Nullable Infrastructure

A Drop-In Replacement for Mocks

⚠️ **WARNING: Coding Ahead**

Before we start:

1) Get source: github.com/jamesshore/agile2022
2) Install Node.js (v16.16.0)
3) `git checkout integration`
4) Run the `build` script

Presented by James Shore

Agile 2022, Nashville Tennessee

July 20, 2022

Web jamesshore.com **Email** jshore@jamesshore.com **Twitter** @jamesshore

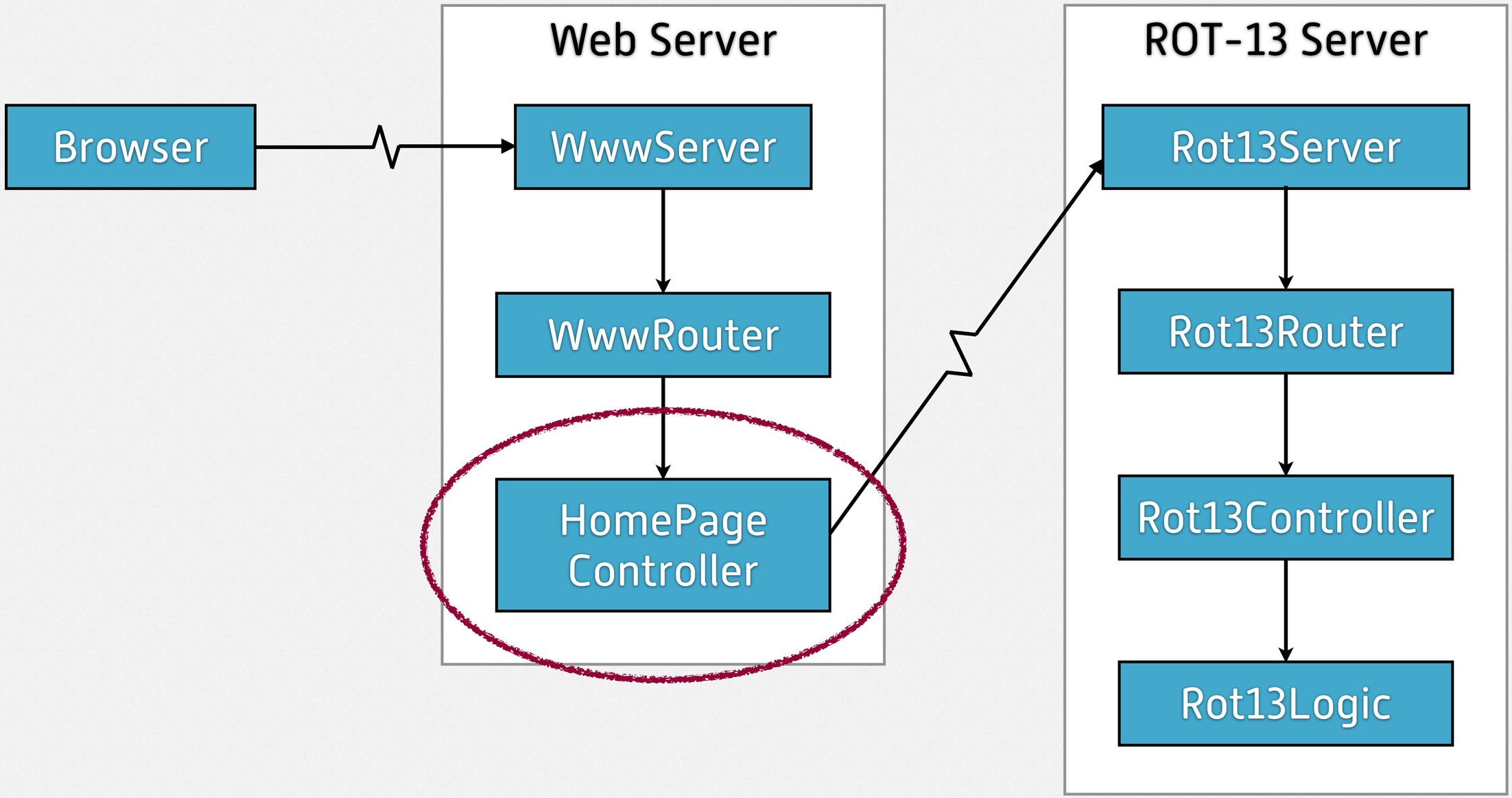
LinkedIn linkedin.com/in/jdlshore **Github** github.com/jamesshore

Nullable Infrastructure

A Drop-In Replacement for Mocks
That Solves Their Problems
While Introducing New Problems of Its Own
Because, After All
Nothing Is Perfect
And Everything Has Tradeoffs
Because, Hey, That's Engineering

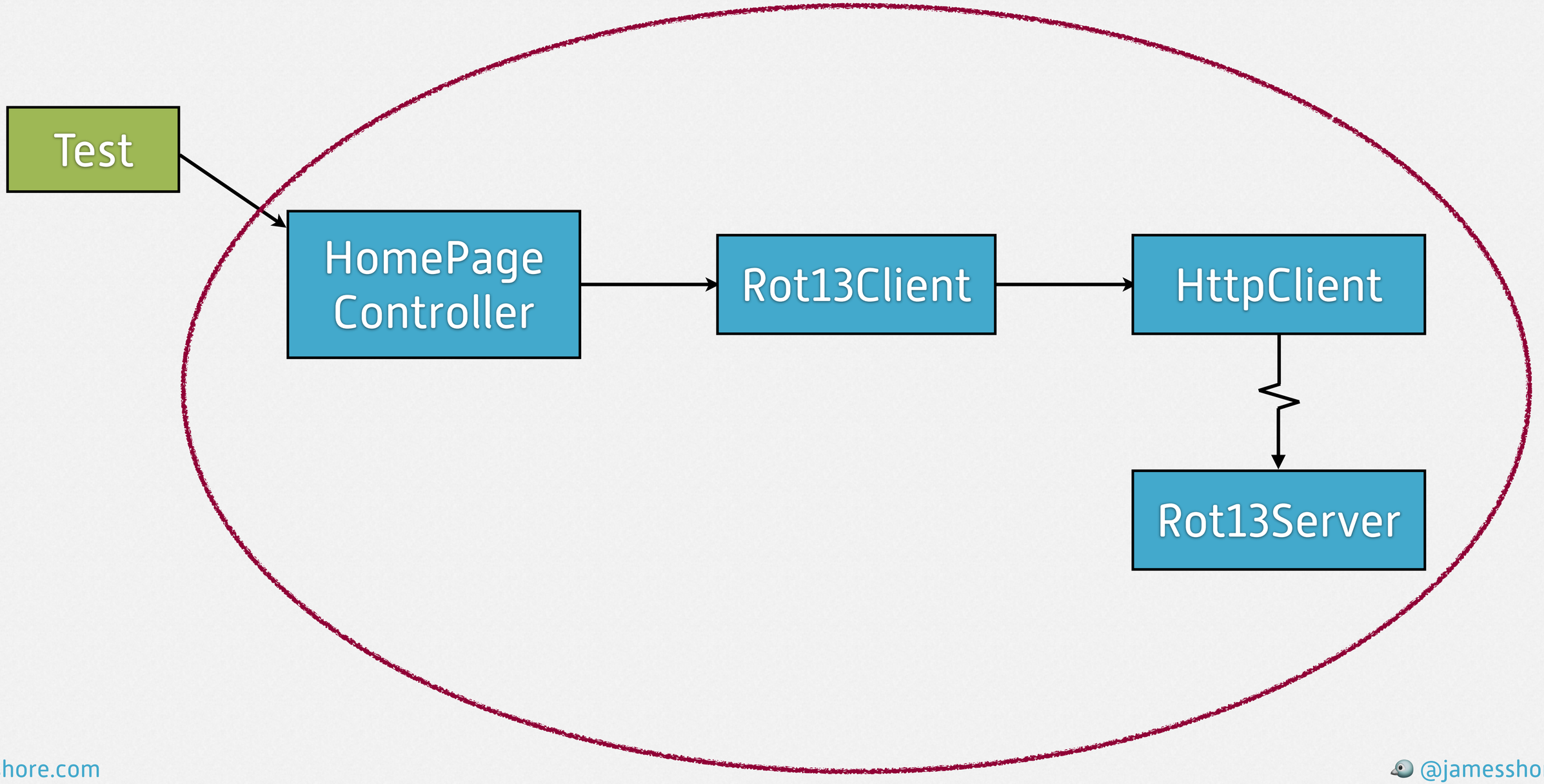
Web jamesshore.com **Email** jshore@jamesshore.com **Twitter** @jamesshore
LinkedIn linkedin.com/in/jdlshore **Github** github.com/jamesshore

1 The Challenge

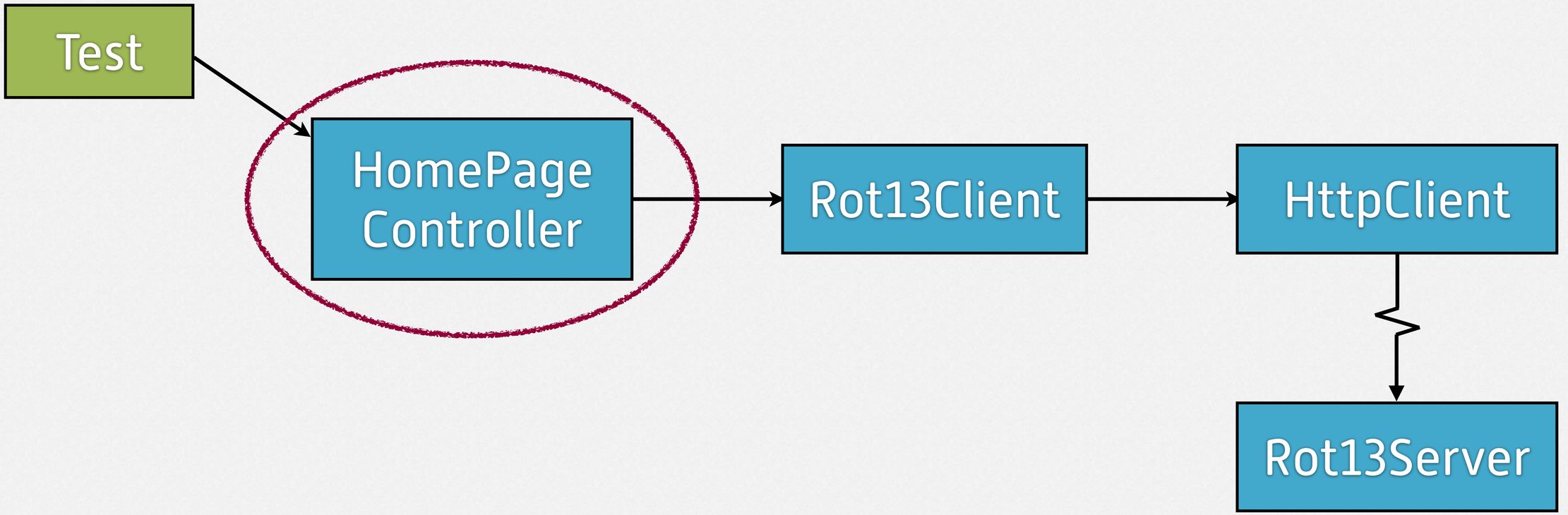

Web Server
ROT-13 Server
Browser
WwwServer
WwwRouter
HomePage Controller
Rot13Server
Rot13Router
Rot13Controller
Rot13Logic

2 Types of Tests

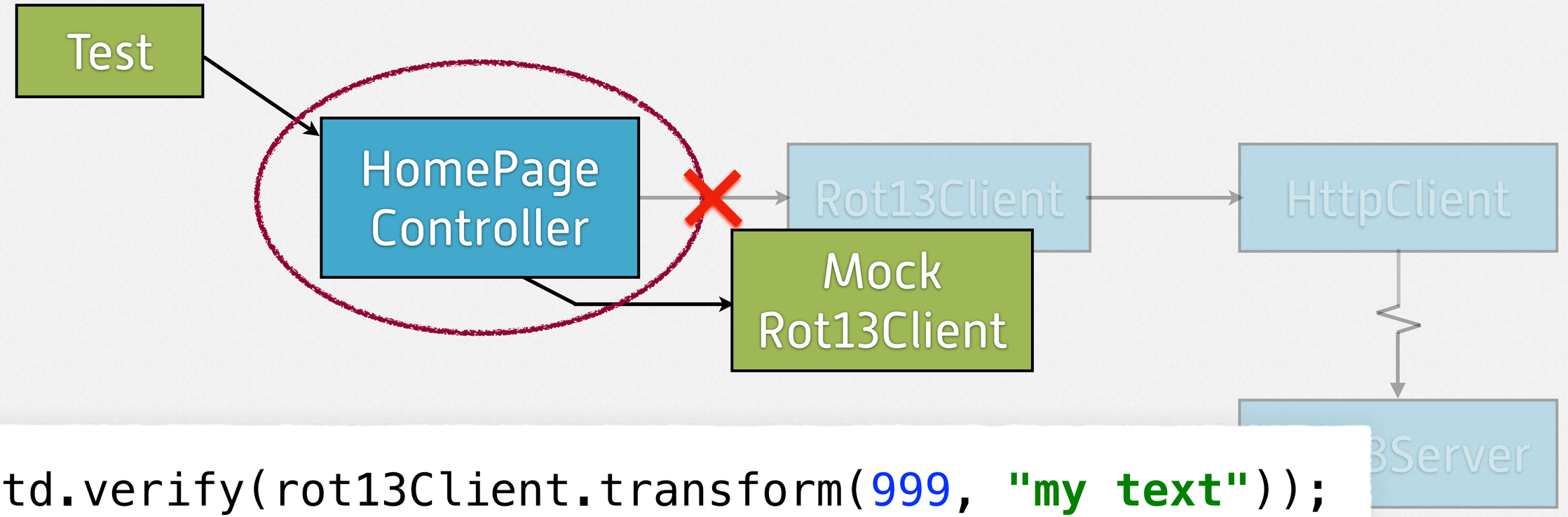
Broad Tests


Test
HomePage Controller
Rot13Client
HttpClient
Rot13Server

Narrow Tests



Solitary Interaction-Based Test




```
td.verify(rot13Client.transform(999, "my text"));
```

Problems with Mocks/Spies

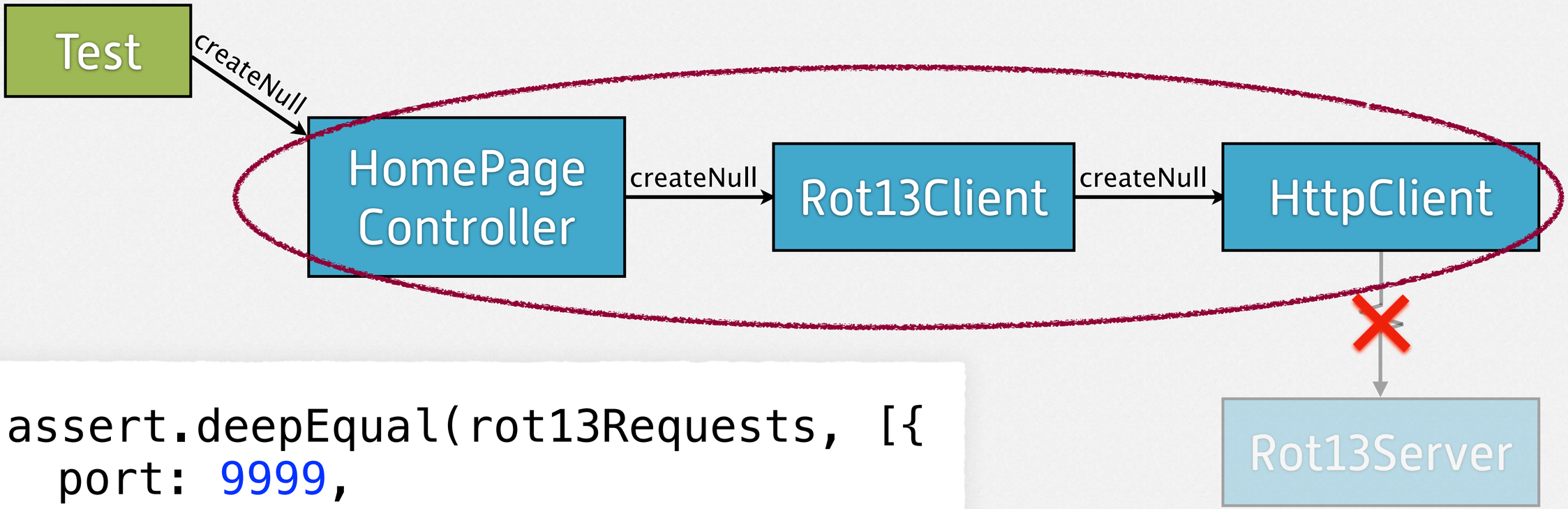
Solitary:

- When a dependency has a **breaking semantic change**, the tests should fail, **but don't.**

Interaction-Based:

- When a dependency **refactors its interface**, the tests shouldn't fail, **but do.**

Sociable State-Based Test




```
assert.deepEqual(rot13Requests, [{
  port: 9999,
  text: "my_text",
}]);
```

Solves the Problems with Mocks/Spies

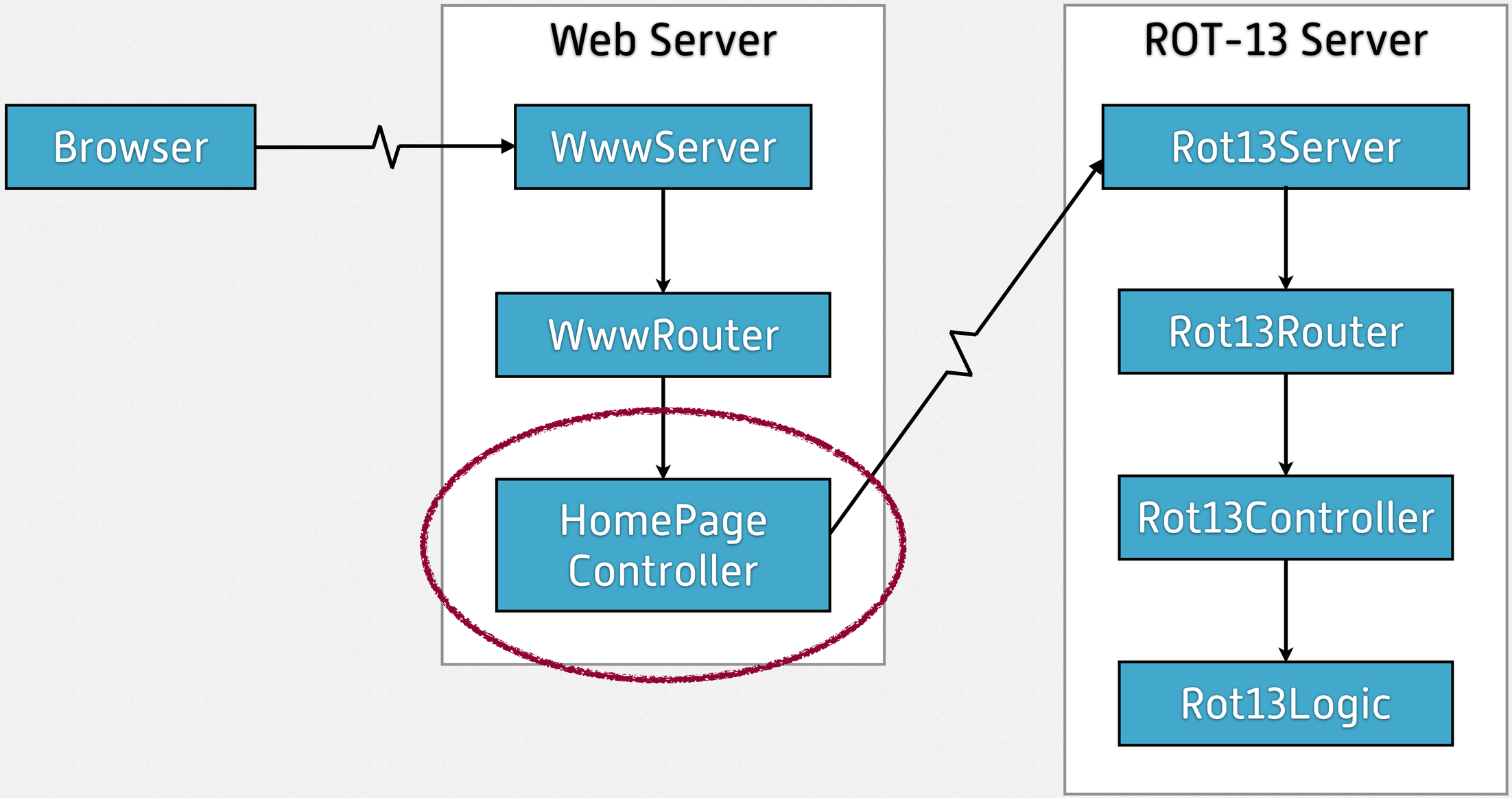
Sociable:

- When a dependency has a **breaking semantic change**, the tests should fail, **and do.**

State-Based:

- When a dependency **refactors its interface**, the tests shouldn't fail, **and don't.**

3 Using Nullable Infrastructure


Web Server
ROT-13 Server
Browser
WwwServer
WwwRouter
HomePage Controller
Rot13Server
Rot13Router
Rot13Controller
Rot13Logic

Exercise: Nullable Infrastructure

1. Get Node.js and the repo code from the USB stick.
2. Check out the “challenge_nullable” branch and run the watch script:

   git checkout challenge_nullable

   watch quick (on Windows) or **./watch.sh quick** (on Mac/Linux)
3. Open the exercise files in your favorite code editor:

   src/www/home_page/_home_page_controller_null_test.js

   src/www/home_page/home_page_controller.js
4. Follow the instructions in the test file. Pair, mob, or work individually as desired.
5. Raise your hand or come get me if you need help.

Demo: Nullable Infrastructure

Branch: **challenge_nullable**

4 From Mocks to Nullables

Demo: Mocks

Branch: **challenge_mock**

Converting Mocks to Nullables

Convert one mock variable at a time

- td.instance(Class) → Class.createNull()
- td.when() → Class.createNull(**config**)
- td.verify() → instance.trackXxx() & assertions

Demo: Mocks to Nullables

Branch: **challenge_mock_to_null**

5 Creating Nullables

To Create a Nullable Infrastructure Wrapper

1. Start with an infrastructure wrapper (aka “gateway”)
2. Test-drive a “createNull()” factory
3. Test-drive “createNull(config)” as needed
4. Test-drive “trackXxx()” methods as needed

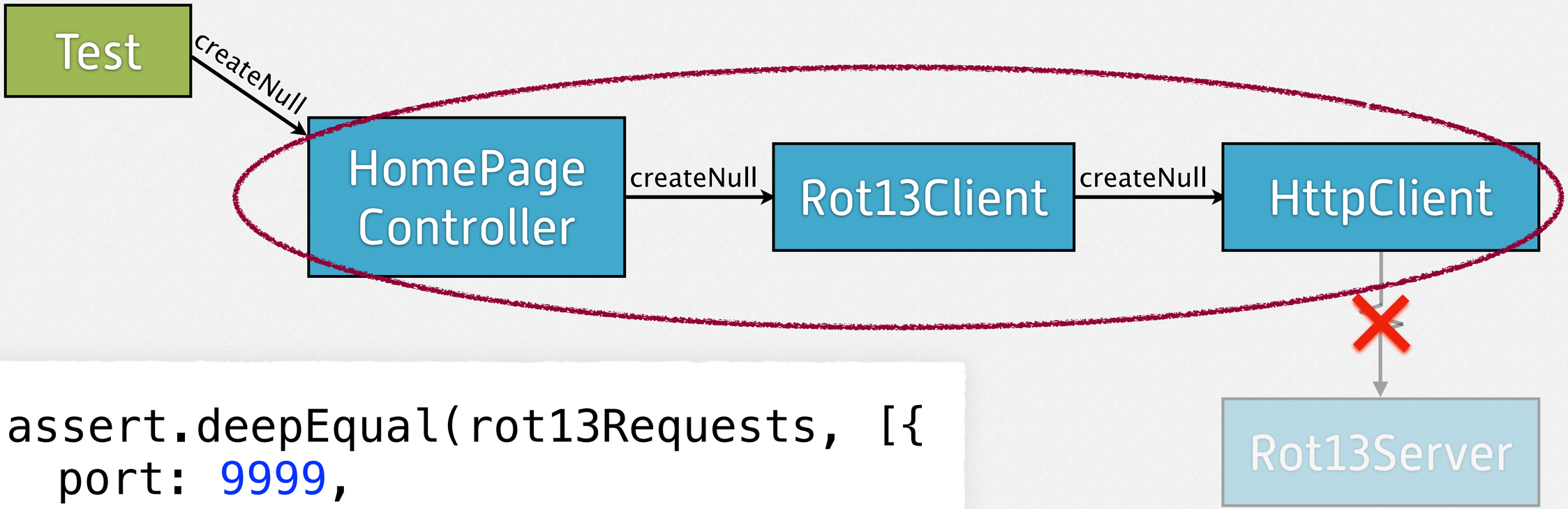
This is **production-grade** code.

Demo: Making Infrastructure Nullable

Take-home exercise branch: **challenge_infrastructure**

6 Summary

Sociable State-Based Test




```
assert.deepEqual(rot13Requests, [{
  port: 9999,
  text: "my_text",
}]);
```

Converting Mocks to Nullables

Convert one mock variable at a time

- td.instance(Class) → Class.createNull()
- td.when() → Class.createNull(**config**)
- td.verify() → instance.trackXxx() & assertions

To Create a Nullable Infrastructure Wrapper

1. Start with an infrastructure wrapper (aka “gateway”)
2. Test-drive a “createNull()” factory
3. Test-drive “createNull(config)” as needed
4. Test-drive “trackXxx()” methods as needed

This is **production-grade** code.

Tradeoffs of Nullable Infrastructure

Benefits:

- Tests **do** fail when you **break** dependencies…
- …and **don't** fail when you **refactor** dependencies.
- Tests easier to **read, write, and abstract**
- **No "magic"** DI frameworks or mocking tools

Requires changing production code:

- Configurable "createNull()" factory
- Assertable state
- Embedded stub in lowest-level infrastructure

For More Information

"Testing Without Mocks: A Pattern Language":
jamesshore.com/s/**nomocks**

"TDD Lunch and Learn" videos:
jamesshore.com/s/**lnl**

Or Search:
James Shore Testing Without Mocks

Nullable Infrastructure

A Drop-In Replacement for Mocks
That Solves Their Problems
While Introducing New Problems of Its Own
Because, After All
Nothing Is Perfect
And Everything Has Tradeoffs
Because, Hey, That's Engineering

Web jamesshore.com **Email** jshore@jamesshore.com **Twitter** @jamesshore
LinkedIn linkedin.com/in/jdlshore **Github** github.com/jamesshore

Nullable Infrastructure

A Drop-In Replacement for Mocks

Presented by James Shore

Agile 2022, Nashville Tennessee

July 20, 2022

Web jamesshore.com **Email** jshore@jamesshore.com **Twitter** @jamesshore

LinkedIn linkedin.com/in/jdlshore **Github** github.com/jamesshore