

INTRO TO GIS_c

GEODATABASES

INTRO TO GIS_c

POTPOURRI

AGENDA

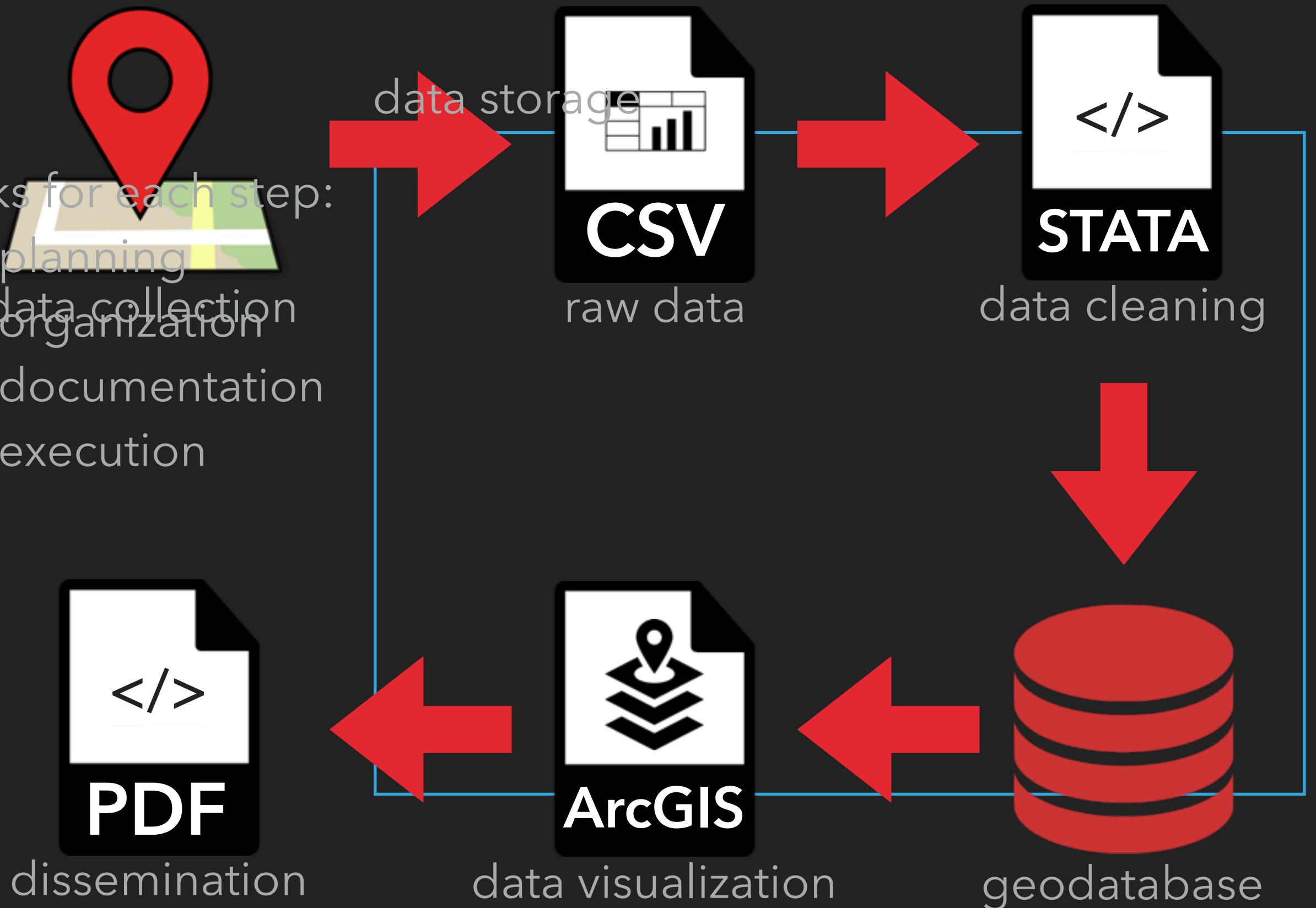
1. Follow-up
2. Advanced GitHub
3. Creating New Variables
4. Geodatabases

1 FOLLOW-UP

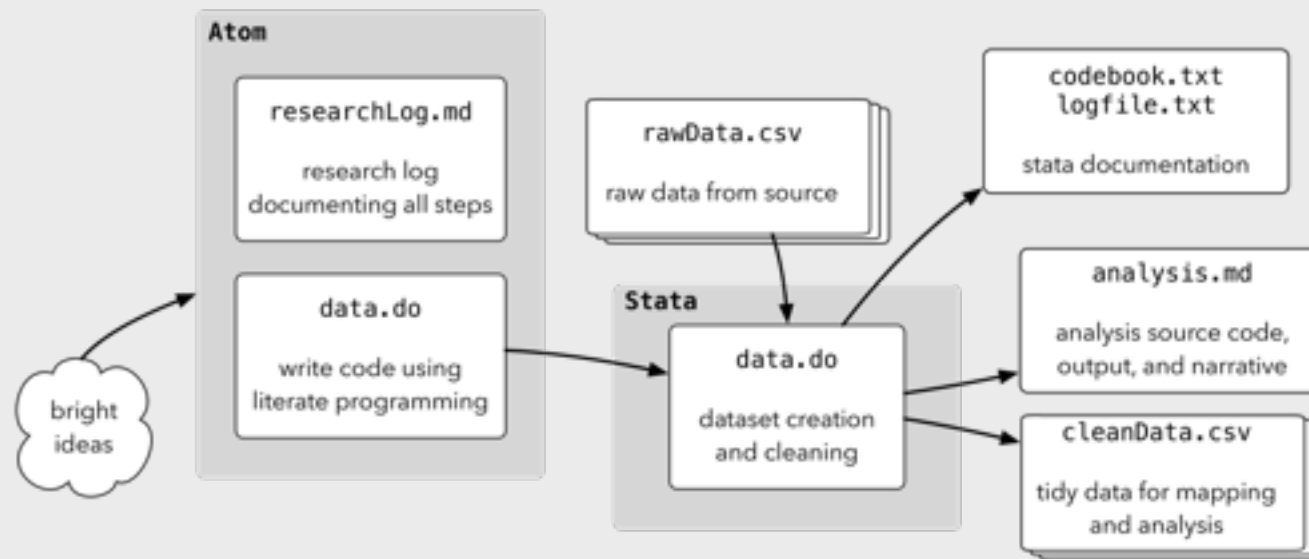
GIS_c WORKFLOW

tasks for each step:

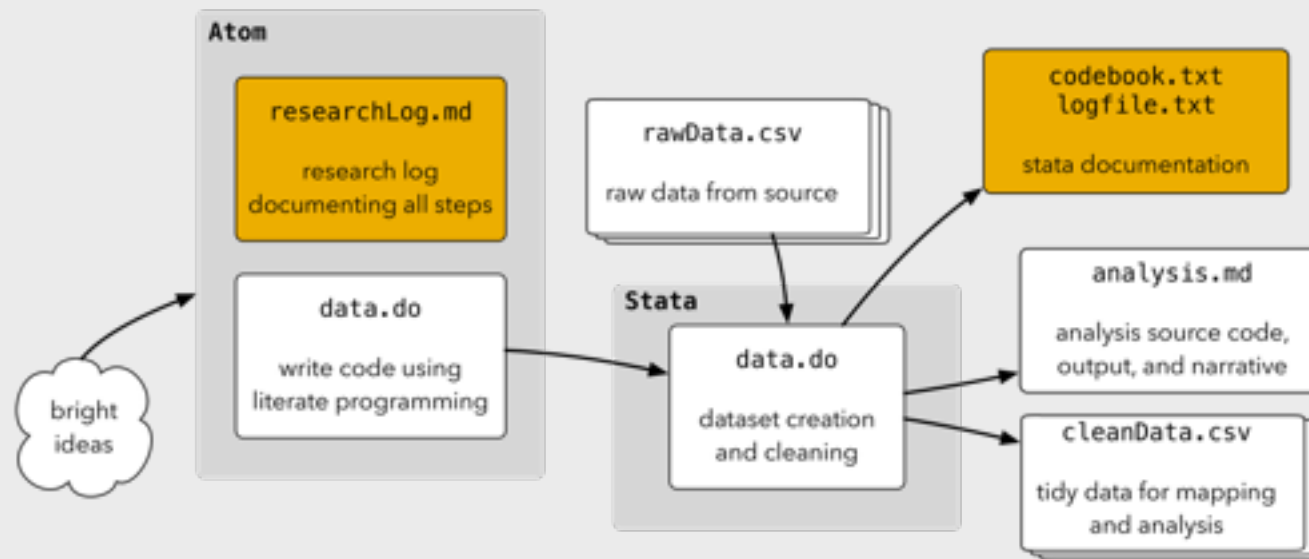
1. planning
2. data collection
organization
3. documentation
4. execution



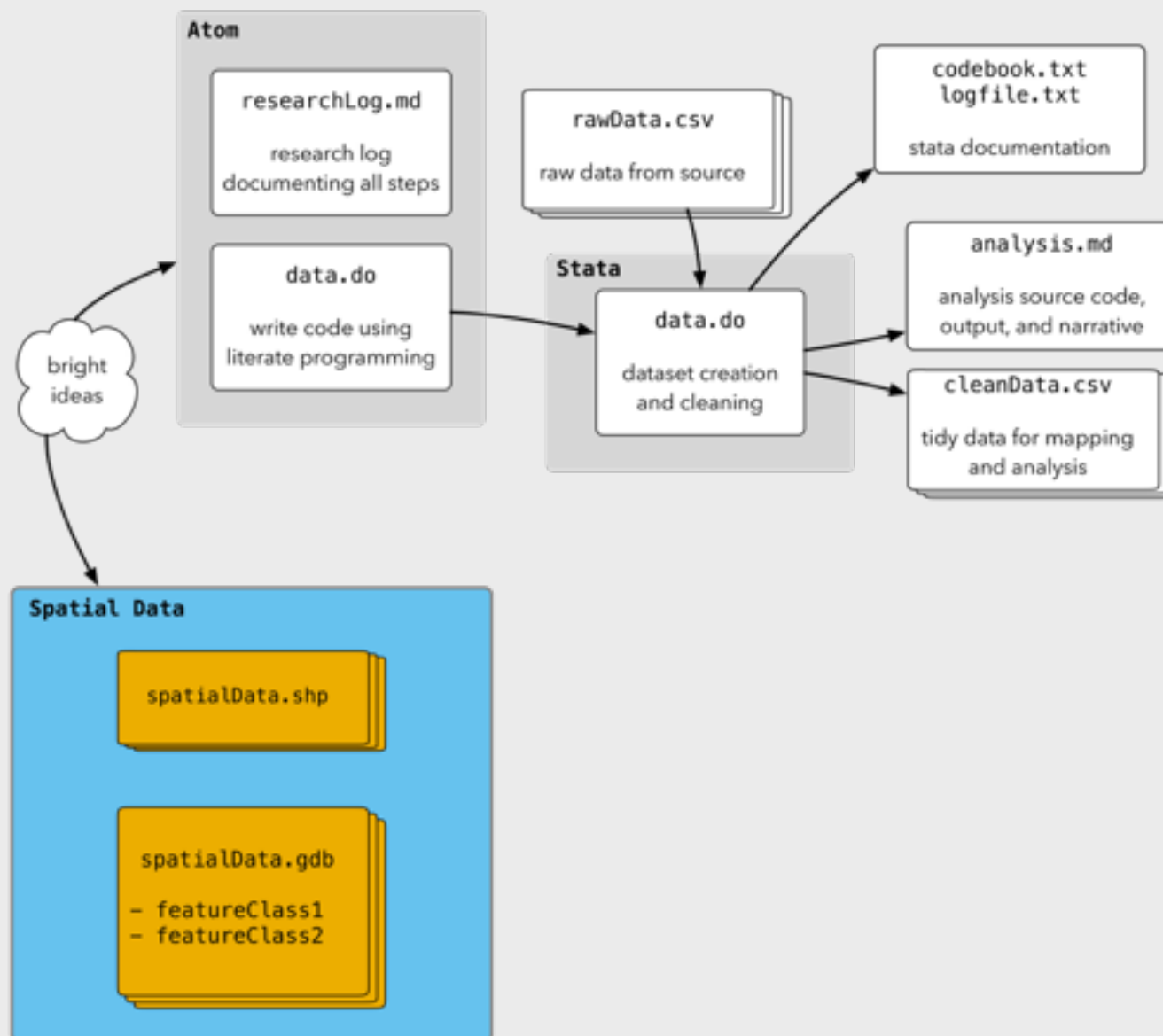
GIS_c WORKFLOW



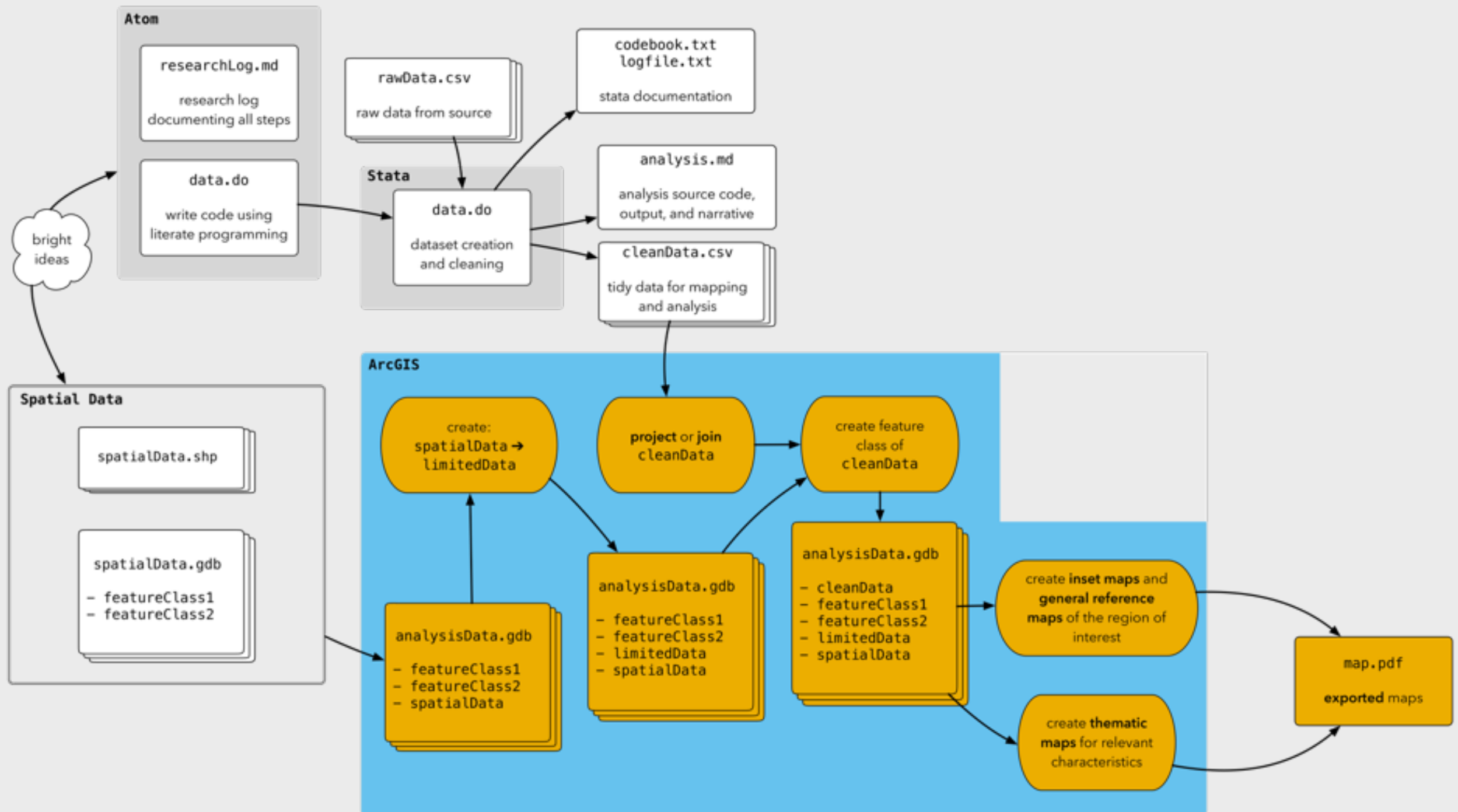
GIS_c WORKFLOW



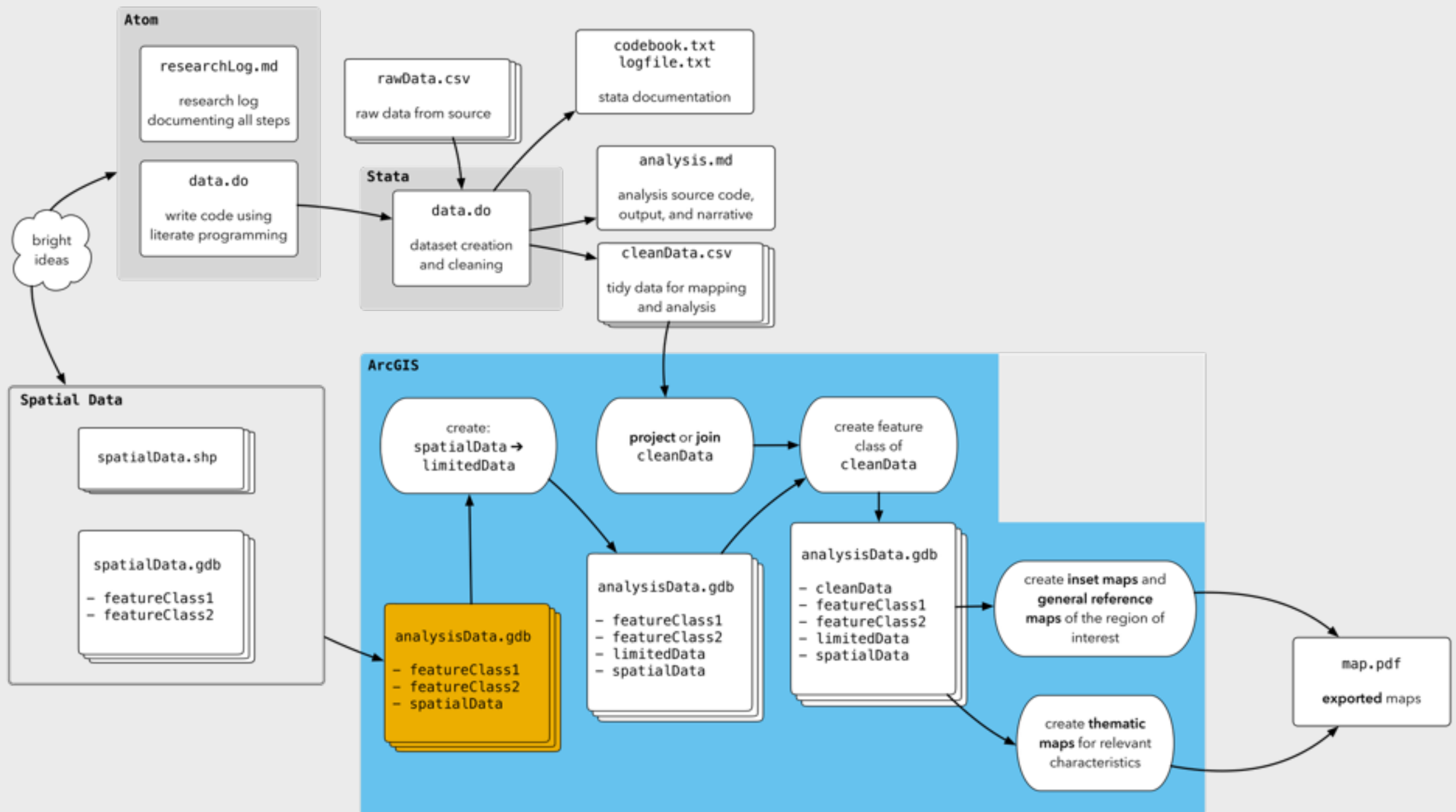
GIS_c WORKFLOW



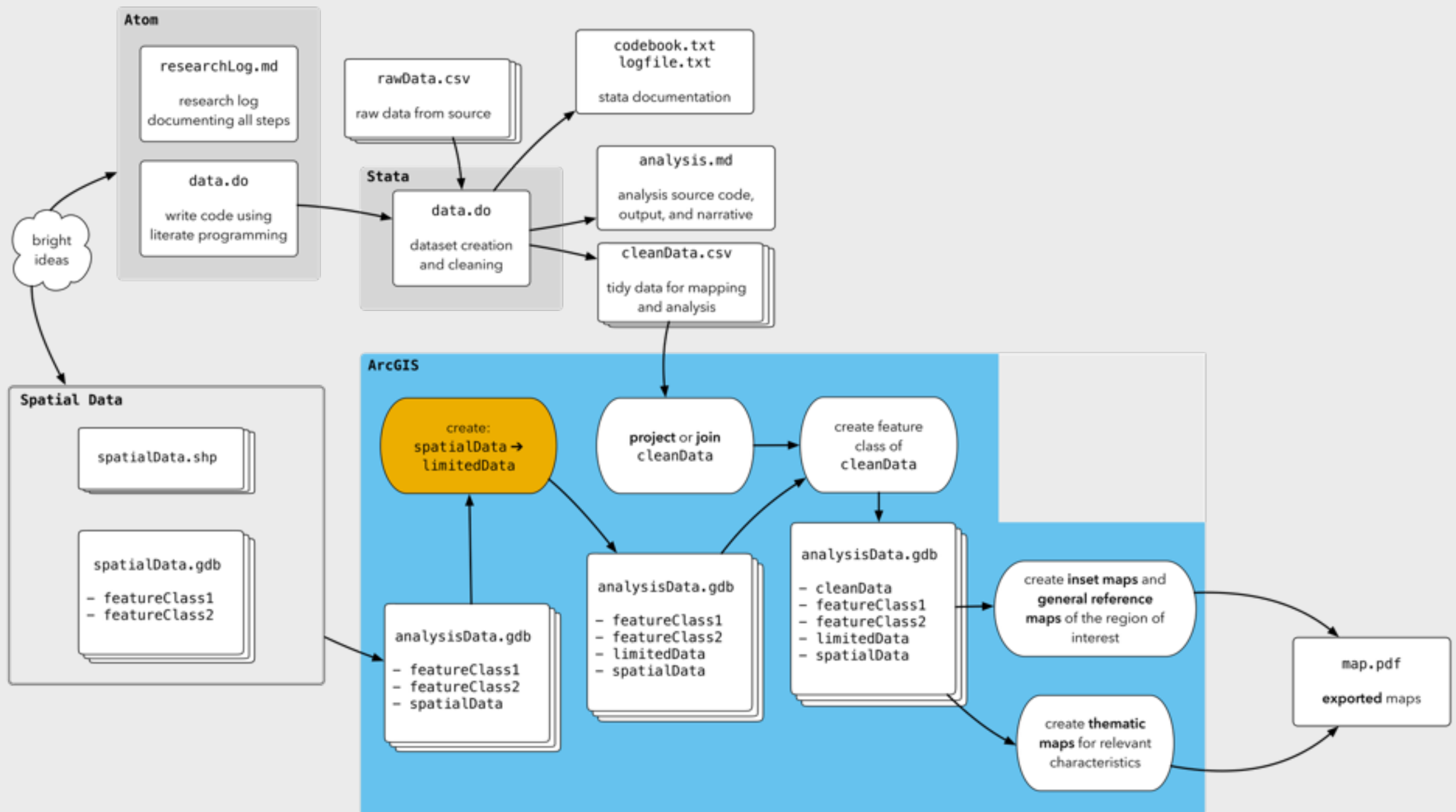
GIS_c WORKFLOW



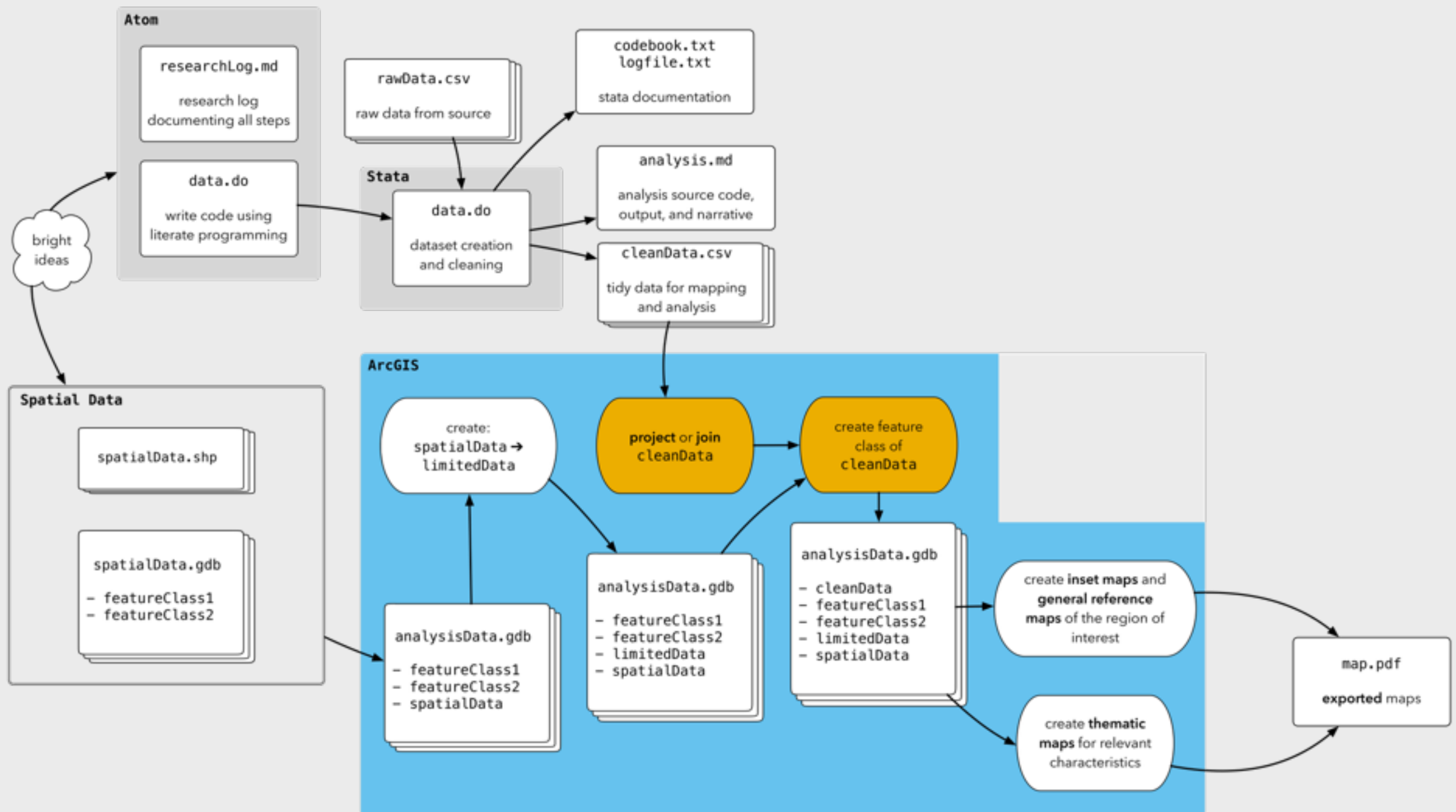
GIS_c WORKFLOW



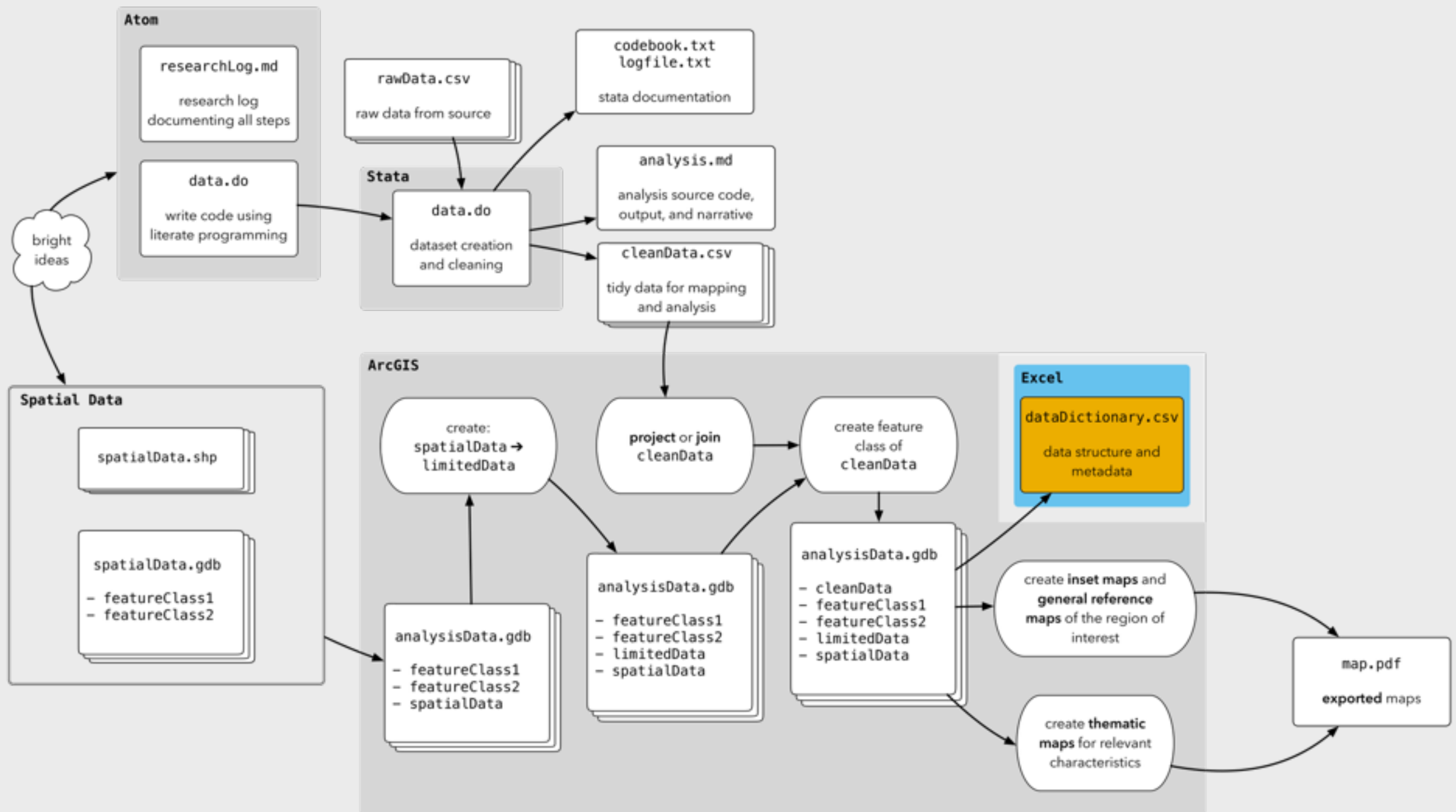
GIS_c WORKFLOW



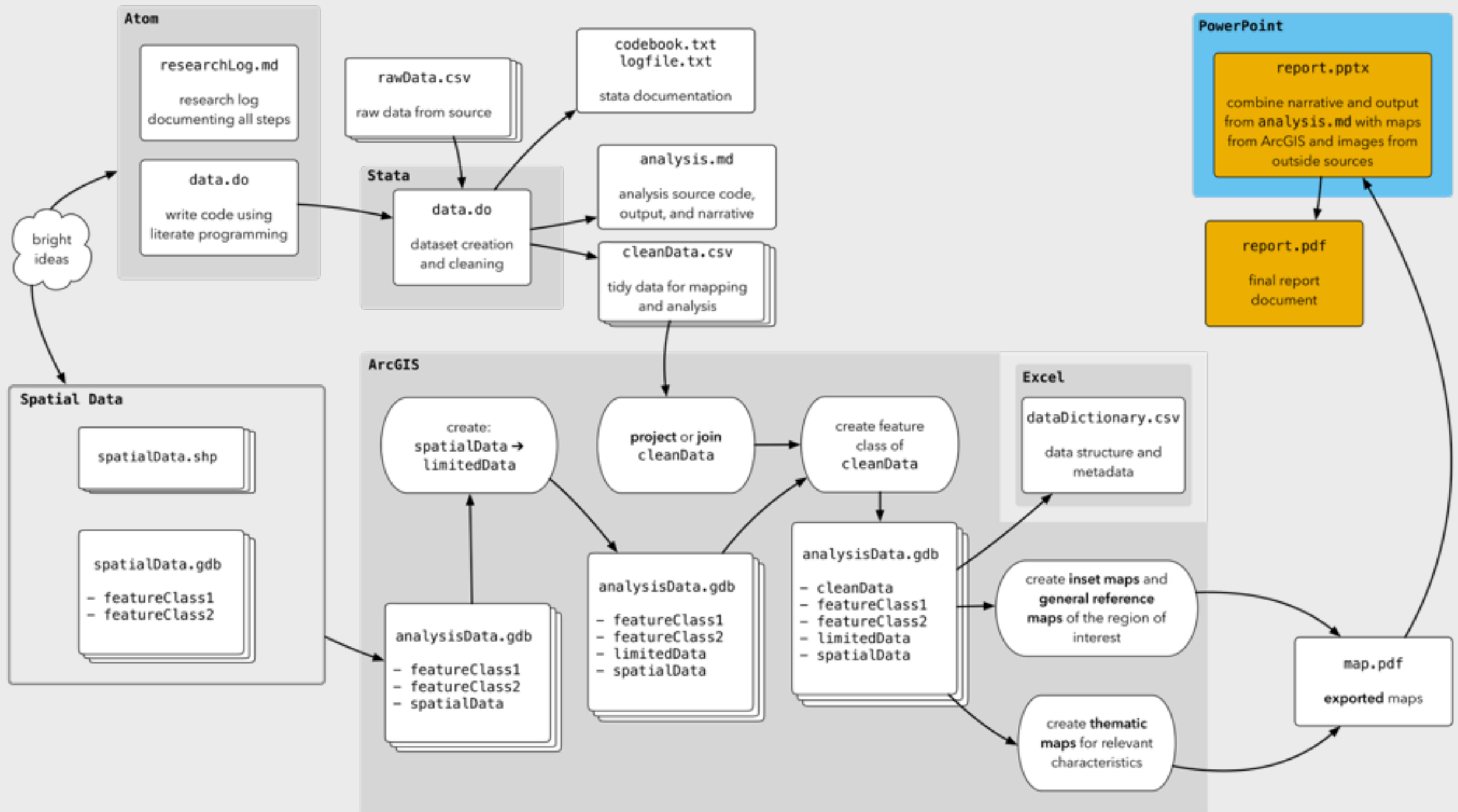
GIS_c WORKFLOW



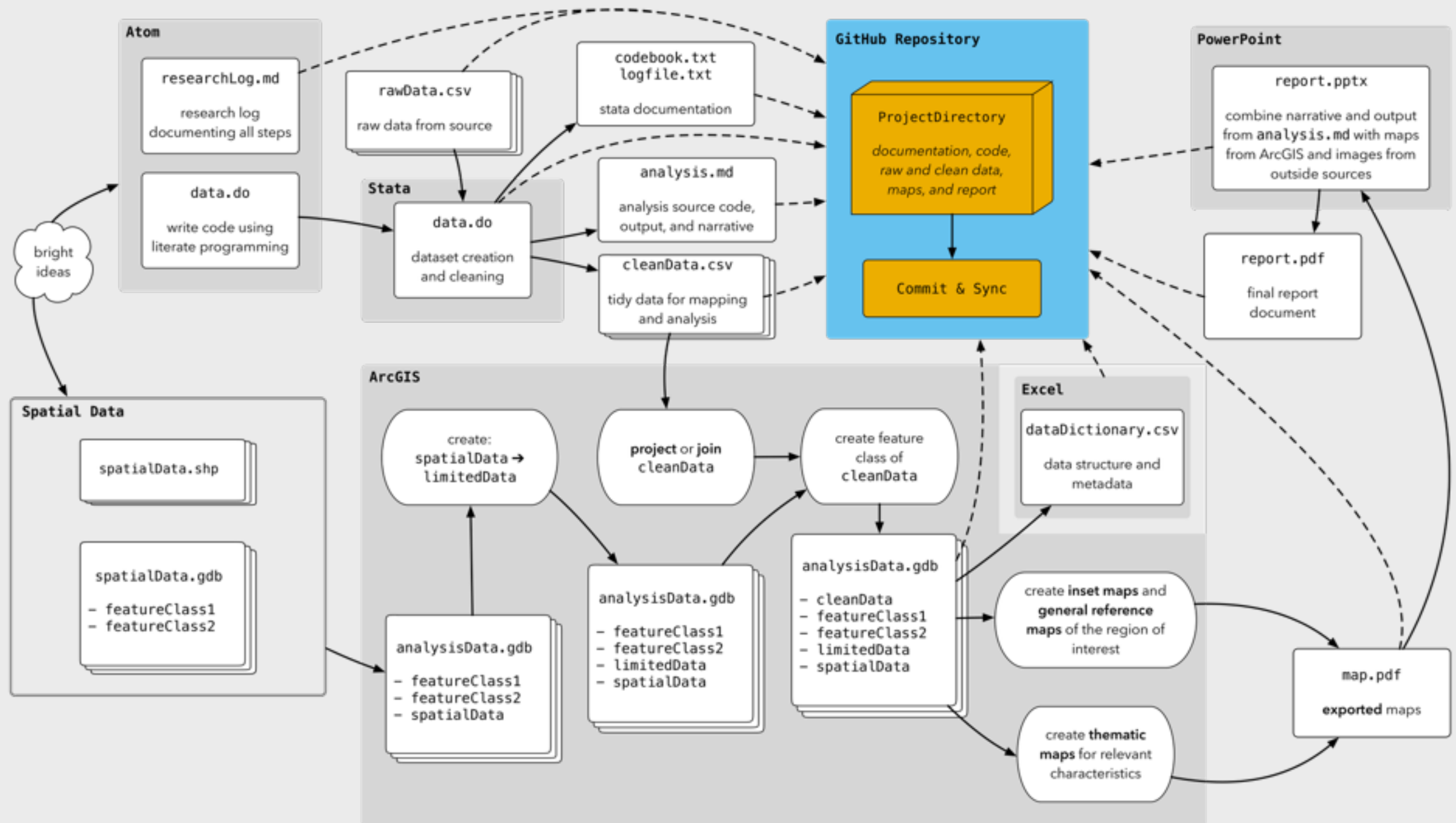
GIS_c WORKFLOW



GIS_c WORKFLOW

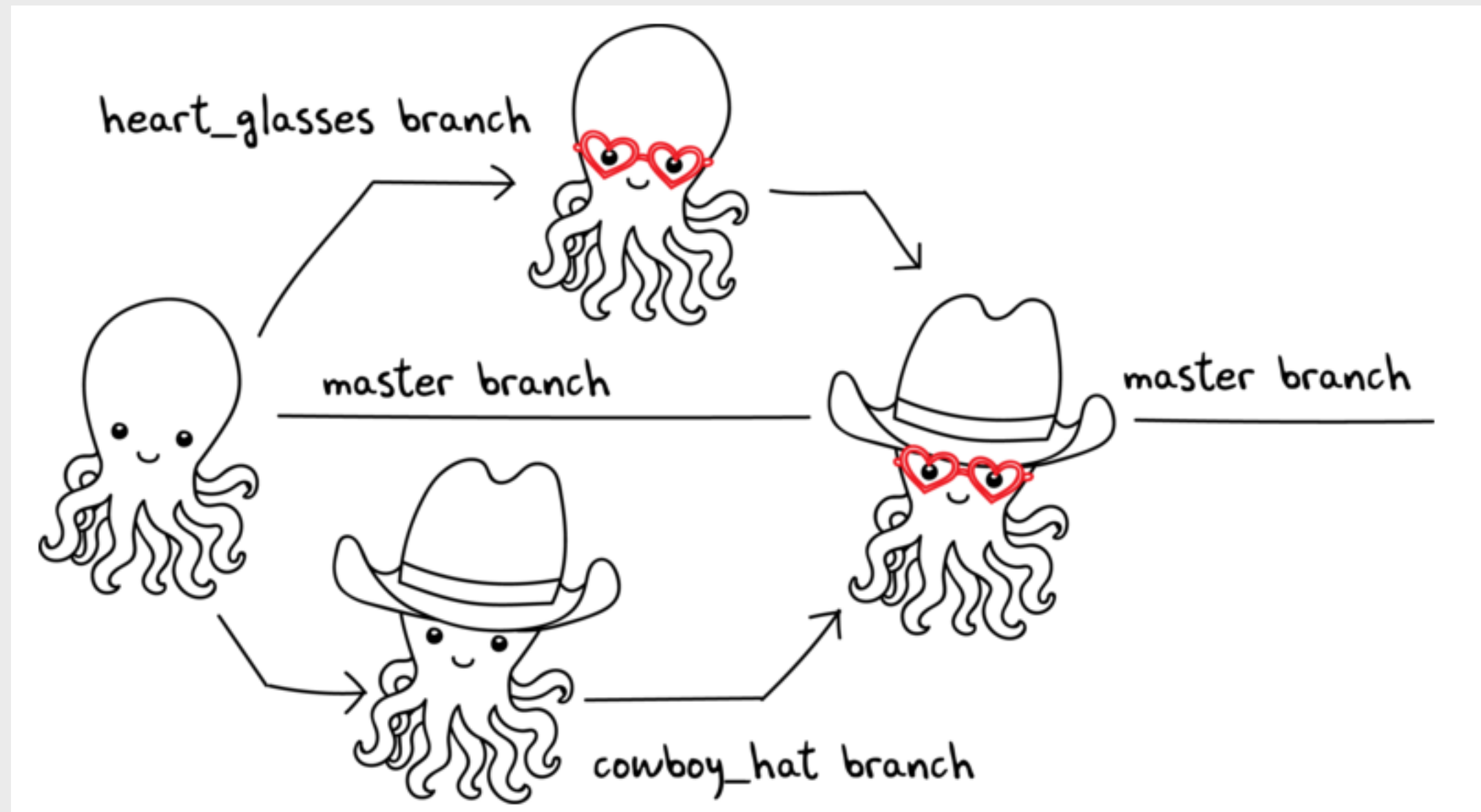


GIS_c WORKFLOW

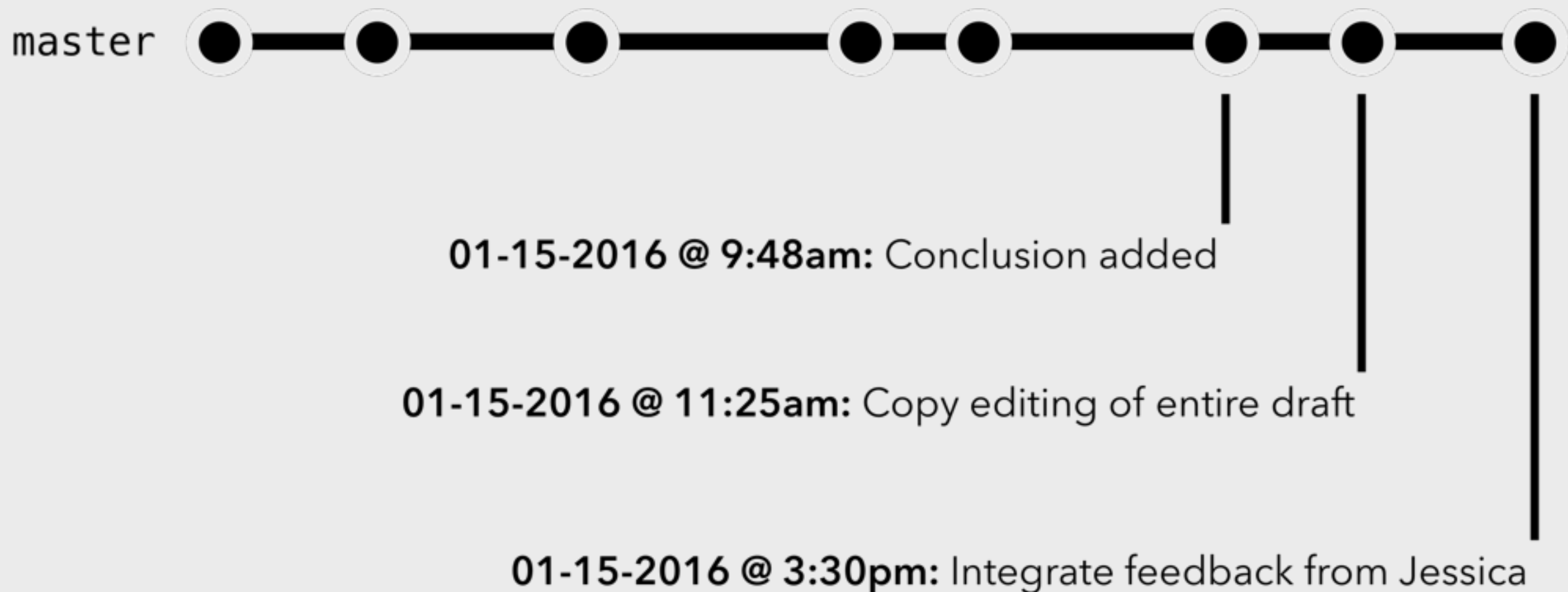


2 ADVANCED GITHUB

BRANCHING & THE FINAL PROJECT

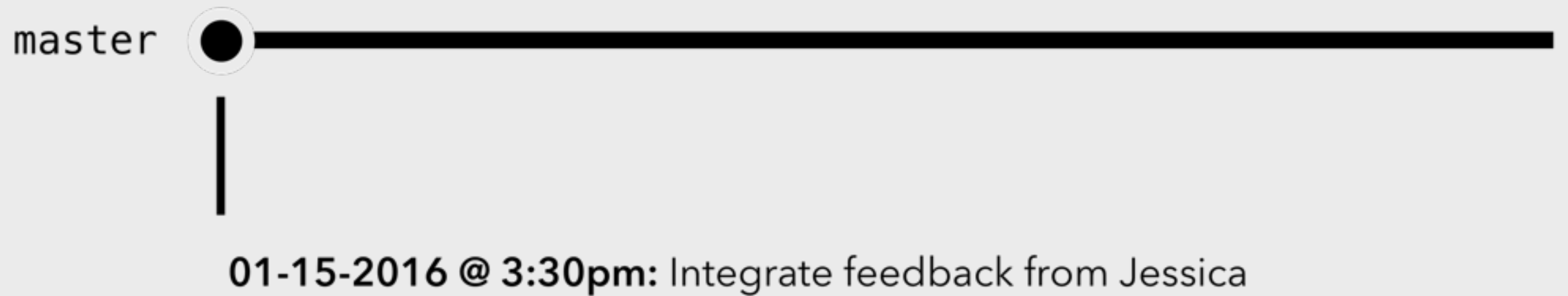


BRANCHING

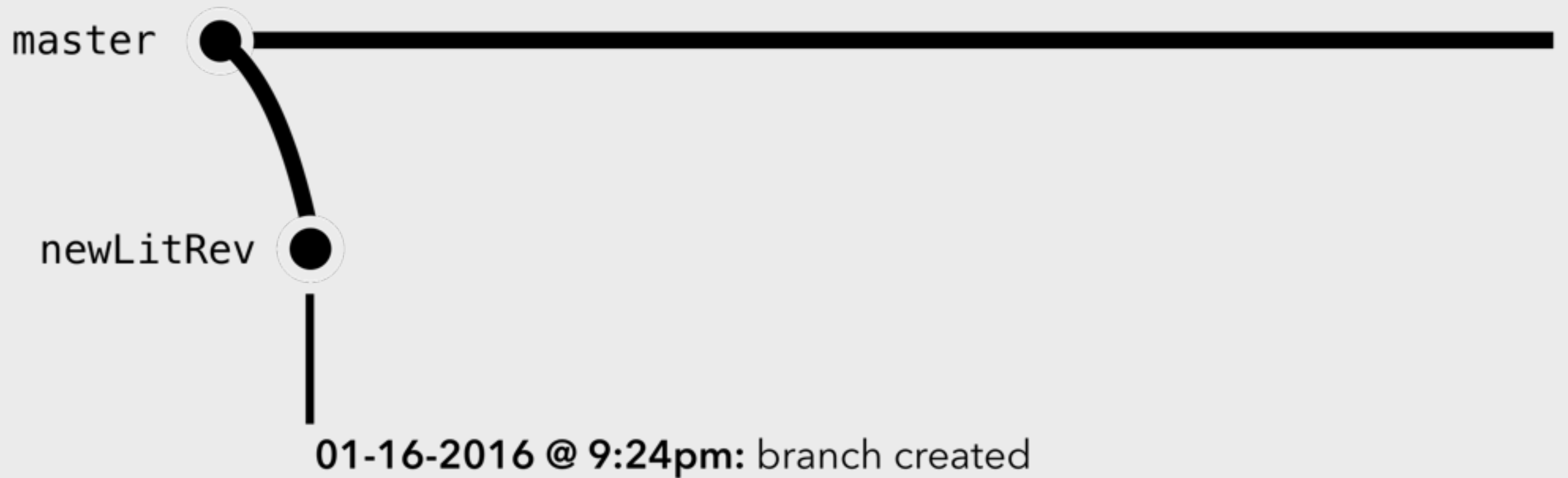


2. ADVANCED GITHUB

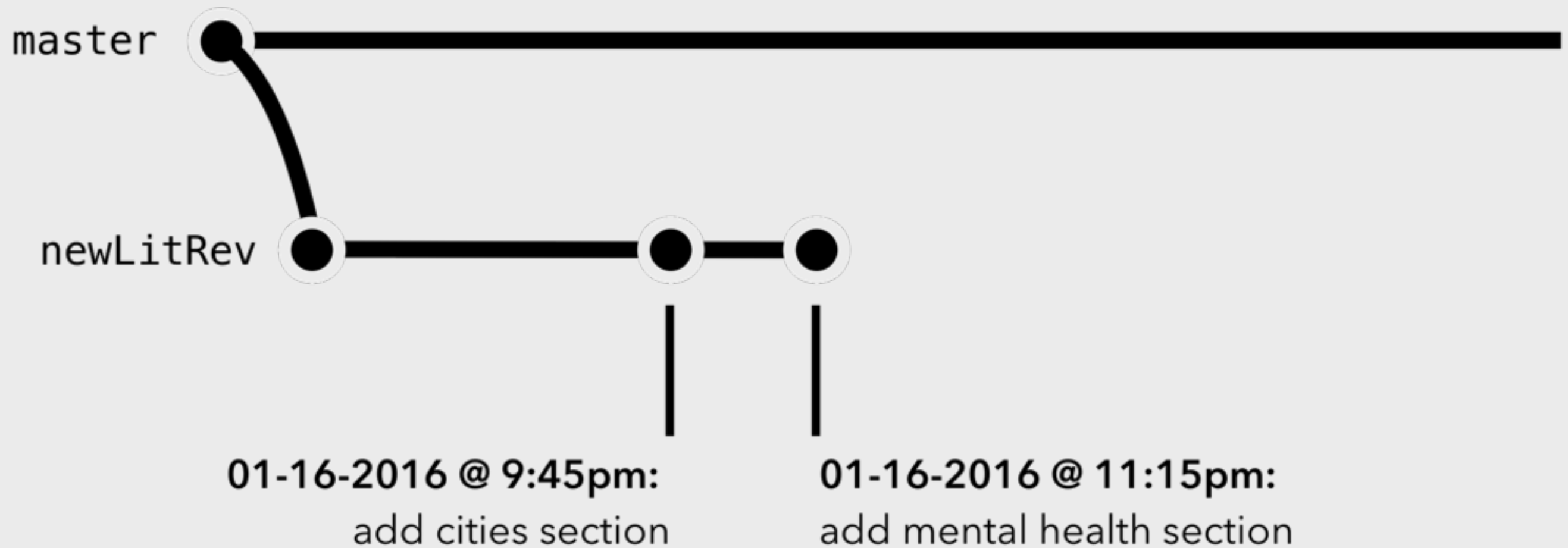
BRANCHING



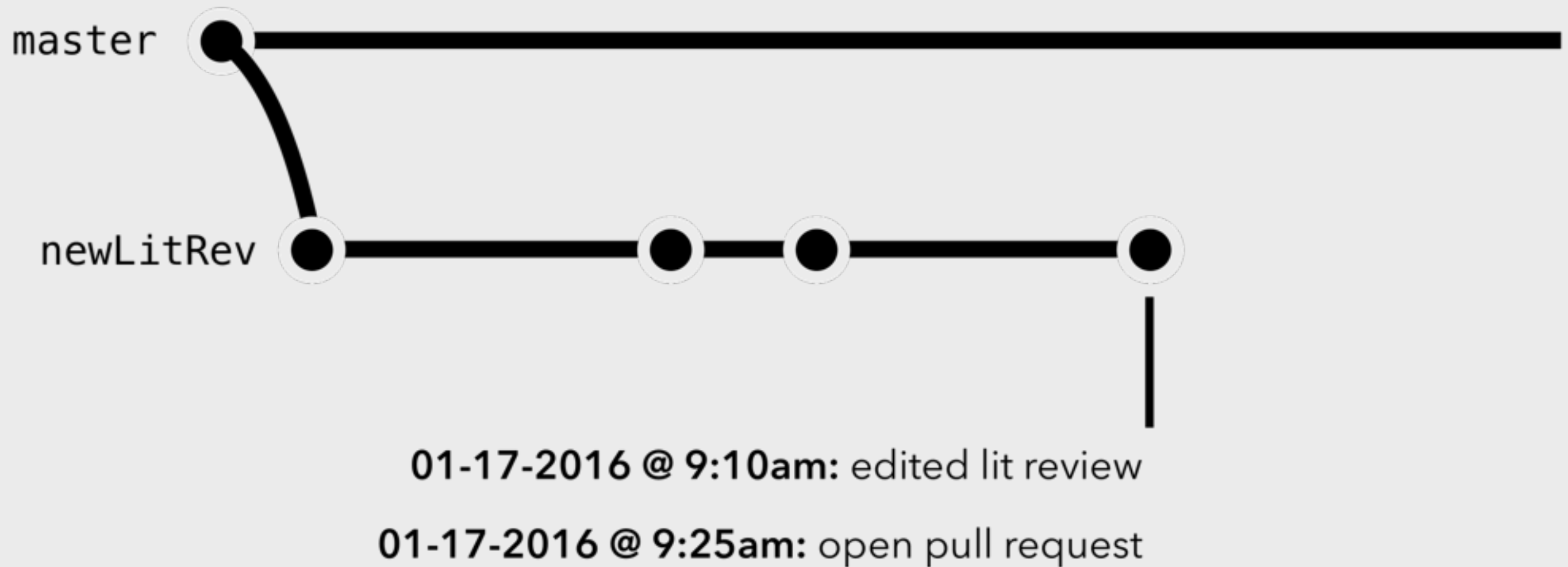
BRANCHING



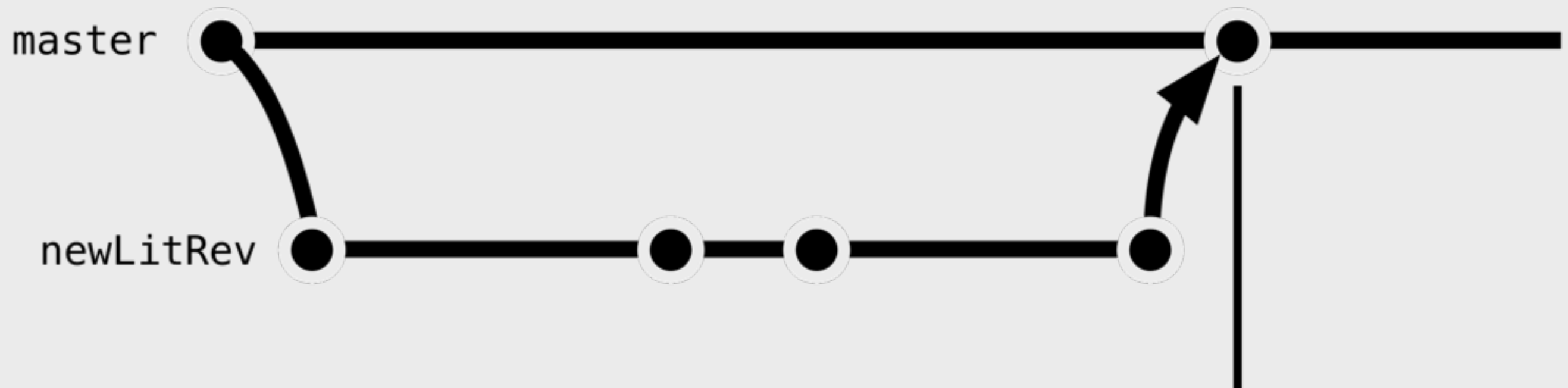
BRANCHING



BRANCHING

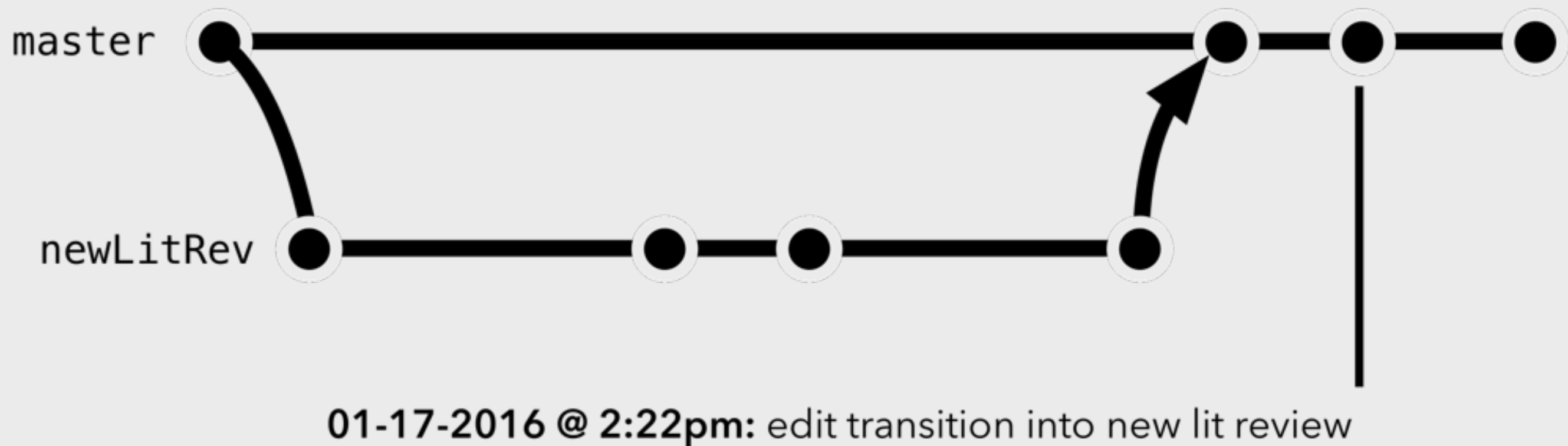


BRANCHING

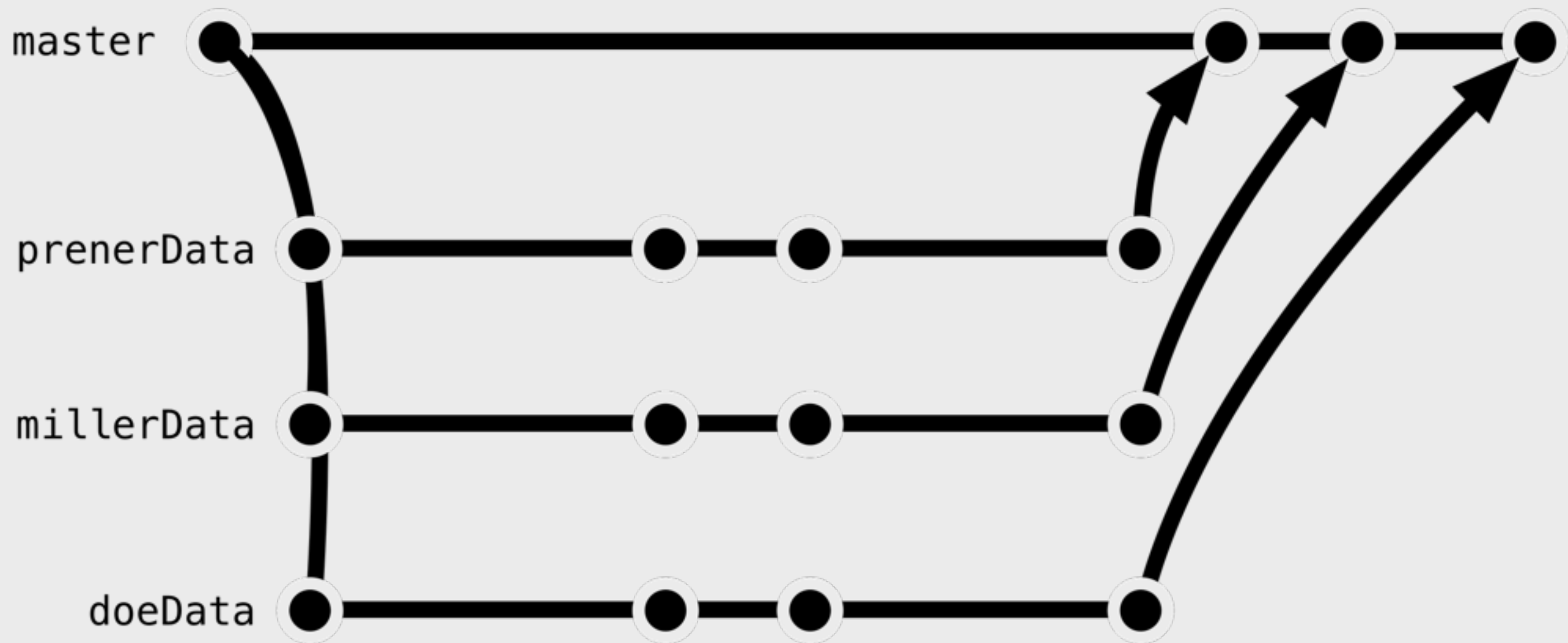


01-17-2016 @ 1:30pm: accept pull request, "delete" branch

BRANCHING



BRANCHING



3 CREATING NEW VARIABLES

GOALS – ‘TIDY’ DATA

- ▶ “Each variable forms a column” with a unique, simple, intuitive name
- ▶ “Each observation forms a row” with no duplicates
- ▶ “Each type of observational unit forms a table”
- ▶ In GISc, it is important to understand if duplicates are truly duplicates, how that will impact mapping
- ▶ Values should be clearly defined and consistently applied
- ▶ Datasets should be subset if possible to make management easier and to condense the size of shapefiles down the road

TWO STRATEGIES

1. "Copy and Recode" - create a duplicate version of the variable and then rearrange the values. This works well for continuous variables that are being recoded into ordinal or binary variables.
 - `generate newVar = oldVar`
 - `recode newVar (1 = 0/10) (2 = 11/20) (3 = 21/30)`

TWO STRATEGIES

2. "Create and Replace" - create a new, empty variable and then replace it with specific values based on another variable. This works well for re-ordering categorical variables and working with string variables.

- `generate newVar = .`
- `replace newVar = 1 if oldVar < 11`
- `replace newVar = 2 if oldVar >= 11 & oldVar < 21`
- `replace newVar = 3 if oldVar >= 21 & oldVar < 31`

TWO STRATEGIES

2. "Create and Replace" - create a new, empty variable and then replace it with specific values based on another variable. This works well for re-ordering categorical variables and working with string variables.

- `generate str newVar = ""`
- `replace newVar = "no" if oldVar == "foo"`
- `replace newVar = "yes" if oldVar == "bar"`

TWO STRATEGIES

2. "Create and Replace" - create a new, empty variable and then replace it with specific values based on another variable. This works well for re-ordering categorical variables and working with string variables.

- `generate str newVar = ""`
- `replace newVar = "no" if strpos(oldVar, "foo")`
- `replace newVar = "yes" if strpos(oldVar, "bar")`

4 GEODATABASES

DATA STORAGE OPTIONS



DATA STORAGE OPTIONS



Ideal for storing and sharing single files, but can be inefficient and difficult to store.



Ideal for storing large amounts of spatial data from both organizational and efficiency approaches. Can get large quickly, which may have ramifications.