

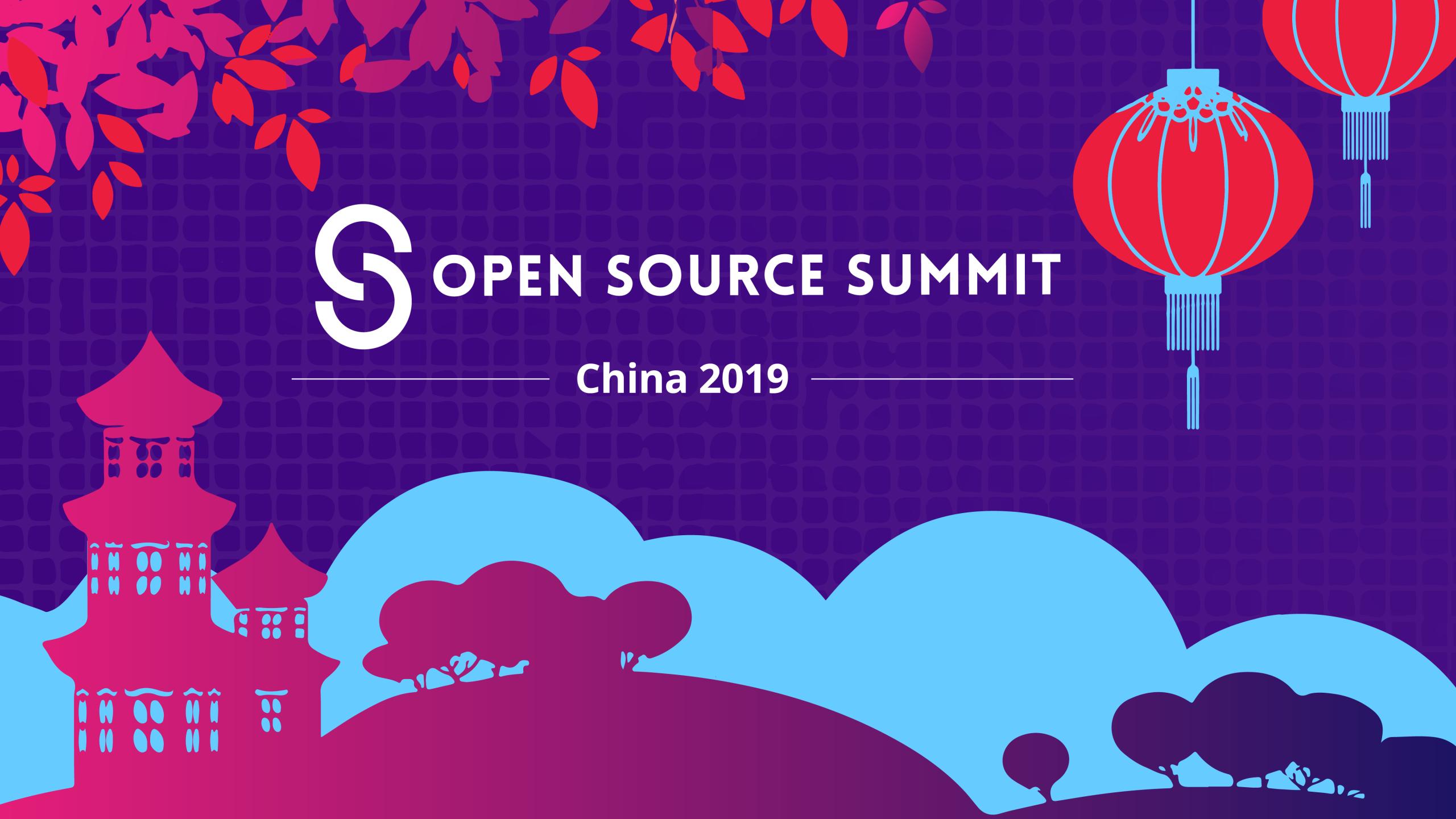


# OPEN SOURCE SUMMIT

---

China 2019

---



# Service Governance in Production-ready containerized Cloud Foundry with Istio



Xuexiang Cui (Matt) [cuixuex@cn.ibm.com](mailto:cuixuex@cn.ibm.com)  
Gong Zhang (Grace) [zhanggbj@cn.ibm.com](mailto:zhanggbj@cn.ibm.com)

# Speakers



**Xue Xiang Cui**, cuixuex@cn.ibm.com

Xue Xiang Cui (Matt) is a Senior Software Engineer at IBM. He is the leader of IBM Cloud BOSH team responsible for the development of BOSH stemcell and CPI for IBM Cloud Infrastructure, and also plays a key role on the development of IBM Cloud Foundry Enterprise Environment (CF on Kubernetes) project.



**Gong Zhang**, zhanggbj@cn.ibm.com

Gong Zhang (Grace) is an Advisory Software Engineer with 6+ years work experience on Cloud Computing. She's a contributor of BOSH Softlayer CPI, Kubernetes CPI and containerized Cloud Foundry solution. Currently, she is a developer working on IBM Cloud Foundry Enterprise Environment and also Istio integration.

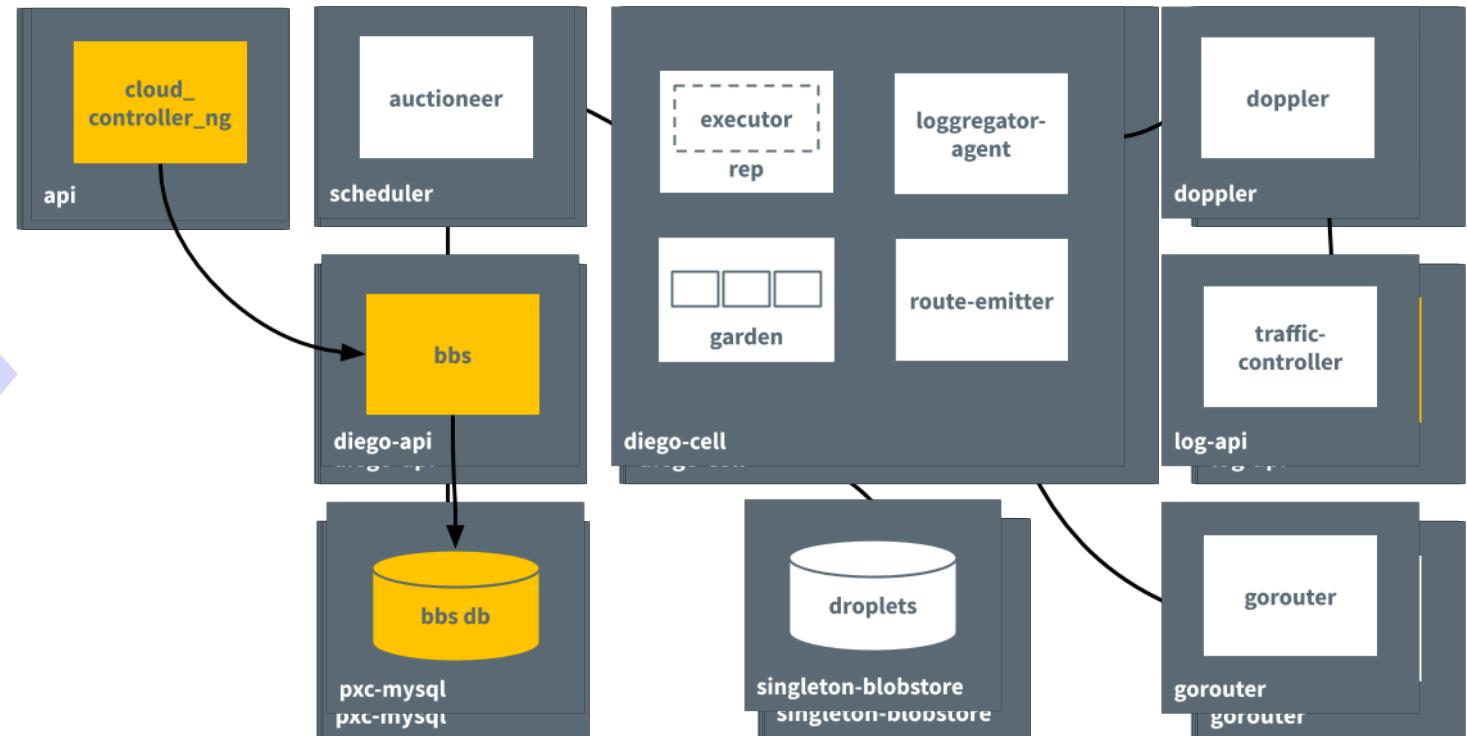
# Agenda

- Project Background
- Istio Introduction
- Motivations for CFEE and Istio Integration
- High-level Architecture and Implementation
- Future Vision
- Use Cases and Demo

# Project Background – Cloud Foundry

**Cloud Foundry** is the industry-standard **open source cloud application platform** for developing and deploying enterprise cloud applications.

```
cf api https://api.<domain>
cf login -u admin -p password
cf push <app>
cf logs <app>
cf ssh <app>
```



# Project Background – CFEE (Cloud Foundry Enterprise Environment)

**Cloud Foundry Enterprise Environment:** an isolated environment for hosting your Cloud Foundry apps with full admin control over configuration, capacity and access. Production-ready release 3.1.0.

## Core Features

- Runs on IBM Cloud container Service (IKS)
- Isolated Cloud Foundry environment
- Full administrative control
- Rapid provisioning and user onboarding
- Eirini (Technical Preview)

The screenshot shows the IBM Cloud interface for configuring a Cloud Foundry Enterprise Environment. The main page title is "Configure Cloud Foundry Enterprise Environment". On the left, there's a sidebar with "Permissions" and a "Plan" section. The "Plan" section compares "Eirini (Technical preview)" and "Standard". The "Standard" plan is selected, described as "Explore the essential capabilities of Cloud Foundry running in full isolation, choosing among different data centers to meet your isolation and data resiliency needs." Below the plan section is a "Details" section with fields for "Name" (set to "default"), "Resource Group" (set to "default"), "Geography" (set to "North America"), "Region" (set to "Dallas"), "Worker Zones" (with options for Dallas 10, Dallas 12, and Dallas 13), "Private VLAN" (with options for Dallas 10 (dal10), Dallas 12 (dal12), and Dallas 13 (dal13)), and "Public VLAN" (with options for Dallas 10 (dal10), Dallas 12 (dal12), and Dallas 13 (dal13)). To the right of the main configuration area are two panels: "Security" and "Capacity". The "Security" panel includes sections for "Hardware Isolation" (with options for "Virtual - Shared" and "Virtual - Dedicated", where "Virtual - Dedicated" is checked) and "Encrypt local disk". The "Capacity" panel includes sections for "Number of cells" (set to 2), "Cell nodes" (set to 2 x 1 zone = 2), "Control plane nodes" (set to 2), and "Total nodes" (set to 4). It also includes a "Node size" section for "4 Cores 16GB RAM b2c4x16" and a "Version" section for "Version: 3.0.2" (Cloud Foundry Deployment: 3.6.0).

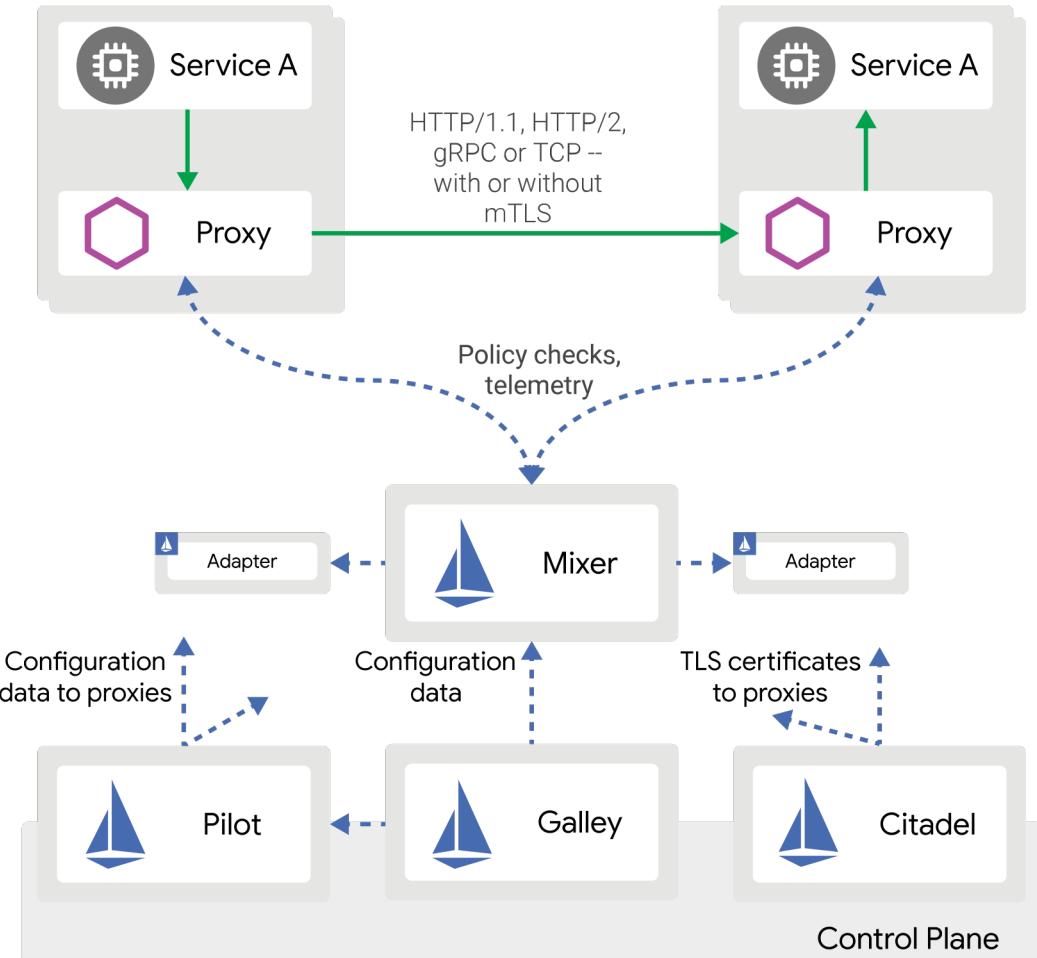
# Istio Introduction

## Istio

- lets you connect, secure, control, and observe services.
- Heavy investments from industry-leading companies such as Google, IBM, and Lyft.
- Production-ready Release 1.1.
- Active community.

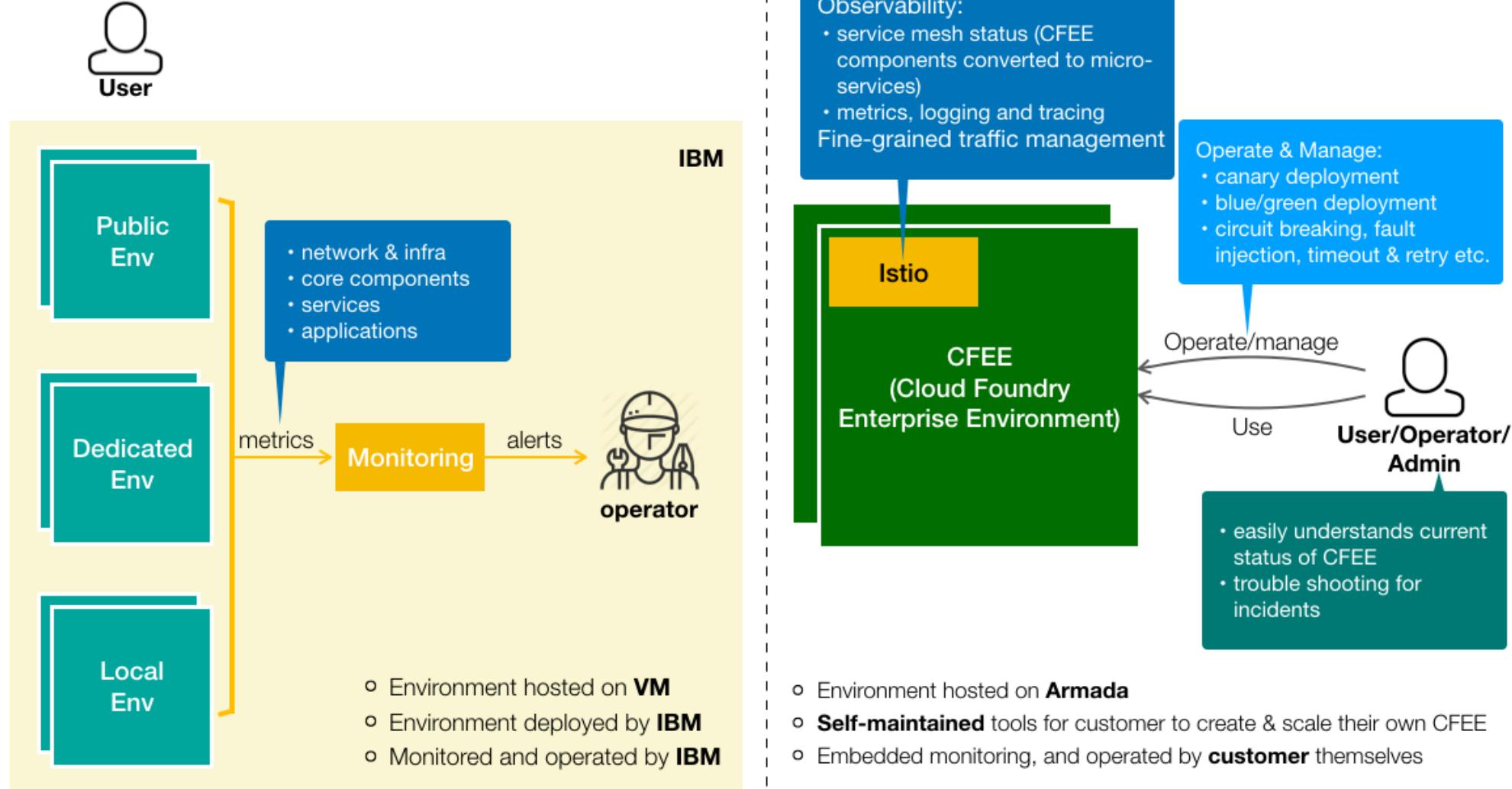
## Core Features

- Observability
- Fine-grained traffic management
- Security
- Platform support
- Integration and customization

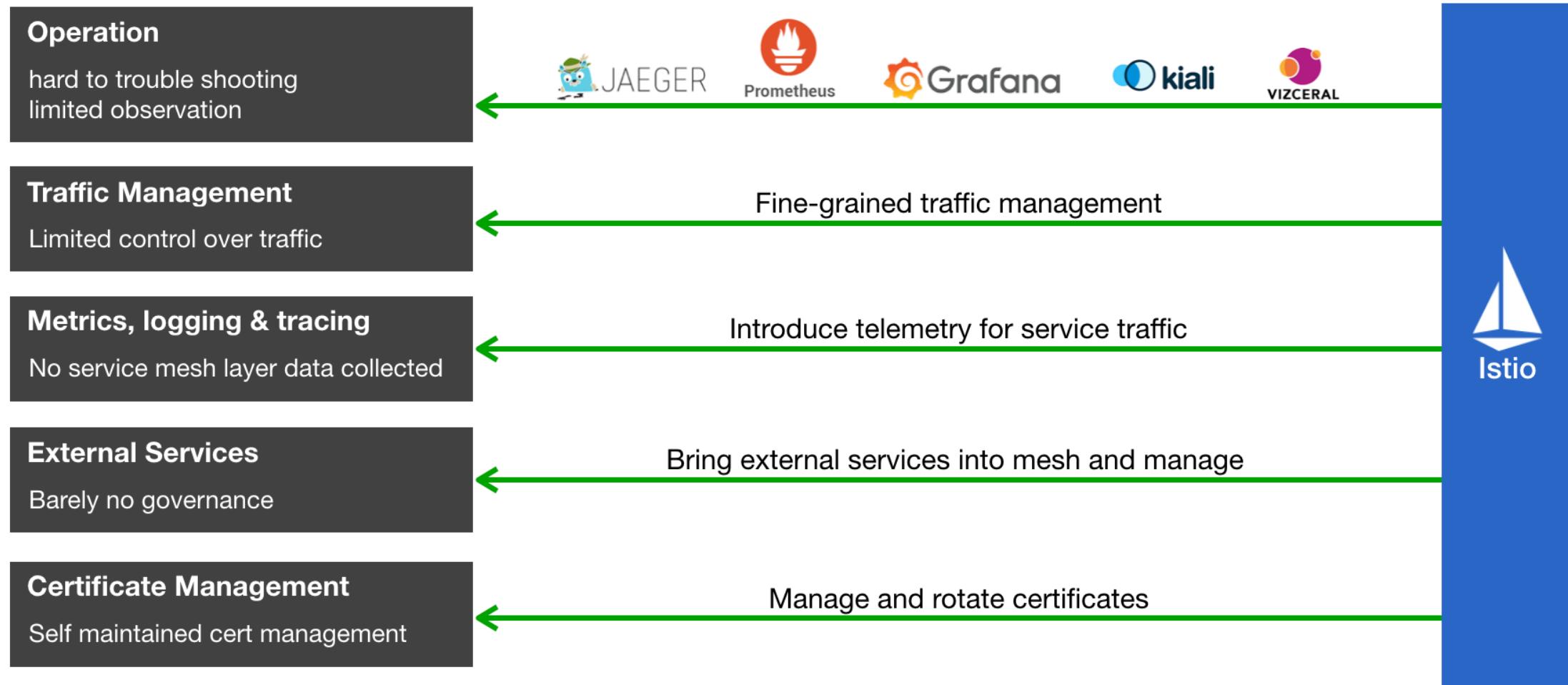


Istio Architecture from [istio.io](https://istio.io)

# Motivations for CFEE and Istio Integration



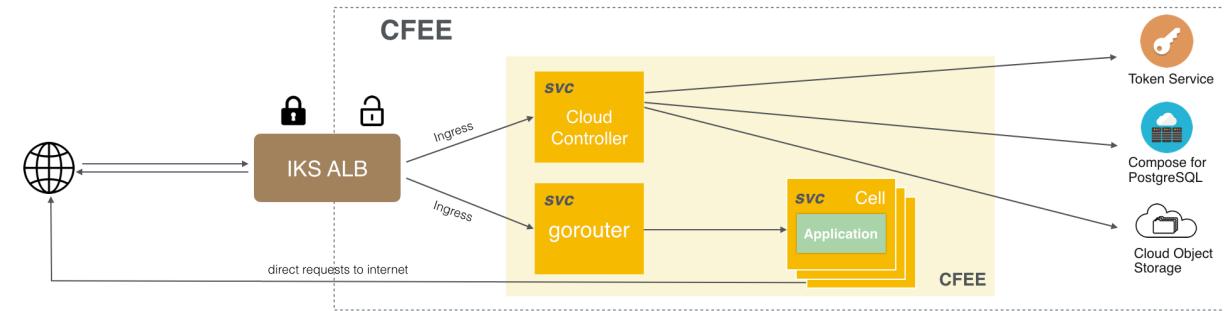
# Motivations for CFEE and Istio Integration



# High-level Architecture and Implementation

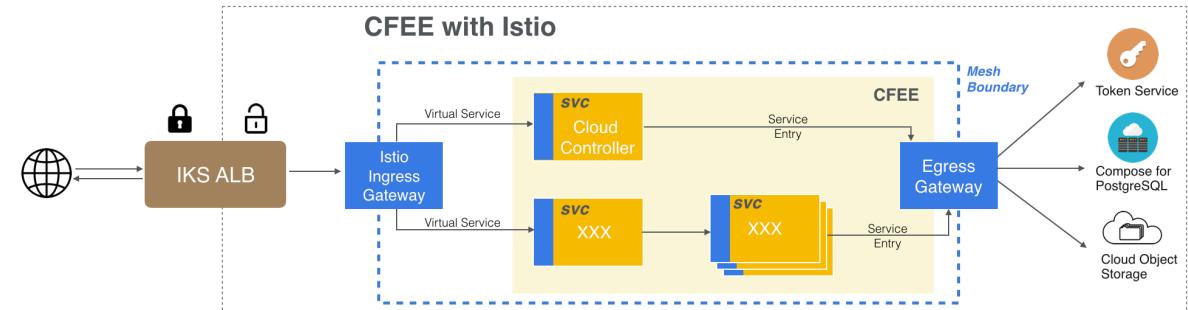
## Difficulties to bring Istio Into CFEE

- Containerized traditional CF
- Istio conventions required
- Complicated CF traffic
- Based on IKS platform
- Code change required to fully-enable tracing



## How to resolve them

- Move to native micro-service architecture
- Refactor and comply
- Refine CF traffic
- Selectively integrate with IKS
- Enable tracing by component



## Future Vision

- Contribute to open source containerized CF community leading by SUSE & IBM (in progress)
- Bring Istio into Eirini who Integrate Application Runtime and Kubernetes
- Leverage Istio mesh expansion to bring operational experience back to VM-based CF

# Use Cases and Demo

**Use Case – Get Operational Insights:** as an operator/administrator, I'd like to:

Get insights of the overview of CFEE status and observe how CF components collaborate on a request with complex behaviors.

With visualization of the service mesh topology, observe Istio behaviors like circuit breakers or request rates and traffic shifting on CFEE etc.

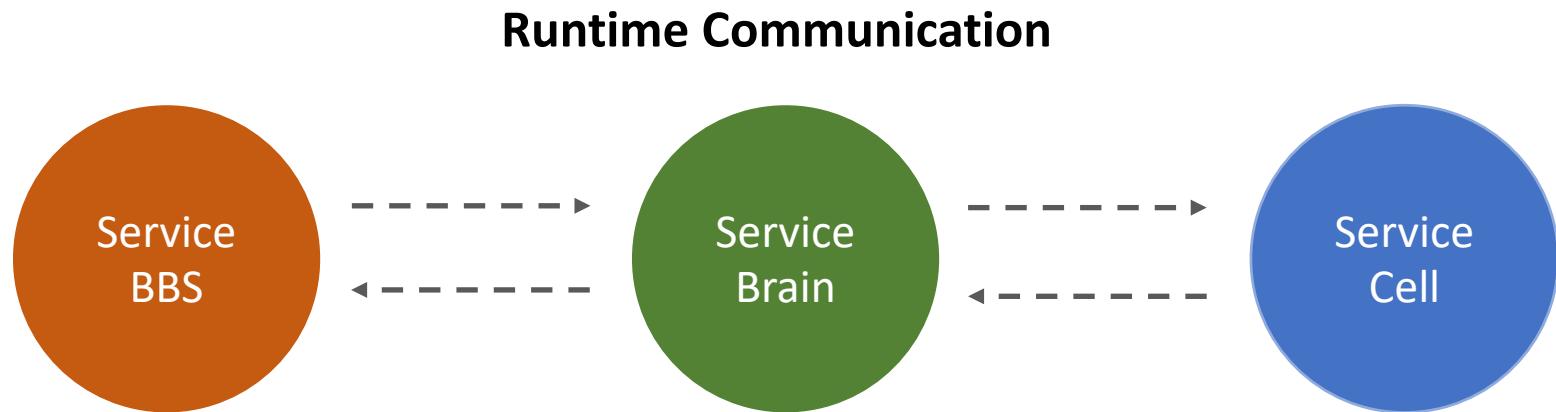


# Use Cases and Demo

**Use Case – Inspect by enhanced Monitoring & Logging & Tracing:** as an operator/administrator, I'd like to:

Gather traffic metrics and collect customized logging for all CFEE components, not only underneath Cloud Foundry metrics and component and application logging.

With tracing abilities, tracking down how user request flow through each of our components, and figure out where the request failed and find the performance bottle neck.



# Use Cases and Demo

Jaeger UI   [Lookup by Trace ID...](#)   [Search](#)   [Compare](#)   [Dependencies](#)   [About Jaeger](#)

### Find Traces

Service (6) [productpage.default](#)

Operation (3) [all](#)

Tags [http.status\\_code=200 error=true](#)

Lookback [Last Hour](#)

Min Duration [e.g. 1.2s, 100ms, 500us](#)

Max Duration [e.g. 1.2s, 100ms, 500us](#)

Limit Results [20](#)

[Find Traces](#)

### 10 Traces

Sort: Most Recent

Compare traces by selecting result items

- istio-ingressgateway: productpage.default.svc.cluster.local:9080/productpage fb1b0e2   38.21ms
  - 8 Spans
  - details.default (1)
  - istio-ingressgateway (1)
  - productpage.default (3)
  - ratings.default (1)
  - reviews.default (2)
- istio-ingressgateway: productpage.default.svc.cluster.local:9080/productpage fb1b0b0   37.76ms
  - 8 Spans
  - details.default (1)
  - istio-ingressgateway (1)
  - productpage.default (3)
  - ratings.default (1)
  - reviews.default (2)
- istio-ingressgateway: productpage.default.svc.cluster.local:9080/productpage fce4b3e   28.3ms
  - 6 Spans
  - details.default (1)
  - istio-ingressgateway (1)
  - productpage.default (3)
  - reviews.default (1)

Jaeger UI   [Lookup by Trace ID...](#)   [Search](#)   [Compare](#)   [Dependencies](#)   [About Jaeger](#)

← [istio-ingressgateway: productpage.default.svc.cluster.local:9080/productpage](#)   [fb1b0e2](#)

Trace Start June 10, 2019 2:37 PM Duration 38.21ms Services 5 Depth 6 Total Spans 8

0ms 9.55ms 19.11ms 28.66ms 38.21ms

### Service & Operation

- istio-ingressgateway productpage.default.svc.cluster.local:9080/productpage
- productpage.default productpage.default.svc.cluster.local:9080/productpage
  - productpage.default details.default.svc.cluster.local:9080/productpage
  - details.default details.default.svc.cluster.local:9080/productpage
- productpage.default reviews.default.svc.cluster.local:9080/productpage
  - reviews.default reviews.default.svc.cluster.local:9080/productpage
  - ratings.default ratings.default.svc.cluster.local:9080/productpage

Distributed tracing from istio.io

Jaeger UI   [Lookup by Trace ID...](#)   [Search](#)   [Compare](#)   [Dependencies](#)   [About Jaeger](#)

← [bbs: DesireDesiredLRP](#) 3340de1

Trace Start June 21, 2019 10:25 PM Duration 11.82ms Services 1 Depth 2 Total Spans 2

0ms 2.95ms 5.91ms 8.86ms 11.82ms

### Service & Operation

- bbs DesireDesiredLRP
  - bbs RequestLRPAuctions

**RequestLRPAuctions**

Tags:  
Process: client-uuid=4928081ae44f48c7 | hostname=diego-api-0 | ip=172.30.125.123 | jaeger.version=Go-2.16.1dev

Service: bbs Duration: 6.04ms Start Time: 5.72ms

SpanID: 6b9624fe06adcc5a

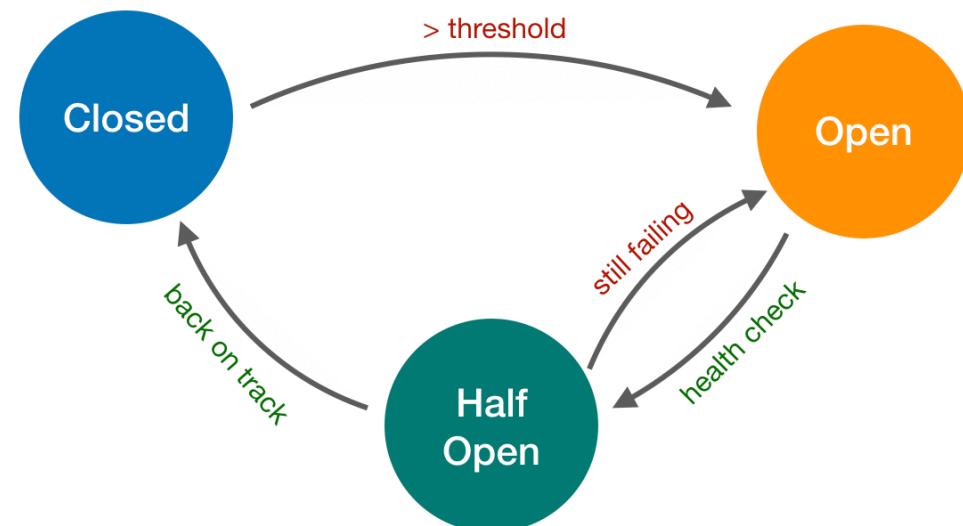
```
func (h *DesiredLRPHandler) RequestLRP(logger lager.Logger, w http.ResponseWriter, req *http.Request) {
    logger = logger.Session("request-lrp")
    span := opentracing.GlobalTracer().StartSpan("RequestLRPAuctions",
        opentracing.ChildOf(opentracing.SpanFromContext(ctx).Context()))
    defer span.Finish()
    ctx = opentracing.ContextWithSpan(ctx, span)
```

```
incoming_headers = [
    'x-request-id',
    'x-b3-traceid',
    'x-b3-spanid',
    'x-b3-parentspanid',
    'x-b3-sampled',
    'x-b3-flags',
    'x-ot-span-context'
]
```

# Use Cases and Demo

**Use Case – Fail Gracefully by Circuit Breaking:** as an operator/administrator, I'd like to:

Retrying against an already-slow service just makes the service and whole environment to lag. With circuit breaker, eliminate slow CFEE components until they recover and avoid bringing down the whole CFEE environment, without changing CFEE component source code or configuration.



# Use Cases and Demo

**Use Case – Blue/Green Deployment:** as an administrator, I'd like to:

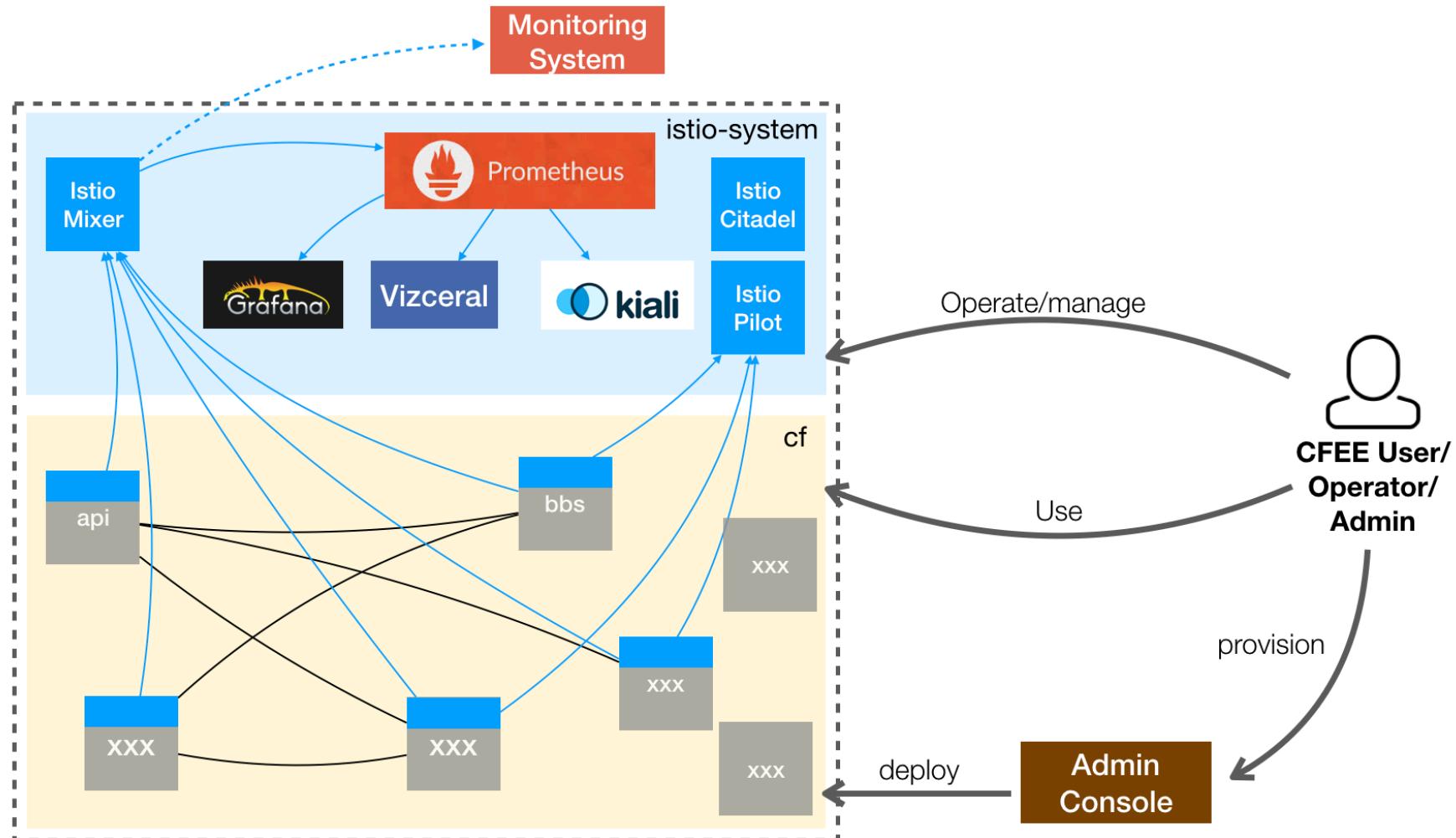
apply Blue/Green Deployment for CFEE, instead of simple K8s rolling out, so we can control and distribute the Blue and Green version based on various properties of the request to do verification, and flip over the traffic to complete upgrade.

**Use Case – Canary Deployment:** as an administrator, I'd like to:

apply Canary Deployment for CFEE, instead of simple K8s rolling out, so we can have the ability to do fine tuned traffic shifting, to verify on actual production traffic, to release a new version to subset of users and finally observe and complete the whole upgrade.

# Backup

# Backup



# Thank You! 😊