Let’s make a Pact

Don’t break my API

Frank Kilcommins / SmartBear

LangDev CON 2024

FS9U

ProxyHands

Seville 17-19 October, 2024

https://langdevcon.org

I’m Frank Kilcommins

- Principal API Technical Evangelist @ SmartBear
- Software Engineer and Architect (❤ APIs & Developer Experience)
- Governance Board member on OpenAPI Initiative (OAI)
- Contributor on the Arazzo Specification

- Connect:

/* insert embarrassing photo here */

@fkilcommins

@frank-kilcommins

frank.kilcommins@smartbear.com

SMARTBEAR

Talk Track

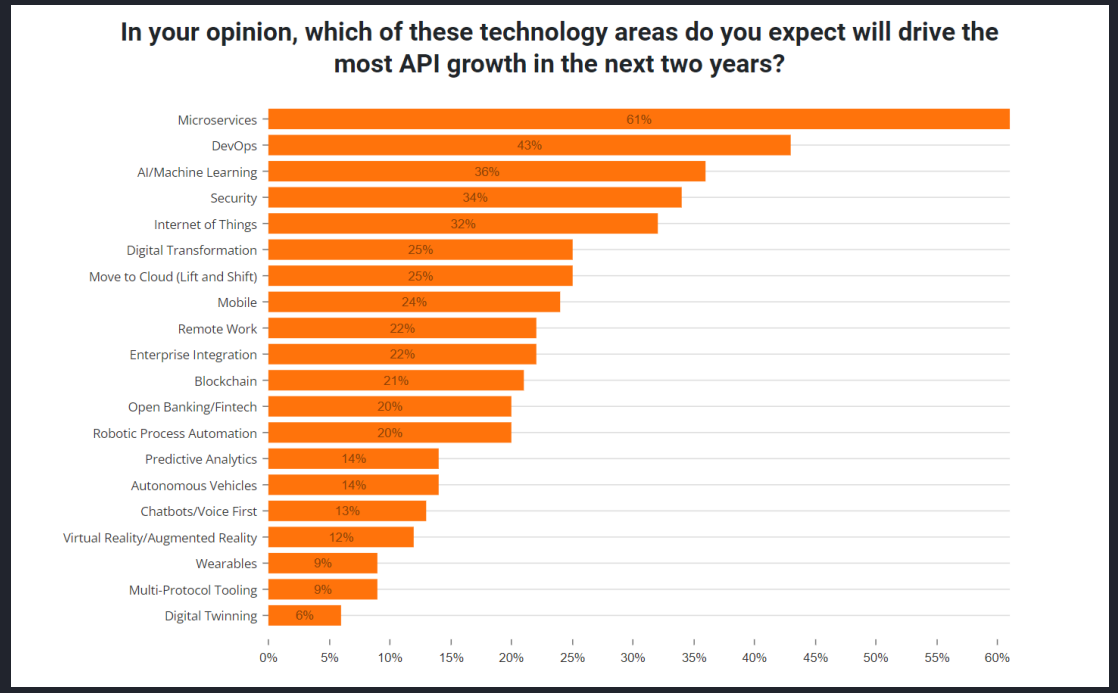
FS9U

ProxyHands

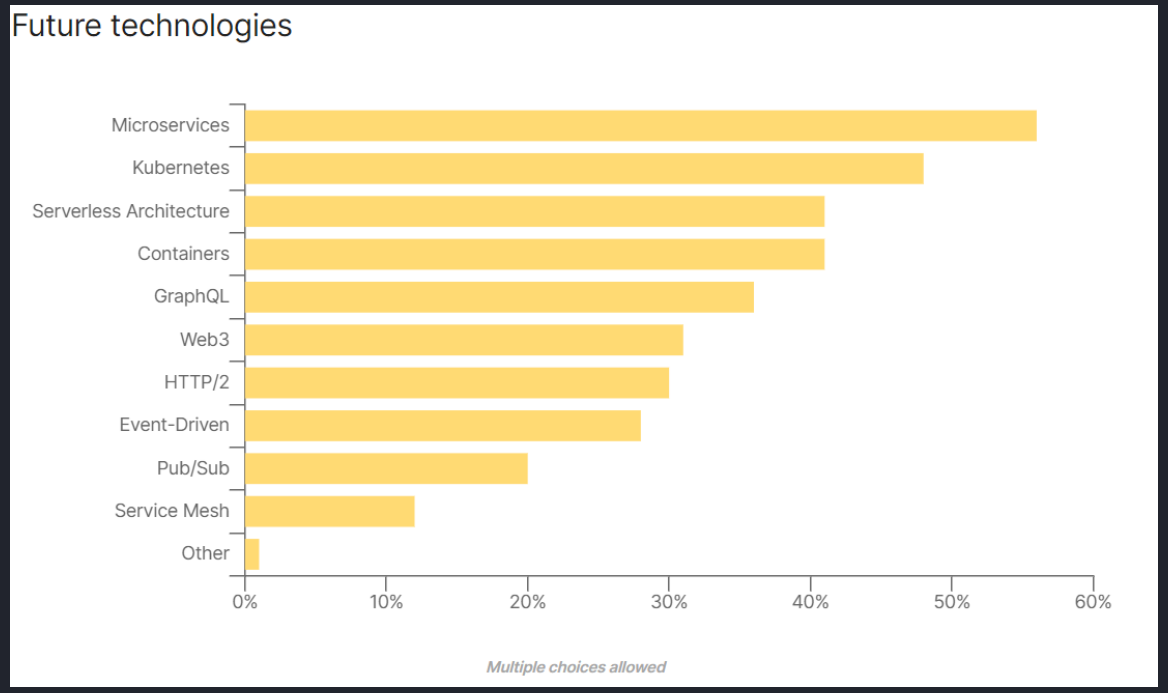
- API Landscape Trends
- Designing for the future
- Is extensibility enough?
- Bi-Directional Contract Testing – An approach to calming the chaos
- Demo
- Takeaways

API Landscape Trends

› Microservices driving API growth



SmartBear – State of Software Quality | API 2023

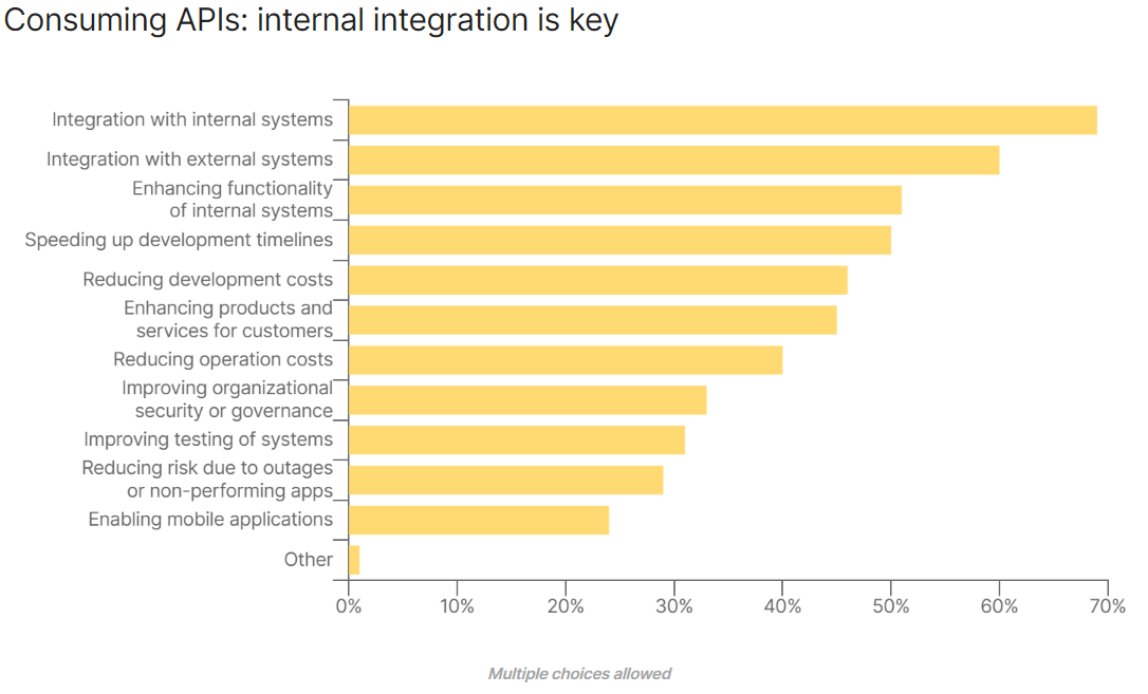


Postman – State of API Report 2023

SMARTBEAR

API Landscape Trends

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



Postman – State of API Report 2023

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

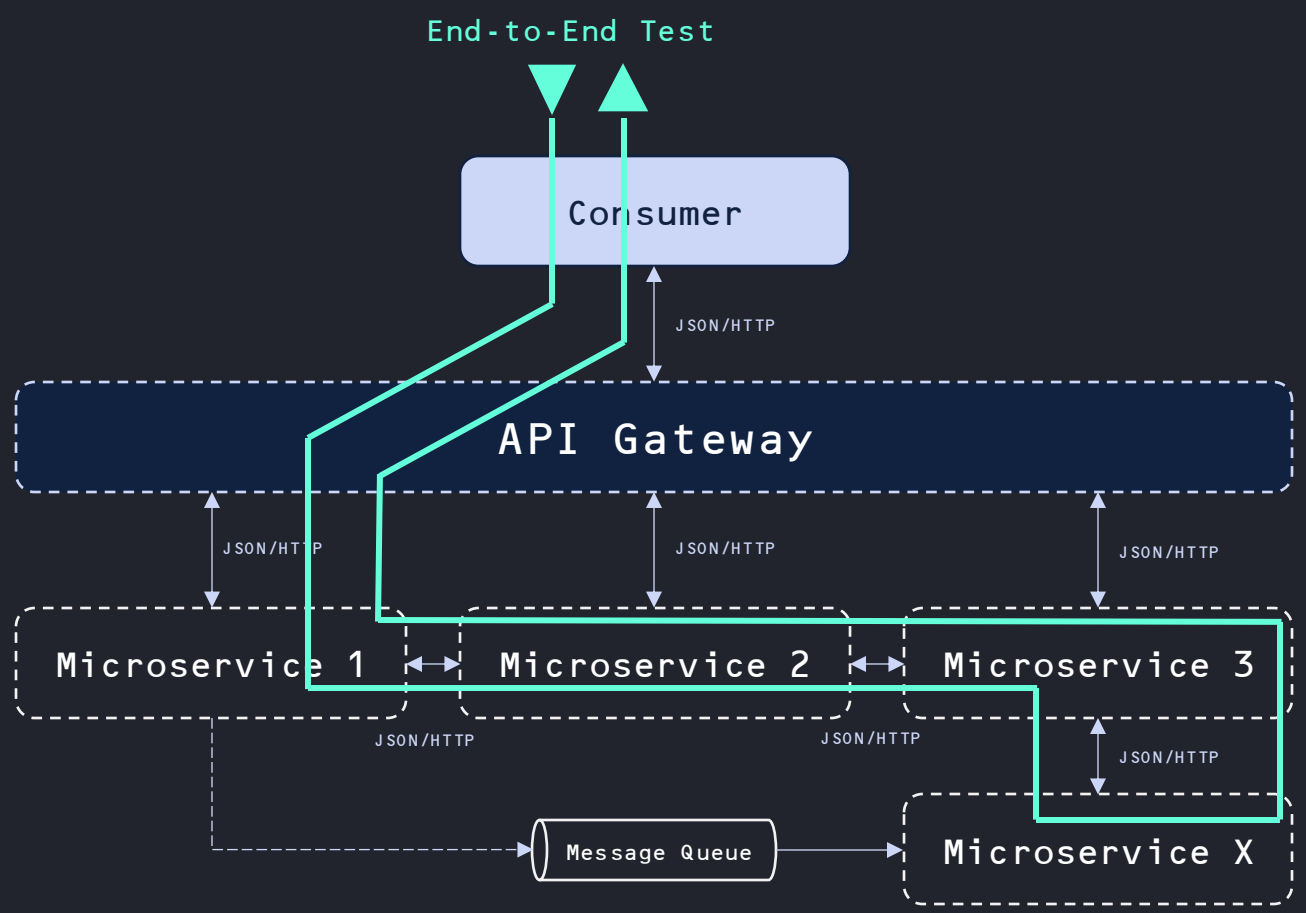
Pitfalls remain ☹

› Major version proliferation

Pitfalls remain ☹

$ › Major version proliferation

› Unbalanced testing approach

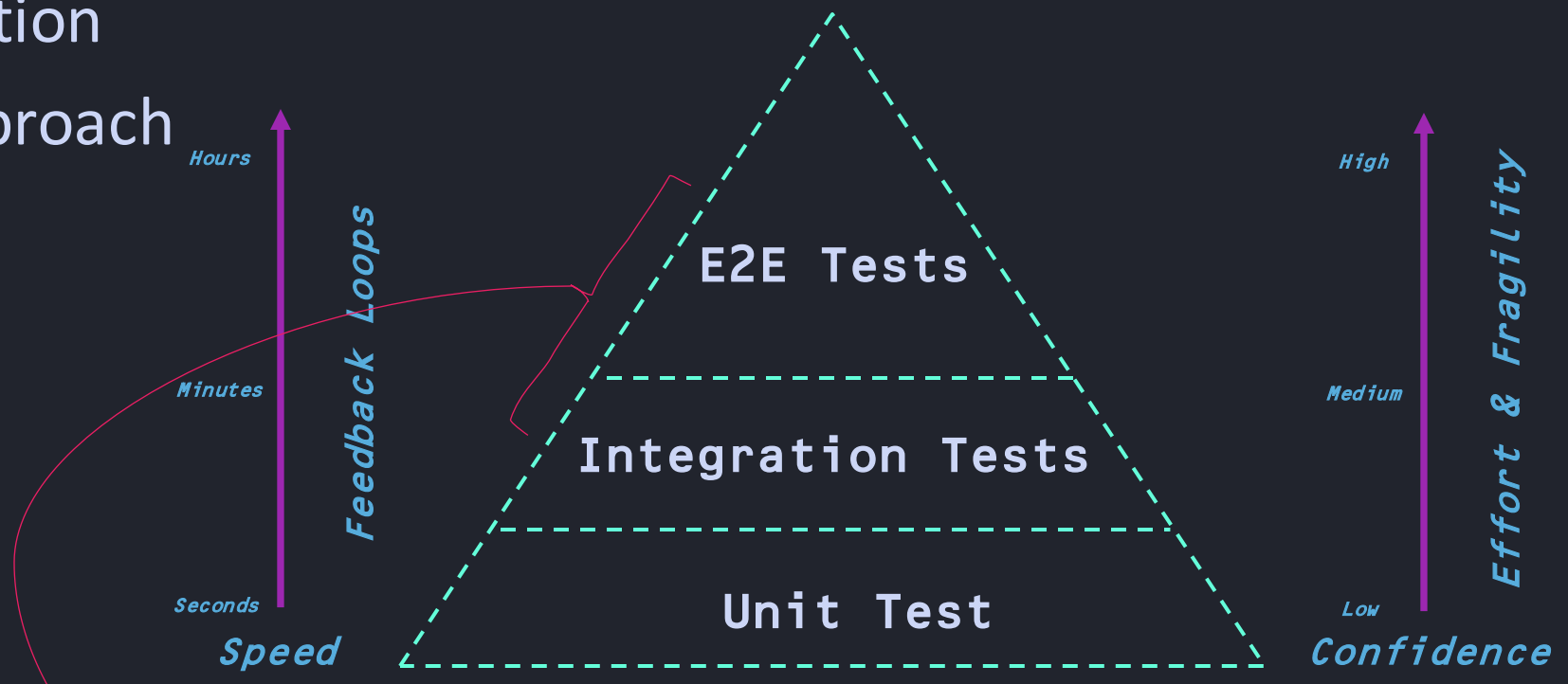


SMARTBEAR

Pitfalls remain ☹

$ › Major version proliferation

$ › Unbalanced testing approach



The test pyramid

x Expensive x Slow x Unreliable x Not targeted

SMARTBEAR

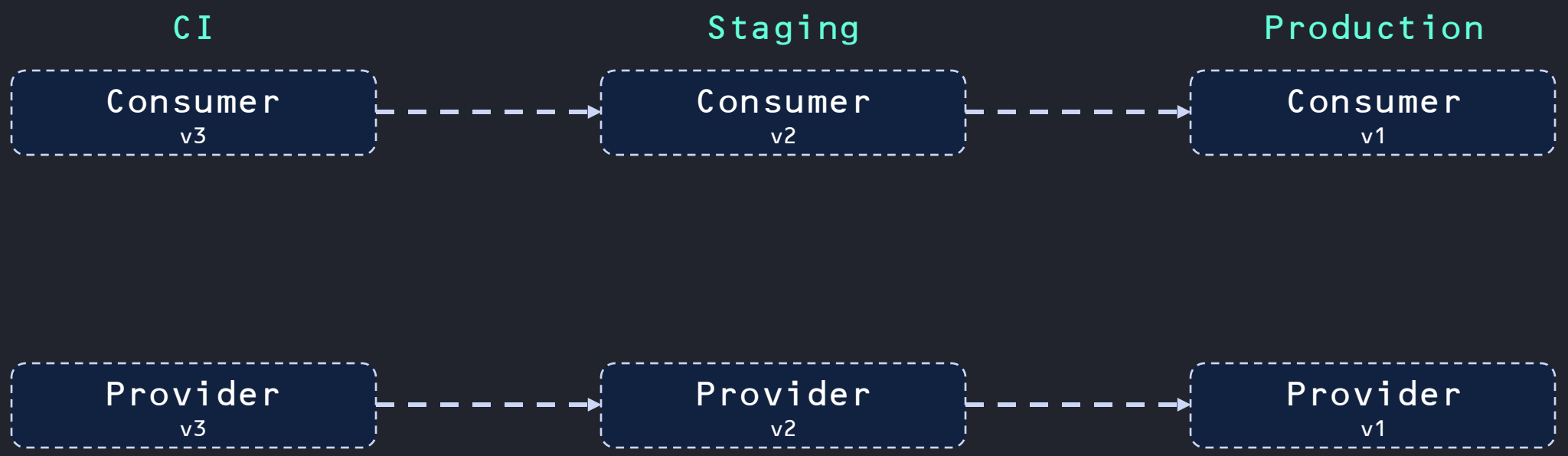
/* ToDo: address the complexity */

Pitfalls remain ☹

$ › Major version proliferation

$ › Unbalanced testing approach

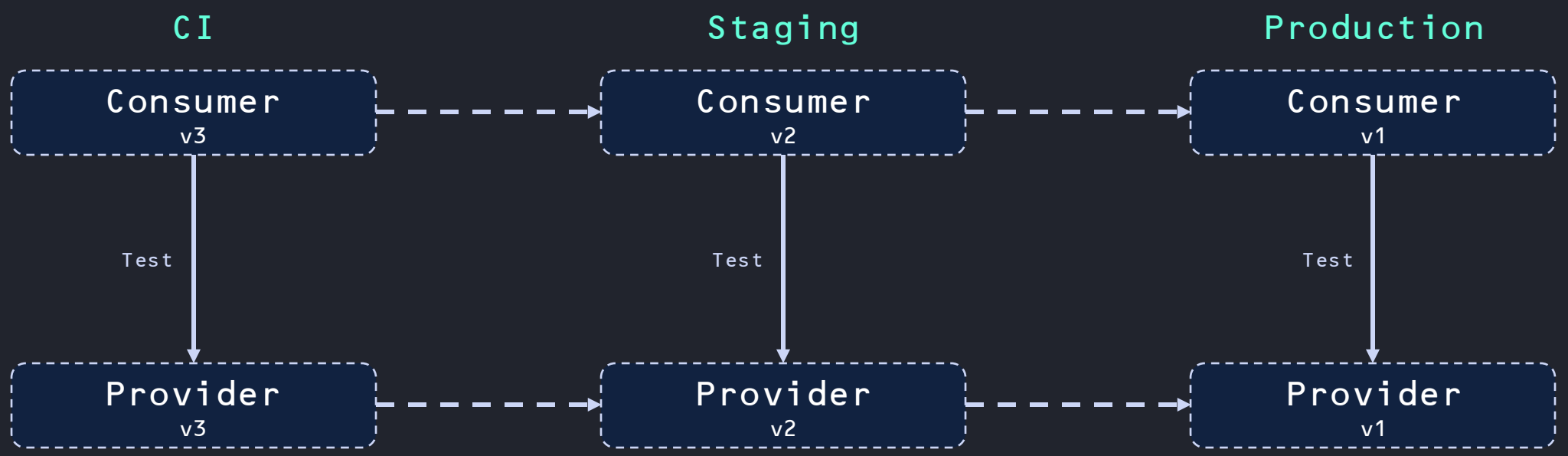
› Environment management (dependencies)



/* ToDo: address the complexity */

Pitfalls remain ☹

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

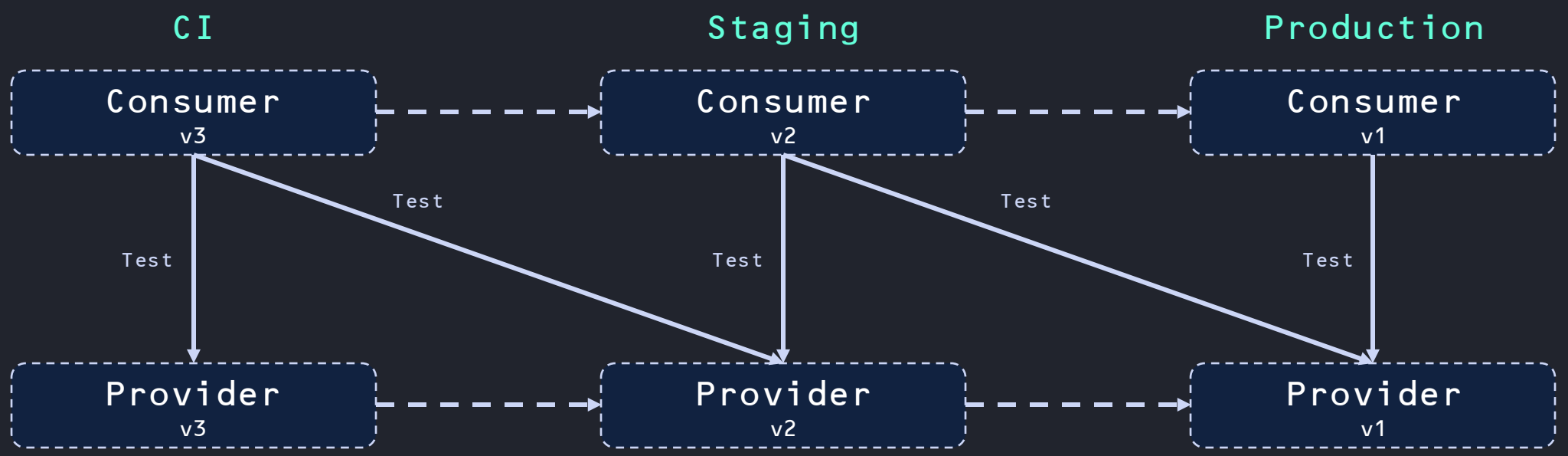


SMARTBEAR

/* ToDo: address the complexity */

Pitfalls remain ☹

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



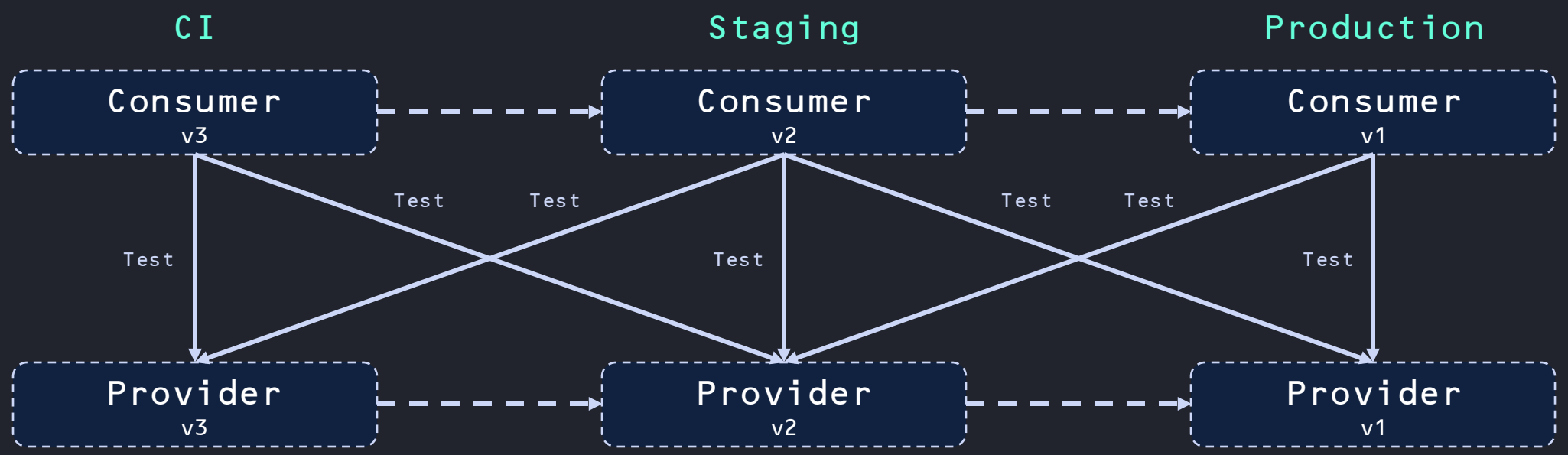
/* ToDo: address the complexity */

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

SMARTBEAR

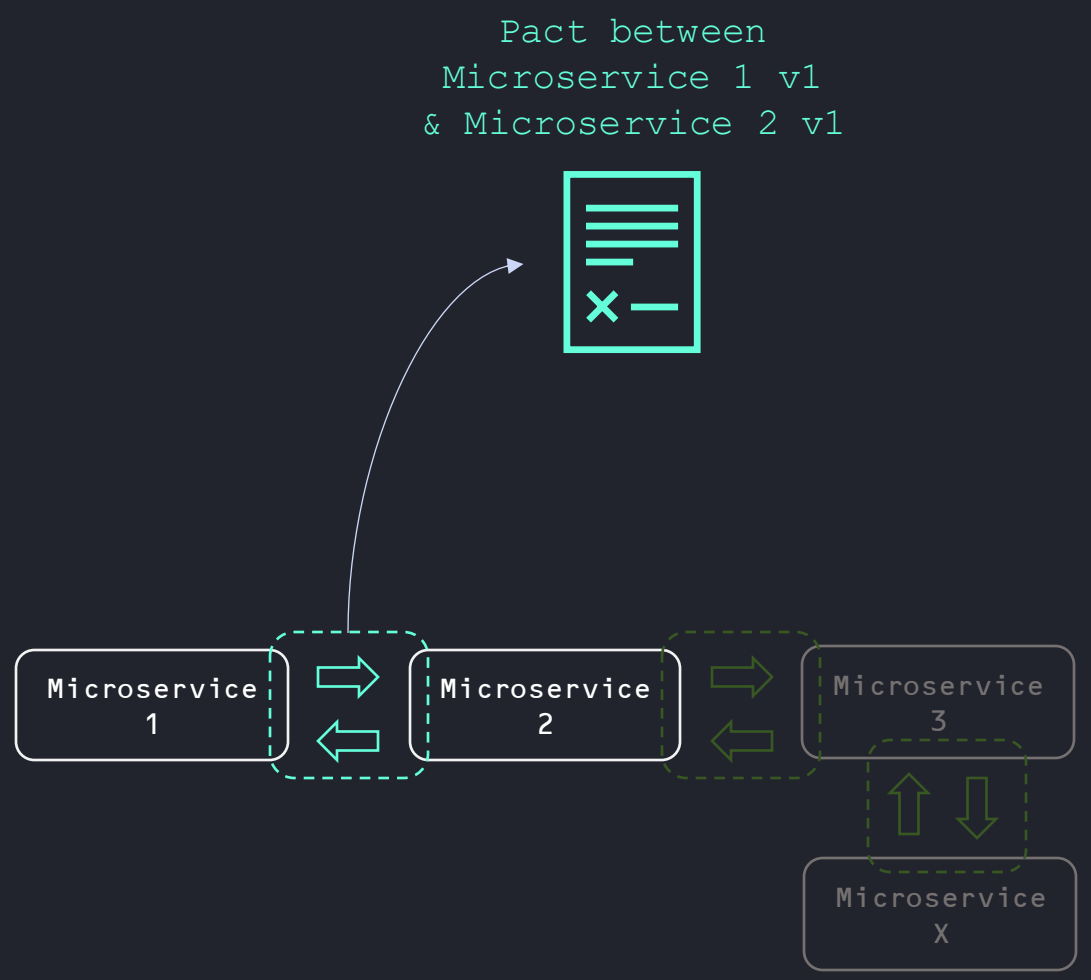
Bi-Directional Contract Testing (BDCT)

- 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)



SMARTBEAR

/* 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

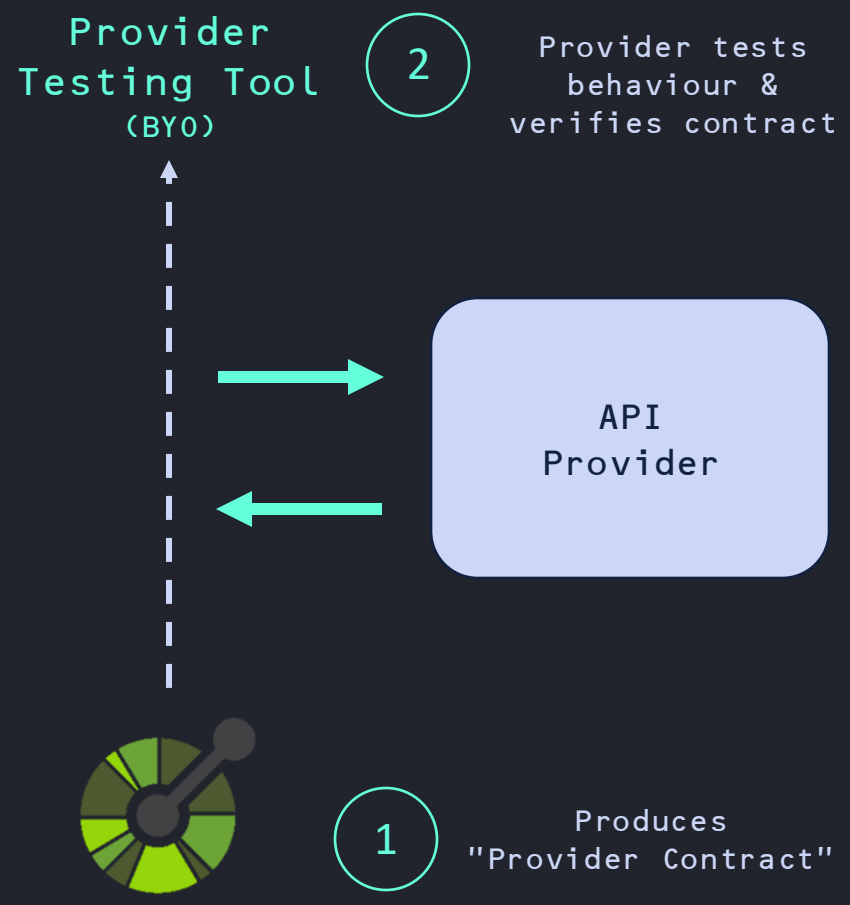
Pact.json file example

```
{
  "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"
    }
  }
}
```

SMARTBEAR

/* walk through */

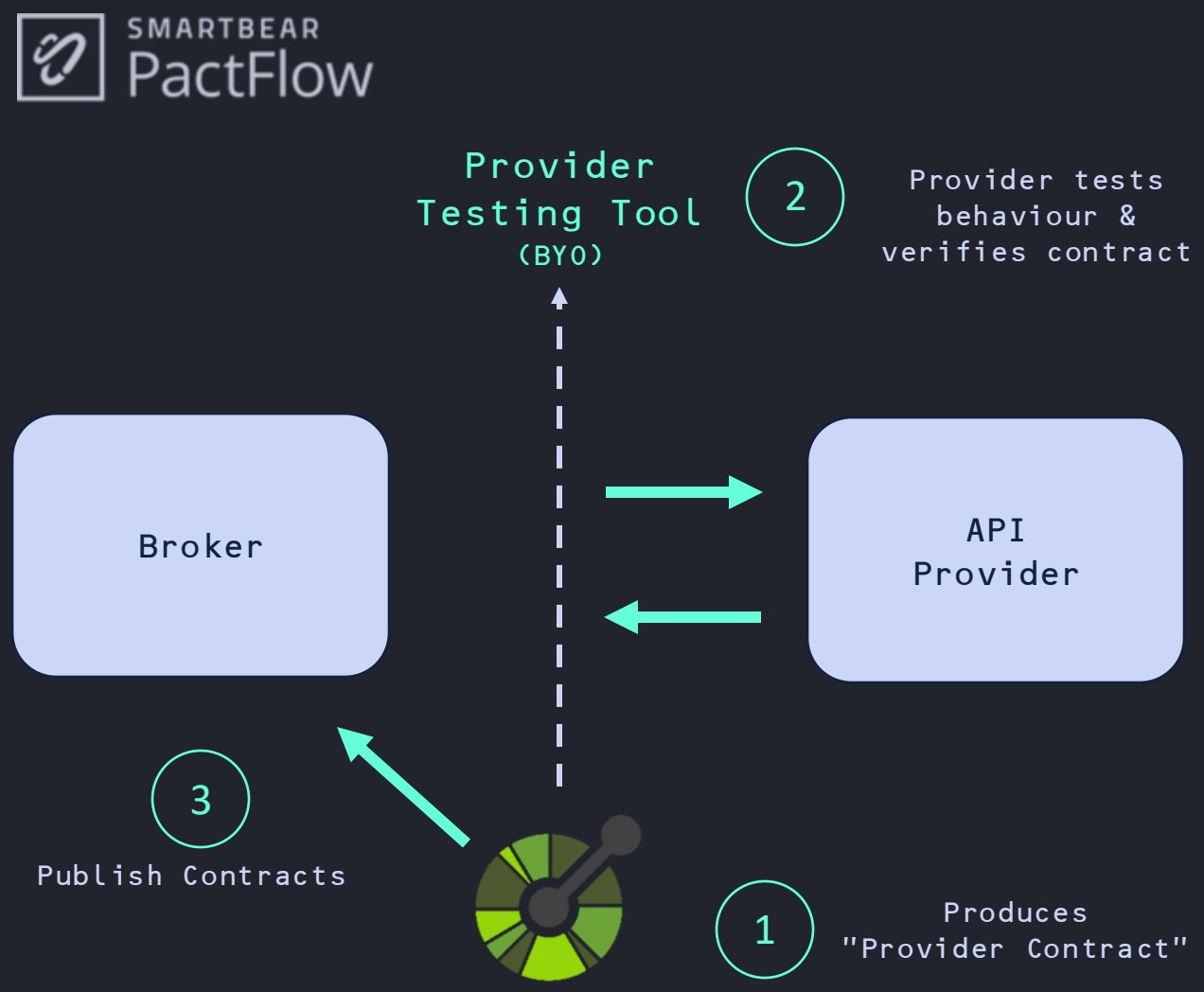
BDCT – How is works



SMARTBEAR

/* walk through */

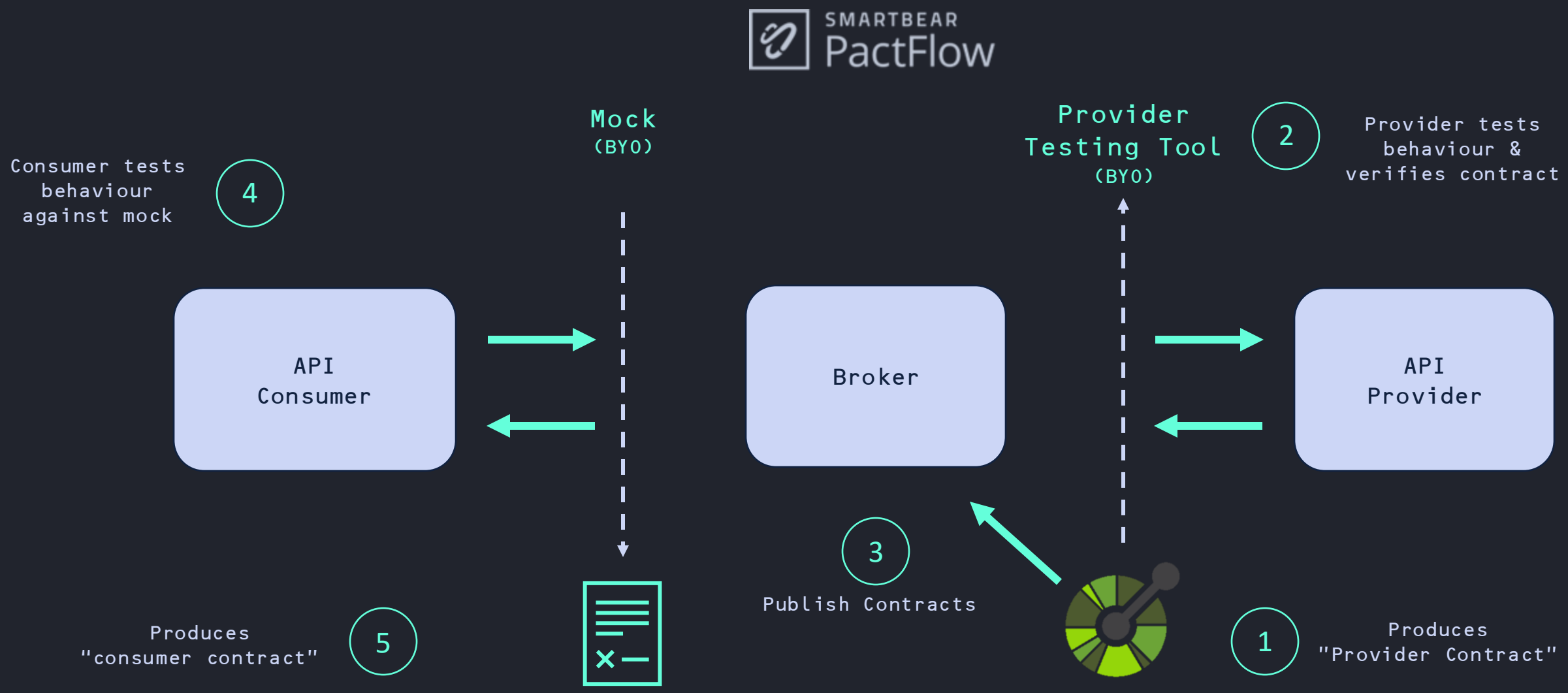
BDCT – How is works



SMARTBEAR

/* walk through */

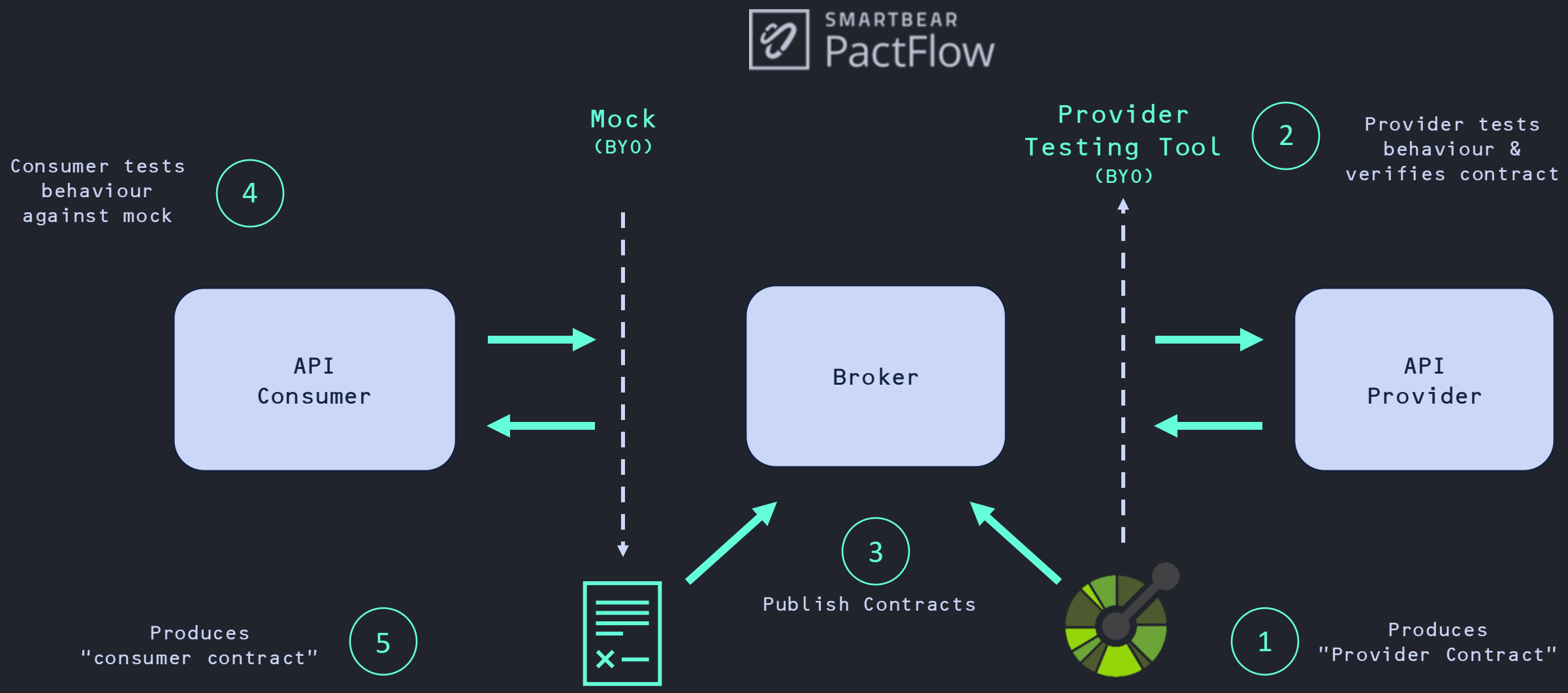
BDCT – How is works



SMARTBEAR

/* walk through */

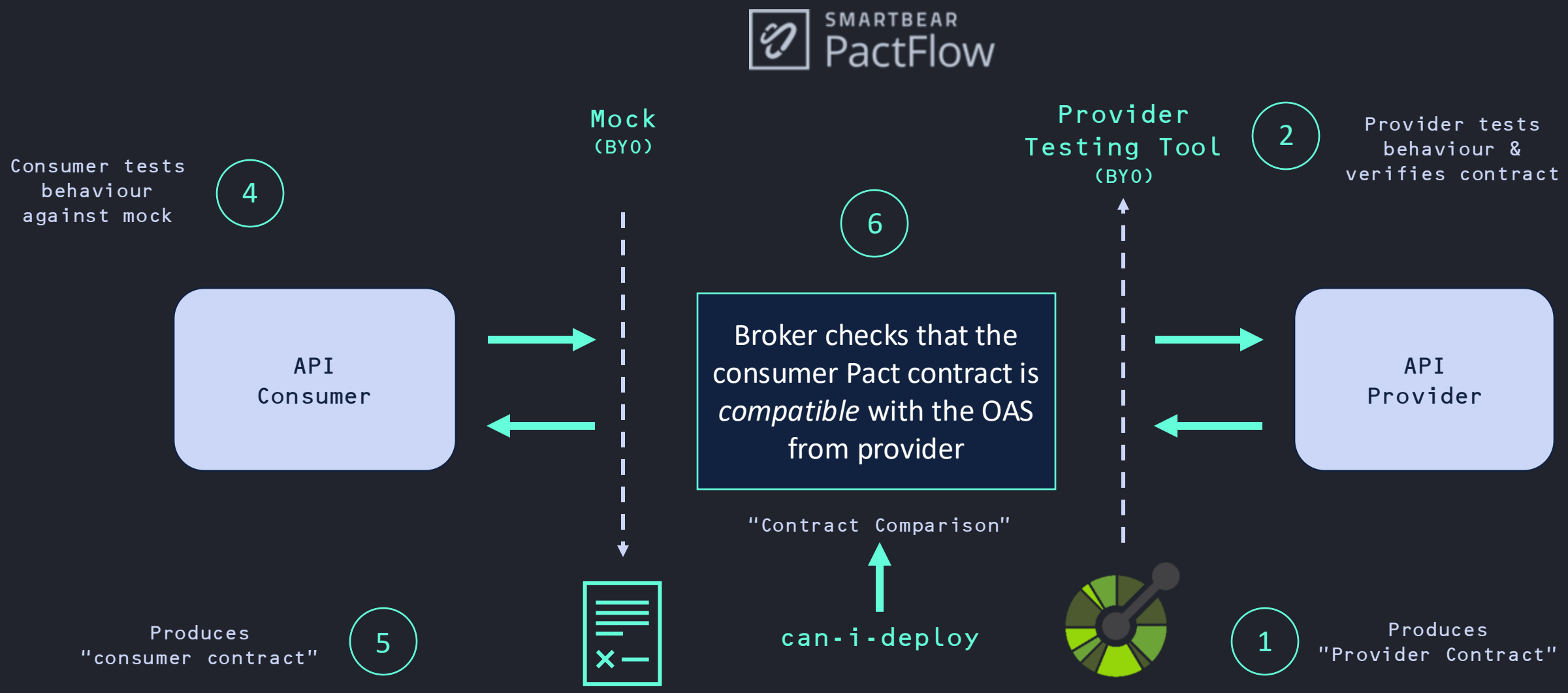
BDCT – How is works



SMARTBEAR

/* walk through */

BDCT – How is works

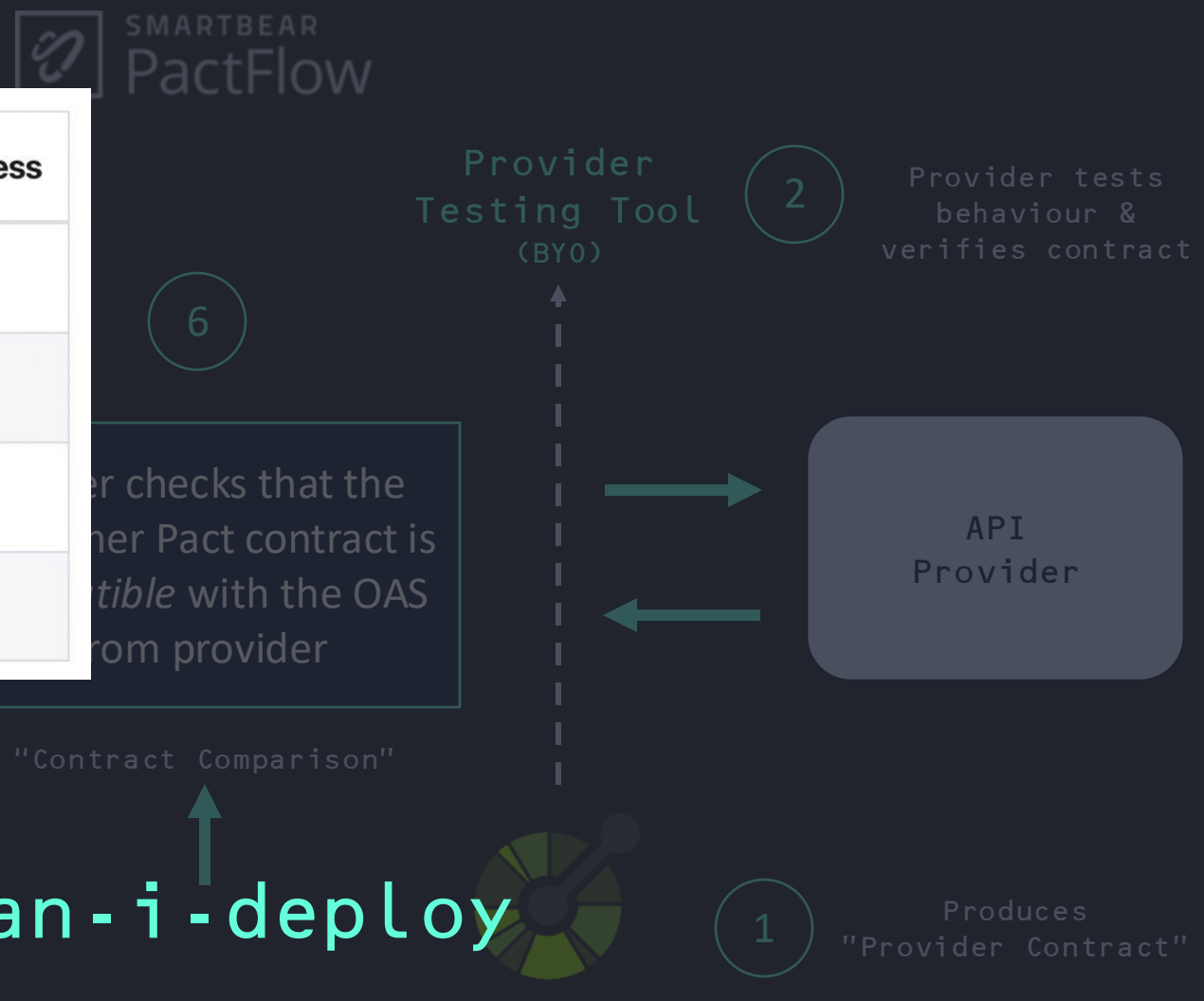


SMARTBEAR

/* walk through */

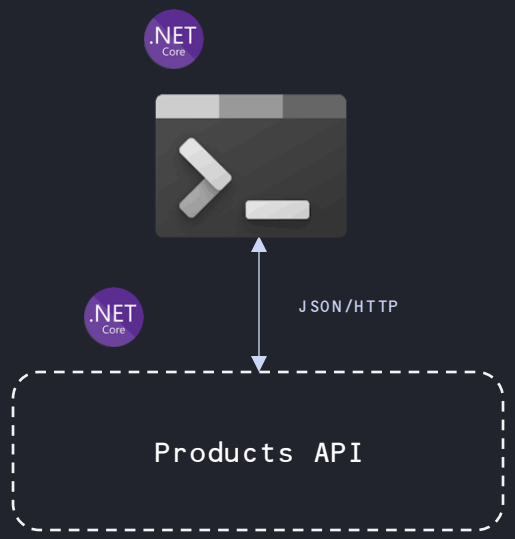
BDCT – How is works

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



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



Swagger Editor

File ▾ Edit ▾ Insert ▾ Generate Server ▾ Generate Client ▾ About ▾

Try our new Editor ↗

```yaml
openapi: 3.0.3
info:
  title: Products API
  description: |
    A sample Products API to demonstrate Bi-Directional Contract Testing for ASC 2022
  version: 1.0.0
paths:
  /Products:
    get:
      summary: Retrieve a list of products
      description: Get a list of products available
      tags:
        - Products
      responses:
        '200':
          description: Success
          content:
            text/plain:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/Product'
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/Product'
            text/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/Product'
  /Products/{id}:
    get:
      summary: Retrieve details on a specific product
      description: Get the full details on a particular product from the catalog
      tags:
        - Products
      parameters:
        - name: id
          in: path
          required: true
          schema:
            type: integer
            format: int32
      responses:
        '200':
          description: Success
          content:
            text/plain:
              schema:
                $ref: '#/components/schemas/Product'
            application/json:
              schema:
                $ref: '#/components/schemas/Product'
            text/json:
              schema:
                $ref: '#/components/schemas/Product'
        '204':
          description: Success
  /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

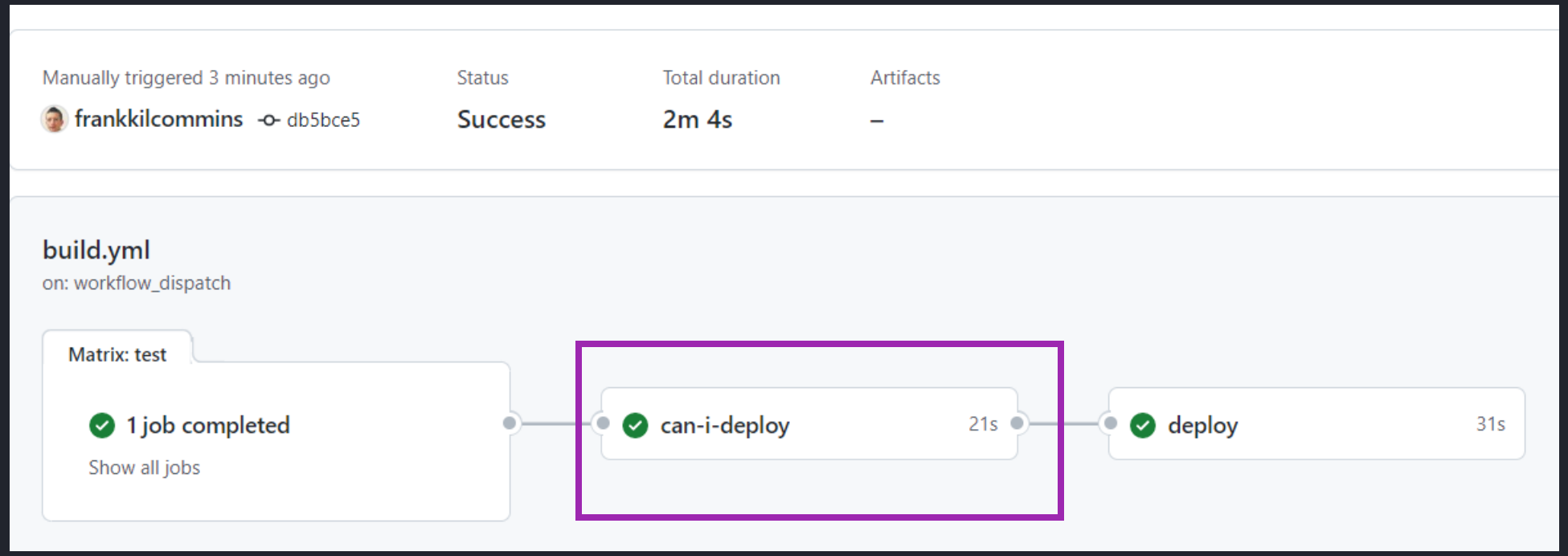
Product ›

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
```

SMARTBEAR

Provider – CI using GitHub Actions



SMARTBEAR

PactFlow – Provider contract published to broker

??? pactflow-example-bi-directional-provider-dotnet

Unknown

CONSUMER VERSION
N/A

BRANCH
N/A

ENVIRONMENTS
N/A

TAGS
N/A

PROVIDER VERSION
b5bce5-main+b5bce5

BRANCH
main

ENVIRONMENTS
Production

TAGS
N/A

PUBLISHED AT
3 minutes ago

VIEW CONTRACT

OVERVIEW CONTRACTS

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 OAS3

Products

GET /Products

GET /Products/{id}

DELETE /Products/{int}

Consumer – Test using xunit and wiremock

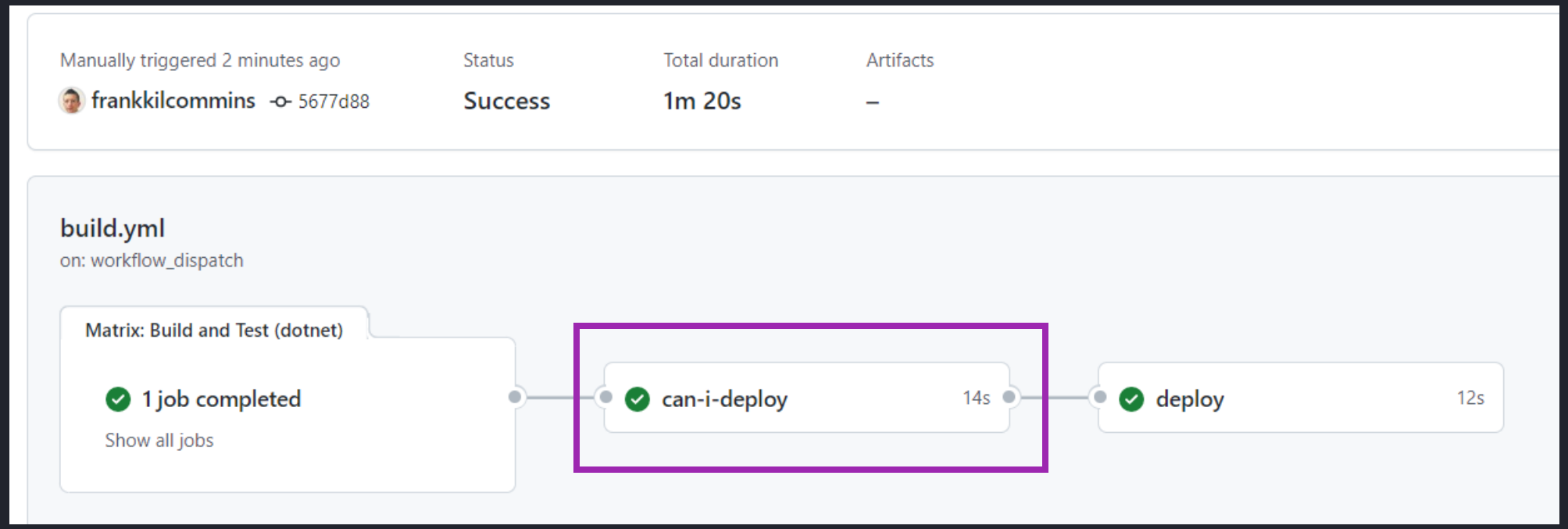
```
[Fact]
0 references | Run Test | Debug Test
public async Task GetProduct_WhenCalledWithInvalidID_ReturnsError()
{
    // Arrange
    var server = WireMockServer.Start();
    String serverUrl = server.Urls[0] + "/";
    server
        .WithConsumer(consumer)
        .WithProvider(provider)
        .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"
  }
}
```


SMARTBEAR

Consumer – Another CI using GitHub Actions



SMARTBEAR

What's New

Can I Deploy

API

OVERVIEW | NETWORK DIAGRAM | MATRIX | WEBHOOKS | CONTRACTS

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

DEPLOYED ENVIRONMENTS
Production

BRANCH
main

Provider Details

PROVIDER VERSION
20be87-main+20be87

More provider details

RELEASED ENVIRONMENTS
N/A

TAGS
N/A

PUBLISHED AT
2 hours ago

DEPLOYED ENVIRONMENTS
Production

BRANCH
main

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"

Try our new Editor ↗

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 ›

Remove delete method Build #8

Summary

Jobs

- test (3.1.x)
- can-i-deploy
- deploy

Triggered via push 1 minute ago
frankkilcommins pushed e20be87 main

Status	Total duration	Artifacts
Success	1m 32s	–

build.yml
on: push

Matrix: test
1 job completed
Show all jobs

can-i-deploy 16s

deploy 15s

Pactflow | Remove delete method · frankki | Build · frankkilcommins/example

smartbear-frank.pactflow.io/contracts/bi-directional/provider/pactflow-example-bi-directional-provider-dotnet/version/20be87-main%2B20be87/consumer/pactflow-example-bi-directi...

PACTFLOW

What's New

Can I Deploy

API

OVERVIEW | NETWORK DIAGRAM | MATRIX | WEBHOOKS | CONTRACTS

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

Consumer Details

CONSUMER VERSION
5677d880a56a136df9ba0e8fddff1c5d490f8f7f

PUBLISHED AT
13 minutes ago

BRANCH
main

More consumer details

Provider Details

PROVIDER VERSION
20be87-main+20be87

PUBLISHED AT
2 minutes ago

BRANCH
main

More provider details

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

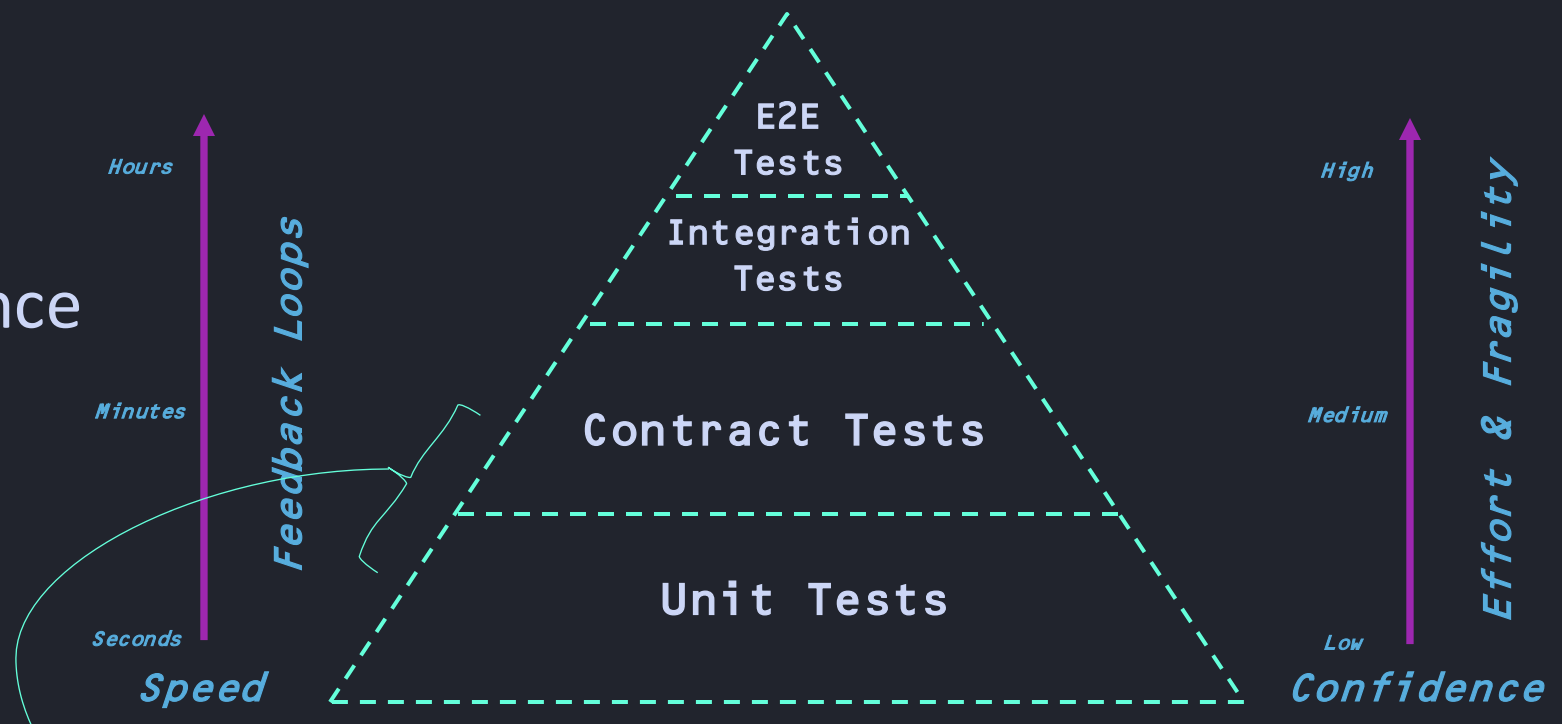
GET /Products

GET /Products/{id}

/* 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



The test pyramid

- ☑ Cheap
- ☑ Fast
- ☑ Reliable
- ☑ Targeted

SMARTBEAR

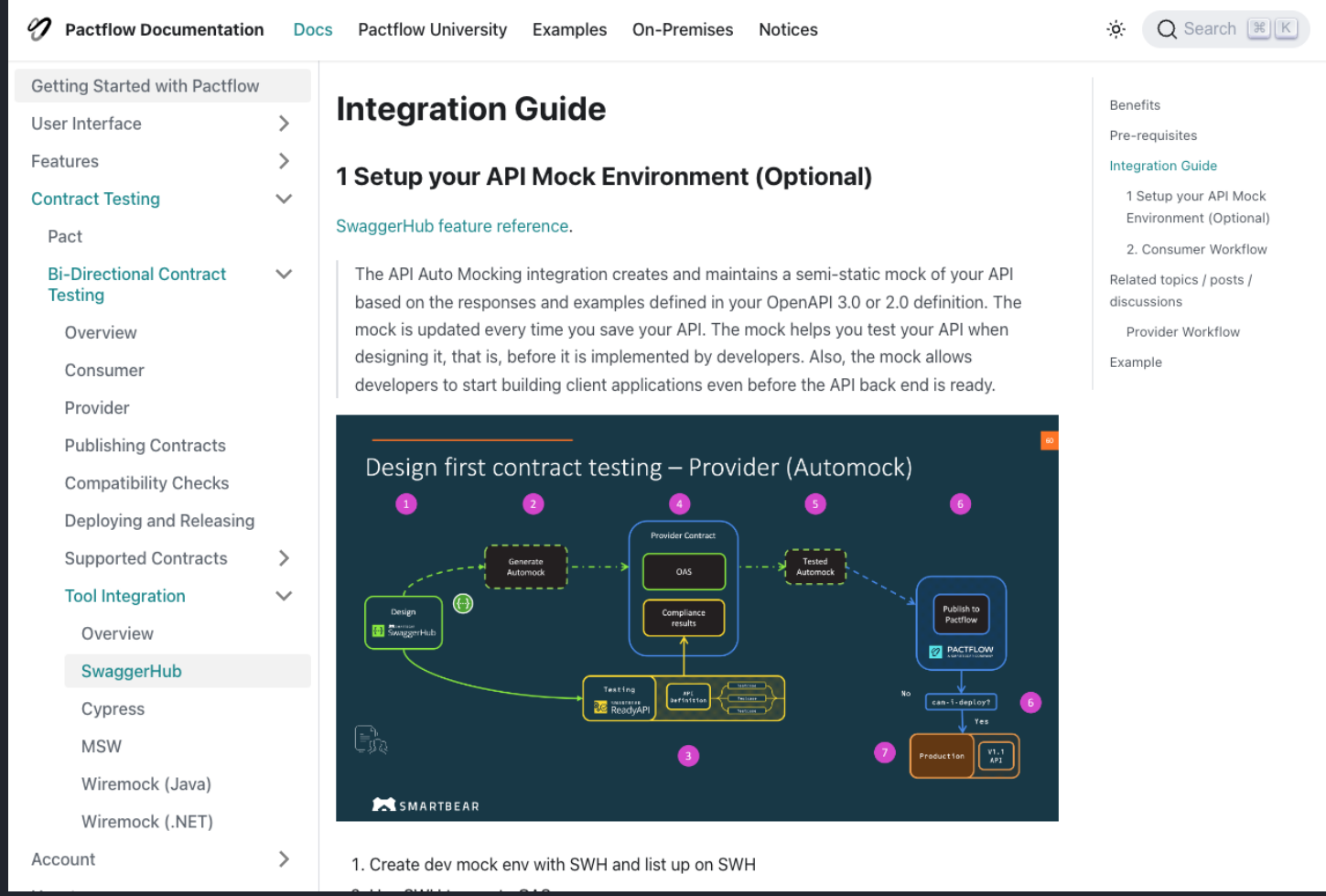
/* Problem addressed: lack of visibility into how consumers are using an API */

Benefits for API Management Experience

- 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

Give it a try

Head to go.pactflow.io/design-first



FS9U

ProxyHands

SMARTBEAR

› Connect:

@fkilcommins

@frank-kilcommins

frank.kilcommins@smartbear.com

Q&A

Thanks

FS9U

ProxyHands

LangDev CON 2024

Seville 17-19 October, 2024

https://langdevcon.org