



Practices of Go Microservices on Post-Kubernetes

Wei Zheng

Background in Shimo

Language

- Go

Framework

- Gin
- Echo
- gRPC
- ...

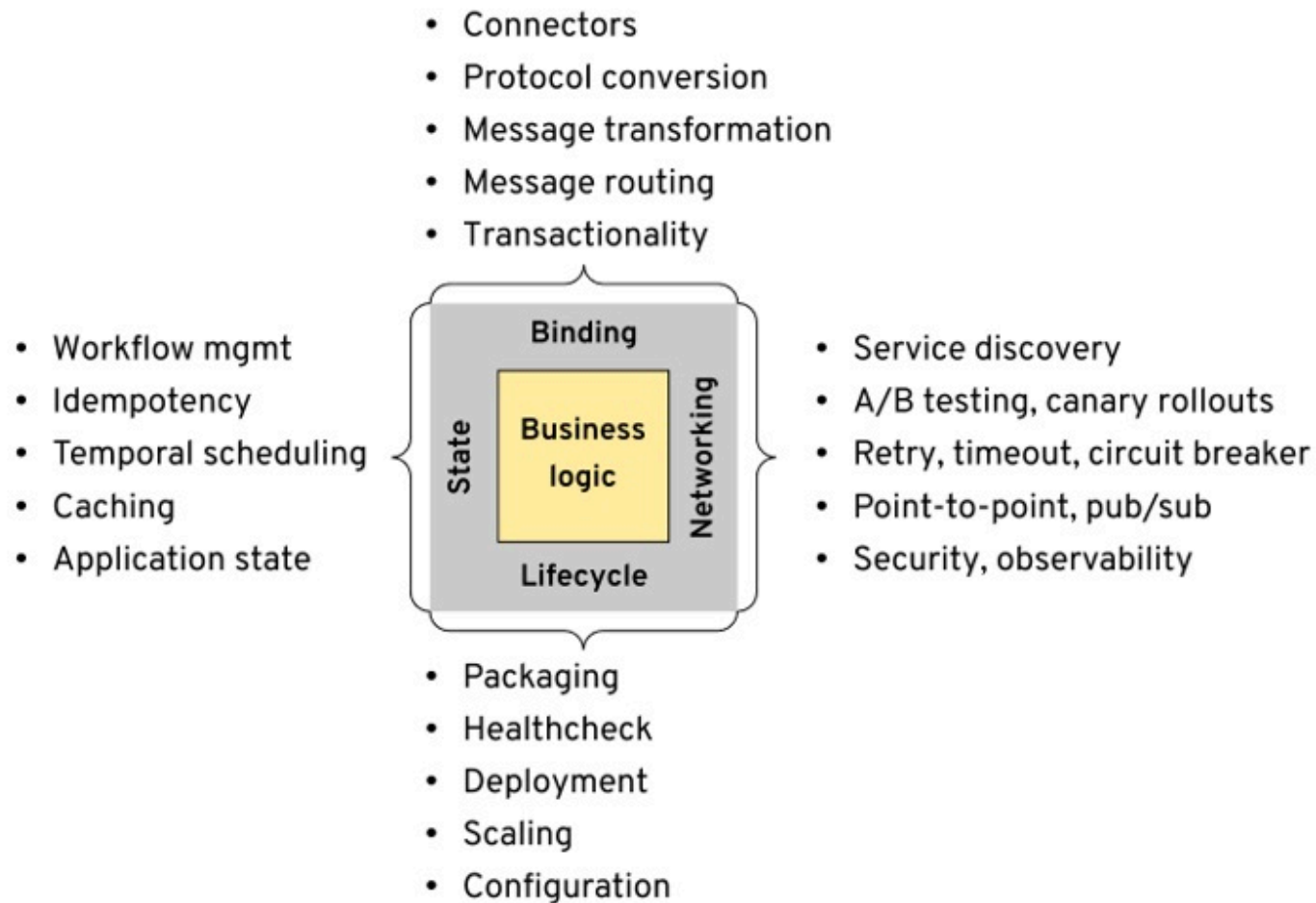
Platform

- All in Kubernetes
 - Ingress Controller
 - Stateless Deployment
 - Stateful Sets
 - ...
- All in Cloud
 - ECS
 - RDS
 - OSS
 - SLB
 - ...

Background in Shimo

- High performance & Stable
- Ecosystem
- Maintainable
- Pluggable
- ...

Background in Shimo

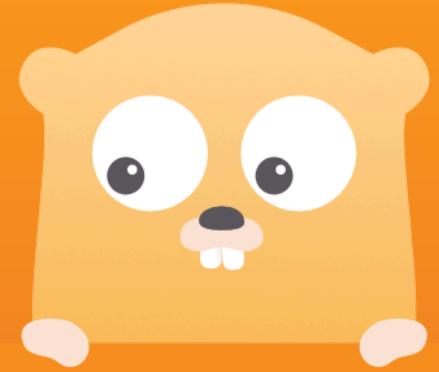


Background in Shimo

- gPRC
 - High performance
 - Multi Language support
 - Pluggable and ecosystem
- Service Mesh (Istio)
 - Cross-Language, out-of-box features, need no application change
 - Additional maintenance cost, not very stable
 - Only implement networking runtime

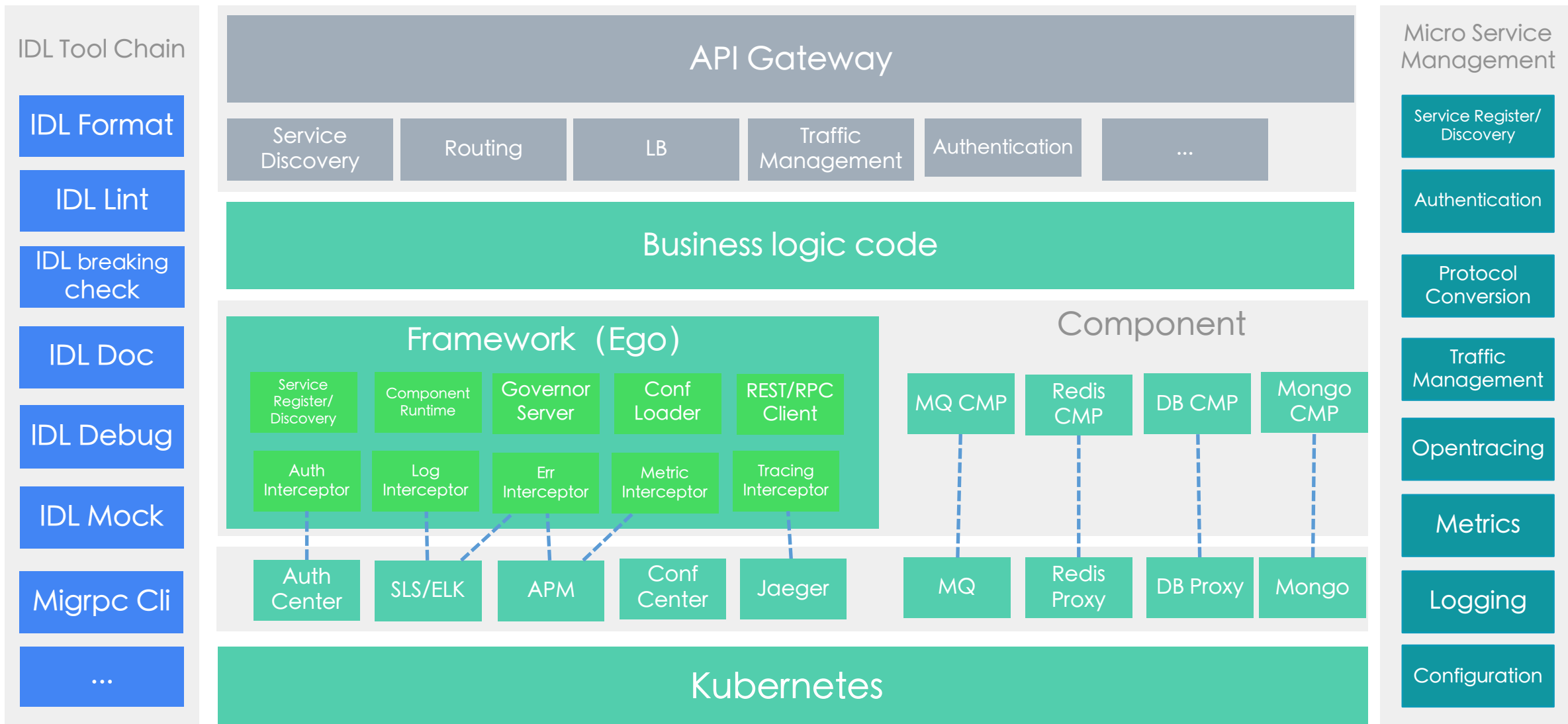
“

Ockham's Razor



Non sunt multiplicanda entia sine necessitate.

Architecture



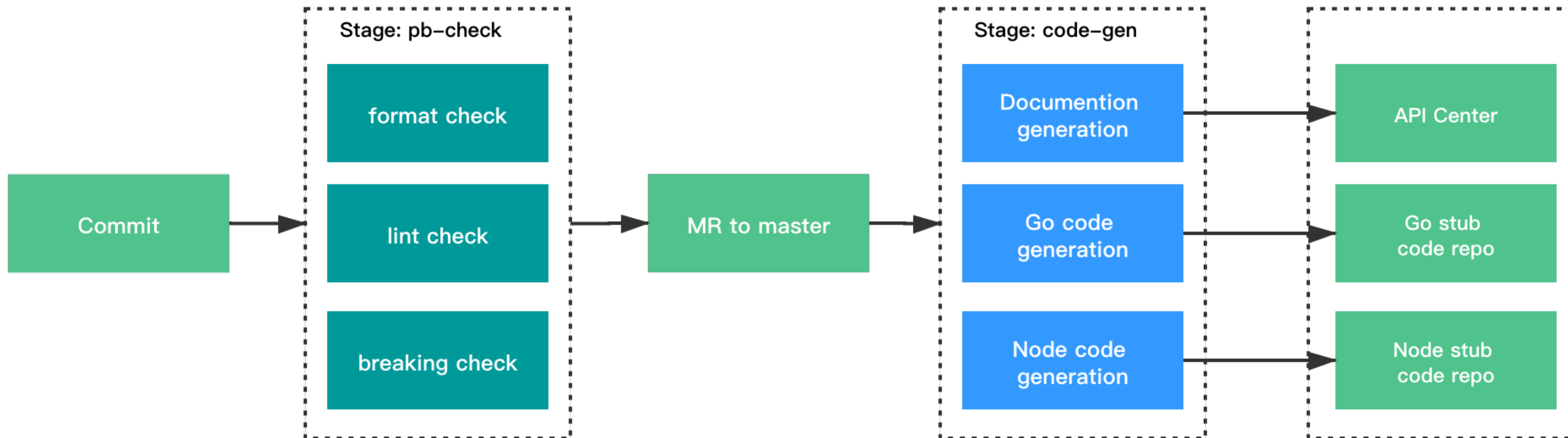
“

The increase of entropy theorem



The entropy of an isolated system can increase, but not decrease.

IDL Tool Chain - CI Workflow



IDL Tool Chain - Style Guide

- Protobuf mono repo
- Git workflow
- Custom cli tool
- IDL style guide
 - Google Style
 - Uber V2 Style

IDL Tool Chain - Style Guide

Rule	Description
Specing	Use two spaces instead of tabs.
Package Naming	<ul style="list-style-type: none"> - A package name is a full package name, i.e. uber.trip.v1. - A package sub-name is a part of a package name, ie uber, trip, or v1. - A package version is the last package sub-name that specifies the version, i.e. v1, v1beta.
Package Version	The last package sub-name should be a major version of the package, or the major version followed by the beta version.
Directory Structure	Files should be stored in a directory structure that matches their package sub-names.
Messages	Messages should always be PascalCase.
Message Fields	Message field names should always be lower_snake_case.
Services/RPCs	Services should always be PascalCase. RPCs should always be PascalCase.
Documentation	All comments should use // and not /* */.
...	




IDL Tool Chain - Cli

- egoctl: (<https://github.com/gotomicro/egoctl>)
 - Builtin Uber prototool
 - Custom IDL
 - Generate front-end code with React and Ant Design
 - Generate back-end code with ego

IDL Tool Chain - gRPC Debugging

- How to debugging service with dependencies?
 - Use VPN to connect develop kubernetes cluster.
 - With Kubernetes API server Resolver, we can resolve dependent services.

IDL Tool Chain - gRPC Debugging

 接口平台

文档

系统

郑伟

概览

gRPC

HTTP

infra.oss.v1

Oss

PutObject

GetObject

DeleteObject

DeleteObjects

ListObjects

SignURL

Target: ☒ Pod ☐ Custom

Pod:

sv7cc-bcwp2

Port:

50051

鉴权: ☒

ClientID:

ClientSecret:

S

发送

Metadata

Key	Value
key	value

Request

```
1 {
2   "prefix": "a",
3   "marker": "a",
4   "maxKeys": 999,
5   "delimiter": ""
6 }
```

Response

```
1 {
2   "keys": [
3     "console-be/4.jpg",
4     "console-be/4.jpg",
5     "console-be/4.jpg",
6     "console-be/4.jpg",
7     "console-be/4.jpg",
8     "console-be/4.jpg",
9     "console-be/4.jpg",
10    "console-be/4.jpg",
11    "console-be/4.jpg",
12    "console-be/4.jpg",
13    "console-be/4.jpg",
14    "console-be/4.jpg",
15    "console-be/4.jpg",
16    "console-be/4.jpg",
17    "console-be/4.jpg",
18    "console-be/4.jpg",
19    "console-be/4.jpg",
20    "console-be/4.jpg",
21    "console-be/4.jpg",
22    "console-be/4.jpg",
23    "console-be/4.jpg",
24    "console-be/4.jpg",
25    "console-be/4.jpg",
26    "console-be/4.jpg",
27    "console-be/4.jpg",
28    "console-be/4.jpg",
29    "console-be/4.jpg",
30    "console-be/4.jpg",
31    "console-be/4.jpg",
32    "console-be/4.jpg",
33    "console-be/4.jpg",
34    "console-be/4.jpg",
35    "console-be/4.jpg",
36    "console-be/4.jpg",
37    "console-be/4.jpg",
38    "console-be/4.jpg",
39    "console-be/4.jpg",
40    "console-be/4.jpg",
41    "console-be/4.jpg",
42    "console-be/4.jpg",
43    "console-be/4.jpg",
44    "console-be/4.jpg",
45    "console-be/4.jpg",
46    "console-be/4.jpg",
47    "console-be/4.jpg",
48    "console-be/4.jpg",
49    "console-be/4.jpg",
50    "console-be/4.jpg",
51    "console-be/4.jpg",
52    "console-be/4.jpg",
53    "console-be/4.jpg",
54    "console-be/4.jpg",
55    "console-be/4.jpg",
56    "console-be/4.jpg",
57    "console-be/4.jpg",
58    "console-be/4.jpg",
59    "console-be/4.jpg",
60    "console-be/4.jpg",
61    "console-be/4.jpg",
62    "console-be/4.jpg",
63    "console-be/4.jpg",
64    "console-be/4.jpg",
65    "console-be/4.jpg",
66    "console-be/4.jpg",
67    "console-be/4.jpg",
68    "console-be/4.jpg",
69    "console-be/4.jpg",
70    "console-be/4.jpg",
71    "console-be/4.jpg",
72    "console-be/4.jpg",
73    "console-be/4.jpg",
74    "console-be/4.jpg",
75    "console-be/4.jpg",
76    "console-be/4.jpg",
77    "console-be/4.jpg",
78    "console-be/4.jpg",
79    "console-be/4.jpg",
80    "console-be/4.jpg",
81    "console-be/4.jpg",
82    "console-be/4.jpg",
83    "console-be/4.jpg",
84    "console-be/4.jpg",
85    "console-be/4.jpg",
86    "console-be/4.jpg",
87    "console-be/4.jpg",
88    "console-be/4.jpg",
89    "console-be/4.jpg",
90    "console-be/4.jpg",
91    "console-be/4.jpg",
92    "console-be/4.jpg",
93    "console-be/4.jpg",
94    "console-be/4.jpg",
95    "console-be/4.jpg",
96    "console-be/4.jpg",
97    "console-be/4.jpg",
98    "console-be/4.jpg",
99    "console-be/4.jpg",
100   "console-be/4.jpg"
    ]
  }
```


概览

gRPC

HTTP

关联Pb

▼ infra.oss.v1

▼ Oss

PutObject

GetObject

DeleteObject

DeleteObjects

ListObjects

SignURL

描述

查询多个对象

请求

ListObjectsRequest

名称	类型	描述
prefix	string	指定前缀
marker	string	标识符
maxKeys	int32	最多查询的对象数量
delimiter	string	分隔符

响应

ListObjectsResponse

名称	类型	描述
keys	string repeated	符合要求的对象键名

Framework - Service Register/Discovery

- Based on Kubernetes
 - Service replicate register its endpoint to Kubernetes
 - Service discovery based on the Kubernetes API resolver

Framework - Service Register/Discovery

- Kubernetes DNS Resolver VS Kubernetes API Server Resolver
 - DNS resolver is builtin in gRPC framework and its out-of-box for users
 - When connection fail DNS Resolver can resolve name immediately
 - In scale-up scenario, DNS Resolver will not resolve name automatically

Framework - Service Register/Discovery

```
...  
informer.Informer().AddEventHandler(cache.Resource  
EventHandlerFuncs{  
    AddFunc:  c.addEndpoints,  
    UpdateFunc: c.updateEndpoints,  
    DeleteFunc: c.deleteEndpoints,  
})  
...
```

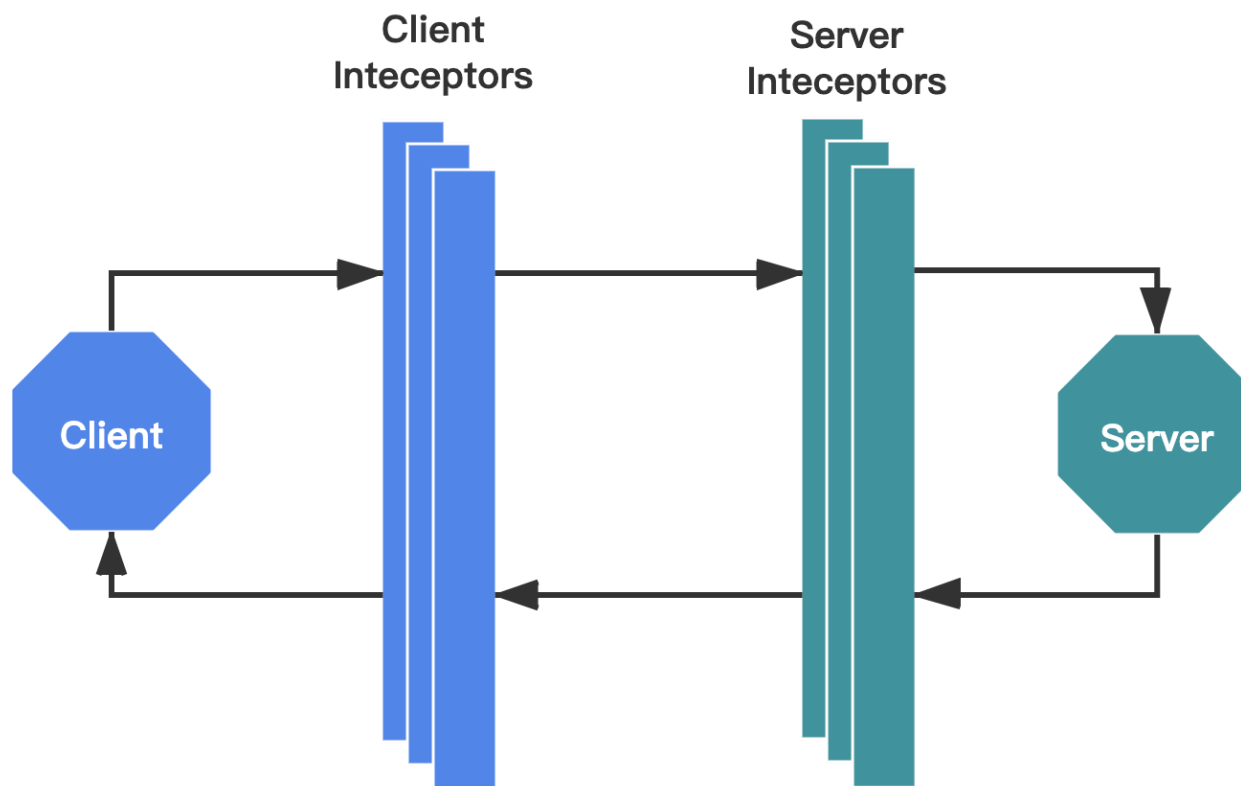
```
[grpc.otherService]  
debug = true  
addr = "k8s:///other-service:9090"  
token = "YOUR_KUBERNETES_API_SERVER_TOKEN"
```

K8s API Server Resolver

Framework - Service Register/Discovery

Service Register/Discovery

Framework - Builtin Interceptors



Framework - Builtin Interceptors

Builtin Interceptors

- Authentication Interceptor
- Logging Interceptor
- Tracing Interceptor
- Metrics Interceptor
- Merrors Interceptor
- ...

Framework - Error Handling

Error Handling

- gRPC Code
- Biz Code
- Biz Message

Framework - Error Handling

Client/Server	gRPC Code	Value	HTTP Code	Description
	OK	0	0	returned on success.
Client	INVALID_ARGUMENT	3	400	The client specified an invalid argument.
	NOT_FOUND	5	404	Some requested entity was not found.
	UNAUTHENTICATED	16	401	The request does not have valid authentication credentials for the operation.
	...			
Server	UNKNOWN	2	500	Unknown error.
	UNIMPLEMENTED	12	501	The operation is not implemented or is not supported/enabled in this service.
	UNAVAILABLE	14	503	The service is currently unavailable.
	...			

Framework - Error Handling

Prefix Fragment	Middle Fragment	Suffix Fragment
100-999	000-999	0000-9999

Framework - Error Handling

```
func New(grpcCode codes.Code, bizCode int32, bizMsg string) error {  
    st, err := status.New(grpcCode, bizMsg).WithDetails(  
        &gstatus.Status{Code: int32(bizCode), Message: bizMsg},  
    )  
    if err != nil {  
        return err  
    }  
    return st.Err()  
}
```

Encoding An Error

Framework - Configuration

- Toml、JSON、Yaml Supported
- Dynamic change configuration on the fly
- Human friendly UI for configuration edit
- Integration with Kubernetes Config Map
- ...

Framework - Configuration

应用: ego-demo

环境: dev

集群: shimo-dev

Namespace: default

实例

配置

监控

日志

资源依赖

PProf

Grpc调试

应用事件

dev.toml

未保存

1 debug = true

2 [redis]

3 | addr="{{dredis.addr@14}}"

4 | password="{{dredis.password@14}}" # refrence of dev redis password

5 [server.governor]

6 | host = "0.0.0.0"

7 | port = 9004

8 [server.http]

9 | host = "0.0.0.0"

10 | port = 9001

11 [logger.default]

12 | debug = true

13 | async = false

14 | level = "info"

15 [logger.ego]

16 | debug = true

17 | async = false

18 | level = "info"

19

资源变量: dredis

Key: password

Description: dev redis for testing

Value: YOUR_PASSWORD

Author: 郑伟

Tags: redis

Stability

- Compatible with semantic version
- Ensure unit test coverage
- Foresight in design
- Defensive and pluggable design for APIs

Iteration

- Manage iteration lifecycle with issues
- Do iteration with you framework users
- Open source for all codes and welcome any necessary PRs, reviews or suggestions

Independent Module

- All components in mono repo
- All components has its own module

Components

```
" Press ? for help
.. (up a dir)
</zheng/shimo/ego-component/
▼ etcd/
  ► examples/
  ► registry/
    component.go
    config.go
    container.go
    go.mod
    go.sum
    lock.go
    watch.go
▼ egorm/
  ► dsn/
  ► examples/
    component.go
    config.go
    container.go
    go.mod
    go.sum
    init.go
    instance.go
    interceptor.go
    logger.go
    option.go
    README.md
```

```
1 module github.com/gotomicro/ego-component/egorm
2 go 1.15
3
4 require (
5     github.com/gotomicro/ego v0.4.1
6     github.com/json-iterator/go v1.1.10
7     github.com/opentracing/opentracing-go v1.1.0
8     github.com/stretchr/testify v1.6.1
9     gorm.io/driver/mysql v1.0.5
10    gorm.io/driver/postgres v1.0.8
11    gorm.io/gorm v1.21.3
12 )
```

```
1 module github.com/gotomicro/ego-component/eetcd
2 go 1.15
3
4 require (
5     github.com/golang/protobuf v1.4.2
6     github.com/gotomicro/ego v0.4.1
7     github.com/grpc-ecosystem/go-grpc-prometheus v1.2.0
8     go.etcd.io/etcd v0.5.0-alpha.5.0.20200425165423-262c93980547
9     golang.org/x/net v0.0.0-20201224014010-6772e930b67b
10    google.golang.org/grpc v1.29.1
11 )
```

Configuration

- Standard configuration schema
- Integrated with Kubernetes ConfigMap

Lifecycle

- Control component lifecycle

Debugging

- Dump Request/Response during debugging

Tracing

- Add context to all APIs for logging tracing span during all operations

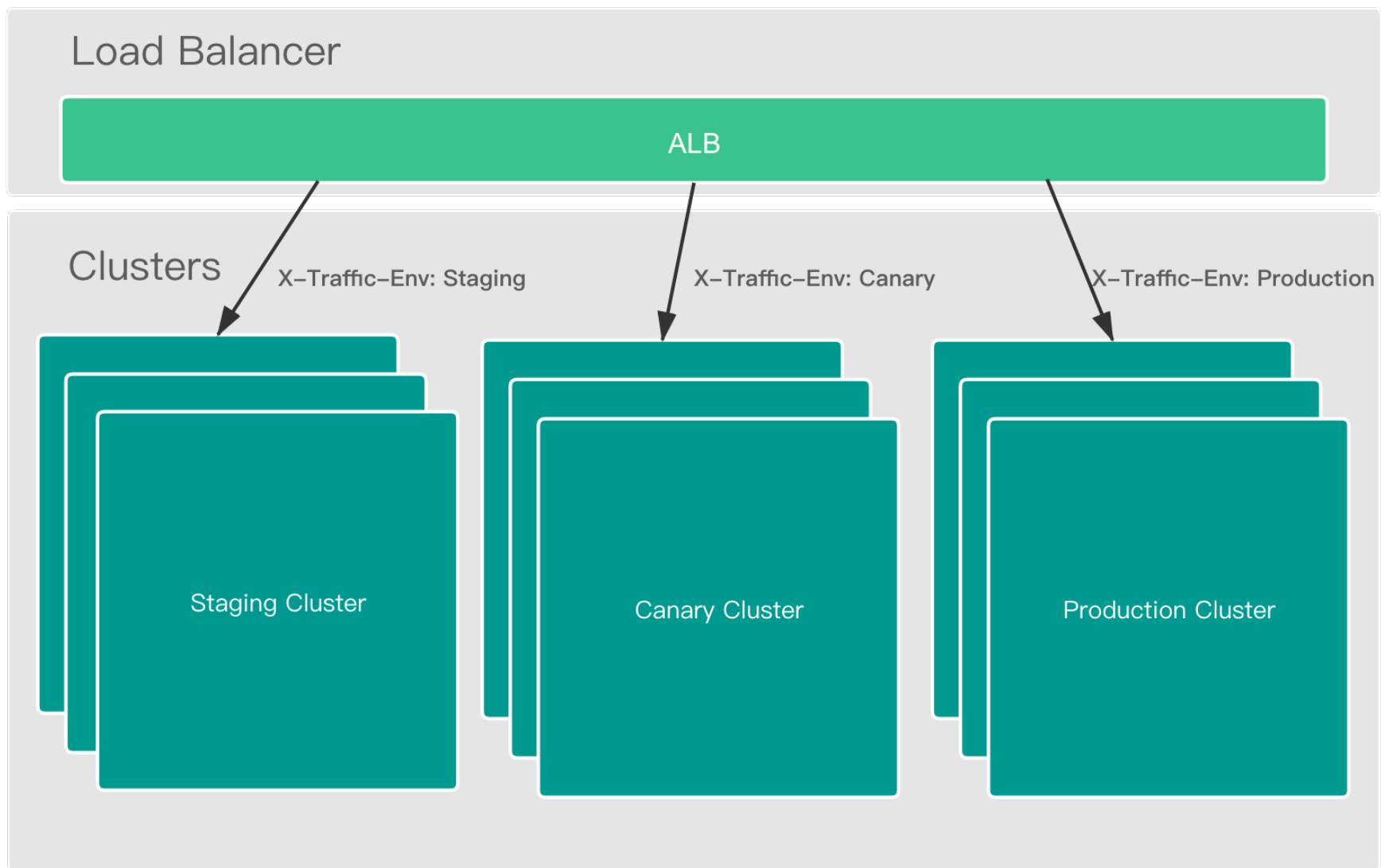
Logging

- Append standard logging when error occurs

Traffic Management - Rate limiting/Circuit Breaker

- Ingress Controller
- Client/Server side interceptor

Traffic Management - Canary



- Based on Kubernetes
 - User manage service resource configuration by themselves
 - User Scaling-up/Scaling-down replica manually or via Kubernetes Autoscaler

Observability - Logging

- Based on Uber zap
 - Auto logging with TraceID、ServiceName、Framework Version...
 - Structured logging for friendly searching
 - Extract business logger from framework logger

Observability - Metrics

- Based on Kubernetes/Prometheus/Grafana
 - Builtin framework and user custom metrics collector
 - Export APIs for user add own metrics
 - Declare Prometheus annotations for service discovery and metrics scraping

Observability - Metrics

Pod Template:

Labels:

app=my-app
enable=true
version=pro

Annotations:

prometheus.io/port: 9003
prometheus.io/scrape: true

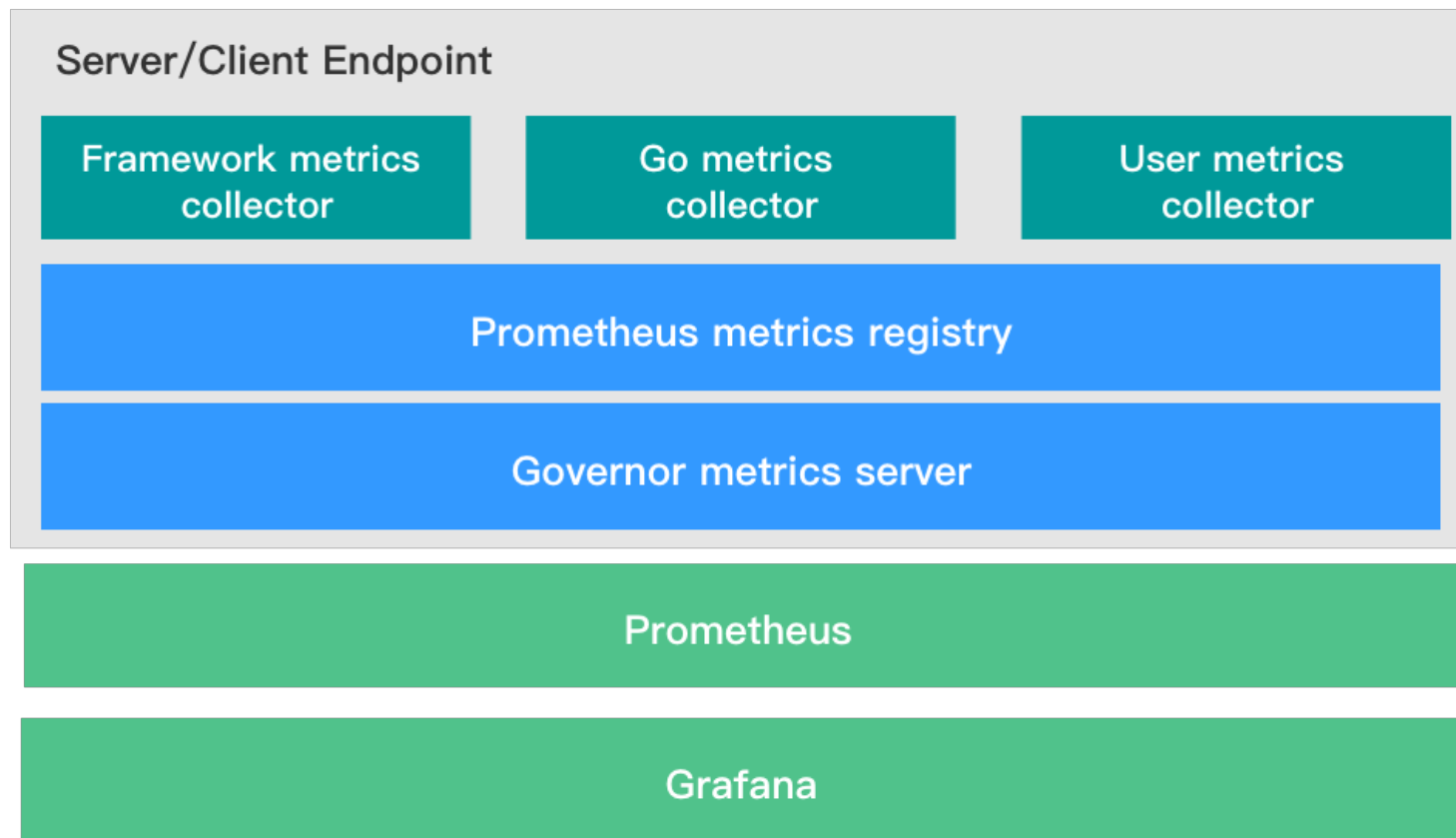
Kubernetes Prometheus Annotations

Observability - Metrics

Naming

- Metric name should be snake_case
- User metric name should start with custom_
- Metric name should end with _total、_seconds、_bytes、_ratio、_info

Observability - Metrics



Observability - Metrics

实例 配置 监控 日志 资源依赖 PProf 应用事件 授权管理

实例监控

API监控

概览监控

Client监控

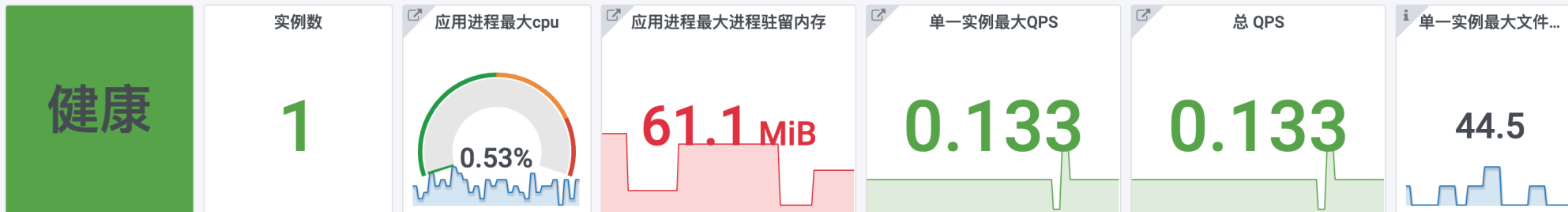
instance ☆ 🔊

📊 📄 ⚙️ 🖨️ ⌚ Last 30 minutes 🔍 ↺️ ▼

数据源 pro. ro ▼ 应用名 s e ▼ 环境(版本) pro ▼ 实例 All ▼ summarize 5m ▼ 方法名 All ▼ appLabel of deployment s ble ▼

namespace default ▼

概览



Instance									
Pod	启动时间	应用版本	框架版本	enable	Go版本	Pod IP	Namespace	pod_template_hash	vers
ple-595...	2021-04-02 14:09:15	cded1c82995b5716992c7...	v0.4.3	true	go1.15.6	9003	default	595857b74c	pro

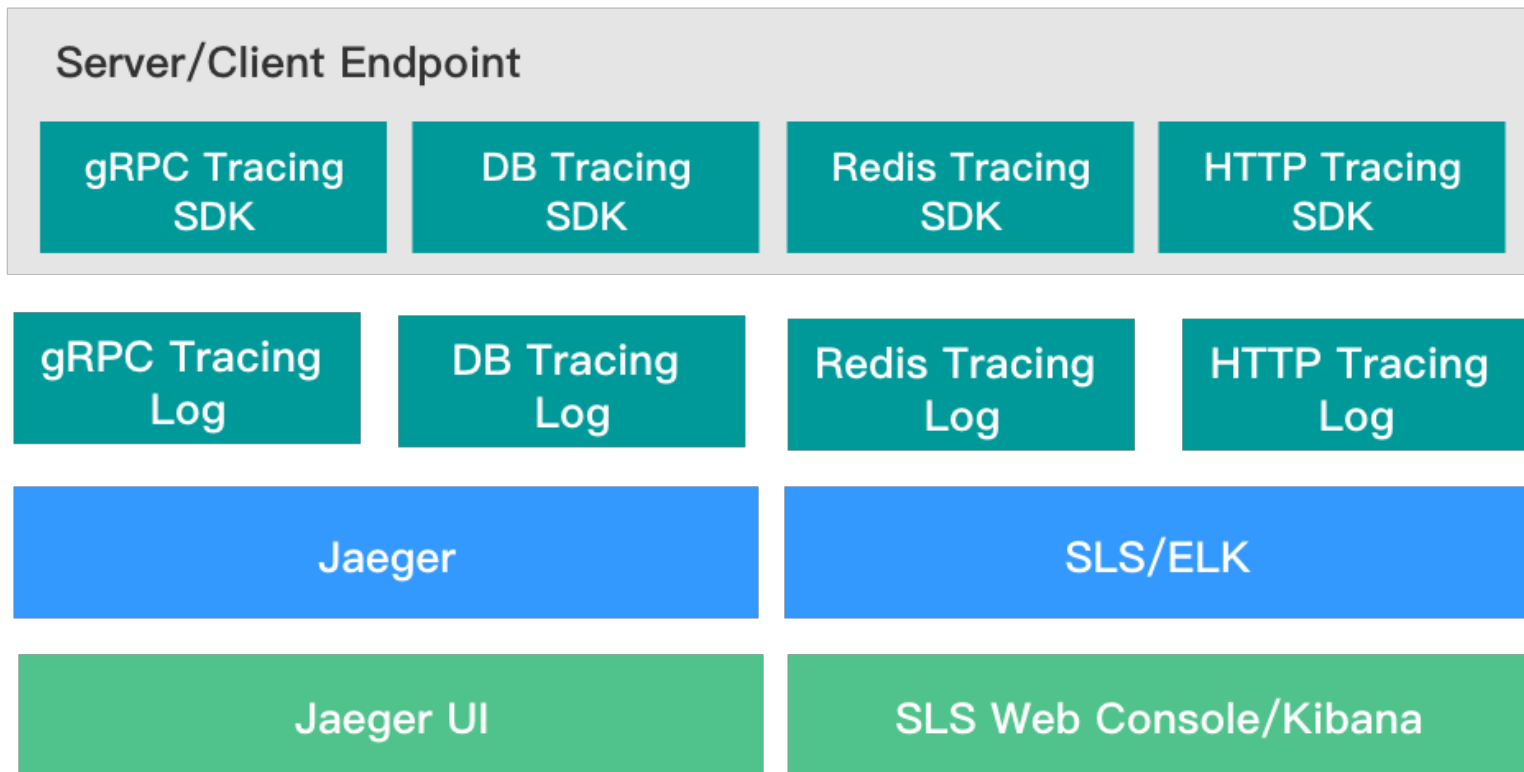
实例



Observability - Opentracing

- Adding context style APIs for all component:
 - HTTP Request
 - DB Query
 - Redis Query
 - Mongo Query
 - Kafka Command
- Following tracing span tags naming guide: db.type、db.instance、db.statement...
- Integrating trace-id when logging

Observability - Opentracing



Q & A

文档实时协同 · 知识沉淀管理 · 数据安全可靠