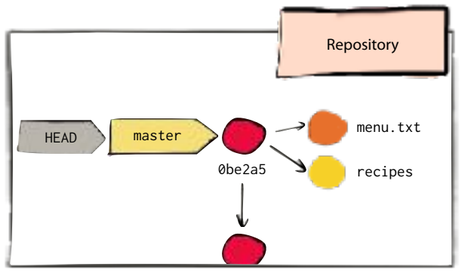
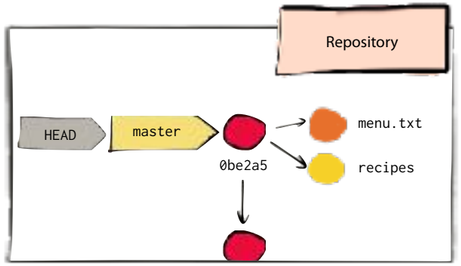
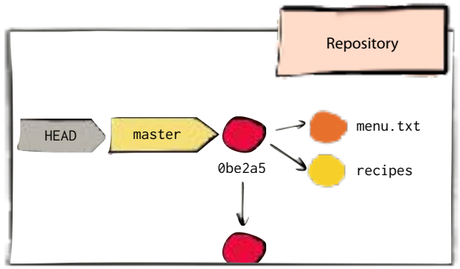
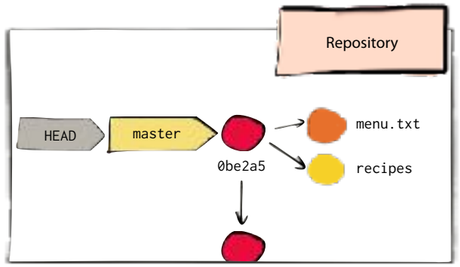
Finding Your Workflow

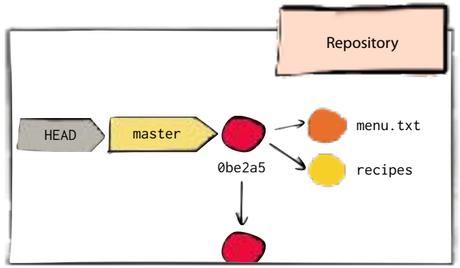
Paolo Perrotta
FREELANCE DEVELOPER
@nusco


Repository
HEAD
master
0be2a5
menu.txt
recipes


Repository
HEAD
master
0be2a5
menu.txt
recipes


Repository
HEAD
master
0be2a5
menu.txt
recipes


Repository
HEAD
master
0be2a5
menu.txt
recipes


Repository
HEAD
master
0be2a5
menu.txt
recipes

Distributed Workflow

Distribution Model

How many repositories do you have? Who can access them? ...

Branching Model

Which branches do you have? How do you use them? ...

Constraints

Do you merge or do you rebase? Can you push unstable code? ...

This is a module about Git workflow patterns.

What You Need to Know

What a Remote Is

Pushing to a Remote

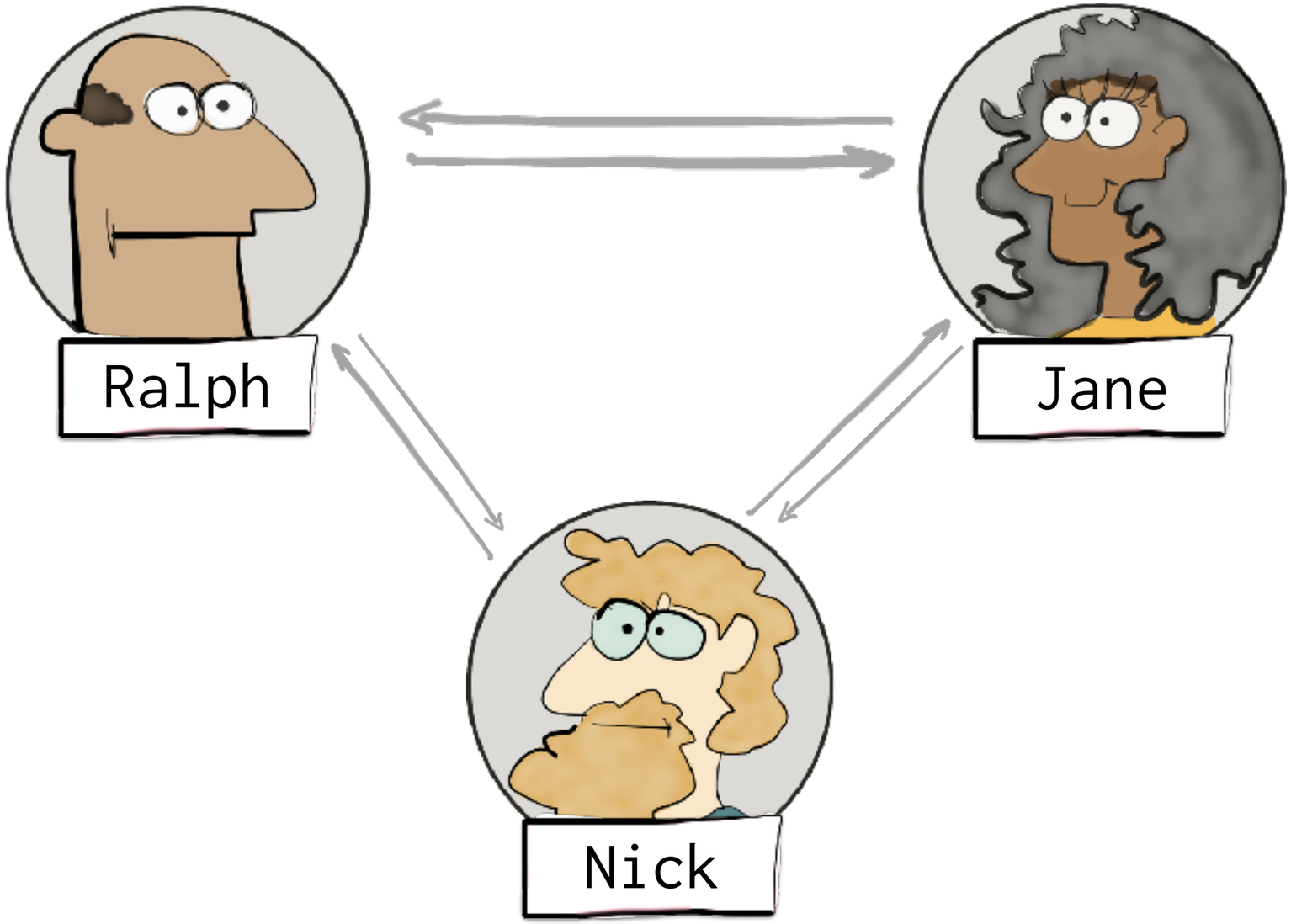
Pulling from a Remote

Distributed Workflow

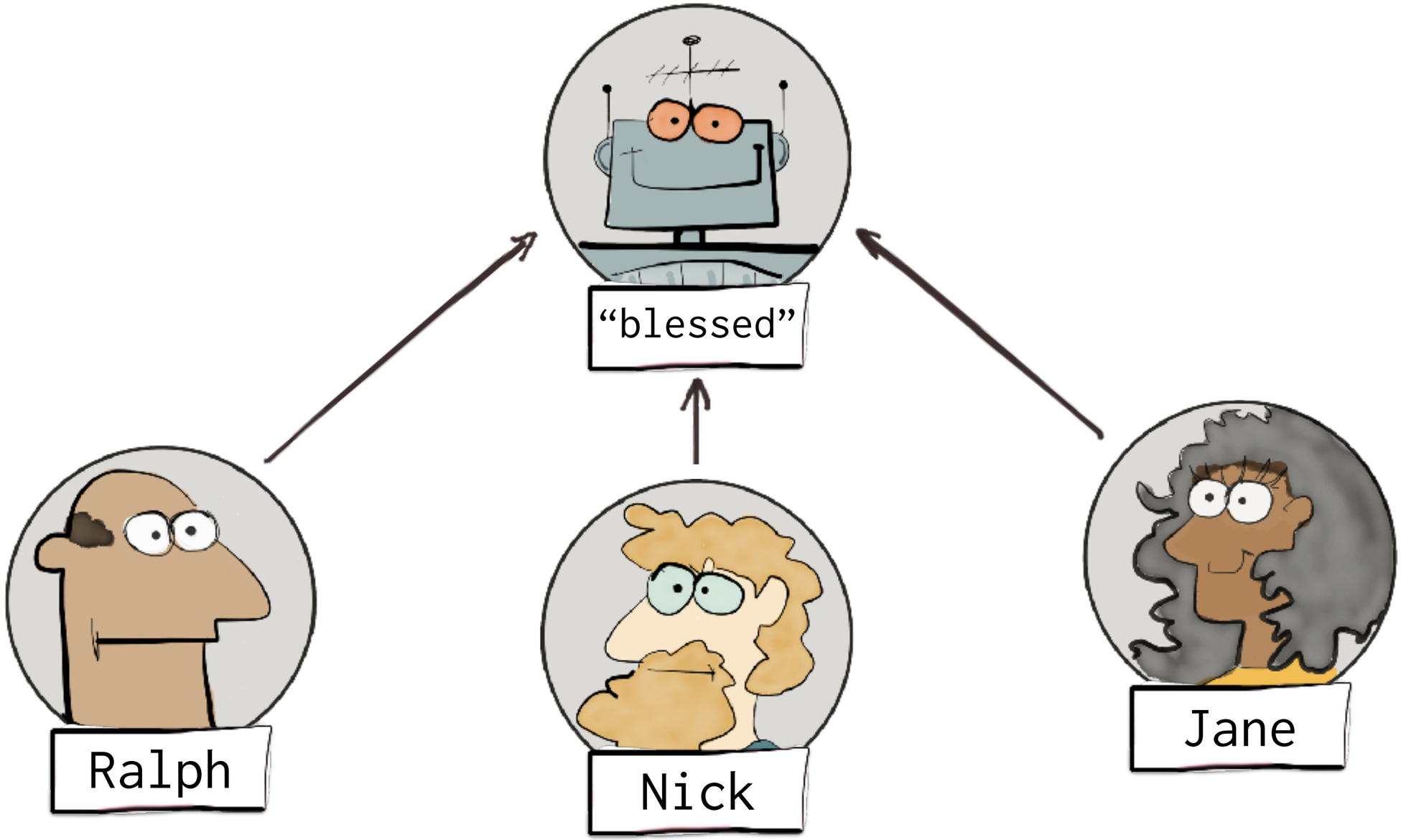
Distribution Model

How many repositories do you have? Who can access them? ...

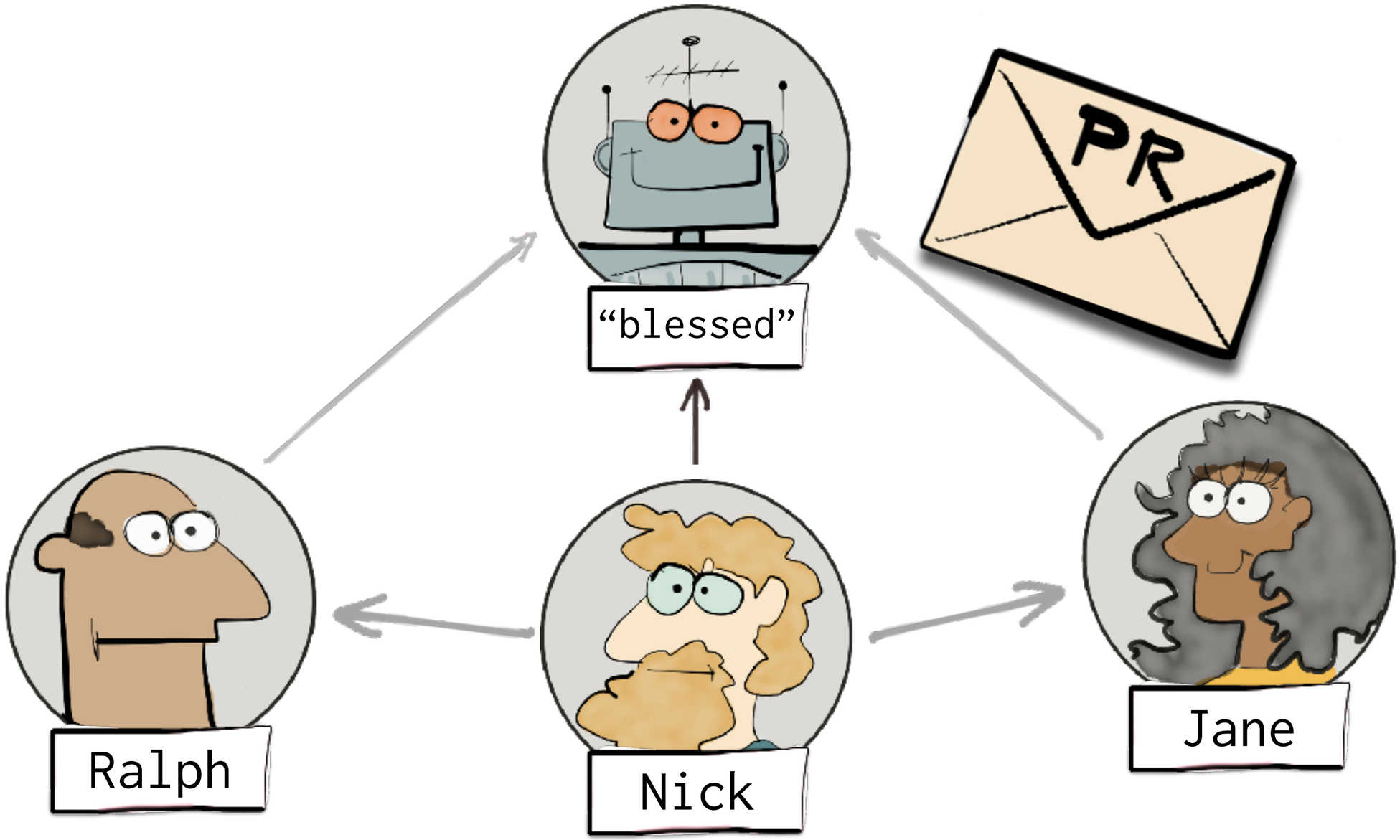
Peer to Peer Model



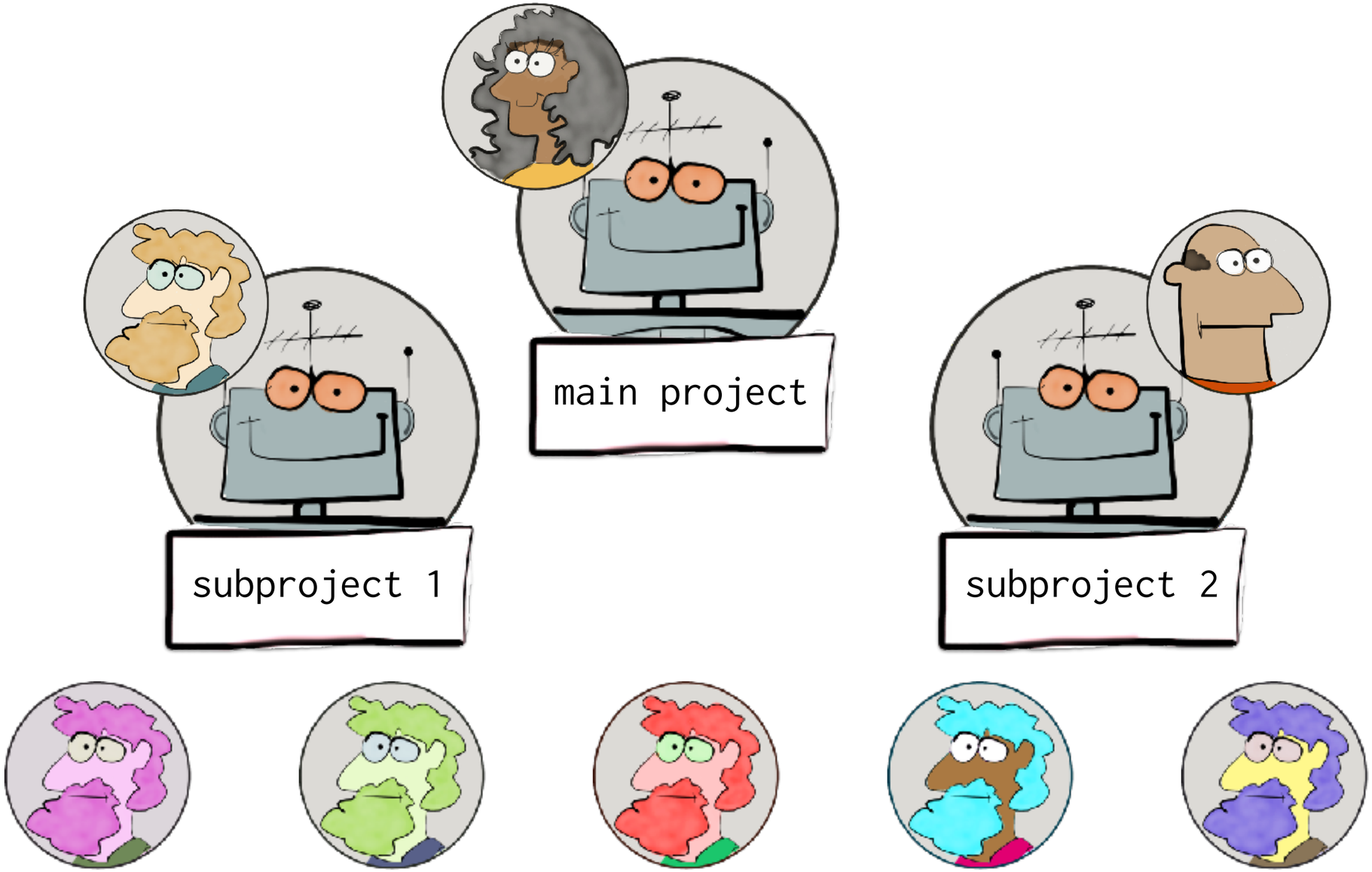
Centralized Model



Pull Request Model



Dictator and Lieutenants Model



Distribution Models

Peer to Peer Model

Centralized Model

Pull Request Model

Dictator and Lieutenants Model

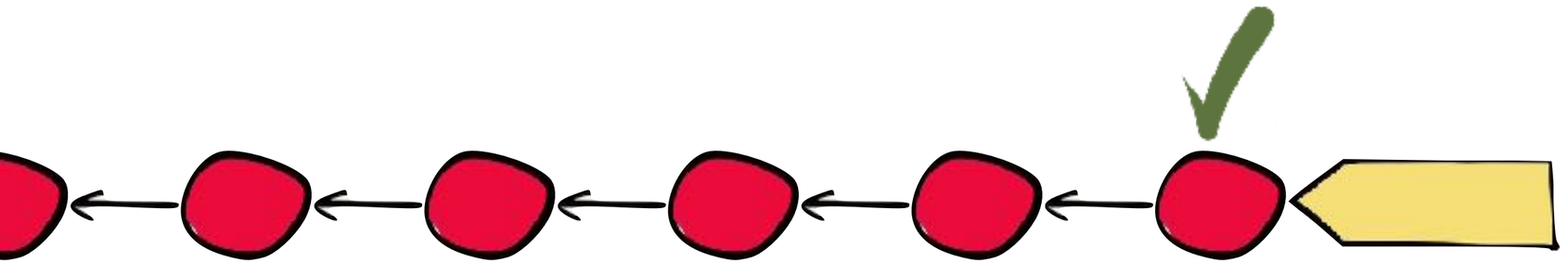
Many projects use a mixed Distribution Model.

Distributed Workflow

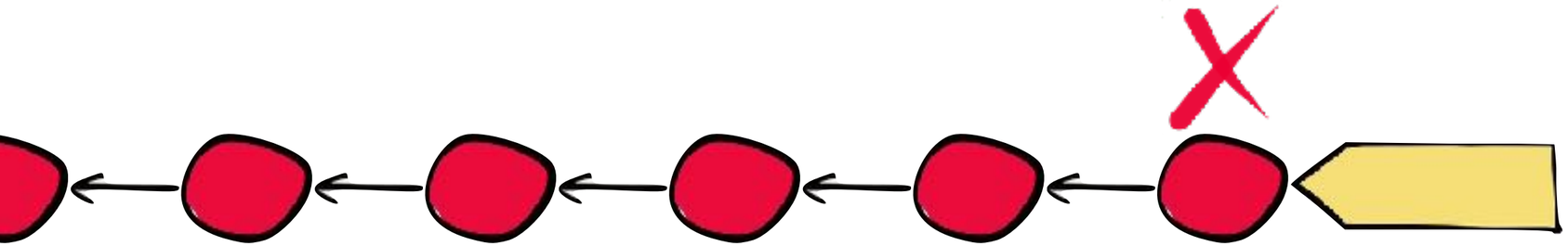
Branching Model

Which branches do you have? How do you use them? ...

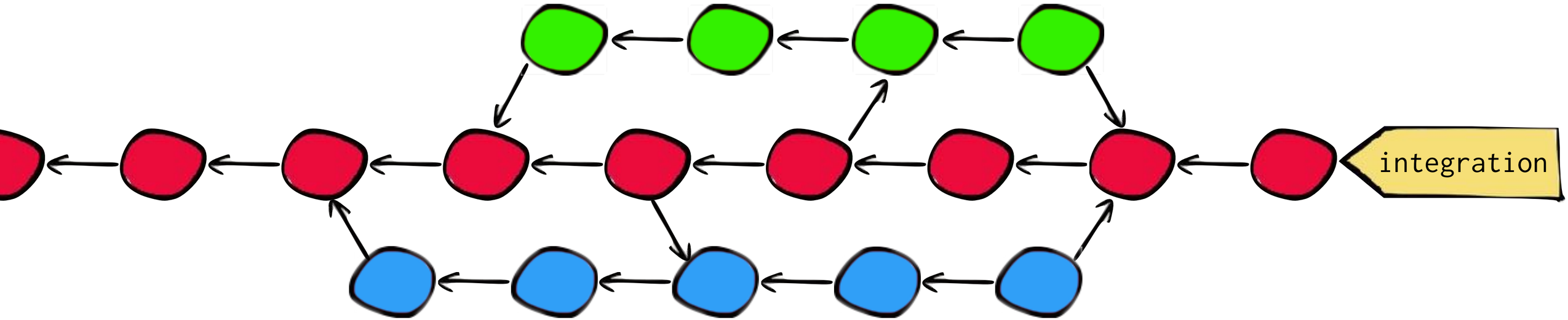
Stable Branch



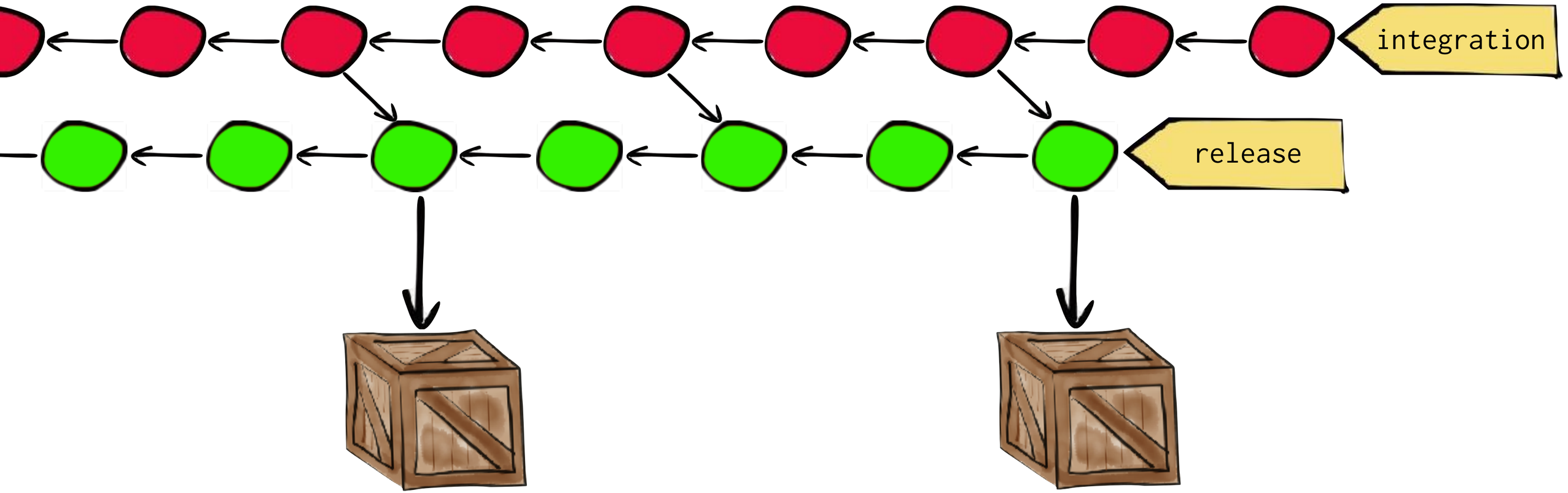
Unstable Branch



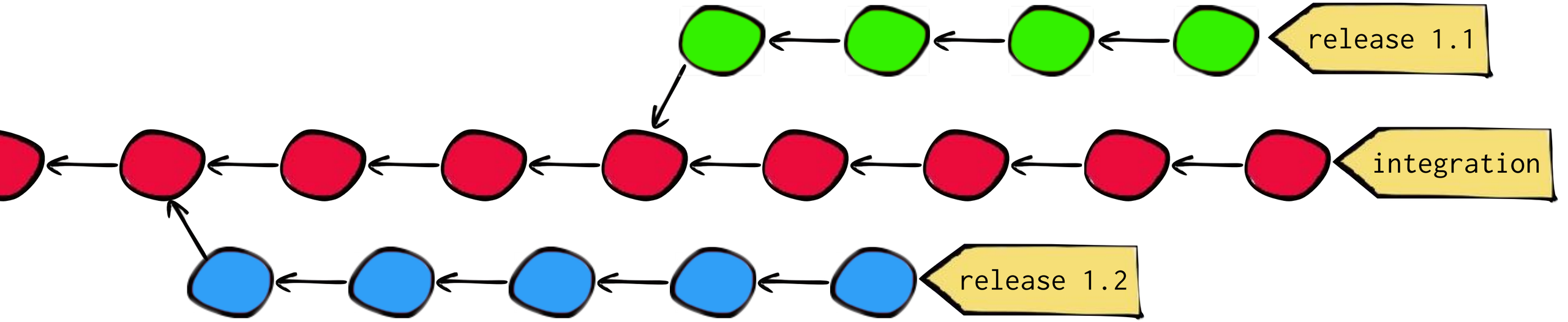
Integration Branch



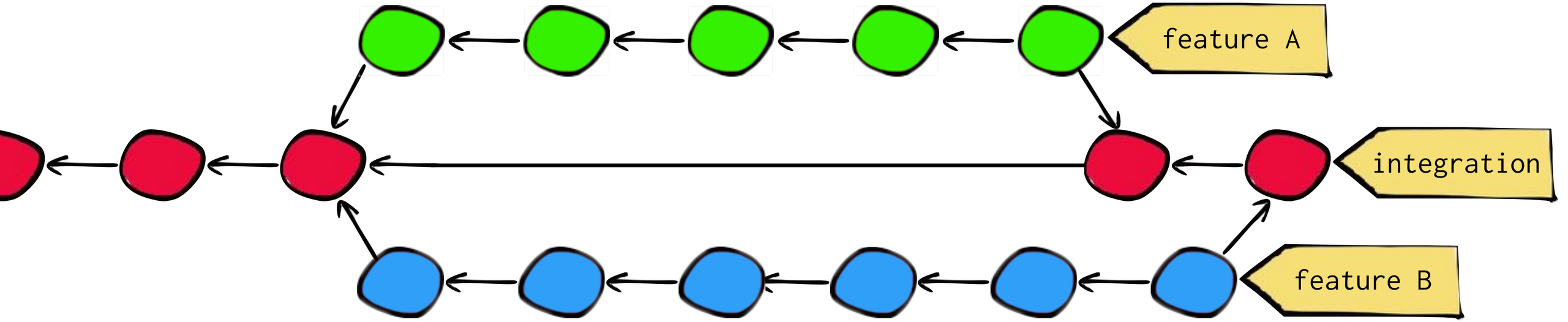
Release Branch

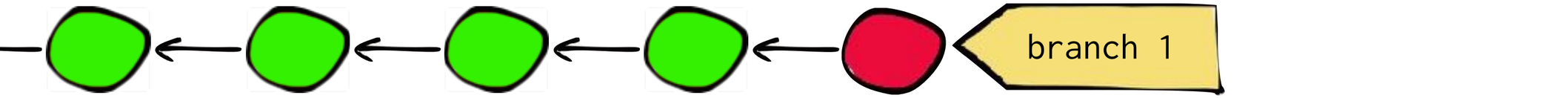


Release Branch

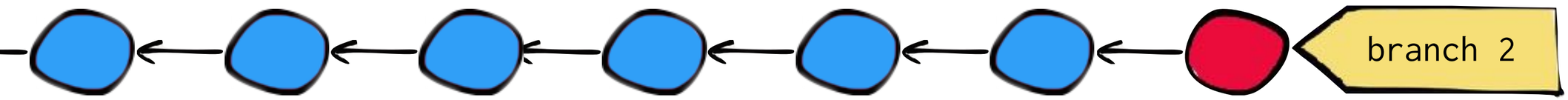


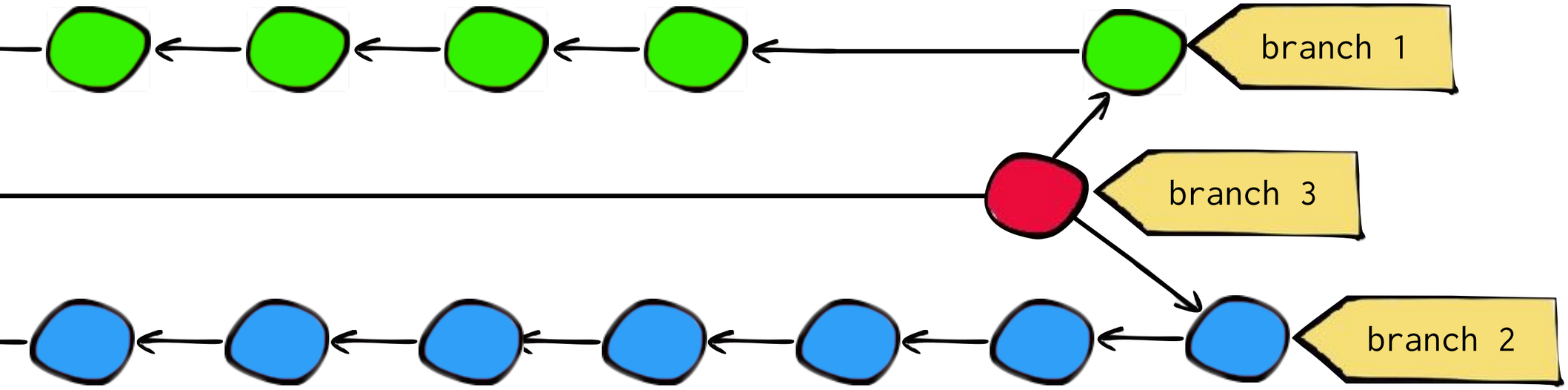
Feature Branch



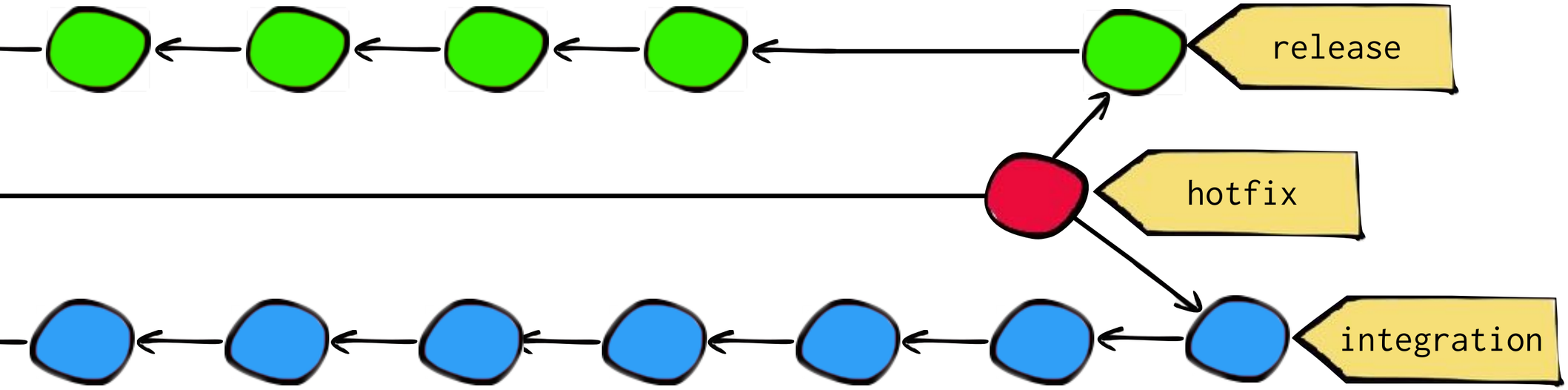


cherry-pick




branch 1
branch 3
branch 2

Hotfix Branch



Branching Models

Stable and unstable branches

Common Branches

- Integration Branch
- Release Branch
- Feature Branch
- Hotfix Branch

Distributed Workflow

Constraints

Do you merge or do you rebase? Can you push unstable code? ...

A Few Examples of Constraints

rebase, don't *merge* /
merge, don't *rebase*

Only developer *X*
can do *Y* on branch *Z*

Don't push to a red build

Squash a feature to a single commit before you merge it to *master*

The Elements of a Distributed Workflow

Distribution Model

Branching Model

Constraints

Don’t “just use GitFlow”.

Don’t design a workflow.
Instead, grow it.

A Simple Workflow to Start With

Distribution Model

- Centralized

Branching Model

- One Integration Branch (*master*)
- One Feature Branch per Feature

Constraints

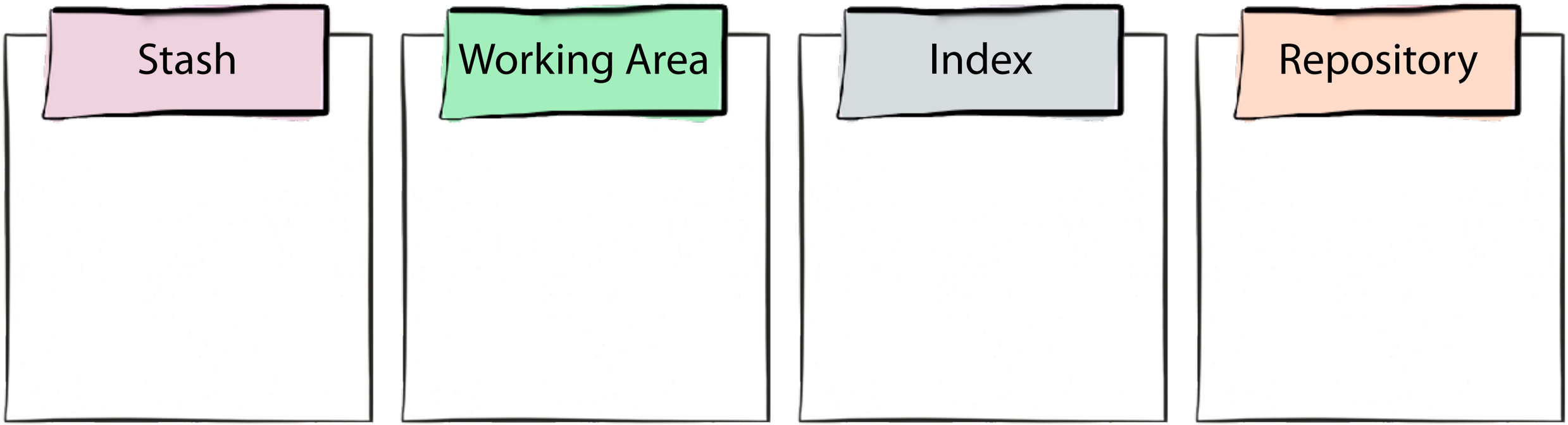
- Keep *master* stable, fix it ASAP if it breaks
- Integrate Feature Branches every few days
- Use *merge* over *rebase* by default

“Simple, clear purpose and principles give rise to complex and intelligent behavior.

Complex rules and regulations give rise to simple and stupid behavior.”

Dee Hock

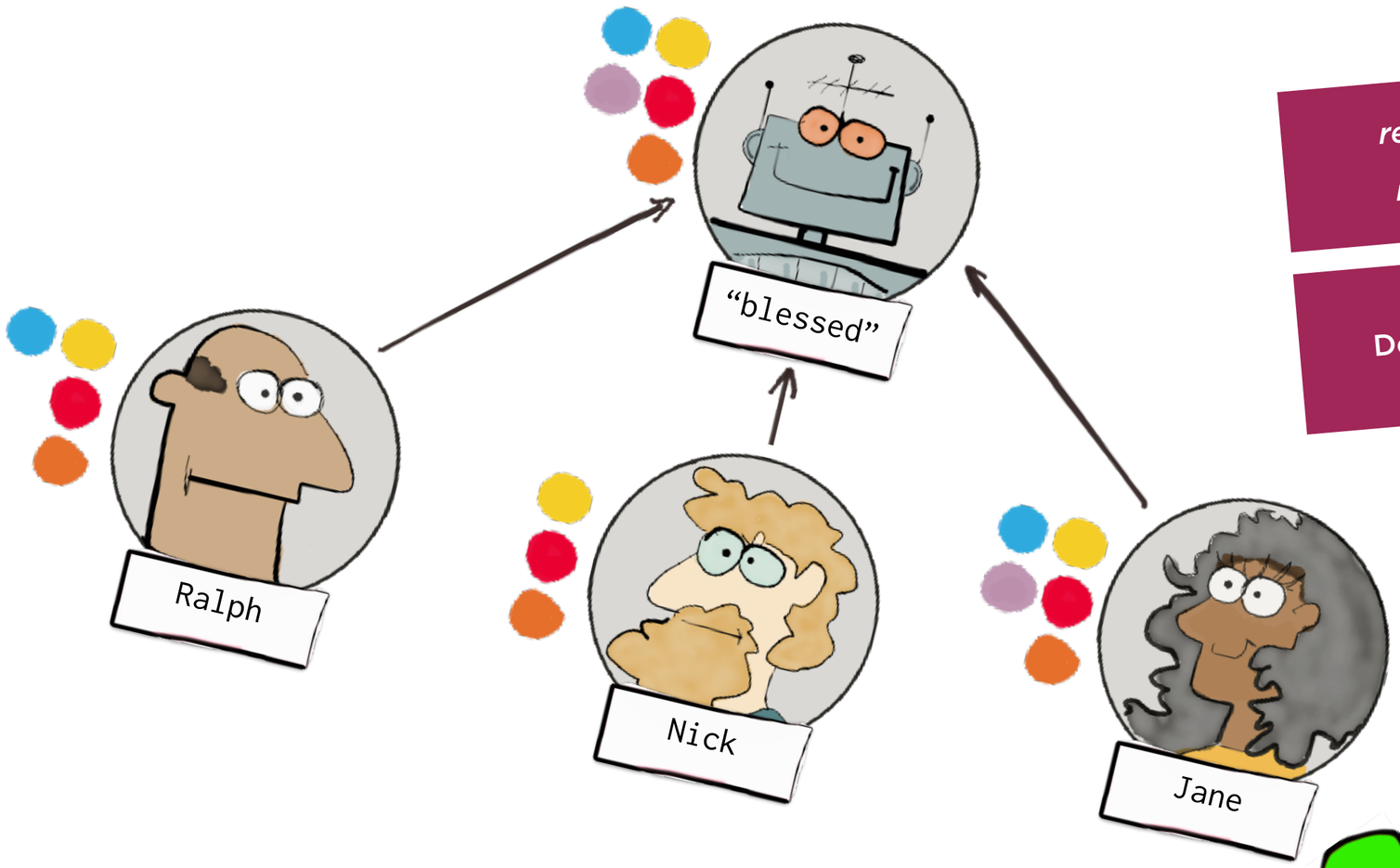
The Four Areas



The Two Questions

How does this command move information across the Four Areas?

How does this command change the Repository?

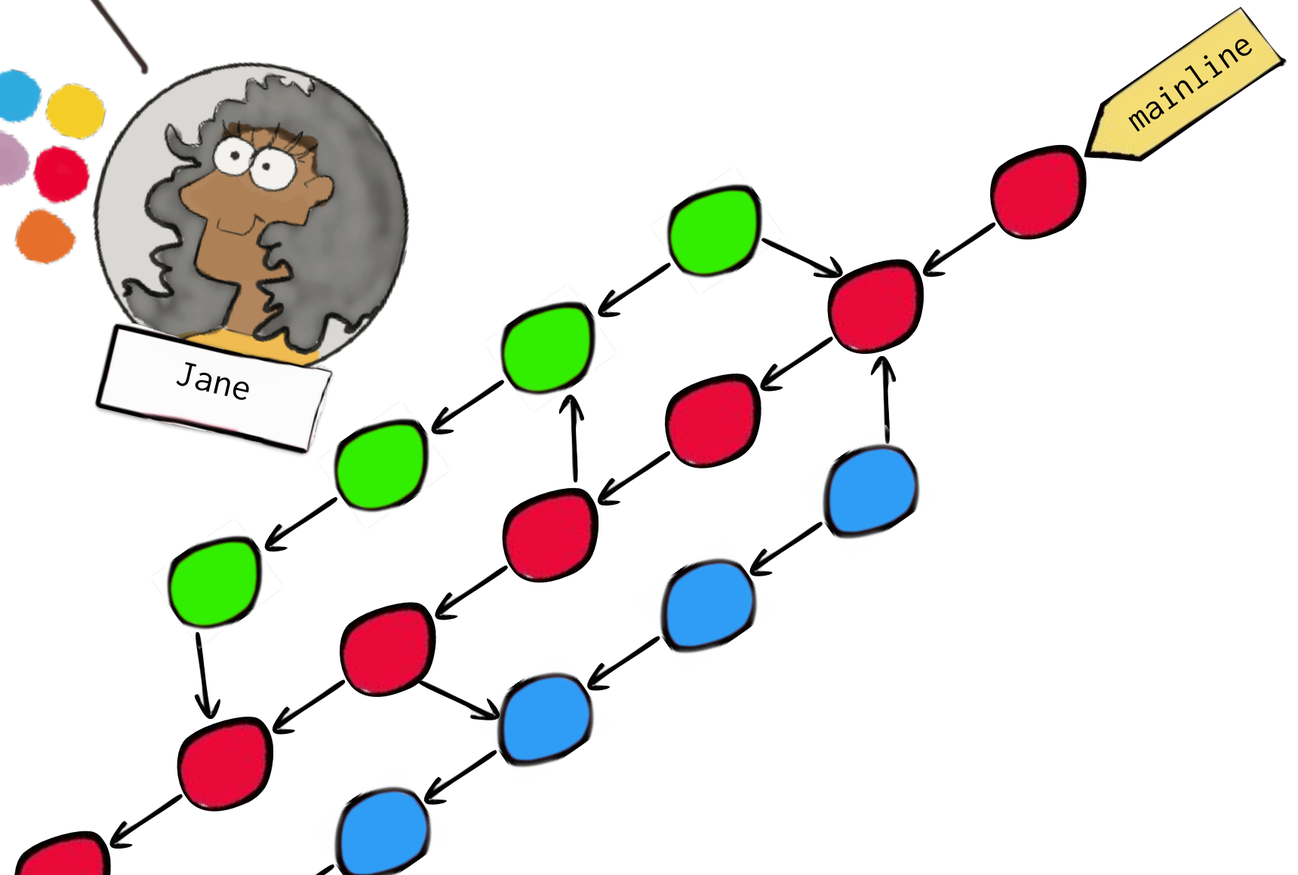


rebase, don't *merge* /
merge, don't *rebase*

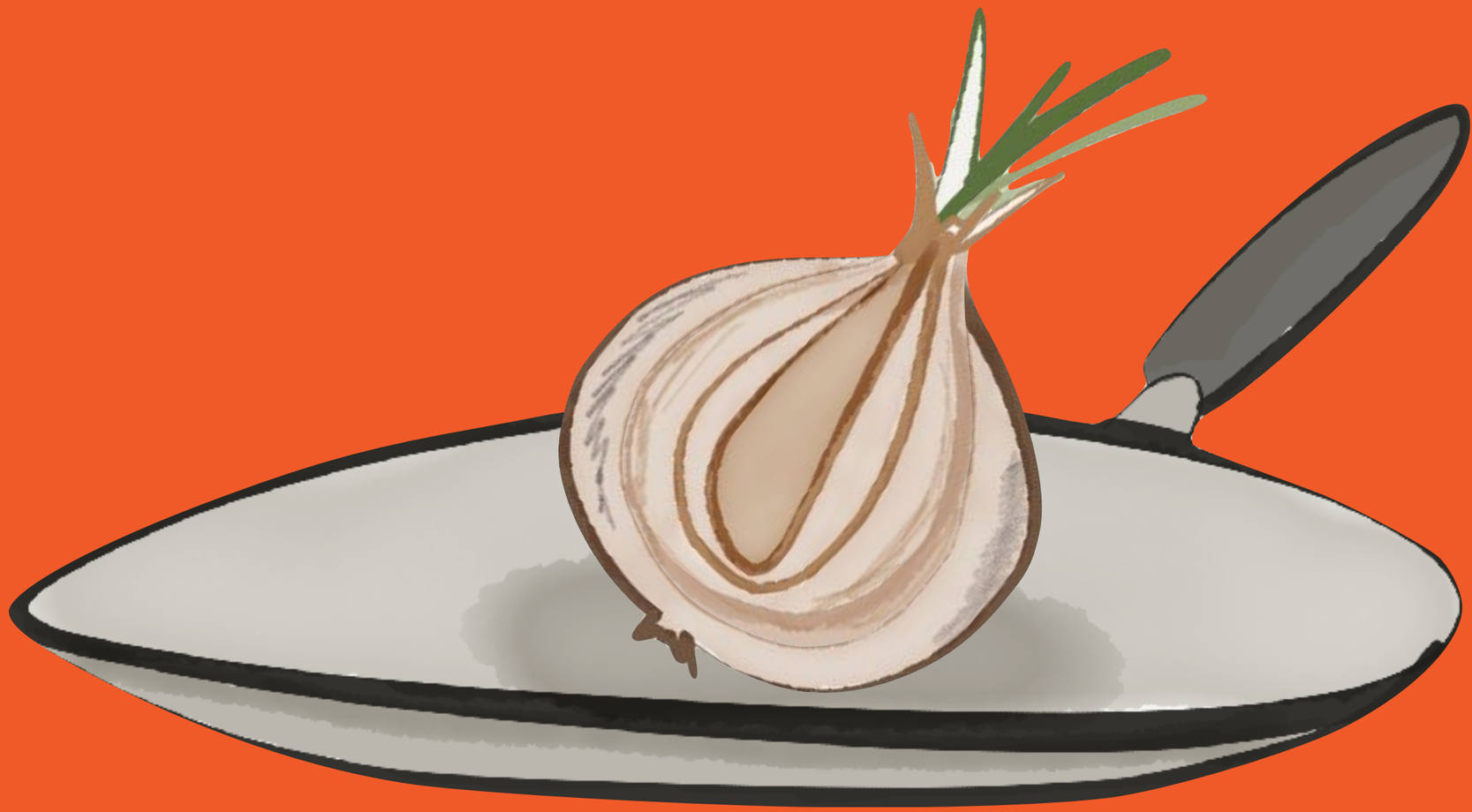
Only developer *X*
can do *Y* on branch *Z*

Don't push to a red build

Squash a feature to a single commit before you merge



Now you’re thinking in Git.



Thank you!