

Nebula Graph: Open Source Distributed Graph Database

Nebula Graph Demystified, Graph on K8s & Know-How of Graph Database

WEY GU

DEVELOPER ADVOCATE @  vesoft



DoK
Community

Live Streaming

Feb. 11, 2022

Wey Gu (Siwei)

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



😺 [wey-gu](#)

🐦 [wey_gu](#)

👤 [siwei.io/about](#)

PPT [siwei.io/talks/2022-DoKC/](#)

Agenda

- Graph, WHY Graph DB? 🎅
- Nebula Graph Design/Arch and Why Nebula Graph? 🚢 ⚓
- (Nebula Graph) Know How
 - 💰: Corp-Rel Search
 - 🤖: Siwi the Basketball Robot
 - 🔍: LF Amundsen, the Metadata Mgmt. Engine
- Nebula Operator ⚙



DoK
Community

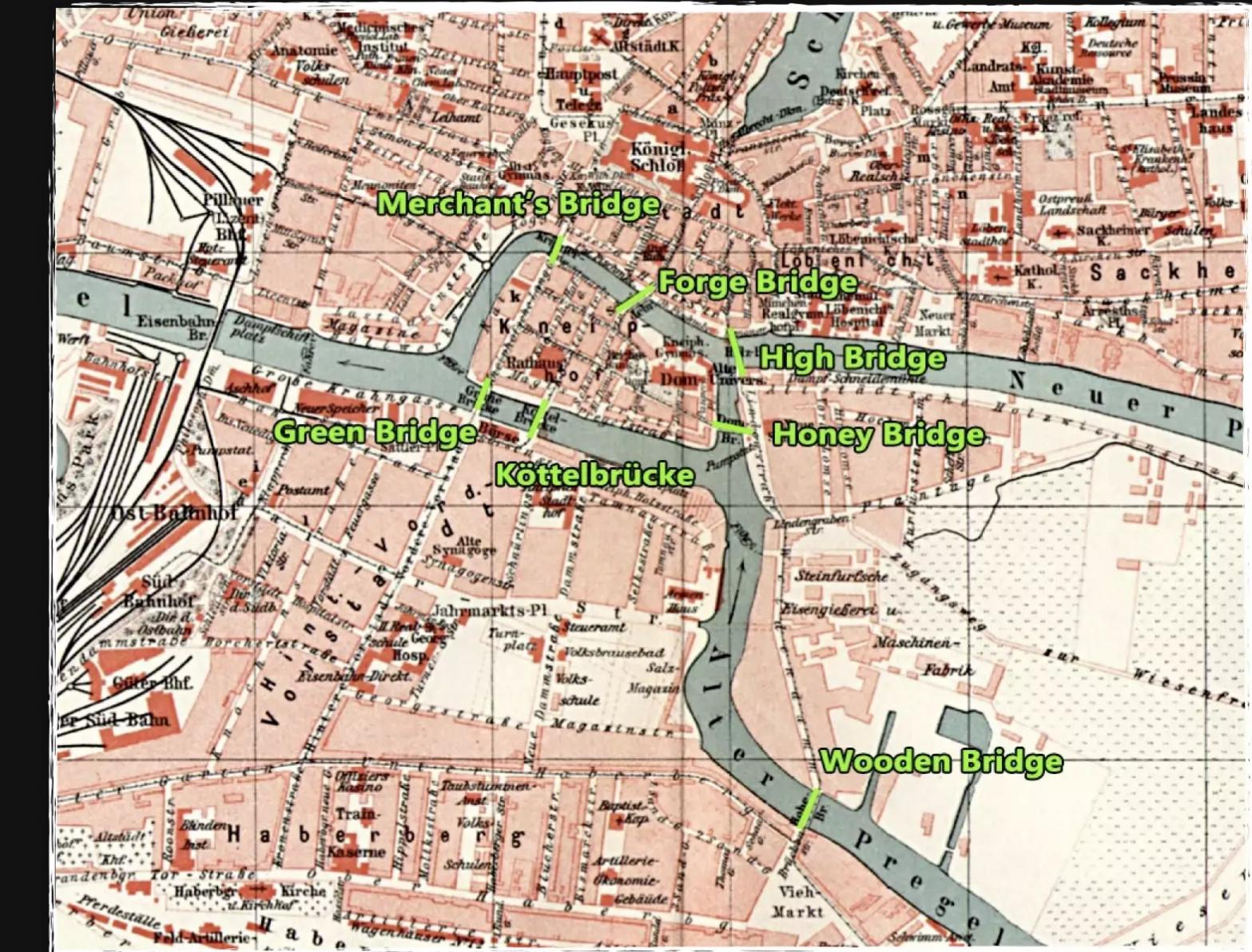
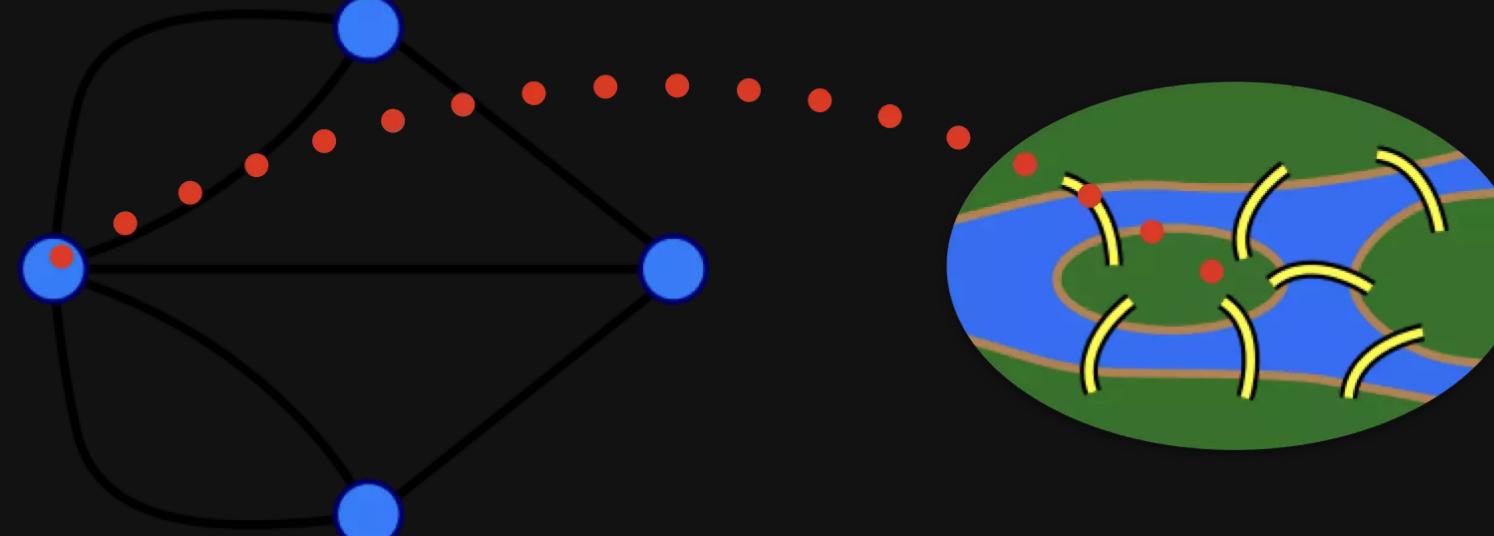
siwei.io/talks/2022-DoKC/

Feb. 11 2022

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

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

[wikipedia.org/wiki/graph_database](https://en.wikipedia.org/wiki/Graph_database)

More on what a GDB is

Why Yet Another DB?

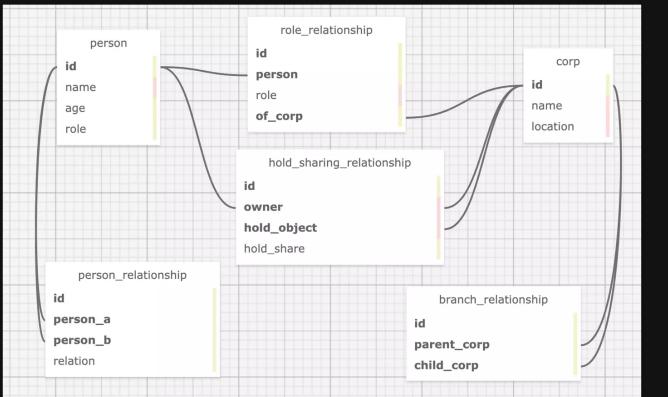
RELATIONAL DB

GRAPH DB

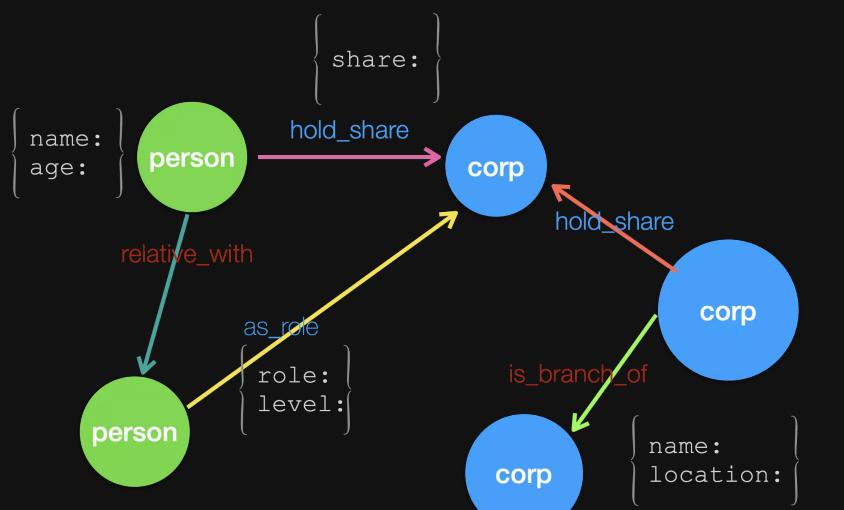
Why Yet Another DB?

Graph Schema

RELATIONAL DB



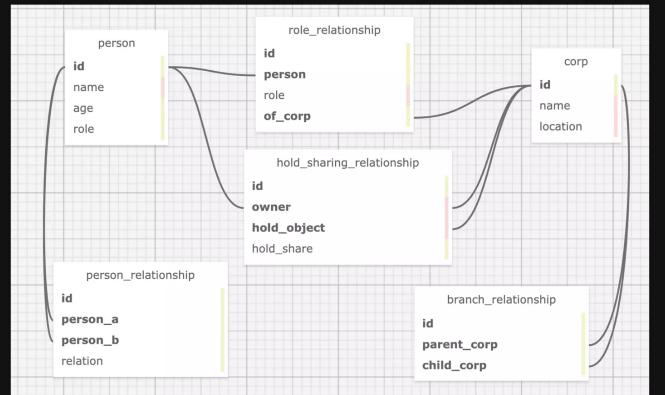
GRAPH DB



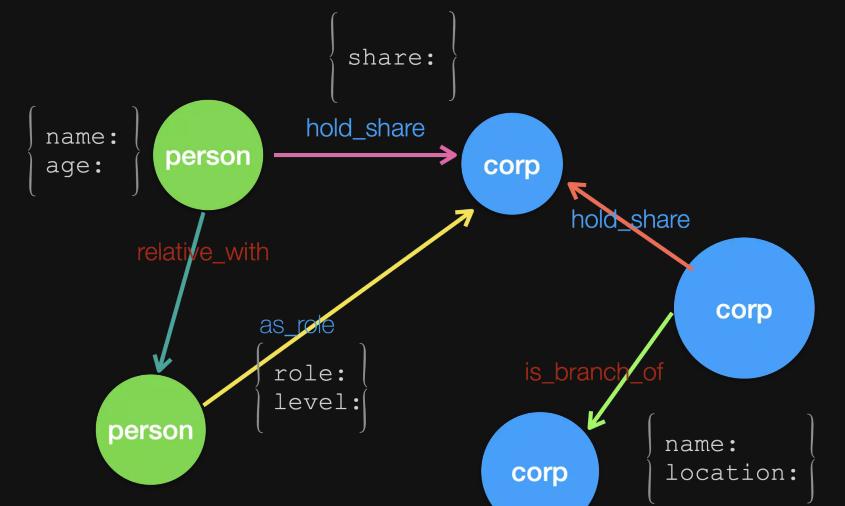
Why Yet Another DB?

Graph Schema

RELATIONAL DB



GRAPH DB



Graph Semantic Queries

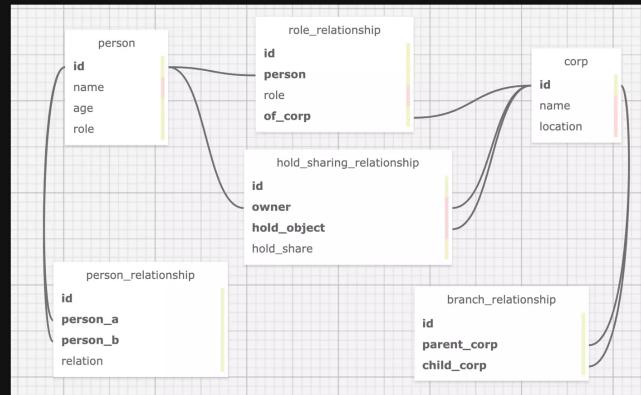
```
SELECT a.id, a.name, c.name
FROM player a
JOIN serve b ON a.id=b.player_id
JOIN team c ON c.id=b.team_id
WHERE c.name IN (SELECT c.name
FROM player a
JOIN serve b ON a.id=b.player_id
JOIN team c ON c.id=b.team_id
WHERE a.name = 'Tim Duncan')
```

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

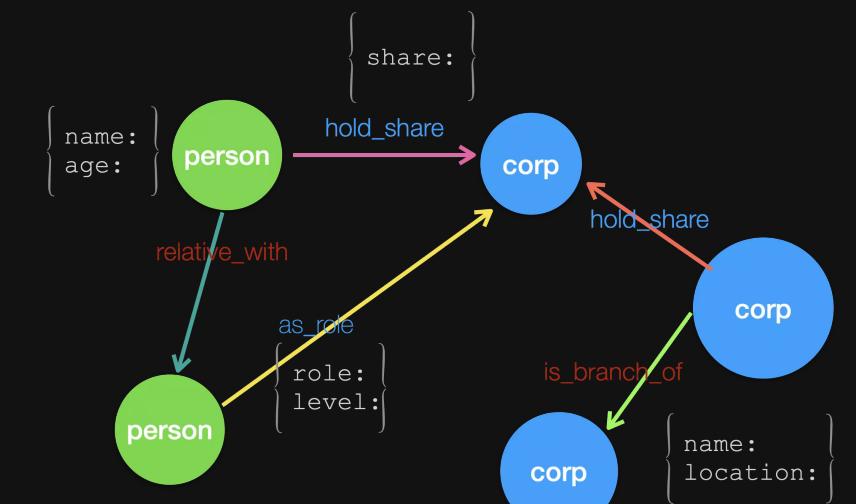
Why Yet Another DB?

Graph Schema

RELATIONAL DB



GRAPH DB



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



Tabular vs. Graph

Tabular vs. Graph

```
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')
```

SQL



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

Graph DB nGQL



Tabular vs. Graph



Nebula Graph!

Why Yet another Graph DB?

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



One of the reasons: Scale

Nebula is a highly performant linearly scalable graph DB designed in a shared-



Wey Gu 古思为
@wey_gu



Scale makes differences
@NebulaGraph



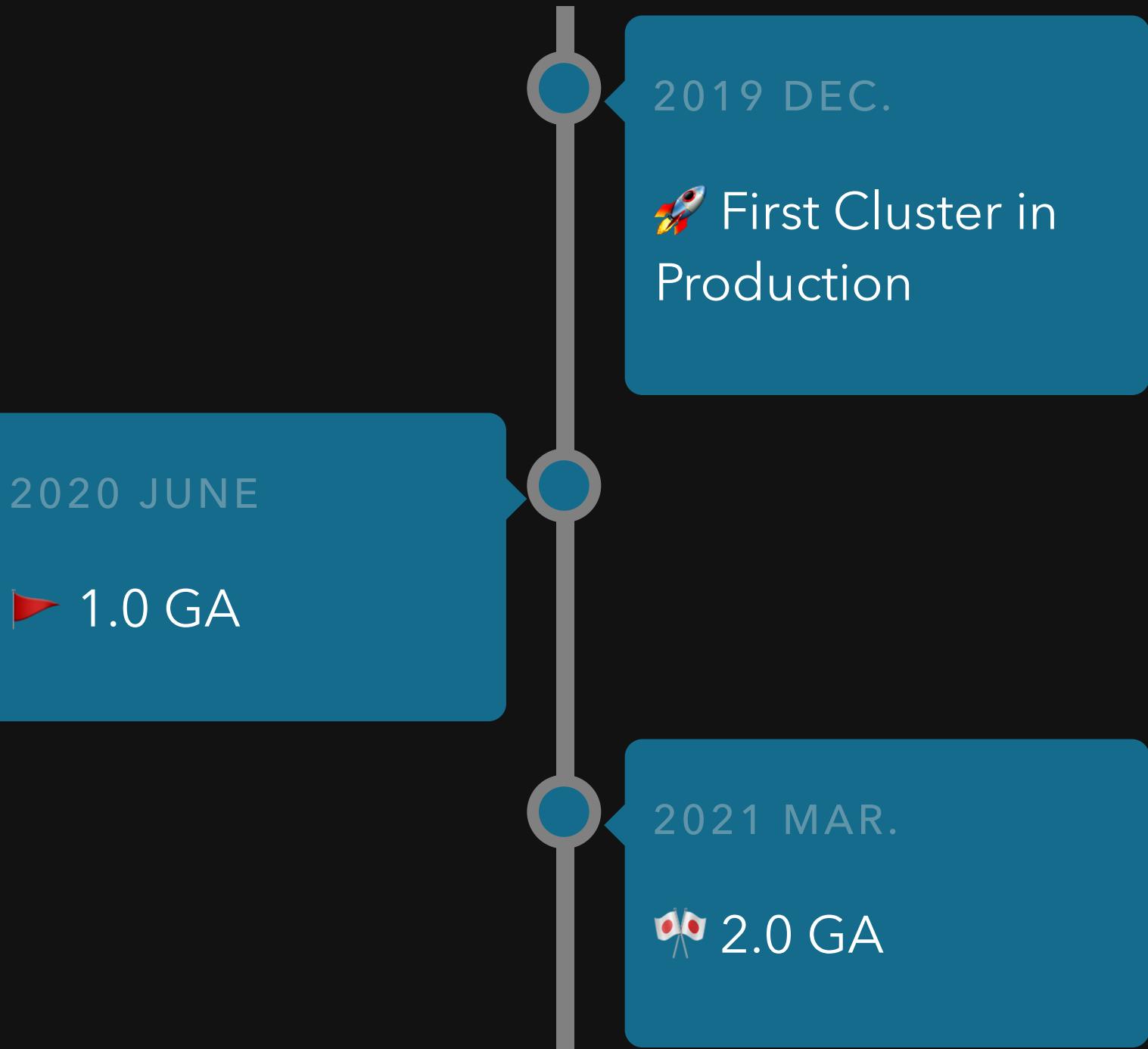
Martin Beeby @thebeebs

Small and medium businesses may have similar problems to big tech companies. But should never assume they need to use the same solution. The answer is not always a shiny new library, a complicated frontend build process, or the latest orchestrator. It is all a matter of scale.



reason: Community

Proven to beat the perf. of competing graph DB by multiple times over in prod. Power users and contributers kept Nebula evolved in next level

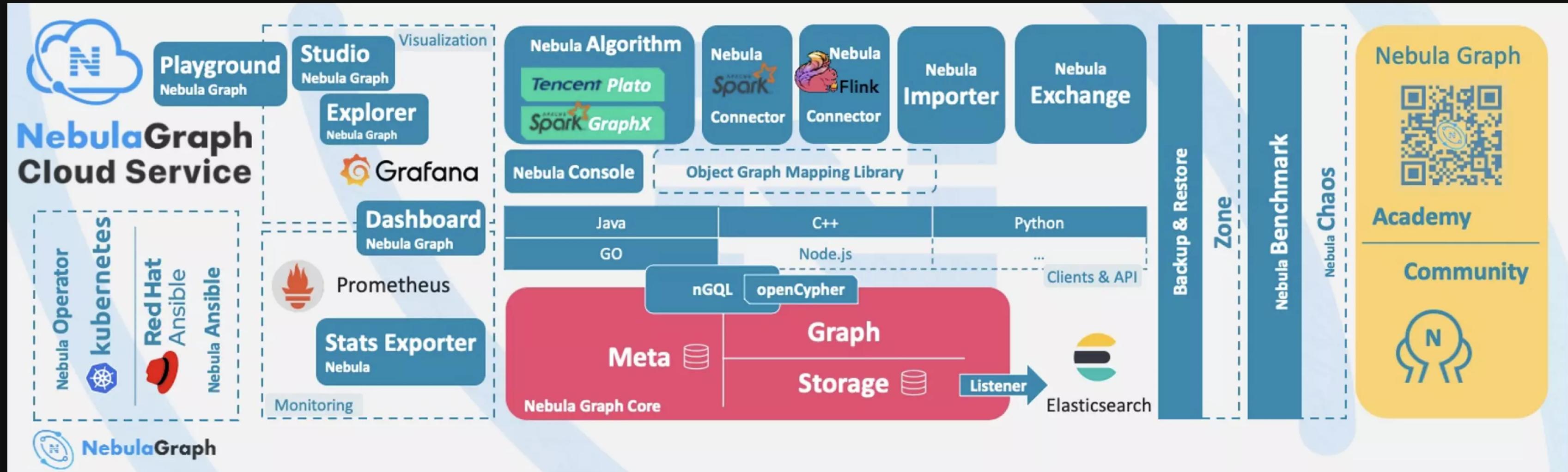


- ① nebula-graph.io/cases/
- ② db-engines.com/en/ranking/graph+dbms

Nebula Landscape

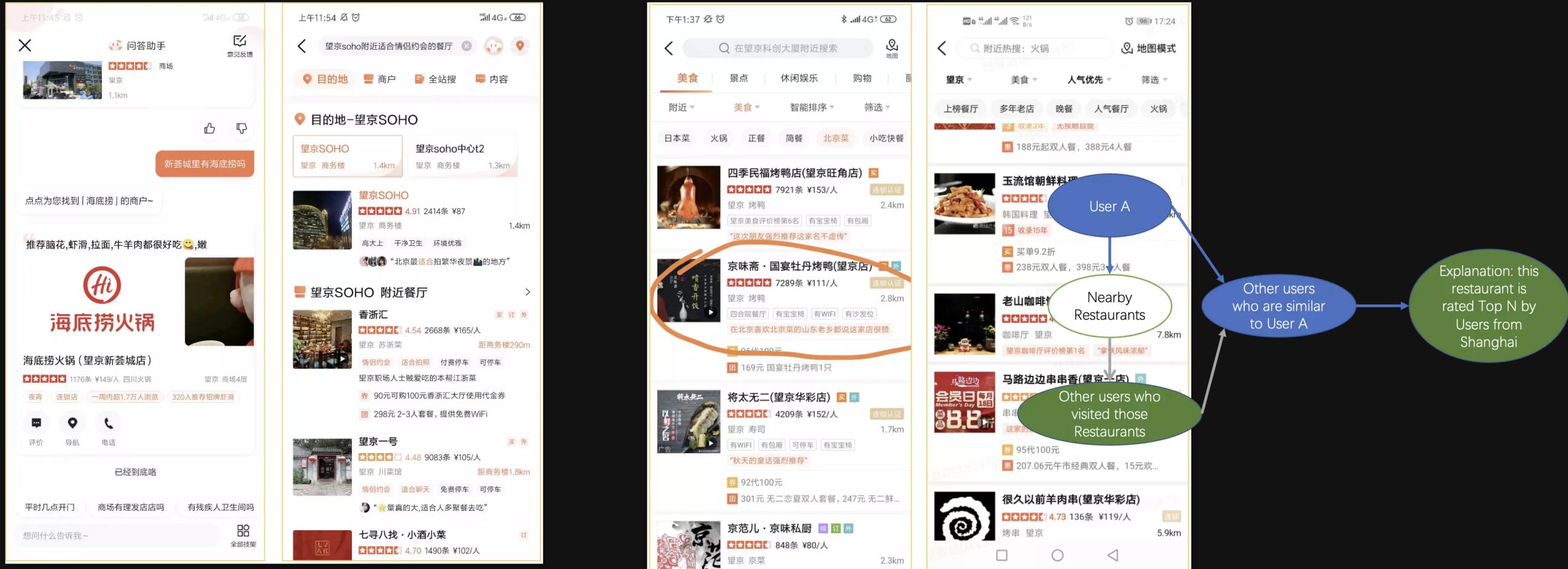
Nebula Community is rich in ecology and still expanding and exploring, welcome to join and contribute!

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



Graph Use Cases

- " Is there any McDonald's nearby street Foofar?
- " Those who from Beijing enjoying the Korean cuisine said this restaurant is great.



SNS

Risk Control

Public Security

Knowledge Graph

ML

Biopharmacy

IoT

Blockchain

Data Lineage

AI Ops



DoK
Community



Holdshare-Rel Analysis

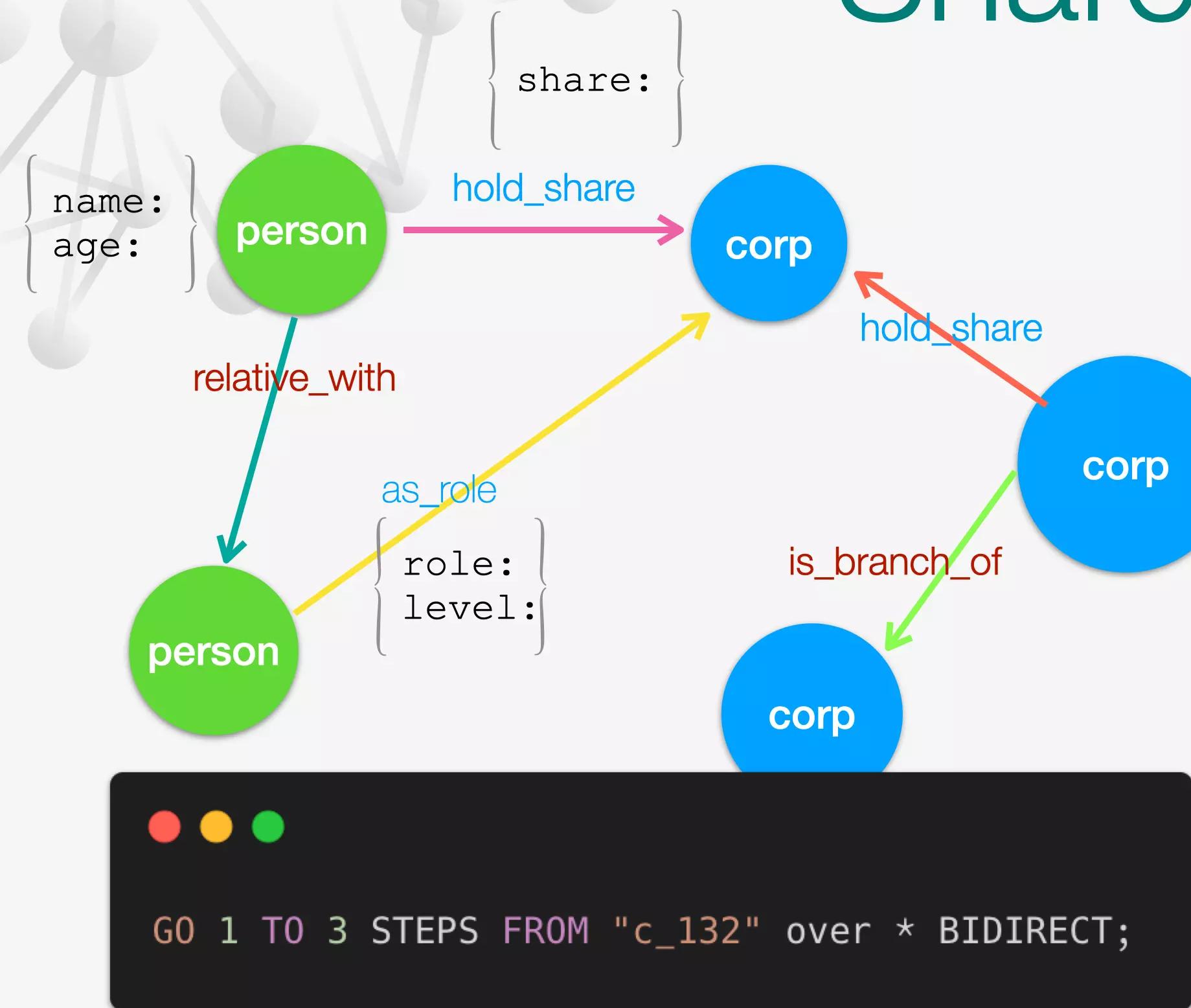
Mini Project



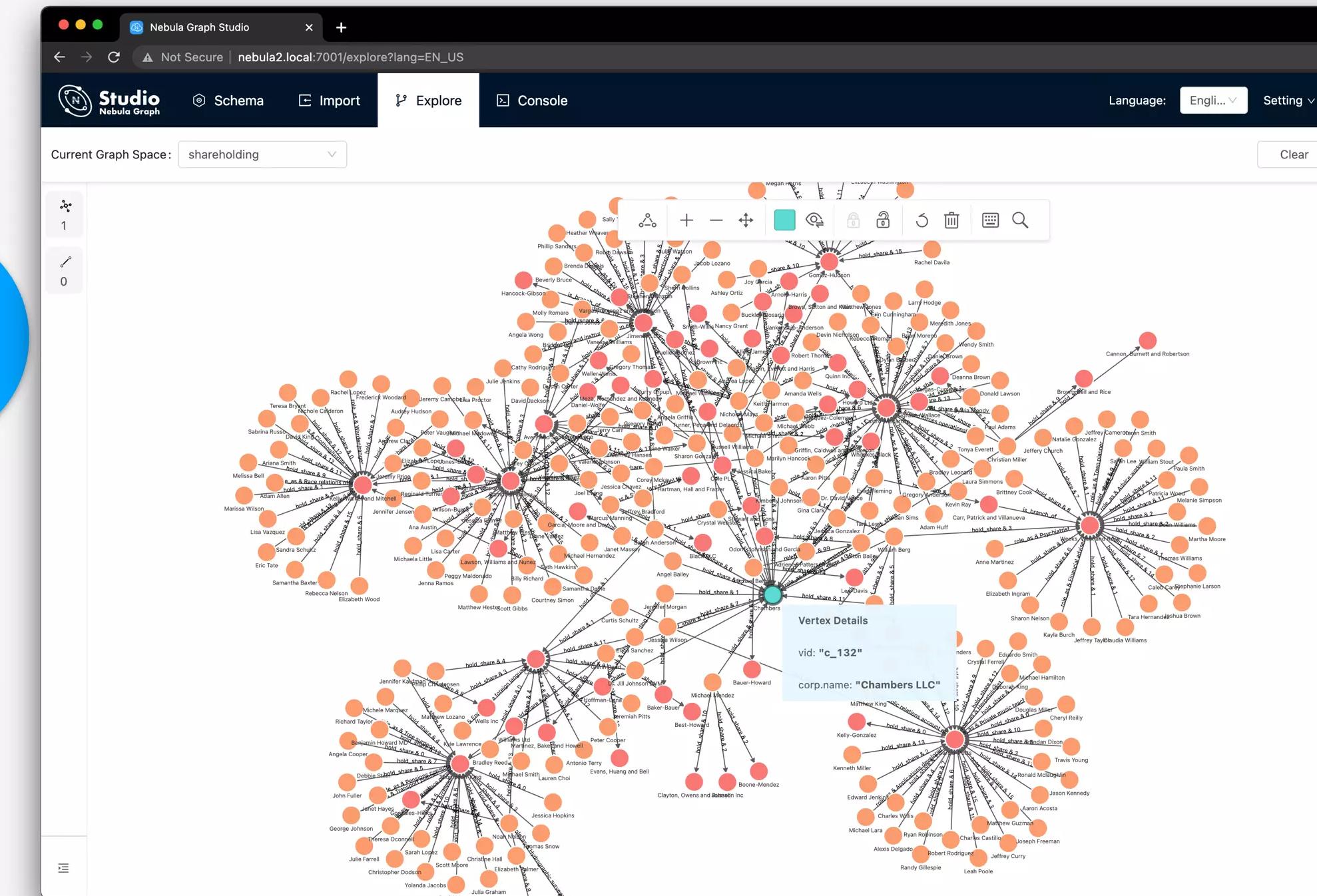
github.com/wey-gu/nebula-corp-rel-search

First Thing First: Property Graph Modeling

<https://nebula-graph.io>



Shareholding

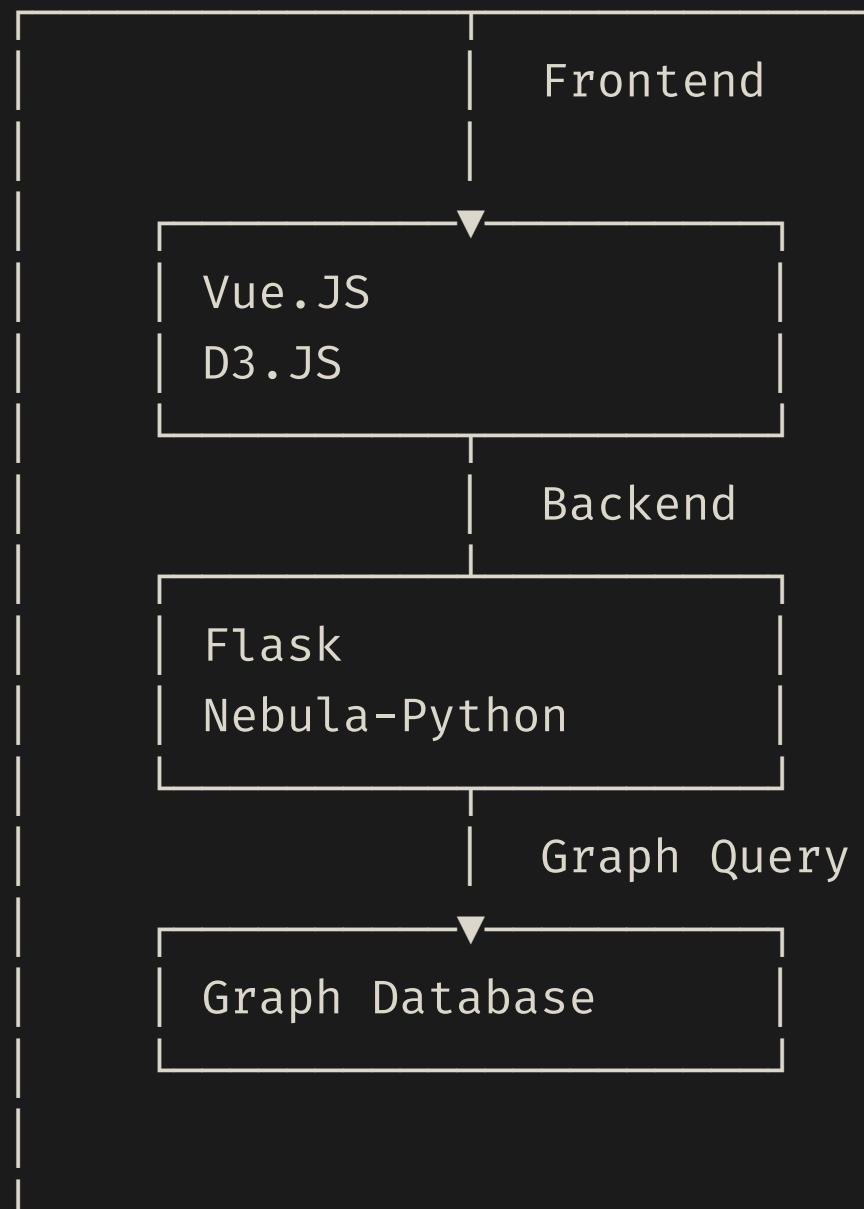


wey-gu/nebula-shareholding-example

Toy project: Corp-Rel Search

WHAT IT'S LIKE FOR A SYSTEM WITH NEBULA GRAPH?

ARCH OF CORP-REL-SEARCH



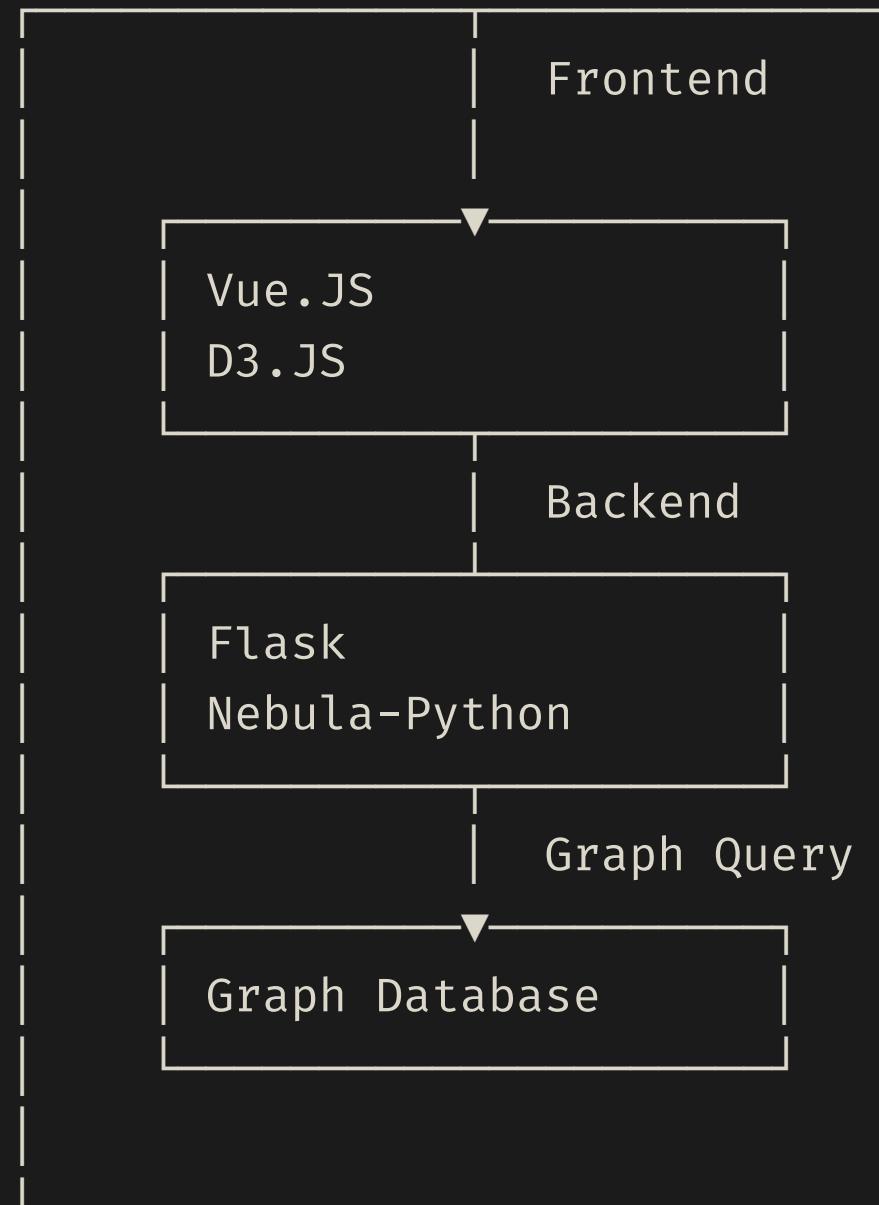
THE CODE

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

```
@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|:releative]-"
        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



THE CODE

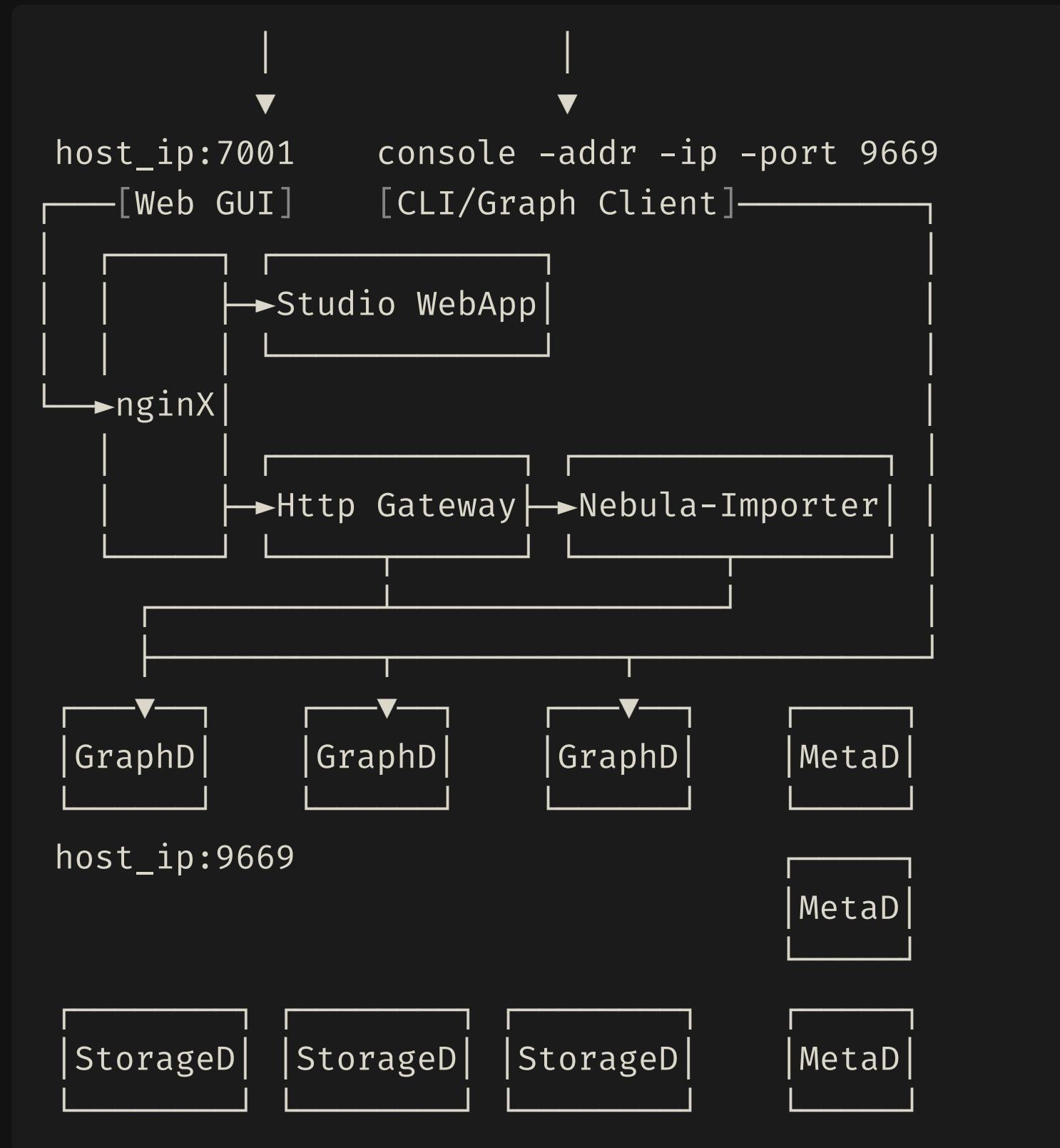
```
├── README.md      # You could find Design Logs here
├── corp-rel-backend
│   └── app.py      # Flask App to handle Request and call
├── corp-rel-frontend
│   └── src
│       ├── App.vue
│       └── main.js    # Vue App to call Flask App and Render
└── requirements.txt
```

```
@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|:releative]-"
        f"WHERE id(v) IN [{entity}] RETURN p LIMIT 100"
    )
    session = connection_pool.get_session('root', 'nebula')
    resp = session.execute(query_string)
    return resp
```

Env Deployment; Fake Data Generation& Import

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

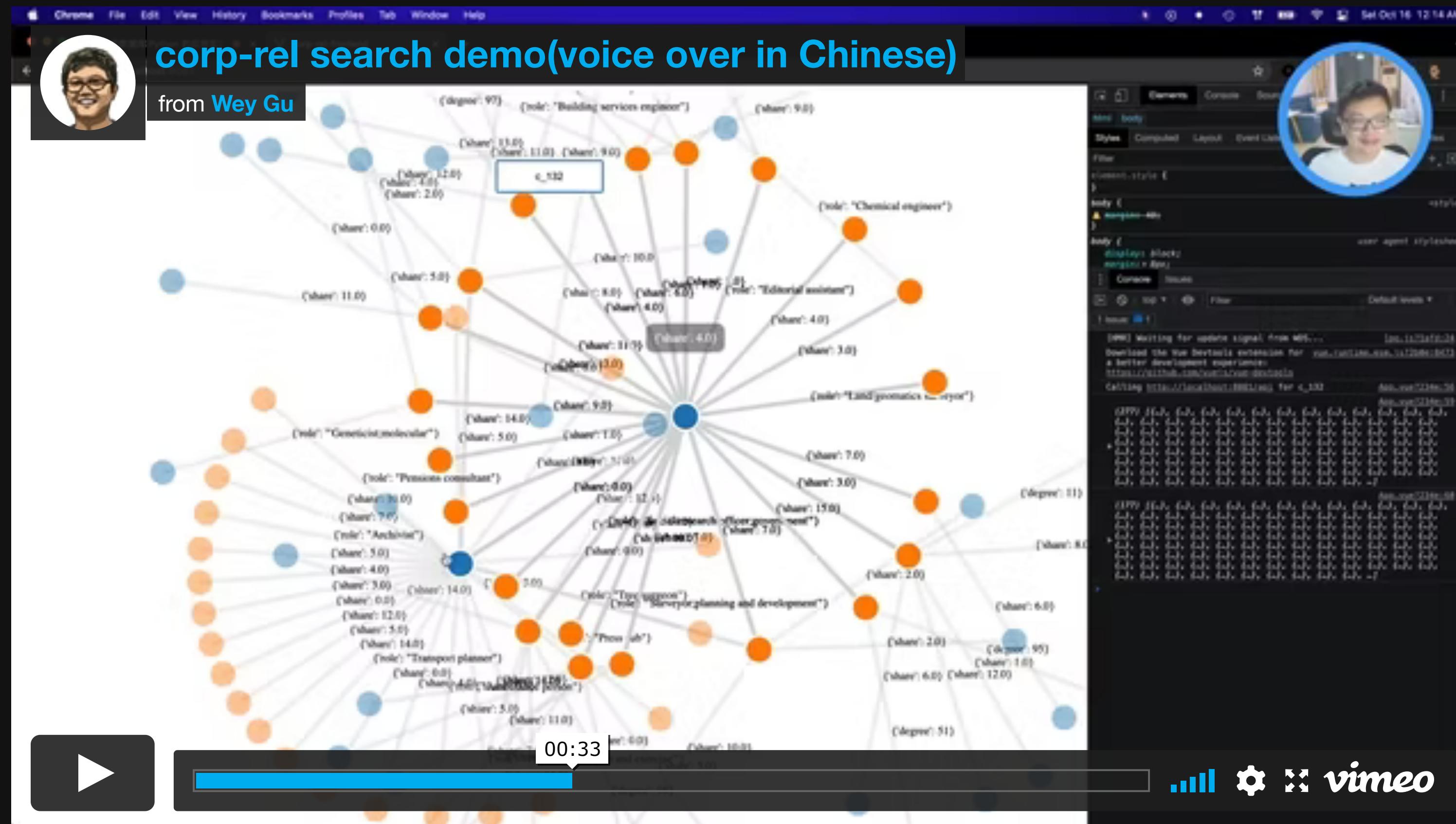


Demo

Corp-Rel Search



[github.com/wey-gu/**nebula-corp-rel-search**](https://github.com/wey-gu/nebula-corp-rel-search)



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



Siwi /'SIWI/

PoC: Dialog System With Graph Database Backed Knowledge Graph



[github.com/wey-gu/**nebula-siwi**](https://github.com/wey-gu/nebula-siwi)

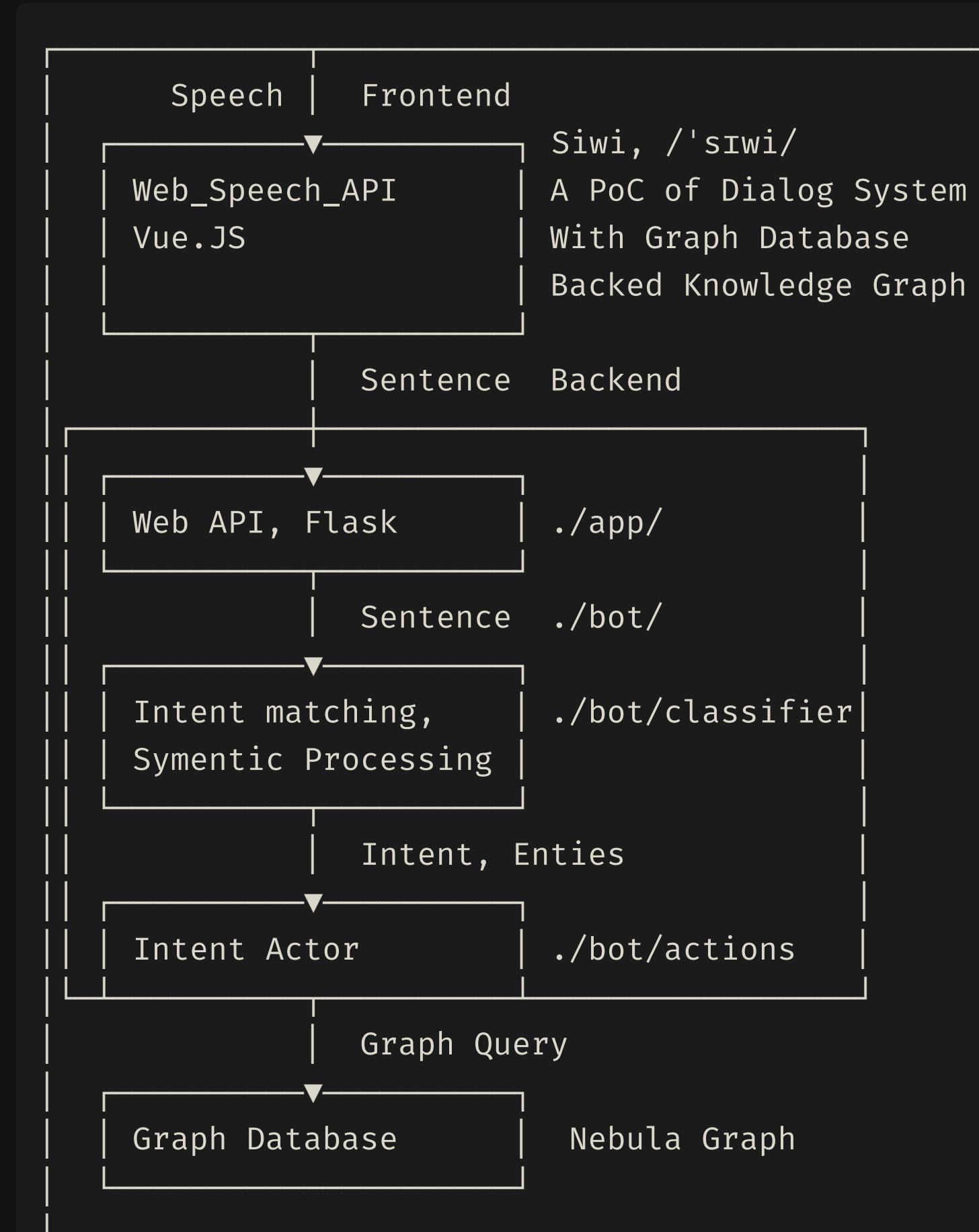


First Thing First: Graph Modeling

THE KNOWLEDGE GRAPH BEHIND SIWI IN NEBULA PLAYGROUND

 nebula-graph.io/demo/

ARCH



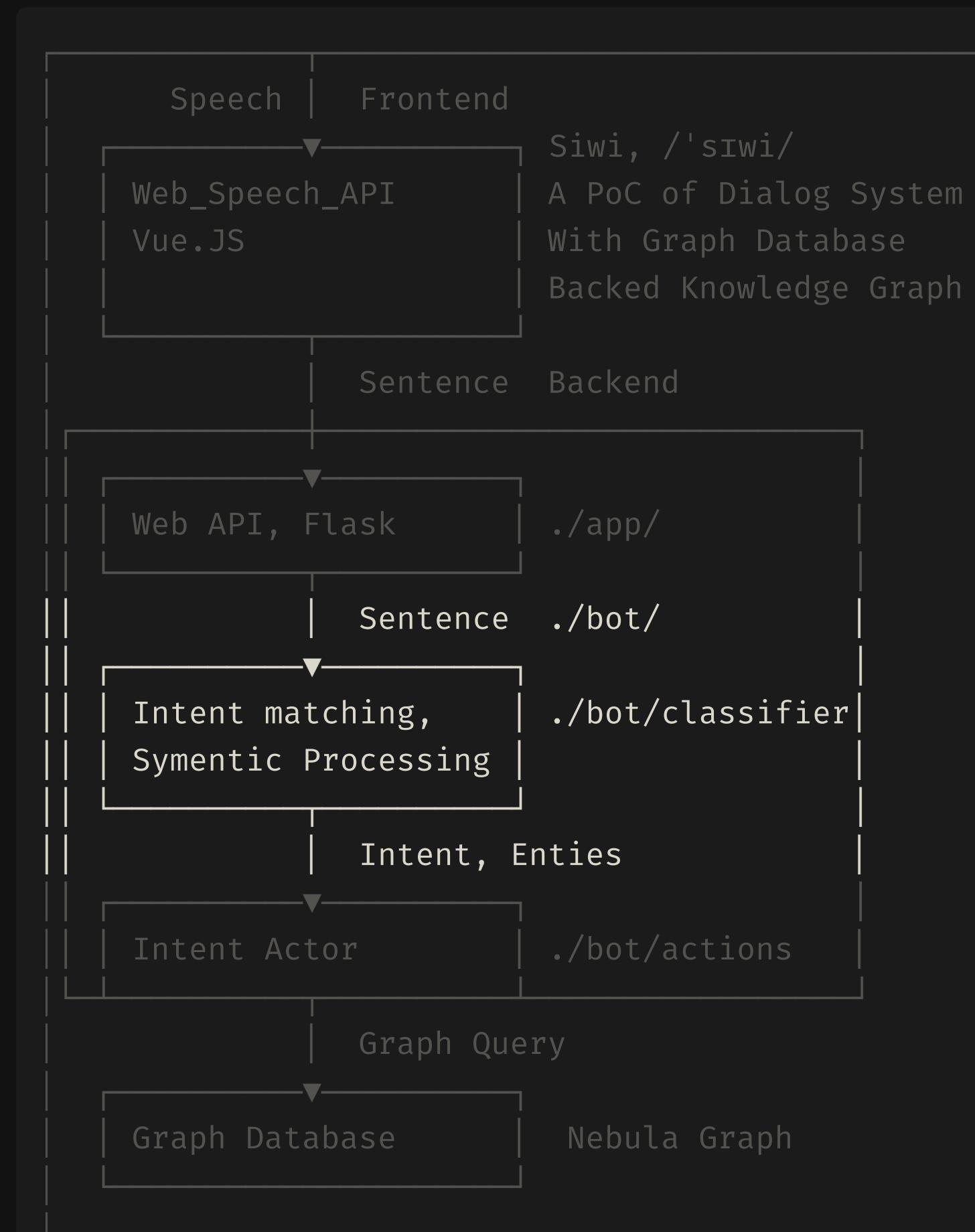
CODE

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



[wey-gu/nebula-siwi](https://github.com/wey-gu/nebula-siwi)

ARCH



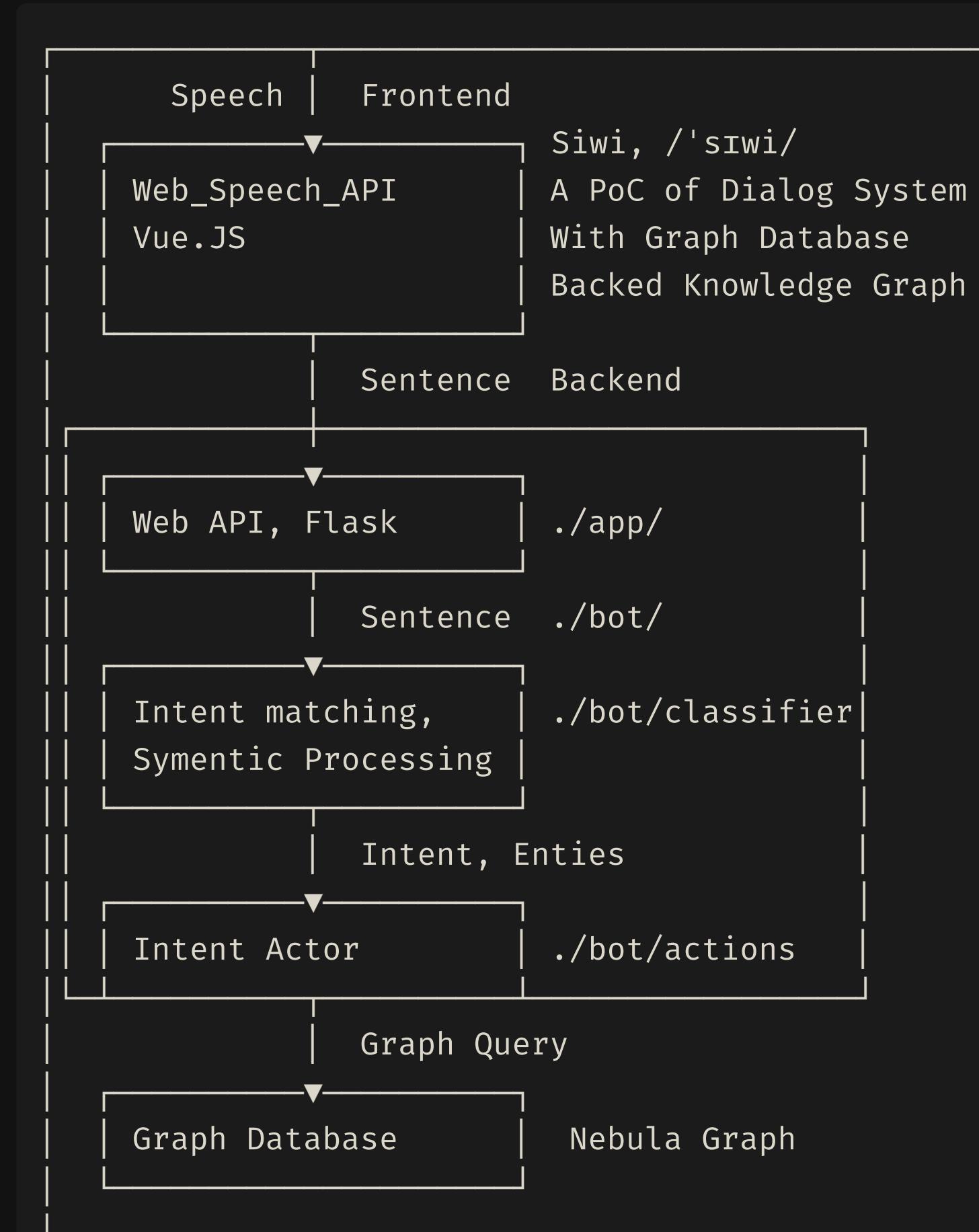
CODE

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



[wey-gu/nebula-siwi](https://github.com/wey-gu/nebula-siwi)

ARCH



CODE

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



[wey-gu/nebula-siwi](https://github.com/wey-gu/nebula-siwi)

Demo

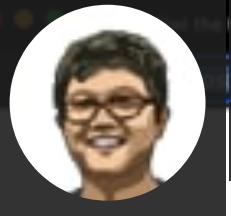
Siwi, Voice Agent with Nebula Graph on K8s



katacoda.com/wey/scenarios/siwi-kgqa

Chrome File Edit View History Bookmarks Profiles Tab Window Help

26 Fri Fri Nov 26 10:01 PM



siwi demo in katacoda(voice over in Chinese)

from **Wey Gu**

O'REILLY Katacoda

LEARN CREATE YOUR PROFILE LOG OUT >

Wey Gu, Scale the Magic to others!



@wey

I am a developer @vesoft working as Developer Advocate of Nebula Graph, the open-source distributed Graph Database I create toolings and content for Nebula Graph Database to help Developers in the open-source community. I am working in open source and consider it is a privilege 1. It took me a couple of my early career years to figure out that my passion lies in helping others with my thoughts & the tech/magic I have learned.

Share Your Success

Share Share

Shareholding Ownership Analysis with Nebula Graph Database

Breakdown multistage relationship of Persons and Corporations leveraging the Nebula Graph Database.

Start Scenario

Siwi the Knowledge Graph Dialog System with Nebula Graph

Setup a KGQA system from scratch with Nebula Graph, Vue.js, Flask on K8s

Start Scenario

05:53

vimeo

<https://katacoda.com/wey/scenarios/siwi-kgqa>



Open source data discovery and metadata engine



github.com/amundsen-io/amundsen

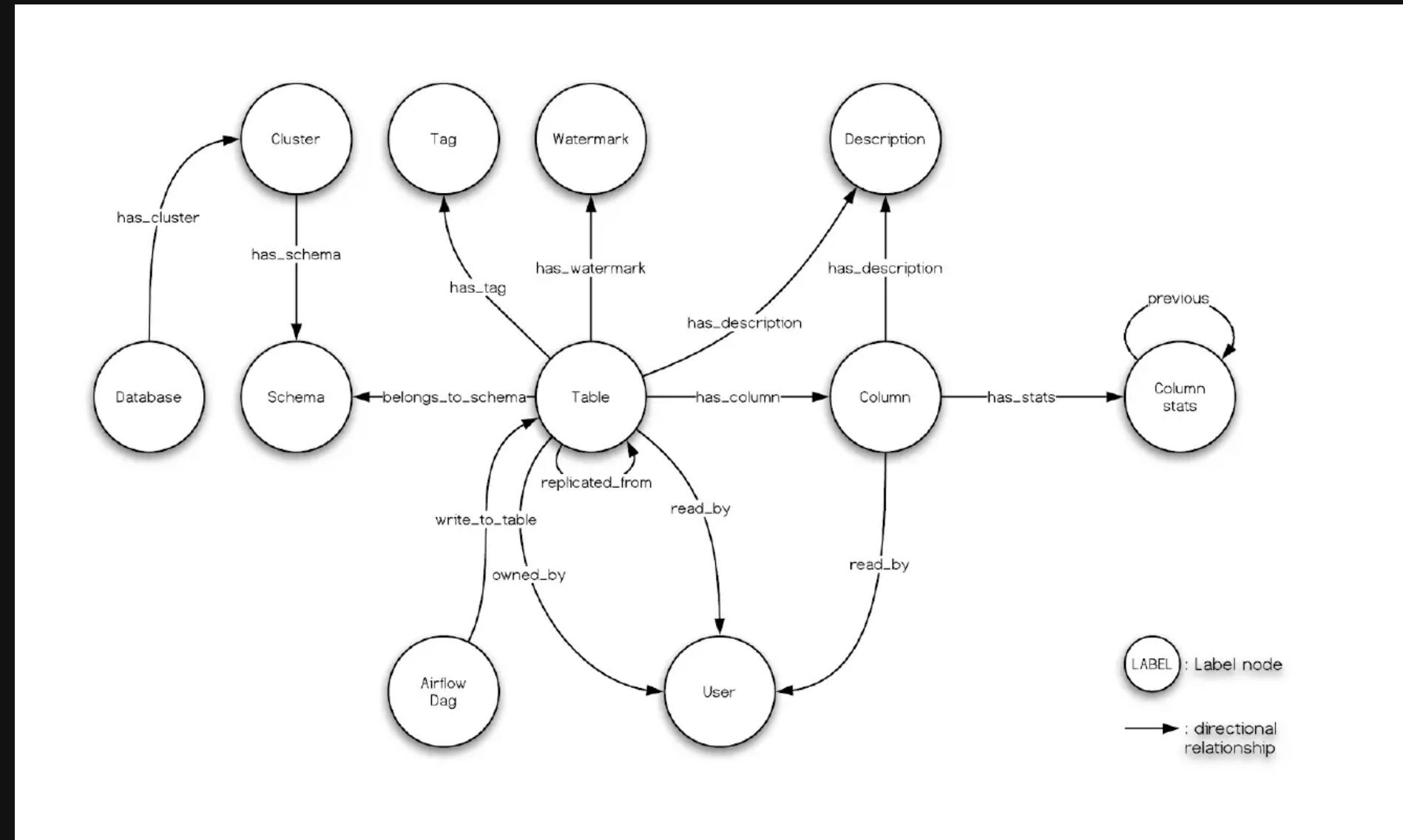


First Thing First: Graph Modeling

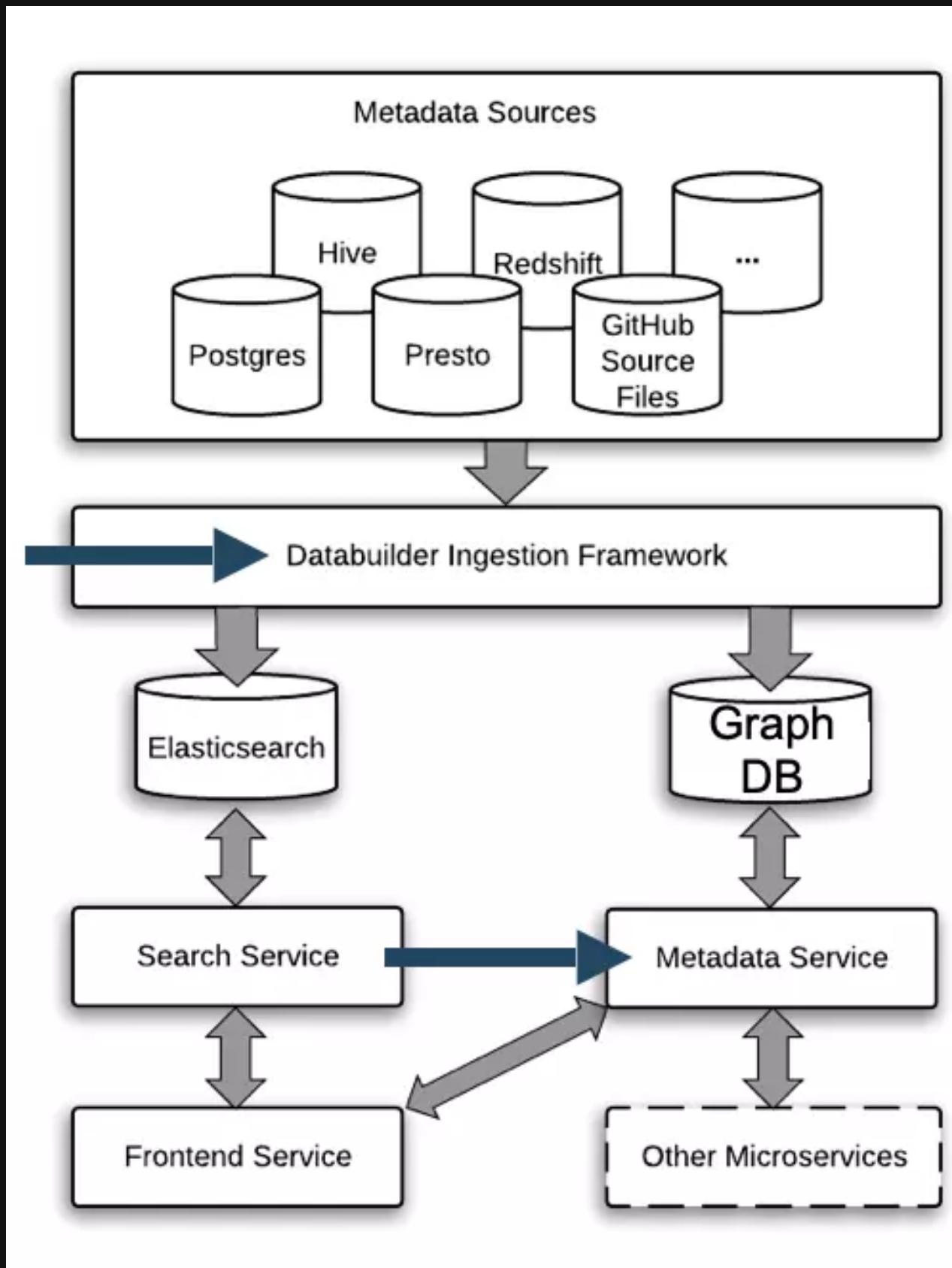
THE GRAPH MODEL OF AMUNDSEN



amundsen.io/amundsen/architecture/#metadata



ARCH



CODE

```
.  
|   common  
|   |   amundsen_common  
|   databuilder  
|   |   clients  
|   |   extractor  
|   |   loader  
|   |   models  
|   |   publisher  
|   |   rest_api  
|   |   utils  
|   docs  
|   frontend  
|   |   amundsen_application  
|   metadata  
|   |   metadata_service  
|   |   |   api  
|   |   |   cli  
|   |   |   client  
|   |   |   proxy  
|   |   |   util.py  
|   search  
|   |   search_service
```



Demo: Amundsen Frontend

Amundsen on Nebula Graph

Chrome File Edit View History Bookmarks Profiles Tab Window Help

26 Sun 26 Sun Dec 26 3:02 AM

Amundsen - Data Discovery Platform

Not Secure | 192.168.8.127:5000

AMUNDSEN

Browse

Search for data resources...

Advanced Search >

Available Badges

Beta Fk json Npl Pl Pk

Popular Tags

cheap 1 delta 1 expensive 1 low_quality 1 needs_documentation 2 recommended 1 tag1 2 tag2 1

Browse all tags

My Bookmarks

Datasets (0)

You don't have any bookmarks. Use the star icon to save a bookmark.

Popular Resources ⓘ

00:44

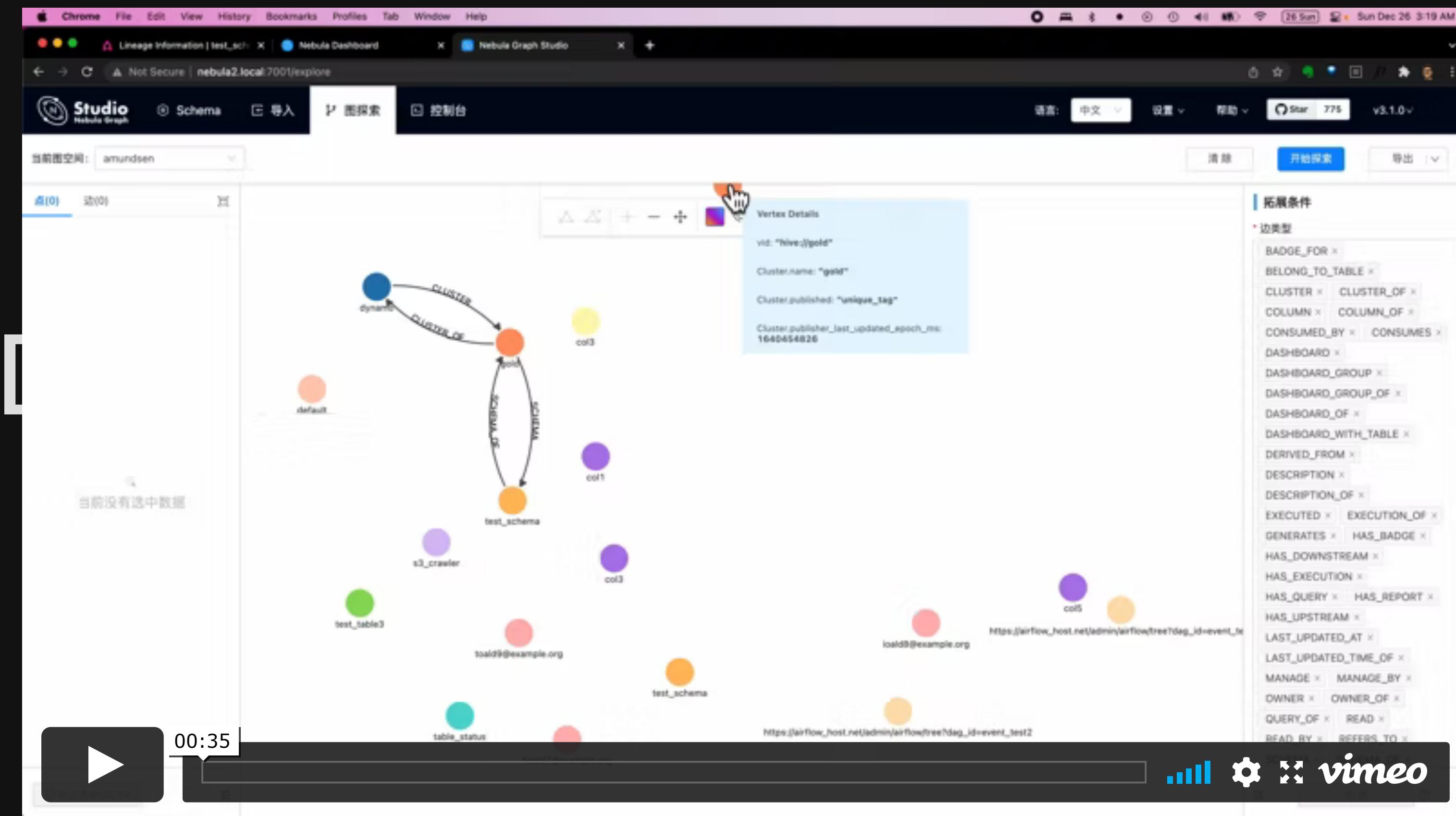
Datasets (0)

Amundsen was last indexed on December 26, 2021 at 11:00:00 am.

vimeo

Demo: Visualization in Studio

Amundsen on Nebula Graph



Demo: Query in Nebula Graph

Amundsen on Nebula Graph

D

```
Alacritty
~/.dev/nebula-console
~/dev/nebula-console remotes/origin/master~1*
> bat ~/Downloads/demo.ngql
File: /Users/weyl/Downloads/demo.ngql

USE amundsen;
MATCH (tbl:Table) \
    WHERE id(tbl) == "hive://gold.test_schema/test_table1" \
        OPTIONAL MATCH (wmk:Watermark)-[:BELONG_TO_TABLE]-(tbl) \
        OPTIONAL MATCH (app_producer:Application)-[:GENERATES]-(tbl) \
        OPTIONAL MATCH (app_consumer:Application)-[:CONSUMES]-(tbl) \
        OPTIONAL MATCH (tbl)-[:LAST_UPDATED_AT]-(t:'Timestamp') \
        OPTIONAL MATCH (owner:'User')<-[:OWNER]-(tbl) \
        OPTIONAL MATCH (tbl)-[:TAGGED_BY]->(`tag`:'Tag') \
        OPTIONAL MATCH (tbl)-[:HAS_BADGE]->(badge:Badge) \
        OPTIONAL MATCH (tbl)-[:SOURCE]->(src:Source) \
        OPTIONAL MATCH (tbl)-[:DESCRIPTION]->(prog_descriptions:Programmatic_Description) \
        OPTIONAL MATCH (tbl)-[:HAS_REPORT]->(resource_reports:Report) \
    RETURN collect(distinct wmk) as wmk_records, \
        collect(distinct app_producer) as producing_apps, \
        collect(distinct app_consumer) as consuming_apps, \
        t.last_updated_timestamp as last_updated_timestamp, \
        collect(distinct owner) as owner_records, \
        collect(distinct `tag`) as tag_records, \
        collect(distinct badge) as badge_records, \
        src, \
        collect(distinct prog_descriptions) as prog_descriptions, \
        collect(distinct resource_reports) as resource_reports

~/dev/nebula-console remotes/origin/master~1*
```

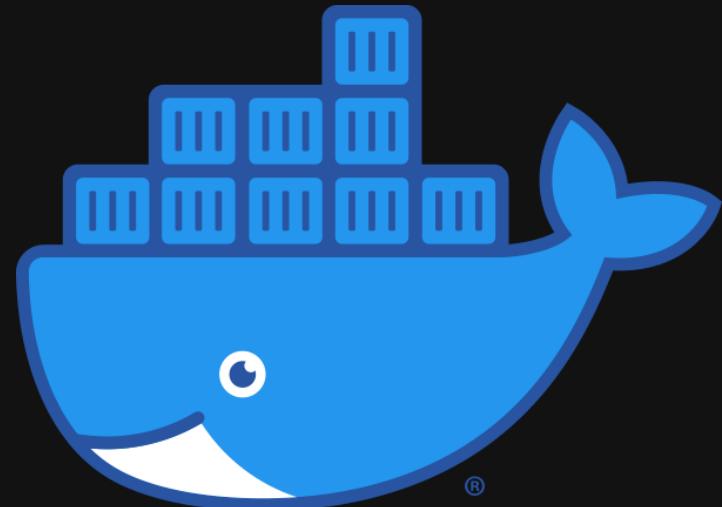


00:09



Nebula Graph 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
```

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
```

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-Kind: Oneliner Nebula-Operator

```
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>        9669:30000/TCP,19669:30001/TCP  3m57s
```

```
$ ~/.nebula-kind/bin/console -u root -p password --address=127.0.0.1 --port=30000
2022/02/01 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



Nebula-Kind: Oneliner Nebula-Operator

```
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>        9669:30000/TCP,19669:30001/TCP  3m57s
```

```
$ ~/.nebula-kind/bin/console -u root -p password --address=127.0.0.1 --port=30000
2022/02/01 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



Nebula-Kind: Oneliner Nebula-Operator, Cont.

```
$ wget https://docs.nebula-graph.io/2.0/basketballplayer-2.X.ngql
$ ~/.nebula-kind/bin/console -u root -p password --address=127.0.0.1 --port=30000 -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)
Tue, 01 Feb 2022 20:47:58 UTC
```

```
[root@wey wey.gu]# ~/.nebula-kind/bin/console -u root -p password --address=127.0.0.1 --port=30000
(root@nebula) [(none)]> show spaces
+-----+
| Name |
+-----+
| "basketballplayer" |
+-----+
(root@nebula) [(none)]> use basketballplayer
(root@nebula) [basketballplayer]> show tags
+-----+
| Name |
+-----+
| "player" |
+-----+
| "team" |
+-----+
```



Doc: Sample Dataset

Nebula-Kind: Oneliner Nebula-Operator, Cont.

```
$ wget https://docs.nebula-graph.io/2.0/basketballplayer-2.X.ngql
$ ~/.nebula-kind/bin/console -u root -p password --address=127.0.0.1 --port=30000 -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)
Tue, 01 Feb 2022 20:47:58 UTC
```

```
[root@wey wey.gu]# ~/.nebula-kind/bin/console -u root -p password --address=127.0.0.1 --port=30000
(root@nebula) [(none)]> show spaces
+-----+
| Name |
+-----+
| "basketballplayer" |
+-----+
(root@nebula) [(none)]> use basketballplayer
(root@nebula) [basketballplayer]> show tags
+-----+
| Name |
+-----+
| "player" |
+-----+
| "team" |
+-----+
```

Nebula-Kind: Oneliner Nebula-Operator, Cont.

```
$ wget https://docs.nebula-graph.io/2.0/basketballplayer-2.X.ngql
$ ~/.nebula-kind/bin/console -u root -p password --address=127.0.0.1 --port=30000 -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)
Tue, 01 Feb 2022 20:47:58 UTC
```

```
[root@wey wey.gu]# ~/.nebula-kind/bin/console -u root -p password --address=127.0.0.1 --port=30000
(root@nebula) [(none)]> show spaces
+-----+
| Name |
+-----+
| "basketballplayer" |
+-----+
(root@nebula) [(none)]> use basketballplayer
(root@nebula) [basketballplayer]> show tags
+-----+
| Name |
+-----+
| "player" |
+-----+
| "team" |
+-----+
```



Doc: Sample Dataset

Recap

- Graph, WHY Graph DB? 🎅
- Nebula Graph Design/Arch and Why Nebula Graph? 🚢 ⚓
- (Nebula Graph) Know How
 - 💰: Corp-Rel Search
 - 🤖: Siwi the Basketball Robot
 - 🔍: LF Amundsen, the Metadata Mgmt. Engine
- Nebula Operator ⚙



DoK
Community

siwei.io/talks/2022-DoKC/

Feb. 11 2022

Recap

- Graph, WHY Graph DB? 🎅
- Nebula Graph Design/Arch and Why Nebula Graph? 🚢 ⚓
- (Nebula Graph) Know How
 - 💰: Corp-Rel Search
 - 🤖: Siwi the Basketball Robot
 - 🔍: LF Amundsen, the Metadata Mgmt. Engine
- Nebula Operator ⚙

😺 wey-gu

🐦 wey_gu

👤 siwei.io



DoK
Community

siwei.io/talks/2022-DoKC/

Feb. 11 2022

Thanks!

Join the Community Meeting:



nebulagraph.slack.com

January							February							March										
Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	
							1		1	2	3	4	5		6	7	8	9	10	11	12	6	7	8
2	3	4	5	6	7	8	6	7	[8]	9	10	11	12	6	7	8	9	10	11	12	13	14	15	
9	10	11	12	13	14	15	13	14	15	16	17	18	19	13	14	15	16	17	18	19	20	21	22	
16	17	18	19	20	21	22	20	21	[22]	23	24	25	26	20	21	[22]	23	24	25	26	27	28	29	
23	24	[25]	26	27	28	29	27	28						27	28	29								

📢 Nebula Graph Cloud will be is Open Beta soon
TODAY in Microsoft Azure!

🐱 [vesoft-inc/nebula](https://github.com/vesoft-inc/nebula)

🐦 [@wey_gu](https://twitter.com/wey_gu)

PPT siwei.io/talks/2022-DoKC/

