Intro to iOS Unit Testing

Or how to pour code into your code so you can code your way out of incorrect code

by Fernando Olivares

Agenda

1-hour session

- Author Introduction
- Also why we're here: What is Unit Testing?
- Live coding
- Conclusion & Where to go from here
- Q&A

Author Introduction

Fernando

- ~10 years of experience
- Worked at small startups (1SecondEveryday) to publicly traded companies (J2 Global Inc.)
- Instructor at Big Nerd Ranch Big Nerd Ranch, Bloc bloc.io, Λ Lambda School
- Won a few awards: The Storyteller Within (Apple), ERA Accelerator Top 10 (ERA NY)
- Product and Project experience
- iOS-only, bought the first iPhone without knowing if he'd be able to use it
- @fromJrToSr

What is Unit Testing?

In theory they're amazing

- A unit test is a function that tests other functions.
- Automated tests that validate the logic and/or performance of one or more functions.
- Allow you to continuously test a wide range of scenarios.
- Protect against regressions and reduce the need for manual testing.
- Opens up the possibility of Test Driven Development

What is Unit Testing?

In practice they are constrained by reality

- Code that tests other code.
- Provide incredible value when they catch bugs.
- You can spend an afternoon creating unit tests for a feature and they will catch *any* changes to the tested function.
- It is very satisfying to know that part of your software is covered from a lot of scenarios you can think of.
- They are very expensive and are usually either the last part of a feature being built or are ignored altogether.

What is Unit Testing?

Assertions

How does XCTest work?

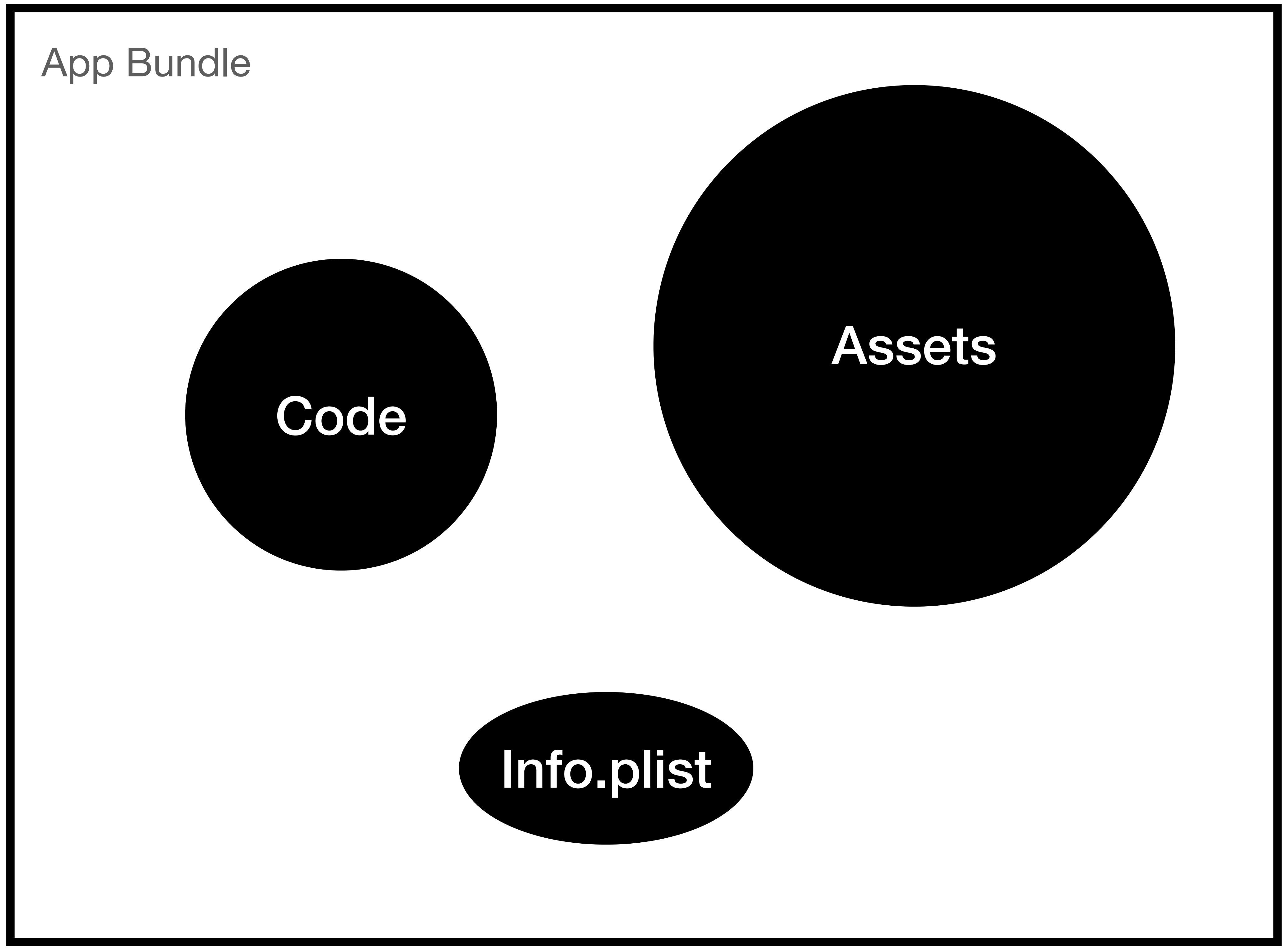
- all functions prefixed with `test` will be run as unit tests
- any function that does not finish execution is considered failed
    - the purpose is to validate each "logical" step **using XCTAssert**
- any function that finishes execution is considered successful

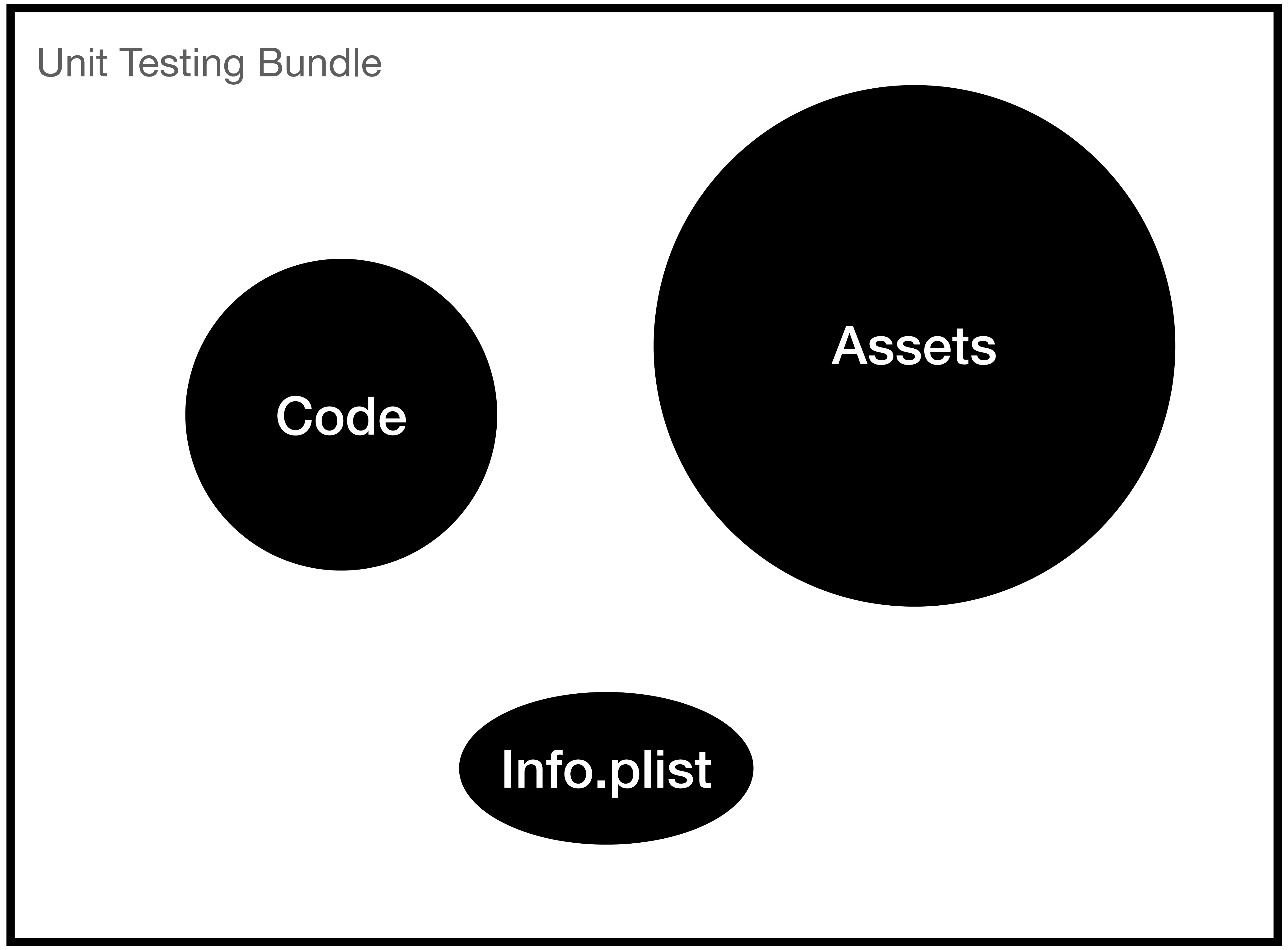
Live Demo

Add our unit test bundle

What is unit testing?

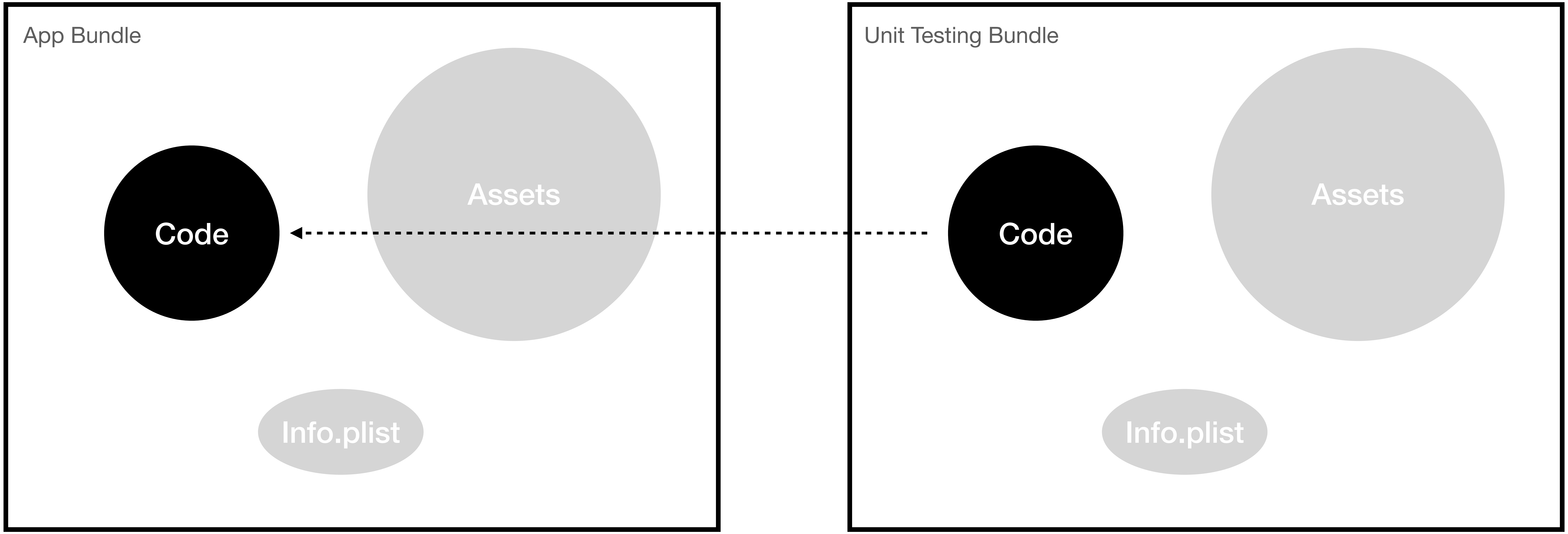
It's in a separate bundle





What is unit testing?

It's in a separate bundle



What is Unit Testing?

It's in a separate bundle

Access Modifiers

- open - all bundles can access and subclass
- public - all bundles can access
- **internal - single-bundle access**
- fileprivate - single-file access
- private - single class/struct access

What is Unit Testing?

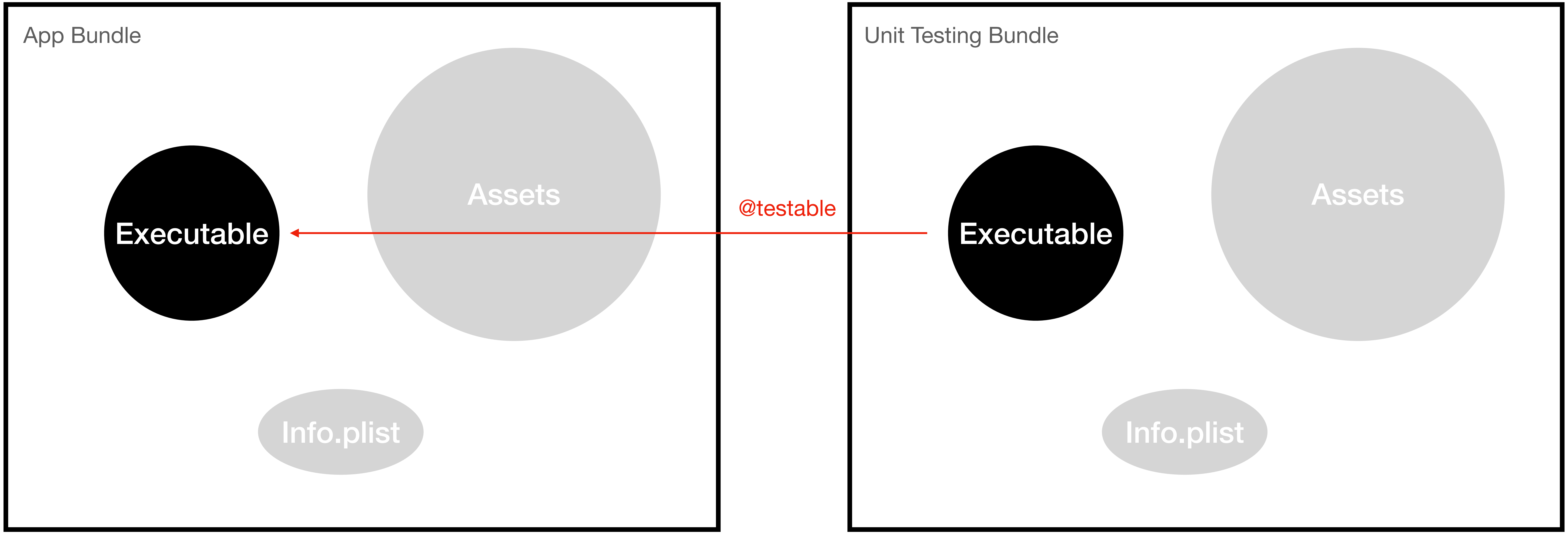
It's in a separate bundle

Access Modifiers with @testable

- **open - all bundles can access and subclass**
- **public - all bundles can access**
- ~~internal - single-bundle access~~
- fileprivate - single-file access
- private - single class/struct access

What is unit testing?

It's in a separate bundle



Live Demo

Test `fetchPlanets`

What is Unit Testing?

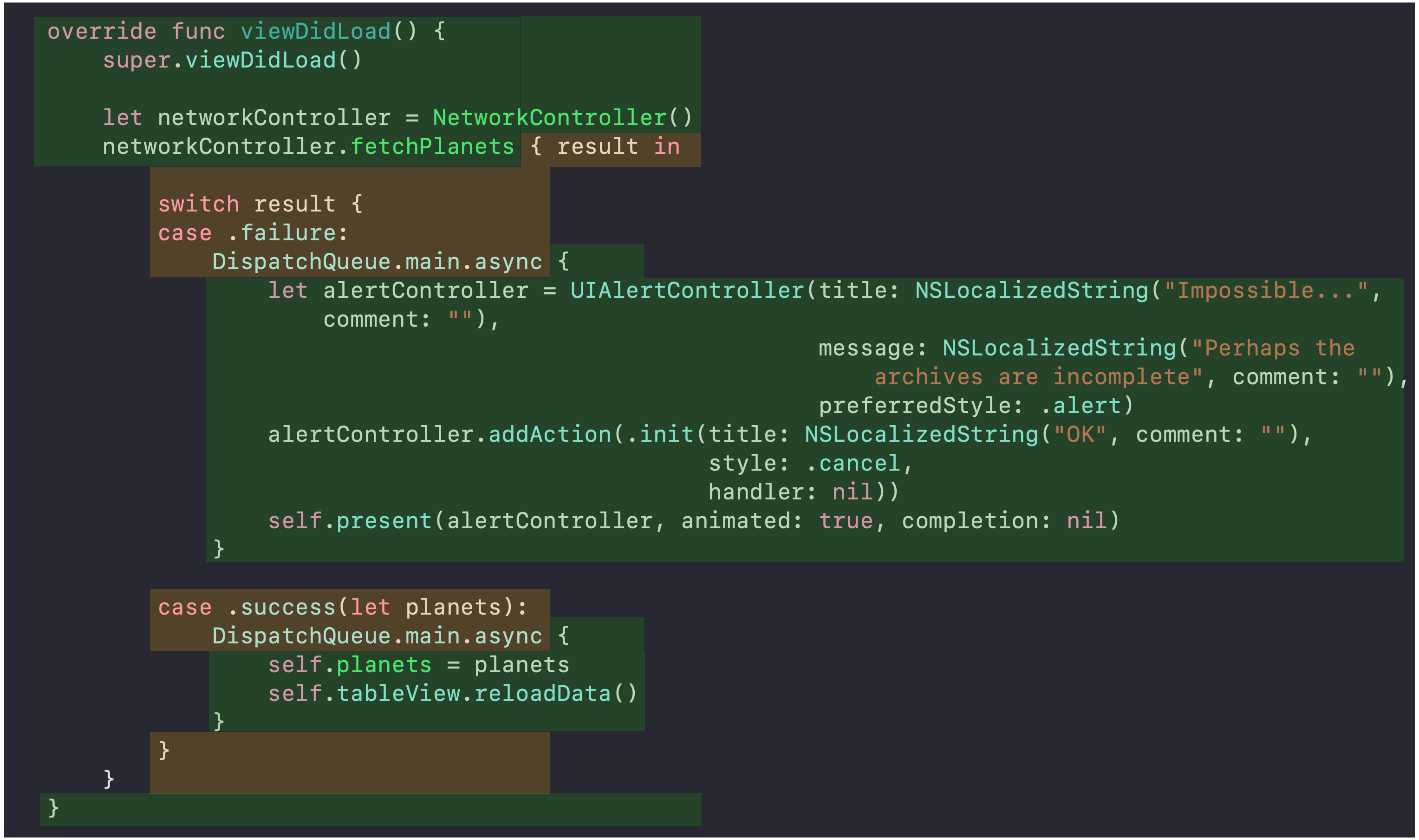
Synchronicity

How does XCTest work?

- all functions prefixed with `test` will be run as unit tests
- any function that does not finish execution is considered failed
  - the purpose is to validate each "logical" step using XCTAssert
- **any function that finishes execution is considered successful**

What is Unit Testing?

Synchronicity



```
override func viewDidLoad() {
    super.viewDidLoad()

    let networkController = NetworkController()
    networkController.fetchPlanets { result in

        switch result {
        case .failure:
            DispatchQueue.main.async {
                let alertController = UIAlertController(title: NSLocalizedString("Impossible...",
                    comment: ""),
                                                        message: NSLocalizedString("Perhaps the
                                                            archives are incomplete", comment: ""),
                                                        preferredStyle: .alert)
                alertController.addAction(.init(title: NSLocalizedString("OK", comment: ""),
                                                style: .cancel,
                                                handler: nil))
                self.present(alertController, animated: true, completion: nil)
            }

        case .success(let planets):
            DispatchQueue.main.async {
                self.planets = planets
                self.tableView.reloadData()
            }
        }
    }
}
```

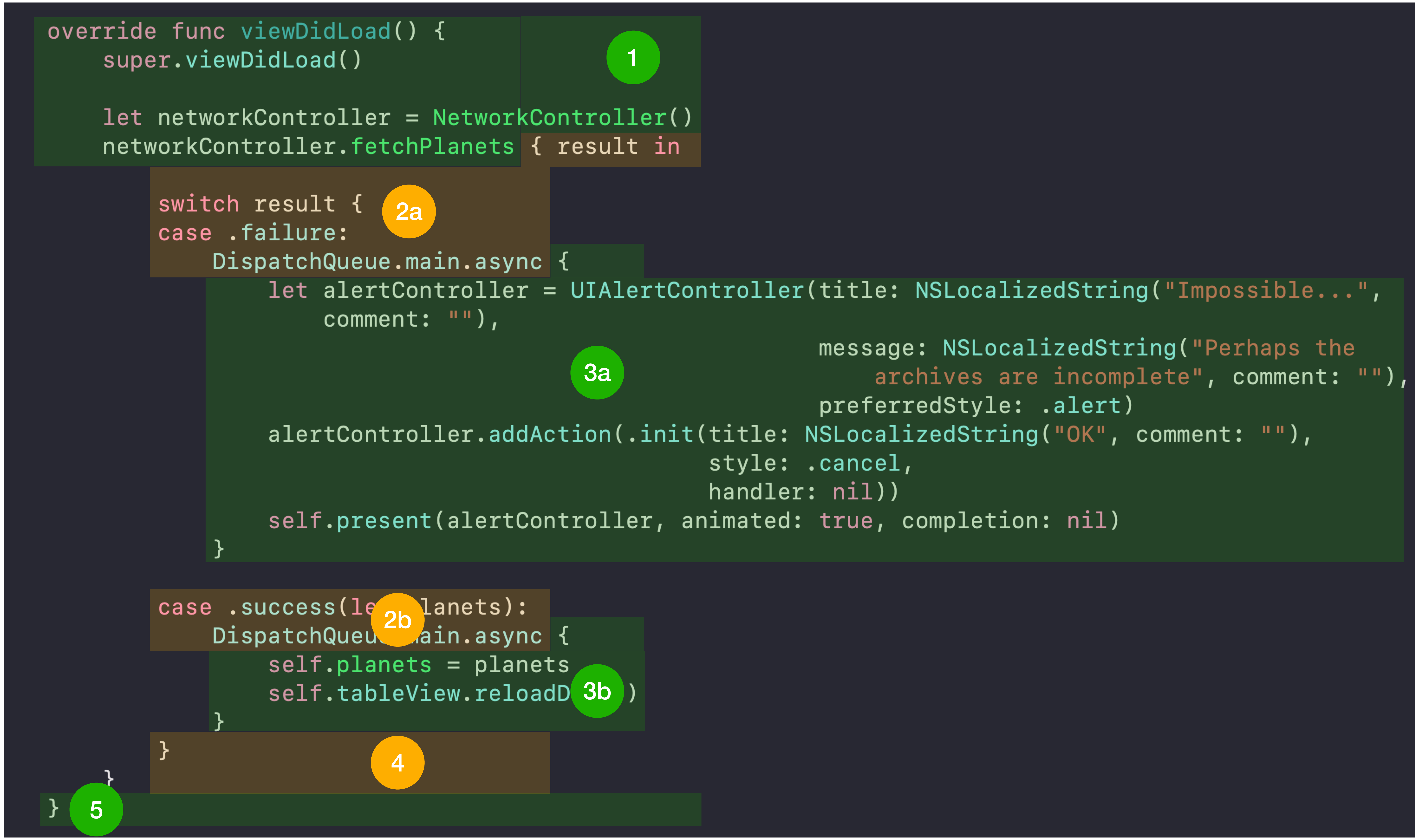
🟩 Main Queue

🟨 Not Main Queue

Time ↓

What is Unit Testing?

Synchronicity

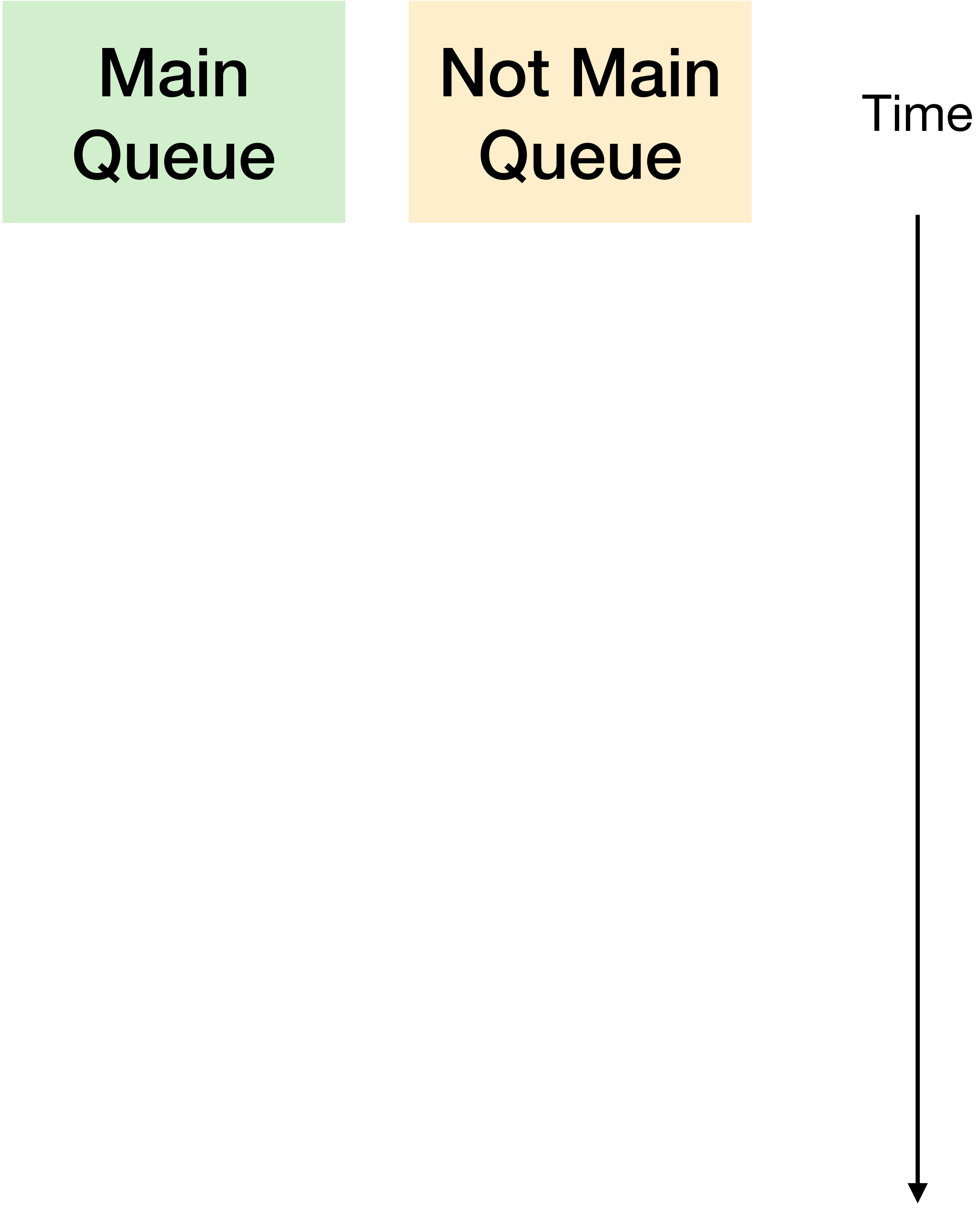


```
override func viewDidLoad() {
    super.viewDidLoad()

    let networkController = NetworkController()
    networkController.fetchPlanets { result in

        switch result {
        case .failure:
            DispatchQueue.main.async {
                let alertController = UIAlertController(title: NSLocalizedString("Impossible...",
                    comment: ""),
                                                        message: NSLocalizedString("Perhaps the
                                                            archives are incomplete", comment: ""),
                                                        preferredStyle: .alert)
                alertController.addAction(.init(title: NSLocalizedString("OK", comment: ""),
                                                style: .cancel,
                                                handler: nil))
                self.present(alertController, animated: true, completion: nil)
            }

        case .success(let planets):
            DispatchQueue.main.async {
                self.planets = planets
                self.tableView.reloadData()
            }
        }
    }
}
```



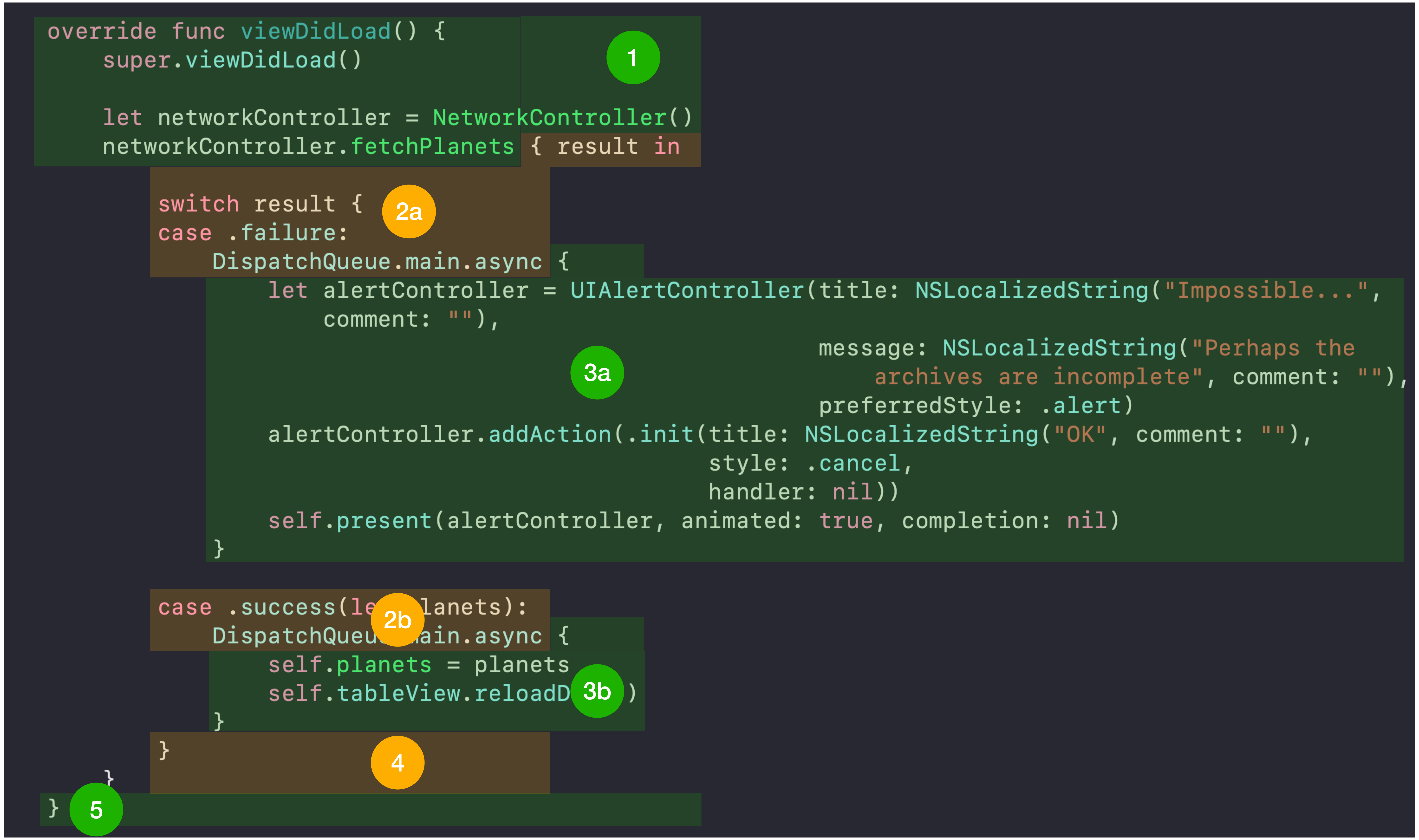
Main Queue

Not Main Queue

Time

What is Unit Testing?

Synchronicity

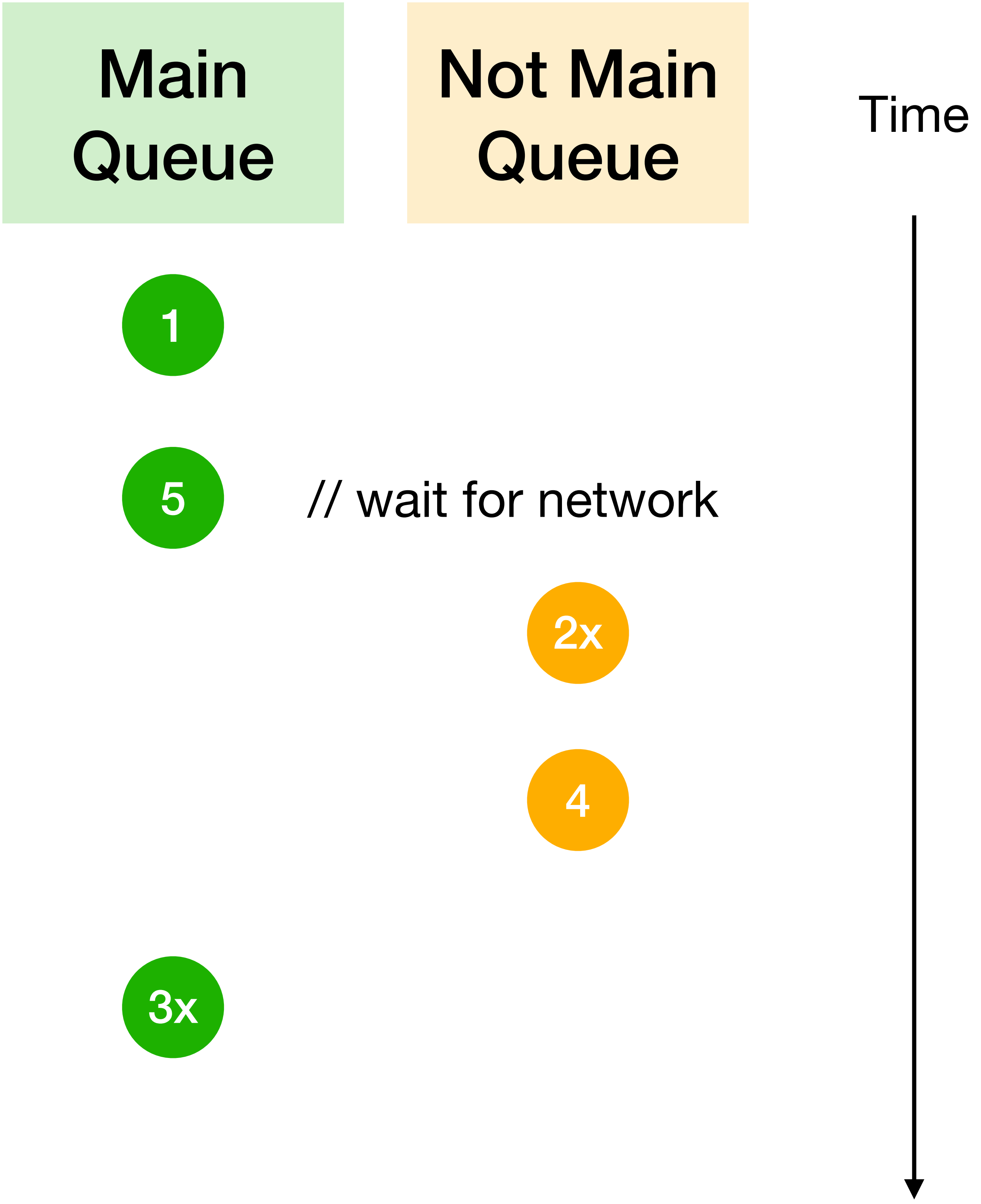


```
override func viewDidLoad() {
    super.viewDidLoad()

    let networkController = NetworkController()
    networkController.fetchPlanets { result in

        switch result {
        case .failure:
            DispatchQueue.main.async {
                let alertController = UIAlertController(title: NSLocalizedString("Impossible...",
                    comment: ""),
                                                        message: NSLocalizedString("Perhaps the
                                                            archives are incomplete", comment: ""),
                                                        preferredStyle: .alert)
                alertController.addAction(.init(title: NSLocalizedString("OK", comment: ""),
                                                style: .cancel,
                                                handler: nil))
                self.present(alertController, animated: true, completion: nil)
            }

        case .success(let planets):
            DispatchQueue.main.async {
                self.planets = planets
                self.tableView.reloadData()
            }
        }
    }
}
```



What is Unit Testing?

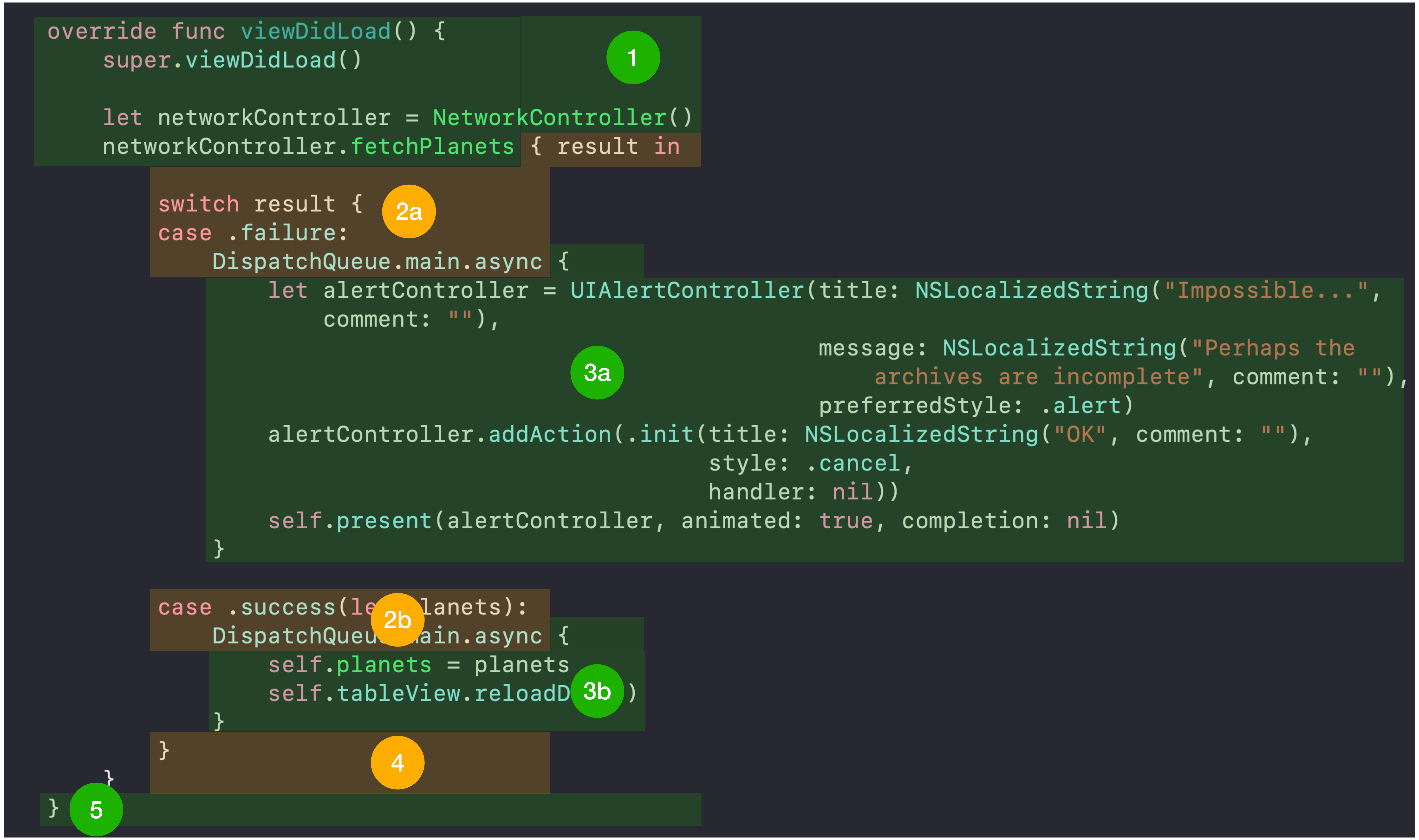
Synchronicity

How does XCTest work?

- all functions prefixed with `test` will be run as unit tests
- any function that does not finish execution is considered failed
    - the purpose is to validate each "logical" step using XCTAssert
- **any function that finishes execution is considered successful**

What is Unit Testing?

When does `func viewDidLoad` end?

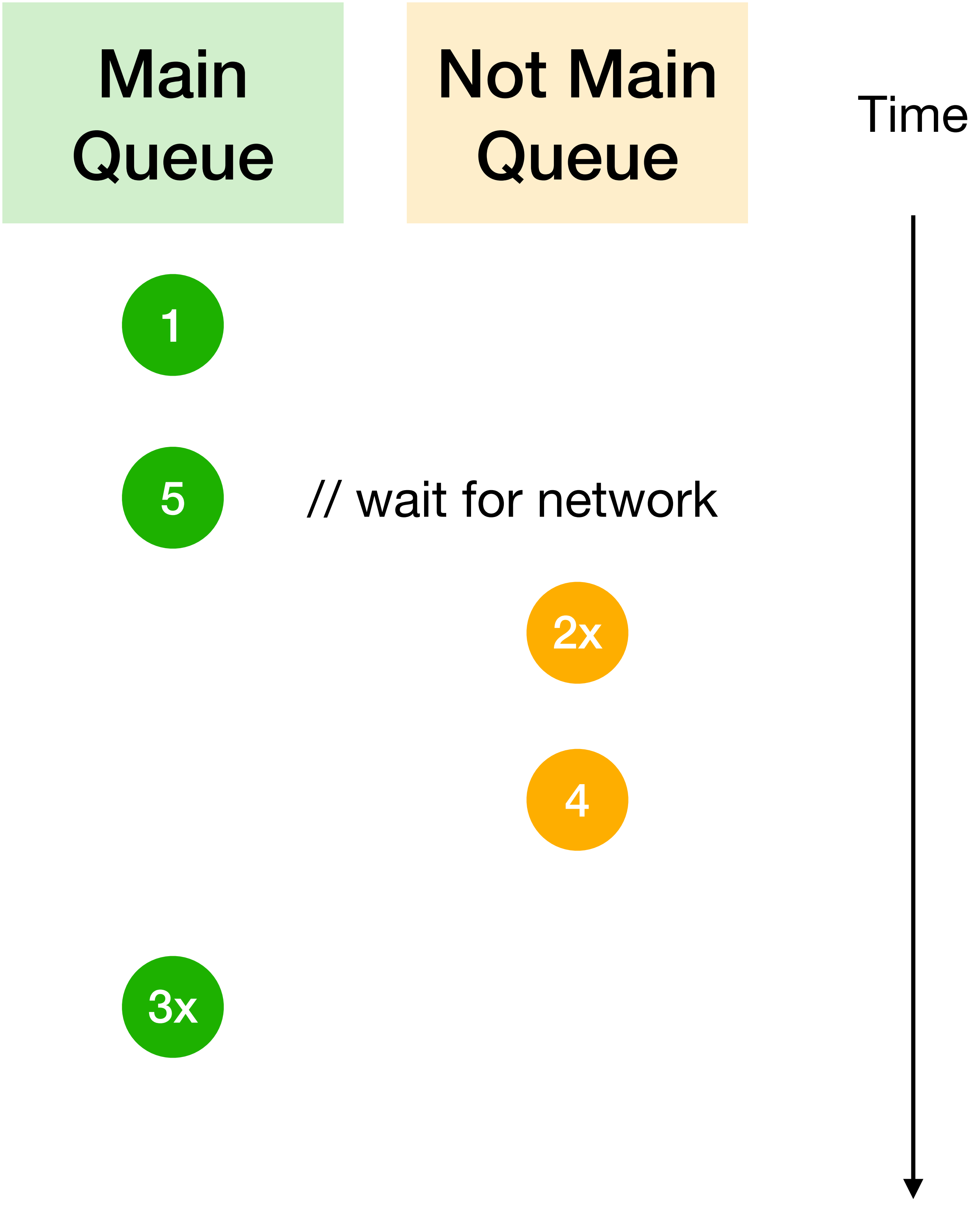


```
override func viewDidLoad() {
    super.viewDidLoad()

    let networkController = NetworkController()
    networkController.fetchPlanets { result in

        switch result {
        case .failure:
            DispatchQueue.main.async {
                let alertController = UIAlertController(title: NSLocalizedString("Impossible...",
                    comment: ""),
                                                        message: NSLocalizedString("Perhaps the
                                                            archives are incomplete", comment: ""),
                                                        preferredStyle: .alert)
                alertController.addAction(.init(title: NSLocalizedString("OK", comment: ""),
                                                style: .cancel,
                                                handler: nil))
                self.present(alertController, animated: true, completion: nil)
            }

        case .success(let planets):
            DispatchQueue.main.async {
                self.planets = planets
                self.tableView.reloadData()
            }
        }
    }
}
```



Live Demo

Test `fetchPlanets` asynchronously

What is Unit Testing?

Dependency Injection

- Using separation of concerns, a function should be connecting other behaviors or it should be processing, not both.
- Currently, fetchPlanets is doing two things:
  - Building a request based on inputs
  - Handing over the request so it goes to the network
  - Handing over the data to Codable so it is parsed

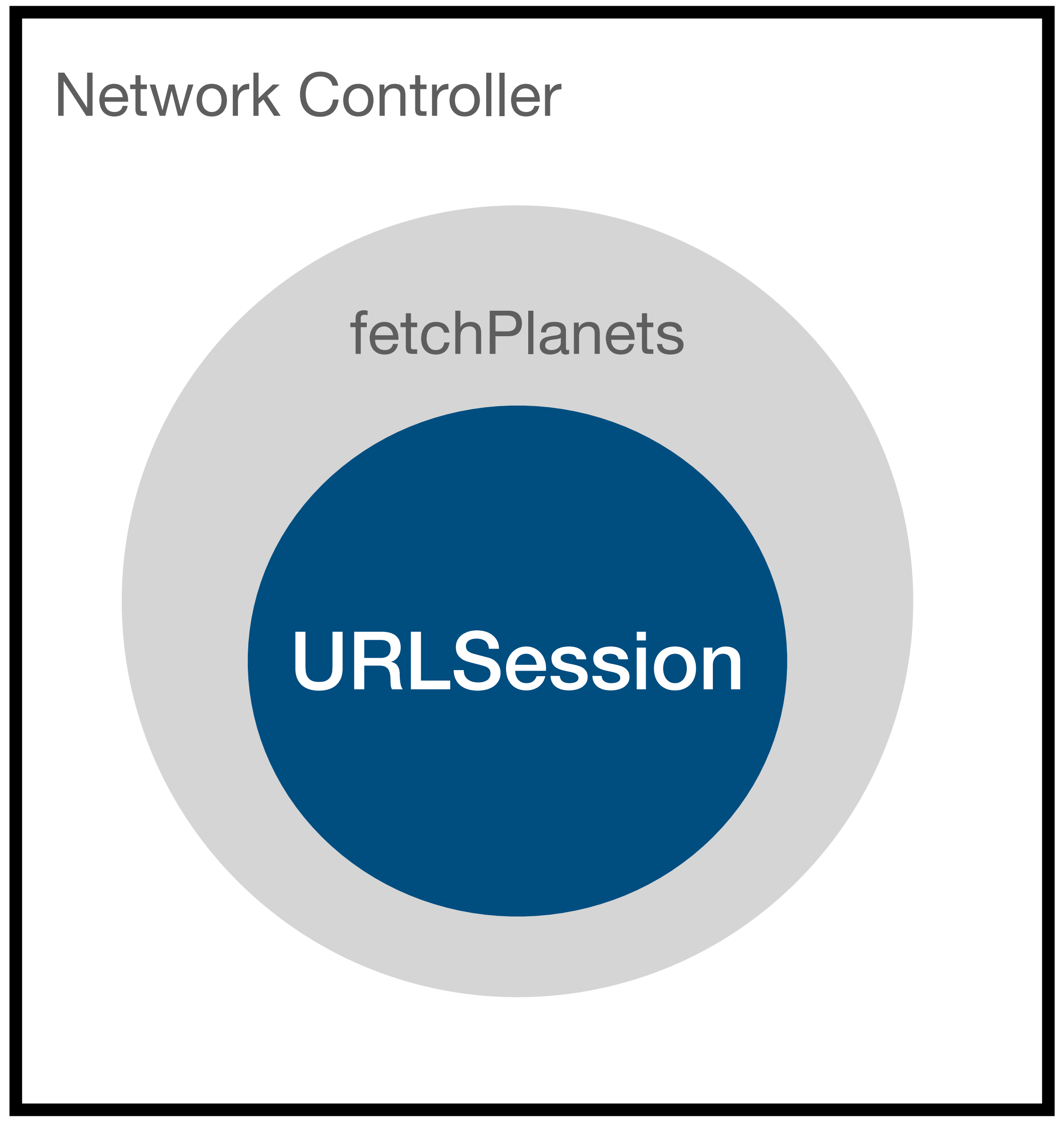
What is Unit Testing?

Dependency Injection

- Using separation of concerns, a function should be connecting other behaviors or it should be processing, not both.
- Currently, fetchPlanets is doing two things:
  - Building a request based on inputs
  - Handing over the request **so it goes to the network**
  - Handing over the data to Codable so it is parsed

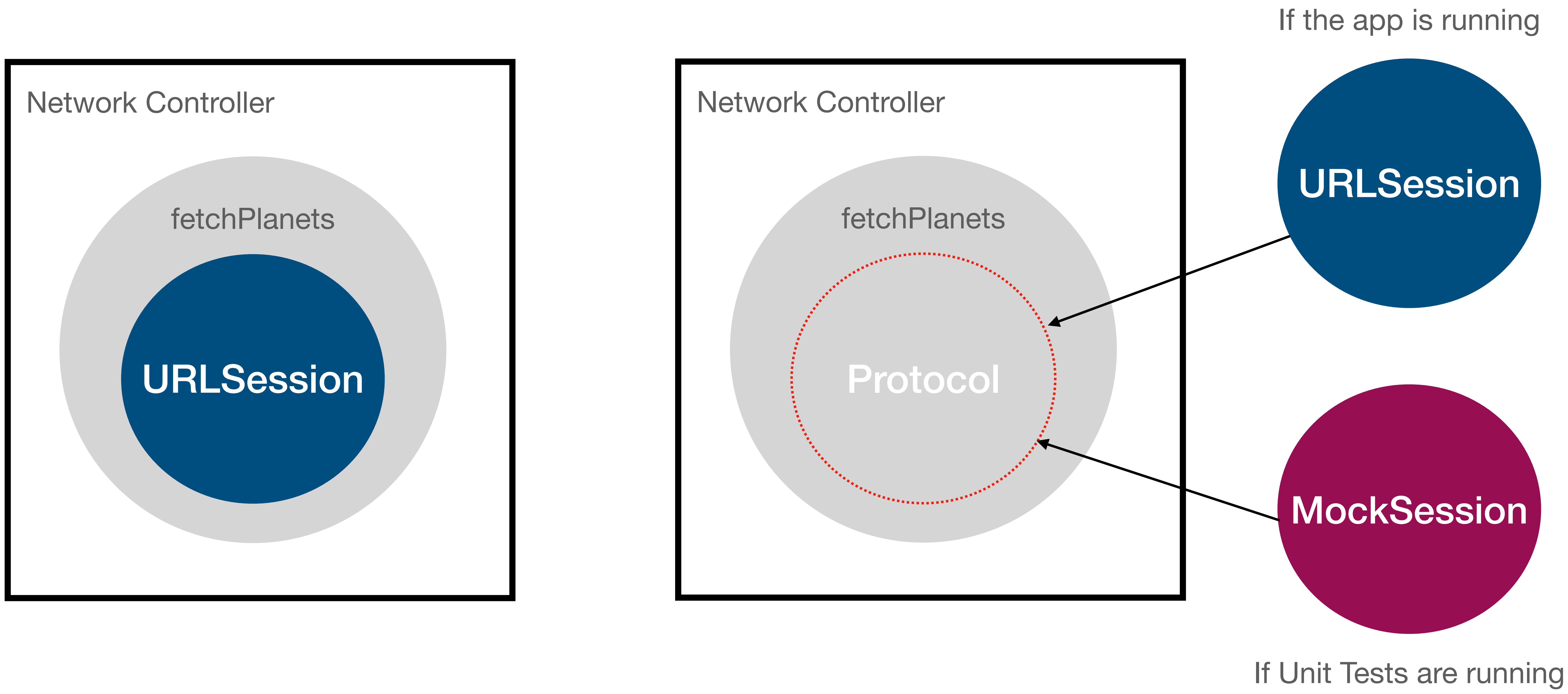
What is unit testing?

It's in a separate bundle



What is unit testing?

It's in a separate bundle



Next class

Use protocols to inject behavior

What is Unit Testing?

Where to go from here?

- Will be uploaded to https://github.com/olivaresf/unit_testing
- You can reach me @fromJrToSr
- Additional info
    - https://www.swiftbysundell.com/basics/unit-testing/
    - https://www.avanderlee.com/swift/unit-tests-best-practices/
    - http://blog.wilshipley.com/2005/09/unit-testing-is-teh-suck-urr.html
    - https://developer.apple.com/library/archive/documentation/ToolsLanguages/Conceptual/Xcode_Overview/UnitTesting.html

Support Fernando

I need to eat

- Practice Swift weekly with a 15-minute exercise: https://mailchi.mp/hey/weekly-swift-exercise-signup
- Help me improve the class: https://www.surveymonkey.com/r/3L2MRH6
- Follow me on Twitter: https://twitter.com/fromJrToSr
- Tweeting about the class!

Q&A