Test SMART not HARD

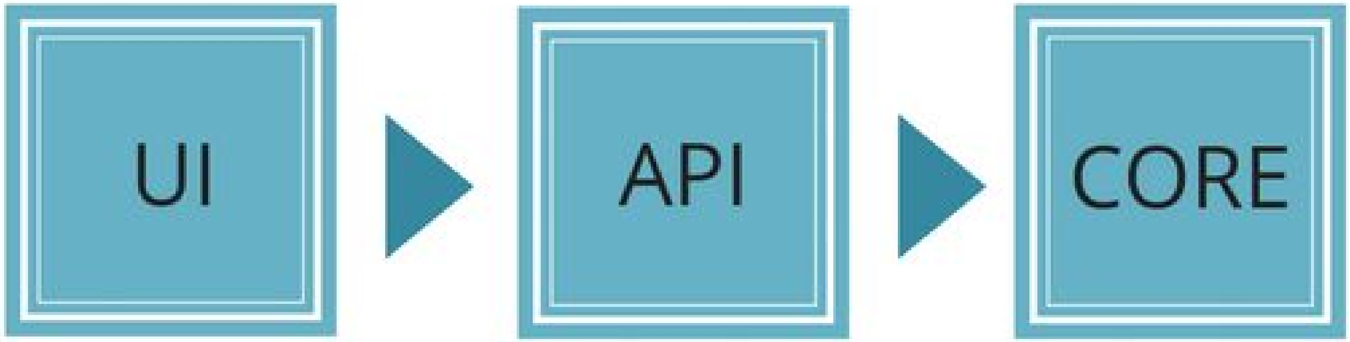
and deliver customer value fast

Beth Skurrie (DiUS)
@bethesque

DiUS
WHERE IDEAS ARE ENGINEERED

Soundify

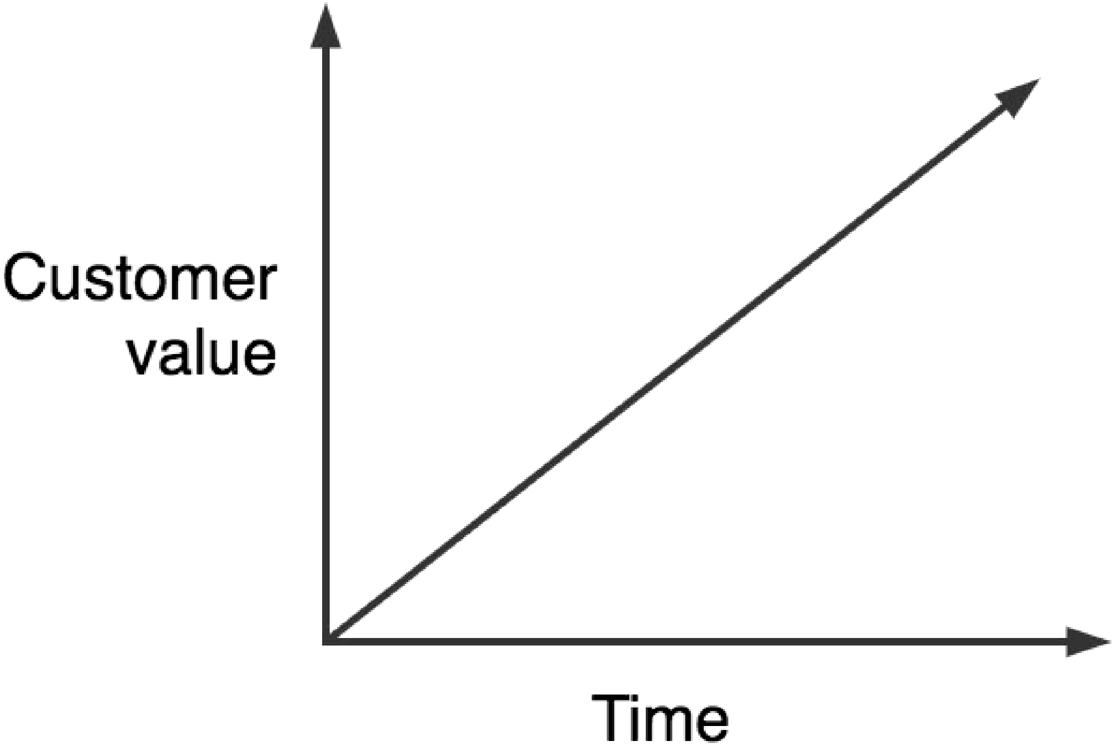
soundify


UI
API
CORE

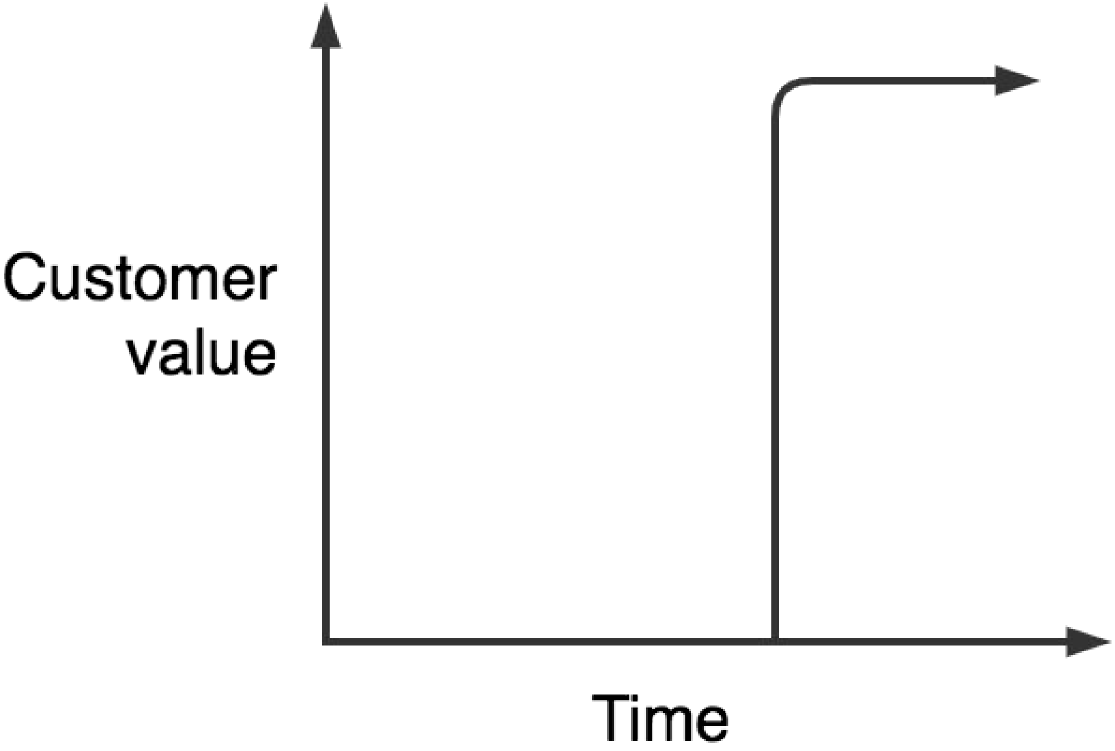
Process

- Experience and API
  - Short iterations
- Core
  - Design
  - Build
  - Test
  - Release

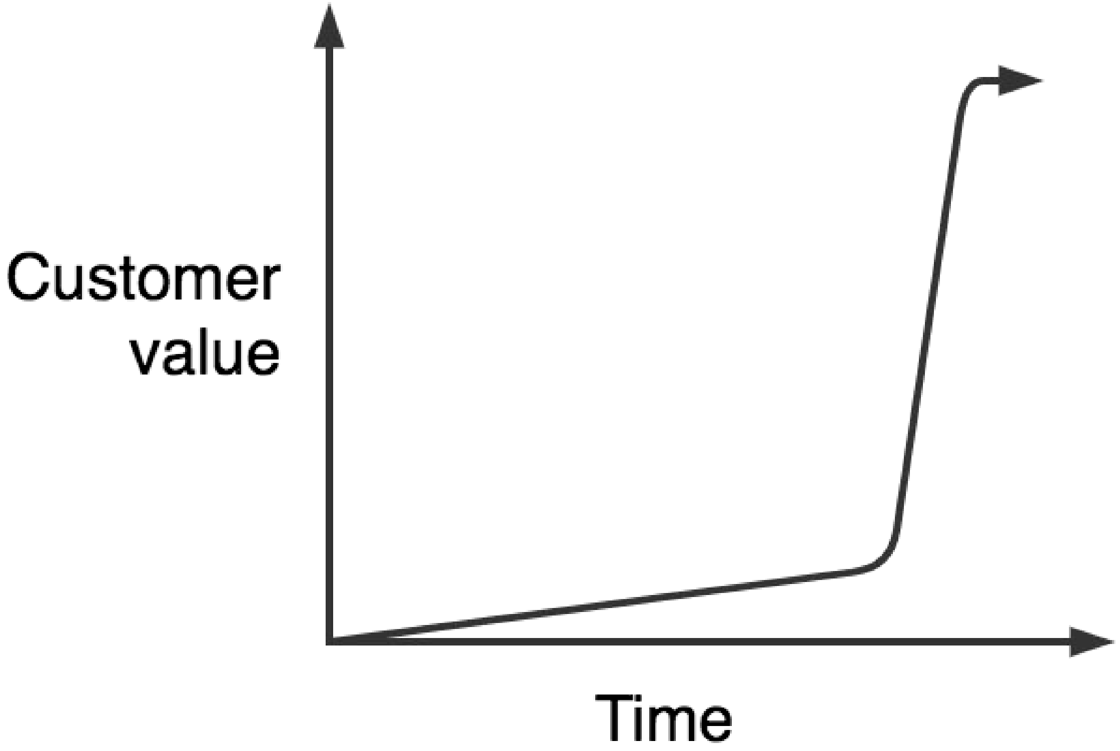
Ideal customer value over time



Core platform customer value over time



Program customer value over time

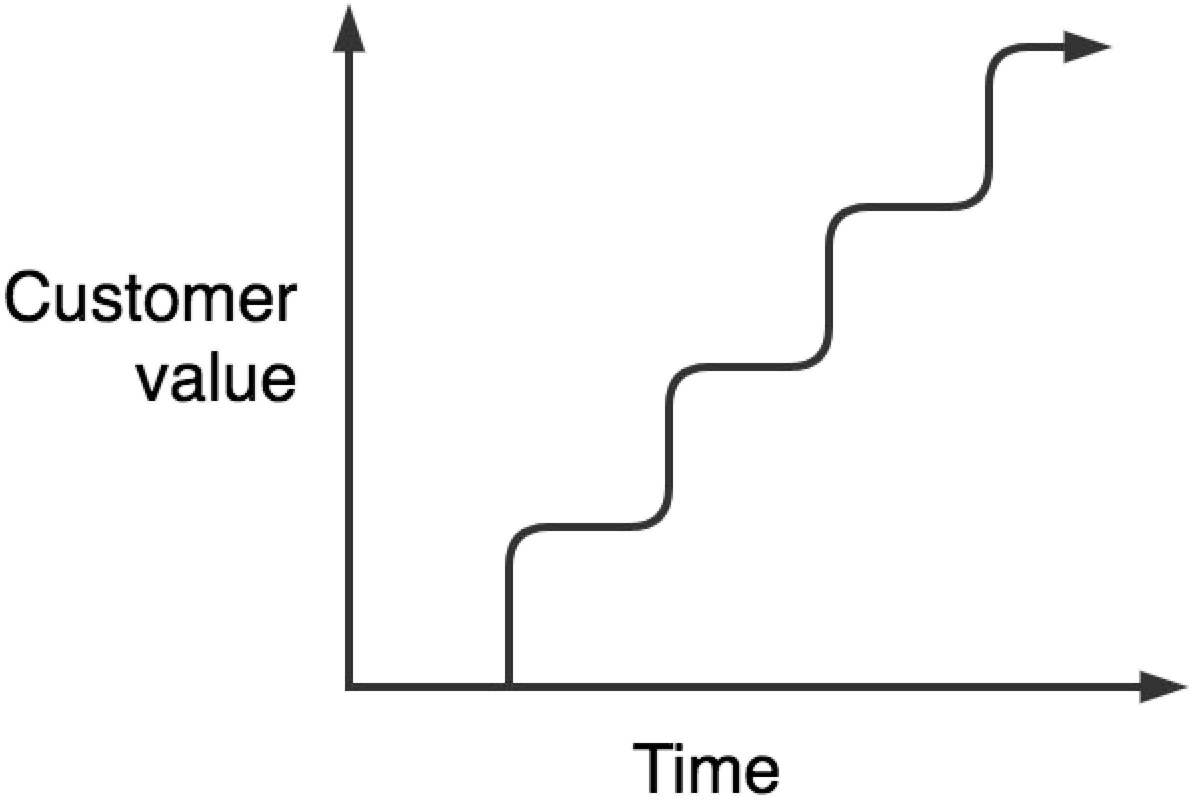


Problems

- Customers 😞
- Business owner 😞
- Delivery manager 😞
- Teams 😞

Core platform team - iterate

Improved customer value over time!



Core team iterations

Solved problems

- Incremental customer value
- Reduced risk

New problems

- Overhead in releases
- Need more testers
- 4 week turnaround for integration bugs

Automated releases

Automated e2e tests

Much smug! So dream!

Automated e2e tests

Solved problems

- Reduced risk
- Integration bug turnaround - days
- Confidence - when tests passed

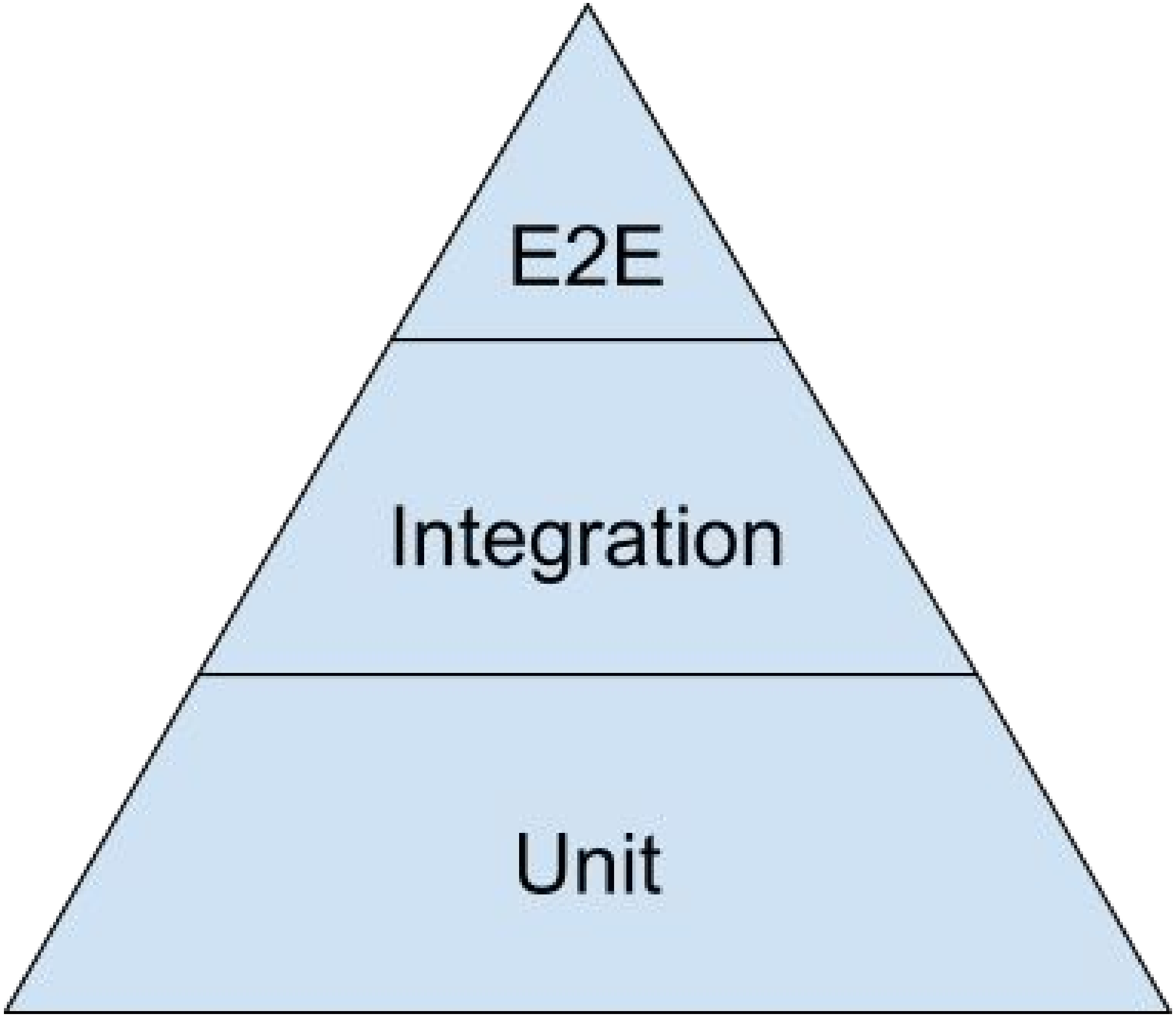
New problems

- Slow tests
- Easy to break
- Hard to fix
- Scale badly
- Lots of set up
- Flakey tests ignored
- Takes dev time away from features

Is this worth it?

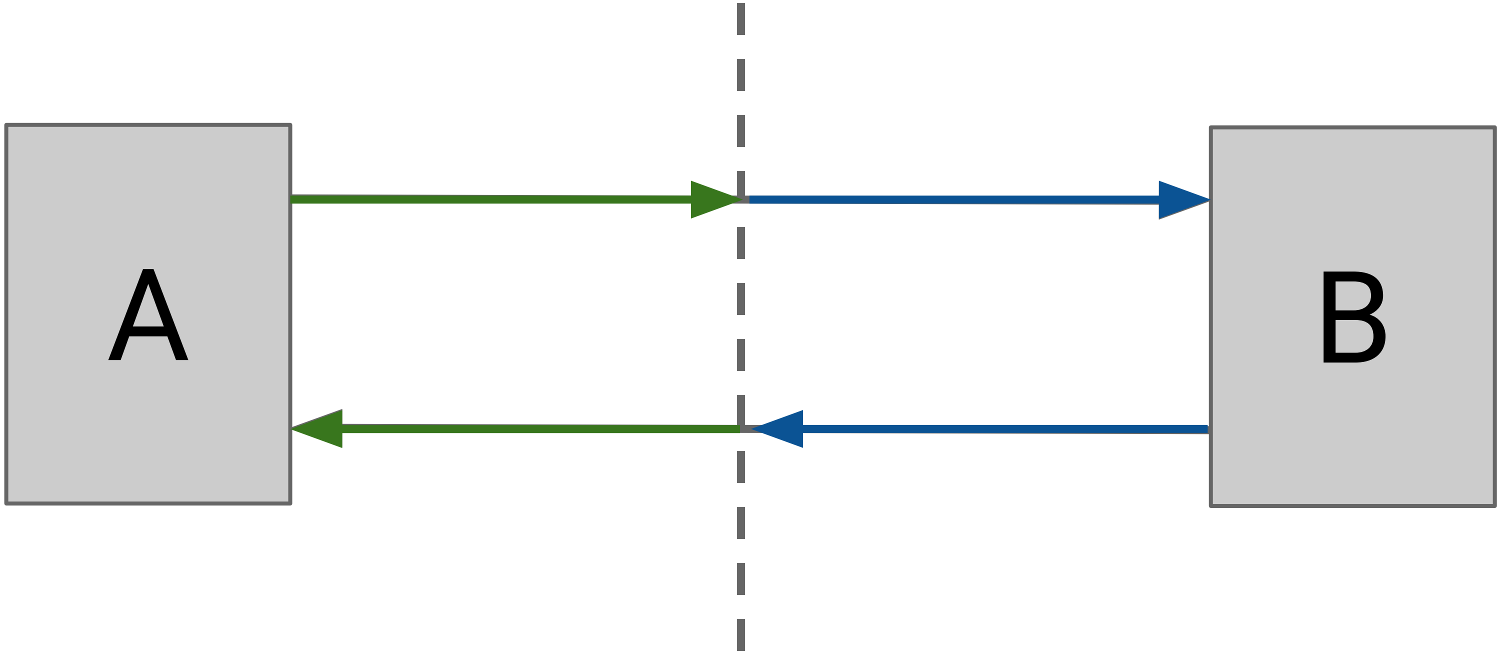
Why do we do integration tests?

We want to find bugs *before* they get to production


E2E
Integration
Unit

To integration test everything or not to integration test everything?

What if there was another way?



mock

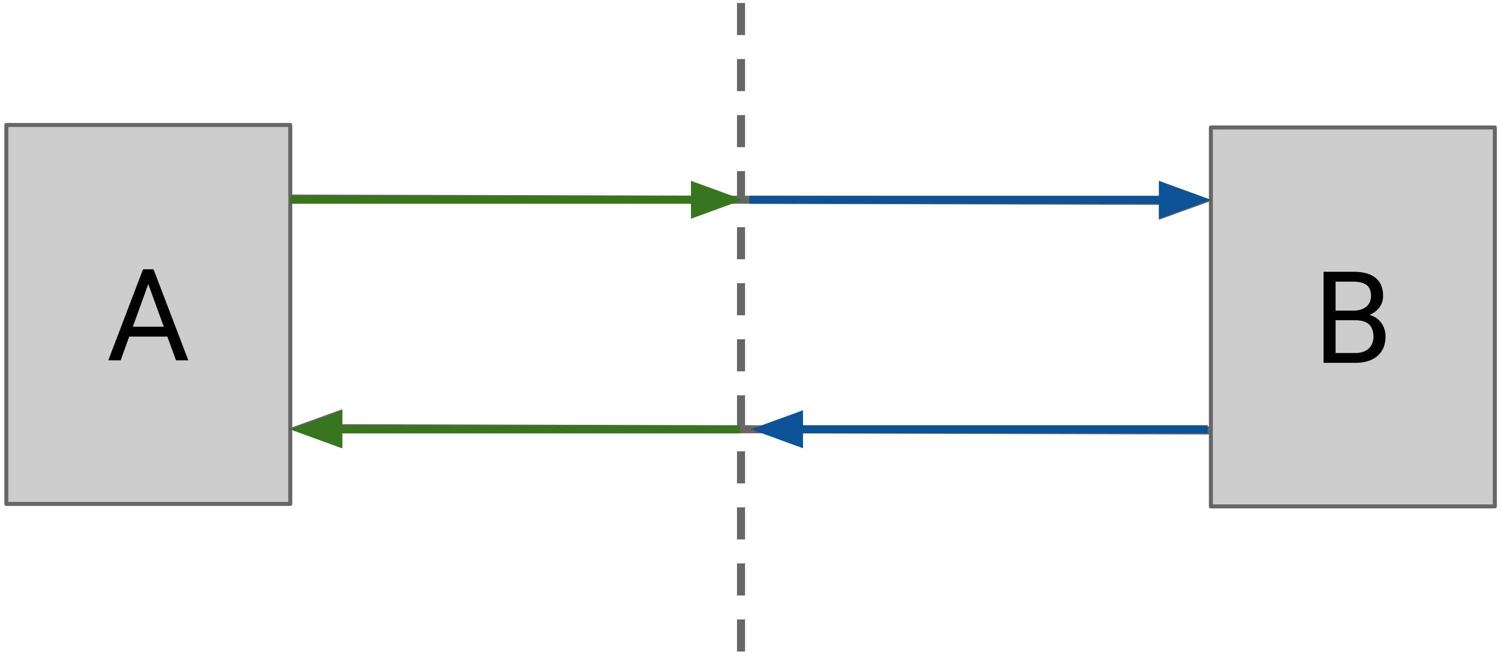
Test symmetry

Solved problems

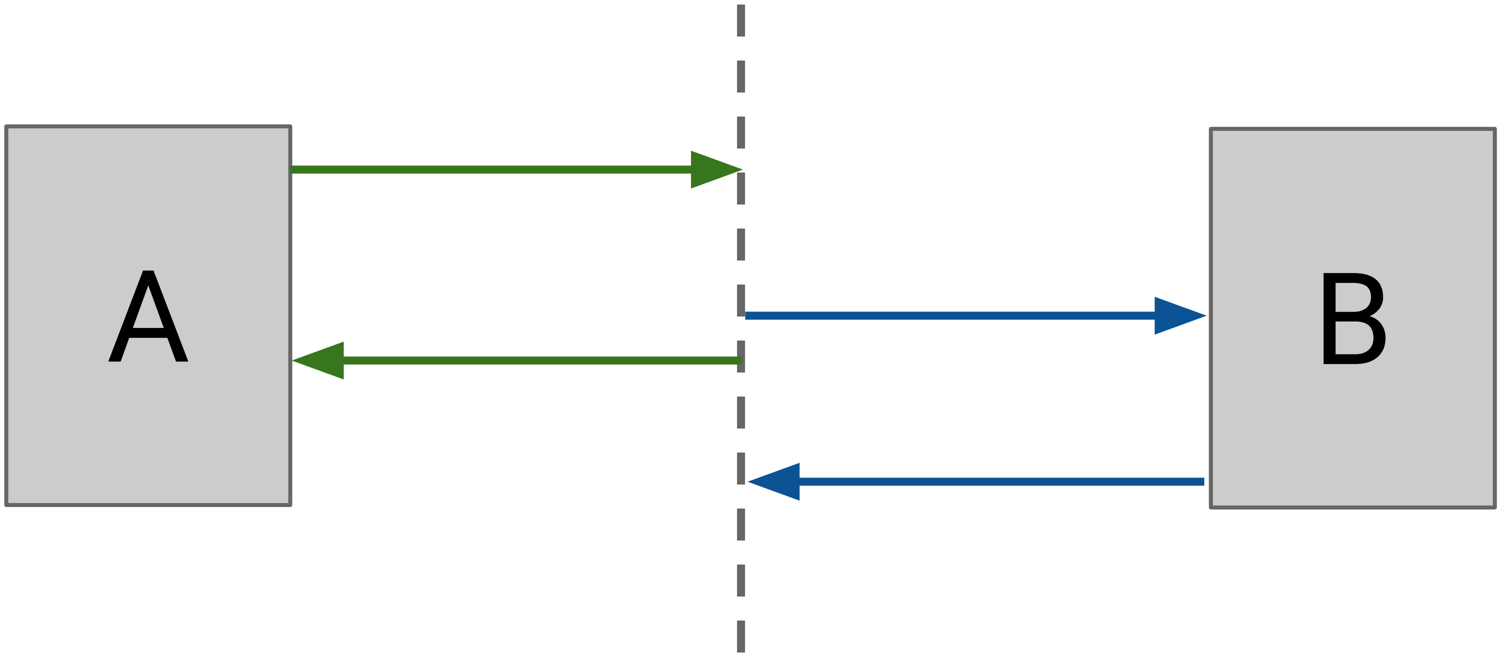
- Fast feedback
- Few dependencies
- Reliable
- Easy to debug
- No extra infrastructure

New problems

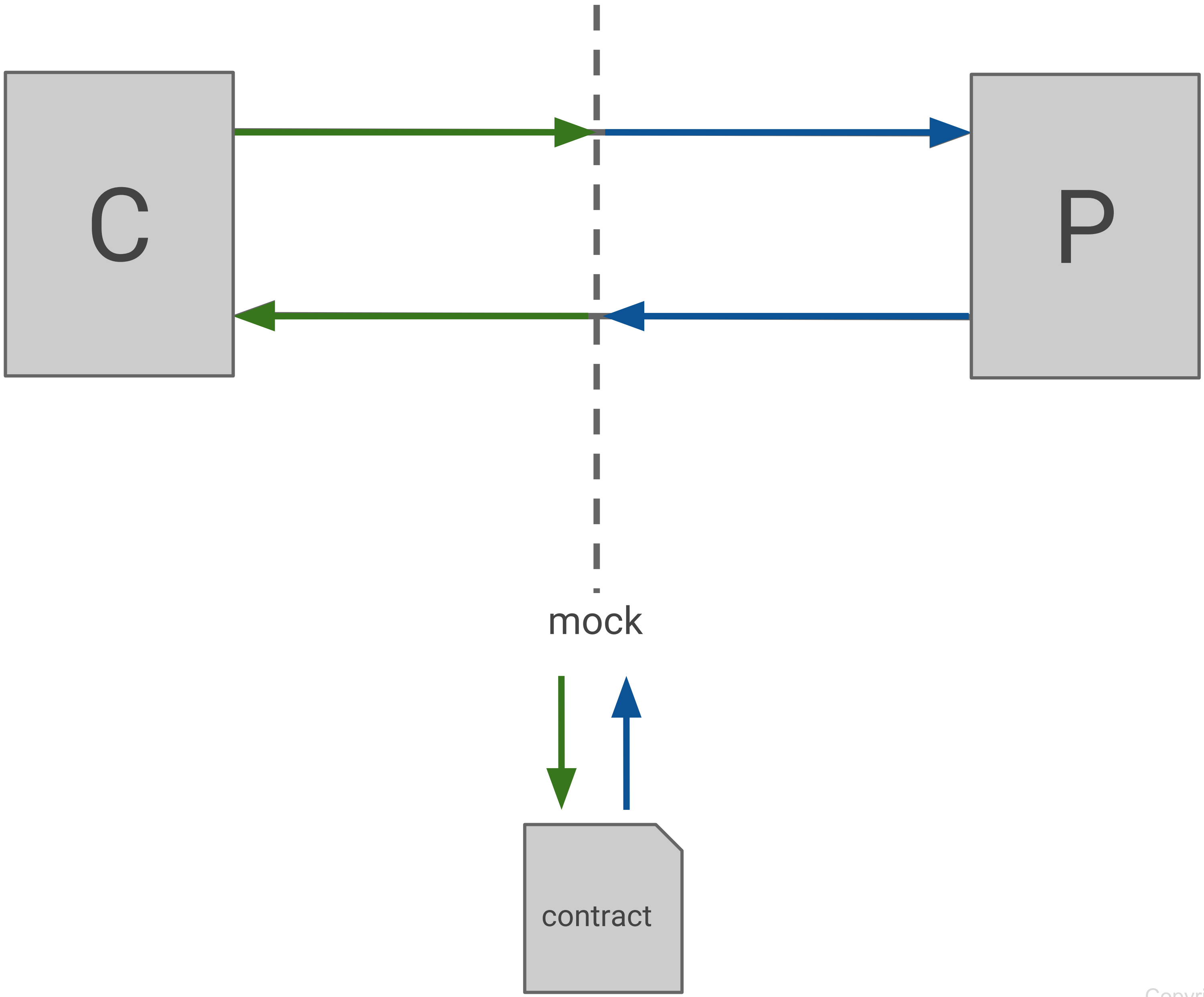
- Hard to keep both sides in sync



mock



mock


C
P
mock
contract

Contracts

Solved problems

- Keeping tests in sync

New problems

- ???

Contracts tests

Bug turnaround - minutes

Contracts tests

Know before you commit

Contracts tests

Make changes with speed and confidence

Contracts tests

Deploy independently

(Consumer) Contracts tests

Better API design

Contracts tests

Are **not** functional tests

Contracts tests

Are **not** load performance tests SLAs

(Consumer) Contracts tests

Are **not** good for public APIs

Contracts tests

Are **not** a silver bullet!

Do I still need integration tests?

Integration test coverage

$$\propto$$

Consequences of bug
x
Time to find bug
x
Time to release fix

Speed up your releases

Do less

- Integration testing

Do more

- Aggregated logging
- Metrics
- Semantic logging
- Alerting

Another story

This time a real one!

PACT

pact.io

pact - noun

A formal agreement between individuals or parties. *"the country negotiated a trade pact with the US"*

synonyms: agreement, protocol, deal, contract

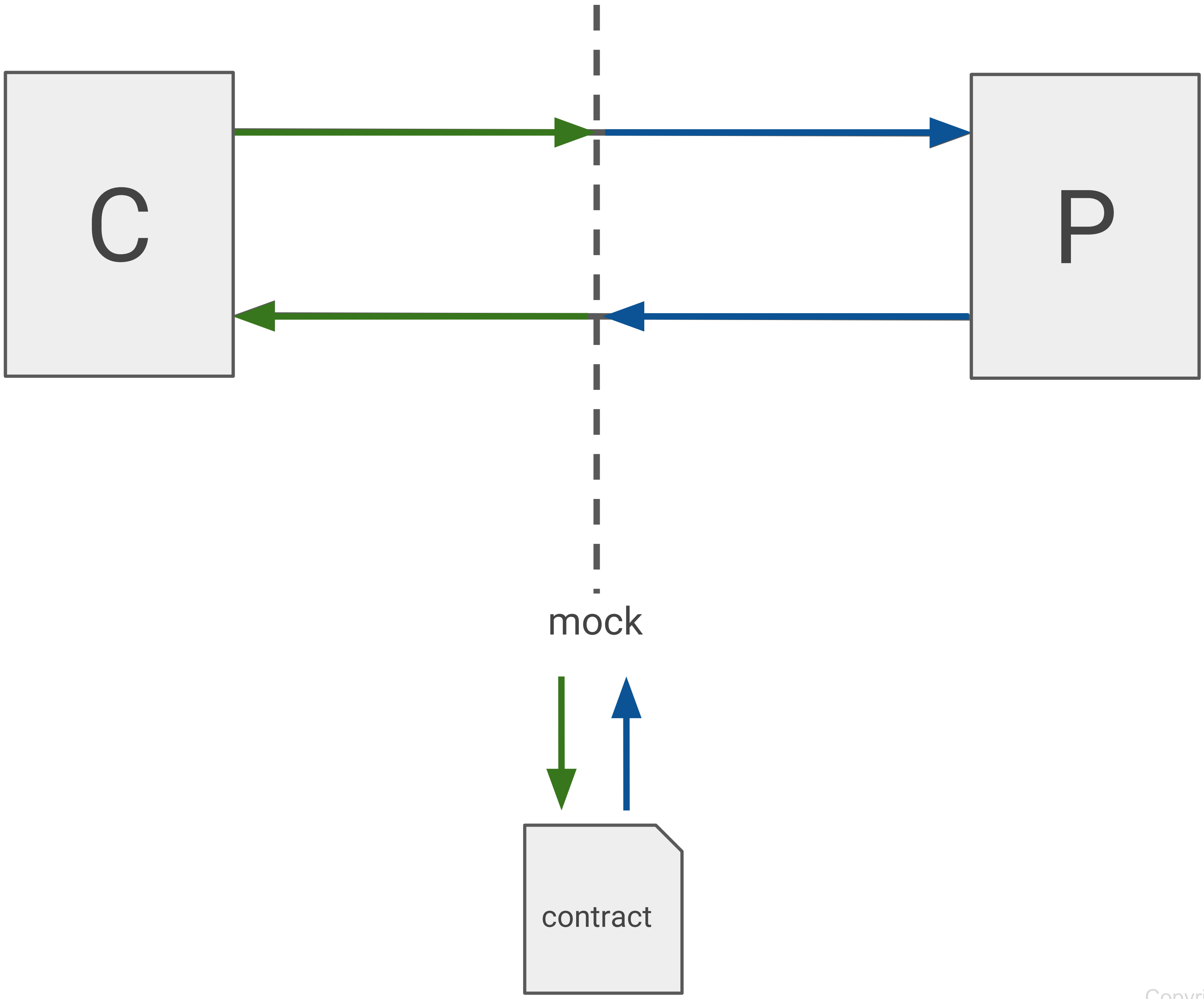
PACT

pact.io

- Open source
- Multiple languages
    - JVM
    - .NET
    - Js
    - Python
    - + more

- HTTP contracts
- Message contracts

Your contract testing story

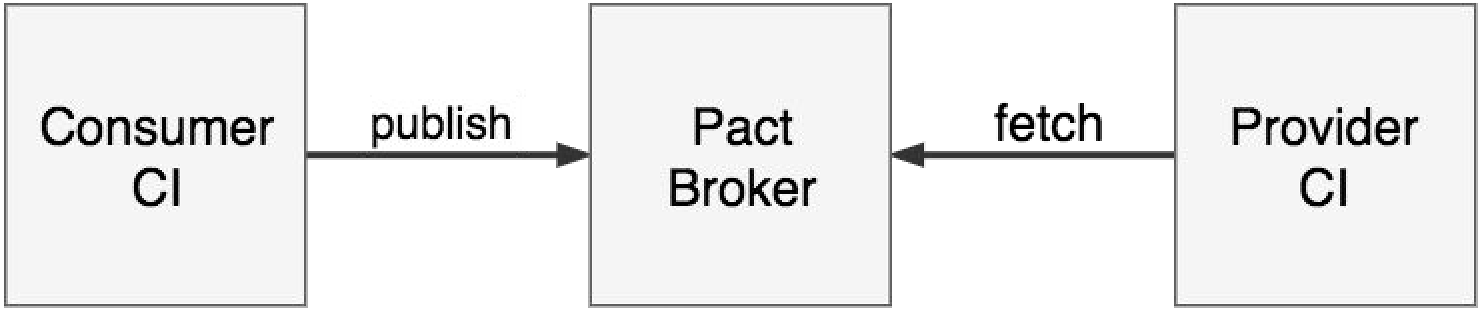

C
P
mock
contract

Consumer CI

?

Provider CI

Pact Broker



Automate the contract exchange

WHEN

the provider receives

<some request>

THEN

it will return

<some response>

WHEN

the provider receives

a GET request for /alligators/Mary

THEN

it will return

a 200 OK response

with a JSON body {“name”: “Mary”}

WHEN

the provider receives

a GET request for /alligators/Mary

THEN

it will return

a 200 OK response

a 404 Not Found response

WHEN

the provider receives

a GET request for /alligators/Mary

THEN

it will return

a 404 Not Found

GIVEN

<the provider is in a certain state>

WHEN

the provider receives

<some request>

THEN

it will return

<some response>

You still need to think about test data

Your contract testing journey

- Automate the contract exchange
- You still need to think about test data

Contracts should focus on the messages, not the technology

Your contract testing journey

- Automate the contract exchange
- You still need to think about test data
- Contracts should focus on the messages, not the technology

Contracts should be as flexible as possible - but no more

Your contract testing journey

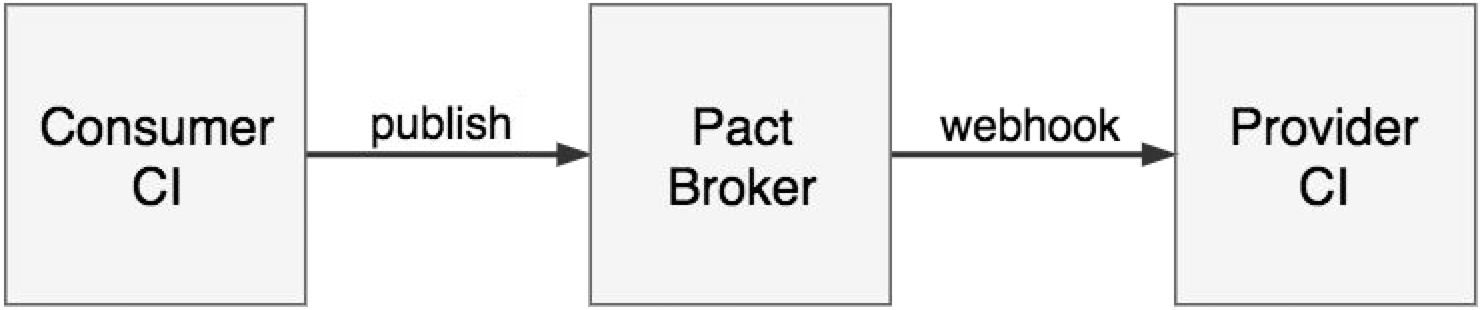
- Automate the contract exchange
- You still need to think about test data
- Contracts should focus on the messages, not the technology
- Contracts should be as flexible as possible - but no more

The other service needs to know when a contract has changed

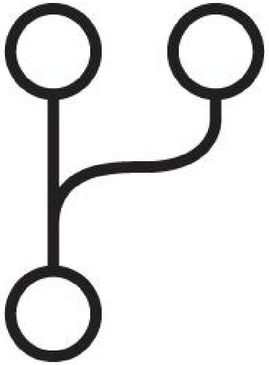
You need a way to introduce changes without breaking everything

Contracts are not a substitute for good communication between teams

Pact Broker webhooks



Pact Broker tags



Contracts are STILL not a substitute for good communication between teams

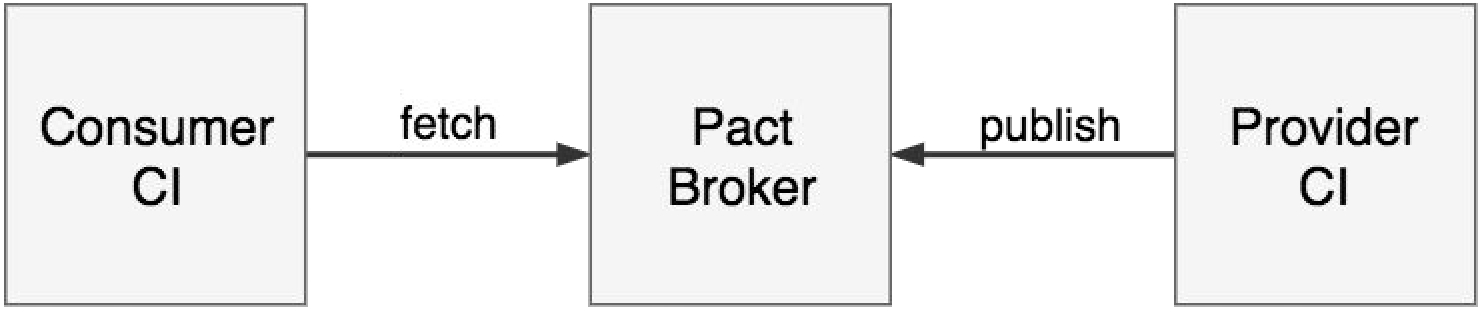
Your contract testing journey

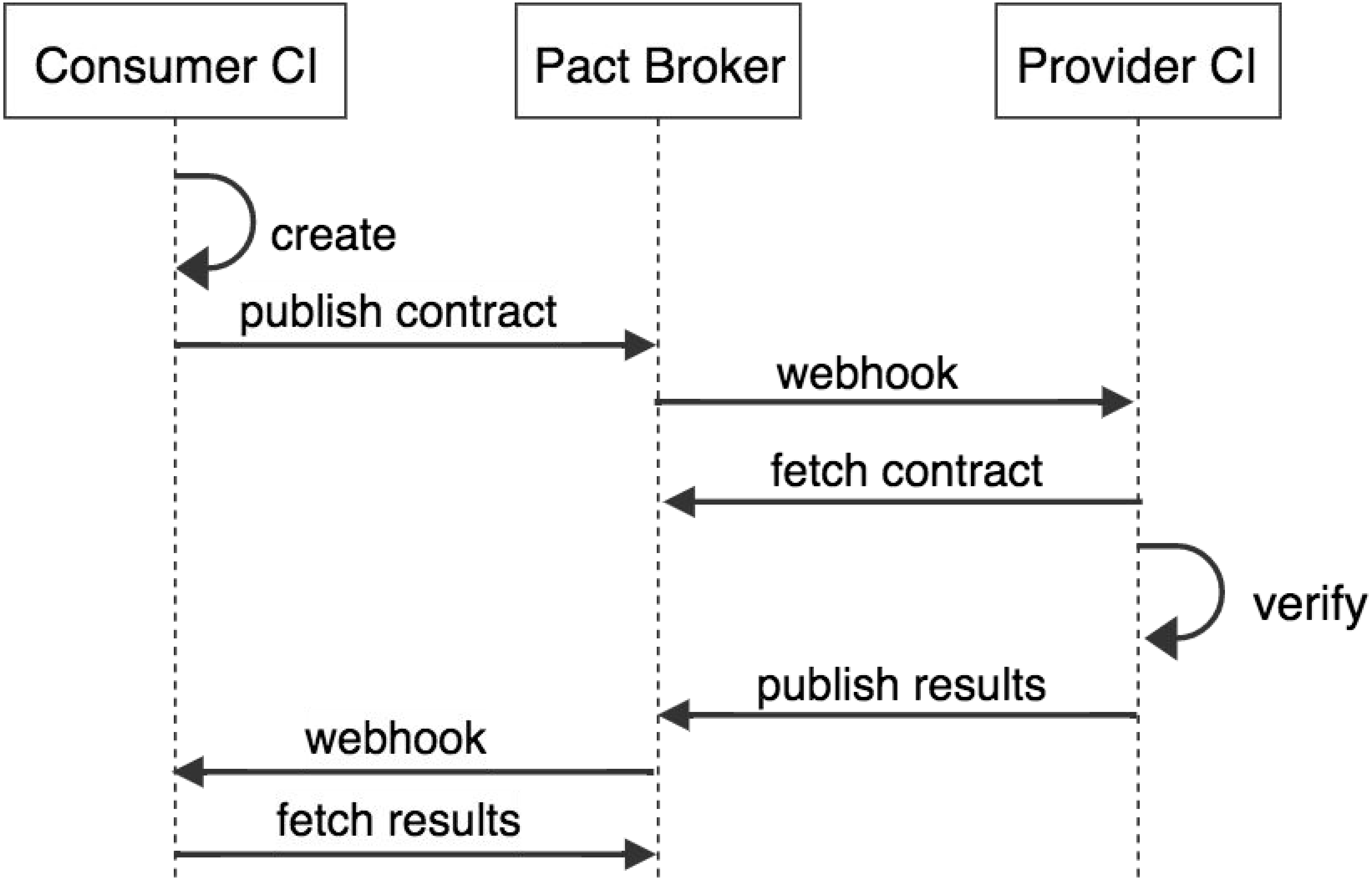
- Automate the contract exchange
- You still need to think about test data
- Contracts should focus on the messages, not the technology
- Contracts should be as flexible as possible - but no more
- The provider needs to know when a contract has changed
- You need a way to introduce changes without breaking everything
- Remember: contracts are not a substitute for good communication between teams

Consumer CI
?
Provider CI

You need to share the verification results as well as the contracts

Pact Broker verifications




Consumer CI
Pact Broker
Provider CI
create
publish contract
webhook
fetch contract
verify
publish results
webhook
fetch results

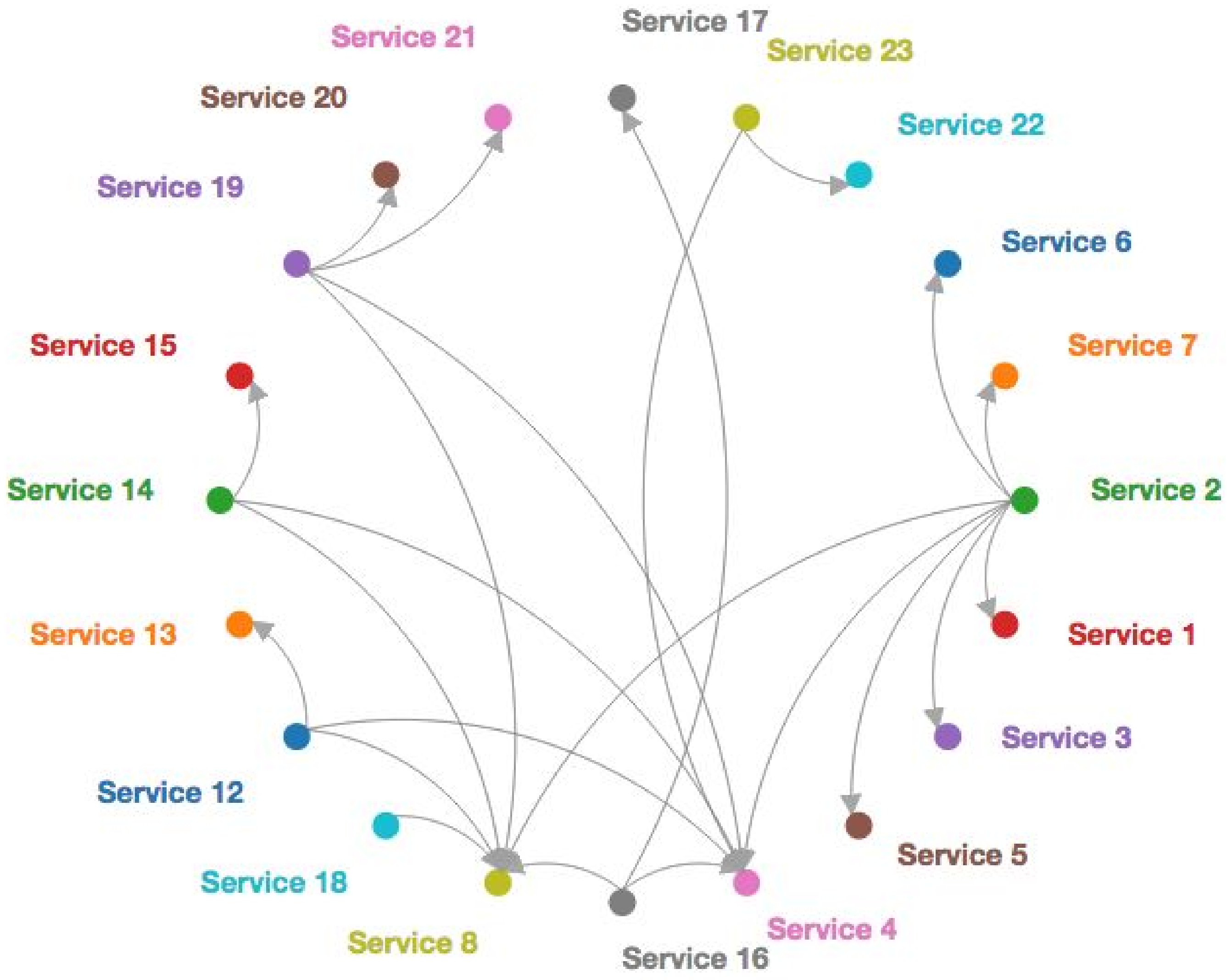
If you can’t deploy your services independently,
you don’t have microservices.
You have a distributed monolith.

“The Matrix”

Consumer version	Provider version	Verification result
11	31	success
12	31	failure
12	32	success
13	32	success

“The Matrix”

Consumer version	**Provider version**	**Verification result**
11	31	success
12	31	failure
12	**32 (prod)**	success
13	**32 (prod)**	success

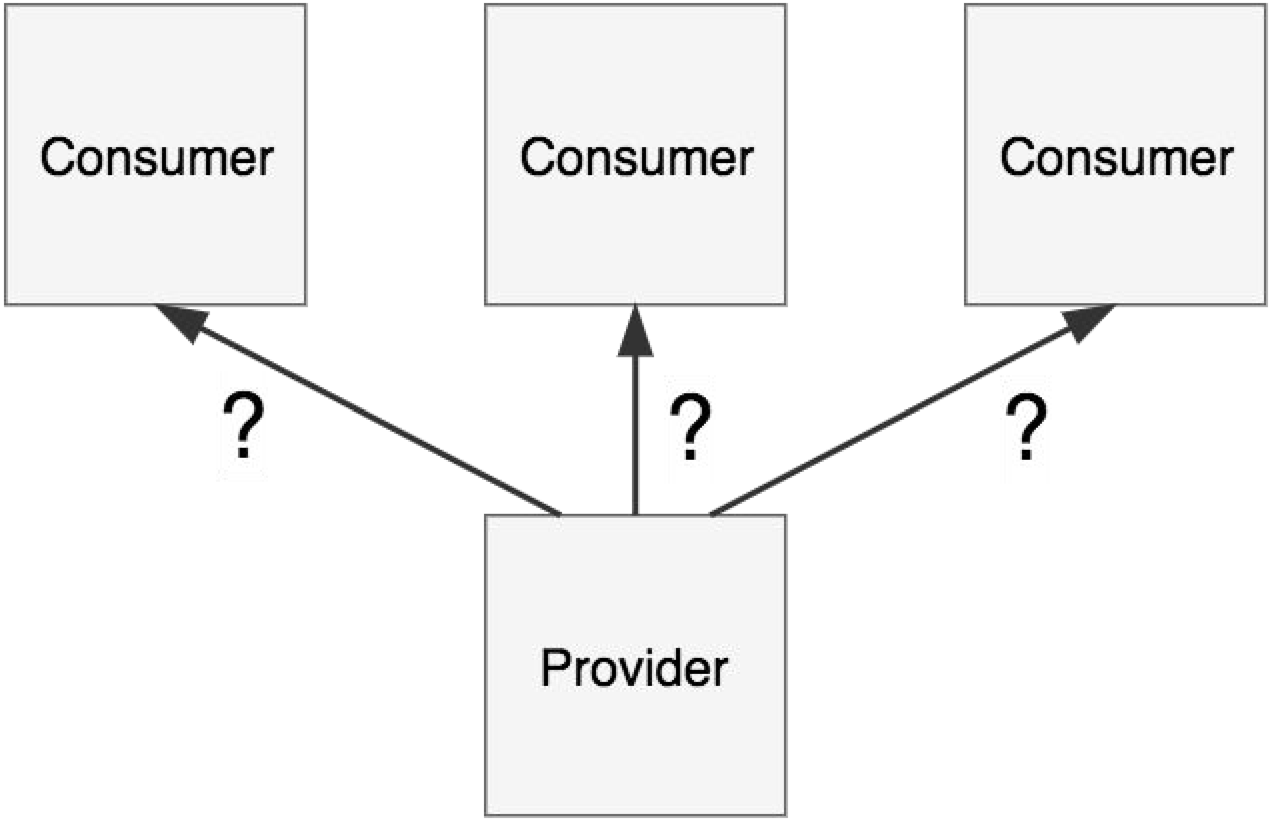

Service 21
Service 17
Service 23
Service 20
Service 22
Service 19
Service 6
Service 15
Service 7
Service 14
Service 2
Service 13
Service 1
Service 3
Service 12
Service 5
Service 18
Service 4
Service 8
Service 16

Spring cloud contracts

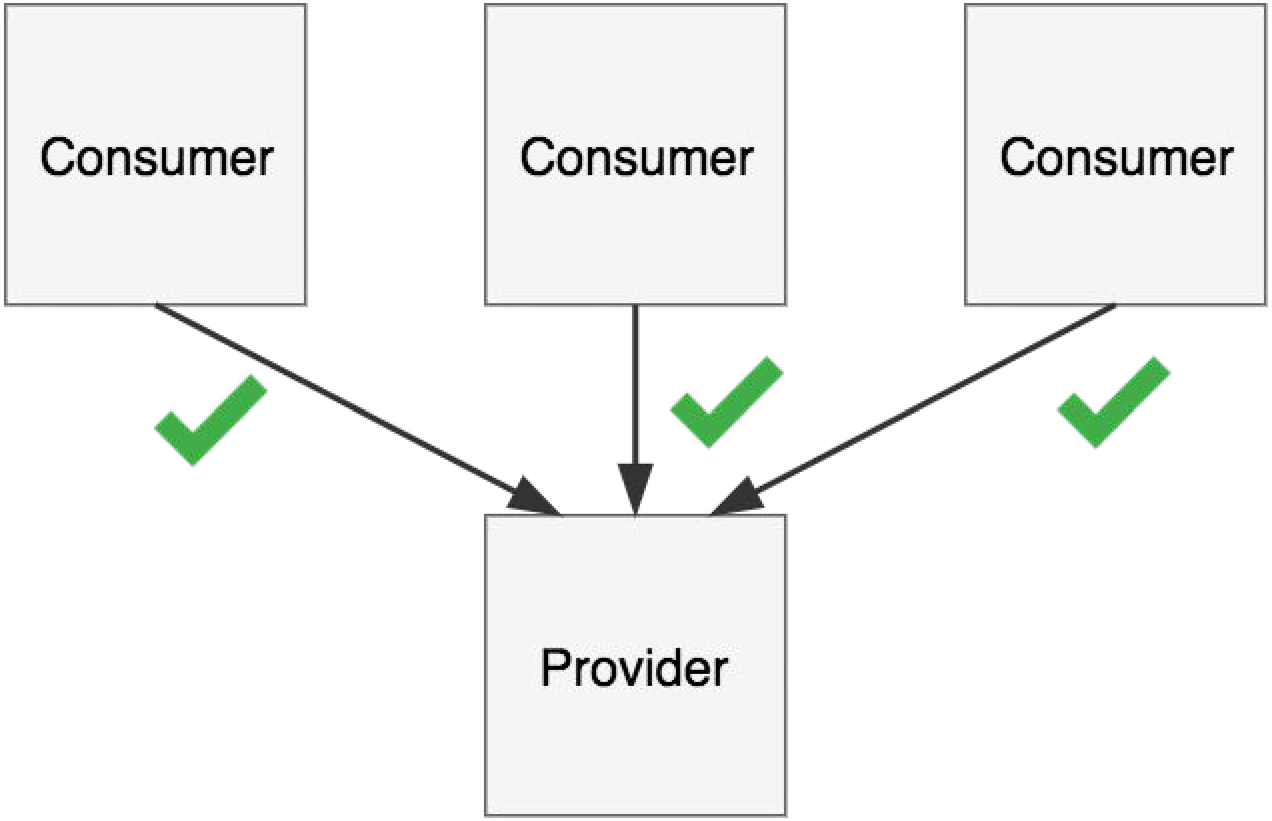
- Designed for JVM but also supports non-JVM

What about Swagger/Open API?

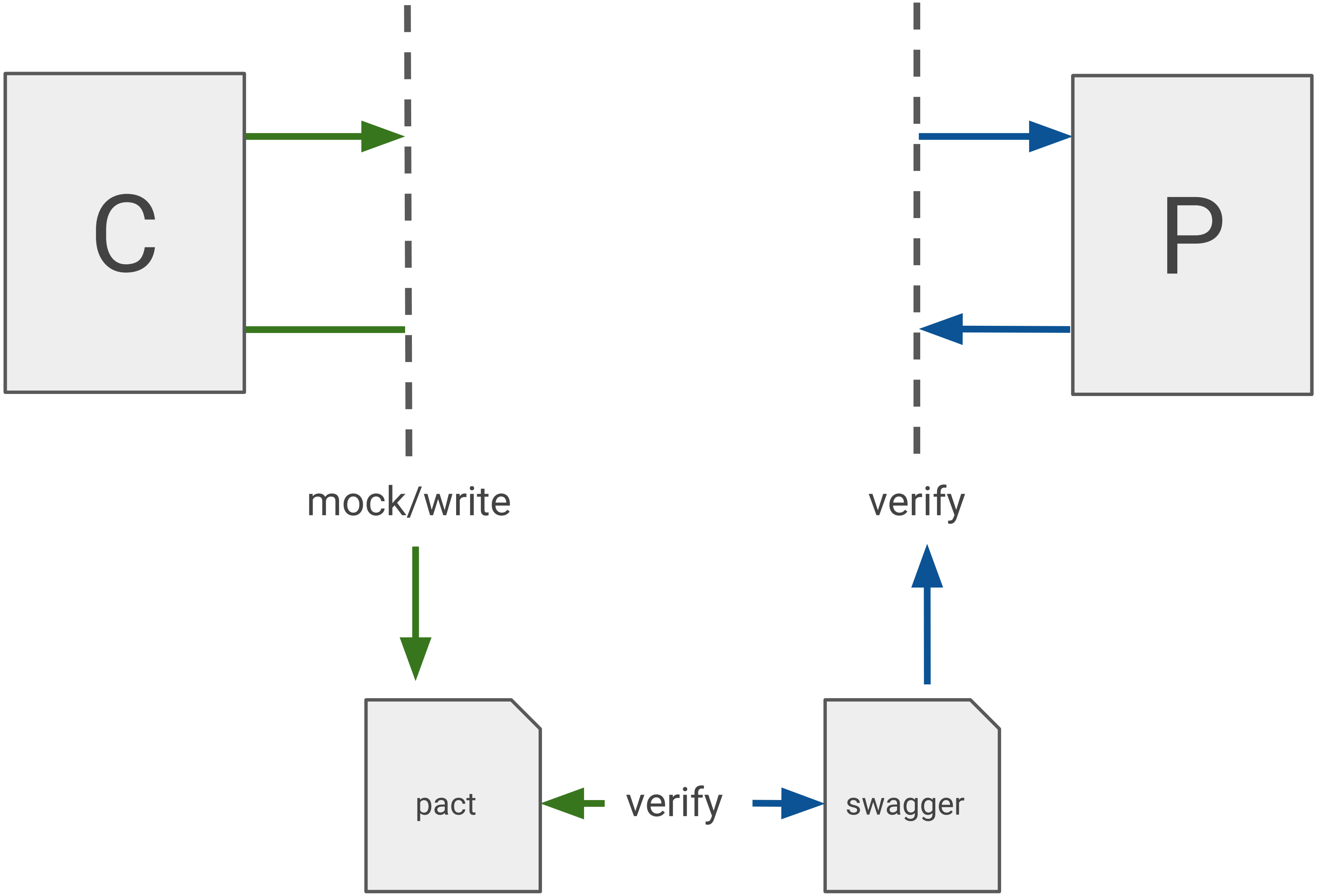
Provider contracts



Consumer contracts



Pact+Swagger



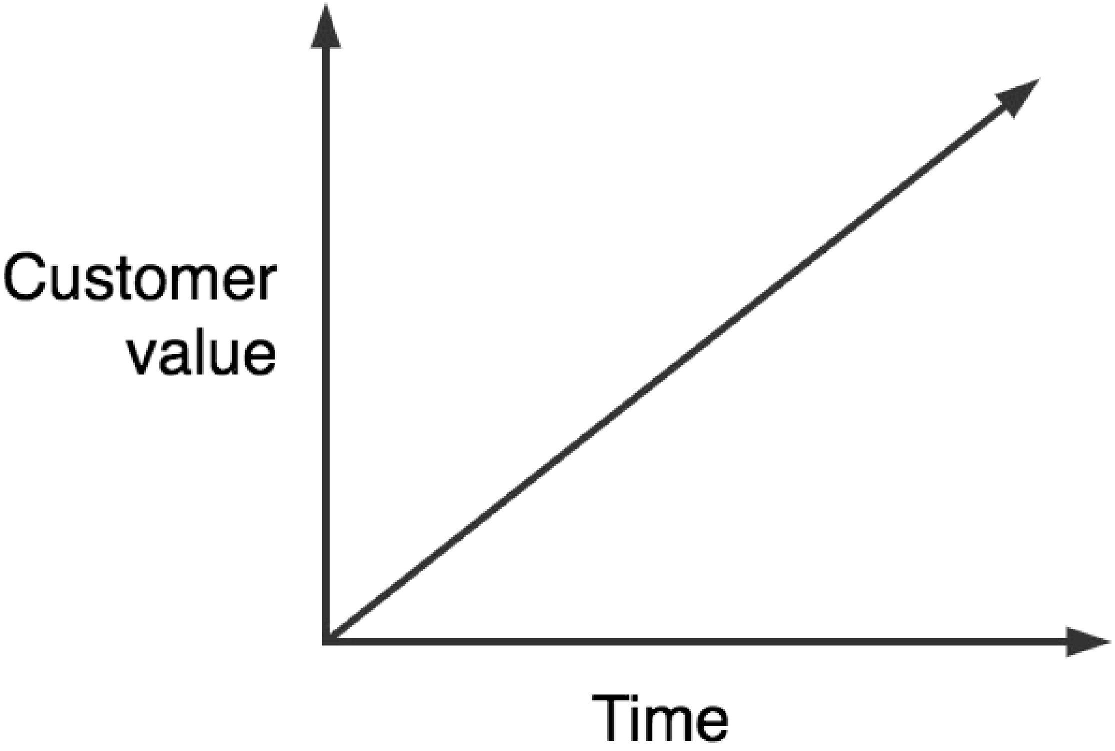
Minimum Viable Contracts

- Consumer
  - mock
  - Serialize
- Provider
  - replay
  - check

Soundify

- Fast, comprehensive contract tests
- Smoke tests
- Semantic monitoring in production
- Alerts
- Fast bug turnaround

Actual customer value over time



Success

- Customers □
- Business owner □
- Delivery manager □
- Teams □

Contracts

Solved problems

- Deliver customer value faster

New problems

- ???

Test SMART not HARD
and deliver customer value fast

dius.tech/contrca4a
pact.io
@pact_up

Beth Skurrie (DiUS)
@bethesque

Copyright