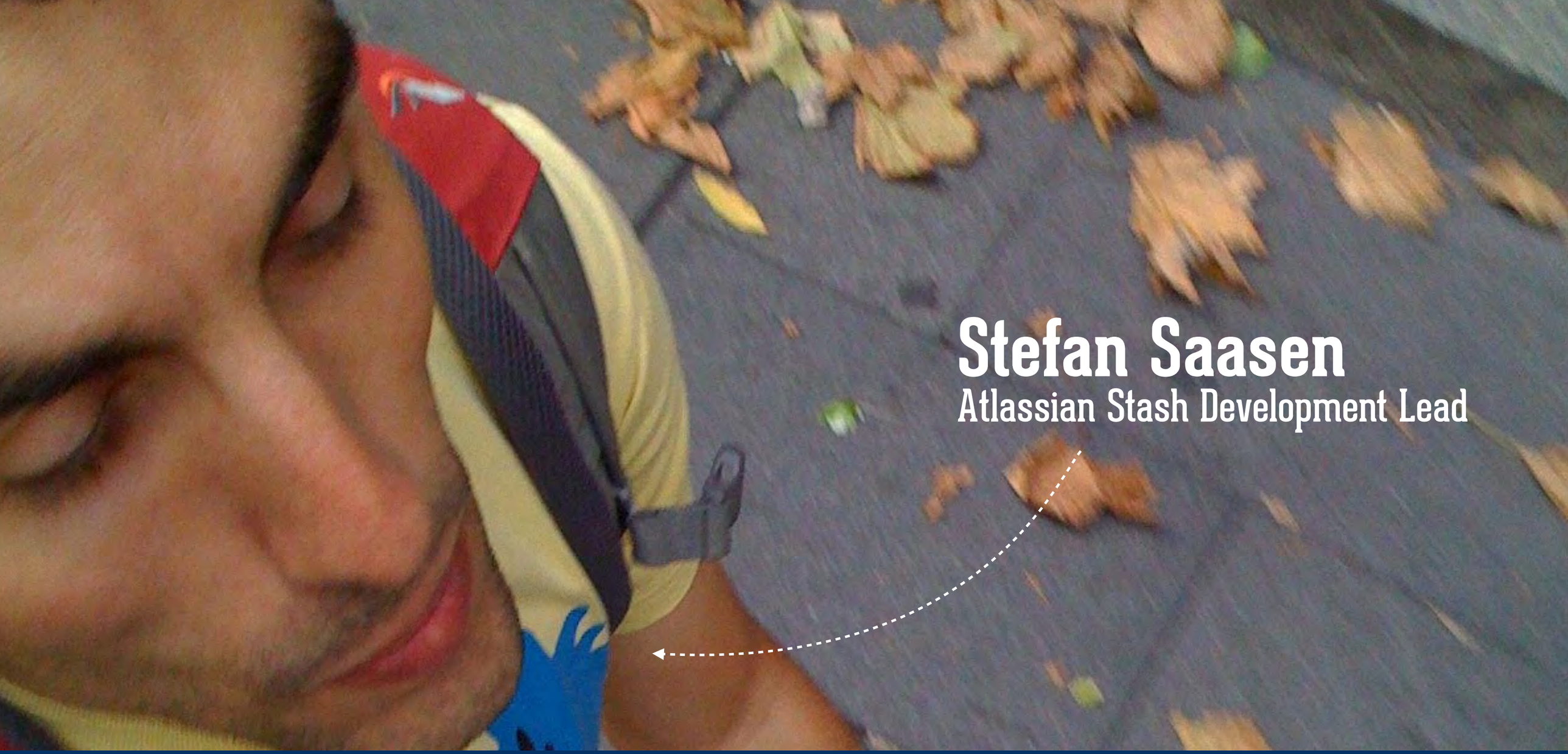
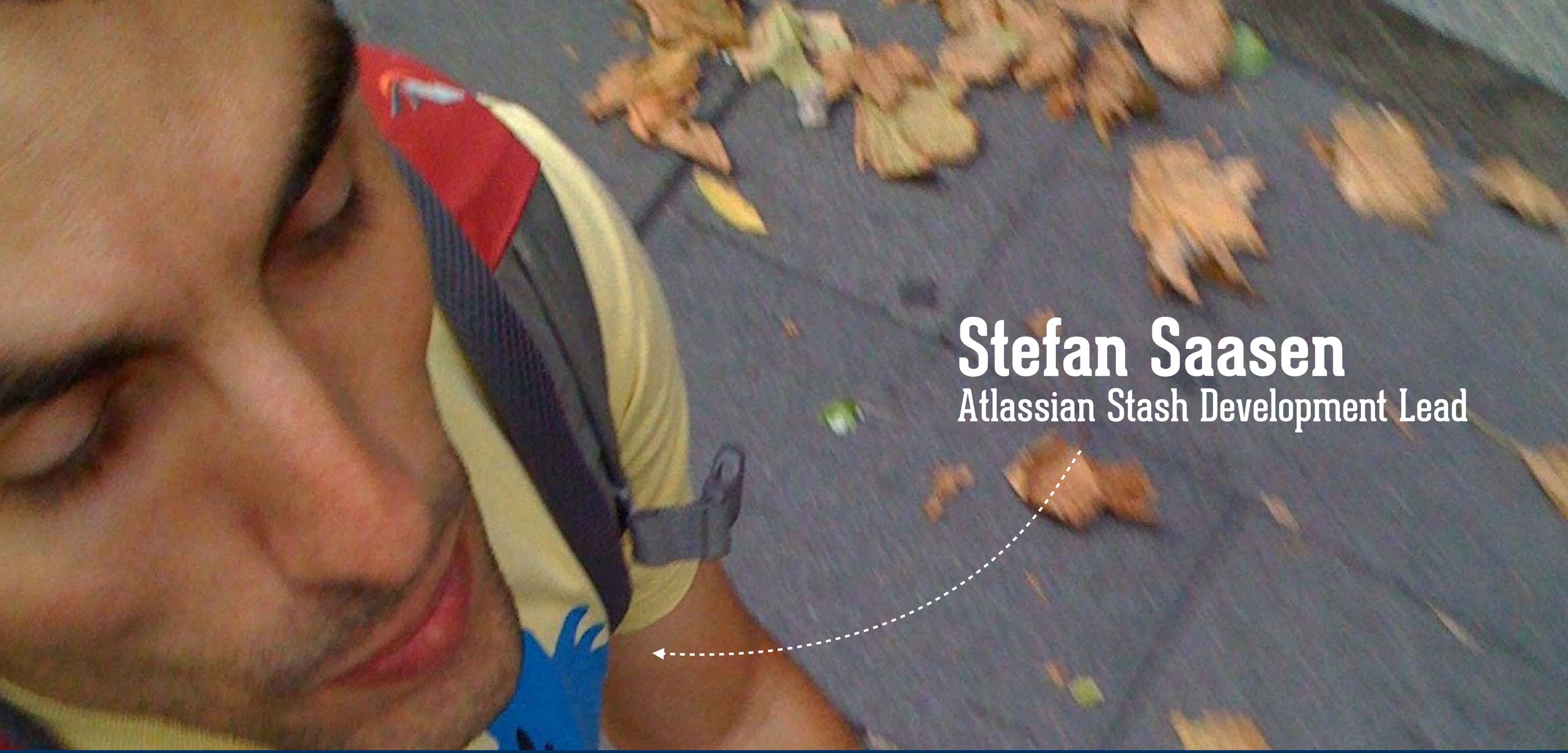
Agile AUSTRALIA 14

17 - 18 JUNE 2014 | MELBOURNE CONVENTION & EXHIBITION CENTRE

Real world Git workflows


Stefan Saasen
Atlassian Stash Development Lead

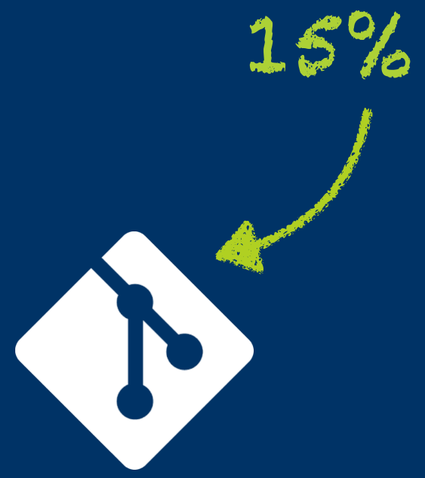


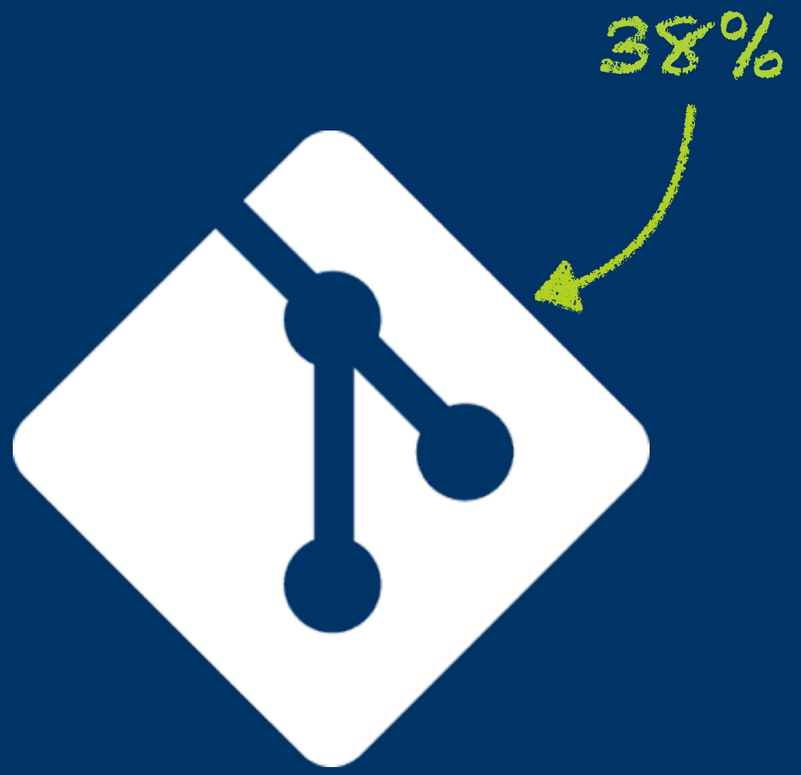
Stefan Saasen

Atlassian Stash Development Lead

@stefansaasen

You heard git is on the rise





from 2011 to 2013

Source: RedMonk Git Survey 2013

You heard git has

Cheap local **branching**

Full local history

Much faster than **svn**

Staging area

Huge **community**

Speed

Superior **Merging**

prominent in **Open Source**

Distributed

cryptographic integrity

We'll cover:

We'll cover:

1. **Collaboration** model

We'll cover:

1. **Collaboration** model
2. **Branching** model

We'll cover:

1. **Collaboration** model
2. **Branching** model
3. **Practices**

1 Which **collaboration** model?

REPOSITORIES &
FILE SHARING
ARE NOT
COLLABORATION.

@kevinmhoffman

ANARCHY

Fully decentralized
Anarchy

I do my thing

I do my thing, too

ANARCHY

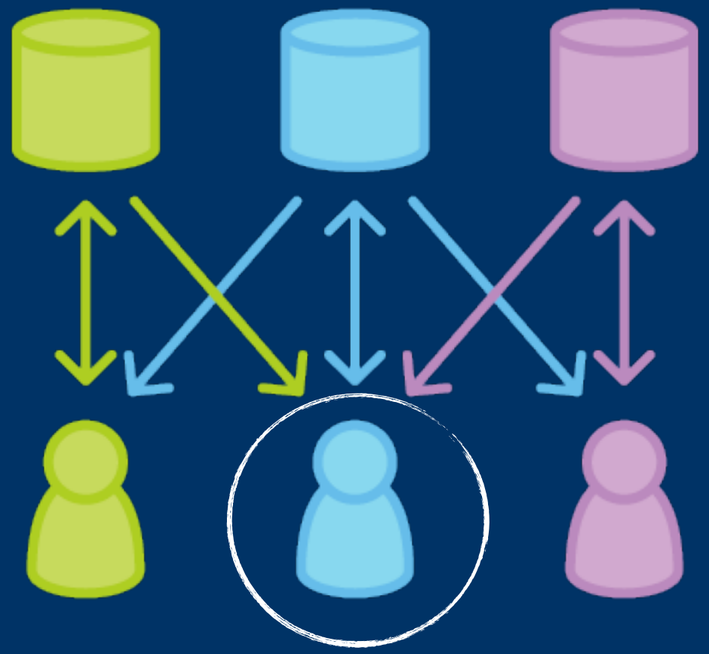
here's mine, who tells john?

look ma, a goat!

Fully decentralized Anarchy

Gatekeeper

Blessed repository with Gatekeeper

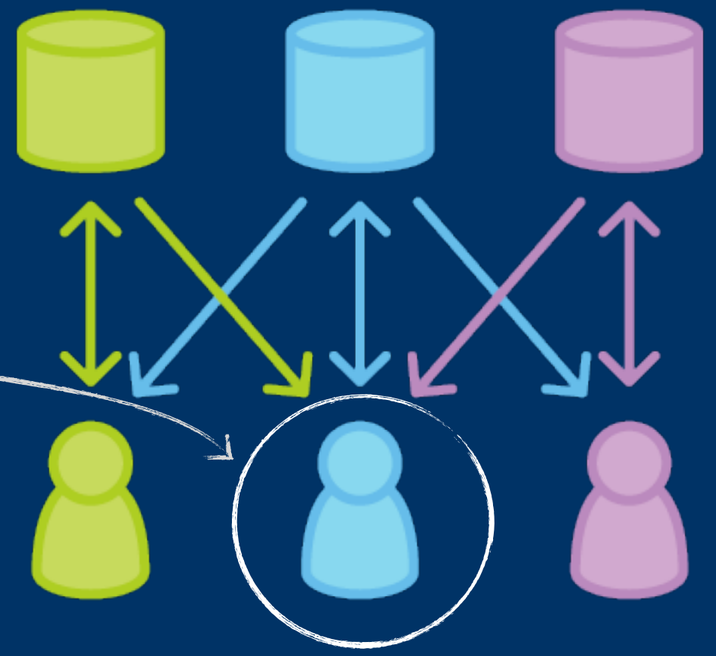


He is cool

Gatekeeper

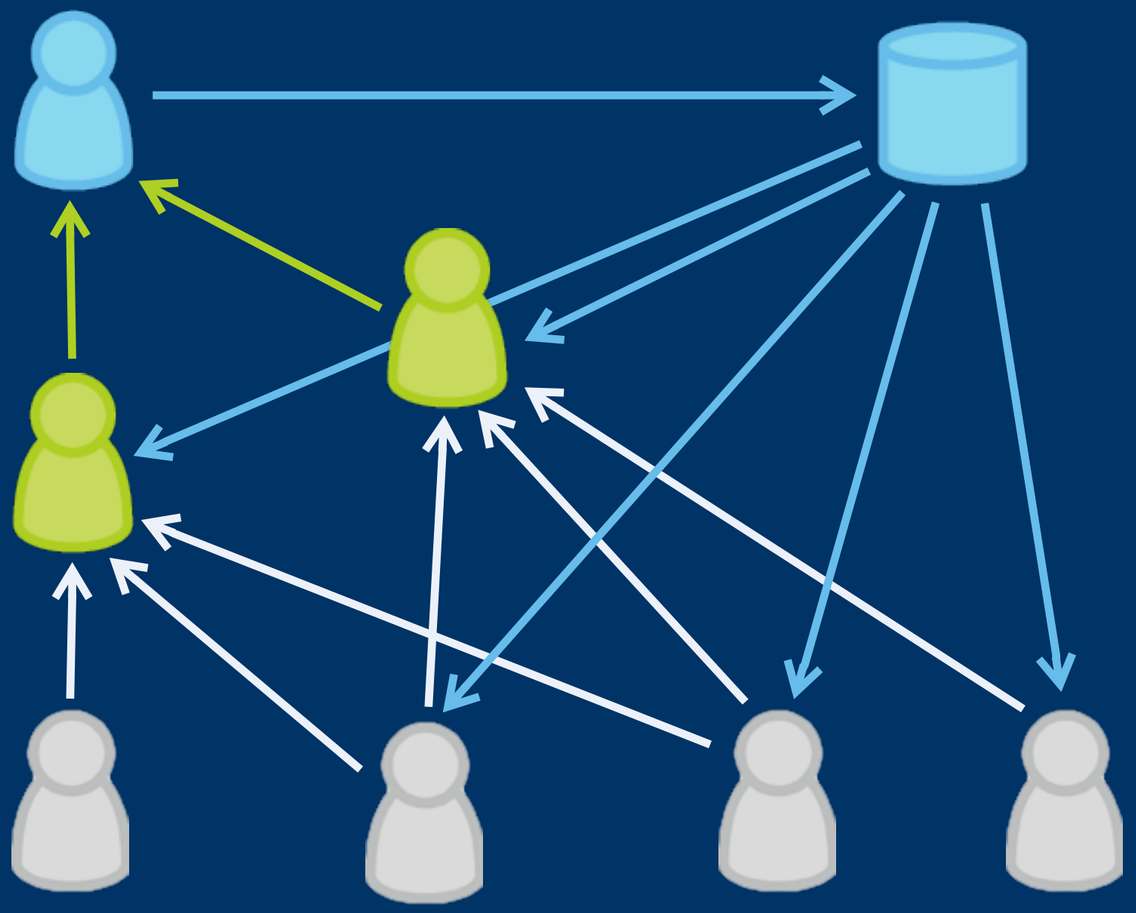
Blessed repository with Gatekeeper

To have your work accepted, talk to him



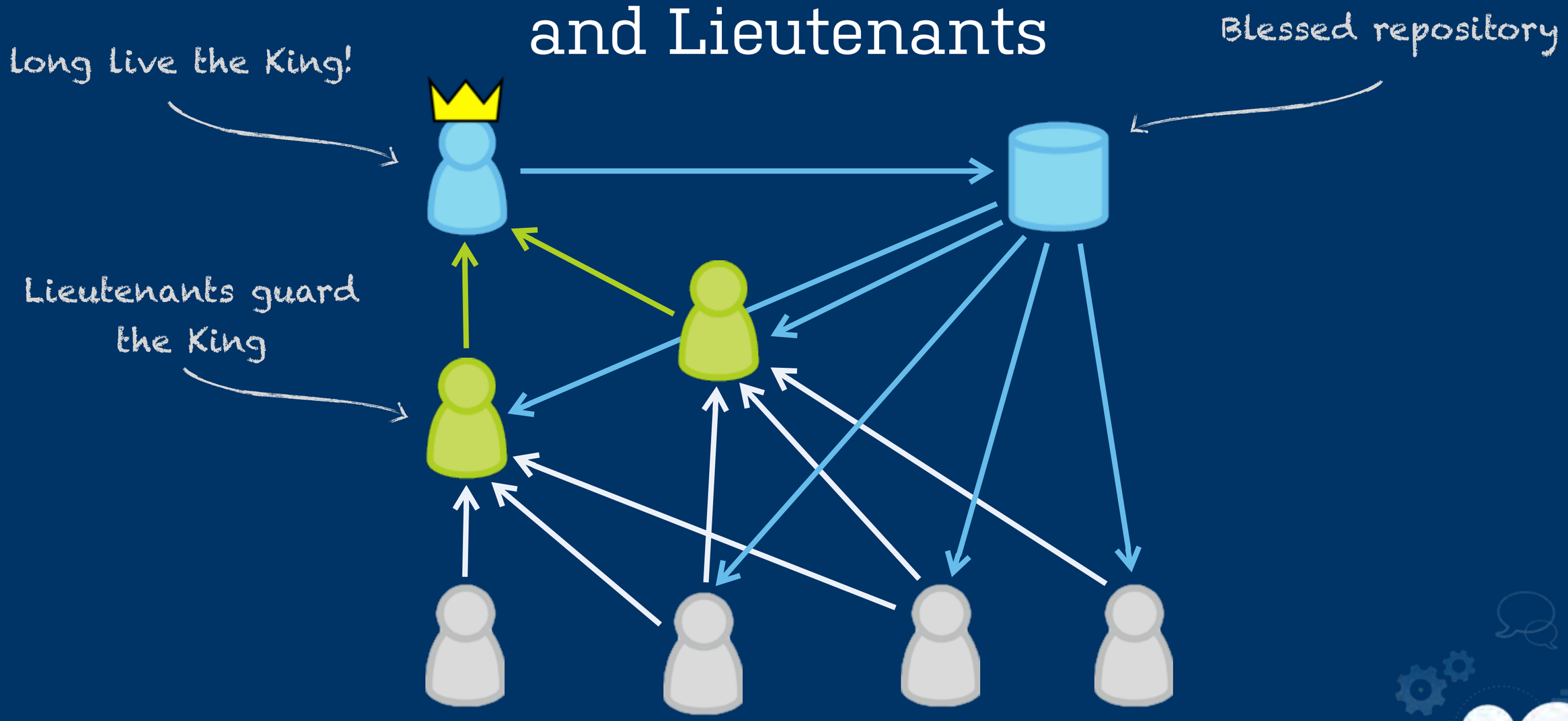
Dictator

and Lieutenants



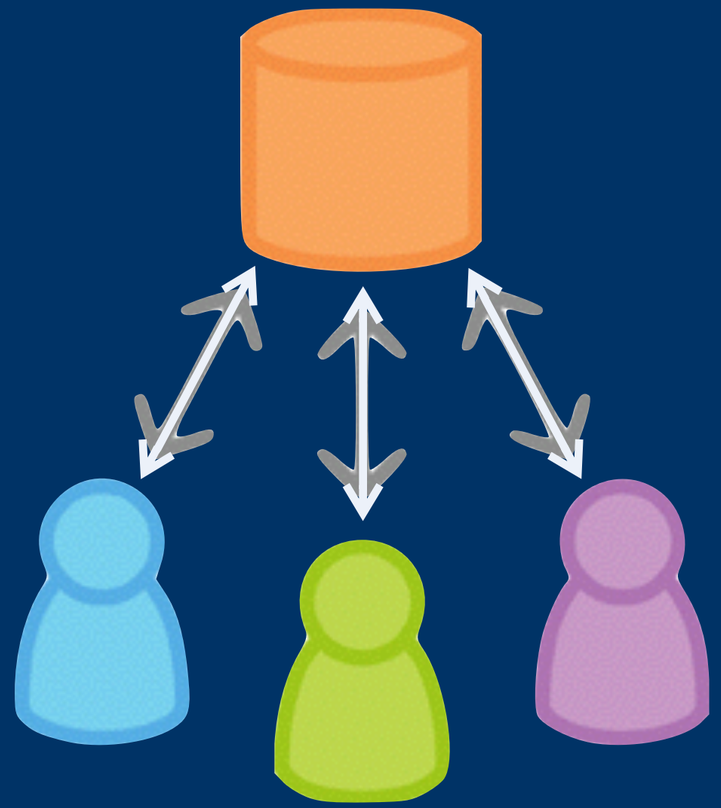
Dictator

and Lieutenants



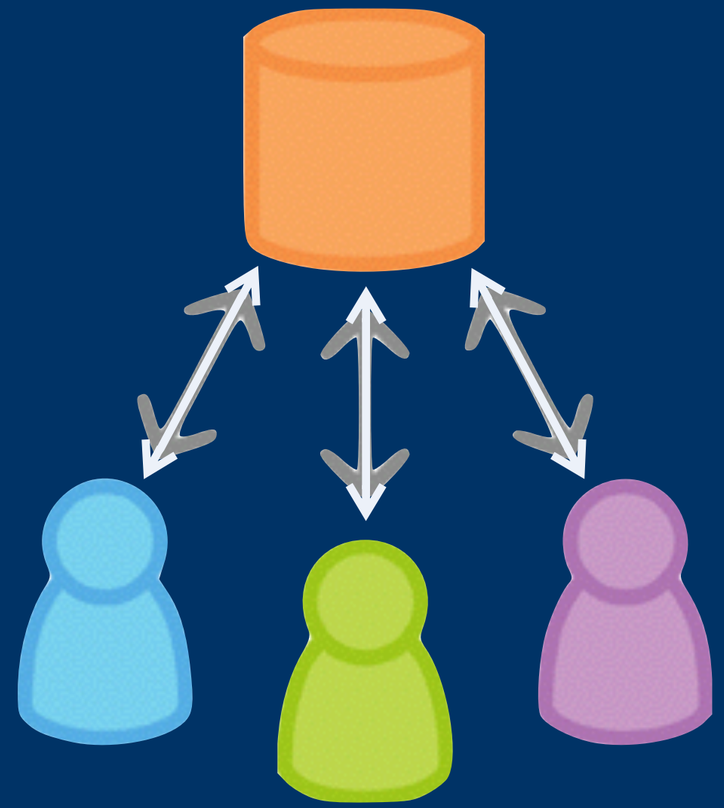
Centralised

Shared common repository



Centralised

Shared common repository



we share "everything"

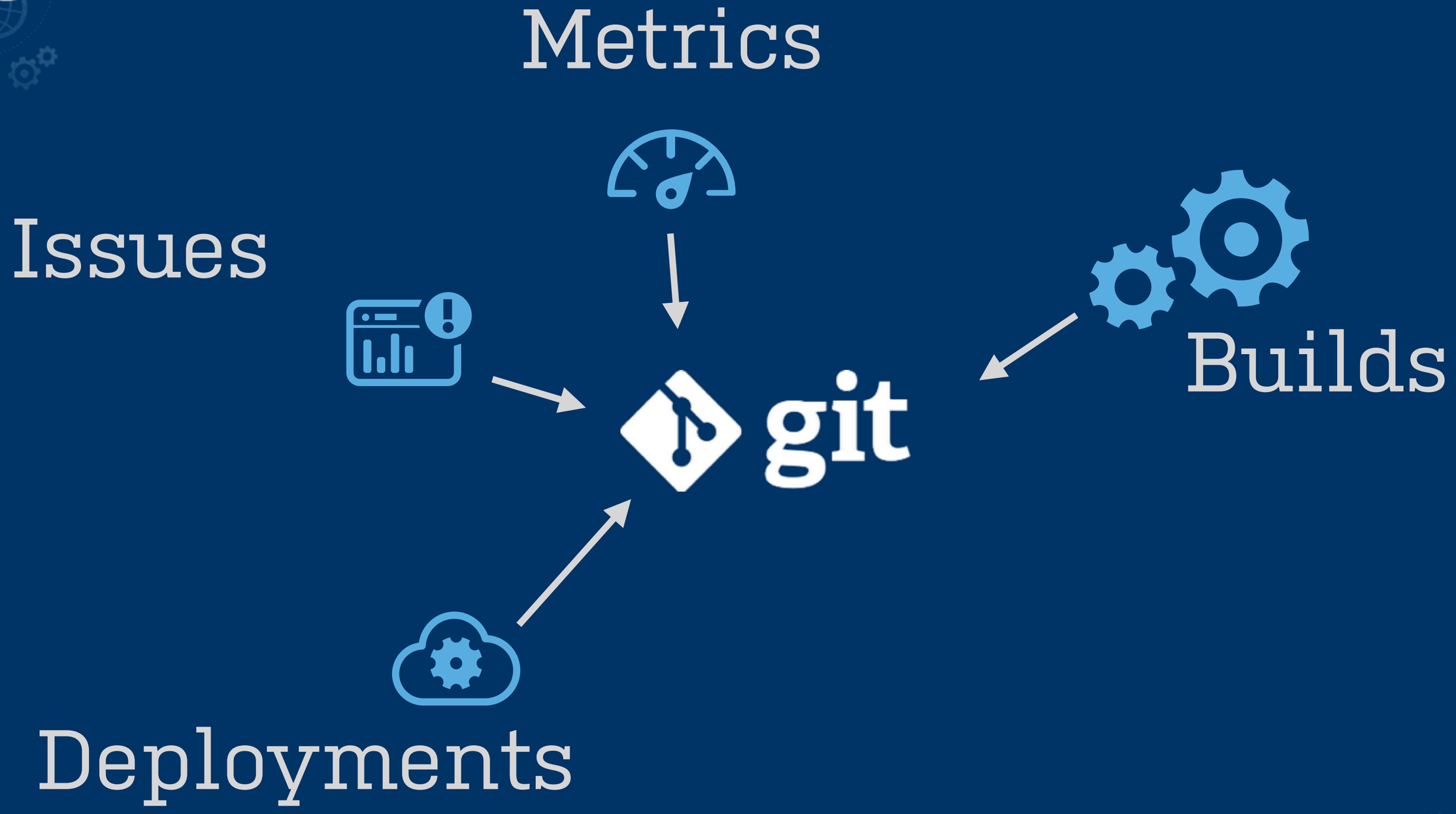
+ git

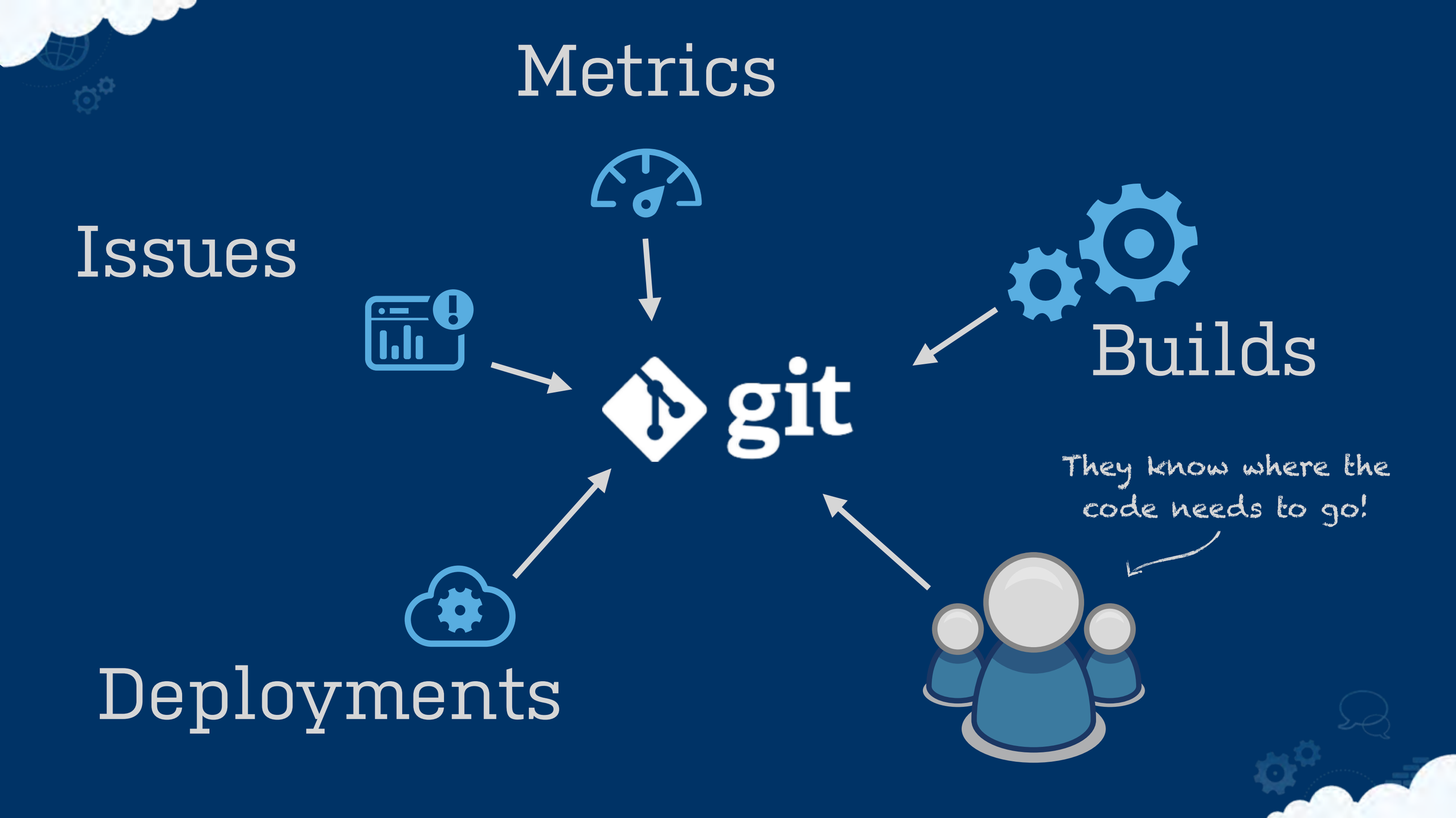
Enterprise

= Centralized

Enterprise

git


Metrics
Issues
Builds
git
Deployments


Metrics
Issues
Builds
git
They know where the code needs to go!
Deployments

Which **branching** model?

2 Which **branching** model?

Can we still fix a bug for the upcoming **Release** *?*

Can we still fix a bug for the upcoming **Release** *?*

Is the code for that **Feature** *complete?*

Can we still fix a bug for the upcoming **Release** *?*

How do we do a **Hotfix** *for the current version?*

Is the code for that **Feature** *complete?*

Can we still fix a bug for the upcoming **Release** *?*

How do we do a **Hotfix** *for the current version?*

Is the code for that **Feature** *complete?*

Has everyone **Reviewed** *the code for this feature ?*

What's the best Git workflow?

We dont know!

What's the best Git workflow?

different cultures

What's the best
Git workflow?

different cultures
+ different products

What's the best
Git workflow?

different cultures
+ different products
+ different teams

What's the best
Git workflow?

different cultures
+ different products
+ different teams

= different workflows

Design your own Workflows

1. **Single** branch workflow - **aka** trunk

Committing locally

Local Repository

Committing locally

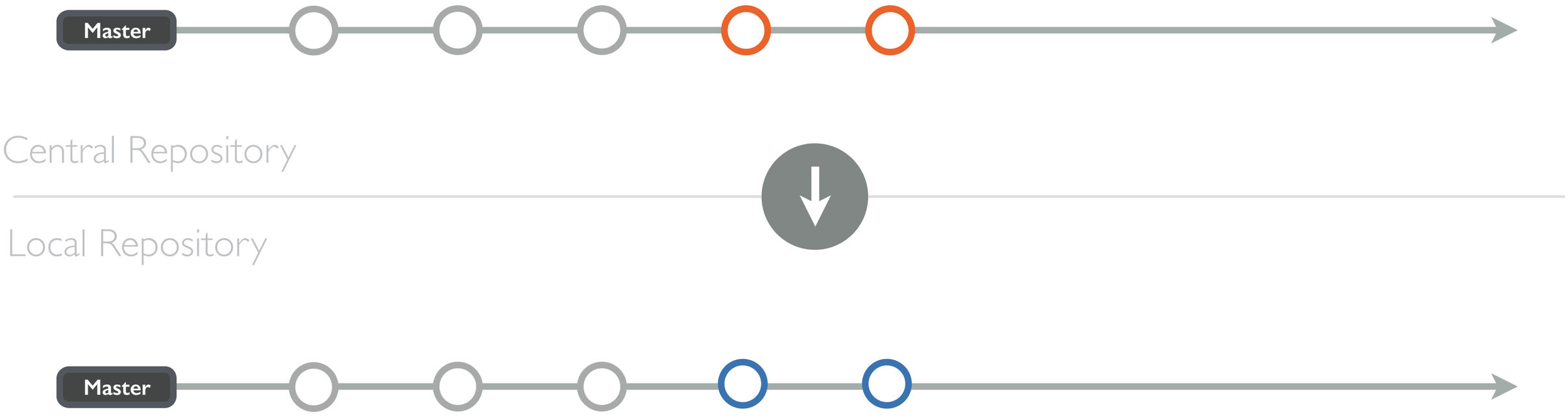
Local Repository

Central repository **has updates**

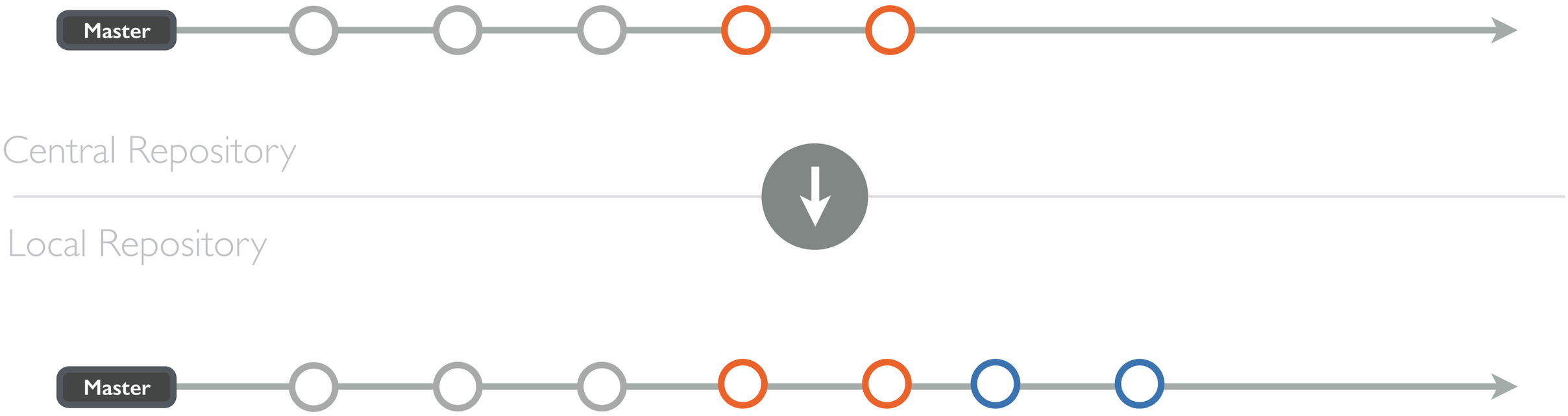
Central Repository

Local Repository

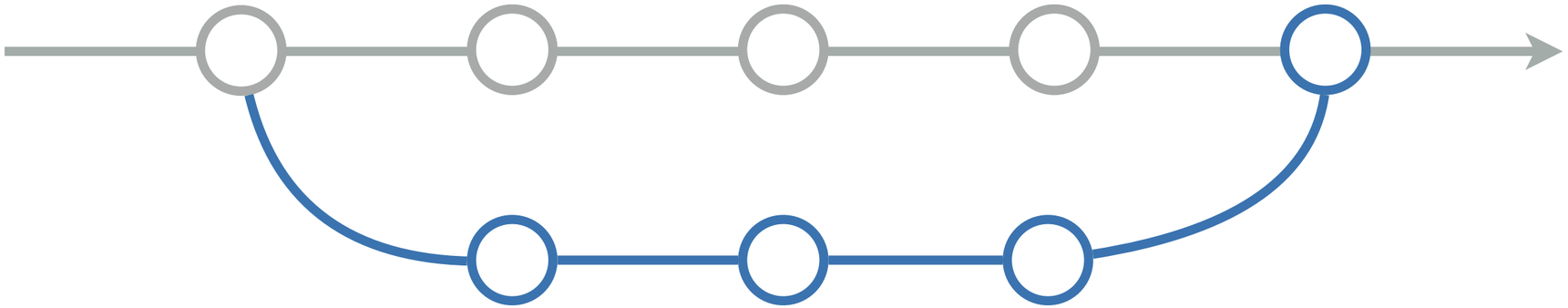
Getting updates via rebase



Getting updates via rebase

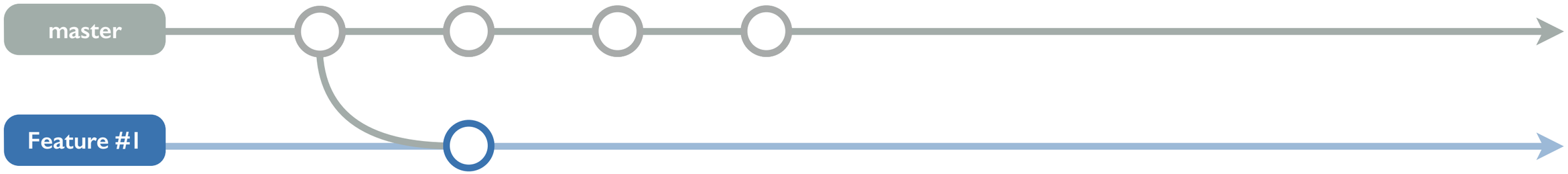


2. **Feature** branching workflow

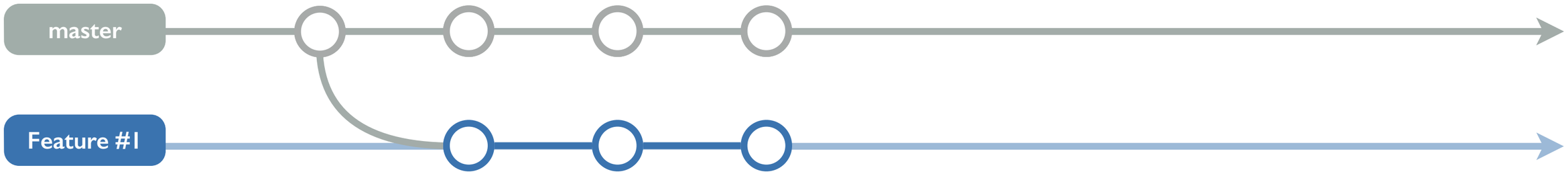


Creating a branch

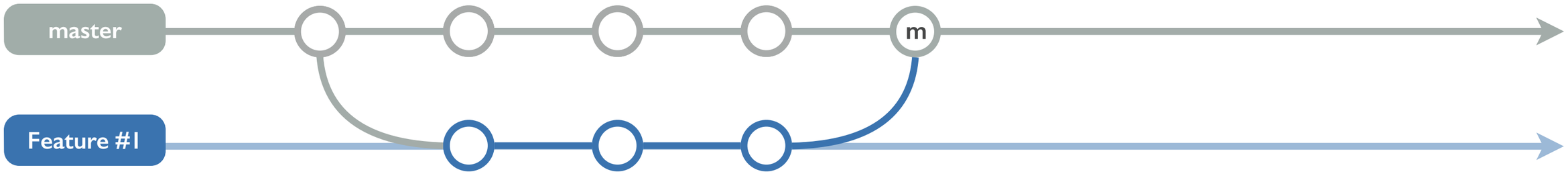
Creating a branch



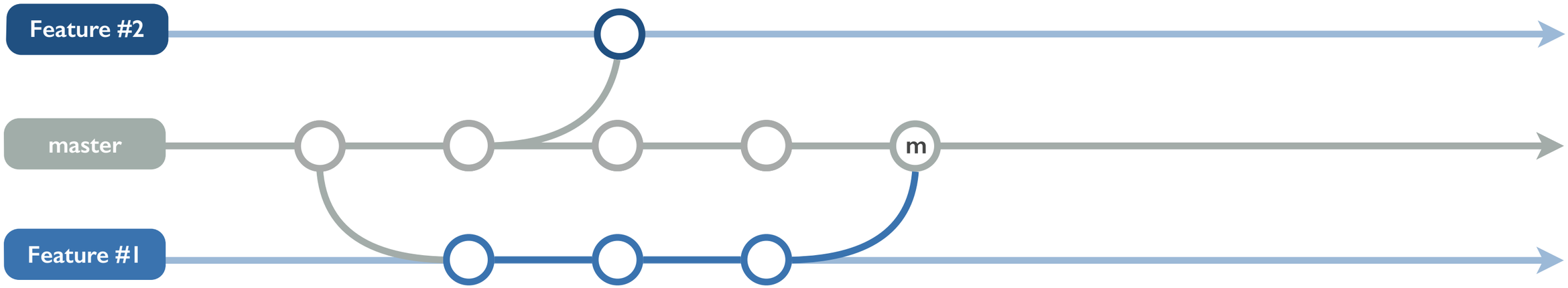
Adding changes



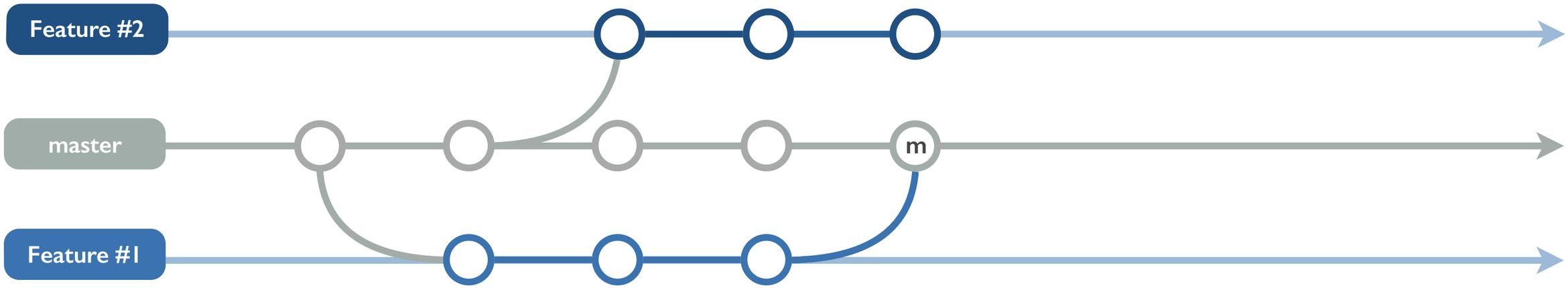
Merge



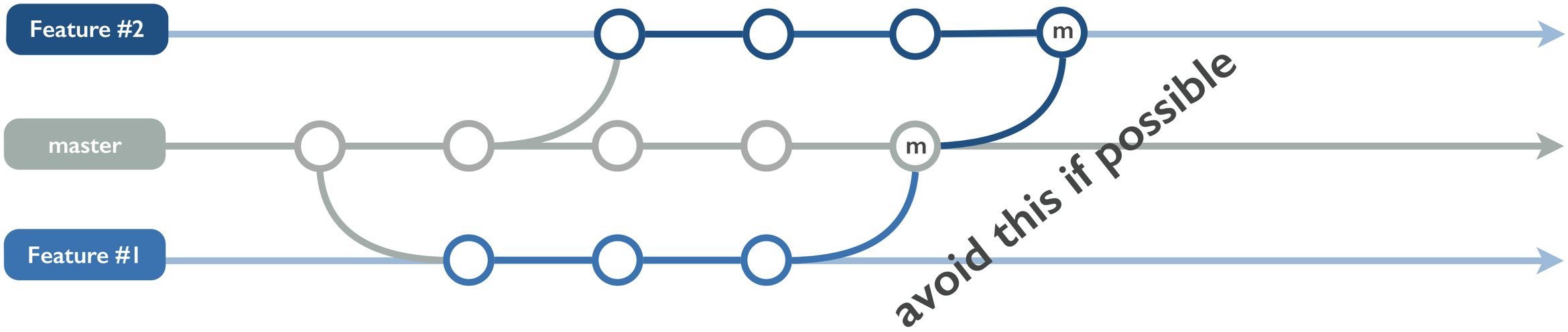
Working on features in parallel



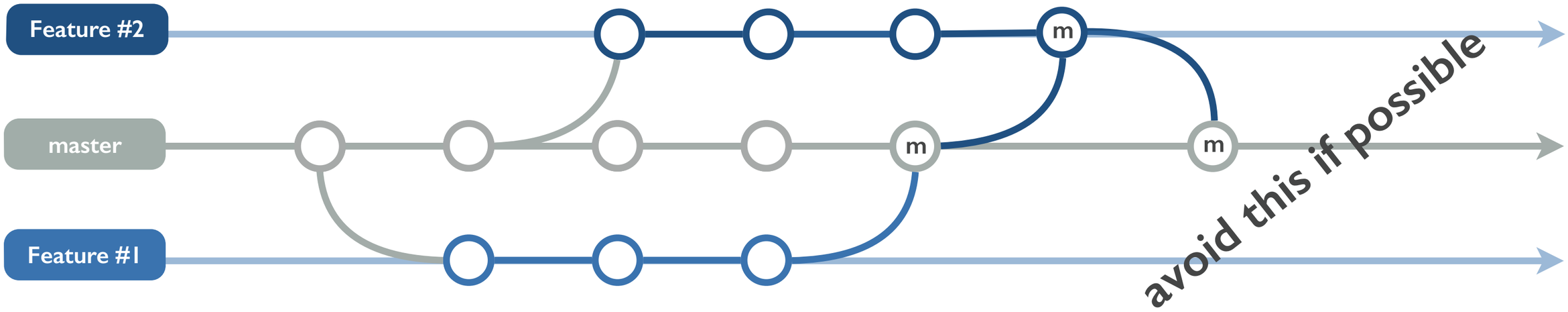
Working on features in parallel



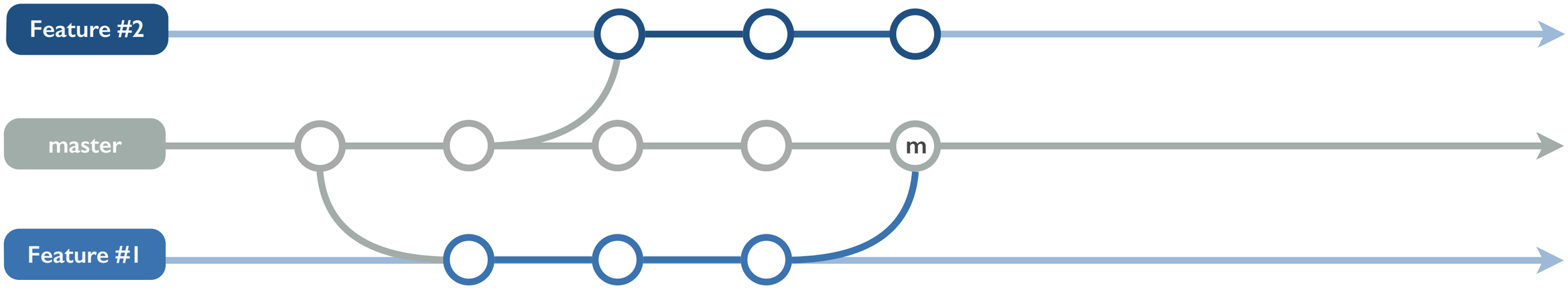
Suboptimal way



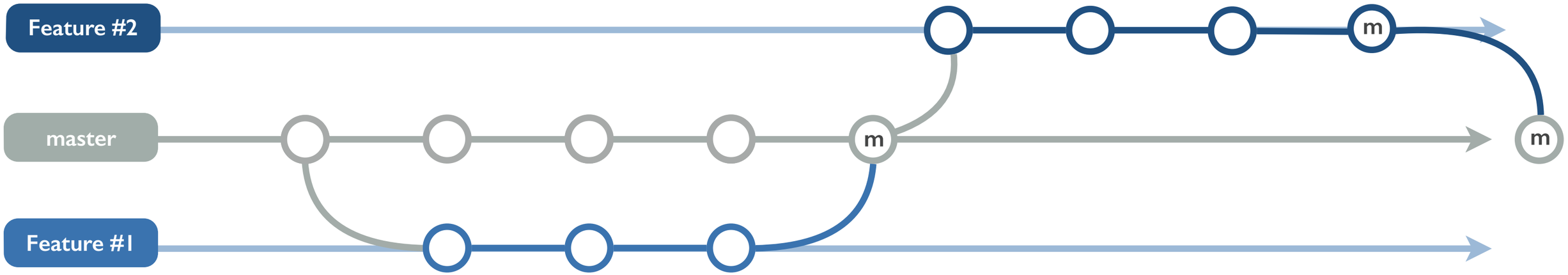
Suboptimal way



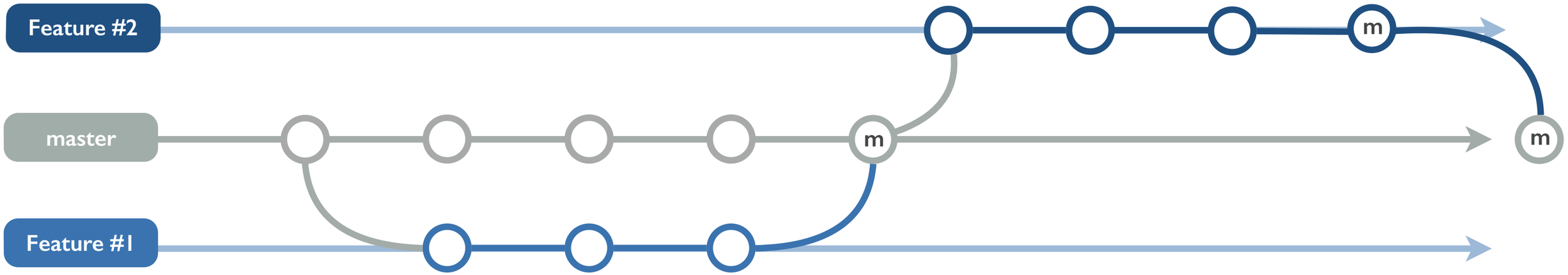
A better way using rebase



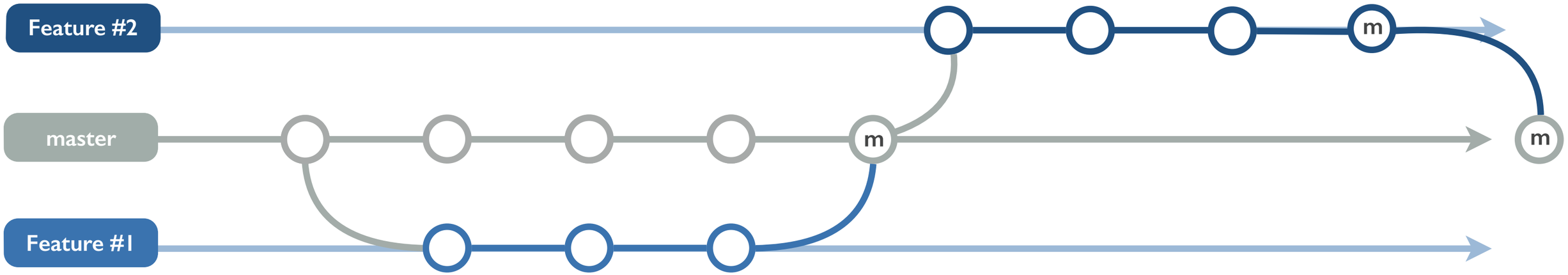
Working on features in parallel



Working on features in parallel

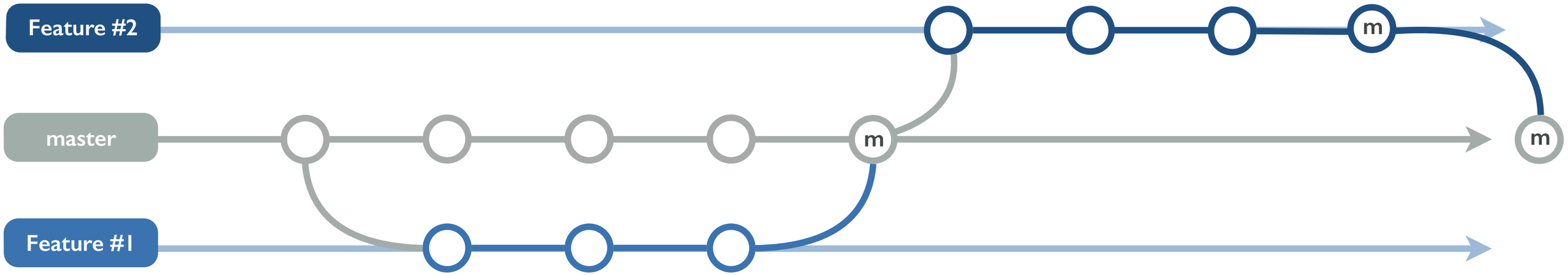


Working on features in parallel



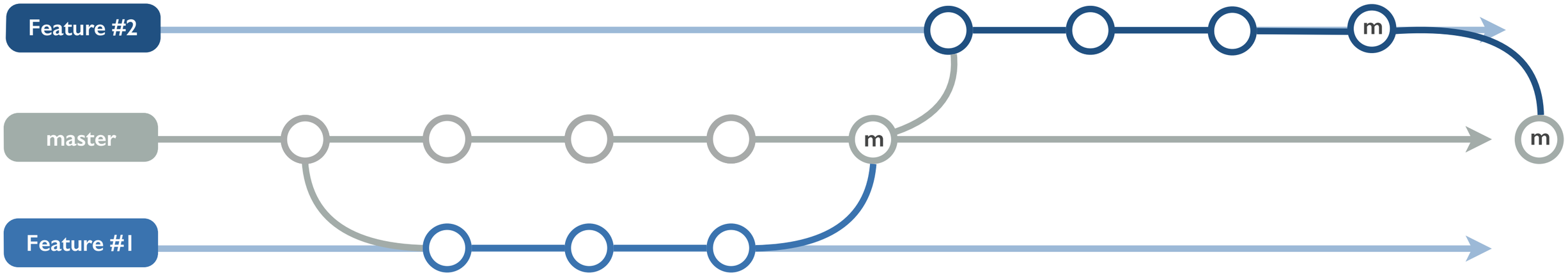
- Always with local branches

Working on features in parallel



- Always with local branches
- With shared branches: After review, before merging

Working on features in parallel



- Always with local branches
- With shared branches: After review, before merging
- Mostly a non-issue for short lived branches when updates from master are not required

3. Continuous delivery workflow

Atlassian Marketplace

1 **master** is in production

promoted from staging, can receive hot-fixes

1 **master** is in production

promoted from staging, can receive hot-fixes

1. **master** is in production
2. **staging** is the next version

promoted from staging, can receive hot-fixes

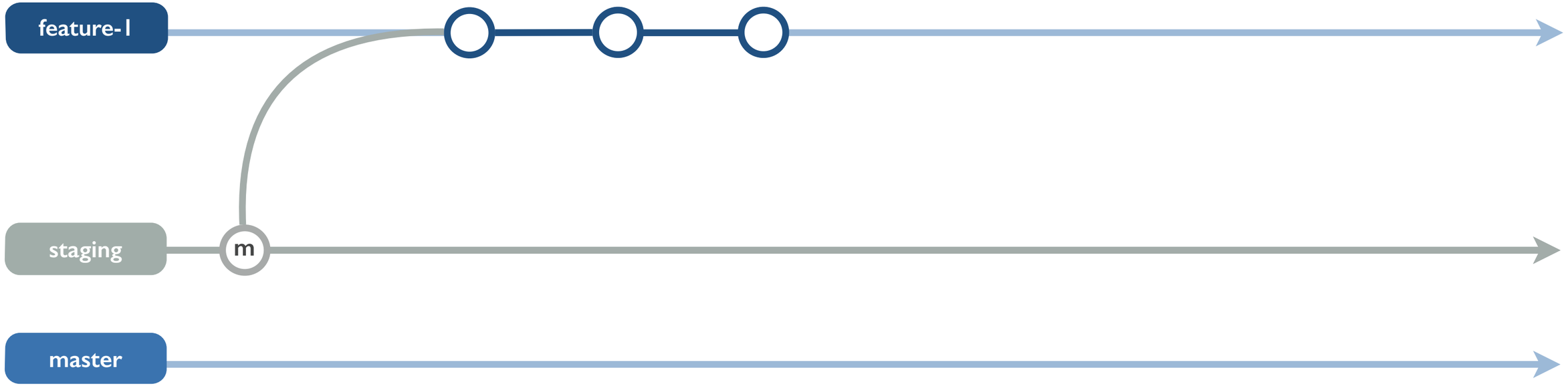
1. **master** is in production
2. **staging** is the next version
3. new features off **staging**

promoted from staging, can receive hot-fixes

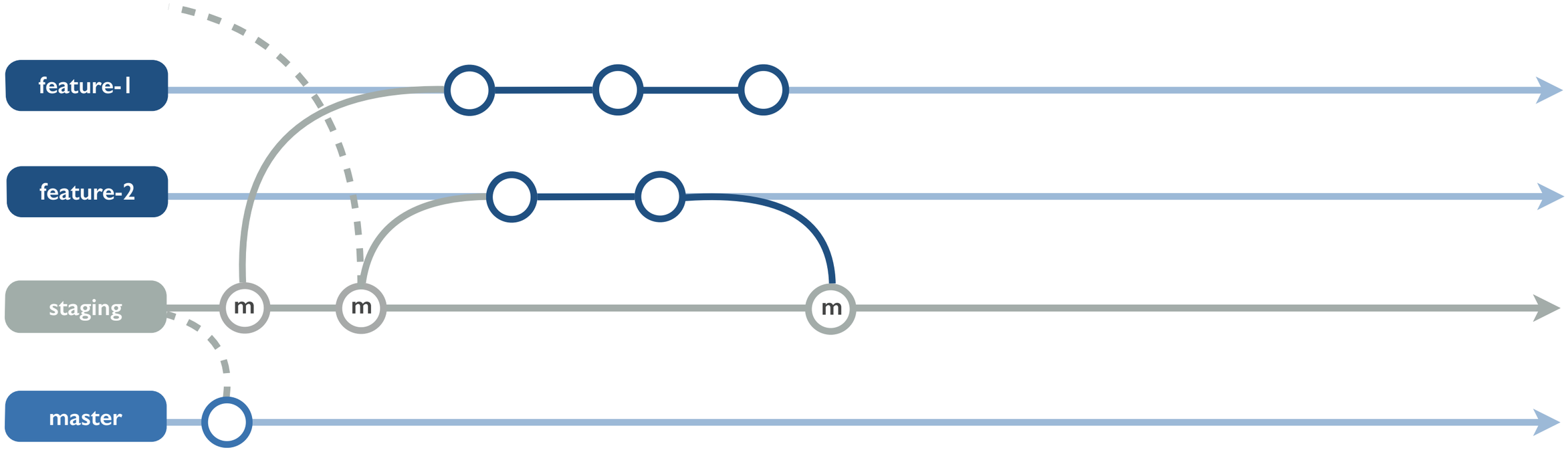
1. **master** is in production
2. **staging** is the next version
3. new features off **staging**

with branch names like: username/ISSUE-KEY-summary

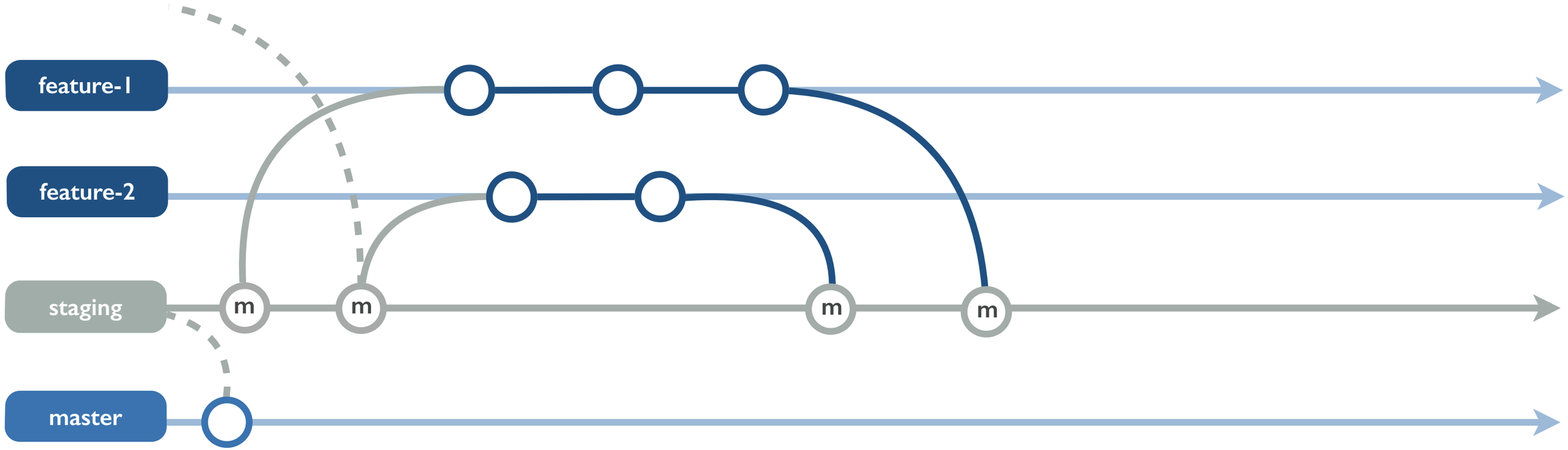
3. Continuous delivery workflow



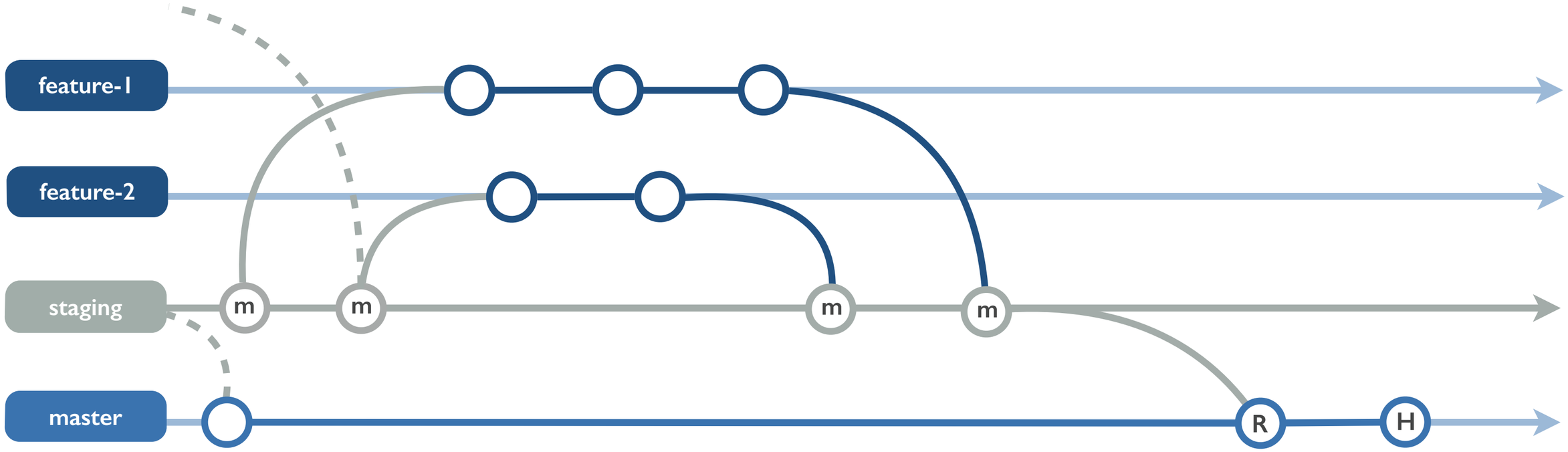
3. Continuous delivery workflow



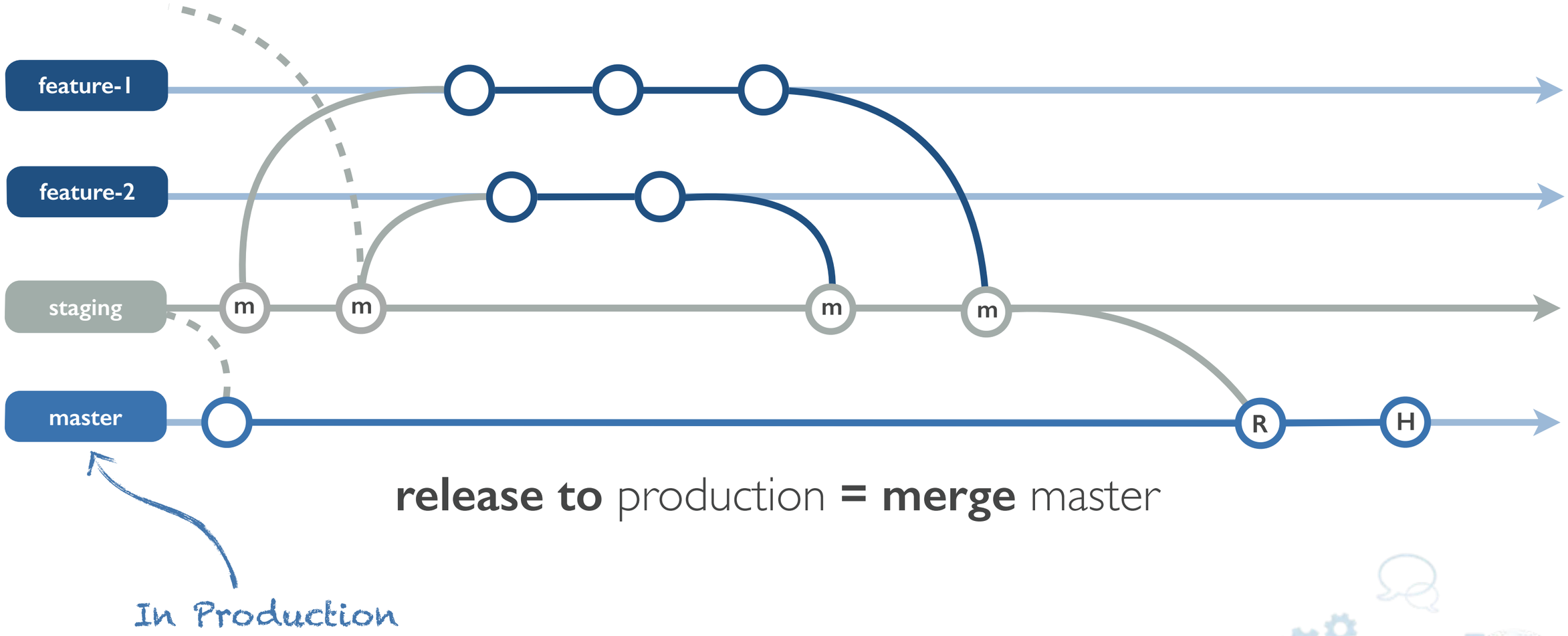
3. Continuous delivery workflow



3. Continuous delivery workflow

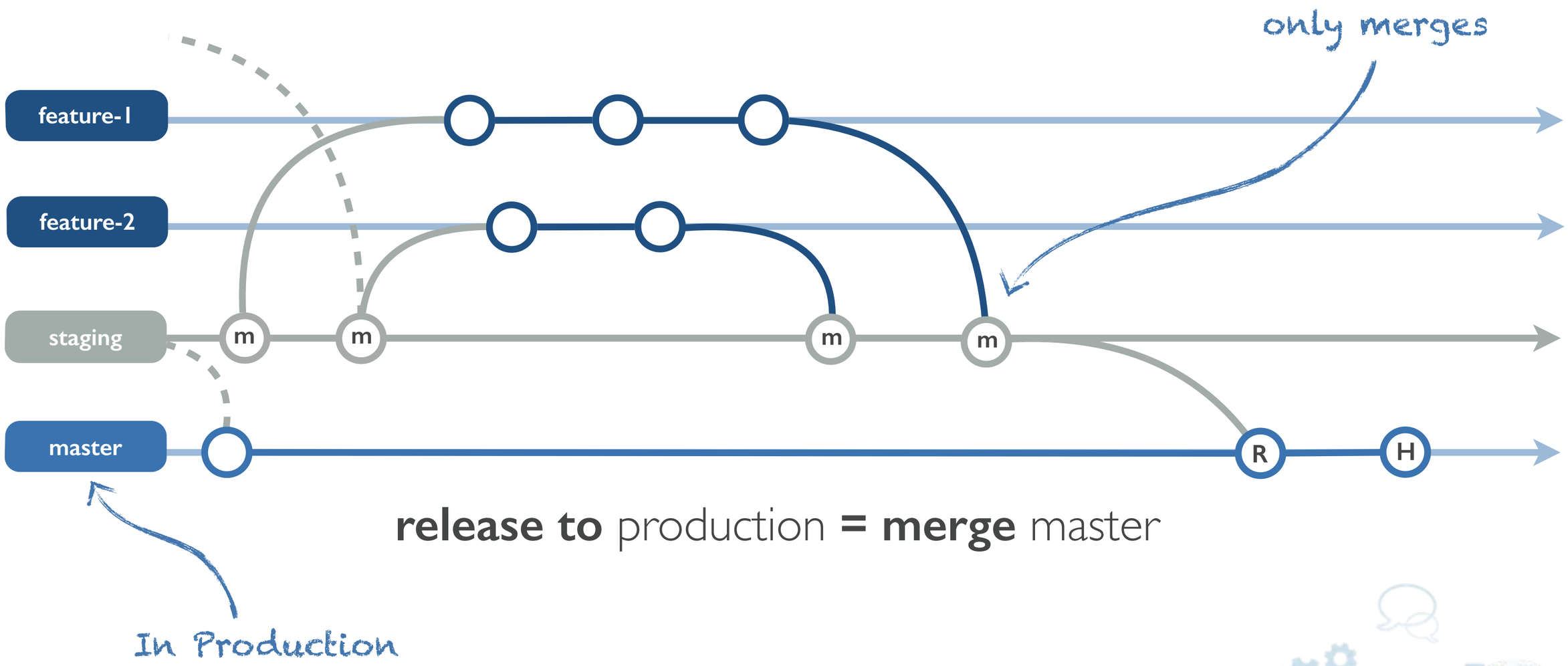


3. **Continuous delivery** workflow



release to production **= merge** master

3. **Continuous delivery** workflow



release to production **= merge** master

4. **Product releases** workflow

Atlassian Stash

1

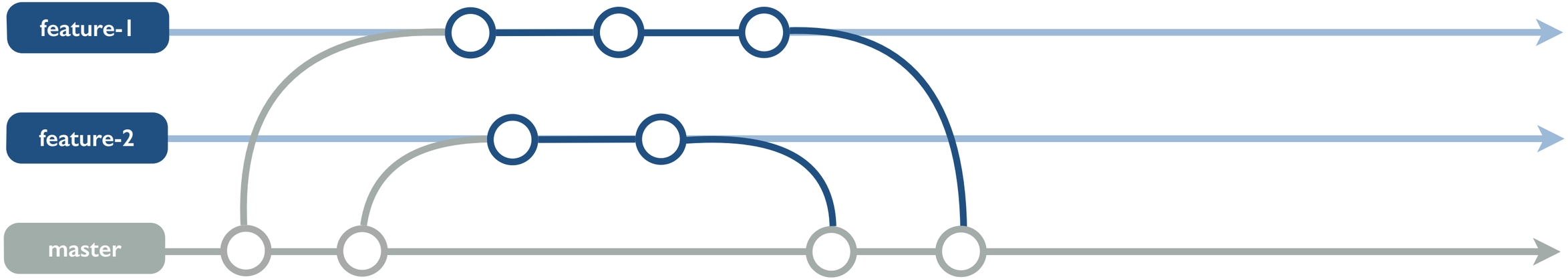
one branch per feature

1. **one branch** per feature
2. **one branch** per bugfix

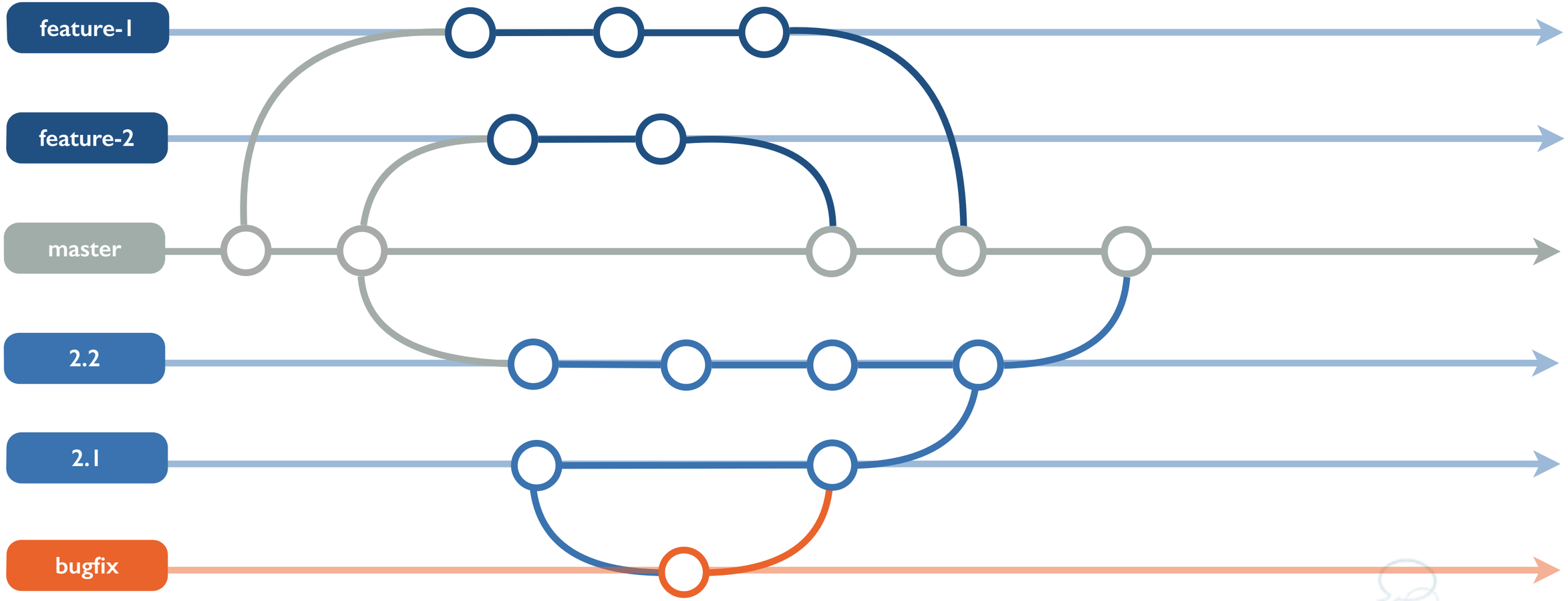
1. **one branch** per feature
2. **one branch** per bugfix
3. long running **stable branches**

1. **one branch** per feature
2. **one branch** per bugfix
3. long running **stable branches**
4. **master** is alpha/RC status

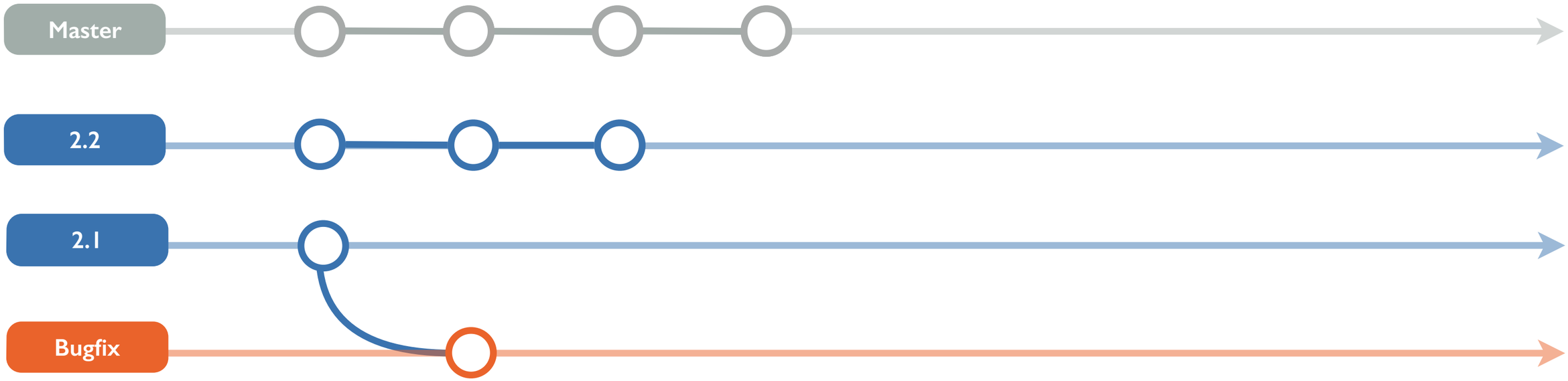
it's just a normal branching workflow…



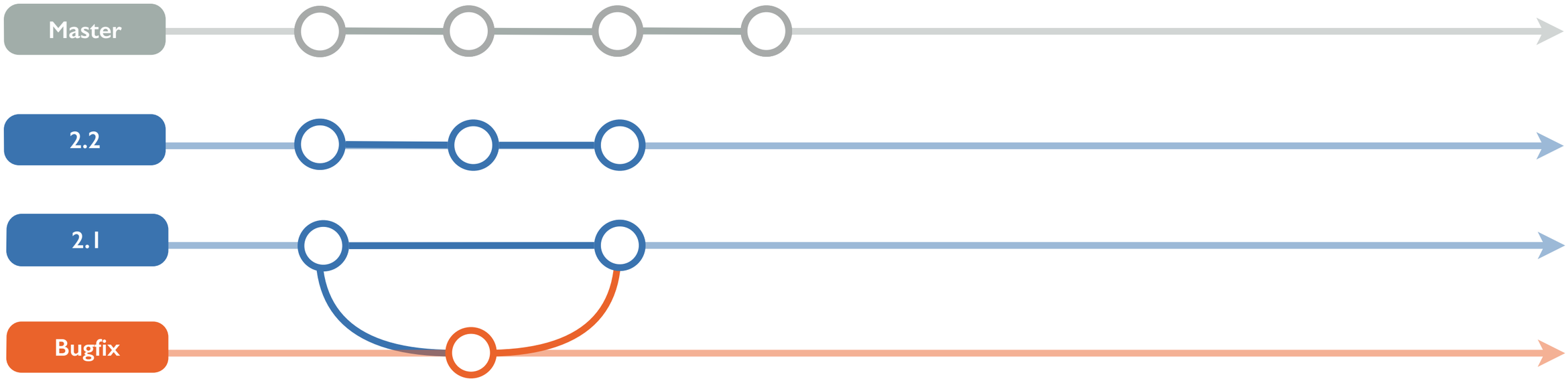
with long running release branches



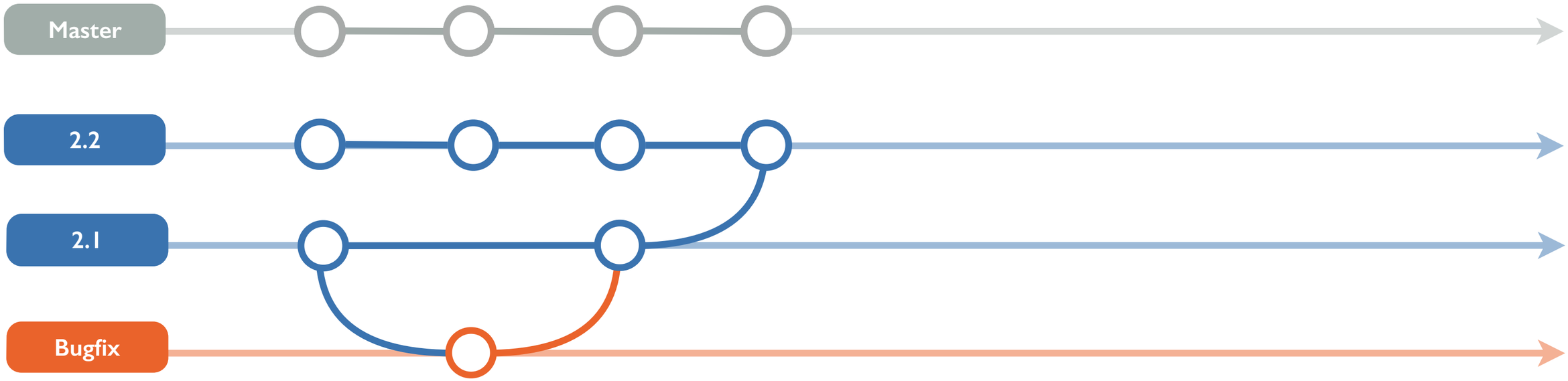
Bugfix Merge



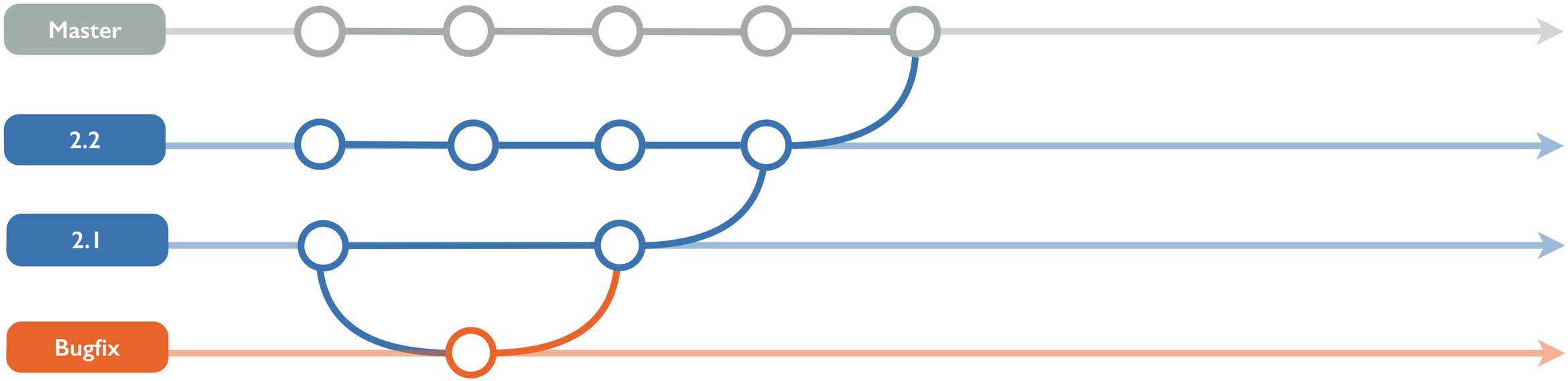
Merge into 2.1



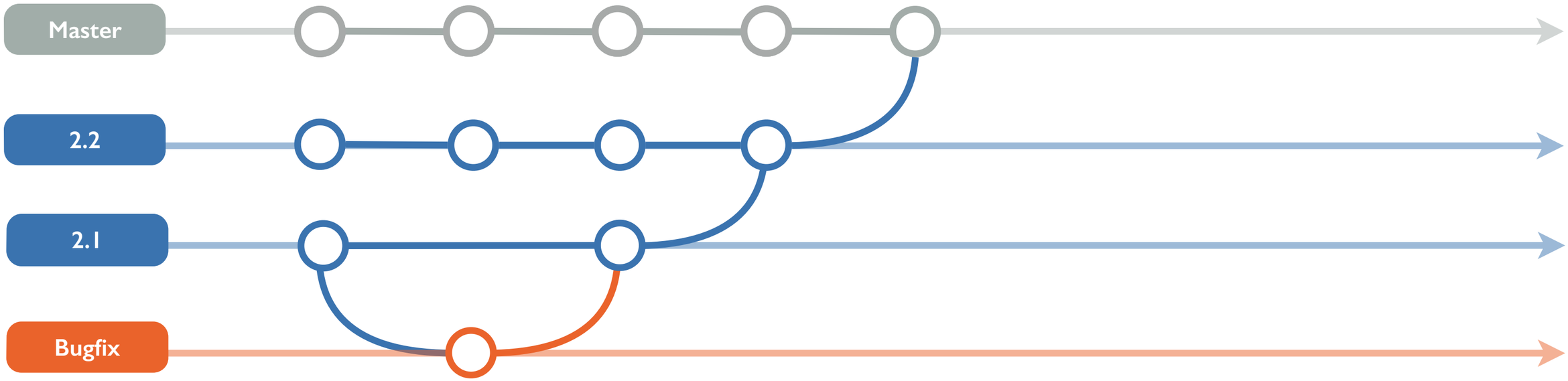
Merge into 2.2



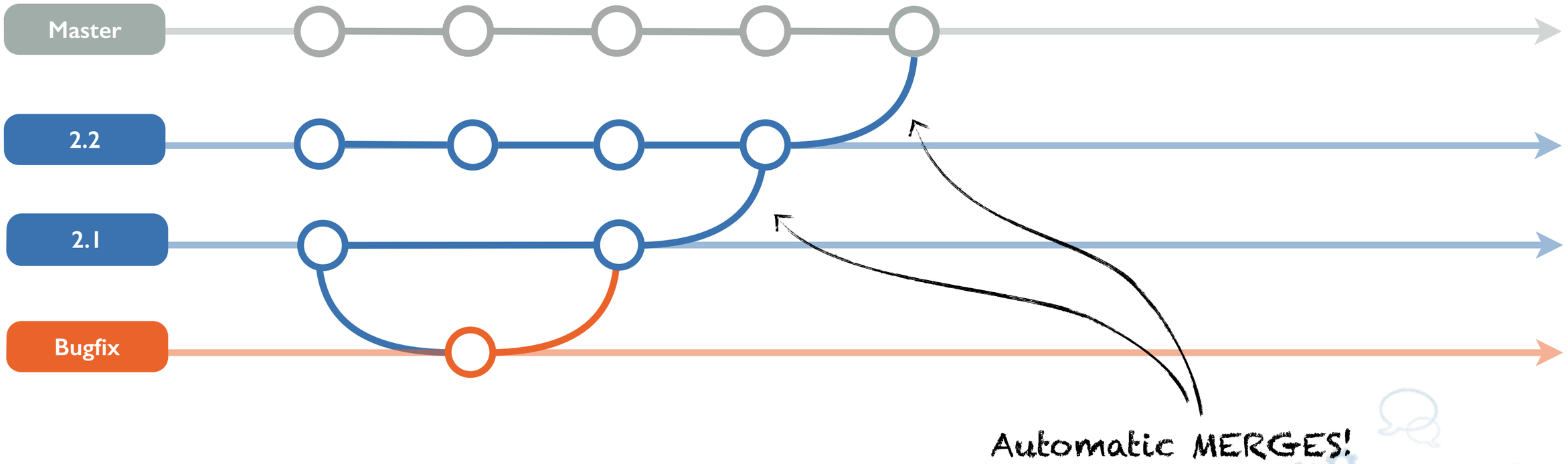
Merge into Master



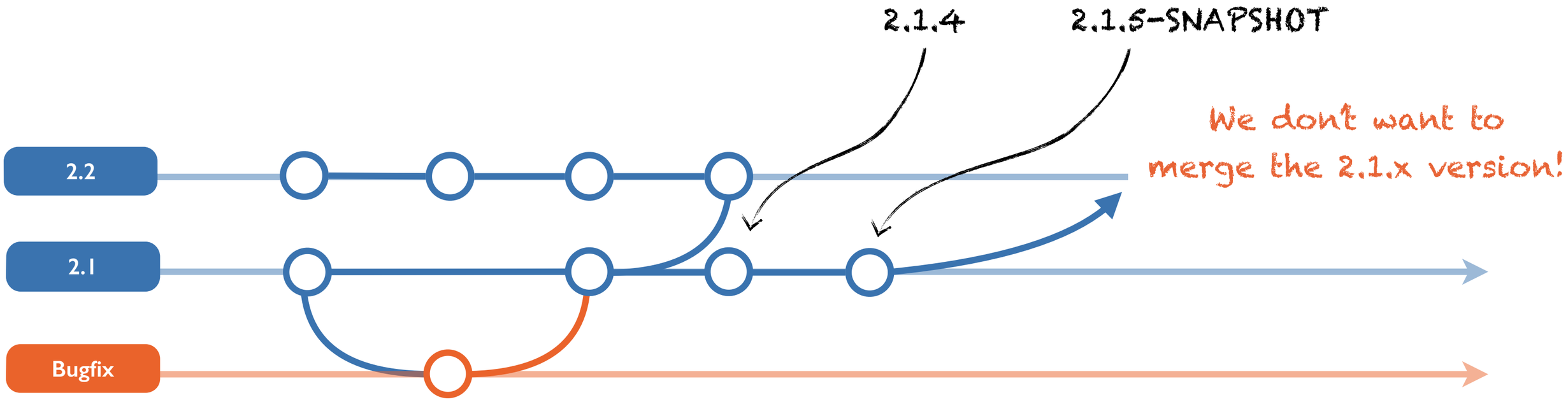
Automatically merging release branches



Automatically merging release **branches**



Automatically merging release branches



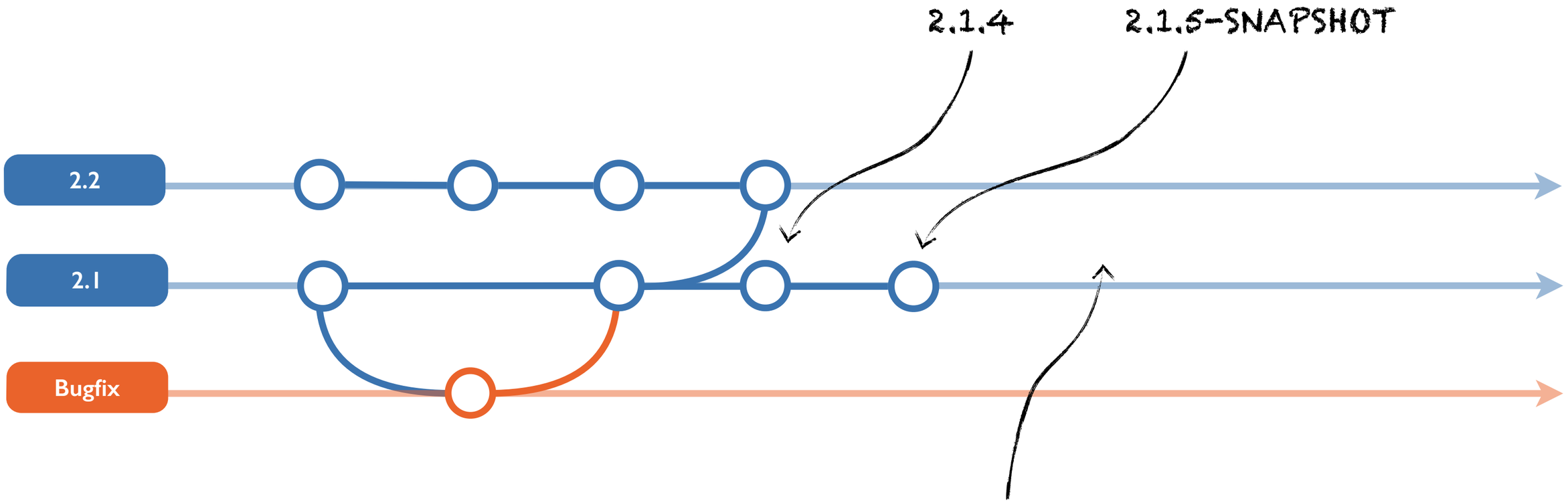
```
git merge --strategy=resolve
```

```
git merge --strategy= recursive
```

```
git merge --strategy=
```

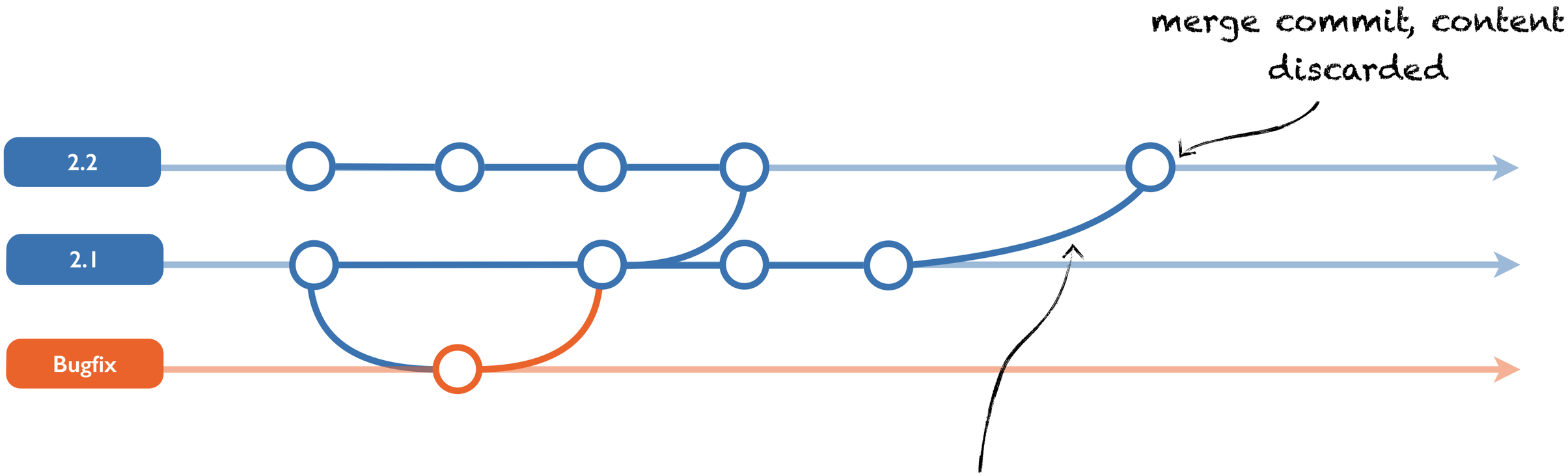
```
git merge --strategy= ours
```

Automatically merging release **branches**



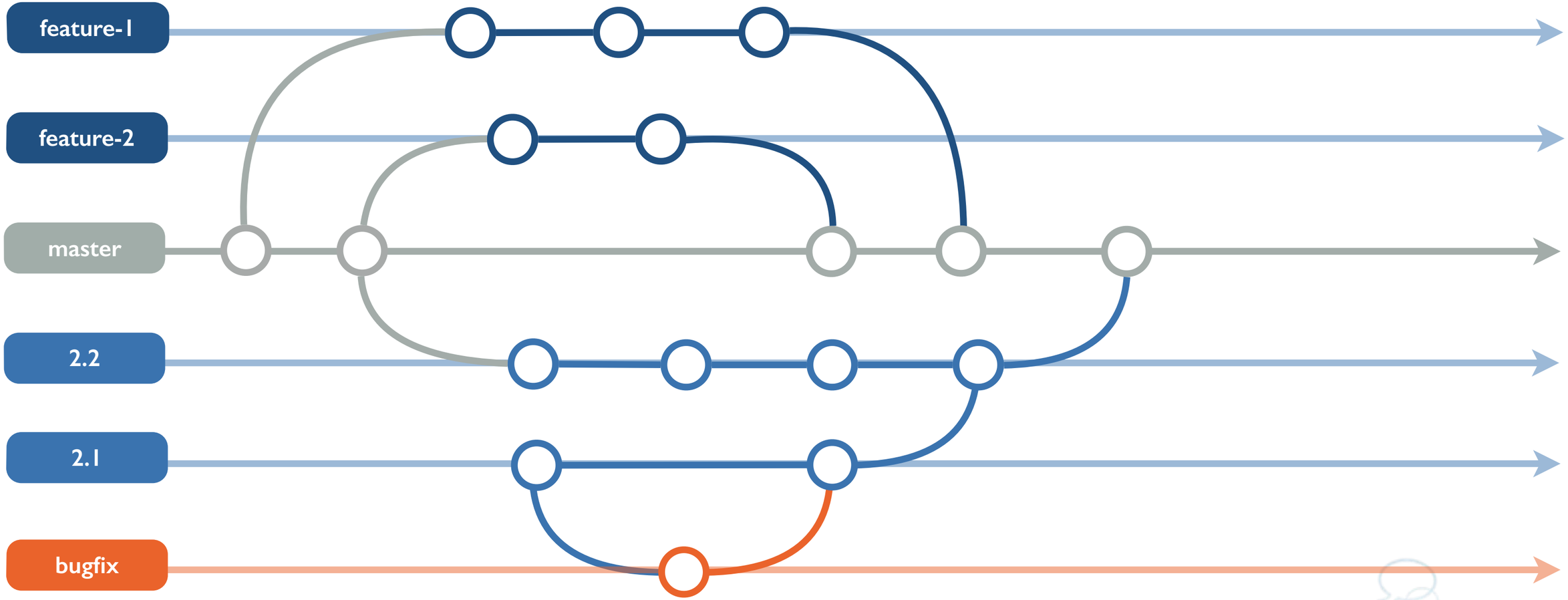

```
$> git checkout stable-2.2
$> git merge -s ours stable-2.1
```

Automatically merging release branches




```
$> git checkout stable-2.2
$> git merge -s ours stable-2.1
```

overall picture



Deep breath, it's really simple

The secret sauce

The **merge protocol**

The secret sauce

The **merge protocol**

When a branch is:	*Change flows from branch to baseline:*	*Change flows from baseline to branch:*

The secret sauce

The **merge protocol**

When a branch is:	*Change flows from branch to baseline:*	*Change flows from baseline to branch:*
More stable *than its baseline*		

The secret sauce

The **merge protocol**

When a branch is:	*Change flows from branch to baseline:*	*Change flows from baseline to branch:*
More stable *than its baseline* Release branch		

The secret sauce

The **merge protocol**

When a branch is:	*Change flows from branch to baseline:*	*Change flows from baseline to branch:*
More stable *than its baseline* Release branch	*Continually*	

The secret sauce

The **merge protocol**

When a branch is:	*Change flows from branch to baseline:*	*Change flows from baseline to branch:*
More stable *than its baseline* Release branch	*Continually*	*Never*

The secret sauce

The **merge protocol**

When a branch is:	*Change flows from branch to baseline:*	*Change flows from baseline to branch:*
More stable *than its baseline* Release branch	*Continually*	*Never*
Less stable *than its baseline*		

The secret sauce

The **merge protocol**

When a branch is:	*Change flows from branch to baseline:*	*Change flows from baseline to branch:*
More stable *than its baseline* Release branch	*Continually*	*Never*
Less stable *than its baseline* Feature branches		

The secret sauce

The **merge protocol**

When a branch is:	*Change flows from branch to baseline:*	*Change flows from baseline to branch:*
More stable *than its baseline* Release branch	*Continually*	*Never*
Less stable *than its baseline* Feature branches	*When code complete*	

The secret sauce

The **merge protocol**

When a branch is:	*Change flows from branch to baseline:*	*Change flows from baseline to branch:*
More stable *than its baseline* Release branch	*Continually*	*Never*
Less stable *than its baseline* Feature branches	*When code complete*	*Continually*

The secret sauce

The **merge protocol**

When a branch is:	*Change flows from branch to baseline:*	*Change flows from baseline to branch:*
More stable *than its baseline* Release branch	*Continually*	*Never*
Less stable *than its baseline* Feature branches	*When code complete*	*Continually*

Credit: Laura Wingerd - The Flow of change

The secret sauce

The **merge protocol**

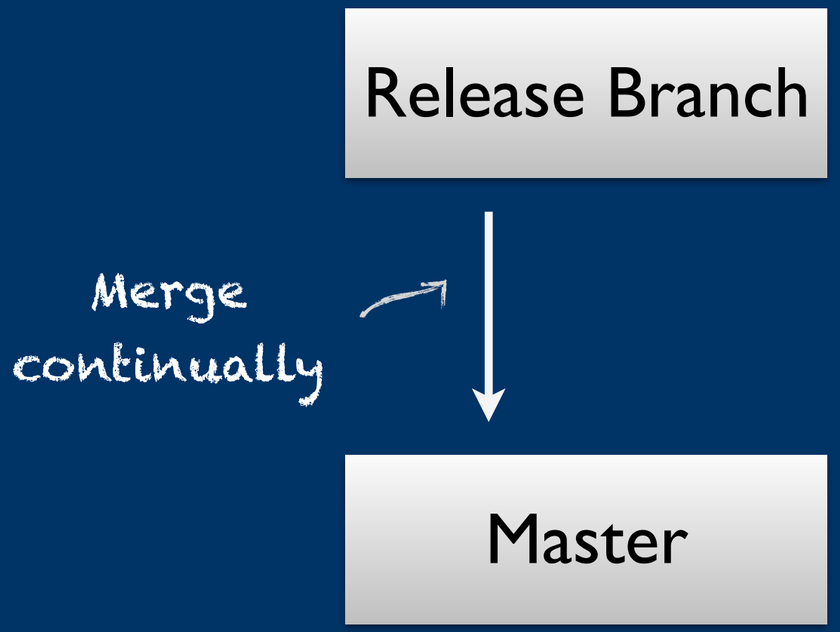
Release Branch

Master

http://bit.ly/branch-based-workflows

The secret sauce

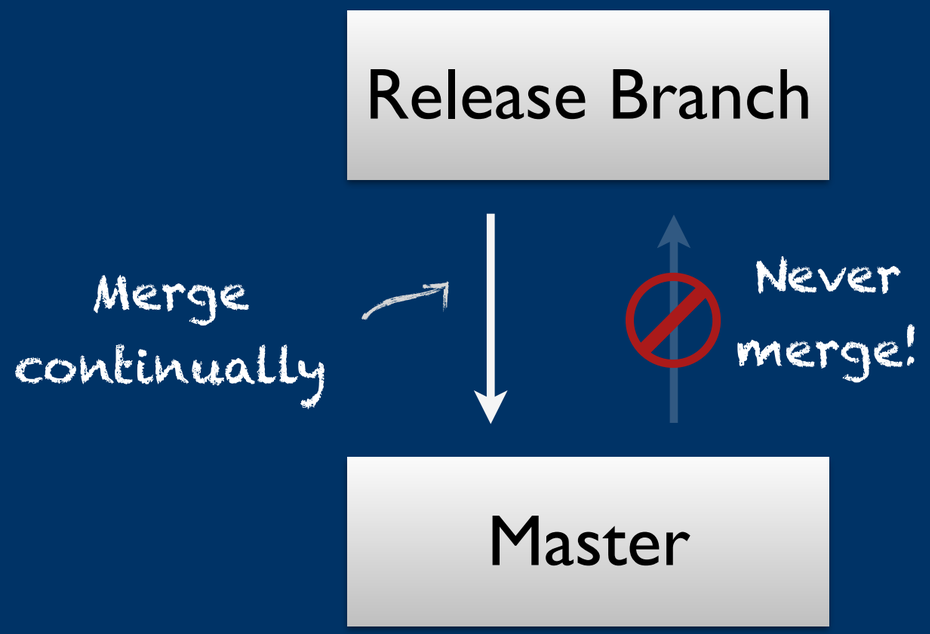
The **merge protocol**



http://bit.ly/branch-based-workflows

The secret sauce

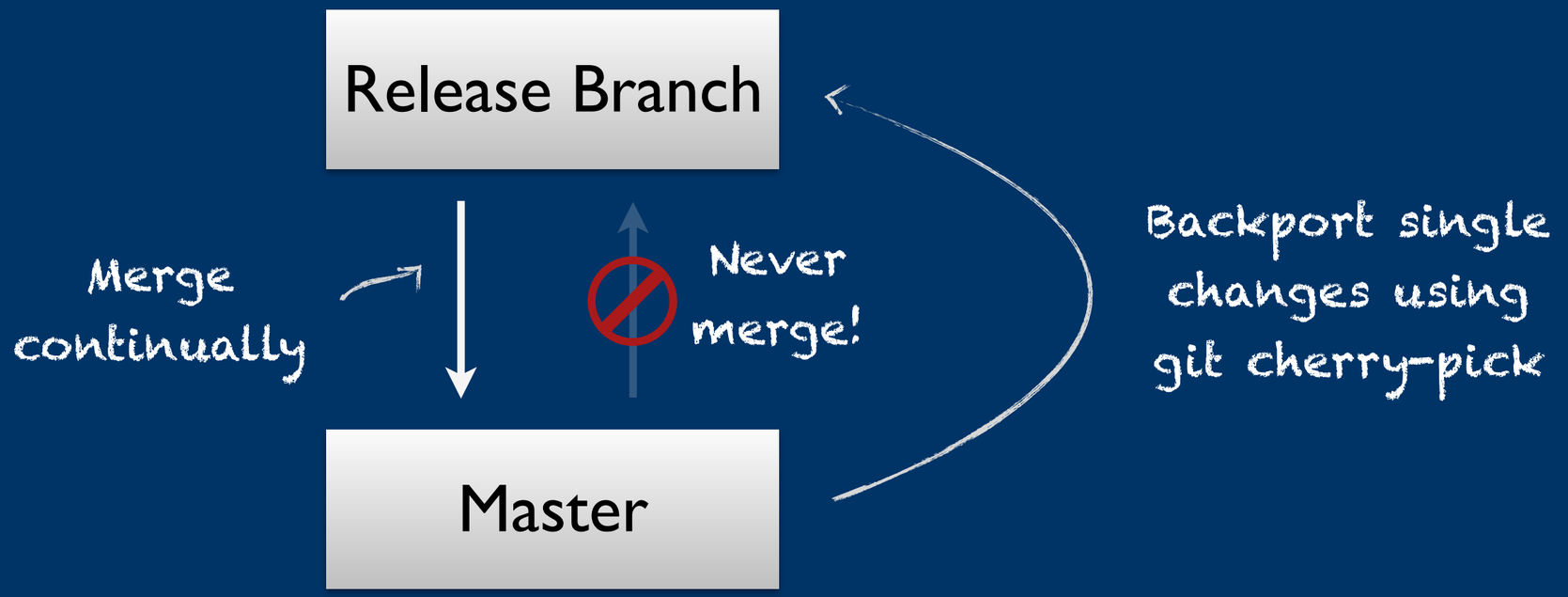
The **merge protocol**



http://bit.ly/branch-based-workflows

The secret sauce

The **merge protocol**



http://bit.ly/branch-based-workflows

The secret sauce

The **merge protocol**

http://bit.ly/branch-based-workflows

More on **git** workflows

atlassian.com/git

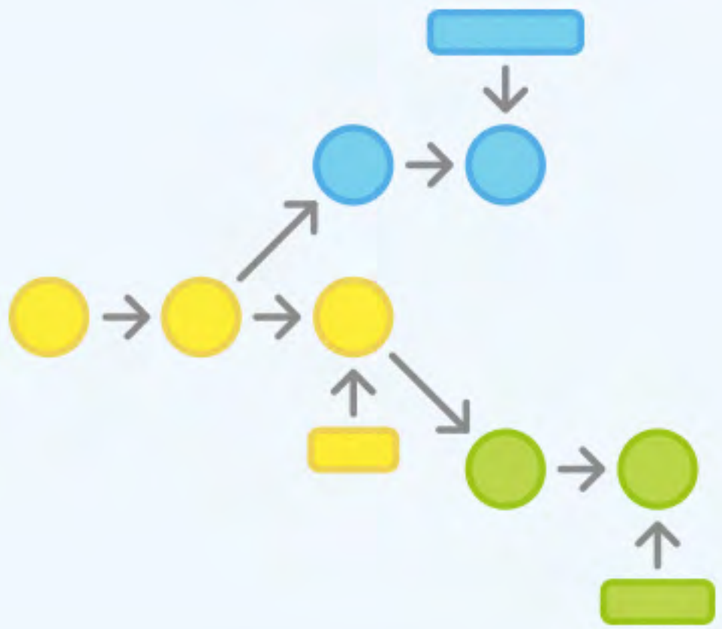
Git Tutorials

Overview | Git Tutorials | Git Workflows | Git Resources

Git Workflows

The array of possible workflows can make it hard to know where to begin when implementing Git in the workplace. This page provides a starting point by surveying the most common Git workflows for enterprise teams.

As you read through, remember that these workflows are designed to be guidelines rather than concrete rules. We want to show you what's possible, so you can mix and match aspects from different workflows to suit your individual needs.

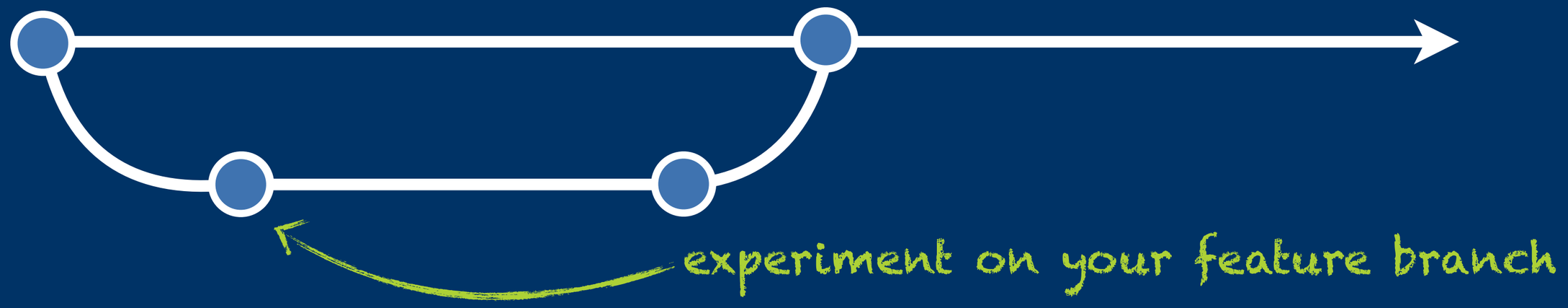


Practices

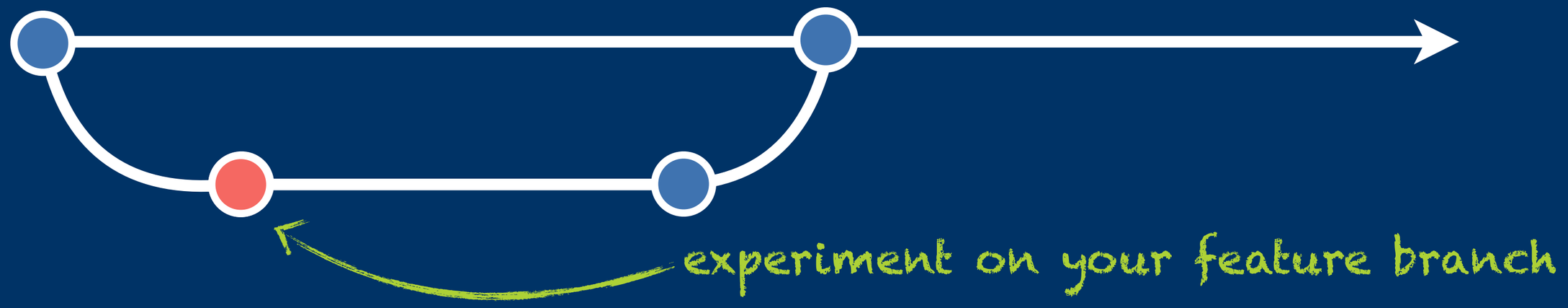
4 Practices

What happens to CI with git?

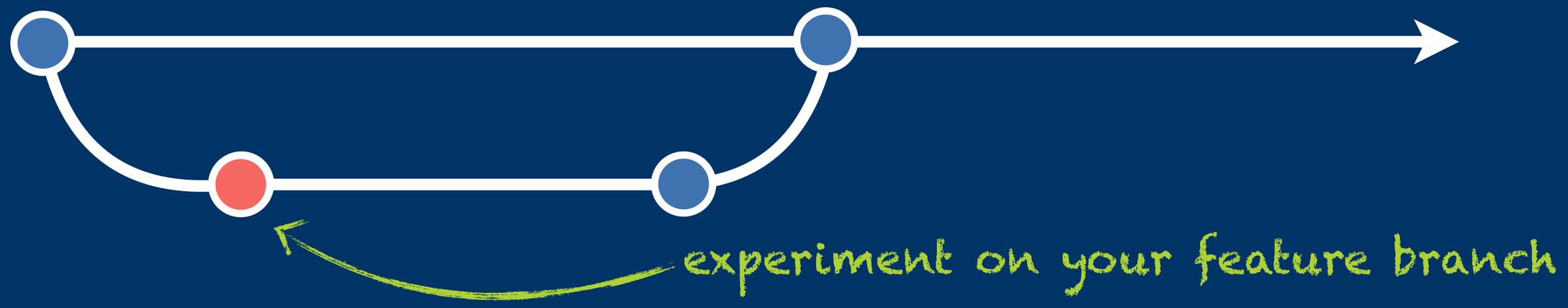
Running builds on feature branches



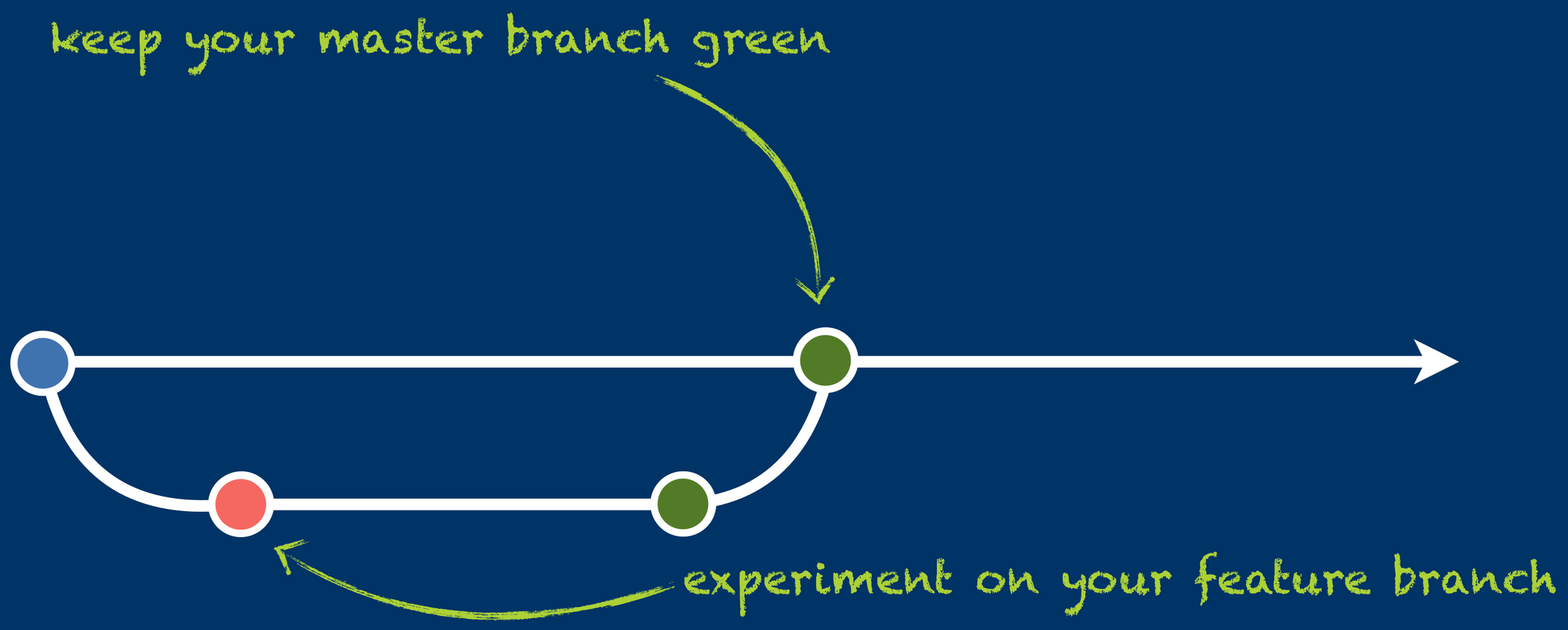
Running builds on feature branches



Running builds on feature branches



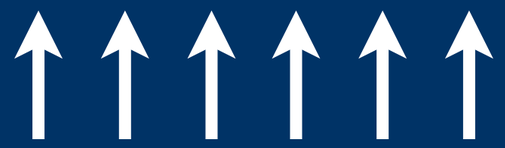
Running builds on feature branches



x developer

x developer

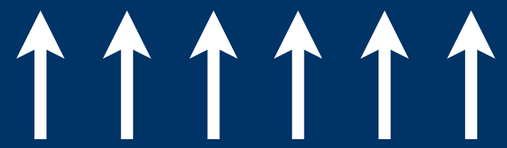
times



x push to remote

x developer

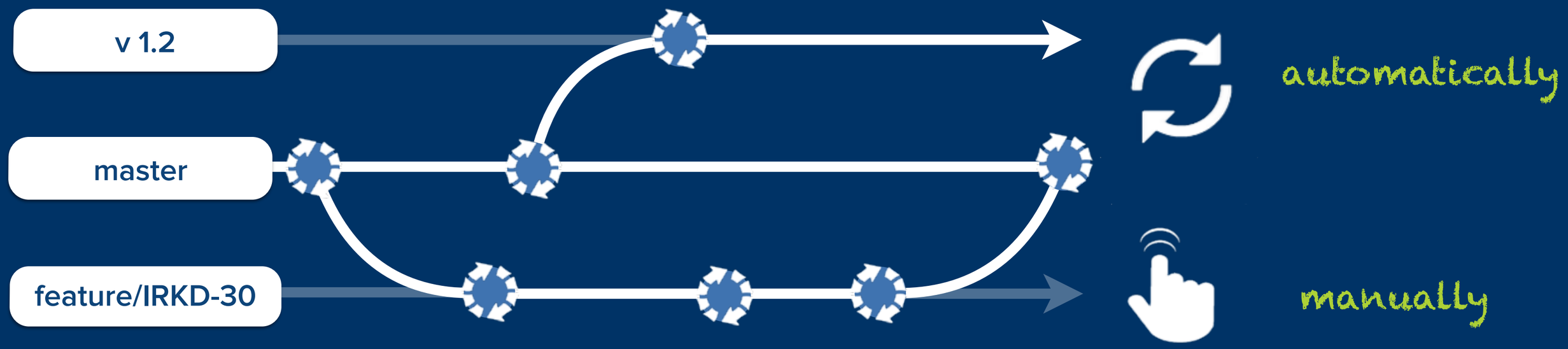
times



x push to remote

lots of builds (waiting)

Running builds on feature branches



Code Reviews

Code Reviews

Better Quality

Find errors

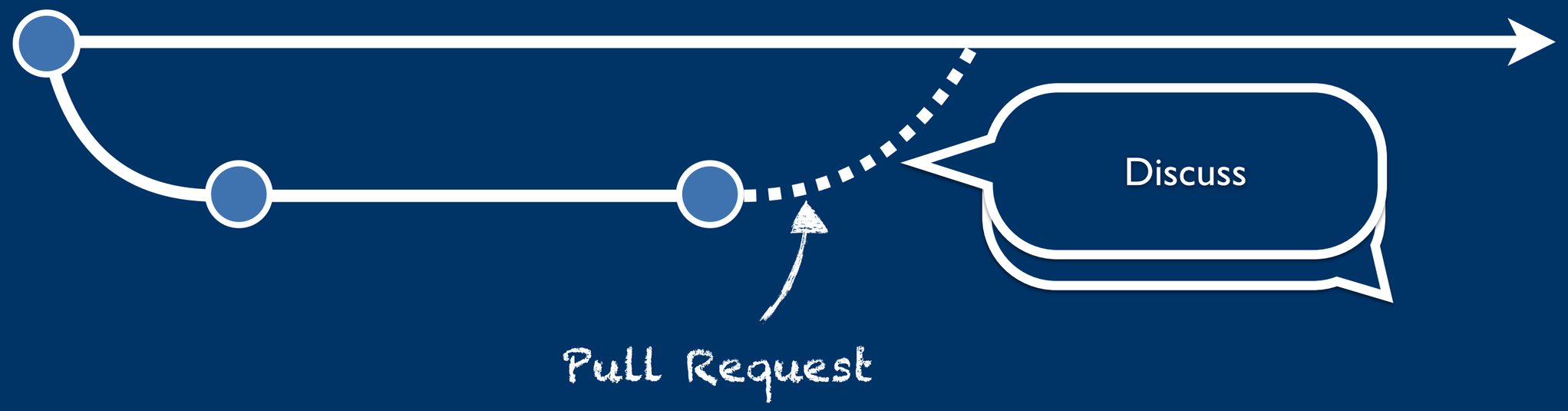
Learn

Pull Requests

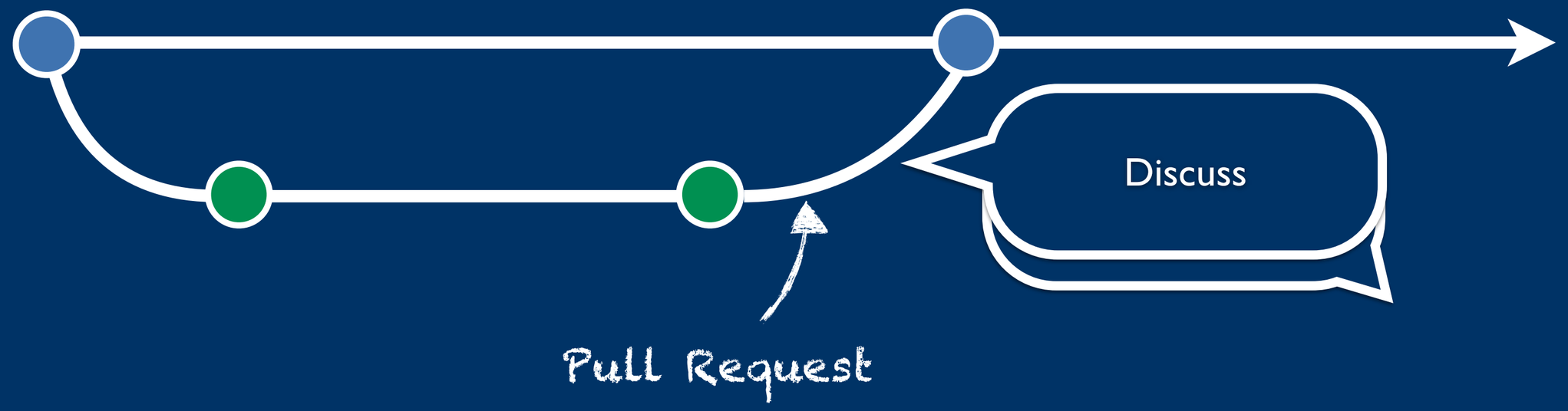
part of your daily workflow

Part of your daily work

Part of your daily work



Part of your daily work



In **Conclusion**: the recipe

Collaboration Model

Branching Model

Adopt Git Practices

Collaboration Model	Branching Model	Adopt Git Practices
Centralized		

Collaboration Model	Branching Model	Adopt Git Practices
Centralized	Product workflow	

Collaboration Model	Branching Model	Adopt Git Practices
Centralized	Product workflow Continuous delivery workflow	

Collaboration Model	Branching Model	Adopt Git Practices
Centralized	Product workflow Continuous delivery workflow Merge protocol	

Collaboration Model	Branching Model	Adopt Git Practices
Centralized	Product workflow Continuous delivery workflow Merge protocol	Build automatically, but leave knobs!

Collaboration Model	Branching Model	Adopt Git Practices
Centralized	Product workflow Continuous delivery workflow Merge protocol	Embrace PR Build automatically, but leave knobs!

Thank you for your attention!

@stefansaasen

ssaasen@atlassian.com

www.atlassian.com

Credits

http://www.flickr.com/photos/45143319@N00/3888895871/

http://www.flickr.com/photos/40145521@N00/460270581/

http://www.flickr.com/photos/41864721@N00/4647696349

http://www.flickr.com/photos48889052497@N01/12613483263/

http://www.flickr.com/photos/30928442@N08/4766664095/

http://www.flickr.com/photos/marfis75/3272079115/