Unleashing the Power of Graphs in Java Code Structure Analysis

Hello 👋

I am Johannes Troppacher

Senior Software Engineer

@Dynatrace

https://github.com/**JohT**

Hello 👋

I am Johannes Troppacher

Passionate about…

- Digital Signal Processing
- Event-driven Architecture
- Graph Data Science

https://github.com/**JohT**

Hello 👋

I am Johannes Troppacher

Contributed to…

- Quarkus
- AxonFramework
- PyTorch

https://github.com/JohT

1.

Story

Once upon a time…

1 — 2 — 3 — 4 — 5 — 6

Story

Introduction to Graphs

Overview of Disciplines

Graph Data Analysis

Graph Data Science

Tools

Let’s make a ***breaking change*** *and update* ***the major version of*** *a commonly used shared library.*

How hard could this be?

> *Next we will update the modules that are using the shared library.*

😕

“

Ok, we would also need to update the ***modules that are using the modules …*** *that are using the shared library.*

☹

Now we’ve got a ***conflict****.. In which order should we actually do the updates?*

Reference:

2.

Introduction to Graphs

Graph Elements

1 — 2 — 3 — 4 — 5 — 6

Story

Introduction to Graphs

Overview of Disciplines

Graph Data Analysis

Graph Data Science

Tools

Introduction

Not a Graph at least in this talk

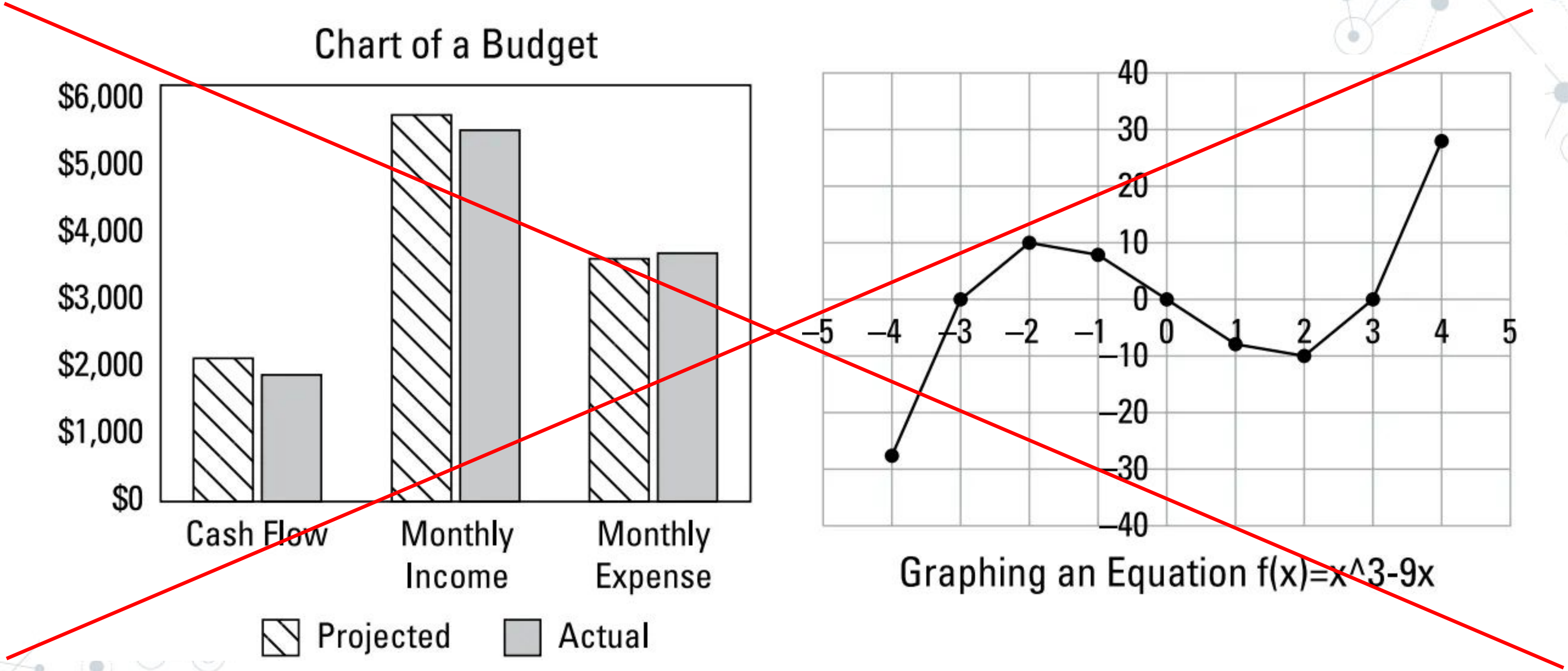
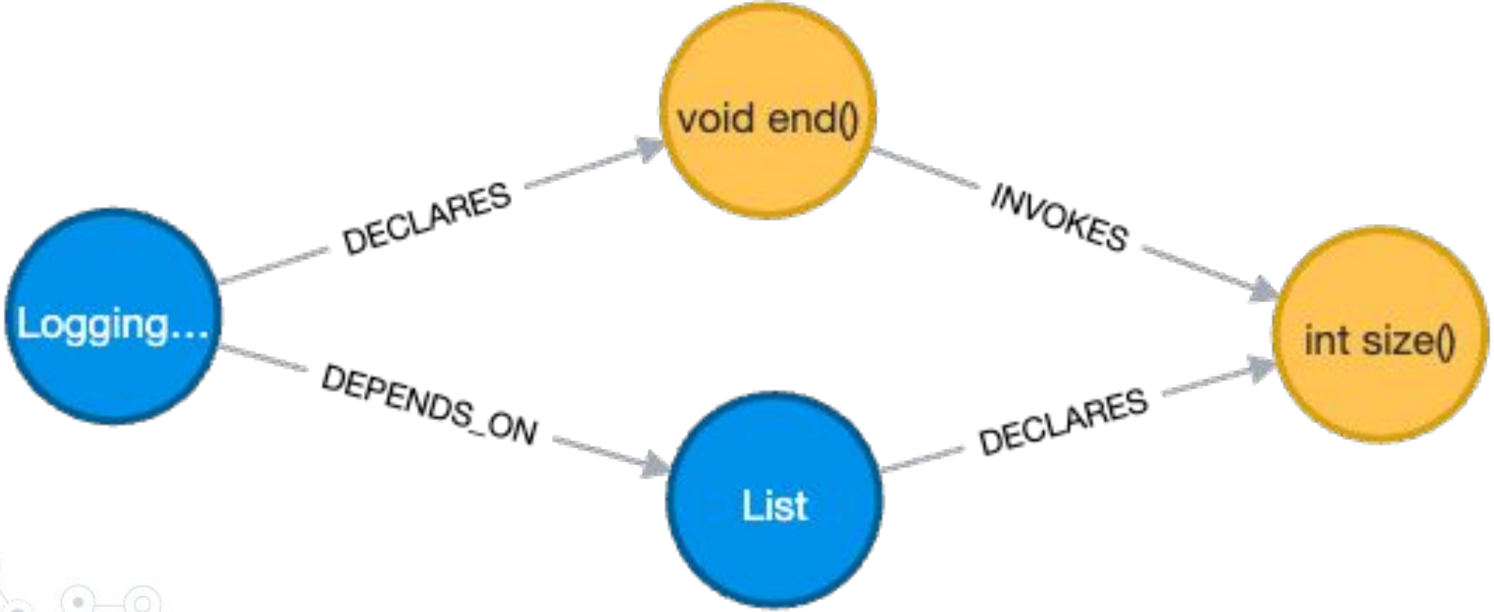


Chart of a Budget

Graphing an Equation f(x)=x^3-9x

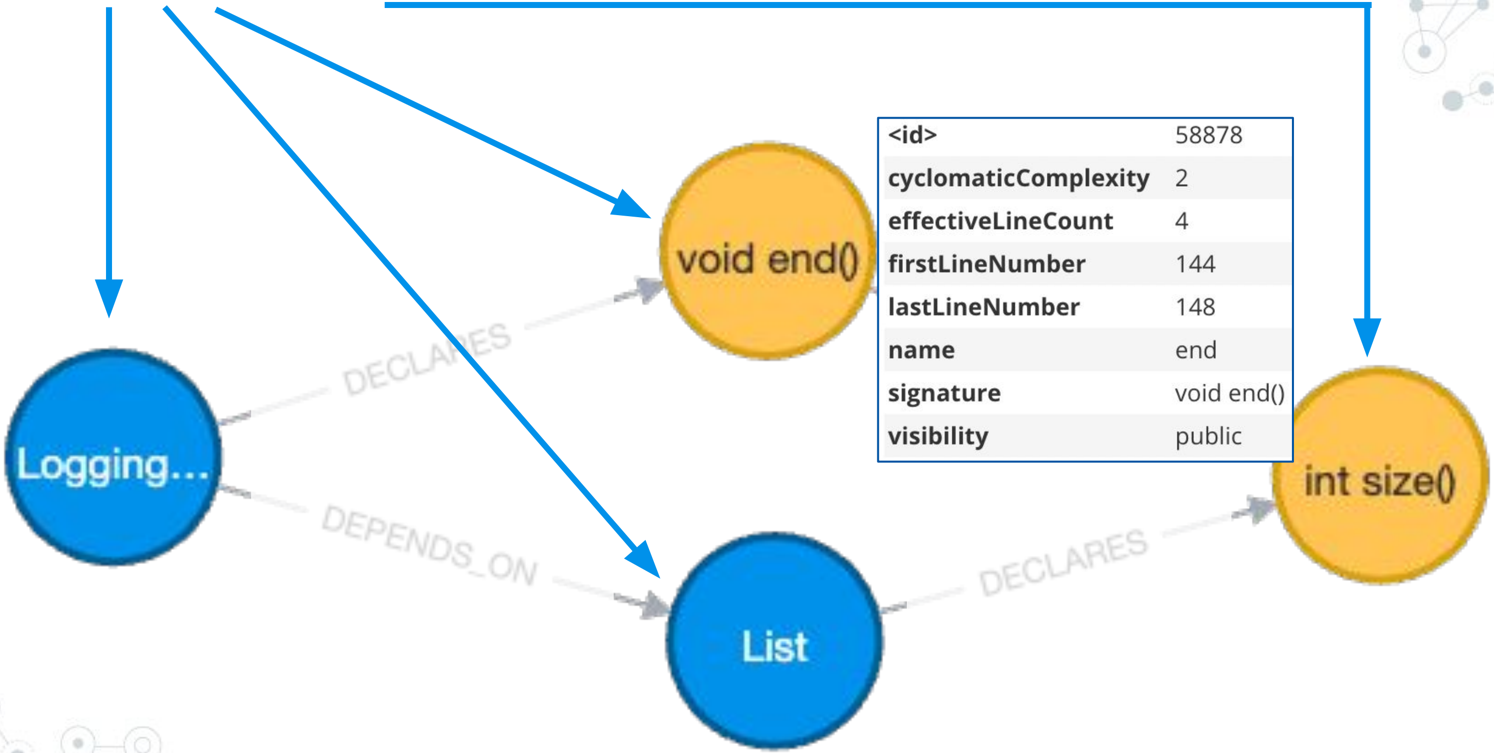
Reference: https://techfirst.medium.com/graph-data-science-101-understanding-graphs-and-graph-data-science-c25055a9db01

This is a Graph



Graph Elements

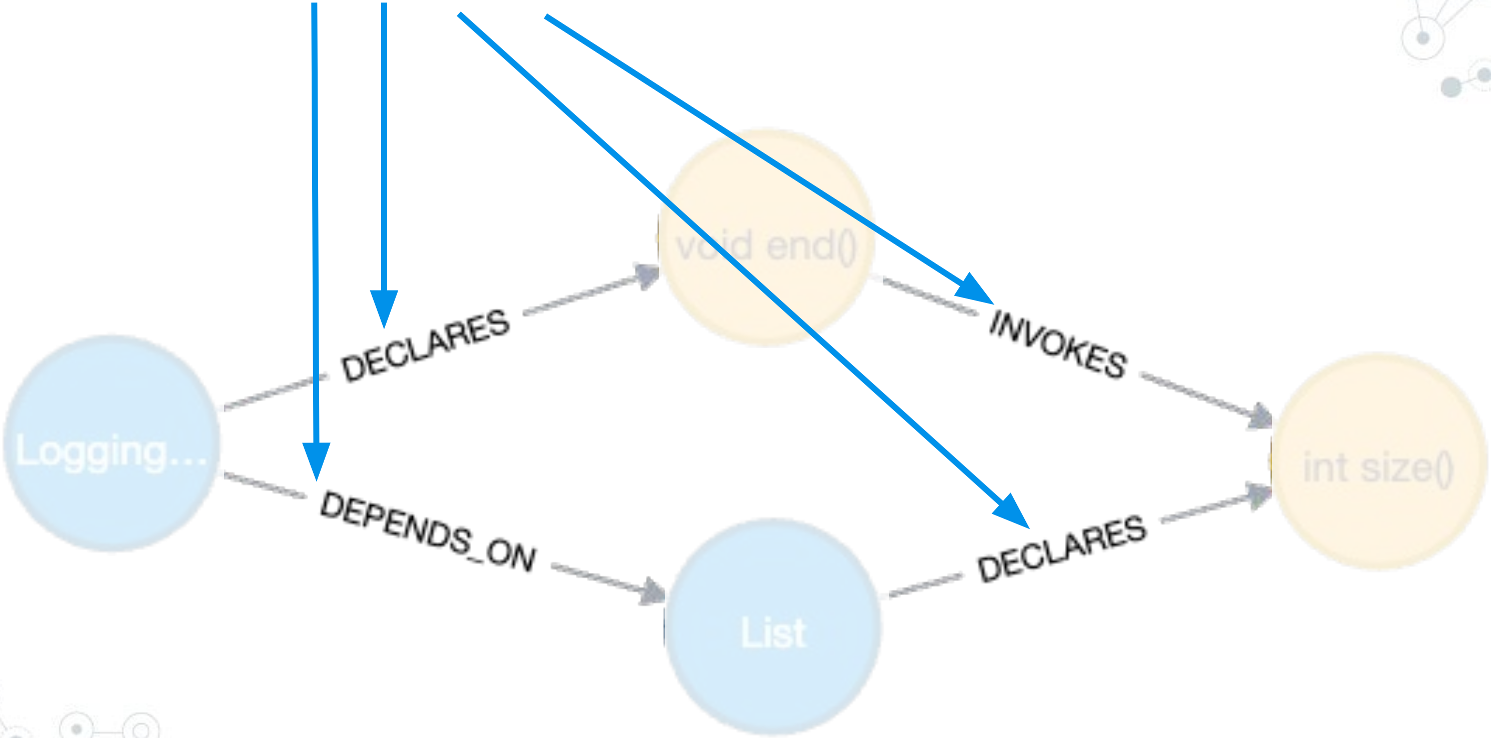
Nodes (=Vertices, ≈Entities)



<id>	58878
cyclomaticComplexity	2
effectiveLineCount	4
firstLineNumber	144
lastLineNumber	148
name	end
signature	void end()
visibility	public

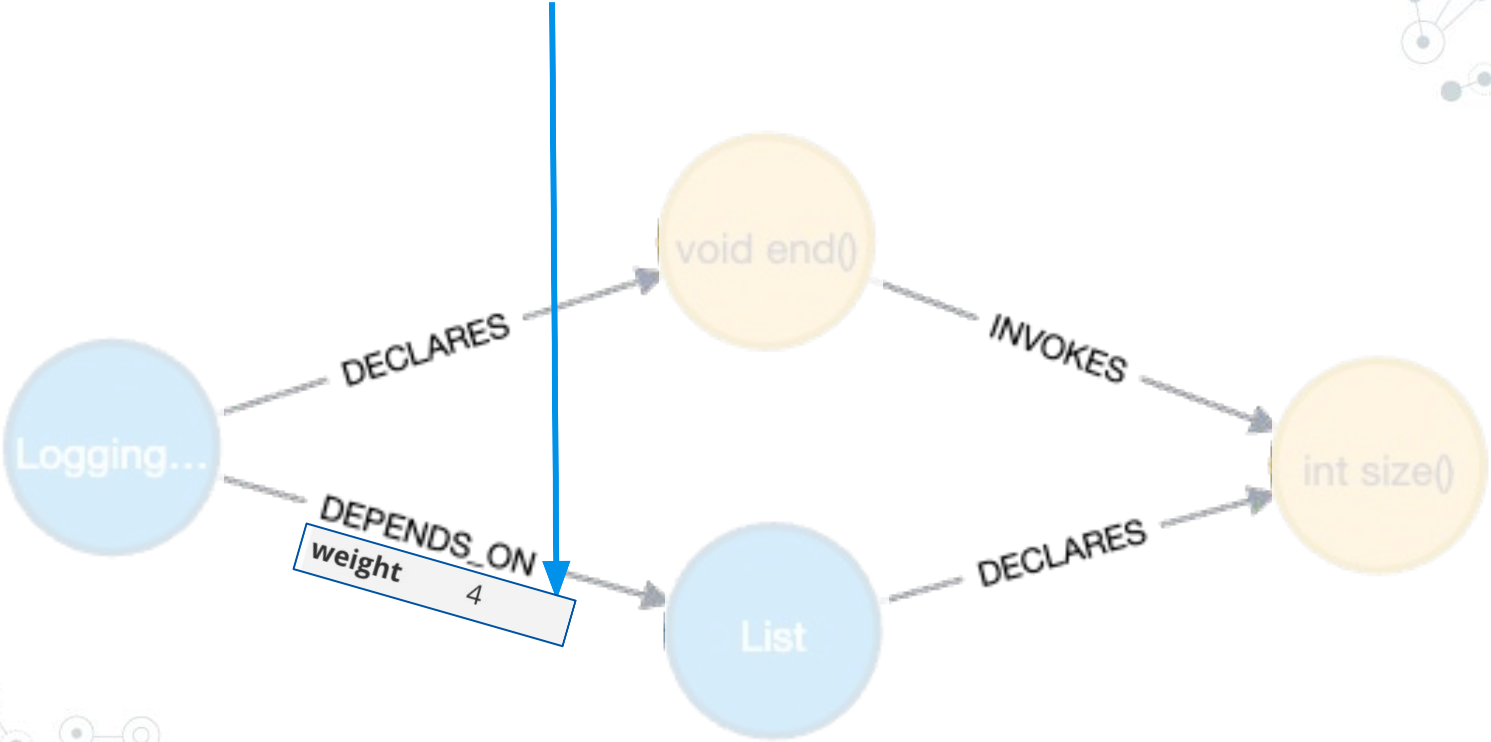
Graph Elements

Relationships (=Edges)



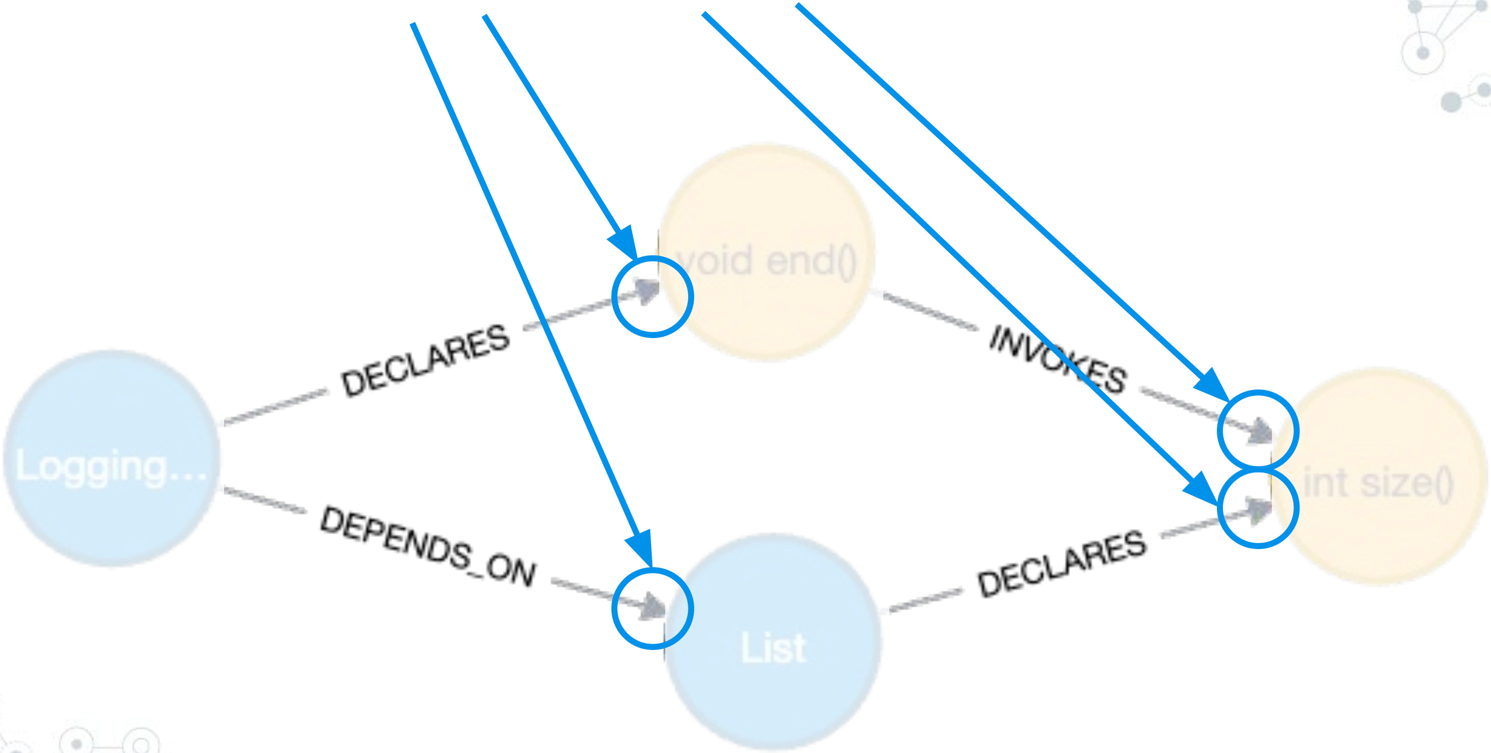
Graph Elements

Weights (weighted vs. unweighted)



Graph Elements

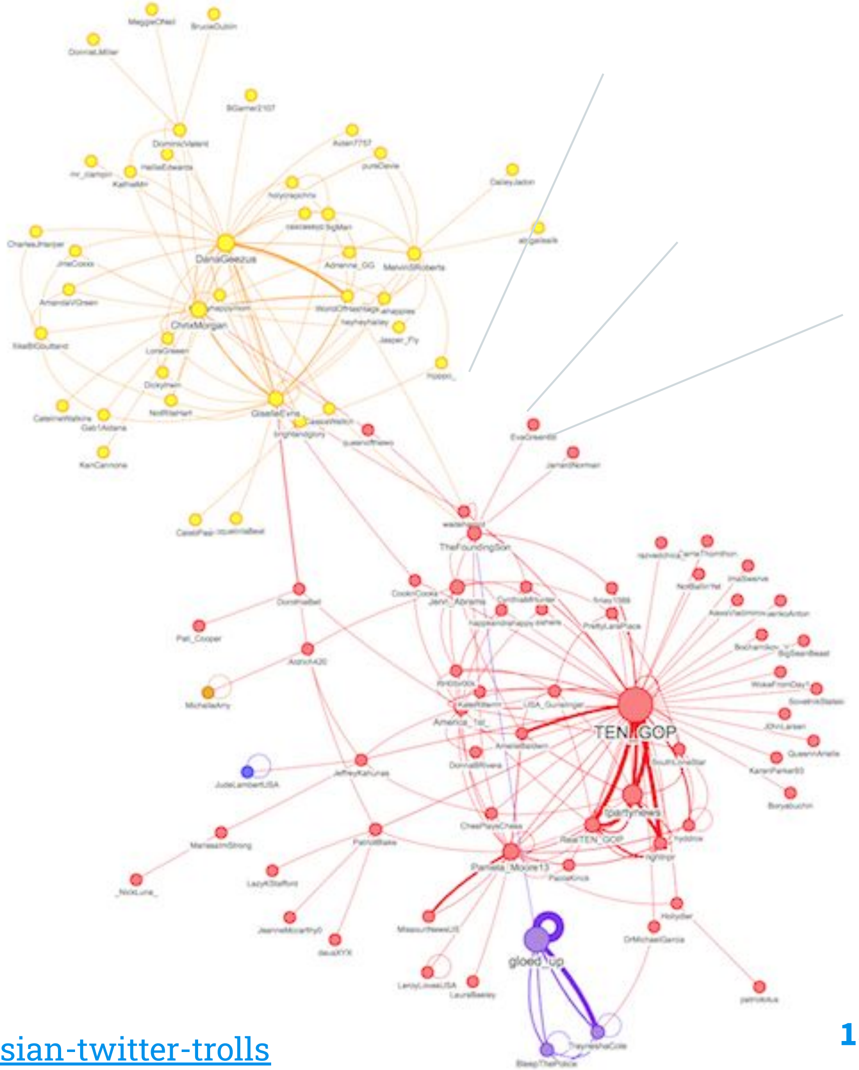
Direction (directed vs. undirected)



Graph Example

Russian Twitter (X) Trolls

“A community detection algorithm shows there are **three clear communities** in the Russian troll retweet network. **Node size** is proportional to the PageRank score for each node, **showing the importance** of the account in the network.”



Reference: https://neo4j.com/blog/story-behind-russian-twitter-trolls

3.

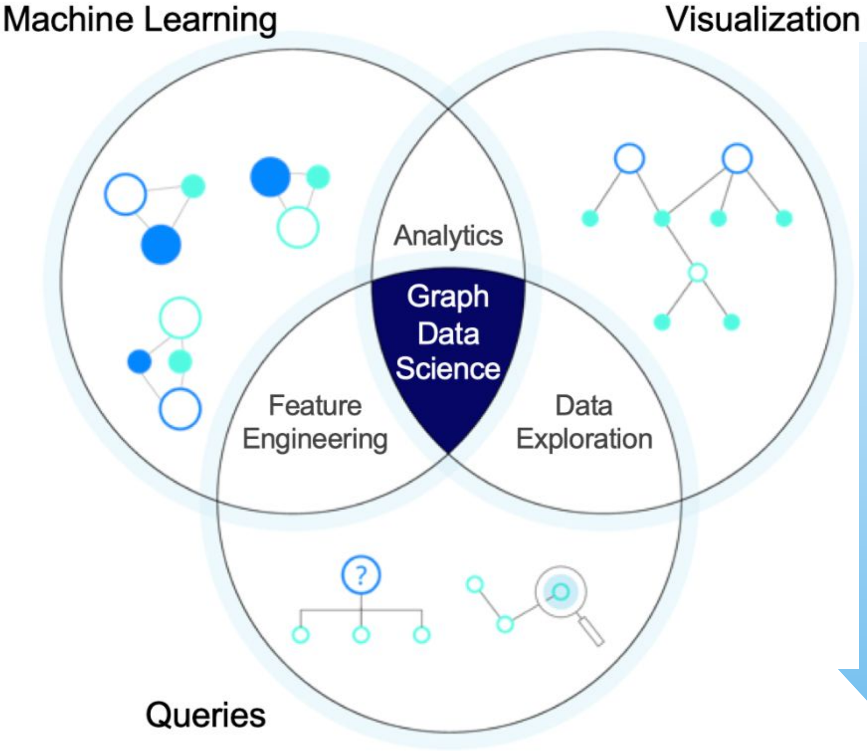
Disciplines

Analysis ➡ Algorithms ➡ Feature Engineering

1 — 2 — 3 — 4 — 5 — 6

Story — Introduction to Graphs — Overview of Disciplines — Graph Data Analysis — Graph Data Science — Tools

Overview of Disciplines



Neo4j, Inc. All rights reserved 2021

- Query
- Analyze & Visualize
- Algorithms
- Node Embeddings
- Feature Engineering
- Machine Learning

Reference: https://de.slideshare.net/neo4j/graphs-for-data-science-and-machine-learning

4.

Graph Data Analysis

Dependencies & Metrics

1 — 2 — 3 — 4 — 5 — 6

Story

Introduction to Graphs

Overview of Disciplines

Graph Data Analysis

Graph Data Science

Tools

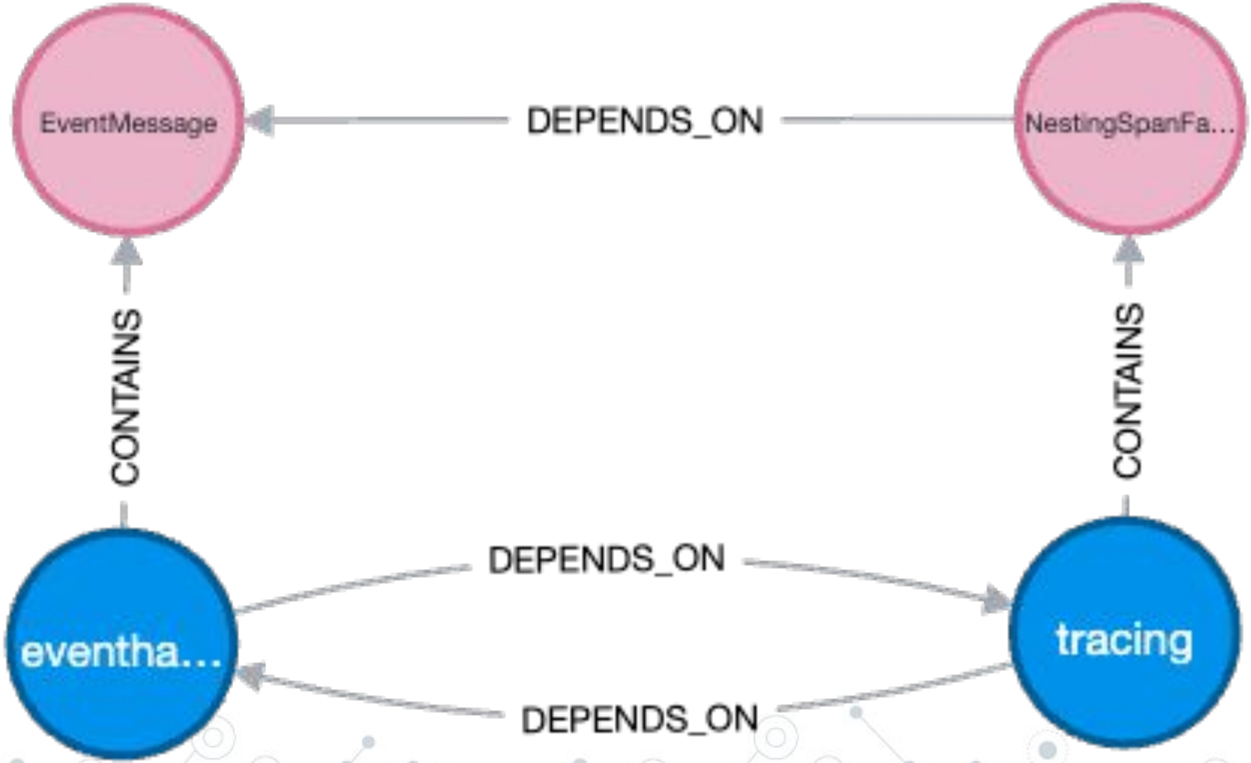
Query
Cyclic Dependencies

Artifact Name	Package Name	Dependent Packagen Name	Forward->Backward Balance	Number Forward	Number Backward
axon-messaging	eventhandling	tracing	0.9	19	1
axon-messaging	queryhandling	messaging.responsetypes	0.8823529411764706	16	1
axon-messaging	queryhandling	tracing	0.8571428571428571	15	1
axon-messaging	eventhandling	messaging	0.8536585365853658	39	3
axon-messaging	eventhandling	messaging.annotation	0.84	23	2
axon-messaging	deadline	tracing	0.8	9	1
axon-messaging	commandhandling	tracing	0.7777777777777778	8	1

Reference: https://github.com/JohT/code-graph-analysis-pipeline/blob/main/results/AxonFramework-4.8.0/internal-dependencies-csv/Cyclic_Dependencies.csv

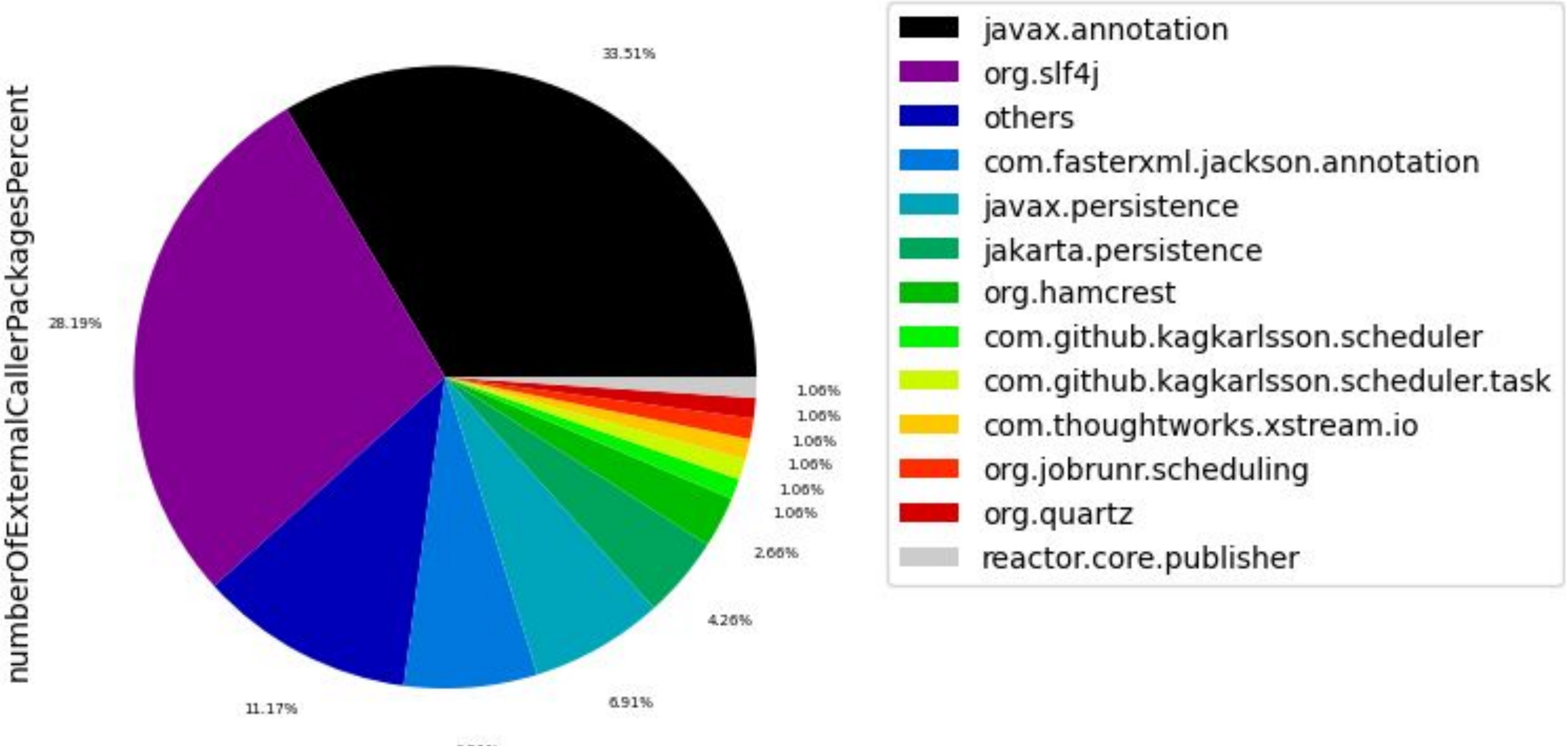
Analyze

Cyclic Dependencies



Visualize
External Dependencies Usage

Top external package usage [%] by package



Reference: https://github.com/JohT/code-graph-analysis-pipeline/blob/main/results/AxonFramework-4.8.0/external-dependencies/ExternalDependencies.md

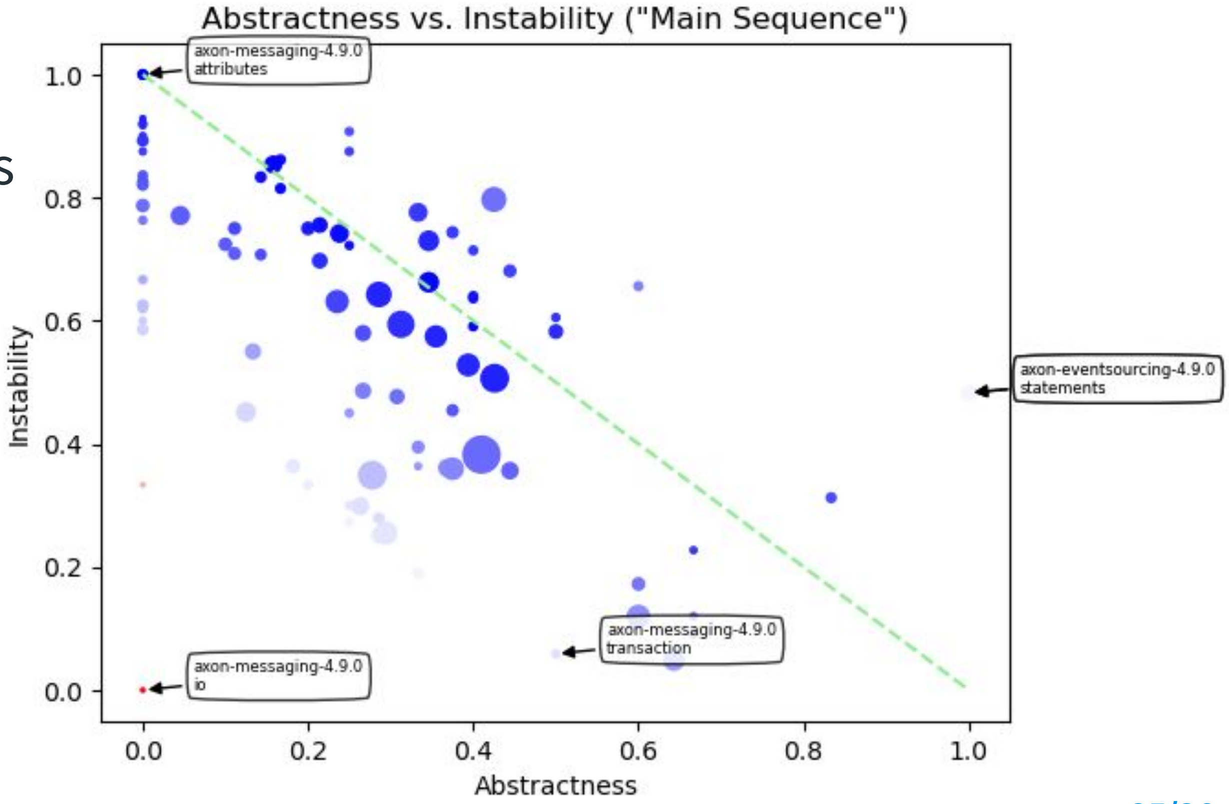
Visualize
Object Oriented Design Metrics by Robert C.Martin

High distance from the **green** line means

⇩

Hard to change (bottom left)

High abstraction (top right)



Reference: https://github.com/JohT/code-graph-analysis-pipeline/blob/main/results/AxonFramework-4.8.0/object-oriented-design-metrics/ObjectOrientedDesignMetrics.md

5.

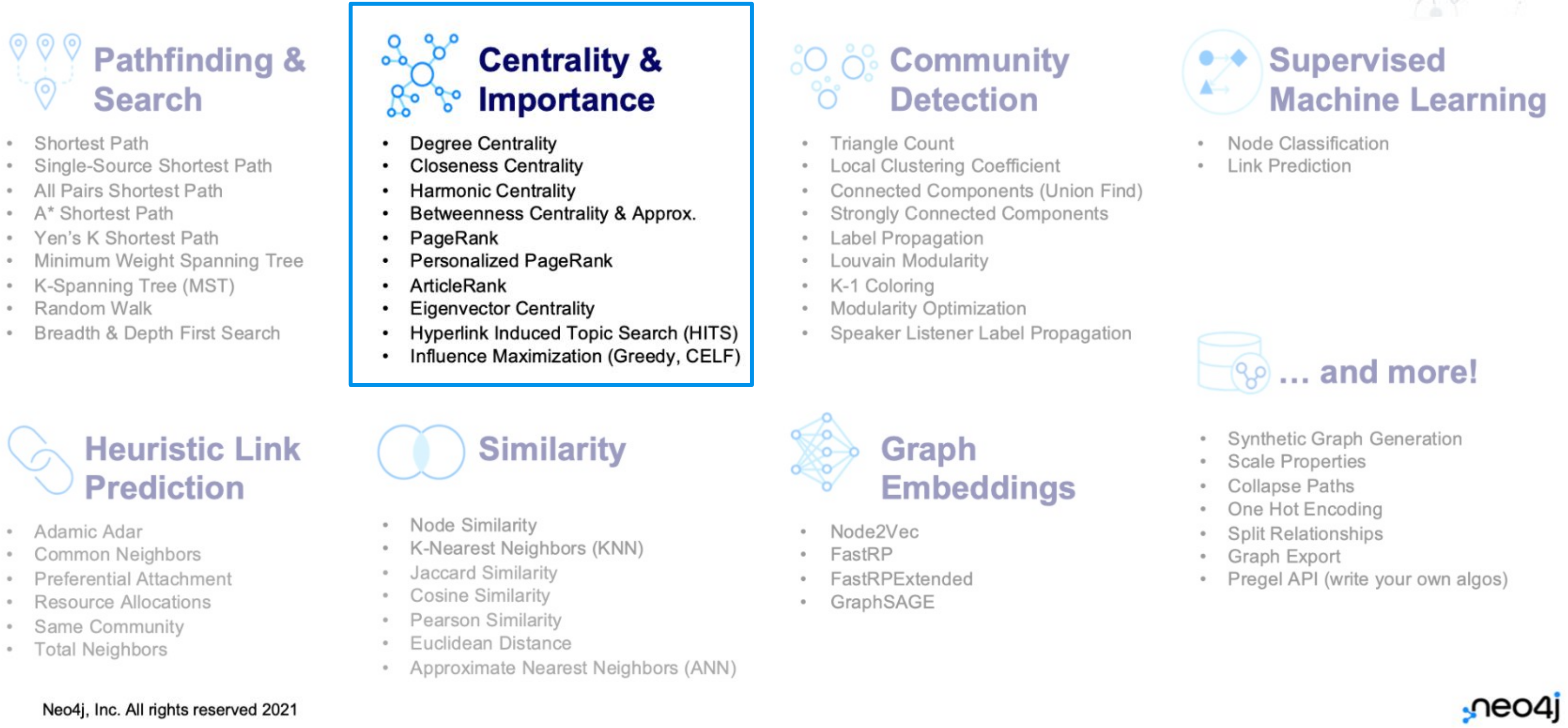
Graph Data Science

Algorithms ➡ Feature Engineering

1 — 2 — 3 — 4 — 5 — 6

Story — Introduction to Graphs — Overview of Disciplines — Graph Data Analysis — Graph Data Science — Tools

Graphs Algorithm Categories



Pathfinding & Search

- Shortest Path
- Single-Source Shortest Path
- All Pairs Shortest Path
- A* Shortest Path
- Yen's K Shortest Path
- Minimum Weight Spanning Tree
- K-Spanning Tree (MST)
- Random Walk
- Breadth & Depth First Search

Centrality & Importance

- Degree Centrality
- Closeness Centrality
- Harmonic Centrality
- Betweenness Centrality & Approx.
- PageRank
- Personalized PageRank
- ArticleRank
- Eigenvector Centrality
- Hyperlink Induced Topic Search (HITS)
- Influence Maximization (Greedy, CELF)

Community Detection

- Triangle Count
- Local Clustering Coefficient
- Connected Components (Union Find)
- Strongly Connected Components
- Label Propagation
- Louvain Modularity
- K-1 Coloring
- Modularity Optimization
- Speaker Listener Label Propagation

Supervised Machine Learning

- Node Classification
- Link Prediction

Heuristic Link Prediction

- Adamic Adar
- Common Neighbors
- Preferential Attachment
- Resource Allocations
- Same Community
- Total Neighbors

Similarity

- Node Similarity
- K-Nearest Neighbors (KNN)
- Jaccard Similarity
- Cosine Similarity
- Pearson Similarity
- Euclidean Distance
- Approximate Nearest Neighbors (ANN)

Graph Embeddings

- Node2Vec
- FastRP
- FastRPExtended
- GraphSAGE

… and more!

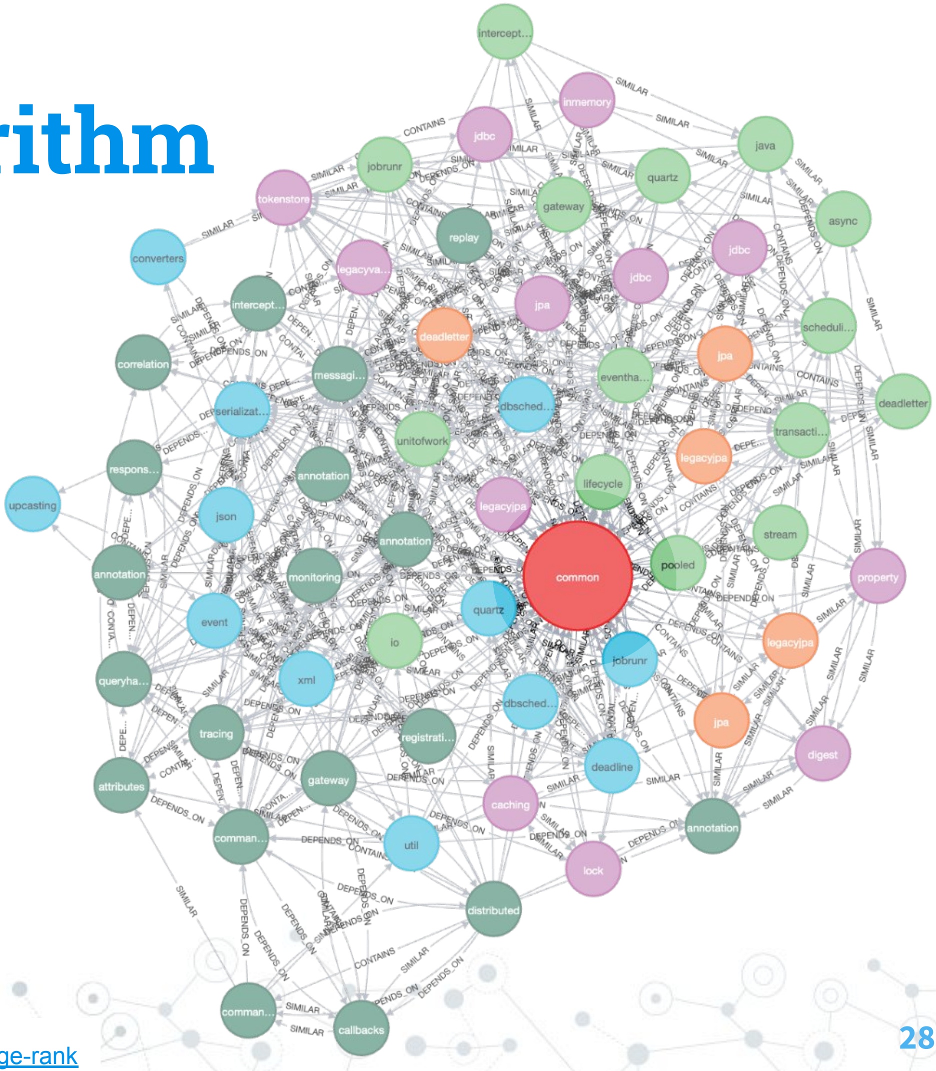
- Synthetic Graph Generation
- Scale Properties
- Collapse Paths
- One Hot Encoding
- Split Relationships
- Graph Export
- Pregel API (write your own algos)

Reference: https://de.slideshare.net/neo4j/graphs-for-data-science-and-machine-learning

Centrality

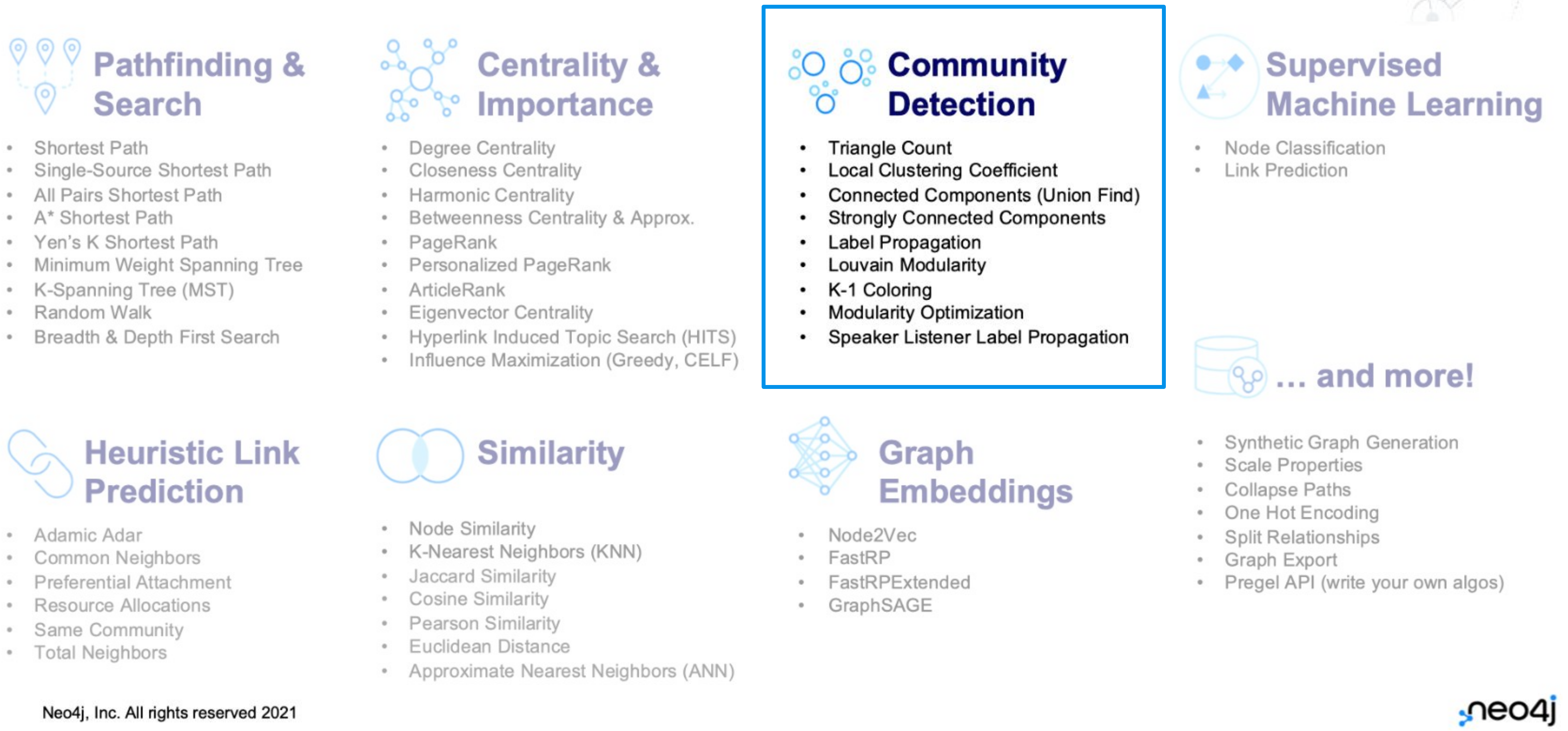
Page Rank Algorithm

- Importance
- Based on incoming relationships and the importance of the source nodes



Reference: https://neo4j.com/docs/graph-data-science/current/algorithms/page-rank

Graphs Algorithm Categories



Pathfinding & Search

- Shortest Path
- Single-Source Shortest Path
- All Pairs Shortest Path
- A* Shortest Path
- Yen's K Shortest Path
- Minimum Weight Spanning Tree
- K-Spanning Tree (MST)
- Random Walk
- Breadth & Depth First Search

Centrality & Importance

- Degree Centrality
- Closeness Centrality
- Harmonic Centrality
- Betweenness Centrality & Approx.
- PageRank
- Personalized PageRank
- ArticleRank
- Eigenvector Centrality
- Hyperlink Induced Topic Search (HITS)
- Influence Maximization (Greedy, CELF)

Community Detection

- Triangle Count
- Local Clustering Coefficient
- Connected Components (Union Find)
- Strongly Connected Components
- Label Propagation
- Louvain Modularity
- K-1 Coloring
- Modularity Optimization
- Speaker Listener Label Propagation

Supervised Machine Learning

- Node Classification
- Link Prediction

… and more!

- Synthetic Graph Generation
- Scale Properties
- Collapse Paths
- One Hot Encoding
- Split Relationships
- Graph Export
- Pregel API (write your own algos)

Heuristic Link Prediction

- Adamic Adar
- Common Neighbors
- Preferential Attachment
- Resource Allocations
- Same Community
- Total Neighbors

Similarity

- Node Similarity
- K-Nearest Neighbors (KNN)
- Jaccard Similarity
- Cosine Similarity
- Pearson Similarity
- Euclidean Distance
- Approximate Nearest Neighbors (ANN)

Graph Embeddings

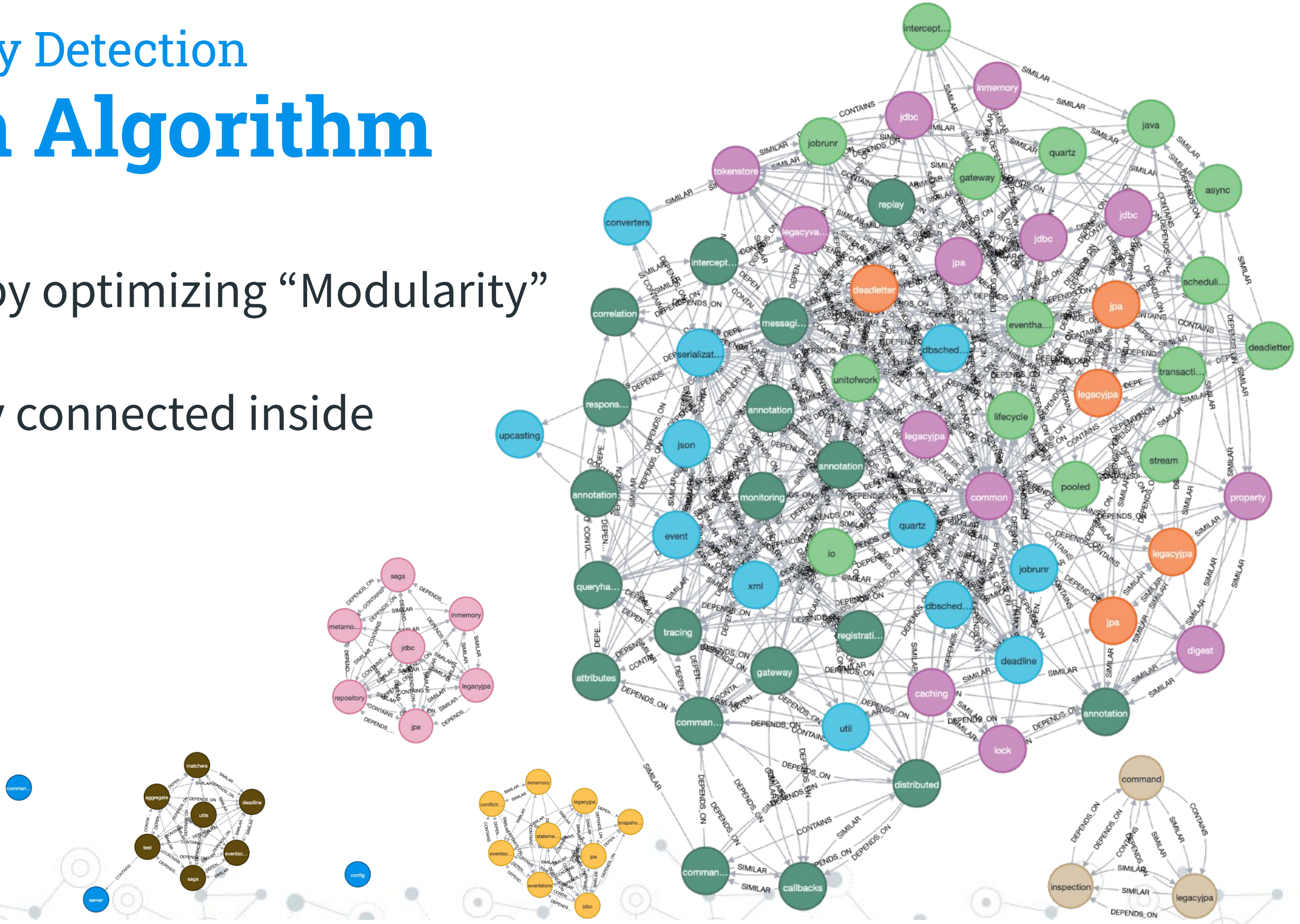
- Node2Vec
- FastRP
- FastRPExtended
- GraphSAGE

Reference: https://de.slideshare.net/neo4j/graphs-for-data-science-and-machine-learning

Community Detection

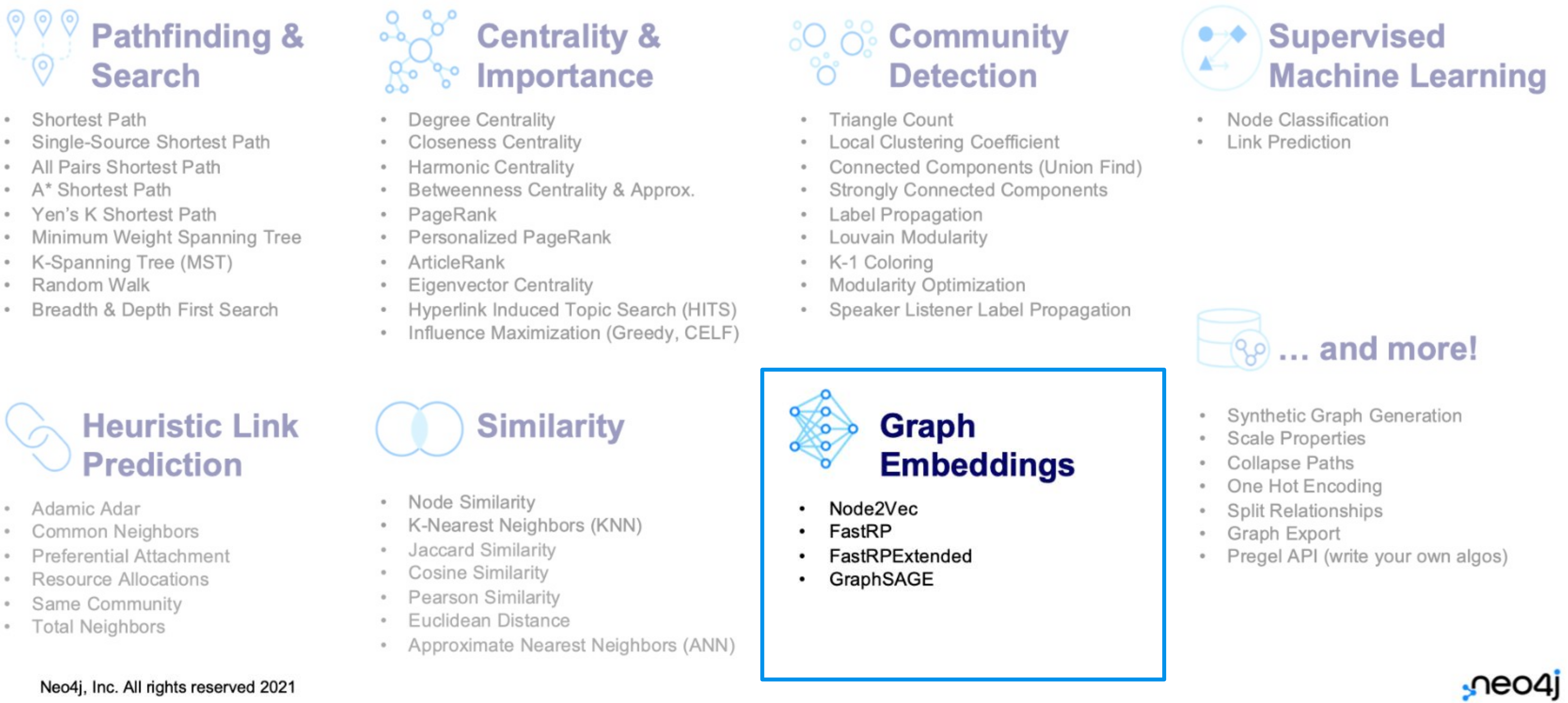
Leiden Algorithm

- Group by optimizing “Modularity”
- Densely connected inside



References: https://neo4j.com/docs/graph-data-science/current/algorithms/leiden, https://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/modularity.pdf

Graphs Algorithm Categories



Pathfinding & Search

- Shortest Path
- Single-Source Shortest Path
- All Pairs Shortest Path
- A* Shortest Path
- Yen's K Shortest Path
- Minimum Weight Spanning Tree
- K-Spanning Tree (MST)
- Random Walk
- Breadth & Depth First Search

Centrality & Importance

- Degree Centrality
- Closeness Centrality
- Harmonic Centrality
- Betweenness Centrality & Approx.
- PageRank
- Personalized PageRank
- ArticleRank
- Eigenvector Centrality
- Hyperlink Induced Topic Search (HITS)
- Influence Maximization (Greedy, CELF)

Community Detection

- Triangle Count
- Local Clustering Coefficient
- Connected Components (Union Find)
- Strongly Connected Components
- Label Propagation
- Louvain Modularity
- K-1 Coloring
- Modularity Optimization
- Speaker Listener Label Propagation

Supervised Machine Learning

- Node Classification
- Link Prediction

Heuristic Link Prediction

- Adamic Adar
- Common Neighbors
- Preferential Attachment
- Resource Allocations
- Same Community
- Total Neighbors

Similarity

- Node Similarity
- K-Nearest Neighbors (KNN)
- Jaccard Similarity
- Cosine Similarity
- Pearson Similarity
- Euclidean Distance
- Approximate Nearest Neighbors (ANN)

Graph Embeddings

- Node2Vec
- FastRP
- FastRPExtended
- GraphSAGE

… and more!

- Synthetic Graph Generation
- Scale Properties
- Collapse Paths
- One Hot Encoding
- Split Relationships
- Graph Export
- Pregel API (write your own algos)

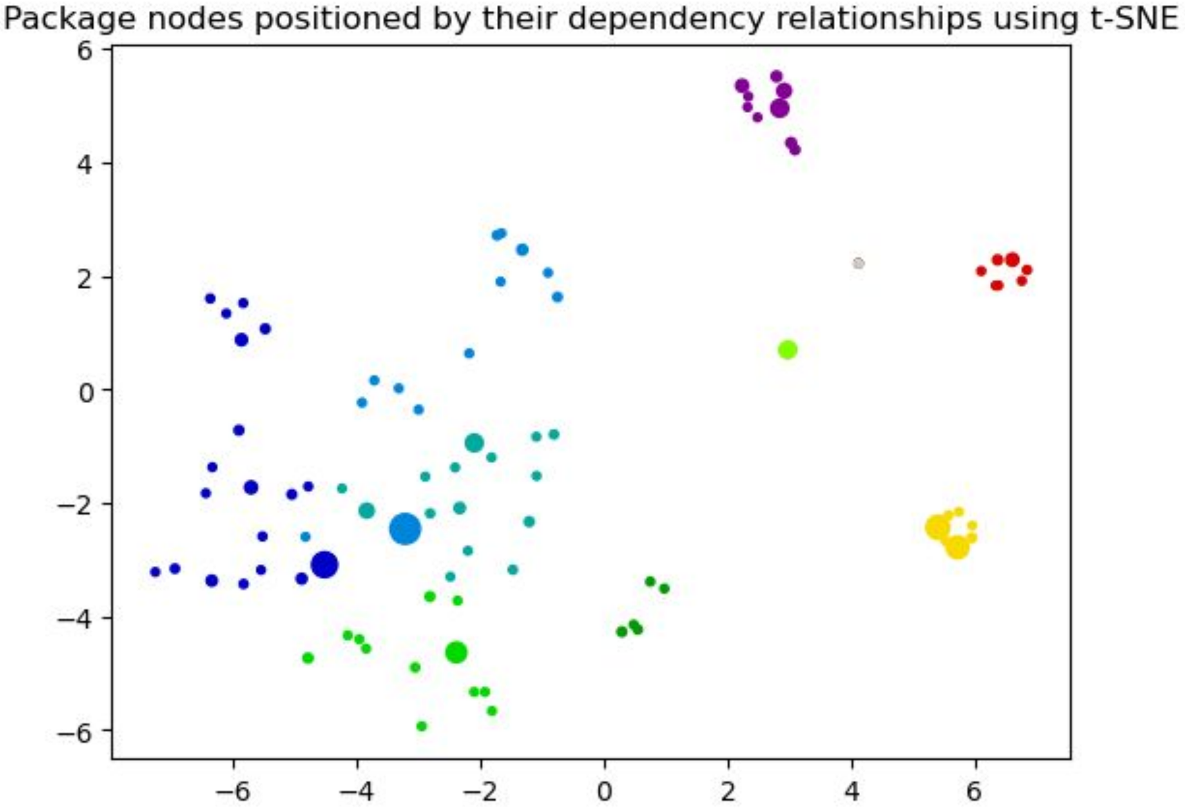
neo4j

Reference: https://de.slideshare.net/neo4j/graphs-for-data-science-and-machine-learning

Node Embeddings

Fast Random Projection + t-SNE

Dimensionality reduction while preserving most of the distance information



References:

- https://neo4j.com/docs/graph-data-science/current/machine-learning/node-embeddings/fastrp
- https://github.com/JohT/code-graph-analysis-pipeline/blob/main/results/AxonFramework-4.9.0/node-embeddings/NodeEmbeddings.md

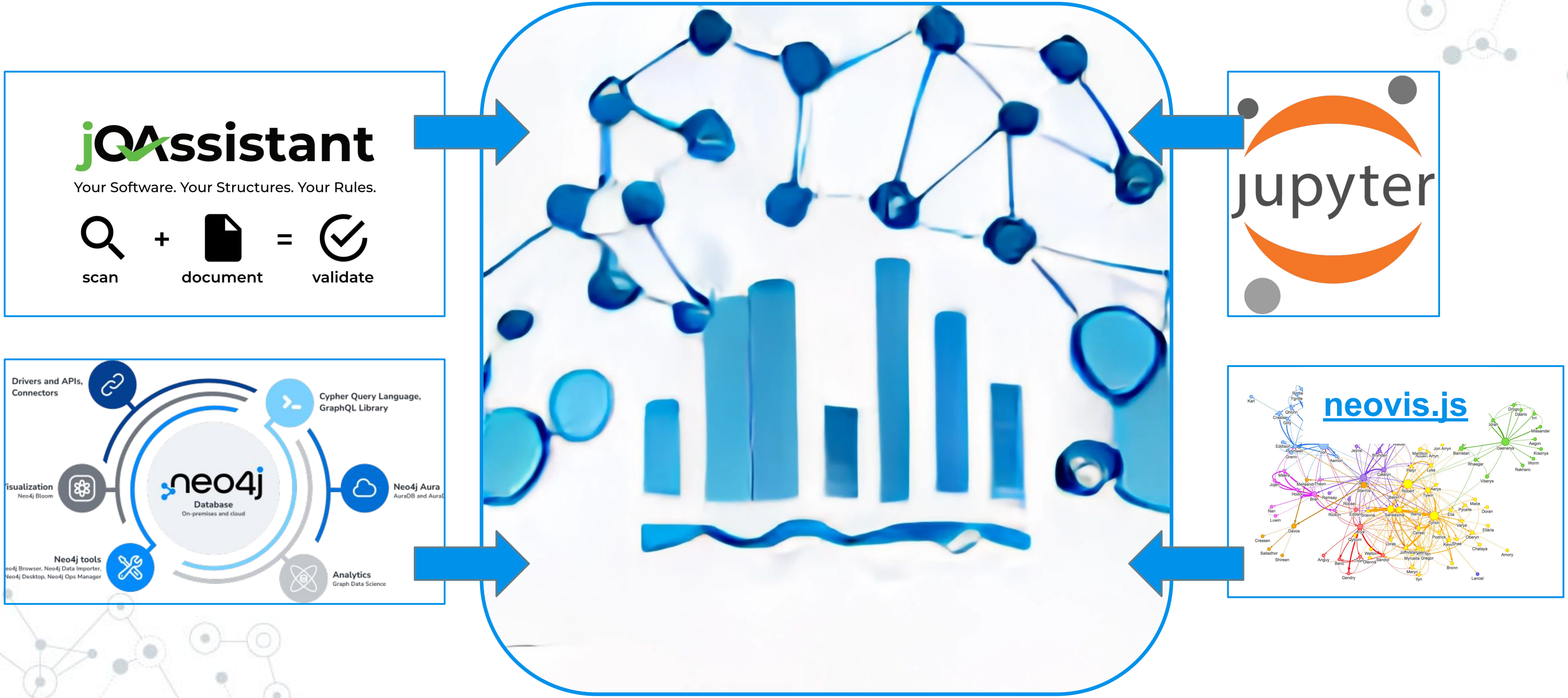
6.

Tools

jQAssistant and Neo4j inside a Pipeline

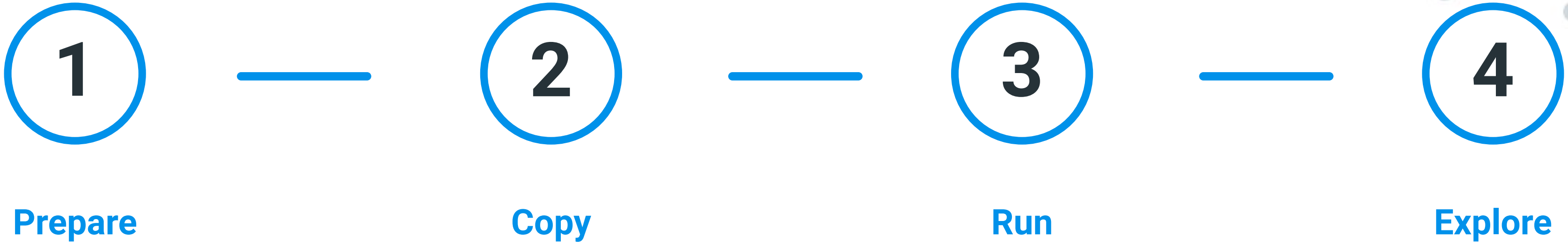
1 Story — 2 Introduction to Graphs — 3 Overview of Disciplines — 4 Graph Data Analysis — 5 Graph Data Science — 6 Tools

Code Graph Analysis Pipeline


jQAssistant
Your Software. Your Structures. Your Rules.
scan
+
document
=
validate
Drivers and APIs, Connectors
Cypher Query Language, GraphQL Library
isualization
Neo4j Bloom
neo4j
Database
On-premises and cloud
Neo4j Aura
AuraDB and AuraI
Neo4j tools
eo4j Browser, Neo4j Data Importer, Neo4j Desktop, Neo4j Ops Manager
Analytics
Graph Data Science
jupyter
neovis.js

https://github.com/JohT/code-graph-analysis-pipeline

Code Graph Analysis Pipeline

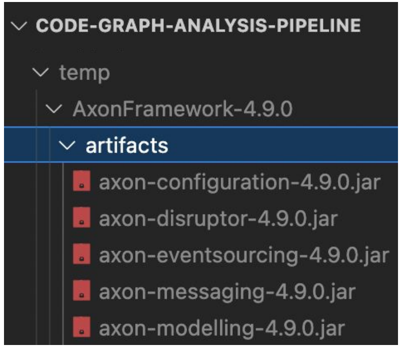


1 Prepare

Clone the Respository
Use Java 17
Set NEO4J_INITIAL_PASSWORD

2 Copy

Copy the artifacts to analyze



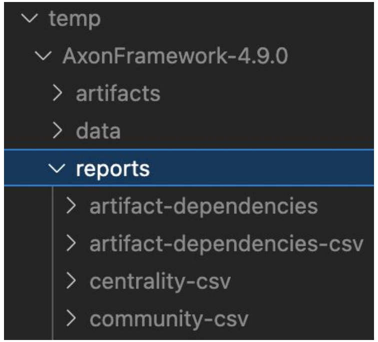
```
⌄ CODE-GRAPH-ANALYSIS-PIPELINE
  ⌄ temp
    ⌄ AxonFramework-4.9.0
      ⌄ artifacts
          axon-configuration-4.9.0.jar
          axon-disruptor-4.9.0.jar
          axon-eventsourcing-4.9.0.jar
          axon-messaging-4.9.0.jar
          axon-modelling-4.9.0.jar
```

3 Run

```
./../../scripts/analysis/
analyse.sh --report Csv
```

4 Explore

Explore the results



```
⌄ temp
  ⌄ AxonFramework-4.9.0
    › artifacts
    › data
    ⌄ reports
      › artifact-dependencies
      › artifact-dependencies-csv
      › centrality-csv
      › community-csv
```

https://github.com/JohT/code-graph-analysis-pipeline

Summary

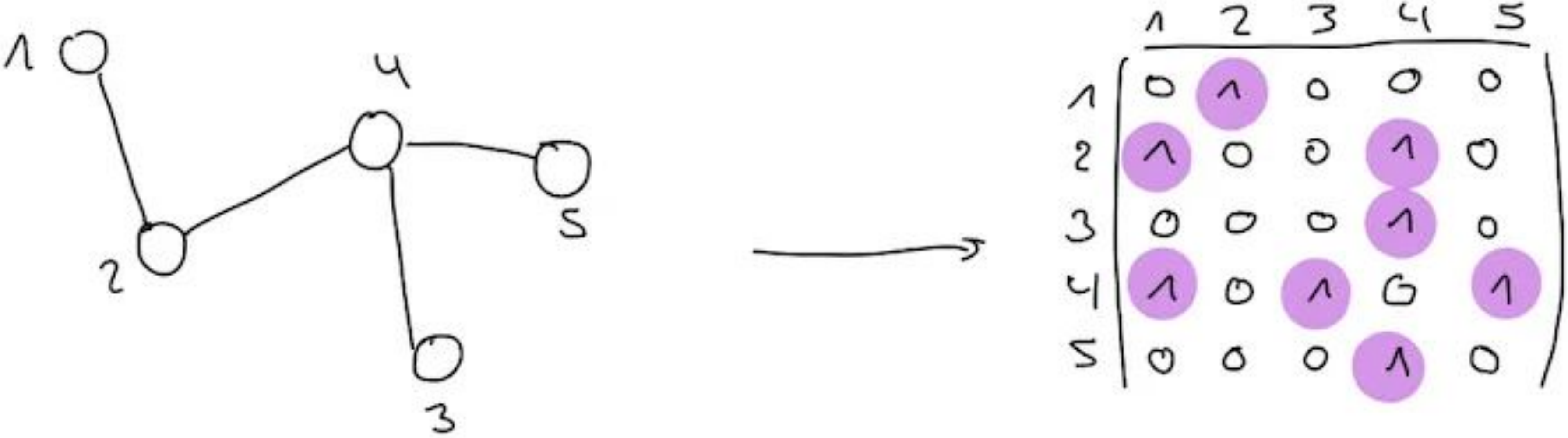
- Story
- Introduction to Graphs
- Query
- Analysis
- Algorithms
- Preparation for Machine Learning

Find me and this project on GitHub

https://github.com/
JohT/code-graph-analysis-pipeline

Node Embeddings

Adjacency Matrix



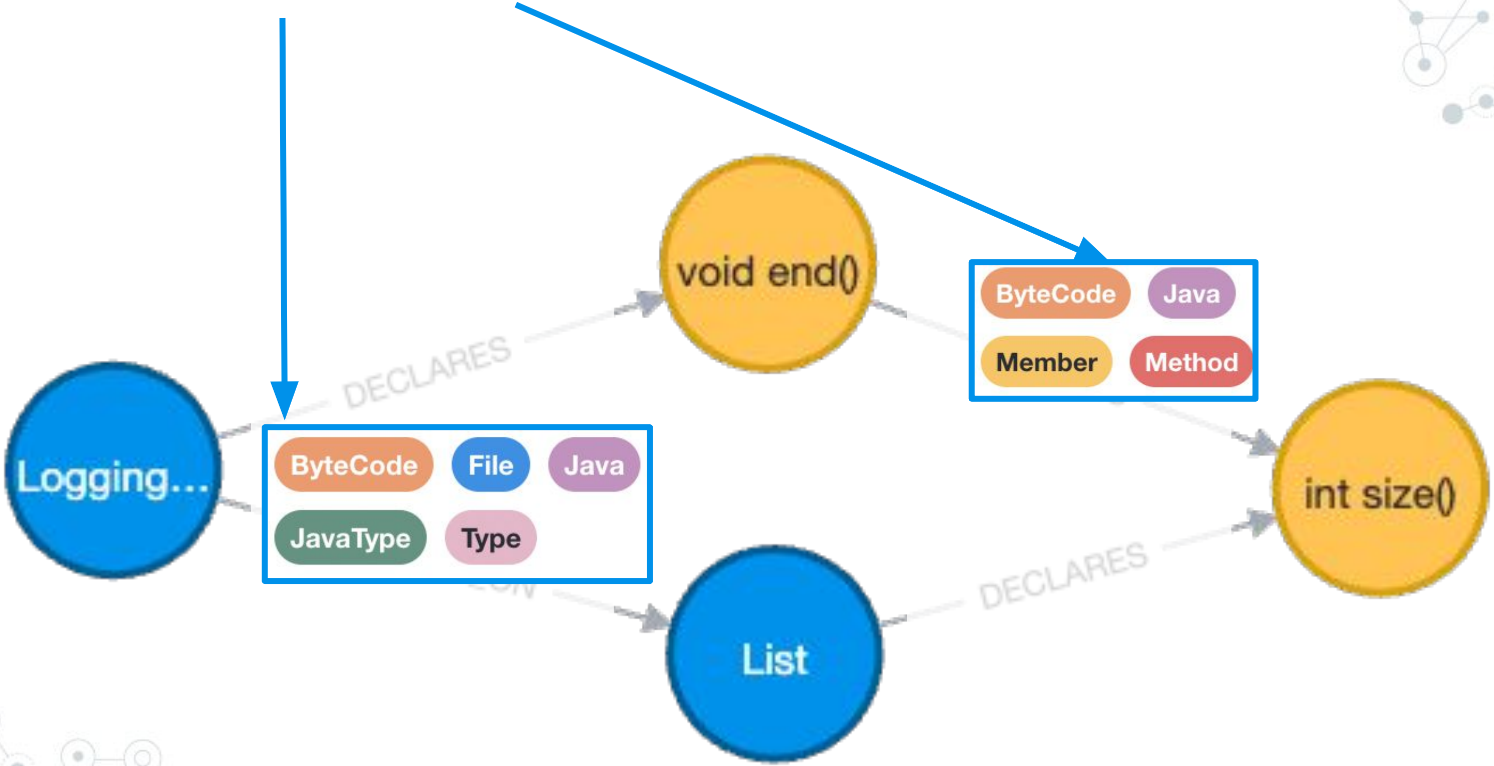
$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 1 & 0 \\ 4 & 1 & 0 & 1 & 0 & 1 \\ 5 & 0 & 0 & 0 & 1 & 0 \end{array}$$

N^2 → doesn’t scale → gets to large

Reference: https://towardsdatascience.com/node-embeddings-for-beginners-554ab1625d98

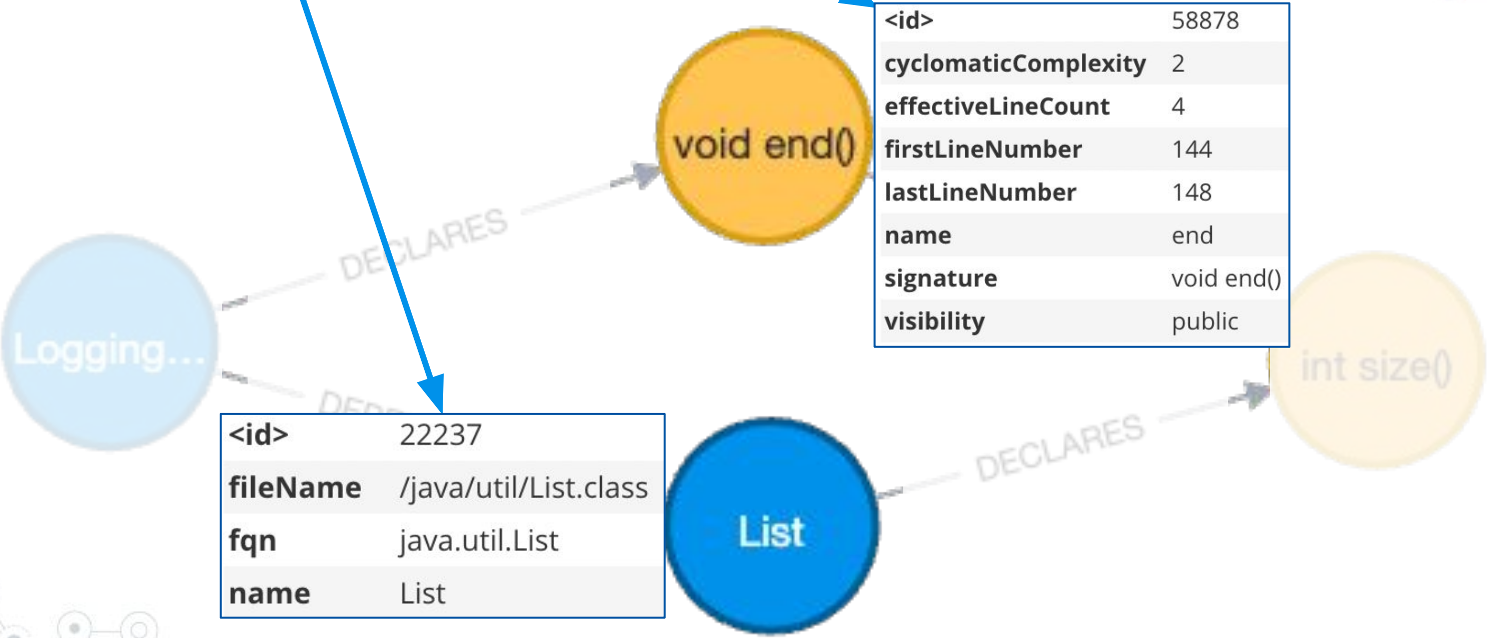
Graph Database Elements

Node Labels



Graph Database Elements

Node Properties

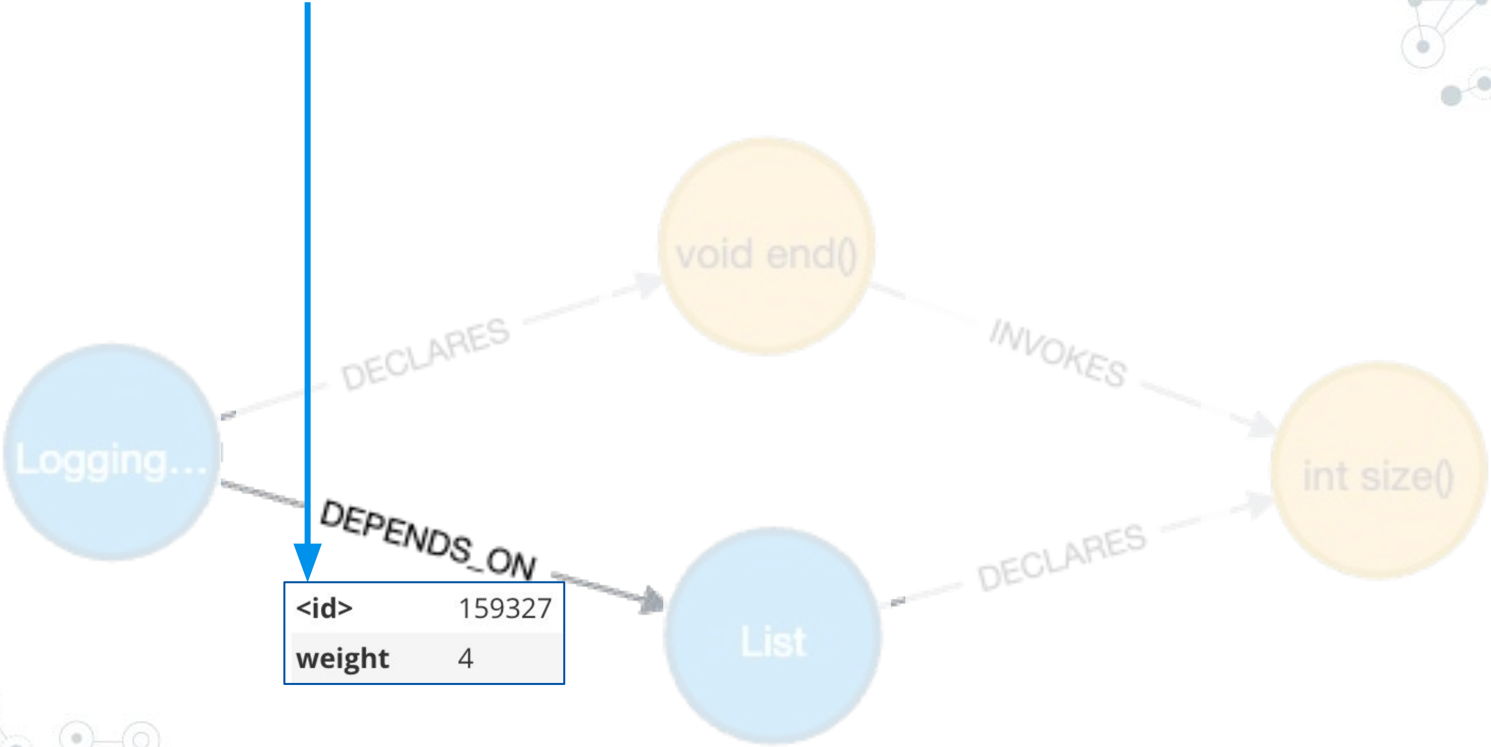


<id>	58878
cyclomaticComplexity	2
effectiveLineCount	4
firstLineNumber	144
lastLineNumber	148
name	end
signature	void end()
visibility	public

<id>	22237
fileName	/java/util/List.class
fqn	java.util.List
name	List

Graph Database Elements

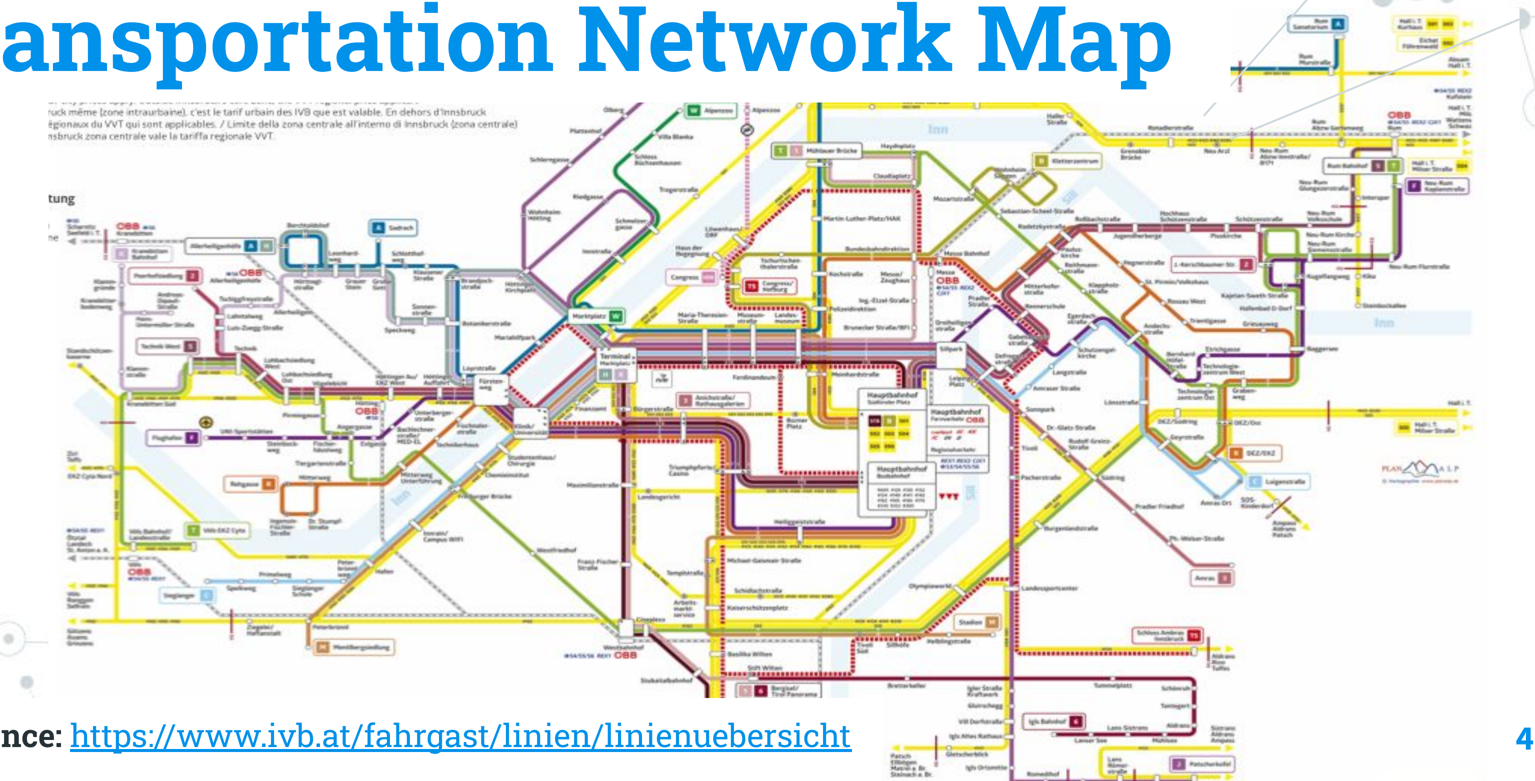
Relationship Properties



<id>	159327
weight	4

Graph Examples

Transportation Network Map


ruck même (zone intraurbaine), c'est le tarif urbain des IVB que est valable. En dehors d'Innsbruck
égionaux du VVT qui sont applicables. / Limite della zona centrale all'interno di Innsbruck (zona centrale)
nsbruck zona centrale vale la tariffa regionale VVT.
tung
Alpenzoo
Inn
Marktplatz
Terminal Marktplatz
Congress
Hauptbahnhof
Sillpark
ÖBB
Flughafen
Technik West
Westbahnhof
Stadion
PLAN ALP

Reference: https://www.ivb.at/fahrgast/linien/linienuebersicht