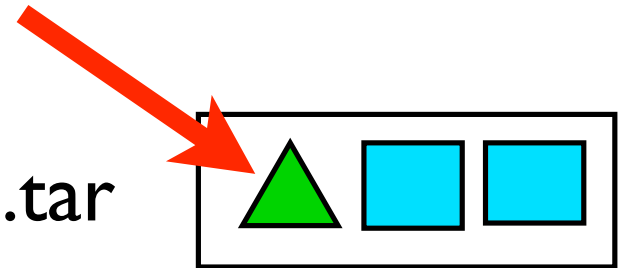


Working
Directory

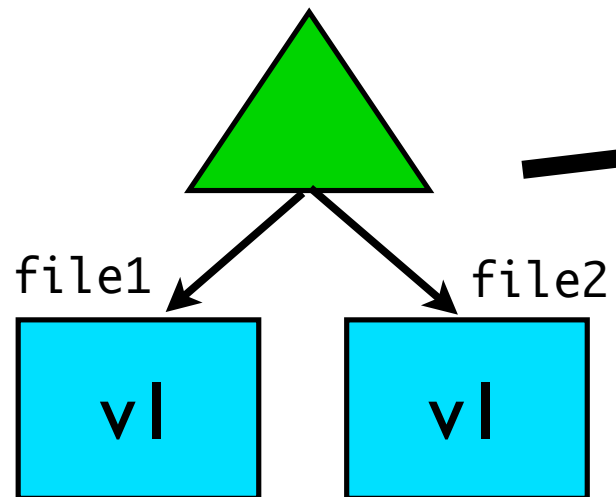


Manifest Project Archive

001.tar

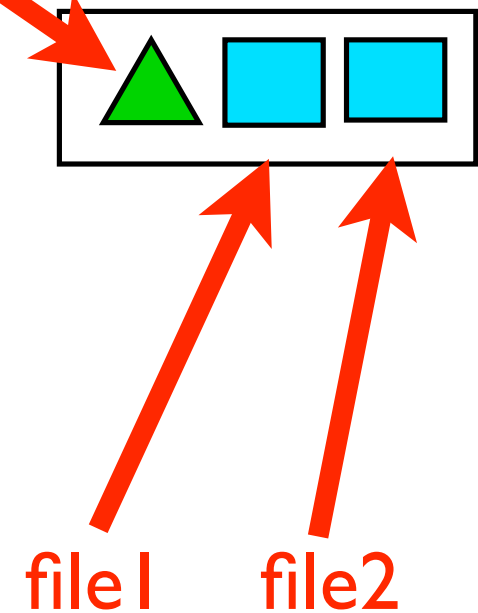


Working
Directory

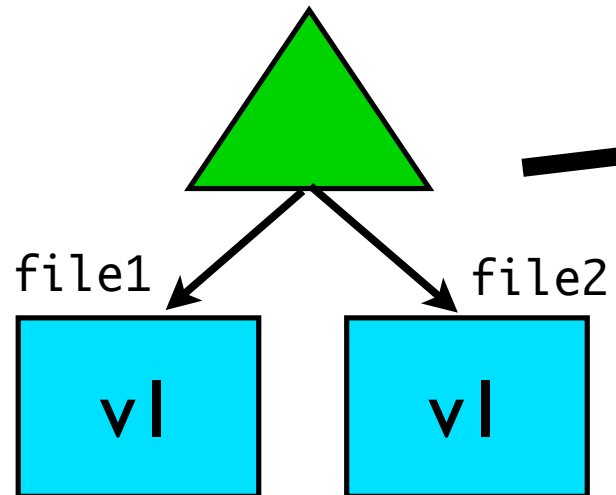


Manifest Project Archive

001.tar

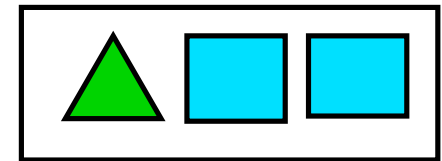


Working
Directory



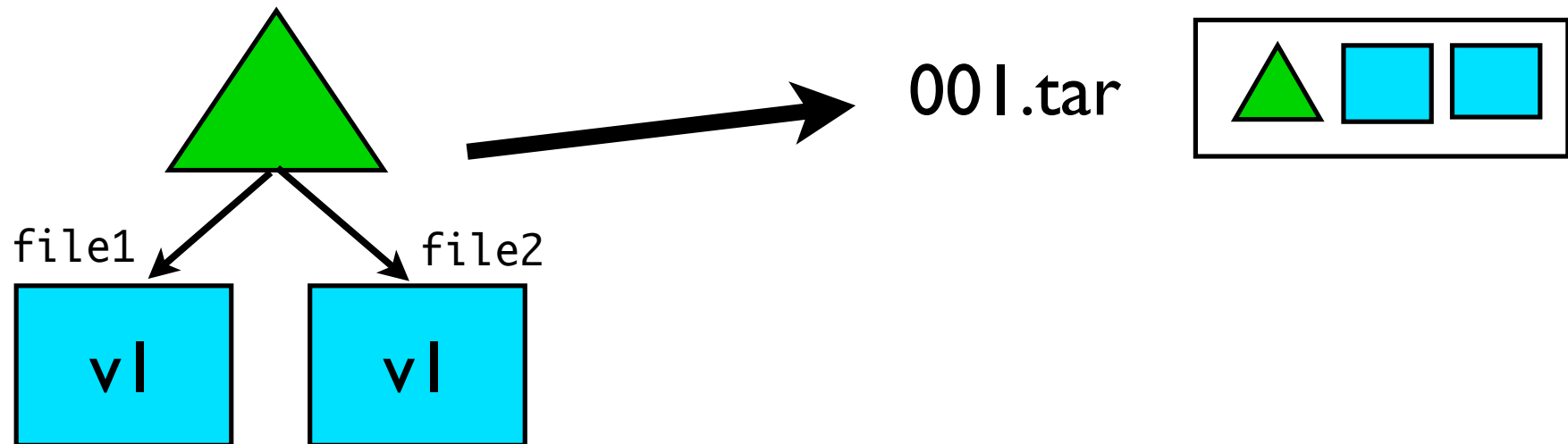
001.tar

Project Archive



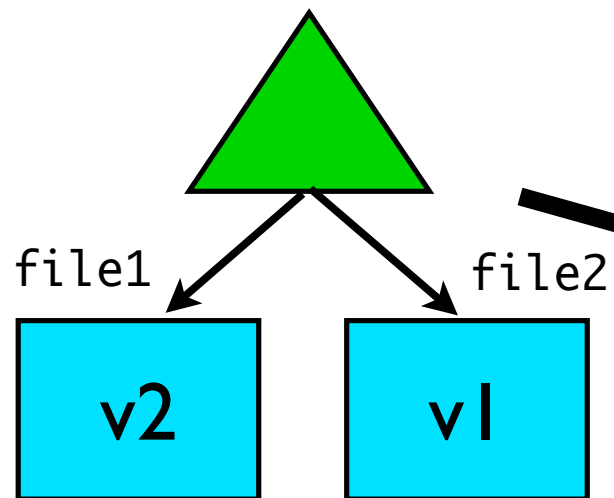
Working
Directory

Project Archive



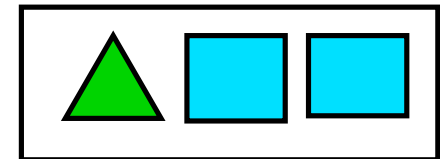
```
$ csc snap
```


Working Directory

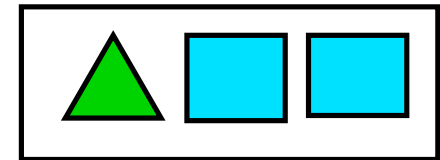


Project Archive

001.tar

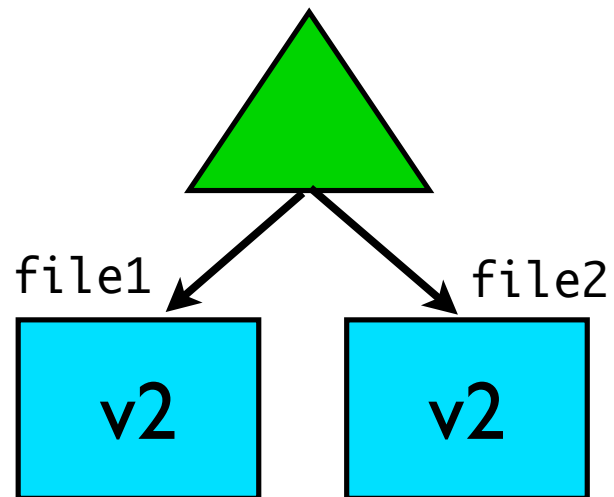


002.tar



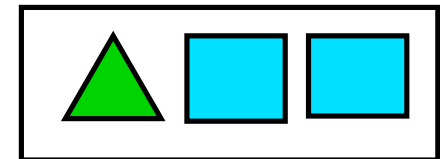
```
$ csc snap
```


Working Directory

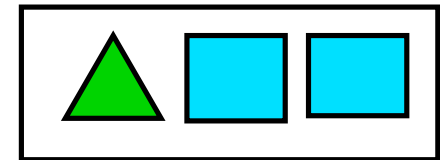


Project Archive

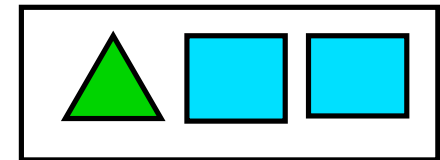
001.tar



002.tar

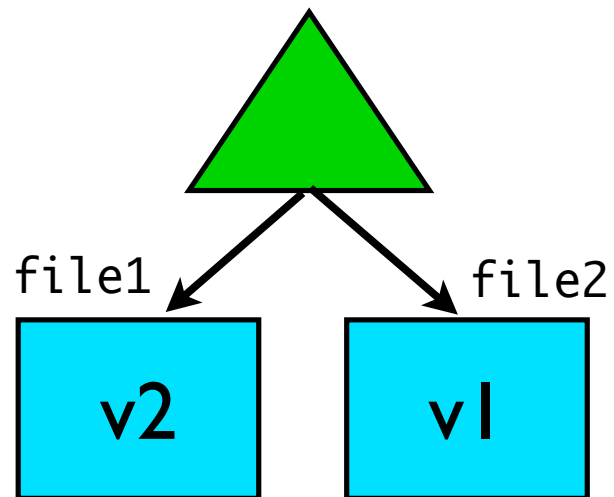


003.tar



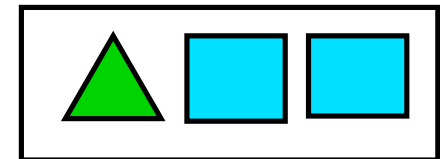
```
$ csc snap
```


Working Directory

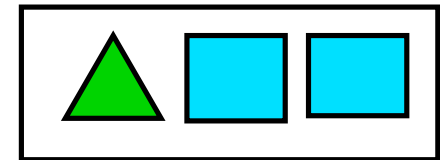


Project Archive

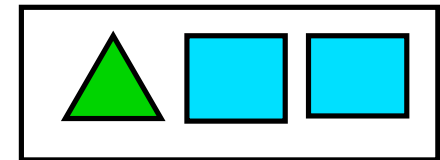
001.tar



002.tar



003.tar



```
$ csc checkout 002
```

Two Basic Commands

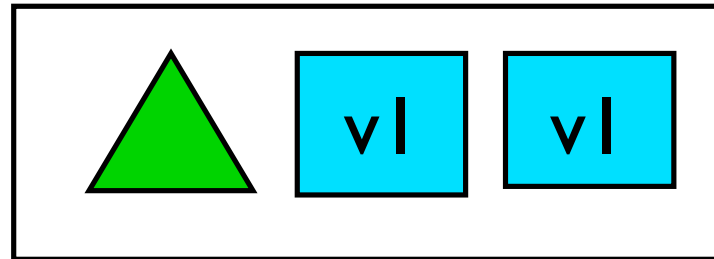
- Snap
- Checkout

Very Easy /
Very Simple

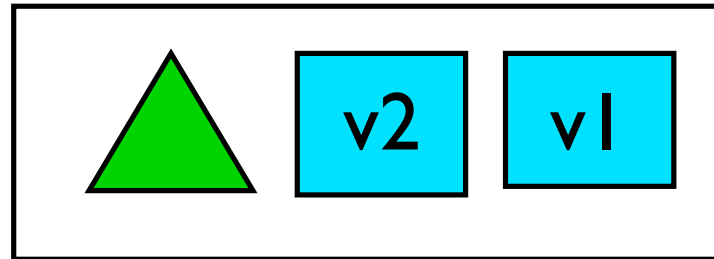
Rather Wasteful

manifest file1 file2

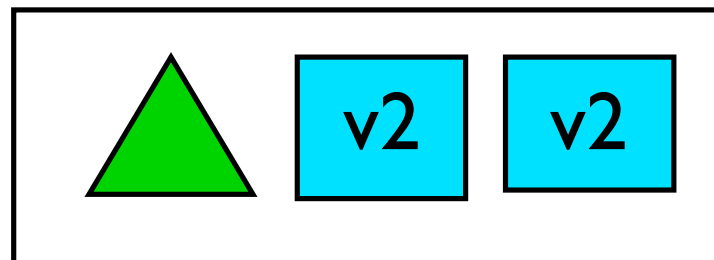
001.tar

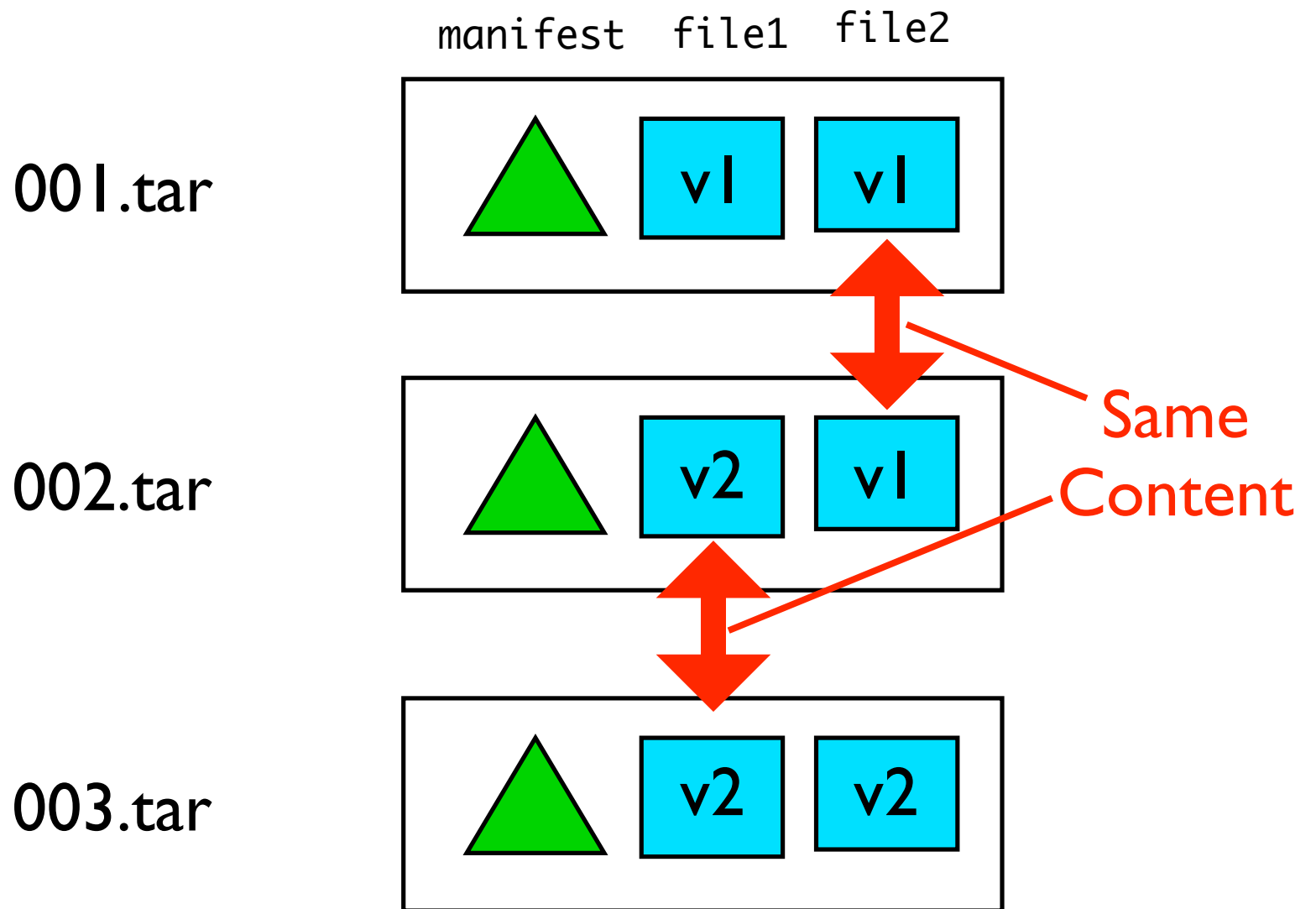


002.tar



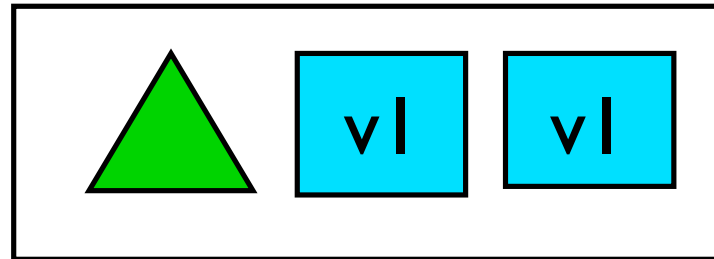
003.tar



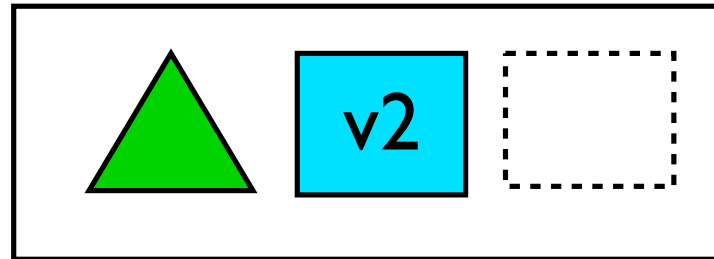


manifest file1 file2

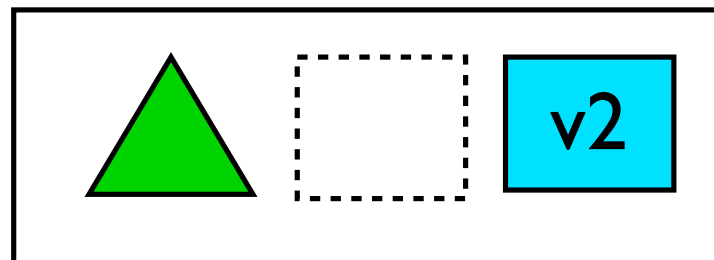
001.tar

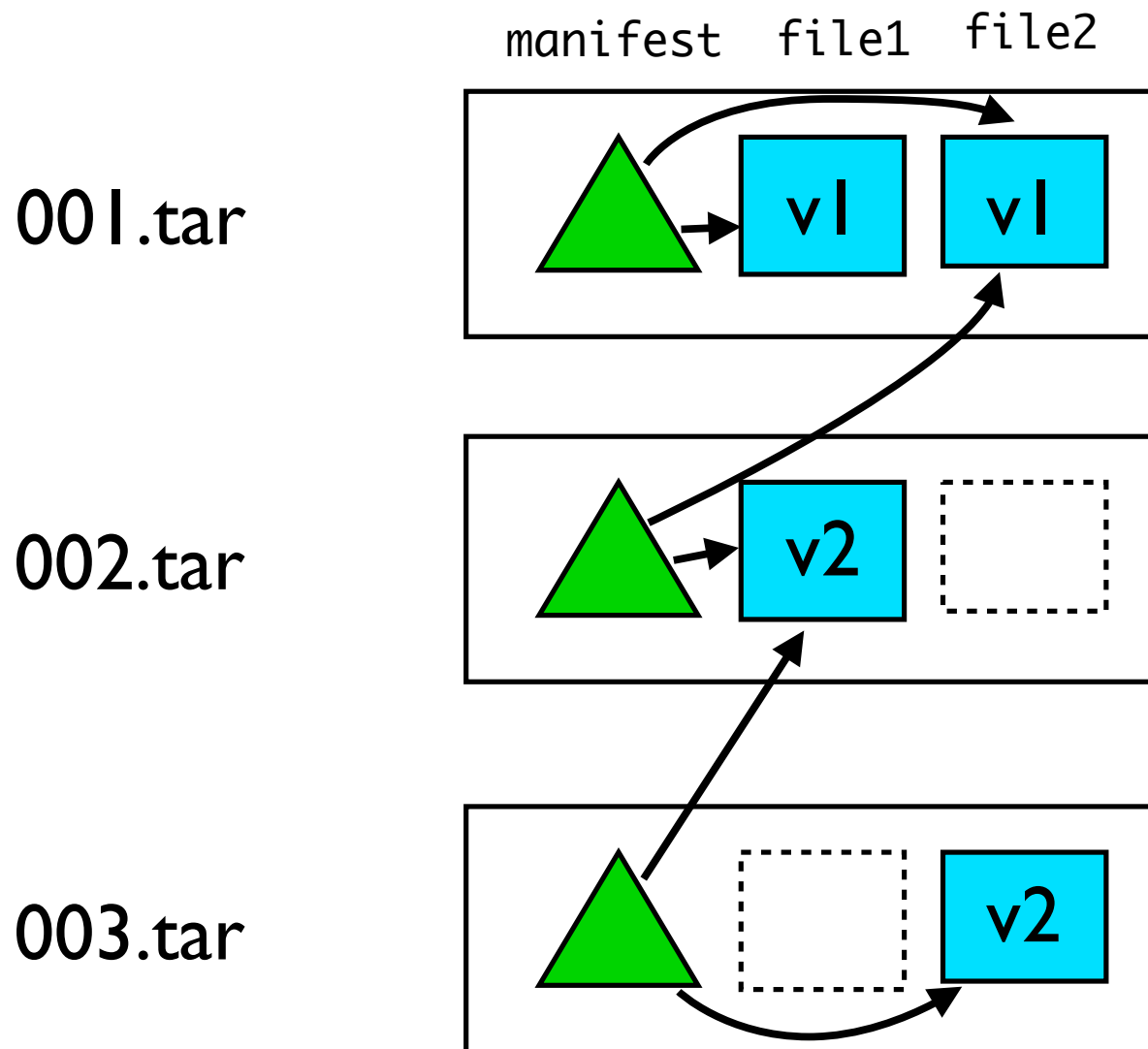


002.tar

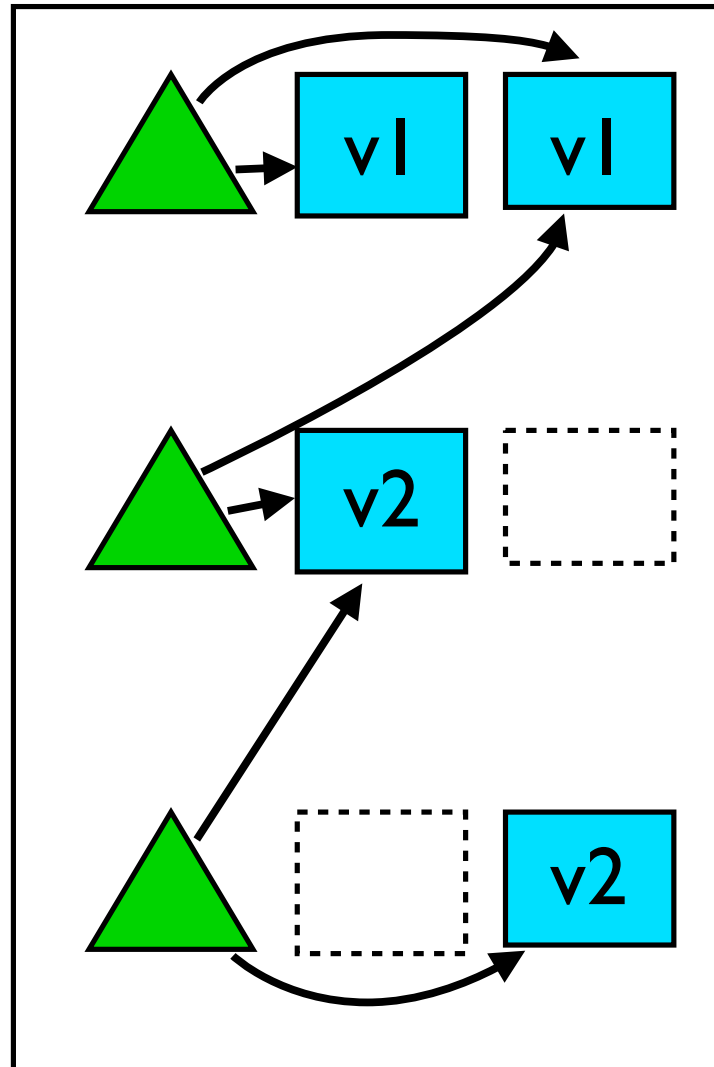


003.tar





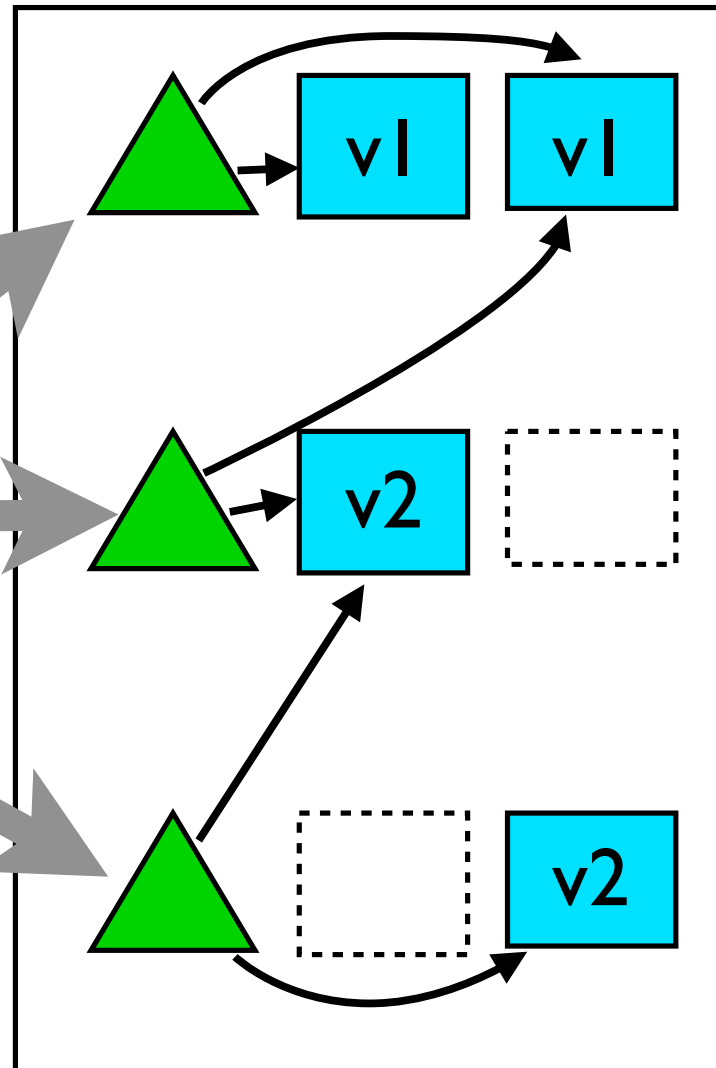
manifest file1 file2










Note:

Each snapshot is totally defined by the manifest file for that snapshot.

manifest file1 file2



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)








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.

File Name Contents

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

```
$ ls
```








```
aaaa aaab aaac aaad aae 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)

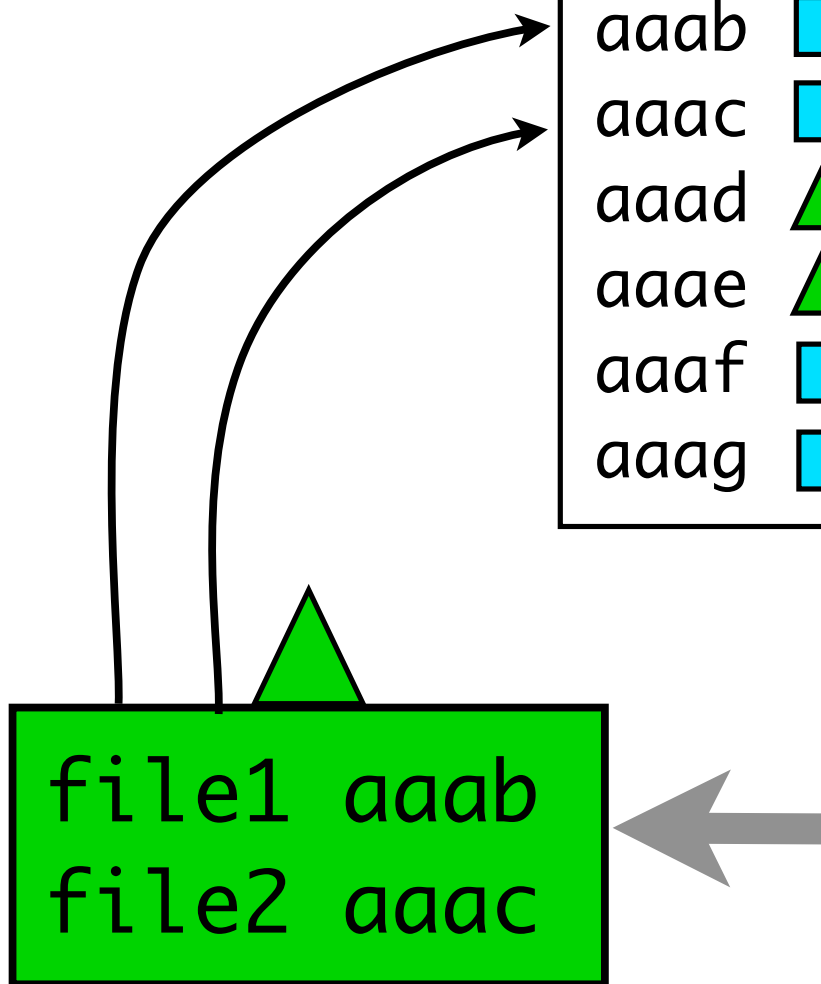


file1 aaab
file2 aaac

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.

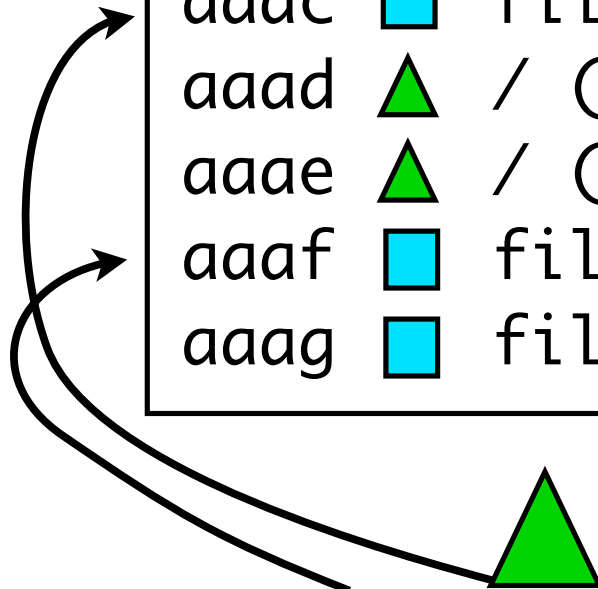
File

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)

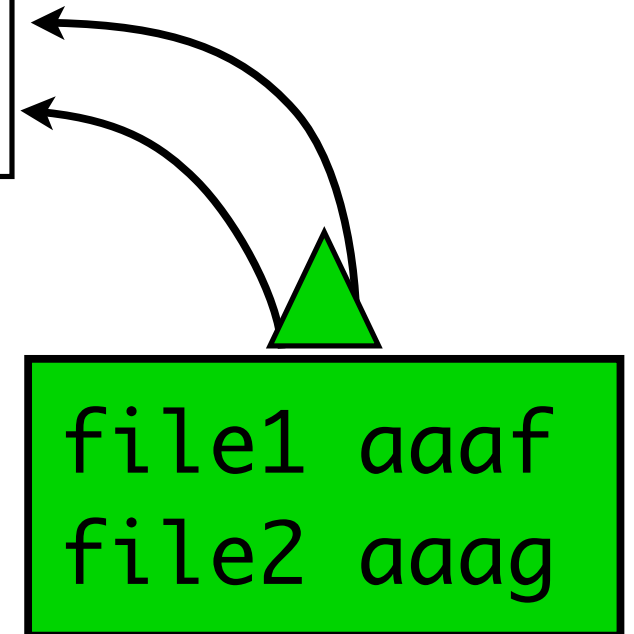
File

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)



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

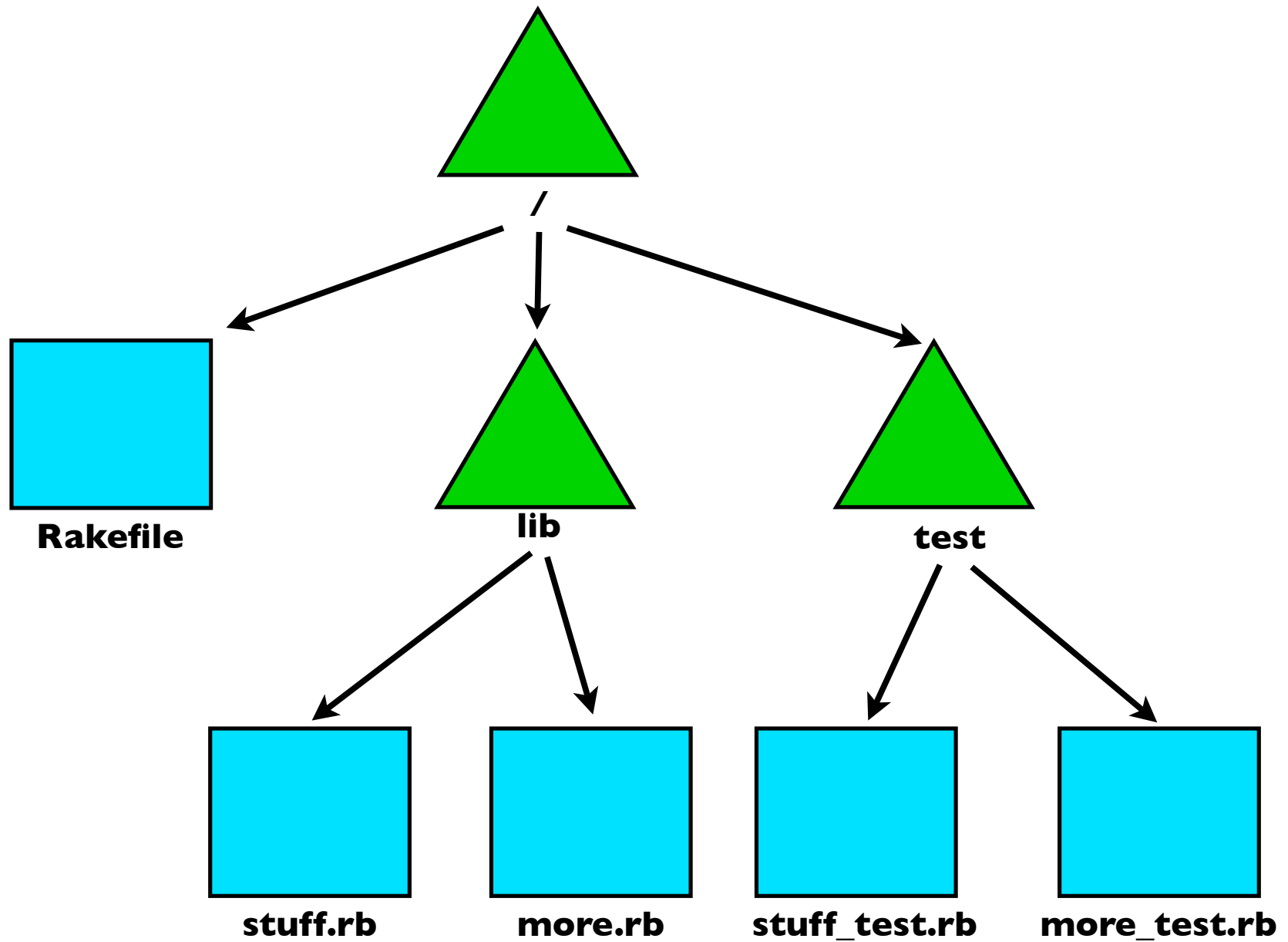
aaaa -- / (001)

file1 aaaf
file2 aaac

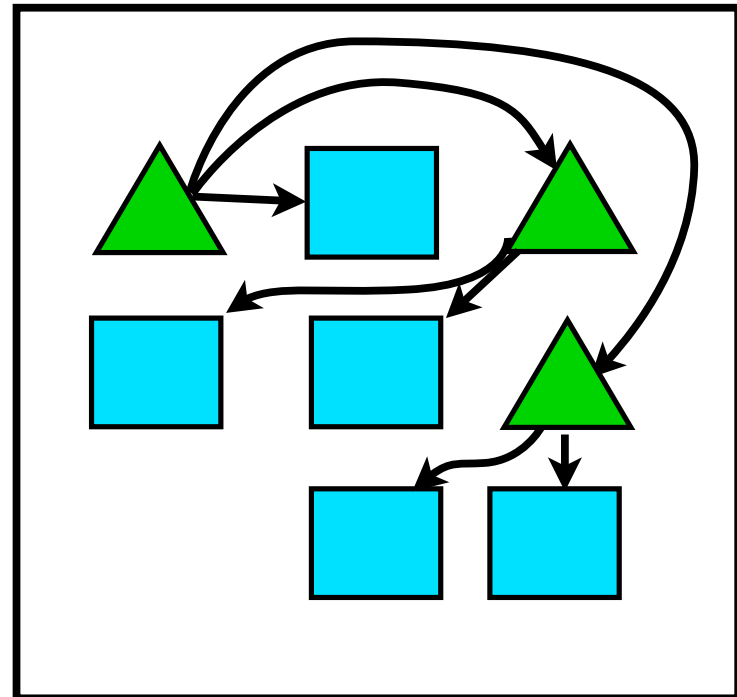
aaac -- / (002)

file1 aaaf
file2 aaag

aaae -- / (003)



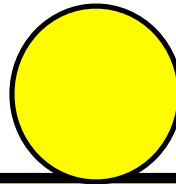
Project Archive



Metadata would
be nice

What Metadata?

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

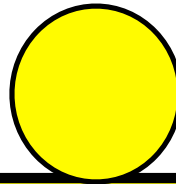


tree aaae

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

Comments describing this snapshot.

Root Manifest
for Snapshot

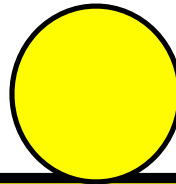


tree aaae

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

Comments describing this snapshot.

Author & EMail

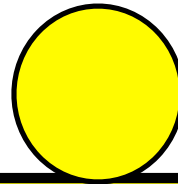


tree daae

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

Comments describing this snapshot.

Time Zone

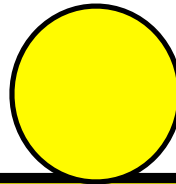


tree aaae

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

Comments describing this snapshot.

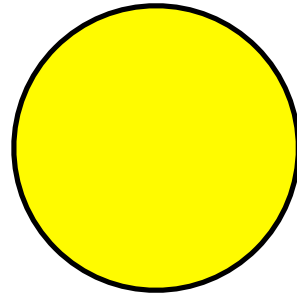
Date/Time
(seconds since 1970)



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

Comments describing this snapshot.

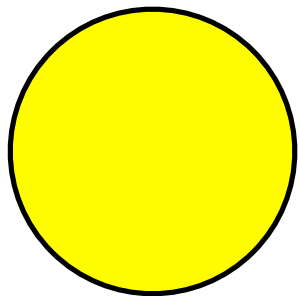
Comment

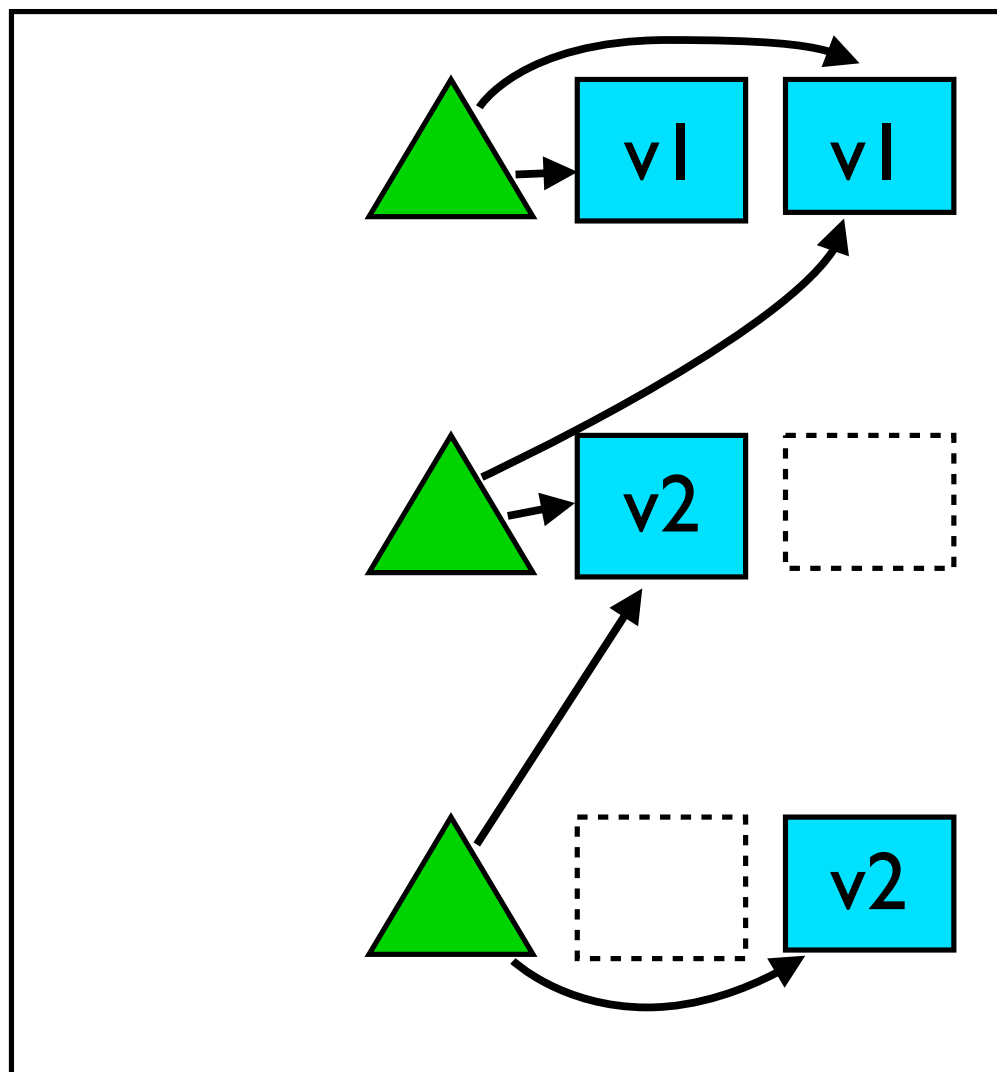
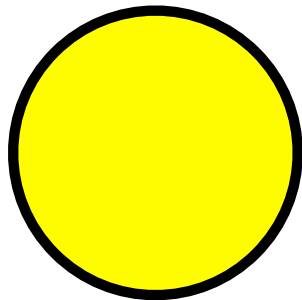


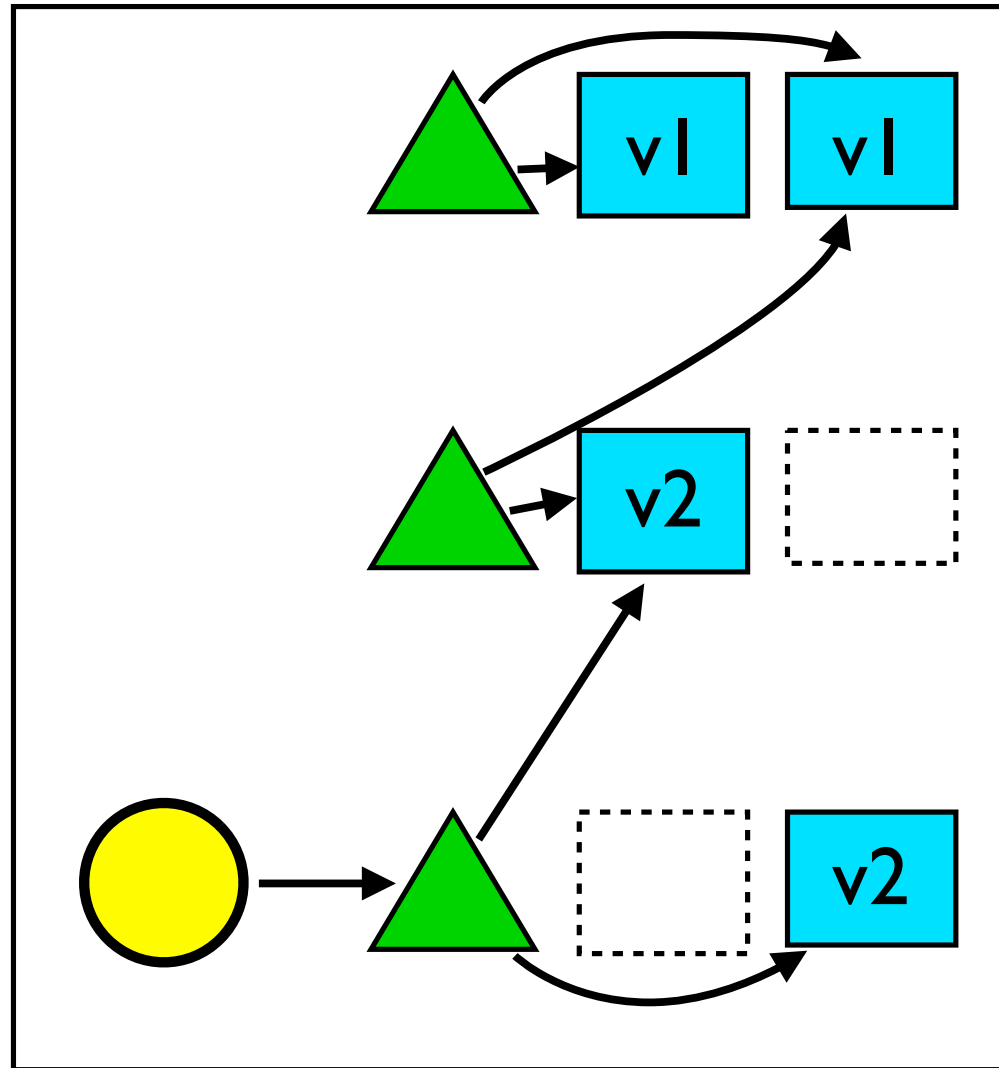
tree aaae

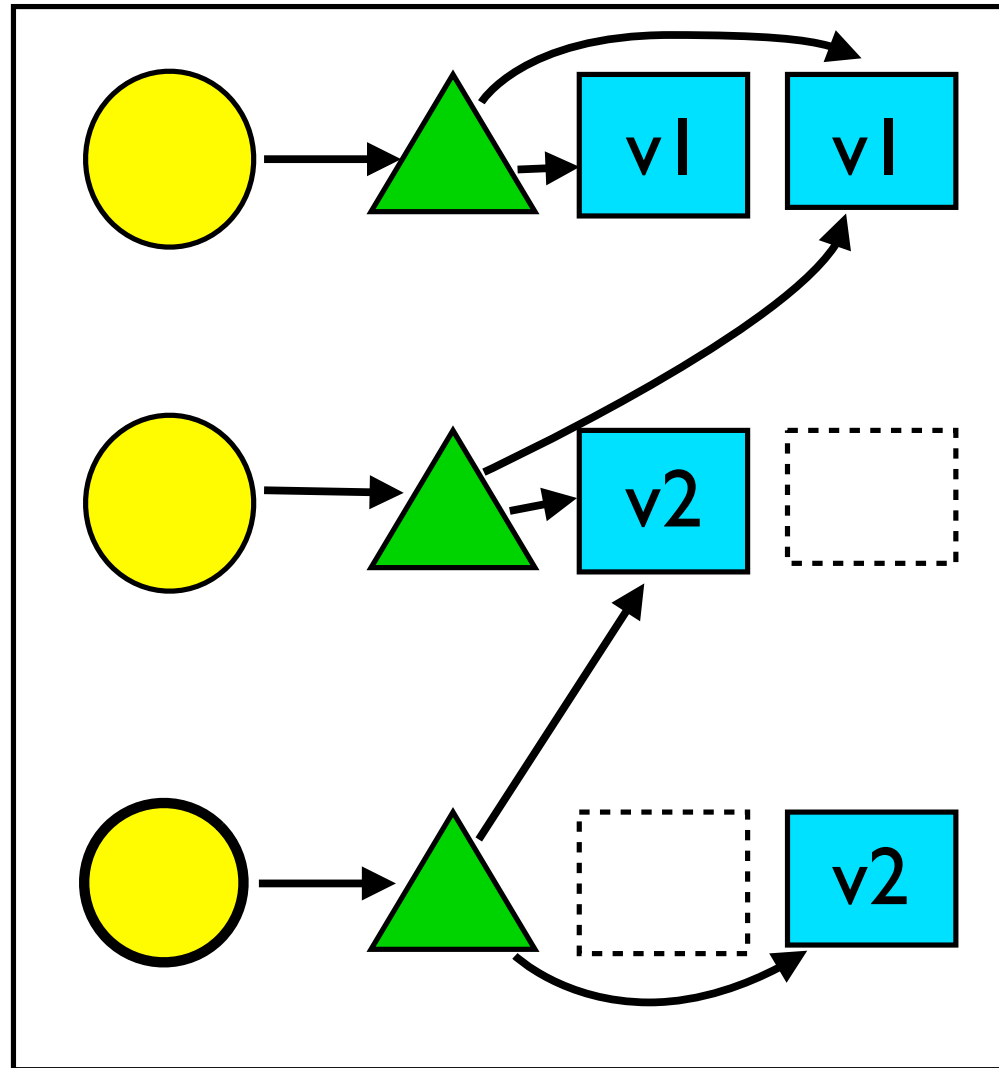
author Jim <me@email.com> 1246996320 -0400

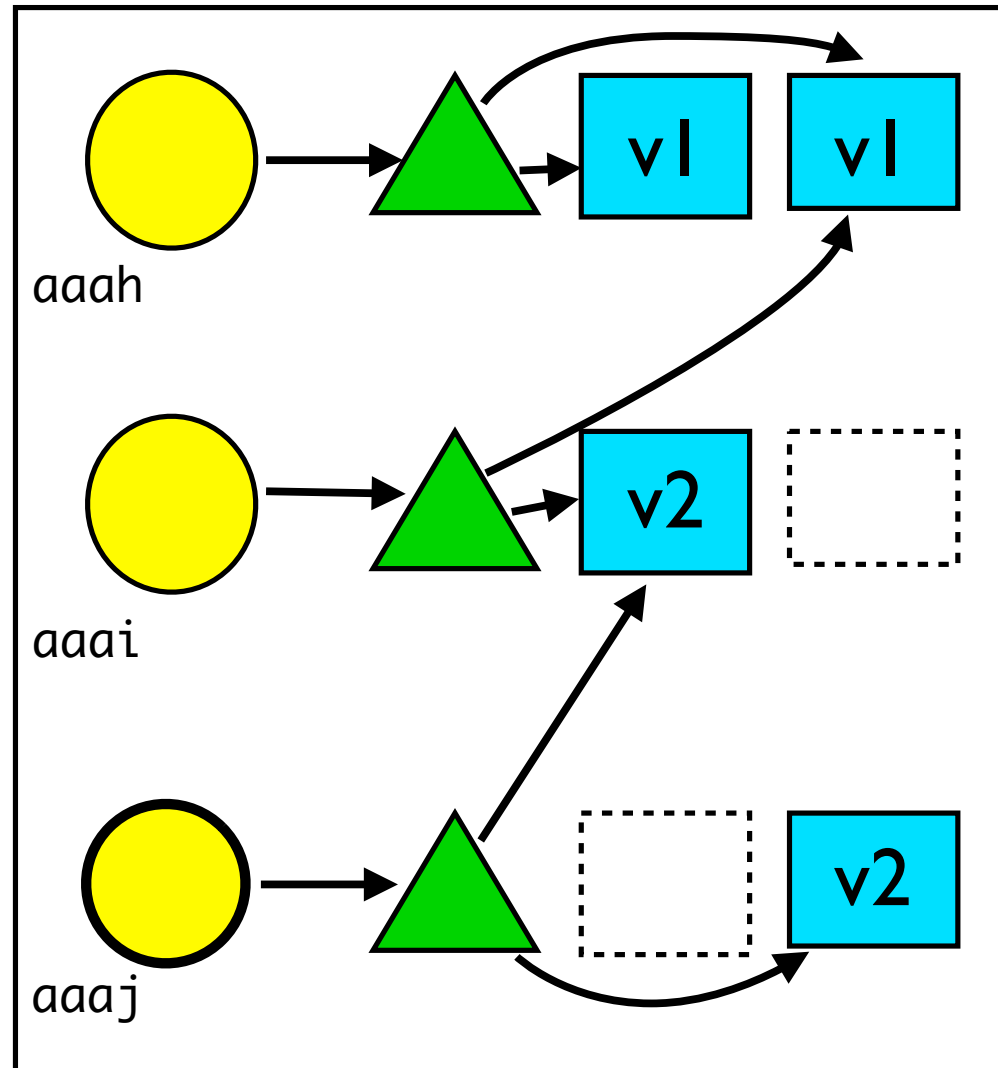
Comments describing this snapshot.











tree aaae

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

Comments describing this snapshot.

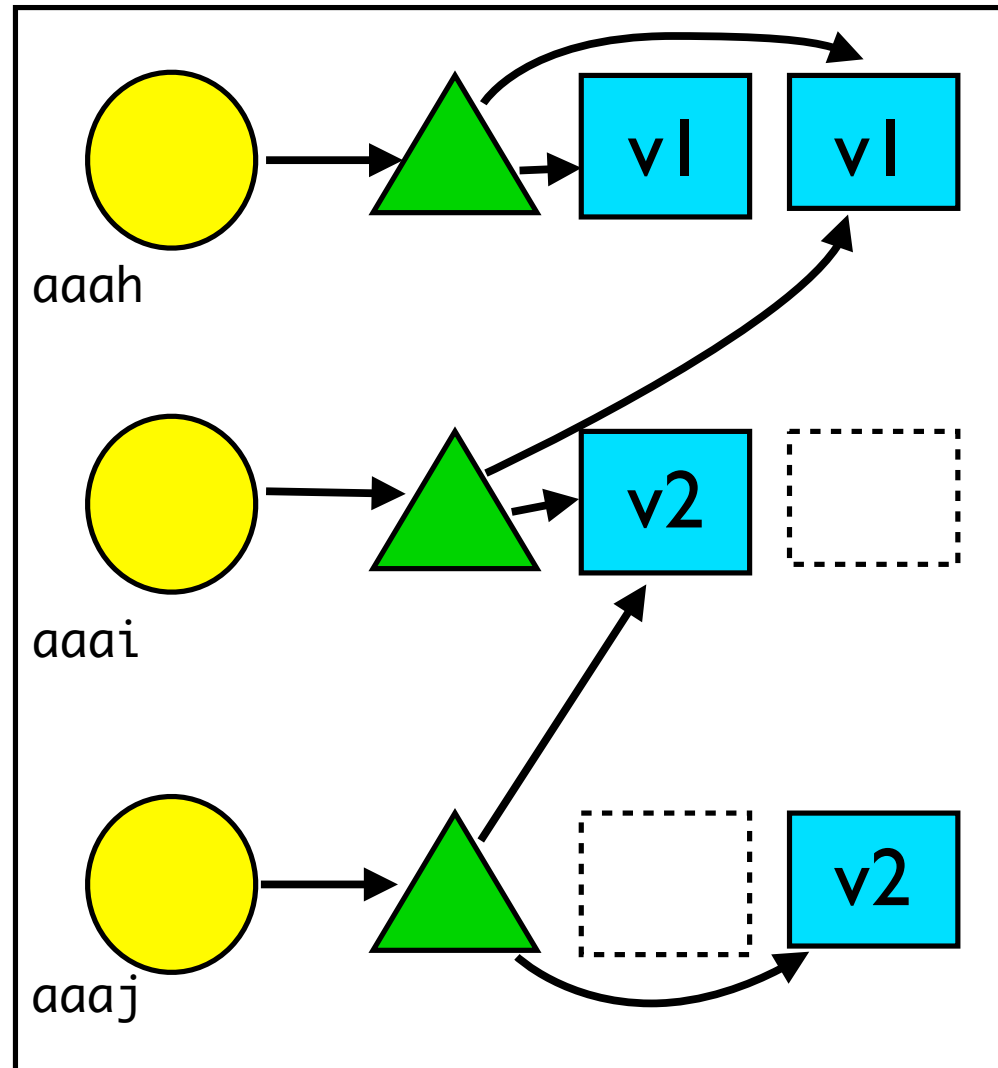
tree aaae

parent aaai

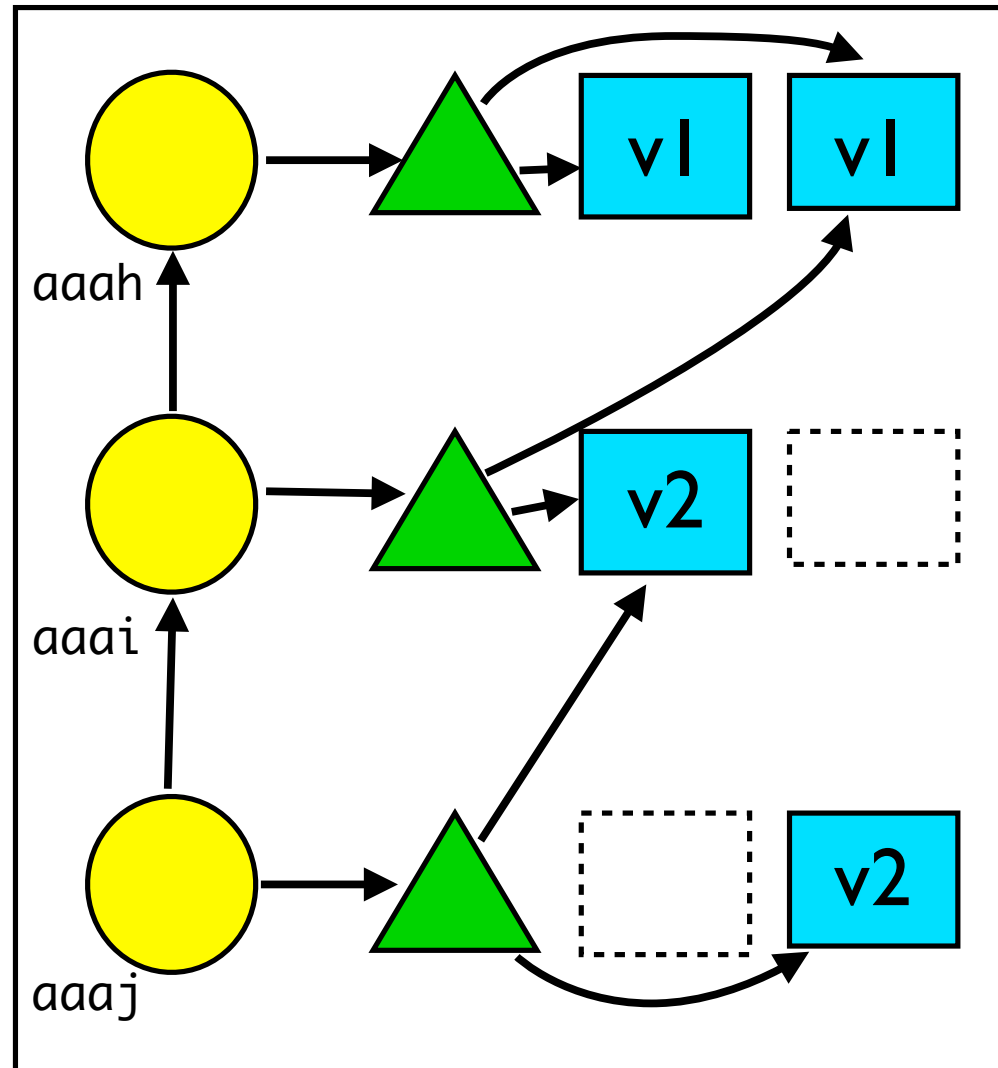
author Jim <me@email.com> 1246996320 -0400

Comments describing this snapshot.

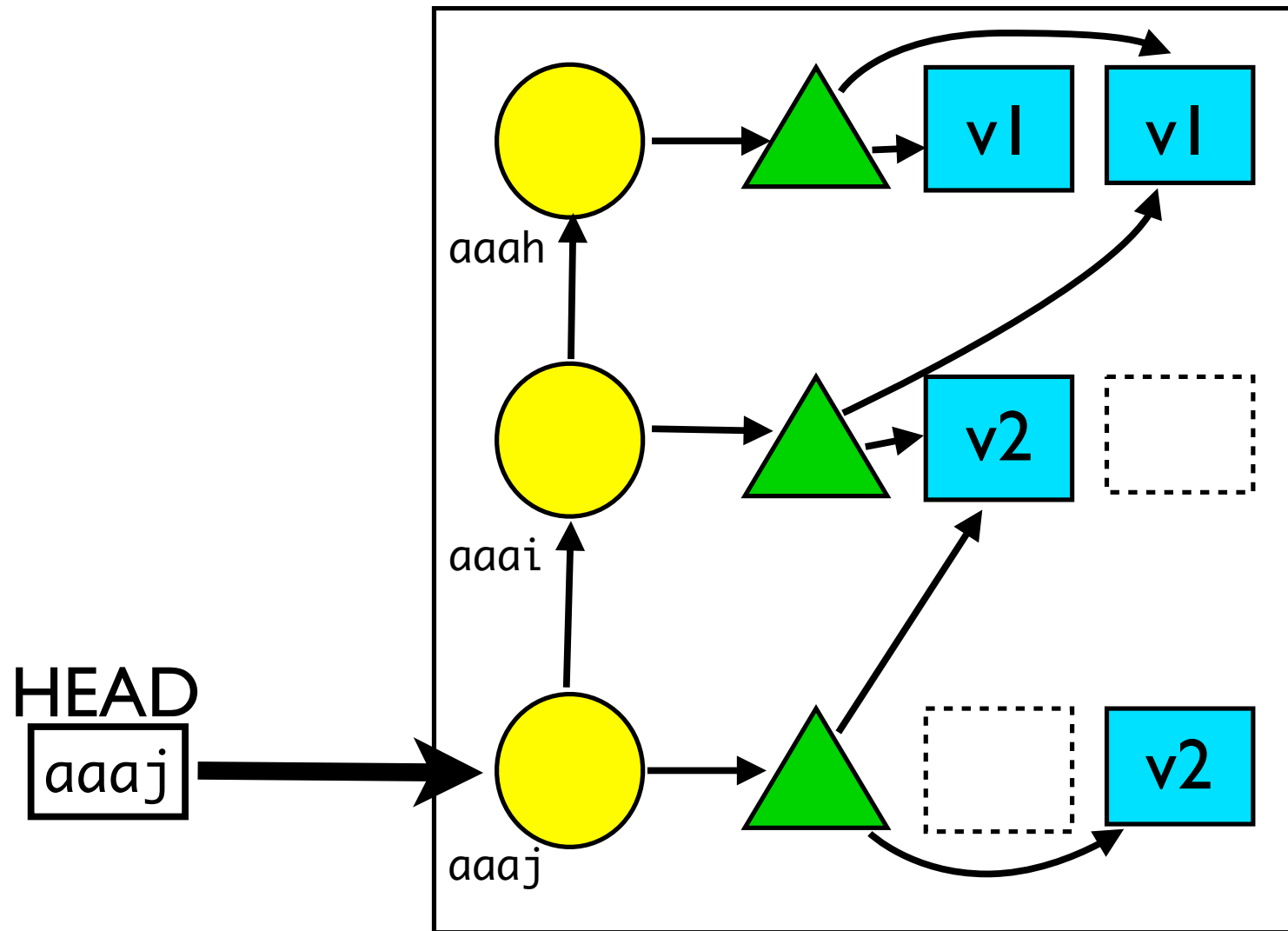
Project Archive



Project Archive

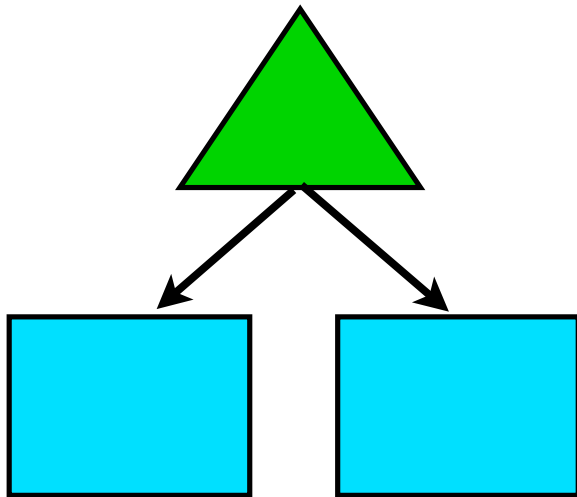


Project Archive



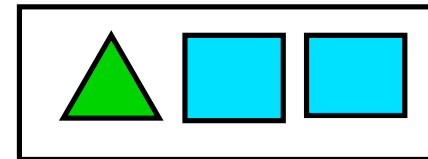
Summary So Far ...

Working Directory

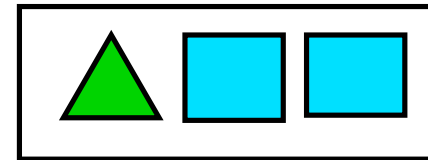


Project Archive

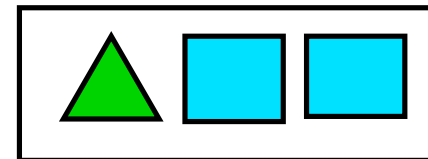
001.tar



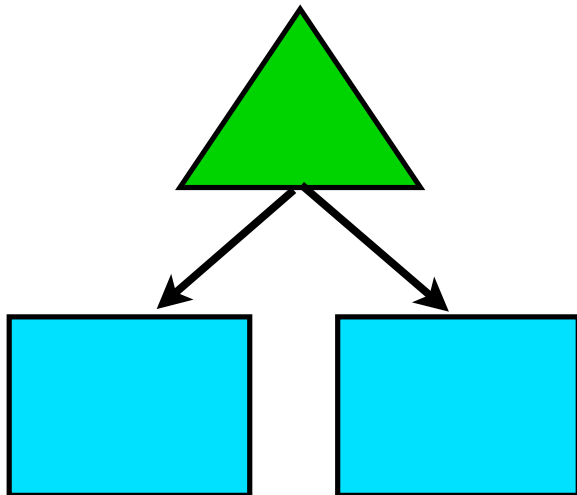
002.tar



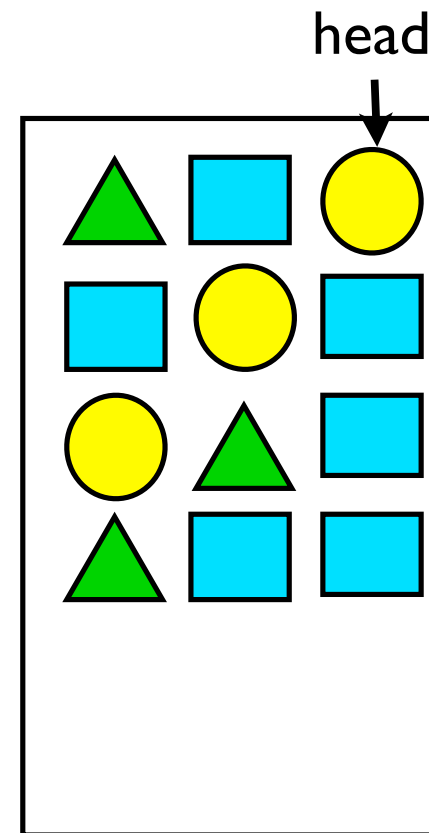
003.tar



Working Directory

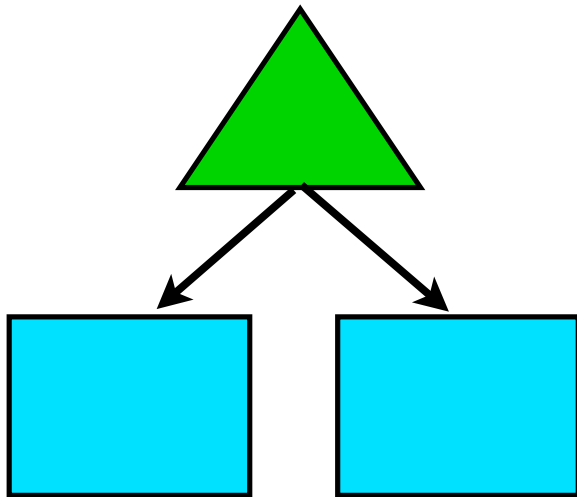


Project Archive



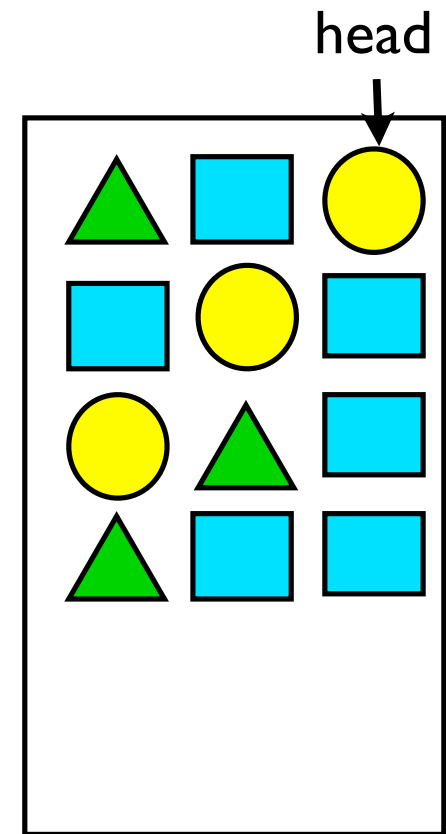
How do we create a
snapshot now?

Working
Directory

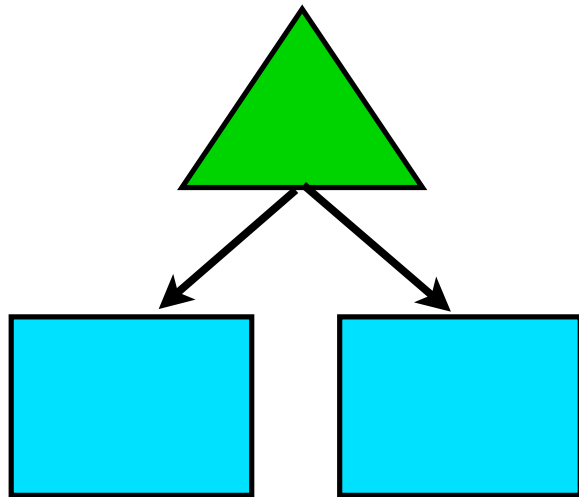


(I) Copy changed files to archive

Project Archive

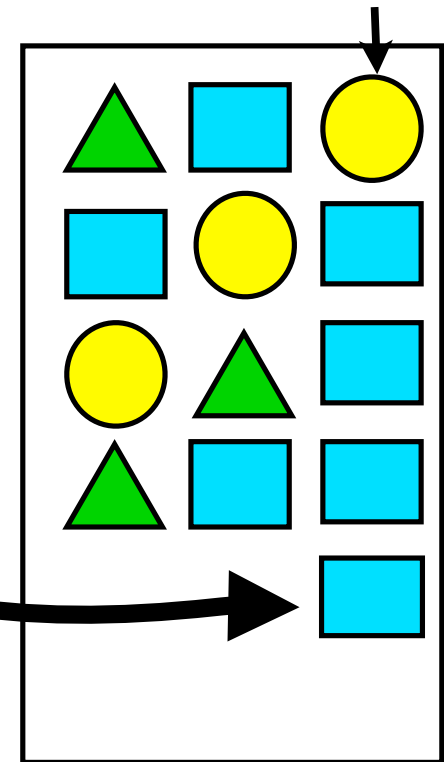


Working
Directory



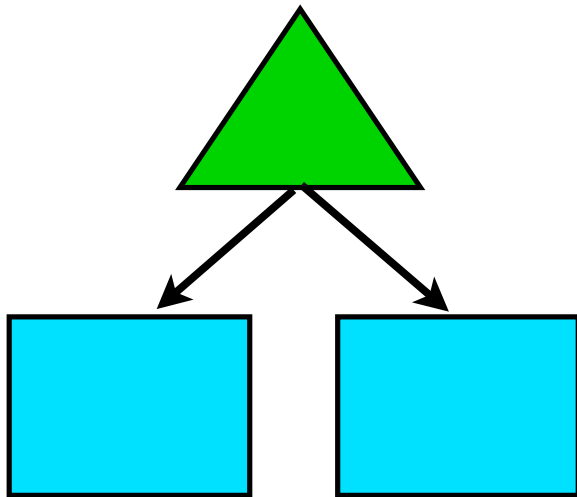
Project Archive

head



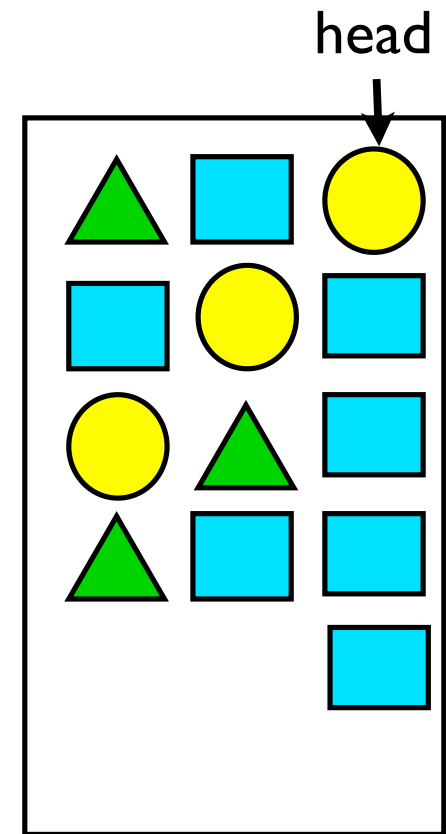
(I) Copy changed files to archive

Working Directory

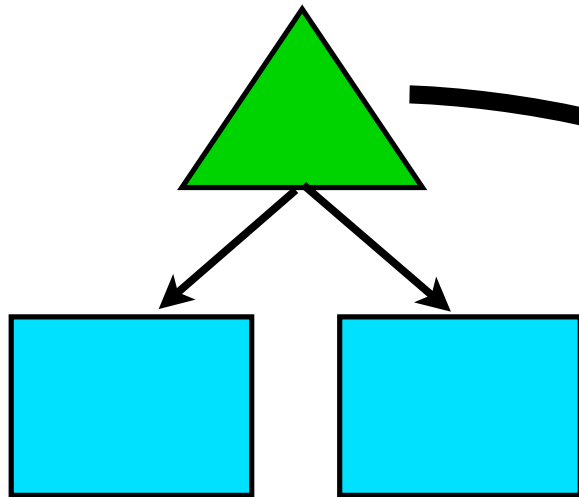


- (1) Copy changed files to archive
- (2) Create new manifest

Project Archive

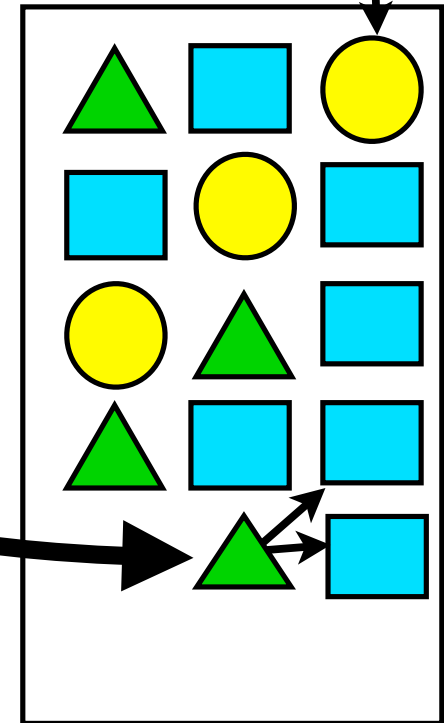


Working
Directory



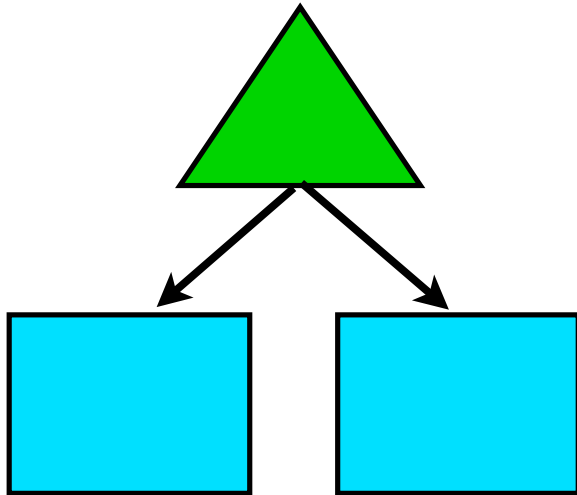
Project Archive

head



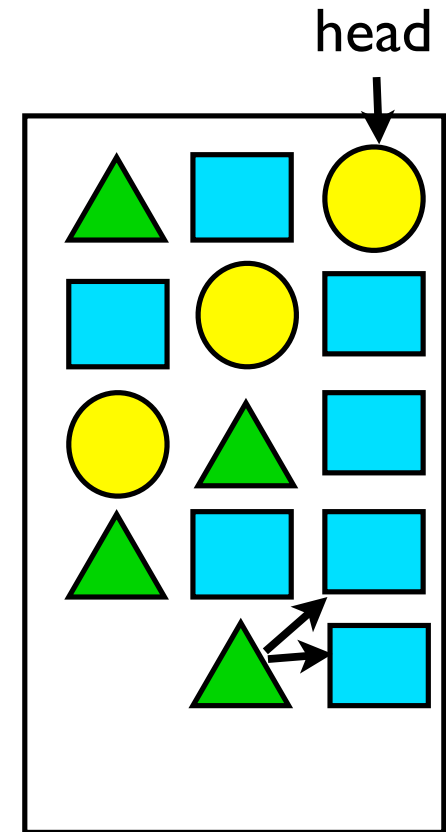
- (1) Copy changed files to archive
- (2) Create new manifest

Working Directory

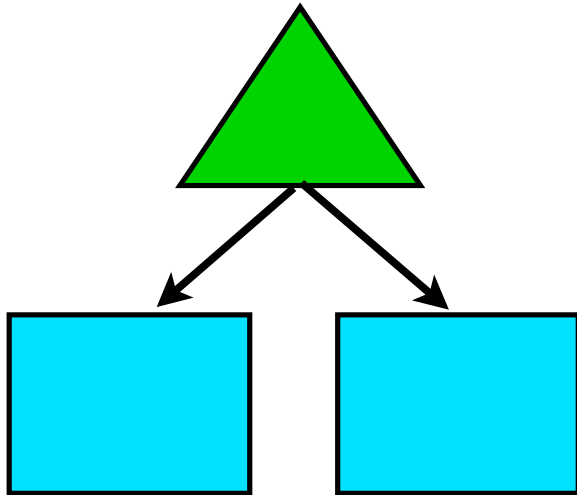


- (1) Copy changed files to archive
- (2) Create new manifest
- (3) Create new snapshot record

Project Archive

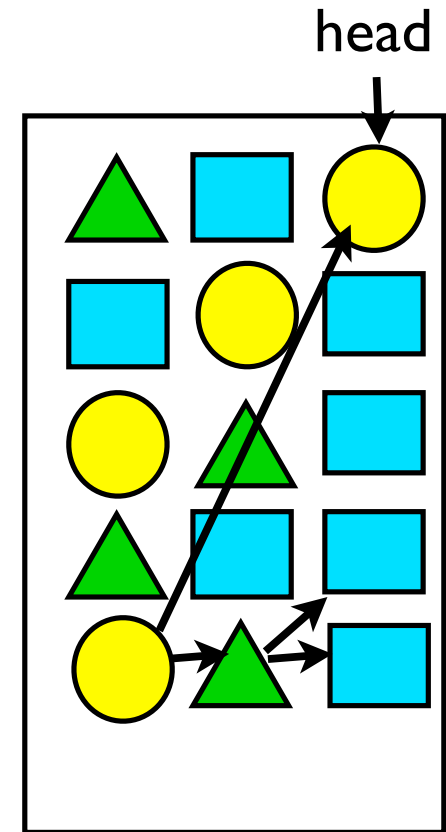


Working Directory

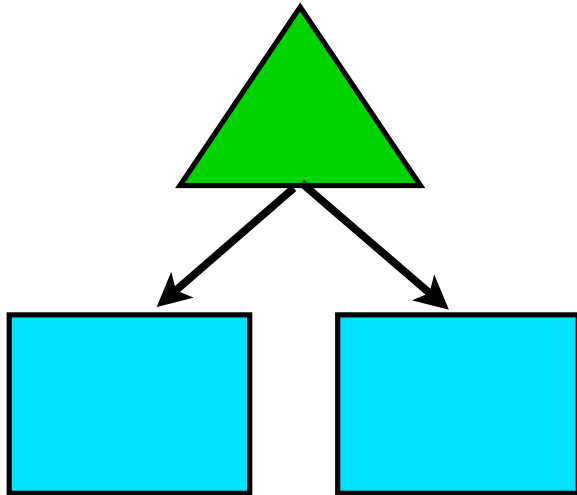


- (1) Copy changed files to archive
- (2) Create new manifest
- (3) Create new snapshot record

Project Archive

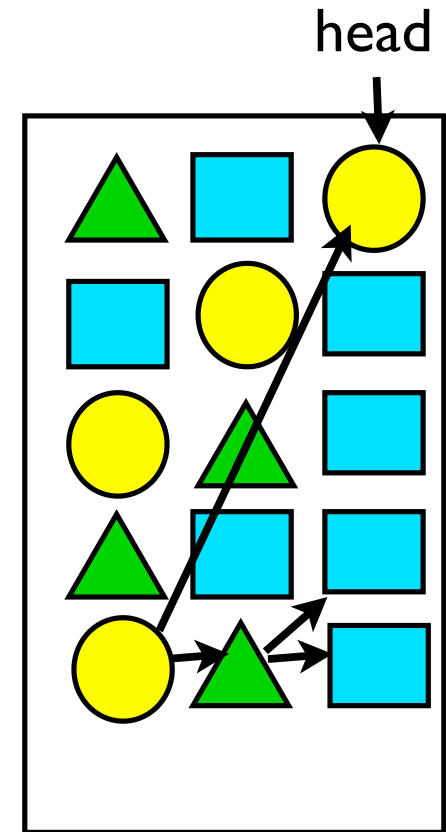


Working Directory

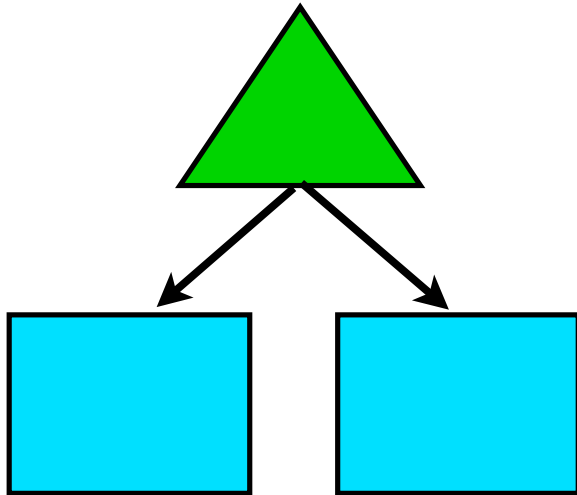


- (1) Copy changed files to archive
- (2) Create new manifest
- (3) Create new snapshot record
- (4) Readjust HEAD

Project Archive

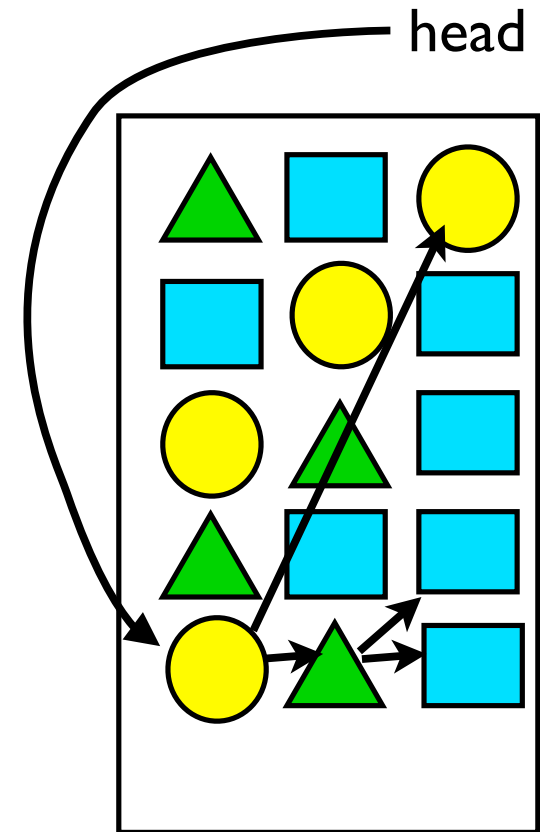


Working Directory



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

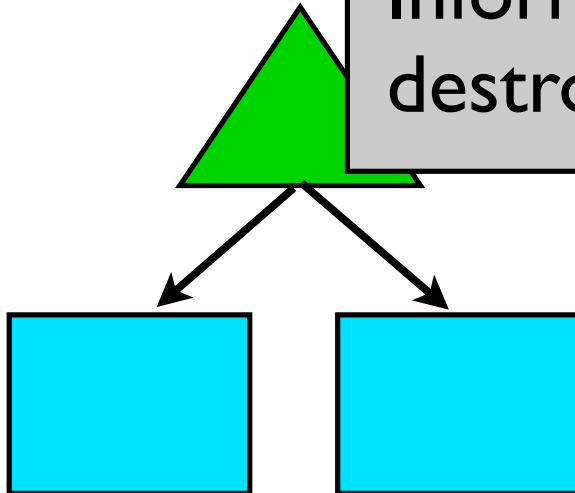
Project Archive



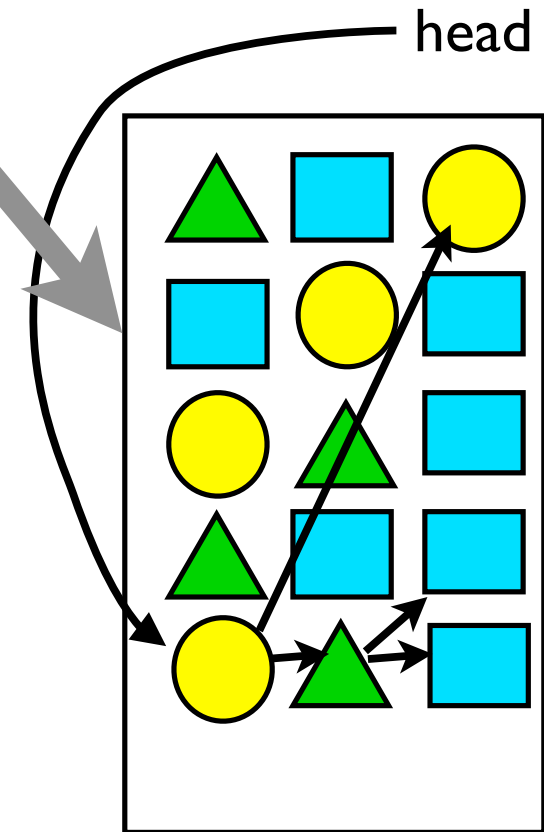
Work
Direct

Note:

The Project Archive
is always *additive*.
Information is never
destroyed.

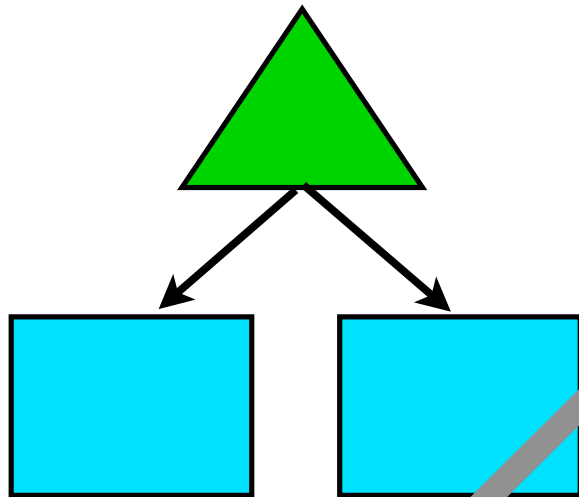


Project Archive



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

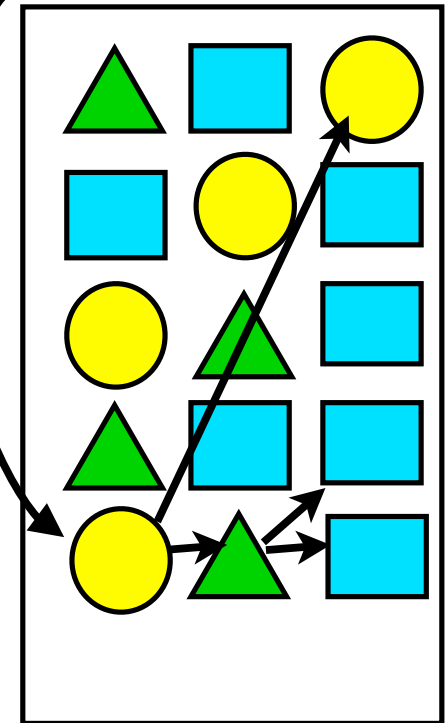
Working
Directory



Question:
How do we
detect changes?

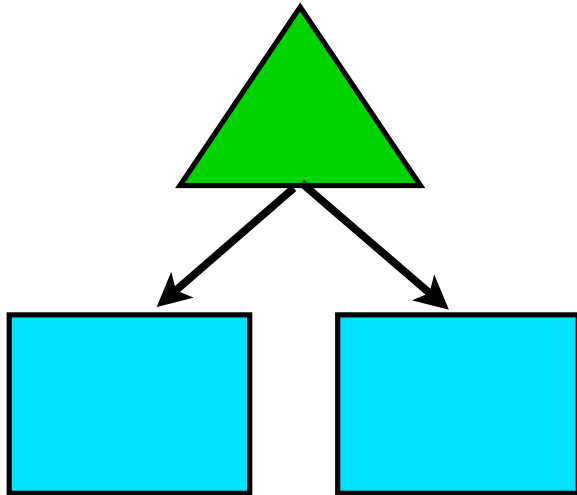
Project Archive

head



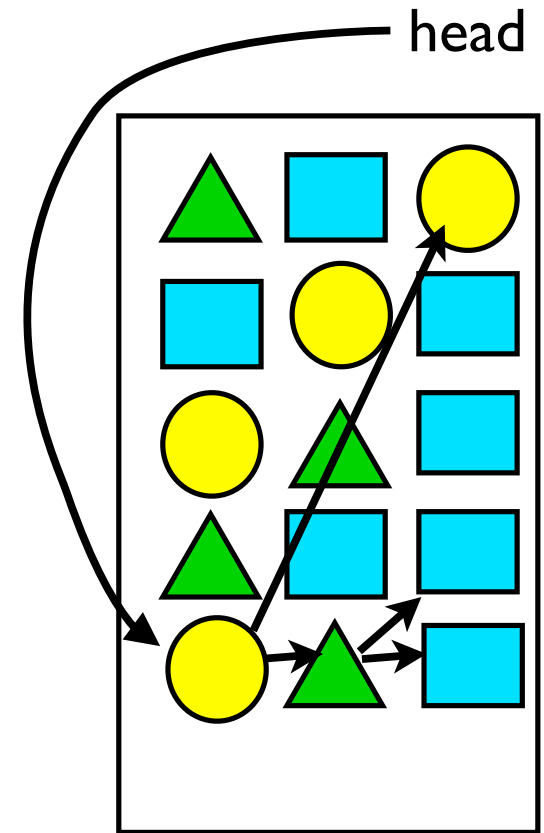
- (1) Copy *changed* files to repo
- (2) Create new directory
- (3) Create new Commit Record
- (4) Readjust HEAD

Working Directory

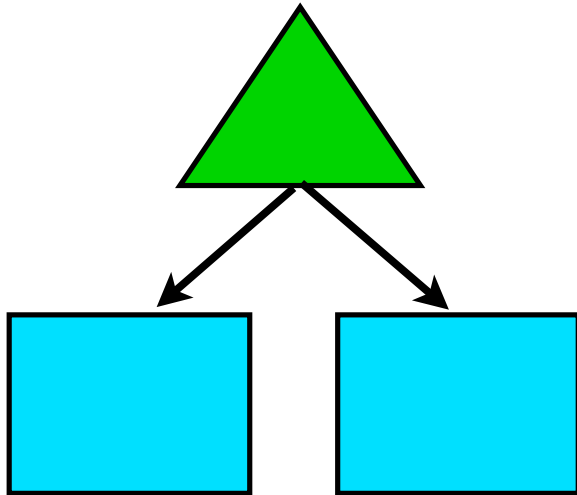


- (1) Copy *changed* files to repo
- (2) Create new directory
- (3) Create new Commit Record
- (4) Readjust HEAD

Project Archive

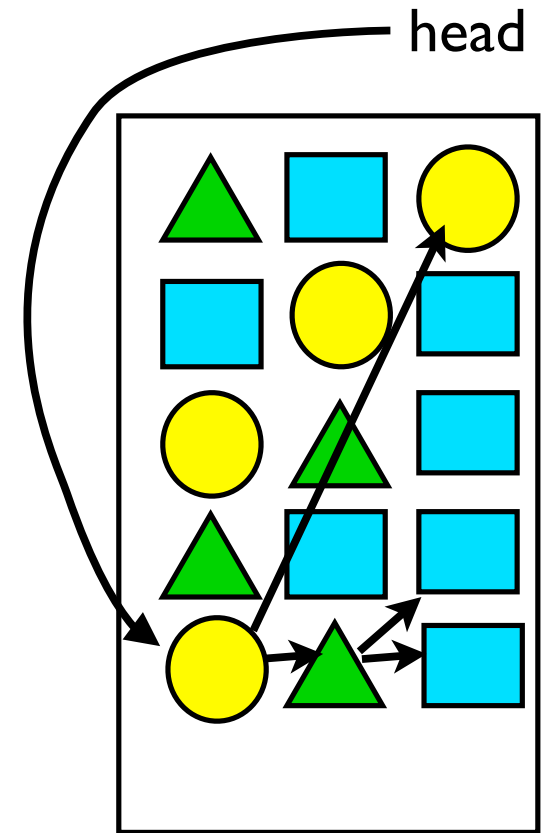


Working Directory

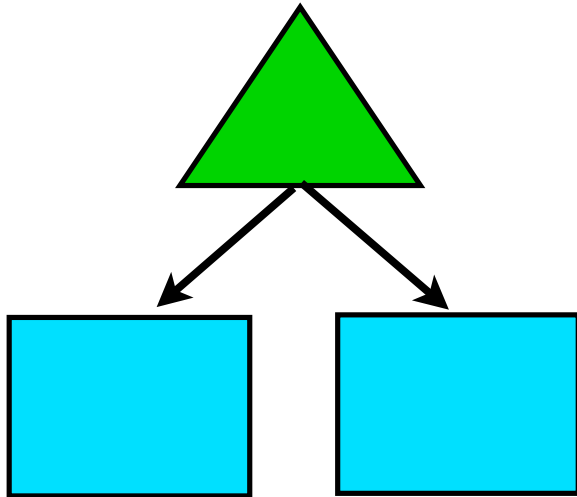


- (1) Copy *changed* files to repo
- (2) Create new directory
- (3) Create new Commit Record
- (4) Readjust HEAD

Project Archive

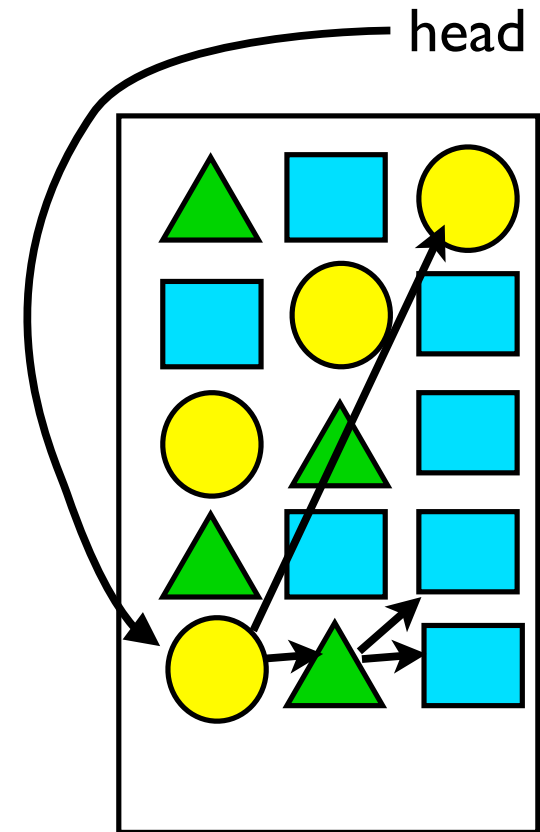


Working Directory

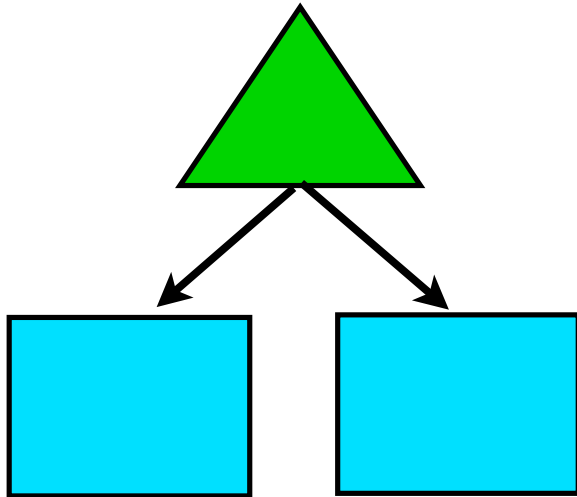


- (1) Copy *changed* files to repo
- (2) Create new directory
- (3) Create new Commit Record
- (4) Readjust HEAD

Project Archive

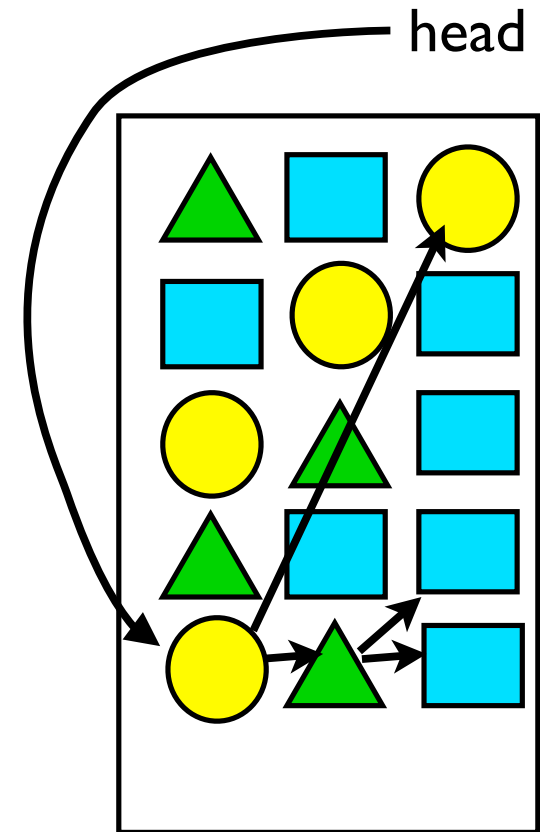


Working Directory



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

Project Archive



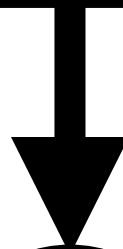
Detecting Duplication

file.txt

Hi, my name is Bob.

file.txt

Hi, my name is Bob.



SHA1

file.txt

Hi, my name is Bob.

SHA1

"55dc96ede241511e35f1973957bbd51f37f3762b"

file.txt

Hi, my name is Joe.

SHA1

"55dc96ede241511e35f1973957bbd51f37f3762b"

file.txt

Hi, my name is Joe.

SHA1

~~"55dc96ede241511e35f1973957bbd51f37f3762b"~~

"53f2509e877de3cad85c547a0c2a22b43c5d81eb"

file1.txt

Hi, my name is Bob.

file2.txt

Hi, my name is Bob.

SHA1

```
graph TD; file1[file1.txt] --> sha1((SHA1)); file2[file2.txt] --> sha1; sha1 --> output["55dc96ede241511e35f1973957bbd51f37f3762b"]
```

The diagram illustrates a SHA1 hashing process. Two input files, 'file1.txt' and 'file2.txt', both containing the text 'Hi, my name is Bob.', are fed into a central circular node labeled 'SHA1'. An arrow points from this node to the resulting SHA1 hash value.

"55dc96ede241511e35f1973957bbd51f37f3762b"

file1.txt

Hi, my name is Bob.








file2.txt

Hi, my name is Bob.

**Identical Content
Products
Identical Hashes**








"55dc96ede241511e35f1973957bbd51f37f3762b"

File Names in the Archive

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

File Names in the Archive










- Arbitrary
- Unique

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

Idea:

Use the SHA1 Hash
for the File Name

Project Archive

aaaa	
aaab	
aaac	
aaad	
aaae	
aaaf	
aaag	
aaah	
aaai	

Project Archive

75e3dee3503e7ad5...



1586ad8975e3dee3...



f6da34481586ad89...



f99b89d6f6da3448...



503d6749f99b89d6...



2b436957503d6749...



944ff522944ff522...



6c024e25944ff522...



503e7ad56c024e25...



Project Archive

HEAD

75e3dee3503e7ad5...

75e3dee3503e7ad5...



1586ad8975e3dee3...



f6da34481586ad89...



f99b89d6f6da3448...



503d6749f99b89d6...



2b436957503d6749...



944ff522944ff522...

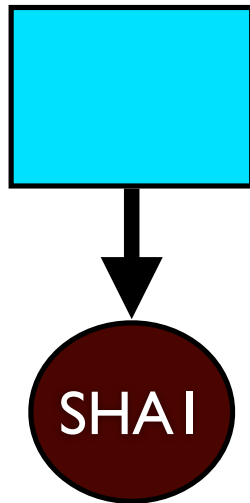


6c024e25944ff522...



503e7ad56c024e25...





Project Archive

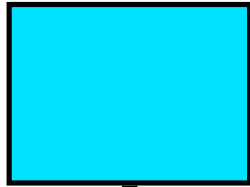
75e3dee3503e7ad5...	●
1586ad8975e3dee3...	■
f6da34481586ad89...	■
f99b89d6f6da3448...	▲
503d6749f99b89d6...	●
2b436957503d6749...	■
944ff522944ff522...	▲
6c024e25944ff522...	■
503e7ad56c024e25...	■



2b436957503d6749 ...

Project Archive

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



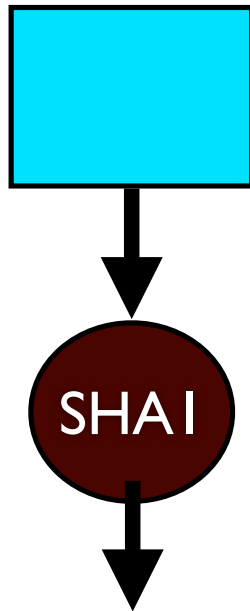
2b436957503d6749 ...

Match!



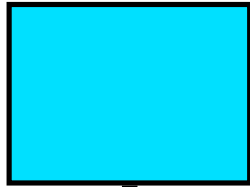
Project Archive

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



Project Archive

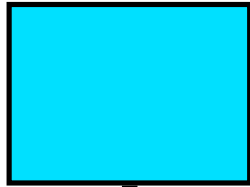
75e3dee3503e7ad5...	●
1586ad8975e3dee3...	■
f6da34481586ad89...	■
f99b89d6f6da3448...	▲
503d6749f99b89d6...	●
2b436957503d6749...	■
944ff522944ff522...	▲
6c024e25944ff522...	■
503e7ad56c024e25...	■



9d6f503d24e2d56c ...

Project Archive

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



9d6f503d24e2d56c ...

No
Match!

Project Archive

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

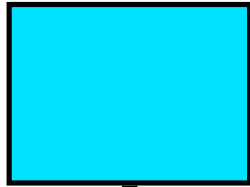


9d6f503d24e2d56c ...

Project Archive

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

Copy File



9d6f503d24e2d56c ...



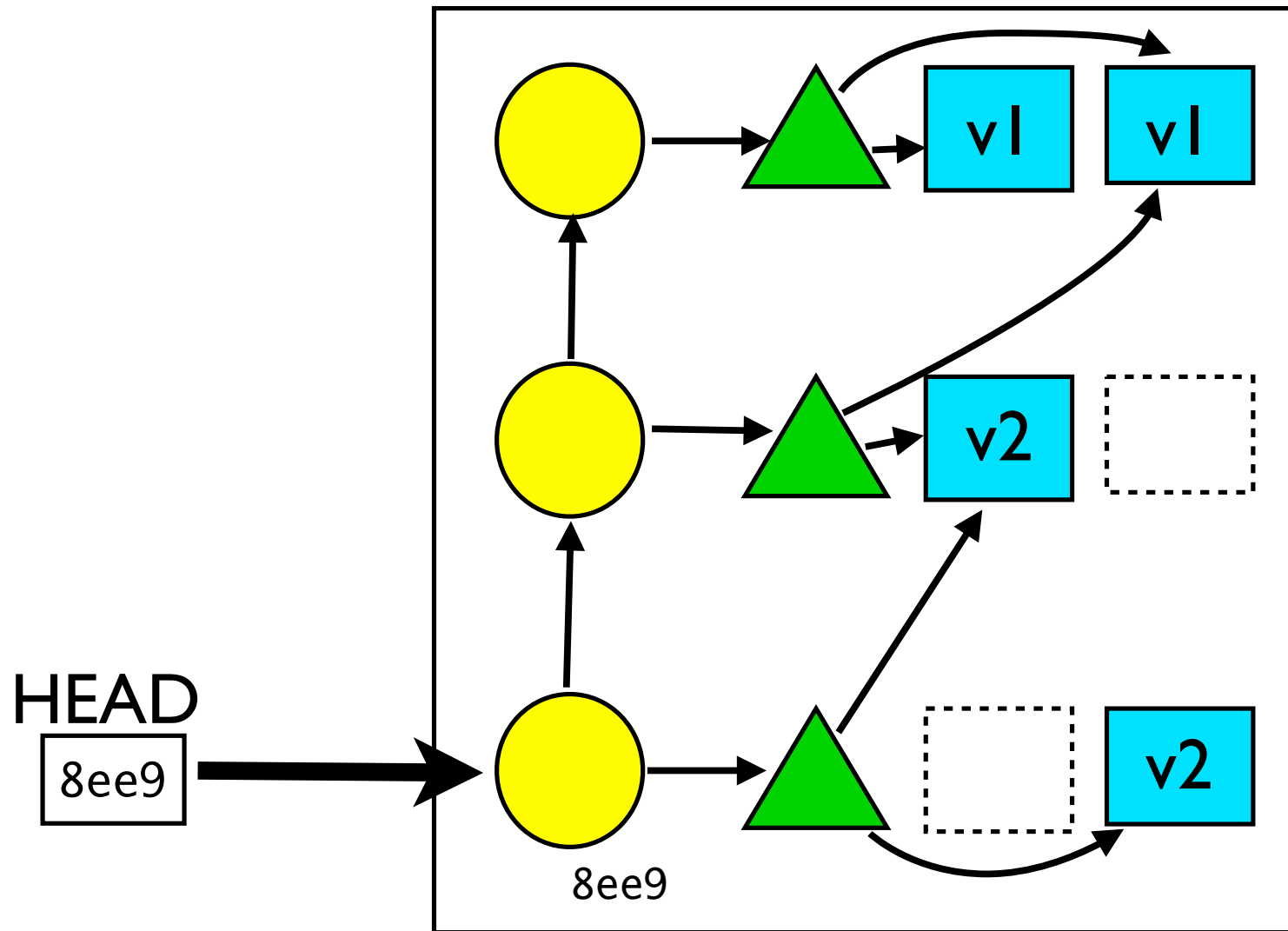
Copy File

Project Archive

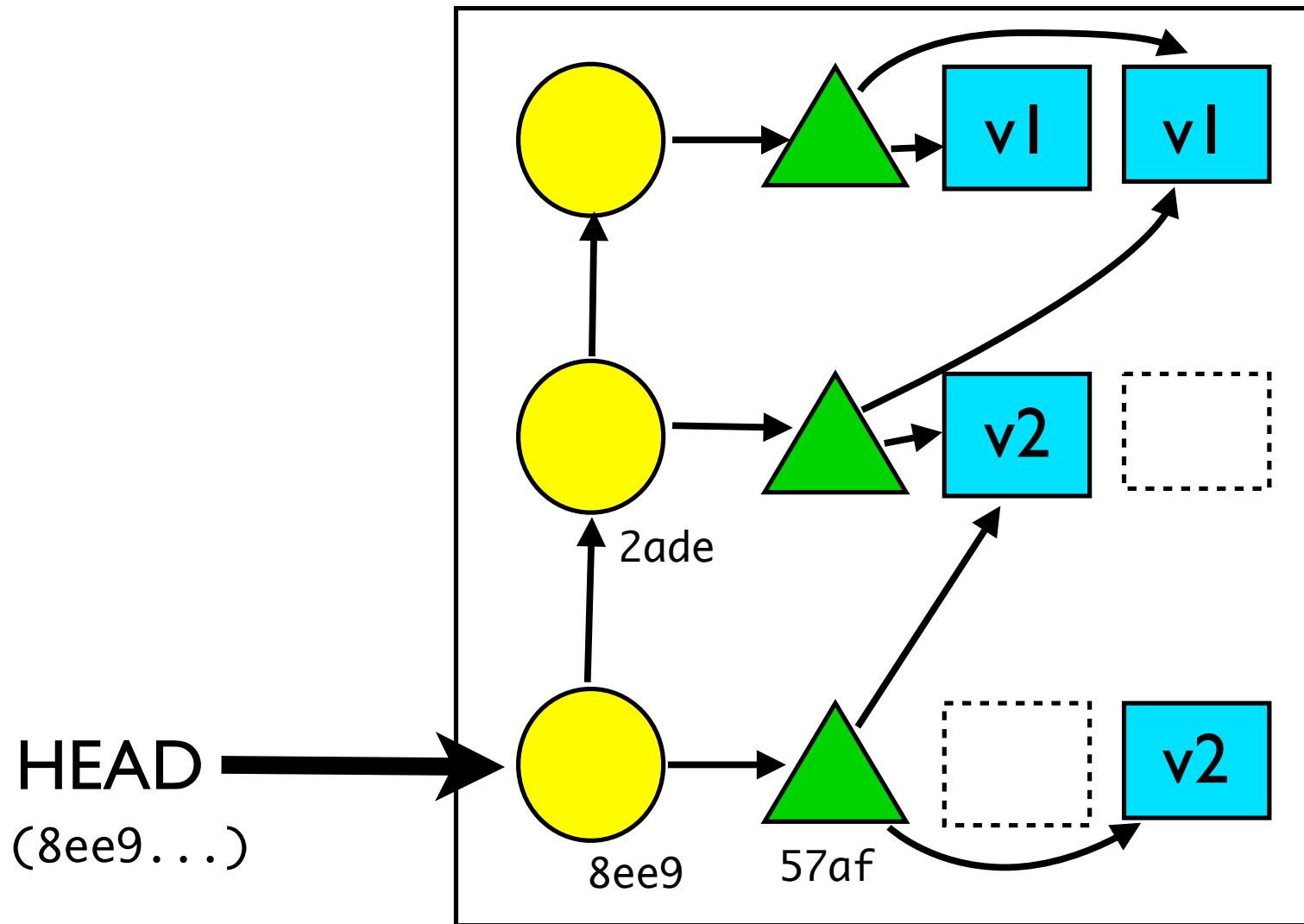
75e3dee3503e7ad5...	●
1586ad8975e3dee3...	■
f6da34481586ad89...	■
f99b89d6f6da3448...	▲
503d6749f99b89d6...	●
2b436957503d6749...	■
944ff522944ff522...	▲
6c024e25944ff522...	■
503e7ad56c024e25...	■
9d6f503d24e2d56c...	■

Project Archive Integrity

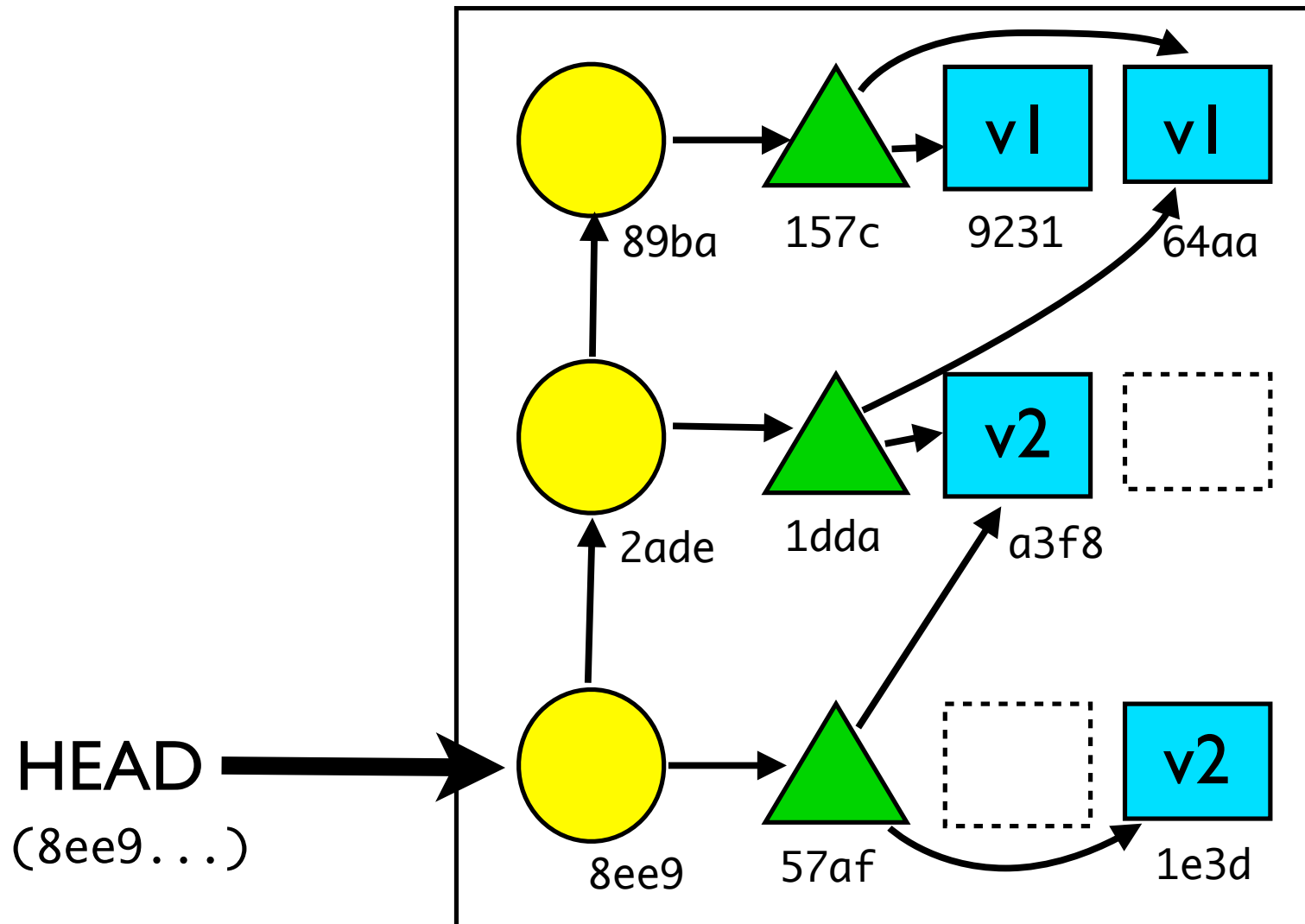
Project Archive

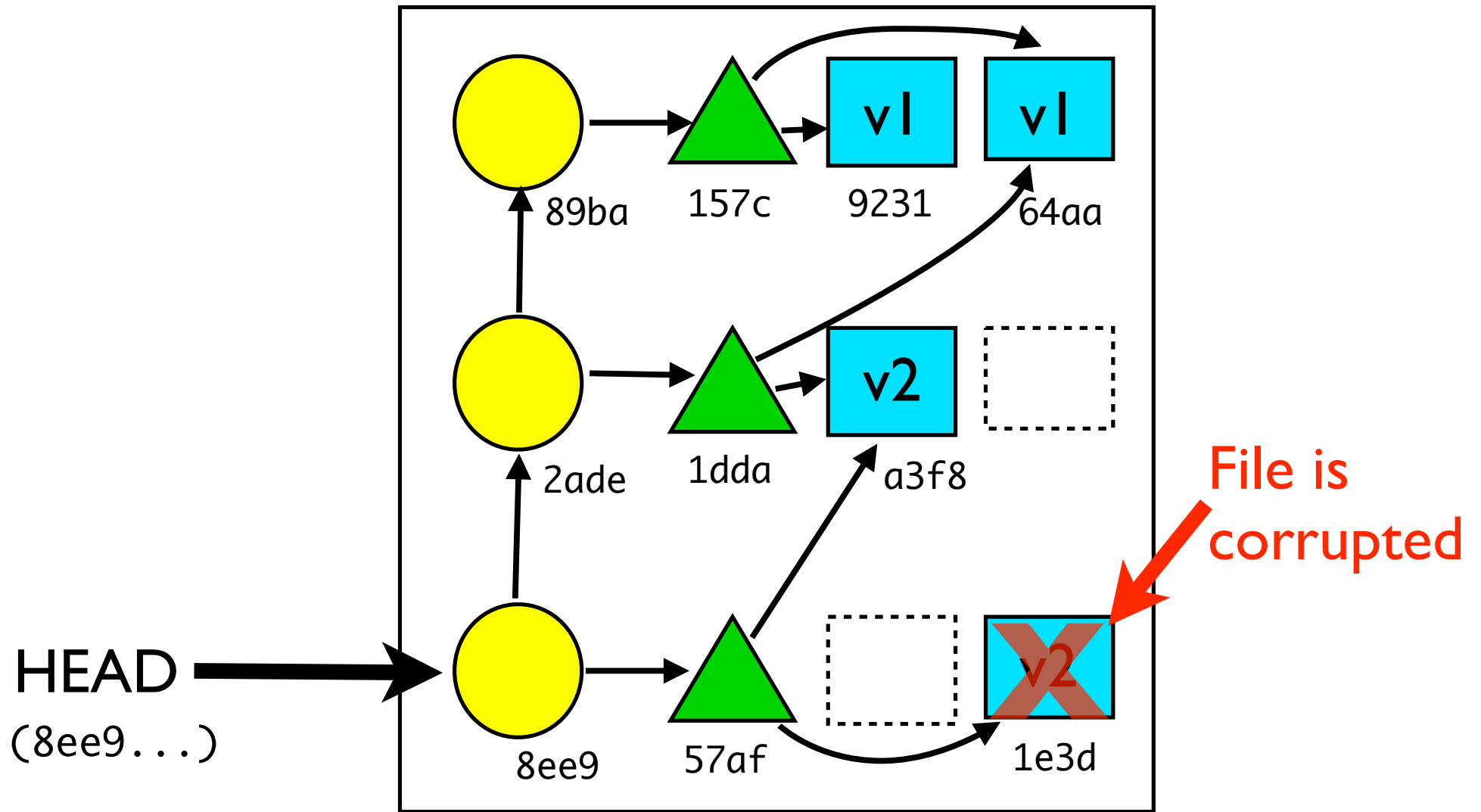


Project Archive

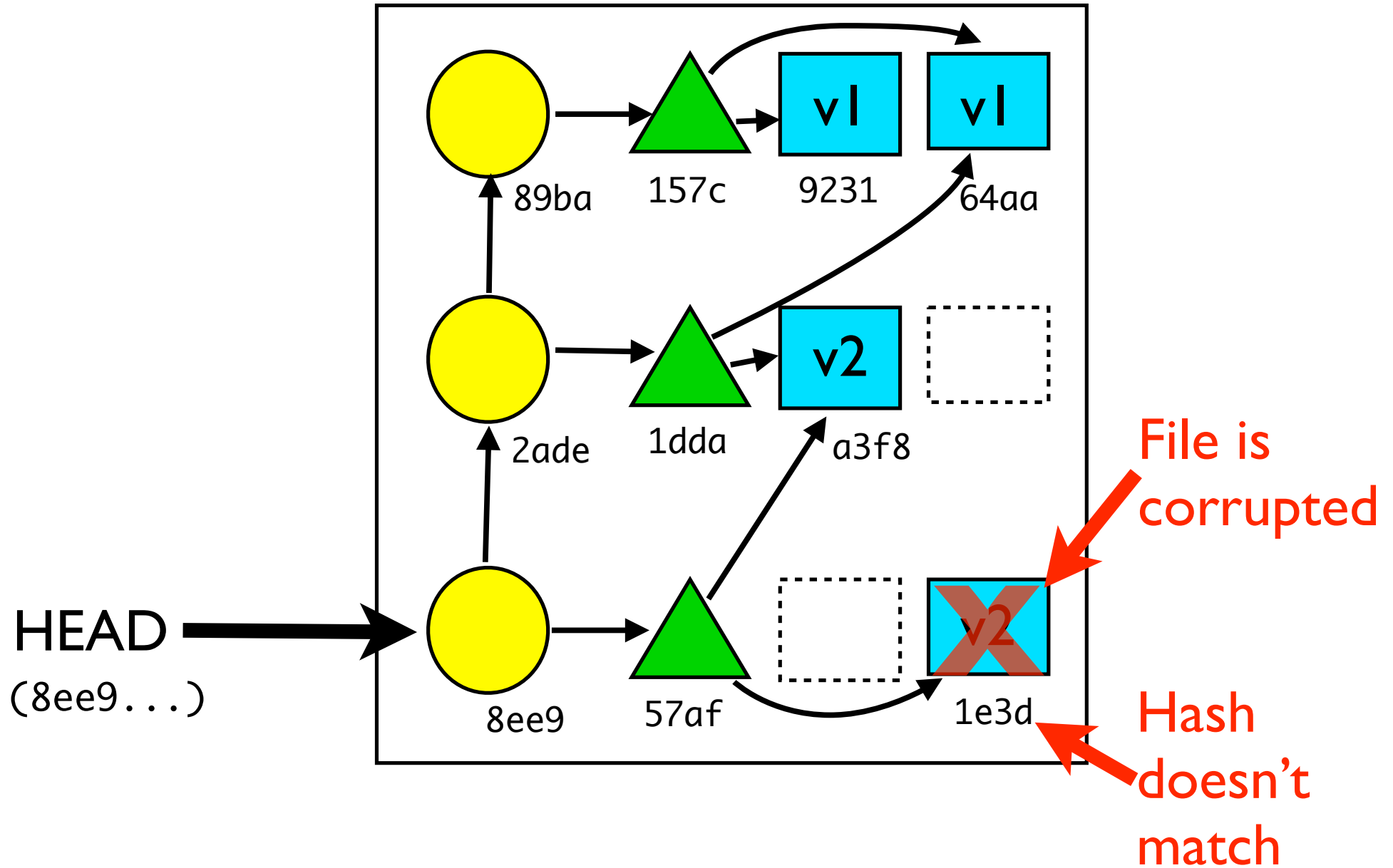


Project Archive

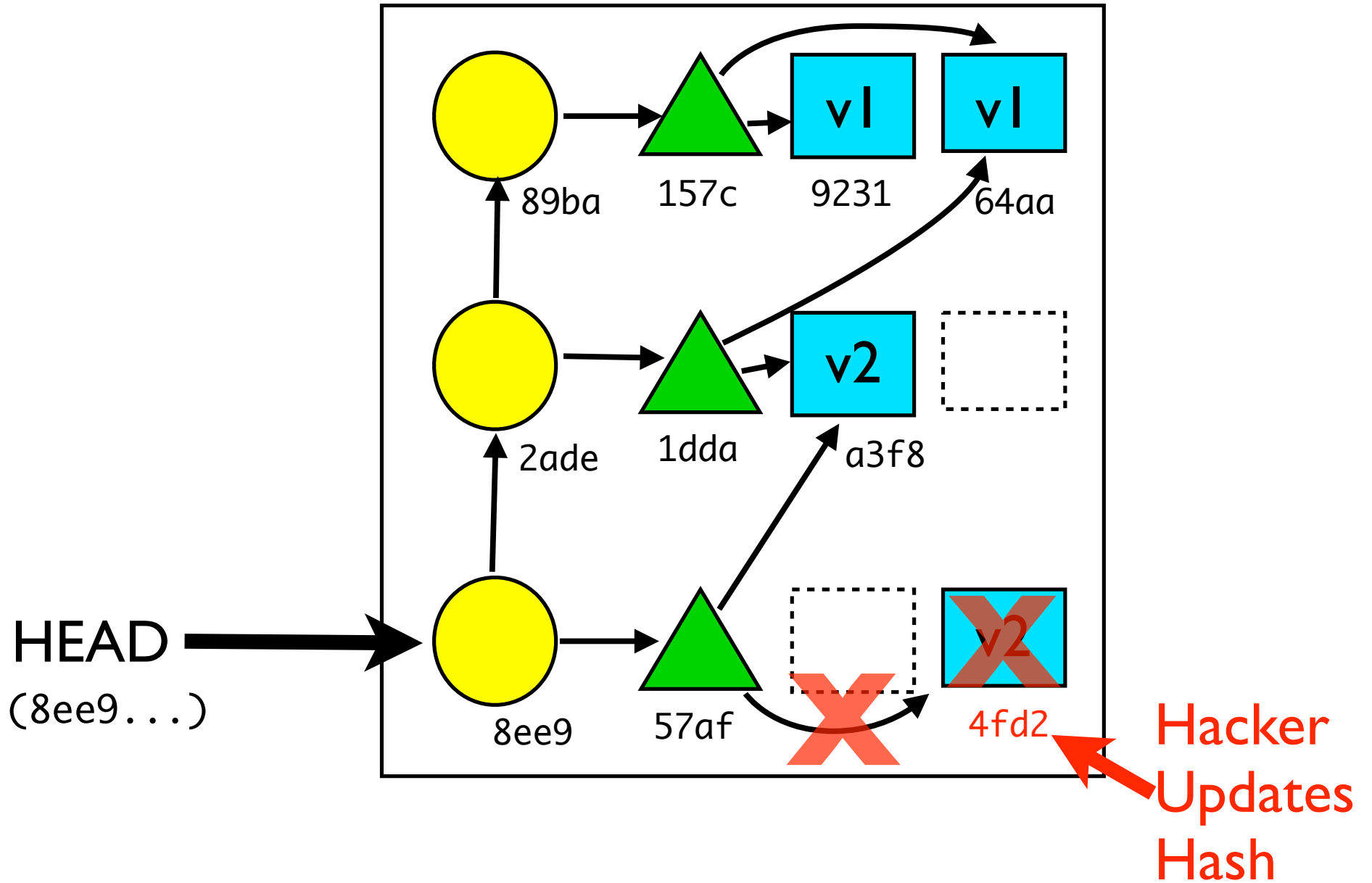




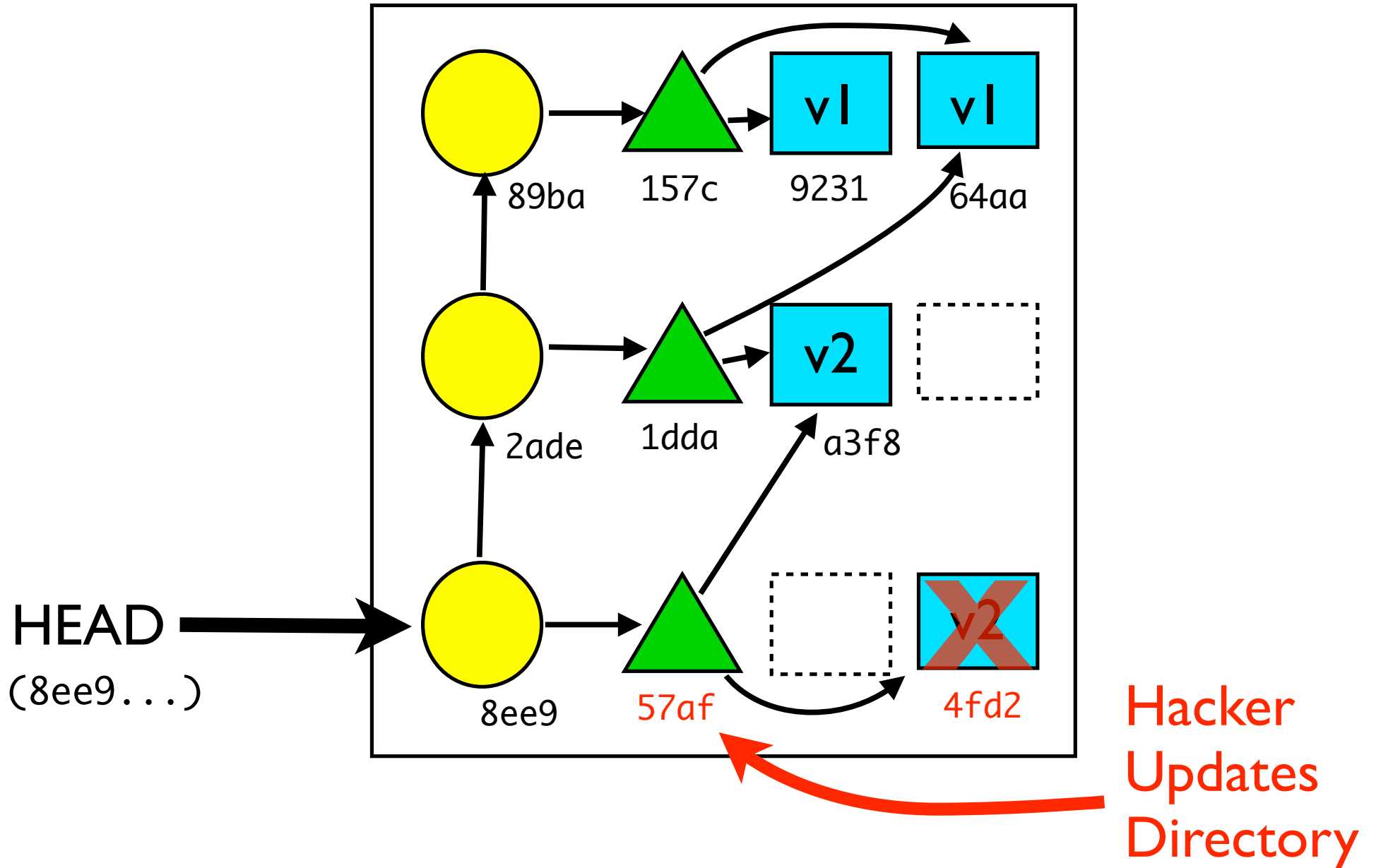
Project Archive



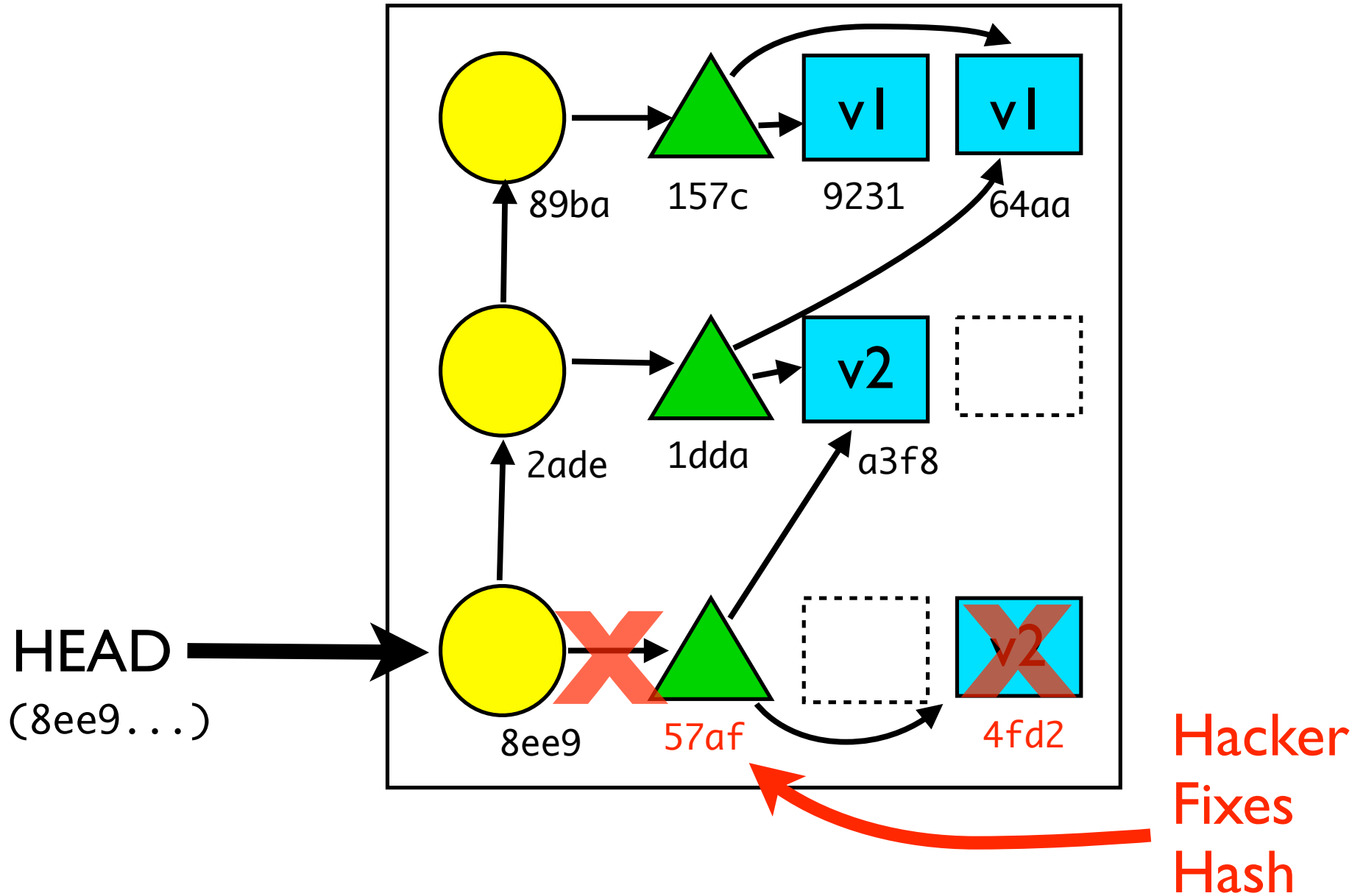
Project Archive



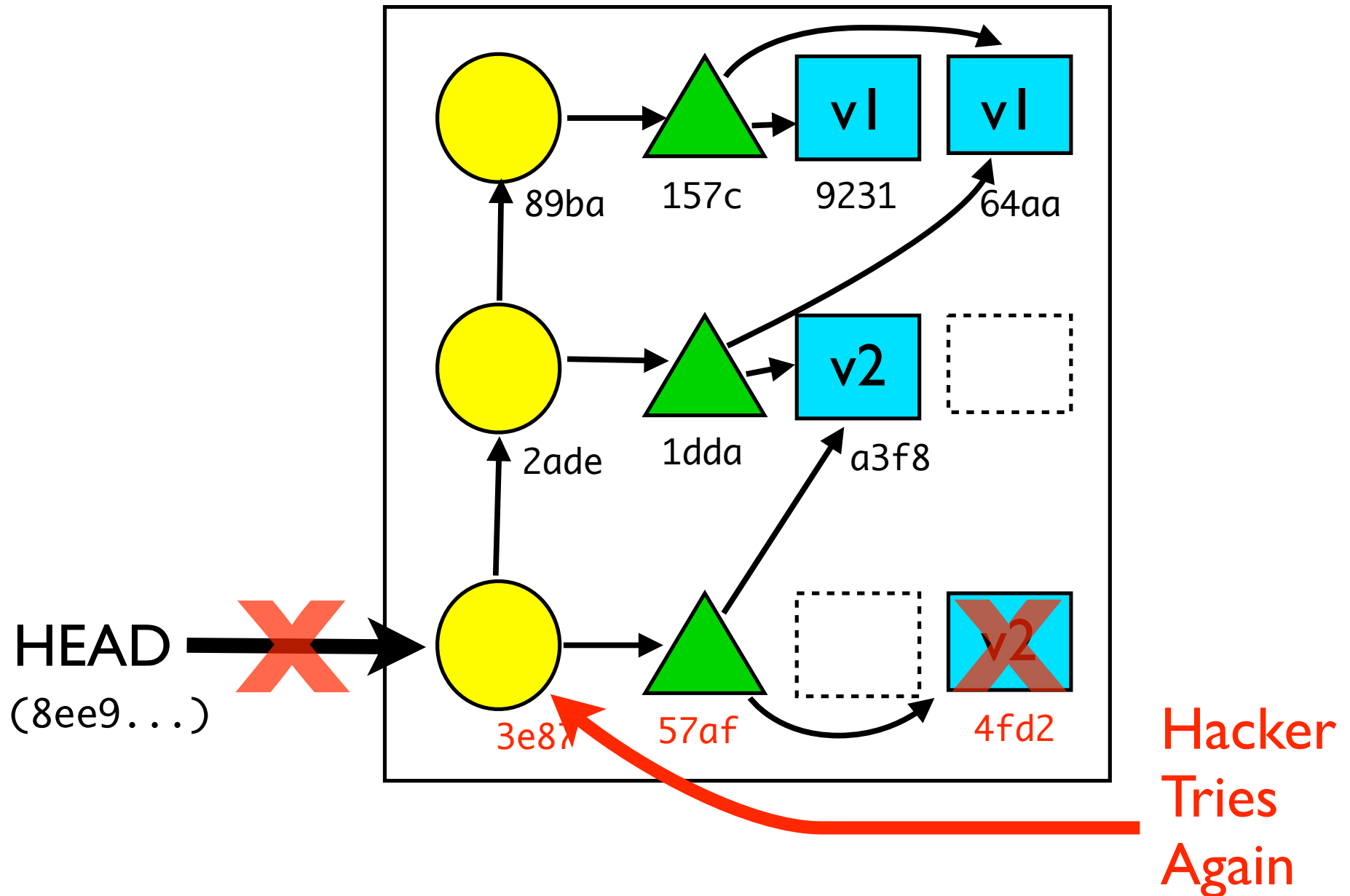
Project Archive



Project Archive

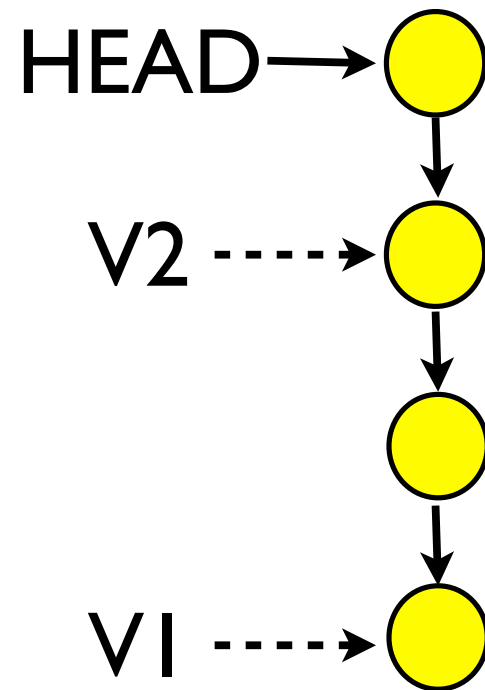


Project Archive

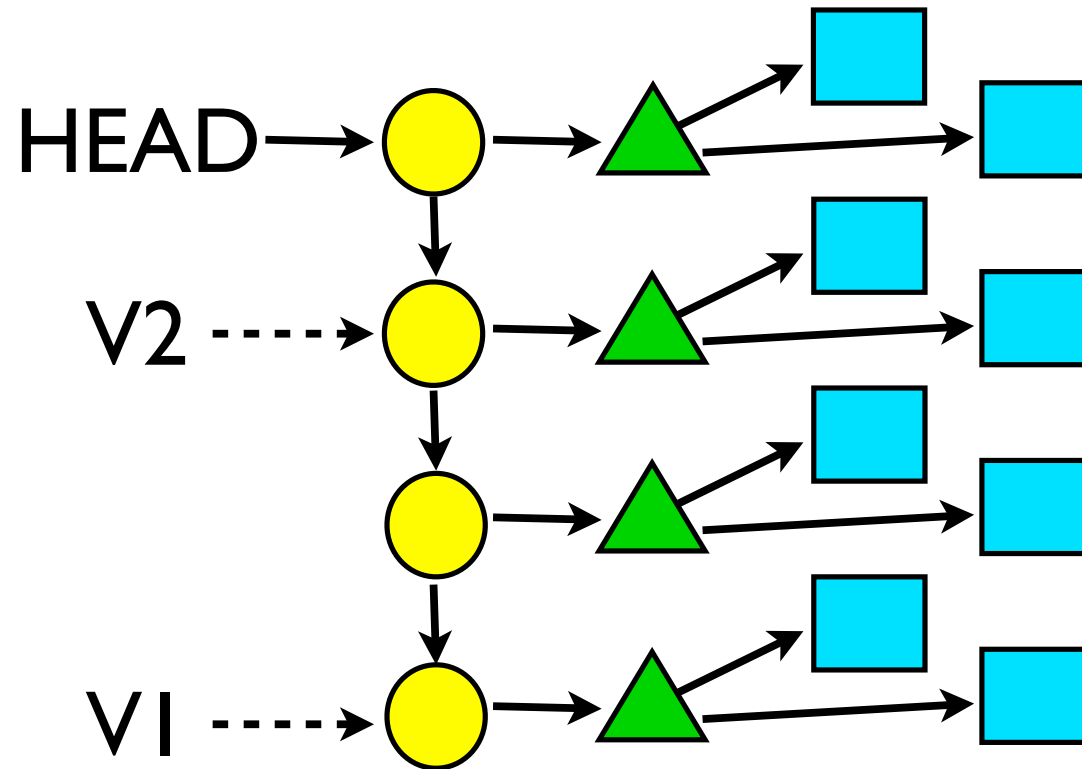


Tracking Snapshots

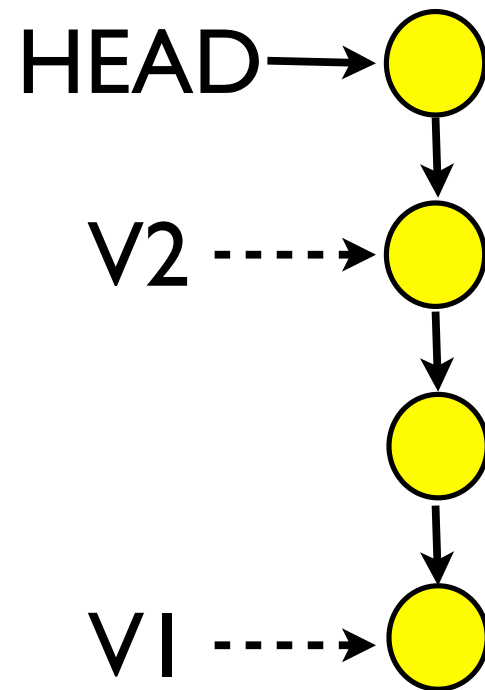
Project Archive (snapshots only)



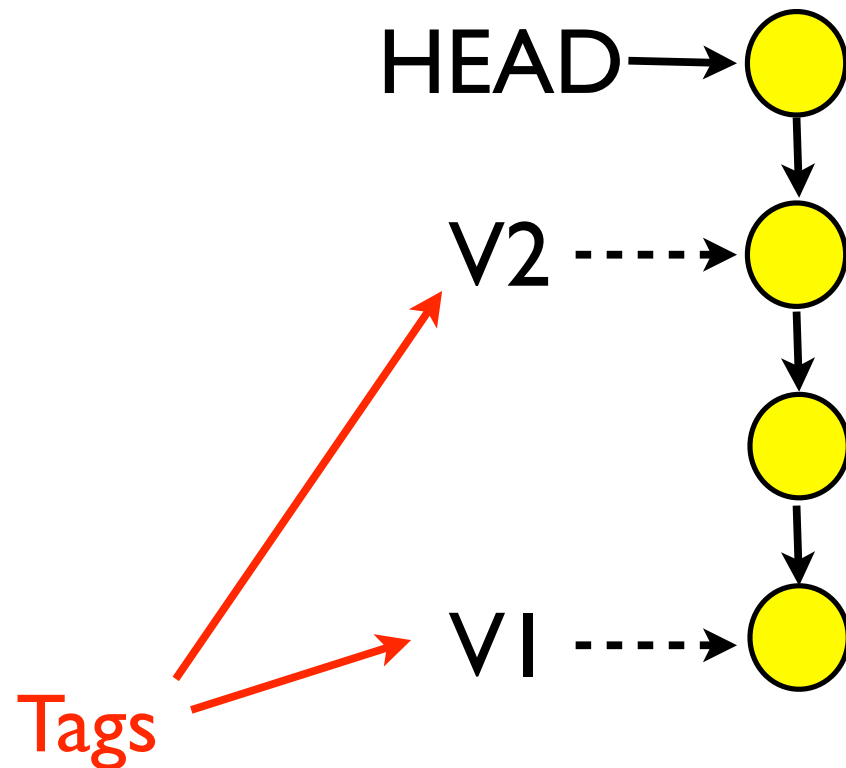
Project Archive



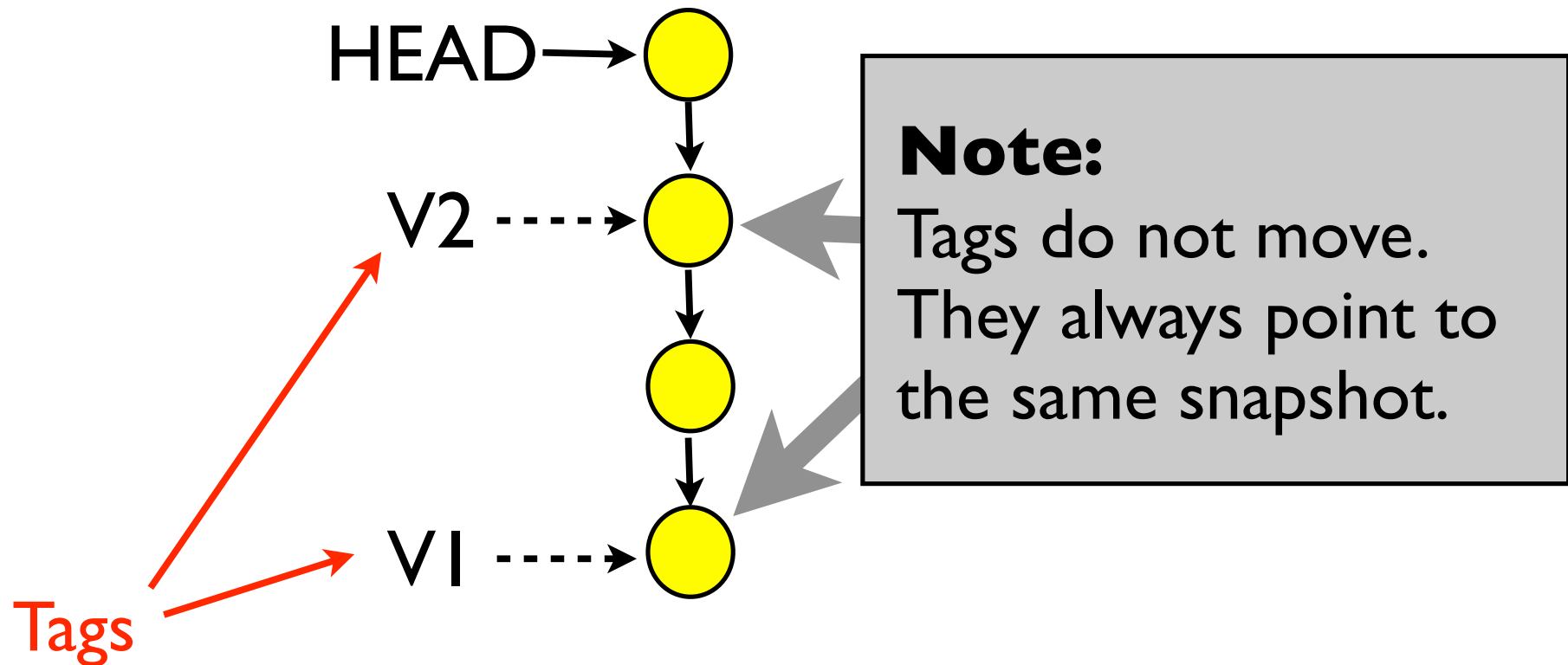
Project Archive (snapshots only)



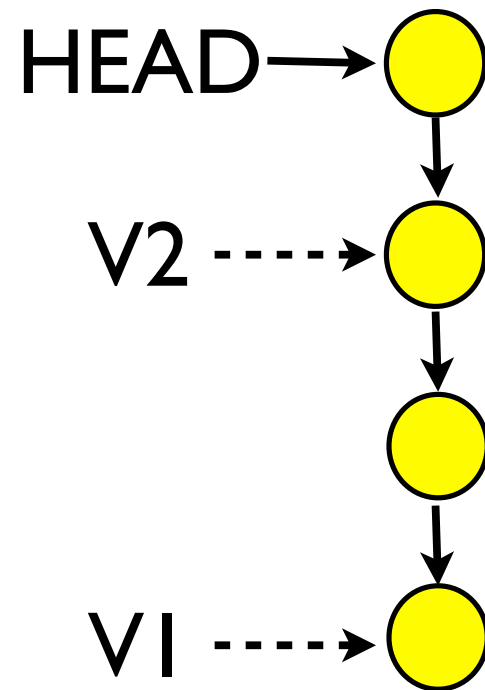
Project Archive (snapshots only)



Project Archive (snapshots only)

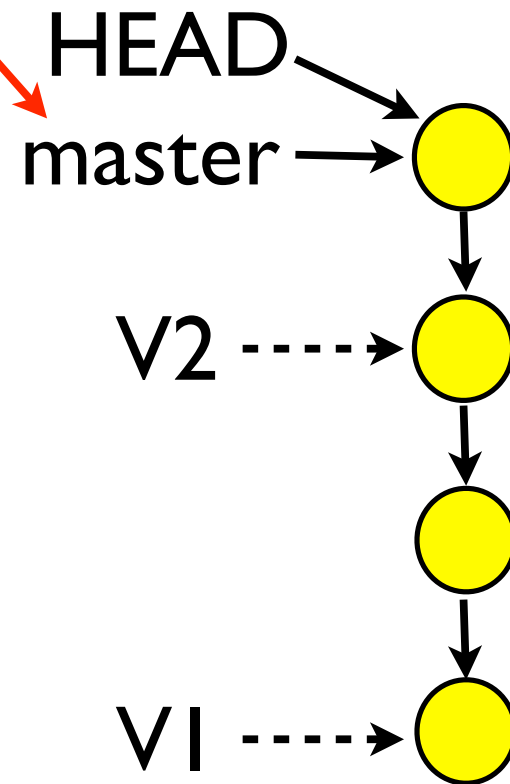


Project Archive (snapshots only)



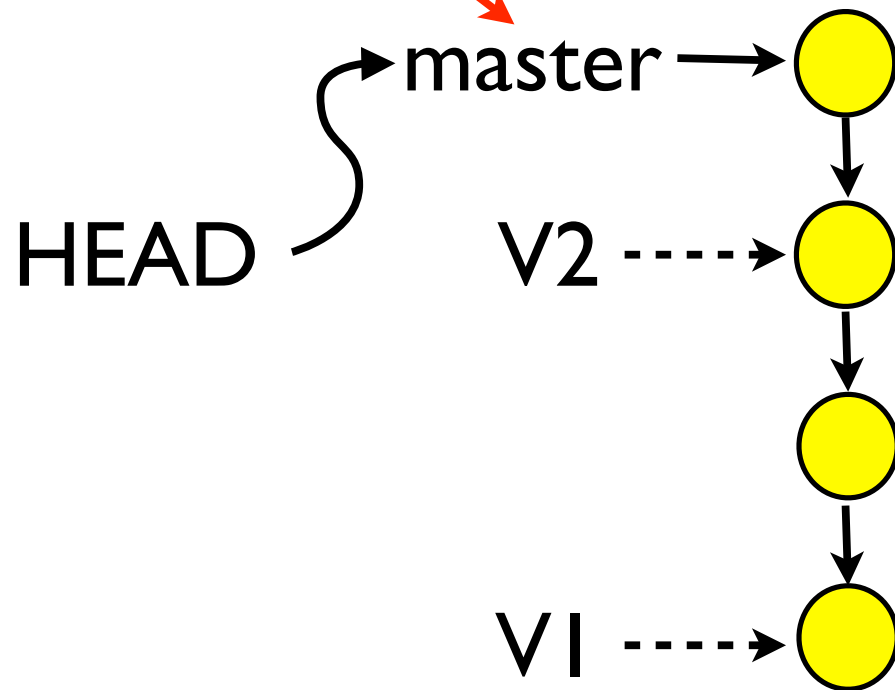
Project Archive (snapshots only)

Branch

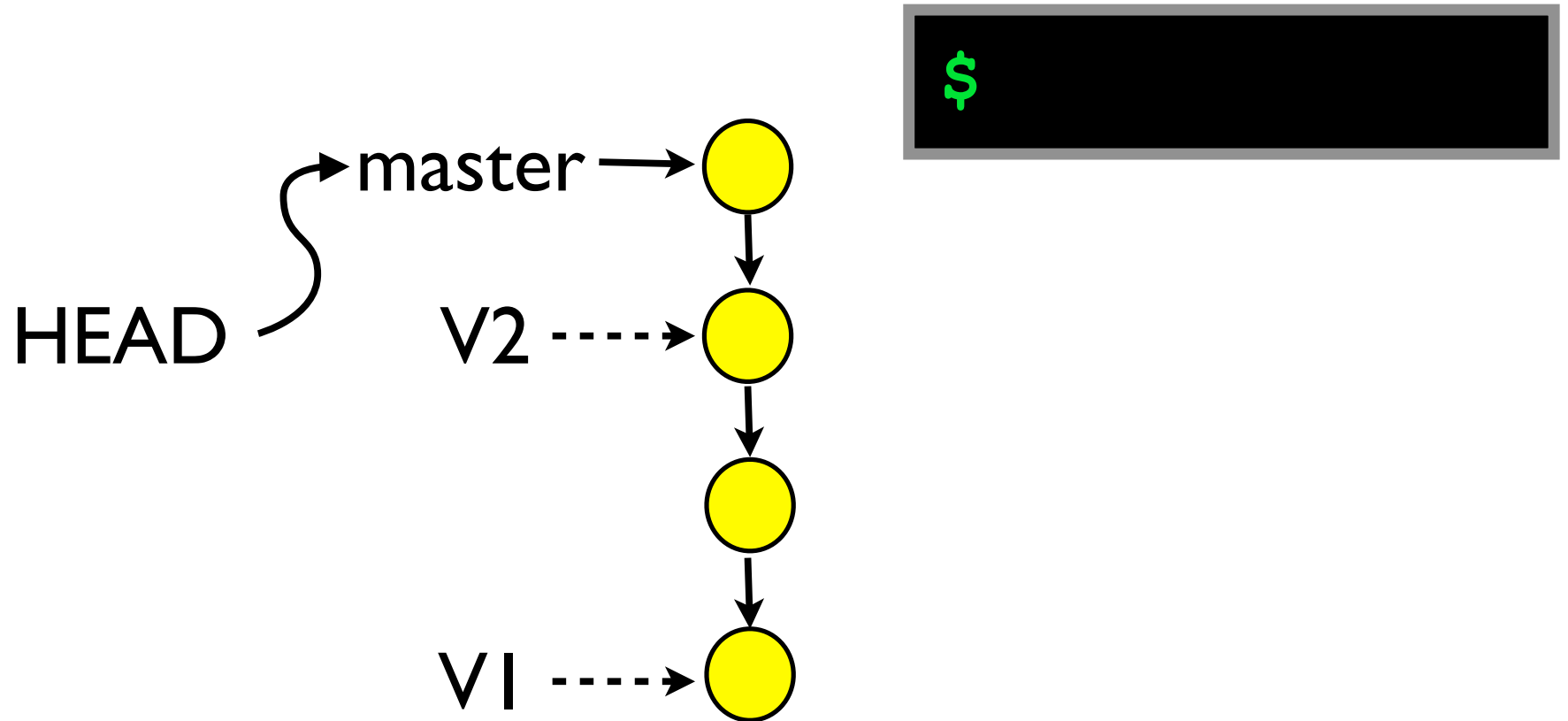


Project Archive (snapshots only)

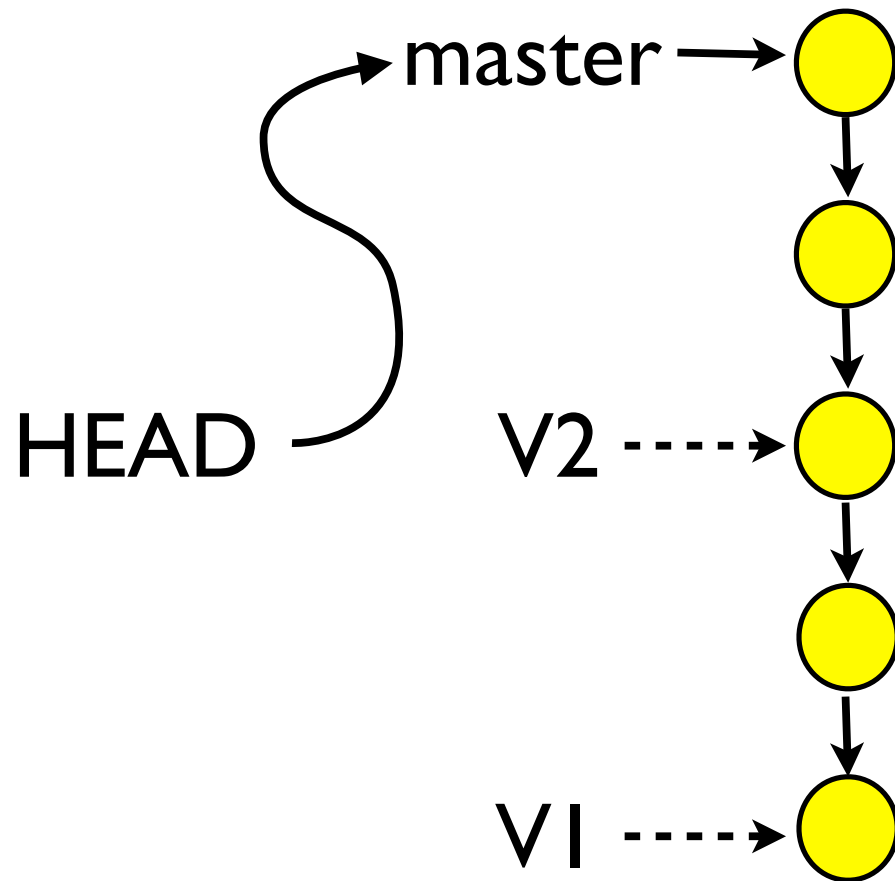
Branch



Project Archive (snapshots only)

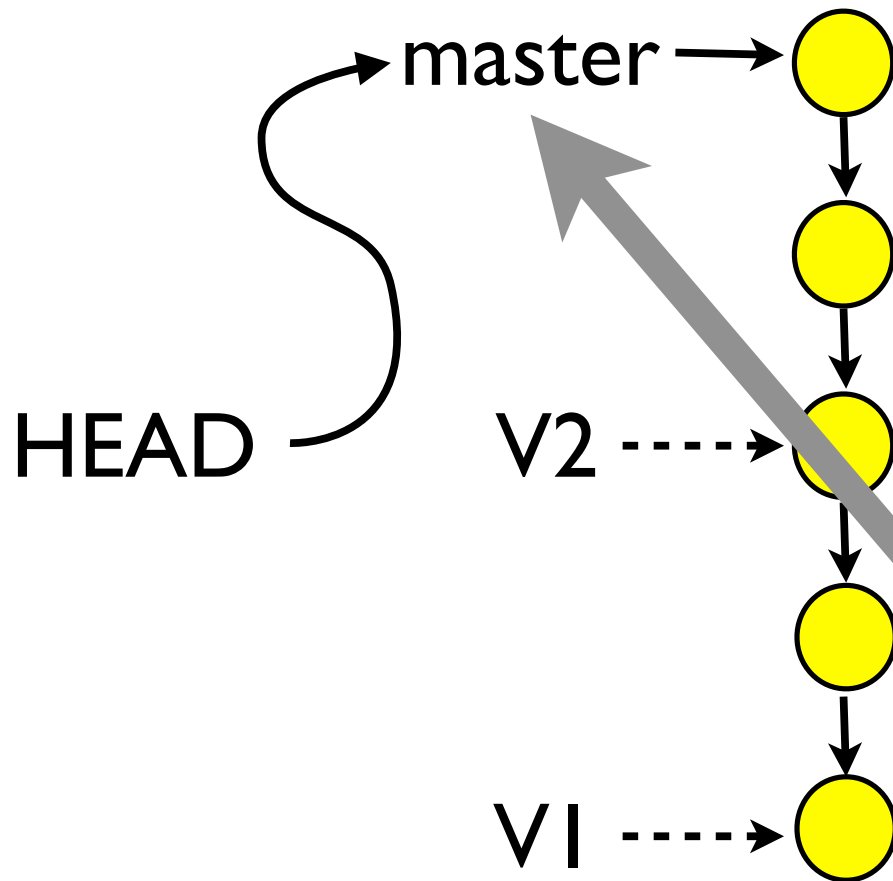


Project Archive (snapshots only)



```
$ csc snap
```


Project Archive (snapshots only)

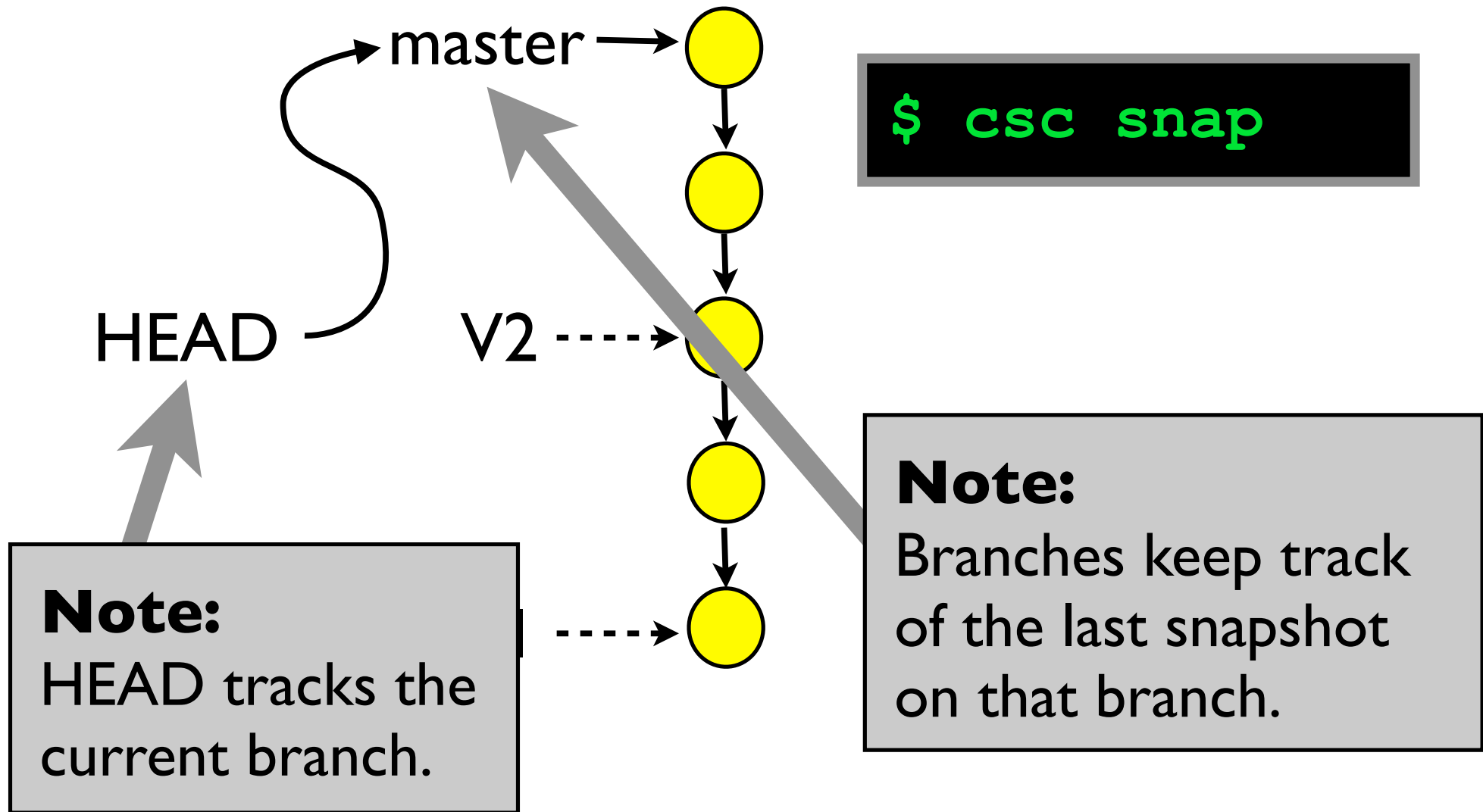


```
$ csc snap
```

Note:

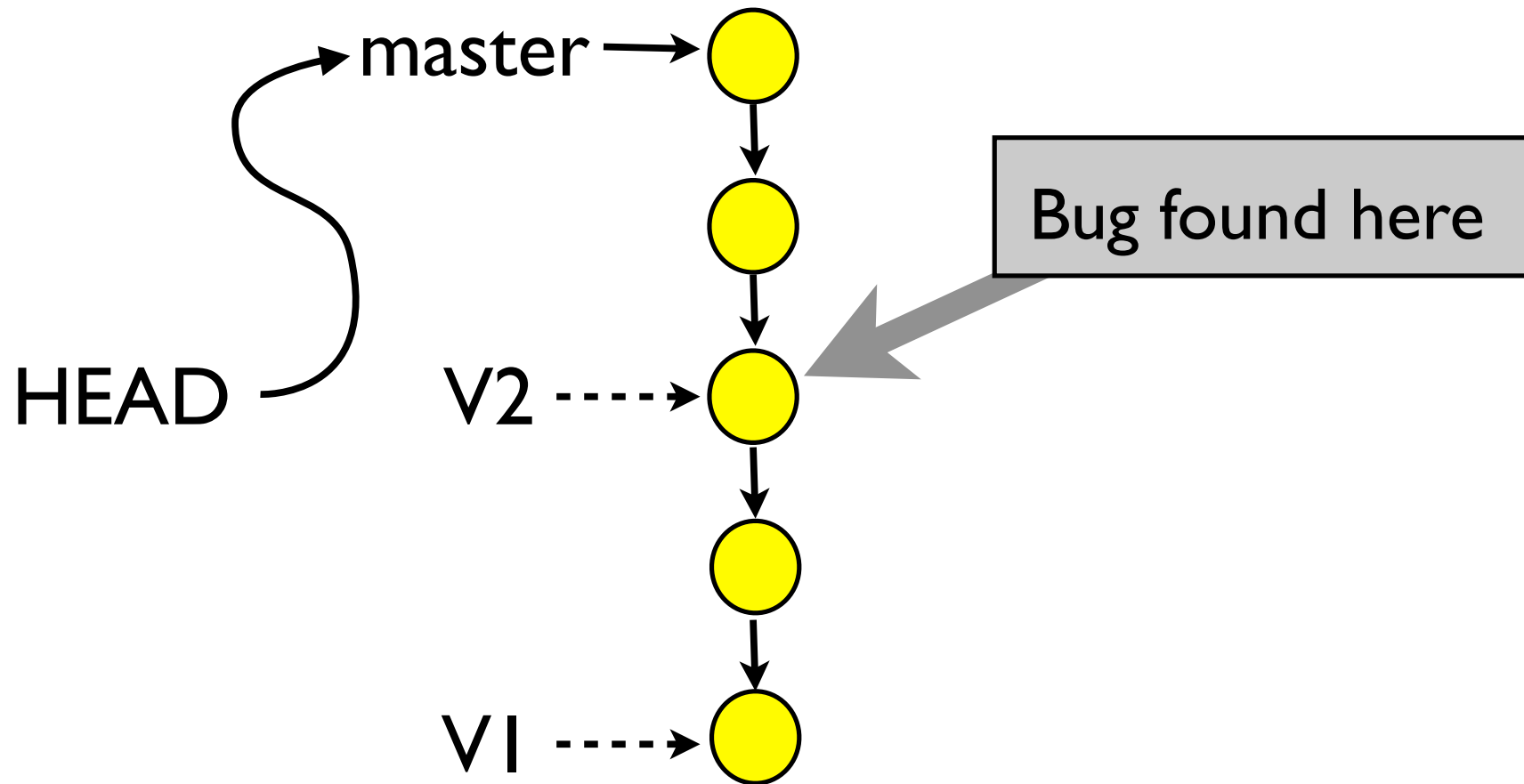
Branches keep track of the last snapshot on that branch.

Project Archive (snapshots only)

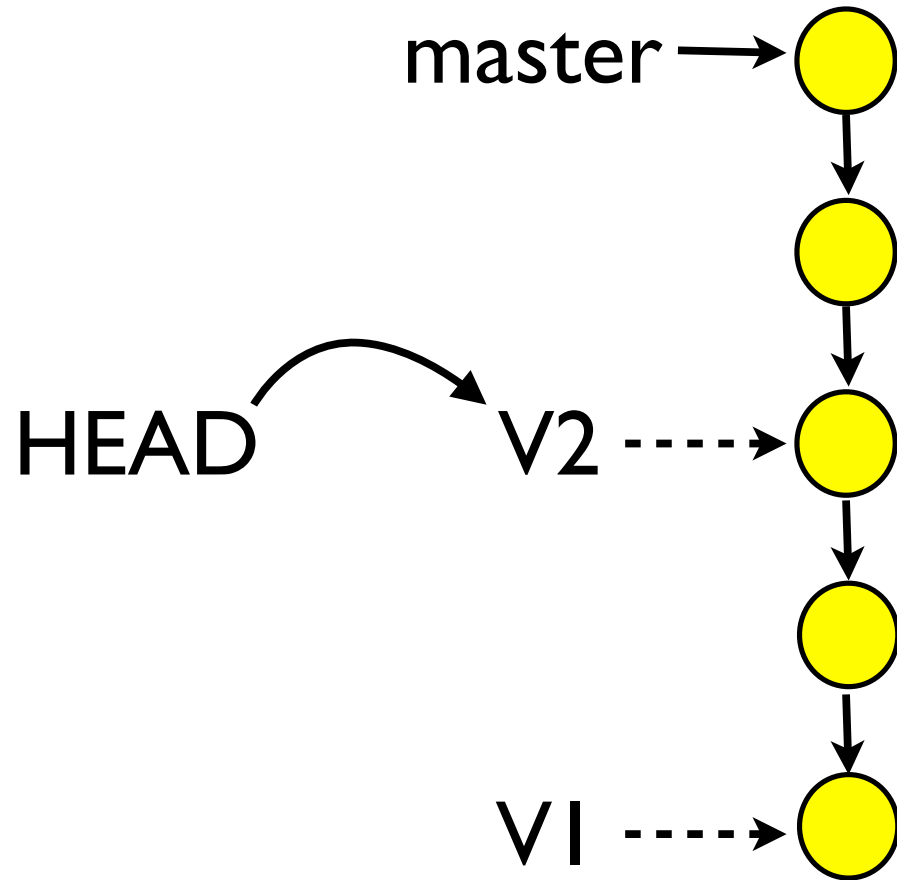


Oops! We found a bug

Project Archive

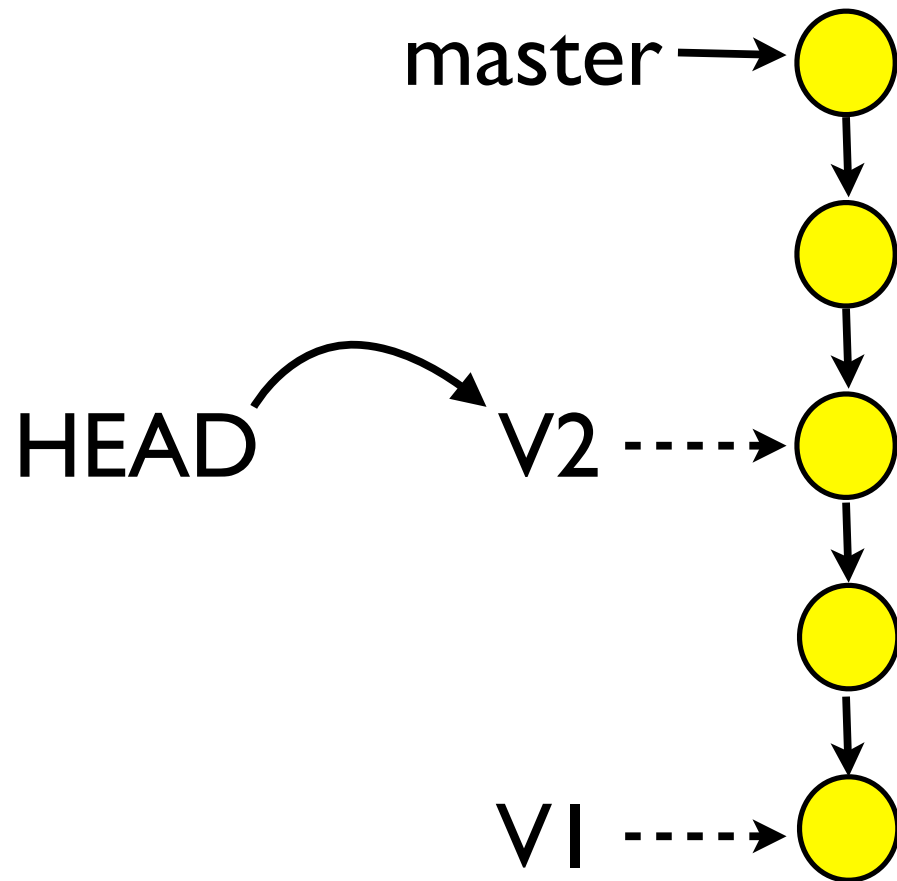


Project Archive



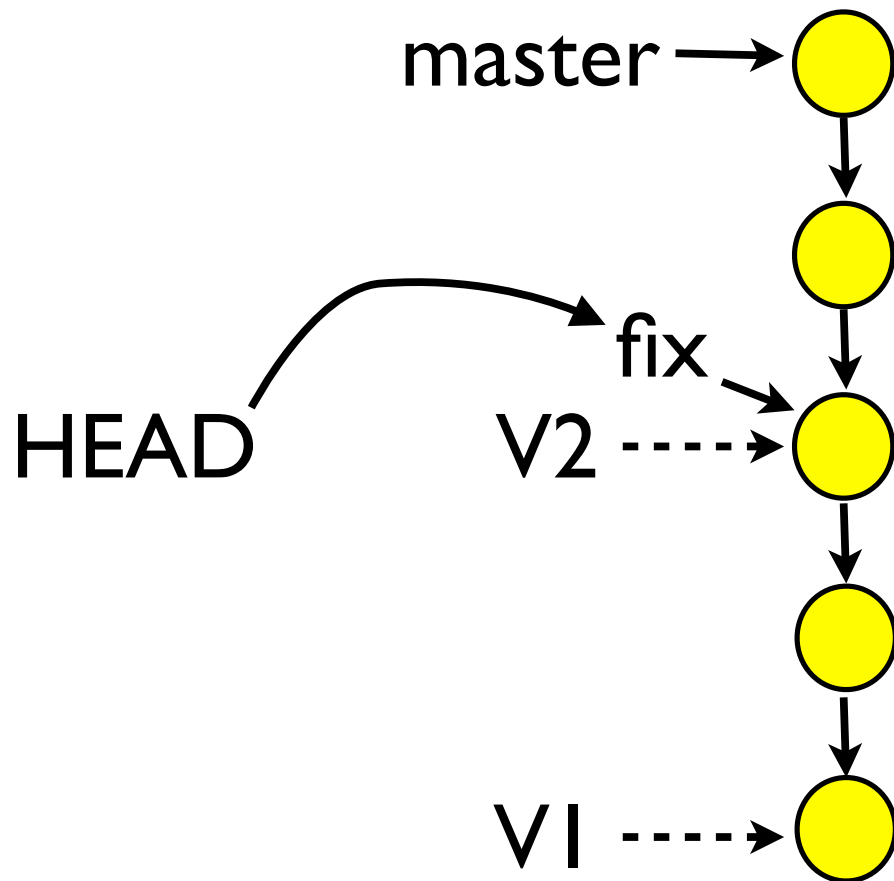
```
$ csc checkout V2
```


Project Archive



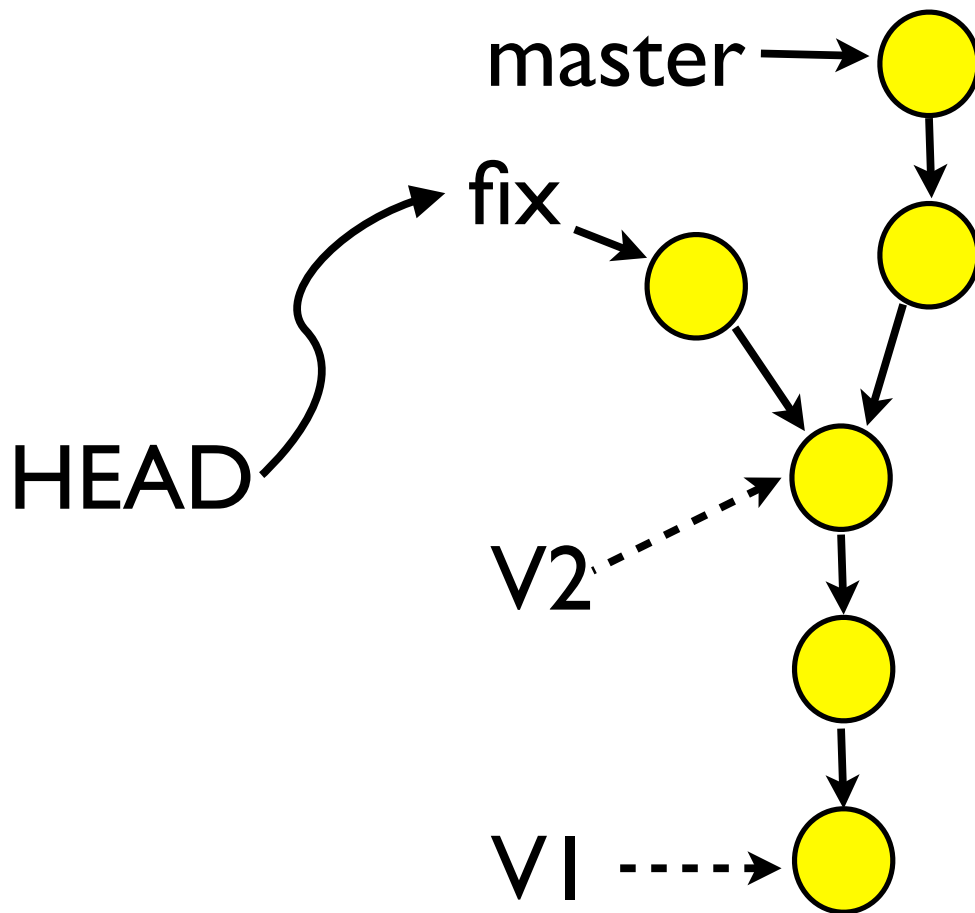
```
$ csc checkout V2  
$ edit files
```

Project Archive



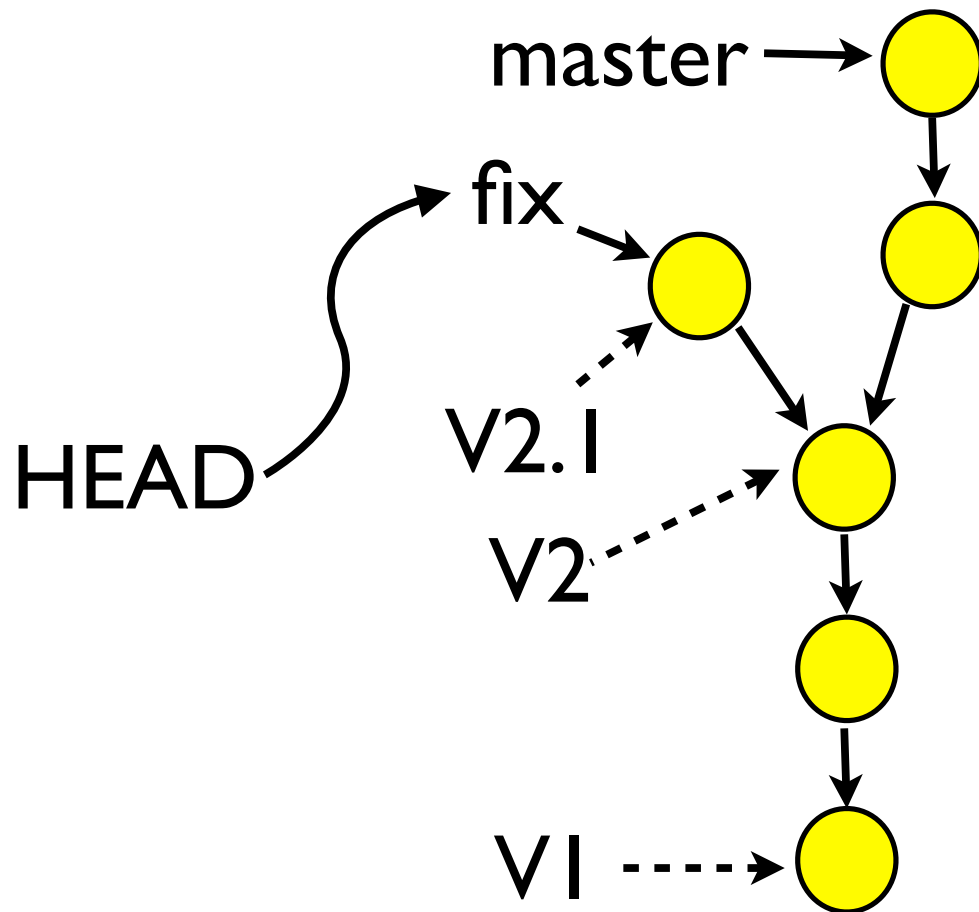
```
$ csc checkout V2  
$ edit files  
$ csc branch fix
```

Project Archive



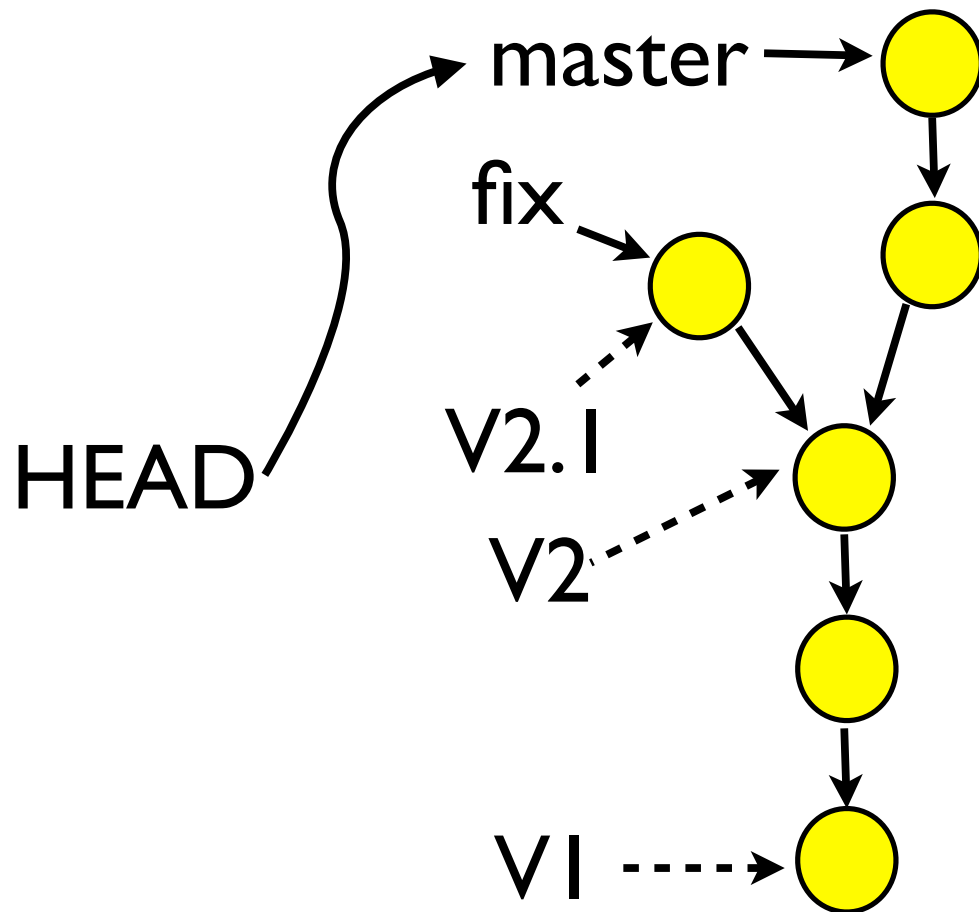
```
$ csc checkout V2  
$ edit files  
$ csc branch fix  
$ csc snap
```

Project Archive



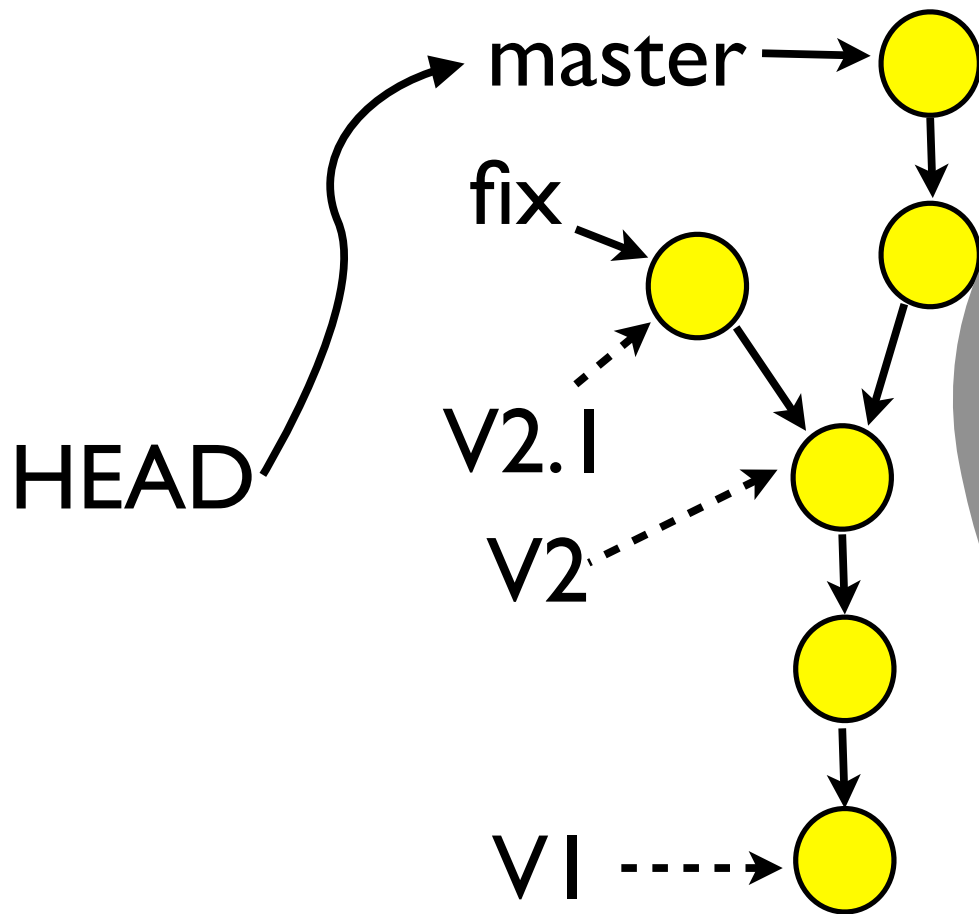
```
$ csc checkout V2
$ edit files
$ csc branch fix
$ csc snap
$ csc tag V2.1
```

Project Archive



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


Project Archive



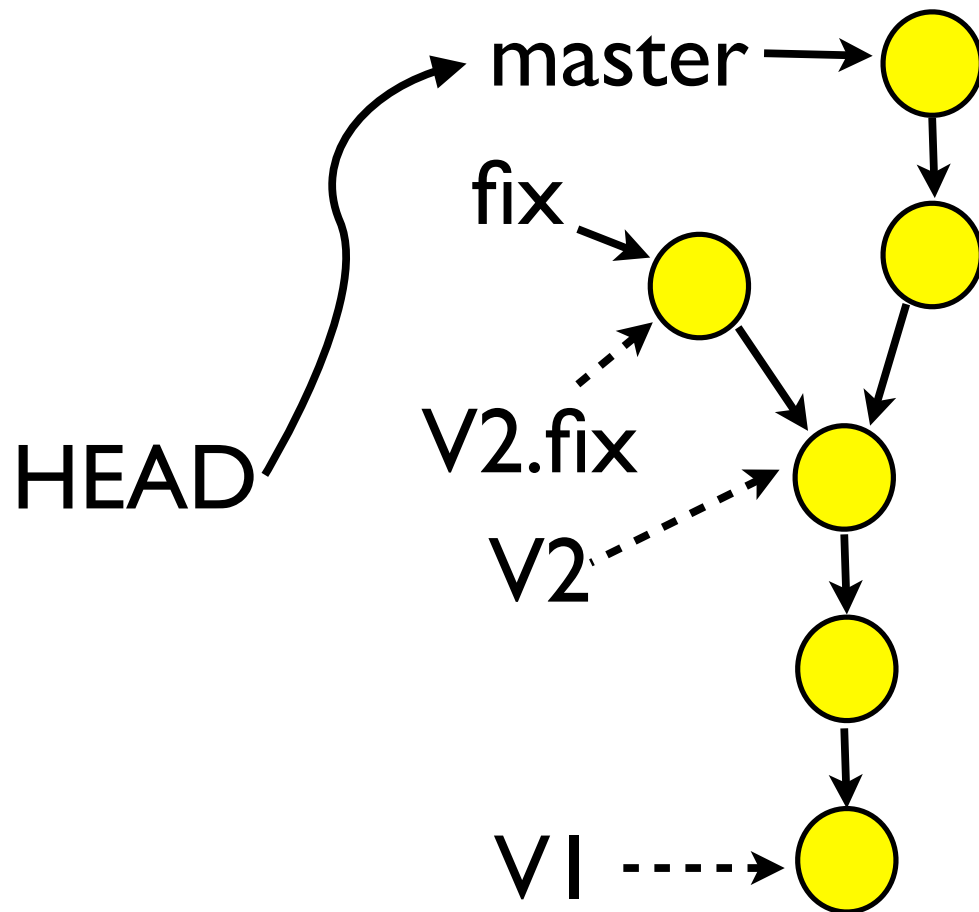
```
$ csc checkout V2
$ edit files
$ csc branch fix
$ csc snap
$ csc tag V2.fix
$ csc checkout master
```

Note:

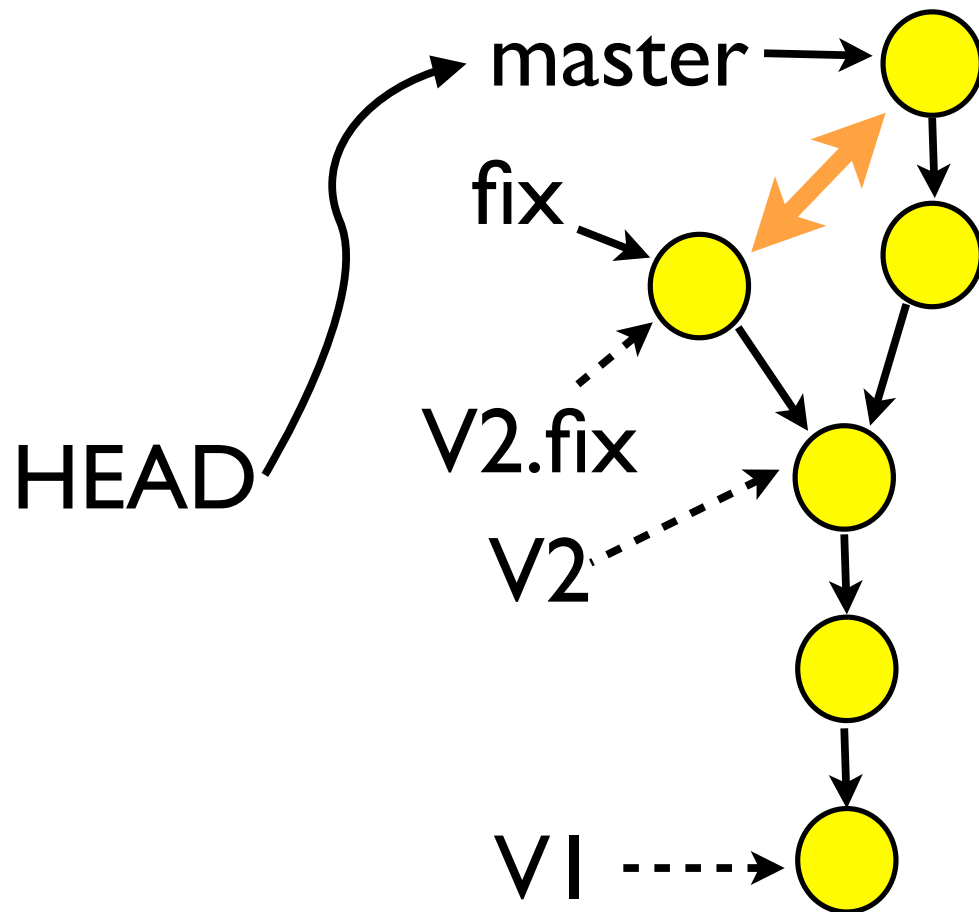
We didn't create a branch until after we edited files.

Merge Bug Fixes to
Master

Project Archive

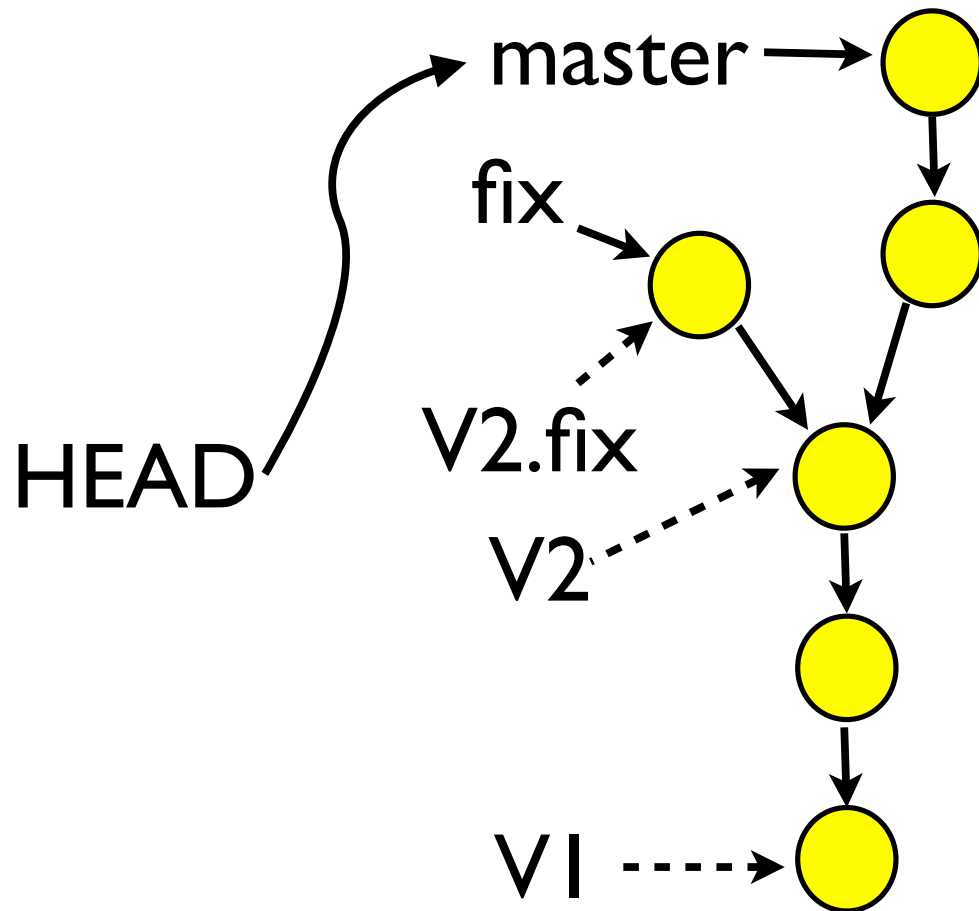


Project Archive

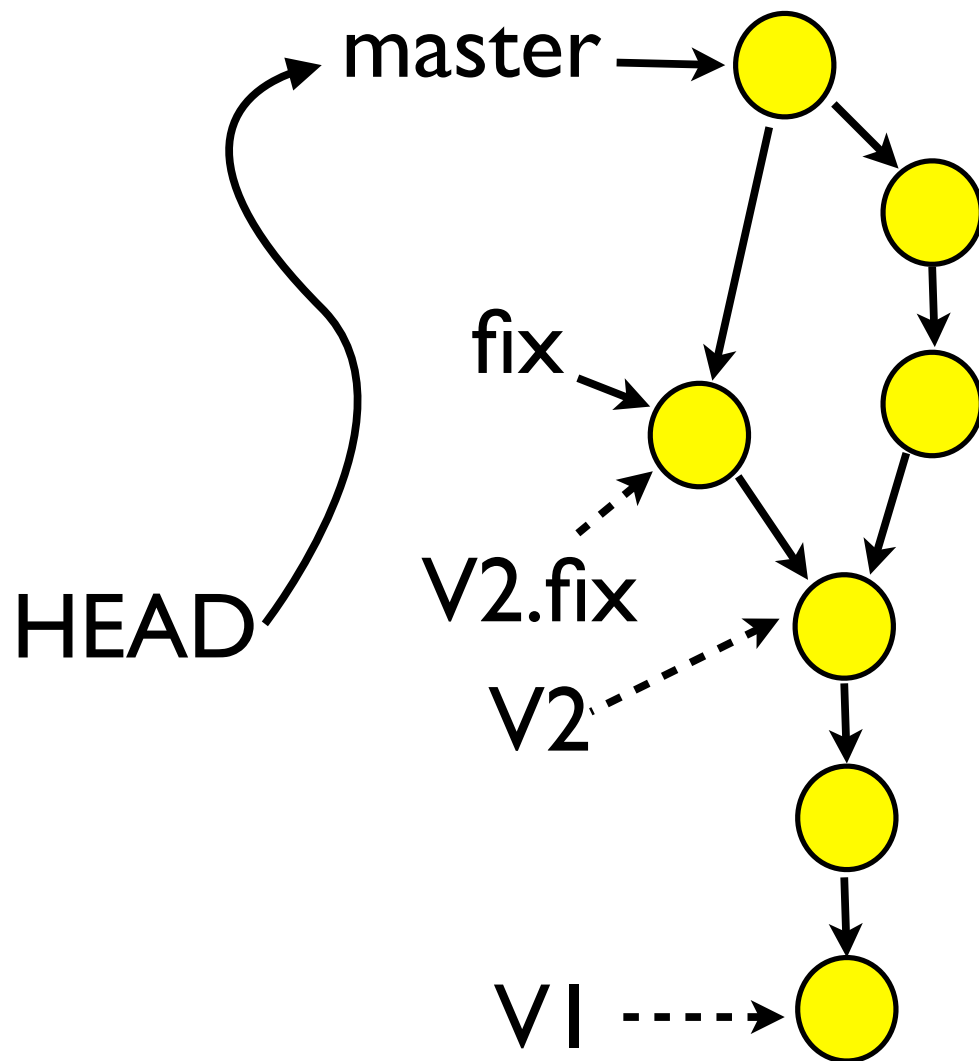


```
$ csc merge fix
```

Project Archive



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

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


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 **Created ENV Branch** to execute arbitrary Ruby code in
- Removed Rake::DSL
- Rakefiles 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_... lib will be flattened to use File::PATH_SEPARATOR
- Added -X option 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

Fix for 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 top dsl option logic.
- Top level DSL is make
- 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

Fix for master Branch

Subject

- **master** Updated CHANGES to mention the warnickr changes for test task.
- **env** **origin/env** Merge branch 'master' into env
- **warnickr** **HEAD** **origin/master** **warnickr/master** make sure rake lib is available
- Avoid command line too long, push
- **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

Merged master to env

Subject

- **master** Updated CHANGES to mention the warnickr changes for test task.
- **env** **origin/env** Merge branch 'env' into 'master'
- **warnickr** **origin/HEAD** **origin/master** **warnickr** Warnickr 0.8.99.2 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

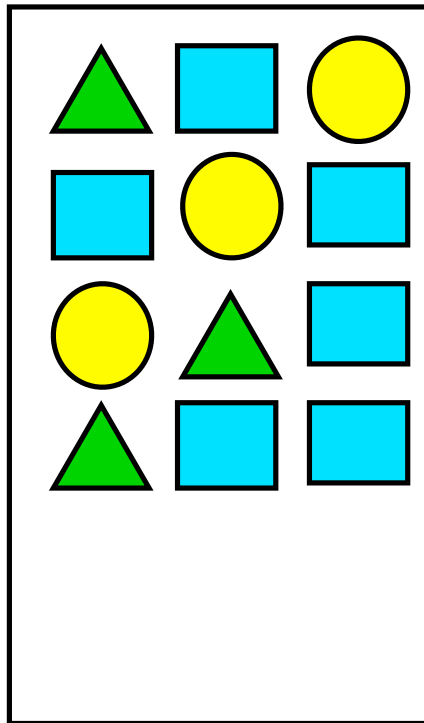
Latest master Version

Subject

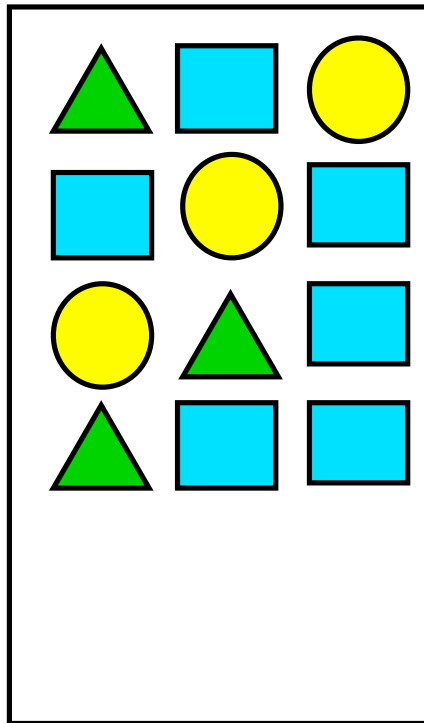
- **master** Updated CHANGES to mention the warnickr changes for test task.
- **env** **origin/env** Merge branch 'master' into env
- **warnickr** **origin/HEAD** **origin/master** **warnickr/master** make sure rake lib is available
- Avoid command line too long problem when running large numbers c
- **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

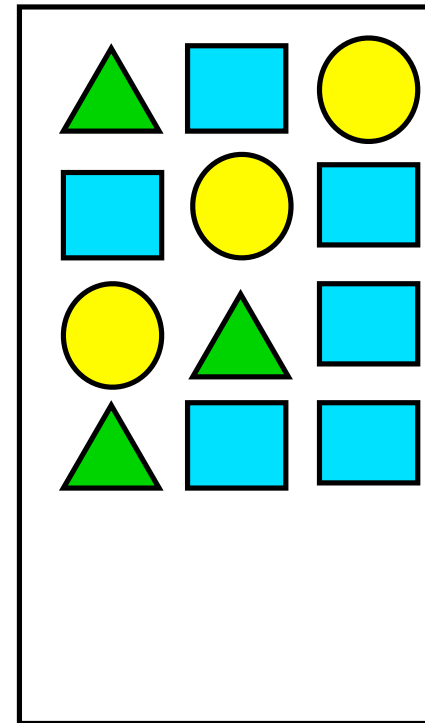
My Archive



My Archive



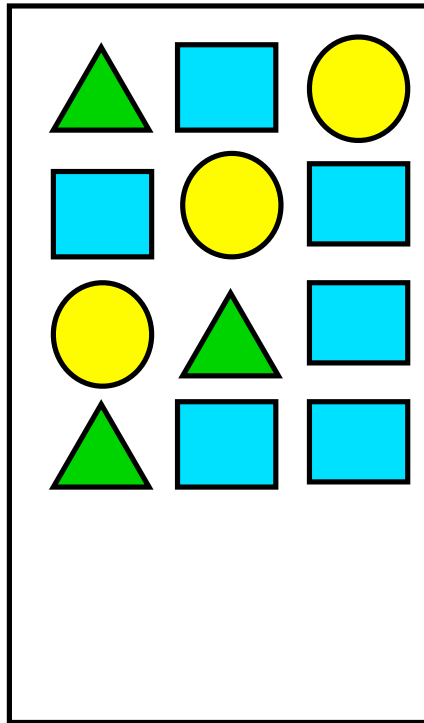
Your Archive



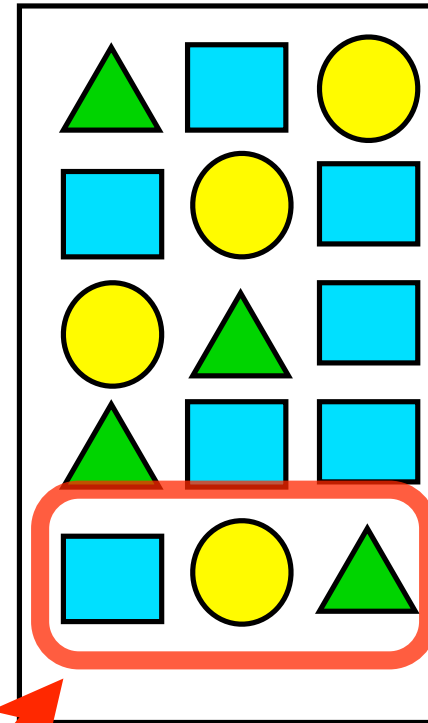
clone

```
you$ csc clone url_for_my_repo
```

My Archive

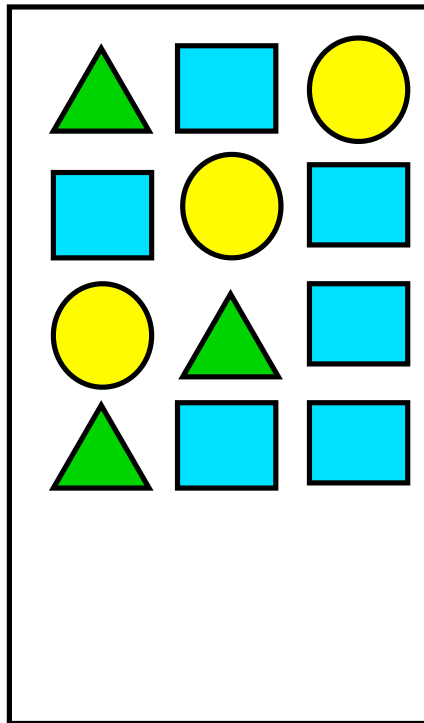


Your Archive

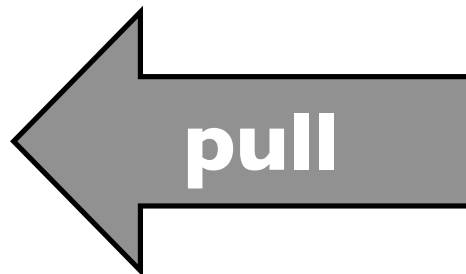
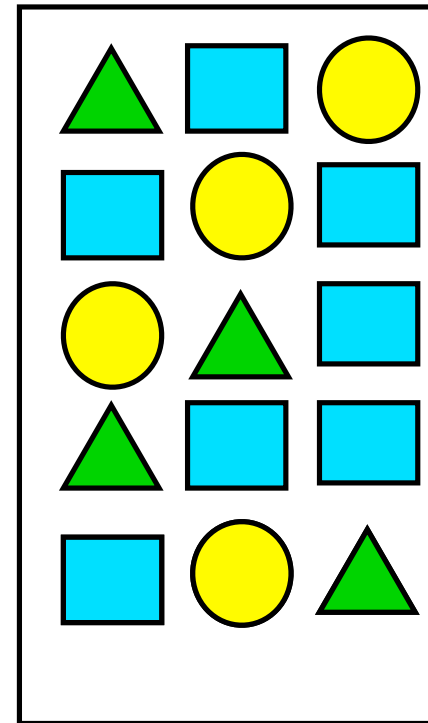


You make changes

My Archive

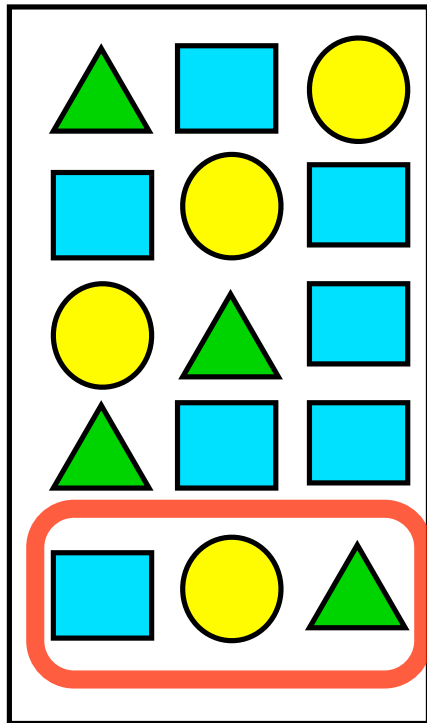


Your Archive

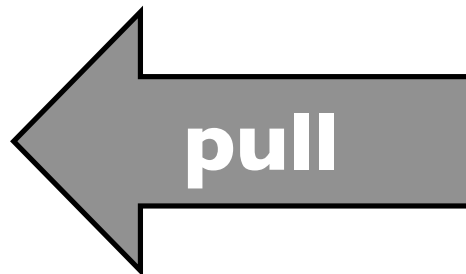
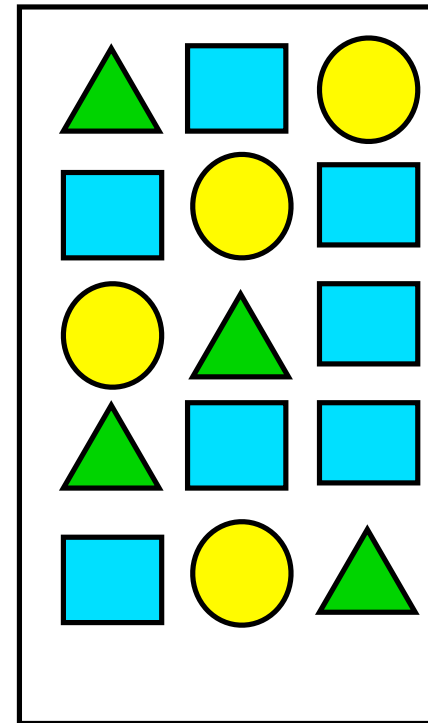


```
me$ csc pull url_for_my_repo
```

My Archive

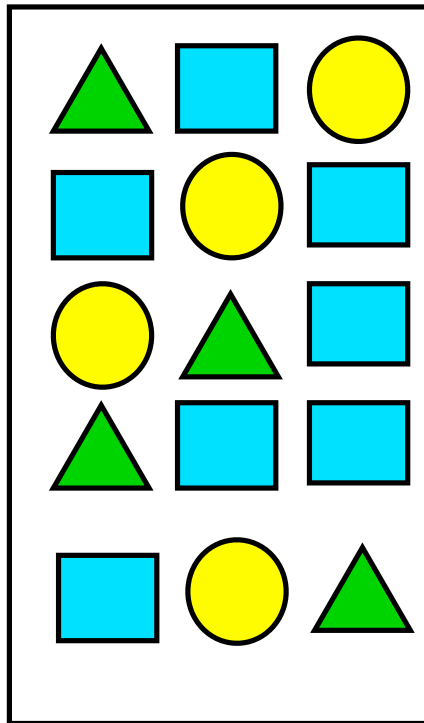


Your Archive

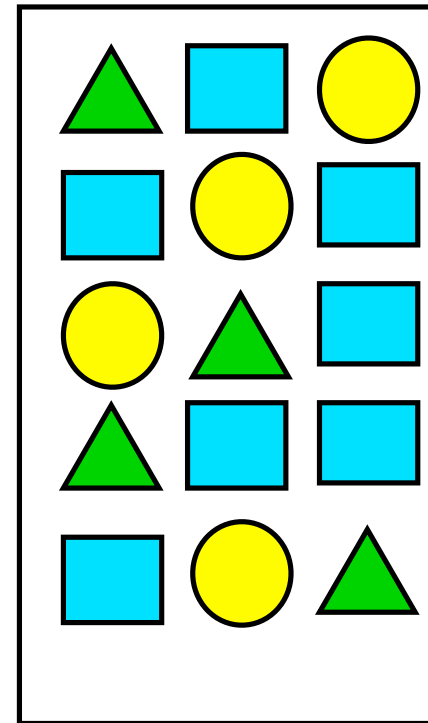


```
me$ csc pull url_for_my_repo
```


My Archive



Your Archive

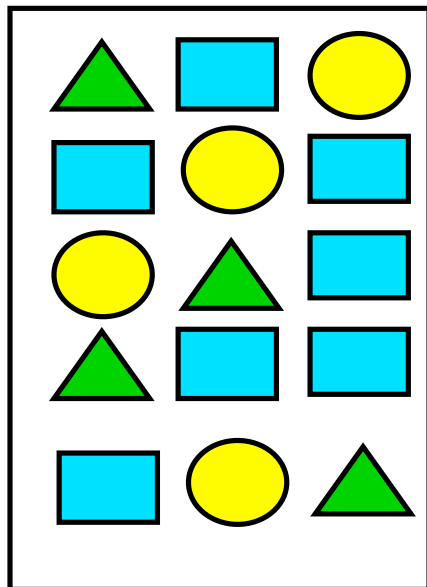


pull

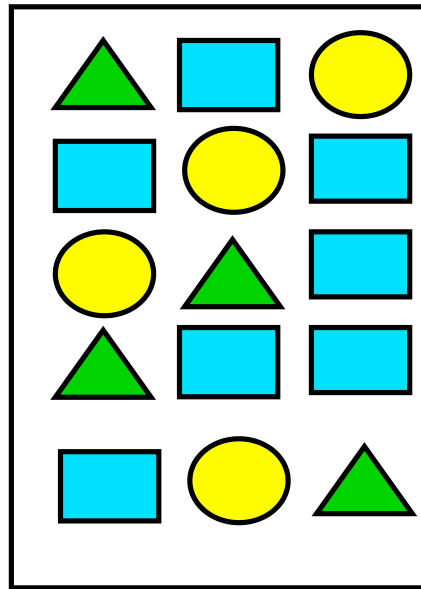
```
you$ csc pull url_for_my_repo
```

Common Remote Uses

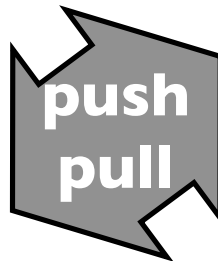
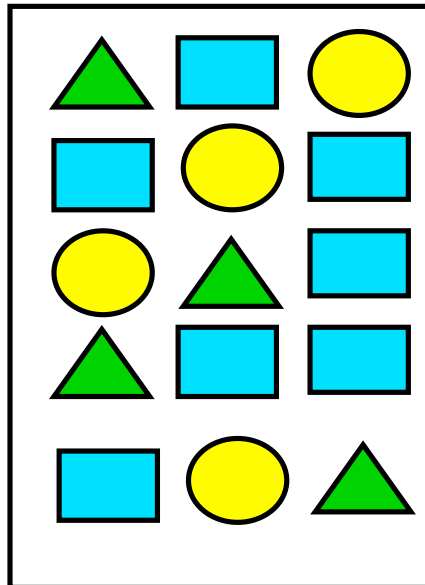
My Archive



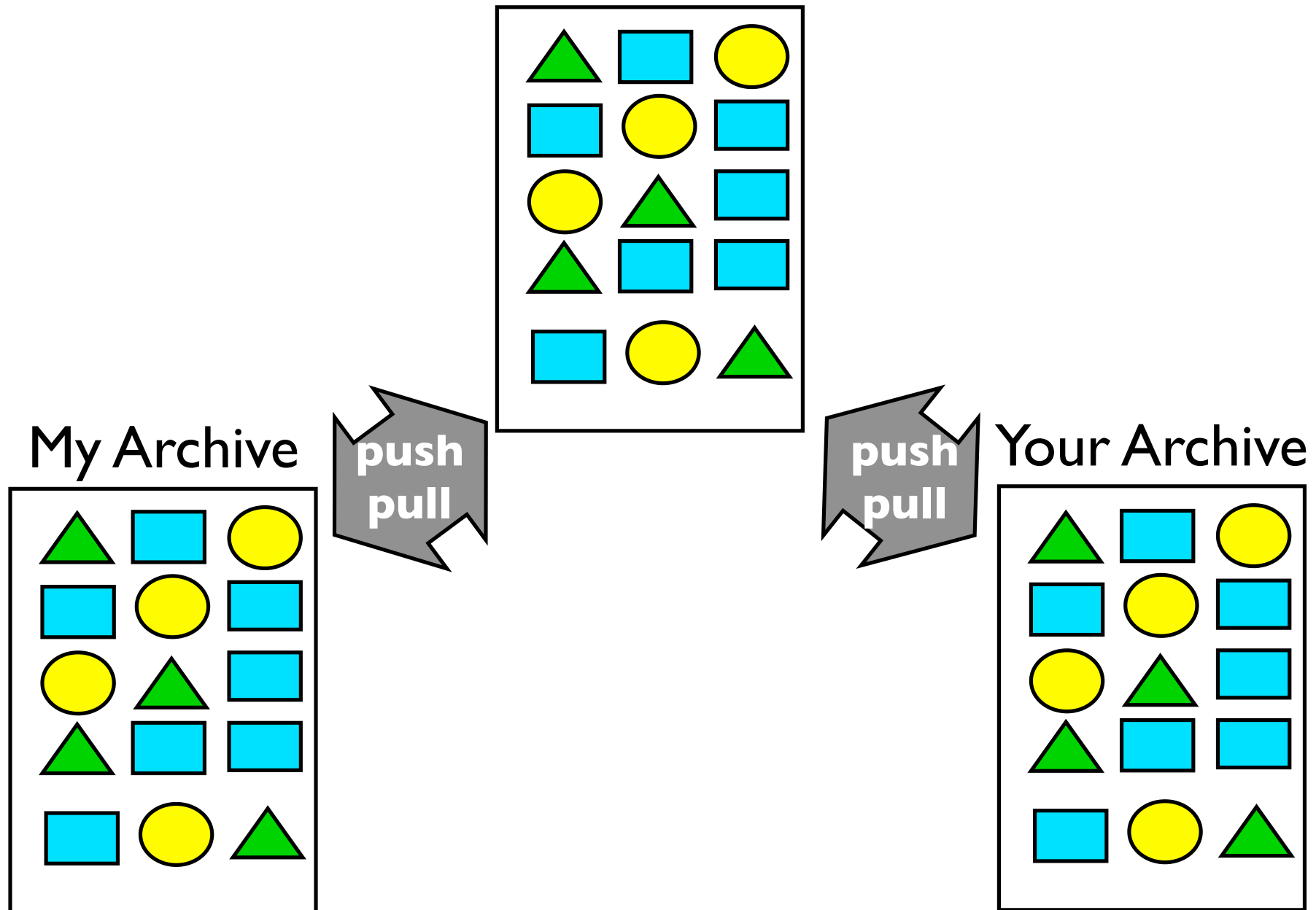
Shared Remote Archive



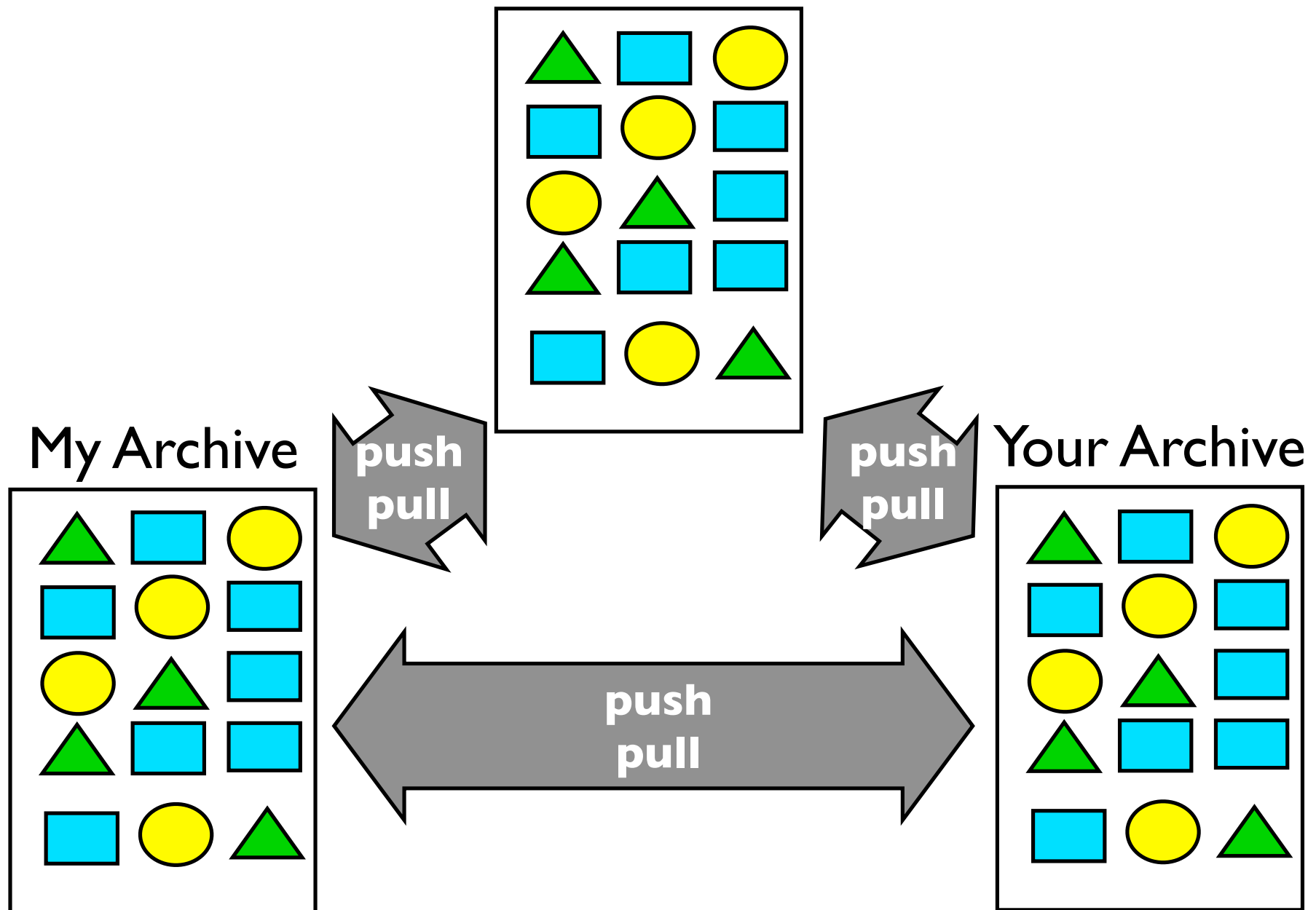
My Archive

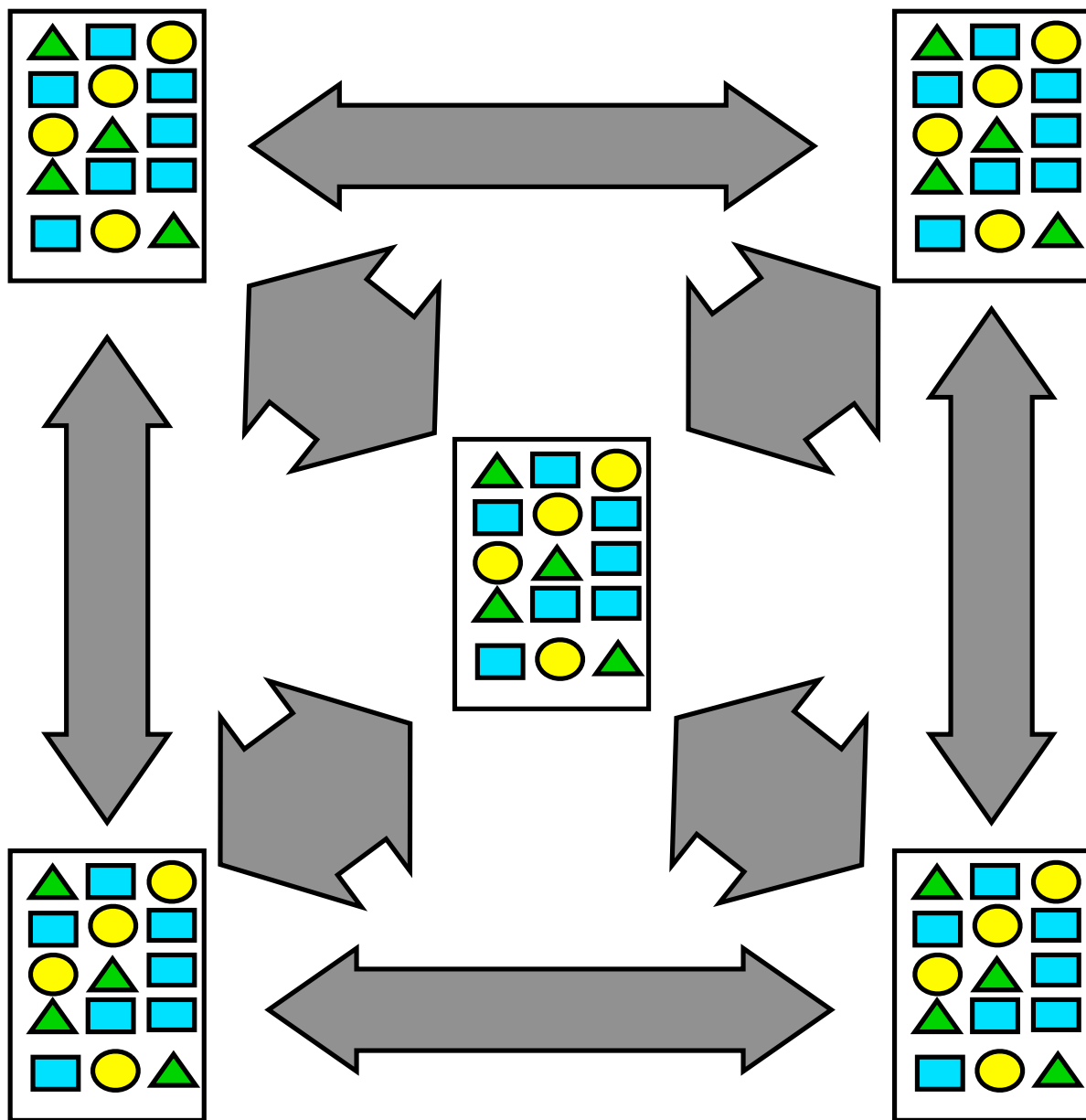


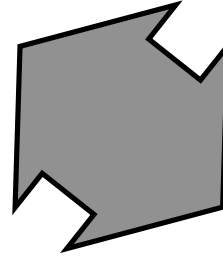
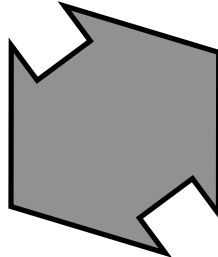
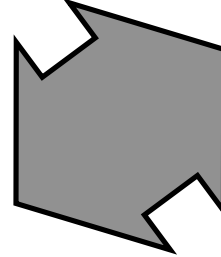
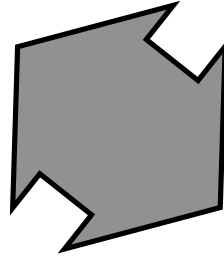
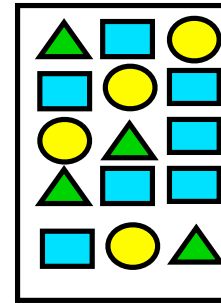
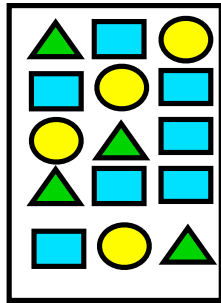
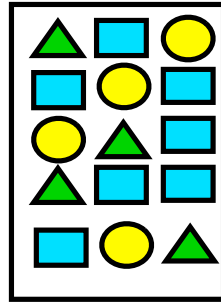
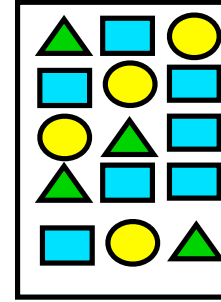
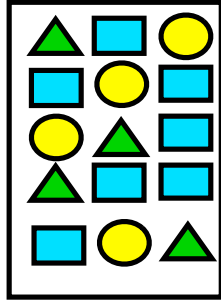
Shared Remote Archive

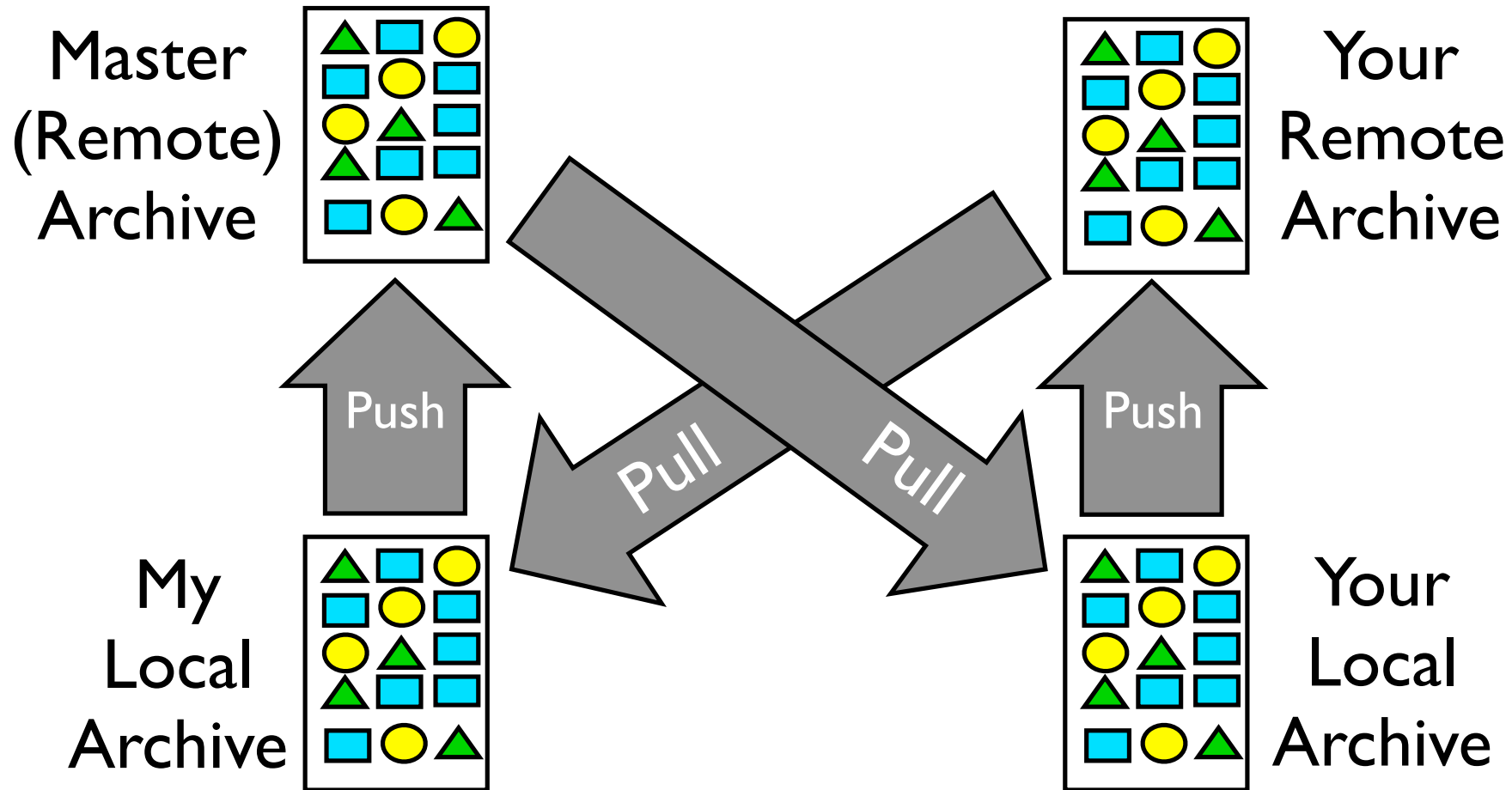


Shared Remote Archive

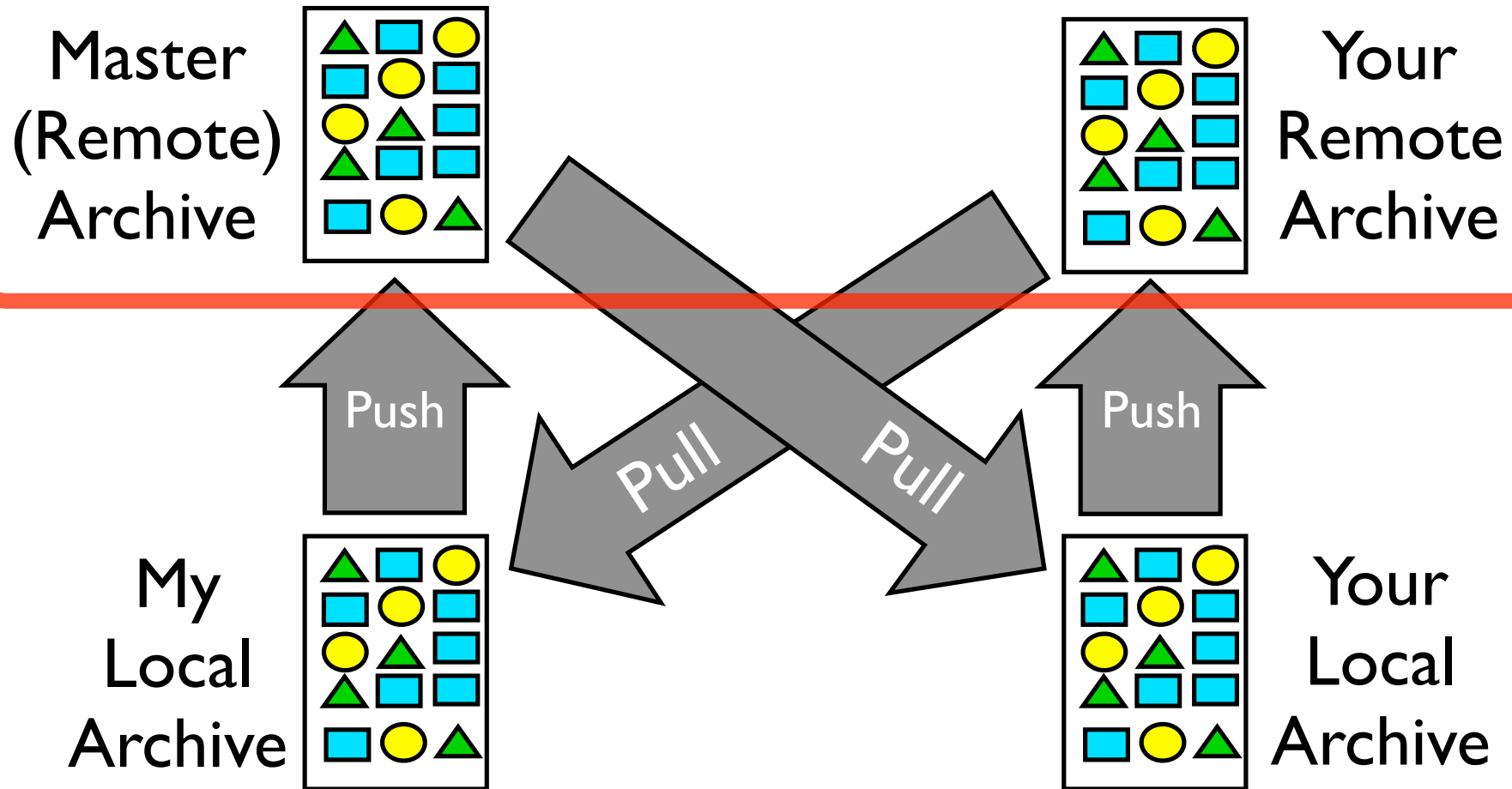




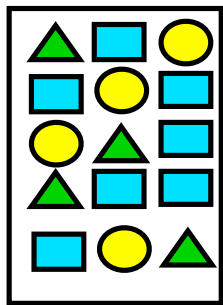




These Repos are Public

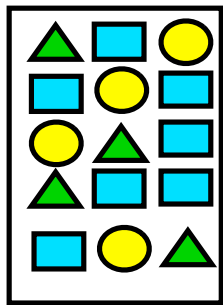


Master
(Remote)
Archive



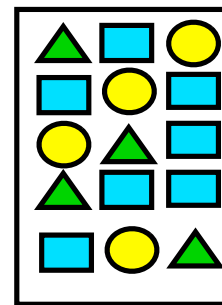
Push

My
Local
Archive



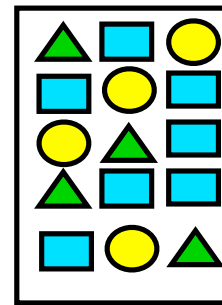
Pull

Your
Remote
Archive



Push

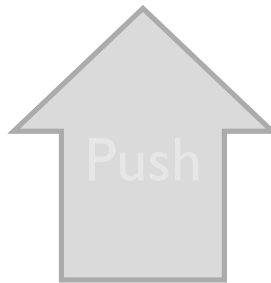
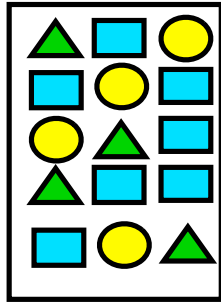
Your
Local
Archive



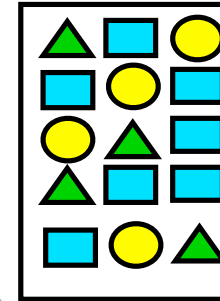
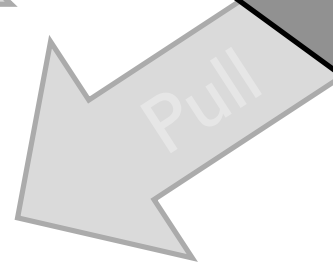
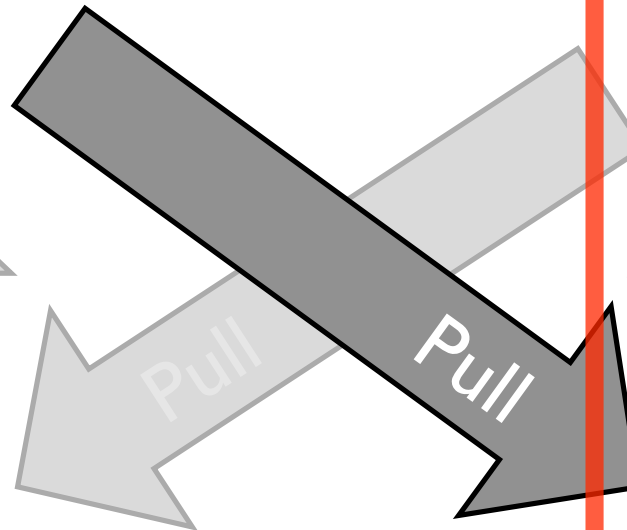
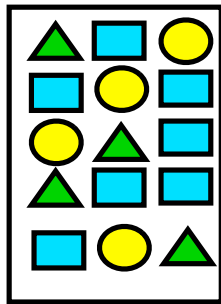
Pull

I Own These Repos

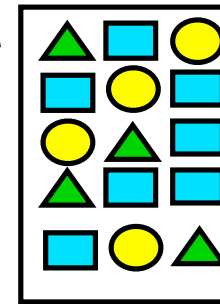
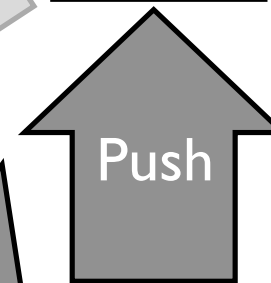
Master
(Remote)
Archive



My
Local
Archive



Your
Remote
Archive



Your
Local
Archive

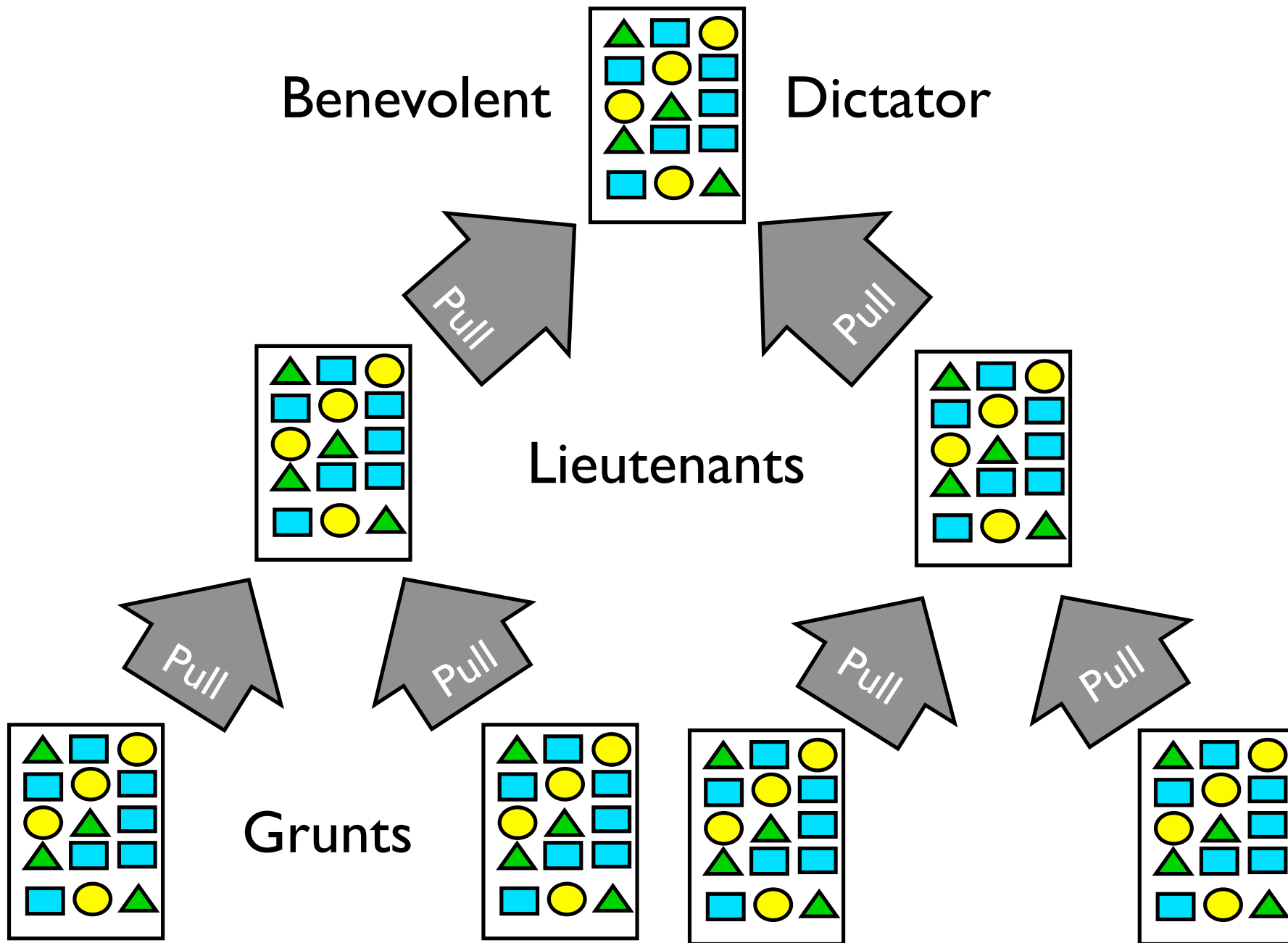
You Own These Repos

Benevolent

Dictator

Lieutenants

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 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_n
- Fixed test_n
- 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

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 ... (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

Us

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

Git

- Git Uses
 - Working Directory
 - Staging Area
 - Repository

Differences

Us

- One level object file storage

Git

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

Differences

Us

- Push/Pull entire repos

Git

- Push/Pull individual branches
- Fetch without merging to local branches
- Metadata to map branches to remote branches

Differences

Us

- Just the basic features

Git

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

Demo

Source Control for People Who Don't Like Source Control
by Jim Weirich
is licensed under a
Creative Commons Attribution-Noncommercial-Share Alike 3.0
United States License.
Based on a work at github.com.



[git://github.com/jimweirich/presentation_source_control.git](https://github.com/jimweirich/presentation_source_control.git)

Questions?