



Контрактное тестирование как альтернатива системному

Дмитрий Шидловский, Head Of QA

Дмитрий Шидловский

- **Head of QA Wallarm**
- QA Lead Soveren (Startup with biggest seed round)
- QA Lead Kaspersky (Cloud High Load Infrastructure)
- SDE Payler (Payment System)
- SDE Alba Plus (ERP System)

Summary 8 year of experience

Проблема системного тестирования

- Долгое и ресурсозатратно
- Неэффективно для микросервисной архитектуры
- Сложно покрывать тяжелые сценарии
- Замедление процесса разработки

Целевая аудитория

- Tech lead / Team lead
- Senior QA / Dev
- Middle Qa / Dev

Определения

- Functional testing - тестирование функциональные требований
 - System
 - Integration
 - Unit
- Consumer - сервис который потребляет api
- Provider - сервис который предоставляет api

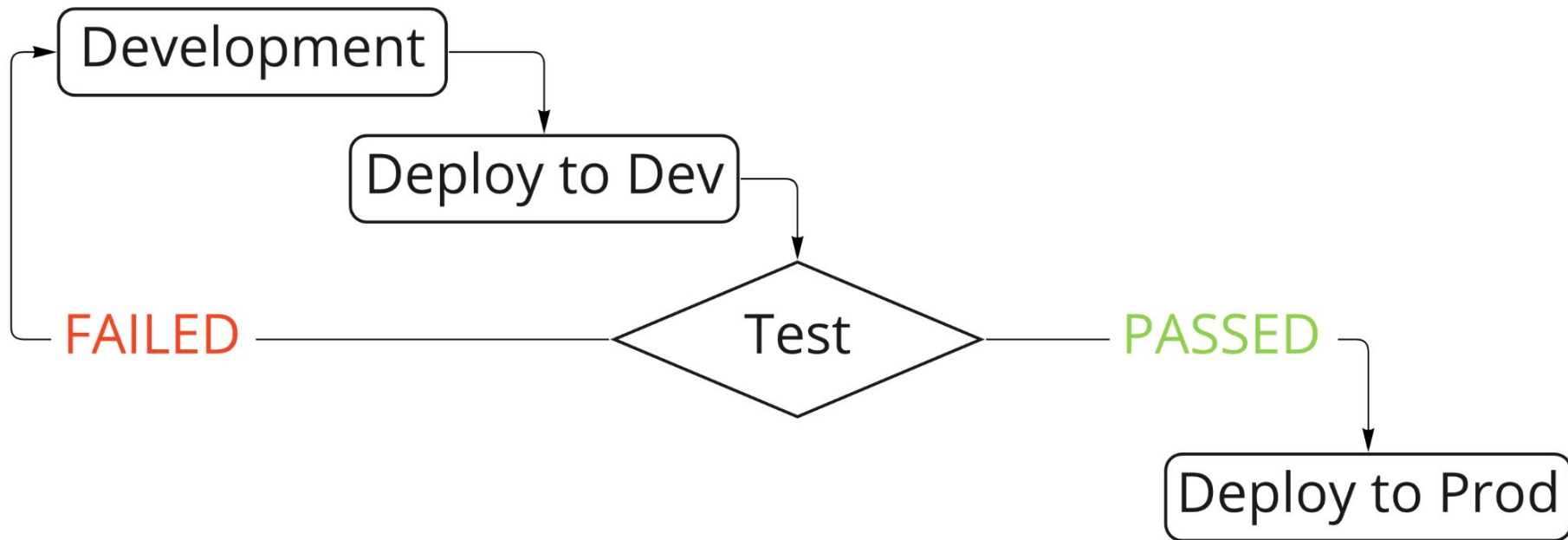
Микросервисная архитектура

- Параллельная разработка разных частей системы
- Параллельный деплой
- Простой процесс CI/CD

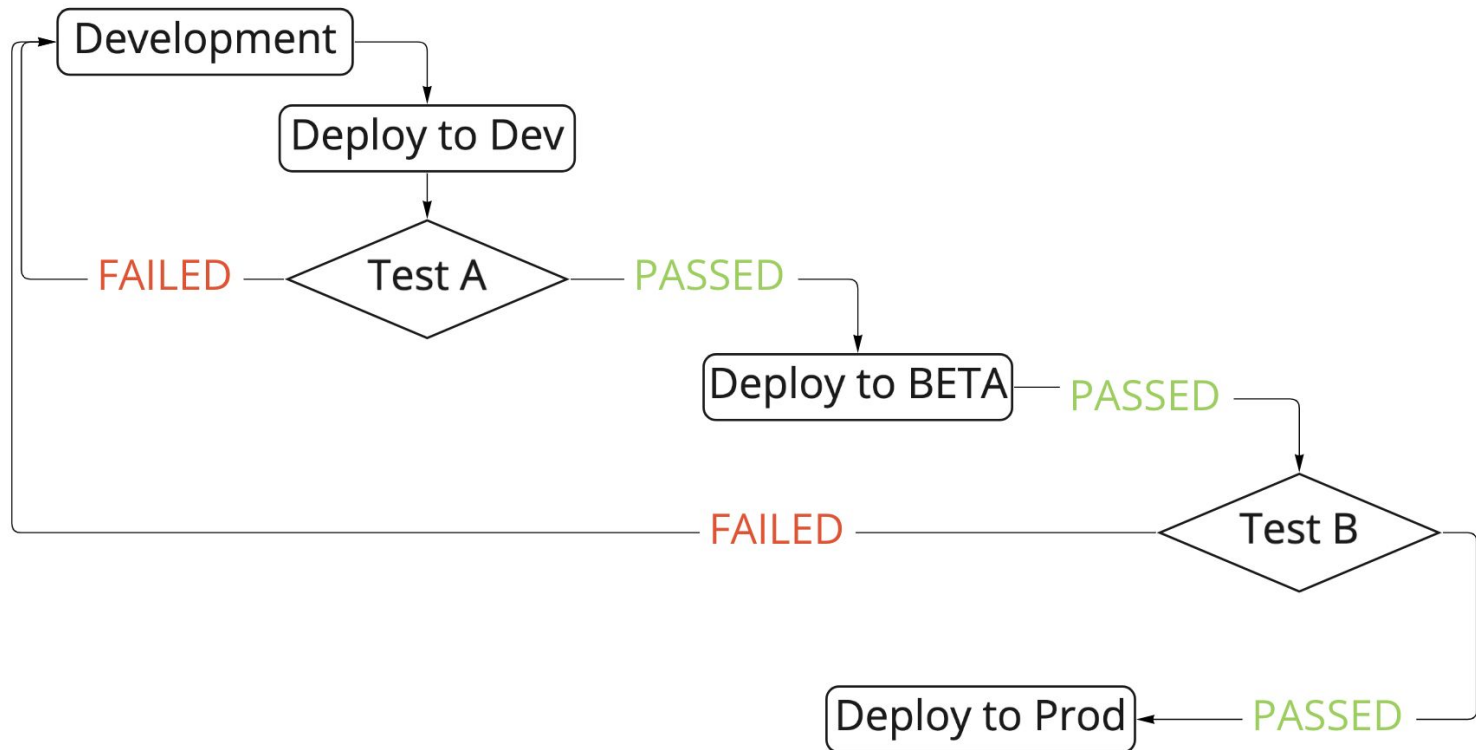
TTM - Time To Market

- Быстрый процесс разработки
- Выявление ошибок на ранних этапах
- CI/CD с минимизацией этапов тестирования

CI/CD при системном тестировании



CI/CD при системном тестировании



TTM при системном тестировании

- Тяжелый процесс CI/CD
- Неудобно выкатывать мелкие фичи
- Сложно разрабатывать параллельно

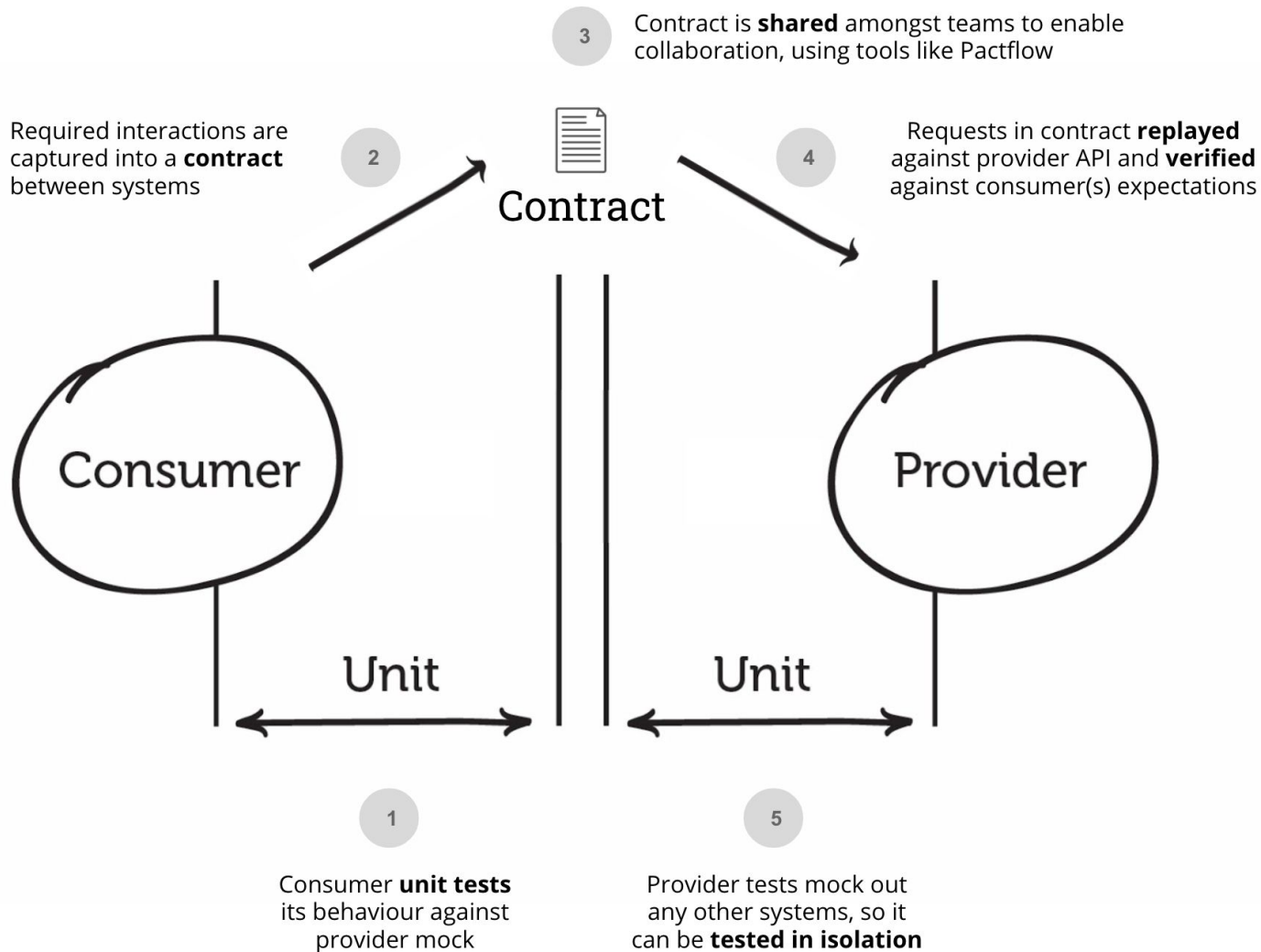
- Проверяем систему как монолит
- Деплой всей системы при изменении любого сервиса

Примеры багов на системном уровне

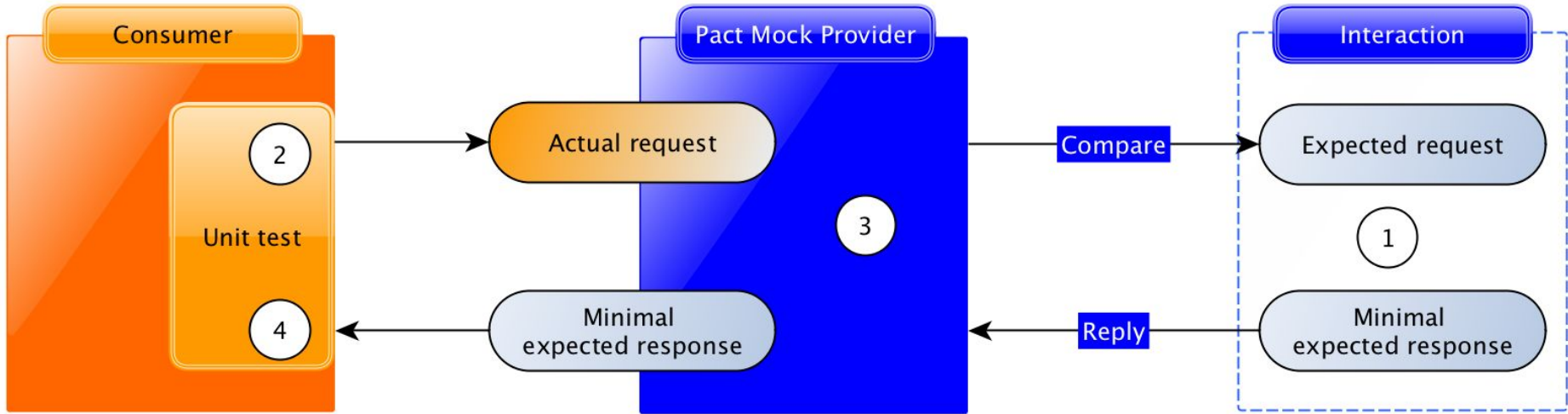
- *Consumer* задеплоился раньше чем **Provider**
- **Provider** задеплоился с новым функционалом на старые *Consumer*
- Баги бизнес логики взаимодействия двух сервисов
- Забыли сделать доработку в каком-то сервисе

Как контрактные тесты решают проблему

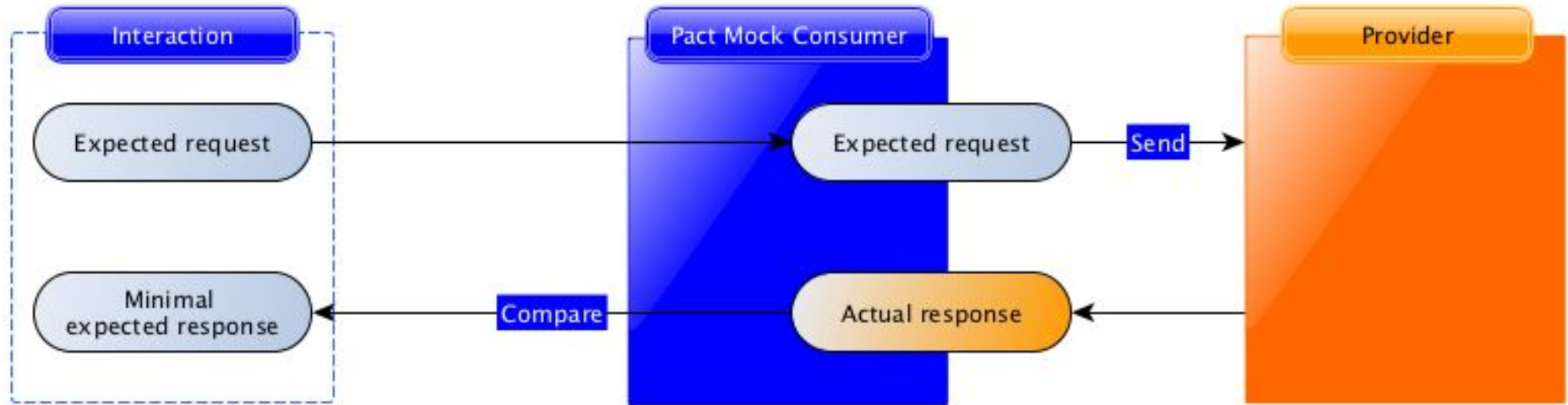
- Закрывают вопрос системных багов
- Быстрый запуск тестов в изоляции для сервиса
- Автоматизировано проверяют возможность деплоя





Consumer test




Provider test





Can I Deploy

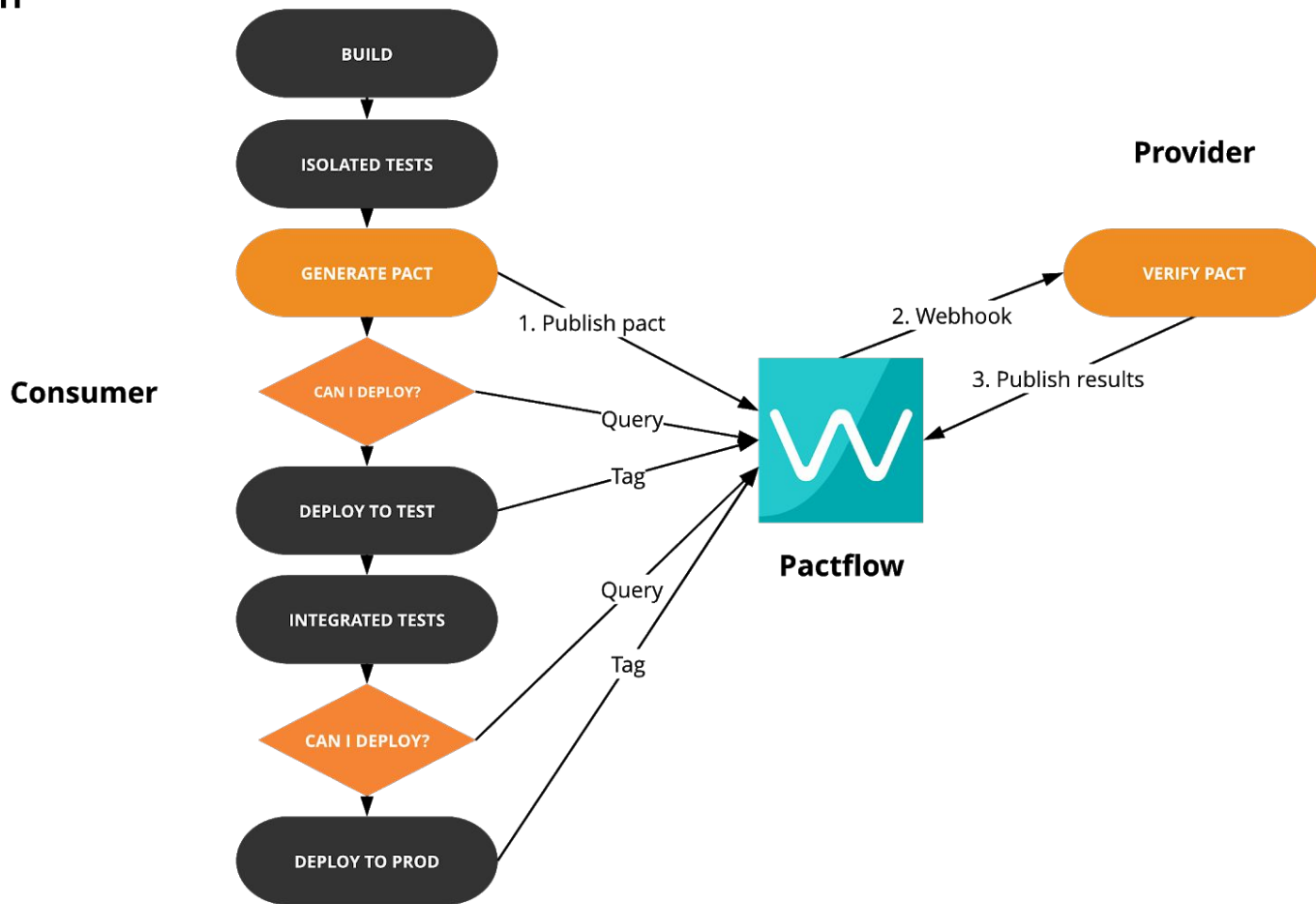
TestConsumer  **TestProvider** ⋮ 

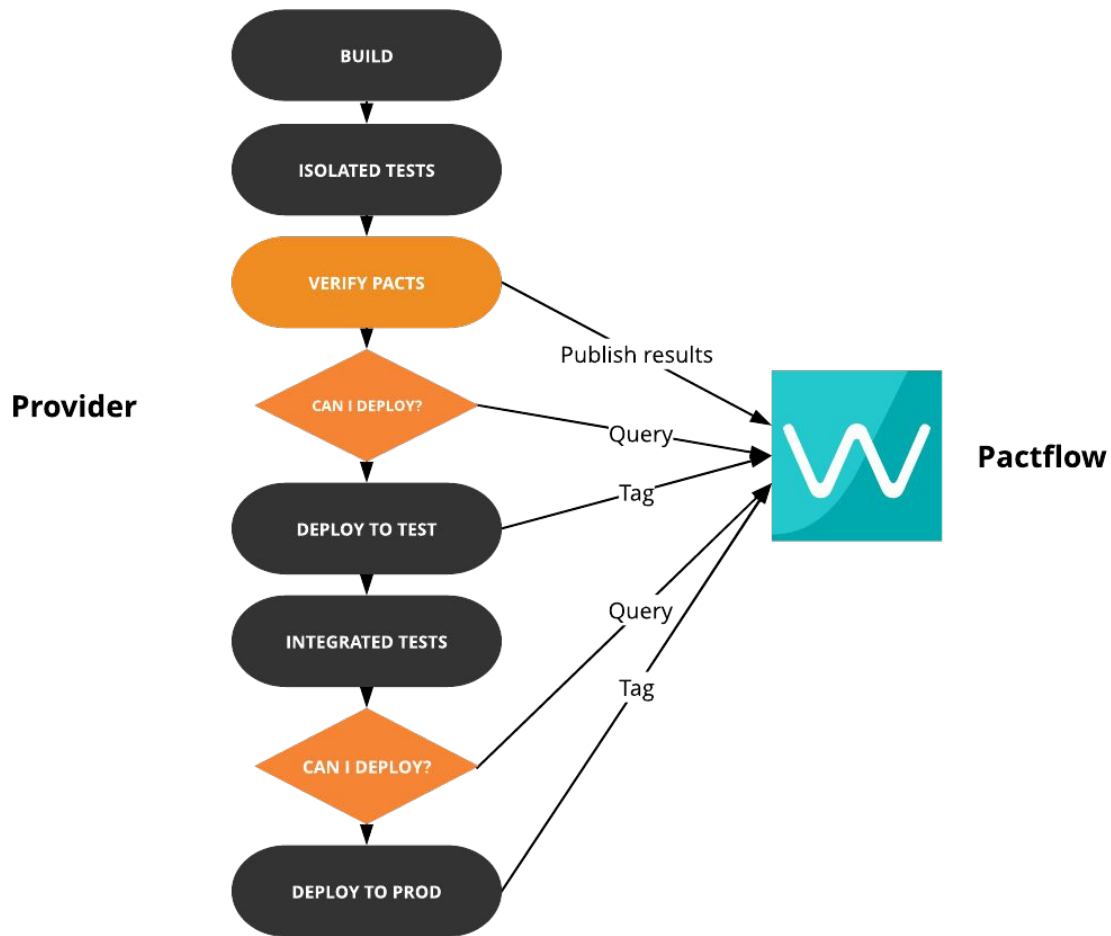
Success



CONSUMER VERSION	1.0.0	PROVIDER VERSION	1.0.0
BRANCH	 master	BRANCH	N/A
ENVIRONMENTS	DEV	ENVIRONMENTS	DEV
TAGS	 master	TAGS	N/A
PUBLISHED AT	an hour ago	VERIFIED AT	an hour ago

[VIEW PACT](#)





Минусы контрактного тестирования

- Эффективен в хорошо спроектированных системах
- Требуется изменять процесс разработки
- Ограниченная область применения
- Тяжело для понимания

Другие альтернативы

- Канареечное тестирование
- Смук тестирование через toggle
- Контрактное тестирование в моно репозитории

Итоги

- Архитектура сервисов не так важна, процесс разработки и тестирования
- Ошибки которые раньше находили только на системном уровне, можно находить на уровне контрактных тестов
- Стремление к быстрому ТТМ ведет к внедрению более эффективных подходов.

Итоги

- Архитектура сервисов не так важна, процесс разработки и тестирования
- Ошибки которые раньше находили только на системном уровне, можно находить на уровне контрактных тестов
- Стремление к быстрому ТТМ ведет к внедрению более эффективных подходов.

Итоги

- Архитектура сервисов не так важна, процесс разработки и тестирования
- Ошибки которые раньше находили только на системном уровне, можно находить на уровне контрактных тестов
- Стремление к быстрому ТТМ ведет к внедрению более эффективных подходов.

Tools

- Pact.io
- Pactflow.io
- Github Actions



This is the end, you know...