



# 01

# Gitshop

## Git - objects, refs

Jiří Jabůrek (jjaburek)  
Red Hat 2016

# Intro

- Git is git, no git vs xyz
- Git = rich suite, won't show everything
  - concepts above features
- Porcelain vs plumbing
- The dash in git commands (git-log)
- Strict argument parsing order
  - POSIXLY\_CORRECT (getopt)

```
git cmd -a arg -b --long-c arg <nonopt> [nonopt] -- file1 file2
```

- Target audience: git users



# Hashes

# Hashes

- Lots of input → fixed-size output
  - deterministic
  - function:  $y = x \bmod 10 \rightarrow y$  always 0-9
  - problem: collisions,  $y(1) = 1$  ;  $y(11) = 1$
- Cryptographic hash function
  - very low chance of collisions
  - very hard to guess input given output
  - small change in input produces very different output
    - `sha1("text"):` 372ea08cab33e71c02c651dbc83a474d32c676ea
    - `sha1("test"):` a94a8fe5ccb19ba61c4c0873d391e987982fbbd3
    - 84 different bits (out of 160)



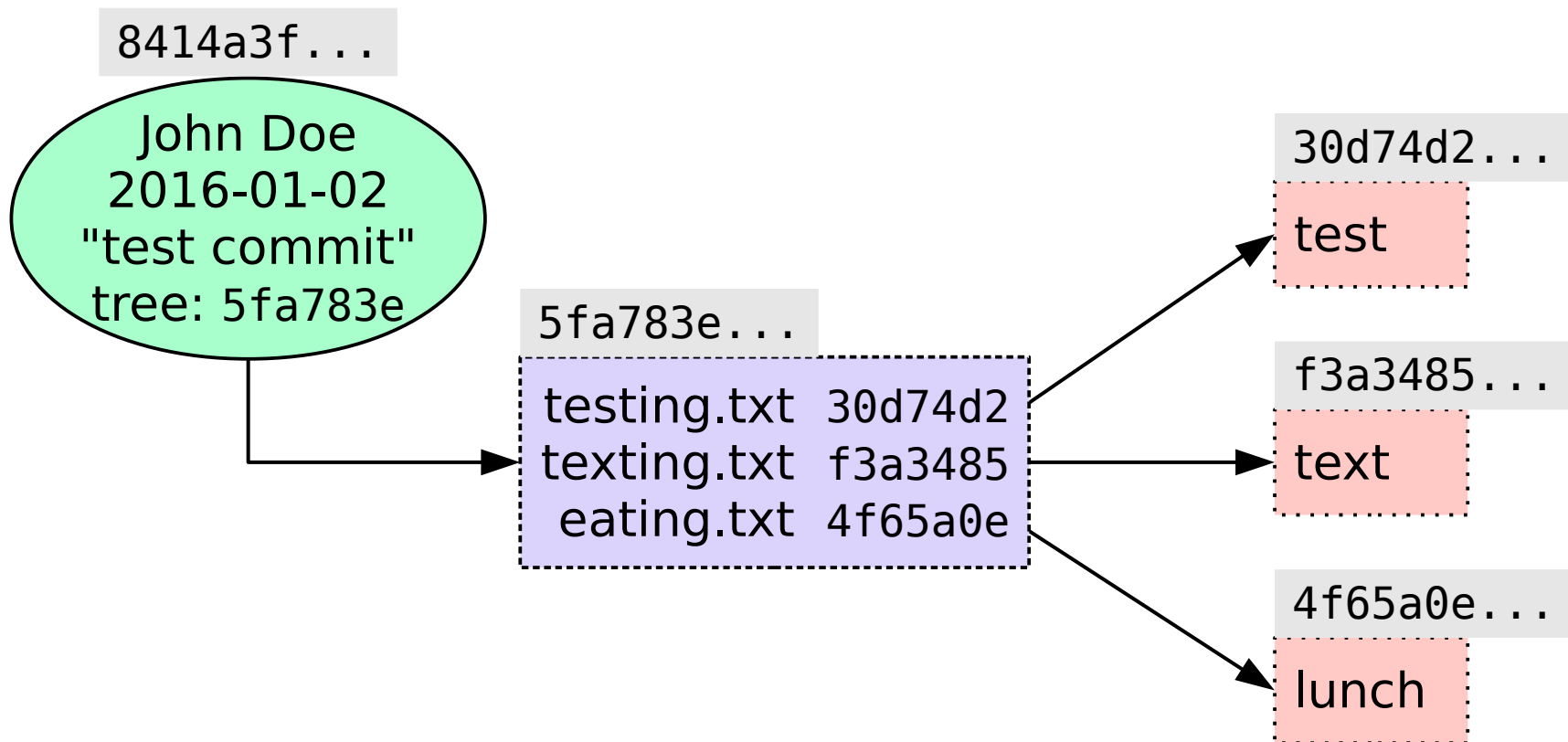
# Objects

# Objects



- Store data, metadata
- Form the Git object database `.git/objects/`
- Are identified by SHA1 hash
- Simple ideas forming a complex system

# Objects



# Objects - blob

blob

## ■ File contents

- regular (text/binary) data
- symlink locations
- zlib-compressed (when unpacked)

```
c7f4dad04c0872f0e24b575b48257d0b72cffb0a
```

```
blob 14\0My first blob.
```

```
4065cc1a7db8b54e50407410bb7b081711e19b8e
```

```
blob 15\0My second blob.
```



# Objects - tree

tree

- Collection of file contents
  - store file mode, file names, etc.

```
292778a18c54dea66754260b9a2d9ff8aef73b46
```

```
tree 58\0
100644 a\0c7f4dad04c0872f0e24b575b48257d0b72cffb0a
100644 b\04065cc1a7db8b54e50407410bb7b081711e19b8e
```

# Objects - tree

tree

- Collection of file contents
  - store file mode, file names, etc.

```
292778a18c54dea66754260b9a2d9ff8aef73b46
```

```
tree 58\0
100644 a\0c7f4dad04c0872f0e24b575b48257d0b72cffb0a
100644 b\04065cc1a7db8b54e50407410bb7b081711e19b8e
```

```
tree 58\x00100644 a\x00\xc7\xf4\xda\xd0\x4c\x08\x72
\xfa\x0e2\x4b\x57\x5b\x48\x25\x7d\x0b\x72\xcf\xfb\x0a
100644 b\x00\x40\x65\xcc\x1a\x7d\xb8\xb5\x4e\x50\x40
\x74\x10\xbb\x7b\x08\x17\x11\xe1\x9b\x8e
```

# Objects - commit

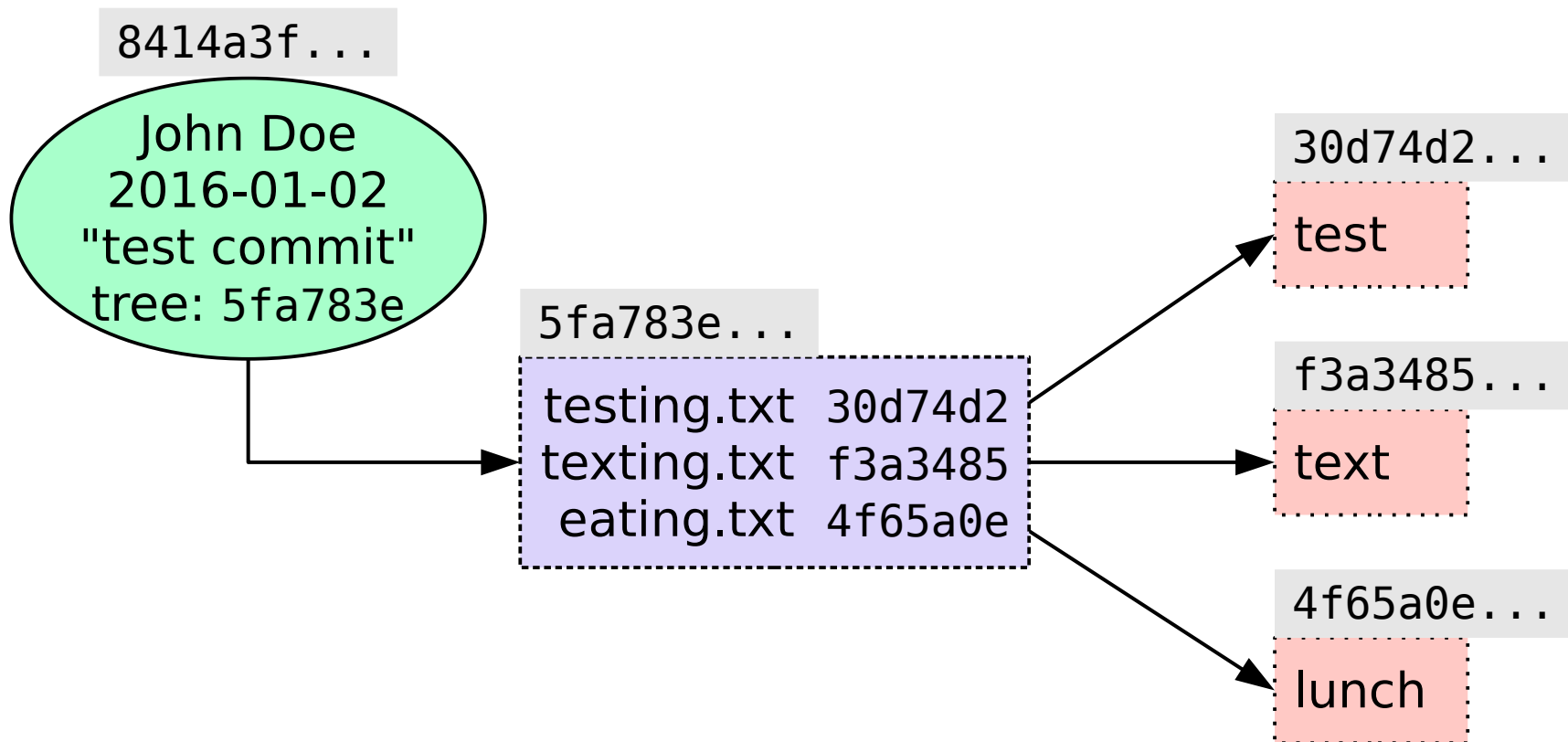
commit

- Annotation of a tree
  - has 0 or more parents
  - has author/commiter/date/...

```
e5d91a30efe8b6f75abfd8f05b752a11d2cc0278
```

```
commit 229\0tree 292778a18c54dea66754260b9a2d9ff8aef73b46\n
parent a95274e7931ab0f5108ab292140620750a0de8ff\n
author Jiri Jaburek <jjaburek@redhat.com> 1456166293 +0100\n
committer Jiri Jaburek <jjaburek@redhat.com> 1456166293 +0100\n
\n
some message\n
```

# Objects





Object = unique contents

# Objects

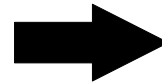


## ■ Data (blob) duplication

b9c44e5...

Completely synergize resource  
taxing relationships via premier  
niche markets. Professionally  
cultivate one-to-one customer  
service with robust ideas.  
Dynamically innovate resource-  
leveling customer service for state  
of the art customer service.

Objectively innovate empowered  
manufactured products whereas  
parallel platforms. Holisticly  
predominate extensible testing  
procedures for reliable supply  
chains. Dramatically engage top-line  
web services vis-a-vis cutting-edge  
deliverables.



f16e63f...

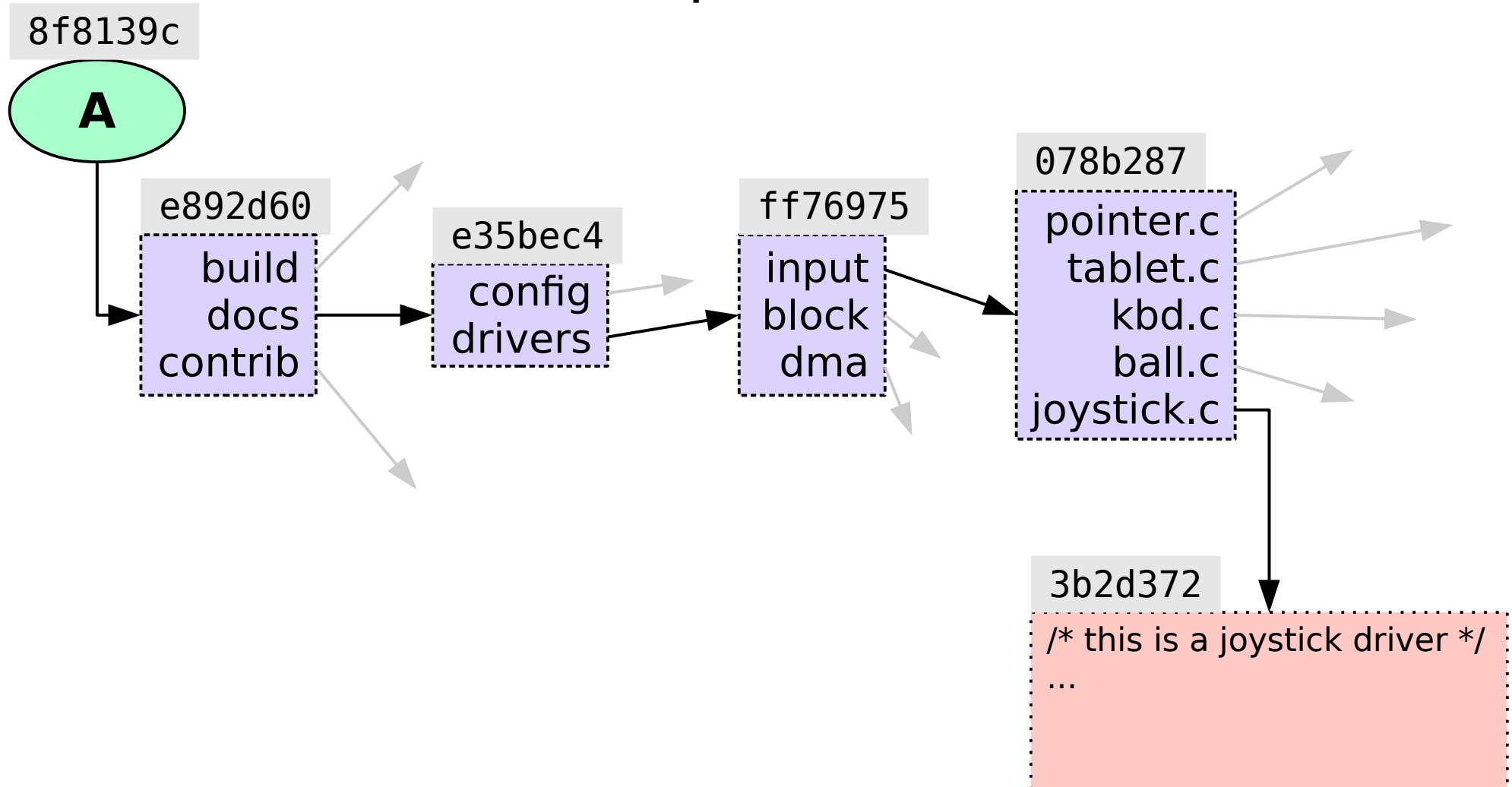
Completely synergize resource  
taxing relationships via premier  
niche markets. Professionally  
cultivate one-to-one customer  
service with robust ideas.  
dynamically innovate resource-  
leveling customer service for state  
of the art customer service.

Objectively innovate empowered  
manufactured products whereas  
parallel platforms. Holisticly  
predominate extensible testing  
procedures for reliable supply  
chains. Dramatically engage top-line  
web services vis-a-vis cutting-edge  
deliverables.

# Objects



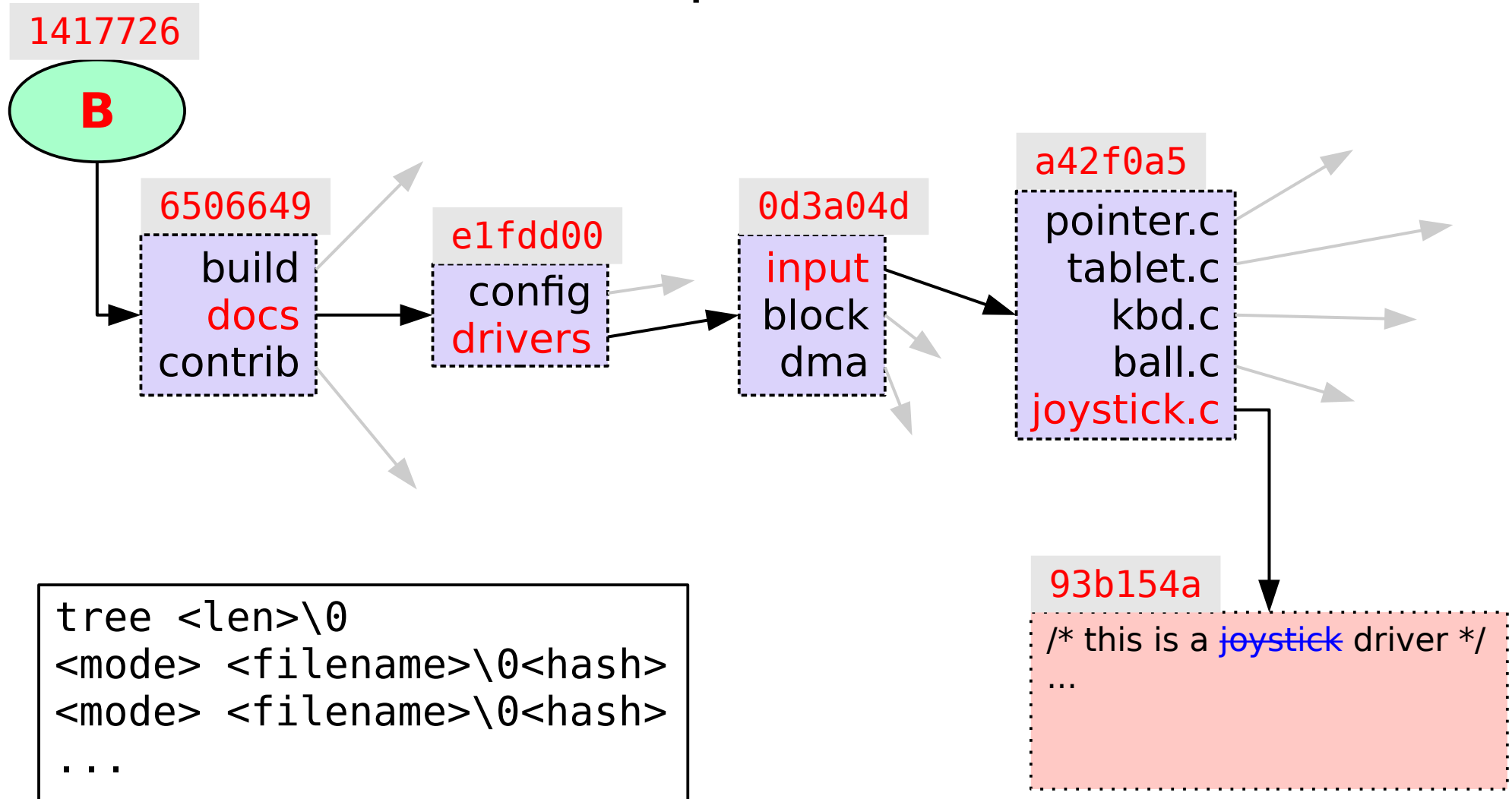
## ■ Metadata (tree) duplication



# Objects



## ■ Metadata (tree) duplication

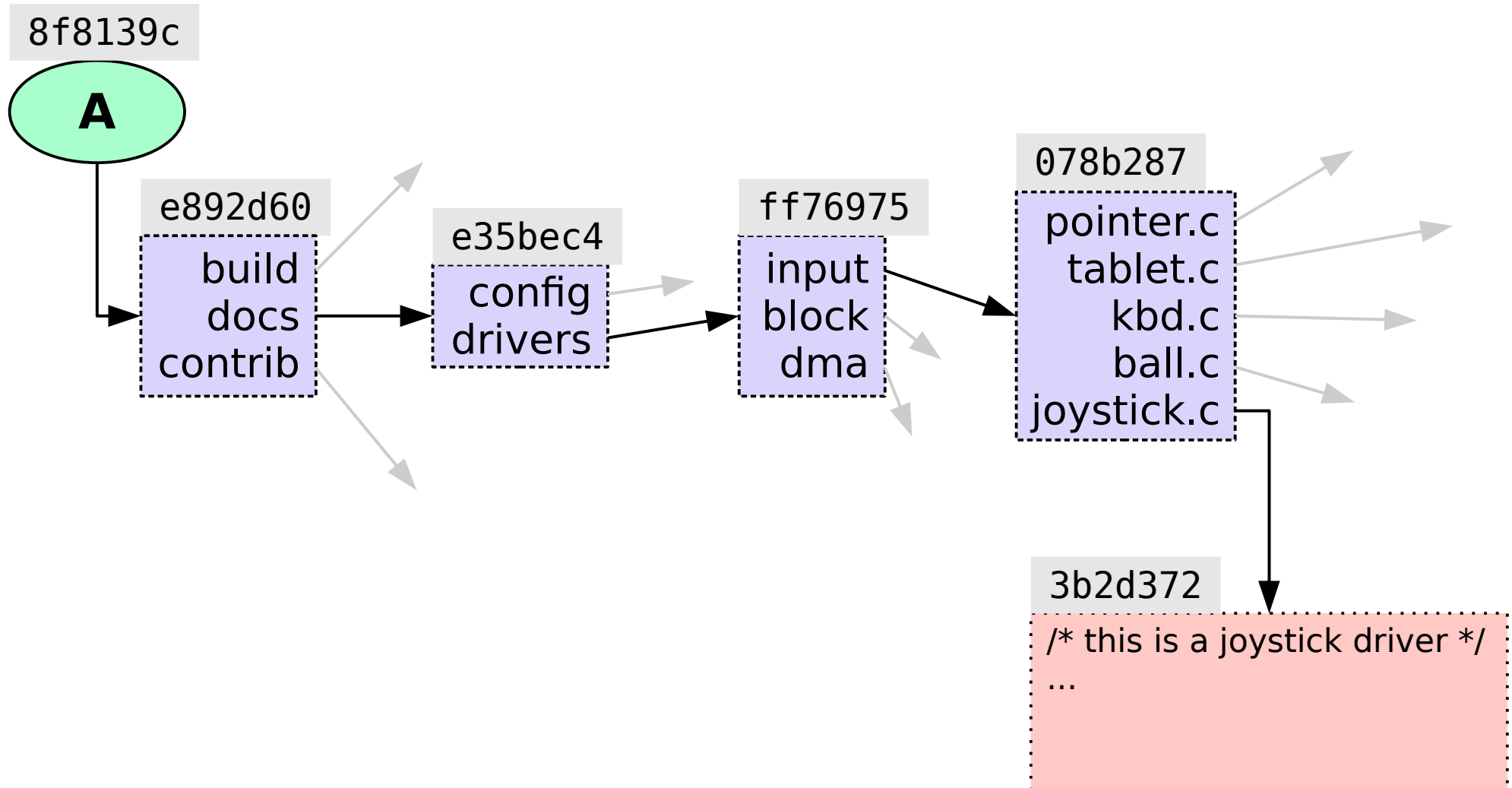




# Objects



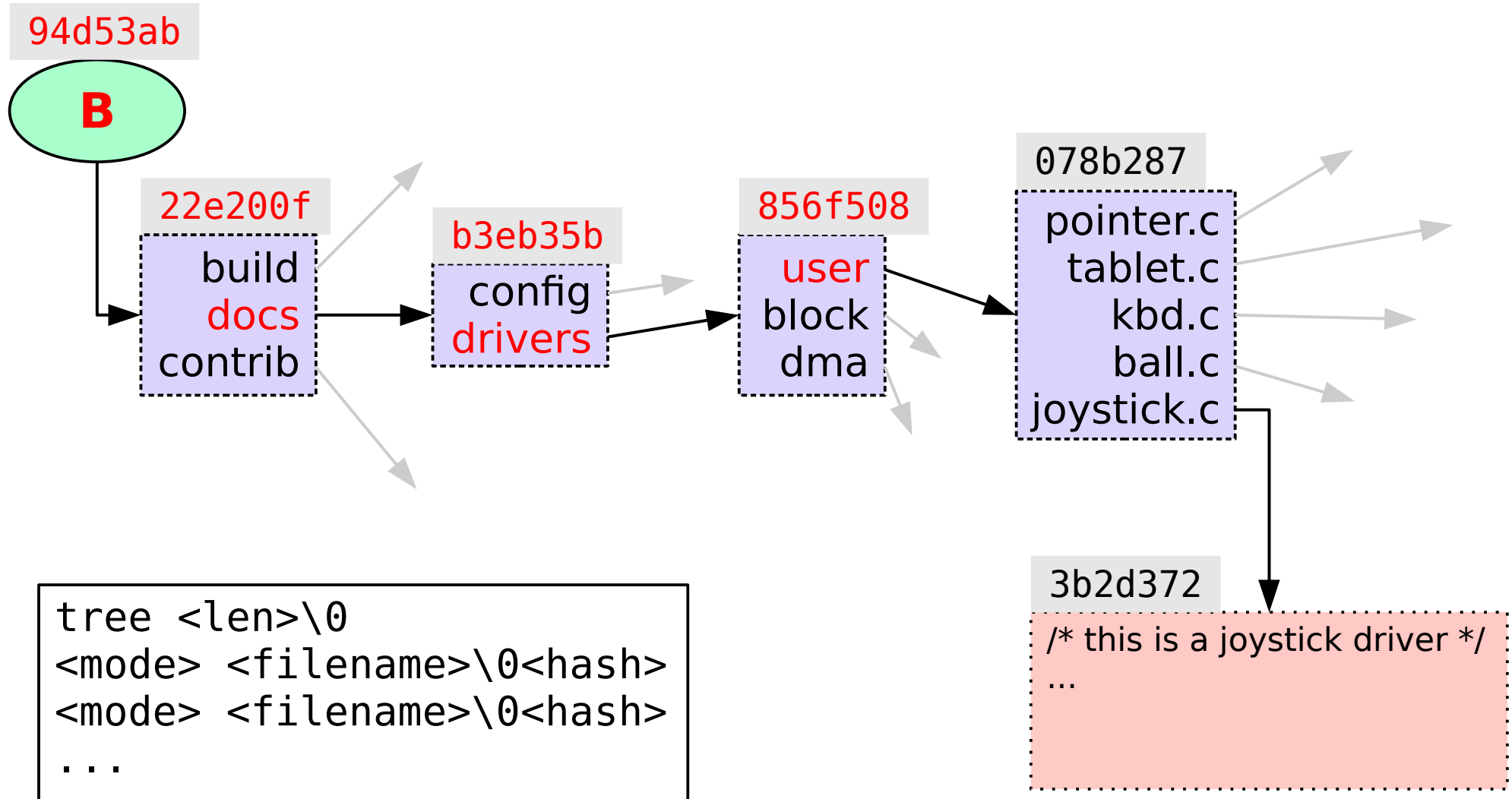
## ■ Renames



# Objects



## ■ Renames



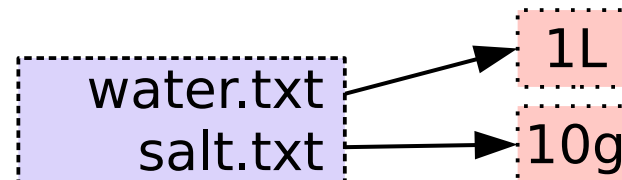


# Commit History

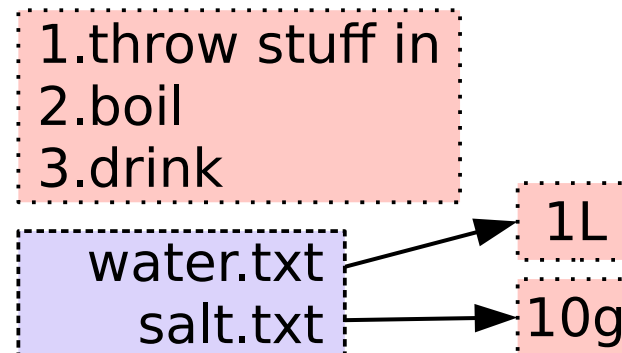
# Commit History



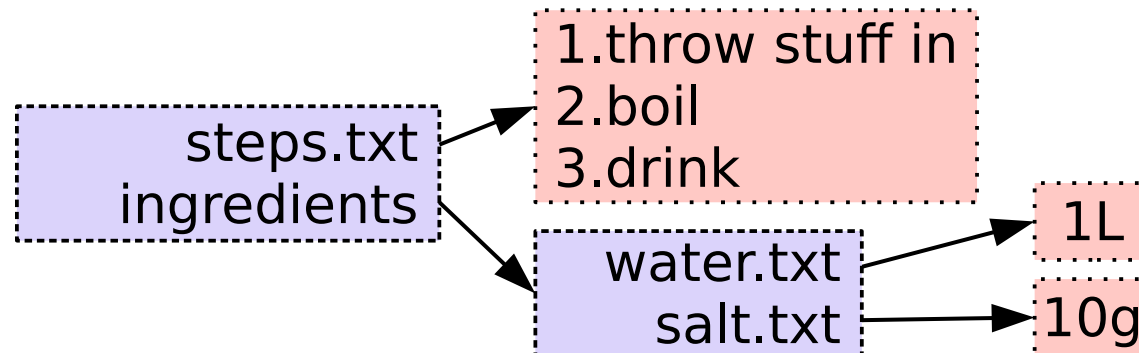
# Commit History



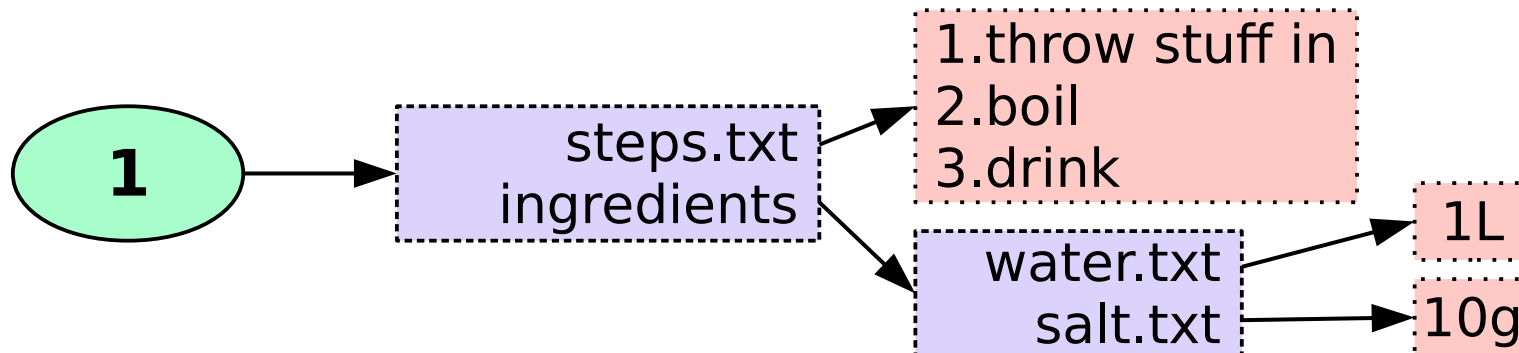
# Commit History



# Commit History

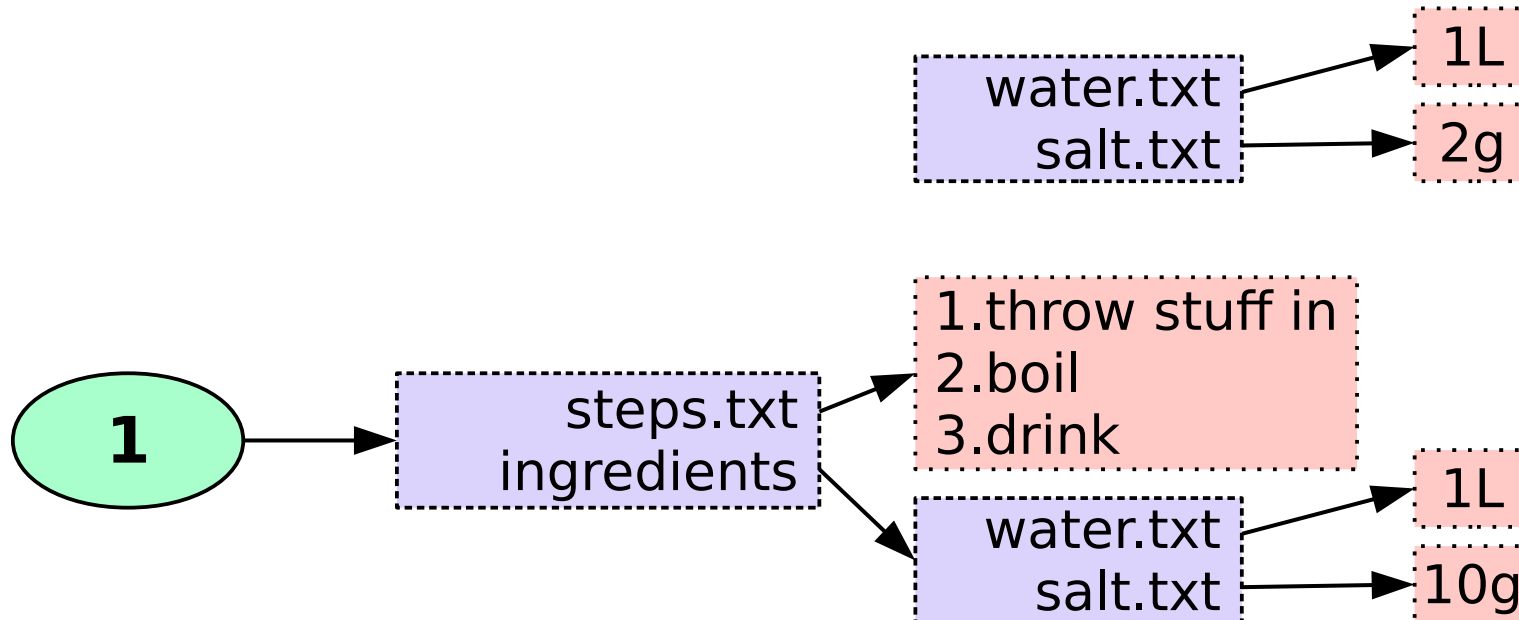


# Commit History

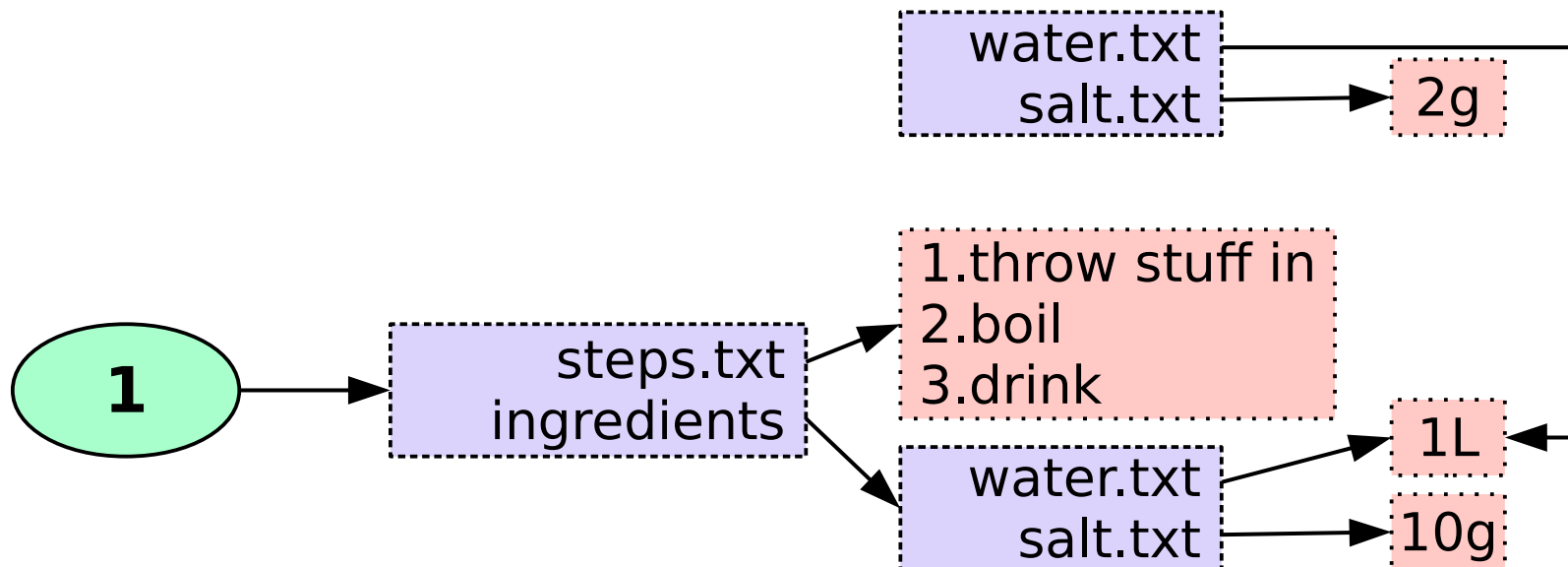




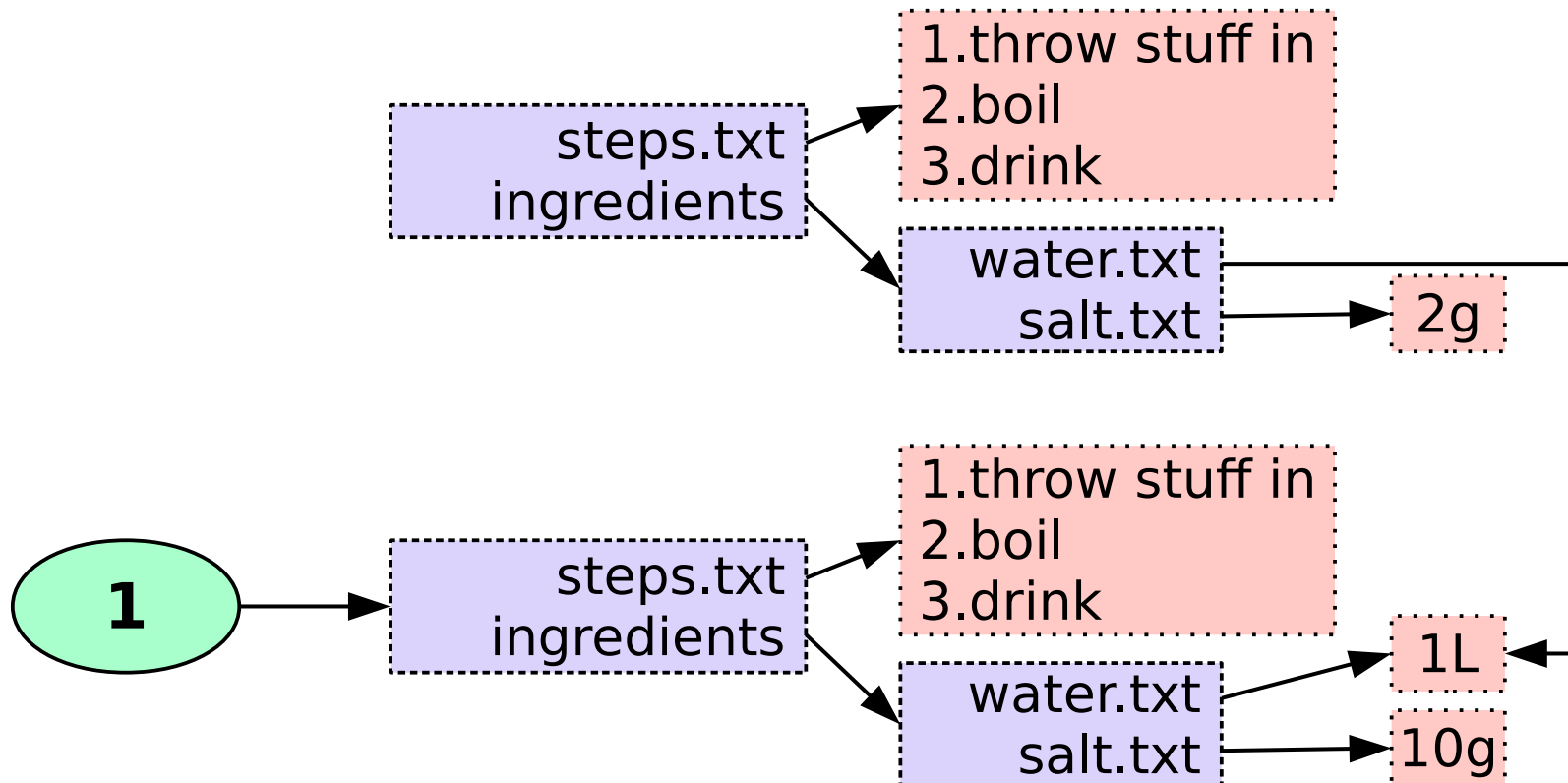
# Commit History



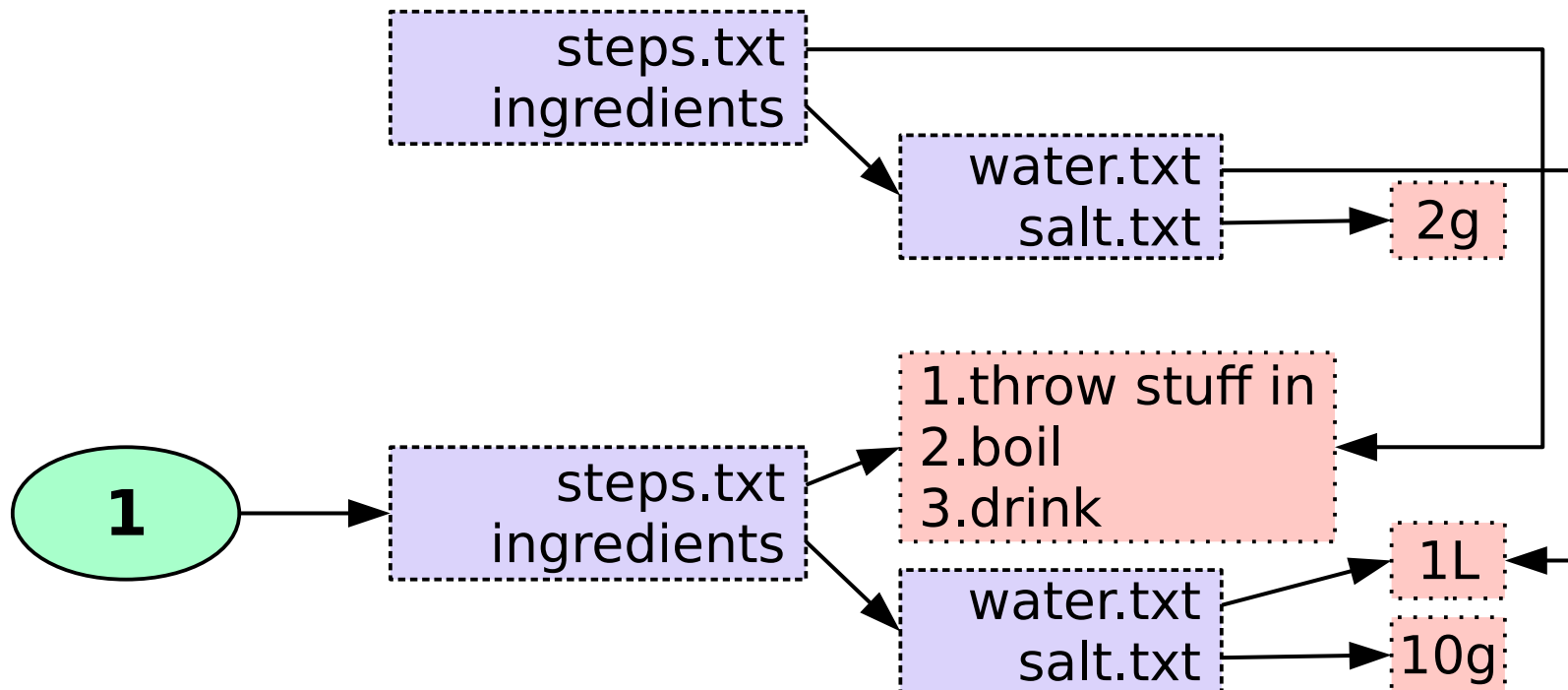
# Commit History



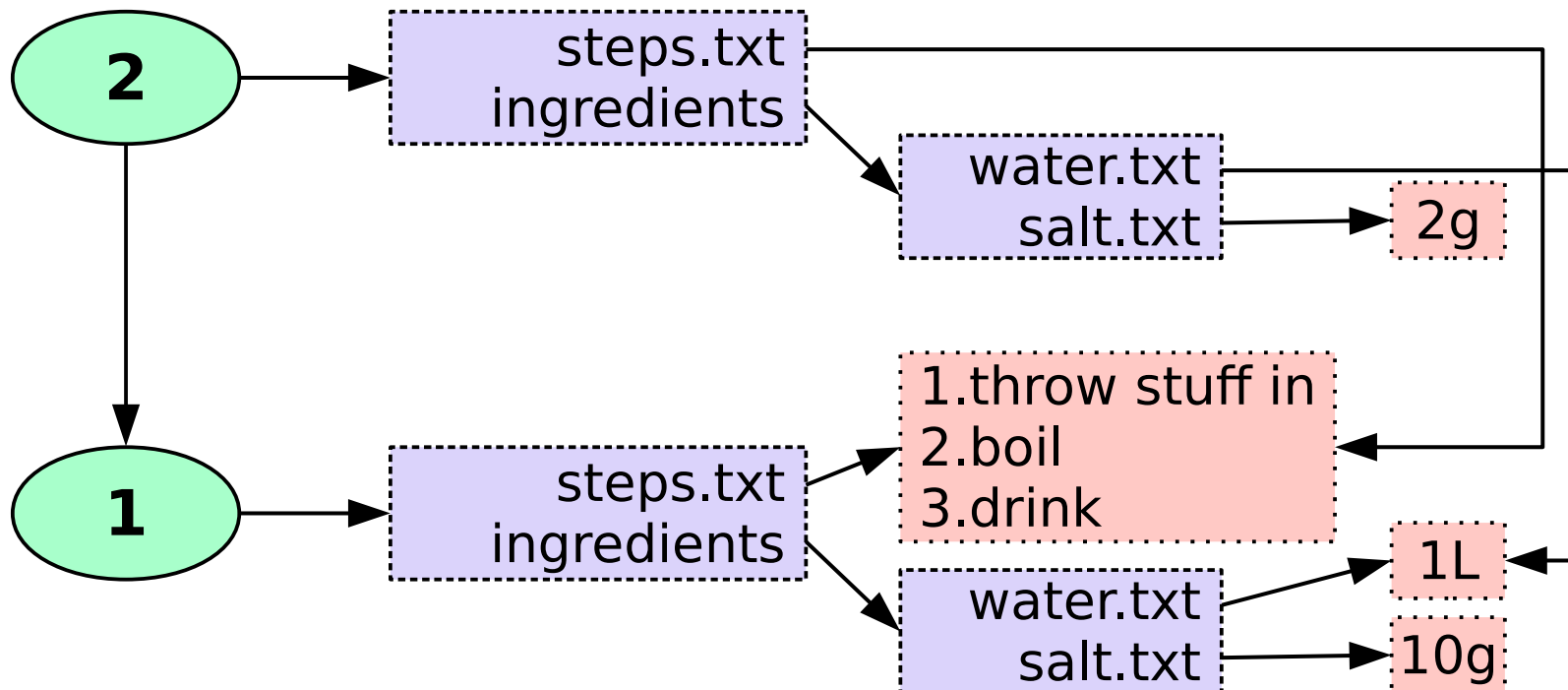
# Commit History



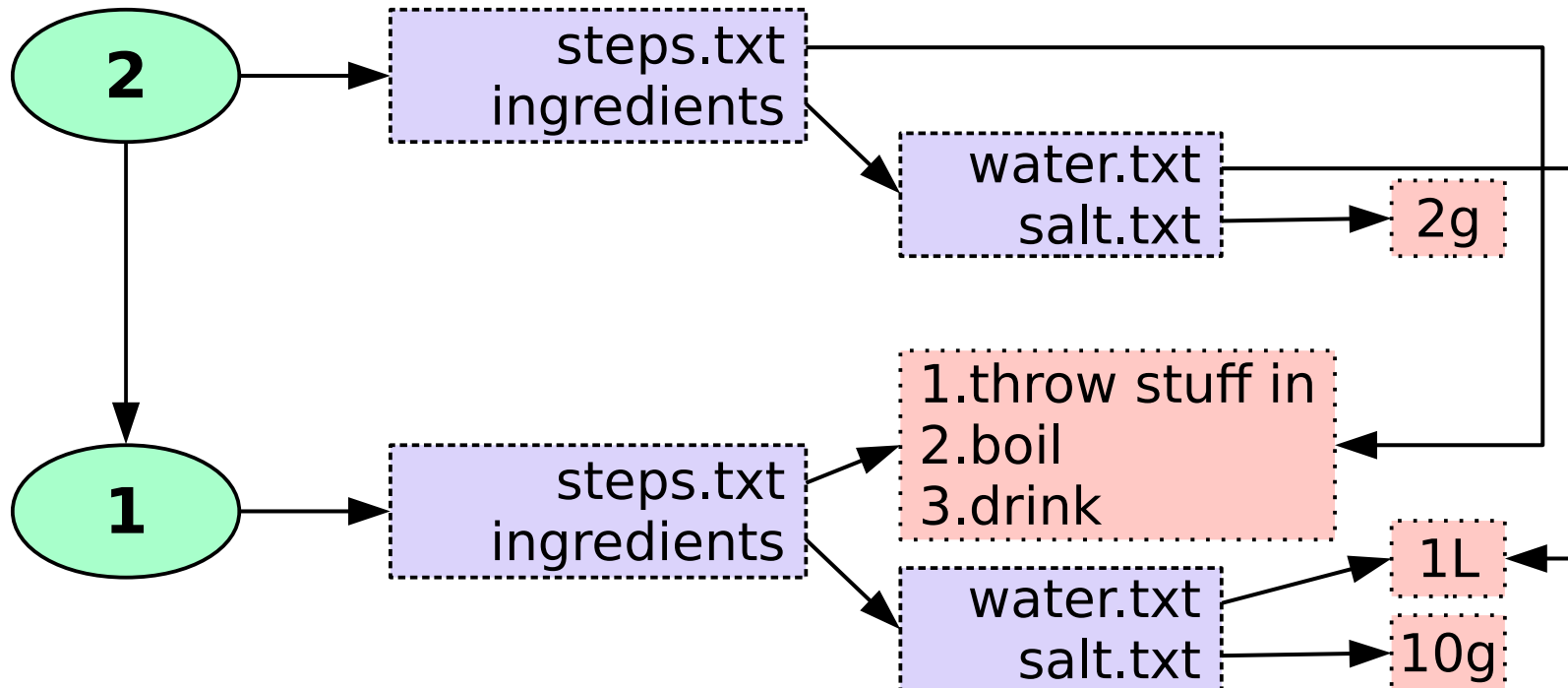
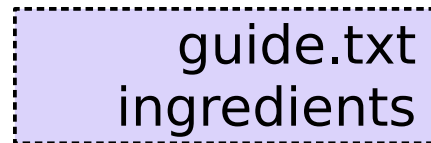
# Commit History



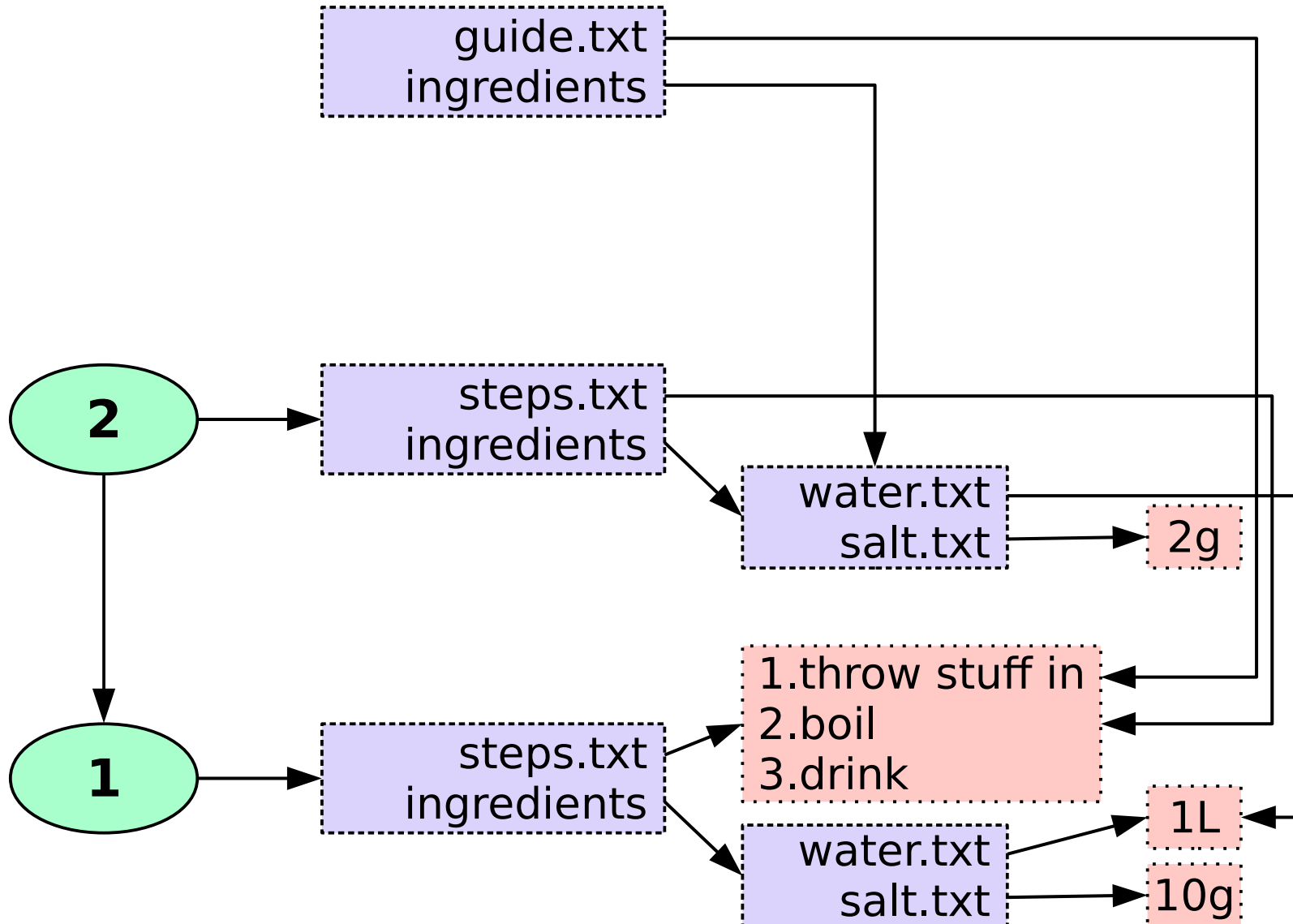
# Commit History



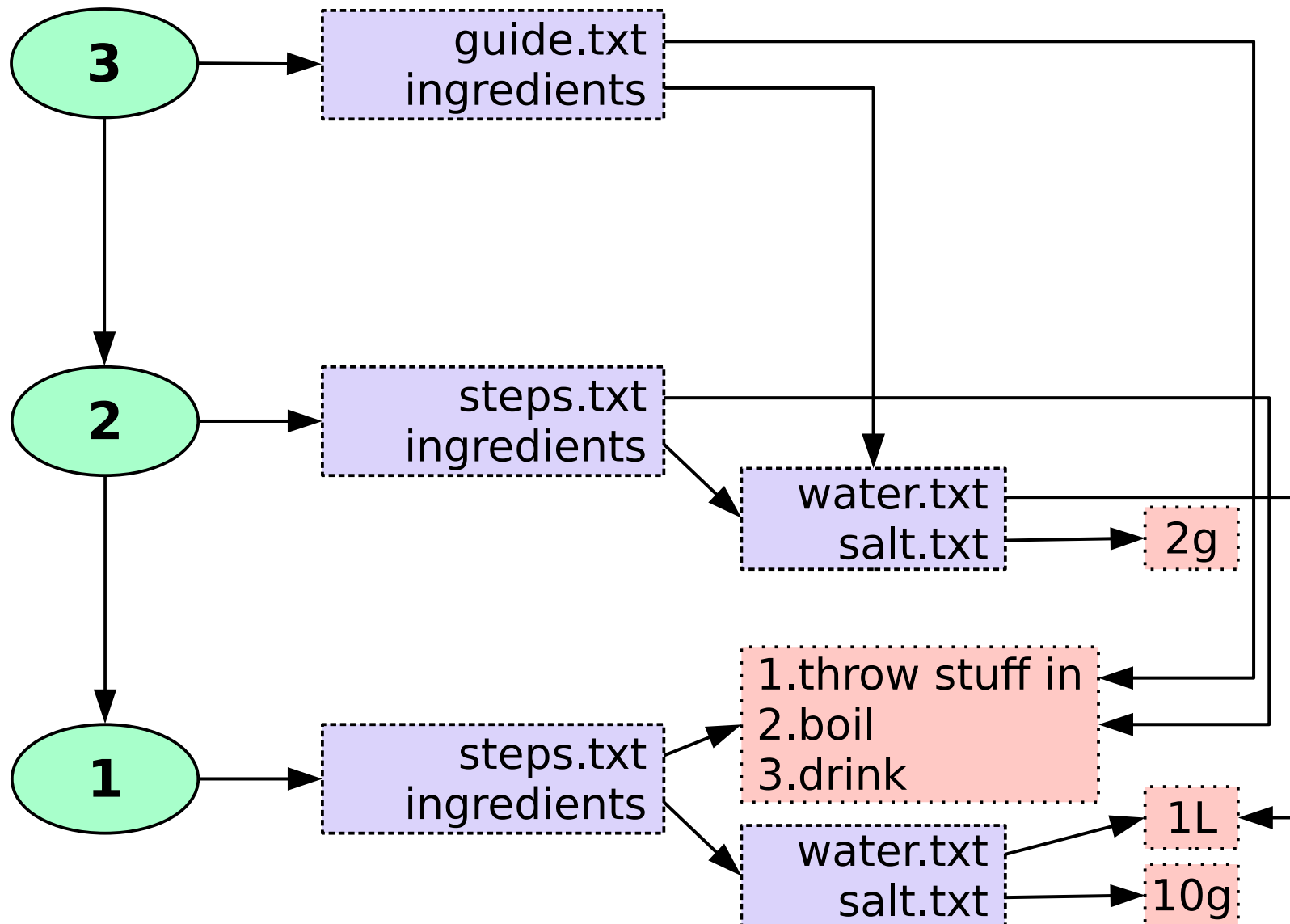
# Commit History



# Commit History

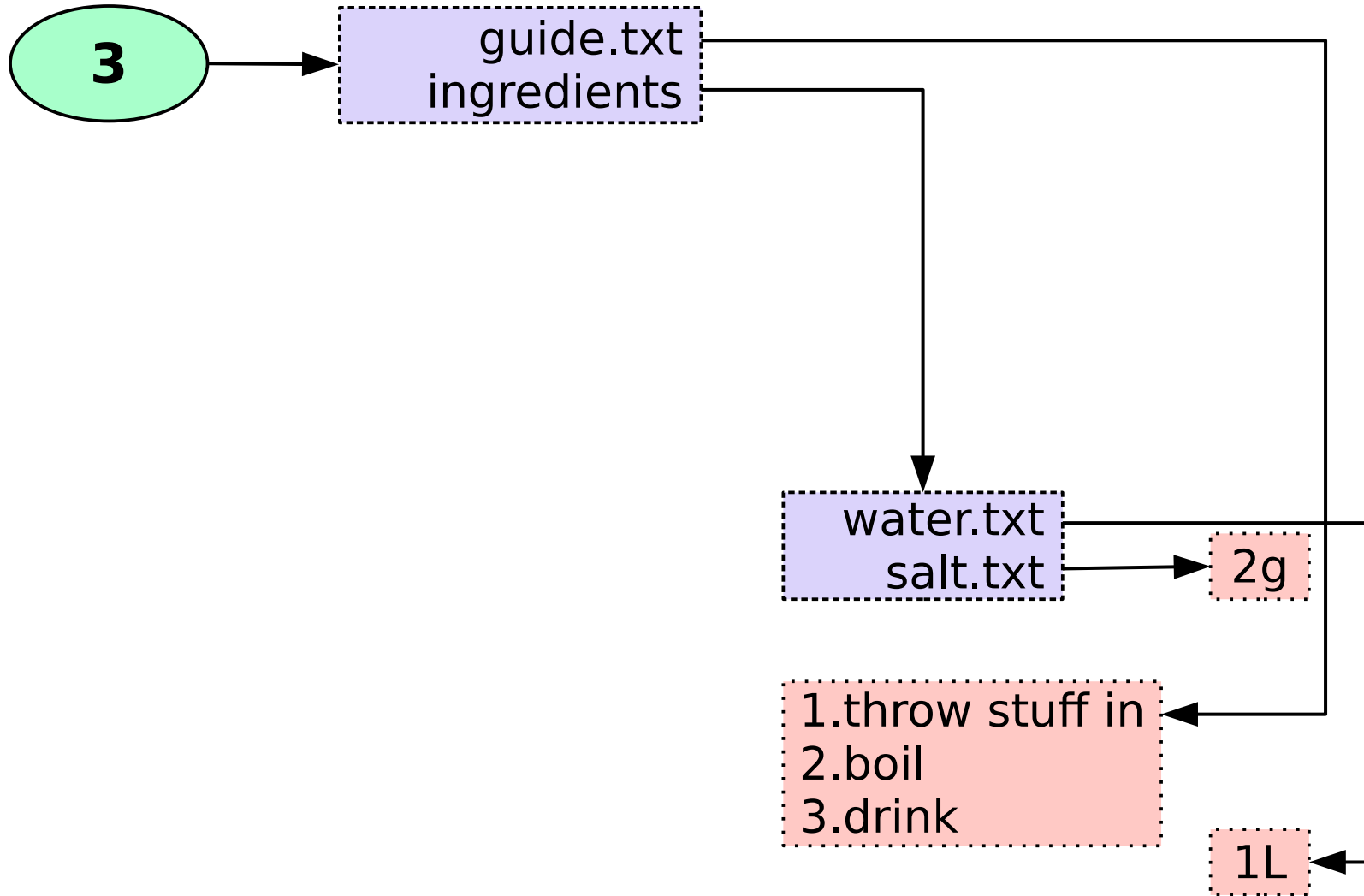


# Commit History

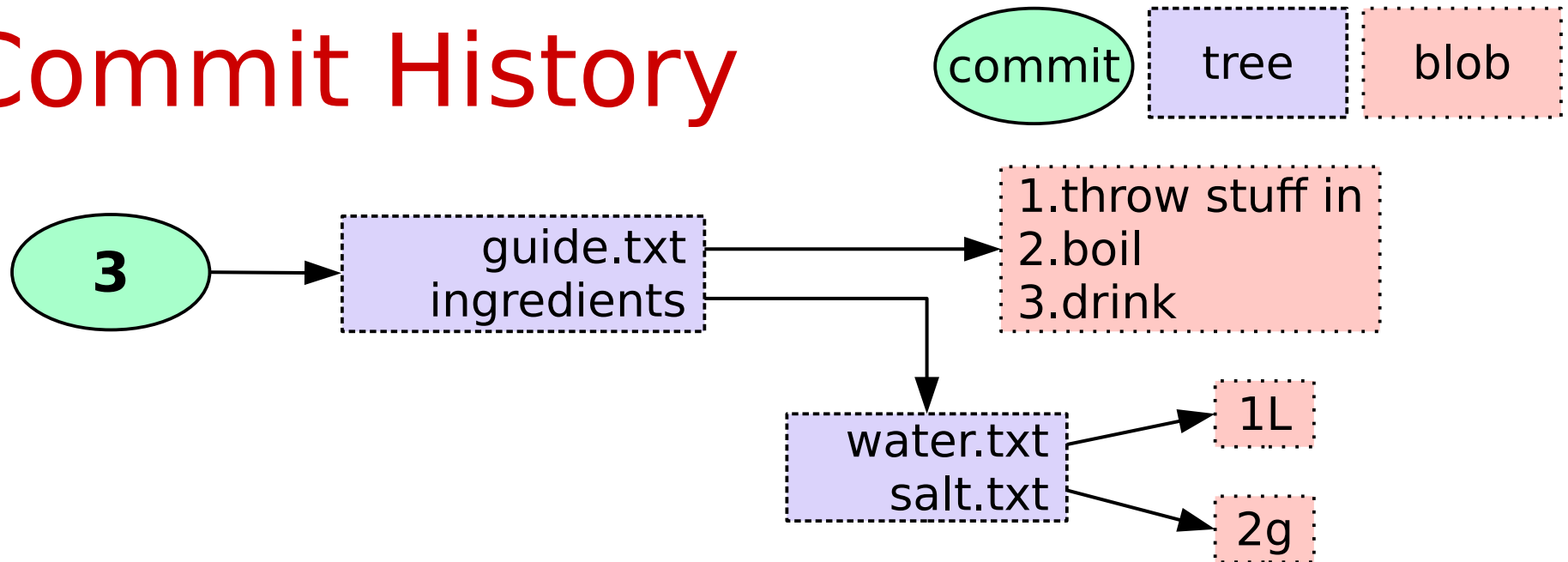




# Commit History



# Commit History





# Objects: as files

# Objects

- First byte (2 hex digits) used as dir name
- Have zlib-compressed contents
  - hash made from **un**compressed data
- In GIT\_DIR (typically .git):

```
objects/  
objects/a5/  
objects/a5/196d1be8fb59edf8062bef36d3a602e0812139  
objects/bd/  
objects/bd/14dffbc42007e30af19cb5b56fb8546f2bd800  
objects/cb/  
objects/cb/089cd89a7d7686d284d8761201649346b5aa1c  
objects/cb/28eb36c80d76b75cc144a6d7b00735f3a3da19
```

# Objects

- Pack files

- smart tar-balls of objects, on steroids
- use delta compression of objects
- are pretty complex

- In GIT\_DIR (typically .git):

```
objects/  
objects/pack/  
objects/pack/pack-<hash>.pack  
objects/pack/pack-<hash>.idx
```



# References

# References



- Point to objects

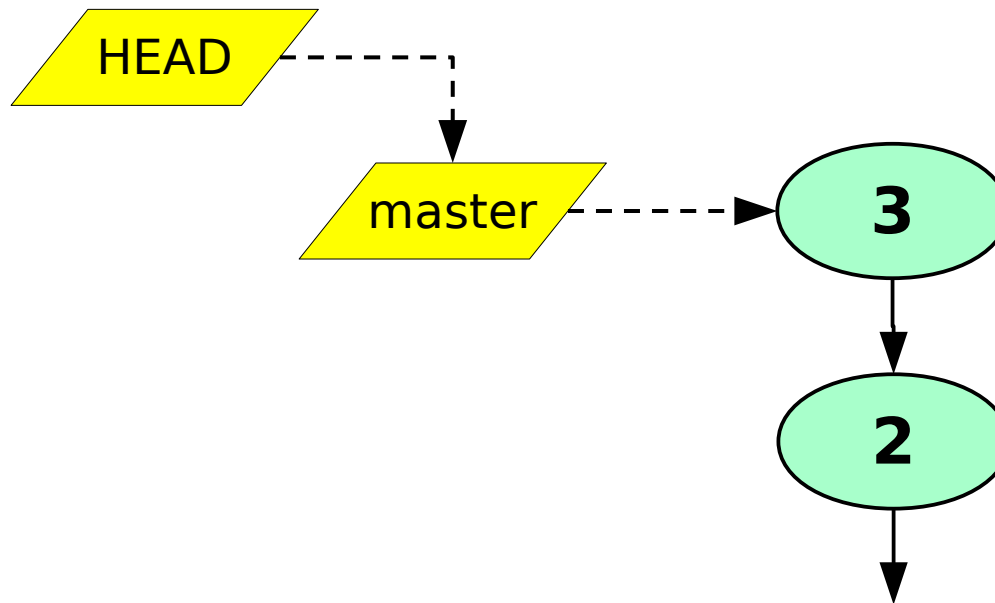
- like hardlinks: 929c2ee5ca19b576b87f43f8cdc44eac447e8404
- like symlinks: ref: path/to/another/ref

# References

ref

## ■ Point to objects

- like hardlinks: 929c2ee5ca19b576b87f43f8cdc44eac447e8404
- like symlinks: ref: path/to/another/ref





# References

- In GIT\_DIR (typically .git):

```
HEAD  
refs/  
refs/heads/  
refs/heads/master  
refs/remotes/  
refs/remotes/origin/  
refs/remotes/origin/master  
refs/tags/  
refs/tags/v1.0  
refs/mycategory/subcat/work
```

# References



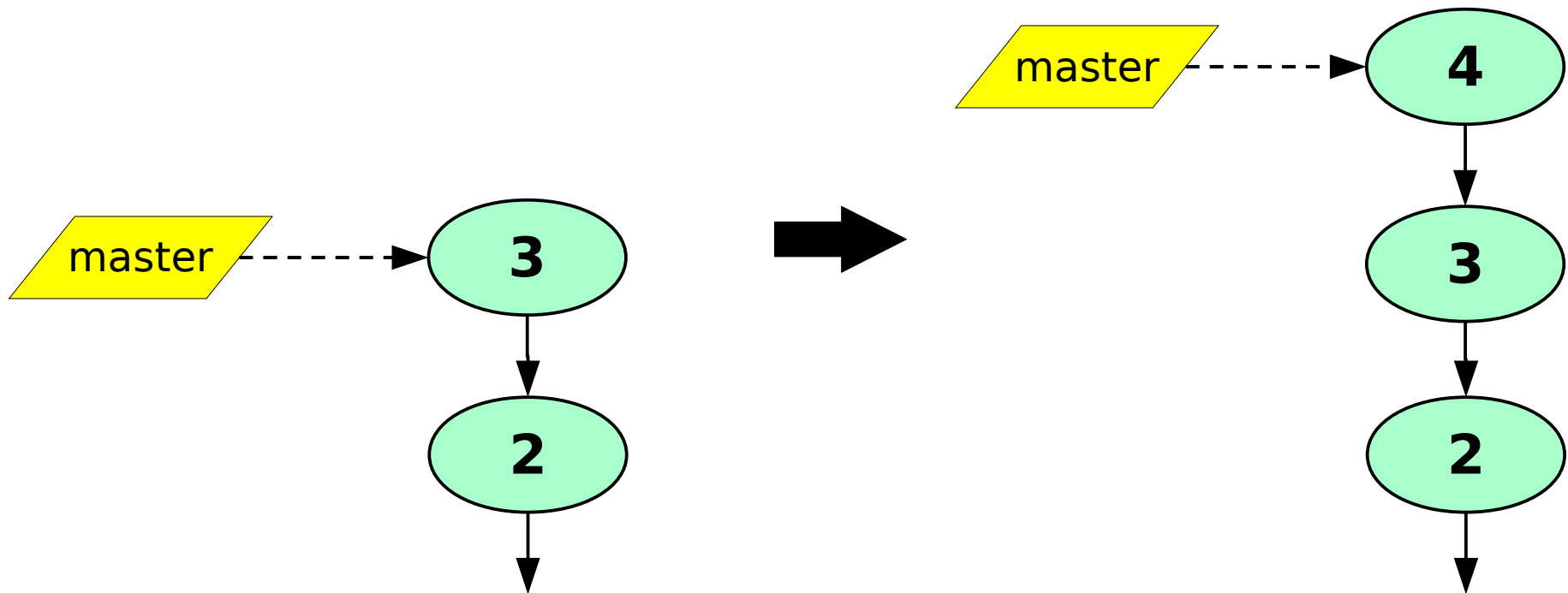
- Branches
  - generally point to commits (hashes)
  - get updated on commit

# References

ref

## ■ Branches

- generally point to commits (hashes)
- get updated on commit



# References



- HEAD
  - currently checked-out branch (commit, tree)

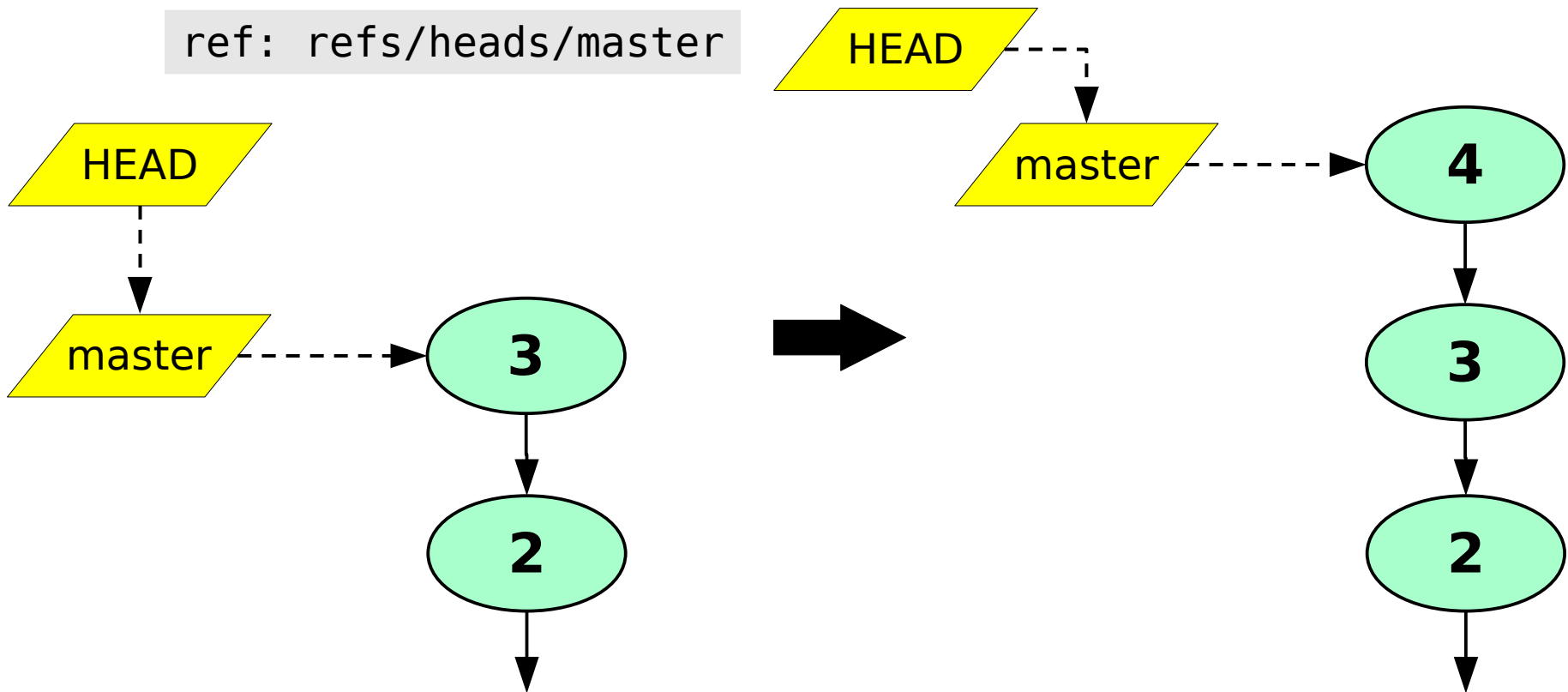
# References

ref

## ■ HEAD

- currently checked-out branch (commit, tree)

ref: refs/heads/master



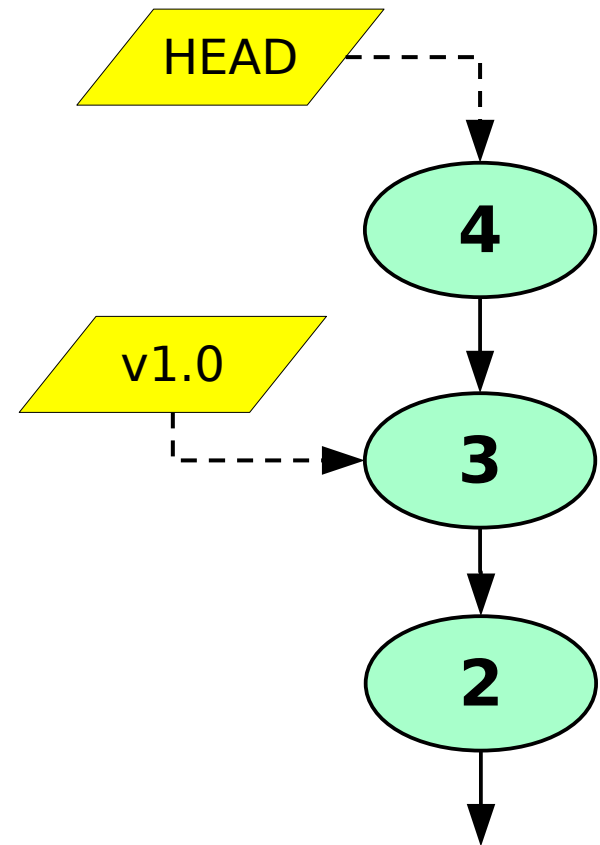
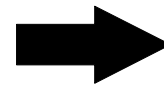
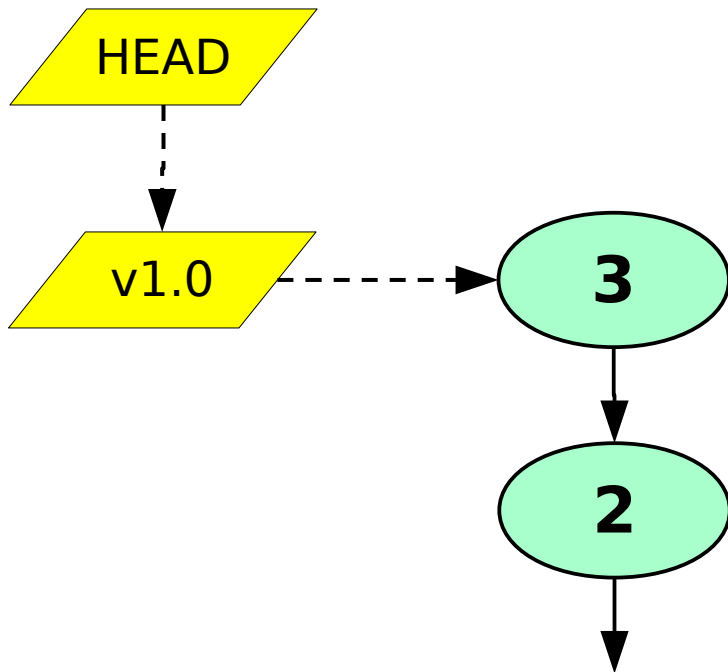
# References



- Tags
  - like branches, but immutable

# References

- Tags
  - like branches, but immutable

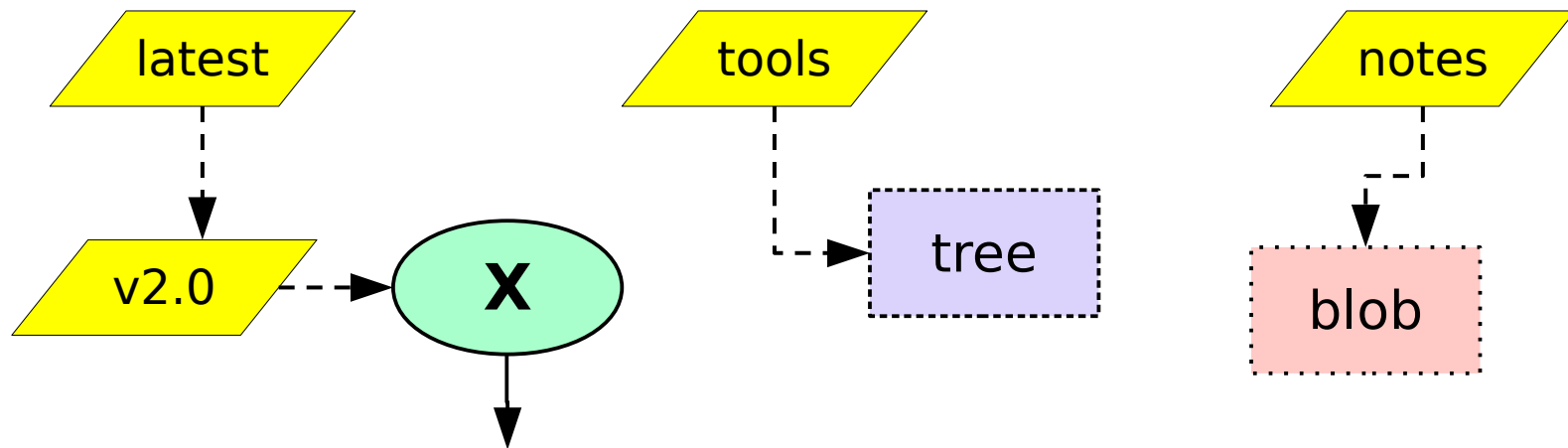


ref

# References

ref

- Custom refs
  - can be anywhere in refs/
    - or outside, like HEAD





# References

- In GIT\_DIR (typically .git):

```
HEAD  
refs/  
refs/heads/  
refs/heads/master  
refs/remotes/  
refs/remotes/origin/  
refs/remotes/origin/master  
refs/tags/  
refs/tags/v1.0
```

- Precedence:

```
<refname>  
refs/<refname>  
refs/tags/<refname>  
refs/heads/<refname>  
refs/remotes/<refname>  
refs/remotes/<refname>/HEAD
```



# Summary

# Summary

## ■ Objects

- blob = contents ; tree = dir tree ; commit = metadata
- Contents hashed by SHA-1, identified by hash
- No two exact copies stored, by design
- Any change in (sub)tree propagates up

## ■ References

- Pointers to objects or other references
- HEAD points to current checkout
  - can be detached (point to commit directly)
- Branch is updated (via HEAD) on new commit



# Questions?

# The End

Thanks for listening



Workshop >>

# Workshop

**Goal:** Initialize a git repository, create 2 files 2 directory levels deep without using regular filesystem. Create a commit and put it on a branch. Make HEAD point to it.

**Setup:** `echo first > /tmp/first`  
`echo second > /tmp/second`

## 1) Initialize a new repository

- `git init [reponame]`

## 2) Hash content

- `git hash-object -w`

## 3) Create tree(s)

- `git mktree`
- See `man git-ls-tree`, section OUTPUT FORMAT
  - mode = UNIX mode, see `man 2 stat`, look for `st_mode`

## 4) Commit root tree

- `git commit-tree`
  - enter commit message (multiline), hit Control-D

## 5) Create branch

- `git branch` or using `echo` manually (`refs/heads/name`)

## 6) Adjust HEAD

- `git symbolic-ref` or using `echo` (HEAD in `.git` directly)

## 7) See result

- `git log --stat --format=raw ; gitk`
- `git checkout -f`