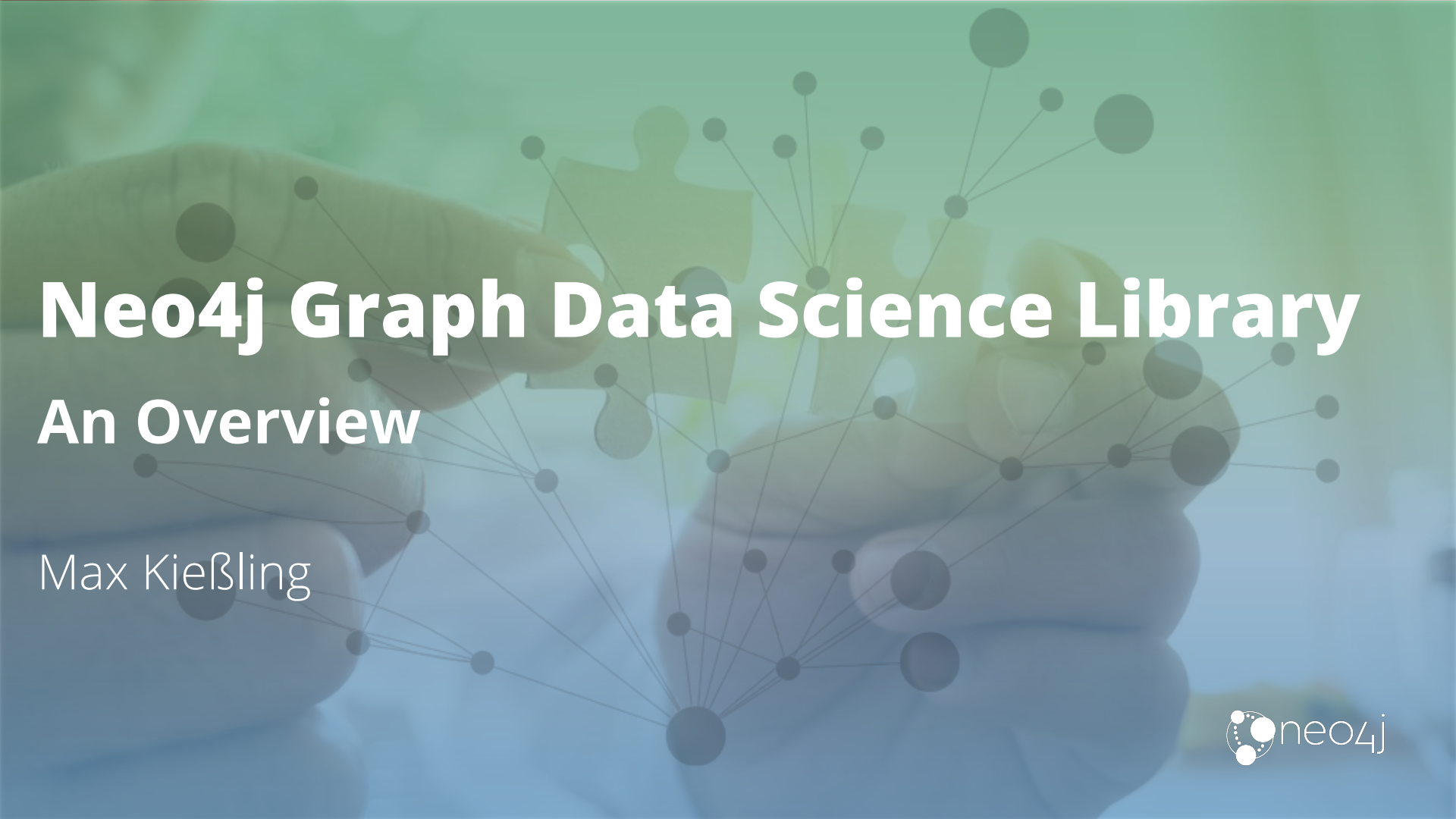


Neo4j Graph Data Science Library

An Overview

Max Kießling

neo4j

What is the Graph Data Science Library?

- Open Source Neo4j Add-On for graph analytics
- Provides a set of high performance graph algorithms
  - Community Detection / Clustering (e.g. Label Propagation)
  - Similarity Calculation (e.g. NodeSimilarity)
  - Centrality Algorithms (e.g. PageRank)
  - PathFinding (e.g. Dijkstra)
  - Link Prediction (e.g. Adamic Adar)
  - and more
- APIs for implementing custom algorithms (e.g. Pregel)

Neo4j GDS - Timeline

Q1 2017

Development started as

Neo4j Contrib - Graph Algorithms

organized by Neo4j Labs,
developed by AVGL

Q1 2019

Neo4j Product Engineering

takes over the project

Q1 2020

Productization of the library

Open Source Preview Release

Q2 2020

Neo4j Graph Data Science Library

Release 1.0

neo4j

Local Patterns to Global Computation

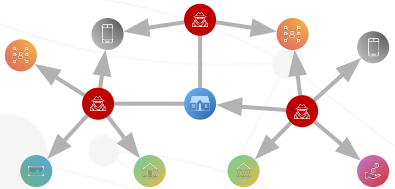
Query (e.g. Cypher/SQL)

Real-time, local decisioning and pattern matching

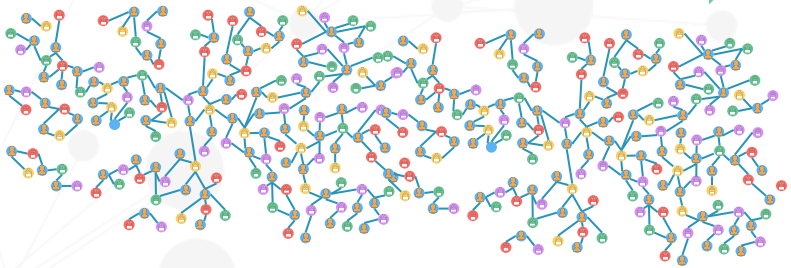
Graph Algorithms Libraries

Global analysis and iterations

Local Patterns → **Global Computation**

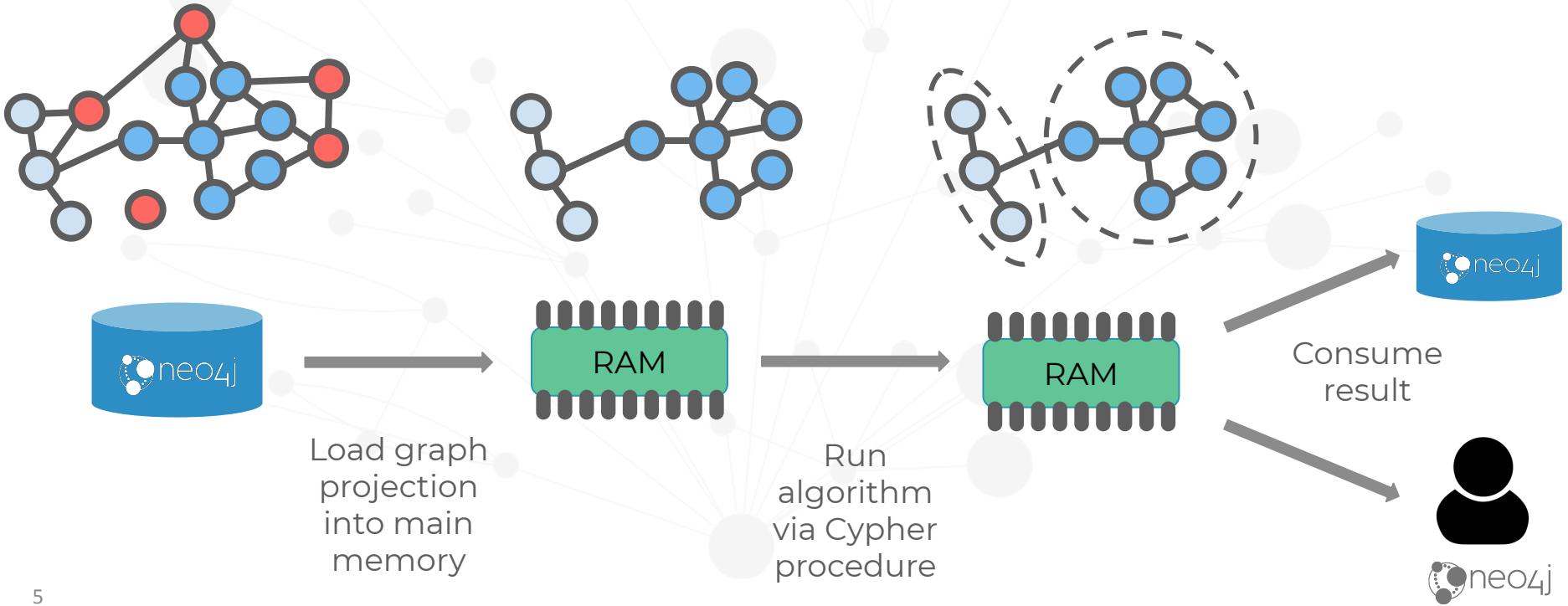


You know what you're looking for and making a decision



You're learning the overall structure of a network, updating data, and predicting

Workflow


neo4j
RAM
RAM
neo4j
Consume result
Load graph projection into main memory
Run algorithm via Cypher procedure
neo4j

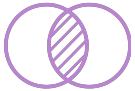
Available Algorithms

Community Detection

- **Label Propagation**
- **Louvain**
- **Weakly Connected Components**
- Triangle Count
- Clustering Coefficients
- Strongly Connected Components
- Balanced Triad (identification)

Centrality / Importance

- **PageRank**
- **Personalized PageRank**
- Degree Centrality
- Closeness Centrality
- Betweenness Centrality
- ArticleRank
- Eigenvector Centrality



Similarity

- **Node Similarity**
- Euclidean Distance
- Cosine Similarity
- Overlap Similarity
- Pearson Similarity

Link Prediction

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

Pathfinding & Search

- Parallel Breadth First Search
- Parallel Depth First Search
- Shortest Path
- Minimum Spanning Tree
- A* Shortest Path
- Yen's K Shortest Path
- K-Spanning Tree (MST)
- Random Walk

Demo Time!

GDS - Algo Syntax

```
CALL gds.<algo-name>.<mode>(
  graphName: STRING,
  configuration: MAP
)
```

Available Modes:

- `write`: writes results to the Neo4j database and returns a summary of the results.
- `stats`: runs the algorithm and only reports statistics.
- `stream`: streams results back to the user.

```
CALL gds.wcc.write(
  "got-interactions",
  {
    writeProperty: "component",
    consecutiveIds: true
  }
) YIELDS writeMillis, componentCount
```

```
CALL gds.wcc.stream(
  "got-interactions",
  {}
) YIELDS nodeId, componentId
```

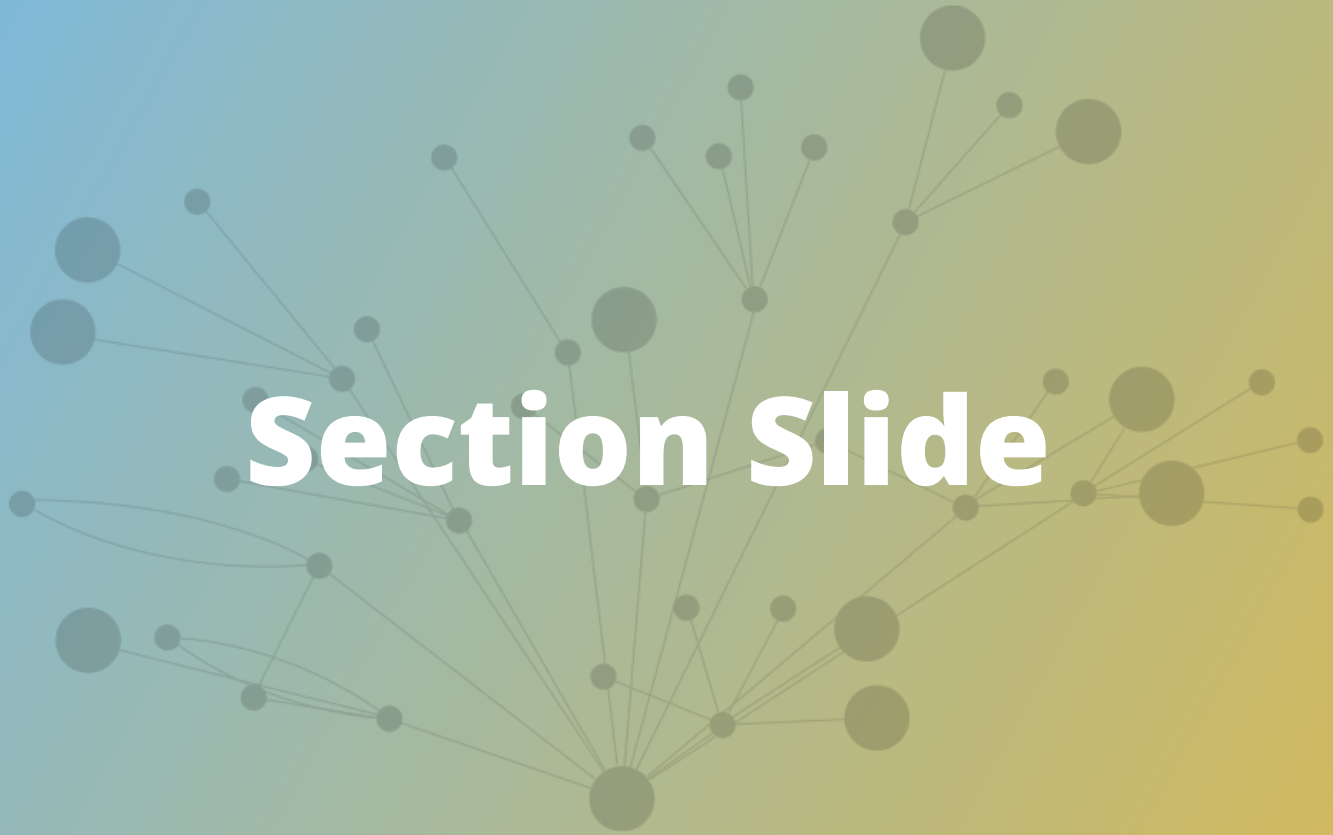
Take a look!

The Neo4j Graph Data Science Library
is Open Source

https://github.com/neo4j/graph-data-science

Section Slide

Section Slide



Section Slide