Microservices – The Easy Way is the Wrong Way

Microservices – The Easy Way is the Wrong Way

Goal

Show that
Microservice Architecture takes
more consideration than you
may expect

Microservices – The Easy Way is the Wrong Way

Agenda

- What are microservices
- Why should I use them
- When should I NOT use them
- Challenges
- Decomposition (high-level)

Microservices – The Easy Way is the Wrong Way

What are microservices

Microservices – The Easy Way is the Wrong Way

What are microservices

"A microservice is simply a tiny monolith with a tight focus on a domain. It's not about size. It's about focus."

Microservices – The Easy Way is the Wrong Way

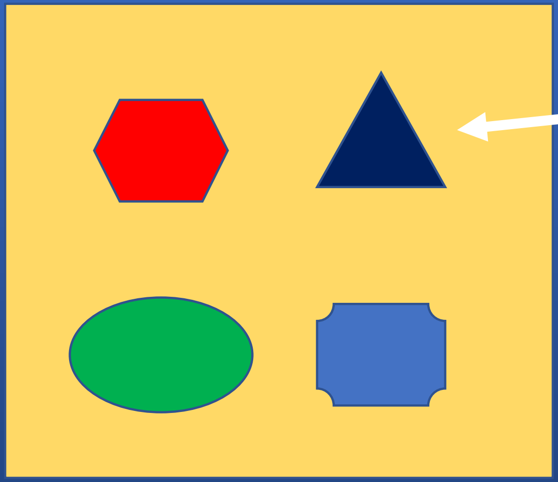
What are microservices

- ✓ Independent applications
- ✓ Communicate over network
- ✓ Has single responsibility
- ✓ Owned by one team
- ✓ Individually deployable
- ✓ Architecture style

Microservices – The Easy Way is the Wrong Way

What are microservices

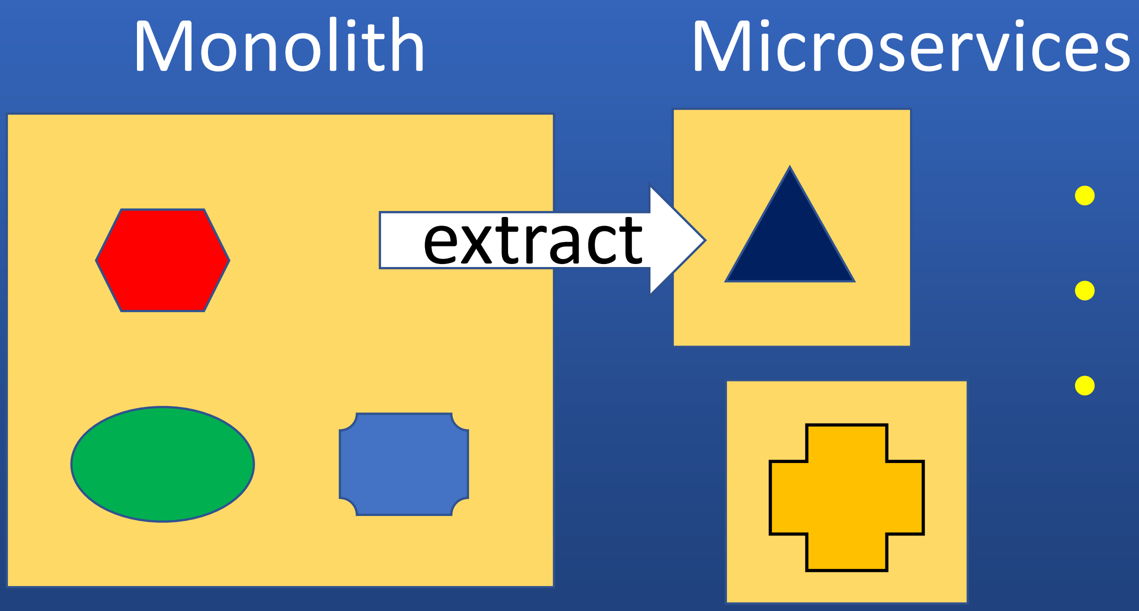
Monolith



Independent Functionality

Microservices – The Easy Way is the Wrong Way

What are microservices

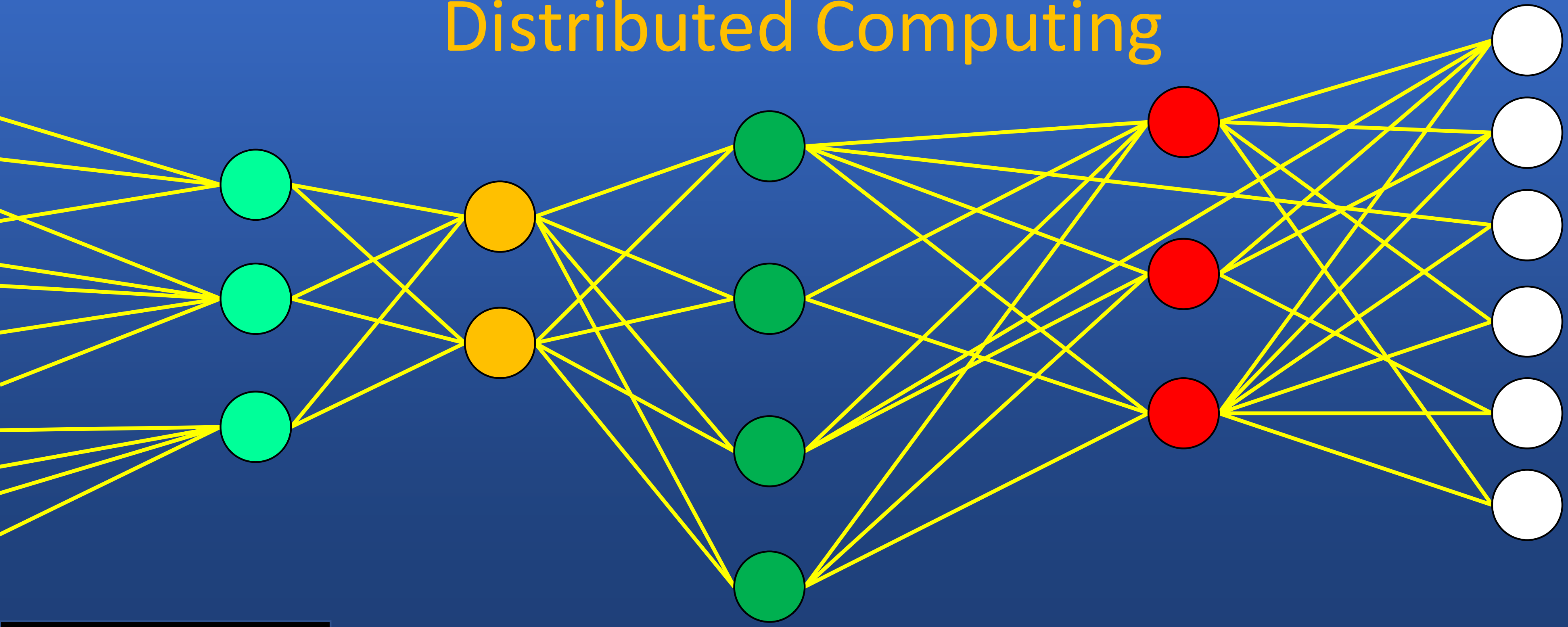


- Move functionality
- Add functionality
- Rare total migration

Microservices – The Easy Way is the Wrong Way

The microservice architecture is an implementation of

Distributed Computing



Microservices – The Easy Way is the Wrong Way

Why should I use microservices

To deliver business functionality

- ✓ Quicker
- ✓ Safer
- ✓ Stability

Microservices – The Easy Way is the Wrong Way

Why should I use microservices

Team Autonomy
Service Autonomy
Scalability
Fault Isolation

Team Autonomy

- Best language for the job
- Focused development
- Agile/Kanban work well

Microservices – The Easy Way is the Wrong Way

Why should I use microservices

Team Autonomy
Service Autonomy
Scalability
Fault Isolation

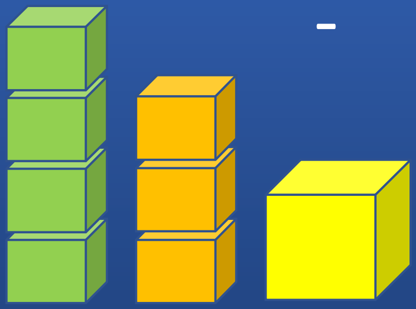
Service Autonomy

- Single Responsibility
- Deployable without other services

Microservices – The Easy Way is the Wrong Way

Why should I use microservices

Team Autonomy
Service Autonomy
Scalability
Fault Isolation



Scalability

- Scale independent of others
- Choice of servers

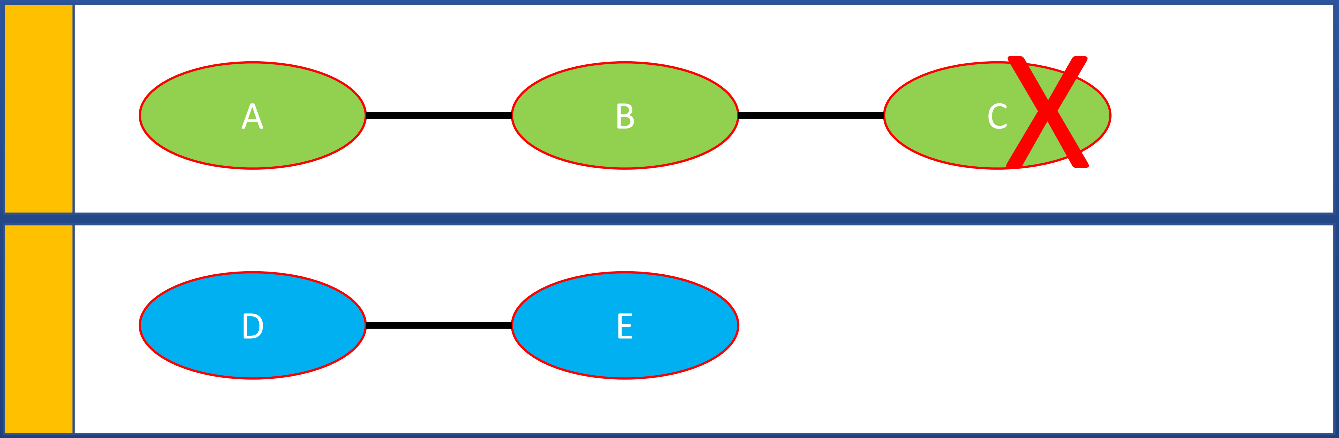
Microservices – The Easy Way is the Wrong Way

Why should I use microservices

Team Autonomy
Service Autonomy
Scalability
Fault Isolation

Fault Isolation

- Your problems aren't my problems
- "Stay in your lane bro"



Microservices – The Easy Way is the Wrong Way

Who uses microservices

amazon
ebay
NETFLIX

Microservices – The Easy Way is the Wrong Way

Why should I NOT use microservices

Microservices – The Easy Way is the Wrong Way

Why should I NOT use microservices

THEY'RE HARD!!!

But so is good programming…

This Photo by Unknown Author is licensed under CC BY-SA-NC

Microservices – The Easy Way is the Wrong Way

Why should I NOT use microservices

- Monolith is too small
- Too much coupling – need refactoring first
- Team unwilling or not ready
- No support structure
- High cost of time and money

Microservices – The Easy Way is the Wrong Way

Challenges to making microservices

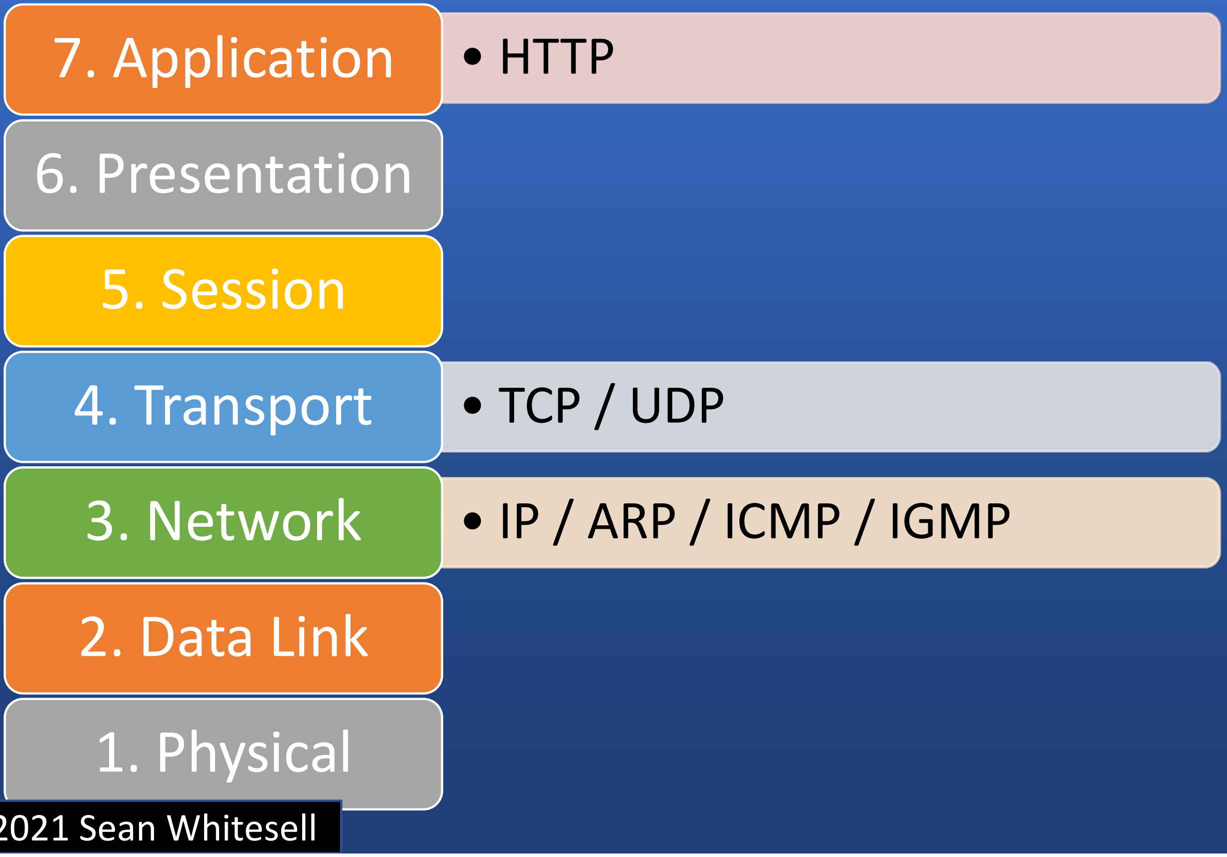
Microservices – The Easy Way is the Wrong Way

Challenges to making microservices

Expectation		Reality
Cheap		Cost time, money, learning, investing
HTTP to IP address		Messaging; RPC; sync vs async; pub/sub
Copy files to servers		CI / CD pipeline per Microservice
Single Repo		Should use Repo per Microservice
Network not my job		Need to know about OSI Model; Layer 4 and 7
Security non-issue		TLS for service communication layer; OAuth for application layer
Common Code		Sharing code keeps services tightly coupled

Microservices – The Easy Way is the Wrong Way

Challenges to making microservices

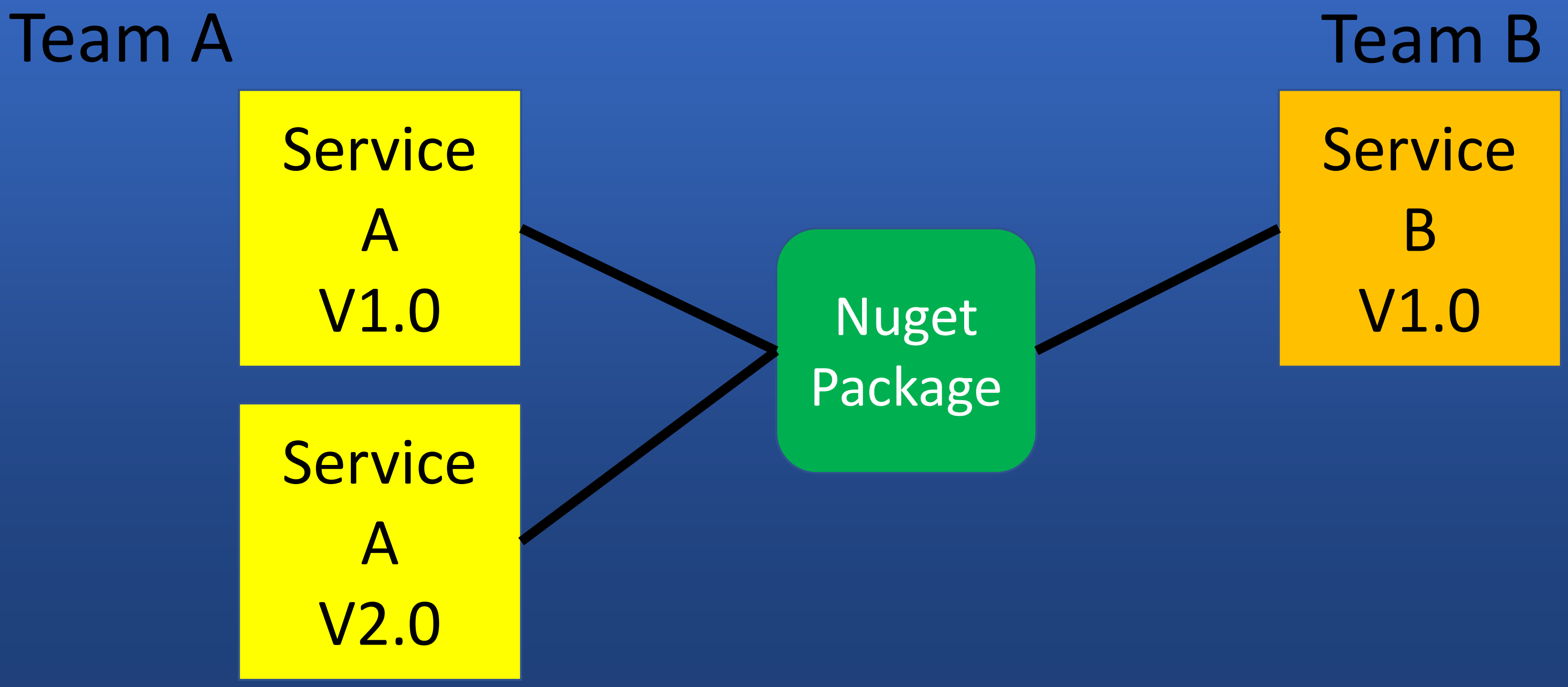


7. Application	• HTTP
6. Presentation	
5. Session	
4. Transport	• TCP / UDP
3. Network	• IP / ARP / ICMP / IGMP
2. Data Link	
1. Physical	

Open Systems Interconnection Model

Microservices – The Easy Way is the Wrong Way

Challenges to making microservices



Team A

Service A V1.0

Service A V2.0

Nuget Package

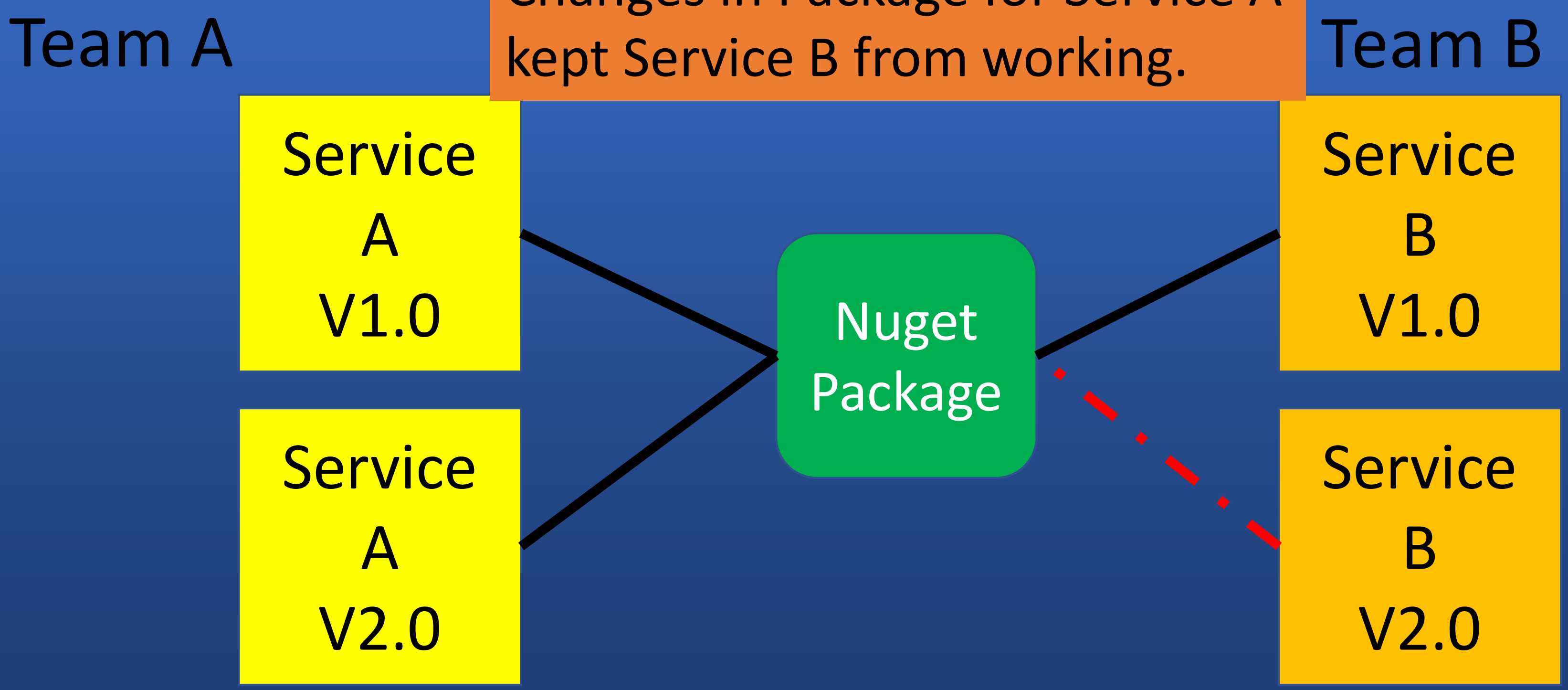
Team B

Service B V1.0

Microservices – The Easy Way is the Wrong Way

Challenges to making microservices

Changes in Package for Service A kept Service B from working.



Team A

Service A V1.0

Service A V2.0

Nuget Package

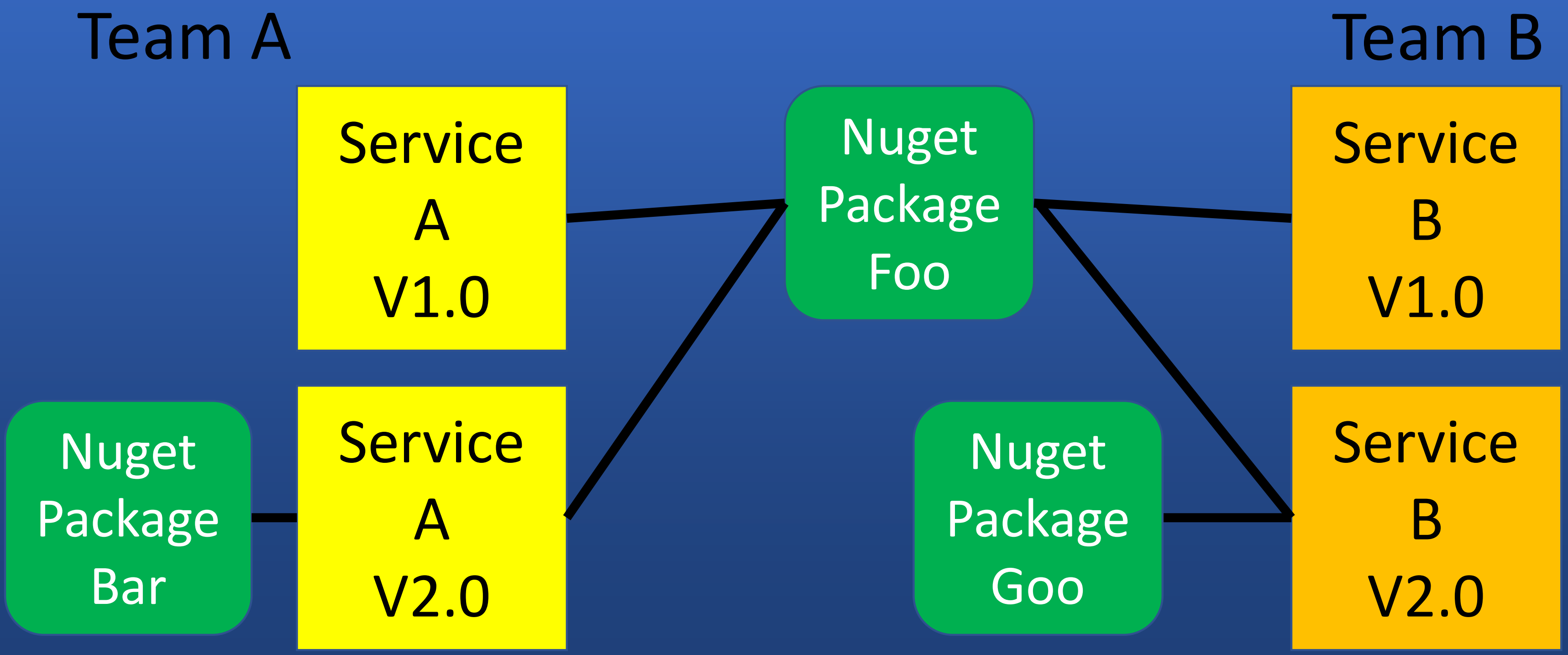
Team B

Service B V1.0

Service B V2.0

Microservices – The Easy Way is the Wrong Way

Challenges to making microservices



Team A

Team B

Service A V1.0

Service A V2.0

Nuget Package Bar

Nuget Package Foo

Nuget Package Goo

Service B V1.0

Service B V2.0

Microservices – The Easy Way is the Wrong Way

Challenges to making microservices

Nuget Package Common

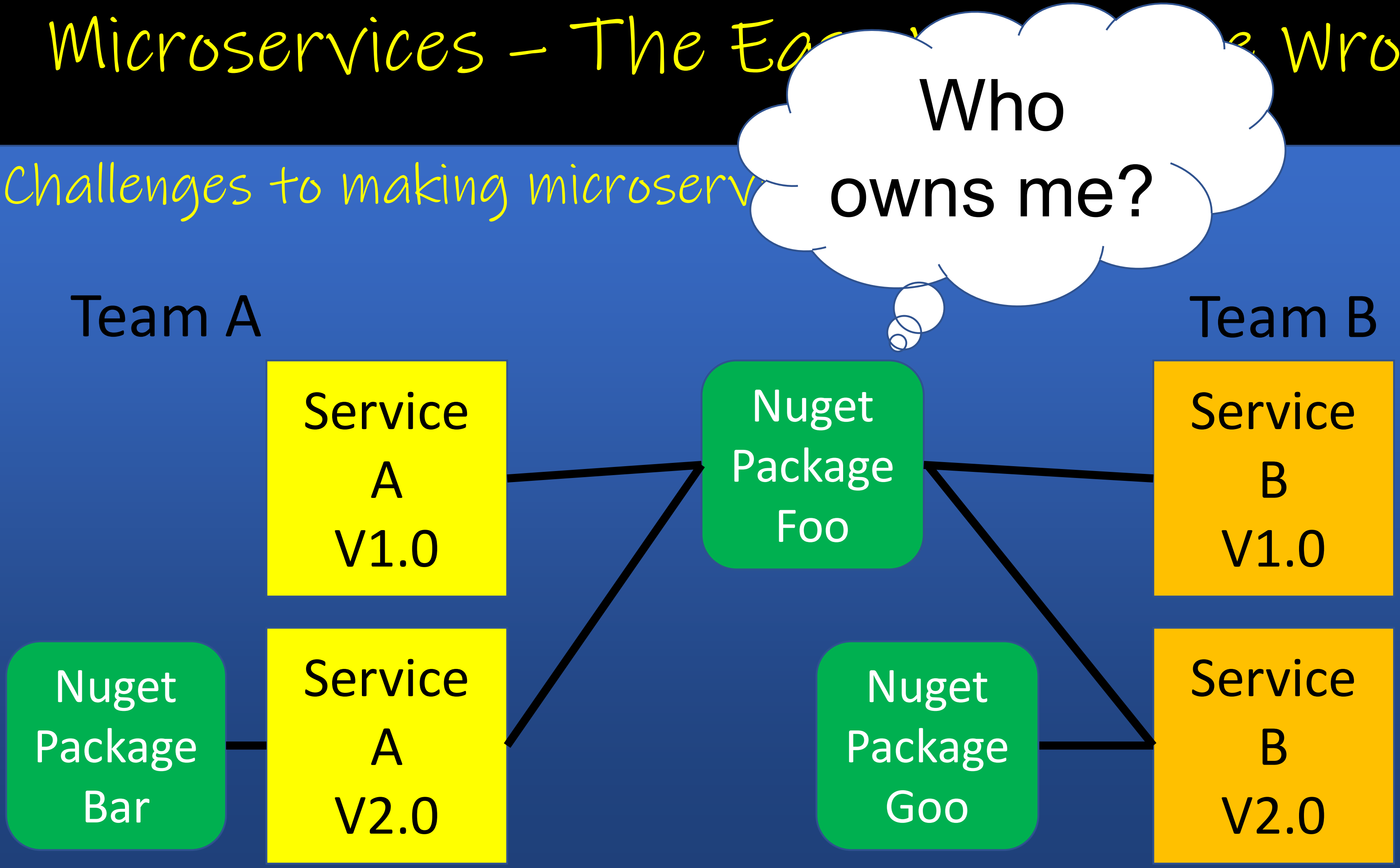
Separate code that changes to their own package for the service that needs it.

Nuget Package Svc A

Nuget Package Svc B

Microservices – The Ea[illegible] Wrong Way

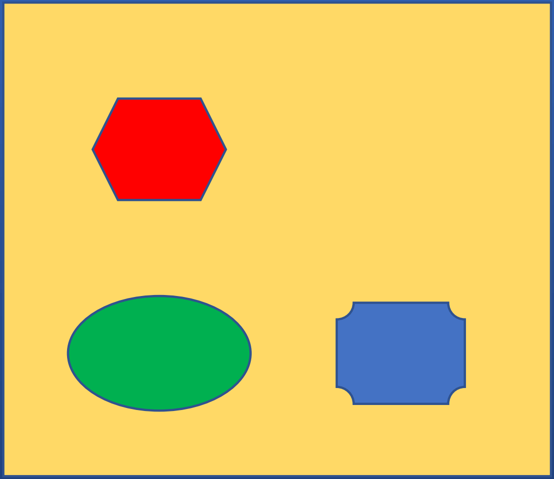
Challenges to making microserv[illegible]



Microservices – The Easy Way is the Wrong Way

Challenges to making microservices

- Distributed monolith



Monolith

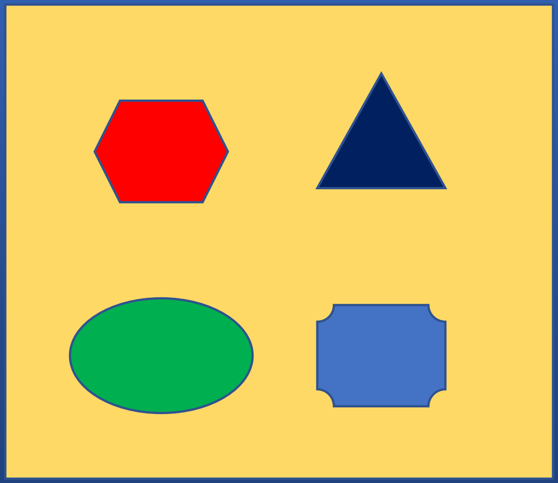
~~Microservice~~ Monolith 2

This Photo by Unknown Author is licensed under CC BY-SA

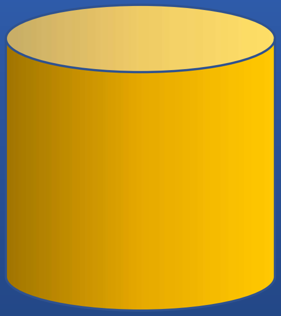
Microservices – The Easy Way is the Wrong Way

Challenges to making microservices

- Decentralized Data



Monolith

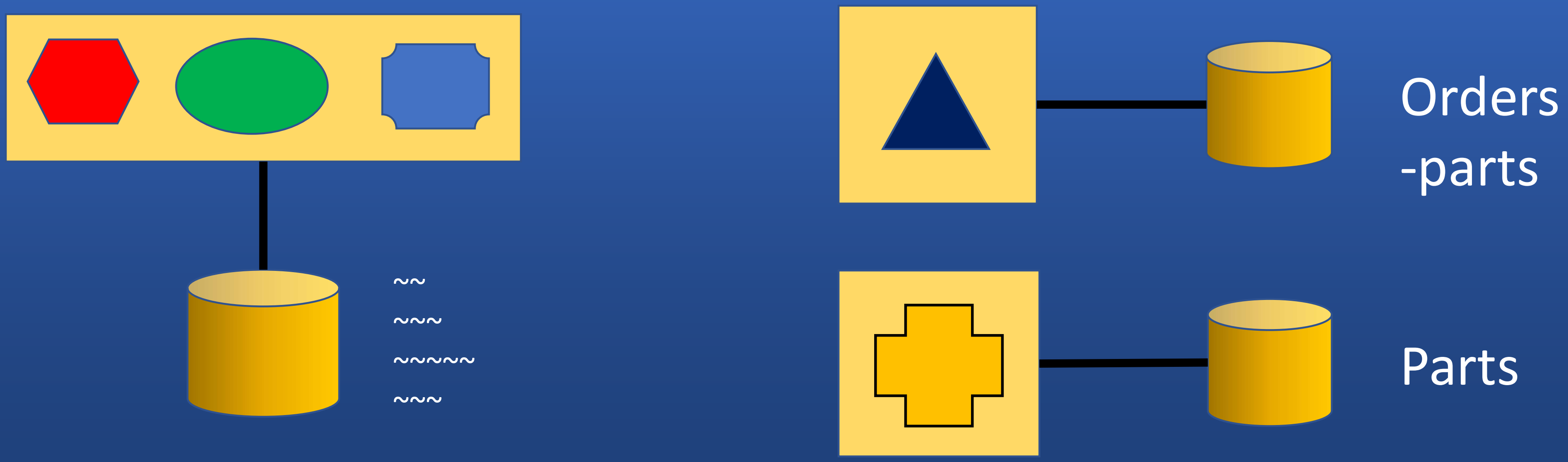


Accounts
Customers
Orders
Order History
Parts
Parts1
Shipping History
Logging

Microservices – The Easy Way is the Wrong Way

Challenges to making microservices

- Decentralized Data



~~

~~~

~~~~~

~~~

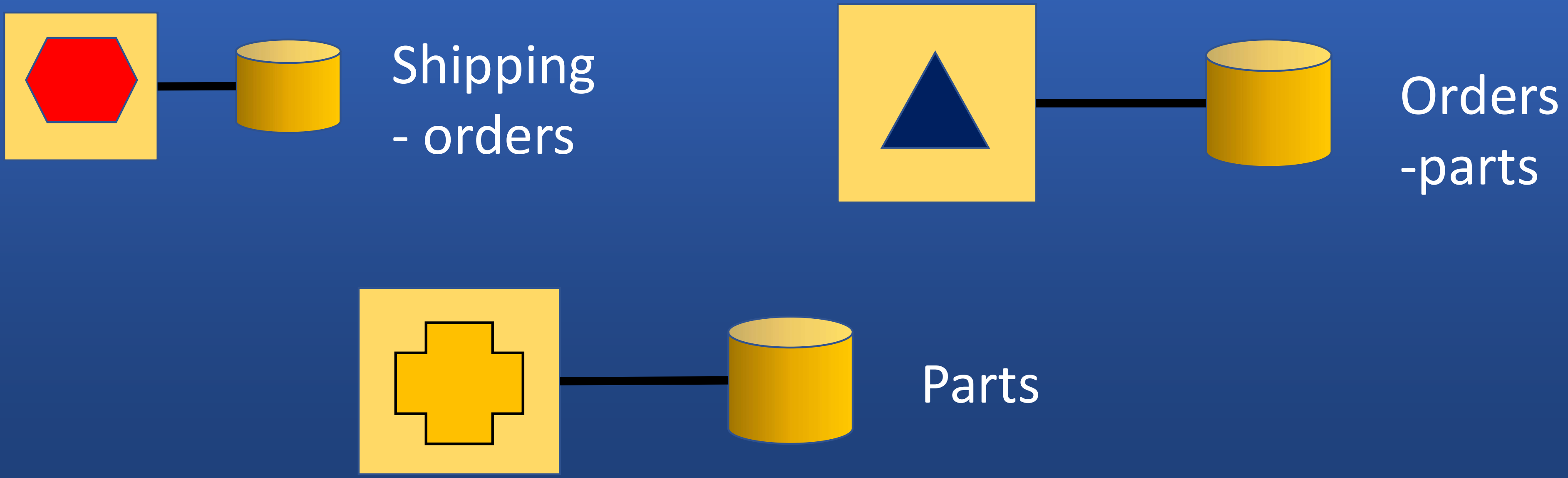
Orders
-parts

Parts
~~~

Microservices – The Easy Way is the Wrong Way

Challenges to making microservices

- Decentralized Data



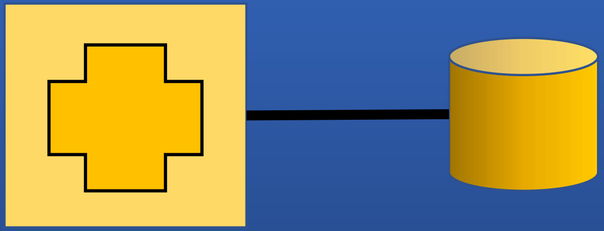
Shipping
- orders

Orders
-parts

Parts

Microservices – The Easy Way is the Wrong Way

Challenges to making microservices

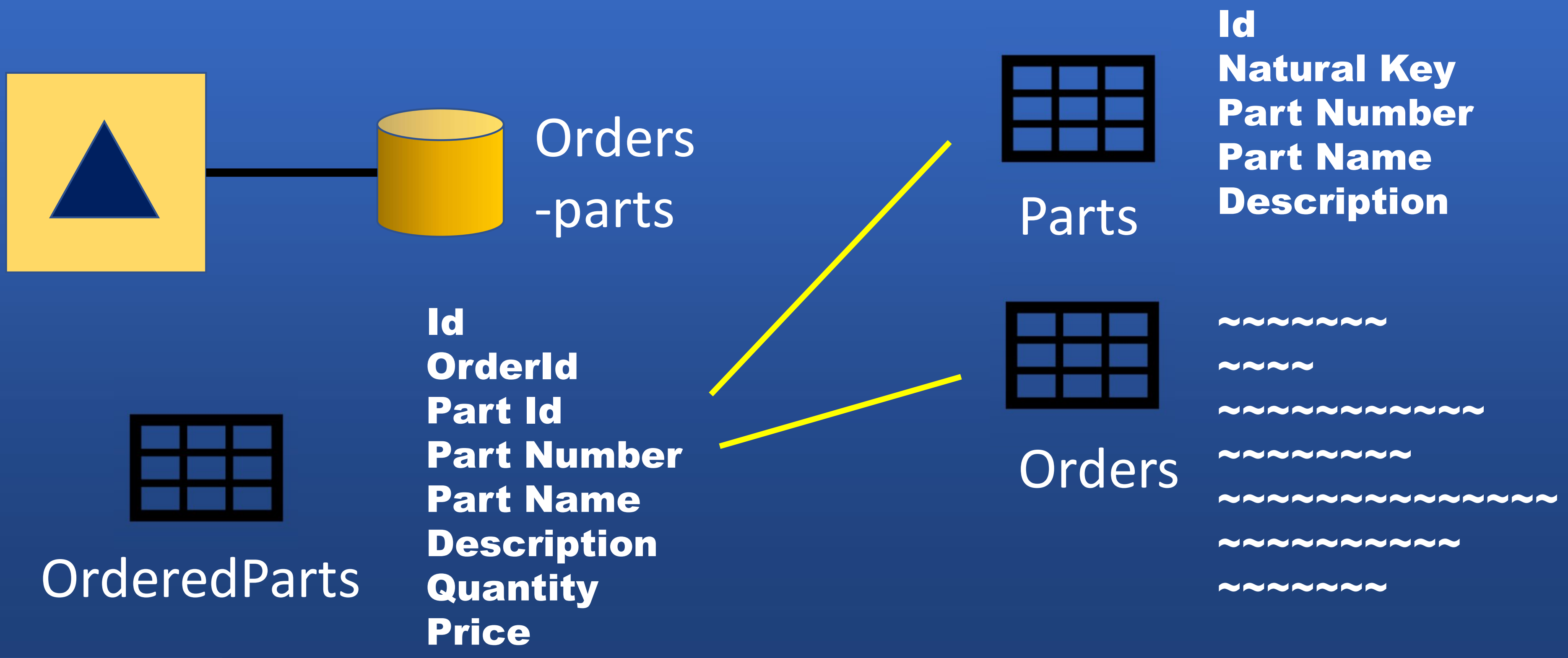


Parts

Id
Natural Key
Part Number
Part Name
Category
Classification
Description
Size Description
Color
Container
Manufacturer
Mfg Part Number
Mfg Date
Replaces Part Number
Discontinued Date

Microservices – The Easy Way is the Wrong Way

Challenges to making microservices



Orders
-parts

Parts

Id
Natural Key
Part Number
Part Name
Description

OrderedParts

Id
OrderId
Part Id
Part Number
Part Name
Description
Quantity
Price

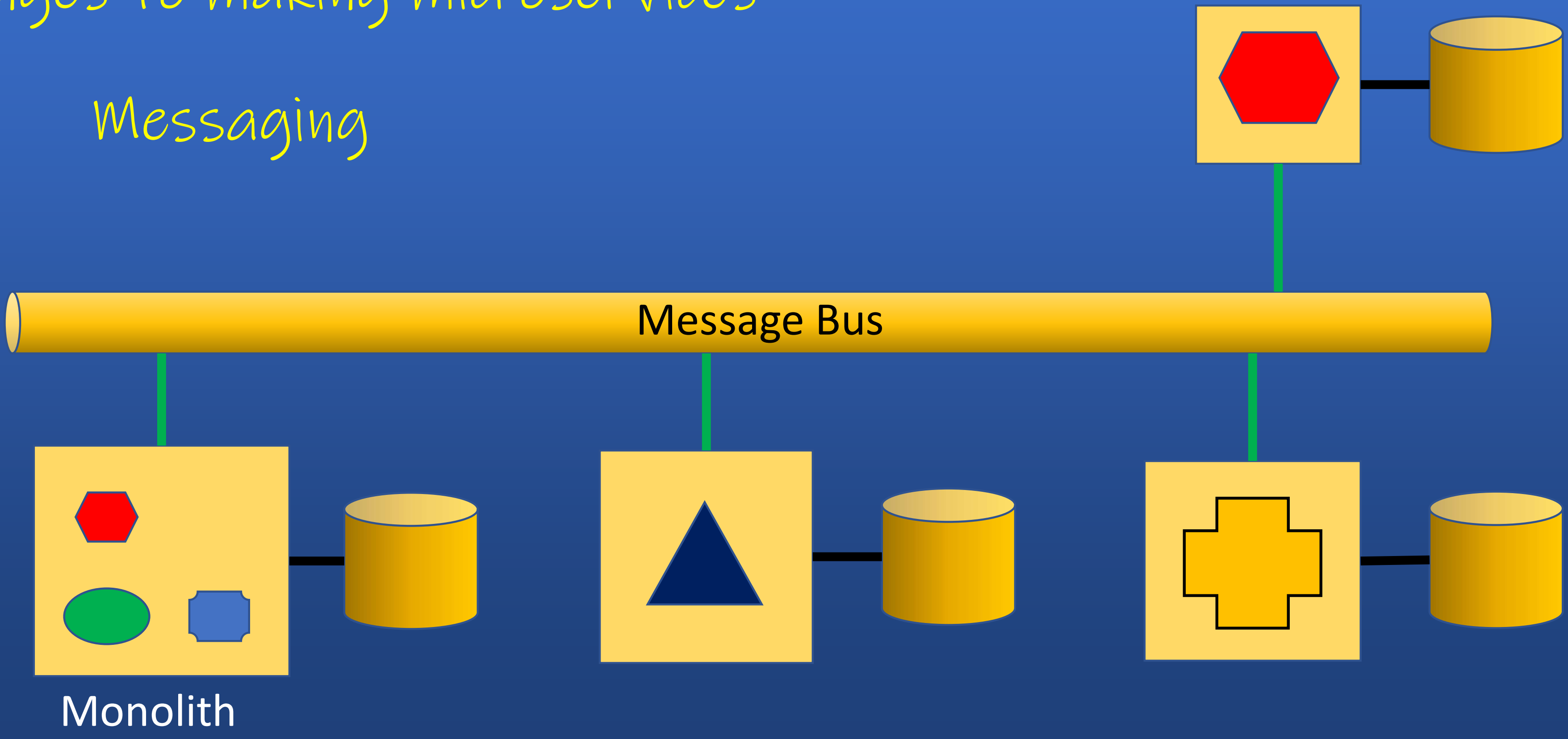
Orders

~~~~~~~
~~~~
~~~~~~~~~~~
~~~~~~~~
~~~~~~~~~~~~~~
~~~~~~~~~~
~~~~~~~
~~~~~~~~~~

Microservices – The Easy Way is the Wrong Way

Challenges to making microservices

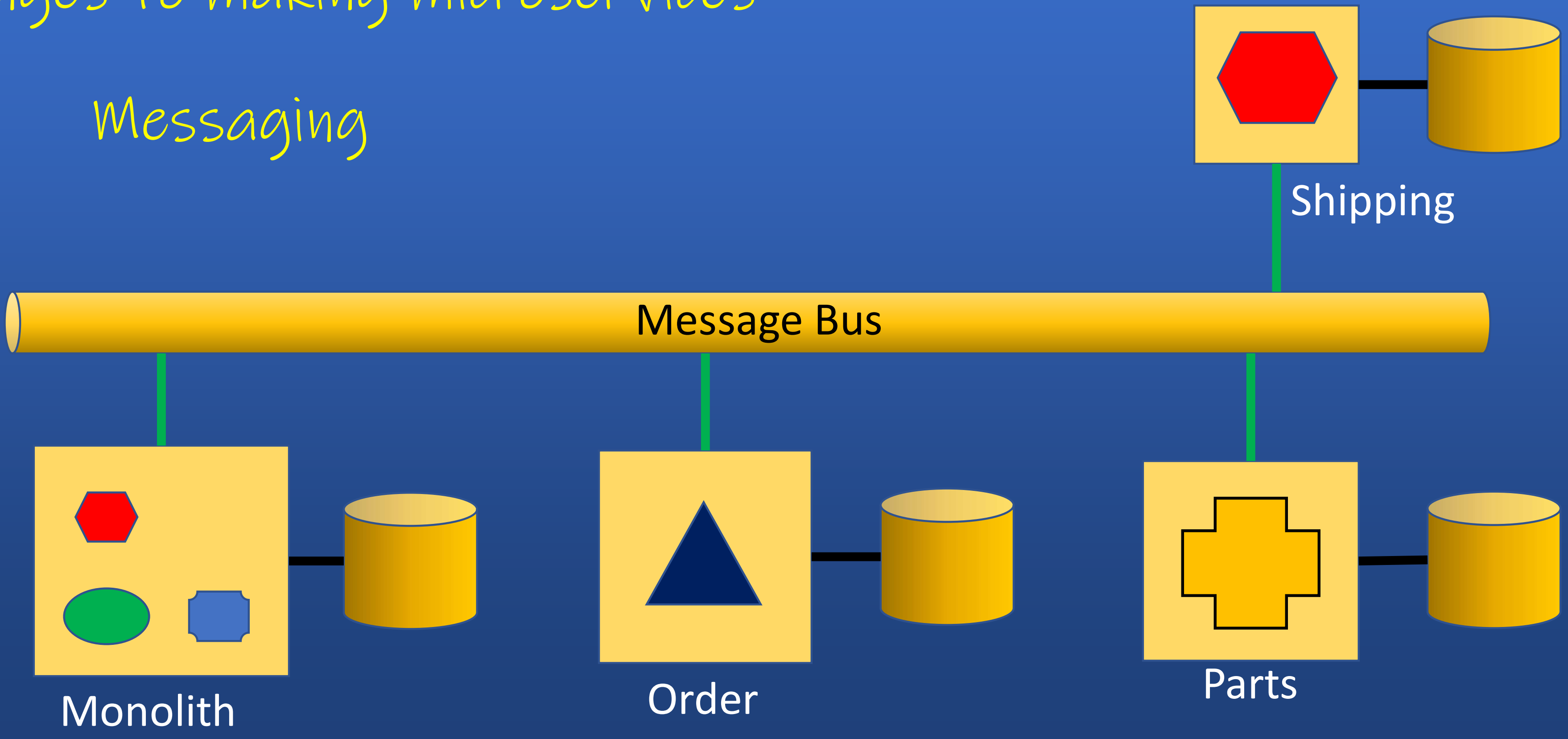
Messaging



Microservices – The Easy Way is the Wrong Way

Challenges to making microservices

Messaging

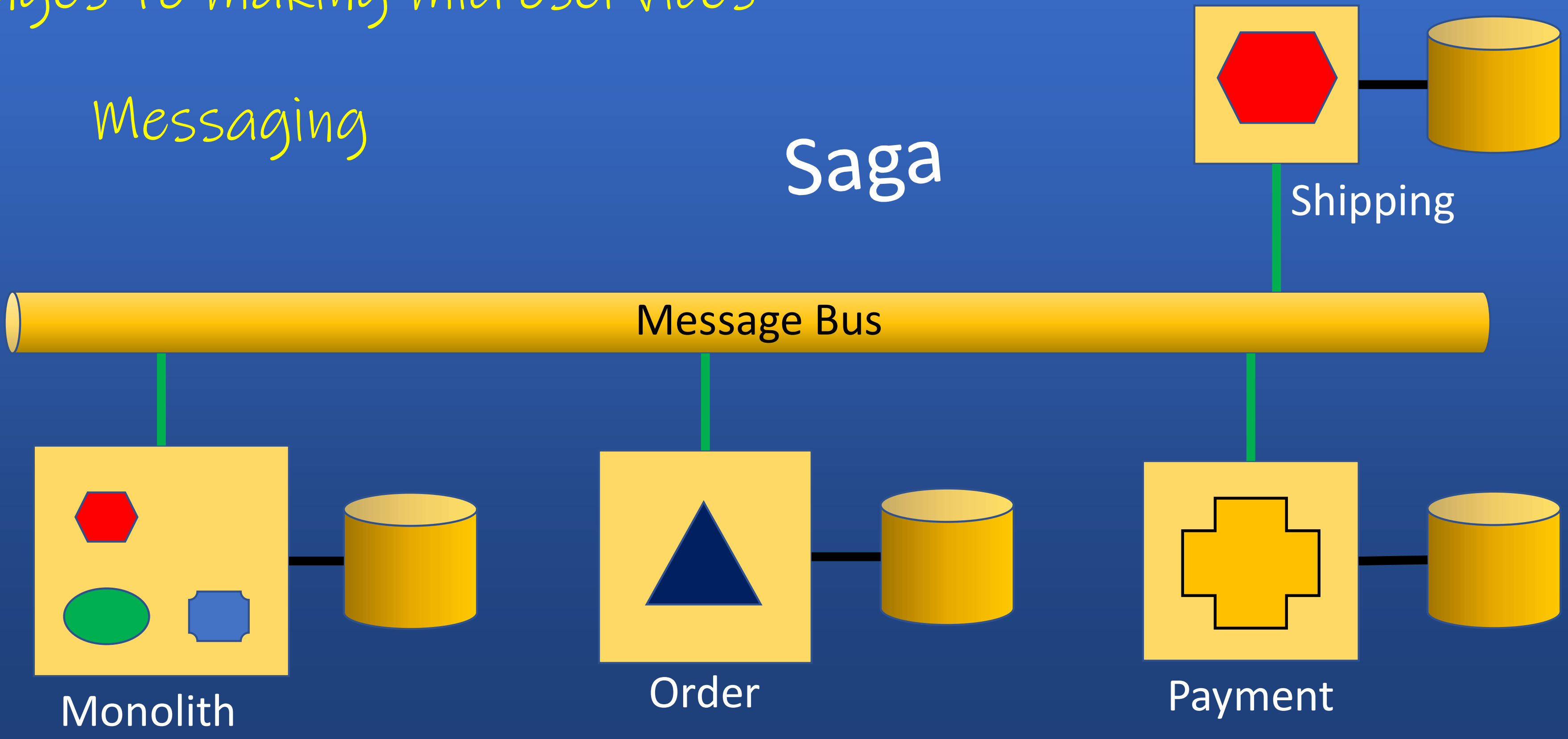


Microservices – The Easy Way is the Wrong Way

Challenges to making microservices

Messaging

Saga



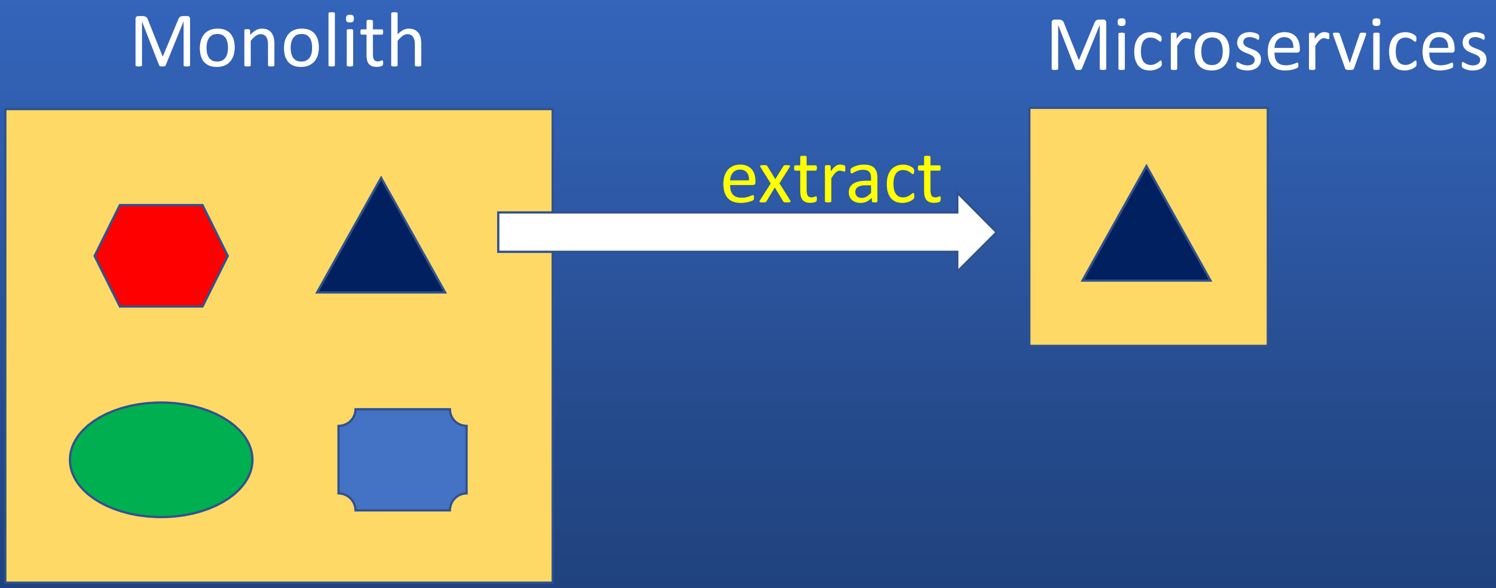
Microservices – The Easy Way is the Wrong Way

Decomposing Monolith to Microservices

(High-Level)

Microservices – The Easy Way is the Wrong Way

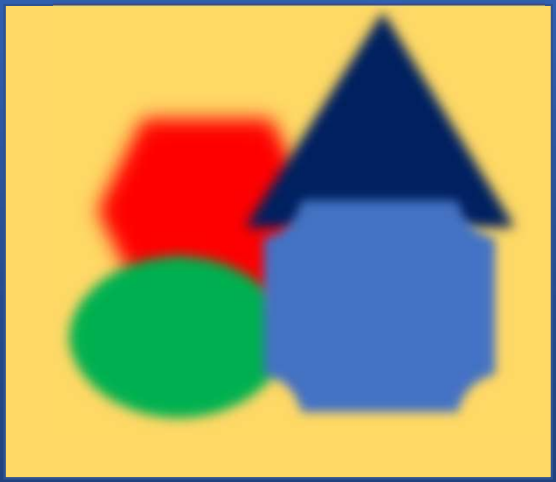
Decomposing Monolith to Microservices



Microservices – The Easy Way is the Wrong Way

Decomposing Monolith to Microservices

Monolith

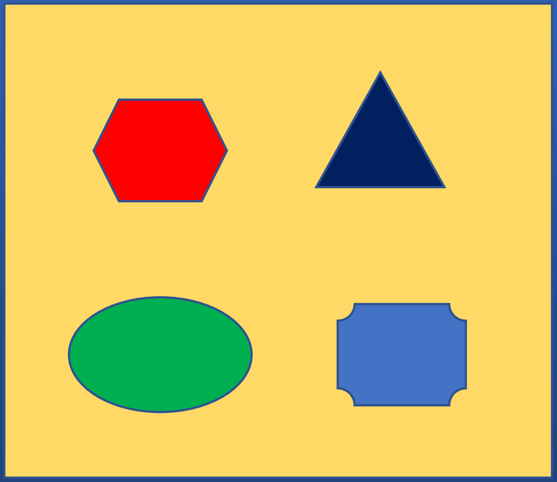


- Boundaries not clearly defined
- Lacking good Dependency Injection
- Requires refactoring

Microservices – The Easy Way is the Wrong Way

Decomposing Monolith to Microservices

Monolith



Start With Small Piece

Payoff in

- Learning
- Building Infrastructure

Microservices – The Easy Way is the Wrong Way

Decomposing Monolith to Microservices

Using DDD

Not the only way!

Company: Wham-Bam
Products: Fidget Widgets

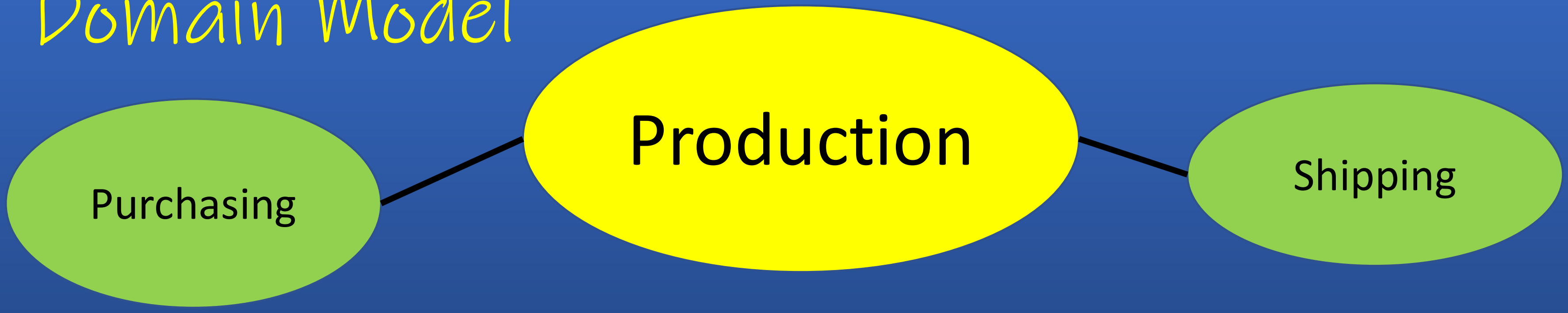
Production

Microservices – The Easy Way is the Wrong Way

Decomposing Monolith to Microservices

Event Storming

Domain Model



Production

Purchasing

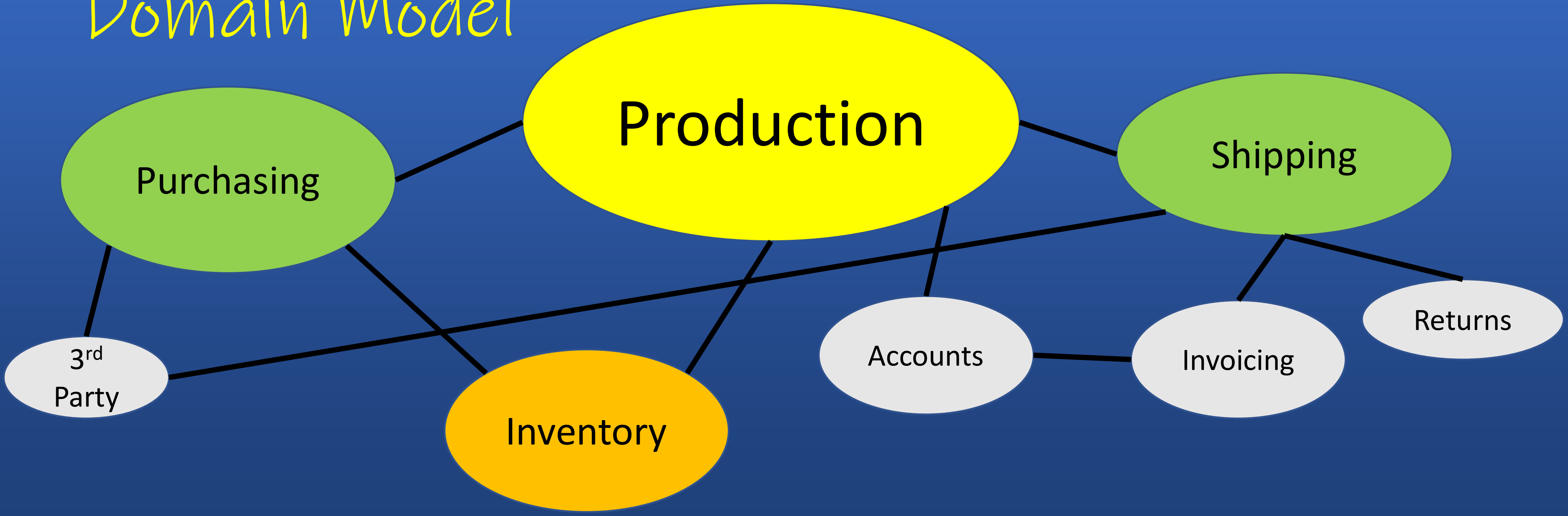
Shipping

Microservices – The Easy Way is the Wrong Way

Decomposing Monolith to Microservices

Event Storming

Domain Model

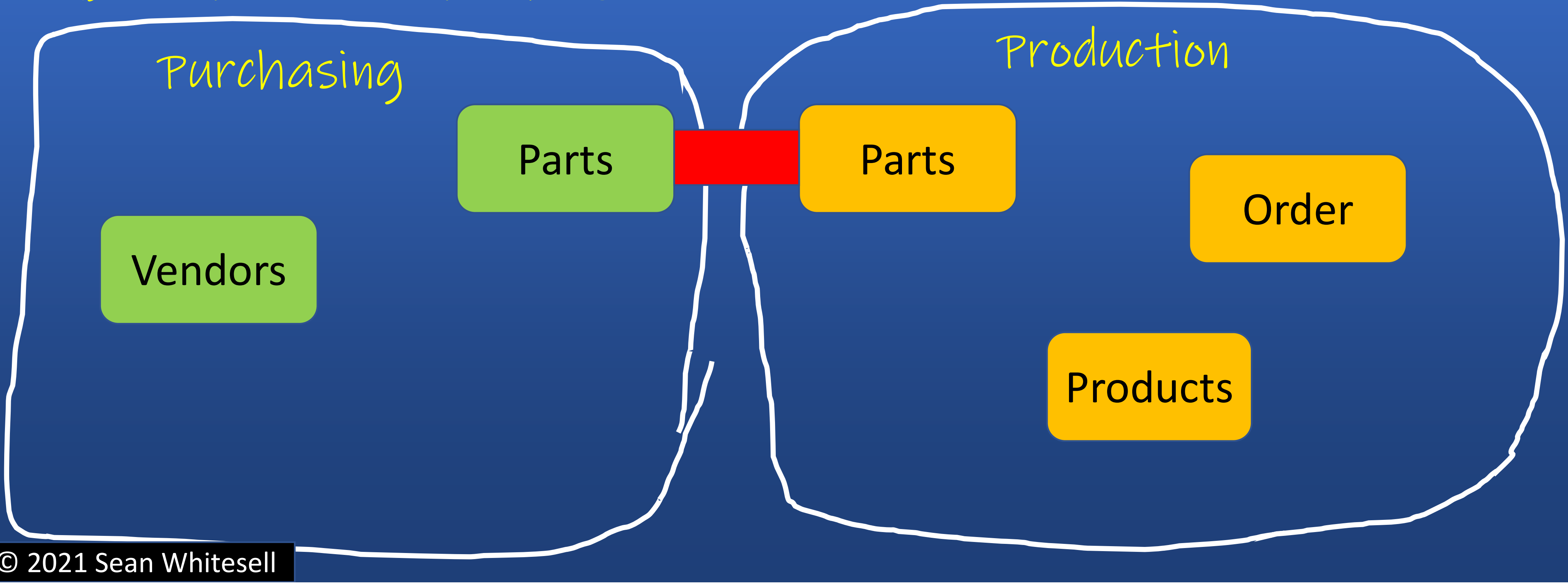


Microservices – The Easy Way is the Wrong Way

Decomposing Monolith to Microservices

Bounded Contexts

Using DDD

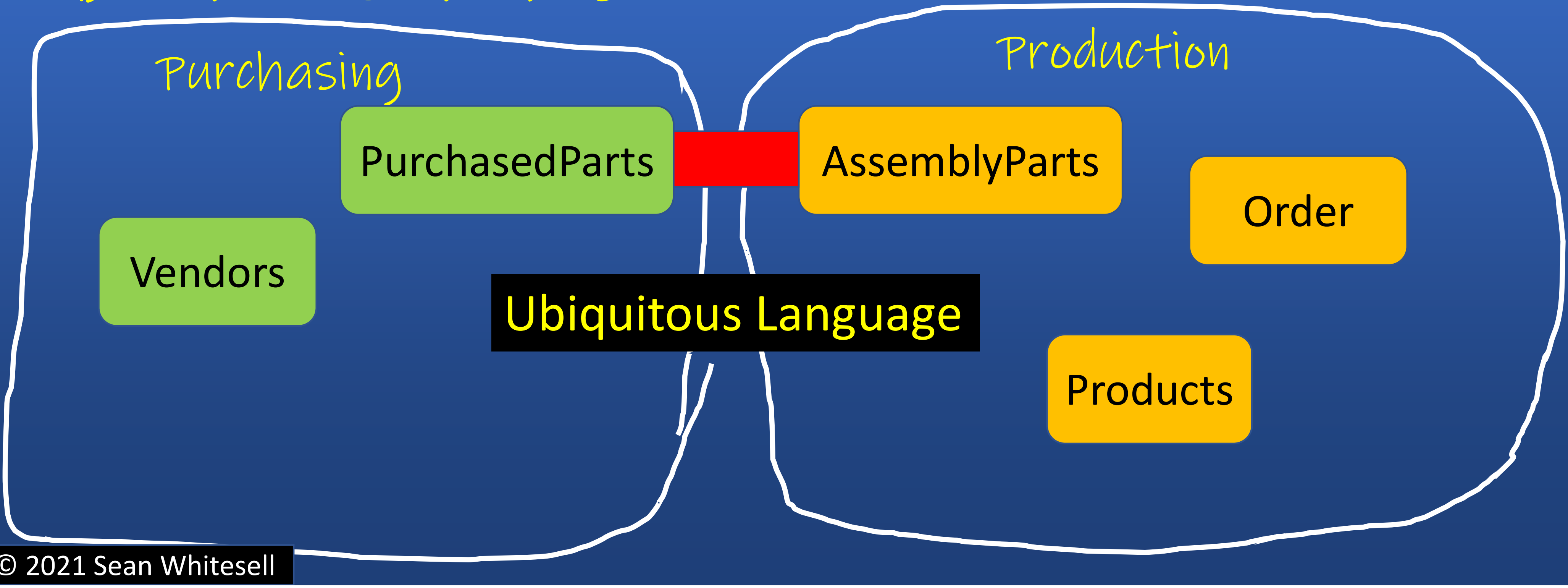


Microservices – The Easy Way is the Wrong Way

Decomposing Monolith to Microservices

Using DDD

Bounded Contexts



Purchasing

PurchasedParts

Vendors

Production

AssemblyParts

Order

Products

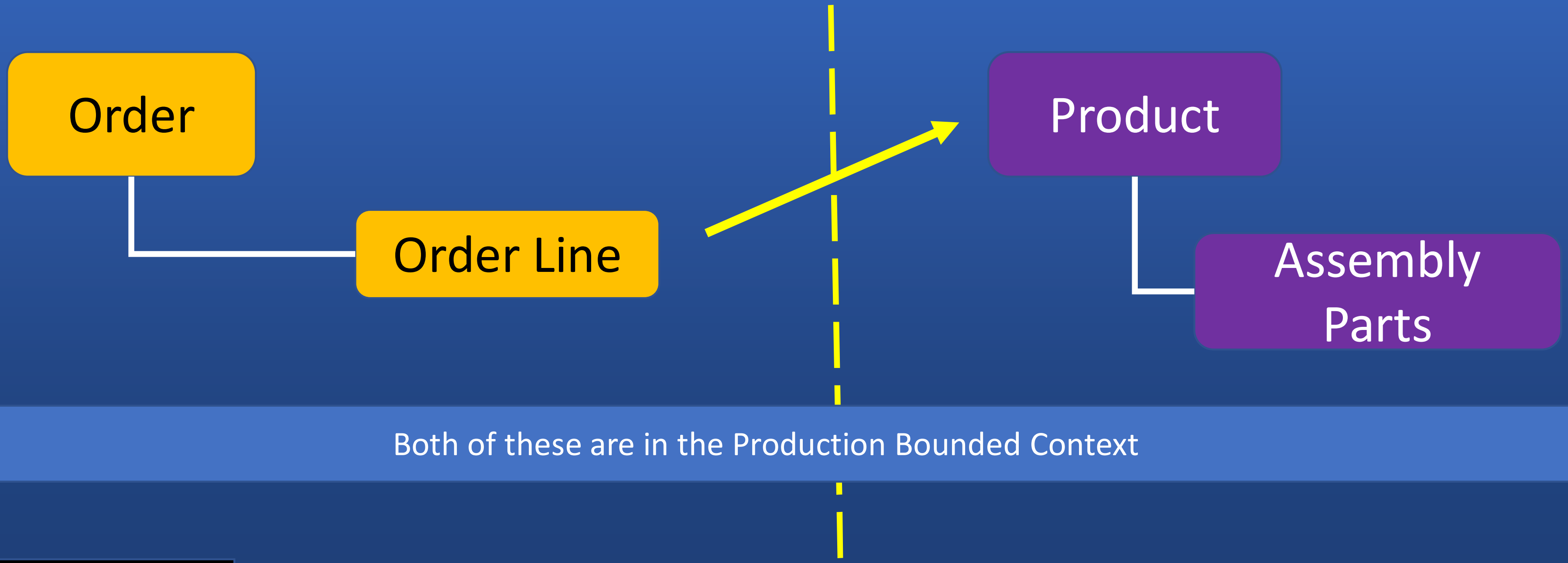
Ubiquitous Language

Microservices – The Easy Way is the Wrong Way

Decomposing Monolith to Microservices

Using DDD

Aggregates

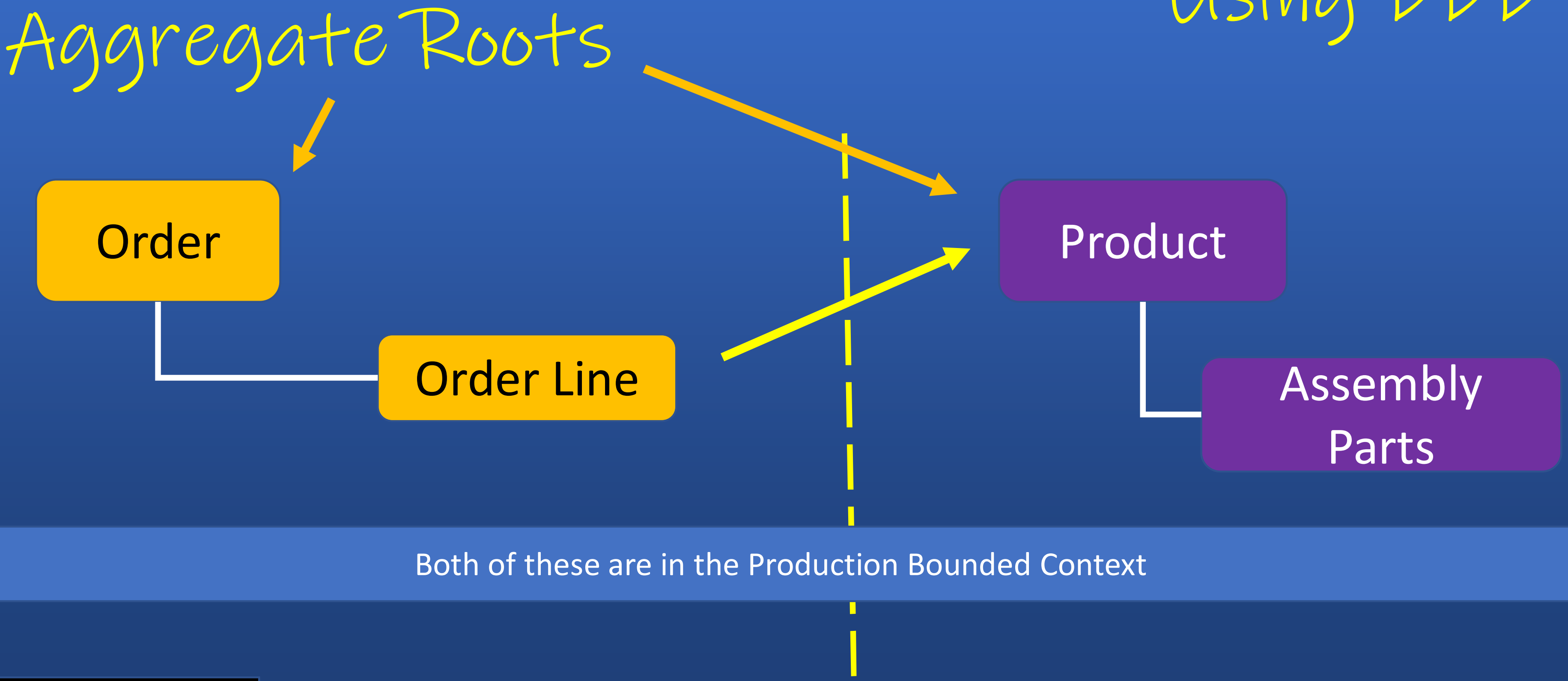


Both of these are in the Production Bounded Context

Microservices – The Easy Way is the Wrong Way

Decomposing Monolith to Microservices

Using DDD



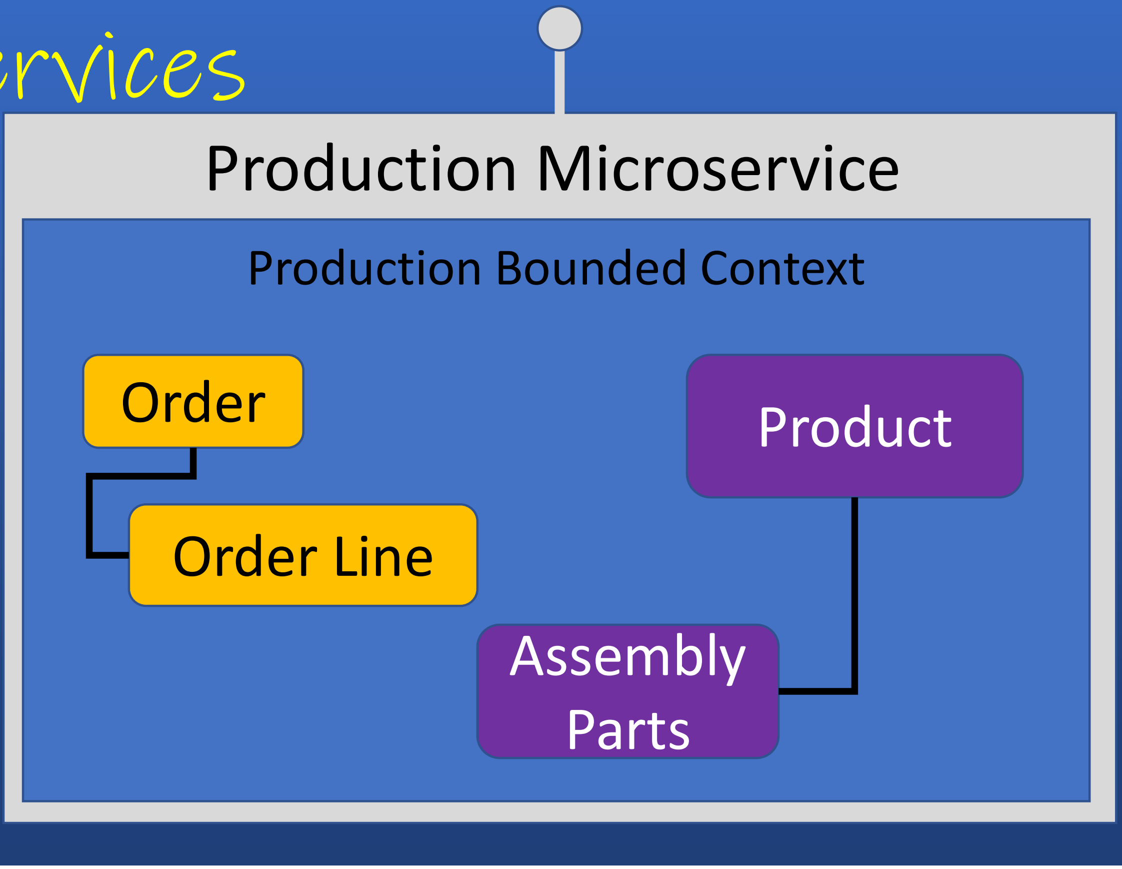
Both of these are in the Production Bounded Context

Microservices – The Easy Way is the Wrong Way

Decomposing Monolith to Microservices

Identify Possible Services

- Assemble Parts
- Assemble Products
- Insert Order
- Revise Order



Production Microservice

Production Bounded Context

Order

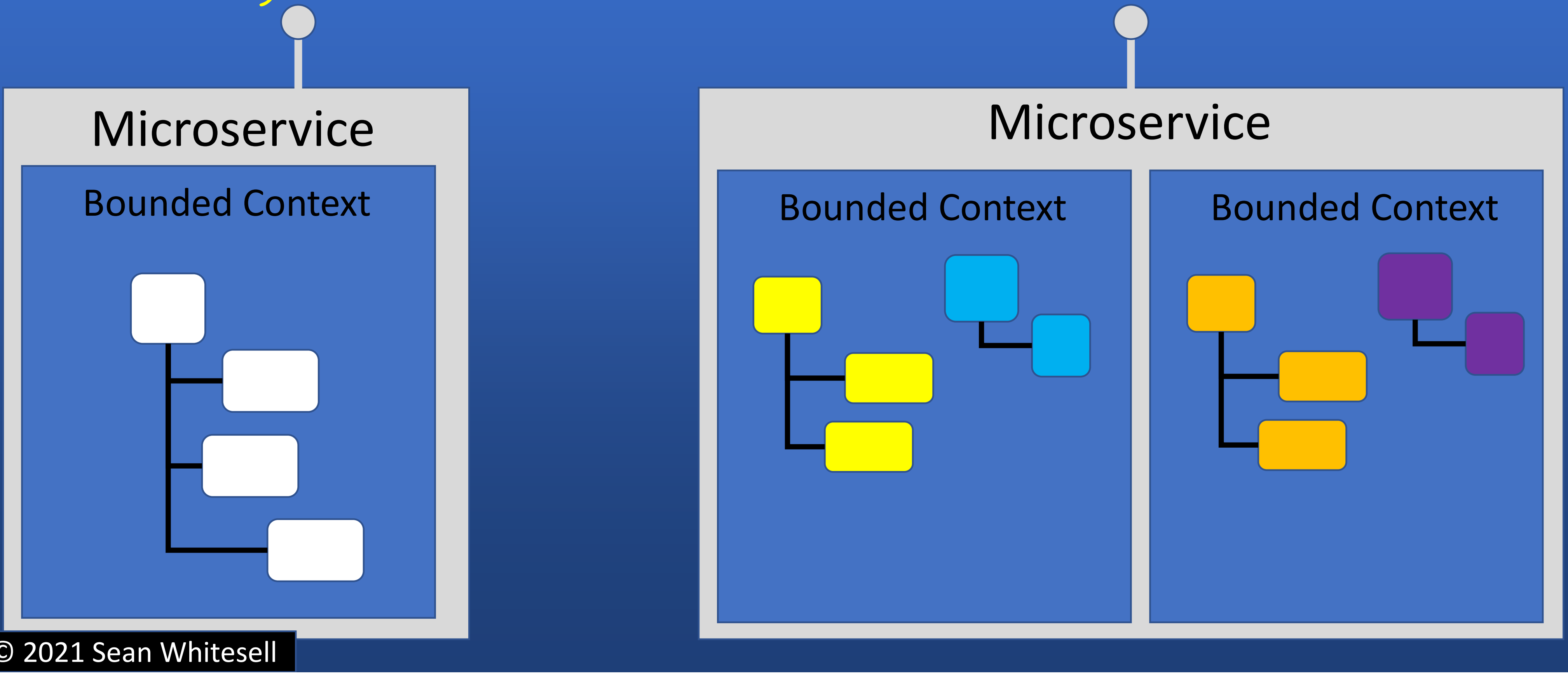
Order Line

Product

Assembly Parts

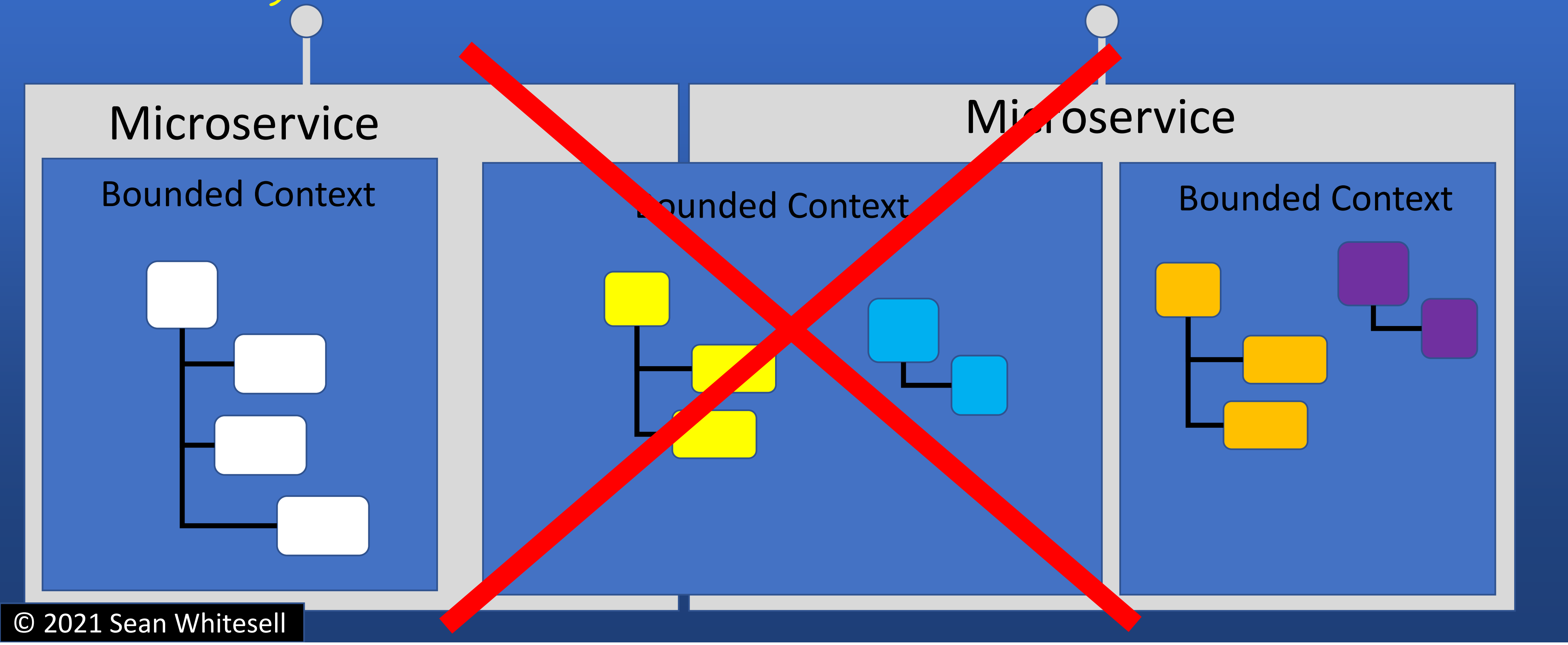
Microservices – The Easy Way is the Wrong Way

Decomposing Monolith to Microservices



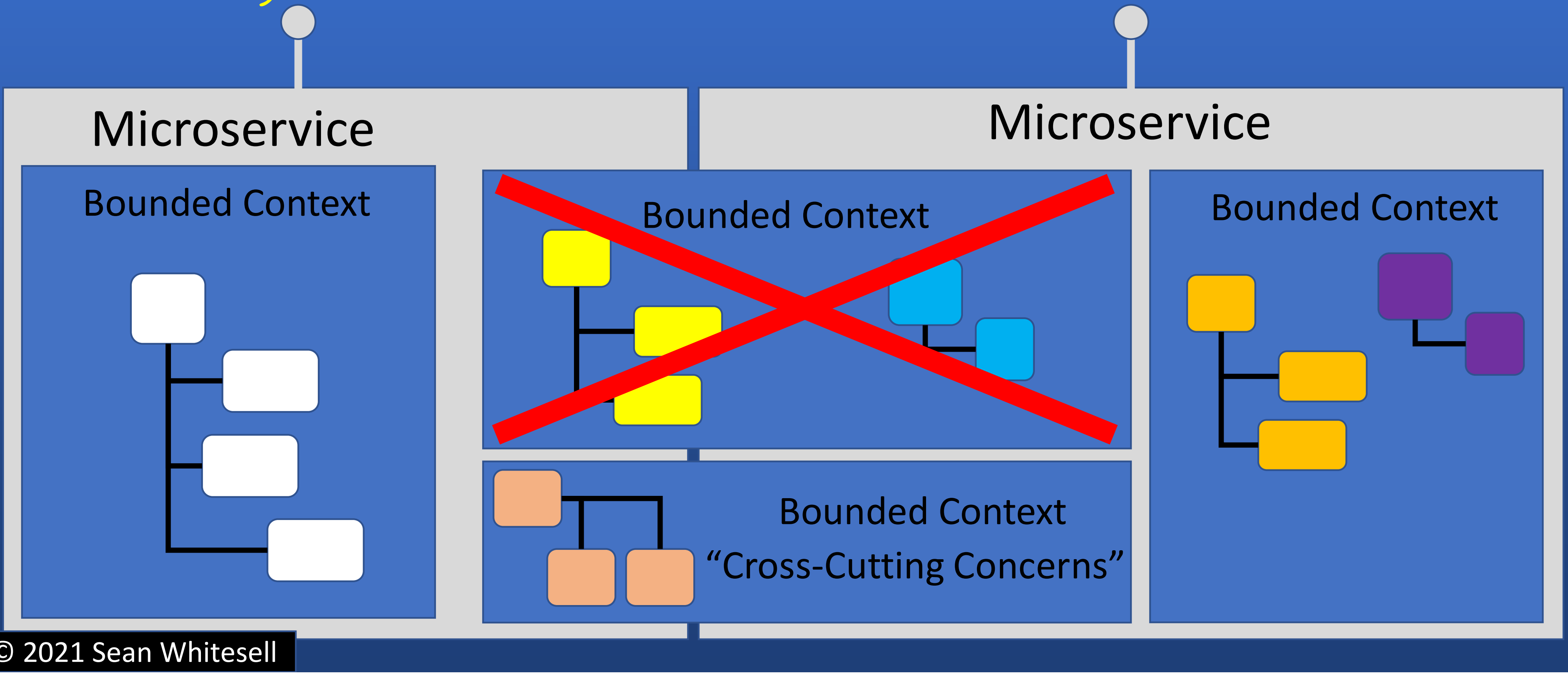
Microservices – The Easy Way is the Wrong Way

Decomposing Monolith to Microservices



Microservices – The Easy Way is the Wrong Way

Decomposing Monolith to Microservices



Microservices – The Easy Way is the Wrong Way

A Quick Review

Microservices are hard

They require good analysis and teamwork

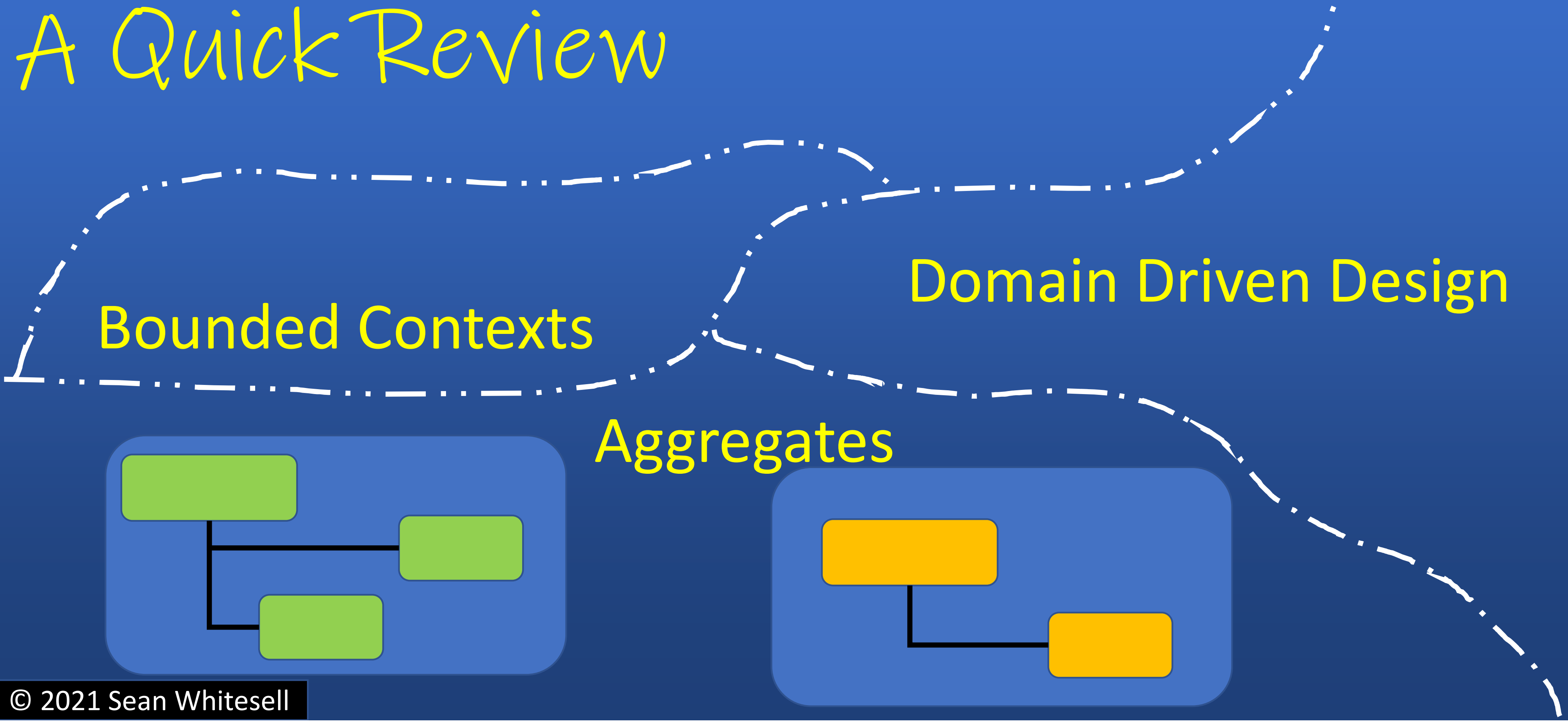
This Photo by Unknown Author is licensed under CC BY-SA-NC

Team Autonomy
Service Autonomy
Scalability
Fault Isolation

Cost time, money, learning, investing
Messaging; RPC; sync vs async; pub/sub
CI / CD & Repo per microservice
Sharing code keeps services tightly coupled
Need to know about Layer 4 and 7
TLS for service communication layer; OAuth for application layer

Microservices – The Easy Way is the Wrong Way

A Quick Review



Domain Driven Design

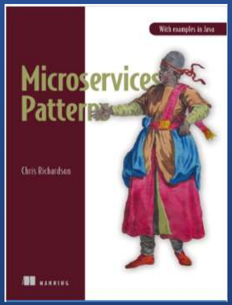
Bounded Contexts

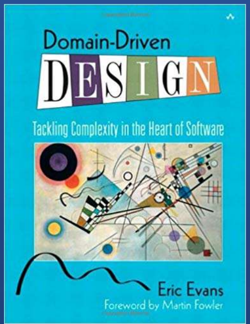
Aggregates

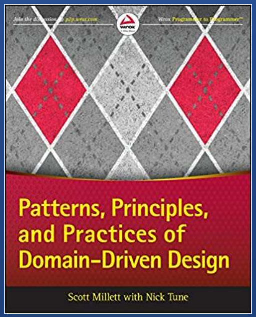
Microservices – The Easy Way is the Wrong Way

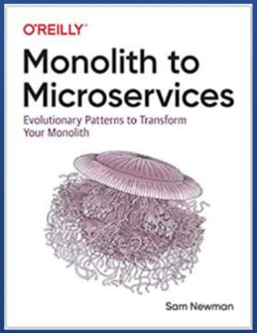
Recommendations

Microservices Patterns – Chris Richardson
Domain-Driven Design: Tackling Complexity in the Heart of Software – Eric Evans
Patterns, Principles, and Practices of Domain-Driven Design – Scott Millett & Nick Tune
Monolith to Microservices – Sam Newman




Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler


Patterns, Principles, and Practices of Domain-Driven Design
Scott Millett with Nick Tune


O'REILLY
Monolith to Microservices
Evolutionary Patterns to Transform Your Monolith
Sam Newman

Microservices – The Easy Way is the Wrong Way

Thank You !

What Questions Do You Have?

Sean Whitesell

@codewithseanw

seanwhitesell.com

President of Tulsa .NET User Group &&
Microsoft MVP, ASP Insider &&
Cloud Architect @ TokenEx, LLC

meetup.com/TulsaDevelopers-net

twitch.tv/codewithsean