

A person wearing a grey suit and a dark tie is holding a large white document titled "CONTRACT" in bold black letters. The person's hands are visible, holding the document and a black pen. The document has several paragraphs of text, but they are mostly illegible due to the angle and focus. The background is a plain white surface.

# CONTRACT

## Contract testing

-

Verification across service  
boundaries



# Who am I?



Peter Czibik  
@peteyycz

[github.com/peteyycz](https://github.com/peteyycz)



kubernetes

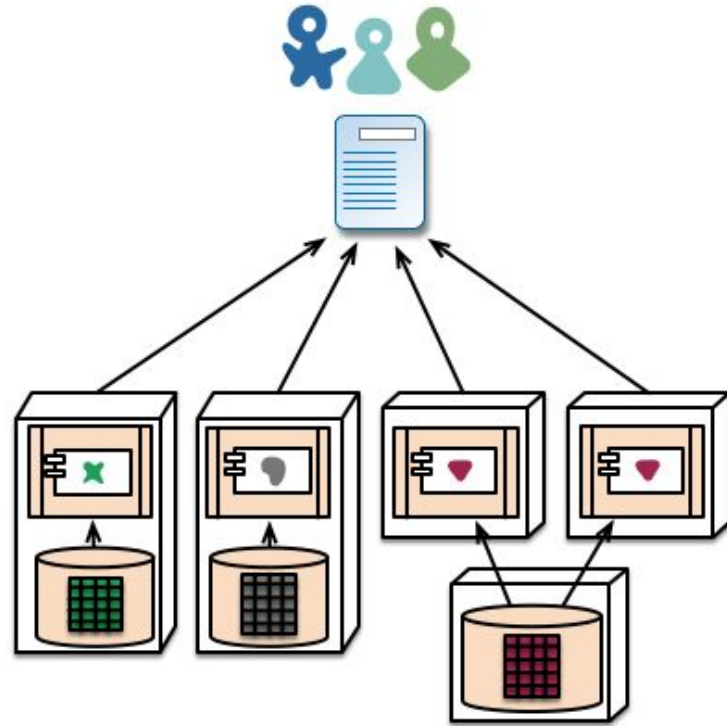
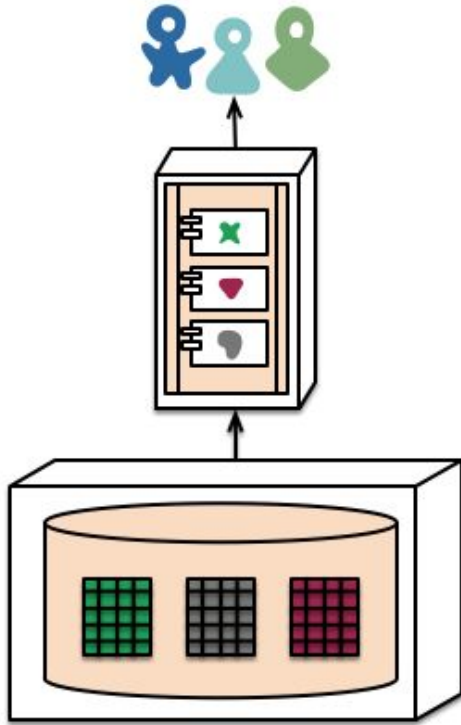


- Nerd
- Senior Node JS developer
- Linux enthusiast
- Lover of programming languages
- Generally a nice guy

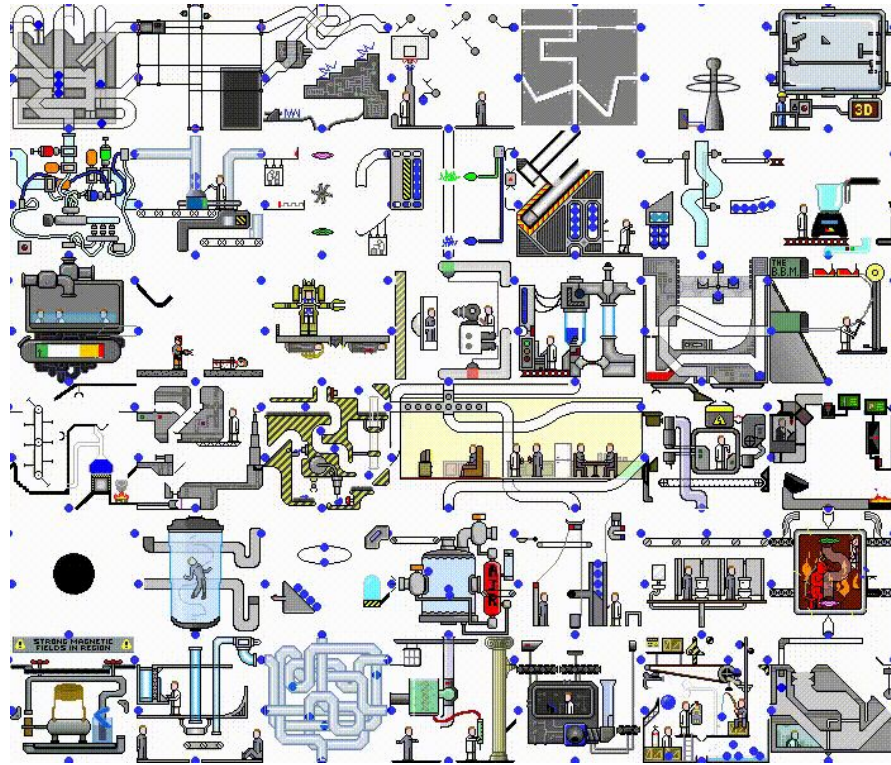


# RisingStack

We help companies succeed with microservices



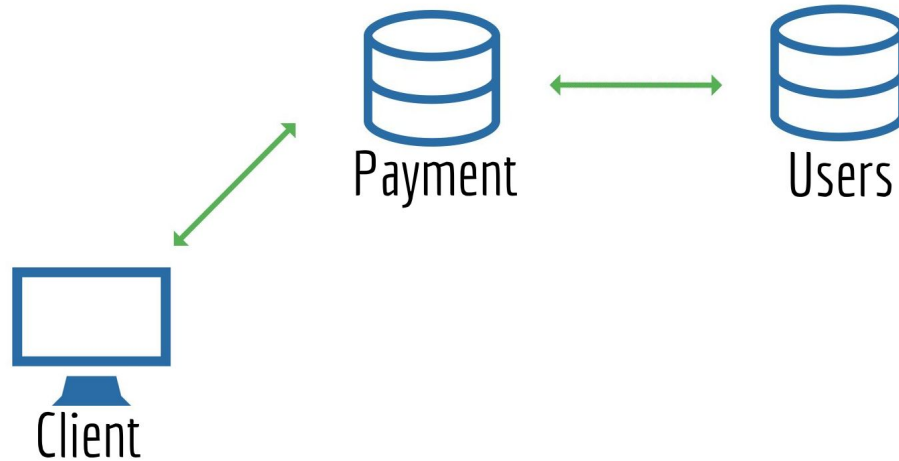
Who here works with microservices?



Everyone loves microservices

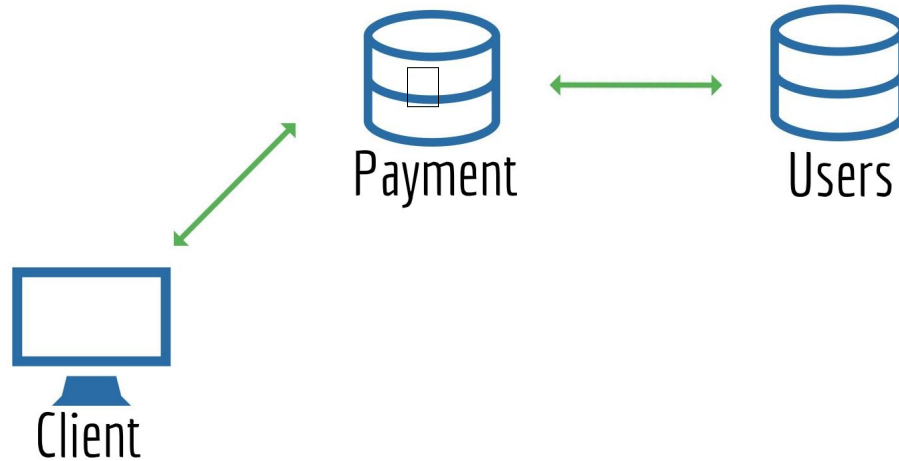
# A sad story of Atlassian

- Two independent services: Payment and Users



# A sad story of Atlassian

- Two independent services: Payment and Users



# A sad story of Atlassian

- Changing a single letter in a json response

```
{  
  "user": [ ... ]  
}  
  
{  
  "users": [ ... ]  
}
```



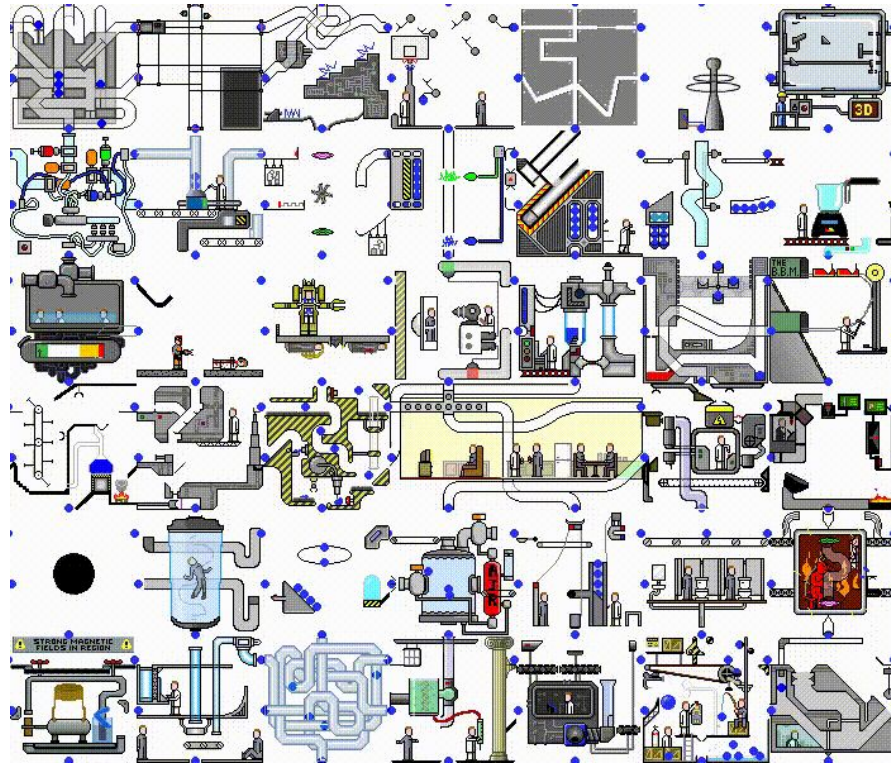


There's no worse thing than not letting users pay.



How can we avoid problems  
like that?



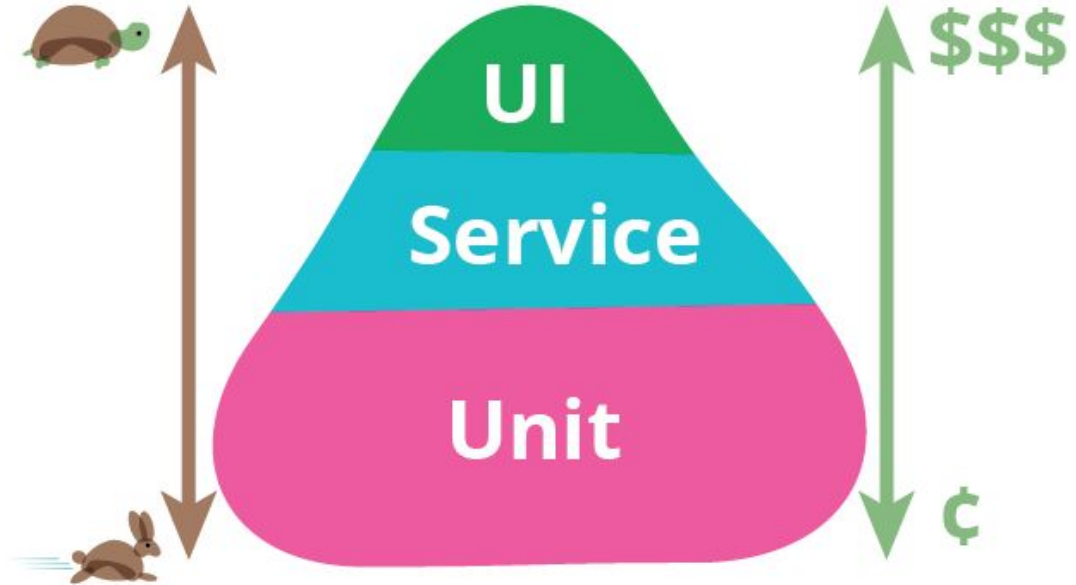


Everyone loves microservices

What kind of tests?

# Testing

- Unit tests
- Integration tests
- End-to-end tests



Those are not enough.

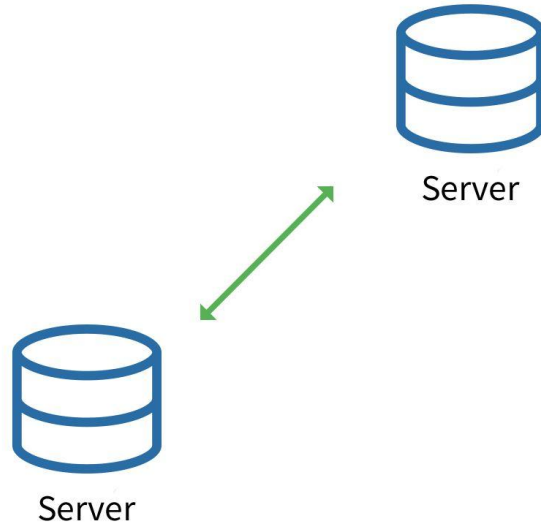




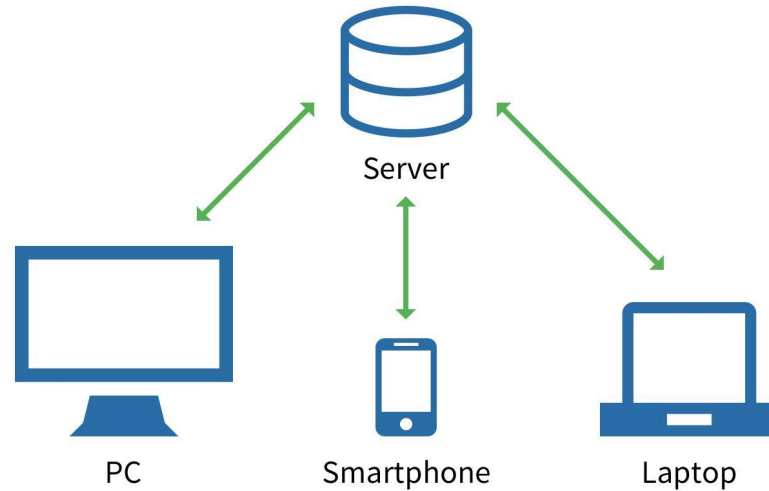
# Testing beyond borders

- Separate codebases
- Individual deployments
- Multiple programming languages

# Testing beyond borders



# Testing beyond borders



These concepts apply to both worlds.

# Solution A

# Mocking

- Light on resources
- No data pollution
- Spares maintenance cost
- Points out issues quickly
- Idempotent
- Not trustworthy



# Solution B

# Using an instance of the upstream service

- Hard to use other team's code
- Trustworthy
- Expensive
- Non-Idempotent



# Wrap up

## Mocking

- Idempotent
- Light on resources
- Cheap (\$\$\$ and code)
- CI compatible
- Not trustworthy

vs.

## Extra instance

- Non-Idempotent
- Resource heavy
- Expensive (\$\$\$ and code)
- Not CI compatible
- Trustworthy

# Solution C

C for Contract

# Contract (📄) Testing

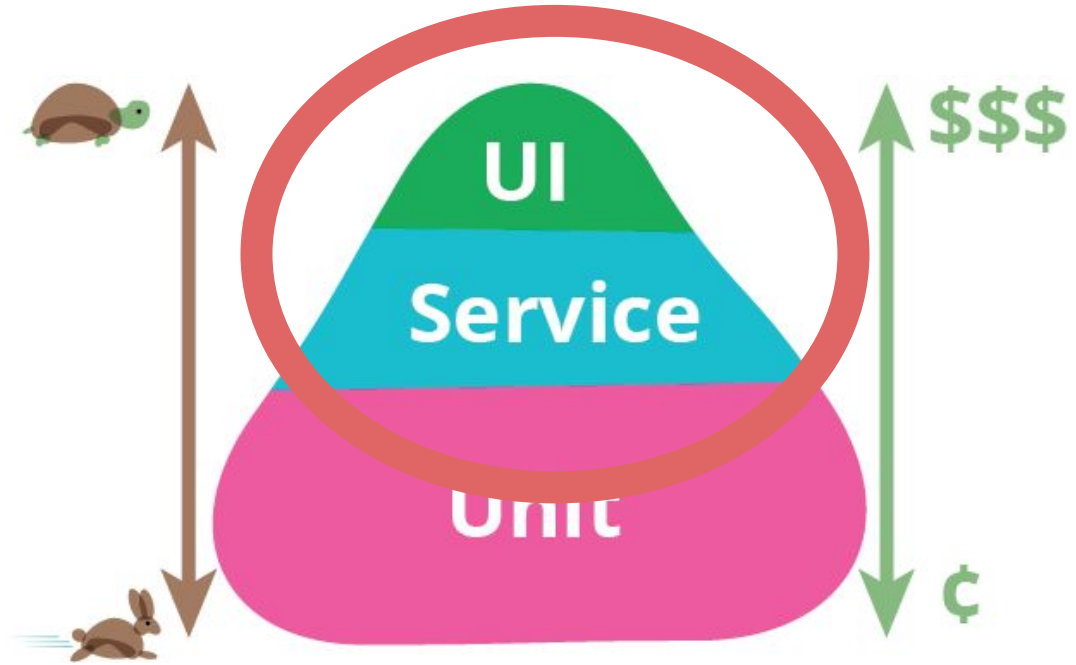
- A contract defines a fix interface (🔌) between two parties
- Both parties use this 🔌 definition to test

# Testing

- Uses **real data**
- No extra infrastructure cost
- **Fast**
- **Easy** to scale
- **Quick** setup
- Stable
- **Reliable**

# Testing

- Unit tests
- Integration tests
- End-to-end tests



# Consumer Driven Testing


Implementation steps:

1. Client makes expectations
2. Client uses these expectations to test
3. Serialize expectations to language independent format
4. Transfer expectations to server
5. Server uses those expectations to test

# PACTS

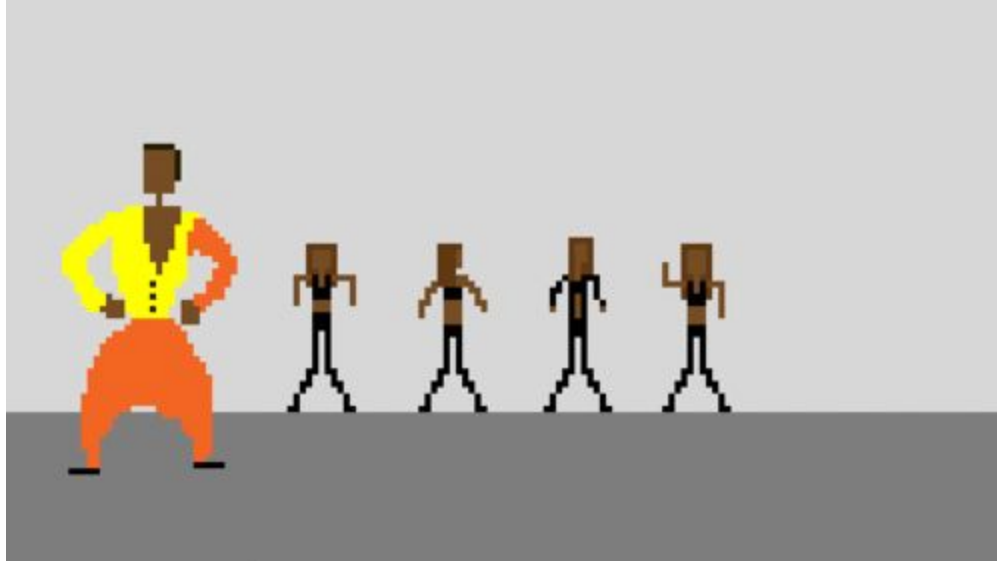
Meet Pact

# Pact

- Implementation of a Consumer Driven  Test Suite
- JSON (language independent)
- Libraries already exist in multiple languages
  - .NET
  - JVM (Java & Scala)
  - Javascript



Everyone stop.



It's demo time!





Thank you for your attention



¿Questions?