

СІ БОЛЬШИХ ПРОЕКТОВ В JetBrains

@antonarhipov



Anton Arhipov

@antonarhipov

Developer Advocate @ JetBrains. TeamCity, IntelliJ IDEA. Java Champion. Kotlin enthusiast. Co-host of [@razbor_poletov](#) podcast. Co-host of [devclub.eu](#)

 Tallinn  Joined December 2009

Hello!

Привет!

Tere!



Anton Arhipov

@antonarhipov

Developer Advocate @ JetBrains. TeamCity, IntelliJ IDEA. Java Champion. Kotlin enthusiast. Co-host of [@razbor_poletov](#) podcast. Co-host of [devclub.eu](#)

 Tallinn  Joined December 2009



ПРЕВОЗМОГАЯ

ТРУДНОСТИ

**Что такое
“большой проект”?**

Количество

Количество

... строк кода

Количество

... строк кода

... модулей

Количество

... строк кода

... модулей

... правок в единицу времени

Количество

... строк кода

... модулей

... правок в единицу времени

... веток в репозитории

Количество

... строк кода

... модулей

... правок в единицу времени

... веток в репозитории

... ресурсов для сборки

Количество

... строк кода

... модулей

... правок в единицу времени

... веток в репозитории

... ресурсов для сборки

... людей причастных к процессу

Главные действующие лица

Главные действующие лица



Главные действующие лица



Главные действующие лица



TeamCity @ JetBrains

4000+ проектов

27000+ конфигураций

~1000 агентов



est. 2006



Projects | ▾

Changes

Agents 953



Build Queue 313



Anton Arhipov | ▾

Administration



Administration

Project-related Settings

Projects

All Builds

Build Time

Disk Usage

Server Health

Audit

Projects

nts

You have **4063** active projects with **27785** build configurations (plus **8877** archived build configurations).

Keyword:

Filter

Show archived

Filter projects and build configurations by name, ID or description





Projects | ▾

Changes

Agents

953



Build Queue

313



Anton Arhipov | ▾

Administration



Administration

Project-related Settings

Projects

All Builds

Build Time

Disk Usage

Server Health

Audit

Projects

nts

You have **4063** active projects with **27785** build configurations (plus **8877** archived build configurations).

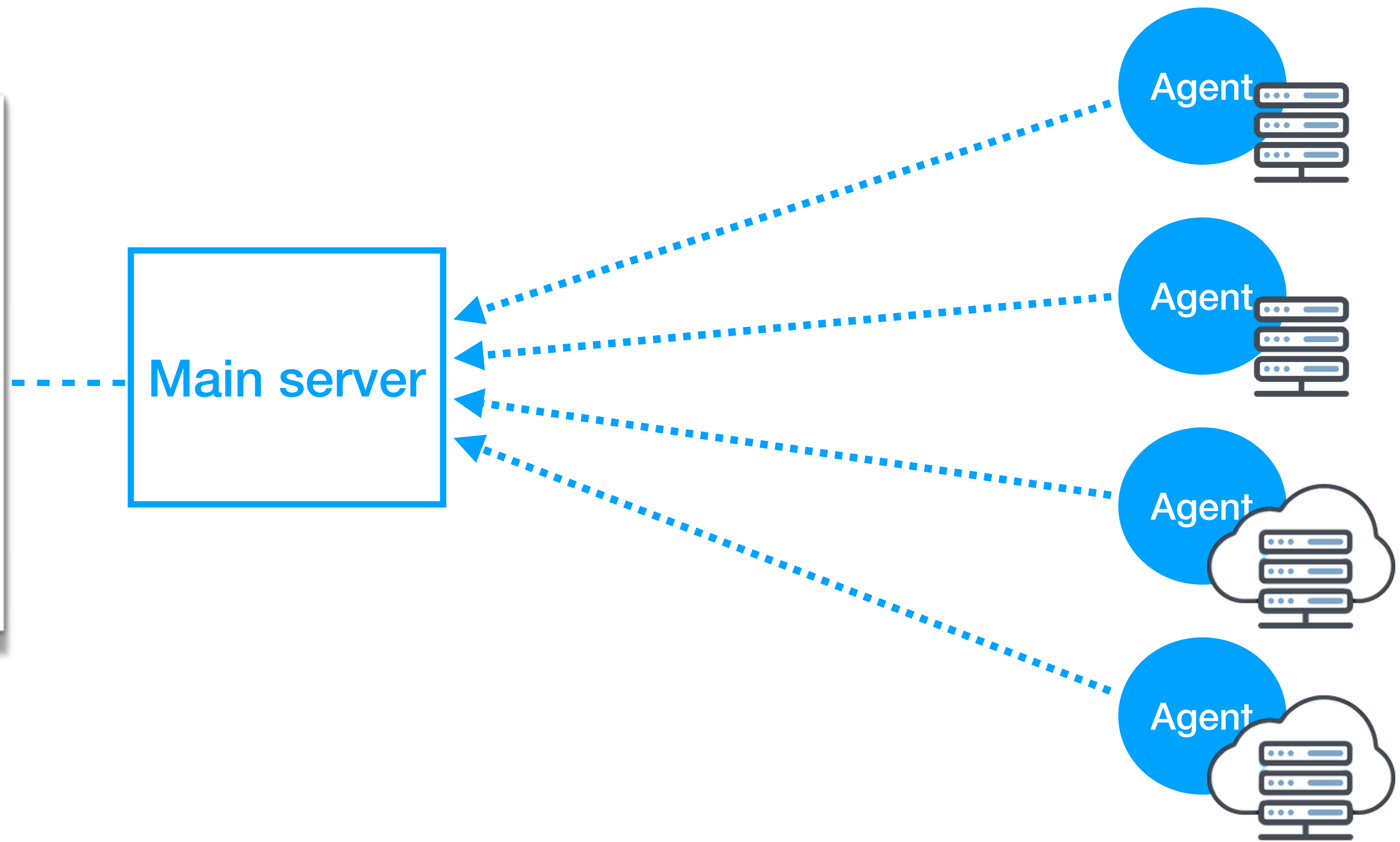
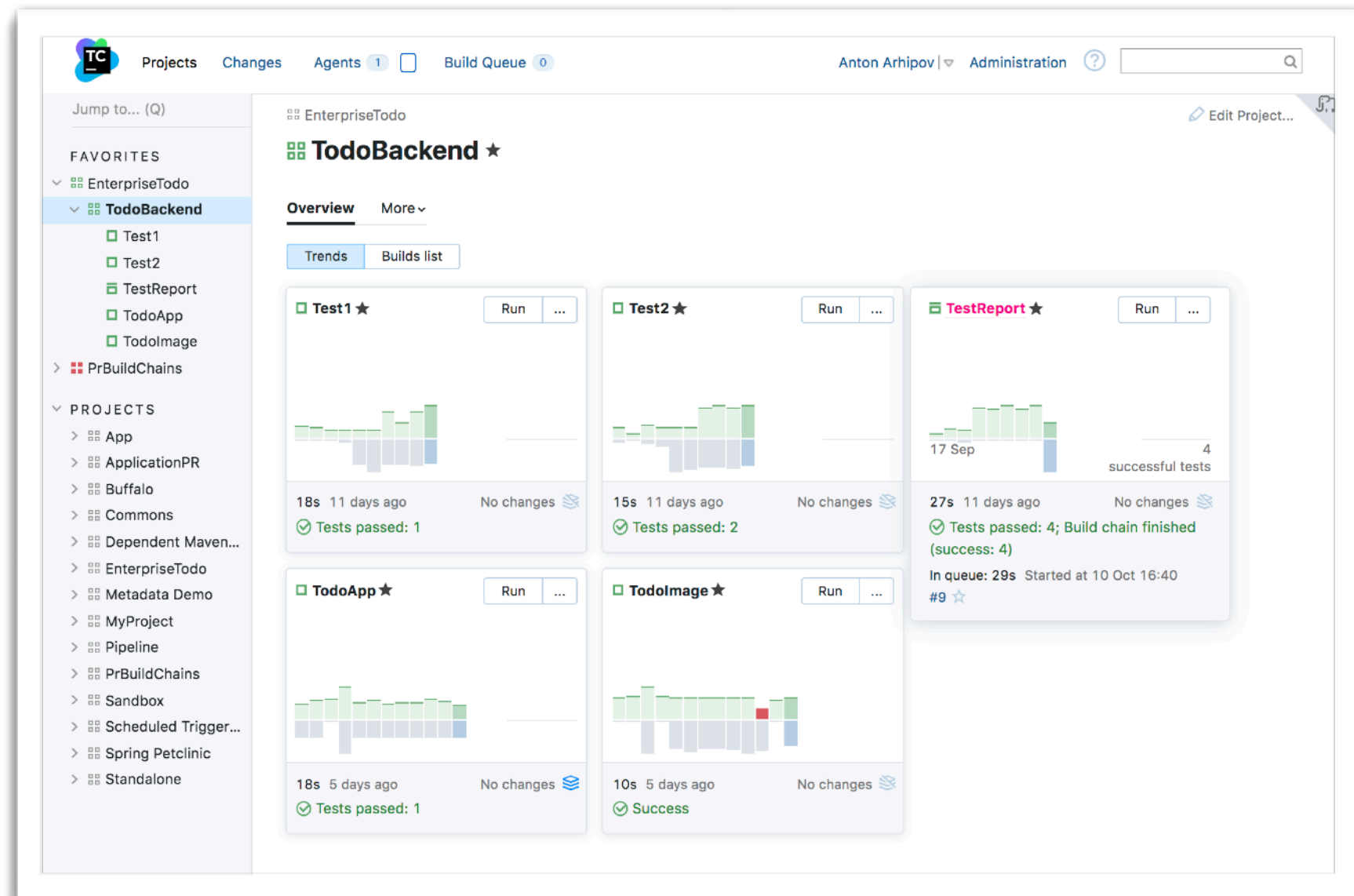
Keyword:

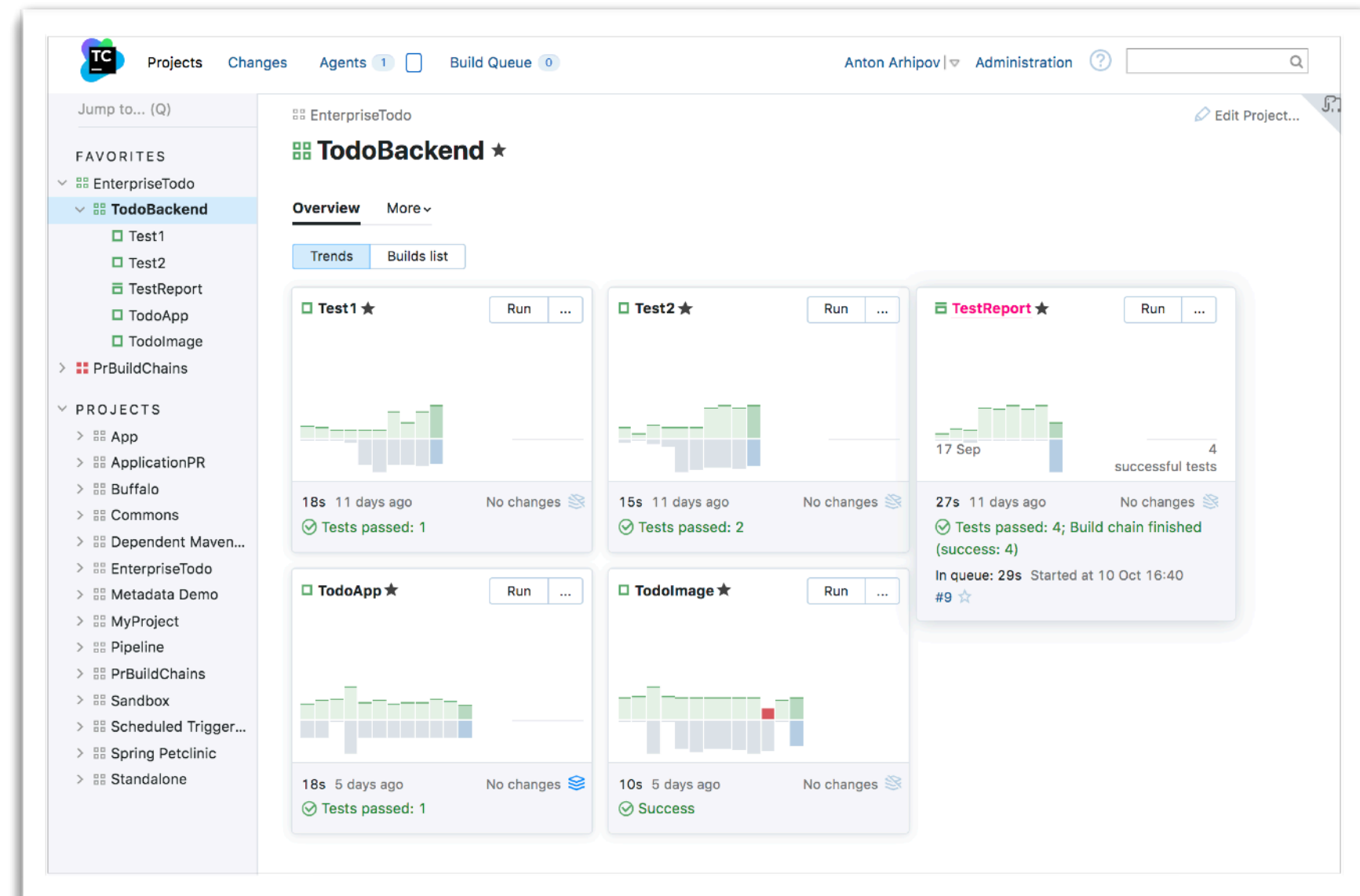
Filter

Show archived

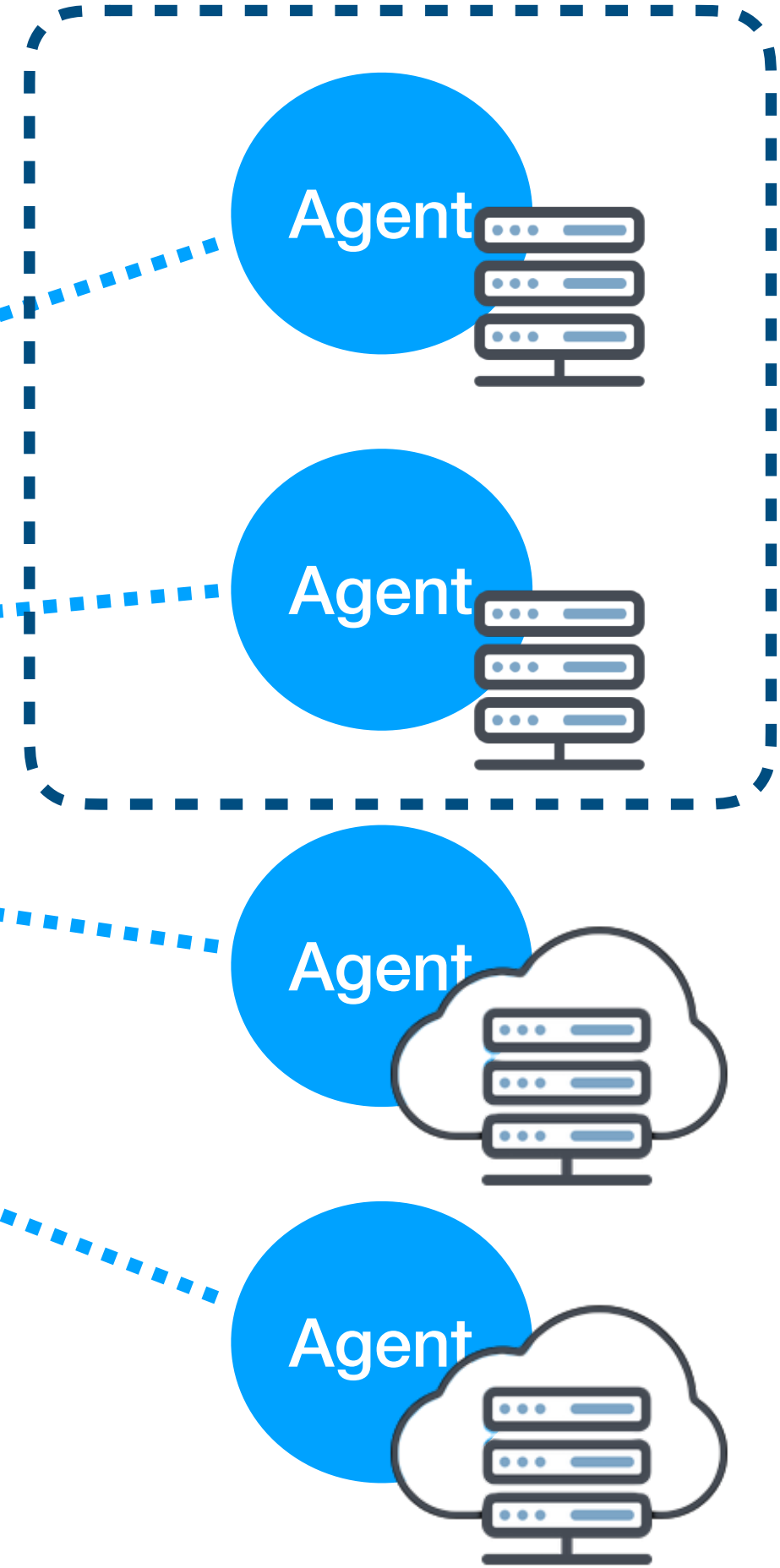
Filter projects and build configurations by name, ID or description

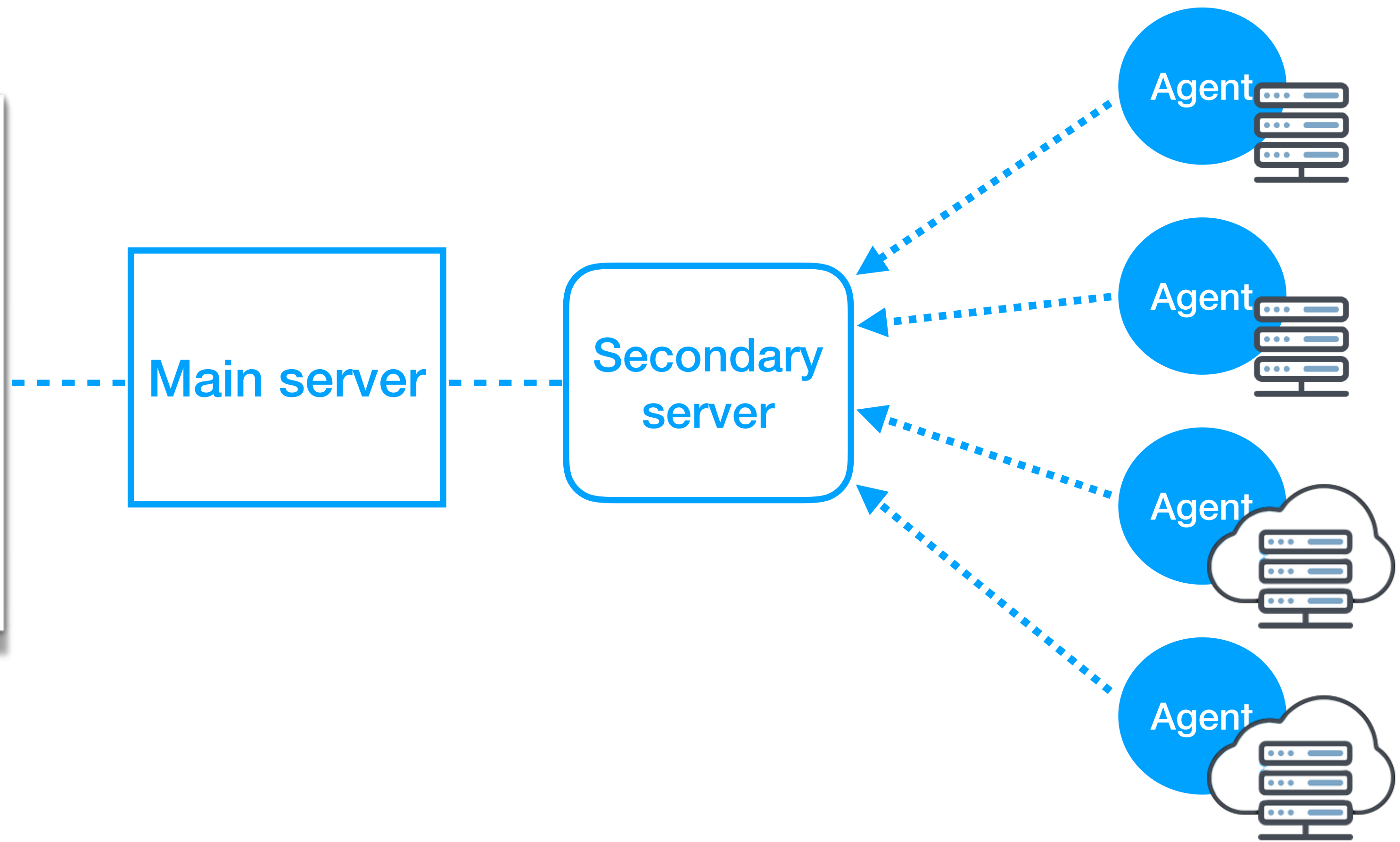
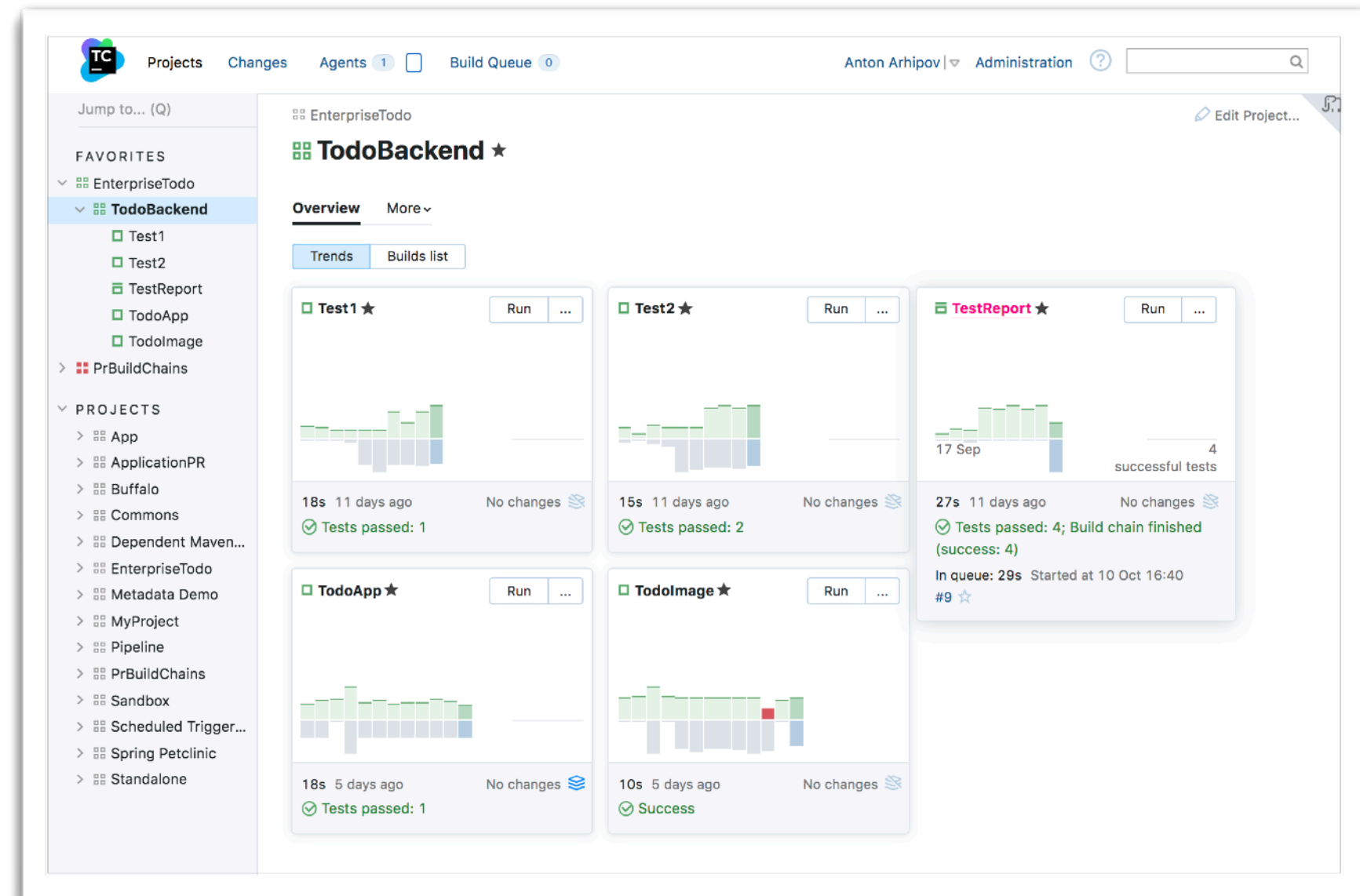


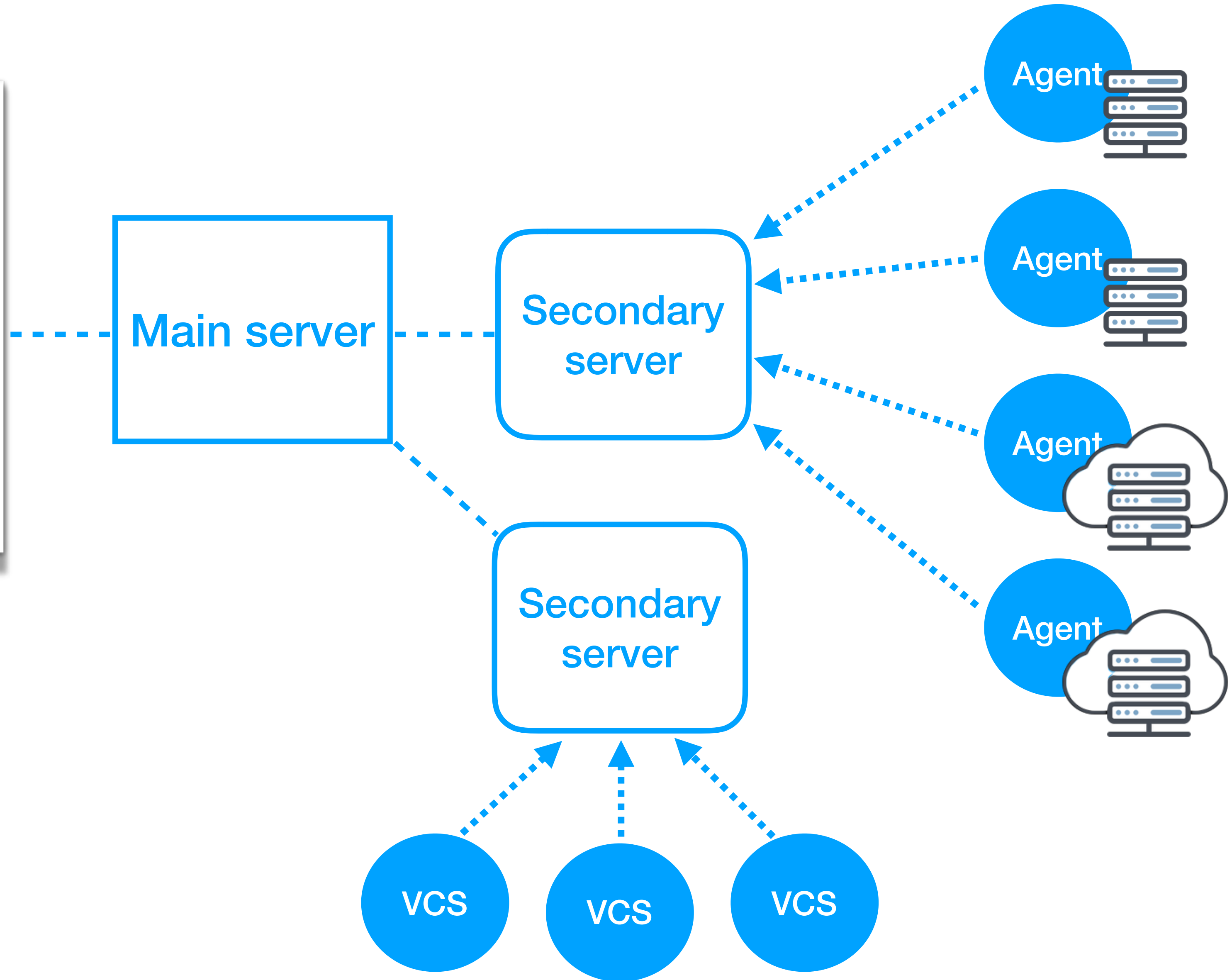
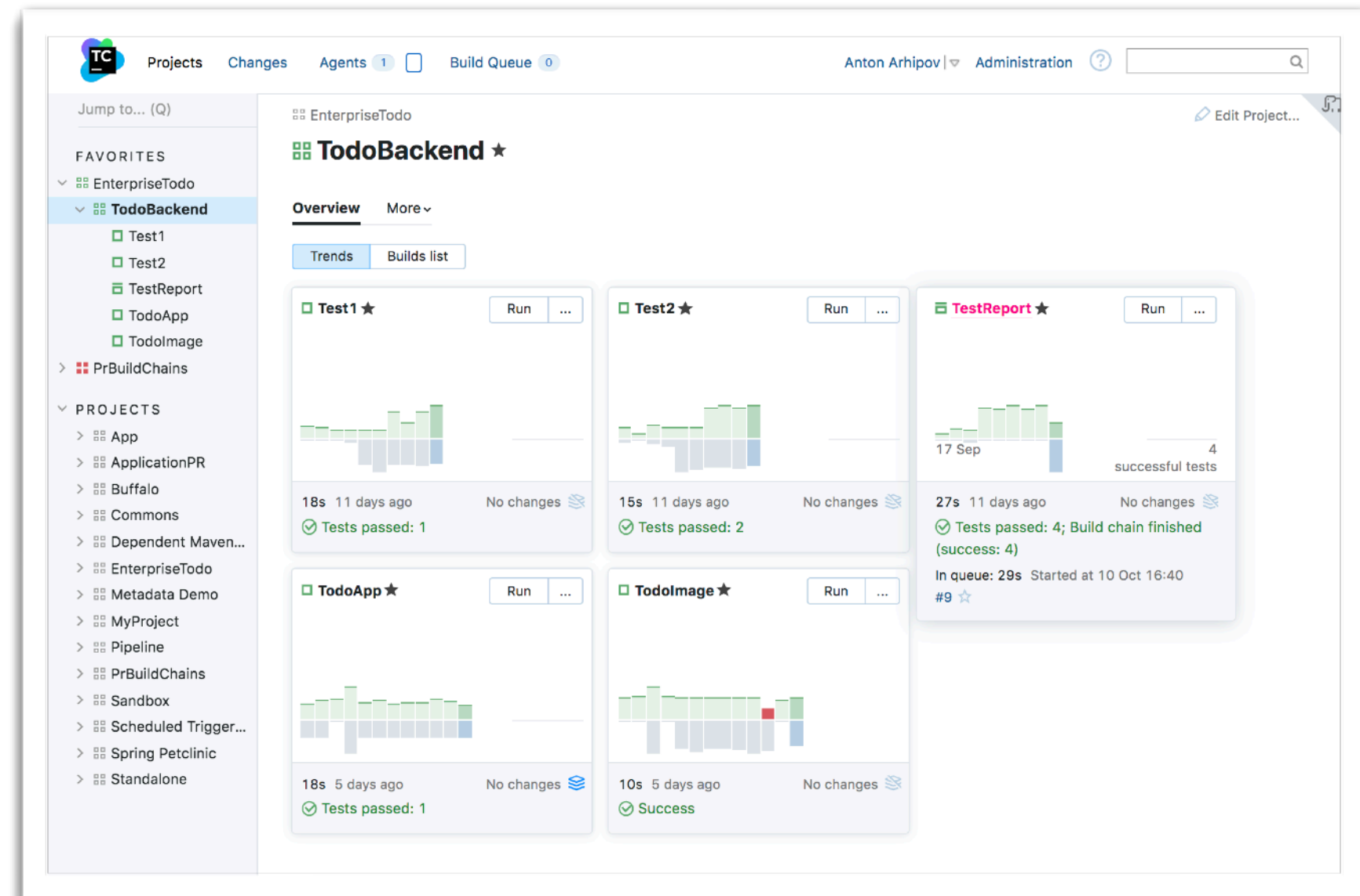




Agent pool







Build configurations

Build chains

Feature branches

Checkout rules

Pull requests

Personal builds

Cleanup

Secondary node

RBAC

Projects hierarchy

Agent pools

...

Build configurations

Build chains

Feature branches

Checkout rules

Pull requests

Personal builds

Cleanup

Secondary node

RBAC

Projects hierarchy

Agent pools

...

General Settings

Version Control Settings 1

Build Steps 3

Triggers 1

Failure Conditions

Build Features

Dependencies

Parameters

Agent Requirements 1

Suggestions

« Hide unconfigured




Build Steps

In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool. [?](#)

+ Add build step

Reorder build steps

 Auto-detect build steps

Build Step	Parameters Description	
1. Gradle	Gradle tasks: clean build Use wrapper script: yes Execute: If all previous steps finished successfully	Edit 
2. Docker	Docker build; Dockerfile location: Dockerfile Execute: If all previous steps finished successfully	Edit 
3. Docker	docker push antonarhipov/app:%build.number%; remove image(s) after push Execute: If all previous steps finished successfully	Edit 

General Settings

Version Control Settings 1

Build Steps 3

Triggers 1

Failure Conditions

Build Features

Dependencies

Parameters

Agent Requirements 1

Suggestions

« Hide unconfigured

```
object Build : BuildType({
    name = "Build"

    vcs {
        root(AppVcs)
    }

    steps {
        gradle {
            tasks = "clean build"
        }
        dockerCommand {
            commandType = build {
                source = path {
                    path = "Dockerfile"
                }
            }
        }
        dockerCommand {
            commandType = push {
                namesAndTags = "antonarhipov/app:%build.number%"
            }
        }
    }

    triggers {
        vcs { }
    }
})
```

General Settings

Version Control Settings 1

Build Steps 70

Triggers

Failure Conditions

Build Features 3

Dependencies 1

Parameters 45

Agent Requirements 1

Suggestions

« Hide unconfigured

Settings are stored in VCS (view history)

View DSL ?

Editing of the build configuration settings is disabled, reason: project uses Kotlin DSL API version v10 which doesn't support UI editing.

Build Steps

In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool. ?

Build Step	Parameters Description	
1. Set up Git	Command Line Custom script: "%env.TEAMCITY_GIT_PATH%" config user.em... (and 1 more line) Execute: If all previous steps finished successfully	View
2. Download Bunch	Ant Build file: custom Targets: default Execute: If all previous steps finished successfully	View
3. Core libraries and compiler dist	Gradle Gradle tasks: dist Use wrapper script: yes Execute: If all previous steps finished successfully	View
4. Additional common IDE artifacts	Gradle Gradle tasks: :prepare:formatter:jar :prepare:ide-lazy-resolver:jar Use wrapper script: yes Execute: If all previous steps finished successfully	View
5. Zip Artifacts (compiler)	Gradle Gradle tasks: zipCompiler zipTestData Use wrapper script: yes Execute: If all previous steps finished successfully	View
6. Publish Artifacts (compiler)	Ant Build file: custom Targets: default Execute: If all previous steps finished successfully	View
7. Get private local maven repo path	Maven Path to POM: libraries/pom.xml Goals: <not specified> Execute: Even if some of the previous steps failed	View
8. Install artifacts to local maven repo	Gradle Gradle tasks: install publish Use wrapper script: yes Execute: Even if some of the previous steps failed	View

General Settings

Version Control Settings

Build Steps 70 ←

Triggers

Failure Conditions

Build Features 3

Dependencies 1

Parameters 45

Agent Requirements 1

Suggestions

« Hide unconfigured

Settings are stored in VCS (view history)

View DSL ?

Editing of the build configuration settings is disabled, reason: project uses Kotlin DSL API version v10 which doesn't support UI editing.

Build Steps

In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool. ?

Build Step	Parameters Description	
1. Set up Git	Command Line Custom script: "%env.TEAMCITY_GIT_PATH%" config user.em... (and 1 more line) Execute: If all previous steps finished successfully	View
2. Download Bunch	Ant Build file: custom Targets: default Execute: If all previous steps finished successfully	View
3. Core libraries and compiler dist	Gradle Gradle tasks: dist Use wrapper script: yes Execute: If all previous steps finished successfully	View
4. Additional common IDE artifacts	Gradle Gradle tasks: :prepare:formatter:jar :prepare:ide-lazy-resolver:jar Use wrapper script: yes Execute: If all previous steps finished successfully	View
5. Zip Artifacts (compiler)	Gradle Gradle tasks: zipCompiler zipTestData Use wrapper script: yes Execute: If all previous steps finished successfully	View
6. Publish Artifacts (compiler)	Ant Build file: custom Targets: default Execute: If all previous steps finished successfully	View
7. Get private local maven repo path	Maven Path to POM: libraries/pom.xml Goals: <not specified> Execute: Even if some of the previous steps failed	View
8. Install artifacts to local maven repo	Gradle Gradle tasks: install publish Use wrapper script: yes Execute: Even if some of the previous steps failed	View

General Settings

Version Control Settings

Build Steps 70 ←

Triggers

Failure Conditions

Build Features 3

Dependencies 1

Parameters 45

Agent Requirements 1

Suggestions

« Hide unconfigured

Settings are stored in VCS (view history)

View DSL ?

Editing of the build configuration settings is disabled, reason: project uses Kotlin DSL API version v10 which doesn't support UI editing.

Build Steps

In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool. ?

Build Step	Parameters Description		
1. Set up Git	Command Line Custom script: "%env.TEAMCITY_GIT_PATH%" config user.em... (and 1 more line) Execute: If all previous steps finished successfully	CLI	View
2. Download Bunch	Ant Build file: custom Targets: default Execute: If all previous steps finished successfully	Ant	View
3. Core libraries and compiler dist	Gradle Gradle tasks: dist Use wrapper script: yes Execute: If all previous steps finished successfully	Gradle	View
4. Additional common IDE artifacts	Gradle Gradle tasks: :prepare:formatter:jar :prepare:ide-lazy-resolver:jar Use wrapper script: yes Execute: If all previous steps finished successfully	Gradle	View
5. Zip Artifacts (compiler)	Gradle Gradle tasks: zipCompiler zipTestData Use wrapper script: yes Execute: If all previous steps finished successfully	Gradle	View
6. Publish Artifacts (compiler)	Ant Build file: custom Targets: default Execute: If all previous steps finished successfully	Ant	View
7. Get private local maven repo path	Maven Path to POM: libraries/pom.xml Goals: <not specified> Execute: Even if some of the previous steps failed	Maven	View
8. Install artifacts to local maven repo	Gradle Gradle tasks: install publish Use wrapper script: yes Execute: Even if some of the previous steps failed	Gradle	View

General Settings

Version Control Settings 1

Build Steps 70

Triggers

Failure Conditions

Build Features 3

Dependencies 1

Parameters 45

Agent Requirements 1

Suggestions

« Hide unconfigured

Settings are stored in VCS (view history)

View DSL ?

Editing of the build configuration settings is disabled, reason: project uses Kotlin DSL API version v10 which doesn't support UI editing.

Build Steps

In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool. ?

Build Step	Parameters Description	
1. Set up Git	Command Line Custom script: "%env.TEAMCITY_GIT_PATH%" config user.em... (and 1 more line) Execute: If all previous steps finished successfully	View
2. Download Bunch	Ant Build file: custom Targets: default Execute: If all previous steps finished successfully	View
3. Core libraries and compiler dist	Gradle Gradle tasks: dist Use wrapper script: yes Execute: If all previous steps finished successfully	View
4. Additional common IDE artifacts	Gradle Gradle tasks: :prepare:formatter:jar :prepare:ide-lazy-resolver:jar Use wrapper script: yes Execute: If all previous steps finished successfully	View
5. Zip Artifacts (compiler)	Gradle Gradle tasks: zipCompiler zipTestData Use wrapper script: yes Execute: If all previous steps finished successfully	View
6. Publish Artifacts (compiler)	Ant Build file: custom Targets: default Execute: If all previous steps finished successfully	View
7. Get private local maven repo path	Maven Path to POM: libraries/pom.xml Goals: <not specified> Execute: Even if some of the previous steps failed	View
8. Install artifacts to local maven repo	Gradle Gradle tasks: install publish Use wrapper script: yes Execute: Even if some of the previous steps failed	View

General Settings

Version Control Settings 1

Build Steps 70

Triggers

Failure Conditions

Build Features 3

Dependencies 1

Parameters 45

Agent Requirements 1

Suggestions

« Hide unconfigured

Settings are stored in VCS (view history)

View DSL ?

Editing of the build configuration settings is disabled, reason: project uses Kotlin DSL API version v10 which doesn't support UI editing.

Build Steps

In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool. ?

Build Step	Parameters Description	
1. Set up Git	Command Line Custom script: "%env.TEAMCITY_GIT_PATH%" config user.em... (and 1 more line) Execute: If all previous steps finished successfully	View
2. Download Bunch	Ant Build file: custom Targets: default Execute: If all previous steps finished successfully	View
3. Core libraries and compiler dist	Gradle Gradle tasks: dist Use wrapper script: yes Execute: If all previous steps finished successfully	View
4. Additional common IDE artifacts	Gradle Gradle tasks: :prepare:formatter:jar :prepare:ide-lazy-resolver:jar Use wrapper script: yes Execute: If all previous steps finished successfully	View
5. Zip Artifacts (compiler)	Gradle Gradle tasks: zipCompiler zipTestData Use wrapper script: yes Execute: If all previous steps finished successfully	View
6. Publish Artifacts (compiler)	Ant Build file: custom Targets: default Execute: If all previous steps finished successfully	View
7. Get private local maven repo path	Maven Path to POM: libraries/pom.xml Goals: <not specified> Execute: Even if some of the previous steps failed	View
8. Install artifacts to local maven repo	Gradle Gradle tasks: install publish Use wrapper script: yes Execute: Even if some of the previous steps failed	View

General Settings

Version Control Settings 1

Build Steps 70

Triggers

Failure Conditions

Build Features 3

Dependencies 1

Parameters 45

Agent Requirements 1

Suggestions

« Hide unconfigured

Settings are stored in VCS (view history)

View DSL ?

Editing of the build configuration settings is disabled, reason: project uses Kotlin DSL API version v10 which doesn't support UI editing.

Build Steps

In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool. ?

Build Step	Parameters Description	
1. Set up Git	Command Line Custom script: "%env.TEAMCITY_GIT_PATH%" config user.em... (and 1 more line) Execute: If all previous steps finished successfully	View
2. Download Bunch	Ant Build file: custom Targets: default Execute: If all previous steps finished successfully	View
3. Core libraries and compiler dist	Gradle Gradle tasks: dist Use wrapper script: yes Execute: If all previous steps finished successfully	View
4. Additional common IDE artifacts	Gradle Gradle tasks: :prepare:formatter:jar :prepare:ide-lazy-resolver:jar Use wrapper script: yes Execute: If all previous steps finished successfully	View
5. Zip Artifacts (compiler)	Gradle Gradle tasks: zipCompiler zipTestData Use wrapper script: yes Execute: If all previous steps finished successfully	View
6. Publish Artifacts (compiler)	Ant Build file: custom Targets: default Execute: If all previous steps finished successfully	View
7. Get private local maven repo path	Maven Path to POM: libraries/pom.xml Goals: <not specified> Execute: Even if some of the previous steps failed	View
8. Install artifacts to local maven repo	Gradle Gradle tasks: install publish Use wrapper script: yes Execute: Even if some of the previous steps failed	View

General Settings

Version Control Settings 1

Build Steps 70

Triggers

Failure Conditions

Build Features 3

Dependencies 1

Parameters 45

Agent Requirements 1

Suggestions

« Hide unconfigured

Settings are stored in VCS (view history)

View DSL ?

Editing of the build configuration settings is disabled, reason: project uses Kotlin DSL API version v10 which doesn't support UI editing.

Build Steps

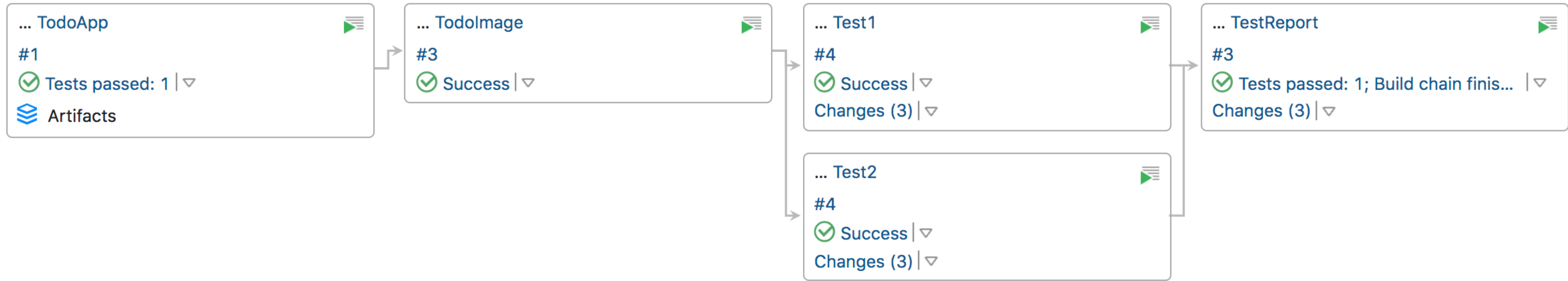
In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool. ?

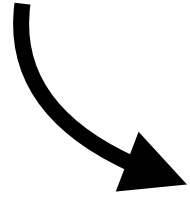
```
object BuildA : BuildType({  
  
    artifactRules = "application.zip"  
    steps {  
        gradle { }  
        ant { }  
        gradle { }  
    }  
  
})
```


```
object BuildB : BuildType({  
  
    dependencies {  
        snapshot(BuildA){ }  
        artifact(BuildA){  
            artifactRules = "application.zip"  
        }  
    }  
  
})
```

Use wrapper script: yes

Execute: Even if some of the previous steps failed







... TodoApp 

#1


✔ Tests passed: 1 | ▾

 Artifacts

... TodoImage 

#3


✔ Success | ▾

... Test1 

#4

✔ Success | ▾


Changes (3) | ▾

... Test2 

#4

✔ Success | ▾

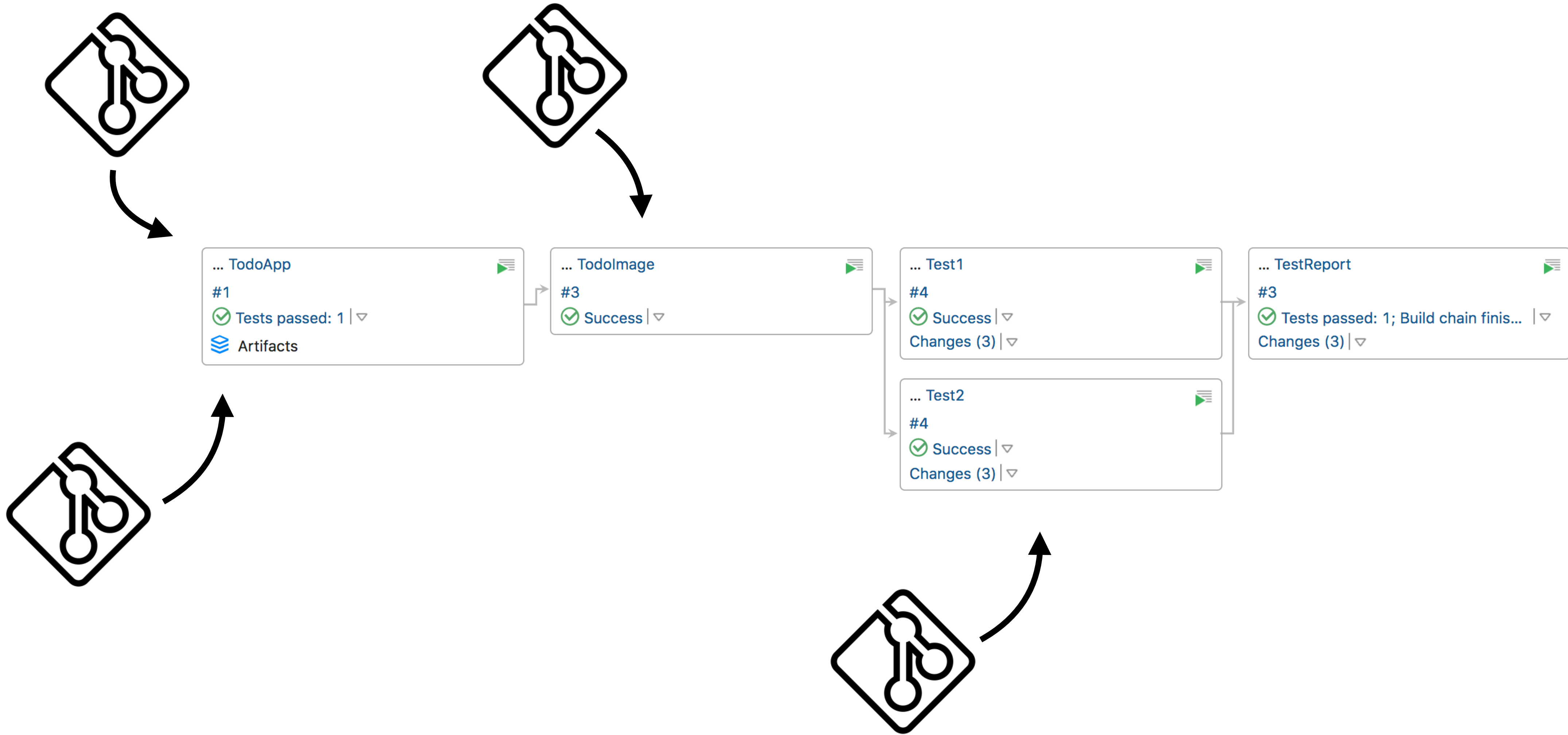
Changes (3) | ▾

... TestReport 

#3

✔ Tests passed: 1; Build chain finis... | ▾

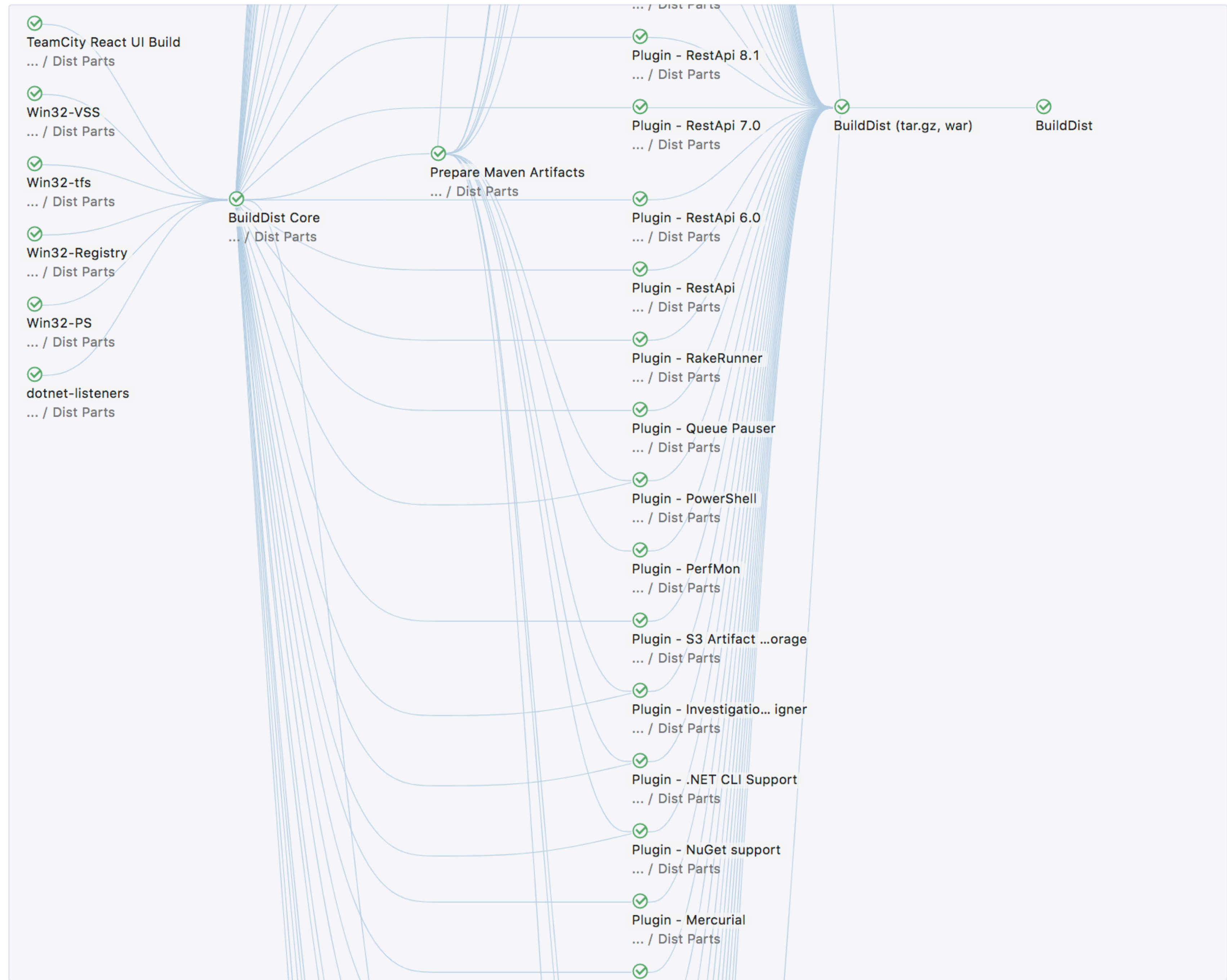
Changes (3) | ▾

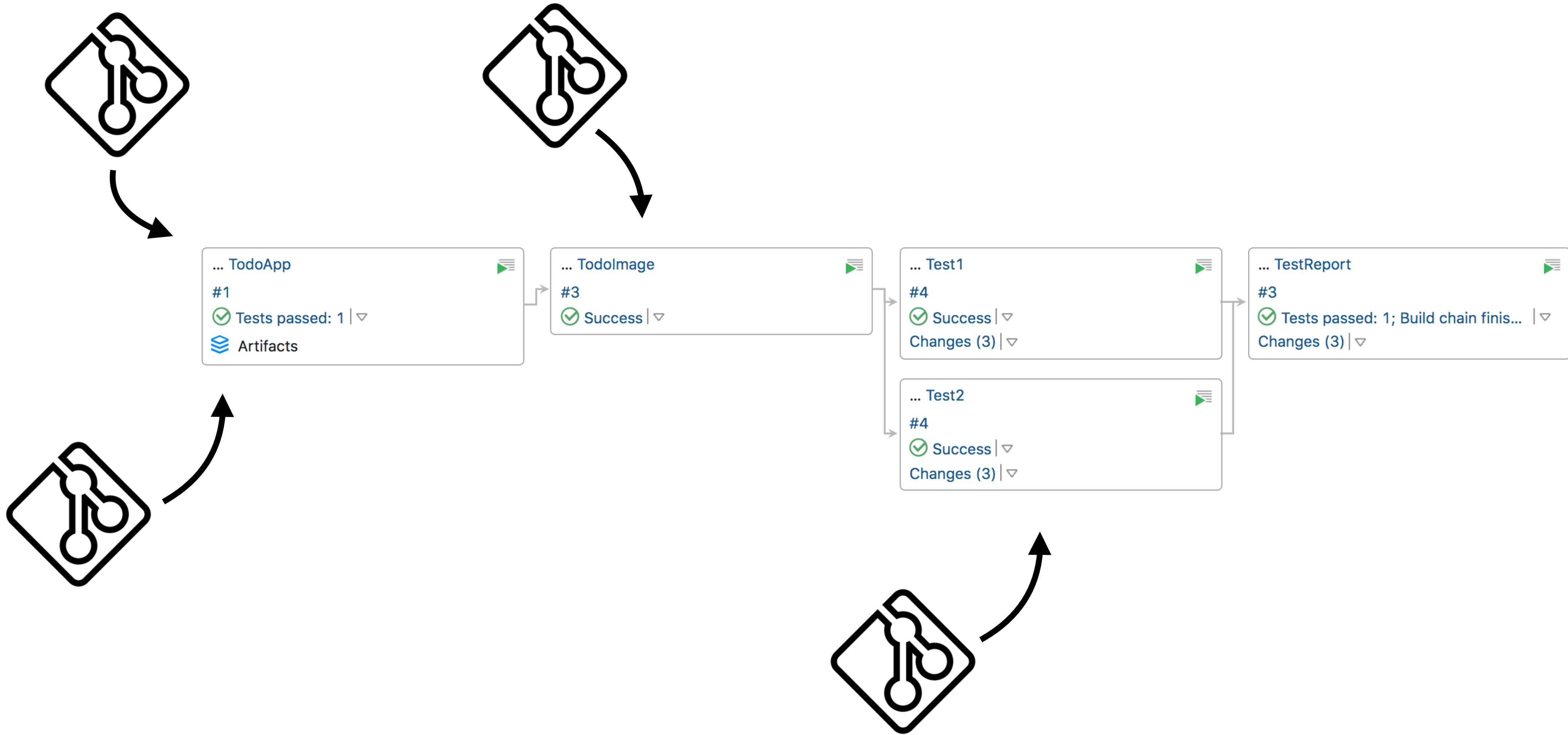


- > IDEA Language Plugins ☆
- > IntelliJ Platform Products ☆
- ▼ TeamCity ☆
 - ▼ Trunk ☆
 - ▼ Integration Tests ☆
 - ▼ Build Runner Tests ☆
 - IDEA-based Runners Te... ☆
 - ▼ IDEA Plugin Tests ☆
 - IntegrationBuild IDEA plug... ☆
 - ▼ Kanpur 2019.1 ☆
 - ▼ Integration Tests ☆
 - ▼ Build Runner Tests ☆
 - IDEA-based Runners Te... ☆
 - ▼ IDEA Plugin Tests ☆
 - IntegrationBuild IDEA plug... ☆
 - ▼ Jaipur 2018.2.x ☆
 - ▼ Integration Tests ☆
 - ▼ Build Runner Tests ☆
 - IDEA-based Runners Te... ☆
 - ▼ IDEA Plugin Tests ☆
 - IntegrationBuild IDEA plug... ☆
 - ▼ Jaipur-2018.1.x ☆
 - ▼ Integration Tests ☆
 - ▼ Build Runner Tests ☆
 - IDEA-based Runners Te... ☆
 - ▼ IDEA Plugin Tests ☆
 - IntegrationBuild IDEA plug... ☆

Need to [log in...](#) or [update data](#)

Select builds to view details and move the area or ⌘+scroll to see entire chain







1. Переиспользование результатов сборки

Alibaba

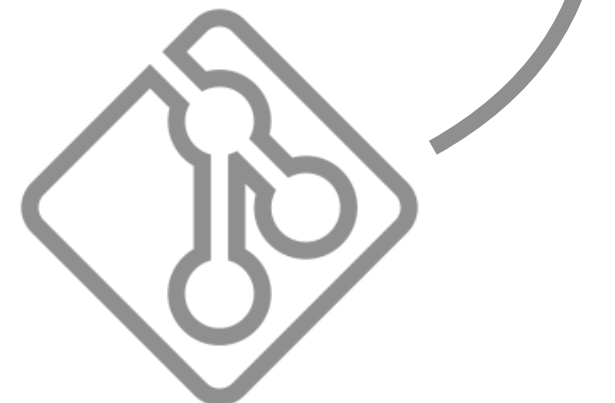
Alibaba

Alibaba

2. Синхронизация ревизий в цепочке сборок



Alibaba





1. Переиспользование результатов сборки

Artifacts

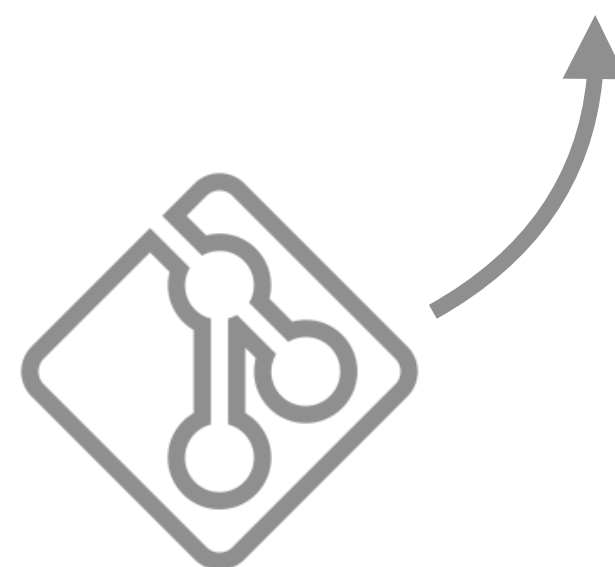
Cache

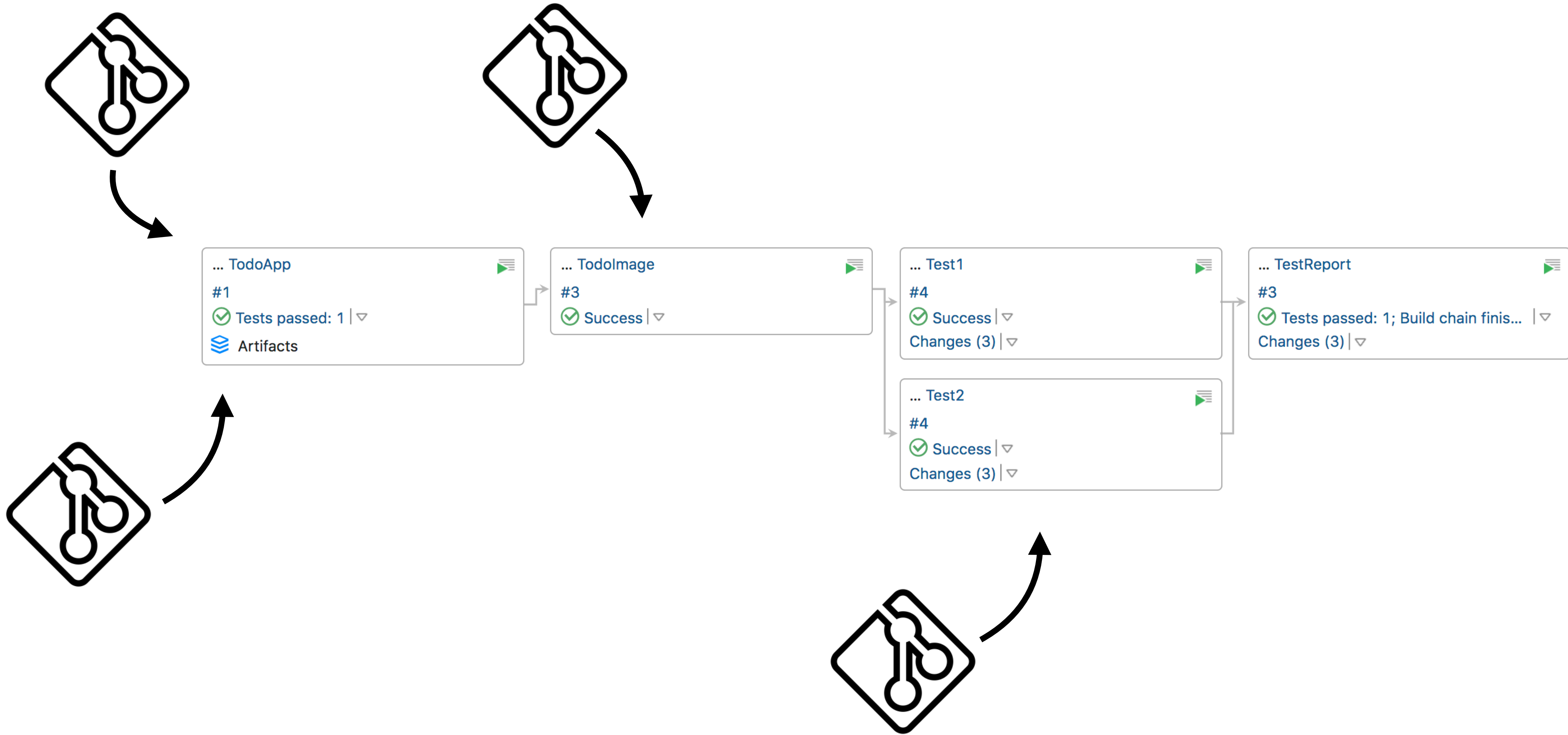
Cache

2. Синхронизация ревизий в цепочке сборок



Changeset







... TodoApp #1
 ✓ Tests passed: 1 | ▾
 Artifacts

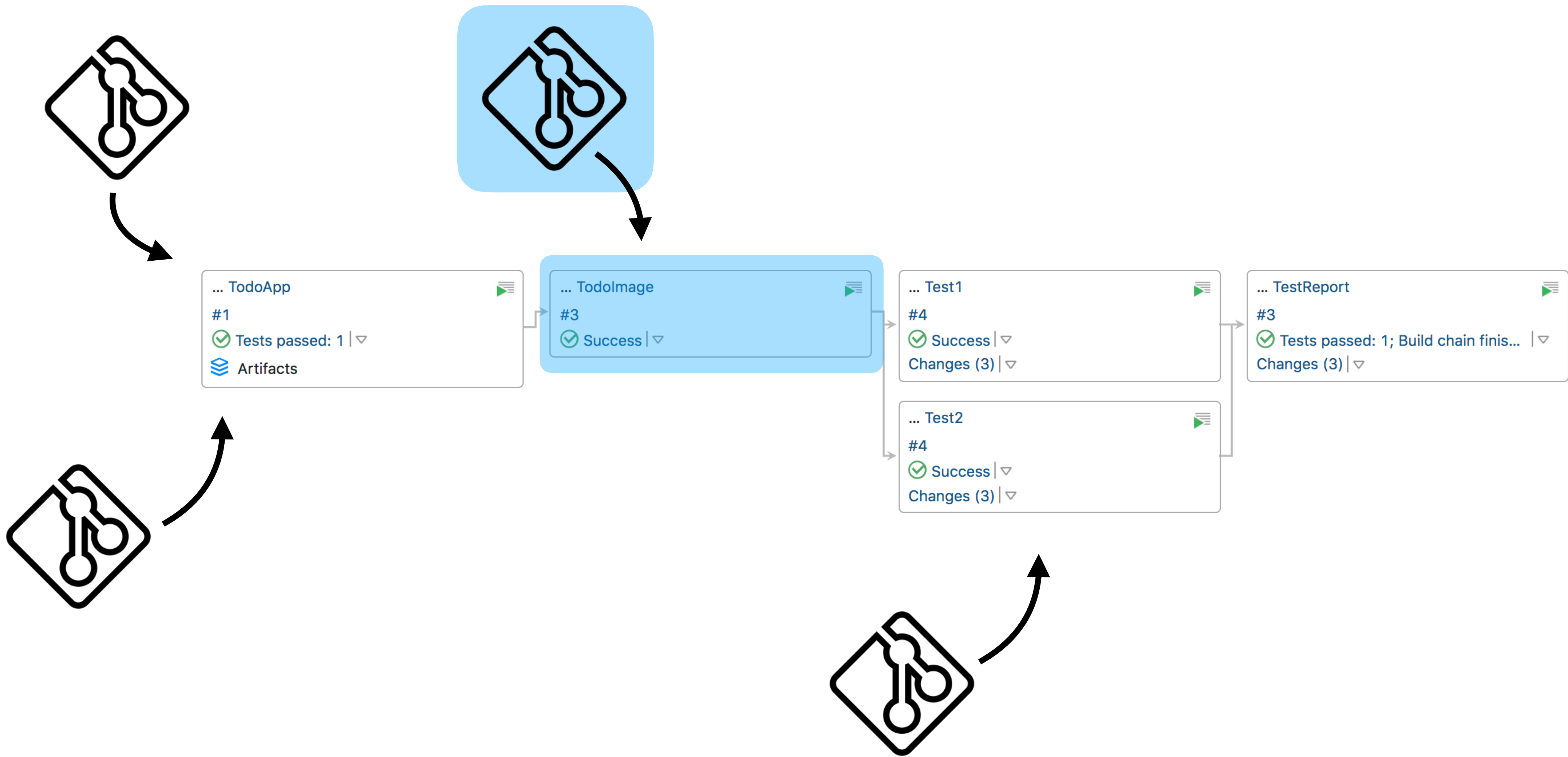
... TodoImage #3
 ✓ Success | ▾

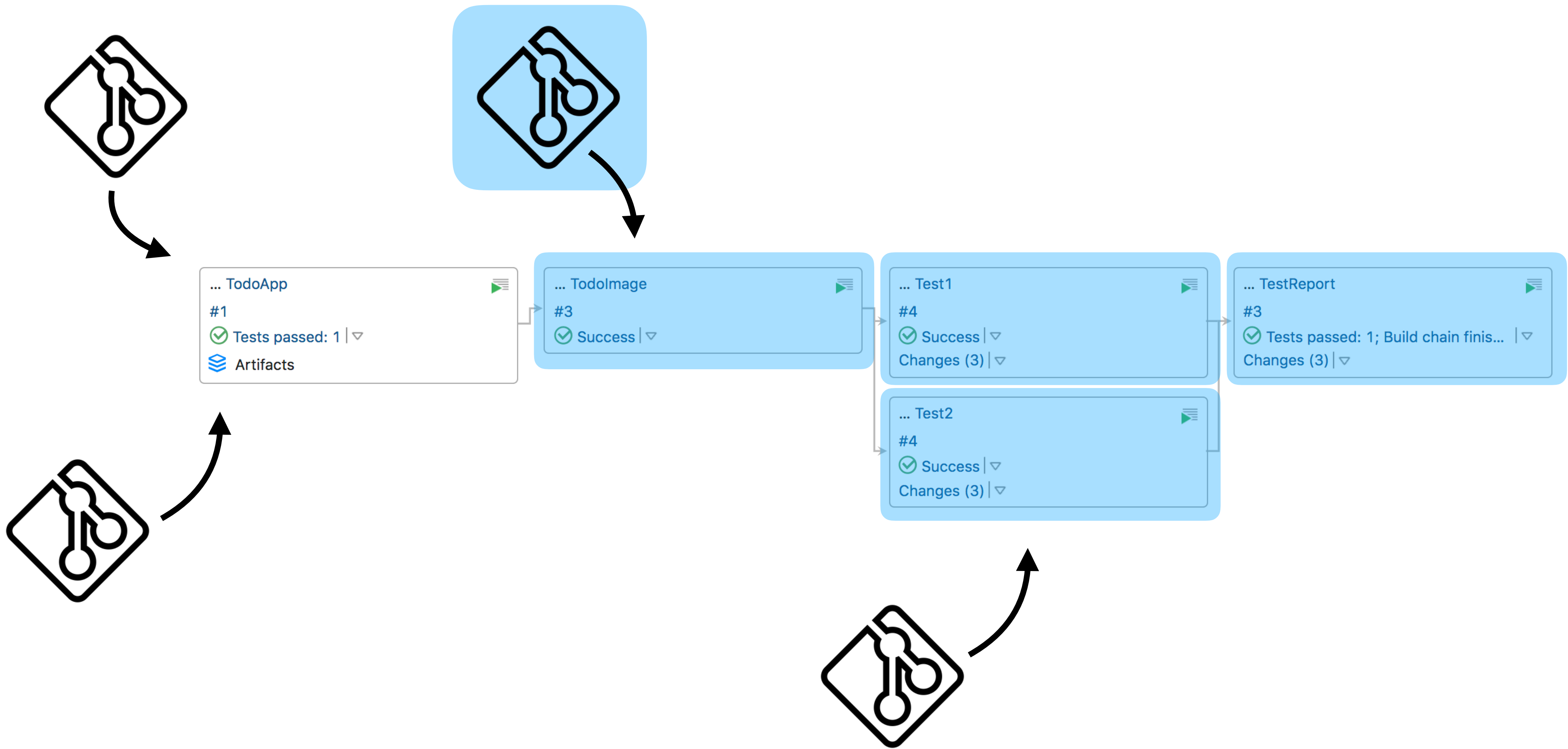
... Test1 #4
 ✓ Success | ▾
 Changes (3) | ▾

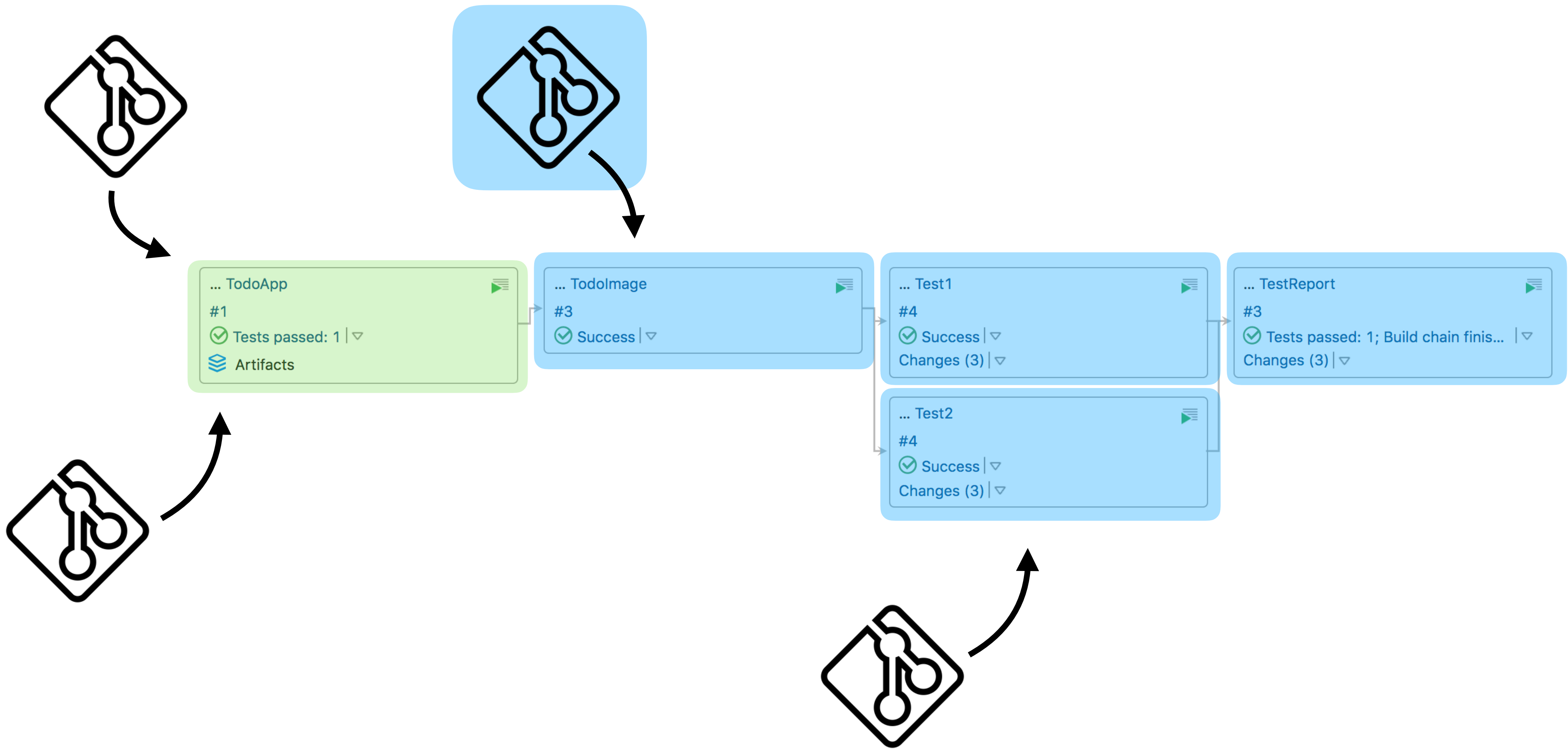
... Test2 #4
 ✓ Success | ▾
 Changes (3) | ▾

... TestReport #3
 ✓ Tests passed: 1; Build chain finis... | ▾
 Changes (3) | ▾











~33%

Optimization	Total builds optimized	Total optimized build time
Build with the same revisions already exist	13 419	2 245 h
Build with fresher changes triggered	3 494	1 368 h

Total build hours = 7 320 h



1. Переиспользование результатов сборки

Artifacts

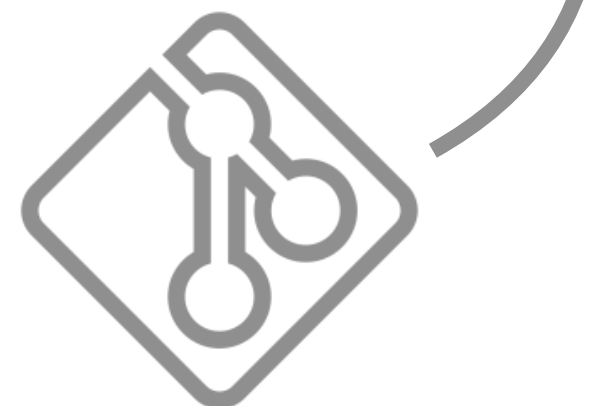
Builds

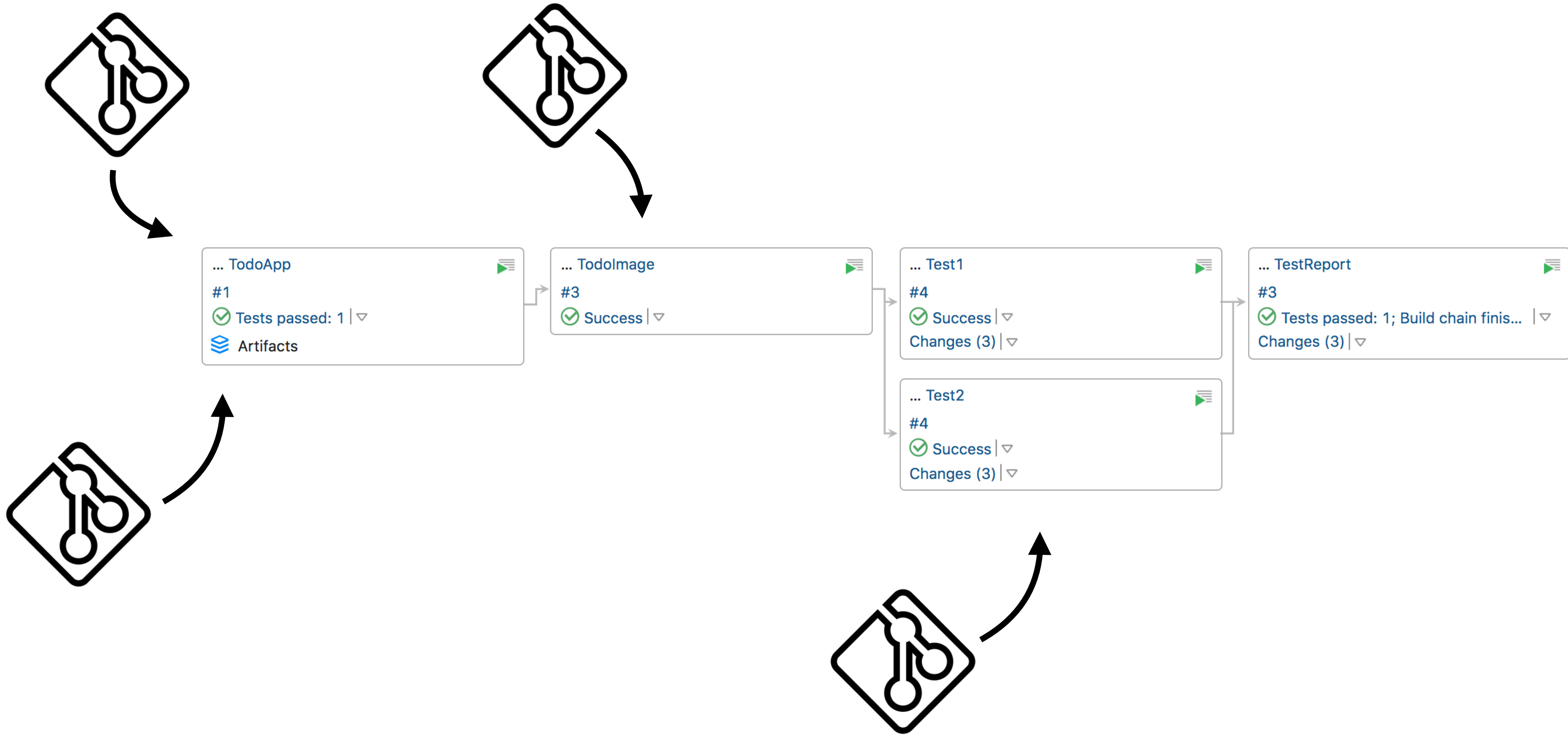
Builds

2. Синхронизация ревизий в цепочке сборок



Builds







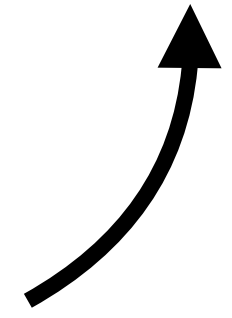
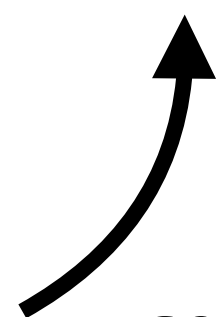
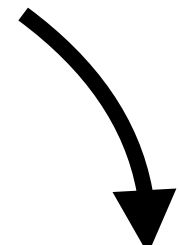
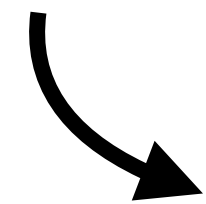
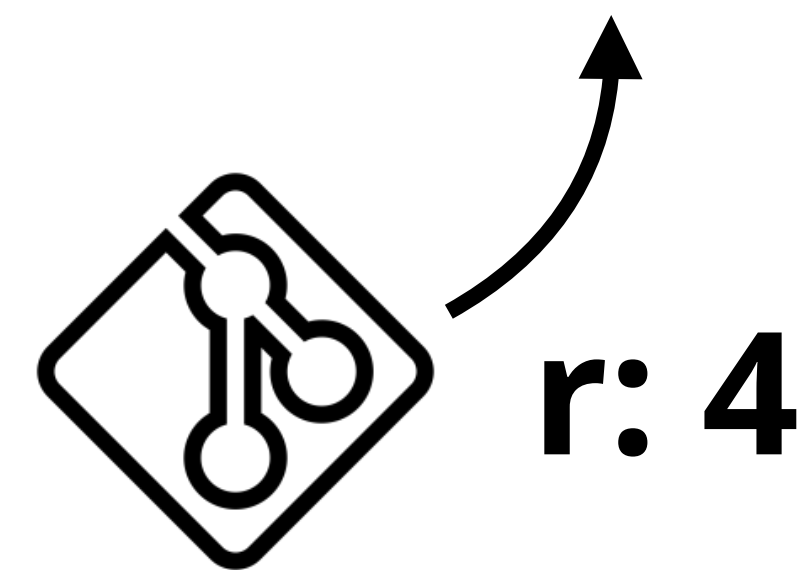
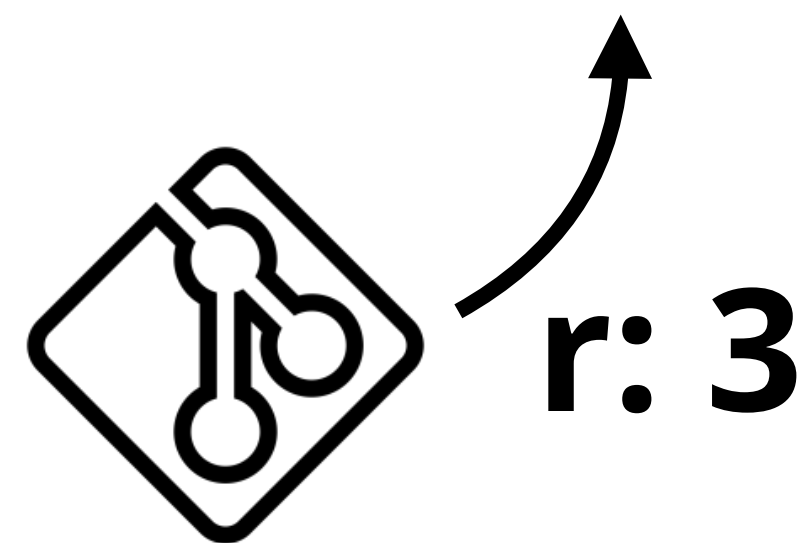
... TodoApp #1
✔ Tests passed: 1 | ▾
📦 Artifacts

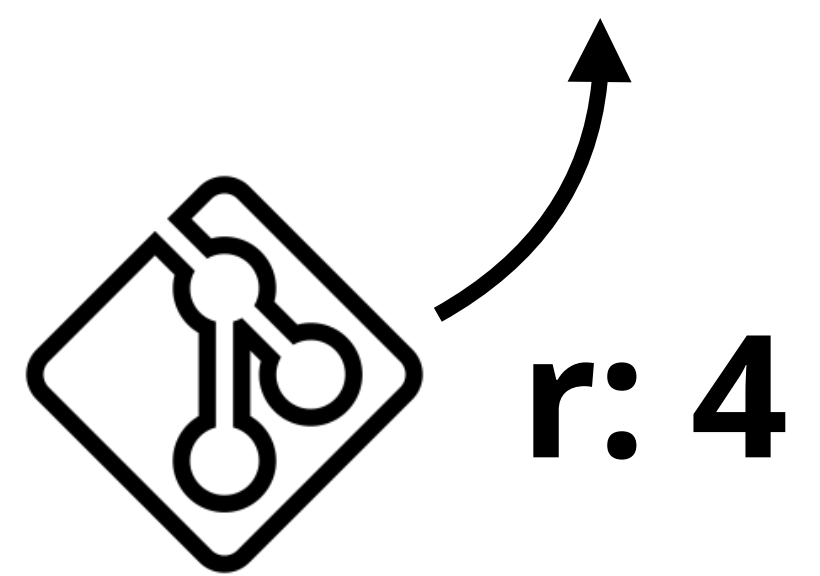
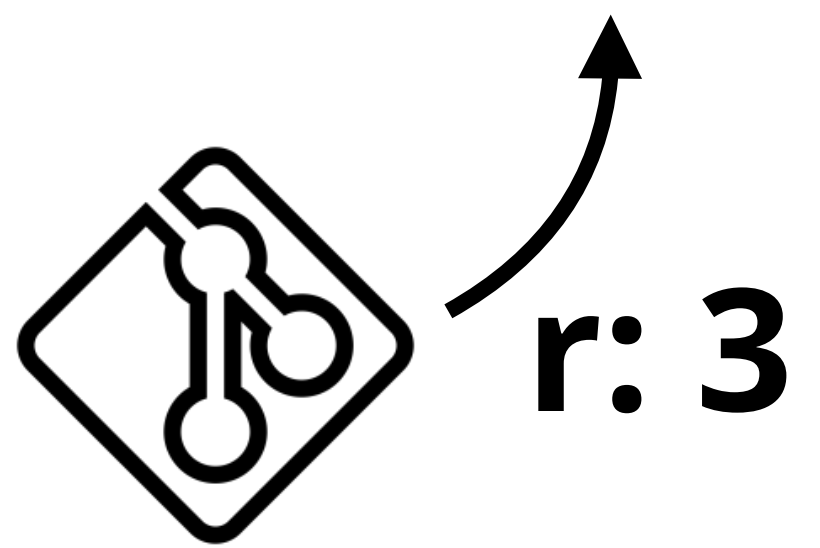
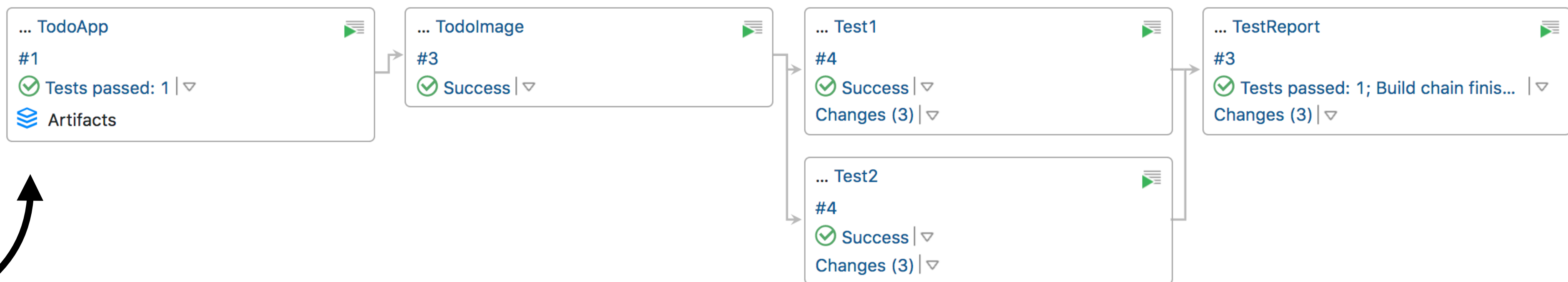
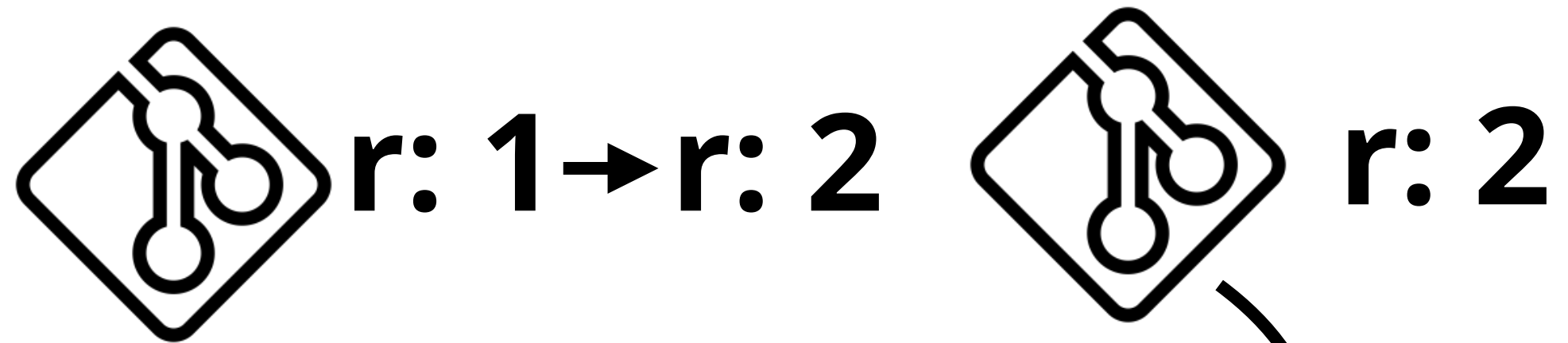
... TodoImage #3
✔ Success | ▾

... Test1 #4
✔ Success | ▾
Changes (3) | ▾

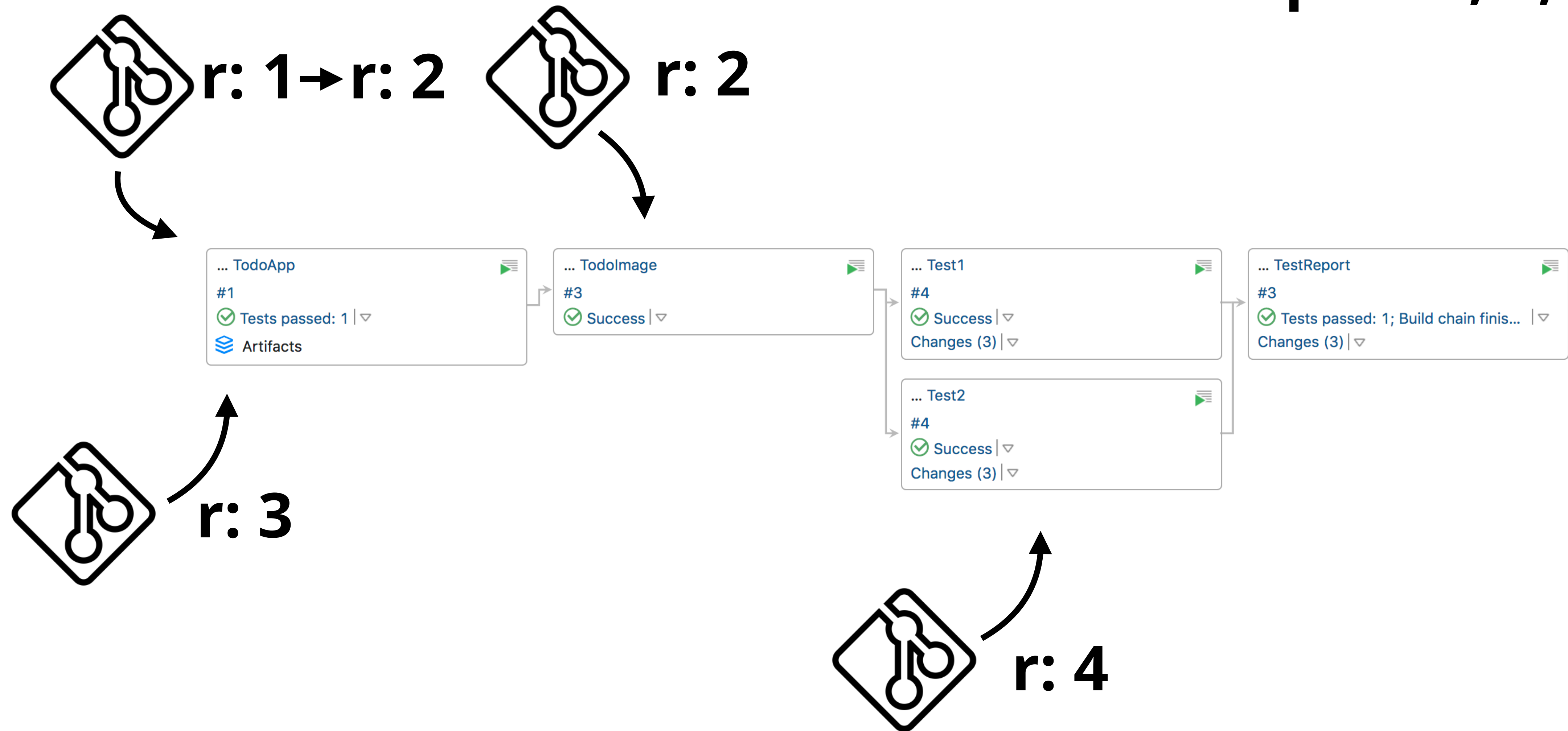
... Test2 #4
✔ Success | ▾
Changes (3) | ▾

... TestReport #3
✔ Tests passed: 1; Build chain finis... | ▾
Changes (3) | ▾

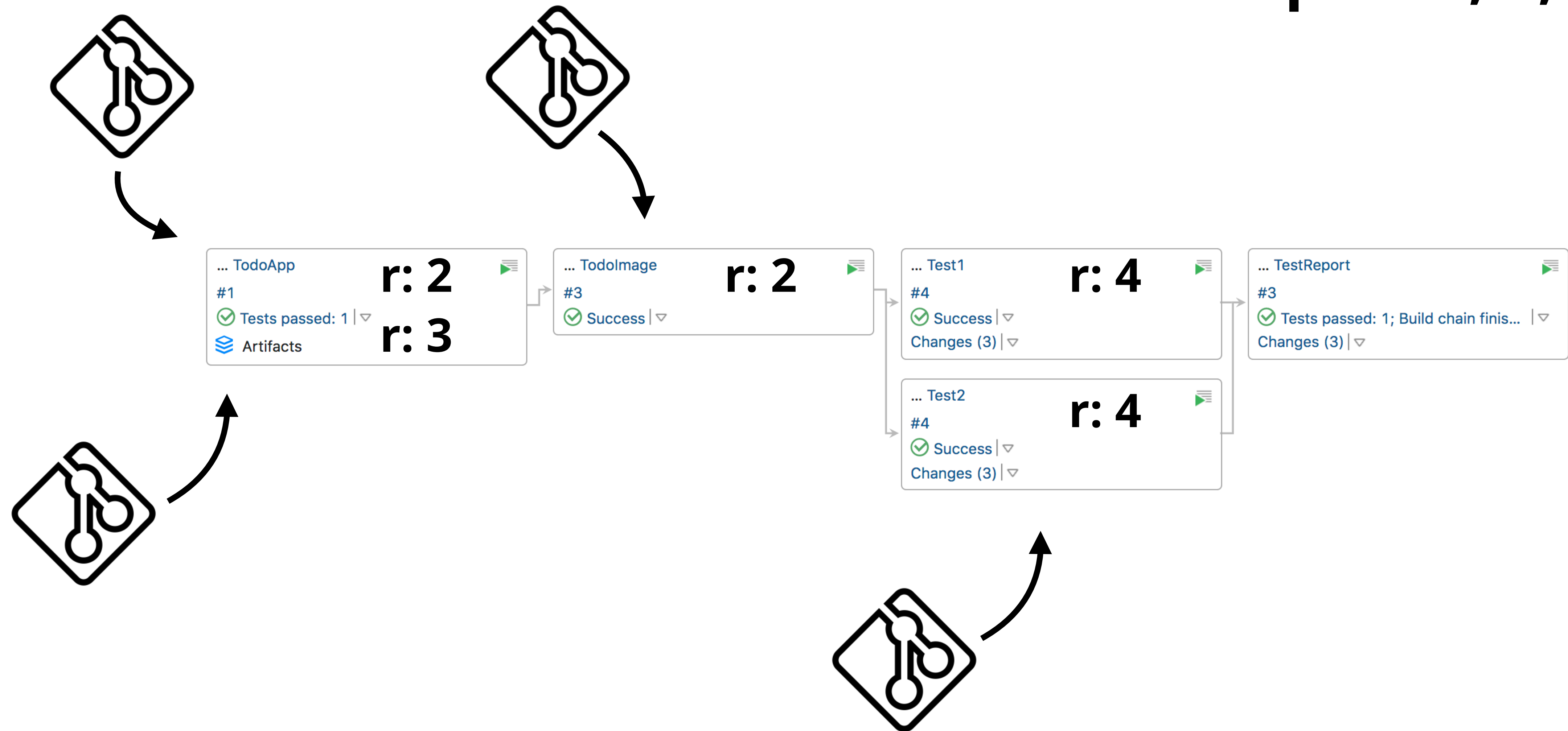




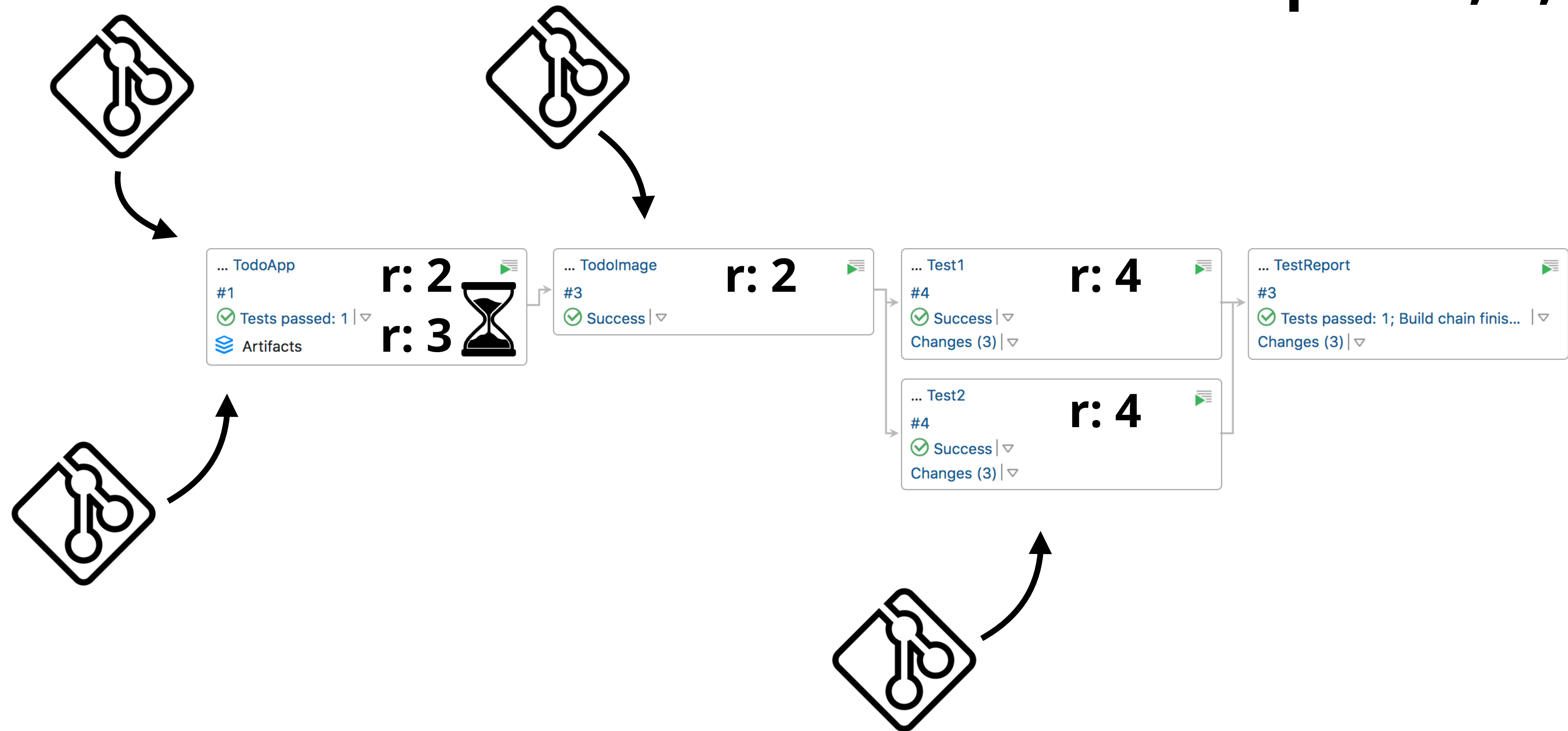
Сборка: 2, 3, 2, 4



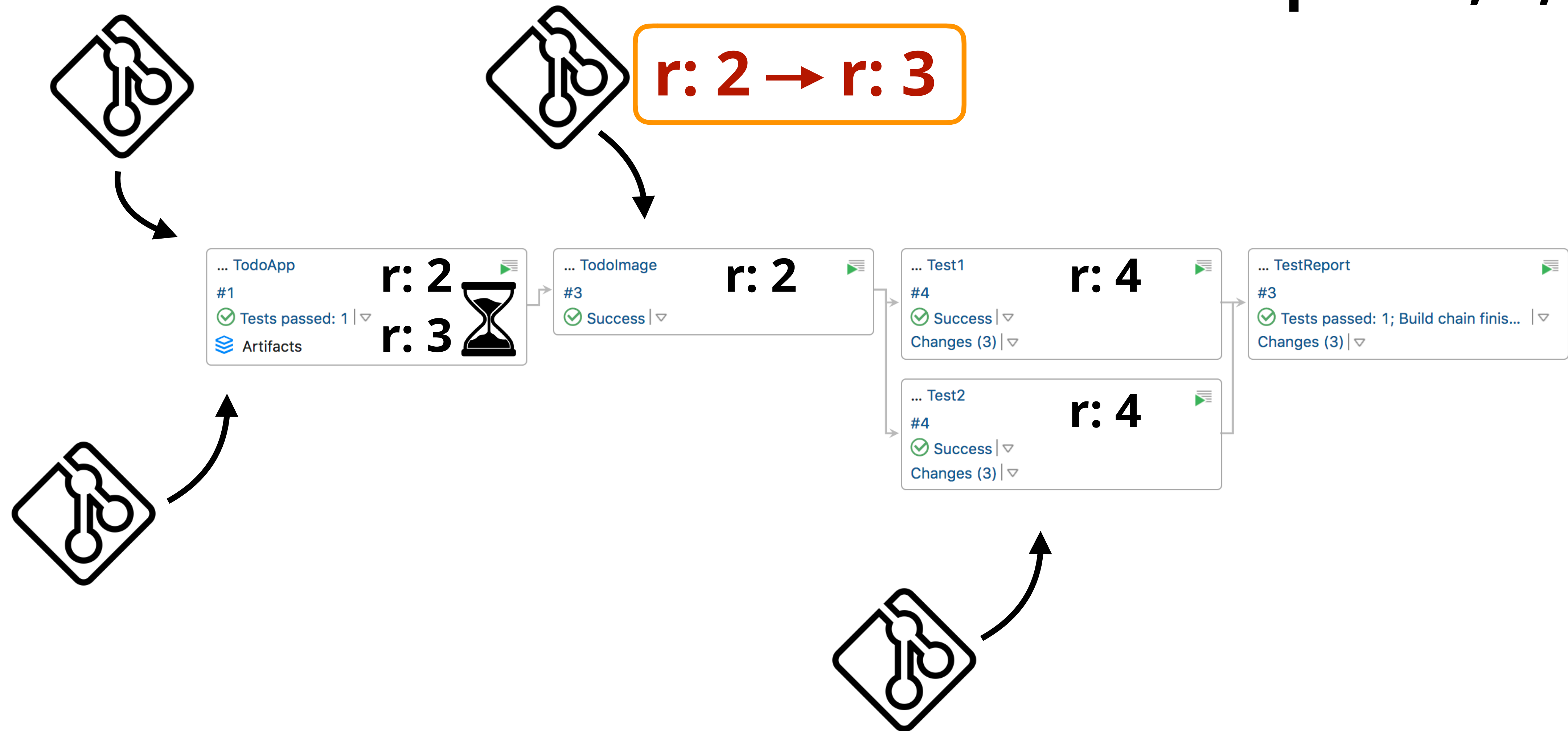
Сборка: 2, 3, 2, 4



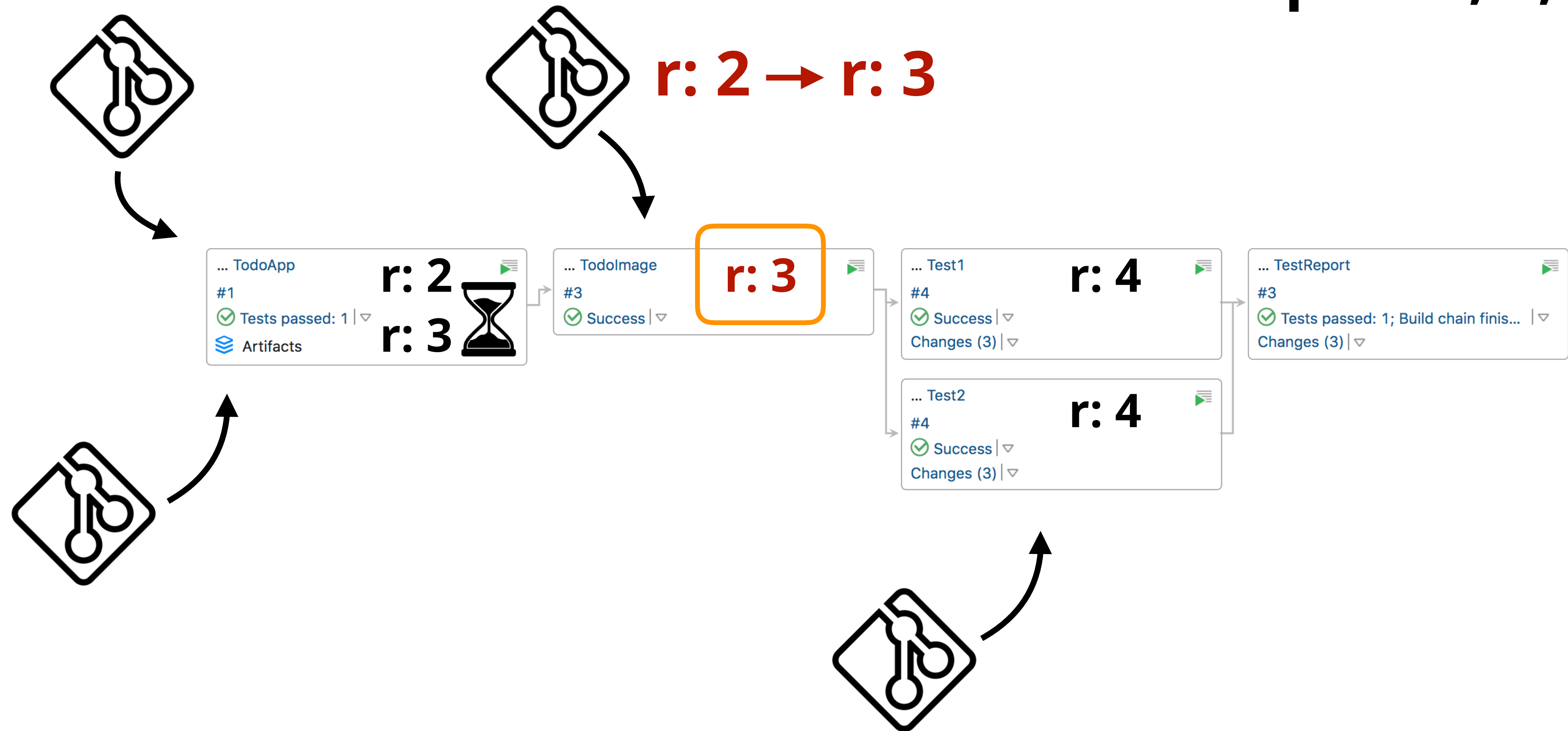
Сборка: 2, 3, 2, 4



Сборка: 2, 3, 2, 4



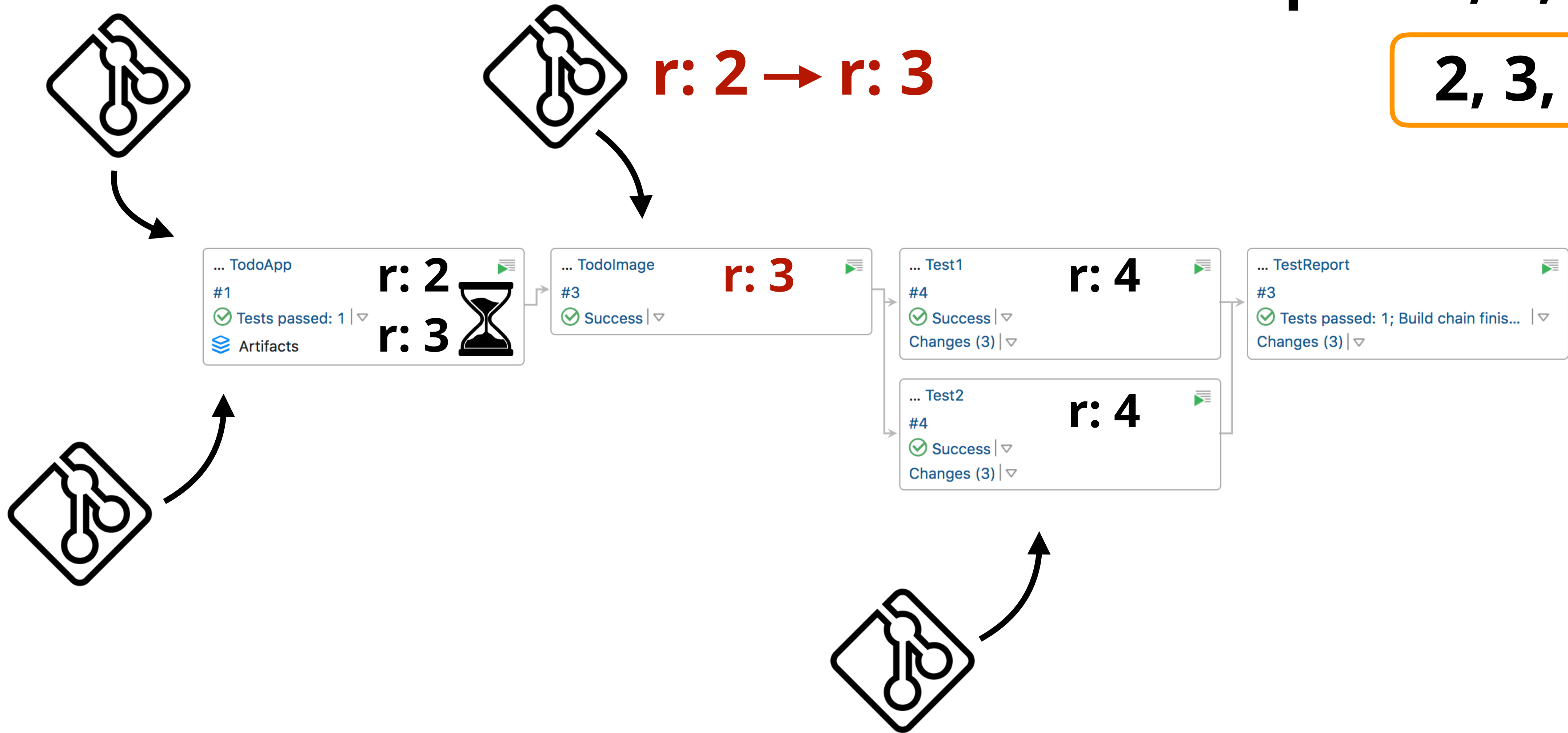
Сборка: 2, 3, 2, 4



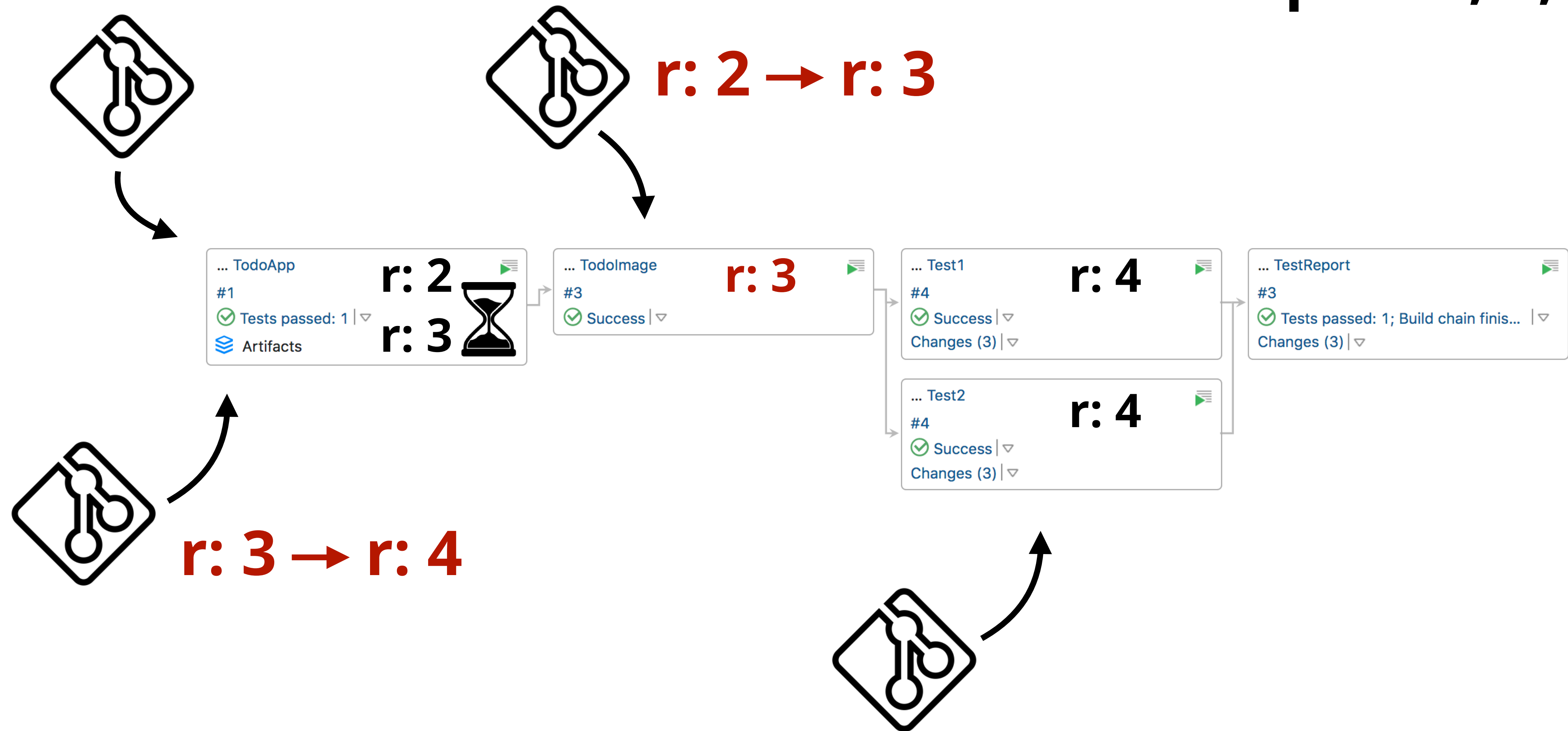
Сборка: 2, 3, 2, 4

2, 3, 3, 4

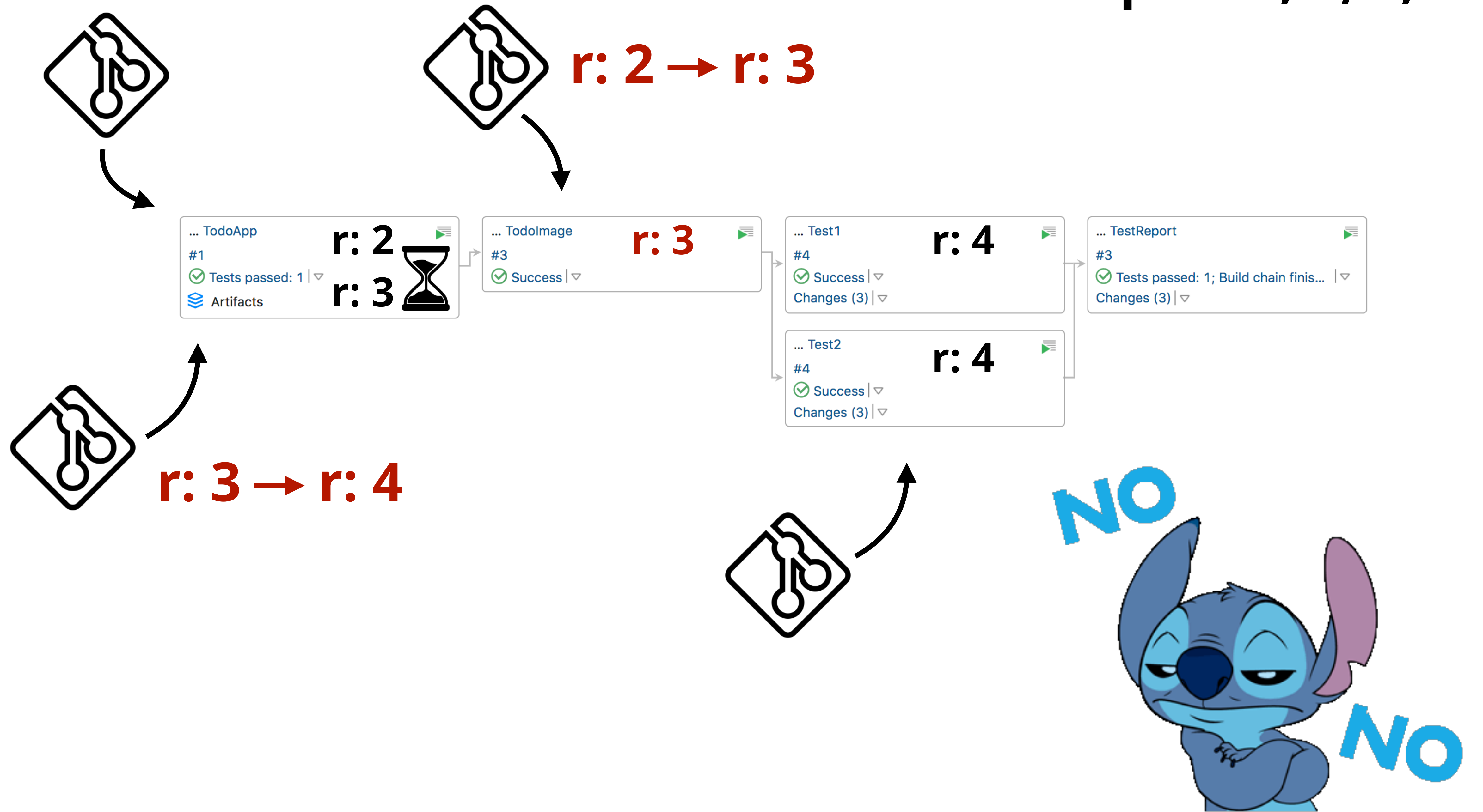
r: 2 → r: 3



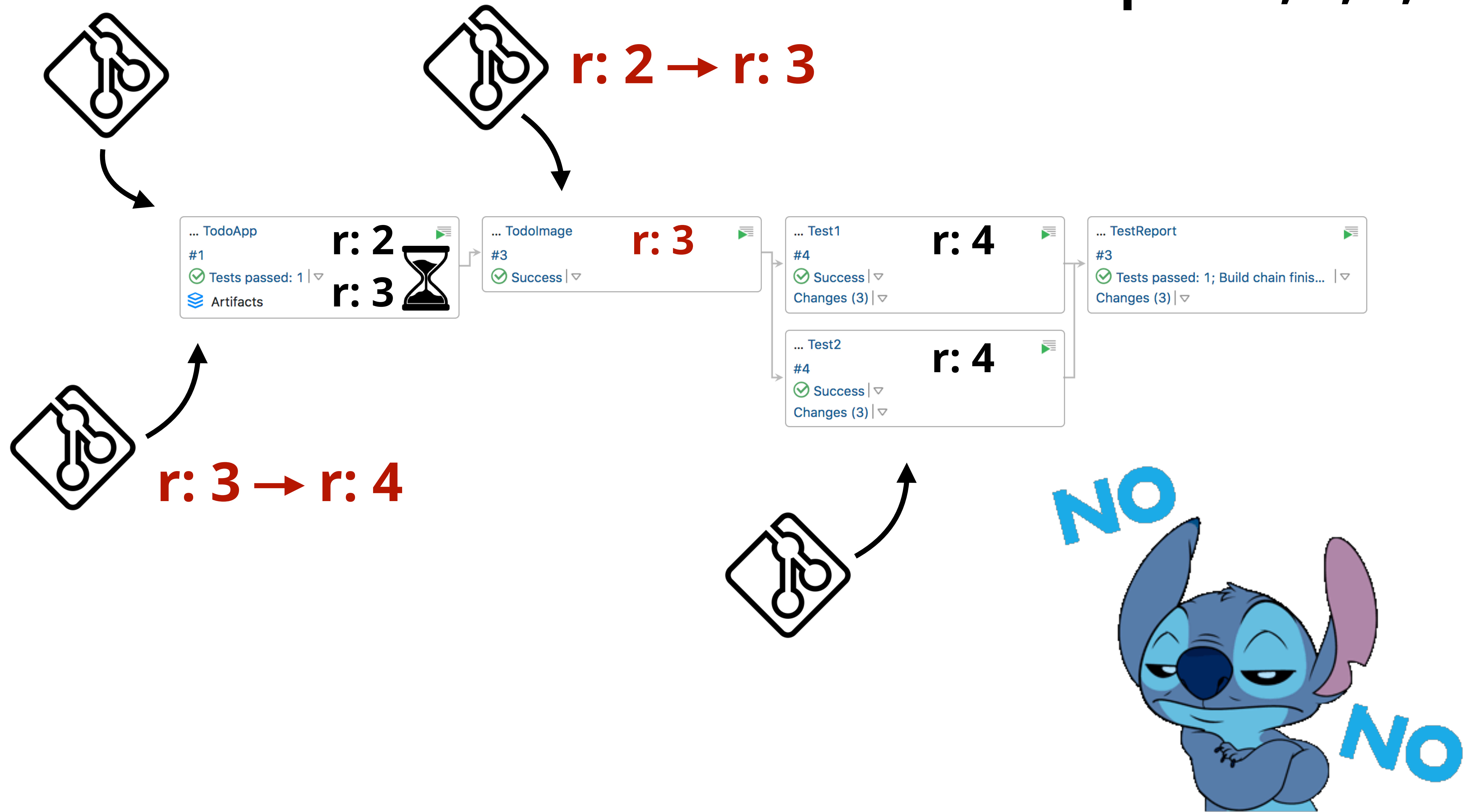
Сборка: 2, 3, 3, 4



Сборка: 2, 3, 3, 4



Сборка: 2, 3, 3, 4





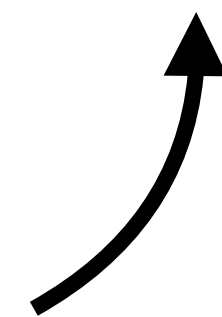
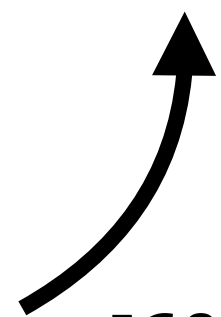
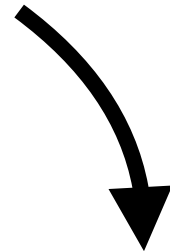
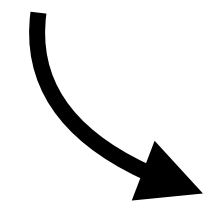
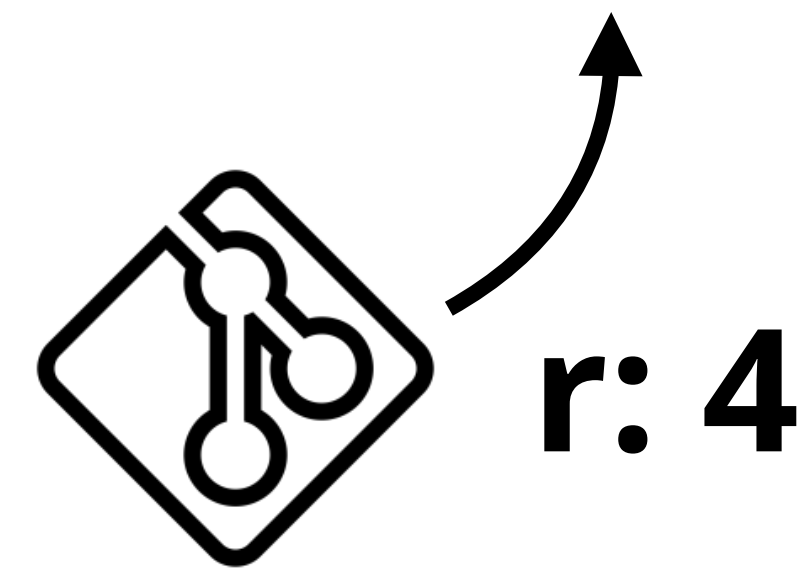
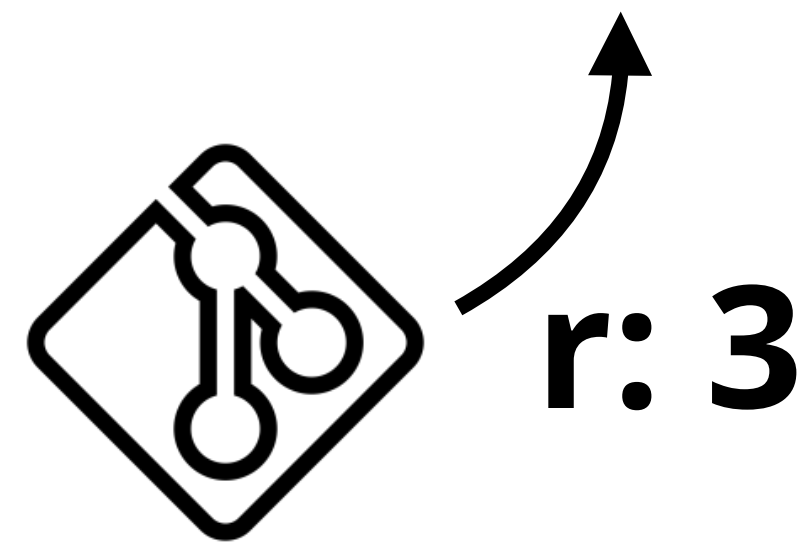
... TodoApp #1
✔ Tests passed: 1 | ▾
📦 Artifacts

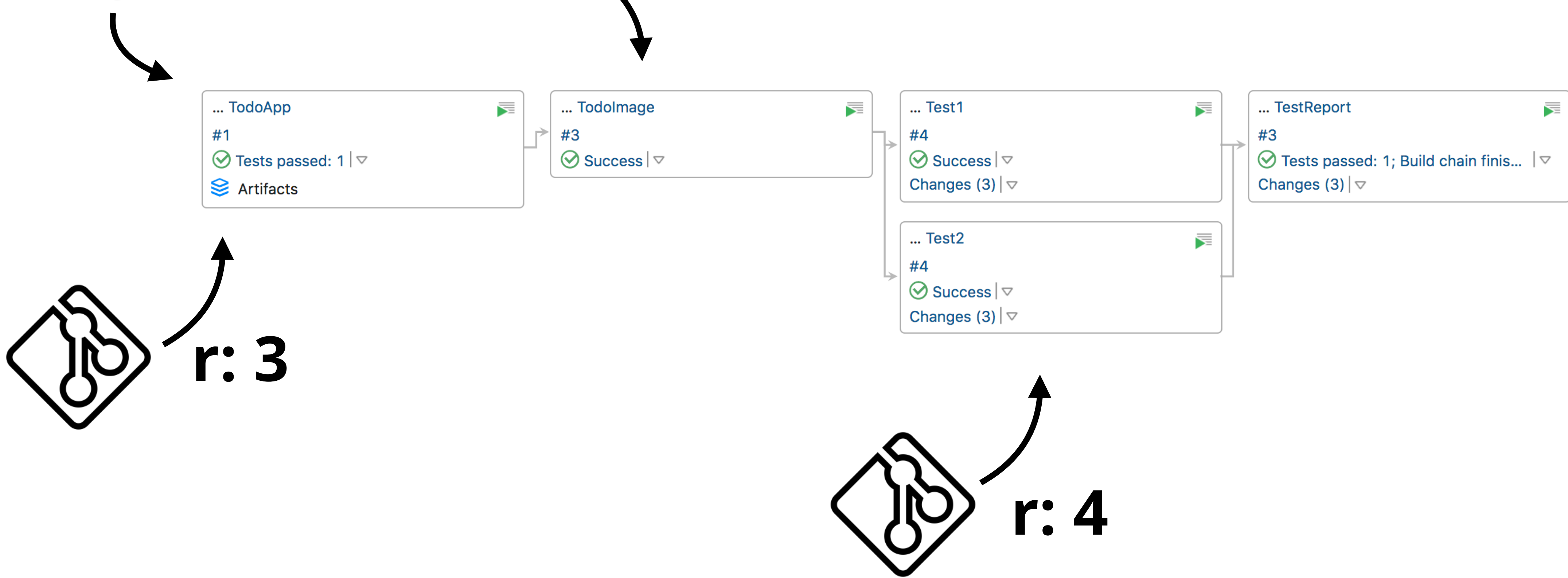
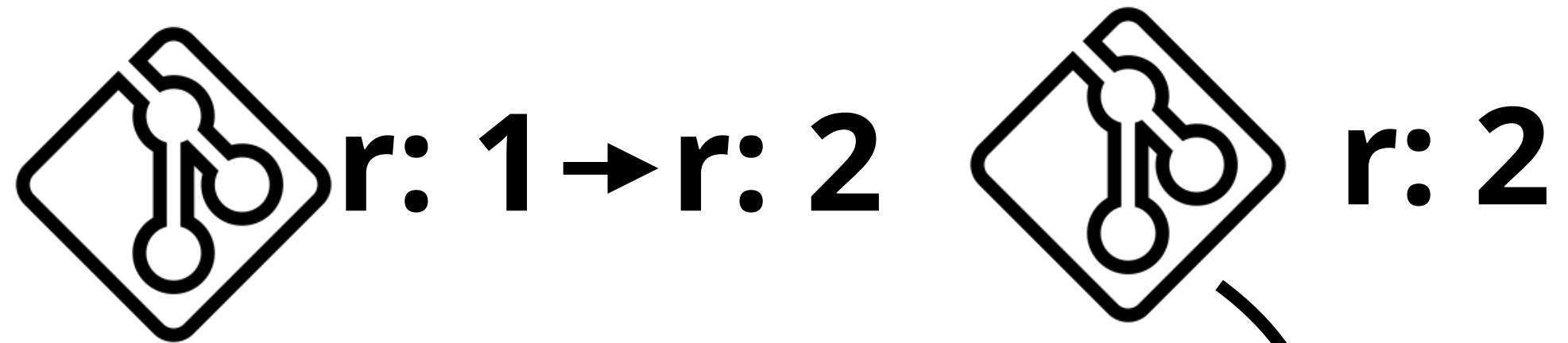
... TodoImage #3
✔ Success | ▾

... Test1 #4
✔ Success | ▾
Changes (3) | ▾

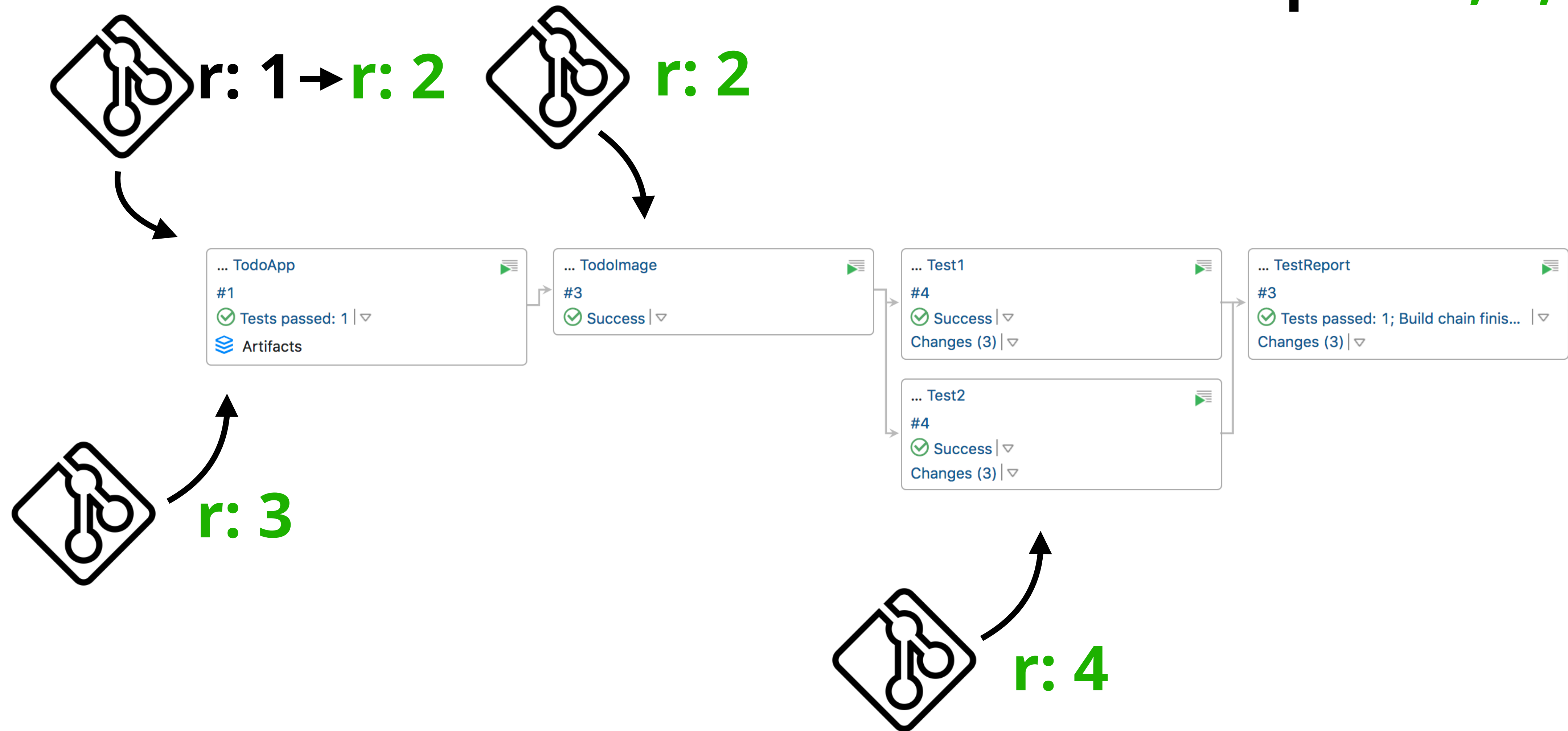
... Test2 #4
✔ Success | ▾
Changes (3) | ▾

... TestReport #3
✔ Tests passed: 1; Build chain finis... | ▾
Changes (3) | ▾

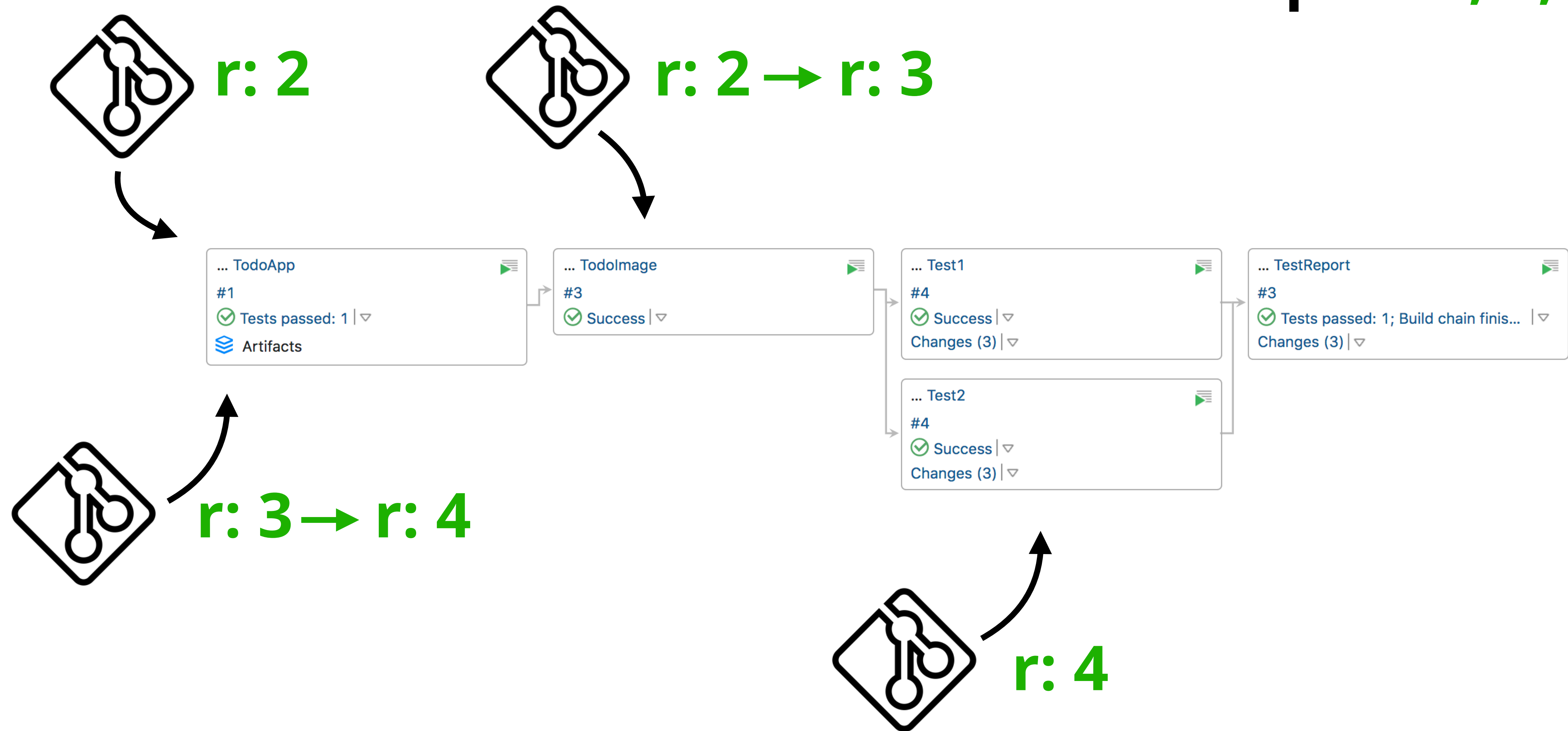




Сборка: 2, 3, 2, 4

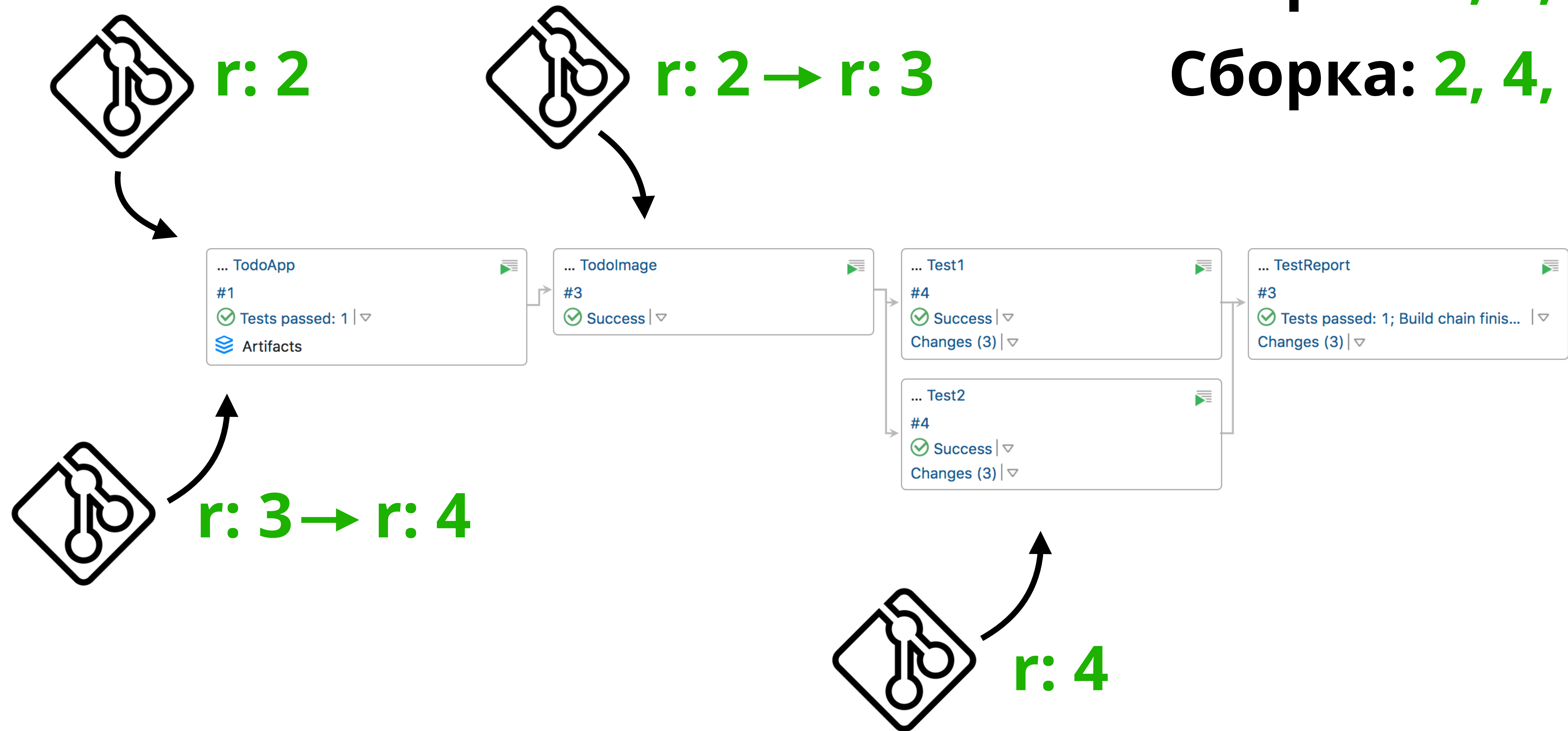


Сборка: 2, 3, 2, 4



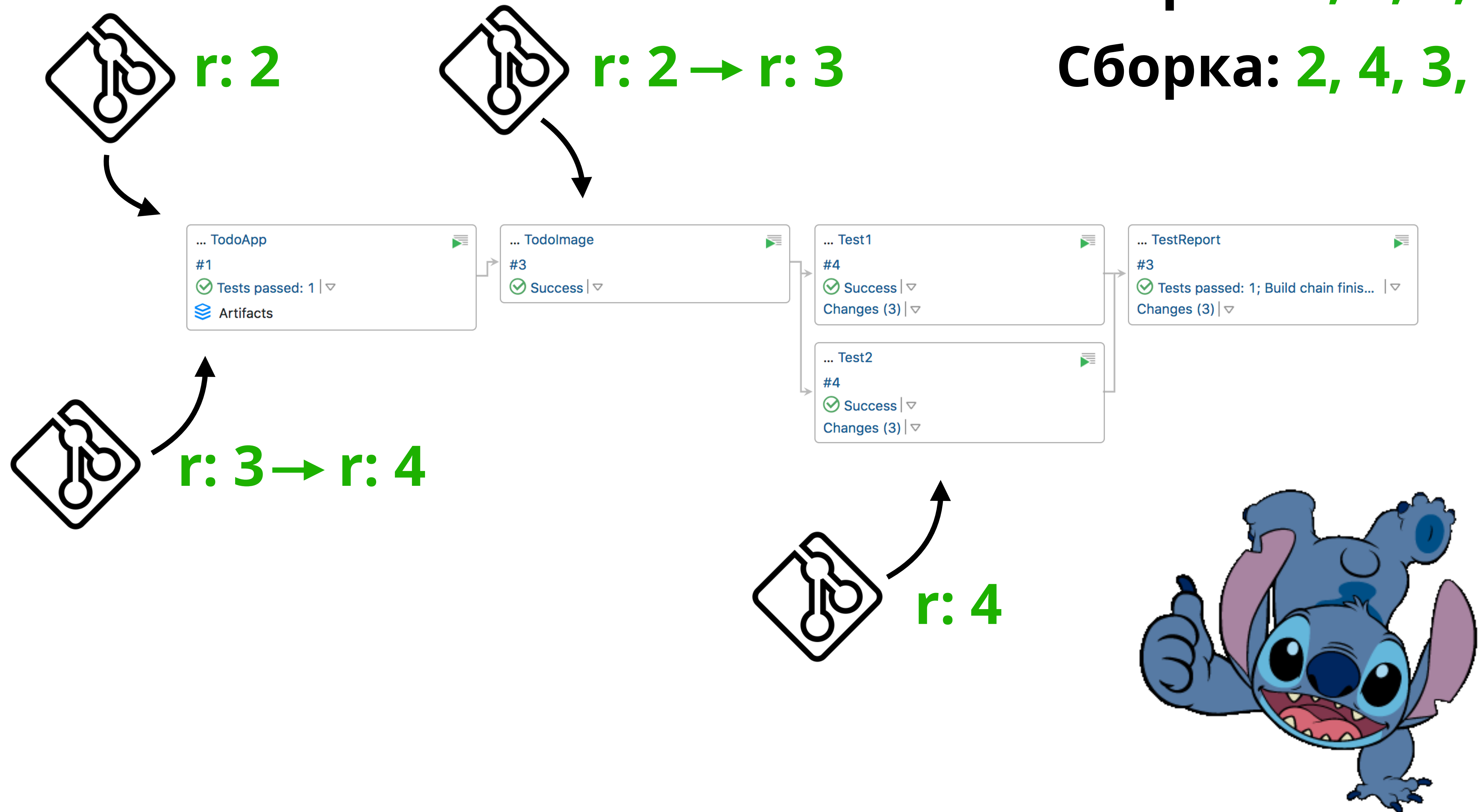
Сборка: 2, 3, 2, 4

Сборка: 2, 4, 3, 4



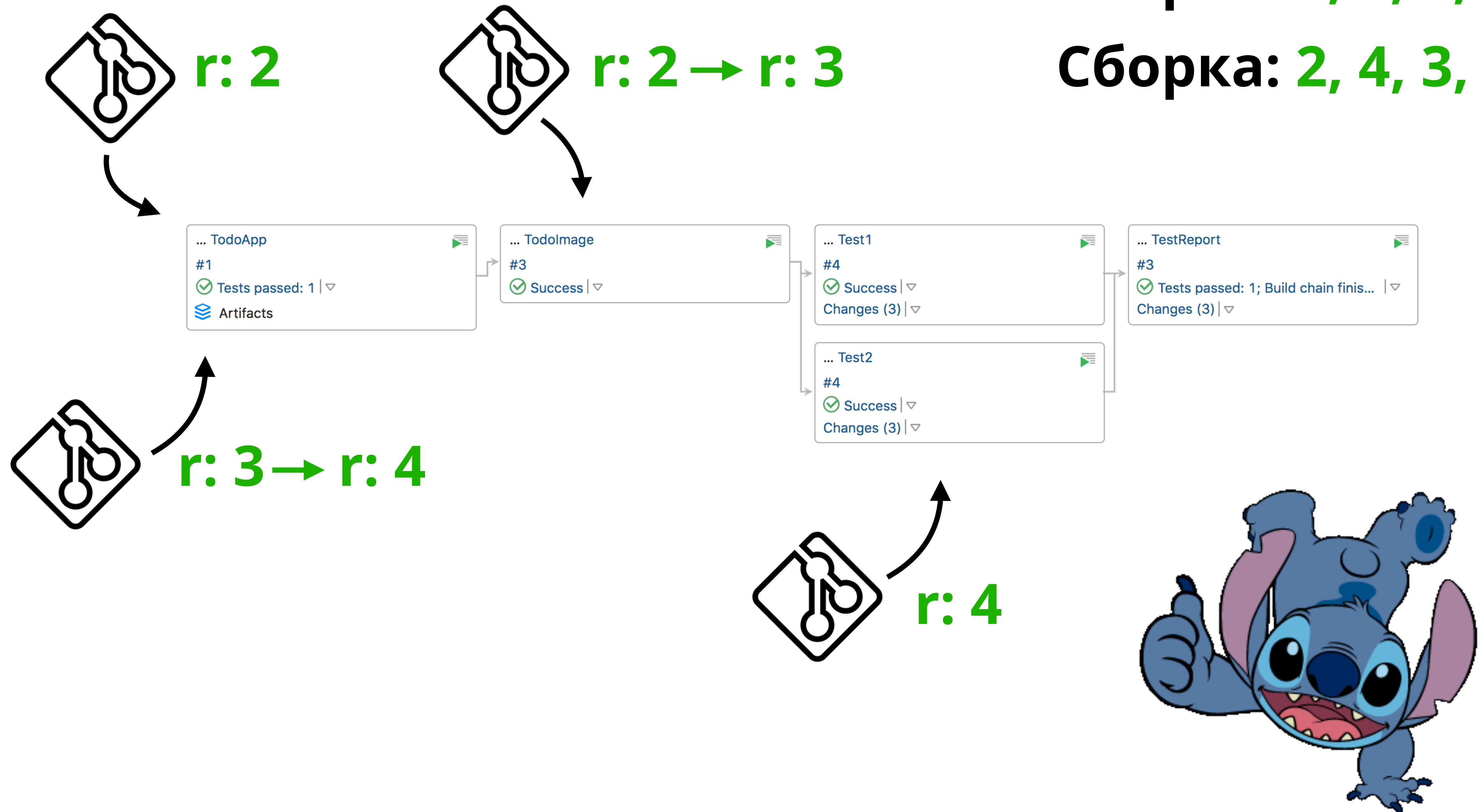
Сборка: 2, 3, 2, 4

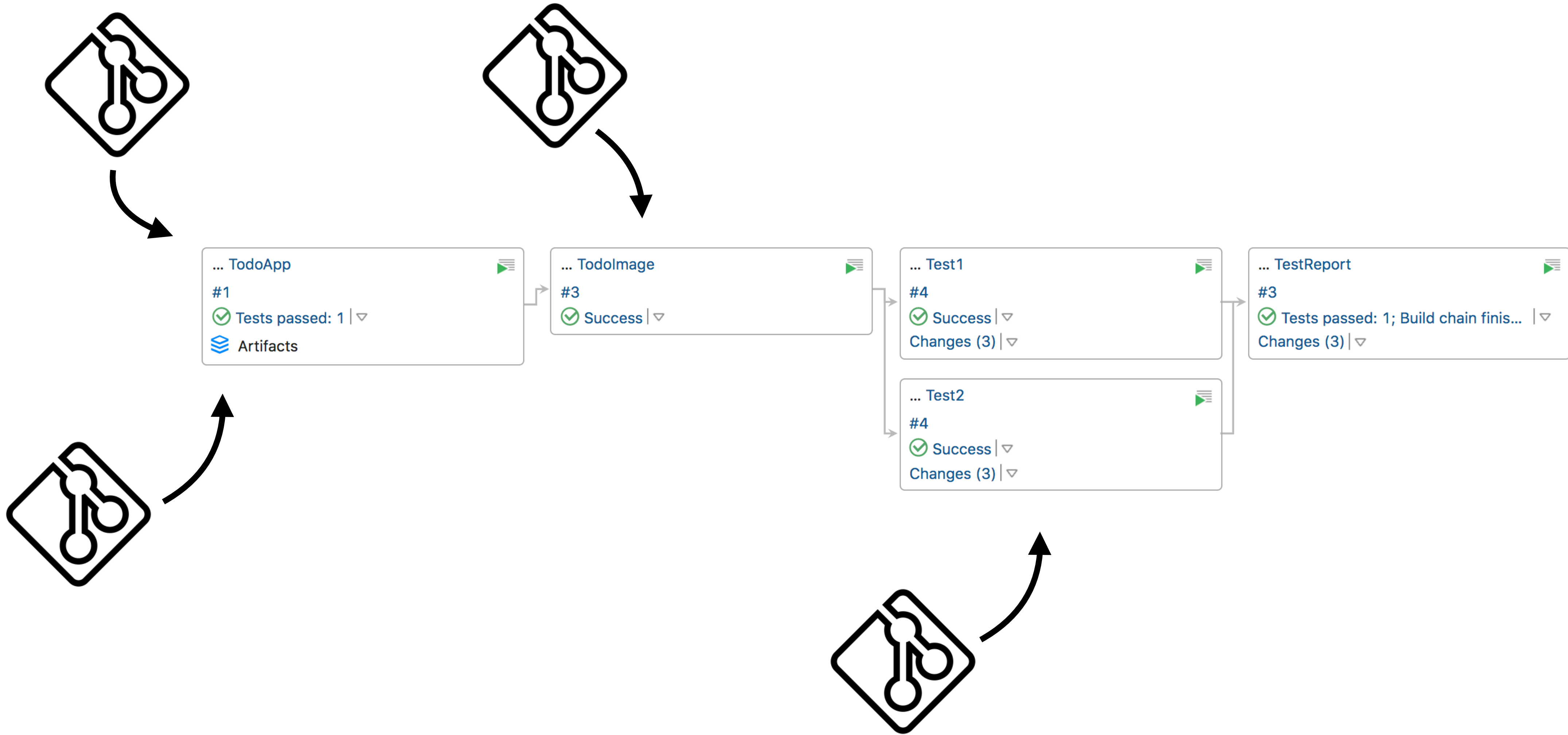
Сборка: 2, 4, 3, 4

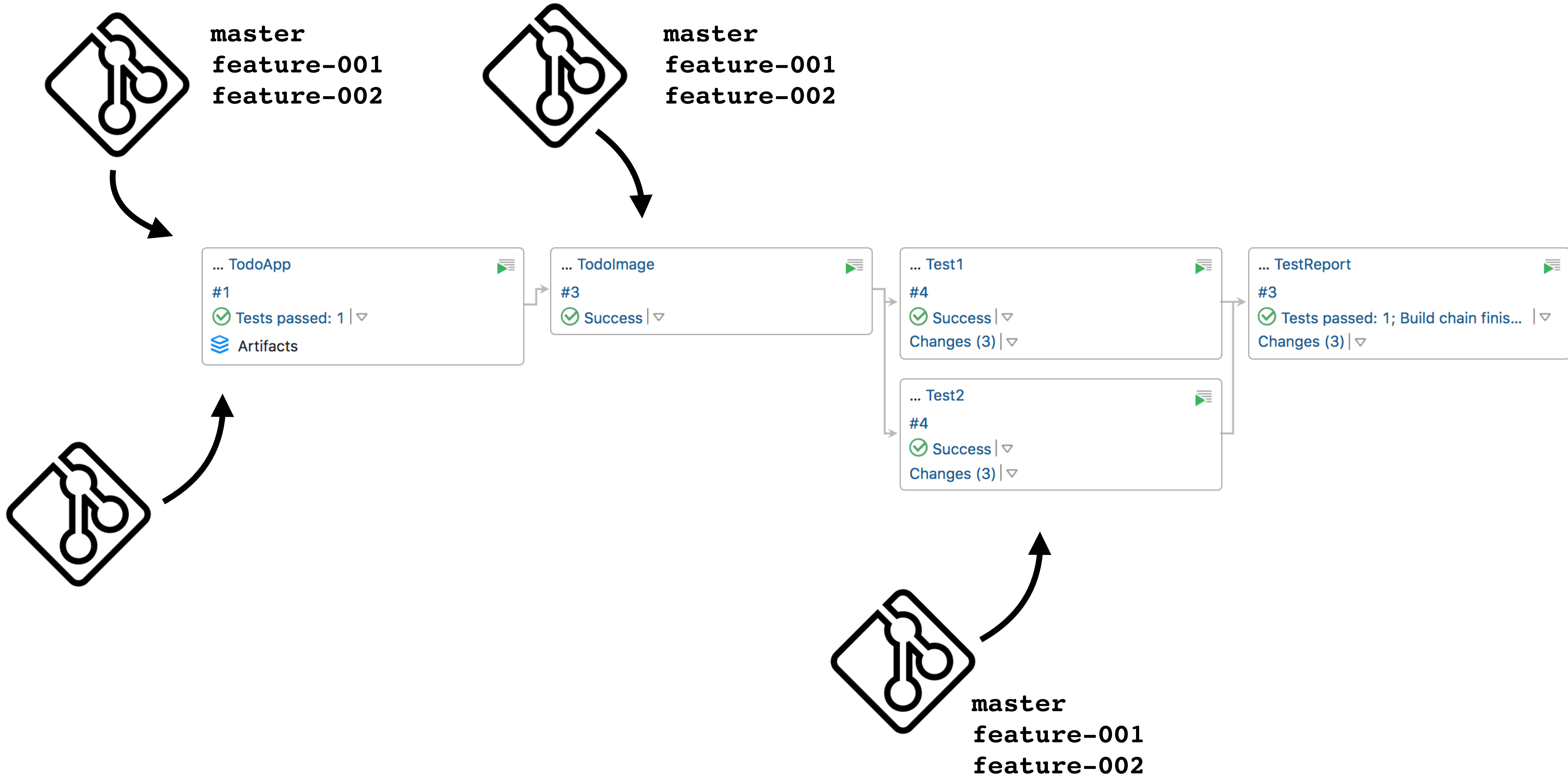


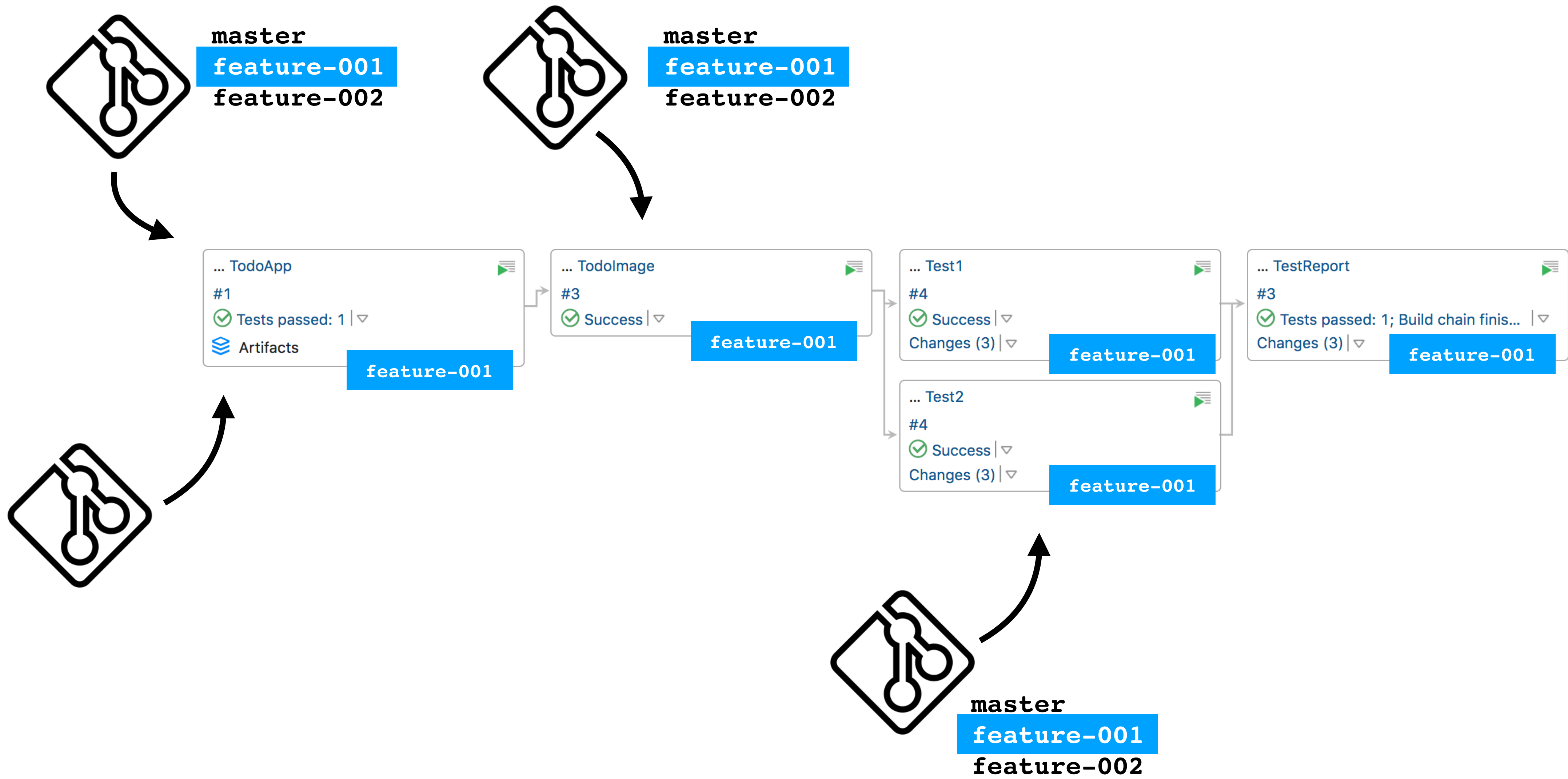
Сборка: 2, 3, 2, 4

Сборка: 2, 4, 3, 4



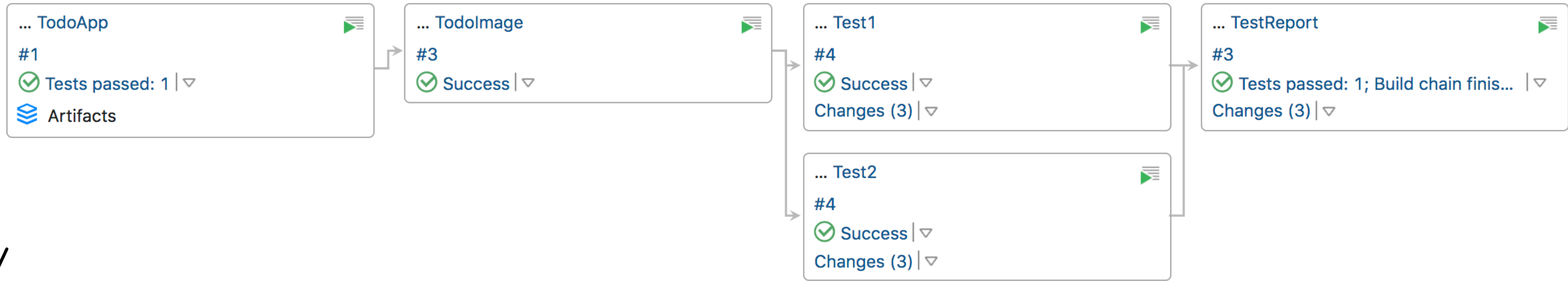






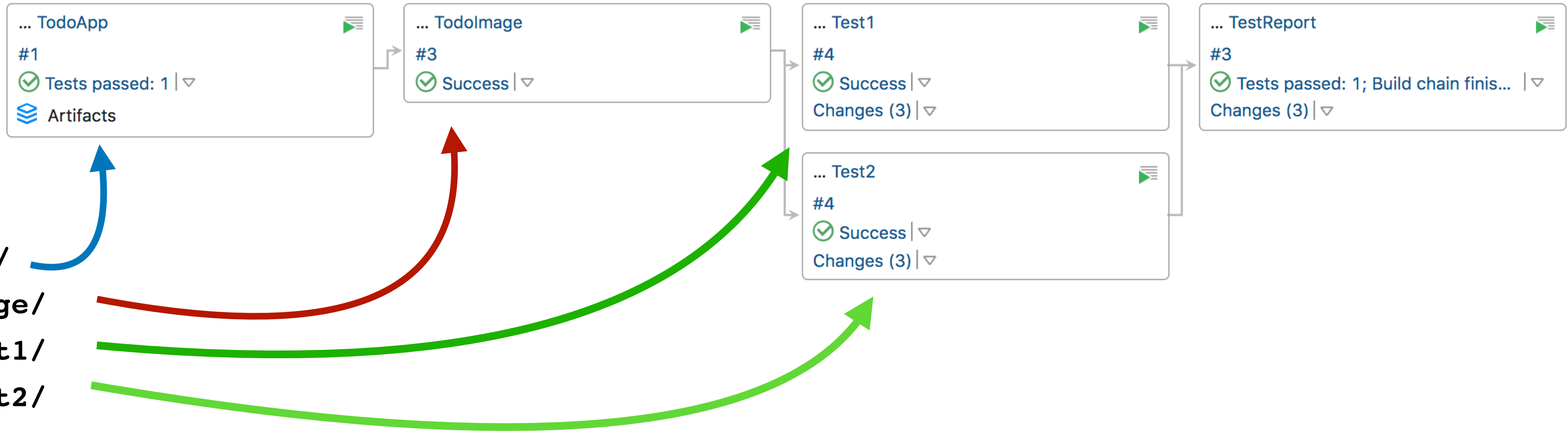


app/
image/
test1/
test2/





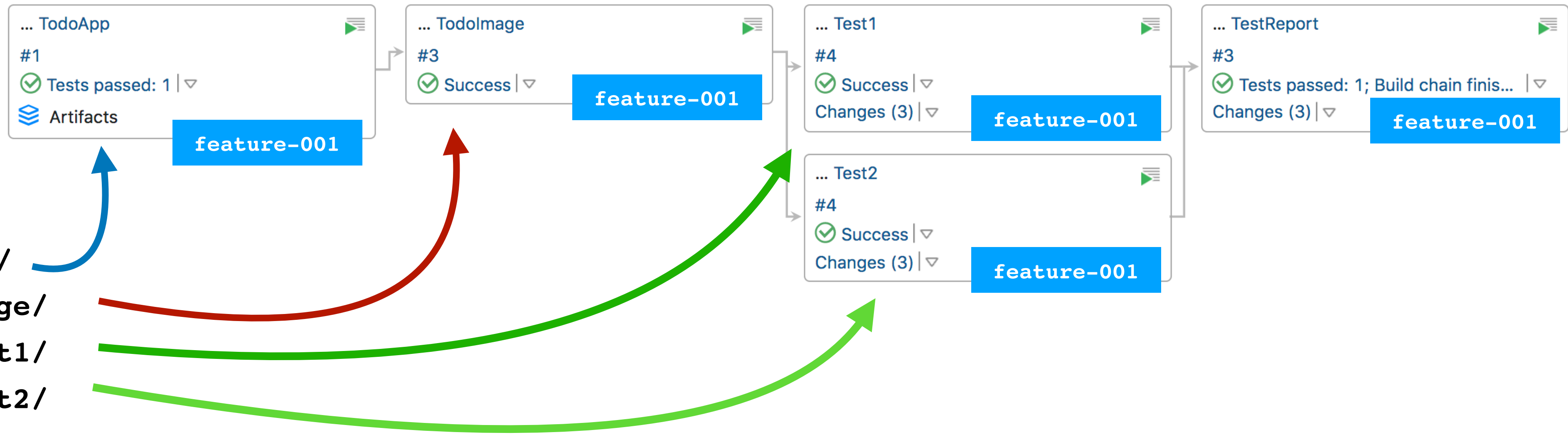
app/
image/
test1/
test2/

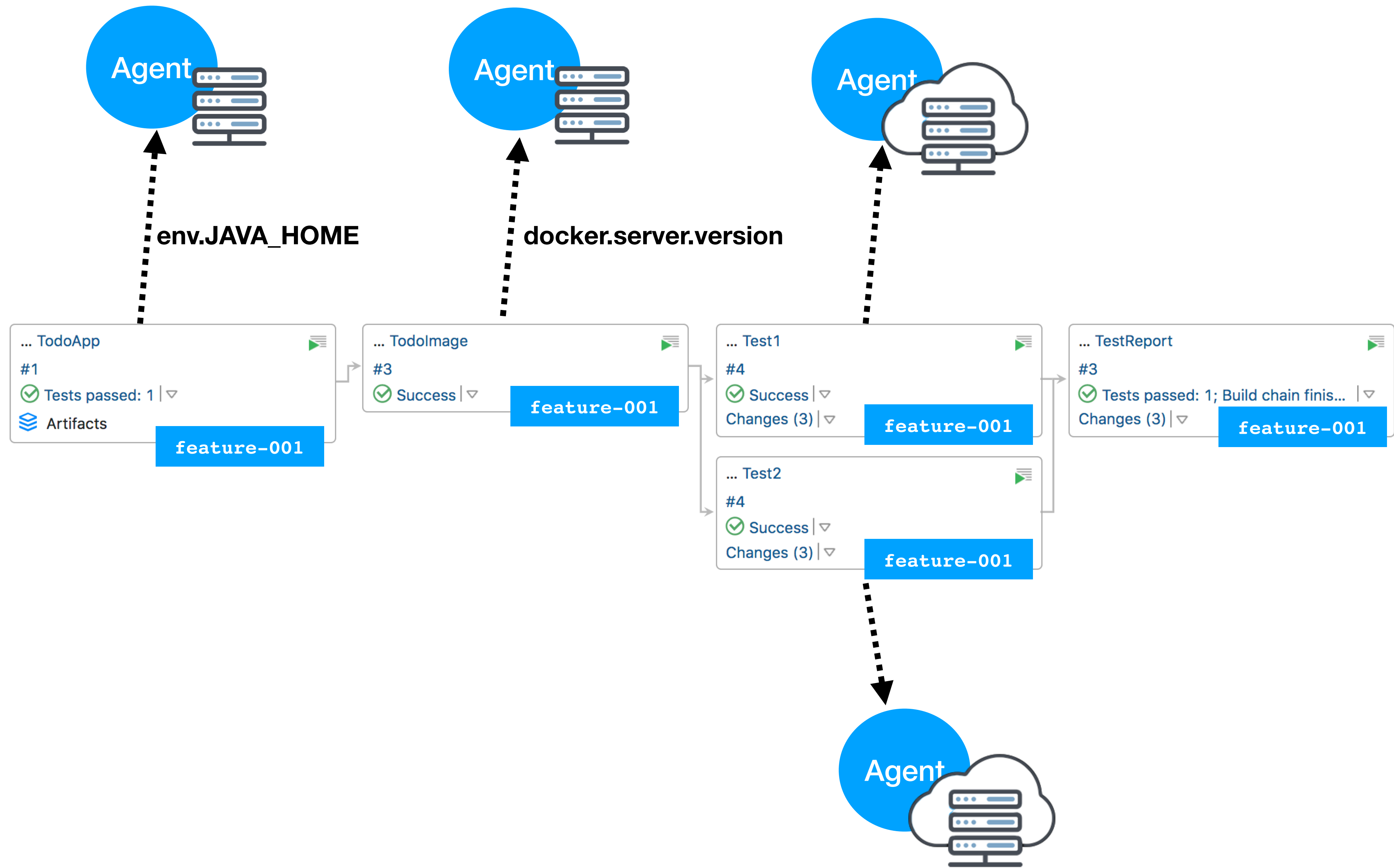




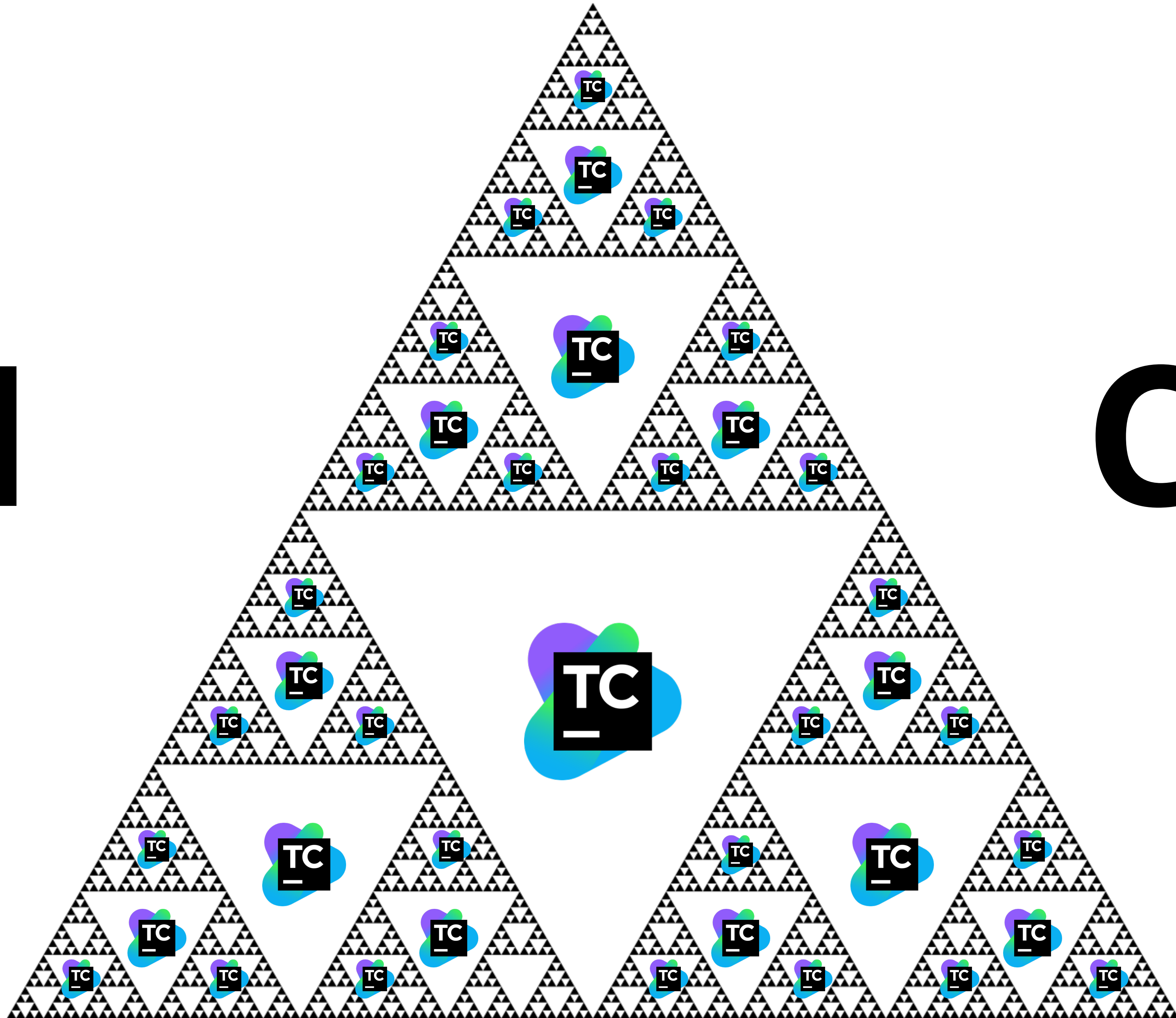
app/
image/
test1/
test2/

master
feature-001
feature-002





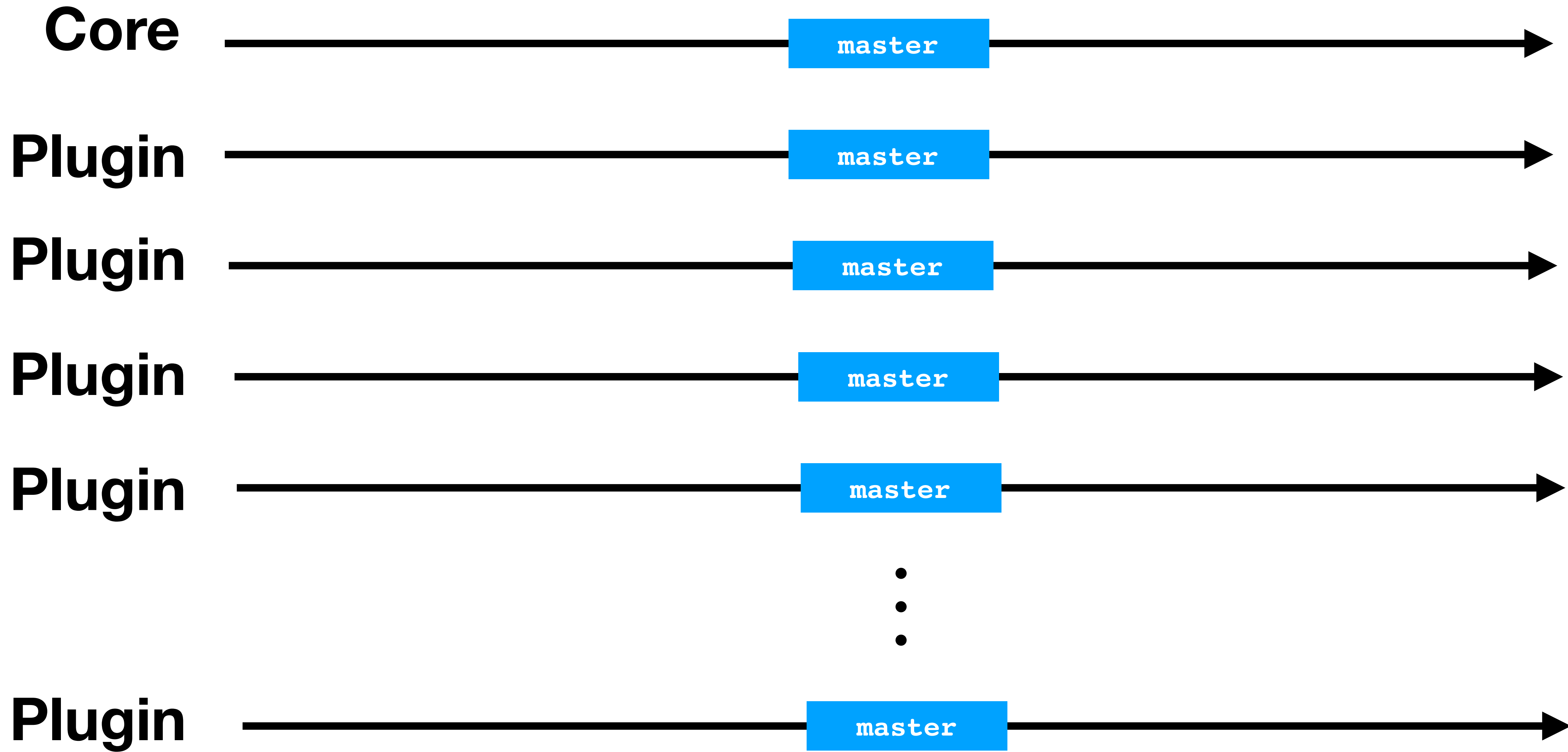
CI



CD

Core





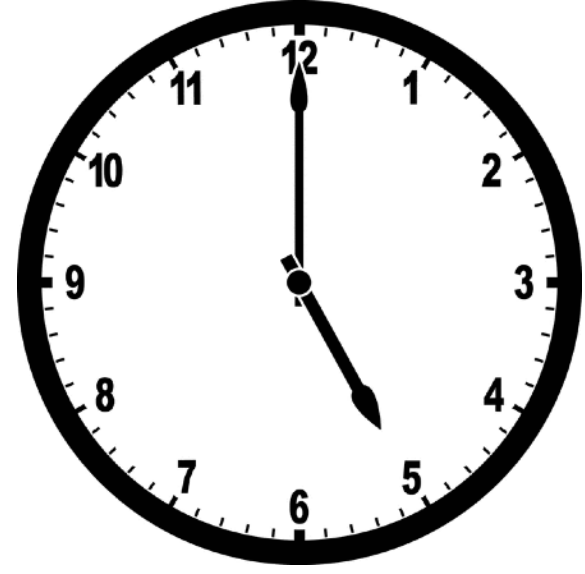
master



master

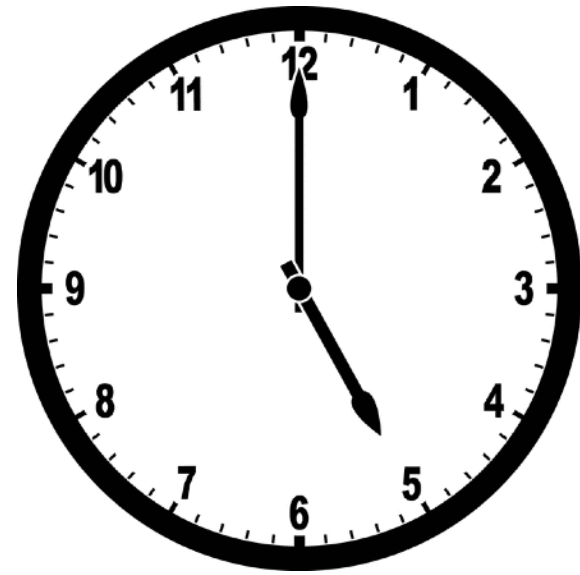


master

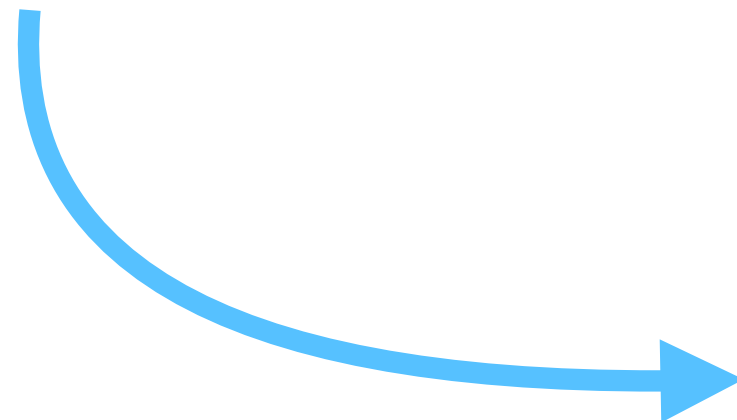


“Review build”

master



“Review build”



🟢 #71030 ☆

Success

👤 master 🏠 Details ▾ ⚙️ Actions ▾

Overview Tests **Changes** Log Dependencies More ▾

> Repositories

21 Changes Include changes in settings [Filter changes by user ▾](#)

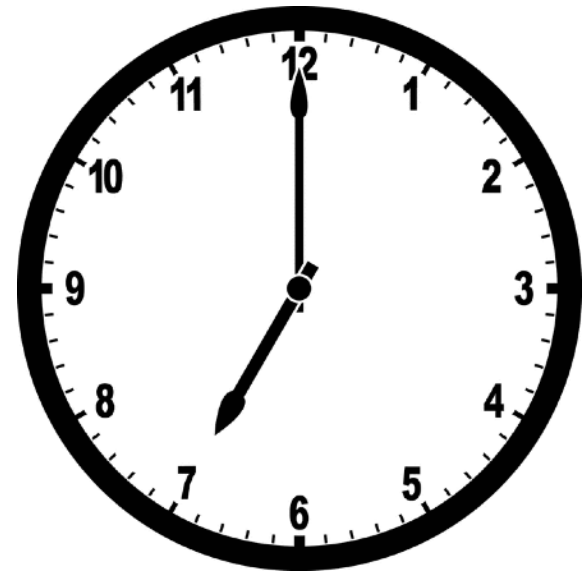
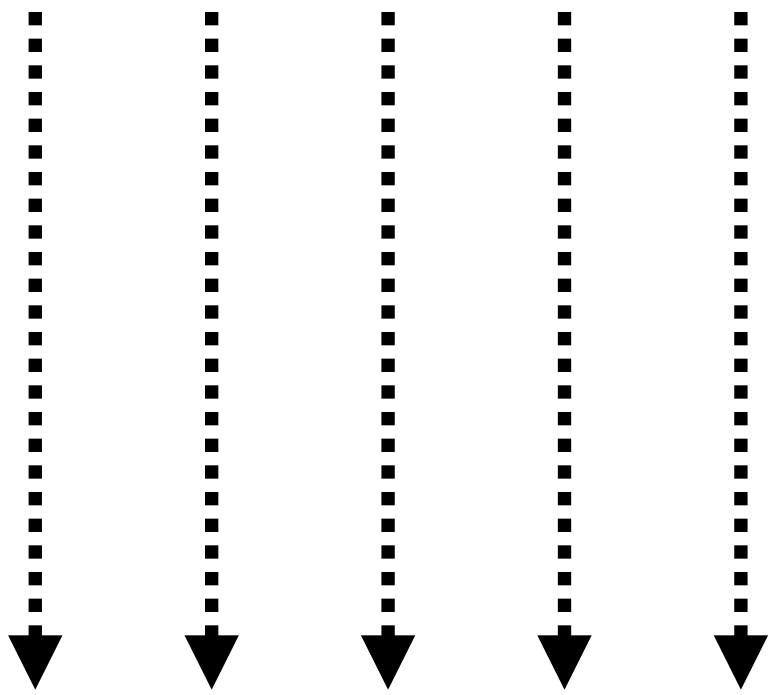
refactoring: store secure value flag inside parameter instead of calculating it each time
Pavel Sher at 24 Oct 19 23:42

[TW-62585](#) ▾ Use project external IDs correctly
Oleg Rybak at 24 Oct 19 18:58

[TW-59766](#) ▾ Target branch filter field is made expandable and a note underneath has been updated
Anton Zamolotskikh at 24 Oct 19 18:49

[TW-60695](#) ▾ Add an ability to specify bucket name without list privileges.
Dmitrii Bogdanov at 24 Oct 19 18:48

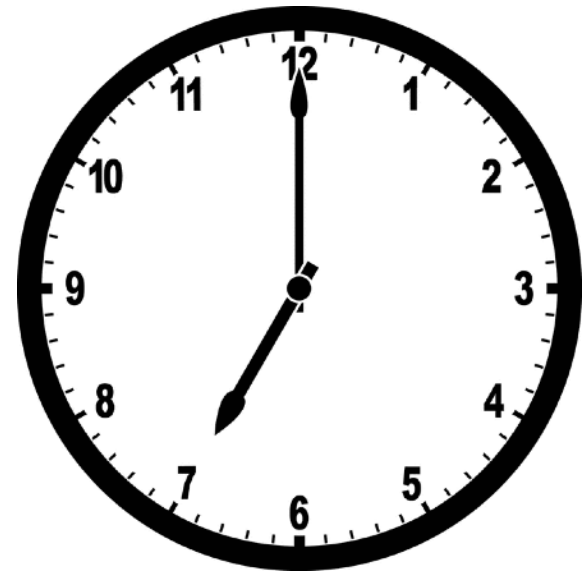
master



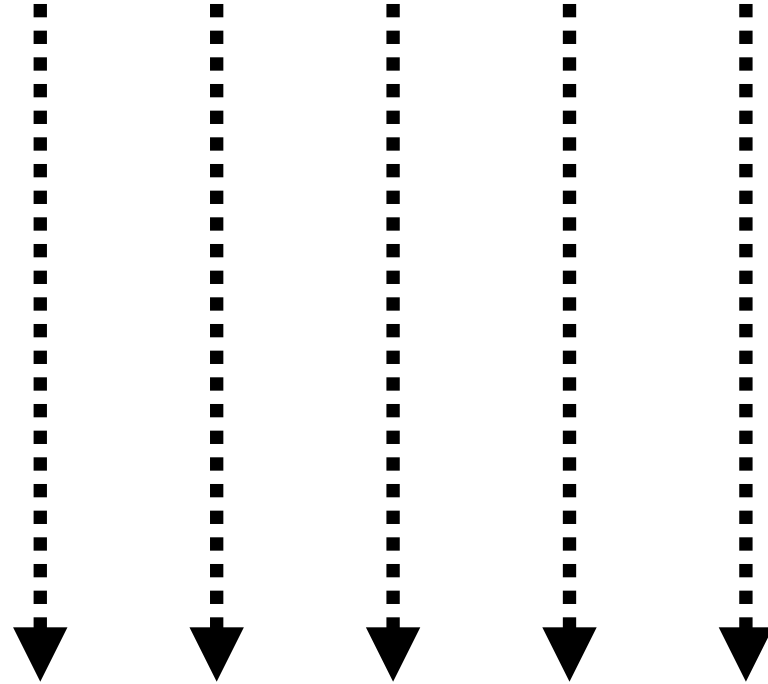
buildserver



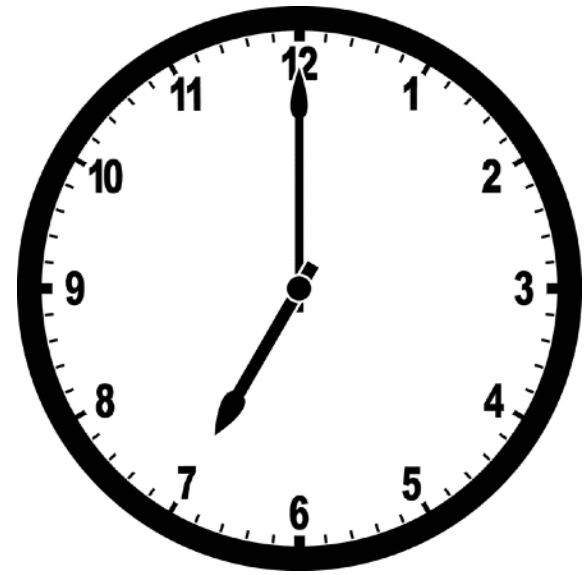
master



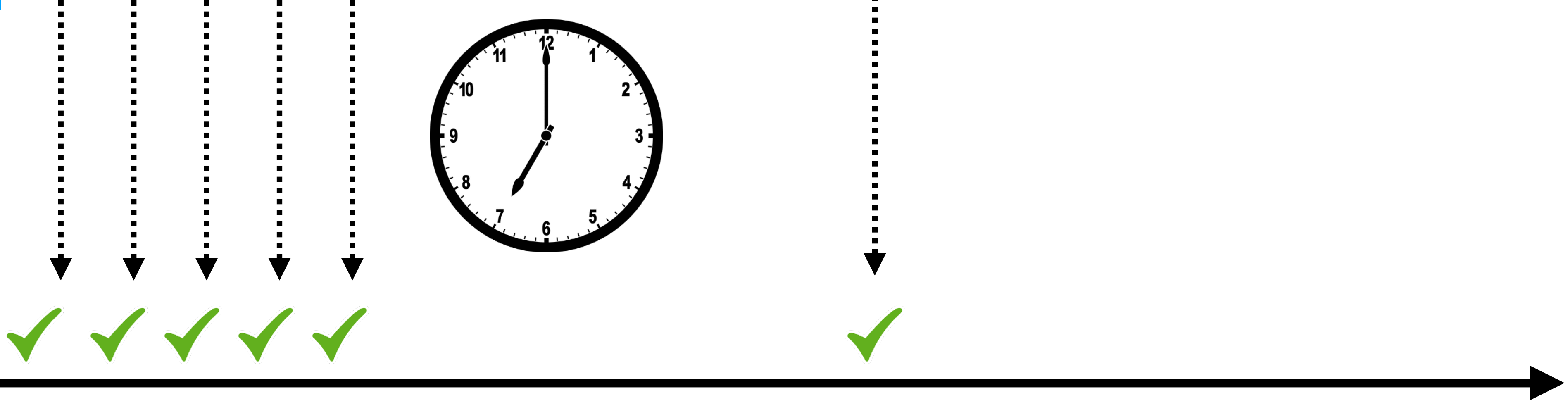
buildserver



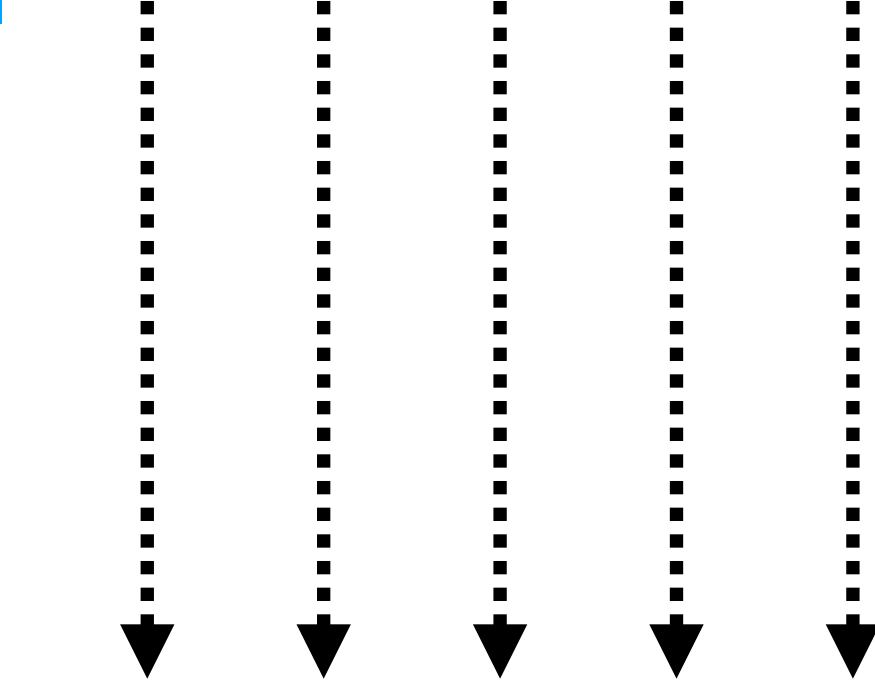
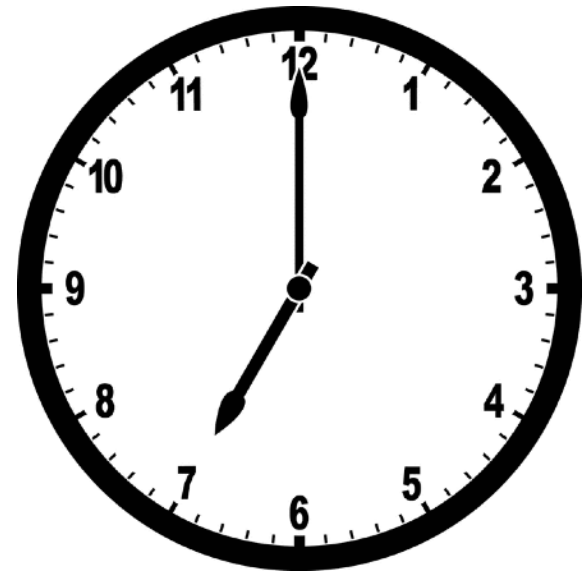
master



buildserver



master

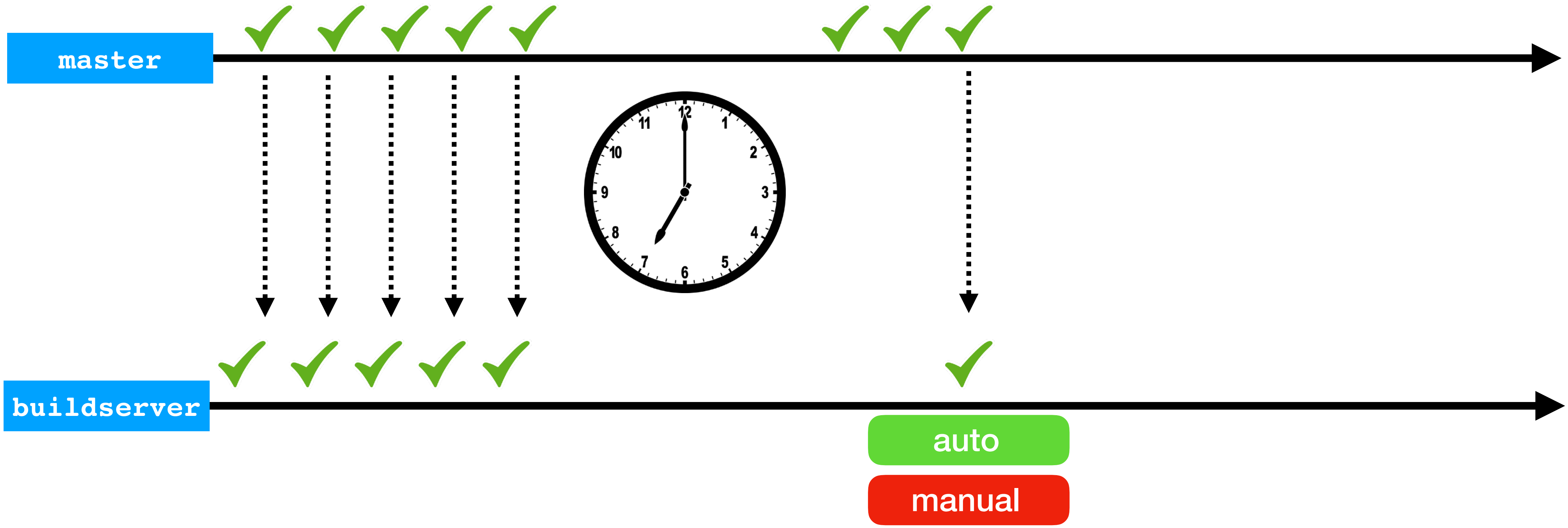


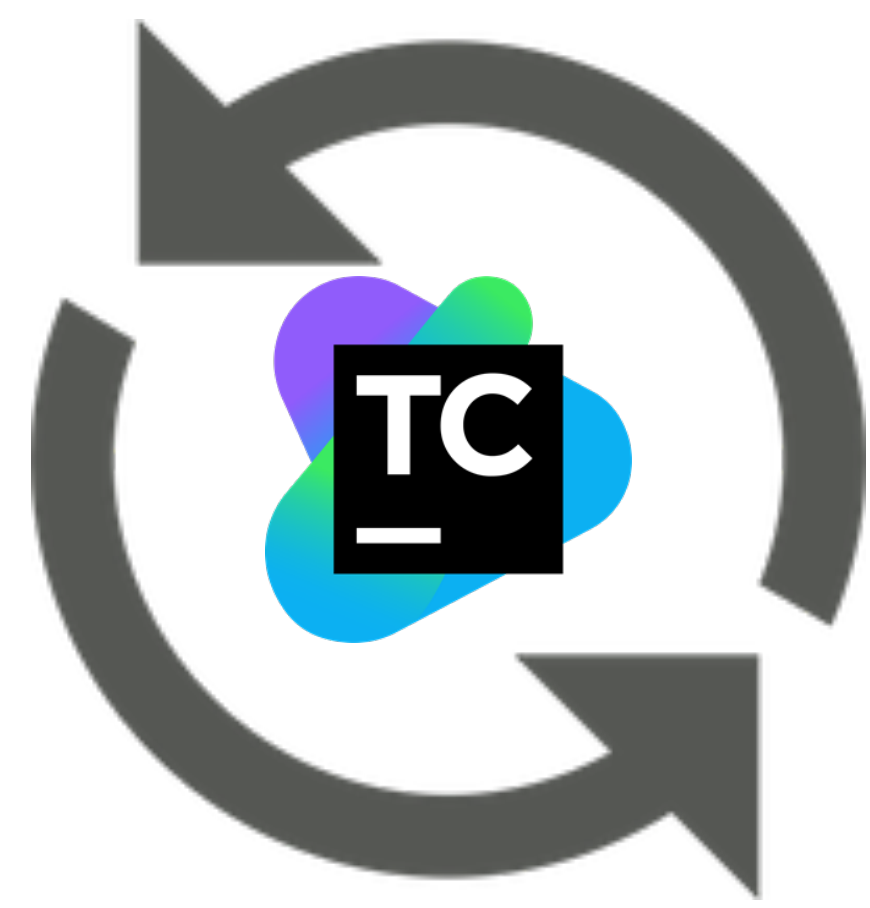
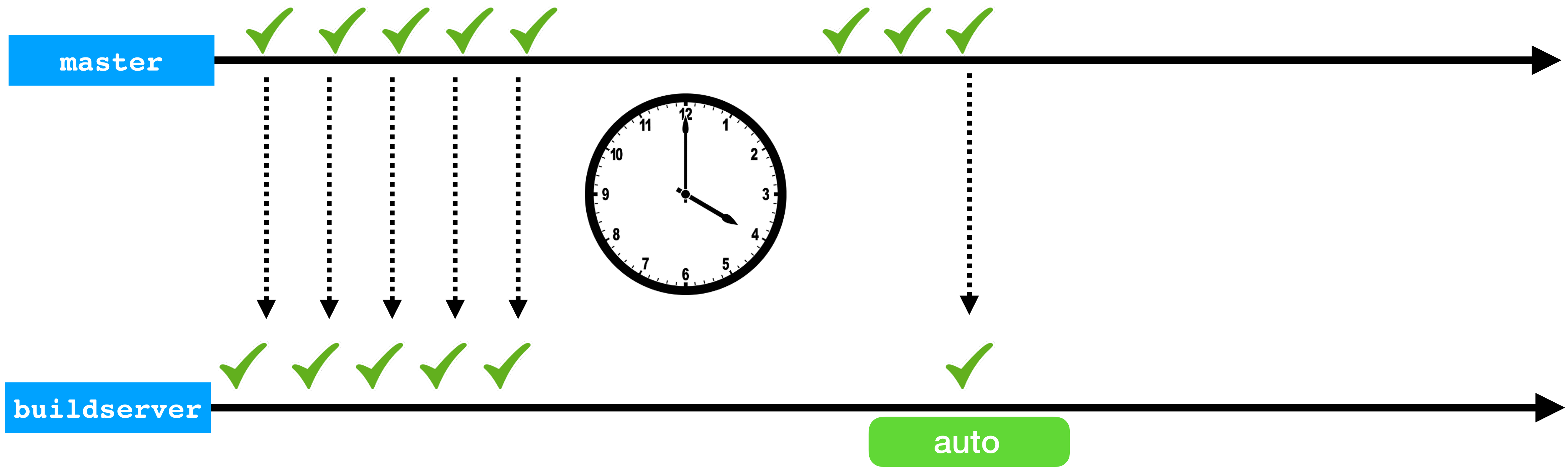
buildserver

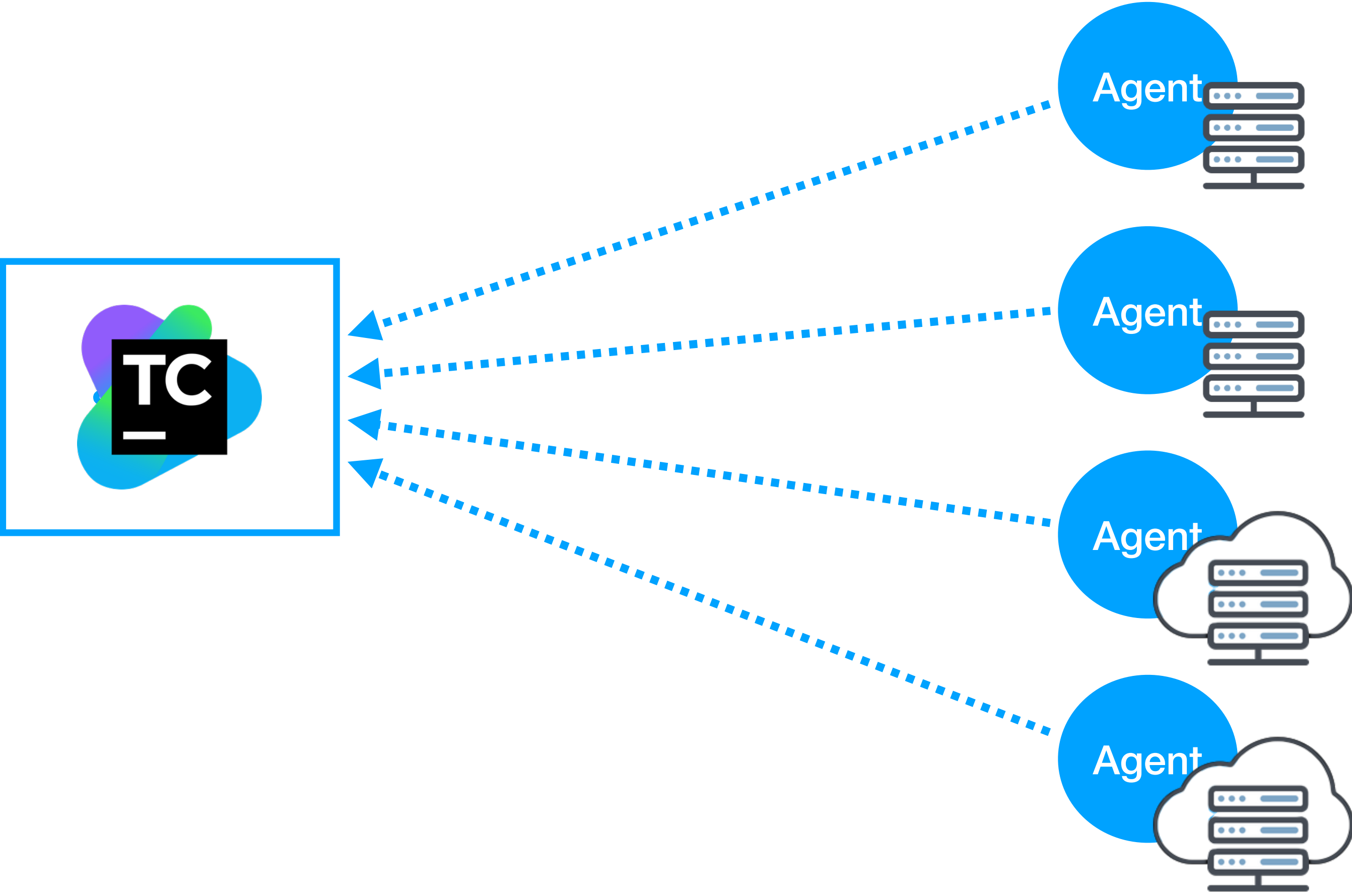


auto

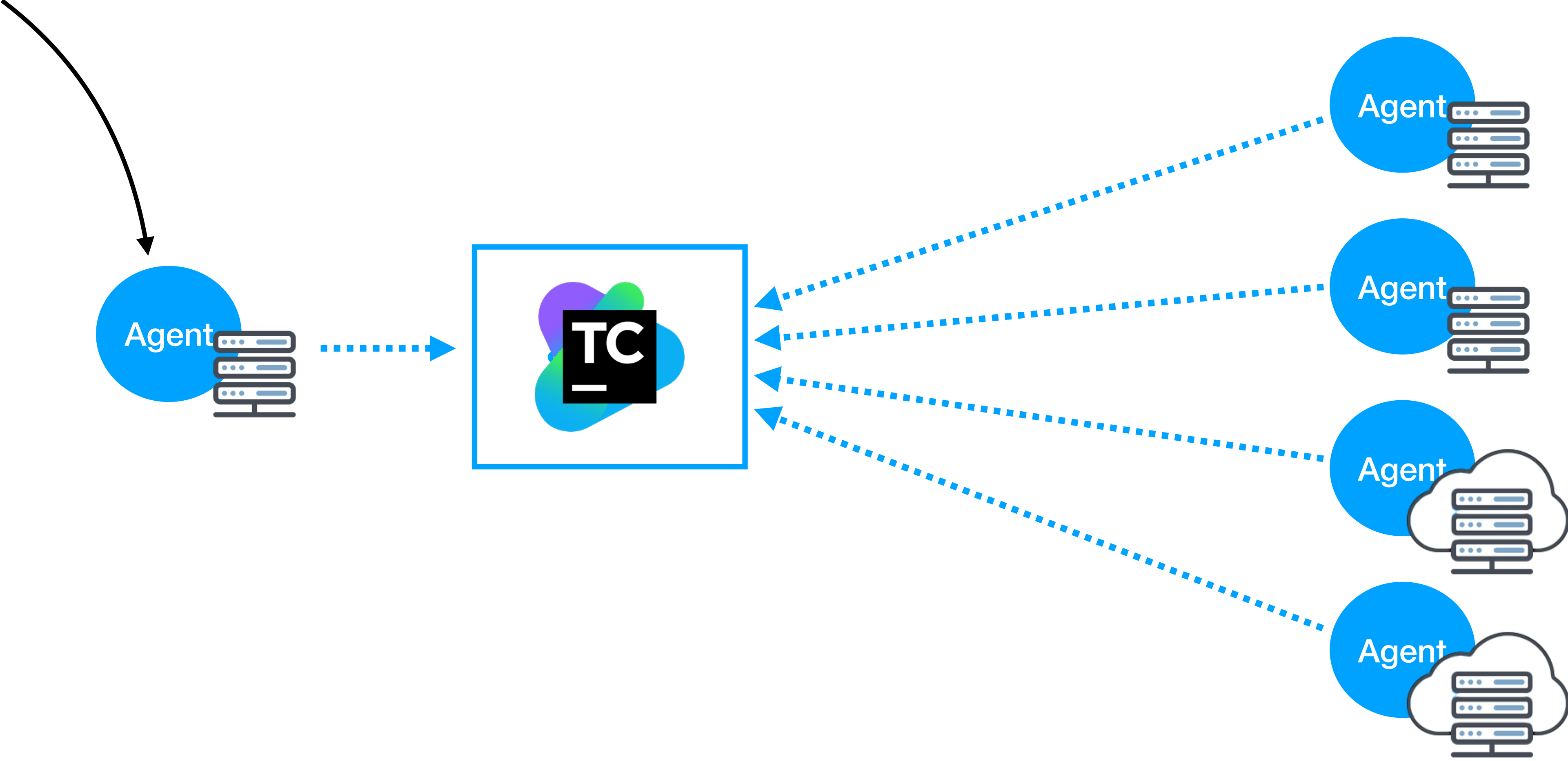


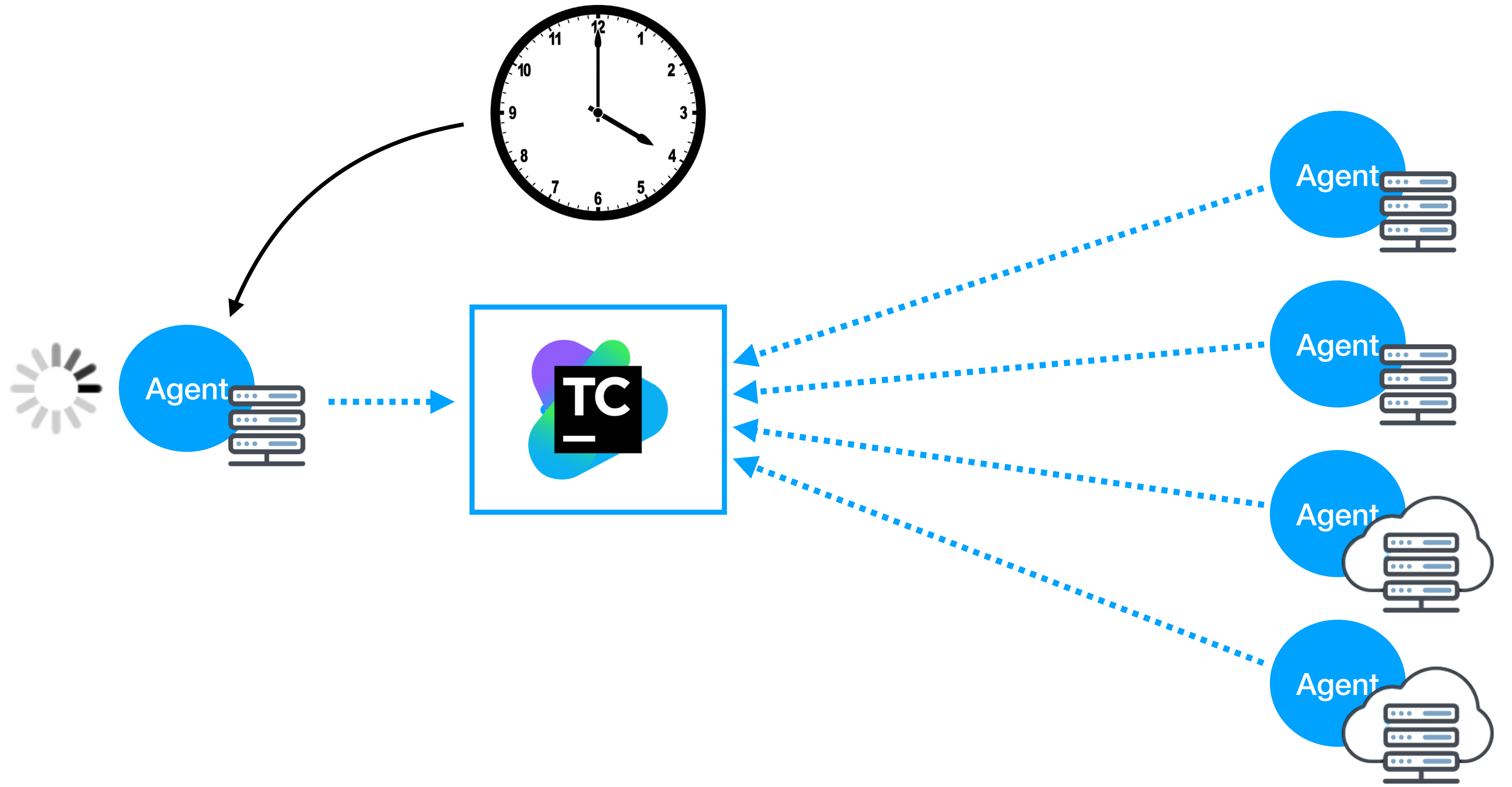


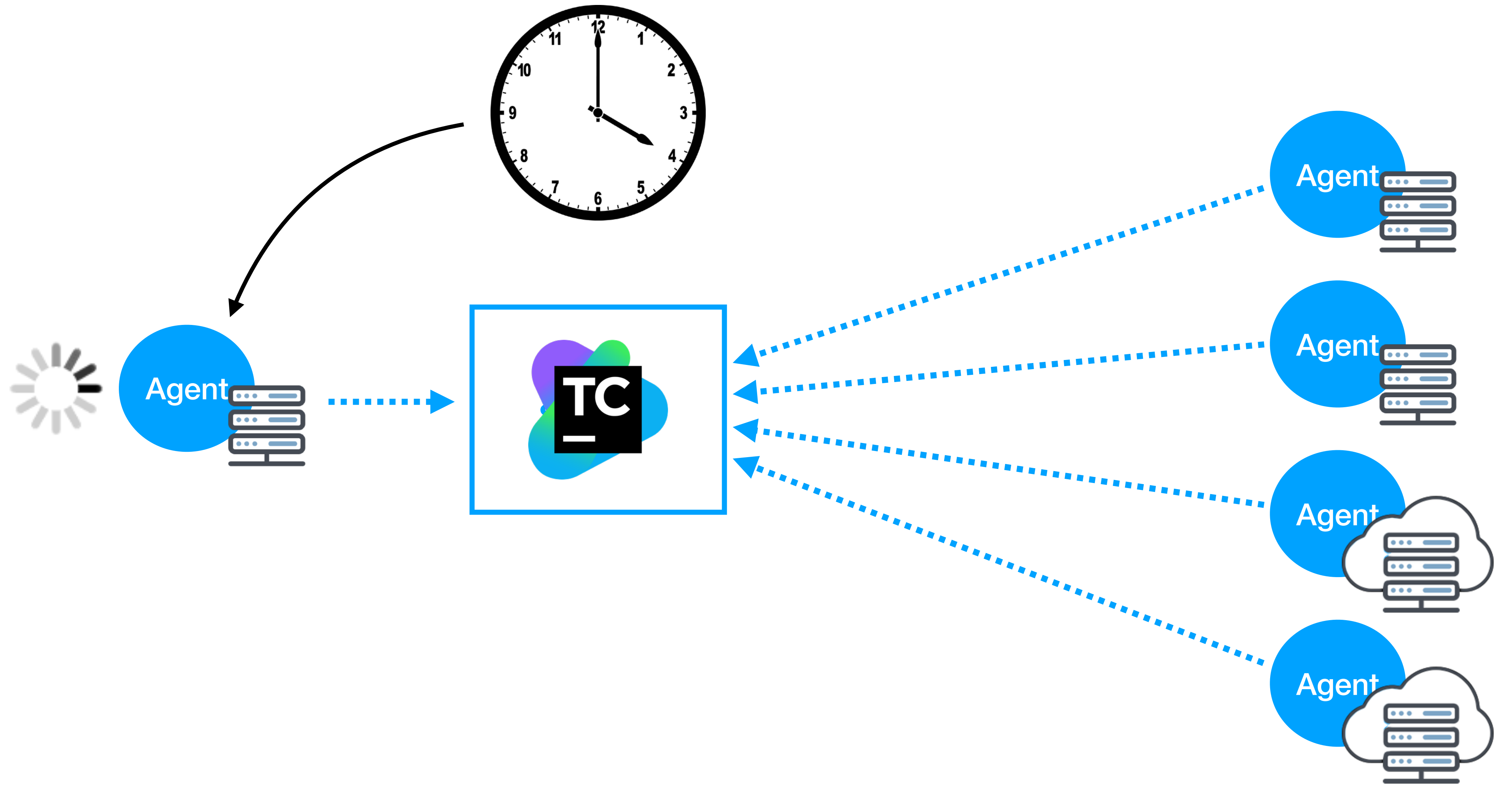


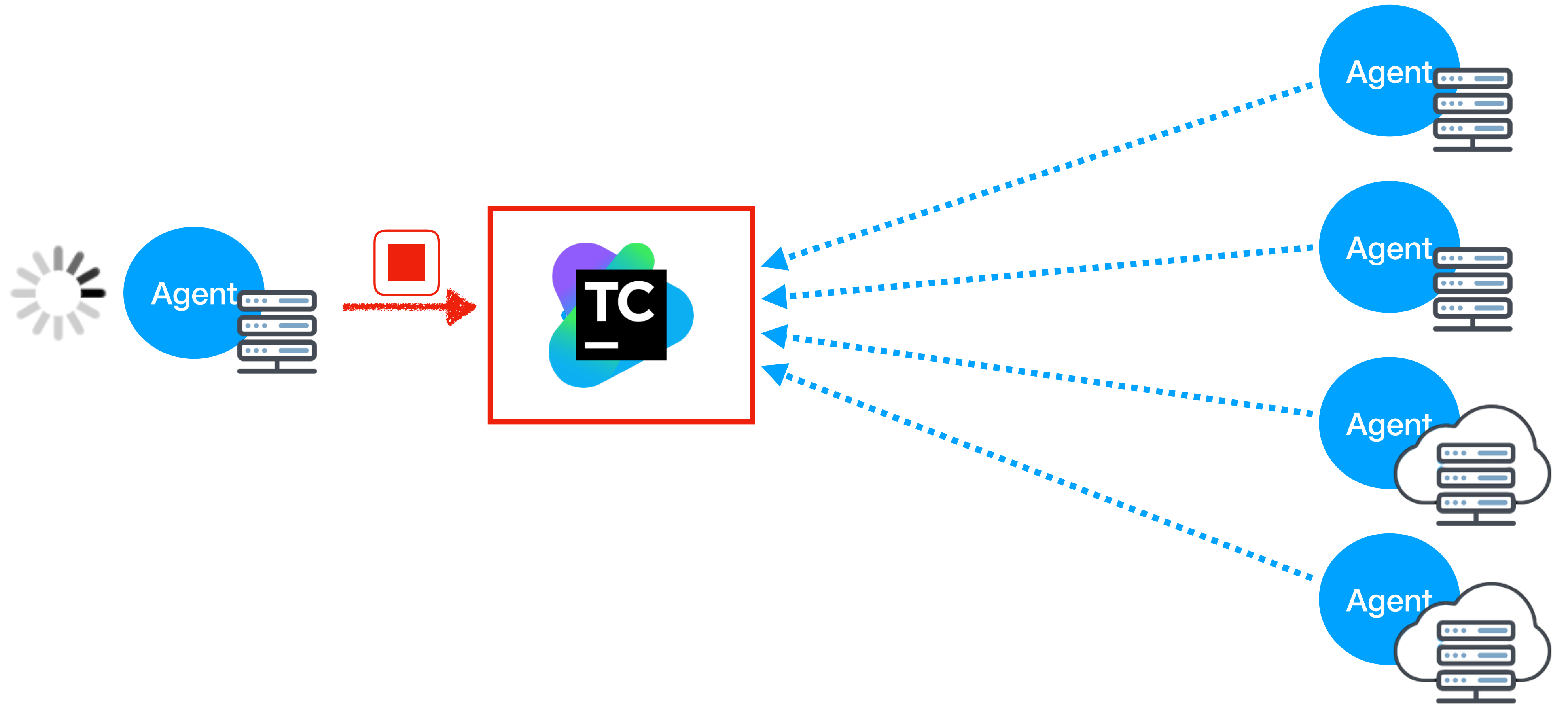


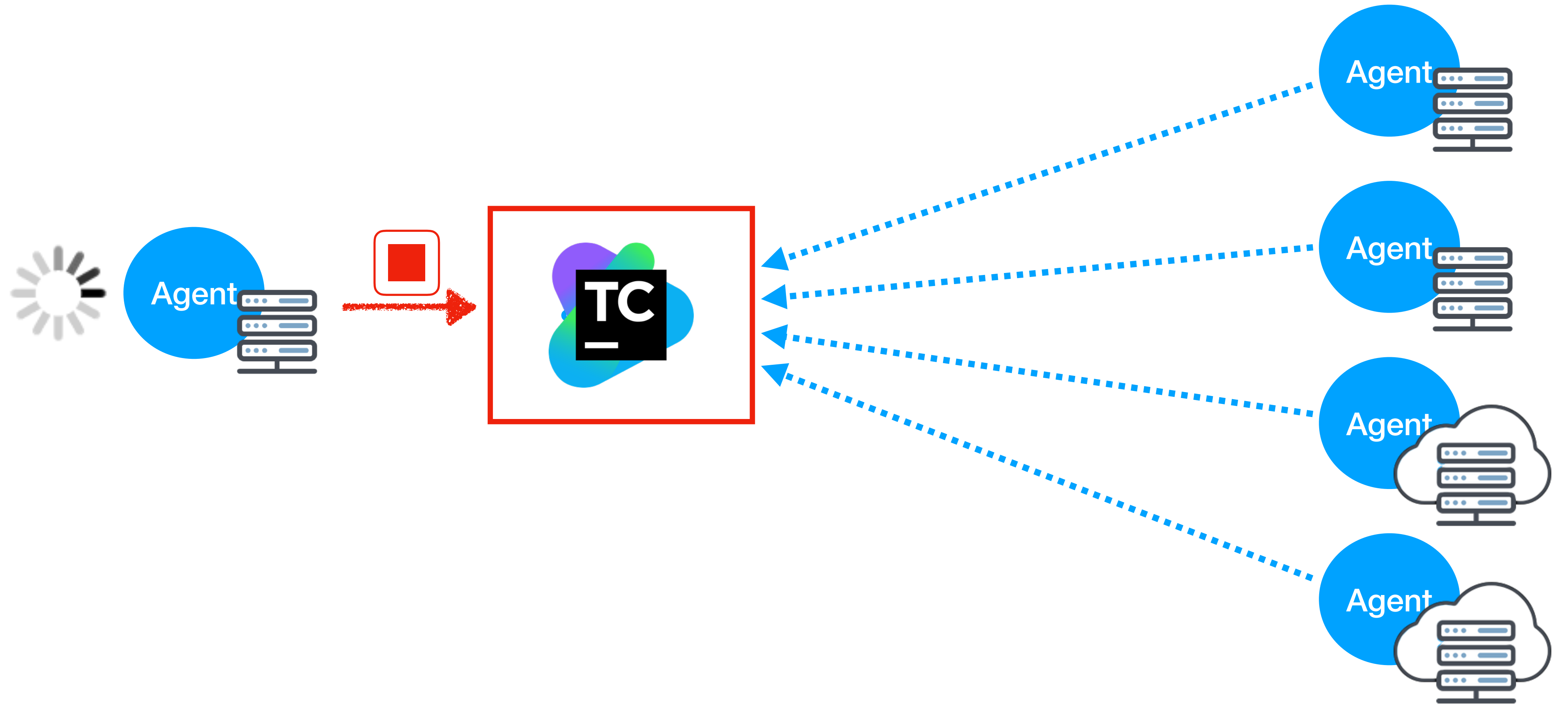
“Агент для обновлений”



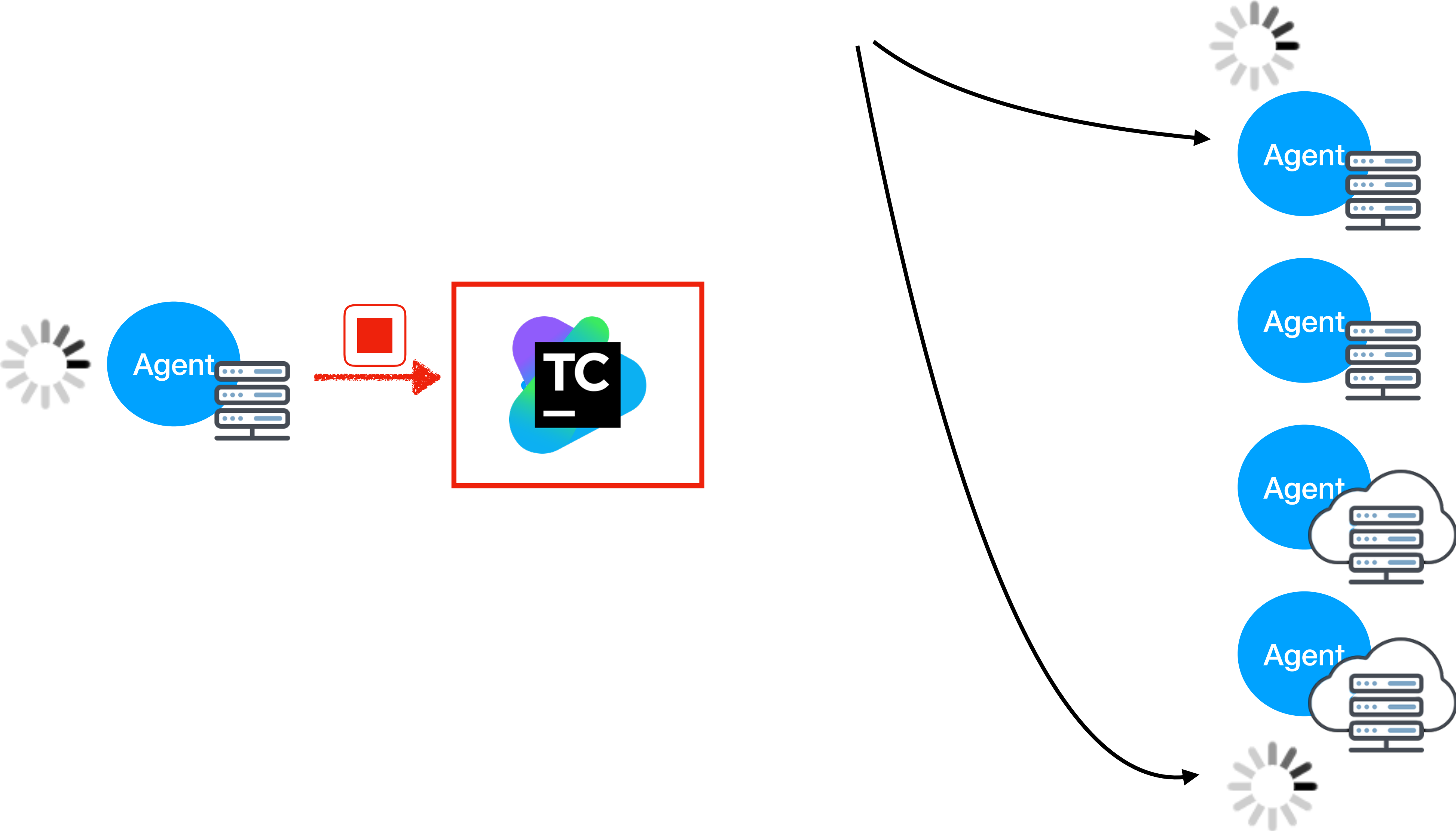




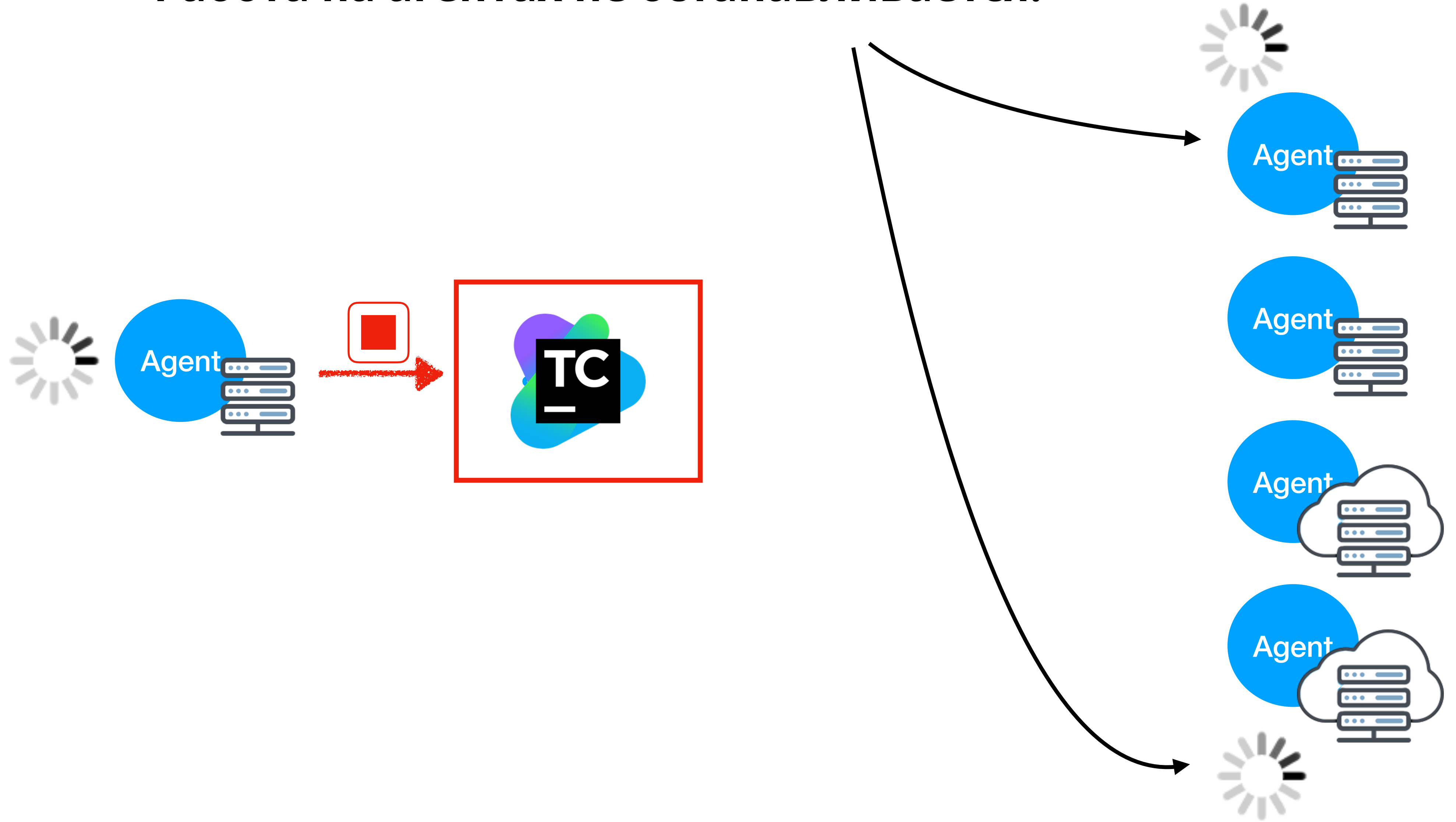


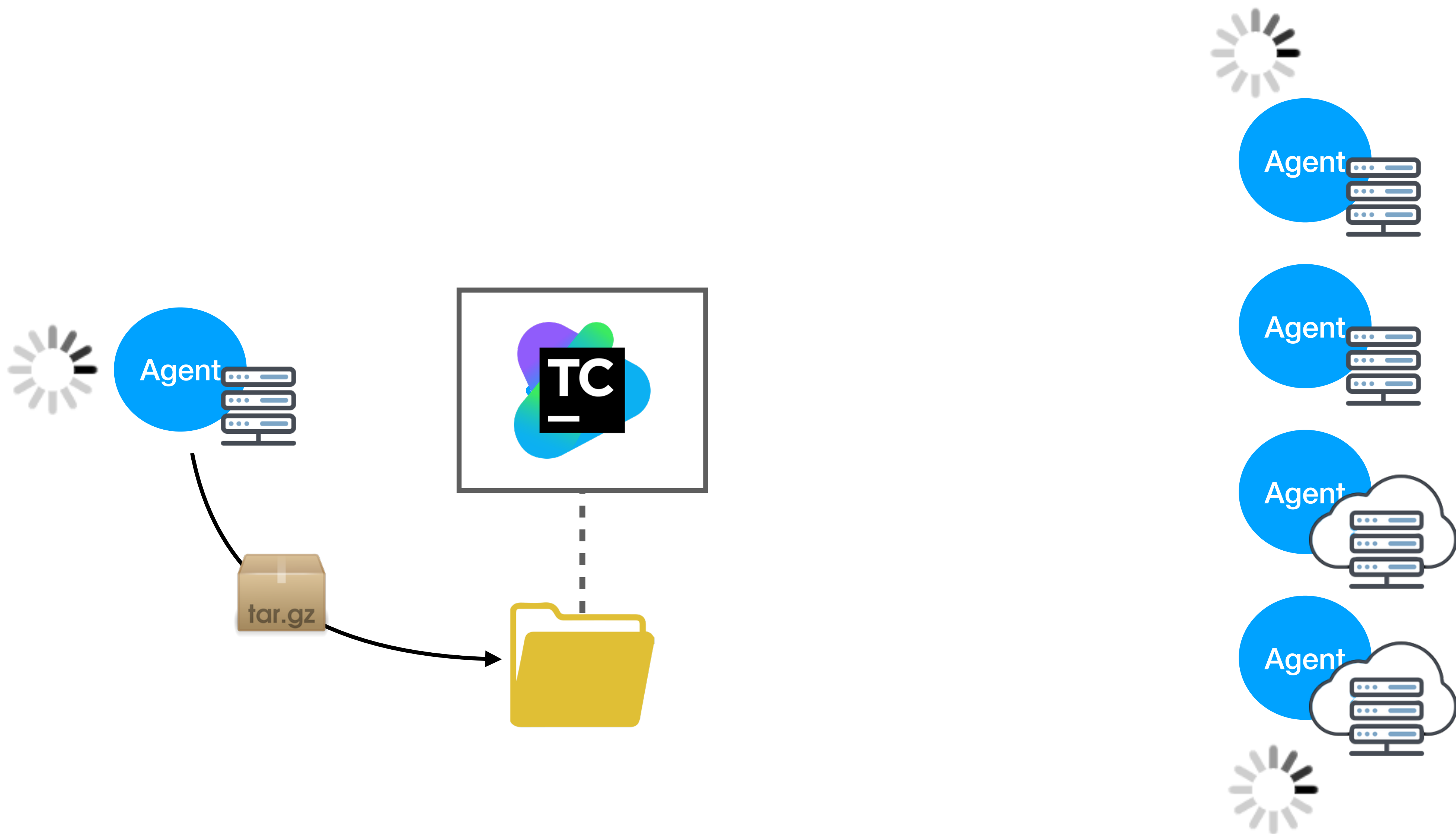


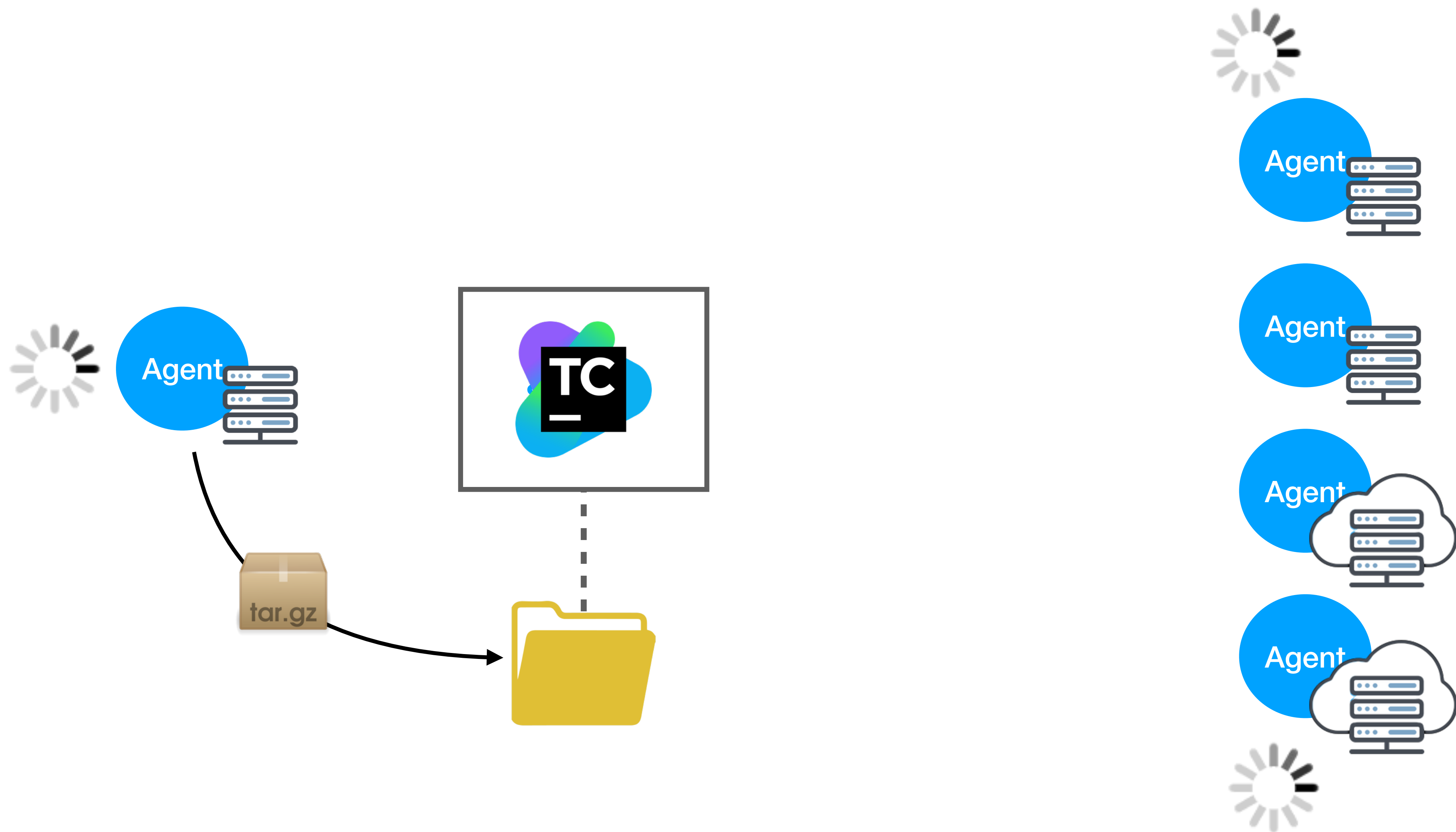
Работа на агентах не останавливается!

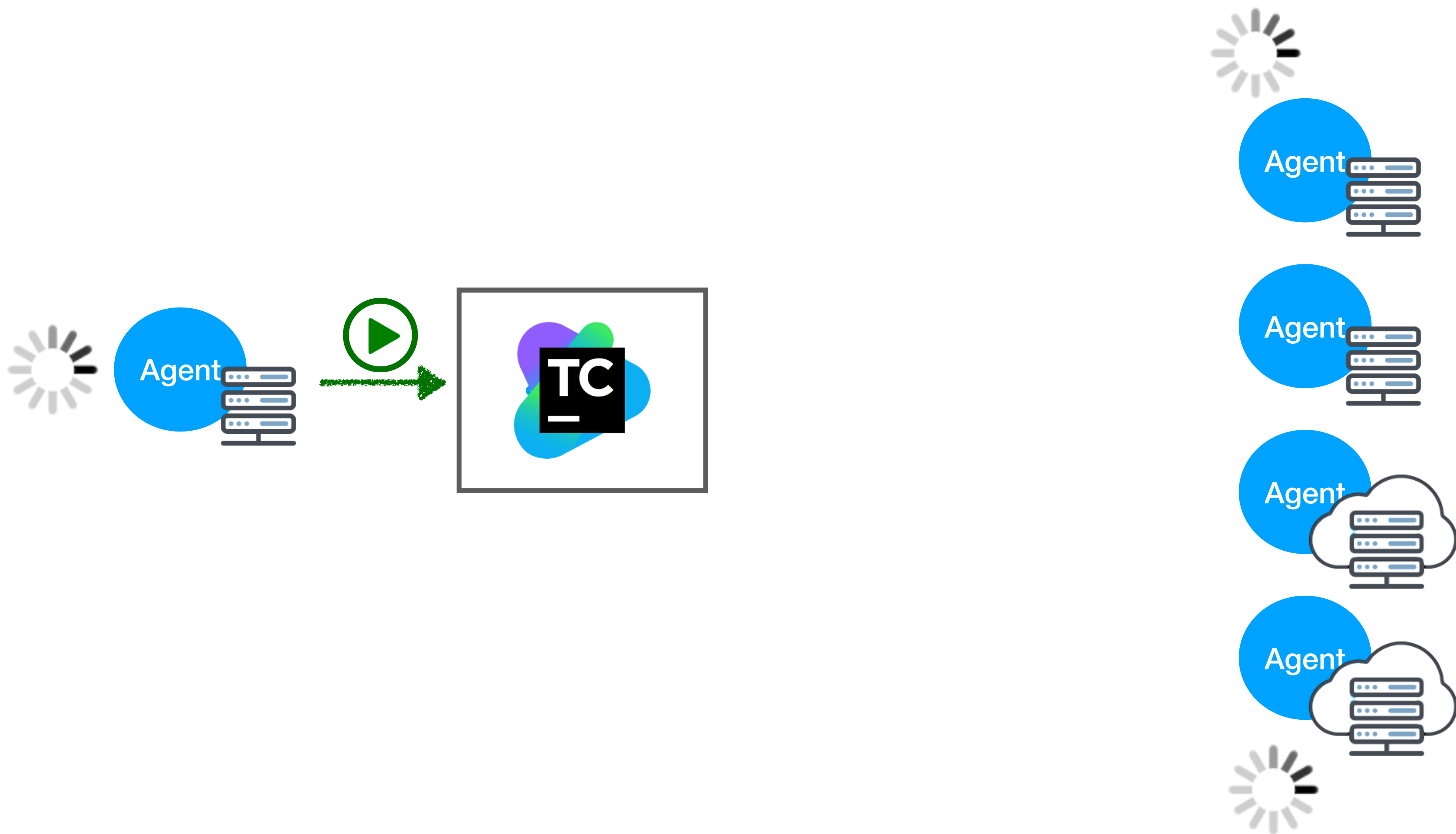


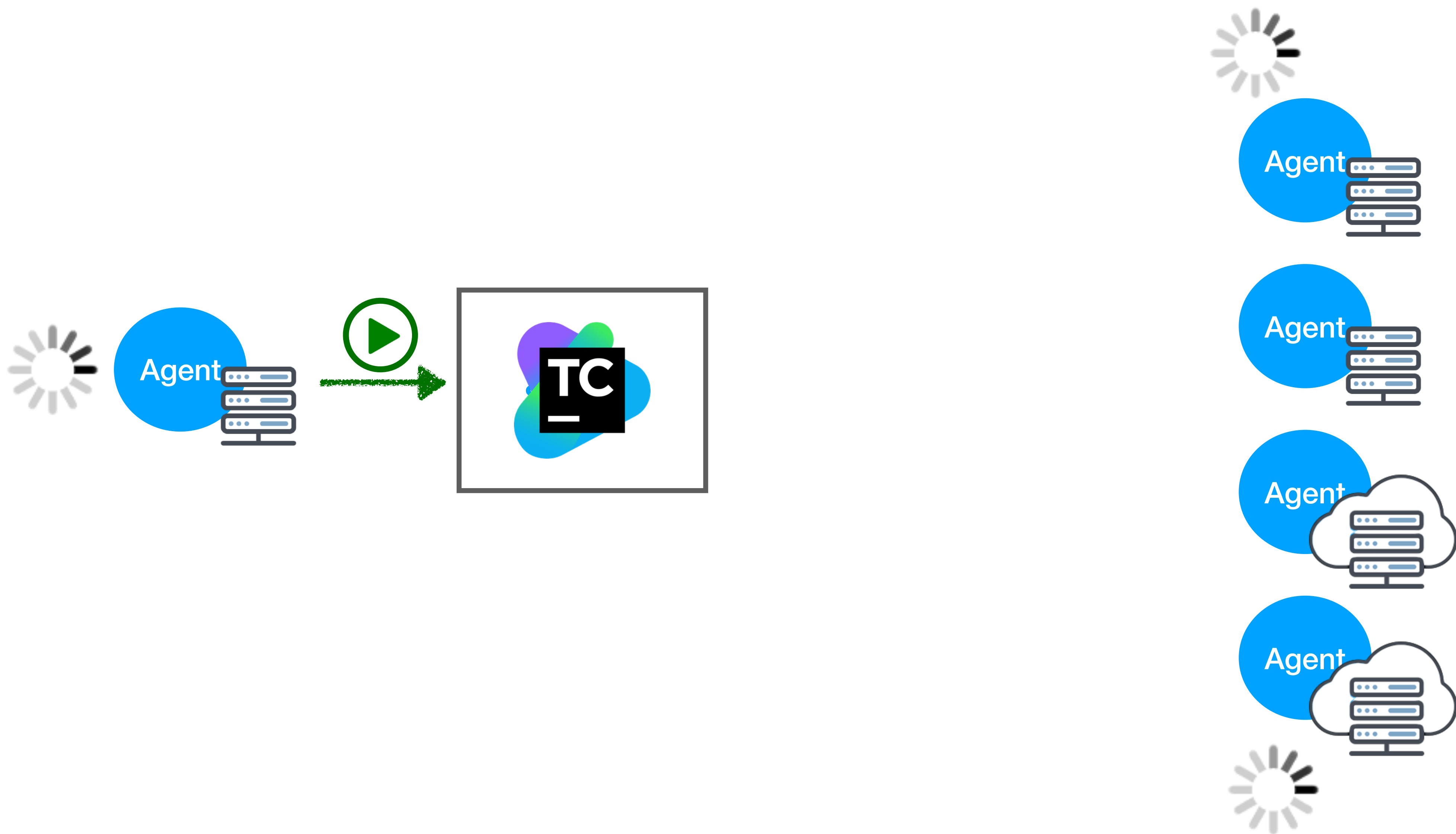
Работа на агентах не останавливается!

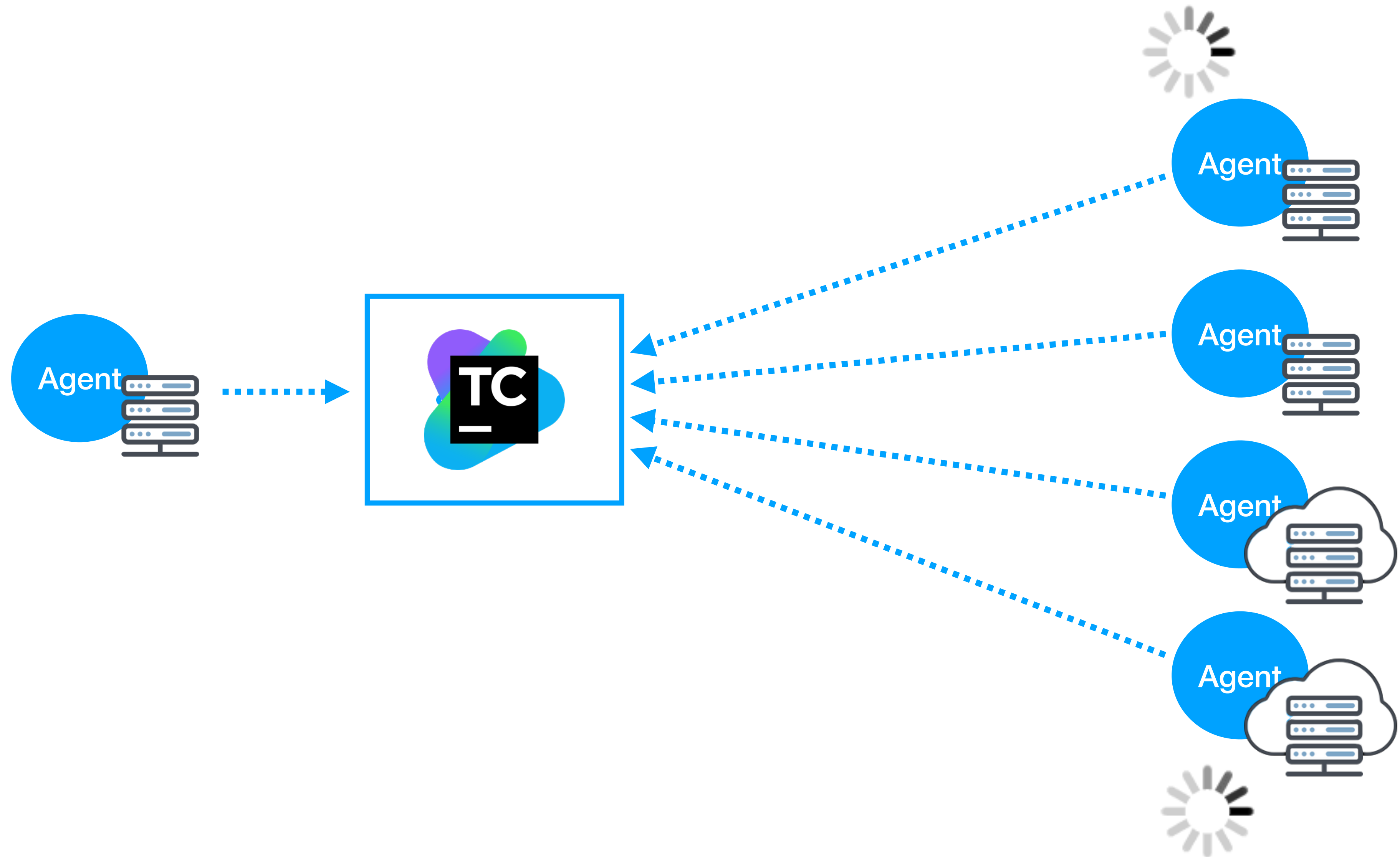


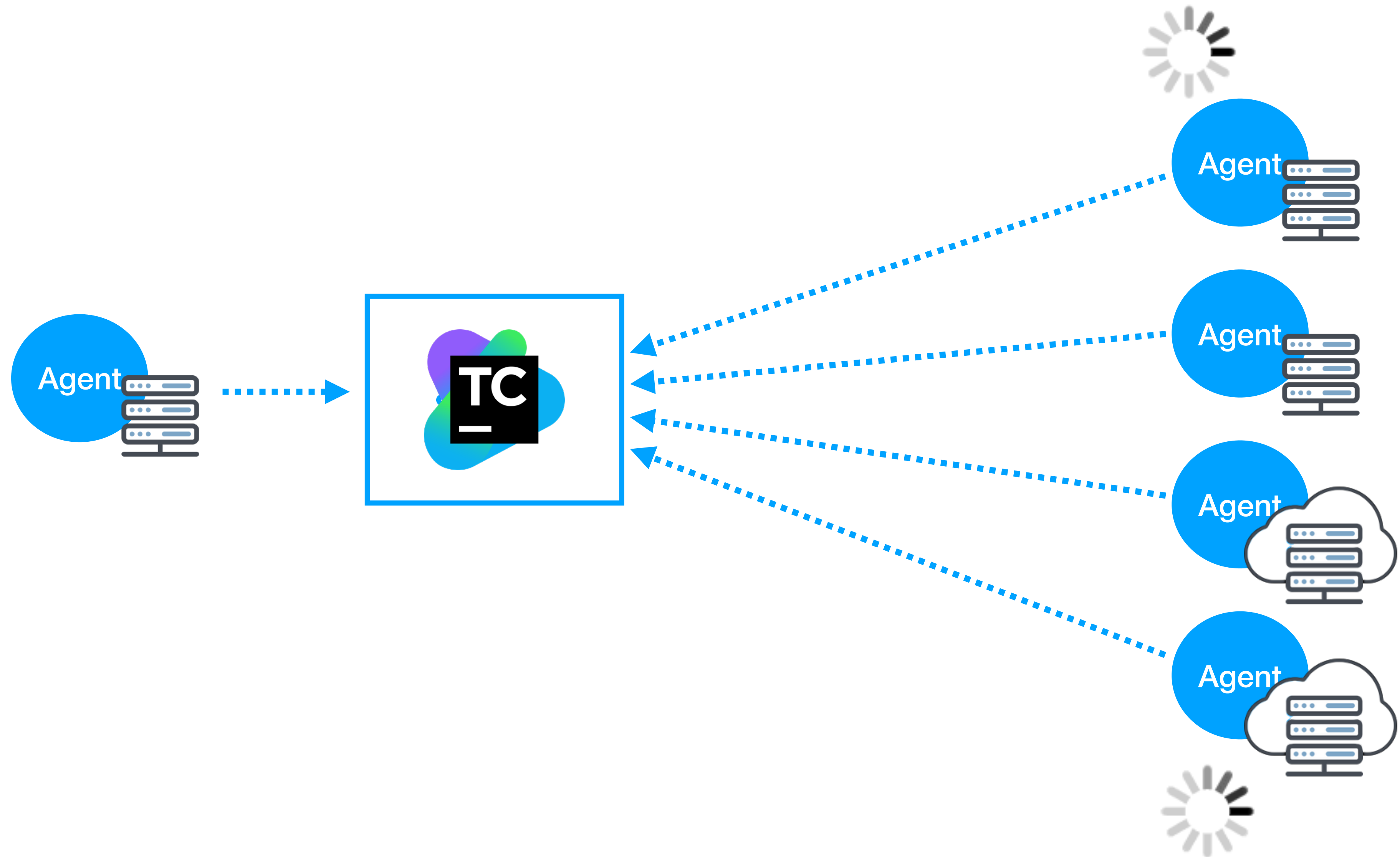


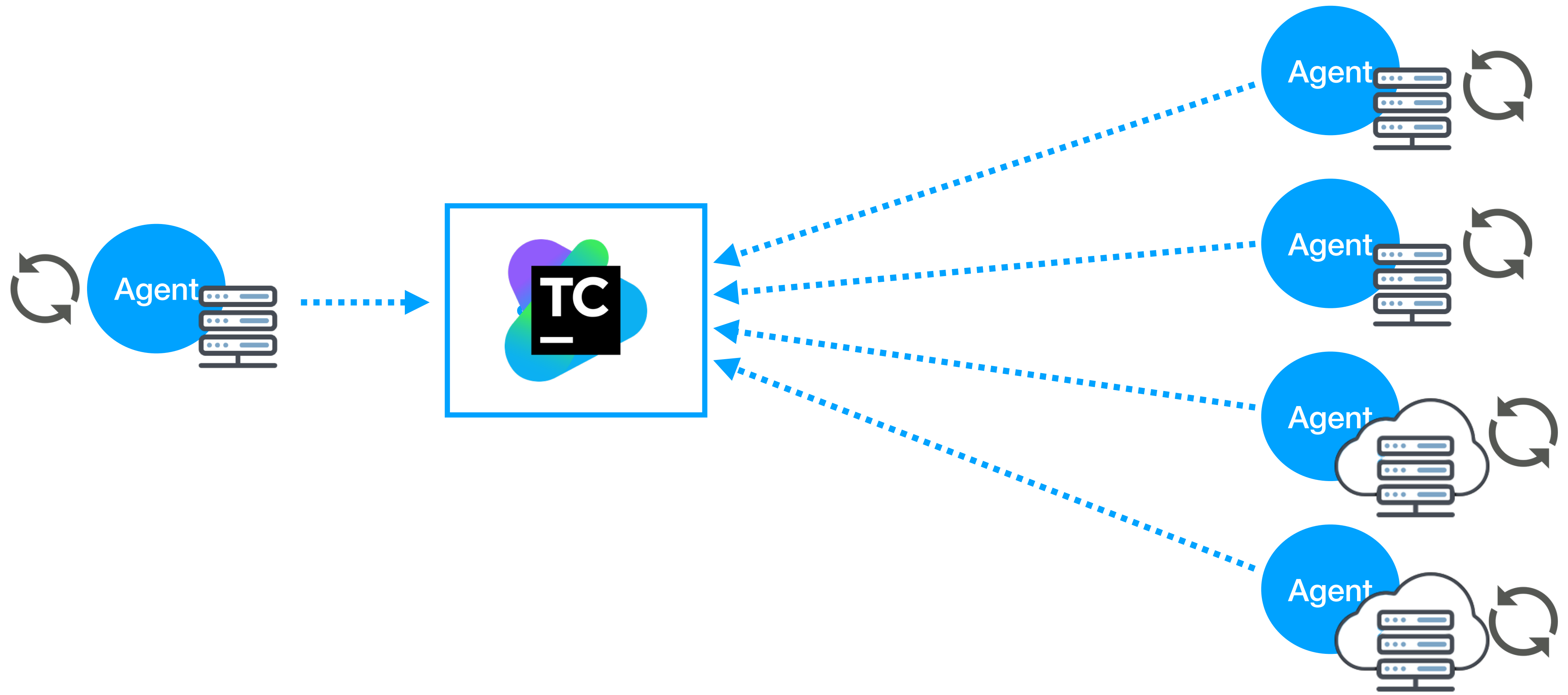


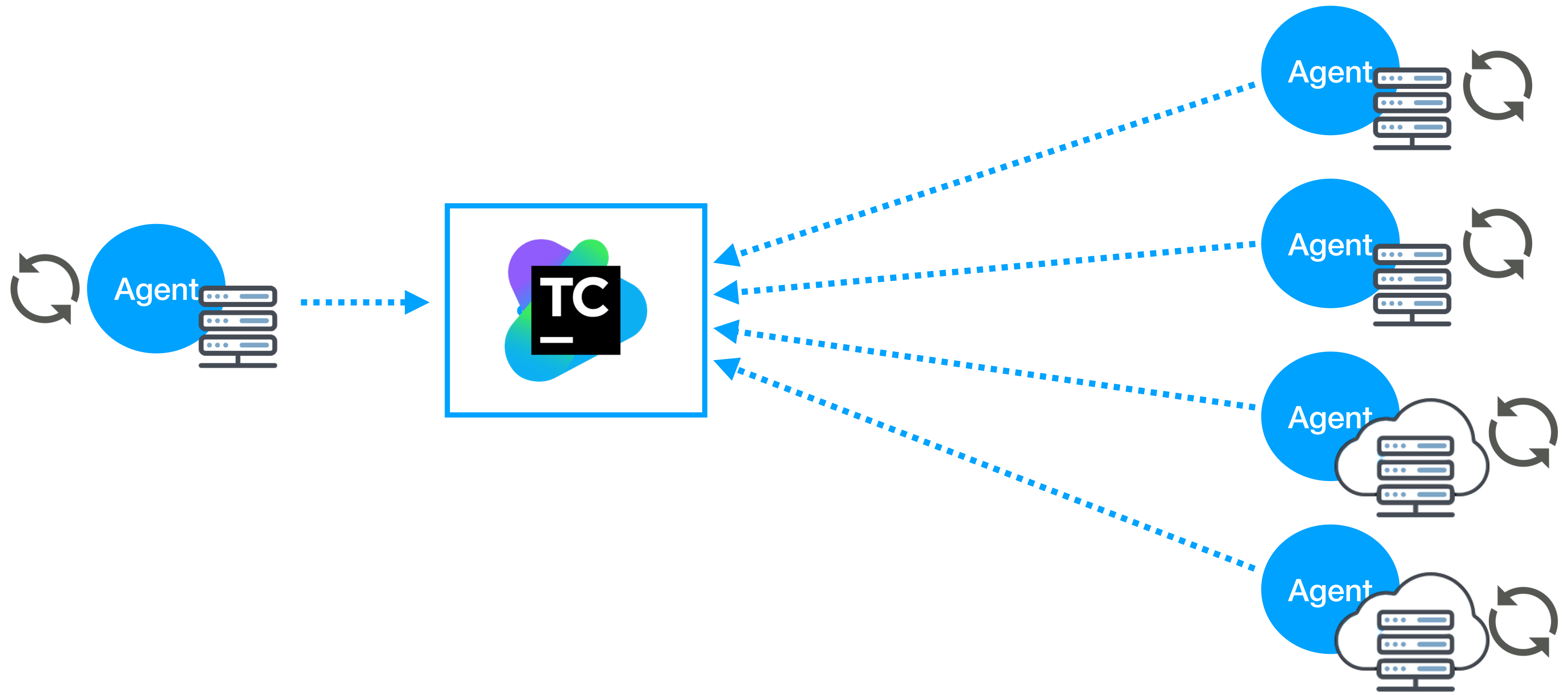


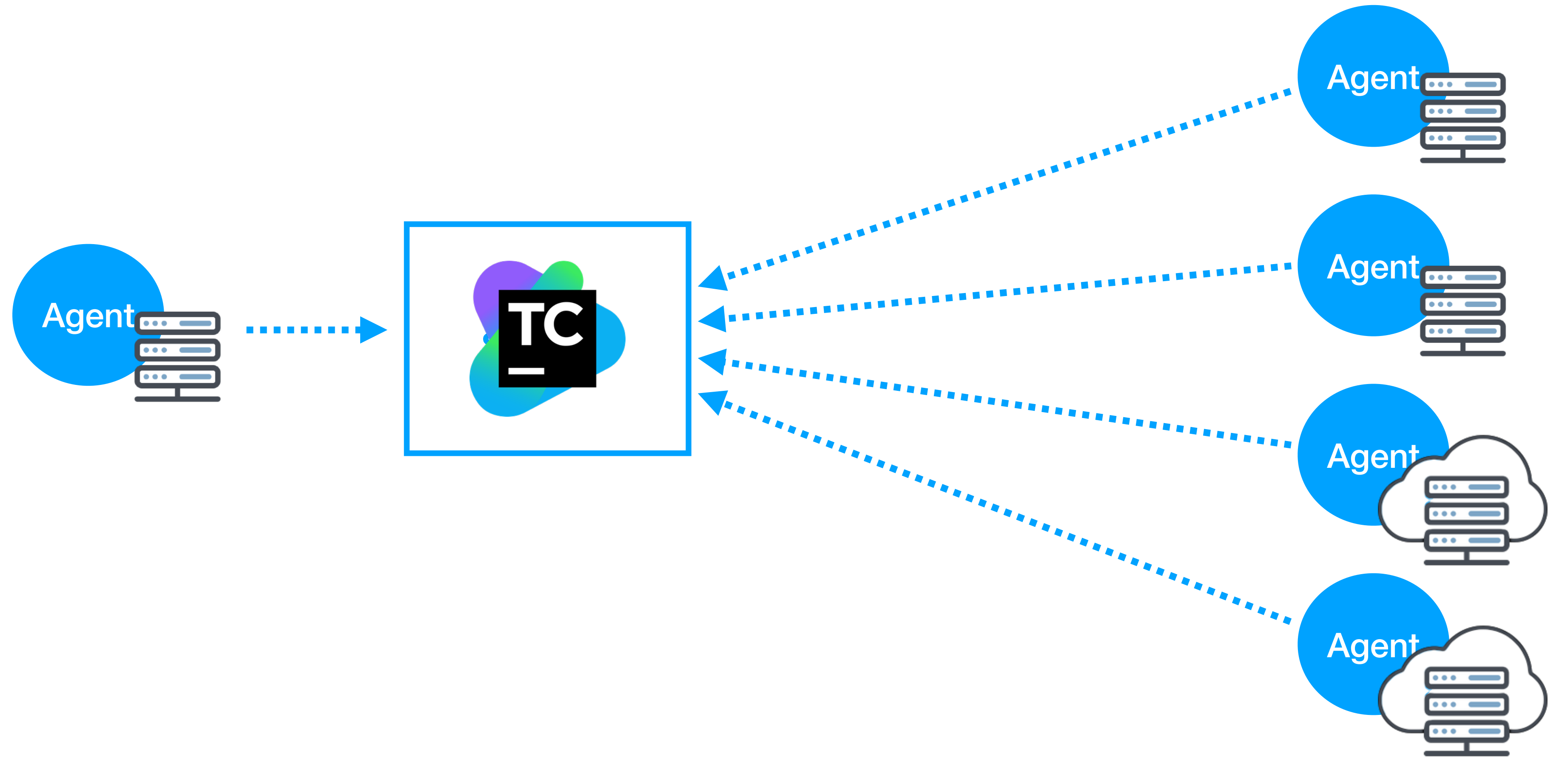






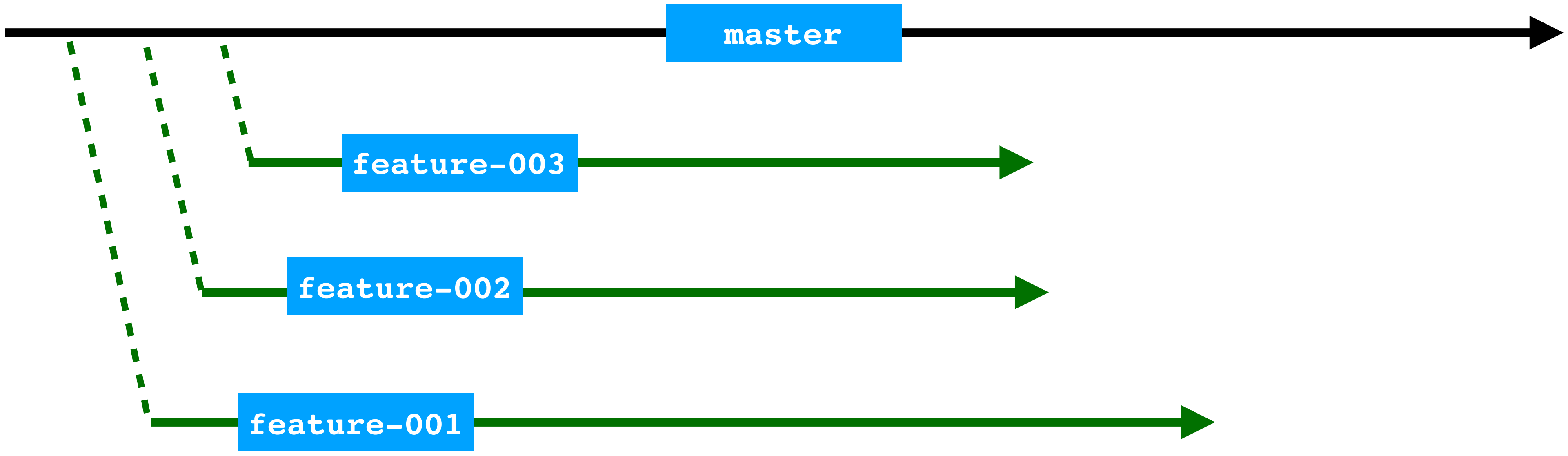


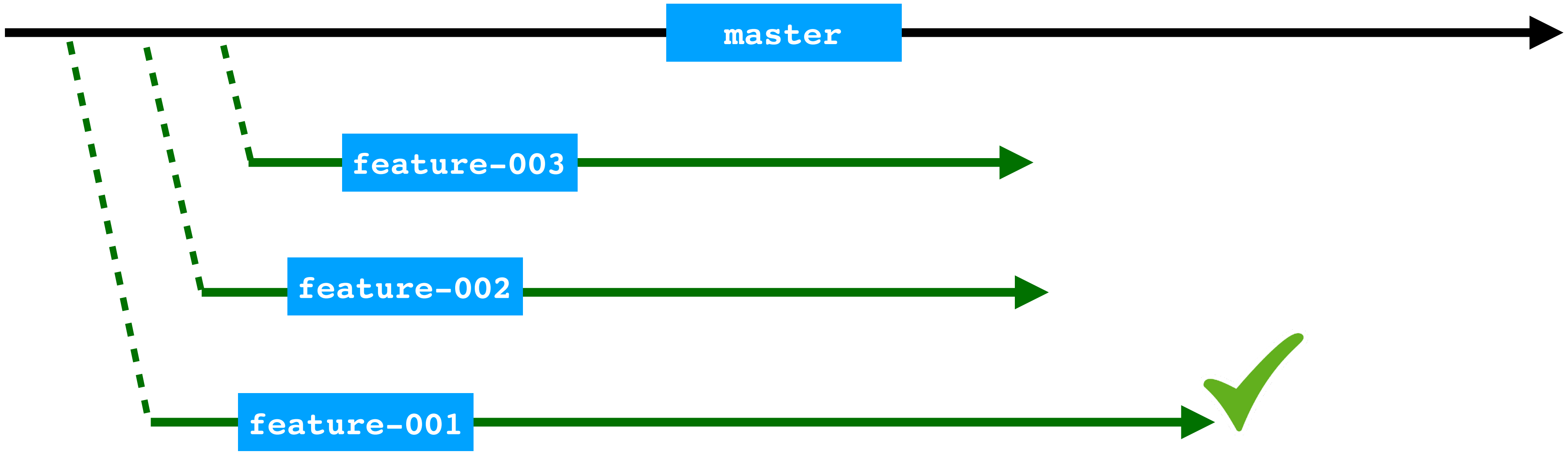


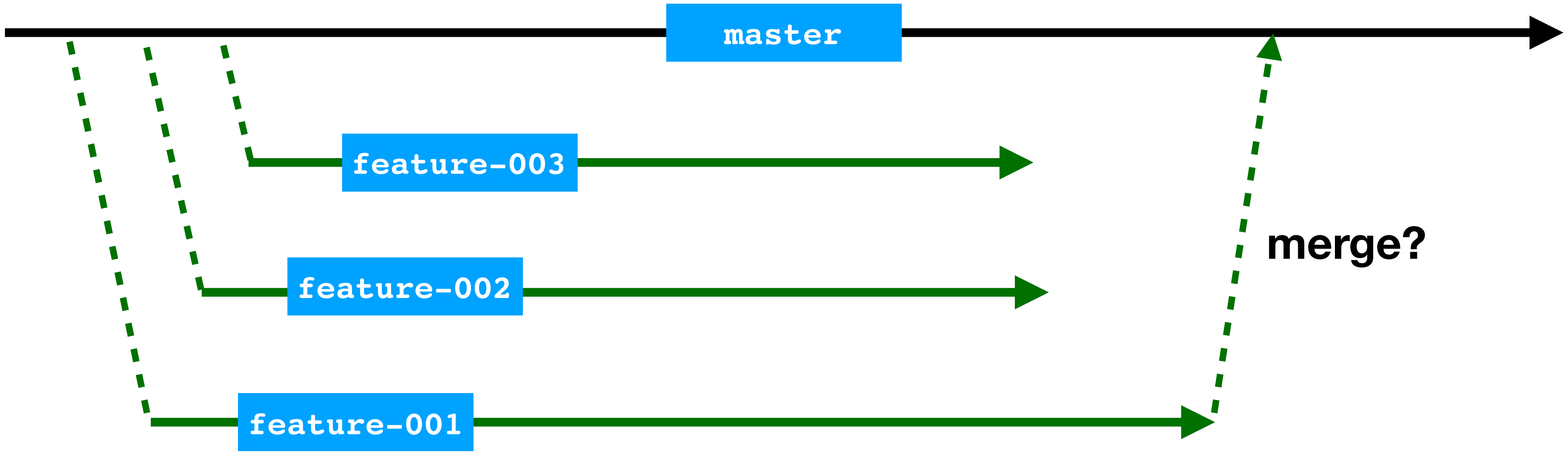


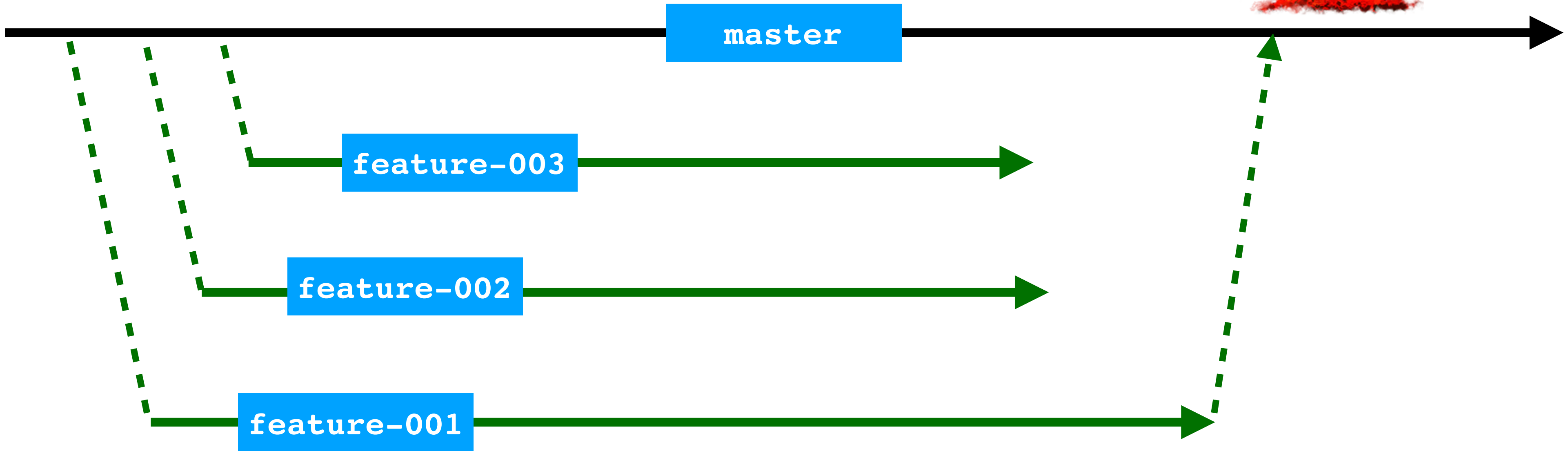


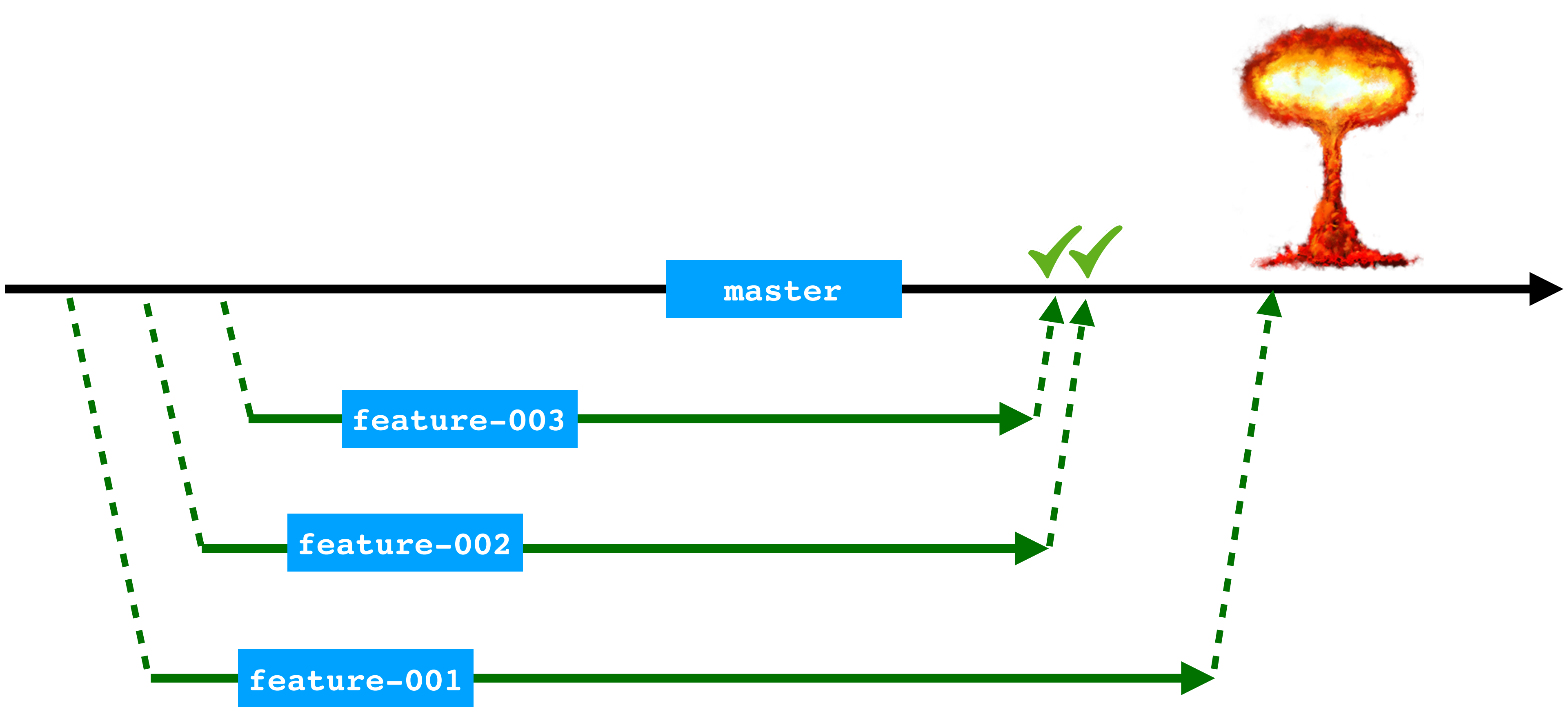


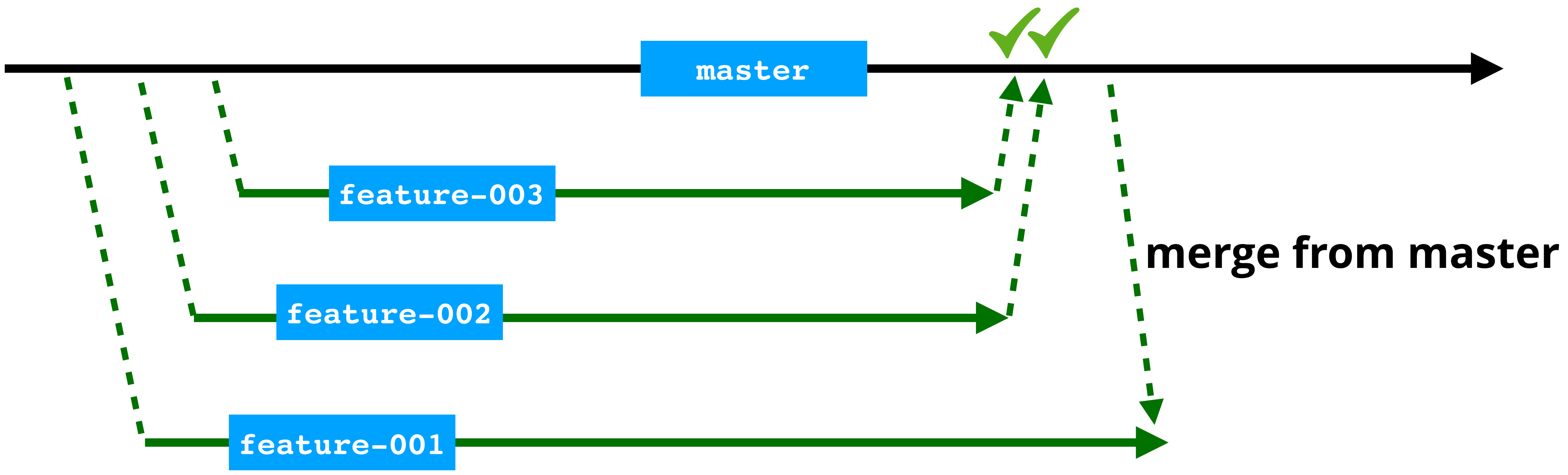


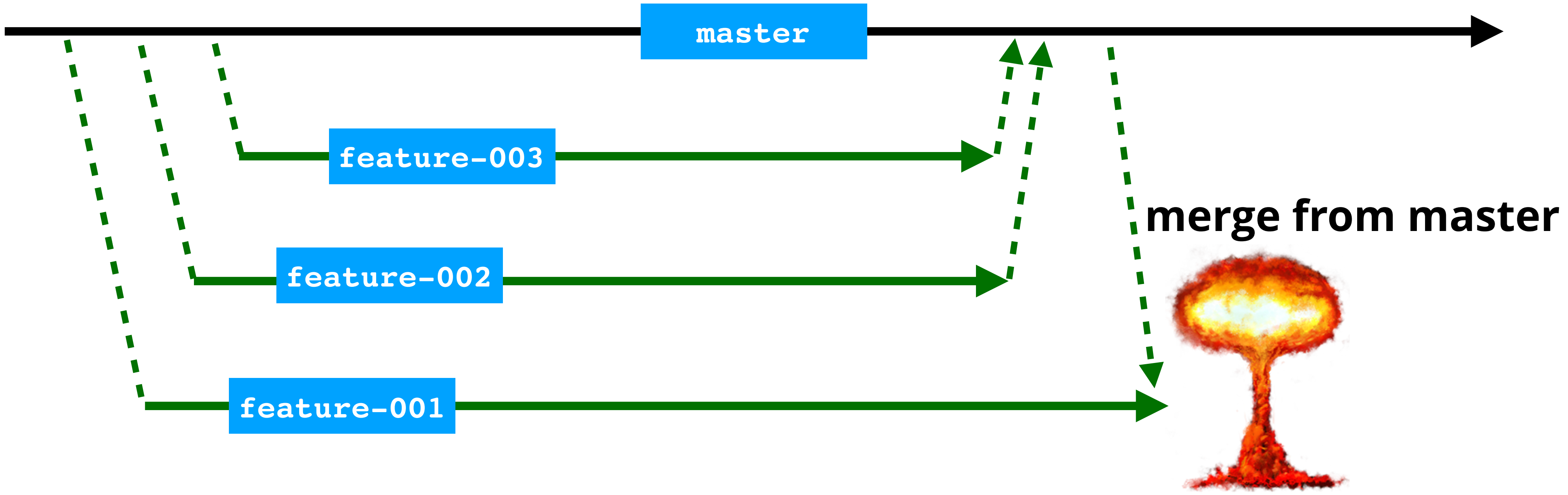


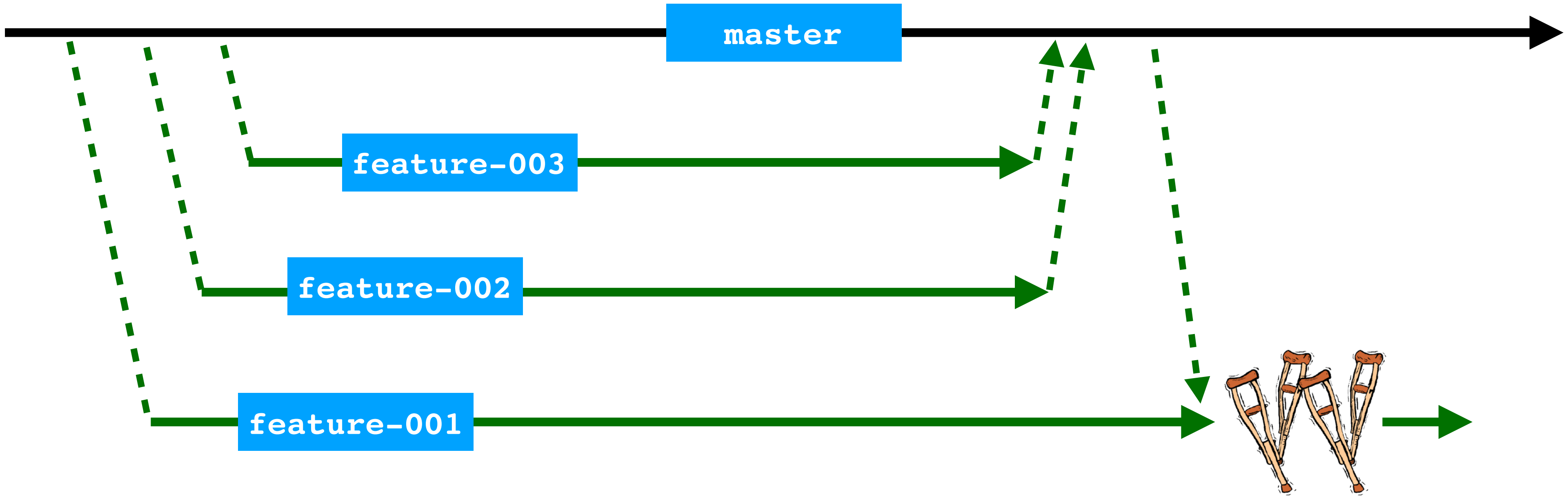


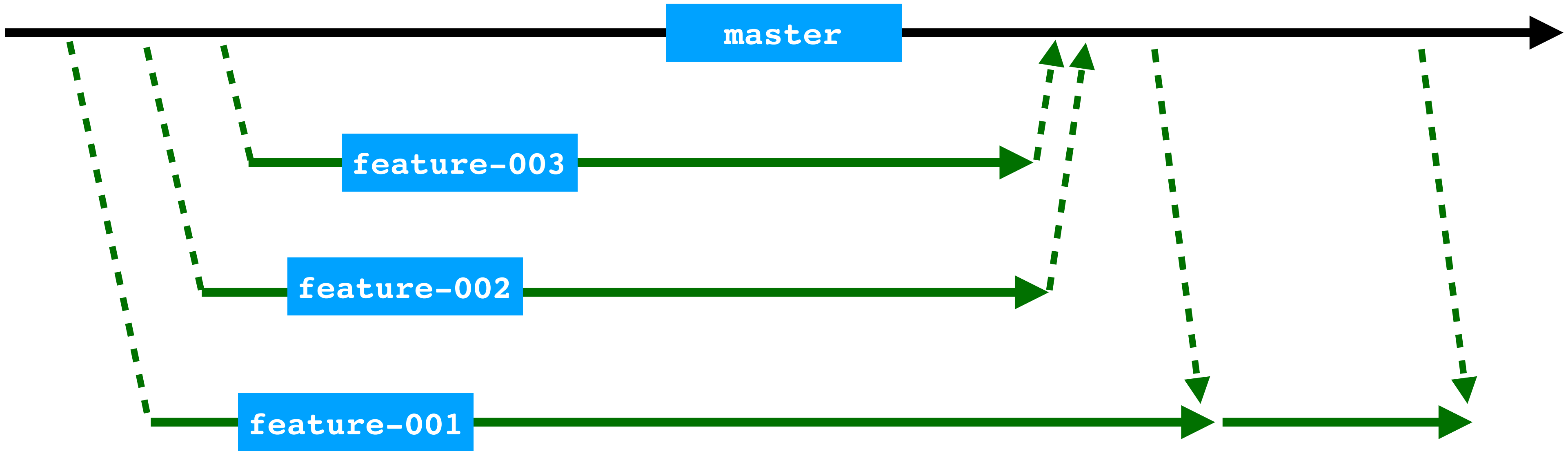




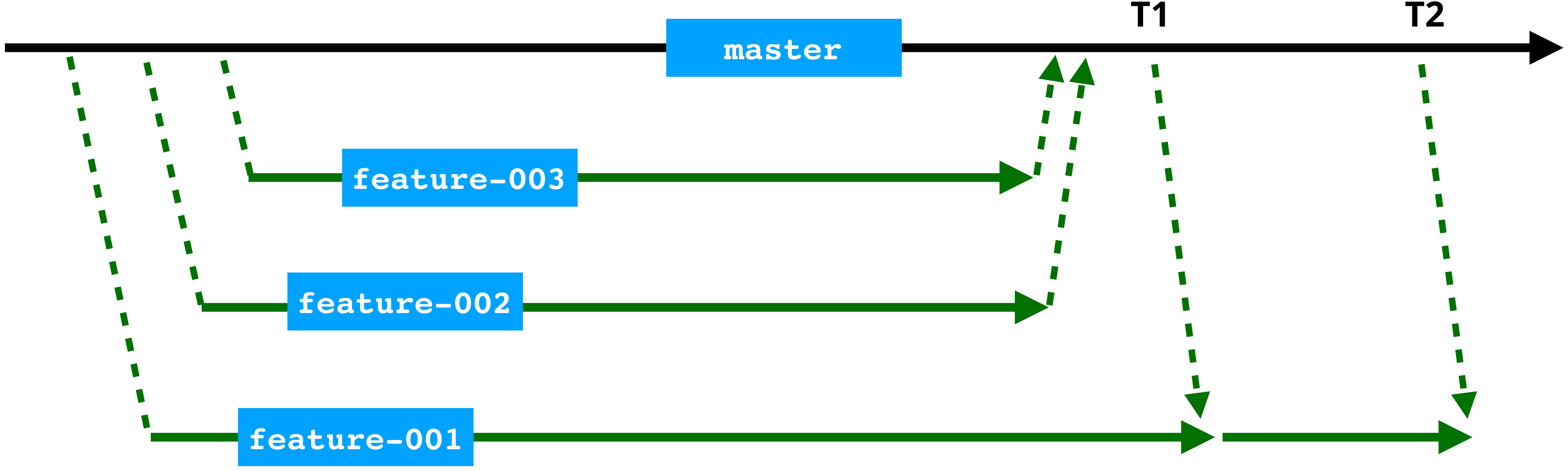


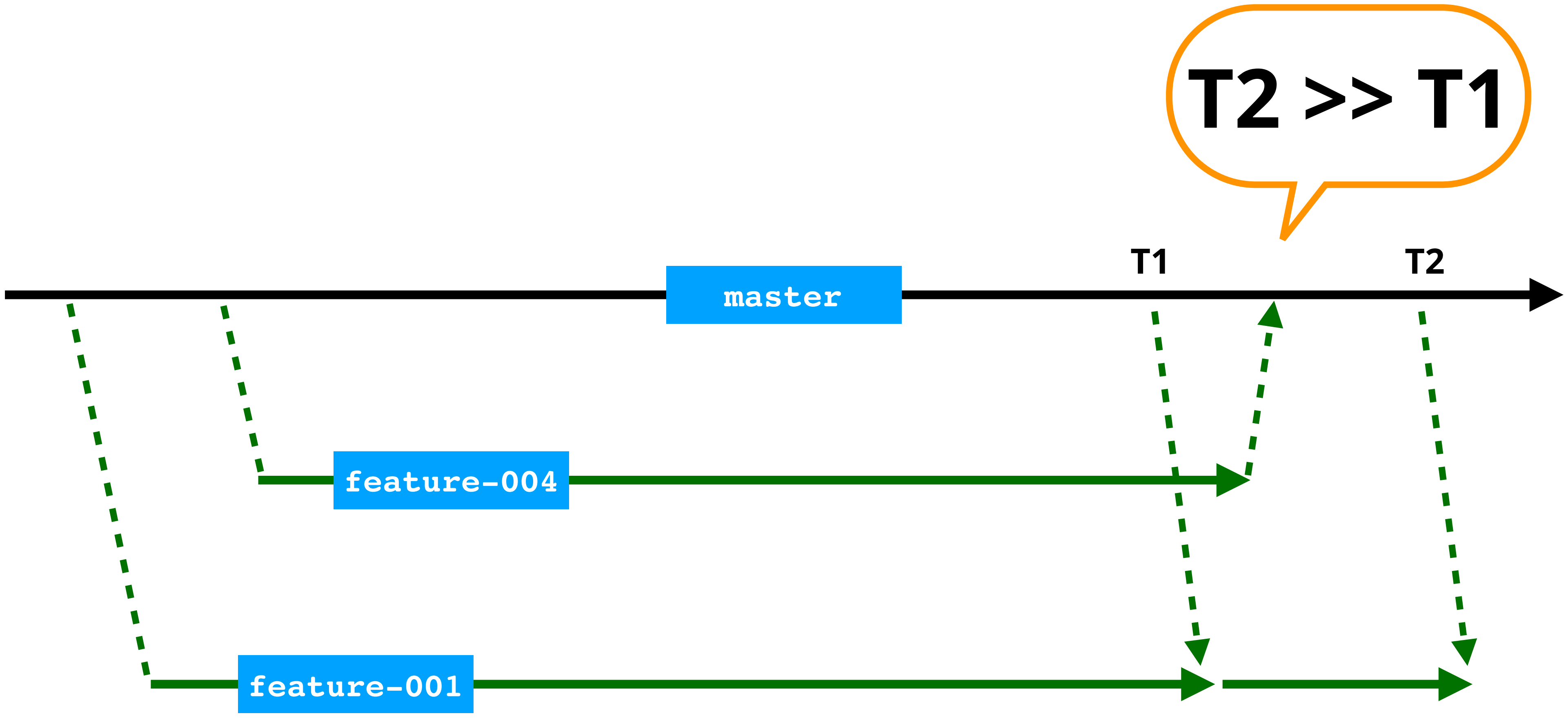




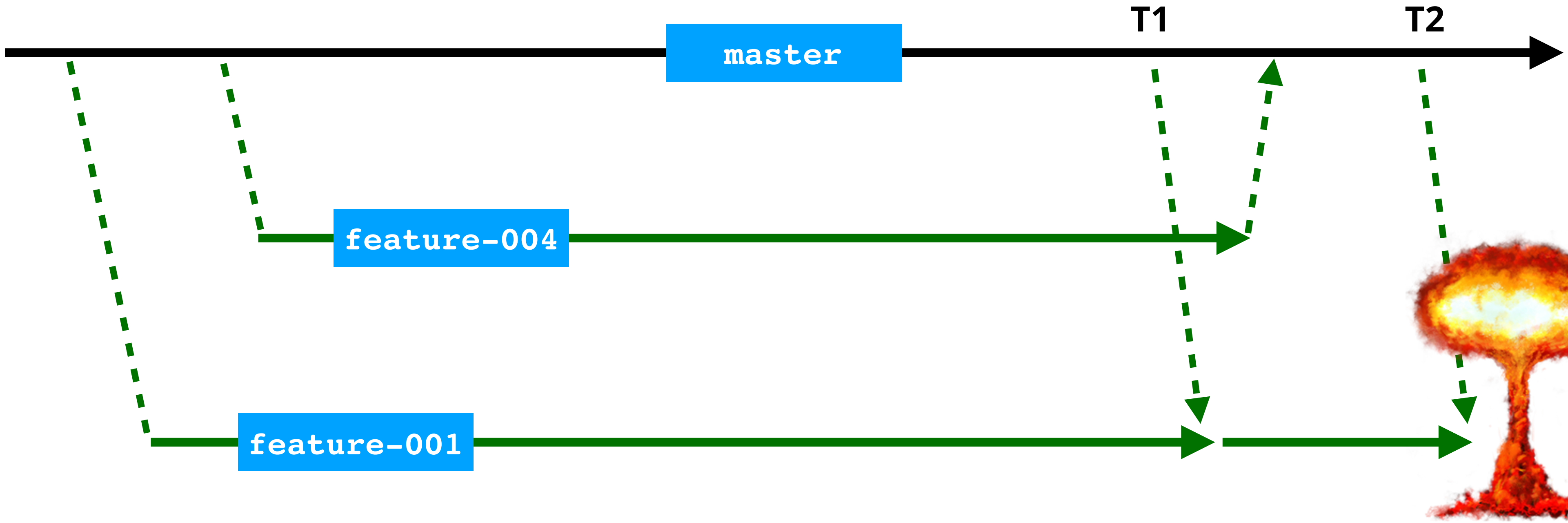


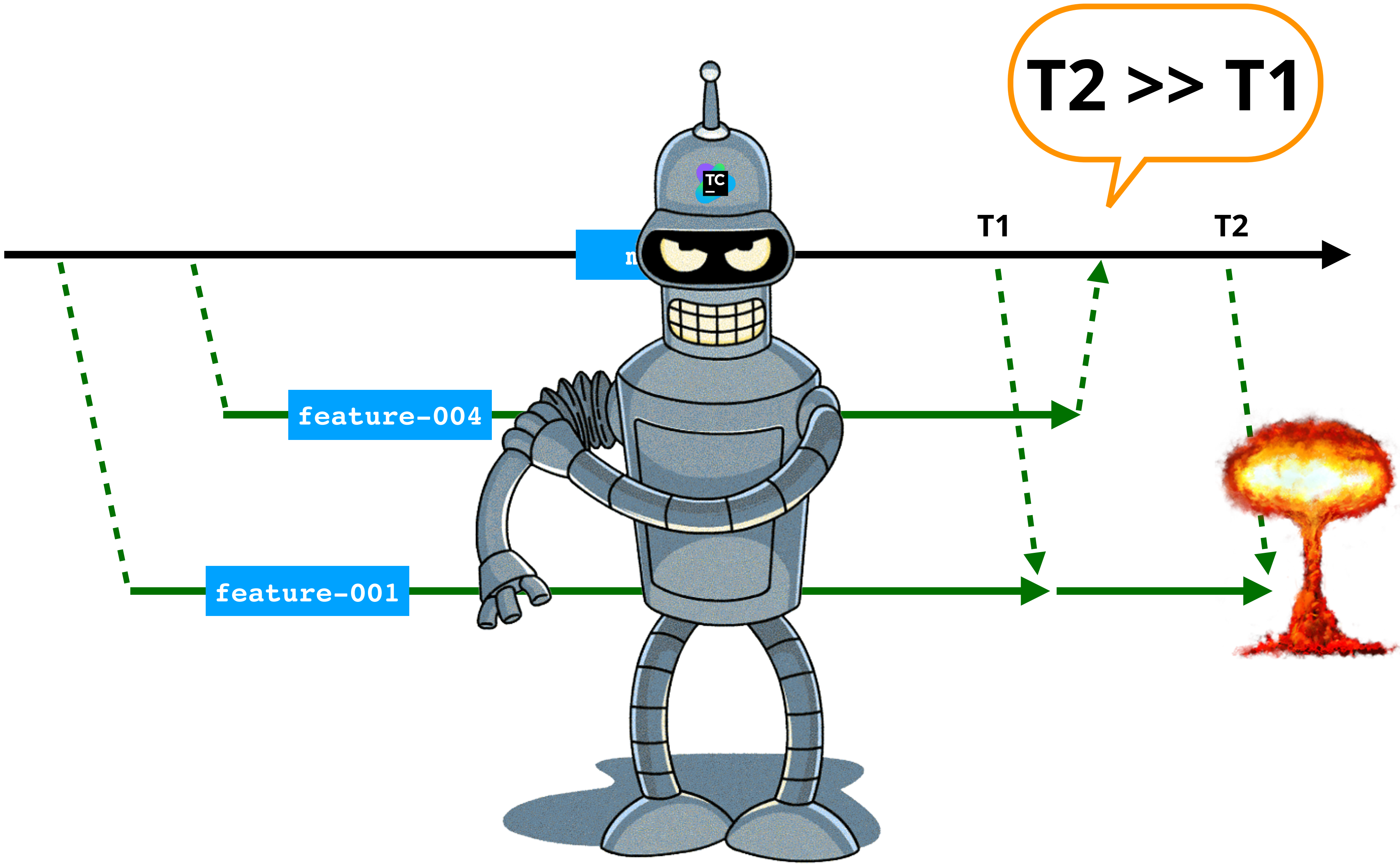
T2 >> T1





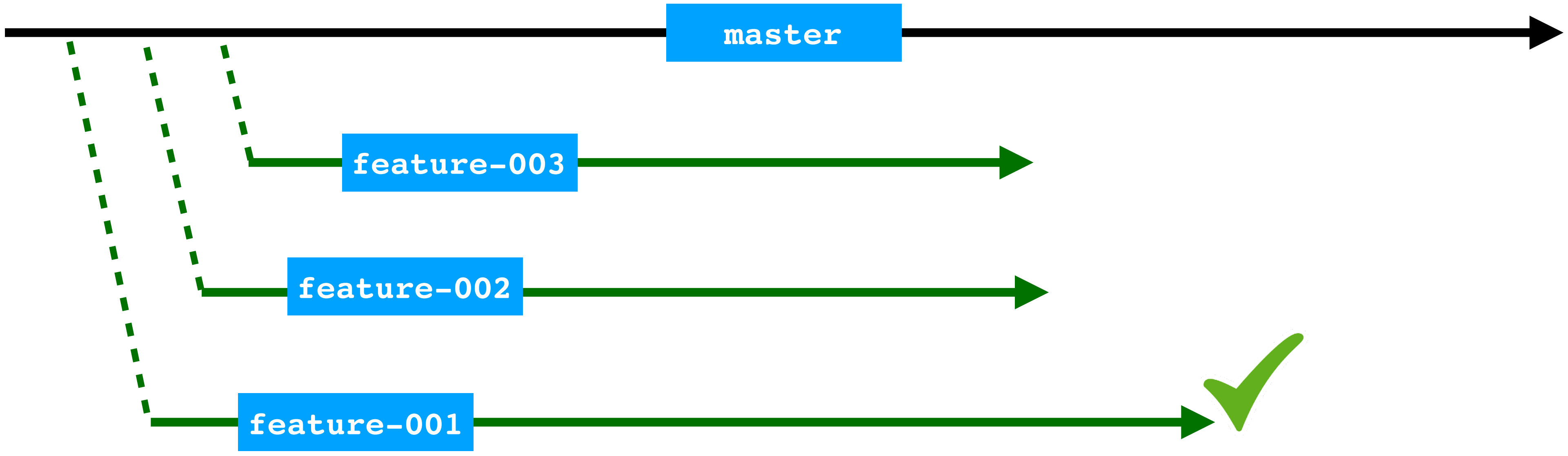
T2 >> T1

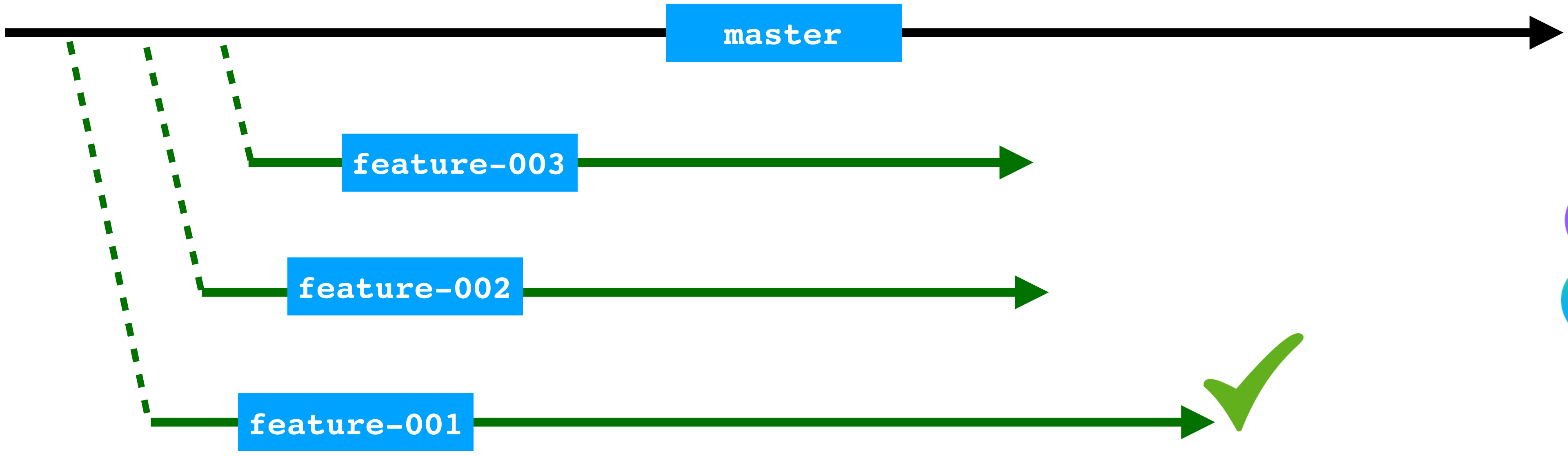


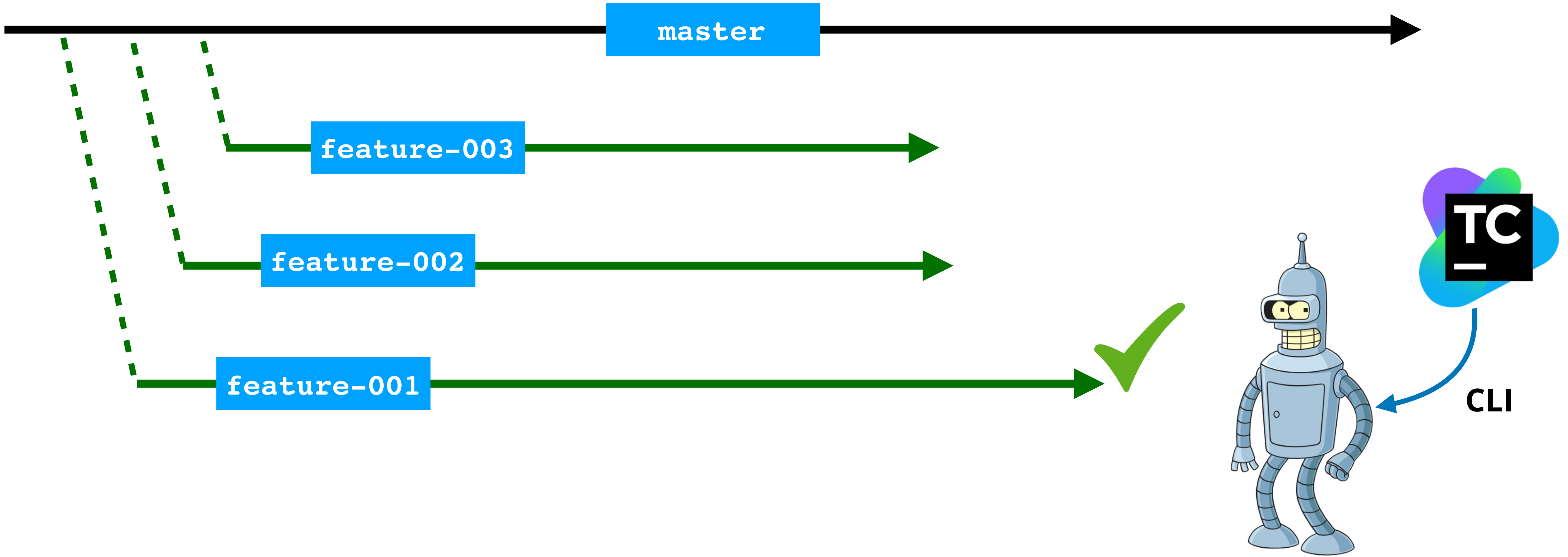


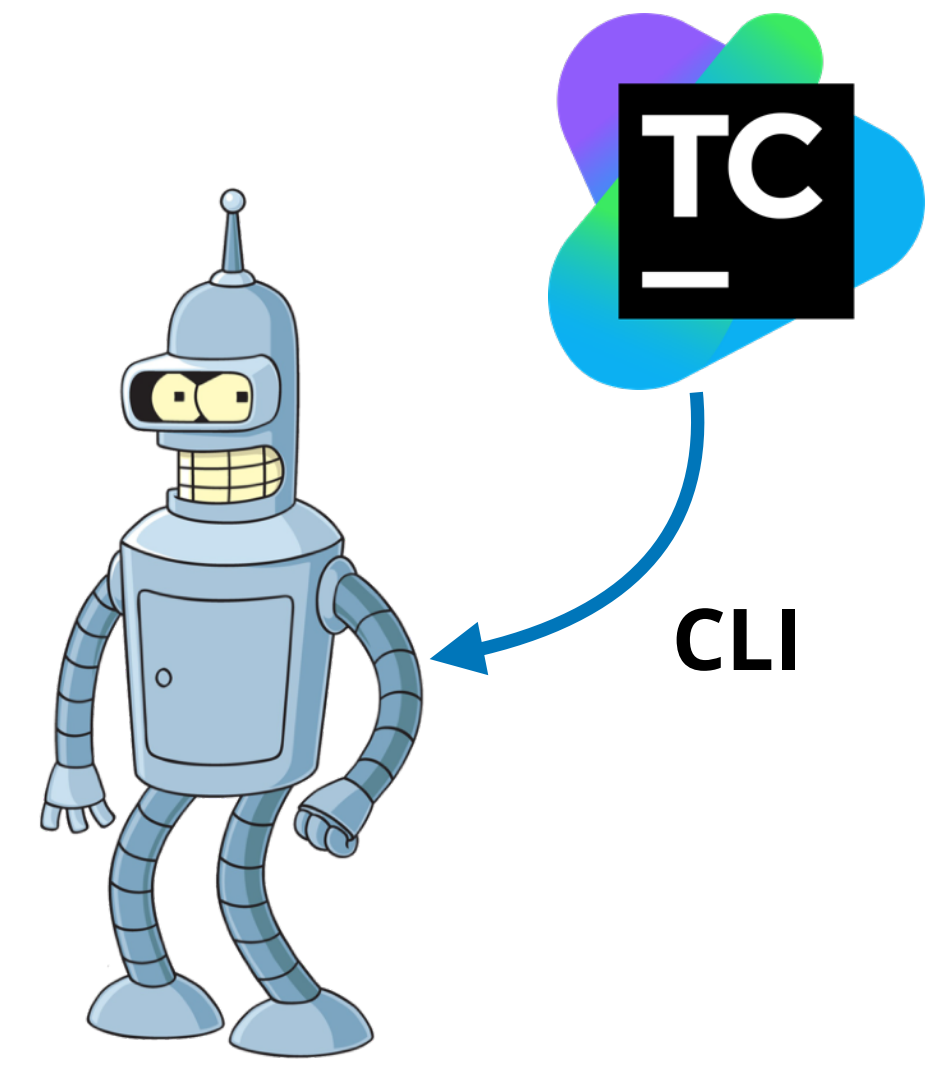
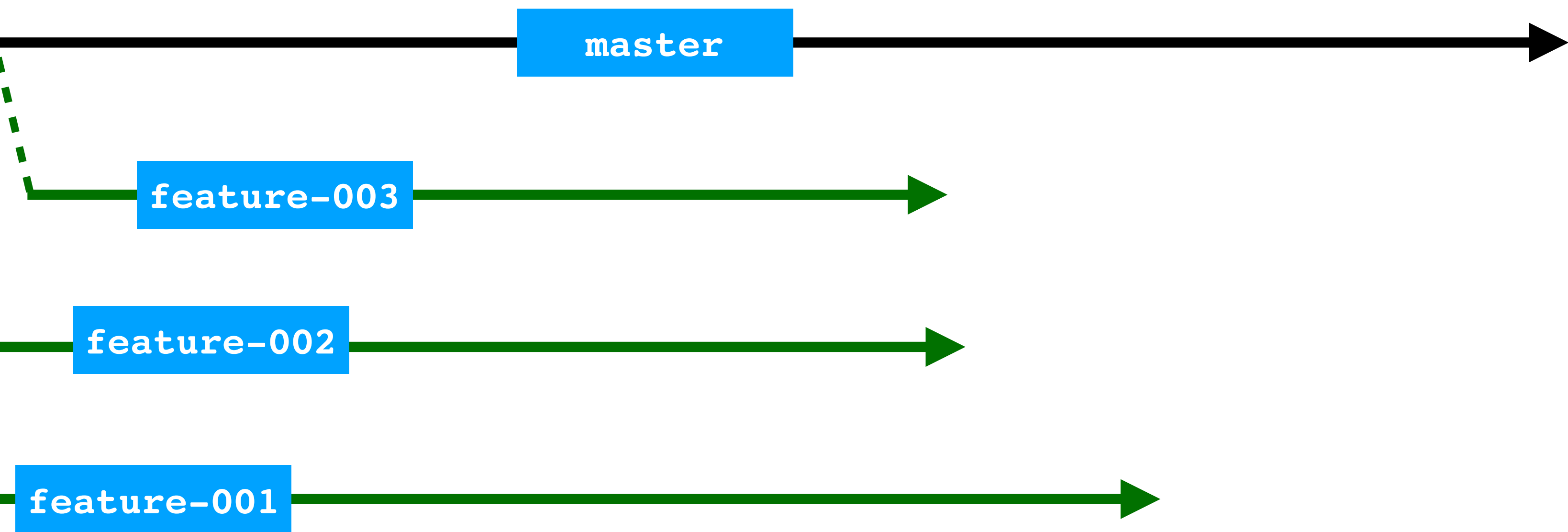
T2 >> T1

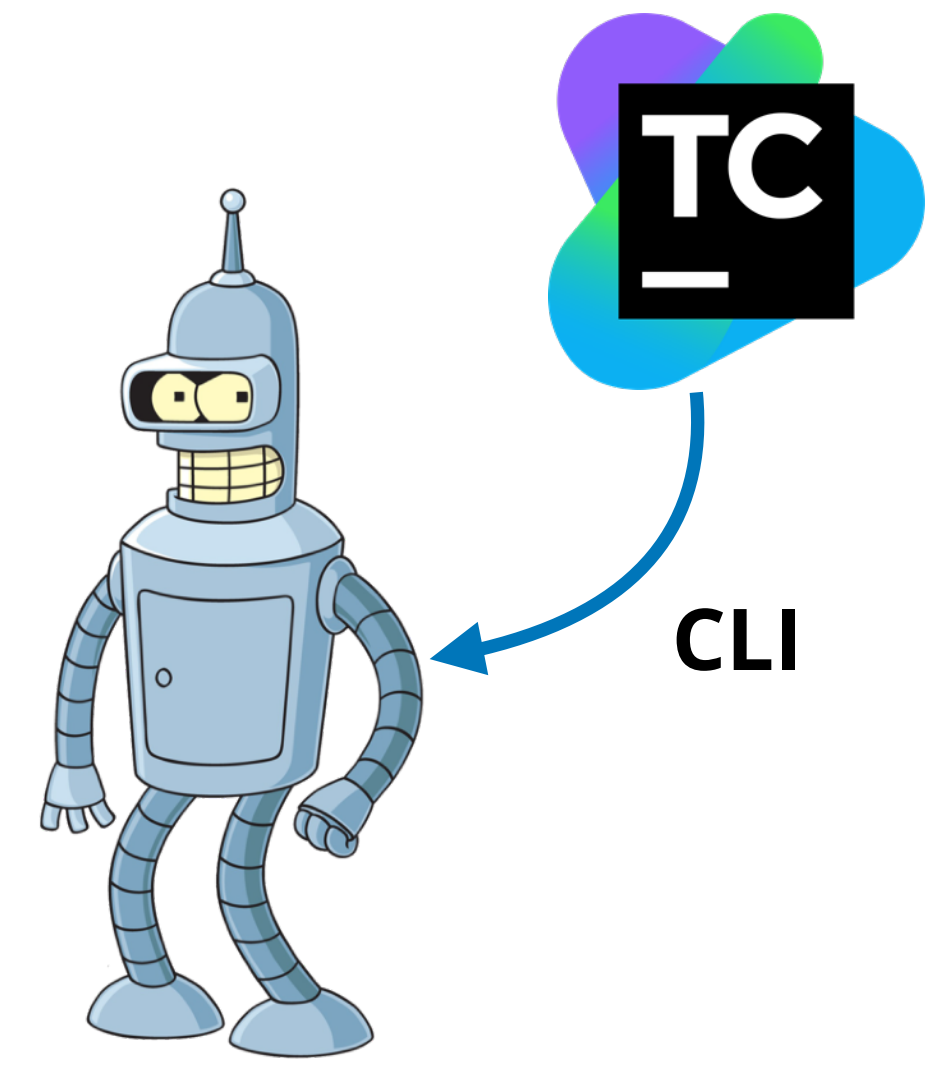
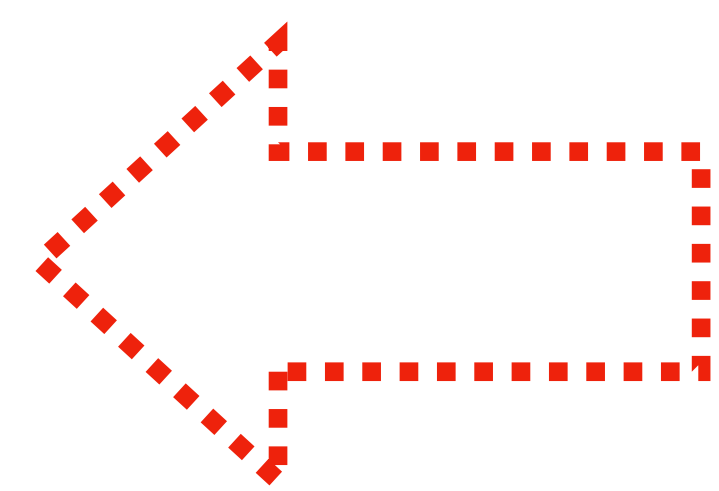
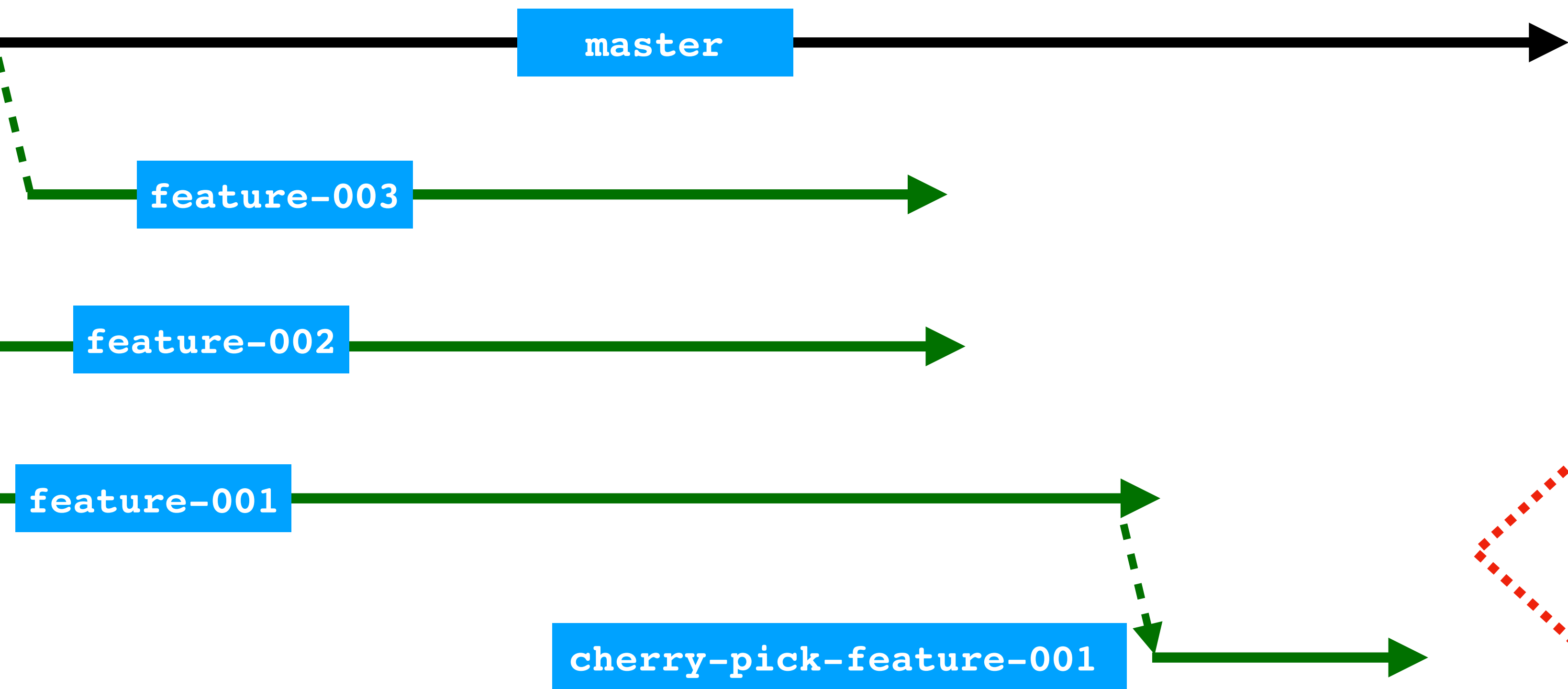
“merge robot”

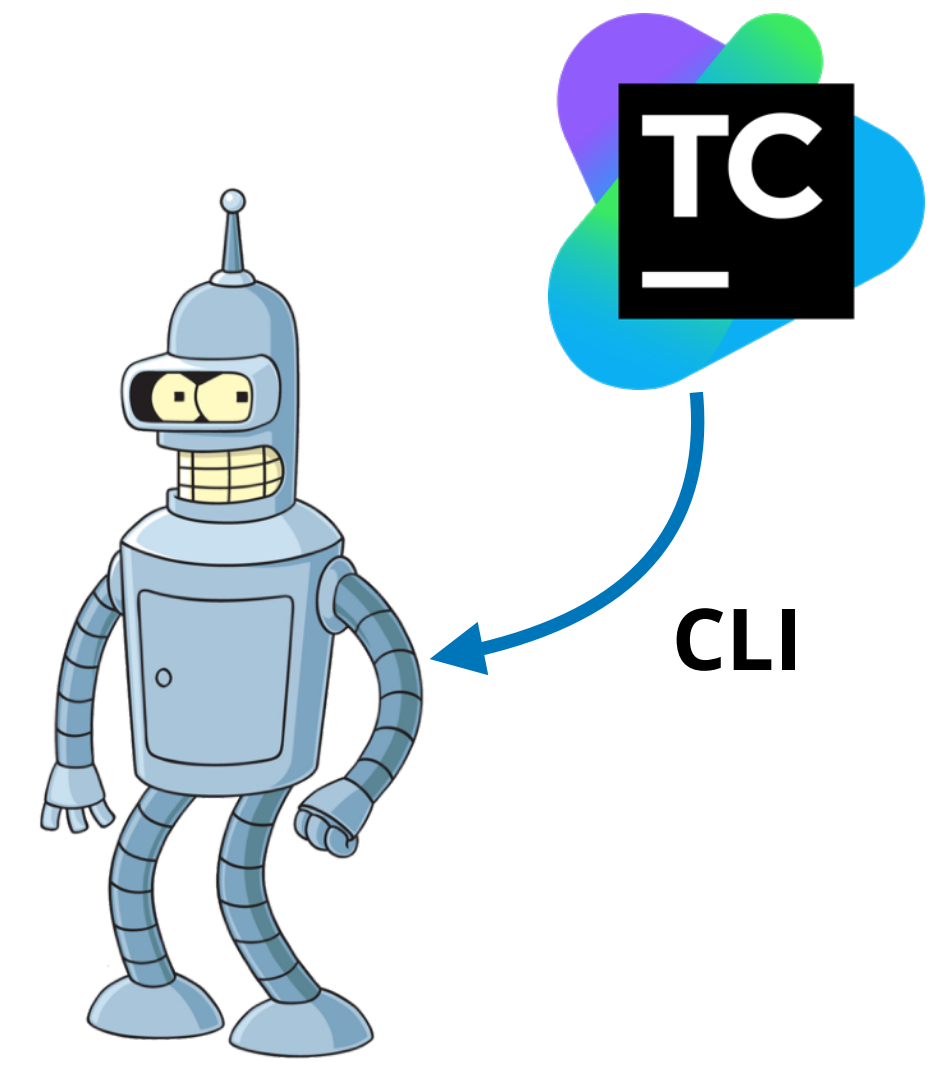
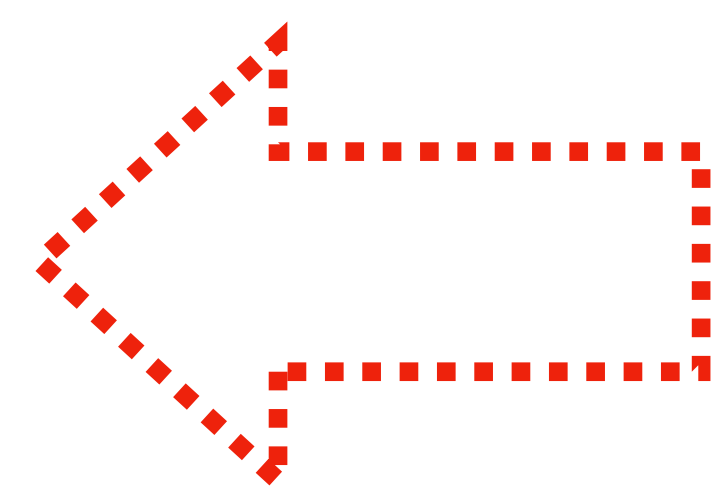
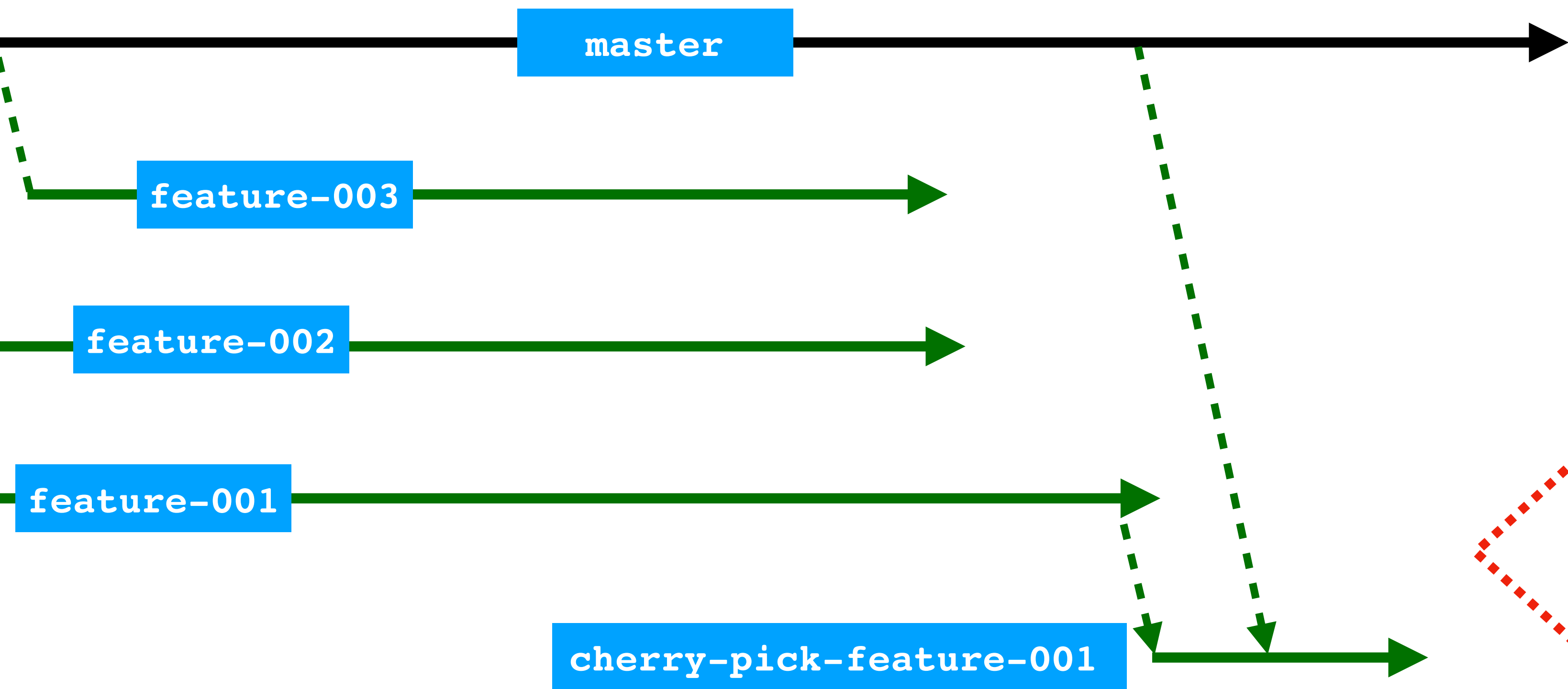


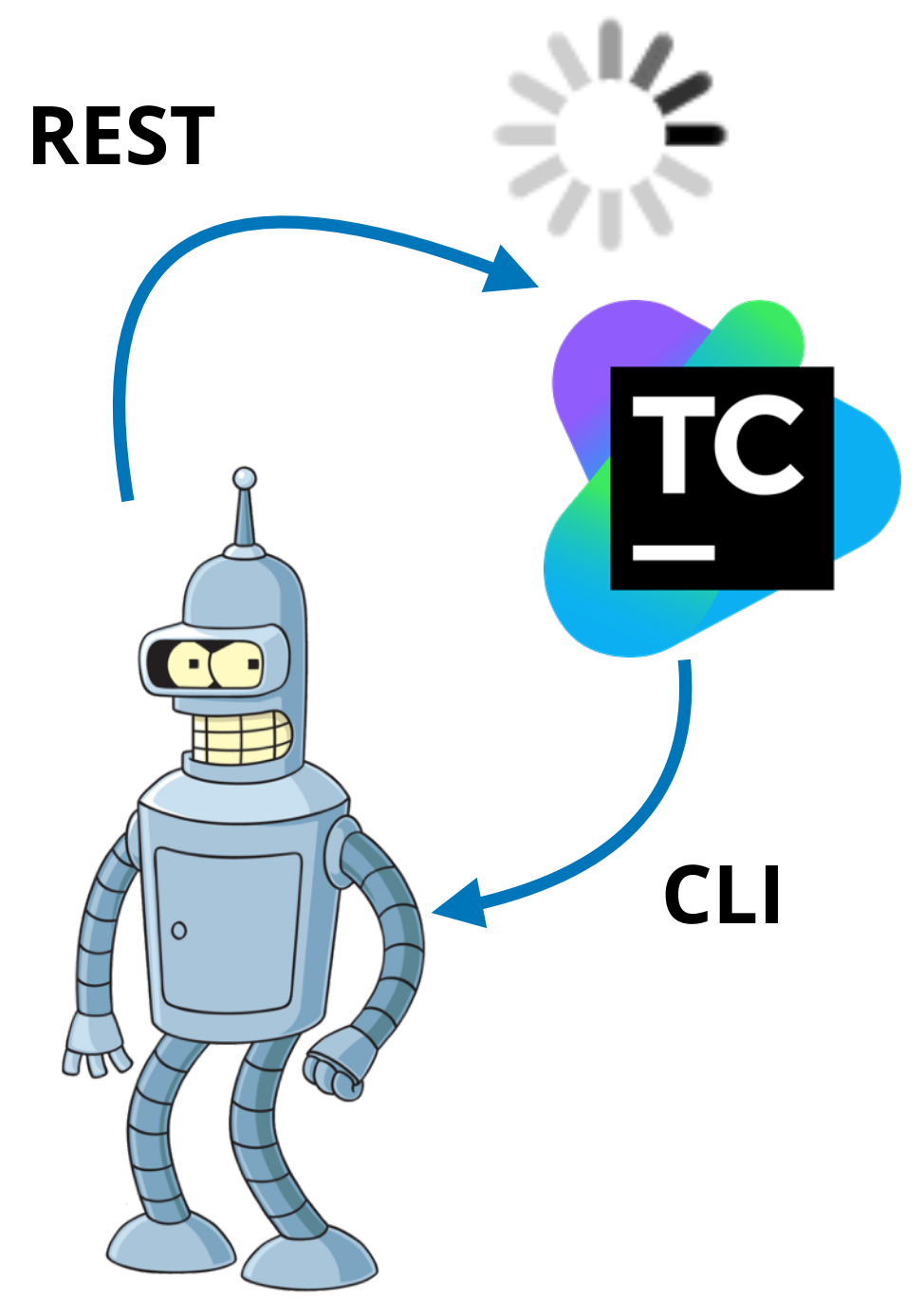
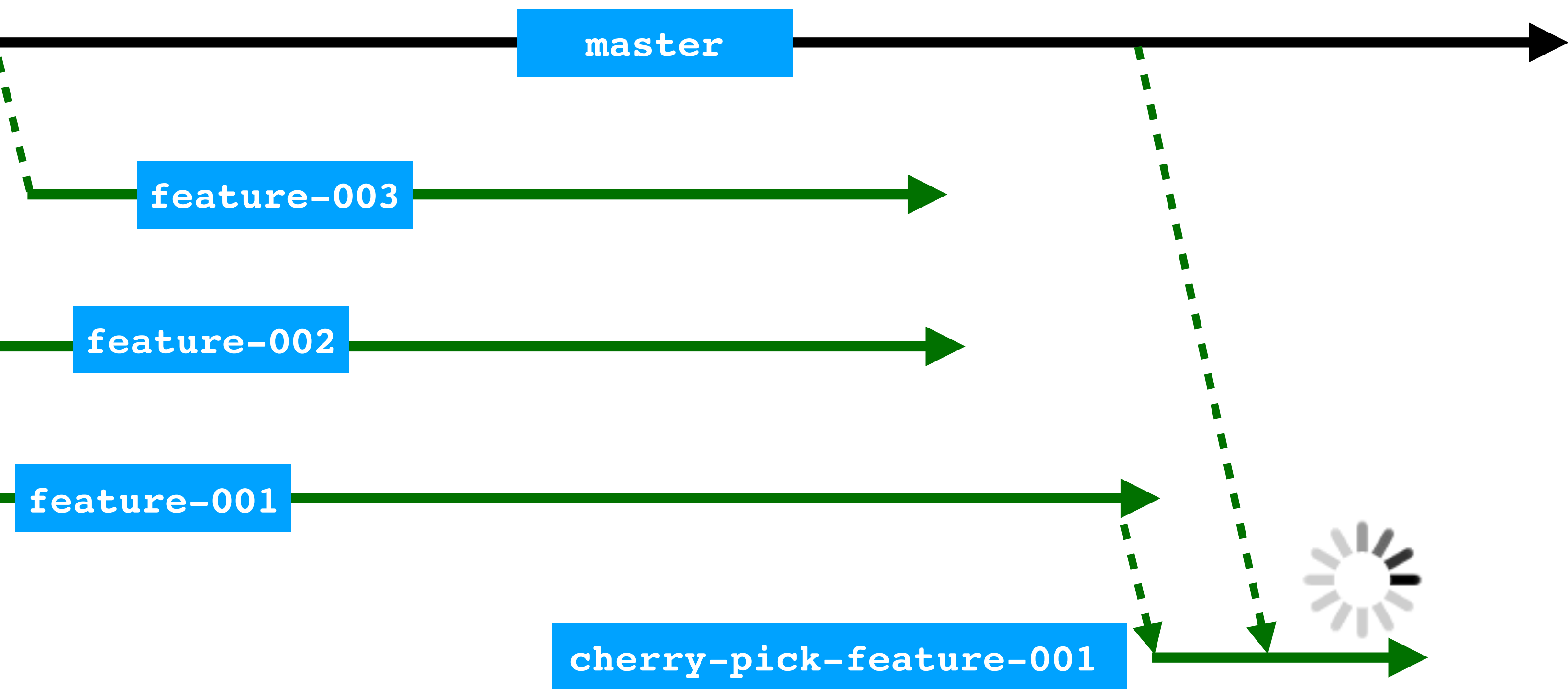


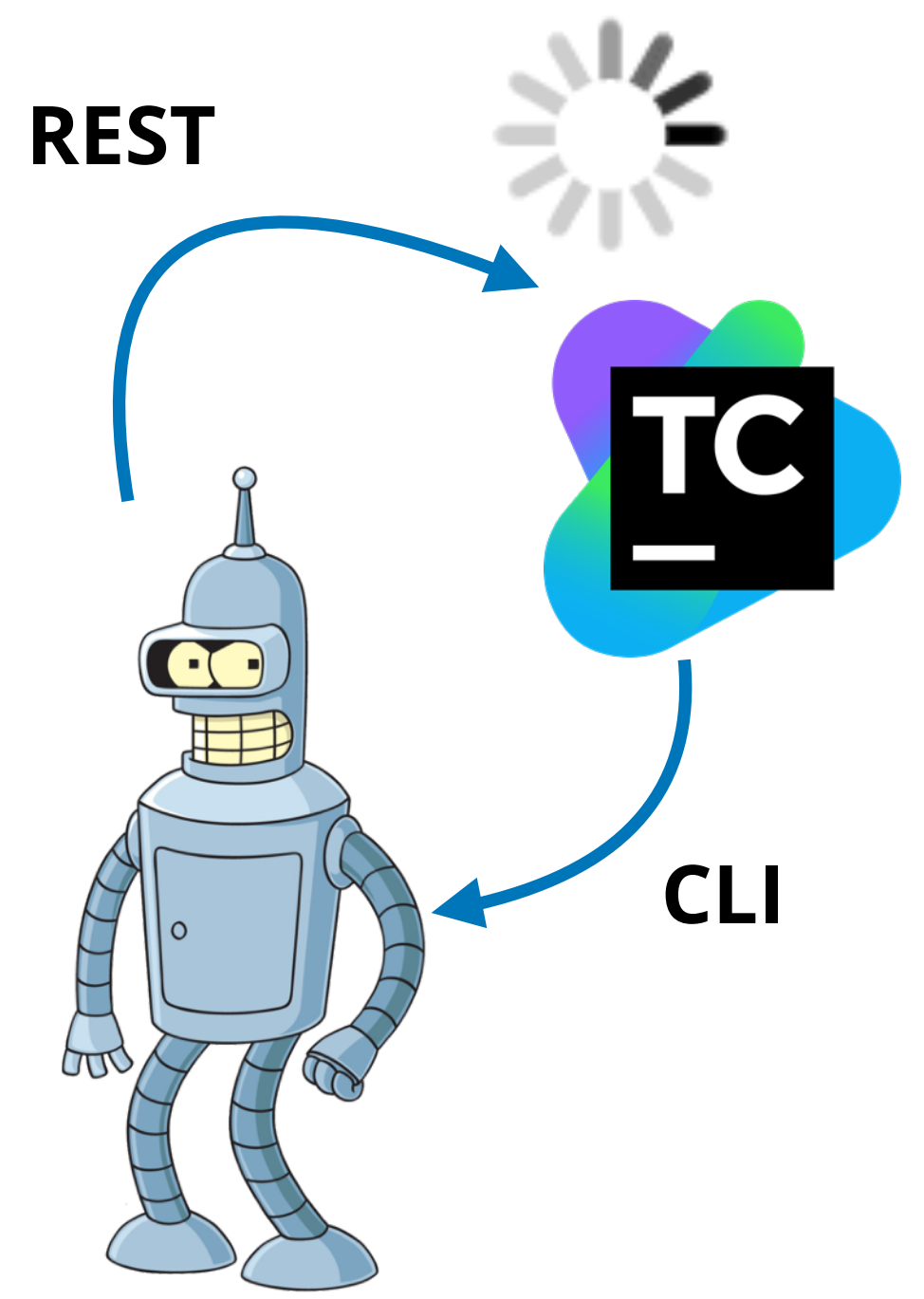
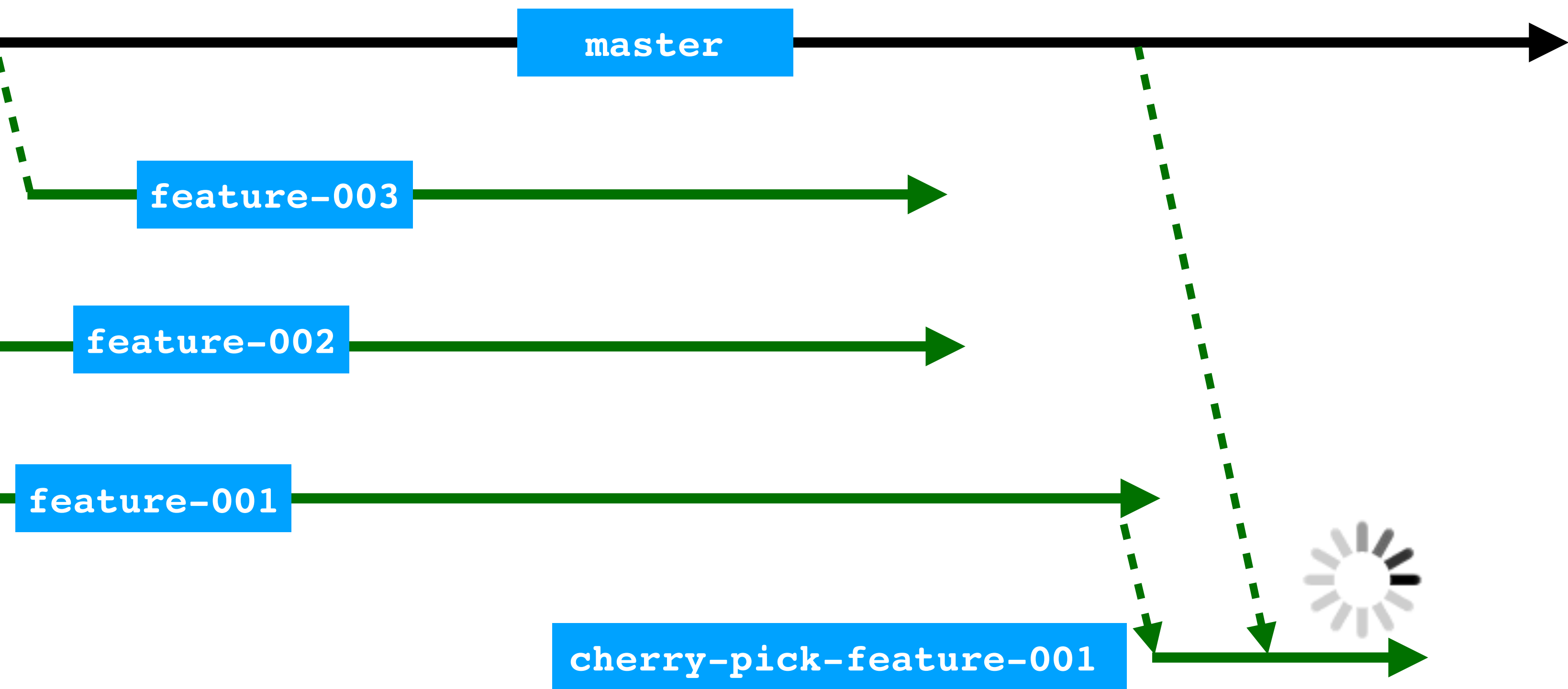


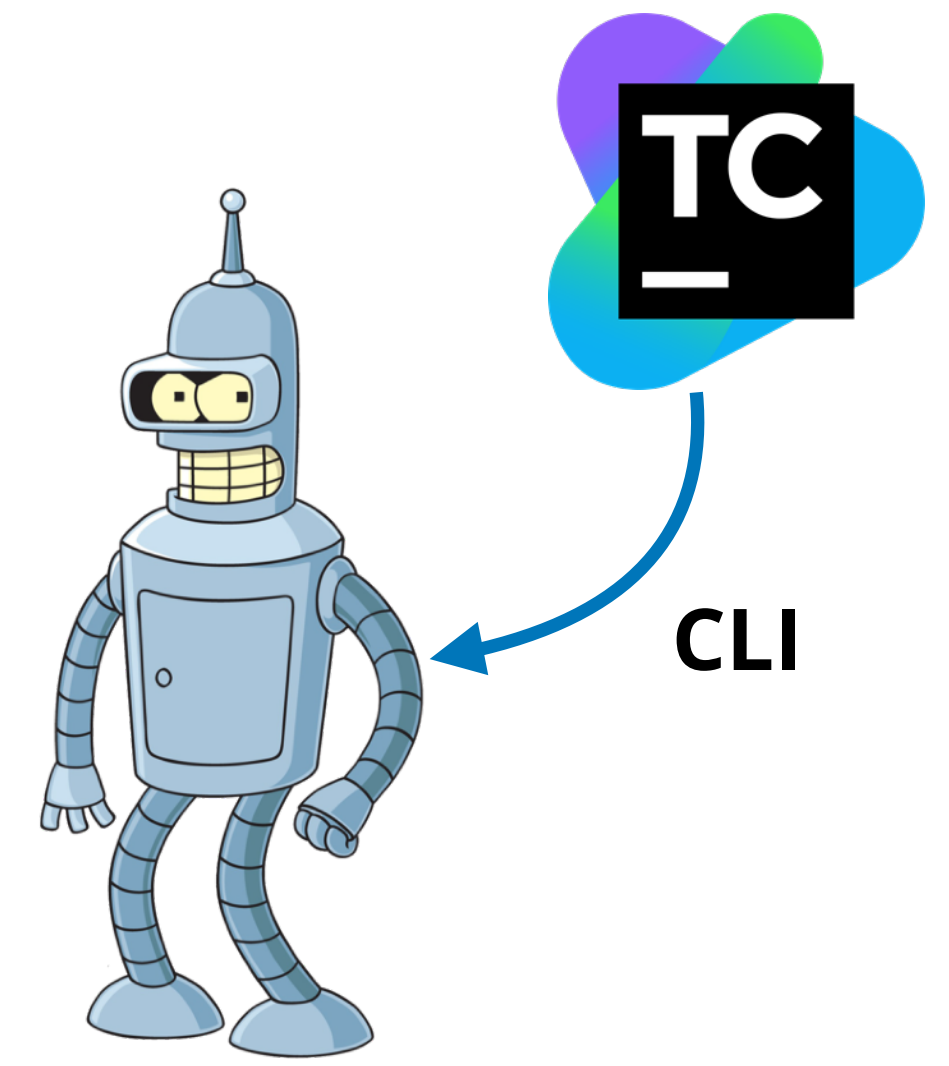
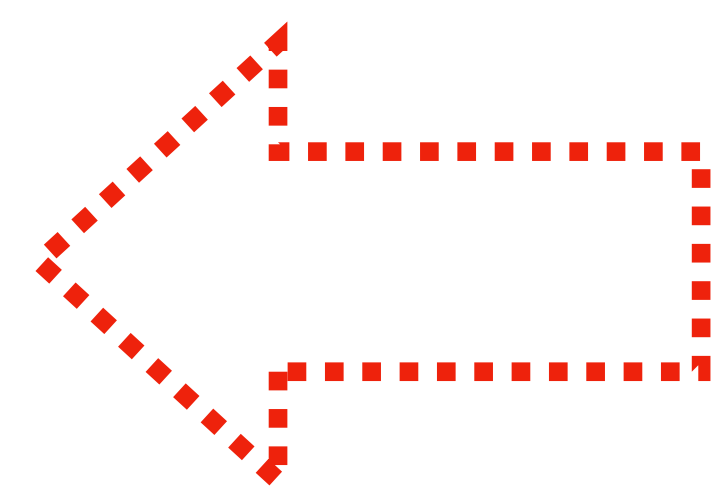
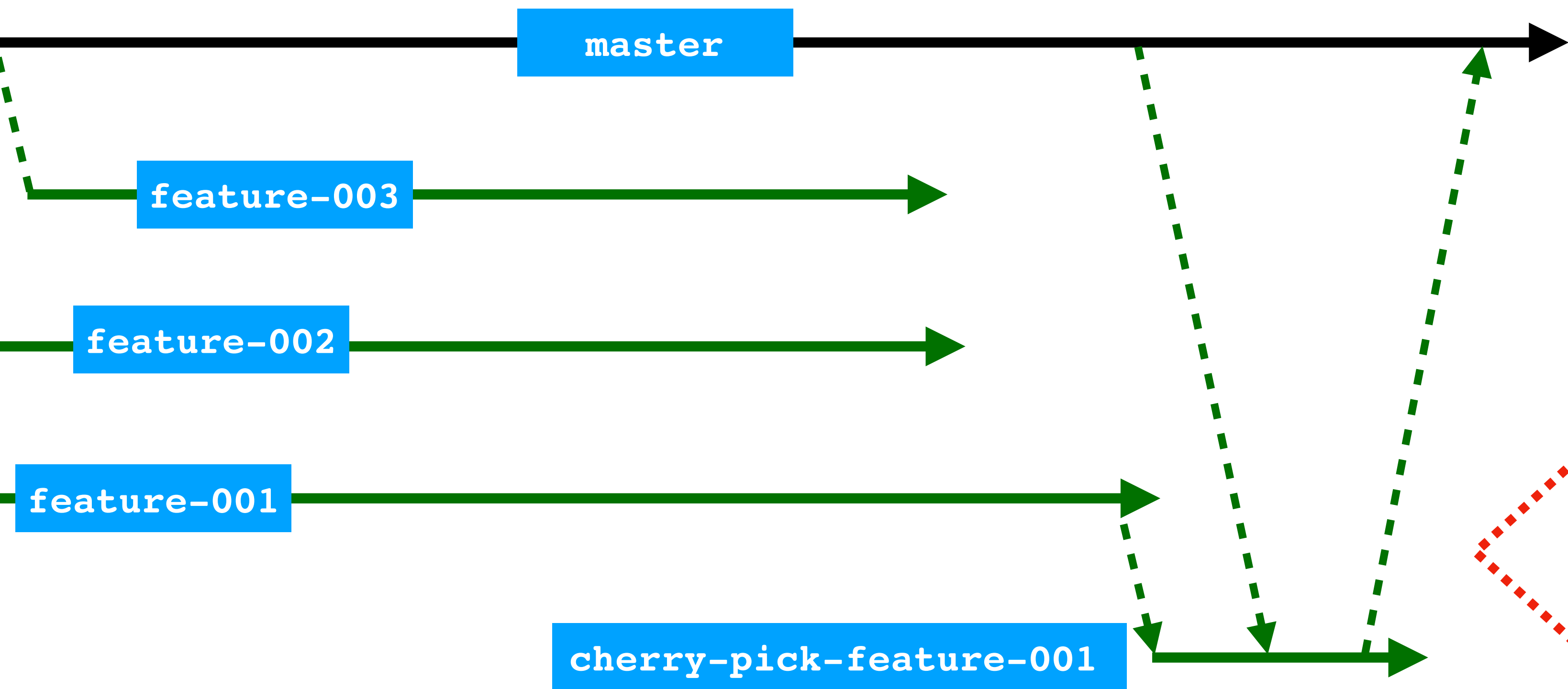


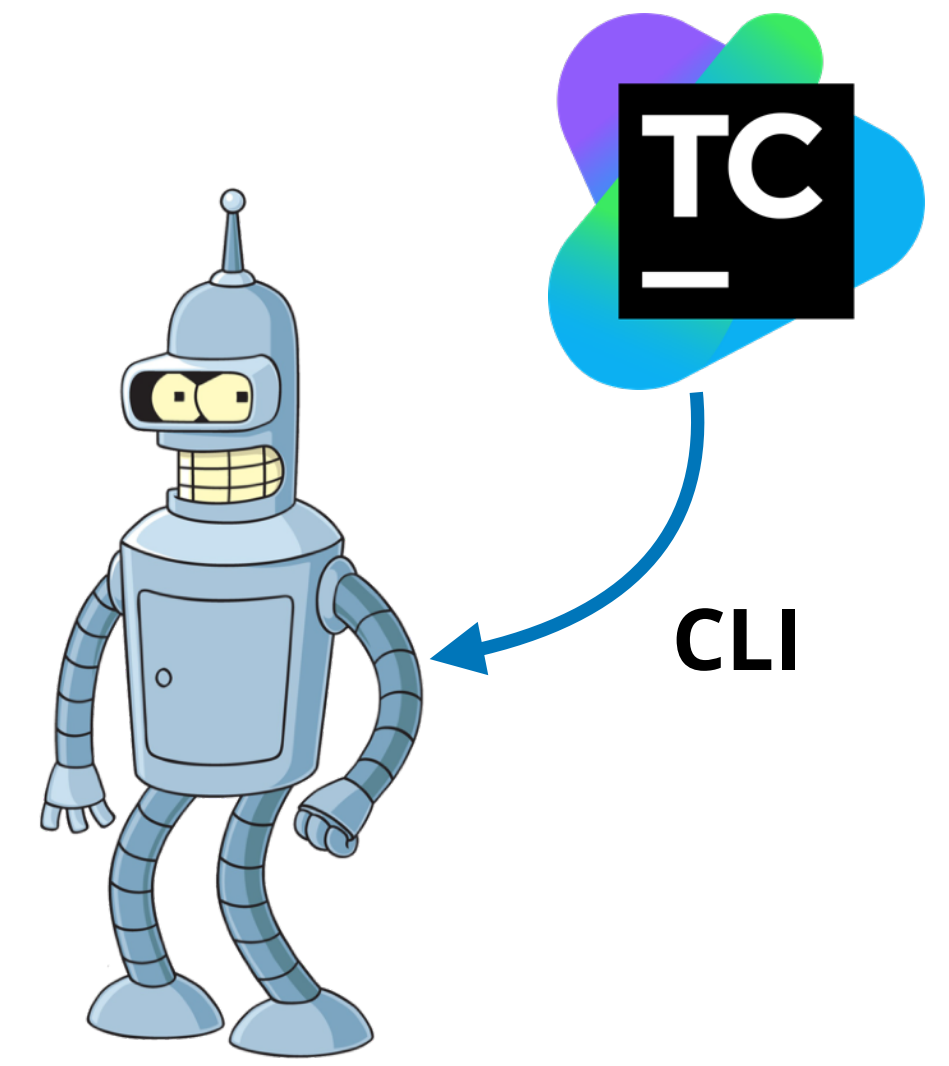
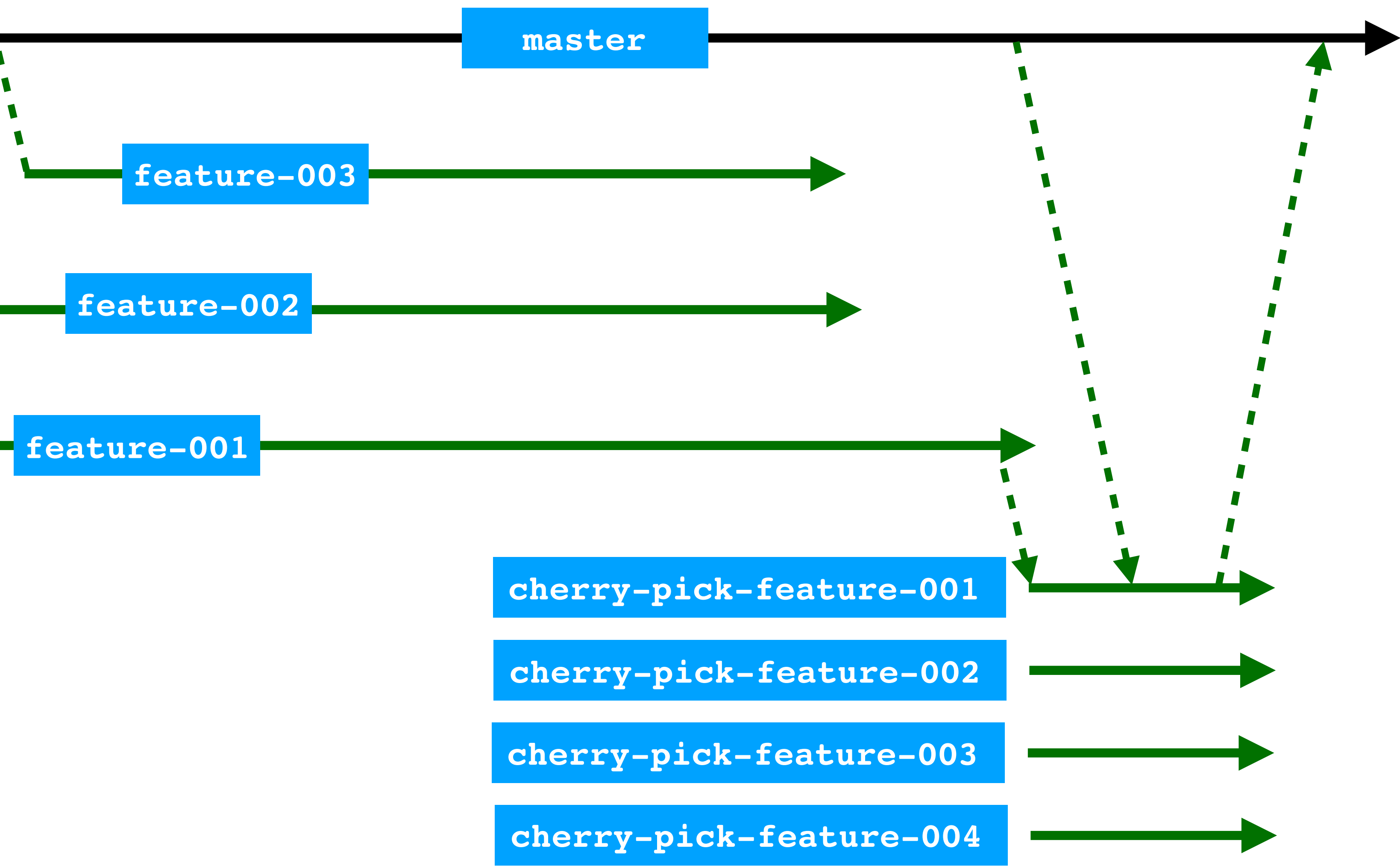




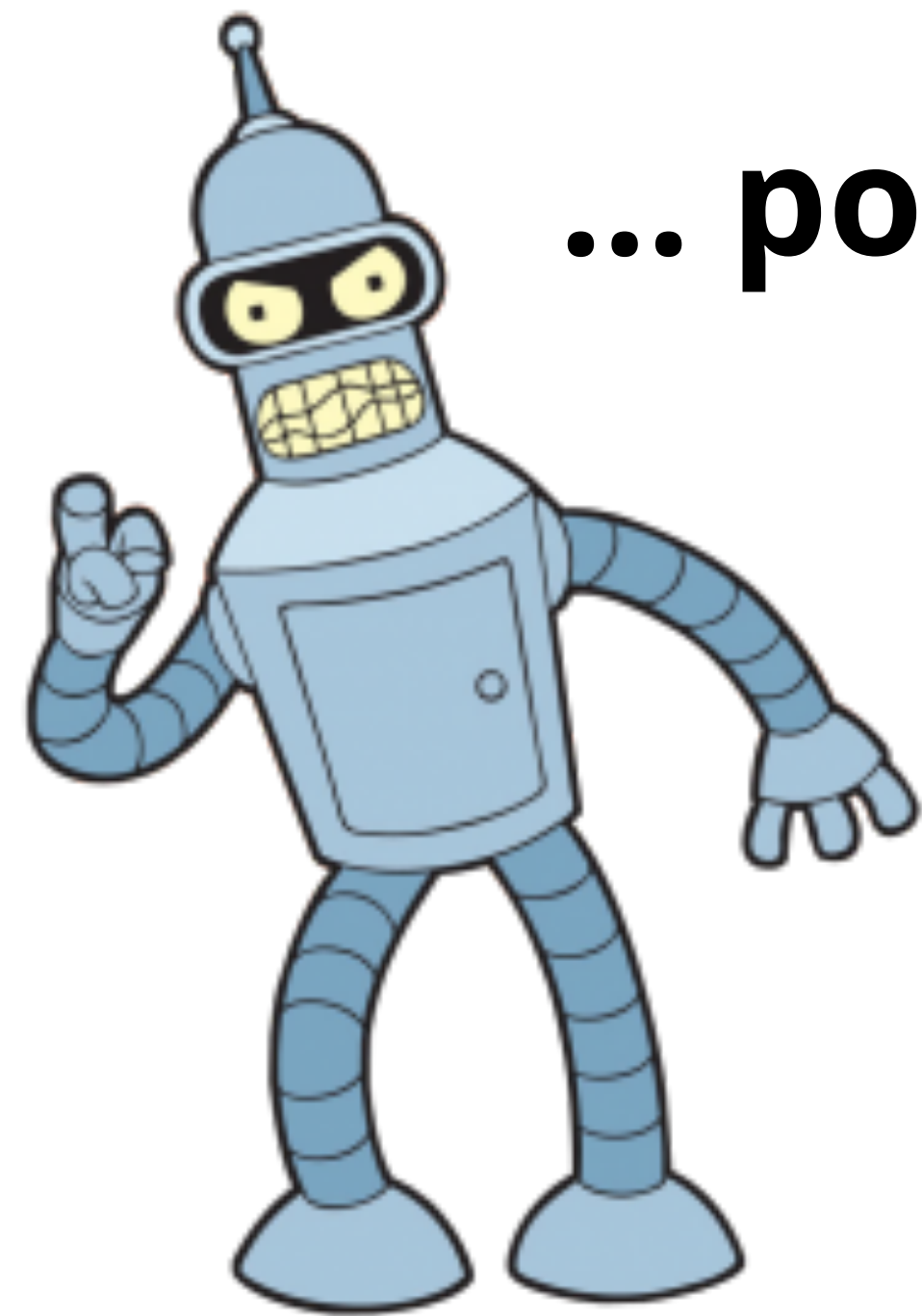




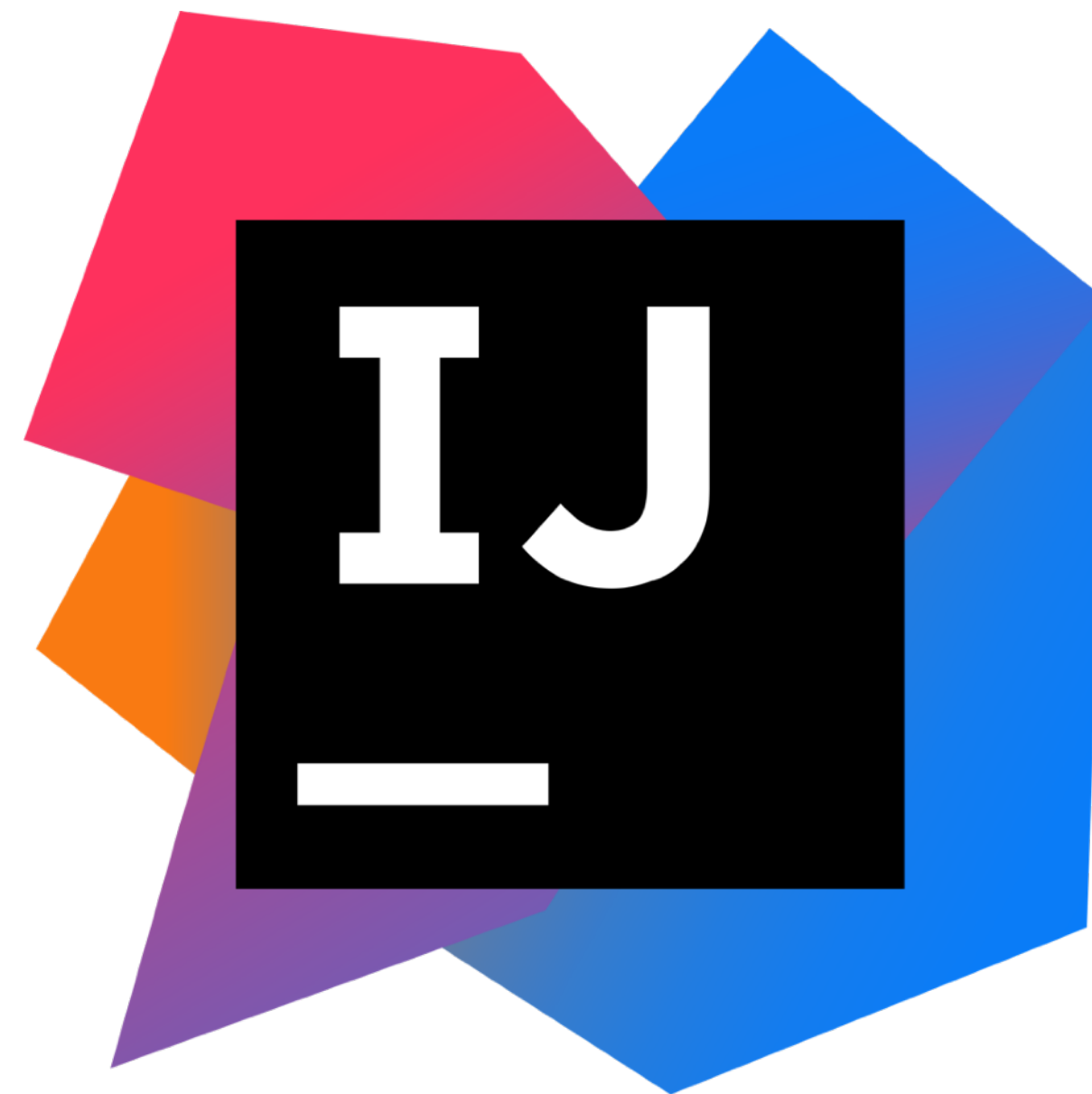




Цель достигнута, мастер зелёный, но...



... робот может стать “бутылочным горлышком”



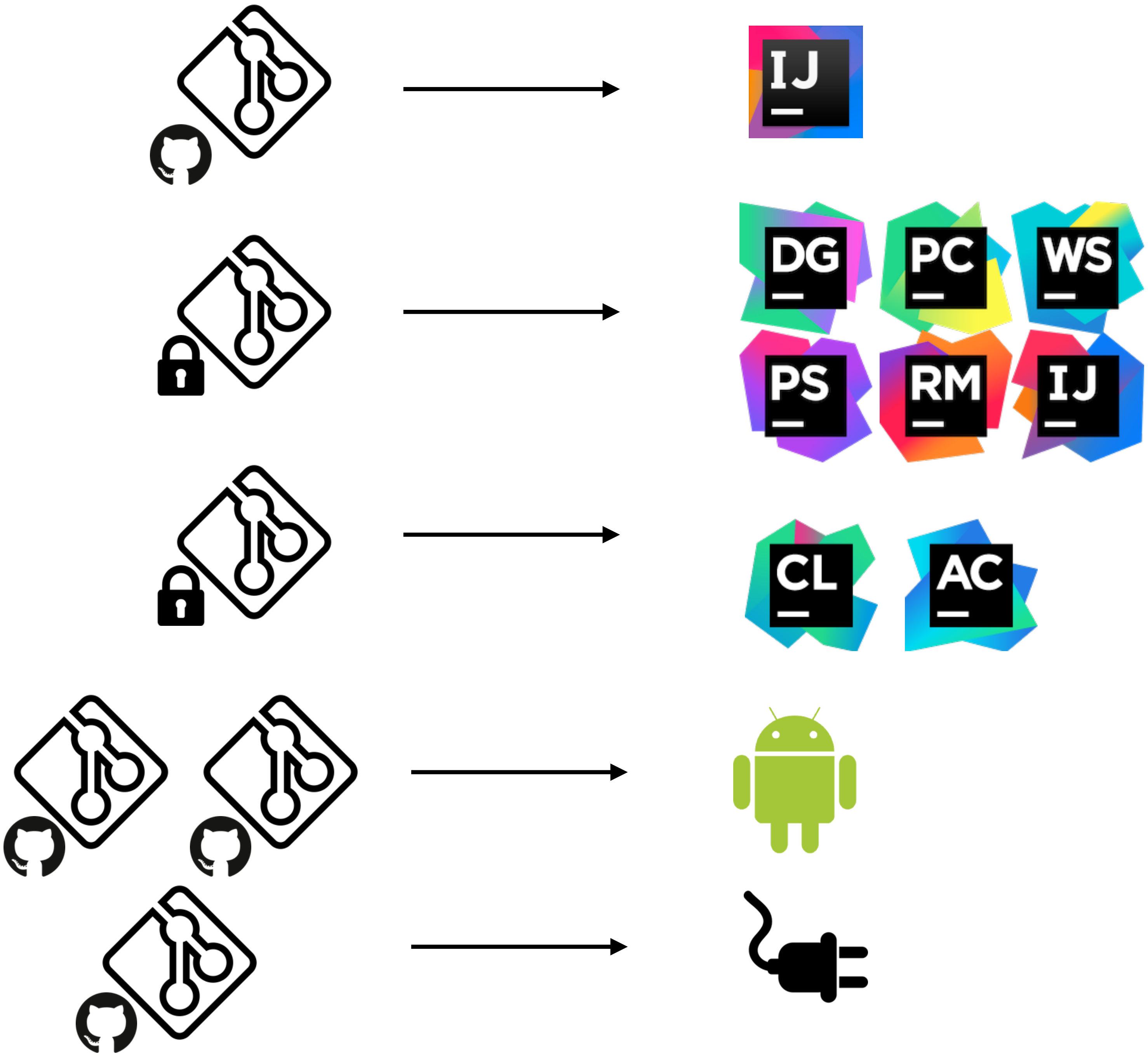
IntelliJ IDEA

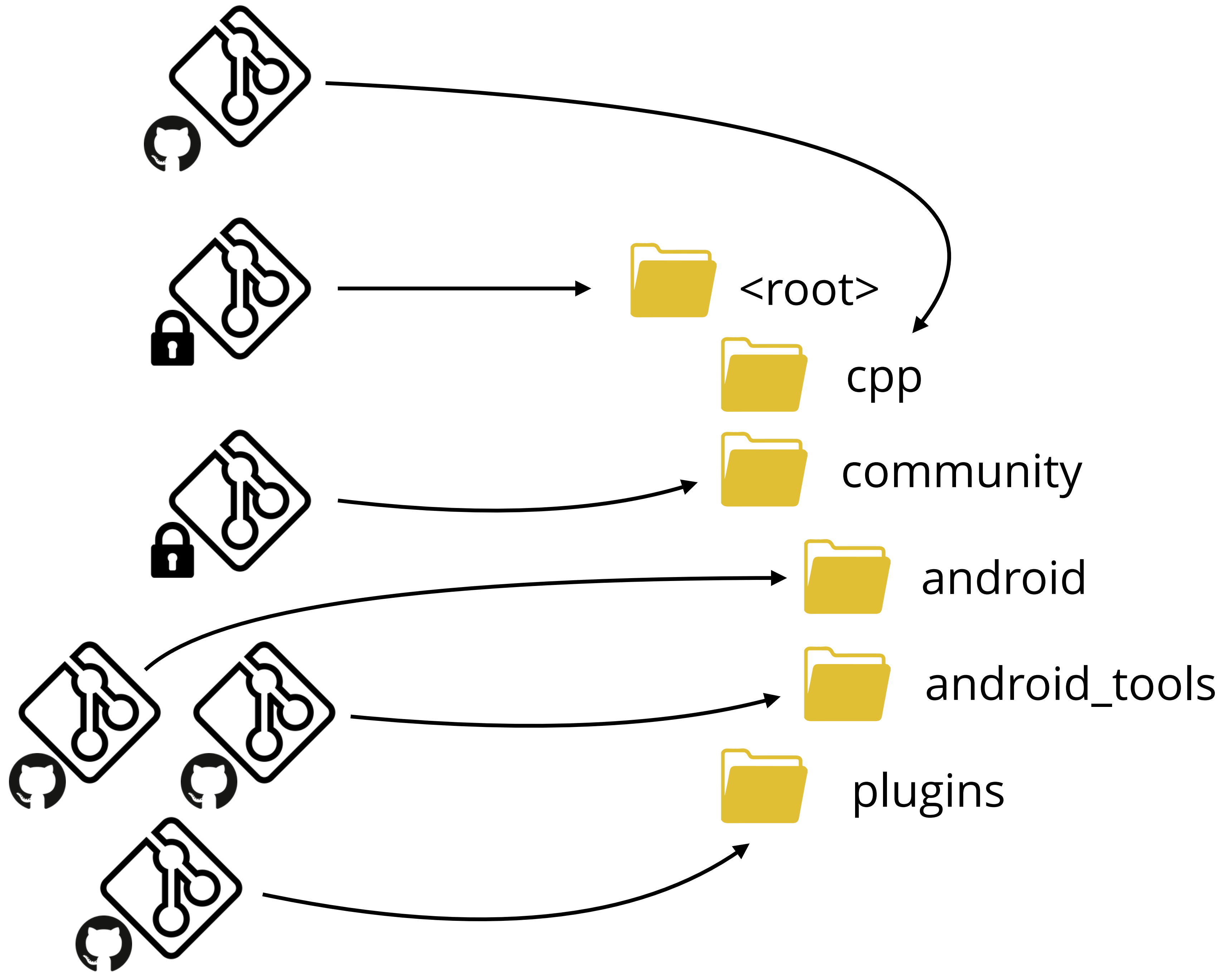
600+ thousand commits

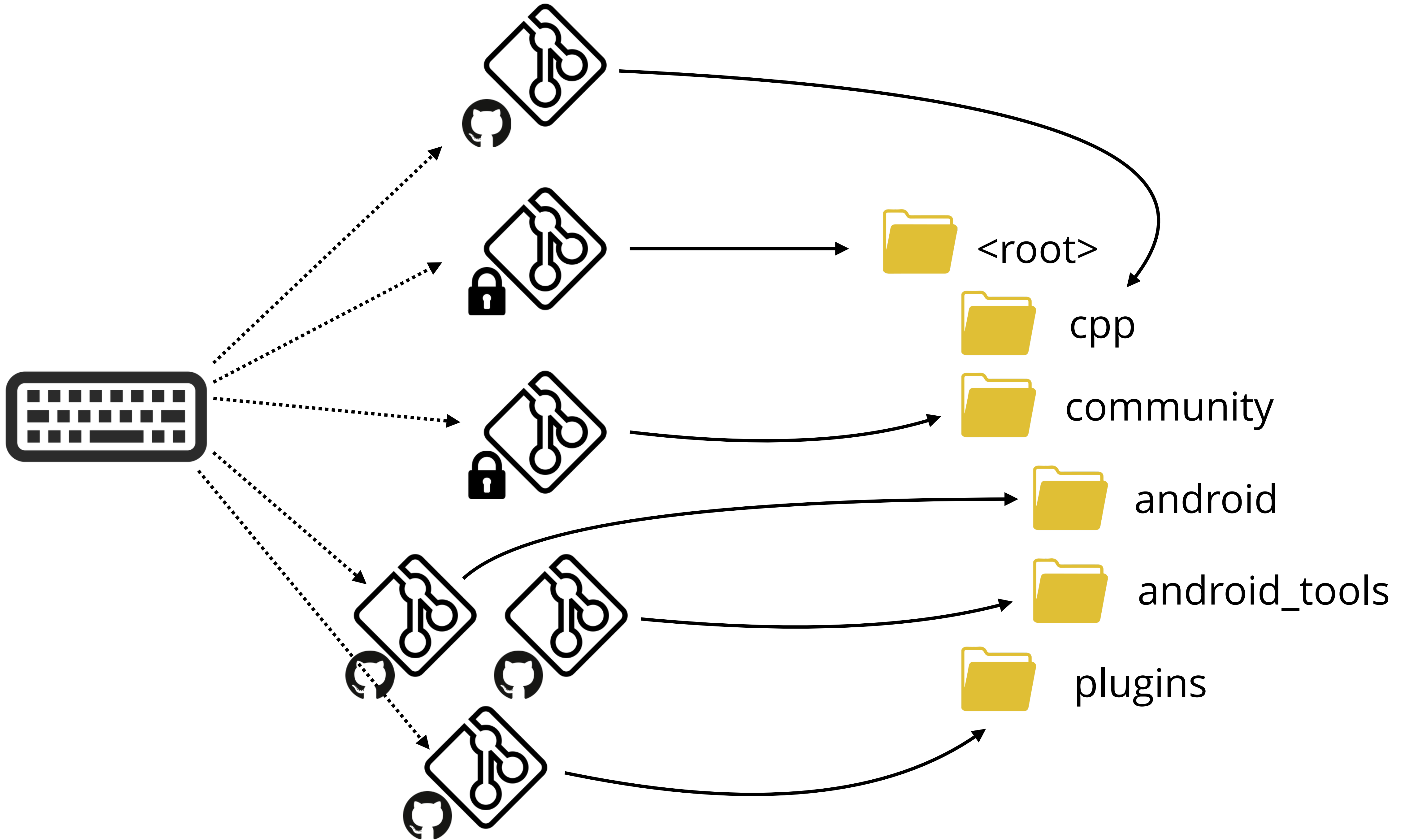
13+ million lines of code

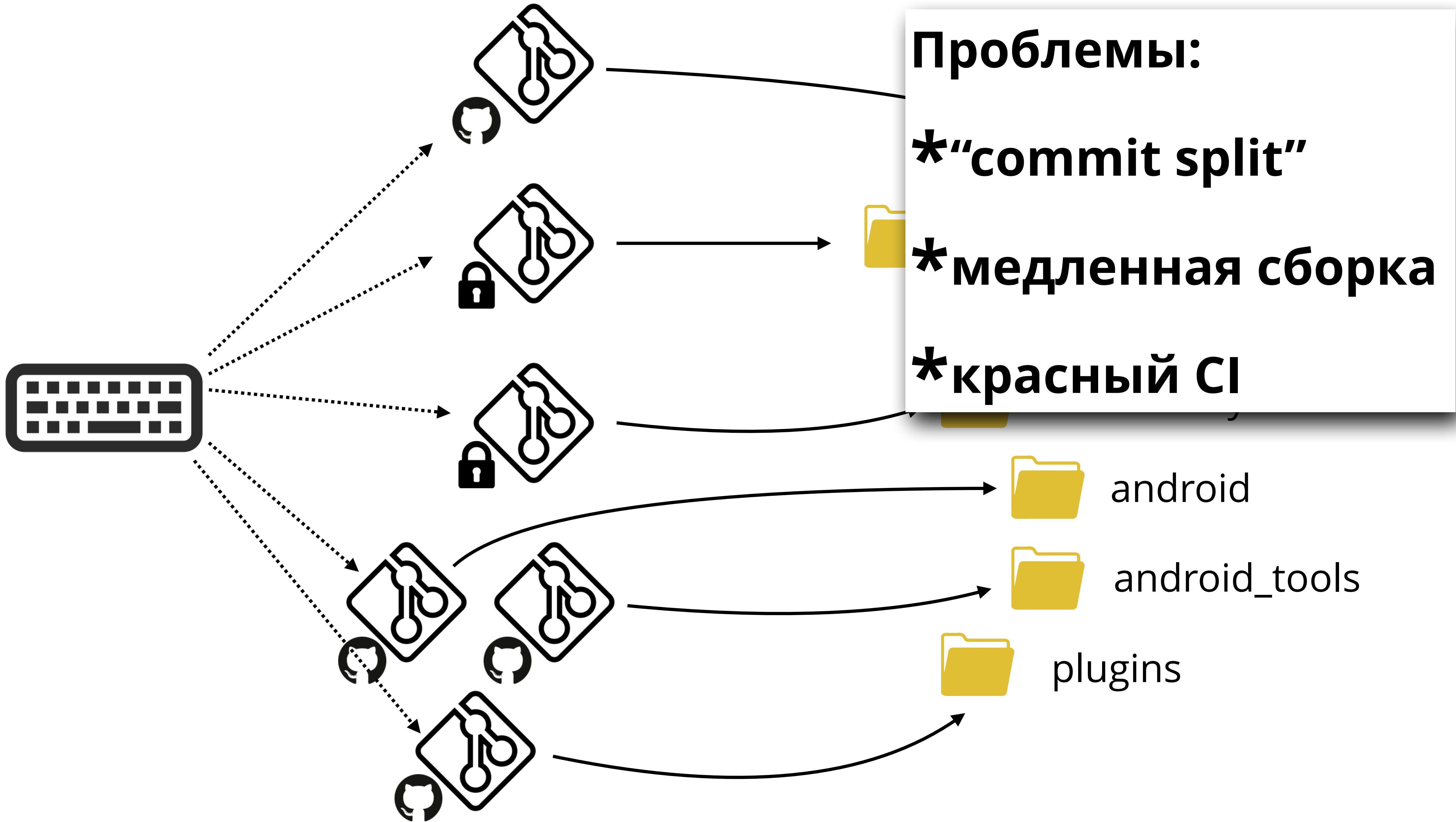
3.5G working tree

7.9G .git/objects









Проблемы:

***"commit split"**

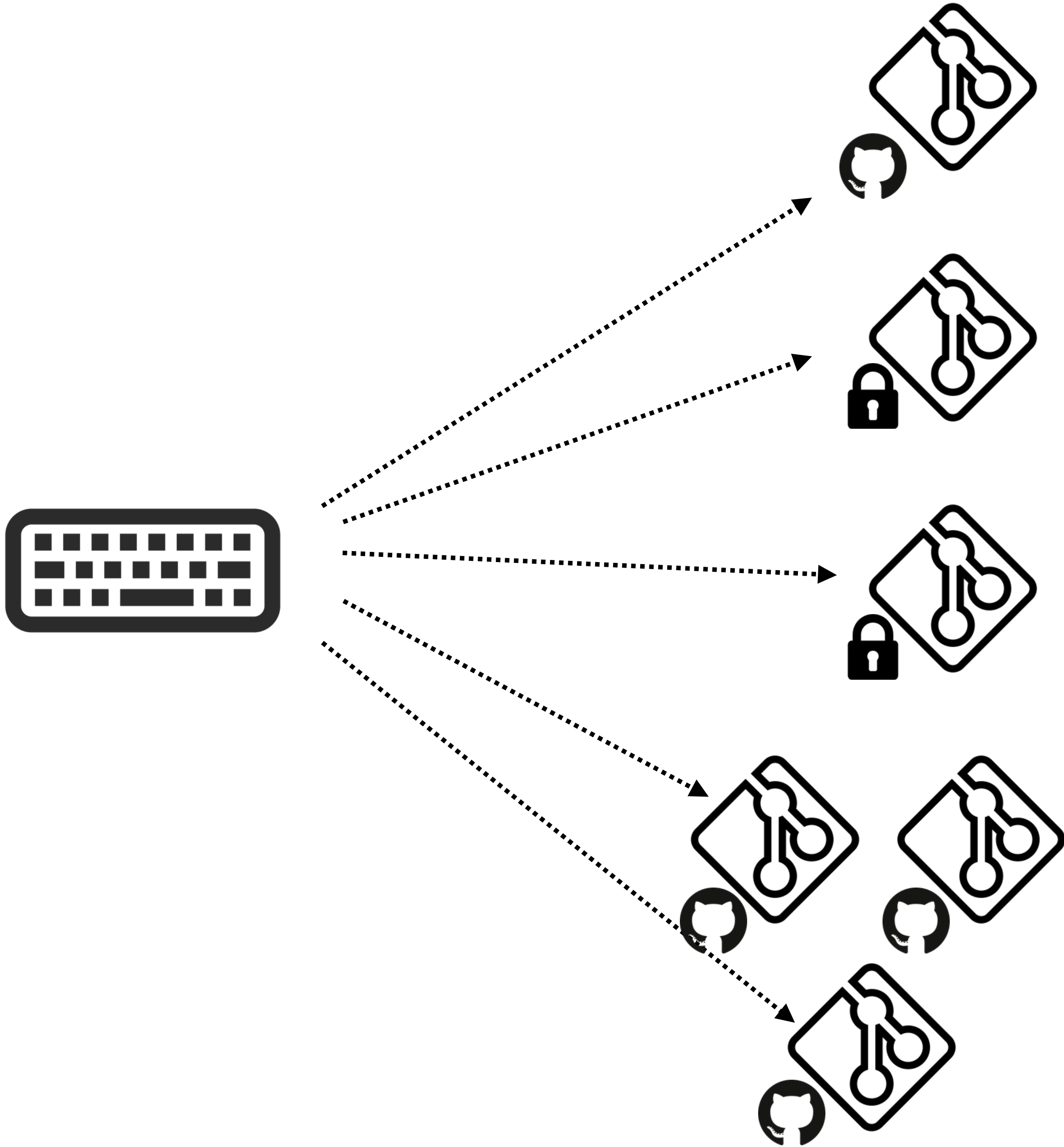
***медленная сборка**

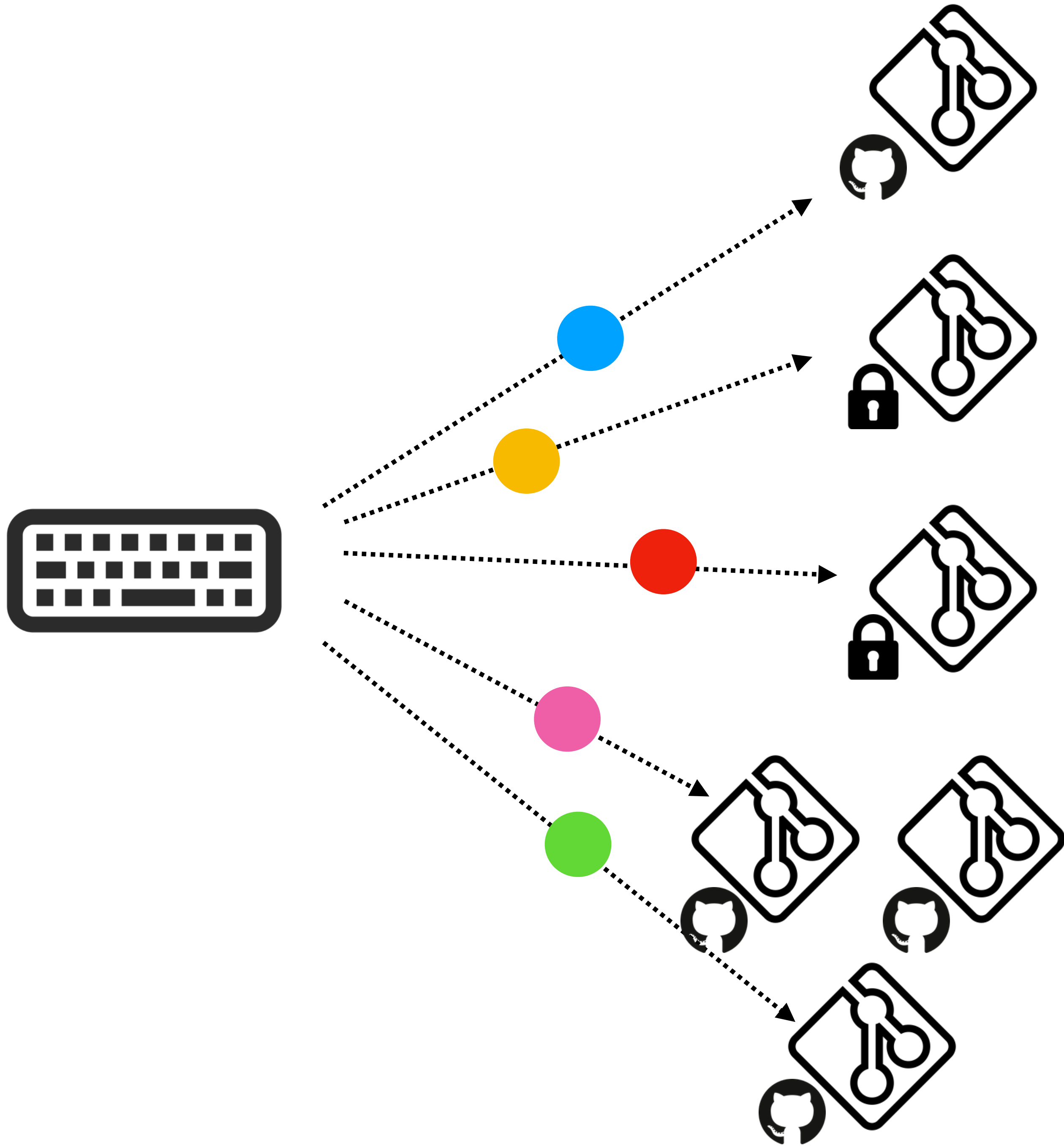
***красный CI**

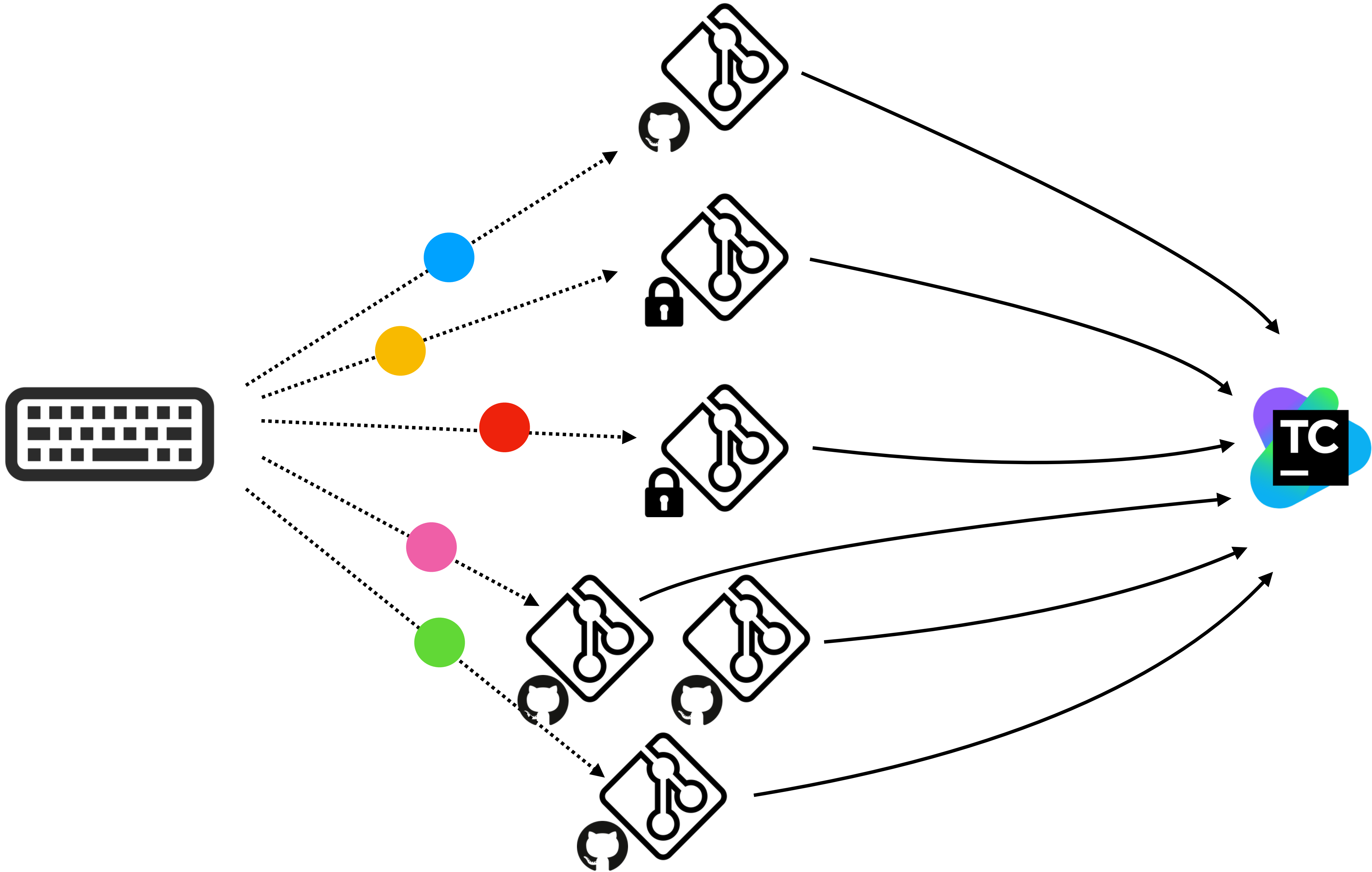
android

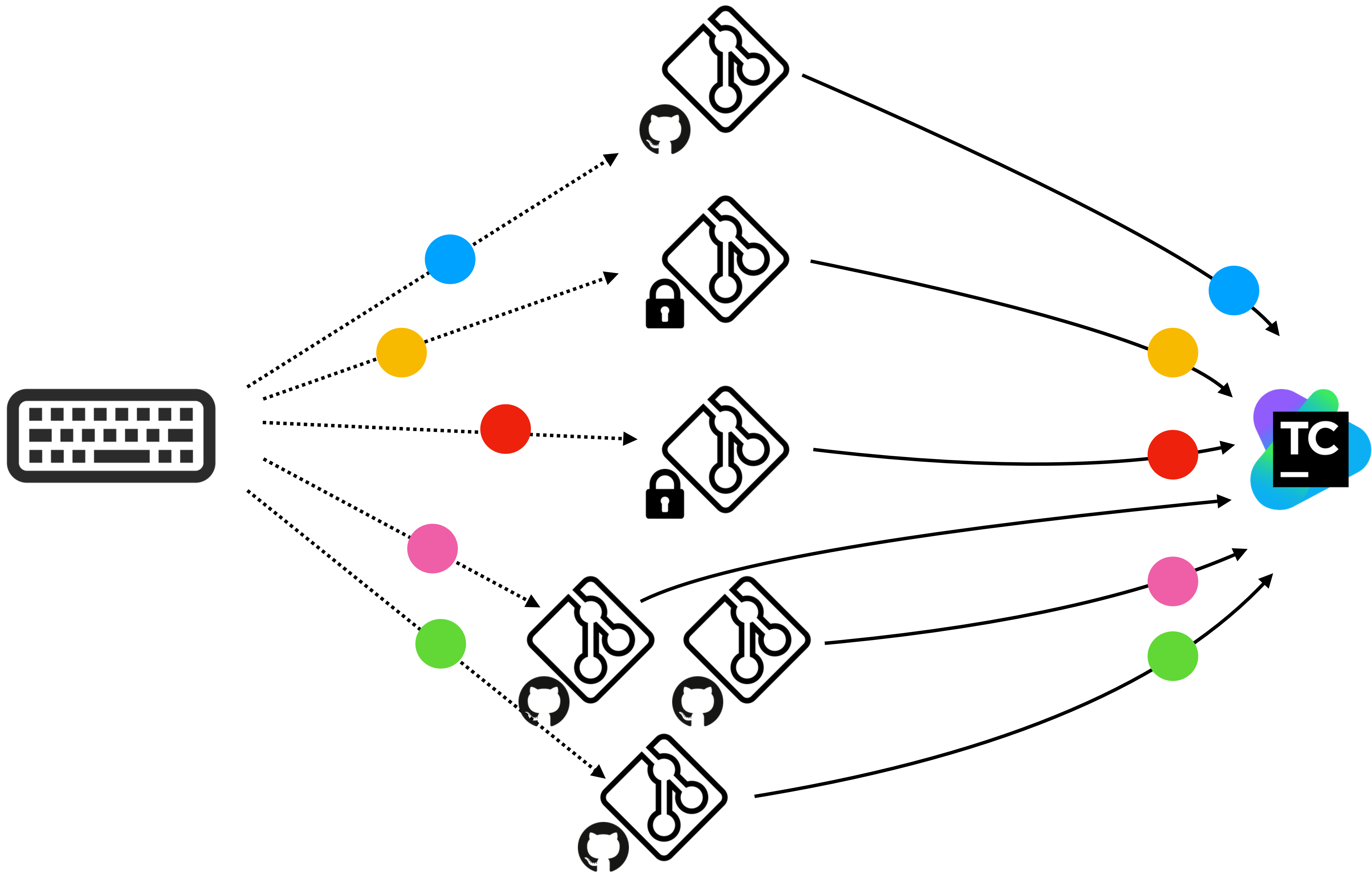
android_tools

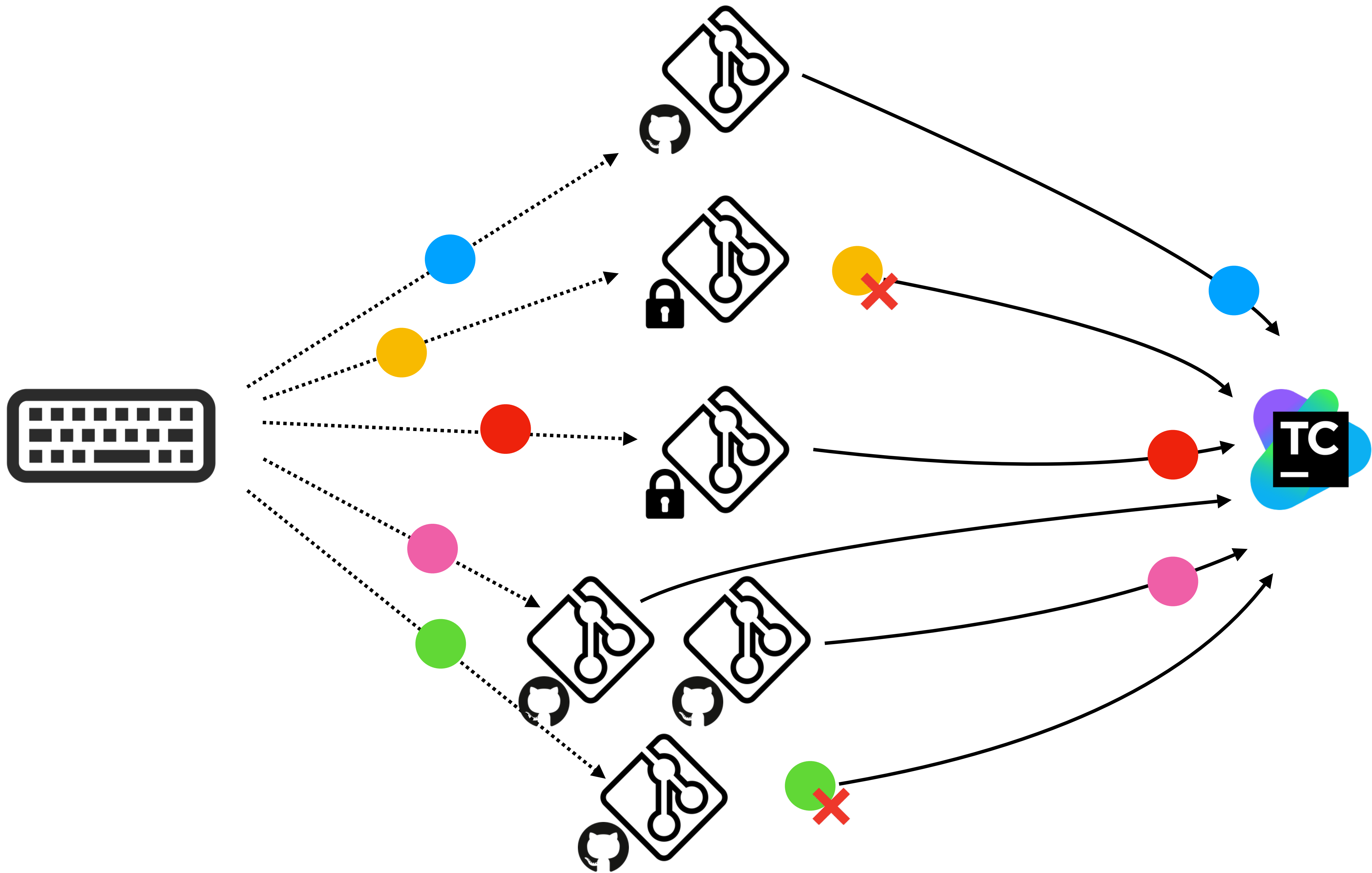
plugins

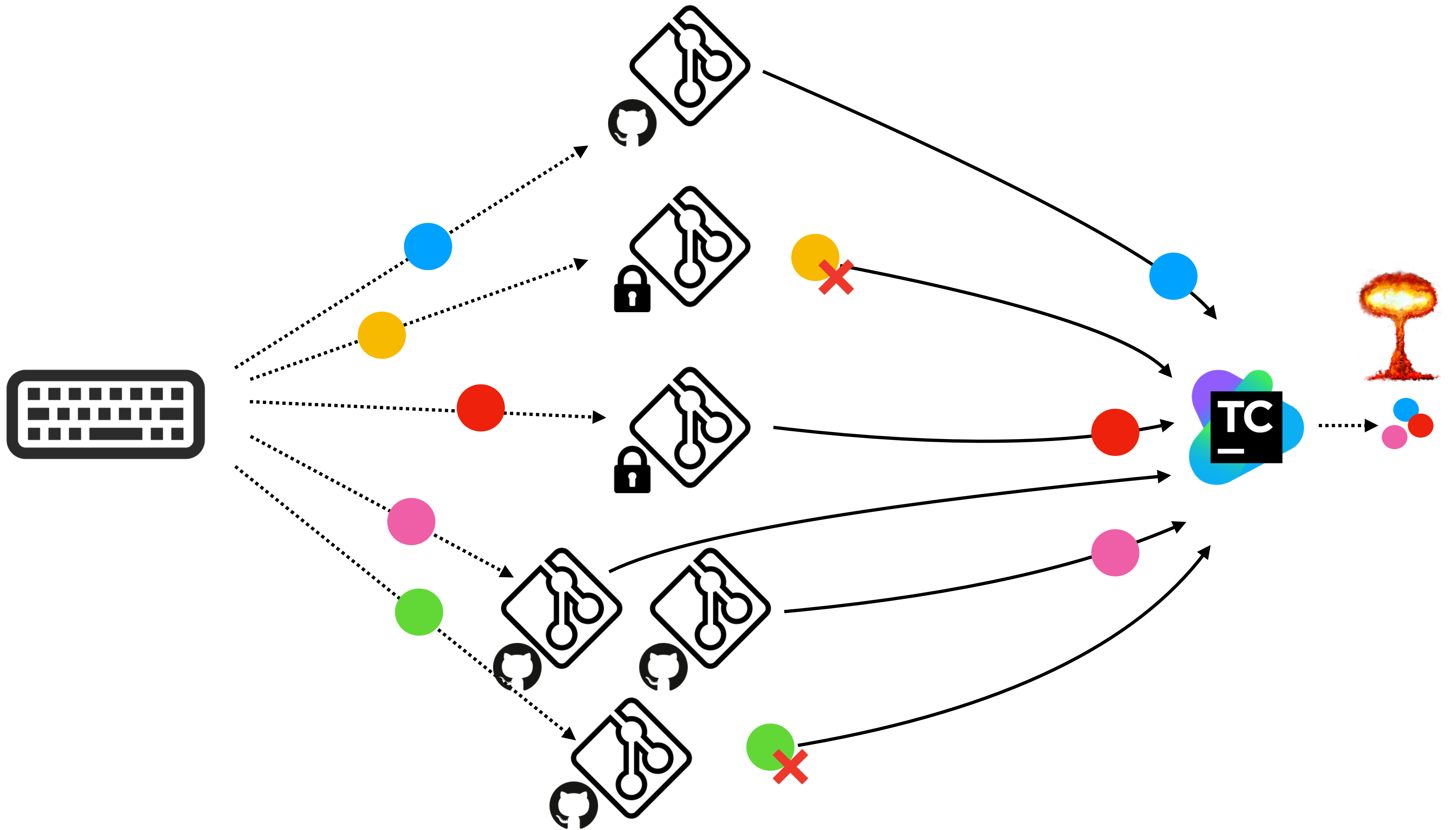


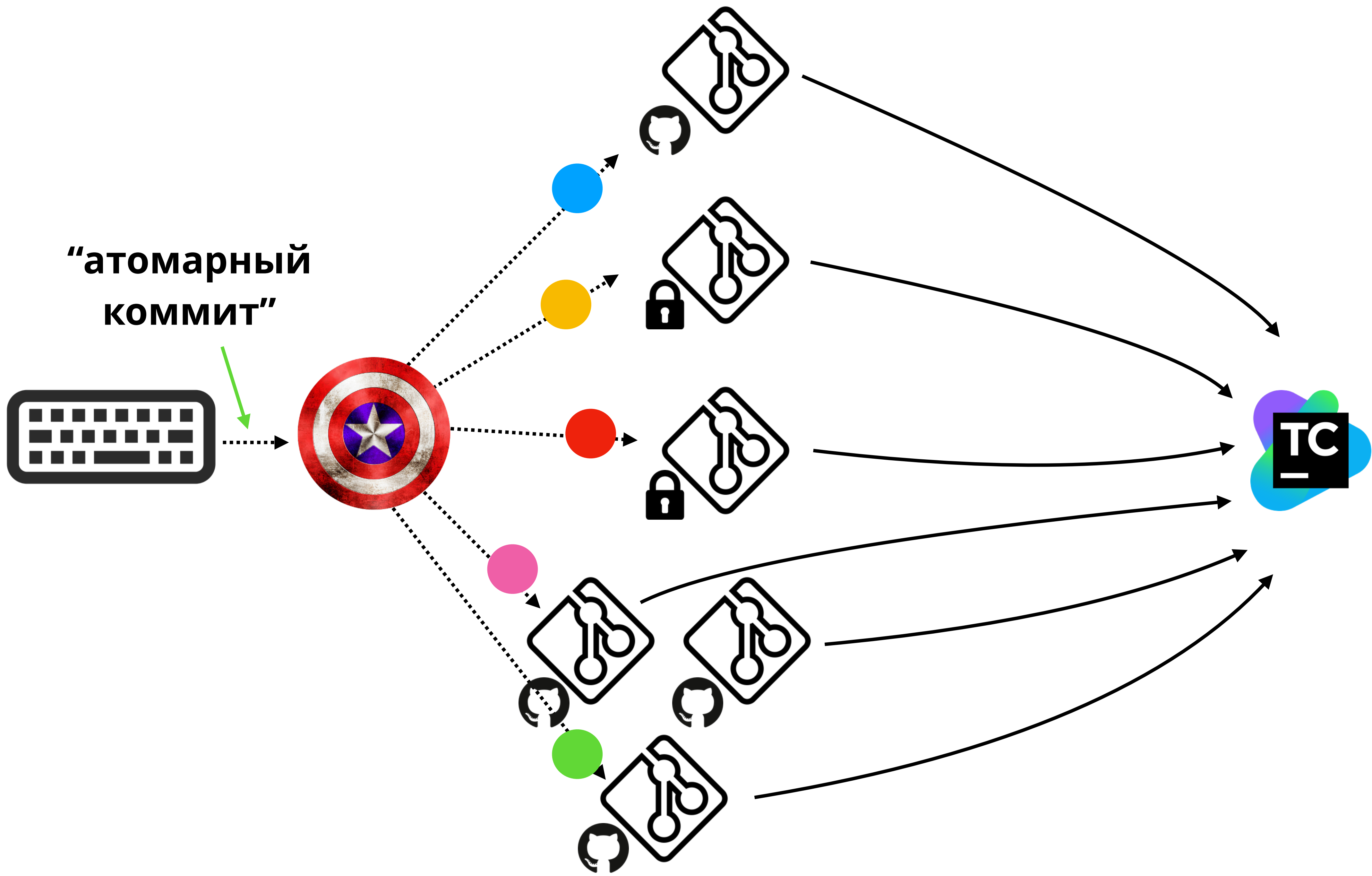


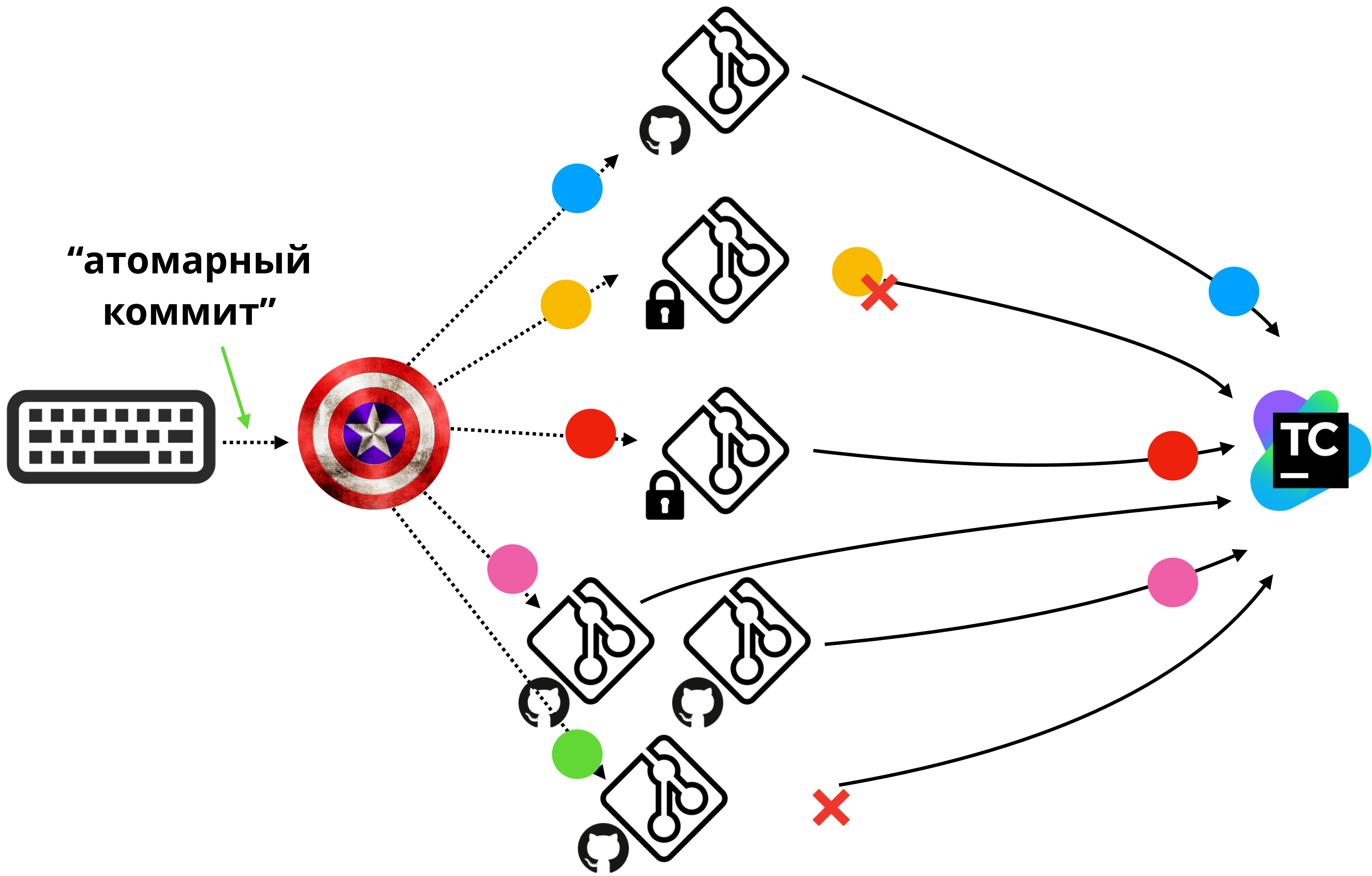


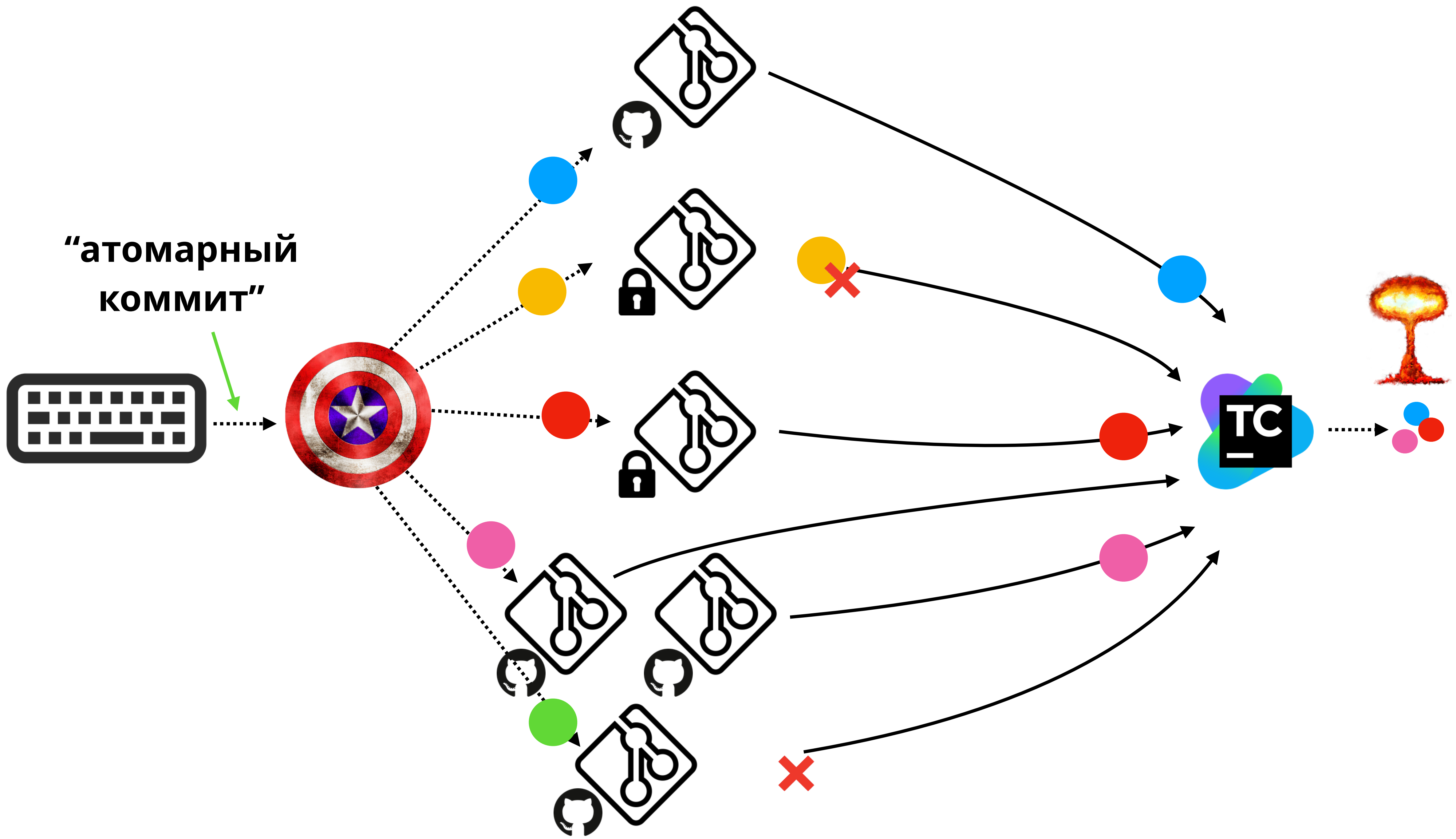




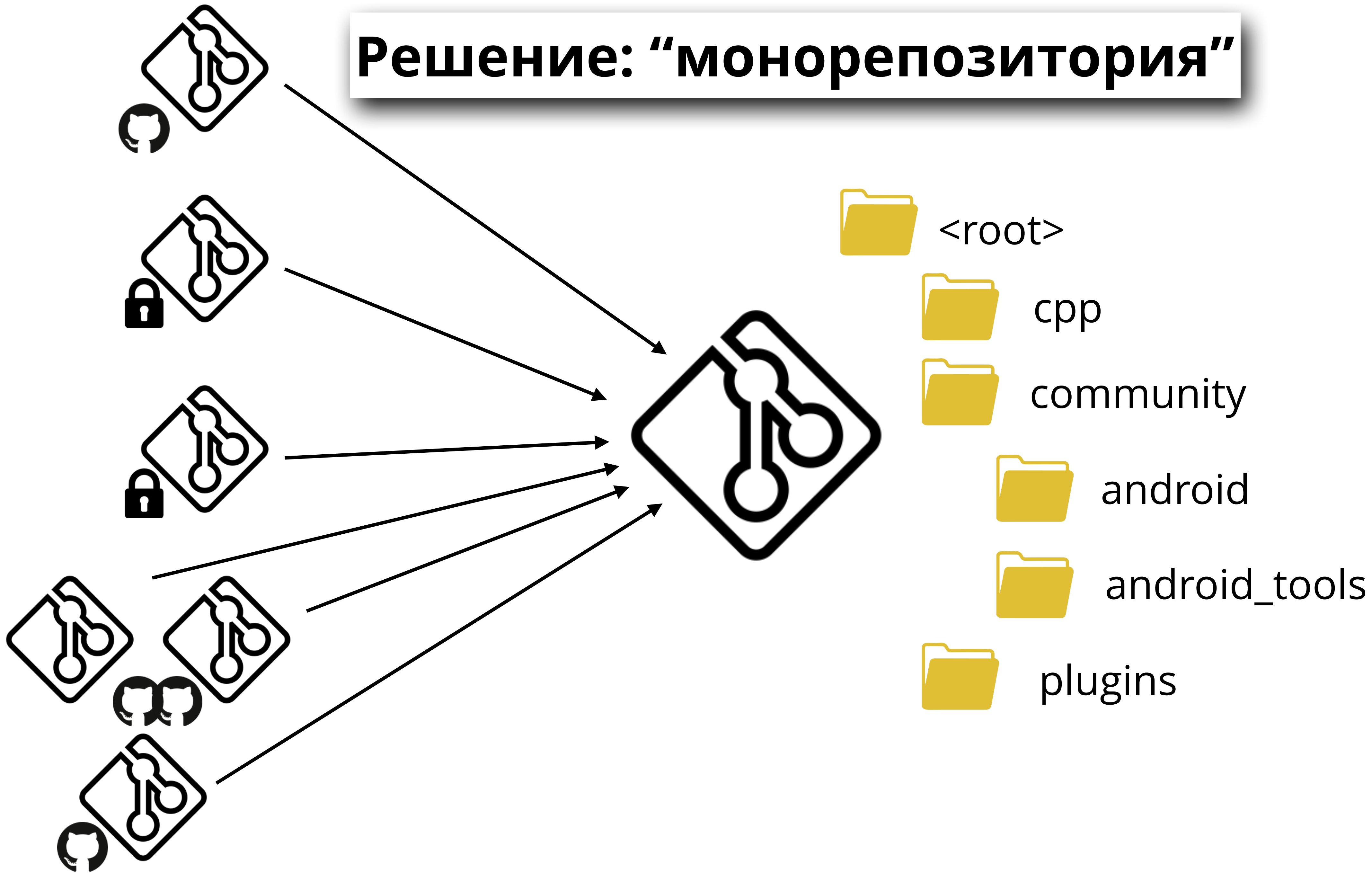


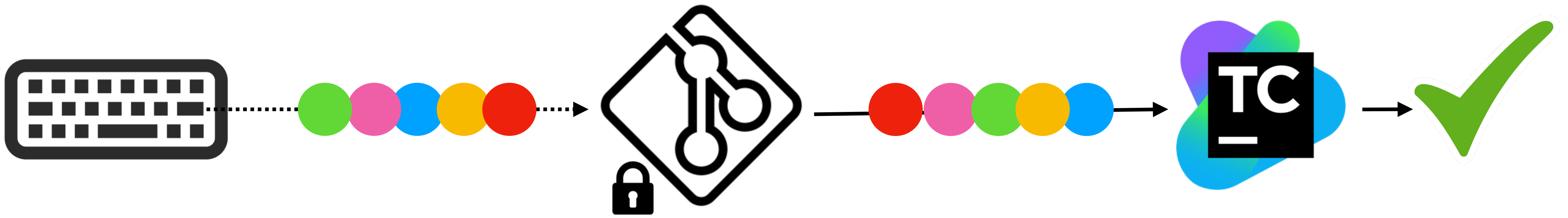


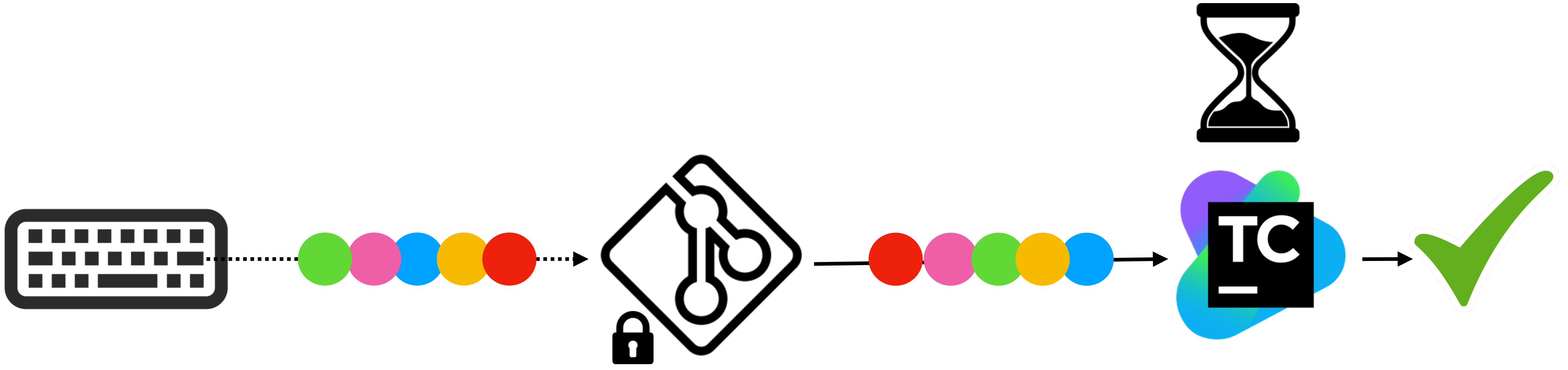




Решение: "монорепозитория"

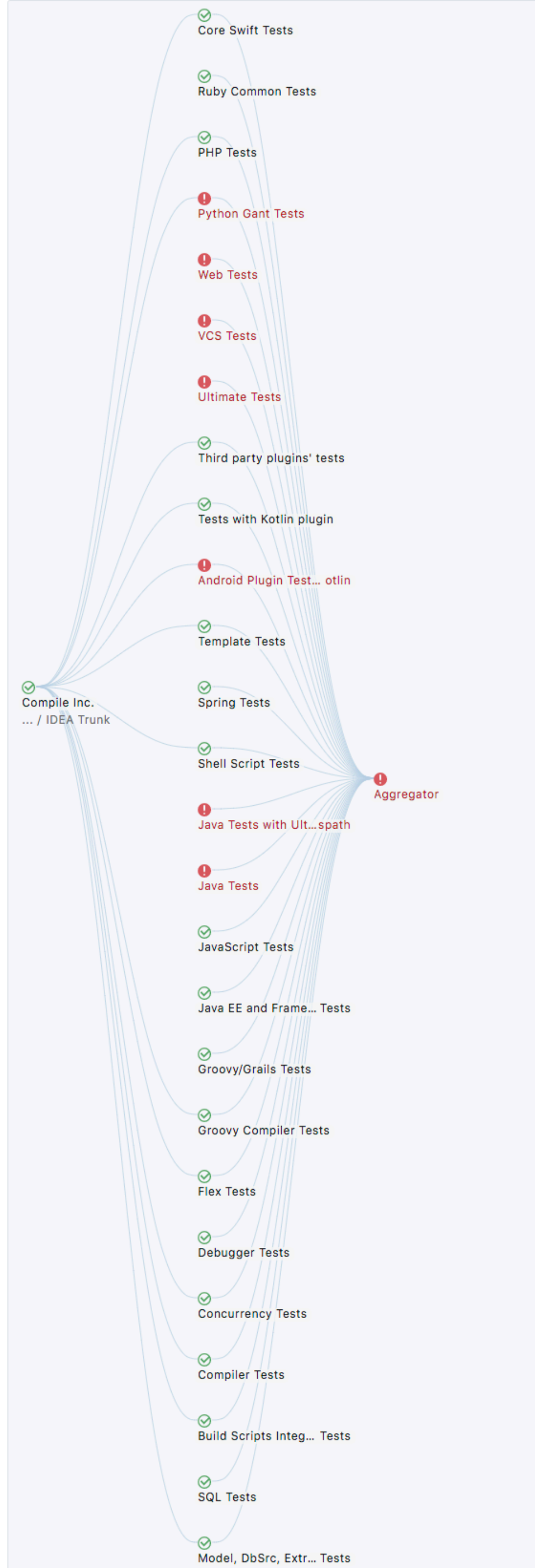








Сборка

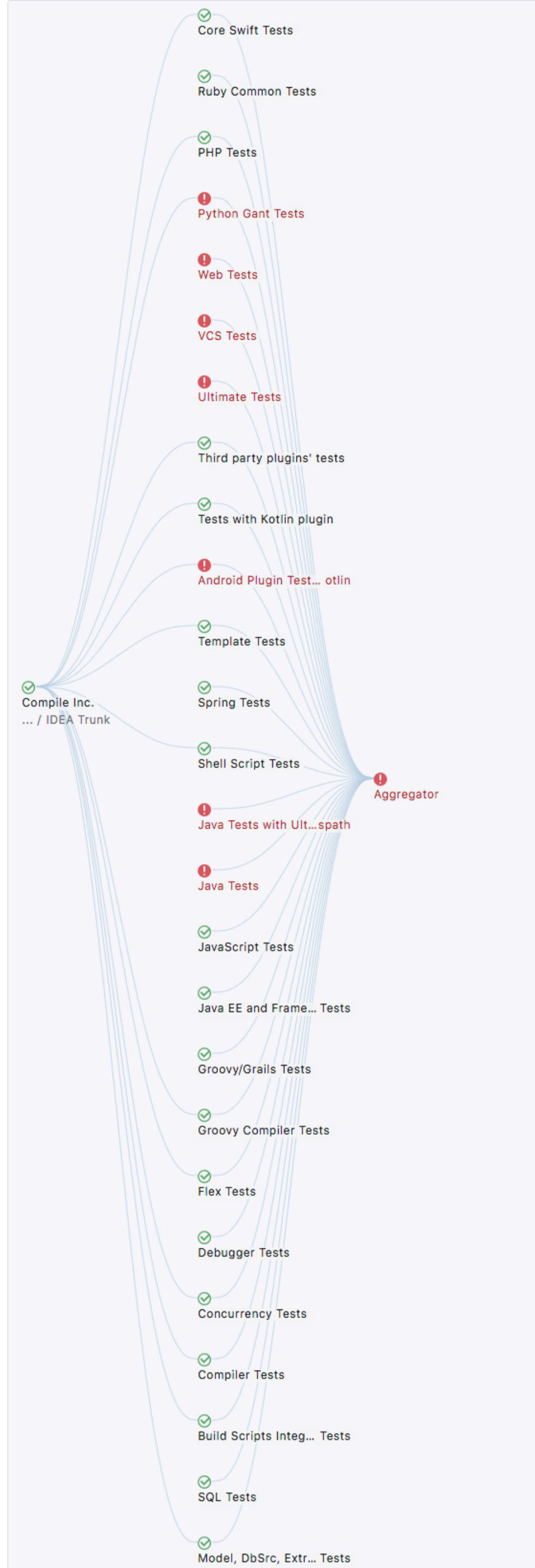


Полная сборка модулей: **~15 мин**

Вместе с тестами: **~30-60 мин**

Количество сборок в день: **~200**

Размер артефакта: **750Mb**



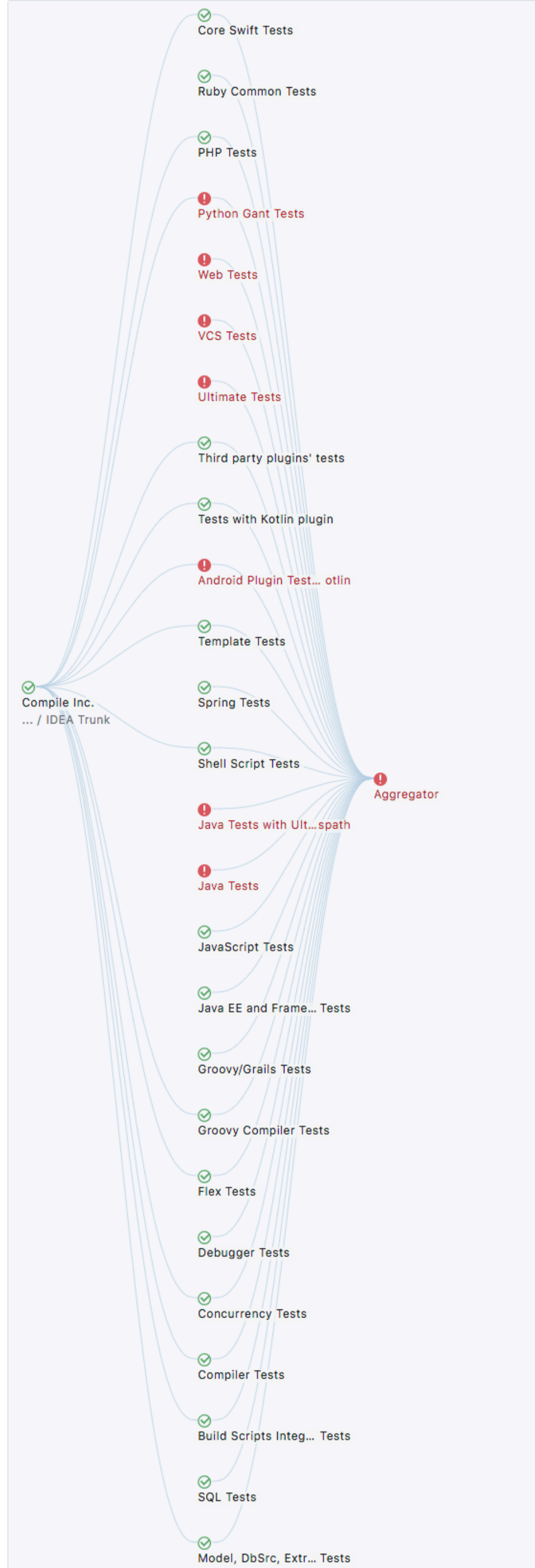
Полная сборка модулей: **~15 мин**

Вместе с тестами: **~30-60 мин**

Количество сборок в день: **~200**

Размер артефакта: **750Mb**

Задача: получить результат быстрее 15 минут



Полная сборка модулей: **~15 мин**

Вместе с тестами: **~30-60 мин**

Количество сборок в день: **~200**

Размер артефакта: **750Mb**

Задача: получить результат быстрее 15 минут

Решение: инкрементальная сборка
и параллелизация тестов

```
"branch": "master",
"prefix": "intellij-compile/v1/master",
"files": {
  "production/intellij.akka": "9167b0471ca230bc974c5a12b0e6409f1",
  "production/intellij.akka.tests": "d819e92ff68c9bf73da62481adb",
  "production/intellij.android.adb": "7b5a1609f8c3939cc3d27720b0",
  "production/intellij.android.adt.branding": "6cdf278f52eb0284f",
  "production/intellij.android.adt.ui": "d67996b6cf3cf6e540f15d6",
  "production/intellij.android.adt.ui.model": "381d03c29453b2e25",
  "production/intellij.android.androidLayoutInspector": "8f56c7ec",
  "production/intellij.android.apkanalyzer": "4a87fc4519fd3f17420ac9679b0095d7af3e94c6290e54dfb9297b42b6ee72b8",
  "production/intellij.android.artwork": "105571d9f948929af52d2b3969f18a97c1007cf6d07ea5a6a2c871b4e840dfc9",
  "production/intellij.android.buildCommon": "e3260e978c8f6c3a0921a0d75d258819edc86e94a2cc40a7d85521b6fc48b431",
  "production/intellij.android.common": "418f1db8a21c3f9092642319f5e1d72b45dfebc08b14f517472bbe19e3c03cba",
  "production/intellij.android.core": "c0efd31fae1bbb56bd1a100ef9fd9adc28c4374265ecb5db572609c158c6eba1",
  "production/intellij.android.databinding": "e3f8c1306e93c377ba12419ec5c0fb64e3d16340e00a7b01c629fd0e2675a6f",
  "production/intellij.android.debuggers": "eaa77707dec9bdca572033ce0332974510dcf651a24d05a4b86c08c46d651441",
  "production/intellij.android.deploy": "c4ed20b62477a85b7173e07909d672d3bf861f2bf564bbf4e63c74dfde751a4b",
  "production/intellij.android.designer": "d536b73f820473138a14f98126b0ac9947bbe710eb518014d245179b526d9ce3",
  "production/intellij.android.jps": "8cfb20300c05f57315a631400bfd7e2a59b1b1c05435a3149215a8fc1c747d35",
  "production/intellij.android.lang": "f10fc36ac04cd3e66a3f6c0caf4c27a7fbfd09ff2222d3f78b972200fbd80c1",
  "production/intellij.android.layoutInspector": "5009f0dd1bb419a3445808d589140059e153c4ca22e5a9b23cc48e9f5c1c2557",
  "production/intellij.android.layoutlibLoader": "488473493ad3d539247dca5b5bbf15d9974c72471e8c5c58c6f4f93169cdaaaa",
  "production/intellij.android.nativeSymbolizer": "d92412349608f77620cc16b86893647ffec136263948545c74104af1b086ba50",
  "production/intellij.android.naveditor": "74352cc54437ec9922090d363c00f8b0b9a766852abcd0a435155497f7119dce",
  "production/intellij.android.observable": "c18e908cdf115e111cd8805243a8f3c0482a73bea193d9bd9e5afb856fd9cd00",
  "production/intellij.android.observable.ui": "a2887b89bede1d4c49627c0d853209e1c70f5e1713941f27b069023cde0c8d9",
  "production/intellij.android.plugin": "73d92178464c27bb58d643c46bcfab6553f0ea907d775c37d208e296f94437e9",
  "production/intellij.android.profilers": "9639586e17aa23bf266e7621a740a60945cd5ede47f1731315b4bd7dd2d17fe3",
  "production/intellij.android.profilers.atrace": "ad36be09165cbdf5c172b6a2f0aa496d7a5e56a806c9091cb21f76203c8d0428",
  "production/intellij.android.profilers.ui": "474cc5c131e2c40cf61b0ad97b74e5235cf9c385c37bf6c6ed112779ede8f450",
  "production/intellij.android.profilersAndroid": "ac2cab943e5547566525d69f252b3b07033c9ab095afb11d3bd31eb37372a42b",
  "production/intellij.android.projectSystem": "0634b2ceaf3c94ed5ee0449706bbcac17211b16aea4b741ce696dde424ab017b",
  "production/intellij.android.projectSystem.gradle": "7cbfde8aa7437f66d238a6938032453b11bc39b80af24e4356ed22c446314c2d",
  "production/intellij.android.propertyEditor": "6b3c606e022f91cd5ad30702d3b3414e1f80032f033df930d6746ef7ffcb8cfc",
  "production/intellij.android.resourcesAar": "d72253e9b8baf8c5a4ac68febd4f9149a9fa22beaca96434efed93a568811d96",
  "production/intellij.android.rt": "be8c403306c68ec23d4978d5895f283197e7f1989801d04009e0d213653abdee",
  "production/intellij.android.sdkUpdates": "5ff3f1180f622016f14ac0bdaa69bc3ec01b1cea02df220c470a9b8ee572ab8e",
  "production/intellij.android.smali": "5a3570312725d67e5937c8b653c7a97676bcf2cbd2825b0d9087b692f24a0f6e",
```

1. Собираем все модули отдельно

2. Отправляем в webdav-хранилище только те, которые изменились

1250 модулей

```
"branch": "master",
"prefix": "intellij-compile/v1/master",
"files": {
  "production/intellij.akka": "9167b0471ca230bc974c5a12",
  "production/intellij.akka.tests": "d819e92ff68c9bf73d",
  "production/intellij.android.adb": "7b5a1609f8c3939cc",
  "production/intellij.android.adt.branding": "6cdf278f",
  "production/intellij.android.adt.ui": "d67996b6cf3cf6",
  "production/intellij.android.adt.ui.model": "381d03c2",
  "production/intellij.android.androidLayoutInspector": "4a87fc4519fd3f17420ac9679b0095d7af3e94c6290e54dfb9297b42b6ee72b8",
  "production/intellij.android.apkanalyzer": "105571d9f948929af52d2b3969f18a97c1007cf6d07ea5a6a2c871b4e840dfc9",
  "production/intellij.android.artwork": "e3260e978c8f6c3a0921a0d75d258819edc86e94a2cc40a7d85521b6fc48b431",
  "production/intellij.android.buildCommon": "418f1db8a21c3f9092642319f5e1d72b45dfebc08b14f517472bbe19e3c03cba",
  "production/intellij.android.common": "c0efd31fae1bbb56bd1a100ef9fd9adc28c4374265ecb5db572609c158c6eba1",
  "production/intellij.android.core": "e3f8c1306e93c377ba12419ec5c0fb64e3d16340e00a7b01c629fd0e2675a6f",
  "production/intellij.android.databinding": "eaa77707dec9bdca572033ce0332974510dcf651a24d05a4b86c08c46d651441",
  "production/intellij.android.debuggers": "c4ed20b62477a85b7173e07909d672d3bf861f2bf564bbf4e63c74dfde751a4b",
  "production/intellij.android.deploy": "d536b73f820473138a14f98126b0ac9947bbe710eb518014d245179b526d9ce3",
  "production/intellij.android.designer": "8cfb20300c05f57315a631400bfd7e2a59b1b1c05435a3149215a8fc1c747d35",
  "production/intellij.android.jsp": "f10fc36ac04cd3e66a3f6c0caf4c27a7fbfd09ff2222d3f78b972200fbd80c1",
  "production/intellij.android.lang": "5009f0dd1bb419a3445808d589140059e153c4ca22e5a9b23cc48e9f5c1c2557",
  "production/intellij.android.layoutInspector": "488473493ad3d539247dca5b5bbf15d9974c72471e8c5c58c6f4f93169cdaaaa",
  "production/intellij.android.layoutlibLoader": "d92412349608f77620cc16b86893647ffec136263948545c74104af1b086ba50",
  "production/intellij.android.nativeSymbolizer": "74352cc54437ec9922090d363c00f8b0b9a766852abcd0a435155497f7119dce",
  "production/intellij.android.naveditor": "c18e908cdf115e111cd8805243a8f3c0482a73bea193d9bd9e5afb856fd9cd00",
  "production/intellij.android.observable": "a2887b89bede1d4c49627c0d853209e1c70f5e1713941f27b069023cde0c8d9",
  "production/intellij.android.observable.ui": "73d92178464c27bb58d643c46bcfab6553f0ea907d775c37d208e296f94437e9",
  "production/intellij.android.plugin": "9639586e17aa23bf266e7621a740a60945cd5ede47f1731315b4bd7dd2d17fe3",
  "production/intellij.android.profilers": "ad36be09165cbdf5c172b6a2f0aa496d7a5e56a806c9091cb21f76203c8d0428",
  "production/intellij.android.profilers.atrace": "474cc5c131e2c40cf61b0ad97b74e5235cf9c385c37bf6c6ed112779ede8f450",
  "production/intellij.android.profilers.ui": "ac2cab943e5547566525d69f252b3b07033c9ab095afb11d3bd31eb37372a42b",
  "production/intellij.android.profilersAndroid": "0634b2ceaf3c94ed5ee0449706bbcac17211b16aea4b741ce696dde424ab017b",
  "production/intellij.android.projectSystem": "7cbfde8aa7437f66d238a6938032453b11bc39b80af24e4356ed22c446314c2d",
  "production/intellij.android.projectSystem.gradle": "6b3c606e022f91cd5ad30702d3b3414e1f80032f033df930d6746ef7ffcb8cfc",
  "production/intellij.android.propertyEditor": "d72253e9b8baf8c5a4ac68febd4f9149a9fa22beaca96434efed93a568811d96",
  "production/intellij.android.resourcesAar": "be8c403306c68ec23d4978d5895f283197e7f1989801d04009e0d213653abdee",
  "production/intellij.android.rt": "5ff3f1180f622016f14ac0bdaa69bc3ec01b1cea02df220c470a9b8ee572ab8e",
  "production/intellij.android.sdkUpdates": "5a3570312725d67e5937c8b653c7a97676bcf2cbd2825b0d9087b692f24a0f6e",
  "production/intellij.android.smali": "5a3570312725d67e5937c8b653c7a97676bcf2cbd2825b0d9087b692f24a0f6e",
```

Результат:

Общий размер пересылаемых артефактов: 25 Мб

Время сборки: ~1 минута (на более слабой машине)

#4669 ☆

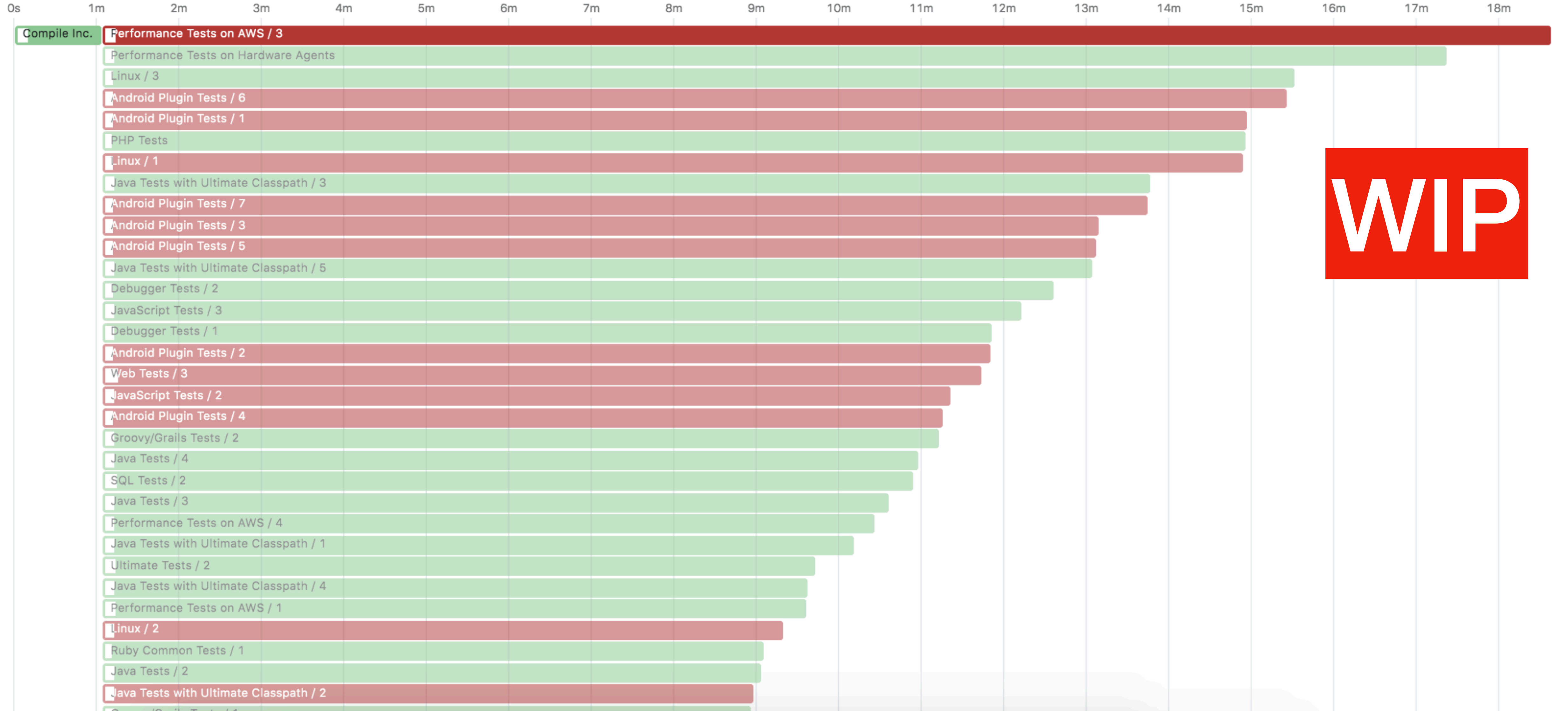
Tests failed: 97 (2 new), passed: 132428, ignored: 1354, muted: 87; Build chain finished (success: 54, failed: 27)



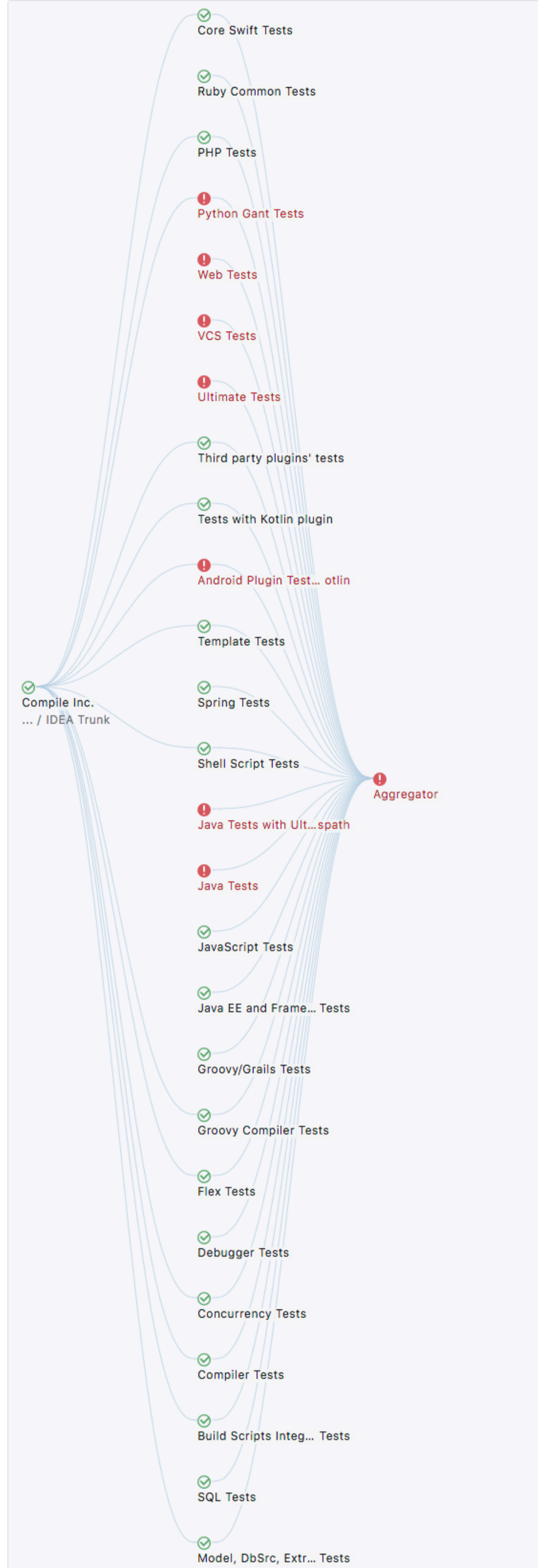
master Investigation was assigned to Alexander Zolotov Actions

Overview Tests Changes Log Dependencies More

List Timeline



WIP



Как распараллелить тесты?



ВЫВОДЫ

**Можно адаптировать
процесс под инструмент**

**Можно адаптировать
инструмент под процесс**

**“Город никогда не будет
построен”**



Вопросы?

