

Consumer-Driven Contract Tests using **PACT**

Alternative integration testing for microservice environments

"Dieses Foto" von Unbekannter Autor ist lizenziert gemäß CC BY-ND



About me & MAXIMAGO

Maik Schöneich



- Born in 1983
- Finished education in 2006
- .NET Developer since then
- Web-Frontend developer since 2013

MAXIMAGO



- UI Special Unit
- Based in Lünen (near Dortmund)
- User centered software development
- Business Software as simple as Apps

Microservice Architecture

One unit does
exactly one
thing well!

Deployment
Units

Modularization

Fail Over /
Backup

Scaling

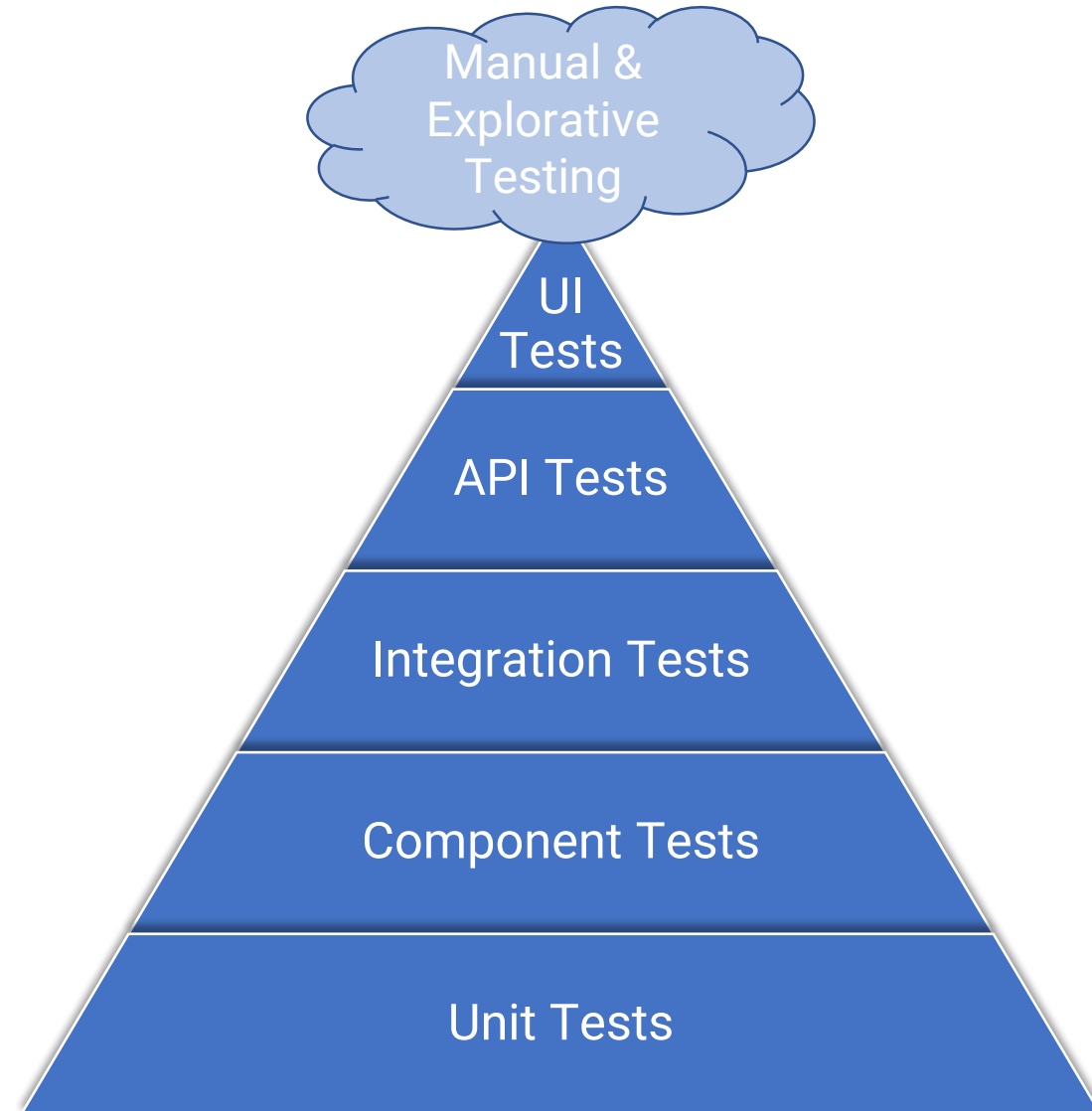
Divide &
Conquer

Decoupling

Testing

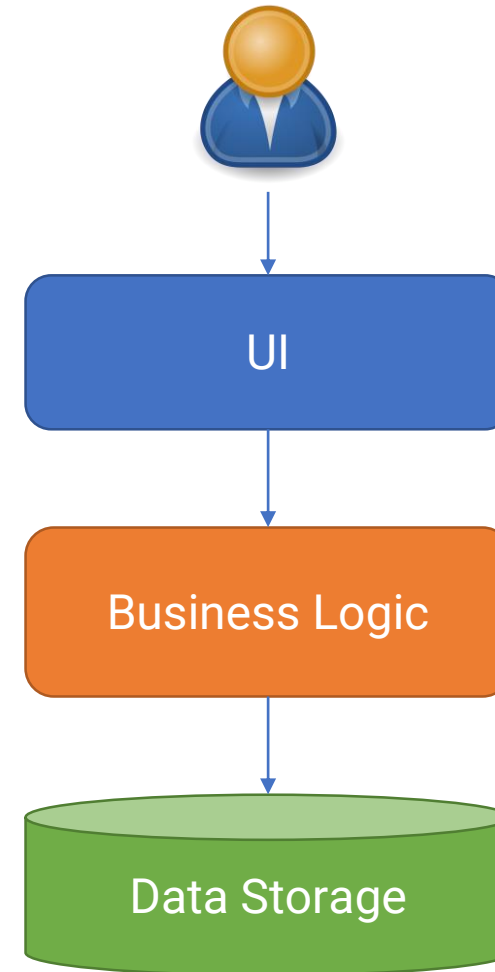
- Ideally test each unit
- No time
- No money

→ Focus on the important parts



End-to-End Tests...

- Testing the users' scenarios
- Focus on the „happy path“
- Involve all parts of the software
 - Frontend
 - Business logic
 - Data storage

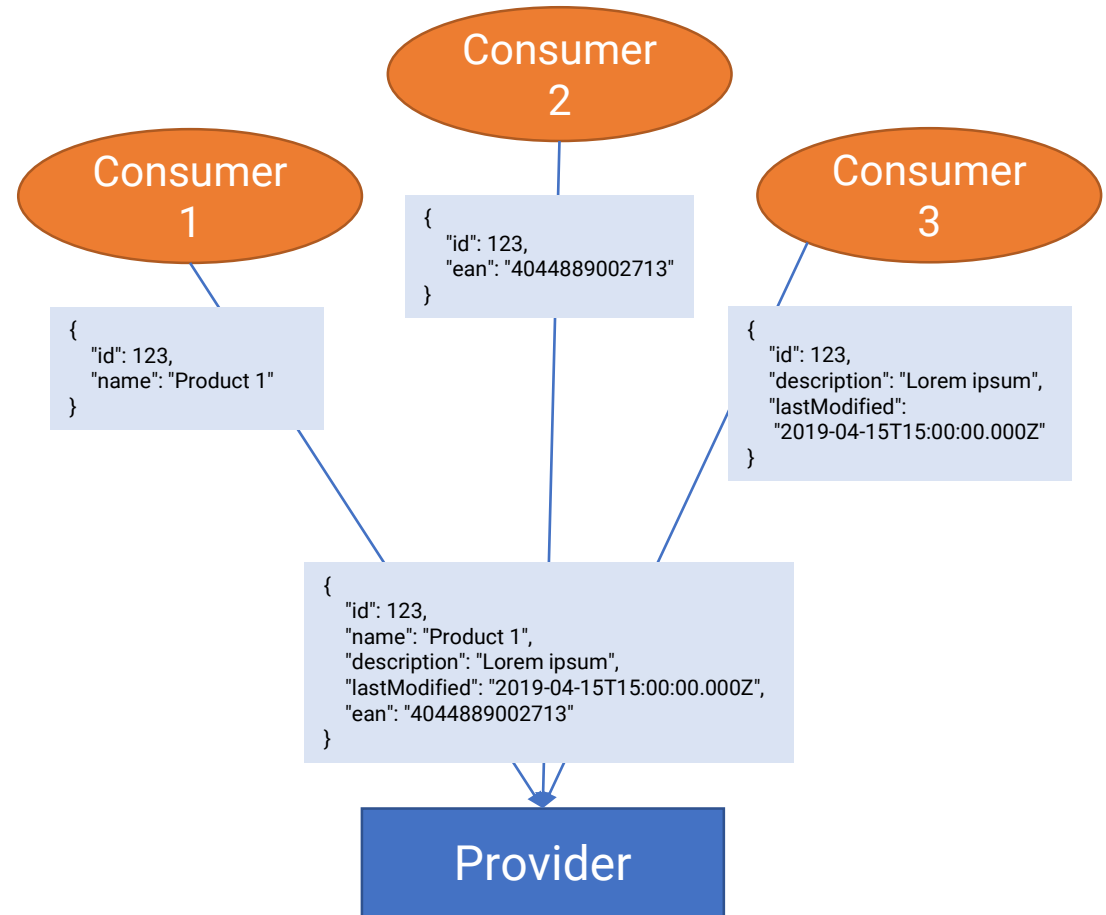


Interaction is missing

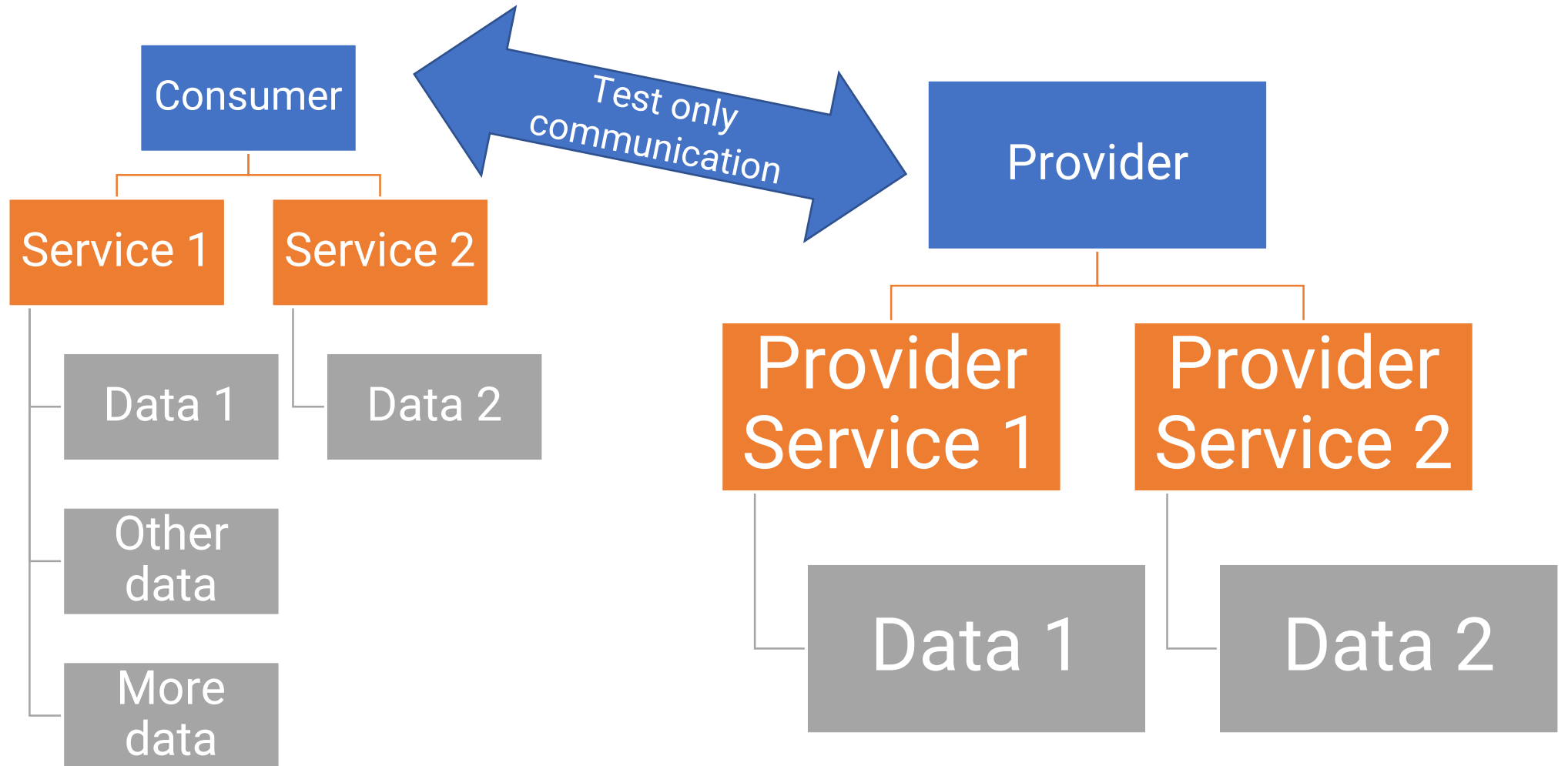
Can systems talk to each other?

Consumer Driven Contracts

- Consumer specifies its requirements
 - Provider implements the requirements
- All requirements of all known consumers build up the provider's API



Consumer Driven Contract Testing



Consumer Driven Contract Testing

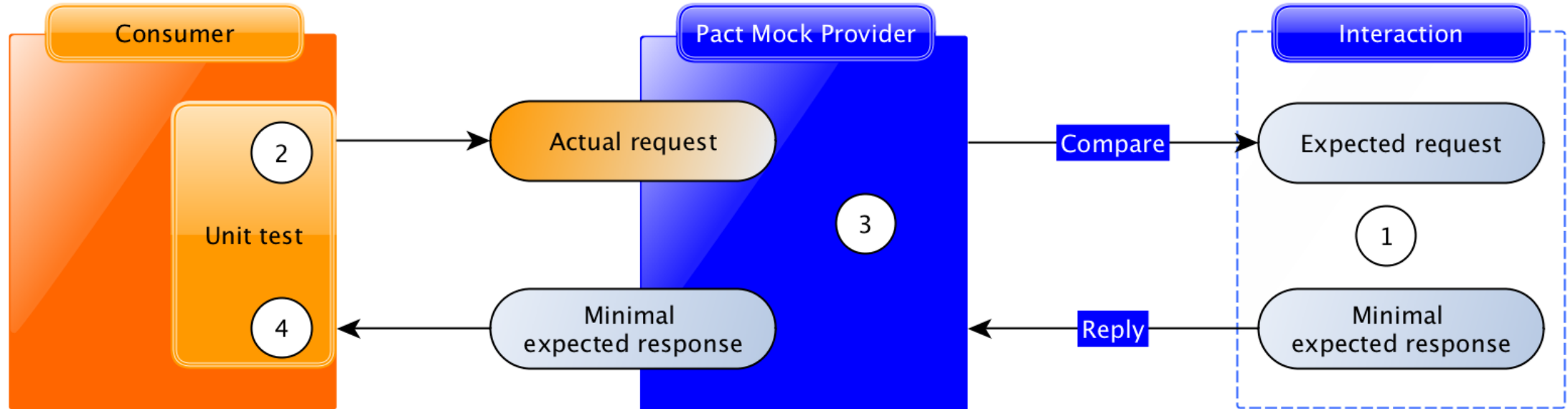
PACT

PACT Framework

<https://docs.pact.io>

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

Consumer Tests



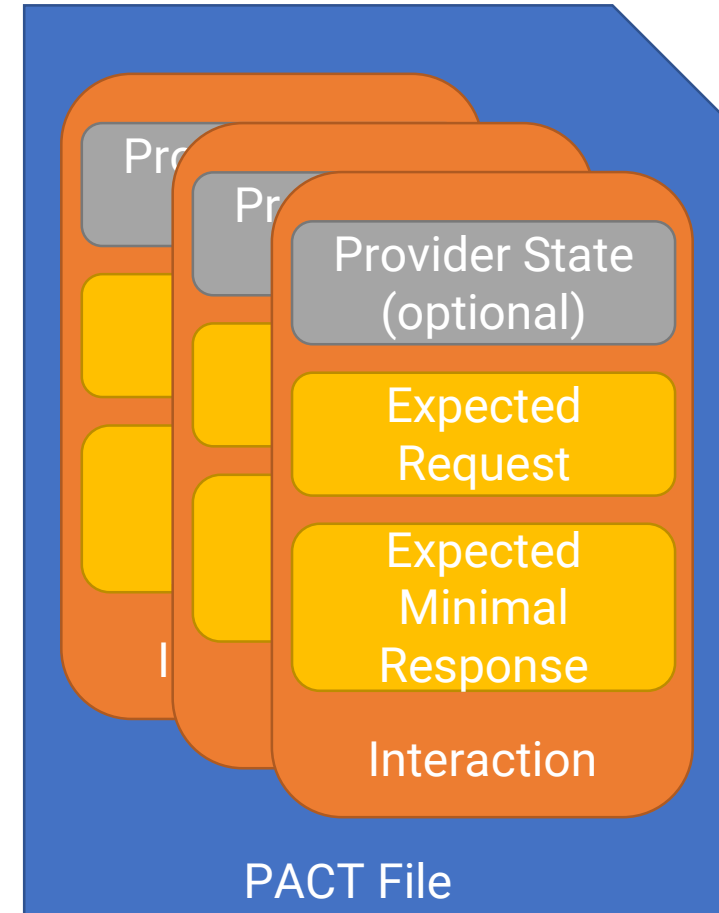
<https://docs.pact.io>

Demo

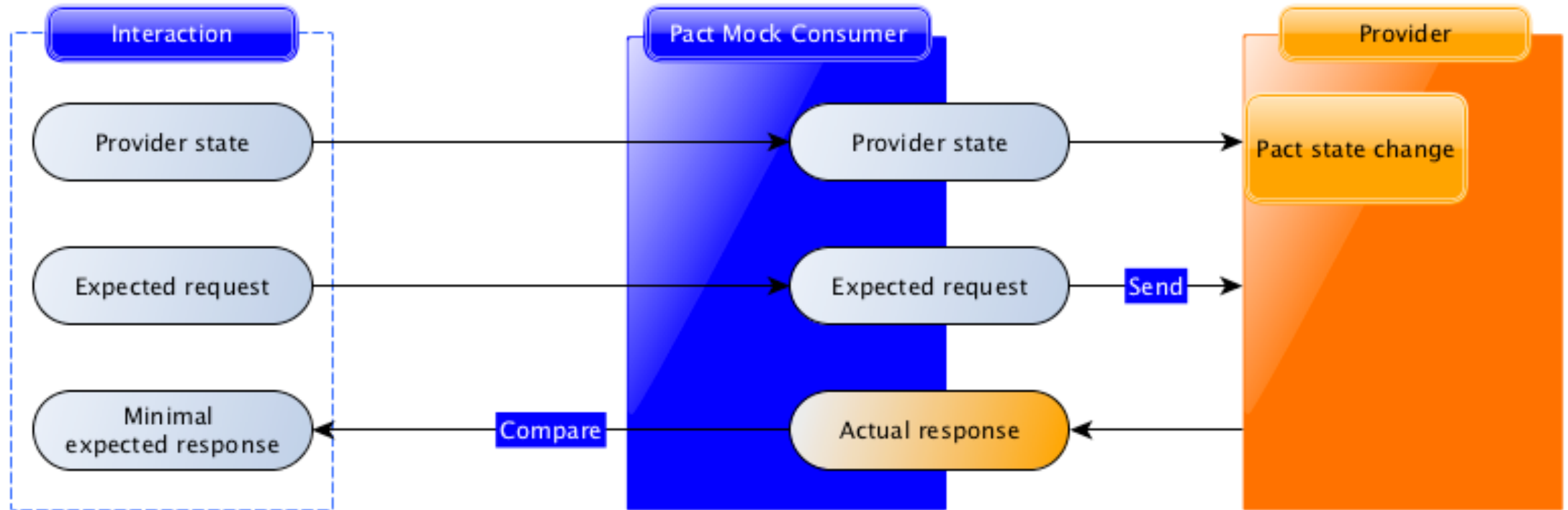
Creating Consumer Tests

What is a Pact?

- List of Interactions
- Interaction
 - Expected request
 - Expected minimal response
 - (Provider State)



Provider Tests



<https://docs.pact.io>

Demo

Creating Provider Tests using PACT file

Demo

Creating Consumer Tests for Web Client

PACT Broker

- Collects PACT files
- Easy access for interested parties
- Versioning for Consumers / Producers
- Compatibility matrix

Demo

Using PACT Broker for file distribution

Conclusion

Does it help to improve my quality?

!! YES !!

Well done

- Low learning curve
- Get started quickly
- Highly automated
- Support multiple programming languages
- Integration in existing test suites possible
- Support scenario based testing
 - Data dependant
 - Edge cases

Things to consider

- Provide more tools for writing provider tests
 - Steps need to be copied from project to project
 - Own library → Why not integrate it into the framework?

Recommendations



Thanks for your time!

<http://tiny.cc/dwx2019-pact>

