

Microservices Architecture

Vidya Vrat Agarwal

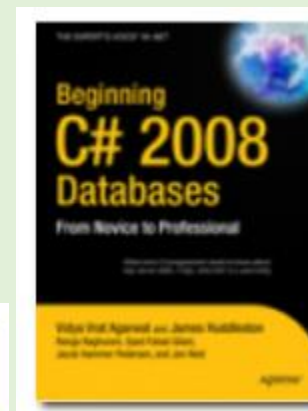
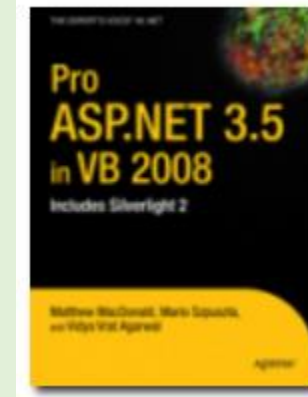
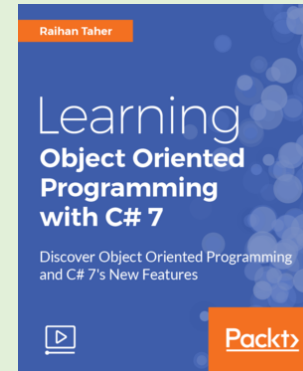
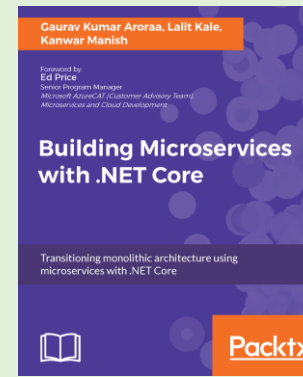
Principal Software Engineering Manager – Microsoft

www.MyPassionFor.Net | [@dotNetAuthor](https://twitter.com/dotNetAuthor)

<https://www.linkedin.com/in/vidyavrat/>

About Me

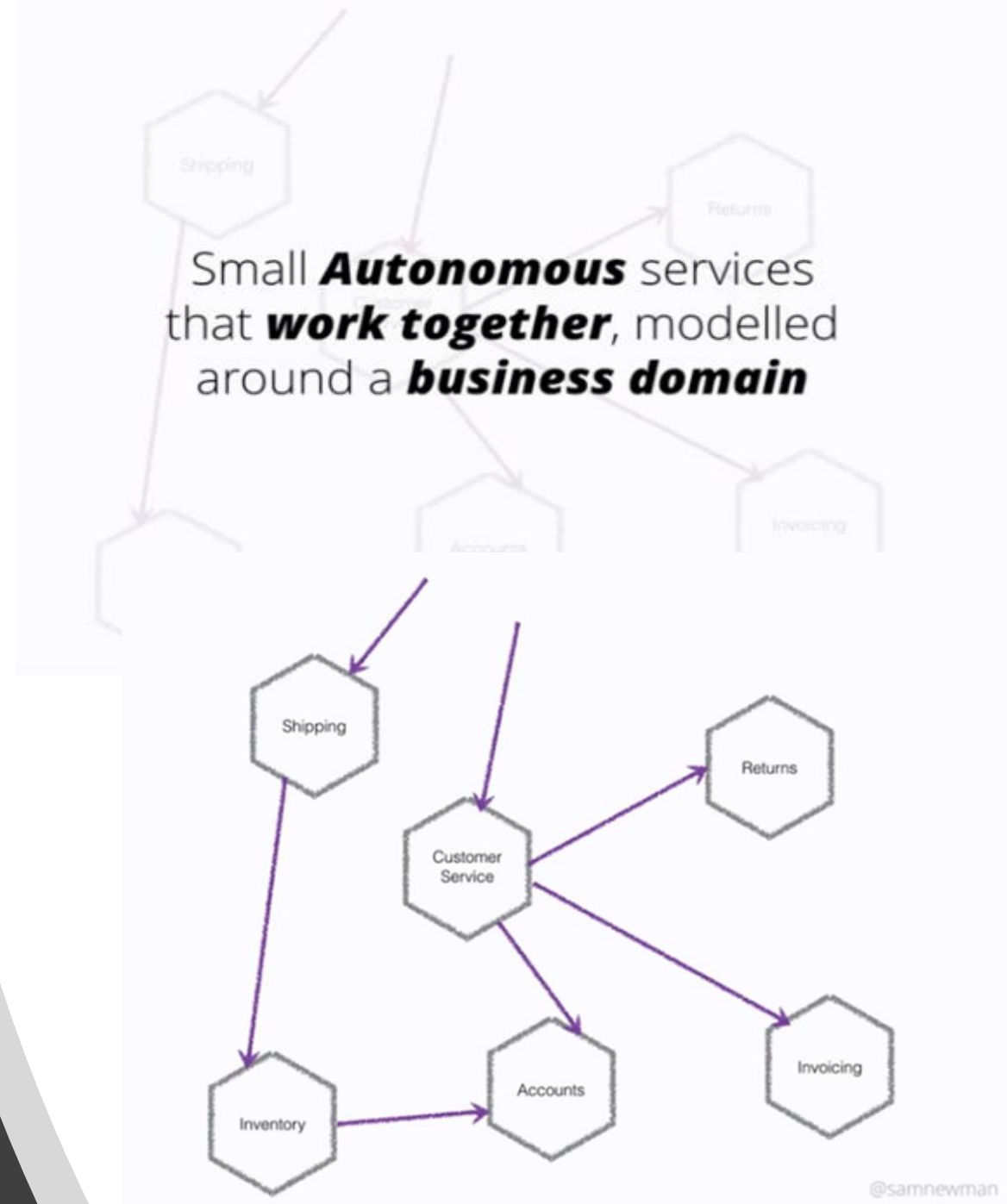
- 18+ years of industry experience
- Principal SWE Manager – Microsoft Azure CXP
- Microsoft MVP
- C# Corner MVP
- TOGAF Certified Architect
- Certified Scrum Master (CSM)
- Microsoft Certified (MCT, MCSD / MCAD .NET, MCTS etc.)
- Published Author (5) and Technical Reviewer (over a dozen)



What is a Microservice?

- “Microservice as an application component which is:

tightly scoped, loosely coupled,
strongly encapsulated,
independently testable,
independently deployable, and
independently scalable “

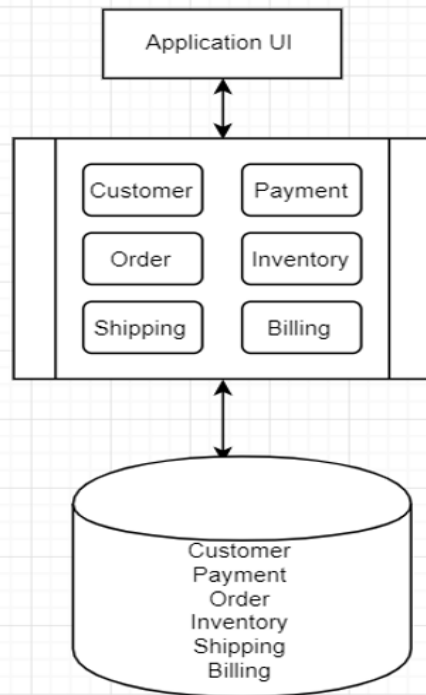


Finding a Bounded Context

A bounded context is an explicit boundary within which a domain model exists.

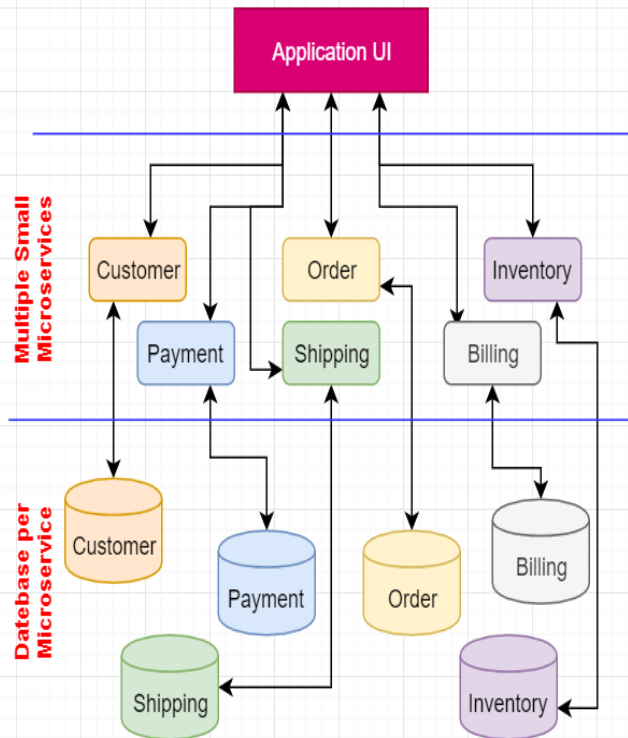
Monolith Architecture

Multiple services are build and deployed together



Single database serving all the features

Microservice Architecture



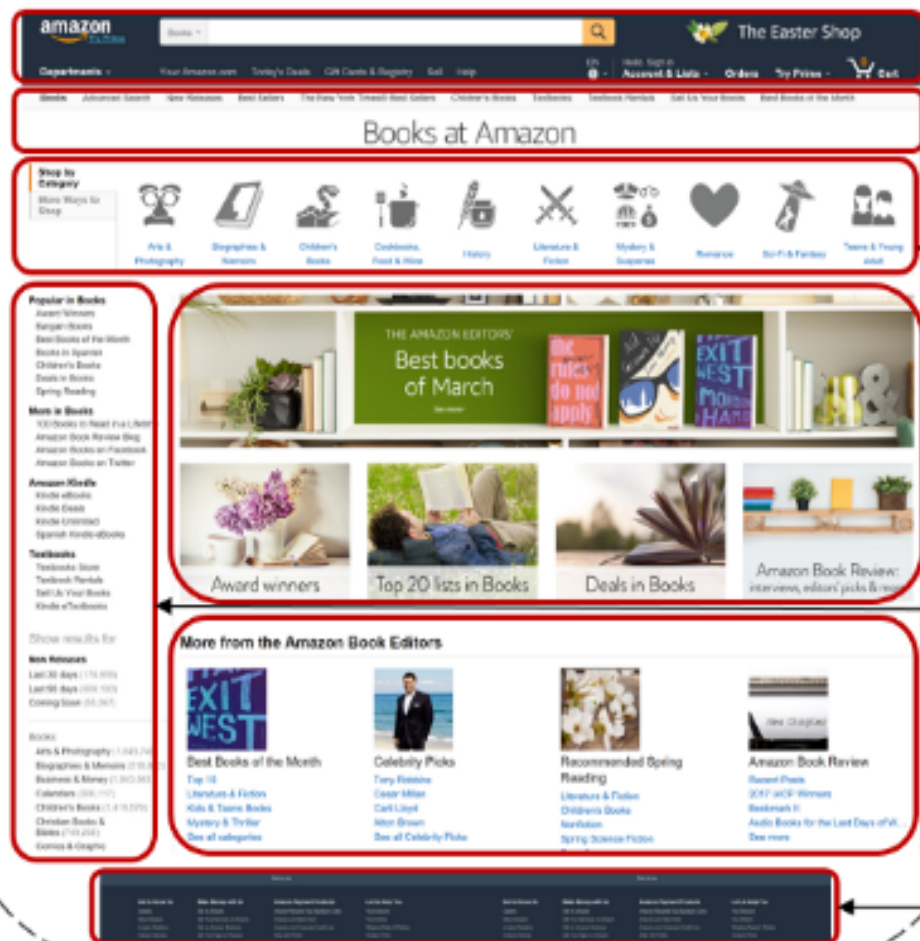
Multiple Small Microservices

Database per Microservice

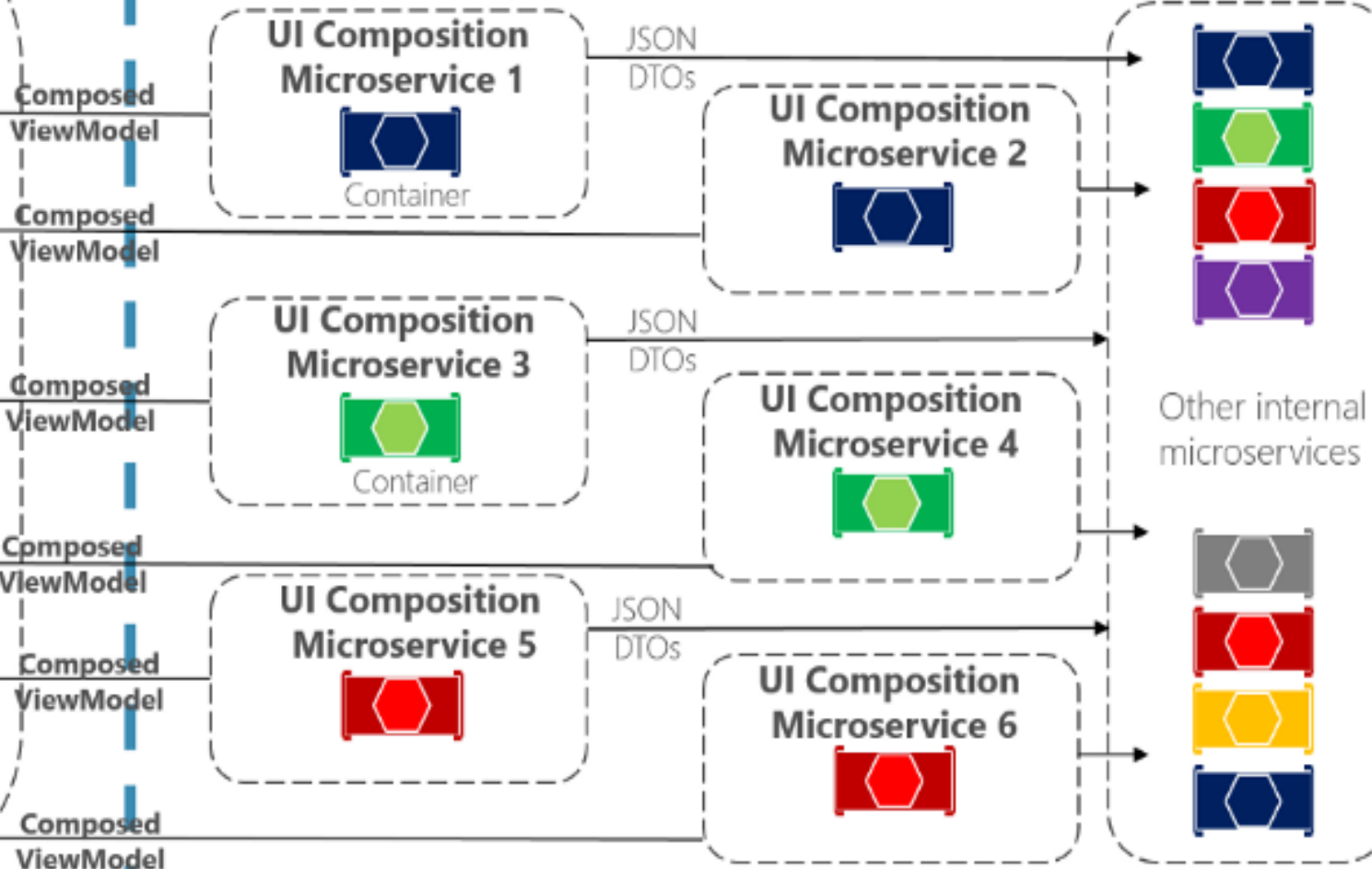
- Billing can collect payment
- Billing can own customer data
- Order can generate receipt
- Order can own inventory
- Payment can own T&Cs

Microservices In Action

Composite UI



Backend Microservices



Microservice Trade-Offs

Microservices provide benefits...

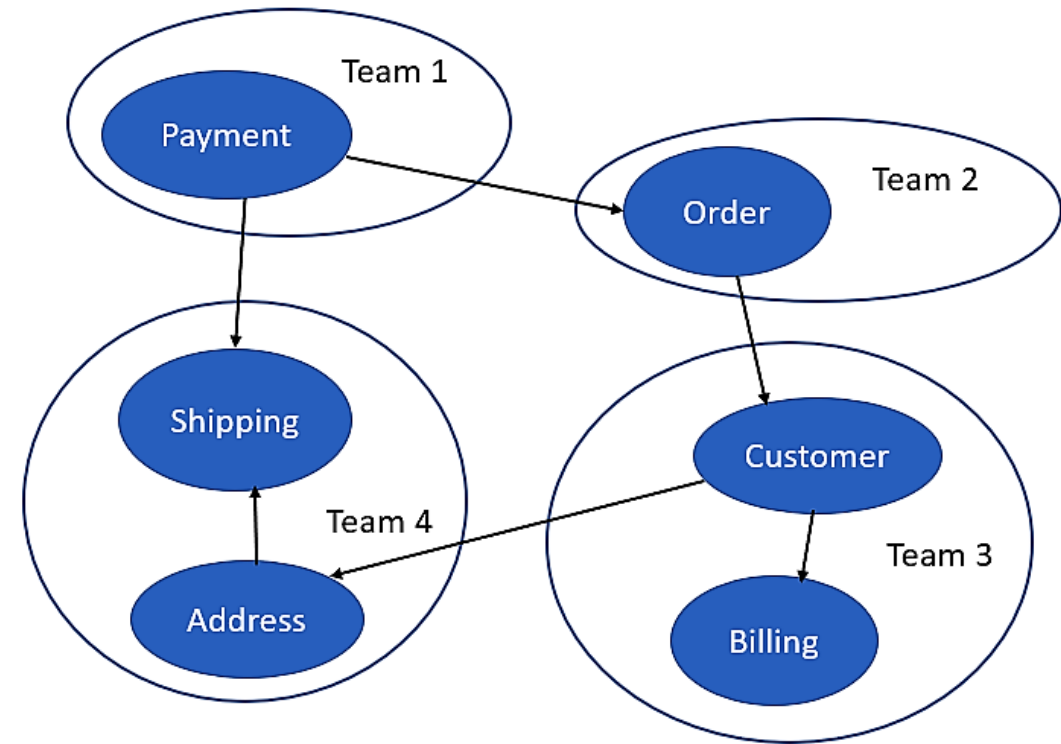
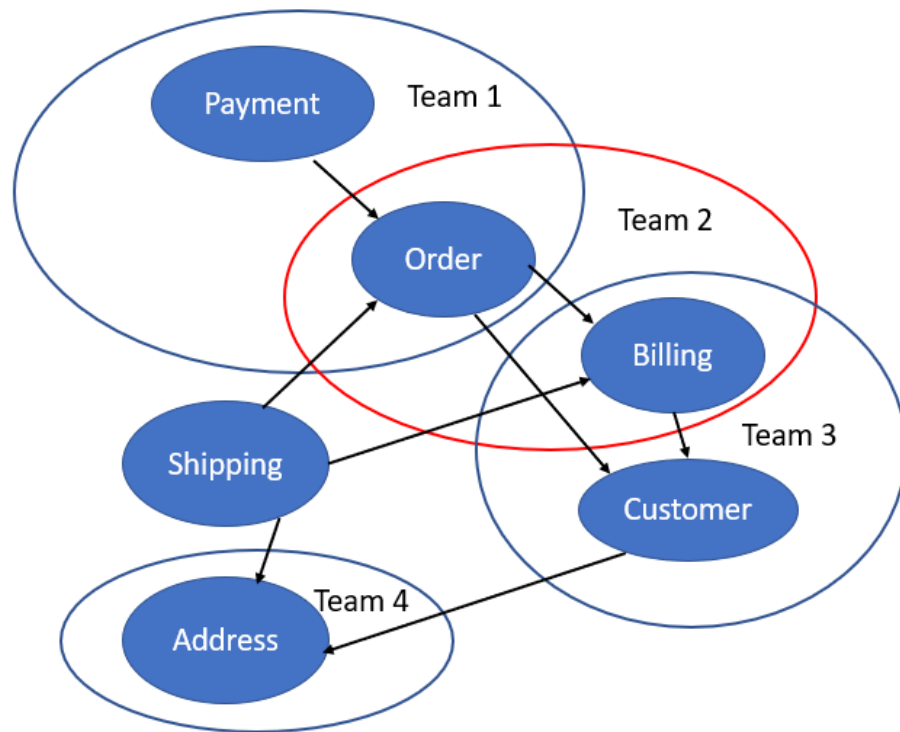
- ✓ Strong Module Boundaries: Microservices reinforce modular structure, which is particularly important for larger teams.
- ✓ Independent Deployment: Simple services are easier to deploy, and since they are autonomous, are less likely to cause system failures when they go wrong.
- ✓ Technology Diversity: With microservices you can mix multiple languages, development frameworks and data-storage technologies.

...but come with costs

- ✗ Distribution: Distributed systems are harder to program, since remote calls are slow and are always at risk of failure.
- ✗ Eventual Consistency: Maintaining strong consistency is extremely difficult for a distributed system, which means everyone has to manage eventual consistency.
- ✗ Operational Complexity: You need a mature operations team to manage lots of services, which are being redeployed regularly.

Service Ownership

- STOSA (Single Team Owned Service Architecture)



DevOps Metrics with Microservice

Deployment
frequency



Change
lead time



Change
fail rate



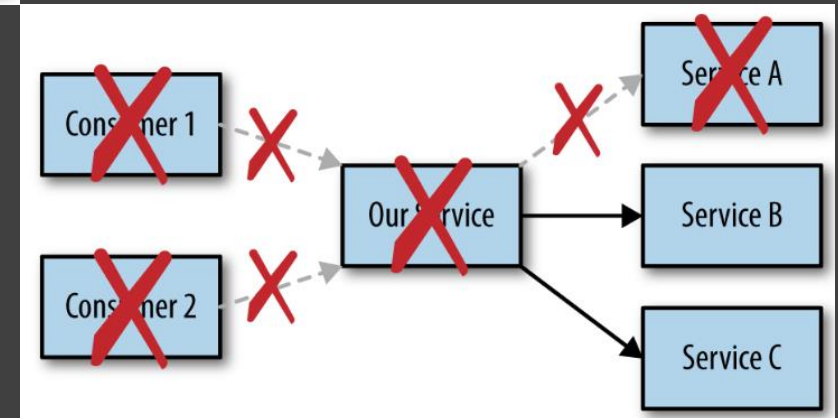
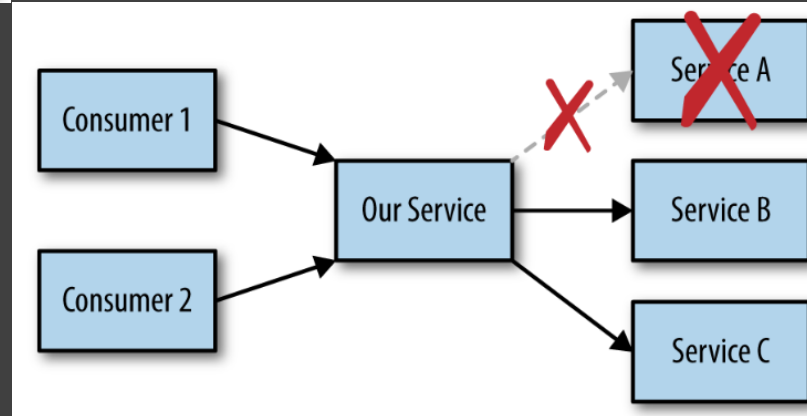
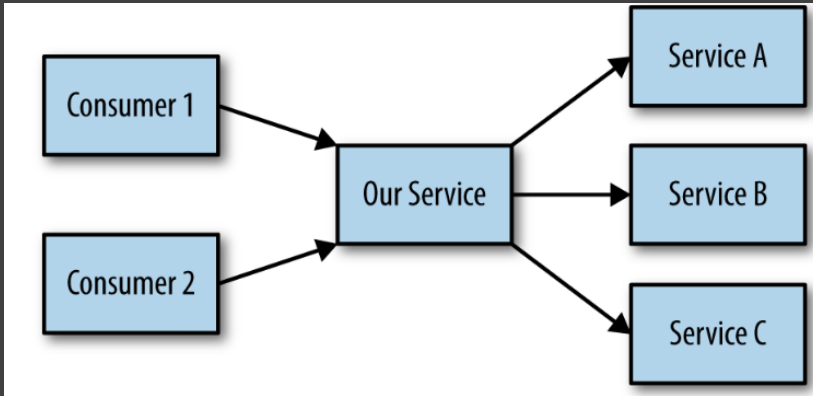
Mean time to
detect & repair

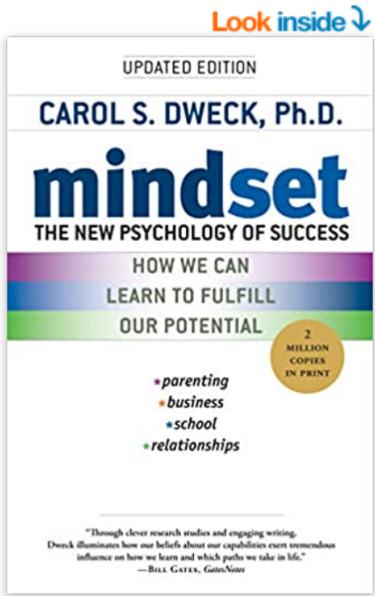


Agility performance indicators

Reliability performance indicators

Dealing With Service Failures





Mindset: The New Psychology of Success Paperback – Illustrated, December 26, 2007

by Carol S. Dweck (Author)
★★★★☆ 8,239 ratings
#1 Best Seller in Graduate Test Guides

Best Deal

> See all formats and editions

Kindle
\$11.99

Read with Our Free App

Audiobook
1 Credit

Hardcover
\$21.49 ✓prime

51 Used from \$6.49
24 New from \$17.49
3 Collectible from \$29.99

Paperback
\$10.14 ✓prime

368 Used from \$2.28
44 New from \$4.39
12 Collectible from \$5.26



Mindset: The New Psychology of Success Paperback – Illustrated, December 26, 2007

by Carol S. Dweck (Author)
★★★★☆ 8,239 ratings
#1 Best Seller in Graduate Test Guides

Best Deal

> See all formats and editions

Kindle
\$11.99

Read with Our Free App

Audiobook
1 Credit

Hardcover
\$21.49 ✓prime

51 Used from \$6.49
24 New from \$17.49
3 Collectible from \$29.99

Paperback
\$10.14 ✓prime

368 Used from \$2.28
44 New from \$4.39
12 Collectible from \$5.26

Watch It Again



Everyone's Watching



Continue Watching for <ProfileName>



2 Second Rule



Order Confirmation

Hello vidyavrat,

Thank you for shopping with us. We'll send a confirmation when your item ships.

Details

Order #112-2417936-6786637

Arriving:
tomorrow, February 1

Ship to:
**Vidya
Redmond, WA**
Order Total: \$16.50

[View or manage order](#)



● **Amazon.com Marketplace** <payments-messages@amazon.com>
To: vidya_mct@yahoo.com



Oct 2 at 10:14 PM



[Your Orders](#) | [Your Account](#) | [Amazon.com](#)

Order Cancellation

Order #113-4827284-6541824

Hello vidyavrat,

We're writing to inform you that your order from Book Depository US has been canceled because the item you purchased is out of stock. Please return and place your order again at a later time. We're sorry for the inconvenience this has caused. In most cases, you pay for items when we ship them to you, so you won't be charged for items that are canceled.*

Pattern: Bulkhead

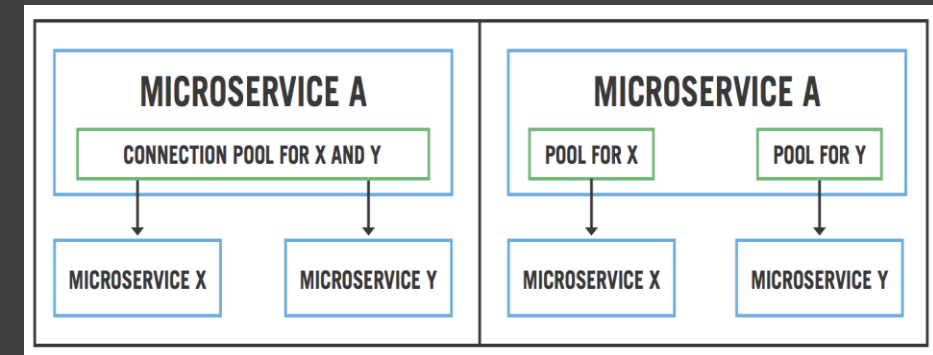
The Bulkhead pattern is a type of application design that is “**tolerant of failure**”. In a bulkhead, elements of an application are isolated into pools so that if one fails, the others will continue to function.

The bulkhead pattern helps to fix a number of different quality of service related issues.

- Propagation of Failure:** Because solutions are contained and do not share resources (storage, synchronous service-to-service calls, etc.), their associated failures are contained and do not propagate. When a service suffers a programmatic (software) or infrastructure failure, no other service is disrupted.

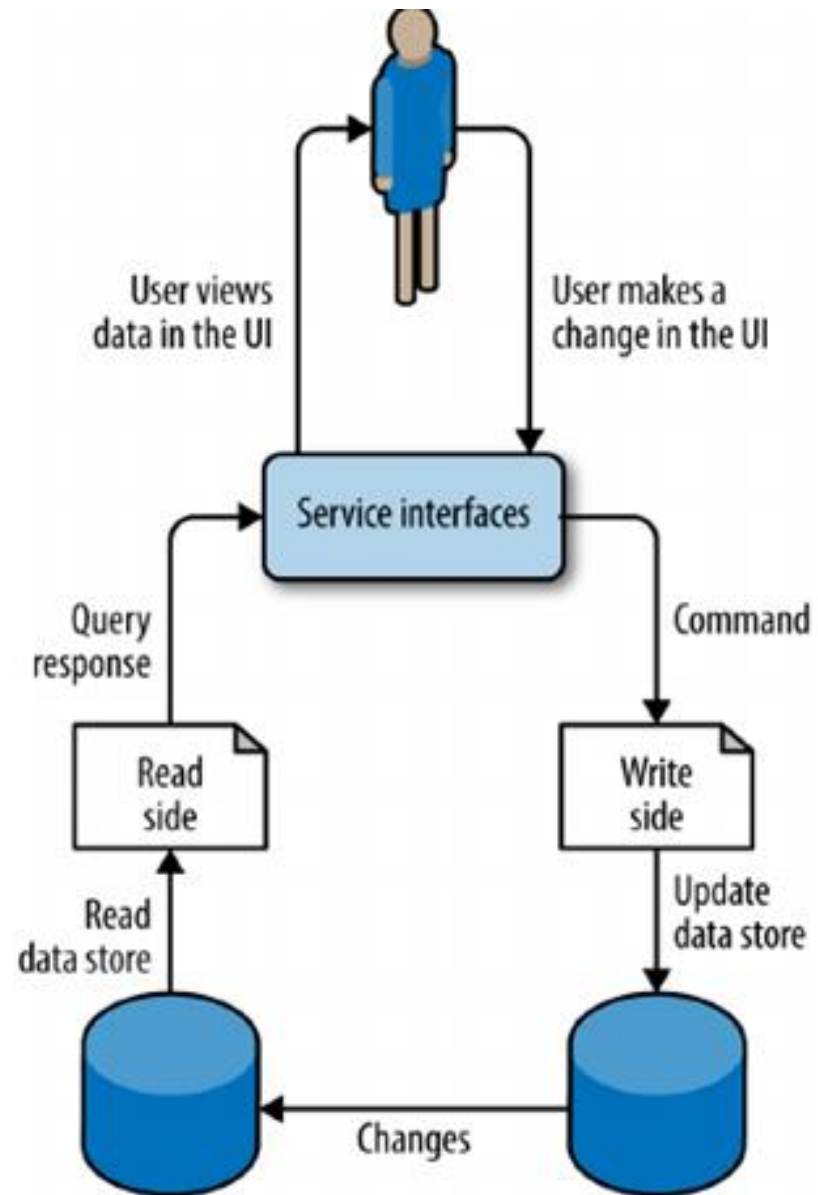
- Noisy Neighbors:** If implemented properly, network, storage and compute segmentation ensure that abnormally large resource utilization by a service does not affect other services outside of the bulkhead (fault isolation zone).

- Unusual Demand:** The bulkhead protects other resources from services experiencing unpredicted or unusual demand.



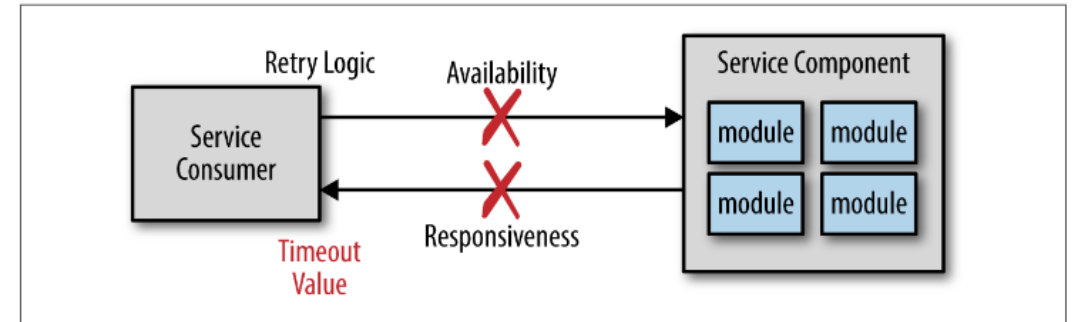
Pattern: CQRS

- Command Query Responsibility Segregation



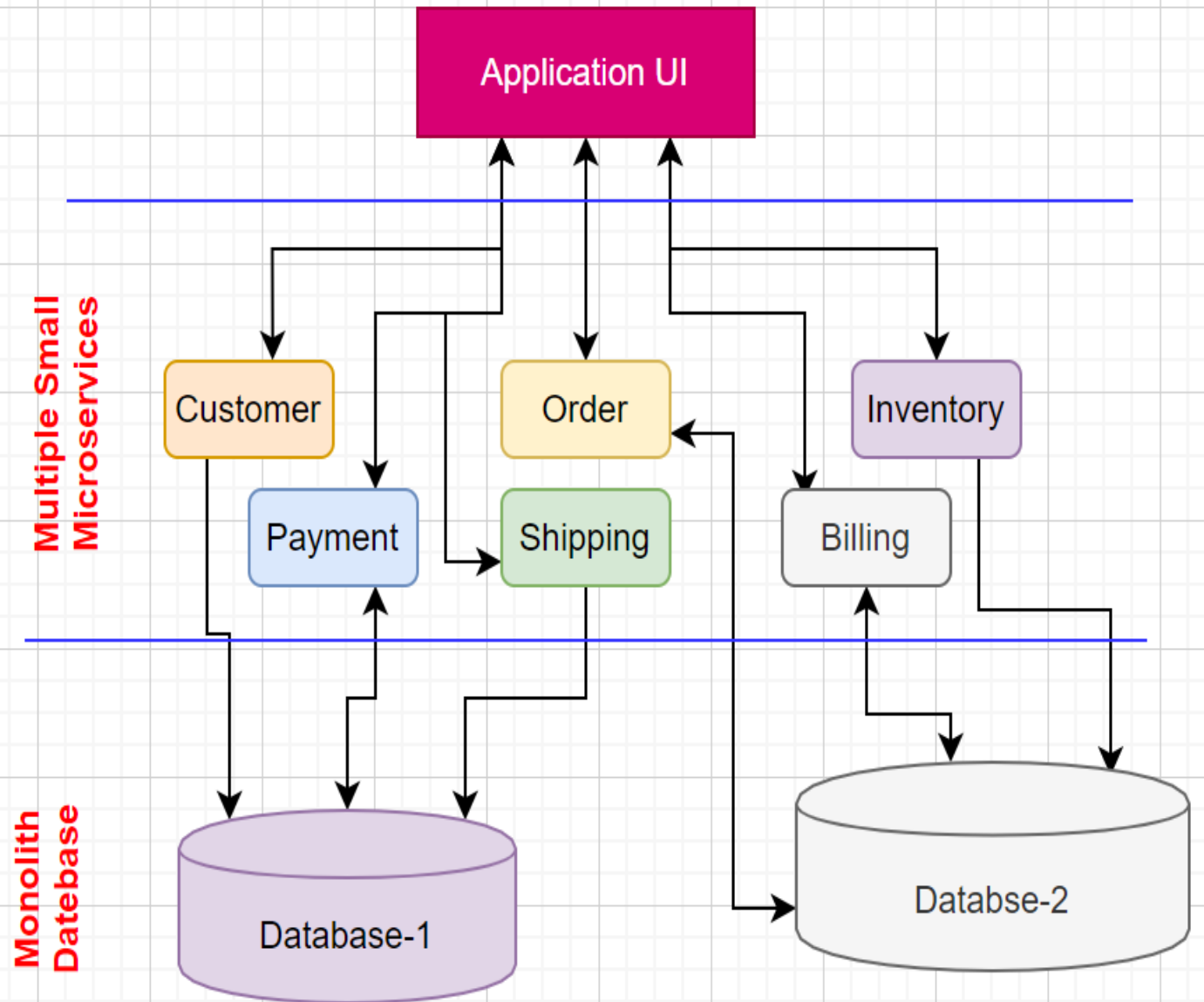
AntiPattern - Timeout

- Microservices is a distributed architecture, meaning all of the components (i.e., services) are deployed as separate applications and are accessed remotely through some sort of remote access protocol. One of the challenges of any distributed architecture is managing remote process availability and responsiveness.
- However, assuming the service was reached, and request was made, what happens if the service doesn't respond? In this case the service consumer can choose to wait indefinitely or leverage some sort of timeout value.

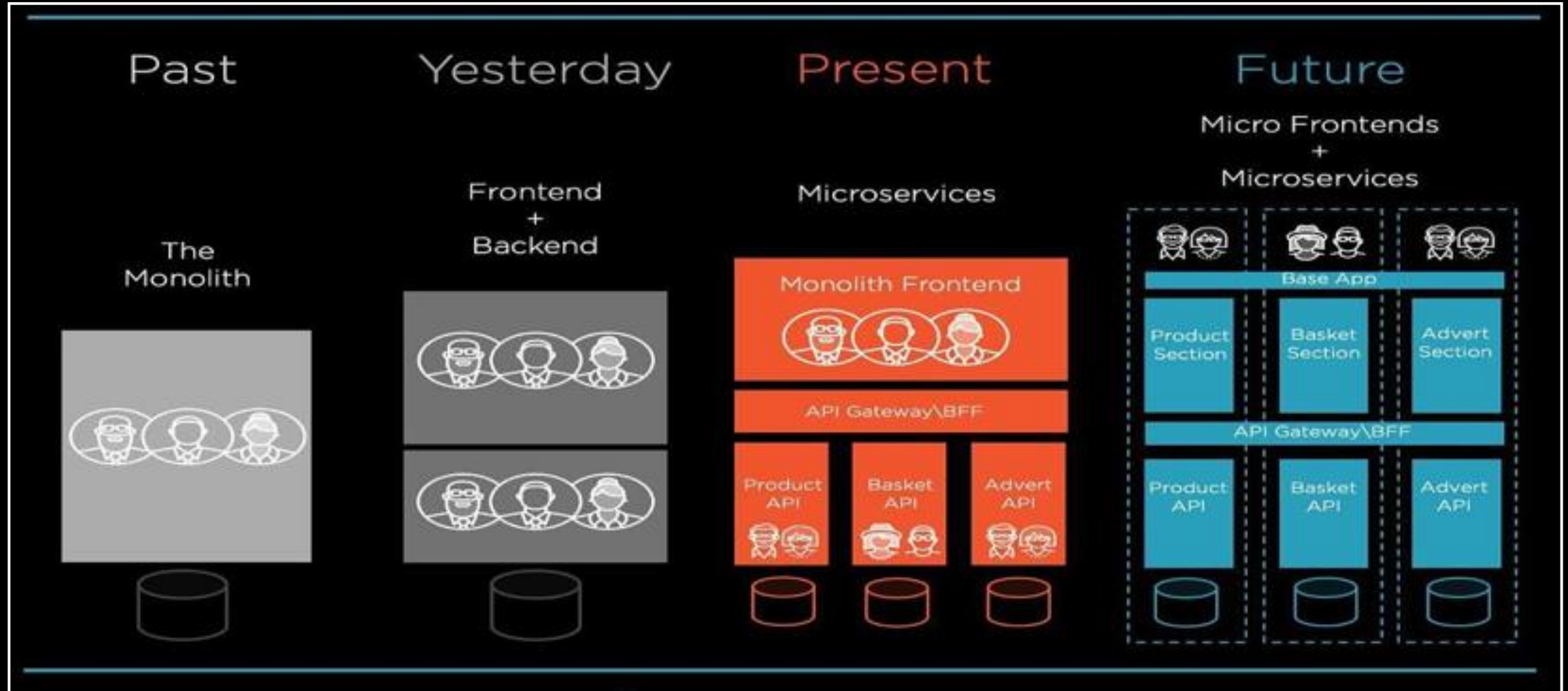


AntiPattern - Distributed Monolith

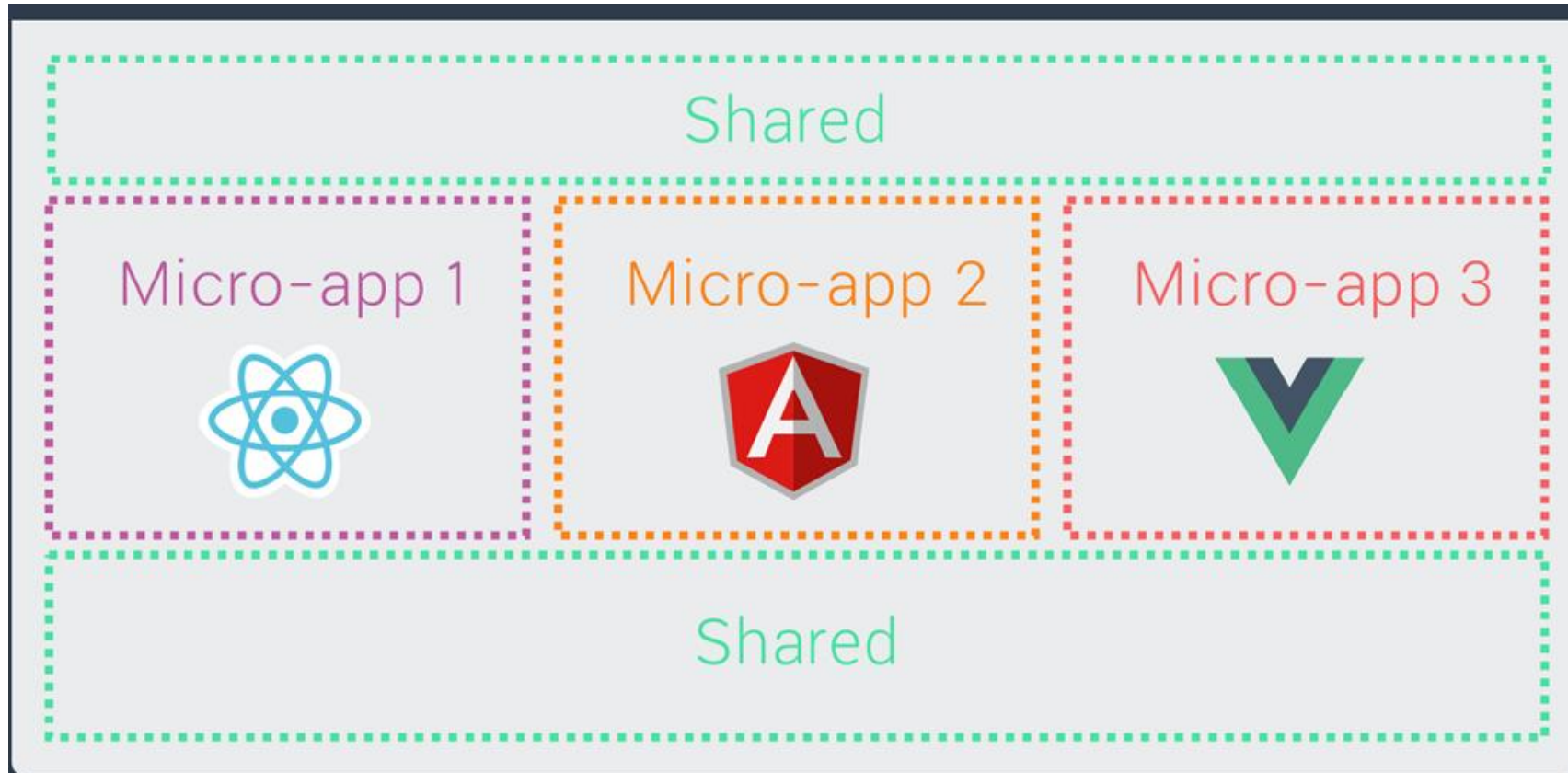
Distributed Monolith



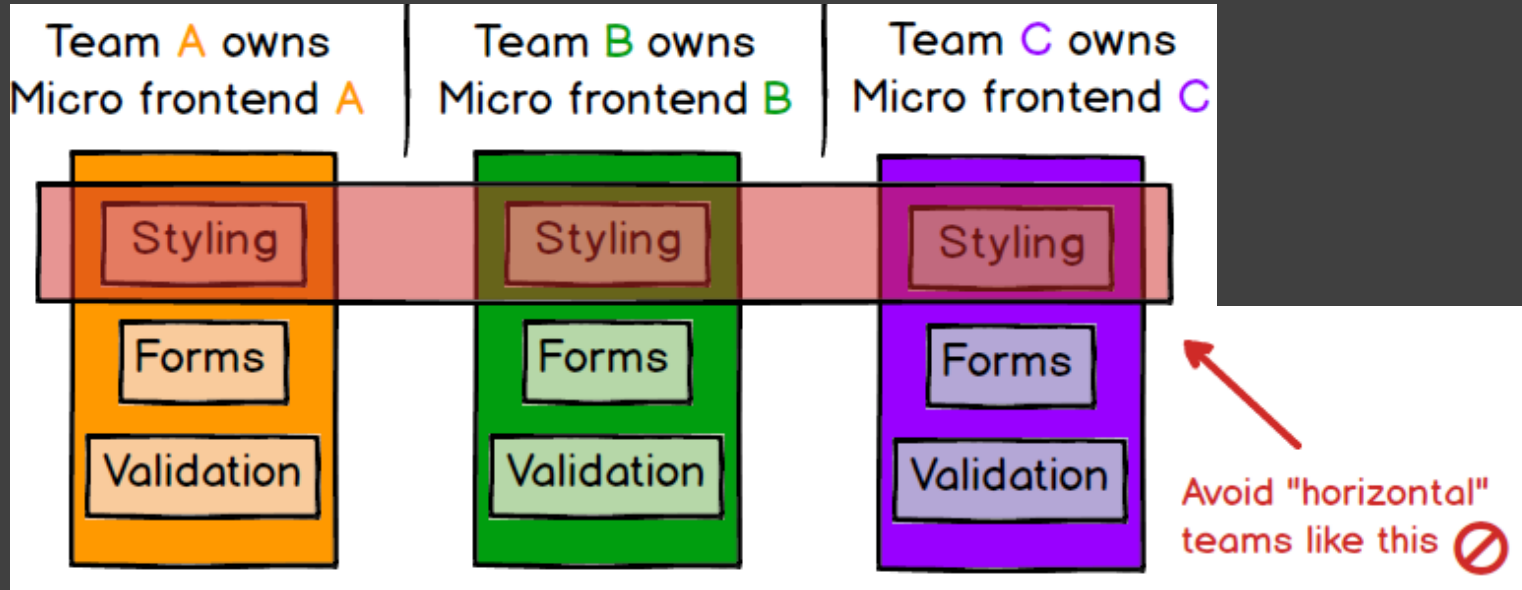
Evolutionary Architecture



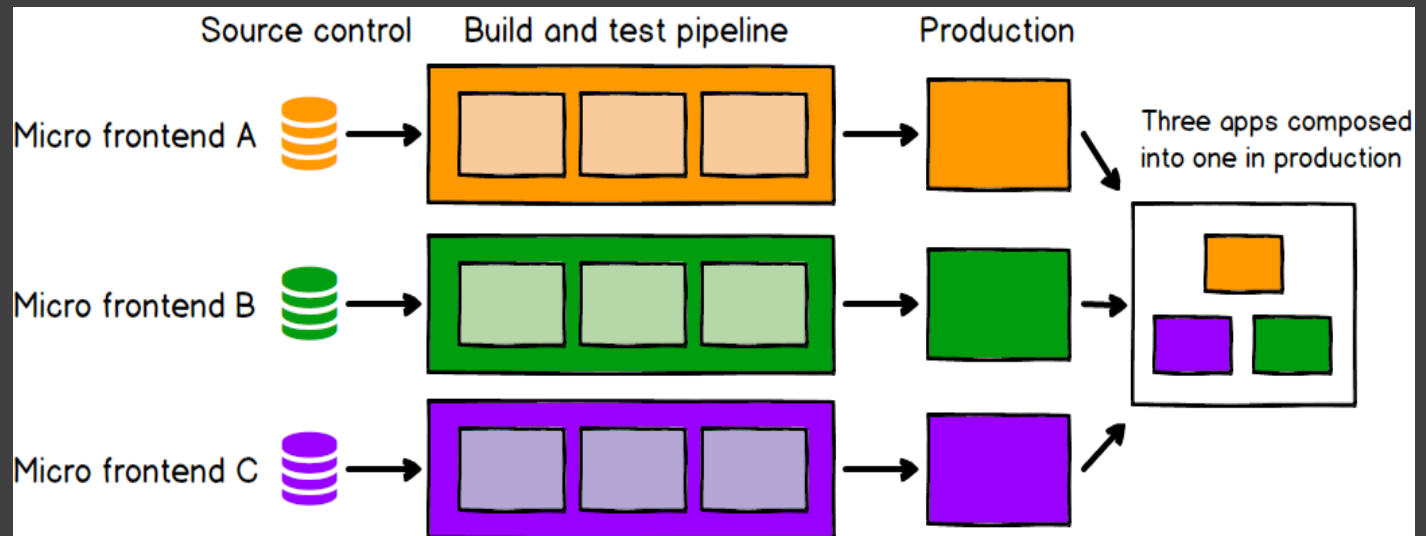
Microfrontend / Microapp



Microapp Ownership



3 product-oriented, "vertical" teams ✓



Thank You

www.MyPassionFor.Net | [@dotNetAuthor](#)

<https://www.linkedin.com/in/vidyavrat/>

<https://github.com/vidyavrat>