

# Page Object is not the only Test Pattern



# Plan



# Plan

1. No patterns.
2. Simple test patterns
3. Test Kits
4. Docker Kits
5. Discussion time

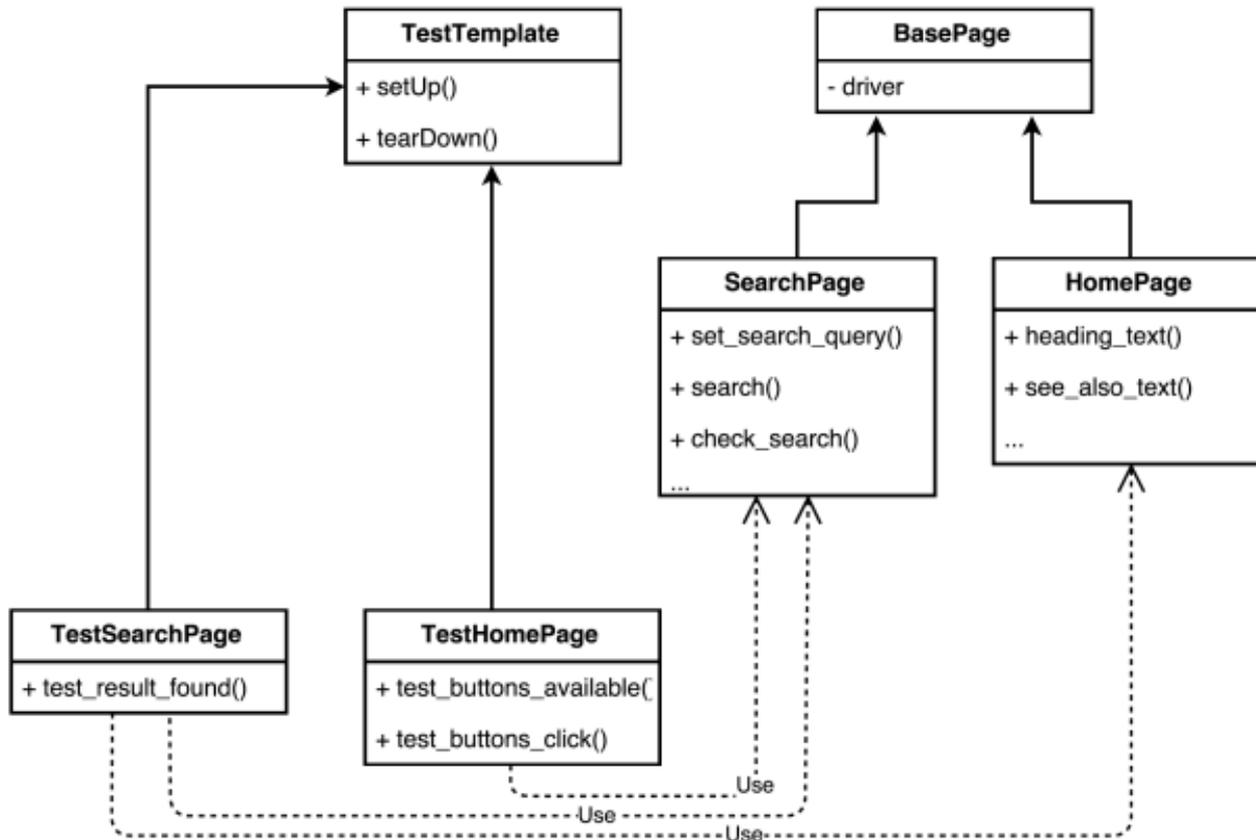
**Note:** common patterns like builder, singleton, multiton and others will not be covered.

No patterns



See Guru99 example

# Page Object (with example)

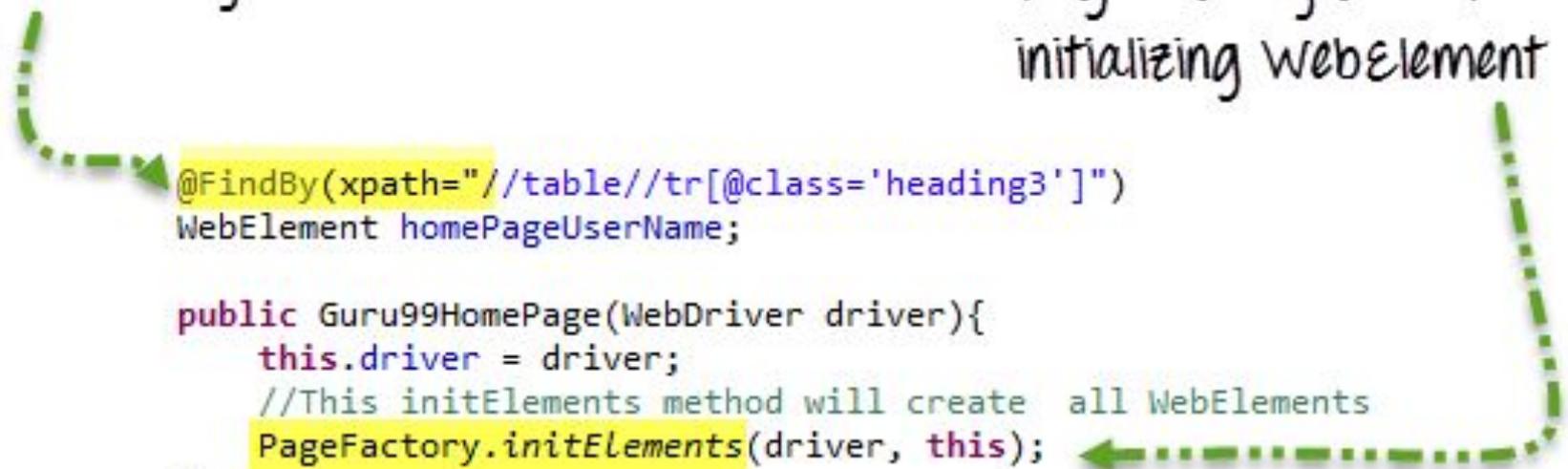


# Page Factory - Do we need it?

WebElements are identify by  
@FindBy Annotation

static initElements method of  
PageFactory class for  
initializing WebElement

```
@FindBy(xpath="//table//tr[@class='heading3']")  
WebElement homePageUserName;  
  
public Guru99HomePage(WebDriver driver){  
    this.driver = driver;  
    //This initElements method will create all WebElements  
    PageFactory.initElements(driver, this);  
}
```

A diagram consisting of two green dashed arrows. One arrow starts from the text 'WebElements are identify by @FindBy Annotation' and points to the '@FindBy' annotation in the code. The other arrow starts from the text 'static initElements method of PageFactory class for initializing WebElement' and points to the 'initElements' method call in the code.

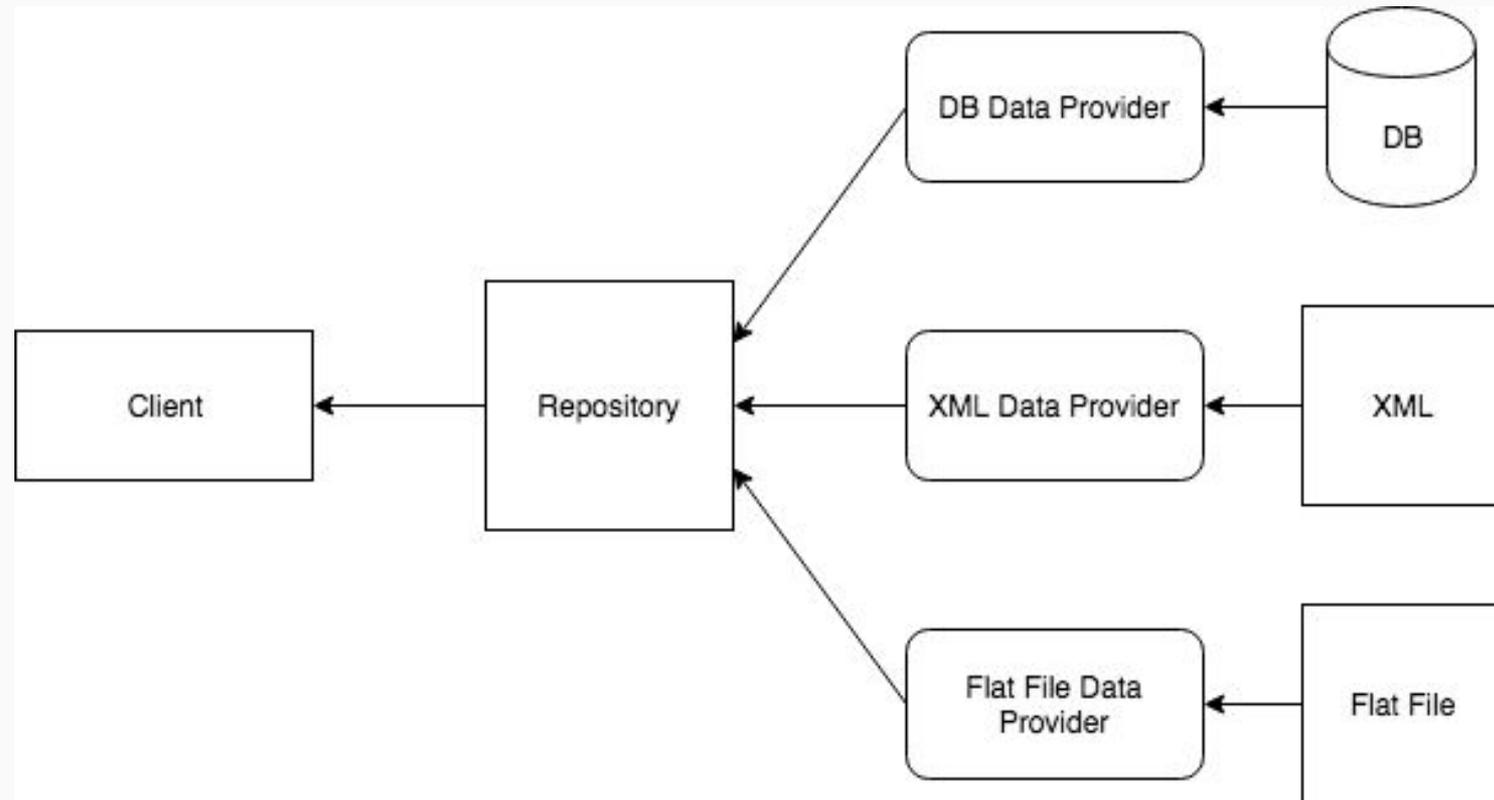
# Some (abstract) Factory (method)

- Create standard user
- Create and checkout cart
- etc.

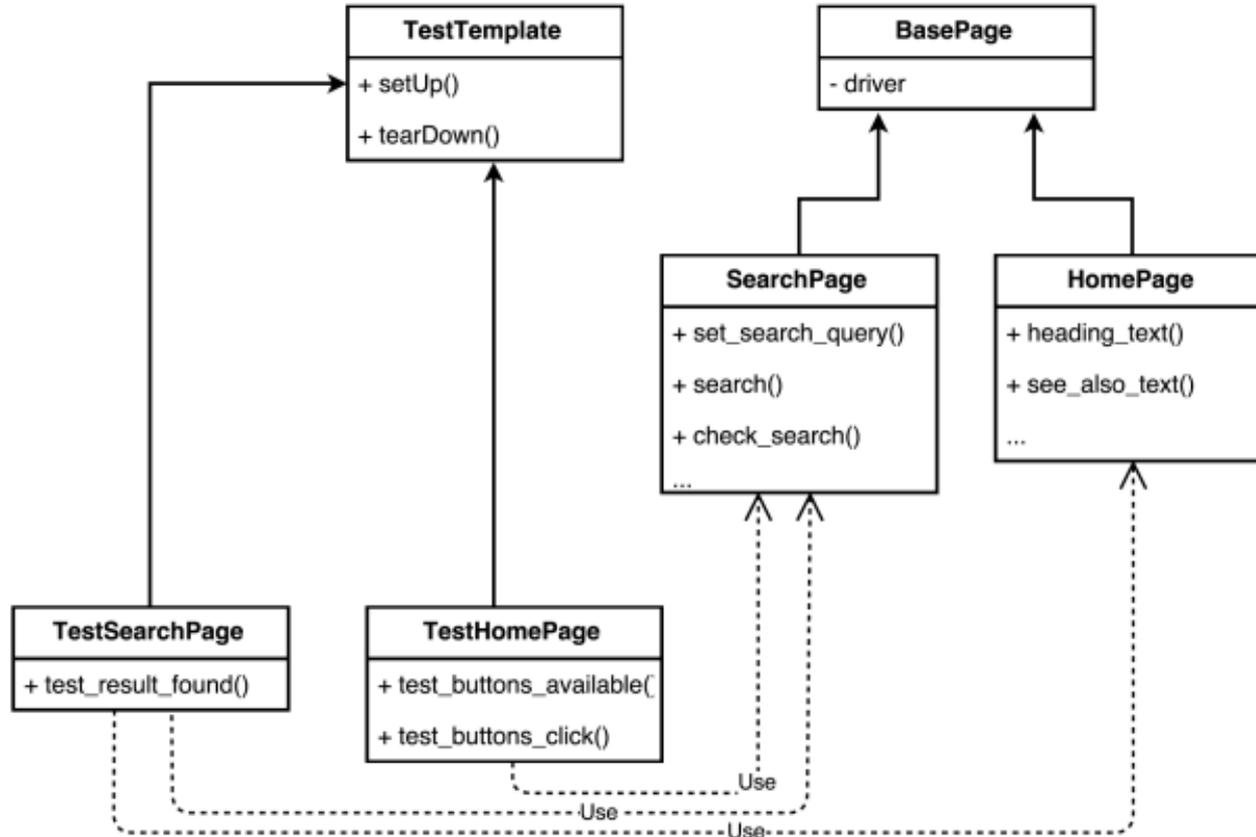
# Data Registry

- Generate objects by providing minimum to no parameters.
- Can use some Factory underhood.
- Can provide additional logic for storing test objects.
- See example.

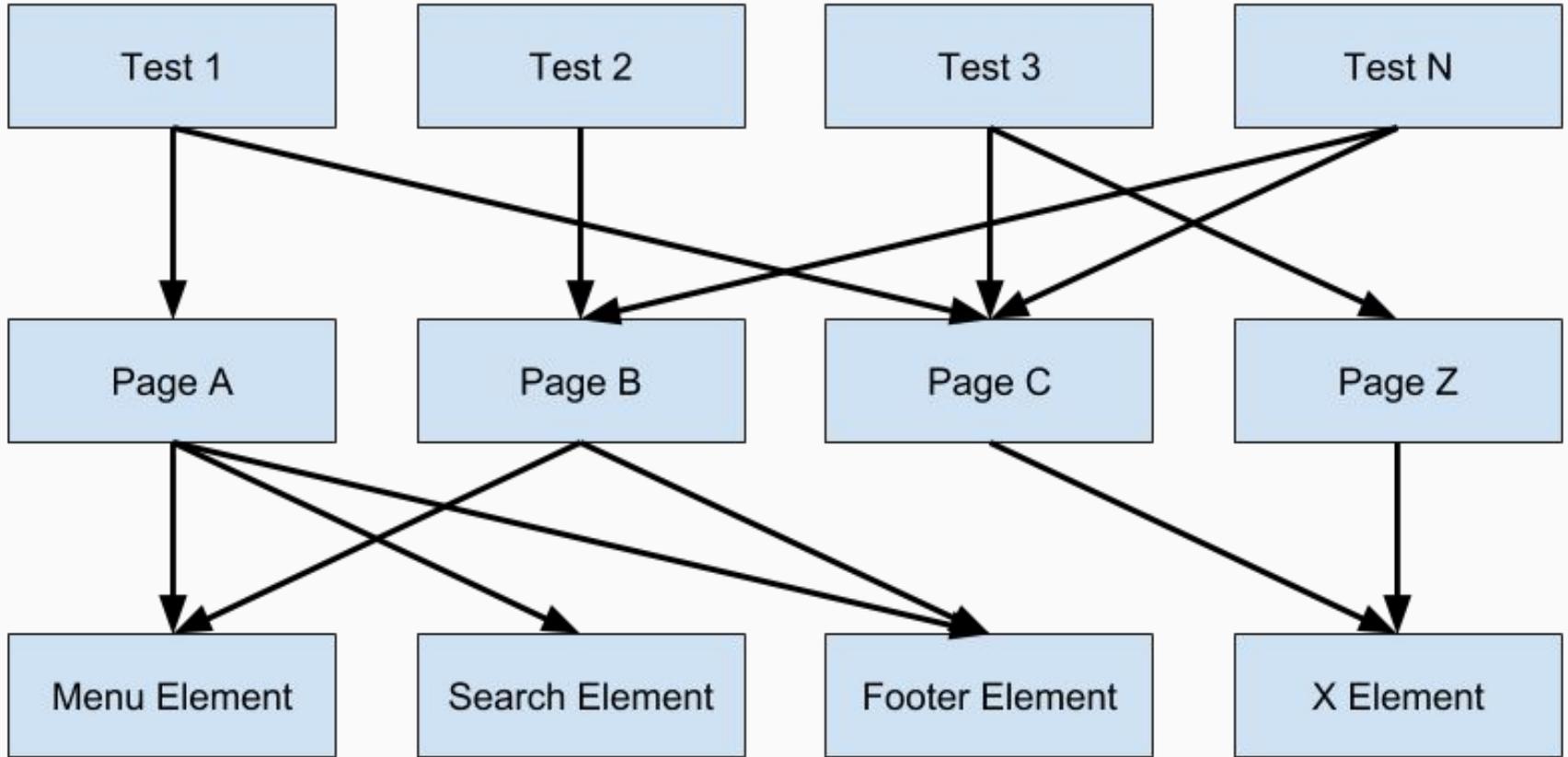
# Data Management



# Page Element/Composite (Problem)



# Page Element/Composite (Solution - with example)



# Steps

```
@Test
public void booksShouldBeSearchedByName() {
    loginToBookstore("u", "pass");
    openBooksSearchPage();
    enterSearchCriteria("name", "selenium");
    searchBooks();
    assertFalse(isEmptySearchResult());
    openBookDetailsByIndex(1);

    assertEquals("selenium 1.0 Testing Tools", getBookTitle());
    assertEquals("David Burns", getBookAuthor());
}
```



Steps:

1. Login to book store as user 'u'/'pass'
2. Open search books page
3. Try to search by word 'selenium'
4. Results should be non empty
5. Open first found book details

Expected results:

1. Book title is 'selenium 1.0 Testing Tools'
2. Author is 'David Burns'

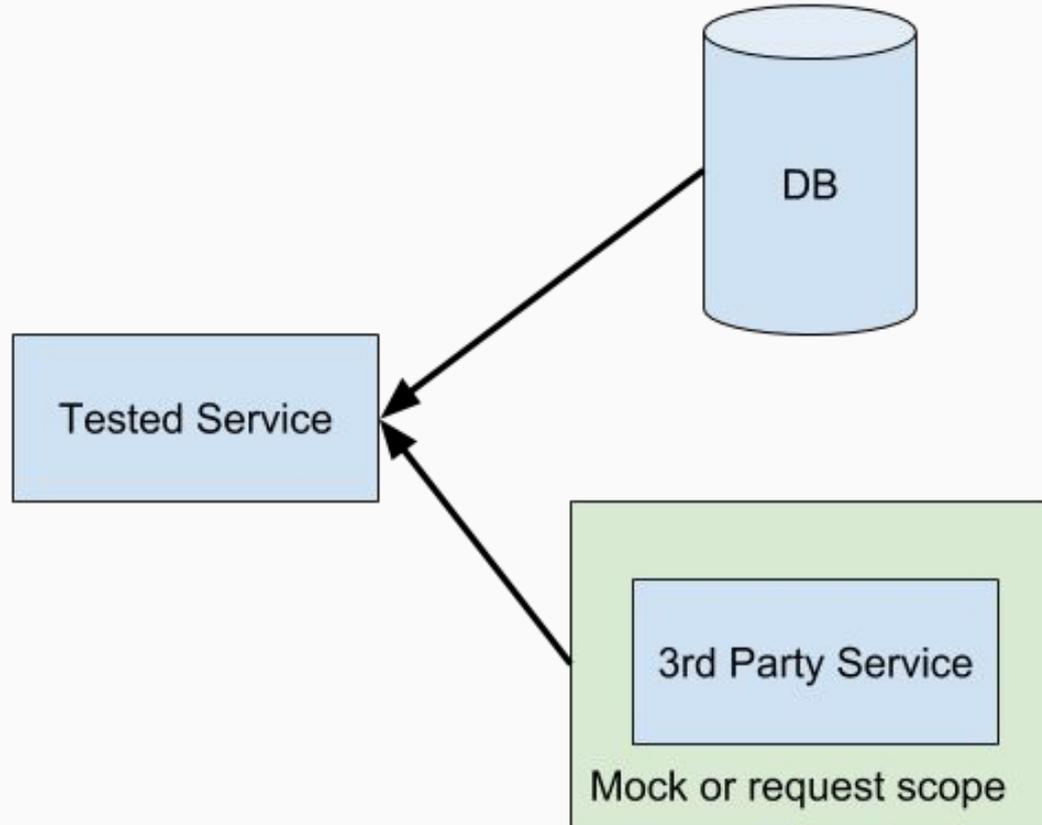
# Loadable Component (with example)

```
public interface LoadableComponent extends Page {  
    void load();  
    void isLoaded();  
}
```

# Chain Of Invocations (with example)

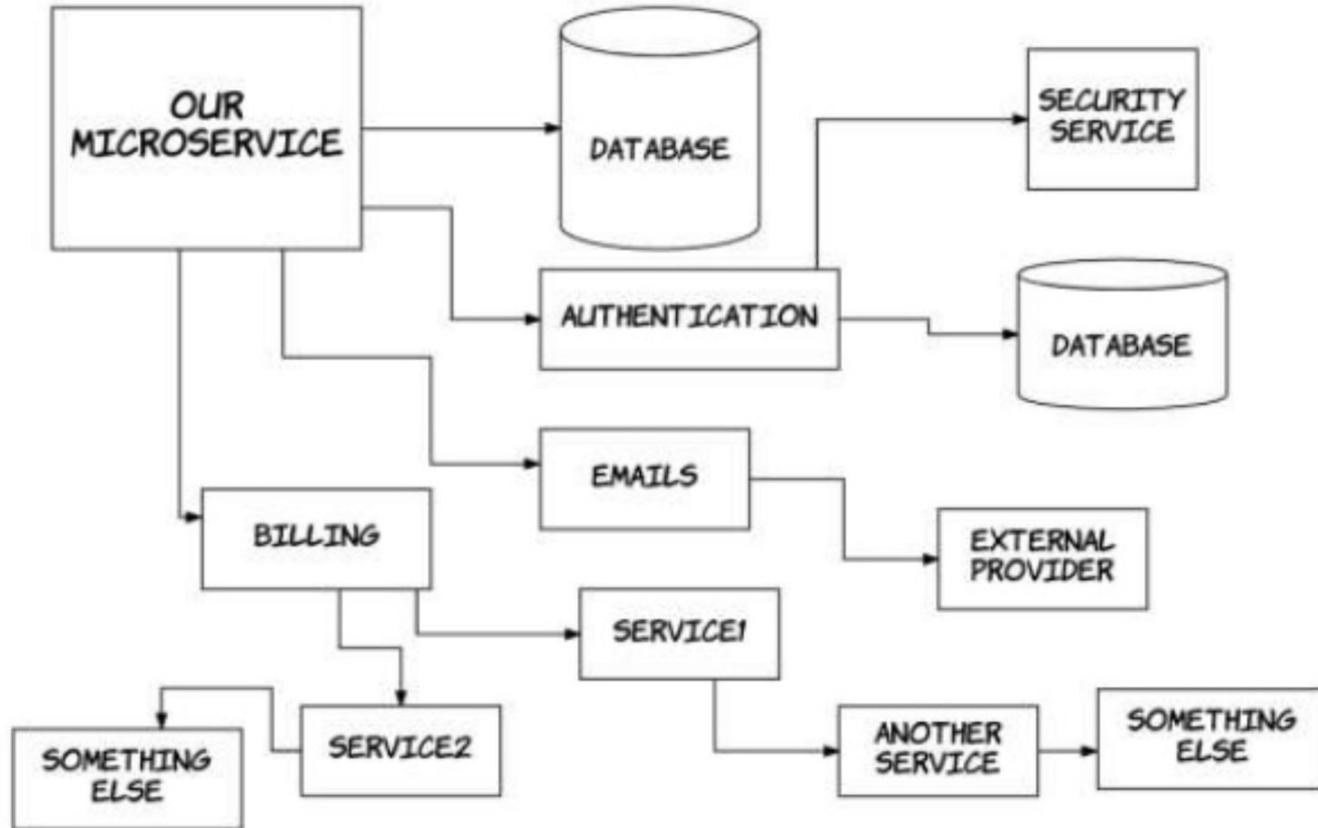
- Each method returns resulting page, when possible
- Kinda functional programming

# Testing average service on real environment

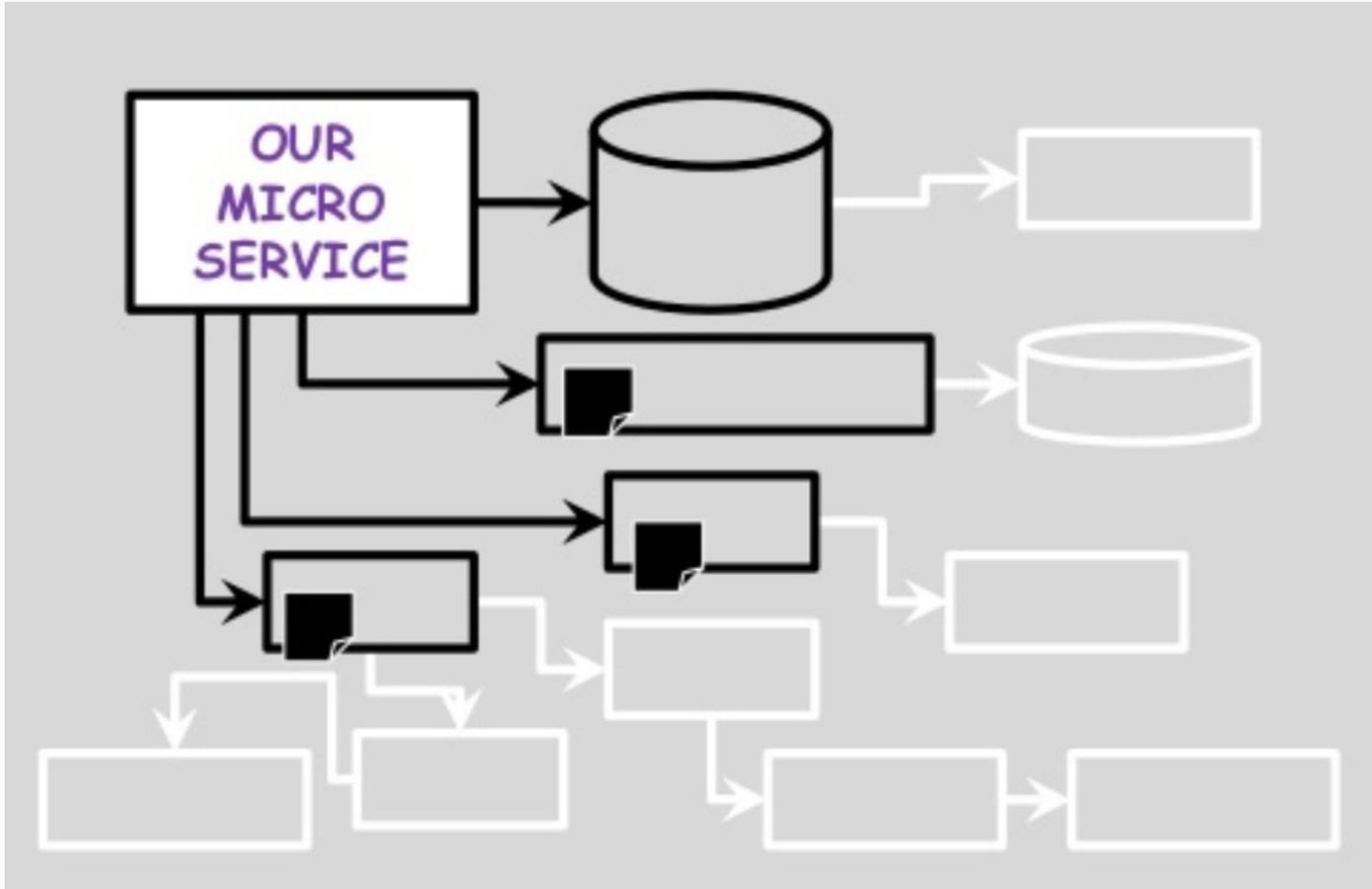




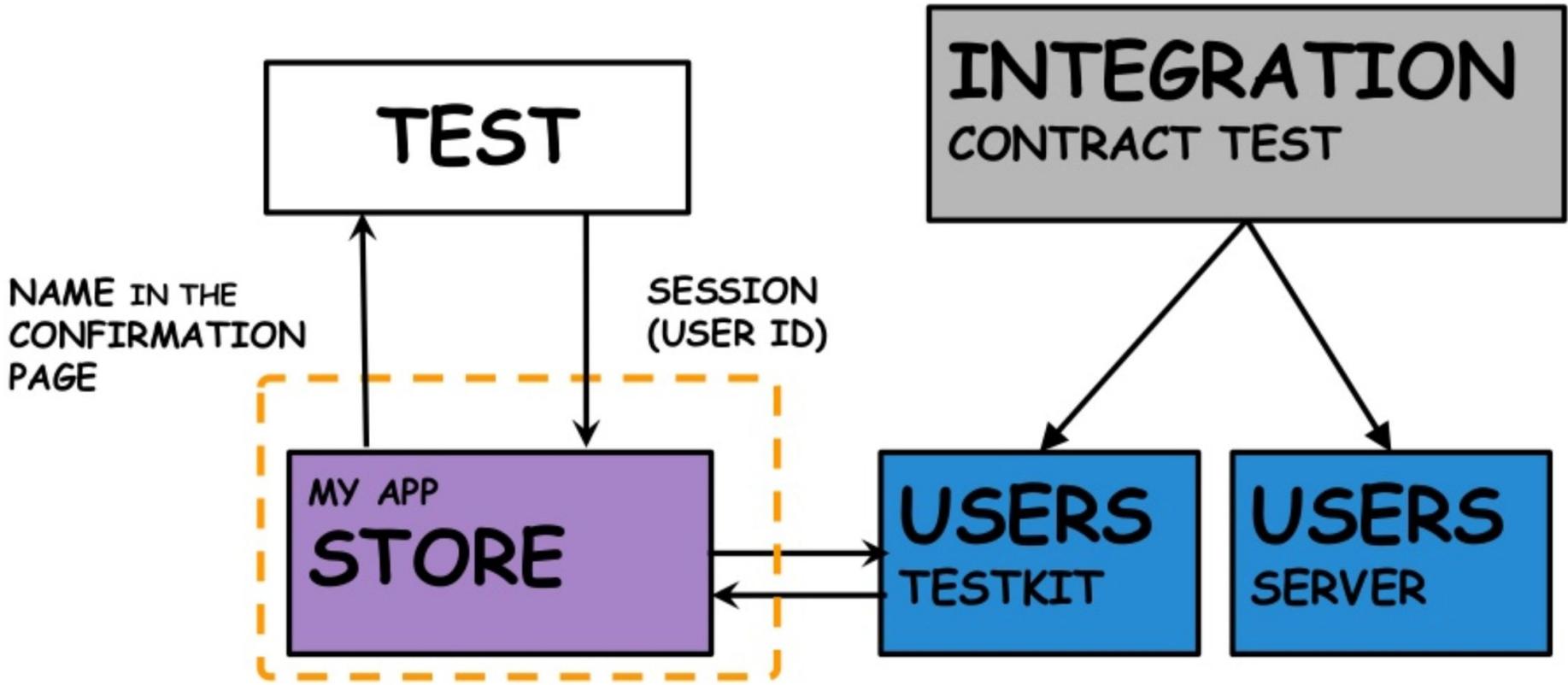
# Tested service with dependencies



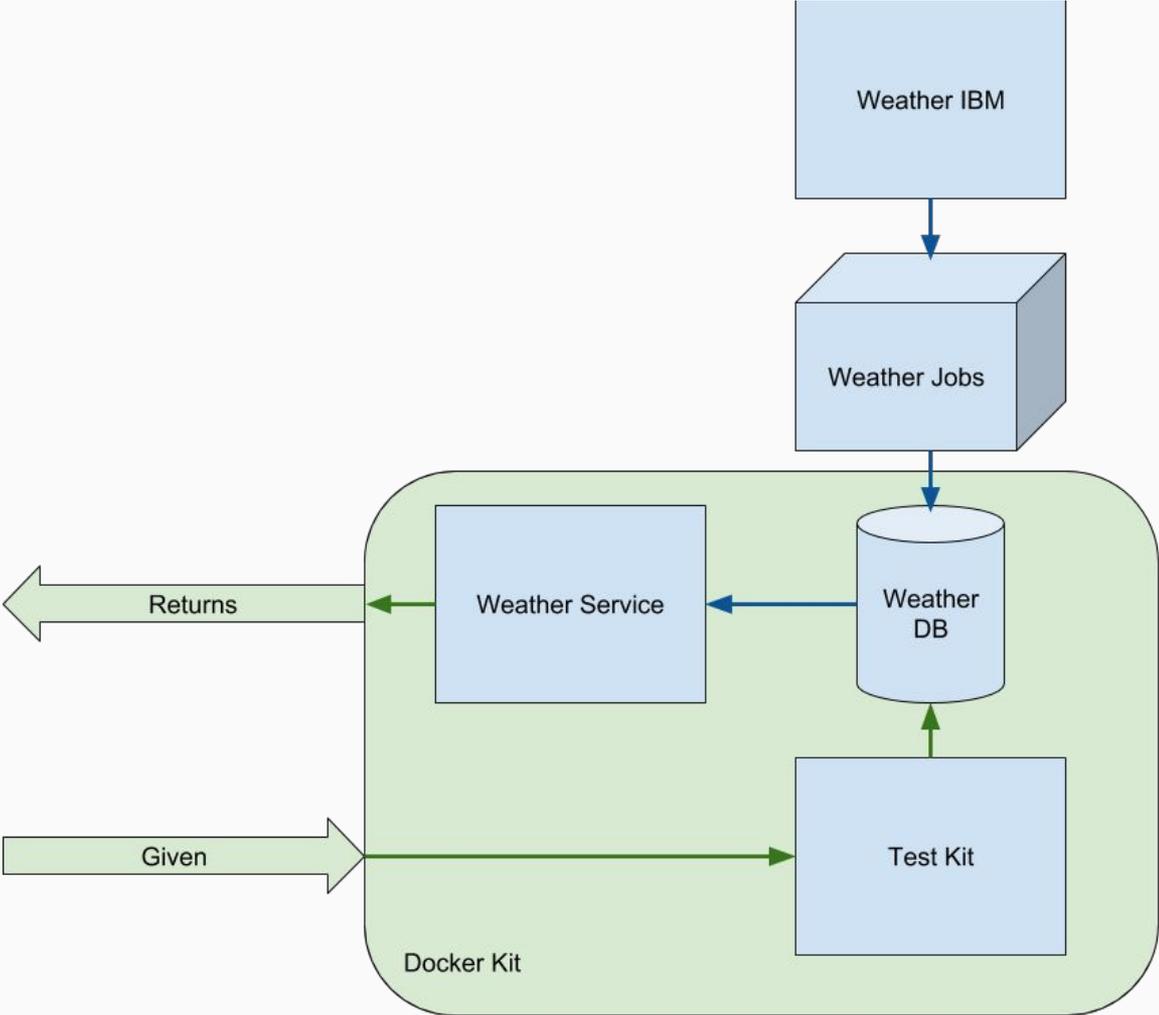
# Mock as much as possible



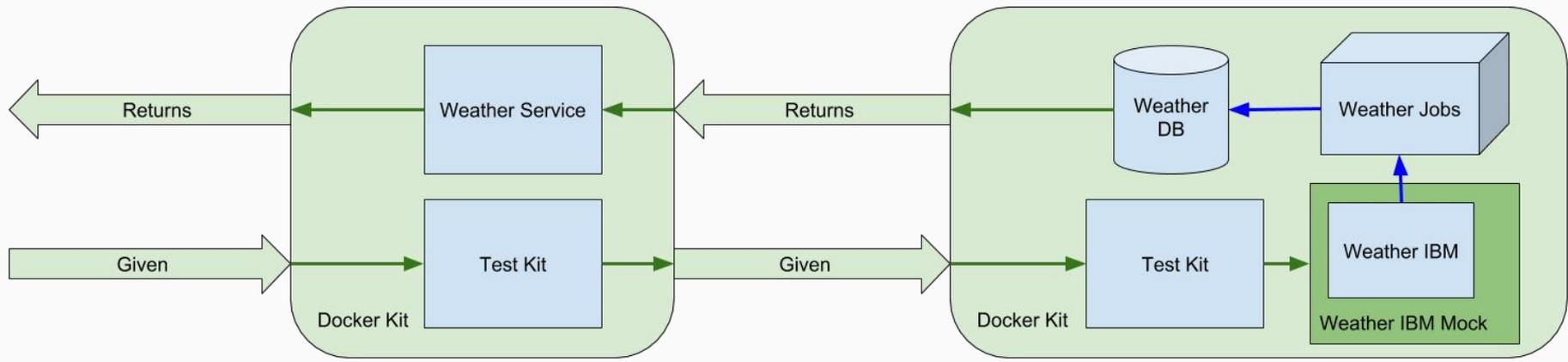
# Test Kits



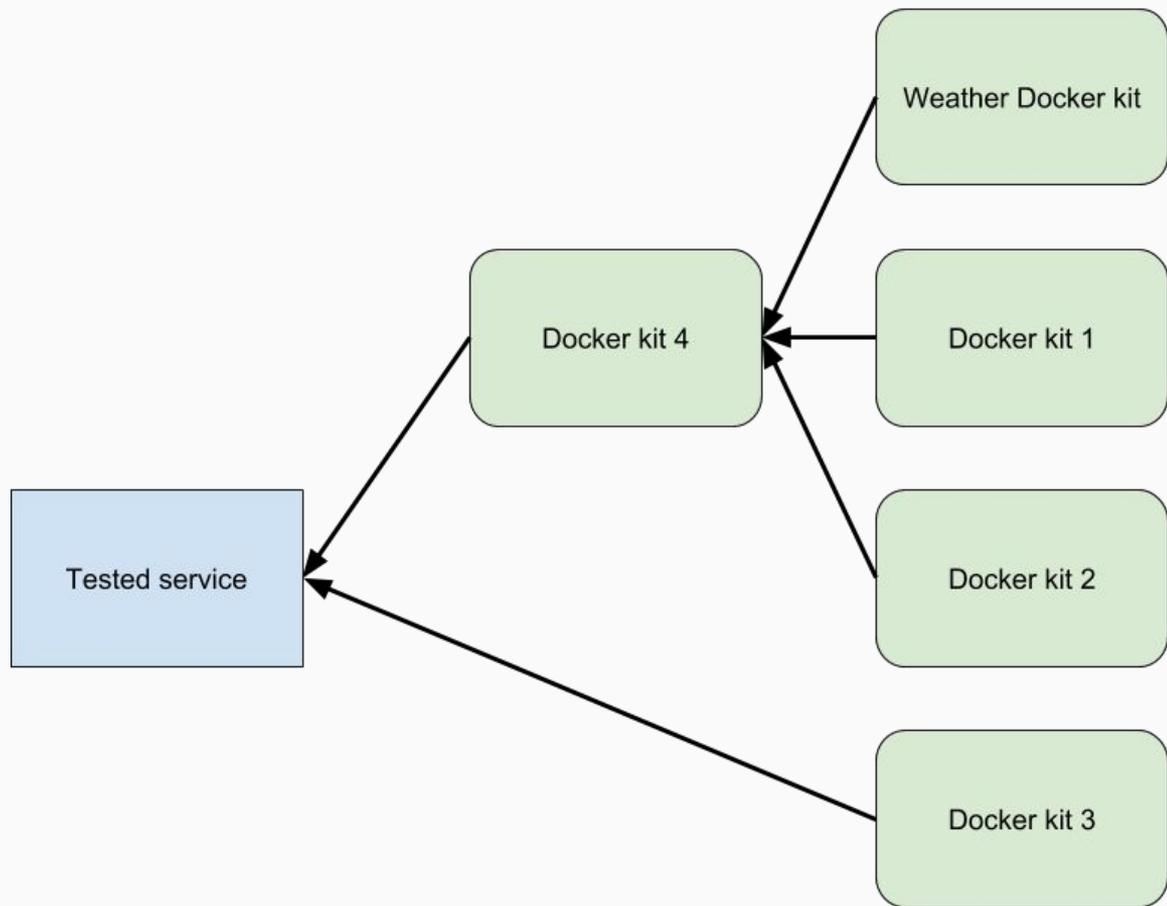
# Docker Kits



# Docker Kits System



# Docker Kits System



# DISCUSSION TIME



# References

- Examples - <https://github.com/alex-d-bondarev/demo-test-patterns>
- Design patterns in test automation - [https://www.youtube.com/watch?v=SEWa5\\_RW2BM](https://www.youtube.com/watch?v=SEWa5_RW2BM)
- No pattern example - <https://www.guru99.com/first-webdriver-script.html>
- Test kits slides - <https://www.slideshare.net/maximnovak/testing-micro-services-using-testkits>
- Test kits Devovx presentation - <https://www.youtube.com/watch?v=i3ancMhpldU>
- Multiple compose files - <https://docs.docker.com/compose/extends/#example-use-case>

*That's all Folks!*