it KONTRAKT

Twój manager na rynku IT

AUTOMATYZACJA APLIKACJI WEBOWYCH

Who Am I?

- Quality Assurance Engineering Leader @ FastWhiteCat
- Selenium & Jenkins fanboy
- Trainer? Coach? Mentor?
- Apple believer ^.^

Show of hands

Tell me something about you ;-)

Purpose of this training?

Agenda

- Selenium architecture
- Locators + examples
- Selenium Webdriver API + examples
- Page Object Pattern + examples
- Example framework
- Workshops

Why do we need to code?

- manual testers will be marginalized
- it’s better paid
- it’s super fun ;-)

Advantages of Test Automation

- Frequent and quick regression testing
- Rapid feedback to developers
- Virtually unlimited iterations of test case execution
- Support for Agile and extreme development methodologies
- Increased test coverage
- Enables QA team to utilize bandwidth in exploratory testing

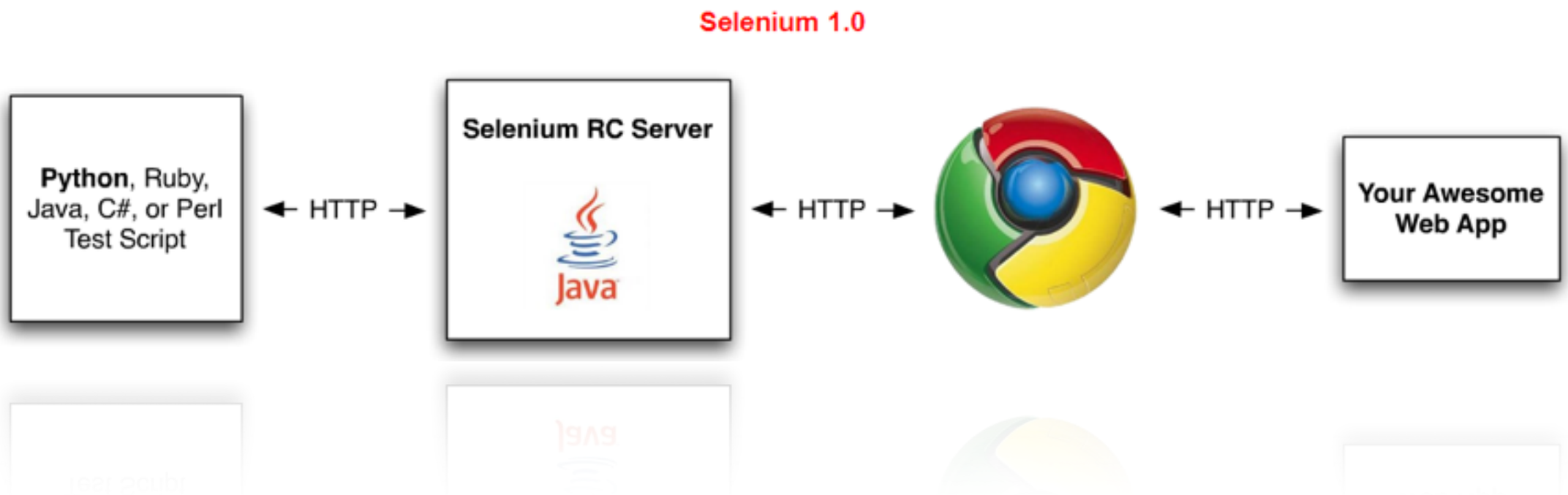
Selenium framework

web application test automation standard

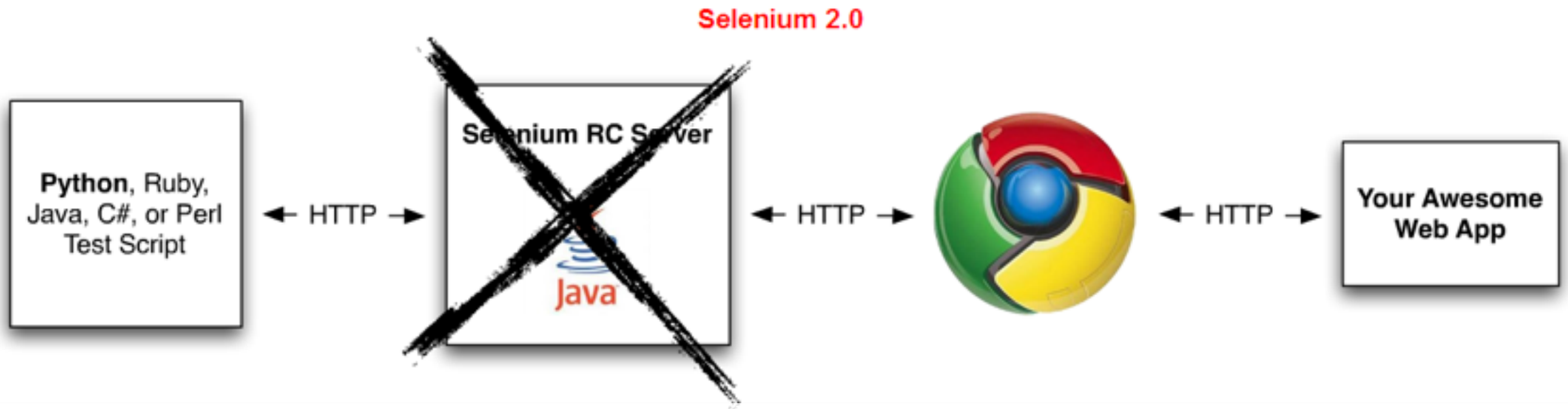
Selenium Components

- Selenium IDE
- Selenium RC
- Selenium Grid
- WebDriver (Selenium 2.0)

Selenium 1 vs Selenium 2



Selenium 1 vs Selenium 2



Selenium Compatibility









API basics

```
WebDriver driver = new ChromeDriver();

WebDriver driver = new FirefoxDriver();

WebDriver driver = new InternetExplorerDriver();

WebDriver driver = new HtmlUnitDriver();
```

Locator strategies

- By.id(“myElementId”)
- By.name(“myElementName”)
- By.xpath(“//input[@id=’myElementId’]”)
- By.cssSelector(“h1[title]”)

Let’s do some examples

Finding elements

- driver.findElement(By.id(…));
- driver.findElements(By.id(…));

Finding elements

```
@FindBy(id = "logonName")

private WebElement logonNameField;
```

Page interactions

- webelement.click();
- webelement.sendKeys(…);
- webelement.submit();
- Action Class -> drag&drop

Waits

- Polling the DOM for n sec

```
driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
```

- Wait until condition occurs

```
WebDriverWait wait = new WebDriverWait(driver, 10);

WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id("someid")));
```

Let’s code something

Page Object Pattern

Why do we need it?

- promotes reuse and reduce duplication
- makes tests readable
- makes tests more robust / less brittle
- improves maintainability, particularly if the application is rapidly evolving

Page Object Pattern

- workshops will be coded in Java
- this technique is **framework-agnostic**
- **everything** we discuss is applicable to other strongly-typed languages (e.g. C#)
- **most things** are applicable to dynamically-typed languages (e.g. Ruby, Python)

What does it do?

- Exposes methods that reflect the things a user can **see** and **do** on that page, e.g.
  - addItemToCart(), getPrice(), publishEntry()
- Hides the details of telling the browser **how** to do those things

Let’s code something

Fully functional Selenium framework

Now it’s your turn ;-)

Thanks

Catch me @ LinkedIn