# 图数据库解谜与数据质量应用

古思为

DEVELOPER ADVOCATE @ vesoft

iSQE 国际软件质量工程
International Software Quality Engineering

# 古思为

- Software Engineer @ Shanghai
- Open Source Believer
- Developer Advocate of NebulaGraph @vesoft

wey-gu

wey_gu

siwei.io/**about**

iSQE

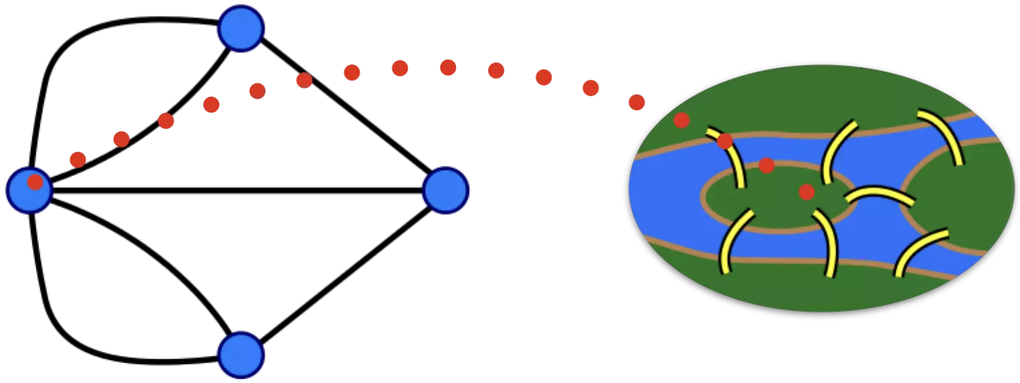
# Overview

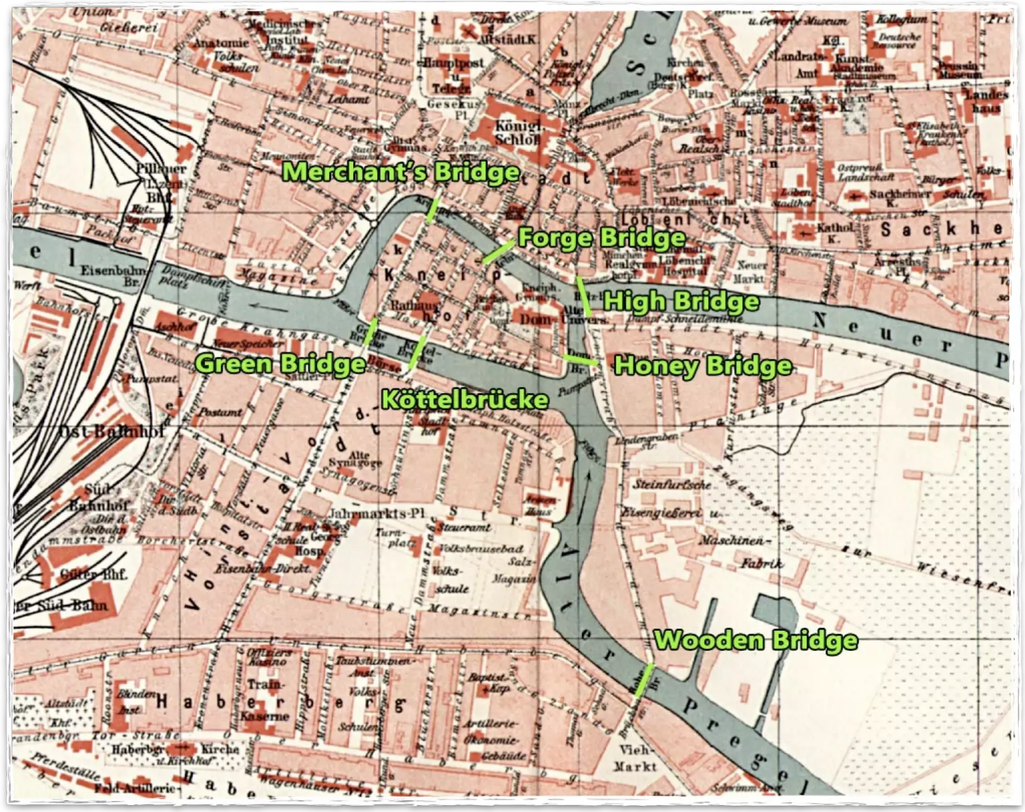
- Graph Database Explained
- Graph Tech Application: Data Lineage

iSQE

# Graph Database

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






*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

More on what a GDB is

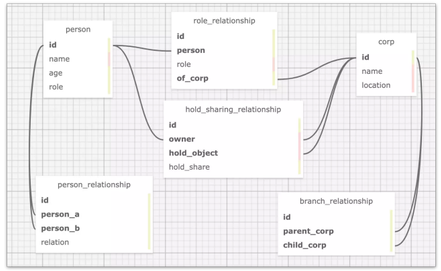
iSQE

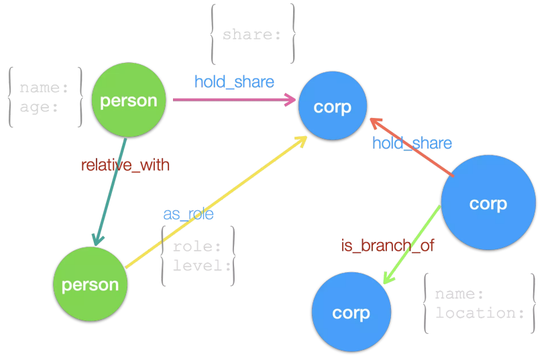
# Why Yet Another DB?

## Relational DB

## Graph DB

Graph Schema




Graph Semantic Queries

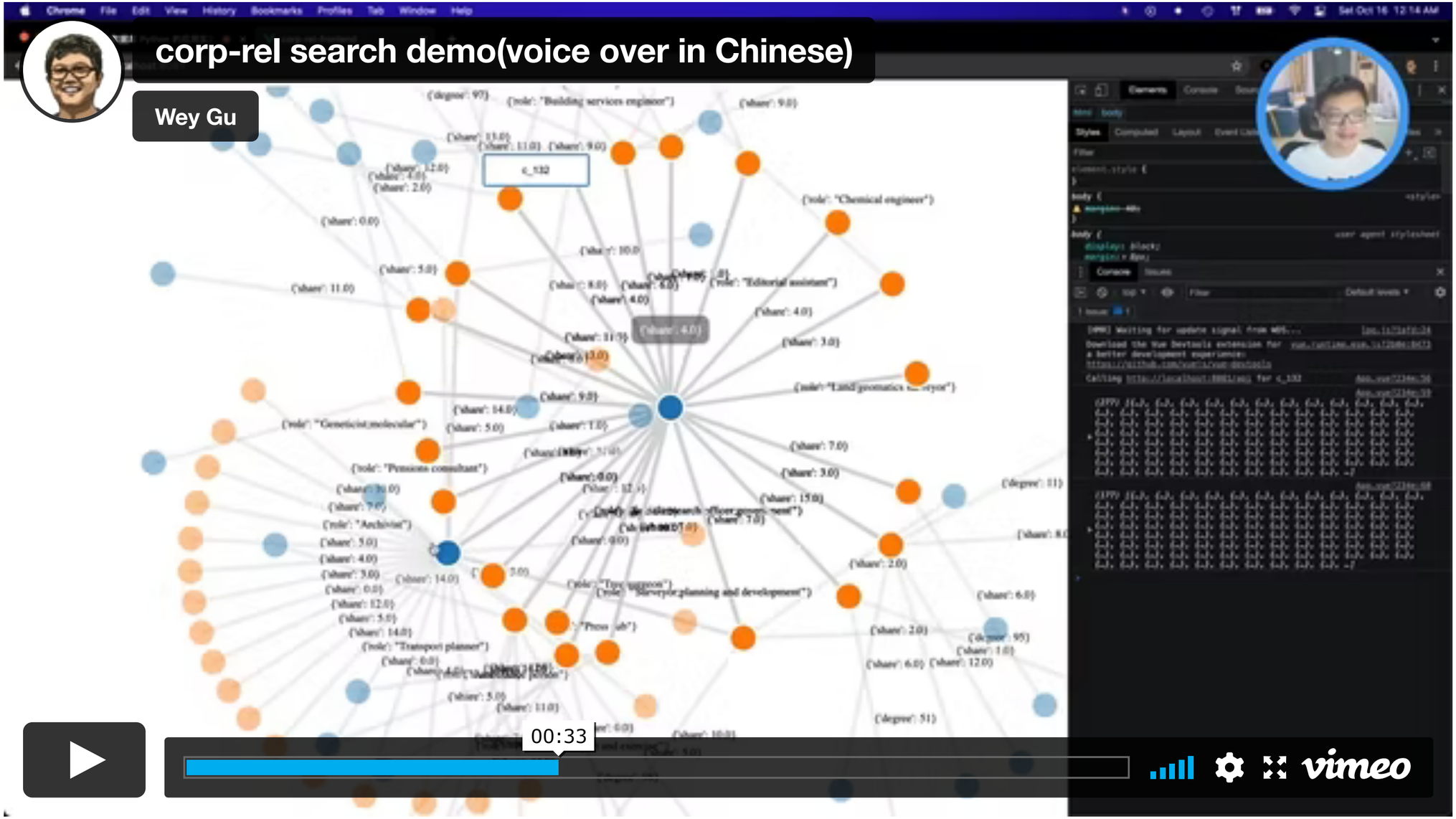
```
SELECT a.id, a.name, c.name
FROM player a
JOIN serve b ON a.id=b.player_id
JOIN team c ON c.id=b.team_id
WHERE c.name IN (SELECT c.name
FROM player a
JOIN serve b ON a.id=b.player_id
JOIN team c ON c.id=b.team_id
WHERE a.name = 'Tim Duncan')
```

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

Performance

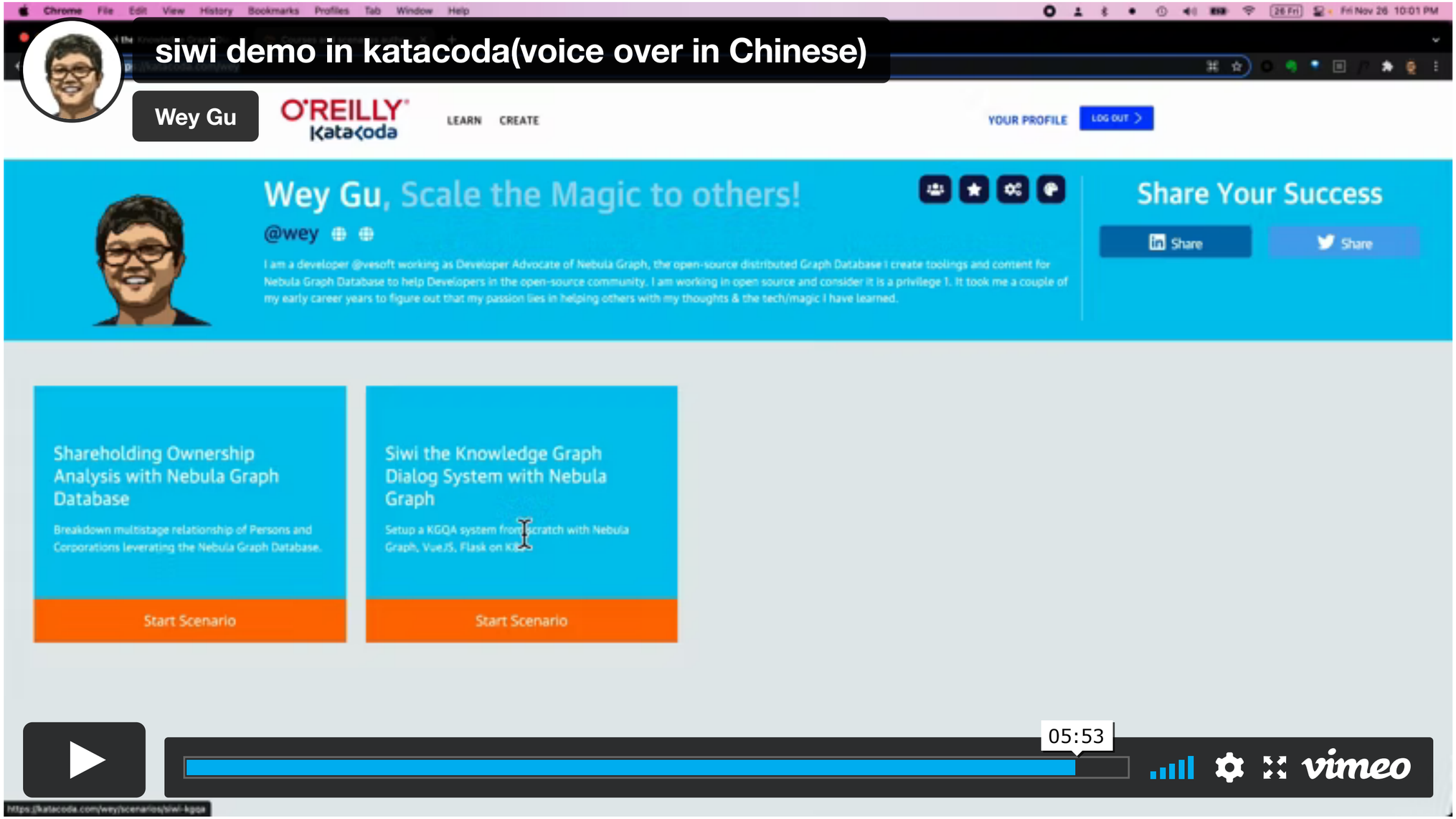
| | Designed Scenario | 2-hop latency (~2.5K) | 3-hop latency (~110K) | 4-hop latency (~600K) |
|---|---|---|---|---|
| **Graph DB** | Relationship Walk | 0.01 sec | 0.168 sec | 1.36 sec |
| **SQL DB** | Information retrieval | 0.016 sec | 30 sec | 1544 sec |

iSQE




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

iSQE




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

iSQE

# 挑战：


Graph DB nGQL

iSQE

db-engines.com/en/**ranking_categories**

# 挑战：数据规模

NebulaGraph is **highly performant** yet **linearly scalable** as it's designed **shared**

| Data Volume | |
|---|---|
| 10 million Edges | Neo4j |
| | HugeGra |
| | Nebula G |
| 100 million Edges | Neo4j |
| | HugeGra |
| | Nebula G |
| 1 billion Edges | Neo4j |
| | HugeGra |
| | Nebula G |
| 8 billion Edges | Neo4j |
| | HugeGra |
| | Nebula G |

**Real Time Write Performance**

| 实时写入<br>最大吞吐量 | 插入点<br>qps | 插入边<br>qps |
|---|---|---|
| Nebula | 84000 | 76000 |
| DGraph | 10138 | 9600 |
| HugeGraph | 410 | 457 |

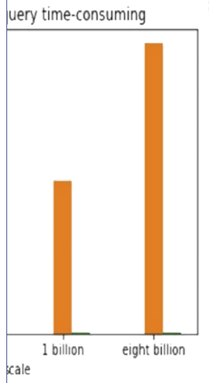
| 实时写入<br>响应时间(ms) | 插入点 qps=100<br>avg / p99 / p999 | 插入边 qps=100<br>avg / p99 / p999 |
|---|---|---|
| Nebula | 3 / 10 / 38 | 3 / 10 / 34 |
| DGraph | 5 / 13 / 54 | 5 / 9 / 43 |
| HugeGraph | 37 / 159 / 1590 | 28 / 93 / 1896 |

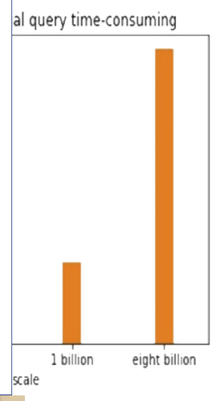
**N-hop Read Queries with Properties**

| N 跳查询 返回属性<br>最大吞吐量(qps) | n = 1<br>平均返回邻居数=62 | n = 2<br>平均返回邻居数=3844 | n = 3<br>平均返回邻居数=238328 |
|---|---|---|---|
| Nebula | 36400 | 730 | 2 |
| Dgraph | 80 | 60s 内未返回 | 60s 内未返回 |
| HugeGraph | 802 | 60s 内未返回 | 60s 内未返回 |

| N 跳查询 返回属性<br>响应时间(ms) | n = 1, qps=100<br>avg / p99 / p999 | n = 2, qps=100<br>avg / p99 / p999 | n = 3, qps=10<br>avg / p99 / p999 |
|---|---|---|---|
| Nebula | 5 / 20 / 49 | 99 / 475 / 1645 | 48052 / >60s / >60s |
| Dgraph | 50 / 210 / 274 | 60s 内未返回 | 60s 内未返回 |
| HugeGraph | 29 / 219 / 1555 | 60s 内未返回 | 60s 内未返回 |

| 共同好友<br>响应时间(ms) | qps = 100<br>avg / p99 / p999 |
|---|---|
| Nebula | 4 / 20 / 42 |
| Dgraph | 4 / 13 / 36 |
| HugeGraph | 39 / 366 / 1630 |




ⓘ nebula-graph.io/cases/

**Wey Gu 古思为**
@wey_gu · **Follow**

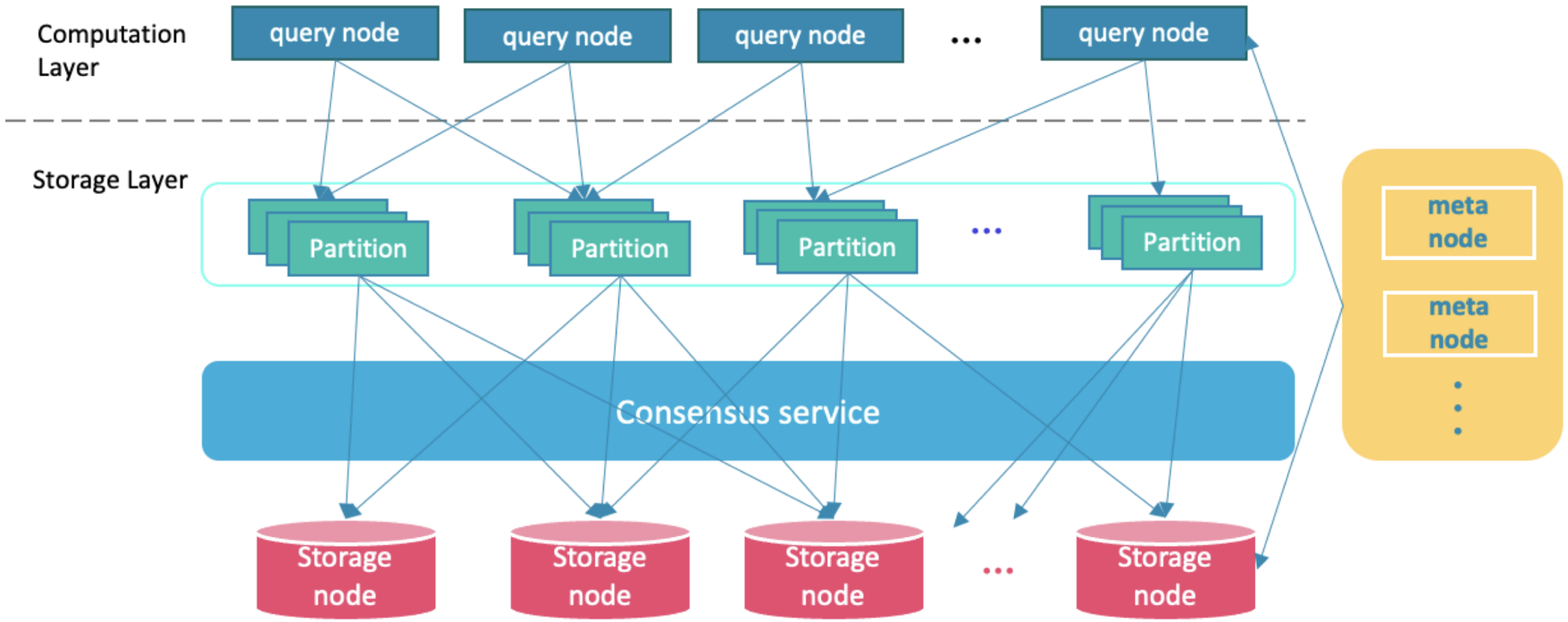
Scale makes differences **@NebulaGraph**

**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.

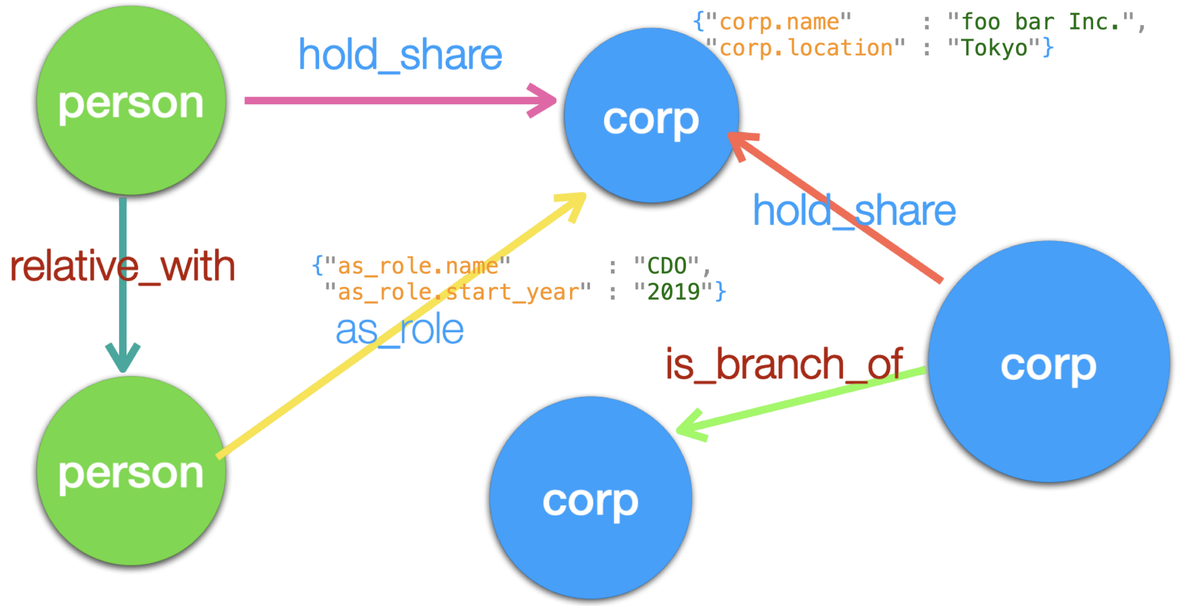
# Distributed GraphDB Architecture

- Separated Compute and Storage: Scale Independently
- Shared-Nothing Architecture: Scale Easily and Fault Tolerance
- Auto Sharding(Hash Key Range, shards are way more than nodes)
- Raft(For both Meta and Storage nodes)



# Graph Database Data Model

- Property Graph Data Model
- Schema:
  - Tag defines a set of properties for a vertex
  - Edge Type defines a set of properties for a edge
- Vertex
  - A vertex is a node in the graph, with a Vertex ID
  - 0, or more Tags could be attached to a vertex
- Edge
  - An edge is a 4-tuple of (Vertex ID, Edge Type, Vertex ID, Rank)
  - Edge connected two vertices in given type and rank



# Graph Query Language

## NGQL

```
# Traversing graph
GO 3 Steps FROM "player102" OVER follow YIELD dst(edge);
```

```
# Index based
LOOKUP ON player WHERE player.name == "Tony Parker" \
  YIELD id(vertex);
```

```
# Directly retrieve properties:
FETCH PROP ON player "player100" YIELD properties(vertex);
```

```
FIND SHORTEST PATH FROM "player102" TO "team204" OVER * \
  YIELD path AS p;

GET SUBGRAPH 5 STEPS FROM "player101" \
  YIELD VERTICES AS nodes, EDGES AS relationships;
```

```
# Connected With Pipeline
GO FROM "player100" OVER follow \
  YIELD dst(edge) AS did, properties($$).name AS Name | \
  GO FROM $-.did OVER follow YIELD dst(edge);
```

## OPENCYPHER

```
MATCH (v:player{name:"Tim Duncan"})⟶(v2)⟵(v3) \
    RETURN v3.player.name AS Name;
```

```
MATCH (v:player) \
    WHERE not (v)--() \
    RETURN v;
```

```
MATCH (v:player)--(v2) \
    WHERE id(v2) IN ["player101", "player102"] \
    RETURN v;
```

```
MATCH (m)-[]→(n) WHERE id(m)=="player100" \
OPTIONAL MATCH (n)-[]→(l) WHERE id(n)=="player125" \
    RETURN id(m),id(n),id(l);
```

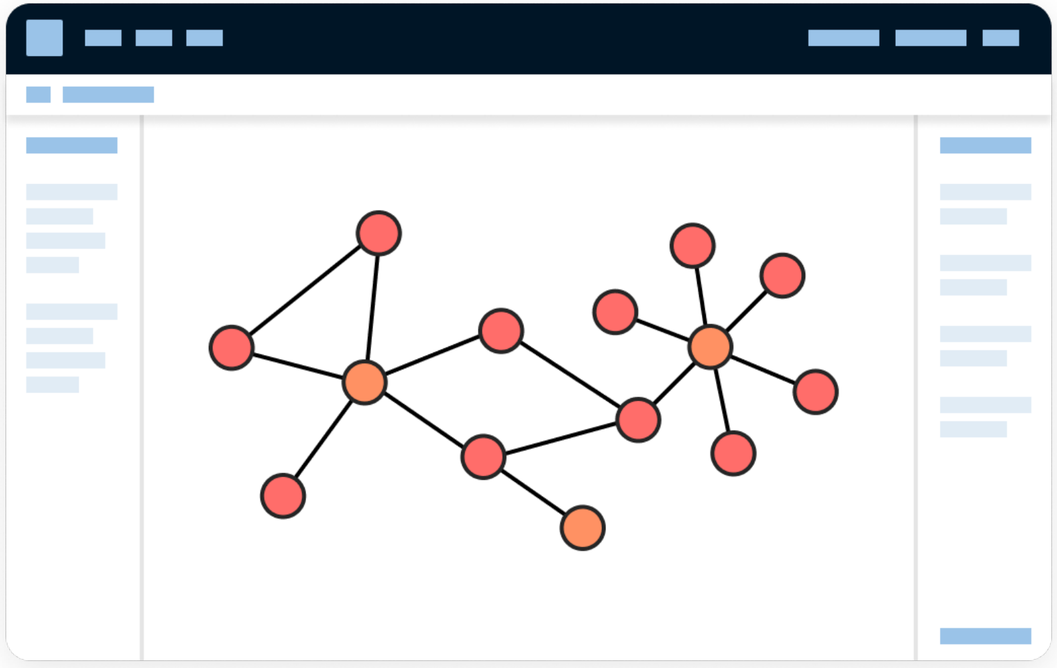
## WILL BE FULLY COMPATIBLE WITH FUTURE ISO GQL STANDARD

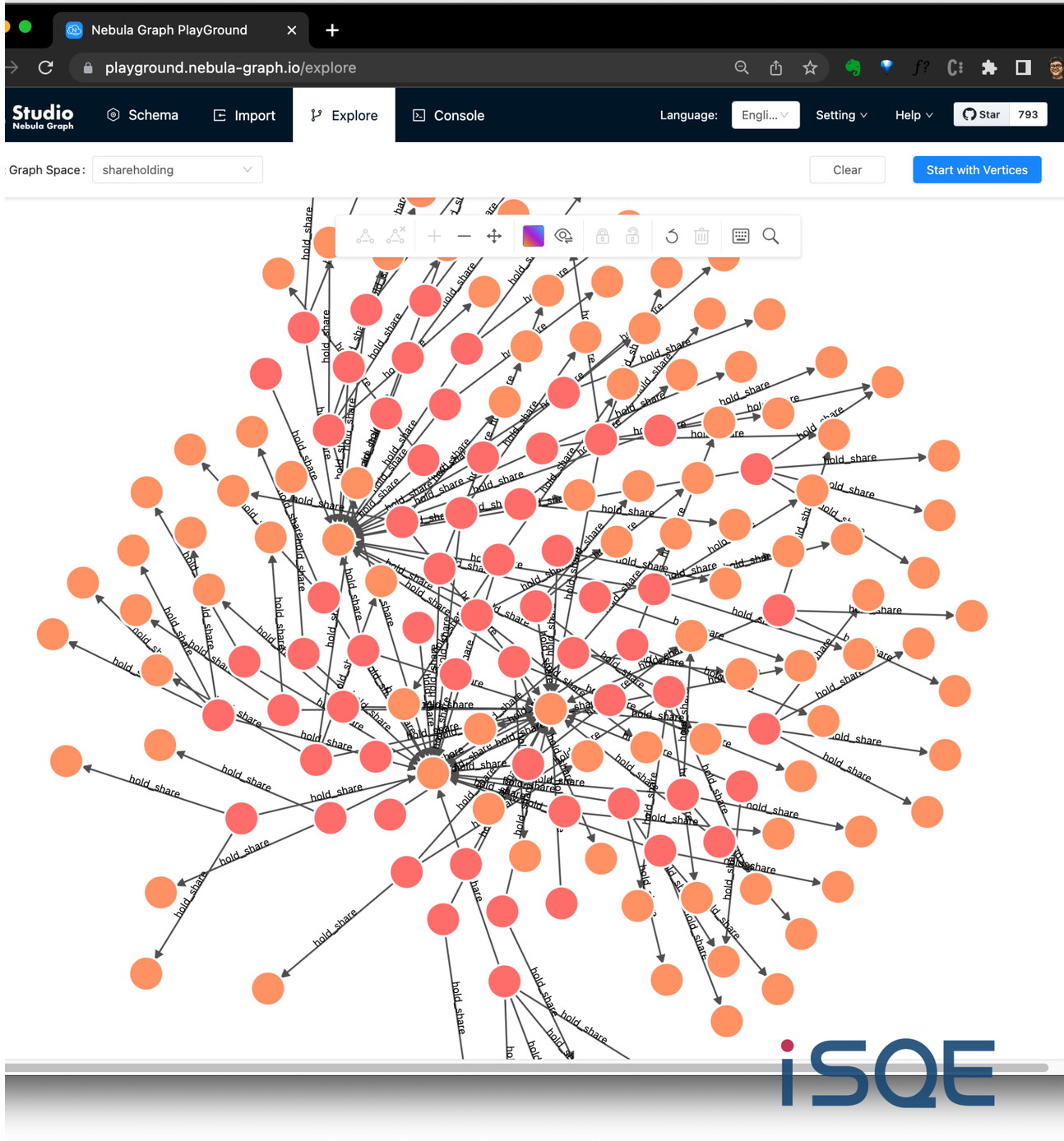
iSQE

# 优势： Data Visualization

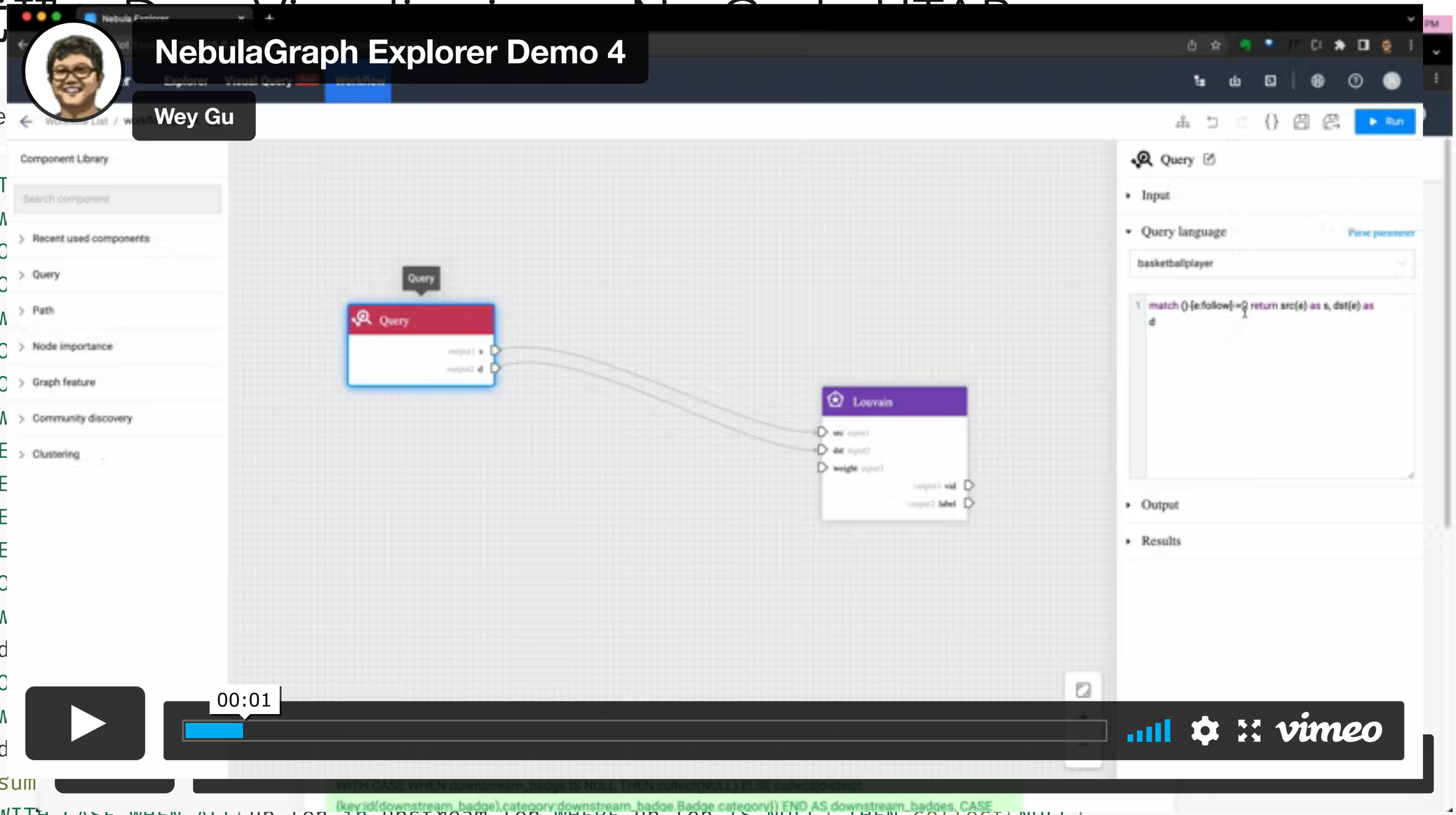
> Get one's 3-depth equity relationship with 1+% share

```
MATCH p=(:person{name: "Debra Ingram"})-[e*1..3]-(x)
  WHERE ALL(e_ in e WHERE e_.share > 1.0)
  RETURN p,e
```





# 优势



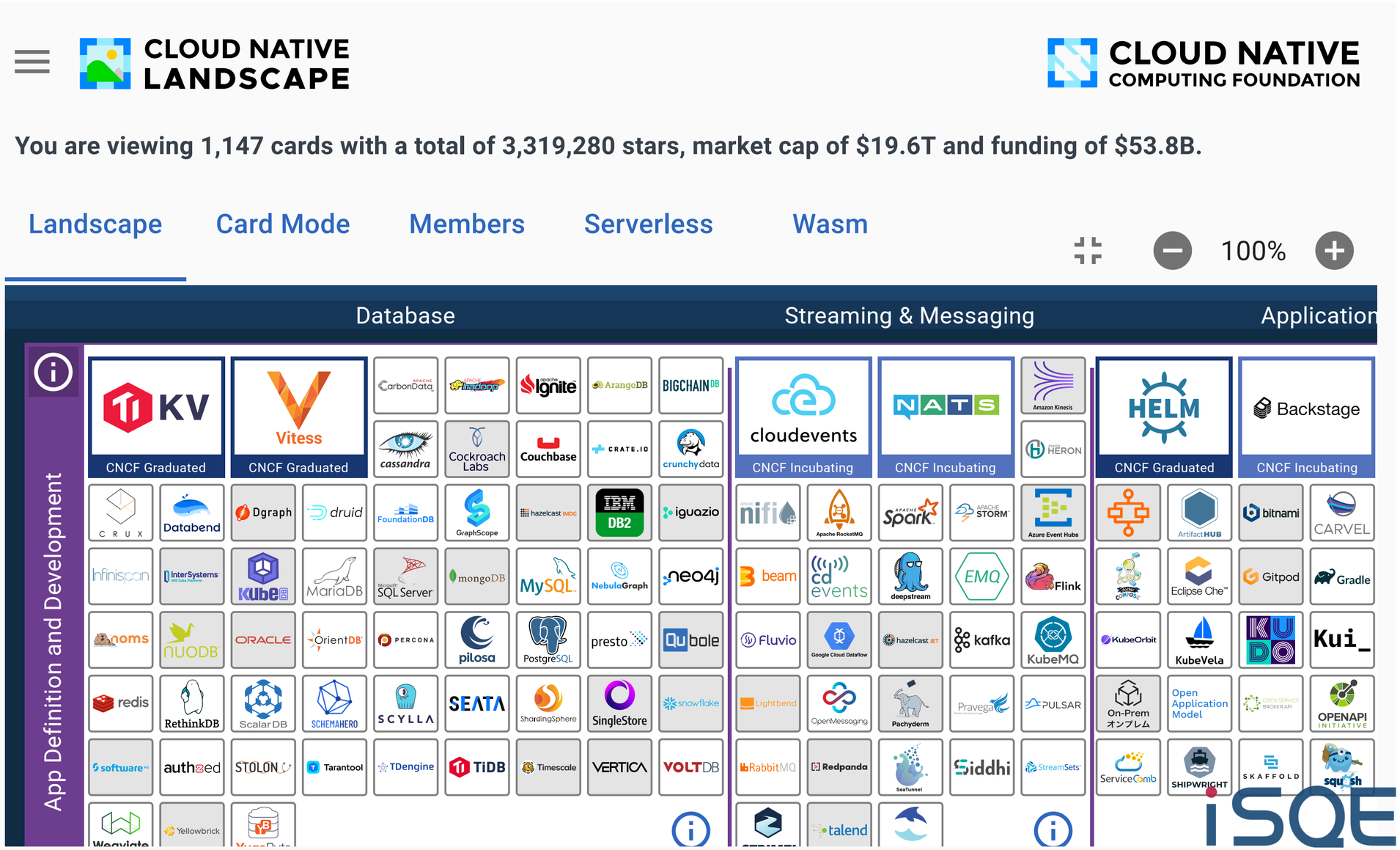

```
MAT
  W
  C
  C
  W
  C
  C
  W
  E
  E
  E
  E
  C
  W
  d
  C
  W
  d
  Sum
WITH CASE WHEN ALL(up_len IN upstream_len WHERE up_len IS NULL) THEN collect(NULL)
ELSE COLLECT(distinct {level:SIZE(upstream_len), source:split(id(upstream_entity),'://')[0],
key:id(upstream_entity), badges:upstream_badges, usage:upstream_read_count, parent:id(nodes(upath)[-2])})
END AS upstream_entities, CASE WHEN ALL(down_len IN downstream_len WHERE down_len IS NULL) THEN collect(NULL)
```

iSQE

# 优势：SRE, Cloud Native

- Nebula K8s Oper
- Ansible/Docker-C
- Grafana
- Prometheus
- Stats Exporter
- Backup & Restore

> Nebula enriches the CN



# 优势：数据管道

- Importer(`.csv`)/Console(`.ngql`)
- Exchange
  - Source: Postgres, Kafka, Clickhouse etc.
  - Target: NebulaGraph Server/SST File
- Connector/Clients
  - Spark/Flink
  - Python/Java/Go/CPP
  - NodeJS/PHP/.NET/JDBC
  - HTTP-Gateway
  - ORM: GO/Java*


## What is Nebula Exchange

Nebula Exchange (Exchange) is an Apache Spark™ application for bulk migration of cluster data to Nebula Graph in a distributed environment, supporting batch and streaming data migration in a variety of formats.

Exchange consists of Reader, Processor, and Writer. After Reader reads data from different sources and returns a DataFrame, the Processor iterates through each row of the DataFrame and obtains the corresponding value based on the mapping between `fields` in the configuration file. After iterating through the number of rows in the specified batch, Writer writes the captured data to the Nebula Graph at once. The following figure illustrates the process by which Exchange completes the data conversion

# Graph Algorithm

Open Source state-of-the-art Graph Algorithm Platform on Spark consuming NebulaGraph.

- PageRank/ Louvain/ KCore
- LabelPropagation/ Hanp
- ConnectedComponent/ Strongly ConnectedComponent
- ShortestPath/ BFS
- TriangleCount/ GraphTriangleCount
- BetweennessCentrality/ Closeness
- DegreeStatic
- ClusteringCoefficient
- Jaccard
- Node2Vec

# Nebula Analytics


## Nebula Analytics

Nebula Analytics is a high-performance graph computing framework tool that performs graph analysis of data in the Nebula Graph database.

**Enterpriseonly**

Only available for the Nebula Graph Enterprise Edition.

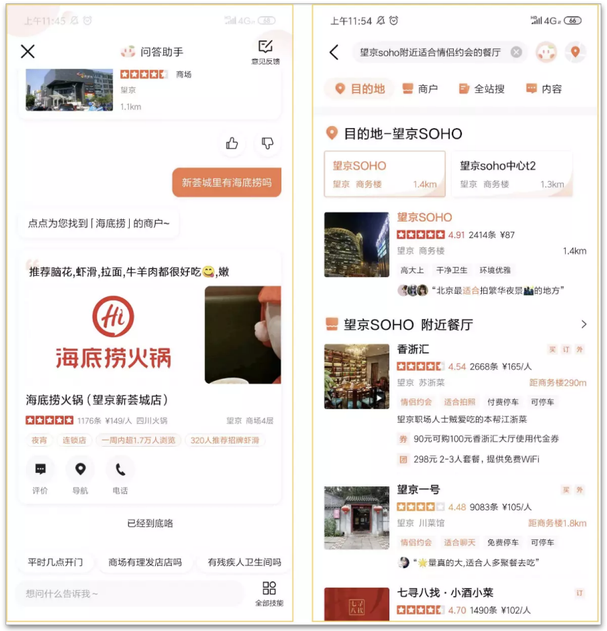
## Scenarios

iSQE

# 广泛的应用

" Is there any McDonald's nearby street Foofar?

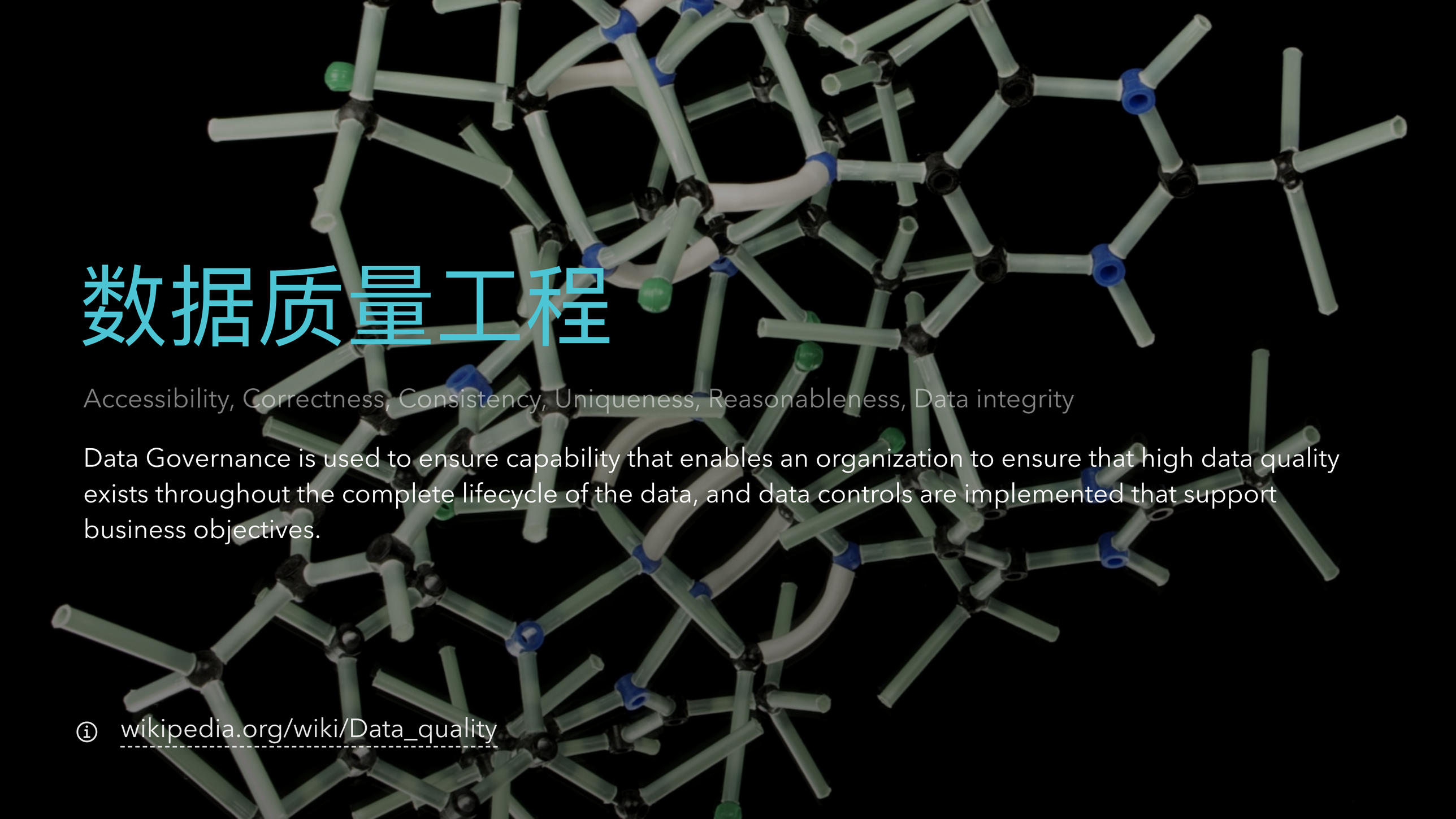
" Those who from Beijing enjoying the Korean cuisine said this restaurant is great.





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

iSQE

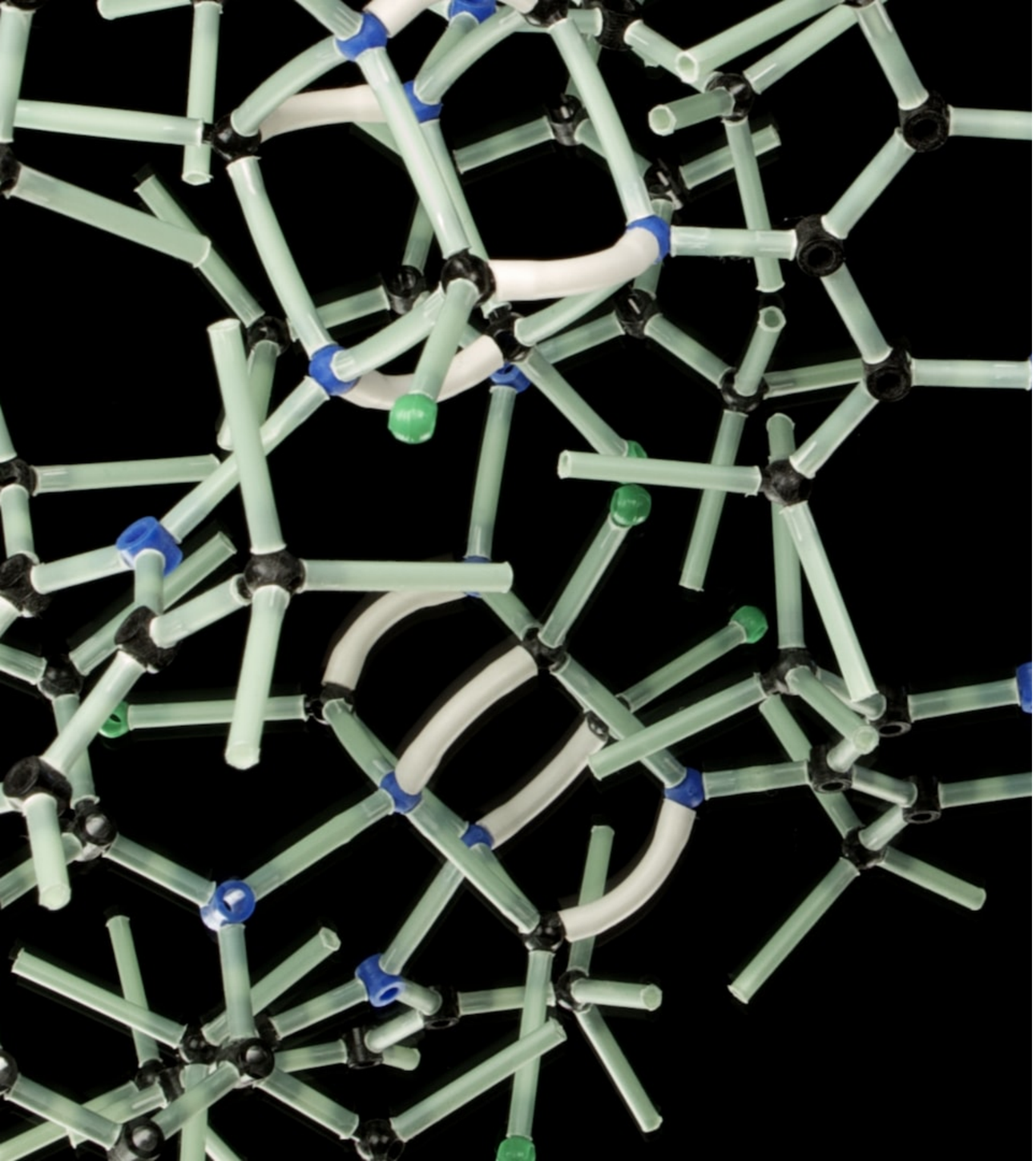


# 数据质量工程

Accessibility, Correctness, Consistency, Uniqueness, Reasonableness, Data integrity

Data Governance is used to ensure capability that enables an organization to ensure that high data quality exists throughout the complete lifecycle of the data, and data controls are implemented that support business objectives.

ⓘ wikipedia.org/wiki/Data_quality



# Data Governance

Data Quality Engineering is about:
Engineering process & tooling to build trust of data, system and team, where the critical parts are:

- Data Ownership
  - Shall we send mails to all or shout in WeChat groups?
- Metadata
  - Manage and discover data schema.
  - Maintain documentation of the schema
- Data Lineage
  - Traceablity of data, service, pipelines etc.

iSQE

# Amundsen

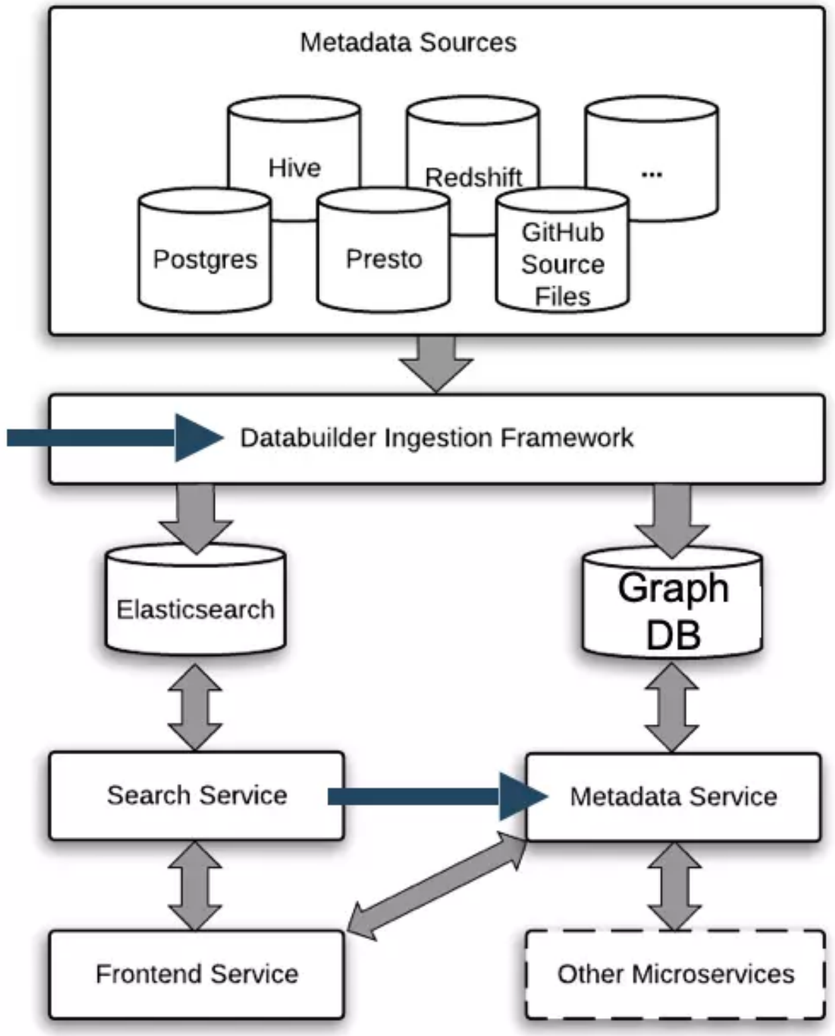
LF AI & DATA

Open source data discovery and metadata engine

github.com/amundsen-io/amundsen

iSQE

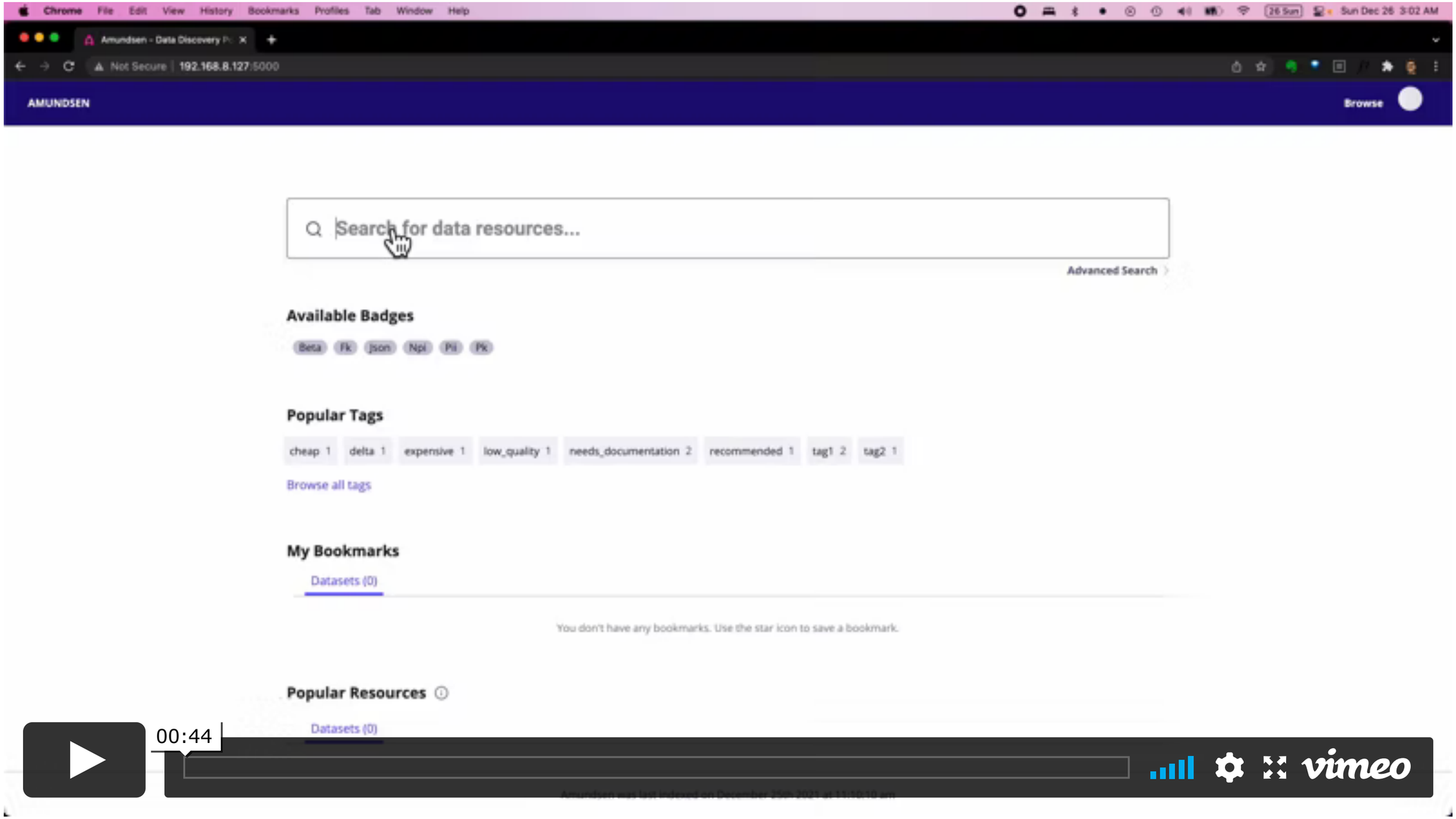
# Arch



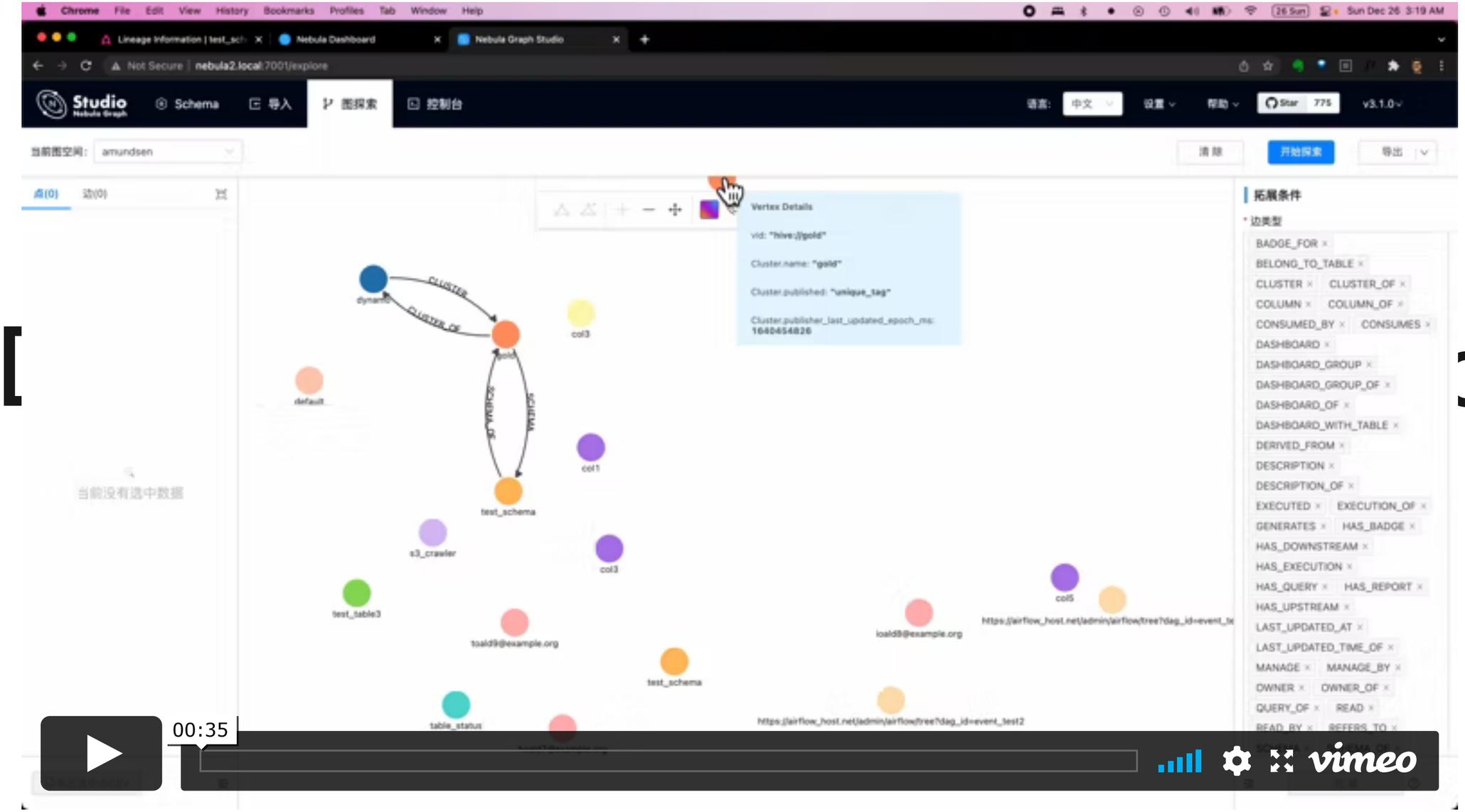

# Code

```
.
├── 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
```

amundsen-io/amundsen

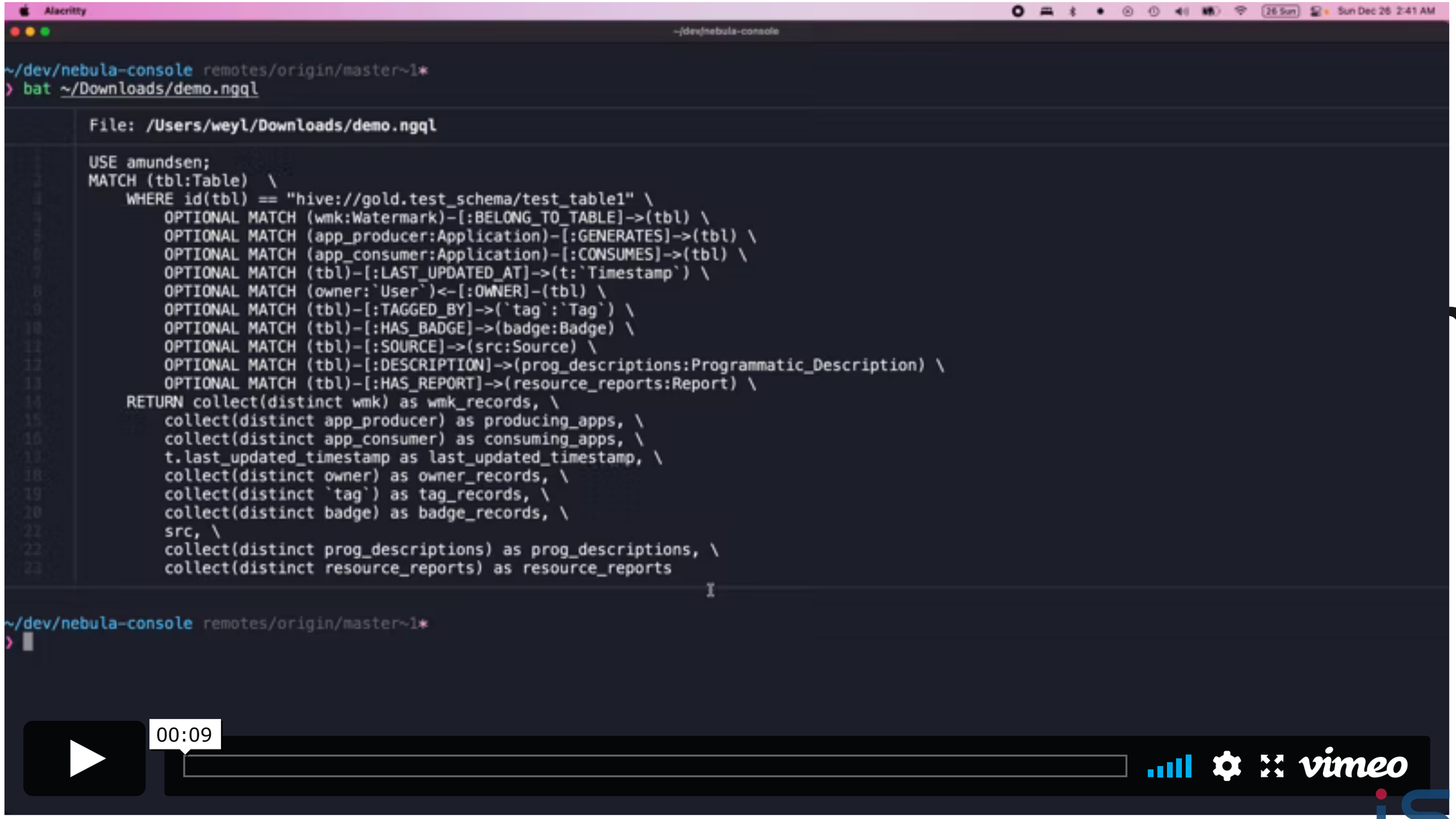

Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun Dec 26 3:02 AM
Amundsen - Data Discovery
Not Secure | 192.168.8.127:5000
AMUNDSEN
Browse
Search for data resources...
Advanced Search
Available Badges
Beta Fk json Npi Pii Pk
Popular Tags
cheap 1 delta 1 expensive 1 low_quality 1 needs_documentation 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
Datasets (0)
00:44
vimeo

iSQE


Studio
Schema
导入
图探索
控制台
amundsen
Vertex Details
vid: "hive://gold"
Cluster.name: "gold"
Cluster.published: "unique_tag"
Cluster.publisher_last_updated_epoch_ms: 1640654826
拓展条件
边类型
BADGE_FOR
BELONG_TO_TABLE
CLUSTER
CLUSTER_OF
COLUMN
COLUMN_OF
CONSUMED_BY
CONSUMES
DASHBOARD
DASHBOARD_GROUP
DASHBOARD_GROUP_OF
DASHBOARD_OF
DASHBOARD_WITH_TABLE
DERIVED_FROM
DESCRIPTION
DESCRIPTION_OF
EXECUTED
EXECUTION_OF
GENERATES
HAS_BADGE
HAS_DOWNSTREAM
HAS_EXECUTION
HAS_QUERY
HAS_REPORT
HAS_UPSTREAM
LAST_UPDATED_AT
LAST_UPDATED_TIME_OF
MANAGE
MANAGE_BY
OWNER
OWNER_OF
QUERY_OF
READ
READ_BY
REFERS_TO
当前没有选中数据
test_schema
s3_crawler
test_table3
table_status
col3
col1
col5
00:35
vimeo

[ ]

D n


Alacritty
Sun Dec 26 2:41 AM
~/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
vimeo

# NebulaGraph + Amundsen

👉🏻 github.com/amundsen-io/amundsen/issues/1816

iSQE

# Open Source

Power users and contributors kept NebulaGraph evolving to the next level in many industries.


2019 Dec.

🚀 First Cluster in Production

2020 June

🚩 1.0 GA

2022 Feb.

🎌 3.0 GA

ⓘ nebula-graph.io/cases/

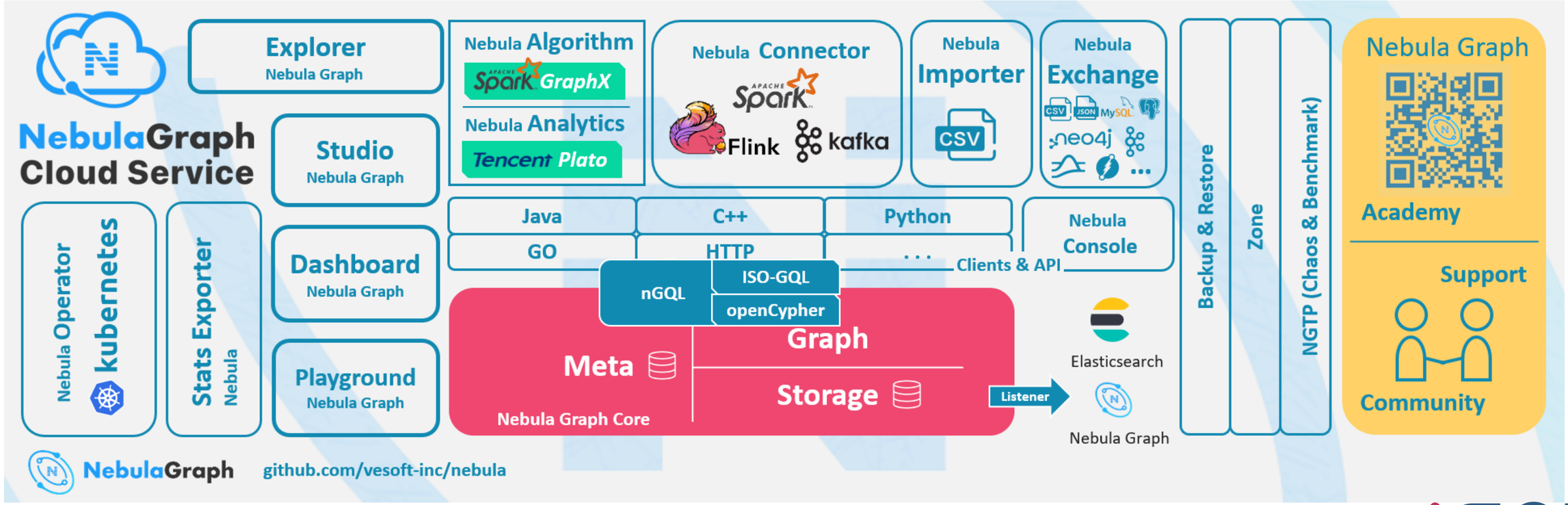
ⓘ db-engines.com/en/**ranking/graph+dbms**

iSQE

# NebulaGraph Ecosystem

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**

iSQE

# Thank YOU!

https://github.com/vesoft-inc/nebula