

Code Mash Edition



MICROSERVICES

Lessons from the Trenches

@gbworld





See Boss, The Money Spent Was Well Spent!





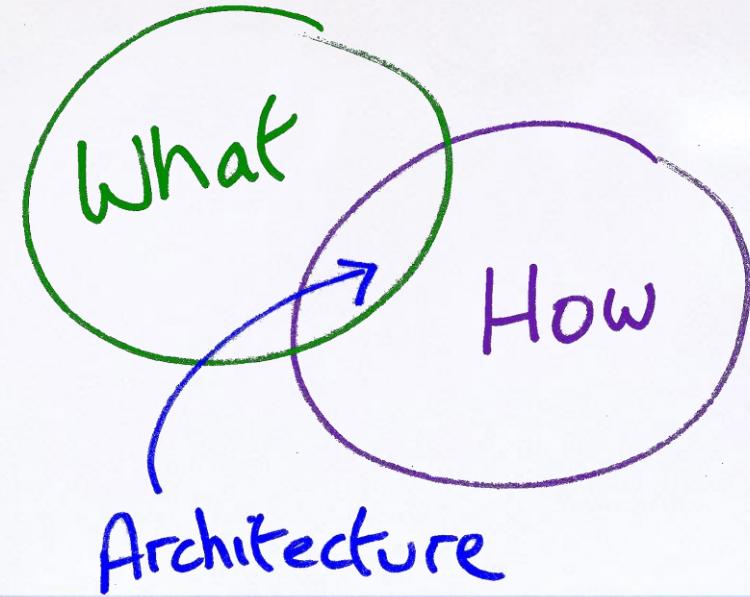
Who?

Gregory A. Beamer



- Senior Architect, UST Global
- Microservices Practice
- Senior Architect
 - Modernization
 - Integration
 - Service Enablement
 - Microservices
- Other
 - Agile
 - DevOps

150.36 Sm Samarium	62 Ar Argon	39.948 Te Tellurium	127.60 As Arsenic	52 S Sulfur
--------------------------	-------------------	---------------------------	-------------------------	-------------------





Opening Remarks

- Insert Greg's Random B*llsh*t Here
- Possibilities
- Video Course
 - Architectural Styles
 - Other B*llsh*t



POP QUIZ TODAY!

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

Boundaries and deployment methodologies



POP QUIZ TODAY!

2. Why would you desire outer complexity over inner?

Easier to Commoditize



POP QUIZ TODAY!

3. Using buzzwords only, name some practices, methodologies, etc. that complement microservices:

Agile, DevOps, SOLID principles, monitoring, API management



POP QUIZ TODAY!

4. What benefit of microservices has no corresponding negative?

Trick Question – There are NONE!!!

What?
Silver Bullet
or Hot Air?



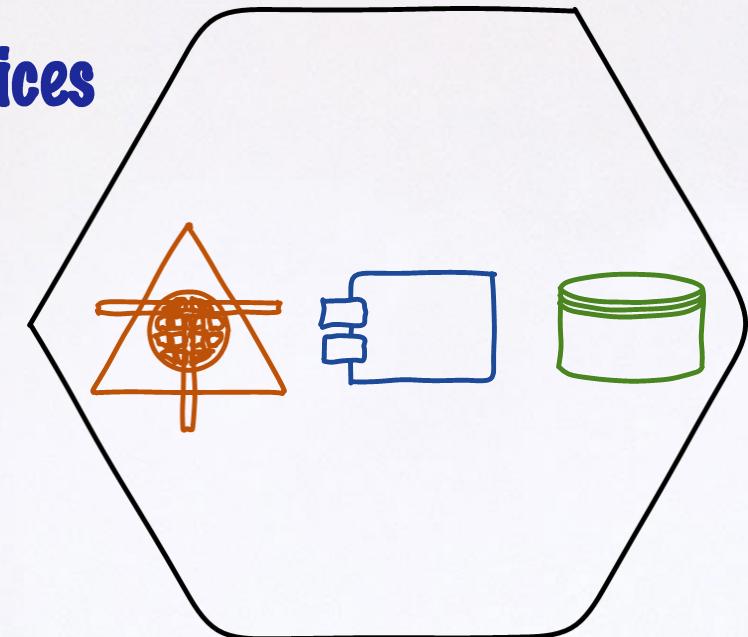
What?
Silver Bullet
or Hot Air?





Agenda

- What are Microservices?
A Brief Introduction
- Why use Microservices?
Benefits and breaking some myths about microservices
- How do we do this?
Prepping for & Implementing Microservices
- Here in the Real World
Inspired by Real Events





What are Microservices?

A Brief Introduction

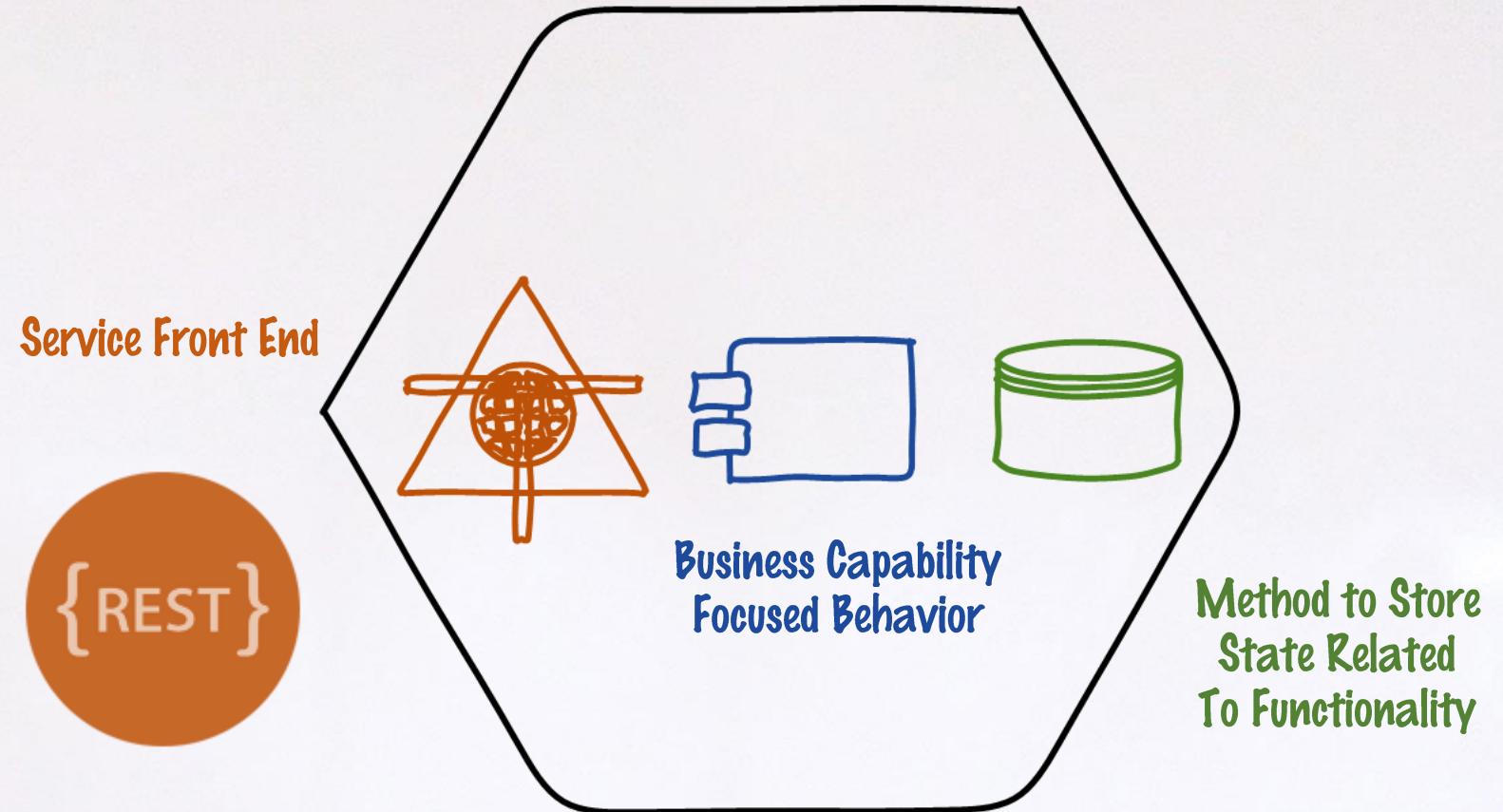


What?

In a Nutshell

PURIST VIEW

Complete Stack in a Single Product



What? Architectural Style

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

Architectural Style

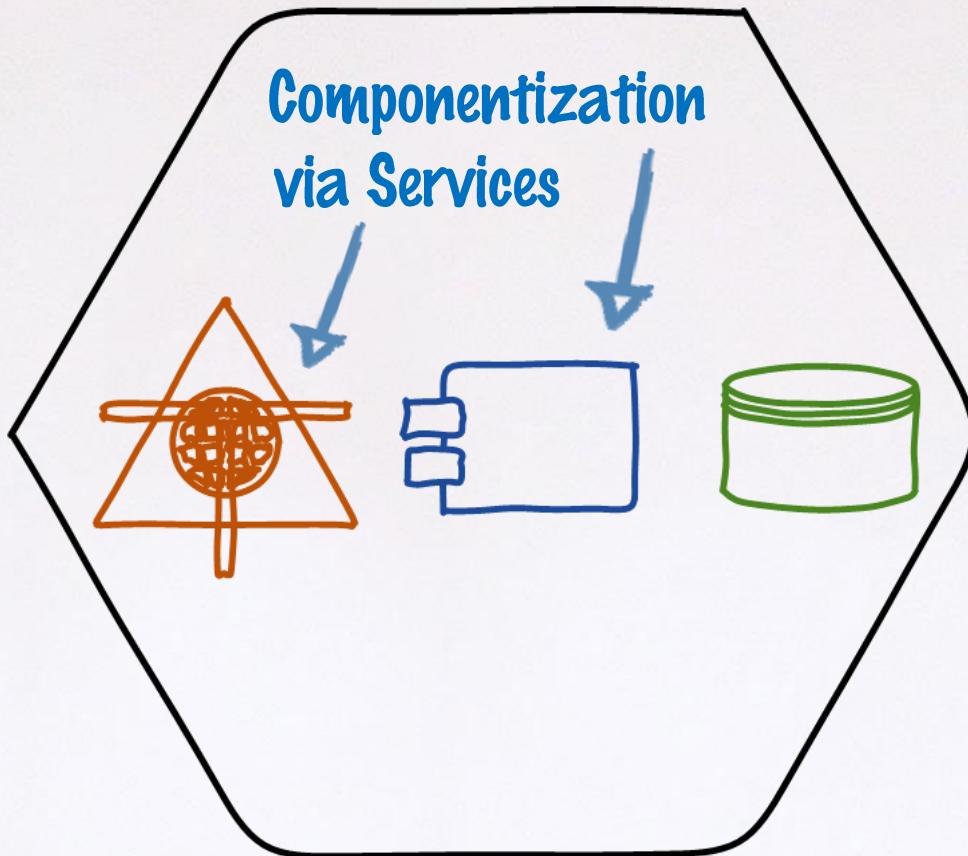
In short, the microservice **architectural style** is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API





What? Characteristics

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





Sidebar 1

What is a Component?

Definition
(contract)

Implementation

Implementation



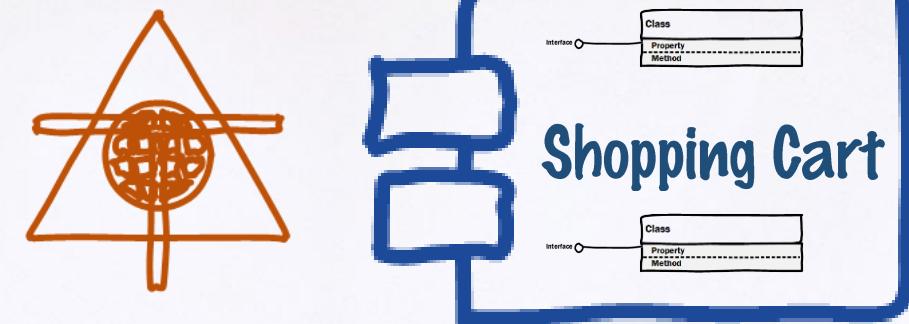
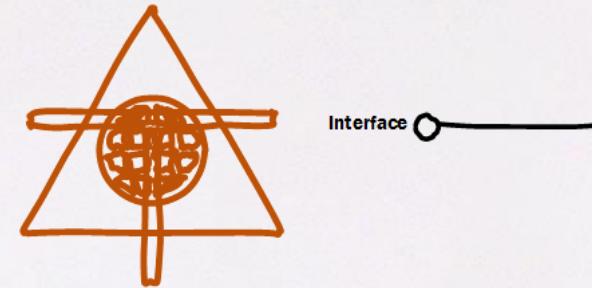
Sidebar 1

What is a Component?



Contract

Implementation

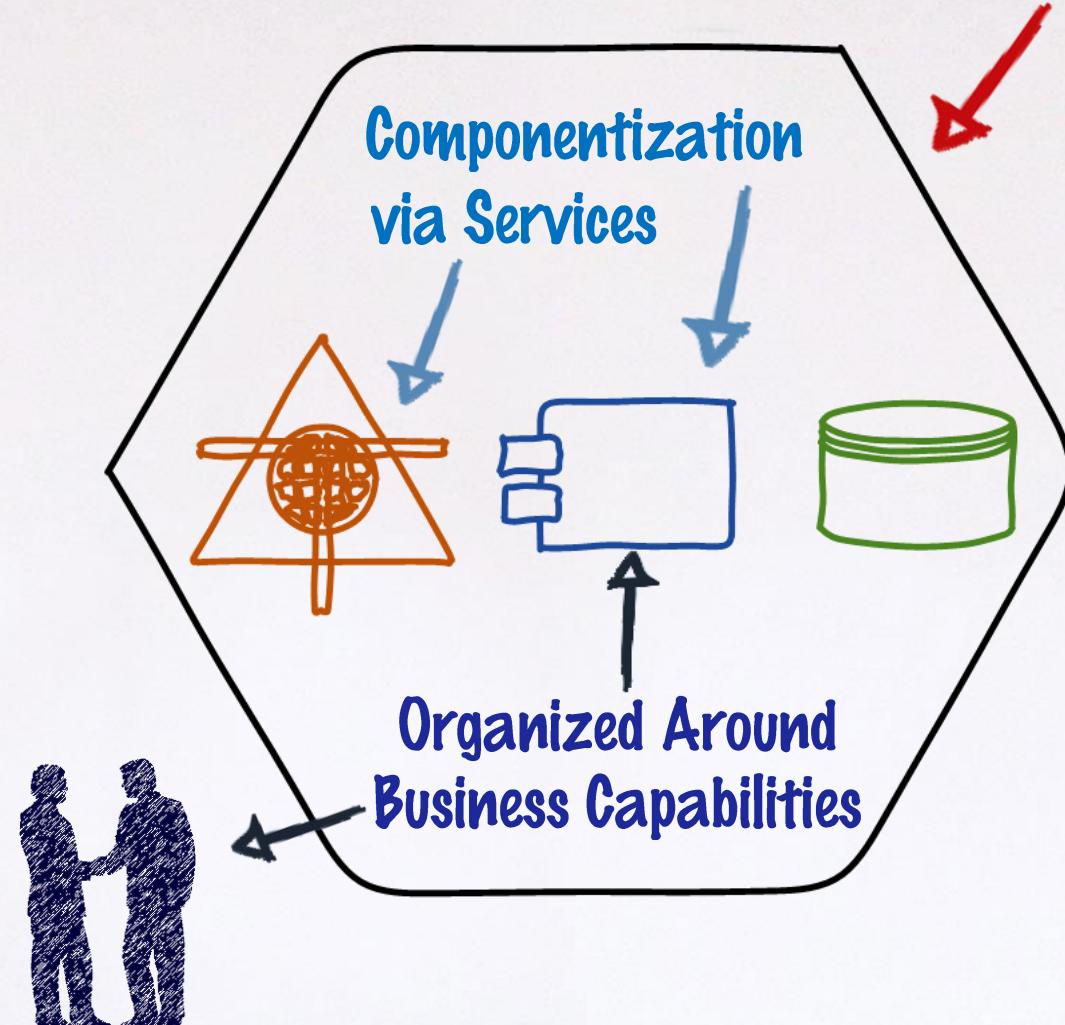




Products, not
Projects

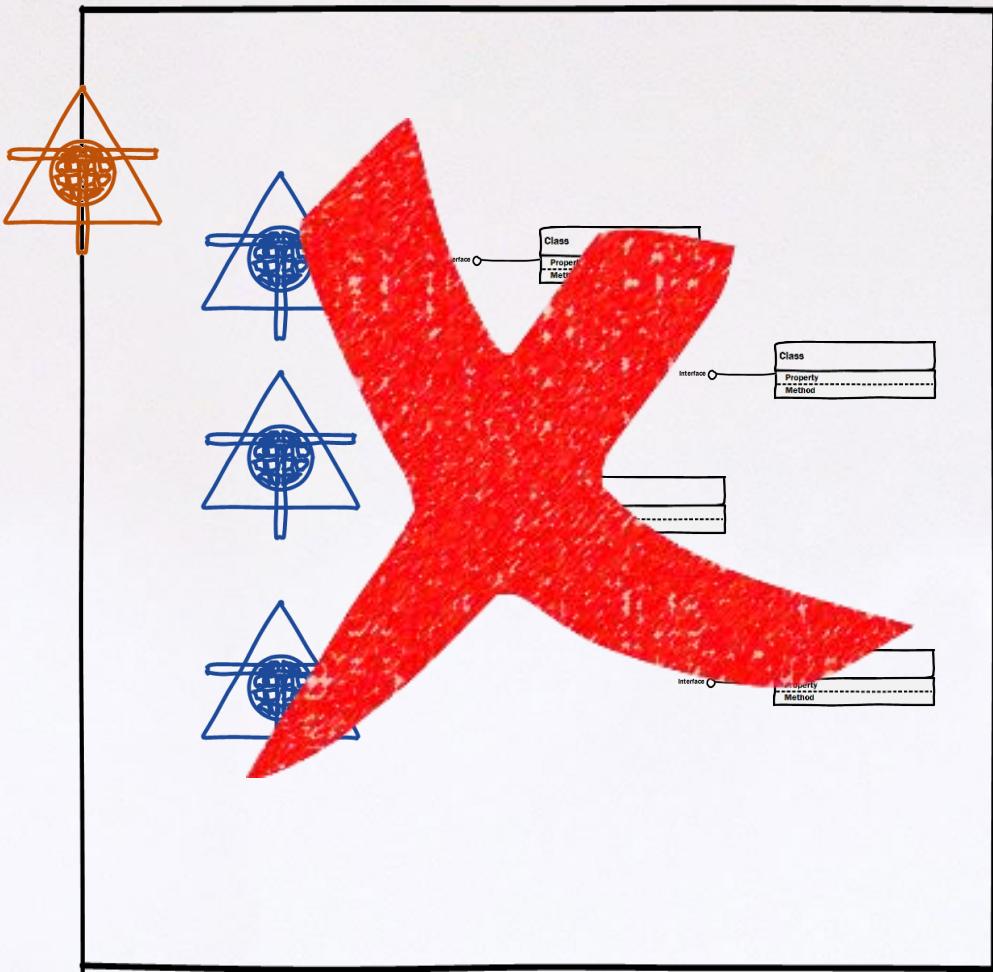
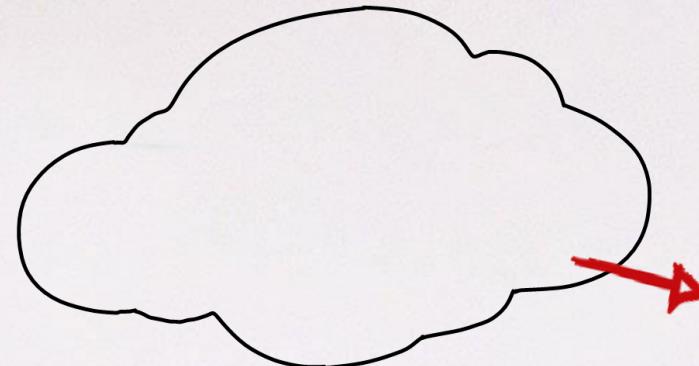
What? Characteristics

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



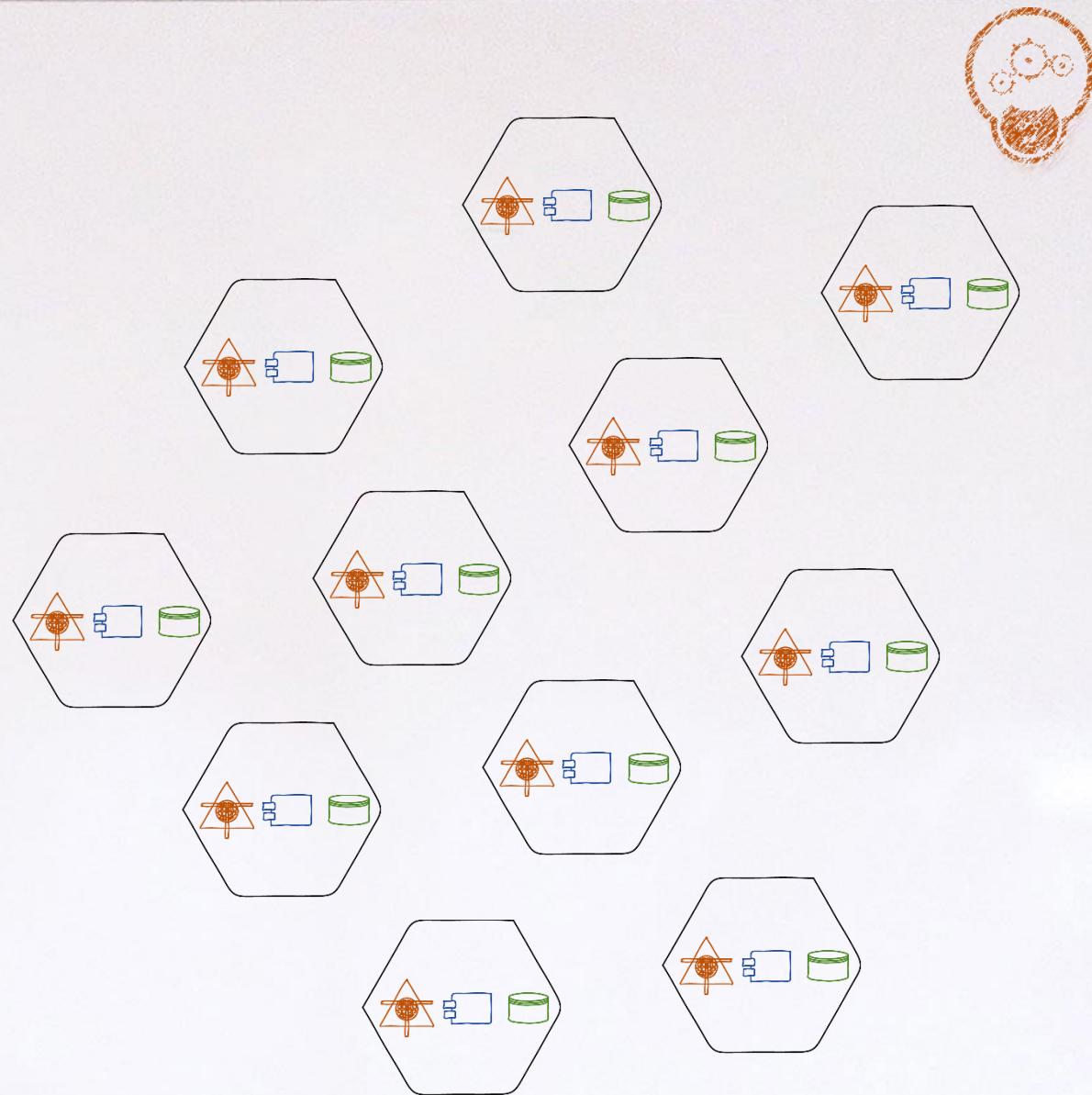
Sidebar 2

What is a Product?



Sidebar 2

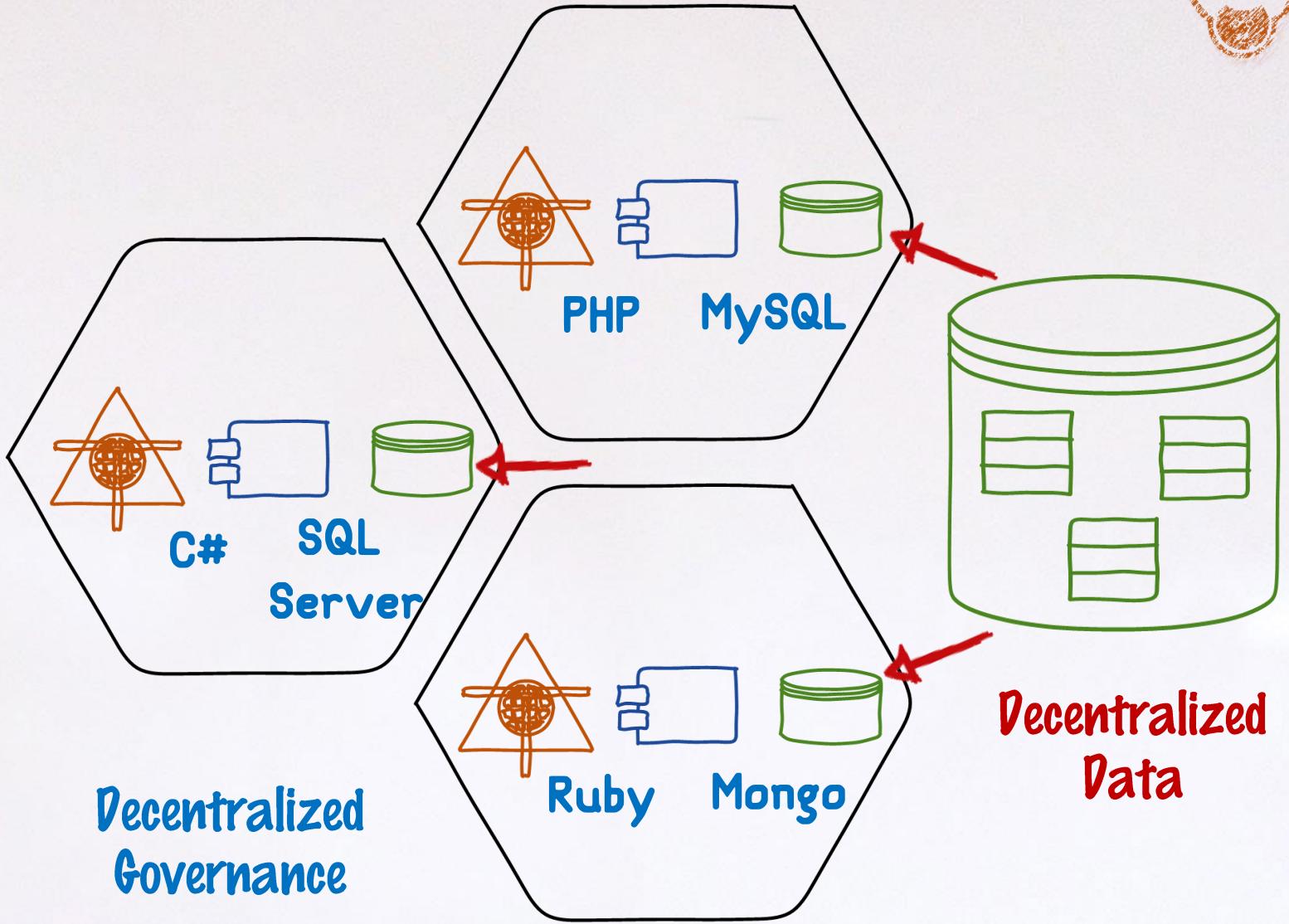
What is a Product?





What? Characteristics

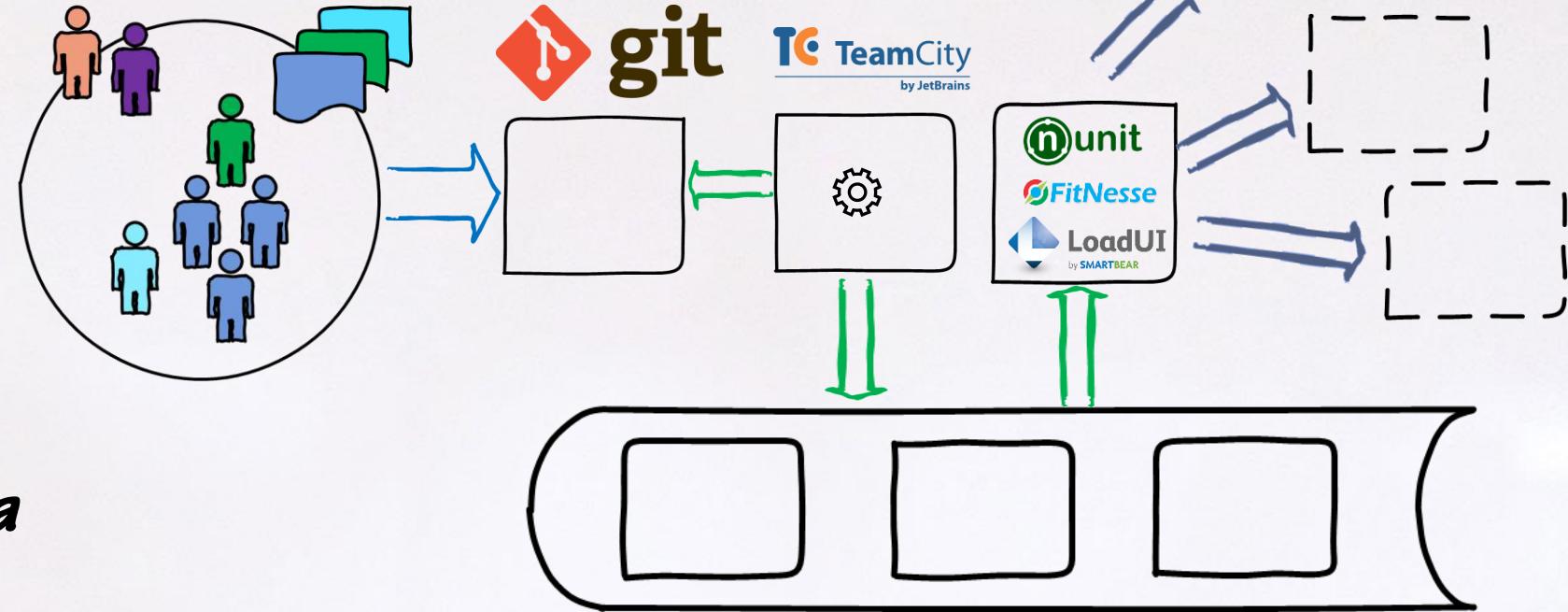
From "Characteristics of a
Microservice"-
Martin Fowler
<http://goo.gl/AZwucZ>



What? Characteristics

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

Infrastructure Automation



puppet

CHEF
CODE CAN

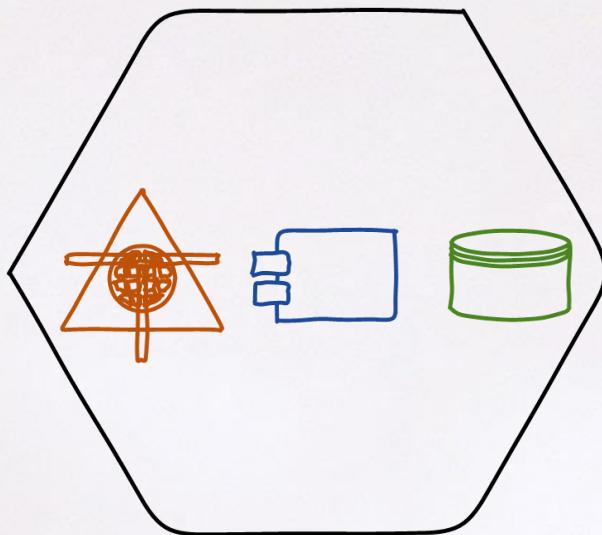
Bamboo



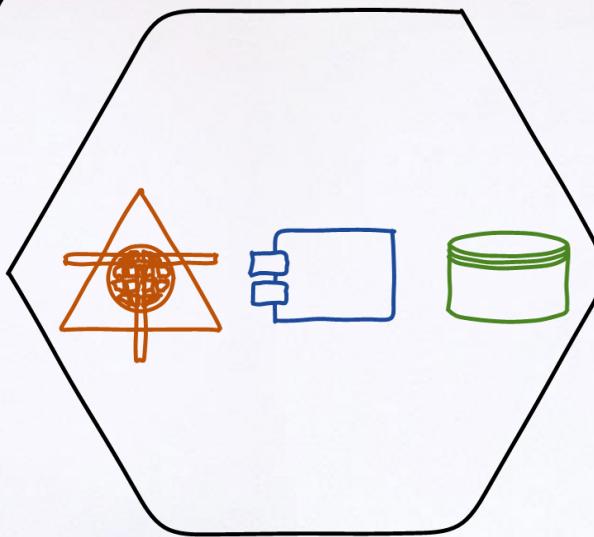
What? Characteristics

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

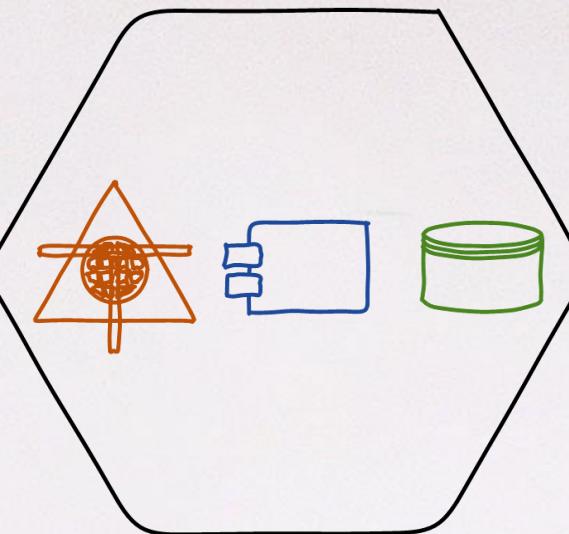
Smart Endpoints
Dumb Pipes



apigee



RabbitMQ™

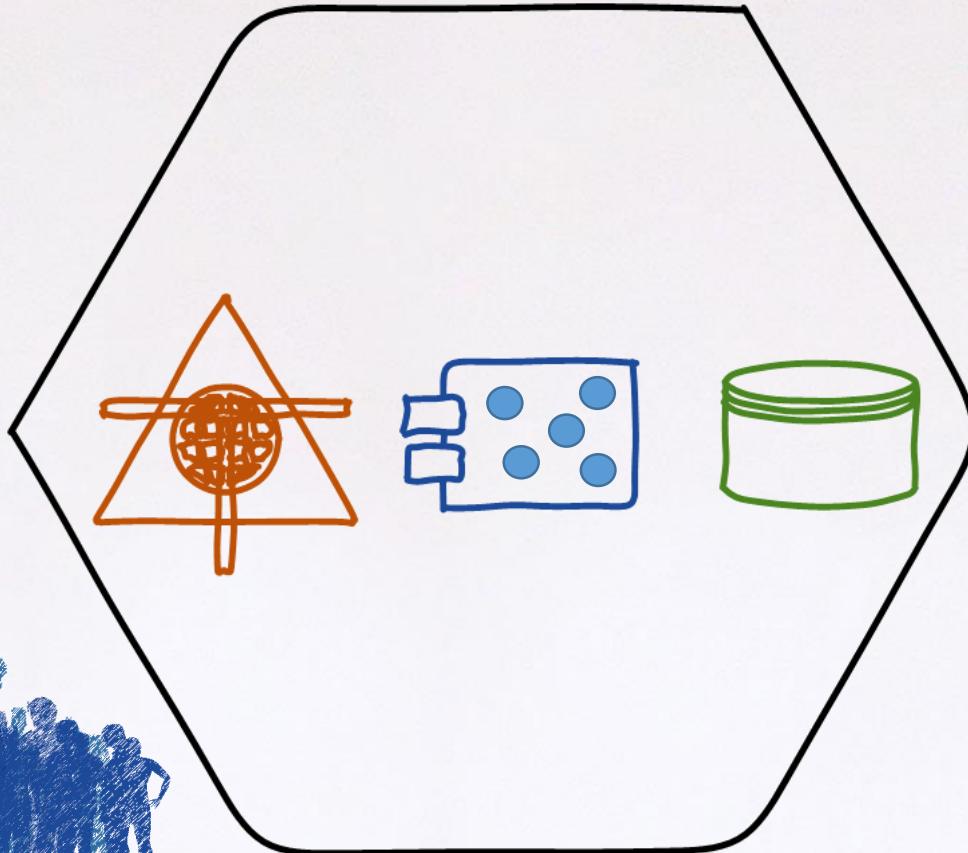




Evolutionary Design

What? Characteristics

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

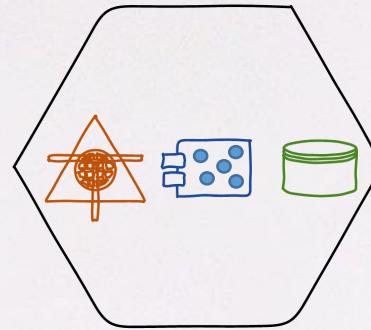




What? Characteristics

Microservices

- Componentization via Services
- Organized around business capabilities



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

Agile Practices

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

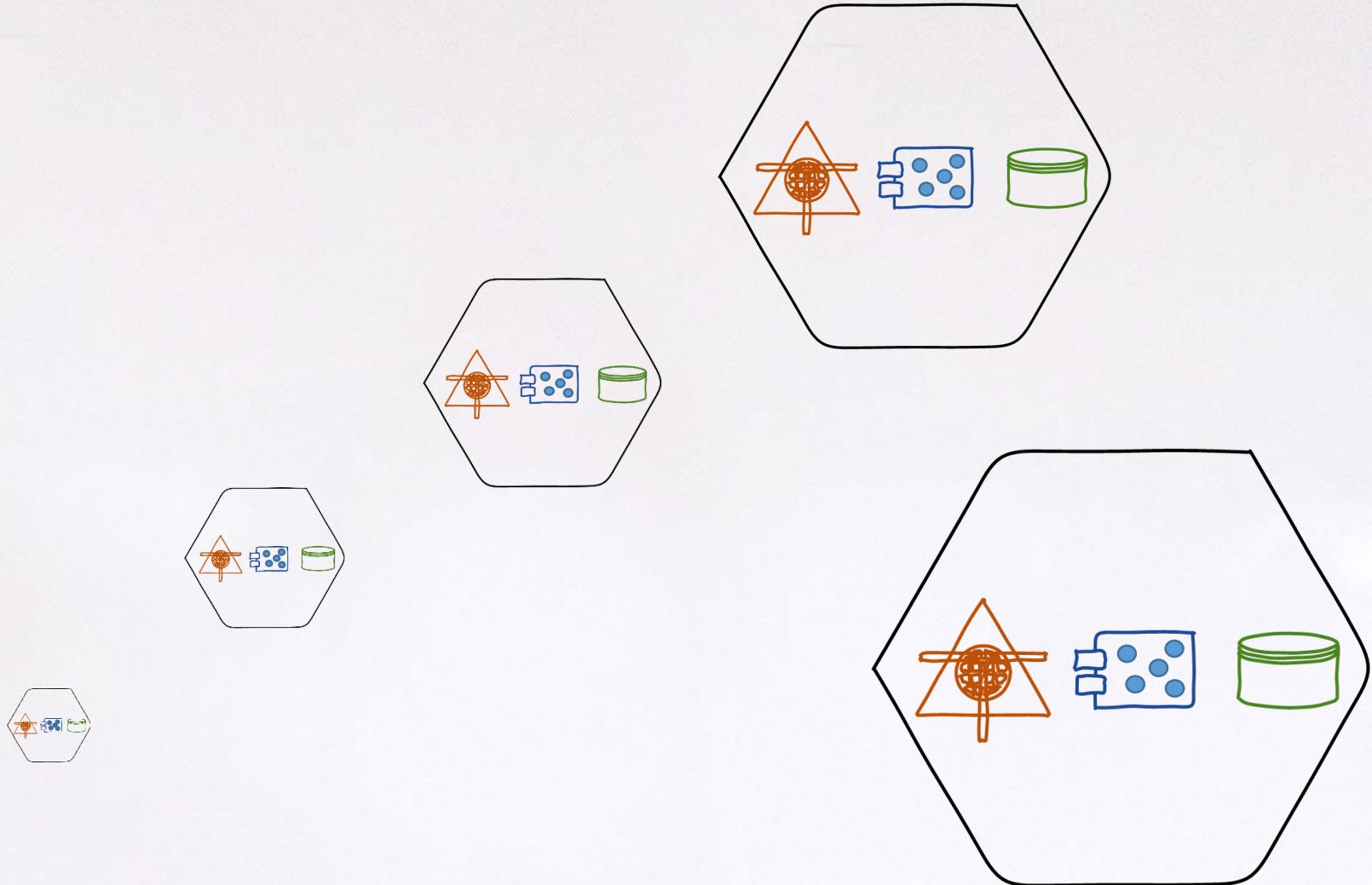
DevOps

- 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



Why Use Microservices?

Benefits and breaking some myths about microservices



Why?

The Benefits of Microservices

- Better than Monoliths
- Easy to
 - Understand
 - Enhance
 - Test
 - Deploy
- Resilience
 - Low impact on other services
- Autonomy of Teams
 - Decentralized Governance



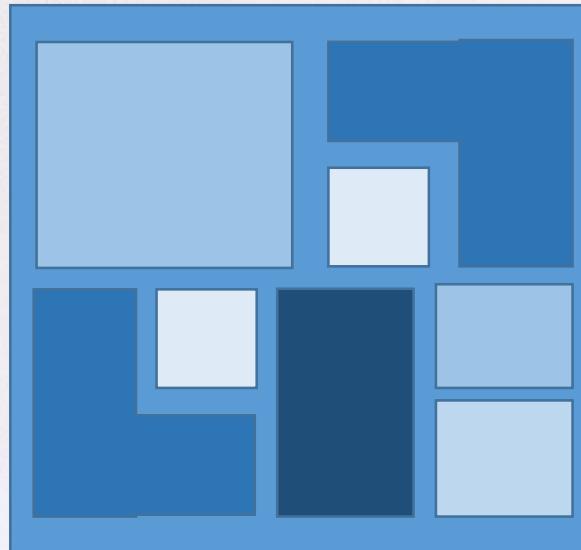
Independent Scale

Flexibility

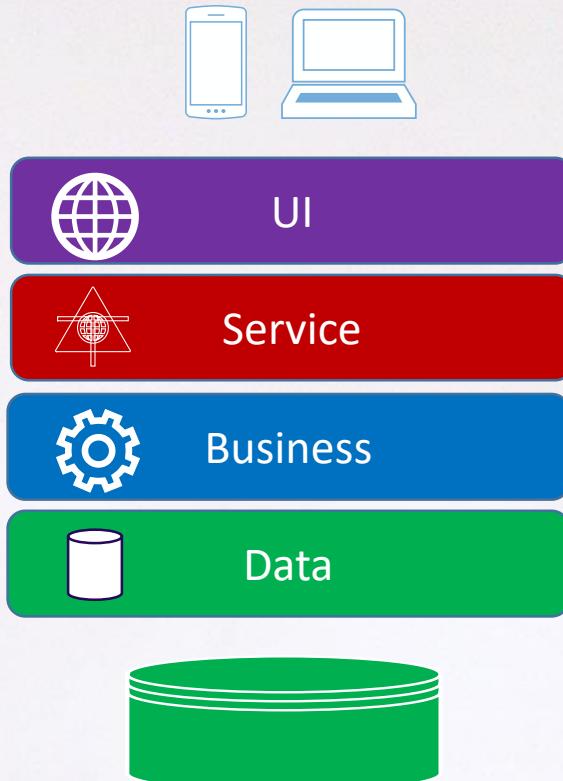
Loose Coupling
Tight Cohesion

Why?

Monolith versus Microservices

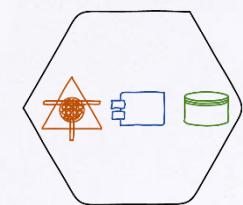
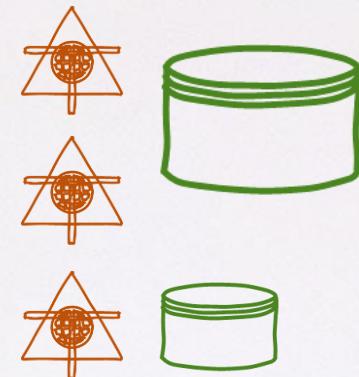


Monolithic Architecture



Layered Architecture

SOA



Microservice
Architecture



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

Boundaries and deployment methodologies



Why?

Microservices are Easy

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



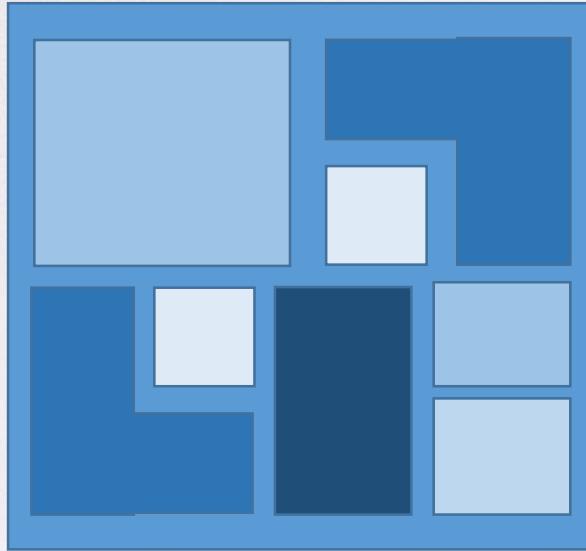
Why? Microservices are Easy

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



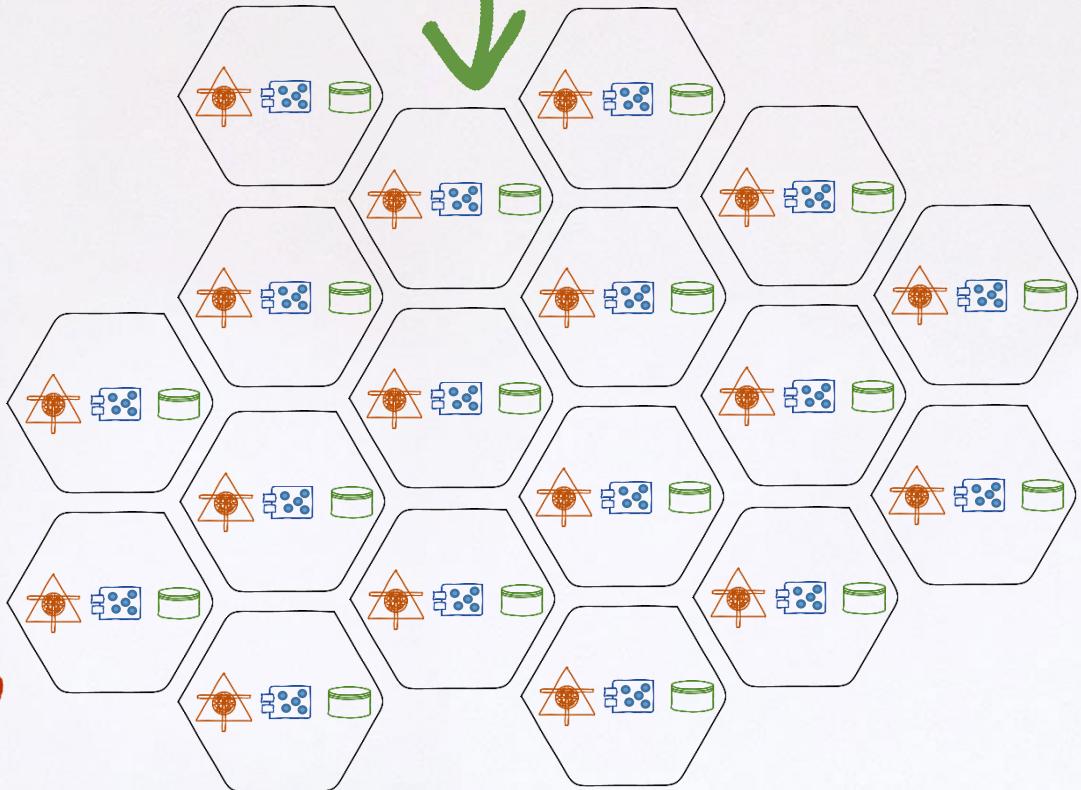
Why?

Microservices are Easy



This is not easy

This is easy

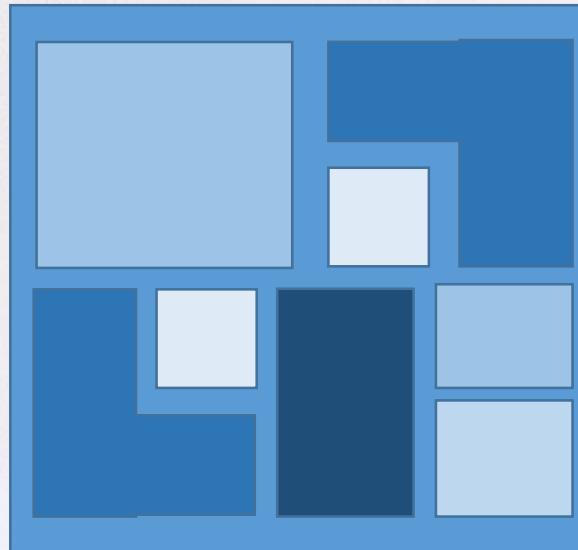


Inner
Complexity

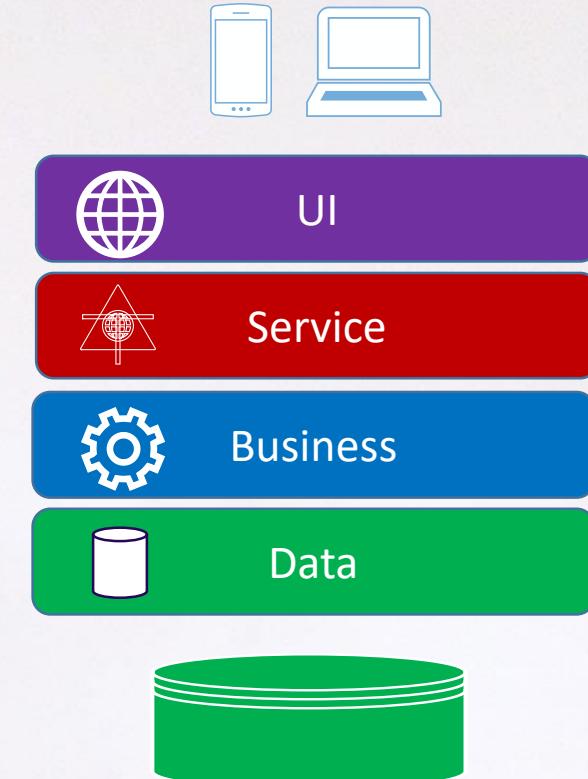
Outer
Complexity



Sidebar
Complexity

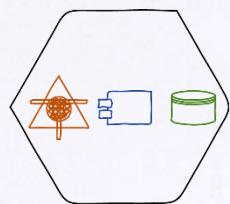


Monolithic Architecture



Layered Architecture

SOA

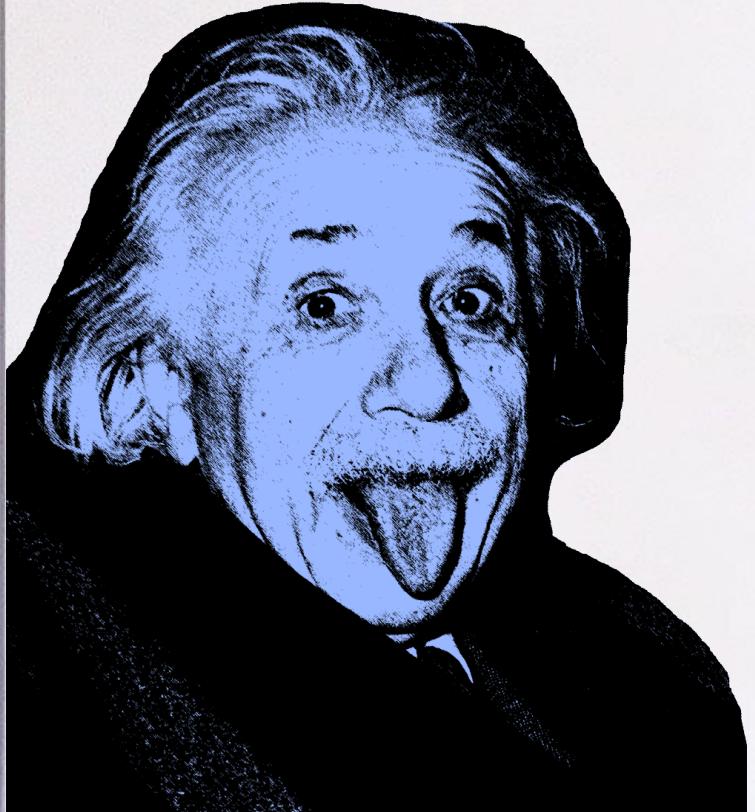


Microservice
Architecture



Why?

Microservices are Easy



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

– Albert Einstein



Why?

Microservices are Easy



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

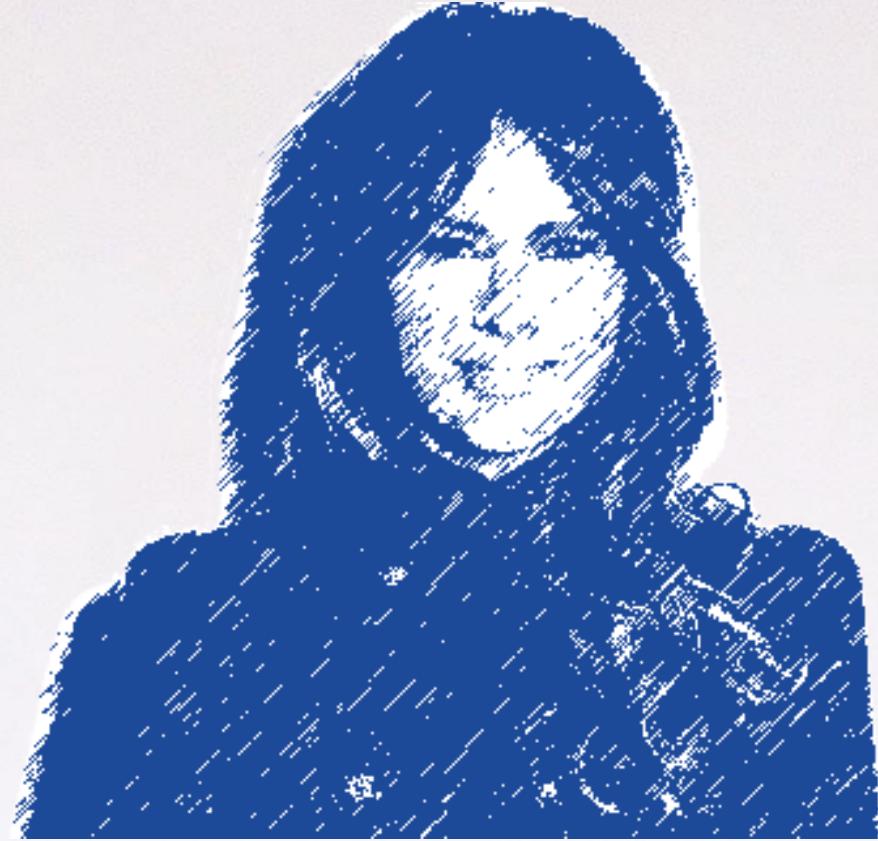
- Gregory A. Beamer



Why?

Microservices are Easy

*Feeling like Melania
Trump at a plagiarism
conference*



"I did not copy off Michelle's Paper!"
- Melania Trump



Why would you prefer outer complexity over inner complexity?

Because you can commoditize it!!!

apigee

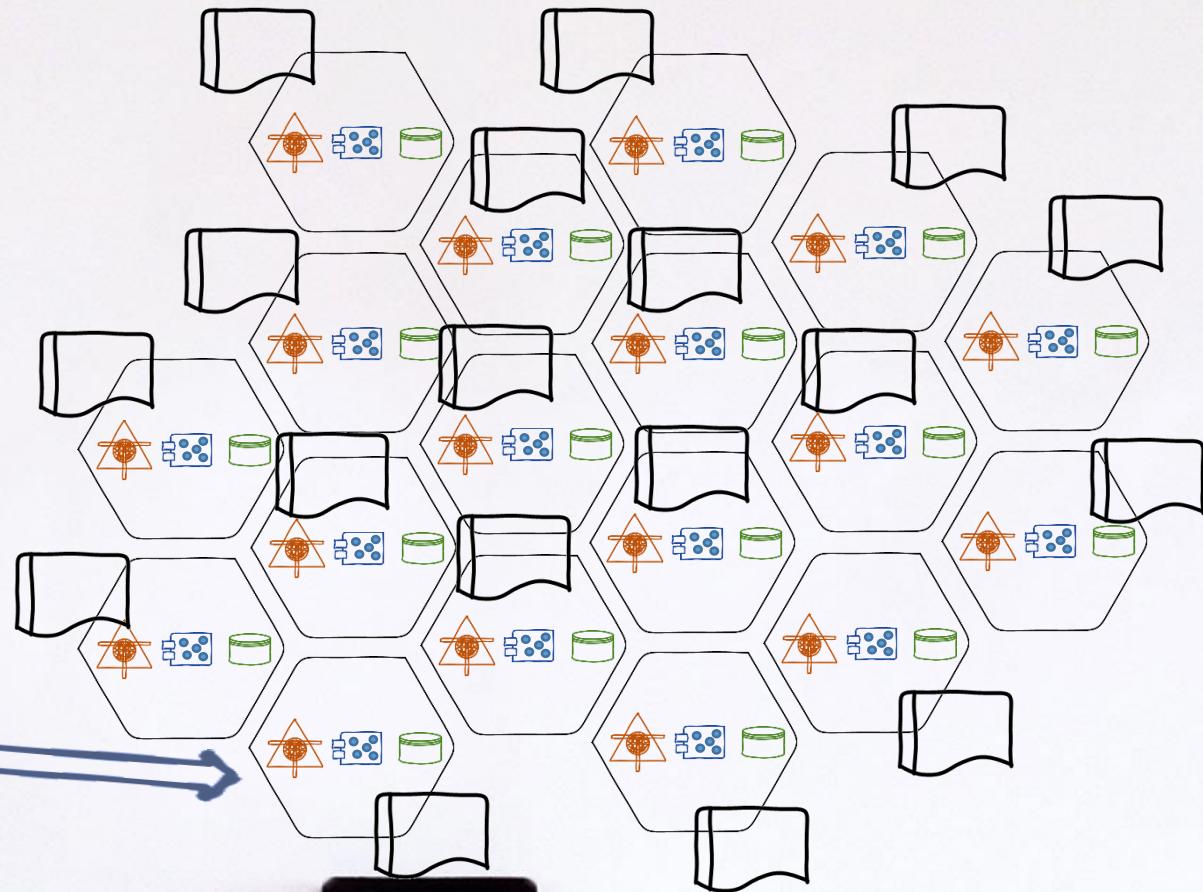
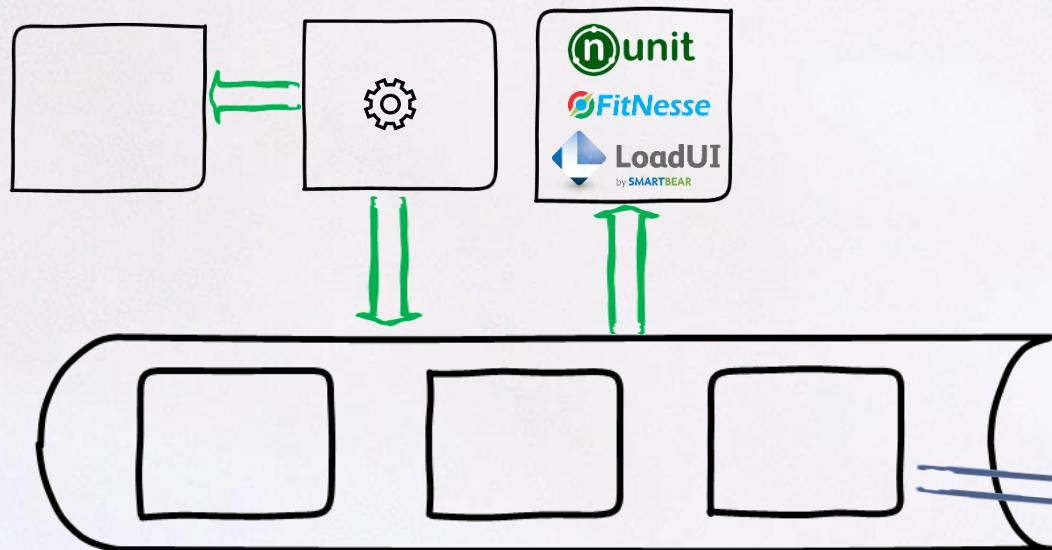
SOA software™





Why?

Microservices are Easy to Deploy





Why spend all that time automating?

Because your ops people
have better things to do

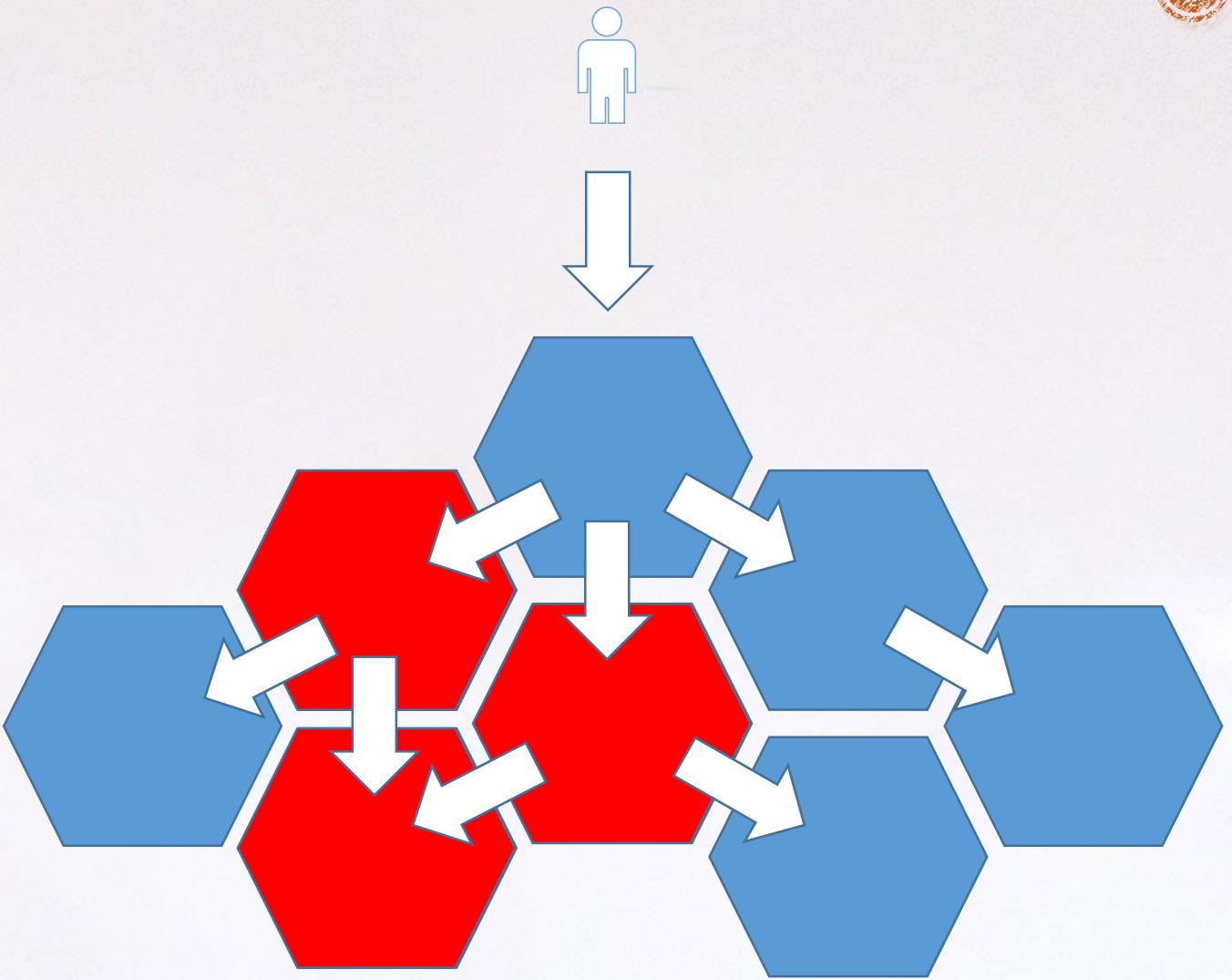
Because you can
commoditize it!!!





Why? Resilience

Low Impact on Other Services





How do you insure low impact on other services?

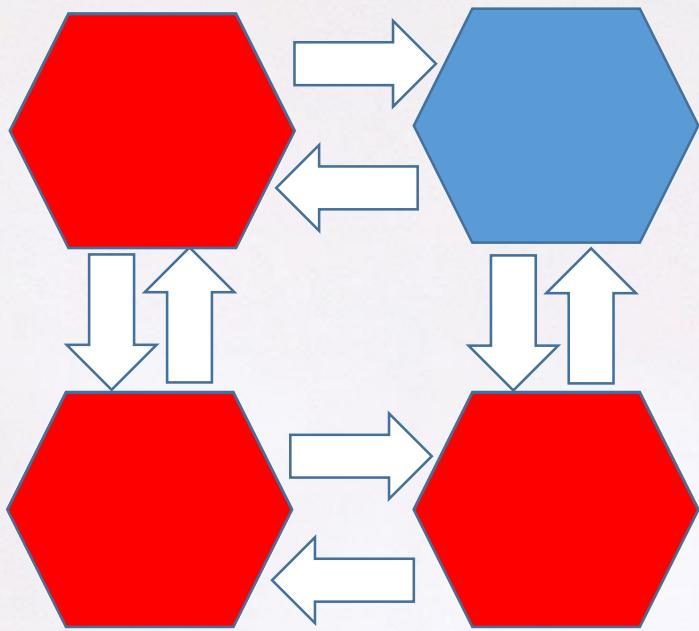
Because you can
commoditize
... and WORLD PEACE!



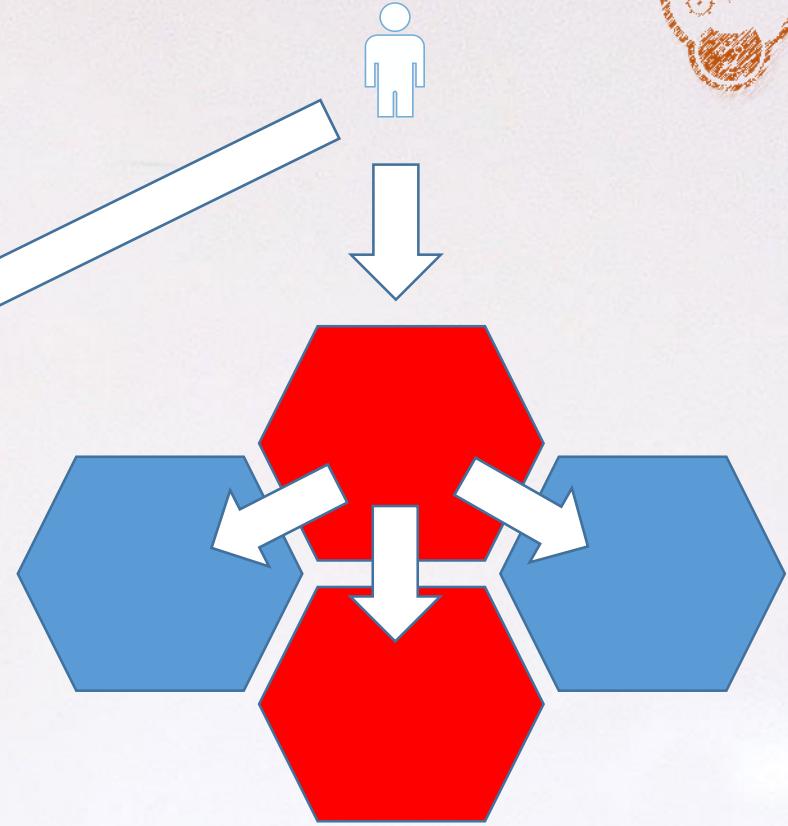
kubernetes
by Google



Why? Resilience



Choreography



Orchestration



**Which is a better pattern:
orchestration or choreography?**

Which is better: A hammer or a screwdriver?

Hammers SUCK!!!

Why?

Autonomy of Teams



C#



SQL Server



Python



PHP



Mongo DB



F#



Java



Oracle



Couch



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

Bankrupt!



How Do I Do This?

Implementing Microservices

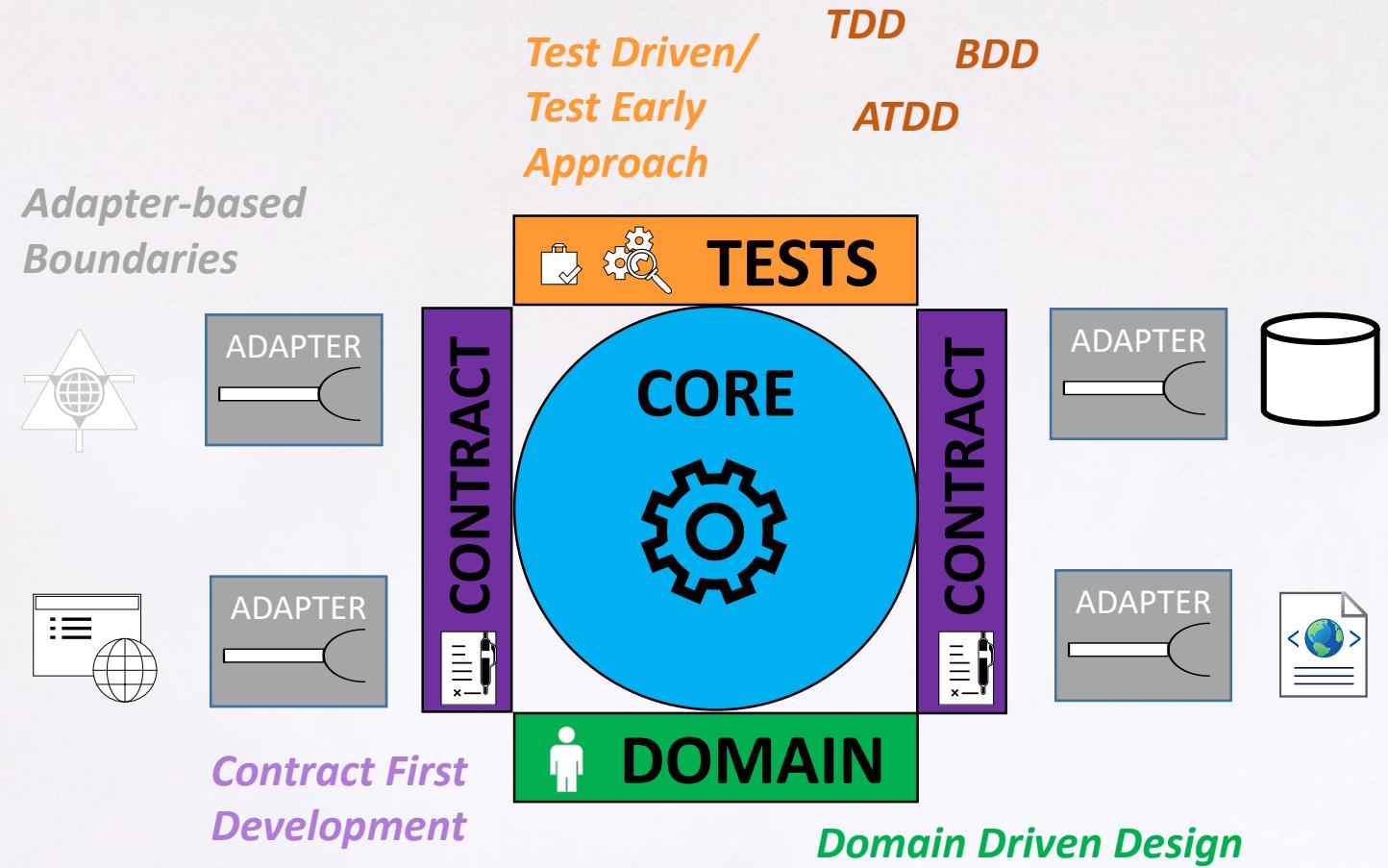
CASA Development Model

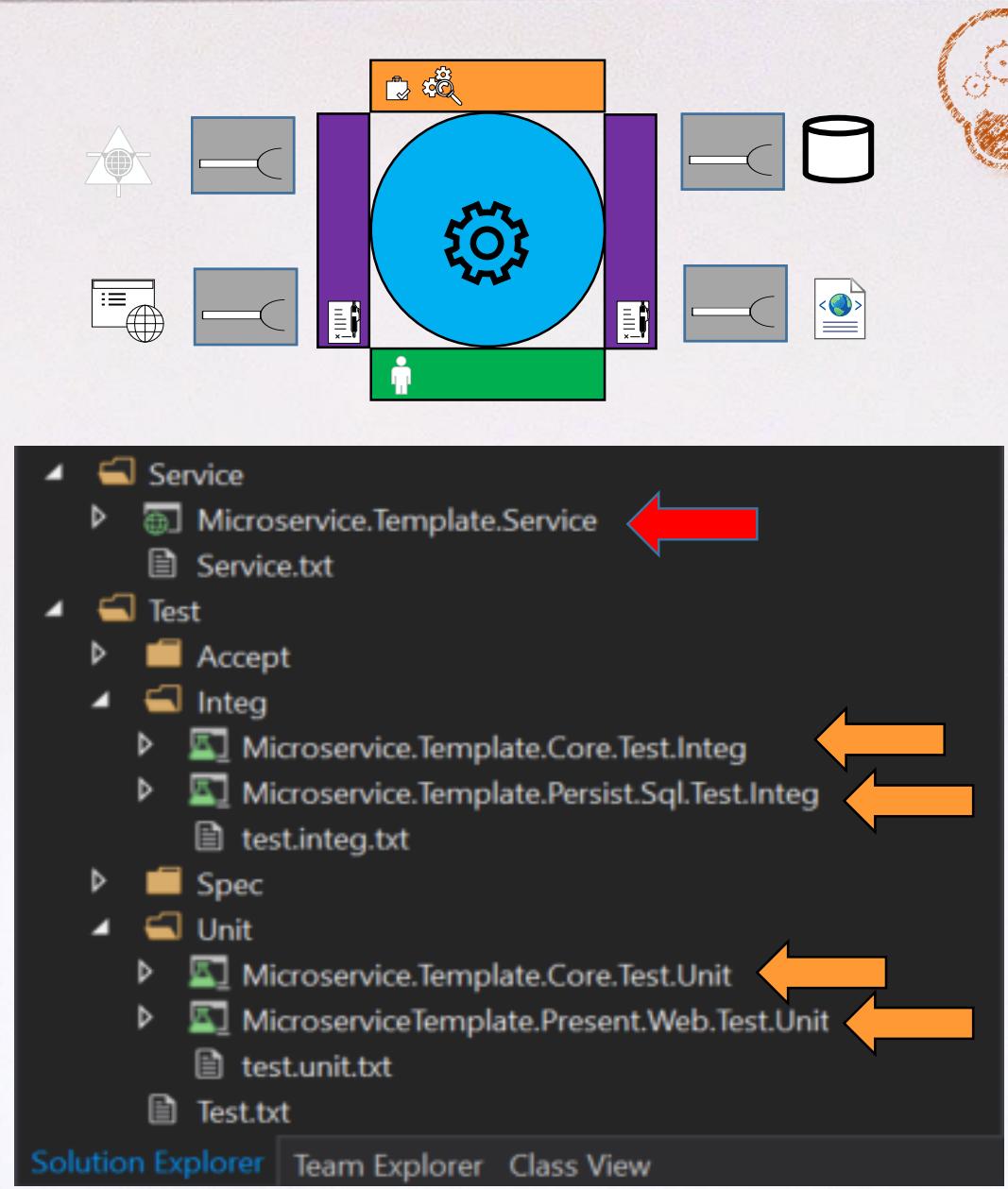
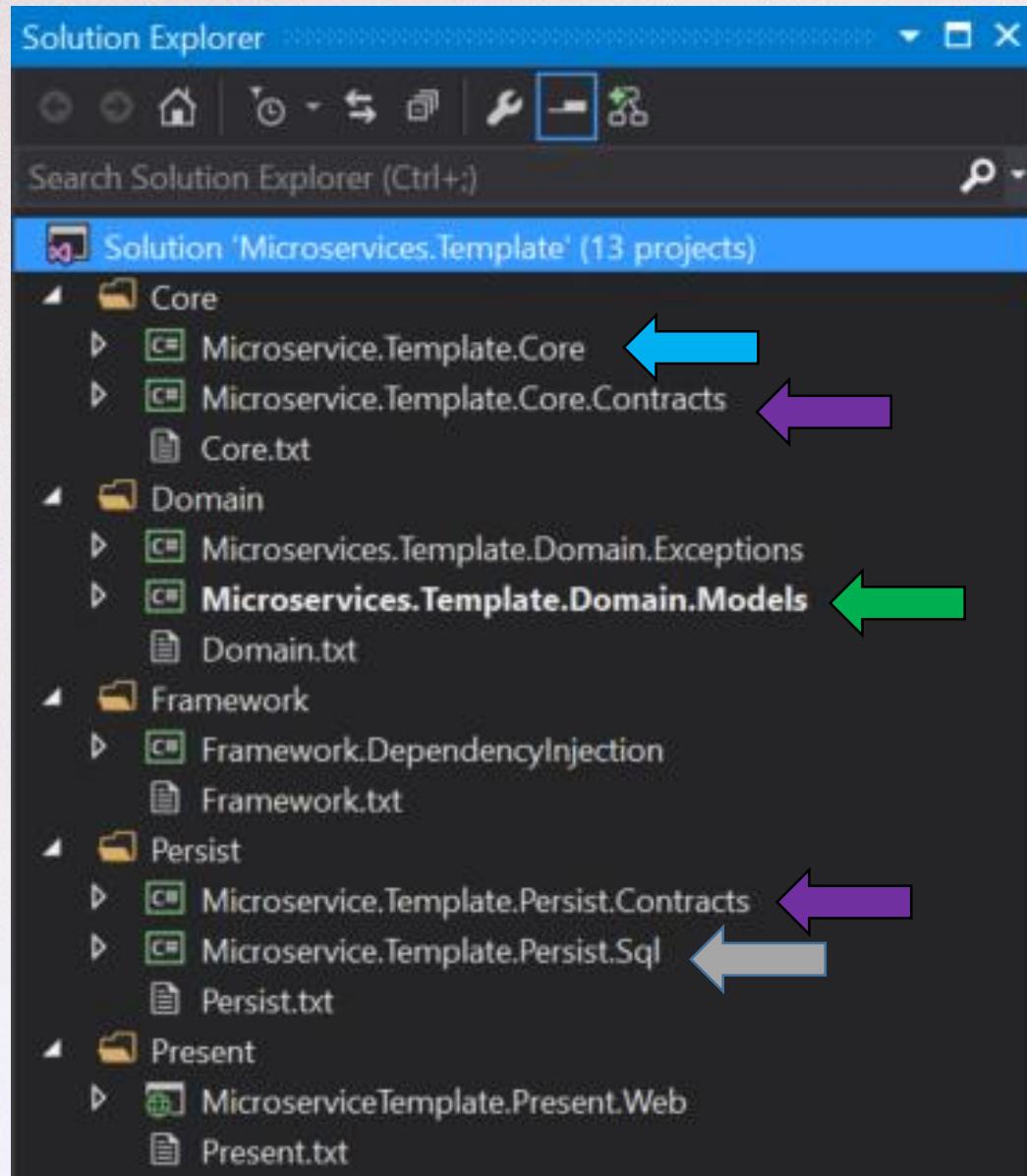
Core AS Application



How?

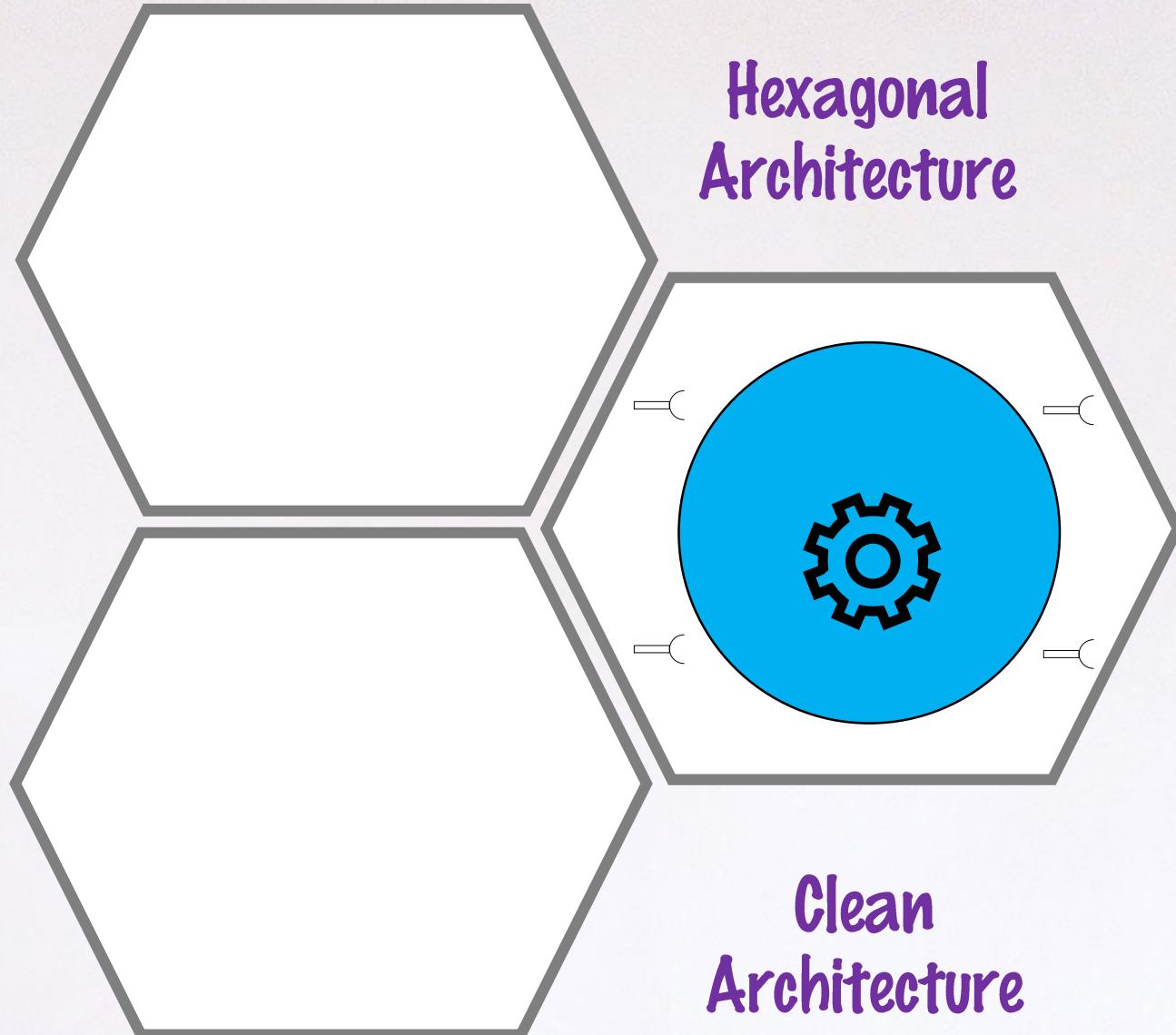
Organize Your Code



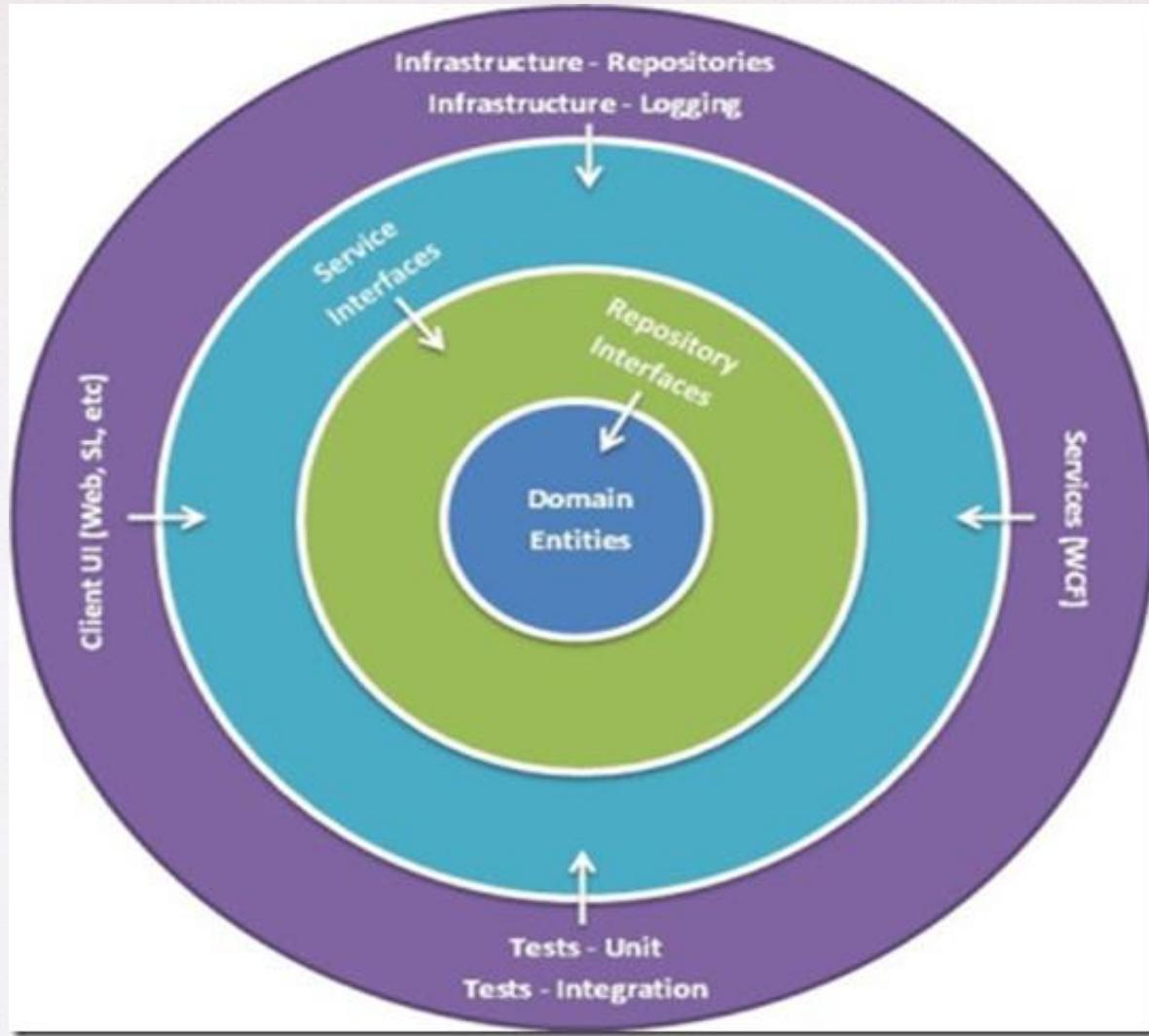




How?
Organize Your Code



How? Organize Your Code



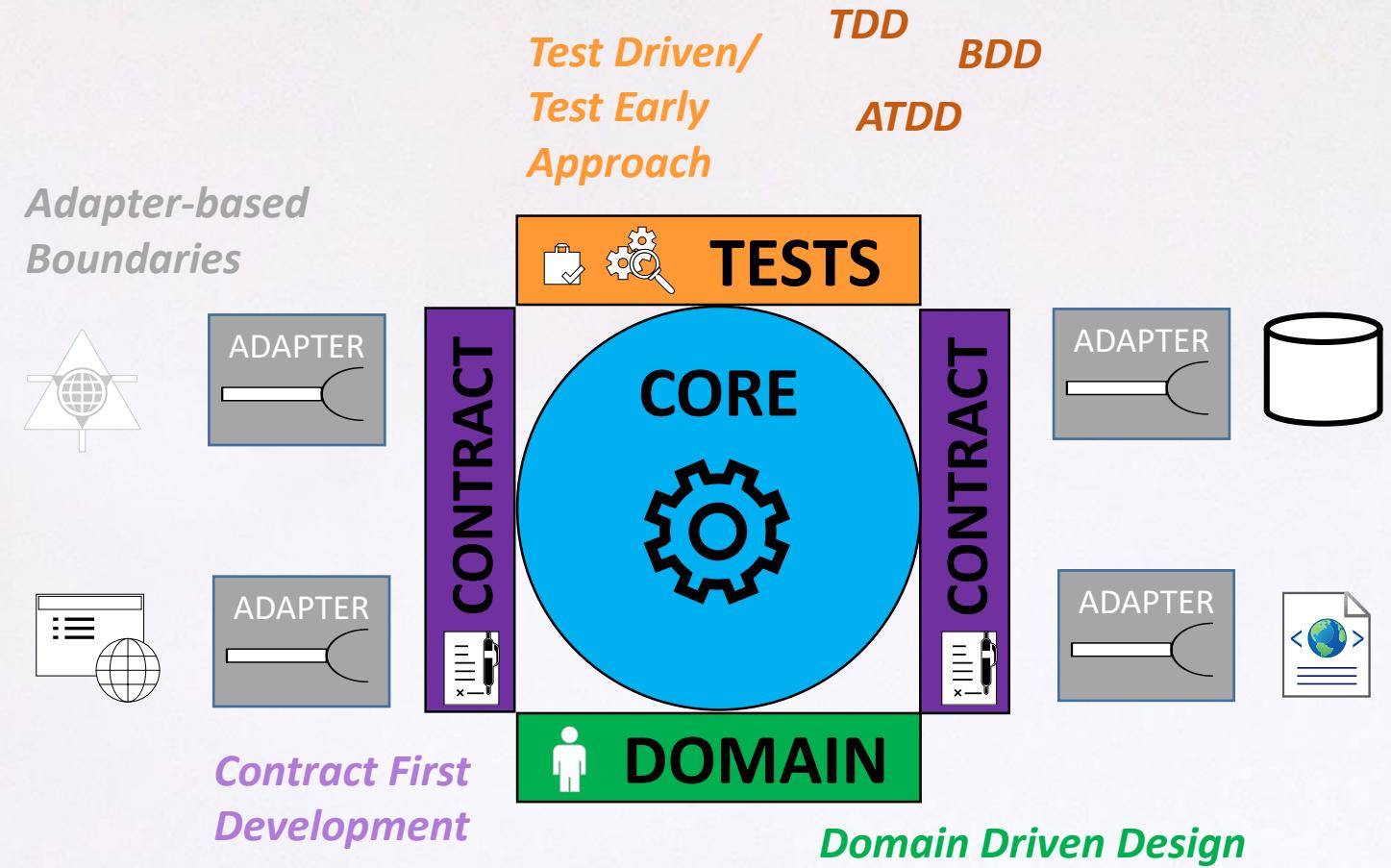
CASA Development Model

Core AS Application



How?

Organize Your Code





How?

Best Practices & Patterns

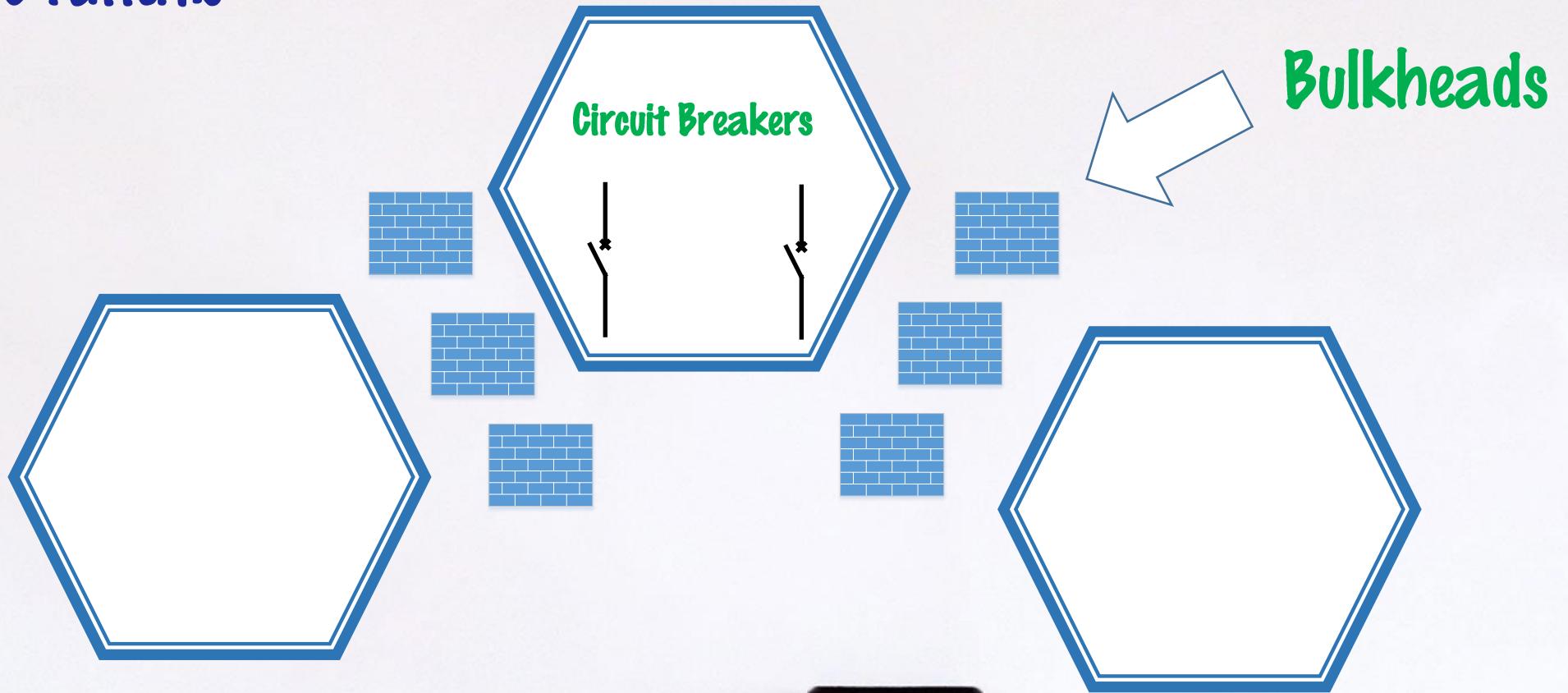
- SOLID Design Principles
 - You know these right?
- Domain Driven Design
 - State Objects
 - Core Business Logic
 - Other Concerns
- Contract First Development
 - Customer Driven Contracts



How?

Best Practices & Patterns

- Circuit Breakers
- Bulkheads





How?

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



How?

Organize Around Capabilities

- Which of these are capabilities?

Shopping Cart?
Purchase Path?
Company Catalog?
User Access?



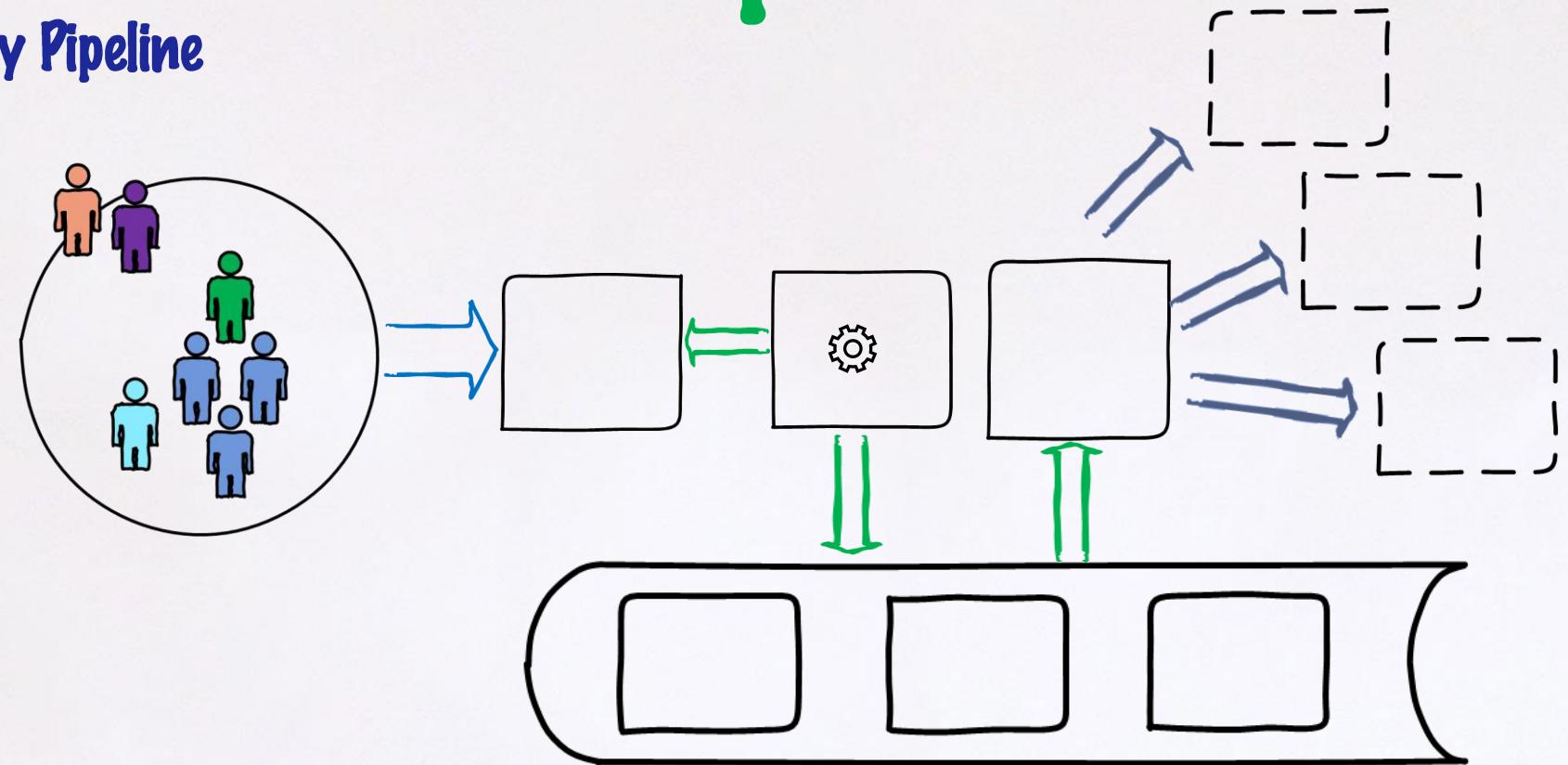
Database Access?

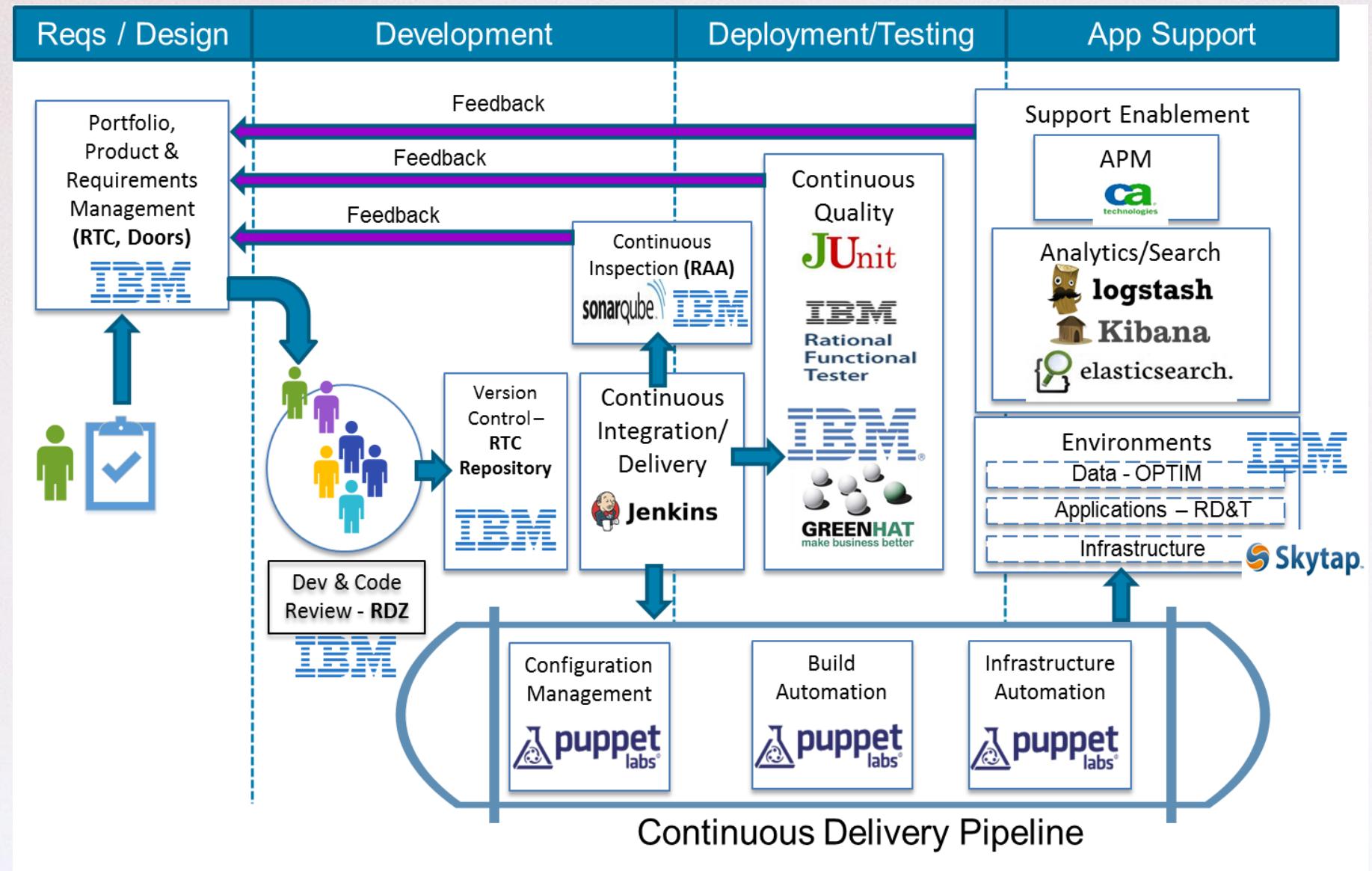


How?

Automate the Delivery Pipeline

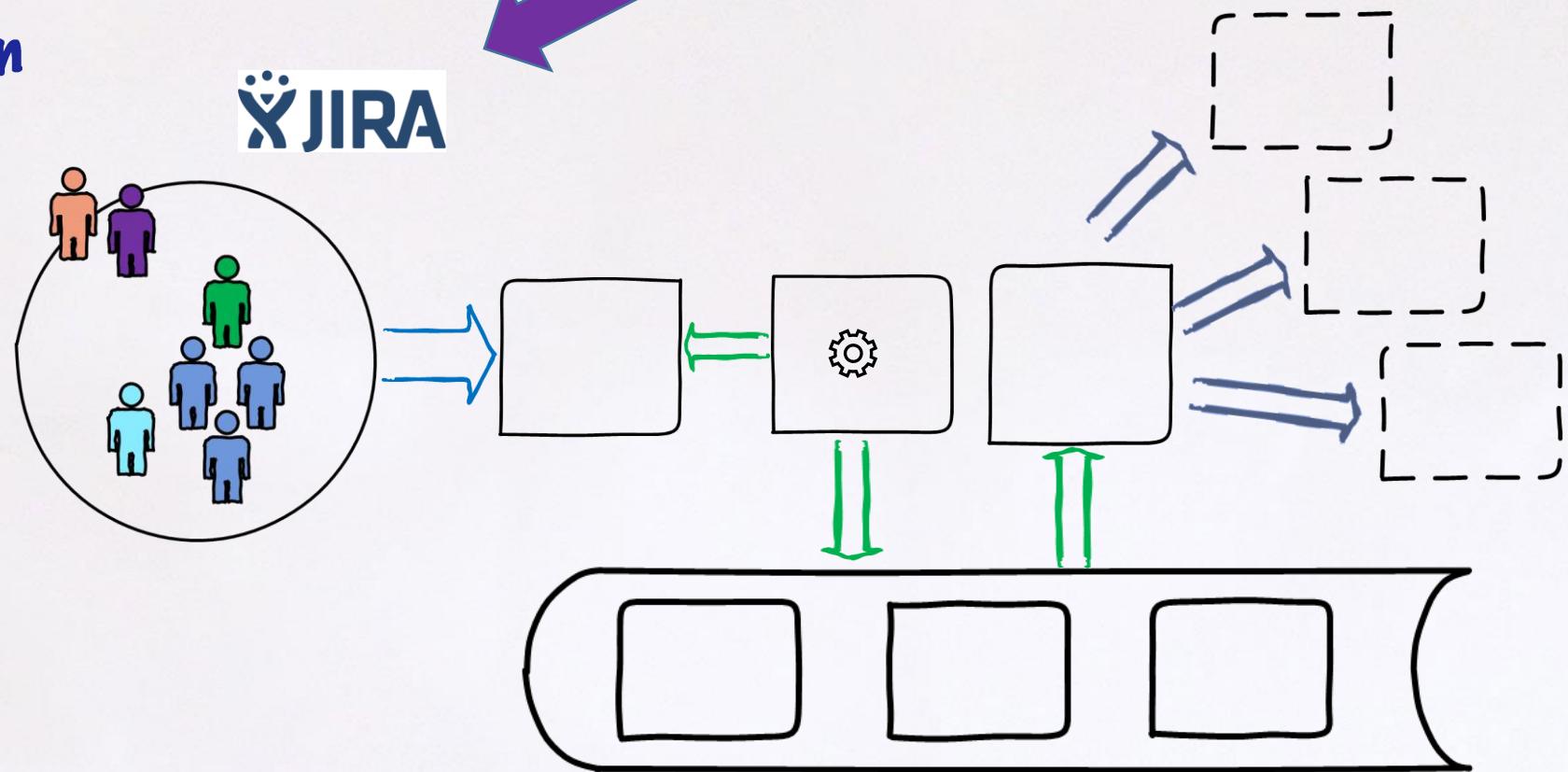
DevOps







How? Monitor Your Solution





How?

Abstracting Service Boundaries

- How do we abstract?
 - API Management
 - Message Queueing

Stay in the room for Shawn's talk
(NEXT) to learn more about
abstraction with an event-driven
approach

How? Abstract

- No ESB, right?



MuleSoft™



Experience APIs (UI Enablement)



Process APIs (Business Focused Microservices)



System APIs (Data Services)

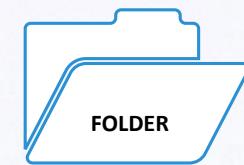


SYSTEM

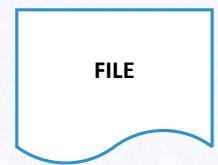
APPLICATION



DATABASE



FOLDER



FILE



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



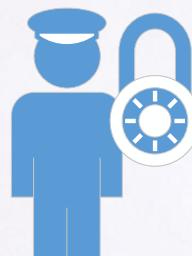
Real World
Determining Services:
Decomposition

Shopping Cart



Shopping
Cart

Item Lookup
Wish Lists



Security



User

Payment



Real World Determining Services: Decomposition

What if we are green field?

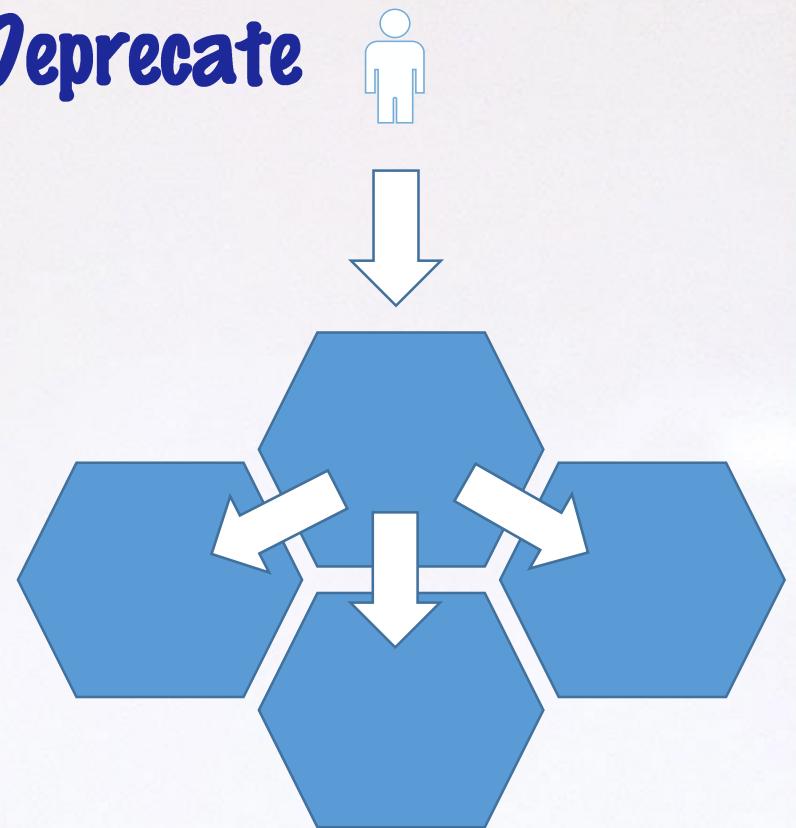
- Same Process
 - Gather Requirements (stories?)
 - Find business capabilities



What if I have larger services?

Real World
Determining Services:
Decomposition

- Orchestrate and Deprecate

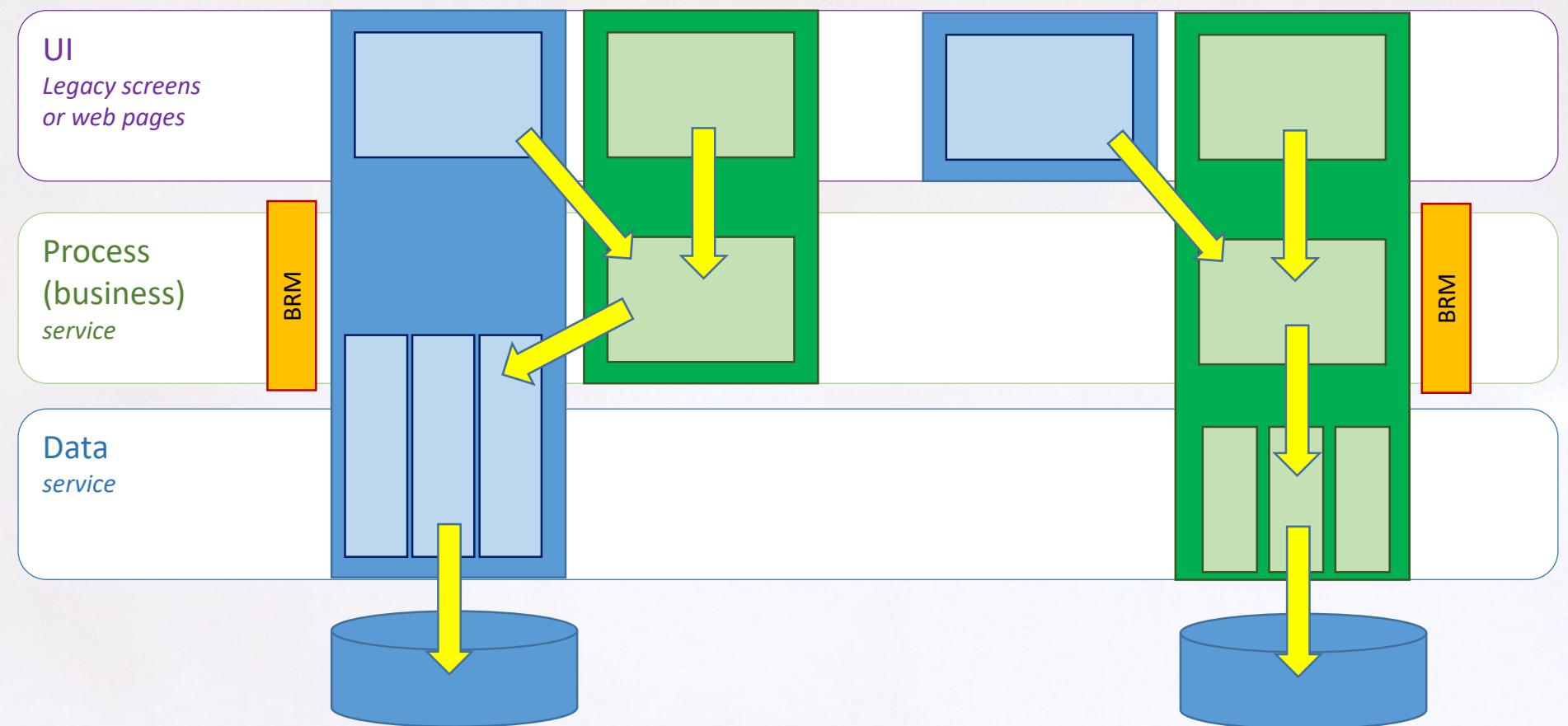


Real World Legacy Modernization



01 **Legacy Refactor**
Risk averse modernization

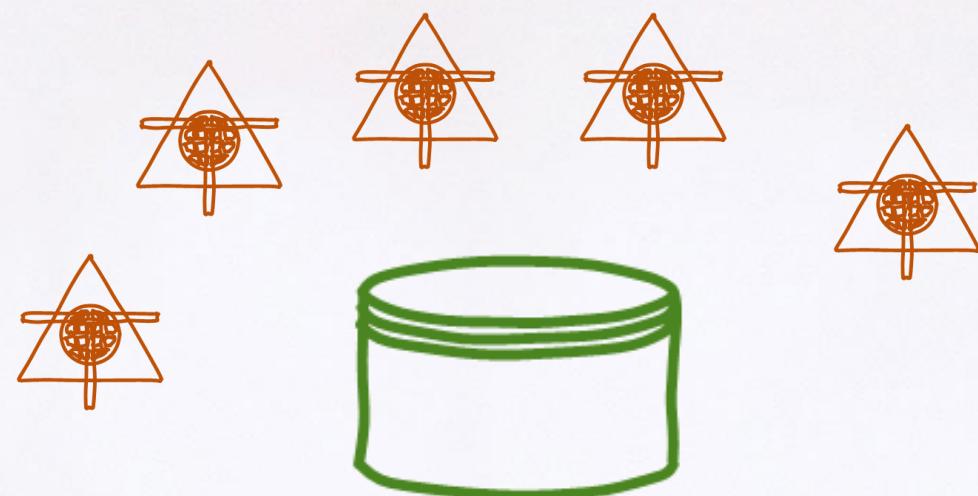
02 **Legacy Façade**
Add functionality to workflow





- Don't do this!!!

Real World Data Focused Microservices



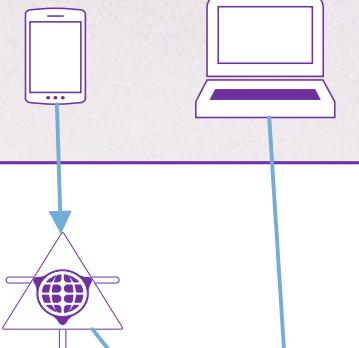
Real World Data Focused Microservices



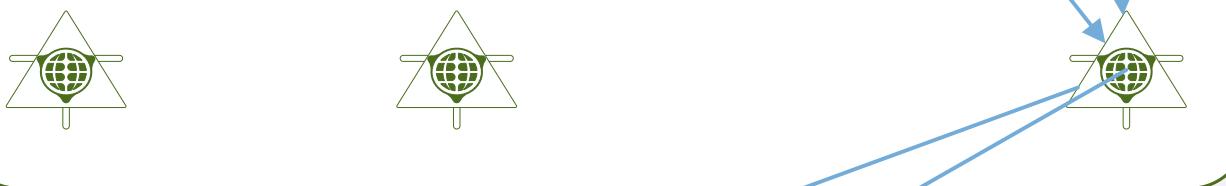
MuleSoft™



Experience APIs (UI Enablement)



Process APIs (Business Focused Microservices)



System APIs (Data Services)



SYSTEM

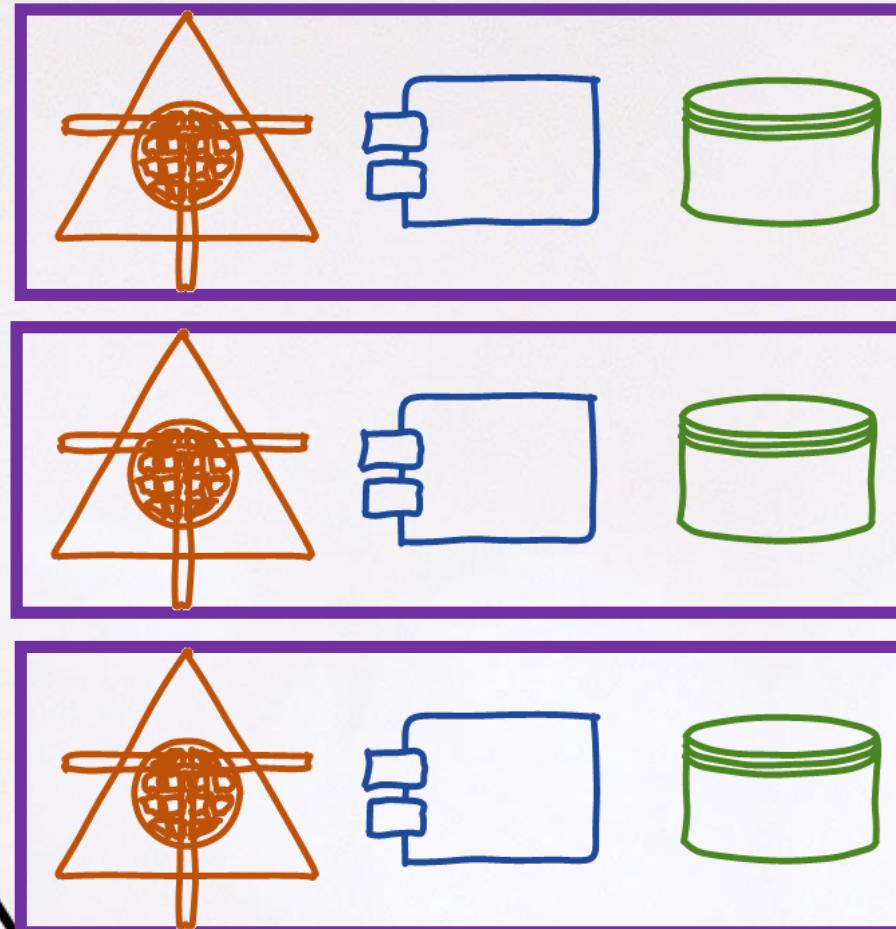
APPLICATION

DATABASE

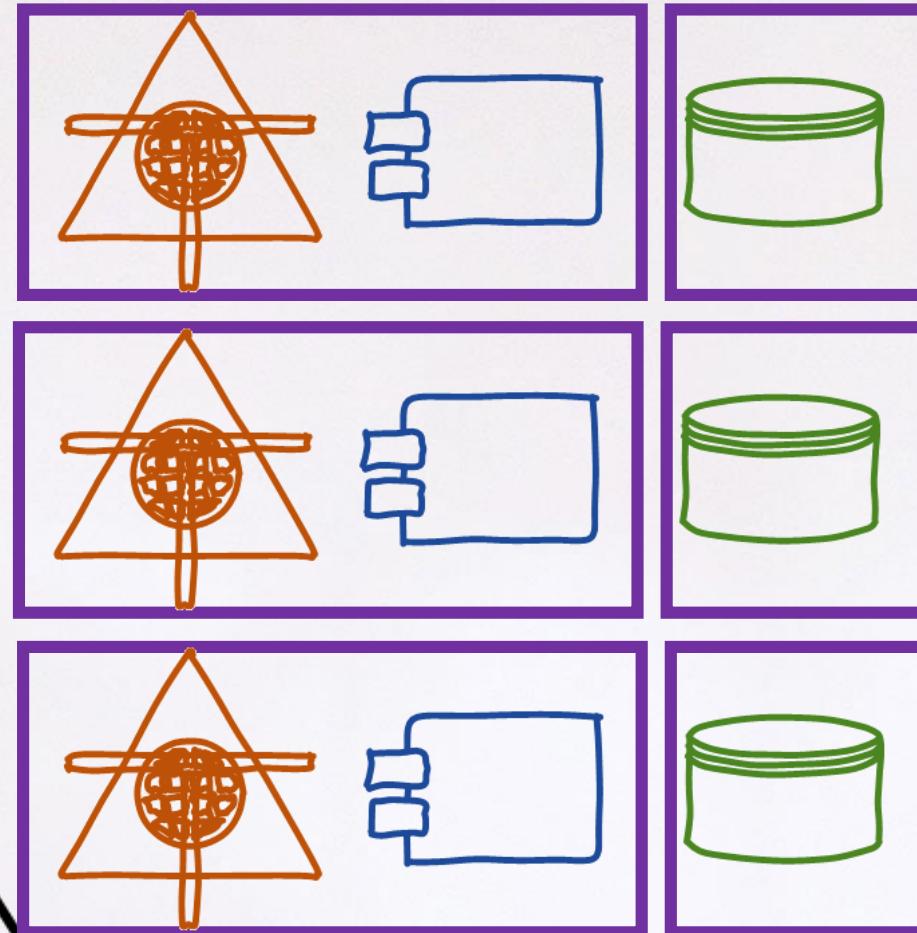
FOLDER

FILE

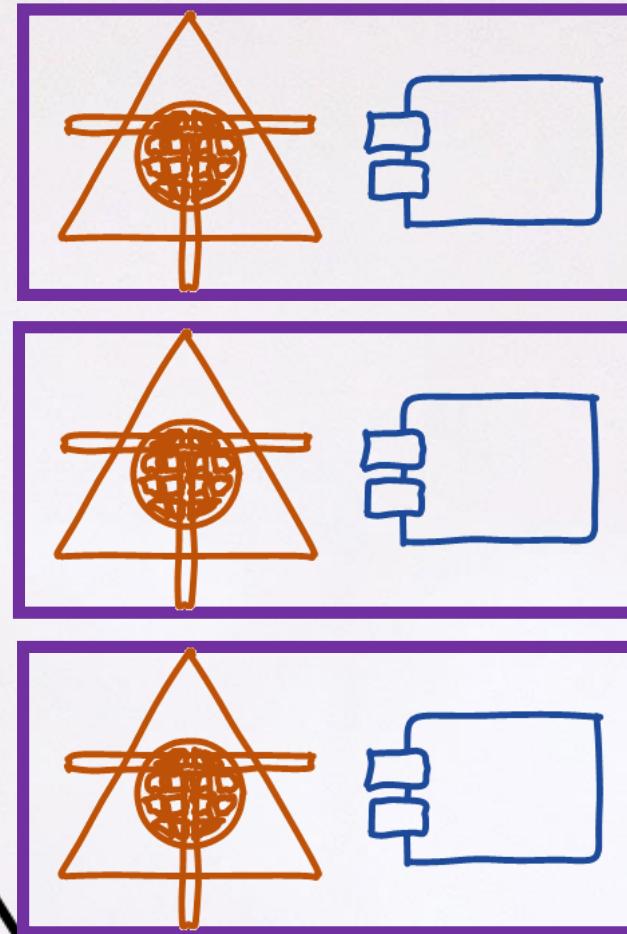
Real World Scaling



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

Code Mash Edition



MICROSERVICES

Lessons from the Trenches

@gbworld

