Geb to Grid

Ralph A. Navarro Jr.
PH: 508-287-0190
GitHub: https://github.com/eaglet3d/GebToGrid

Geb to Grid

Geb to Grid

Overview

1st Half

- Problem
- Groovy
- Spock
- Selenium
- Geb

2nd Half

- Docker
- Docker Compose
- Selenium Grid
- Putting it together
- Q&A

Geb to Grid

Overview

1st Half

- **Problem**
- Groovy
- Spock
- Selenium
- Geb

2nd Half

- Docker
- Docker Compose
- Selenium Grid
- Putting it together
- Q&A

Geb to Grid
Simplified SDLC

1) Create requirements
2) Develop code
3) **Test to verify against requirements (PASS/FAIL)**
4) Repeat 2+3 until all requirements are met
5) Send to production

Geb to Grid Problem

When testing your web application against a browser...

- Manual testing is too slow
  - One Tester with one browser
- Need way to automate testing
- Test against many browsers
- Test in parallel

Requirement

Use Case

- When user navigates to Login page
- Then Login page is displayed
- When user provides valid Admin credentials and clicks on login button
- Then user is taken to Admin page

Not Geb

```
import org.openqa.selenium.*;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;

public class LogIn_Page {
    private static WebElement element = null;
    public static WebElement txtbx_UserName(WebDriver driver){
        element = driver.findElement(By.id("log"));
        return element;
        }
      public static WebElement txtbx_Password(WebDriver driver){
        element = driver.findElement(By.id("pwd"));
        return element;
        }
      public static WebElement btn_LogIn(WebDriver driver){
        element = driver.findElement(By.id("login"));
        return element;
        }
 }
```

Problem

- Automated Web test development effort is similar to application development effort.
- Wouldn’t it be great if tests could align closer to the requirements?
- How can we improve test development and execution efficiency?

Geb to Grid

Overview

1st Half

- Problem
- **Groovy**
- Spock
- Selenium
- Geb

2nd Half

- Docker
- Docker Compose
- Selenium Grid
- Putting it together
- Q&A

Groovy

http://groovy-lang.org

Groovy

Why Groovy?

- Design Goals
  - ✔ Improve developer productivity
  - ✔ Scripting and testing capabilities
  - ✔ Write concise and maintainable
    - ✔ Tests
    - ✔ Builds
    - ✔ Automation
  - ✔ Smooth Java integration

Groovy

Why Groovy?

- Features
  - Closures
  - builders
  - Runtime & Compile time Meta-programming
  - Domain Specific Language Authoring

Groovy

Why Groovy?

- Features
  - **Closures**
  - builders
  - Runtime & Compile time Meta-programming
  - Domain Specific Language Authoring

Groovy

Closures

A closure in Groovy is:

- An open, anonymous, block of code
- Can take arguments
- Return a value
- And be assigned to a variable
- May reference variables in its surrounding scope
- **Can also contain free variables defined outside of its surrounding scope**

Groovy

Closures

- Example

  [0, 1, 2, 3, 4, 5].each { print it }

- Output:

  012345

Groovy

Why Groovy?

- Features
  - Closures
  - **builders**
  - Runtime & Compile time Meta-programming
  - Domain Specific Language Authoring

Groovy

Builders

- Simplifies creation of data structures
- Includes JsonBuilder and MarkupBuilder out of the box
- Or build your own builder

Groovy

JsonBuilder

Example:

```
def json = new groovy.json.JsonBuilder()

json name: "Guillaume", age: 33

assert json.toString() == '{"name":"Guillaume","age":33}'
```

Groovy

Why Groovy?

- Features
  - Closures
  - builders
  - **Runtime & Compile time Meta-programming**
  - Domain Specific Language Authoring

Groovy

Meta-programming

- Add methods to classes or objects, dynamically at runtime
- Including Java clsses, or final Java classes

Comes with built-in methods or add your own.

Groovy

Meta-programming

Example:

“http://google.com”.toURL().text

Returns:

The Google home page html as text

<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en"><head><meta content="Search the world's information, including webpages, images, videos and more. Google has many special features to help you find exactly what ...

Groovy

Why Groovy?

- Features
  - Closures
  - builders
  - Runtime & Compile time Meta-programming
  - **Domain Specific Language Authoring**

Groovy

DSL

"*A DSL is a mini-language aiming at representing constructs for a given domain.*"

Fergal Dearle

"Groovy for Domain-Specific Languages"

Groovy

DSL

Spock and Geb are mini-languages

Thanks to Groovy

Groovy

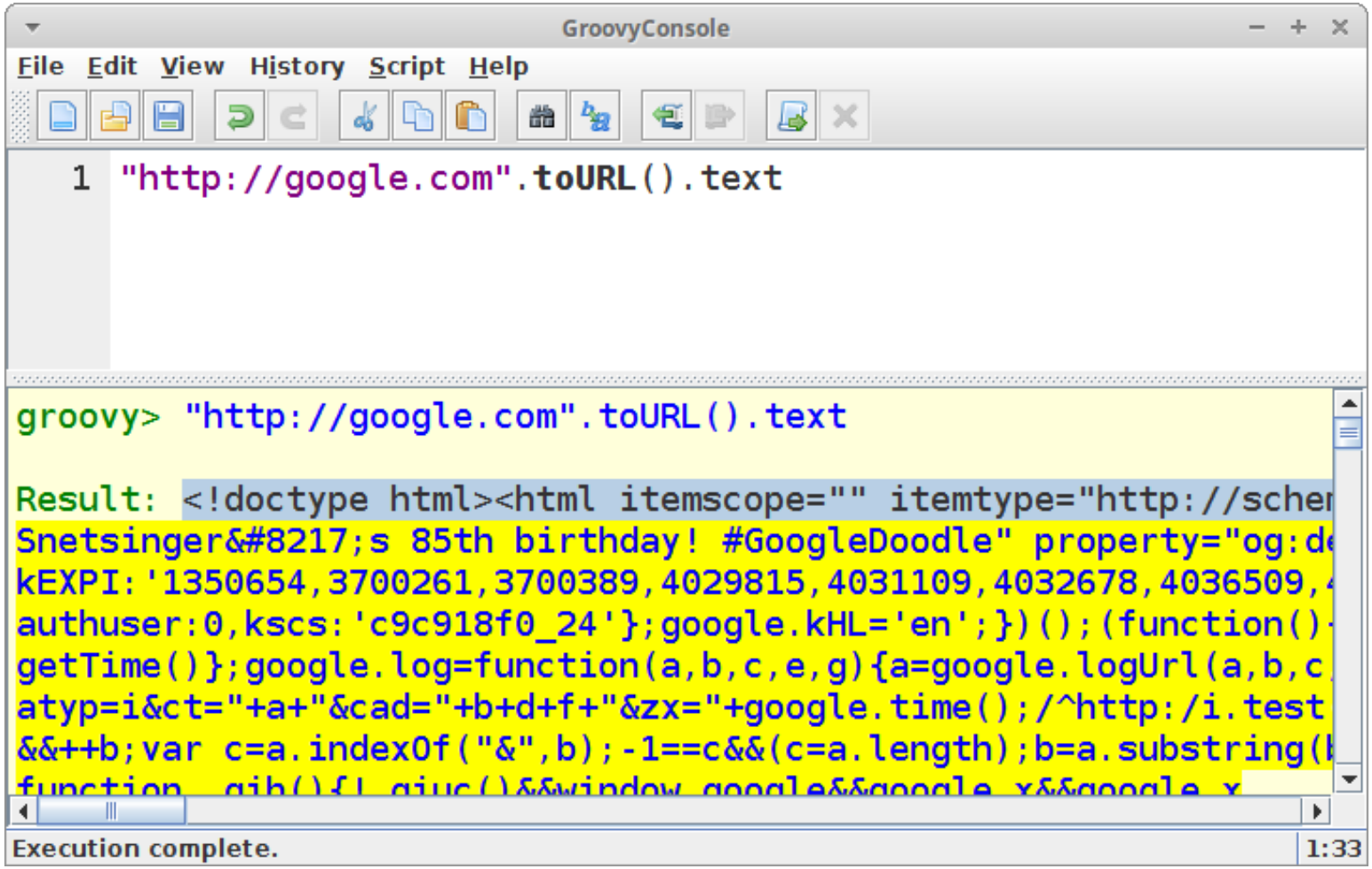
Getting Started

- Install SDKMAN
- Simplifies Groovy install
- Written in Bash

```
$ curl -s "https://get.sdkman.io" | bash
```

Groovy

Getting Started



Geb to Grid

Overview

1st Half

- Problem
- Groovy
- **Spock**
- Selenium
- Geb

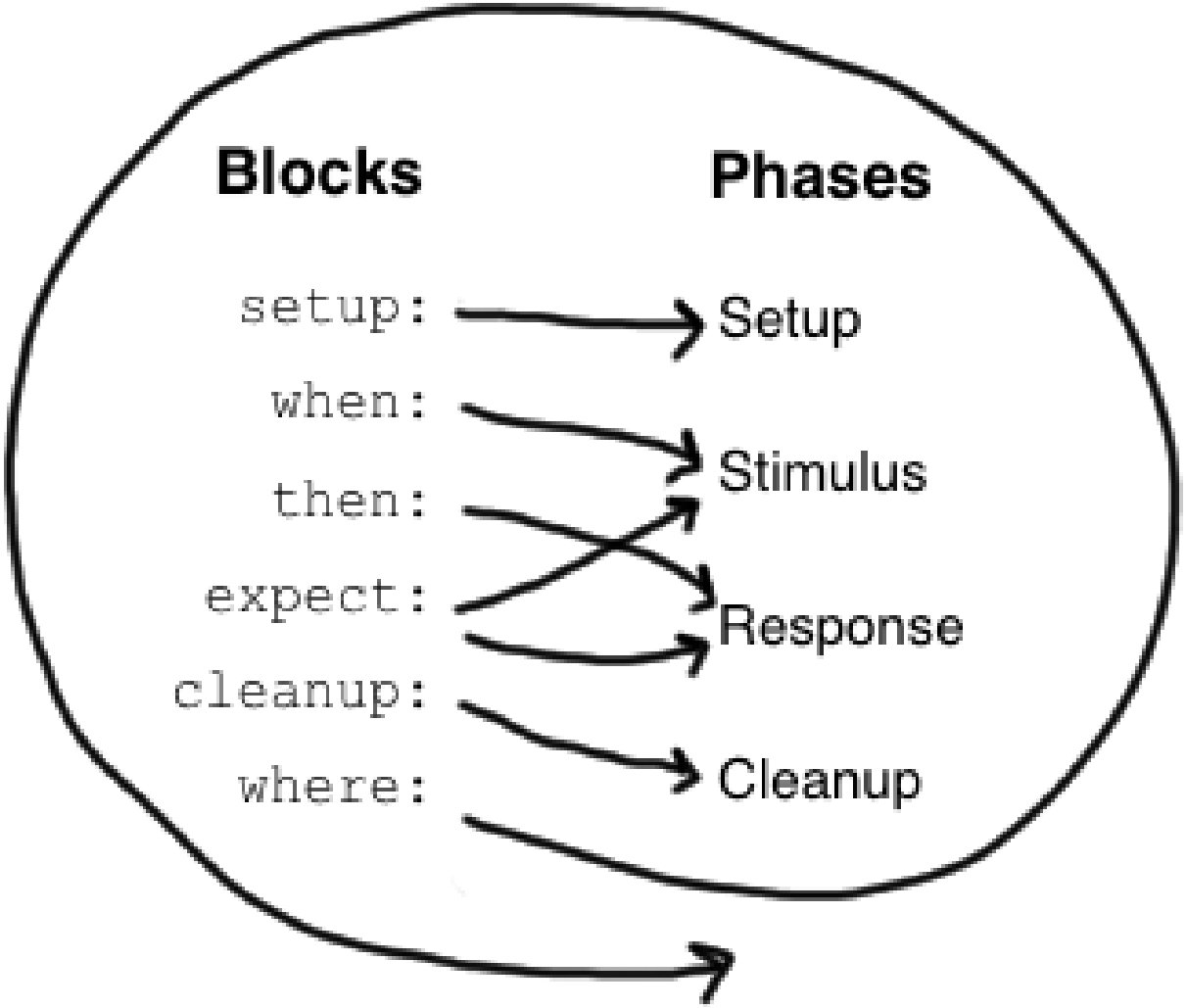
2nd Half

- Docker
- Docker Compose
- Selenium Grid
- Putting it together
- Q&A

Spock

- A more logical way to test
- Has Junit runner
- Highly expressive specification language

Spock


Blocks
Phases
setup:
when:
then:
expect:
cleanup:
where:
Setup
Stimulus
Response
Cleanup

Spock

Specification

- Groovy Class
- Extends from spock.lang.Specification
- Name should relate to system or system operation
- Instructs Junit to run specification using Sputnik

Spock

Specification

```
class MyFirstSpecification
extends Specification {
  // fields
  // fixture methods
  // feature methods
  // helper methods
}
```

Spock

Specification Components

- Fields
- Fixture Methods
- Feature Methods
- Helper Methods

- Store objects
- Belong to fixture
- Initialize at declaration
- Not shared between feature methods
- Every feature method gets its own object
- @Shared

Spock

Specification

- Fields
- Fixture Methods
- Feature Methods
- Helper Methods

Spock

Fixture Methods

```
def setup() {}          // run before every feature method

def cleanup() {}        // run after every feature method

def setupSpec() {}      // run before the first feature method

def cleanupSpec() {}    // run after the last feature method
```

Spock

Specification

- Fields
- Fixture Methods
- Feature Methods
- Helper Methods

Spock

Feature Method

- Heart of a specification
- Describes the features (properties, aspects) that is expected to be in the System Under Specification
- Named with string literals

```
def "pushing an element on the stack"() {
  // blocks go here
}
```

Spock

Feature Method

Four Phases

1. Set up the feature's fixture (optional)
2. Provide a *stimulus* to the system under specification
3. Describe the *response* expected from the system
4. Clean up the feature's fixture (optional)

Spock

Specification

- Fields
- Fixture Methods
- Feature Methods
- Helper Methods

- Setup/Cleanup logic
- Complex conditions

Spock

Helper Methods

```
def "offered PC matches preferred configuration"() {
  when:
  def pc = shop.buyPc()

  then:
  matchesPreferredConfiguration(pc)
}

def matchesPreferredConfiguration(pc) {
  pc.vendor == "Sunny"
  && pc.clockRate >= 2333
  && pc.ram >= 4096
  && pc.os == "Linux"
}
```

Spock

Spock Web Console

https://meetspock.appspot.com

Spock Web Console

```
import spock.lang.*

// Hit 'Run Script' below
class MyFirstSpec extends Specification {
  def "let's try this!"() {
    expect:
    Math.max(1, 2) == 3
  }
}
```

Actions ➤ | Run Script | New Script | Publish Script | View Recent Scripts

Subscribe

Result | Output | Stack Trace

```
MyFirstSpec
 - let's try this!   FAILED

   Condition not satisfied:

   Math.max(1, 2) == 3
        |         |
        2         false

   at MyFirstSpec.let's try this!(Script1.groovy:7)
```

Spock

Spock Web Console

Demo

Geb to Grid

Overview

1st Half

- Problem
- Groovy
- Spock
- **Selenium**
- Geb

2nd Half

- Docker
- Docker Compose
- Selenium Grid
- Putting it together
- Q&A

Selenium

Selenium automates browsers.

Selenium

Geb to Grid

Overview

1st Half

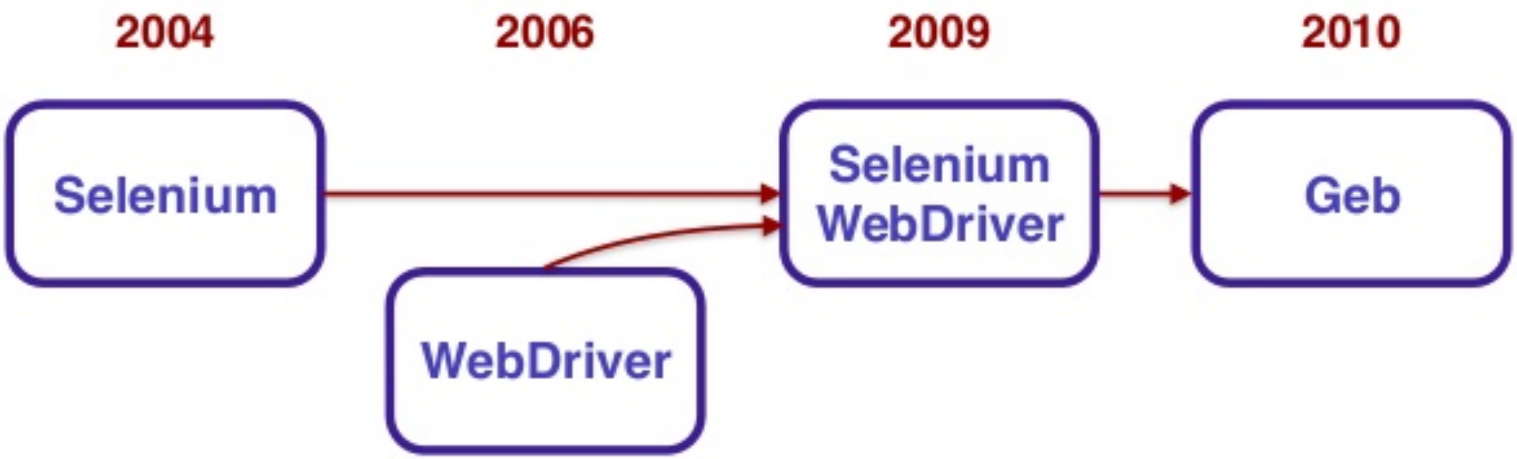
- Problem
- Groovy
- Spock
- Selenium
- **Geb**

2nd Half

- Docker
- Docker Compose
- Selenium Grid
- Putting it together
- Q&A

Geb

History of Geb



Unless otherwise indicated, these slides are © 2013-2015 Pivotal Software, Inc. and licensed under a
Creative Commons Attribution-NonCommercial license: http://creativecommons.org/licenses/by-nc/3.0/

Geb uses WebDriver

- A Domain Specific Language that simplifies Selenium coding
- Leverages WebDriver interface
- WebDriver is a remote control interface
- Enables user agent introspection & control

W3C Working Draft of WebDriver:

https://www.w3.org/TR/webdriver/

Geb

Demo

- Spock + Geb
- Login

Geb

```
import geb.Page

class LoginPage extends Page {
    static url = "http://myapp.com/login"
    static at = { heading.text() == "Please Login" }
    static content = {
        heading { $("h1") }
        loginForm { $("#login-form") }
        loginButton(to: AdminPage) { loginForm.login() }
    }
}
```

Geb + Spock

```
import geb.spock.GebReportingSpec
import spock.lang.*
import pages.*

Class LoginSpec extends GebReportingSpec {

    def "Admin logs in with valid credentials"() {
        when:
        to LoginPage

        then:
        at LoginPage

        when:
        loginForm.username = "IronMan"
        loginForm.password = "jarv1s"
        loginButton.click()

        then:
        at AdminPage
    }
}
```

Requirement

Use Case

- When user navigates to Login page
- Then Login page is displayed
- When user provides valid Admin credentials and clicks on login button
- Then user is taken to Admin page

Not Geb

```
package com.example.test.pageobjects;

import static com.example.test.Constants.loginUrl;  //URL definitions

import com.example.test.UserCredentials; //username and password

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

/* Using PageFactory to inject WebElements, defined by annotations. */
public class LoginPageObject {

    private final WebDriver driver;

    @FindBy(id = "d-name")
    private WebElement name;

    @FindBy(id = "d-pass")
    private WebElement pass;

    /* This one is not always there! */
    @FindBy(id = "d-error-message")
    private WebElement message;
```

Not Geb Continued ...

```
/* This one is not always there! */
@FindBy(id = "d-error-message")
private WebElement message;

public LoginPageObject(WebDriver driver) {
    this.driver = driver;
    PageFactory.initElements(driver, this);

    /* Make sure the Login Page has rendered as expected. LIKE THE GEB "at" assertions */
    if (!name.isDisplayed()) {
        throw new RuntimeException("LoginPageObject name field not displayed");
     }
     if (!pass.isDisplayed()) {
         throw new RuntimeException("LoginPageObject pass field not displayed");
     }
}

/**
 * Example:
 * <pre>
 *  WebDriver driver = // details elided
 *  LoginPageObject loginPage = LoginPageObject.browse(driver, "#optional");
 *  // do some work ...
 *  loginPage.logout();
 * </pre>
 */
public static LoginPage browse(WebDriver driver, String ... suffix) {
    final String url =
            (0 < suffix.length) ?
                    loginUrl + "/" + suffix[0] :
                    loginUrl;
    driver.get(url);
    return new LoginPage(driver);
}
```

Not Geb ...done.

```
    public AnotherPageObject login(UserCredentials user) {
        name.clear();
        name.sendKeys(user.getUsername());
        pass.clear();
        pass.sendKeys(user.getPassword() + "\n");
        return new AnotherPageObject(driver);  // LIKE THE GEB "to" NAVIGATION
    }

    public LoginPageObject loginWithoutPermissions(UserCredentials user) {
        name.clear();
        name.sendKeys(user.getUsername());
        pass.clear();
        pass.sendKeys(user.getPassword() + "\n");

        LoginPageObject result = new LoginPageObject(driver);
        if (!result.getMessage().contains("Access Denied")) {
            throw new RuntimeException("Access Denied message expected but not seen.");
        }
        return result;
    }

    public String getMessage() {
        return message.getText();
    }

    //.... More elided
}
```

Geb

With Geb, less code is more!

Break

Geb to Grid

Overview

1st Half

- Problem
- Groovy
- Spock
- Selenium
- Geb

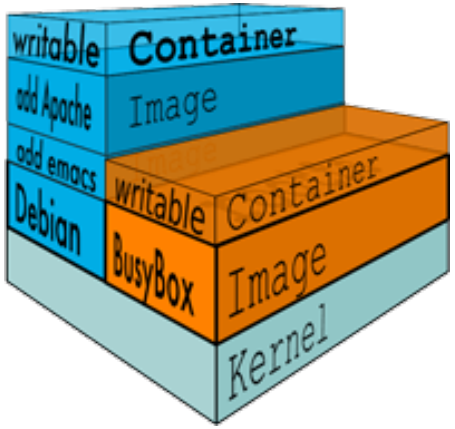
2nd Half

- **Docker**
- Docker Compose
- Selenium Grid
- Putting it together
- Q&A

Docker

What is Docker?

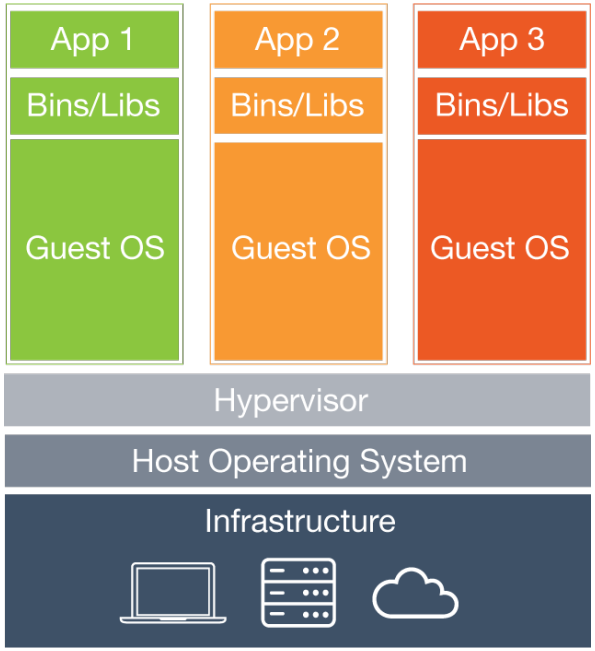
"*Docker allows you to package an application with all of its dependencies into a standardized unit for software development.*"



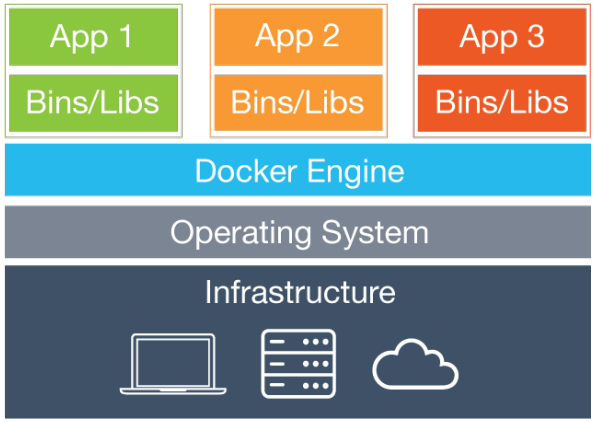
Docker

How is this different from a VM?

VM abstracts at the Hardware Layer



Container Abstracts at the Kernel Layer



Docker

- Available for Mac and Windows running in a VirtualBox VM.
- March 2016, Beta program announced for Mac and Windows
- Soon no VM to deal with on Mac and Windows

Docker
Dockerfile

Models the provisioning of the docker image

```
FROM ubuntu

ENV VERSION 2.53.0

RUN apt-get update -qqy \
    && apt-get -qqy --no-install-recommends install \
    software-properties-common \
    && rm -rf /var/lib/apt/lists/*
RUN add-apt-repository -y ppa:webupd8team/java

RUN echo debconf shared/accepted-oracle-license-v1-1 select true | debconf-set-selections
RUN echo debconf shared/accepted-oracle-license-v1-1 seen true | debconf-set-selections
```

Docker
Dockerfile (cont.)

```
RUN apt-get update -qqy \
    && apt-get -qqy --no-install-recommends install \
    bzip2 \
    ca-certificates \
    oracle-java7-installer \
    sudo \
    unzip \
    vim \
    && rm -rf /var/lib/apt/lists/*

RUN cd /var && wget http://selenium-release.storage.googleapis.com/${VERSION%.*}/selenium-server-standalone-${VERSION}.jar

#========================================
# Add normal user with passwordless sudo
#========================================
RUN sudo useradd seluser --shell /bin/bash --create-home \
    && sudo usermod -a -G sudo seluser \
    && echo 'ALL ALL = (ALL) NOPASSWD: ALL' >> /etc/sudoers \
    && echo 'seluser:secret' | chpasswd
```

Geb to Grid

Overview

1st Half

- Problem
- Groovy
- Spock
- Selenium
- Geb

2nd Half

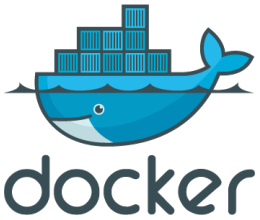
- Docker
- **Docker Compose**
- Selenium Grid
- Putting it together
- Q&A

Docker

Docker Compose

- Multi-container Docker Applications
- Configure with docker-compose.yml

Docker
Docker Compose



docker-compose.yml

```
version: '2'

services:
  hub:
    build: hub/.
    image: selenium_grid/hub:hub
    ports:
     - "4433:4433"
    logging:
      driver: "json-file"
    network_mode: "host"
    volumes:
      - ./hub/var/log:/var/log
```

```
  firefox:
    build: firefox/.
    image: selenium_grid/firefox:firefox
    links:
     - hub
    expose:
     - "5555"
    logging:
      driver: "json-file"
    volumes:
      - ./firefox/var/log:/var/log
```

Docker

Docker Compose

1) Define your application

(Dockerfile)

2) Define your services

(docker-compose.yml)

3) Start your entire application

(docker-compose up)

Docker

Docker Compose

1) Define your application

(Dockerfile)

2) Define your services

(docker-compose.yml)

3) Start your entire application

(docker-compose up)

Geb to Grid

Overview

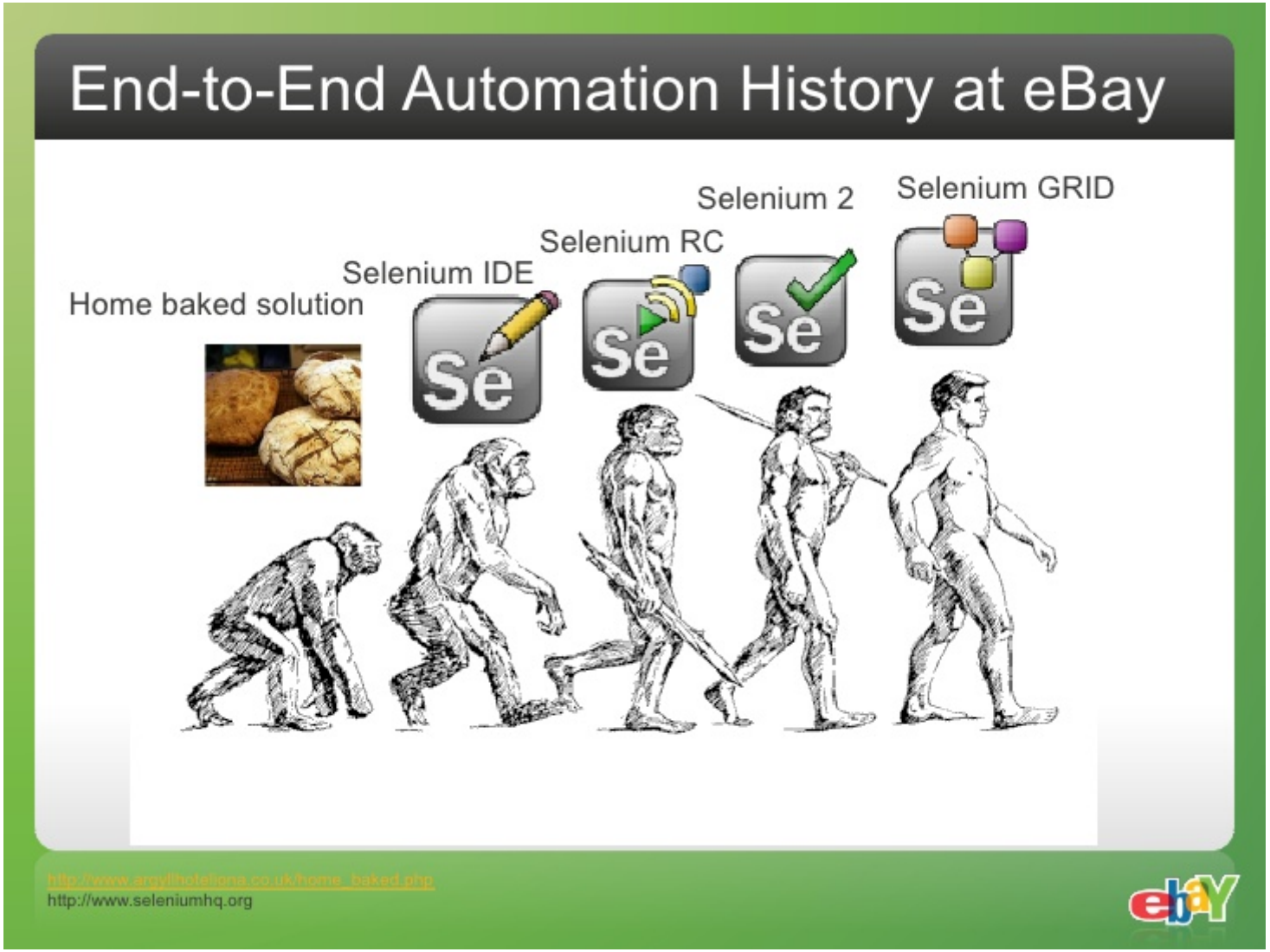
1st Half

- Problem
- Groovy
- Spock
- Selenium
- Geb

2nd Half

- Docker
- Docker Compose
- **Selenium Grid**
- Putting it together
- Q&A

Selenium Grid

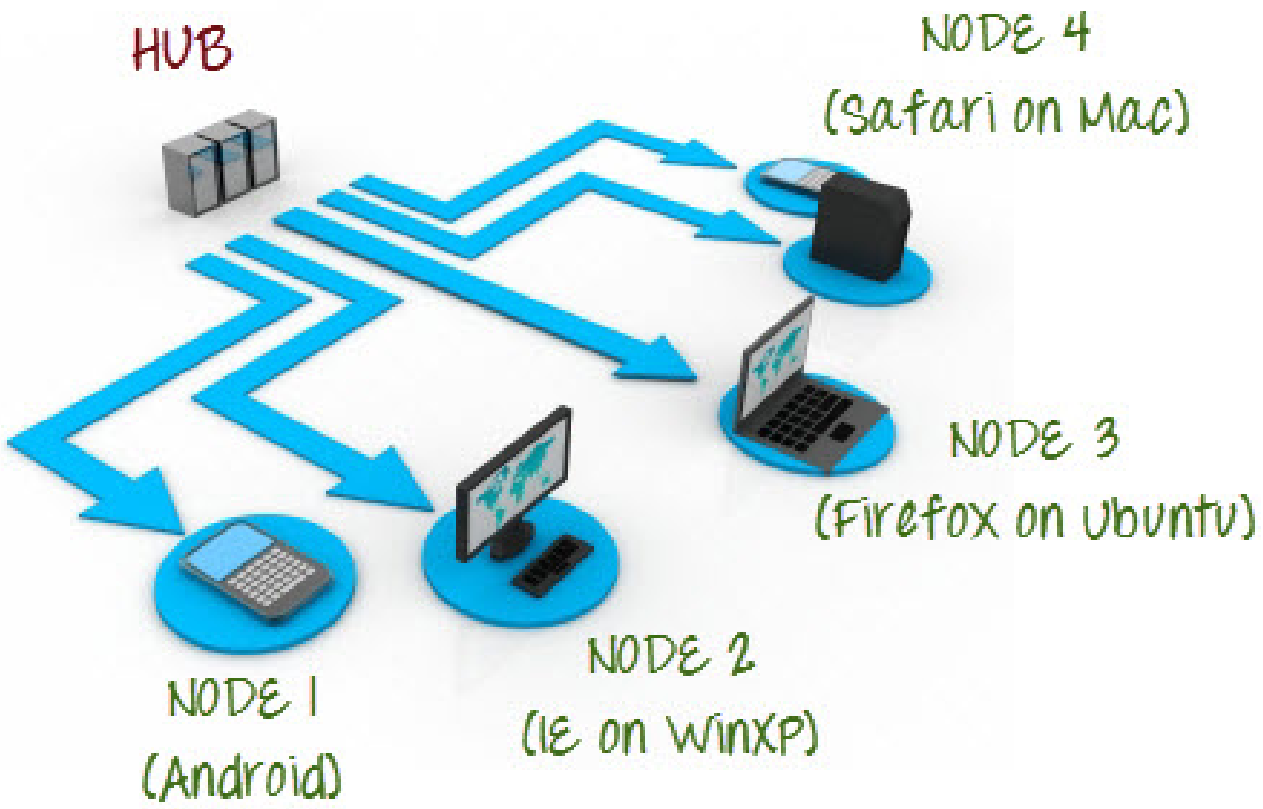


Selenium Grid

What is a Selenium Grid?

A Grid is basically a Hub connected to any number of nodes.

Selenium Grid Topology



Selenium Grid Simplifies Testing

- Client always connects to known endpoint
- Hub takes care of browser allocation
- Nodes can be added dynamically
- Hub handles bad nodes

Selenium Grid Resiliency

- Hub can send heartbeat requests to node to tell if it has gone down
- Hub can gracefully kill idle sessions
- Hub can kill sessions that take to long to create

Geb to Grid

Overview

1st Half

- Problem
- Groovy
- Spock
- Selenium
- Geb

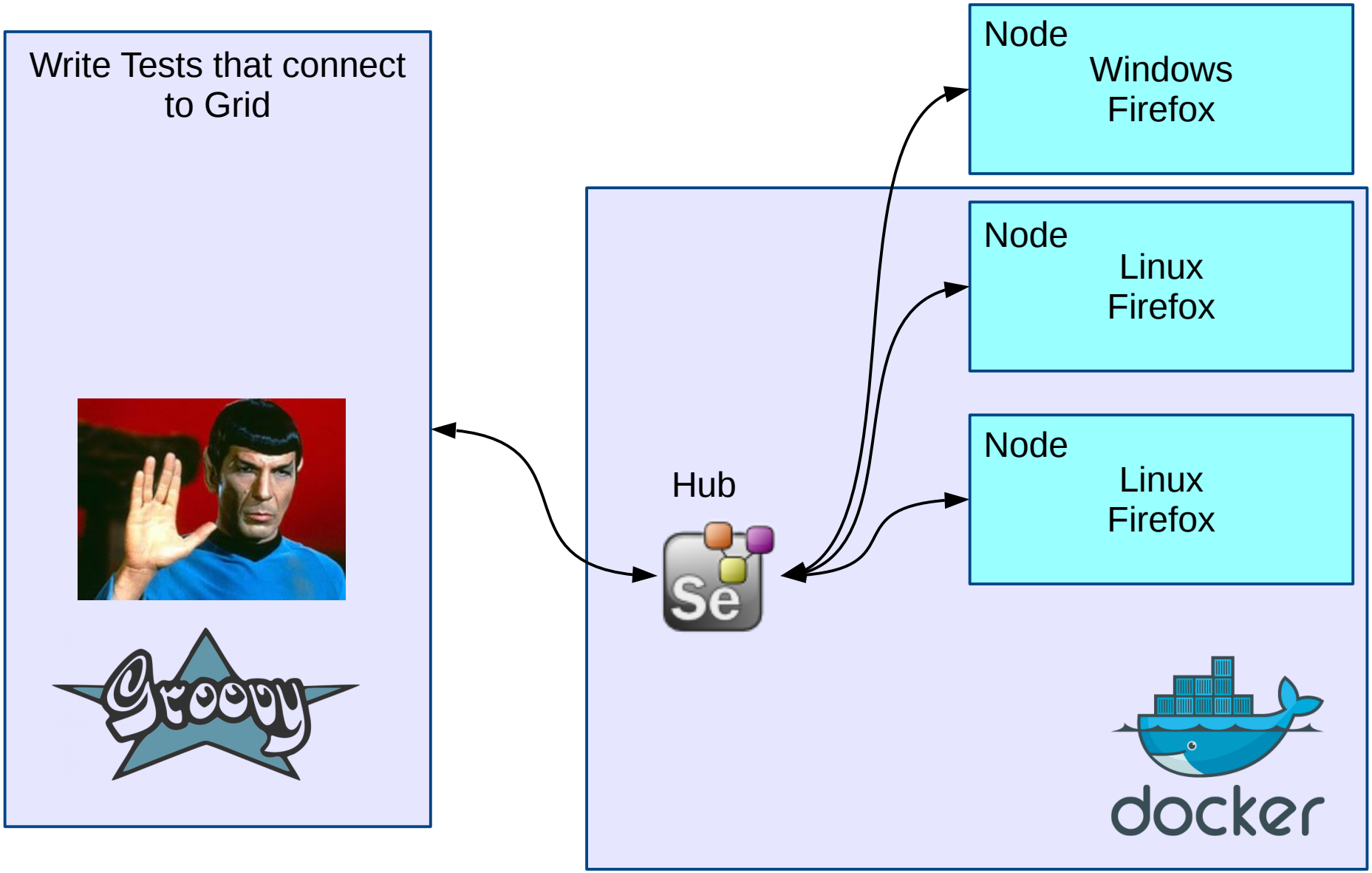
2nd Half

- Docker
- Docker Compose
- Selenium Grid
- **Putting it together**
- Q&A

Geb to Grid

Putting it all together

Geb to Grid


Write Tests that connect to Grid
Node
Windows
Firefox
Node
Linux
Firefox
Node
Linux
Firefox
Hub
Se
Groovy
docker

Geb to Grid

Demo

Geb to Grid Resources

- SdkMan

  http://sdkman.io

- Groovy

  http://groovy-lang.org

  closures: http://groovy-lang.org/closures.html

- Spock

  http://spockframework.github.io/spock/docs

  Spock Web Console: https://meetspock.appspot.com

- Geb

  http://gebish.org

Geb to Grid Resources

- Docker

  http://www.docker.com

- Docker Compose

  https://docs.docker.com/compose

- Selenium Grid

  http://www.seleniumhq.org/docs/07_selenium_grid.jsp

- Selenium

  http://www.seleniumhq.org

  W3C Webdriver: https://www.w3.org/TR/webdriver/

Q&A

Geb to Grid

Ralph A. Navarro Jr.
PH: 508-287-0190
GitHub: https://github.com/eaglet3d/GebToGrid