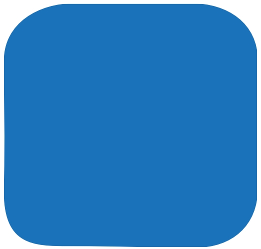
BUILDING SYSTEMS THAT PIVOT

Zhamak Dehghani
@zhamakd

Sam Newman
@samnewman

DO YOU HAVE AN ARCHITECTURE PROBLEM?

DO YOU HAVE AN ARCHITECTURE PROBLEM?



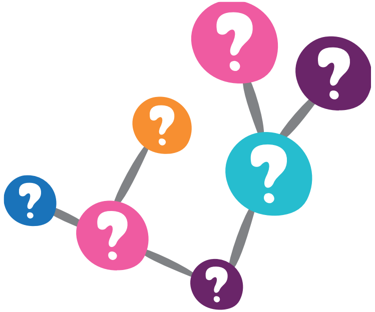
Overhead of releasing a large software

DO YOU HAVE AN ARCHITECTURE PROBLEM?

Infrequent releases when orchestrating many dependent teams

🟦 Large software release

DO YOU HAVE AN ARCHITECTURE PROBLEM?



High risk release with untested dependencies

Multiple teams orchestration

Large software release

DO YOU HAVE AN ARCHITECTURE PROBLEM?

Untested system dependencies

Multiple teams orchestration

Large software release

ARCHITECTURE PROBLEM!

A LONG AND BROKEN DIALOGUE

A LONG AND BROKEN DIALOGUE

6 Months

>50%

A LONG AND BROKEN DIALOGUE

6 Months

<30%

>50%

AGENDA

- What is **Microservices** Architecture
    - *benefits*
- How to **implement** a Microservices architecture
- How to **operate** a Microservices architecture

CONTINUOUS DELIVERY AS AN ENABLER

Addison Wesley Signature Series
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
A Martin Fowler Signature Book
Foreword by Martin Fowler

CONTINUOUS DELIVERY AS AN ENABLER

- *automated build*

The Addison Wesley Signature Series
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
A Martin Fowler Signature Book
Foreword by Martin Fowler

CONTINUOUS DELIVERY AS AN ENABLER

- *automated build*
- *automated deployment*

Addison Wesley Signature Series
CONTINUOUS DELIVERY
RELIABLE SOFTWARE RELEASES THROUGH BUILD, TEST, AND DEPLOYMENT AUTOMATION
JEZ HUMBLE
DAVID FARLEY
A MARTIN FOWLER SIGNATURE BOOK
Foreword by Martin Fowler

CONTINUOUS DELIVERY AS AN ENABLER

- *automated build*
- *automated deployment*
- *automated environment provisioning*

The Addison Wesley Signature Series
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
A Martin Fowler Signature Book
Foreword by Martin Fowler

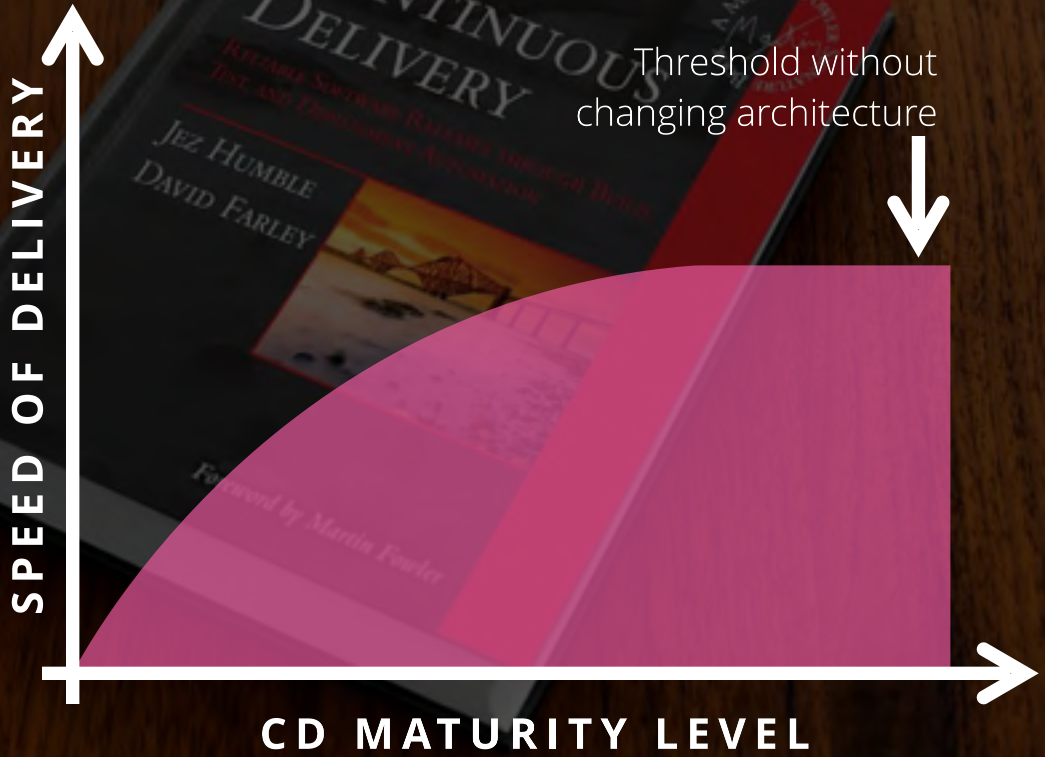
CONTINUOUS DELIVERY AS AN ENABLER

- *automated build*
- *automated deployment*
- *automated environment provisioning*
- *automated testing*

The Addison Wesley Signature Series
Continuous Delivery
Reliable Software Releases through Build, Test, and Deployment Automation
Jez Humble
David Farley
A Martin Fowler Signature Book
Foreword by Martin Fowler

CONTINUOUS DELIVERY AS AN ENABLER

- *automated build*
- *automated deployment*
- *automated environment provisioning*
- *automated testing*

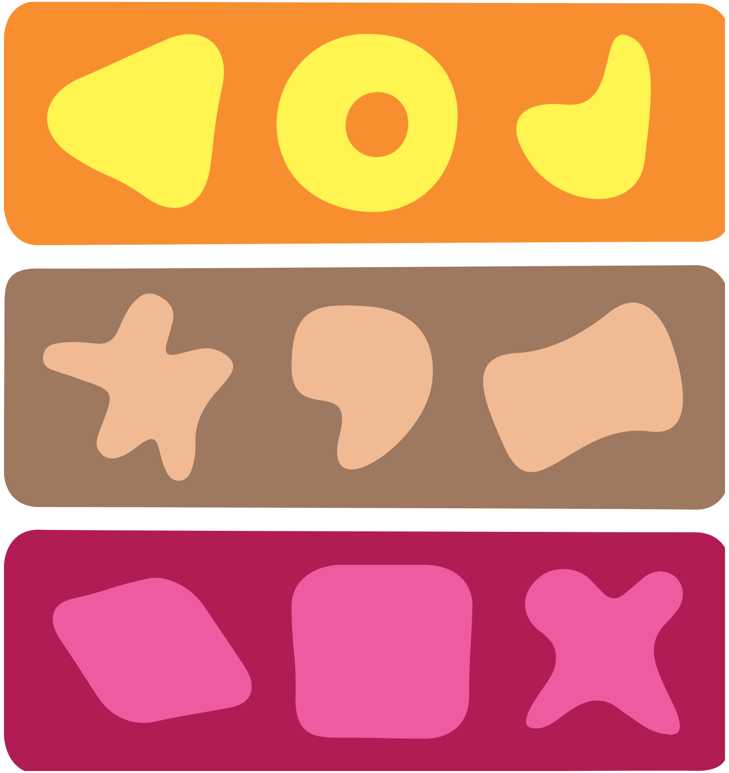


ARCHITECTURE

MONOLITHIC SYSTEM

MONOLITHIC SYSTEM

LAYERED/TIERED ARCHITECTURE



User Interface

Channels

Application

Business Logic & Rules

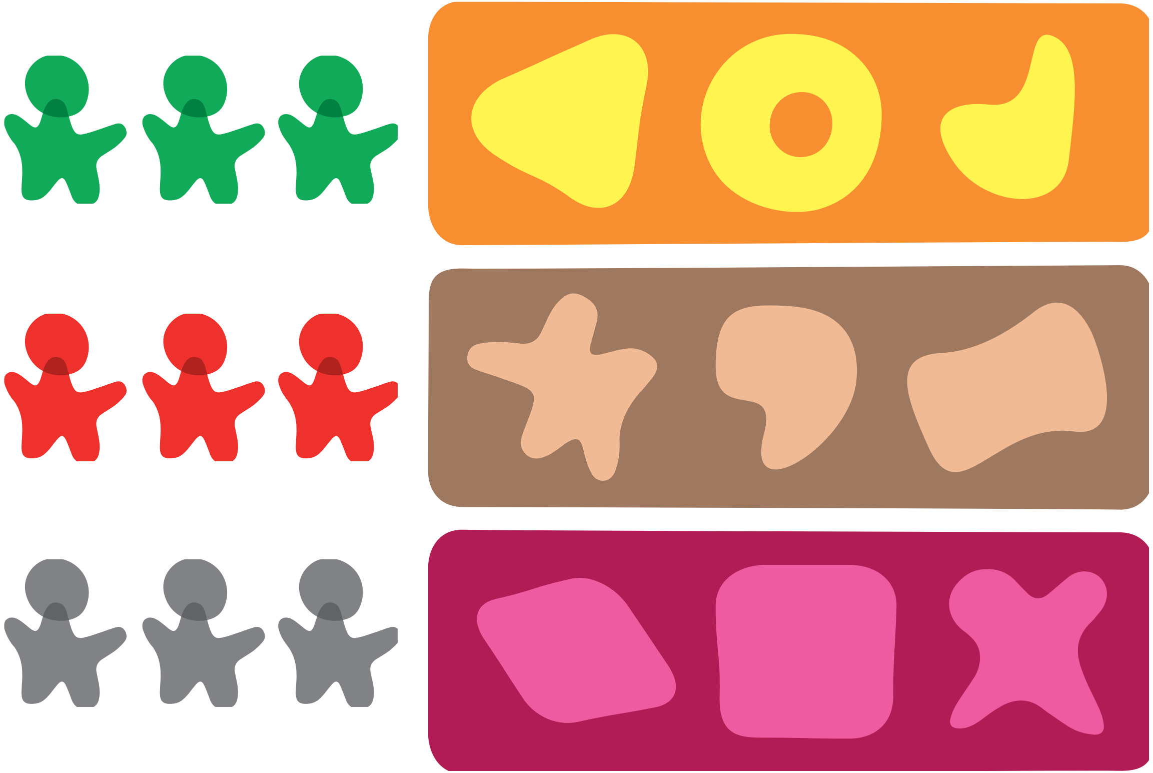
Middleware

Services platform

Database

Systems of Record

LAYERED/TIERED ARCHITECTURE

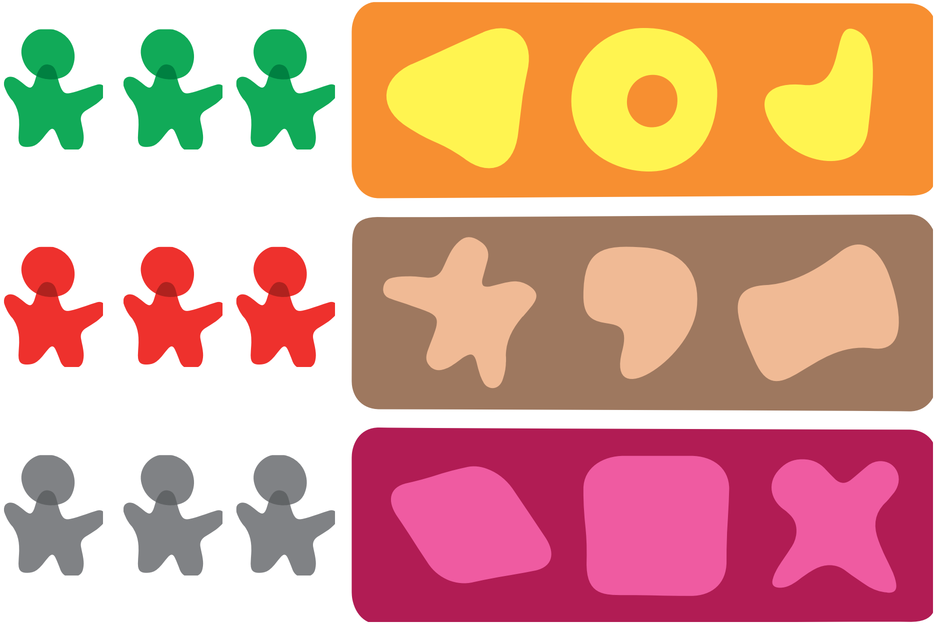


User Interface
Channels
Application

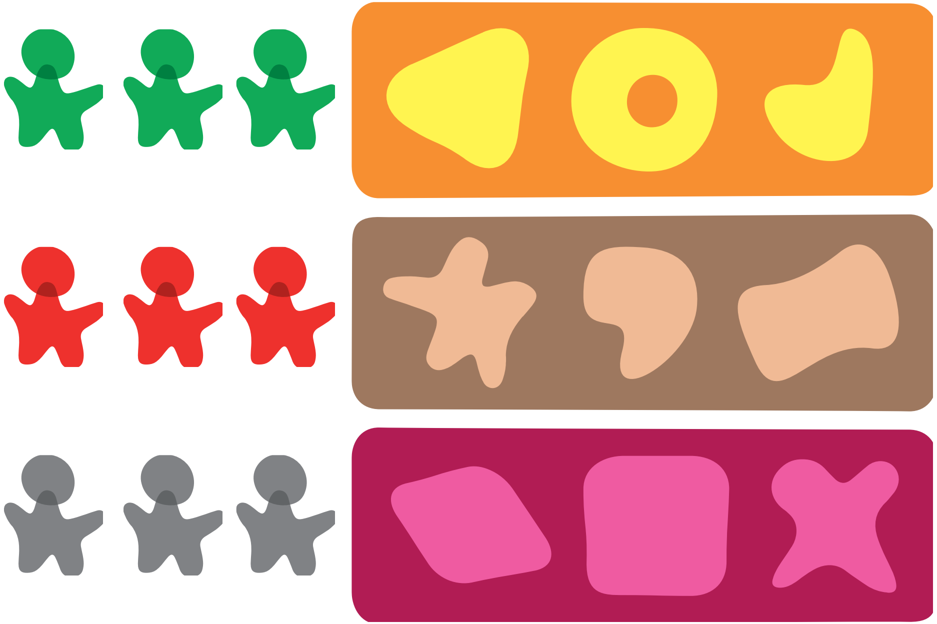
Business Logic & Rules
Middleware
Services platform

Database
Systems of Record

CHALLENGES OF THE TIERED ARCHITECTURE

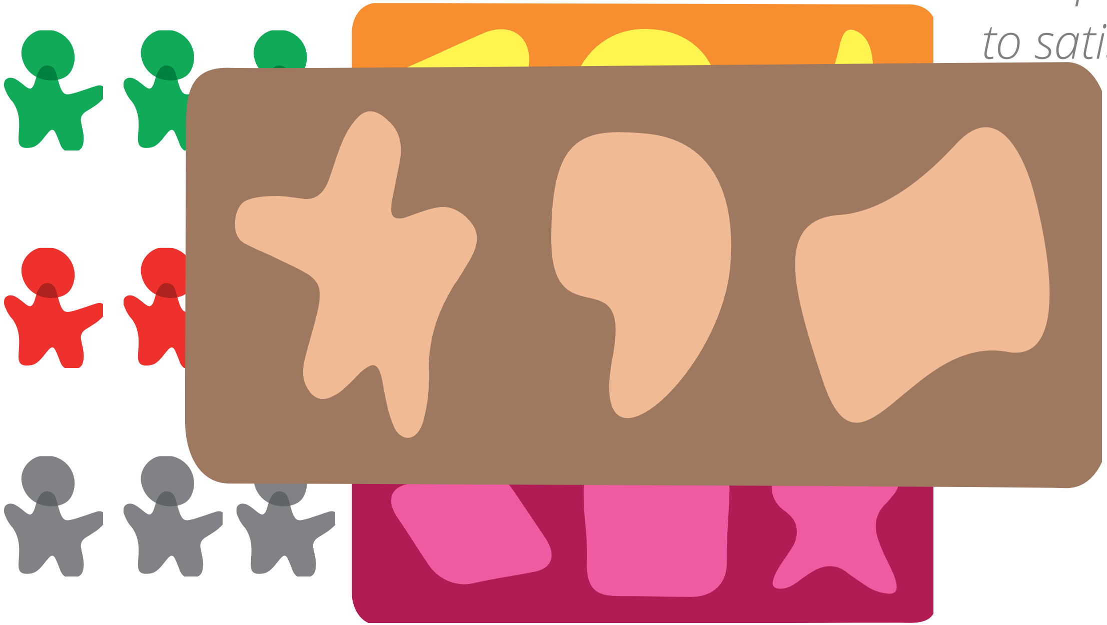


CHALLENGES OF THE TIERED ARCHITECTURE



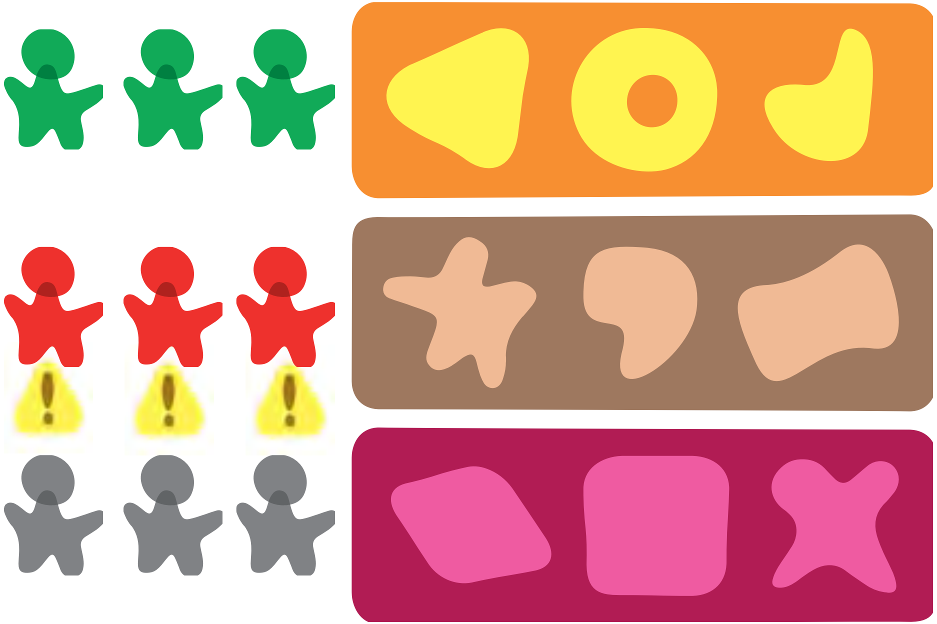
Compromised quality to satisfy the user

CHALLENGES OF THE TIERED ARCHITECTURE



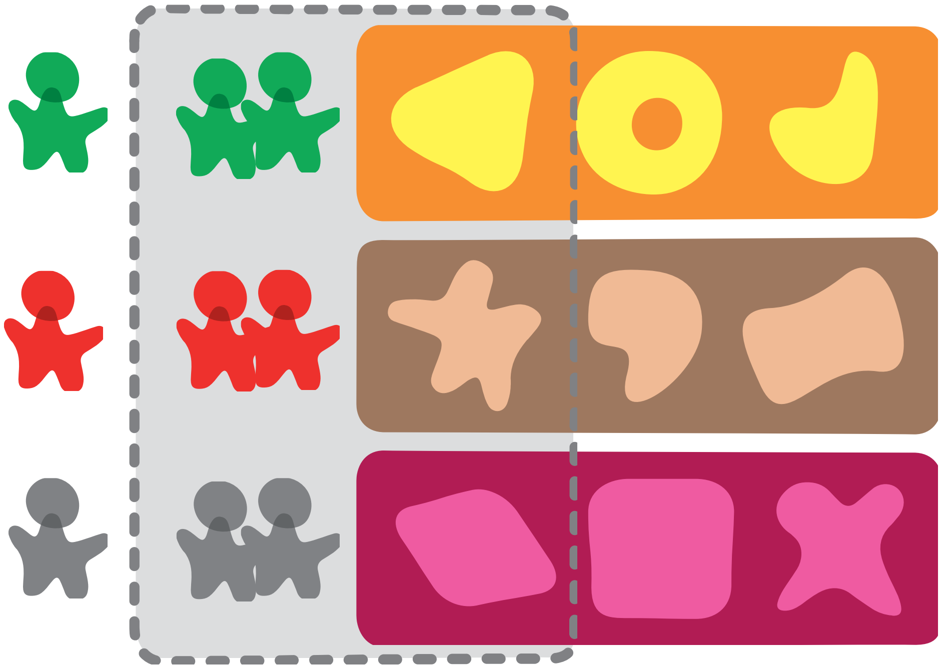
Compromised quality to satisfy the user

CHALLENGES OF THE TIERED ARCHITECTURE



Compromised quality to satisfy the user

CHALLENGES OF THE TIERED ARCHITECTURE



Compromised quality to satisfy the user

Longer release cycle

FINE-GRAINED ARCHITECTURE

FINE-GRAINED ARCHITECTURE

Broken down based on vertical business functions

FINE-GRAINED ARCHITECTURE

Technology layers
are encapsulated
Inside each service

MICROSERVICES

MICRO

SERVICES

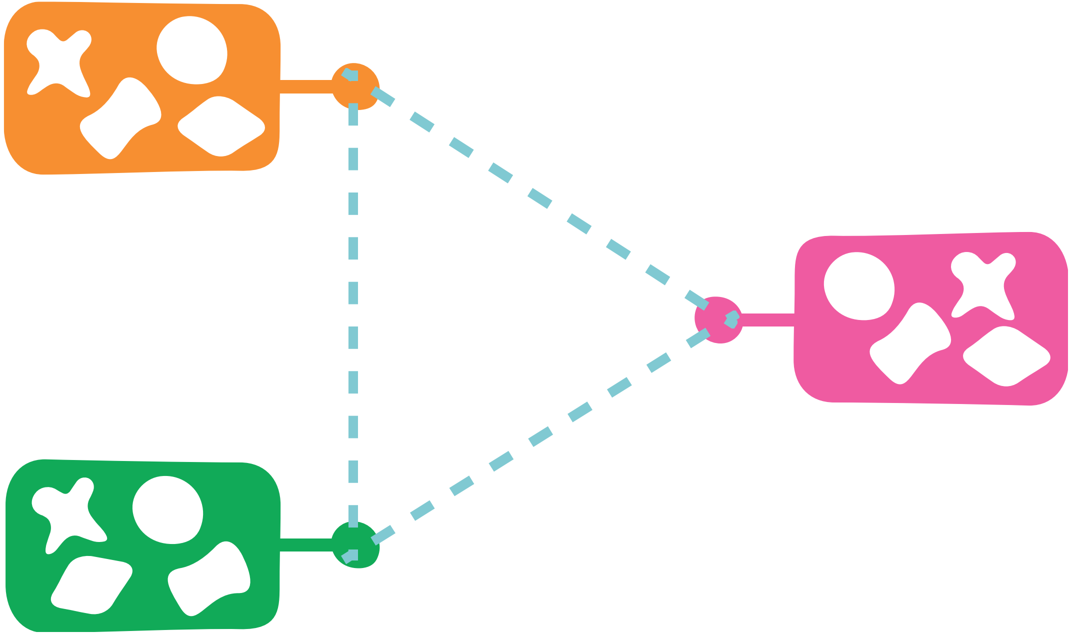
- *Independently releasable*
- *Business capability*
- *Language agnostic API*

MICRO

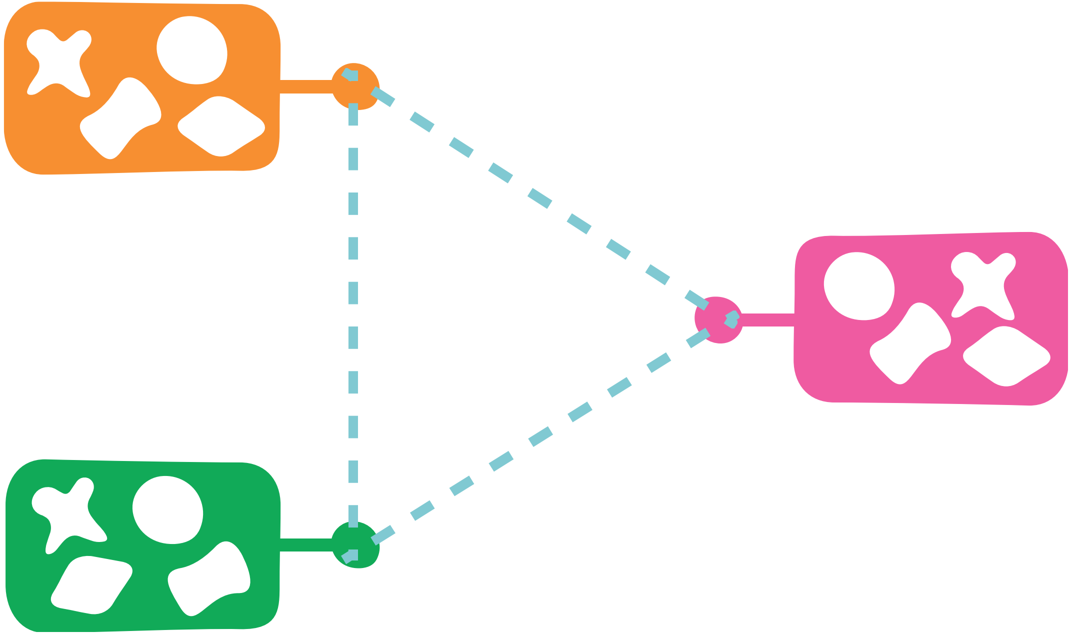
- *Small*
- *100s, not 1000s*

SERVICES

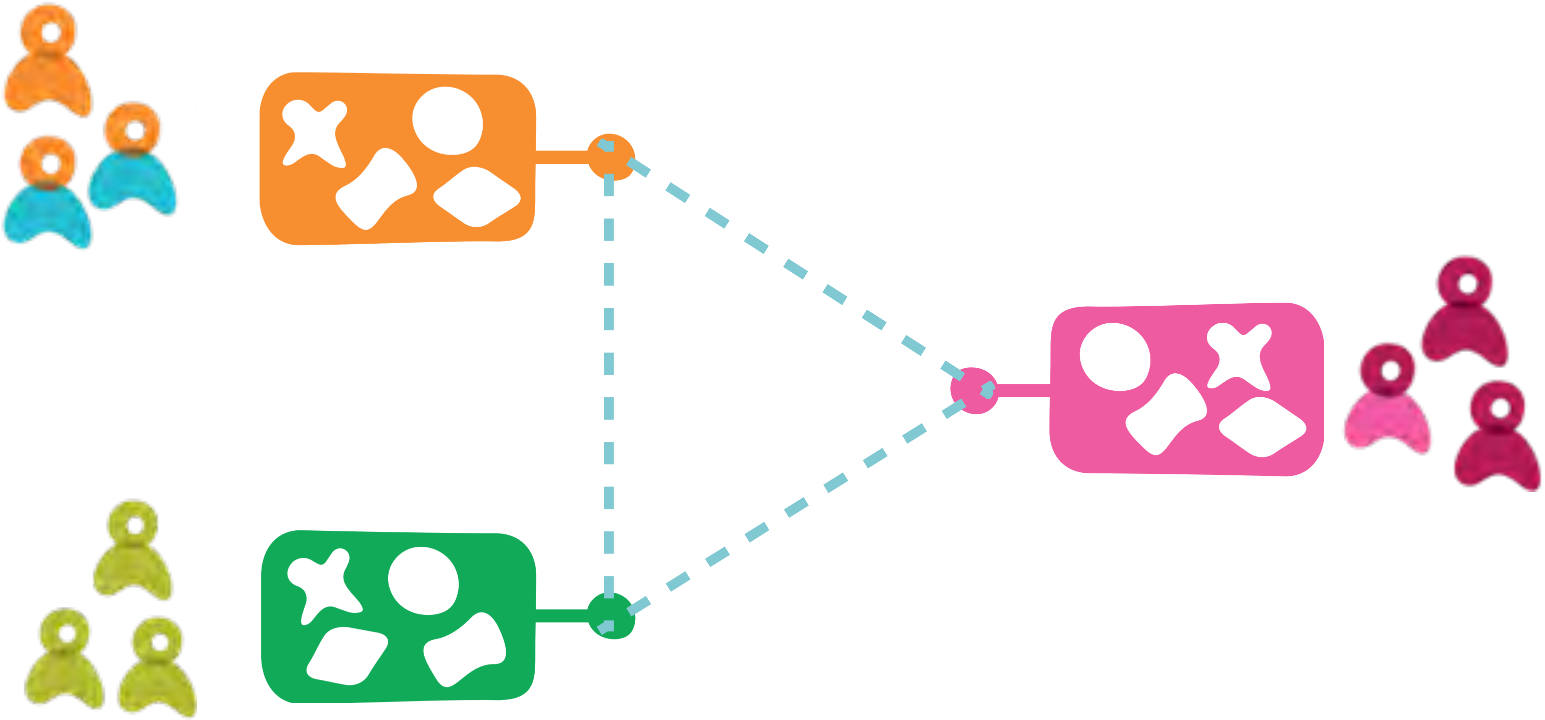
- *Independently releasable*
- *Business capability*
- *Language agnostic API*



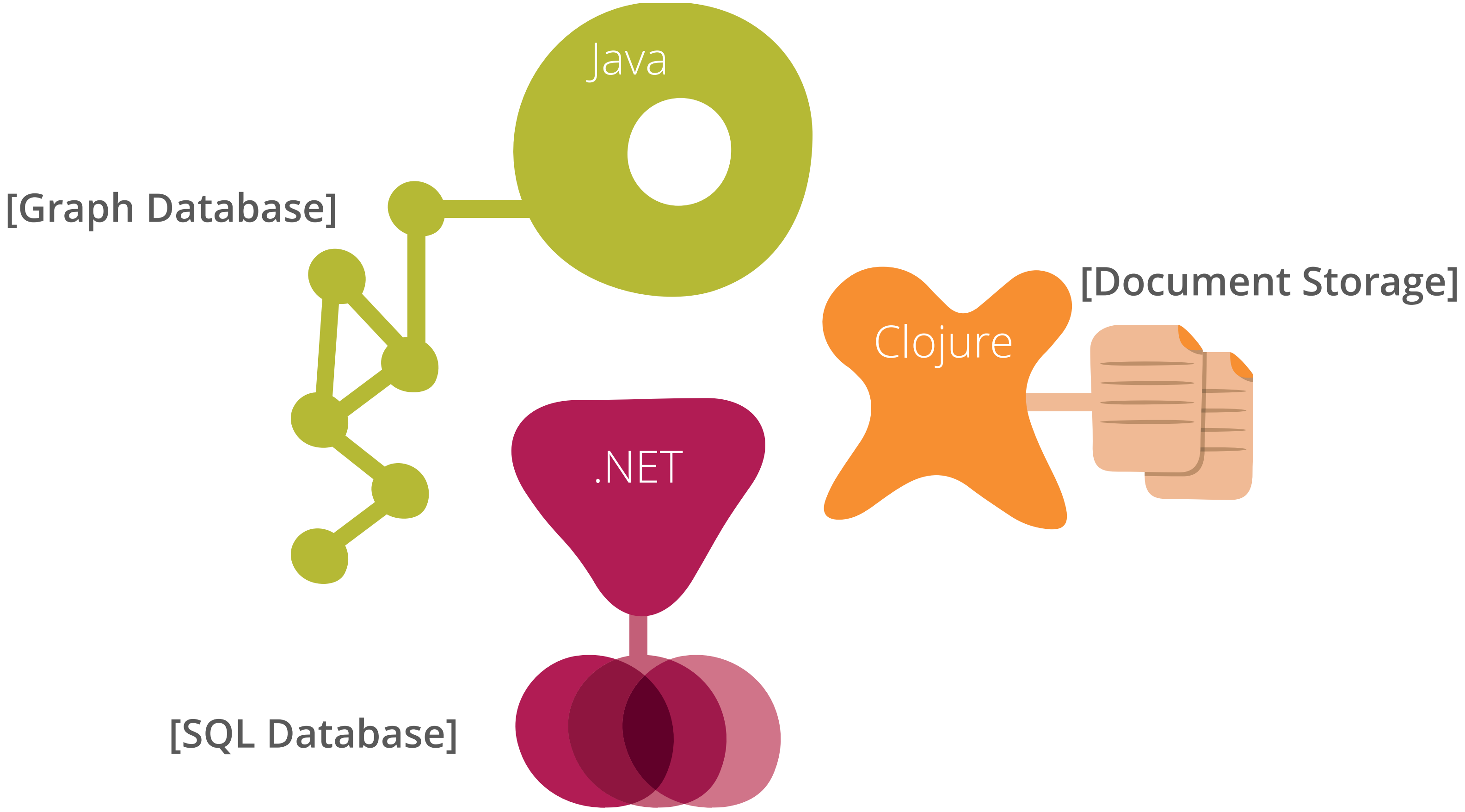
AUTONOMY



AUTONOMY



ABSORBS TECHNOLOGY FASTER



Java

[Graph Database]

Clojure

[Document Storage]

.NET

[SQL Database]

COMPOSE-ABILITY

COMPOSE-ABILITY

COMPOSE-ABILITY

COMPOSE-ABILITY

COMPOSE-ABILITY

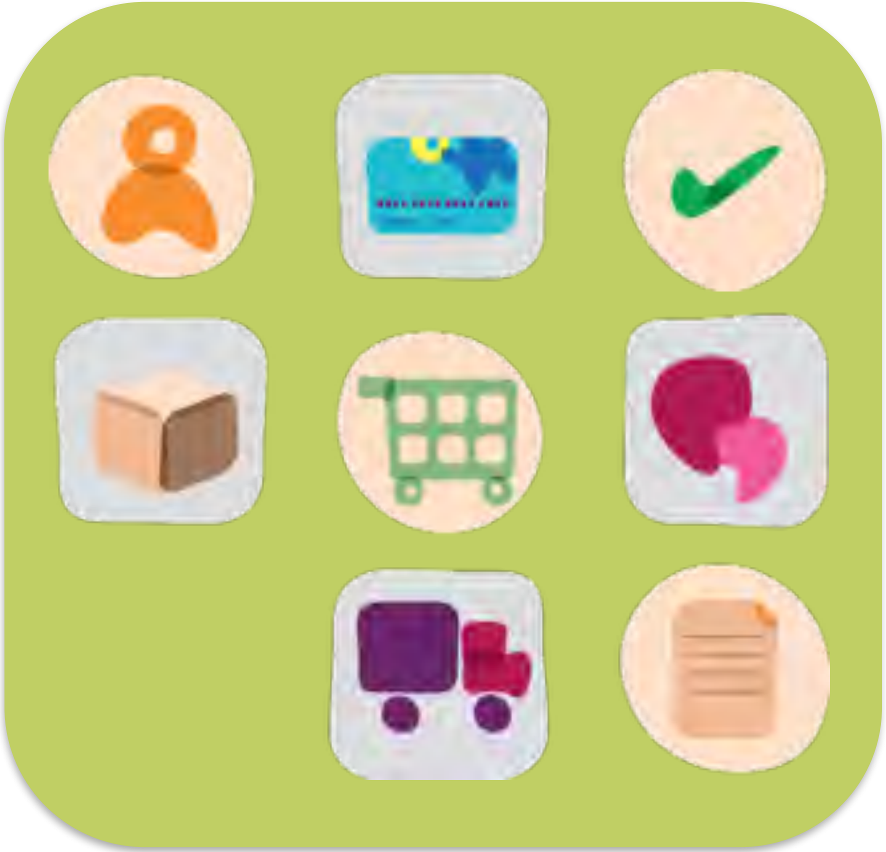
REPLACABILITY

HOW TO **IMPLEMENT** MICROSERVICES

BOUNDARIES

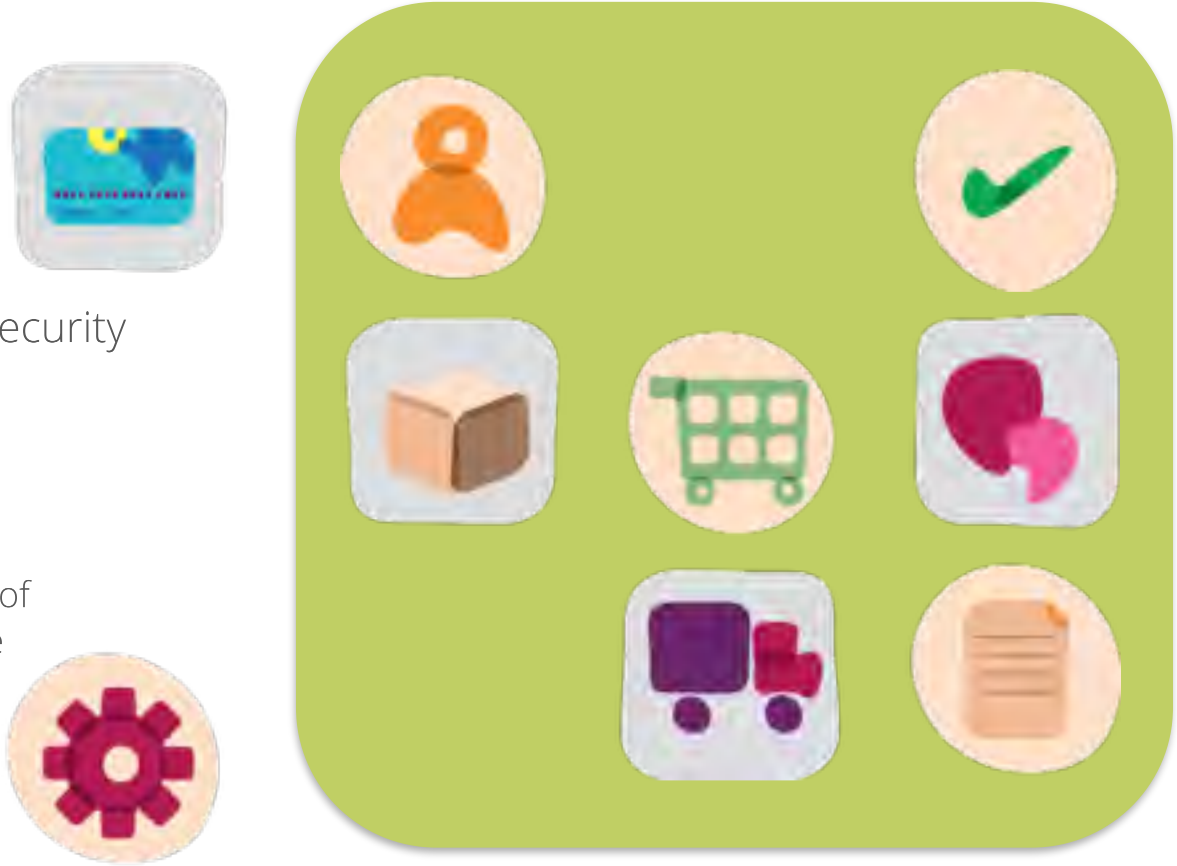
SPLITTING

SPLITTING



Speed of change

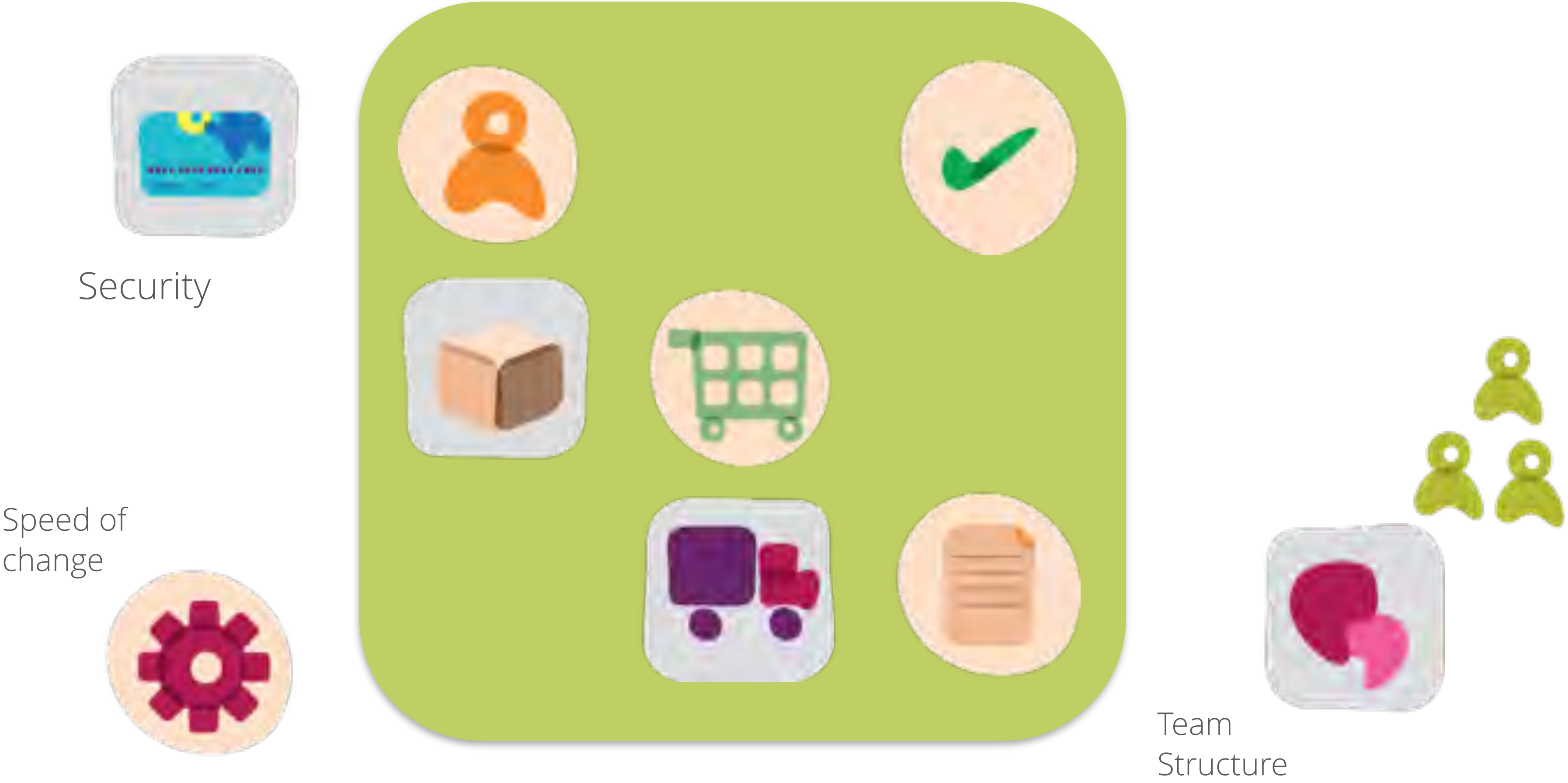
SPLITTING



Security

Speed of change

SPLITTING



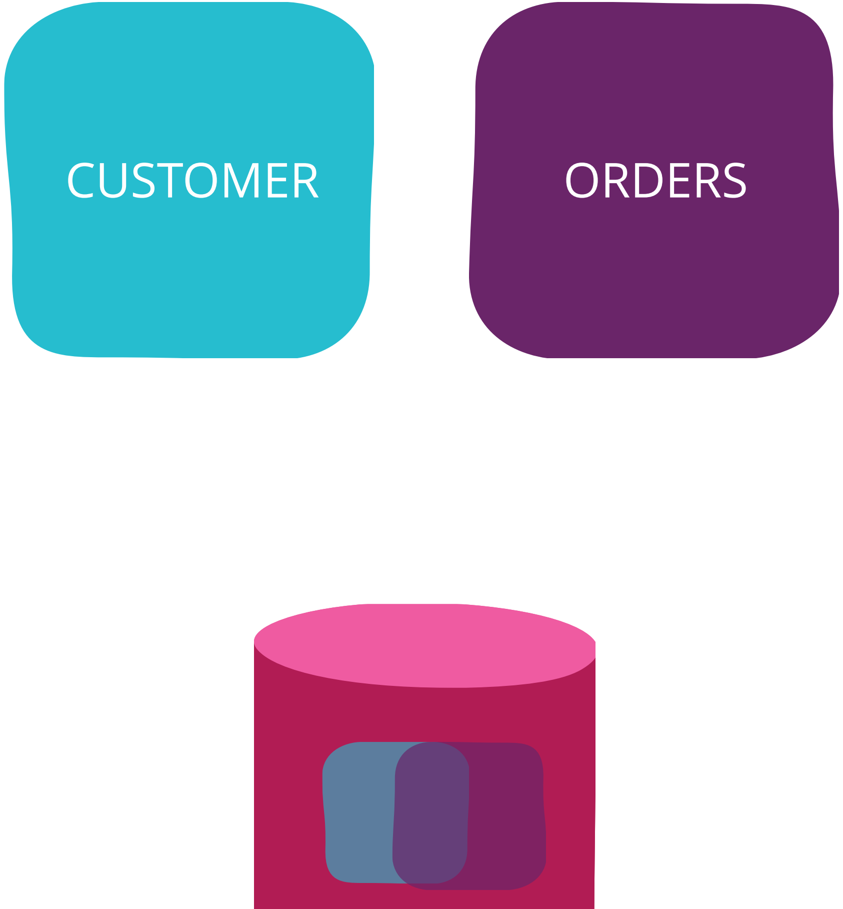
Security

Speed of change

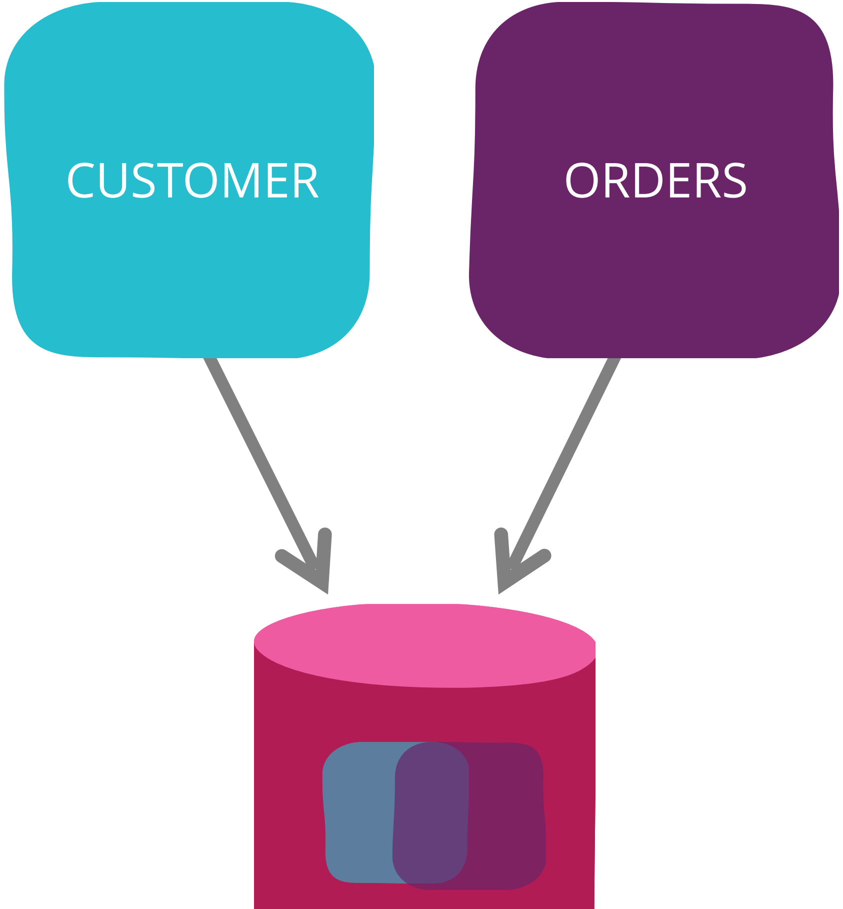
Team Structure

CAN I **RELEASE** THEM **SEPARATELY**?

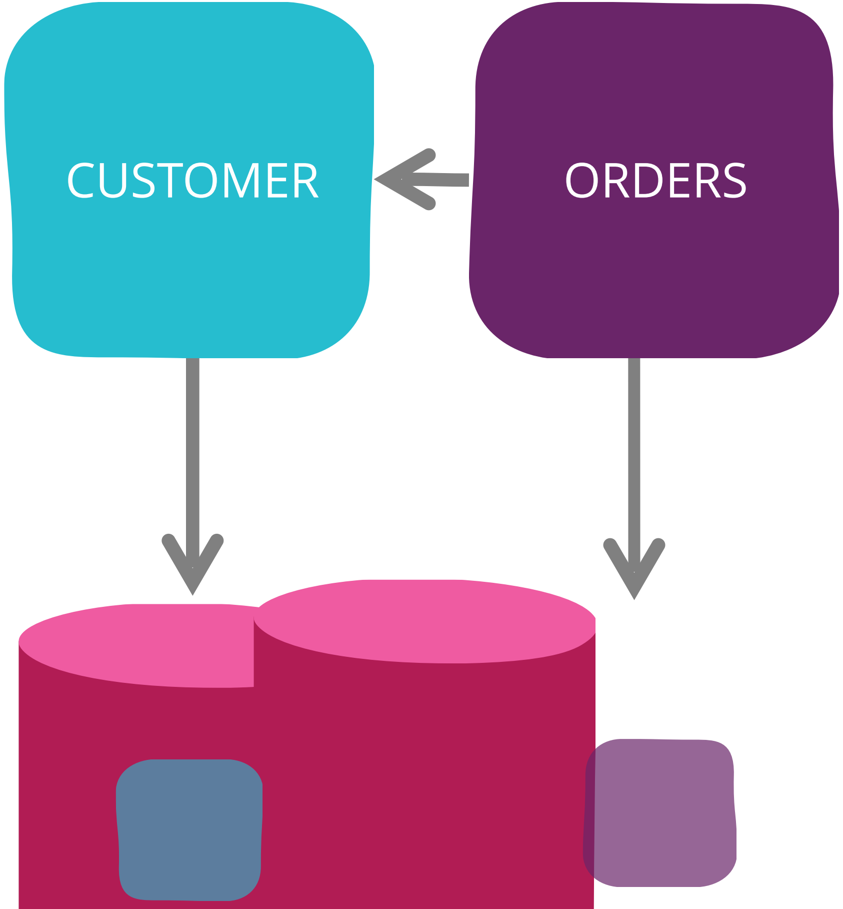
COUPLING THROUGH DATABASE



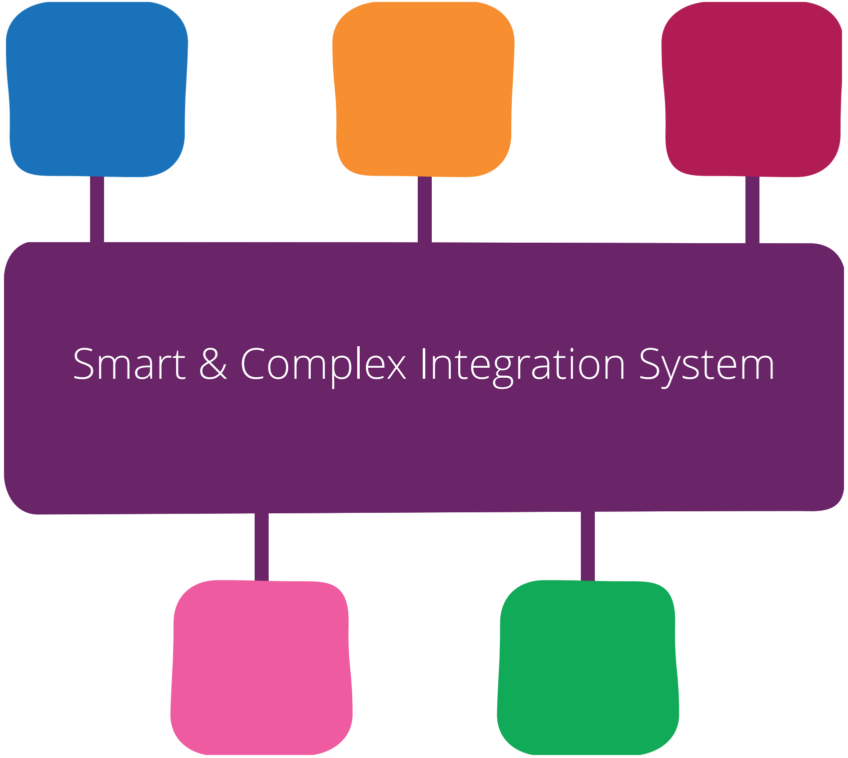
COUPLING THROUGH DATABASE



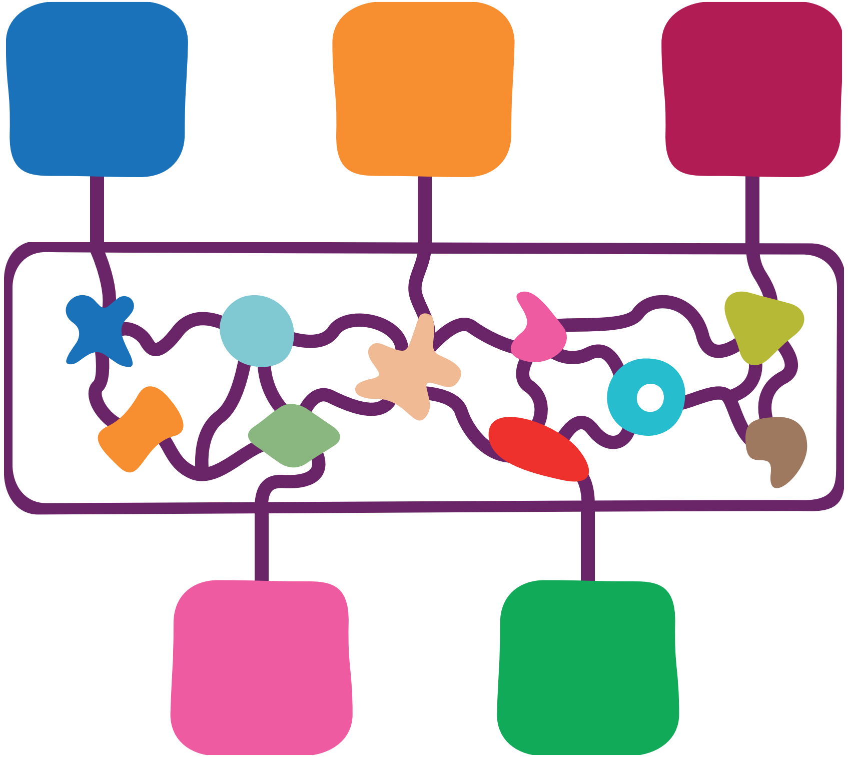
COUPLING THROUGH DATABASE



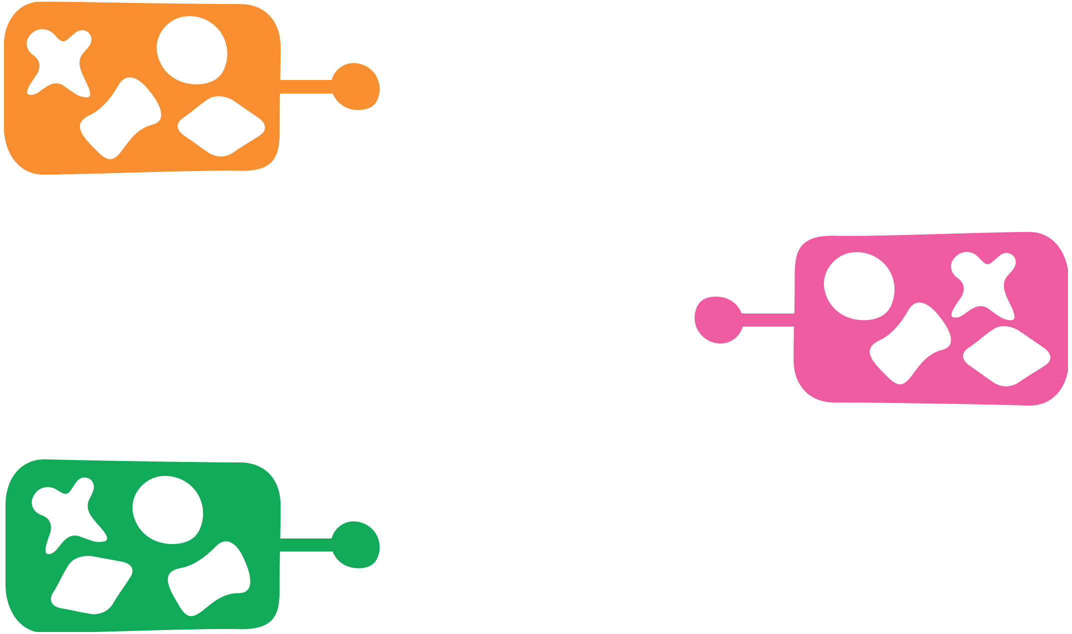
COUPLING THROUGH ENTERPRISE SERVICE BUS



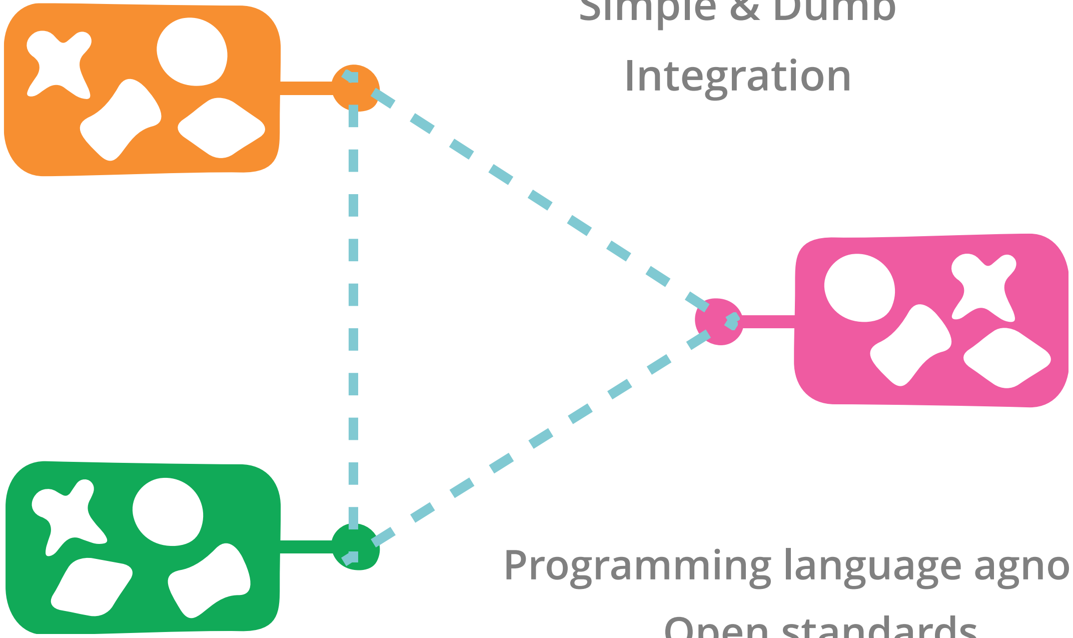
COUPLING THROUGH ENTERPRISE SERVICE BUS



MICROSERVICES INTEGRATION



MICROSERVICES INTEGRATION



Smart Service &

Endpoint

Simple & Dumb

Integration

Programming language agnostic

Open standards

HTTP/SON

WHAT ABOUT **SOA?**

OR **OSGI?**

HOW TO **OPERATE** MICROSERVICES

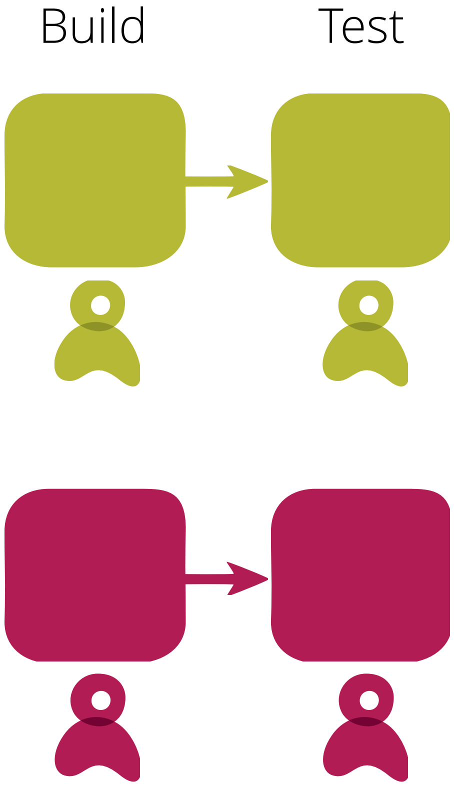
AUTOMATE!

AUTOMATE!

AUTOMATE!

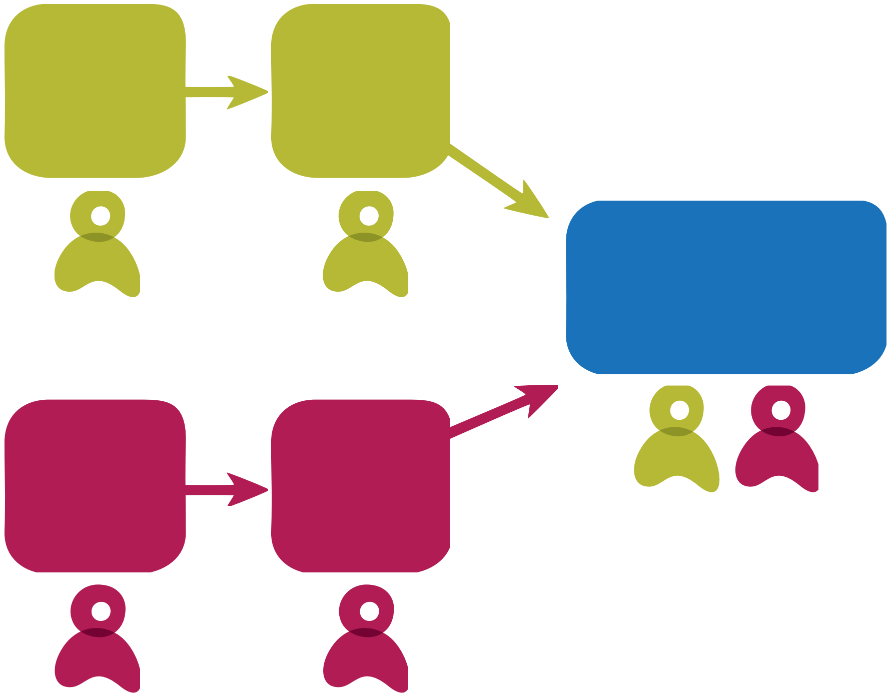
AUTOMATION

AUTOMATION

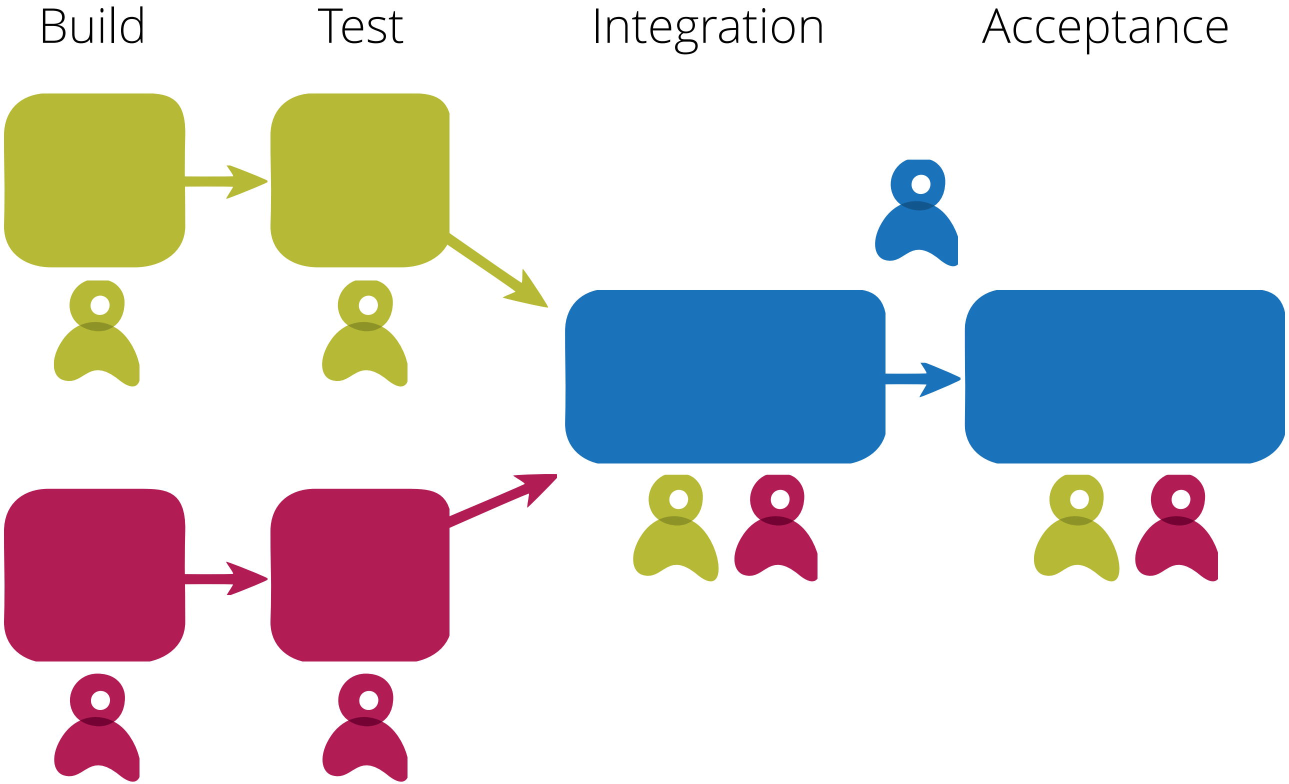


AUTOMATION

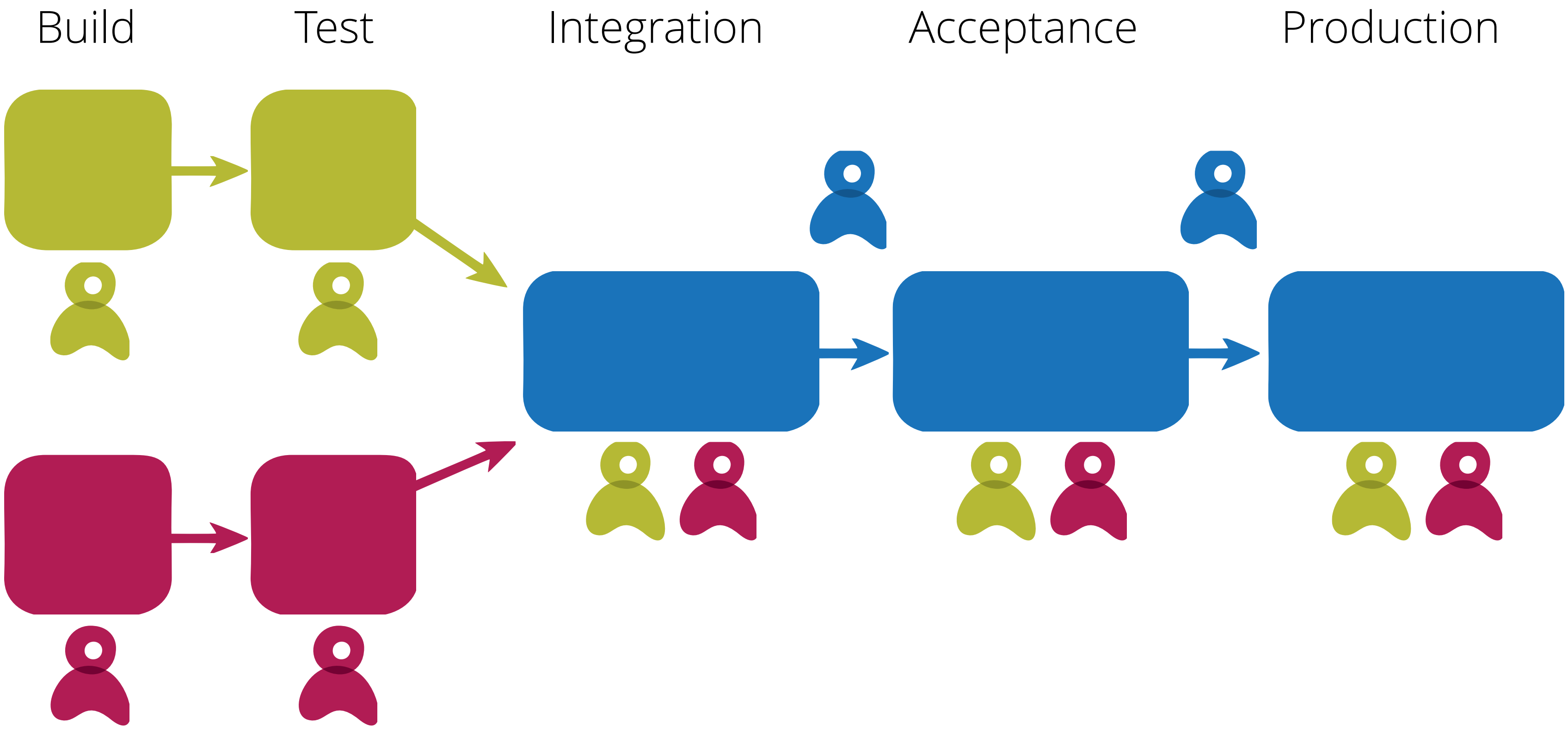
Build Test Integration



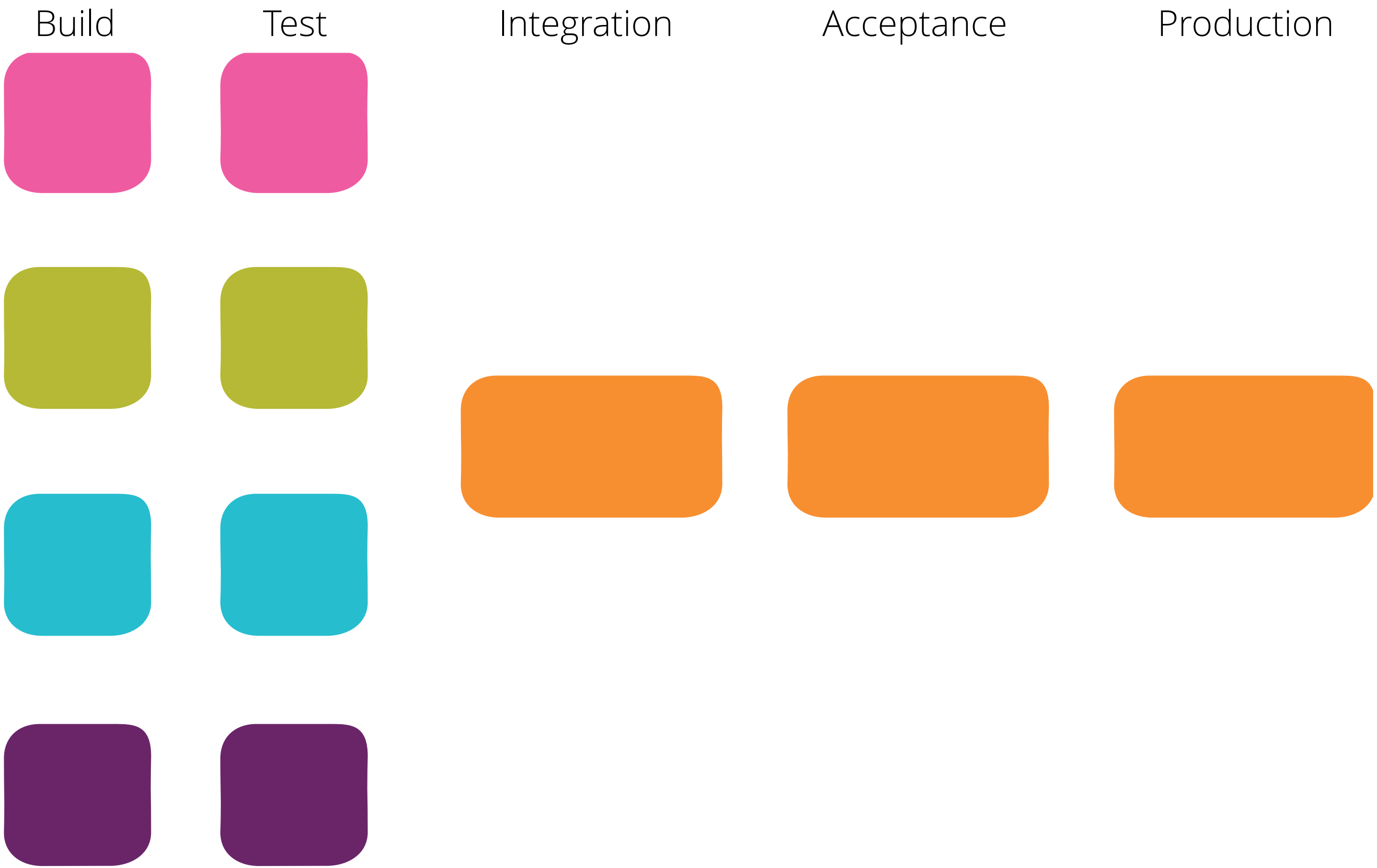
AUTOMATION



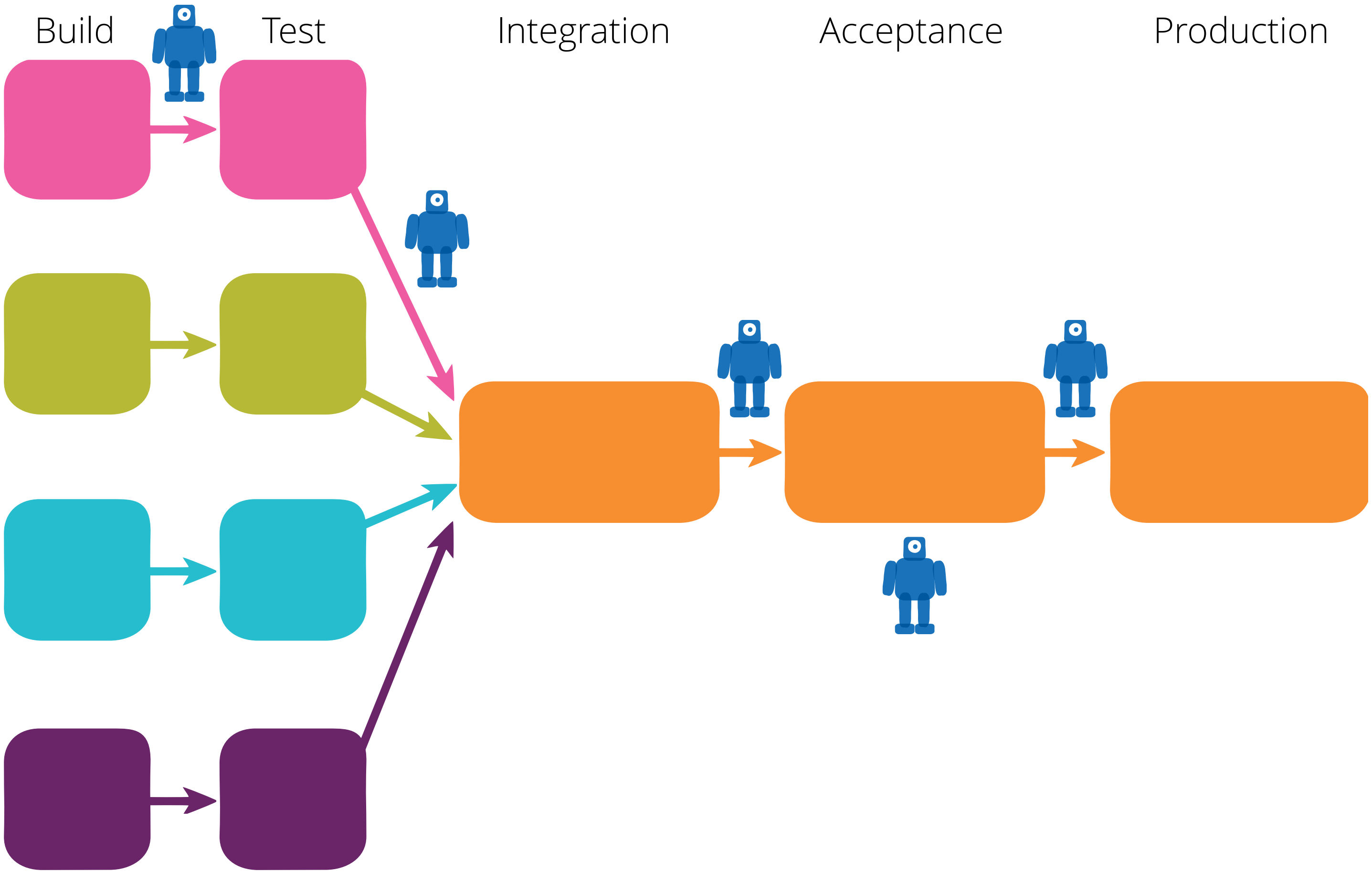
AUTOMATION



MICROSERVICES AUTOMATION



MICROSERVICES AUTOMATION



CONSUMER-DRIVEN CONTRACTS

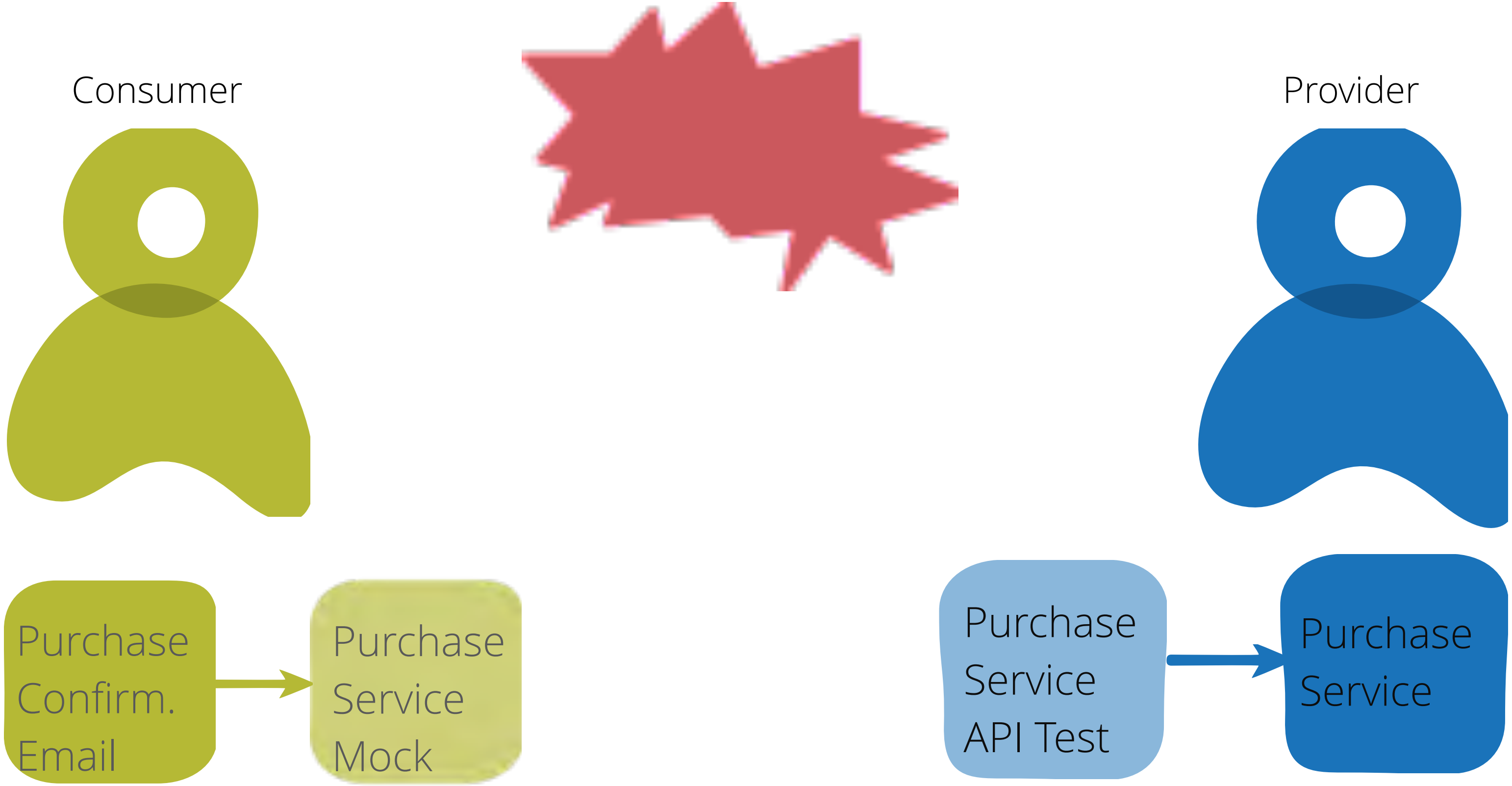
Consumer

Provider

Purchase
Confirm.
Email

Purchase
Service

CONSUMER-DRIVEN CONTRACTS



Consumer

Provider

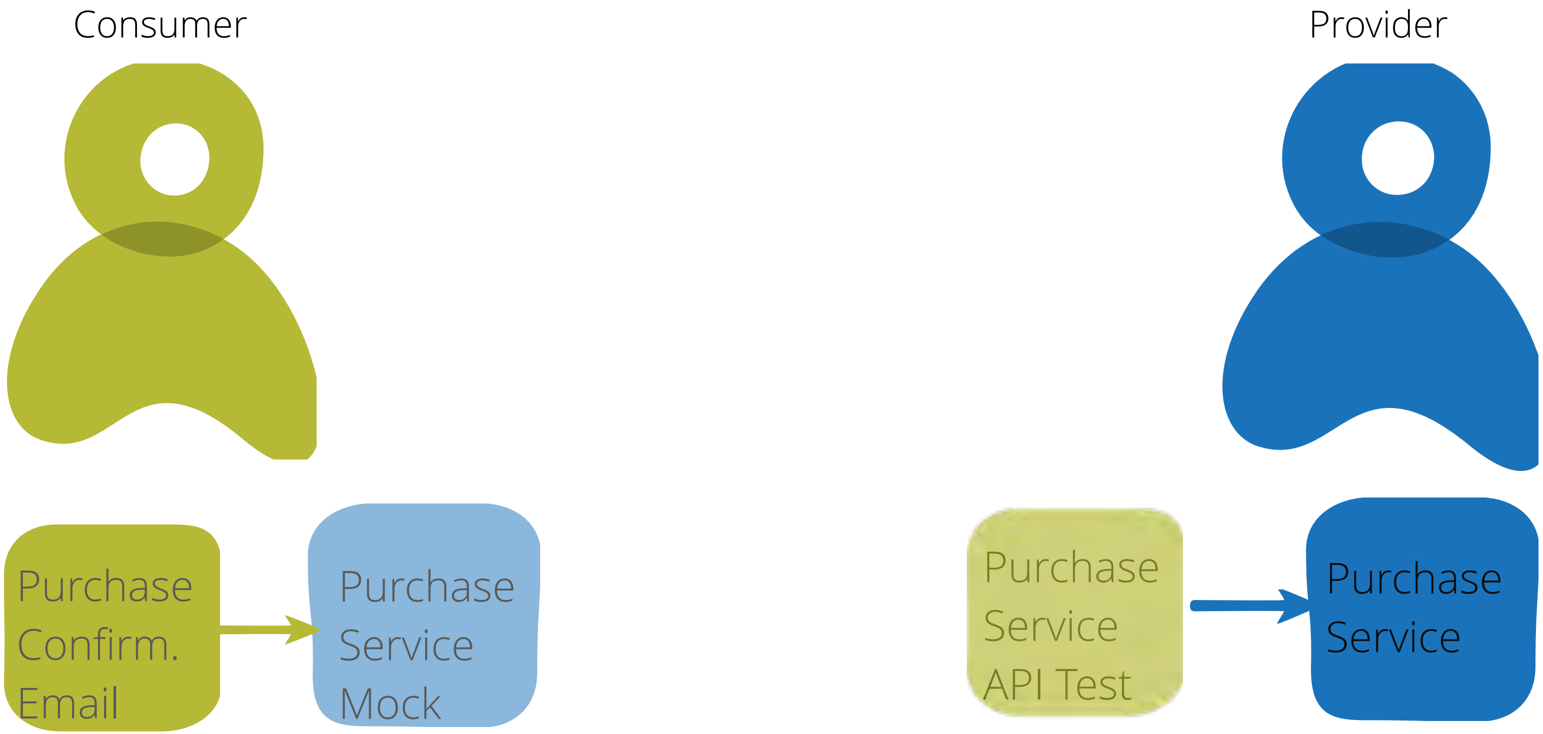
Purchase Confirm. Email

Purchase Service Mock

Purchase Service API Test

Purchase Service

CONSUMER-DRIVEN CONTRACTS



Consumer

Provider

Purchase Confirm. Email

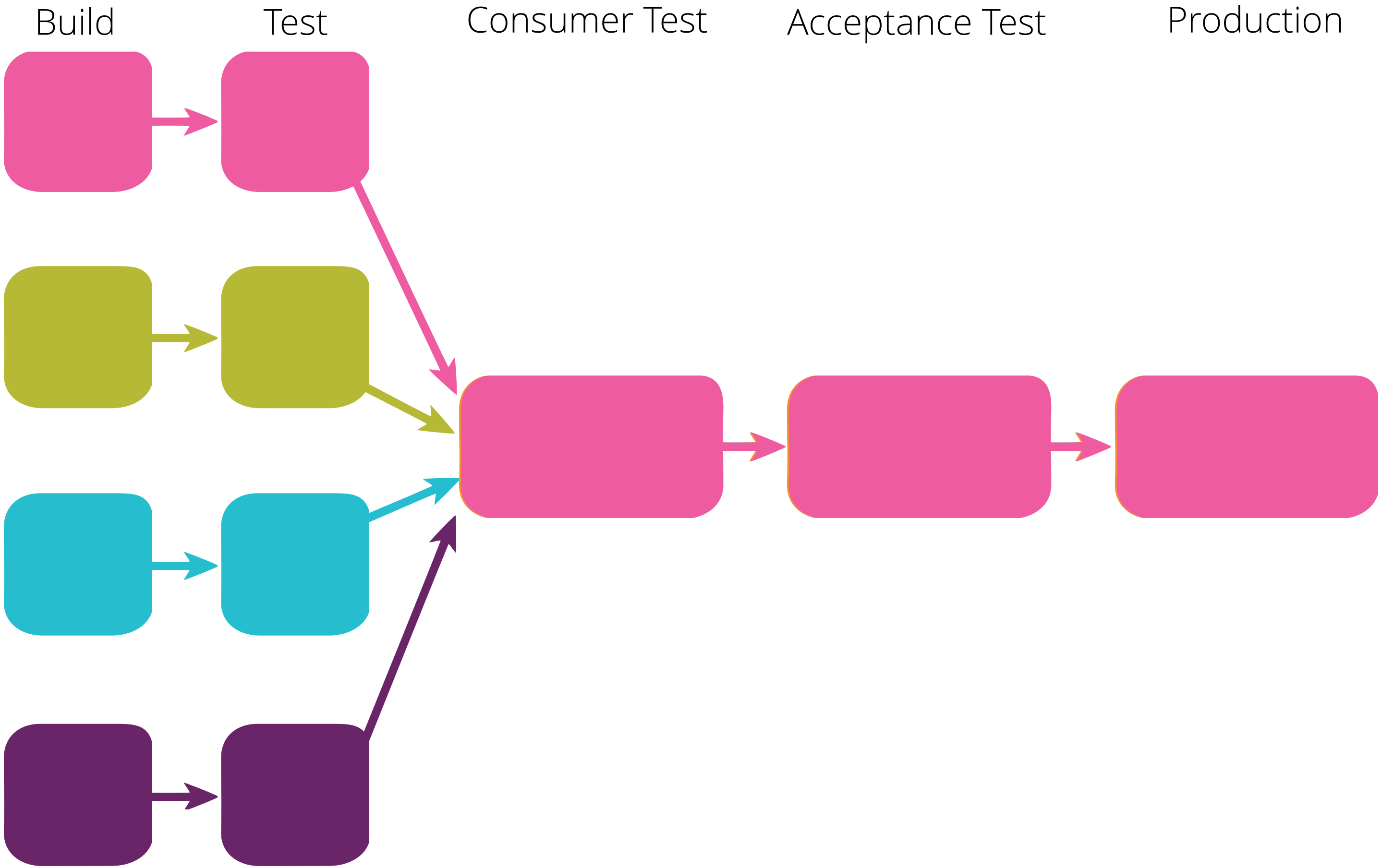
Purchase Service Mock

Purchase Service API Test

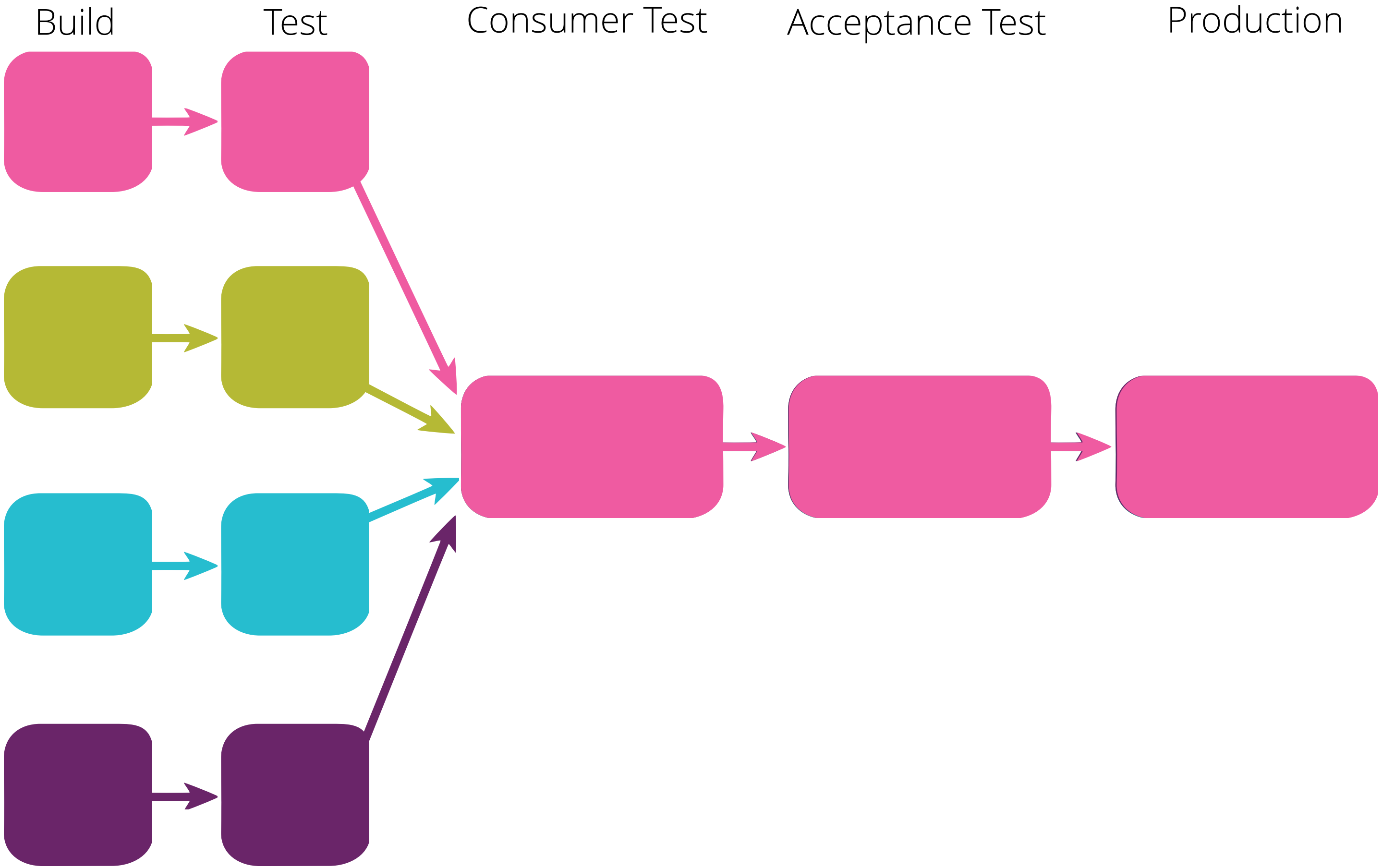
Purchase Service

Consumer contract tests

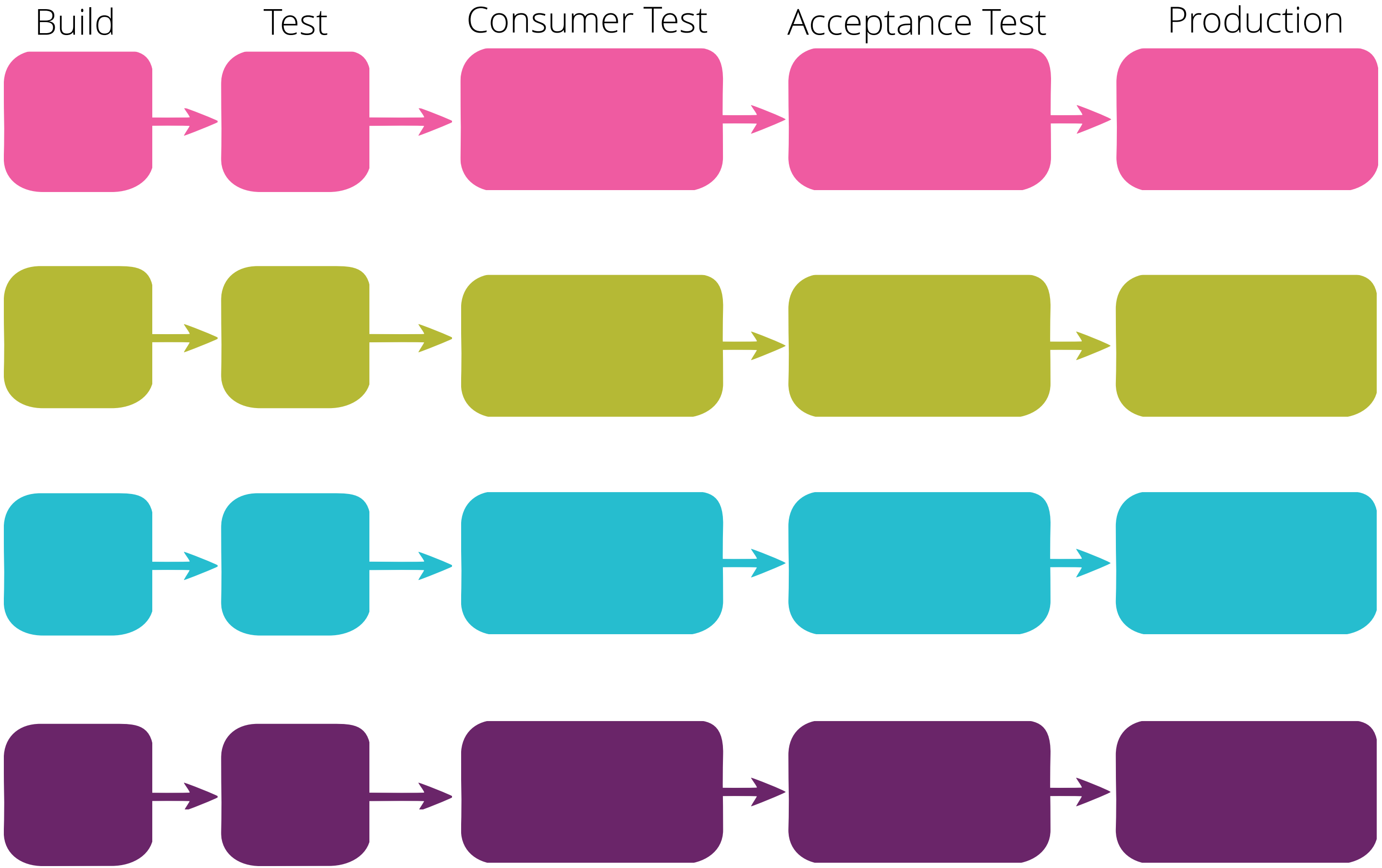
MICROSERVICES’ PATH TO PRODUCTION



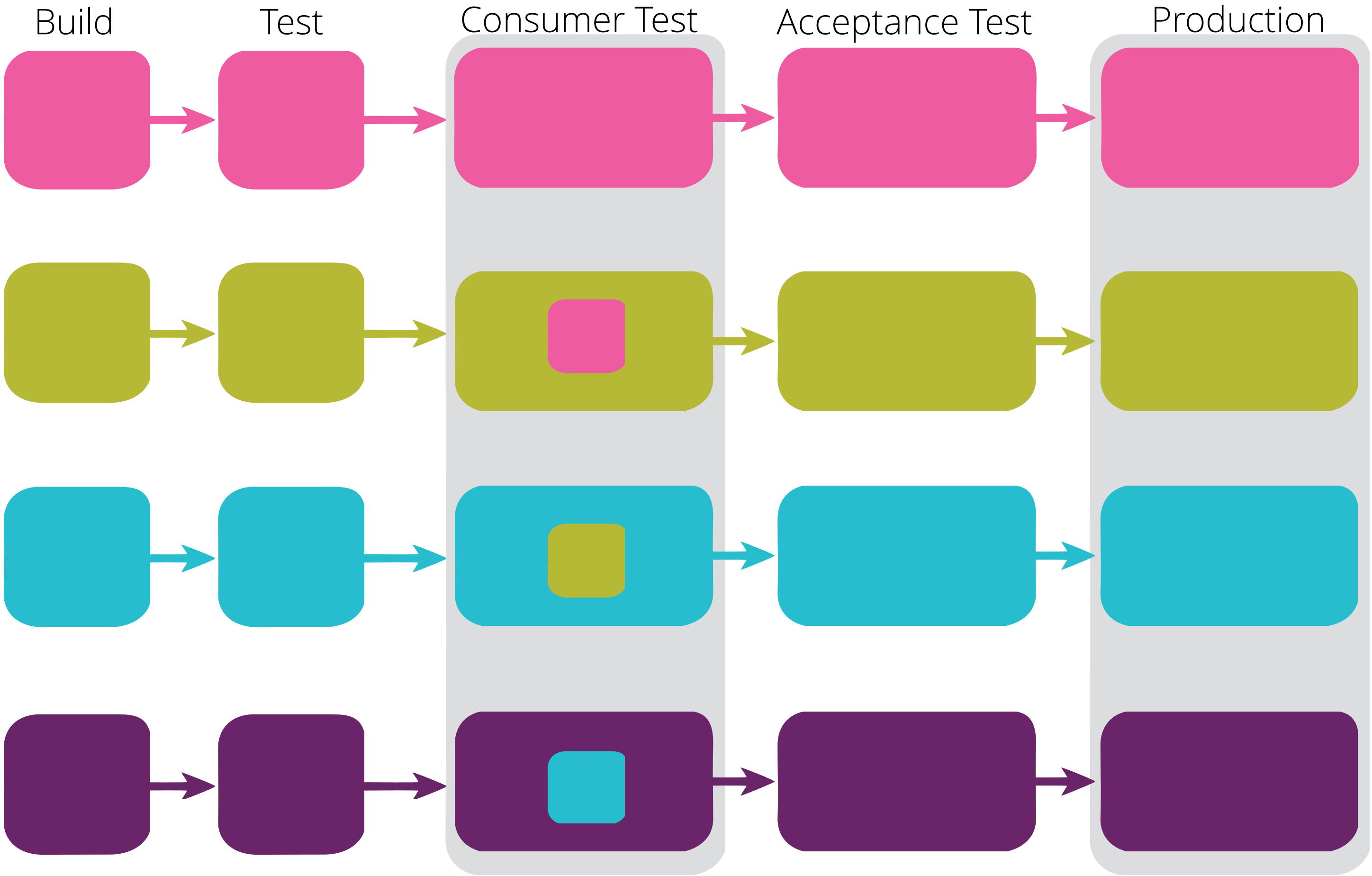
MICROSERVICES’ PATH TO PRODUCTION



MICROSERVICES' PATH TO PRODUCTION



MICROSERVICES’ PATH TO PRODUCTION



MONITORING

MONITORING

STATUS

- Up?

MONITORING

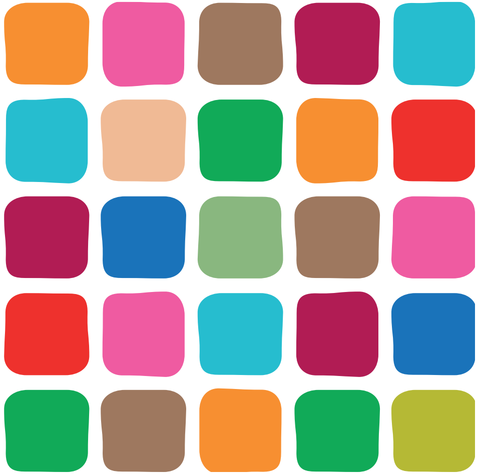
STATUS

- Up?
- Down?

MONITORING

STATUS

- Up?
- Down?



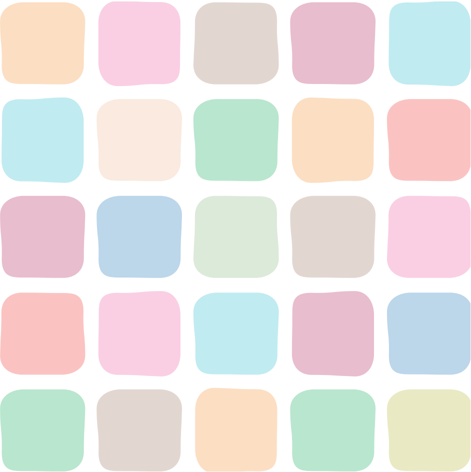
STATUS

- All up?

MONITORING

STATUS

- Up?
- Down?



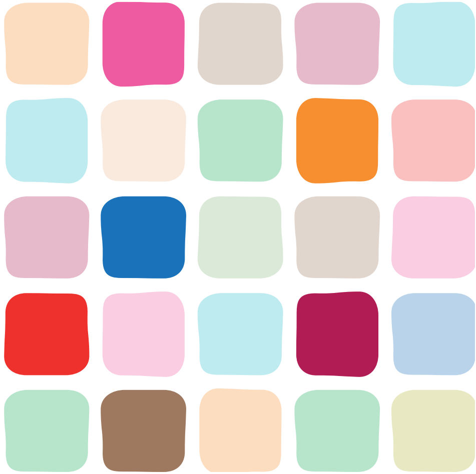
STATUS

- All up?
- All down?

MONITORING

STATUS

- Up?
- Down?



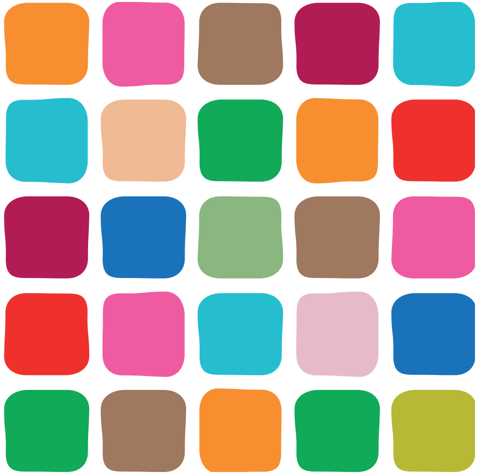
STATUS

- All up?
- All down?
- Mostly up?

MONITORING

STATUS

- Up?
- Down?



STATUS

- All up?
- All down?
- Mostly up?
- Up apart from 1?

MONITORING

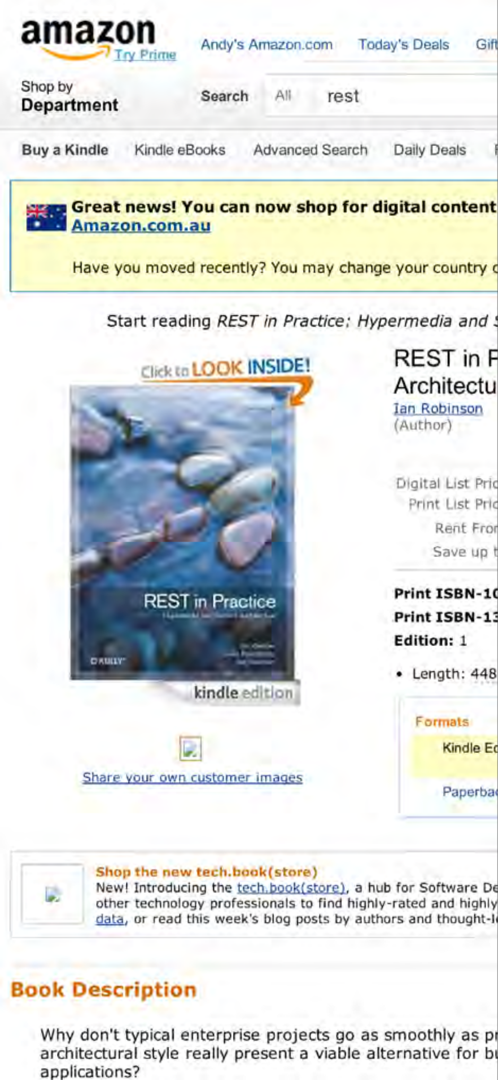
STATUS

- Up?
- Down?

STATUS

- All up?
- All down?
- Mostly up?
- Up apart from 1?
- All of the important bits are up?
- …

AMAZON PRODUCT PAGE



Great news! You can now shop for digital content
Amazon.com.au

Have you moved recently? You may change your country

Start reading *REST in Practice: Hypermedia and S*

REST in P
Architectu
Ian Robinson
(Author)

Digital List Pric
Print List Pric
Rent From
Save up t

Print ISBN-10
Print ISBN-13
Edition: 1

- Length: 448

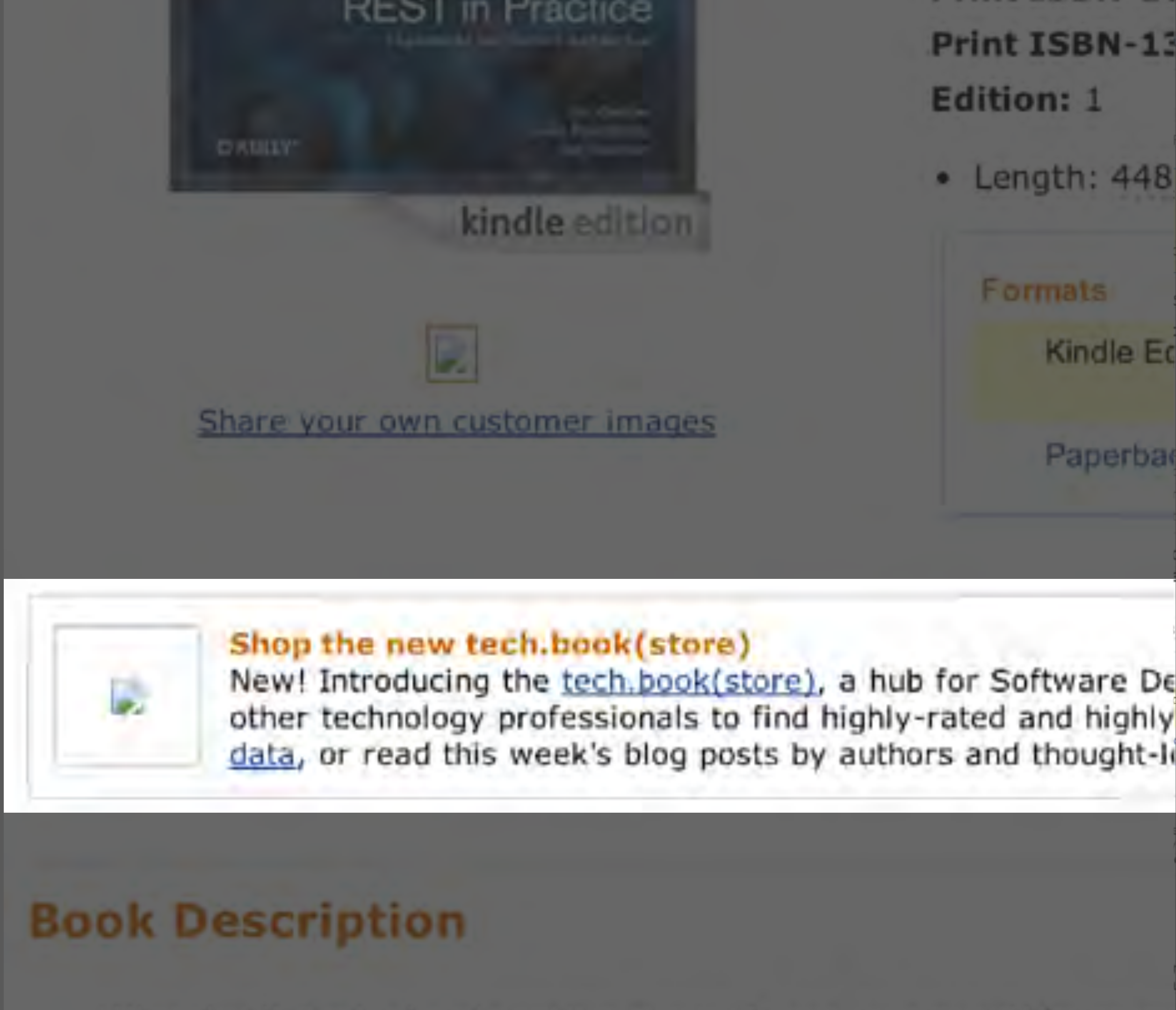
Share your own customer images

Shop the new tech.book(store)
New! Introducing the tech.book(store), a hub for Software De
other technology professionals to find highly-rated and highly
data, or read this week's blog posts by authors and thought-le

Book Description

Why don't typical enterprise projects go as smoothly as pr
architectural style really present a viable alternative for bu
applications?

AMAZON PRODUCT PAGE



REST in Practice

O'REILLY

kindle edition

Print ISBN-13

Edition: 1

- Length: 448

Formats

Kindle Ed

Paperba

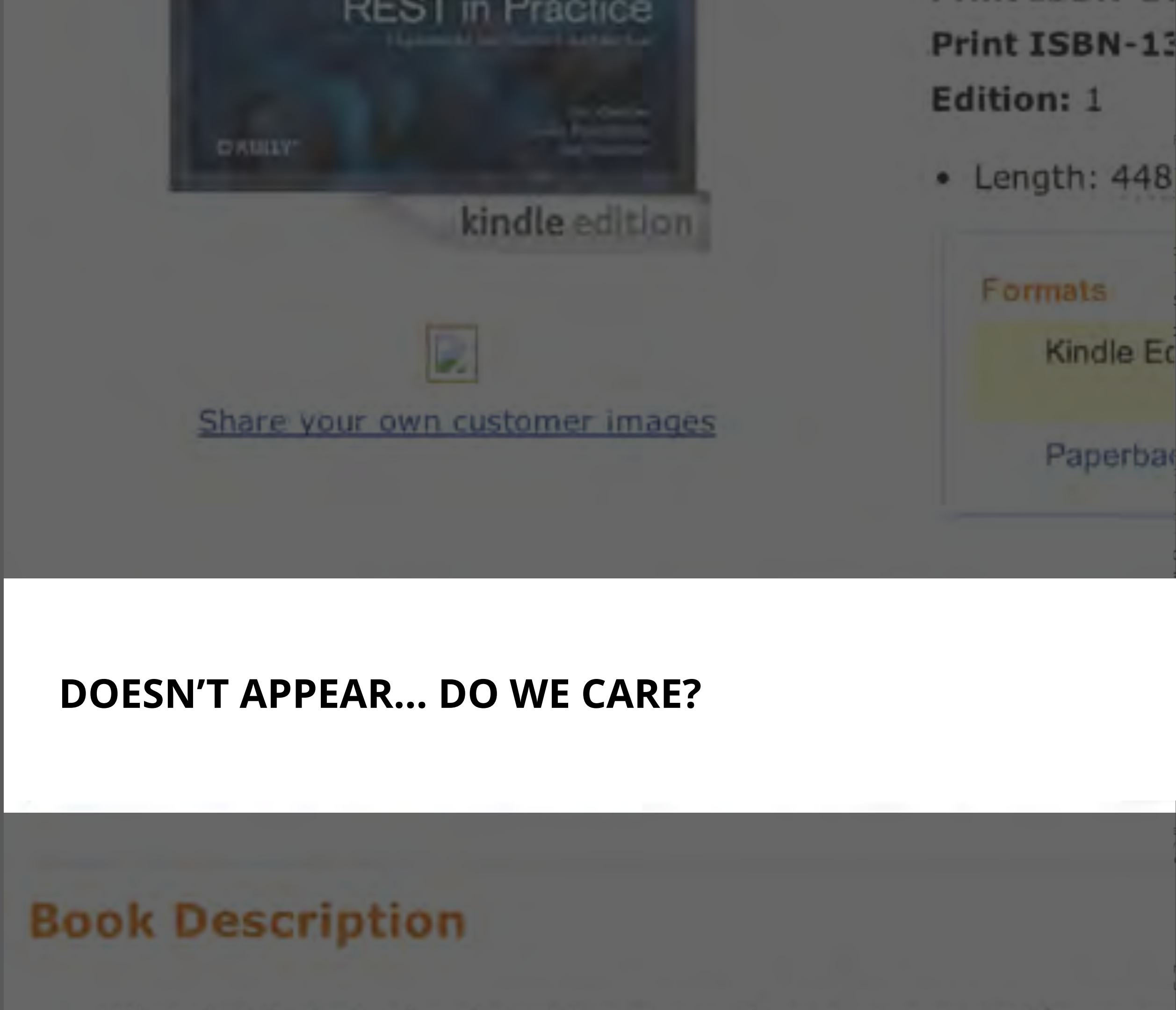
Share your own customer images

Shop the new tech.book(store)

New! Introducing the tech.book(store), a hub for Software De
other technology professionals to find highly-rated and highly
data, or read this week's blog posts by authors and thought-l

Book Description

AMAZON PRODUCT PAGE



REST in Practice

kindle edition

Print ISBN-13

Edition: 1

- Length: 448

Share your own customer images

Formats

Kindle Ed

Paperba

Book Description

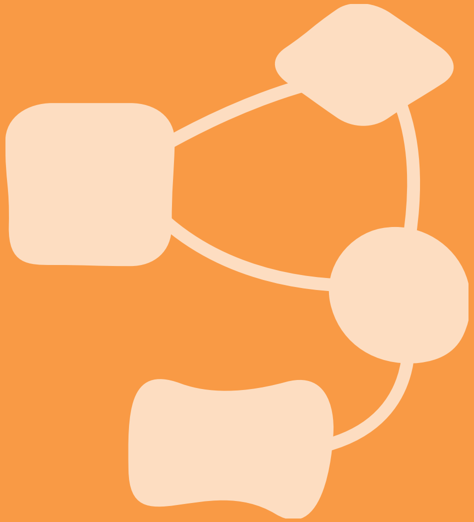
DOESN’T APPEAR… DO WE CARE?

SUMMARY

RECOMMENDATIONS

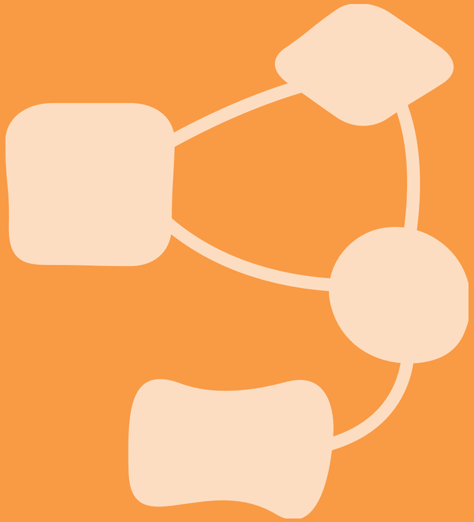
RECOMMENDATIONS

- *Build independently releasable microservices*
  - *align microservice with business capability*



RECOMMENDATIONS

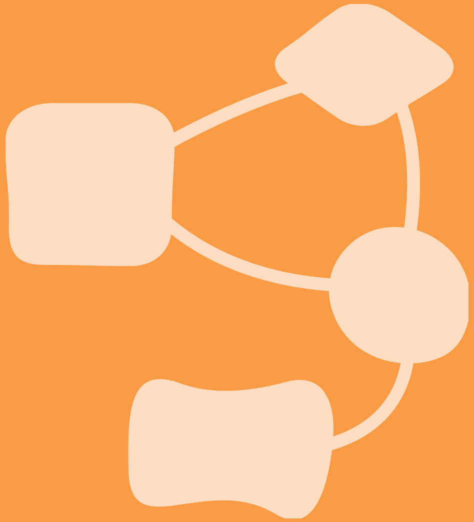
- *Build independently releasable microservices*
  - *align microservice with business capability*
- *Integrate microservices with*
  - *dumb, open language agnostic*
  - *outside database*



RECOMMENDATIONS

- *Build independently releasable microservices*
  - *align microservice with business capability*
- *Integrate microservices with*
  - *dumb, open language agnostic*
  - *outside database*
- *Operate microservices architecture by*
  - *decentralizing and automating all aspects of build, test and deployment*

RECOMMENDATIONS



- *Build independently releasable microservices*
  - *align microservice with business capability*
- *Integrate microservices with*
  - *dumb, open language agnostic*
  - *outside database*
- *Operate microservices architecture by*
  - *decentralizing and automating all aspects of build, test and deployment*

CONCLUSION

ThoughtWorks®

O'REILLY

Building Microservices

DESIGNING FINE-GRAINED SYS

Sam Newman

THANK YOU

Zhamak Dehghani

zdehghan@thoughtworks.com @zhamakd

Sam Newman

snewman@thoughtworks.com @samnewman