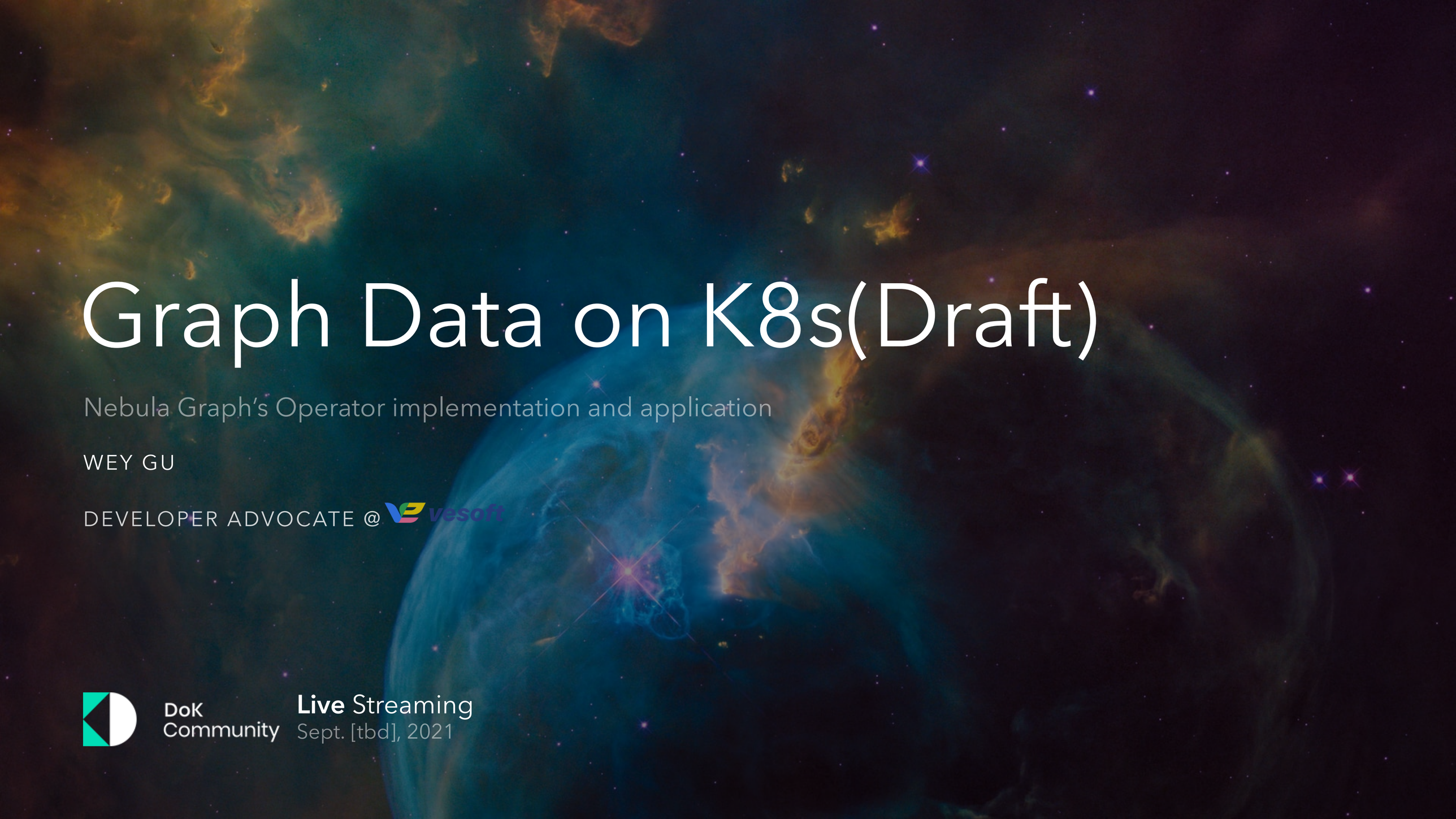


Graph Data on K8s(Draft)

Nebula Graph's Operator implementation and application

WEY GU

DEVELOPER ADVOCATE @ vesoft

DoK Community

Live Streaming

Sept. [tbd], 2021

Wey Gu (Siwei)

- Software Engineer @ Shanghai
- Open Source Believer
- Developer Advocate of Nebula Graph
- Ex-OpenStacker

wey-gu

wey_gu

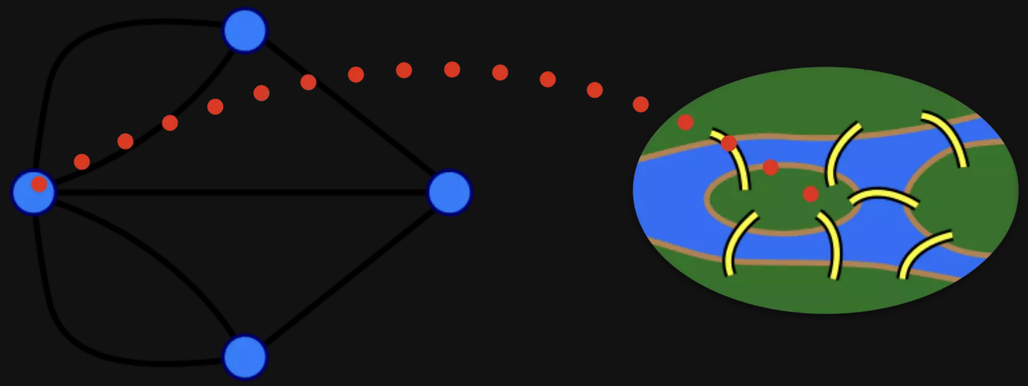
siwei.io/**about**

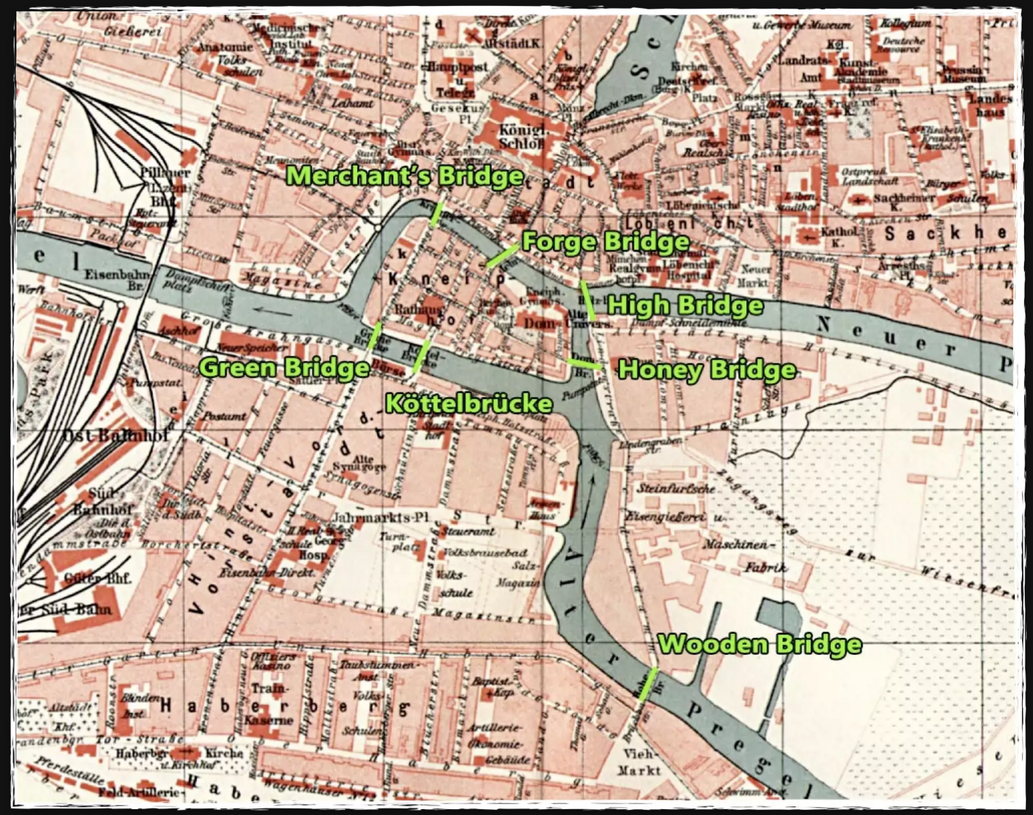
Graph Database

What is Graph? What is Graph DB? Why yet another DB?

DoK Community

What is Graph?





Map of Königsberg with the seven bridges labeled, circa 1905

NebulaGraph

DoK
Community

"A database that uses graph structures for semantic queries with nodes, edges, and properties to represent and store data

wikipedia.org/wiki/graph_database

More on what a GDB is

Why Yet Another DB?

RELATIONAL DB

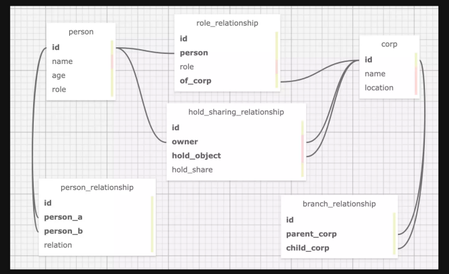
GRAPH DB

DoK
Community

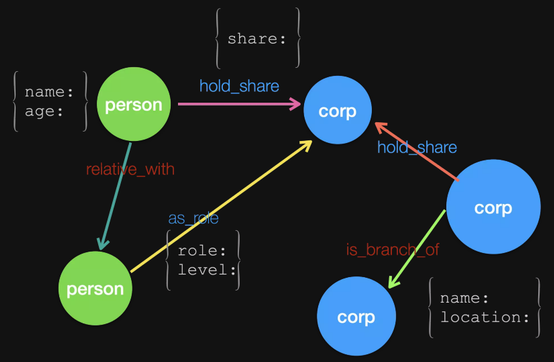
Why Yet Another DB?

Graph Schema

RELATIONAL DB



GRAPH DB



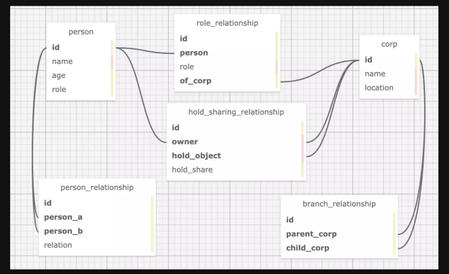
DoK Community

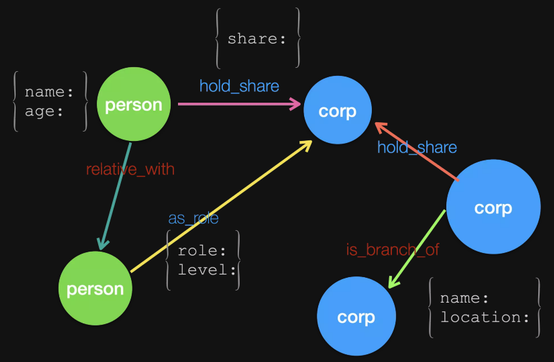
Why Yet Another DB?

RELATIONAL DB

GRAPH DB

Graph Schema





Graph Semantic Queries

```
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;
```

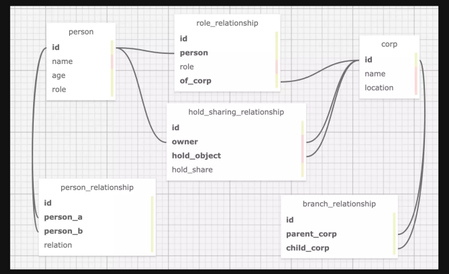
DoK Community

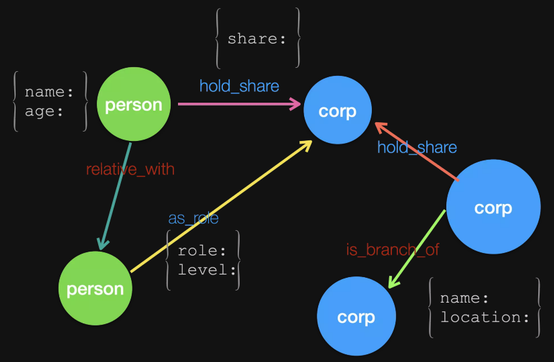
Why Yet Another DB?

RELATIONAL DB

GRAPH DB

Graph Schema





Graph Semantic Queries

```
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;
```

Performance

	Designed Scenario	2-hop latency (~2.5K)	3-hop latency (~110K)	4-hop latency (~600K)
Graph DB	Relationship Walk	0.01 sec	0.168 sec	1.36 sec
SQL DB	Information retrieval	0.016 sec	30 sec	1544 sec

DoK Community

Nebula Graph!

Features? Why Yet another Graph DB?

nebula-graph.io/about/**#why-nebula-graph**

DoK
Community

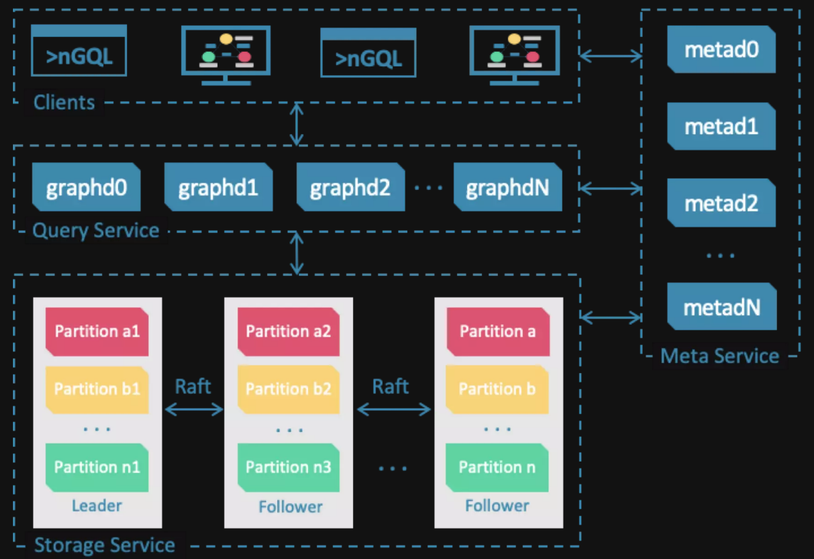
Nebula Graph Introduced

A highly performant linearly scalable graph DB designed in a shared-nothing distributed model. It has been proven to beat the performance of competing graph databases by multiple times over in production.

DoK
Community

Nebula Graph Introduced

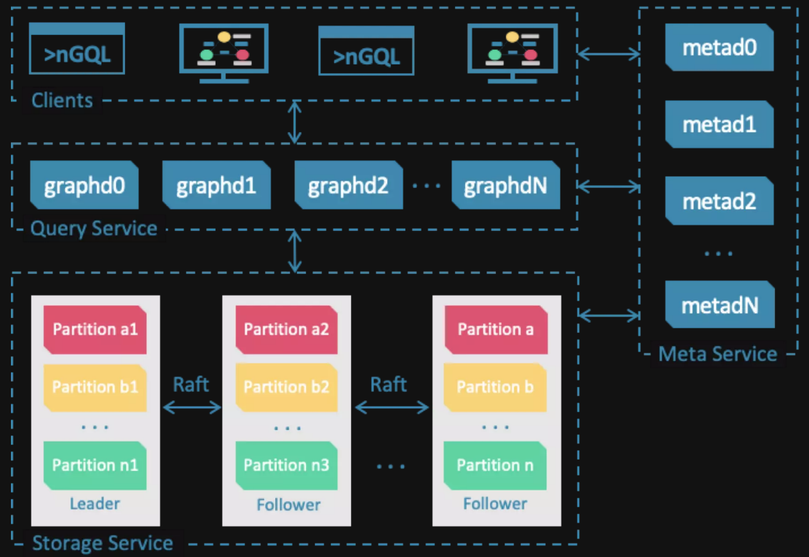
A highly performant linearly scalable graph DB designed in a shared-nothing distributed model. It has been proven to beat the performance of competing graph databases by multiple times over in production.



DoK Community

Nebula Graph Introduced

A highly performant linearly scalable graph DB designed in a shared-nothing distributed model. It has been proven to beat the performance of competing graph databases by multiple times over in production.



Nebula Arch.

Use Cases

京东数科 · 美团 · 有赞 · 小红书 · WiFi万能钥匙

Tencent 腾讯 · WeBank 微众银行 · vivo · 企查查 · 快手

知乎 · 360数科 · SUNING 苏宁科技 · BOSS直聘 · 网易

Seeyii 视野金服 · 思知 · 半云科技 · 同盾科技 · 携程金融

akulaku · 合合信息 · 中国移动 · oppo

微信 · 众安保险 · 网易游戏 · Soul · ziroom自如

HUAWEI

DoK Community

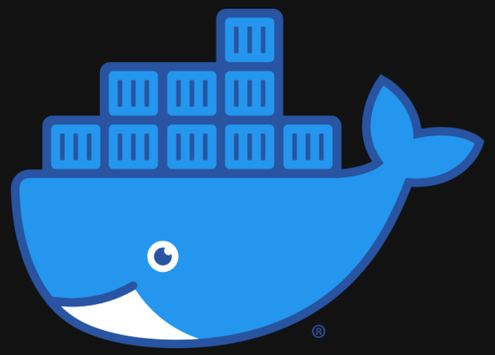
DoK
Community

GDB in the Cloud Native Era

Containerization evolution

Containerization evolution

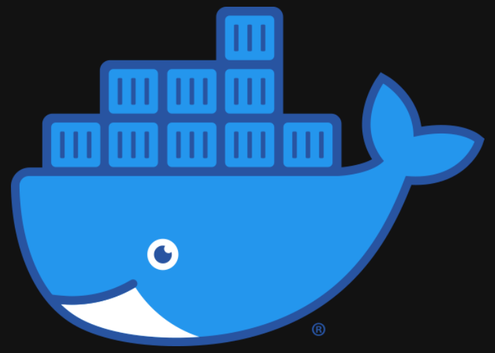
NEBULA DOCKER



DoK
Community

Containerization evolution

NEBULA DOCKER

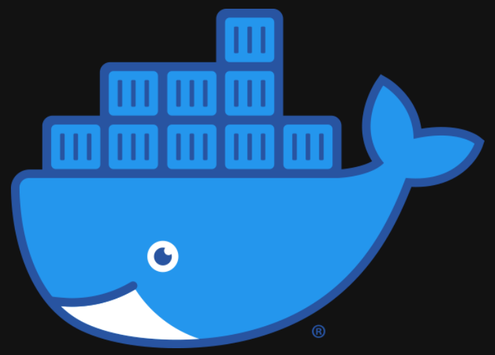


NEBULA K8S

DoK
Community

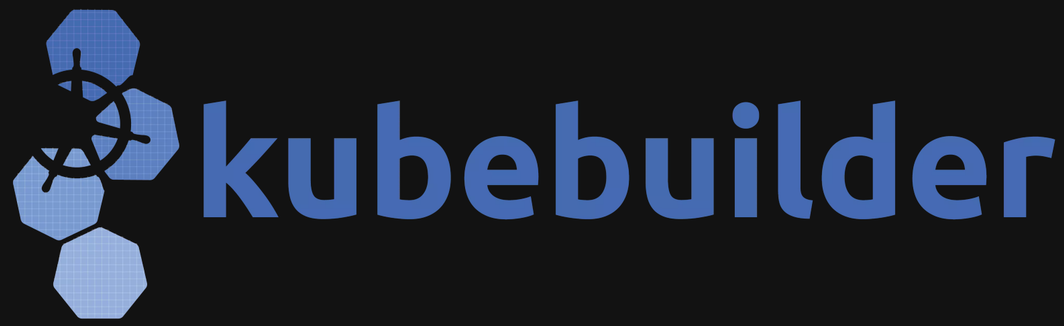
Containerization evolution

NEBULA DOCKER



NEBULA K8S

NEBULA OPERATOR



Nebula Operator Implementation

KUBEBUILDER SCAFFOLD

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

vesoft-inc/**nebula-operator**

Nebula Operator Implementation

KUBEBUILDER SCAFFOLD

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

CRD

```yaml
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**

Nebula Operator Implementation

KUBEBUILDER SCAFFOLD

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

CRD

```yaml
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
```

CONTROL LOOP

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

vesoft-inc/**nebula-operator**

Nebula Operator Implementation

KUBEBUILDER SCAFFOLD

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

CRD

```yaml
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 Implementation

KUBEBUILDER SCAFFOLD

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

CRD

```yaml
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
```

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
```

vesoft-inc/**nebula-operator**

Nebula Operator Implementation

KUBEBUILDER SCAFFOLD

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

CRD

```yaml
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 Implementation

KUBEBUILDER SCAFFOLD

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

CRD

```yaml
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

```go
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 Implementation

KUBEBUILDER SCAFFOLD

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

CRD

```yaml
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

```go
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

roadmap of **nebula-operator**

Nebula Operator Roadmap

- Rolling Upgrade
- Auto Scaling
- Integration with other Services

roadmap of **nebula-operator**

Nebula Operator Roadmap

- Rolling Upgrade
- Auto Scaling
- Integration with other Services

Check out our Github Repo and contribute!

vesoft-inc/**nebula-operator**

roadmap of **nebula-operator**

Nebula Operator Roadmap

- Rolling Upgrade
- Auto Scaling
- Integration with other Services

Check out our Github Repo and contribute!

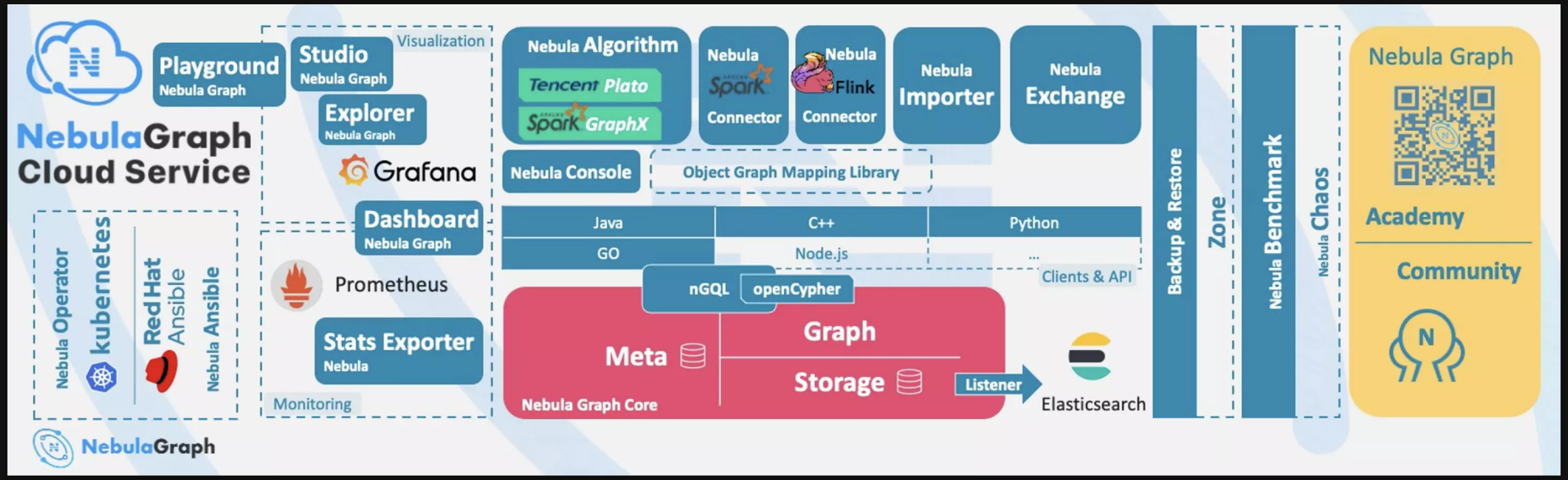
vesoft-inc/**nebula-operator**

roadmap of **nebula-operator**

Nebula Landscape

Nebula Community is rich in ecology and still expanding and exploring, welcome to join and contribute!

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



vesoft-inc/**nebula**

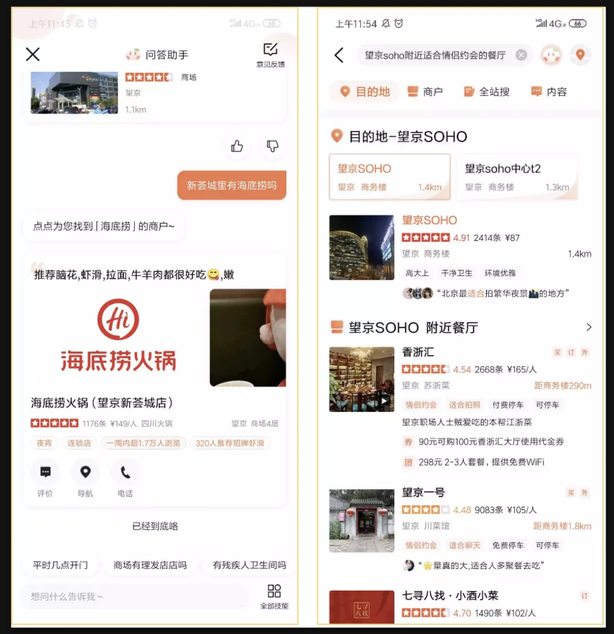
DoK Community

Hands on GraphDB

Nebula on K8s - Demo

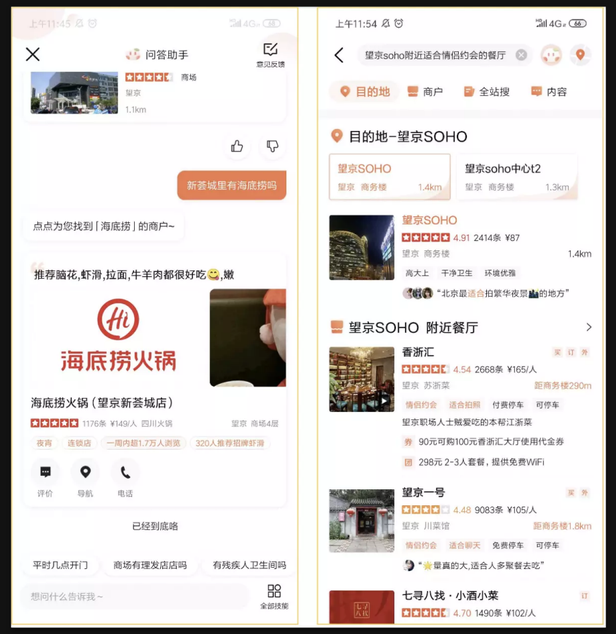
Use Case Explained

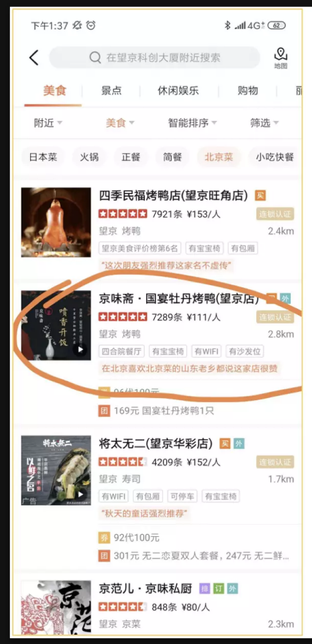
Use Case Explained

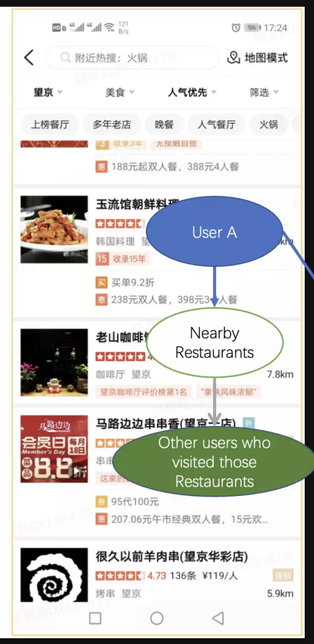


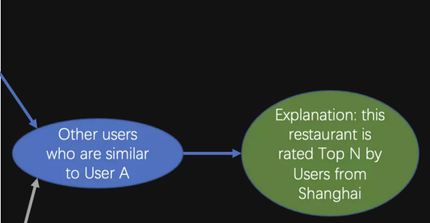
DoK Community

Use Case Explained



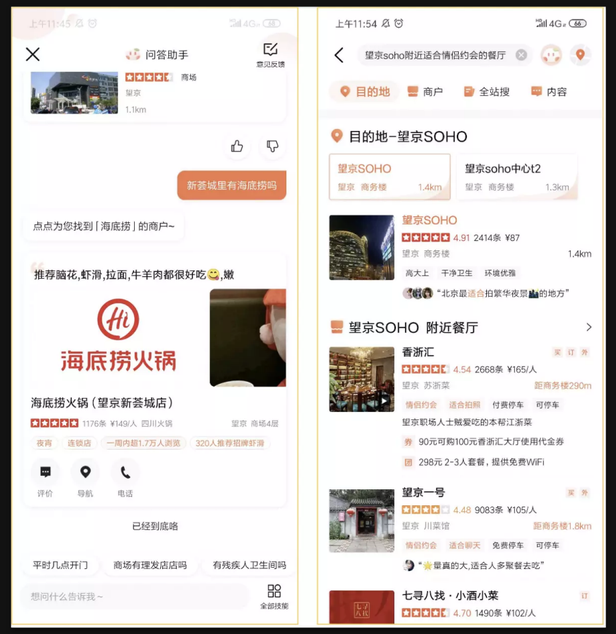


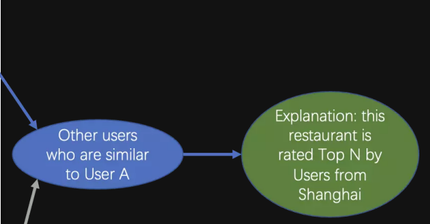




DoK Community

Use Case Explained

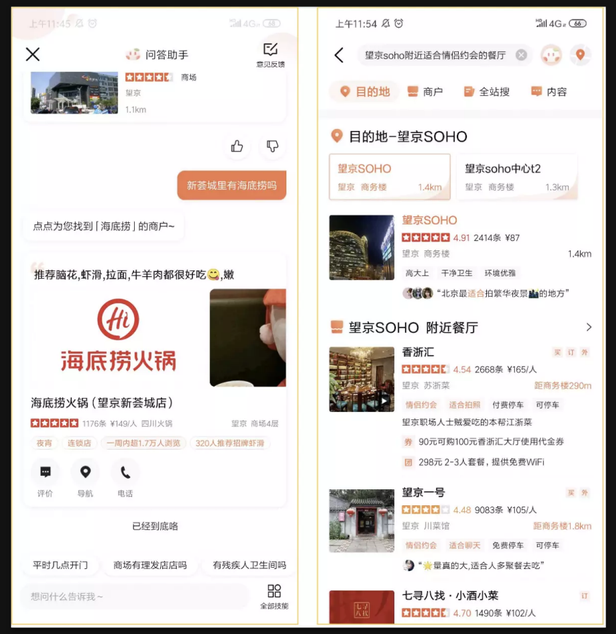


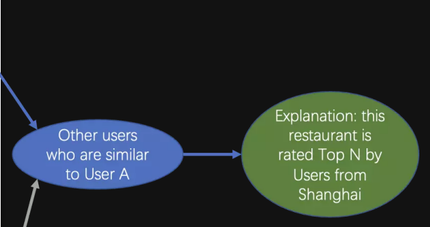


tech.meituan.com/2021/04/01/nebula-graph-practice-in-meituan.html

DoK Community

Use Case Explained





SNS	Risk Control	Public Security	Knowledge Graph	ML
Biopharmacy	IoT	Blockchain	Data Lineage	AIOps

tech.meituan.com/2021/04/01/nebula-graph-practice-in-meituan.html

DoK Community

KinD + Nebula Graph

```
curl -sL nebula-kind.siwei.io/install.sh | bash
```

```
$ 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 distr
+-------------------------------------------------------------------------+------+----------+--------------+-------------
| "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            |
+-------------------------------------------------------------------------+------+----------+--------------+-------------
```

KinD + Nebula Graph

```
curl -sL nebula-kind.siwei.io/install.sh | bash
```

```
$ 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 distr
+-------------------------------------------------------------------------+------+----------+--------------+-------------
| "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            |
+-------------------------------------------------------------------------+------+----------+--------------+-------------
```

KinD + Nebula Graph

```
curl -sL nebula-kind.siwei.io/install.sh | bash
```

```
$ 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 distr
+-------------------------------------------------------------------+------+----------+--------------+-------------
| "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            |
+-------------------------------------------------------------------+------+----------+--------------+-------------

Got 4 rows (time spent 2510/2955 us)
```

Nebula Graph Data Import

```
$ 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"   |
+----------+
```

Doc: **Sample Dataset**

Nebula Graph Data Import

```
$ 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"   |
+----------+
```

Doc: **Sample Dataset**

Nebula Graph Data Import

```
$ 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"   |
+----------+
```

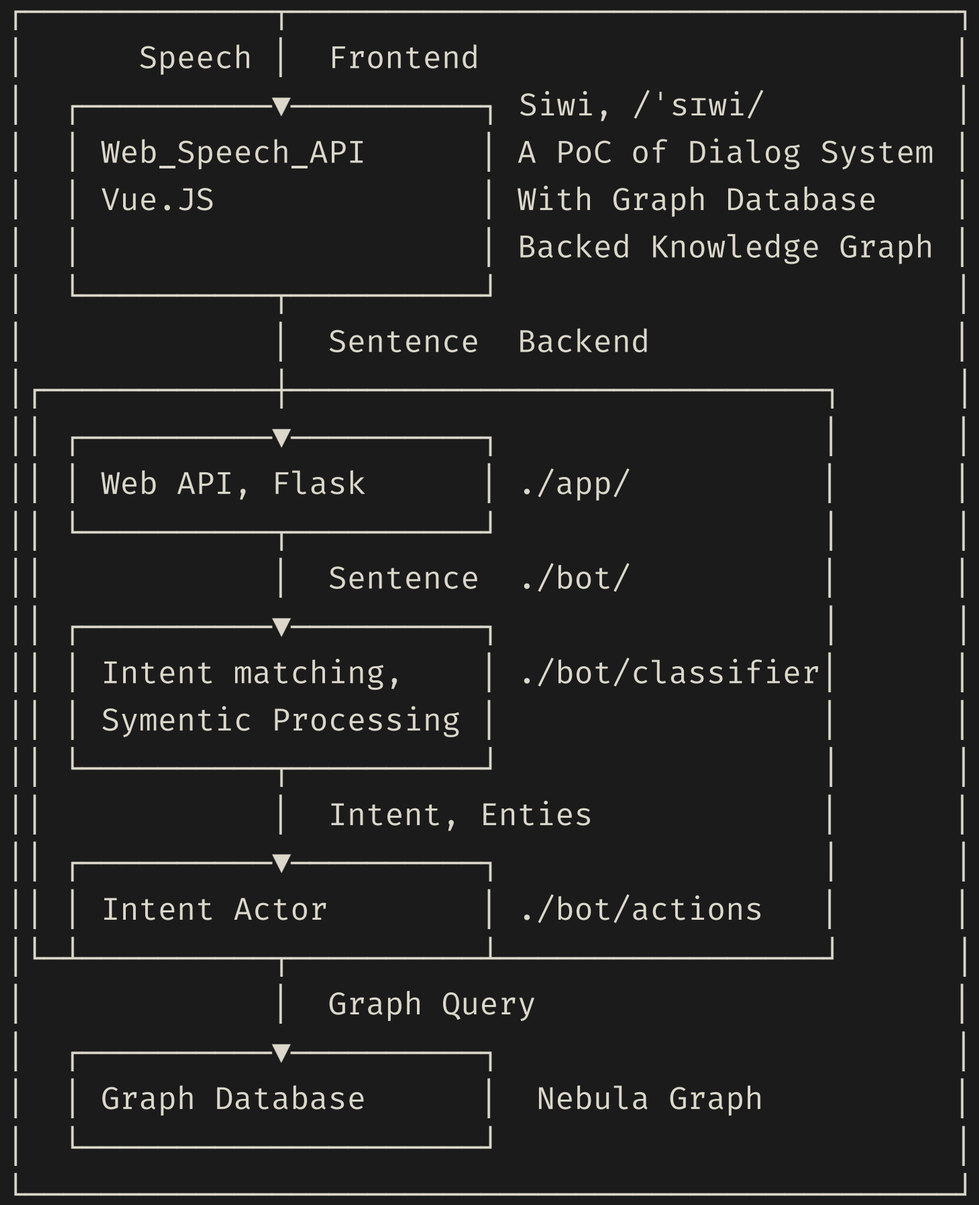
Doc: **Sample Dataset**

Siwi, the voice agent

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

DoK
Community

ARCH

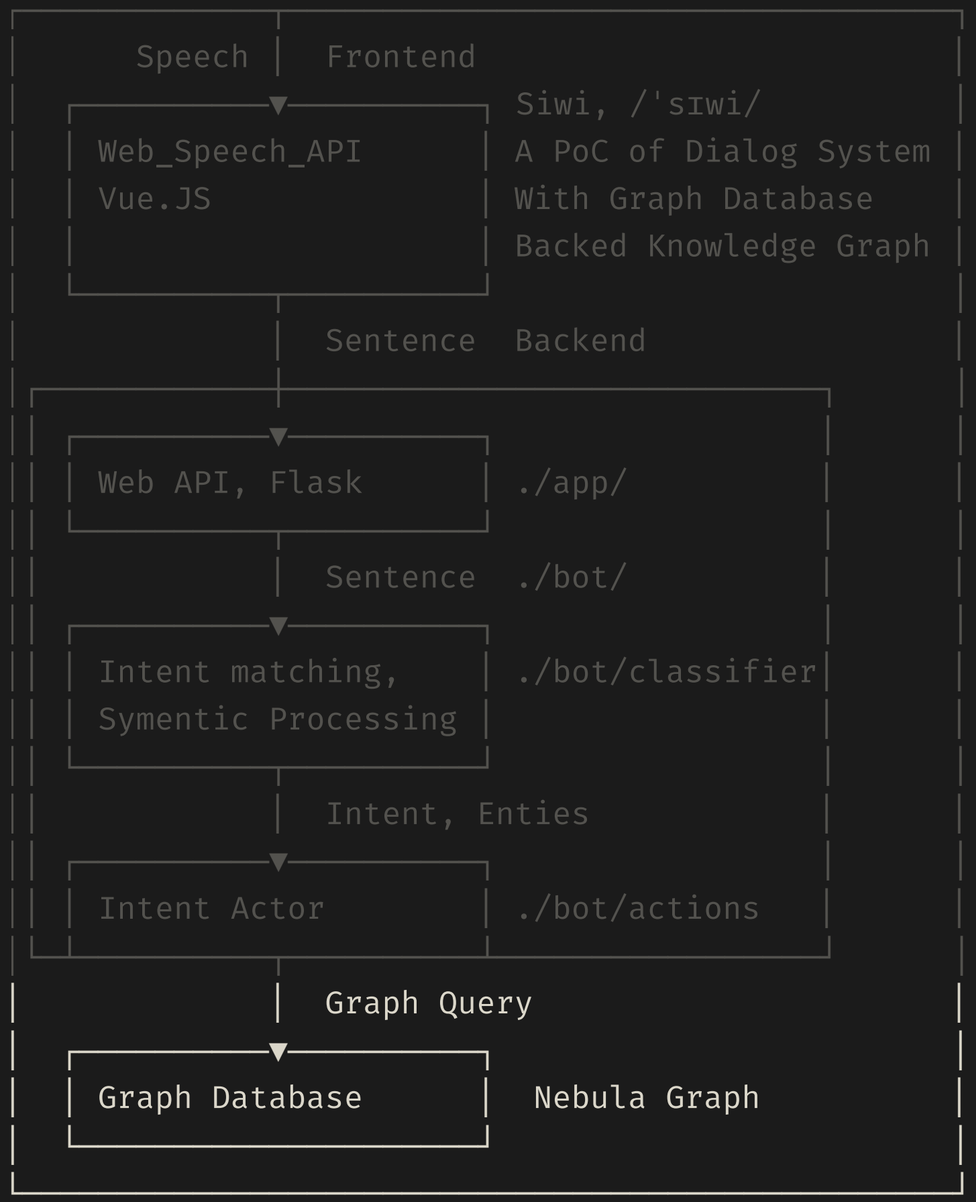


CODE

```
.
├── README.md
├── src
│   ├── siwi                # Siwi-API Backend
│   │   ├── app             # Web Server, take HTTP req > ca
│   │   └── bot             # Bot API
│   │       ├── actions     # Take Intent, Slots, Query KG h
│   │       ├── bot         # Entrypoint of the Bot API
│   │       ├── classifier # Symentic Parse, Intent Match,
│   │       └── test        # Example Data as equivalent/moc
│   └── siwi_frontend       # Browser End
│       ├── README.md
│       ├── package.json
│       └── src
│           ├── App.vue     # Listen to user and pass Qs to
│           └── main.js
└── wsgi.py
```

wey-gu/**nebula-siwi**

ARCH

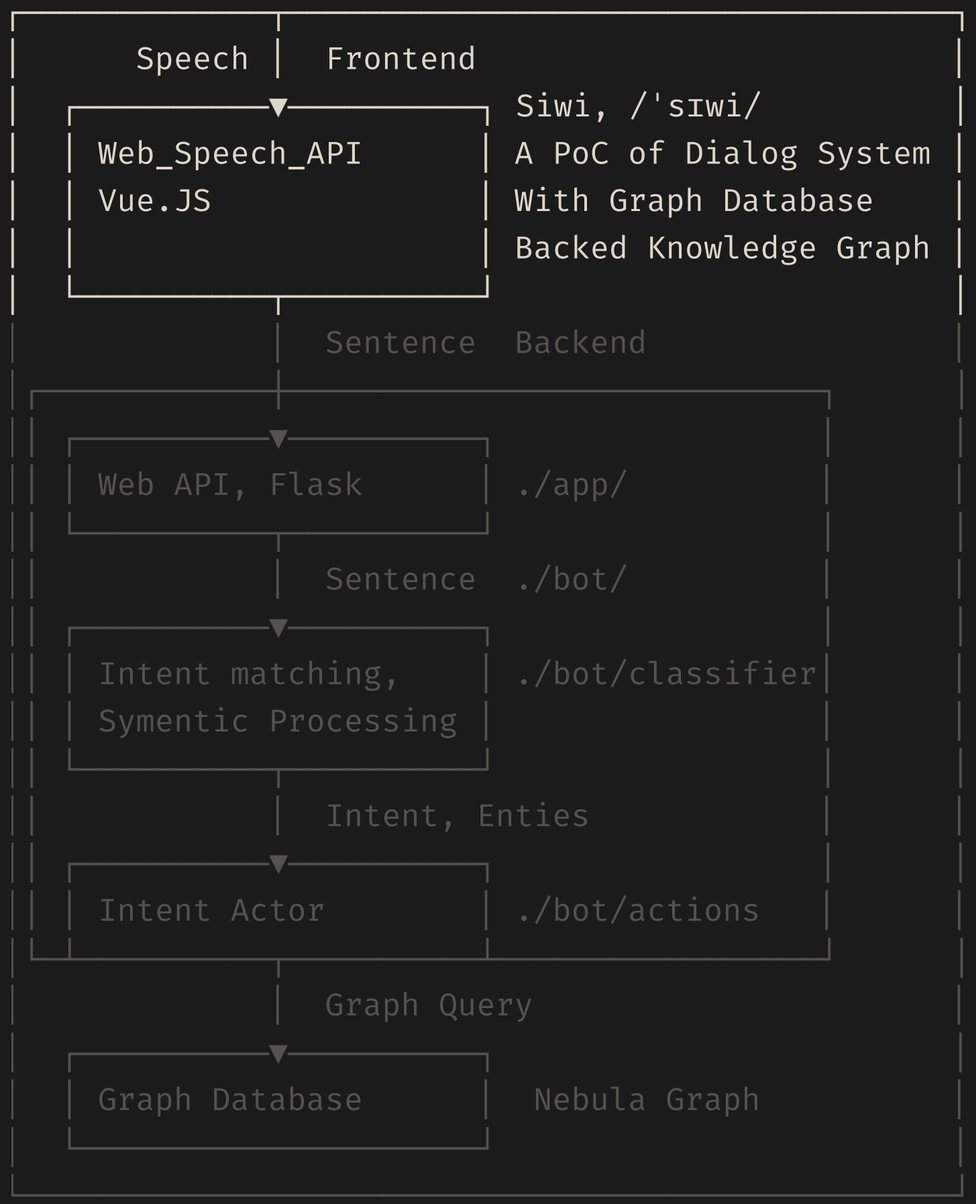


CODE

```
.
├── README.md
├── src
│   ├── siwi                # Siwi-API Backend
│   │   ├── app             # Web Server, take HTTP req > ca
│   │   └── bot             # Bot API
│   │       ├── actions     # Take Intent, Slots, Query KG h
│   │       ├── bot         # Entrypoint of the Bot API
│   │       ├── classifier # Symentic Parse, Intent Match,
│   │       └── test        # Example Data as equivalent/moc
│   └── siwi_frontend       # Browser End
│       ├── README.md
│       ├── package.json
│       └── src
│           ├── App.vue     # Listen to user and pass Qs to
│           └── main.js
└── wsgi.py
```

wey-gu/**nebula-siwi**

ARCH

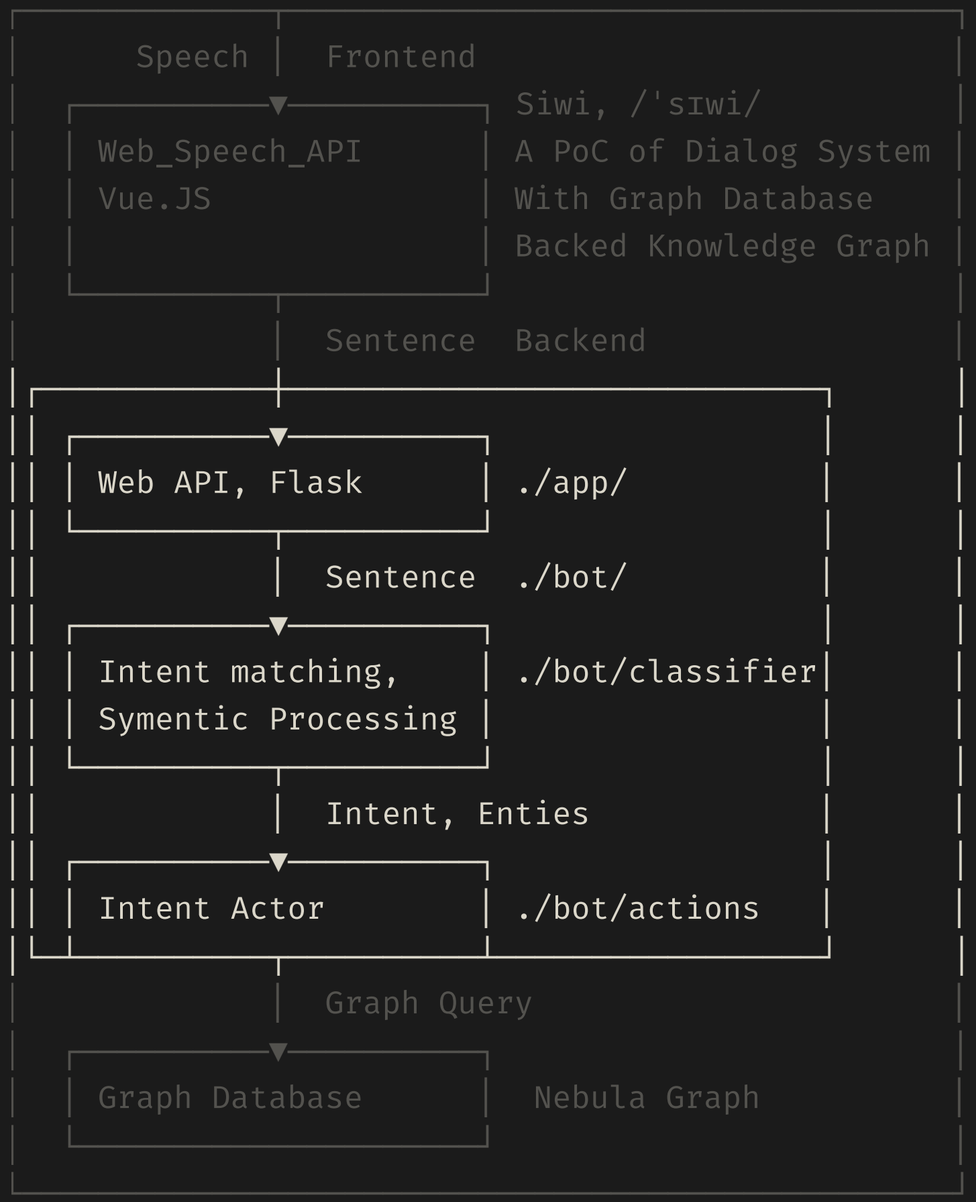


CODE

```
.
├── README.md
├── src
│   ├── siwi                # Siwi-API Backend
│   │   ├── app             # Web Server, take HTTP req > ca
│   │   └── bot             # Bot API
│   │       ├── actions     # Take Intent, Slots, Query KG h
│   │       ├── bot         # Entrypoint of the Bot API
│   │       ├── classifier # Symentic Parse, Intent Match,
│   │       └── test        # Example Data as equivalent/moc
│   └── siwi_frontend       # Browser End
│       ├── README.md
│       ├── package.json
│       └── src
│           ├── App.vue     # Listen to user and pass Qs to
│           └── main.js
└── wsgi.py
```

wey-gu/**nebula-siwi**

ARCH

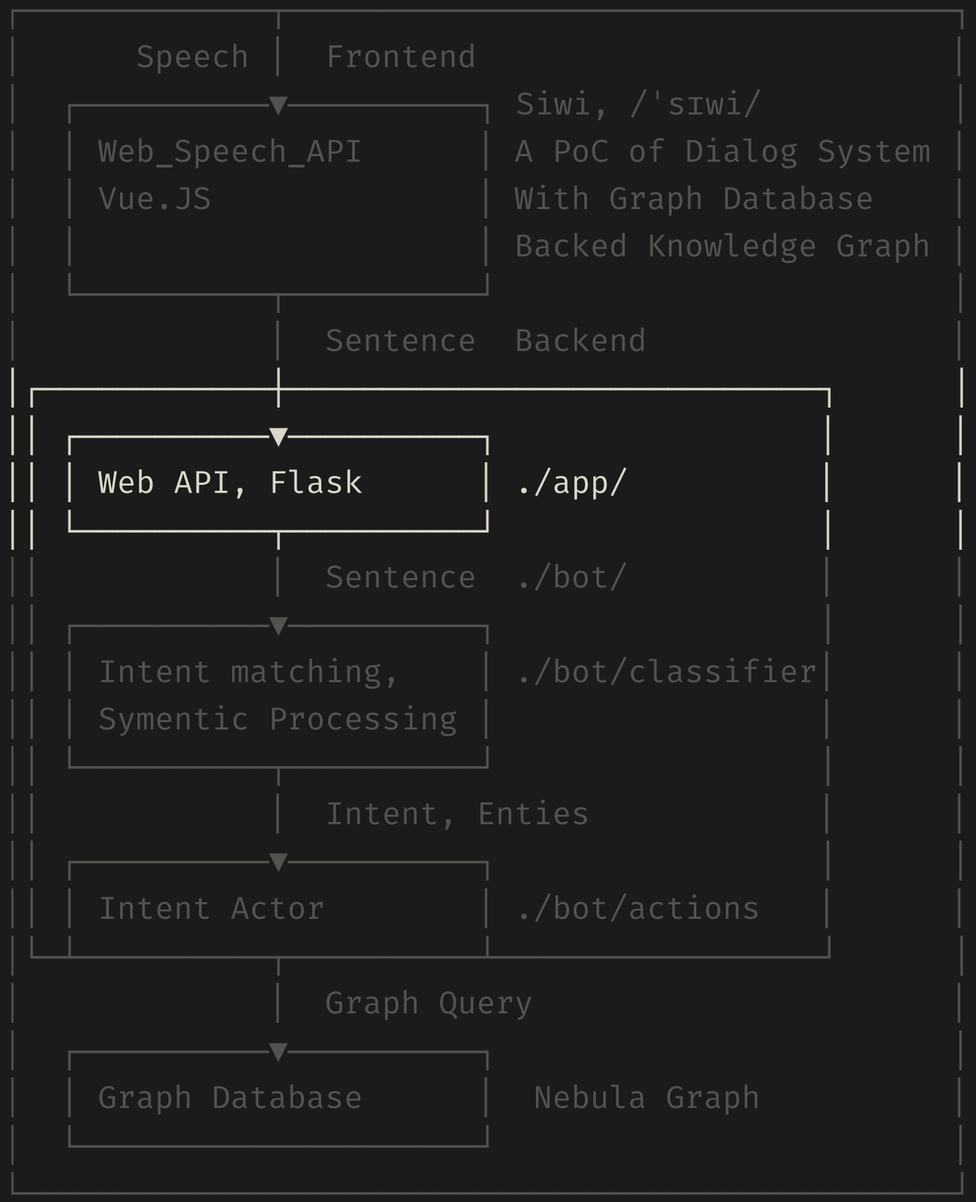


CODE

```
.
├── README.md
├── src
│   ├── siwi                  # Siwi-API Backend
│   │   ├── app               # Web Server, take HTTP req > ca
│   │   └── bot               # Bot API
│   │       ├── actions       # Take Intent, Slots, Query KG h
│   │       ├── bot           # Entrypoint of the Bot API
│   │       ├── classifier    # Symentic Parse, Intent Match,
│   │       └── test          # Example Data as equivalent/moc
│   └── siwi_frontend         # Browser End
│       ├── README.md
│       ├── package.json
│       └── src
│           ├── App.vue       # Listen to user and pass Qs to
│           └── main.js
└── wsgi.py
```

wey-gu/**nebula-siwi**

ARCH

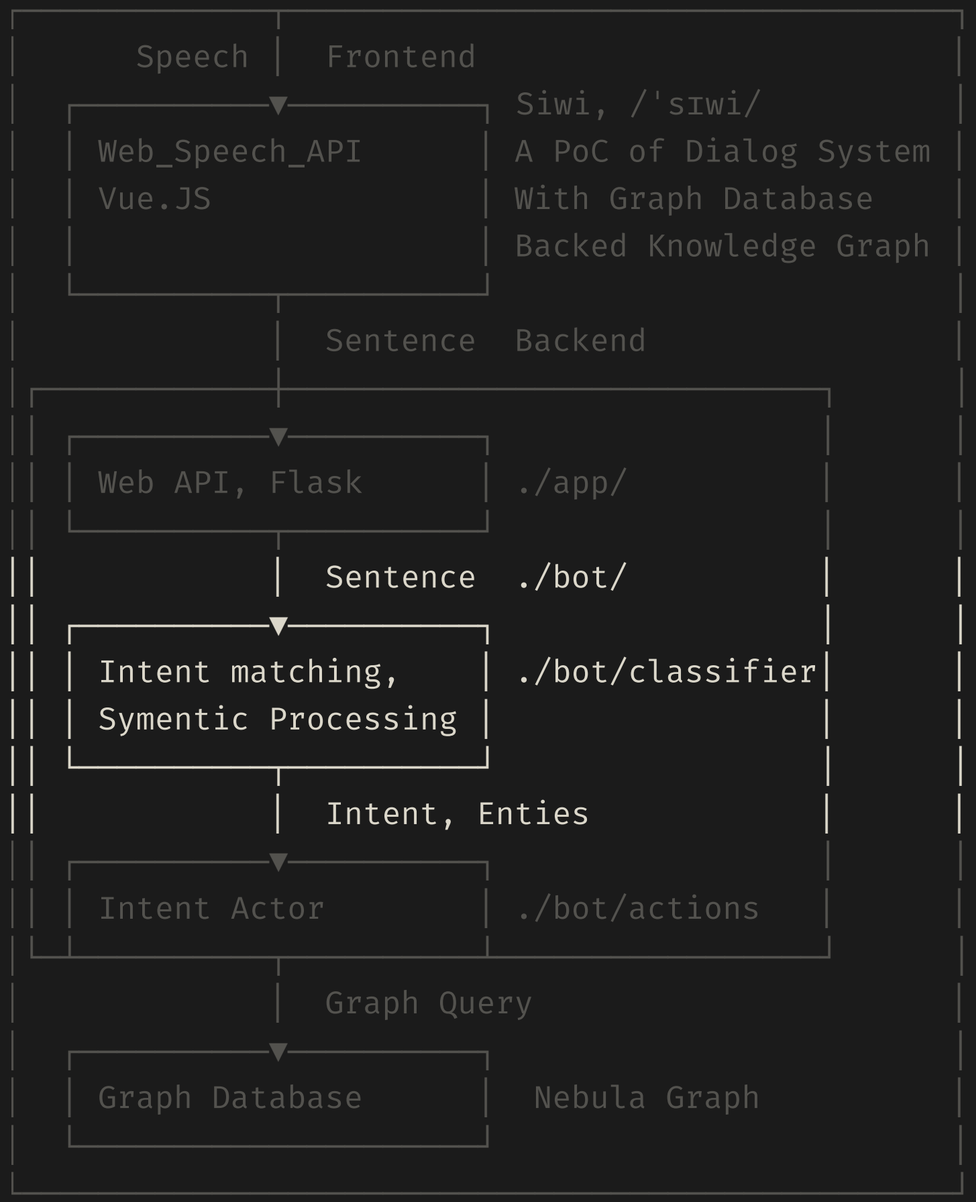


CODE

```
.
├── README.md
├── src
│   ├── siwi                  # Siwi-API Backend
│   │   ├── app               # Web Server, take HTTP req > ca
│   │   └── bot               # Bot API
│   │       ├── actions       # Take Intent, Slots, Query KG h
│   │       ├── bot           # Entrypoint of the Bot API
│   │       ├── classifier    # Symentic Parse, Intent Match,
│   │       └── test          # Example Data as equivalent/moc
│   └── siwi_frontend         # Browser End
│       ├── README.md
│       ├── package.json
│       └── src
│           ├── App.vue       # Listen to user and pass Qs to
│           └── main.js
└── wsgi.py
```

wey-gu/**nebula-siwi**

ARCH

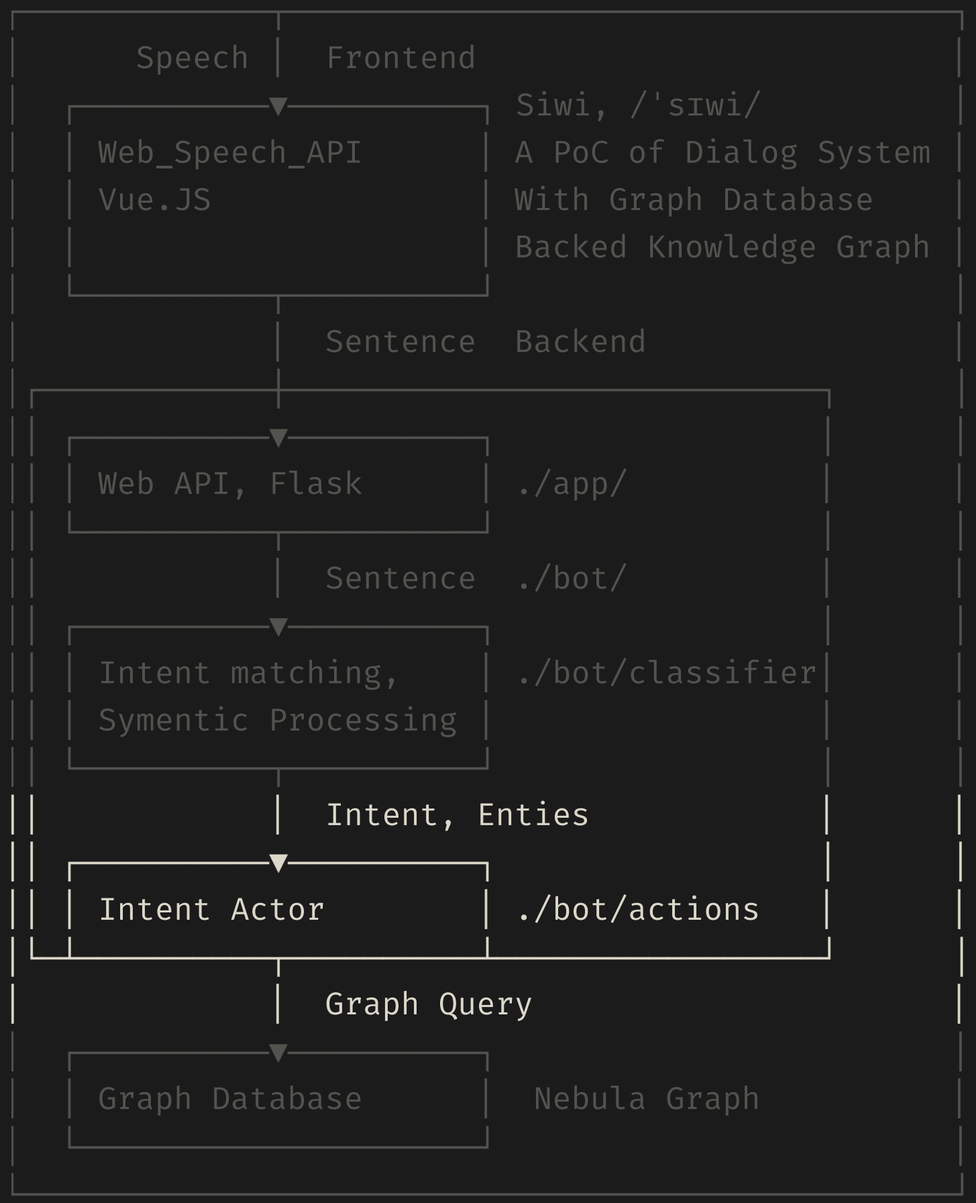


CODE

```
.
├── README.md
├── src
│   ├── siwi                  # Siwi-API Backend
│   │   ├── app               # Web Server, take HTTP req > ca
│   │   └── bot               # Bot API
│   │       ├── actions       # Take Intent, Slots, Query KG h
│   │       ├── bot           # Entrypoint of the Bot API
│   │       ├── classifier    # Symentic Parse, Intent Match,
│   │       └── test          # Example Data as equivalent/moc
│   └── siwi_frontend         # Browser End
│       ├── README.md
│       ├── package.json
│       └── src
│           ├── App.vue       # Listen to user and pass Qs to
│           └── main.js
└── wsgi.py
```

wey-gu/**nebula-siwi**

ARCH

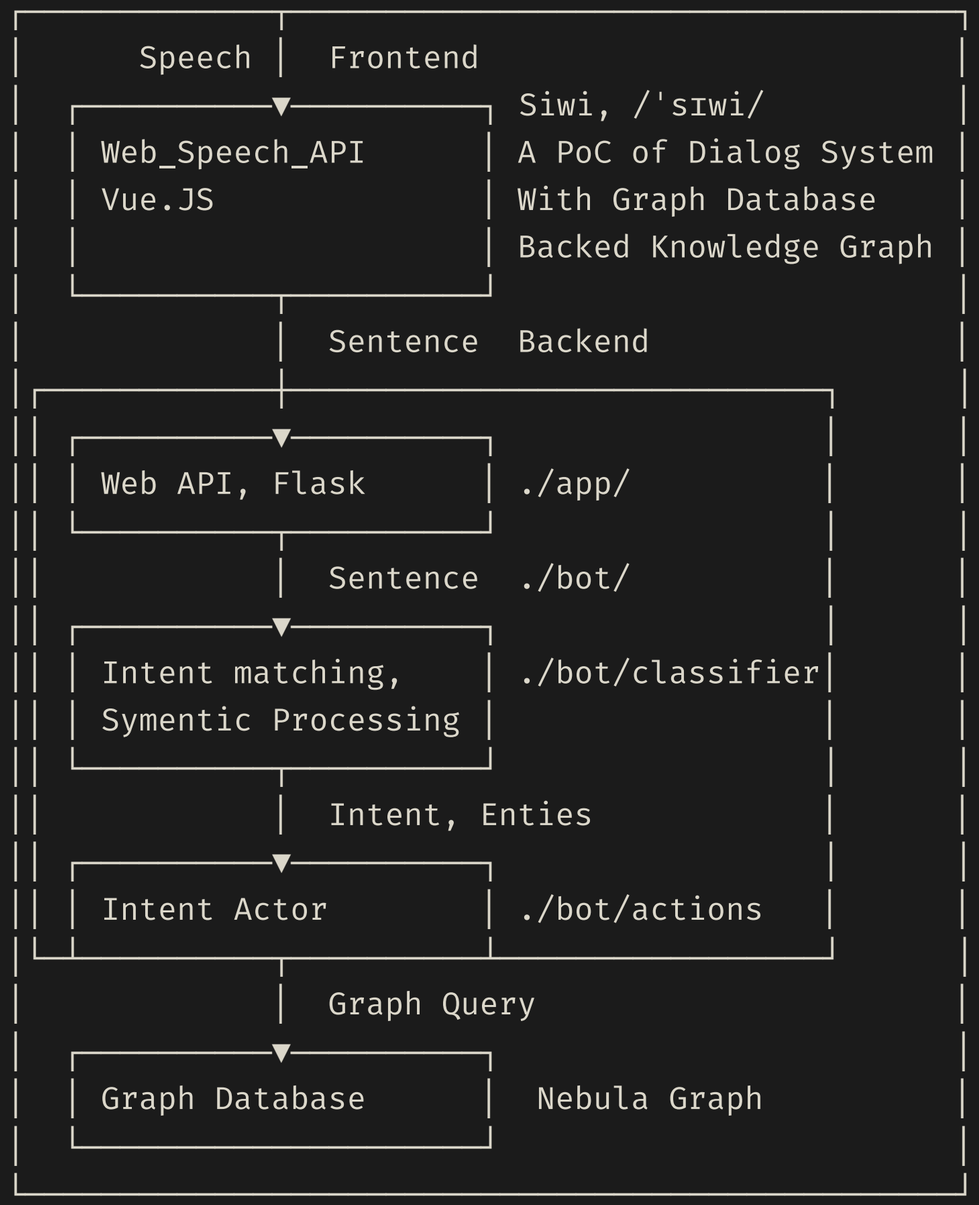


CODE

```
.
├── README.md
├── src
│   ├── siwi                  # Siwi-API Backend
│   │   ├── app               # Web Server, take HTTP req > ca
│   │   └── bot               # Bot API
│   │       ├── actions       # Take Intent, Slots, Query KG h
│   │       ├── bot           # Entrypoint of the Bot API
│   │       ├── classifier    # Symentic Parse, Intent Match,
│   │       └── test          # Example Data as equivalent/moc
│   └── siwi_frontend         # Browser End
│       ├── README.md
│       ├── package.json
│       └── src
│           ├── App.vue       # Listen to user and pass Qs to
│           └── main.js
└── wsgi.py
```

wey-gu/**nebula-siwi**

ARCH



CODE

```
.
├── README.md
├── src
│   ├── siwi                # Siwi-API Backend
│   │   ├── app             # Web Server, take HTTP req > ca
│   │   └── bot             # Bot API
│   │       ├── actions     # Take Intent, Slots, Query KG h
│   │       ├── bot         # Entrypoint of the Bot API
│   │       ├── classifier # Symentic Parse, Intent Match,
│   │       └── test        # Example Data as equivalent/moc
│   └── siwi_frontend       # Browser End
│       ├── README.md
│       ├── package.json
│       └── src
│           ├── App.vue     # Listen to user and pass Qs to
│           └── main.js
└── wsgi.py
```

wey-gu/**nebula-siwi**

Live Demo

Siwi on K8s

katacoda.com/wey/scenarios/**siwi-kgqa**

Recap

- Graph, Graph DB introduced
- Nebula Graph!
- Nebula Operator
- GDB on K8s
- Siwi, a KBQA system based on GDB
- Nebula-Siwi on FaaS on KubeSphere

DoK Community

Live Streaming
Sept. [tbd], 2021

Recap

- Graph, Graph DB introduced
- Nebula Graph!
- Nebula Operator
- GDB on K8s
- Siwi, a KBQA system based on GDB
- Nebula-Siwi on FaaS on KubeSphere

wey-gu

wey_gu

siwei.io

DoK Community

Live Streaming

Sept. [tbd], 2021