Code PaLOUsa Edition

MICROSERVICES

Lessons from the Trenches

@gbworld

{Code PaLOUsa}

Our Sponsors

KiZAN

CBS Interactive

Bastian SOLUTIONS

elastic

HMB

COMPOSABLE SYSTEMS

ups

<code> LOUISVILLE

SOFTWARE GUILD

CLICK
The relationship-driven IT firm

TEKsystems

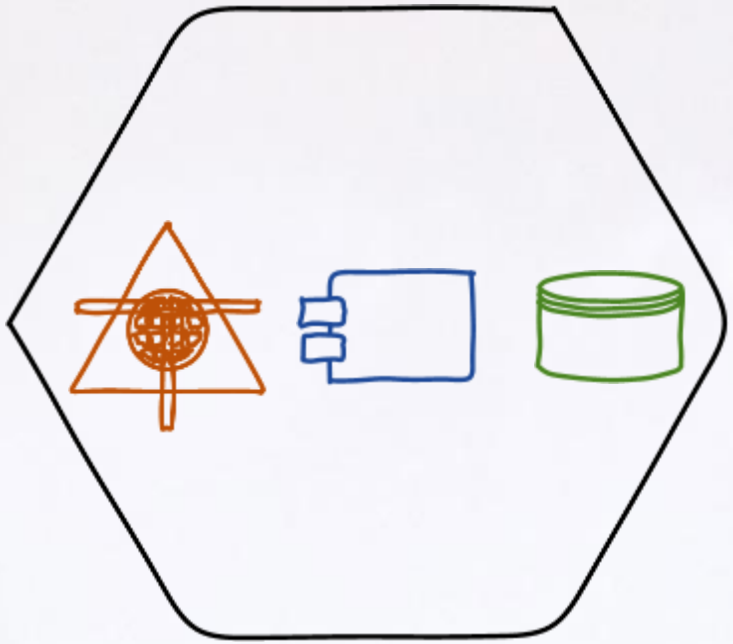
MS Manifest Solutions

MODIS

SAIC
Redefining Ingenuity™

Agenda

- What are Microservices?
  A "Purist" Introduction
- Focus on Architecture
  A Pragmatic Look
- Microservices & the Real World
  Lessons Learned in the Trenches



{Code PaLOUsa}

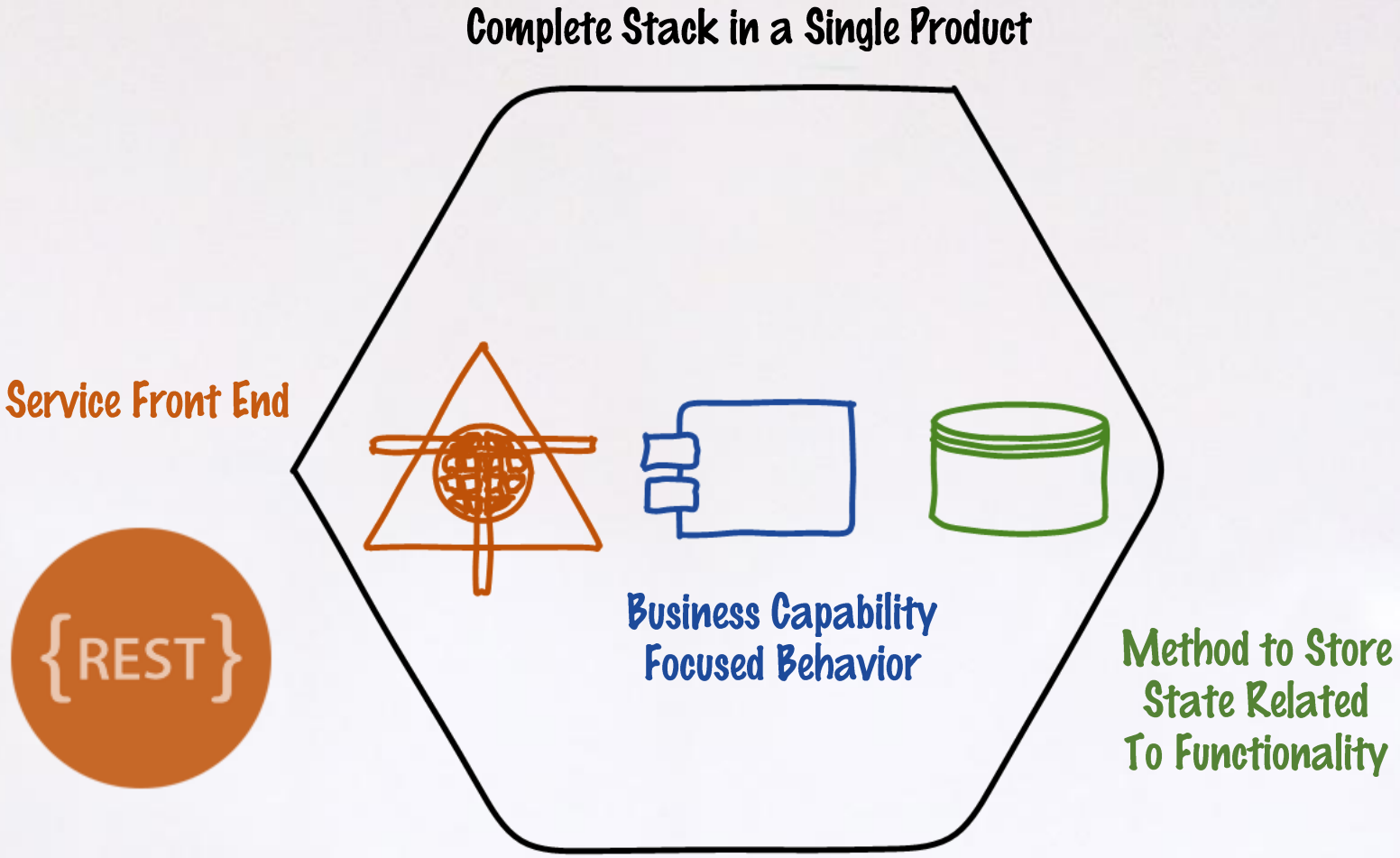
What are Microservices?

A "Purist" Introduction

What?

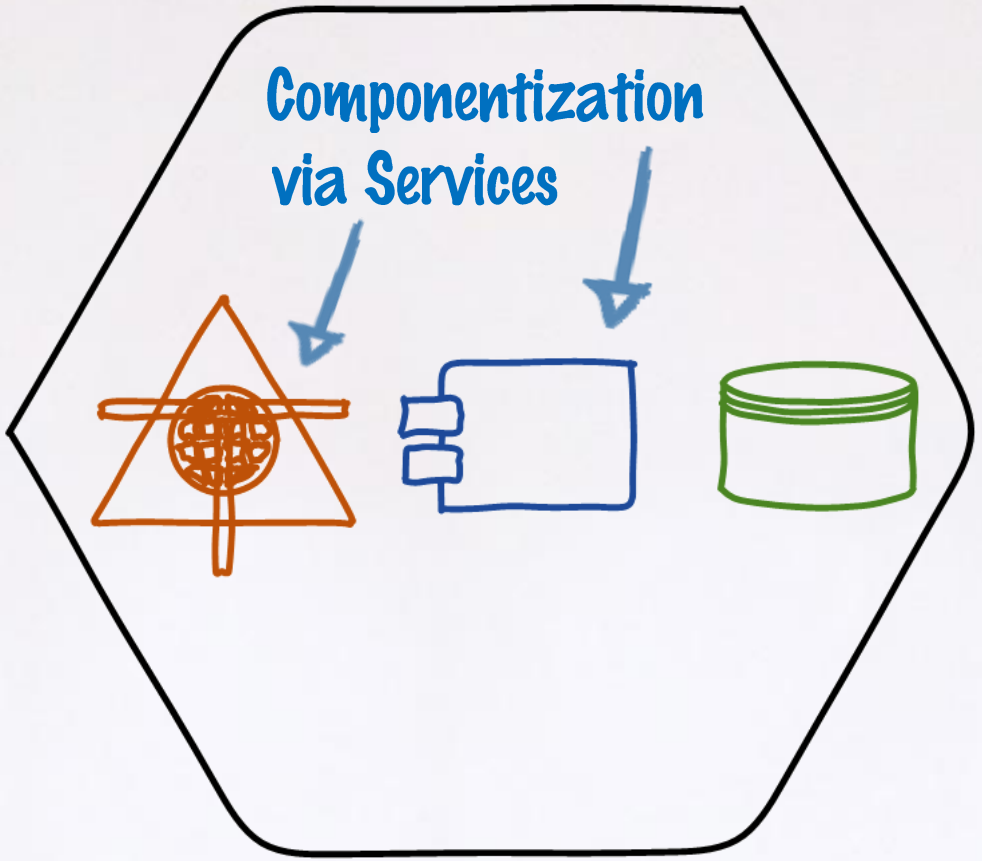
In a Nutshell

PURIST VIEW



What?

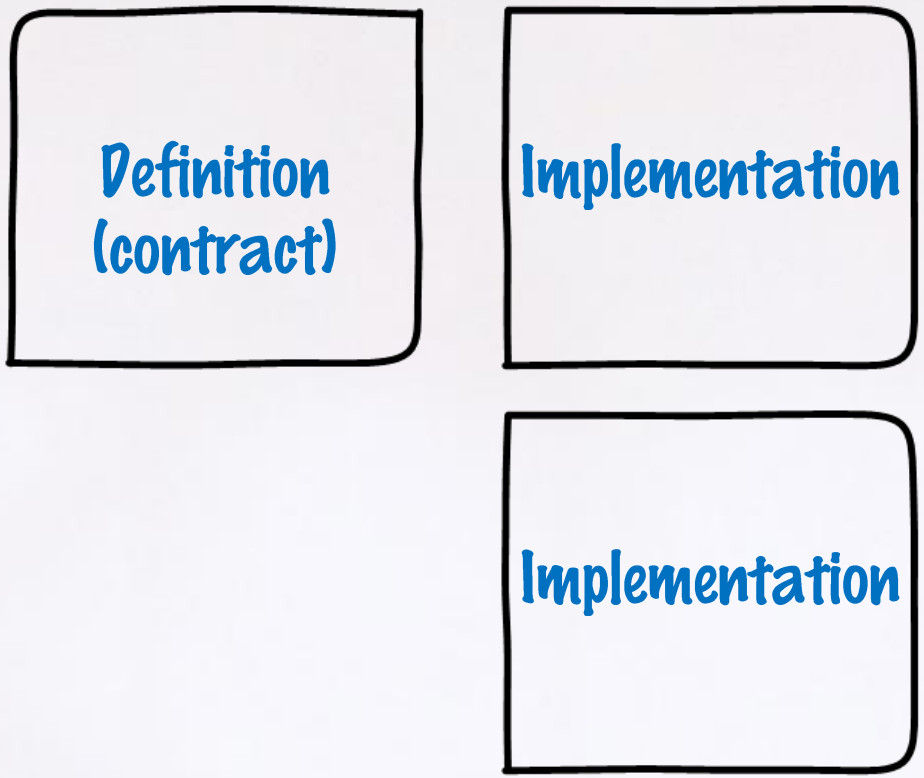
Characteristics



From "Characteristics of a Microservice" - Martin Fowler
http://goo.gl/AZwucZ

Sidebar 1

What is a Component?

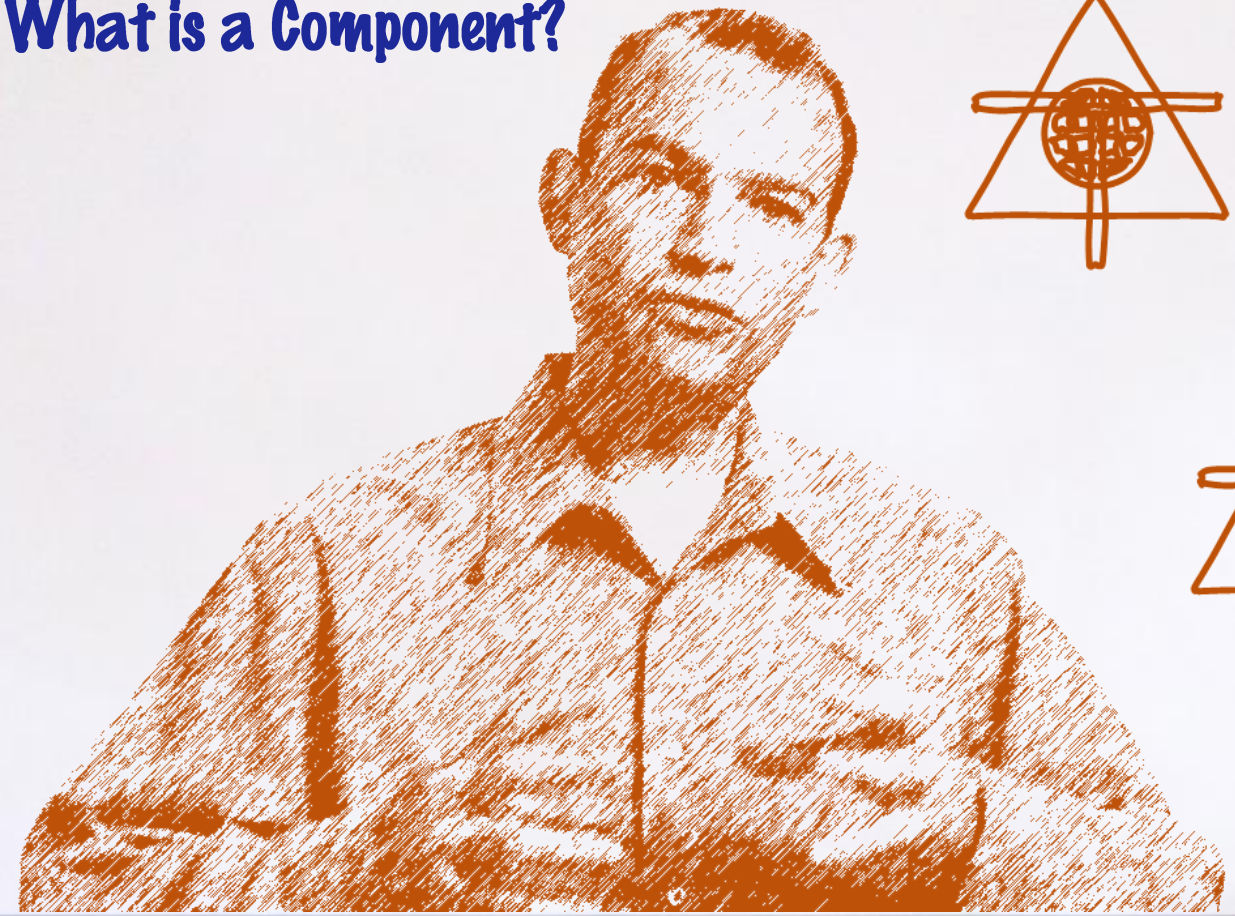


{Code PaLOUsa}

Microsoft .NET WCF

Sidebar 1

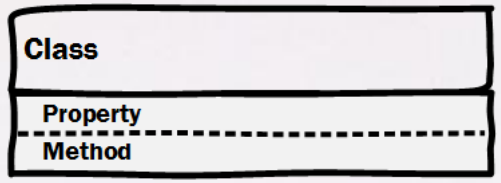
What is a Component?



Contract

Interface

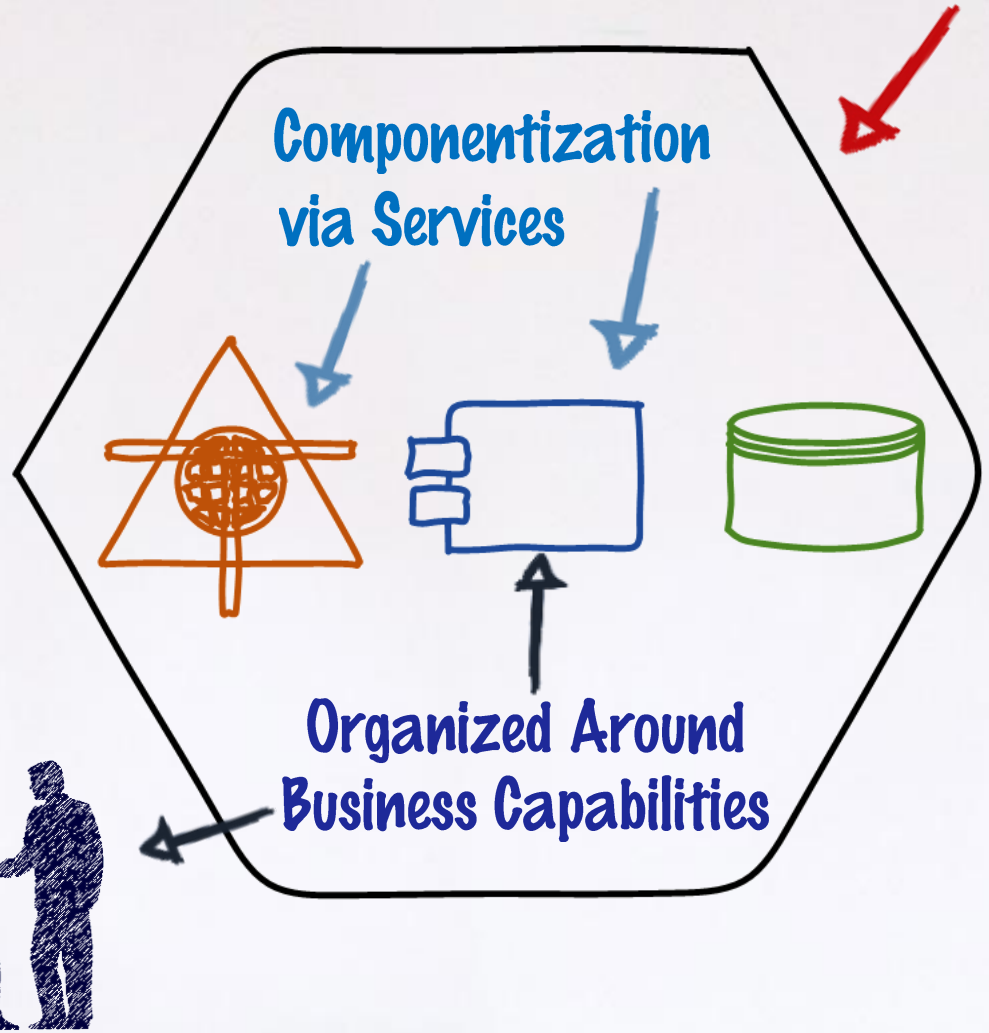
Implementation



{Code PaLOUsa}

What?

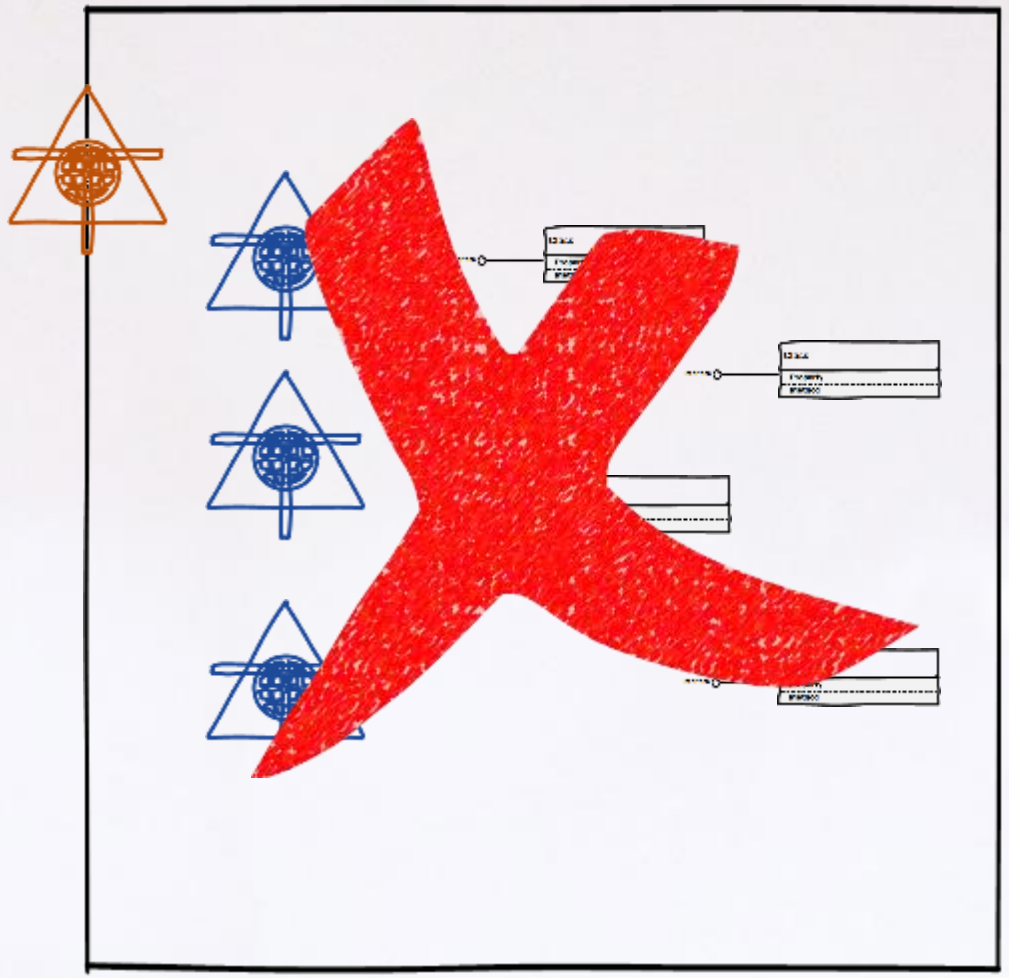
Characteristics



From "Characteristics of a Microservice" - Martin Fowler
http://goo.gl/AZwucZ

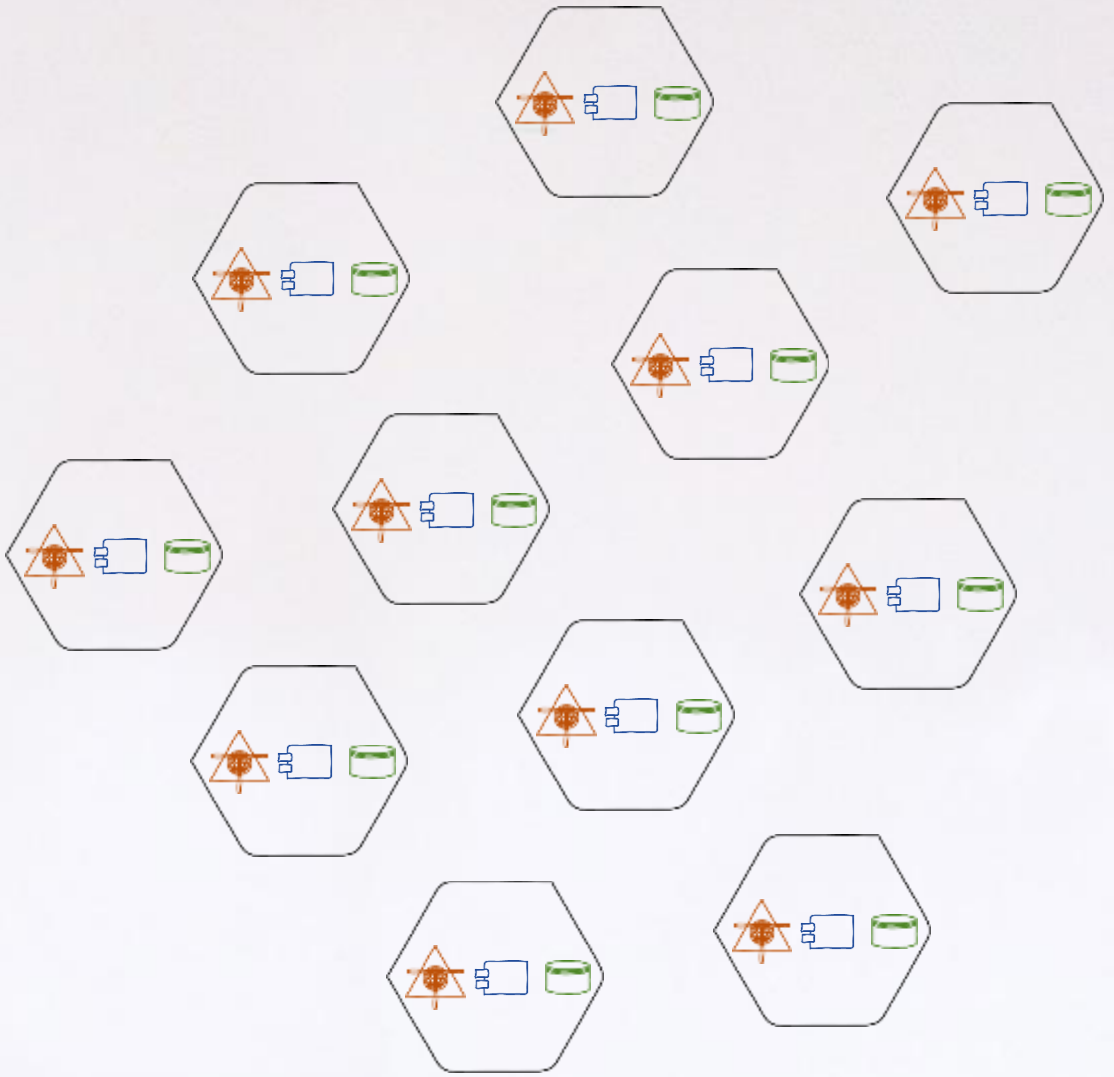
Sidebar 2

What is a Product?



Sidebar 2

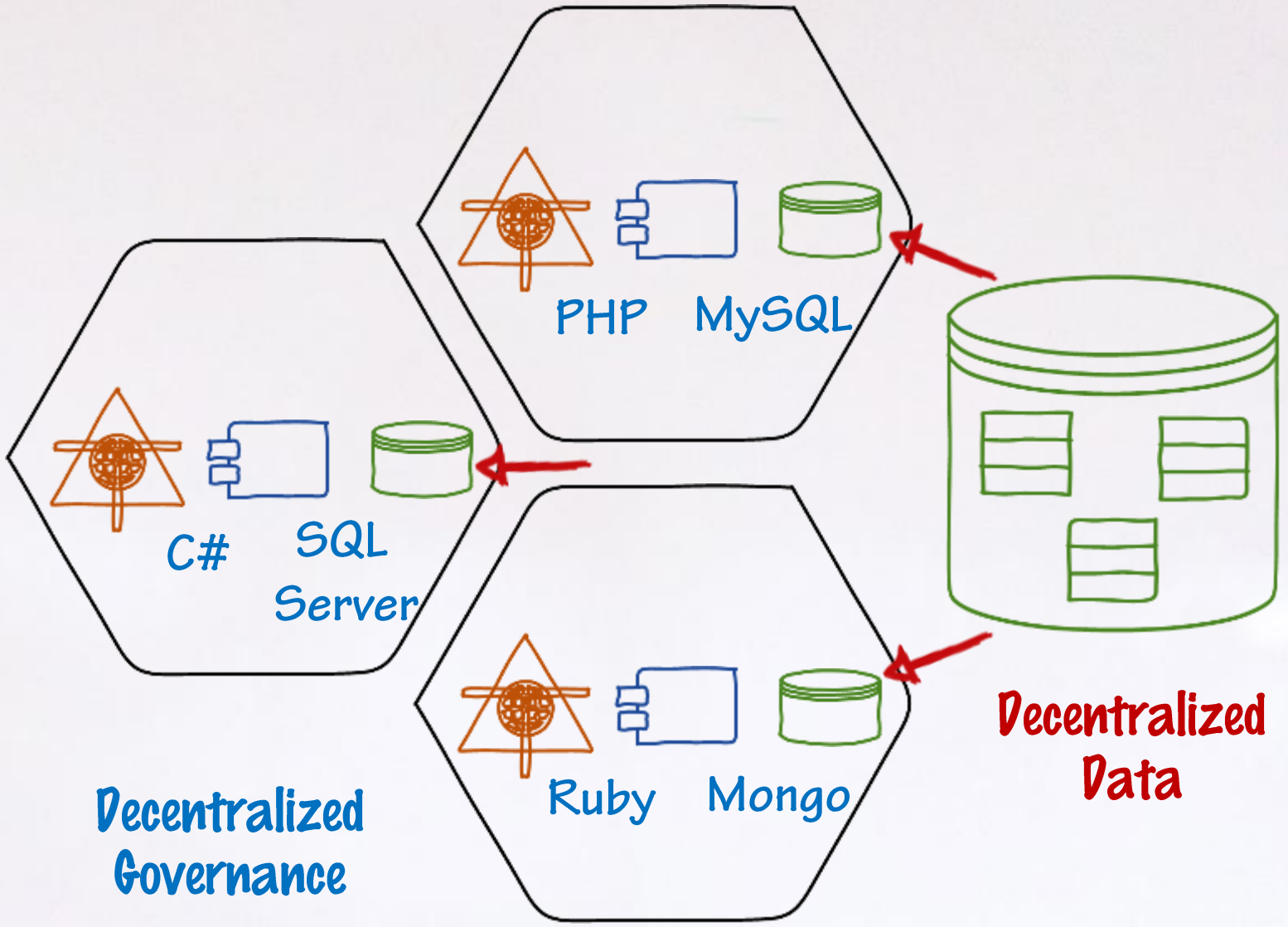
What is a Product?



What?

Characteristics

From "Characteristics of a Microservice"- Martin Fowler
http://goo.gl/AZwucZ

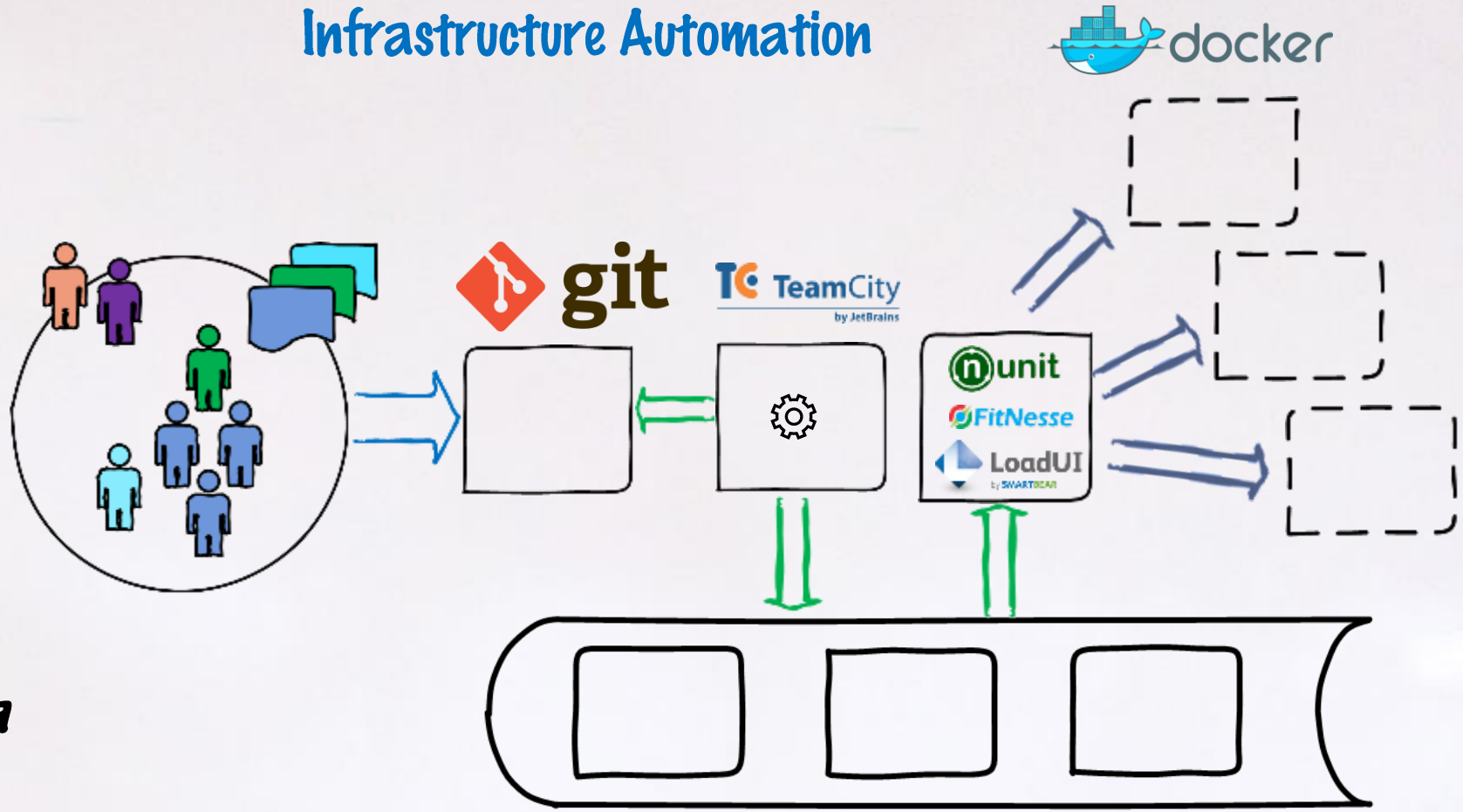


{Code PaLOUsa}

Infrastructure Automation

What?

Characteristics



From "Characteristics of a Microservice"-
Martin Fowler
http://goo.gl/AZwucZ

puppet

CHEF
CODE CAN

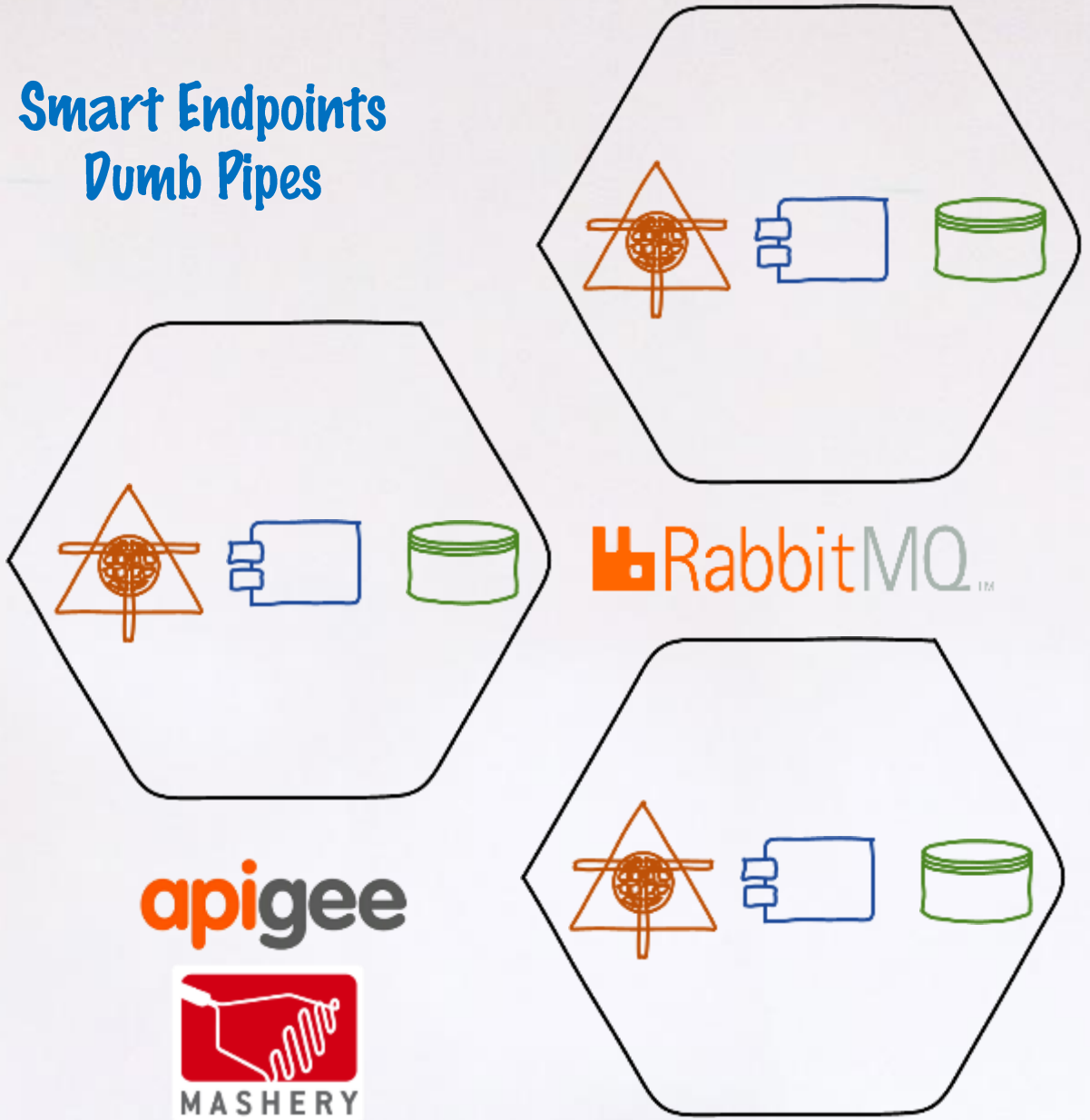
Bamboo

{Code PaLOUsa}

What?

Characteristics

Smart Endpoints
Dumb Pipes



RabbitMQ

apigee

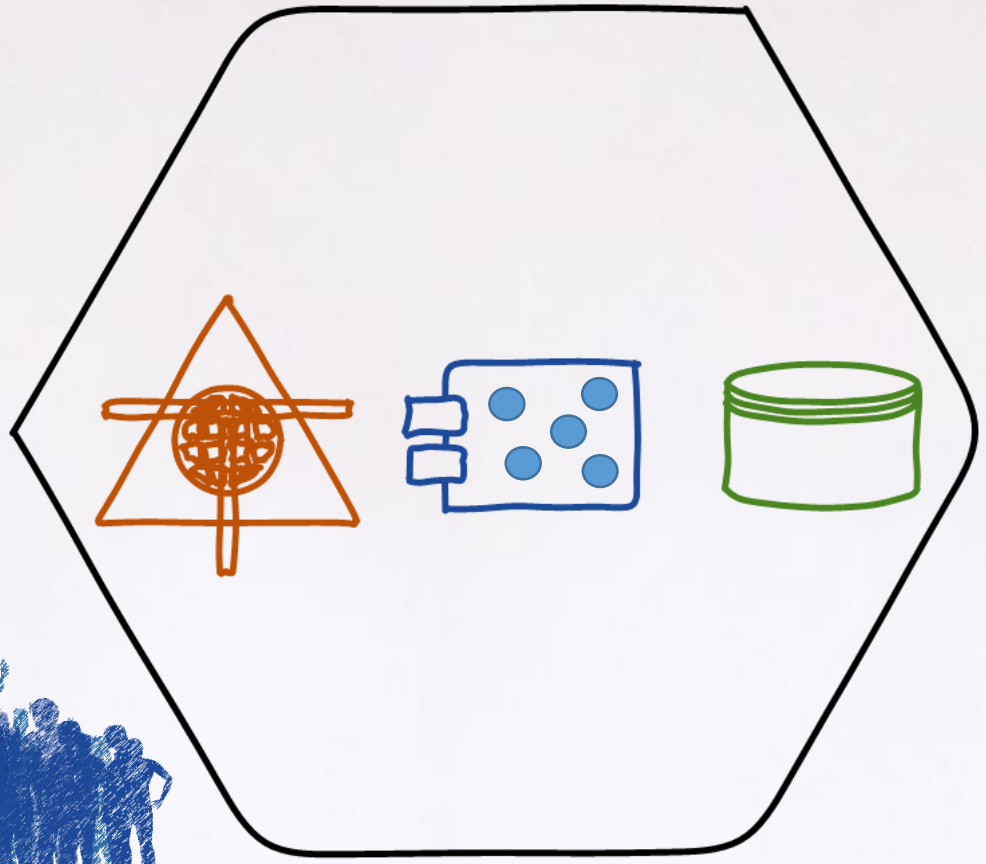
MASHERY

From "Characteristics of a Microservice"- Martin Fowler
http://goo.gl/AZwucZ

Evolutionary Design

What?

Characteristics



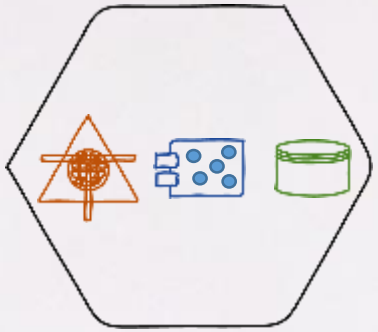
From "Characteristics of a Microservice"- Martin Fowler
http://goo.gl/AZwucZ

What?

Characteristics

Microservices

- Componentization via Services
- Organized around business capabilities



Agile Practices

- Products not Projects
- Decentralization (governance, data)
- Evolutionary Design

From "Characteristics of a Microservice" - Martin Fowler
http://goo.gl/AZwucZ

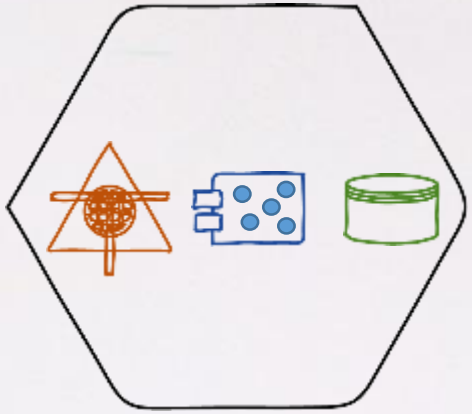
DevOps

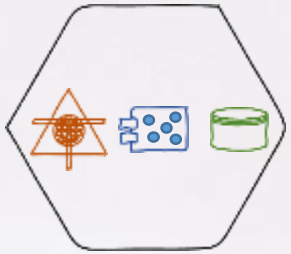
- Infrastructure Automation Smart Endpoints, Dumb Pipes

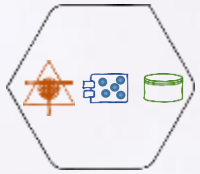
What?

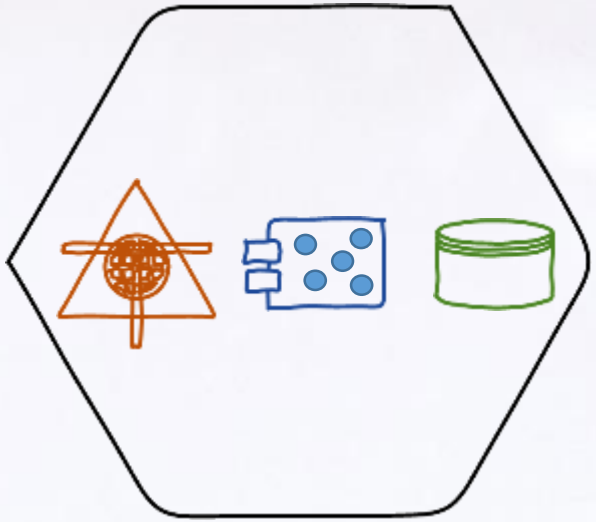
How Big is a Microservice?

It's **not** about lines of code!









If it is not about lines of code, what is it about?

Business Capabilities

{Code PaLOUsa}

Focus on Architecture

A Pragmatic Approach

Architectural Style

Focus on Architecture

Architectural Style

In short, the microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API

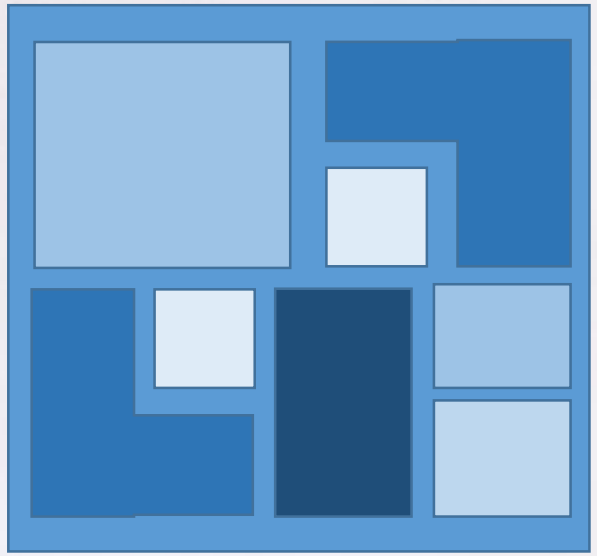
From "Characteristics of a Microservice"-
Martin Fowler
http://goo.gl/AZwucZ

{Code PaLOUsa}

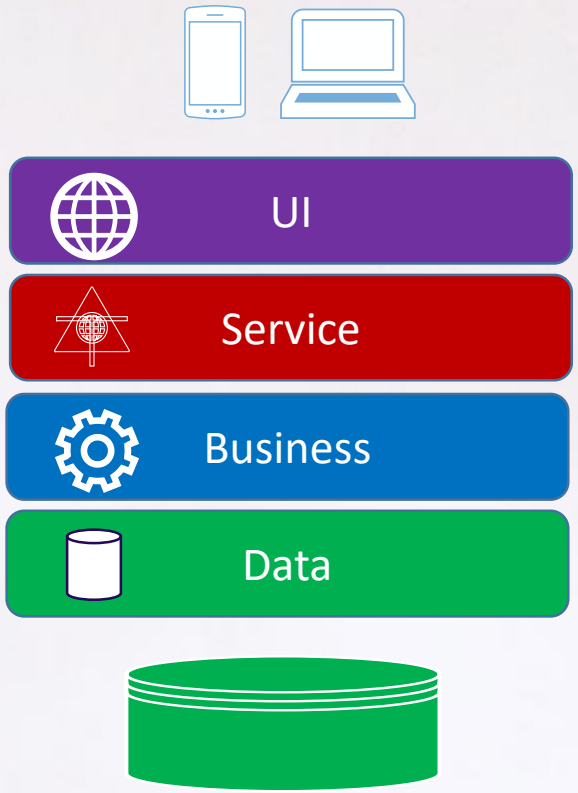
Independent Scale — Flexibility — Loose Coupling — Tight Cohesion

Focus on Architecture

Monolith versus Microservices

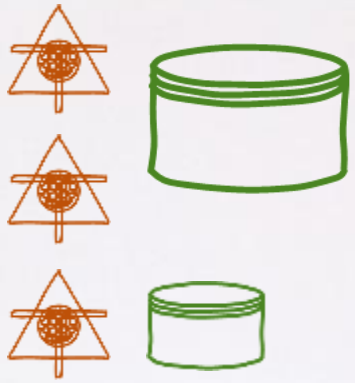


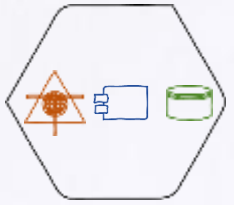
Monolithic Architecture



Layered Architecture

SOA





Microservice Architecture

In a Properly Architected Solution, What is the difference between a monolith and a microservice

Boundaries and deployment methodologies

Focus on Architecture

Microservices are Easy?

- Easy to Understand
- Easy to Enhance
- Easy to Test
- Easy to Deploy

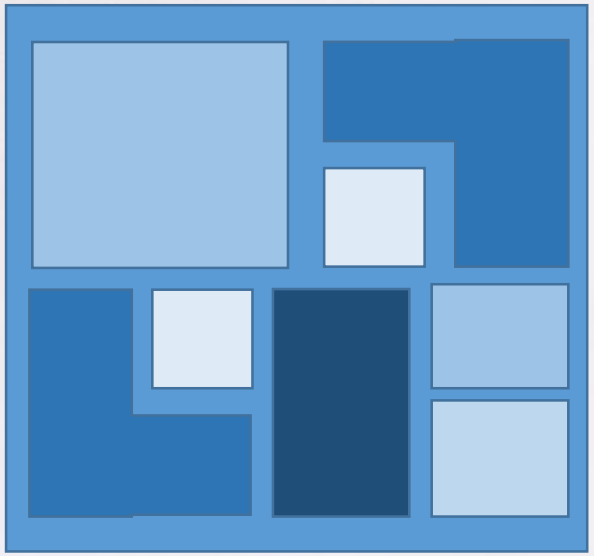
Focus on Architecture

Microservices are Easy?

Item	Microservice	Monolith
Lines of Code	500	1,000,000
Easier to understand	✓	A+
Easier to enhance	✓	
Easier to test	✓	
Easier to deploy	✓	

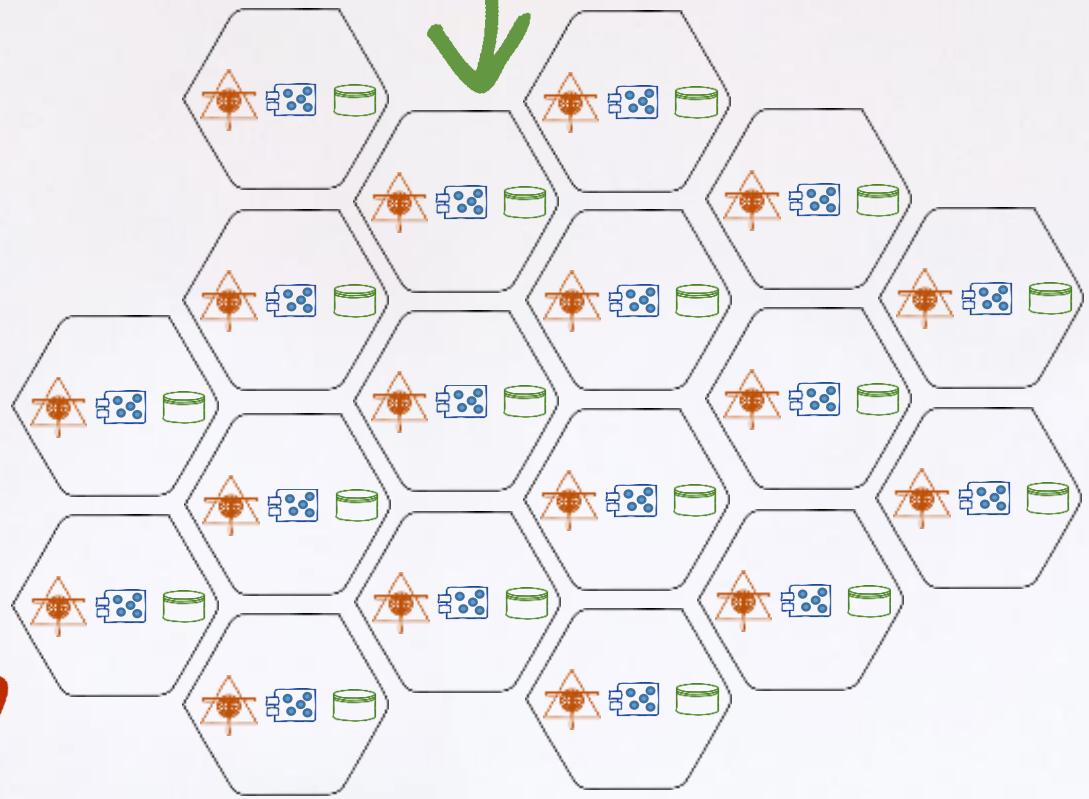
Focus on Architecture

Microservices are Easy?



This is not easy

This is easy

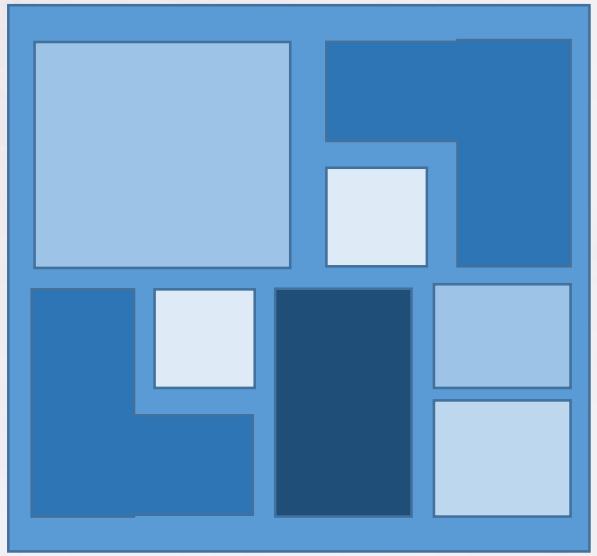


Inner Complexity

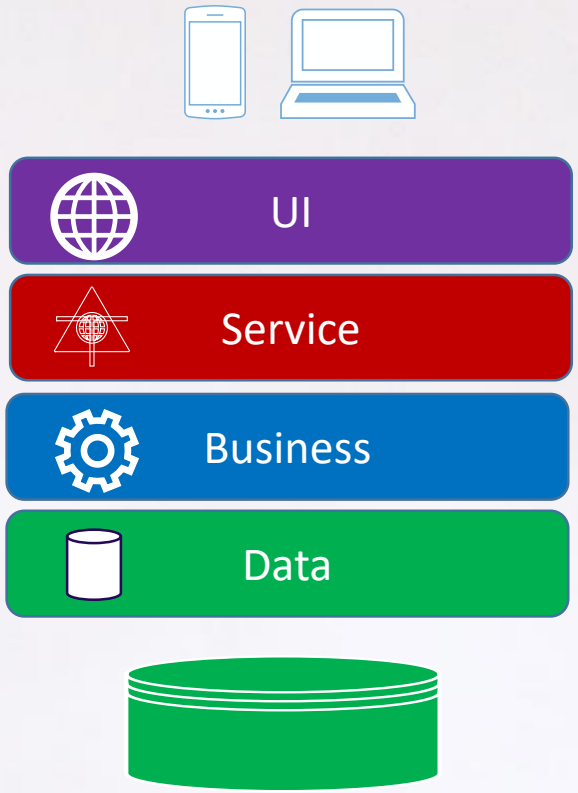
Outer Complexity

Architecture

Complexity

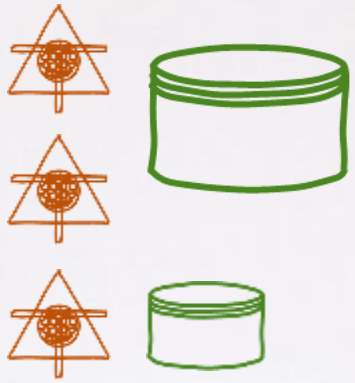


Monolithic Architecture

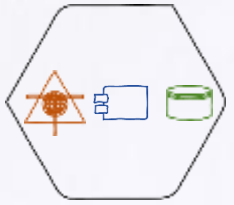


Layered Architecture

SOA

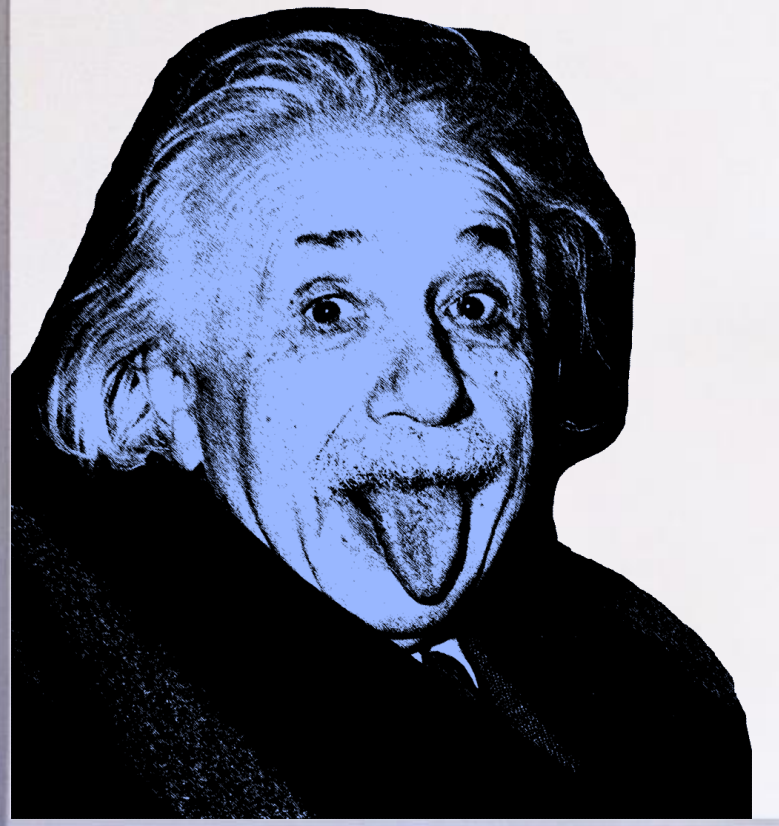


Microservice Architecture



Focus on Architecture

Plagiarism



Energy cannot be created or destroyed, it can only be changed from one from to another.

– Albert Einstein

Focus on Architecture

Plagiarism

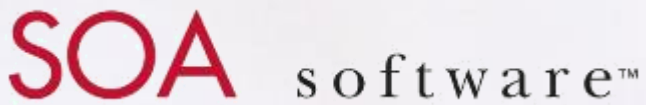
Complexity cannot be destroyed, it can only be changed from one from to another.

– Gregory A. Beamer

{Code PaLOUsa}

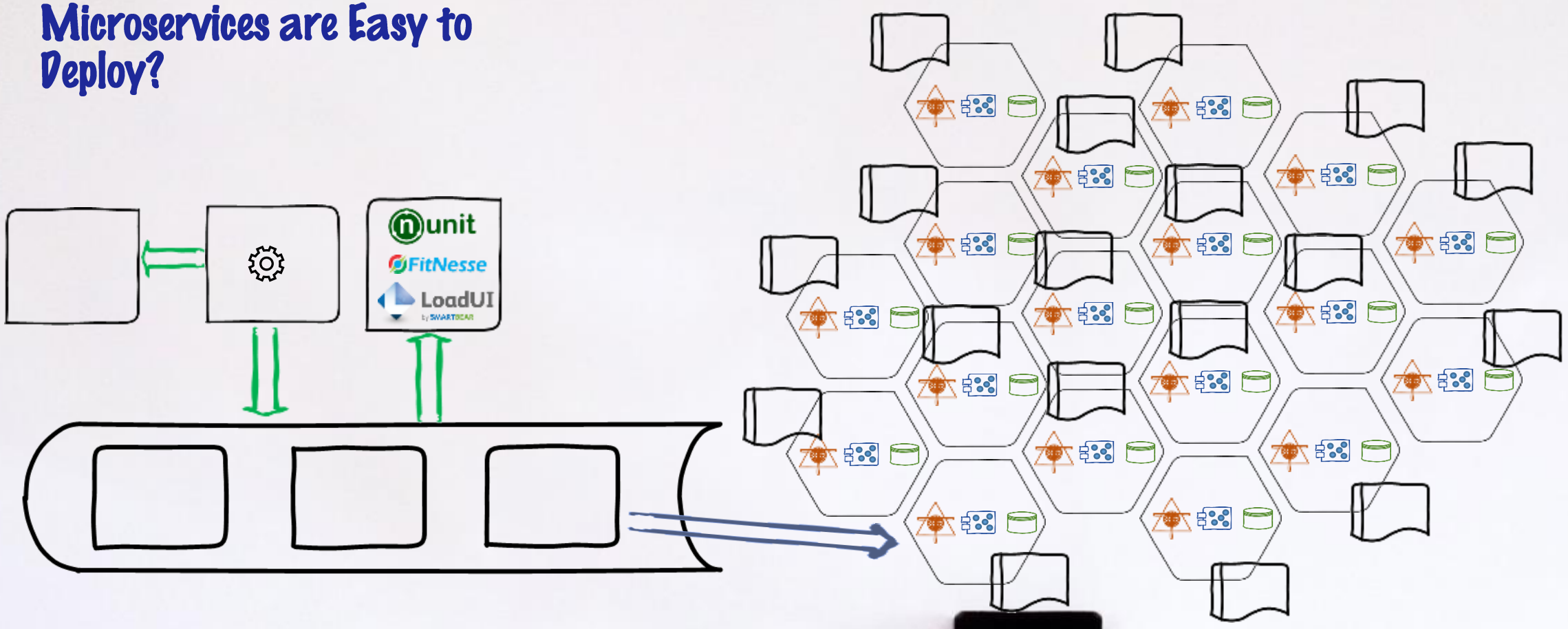
Why would you prefer outer complexity over inner complexity?

Because you can commoditize it!!!



Focus on Architecture

Microservices are Easy to Deploy?



{Code PaLOUsa}

Why spend all that time automating?

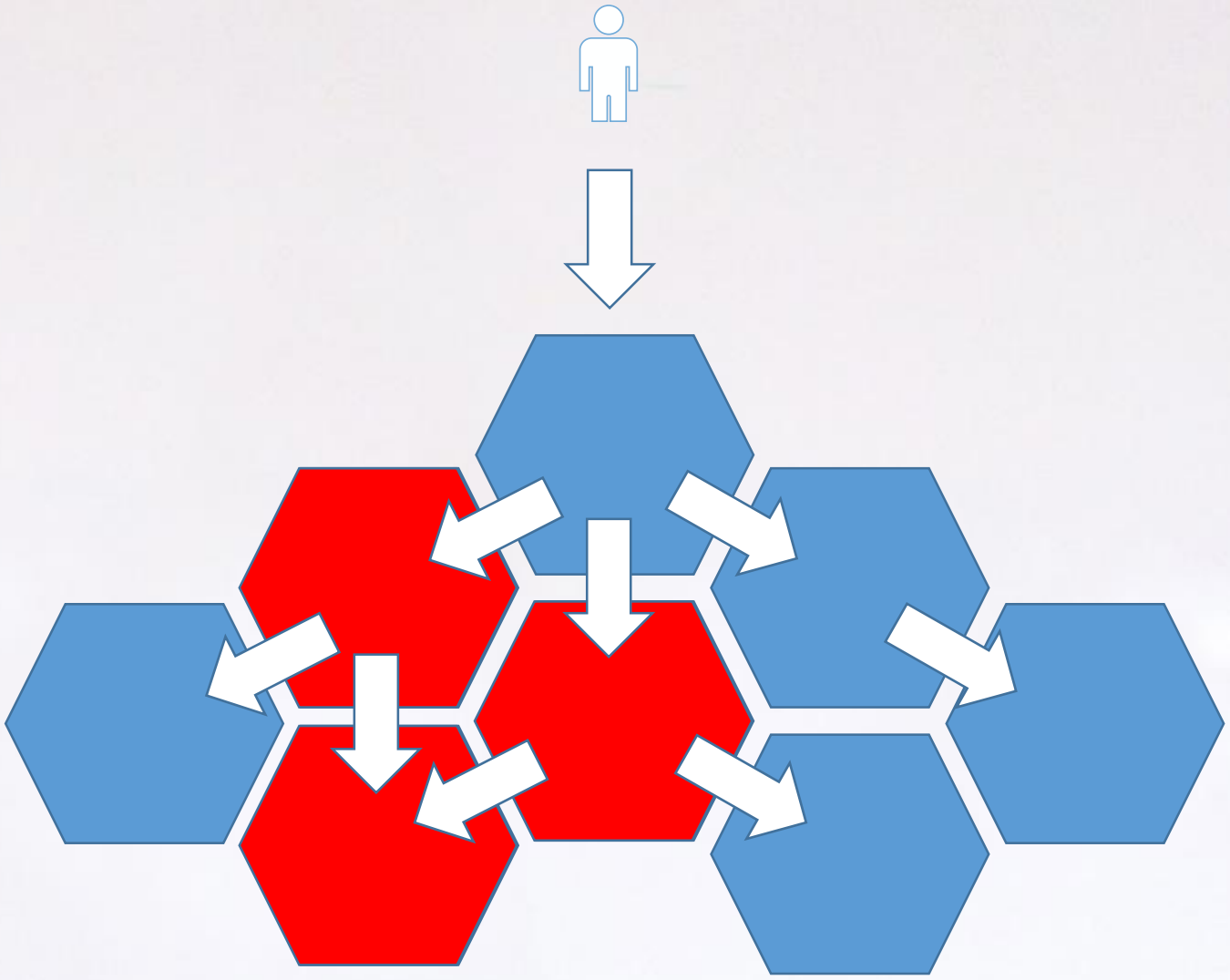
Because your ops people have better things to do

Because you can commoditize it!!!

Focus on Architecture

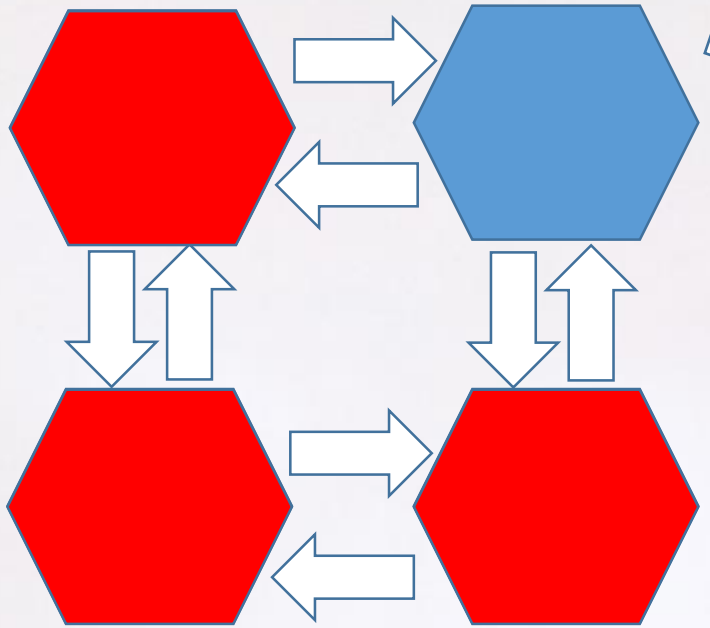
Microservices are Reslient?

Low Impact on Other Services

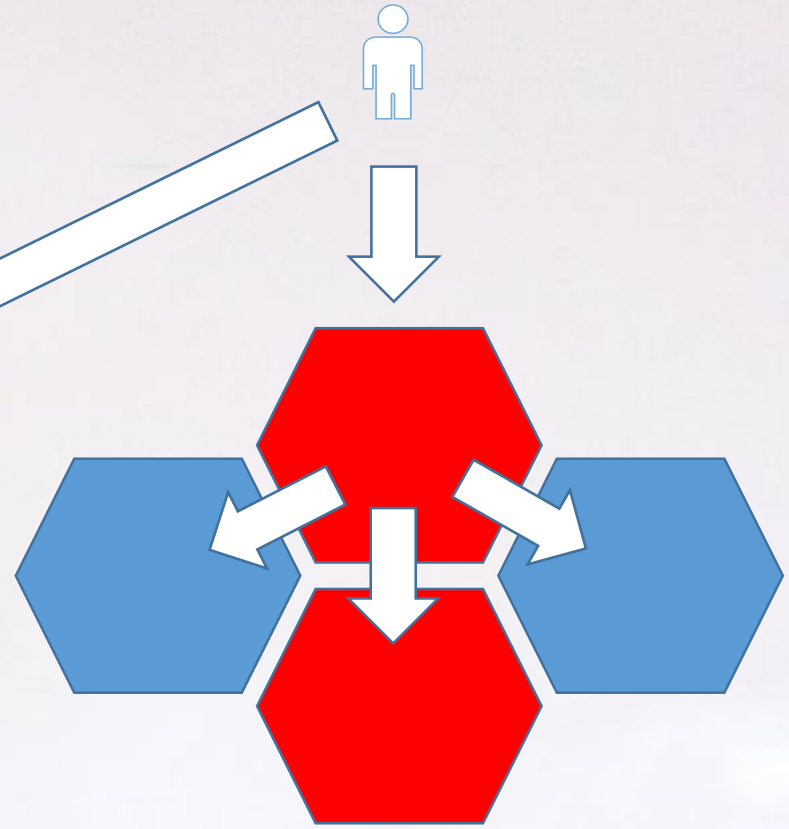


Focus on Architecture

Microservices are Resilient?



Choreography



Orchestration

{Code PaLOUsa}

How do you insure low impact on other services?

kubernetes by Google

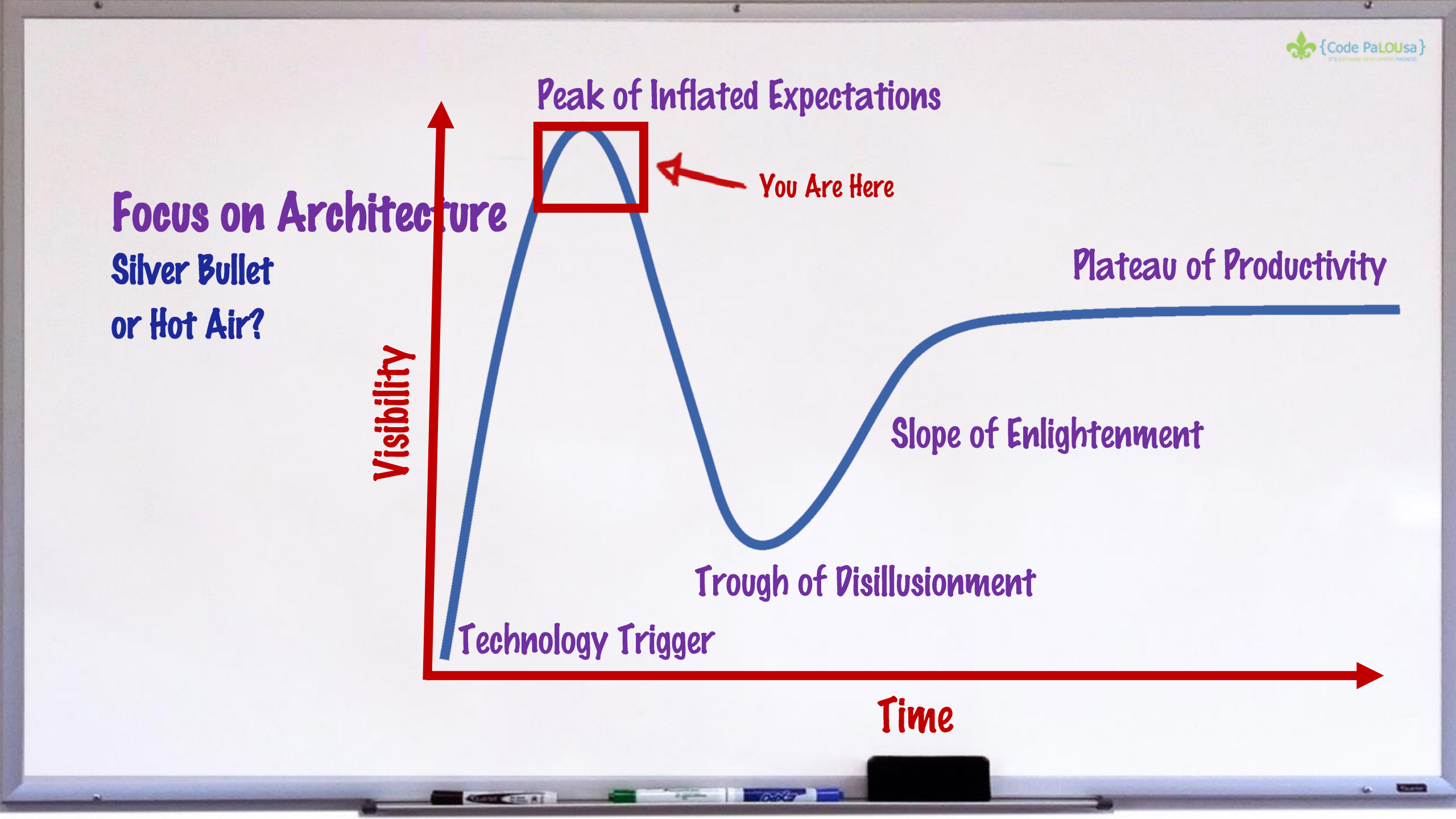
docker

Commoditize ...

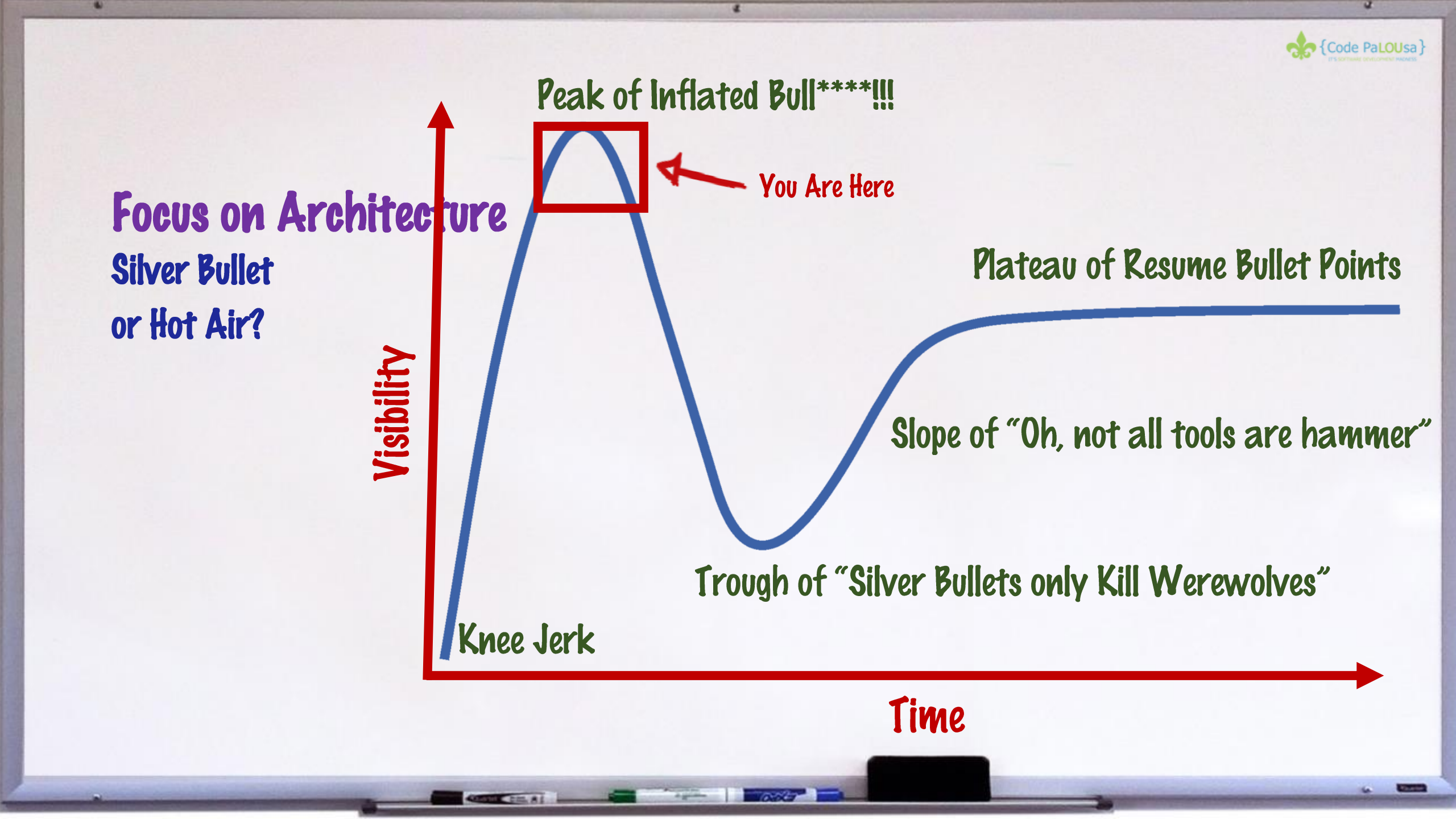
AppDynamics

apigee

... and WORLD PEACE!



{Code PaLOUsa}



Microservices & the Real World

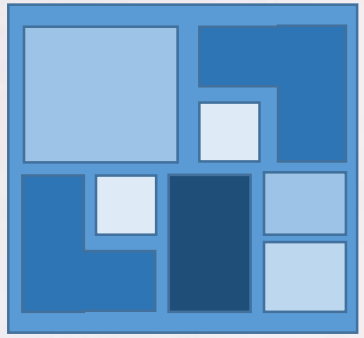
Implementing Microservices

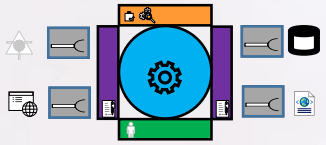
Microservices & the Real World

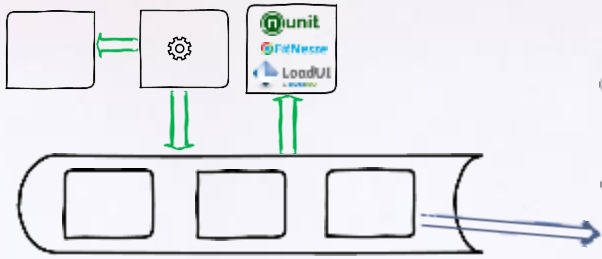
Architecture

Are You Business Focused?

Do You Have Clear Boundaries?







Current State

Is the Code Organized?

Can You Validate?

Are you Automated?

Desired State

CASA Development Model

Core AS Application

Microservices & the Real World

Organize

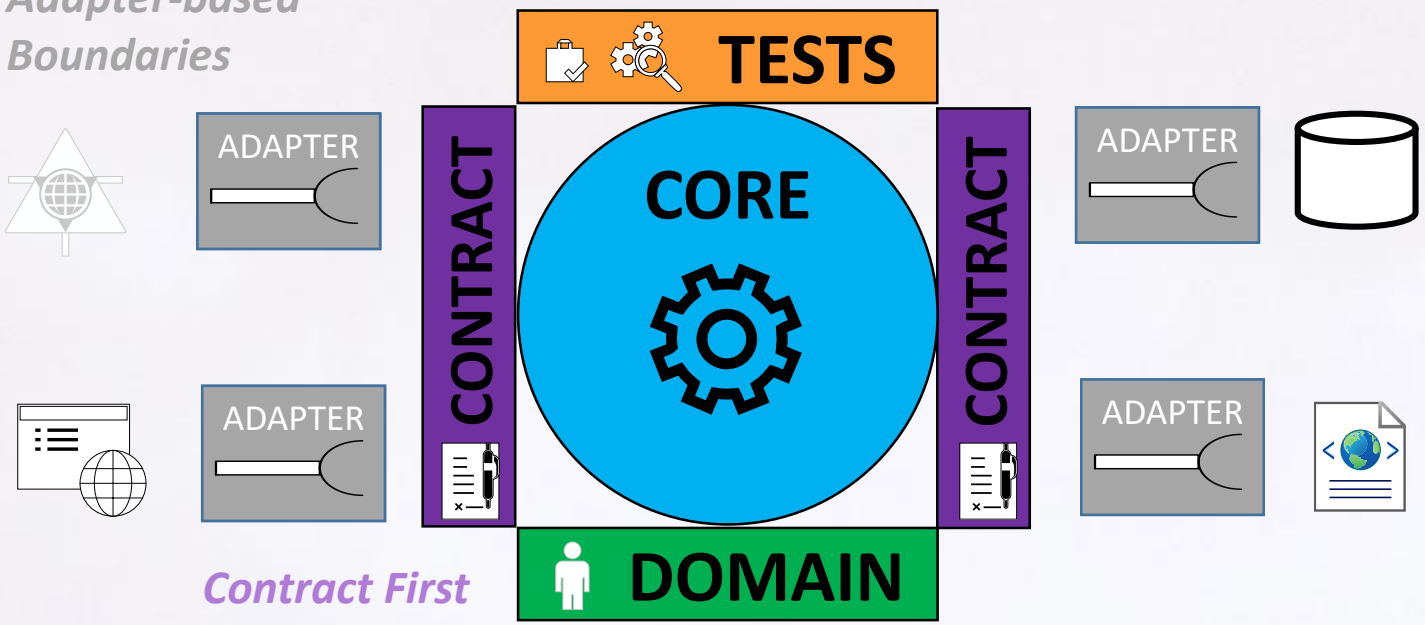
Test Driven/ Test Early Approach

TDD

BDD

ATDD

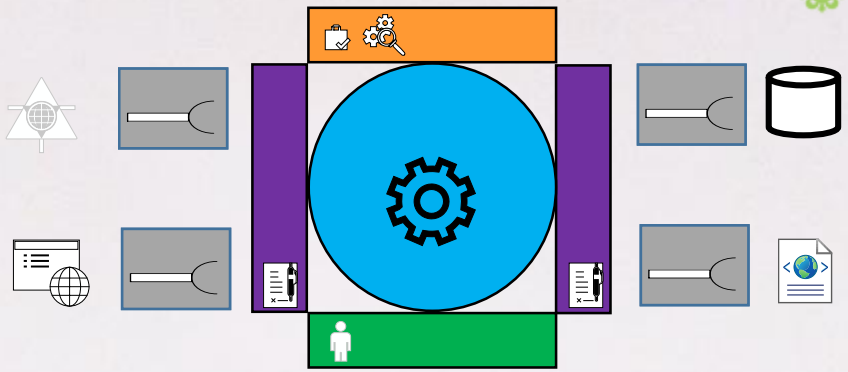
Adapter-based Boundaries



Contract First Development

Domain Driven Design

{Code PaLOUsa}



Solution Explorer

Search Solution Explorer (Ctrl+;)

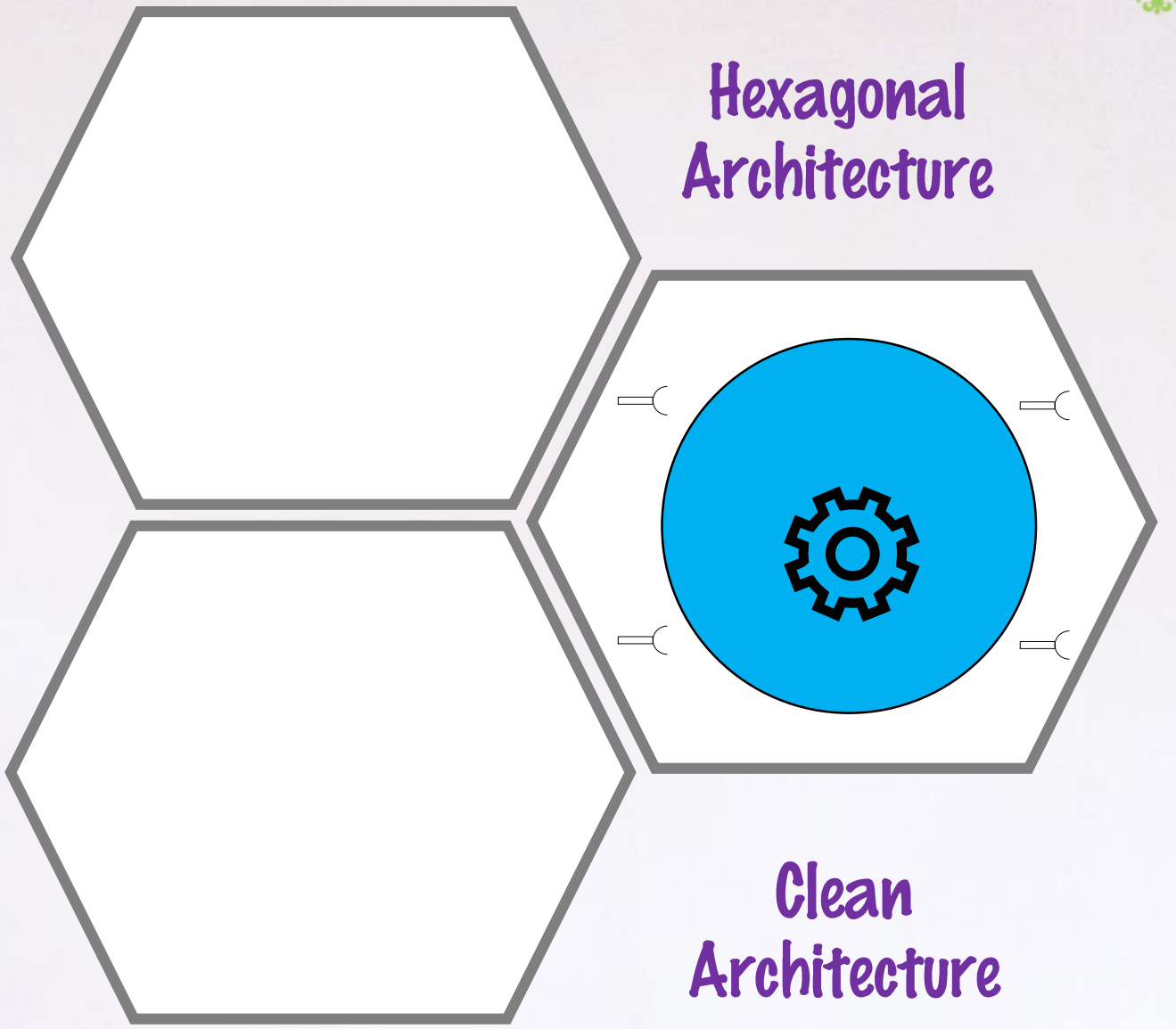
- Solution 'Microservices.Template' (13 projects)
  - Core
    - Microservice.Template.Core
    - Microservice.Template.Core.Contracts
    - Core.txt
  - Domain
    - Microservices.Template.Domain.Exceptions
    - **Microservices.Template.Domain.Models**
    - Domain.txt
  - Framework
    - Framework.DependencyInjection
    - Framework.txt
  - Persist
    - Microservice.Template.Persist.Contracts
    - Microservice.Template.Persist.Sql
    - Persist.txt
  - Present
    - MicroserviceTemplate.Present.Web
    - Present.txt

- Service
  - Microservice.Template.Service
  - Service.txt
- Test
  - Accept
  - Integ
    - Microservice.Template.Core.Test.Integ
    - Microservice.Template.Persist.Sql.Test.Integ
    - test.integ.txt
  - Spec
  - Unit
    - Microservice.Template.Core.Test.Unit
    - MicroserviceTemplate.Present.Web.Test.Unit
    - test.unit.txt
  - Test.txt

Solution Explorer | Team Explorer | Class View

Microservices & the Real World

Organize

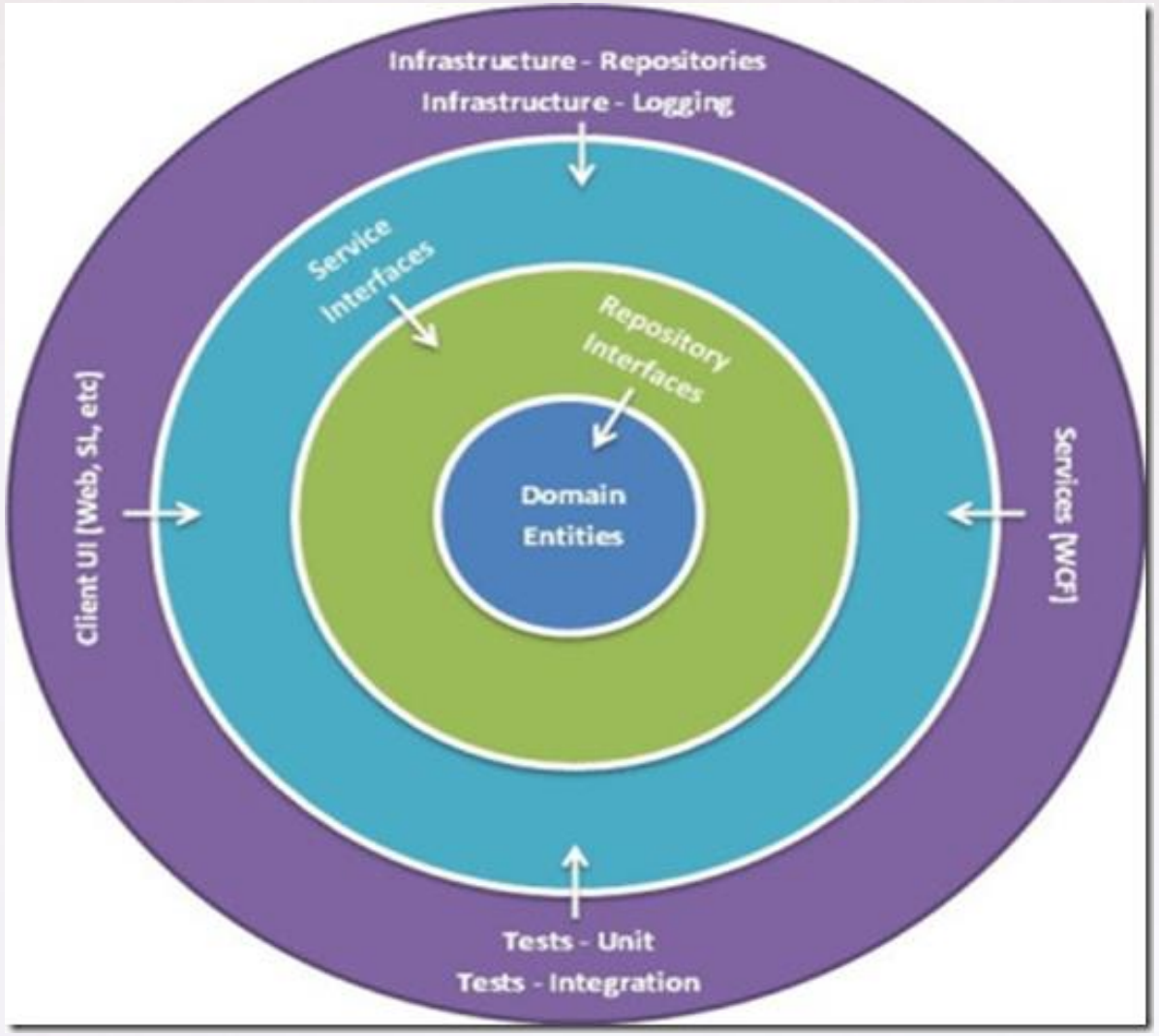


Hexagonal Architecture

Clean Architecture

Microservices & the Real World

Organize



CASA Development Model

Core AS Application

Microservices & the Real World

Organize

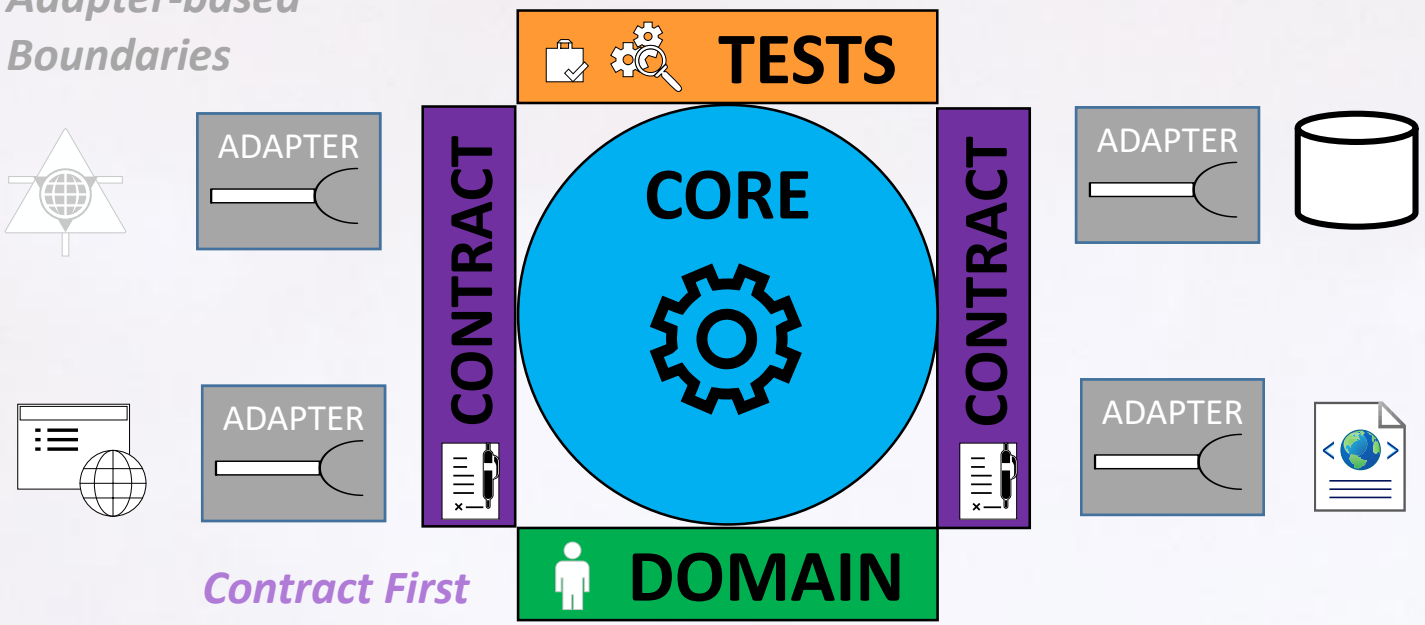
Test Driven/ Test Early Approach

TDD

BDD

ATDD

Adapter-based Boundaries

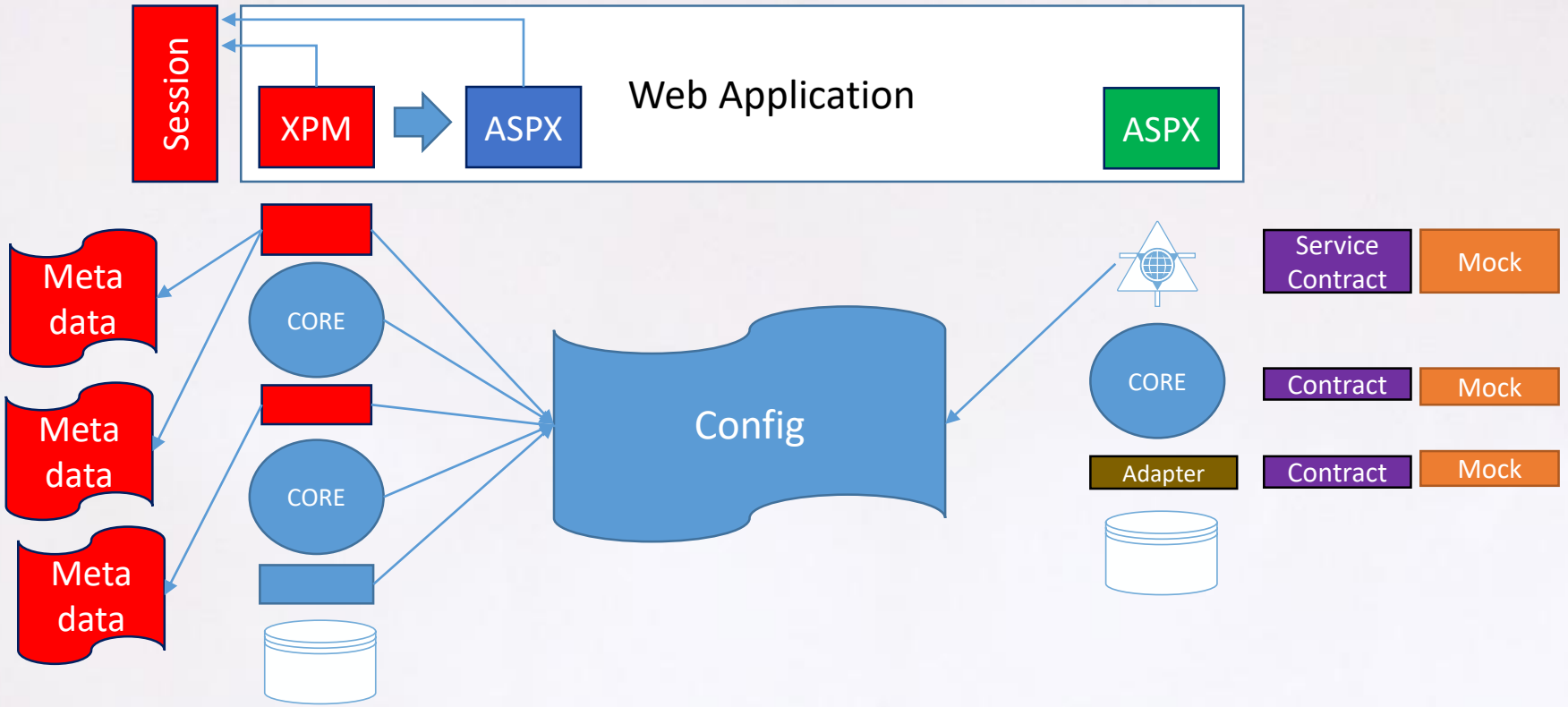


Contract First Development

Domain Driven Design

Microservices & the Real World

Organization



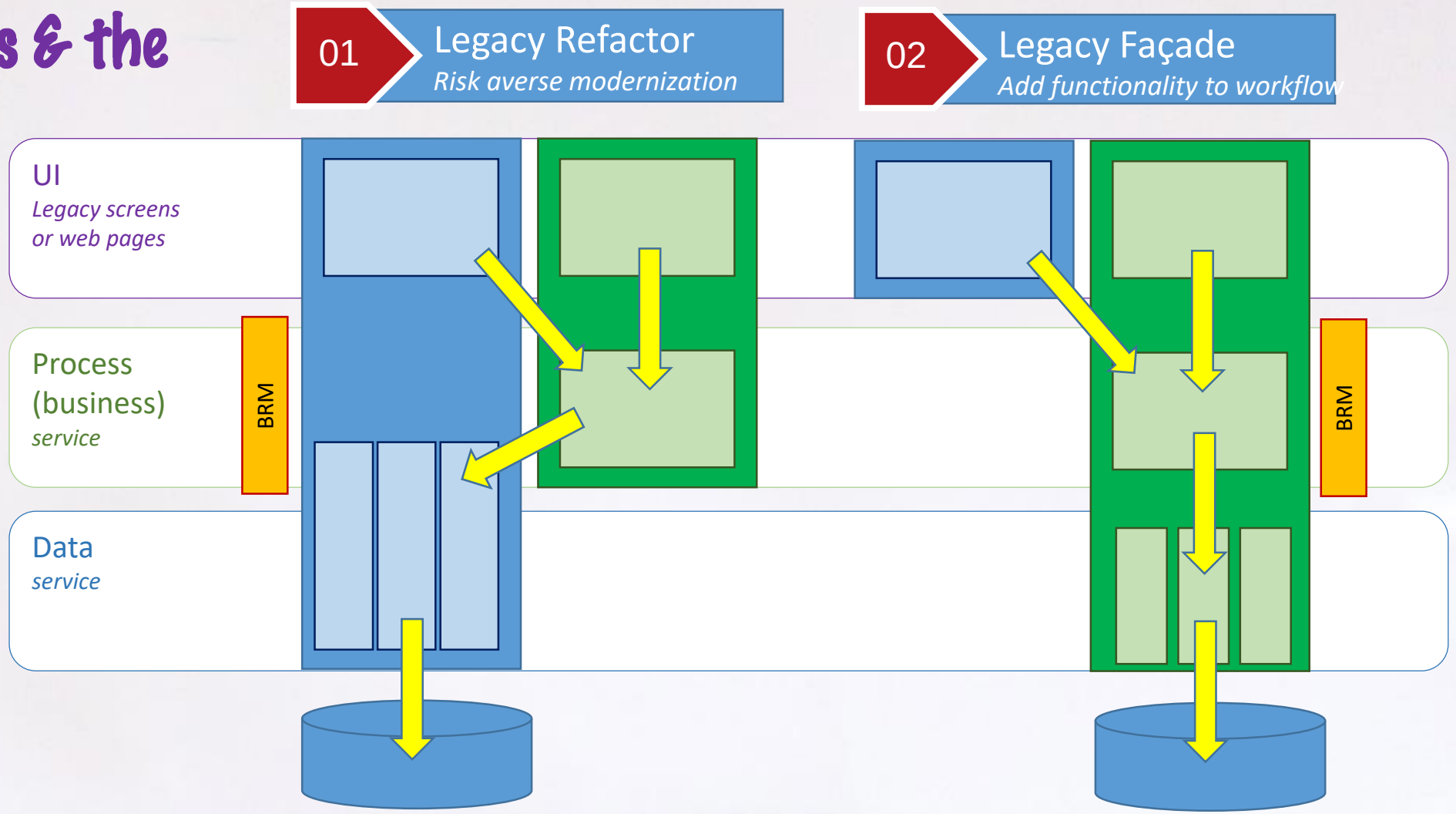
EXAMPLE: Strangulation

Improper Patterns

Microservices & the Real World

Organization

EXAMPLE: Legacy Modernization



Microservices & the Real World

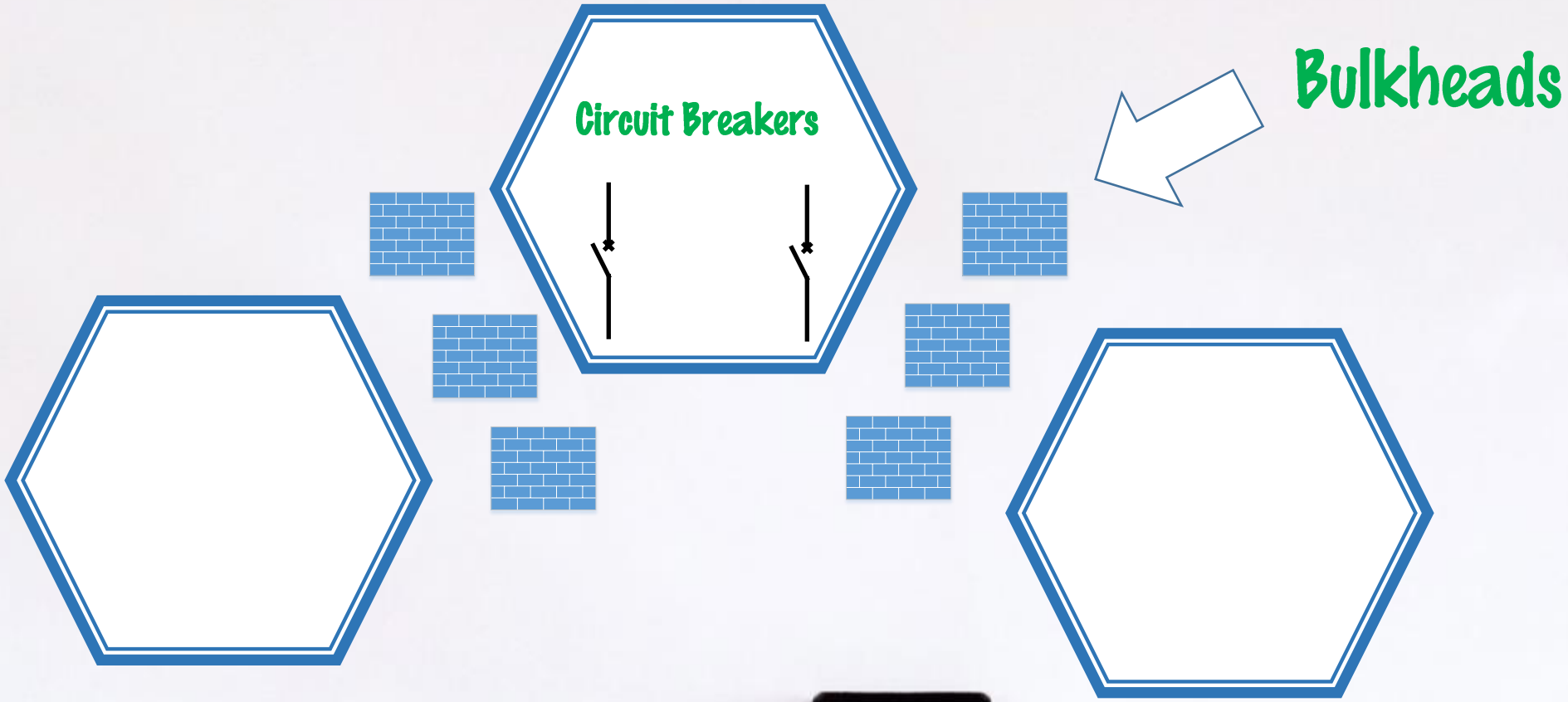
Best Practices

- SOLID Design Principles
  - You know these right?
- Domain Driven Design
  - State Objects
  - Core Business Logic
  - Other Concerns
- Contract First Development
  - Customer Driven Contracts

Microservices & the Real World

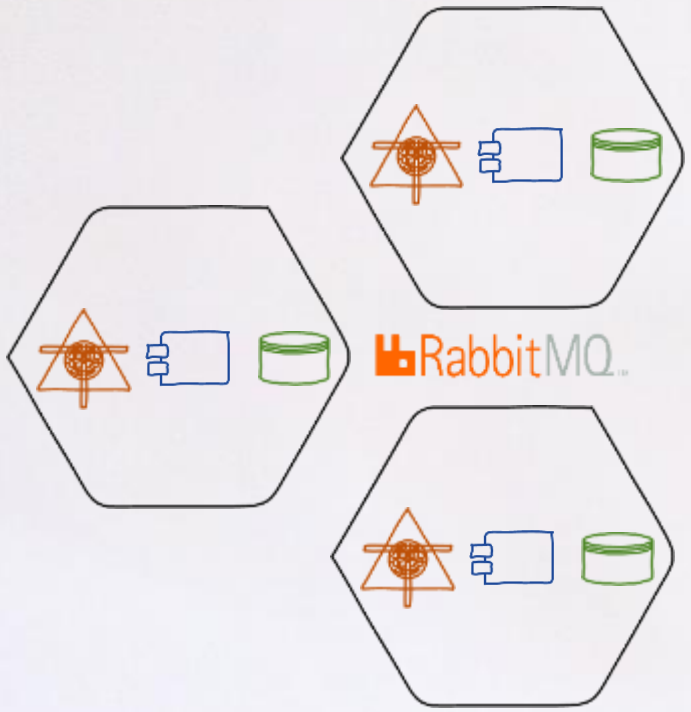
Safety First

- Circuit Breakers
- Bulkheads



Microservices & the Real World

Clear Boundaries



MuleSoft™

{Code PaLOUsa}

Experience APIs (UI Enablement)

Process APIs (Business Focused Microservices)

System APIs (Data Services)

SYSTEM

APPLICATION

DATABASE

FOLDER

FILE

Microservices & the Real World

Organize Around Capabilities

- What is a Capability?

Any business function that can be coded as a stand alone

Any business function that can easily be handed off to another team

Any business function that can be handled by another company

Microservices & the Real World

Organize Around Capabilities

- Which of these are capabilities?

Shopping Cart?

Purchase Path?

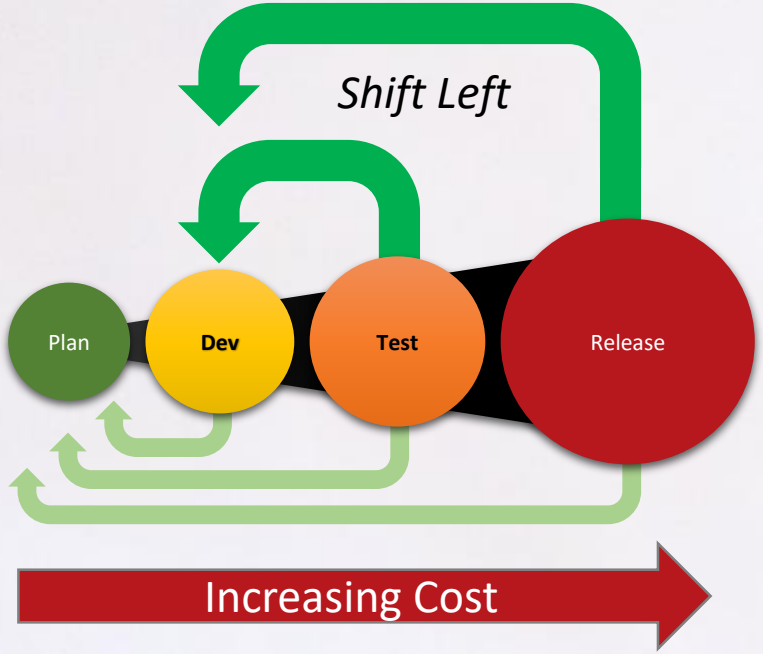
Company Catalog?

User Access?

Database Access?

Microservices & the Real World

Testability



- Developer Tests
  - Unit Tests
  - Integration Tests (basic regression)
  - Acceptance Tests
- Frameworks
  - {x}Unit – MSTest, nUnit, xUnit, Jasmine, Karma
  - Behavior – SpecFlow
  - Acceptance - Fitnesse

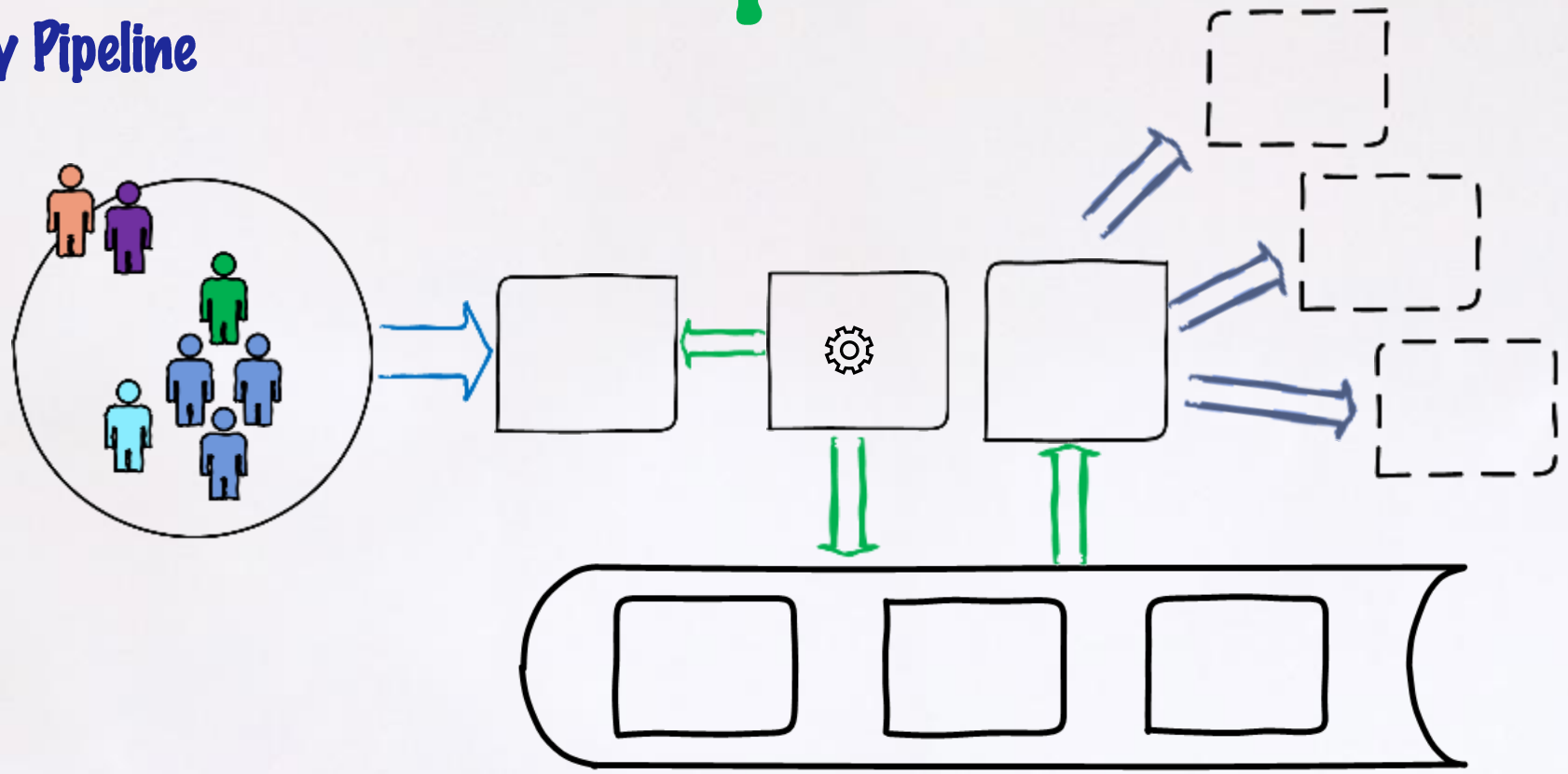
Microservices & the Real World

Testability

Microservices & the Real World

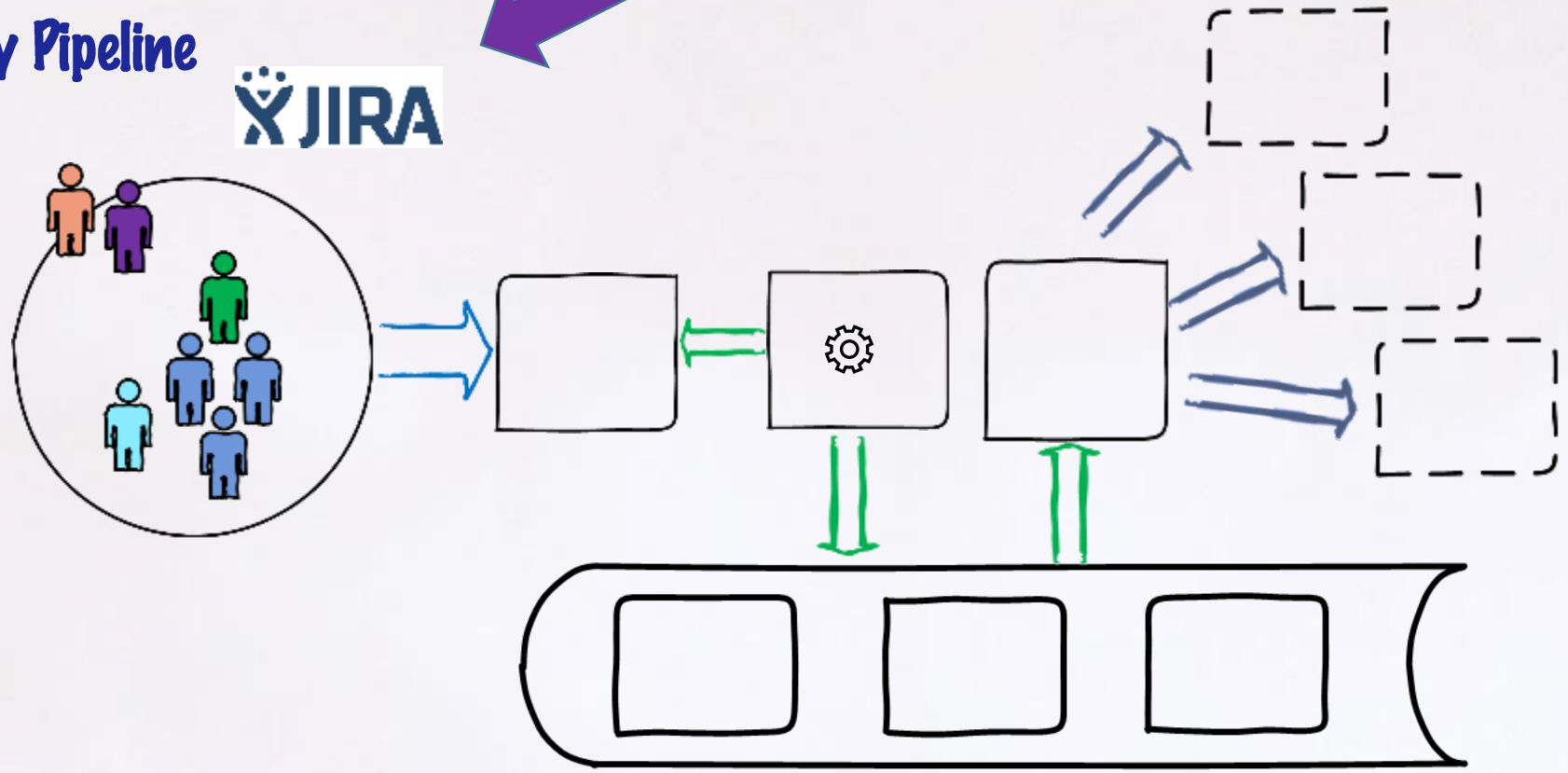
Automate the Delivery Pipeline

DevOps

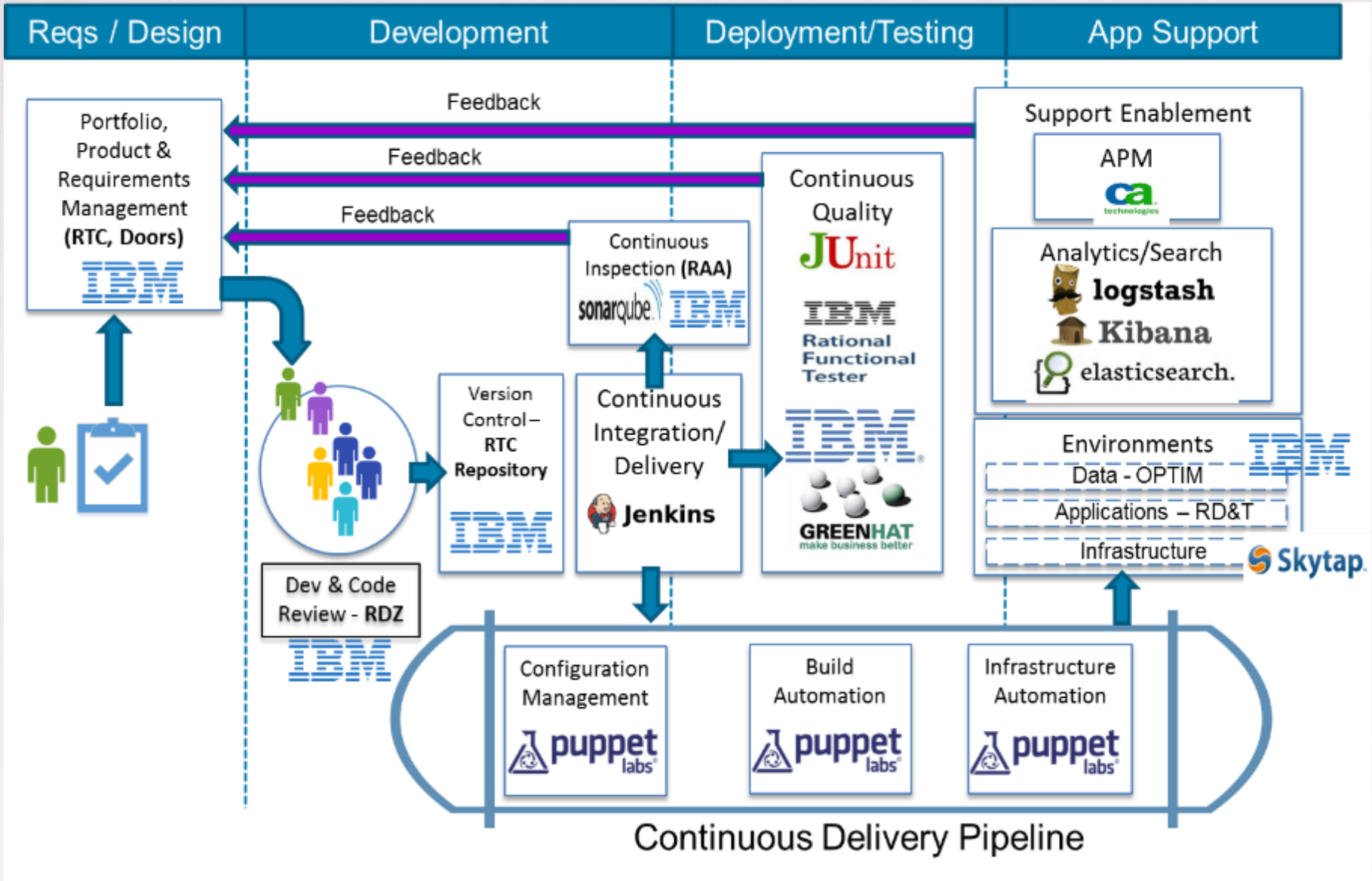


Microservices & the Real World

Automate the Delivery Pipeline



Feedback loops



Questions and Random BS

If you don't ask anything, I will continue talking!!!

Code Mash Edition

{Code PaLOUsa}

MICROSERVICES

Lessons from the Trenches

@gbworld

Here in the Real World

Inspired by Real Events

Real World

Determining Services: Decomposition

Let's Make Asssumptions

- We are a board game company
- We recently completed a kickstarter
- The game is selling
- We need to go beyond the kickstarter and sell on our own

{Code PaLOUsa}

Shopping Cart

Real World

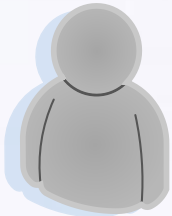
Determining Services: Decomposition

Shopping Cart

Item Lookup

Wish Lists

Security



User

Payment

Real World

Determining Services: Decomposition

What if we are green field?

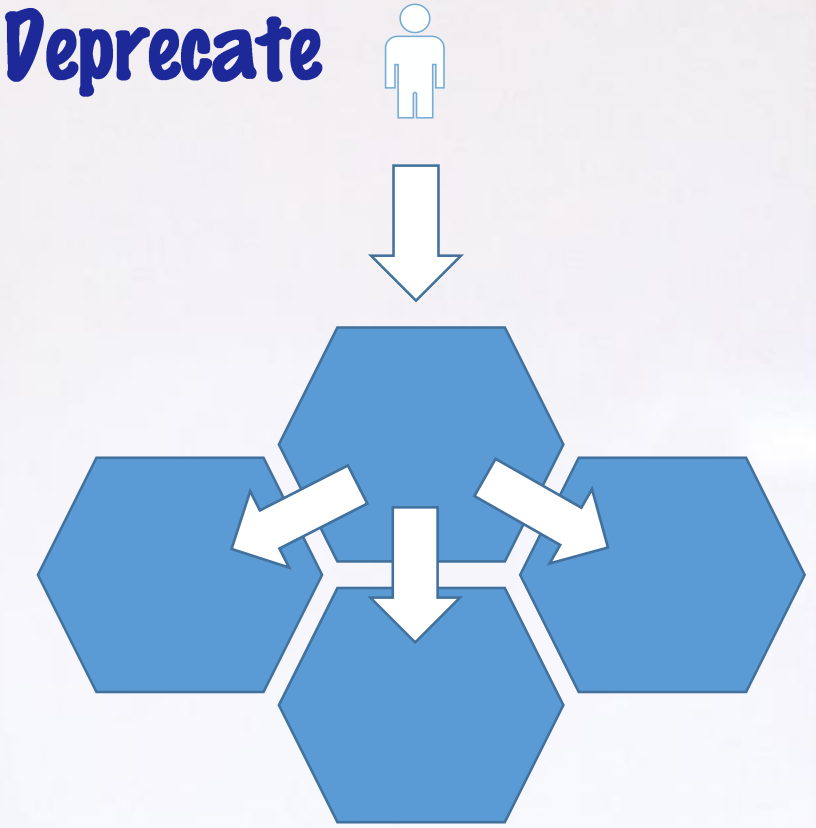
- Same Process
  - Gather Requirements (stories?)
  - Find business capabilities

Real World

Determining Services: Decomposition

What if I have larger services?

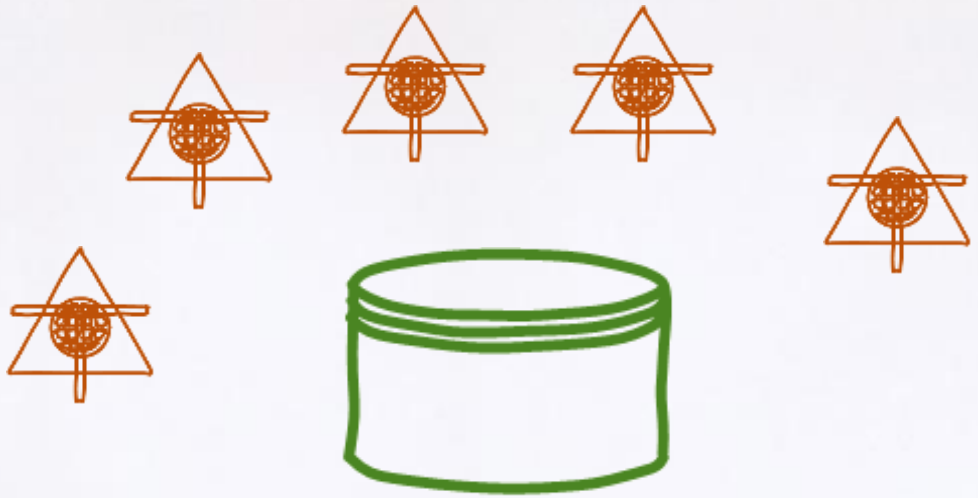
- Orchestrate and Deprecate



Real World

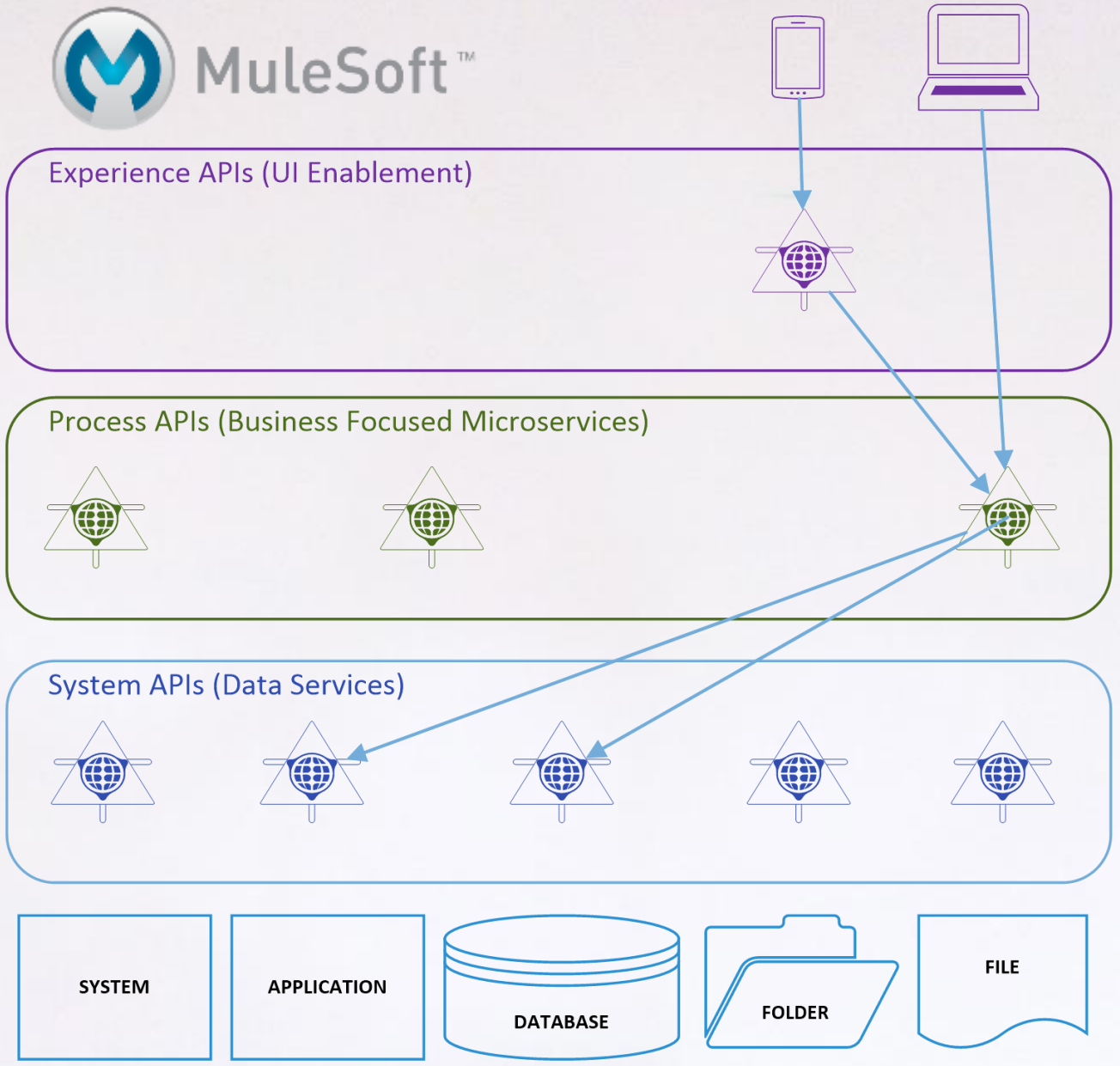
Data Focused Microservices

- Don't do this!!!



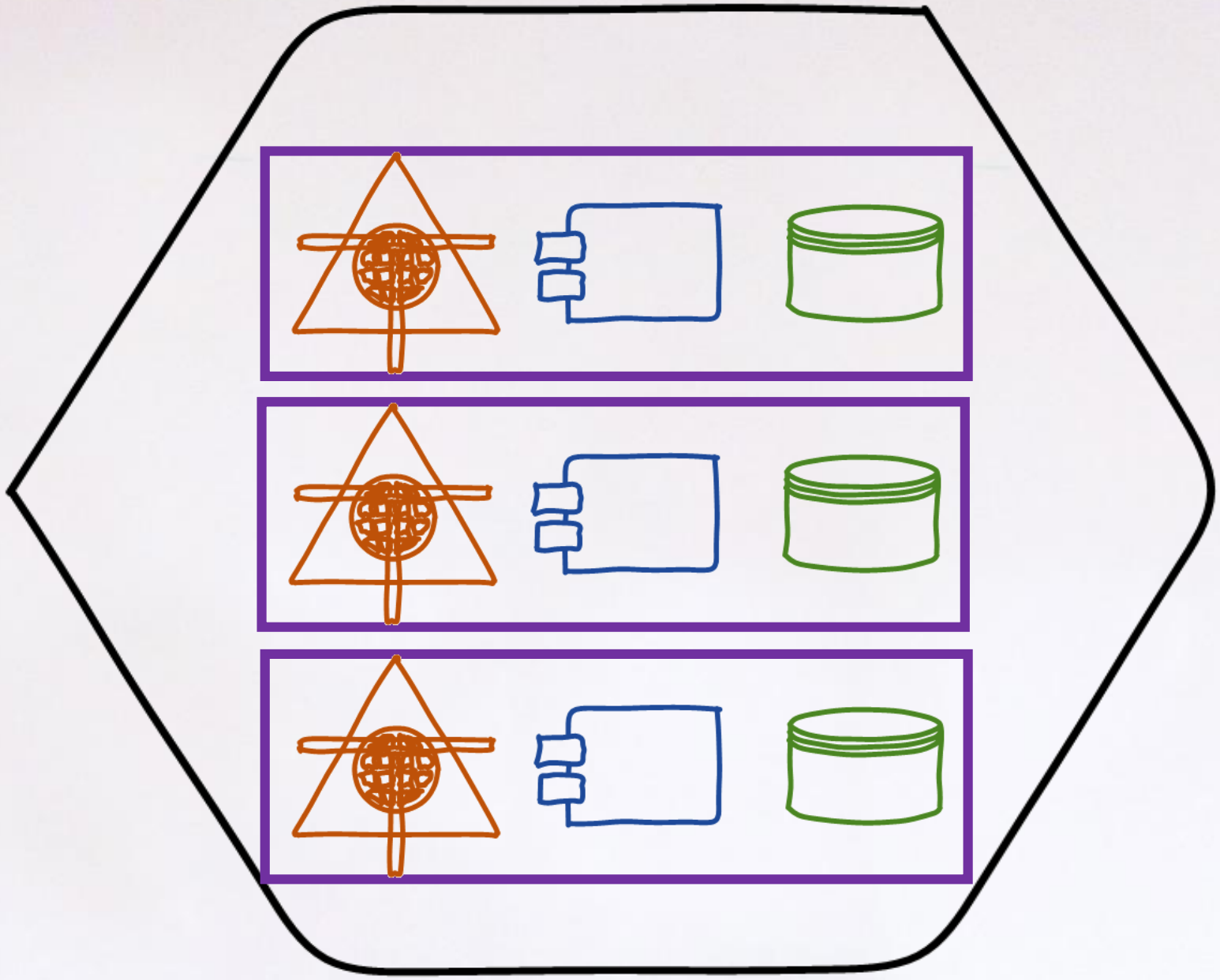
Real World

Data Focused Microservices



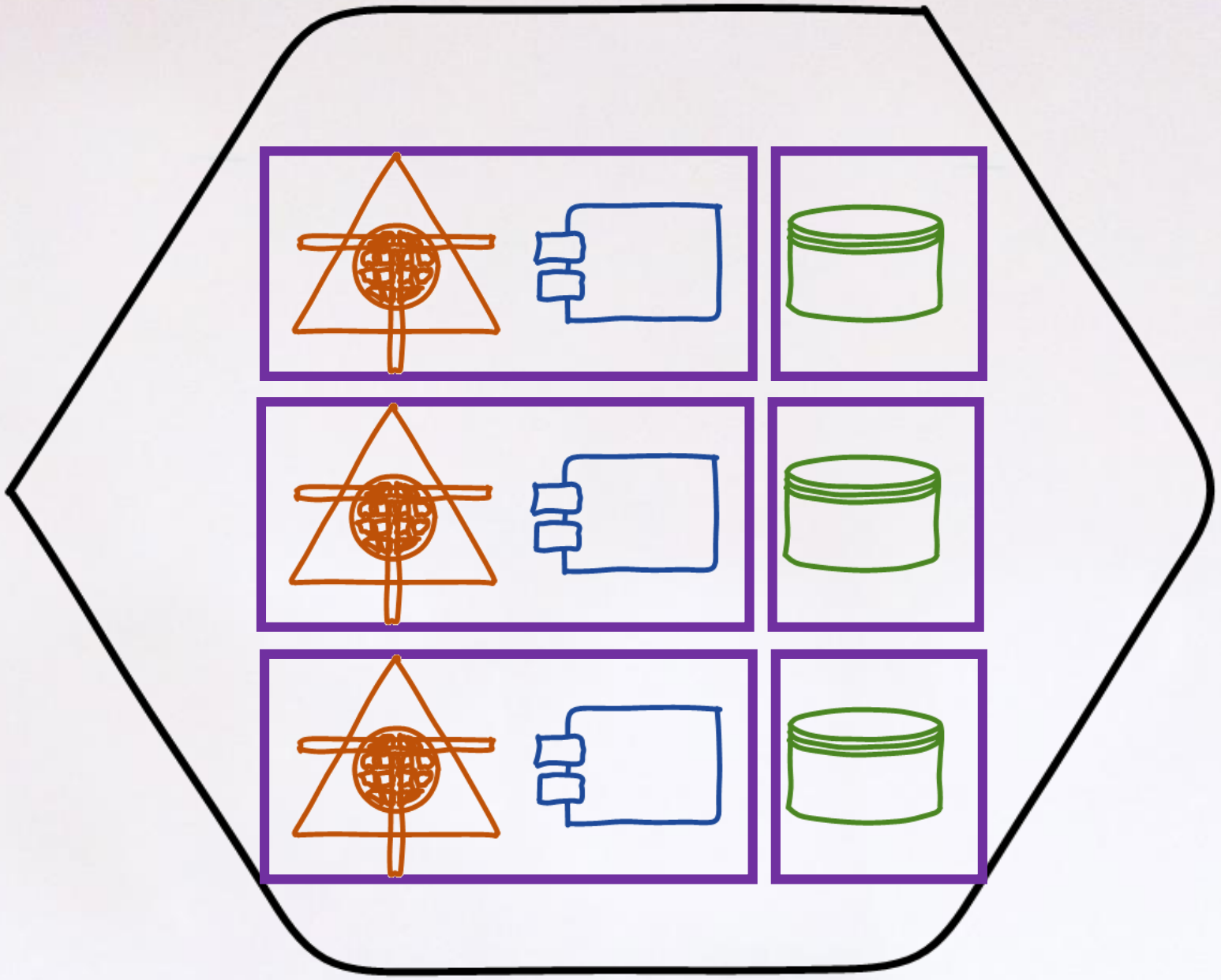
Real World

Scaling



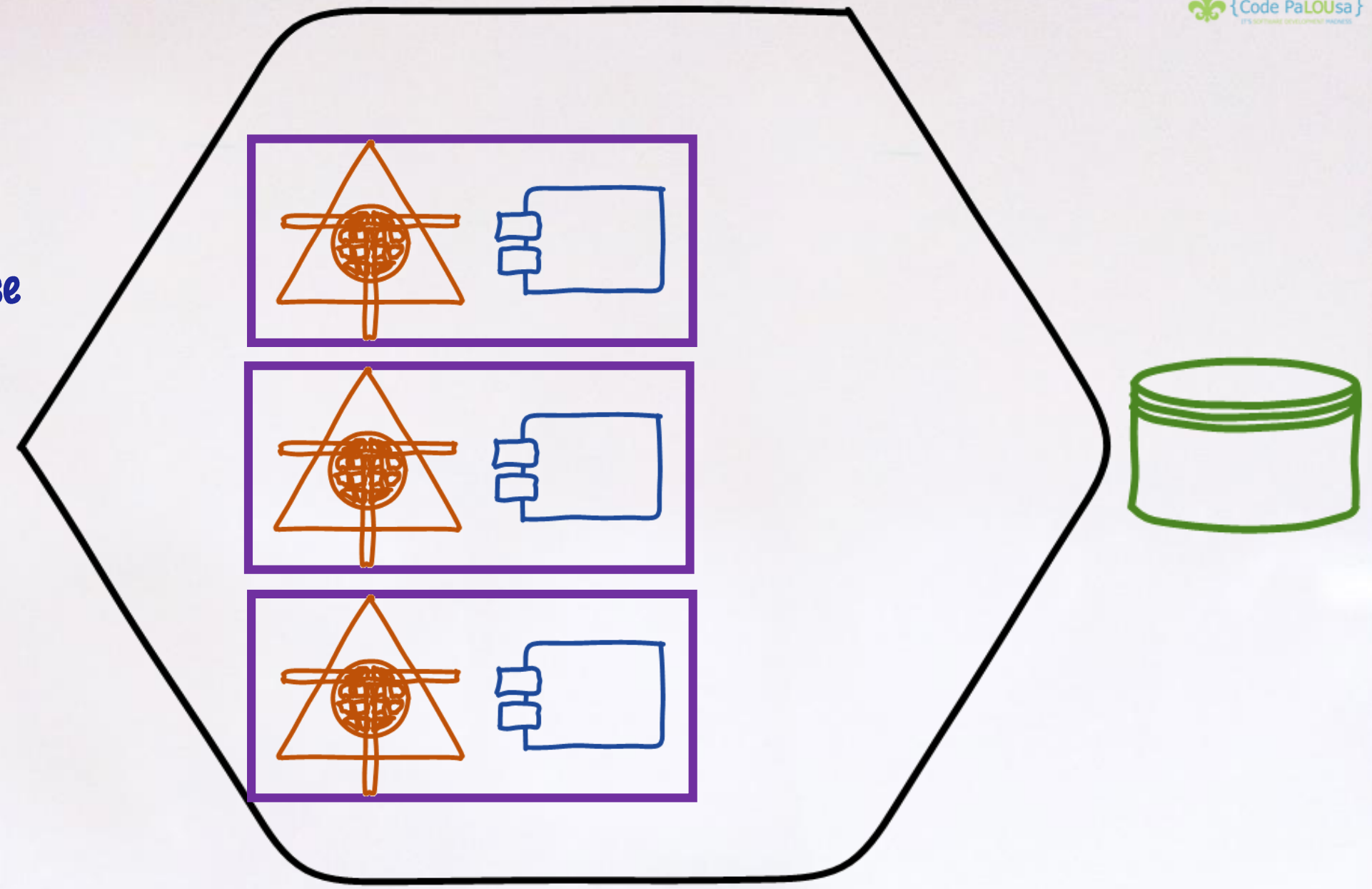
Real World

Scaling



Real World

Central Database



Real World

Central Database

Best Practices

- A single microservice owner for each table – if possible
- All access through the microservice

Real World

Other Notes

- In a purist environment, you will have to consolidate the data at some point
- Be prepared to introduce Master Data concepts
- How do you eat an elephant?
  - One bite at a time

Appendix

Slides that did not make the Code PaLOUsa cut

Architecture

Autonomy of Teams?


C#


SQL Server


Python


PHP


F#


Mongo DB


Oracle


Java


Couch

{Code PaLOUsa}

What do you call an organization with hundreds of microservices & every new technology known to man?

Bankrupt!