

How to Test Between Microservices Using Contract Testing

Daniel Ward





daninacan.com

Repo: <https://tinyurl.com/9wevymn8>



Me



- Software developer, consultant
- Microsoft MVP .NET
- Co-organizer San Antonio/Austin .NET User Group
-  daninacan.com
-  @danielwarddev
-  daniel-ward-dev
-  danielwarddev.bsky.social



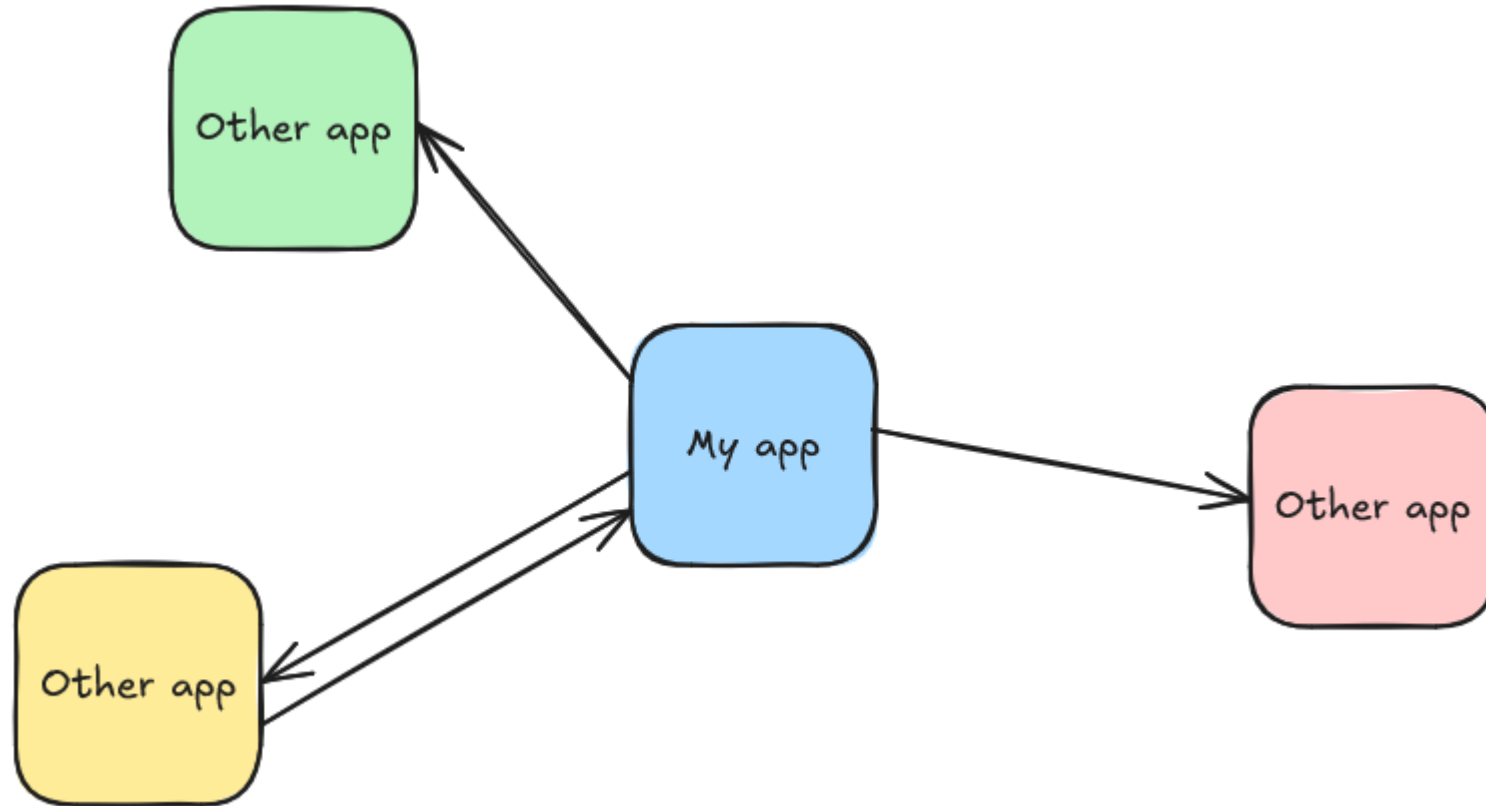
Outline

- Motivation
- What is contract testing?
- Why contract testing?
- Pact
- Coding demo
- Recap + resources

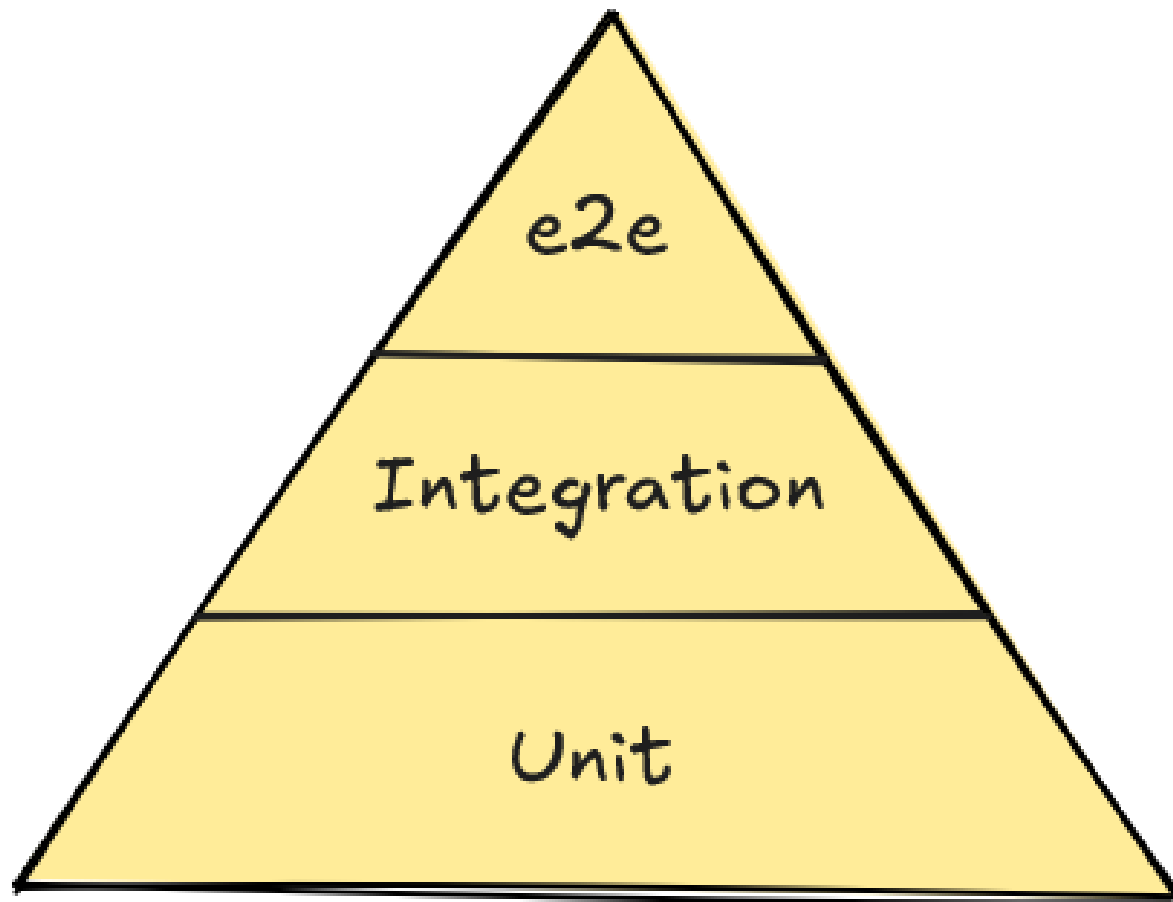
Motivation

- How many times have you [Skip]ped a test?

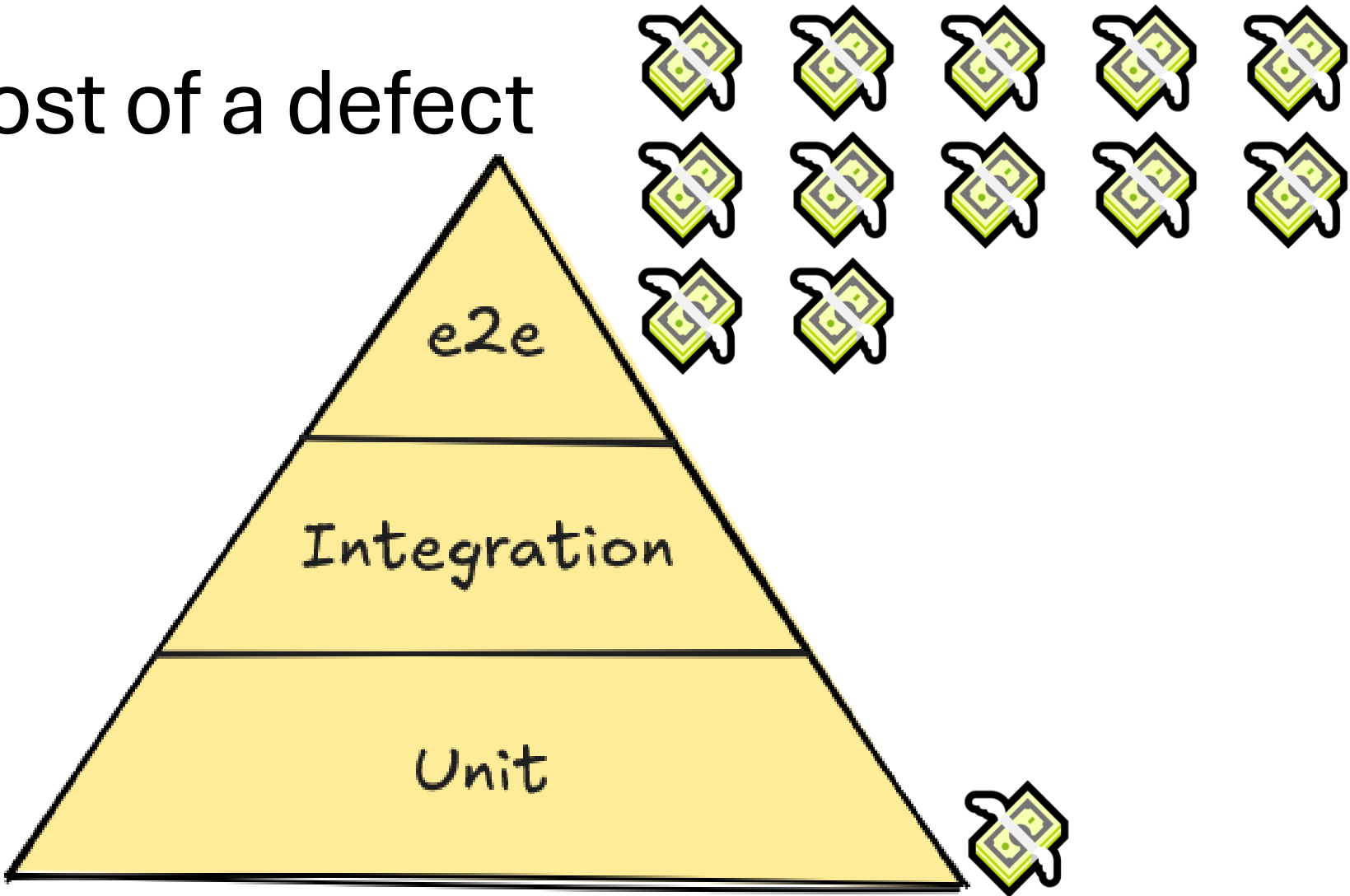
Microservices



Testing



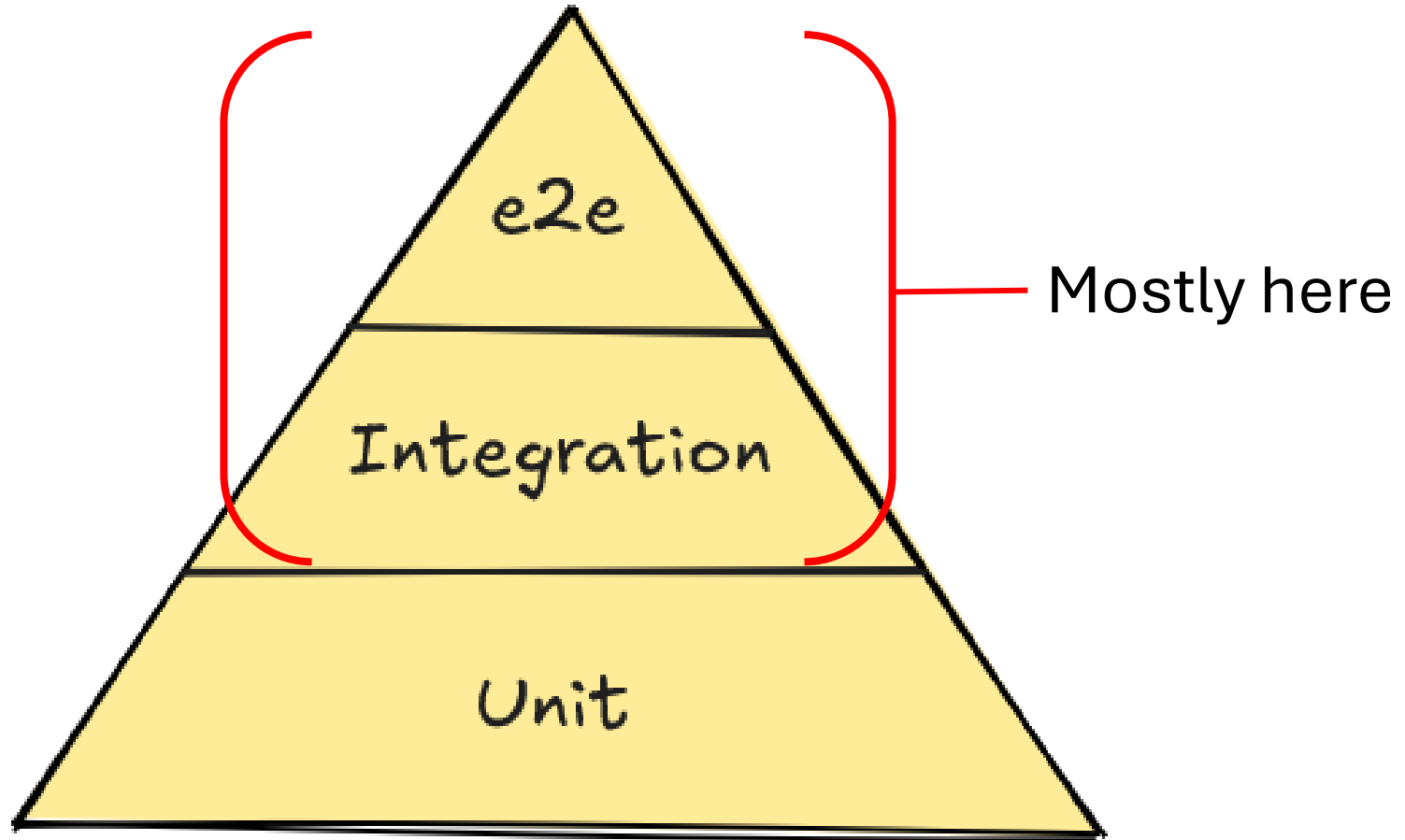
Testing - cost of a defect



Pros/cons

- Pros
 - It works!
- Cons
 - Slow feedback
 - More state to manage
 - If an external service breaks, so does your build
 - The more that teams create services, the more e2e tests are needed

More services, more tests



Why do we test?

- To have confidence in our system

Can we get the best of both worlds?

1. Confidence that our services work between boundaries
2. Quick, reliable tests (mock external calls)

Contract testing

- Tests that produce a contract (JSON file) describing the expectations of an API
- Services communicate through each others' interfaces
 - REST
 - gRPC
 - Distributed – queues, pub/sub, etc.
 - More
- Test with the interfaces instead of the real services

Do you set
your house
on fire to test
your smoke
alarm?



History

Contract Test

12 January 2011



Martin Fowler

TEST CATEGORIE

One of the most common cases of using a TestDouble is when you are communicating with an external service. Typically such services are being maintained by a different team, they may be subject to slow, and unreliable networks, and maybe unreliable themselves. That's why a test double is handy, it stops your own tests from being slow and unreliable. But testing against a double always raises the question of whether the double is indeed an accurate representation of the external service, and what happens if the external service changes its contract?



<https://martinfowler.com/bliki/ContractTest.html>

Consumer-Driven Contracts: A Service Evolution Pattern

This article discusses some of the challenges in evolving a community of service providers and consumers. It describes some of the coupling issues that arise when service providers change parts of their contract, particularly document schemas, and identifies two well-understood strategies - adding schema extension points and performing "just enough" validation of received messages - for mitigating such issues. Both strategies help protect consumers from changes to a provider contract, but neither of them gives the provider any insight into the ways it is being used and the obligations it must maintain as it evolves. Drawing on the assertion-based language of one of these mitigation strategies - the "just enough" validation strategy - the article then describes the "Consumer-Driven Contract" pattern, which imbues providers with insight into their consumer obligations, and focuses service evolution around the delivery of the key business functionality demanded by consumers.

12 June 2006



Ian Robinson

CONTENTS

- [Evolving a Service: An Example](#)
- [Interlude: Burdened With Services](#)
- [Schema Versioning](#)
 - [Extension Points](#)
 - [Breaking Changes](#)

<https://martinfowler.com/articles/consumerDrivenContracts.html>

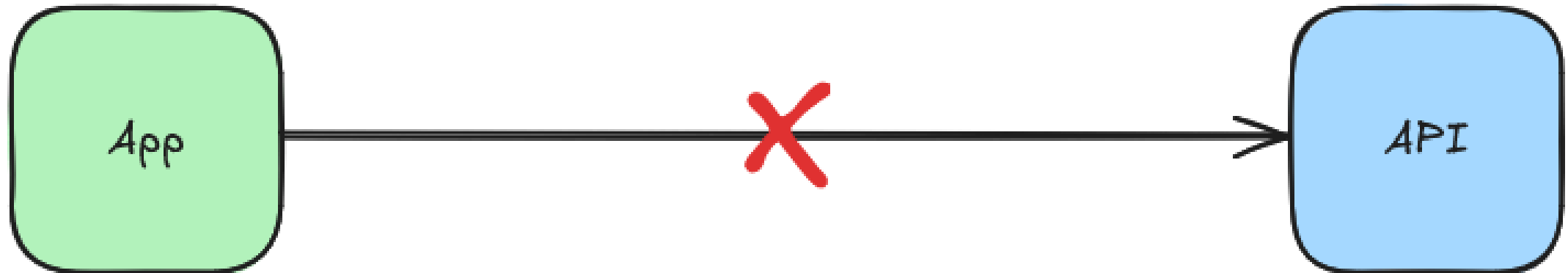
Pact

- Library for writing contract tests
- Javascript, Java, .NET, Go, Python, Swift, Scala, PHP, Ruby, Rust, C++
- Backend written in Rust

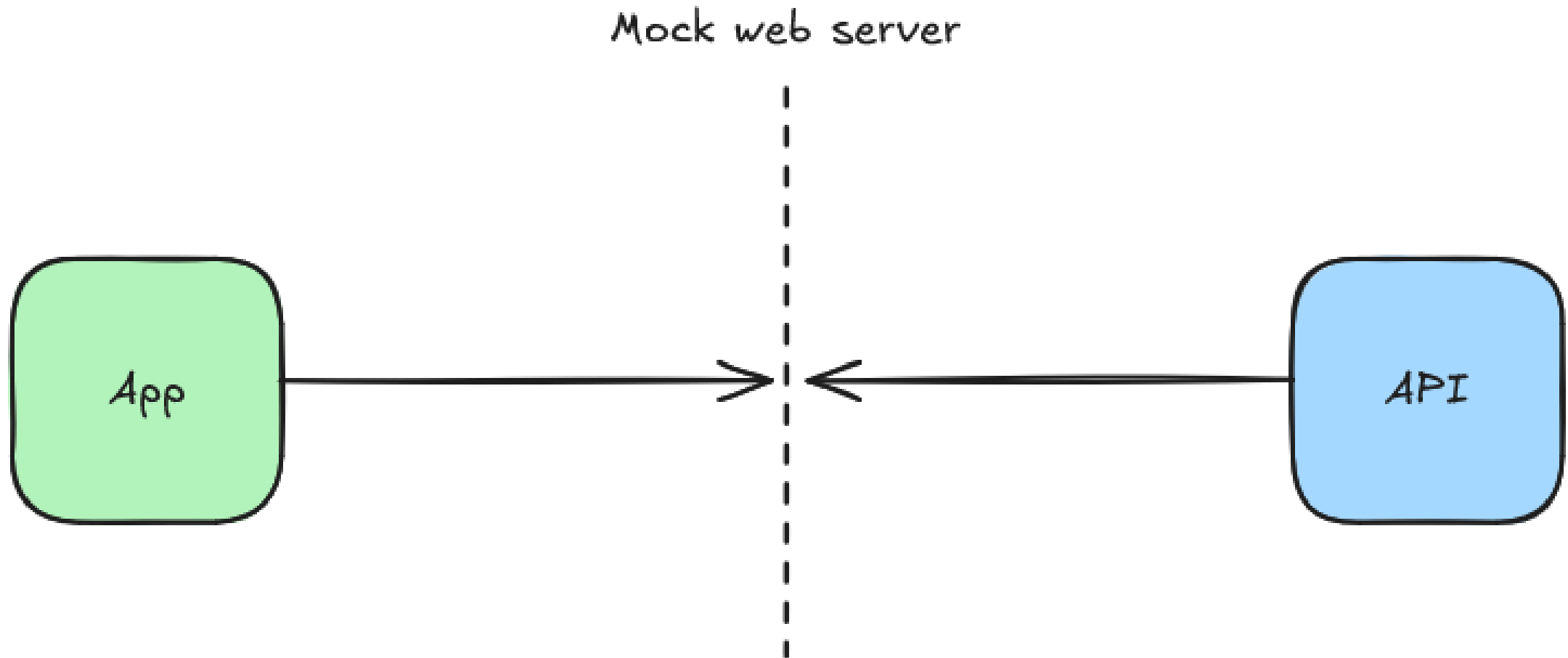
Contract testing



Contract testing



Contract testing



Contract testing

```
GET /product/1  
If a product with id 1 exists  
Return 200 with this body: {...}  
...
```



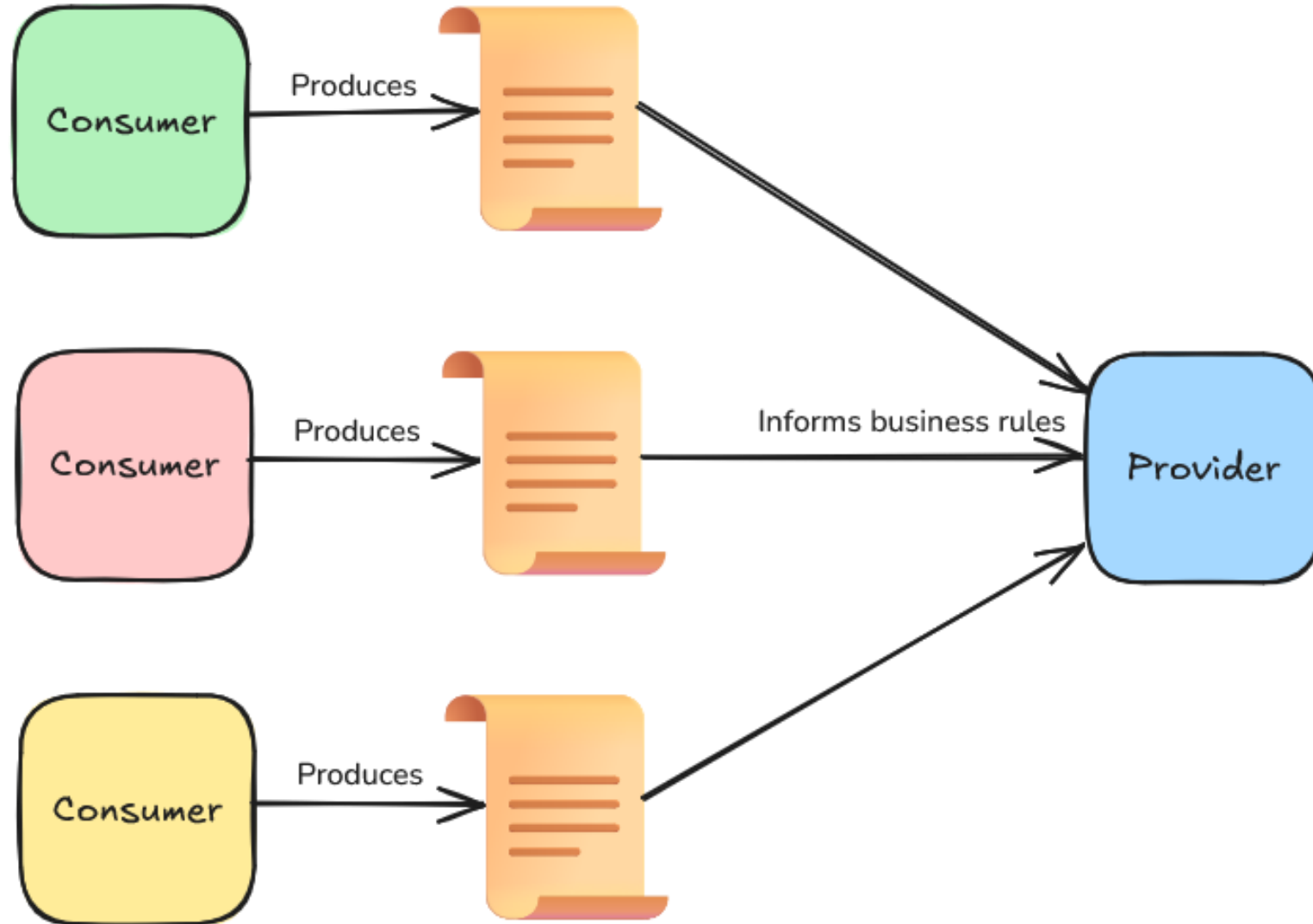
Consumer-driven contracts (CDC)

- Ask the customer first!
- The consumer sends what it wants before the provider implements it
- Customer-driven

CDC 2 main benefits

1. API design is coupled to business value (leaner contract, better aligned with the business goals)
2. Gives feedback into how consumers will be affected (prevents breaking changes)

Consumer-driven design



What about OpenAPI?

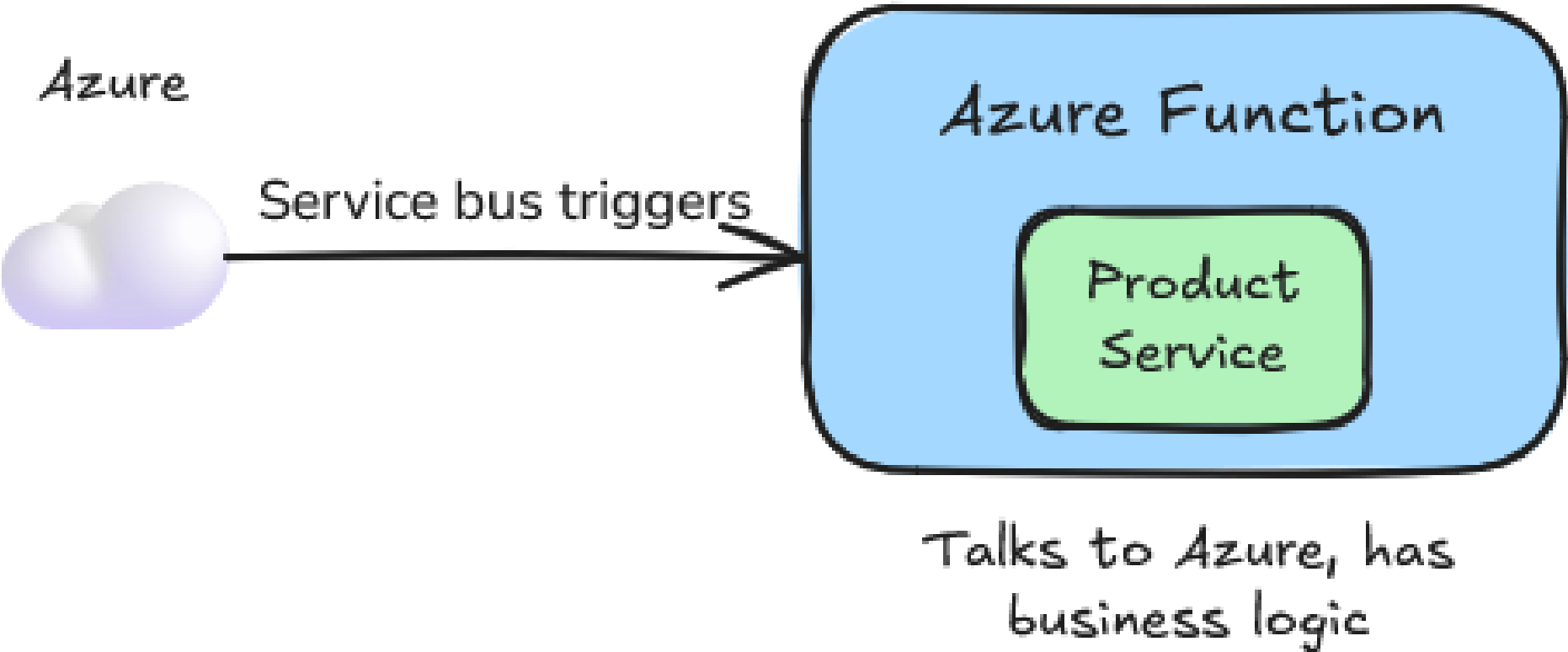
- Close!
- Has all the requests and their responses
- Does not have what causes each response to happen

```
▼ /product/{productId}:
  ▼ get:
    ▼ tags:
      0: "Product"
    ▼ parameters:
      ▼ 0:
        name: "productId"
        in: "path"
        required: true
        ▼ schema:
          type: "integer"
          format: "int32"
    ▼ responses:
      ▼ 200:
        description: "OK"
        ▼ content:
          ▼ text/plain:
            ▶ schema: { "$ref": "#/components/schemas/ProductDto" }
          ▼ application/json:
            ▶ schema: { "$ref": "#/components/schemas/ProductDto" }
          ▼ text/json:
            ▶ schema: { "$ref": "#/components/schemas/ProductDto" }
      ▼ 404:
        description: "Not Found"
        ▼ content:
          ▼ text/plain:
            ▶ schema: { "$ref": "#/components/schemas/ProblemDetails" }
          ▼ application/json:
            ▶ schema: { "$ref": "#/components/schemas/ProblemDetails" }
          ▼ text/json:
            ▶ schema: { "$ref": "#/components/schemas/ProblemDetails" }
```

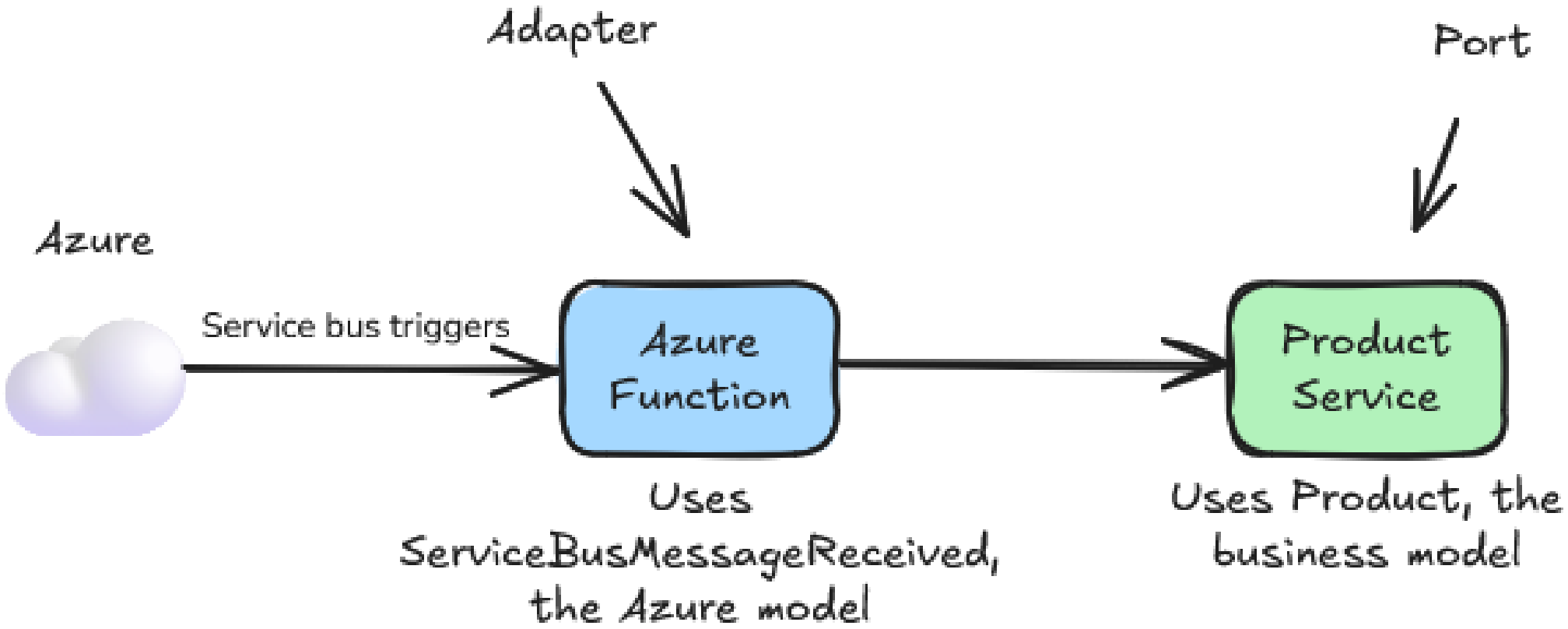
Contract testing examples

- HTTP contract
- Messaging contracts
- Pact broker

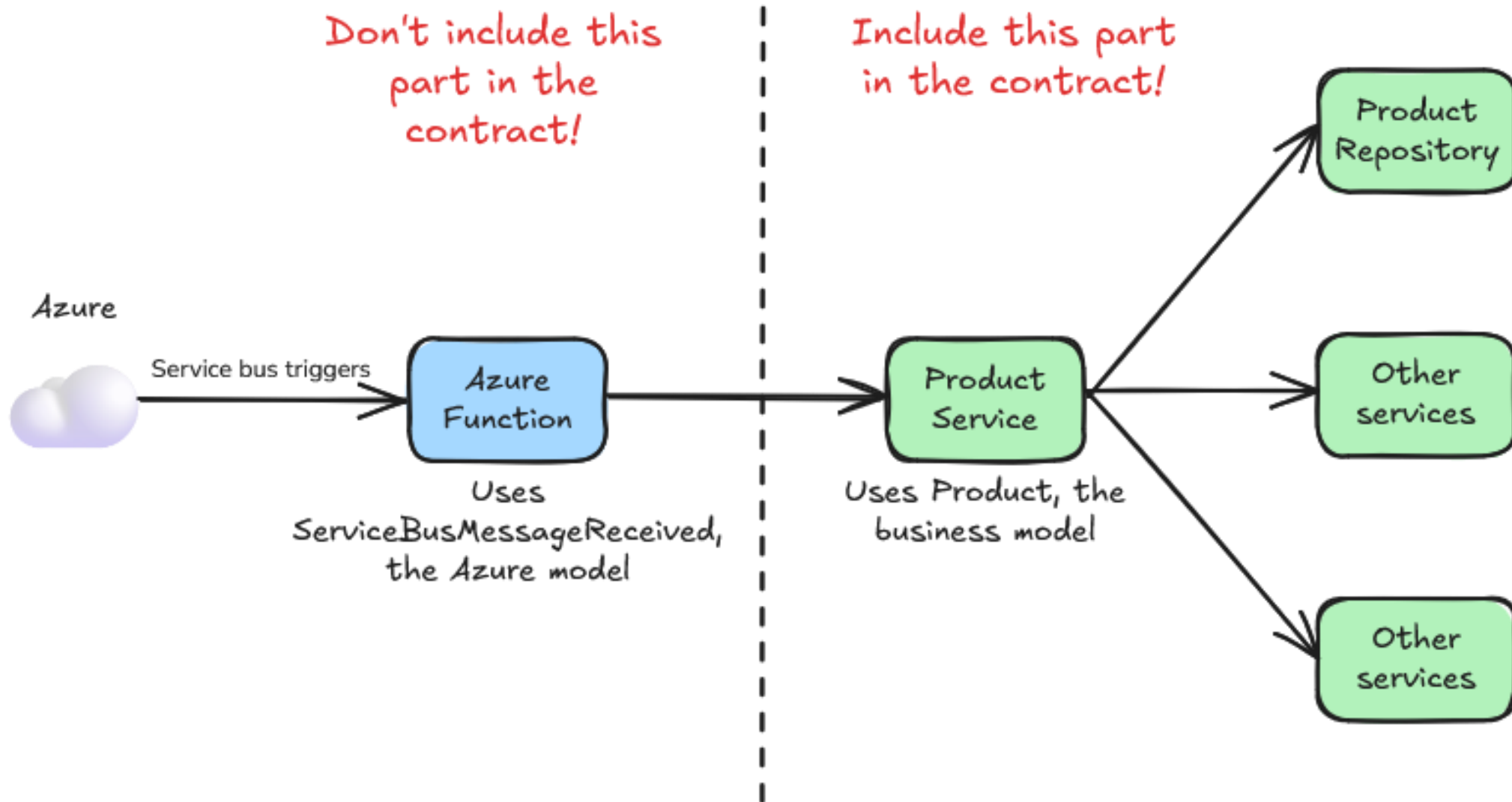
Adapter pattern



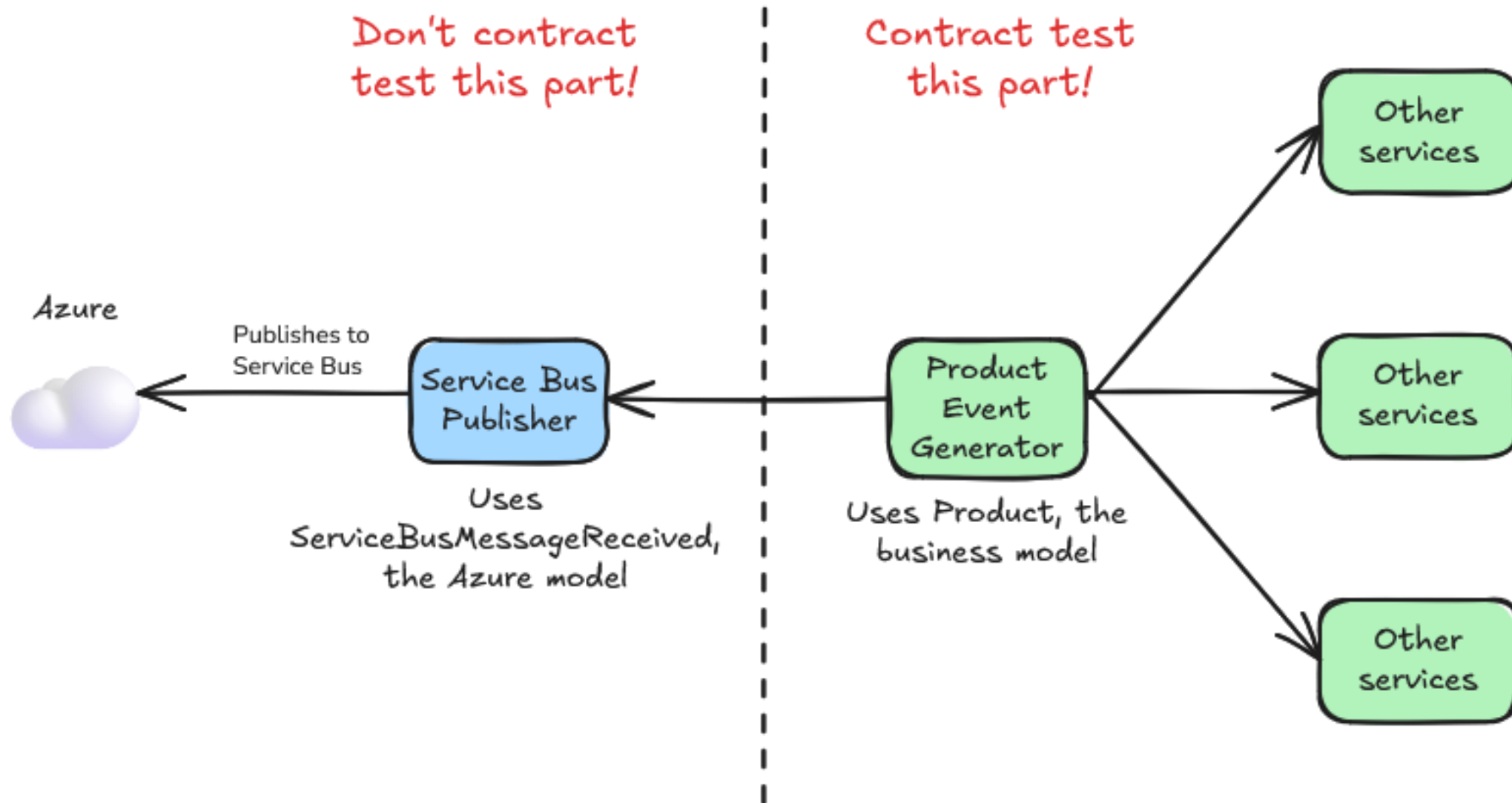
Adapter pattern



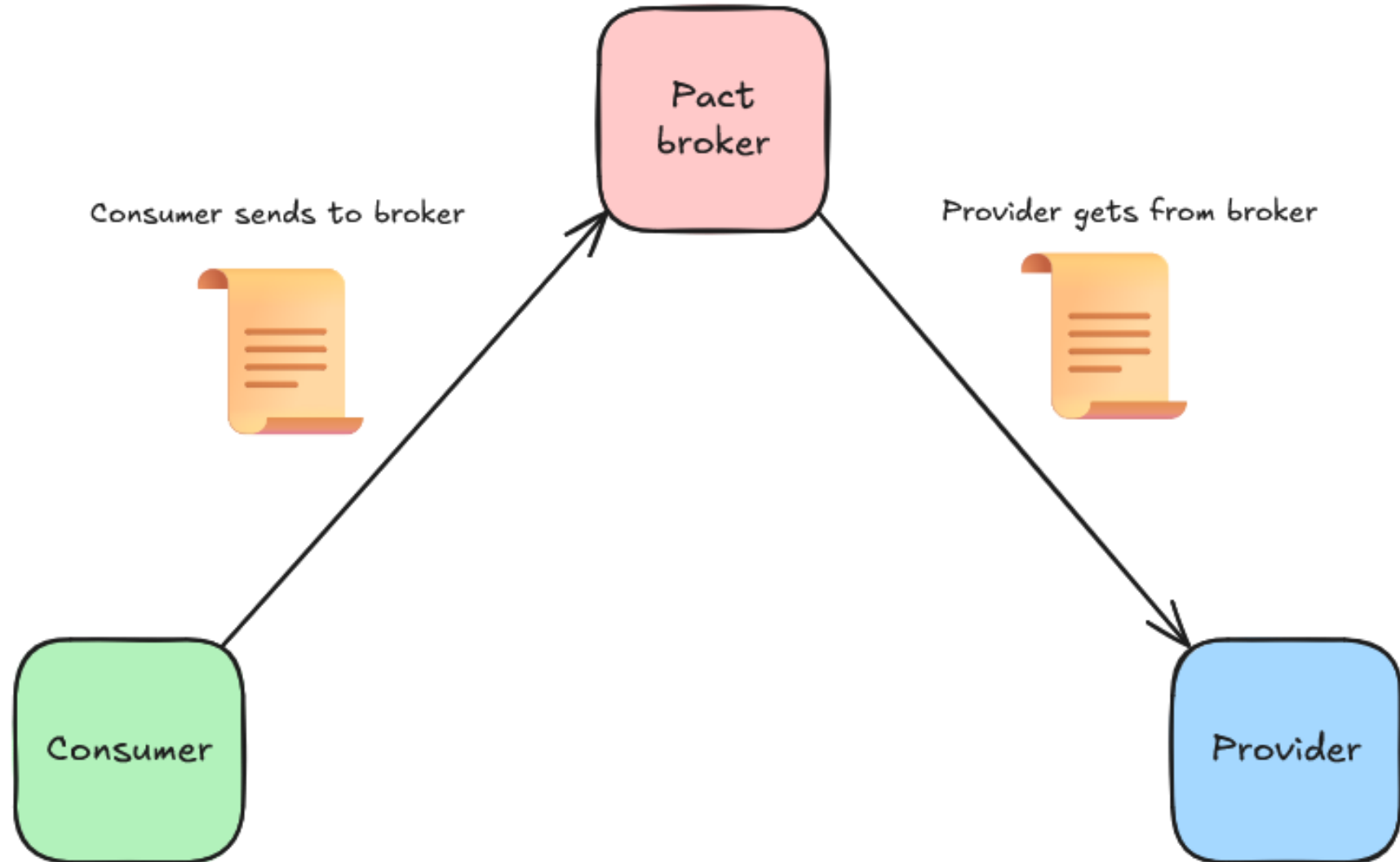
Message contracts - Consumer



Message contracts - Provider



Pact Broker



When to NOT use contract testing

- Public APIs with hundreds, thousands, millions of consumers
- Pass-through APIs like BFFs
- Authorization

How can I convince people at my work?

- Probably easier to do if you're the providing team
- Don't try to implement all the tools at once
- Wait until a problem arises that it can fix

More stuff

- Pact's bronze-diamond tier guide https://docs.pact.io/pact_nirvana
- <https://martinfowler.com/bliki/ContractTest.html>
- <https://martinfowler.com/articles/consumerDrivenContracts.html>
- My series <https://daninacan.com/what-contract-testing-is-and-why-to-write-them/>

Recap

- Motivation
- What is contract testing?
- Why contract testing?
- How to contract test with Pact
- When to not use contract testing
- Recap + resources

Thank you!

Daniel Ward

daninacan.com

Repo: <https://tinyurl.com/9wevymn8>

