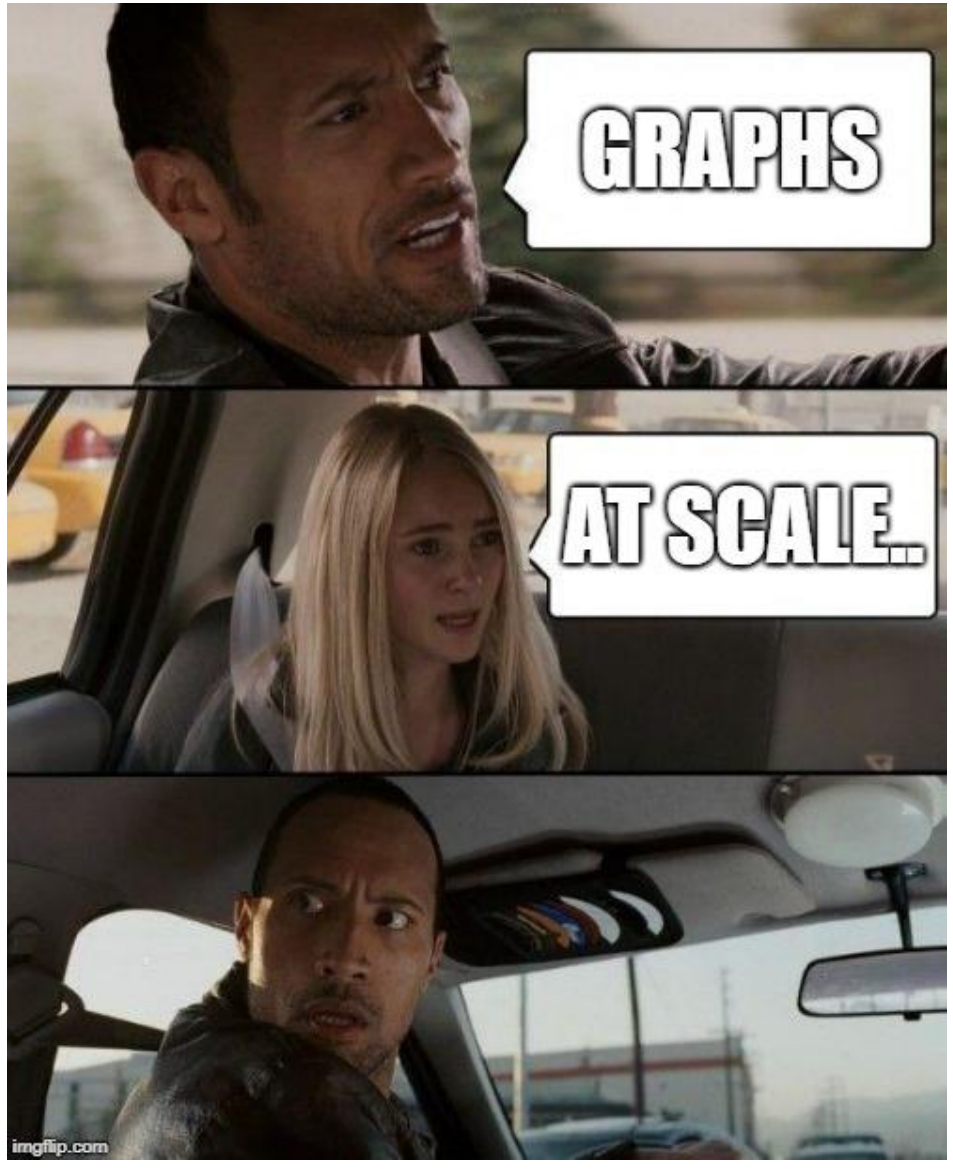
Large Scale Graph Solutions: Use-cases And Lessons Learnt

Principal Engineer, AI/Cloud Platforms

Microsoft

https://www.linkedin.com/in/rekhajoshm/

@rekhajoshm



https://www.linkedin.com/in/rekhajoshm/

@rekhajoshm

Abstraction Is The Art

Euler's Bridges - Seven Bridges of Königsberg

G = (V, E); V(id, attr1, attr2,..); E(src, dst, attr1, attr2,..)
Anonymized, attributes, direction, degrees

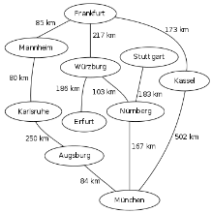
G = (V, E); V(id, attr1, attr2,..); E(src, dst, attr1, attr2,..)
Anonymized, attributes, direction, degrees

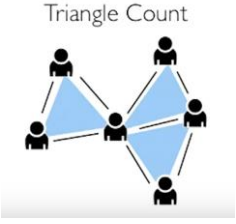
Graph Algorithms

Shortest Path (BFS)

Connected Components (DFS)

Nearness/Similarity/Ranking Algorithms





Places like berlin, with museums

Recommend connections/create communities based on life patterns

All Books read by people, who have read ‘n’ books similar to my reading list

Help me save/invest money, based on experiences of ‘xyz’ who faced similar financial situation like mine

https://www.linkedin.com/in/rekhajoshm/

@rekhajoshm

Graphs == Intelligence Of The System

Knowledge Graph

Recommendation/Personalization Engines

Fraud Detection/Risk Assessments

Advices to Users
(travel, financial, healthcare, shopping advice)

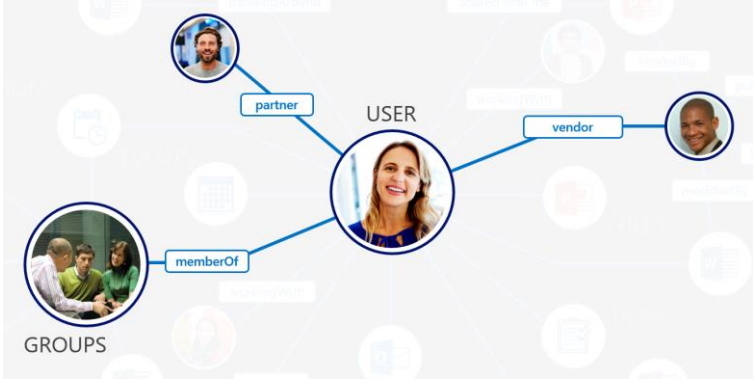
Insights

Graph <-> Data Science/AI

https://www.linkedin.com/in/rekhajoshm/

@rekhajoshm

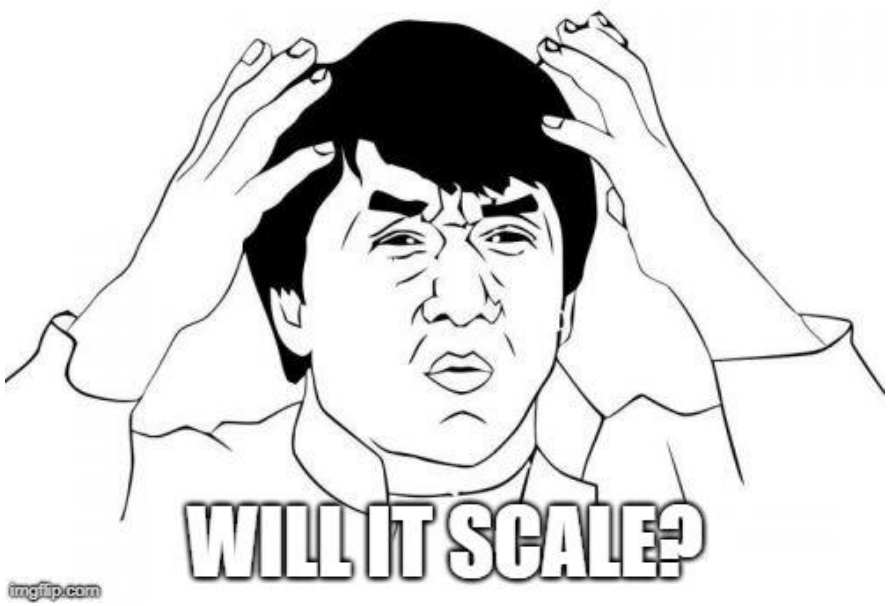
Graph <-> AI



Explicit knowledge in data not per model specific
Create graphs with inferred and applied intelligence
Create relationships based on claimed, verified, observed, derived data
Semantically unified and connected data view
Narrow relevant subgraph for AI and computational purposes

https://www.linkedin.com/in/rekhajoshm/

@rekhajoshm



Traffic

- Continuous Data Flow/Streams
- Millions of Customers
- Billions of Relationships

Data

- Incomplete/Incorrect Data(Inference)
- Raw/Aggregate Data
- Time Relevance

System Capability

- Storage, IO, Data Transfer, In-memory
- Backups/Retention
- Automation

Ease Of Use

API endpoints for CRUD operations.

Query/Update/Remove by set of attributes

Bulk api support for ease of usage

Security

Authorized API

Encryption/Decryption at attribute level

Key rotation baked in

Governance - Classification, metadata, access control, lineage

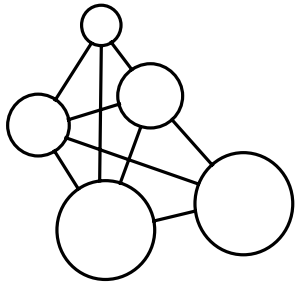
Monitoring

Support 99.99 Availability

Latency of API calls in 30-100 ms

Support DR with RTO: 10 mins, RPO: 60 mins

Support current, expected TPS



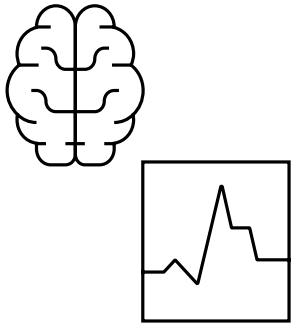
Graph DB (OLTP)

Neo4J

Marklogic DB

DSE Graph/Titan

Dgraph



Graph Pipelines (OLAP)

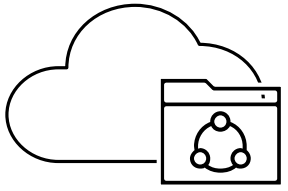
Cassandra + Solr

Cassandra/Neo4J + Spark Connector

Spark GraphX (RDD)

Spark GraphFrames (DataFrames)

Flink Graph/Gelly (Datasets)



Graph Cloud

Microsoft Azure Cosmos Graph DB

Other Cloud Providers

G = (V, E); V(id, attr1, attr2,..); E(src, dst, attr1, attr2,..)

Anonymized, attributes, direction, degrees

Graph Databases

Reduce Computational Cost

Data Stored Linked Together, Natural, Simple, efficient retrievals

RDBMS -> Graph/NoSQL/Ontology

Multi-model, ACID, Medium to Large Data

Expressive Graph Queries(Gremlin, SPARQL, Cypher, API)

Pattern Matching

Find Core Vertices

Group on Attribute, filter/compute with condition, Subgraph,

Expressive Graph Query Languages

Cypher
MATCH (n:Person) RETURN n.name

MATCH (d:Database)-[:USES]->(**Cypher**)-[:QUERIES]->(:Model:Graph)
WHERE d.name IN [‘DATABASE1',...] RETURN Cypher.features

Gremlin
gremlin> g.V.has('name','hercules').name **==>**hercules
gremlin> g.V.has('name','hercules').out('father').name **==>**jupiter
gremlin> g.V.has('name','hercules').out('father').out('father').name
==>saturn

Query Languages

Gremlin
$G = (V, E)$

W3C SPARQL
RDF Triples

Cypher
openCypher
Cypher For Spark
GraphQL

Graph DB (OLTP)

Neo4J

https://neo4j.com/

Marklogic DB

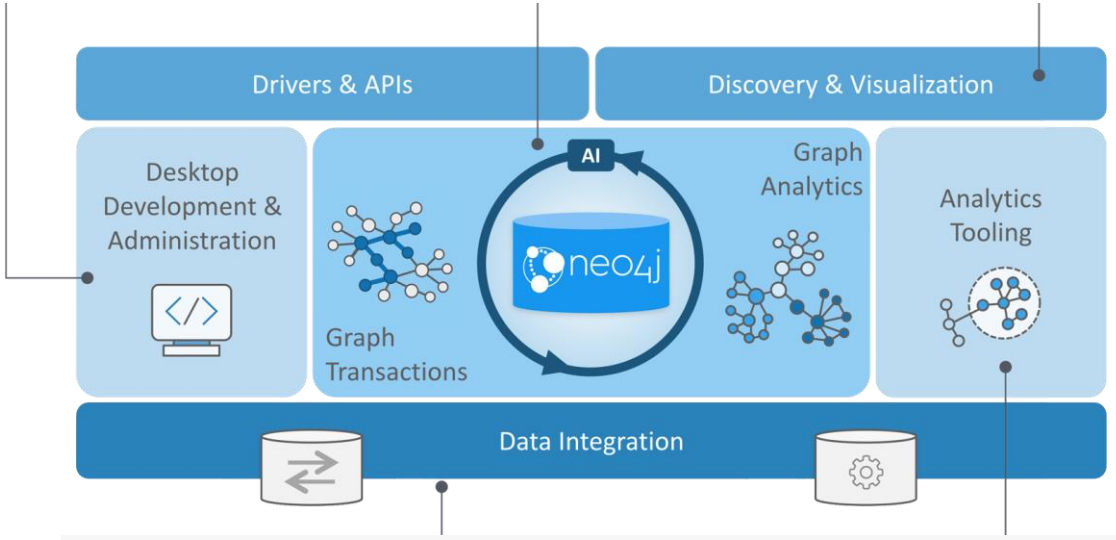
https://www.marklogic.com/

DSE Graph/Titan

http://titan.thinkaurelius.com/

Dgraph

https://dgraph.io/



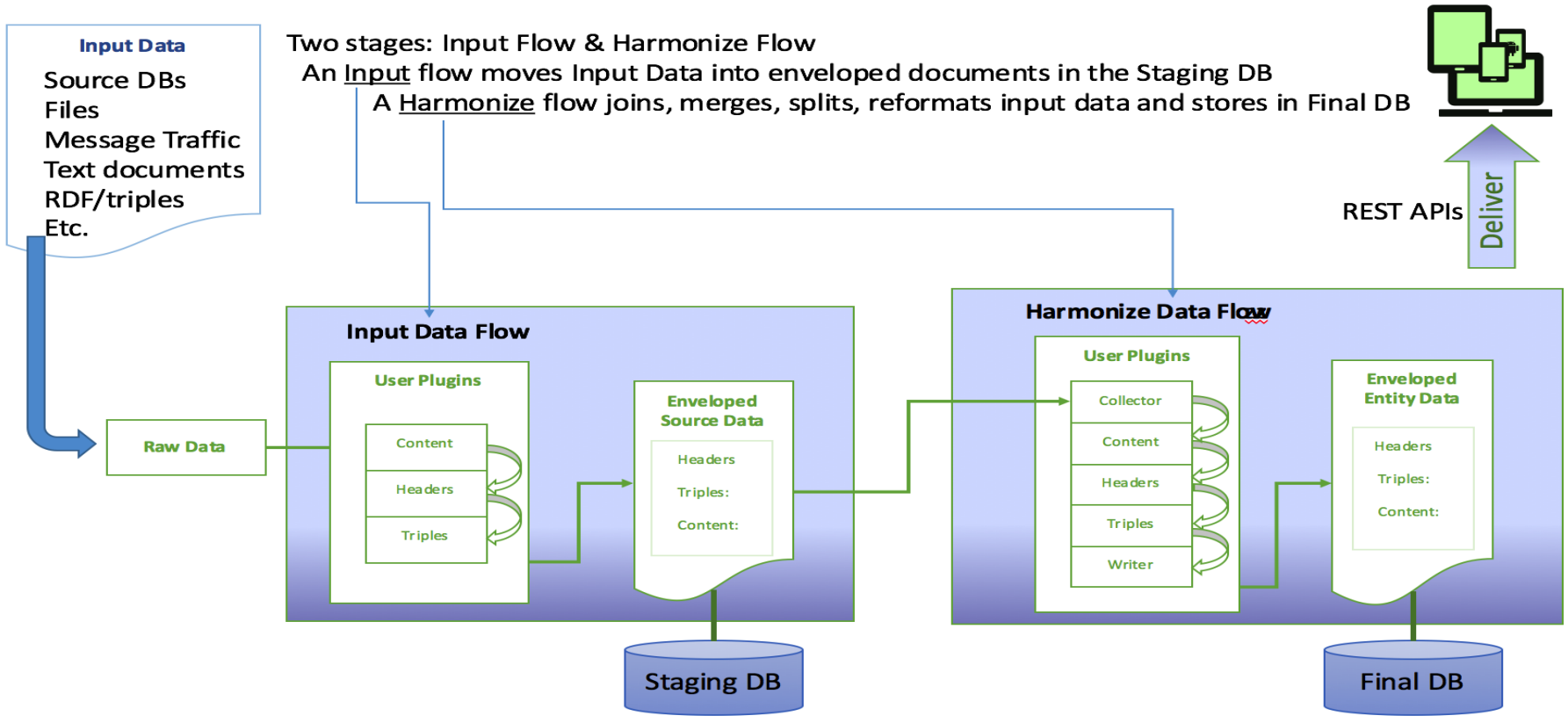
https://www.zdnet.com/article/neo4j-and-nasa-where-graph-databases-technology-really-is-rocket-science

https://www.linkedin.com/in/rekhajoshm/

@rekhajoshm

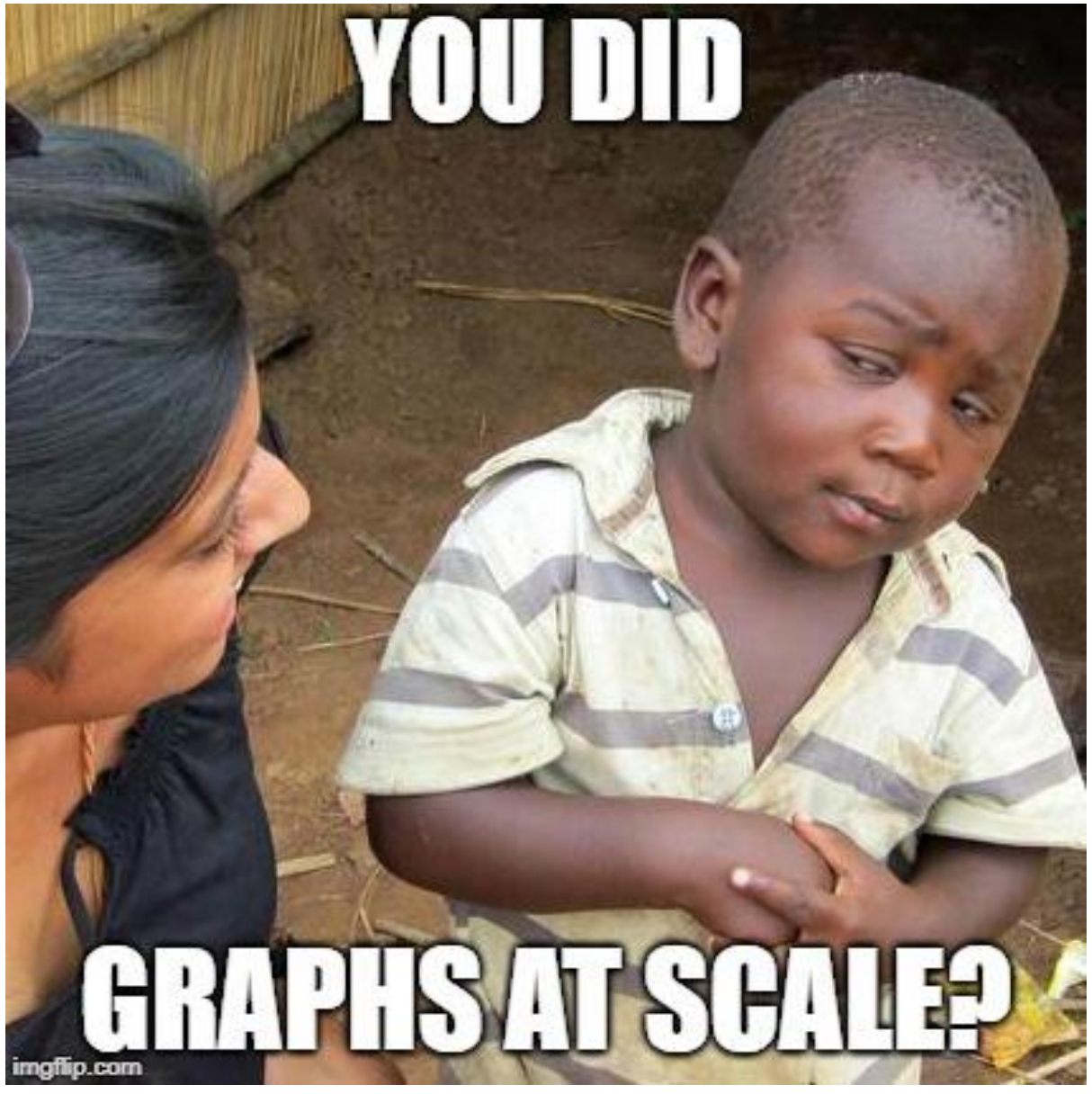
Marklogic 9 Data Hub, Harmonization
Create graph based on ontological inference

https://github.com/marklogic/marklogic-data-hub



https://www.linkedin.com/in/rekhajoshm/

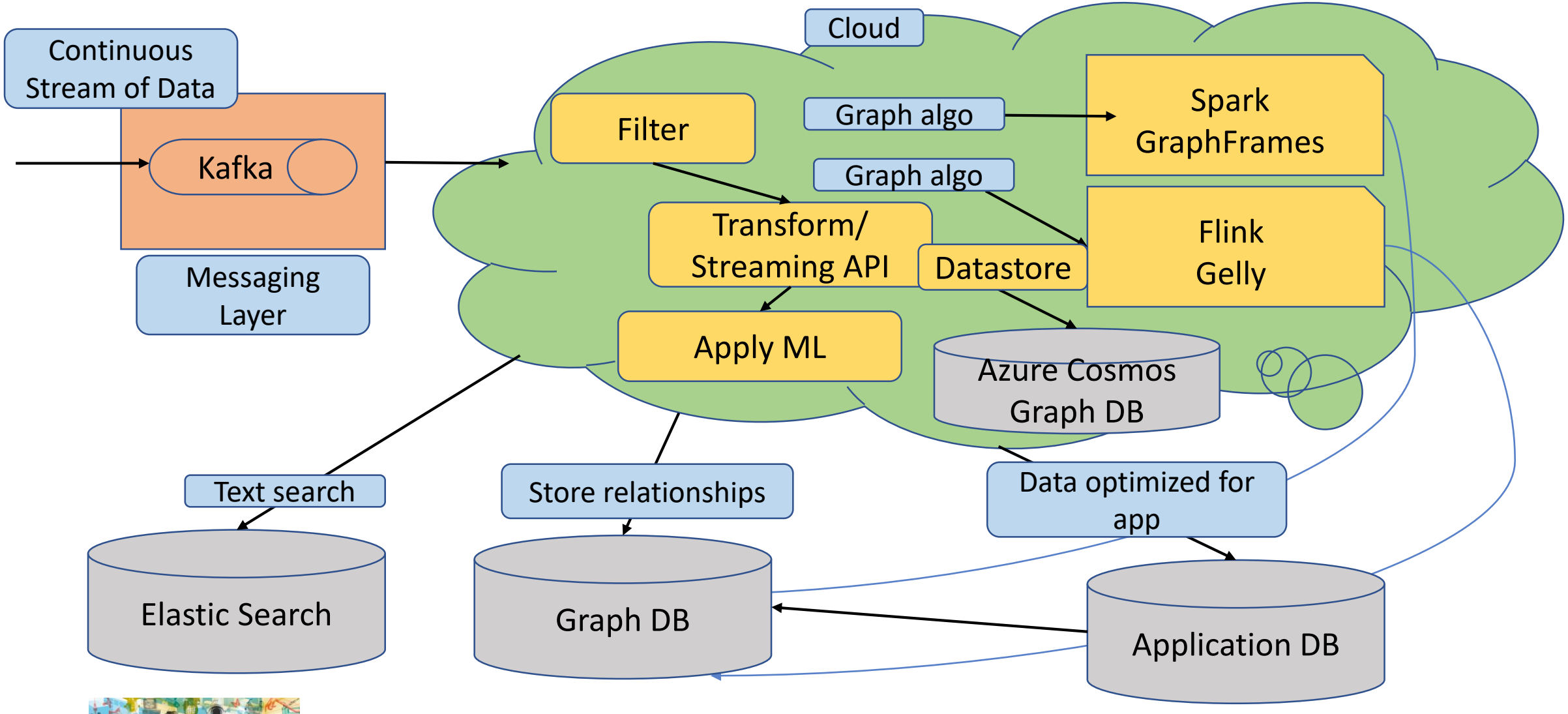
@rekhajoshm



https://www.linkedin.com/in/rekhajoshm/

@rekhajoshm

Graph Pipelines (OLAP)

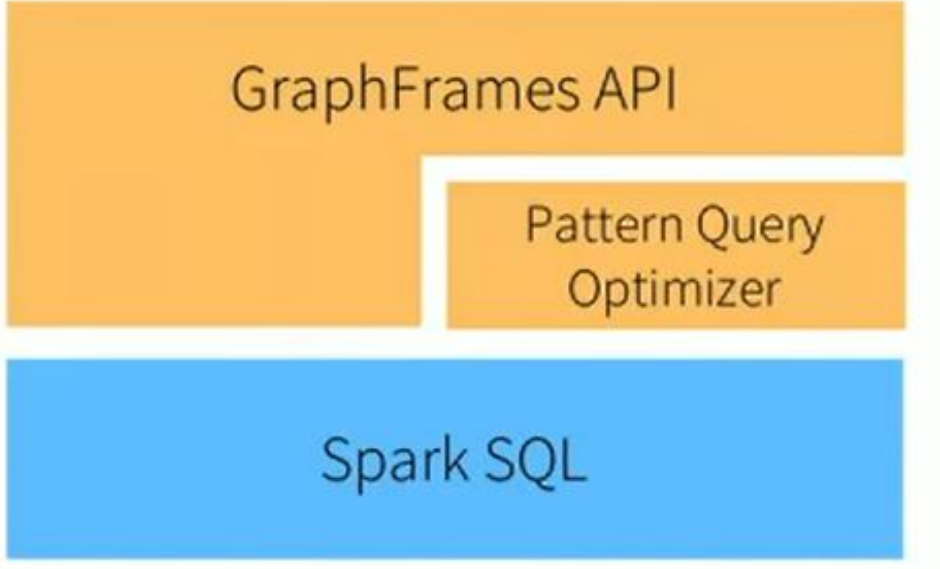


https://www.linkedin.com/in/rekhajoshm/

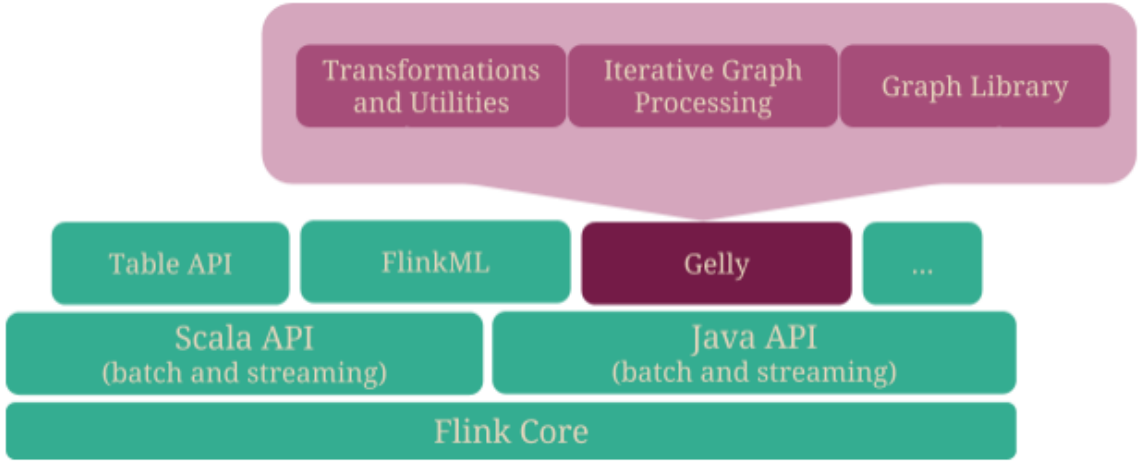
@rekhajoshm

Pregel
Apache Giraph
Apache Spark GraphX / GraphFrames
Apache Flink Graph API/Gelly
ElasticSearch Graph API (Lucene Indexes)

Spark
GraphFrames



Flink
Gelly



Graph Cloud

Modern apps face new challenges

Managing and syncing data distributed around the globe

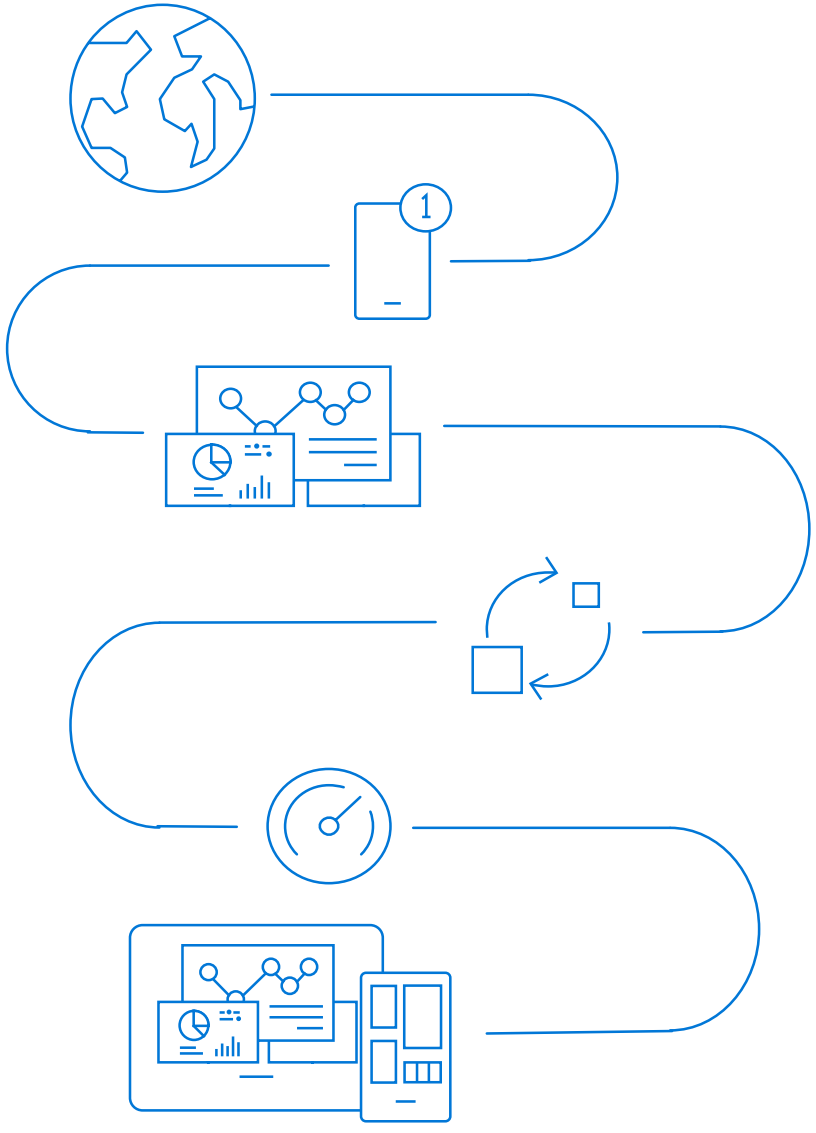
Delivering highly-responsive, real-time personalization

Processing and analyzing large, complex data

Scaling both throughput and storage based on global demand

Offering low-latency to global users

Modernizing existing apps and data



https://www.linkedin.com/in/rekhajoshm/

@rekhajoshm

86%
of Fortune 500 on Microsoft Cloud

Azure Cosmos db

A FULLY-MANAGED GLOBALLY DISTRIBUTED DATABASE SERVICE BUILT TO GUARANTEE EXTREMELY LOW LATENCY AND MASSIVE SCALE FOR MODERN APPS

What is Azure Cosmos DB

A globally distributed, massively scalable, multi-model database service

SQL

cassandra

MongoDB

Table API

Gremlin

JS

Key-value

Column-family

Document

Graph

Guaranteed low latency
at the 99th percentile

Elastic scale out
of storage & throughput

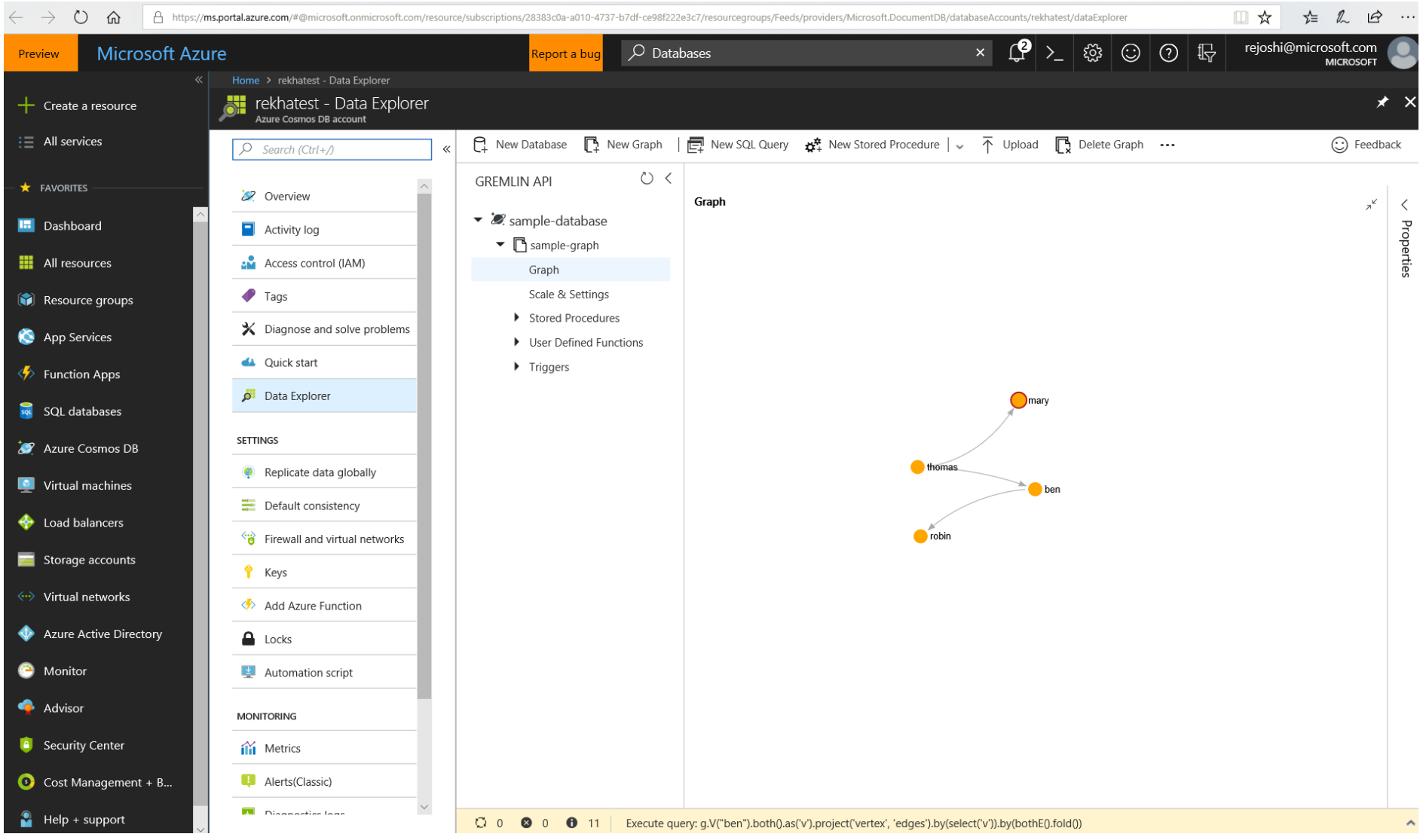
Five well-defined
consistency models

Turnkey global
distribution

Comprehensive
SLAs

https://www.linke

@rekhajoshm



https://www.linkedin.com/in/rekhajoshm/

@rekhajoshm

```
Now, let's add some edges between the vertices. Press any key to continue...
       Running this Gremlin query:
       g.V('thomas').addE('knows').to(g.V('mary'))

       Inserted this edge:
       [{'id': 'fb0e8313-5cf6-44f8-8ad0-14c0917a85e0', 'label': 'knows', 'type': 'edge', 'inVLabel': 'person', 'o
el': 'person', 'inV': 'mary', 'outV': 'thomas'}]

       Running this Gremlin query:
       g.V('thomas').addE('knows').to(g.V('ben'))

       Inserted this edge:
       [{'id': '4e5c94cd-5597-42d8-abdf-8892e76b90ed', 'label': 'knows', 'type': 'edge', 'inVLabel': 'person', 'o
el': 'person', 'inV': 'ben', 'outV': 'thomas'}]

       Running this Gremlin query:
       g.V('ben').addE('knows').to(g.V('robin'))

       Inserted this edge:
       [{'id': '2ca50275-77e7-4567-ba09-09add43e3fa2', 'label': 'knows', 'type': 'edge', 'inVLabel': 'person', 'o
el': 'person', 'inV': 'robin', 'outV': 'ben'}]
```

```
Now, let's add some edges between the vertices. Press any key to continue...
       Running this Gremlin query:
       g.V('thomas').addE('knows').to(g.V('mary'))

       Inserted this edge:
       [{'id': 'fb0e8313-5cf6-44f8-8ad0-14c0917a85e0', 'label': 'knows', 'type': 'edge', 'inVLabel': 'person', 'outVLa
el': 'person', 'inV': 'mary', 'outV': 'thomas'}]

       Running this Gremlin query:
       g.V('thomas').addE('knows').to(g.V('ben'))

       Inserted this edge:
       [{'id': '4e5c94cd-5597-42d8-abdf-8892e76b90ed', 'label': 'knows', 'type': 'edge', 'inVLabel': 'person', 'outVLa
el': 'person', 'inV': 'ben', 'outV': 'thomas'}]

       Running this Gremlin query:
       g.V('ben').addE('knows').to(g.V('robin'))

       Inserted this edge:
       [{'id': '2ca50275-77e7-4567-ba09-09add43e3fa2', 'label': 'knows', 'type': 'edge', 'inVLabel': 'person', 'outVLa
el': 'person', 'inV': 'robin', 'outV': 'ben'}]
```

```
      [{'id': 'fb0e8313-5cf6-44f8-8ad0-14c0917a85e0', 'label': 'knows', 'type': 'edge', 'inVLabel': 'person', 'outVLab
': 'person', 'inV': 'mary', 'outV': 'thomas'}]

      Running this Gremlin query:
      g.V('thomas').addE('knows').to(g.V('ben'))

      Inserted this edge:
      [{'id': '4e5c94cd-5597-42d8-abdf-8892e76b90ed', 'label': 'knows', 'type': 'edge', 'inVLabel': 'person', 'outVLab
l': 'person', 'inV': 'ben', 'outV': 'thomas'}]

      Running this Gremlin query:
      g.V('ben').addE('knows').to(g.V('robin'))

      Inserted this edge:
      [{'id': '2ca50275-77e7-4567-ba09-09add43e3fa2', 'label': 'knows', 'type': 'edge', 'inVLabel': 'person', 'outVLab
l': 'person', 'inV': 'robin', 'outV': 'ben'}]

ı, sorry. I made a mistake. Let's change the ages of these two vertices. Press any key to continue...
      Running this Gremlin query:
      g.V('thomas').property('age', 44)

      Updated this vertex:
      [{'id': 'thomas', 'label': 'person', 'type': 'vertex', 'properties': {'firstName': [{'id': '42169e5b-fbee-45d7-b
}0-fbb511adf36b', 'value': 'Thomas'}], 'age': [{'id': 'fac71f12-af04-47a0-944f-4612ade35f01', 'value': 44}]}}]
```

https://www.linkedin.com/in/rekhajoshm/

@rekhajoshm

Lessons Learnt

Ease Of Use(customer)

Data features(granularity, skewness)

Sampling, Subgraphs

Updating Graph/CRUD

Graph Visualization

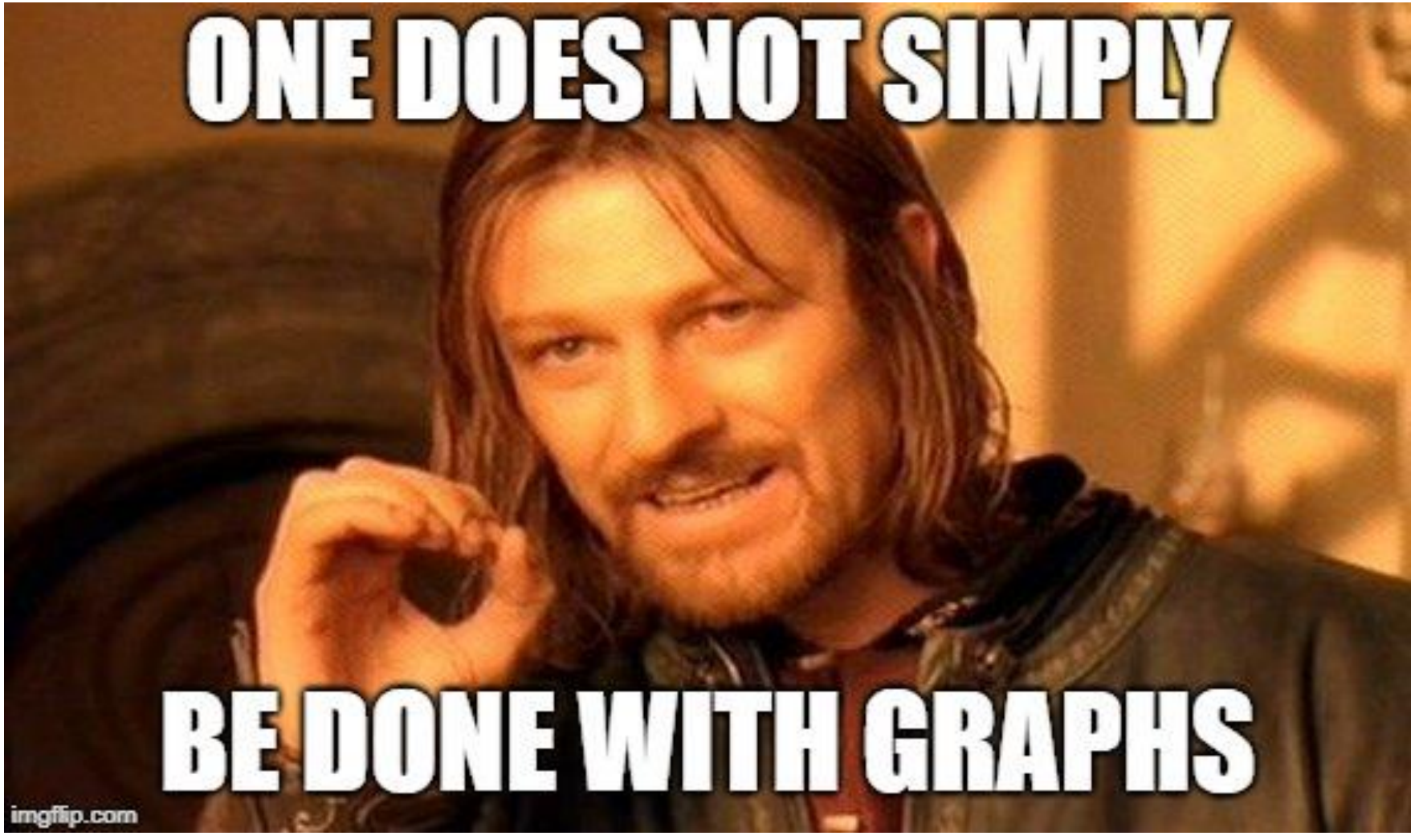
Performance/Efficiency/Caching

REST API Interfaces

Let 1000 flowers bloom, but then..

Miminize Adhoc ETL/data analysis

Sync of Analytics and End user application



https://www.linkedin.com/in/rekhajoshm/

@rekhajoshm