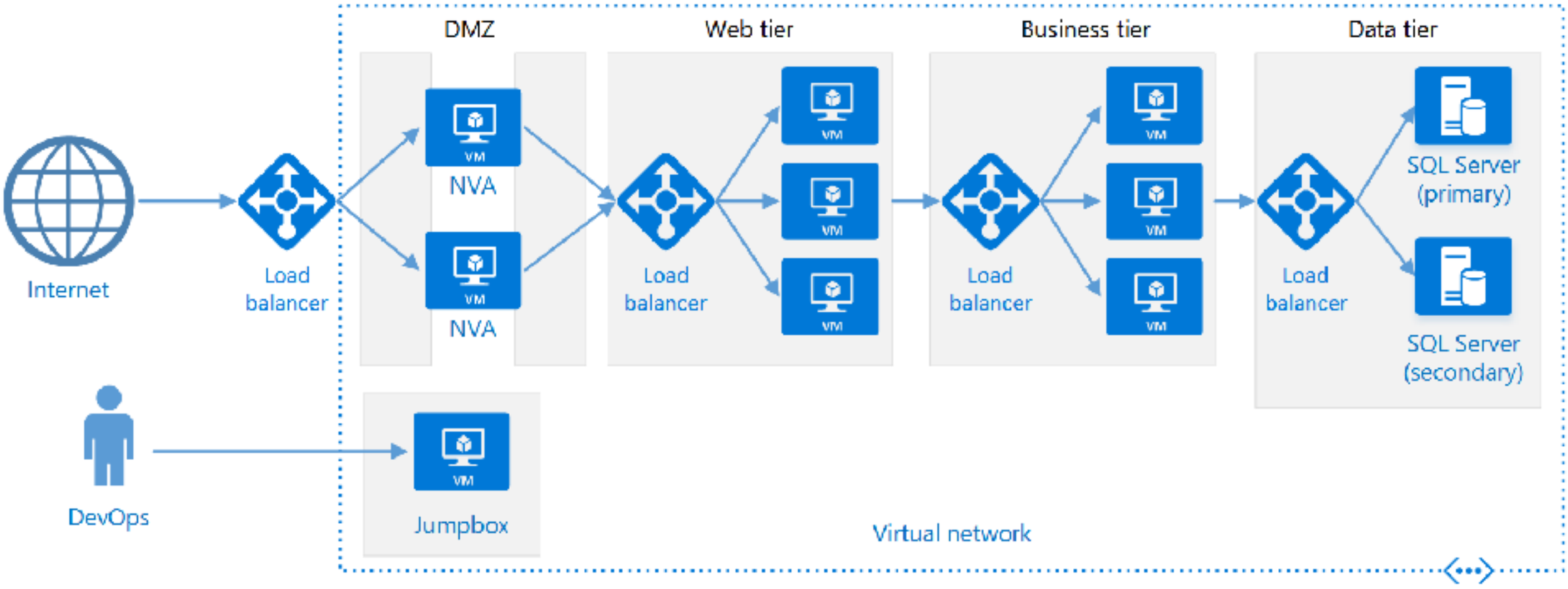
Microservices

Introduction

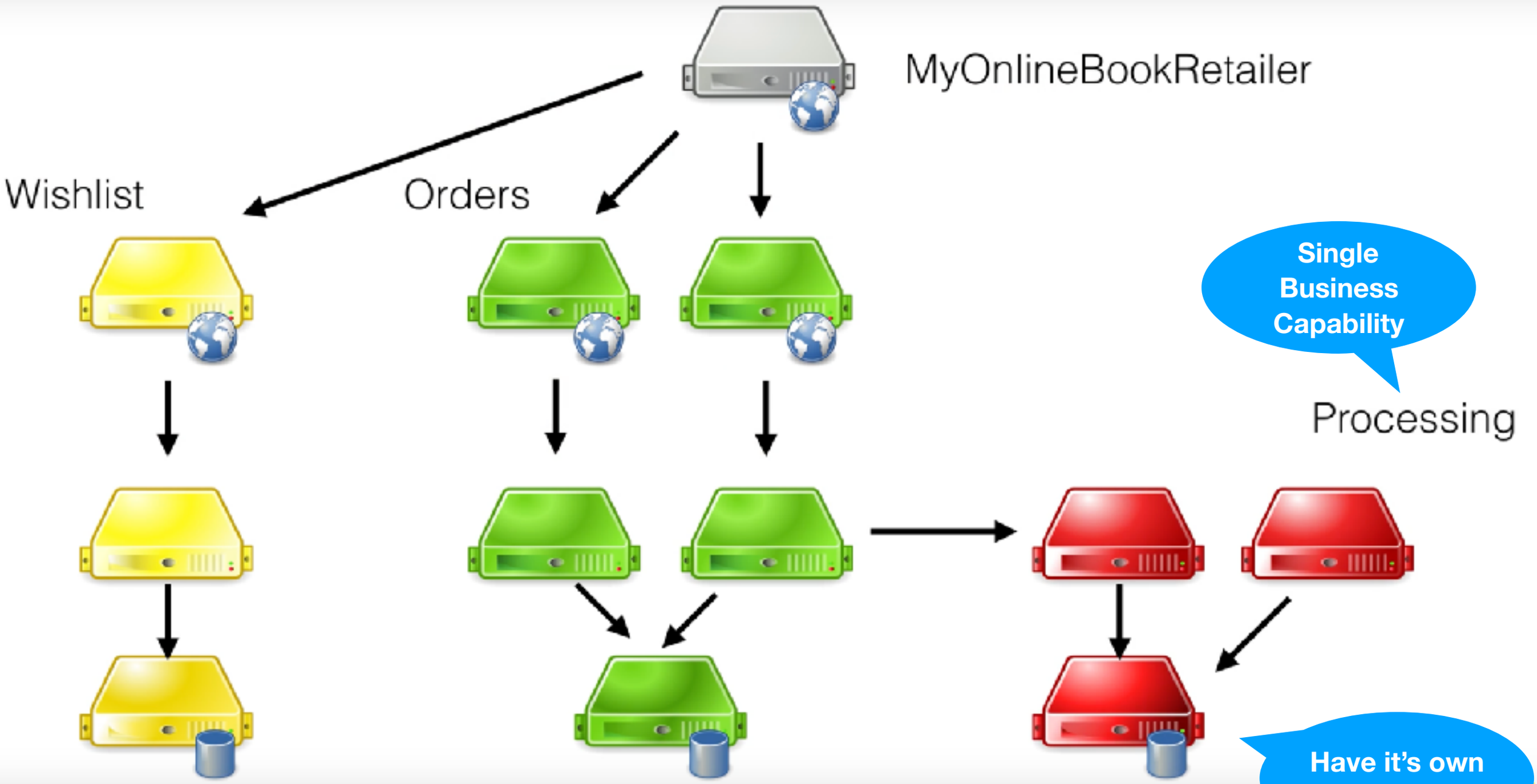
Microservices

- Definition
    - Self-contained process.
    - Provide unique **Business Capability**.
- Architecture
    - Design for failure.
    - Loosely coupling.
    - Communication Stateless, Easy to scale.
    - All services should not break if any services broke.

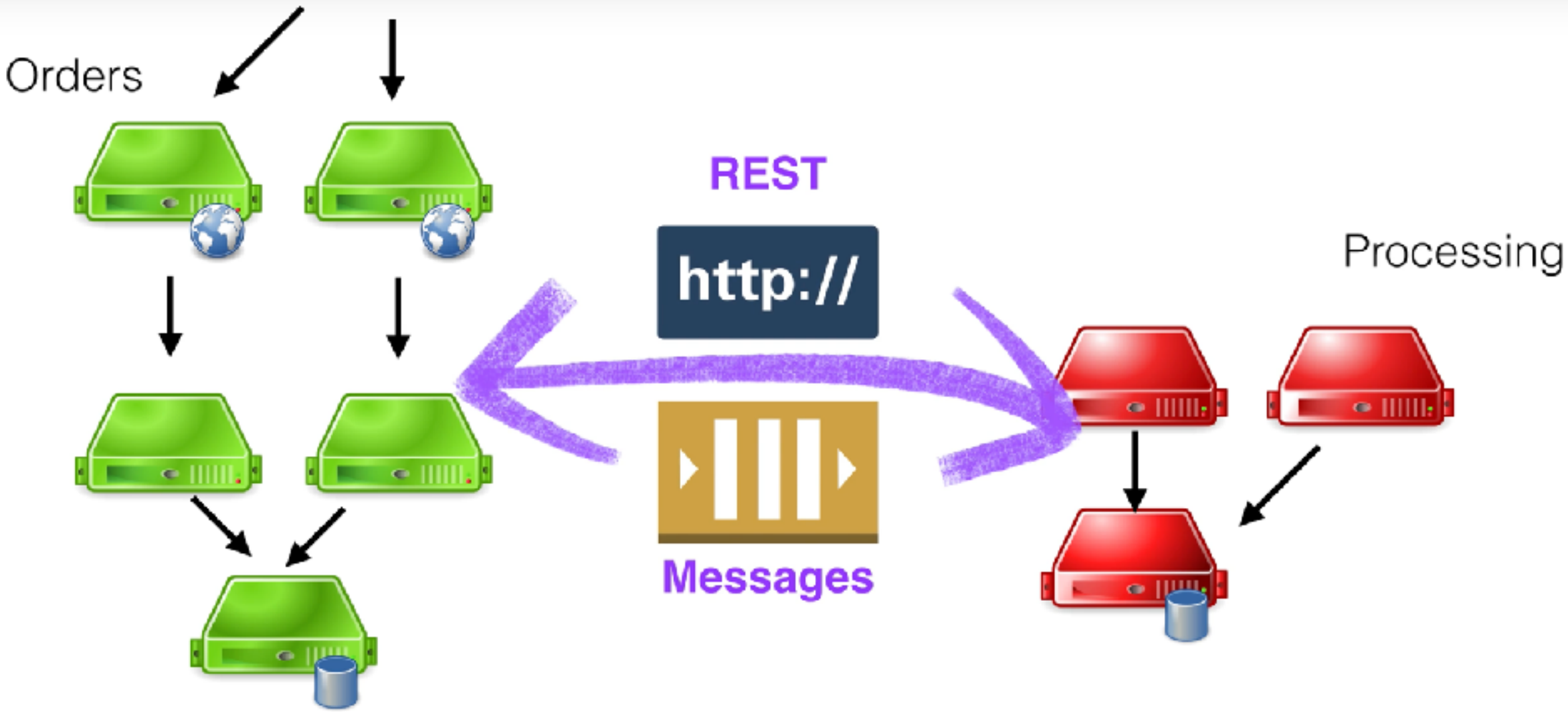
What’s not Microservices



Example: e-Commerce

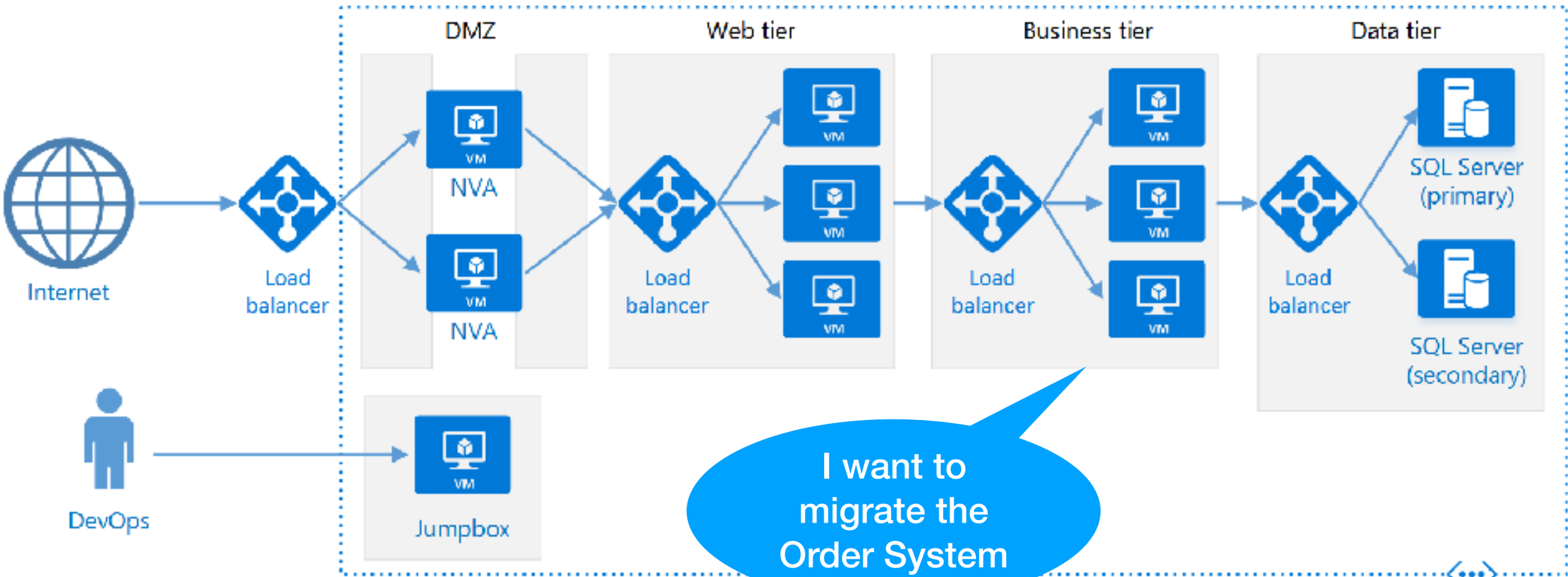


Communication

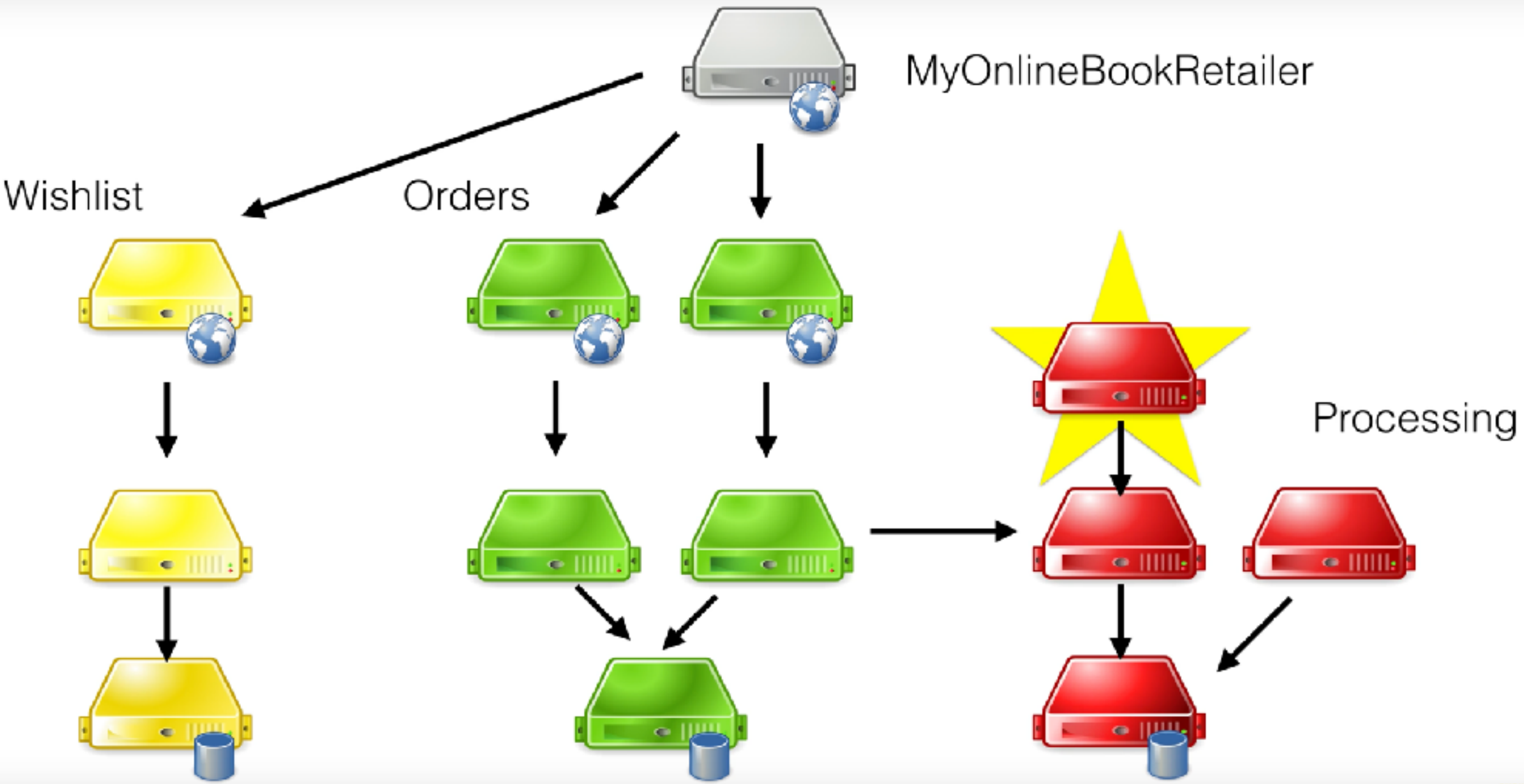


Any Language, Platform, but Same Protocol

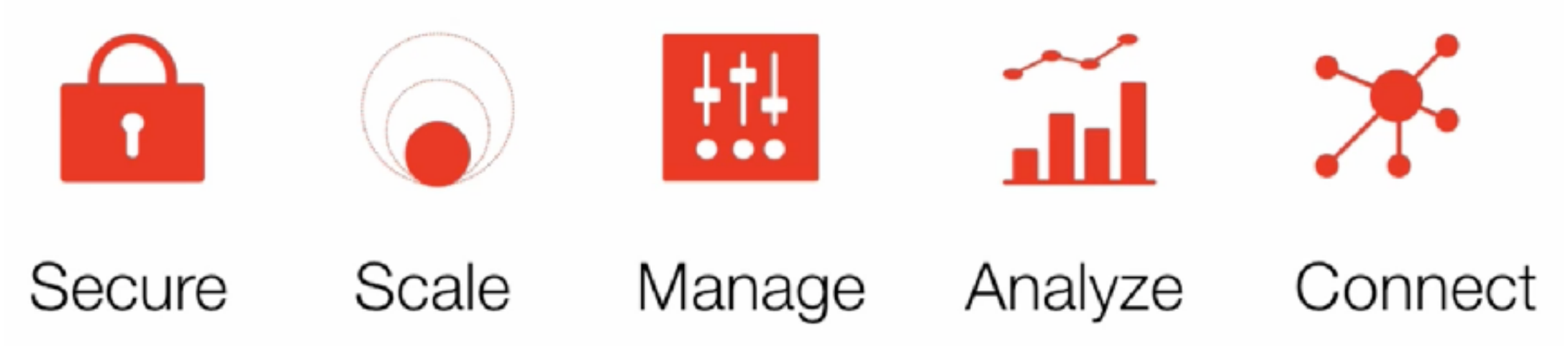
Replace? Migration? M/A?



Blue/Green Deployment on a Single Service is not a Big Deal



API Gateway, Why?



API Gateway

API
Gateway

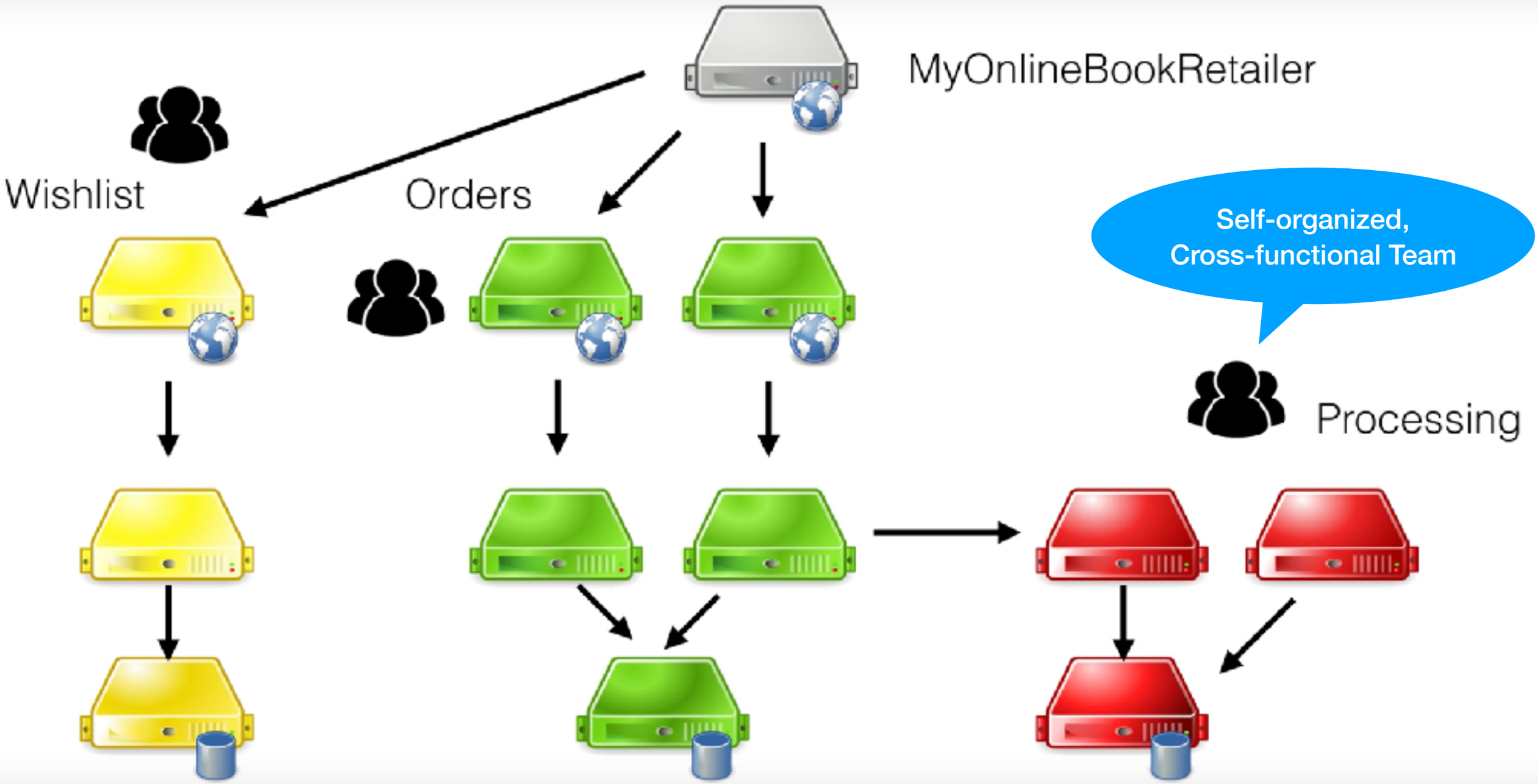
- Proxy
  - Transport Security
  - Monitoring (SLA, Performances, etc.)
  - Paths
- Gateway
  - Orchestration (combine some API together)
  - Mediation (format transformation)
  - Message Security
  - Authorization
  - DoS
- All of these features should be able to add on required.

Orders
Cluster

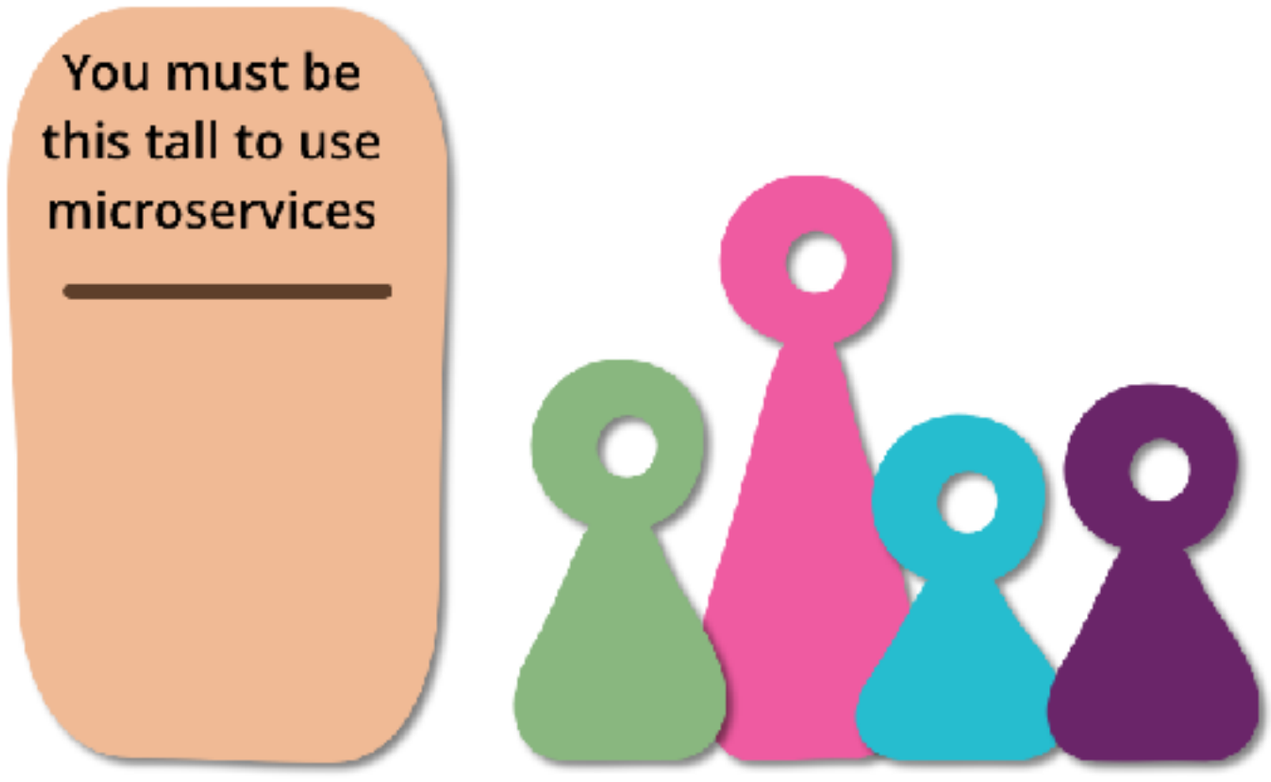
Inventory
Cluster

Credit Cards
Cluster

Good Team make Good Microservices



Before you goto Microservices



- Rapid provisioning
- Basic Monitoring
- Rapid application deployment

https://martinfowler.com/bliki/MicroservicePrerequisites.html

When should I Use Them?

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.

https://martinfowler.com/microservices/