



CLOUD NATIVE  
COMPUTING FOUNDATION

# K8s 上的图数据库

古思为 — 欧若数网 — Nebula Graph 开发者布道师



# 古思为

- Nebula Graph 开发者布道师
- 程序员
- 开源信徒



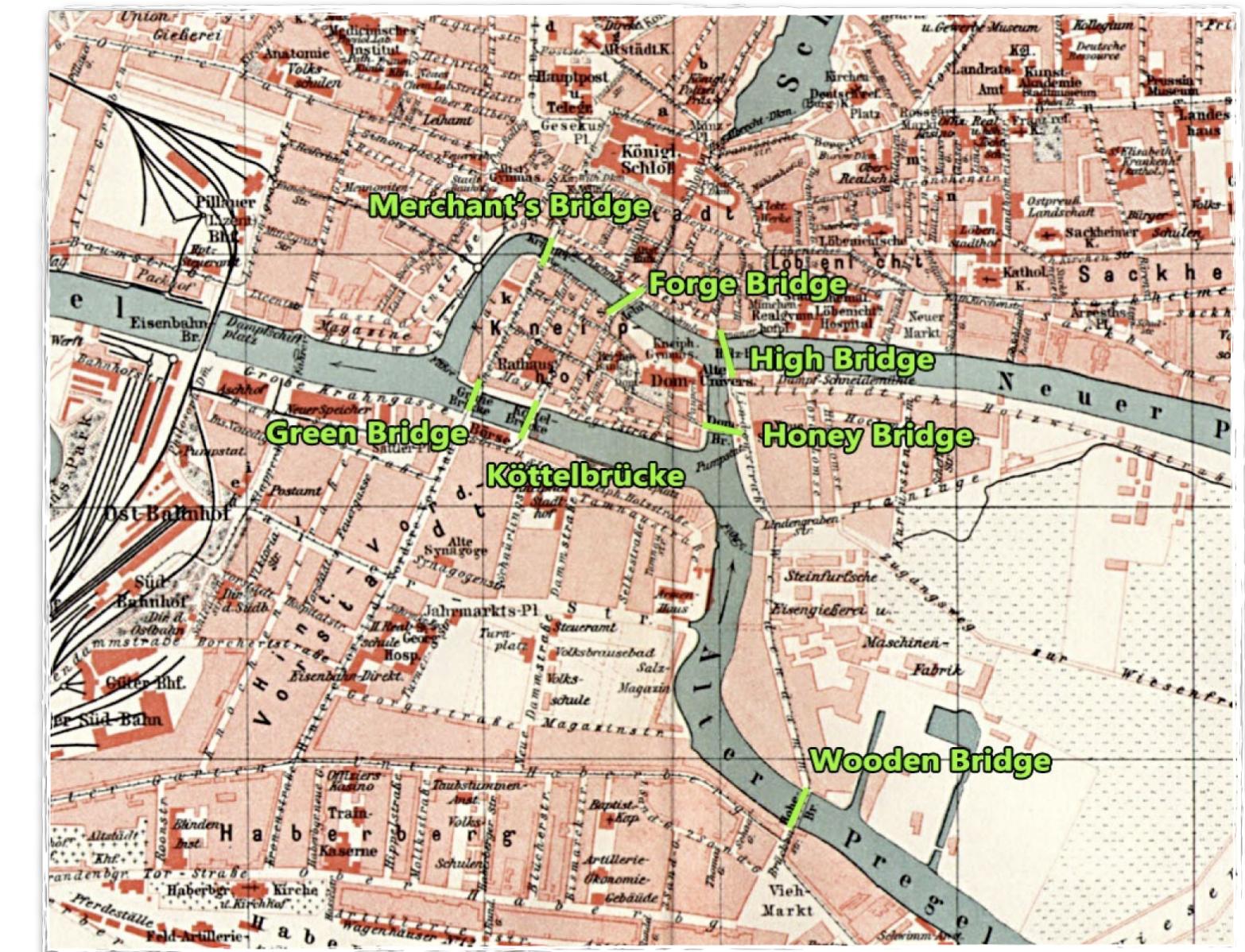
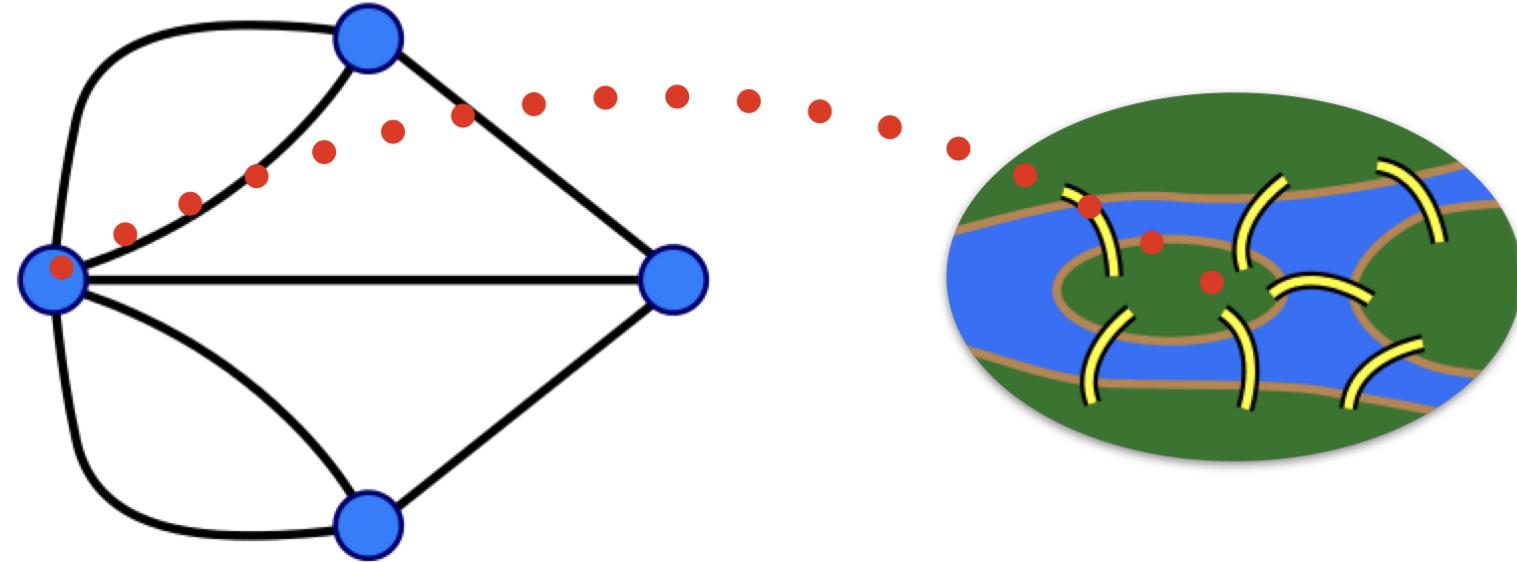
❏ wey-gu

❏ wey\_gu

❏ siwei.io

# 图数据库简介

什么是图？ 什么是图数据库？ 为什么我们需要一个专门的数据库？



Map of Königsberg with the seven bridges labeled, circa 1905

"以图结构、图语义来用点、边、属性来查询、表示存储数据的数据库

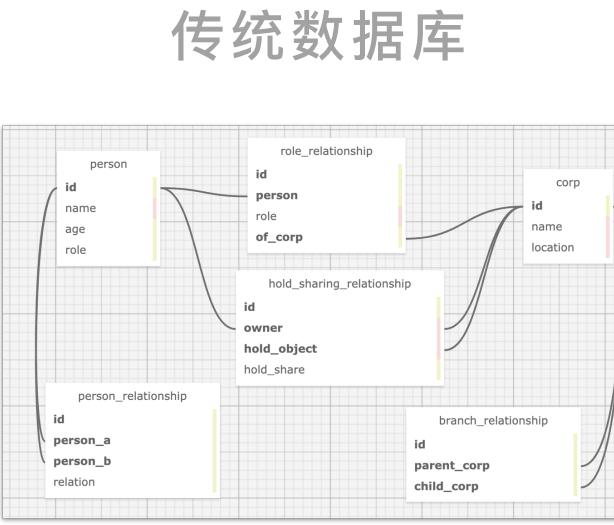
什么是图数据库

[wikipedia.org/wiki/graph\\_database](https://en.wikipedia.org/wikipedia/en/wiki/Graph_database)

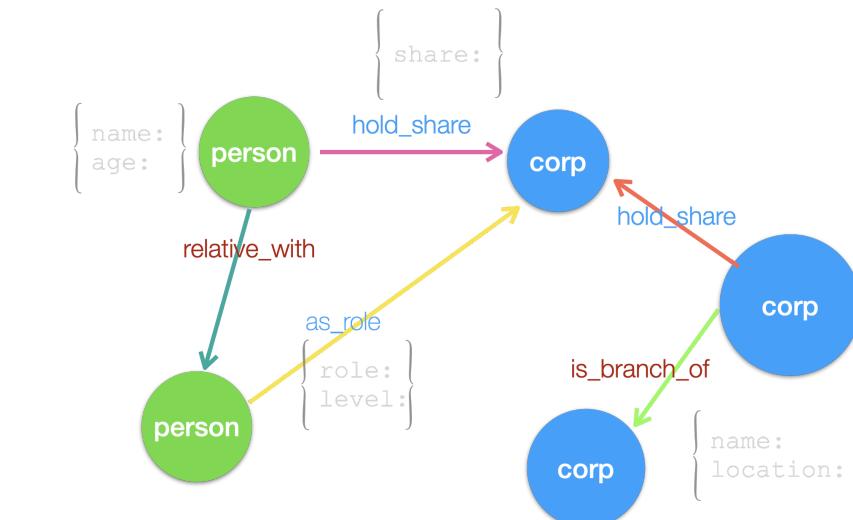
了解更多关于 [什么是图数据库](#)

# 为什么需要图数据库？

图模型的结构



图数据库



图语义的查询

```
SELECT a.id, a.name, c.name
FROM player a
JOIN serve b ON a.id=b.player_id
JOIN team c ON c.id=b.team_id
WHERE c.name IN (SELECT c.name
FROM player a
JOIN serve b ON a.id=b.player_id
JOIN team c ON c.id=b.team_id
WHERE a.name = 'Tim Duncan')
```

```
GO FROM 100 OVER serve YIELD serve._dst AS Team | \
GO FROM $-.Team OVER serve REVERSELY YIELD $$ .player.name;
```

性能

	主要应用场景	2-hop 延时 (~2.5K)	3-hop 延时 (~110K)	4-hop 延时 (~600K)
图数据库	关系遍历	0.01 秒	0.168 秒	1.36 秒
SQL数据库	信息检索	0.016 秒	30 秒	1544 秒

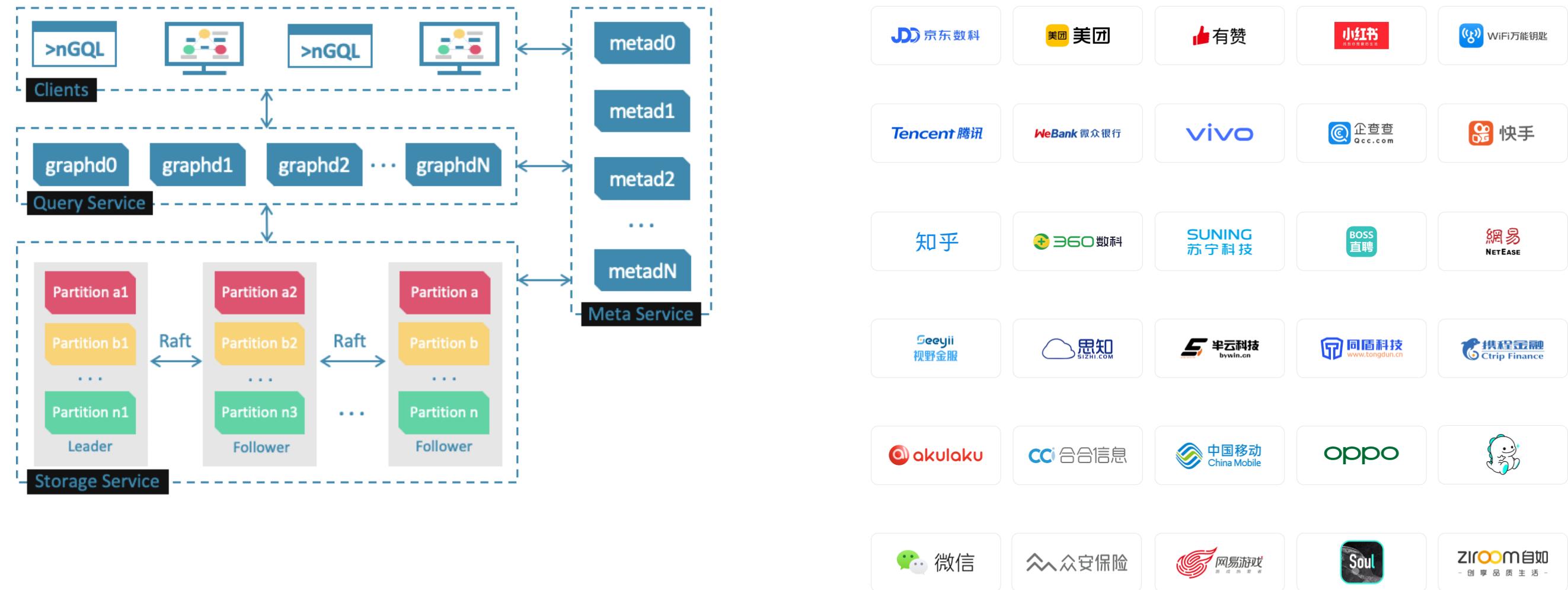
# Nebula Graph!

如何发音：['nebjələ]，它那些特点特点？

# Nebula Graph 介绍

一个可靠的分布式、线性扩容、性能高效的图数据库

世界上唯一能够容纳千亿顶点和万亿条边，并提供毫秒级查询延时的图数据库解决方案



了解更多 >>>

文档: [Nebula 架构](#)

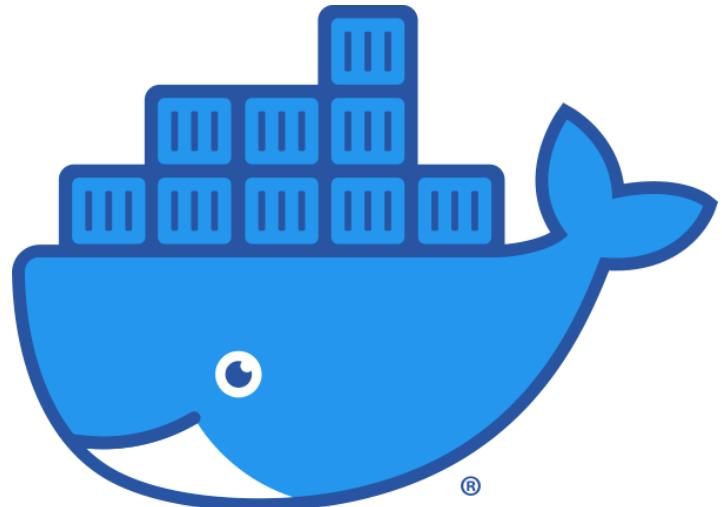
官网: [用户案例](#)



# 云原生时代的图数据库

# 容器化部署演进

NEBULA DOCKER



NEBULA K8S



NEBULA OPERATOR



# Nebula Operator 实现

## KUBEBUILDER SCAFFOLD

```
.  
├── apis  
..  
|   └── apps  
|   └── cmd  
|       └── ngctl  
|       └── controller-manager  
|   └── config  
|       └── crd  
└── pkg  
    ├── controller  
    ├── ngctl  
    ├── nebula  
    └── scheduler  
└── hack  
└── doc  
└── tests  
    └── e2e
```

## CRD

```
apiVersion: apps.nebula-graph.io/v1alpha1  
kind: NebulaCluster  
metadata:  
  name: nebula  
spec:  
  graphd:  
    resources:  
      requests:  
        cpu: "500m"  
        memory: "500Mi"  
    replicas: 3  
    image: vesoft/nebula-graphd  
    version: v2.5.0  
...  
  reference:  
    name: statefulsets.apps.kruise.io  
    version: v1  
  schedulerName: default-scheduler
```

 [vesoft-inc/nebula-operator](#)

## CONTROL LOOP

```
while True  
    actual_state = get_state(context)  
    expected_state = get_expected(context)  
    if actual_state == expected_state:  
        continue  
    else:  
        reconcile(context)
```

## CALLING NEBULA CLUSTER

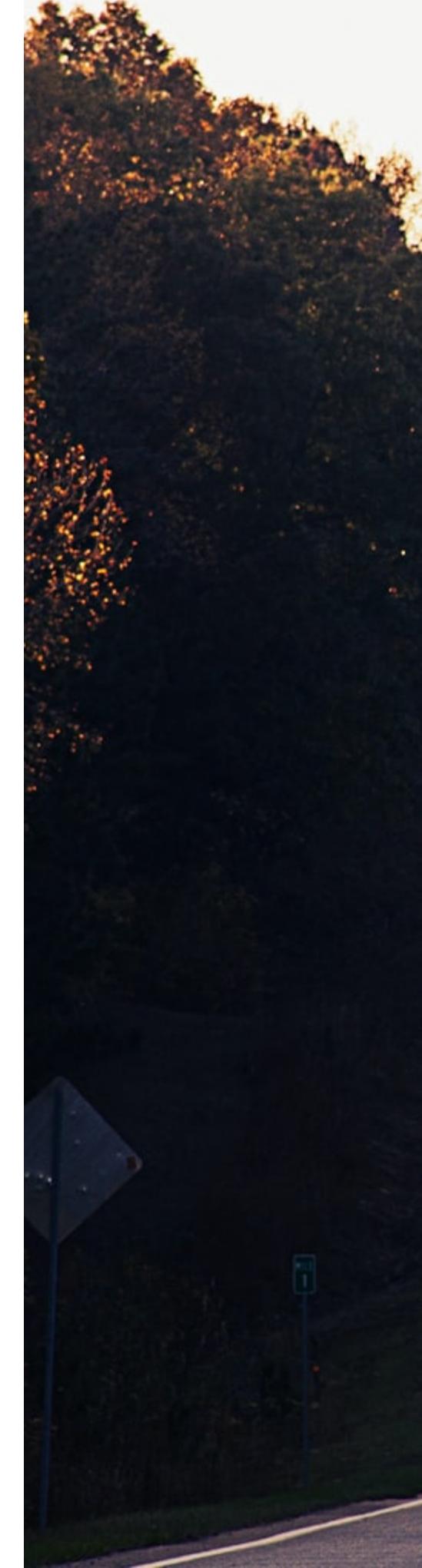
```
func (s *storUpd) updPhase(mc nebula.MI) {  
    if err := mc.Balance(); err != nil {  
        return err  
    }  
    hostItem, err := mc.ListHosts()  
    if err != nil {  
        return err  
    }  
    if !mc.IsBalanced(hostItem) {  
        if err := mc.Balance(); err != nil {  
            return err  
        }
```

# Nebula Operator Roadmap

- Rolling Upgrade
- Auto Scaling
- Integration with other Services

Check out our Github Repo and contribute!  
[vesoft-inc/nebula-operator](https://github.com/vesoft-inc/nebula-operator)

📍 roadmap of nebula-operator





Wey Gu 古思为  
@wey\_gu

Replying to @wey\_gu

Inspired by Carlos Santana's knative-kind, here comes the NGonK [#nebulaGraph #operator](#) in KinD.  
You can now have K8s+all-Deps+Nebula-Operator in a one-liner.



Nebula Operator Kind, 一键单机玩转 Nebula K8s 集群  
[siwei.io](http://siwei.io)

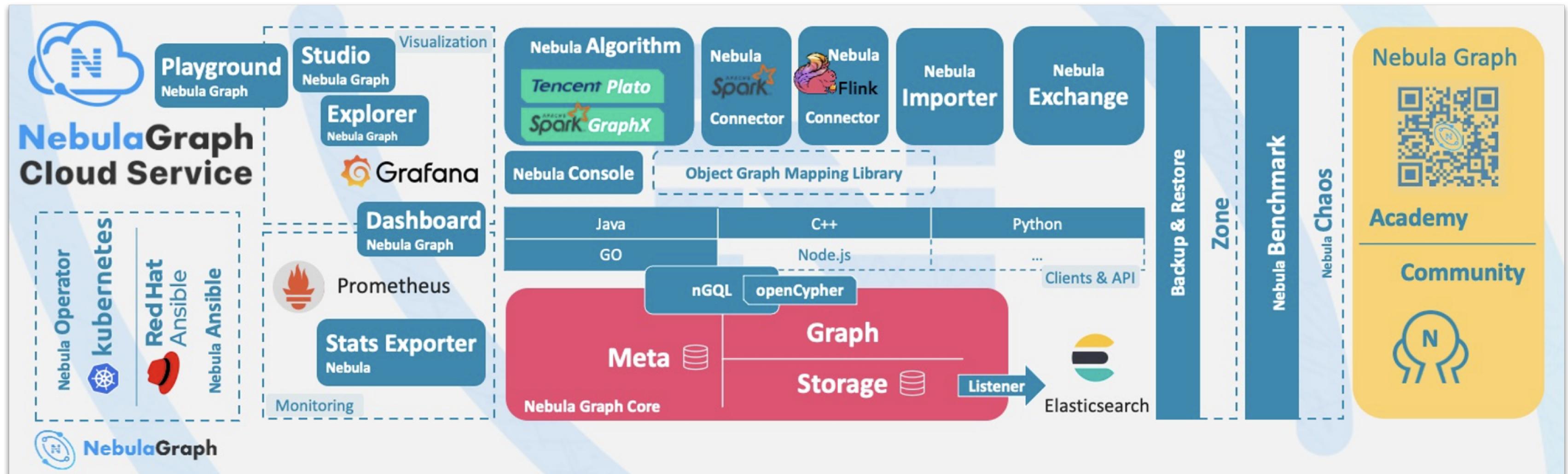
5:24 PM · Sep 1, 2021

1 3 1 Copy link t...

# Nebula Landscape

Nebula 社区生态非常丰富，并且还在日益拓展，欢迎同学们了解、参与贡献。

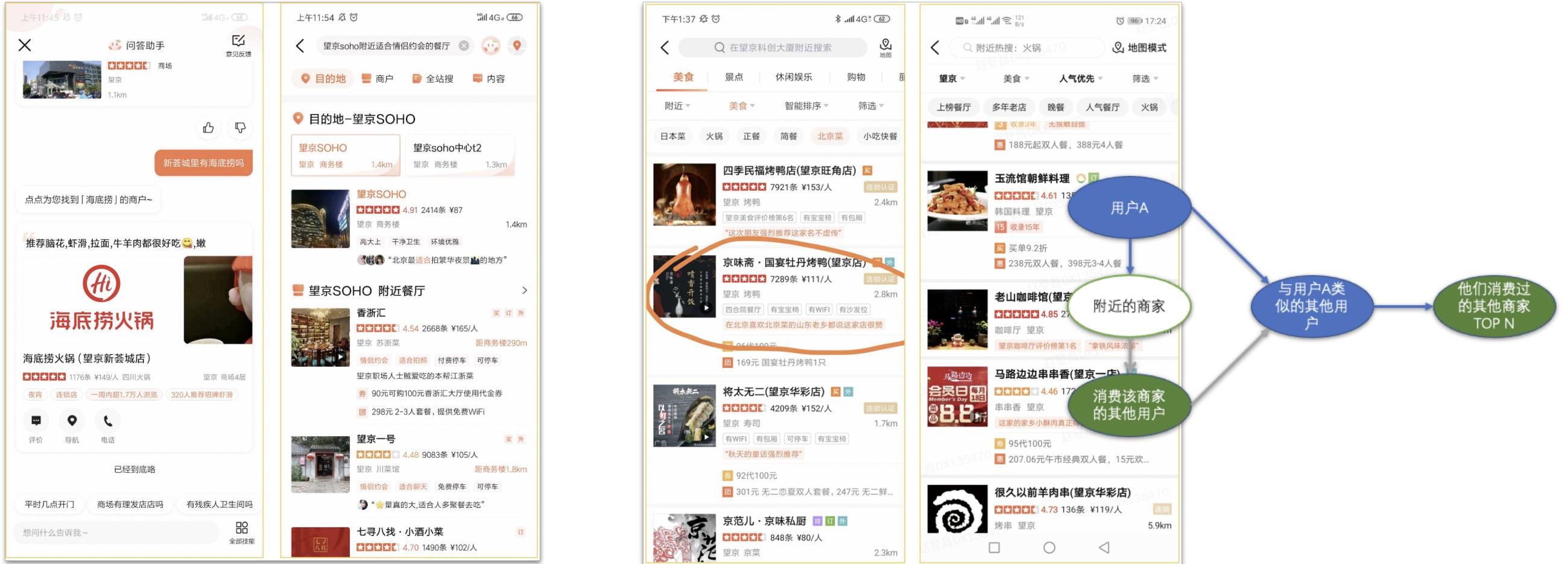
- Deployment, Monitoring
- Data Visualization
- Algorithm, Analytic
- Clients, Connectors, ETL



# 上手 GraphDB on K8s

应用场景 - Nebula on Kubesphere - Demo

# 图数据库的应用场景



## 典型场景

社交网络

风险控制

公共安全

知识图谱

机器学习

生化制药

物联网

区块链

数据血缘

智能运维

# KubeSphere + OpenFunction

```
kk create cluster --with-kubernetes v1.20.4 --with-kubesphere v3.1.1
git clone https://github.com/OpenFunction/OpenFunction.git
cd OpenFunction
sh hack/deploy.sh --all
kubectl apply -f https://raw.githubusercontent.com/OpenFunction/OpenFunction/main/config/bundle.yaml
kubectl create secret docker-registry push-secret \
  --docker-server=$REGISTRY_SERVER \
  --docker-username=$REGISTRY_USER \
  --docker-password=$REGISTRY_PASSWORD
kubectl --namespace kourier-system edit service kourier # externalIP
vim config/samples/function-sample.yaml # container image registry
kubectl apply -f config/samples/function-sample.yaml
$ kubectl get function
NAMESPACE      NAME          BUILDSTATE    SERVINGSTATE   BUILDER           SERVING
default        function-sample Succeeded     Created        function-sample-builder-s2pfg function-sample-serving-9sszk
$ kubectl get ksvc
NAME                           URL
function-sample-serving-9sszk-ksvc-xlfkz http://function-sample-serving-9sszk-ksvc-xlfkz.default.example.com
$ vim /etc/hosts

$ curl http://function-sample-serving-9sszk-ksvc-xlfkz.default.example.com
Hello, World!
```

# KubeSphere + Nebula Graph

```
curl -sL nebula-kind.siwei.io/install-ks-1.sh | bash # kubesphere-all-in-one nebula installer
```

```
$ kubectl get svc nebula-graphd-svc-nodeport
NAME                  TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)           AGE
nebula-graphd-svc-nodeport   NodePort  10.233.62.198  <none>        9669:32669/TCP,19669:32001/TCP  3m57s
$ kubectl edit svc nebula-graphd-svc-nodeport
service/nebula-graphd-svc-nodeport edited
$ ./nebula-kind/bin/console -u root -p password --address=192.168.8.137 --port=32669
2021/09/01 20:38:39 [INFO] connection pool is initialized successfully
Welcome to Nebula Graph!
```

```
(root@nebula) [(none)]> show hosts
+-----+-----+-----+-----+
| Host          | Port | Status | Leader count | Leader dist |
+-----+-----+-----+-----+
| "nebula-storaged-0.nebula-storaged-headless.default.svc.cluster.local" | 9779 | "ONLINE" | 0           | "No valid pa
+-----+-----+-----+-----+
| "nebula-storaged-1.nebula-storaged-headless.default.svc.cluster.local" | 9779 | "ONLINE" | 0           | "No valid pa
+-----+-----+-----+-----+
| "nebula-storaged-2.nebula-storaged-headless.default.svc.cluster.local" | 9779 | "ONLINE" | 0           | "No valid pa
+-----+-----+-----+-----+
| "Total"        |      |         | 0           |           |
+-----+-----+-----+-----+
```

# Nebula Graph 数据导入

```
$ wget https://docs.nebula-graph.io/2.0/basketballplayer-2.X.ngql
$ ~/.nebula-kind/bin/console -u root -p password --address=192.168.8.137 --port=32669 -f basketballplayer-2.X.ngql
...
(root@nebula) [basketballplayer]> insert edge serve(start_year,end_year) values "player150"->"team213":(2018, 2019);
Execution succeeded (time spent 946/1091 us)
Wed, 01 Sep 2021 20:47:58 UTC

[root@wey wey.gu]# ~/.nebula-kind/bin/console -u root -p password --address=192.168.8.137 --port=32669
(root@nebula) [(none)]> show spaces
+-----+
| Name          |
+-----+
| "basketballplayer" |
+-----+
(root@nebula) [(none)]> use basketballplayer
(root@nebula) [basketballplayer]> show tags
+-----+
| Name          |
+-----+
| "player"      |
+-----+
| "team"        |
+-----+
```

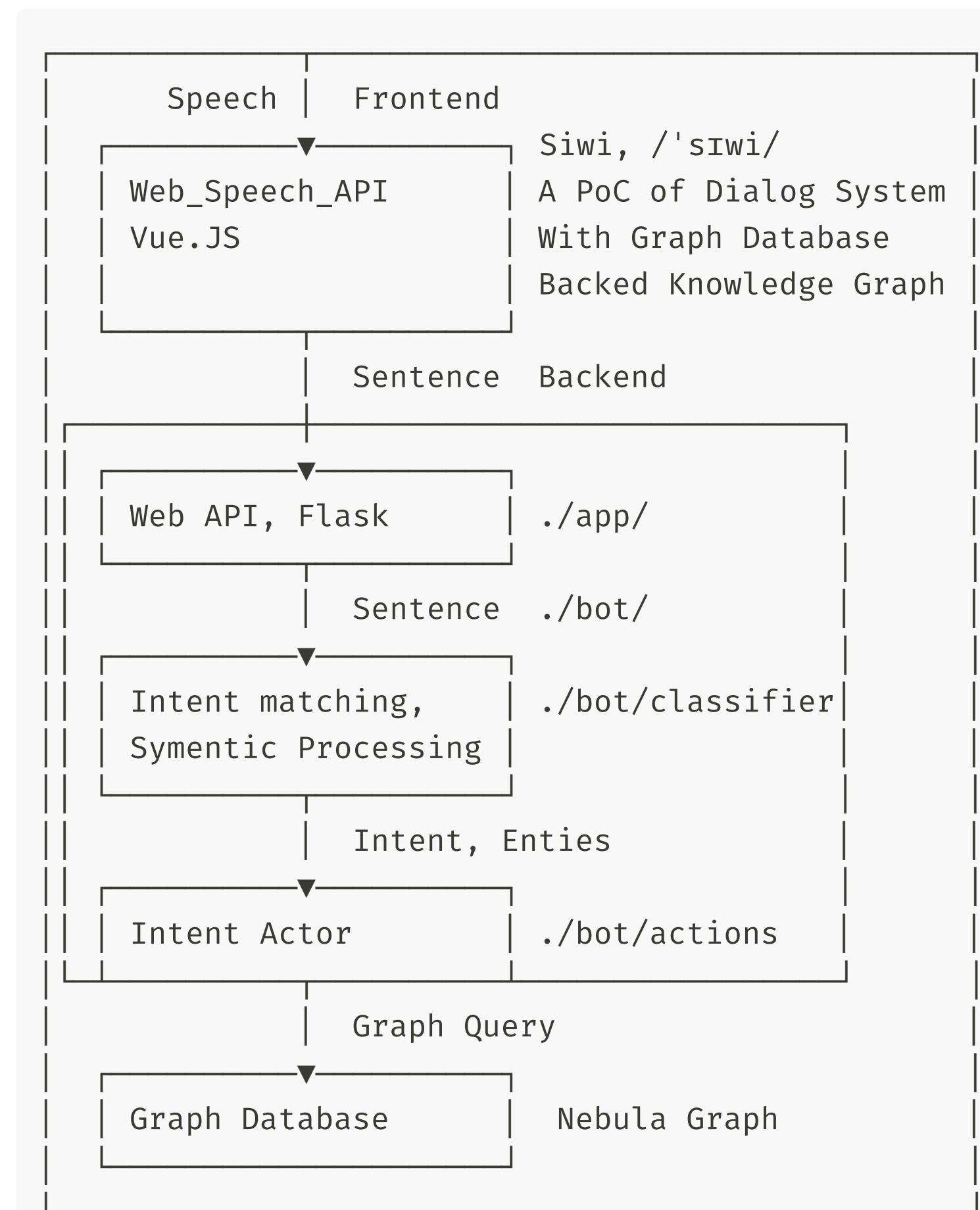


参考文档

# **Siwi on KubeSphere + OpenFunction**

Siwi (/ˈsɪwi/) is a PoC of Dialog System With Graph Database Backed Knowledge Graph.

## ARCH



## CODE

```
.
├── README.md
└── src
    ├── siwi
    │   ├── app
    │   └── bot
    │       ├── actions
    │       ├── bot
    │       ├── classifier
    │       │   └── test
    │       └── test
    └── siwi_frontend
        ├── README.md
        ├── package.json
        └── src
            ├── App.vue
            └── main.js
wsgi.py
```



wey-gu/nebula-siwi

# Live Demo

Siwi on KubeSphere + OpenFunction

# 回顾

- 图、图数据库简介
- Nebula Graph!
- Nebula Operator
- KubeSphere 上的图数据库
- KubeSphere 上的 OpenFunction
- Siwi, 一个基于 Nebula 的单一领域问答机器人
- Nebula-Siwi on FaaS on KubeSphere