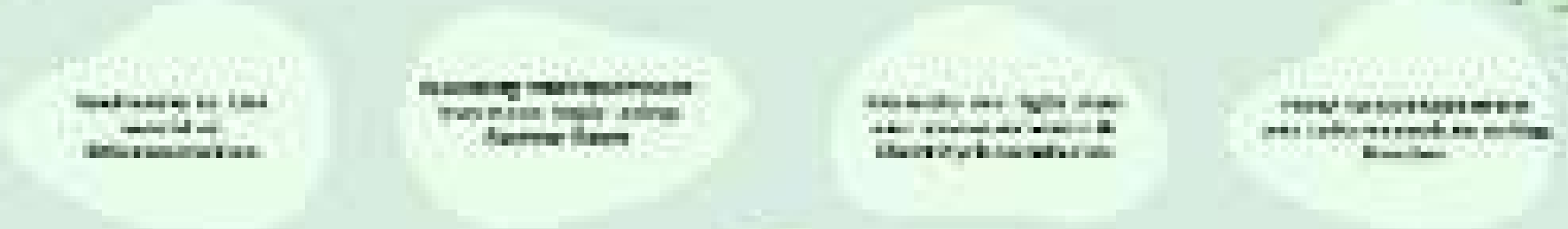
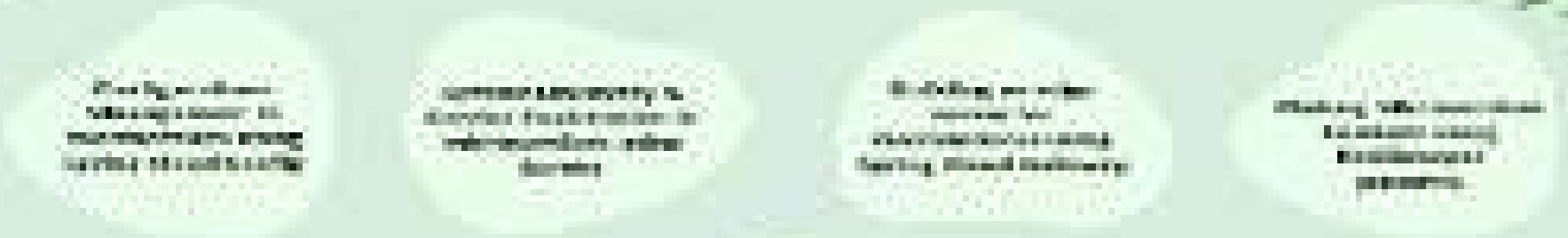
MICROSERVICES

WITH SPRINGBOOT, DOCKER, KUBERNETES

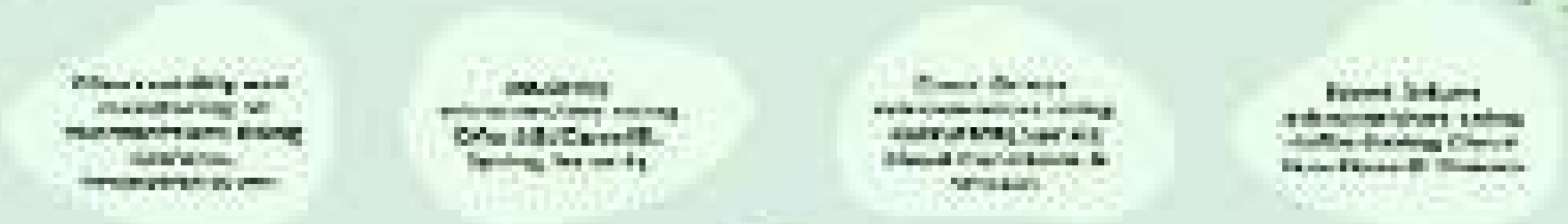
COURSE AGENDA



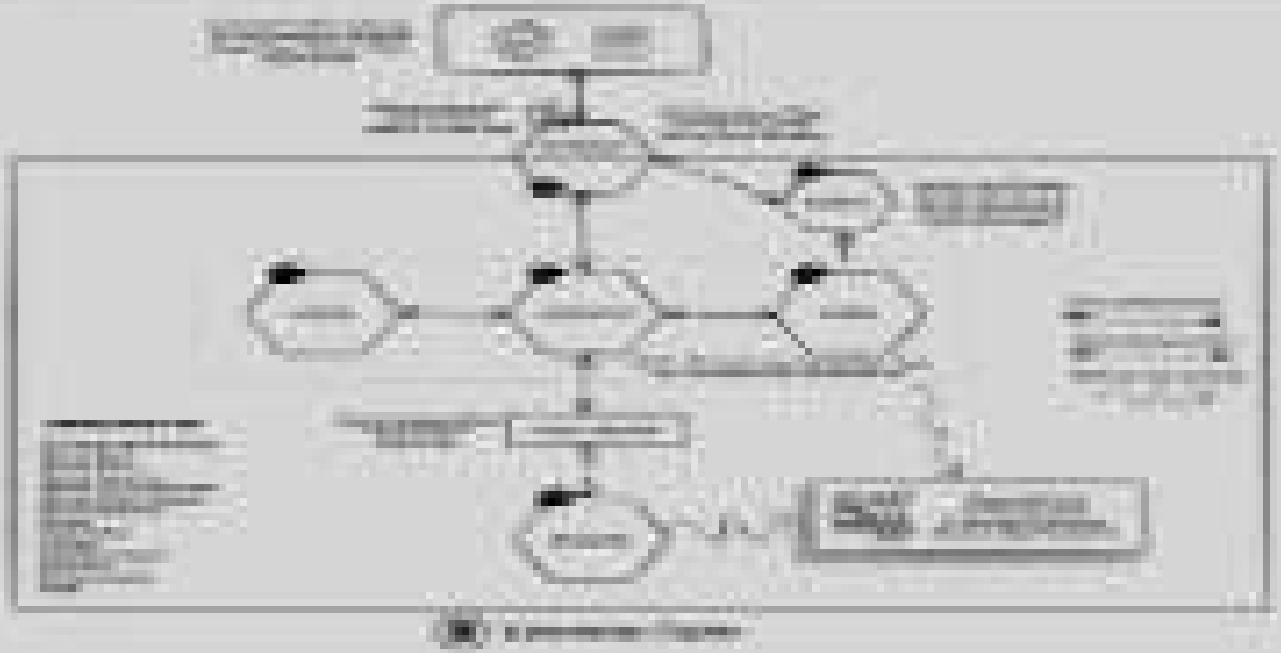
COURSE AGENDA



COURSE AGENDA

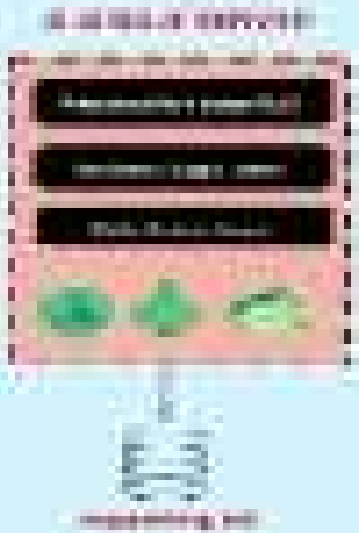


COURSE AGENDA



What are Microservices ?

The Monolith



Pros

Cons

The Monolith

The SOA

[illegible]

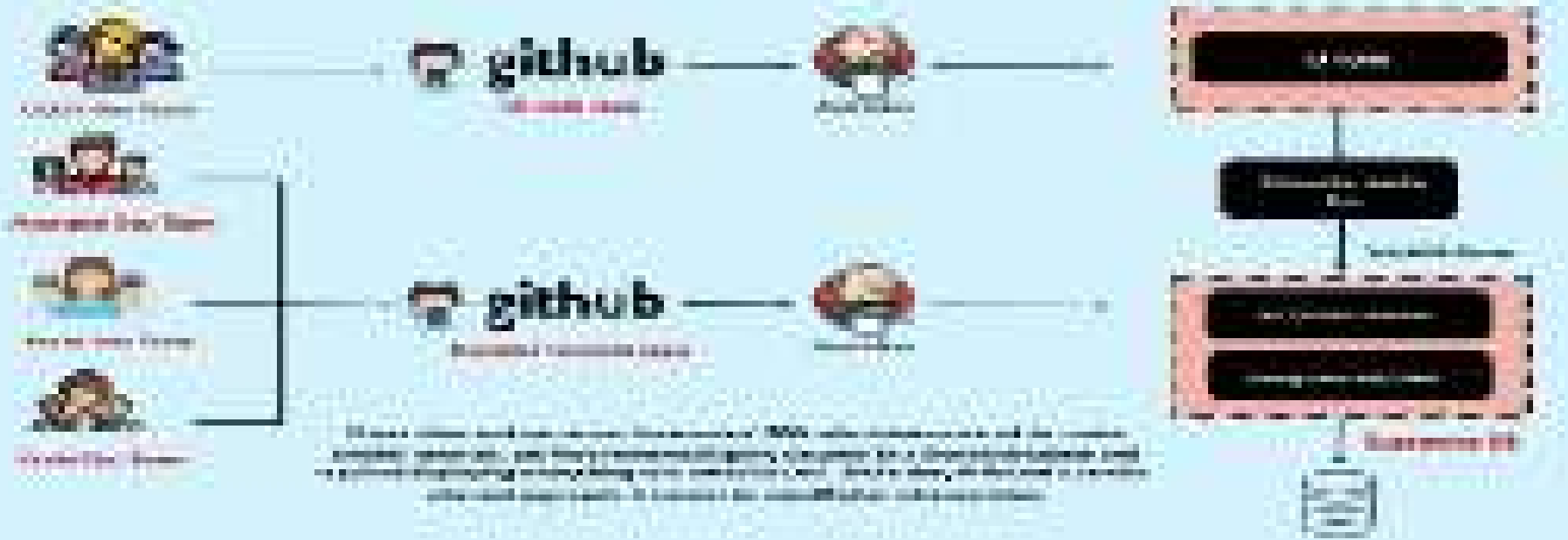
Pros

[illegible]

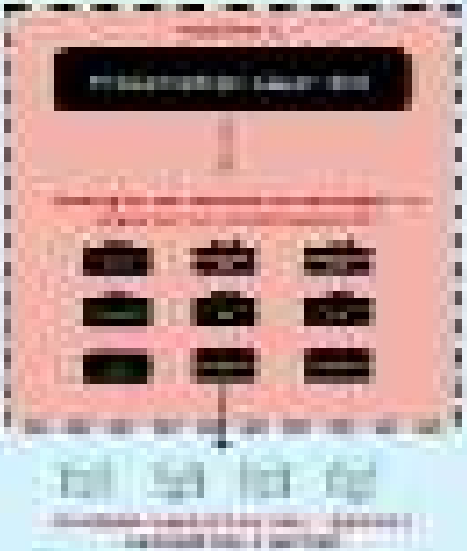
Cons

[illegible]

The SOA



The GREAT MICROSERVICES



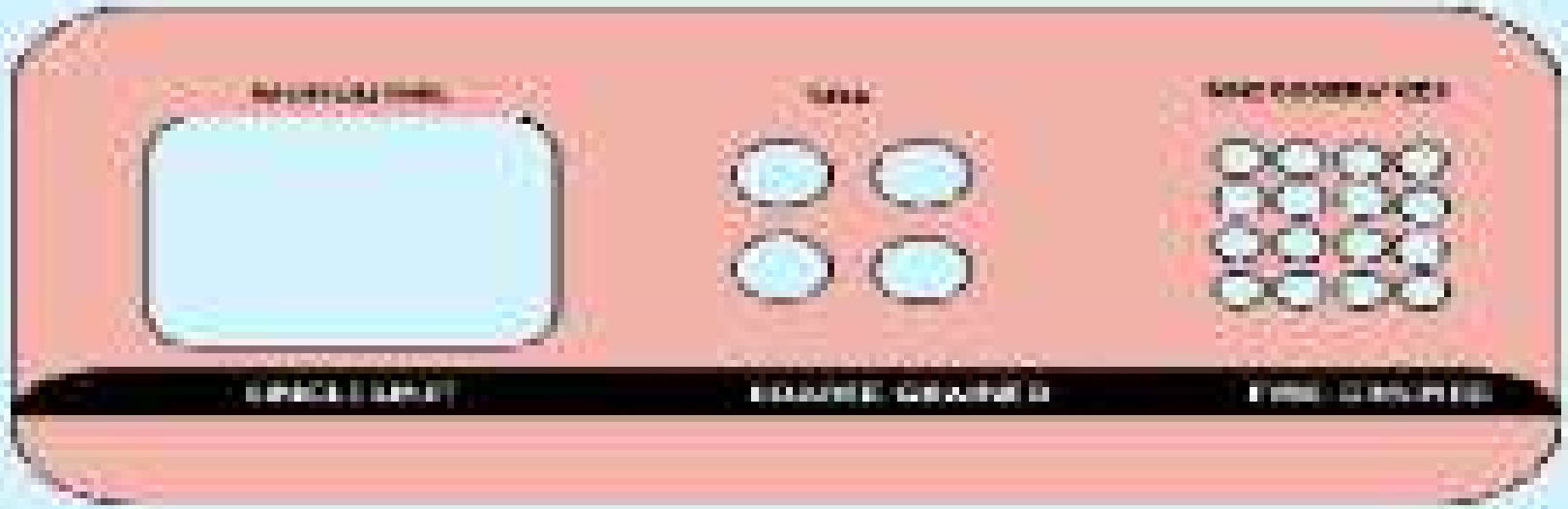
Pros

Cons

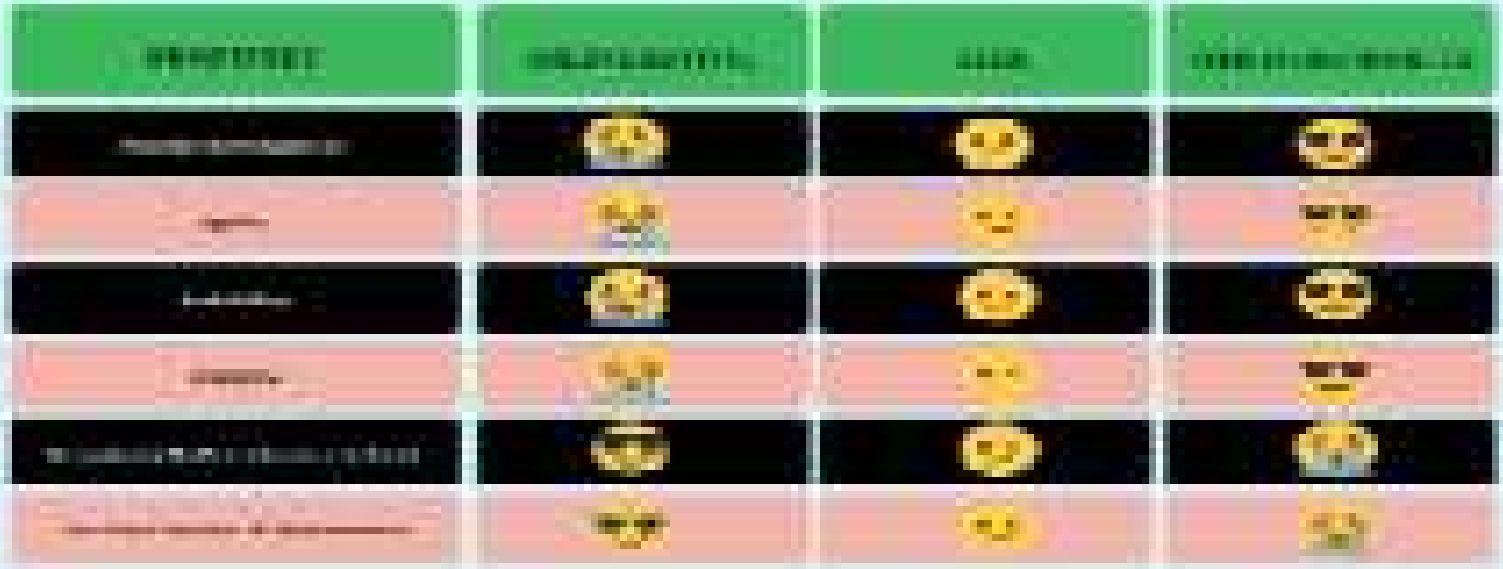
The GREAT MICROSERVICES

MONOLITHIC vs SOA vs MICROSERVICES

MONOLITHIC vs SOA vs MICROSERVICES



MONOLITHIC VS SOA VS MICROSERVICES

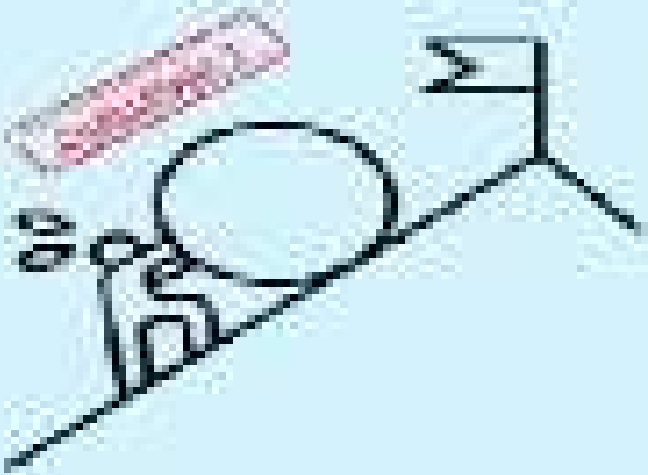


DEFINITION OF MICROSERVICE ?

"Microservices is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, built around business capabilities and independently deployable by fully automated deployment machinery."

– Martin Fowler by James Lewis and Martin Fowler

HOW TO BUILD MICROSERVICES ?



[illegible]

[illegible]

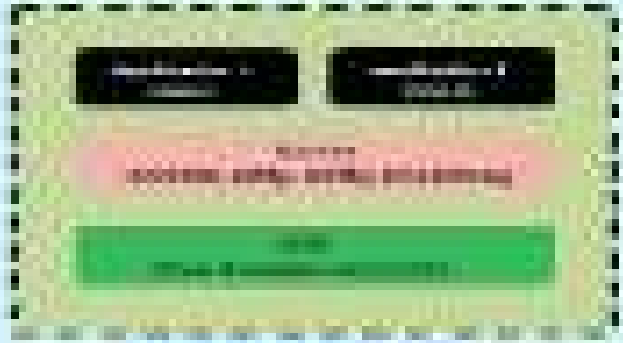
[illegible]

How to overcome the challenge?

The answer is **Spring Boot**

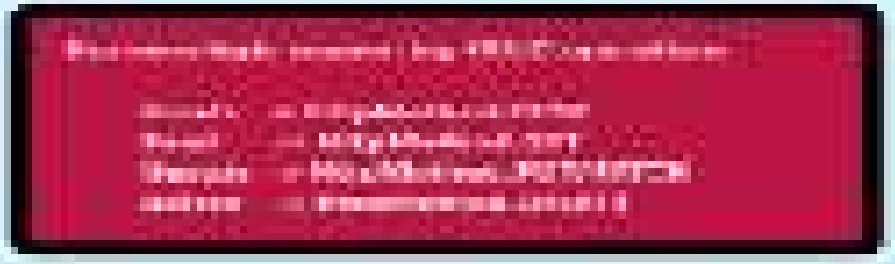
WHY SPRING BOOT FOR MICROSERVICES?

WHY SPRING BOOT FOR MICROSERVICES?

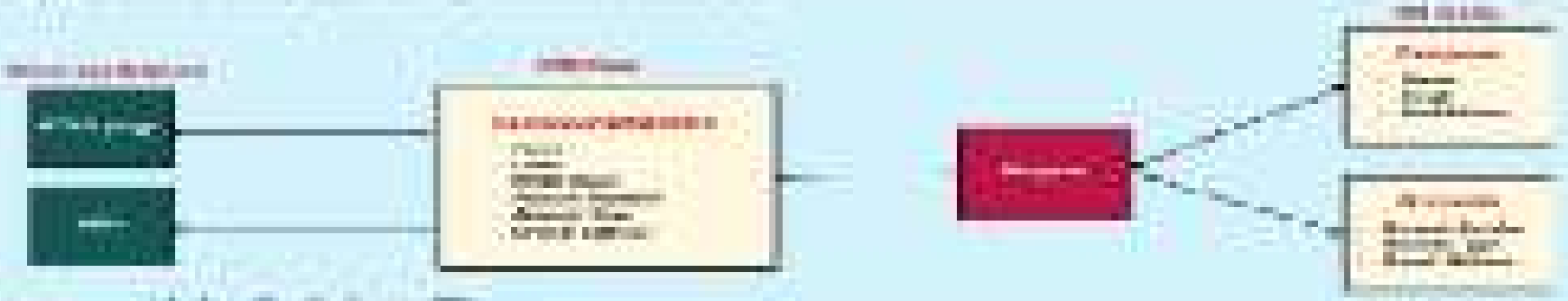


Implementing REST Services

Implementing REST Services



DTO (Data Transfer Object) pattern

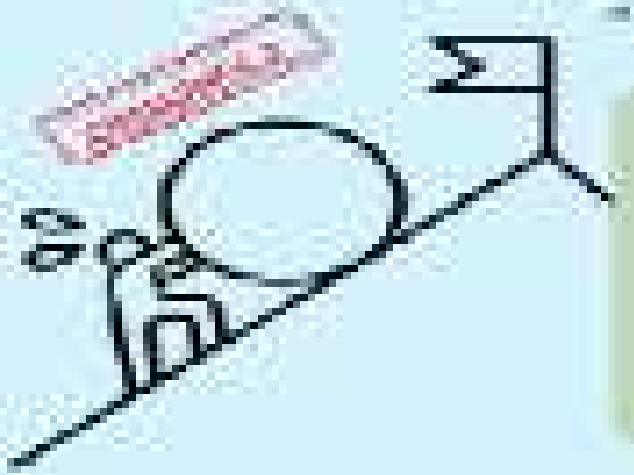


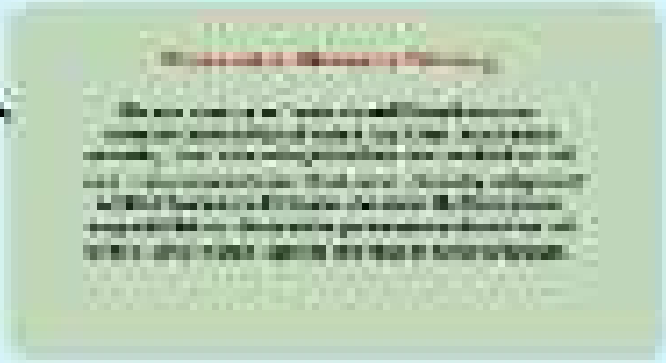
Different Annotations & Classes that supports building REST services

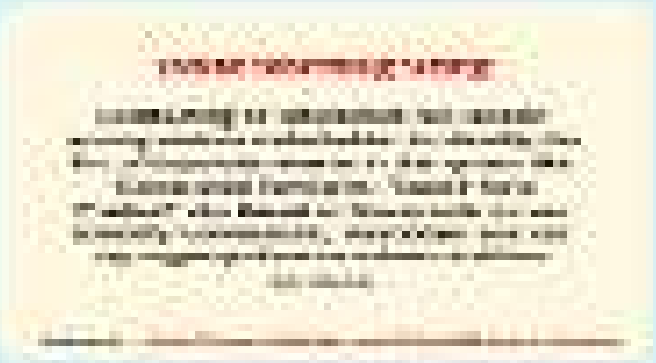
Summary of the steps Followed to build microservices

S
M
L
XL
XXL

HOW TO RIGHT SIZE & IDENTIFY SERVICE BOUNDARIES IN MICROSERVICES ?

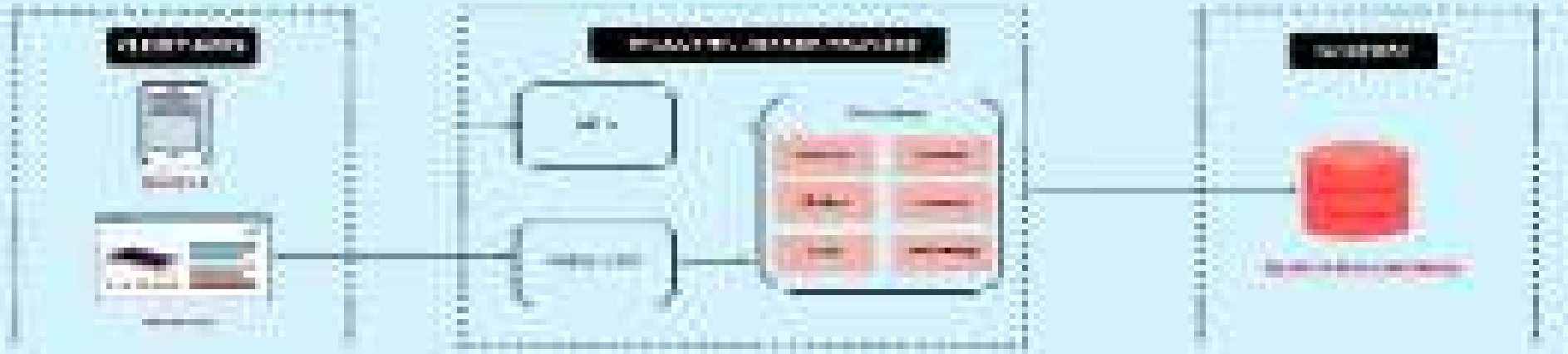






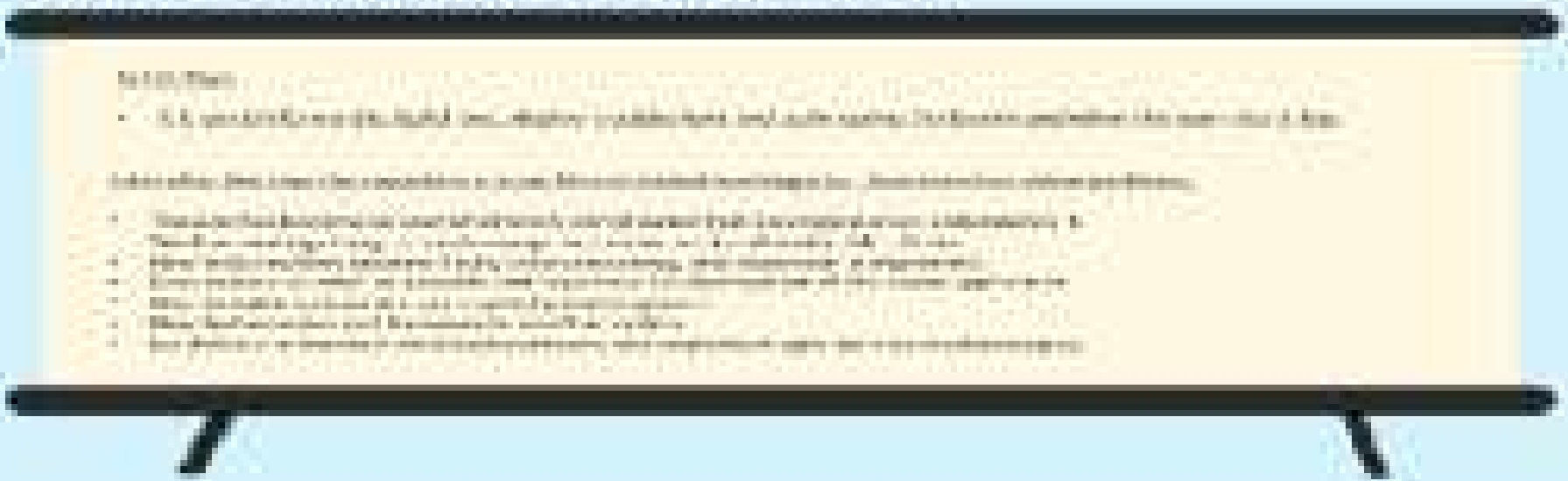
RIGHT SIZING & IDENTIFYING SERVICE BOUNDARIES

MONOLITHIC TO MICROSERVICES



MONOLITHIC TO MICROSERVICES

IN REAL-TIME USAGE

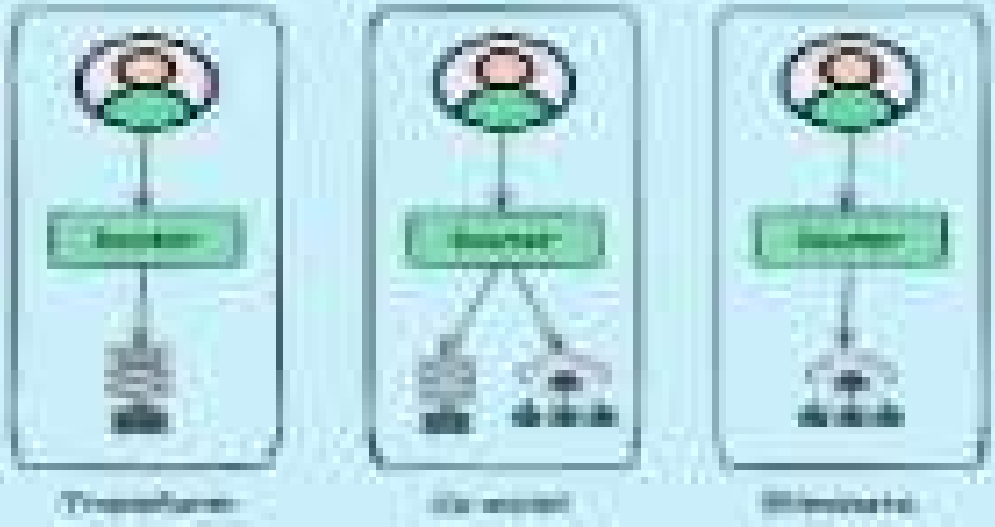


MONOLOTHIC TO MICROSERVICES

Strangler Fig pattern

Strangler Fig pattern

Strangler Fig pattern



DEPLOYMENT, PORTABILITY & MODERNITY OF MICROSERVICES

[illegible]

WHAT ARE CONTAINERS & Docker ?

What is a container ?

What is Docker ?

What is Kubernetes ?

WHAT EXACTLY HAPPENS IN CONTAINERIZATION?

BBC Bytes

[illegible]

[illegible]

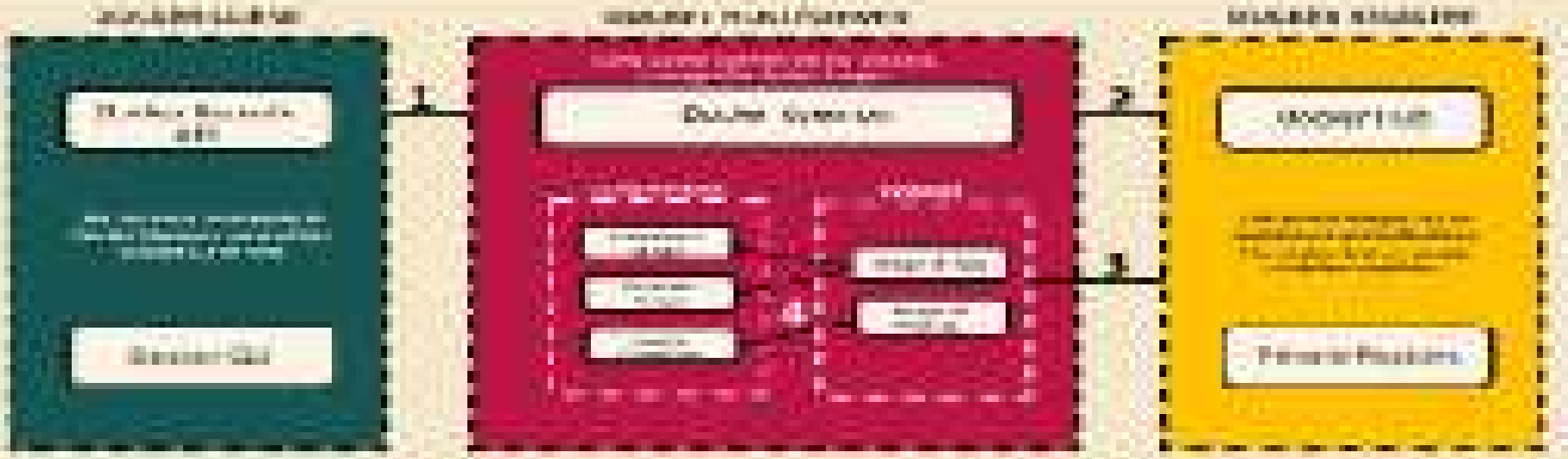
[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



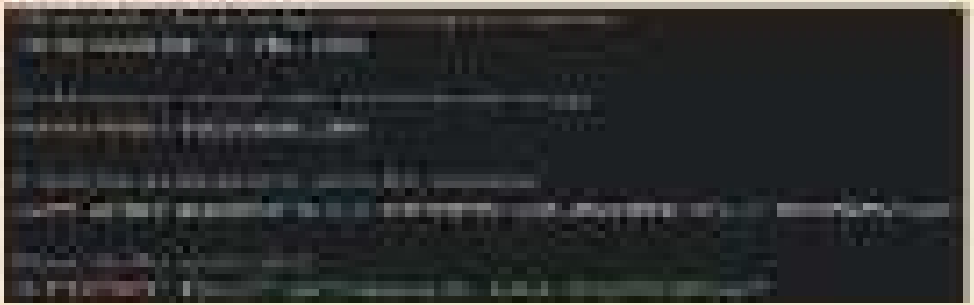
1. [illegible]
2. [illegible]
3. [illegible]
4. [illegible]

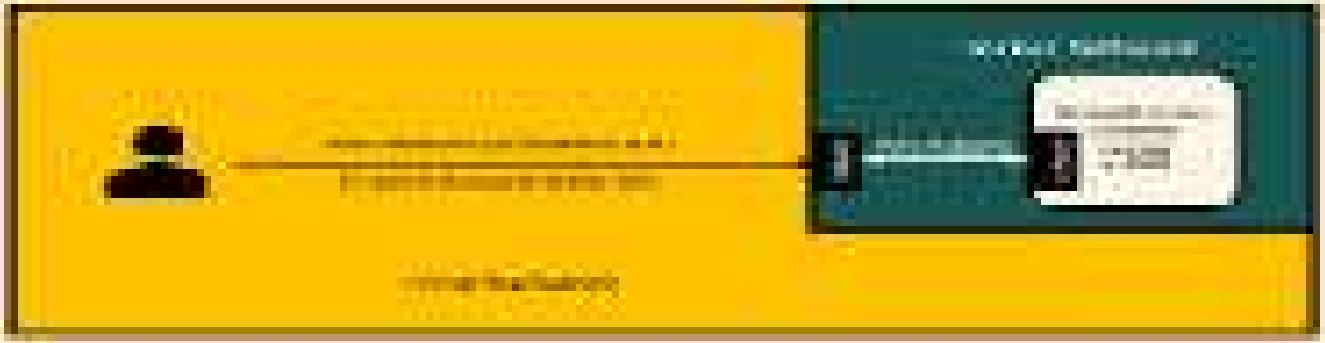
01

02

03

Example Dockerfile





Running a Spring Boot app as a container using Buildpacks

[illegible]

[illegible]

[illegible]

Execute: mvnw.cmd -Dmvn.file

```
[illegible]
```

[illegible]

Running a Spring Boot app as a container using Google Jib

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

Using

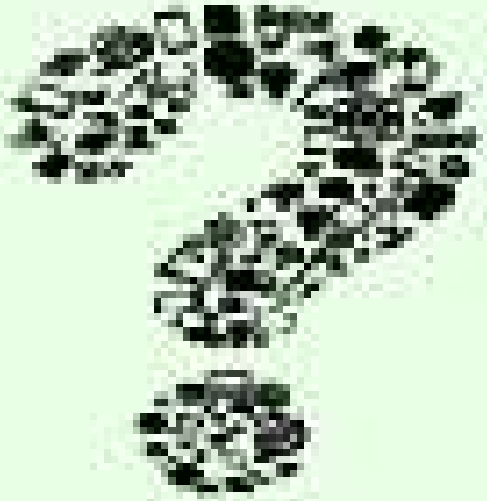
Docker Compose

to handle multiple containers

[illegible]

[illegible]

[illegible]

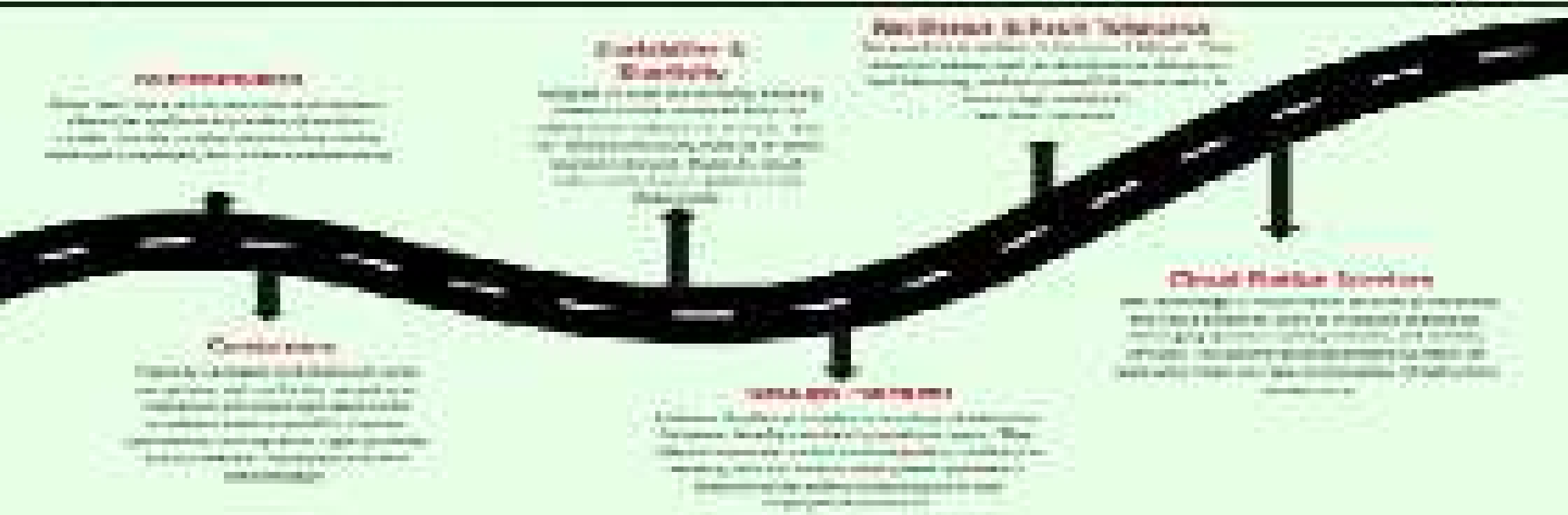


The Layman definition

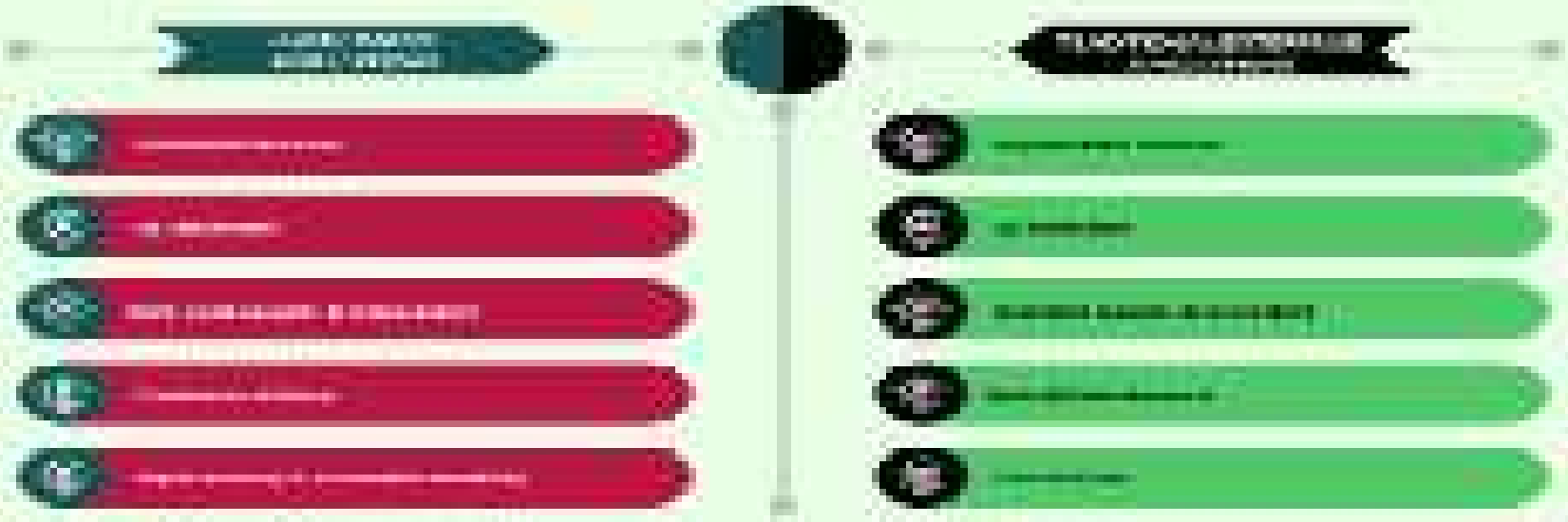
[illegible]

The Cloud Native Computing Foundation (CNCF) definition

[illegible]


Cloud Native Services

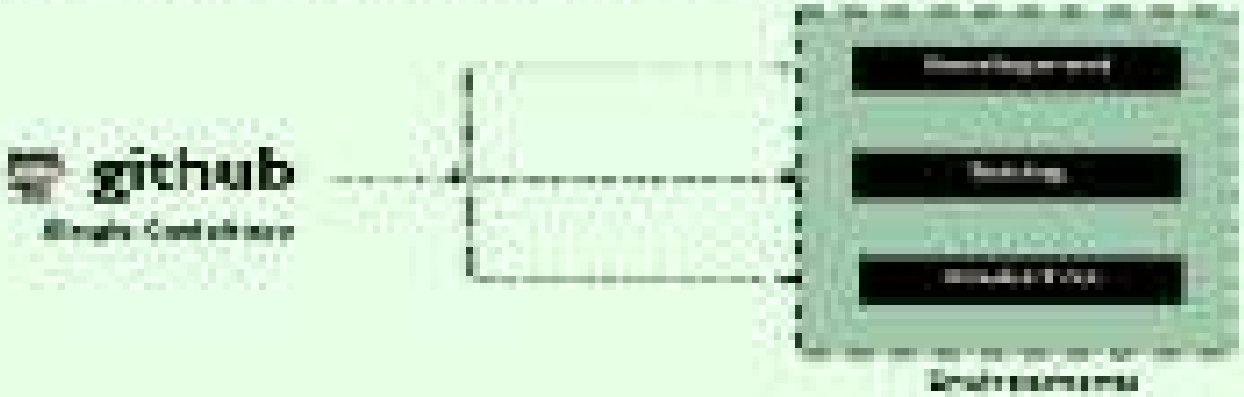
DIFFERENCE B/W CLOUD-NATIVE & TRADITIONAL APPS



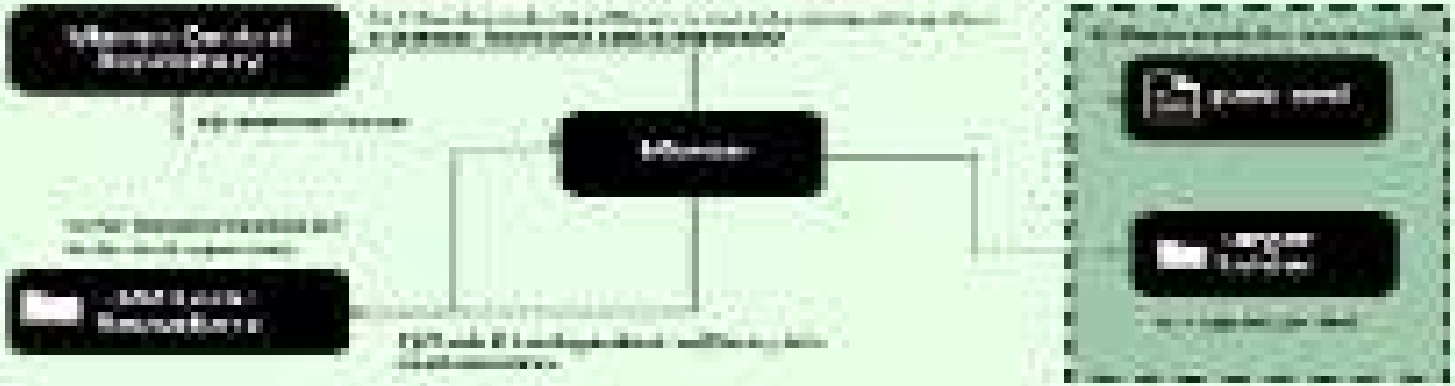
13-Factor methodology

[illegible]

[illegible]

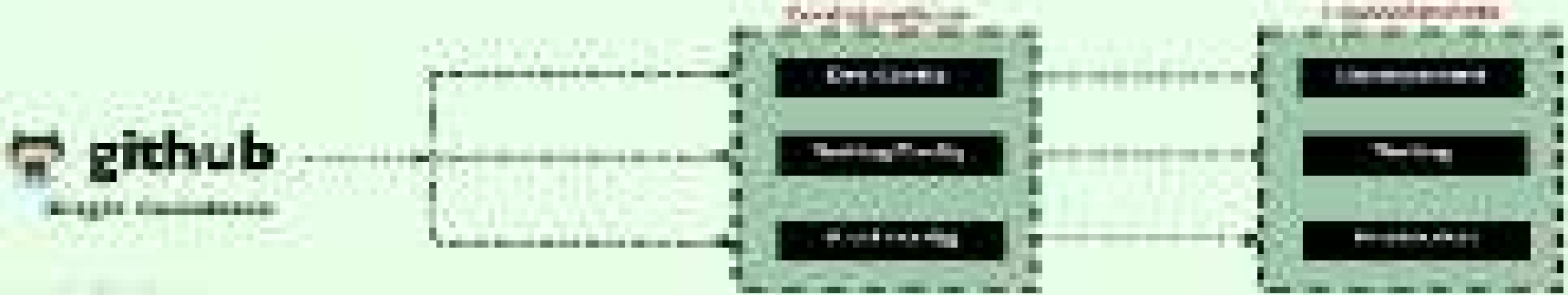

github
Single Codebase

API

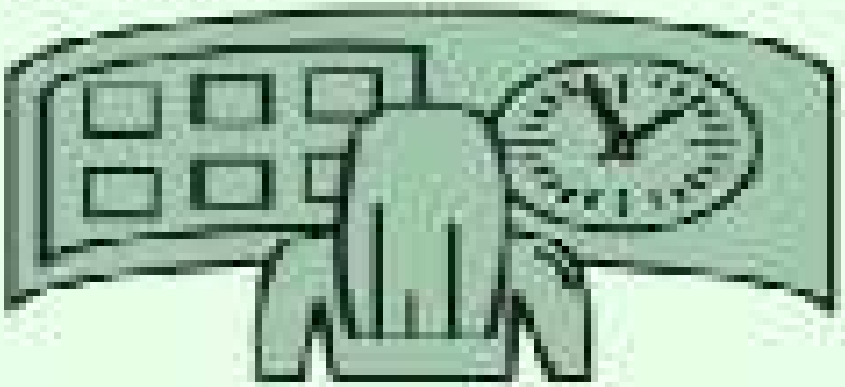


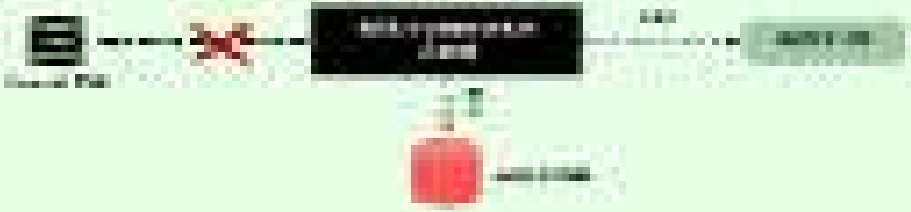
[illegible]

Configuration, credentials & code



[illegible]





[illegible]

[illegible]

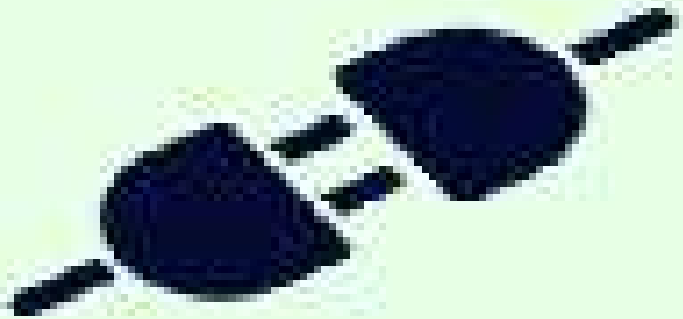
[illegible]

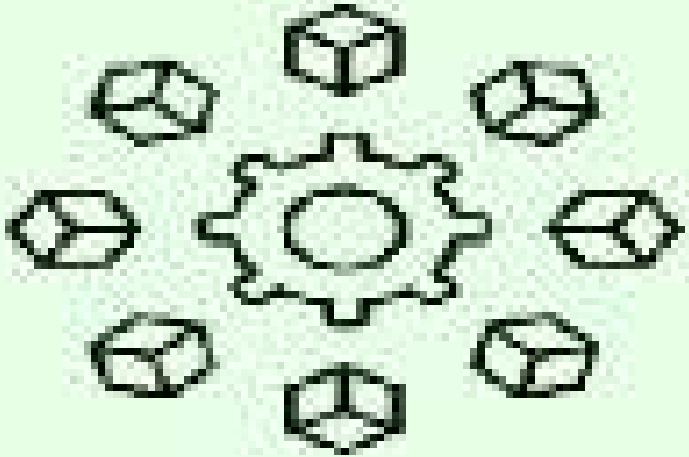
[illegible]

[illegible]

[illegible]

[illegible]



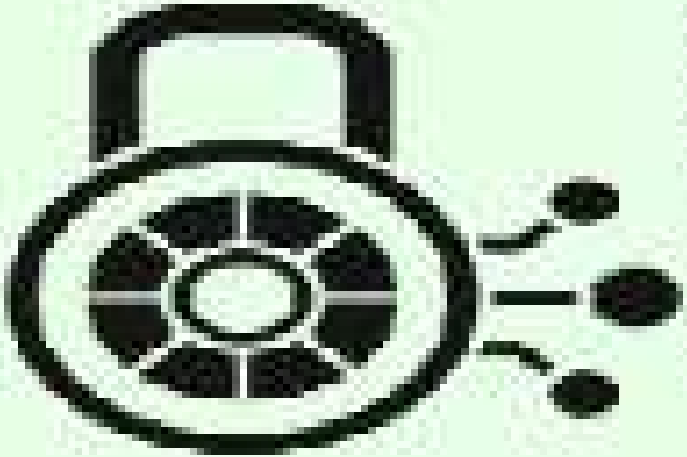


Vertical Scalability
Horizontal Scalability

[illegible]

[illegible]

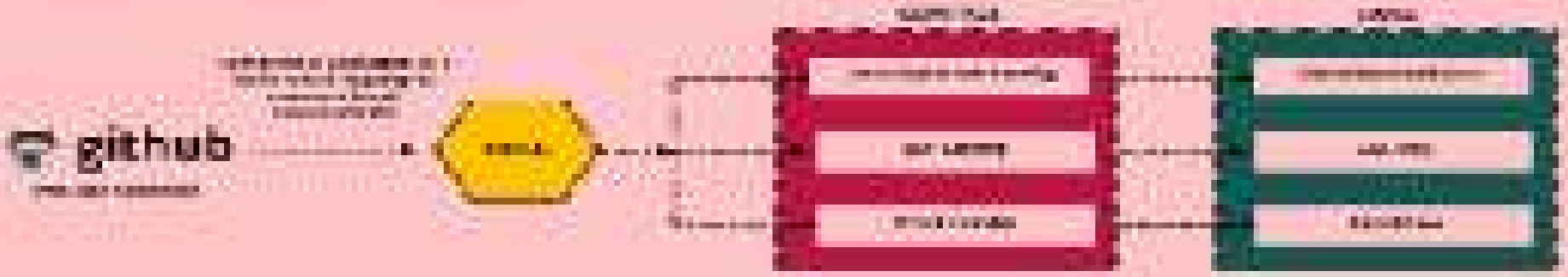
[illegible]



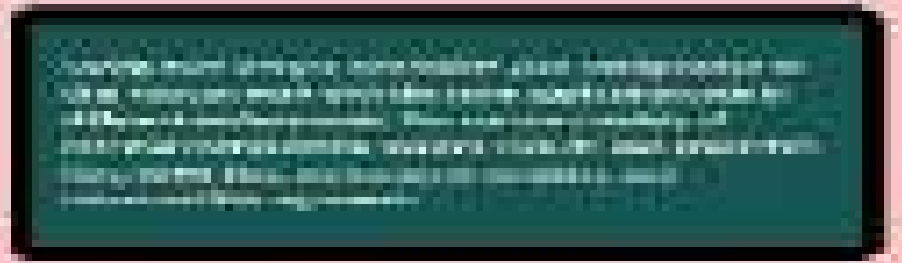
[illegible]

[illegible]

HOW CONFIGURATIONS HANDLED IN TRADITIONAL APPS & MICROSERVICES



HOW CONFIGURATIONS WORK IN SPRINGBOOT ?



HOW TO READ PROPERTIES IN SPRINGBOOT APPs

Profiles

[illegible]

[illegible]

[illegible]

How to externalize configurations using command line arguments

How to externalized configurations using JVM system properties ?

[illegible]

[illegible]

[illegible]

How to externalized configurations using environment variables ?

mike
bytes

1. [illegible]

2. [illegible]

3. [illegible]

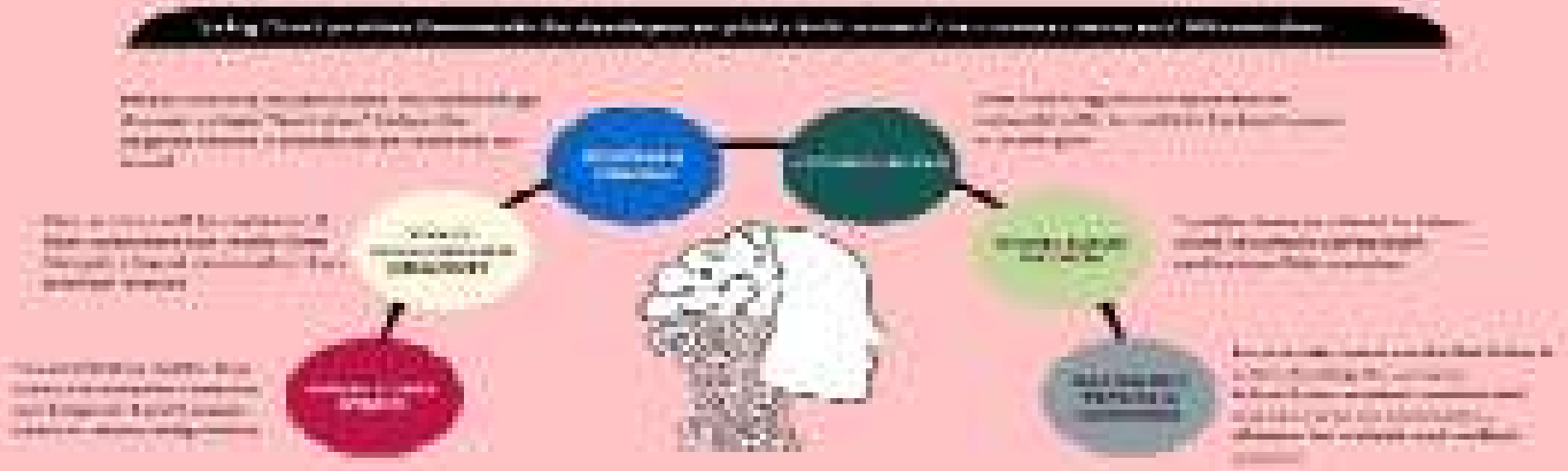
4. [illegible]

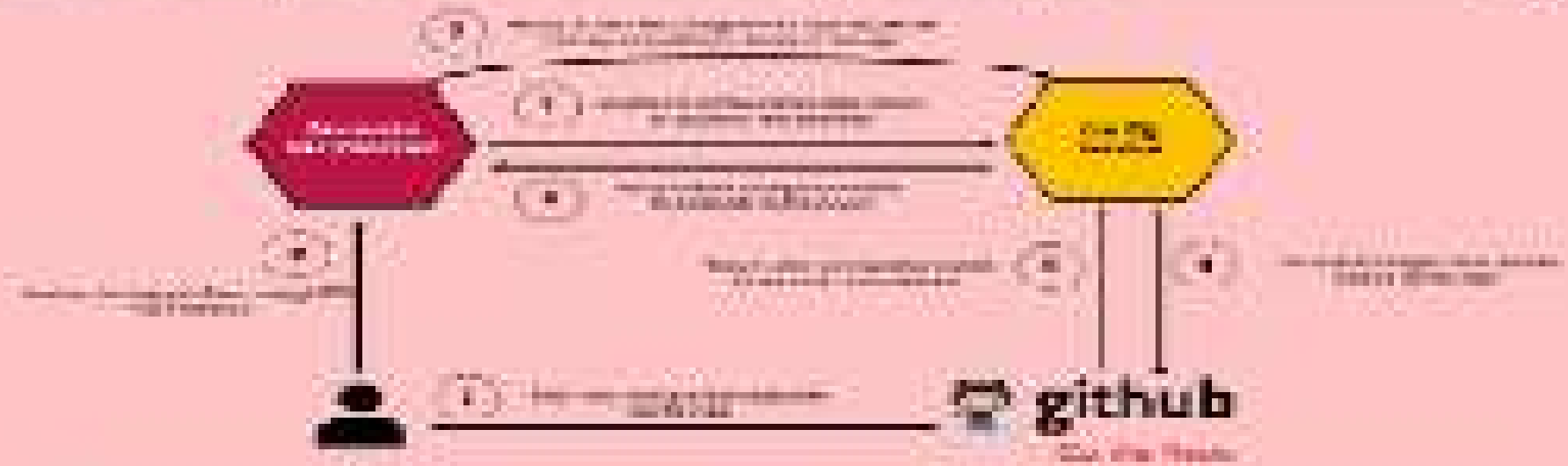
5. [illegible]

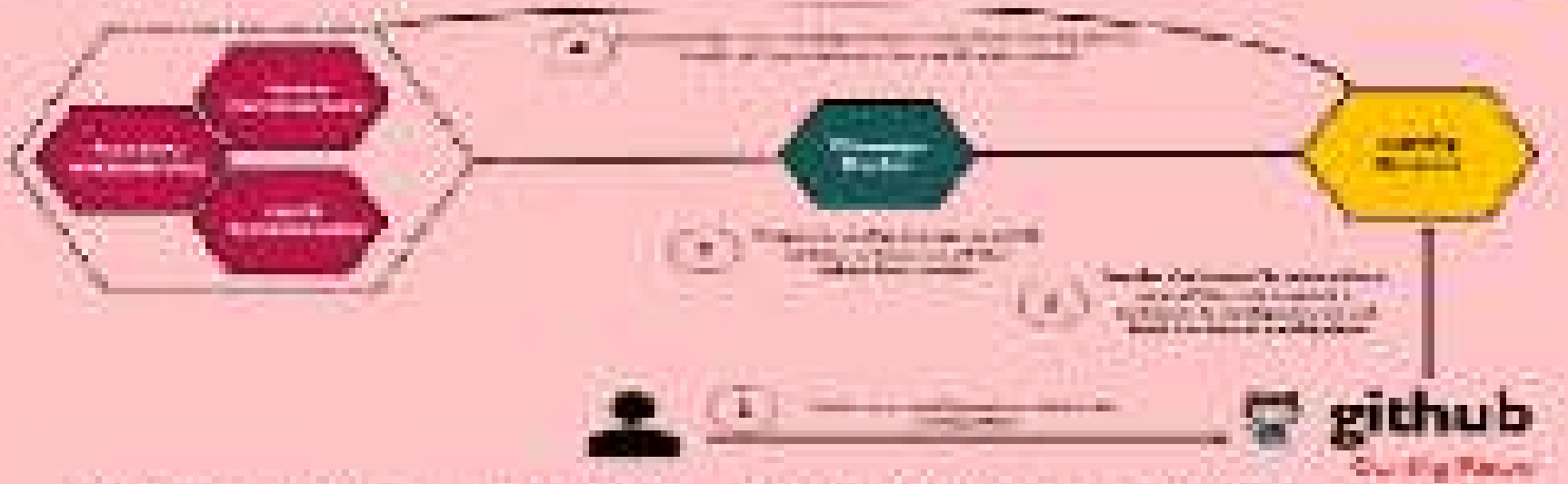
6. [illegible]

gitlab

WHAT IS SPRING CLOUD?




github


github

Refresh configurations at runtime using Spring Cloud Bus & Spring Cloud Config Monitor

Dmitry Mamchin

DEVOXX UK 2016

[illegible]

1. [illegible]
2. [illegible]

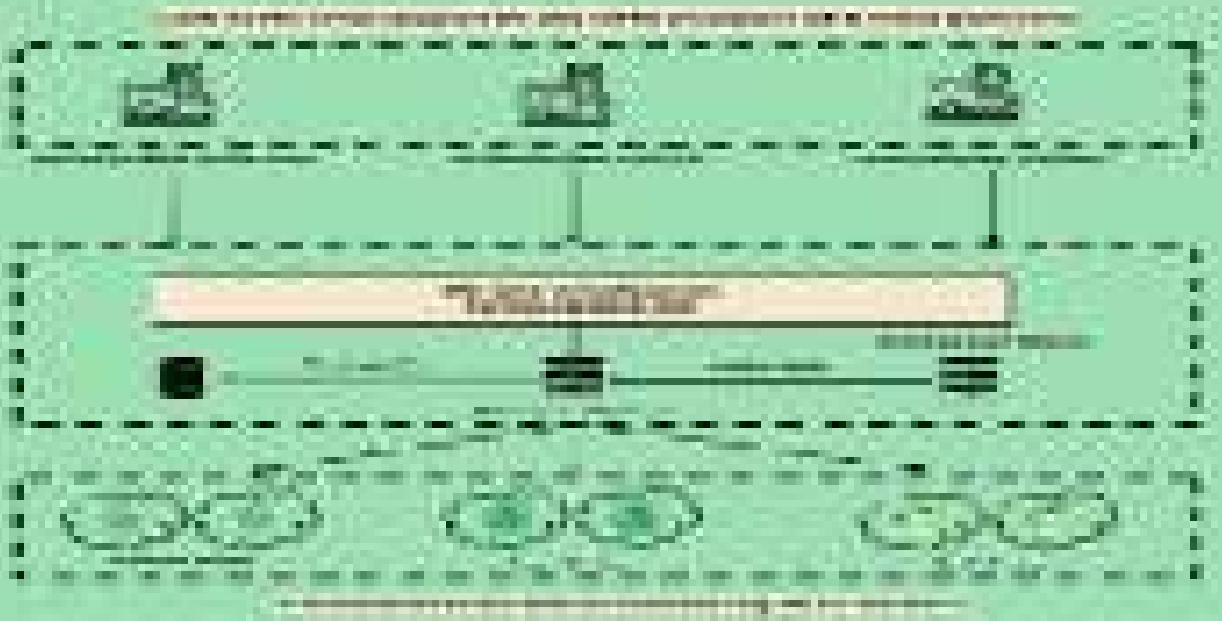
3. [illegible]
4. [illegible]
5. [illegible]
6. [illegible] to the [illegible] [illegible].

github

How service communication happens in traditional apps ?

[illegible]

How traditional Loadbalancers works ?



Limitations With Traditional Loadbalancers ?

How to solve the problem for cloud native applications

[illegible]

[illegible]

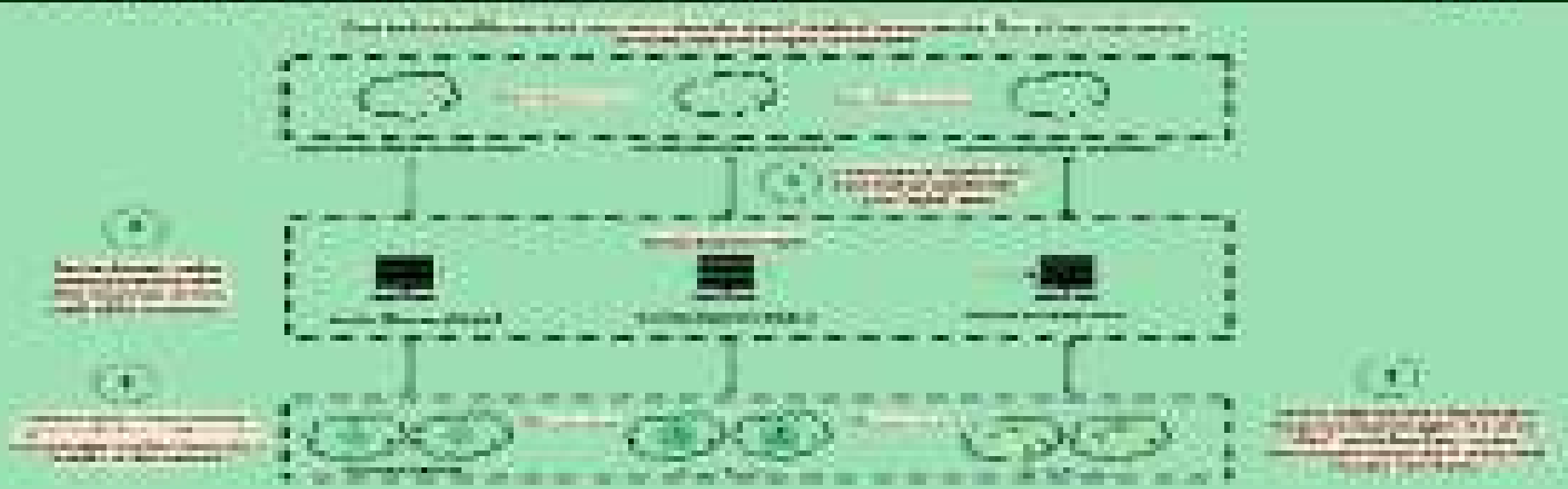
[illegible]

How to solve the problem for cloud native applications ?

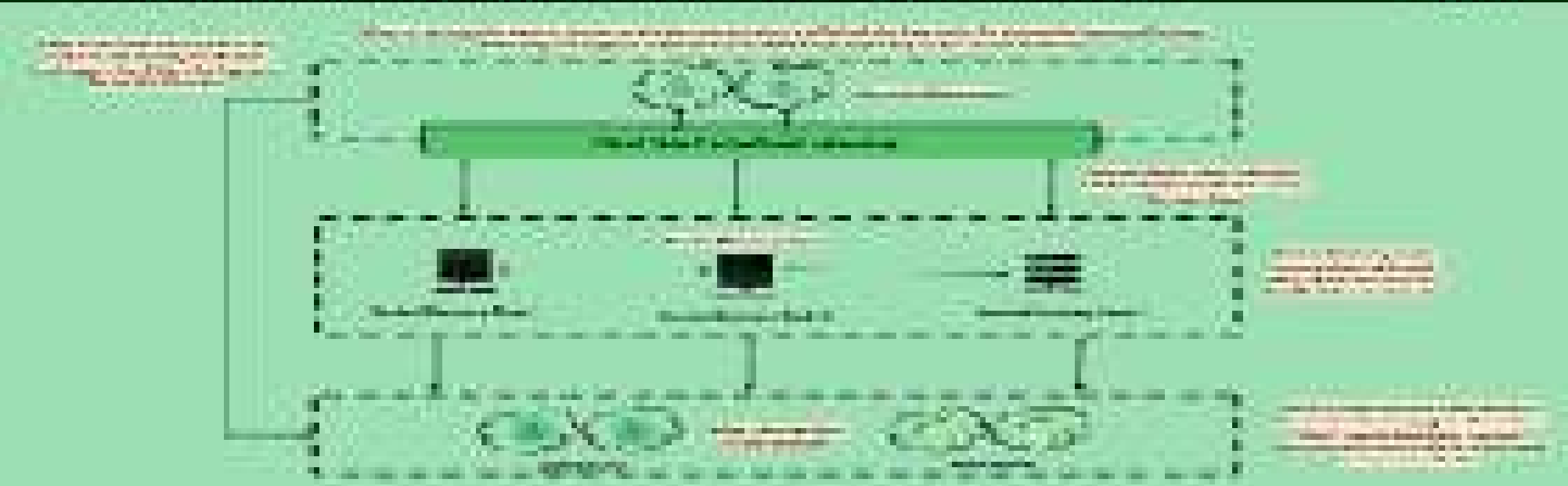
Client-side service discovery and load balancing

Client-side service discovery and load balancing

How Client-side service discovery works ?



How loadbalancing works in Client-side service discovery ?



Spring Cloud support for client-side service discovery

[illegible]

1. [illegible]

2. [illegible]

```
[illegible]
```

3. [illegible]

4. [illegible]

[illegible]

1. [illegible]

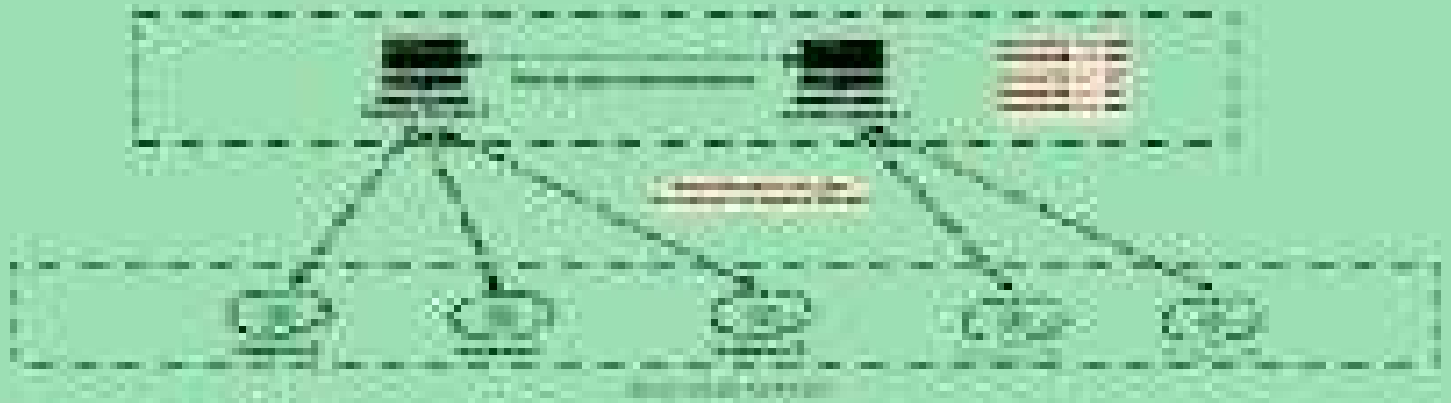
2. [illegible]

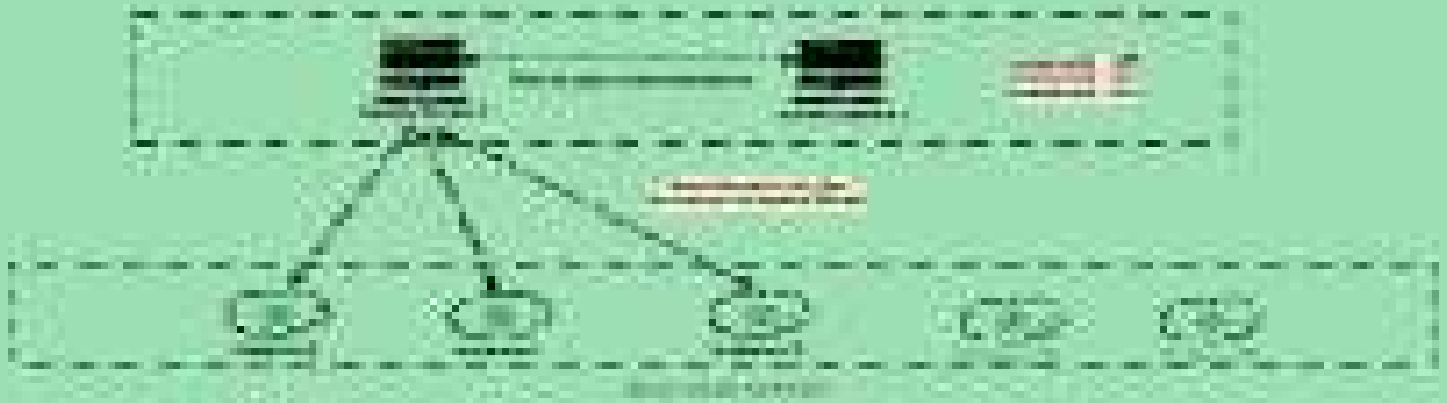
```
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```

3. [illegible]

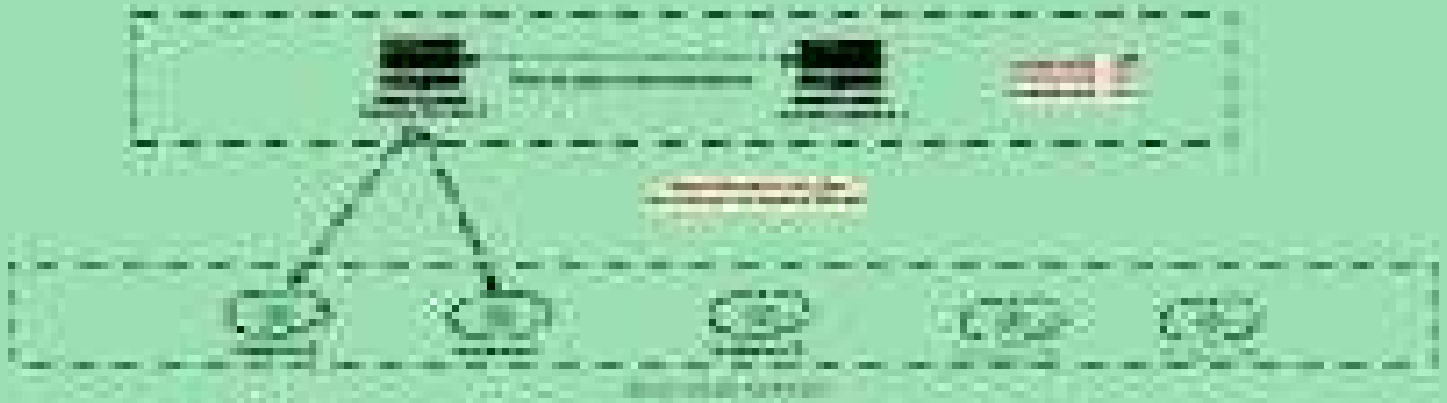
EUREKA SELF-PRESERVATION TO AVOID TRAPS IN NETWORK

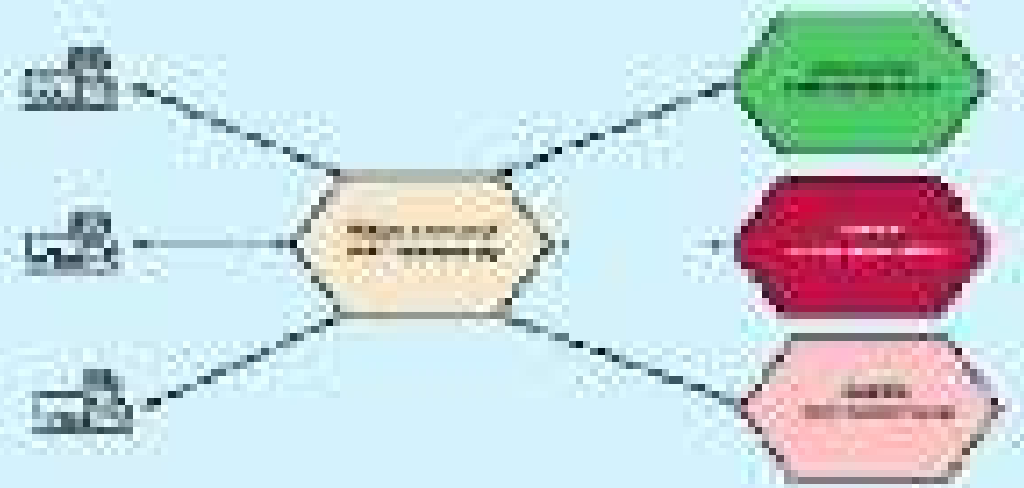
[illegible]



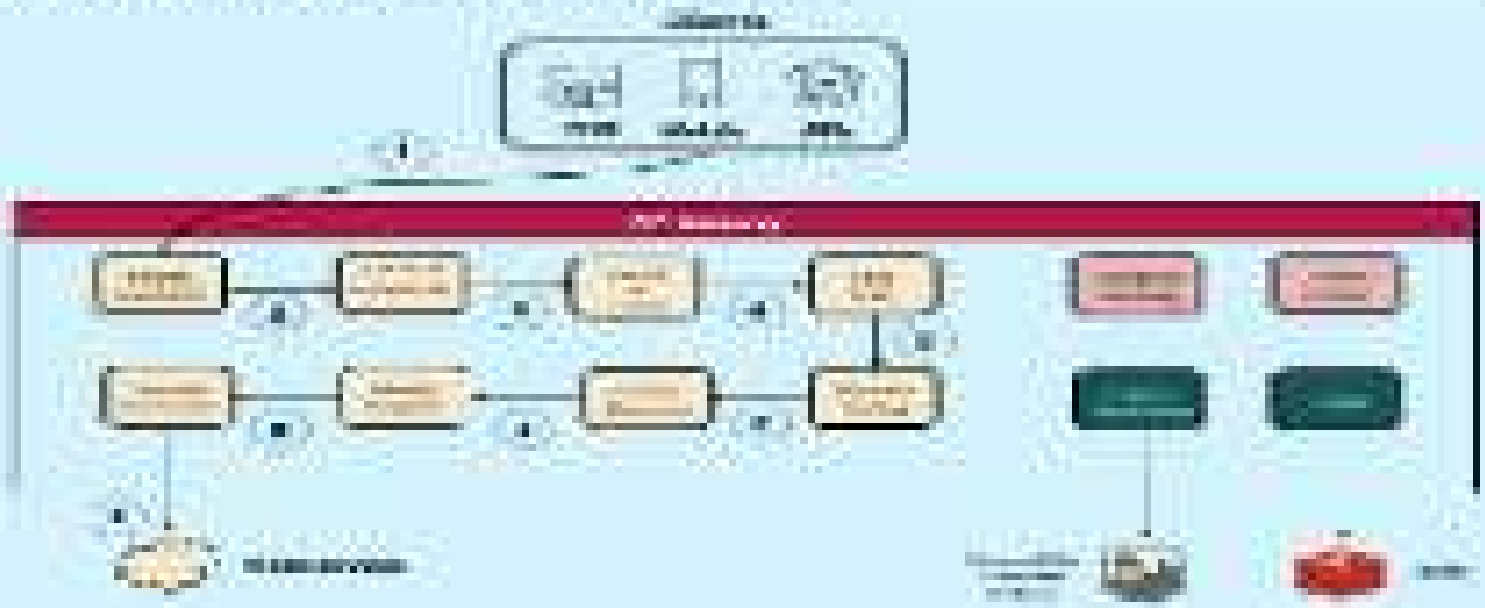


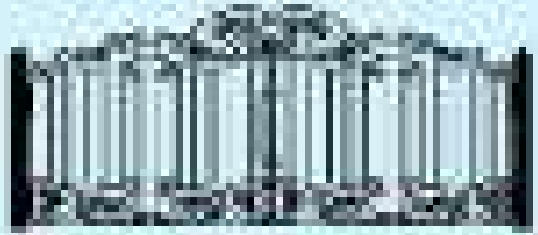
[illegible]



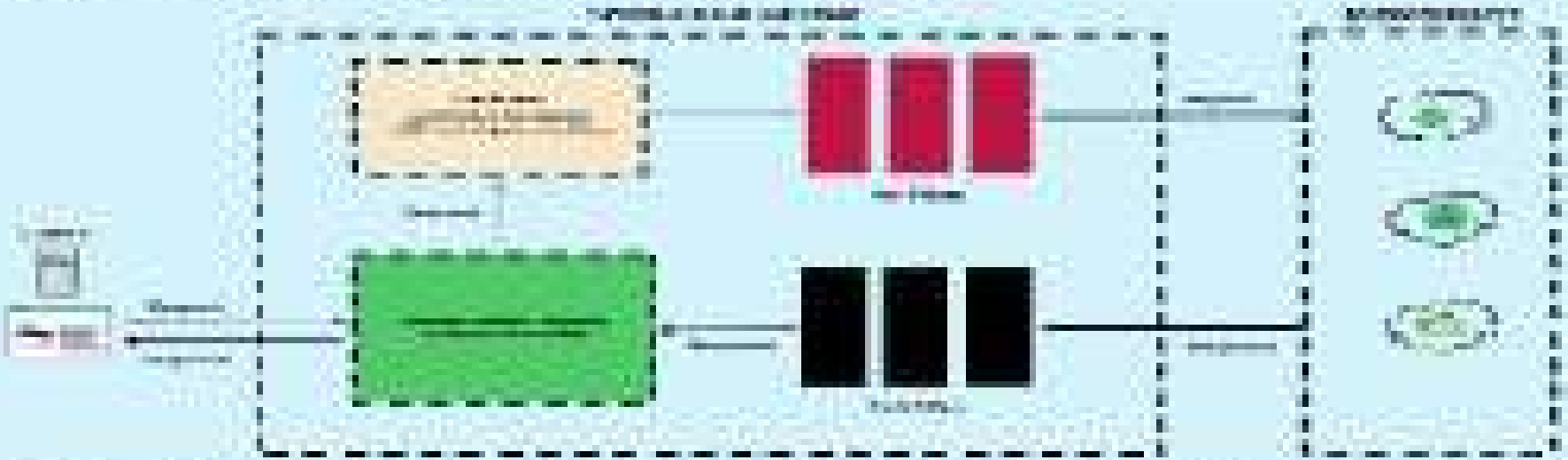


Few important tasks that API Gateway does





Spring Cloud Gateway Internal Architecture



[illegible]

1. [illegible]

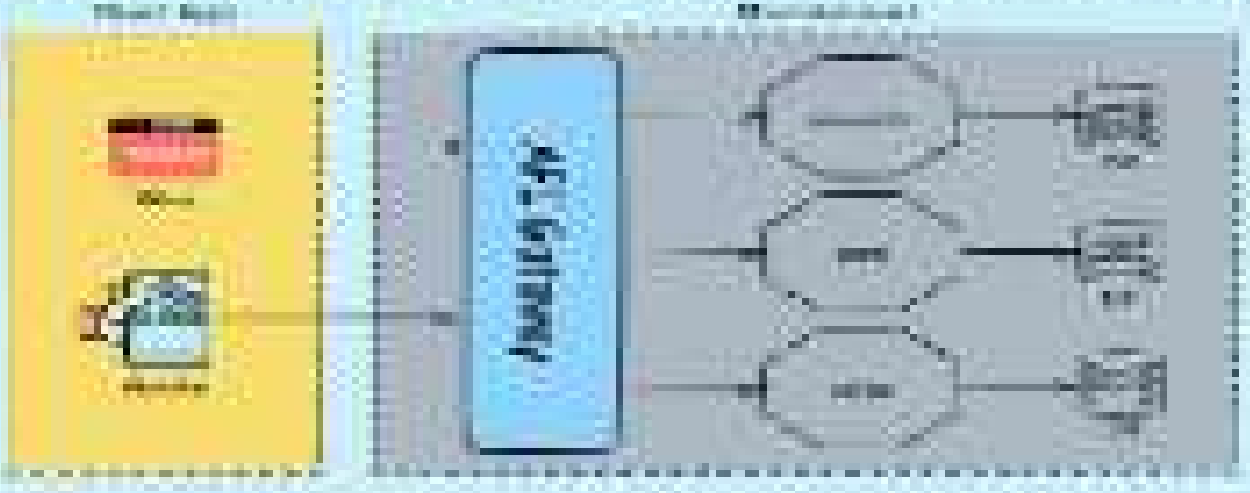
2. [illegible]

3. [illegible]

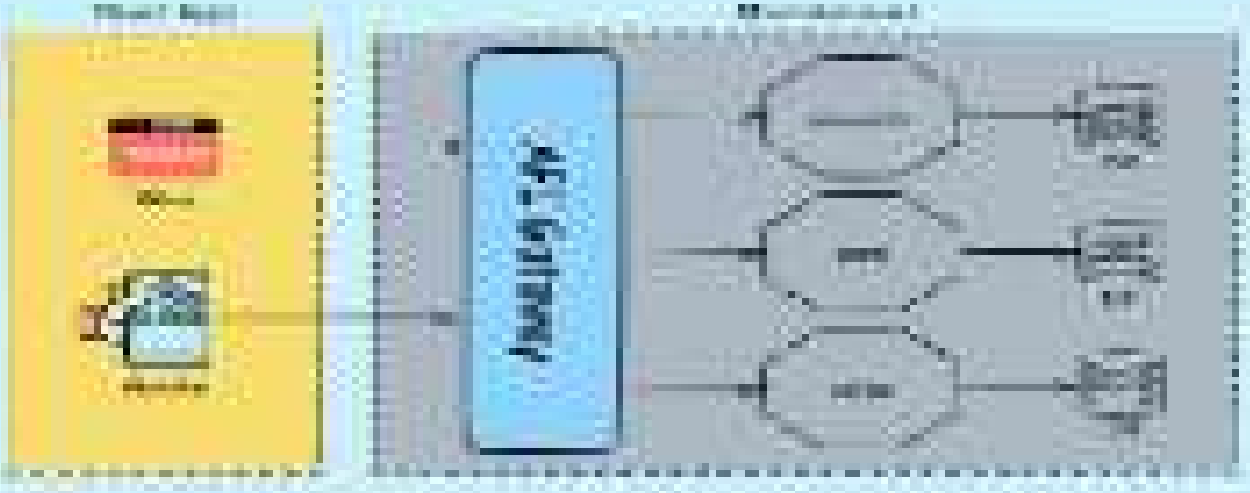
```[illegible]
```

4. [illegible]

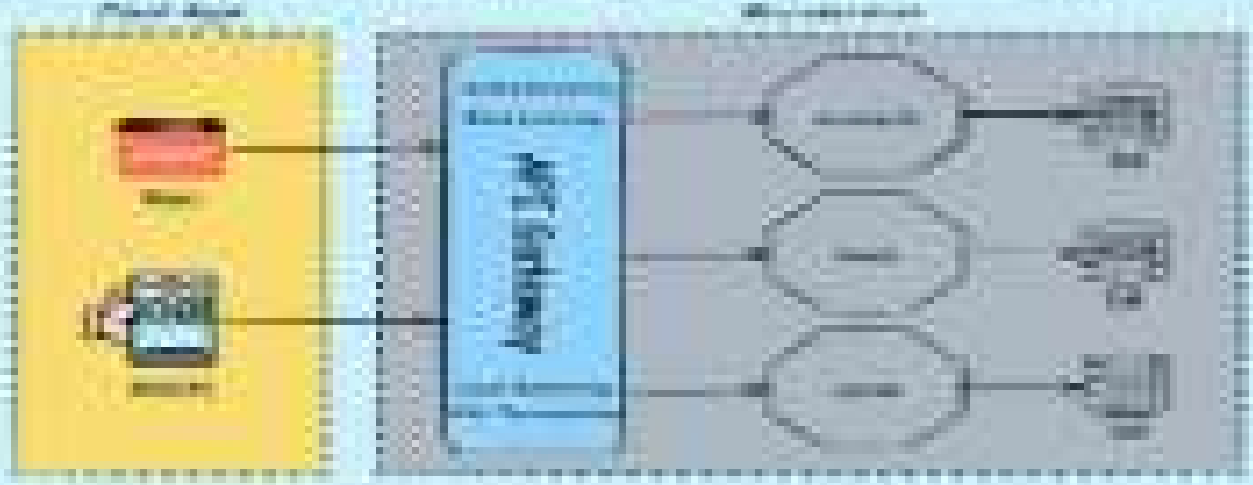
API Gateway Pattern



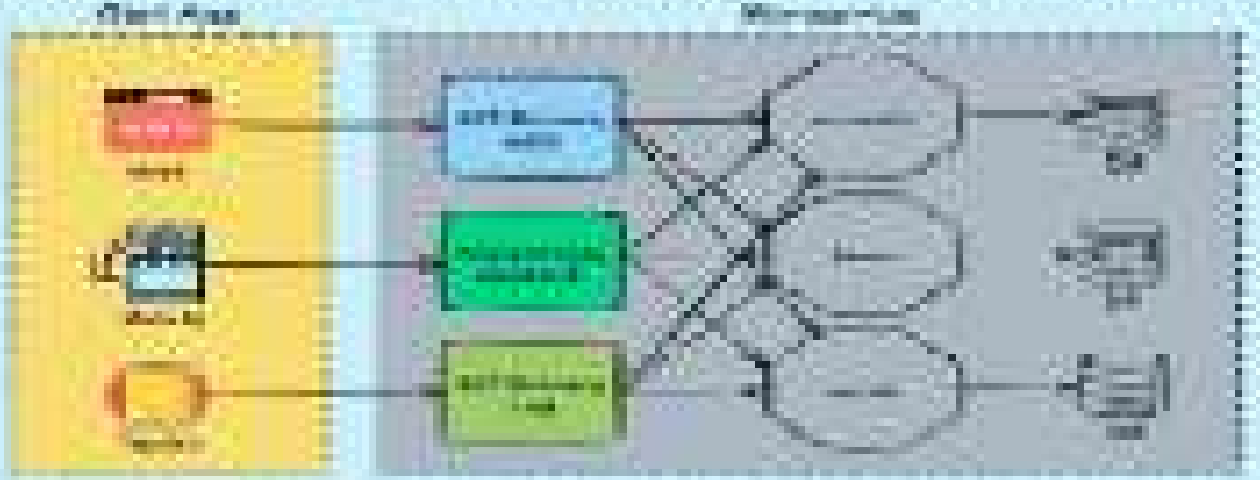
Gateway Routing pattern



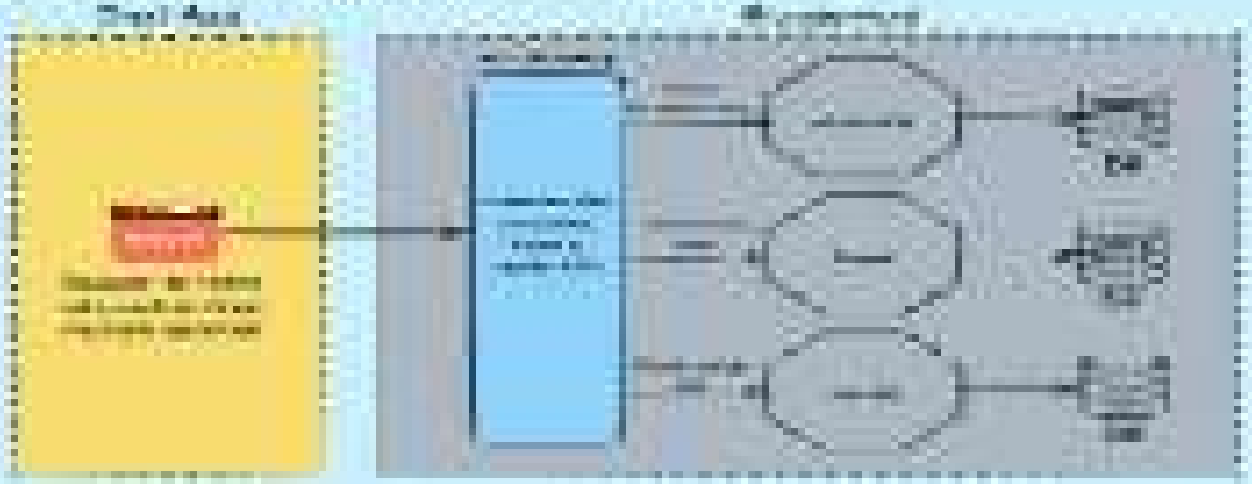
Gateway offloading Pattern

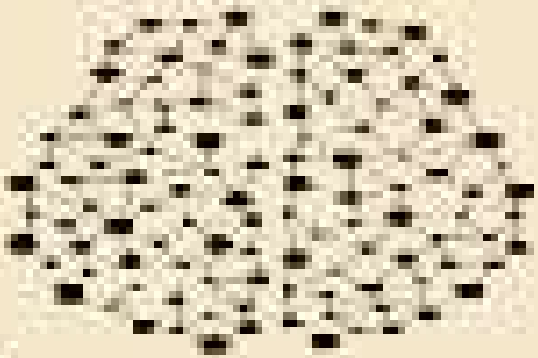


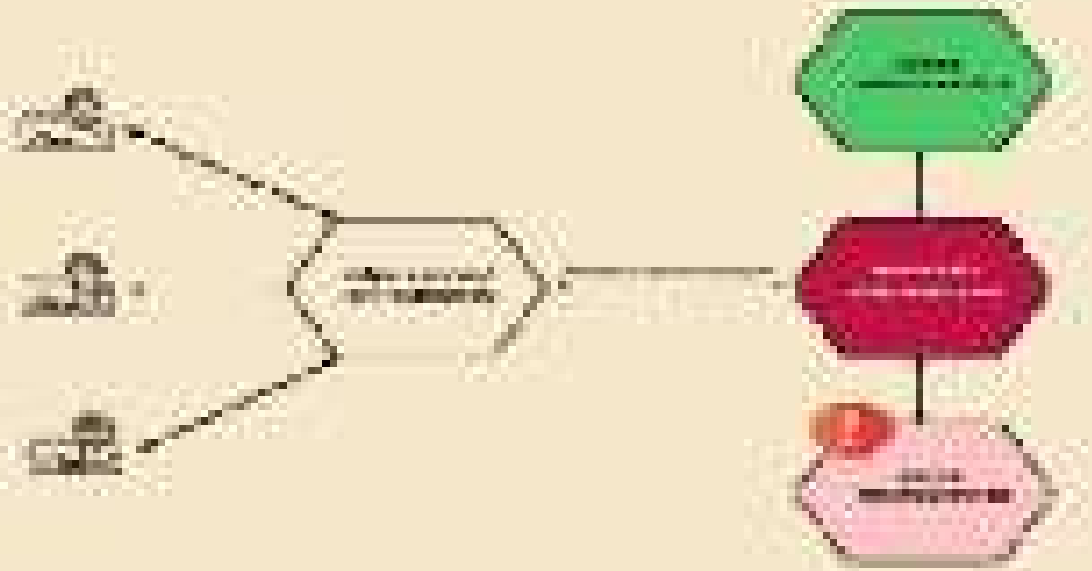
Backend for Frontend (BFF) Pattern

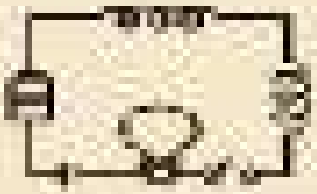


Gateway Aggregator/Composition pattern








BUSY

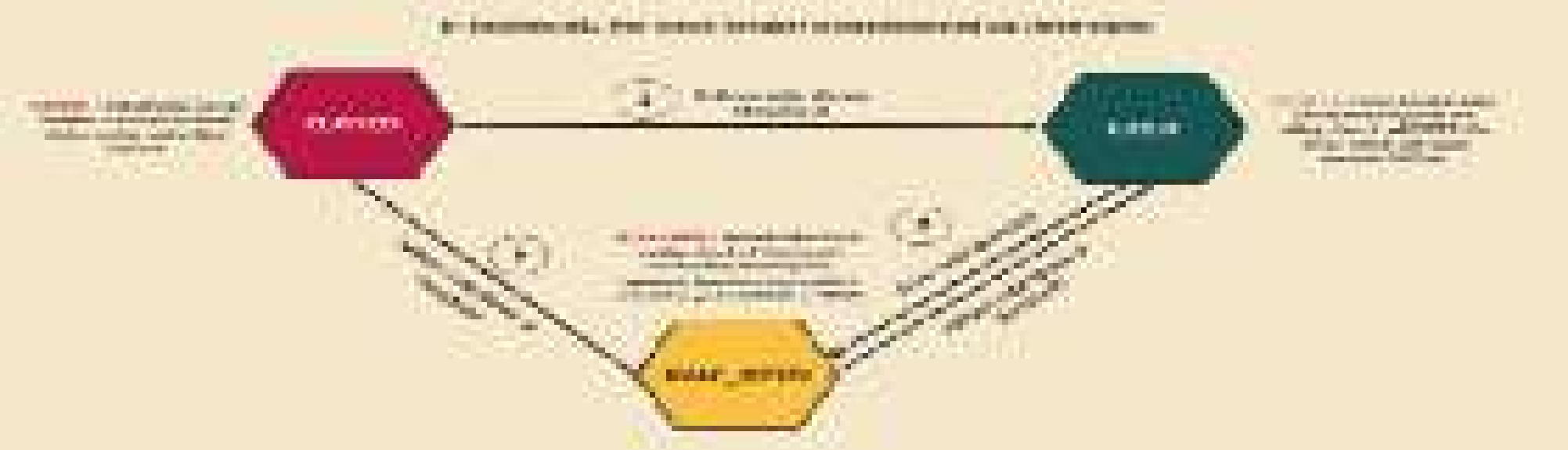
[illegible]

[illegible]

[illegible]

[illegible]

- [illegible]
- [illegible]
- [illegible]



[illegible]

1. [illegible]

2. [illegible]

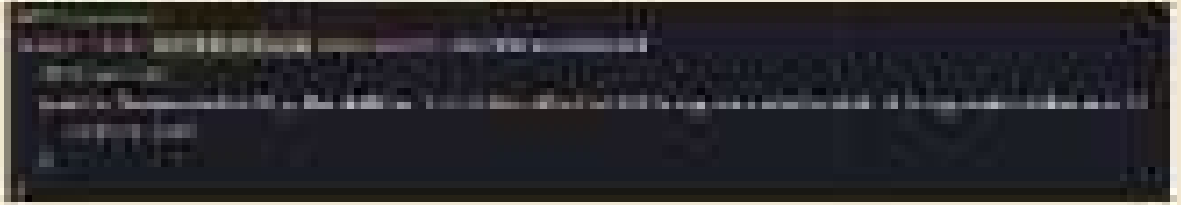
```[illegible]
```

3. [illegible]

```[illegible]
```

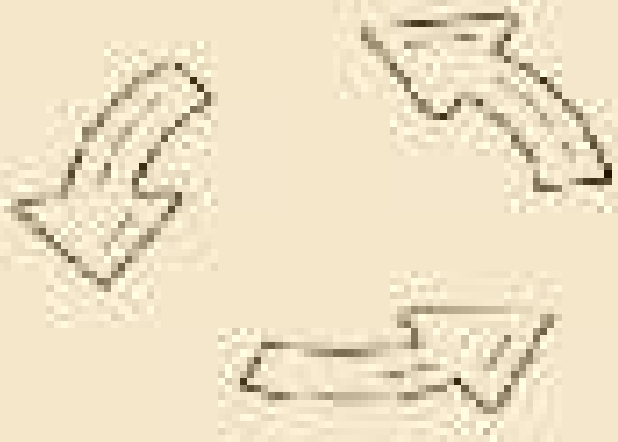
[illegible]

1. [illegible]
2. [illegible]



[illegible]

[illegible]



[illegible]

1. [illegible]

```[illegible]
```

[illegible]

a [illegible]

```
[illegible]
```

[illegible]

[illegible]

1. [illegible]
2. [illegible]

```[illegible]
```

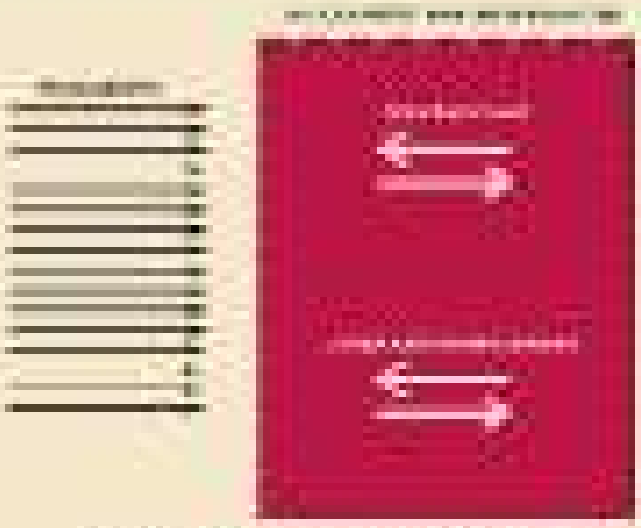
Return one line per case in Delhi order. Return nothing extra. Return Exactly the lines.

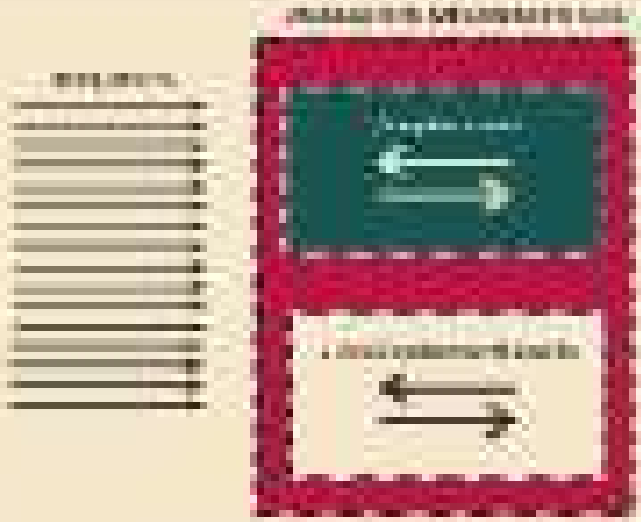
(a) [illegible]

```
[illegible]
[illegible]
[illegible]
}
[illegible]
}
```

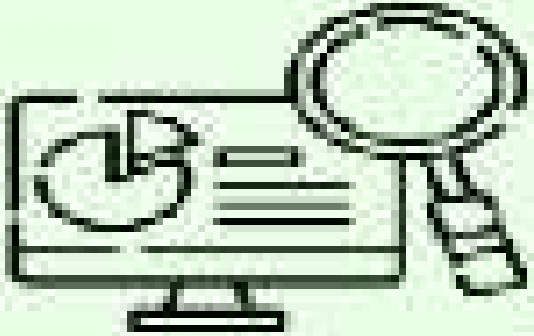
(b) [illegible]

```
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```





WHAT IS OBSERVABILITY ?



WHAT IS NEUMETCRING ?

- [illegible]
- [illegible]
- [illegible]

[illegible]

Observability vs. Monitoring

Criterion	Monitoring	Observability
Complexity	[illegible]	[illegible]
Data	[illegible]	[illegible]
Costs	[illegible]	[illegible]
Approach	[illegible]	[illegible]

In a nutshell, monitoring is about collecting data and observability is about understanding data.

Monitor is reactive to problems while observability is having [illegible] real time.

[illegible]

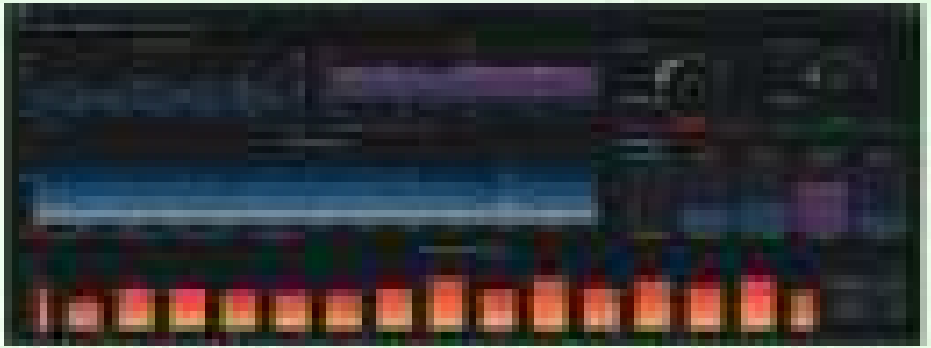
[illegible]

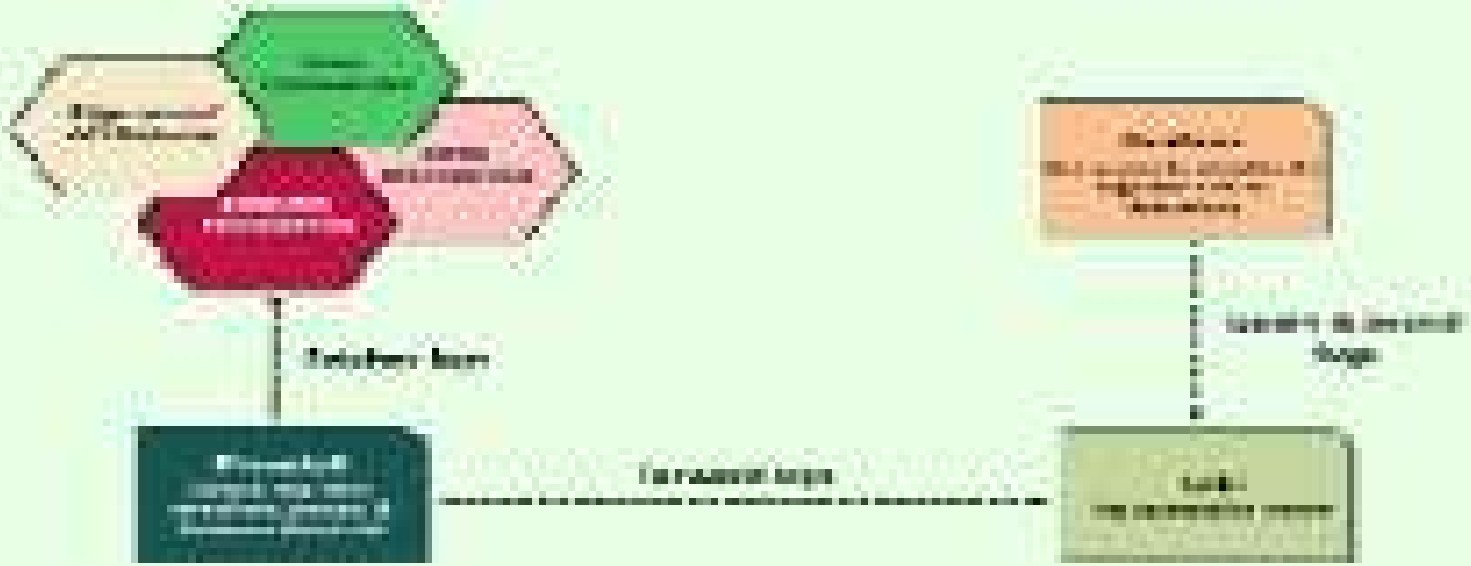
[illegible]

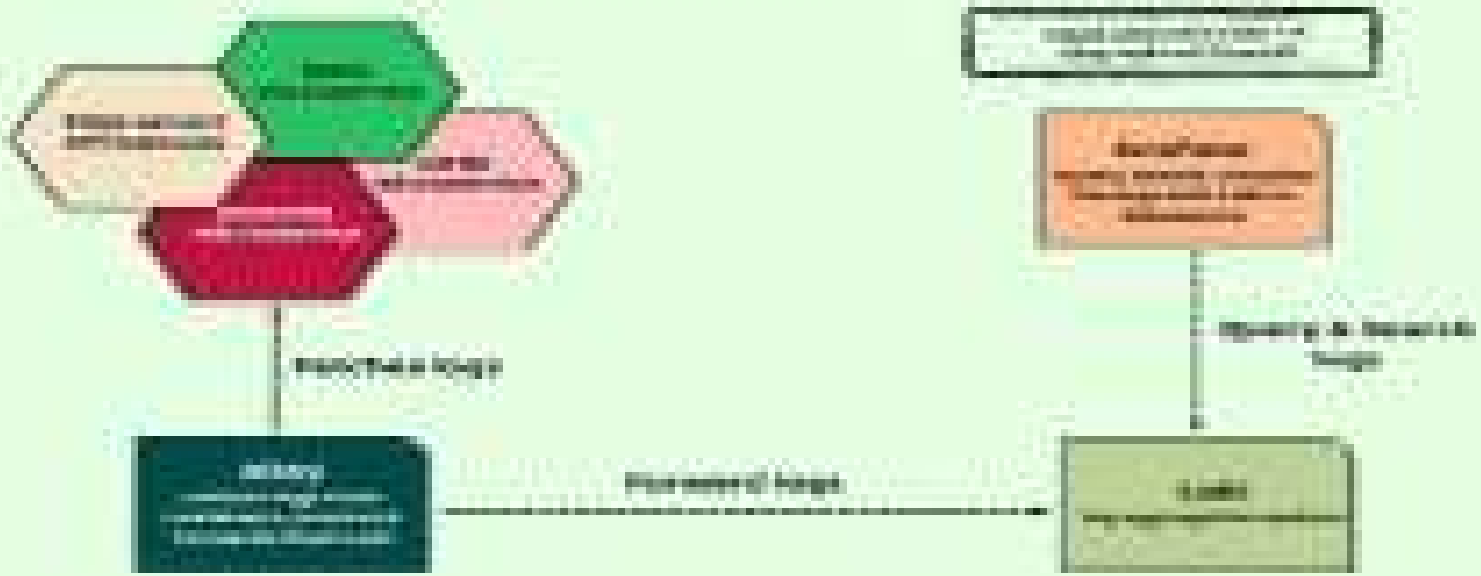
[illegible]

[illegible]

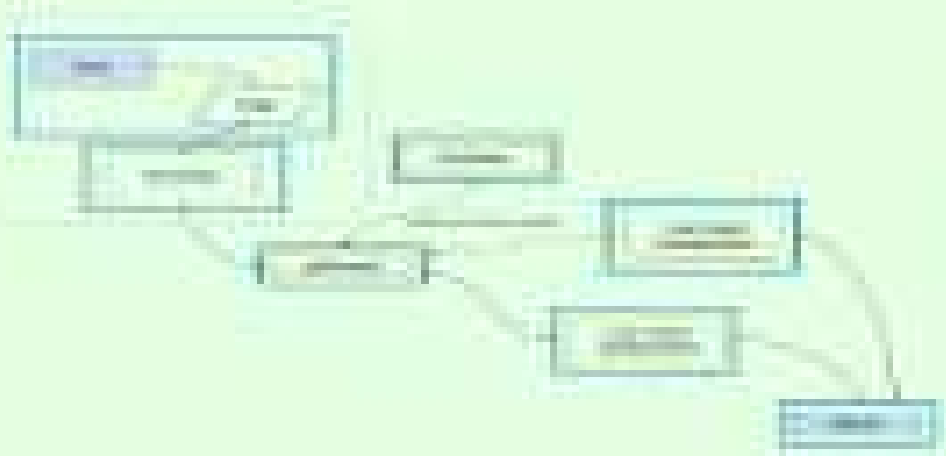
Managing logs with Grafana, Loki & Promtail





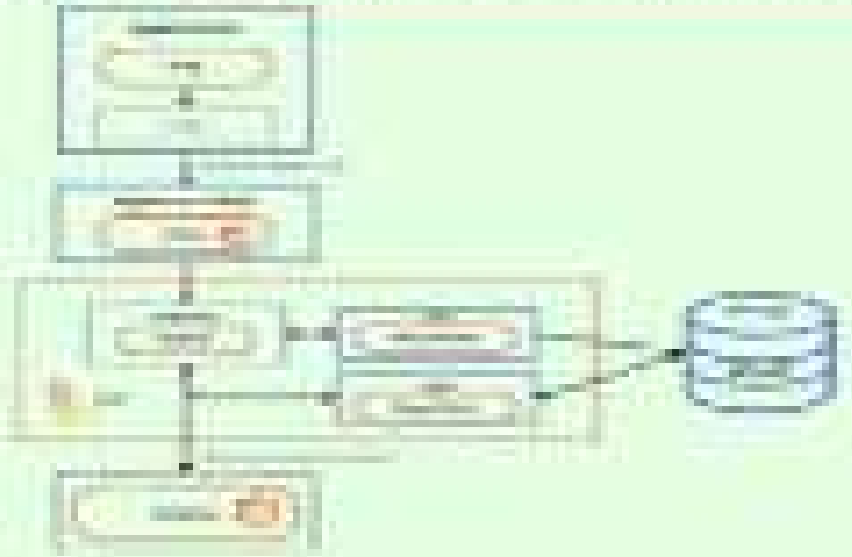


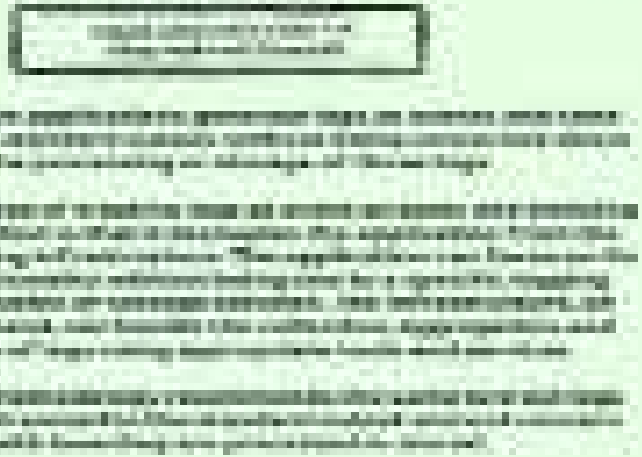
Sample demo of logging using Grafana, Loki & promotail



[illegible]

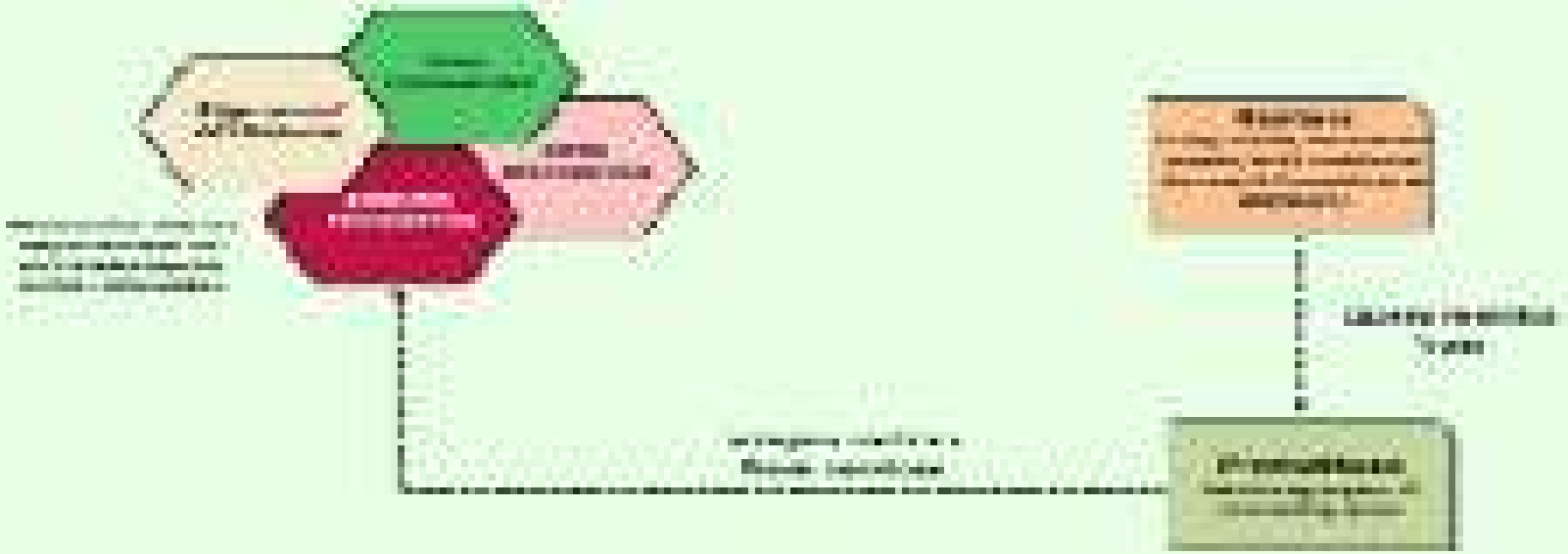
Sample demo of logging using Grafana, Loki & Alloy

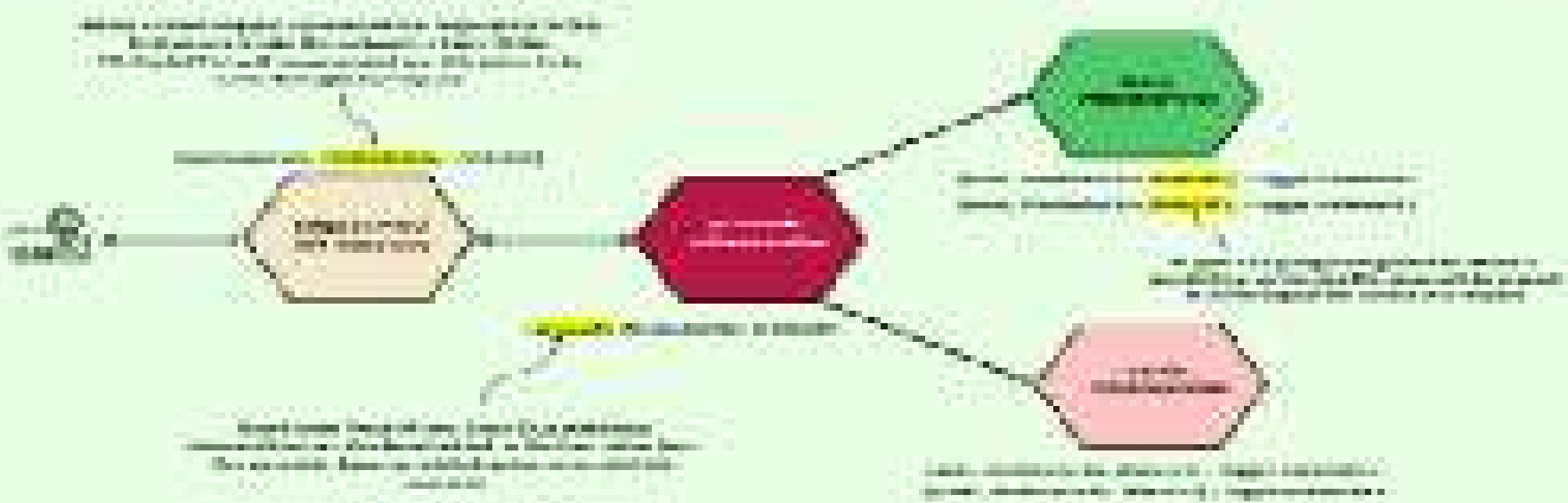




Metrics & monitoring with Spring Boot Actuator, Micrometer, Prometheus & Grafana

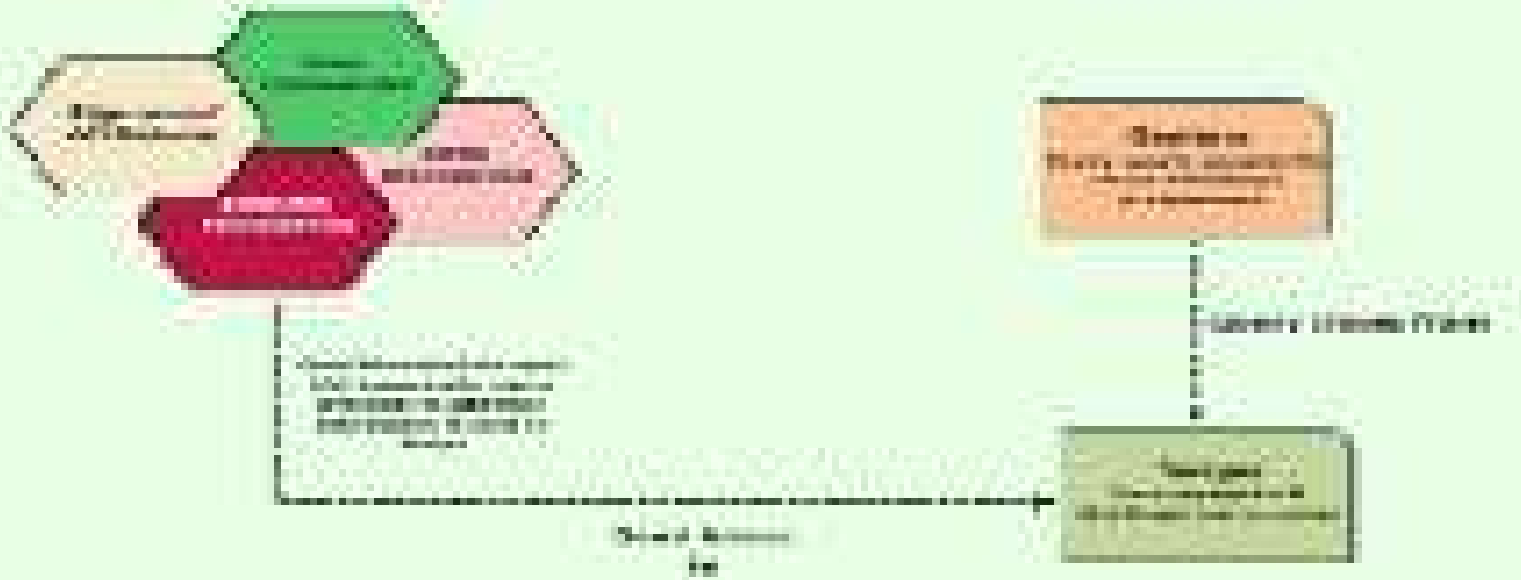
Metrics & monitoring with Spring Boot Actuator, Micrometer, Prometheus & Grafana



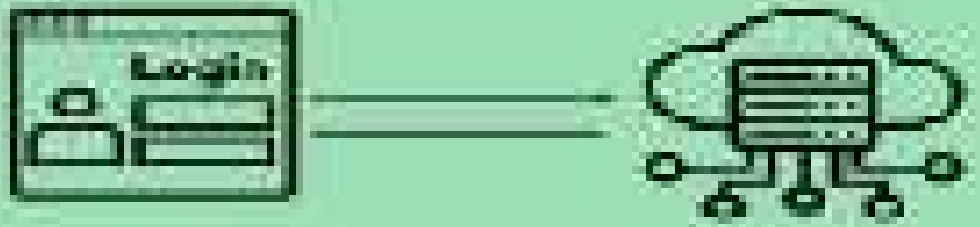


Distributed tracing with OpenTelemetry, Tempo & Grafana

Distributed tracing with OpenTelemetry, Tempo & Grafana

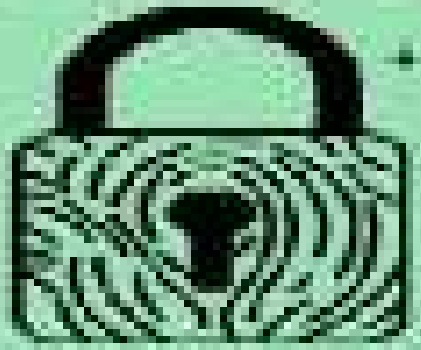


PROBLEM THAT OAUTH2 SOLVES



PROBLEM THAT OAUTH2 SOLVES

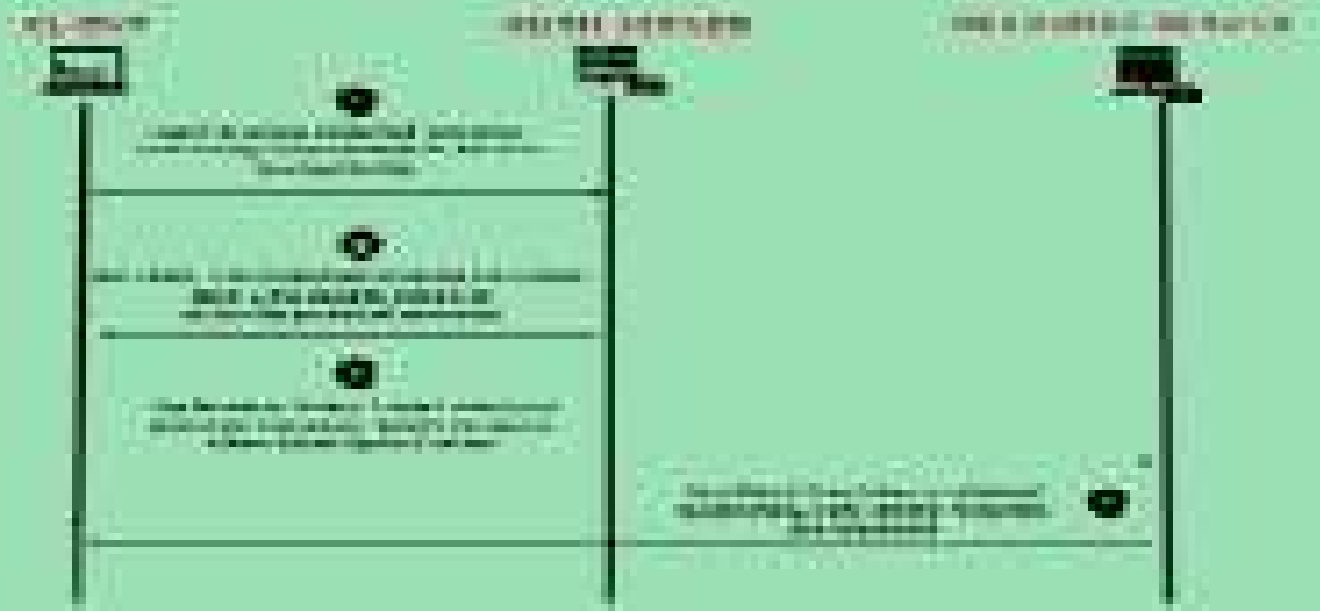
INTRODUCTION TO OMITH2



WHAT IS OPENID CONNECT & WHY IT IS IMPORTANT ?

WHAT IS OPENED CONNECT & WHY IT IS IMPORTANT ?

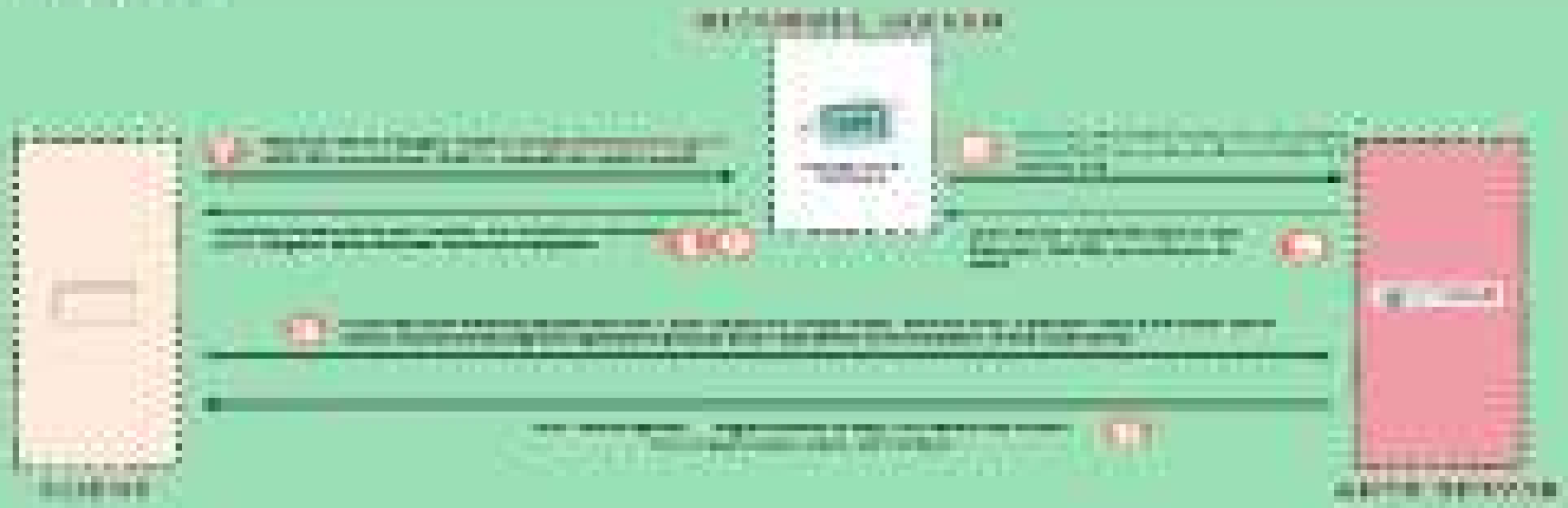
CLIENT CREDENTIALS GRANT TYPE FLOW IN OAUTH2



CLIENT CREDENTIALS GRANT TYPE FLOW IN OAUTH2

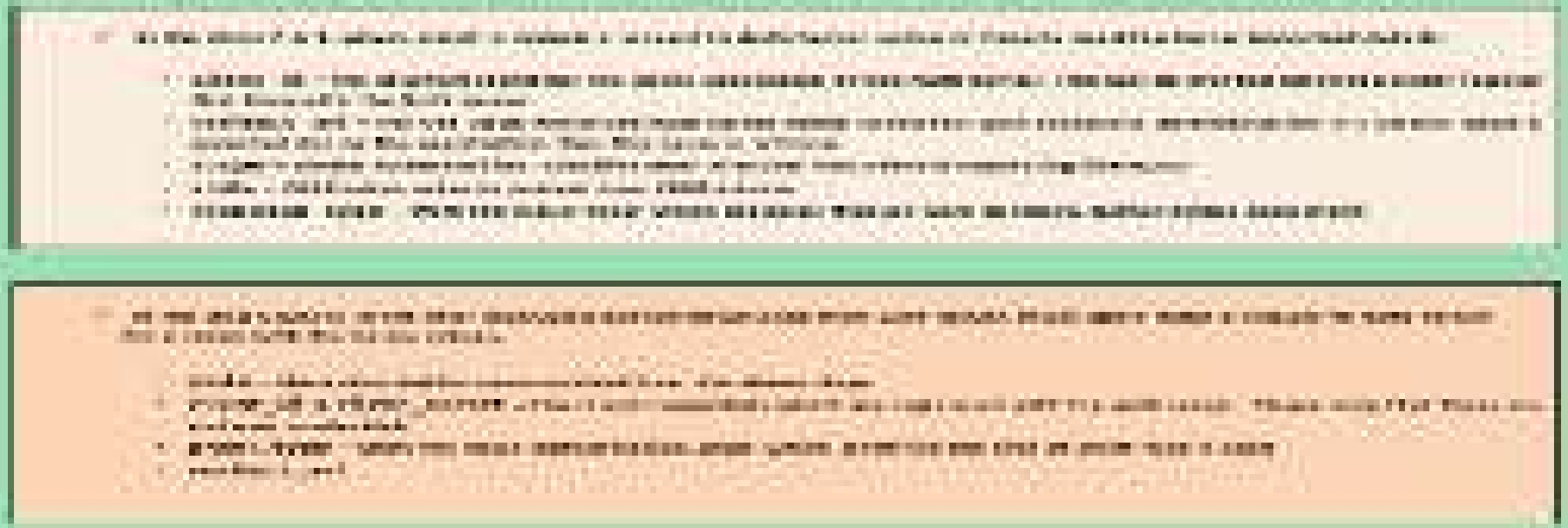
- In the case 1, where client is making a request to Auth Server endpoint, have to send the below mandatory details.
  - client_id: a unique identifier issued to the client to identify itself to the authorization server
  - client_secret: a secret to authenticate identifies type of access that client is requesting (for example, read or write)
  - grant_type: Will be the value 'client_credentials' which indicates that the client, or resource owner is requesting a grant type
- This is the most simple grant type in Oauth2 flow (see RFC-6749).
- The client is authenticated (because it is trusted as well as confidential), which is the reason behind a client-specific identifier and client secret. Client sending access token.

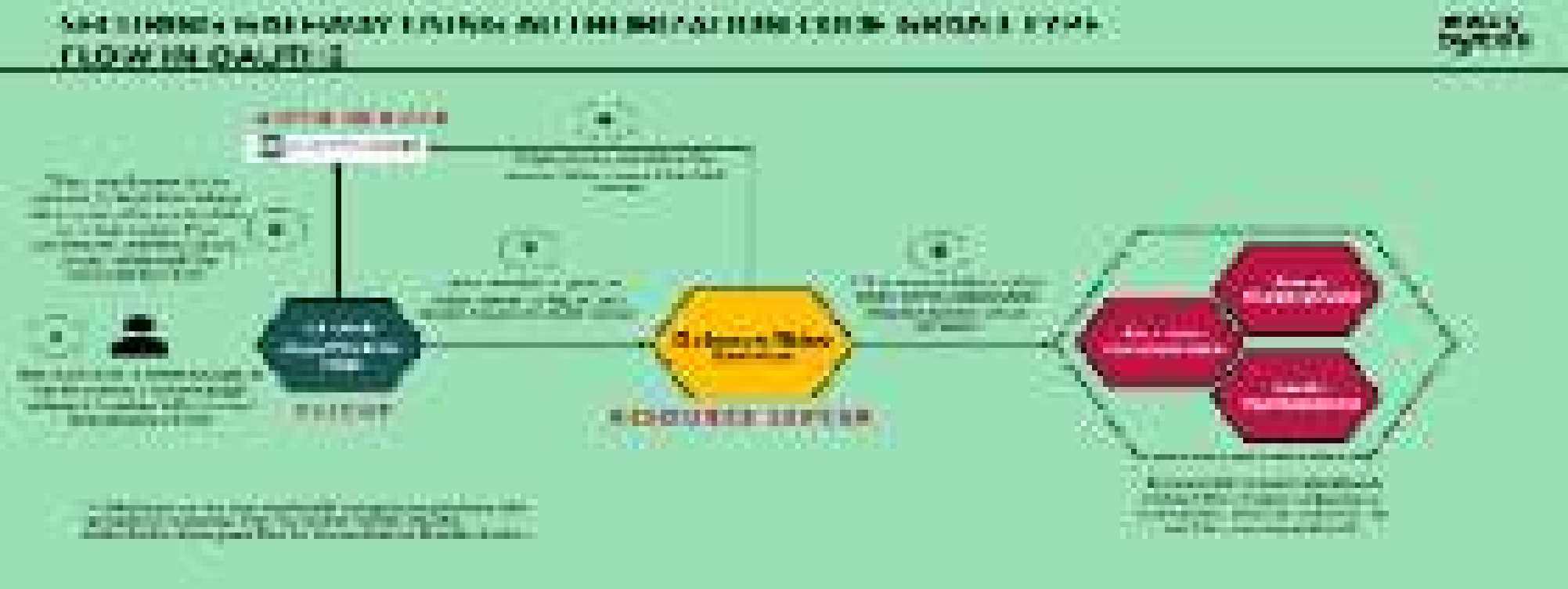
[illegible] IN GAUTENG

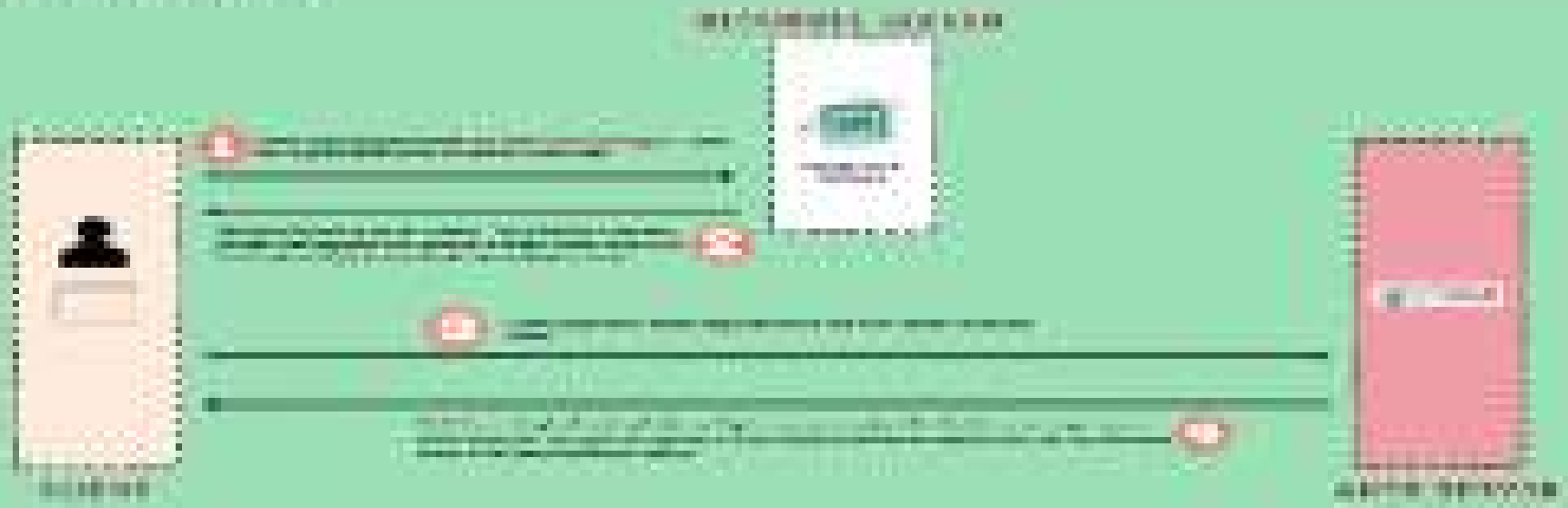


AUTHORIZATION CODE GRANT TYPE FLOW IN OAUTH2

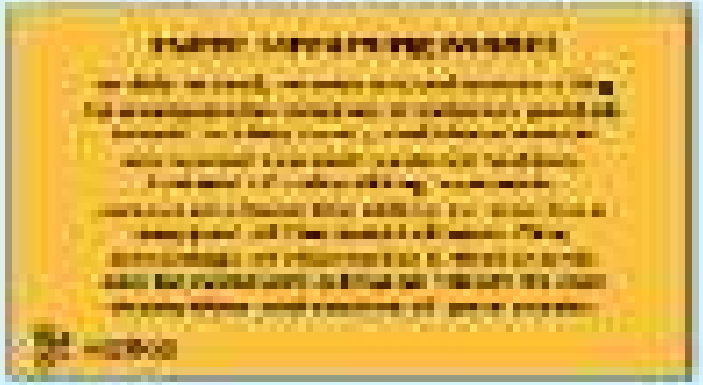
AUTHORIZATION CODE GRANT TYPE FLOW IN OAUTH2



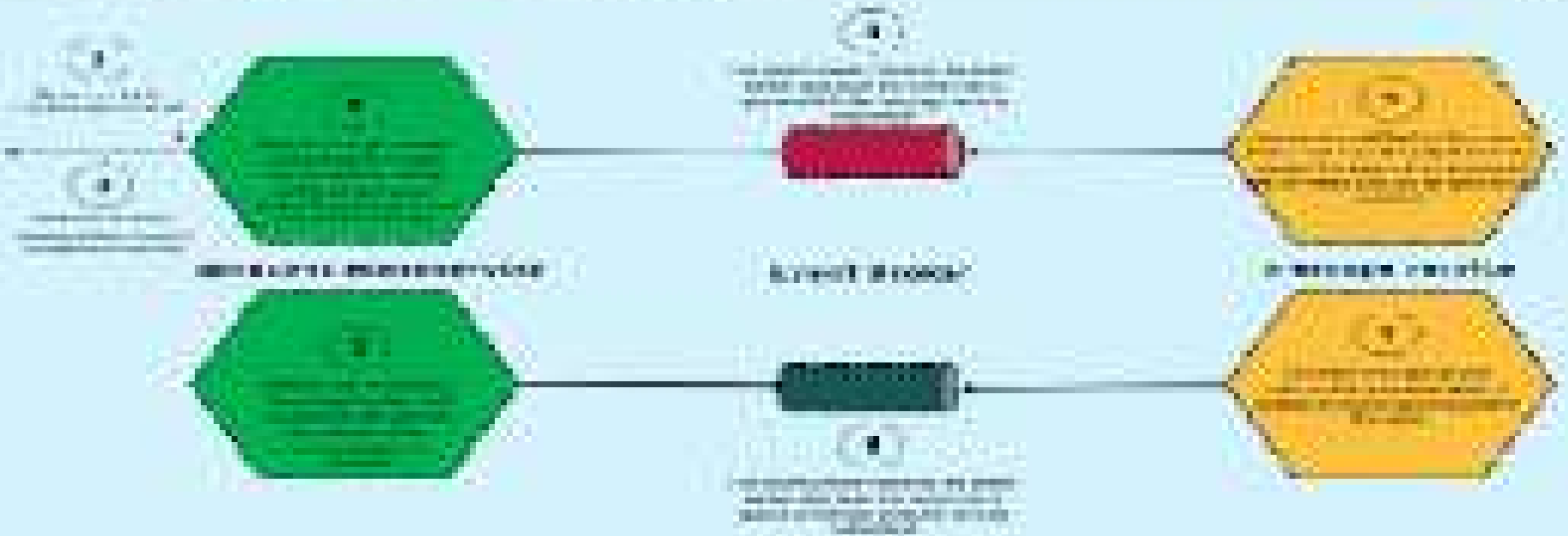




Event-driven models



What we are going to build using a poly/cyclic model



[illegible]

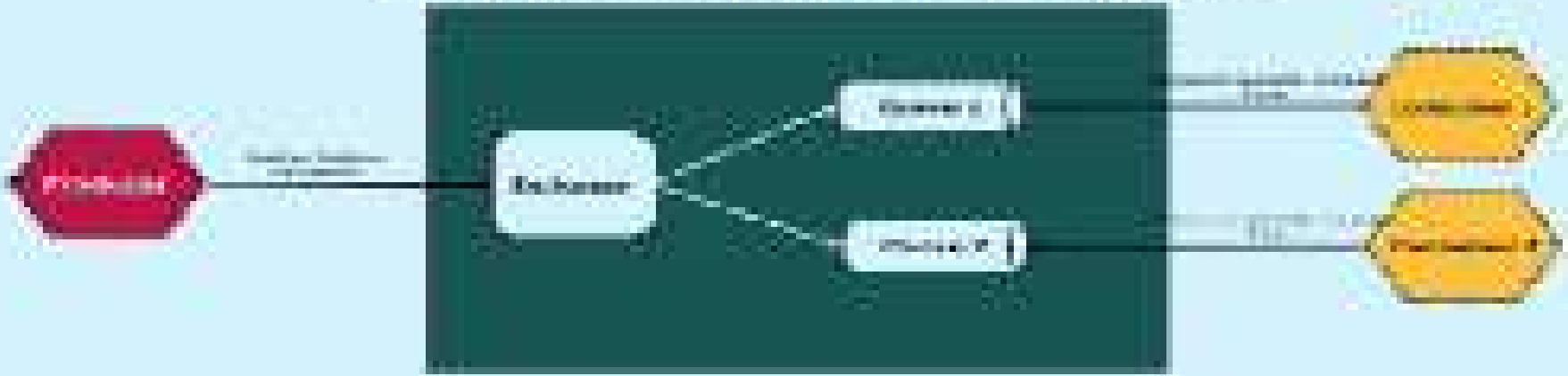
[illegible]

Prioritization | [illegible]

Consistent | [illegible]

Message Analysis | [illegible]

[illegible]


Producer
Exchange
Consumer 1
Consumer N

Why do use Spring Circuit Breaker?

Spring Circuit Breaker is a library that provides a set of functions for logic by utilizing functions such as retry, timeout, and fallback. It helps to build a resilient and robust system.

- **Resilience:** [illegible]
- **Functions:** [illegible]
- **Observability:** [illegible]

Spring Circuit Breaker Features

- Circuit Breaker when order, inquiry, remember is [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]
- [illegible]

Steps to create functions using Spring Cloud Functions

[illegible]

1. [illegible]

2. [illegible]

[illegible]

3

[illegible]

[illegible]

[illegible]

[illegible]

1. [illegible]

2. [illegible]

[illegible]

[illegible]

[illegible]

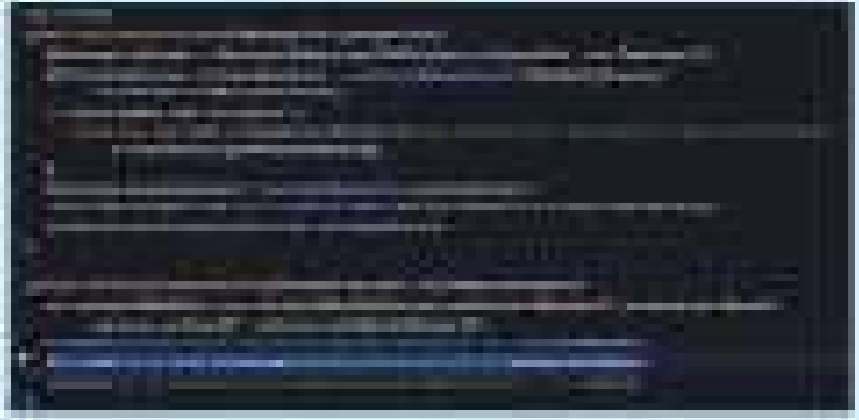
```
[illegible]
```

Event processing are consuming in accounts microservice

[redacted]

1. [redacted]

2. [redacted]



Event producing and consuming in accounts microservice

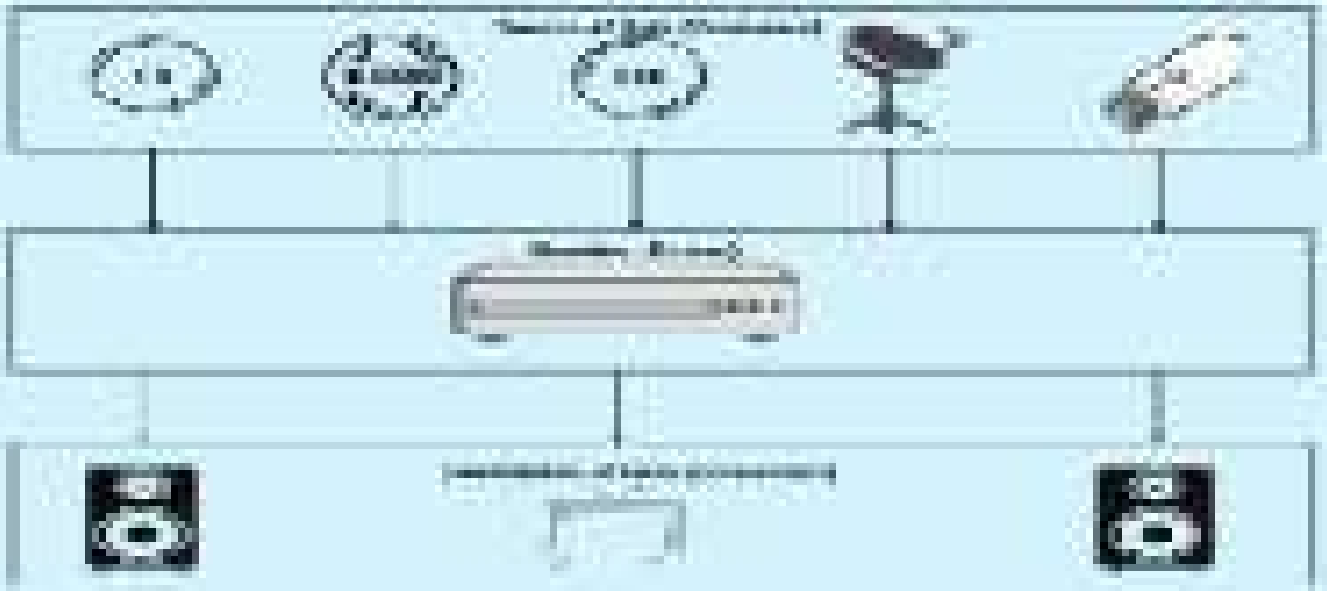
9

Event processing are consuming in accounts microservice

[illegible]

- **Language:** [illegible]
- **Data replication:** [illegible]
- **Clients:** [illegible]
- **Scalability:** [illegible]

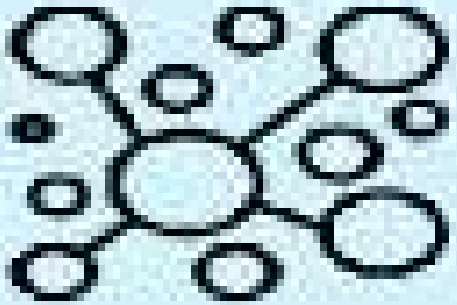
[illegible]



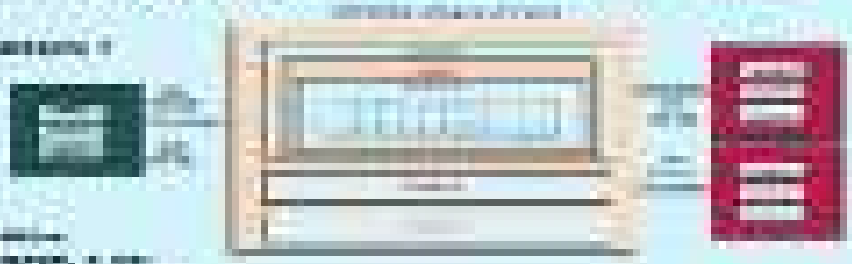
[illegible]

[illegible]

- **Producers:** [illegible]
- **Consumers:** [illegible]
- **Brokers:** [illegible]
- **Partitions:** [illegible]
- **Offsets:** [illegible]

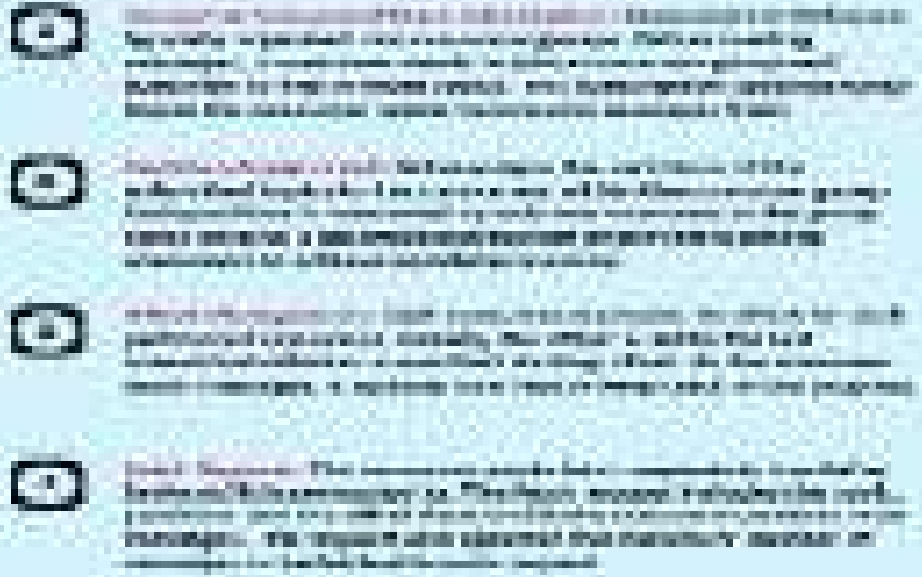


- **Rapid Growth:** [illegible] reduced to zero. Checking for failures and high availability.
- **Connectivity:** [illegible]
- **Expensive Storage:** [illegible]
- **Disparate:** [illegible]



Producer slide story

Consumer slide story



Steps to use Apache Kafka in the place of RabbitMQ

[illegible]

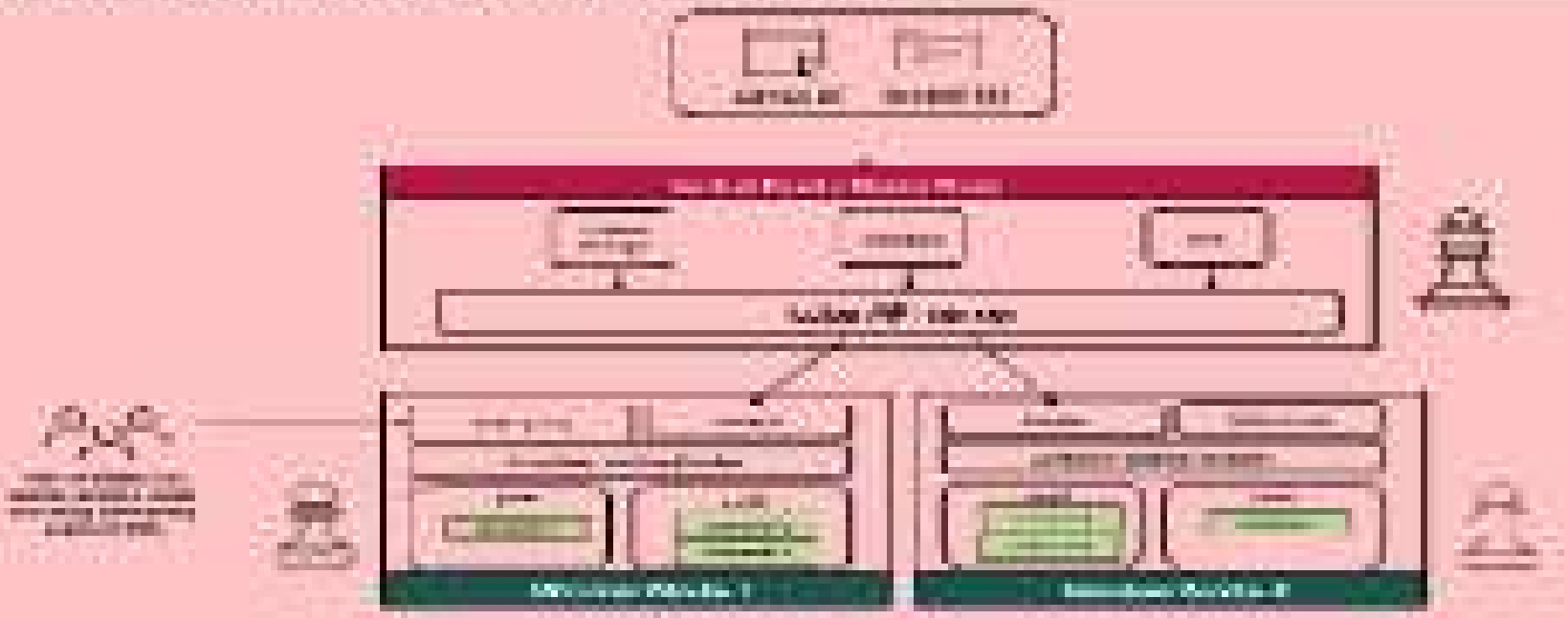
1. [illegible]

2. [illegible]

```
[illegible]
```

```
[illegible]
```

WHAT IS KUBERNETES ?



Components of Control Panel (Master Node)

[illegible]

[illegible]

- **AP** [illegible]
- **[illegible]** [illegible]
- **[illegible]** [illegible]
- **[illegible]** [illegible]

Components of Worker Node

Kubernetes manifest file to create ConfigMap

A Kubernetes-native ConfigMap is an resource kind Kubernetes manifest model for storing configuration data independently from the application code.

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: complex-configmap
data:
  application.properties: |-
```

This [illegible] and kind fields are required for all Kubernetes objects. When creating a ConfigMap inside a file, set kind value to "ConfigMap".

The metadata field contains the name of the ConfigMap and other metadata about the object.

The data field is where the key-value pairs are stored. The keys can be any alphanumeric string, and the values can be strings, numbers, or Boolean data.

Kubernetes manifest file to deploy a Container

Kubernetes manifest file to deploy a Container

Kubernetes manifest file to create a service

[illegible]

Kubernetes manifest file to create a service

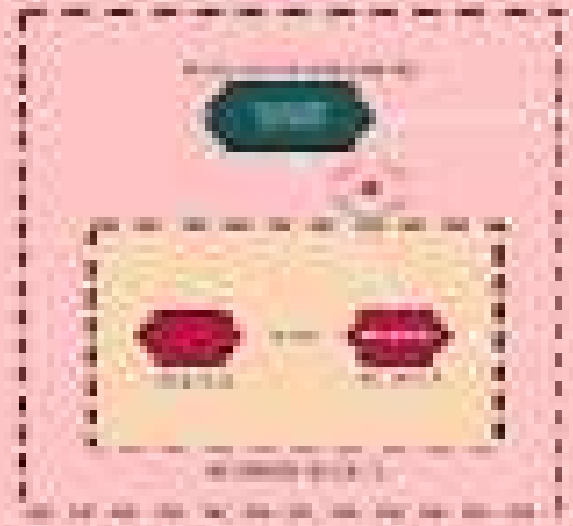
[illegible]

[illegible]

[illegible]

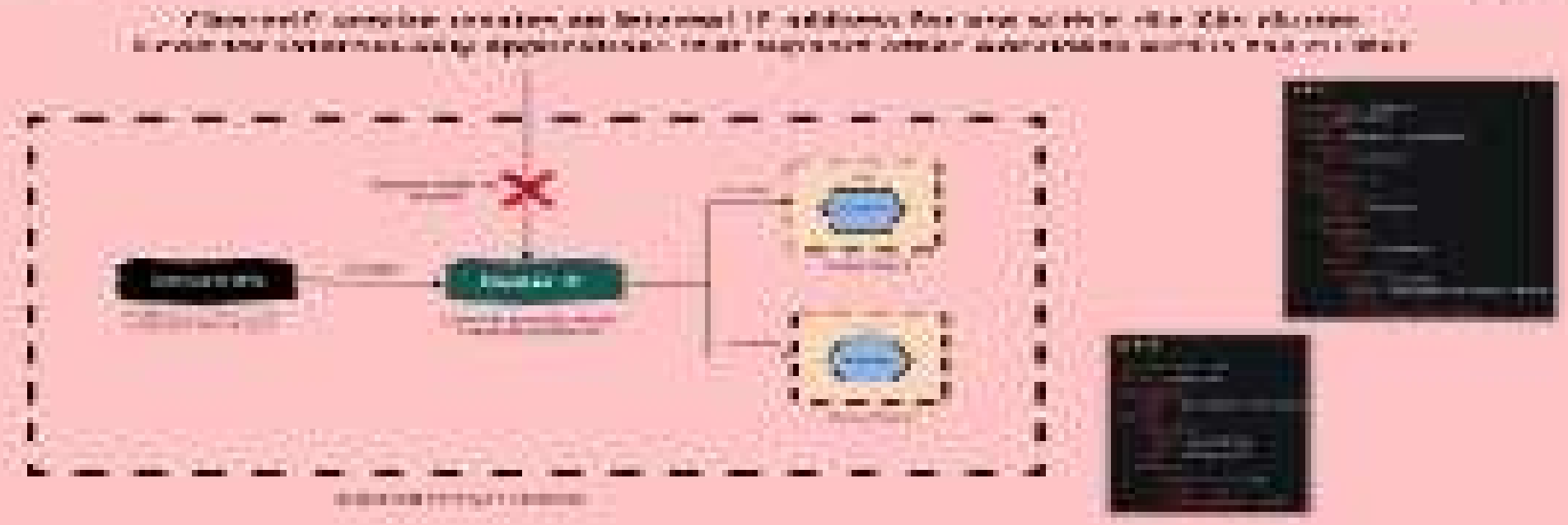
[illegible]

How K8s deployment & services tied together

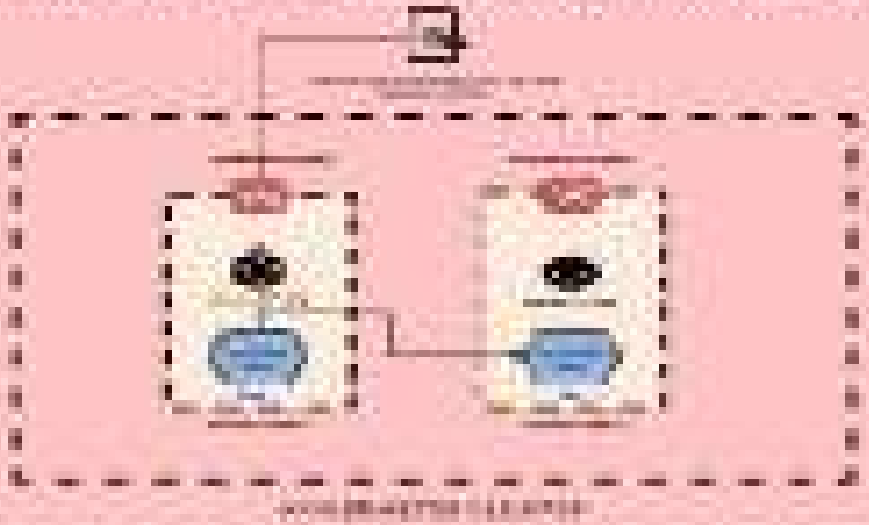


Entrepreneurial Service Types

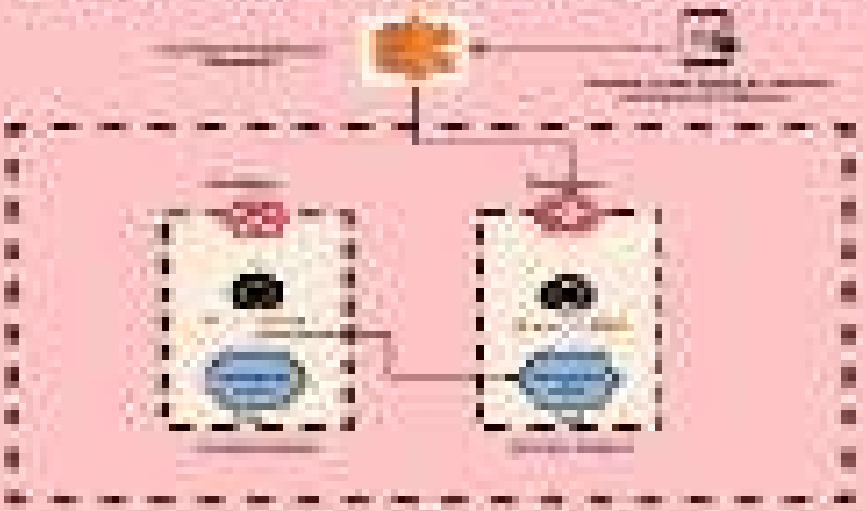
K8s CLUSTER IP SERVICE



EKS NODEPORT SERVICE



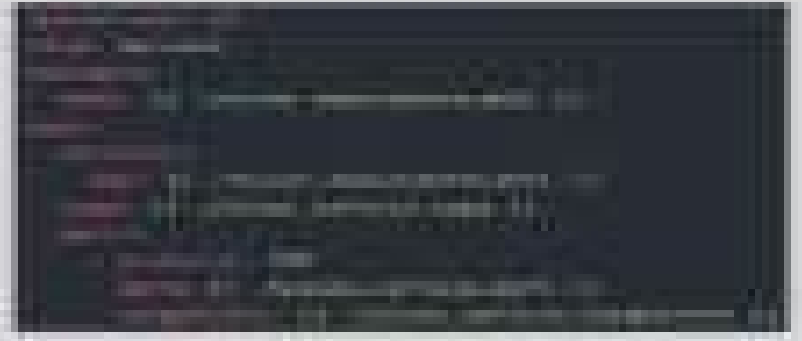
ELB LOADBALANCER SERVICE



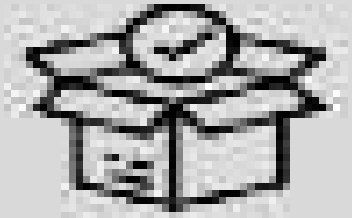
What is HELM ?

Problems that Helm solves

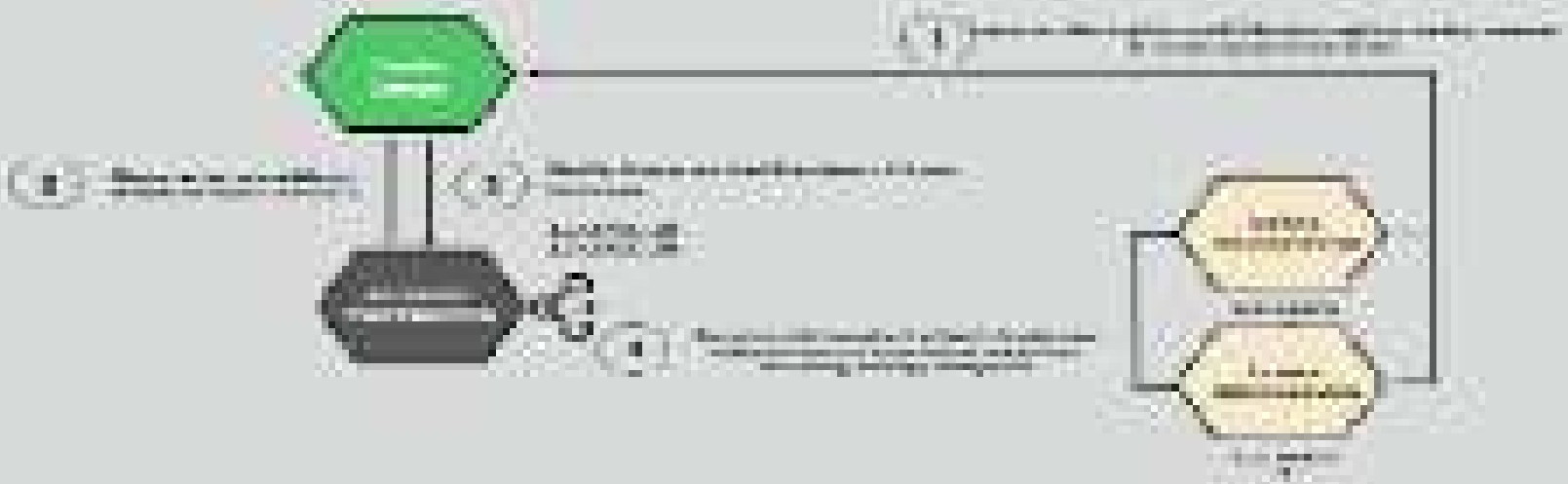
Problems that Helm solves

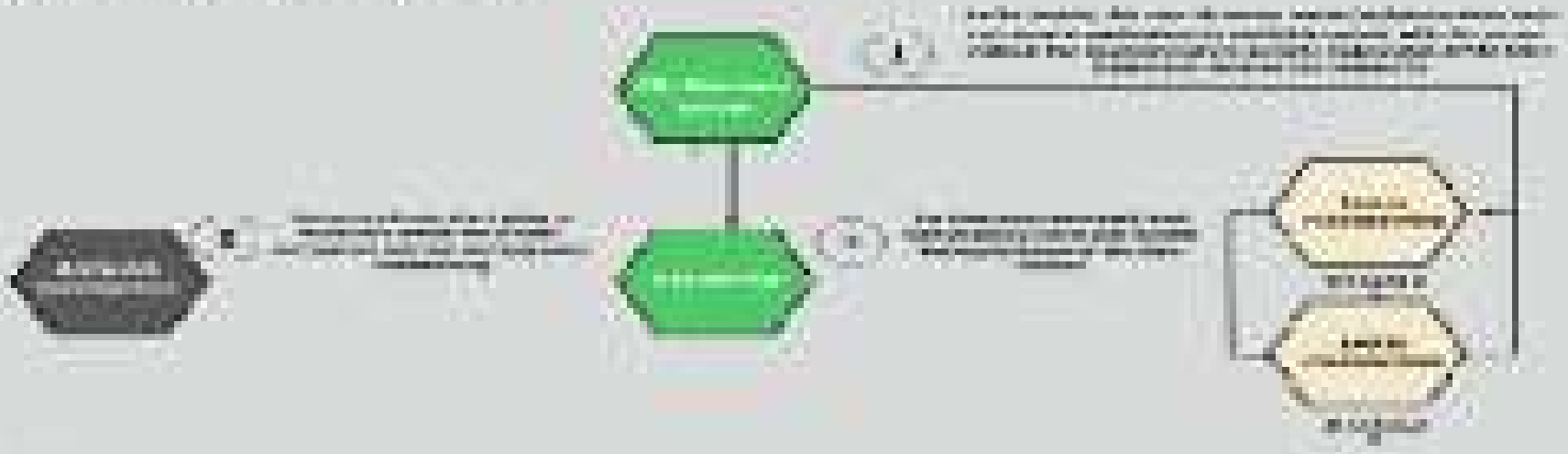


Problems that Helm solves



[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]



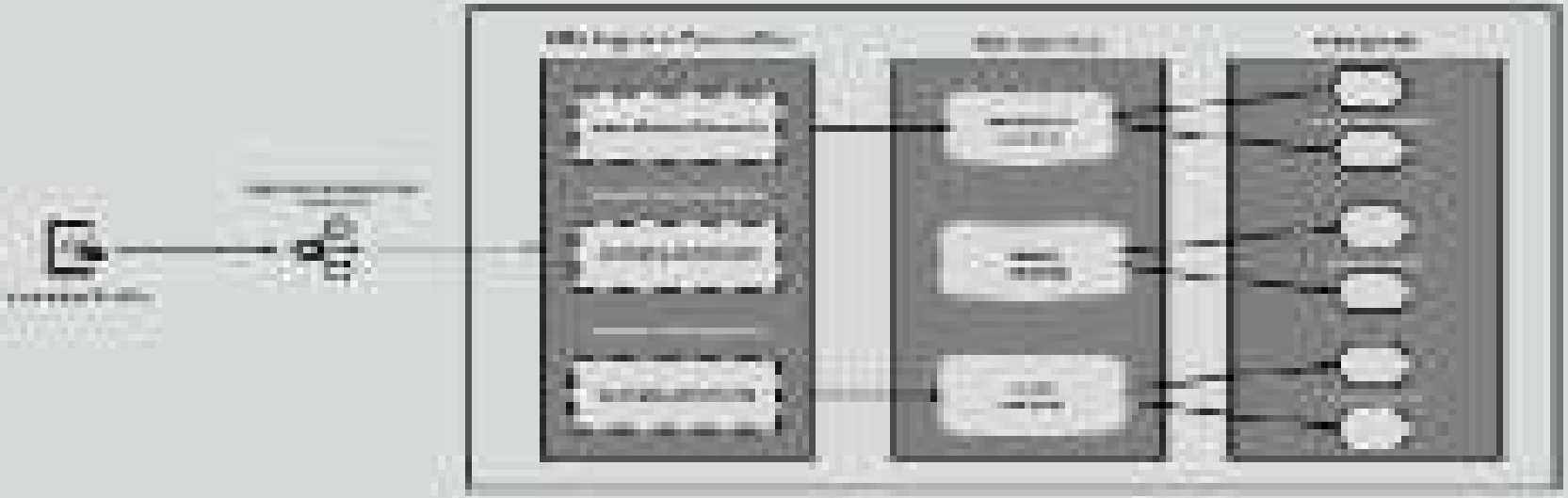


Kubernetes support by Cloud providers

What is Ingress?

Ingress is Layer 7 (HTTP/HTTPS) Load Balancer

- [illegible]
- [illegible]



What's the Issue?

Ingress provides simple HTTP/S routing

Centralized Policy: Ingress can be configured to enforce single sign-on, rate limiting, and other policies across multiple services, ensuring consistent security and compliance.

Scalable Architecture: Ingress can handle traffic load balancing and routing, allowing your applications to scale horizontally without complex configurations.

Path-Based Routing: You can route traffic to different services based on URL paths, enabling microservices architectures and API gateways.

Host-Based Routing: Traffic is routed to services based on the requested host name, supporting multi-tenant applications and virtual hosting.

Load Balancing: Ingress distributes incoming traffic across multiple pods, ensuring high availability and optimal resource utilization.

Simplification:
Ingress provides a single entry point for all HTTP/S traffic, simplifying network configurations and enhancing security by centralizing access control.

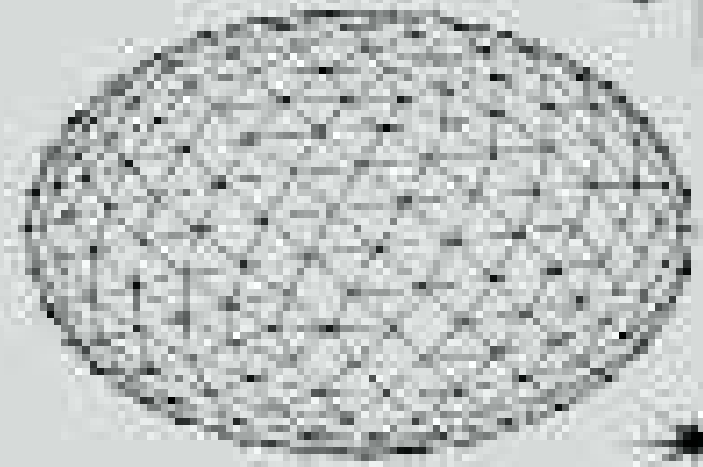
Ingress Controller vs. Service Type LoadBalancer

[illegible]

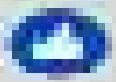
Types of Traffic Handled by Ingress Controller

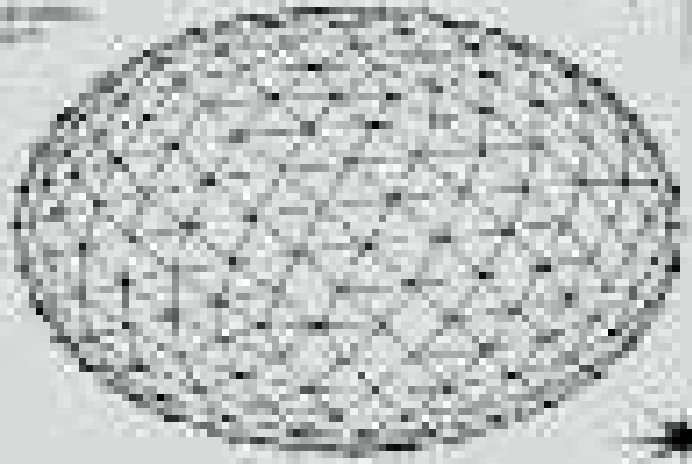
[illegible]

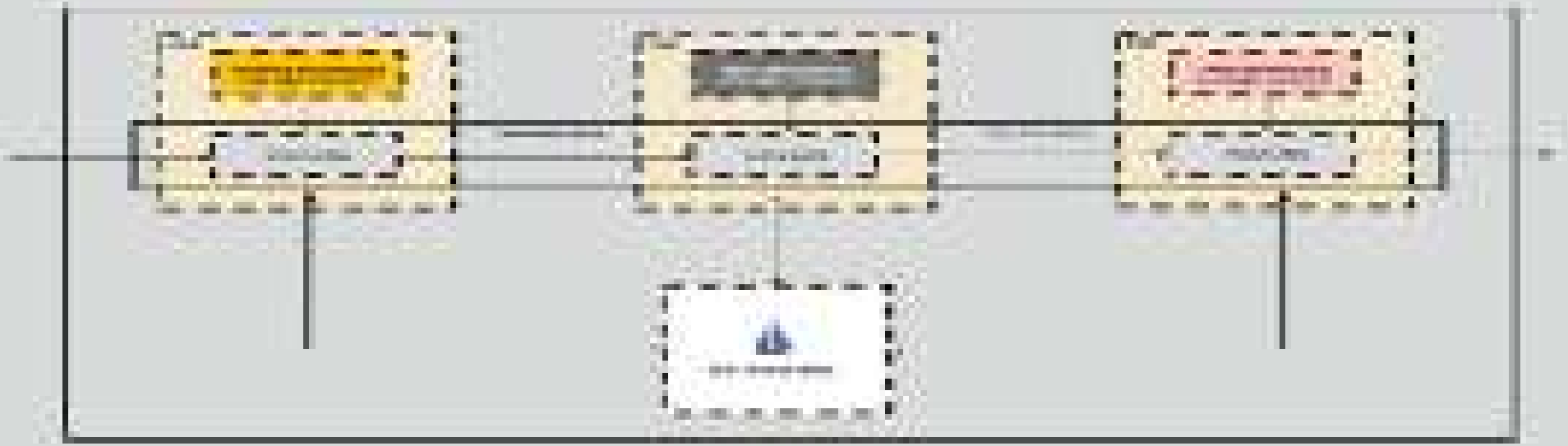
Who handles service-service traffic?



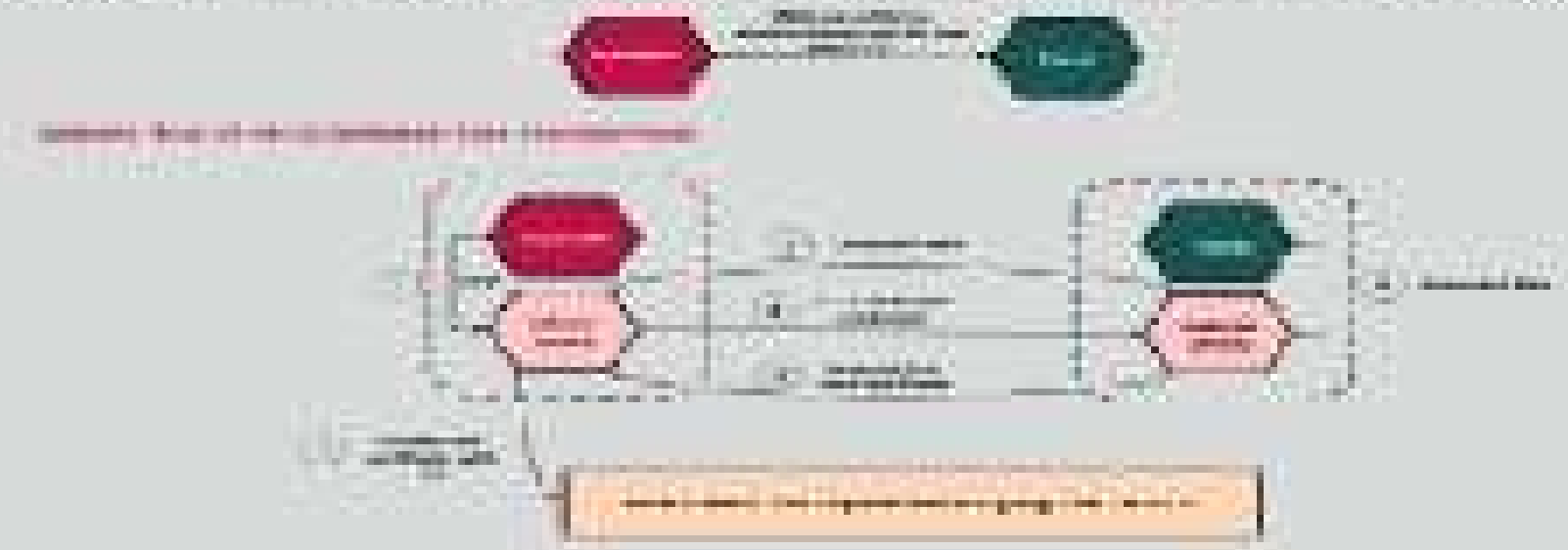
How Nirvana Sketch makes our life easy with its building information







How Is mTLS Different from TLS?



Why Leo mirks?

Why can't he support Leave Social By [illegible] [illegible] Support [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

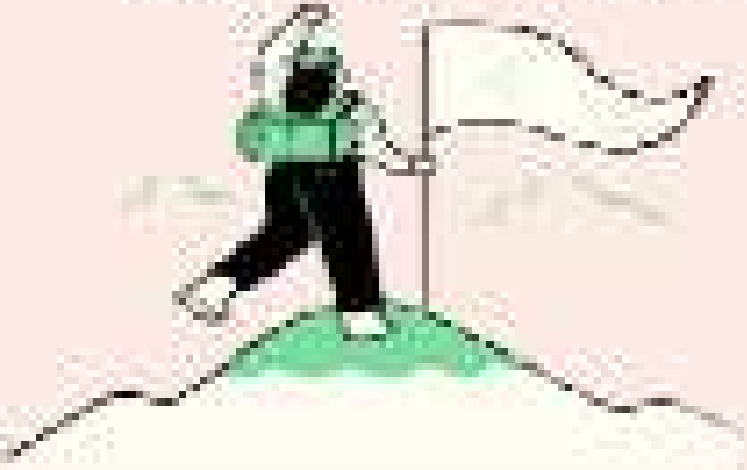
[illegible]

[illegible]

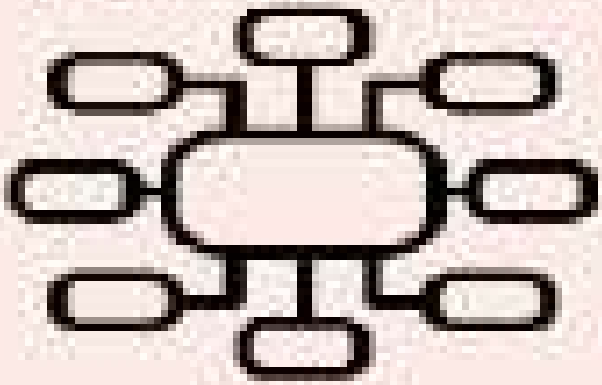
[illegible]

[illegible]

Optimizing Microservices Development with Spring Boot BOM



Shared Libraries in Microservices



CONGRATULATIONS

A BIG THANK YOU