Testing Matters

And so does your sleep.

Rahul C

Senior Developer

ThoughtWorks®

Rahul C

I write about tech, politics and food.

blog.bitsapien.dev

Senior Developer

ThoughtWorks®

Why am I talking about testing?

Files > 500 LOC

Why am I talking about testing?

Files > 500 LOC

Tests that mocked so much, they were just testing mocks

Why am I talking about testing?

Files > 500 LOC

Tests that mocked so much, they were just testing mocks

Why am I talking about testing?

Can’t make no correlation between a business feature and a test

What should you expect?

Ways to make your tests suite more readable

What should you expect?

Ways to make your tests suite more readable

Spend less time ripping your head when writing or reading tests

What should you expect?

Ways to make your tests suite more readable

Spend less time ripping your head when writing or reading tests

What should you expect?

Spend less time refactoring

Ways to make your tests suite more readable

Spend less time ripping your head when writing or reading tests

What should you expect?

Test with more confidence

Spend less time refactoring

Ways to make your tests suite more readable

Spend less time ripping your head when writing or reading tests

Sleep Well At Night

Test with more confidence

Spend less time refactoring

Software Engineers

Why do we do software development?

I ❤️ code.

But…

Business Impact

Always fast. Always confident.

All Hail Capitalism 💰

There is always a cost.

Optimise on that cost.

Code is the cost.

Code is the cost.

Writing code costs.

Code is the cost.

Writing code costs.

There is cost to waiting for code to be written - agility.

Code is the cost.

Writing code costs.

There is cost to waiting for code to be written - agility.

Changing code costs.

Code is the cost.

Writing code costs.

There is cost to waiting for code to be written - agility.

Changing code costs.

Understanding code costs.

We read code more than we write.

Clean Code

Reading Code.

Reading Code.

Purpose? Boundaries? Responsibilities?

Documentation


UP TO DATE DOCUMENTATION
NOW THERE IS SOMETHING I HAVEN'T SEEN IN A LONG TIME.
imgflip.com

on

Let’s care about our tests more.

Test code is also code.

Let’s look at these problems.

Large Test Suites

Large Test Suites

A Vue Frontend Application

Business Logic

Language	Files	Lines	Blanks	Comments	Code	Complexity
Vue	25	1987	78	0	1909	62
JavaScript	9	286	21	3	262	13
CSS	1	3	0	0	3	0
HTML	1	17	0	4	13	0
JSON	1	50	0	0	50	0
Sass	1	60	10	0	50	0
Total	38	2403	109	7	2287	75

Estimated Cost to Develop $64,390
Estimated Schedule Effort 4.850287 months
Estimated People Required 1.179433

Processed 64032 bytes, 0.064 megabytes (SI)

Tests

Language	Files	Lines	Blanks	Comments	Code	Complexity
JavaScript	28	2953	370	8	2575	36
JSON	2	79	0	0	79	0
Total	30	3032	370	8	2654	36

Estimated Cost to Develop $75,282
Estimated Schedule Effort 5.147031 months
Estimated People Required 1.299423

Processed 95415 bytes, 0.095 megabytes (SI)

A Java Backend application

Business Logic

Language	Files	Lines	Blanks	Comments	Code	Complexity
Java	93	2705	384	10	2311	69
YAML	2	41	2	0	39	0
JSON	1	39	0	0	39	0
XML	1	14	0	0	14	0
Total	97	2799	386	10	2403	69

Estimated Cost to Develop $67,824
Estimated Schedule Effort 4.946990 months
Estimated People Required 1.218040

Processed 90092 bytes, 0.090 megabytes (SI)

Tests

Language	Files	Lines	Blanks	Comments	Code	Complexity
JSON	40	2117	0	0	2117	0
Java	24	3406	499	27	2880	11
Total	64	5523	499	27	4997	11

Estimated Cost to Develop $146,298
Estimated Schedule Effort 6.625296 months
Estimated People Required 1.961782

Processed 196209 bytes, 0.196 megabytes (SI)

Refactoring 🤯

Refactoring 🤯

Refactoring is changing implementation details without breaking our tests / interfaces.

Refactoring 🤯

What if it breaks existing functionality !

Refactoring 🤯

Don’t touch that piece of code atleast

Refactoring 🤯

The fear is real.

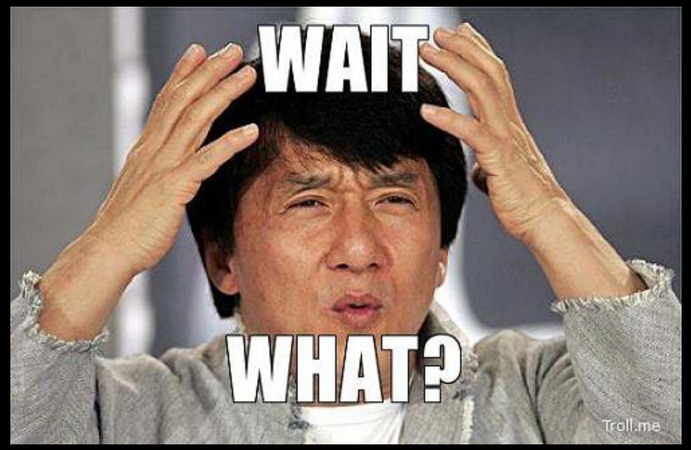
Refactoring 🤯

And it’s not because you don’t have tests.

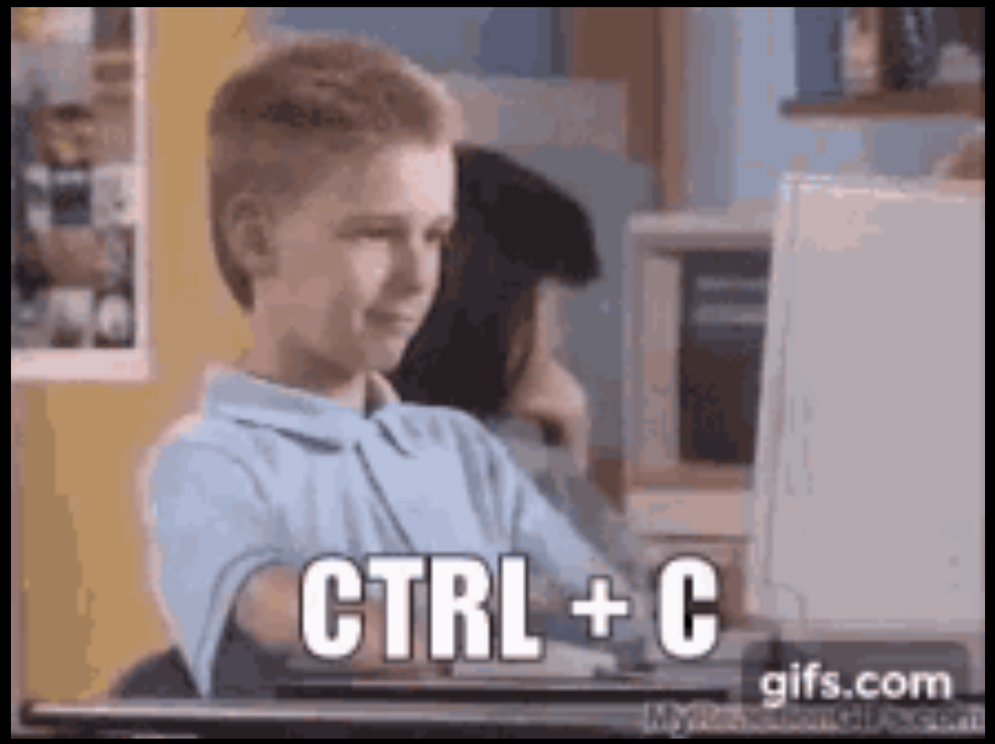
Refactoring 🤯

Why?

The Ghost Tests



Copy Pasta



Poor test suites will lead to poorer test suites.

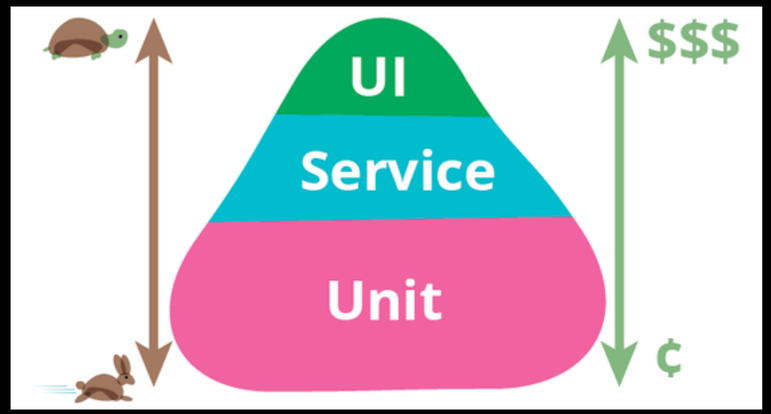
Duct Tape Programmer

Cucumber anyone ? 

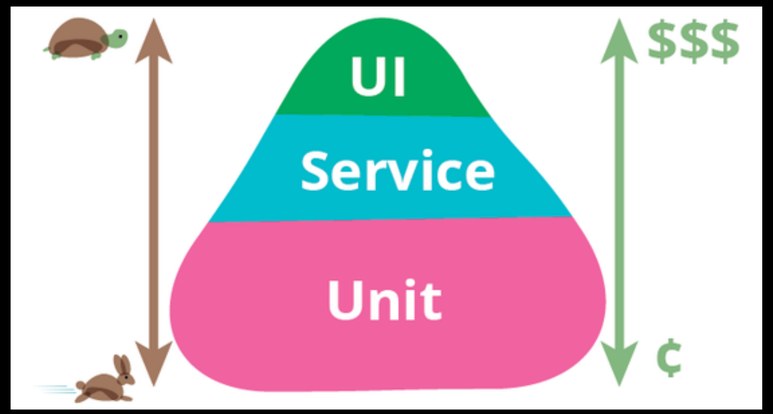
I ❤️ writing tests

Ideas

The Test Pyramid : What's missing?



The Test Pyramid : What's missing?



Confidence

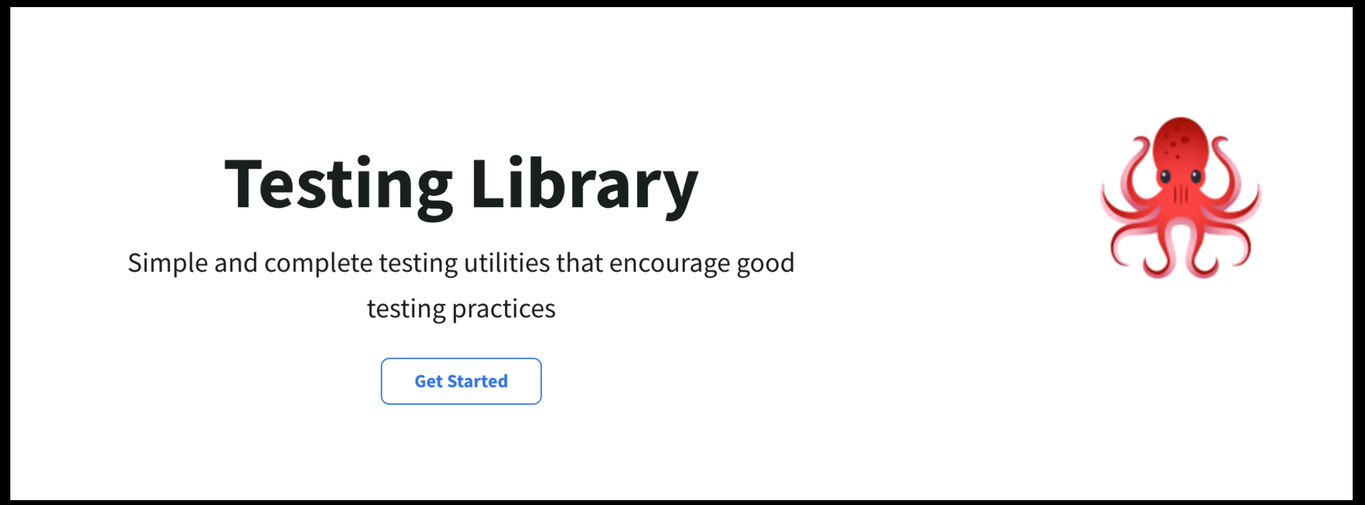
The contract with the world is important.

“When we write a test, we imagine the perfect interface for our operation. We are **telling ourselves a story about how the operation will look from the outside**. Our story won't always come true, but it's better to **start from the best-possible application program interface (API)** and work backward than to make things complicated, ugly, and “realistic” from the get-go.”

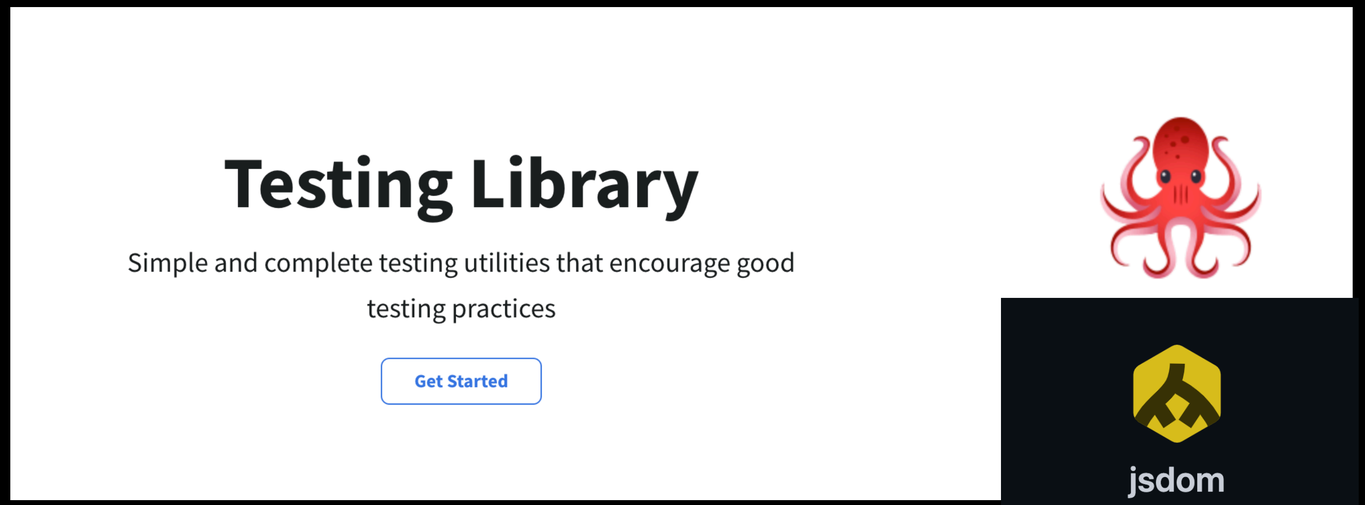
Behaviour is what the business cares about.

If E2E tests gave you faster feedback?

If E2E tests gave you faster feedback?



If E2E tests gave you faster feedback?



Transformations and Side Effects

Transformations and Side Effects

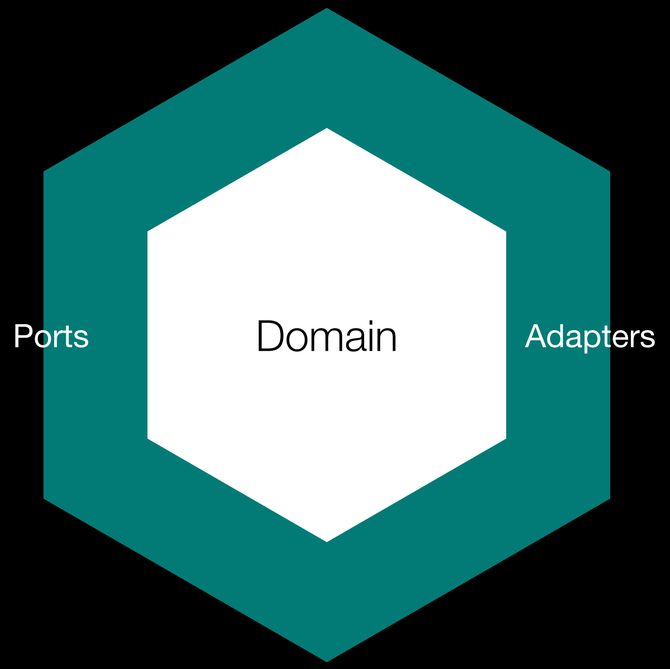
External HTTP Calls
Databases
File System
…

Any kind of I/O
….

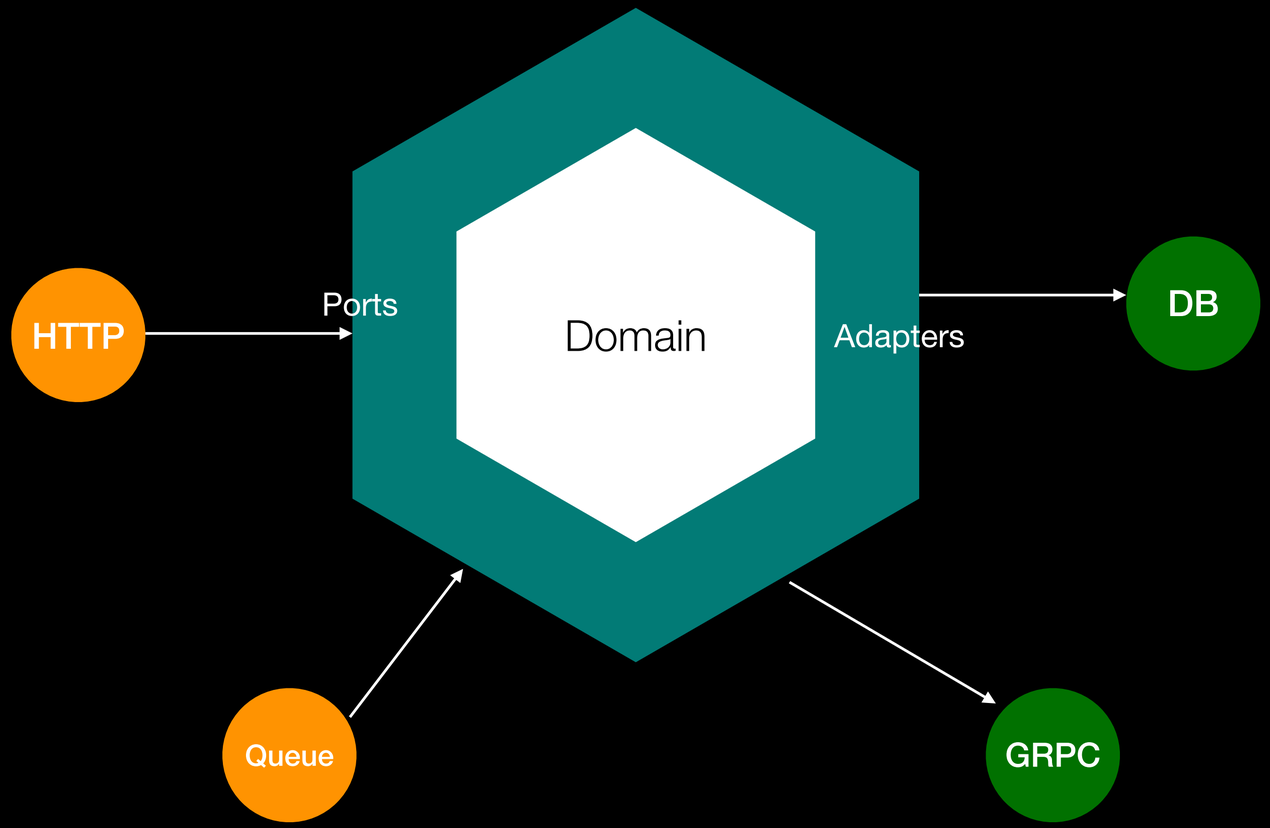
Basically anything that runs beyond your isolated system

System Under Test

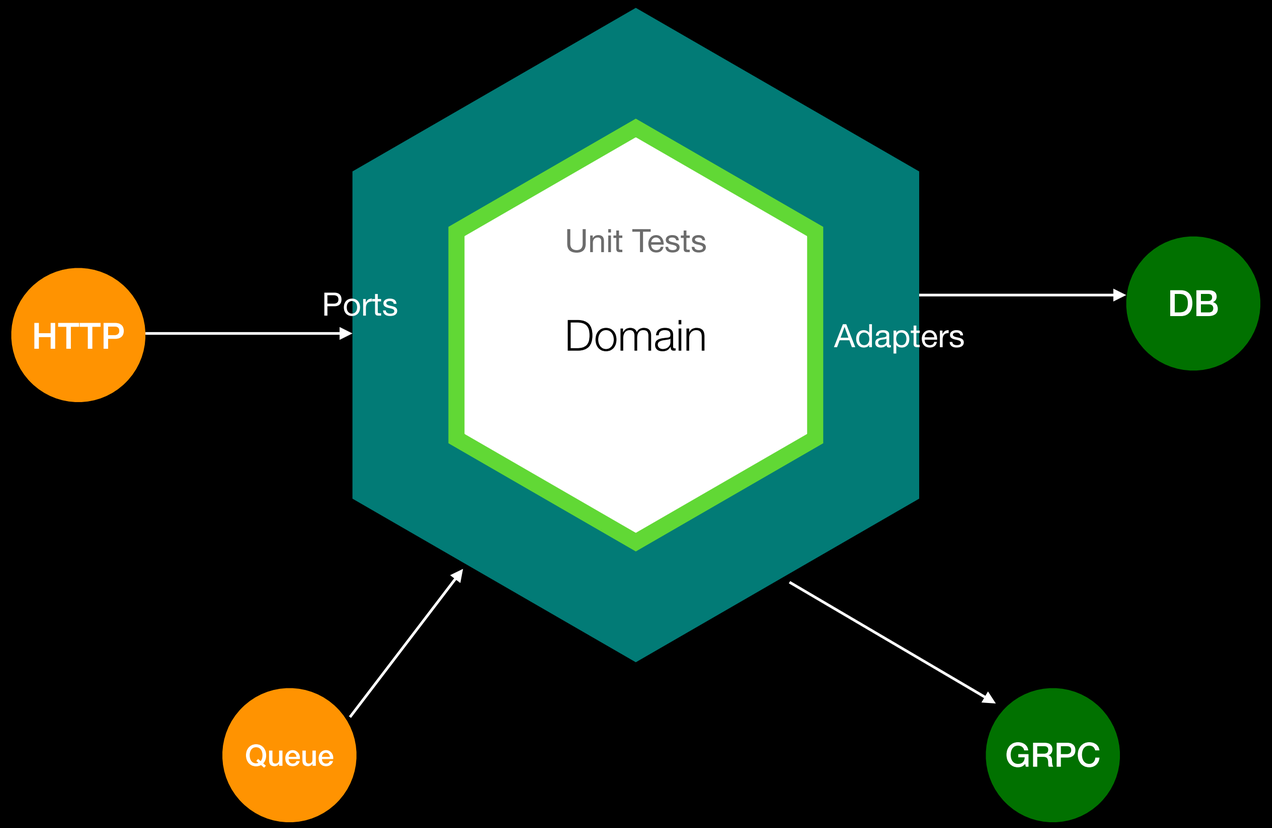
Hexagonal Architecture



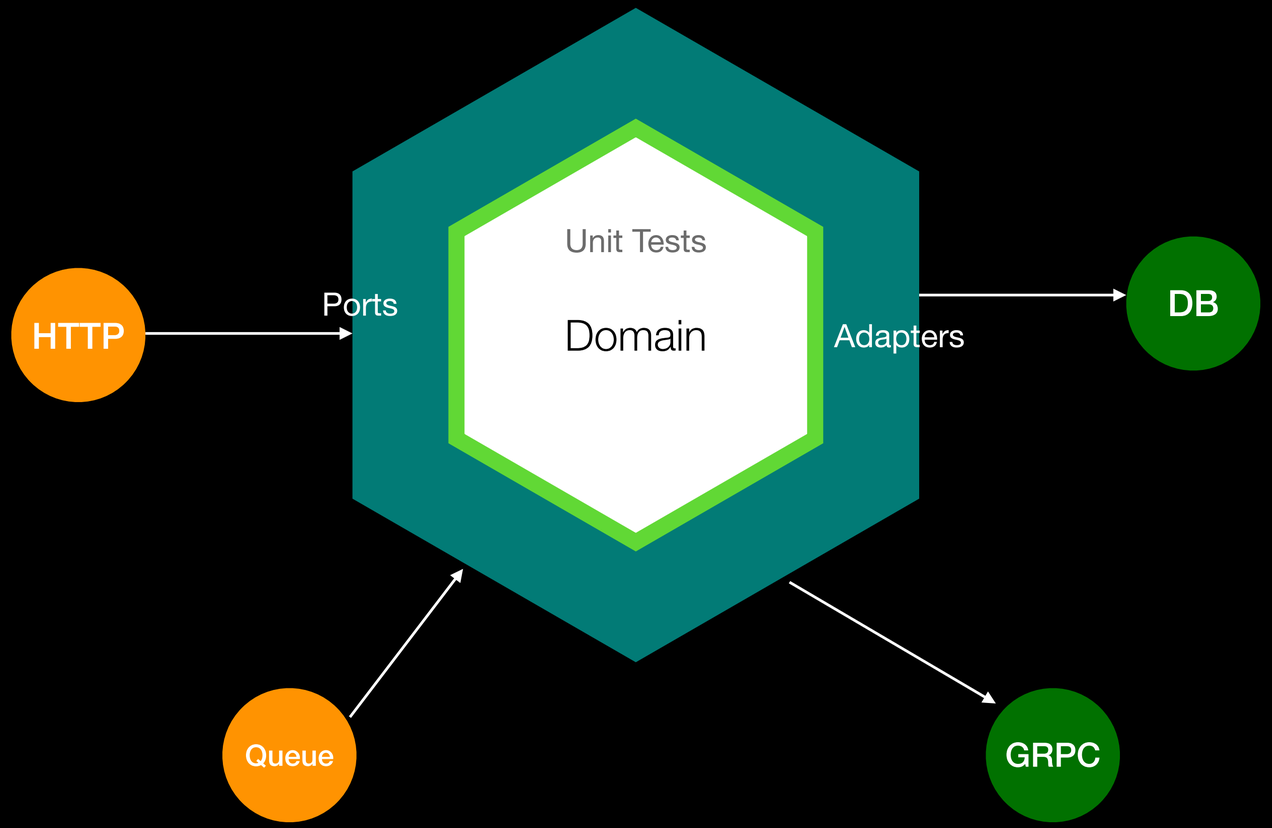
Hexagonal Architecture

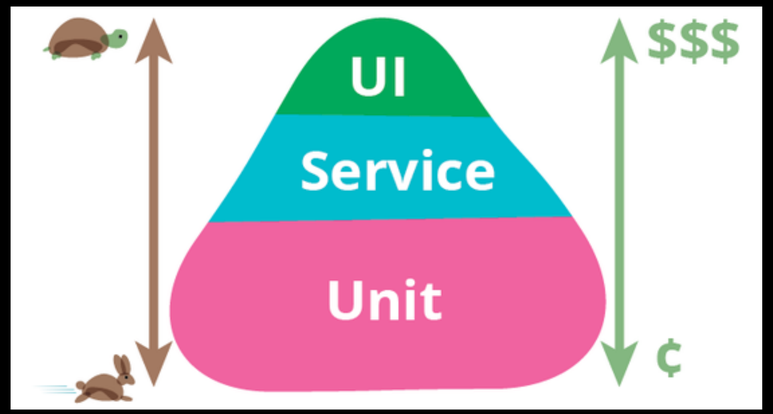


Hexagonal Architecture

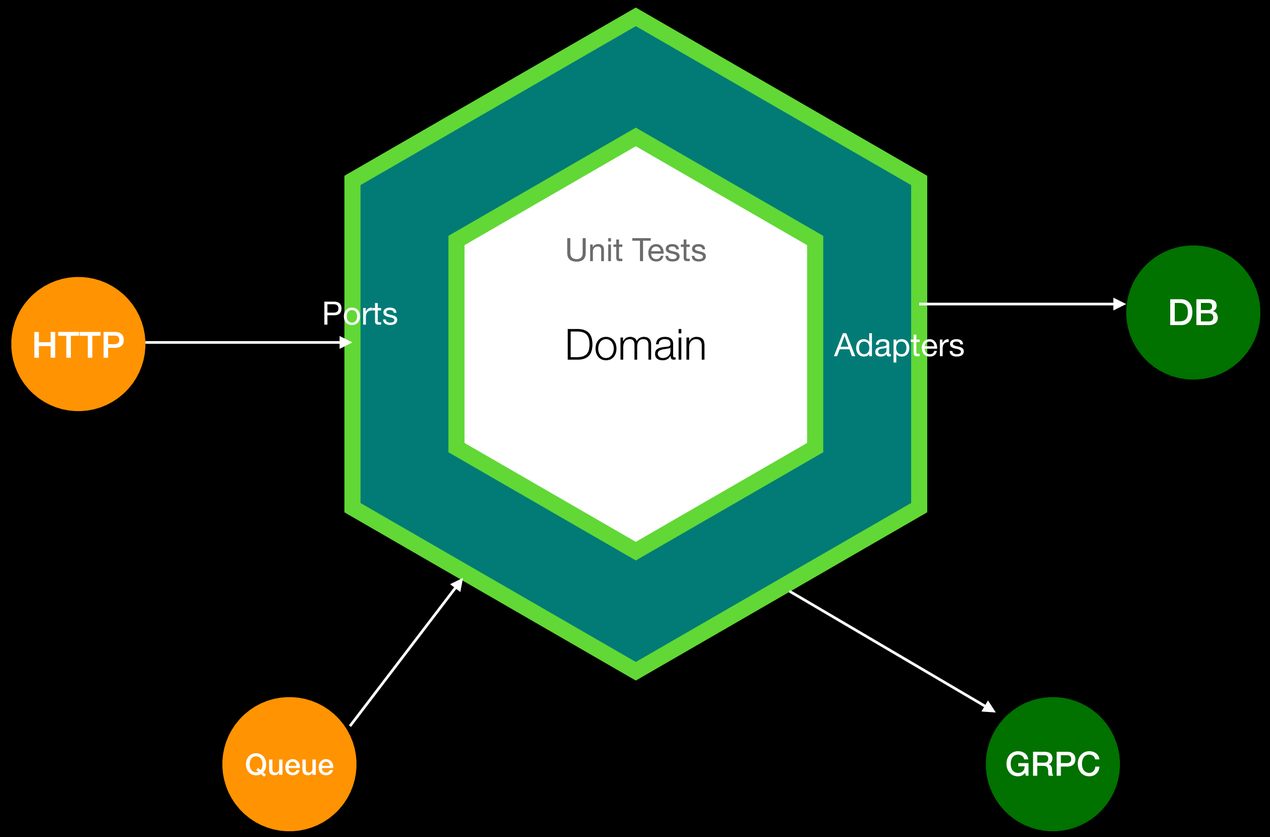


Hexagonal Architecture

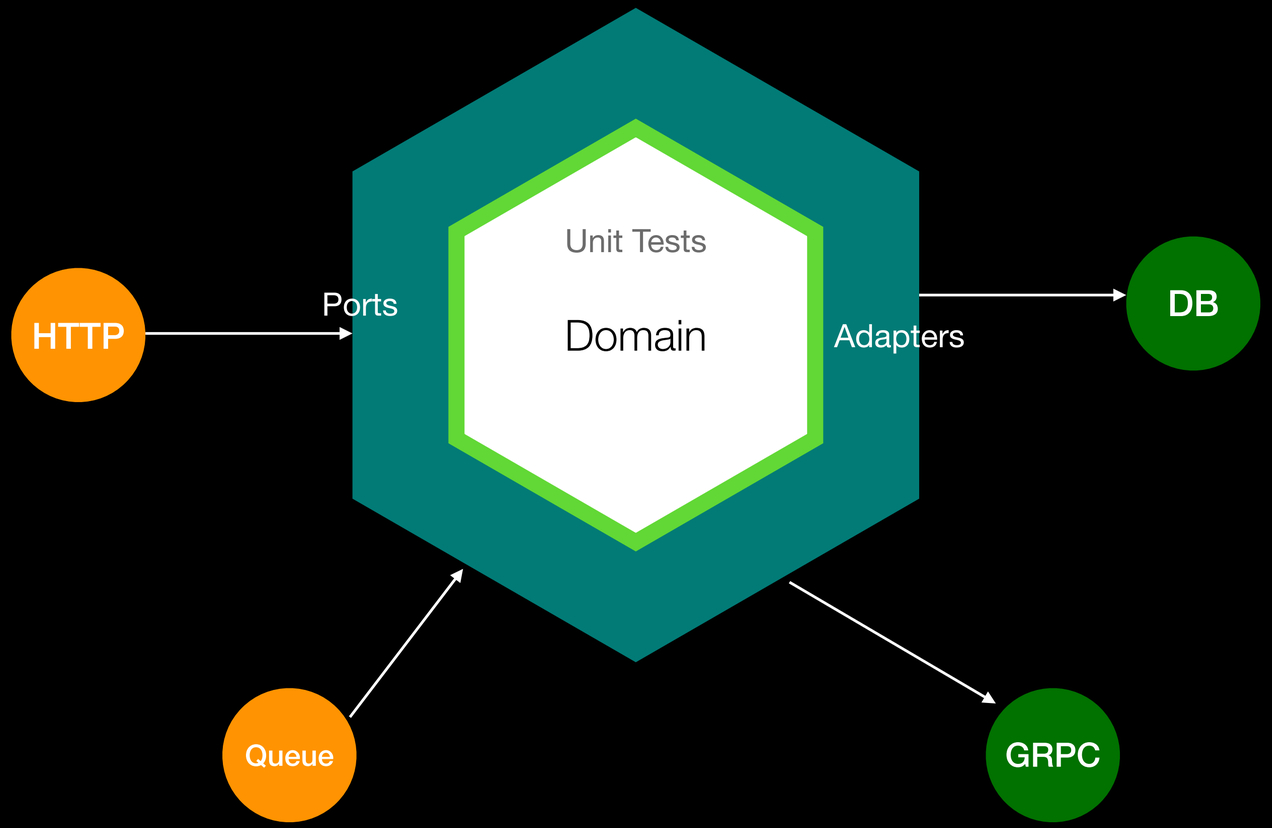



UI
Service
Unit
$$$
¢

Hexagonal Architecture



Back to Unit Tests



Back to Unit Tests

Teams' product owner:

Can we make the user login using OTPs on phones?


Unit
¢

Back to Unit Tests

Teams' product owner:

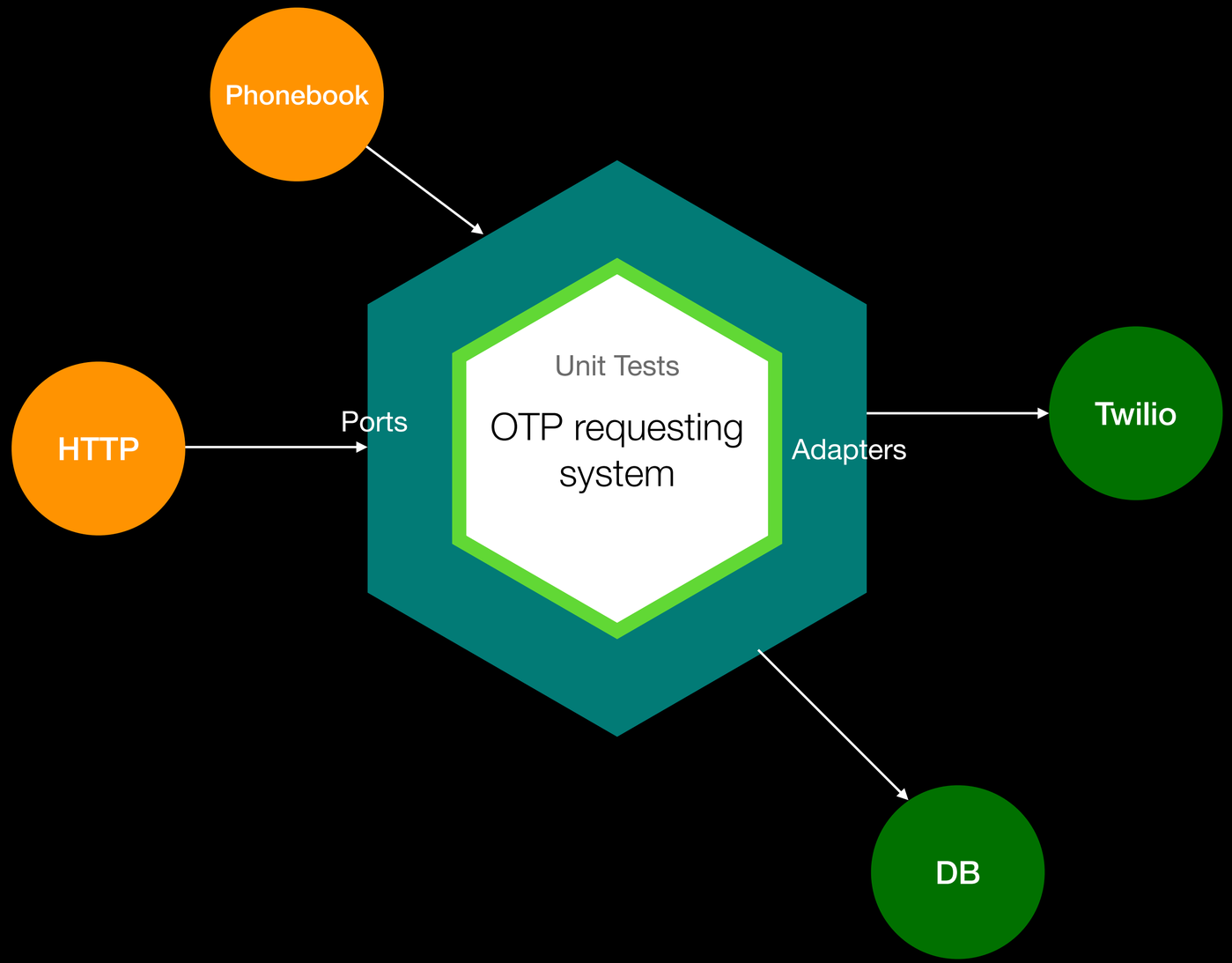
Can we make the user login using OTPs on phones?

Given a phone number of Alice who is a registered user
When Alice wants to login using OTP on their phone
Then give them an OTP

Given a phone number of Bob who is not a registered user
When Bob wants to login using OTP on their phone
Then do not send them an OTP

Given the shared OTP
When Alice shares it with our system
Then our systems must authenticate them in.

Back to Unit Tests



Approach

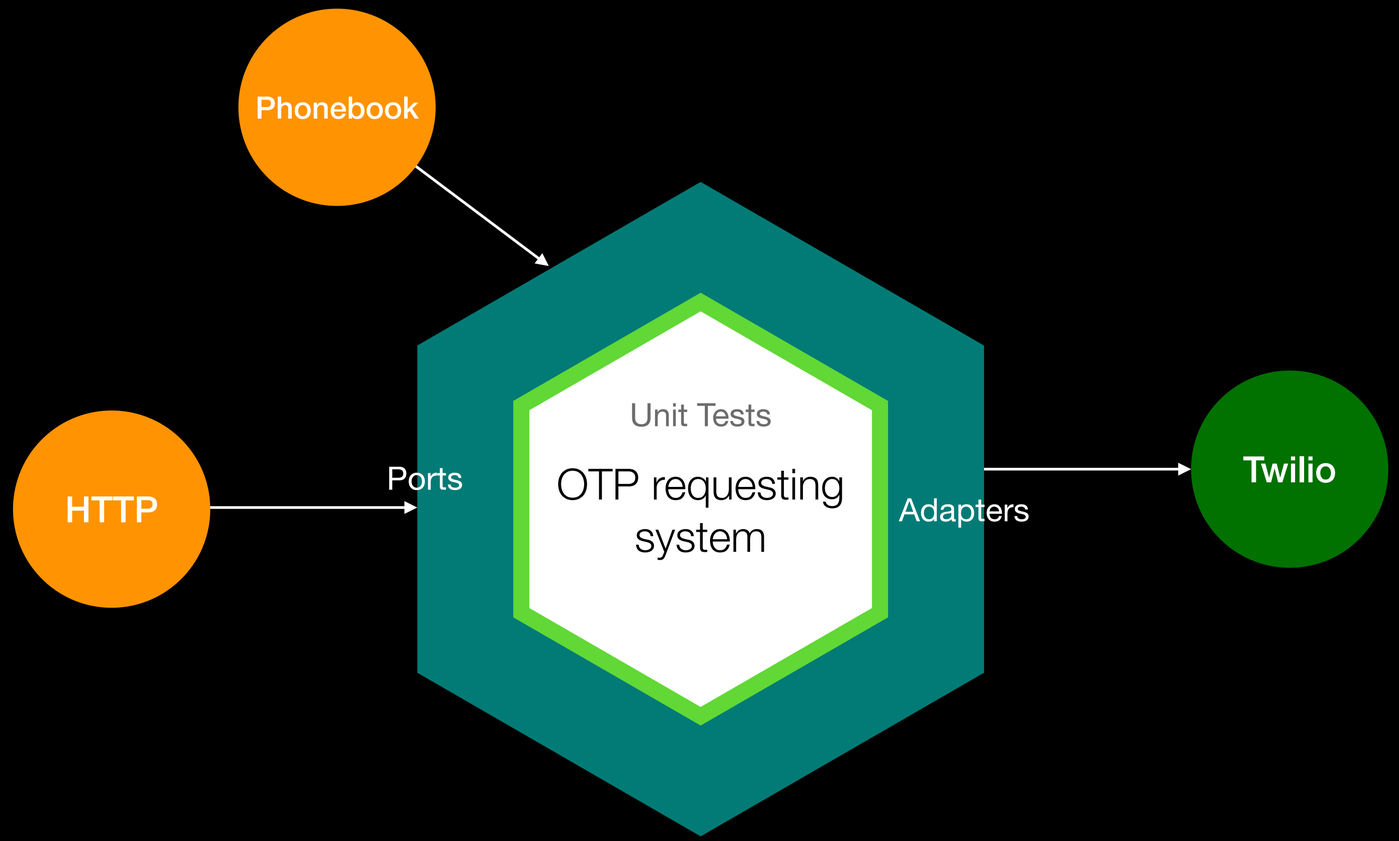
Back to Unit Tests

```java
@Test
void shouldAllowSendingSMSIfRequestedPhoneNumberIsRegistered() {
  ArrayList<String> phonebook = new ArrayList<String>();
  phonebook.add("8899338354");
  String requestedOTPOn = "8899338354";

  OTPSendable otpSendable = prepareOTP(requestedOTPOn, phonebook);

  assertThat(otpSendable.canSend).isTrue();
  assertThat(otpSendable.otp).hasSize(8);
  assertThat(otpSendable.phoneNumber).isEqualTo(requestedOTPOn);
}
```

```java
@Post("/api/otp-request")
@Produces(APPLICATION_JSON)
public HttpResponse getOtp(@Body String requestedOTPOn) { Complexity is 4 Everything is cool!
  ArrayList<String> phonebook = PhoneRepository.fetch();
  OTPSendable otpSendable = prepareOTP(requestedOTPOn, phonebook);
  if(otpSendable.canSend) {
    TwilioClient.send(otpSendable.phoneNumber, otpSendable.otp);
    OTPRepository.store(otpSendable);
    return HttpResponse.created();
  } else {
    return HttpResponse.unauthorized();
  }
}
```



Back to Unit Tests

```
@Test
void shouldAllowSendingSMSIfRequestedPhoneNumberIsRegistered() {
```

```
@Post("/api/otp-request")
@Produces(APPLICATION_JSON)
public HttpResponse getOtp(@Body String requestedOTPOn) {
  ArrayList<String> phonebook = PhoneRepository.fetch();
  OTPSendable otpSendable = prepareOTP(requestedOTPOn, phonebook);
  if(otpSendable.canSend) {
    TwilioClient.send(otpSendable.phoneNumber, otpSendable.otp);
    OTPRepository.store(otpSendable);
    return HttpResponse.created();
  } else {
    return HttpResponse.unauthorized();
  }
}
```

Complexity is 4 Everything is cool!

Phonebook

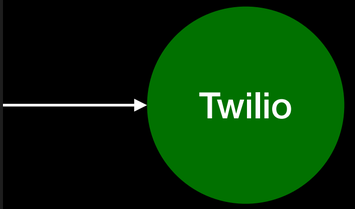
Twilio

Back to Unit Tests

```java
@Test
void shouldAllowSendingSMSIfRequestedPhoneNumberIsRegistered() {
    ArrayList<String> phonebook = new ArrayList<String>();
    phonebook.add("8899338354");
    String requestedOTPOn = "8899338354";

    OTPSendable otpSendable = prepareOTP(requestedOTPOn, phonebook);

    assertThat(otpSendable.canSend).isTrue();
    assertThat(otpSendable.otp).hasSize(8);
    assertThat(otpSendable.phoneNumber).isEqualTo(requestedOTPOn);
}
}
```



Back to Unit Tests

```
@Test
void shouldAllowSendingSMSIfRequestedPhoneNumberIsRegistered() {
```

```
@Post("/api/otp-request")
@Produces(APPLICATION_JSON)
public HttpResponse getOtp(@Body String requestedOTPOn) {
  ArrayList<String> phonebook = PhoneRepository.fetch();
  OTPSendable otpSendable = prepareOTP(requestedOTPOn, phonebook);
  if(otpSendable.canSend) {
    TwilioClient.send(otpSendable.phoneNumber, otpSendable.otp);
    OTPRepository.store(otpSendable);
    return HttpResponse.created();
  } else {
    return HttpResponse.unauthorized();
  }
}
```

Complexity is 4 Everything is cool!

Phonebook

Twilio

Mocking

Mocking : Bad Idea

It is slow and stateful.

Mocking : Bad Idea

It is slow and stateful.

They end up making you test implementation thus create hell when refactoring.

Mocking : Bad Idea

It is slow and stateful.

They end up making you test implementation thus create hell when refactoring.

If you are having to use a lot of mocks, that is an alarm of growing complexity.

Mocking makes us look away from the problem

Mocking : Bad Idea

It is slow and stateful.

They end up making you test implementation thus create hell when refactoring.

If you are having to use a lot of mocks, that is an alarm of growing coupling.
Mocking makes us look away from the problem

Mocks are more work, and more code.

Mocking : Bad Idea

It is slow and stateful.

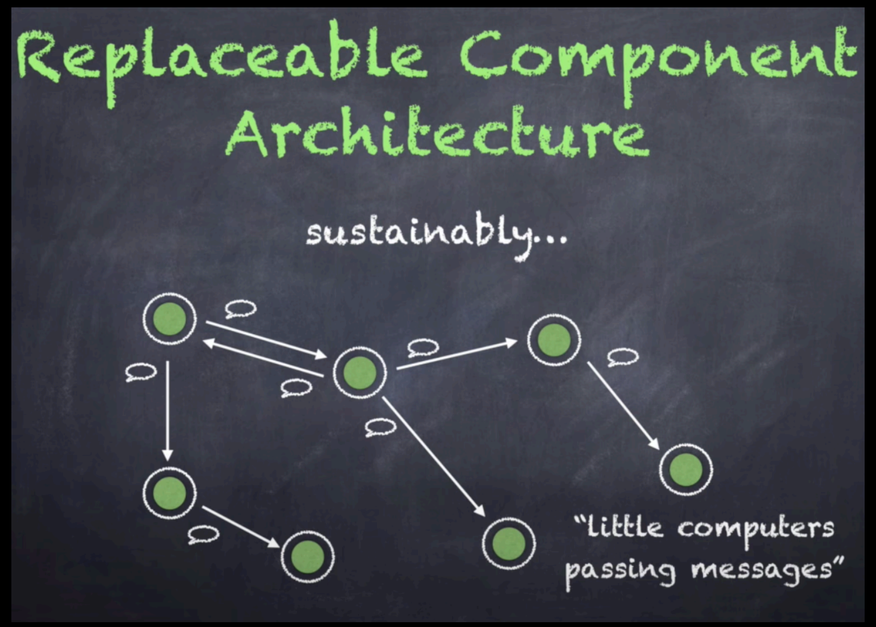
They end up making you test implementation thus create hell when refactoring.

If you are having to use a lot of mocks, that is an alarm of growing coupling.
Mocking makes us look away from the problem

Mocks are more work, and more code.

Use test doubles instead if you really need to mock.

How microservices should make you write better software and tests.



The social impact of tests

Takeaways

“kill as much code as you can”

“kill as much code as you can”

“we read code more than we write, so optimise for reading” true for tests

“kill as much code as you can”

“we read code more than we write, so optimise for reading” true for tests

“test behaviour and never the implementation”

“kill as much code as you can”

“we read code more than we write, so optimise for reading” true for tests

“test behaviour and never the implementation”

“tests will help you remove coupling, don’t look away by mocking away everything”

“kill as much code as you can”

“we read code more than we write, so optimise for reading” true for tests

“test behaviour and never the implementation”

“tests will help you remove coupling, don’t look away by mocking away everything”

“don't use mocks”

“kill as much code as you can”

“we read code more than we write, so optimise for reading” true for tests

“test behaviour and never the implementation”

“tests will help you remove coupling, don’t look away by mocking away everything”

“don't use mocks”

“you really don’t need such a large test suite”

“kill as much code as you can”

“we read code more than we write, so optimise for reading” true for tests

“test behaviour and never the implementation”

“tests will help you remove coupling, don’t look away by mocking away everything”

“don't use mocks”

“you really don’t need such a large test suite”

“test your public interface, that gives you more confidence, this is what the business care about”

“kill as much code as you can”

“we read code more than we write, so optimise for reading” true for tests

“test behaviour and never the implementation”

“tests will help you remove coupling, don’t look away by mocking away everything”

“don't use mocks”

“you really don’t need such a large test suite”

“test your public interface, that gives you more confidence, this is what the business care about”

“choose tools that give you faster feedback”

Rahul C

 github.com/bitsapien

 twitter.com/bitsapien_logs

 blog.bitsapien.dev/now

References

Dan North : Software that fits in your head - https://www.youtube.com/watch?v=4Y0tOi7QWqM
Ian Cooper: TDD, Where did it all go wrong - https://www.youtube.com/watch?v=EZ05e7EMOLM
DHH: TDD is Dead, Long Live Testing - https://dhh.dk/2014/tdd-is-dead-long-live-testing.html
TW Hangouts: Is TDD Dead? - https://www.youtube.com/watch?v=z9quxZsLcfo
Hexagonal Architecture - https://fideloper.com/hexagonal-architecture