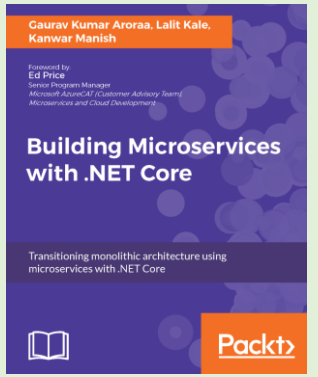
# Microservices: What, Why & How?

**Vidya Vrat Agarwal**
**Principal Architect T-Mobile Inc.**

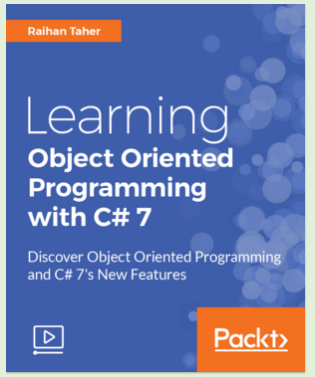
www.MyPassionFor.Net | **@dotNetAuthor**
**https://www.linkedin.com/in/vidyavrat/**

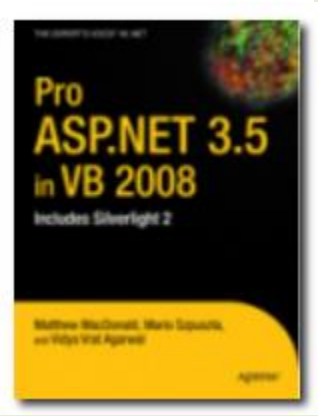
# About Me

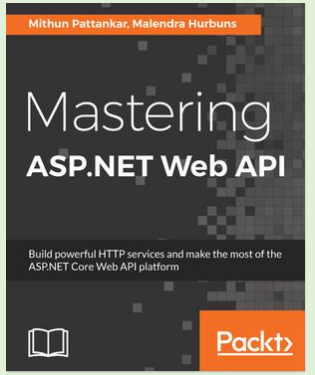
- Over 18+ years of industry experience
- Working as Principal Architect with T-Mobile Inc USA
- Microsoft MVP
- C# Corner MVP
- TOGAF Certified Architect
- Certified Scrum Master (CSM)
- Microsoft Certified (MCT, MCSD / MCAD .NET, MCTS etc.)
- Published Author (5) and Technical Reviewer (over a dozen)

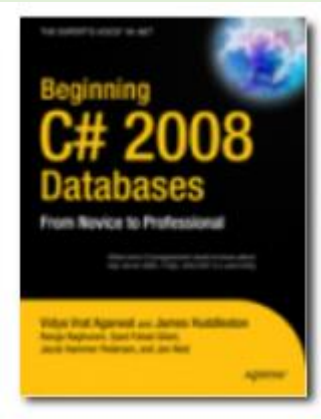






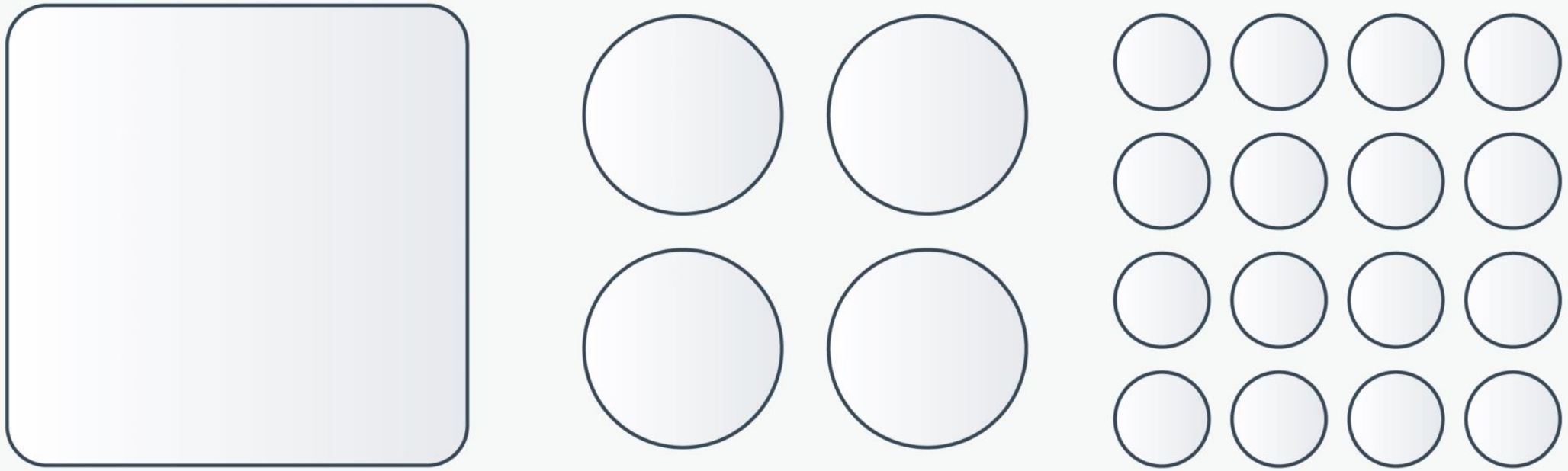

Microsoft Most Valuable Professional

C# Corner MVP

THE Open GROUP TOGAF® 9 Certified

Scrum Alliance®

# Monolithic vs. SOA vs. Microservices



**Monolithic**

Single Unit

**SOA**

Coarse-grained

**Microservices**

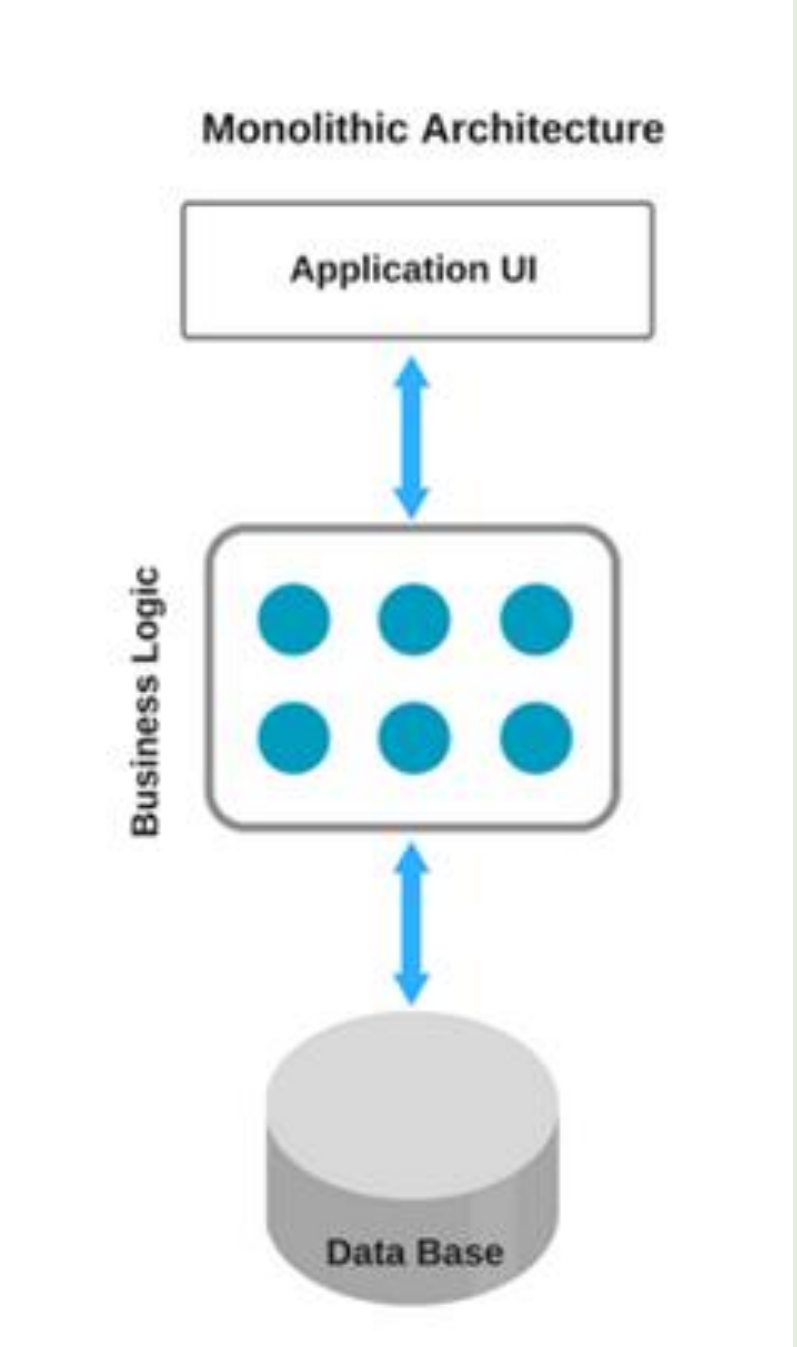
Fine-grained

# Monolith Pros:

•**Fewer Cross-cutting Concerns**: The major advantage of the monolithic architecture is that most apps typically have a large number of cross-cutting concerns, such as logging, rate limiting, and security features such audit trails and DOS protection. When everything is running through the same app, it's easy to hook up components to those cross-cutting concerns.
•**Less Operational Overhead**: Having one [large] application means there's only one application you need to set up logging, monitoring, testing for. It's also generally less complex to deploy.
•**Performance**: There can also be performance advantages, since shared-memory access is faster than inter-process communication (IPC).

# Monolith Cons:

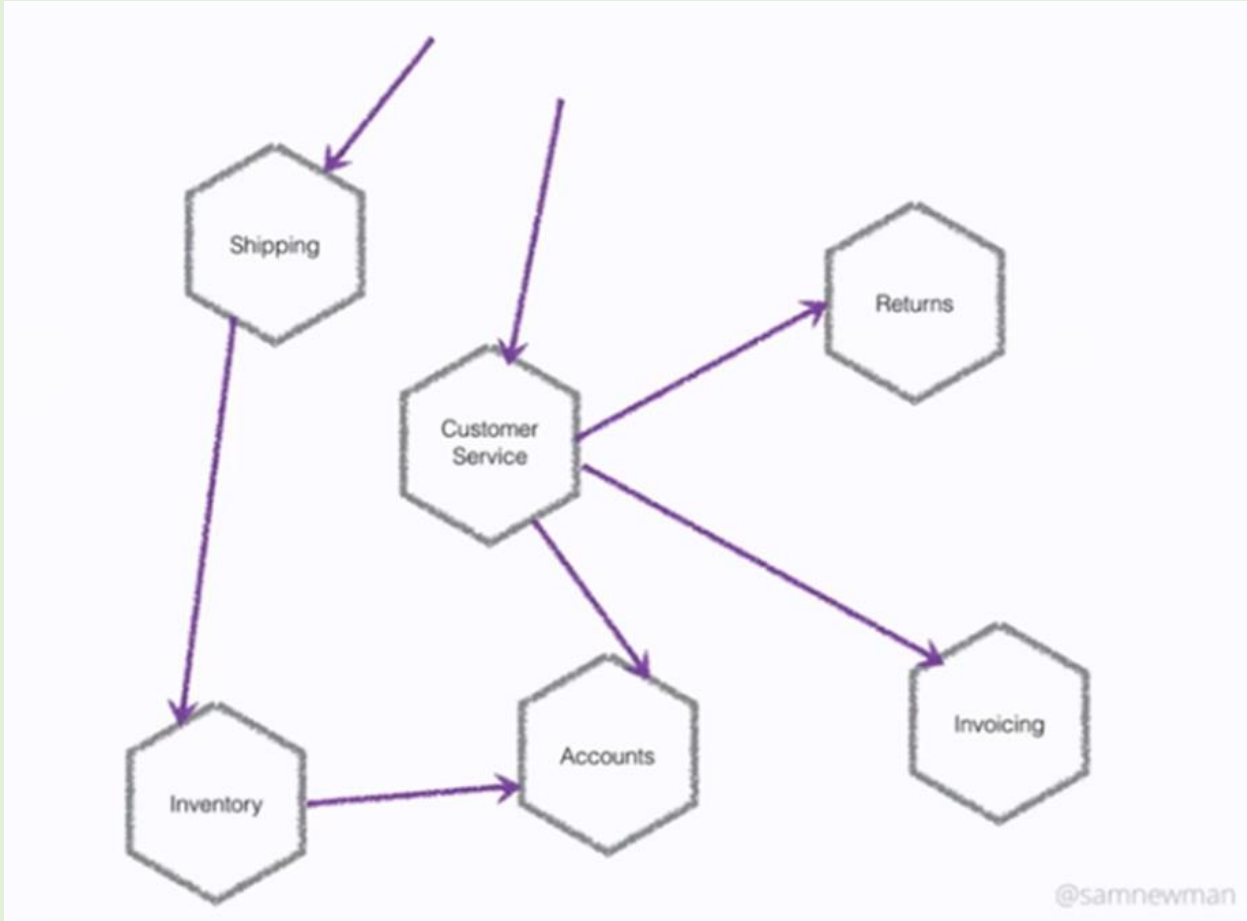
•**Tightly Coupled**: Monolithic app services tend to get tightly coupled and entangled as the application evolves, making it difficult to isolate services for purposes such as independent scaling or code maintainability.
•**Harder To Understand**: Monolithic architectures are also much harder to understand, because there may be dependencies, side-effects, and magic which are not obvious when you're looking at a particular service or controller.
•**Deploy all or none:** When a new change needs to be pushed, whole service needs to be deployed.
•**Scale all or none:** Scale up/down, its for entire functionality.
•Single point of failure: if a server is down, entire functionality is broken.

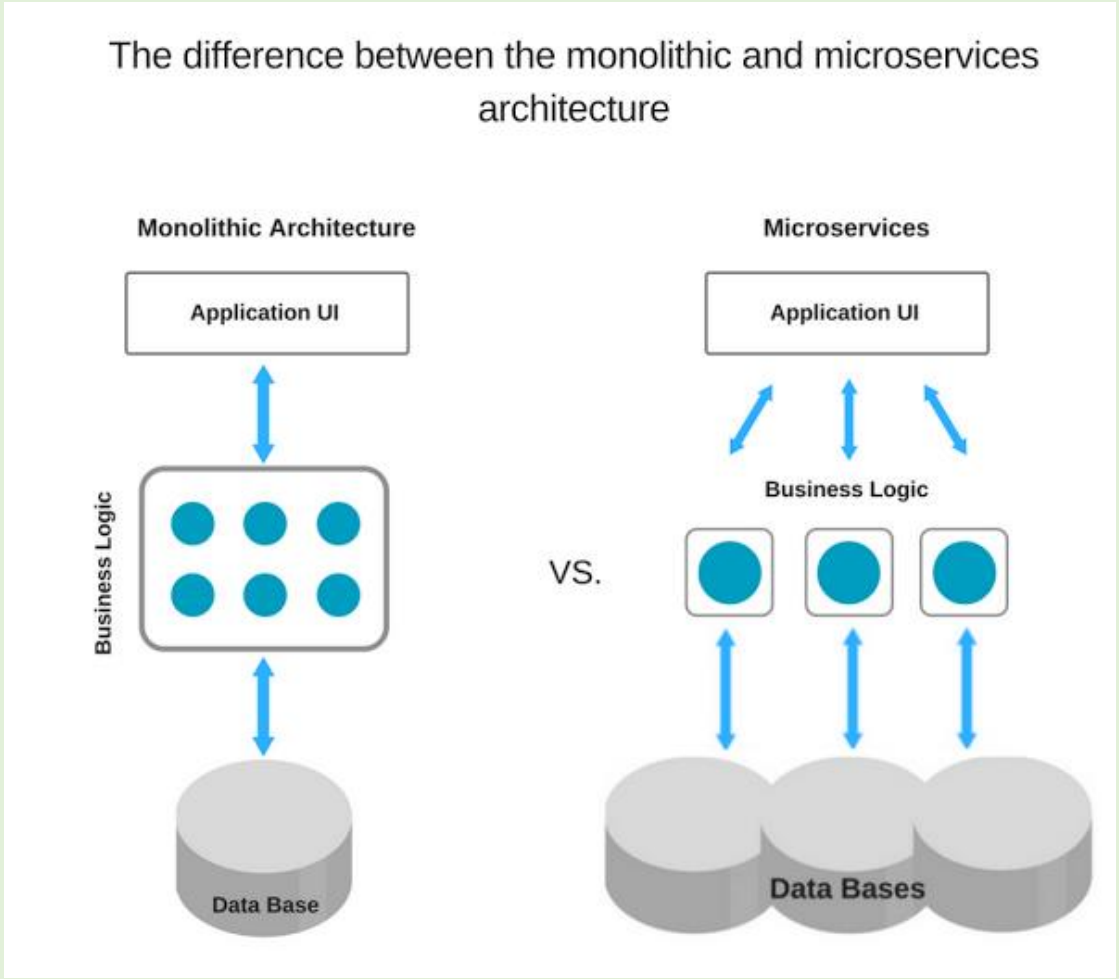


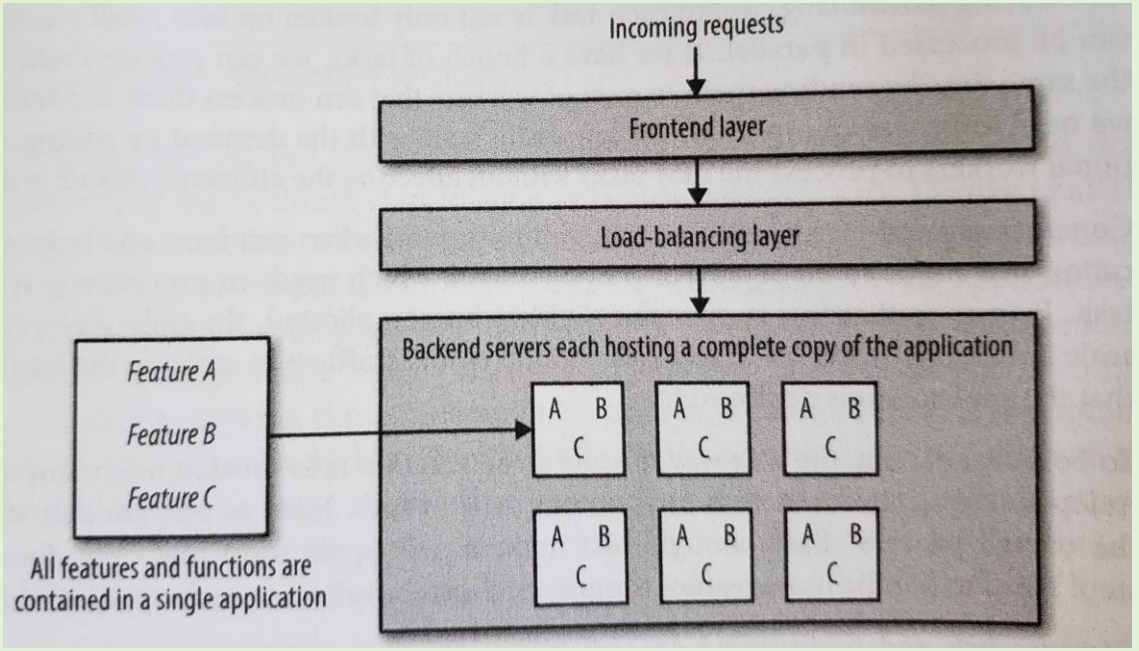
# What is a Microservice?

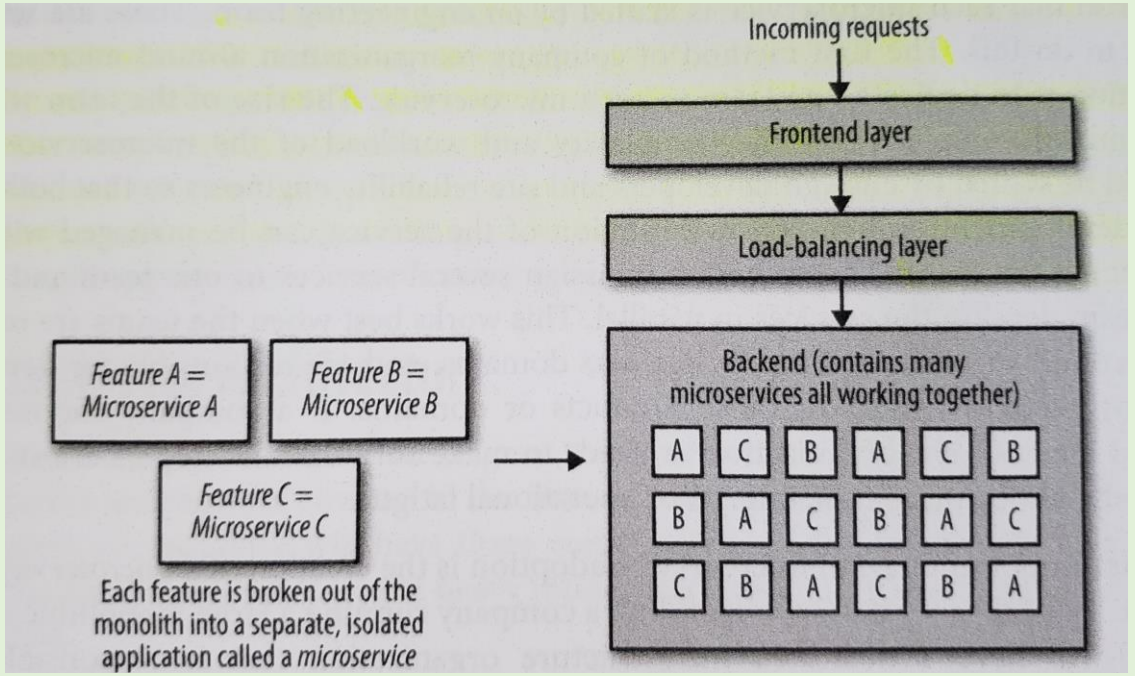
- Gartner defines a “Microservice as a **tightly scoped**, **loosely coupled**, , **strongly encapsulated, independently deployable,** and **independently scalable** application component.“

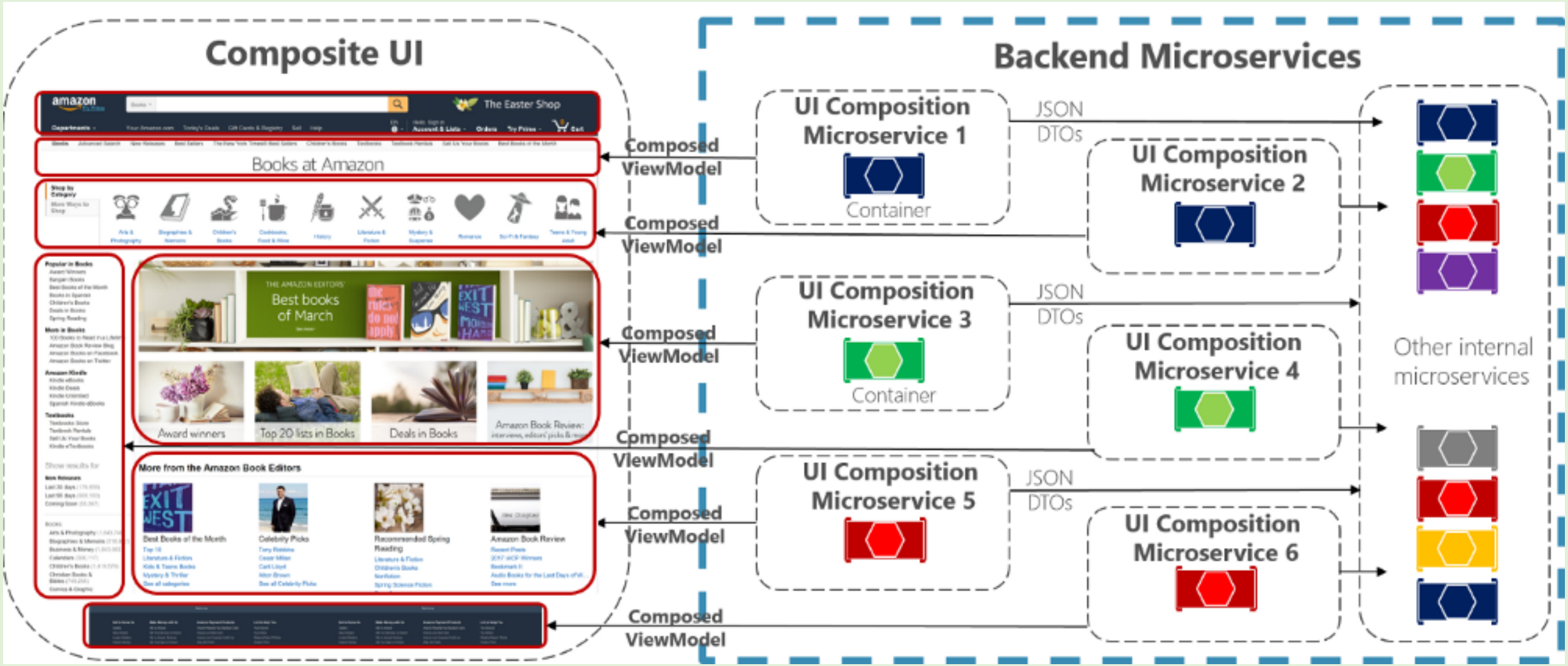
- Global Microservice Architecture Market anticipated accreting to US$ 33 Billion by 2023.


The difference between the monolithic and microservices architecture
Monolithic Architecture
Application UI
Business Logic
Data Base
VS.
Microservices
Application UI
Business Logic
Data Bases


Incoming requests
Frontend layer
Load-balancing layer
Backend servers each hosting a complete copy of the application
Feature A
Feature B
Feature C
All features and functions are contained in a single application


Incoming requests
Frontend layer
Load-balancing layer
Backend (contains many microservices all working together)
Feature A = Microservice A
Feature B = Microservice B
Feature C = Microservice C
Each feature is broken out of the monolith into a separate, isolated application called a microservice


Composite UI
Backend Microservices
Books at Amazon
Best books of March
Award winners
Top 20 lists in Books
Deals in Books
Amazon Book Review:
More from the Amazon Book Editors
Best Books of the Month
Celebrity Picks
Recommended Spring Reading
Amazon Book Review
Composed ViewModel
Composed ViewModel
Composed ViewModel
Composed ViewModel
Composed ViewModel
UI Composition Microservice 1
Container
UI Composition Microservice 2
UI Composition Microservice 3
Container
UI Composition Microservice 4
UI Composition Microservice 5
UI Composition Microservice 6
JSON DTOs
JSON DTOs
JSON DTOs
Other internal microservices

# Why - Advantages of Microservice

- Scalability
- Easier maintainability
- Easier Deployments
- Problem isolation
- Single Responsibility
- Separation of Concern
- Deep domain knowledge
- Polyglot programming

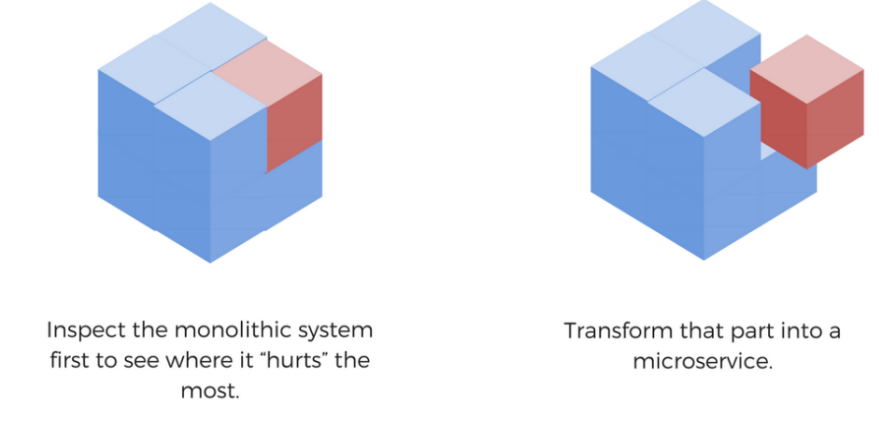
# Why not -Disadvantages of Microservice

- Deployment and interoperability
- Many programing languages
- Communication between services
- Harder to do integration tests
- Well thought architecture right from beginning
- Complexity

# How to begin with Microservice Architecture?

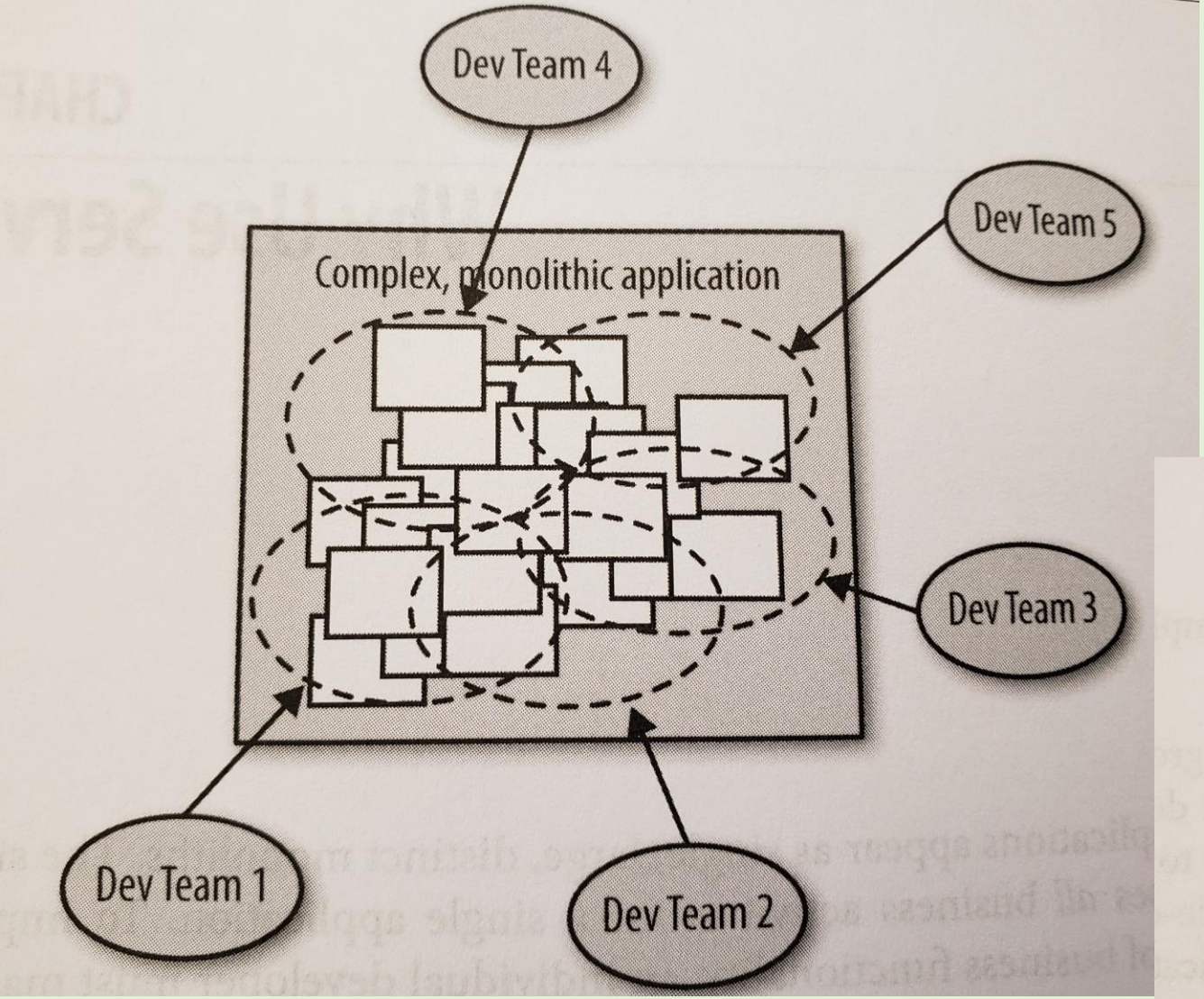
- Don’t switch from monolith to microservice all at once.

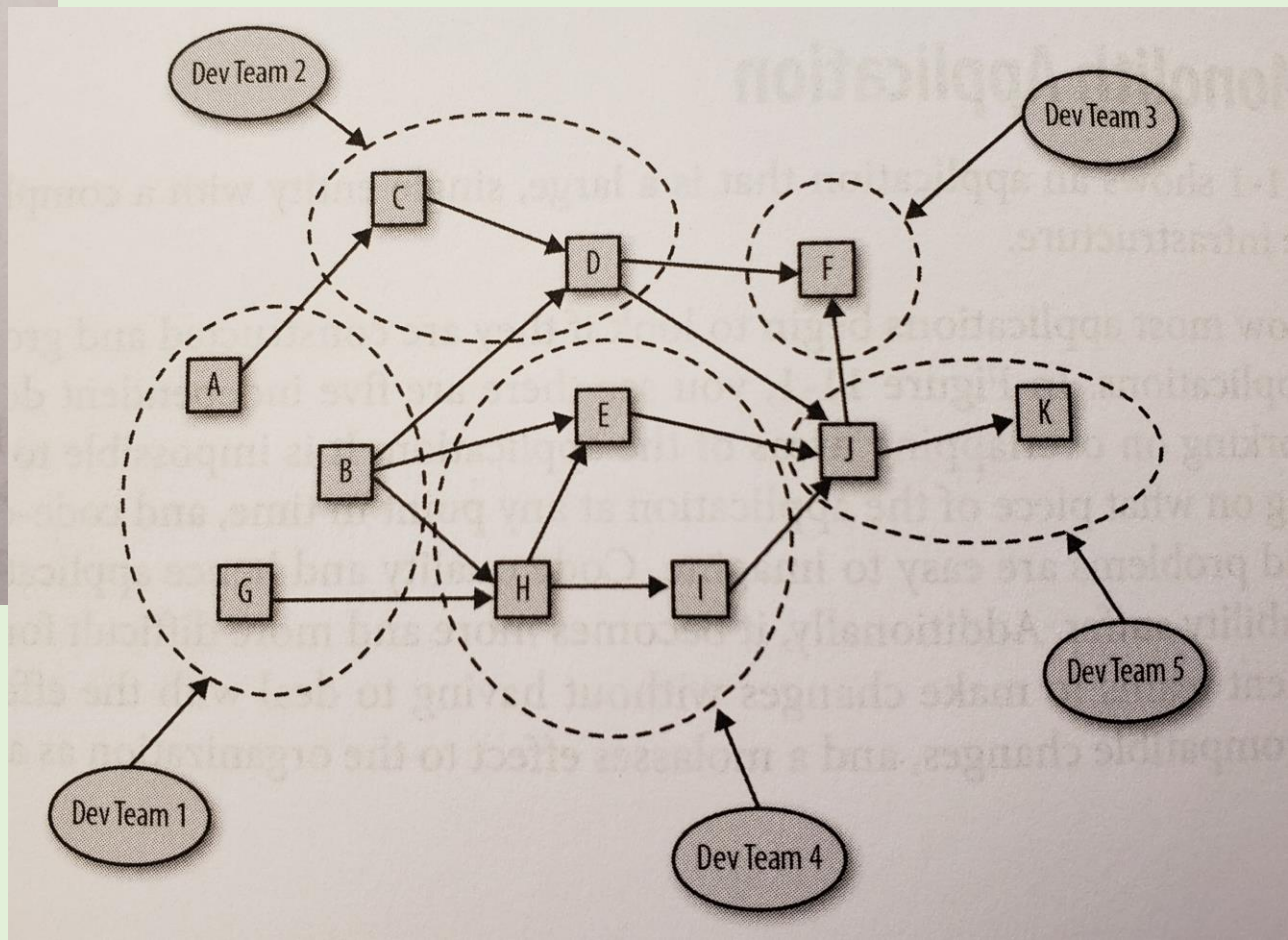

- Divide and conquer

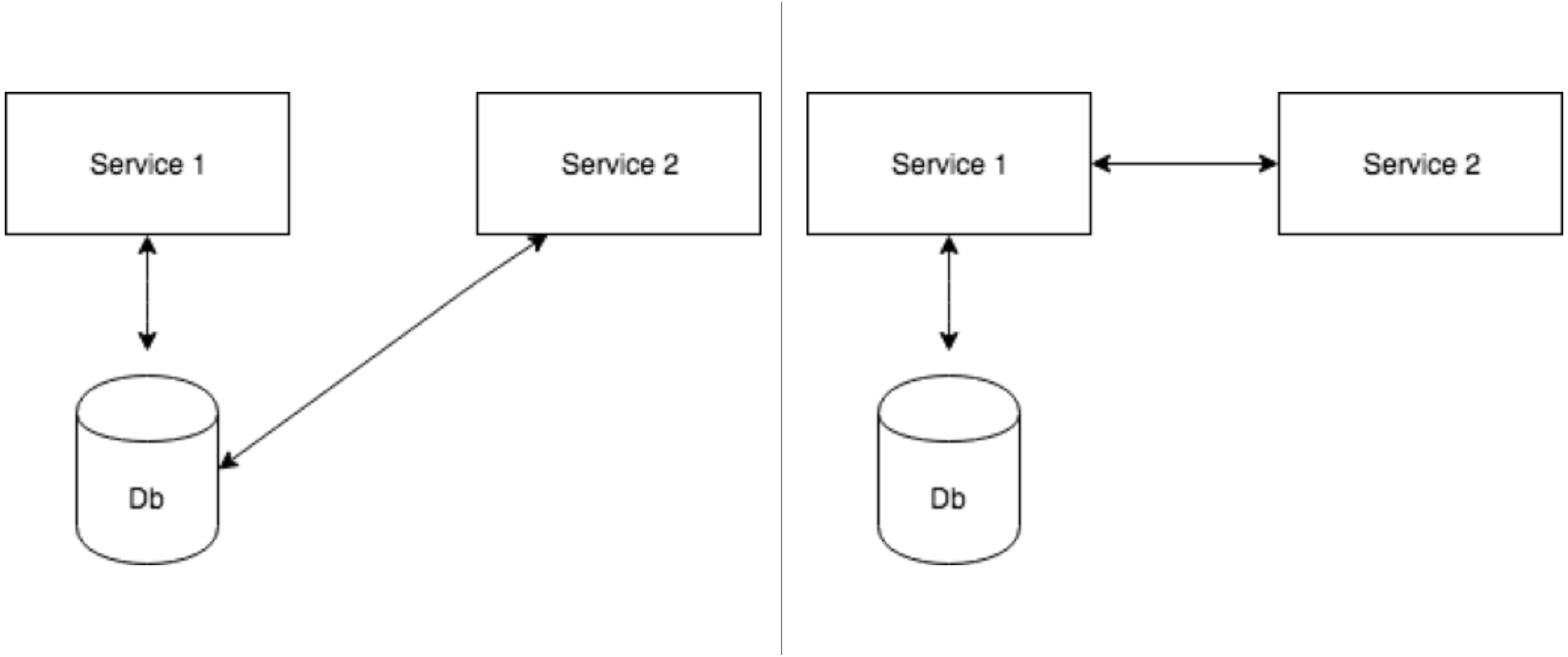


# How Big is a Microservice?

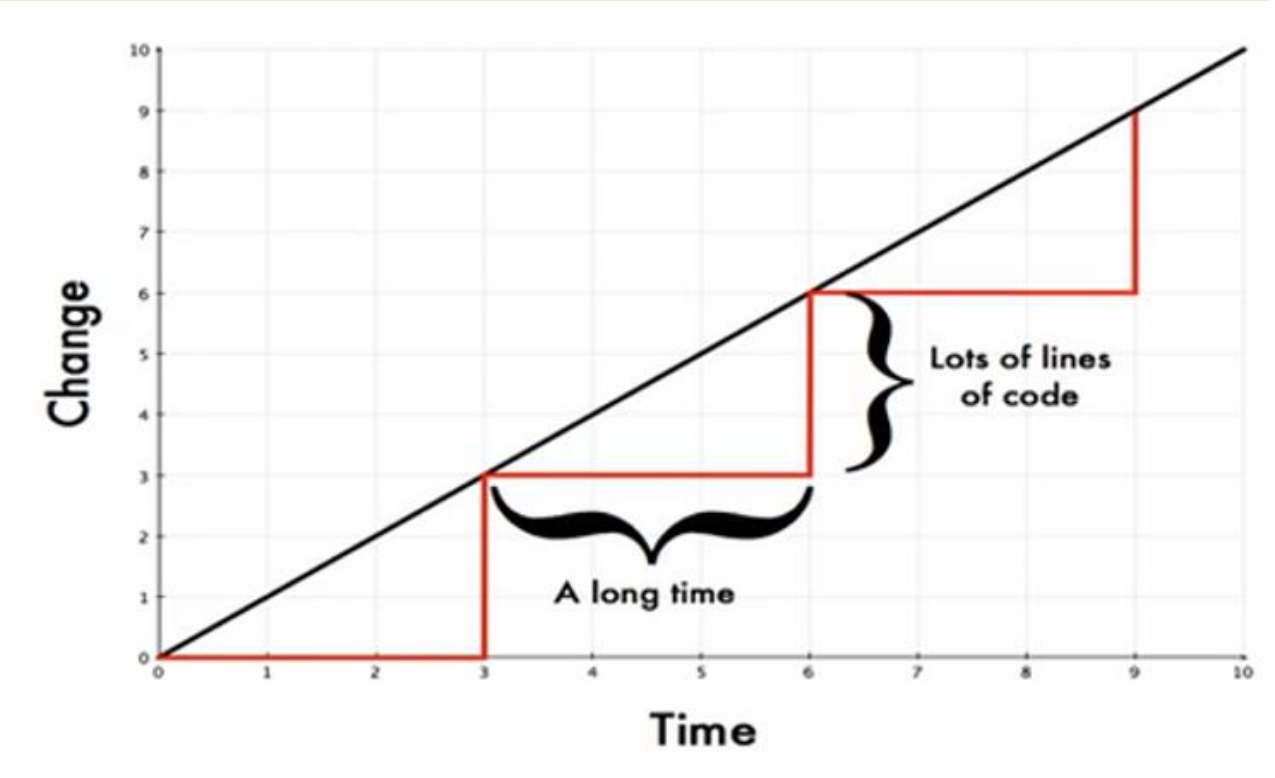
- Each microservice has a lean connection to every other microservice, usually through a RESTful API.
- Microservice boundaries are drawn around organizational capabilities — perhaps around particular development teams.

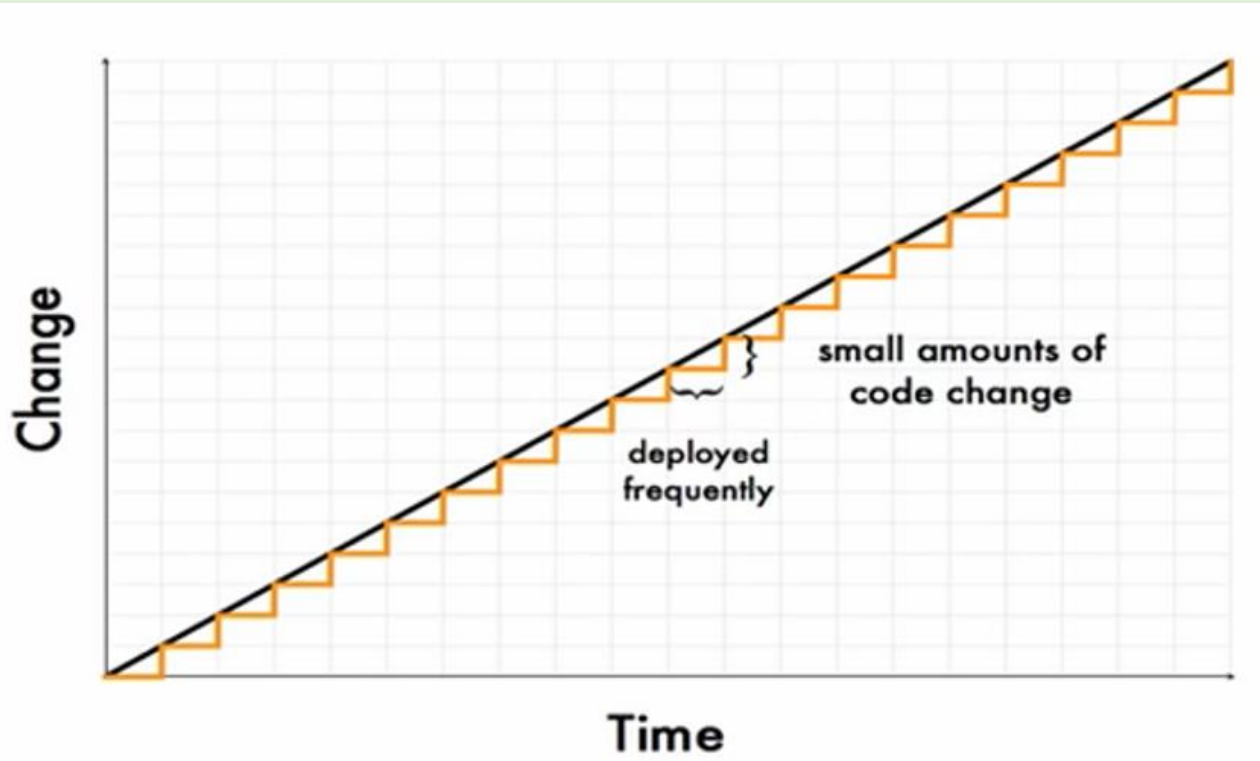

Dev Team 4
Dev Team 5
Complex, monolithic application
Dev Team 3
Dev Team 1
Dev Team 2


Dev Team 2
Dev Team 3
C
D
F
A
E
K
B
J
G
H
I
Dev Team 5
Dev Team 1
Dev Team 4


Service 1
Service 2
Db
Service 1
Service 2
Db

# Delivery Cycles – Monolith vs Microservice

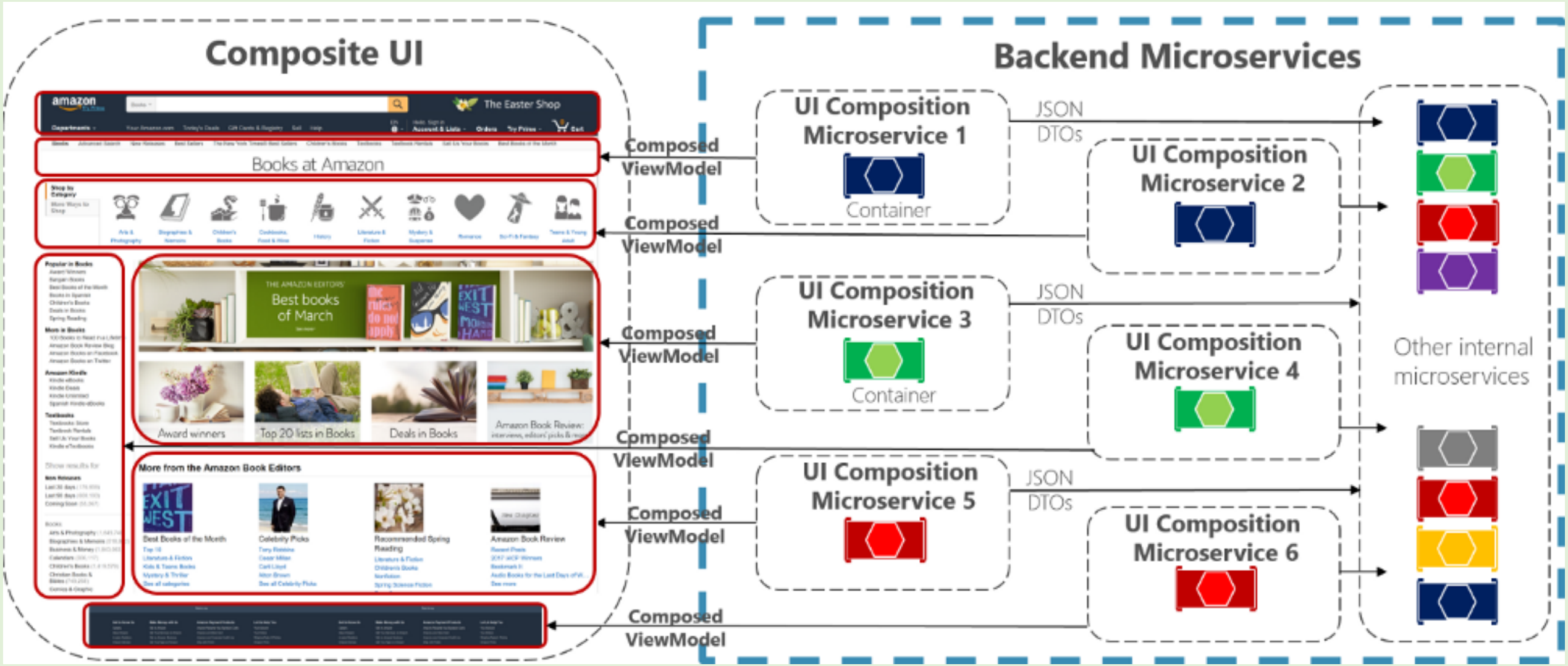


# DevOps Practices a must for Microservice

- Configuration Management
- Release Management
- Continuous Integration
- Continuous Deployment
- Infrastructure as Code
- Test Automation
- Application Performance Monitoring

# DevOps Metrics with Microservice


Composite UI
Backend Microservices
Books at Amazon
Best books of March
Award winners
Top 20 lists in Books
Deals in Books
Amazon Book Review:
More from the Amazon Book Editors
Best Books of the Month
Celebrity Picks
Recommended Spring Reading
Amazon Book Review
Composed ViewModel
Composed ViewModel
Composed ViewModel
Composed ViewModel
Composed ViewModel
UI Composition Microservice 1
Container
UI Composition Microservice 2
UI Composition Microservice 3
Container
UI Composition Microservice 4
UI Composition Microservice 5
UI Composition Microservice 6
JSON DTOs
JSON DTOs
JSON DTOs
Other internal microservices

# Thank You

**Vidya Vrat Agarwal**

www.MyPassionFor.Net | **@dotNetAuthor**

**https://www.linkedin.com/in/vidyavrat/**