

# Principles Of Microservices

Alper  
Hankendi

 @alper\_hankendi

 [github.com/alperhankendi](https://github.com/alperhankendi)

# About Me



Working as Product Engineering Director at Hepsiburada

20 Years Experience... approximately (professional)

Ninja Developer, Servant Leader and Mentor

Like to build useless stuff

Drummer, Brewer and.... Father

Golang and Linux Lover

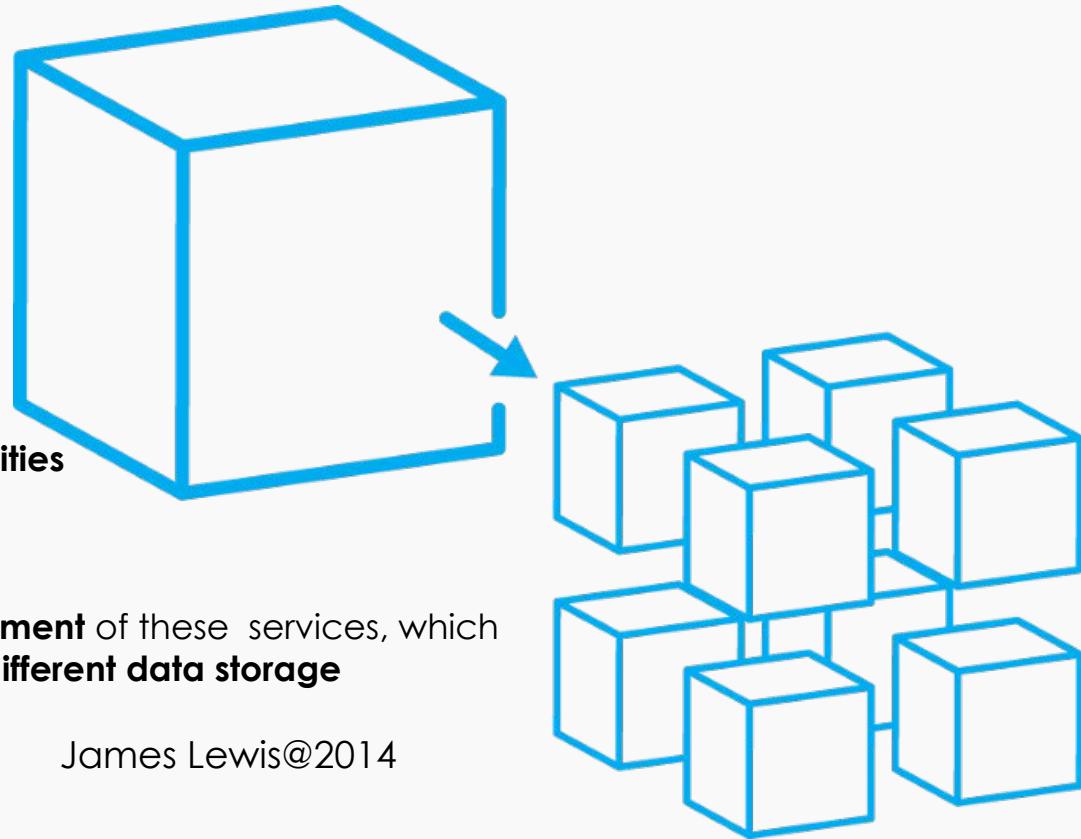
# AGENDA



- Definition of Microservices
- How big is a Microservices
- Scaling the organization
- Principles Of Microservices
  - Modelled Around business domain
  - Culture Of Automation
  - Hide Implementation Details
  - Decentralise All The Things
  - Deploy Independently
  - Isolate Failure
  - Consumer First
  - Highly Observable

# Definition Of Microservices

The microservice architectural style is an approach to developing a single application as a **suite of small service**, each **running in its own process** and communicating with lightweight mechanisms, often an HTTP resource API.



These services are **built around business capabilities** and **independently deployable** by fully automated deployment machinery.

There is a **bare minimum of centralized management** of these services, which may be **written in different languages** and **use different data storage** technologies.

James Lewis@2014



How Big is a Microservices

# Microservices Size Categories

“Micro”services

Ex: price, availability and shipping feature



“Mini”services

Ex: shopping cart domain calling the price, availability and shipping microservices

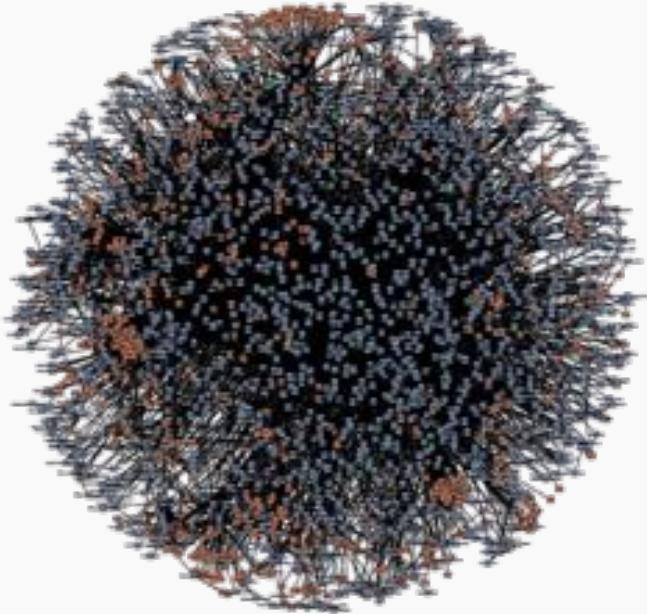


“Macro”services

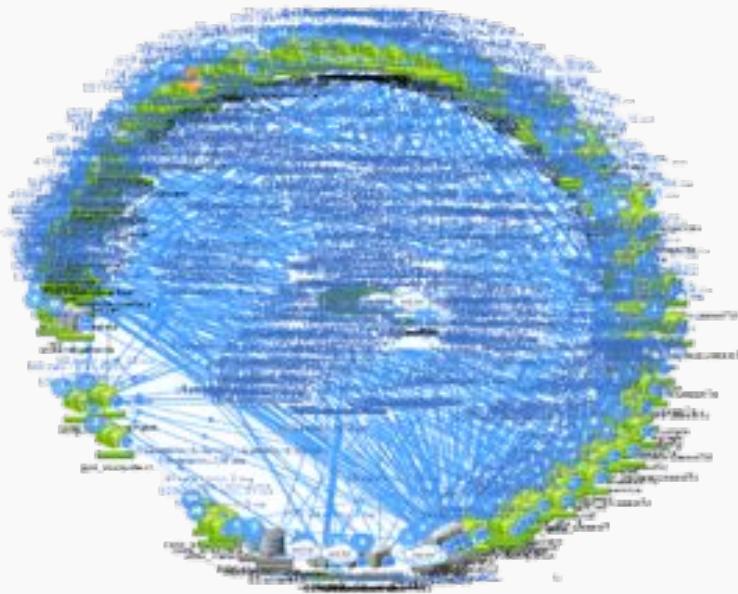
Ex: ordering business capability using different miniservices, such as the shopping cart && || payment



# Scaling in Amazon & Netflix

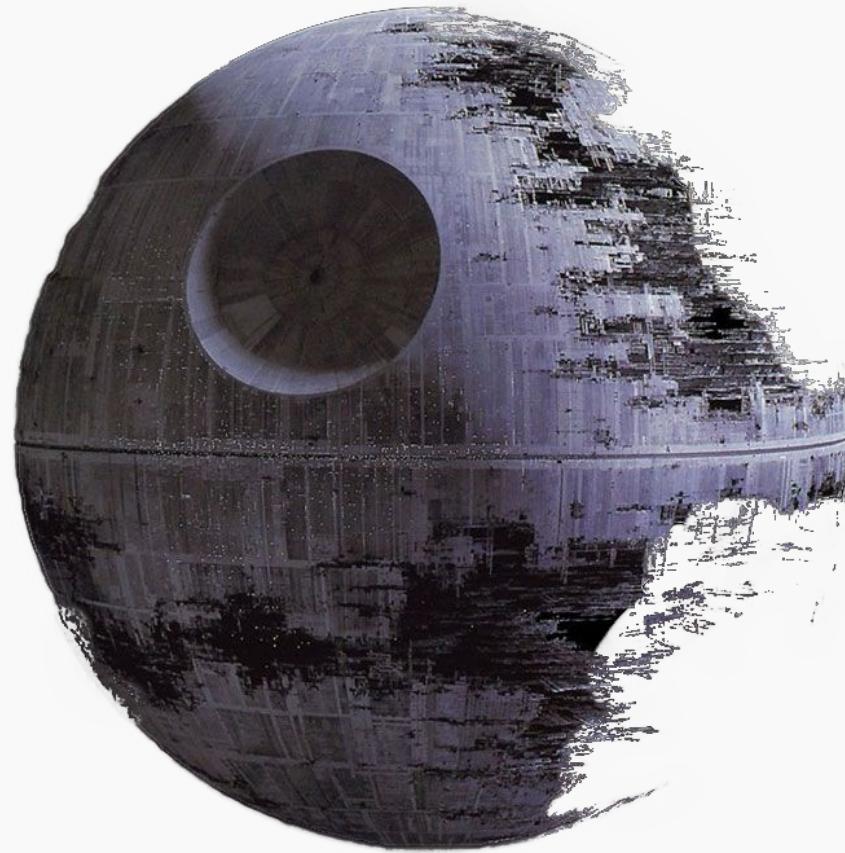


amazon.com®



NETFLIX

# Starwars... Death Star ?





Where to cut the services ?

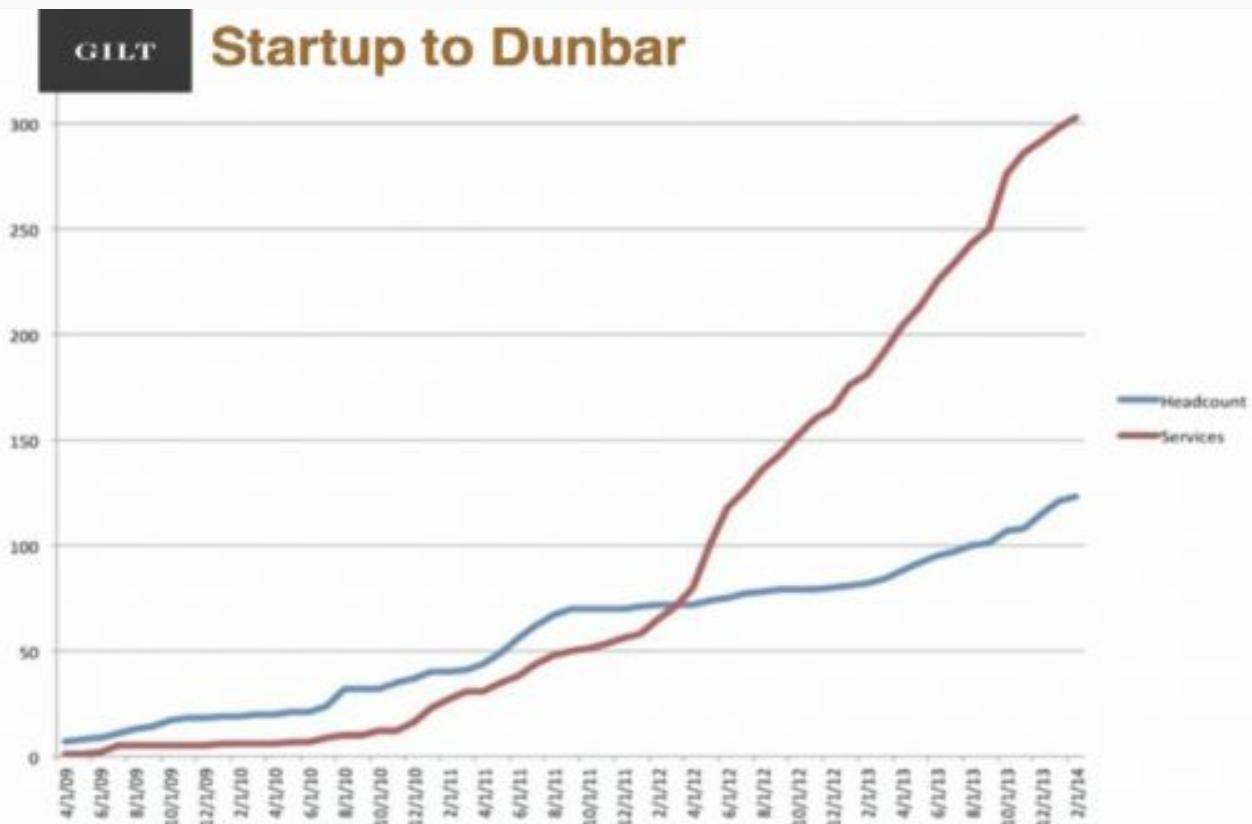
# Where to cut the services?

Social

Tech



# Where to cut the services?



# Three Things You Can Do...

1

Right-size your teams

2

Recognize Conway's Law

3

Enable unplanned innovation

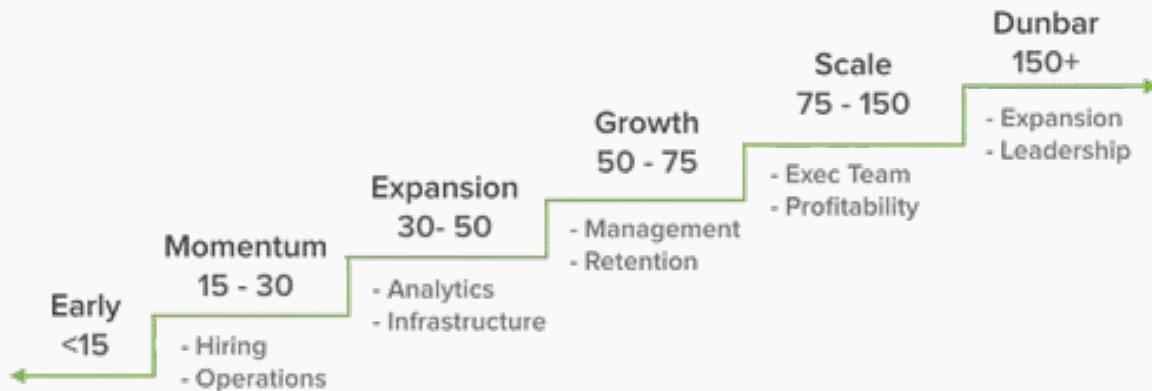
# Scaling the Organization

TL;DR



## Dunbar's Number

Aim for a team size of Dunbar Level 1 (5),  
Possibly Dunbar Level 2 (15)



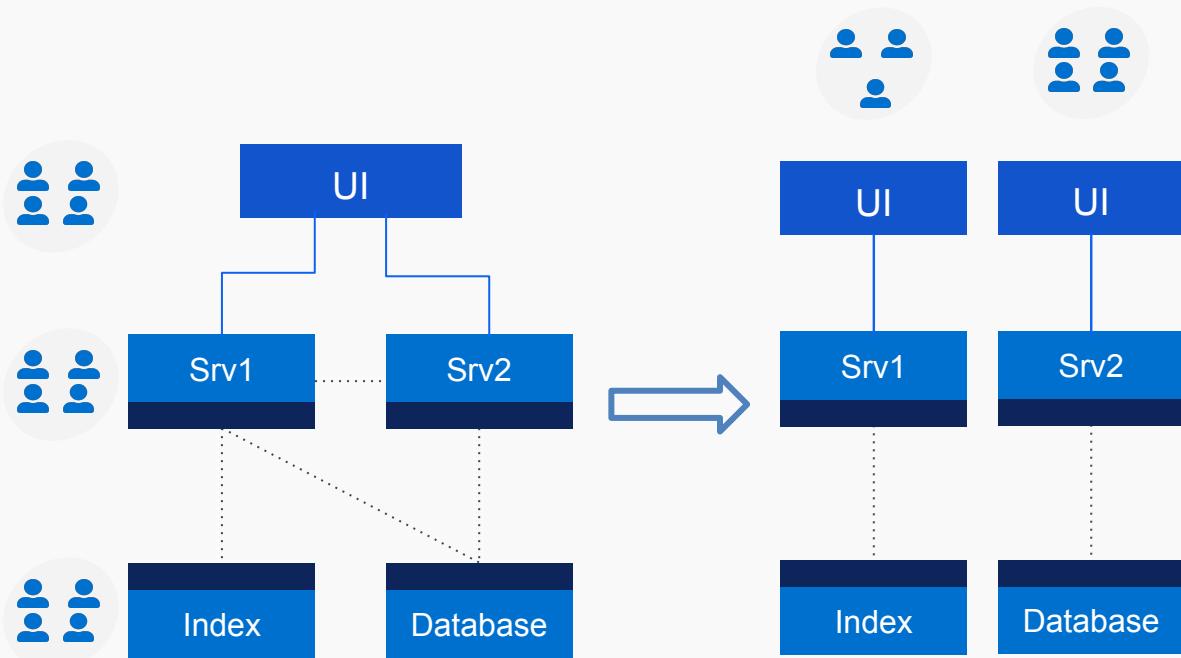
# Scaling the Organization

## 1- Right-sizing Teams



"A system's design is a copy of the organization's communication structure"

**Mel Conway@1967**



# Scaling the Organization

2- Conway's Law



If you want to achieve greatness,  
stop asking for permission

# Scaling the Organization

3- Unplanned Innovation





Principles Of Microservices

# Principles Of Microservices

TL;DR



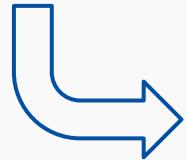
- 1 Modelled Around Business
- 2 Culture Of Automation
- 3 Deploy Independently
- 4 Decentralise All The Things
- 5 Hide Implementation Details
- 6 Isolate Failure
- 7 Consumer First
- 8 Highly Observable

# Principles Of Microservices

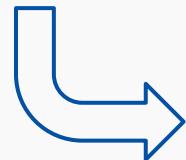
Modelled Around Business Domain



Analyze your domain



Define Bounded Context



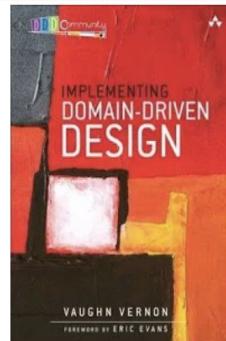
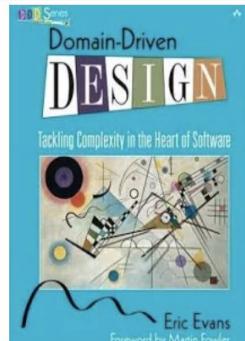
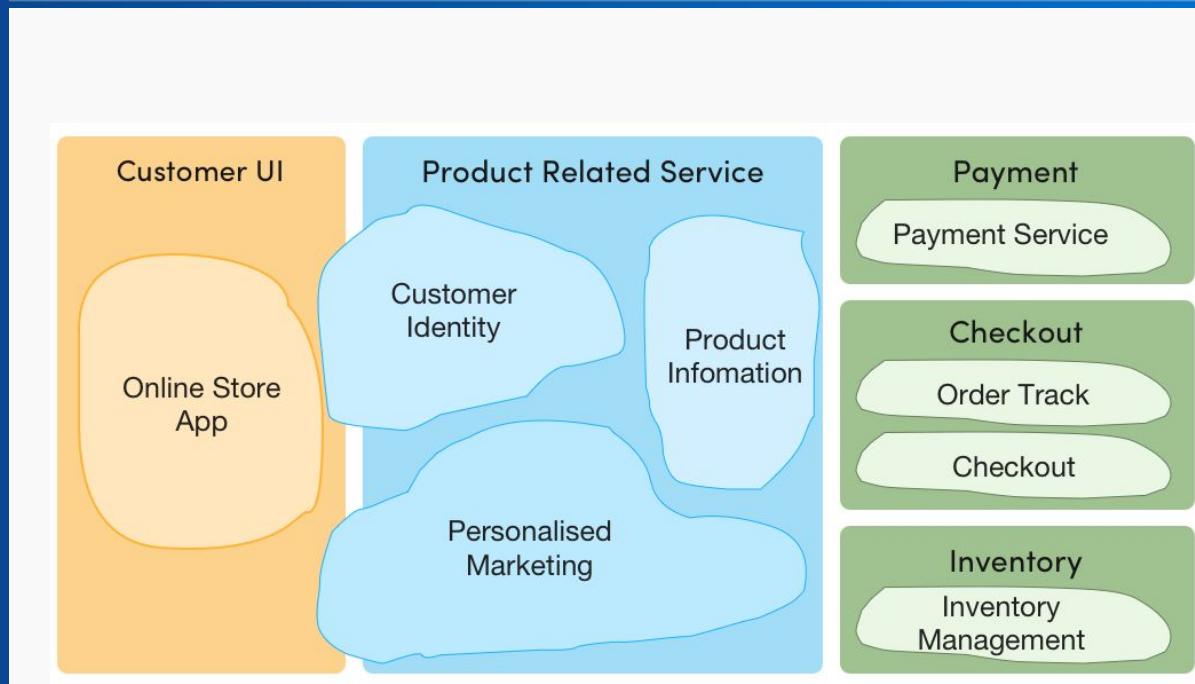
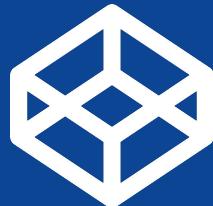
Define entities, aggregates, and services



Identify microservices

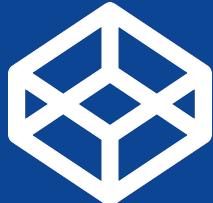
# Principles Of Microservices

Modelled Around Business Domain



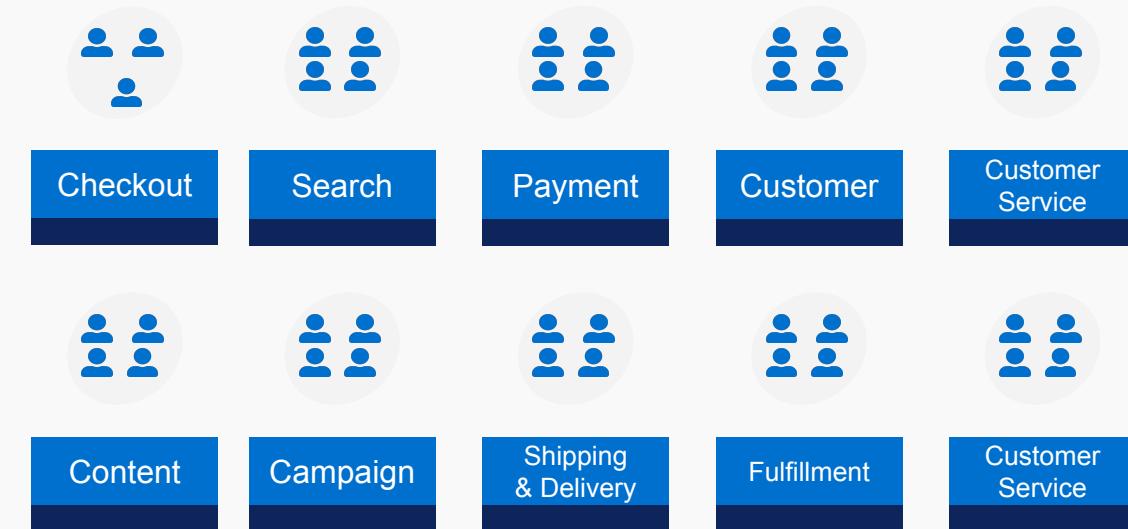
# Principles Of Microservices

Modelled Around Business Domain



## Products not Projects

per team per product



Amazon's notion of "you build, you run it". This brings developers into contact with the day-to-day operation of their software. It also brings them into day-to-day contact with the customer. This customer feedback loop is essential for improving the quality of the service.

# Principles Of Microservices

Culture Of Automation



## Infrastructure Automation

(provision an isolated operation system or service, database, cache service, e.g)



## Automated Testing

(sufficient testing in place that helps me understand whether I can release my software)



## Continuous Delivery

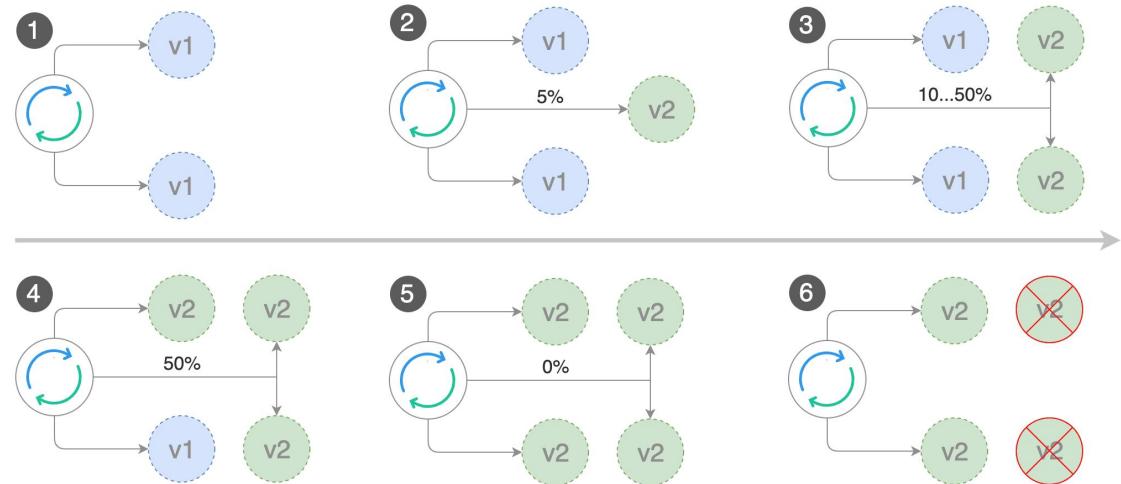
(treating every check-in as a release candidate , building deployment pipelines , canary)

# Principles Of Microservices

Culture Of Automation



## Canary Deployment



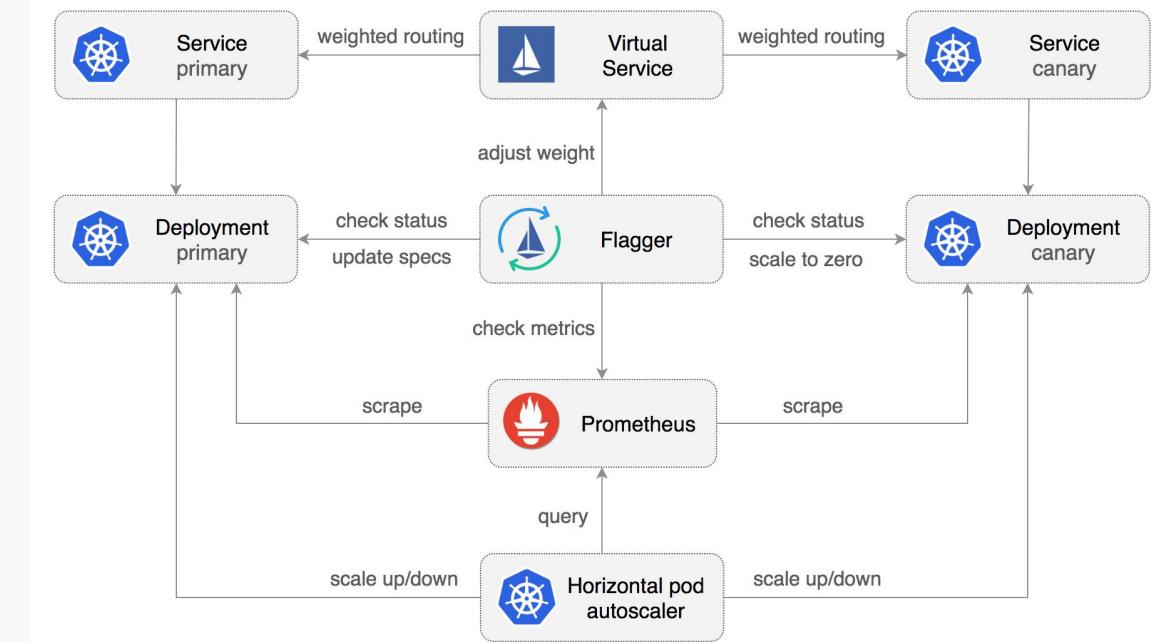
# Principles Of Microservices

Culture Of Automation



## Canary Deployment

Istio and Flagger to automate canary deployments



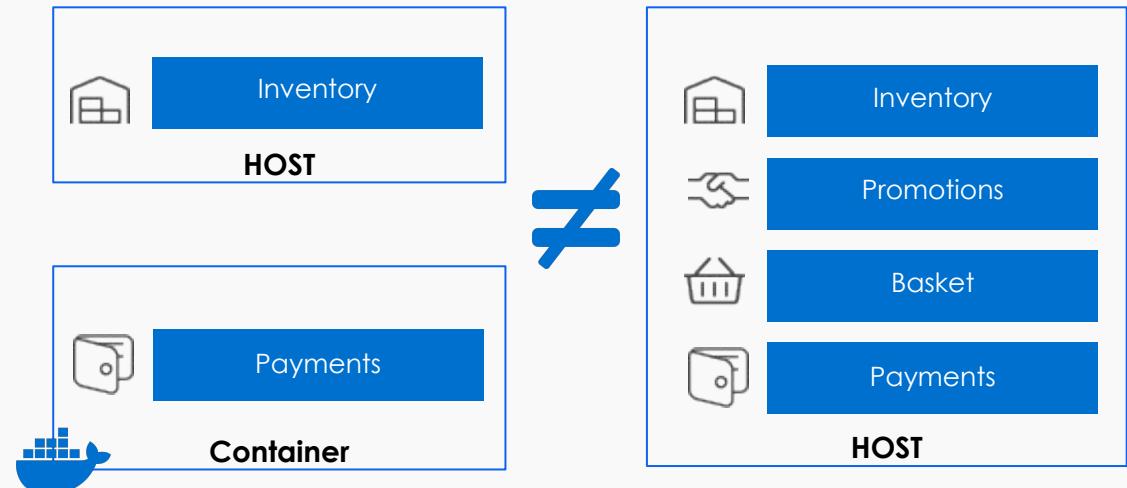
# Principles Of Microservices

Deploy Independently



## Single Service Instance Per Host

- Services are written using a variety of languages, frameworks, and framework versions
- Each service consists of multiple service instances for throughput and availability
- Service must be independently deployable and scalable
- Service instances need to be isolated from one another
- You need to be able to quickly build and deploy a service
- You need to be able to constrain the resources (CPU and memory) consumed by a service
- You need to monitor the behavior of each service instance
- You want deployment to reliable
- You must deploy the application as cost-effectively as possible

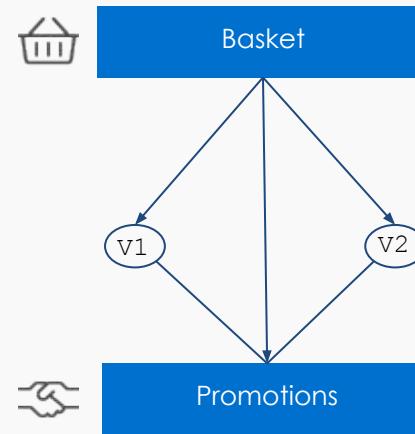


# Principles Of Microservices

Deploy Independently



## CO-Exist Endpoints



# Principles Of Microservices

Decentralise All The Things



Make your teams more Autonomous

What is autonomy ?

Giving people as much freedom as possible  
to do the job at hand



# Hepsiburada Tech stack



Uygulama & Veri (46)



JavaScript



nginx



Python



Node.js



React



Java



HTML5



PostgreSQL



DevOps (30)



GitHub



Git



Docker



Visual Studio Code



MongoDB



Ubuntu



AngularJS



Google Drive



Redis



ES6



C#



Vue.js



Kubernetes



Webpack



Visual Studio



IntelliJ IDEA



Ruby



React Native



Redux



Spring Boot



Go



RabbitMQ



Kafka



Microsoft SQL Server



Docker Compose



Kibana



Yarn



Selenium



CentOS



Material Design for Angular



Scala



Memcached



Cassandra



Hadoop



Apache Spark



R Language



npm



Jenkins



GitLab



Bitbucket



.NET Core



InfluxDB



Akka



HBase



Apache Hive



Druid



Hazelcast



Apache Storm



Vim



Varnish



PyCharm



Ansible



Apache NiFi



Echo



Scylla



Event Store



Hue



Apache Oozie



Vagrant



SourceTree



Grafana



Gradle



Docker Swarm



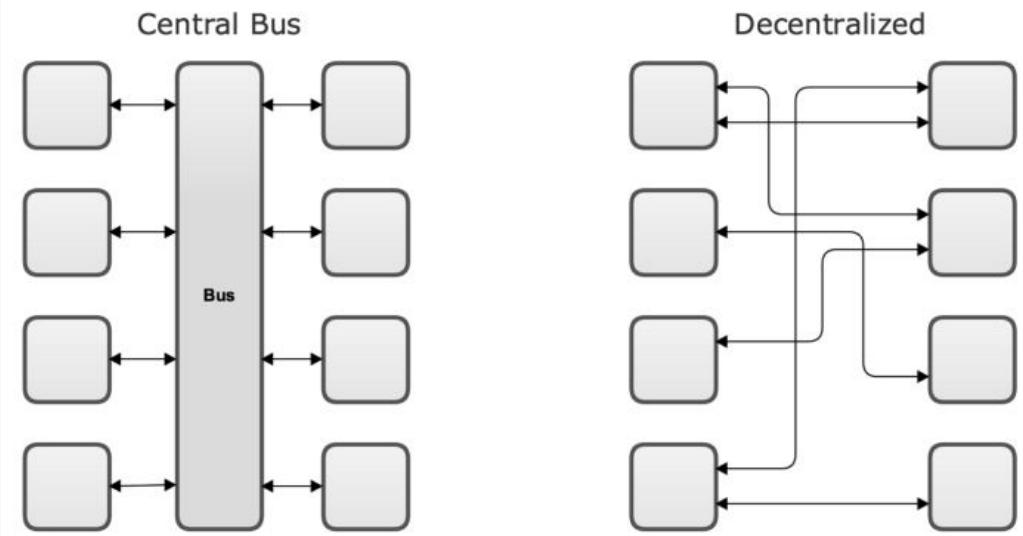
Jaeger

# Principles Of Microservices

Decentralise All The Things



Dumb-Pipes... Smart Endpoints...  
Anti-pattern: Centralized Service Bus



# Principles Of Microservices

Hide Implementation Details

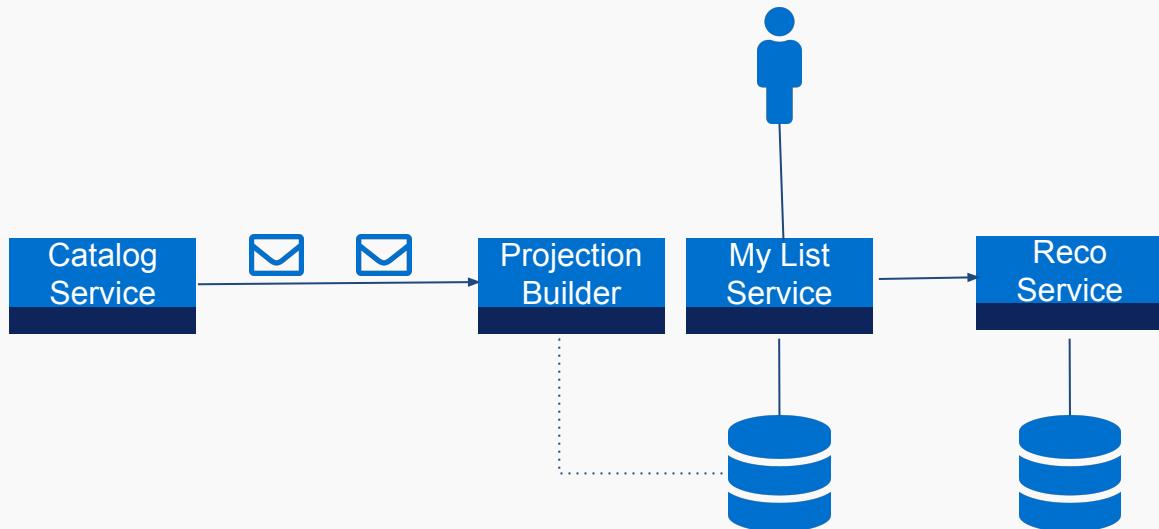


## Database Per Service

Service must be loosely coupled so that they can be developed, deployed and scaled independently.

Different services have different data storage requirements.

Keep each microservice's persistent data private to that service and accessible only via its API.

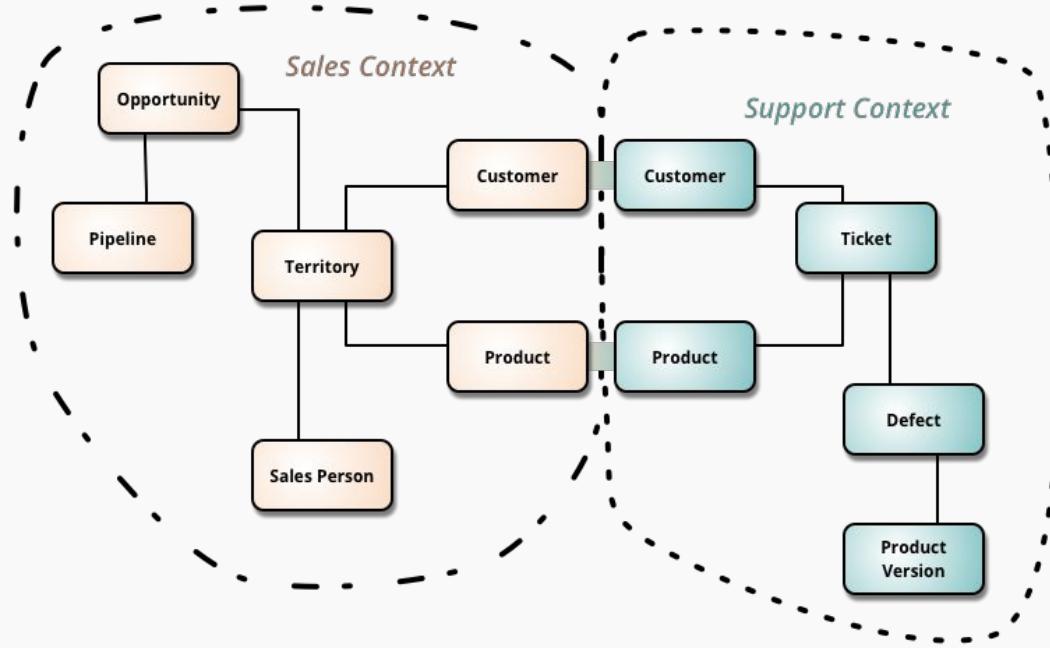


# Principles Of Microservices

Hide Implementation Details



## Bounded Context



Elektronik

Moda

Ev, Yaşam,  
Kırtasiye, OfisOto, Bahçe, Yapı  
MarketAnne, Bebek,  
OyuncakSpor,  
OutdoorKozmetik,  
Kişisel BakımSüpermarket,  
Pet ShopKitap, Müzik,  
Film, Hobi

Bilgisayar (1)

## Markalar

 Samsung (1)

## Kategoriler

 Led Monitörler (1)

## Fiyat Aralığı

En az - En çok

 2000 - 2500 TL (1)

## Ürün Puanı

 ★★★★☆ (1)

## Kullanım Amacı

 Ev Kullanıcısı (1)

## 3D Desteği

 Yok (1)

## Tepki Süresi

 4 ms (1)

Samsung LS27R750QEMXUF ile ilgili 1 ürün bulundu.

Kategoride ara:

Led Monitörler

Çok satanlar

En yeniler

En düşük fiyat

En yüksek fiyat

Diğer



★★★★☆ (7)

Samsung LS27R750QEMXUF 27" 144Hz  
(miniDisplay+HDMI) 2K QHD HAS 10bit...%31 3.599,00 TL  
**2.499,00 TL**

Search

**SAMSUNG**Teknolojileriyle hayatınıza fark  
katacak Samsung ürünler burada**Samsung LS27R750QEMXUF 27" 144Hz (miniDisplay+HDMI)  
2K QHD HAS 10bit Çerçevesiz VA Space Oyuncu Monitör  
Samsung****%31  
indirim**

3.599,00 TL

**2.499,00 TL**

Yorum (7) | Yorum Yap

- 220,93 TL x 12 ay'a varan Taksit Seçenekleri
- 12 ay'a varan vade ile Alışveriş Kredisi seçeneği

Satıcı: Samsung Türkiye | Puan 9.8

**- 1  
Adet +****Sepete Ekle**

- En geç 16 Haziran Salı günü kargoda

**Alışveriş  
Kredisi  
Seçeneği****Çoklu kredi  
kartı ile  
ödeme****hepsiJET  
Anlaşmalı  
Satıcı****Kargo  
Bedava****Beğen****Listeme ekle****Karşılaştır****Product Detail**



Herkese Açık • Alper Hankendi

# Monitor

4 ürün

Paylaş



Hepsini Seç

Hepsiburada

Beğendiklerim

Daha sonra alacağım

Listelerim

Monitor

duman keyfim

koltuk

+5 Liste Daha

BenQ Zowie XL2731 27"  
144Hz 1ms (DVI-D)

2.199,00 TL

ViewSonic VX2758-P-MHD  
27" 144Hz 1ms...

2.278,56 TL

LG 27GL650F-B 27" 144Hz  
1ms (HDMI+Display) Full H...

2.139,00 TL

Samsung LS27R750QEMXUF  
27" 144Hz...

2.499,00 TL

Takip Ettiklerim

My List



Sipariş No **917 264 575**

25 Mayıs Pts, 15:13



Sipariş tamamlandı

**55,80 TL**

Kredi Kartı



Sipariş No **730 695 370**

24 Mayıs Paz, 17:12



Sipariş tamamlandı

**2.639,04 TL**

Kredi Kartı



Tekrar al



Siparişi gizle



Çözüm merkezi



Samsung LS27R750QEMXUF 27" 144Hz  
(miniDisplay+HDMI) 2K QHD HAS 10bit Çerçevesiz VA  
Space Oyuncu Monitör

Satıcı: [Samsung Türkiye](#)

**2.639,04 TL**

Ürünü değerlendir



**Alper Hankendi** adlı kişiye teslim edildi

Teslim tarihi **28 May Per, 12:00**



Kargo Takibi



Talep Oluştur

Order Preview

# Principles Of Microservices

Consumer First



API Documentation

**store** Access to Petstore orders

**POST** /store/order Place an order for a pet

**GET** /store/order/{orderId} Find purchase order by ID

**DELETE** /store/order/{orderId} Delete purchase order by ID

**GET** /store/inventory Returns pet inventories by status

**USER** Operations about user

Find out more about our store: <http://swagger.io>

**POST** /user/createWithArray Creates list of users with given input array

Parameters

Try it out

Name	Description
<b>body</b> * required array[object] (body)	List of user object <a href="#">Example Value</a>   <a href="#">Model</a>

```
[  
  {  
    "id": 0,  
    "username": "string",  
    "firstName": "string",  
    "lastName": "string",  
    "email": "string",  
    "password": "string",  
    "phone": "string",  
    "userStatus": 0  
  }  
]
```

Parameter content type

Responses

Response content type

Code	Description
default	successful operation

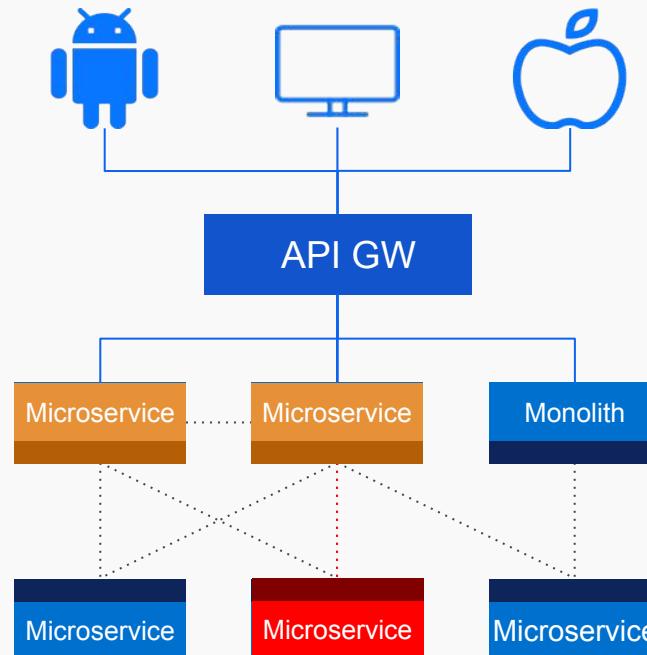
# Principles Of Microservices

Isolate Failure



## Strategies to handle partial failure

- Grained Timeout
- Health Checks API
- Caching



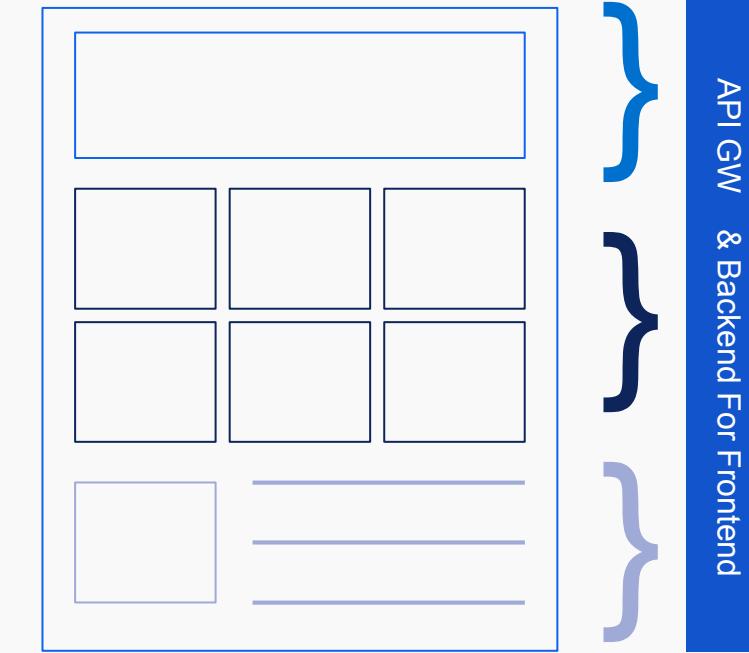
# Principles Of Microservices

Isolate Failure



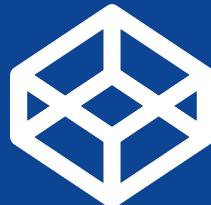
## Strategies to handle partial failure

- Circuit Breakers (even on your database)
- Bulkhead
- Throttling ( Rate limiting, User Quota )



# Principles Of Microservices

Back to  
Consumer First



Question : Should I Provide a client library for access to my service...

Well, It depends

Client libraries can be useful for many reasons, one of which is isolating the specifics of communicating with a remote resource. For service registries, client libraries like those provided by Consul or Netflix Eureka handle service registration and heart beating.

Other libraries, such as Netflix Ribbon, provide client-side load balancing.

But not good in polyglot development ecosystem.

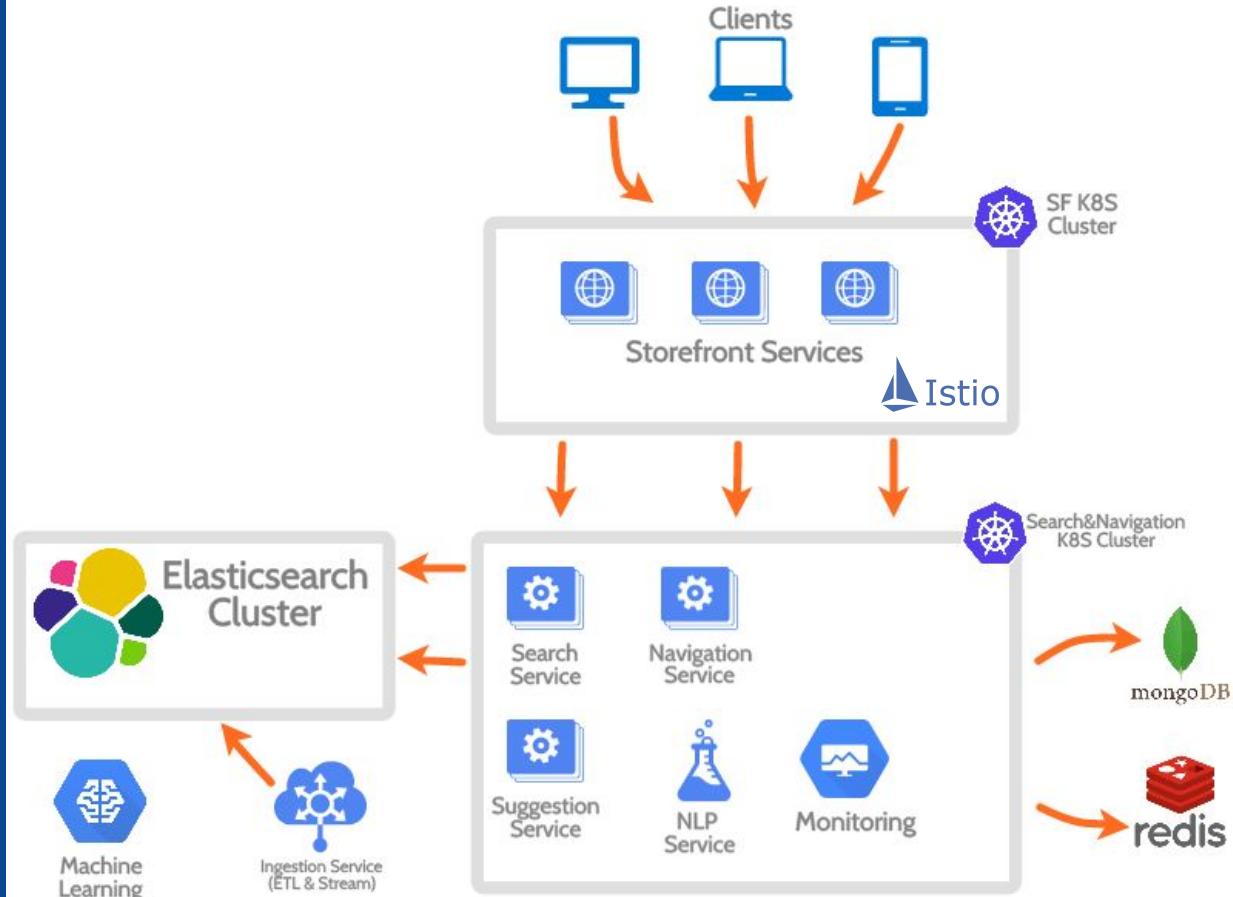
# Principles Of Microservices

Isolate Failure  
Need more Resilience!



## Containerization

- K8S, Istio (Service Mesh) , pod2pod communication thx to etcd



# Principles Of Microservices

Observable



## 3 Pillars of Observability



Logging  
Splunk , ELK

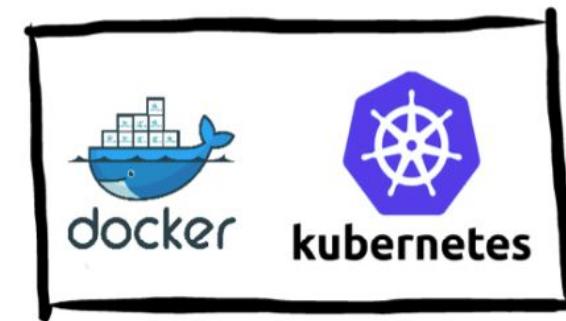
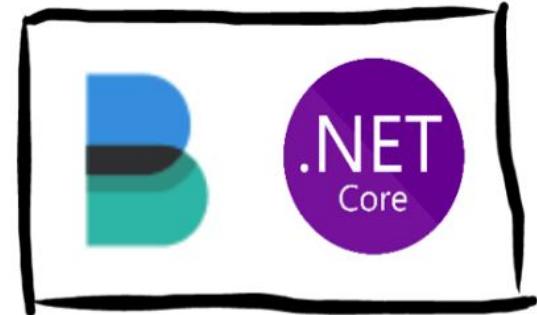
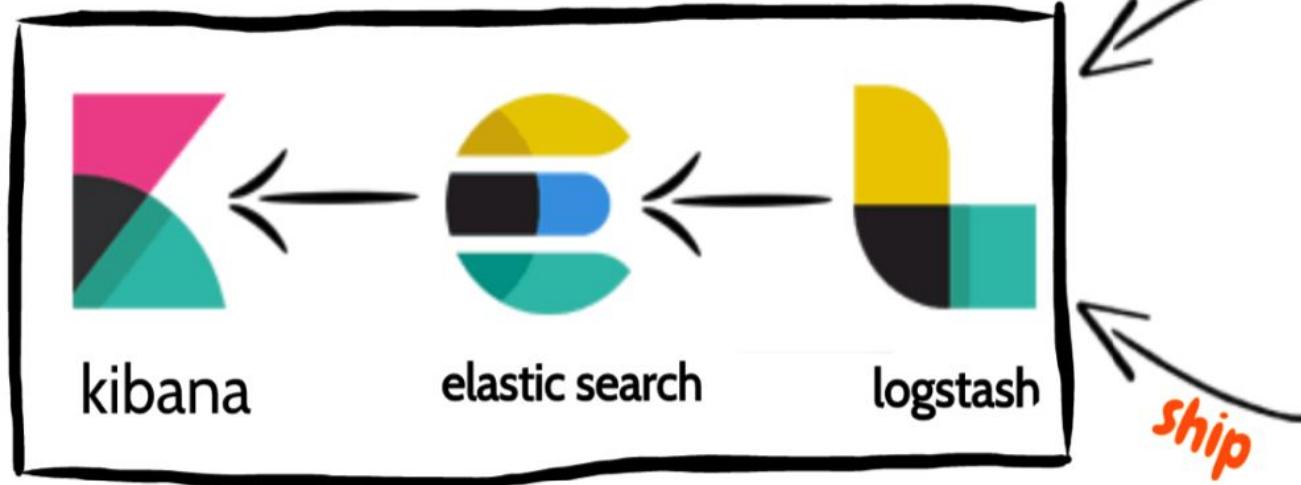


Metrics  
Prometheus , statsd , Grafana

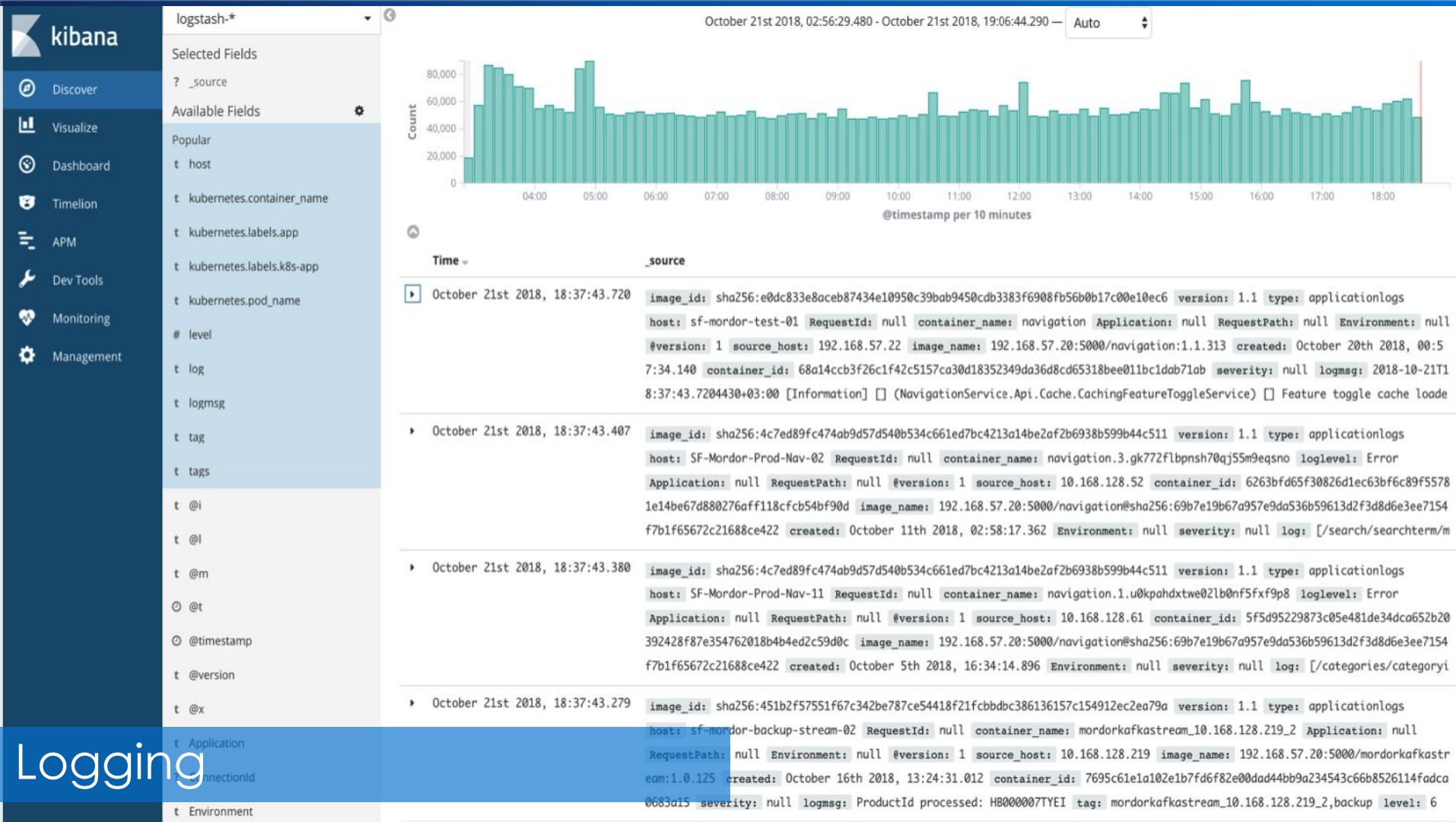


Distributed Tracing  
Jaeger, Zipkin , Kiali

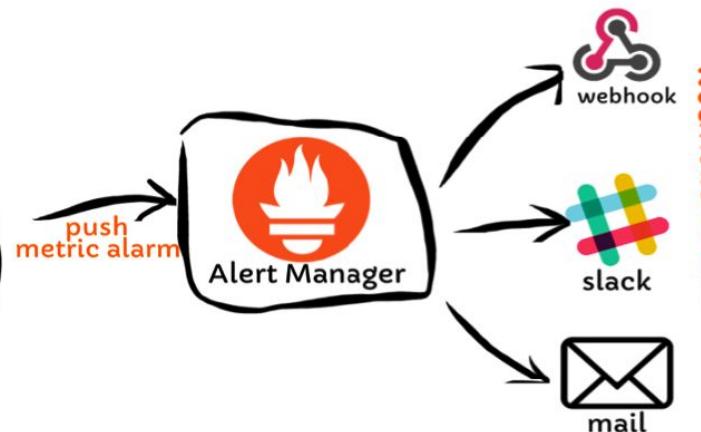
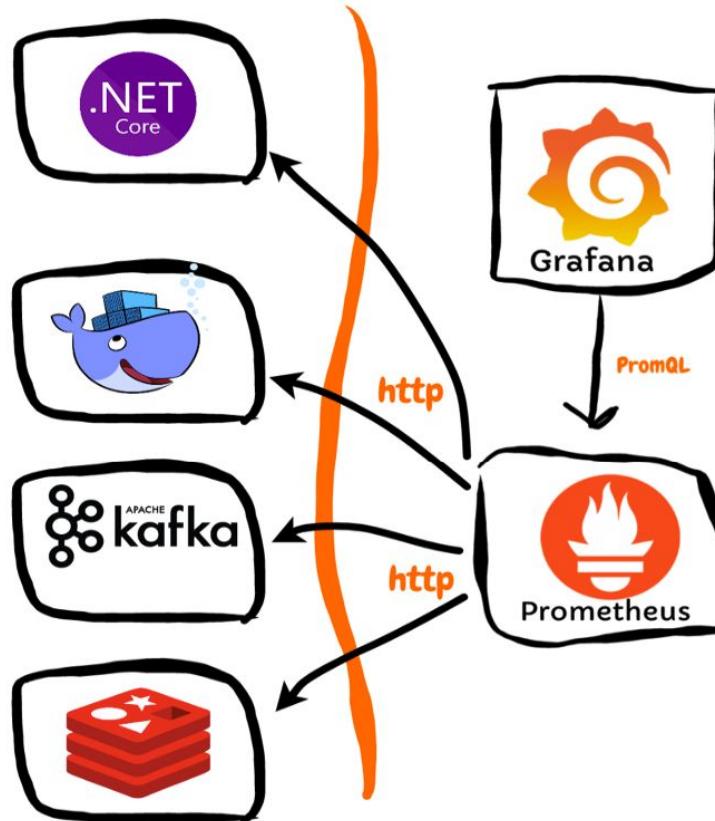
# ELK Stack Server



Logging



## Targets / Exporters



## Notification Channels

Metrics



environment

All ▾



/Product Throughput



/ProductDescription Thro...



/Category Throughput



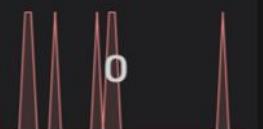
/DefinitionAttribute Thro...



/CompatibleProducts Th...



/BuyingCategory Thro...



API RPS



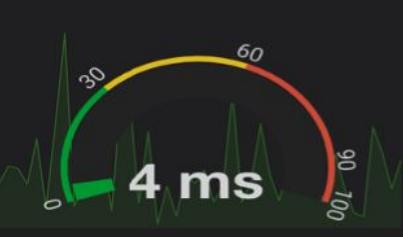
Average Response Time



/Product Response



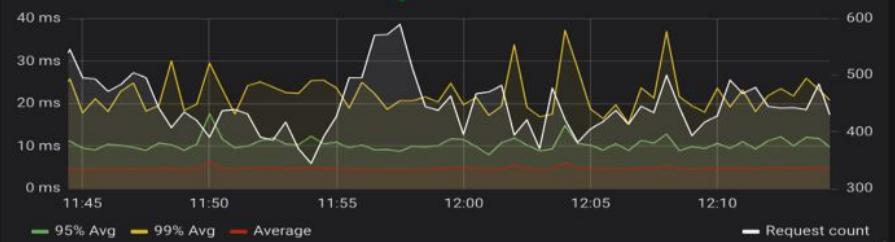
/Category Response



/ProductDescription Res...



/Product



/ProductDescription



/Category



/CompatibleProducts



Metrics

## SERVICE: co-basket-api-service.default.svc.cluster.local

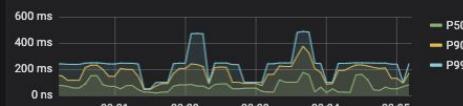
Client Request Volume

1.1 ops

Client Success Rate (non-5xx responses) ▾

100%

Client Request Duration



TCP Received Bytes

N/A

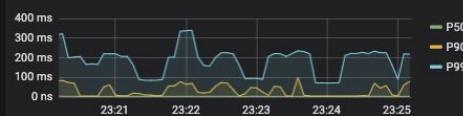
Server Request Volume

4.3 ops

Server Success Rate (non-5xx responses)

100%

Server Request Duration

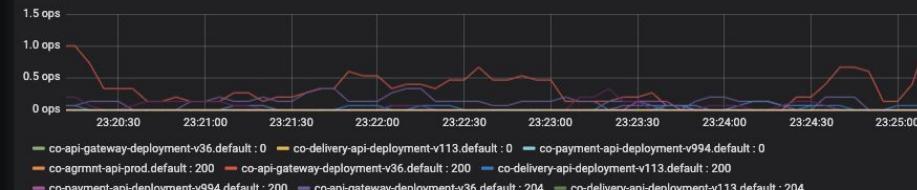


TCP Sent Bytes

N/A

## CLIENT WORKLOADS

Incoming Requests by Source And Response Code



Incoming Success Rate (non-5xx responses) By Source



Metrics

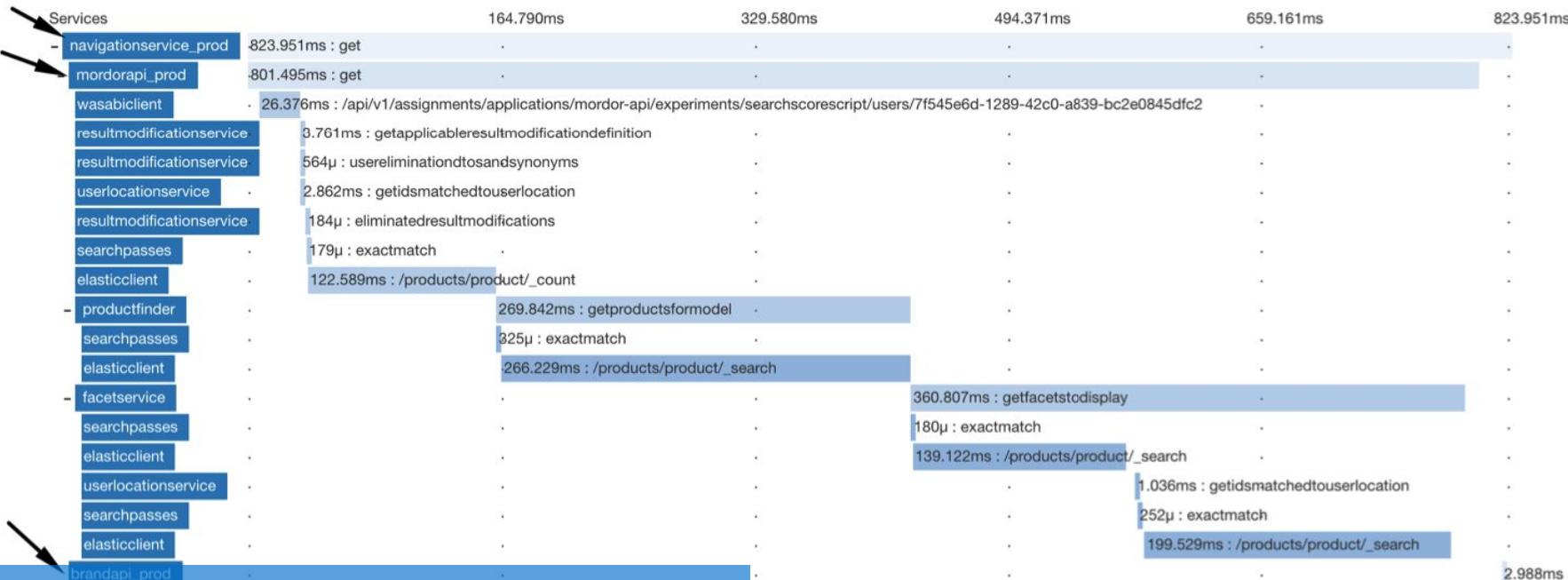
Duration: 823.951ms

Services: 10

Depth: 4

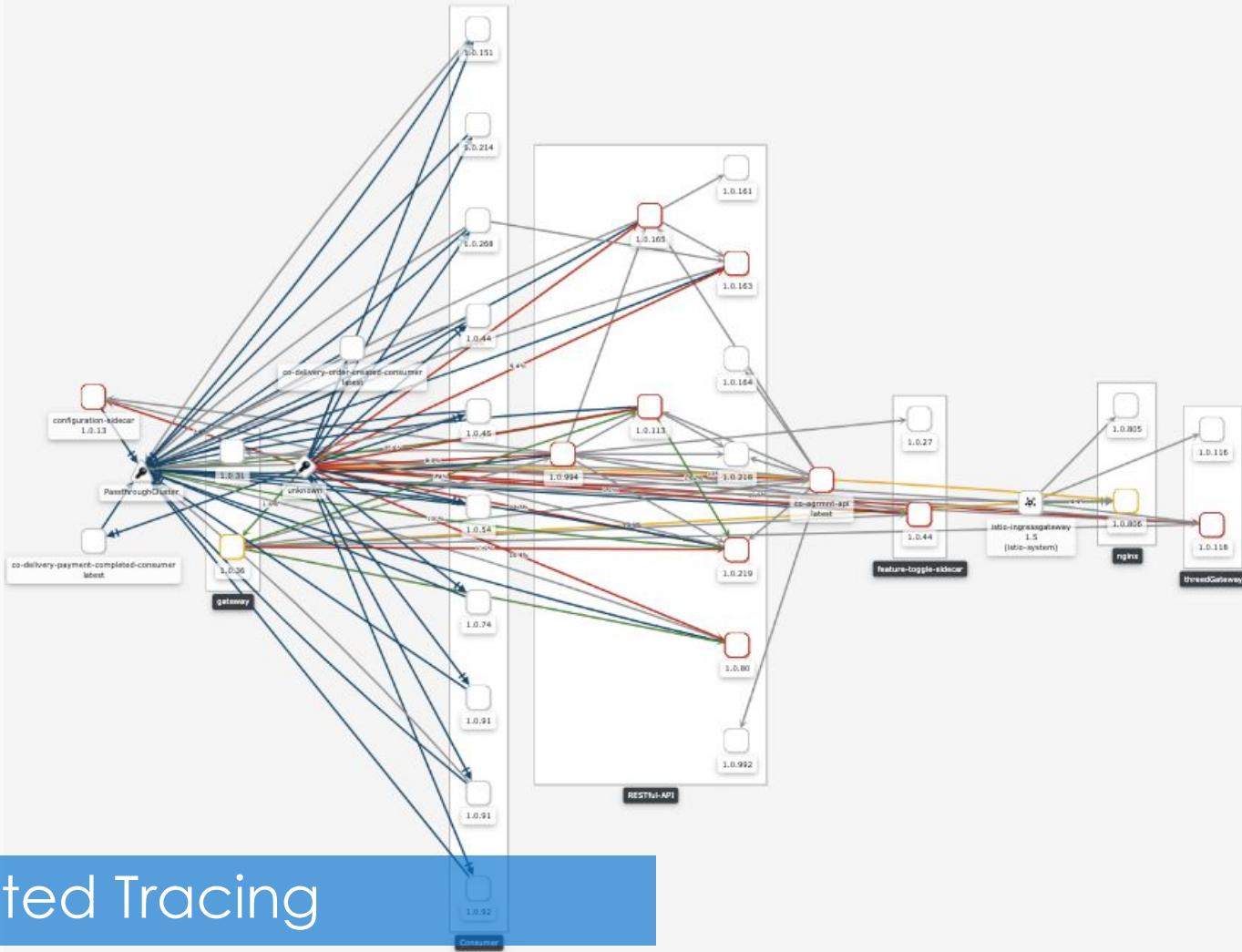
Total Spans: 19

JSON

[Expand All](#)[Collapse All](#)[Filter Service Search ▾](#)[brandapi\\_prod x1](#) [elasticclient x4](#) [facetservice x1](#) [mordorapi\\_prod x1](#) [navigationservice\\_prod x1](#) [productfinder x1](#) [resultmodificationservice x3](#) [searchpasses x4](#) [userlocationservice x2](#) [wasabclient x1](#)

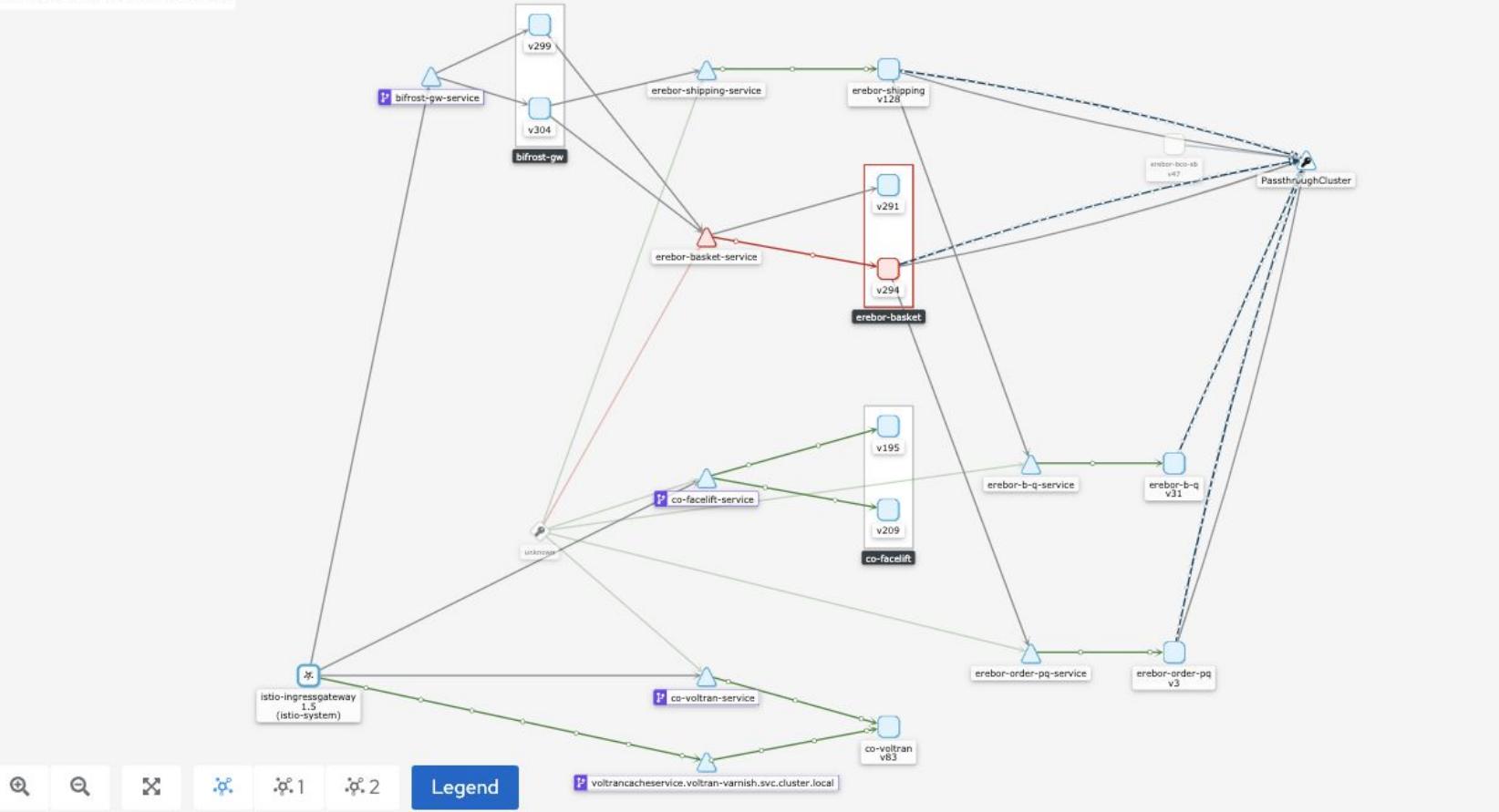
# Distributed Tracing

Hide



# Distributed Tracing

Jun 12, 12:28:30 AM ... 12:29:30 AM



# Distributed Tracing

# Questions?



@alper\_hankendi



[github.com/alperhankendi](https://github.com/alperhankendi)