



End-to-End test environments

A dead End road

Roy Braam

 @rbraam



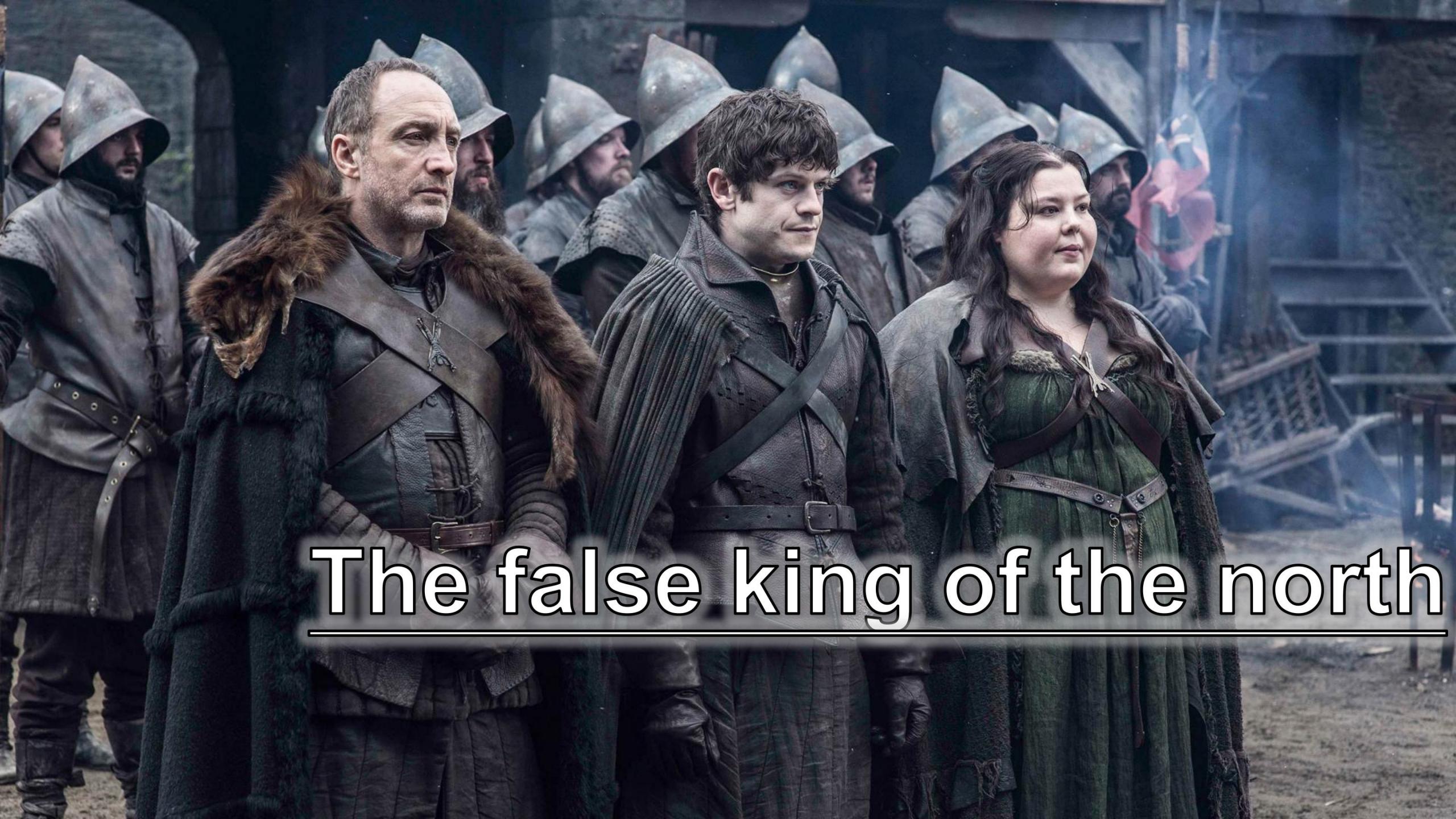
The slayer of automation



The Duke of false positives



The Prince of time consumption



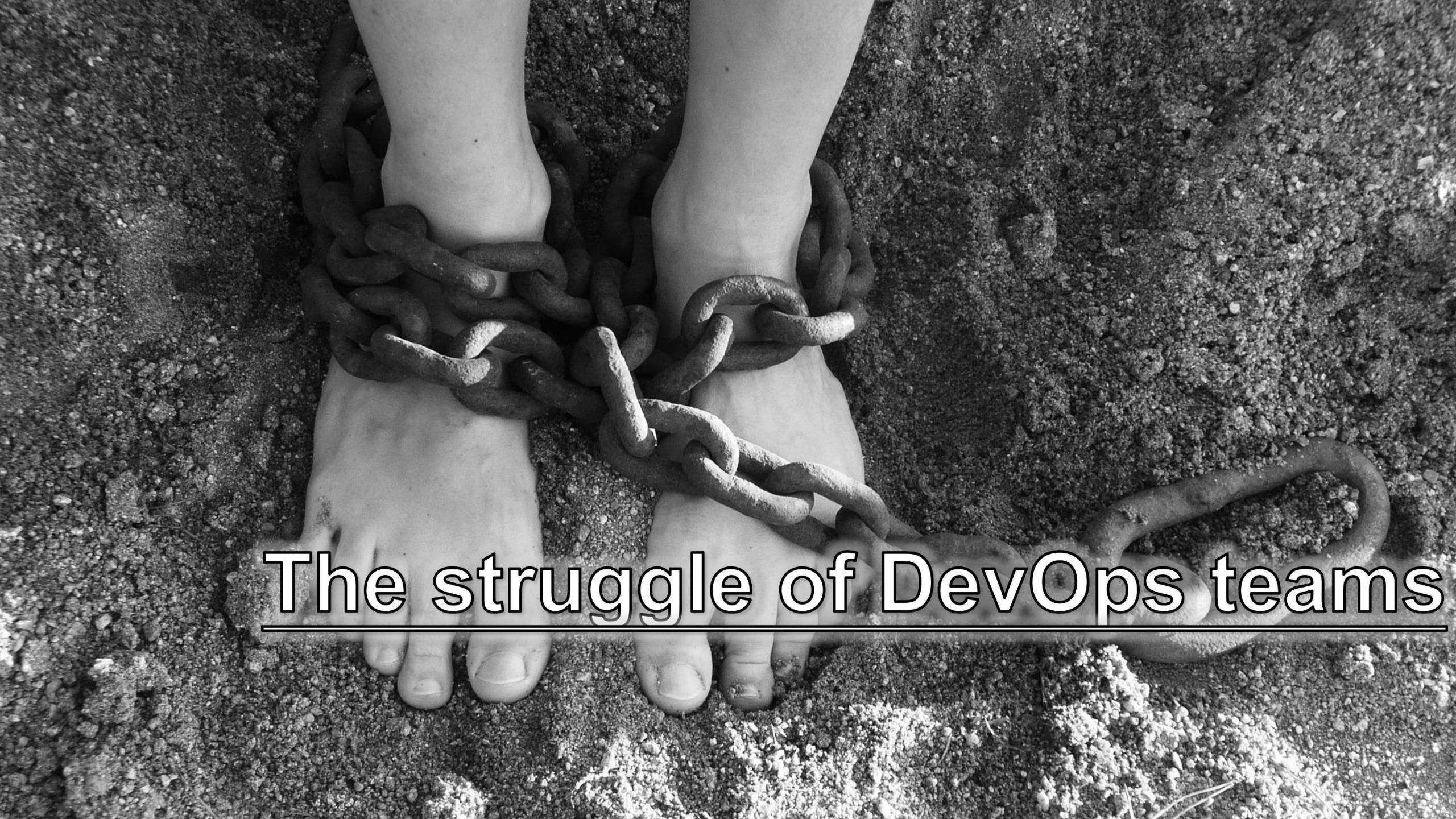
The false king of the north



The feared opponent of CI/CD

A large pile of gold bars of various sizes, some with red markings, stacked haphazardly against a light background.

The gold digger of IT



The struggle of DevOps teams



The challenger of patience

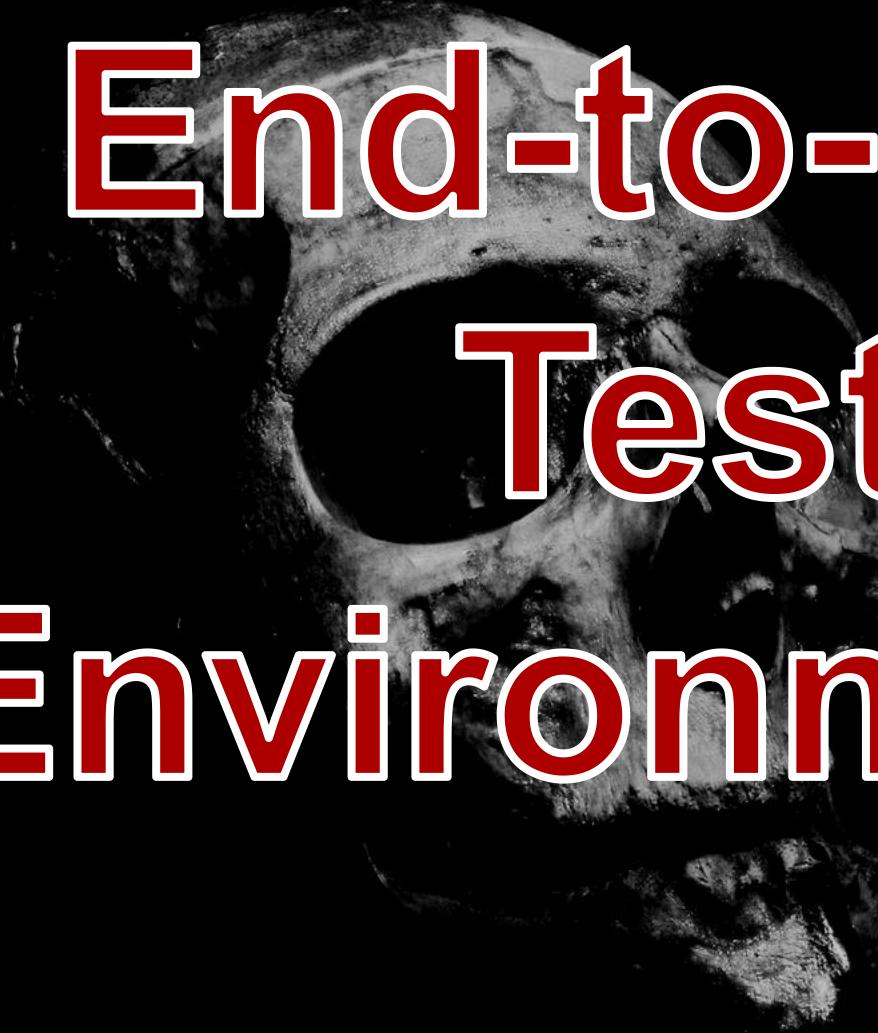


The master of frustration

A photograph of a weathered wooden utility pole against a clear blue sky. Numerous dark, twisted wires are attached to the pole, many of which are crisscrossed and hang loosely, creating a complex web-like pattern.

The dark side of Architecture

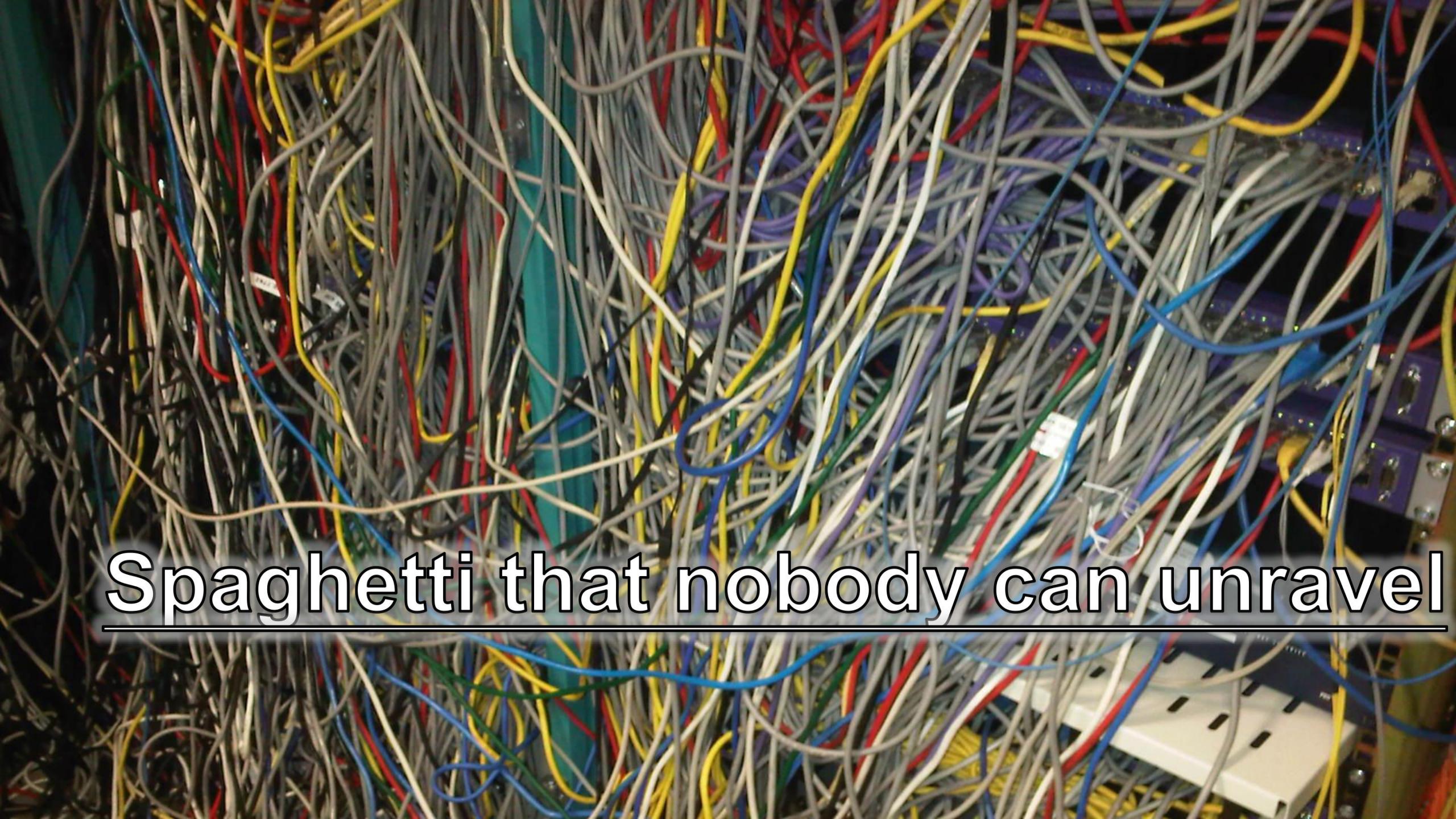
**End-to-End
Test
Environments**



**End-to-End
Test
Environments**

A wide-angle photograph of a large, abandoned industrial building. The interior is filled with debris, including twisted metal, wood, and plastic. The floor is covered in a thick layer of dust and trash. Large windows on the right side are shattered, looking out onto a bright exterior where other buildings are visible. The ceiling is made of steel beams and corrugated metal, with some panels missing. The overall atmosphere is one of decay and neglect.

The place nobody wants to be



Spaghetti that nobody can unravel

A photograph of a person standing on a mountain peak, silhouetted against a bright, hazy sky. The person has their arms outstretched to the sides. In the background, there are more mountain peaks, some of which are partially obscured by mist or low-hanging clouds.

It currently looks like production....

....and then it isn't anymore.



End-to-End test environments



jPoint

2/25/2020

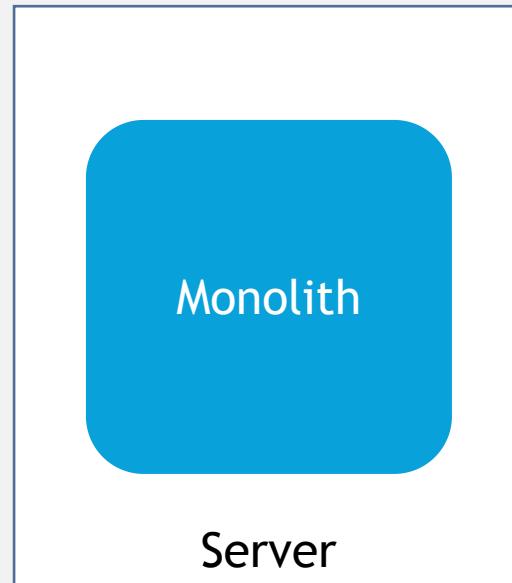
Roy Braam

 @rbraam

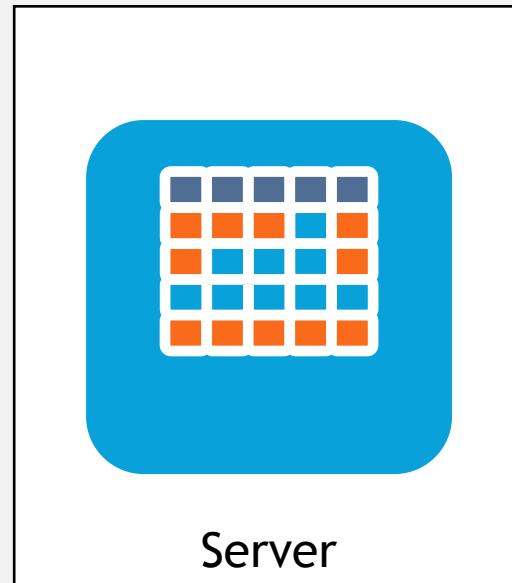
Architecture



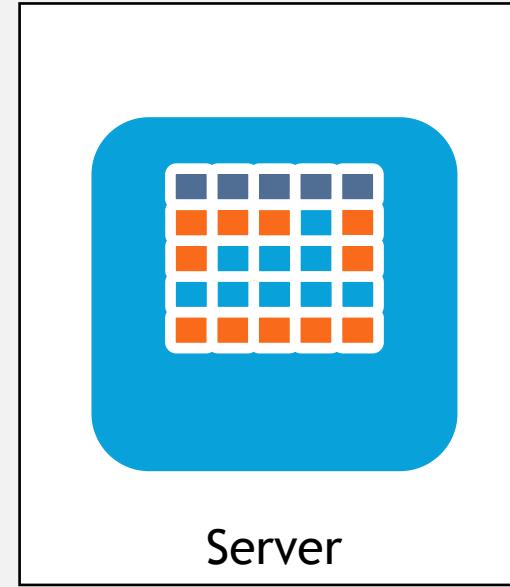
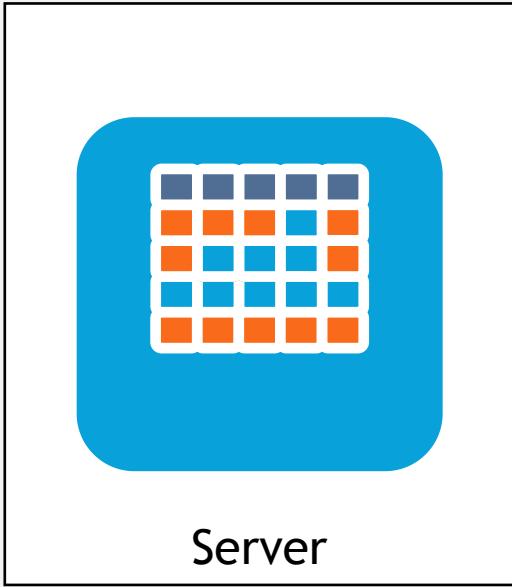
Monolith



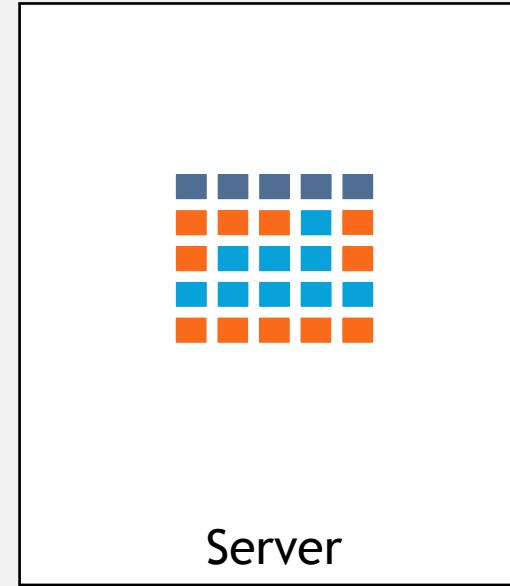
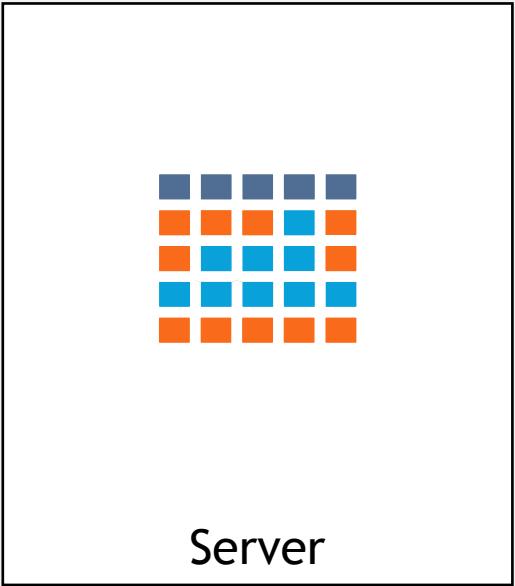
Modular Monolith



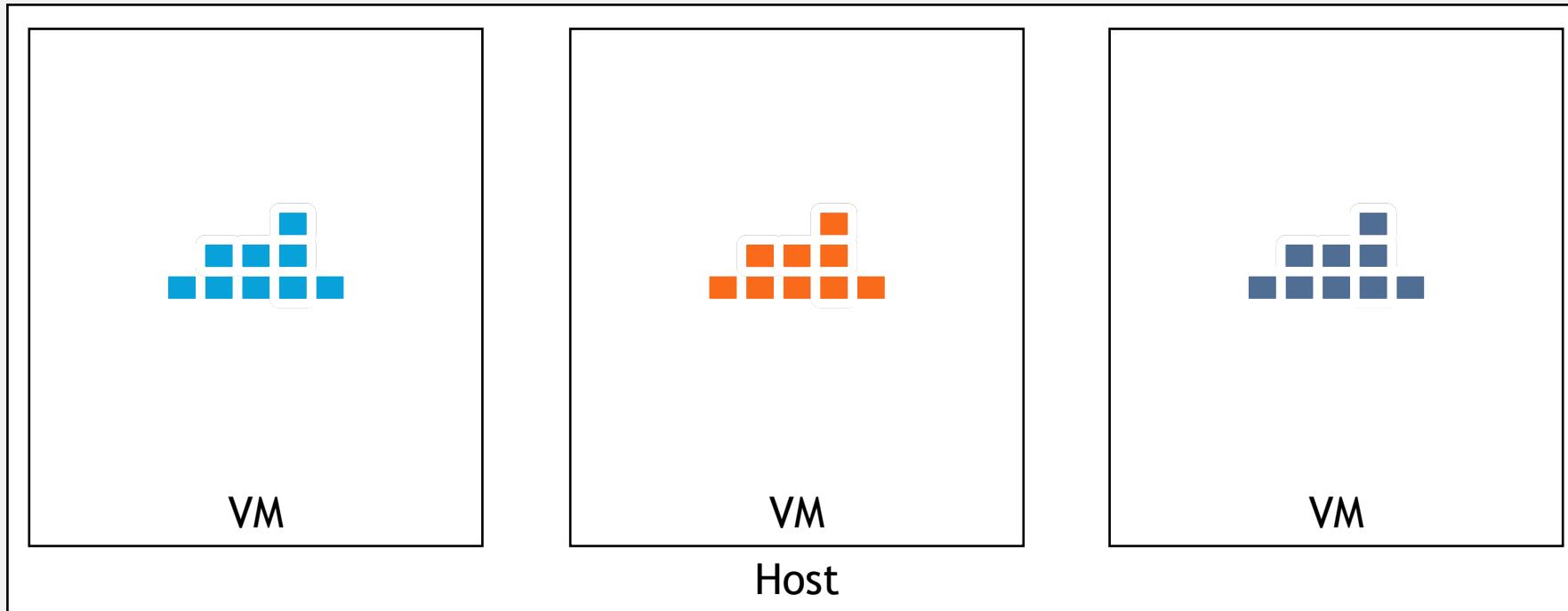
Multiple (Modular) Monoliths



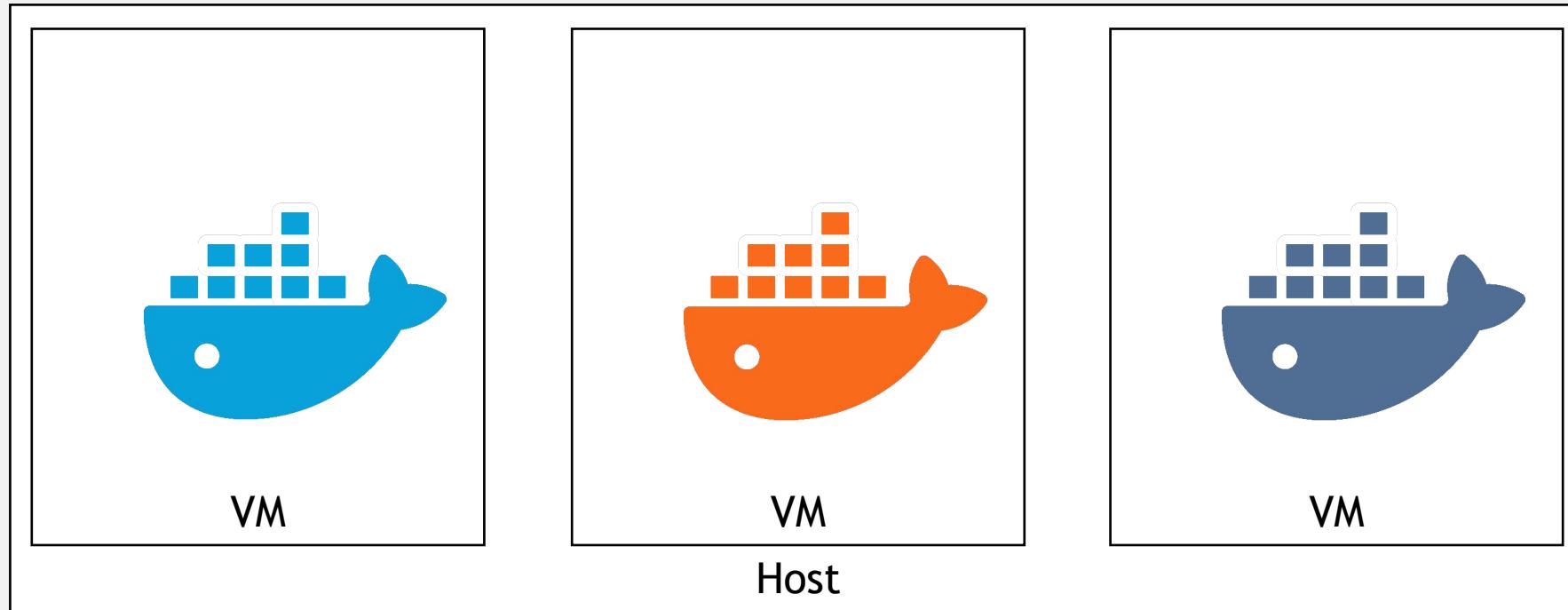
Microservices



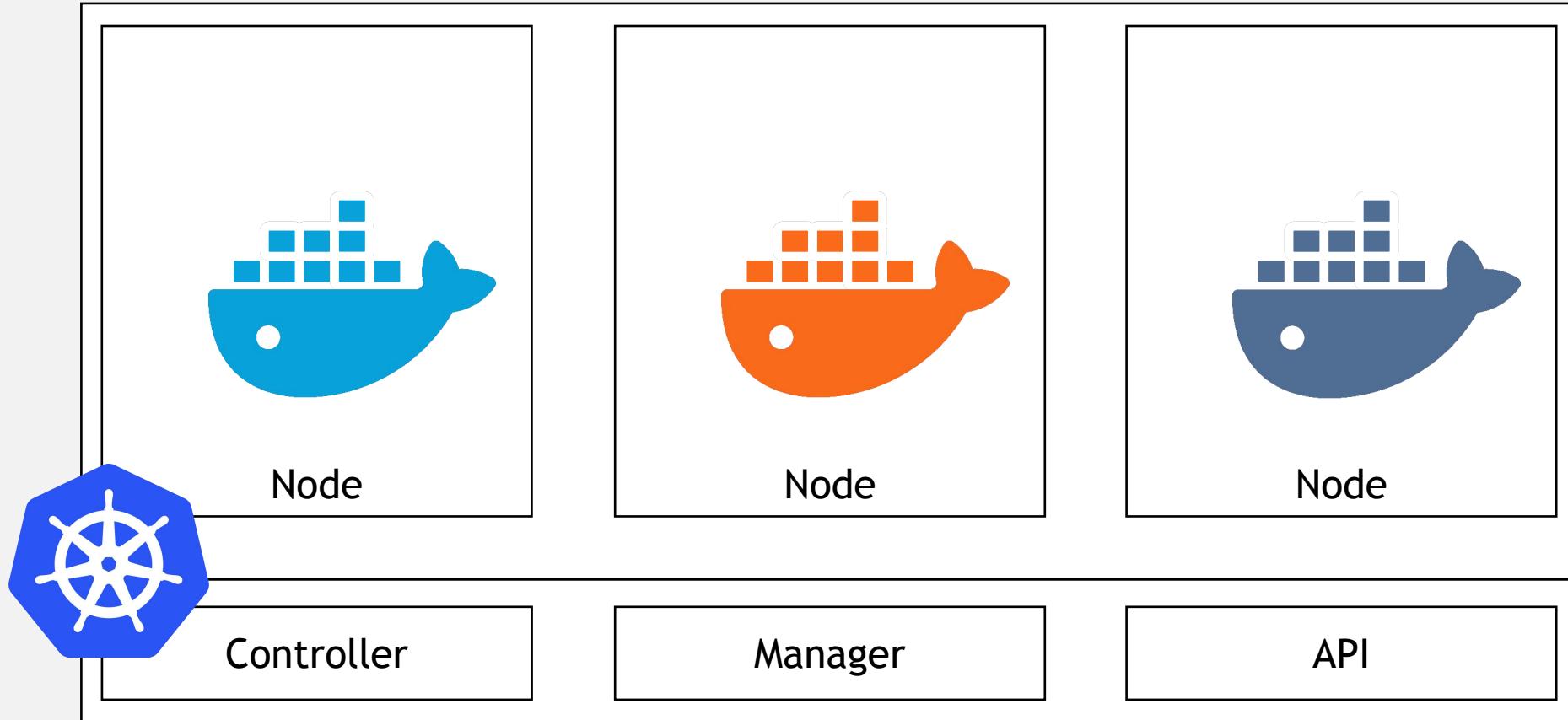
Virtualization



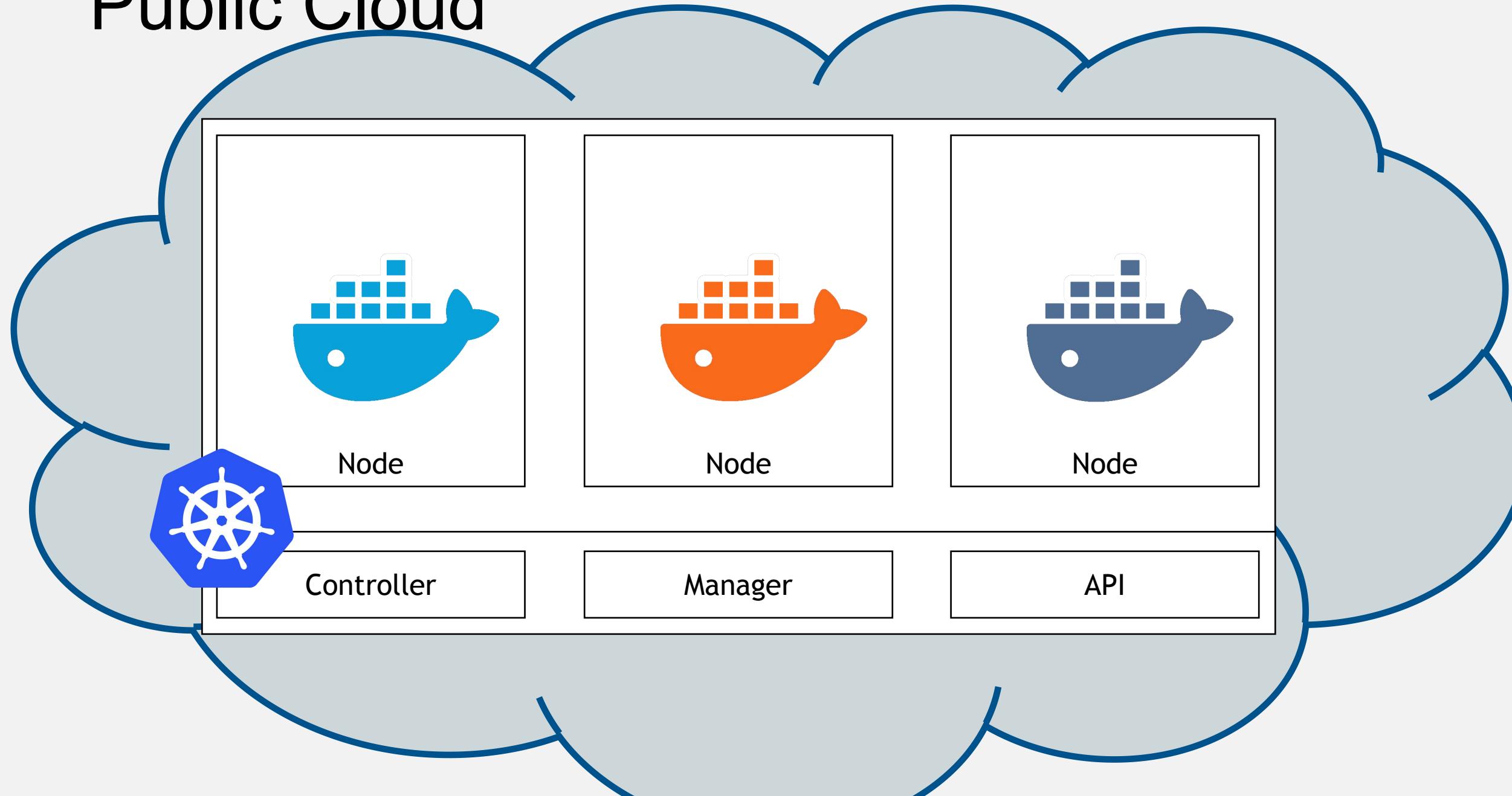
Containers



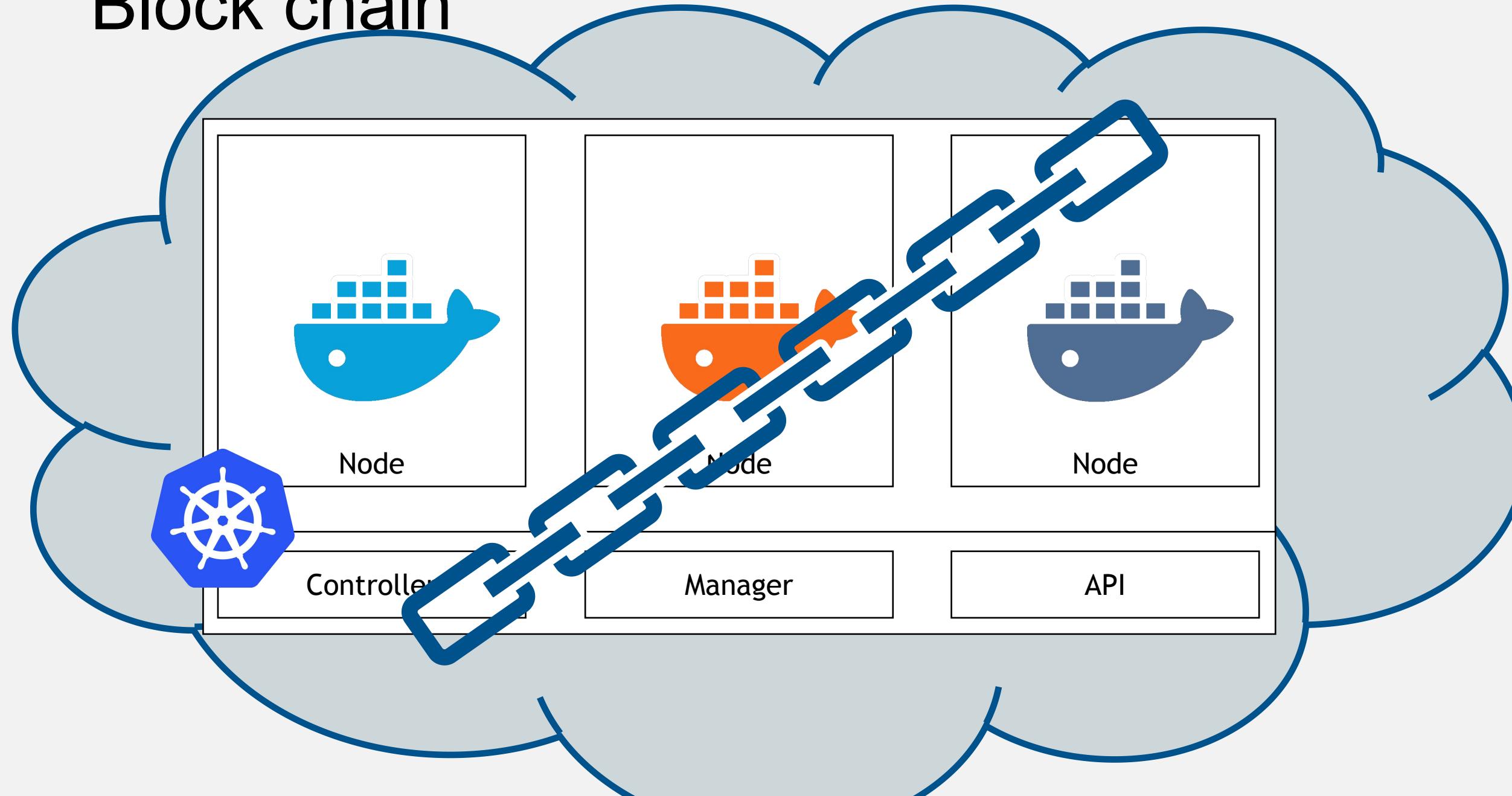
Platform



Public Cloud



Block chain





Source: @dexhorthy

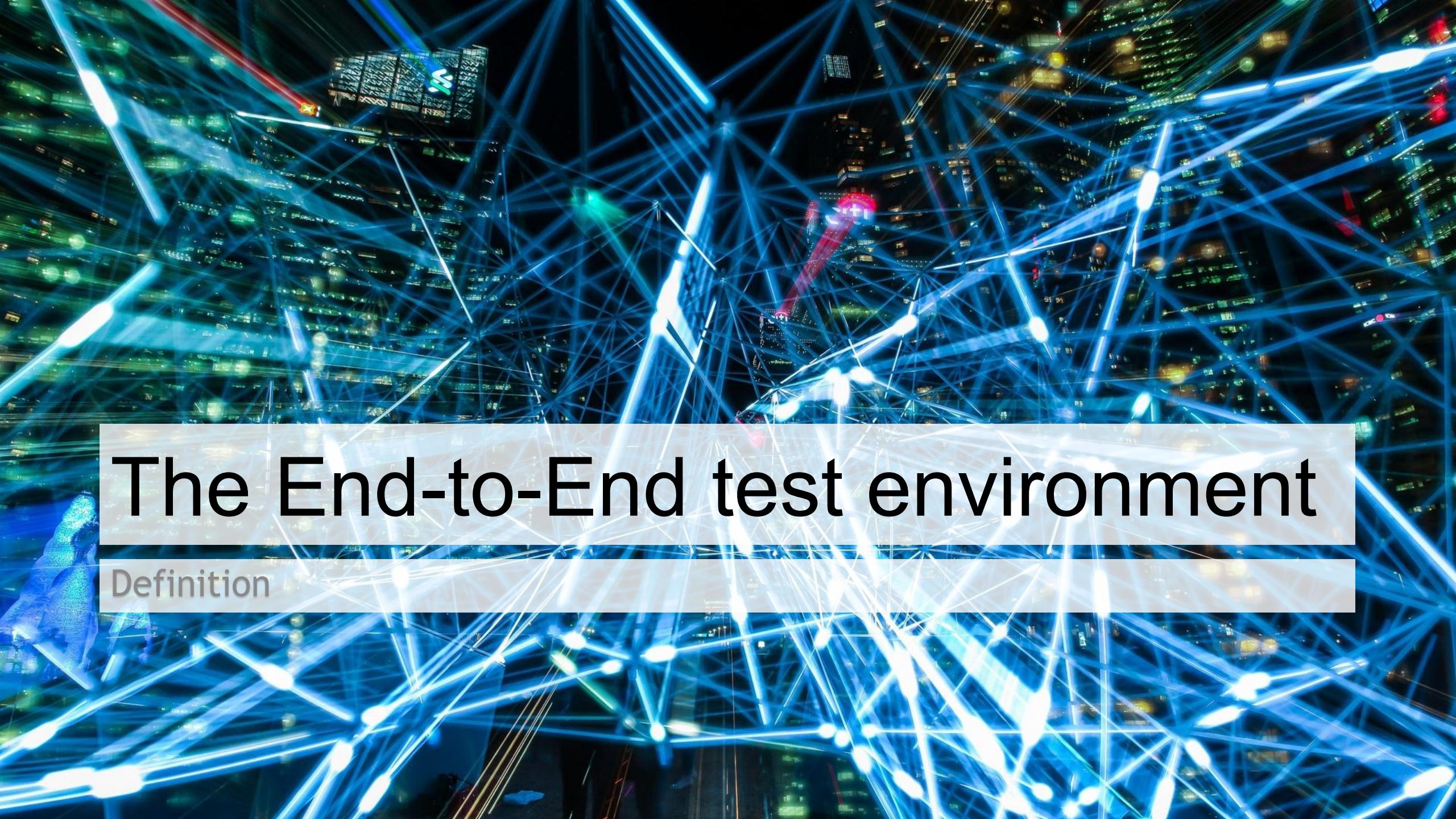


Architecture

- All about production architecture (and infrastructure)
- Less about !production environments
- Requirement: Should be like production

End-to-End test environments

- Definition
- Problem
- Experience
- Reasons to use
- Alternatives
- Questions



The End-to-End test environment

Definition

“A complete integrated environment with production-like systems, with the single purpose of testing”

Aliases

- Chain Environment
- User Acceptance Environment
- Production Acceptance Environment
- Broad-stack Environment
- NGA Environment
- Retailer Test Environment
- Acceptance Environment
- Pre Production Environment
- Enterprise-wide Integration Test Environment
- Full-stack Environment
- Production-like Environment
- Fake news Environment
- Staging

DTAP

- Develop Local development environments
- Test Shared team infrastructure
- Acceptance Shared and integrated environment
- Production The place you want to be



The dead end road

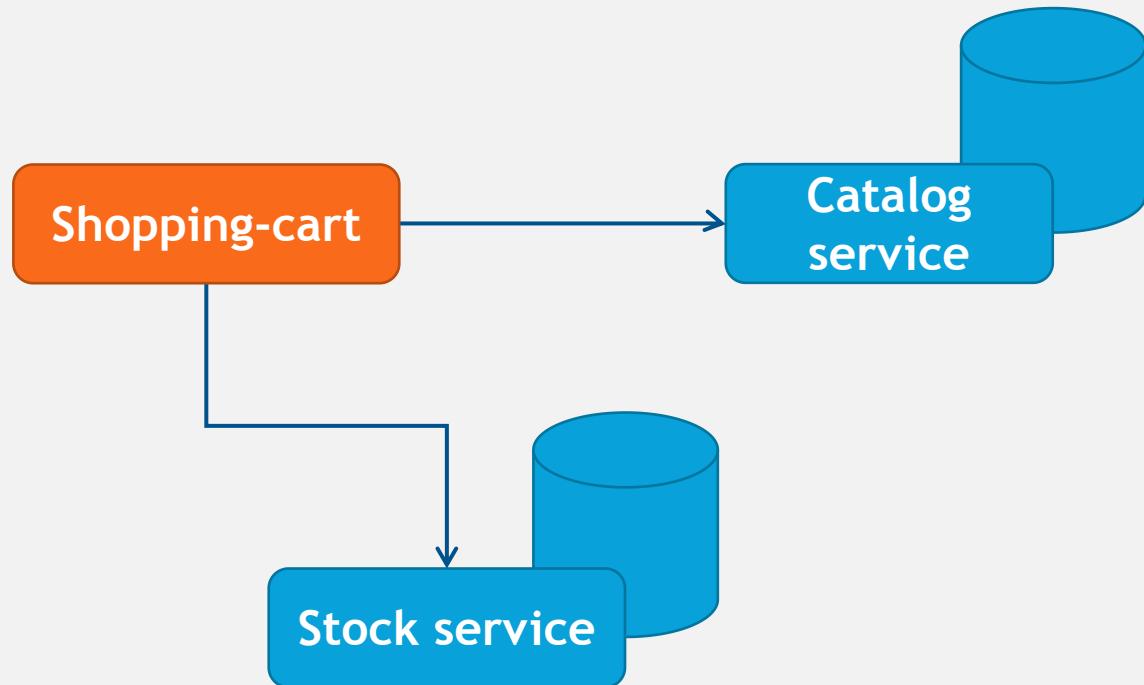
What is the problem?



Problems with E2E

- Test data hell

Test Data hell



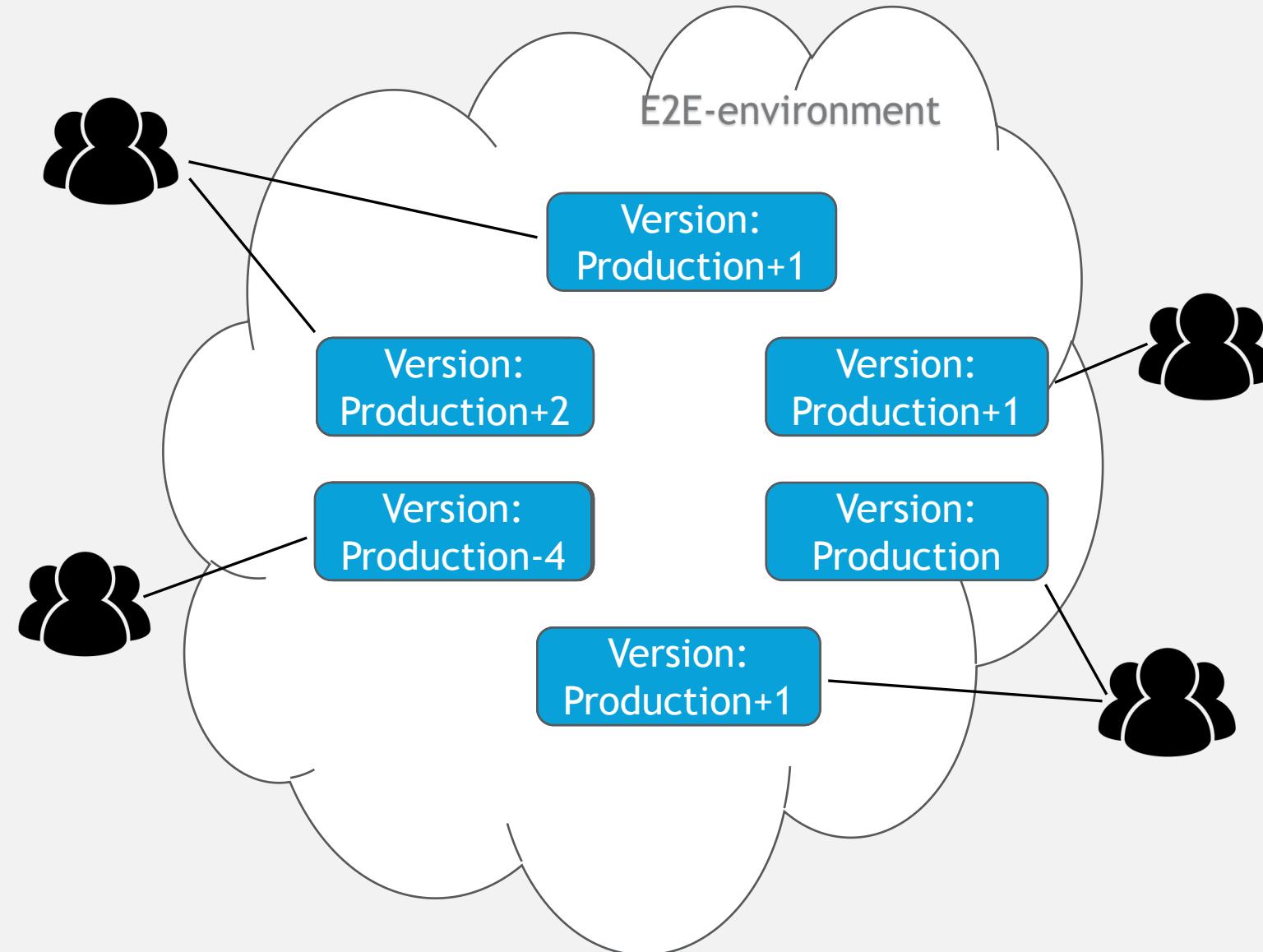
- Arrange test-data
- Consuming means re-arrange data
- Not 'your' test-data, others may change it
- Test-data needs to be correlated

Bye-bye independent DevOps

Problems with E2E

- Test data hell
- Not so “production-like”

Not so “production-like”

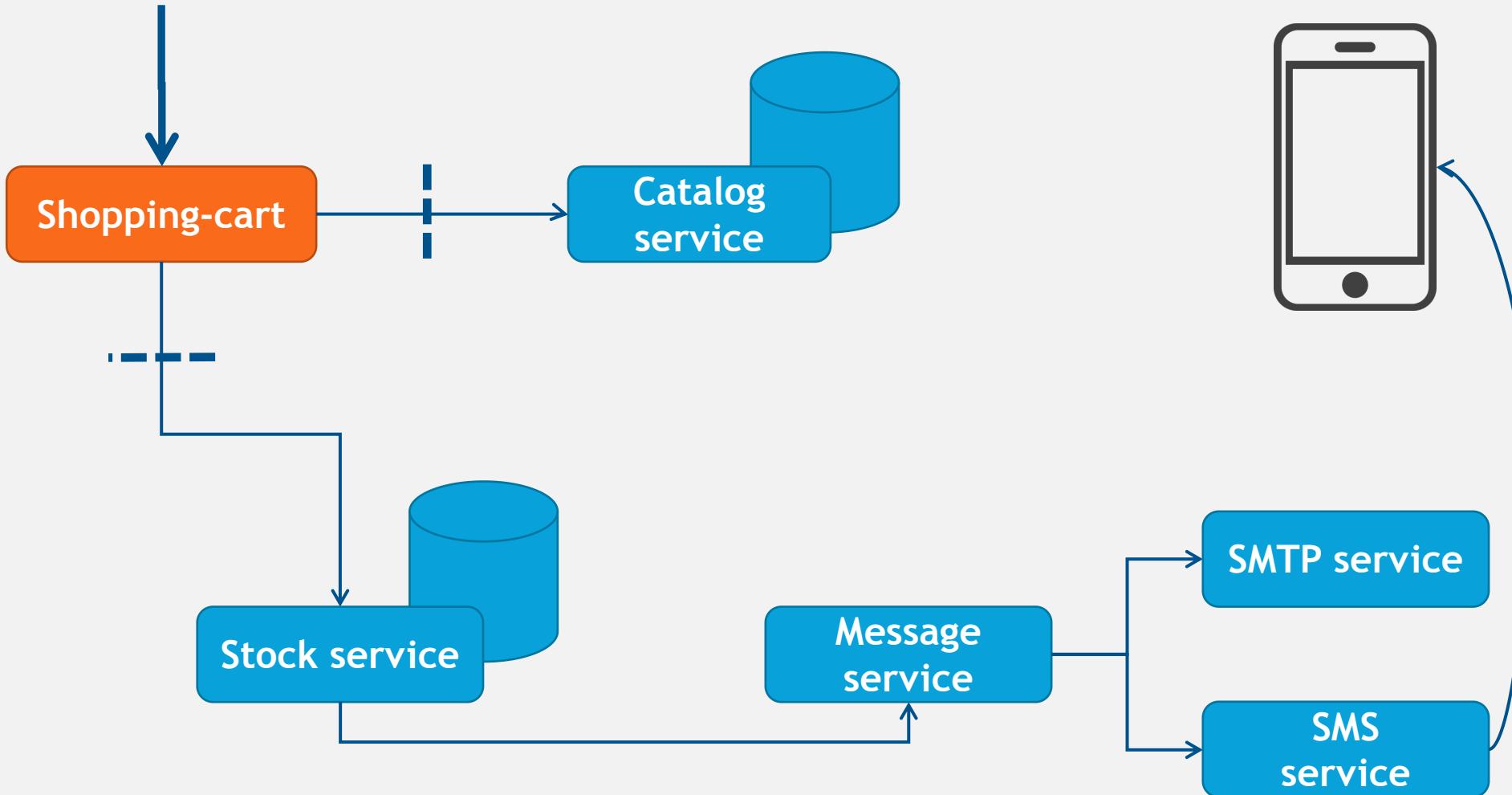


- Simultaneously testing new version
- Multiple updates before deploying in production
- Not deploying in the End-to-End environment

Problems with E2E

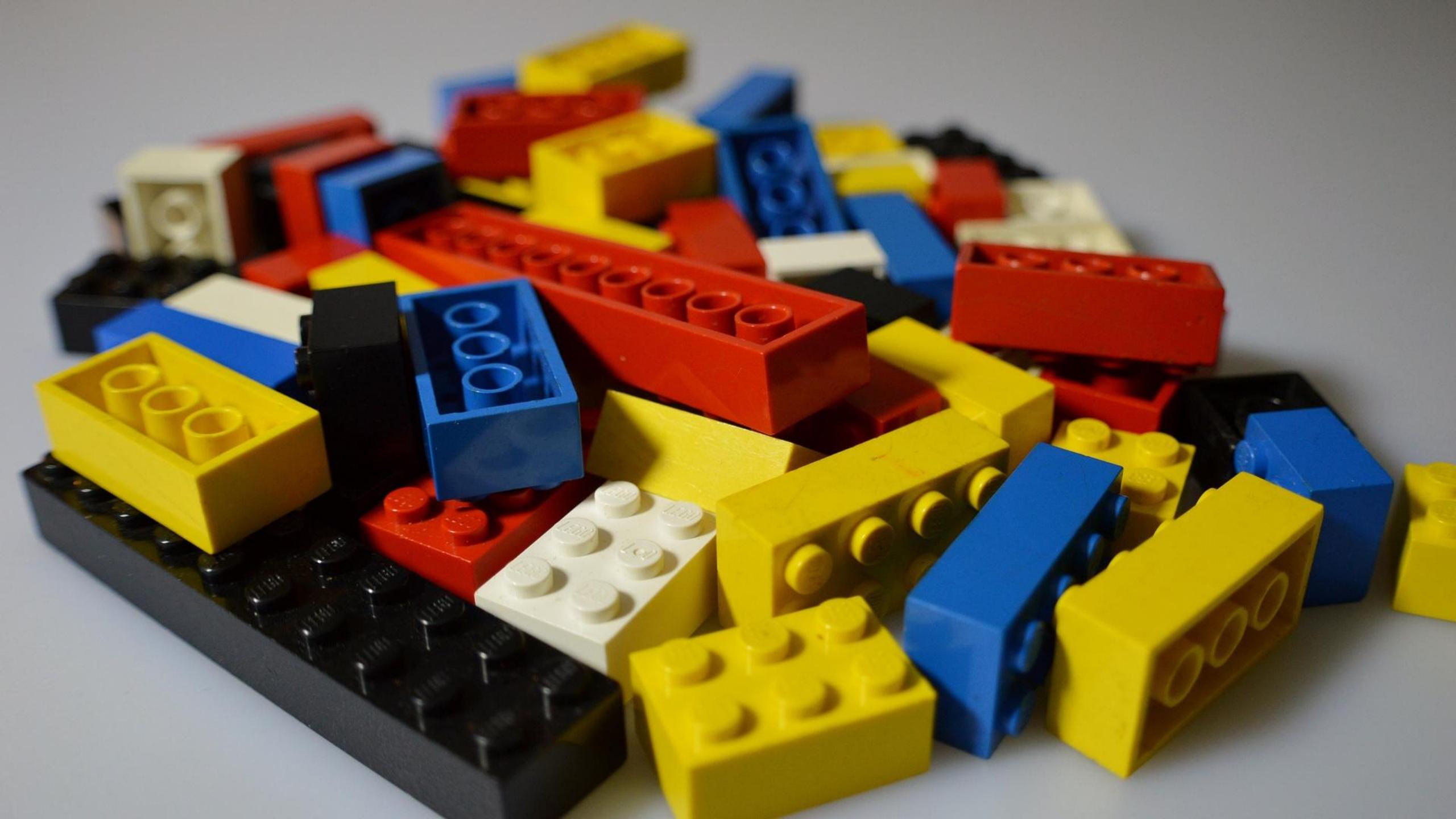
- Test data hell
- Not so “production like”
- Test scope

Test scope



Problems with E2E

- Test data hell
- Not so “production like”
- Test scope
- Unstable tests
- Slow Feedback time
- Increased time-to-market
- Costs

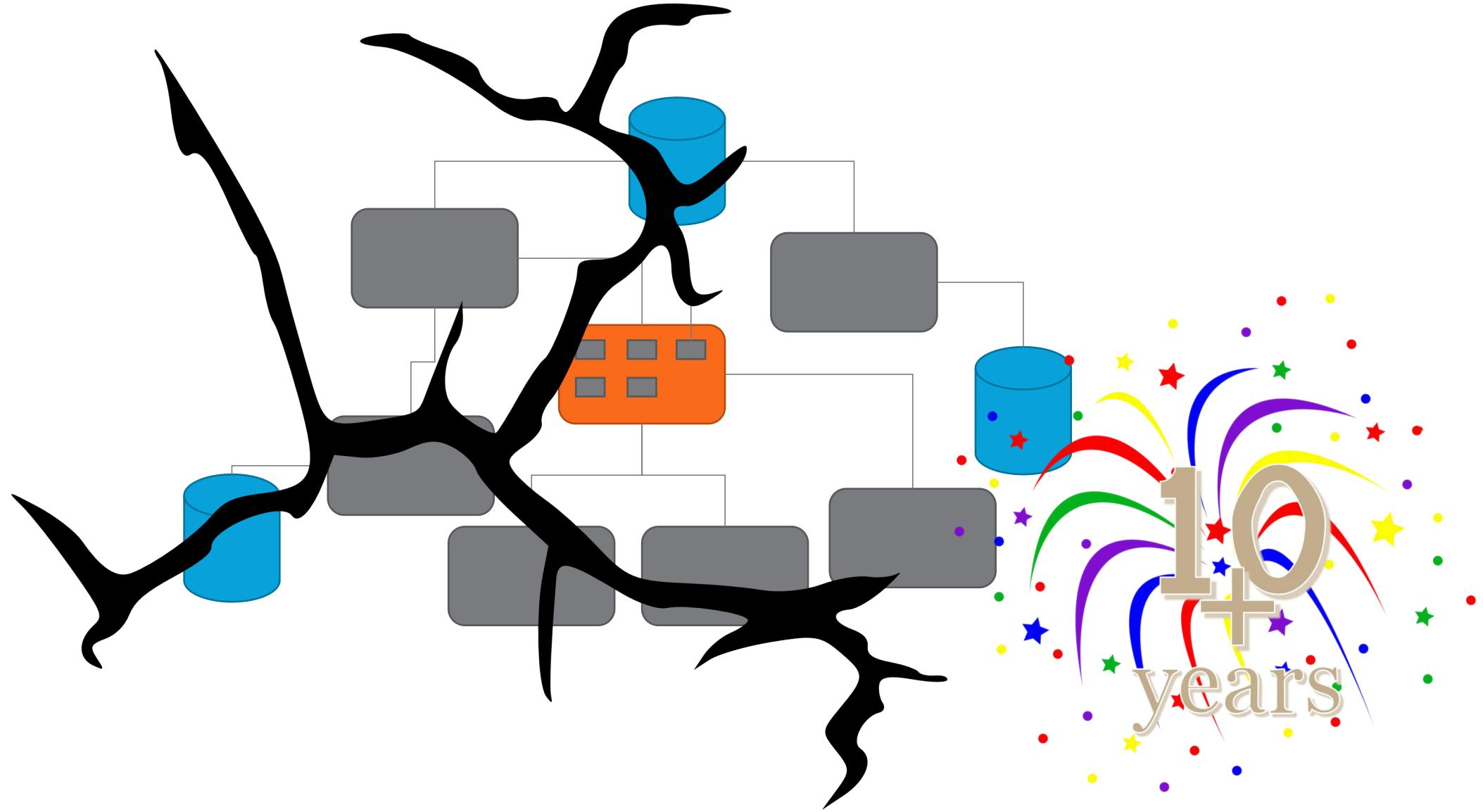




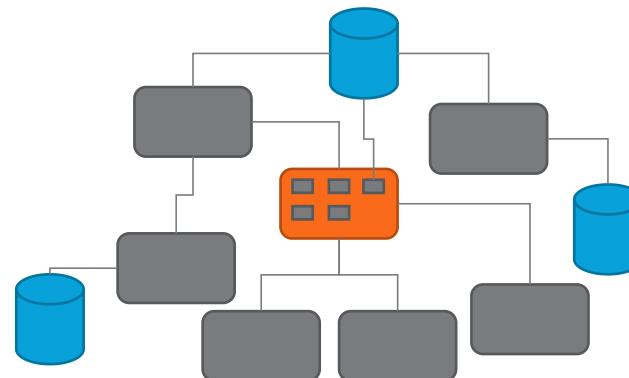
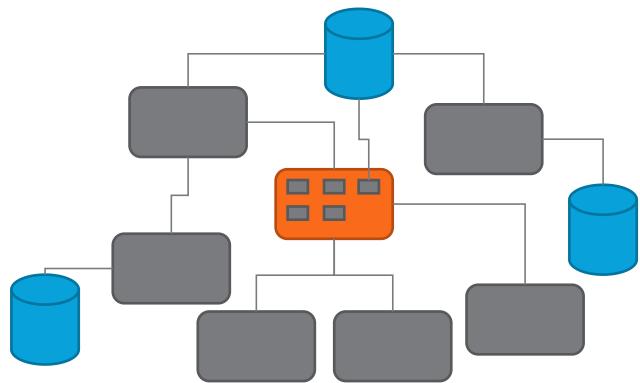
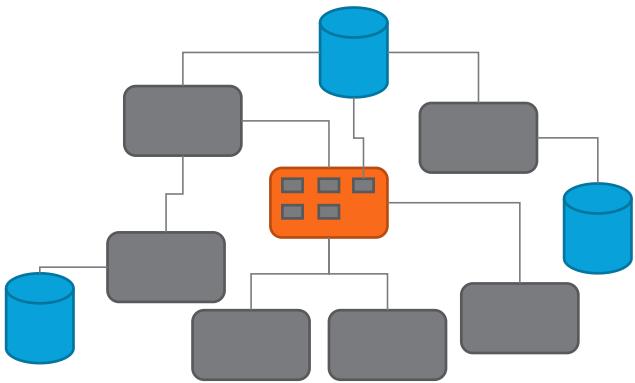
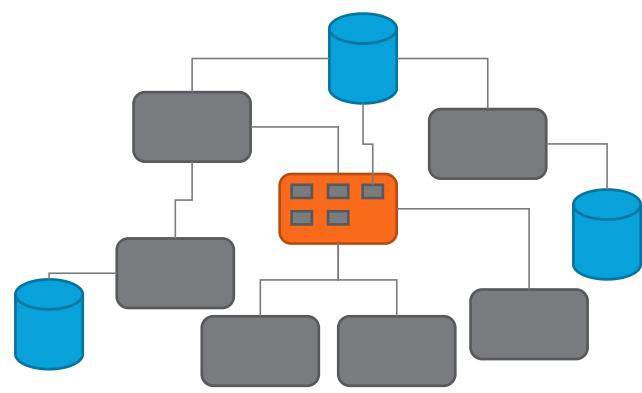
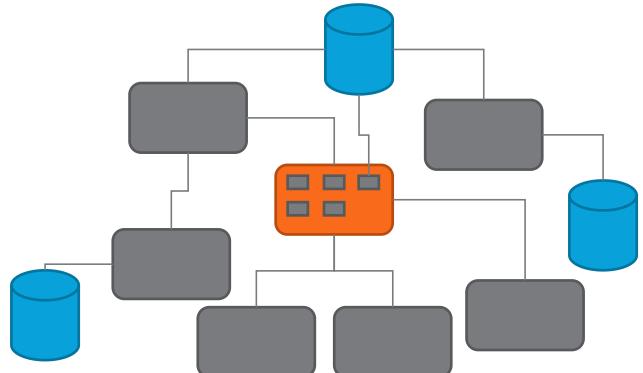
@ a big Dutch Bank

In love with End-2-End environments

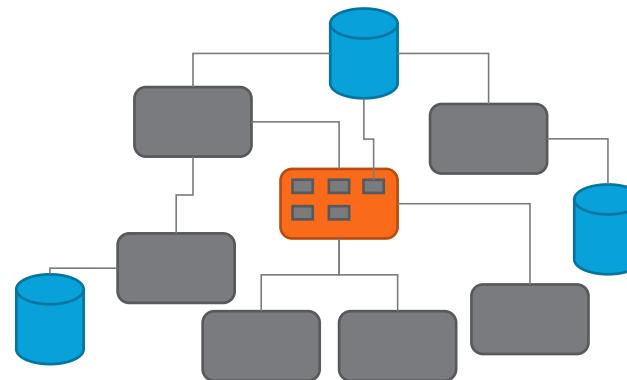
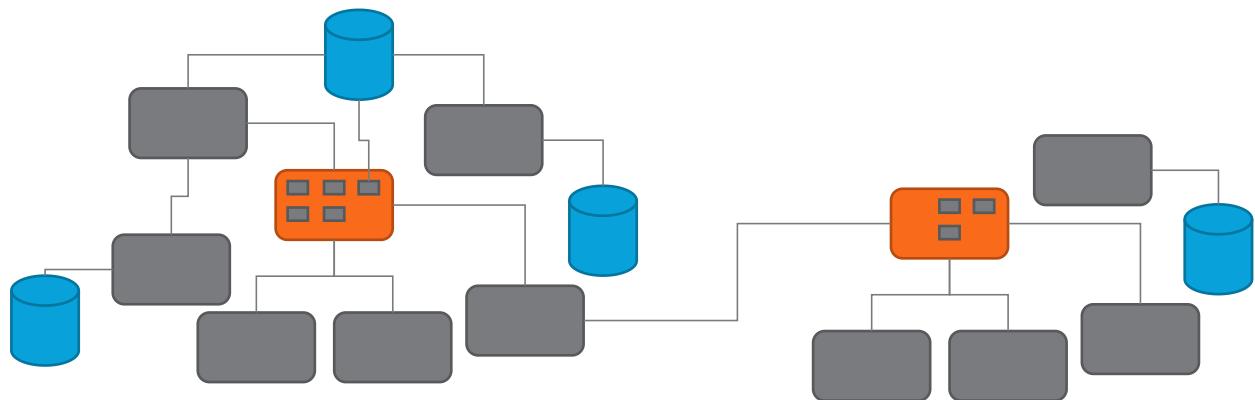
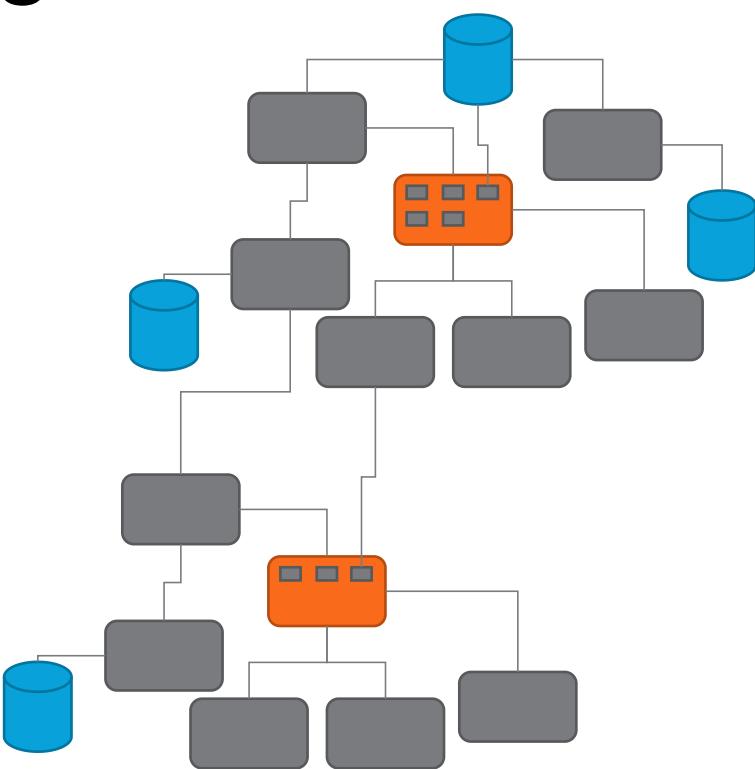
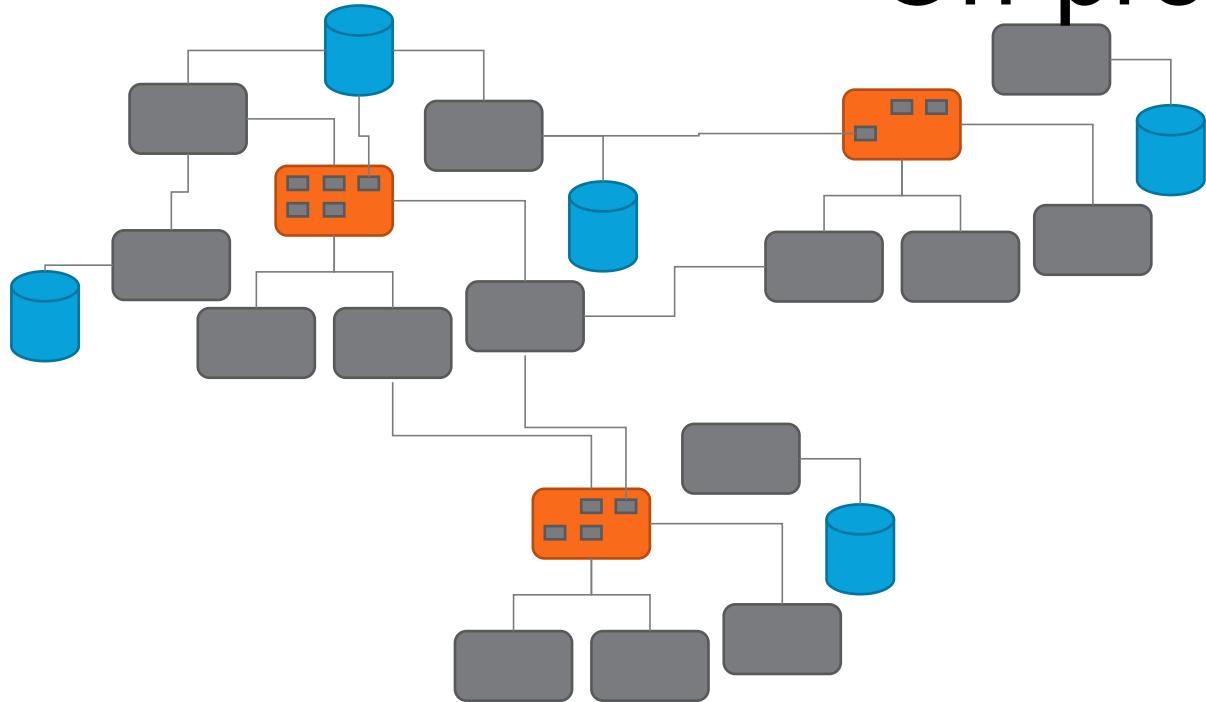
On-premises



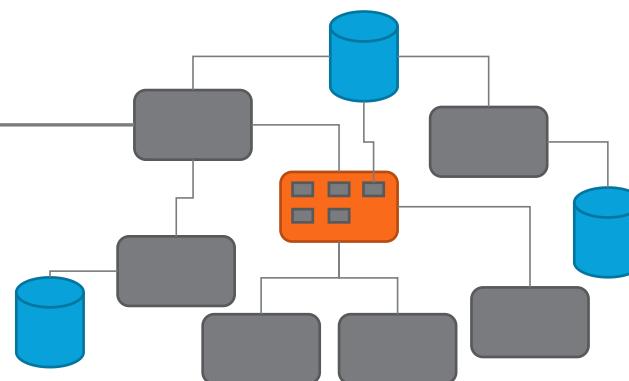
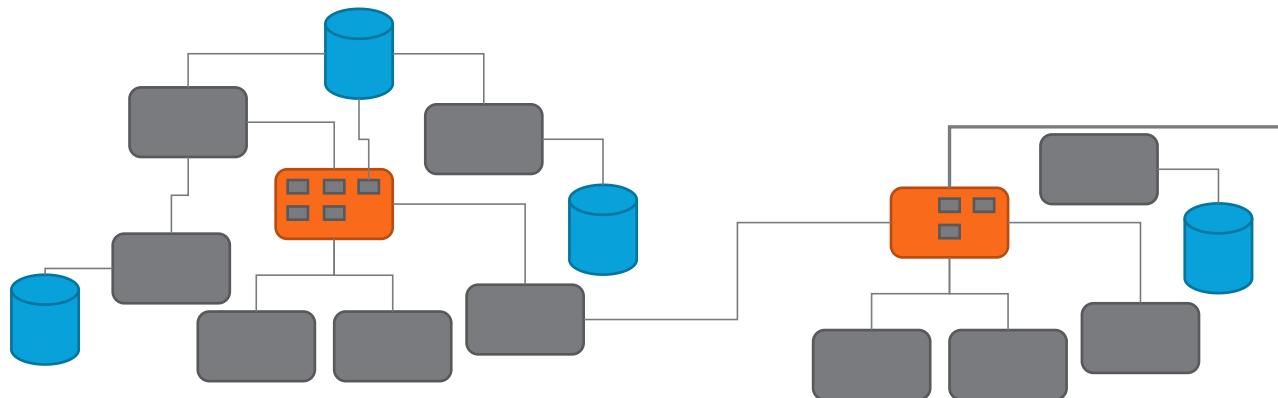
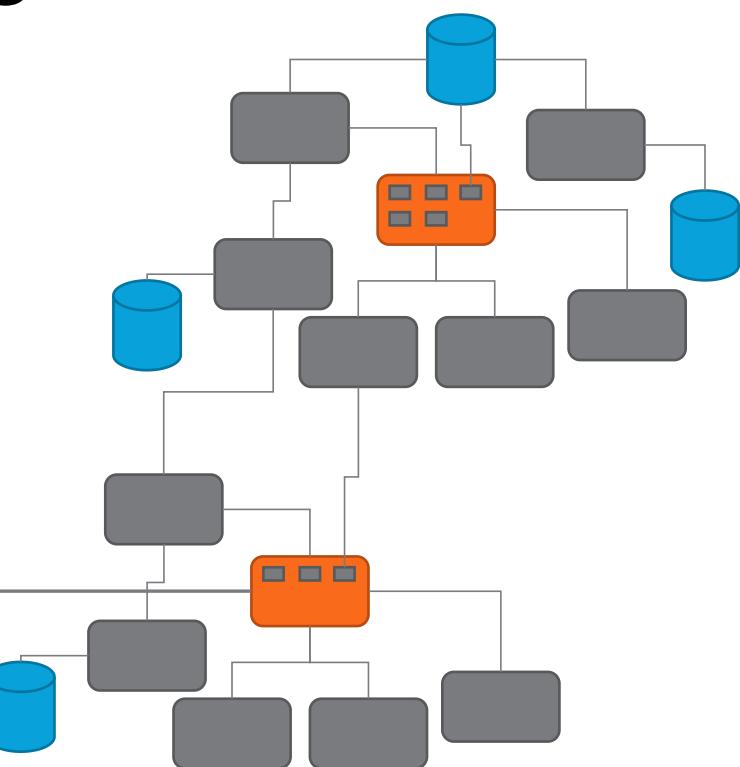
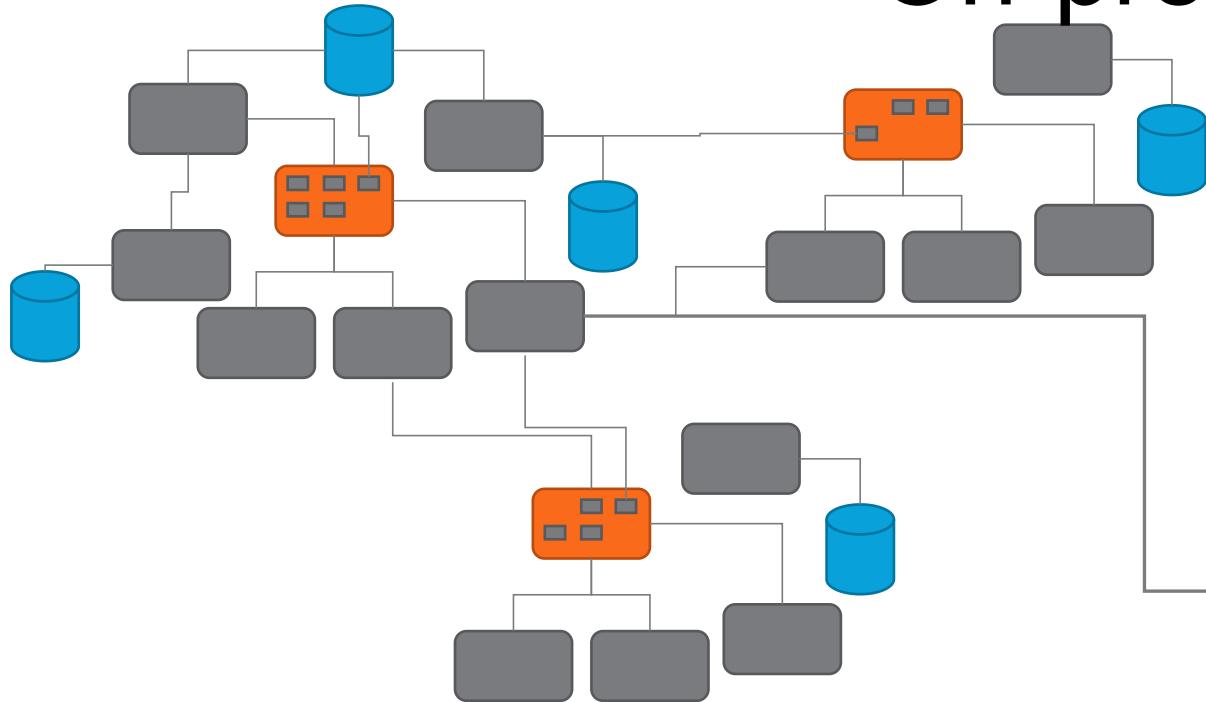
On-premises



On-premises



On-premises





Support conversation to a team

Dev: Hey, can you help me out. Our flow is not working on the environment 3, can you have a look if the platform is the cause

Me: Sure. I don't see any problem with the platform. Can you show me the broken flow on my device?

Dev: Yes.... Hey, now it's working. O, wait. You are looking at environment 3 small.

Me: Yes, that is what you said, environment 3...

Dev: No, I meant environment 3 **big**

Me: But that's the same platform, it's environment 3.

Dev: No, it is a **small** environment 3

Me: But it connects to the same services for us.

Dev: But not to us, in environment 3 small we connect to services in environment 5 big

Me: What !!?!?





Migrating Platform to the public Cloud



Pivotal
Cloud Foundry[®]



~~End-to-End
Environment~~

Why?

What are the reasons teams rely on End-to-End environments

Because of.... reasons

- Test data
- Trust
- Immaturity
- Compensation
- Requirement

Test data

*“Tests with semantic correct data
can only be run in a End-2-End test
environment.”*

HTTP : Get
/customer/<customerId>

Response-body :

```
{  
  "id"      : "1",  
  "name"    : "Roy",  
  "surname": "Braam"  
  "type"    : "??"  
}
```

- What “type” to expect?
 - “Wholesale”
 - “Retail”
 - “Private”
 - “Boring”
 - “Bad”
 - “Greedy”

Improve

- Make clear specifications
- Stricter contracts
- Move test-data from e2e environment to stubs
- Business logics in the lower level of the Test Pyramid
- Build for failure

Trust



“I want to see it working before I go to production”

What I hear:
*“I don't trust the automated tests
we implemented.”*

Or:

*“We don't have automated
integration tests, we like manual
testing labour”*

!!Dangerous!!

What if you don't
“Talk the Talk, Click the Click”?

Gain trust

- Automate, automate, automate
- Bug: create a failing test case
- Invest in tests to gain trust
- Move tests down the pyramid
- Proof no new issues are found

Immaturity



“System X had a new release that broke our system, even though the contract is the same.”

Improve by

- Communicate
- Stricter and clearer contracts
- Consumer Driven Contract testing
- Resilience: Build for failure
- Automate a 1-on-1 integration system if needed

A large, textured tree trunk dominates the left side of the frame. A white-painted metal arm and hand sculpture are wrapped around the trunk, with the hand gripping the branch. The sculpture is mounted on a wooden post with a chevron pattern. In the background, a paved path leads through a park-like setting with other trees and a bus stop visible.

Compensation

*“When they release a new version
we first want to test our system,
they often break things.”*

Improve by

- Put the responsibility where it belongs
- Resilience
- Run expectations against environment

A close-up photograph of a man's torso and hand. He is wearing a dark suit jacket, a light-colored shirt, and a striped tie. His right hand is pointing directly at the viewer with his index finger. The background is blurred, showing warm, out-of-focus lights.

Requirement

“Our stakeholders demand that we first show it in an End-2-End environment, with that we guarantee our quality”

Improve by

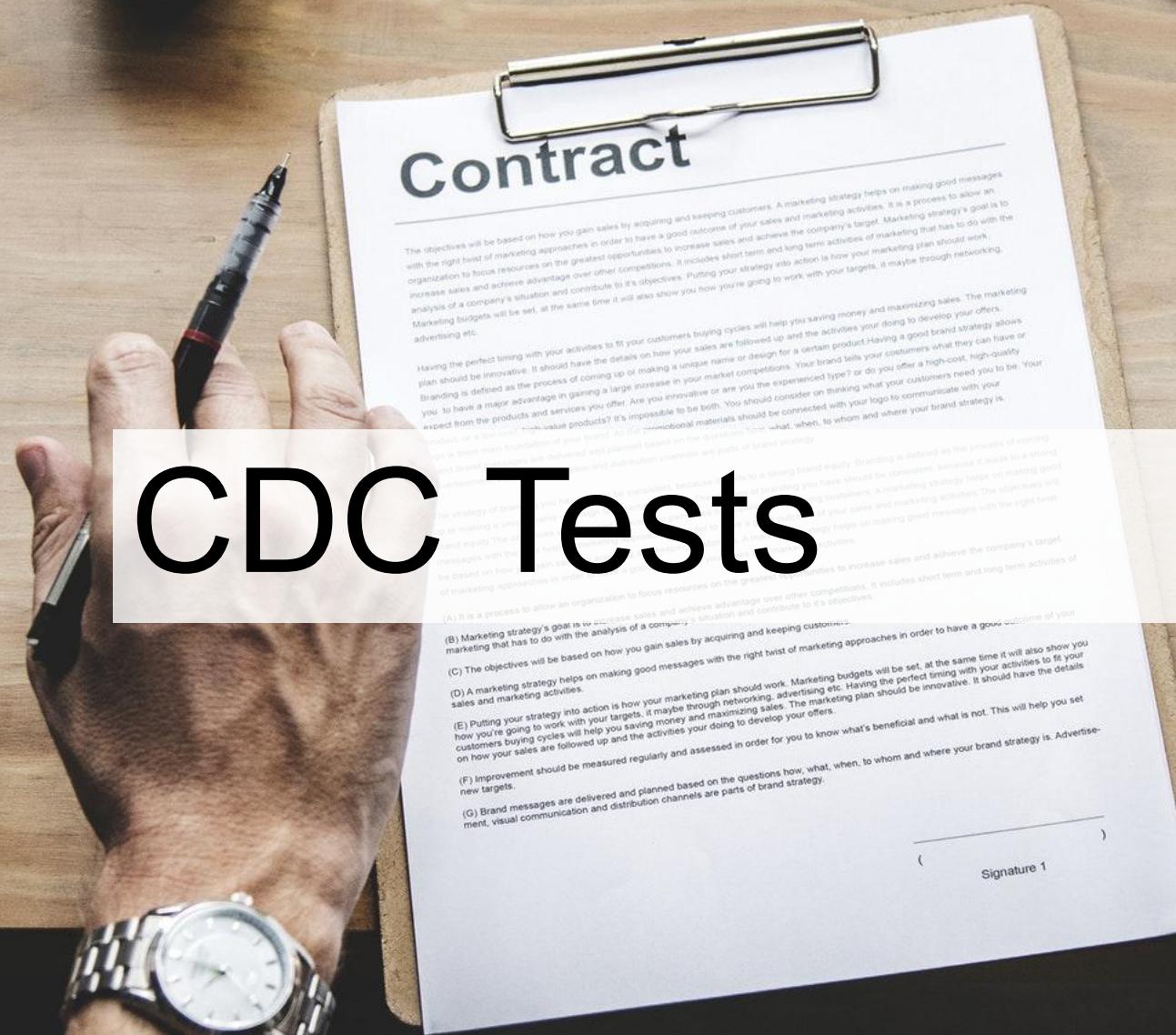
- Are stakeholders telling us how to guarantee quality?
- Gain their trust
- Improve the (non technical) process

A row of doors against a dark background. All doors are dark grey except for one which is bright red. This visual metaphor represents the concept of alternatives, where one option stands out from the rest.

Alternatives

Ways to minimize usage of E2E environments

CDC Tests



CDC Tests

- Consumer Driven Contract Tests
- Testing pattern
- Tests the agreement between Consumer and Producer
- Isolation
- Consumer offers expectations
- Producer tests with these expectations

“Contract testing is a technique for testing an integration point by checking each application in isolation to ensure the messages it sends or receives conform to a shared understanding that is documented in a contract.”

<https://docs.pact.io>

/customers/id-unknown

HTTP: 404 with empty body

/customers/

HTTP: 400

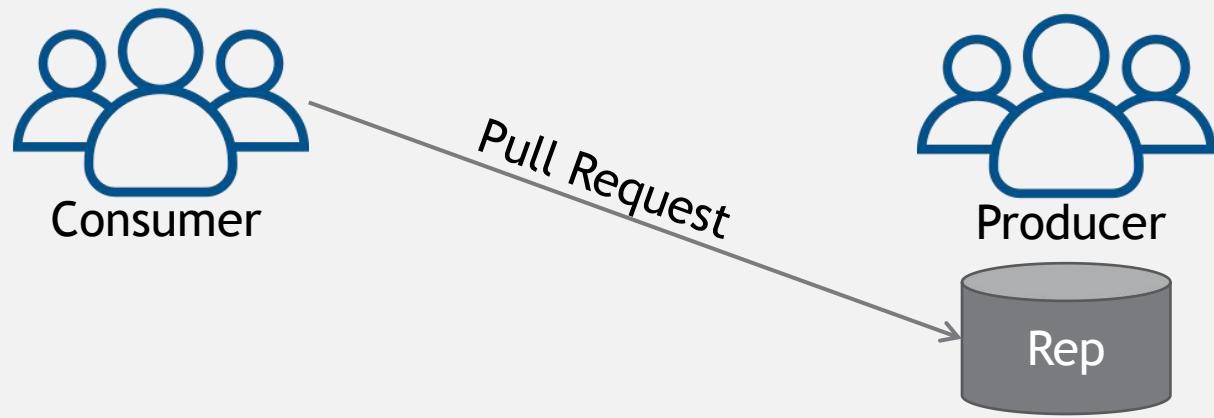
/customers

**HTTP: 200 list of
customers**

/customers/id

**HTTP: 200 with valid JSON
body**

CDC Tests



- Expectations provided as tests by consumer
- Producer: Trust to not break consumers
- Consumer: Guarantee to not break contracts
- API design
- API usage

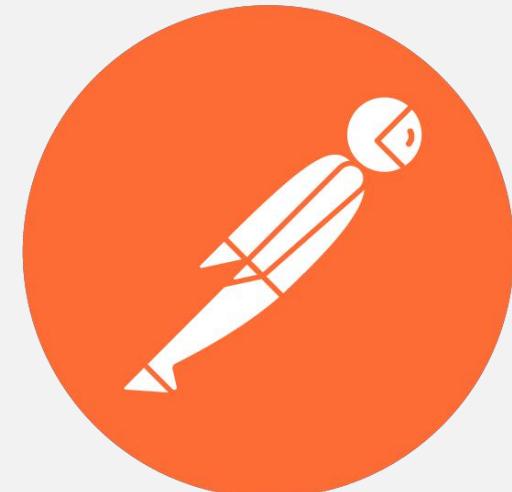
	E2E Tests	Mock Tests	CDC Tests
Isolation	-	+	+
Complexity	-	+	+
Test Data Setup	-	+	+
Testing Data Semantics	+	-	-
Feedback time	-	+	+
Stability	-	+	+
Reveal Unused Interfaces	-	-	+
Well-Fittedness	-	-	+
Costs	-	+	+
Maintenance Effort	-	+	+

CDC tools

- Spring Cloud Contract (SCC)
- Pact
- Postman/Newman



PACT
FOUNDATION



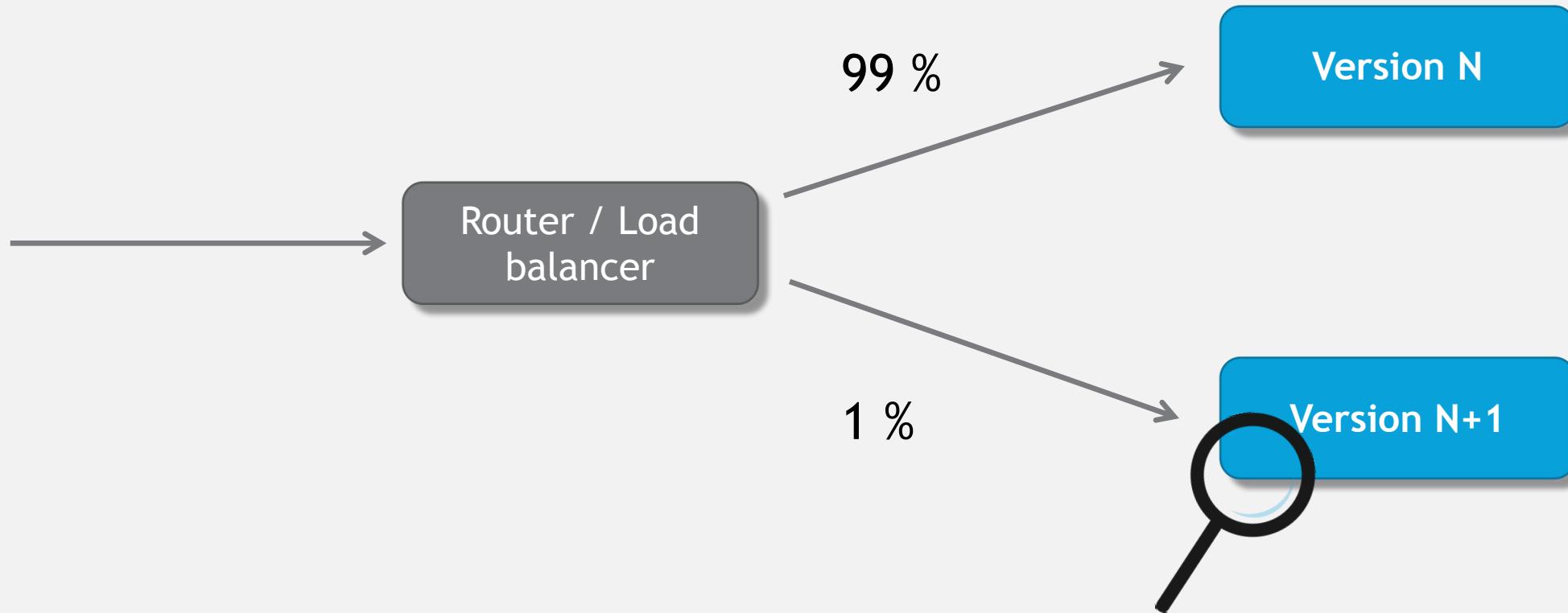
A photograph of a hiker standing on a rocky outcrop, looking out over a vast, misty mountain range under a bright blue sky with scattered white clouds. The hiker is wearing a red jacket, black pants, a black cap, and a brown backpack. A large white rectangular box is overlaid on the lower half of the image, containing the text.

Minimize Risks

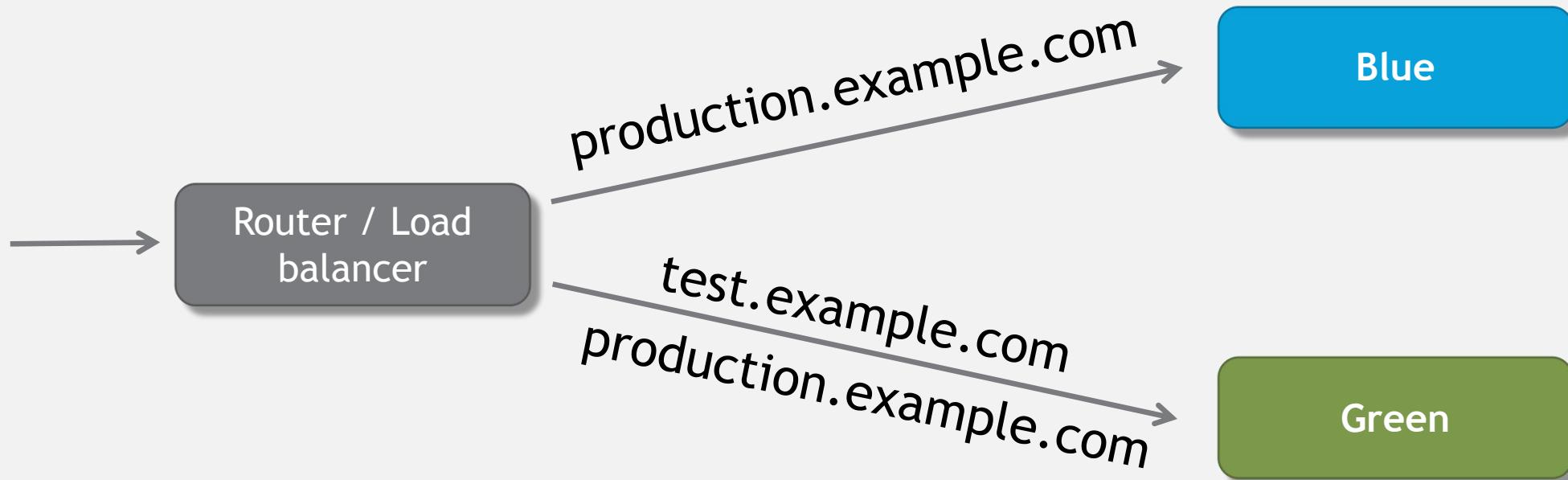
Minimize Risks

- Decouple deployment from release
- Feature toggle
- (Beta)Test groups
- Monitor and threshold alerts
- (Automated) Rollback
- Canary releases
- Blue/Green deployments

Canary release



Blue/Green deployments



**DEAD
END**

A long exposure photograph of a graffiti tunnel. The floor is dark and textured. The walls are covered in colorful, abstract graffiti. A large, illuminated letter 'P' is the central focus, glowing with red and white lights. The 'P' is outlined in red and filled with white, with bright highlights at the top left and bottom right. The background is dark, making the illuminated 'P' stand out.

Feedback

Sources

- Practical test pyramid:
<https://martinfowler.com/articles/practical-test-pyramid.html>
- 7 reasons for CDC:
<https://reflectoring.io/7-reasons-for-consumer-driven-contracts/>
- Thoughtworks tech radar
<https://www.thoughtworks.com/radar/techniques/enterprise-wide-integration-test-environments>