{ run(microservices) }

@sendilkumarn

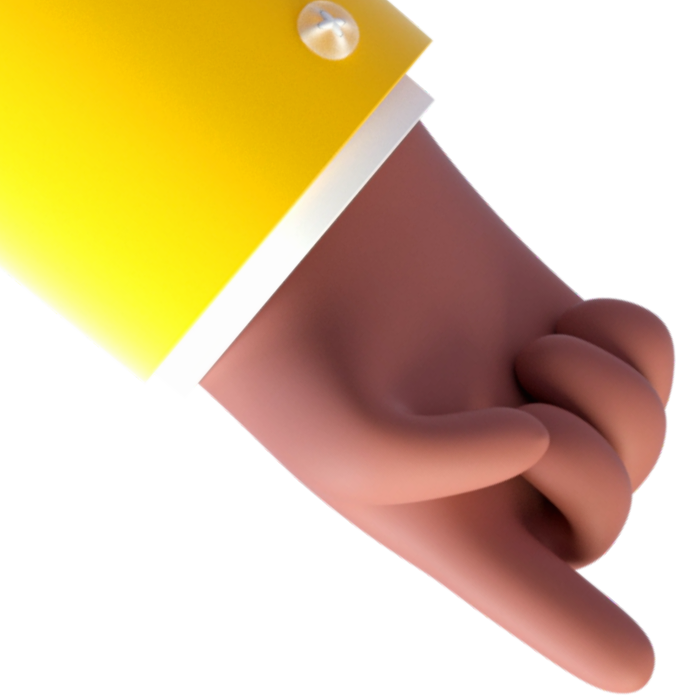
Cloud?.let
{ run(microservices) }

@sendilkumarn

Pitch

Kubernetes?.
Cloud?.let
{ run(microservices) }

Kotlin?.
Kubernetes?.
Cloud?.let
{ run(microservices) }



Microservice

Let us define the applications

@sendilkumarn

Microservice

Store Service

Gateway

User Interface

Users / Authentication

SQL Based Database

Product Service

Microservice

NoSQL Database

No User Interface

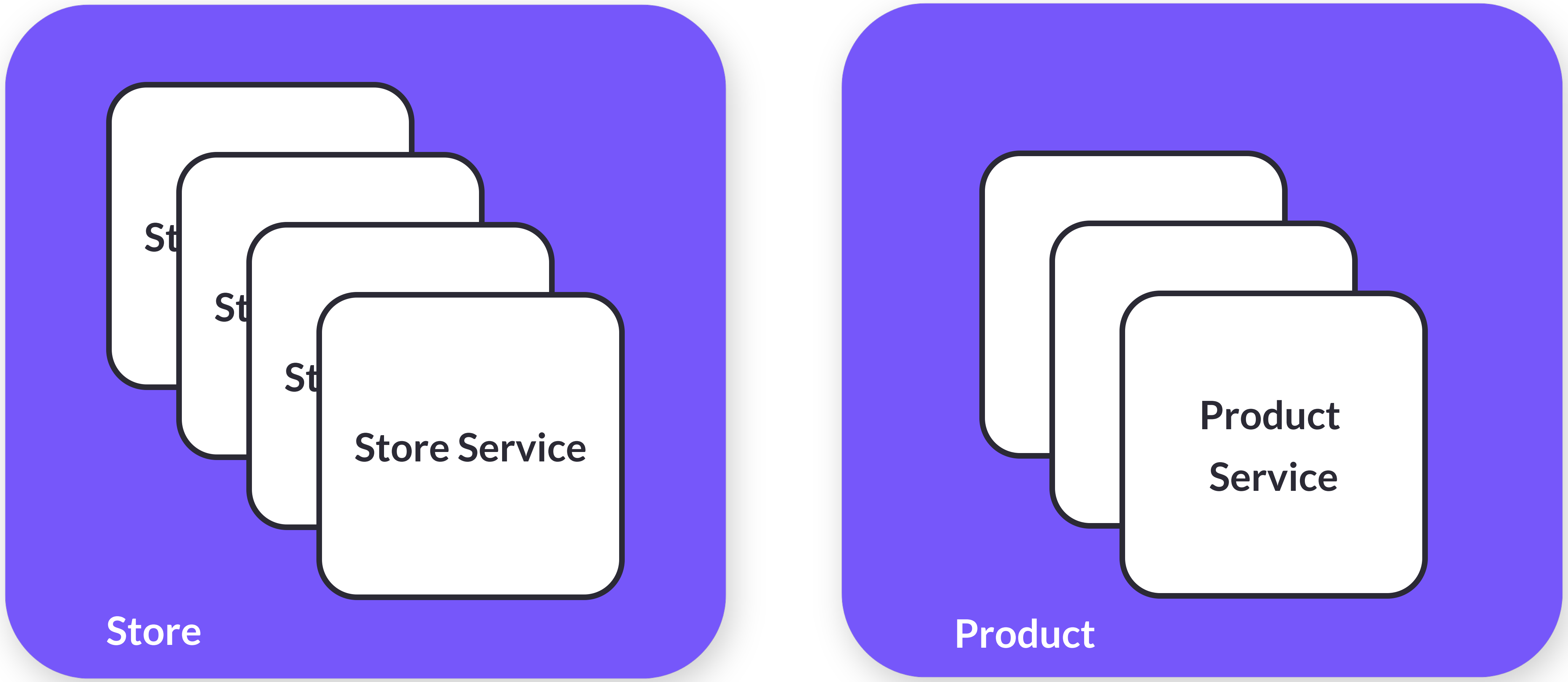
Store Service

```
application {
    config {
       baseName          stoapp
       applicationType   gateway
       prodDatabaseType  postgresql
       clientFramework   react
    }
}
```

Product Service

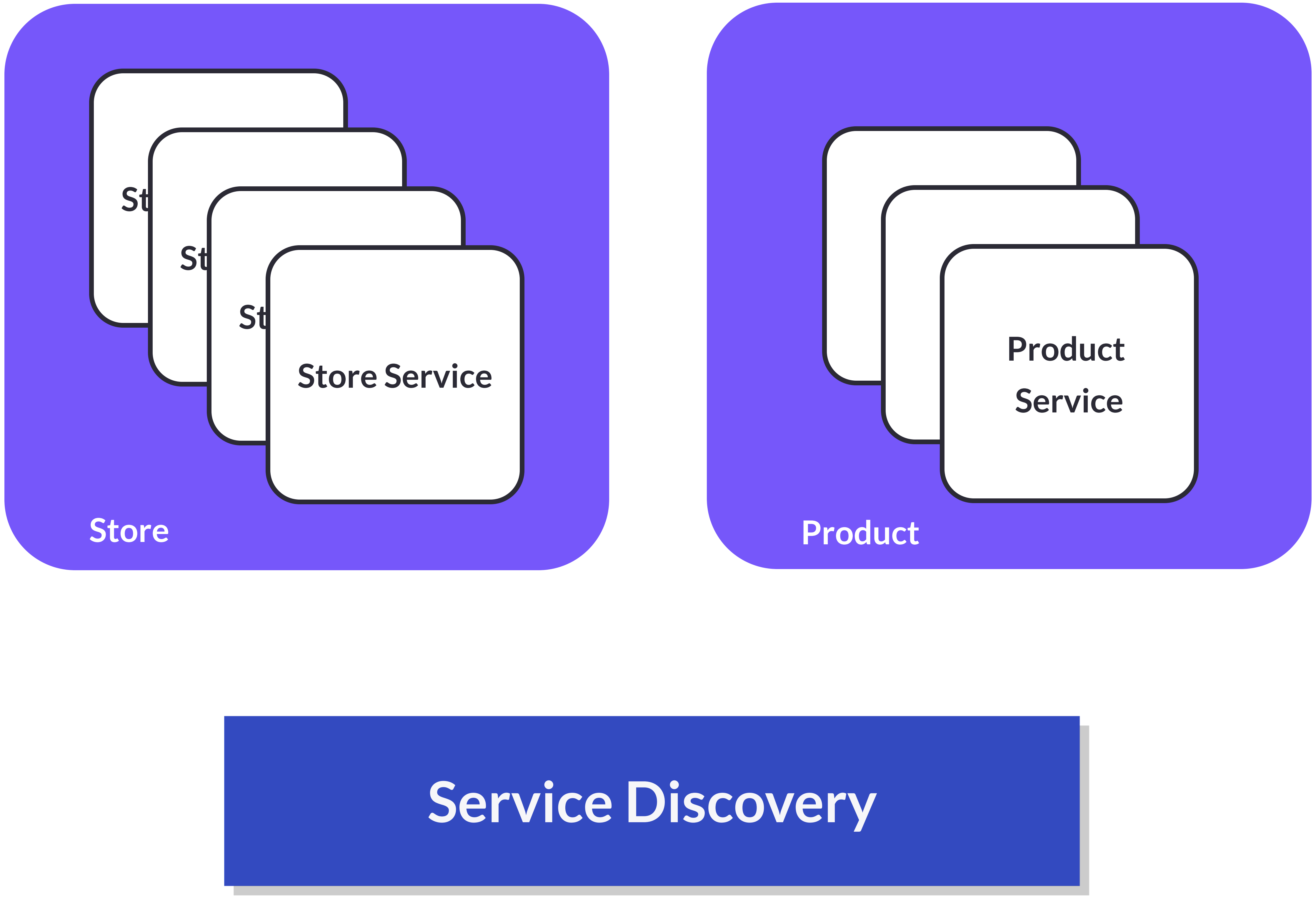
```
application {
    config {
       baseName          proapp
       applicationType   microservice
       databaseType      mongodb
    }
}
```

Services are scalable & isolated



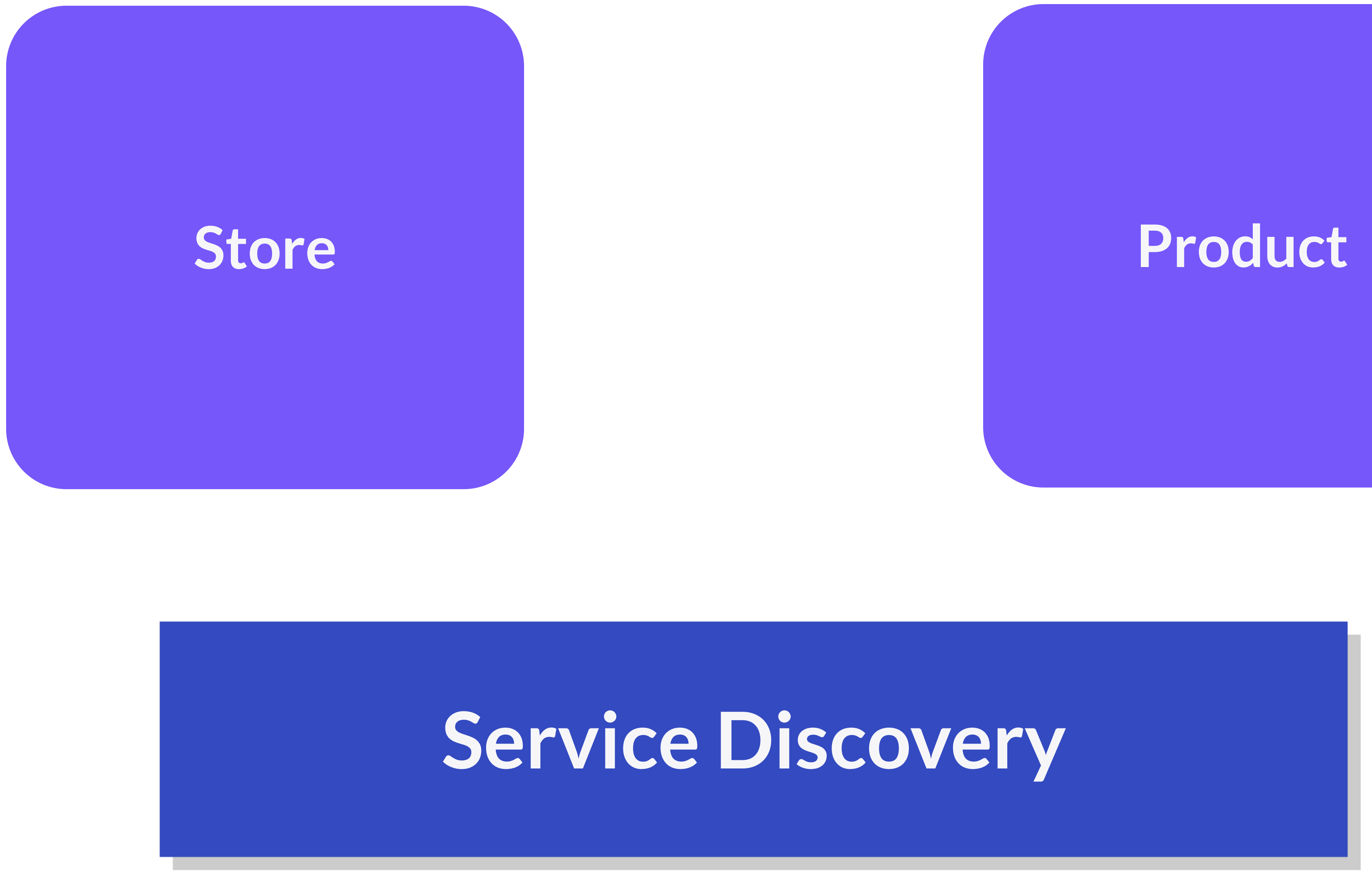
@sendilkumarn

Service Discovery

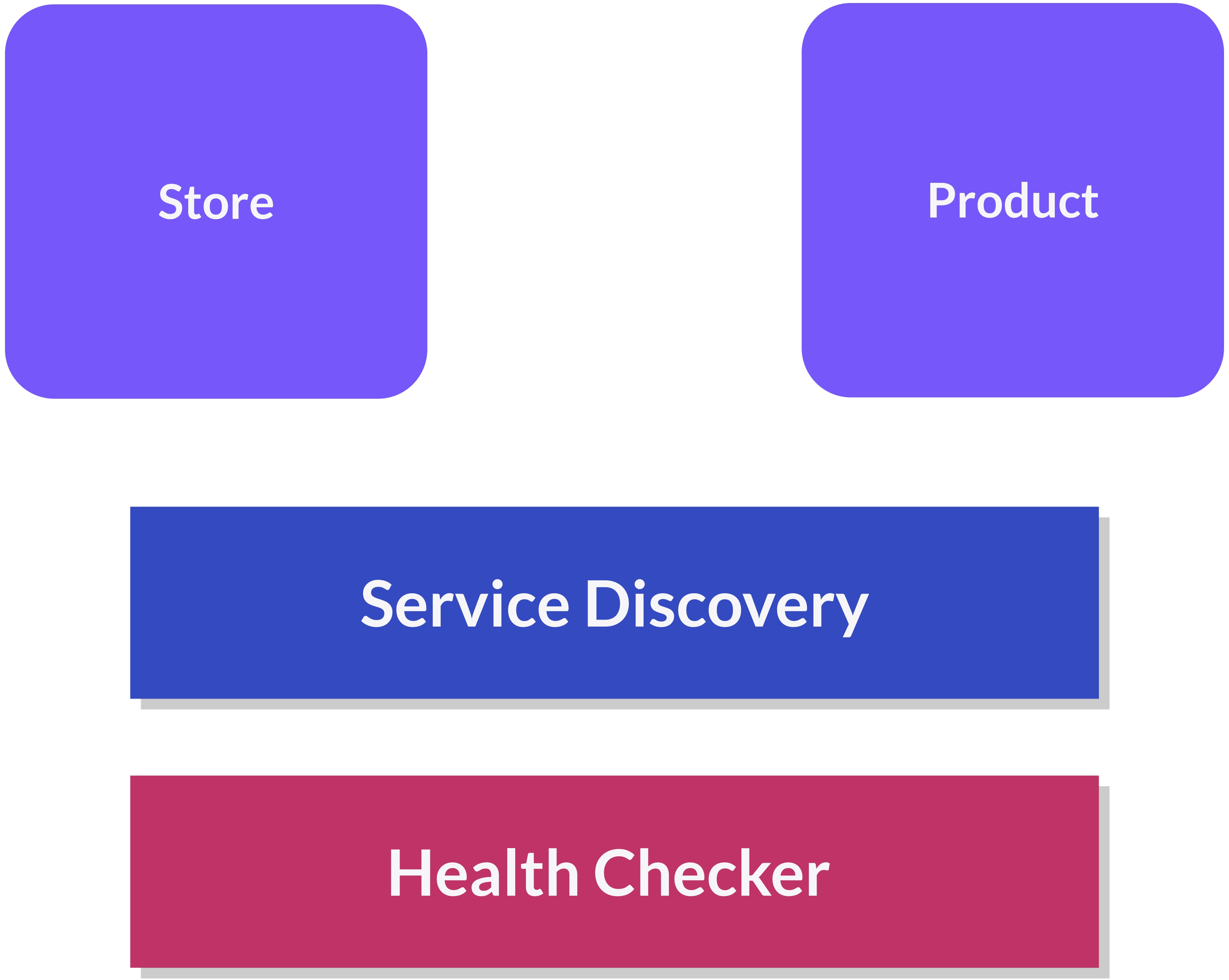


@sendilkumarn

Service Discovery



Health Checker



Containarize

Store

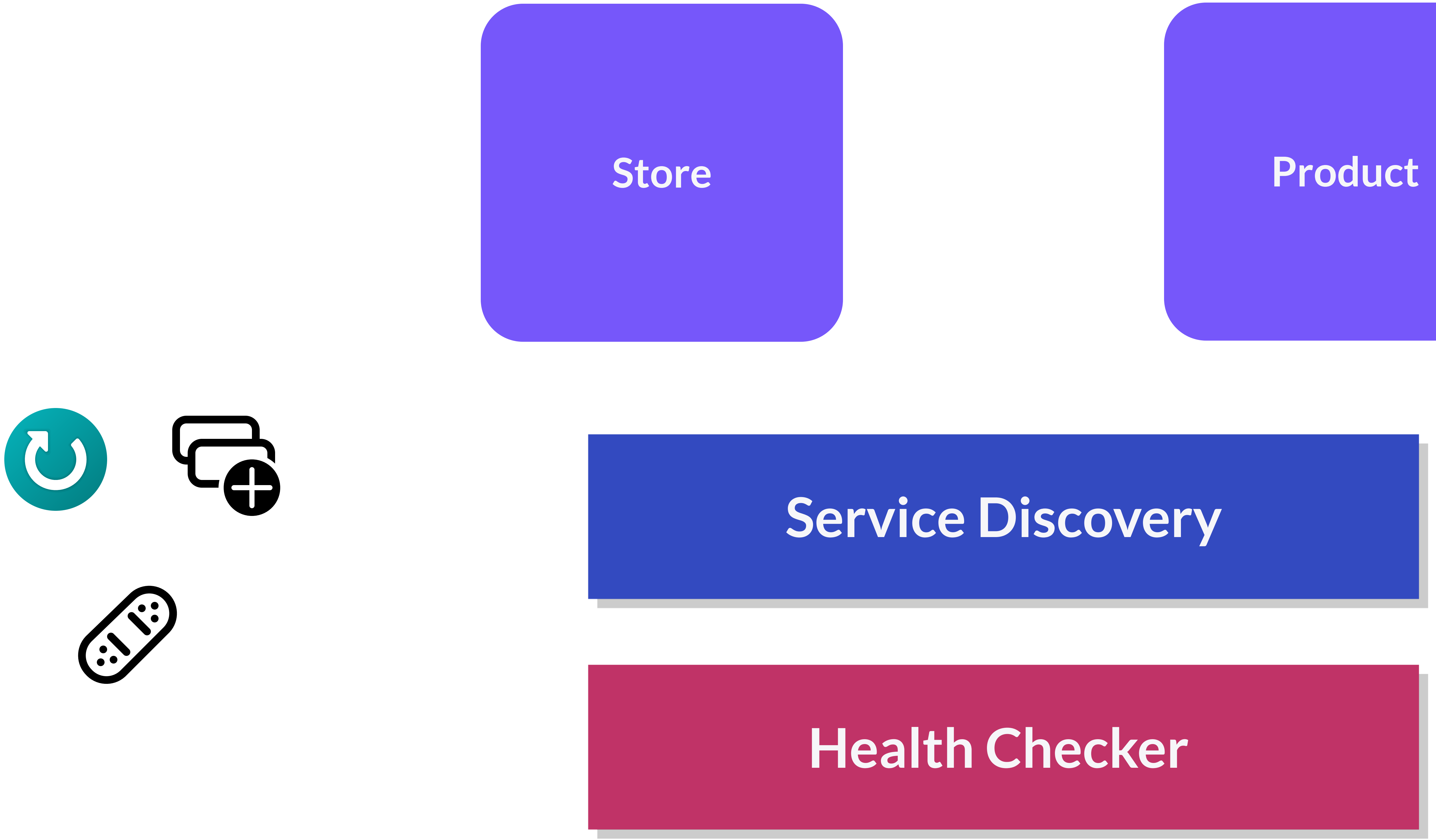
Product

Service Discovery

Health Checker

@sendilkumarn

Restart / Replicas / Scaling



Secure / Configure / Control

Store

Product

Service Discovery

Health Checker

Metrics / Measure / Monitor

Store

Product

Service Discovery

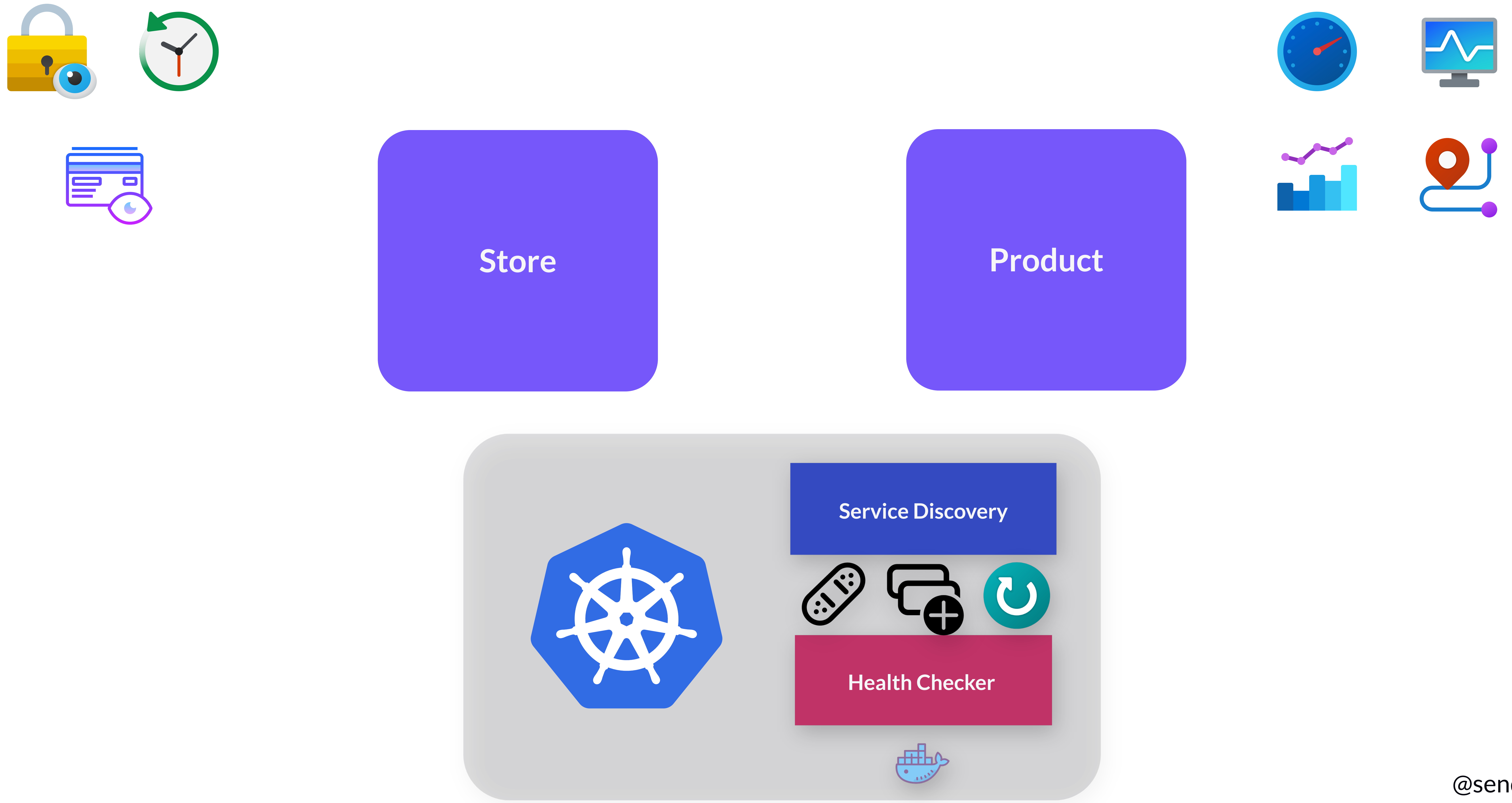
Health Checker

@sendilkumarn

Pitch

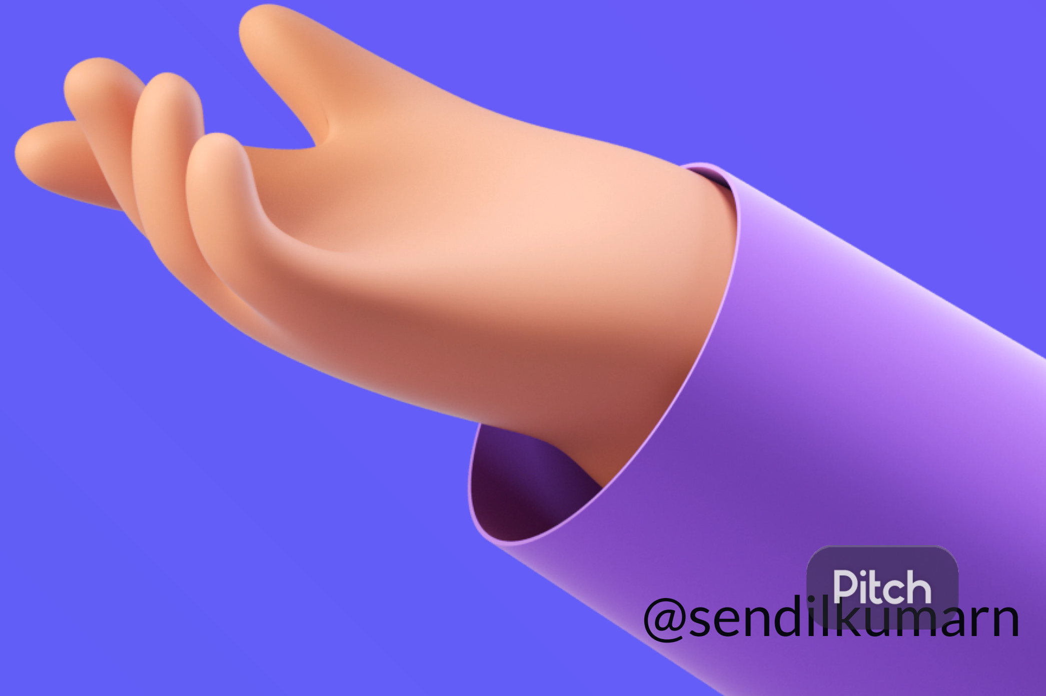
What if?

Kubernetes



Kubernetes

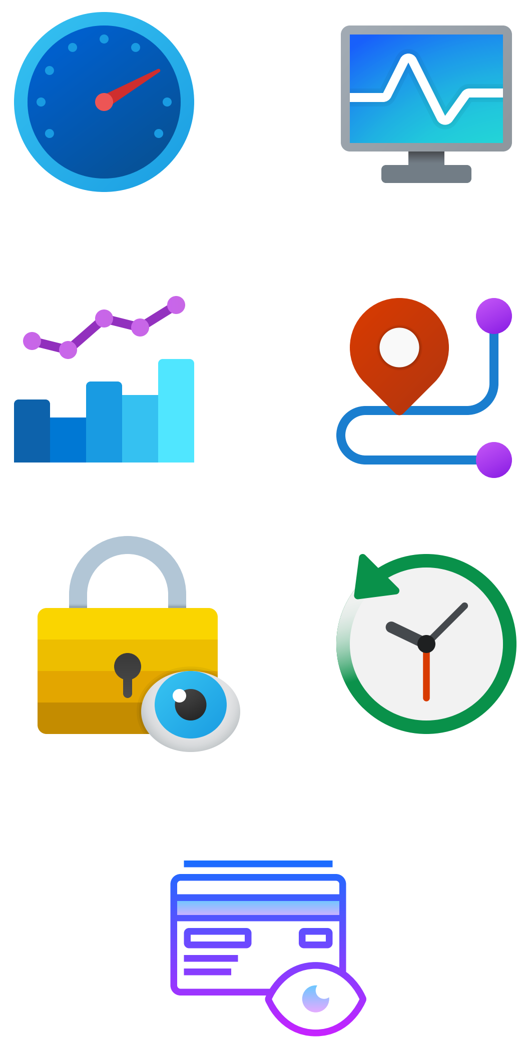
Takes care of infrastructure / housekeeping and others



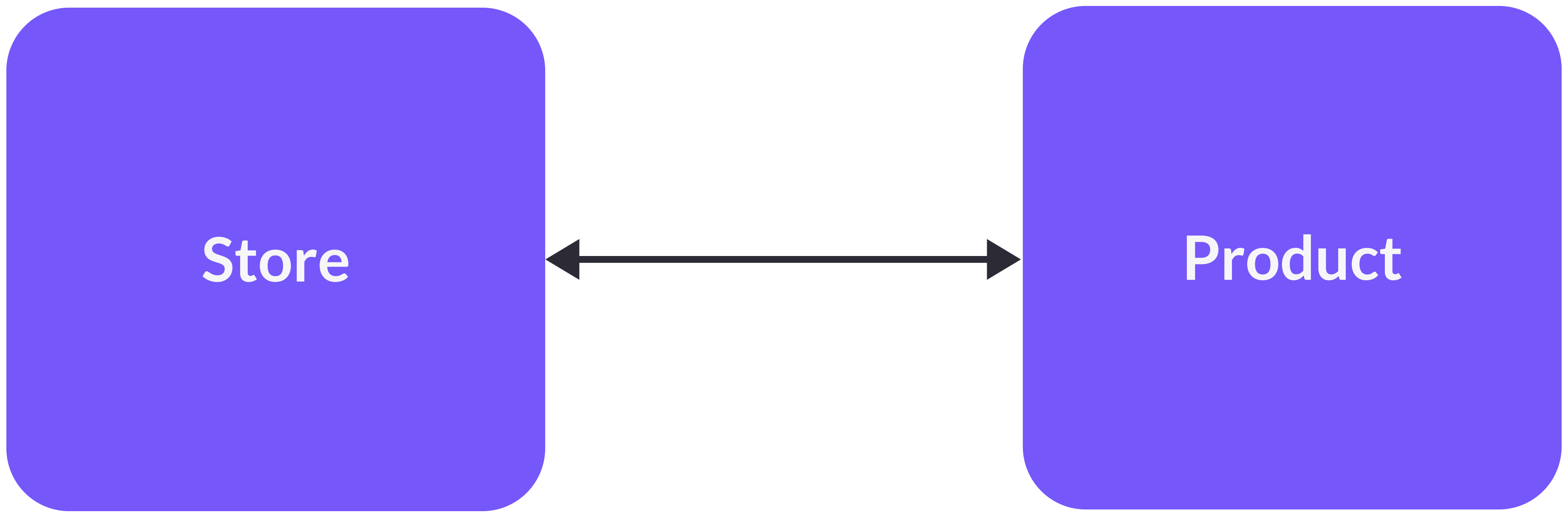
But what about these?

Store

Product

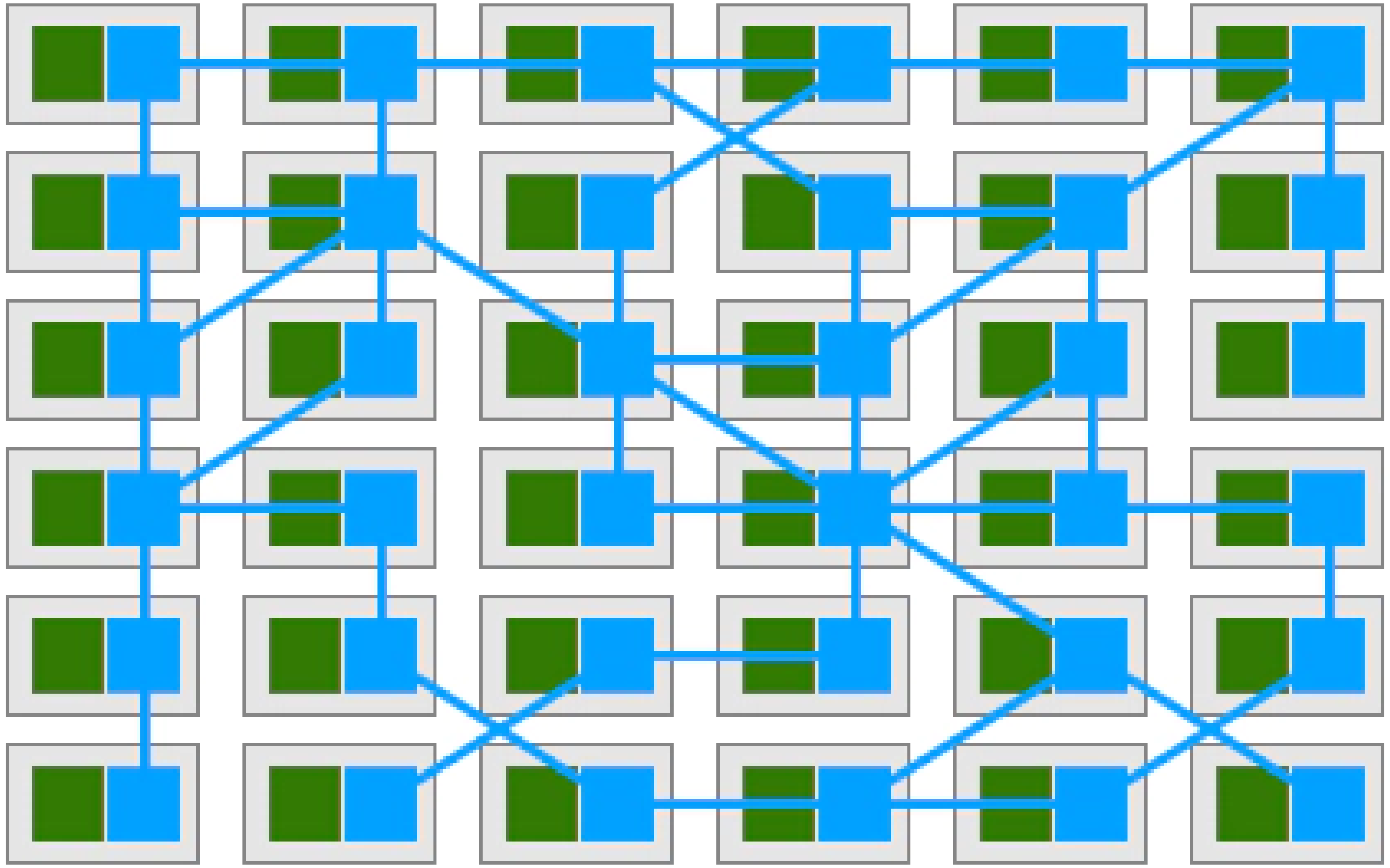


@sendilkumarn



@sendilkumarn

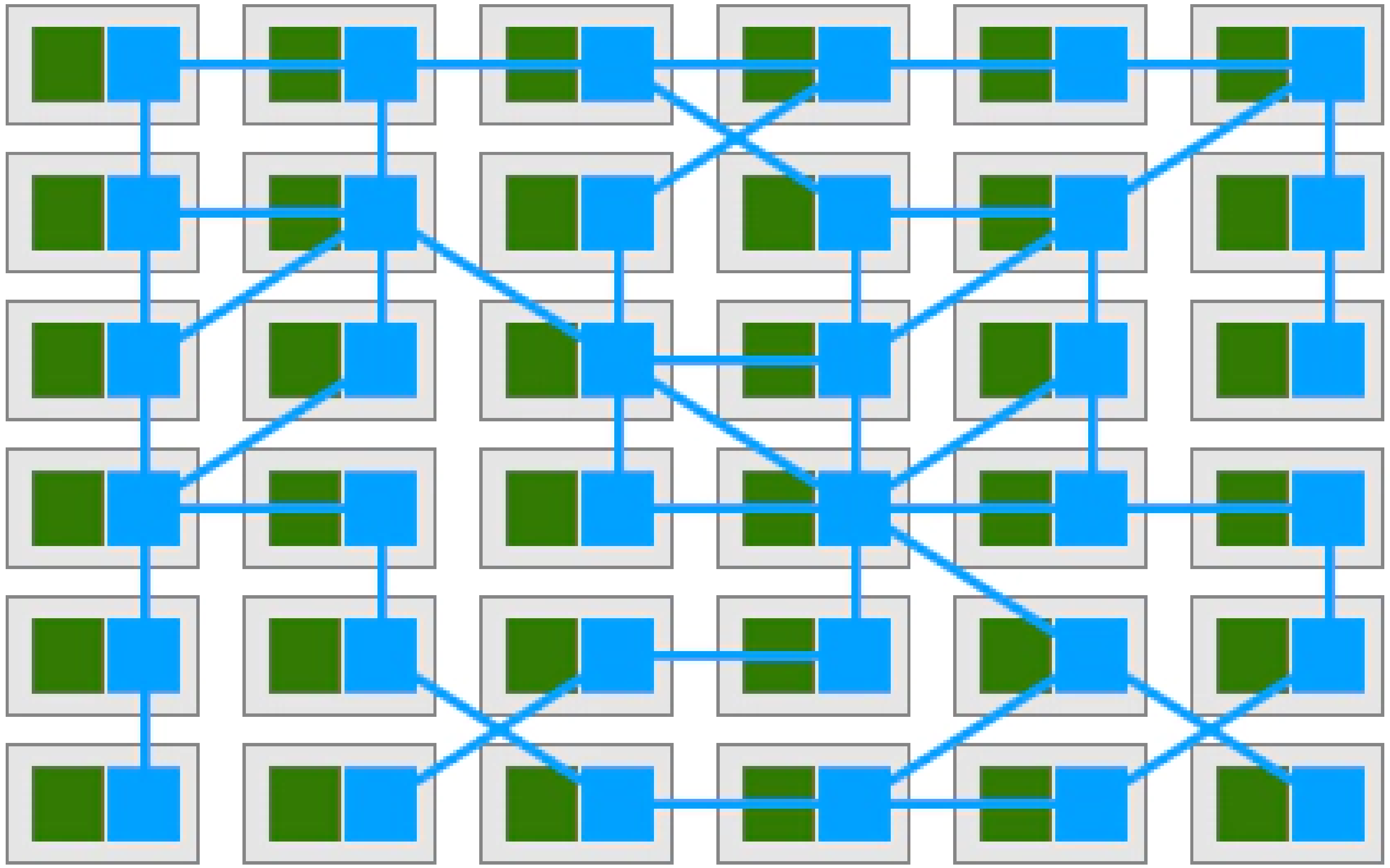
Service Mesh



@sendilkumarn

Pitch

Service Mesh



@sendilkumarn

Istio

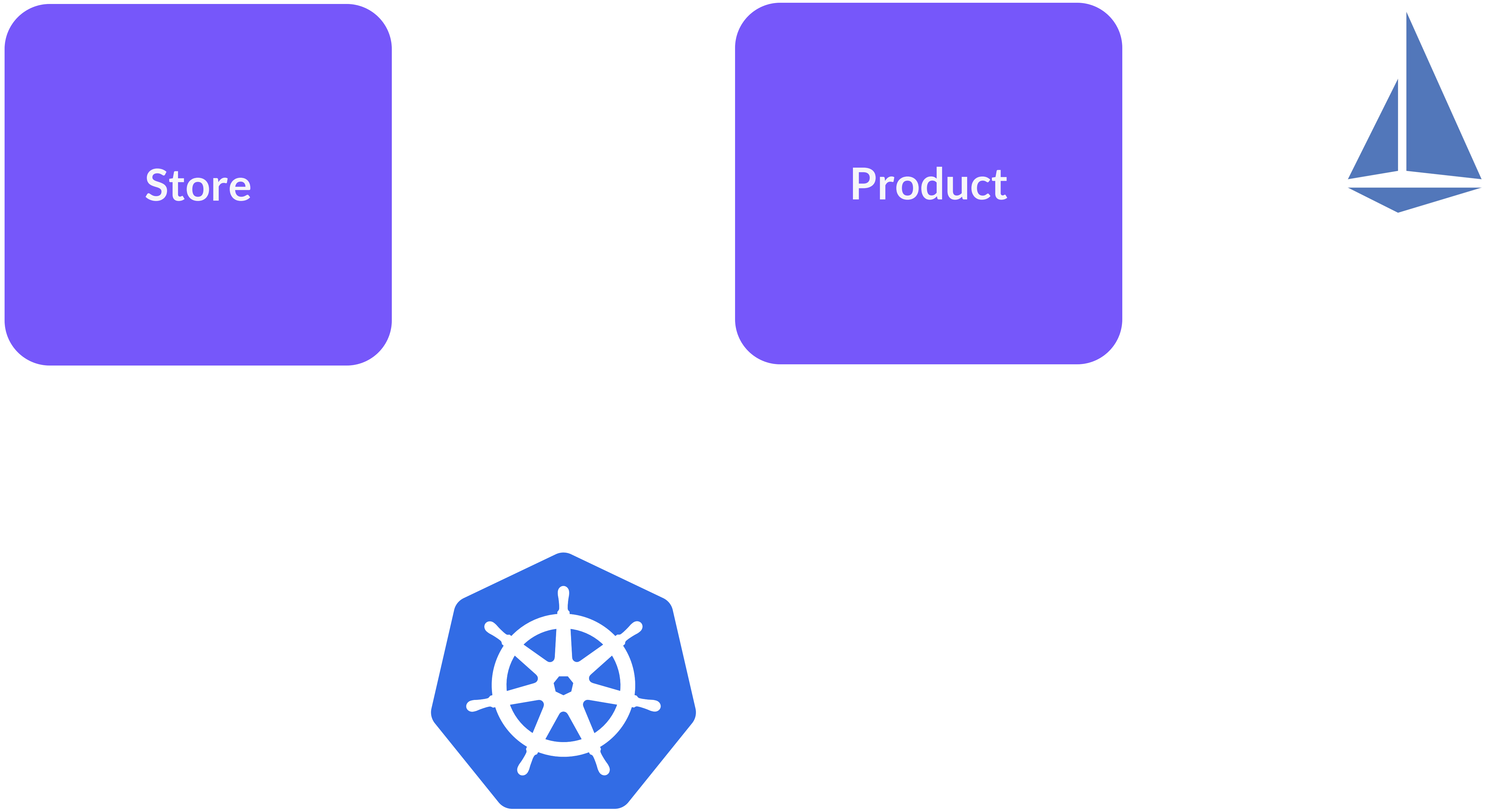
Store

Product

Istio

Control / Manage / Secure / Connect

Easy Microservices world



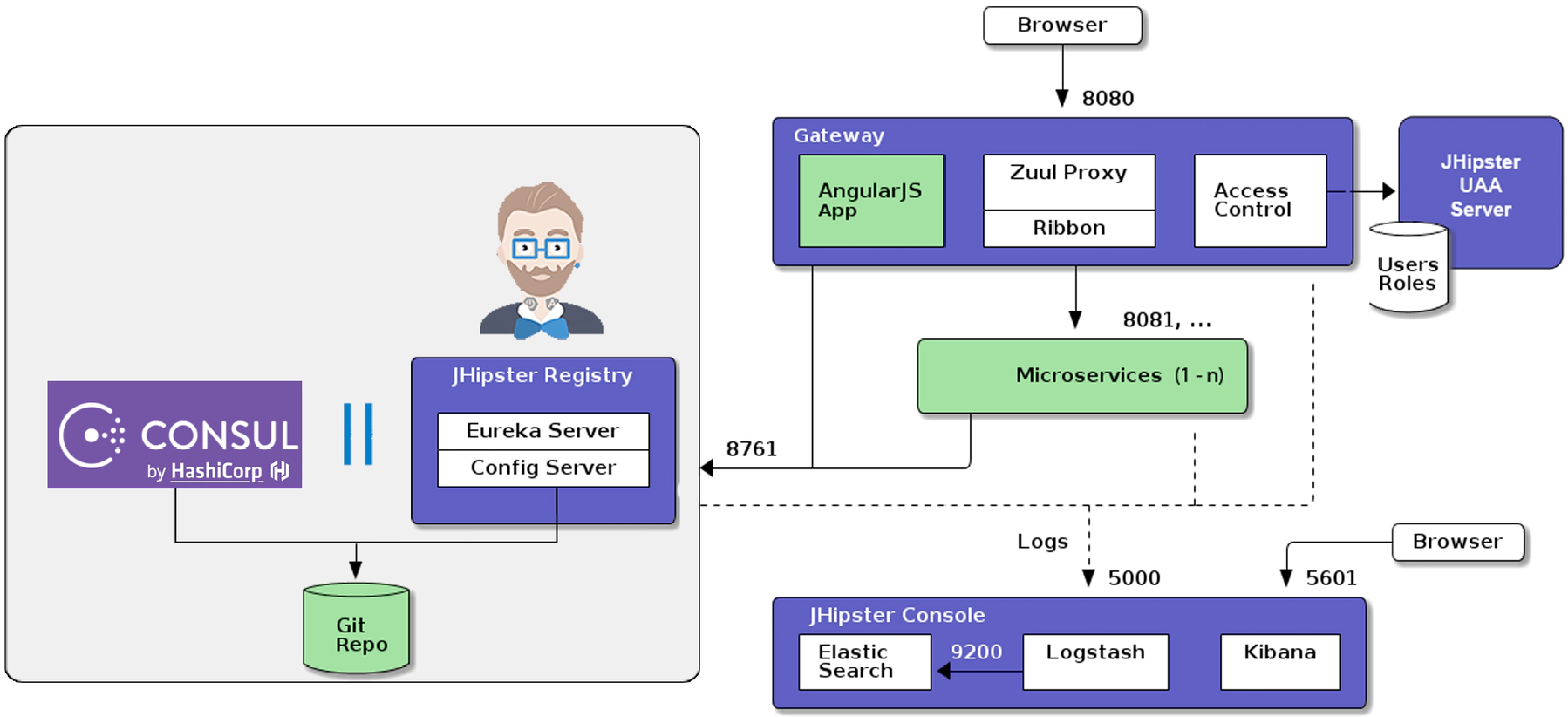
```
application {
    config {
      baseName              stoapp
      applicationType       gateway
      databaseType          postgresql
      clientFramework       react
      serviceDiscoveryType no
    }
}
application {
   config {
     baseName              proapp
     applicationType       microservice
     databaseType          mongodb
     serviceDiscoveryType no
   }
}
```

```
deployment {
    deploymentType        k8s
    appsFolder            [stoapp, proapp]
    dockerRepositoryName  "sendilkumarn"
    kubernetesServiceType ingress
    istio                 true
    ingressDomain         "<ip-address>"
    serviceDiscoveryType  no
}
```

KHipster

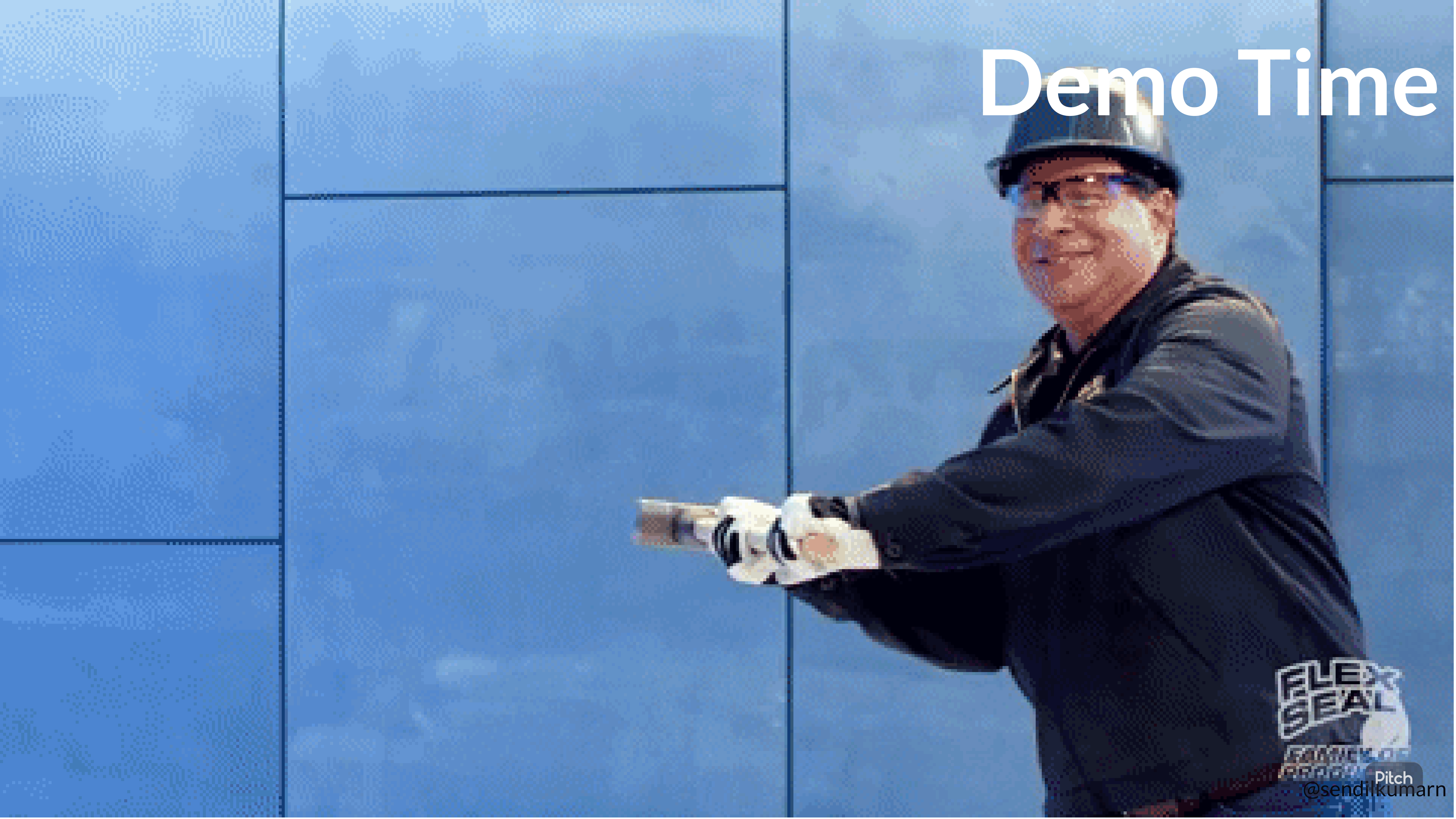
KHipster is a **development platform** to quickly **generate, develop, & deploy** modern web applications.

```
npm i -g generator-jhipster-kotlin
khipster
```



@sendilkumarn

Demo Time



@sendilkumarn