

Real World iOS Development

Felix J. Acero

iOS/OSX Developer

Demo App



News Digest

- Simple app
- Relies on Yahoo APIs
- Apple Design Award



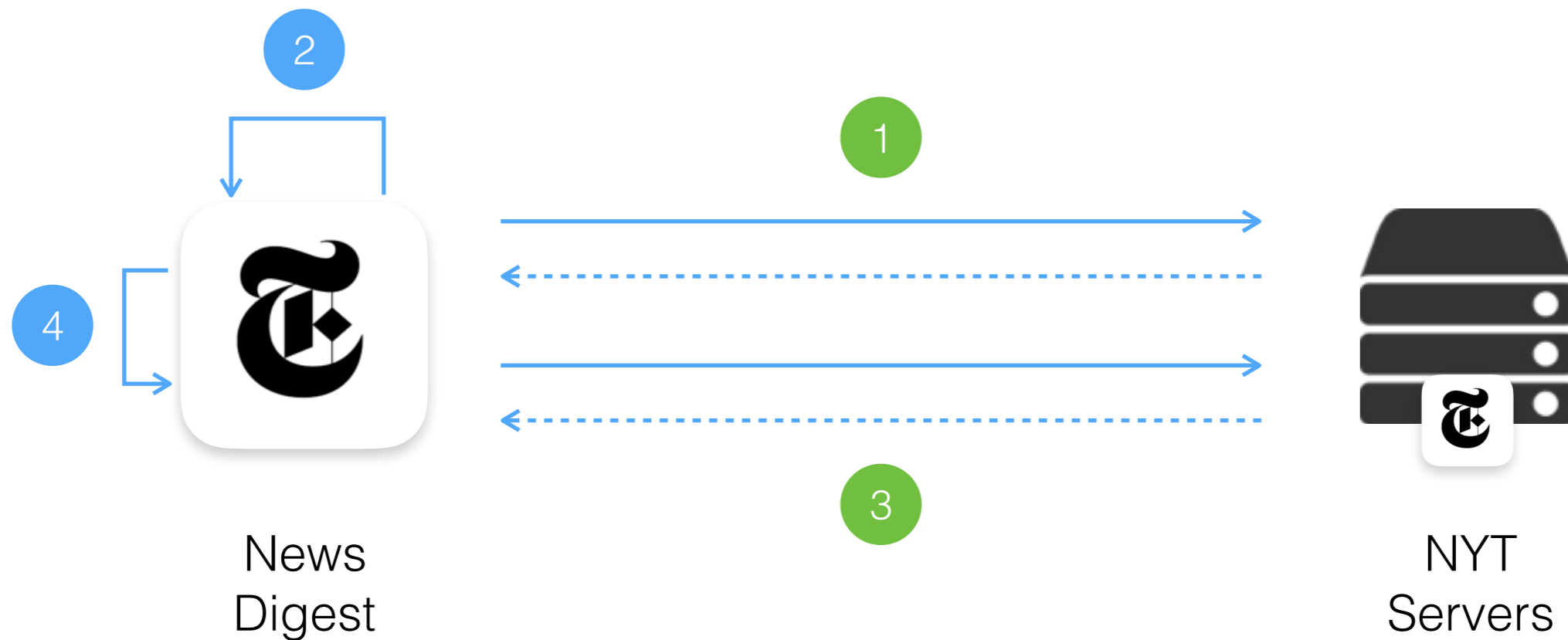
News Digest

- Proof of Concept
- Based on NYT APIs

DEMO

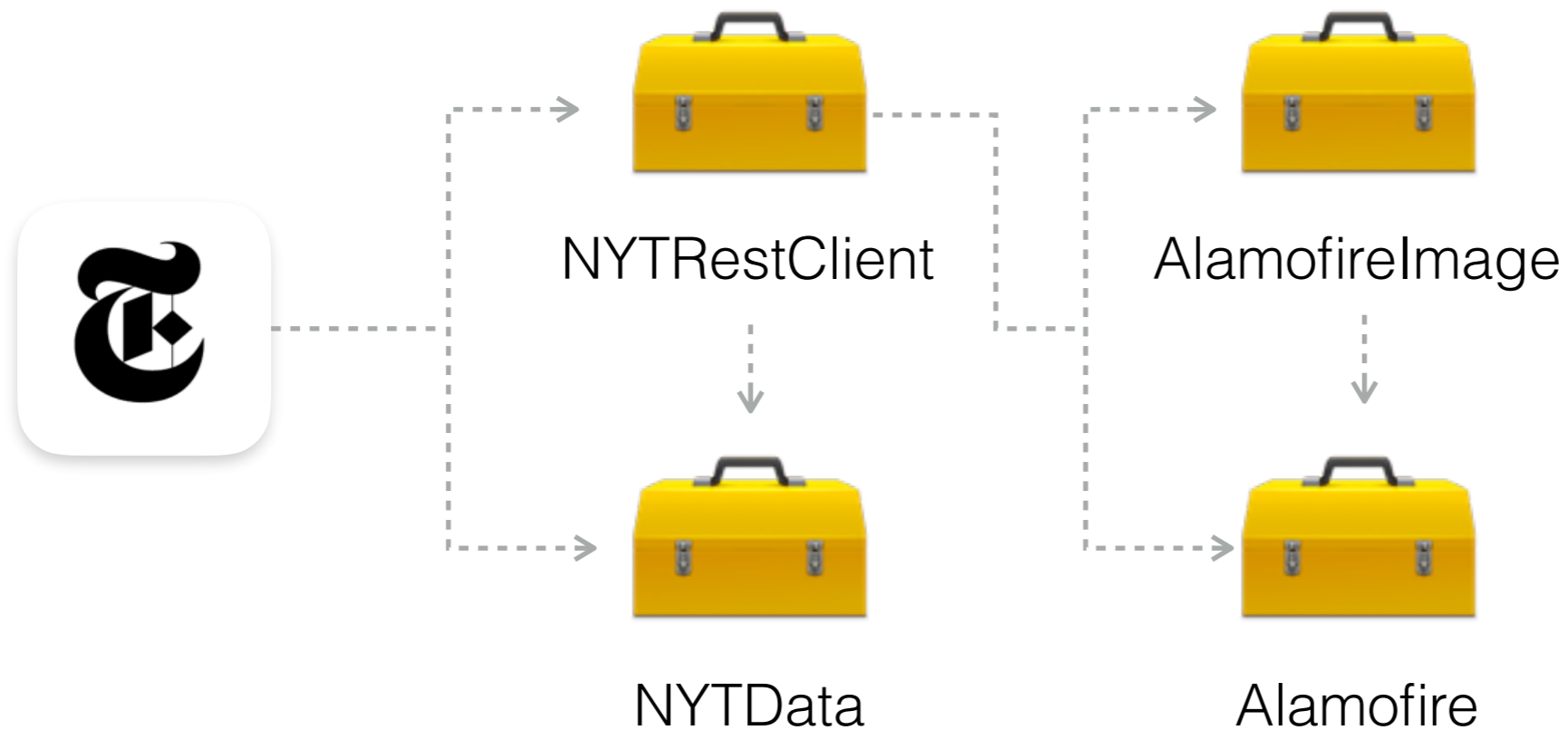
News Digest

Demo App – High level interactions



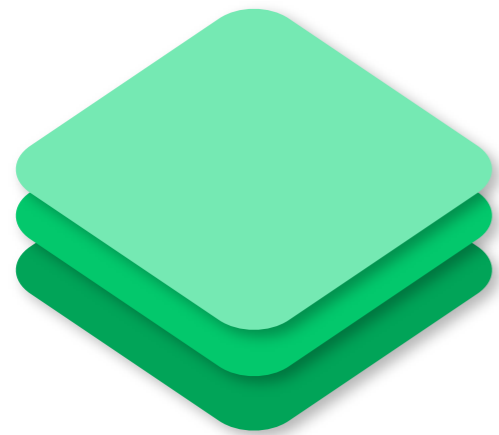
1. **Fetch** most viewed stories during the last week
2. **Format** retrieved data. Present list to the user
3. **Fetch** details and image for story ID: 123
4. **Format** retrieved data. Present details and image to the user

Demo **App** – Dependency Mesh



- [NYTRestClient](#) - Mediates communications with the NYT servers
- [NYTData](#) - Contains basic data structures - e.g NYTStory
- [AlamoFireImage](#) - An image component library for Alamofire
- [AlamoFire](#) - Elegant HTTP Networking in Swift

Real World iOS Development



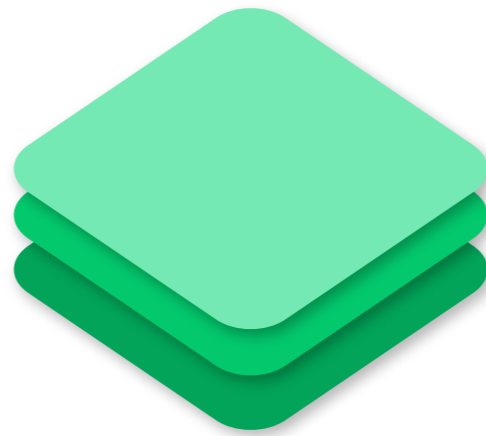
Dependency
Management



Testing



Continuous
Integration



“With great projects come
huge dependencies”

– Anonymous

Dependency Management

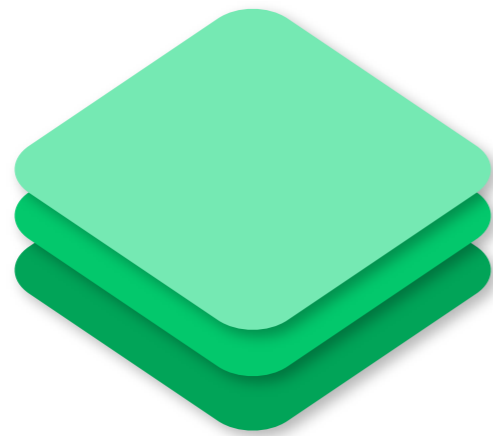
WHAT IS A DEPENDENCY MANAGER?

A tool that automates the process of

- Installing
- Upgrading
- Configuring

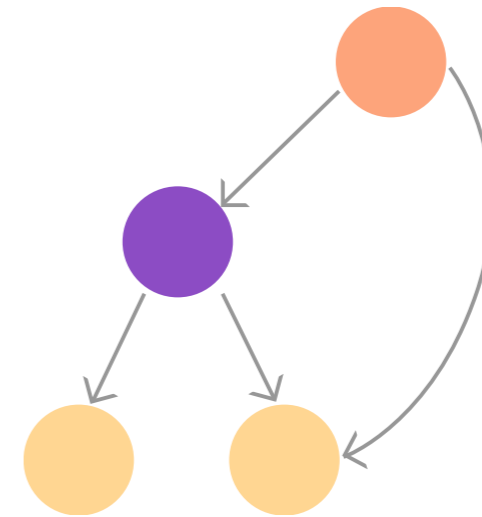
... software packages

Dependency Management – Under the Hood



Package

Contains the source files and metadata (name, description, version, etc)



Dependency Analyser

Resolves the dependency graph respecting the version requirements

Package Managers for Swift



Cocoa pods

- First one to appear
- Easy to use
- Modifies your project setup



Carthage

- More recent
- Minimalistic
- Requires more manual steps



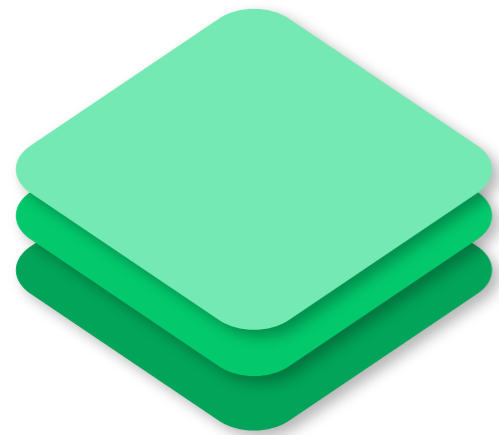
Swift Package Man.

- On its early days
- Will hopefully become standard
- swift.org project

DEMO

Dependency Management

Real World iOS Development



Dependency
Management



Testing



Continuous
Integration



“Program testing can be a very effective way to show the presence of bugs, but is hopelessly inadequate for showing their absence”

– Edsger Dijkstra

Software Testing

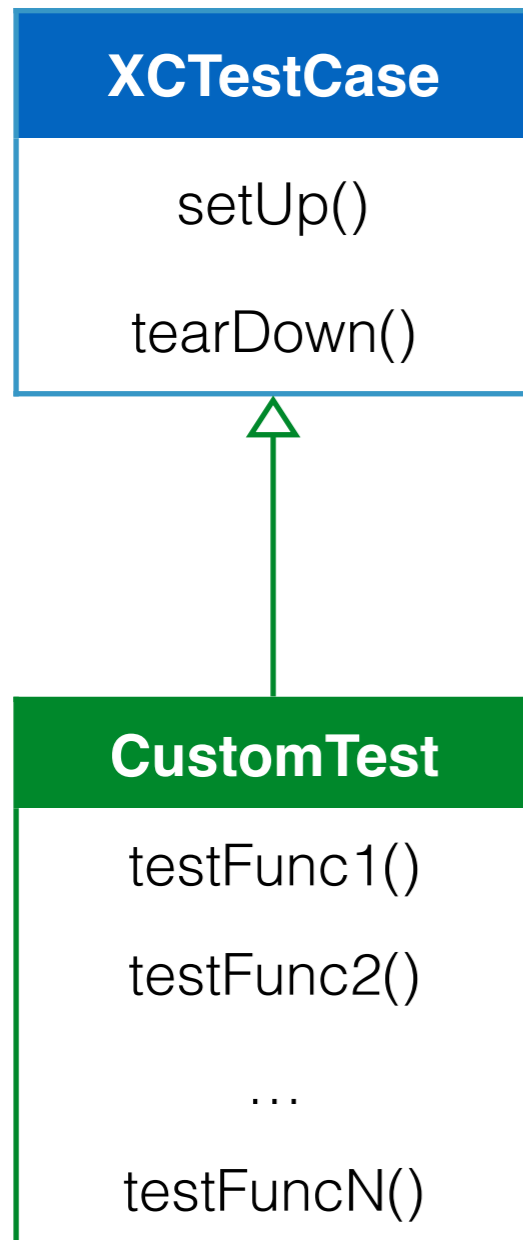
WHAT SOFTWARE TESTING **IS**

- Activity aimed at evaluating an attribute or capability of a program
- The purpose of testing is to show that the software works

WHAT SOFTWARE TESTING **IS NOT**

- Testing does **NOT** demonstrate the absence of bugs
- Testing is Quality Assurance **NOT** Quality Insertion

Testing in Xcode 7

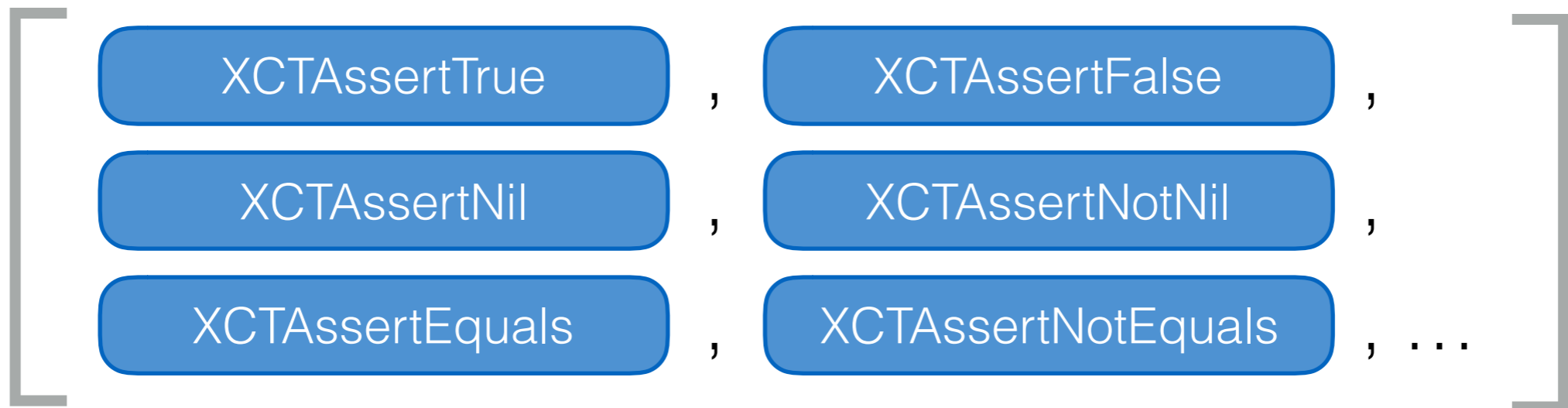


- Test are grouped into classes that subclass `XCTestCase`
- Every test has a `setUp` and `tearDown` phase:
 - `setUp()`
 - `testFunc()`
 - `tearDown()`
- Each method name should start with the word: `test` – e.g. `testSum`
- Each method should contain at least one assertion

Testing in Xcode 7

Assertions are boolean expressions that check certain aspect of you code

XCTAssert



XCTFail

Testing in Xcode 7

```
func testSum() {  
    // setUp()  
  
    // Given: A calculator  
    let calculator = Calculator()  
  
    // When: Summing two integers  
    let result = calculator.sum(2,3)  
  
    // Then: It should return their arithmetic sum  
    XCTAssertEqual(result, 5, "Incorrect result")  
  
    // tearDown()  
}
```

FIRST Principle

FAST

Test should be able to be executed often

ISOLATED

Tests on their own cannot depend on external factors or on the results of other tests

REPEATABLE

Test should have the same result every time you run them

SELF-VERIFIABLE

Test should include assertions. No human intervention should be needed

TIMELY

Tests should be written along with (before) production code

TDD Workflow (red > green > refactor)

- 1** Write a failing test – red
- 2** Write enough code to make it pass – green
- 3** Refactor your code – refactor
- 4** Rinse and repeat :)

TDD Benefits

- Focus your work:
 - Know what you are trying to build before you build it
 - Know when you are done
 - Modify only the pieces that are required to make your tests pass
- Refactor your code without worrying about breaking something

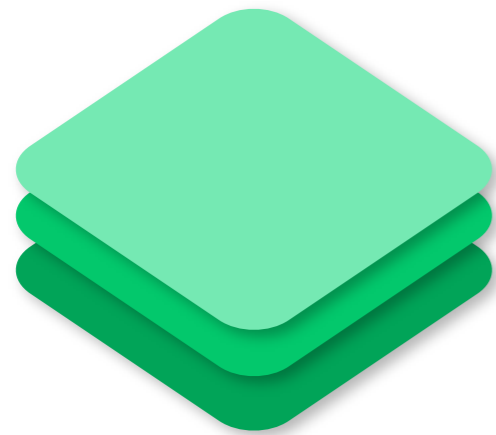
TDD Benefits

- Write code that is easier to use
 - Your test becomes the first client of your code
 - Your test acts as an example of how to use your code
- Write code that is easier to maintain
 - “Breaking changes” should make your tests fail

DEMO

Unit Testing

Real World iOS Development



Dependency
Management



Testing



Continuous
Integration



“I don’t care if it works on your machine! We are not shipping your machine!”

– Vidiu Platon

Continuous Integration (CI)

WHAT IS CONTINUOUS INTEGRATION ?

- A set of practices that require developers to:
 - Integrate their code in a shared code repository
 - Create and maintain tests for their code
 - Automate the application's build process

WHAT IS THE GOAL OF CONTINUOUS INTEGRATION ?

- Improve collaboration within the developer team
- Increase software quality by detecting errors as early as possible

Continuous Integration Tools



Xcode Server

- Developed by Apple
- Deeply integrated with Xcode
- Some rough edges



Jenkins

- Cross-platform tool
- Support for iOS/OSX through a plugin
- Well documented



Travis

- Cross-platform tool
- Free for open source projects
- Github integration

Continuous Integration with Xcode Server



- Automates the build and test process
- Performs static analysis and archives your code
- Can test your app against a suite of devices
- Gathers interesting data - e.g test coverage, build history, etc
- Produces reports and notifications about the integration outcomes
- Gives you hooks to perform custom actions

Continuous Integration in Xcode 7



Scheme

Recipe for building your project



Bot

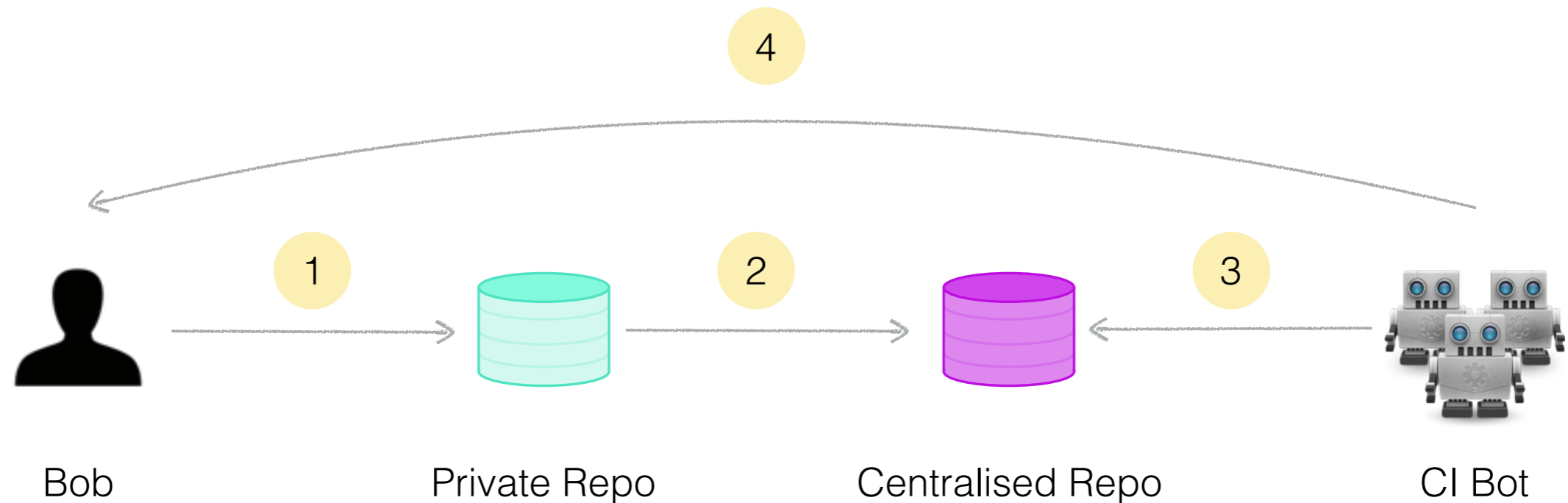
Agent that analyses, builds and archives your project



Integration

A single run from a bot

Continuous Integration Workflow

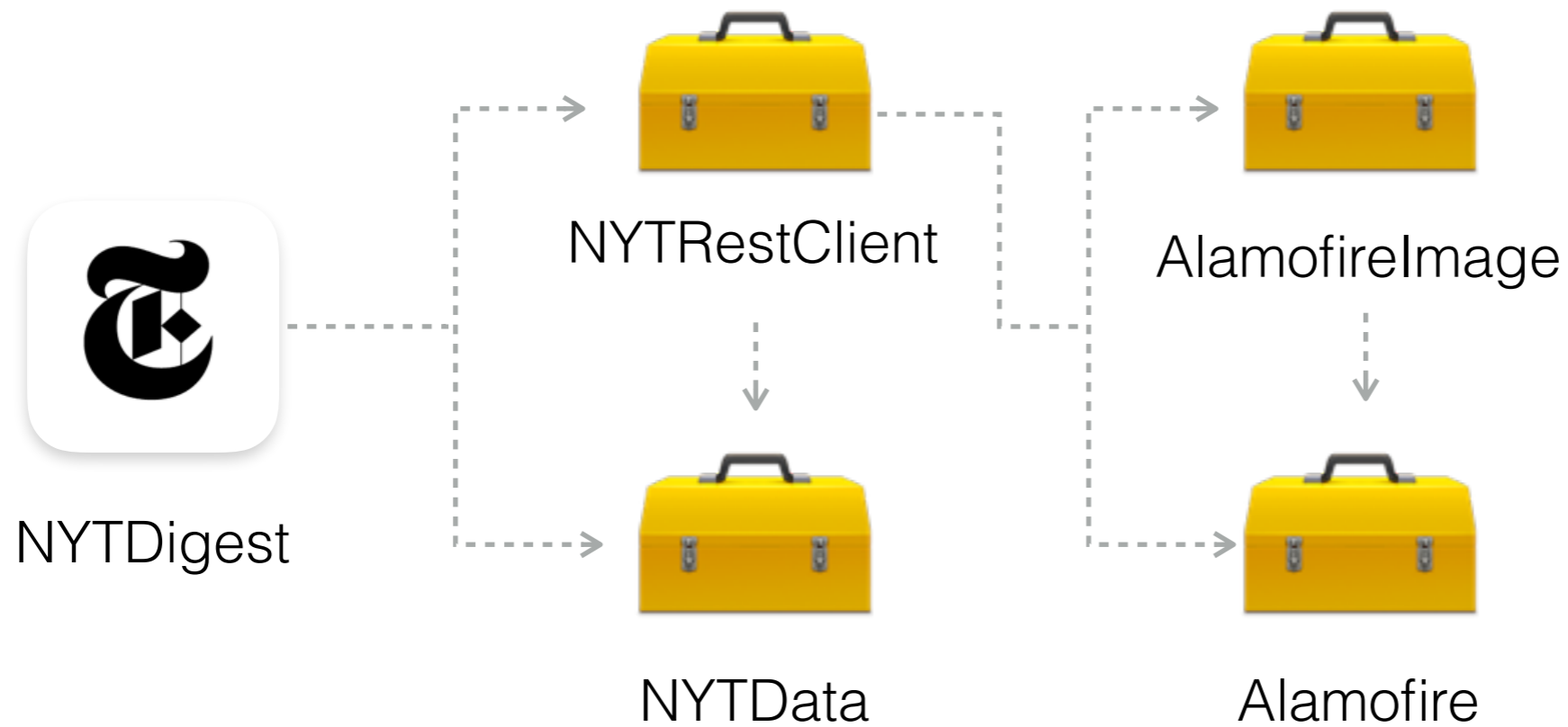


1. Developer works on a feature and commits to her [local repo](#)
2. When done, developer pushes changes to the [centralised repo](#)
3. CI-Bot checks out the repo, builds, tests and archives the result
4. CI-Bot informs developers about the [integration](#) outcome

DEMO

Continuous Integration

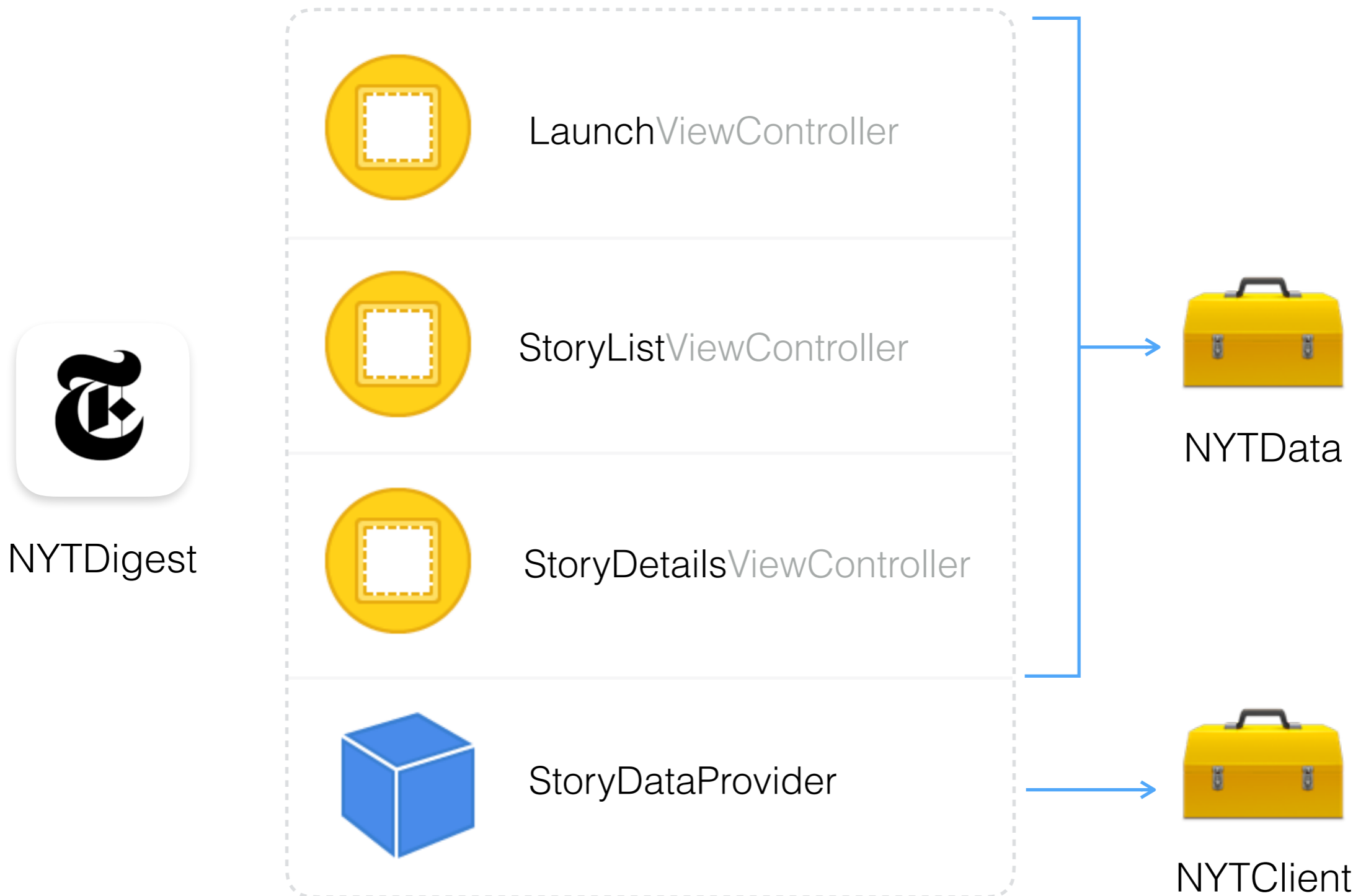
Demo App – UI



NYTDigest

- Capture user interactions
- Format data and present it on the screen

Demo App – UI



DEMO

NYTDigest

Repositories

Project	Repository
NYTData	<u>https://github.com/acerosalazar/NYTData</u>
NYTClient	<u>https://github.com/acerosalazar/NYTClient</u>
NYTDigest	<u>https://github.com/acerosalazar/NYTDigest</u>