



# Introduction to Microservices

Mohammad Hesam Nasiri



# Type of Architectures

- Monolithic
- SOA ( Service Oriented Architecture)
- MicroServices



architecture



microservices



meaning



concrete



application



utah



design



Wikipedia  
Utah monolith - Wik...



The Conversation  
Mystery monoliths: why ...



Condé Nast Traveller In...  
The monolith has been t...



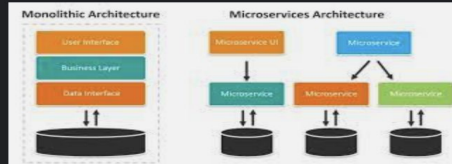
Architizer  
Monolithic Building Techniques ...



Vox  
The monoliths in Utah, ...



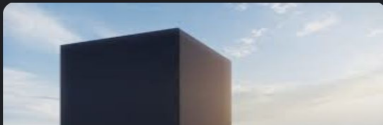
Krusche & Company  
Monolithic vs microservices vs ...



SUSE  
Microservices vs. Monolithic ...



www.monolithic.com  
Monolithic Dome construction and ...



# Monolithic



# Pros of Monolithic

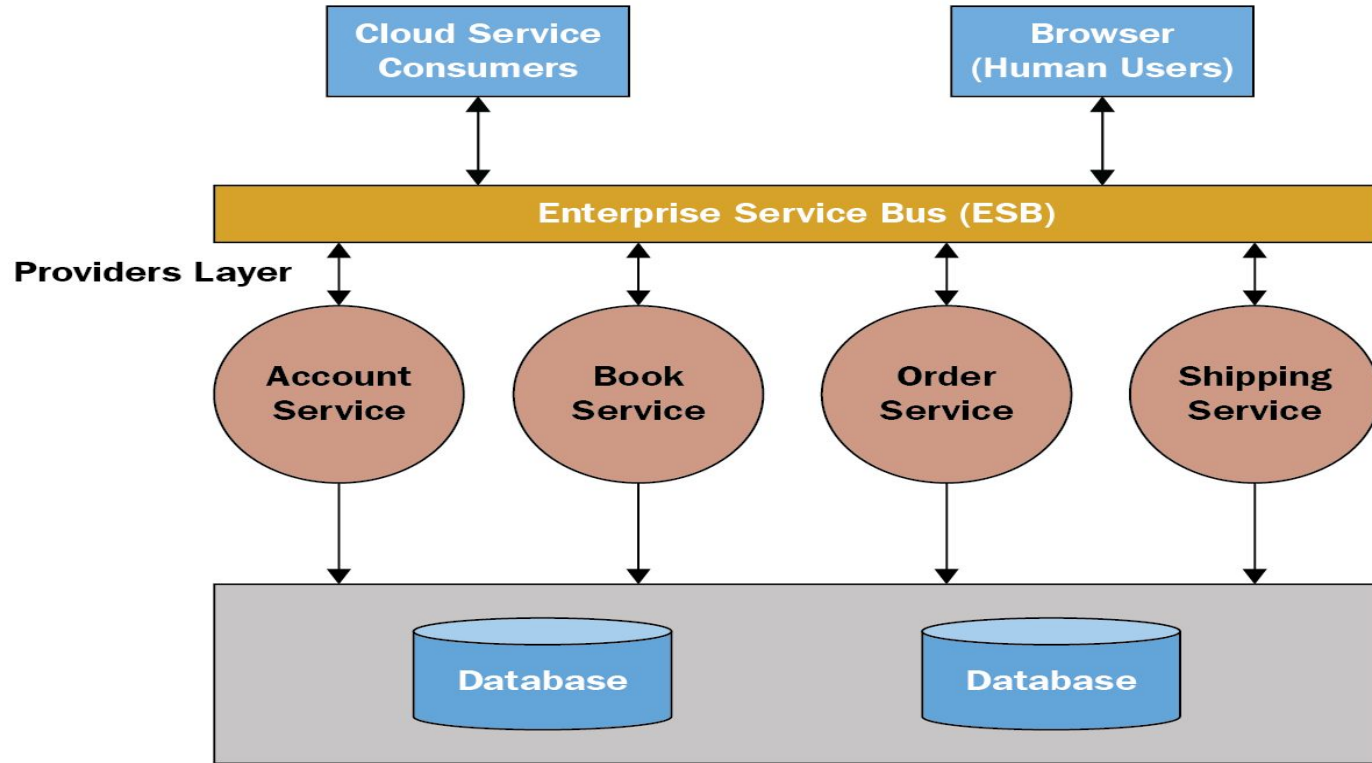
- Easy Test
- Easy Develop in Small Projects
- Easy to Scale !
- Easy to Deployment
- Less complexity



# Cons of Monolithic

- Lack of Agility
- Lack of Innovation
- New release take month
- Operation is Nightmare!
- Difficult to Scale ?

## Consumers Layer





## Pros of SoA

- Reusability
- Contract Based
- Stateless
- Loose coupling
- ..

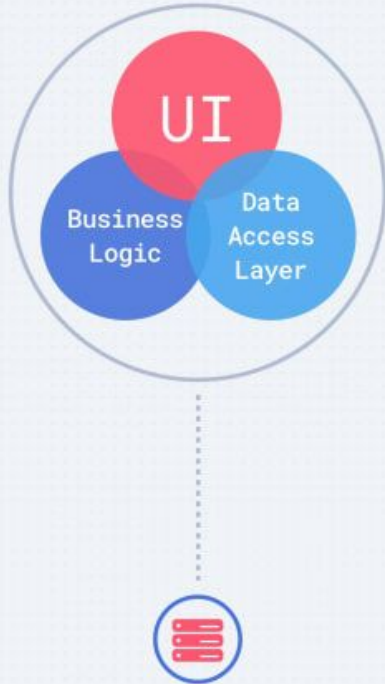


# MicroServices

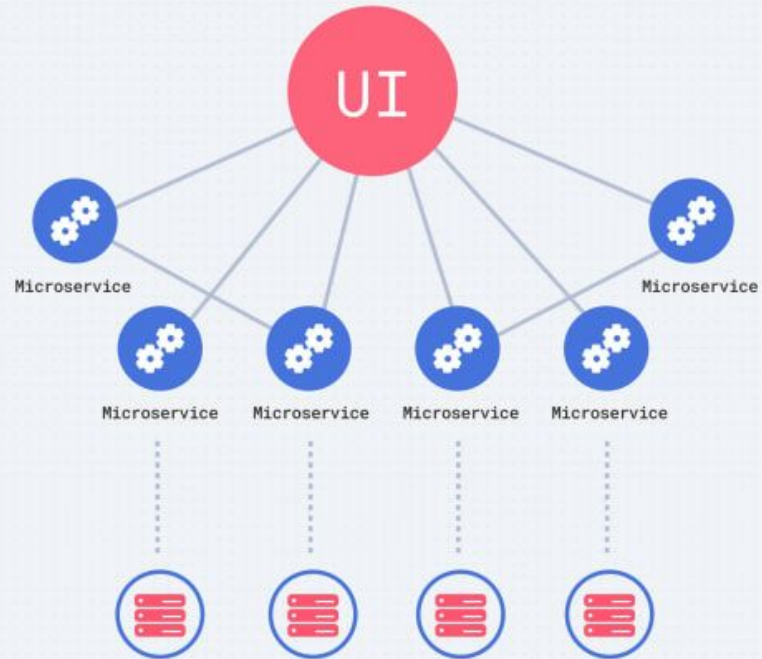
Microservice is a way to divide an application (software) into **small, lightweight, independent** and **manageable** parts or services. In other words, microservice is a so-called distributed software development architecture.

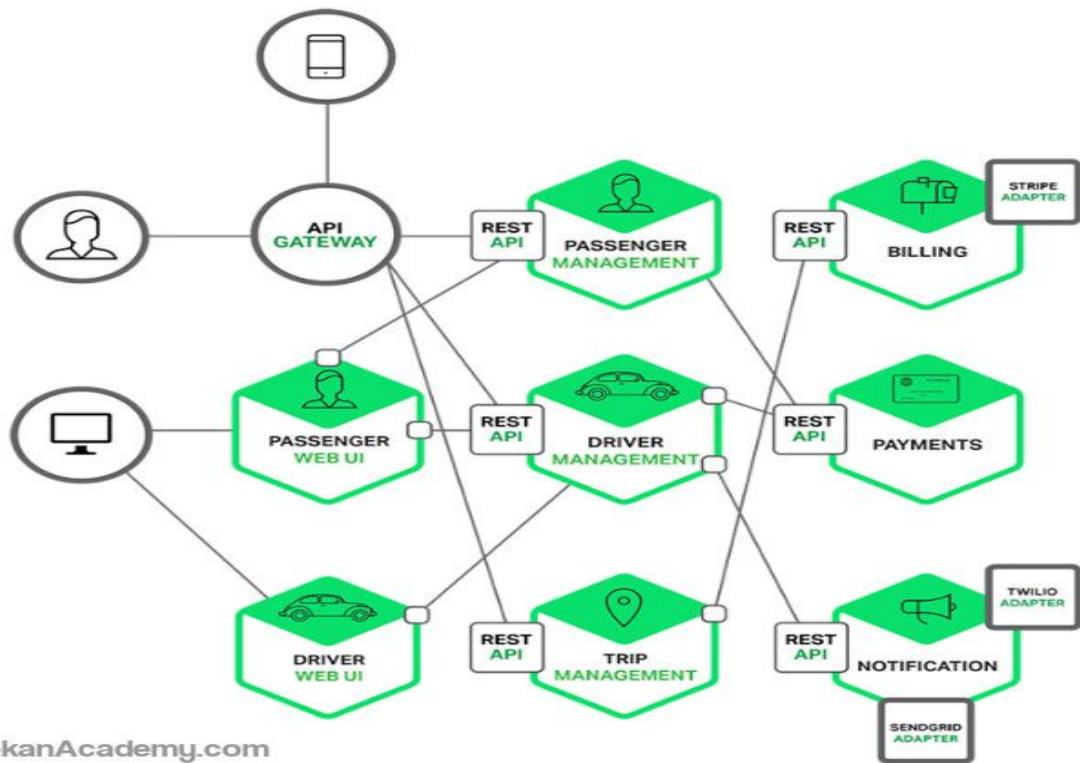


## Monolithic Architecture



## Microservices Architecture





SokanAcademy.com

## Microservice



## Pros of MicroServices

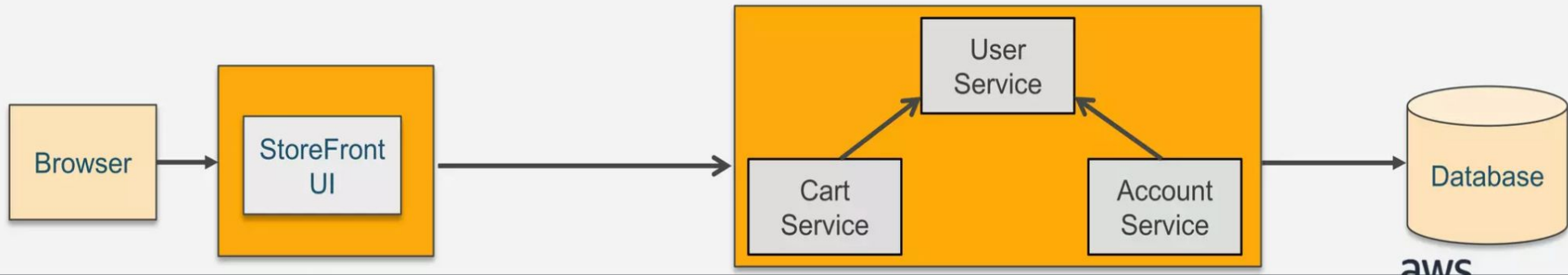
- Ease of Understanding of the Codebase
- Improved Fault Tolerance
- isolated services are easier to debug and maintain
- faster time to market
- freedom of choice of software and hardware
- can be easily used together with modern technologies (e.g. containerization and cloud computing)
- better scalability
- ....



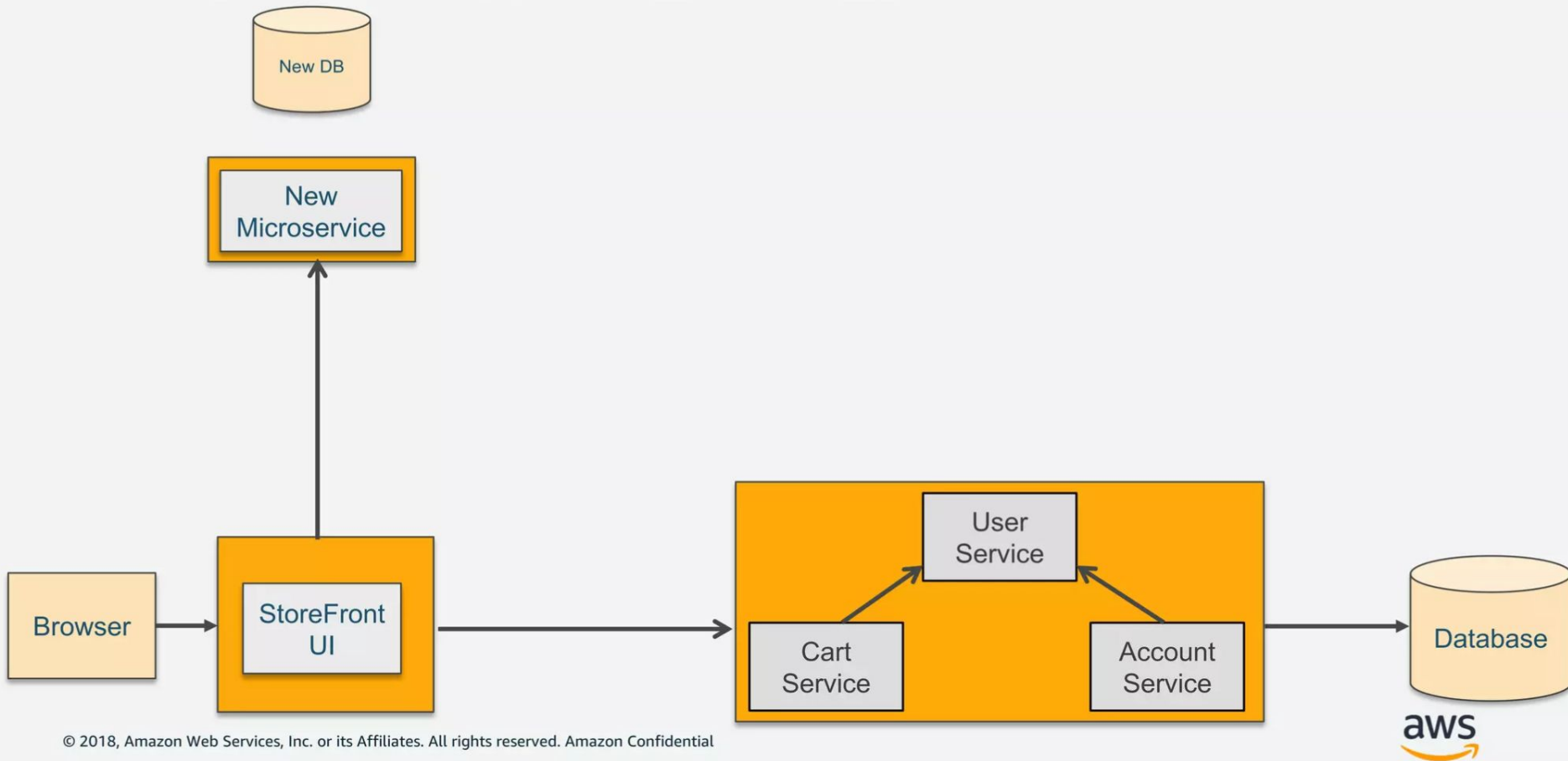
## Cons of MicroServices

- Hard Deployment
- extra complexity that smaller teams are not always prepared for
- messaging between the services has a performance overhead (e.g. network latency, message processing, etc.)
- additional security issues
- Requires More Resources
- Global Testing and Debugging is Difficult

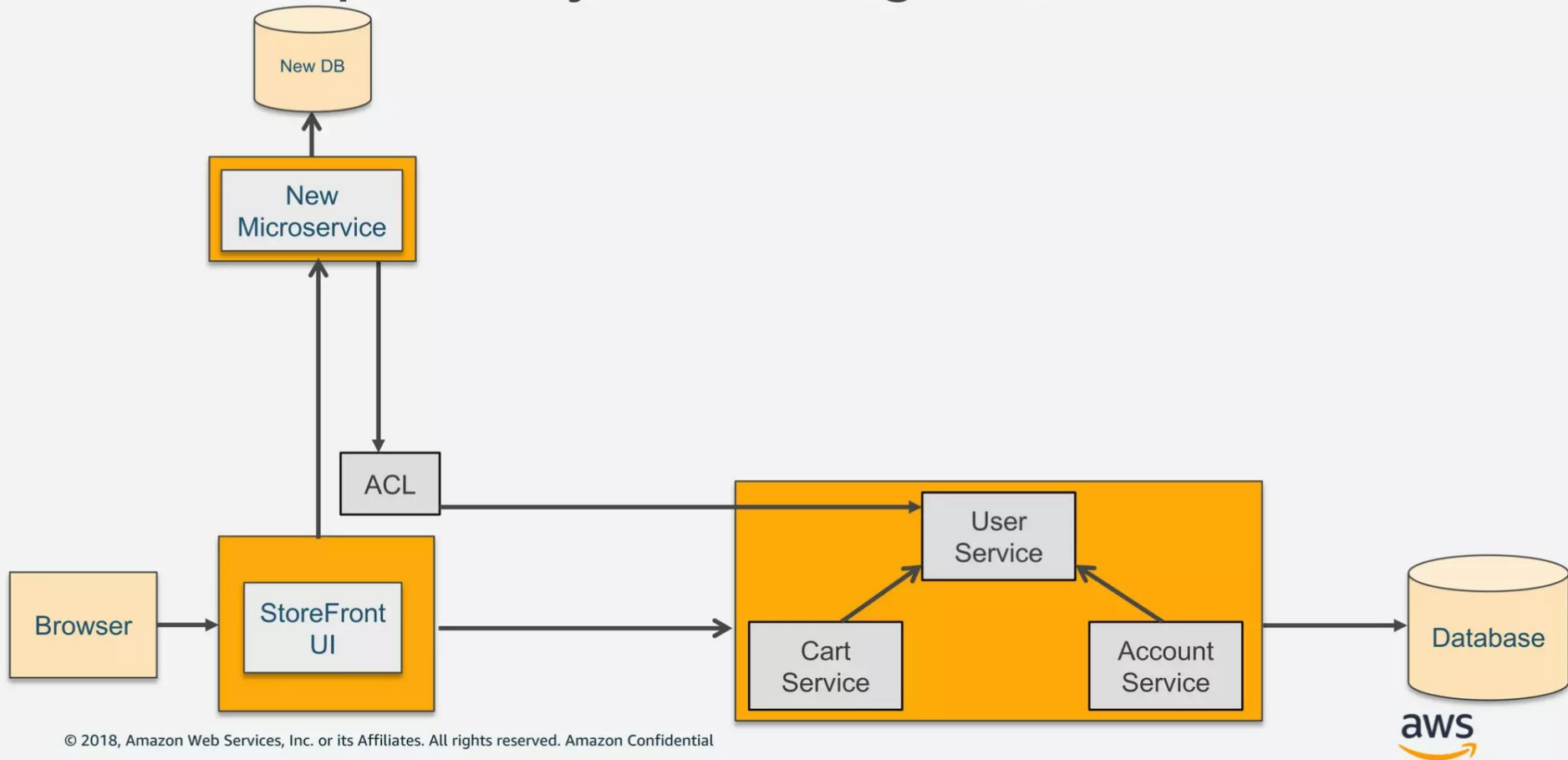
# Strangling the Monolith



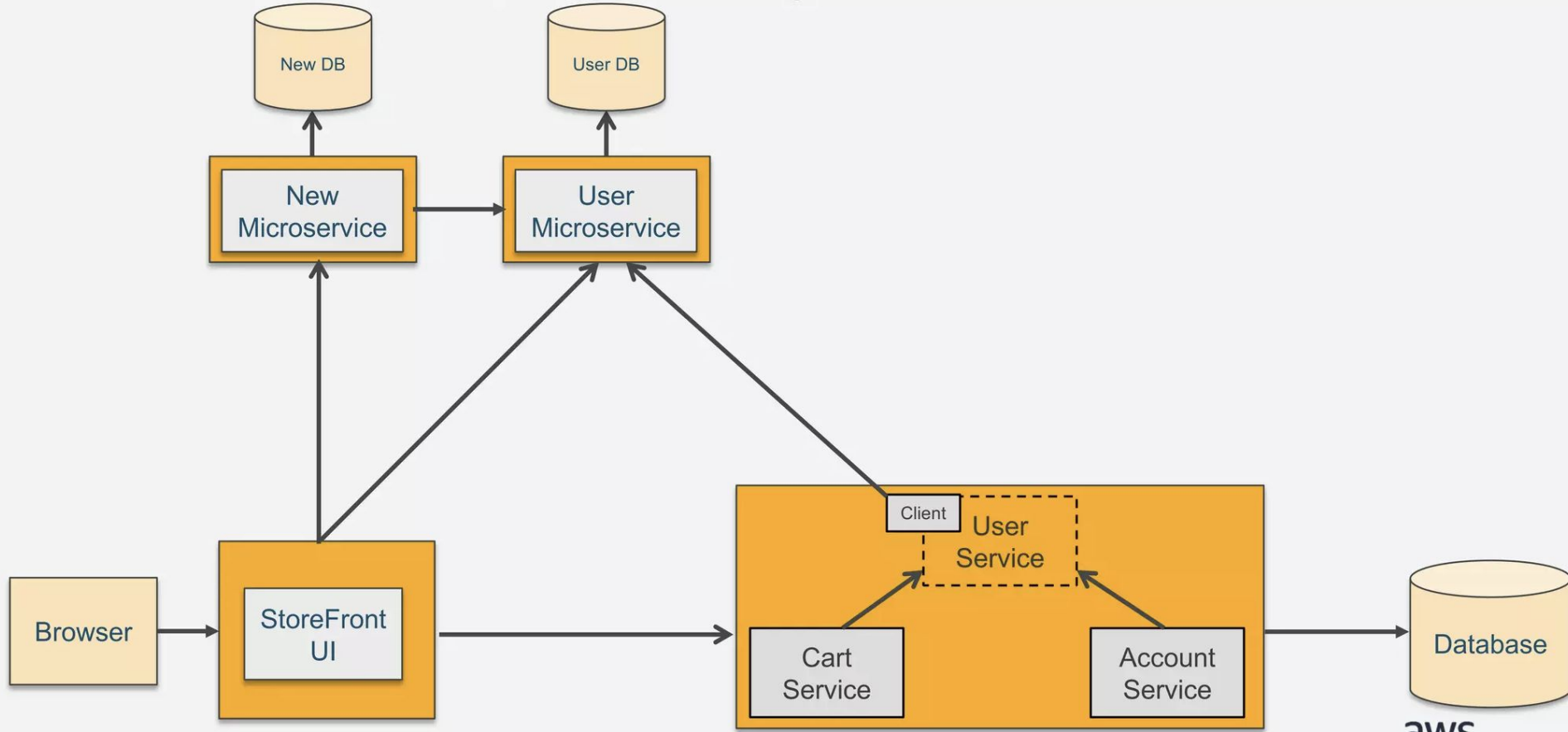
# No New Feature on the Monolith



# Anti-Corruption Layer for Integration

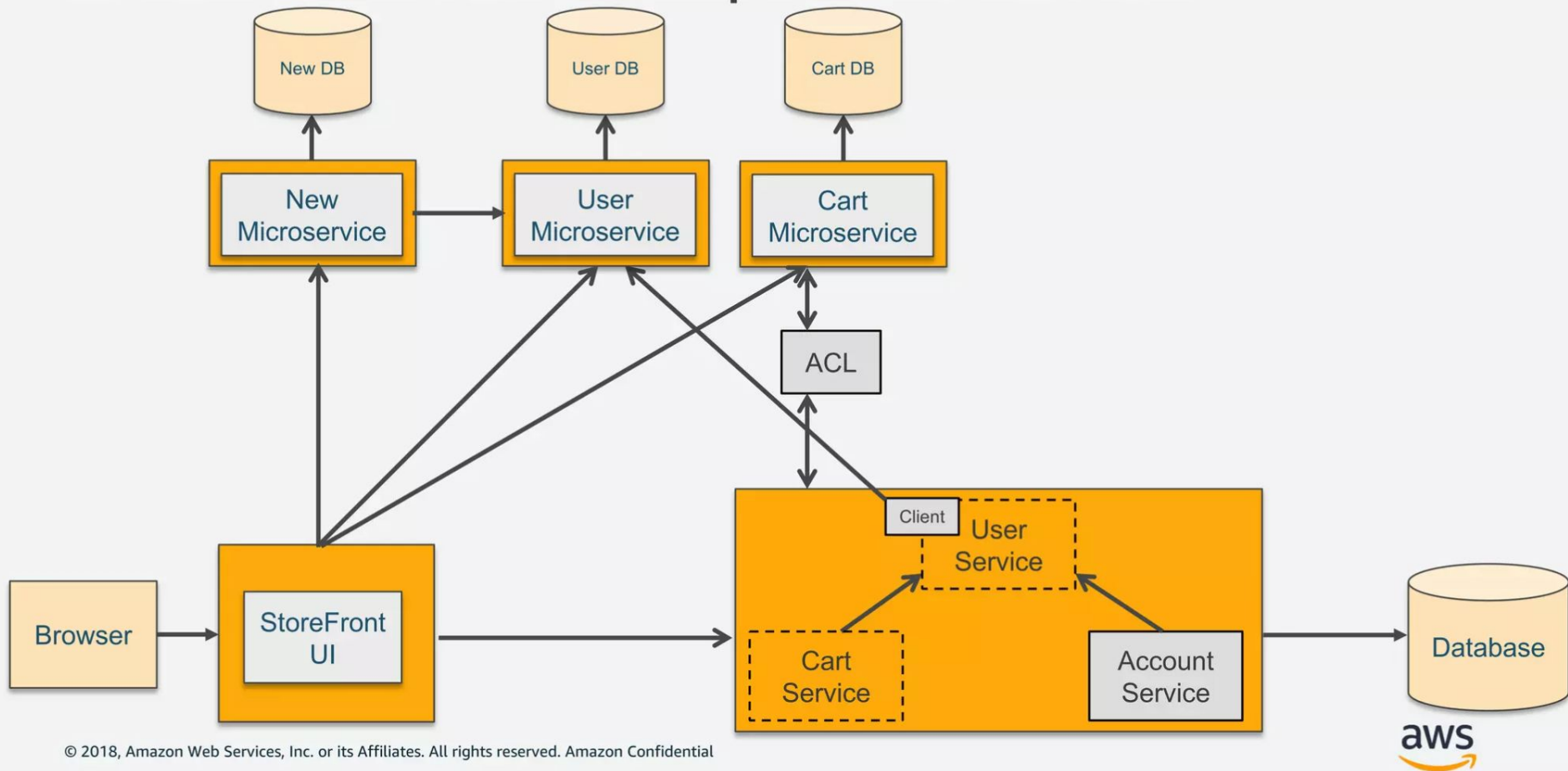


# Start with a Simple Existing Service...

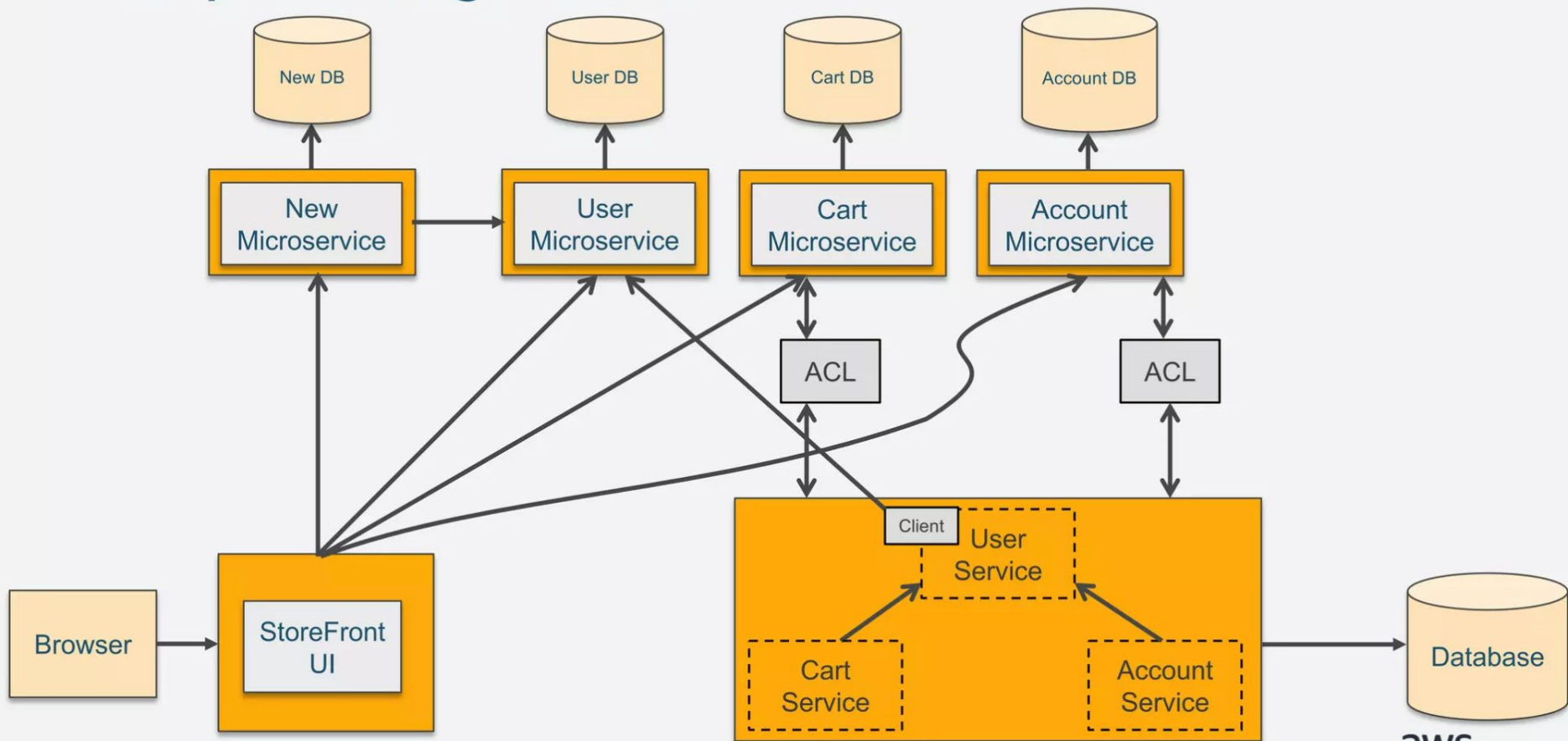




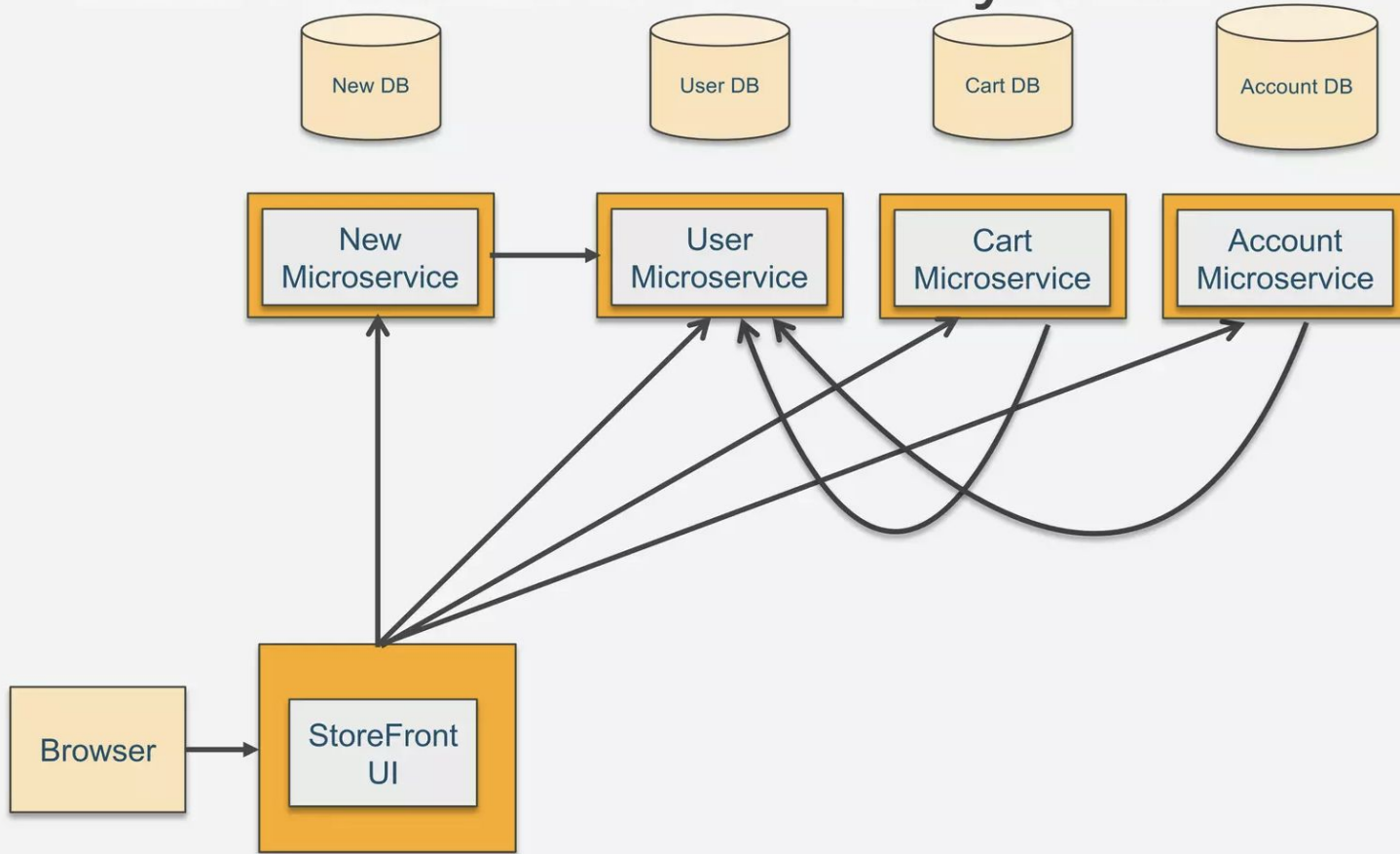
# ...Move on to More Complicated Services

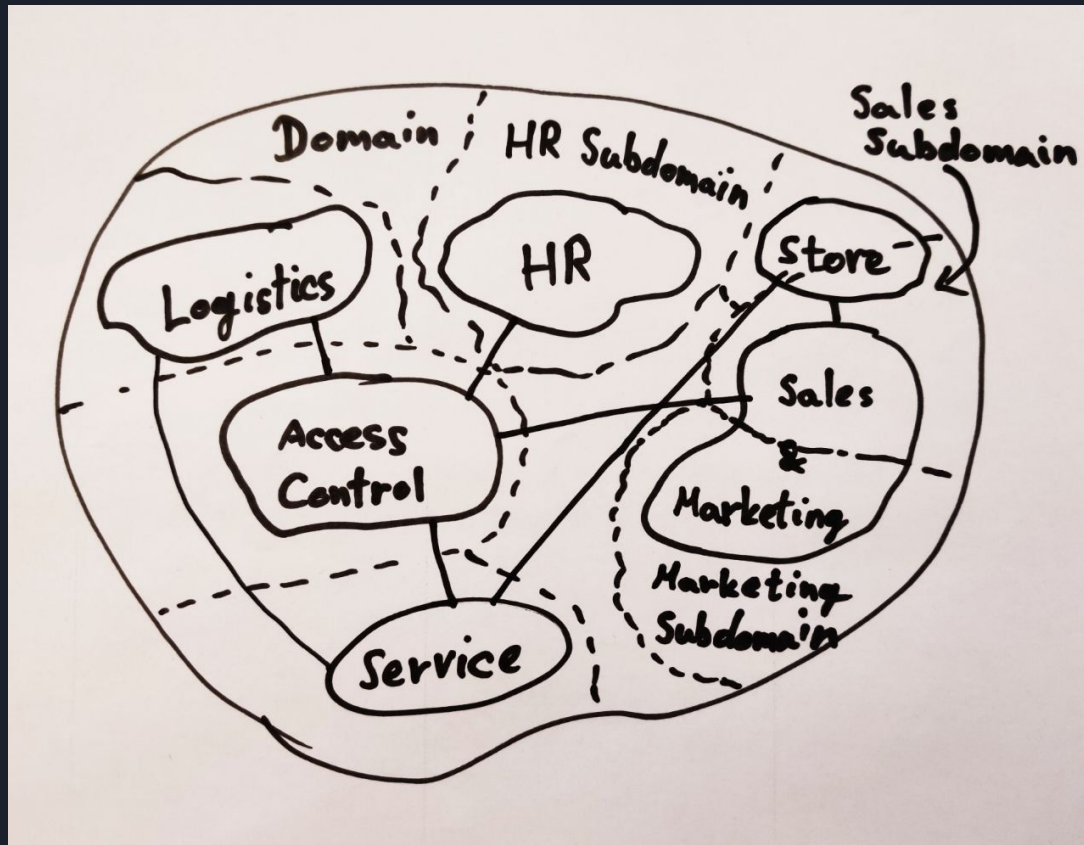


# ...Keep Walking and...



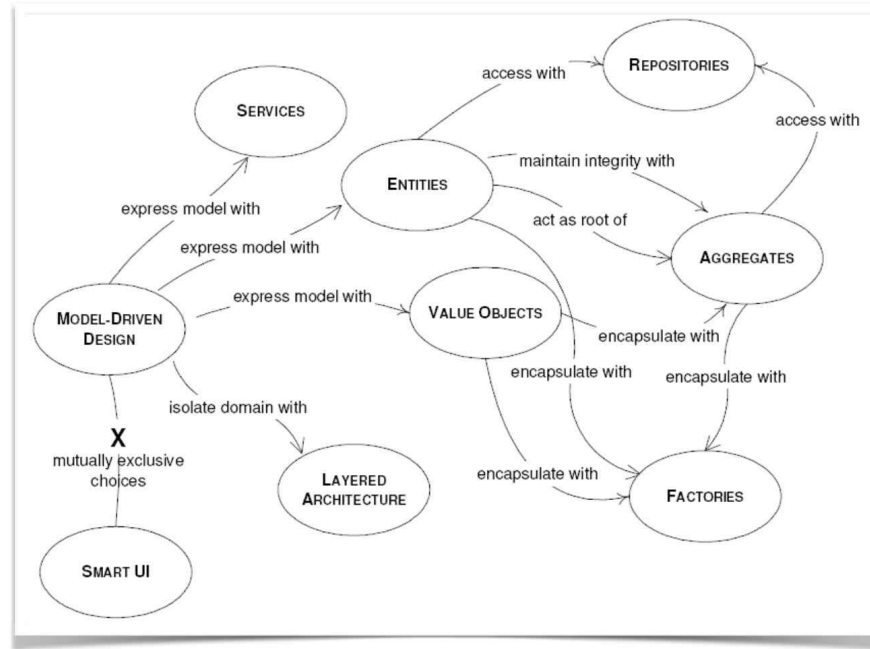
# ...The Monolith is Eventually Gone





Bounded Context

# Domain Driven Design

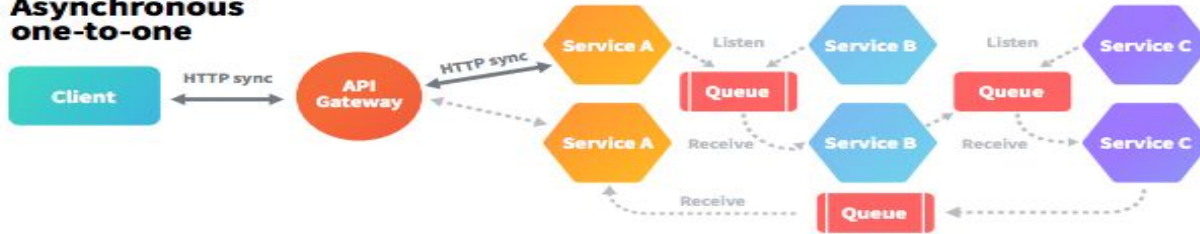


DDD in Microservices

### Synchronous



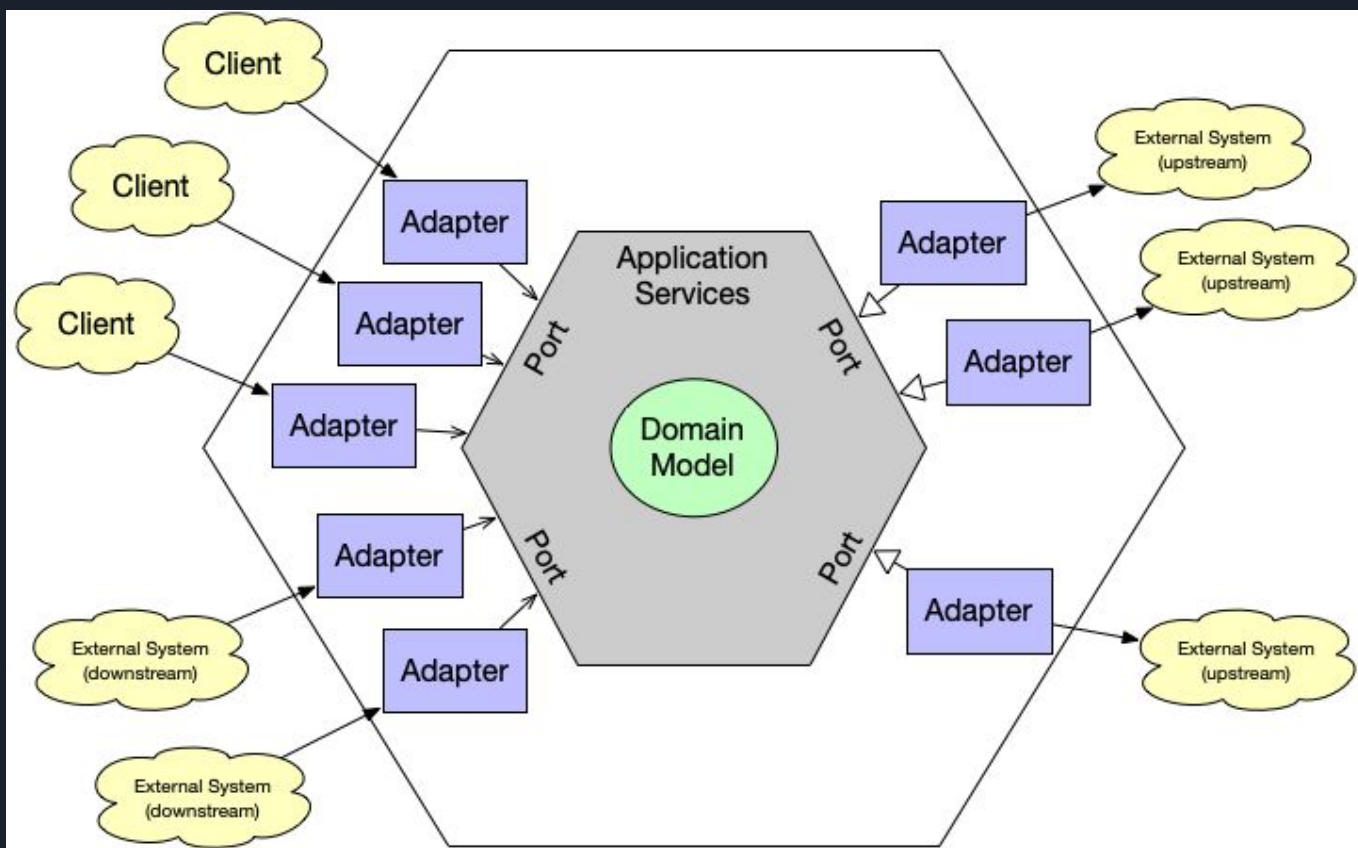
### Asynchronous one-to-one



### Polish/Subscribe



Communication between services

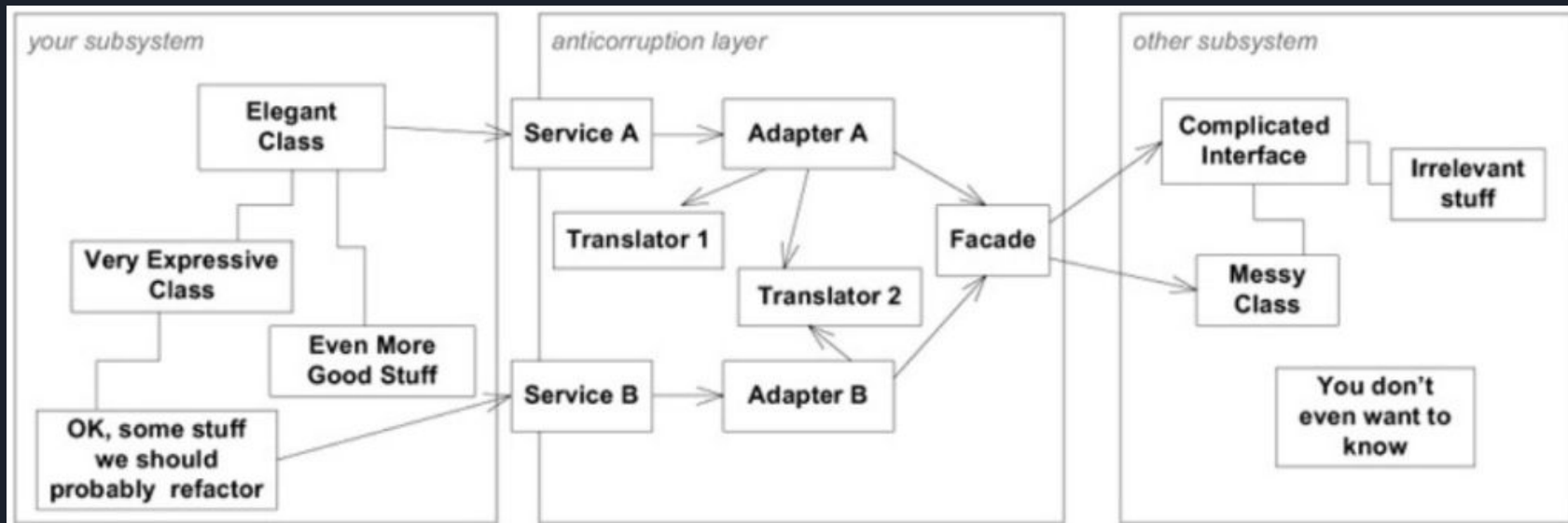


## DDD (Domain Driven Design)

- (Ubiquitous language)
- (Layers)
- (Bounded Contexts)
- (Anti-Corruption Layer)
- (Shared Kernel)
- (Generic Subdomain)







**Figure 14.8. The structure of an ANTICORRUPTION LAYER**

Anti-Corruption Layer

# How to start **DDD**?

## Ubiquitous Language

- Vocabulary of domain-specific terms
- Nouns, verbs, adjectives
- Shared by all parties involved in the project
- Primary goal of avoiding misunderstandings
- Used in all forms of spoken and written communication
- Universal language of business

