

Let's make a Pact

Don't break my API!

I'm Frank Kilcommins

- › API Technical Evangelist at SmartBear
- › Developer and Architect passionate about APIs & Developer Experience

`/* insert embarrassing photo here */`

› Connect:



@fkilcommins



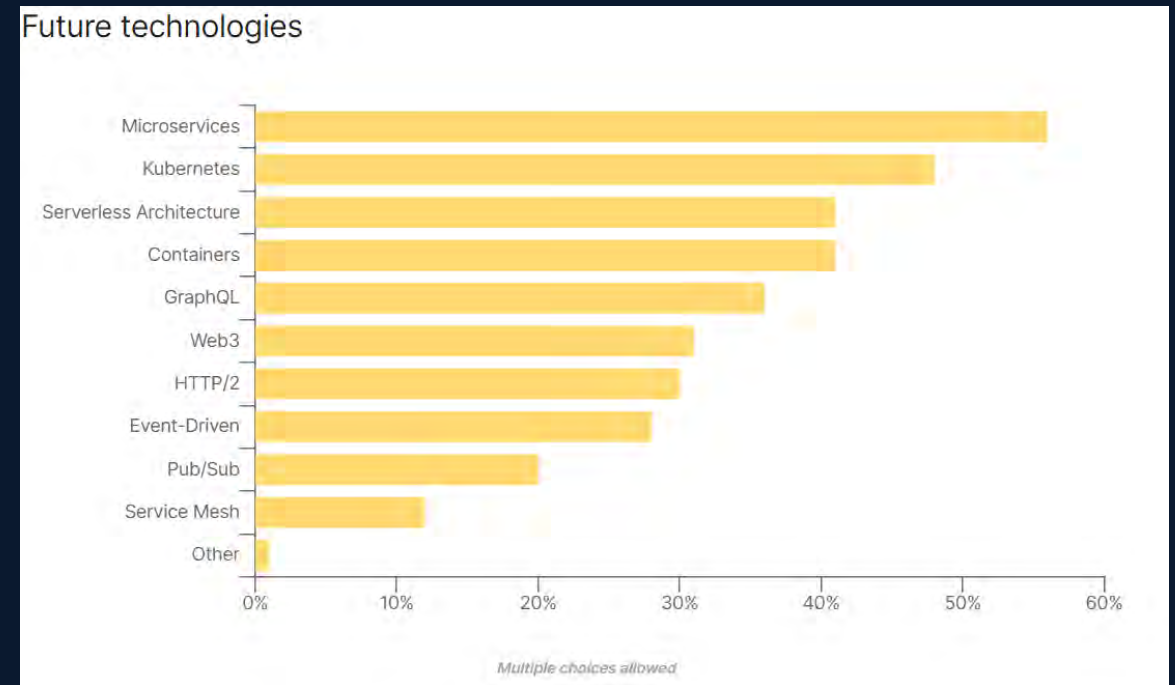
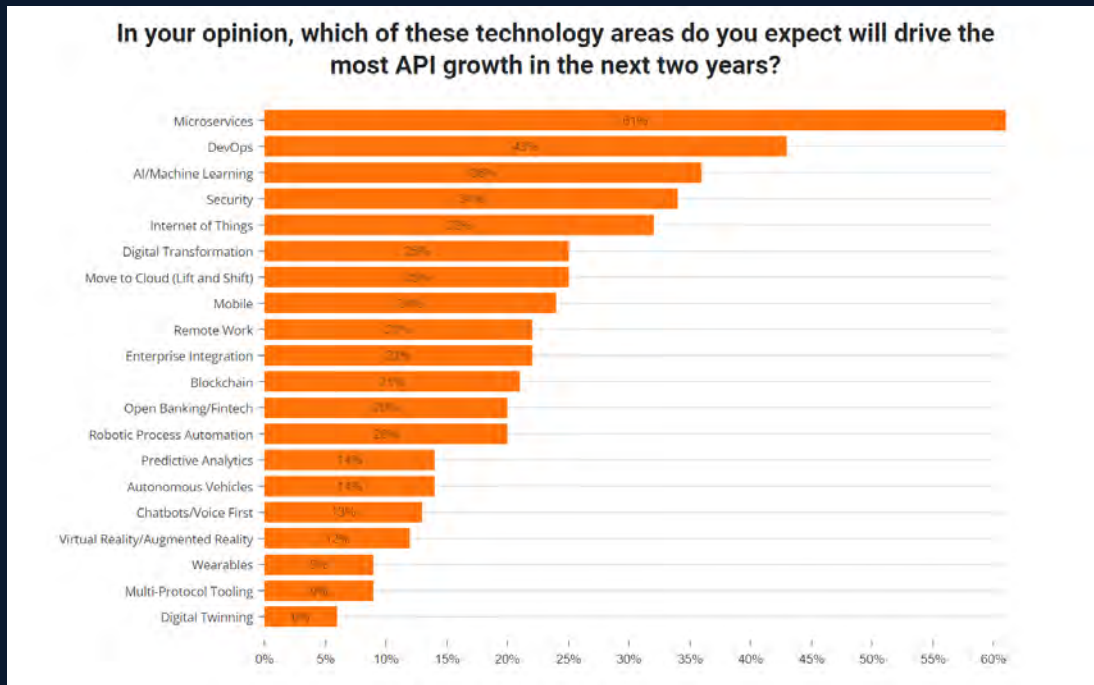
@frank-kilcommins

Talk Track

- › API Landscape Trends
- › Designing for the future
- › Is extensibility enough?
- › Bi-Directional Contract Testing – An approach to calming the chaos
- › Demo
- › Takeaways

API Landscape Trends

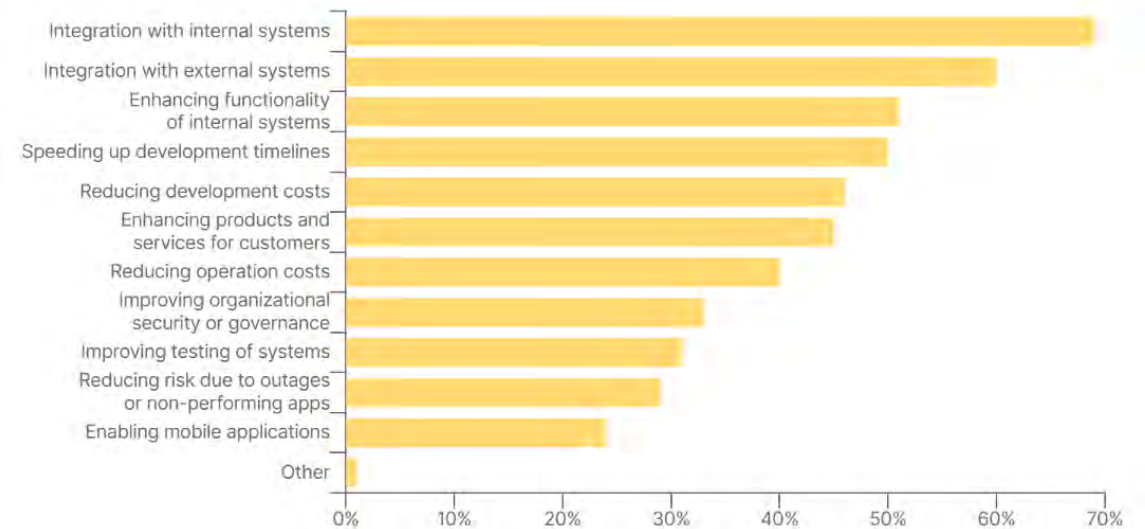
› Microservices driving API growth



API Landscape Trends

- › Microservices driving API growth
- › Microservices are more than a *fad*

Consuming APIs: internal integration is key



Multiple choices allowed

API Landscape Trends

- › Microservices driving API growth
- › Microservices are more than a *fad*
- › *Managing the sprawl* will get harder

“By 2025, less than 50% of enterprise APIs
will be managed, as explosive growth in
APIs surpasses the capabilities of API
management tools.”

Gartner

Designing for Extensibility....helps!

- › A successful API is long living and can evolve gracefully
- › Bake extensibility into your design practices

Designing for Extensibility....helps!

- › A successful API is long living and can evolve gracefully
- › Bake extensibility into your design practices

Do

- › Treat your microservices as APIs (and APIs as Products)
- › Define your extension points
- › Communicate robust extensibility pattern
- › Apply semantic versioning
- › Test for extensibility
- › Communicate

Designing for Extensibility....helps!

- › A successful API is long living and can evolve gracefully
- › Bake extensibility into your design practices

Do

- › Treat your microservices as APIs (and APIs as Products)
- › Define your extension points
- › Communicate robust extensibility pattern
- › Apply semantic versioning
- › Test for extensibility
- › Communicate

Don't

- › Don't add required inputs
- › Don't remove outputs or make them optional
- › Don't change the type of a property
- › Don't change property meaning by adding new property
- › Use Booleans sparingly
- › Be inconsistent in your process



Failure warning: extensibility alone is not enough

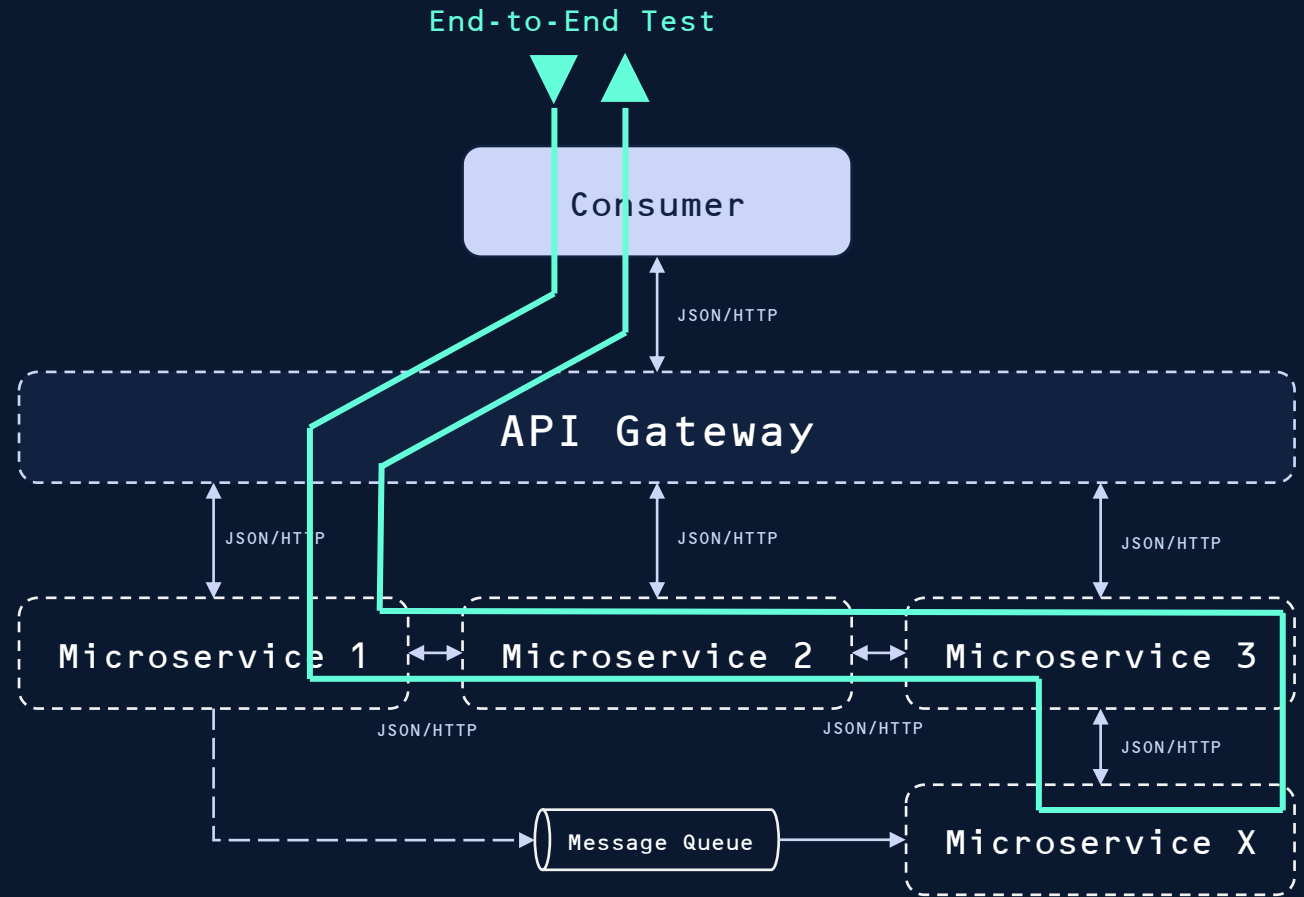
Pitfalls remain ☹️

- › Major version proliferation



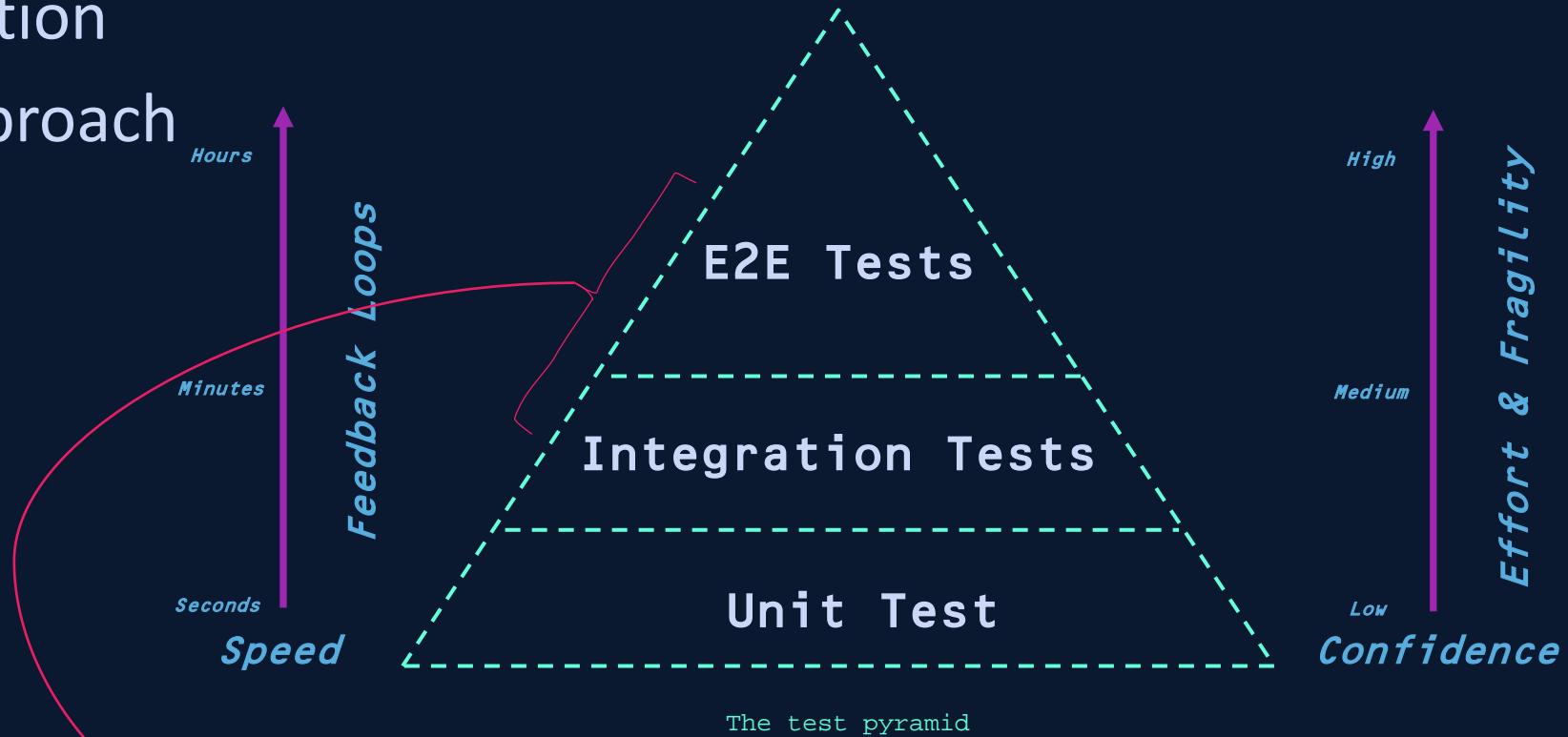
Pitfalls remain ☹️

- › Major version proliferation
- › Unbalanced testing approach



Pitfalls remain ☹️

- › Major version proliferation
- › Unbalanced testing approach

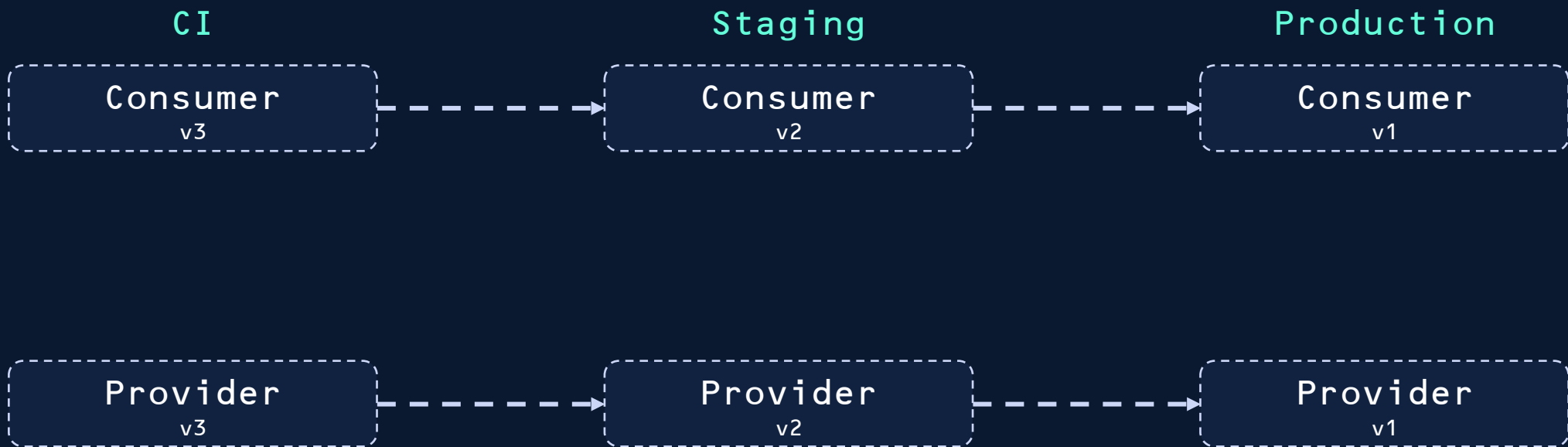


- ✗ Expensive
- ✗ Slow
- ✗ Unreliable
- ✗ Not targeted

```
/* ToDo: address the complexity */
```

Pitfalls remain ☹️

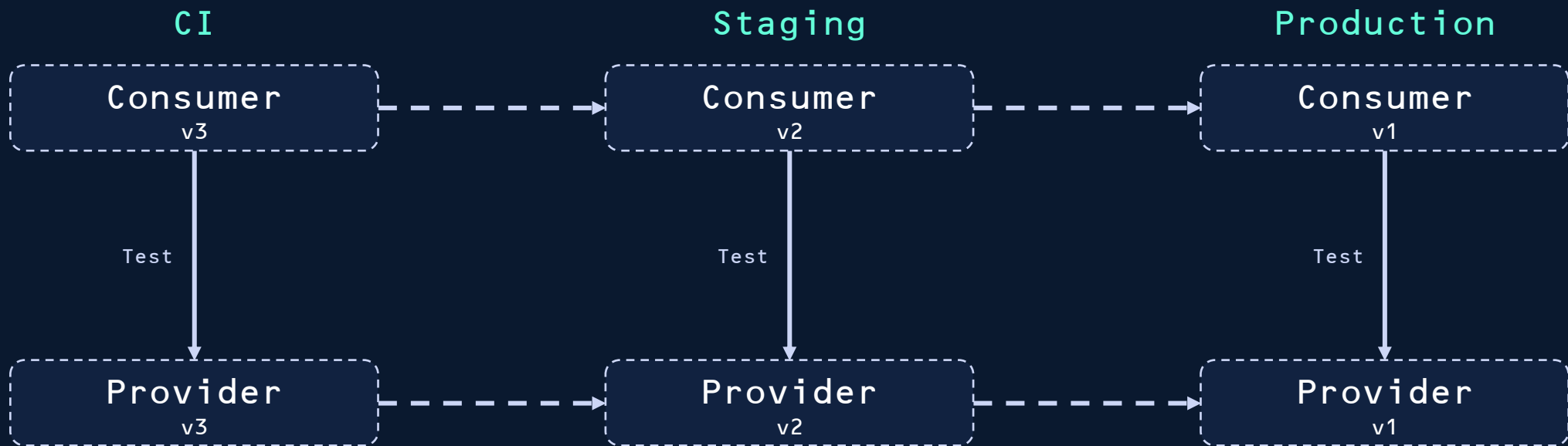
- › Major version proliferation
- › Unbalanced testing approach
- › Environment management (dependencies)



```
/* ToDo: address the complexity */
```

Pitfalls remain 😞

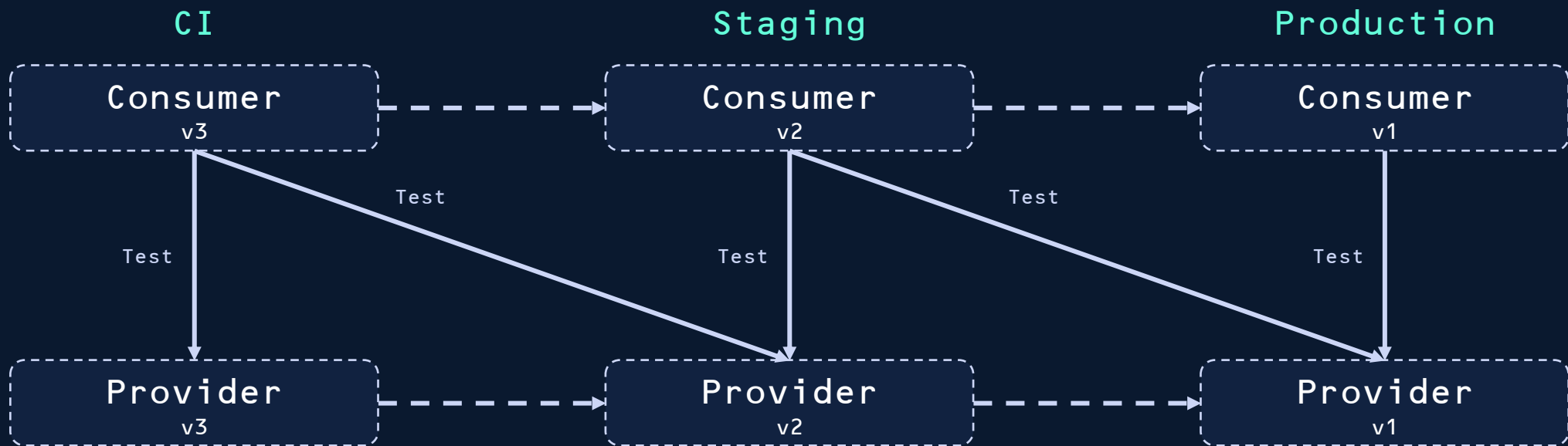
- › Major version proliferation
- › Unbalanced testing approach
- › Environment management (dependencies)



```
/* ToDo: address the complexity */
```

Pitfalls remain ☹️

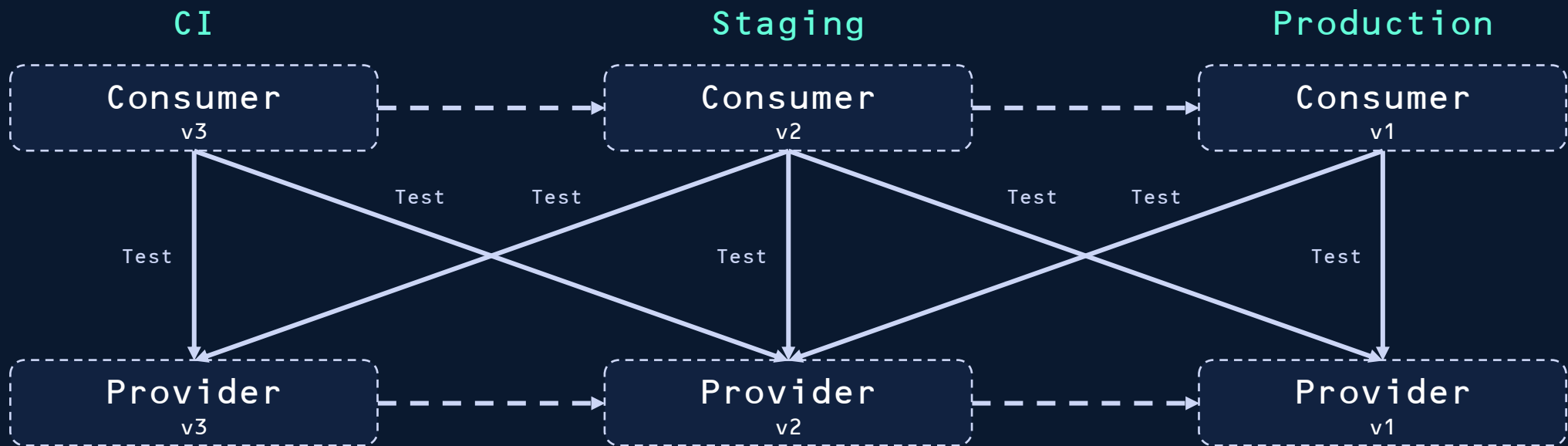
- › Major version proliferation
- › Unbalanced testing approach
- › Environment management (dependencies)




```
/* ToDo: address the complexity */
```

Pitfalls remain ☹️

- › Major version proliferation
- › Unbalanced testing approach
- › Environment management (dependencies)



*“If you can’t deploy services
independently, you don’t have
microservices”*

– Beth Skurrie

*“If you can’t deploy services
independently, you don’t have
microservices”
– Beth Skurrie*

You have a distributed monolith

```
/* Insert viable solution */
```

Bi-Directional Contract Testing

Making a pact to evolve safely

```
/* A new approach to contract testing */
```

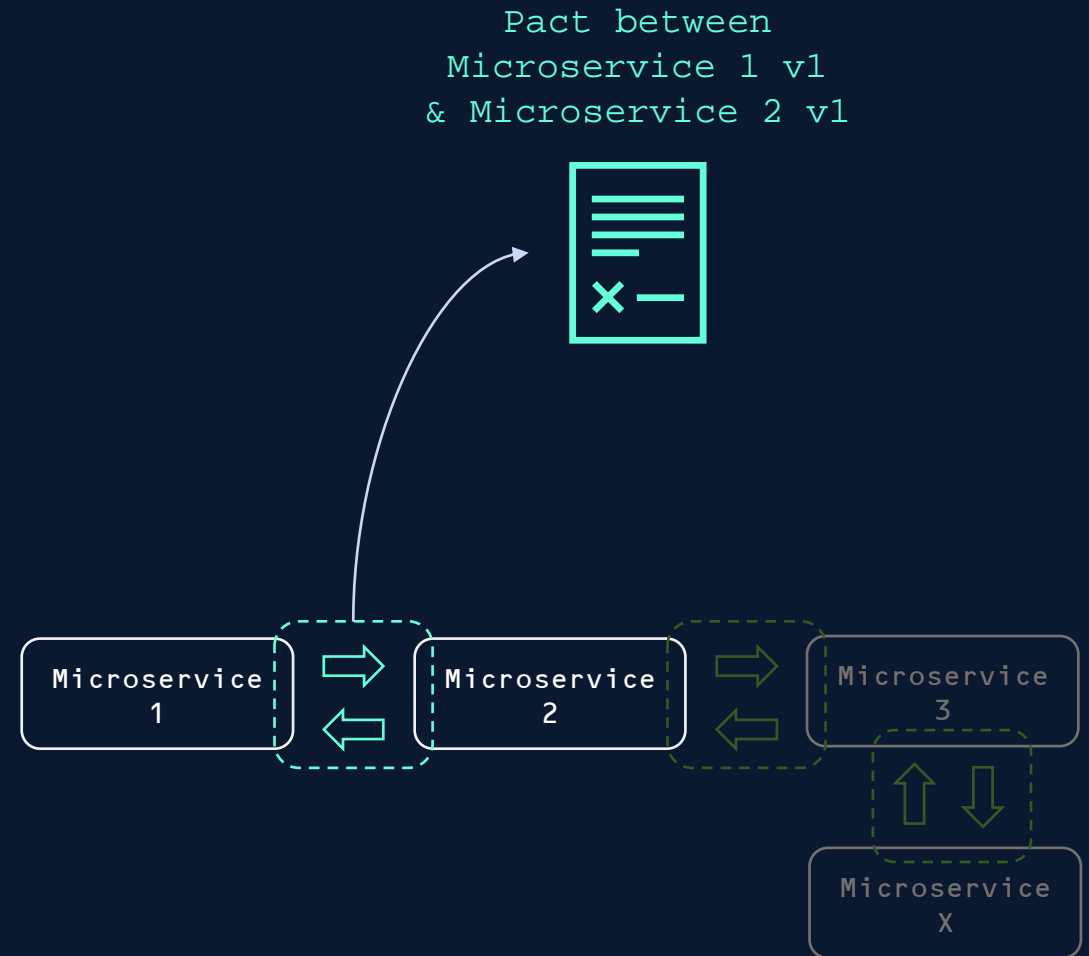
Bi-Directional Contract Testing (BDCT)

- › Schema based rather than specification by example
- › Supports design-first provider workflow
- › Well suited to retrofit onto existing systems
- › BYO tools, tests and artifacts:
 - › **OpenAPI** documents
 - › Capture contracts (e.g, **Cypress, Wiremock, Mountebank**)
 - › Contract verification (e.g., **Dredd, Restassured, ReadyAPI, Postman**)
- › More inclusive support for wider demographic of contract testers (e.g., Designers, QAs, SDETs, Devs)

```
/* Additional Context */
```

What's a Pact

- › Pact (noun): *A formal agreement between individuals or parties*
- › Creates a contract between consumer and provider, which is **independently** verifiable
- › Captures interaction expectations between software components (both explicit and implicit)



```
/* Additional Context */
```

What's a Pact

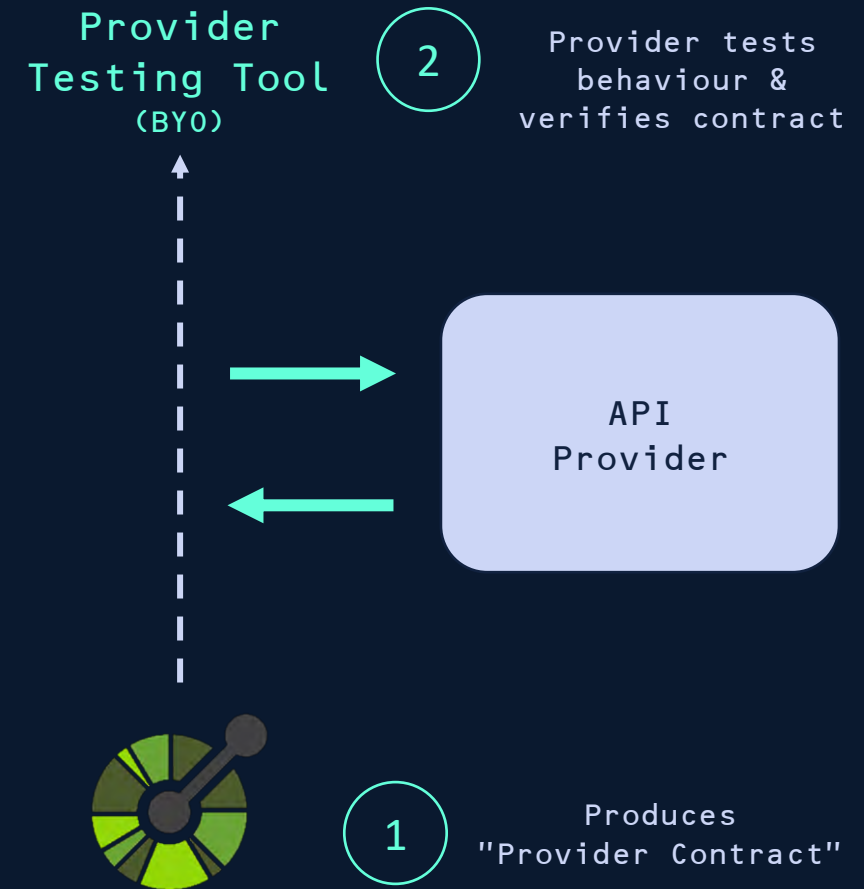
- › Pact (noun): *A formal agreement between individuals or parties*
- › Creates a contract between consumer and provider, which is independently verifiable
- › Captures interaction expectations between software components (both explicit and implicit)
- › Keep assumptions in sync
- › Ability to verify consumer-provider pairs in an asynchronous fashion

Pact.json file example

```
{
  "consumer": { "name": "microservice1-consumer-wiremock" },
  "provider": { "name": "microservice2-provider-restassured" },
  "interactions": [
    {
      "description": "GET_/products_f25f7b8e-35f2",
      "request": {
        "method": "GET",
        "path": "/products",
        "query": "name=pizza&type=food",
        "headers": { "Content-Type": "application/json" }
      },
      "response": {
        "status": 200,
        "headers": { "Content-Type": "application/json" },
        "body": { "id": "27", "name": "pizza", "type": "food" }
      }
    }
  ],
  "metadata": {
    "pactSpecification": { "version": "2.0.0" },
    "client": {
      "name": "optional name of the adapter",
      "version": "semver compatible version of the adapter"
    }
  }
}
```

```
/* walk through */
```

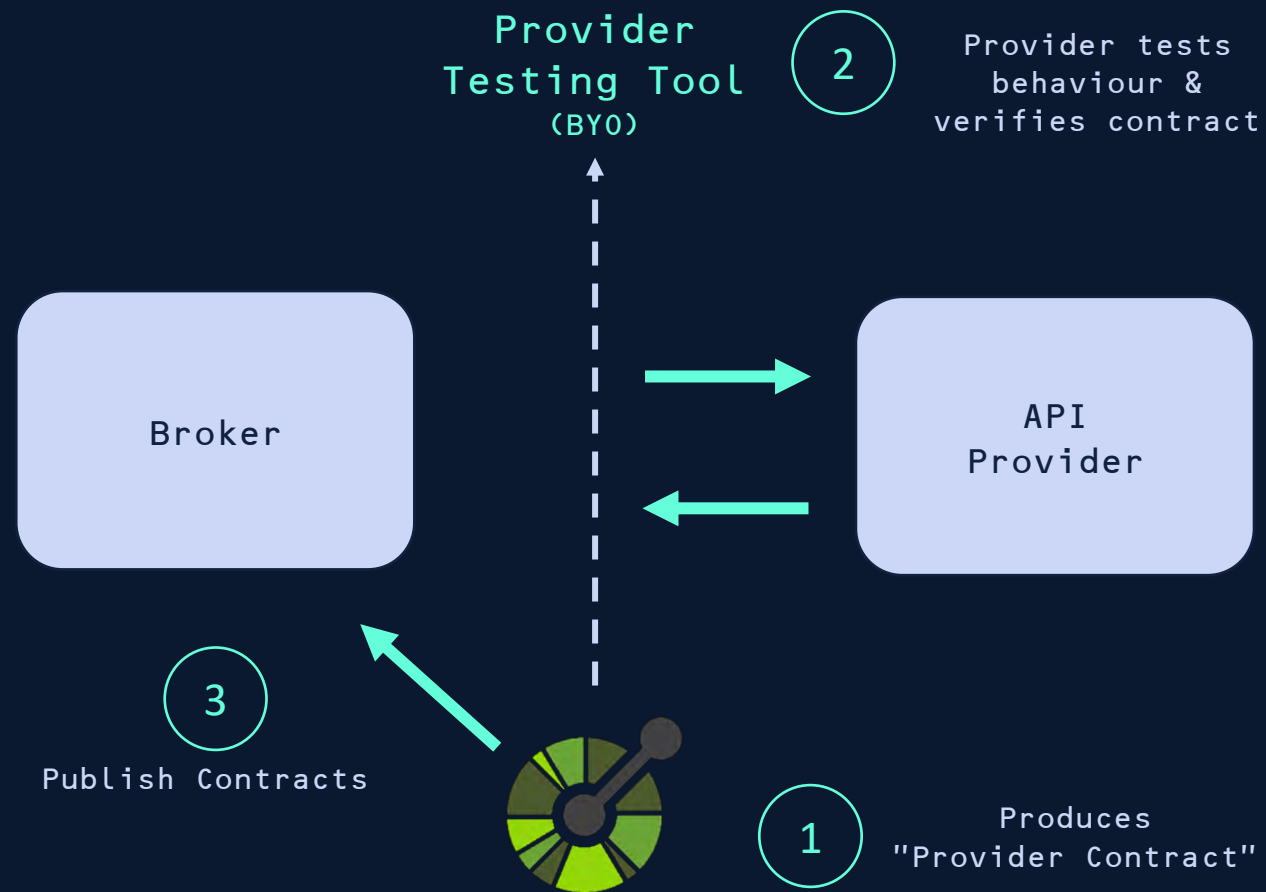
BDCT – How is works




```
/* walk through */
```

BDCT – How is works

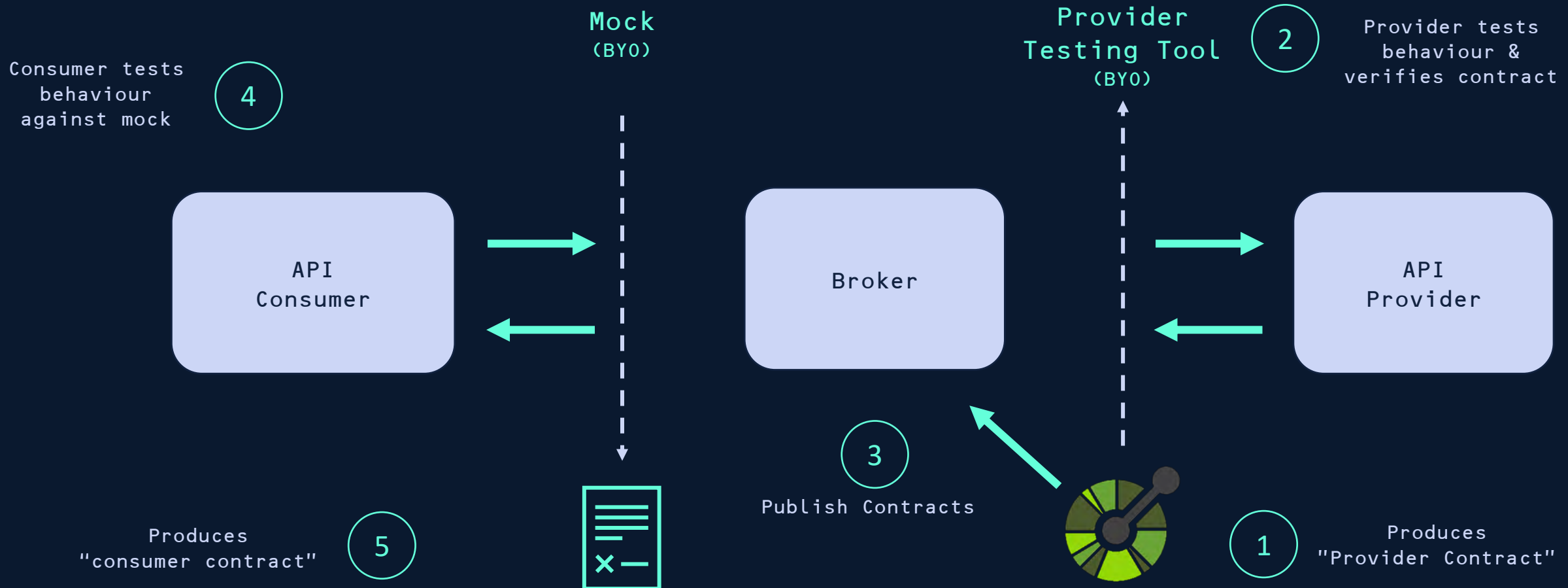
PACTFLOW



```
/* walk through */
```

BDCT – How is works

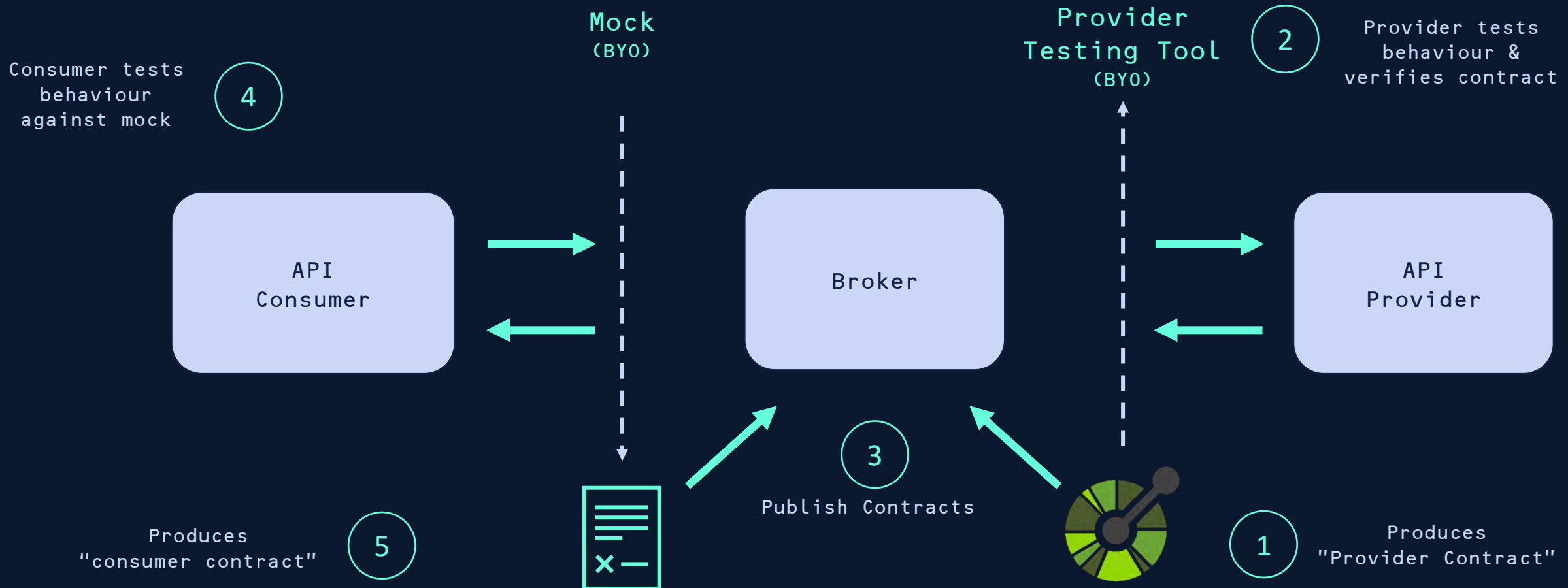
PACTFLOW



```
/* walk through */
```

BDCT – How is works

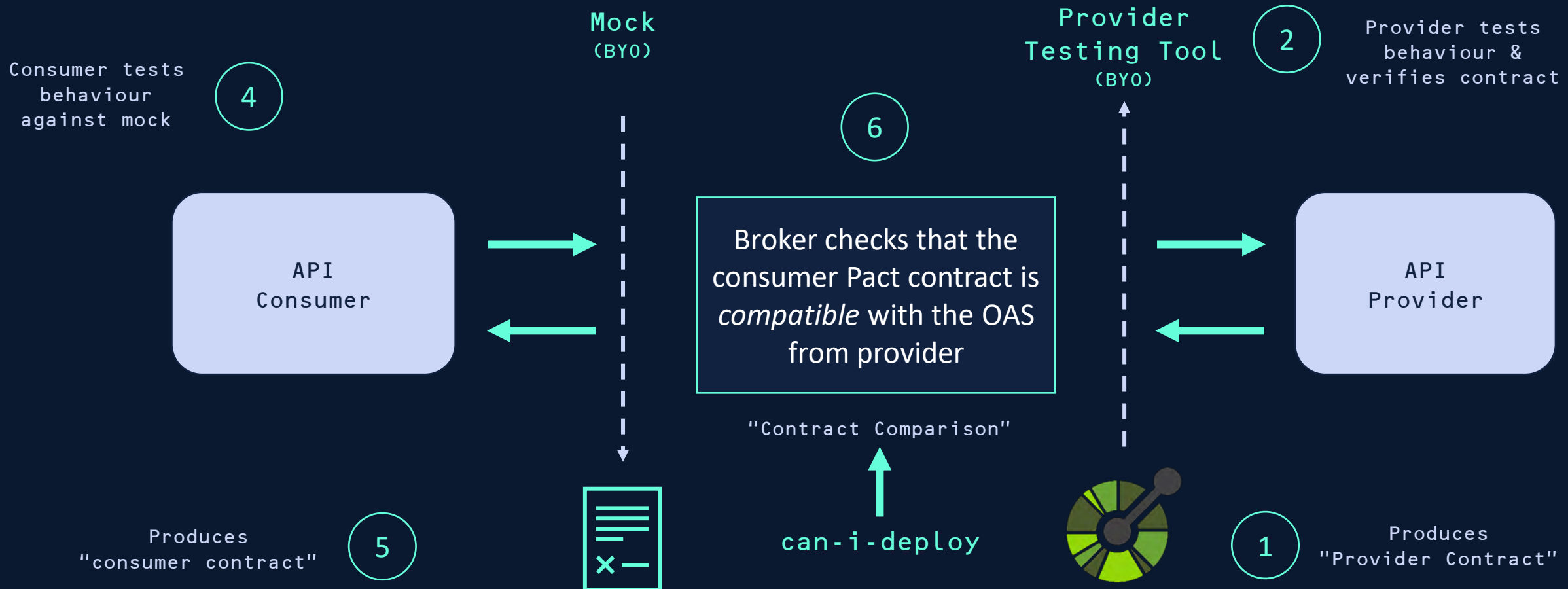
PACTFLOW



`/* walk through */`

BDCT – How is works

PACTFLOW



```
/* walk through */
```

BDCT – How is works

PACTFLOW

Consumer	Consumer version	Provider	Provider version	Success
Foo	1 (prod)	Bar	2	true
Foo	1 (prod)	Bar	3 (prod)	true
Foo	2	Bar	4 (prod)	true
Foo	2	Bar	5	true

Produces "consumer contract"

5



can-i-deploy

"Contract Comparison"

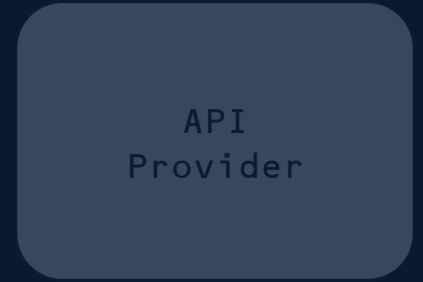
6

Consumer checks that the other Pact contract is compatible with the OAS from provider

Provider Testing Tool (BYO)

2

Provider tests behaviour & verifies contract



1

Produces "Provider Contract"



DEMO ...ish

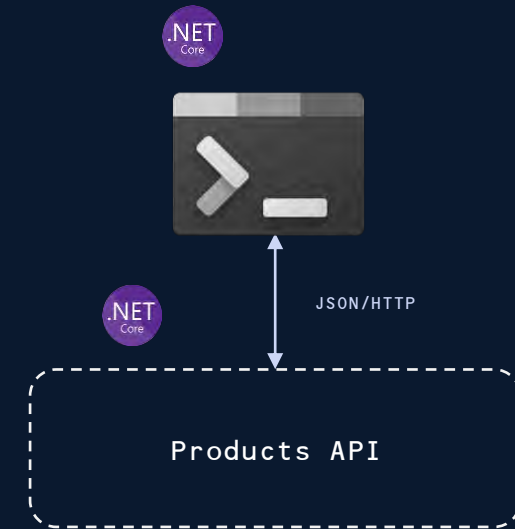
› Provider API

- › Products API written in C#
- › Using Schemathesis to test the API

› Consumer

- › Product API Consumer Client (C#/.NET core)
- › Consumer testing using Wiremock as mocking tool

› GitHub Actions for CI



```

1 openapi: 3.0.3
2 info:
3   title: Products API
4   description: |
5     A sample Products API to demonstrate Bi-Directional Contract Testing for ASC 2022
6   version: 1.0.0
7 paths:
8   /Products:
9     get:
10      summary: Retrieve a list of products
11      description: Get a list of products available
12      tags:
13        - Products
14      responses:
15        '200':
16          description: Success
17          content:
18            text/plain:
19              schema:
20                type: array
21                items:
22                  $ref: '#/components/schemas/Product'
23            application/json:
24              schema:
25                type: array
26                items:
27                  $ref: '#/components/schemas/Product'
28            text/json:
29              schema:
30                type: array
31                items:
32                  $ref: '#/components/schemas/Product'
33   /Products/{id}:
34     get:
35      summary: Retrieve details on a specific product
36      description: Get the full details on a particular product from the catalog
37      tags:
38        - Products
39      parameters:
40        - name: id
41          in: path
42          required: true
43          schema:
44            type: integer
45            format: int32
46      responses:
47        '200':
48          description: Success
49          content:
50            text/plain:
51              schema:
52                $ref: '#/components/schemas/Product'
53            application/json:
54              schema:
55                $ref: '#/components/schemas/Product'
56            text/json:
57              schema:
58                $ref: '#/components/schemas/Product'
59        '204':
60          description: Success
61   /Products/{int}:

```

Products API 1.0.0 OAS3

A sample Products API to demonstrate Bi-Directional Contract Testing for ASC 2022

Products ^

- GET /Products Retrieve a list of products ▼
- GET /Products/{id} Retrieve details on a specific product ▼
- DELETE /Products/{int} Delete a product from the catalog ▼

Schemas ^

- Product >

Provider – Test using Schemathesis

```
PS C:\Users\frank.kilcommins\GitHub\forks\example-bi-directional-provider-dotnet> make verify_swagger
sh ./example-bi-directional-provider-dotnet/scripts/verify_swagger.sh
Started dotnet API with process ID: 829
Running schemathesis test to generate report
Stopping dotnet API
PS C:\Users\frank.kilcommins\GitHub\forks\example-bi-directional-provider-dotnet> █
```

```
===== SUMMARY =====
```

Performed checks:

not_a_server_error	101 / 101 passed	PASSED
status_code_conformance	101 / 101 passed	PASSED
content_type_conformance	101 / 101 passed	PASSED
response_headers_conformance	101 / 101 passed	PASSED
response_schema_conformance	101 / 101 passed	PASSED

Provider – CI using GitHub Actions

Manually triggered 3 minutes ago

frankkilcommins db5bce5

Status: Success

Total duration: 2m 4s

Artifacts: –

build.yml
on: workflow_dispatch

Matrix: test

- ✓ 1 job completed
Show all jobs

✓ can-i-deploy 21s

✓ deploy 31s

```
graph LR; A[Matrix: test] --> B[can-i-deploy 21s]; B --> C[deploy 31s];
```

Pactflow – Provider contract published to broker

???

Unknown

CONSUMER VERSION	N/A	PROVIDER VERSION	b5bce5-main+b5bce5
BRANCH	N/A	BRANCH	↙ main
ENVIRONMENTS	N/A	ENVIRONMENTS	Production
TAGS	N/A	TAGS	N/A
		PUBLISHED AT	3 minutes ago

[VIEW CONTRACT](#)

OVERVIEW CONTRACTS

Unknown **pactflow-example-bi-directional-provider-dotnet**

Provider Details

PROVIDER VERSION	b5bce5-main+b5bce5	PUBLISHED AT	5 minutes ago	BRANCH	↙ main
------------------	--------------------	--------------	---------------	--------	--------

[More provider details](#)

Provider Contract

CONTRACT TYPE	OpenAPI Swagger	SELF-VERIFICATION TEST RESULT	Success View Log
---------------	-----------------	-------------------------------	----------------------------------

Products API 1.0.0 QAS3

Products

- [GET](#) /Products
- [GET](#) /Products/{id}
- [DELETE](#) /Products/{int}

Consumer – Test using xunit and wiremock

```
[Fact]
0 references | Run Test | Debug Test
public async Task GetProduct_WhenCalledWithInvalidID_ReturnsError()
{
    // Arrange
    var server = WireMockServer.Start();
    String serverUrl = server.Url[0] + "/";
    server
        .WithConsumer(consumer)
        .WithProvider(provider)
        .Given(Request.Create().UsingGet().WithPath("/Products/10"))
        .WithTitle("a request to retrieve a product id that does not exist")
        .RespondWith(Response
            .Create()
            .WithStatusCode(HttpStatusCode.NotFound)
            .WithHeader("Content-Type",
                "application/json; charset=utf-8"));

    // Act
    var client = new ProductClient();
    var ex =
        await Assert
            .ThrowsAsync<HttpRequestException>(() =>
                client.GetProduct(serverUrl, 10, null));

    // Assert
    Assert
        .Equal("Response status code does not indicate success: 404 (Not Found).",
            ex.Message);

    server
        .SaveStaticMappings(Path
            .Combine("../", "../", "../", "wiremock-mappings"));



    // Save pact
    server
        .SavePact(Path.Combine("../", "../", "../", "pacts"),
            "get-product-by-id-not-exist.json");
}
```

Consumer – Test generates pact.json file

```
{
  "consumer": {
    "name": "pactflow-example-bi-directional-consumer-wiremock-dotnet"
  },
  "interactions": [
    {
      "providerState": "a request to retrieve a product id that does not exist",
      "request": {
        "method": "GET",
        "path": "/Products/10"
      },
      "response": {
        "headers": {
          "Content-Type": "application/json; charset=utf-8"
        },
        "status": 404
      }
    }
  ],
  "provider": {
    "name": "pactflow-example-bi-directional-provider-dotnet"
  }
}
```

Consumer – Another CI using GitHub Actions

Manually triggered 2 minutes ago Status Total duration Artifacts

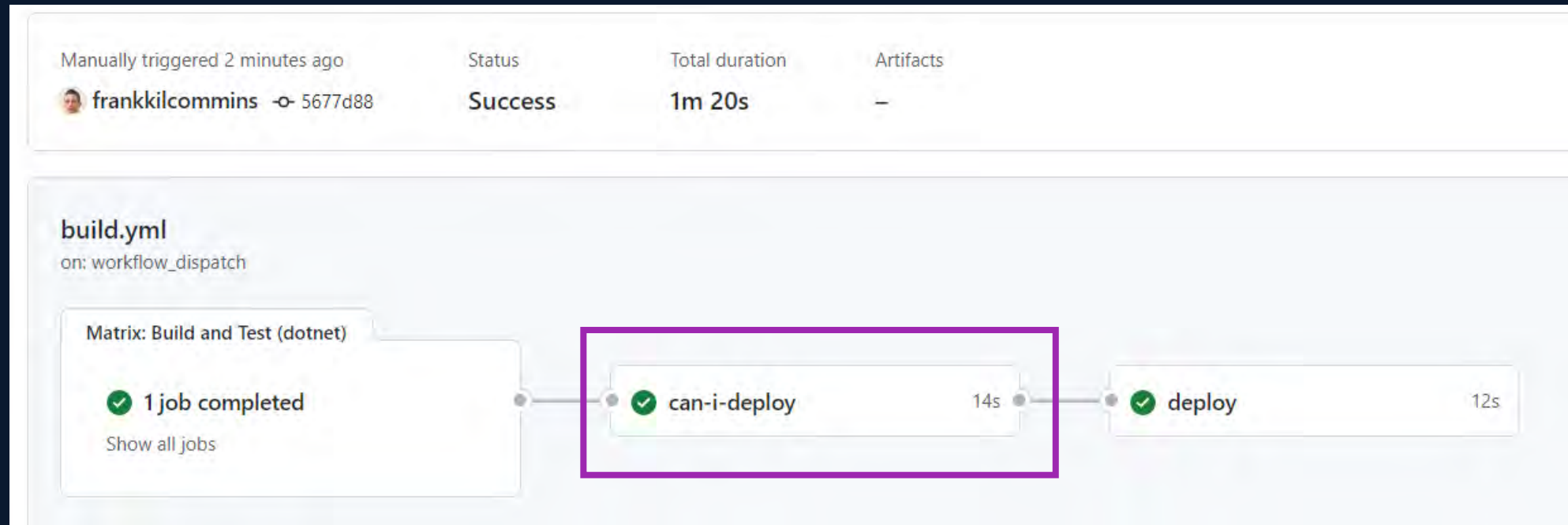
 frankkilcommins  5677d88 **Success** **1m 20s** –

build.yml
on: workflow_dispatch

Matrix: Build and Test (dotnet)

✔ 1 job completed
[Show all jobs](#)

```
graph LR; A[✔ 1 job completed] --> B[✔ can-i-deploy 14s]; B --> C[✔ deploy 12s];
```



A pact between pactflow-example-bi-directional-consumer-wiremock-dotnet and pactflow-example-bi-directional-provider-dotnet



Consumer Details

CONSUMER VERSION

5677d880a56a136df9ba0e8fddff1c5d490f8f7f

[More consumer details](#)

RELEASED ENVIRONMENTS

N/A

TAGS

main

PUBLISHED AT

2 hours ago

BRANCH

main

DEPLOYED ENVIRONMENTS

Production

Provider Details

PROVIDER VERSION

20be87-main+20be87

[More provider details](#)

RELEASED ENVIRONMENTS

N/A

TAGS

N/A

PUBLISHED AT

2 hours ago

BRANCH

main

DEPLOYED ENVIRONMENTS

Production

CONTRACT COMPARISON

CONSUMER CONTRACT

PROVIDER CONTRACT

Consumer Contract



CONSUMER CONTRACT STATUS

Compatible

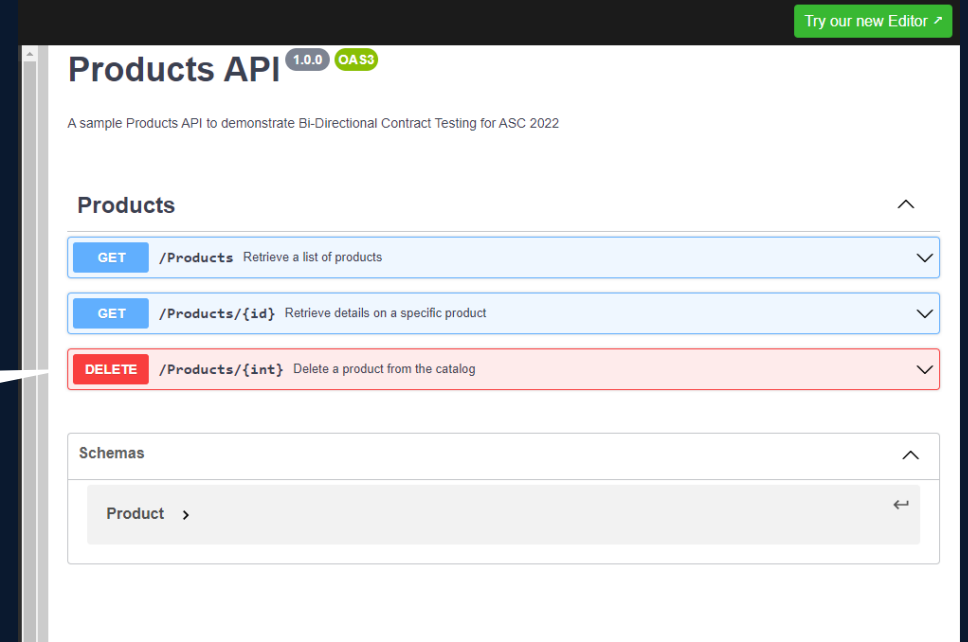
PACT SPEC VERSION

Unknown

- ✓ given a request to retrieve a product id that does not exist
- ✓ given a request to retrieve a product with existing id
- ✓ given a request to retrieve all products

Potentially Breaking Change

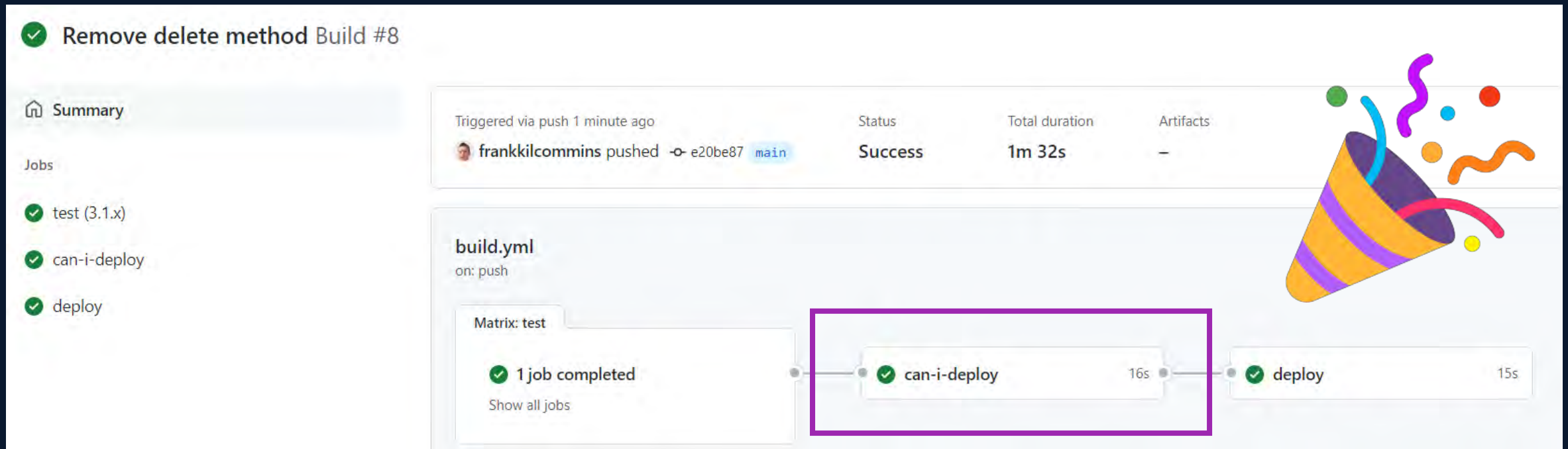
Product Owner:
"Please remove DELETE method,
it's for admin API only"



The screenshot shows the 'Products API' documentation. At the top right, there is a green button that says 'Try our new Editor'. Below the title, it says 'A sample Products API to demonstrate Bi-Directional Contract Testing for ASC 2022'. Under the 'Products' section, there are three API endpoints listed:

- GET** /Products: Retrieve a list of products
- GET** /Products/{id}: Retrieve details on a specific product
- DELETE** /Products/{int}: Delete a product from the catalog

The 'DELETE' endpoint is highlighted with a red background. Below this, there is a 'Schemas' section with a 'Product' schema listed.



The screenshot shows a CI/CD pipeline build page for 'Remove delete method Build #8'. The build is triggered via push 1 minute ago by 'frankkilcommins' on the 'main' branch. The status is 'Success' with a total duration of '1m 32s'. The pipeline consists of three jobs: 'test (3.1.x)', 'can-i-deploy', and 'deploy'. The 'can-i-deploy' job is highlighted with a purple box and has a duration of 16s. The 'deploy' job has a duration of 15s. A confetti icon is visible on the right side of the page.

Select team

+ Create Example Project

Filter your pacts

Integration

- pactflow-example-bi-directional-consumer-wiremock-dotnet
- pactflow-example-bi-directional-provider-dotnet
- Example Bi-Directional Consumer
- Example Bi-Directional Provider
- Example App
- Example API

OVERVIEW NETWORK DIAGRAM MATRIX WEBHOOKS **CONTRACTS**

Consumer Details

CONSUMER VERSION 5677d880a56a136df9ba0e8fddf1c5d490f8f7f More consumer details	PUBLISHED AT 13 minutes ago	BRANCH main
--	--------------------------------	----------------

Provider Details

PROVIDER VERSION 20be87-main+20be87 More provider details	PUBLISHED AT 2 minutes ago	BRANCH main
---	-------------------------------	----------------

CONTRACT COMPARISON CONSUMER CONTRACT **PROVIDER CONTRACT**

Provider Contract

CONTRACT TYPE: OpenAPI Swagger SELF-VERIFICATION TEST RESULT: Success [View Log](#)

Products API 1.0.0 OAS3

Products

- GET /Products
- GET /Products/{id}

A Breaking Change

Product Owner:
"The Metrics are killing us –
reduce the errors ASAP"

✖ Change GetByld response for no match to 204 Build #9

🏠 Summary

Jobs

✔ test (3.1.x)

✖ can-i-deploy

🕒 deploy

Triggered via push 1 minute ago

👤 frankkilcommins pushed `f64247a` `main`

Status

Failure

Total duration

1m 14s

Artifacts

–

build.yml

on: push

Matrix: test

✔ 1 job completed

Show all jobs

✖ can-i-deploy

12s

🕒 deploy

0s

A Breaking Change

```
Can I deploy? 6s
10
11 ===== STAGE: can-i-deploy? 🚩 =====
12
13 docker run --rm -v //home/runner/work/example-bi-directional-provider-dotnet/example-bi-directional-provider-dotnet://home/runner/work/example-bi-directional-provider-dotnet/example-bi-directional-provider-dotnet -w /home/runner/work/example-bi-directional-provider-dotnet/example-bi-directional-provider-dotnet -e PACT_BROKER_BASE_URL -e PACT_BROKER_TOKEN pactfoundation/pact-cli:latest pact-broker can-i-deploy --pacticipant "pactflow-example-bi-directional-provider-dotnet" --version 64247a-main+64247a --to-environment production
14 Computer says no 🙄_/_
15
16 CONSUMER | C.VERSION | PROVIDER | P.VERSION | SUCCESS? |
17 RESULT# |-----|-----|-----|-----|-----|
18 pactflow-example-bi-directional-consumer-wiremock-dotnet | 5677d88... | pactflow-example-bi-directional-provider-dotnet | 64247a-main+64247a | false | 1
19
20 VERIFICATION RESULTS
21 -----
22 1. https://smartbear-frank.pactflow.io/contracts/bi-directional/provider/pactflow-example-bi-directional-provider-dotnet/version/64247a-main%2B64247a/consumer/pactflow-example-bi-directional-consumer-wiremock-dotnet/version/5677d880a56a136df9ba0e8fddff1c5d490f8f7f/cross-contract-verification-results (failure)
23
24 The cross contract verification between the pact for the version of pactflow-example-bi-directional-consumer-wiremock-dotnet currently deployed or released to production (5677d880a56a136df9ba0e8fddff1c5d490f8f7f) and the oas for version 64247a-main+64247a of pactflow-example-bi-directional-provider-dotnet failed
25 make: *** [Makefile:95: can_i_deploy] Error 1
26 Error: Process completed with exit code 2.
```

OVERVIEW

NETWORK DIAGRAM

MATRIX

WEBHOOKS

CONTRACTS

example-bi-directional-provider-dotnet

Consumer Details

CONSUMER VERSION
5677d880a56a136df9ba0e8fdff1c5d490f87f

PUBLISHED AT
21 minutes ago

BRANCH
main

[More consumer details](#)

Provider Details

PROVIDER VERSION
64247a-main+64247a

PUBLISHED AT
3 minutes ago

BRANCH
main

[More provider details](#)

CONTRACT COMPARISON

CONSUMER CONTRACT

PROVIDER CONTRACT

Contract Comparison Detail

CONTRACT COMPARISON STATUS
Incompatible

COMPARED AT
2 minutes ago

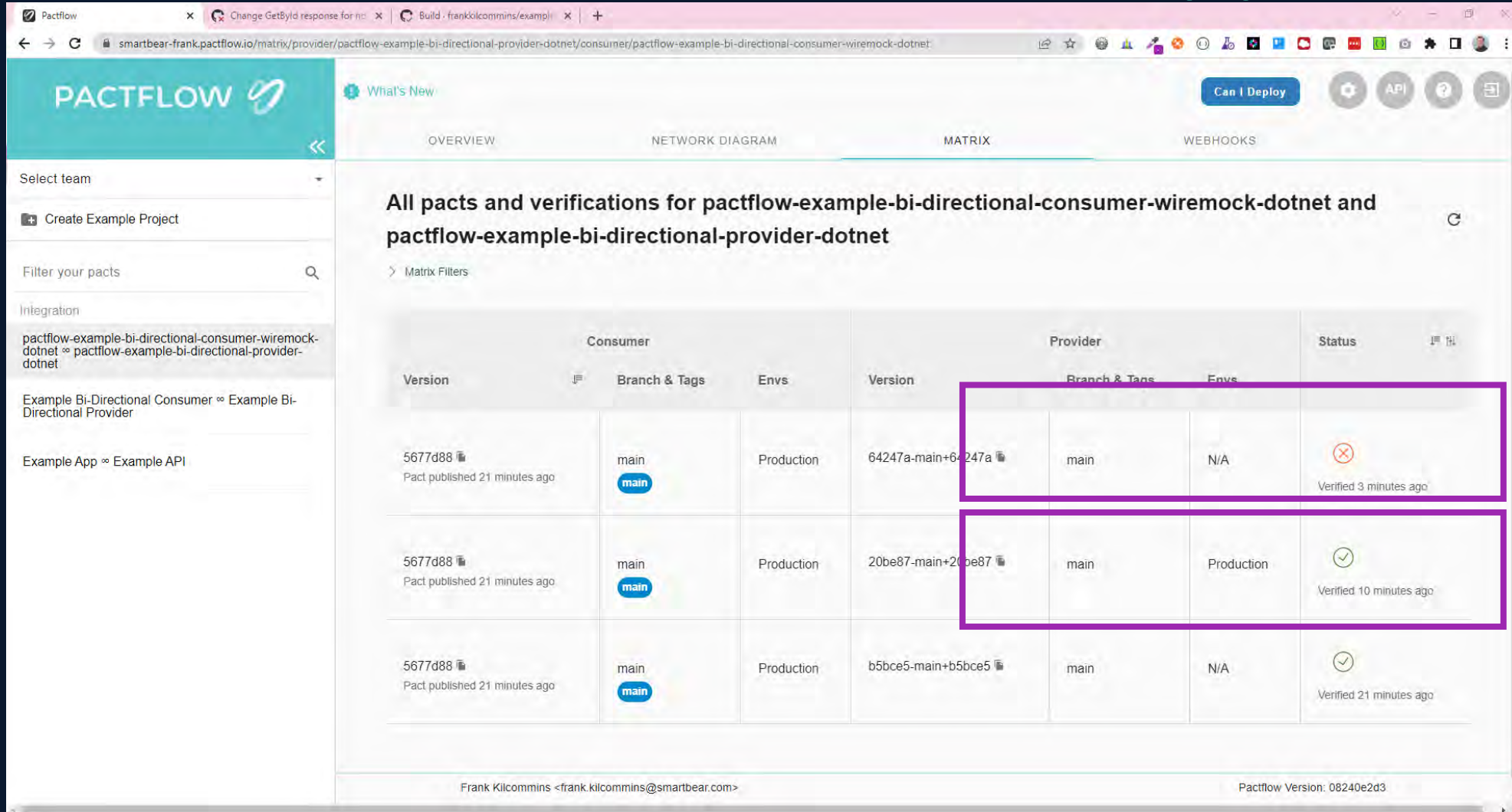
get /Products/{id}

Unknown Incompatibility
Response status code not defined in spec file: 404










Mismatched Pact Path:
`[root].interactions[0].response.status`

Mismatched Provider Path:
`[root].paths./Products/{id}.get.responses`

The Matrix – information to safely promote



The screenshot displays the Pactflow Matrix interface. The main heading reads: "All pacts and verifications for pactflow-example-bi-directional-consumer-wiremock-dotnet and pactflow-example-bi-directional-provider-dotnet". Below this is a table with columns for Consumer (Version, Branch & Tags, Envs) and Provider (Version, Branch & Tags, Envs), along with a Status column. Two rows are highlighted with a purple box:

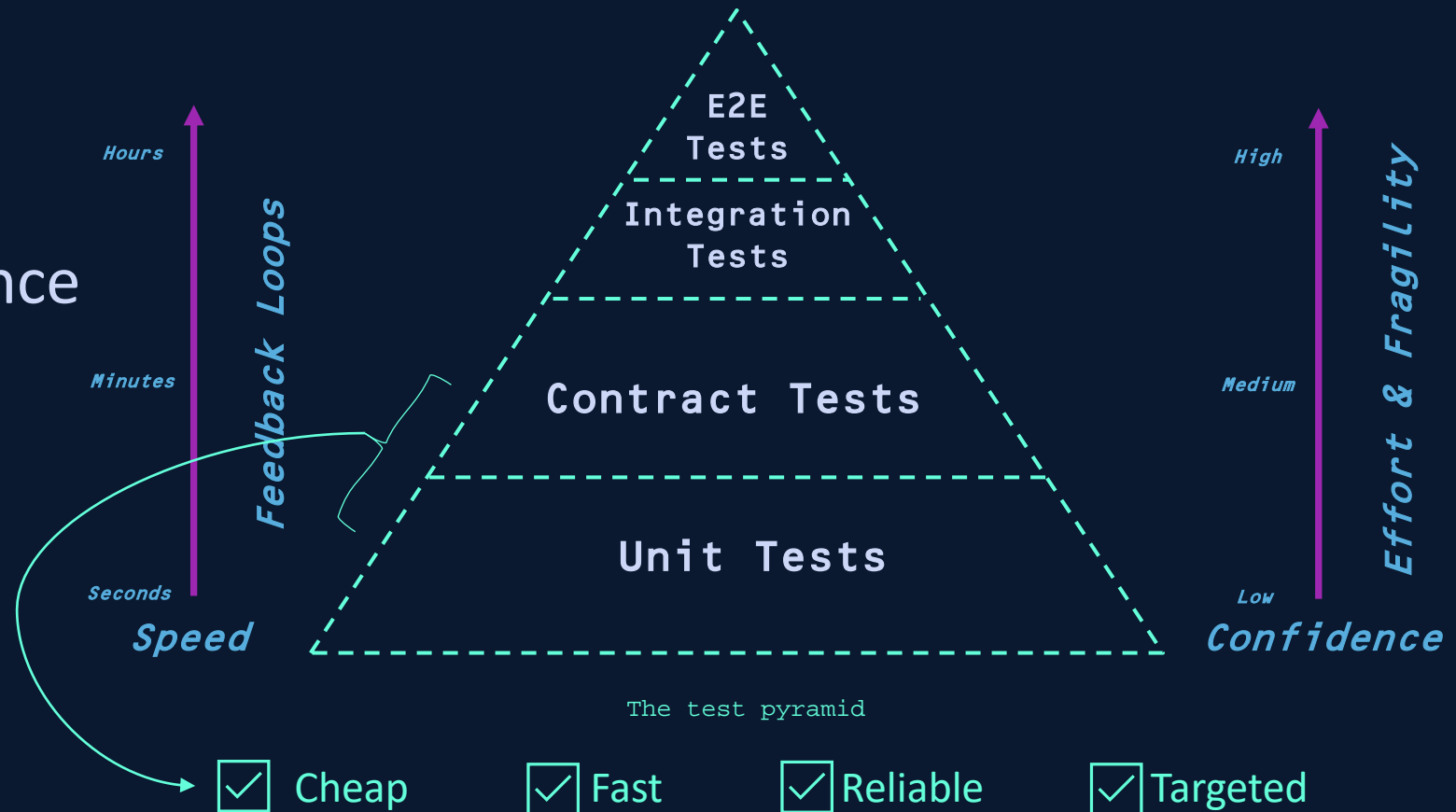
Consumer			Provider			Status
Version	Branch & Tags	Envs	Version	Branch & Tags	Envs	
5677d88 Pact published 21 minutes ago	main 	Production	64247a-main+64247a 	main	N/A	 Verified 3 minutes ago
5677d88 Pact published 21 minutes ago	main 	Production	20be87-main+20be87 	main	Production	 Verified 10 minutes ago
5677d88 Pact published 21 minutes ago	main 	Production	b5bce5-main+b5bce5 	main	N/A	 Verified 21 minutes ago

At the bottom of the interface, the user is identified as "Frank Kilcommins <frank.kilcommins@smartbear.com>" and the Pactflow version is "08240e2d3".

```
/* Problem addressed: unbalanced testing strategy */
```

Rebalance the approach to microservices testing

- › Reduced E2E tests
- › Reduce integration tests
- › Reduce assumptions
- › Increase delivery confidence
- › Deploy independently
- › Scale predictably



```
/* Problem addressed: lack of visibility into how consumers are using an API */
```

Benefits for Design-First

- › Visibility into consumers
- › Reduce the need for API major versioning
- › Prevent breaking changes – reducing assumptions (drift)
- › Know when it's safe to deploy new changes
- › Better conversations
- › Design-first with confidence

Thank You!

› Connect:



@fkilcommins



@frank-kilcommins



frank.kilcommins@smartbear.com

Give it a try

Head to go.pactflow.io/design-first



The screenshot shows the Pactflow Documentation website. The navigation menu on the left includes sections like 'Getting Started with Pactflow', 'User Interface', 'Features', 'Contract Testing', 'Pact', 'Bi-Directional Contract Testing', 'Overview', 'Consumer', 'Provider', 'Publishing Contracts', 'Compatibility Checks', 'Deploying and Releasing', 'Supported Contracts', 'Tool Integration', 'SwaggerHub', 'Cypress', 'MSW', 'Wiremock (Java)', 'Wiremock (.NET)', and 'Account'. The main content area is titled 'Integration Guide' and '1 Setup your API Mock Environment (Optional)'. It includes a 'SwaggerHub feature reference' and a paragraph explaining that the API Auto Mocking integration creates a semi-static mock of the API based on OpenAPI 3.0 or 2.0 definitions. Below the text is a diagram titled 'Design first contract testing - Provider (Automock)' which illustrates a workflow from design to testing. The diagram shows a 'Design' step leading to 'Consumer Automock' and 'Provider Automock', which then leads to 'Test' and 'Publish to Pactflow'. The 'Test' step is further divided into 'Test' and 'Test (API)'. The 'Publish to Pactflow' step leads to 'Test' and 'Test (API)'. The diagram is numbered 1 through 6, corresponding to the steps in the text below.

1. Create dev mock env with SWH and list up on SWH

Resources contributing to this talk:

- › <https://smartbear.com/state-of-software-quality/api/>
- › <https://www.postman.com/state-of-api/>
- › <https://docs.pactflow.io/docs/workshops>
- › <https://www.youtube.com/watch?v=a-JoVnpZzNg> - (Erik Wilde - Designing APIs for Extensibility)
- › <https://www.sealights.io/webinars/achieving-quality-at-speed-in-a-modern-software-world-microservices-delivery-without-the-pitfalls/> - (Seb Rose)