



How to never use WebDriver again

Modern Javascript CI



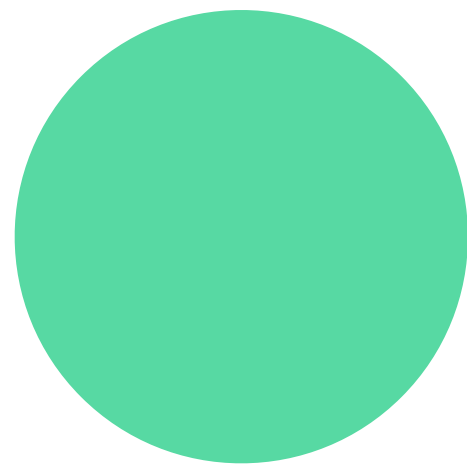
DAVID PIPER | FRONTEND DEVELOPER | ATASSIAN

INTRODUCTION

**Comprehensive
Javascript test
framework
designed to
replace WebDriver**

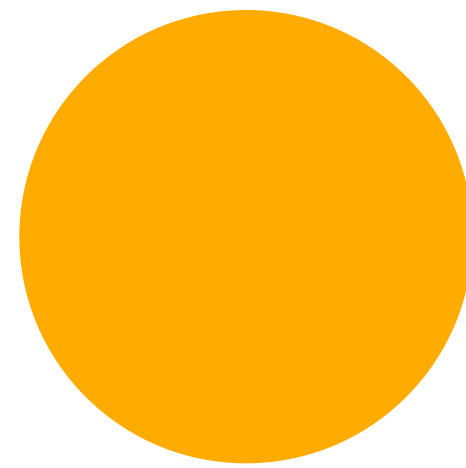


Three categories of CI tools



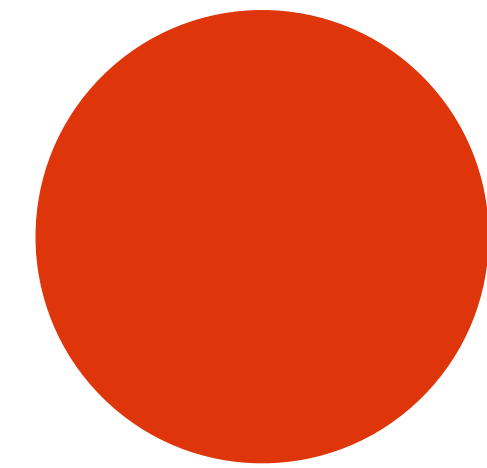
Essential

Don't leave home without these tools and services



Nice to have

Benefits usually outweigh the costs of maintenance

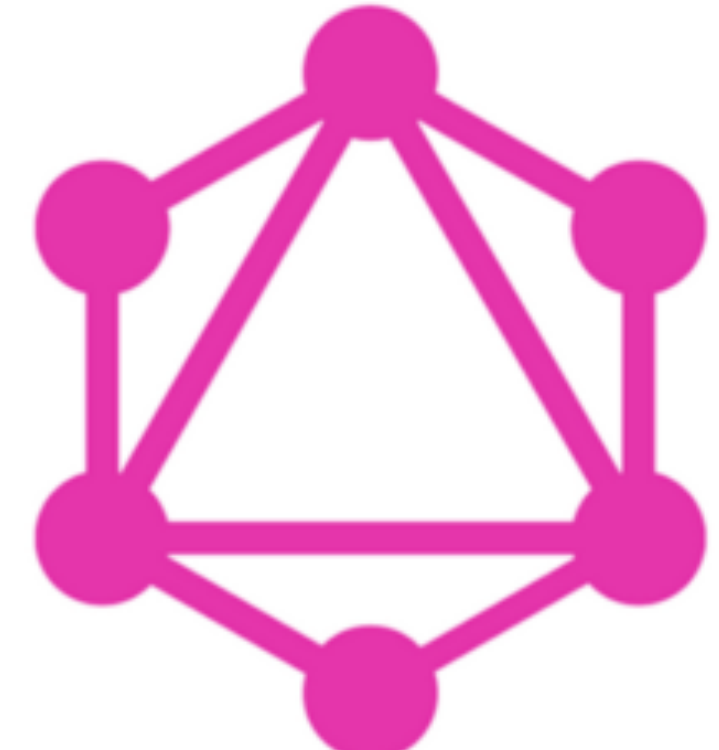
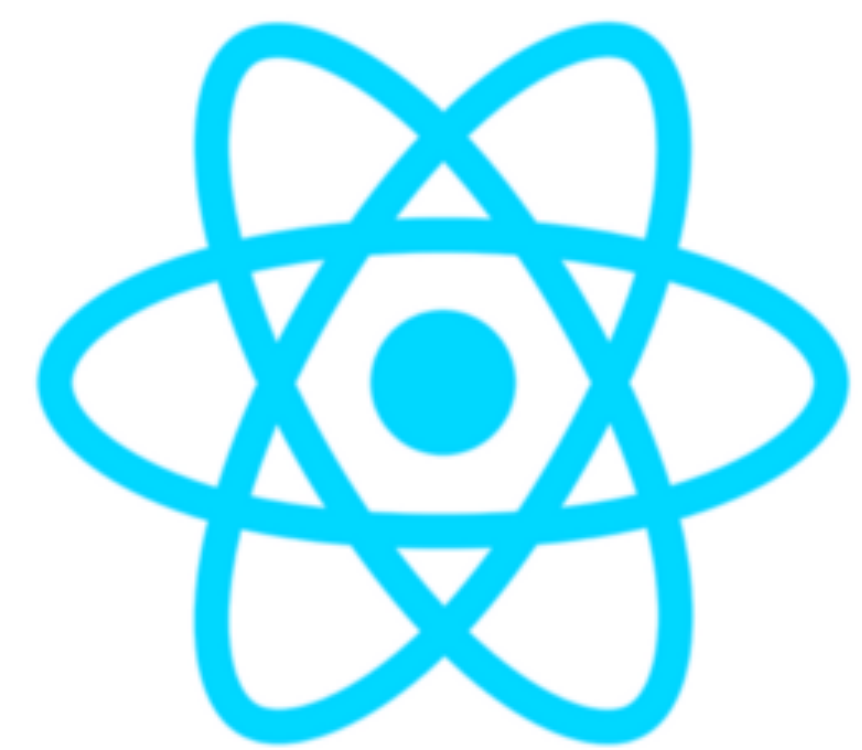


Optional

If you won't commit to maintaining them, don't.

What aren't we going to talk about?

MODERN JAVASCRIPT CI



MODERN JAVASCRIPT CI



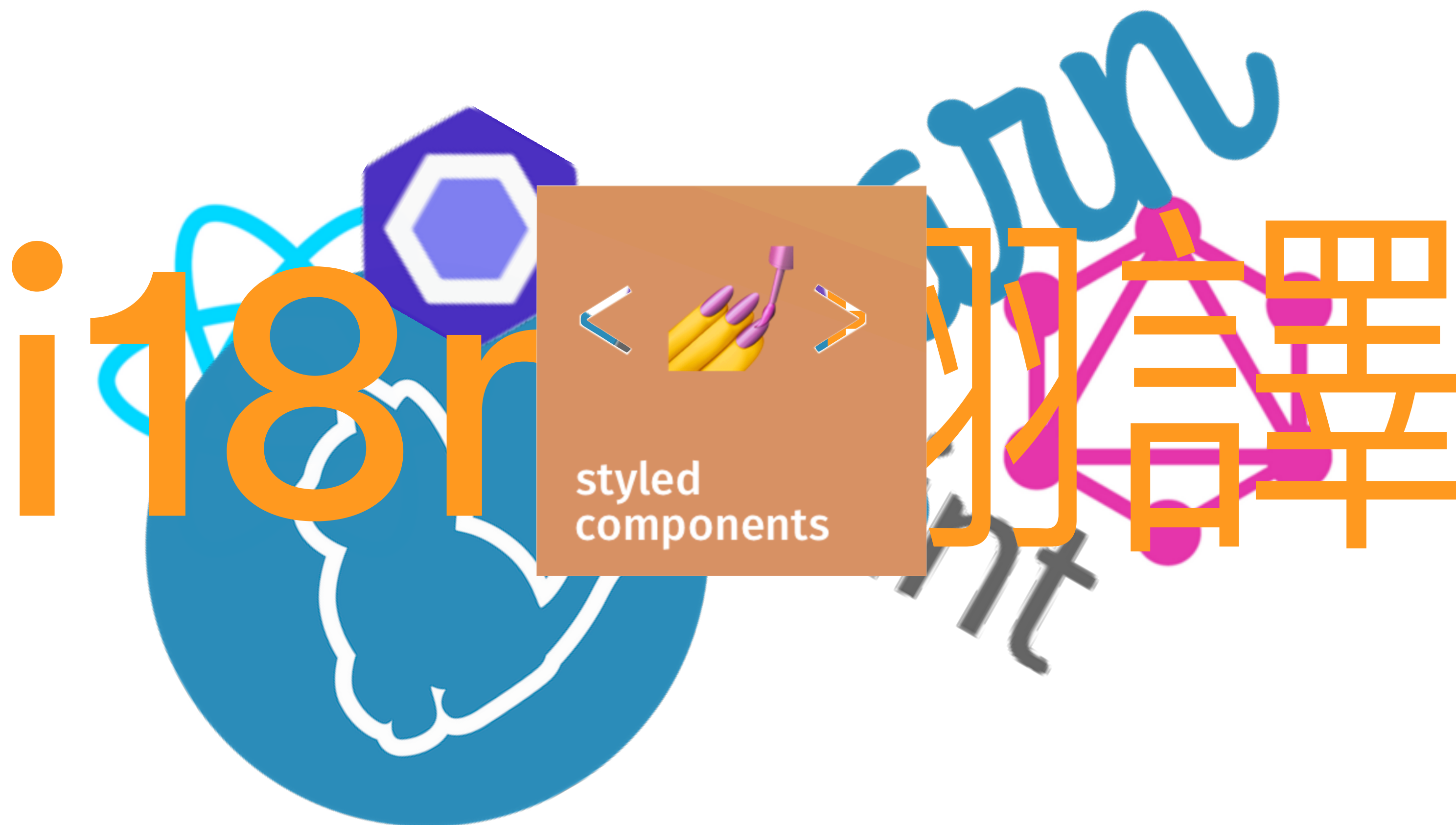
MODERN JAVASCRIPT CI



MODERN JAVASCRIPT CI



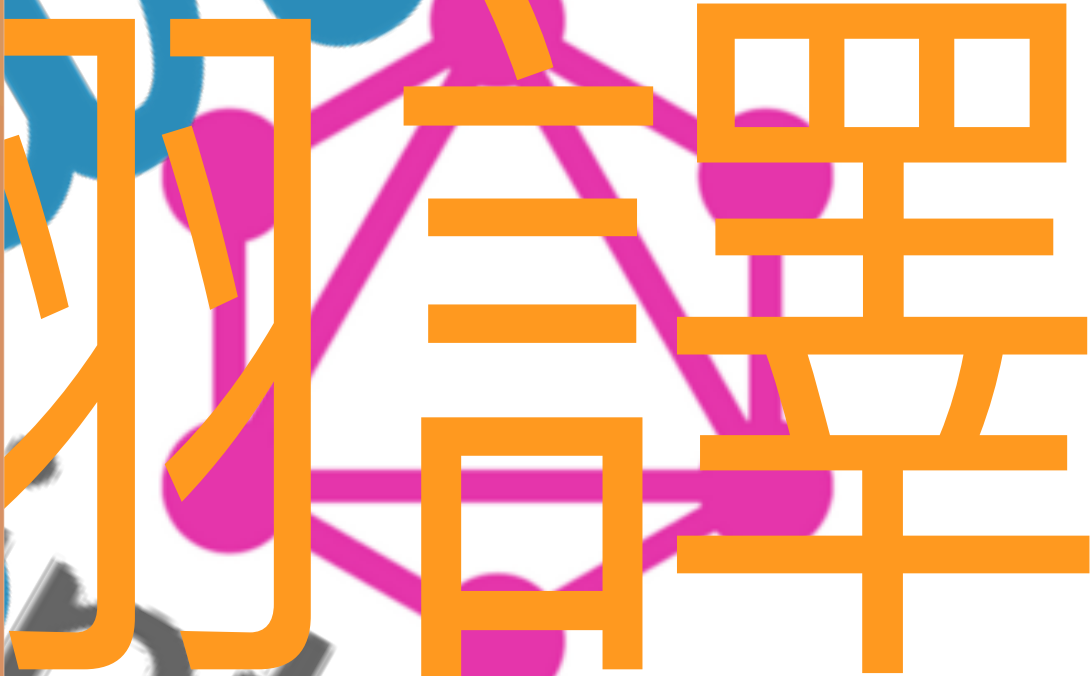
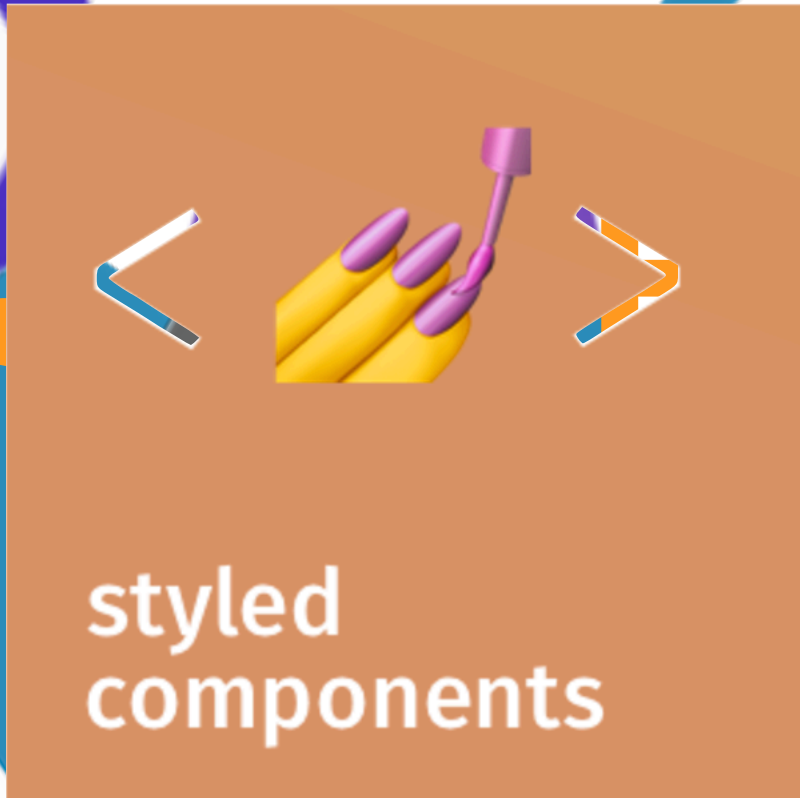
MODERN JAVASCRIPT CI



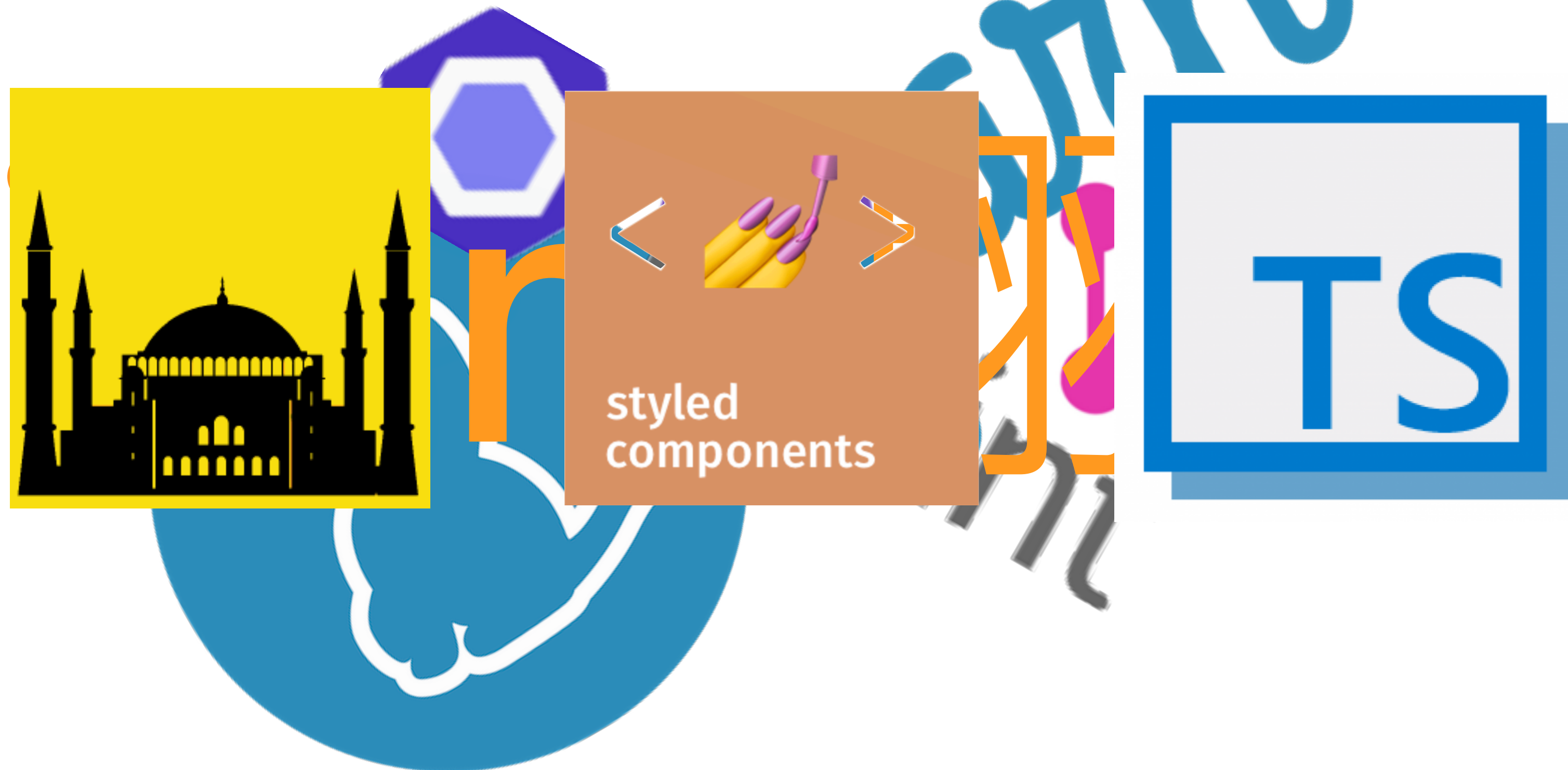
MODERN JAVASCRIPT CI



r



MODERN JAVASCRIPT CI



MODERN JAVASCRIPT CI





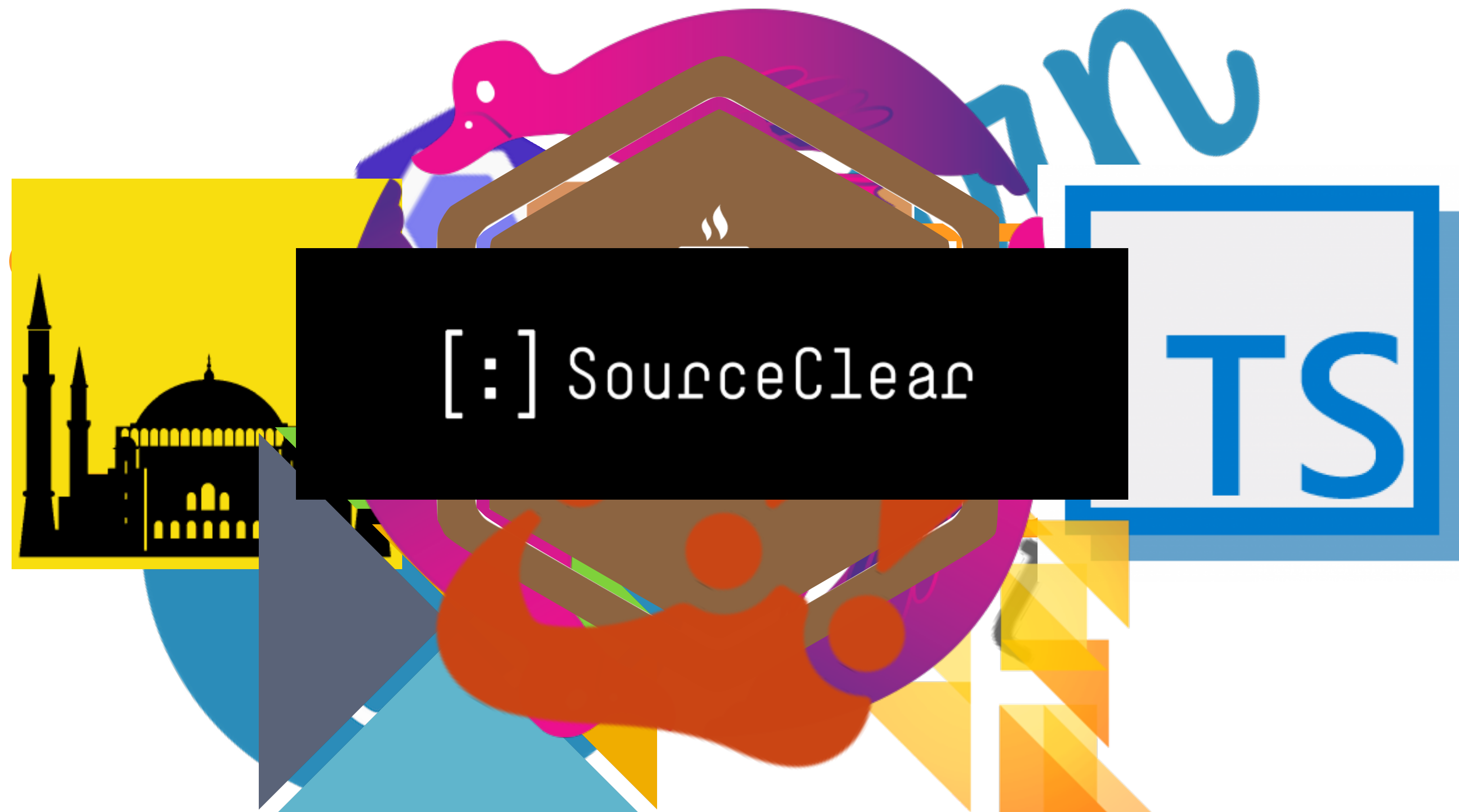
MODERN JAVASCRIPT CI



MODERN JAVASCRIPT CI



MODERN JAVASCRIPT CI



What are we going to talk about?

JAVASCRIPT CI?



OBSOLETE JAVASCRIPT CI



MODERN JAVASCRIPT CI



UI component tests



enzyme



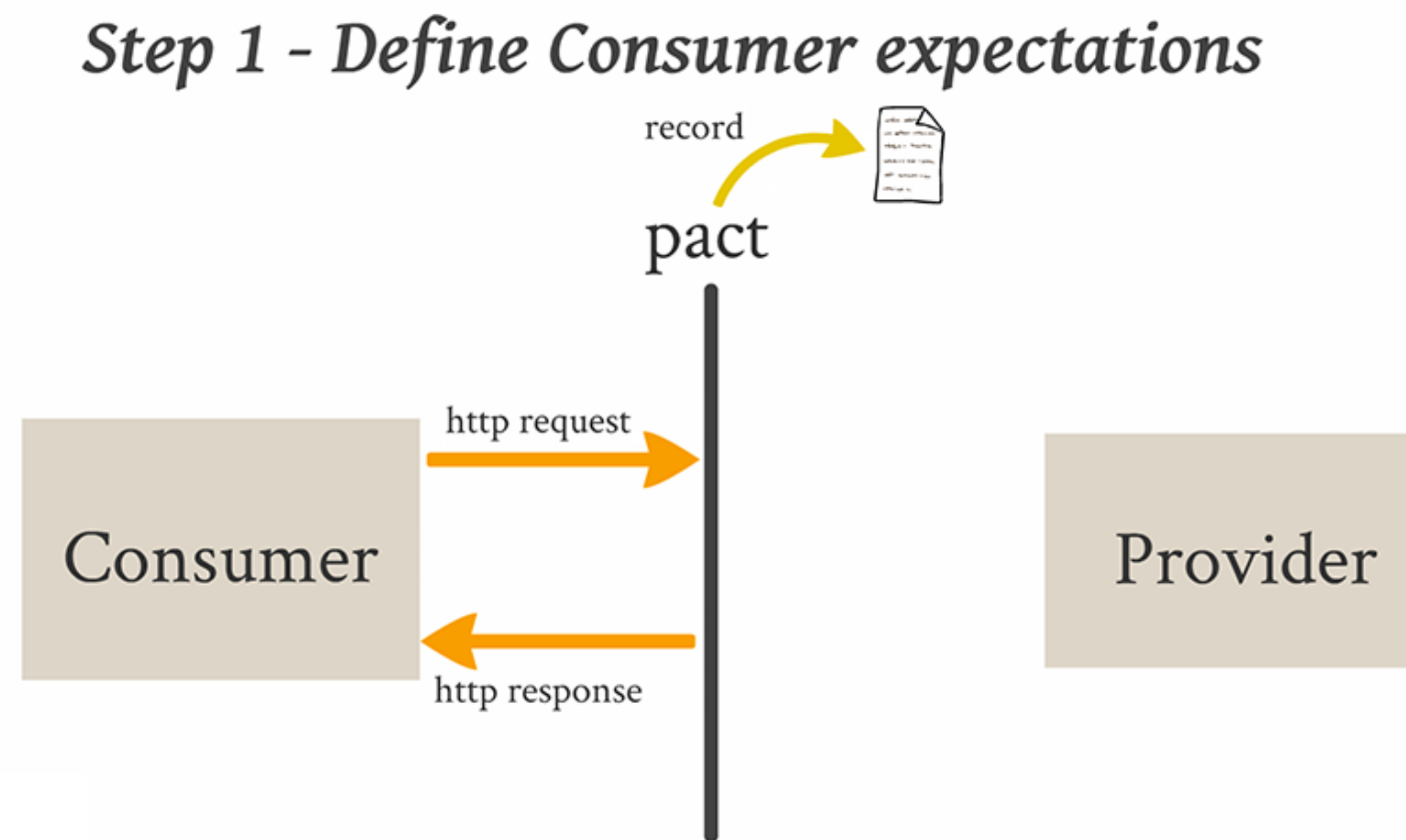
enzyme

+

jsdom

2 Pact (contract) tests

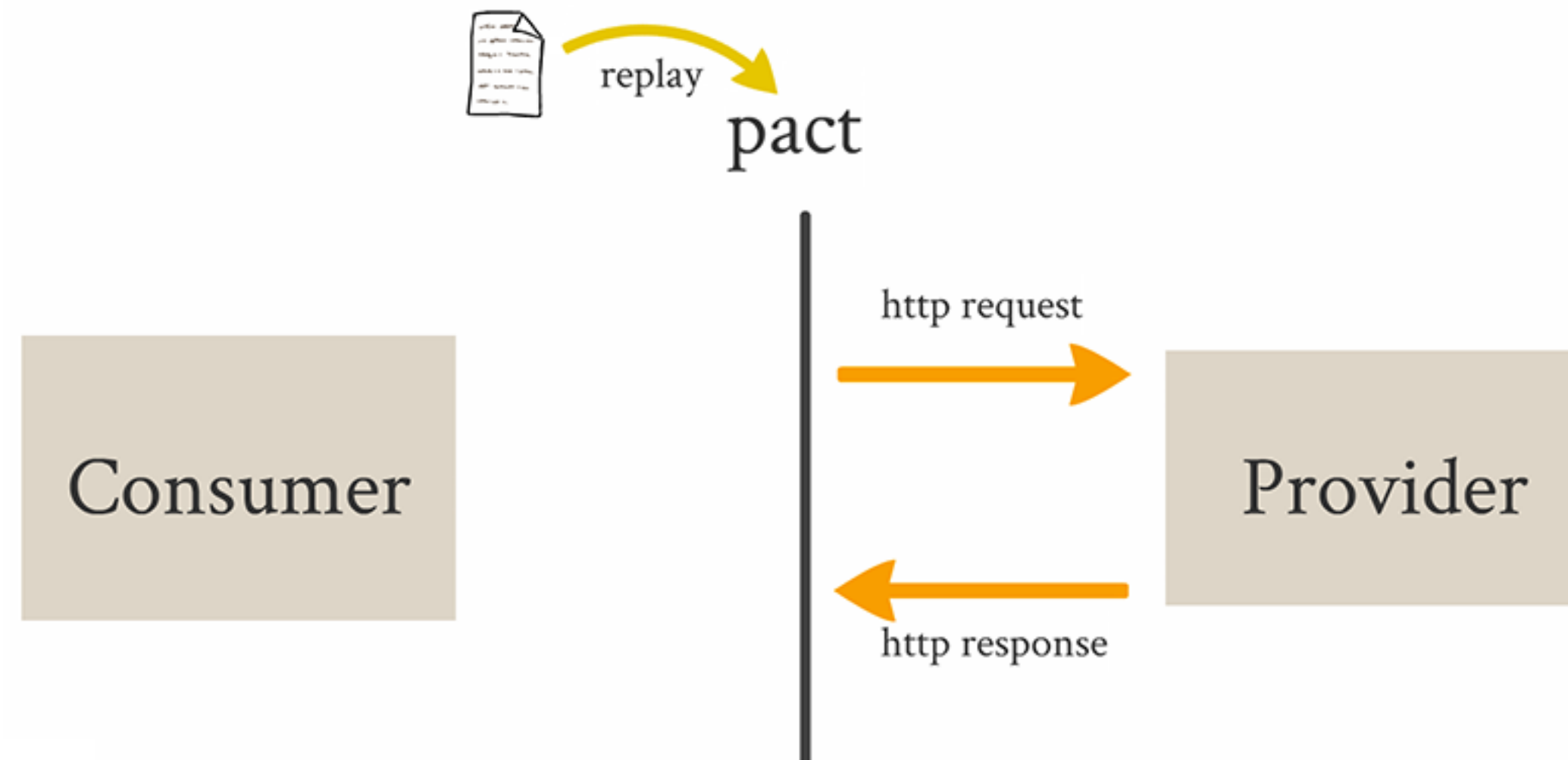
TRADITIONAL PACT FLOW: CONSUMER



<http://dius.com.au/2016/02/03/microservices-pact/>

TRADITIONAL PACT FLOW: PROVIDER

Step 2 - Verify expectations on Provider



<http://dius.com.au/2016/02/03/microservices-pact/>

Problems with the traditional flow



Consumer driven

Consumers cannot 'break'.
There is no hard requirement
for the Consumer to conform
to the Provider.



Authentication

No support intended for
authentication. You don't
want to bake credentials in.
Must be injected in the
Provider tests.



Live Provider

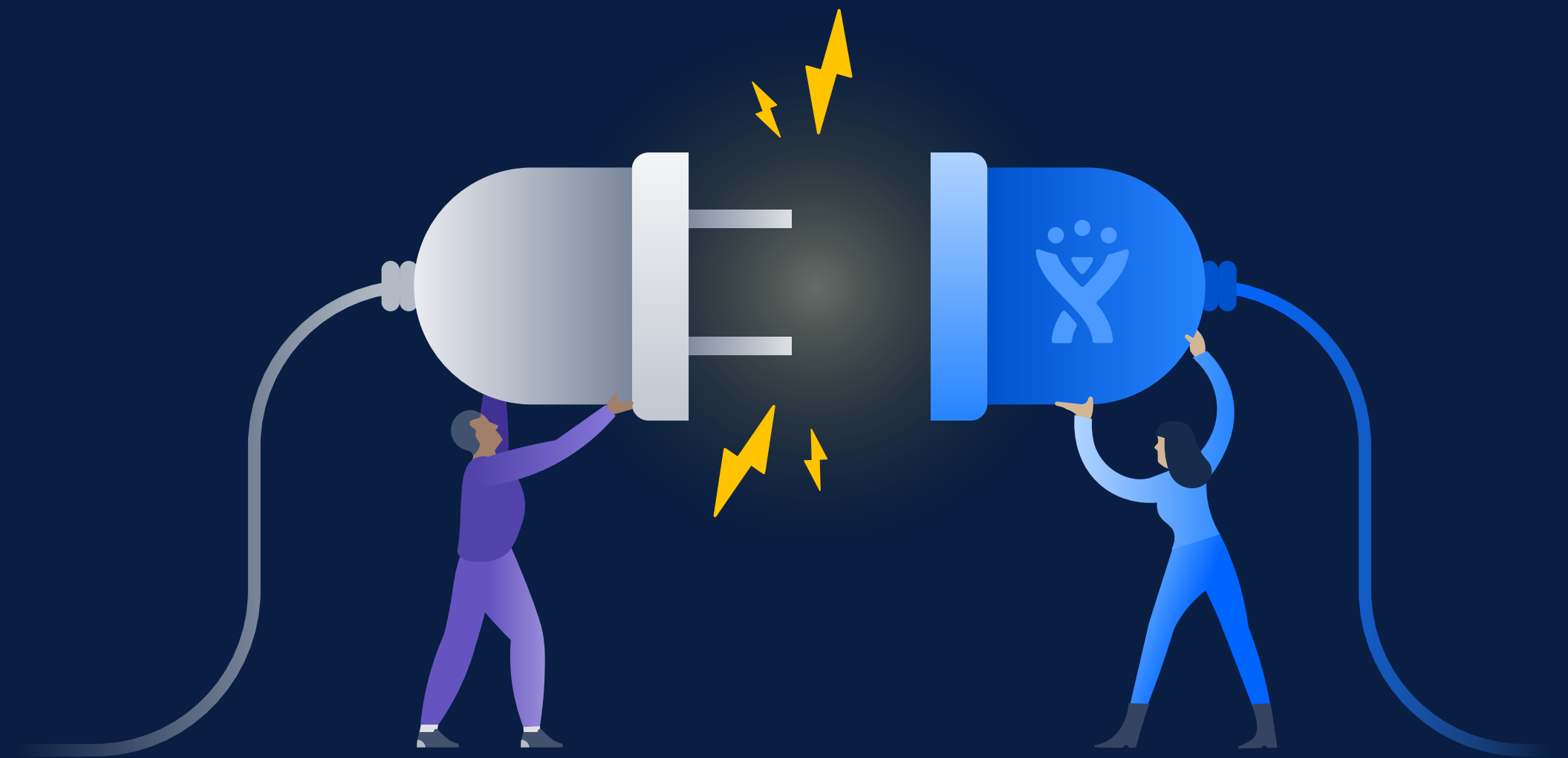
Either spin up the service in
CI, or test a live service. You
need to manage your test
data as well.



Swagger validation

SWAGGER SPECIFICATIONS

A complete* API
description,
published by the
Provider service

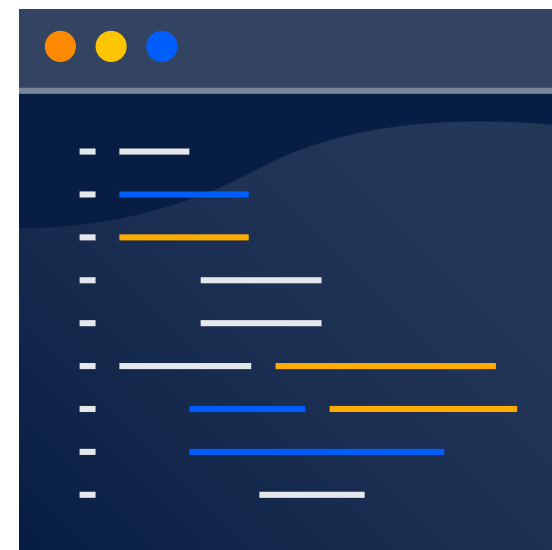


Ways to generate Swagger specs



Manually

Don't do this for large services.



Code comments

E.g: JavaDoc. Still a manual and error-prone process.



Code annotations

Make your tooling generate your Swagger for you

Remember these?



Consumer driven



Authentication



Live Provider

Consumer flow

Write Pact tests

Libraries are available for a large number of frontend projects:

docs.pact.io

Swagger validation

Verify that the API expectations in the Pact files can be met by the Provider's published spec:
bitbucket.org/atlassian/swagger-request-validator

Upload to Pact broker

This could be any file upload service, but there's an official one already built at:

github.com/pact-foundation/pact_broker

Provider flow

Pull latest pact file(s)

These will come from the Pact broker

Swagger validation

Verify that the API expectations in the Pact files can be met by the Provider's published spec:
bitbucket.org/atlassian/swagger-request-validator

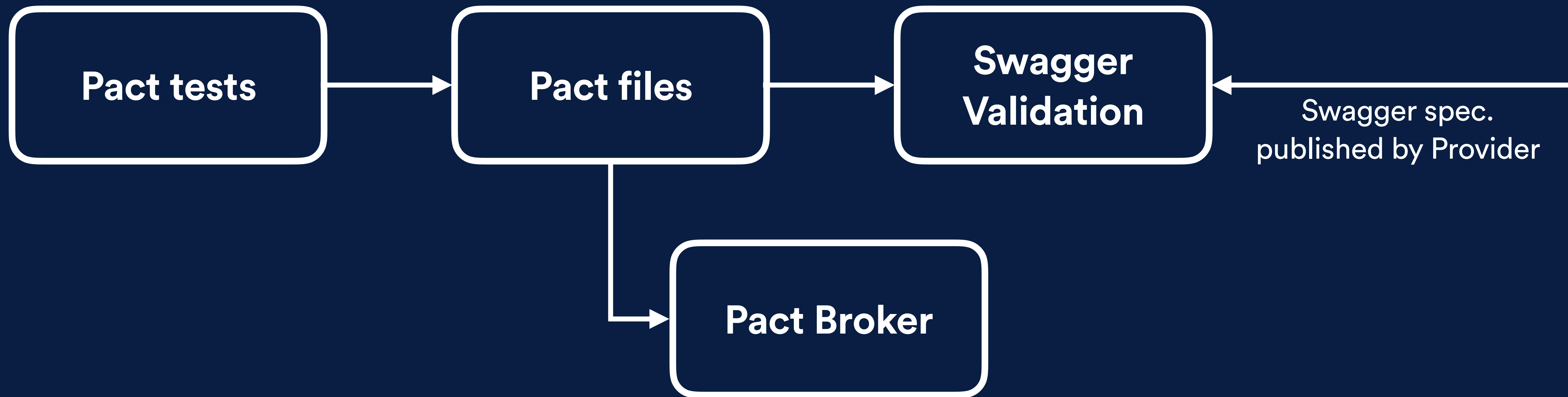
What do we have so far?

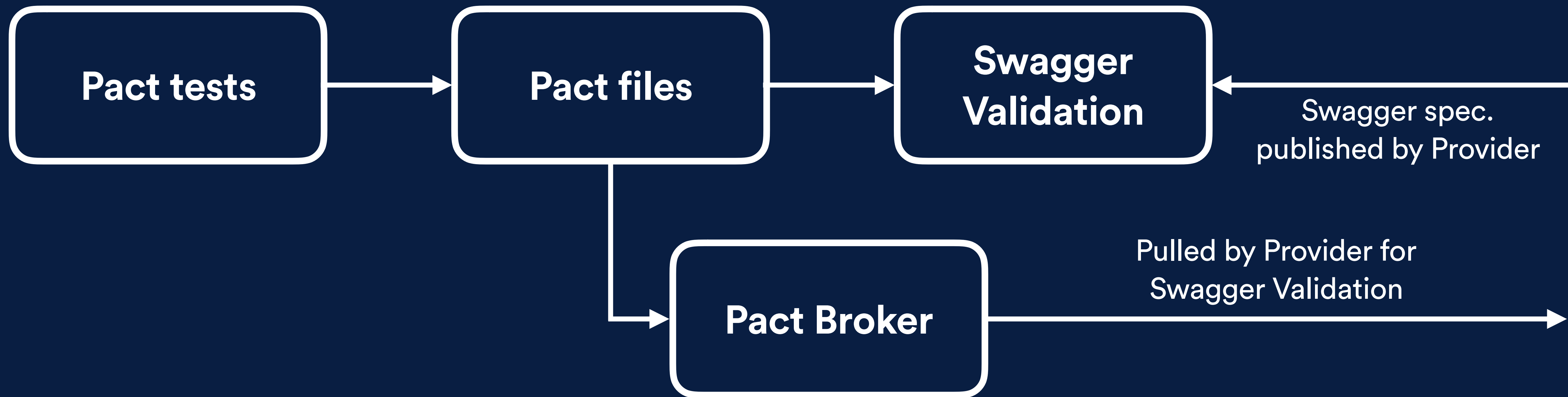
Pact tests



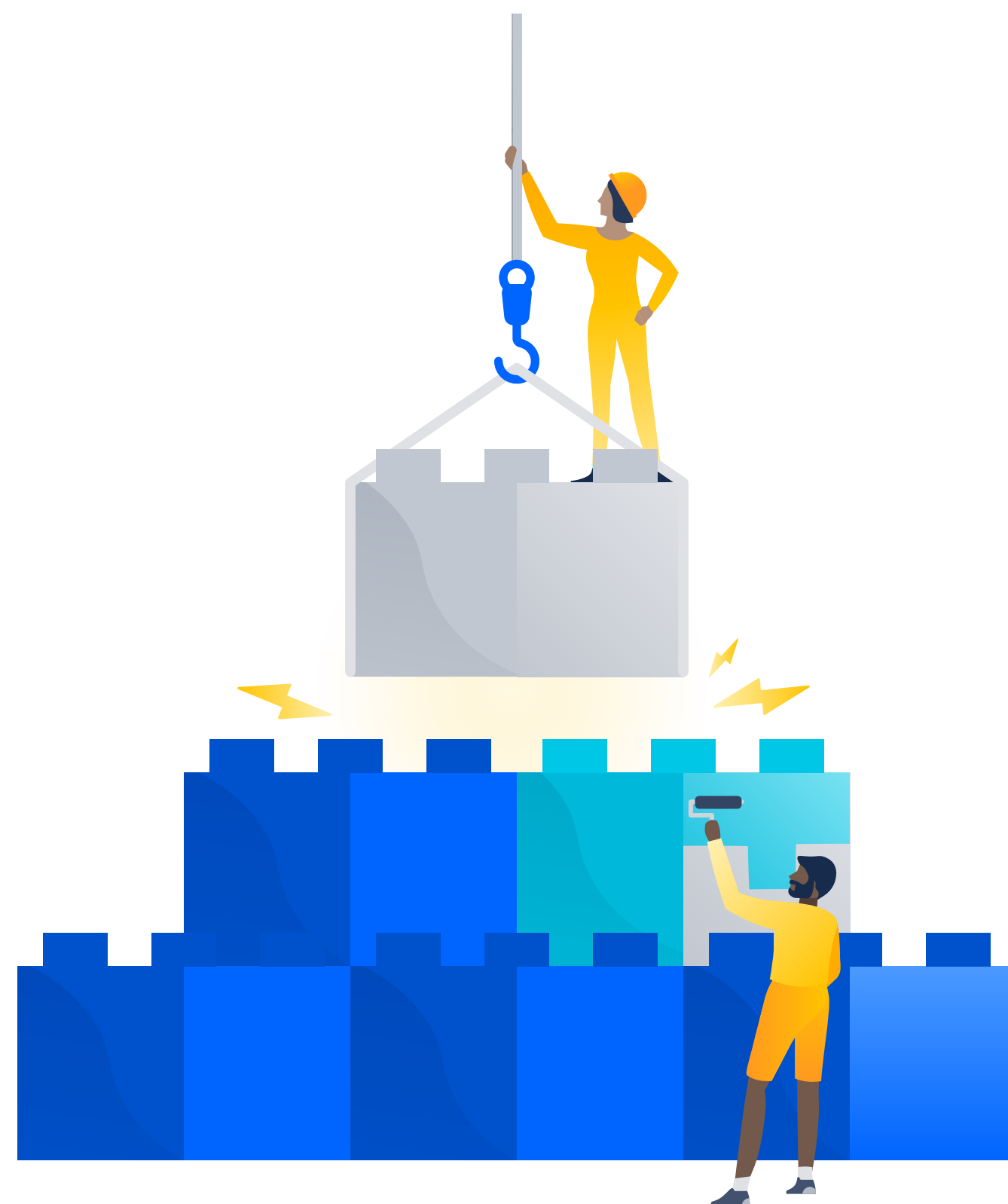








We're nearly there



MODERN JAVASCRIPT CI

4 Unpack

bitbucket.org/atlassianlabs/unpack

UNPACT

**A Javascript library that
guarantees your test mocks
are consistent with your Pact
expectations.**

UNPACT

```
const spec = {
  age: spec(24).type(),
  name: spec('David').regex('[a-z]+'),
  roles: spec([
    {
      title: spec('Developer').regex('.+'),
    },
    {
      title: spec('Quality Engineer').regex('.+'),
    },
  ]).array(),
};

export default spec; // Used to generate Pacts
export const mock = unpact(pactSpec); // Test mocks
```

UNPACT

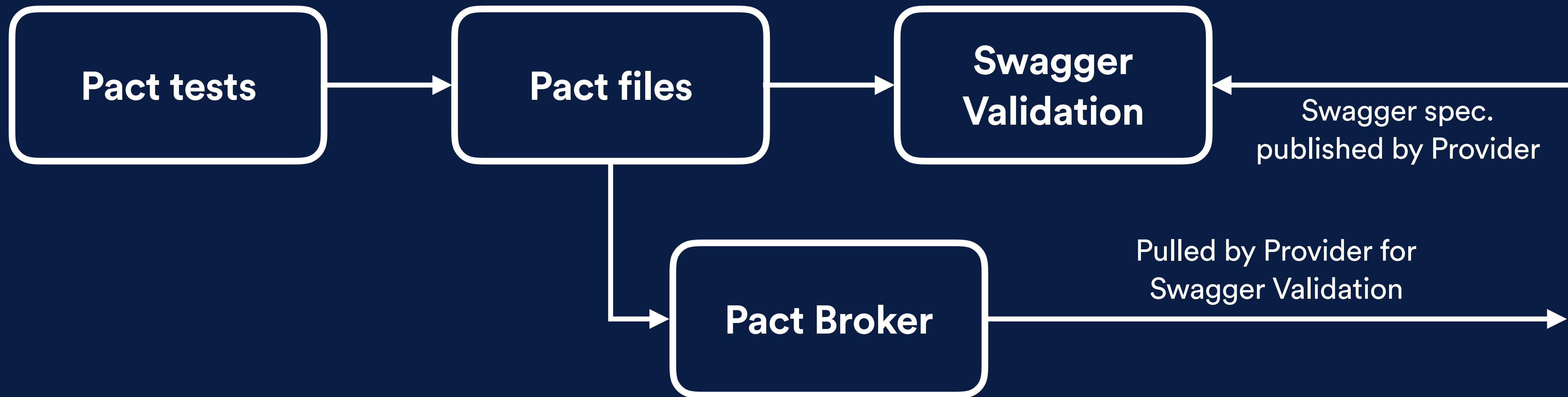
```
const spec = {
  age: spec(24).type(),
  name: spec('David').regex('[a-z]+'),
  roles: spec([
    {
      title: spec('Developer').regex('.+'),
    },
    {
      title: spec('Quality Engineer').regex('.+'),
    },
  ]).array(),
};

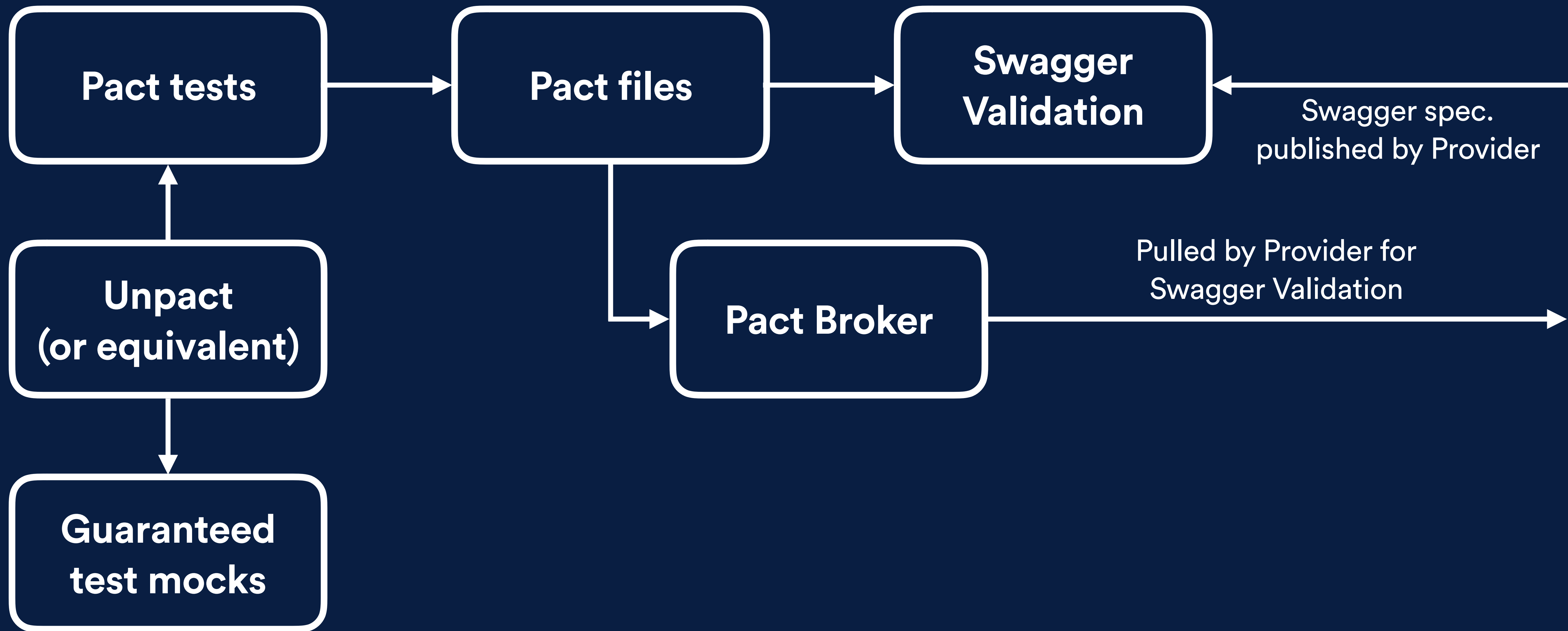
export default spec; // Used to generate Pacts
export const mock = unpact(pactSpec); // Test mocks
```

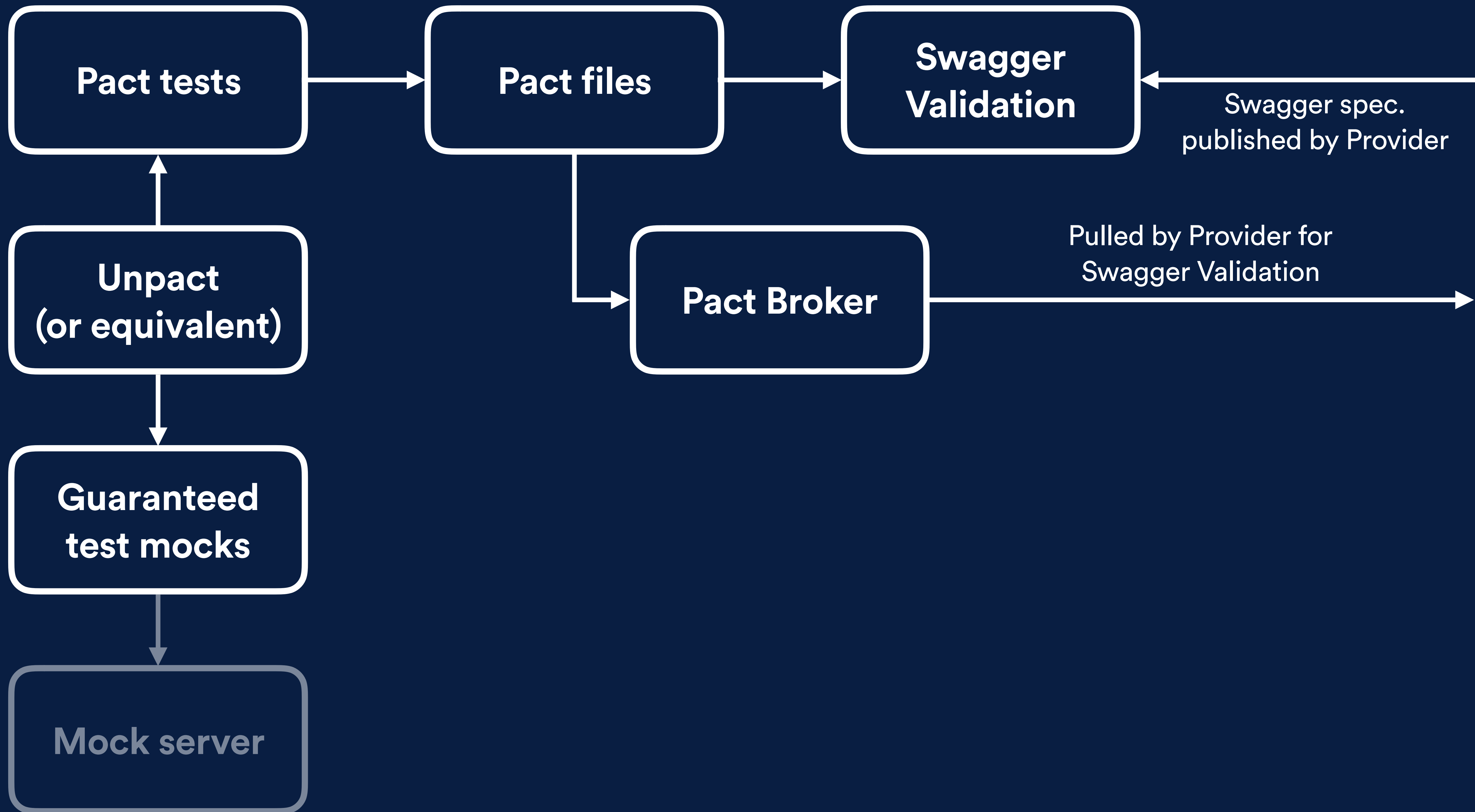

MODERN JAVASCRIPT CI

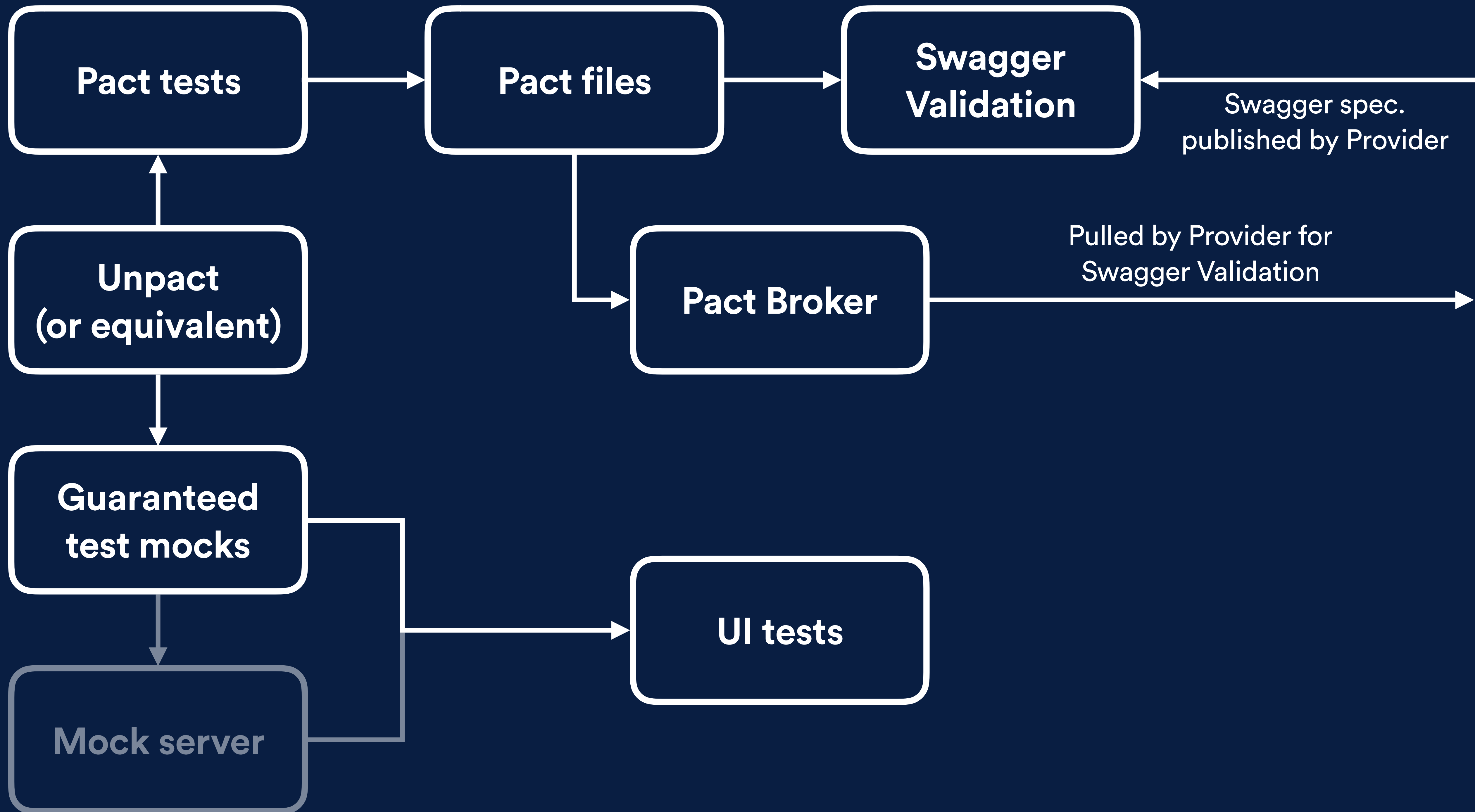


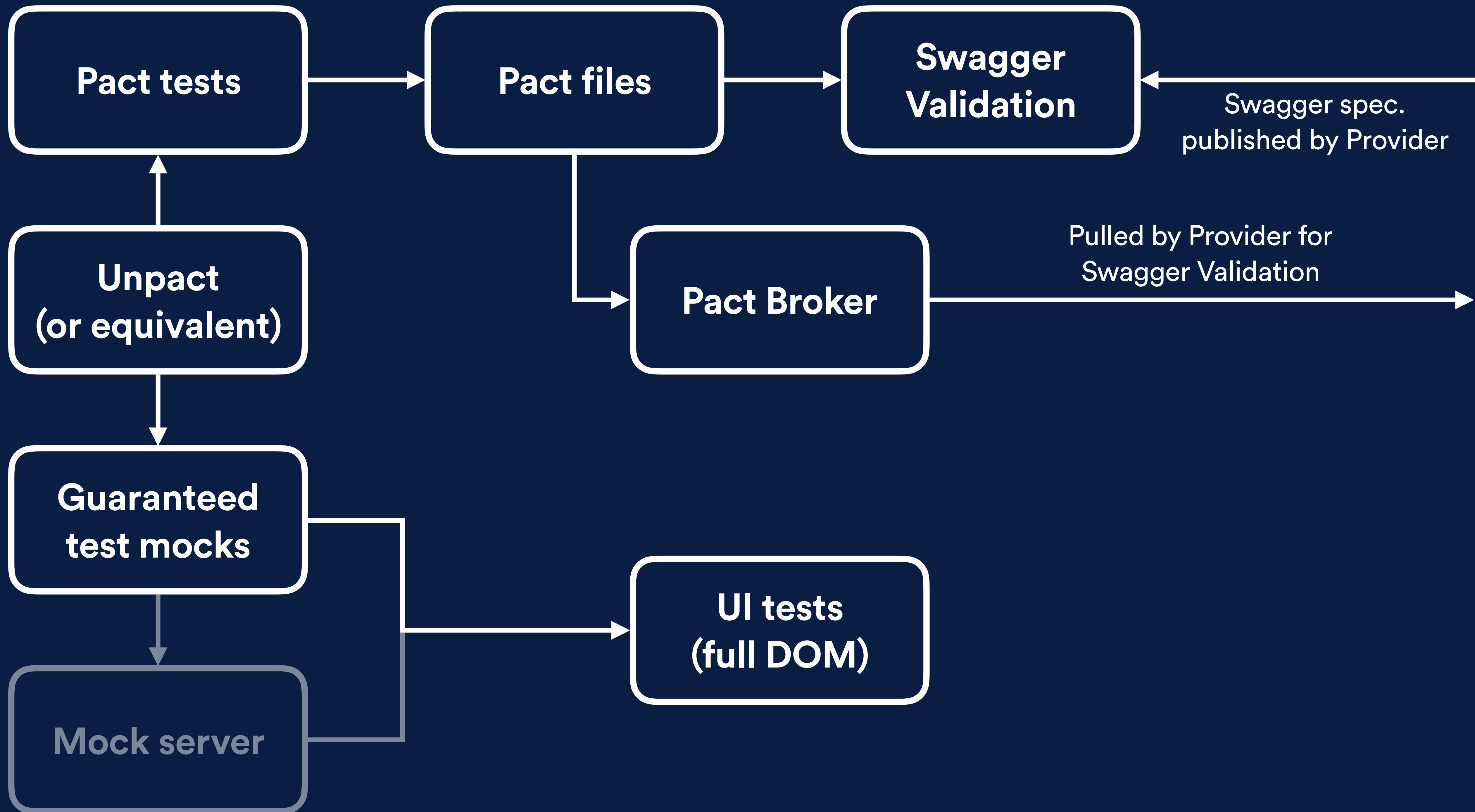
Integration tests

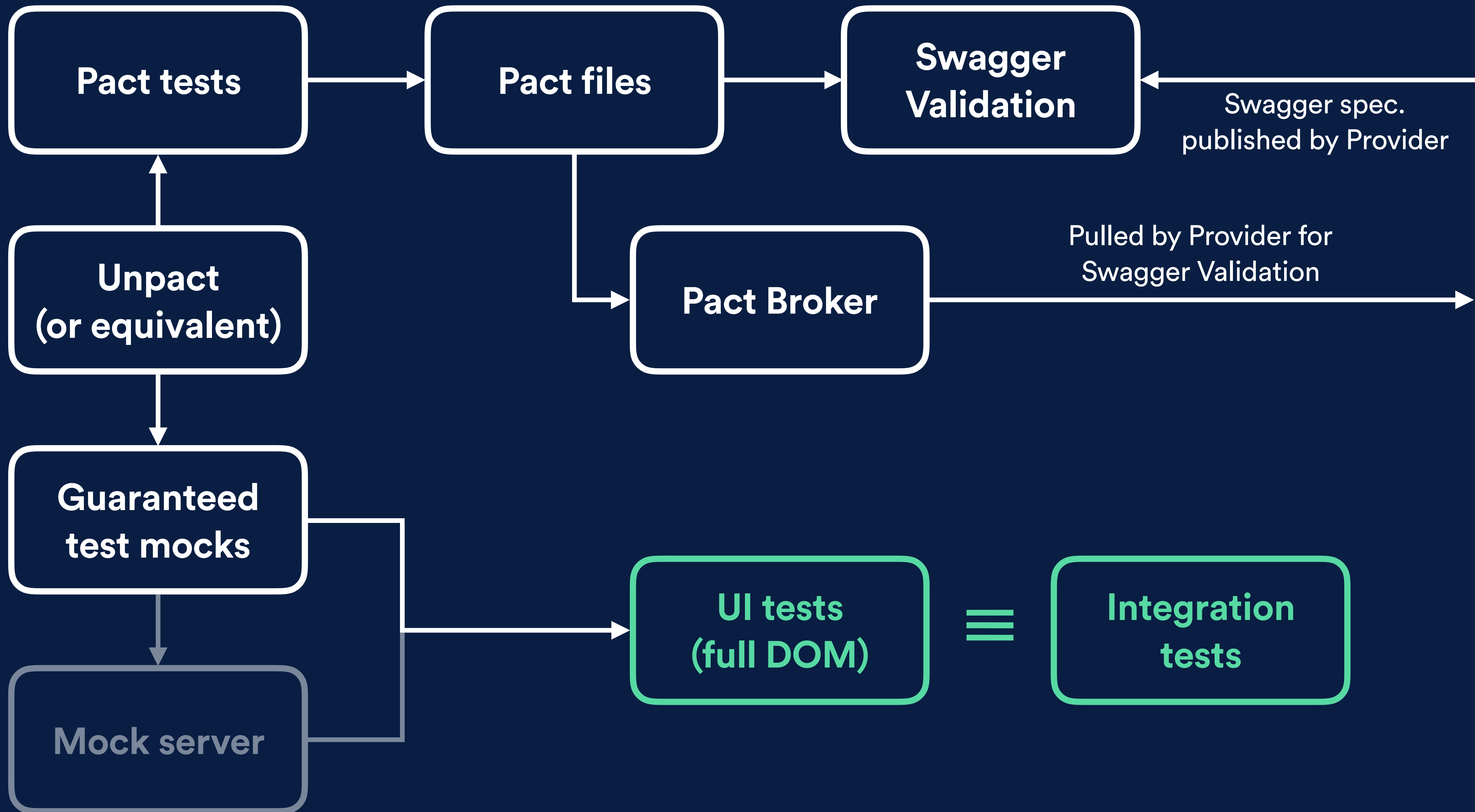




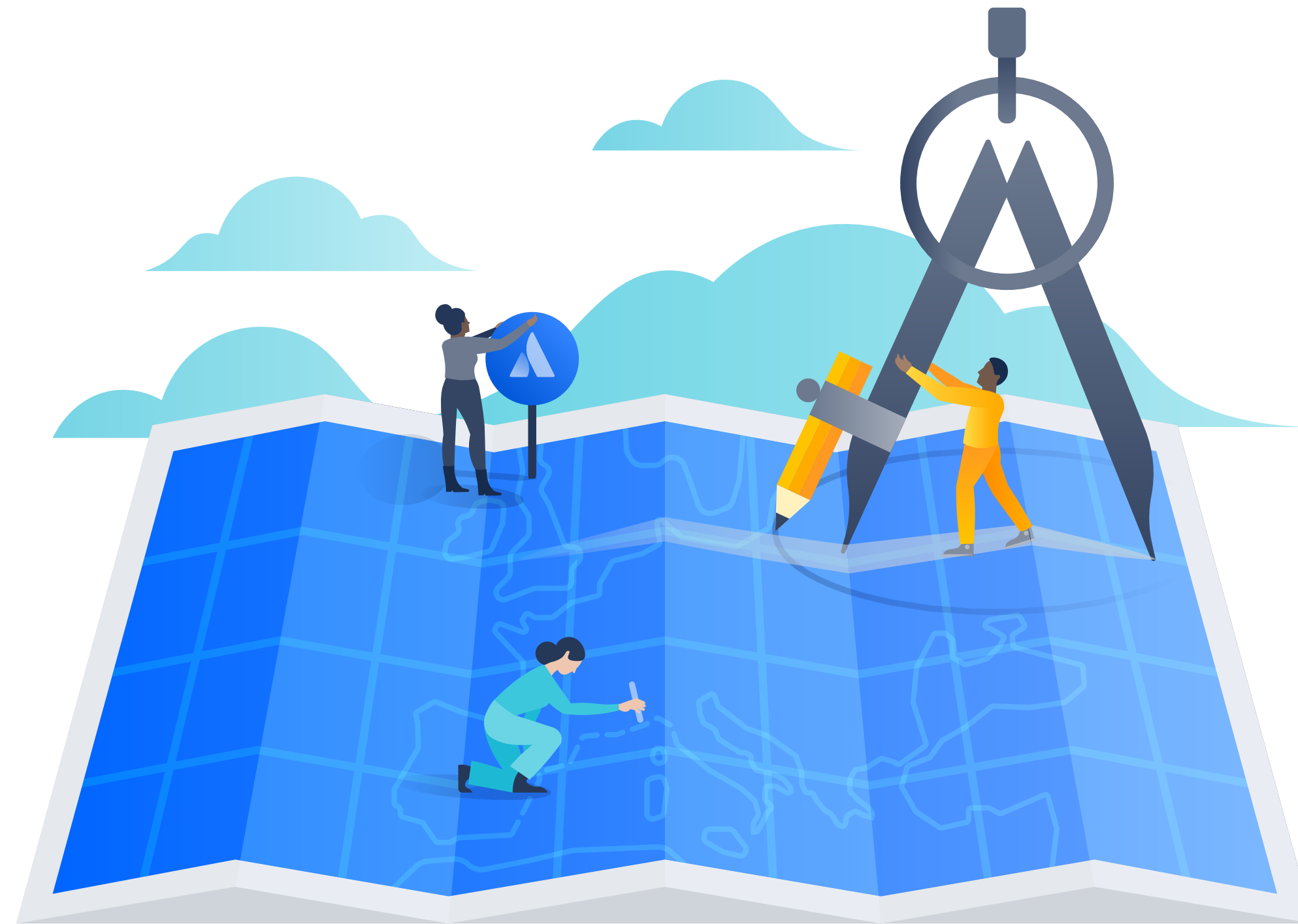








SUMMARY



FRONTEND INTEGRATION TESTING FRAMEWORK

- 1** **UI component tests**
- 2** **Pact (contract) tests**
- 3** **Swagger validation**
- 4** **Unpact**



Thank you!



DAVID PIPER | FRONTEND DEVELOPER | ATlassian