PyCon China 2021

2021 中国 PYTHON 开发者大会

2021.10.16-17

图数据库 Python 的应用实践

Python 在你不能错过的图技术领域上手分享

古思为

DEVELOPER ADVOCATE @ vesoft

PyCon China

Oct. 16-17, 2021

Wey Gu (古思为)

I build things with Magic, and Scale the Magic to help others

- 👨‍💻 Software Engineer @ Shanghai
- 🏴‍☠️ Open Source Believer
- 🥑 Developer Advocate of Nebula Graph
- 🐍 Ex-OpenStacker (Python)

wey-gu

wey_gu

siwei.io/**about**

Overview

- ❓ Why Graph Database?
- 🤖 Siwi, a Knowledge Graph Dialog System in Python
- 🕵️ Corp-Rel Search, a Corproration Investment Relation Search System in Python

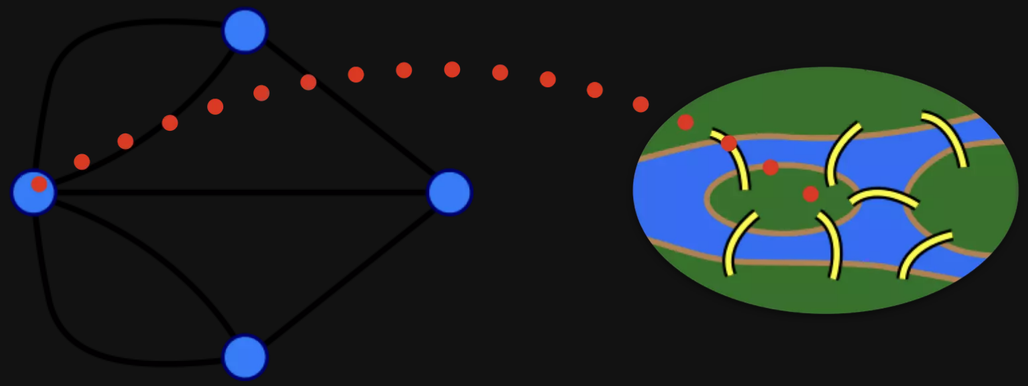
siwei.io/talks/2021-PyCon

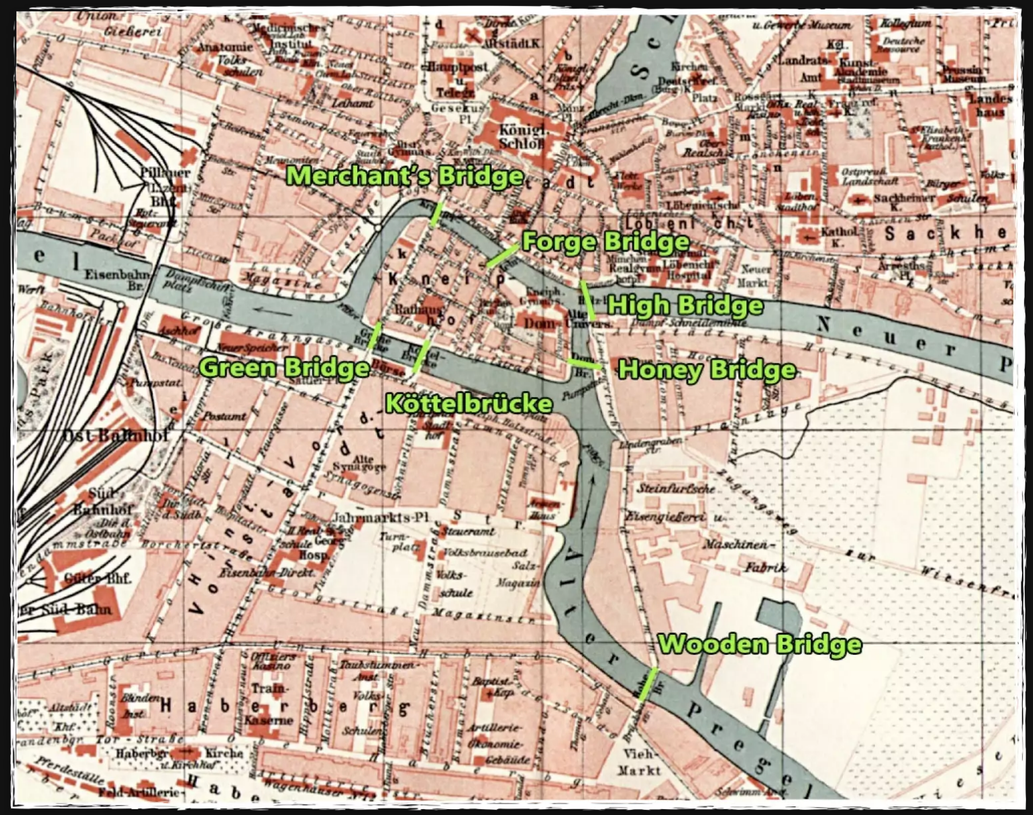
PyCon China
Oct. 16-17, 2021

Graph Database

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

What is Graph?





Map of Königsberg with the seven bridges labeled, circa 1905

NebulaGraph

"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?

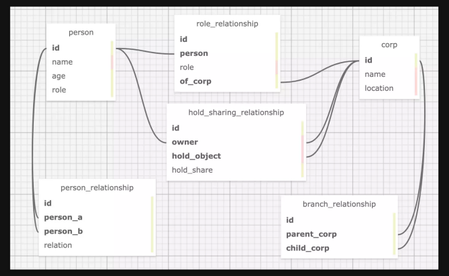
Relational DB

Graph DB

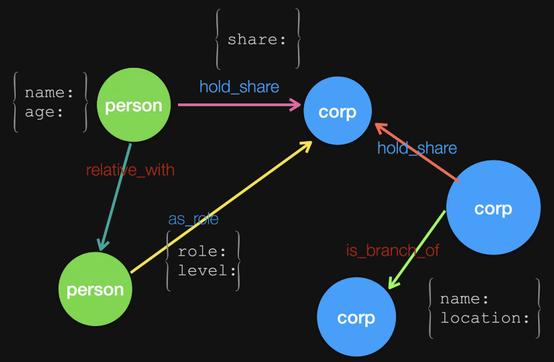
Why Yet Another DB?

Relational DB

Graph Schema



Graph DB

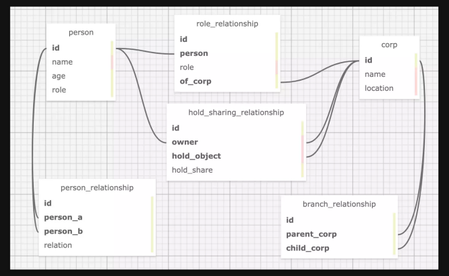


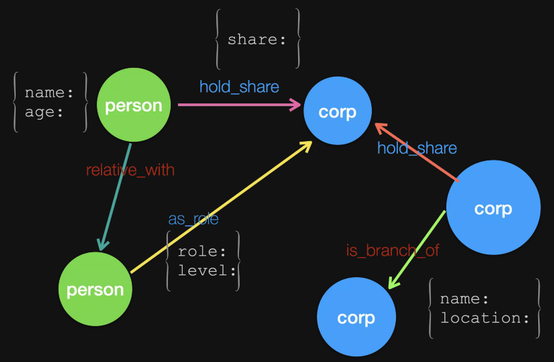
Why Yet Another DB?

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')
```

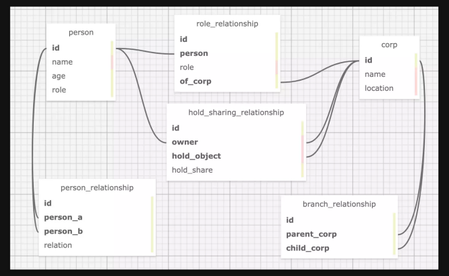
```
GO FROM 100 OVER serve YIELD serve._dst AS Team | \
GO FROM $-.Team OVER serve REVERSELY YIELD $$.player.name;
```

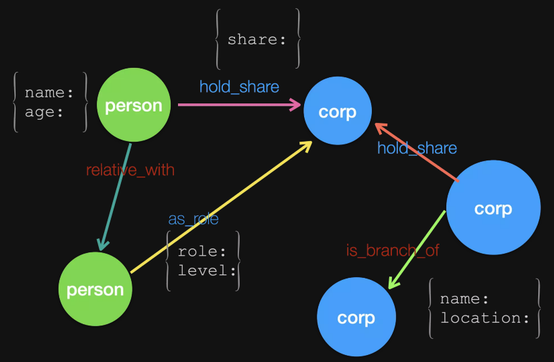
Why Yet Another DB?

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

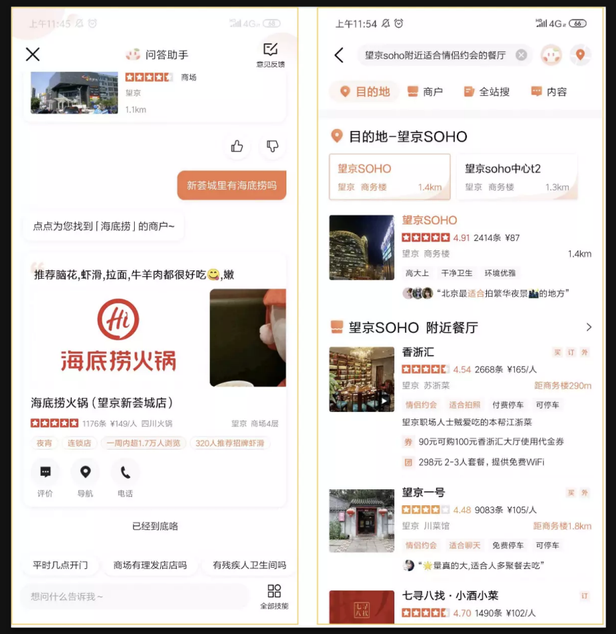
	Designed Scenario	2-hop latency (~2.5K)	3-hop latency (~110K)	4-hop latency (~600K)
Graph DB	Relationship Walk	0.01 sec	0.168 sec	1.36 sec
SQL DB	Information retrieval	0.016 sec	30 sec	1544 sec

GDB in Action with Python

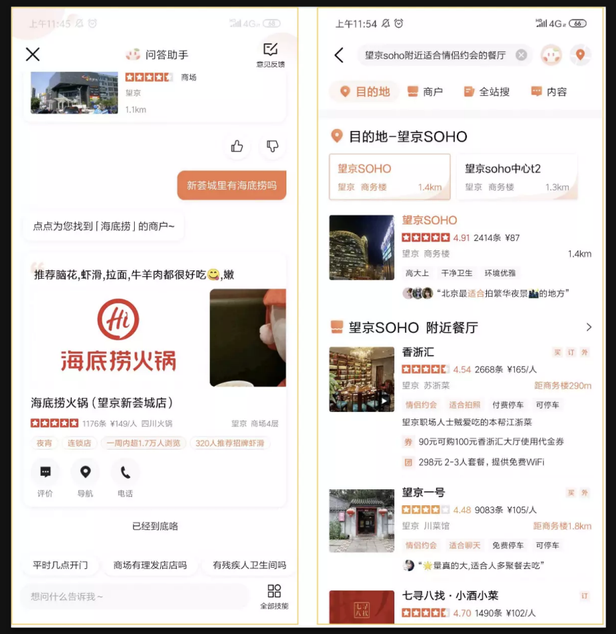
Nebula Graph + Python - Demo

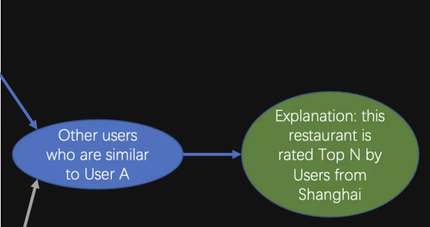
Use Case Explained

Use Case Explained

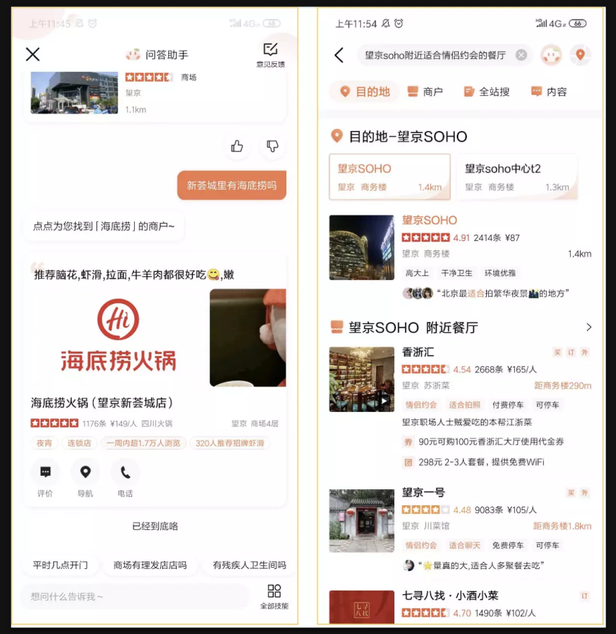


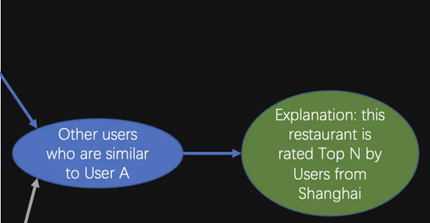
Use Case Explained





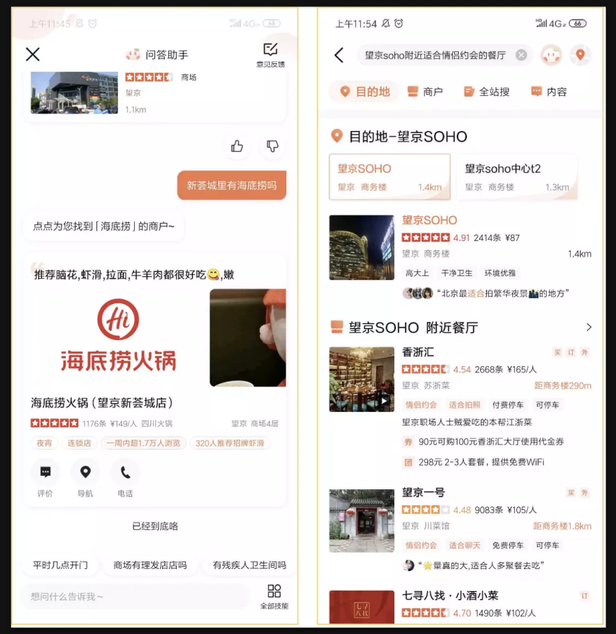
Use Case Explained





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

Use Case Explained



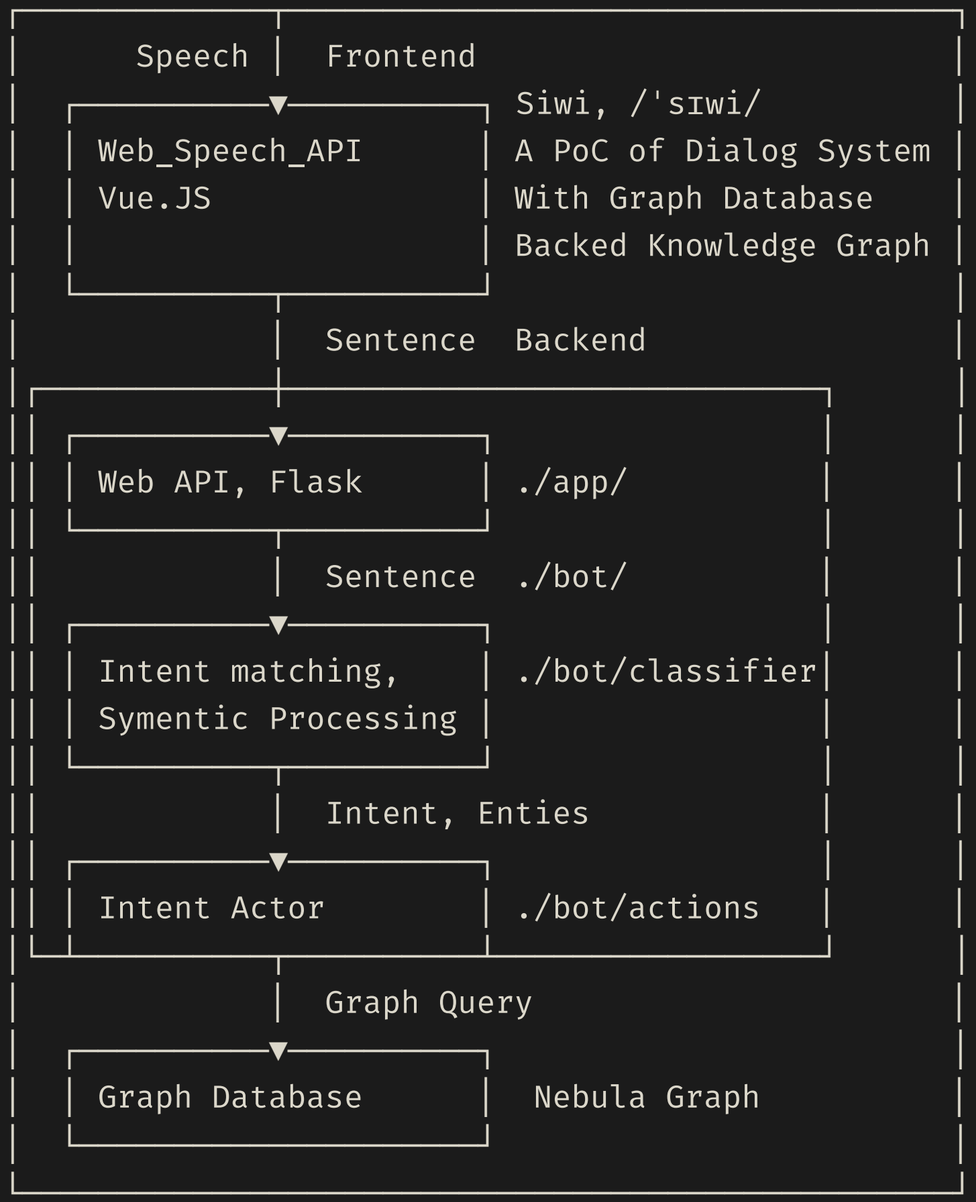
SNS	Risk Control	Public Security	Knowledge Graph	ML
Biopharmacy	IoT	Blockchain	Data Lineage	AIOps

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

Siwi, the voice agent

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

Arch of Siwi

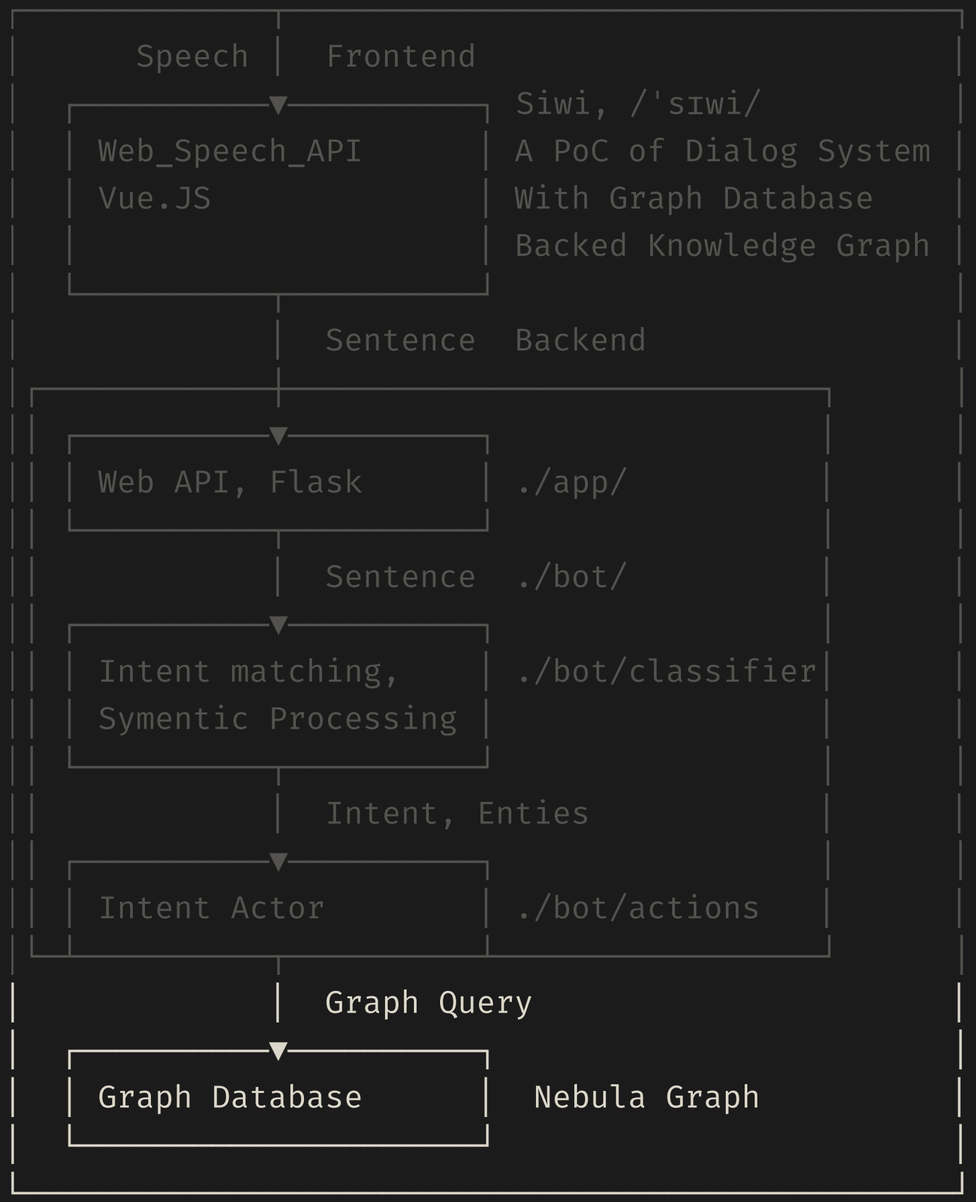


Code

```
.
├── 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 of Siwi

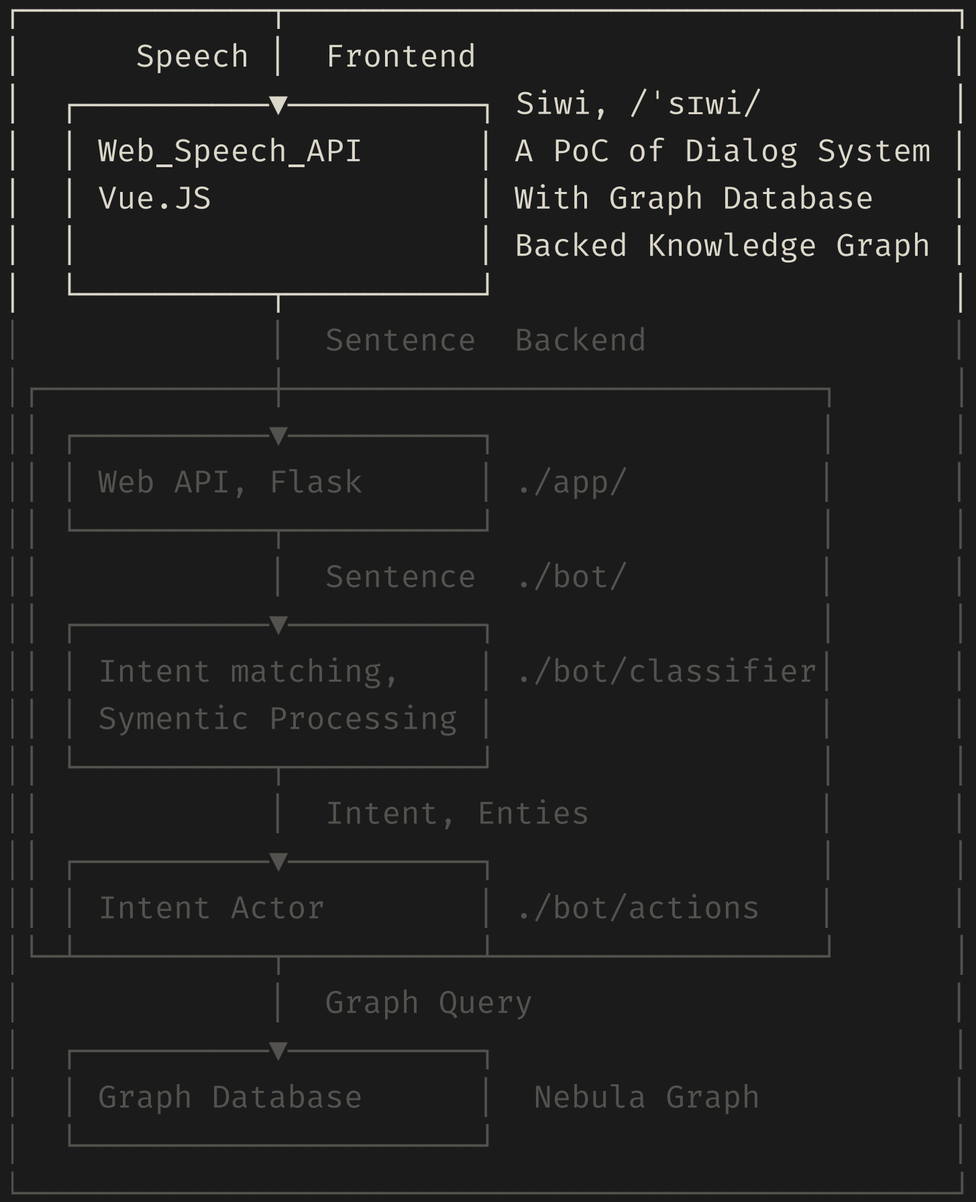


Code

```
.
├── 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 of Siwi

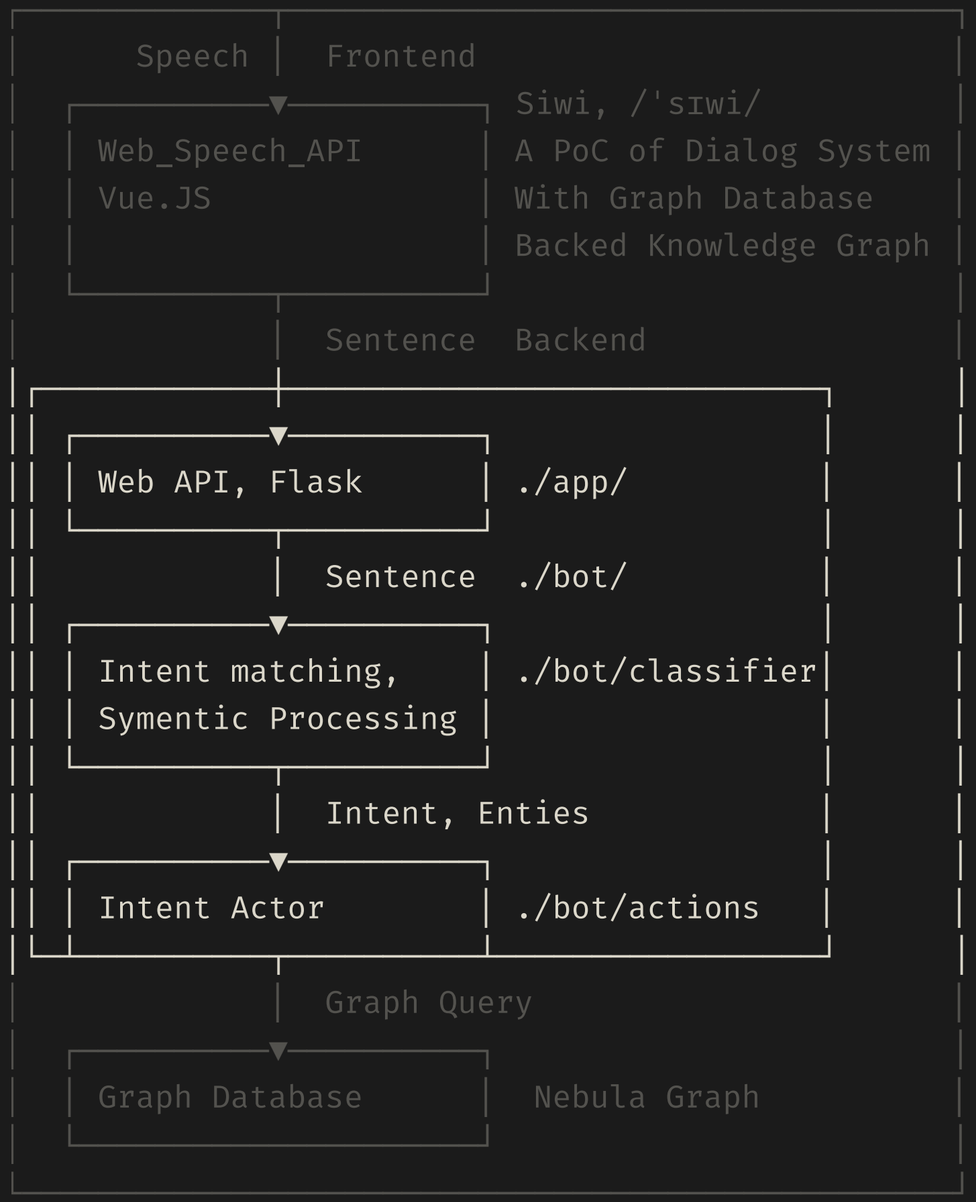


Code

```
.
├── 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 of Siwi

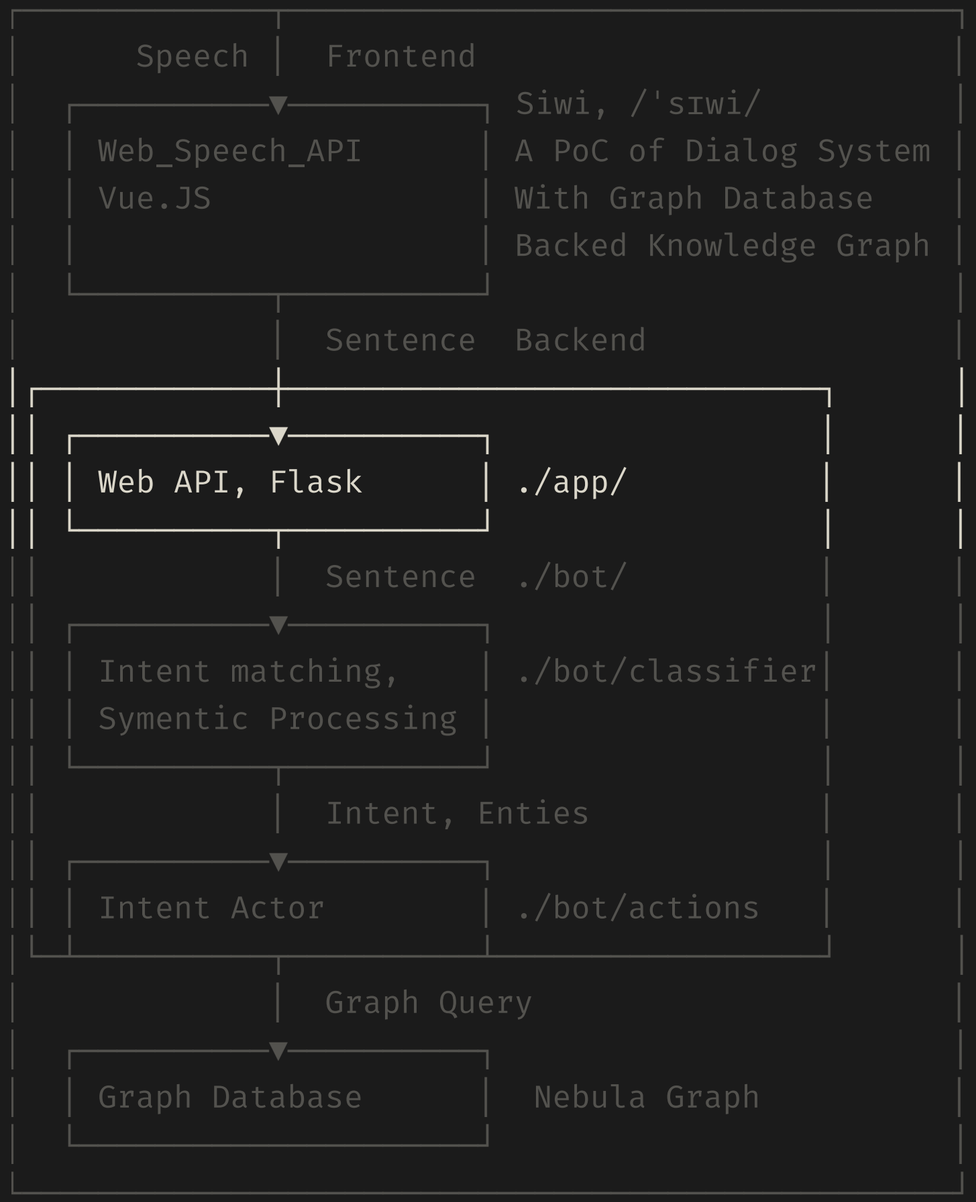


Code

```
.
├── 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 of Siwi

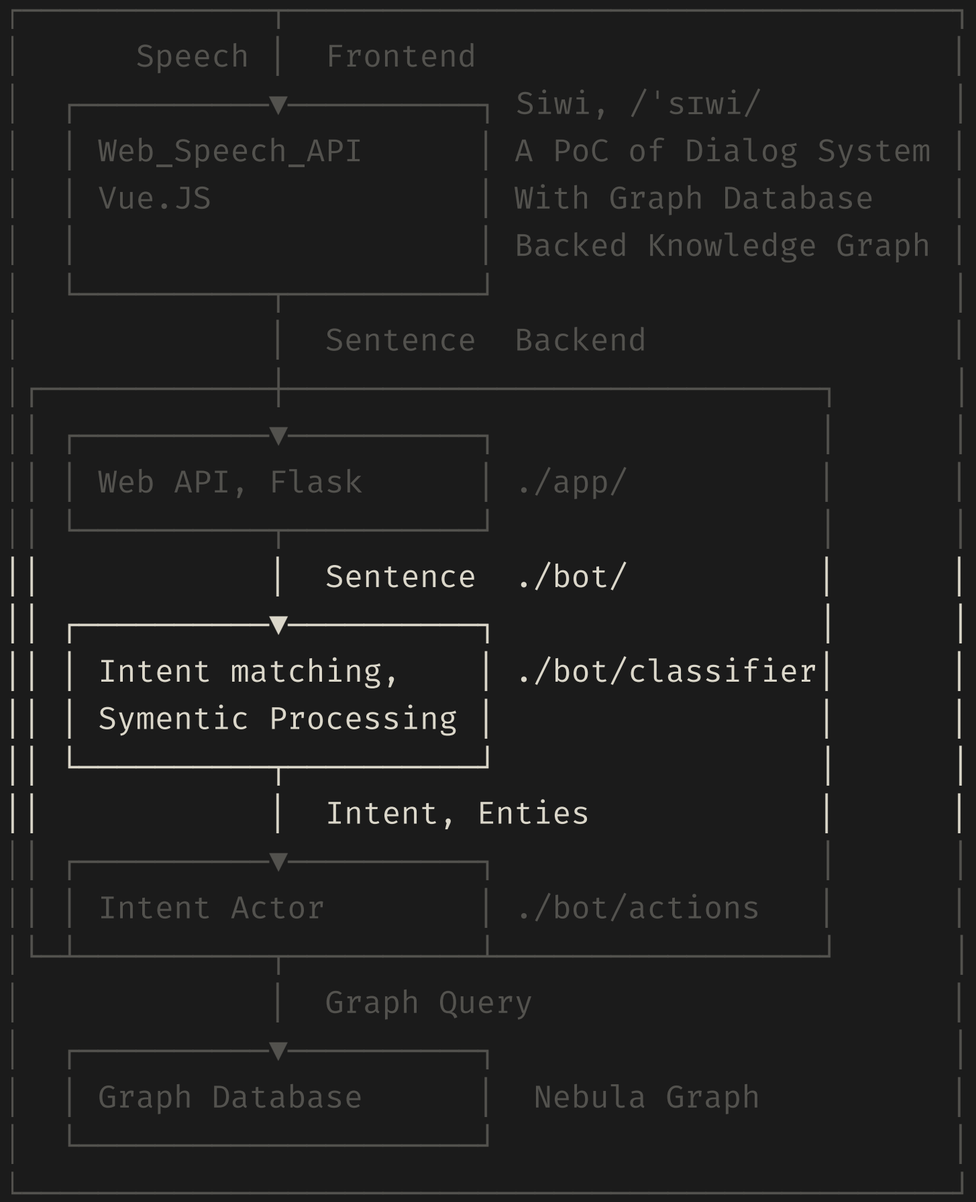


Code

```
.
├── 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 of Siwi

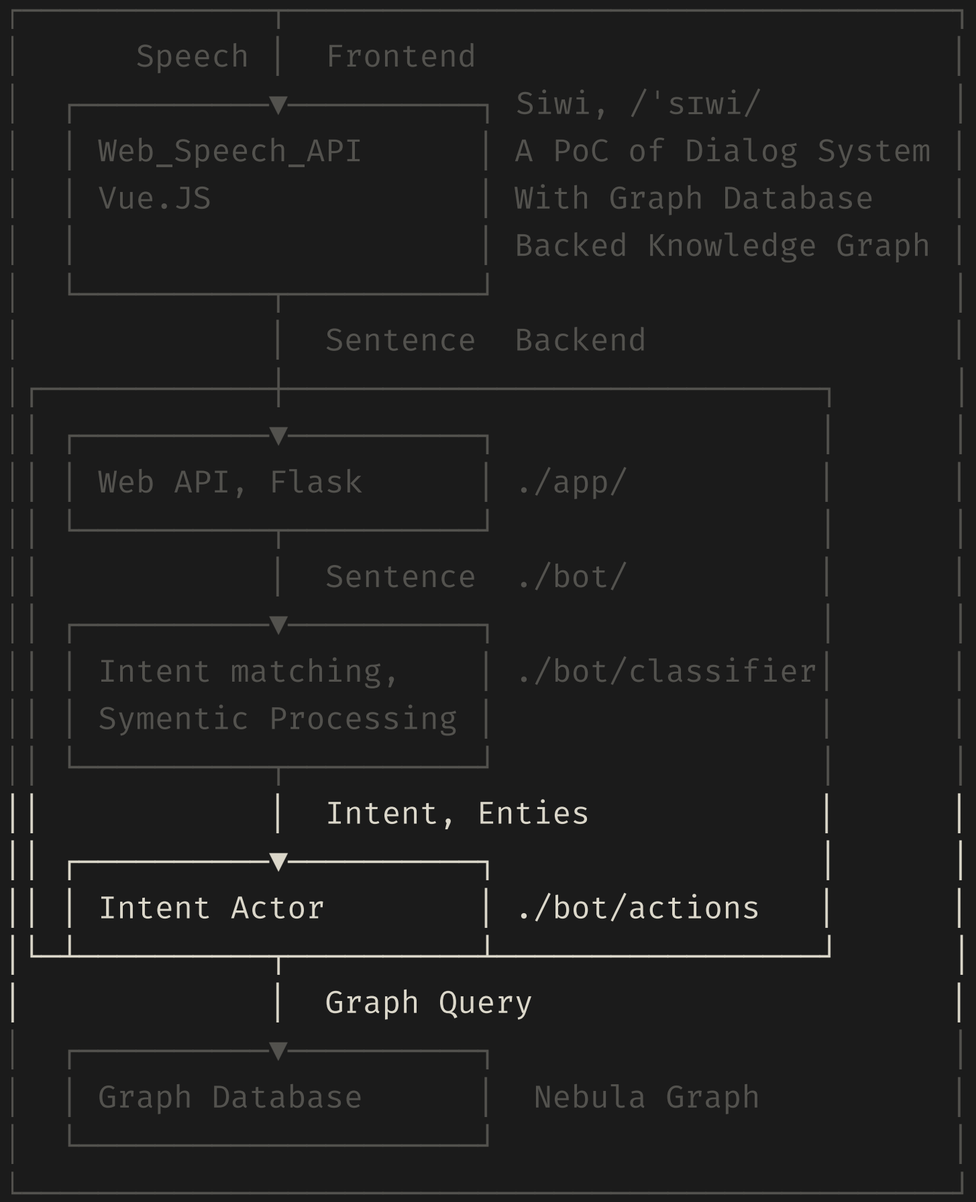


Code

```
.
├── 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 of Siwi

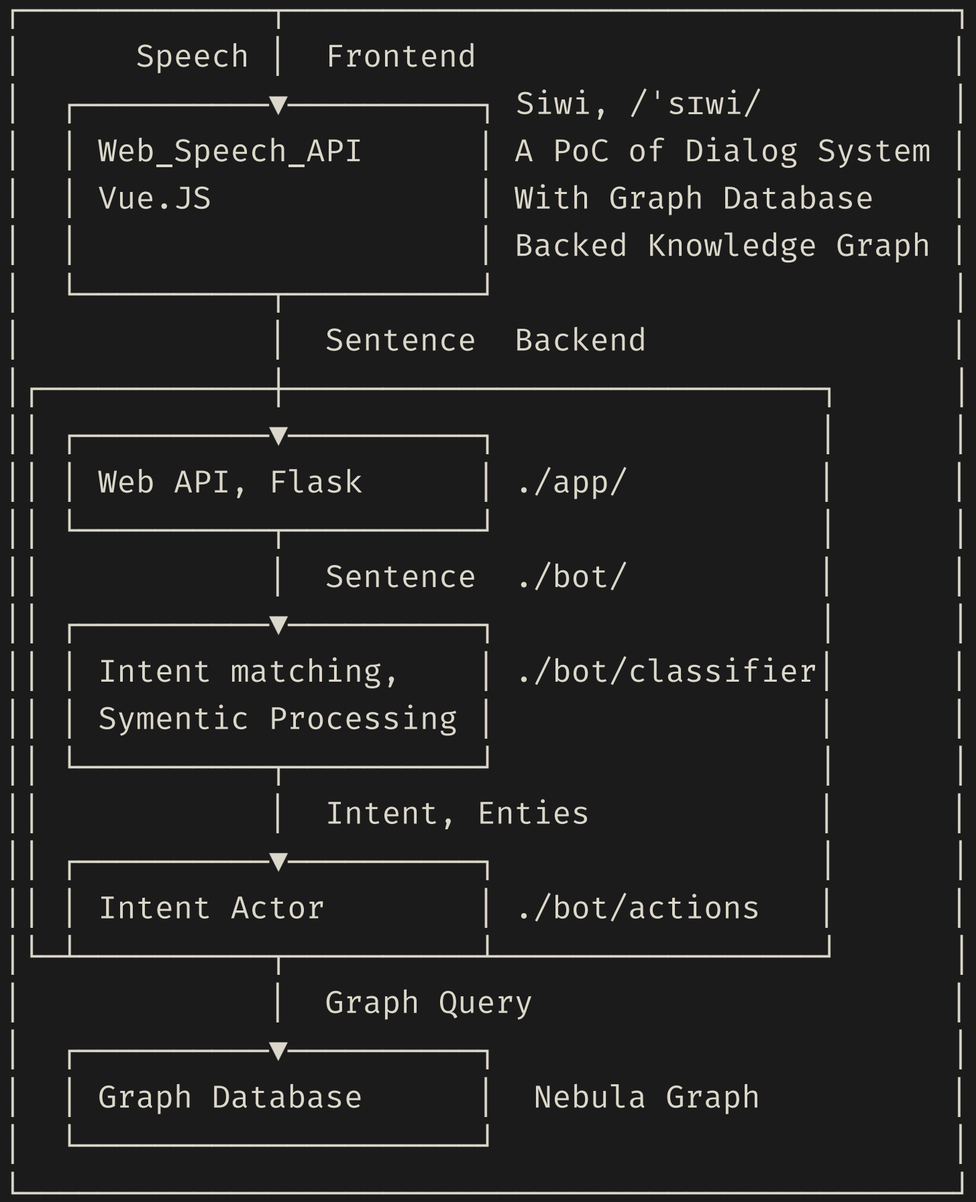


Code

```
.
├── 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 of Siwi



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 0

Py-Siwi with Nebula Graph on K8s

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

GDB: Nebula Graph in KinD(K8s in Docker)

```
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
$ ~/.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 distribution  |
+-----------------------------------------------------------------------------+------+----------+--------------+----------------------+
| "nebula-storaged-0.nebula-storaged-headless.default.svc.cluster.local" | 9779 | "ONLINE" | 0            | "No valid partition  |
+-----------------------------------------------------------------------------+------+----------+--------------+----------------------+
| "nebula-storaged-1.nebula-storaged-headless.default.svc.cluster.local" | 9779 | "ONLINE" | 0            | "No valid partition  |
+-----------------------------------------------------------------------------+------+----------+--------------+----------------------+
| "nebula-storaged-2.nebula-storaged-headless.default.svc.cluster.local" | 9779 | "ONLINE" | 0            | "No valid partition  |
+-----------------------------------------------------------------------------+------+----------+--------------+----------------------+
| "Total"                                                                     |      |          | 0            |                      |
+-----------------------------------------------------------------------------+------+----------+--------------+----------------------+
Got 4 rows (time spent 2510/2955 us)

Wed, 01 Sep 2021 20:38:42 UTC
```

GDB: Nebula Graph in KinD(K8s in Docker)

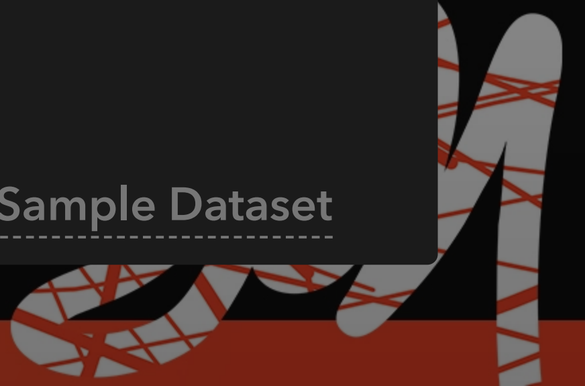
```
curl -sL nebula-kind.siwei.io/install.sh | bash
```

```
$ 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
$ ~/.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 distribution
+-----------------------------------------------------------------------+------+----------+--------------+----------------------
| "nebula-storaged-0.nebula-storaged-headless.default.svc.cluster.local" | 9779 | "ONLINE" | 0            | "No valid partition
+-----------------------------------------------------------------------+------+----------+--------------+----------------------
| "nebula-storaged-1.nebula-storaged-headless.default.svc.cluster.local" | 9779 | "ONLINE" | 0            | "No valid partition
+-----------------------------------------------------------------------+------+----------+--------------+----------------------
| "nebula-storaged-2.nebula-storaged-headless.default.svc.cluster.local" | 9779 | "ONLINE" | 0            | "No valid partition
+-----------------------------------------------------------------------+------+----------+--------------+----------------------
| "Total"                                                               |      |          | 0            |
+-----------------------------------------------------------------------+------+----------+--------------+----------------------
Got 4 rows (time spent 2510/2955 us)

Wed, 01 Sep 2021 20:38:42 UTC
```

Data Import with Console

```
$ 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@nebula]# ~/.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**



Data Import with Console

```
$ 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@nebula]# ~/.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**

Data Import with Console

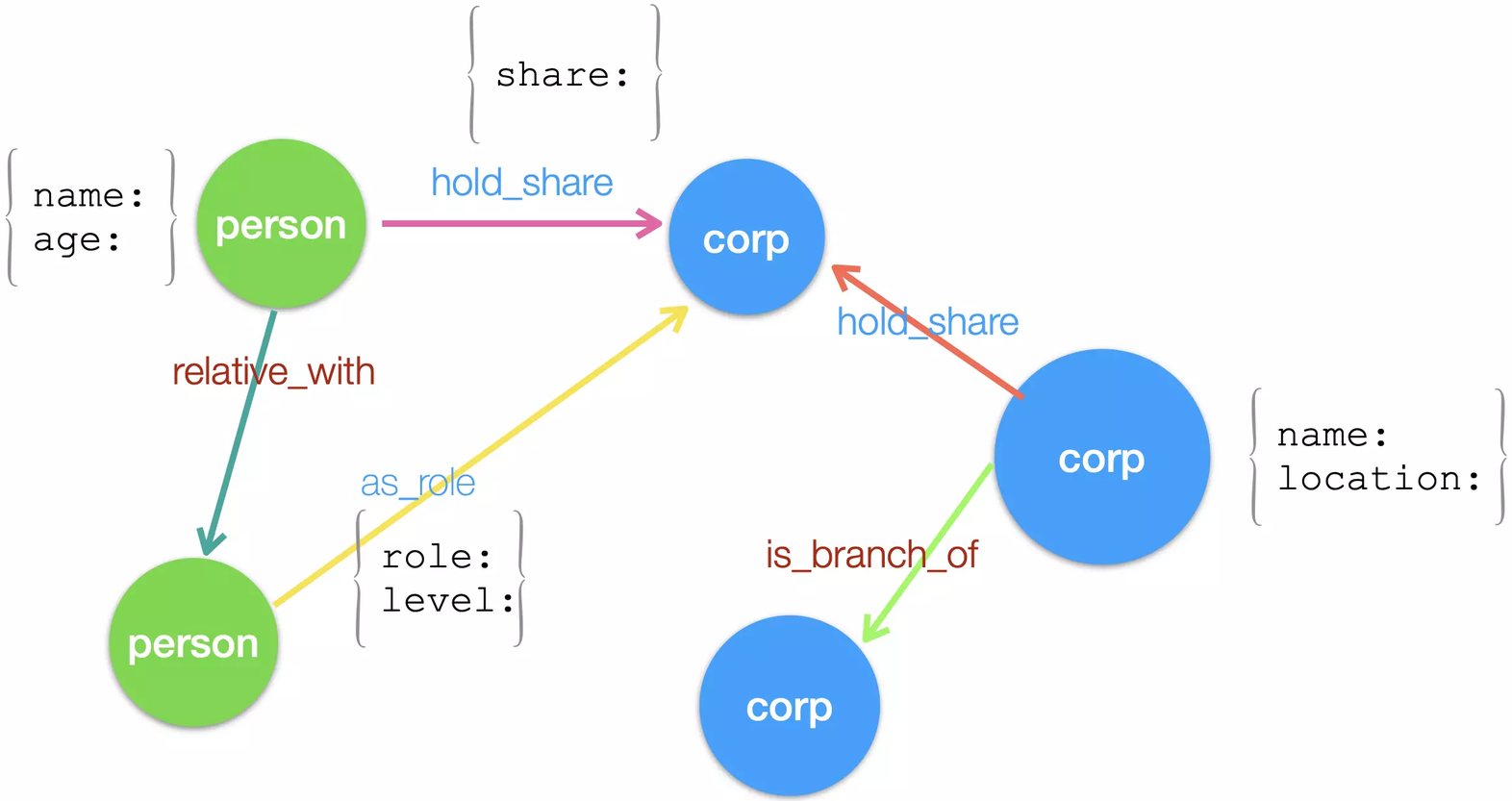
```
$ wget https://docs.nebula-graph.io/2.0/basketballplayer-2.X.ngql
$ ~/.nebula-kind/bin/console -u root -p password --address=192.168.8.137 --port=32669 -f basketballplayer-2.X.ngql
...
(root@nebula) [basketballplayer]> insert edge serve(start_year,end_year) values "player150"->"team213":(2018, 2019);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

[root@nebula]# ~/.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**

Corp-Rel-Search 企业关系图谱、股权穿透系统

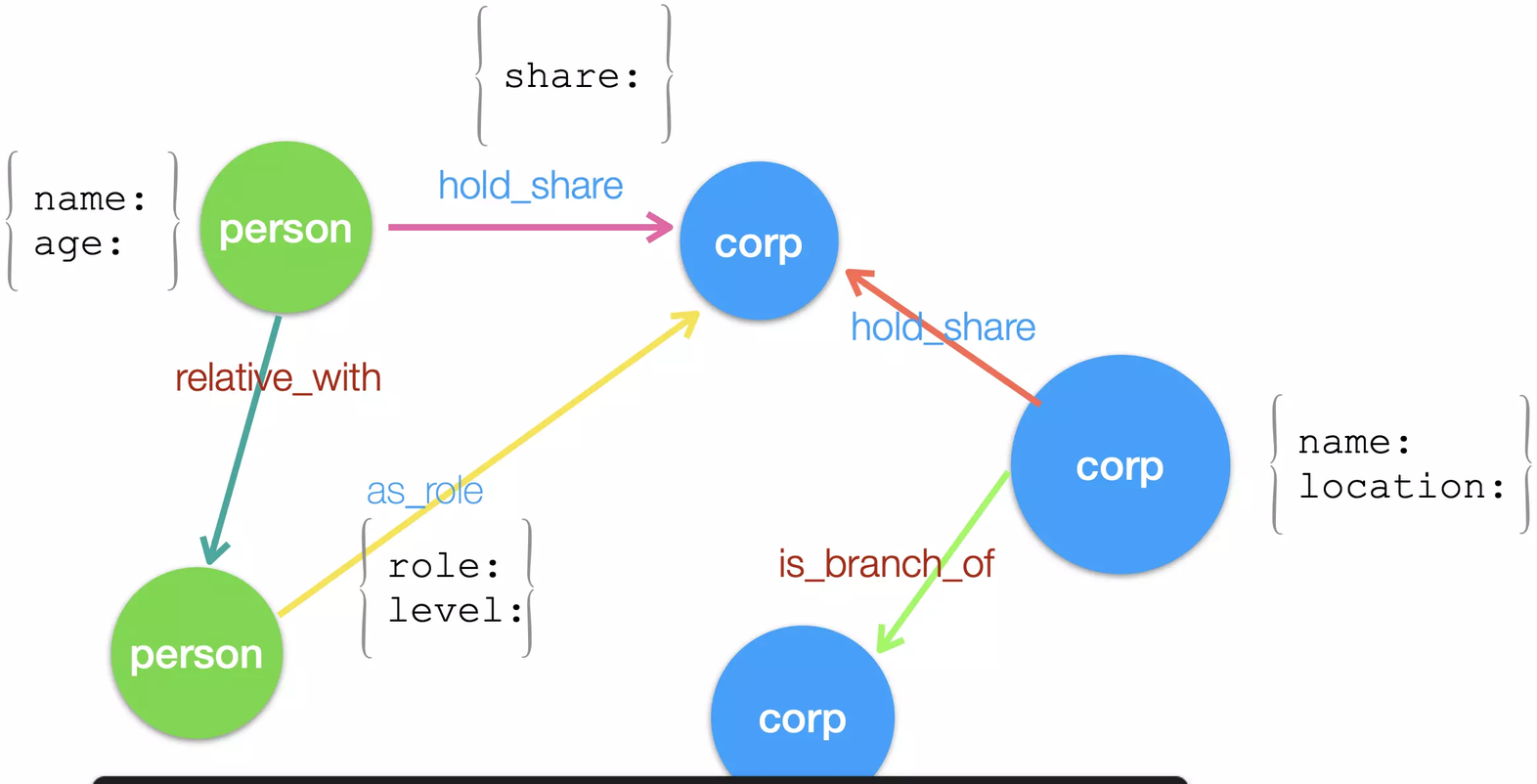
Corp-Rel-Search is a PoC of Corporation Relationship Knowledge Graph System on top of Nebula Graph.

Corp Shareholding Dataset/Graph Model



wey-gu/**nebula-shareholding-example**

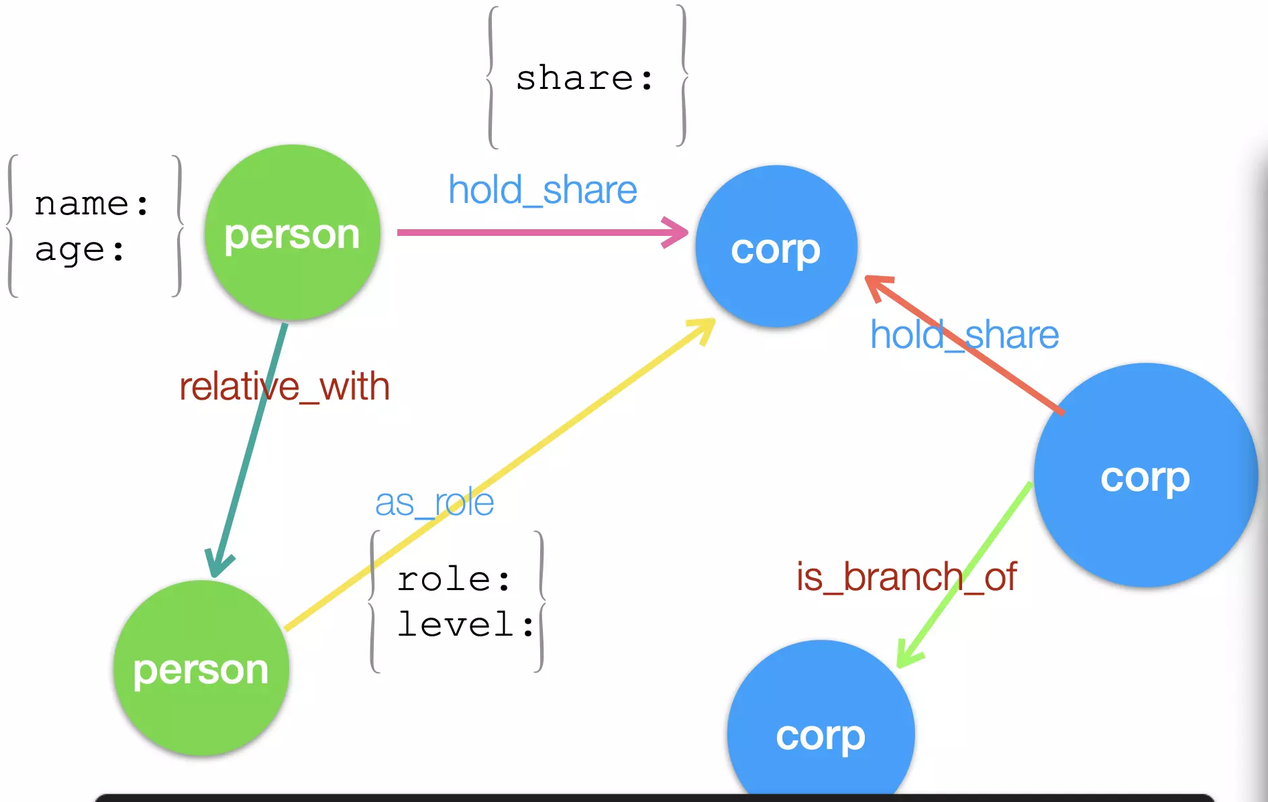
Corp Shareholding Dataset/Graph Model

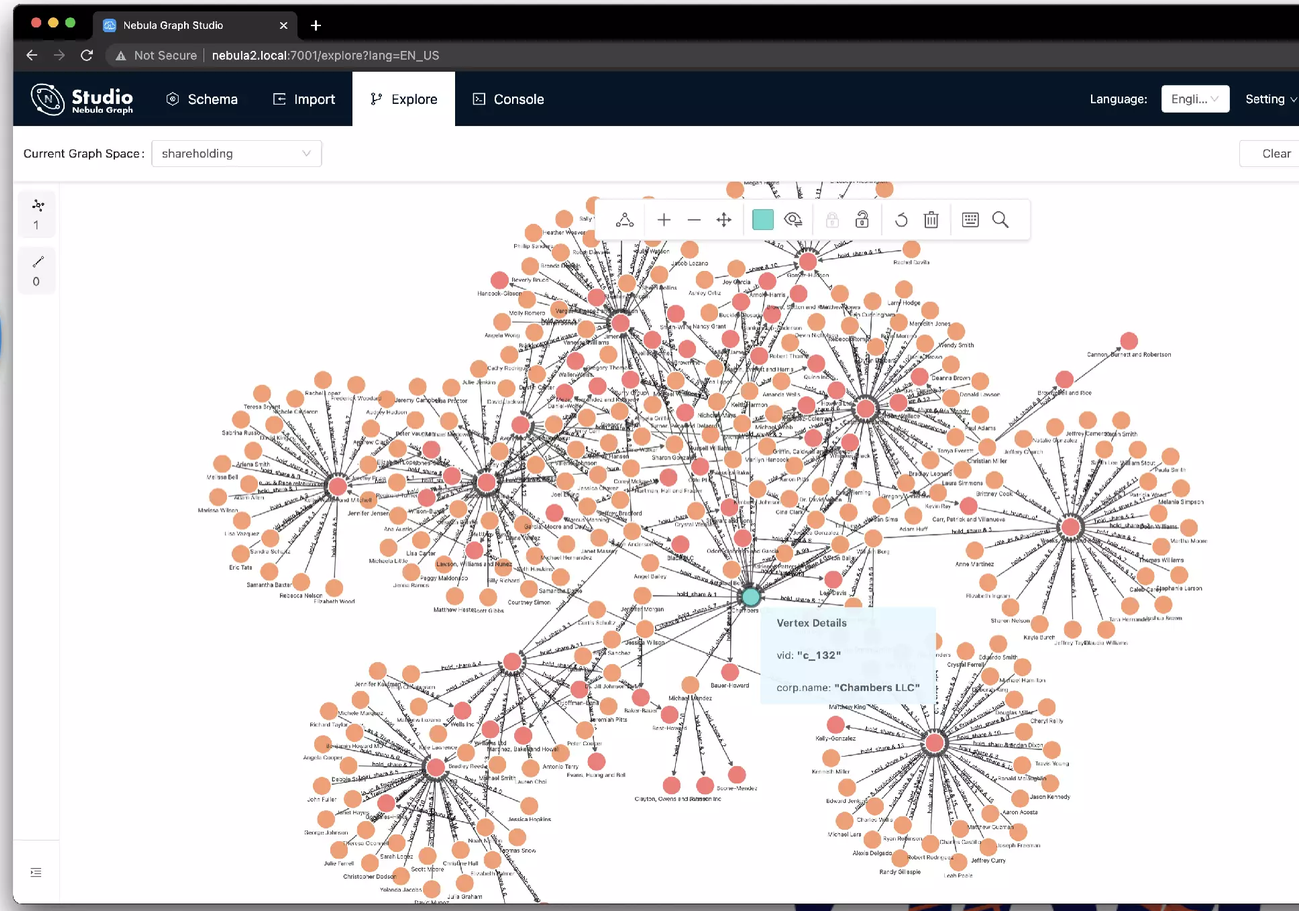



```
GO 1 TO 3 STEPS FROM "c_132" over * BIDIRECT;
```

wey-gu/**nebula-shareholding-example**

Corp Shareholding Dataset/Graph Model

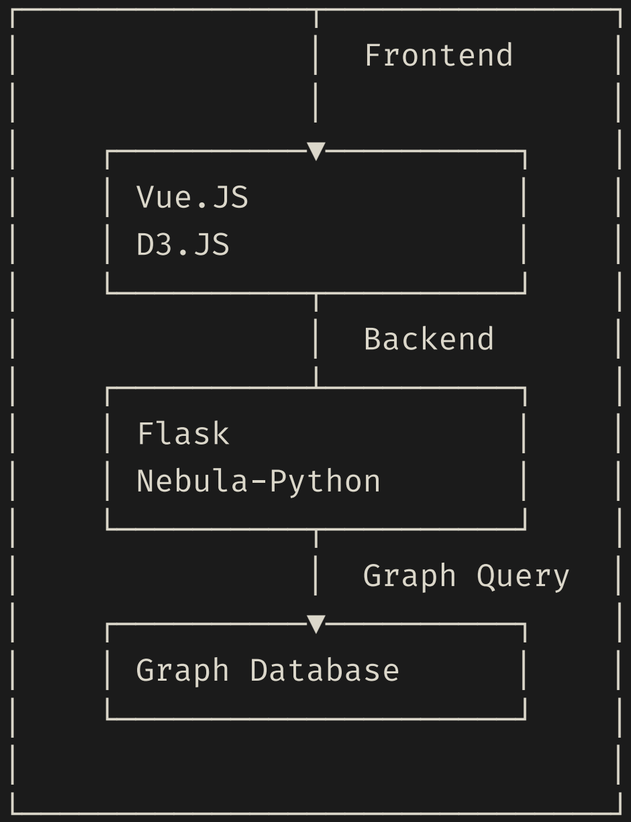





```
GO 1 TO 3 STEPS FROM "c_132" over * BIDIRECT;
```

wey-gu/**nebula-shareholding-example**

Arch of Corp-Rel-Search



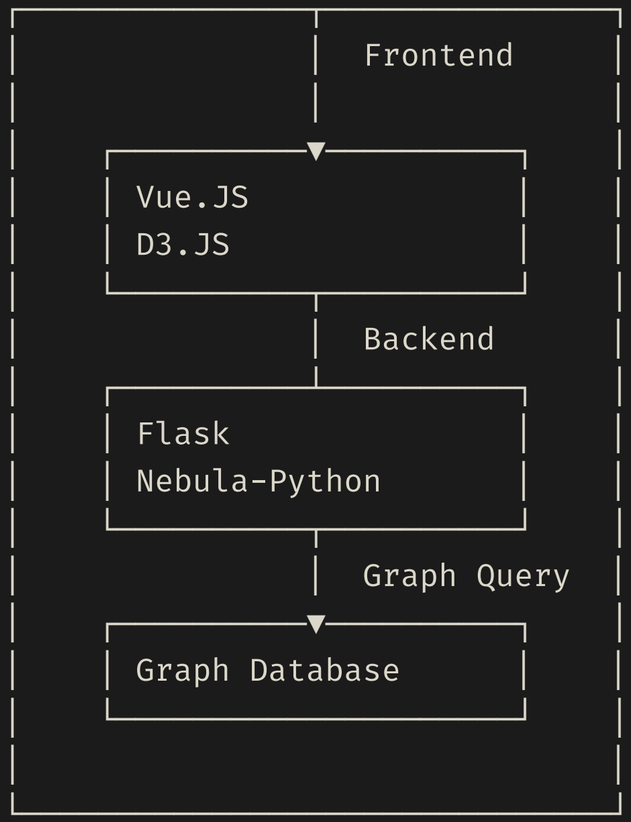
wey-gu/**nebula-corp-rel-search**

The Code

```
├── README.md          # You could find Design Logs here
├── corp-rel-backend
│   └── app.py         # Flask App to handle Requst and call
├── corp-rel-frontend
│   └── src
│       ├── App.vue
│       └── main.js    # Vue App to call Flask App and Rende
└── requirements.txt
```

```python
@app.route("/api", methods=["POST"])
def api():
    entity = request.get_json().get("entity", "")
    if entity:
        resp = query_shareholding(entity)
        data = make_graph_response(resp)
    # ...
    return jsonify(data)

def query_shareholding(entity):
    query_string = (
        f"MATCH p=(v)-[e:hold_share|:is_branch_of|:reletive
        f"WHERE id(v) IN ['{ entity }'] RETURN p LIMIT 100"
    )
    session = connection_pool.get_session('root', 'nebula')
    resp = session.execute(query_string)
    return resp
```

Arch of Corp-Rel-Search



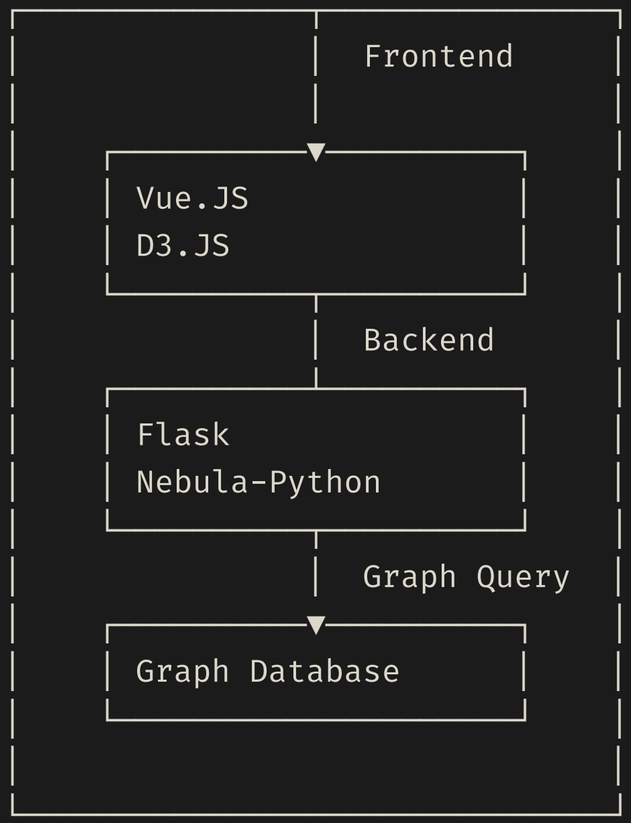
wey-gu/**nebula-corp-rel-search**

The Code

```
├── README.md          # You could find Design Logs here
├── corp-rel-backend
│   └── app.py         # Flask App to handle Requst and call
├── corp-rel-frontend
│   └── src
│       ├── App.vue
│       └── main.js    # Vue App to call Flask App and Rende
└── requirements.txt
```

```python
@app.route("/api", methods=["POST"])
def api():
    entity = request.get_json().get("entity", "")
    if entity:
        resp = query_shareholding(entity)
        data = make_graph_response(resp)
    # ...
    return jsonify(data)

def query_shareholding(entity):
    query_string = (
        f"MATCH p=(v)-[e:hold_share|:is_branch_of|:reletive
        f"WHERE id(v) IN ['{ entity }'] RETURN p LIMIT 100"
    )
    session = connection_pool.get_session('root', 'nebula')
    resp = session.execute(query_string)
    return resp
```

Arch of Corp-Rel-Search



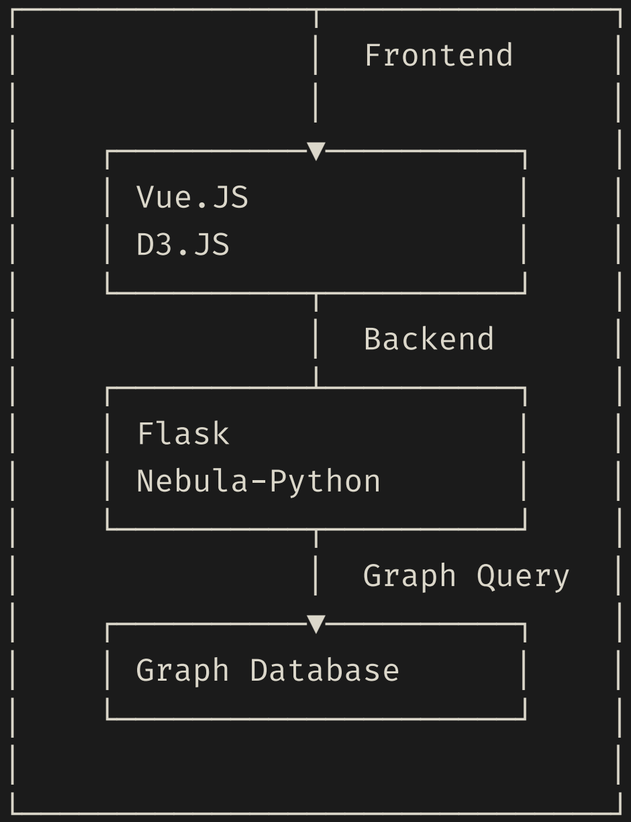
wey-gu/**nebula-corp-rel-search**

The Code

```
├── README.md          # You could find Design Logs here
├── corp-rel-backend
│   └── app.py         # Flask App to handle Requst and call
├── corp-rel-frontend
│   └── src
│       ├── App.vue
│       └── main.js    # Vue App to call Flask App and Rende
└── requirements.txt
```

```python
@app.route("/api", methods=["POST"])
def api():
    entity = request.get_json().get("entity", "")
    if entity:
        resp = query_shareholding(entity)
        data = make_graph_response(resp)
    # ...
    return jsonify(data)

def query_shareholding(entity):
    query_string = (
        f"MATCH p=(v)-[e:hold_share|:is_branch_of|:reletive
        f"WHERE id(v) IN ['{ entity }'] RETURN p LIMIT 100"
    )
    session = connection_pool.get_session('root', 'nebula')
    resp = session.execute(query_string)
    return resp
```

Arch of Corp-Rel-Search



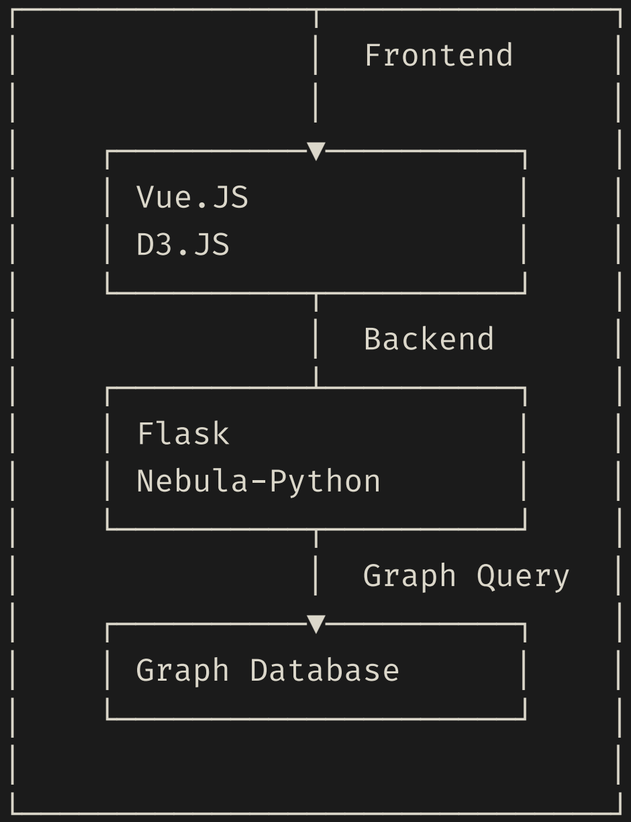
wey-gu/**nebula-corp-rel-search**

The Code

```
├── README.md          # You could find Design Logs here
├── corp-rel-backend
│   └── app.py         # Flask App to handle Requst and call
├── corp-rel-frontend
│   └── src
│       ├── App.vue
│       └── main.js    # Vue App to call Flask App and Rende
└── requirements.txt
```

```python
@app.route("/api", methods=["POST"])
def api():
    entity = request.get_json().get("entity", "")
    if entity:
        resp = query_shareholding(entity)
        data = make_graph_response(resp)
    # ...
    return jsonify(data)

def query_shareholding(entity):
    query_string = (
        f"MATCH p=(v)-[e:hold_share|:is_branch_of|:reletive
        f"WHERE id(v) IN ['{ entity }'] RETURN p LIMIT 100"
    )
    session = connection_pool.get_session('root', 'nebula')
    resp = session.execute(query_string)
    return resp
```

Arch of Corp-Rel-Search



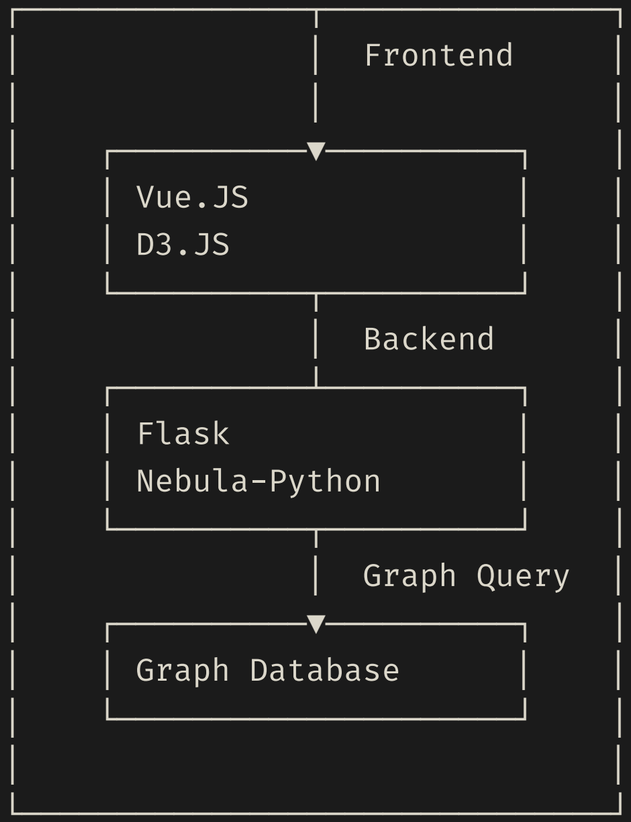
wey-gu/**nebula-corp-rel-search**

The Code

```
├── README.md          # You could find Design Logs here
├── corp-rel-backend
│   └── app.py         # Flask App to handle Requst and call
├── corp-rel-frontend
│   └── src
│       ├── App.vue
│       └── main.js    # Vue App to call Flask App and Rende
└── requirements.txt
```

```python
@app.route("/api", methods=["POST"])
def api():
    entity = request.get_json().get("entity", "")
    if entity:
        resp = query_shareholding(entity)
        data = make_graph_response(resp)
    # ...
    return jsonify(data)

def query_shareholding(entity):
    query_string = (
        f"MATCH p=(v)-[e:hold_share|:is_branch_of|:reletive
        f"WHERE id(v) IN ['{ entity }'] RETURN p LIMIT 100"
    )
    session = connection_pool.get_session('root', 'nebula')
    resp = session.execute(query_string)
    return resp
```

Arch of Corp-Rel-Search



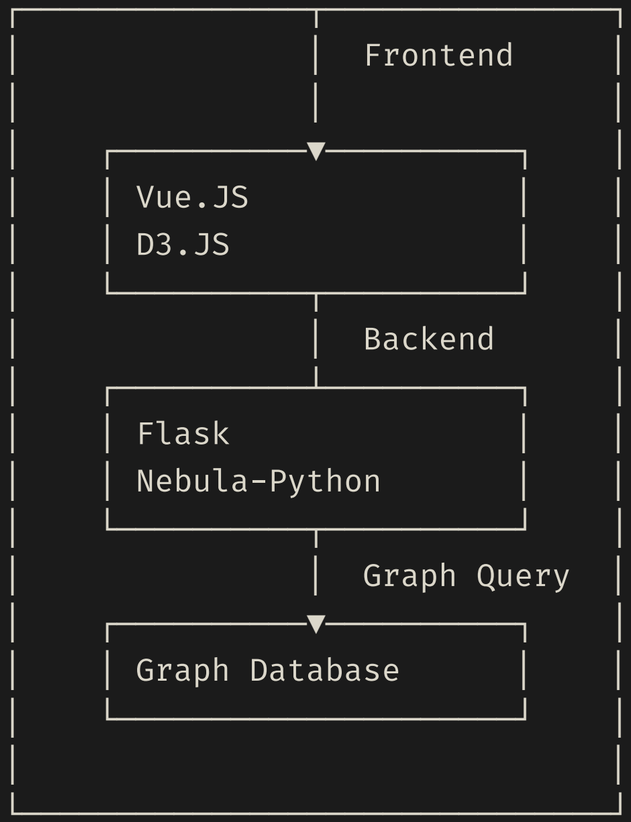
wey-gu/**nebula-corp-rel-search**

The Code

```
├── README.md          # You could find Design Logs here
├── corp-rel-backend
│   └── app.py         # Flask App to handle Requst and call
├── corp-rel-frontend
│   └── src
│       ├── App.vue
│       └── main.js    # Vue App to call Flask App and Rende
└── requirements.txt
```

```python
@app.route("/api", methods=["POST"])
def api():
    entity = request.get_json().get("entity", "")
    if entity:
        resp = query_shareholding(entity)
        data = make_graph_response(resp)
    # ...
    return jsonify(data)

def query_shareholding(entity):
    query_string = (
        f"MATCH p=(v)-[e:hold_share|:is_branch_of|:reletive
        f"WHERE id(v) IN ['{ entity }'] RETURN p LIMIT 100"
    )
    session = connection_pool.get_session('root', 'nebula')
    resp = session.execute(query_string)
    return resp
```

Arch of Corp-Rel-Search



wey-gu/**nebula-corp-rel-search**

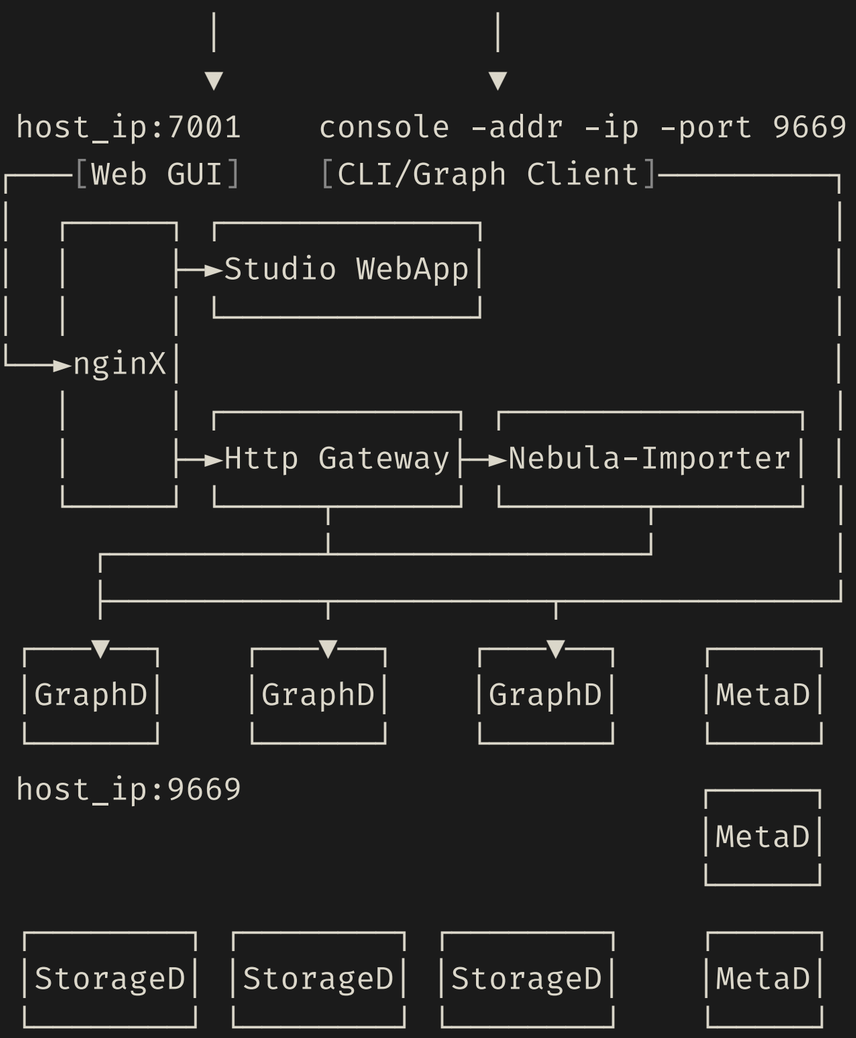
The Code

```
├── README.md          # You could find Design Logs here
├── corp-rel-backend
│   └── app.py         # Flask App to handle Requst and call
├── corp-rel-frontend
│   └── src
│       ├── App.vue
│       └── main.js    # Vue App to call Flask App and Rende
└── requirements.txt
```

```python
@app.route("/api", methods=["POST"])
def api():
    entity = request.get_json().get("entity", "")
    if entity:
        resp = query_shareholding(entity)
        data = make_graph_response(resp)
    # ...
    return jsonify(data)

def query_shareholding(entity):
    query_string = (
        f"MATCH p=(v)-[e:hold_share|:is_branch_of|:reletive
        f"WHERE id(v) IN ['{ entity }'] RETURN p LIMIT 100"
    )
    session = connection_pool.get_session('root', 'nebula')
    resp = session.execute(query_string)
    return resp
```

Nebula Graph in Docker and Dataset Import

```
curl -fsSL nebula-up.siwei.io/install.sh | bash
```



Live Demo 1

Py-Corp-Rel-Search with Nebula Graph on Docker

github.com/wey-gu/**nebula-corp-rel-search**

Nebula Graph!

Features? Why Yet another Graph DB?

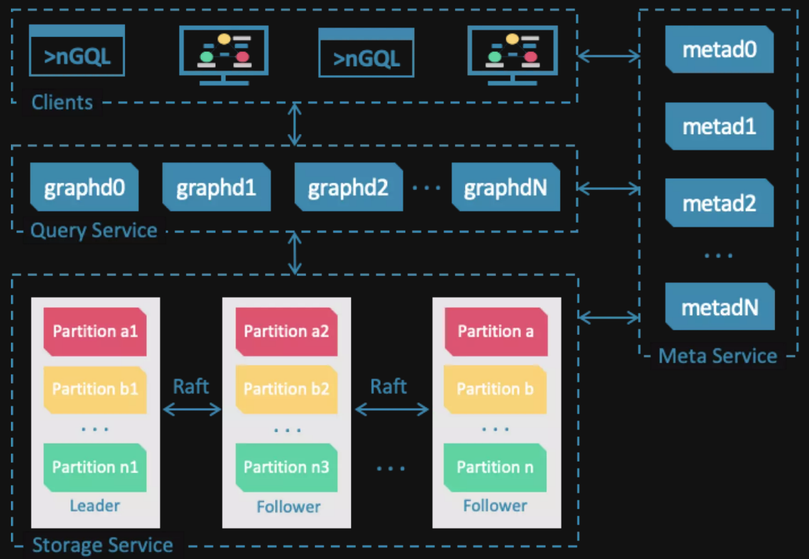
nebula-graph.io/about/**#why-nebula-graph**

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.

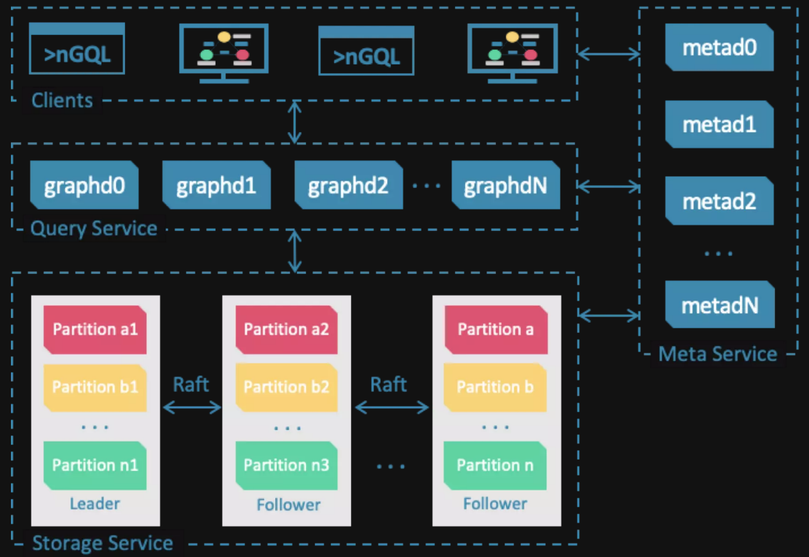
Nebula Graph Introduced

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 Graph Introduced

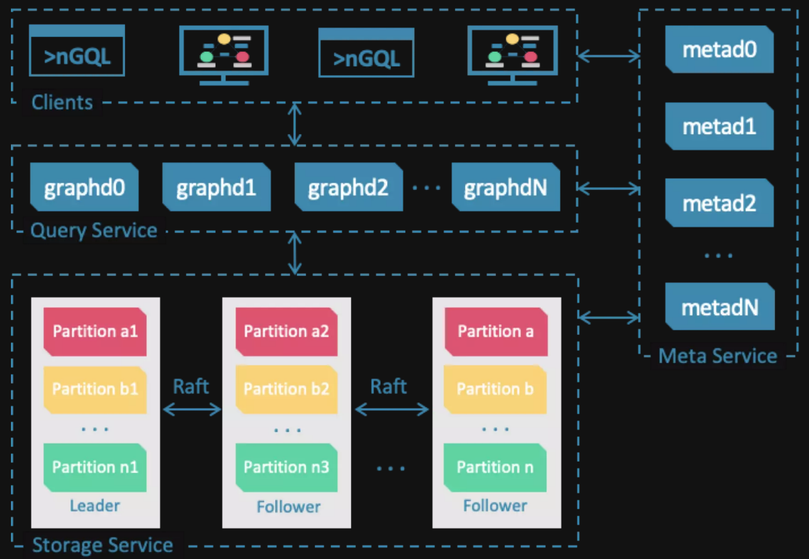
A highly performant linearly scalable graph DB designed in a shared-nothing distributed model. It has been proven to beat the performance of competing graph databases by multiple times over in production.



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

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.

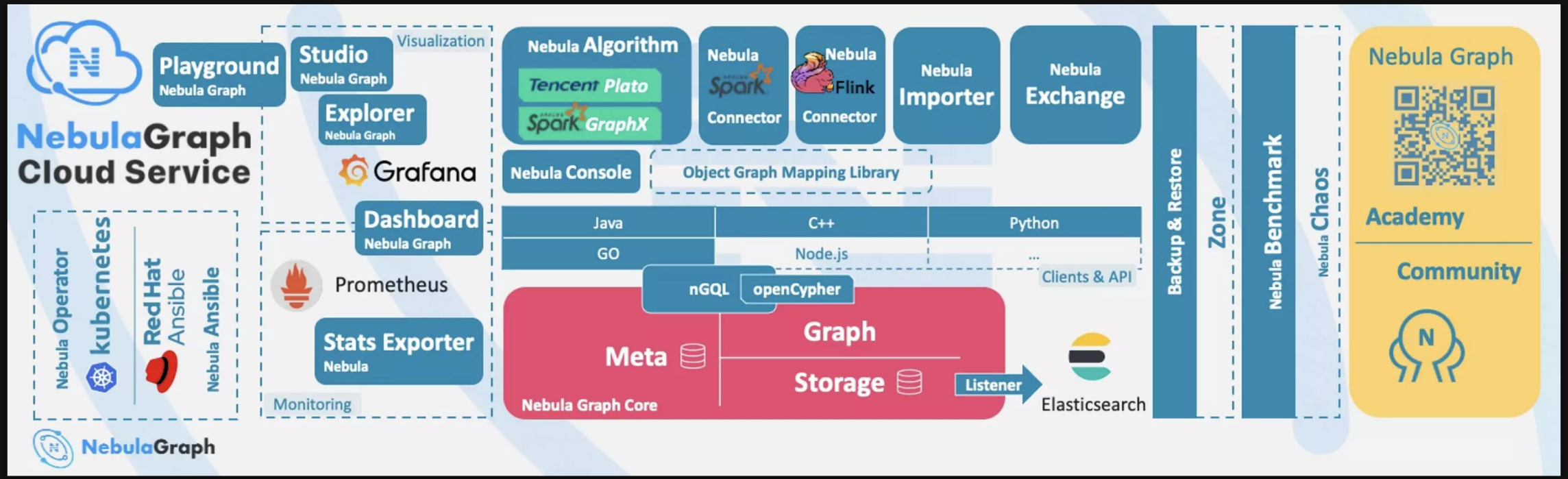
-> Use Cases

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

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**

Recap

- ❓ Why Graph Database? **the Relationship Matters**
- 💡 How Graph Database + Python work together? https://github.com/vesoft-inc/nebula-python
- 🤖 Siwi, a KGDS in Python
- 🕵️‍♂️ Corp-Rel Search in Python
- 🏂 Join the Nebula Community! 🤾‍♀️ https://github.com/vesoft-inc/nebula

siwei.io/talks/2021-PyCon

PyCon China
Oct. 16-17, 2021

PyCon China 2021

2021 中国 PYTHON 开发者大会

2021.10.16-17