

**JUG**

**Real World  
Kubernetes**



# Agenda

- Introduction Agenturclient
- Architecture Agenturclient
- Kubernetes Deployment
  - HA-Setup
  - Demo
- Istio Introduction





# Florian Lüscher

Solution Architect and Co-Founder

## Skills

- Solution Architecture
- Cloud & Continuous Delivery
- Machine Learning
- Java, Python, .NET (Core)
- Web

## CV

- seit 2018  
dsi engineering ag
- 2013 - 2018  
Zühlke Engineering AG
- 2010 - 2013  
FHNW Computer Science
- 2005 - 2009  
Software Development  
Apprenticeship





*Wir helfen unseren Partnern  
intelligente Services zu entwickeln.*

**Agenturclient**

 Timetable

 Products

 Bookings

 Info

## Buy tickets

From

To

+ ADD VIA

Depart

18.06.2019

Time

07:24

Dep



Arr

Return

Time

Dep



Arr

Passenger

Adult (16+) ▼

Discount card

No discount ▼

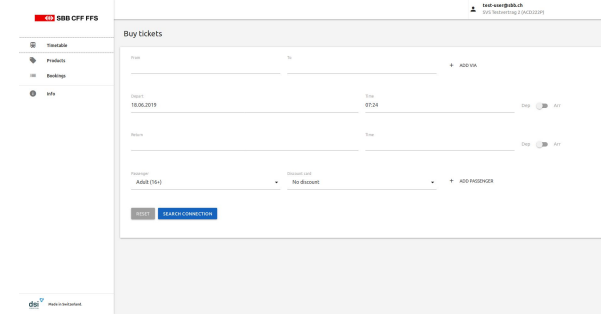
+ ADD PASSENGER

RESET

SEARCH CONNECTION

# Agenturclient

- Allows SBB business customers to sell tickets
  - SBB is contract partner and responsible for customer care
- These business customers usually are domestic and foreign travel agencies
- It allows to sell regular tickets, touristic offerings and super saver tickets
- Refunds are possible too







*How much did we earn building the software?*



*How much do we earn operating the software?*



*How much do we earn maintaining the software?*

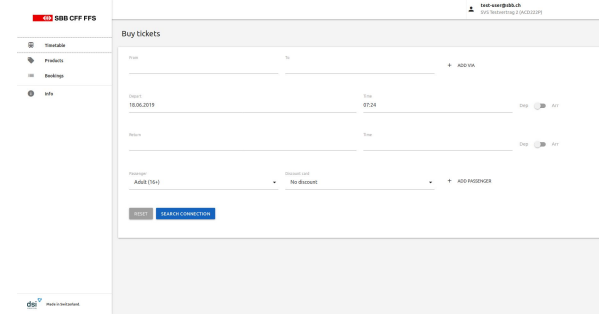


*How do we earn money?*



# Aligned Business Models

- We aligned our business models
  - SBB  
Gets income over Tickets sales via their platform
  - Travel Agencies  
Earn a commission when selling tickets
  - dsi engineering  
Charges a fee for every ticket sold



# Aligned Business Models

No discussions and contract negotiations over change requests

*instead*

business driven discussions about **return on investment**



# Aligned Business Models

No finger pointing or blaming during operation

*instead*

**mutual interest** in operating high quality software



# Aligned Business Models

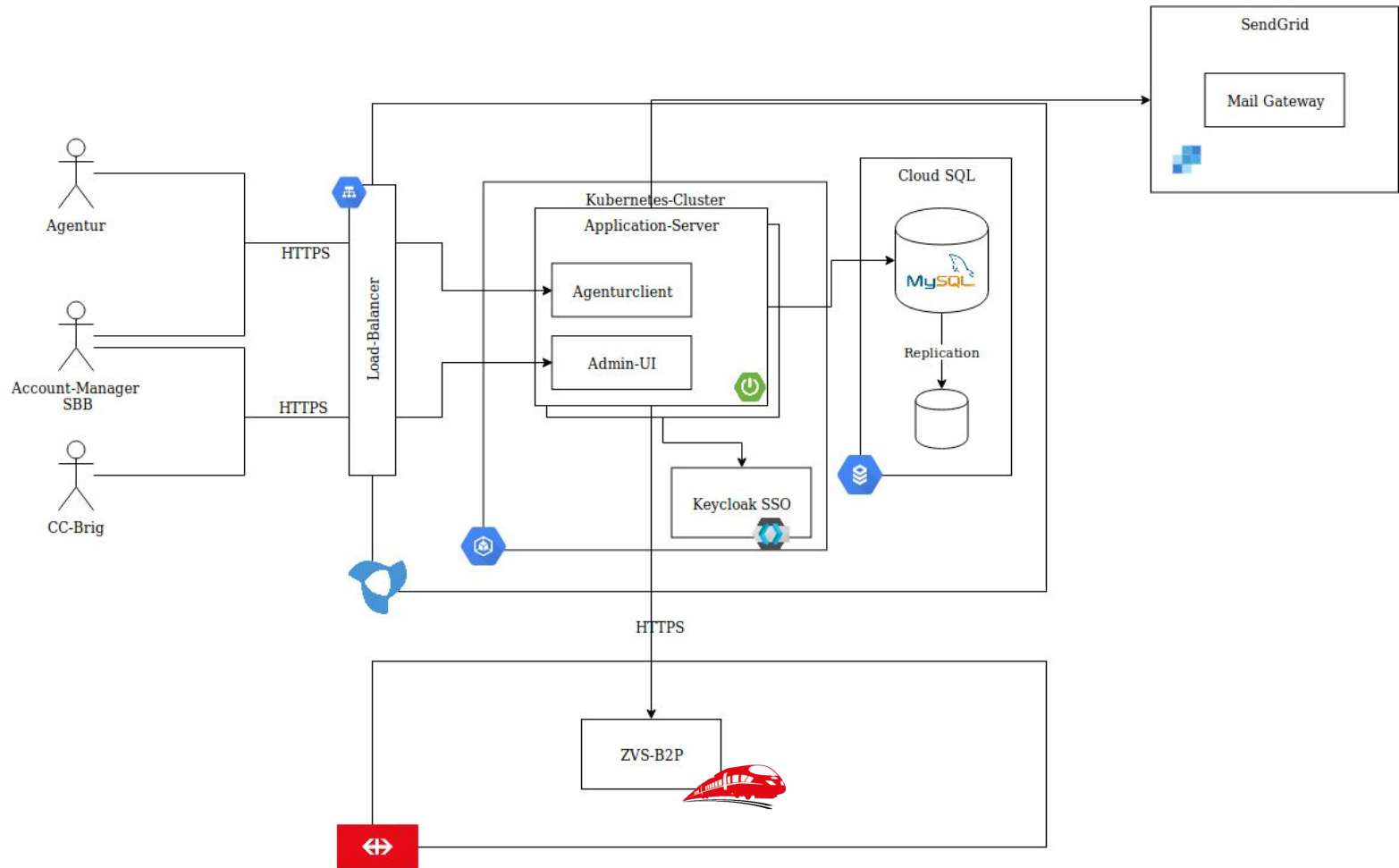
No development project and  
application lifecycle management from the customer  
*instead*  
everybody does what **they do best**





**Agenturclient**

**Architecture**



# Agenturclient - Technologies

- Vue frontend
  - Served directly from Spring Boot Backend
- Spring Boot MVC Application
  - Offers REST interface to frontend
  - Authorizes users using Keycloak groups
  - Stateless
- MySQL as storage backend



# Agenturclient - HA-Setup

- Our Spring Boot backend is completely stateless
  - We run multiple instances
- Accessing SBB's B2P service
  - We want to have control over timeouts
  - Automatic retries within timeout on network errors
  - Circuit breaking is disabled
- We use Hystrix to achieve this. Today, resilience4j would be the tool of choice.



# Agenturclient - Keycloak



- Open Source Identity and Access Management
  - Upstream of RedHat SSO
- Implements standard protocols
  - OpenID Connect, OAuth 2.0 and SAML 2.0
- Allows Central Management of Users, Roles and Groups
- Identity Brokering is possible
  - OpenID Connect or SAML 2.0 IdPs
- Clustering is supported
  - For scalability and availability



**Agenturclient**

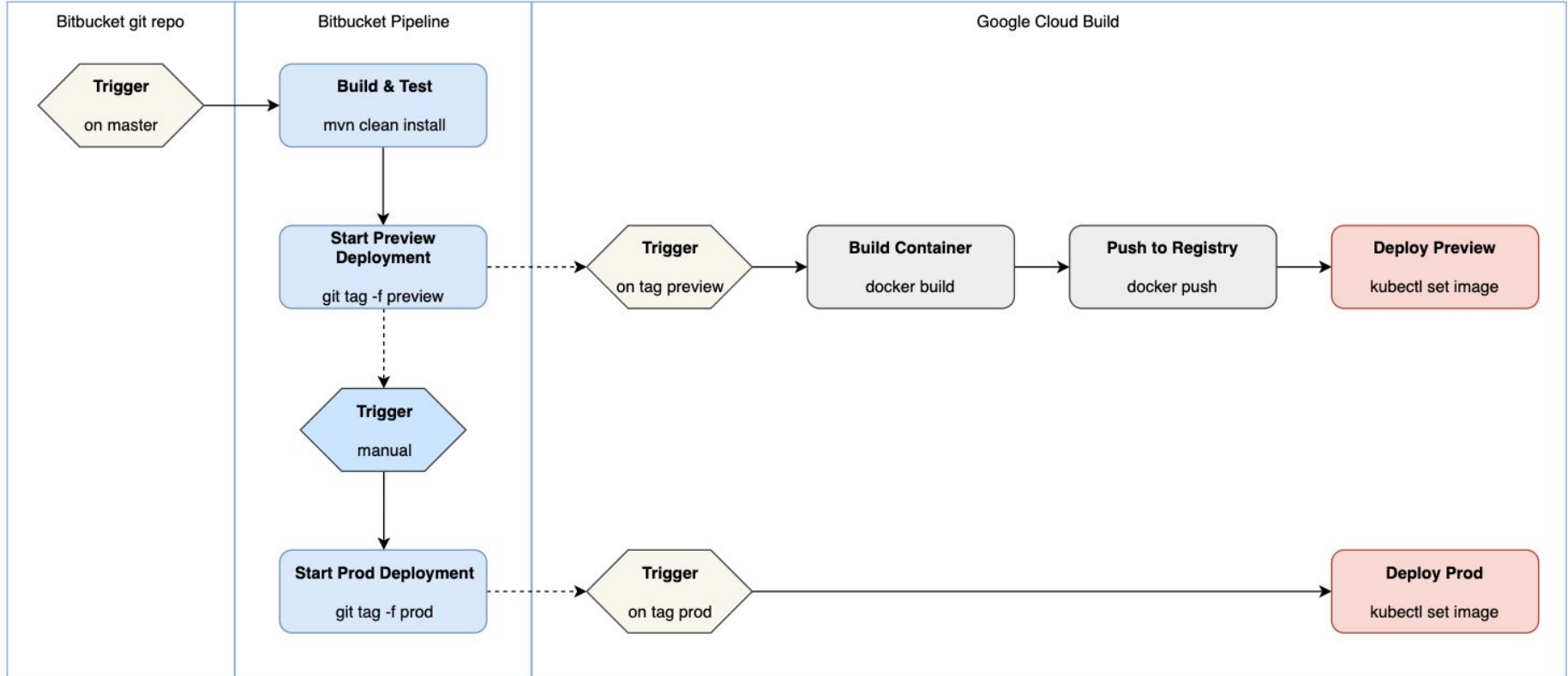
**Deployment**

# Deployment - Requirements

- We don't have our own infrastructure
  - We deploy to the cloud, from the cloud
  - No operation of own build server
- We want to be able to deploy to prod as frequently as we like
- Number of concurrent users is limited
- 98.3% availability is promised to clients
  - 15 minutes response time



# Agenturclient - Pipeline Overview





# Agenturclient - Pipeline

DSI AG / ... / Pipelines

✓ #155 Rerun

↓ df3a7f7 disable wallet ticket for flexpass  
↳ master

🕒 10 min 14 sec 📅 14 days ago RB

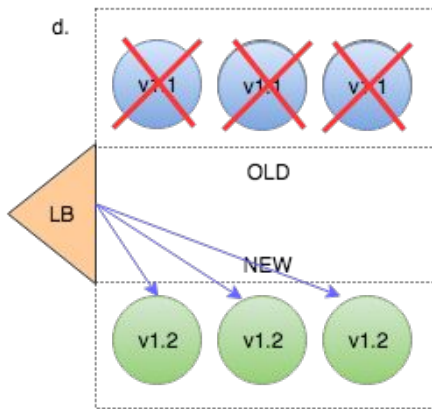
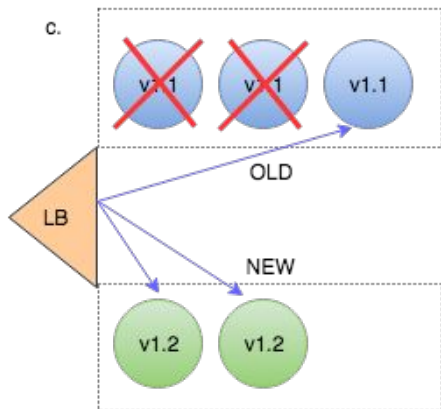
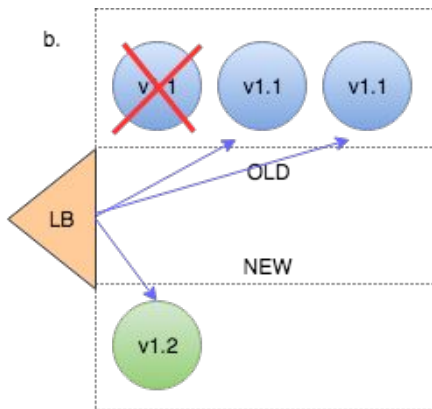
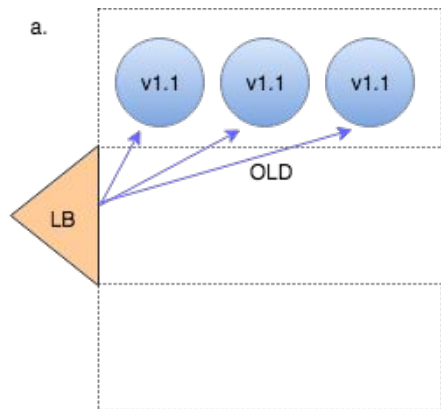
**Pipeline** ⚙️

- ✓ Build 75 tests passed • 3m 00s
- ✓ run zvs integrati... 106 tests passed • 🧑 • 6m 50s
- ✓ Deploy to test 9s
- ✓ Deploy to prod 🧑 • 14s

```
Build +  
  
Build setup  
  
echo "Creating Tag prod that triggers cloud build deploying the test container to prod environment"  
  
git tag -f prod  
  
git push --tags --force  
  
Build teardown
```



# Zero Downtime Deployments



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-dep
  namespace: default
spec:
  replicas: 2
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 25%
      maxSurge: 1
  template:
    spec:
      containers:
        - image: test
```



# Agenturclient - Graceful Shutdown

- We don't want to lose requests during shutdown.
- Therefore we wait on the internal Tomcat Thread Pool to finish all requests

```
@Override
public void onApplicationEvent(ContextClosedEvent event) {
    Log.info("Received SIGTERM. Shutdown Gracefully.");

    if (this.connector != null) {
        this.connector.pause();
        Executor executor = this.connector.getProtocolHandler().getExecutor();
        if (executor instanceof ThreadPoolExecutor) {
            try {
                ThreadPoolExecutor threadPoolExecutor = (ThreadPoolExecutor) executor;
                threadPoolExecutor.shutdown();
                if (!threadPoolExecutor.awaitTermination(TIMEOUT, TimeUnit.SECONDS)) {
                    Log.warn("Tomcat thread pool did not shut down gracefully within "
                        + TIMEOUT + " seconds. Proceeding with forceful shutdown");

                    threadPoolExecutor.shutdownNow();

                    if (!threadPoolExecutor.awaitTermination(TIMEOUT, TimeUnit.SECONDS)) {
                        Log.error("Tomcat thread pool did not terminate");
                    }
                }
            } catch (InterruptedException ex) {
                Thread.currentThread().interrupt();
            }
        }
    }
}
```



# Agenturclient - Keycloak



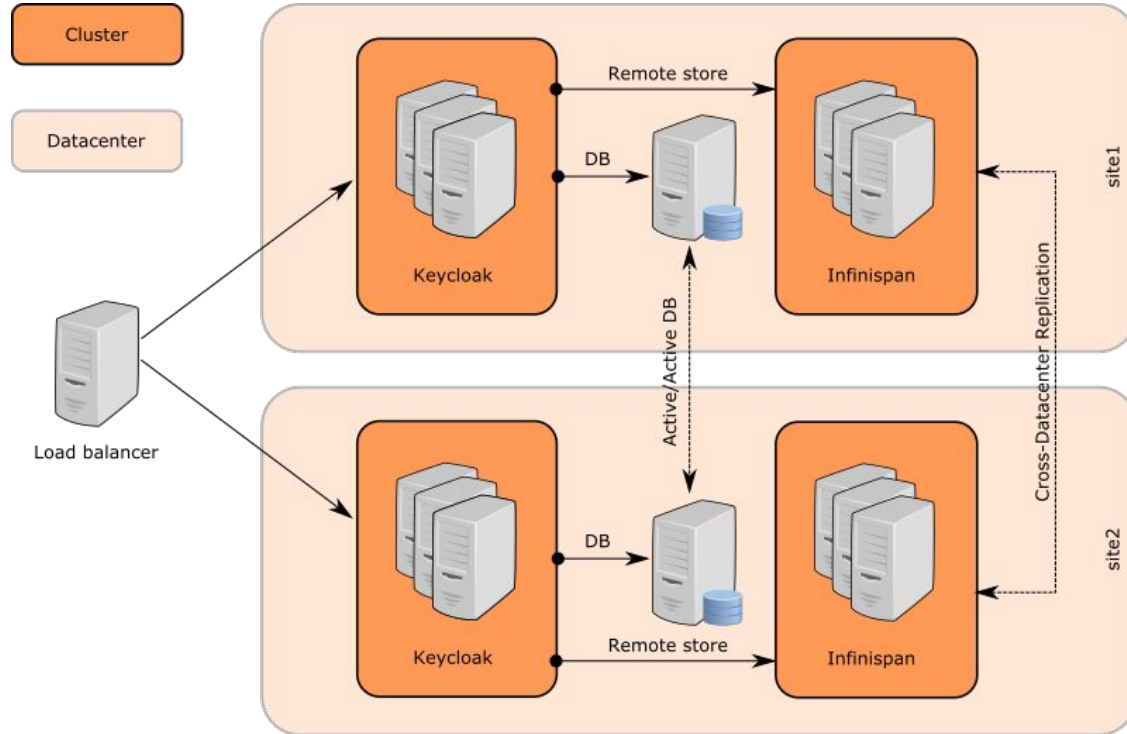
- Agenturclient has about 3500 concurrent users
  - Users can be managed by the travel agencies
  - Change rate is low
- They usually login in the morning and stay logged in the whole day
- Therefore we don't have high performance requirements
  - Scalability not needed
- Cluster is still needed
  - In order to achieve a highly available setup



**Keycloak**

**High Availability**

# Agenturclient - Keycloak HA-Setup



# Agenturclient - Keycloak HA-Setup



- Keycloak HA requires an Infinispan In-Memory Grid to store sessions and users
  - Users and sessions are stored in an Infinispan Replicated Cache
- Infinispan uses JGroups for networking in Clustered-Mode
- JGroups
  - Requires a discovery mechanism to discover all cluster nodes
  - Establishes IP-Multicast between the cluster nodes



# Agenturclient - Keycloak@K8s



- Service discovery done via Kubernetes Services
- Two services are created
  - Cluster-IP service to access keycloak nodes
  - Headless service allows discovery of all cluster nodes

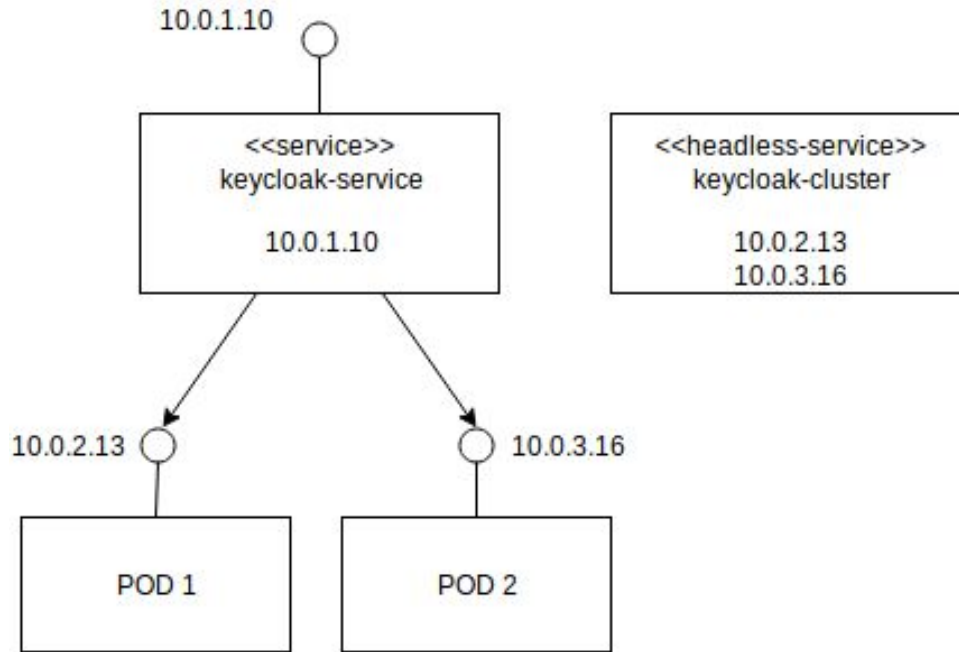
```
JGROUPS_DISCOVERY_PROTOCOL="dns.DNS_PING"
```

```
JGROUPS_DISCOVERY_PROPERTIES="dns_query=keycloak-cluster.default.svc.cluster.local"
```



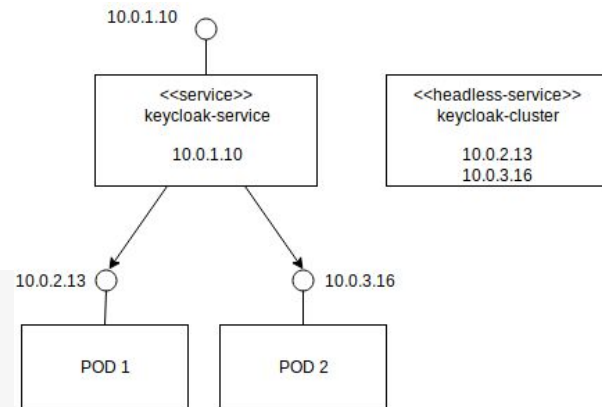


# Agenturclient - Keycloak@K8s



# Agenturclient - Keycloak@K8s

```
/# dig keycloak-service.default.svc.cluster.local  
  
; <<>> DiG 9.11.3-1ubuntu1.3-Ubuntu <<>>  
keycloak-service.default.svc.cluster.local  
;; QUESTION SECTION:  
;keycloak-service.default.svc.cluster.local. IN A  
  
;; ANSWER SECTION:  
keycloak-service.default.svc.cluster.local. 30 IN A 10.0.1.10
```



# Agenturclient - Keycloak@K8s

```
/# dig keycloak-cluster.default.svc.cluster.local
```

```
; <<>> DiG 9.11.3-1ubuntu1.3-Ubuntu <<>>
```

```
keycloak-cluster.default.svc.cluster.local
```

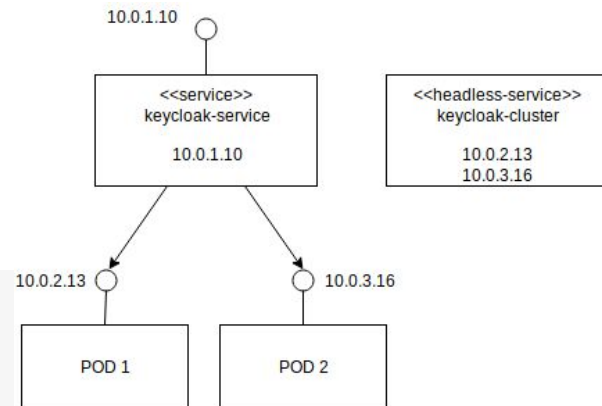
```
;; QUESTION SECTION:
```

```
;keycloak-cluster.default.svc.cluster.local. IN A
```

```
;; ANSWER SECTION:
```

```
keycloak-cluster.default.svc.cluster.local. 30 IN A 10.0.2.13
```

```
keycloak-cluster.default.svc.cluster.local. 30 IN A 10.0.3.16
```



# Agenturclient - Keycloak@Cloud



- JGroups IP-Multicast is not supported in public clouds
- Switch transport stack to TCP
- We updated Keycloak Docker-Files to
  - have TCP as standard
  - Allow reconfiguration using environment variables

 **KEYCLOAK-10198 KEYCLOAK-10199: Allow JGroups transport stack configuration**  16

**Missing JIRA**

#159 by fluescher was merged 5 days ago

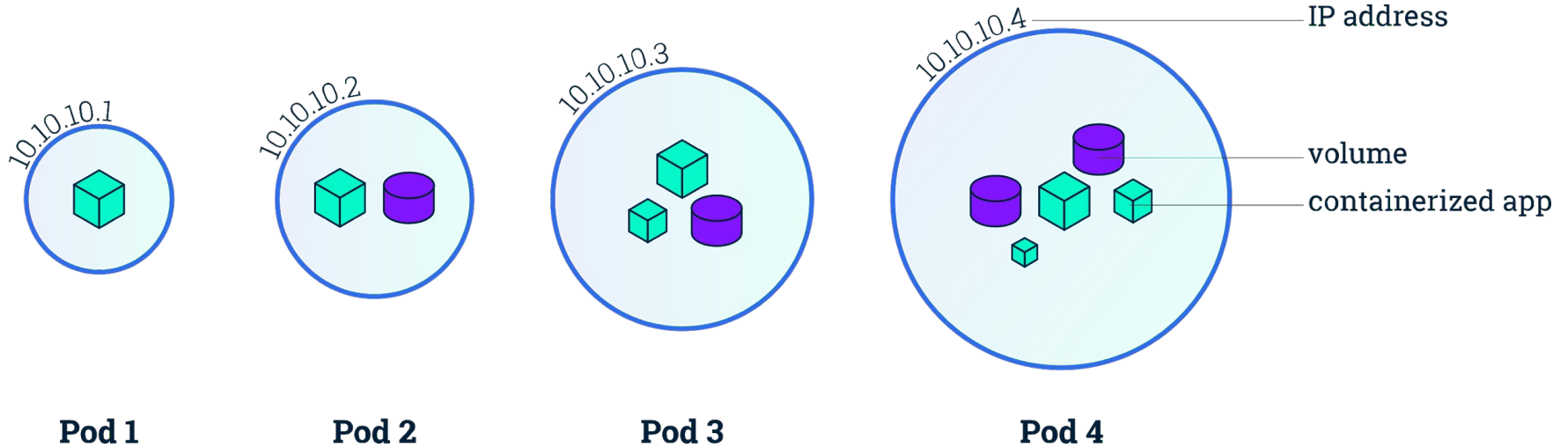


Google Cloud



# Pod

A Pod can host multiple containers.



# Pod

These containers share:

- **Network**

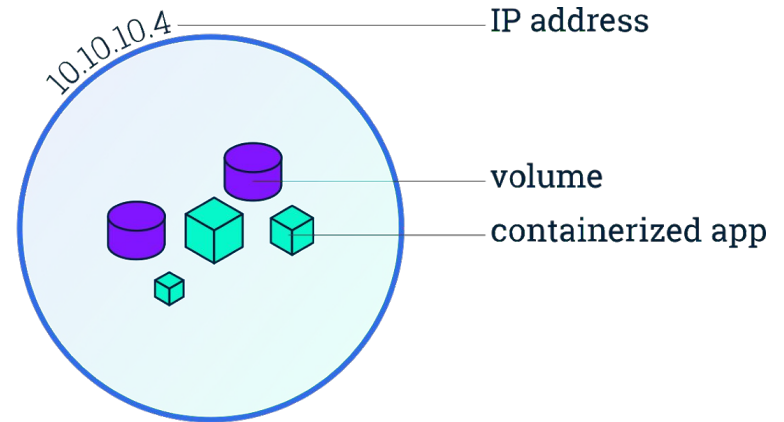
Same unique IP Address

Same Port Range

Can communicate using `localhost`

- **Storage**

Volumes can be accessed by all containers



**DEMO**

# Are we really highly available now?

- What happens if a node goes down?
- What happens if cluster maintenance requires to take a node offline?





# Node Affinity

We want to tolerate the outage of a node.

Kubernetes offers **affinity rules** to decide which nodes are eligible for a pod to be scheduled upon.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: agenturclient-deployment
  labels:
    app: agenturclient
spec:
  replicas: 2
  template:
    spec:
      affinity:
        podAntiAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            - labelSelector:
                matchExpressions:
                  - key: "app"
                    operator: In
                    values:
                      - agenturclient
              topologyKey: "kubernetes.io/hostname"
```



# Disruption Budget

We don't want to take all nodes offline when upgrading the cluster.

A **PodDisruptionBudget** controls how many Pods of a deployment should be available during regular maintenance.

A node is not evicted if it would violate a Pod disruption budget.

```
apiVersion: policy/v1beta1
kind: PodDisruptionBudget
metadata:
  name: agenturclient-pdb
spec:
  minAvailable: 1
  selector:
    matchLabels:
      app: agenturclient
```



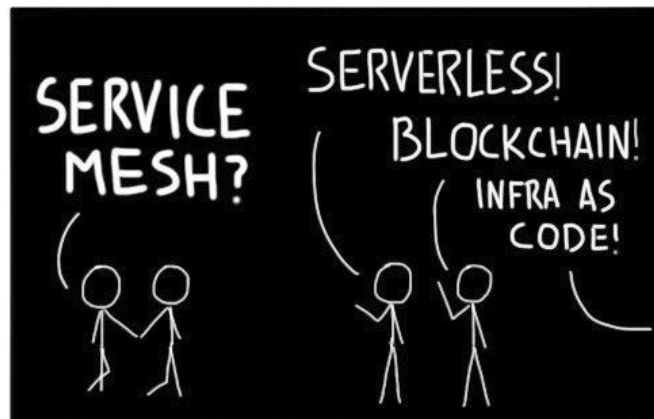
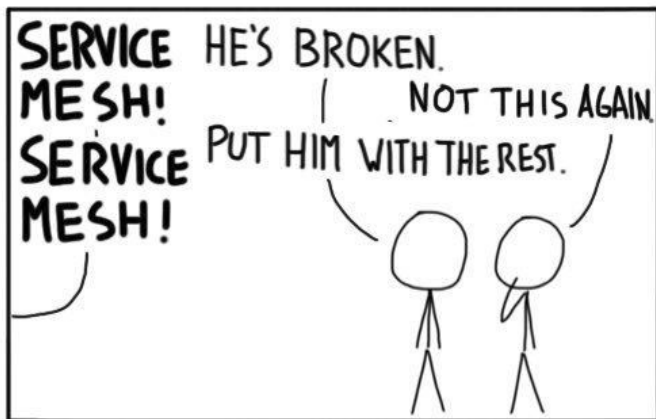
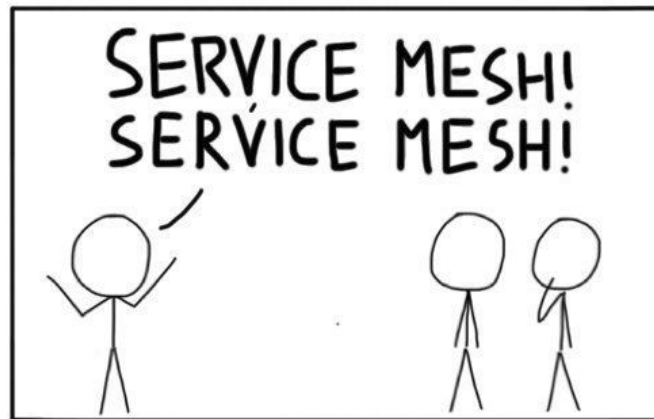
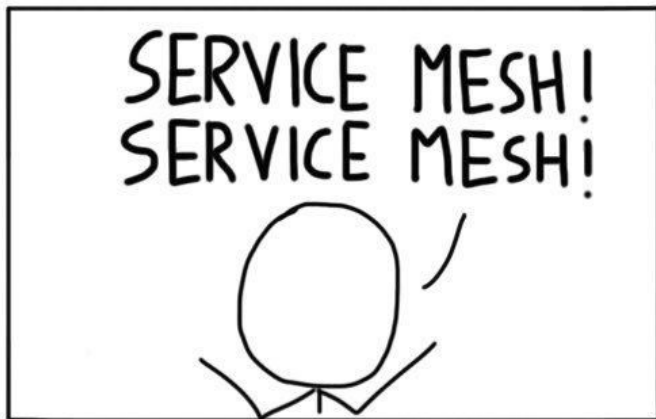
# Agenturclient @ Google Kubernetes Engine

What's missing?

- More insights into network traffic
  - Between our services and to external systems
- Traffic encryption everywhere, by default
  - Because we run in a public cloud
- Policy enforcement
  - Restrict unallowed network traffic



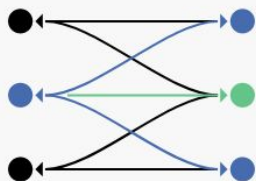
# Istio Service Mesh



@sebiwicb



# Istio Overview



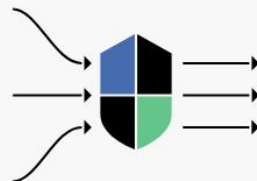
## Connect

Intelligently control the flow of traffic and API calls between services, conduct a range of tests, and upgrade gradually with red/black deployments.



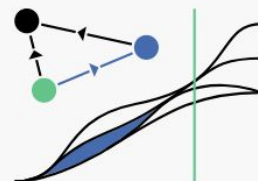
## Secure

Automatically secure your services through managed authentication, authorization, and encryption of communication between services.



## Control

Apply policies and ensure that they're enforced, and that resources are fairly distributed among consumers.

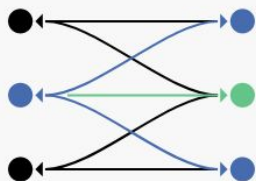


## Observe

See what's happening with rich automatic tracing, monitoring, and logging of all your services.



# Istio Overview



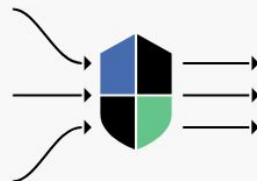
## Connect

Intelligently control the flow of traffic and API calls between services, conduct a range of tests, and upgrade gradually with red/black deployments.



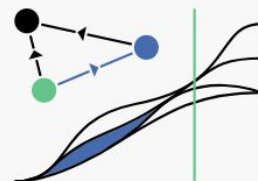
## Secure

Automatically secure your services through managed authentication, authorization, and encryption of communication between services.



## Control

Apply policies and ensure that they're enforced, and that resources are fairly distributed among consumers.

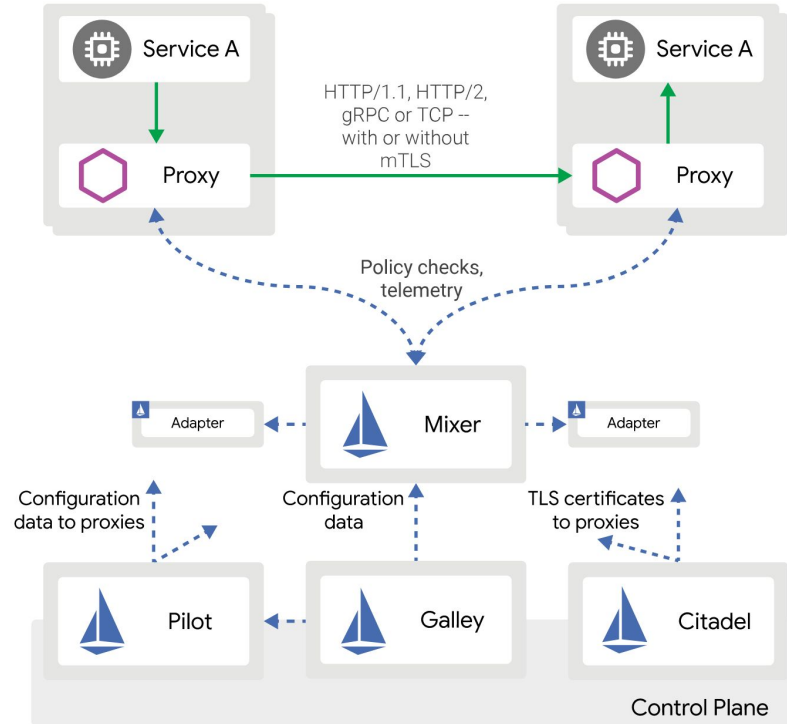


## Observe

See what's happening with rich automatic tracing, monitoring, and logging of all your services.



# Istio Overview





**DEMO**

# Istio Overview - Envoy



*Envoy is an open source edge and service proxy, designed for cloud-native applications.*

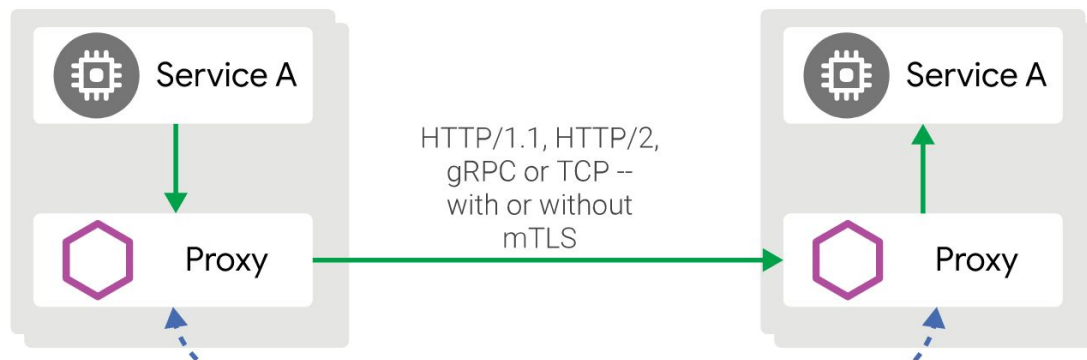


# Istio Overview - Envoy



Sits between every network connection. This allows for:

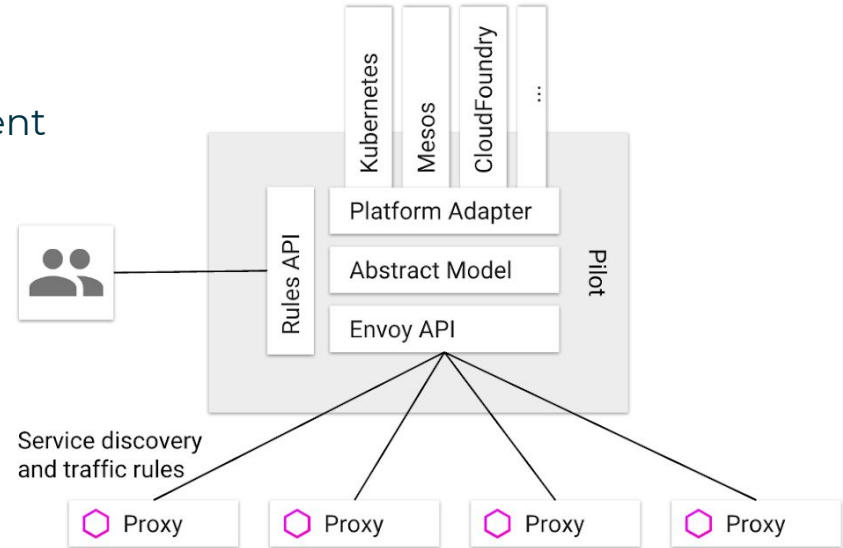
- Circuit Breaking
- Retries
- Logging, Tracing & Monitoring
- Policy Enforcement
- Fault Injection
- Traffic Routing
- mTLS



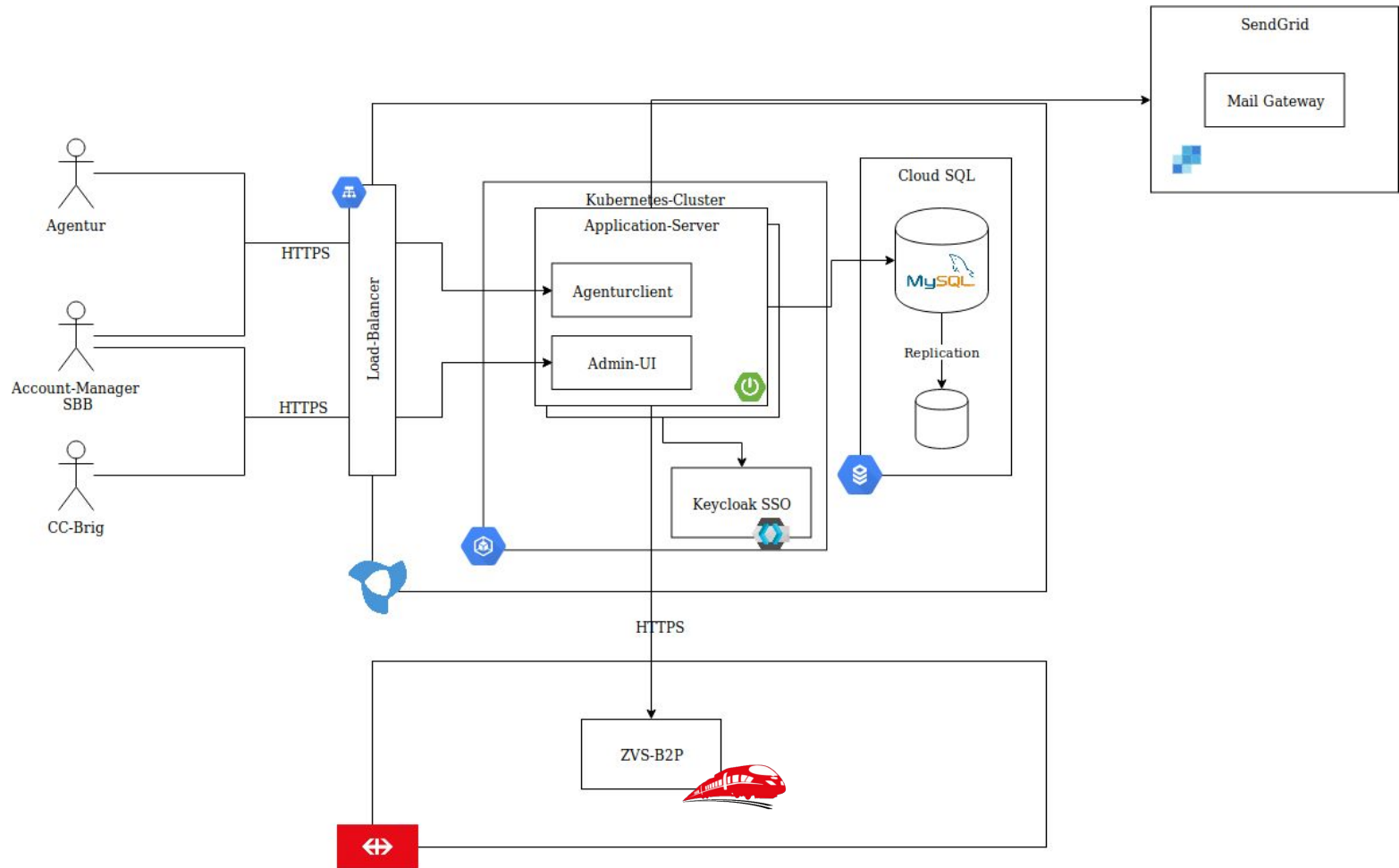
# Istio Overview - Pilot

Pilot configures all the envoy sidecar proxies.

It uses metadata it receives from the environment it runs in (e.g. Kubernetes).



**DEMO**



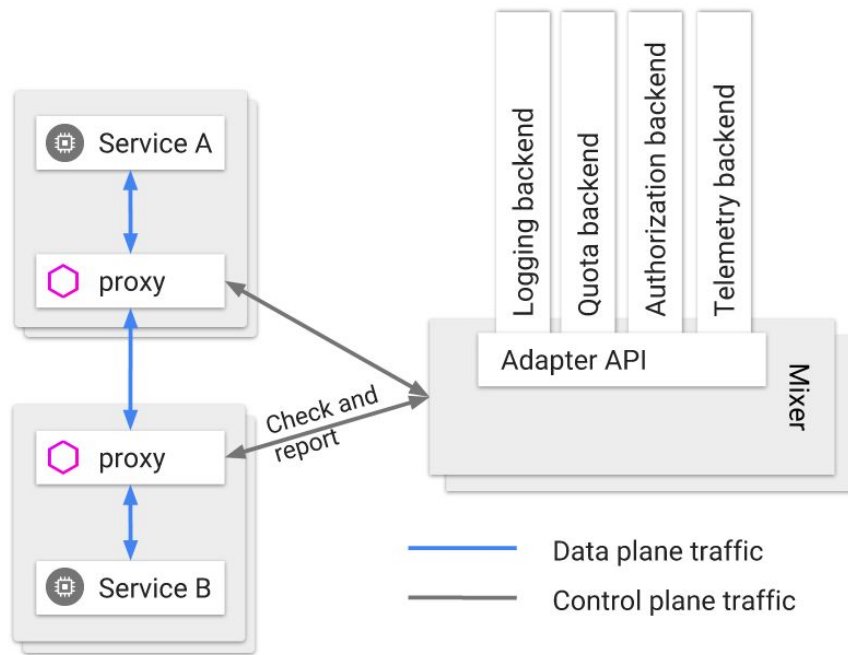
# Istio Overview - Mixer

Mixer is responsible for

- Checking request against policy rules
- Forward telemetry and logging to Backends

To avoid a single point of failure the sidecar proxies cache policies and mixer itself serves as a cache in front of backend systems.

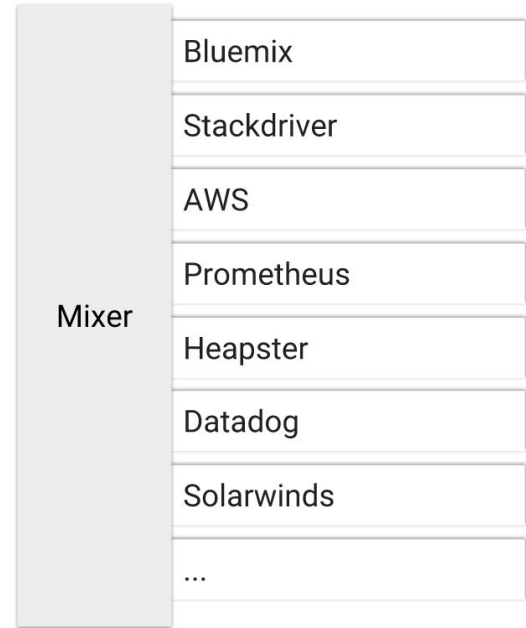
Mixer provides several adapters to monitoring systems.



# Istio Overview - Mixer

Mixer provides several adapters to monitoring systems.

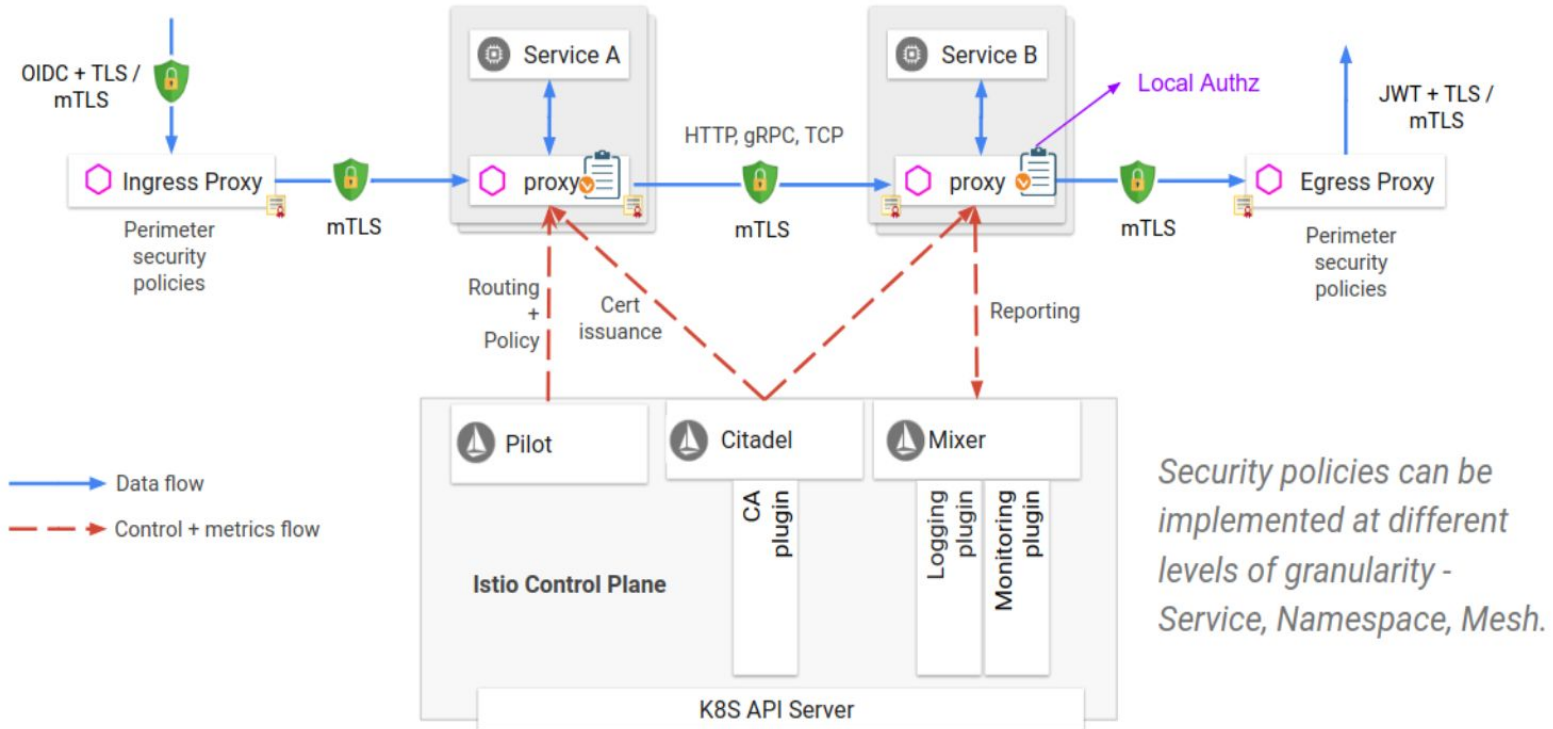
It is possible to use these adapters to add more attributes to requests. These attributes can be used in expressions to specify different rules.





**DEMO**

# Istio Overview - Citadel



# Istio Overview - Citadel

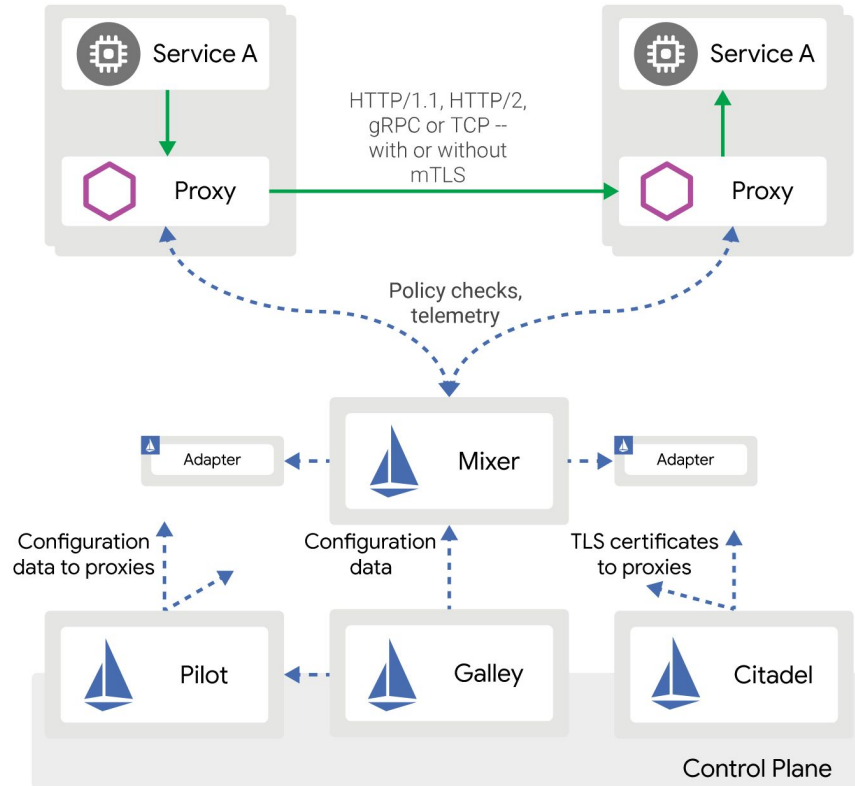
Citadel uses SPIFFE (<https://spiffe.io>) to issue certificates.

On Kubernetes:

1. Citadel watches the Kubernetes apiserver. Every Service Account gets a X509 cert
2. When a Pod is started, the certificate information is mounted
3. Citadel rotates these certificates regularly
4. Pilot configured the envoy proxies



# Istio Overview



# Conclusions

# Learnings - Agenturclient @ Google Kubernetes Engine

- GKE is very easy to use and very reliable
- Cluster Upgrades are smooth and we have zero downtime
  - We use a regional cluster so one master is always responding
- Google Cloud SQL runs without any issue for a year now
- Bitbucket Pipelines and Google Cloud Build allow an “infrastructure-less” build pipeline



# Learnings - Agenturclient @ Google Kubernetes Engine



# Learnings - Agenturclient @ Google Kubernetes Engine

- Google Regions might be booked out
  - To save cost, we use preemptible instances on our preview environment
  - We hit that issue in europe-west3 (Frankfurt) several times
  - Created an autoscaler on our preview cluster to spin up more nodes in case this happens

The zone 'europe-west3-c' does not have enough resources available to fulfill the request. Try a different zone, or try again later.





# Istio - The good

- Istio can be a unified solution for access management and logging
  - Technology independent (JVM, .NET, etc.)
- Istio does not require any updates to your software
  - It is suitable for bought software as well (Keycloak)
- Istio is widely backed by industry leaders
- Istio is orchestrator independent
  - Kubernetes / Mesos / On-Premise
- Mixer integrates with a lot of different monitoring tools

Enable Istio (beta) ?

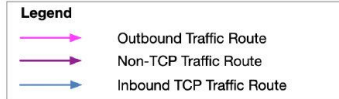
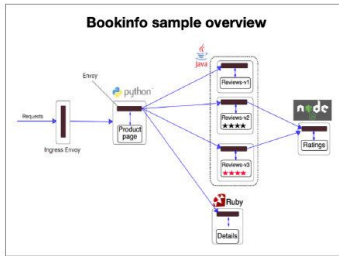


# Istio - The Bad

How Envoy Proxy Working As Sidecar Proxies to Intercept and Route Traffic in Istio Service Mesh



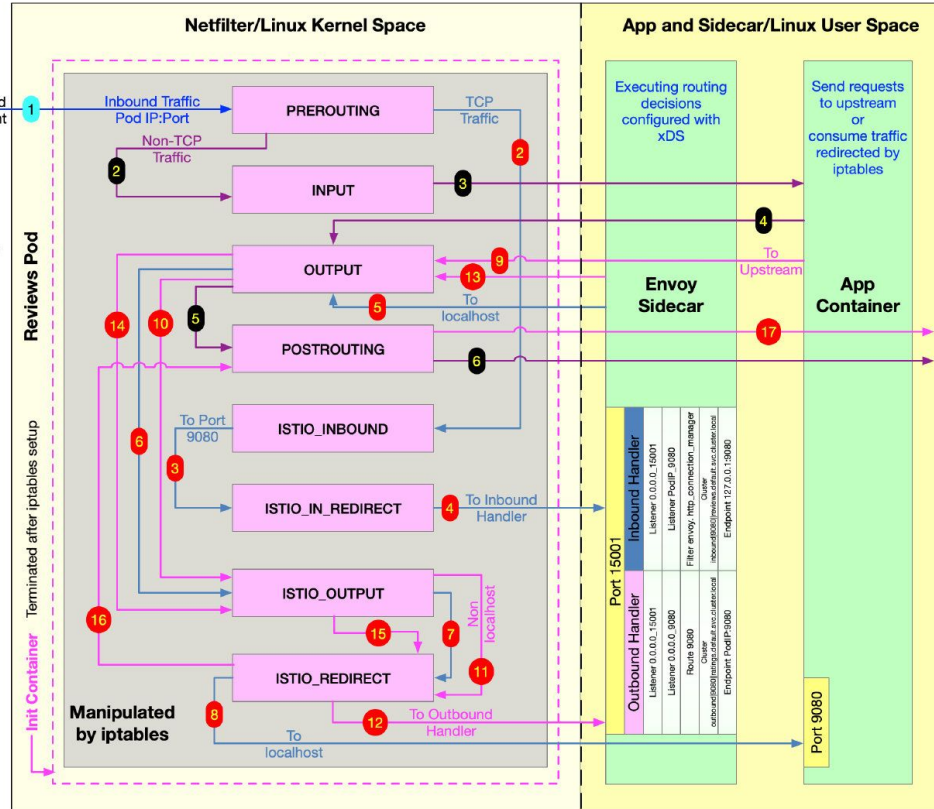
Productpage service send requests to `http://reviews.default.svc.cluster.local:9080/`



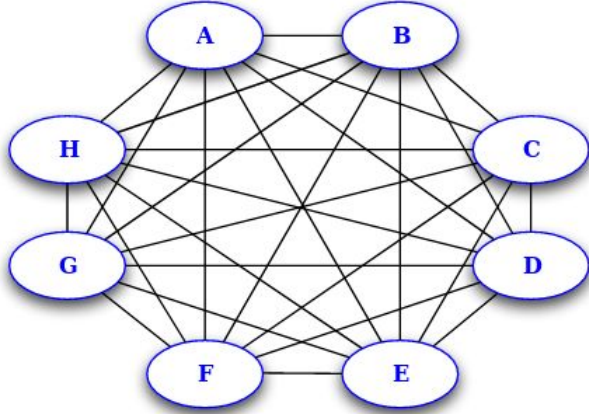
This sample based on the bookinfo sample scenario, please refer to <https://istio.io/docs/examples/bookinfo/> for details.

Source <https://jimmysong.io>

Version Dec 27, 2018  
by Jimmy Song

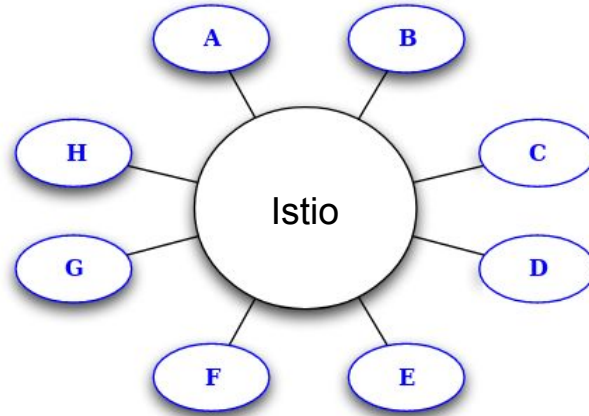
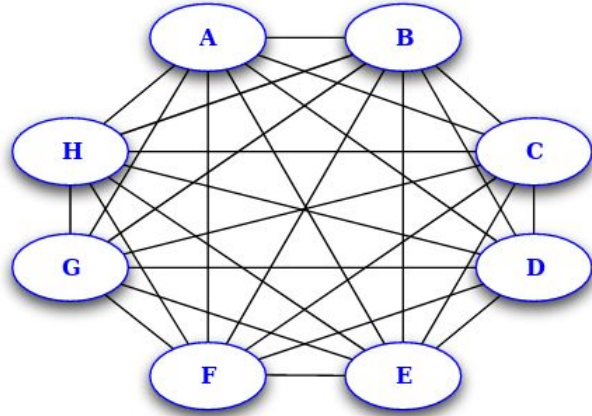


# Istio - an intelligent middle man



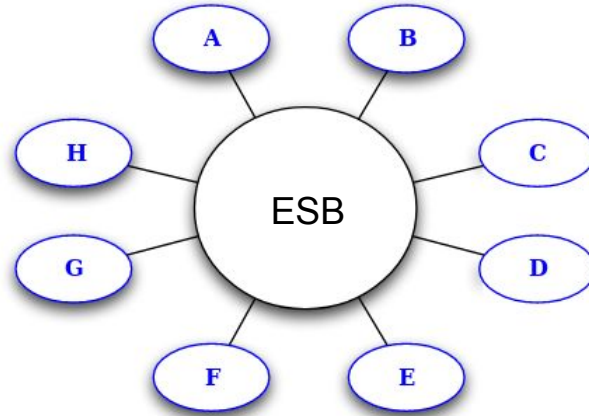
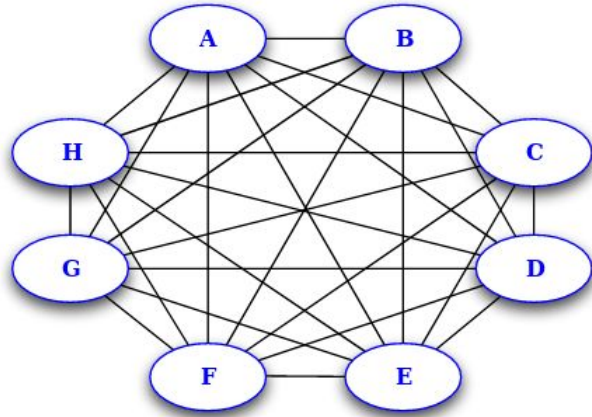
# Istio - an intelligent middle man

What about “smart endpoints - dumb pipes”?



# Istio - an intelligent middle man

What about “smart endpoints - dumb pipes”?



Feedback  
Welcome!



<https://forms.gle/XU4dj9DNAHkLMspYA>

