

AGENDA

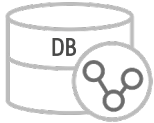
Marco De Luca, Neo4j Field Engineering

1. Neo4j Graph Datenbank - Was oft unklar ist
2. Knowledge Graph als Grundlage für weitere Mehrwerte
3. Vorgehensweise - Aufbau eines Knowledge Graphen (KG)
4. Tools für den Einstieg in Neo4j
 - Daten Modellierung
 - Daten laden
 - Data analysieren
5. Demo: Neo4j Workspace*



Neo4j Graph Datenbank - Was oft unklar ist

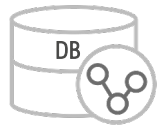
Neo4j ist eine **NORMALE**
Datenbank (DB)



- Vergleichbar mit anderen DBs wie Postgres, MySQL / MariaDB, Oracle, HANA DB, etc.
- Daten sicher speichern, verwalten und abfragen
- Backup + PiT Recovery
- Hoch skalierbar durch Clustering

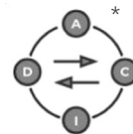
Neo4j Graph Datenbank - Was oft unklar ist

Neo4j ist eine **NORMALE** Datenbank (DB)



- Vergleichbar mit anderen DBs wie Postgres, MySQL / MariaDB, Oracle, HANA DB, etc.
- Daten sicher speichern, verwalten und abfragen
- Backup + PiT Recovery
- Hoch skalierbar durch Clustering

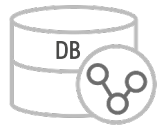
Neo4j ist “ACID compliant” und somit transaktionssicher



- Auf deutsch AKID konform
- Daten werden auf Transaktionsebene sicher gespeichert ohne Verlust bei Ausfällen
- AKID – Atomarität, Konsistenz, Isolation und Dauerhaftigkeit

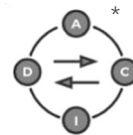
Neo4j Graph Datenbank - Was oft unklar ist

Neo4j ist eine **NORMALE** Datenbank (DB)



- Vergleichbar mit anderen DBs wie Postgres, MySQL / MariaDB, Oracle, HANA DB, etc.
- Daten sicher speichern, verwalten und abfragen
- Backup + PiT Recovery
- Hoch skalierbar durch Clustering

Neo4j ist “ACID compliant” und somit transaktionssicher



- Auf deutsch AKID konform
- Daten werden auf Transaktionsebene sicher gespeichert ohne Verlust bei Ausfällen
- **AKID** – Atomarität, Konsistenz, Isolation und Dauerhaftigkeit

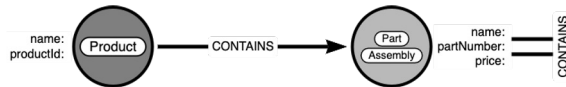
Neo4j wird zu 80%+ für OLTP Workloads genutzt



- > 80% der Kunden nutzen Neo4j in erster Instanz als normale Datenbank
- Daten speichern, ändern und löschen
- Mit gleicher oder oft besserer Performance als bei relationale DBs

Aber was macht eine native Graph-Datenbank anders?

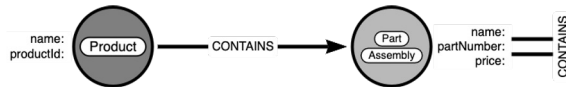
Die Speicherung der Daten



- Daten werden verknüpft gespeichert
- Daten liegen auf der Platte nah “beieinander”
Stichwort: “Index-free Adjacency”
- Sie speichert **Knoten** und **Verbindungen** statt Zeilen und Spalten
- **Semantik** ist abbildbar!

Aber was macht eine native Graph-Datenbank anders?

Die Speicherung der Daten



- Daten werden verknüpft gespeichert
- Daten liegen auf der Platte nah “beieinander”
Stichwort: “Index-free Adjacency”
- Sie speichert **Knoten** und **Verbindungen** statt Zeilen und Spalten
- **Semantik** ist abbildbar!

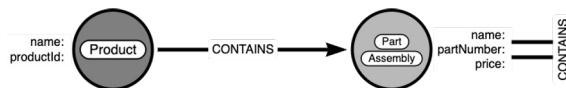
Die Abfragen der Daten

`(:Product) -[:CONTAINS]->(:Part)`

- Cypher Query Language statt SQL (ISO -> GQL)
- Einfacher, weniger Code-Zeilen, besser lesbar
- Abfragen bis in Tiefen von 100+ Hops, vergleichbar von SQL Joins über 100+ Tabellen!

Aber was macht eine native Graph-Datenbank anders?

Die Speicherung der Daten



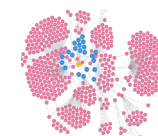
- Daten werden verknüpft gespeichert
- Daten liegen auf der Platte nah “beieinander”
Stichwort: “Index-free Adjacency”
- Sie speichert **Knoten** und **Verbindungen** statt Zeilen und Spalten
- **Semantik** ist abbildbar!

Die Abfragen der Daten

```
(:Product) -[:CONTAINS] -> (:Part)
```

- Cypher Query Language statt SQL (ISO -> GQL)
- Einfacher, weniger Code-Zeilen, besser lesbar
- Abfragen bis in Tiefen von 100+ Hops, vergleichbar von SQL Joins über 100+ Tabellen!

Komplexe Datennetzwerke speichern



- Speicherung und Analyse von komplexen Zusammenhängen
- Auswertung von Daten, die bis jetzt nie/unzureichend miteinander verknüpft wurden
- Erweiterbar mit Data Science Algorithmen

Was und Warum – Knowledge Graphen

“Ist meine **Wissensbasis**.”

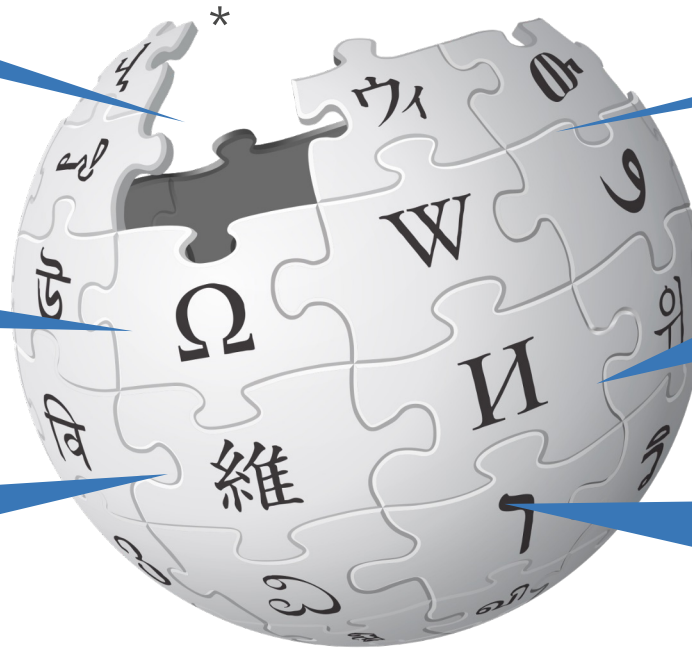
(Datenbank als Graph gespeichert)

“Beantwortet viele
grundsätzlichen Fragen.”

(einfache bis komplexe Queries)

“Verknüpft meine
“*Kronjuweldaten*” in einem
Daten- pool.”

(Daten aus verschiedensten Quellen!)



“Semantik zur Erweiterung des
Wissens.”

“Wissen weiterentwickelt zur
Basis für Entscheidungen ...”

“... und dann bis hin zur
Erstellung von Vorhersagen mit
maschinellem Lernen.”

Vorgehensweise - Aufbau eines Knowledge Graphen (grob)

1. Prüfung des Problems - Ist Graph die Lösung?
2. Fachbereich + IT Personal - Anforderungen niederschreiben
3. Aus den Anforderungen ein Datenmodell ableiten

} Vorbereitung

Vorgehensweise - Aufbau eines Knowledge Graphen (grob)

1. Prüfung des Problems - Ist Graph die Lösung?
2. Fachbereich + IT Personal - Anforderungen niederschreiben
3. Aus den Anforderungen ein Datenmodell ableiten



4. Daten präparieren und laden
5. Fragen/Anforderungen versuchen zu beantworten
6. Datenmodell überarbeiten, für bessere/schnellere Antworten

} Vorbereitung

} Rekursive Arbeit

Vorgehensweise - Aufbau eines Knowledge Graphen (grob)

1. Prüfung des Problems - Ist Graph die Lösung?
2. Fachbereich + IT Personal - Anforderungen niederschreiben
3. Aus den Anforderungen ein Datenmodell ableiten

} Vorbereitung



4. Daten präparieren und laden
5. Fragen/Anforderungen versuchen zu beantworten
6. Datenmodell überarbeiten, für bessere/schnellere Antworten

} Rekursive Arbeit

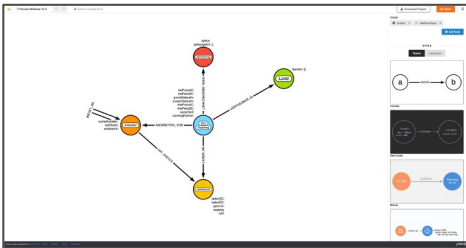


7. ETL Pipeline aufbauen
8. Skripte, Monitoring und ggf. Applikation entwickeln/einkaufen

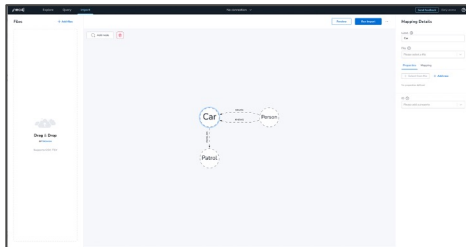
} Erster Pilot

Tools für den Einstieg

Daten modellieren



arrows.app



[Neo4j](#)
[Workspace*](#)

Daten laden

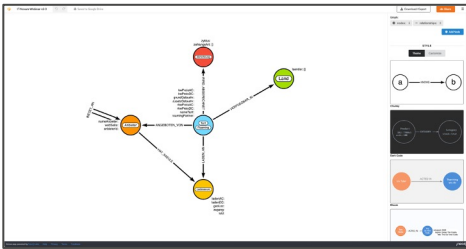
Daten speichern

Daten analysieren

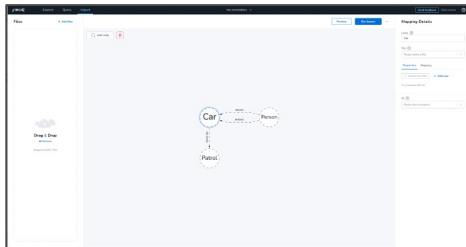
* Roadmap Tool in Preview + initial als SaaS Tool, on-prem später

Tools für den Einstieg

Daten modellieren



[arrows.app](#)

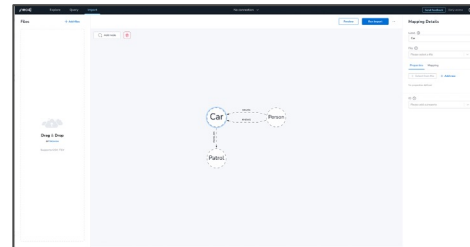


[Neo4j
Workspace*](#)

Daten laden

- Cypher CSV Load
- APOC Library
- neo4j admin import
- ETL Tool
- Python API

[Daten laden, etc.](#)



[Neo4j
Workspace*](#)

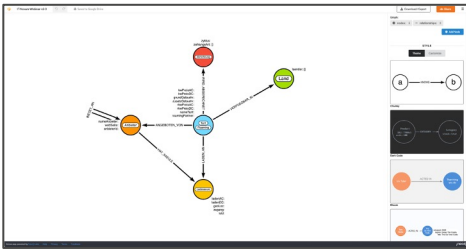
Daten speichern

Daten analysieren

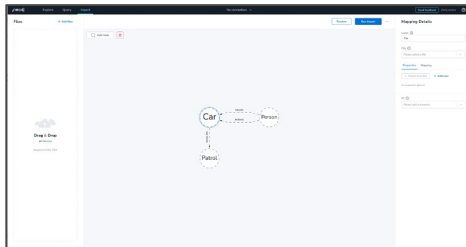
* Roadmap Tool in Preview + initial als SaaS Tool, on-prem später

Tools für den Einstieg

Daten modellieren



arrows.app

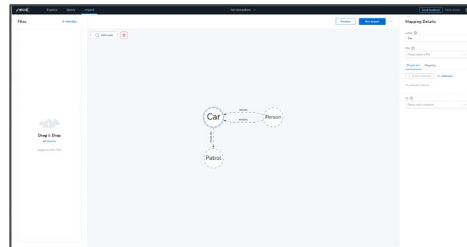


Neo4j
Workspace*

Daten laden

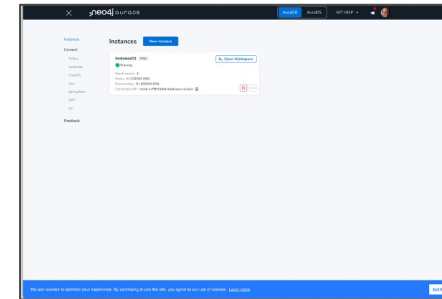
- Cypher CSV Load
- APOC Library
- neo4j admin import
- ETL Tool
- Python API

Daten laden, etc.

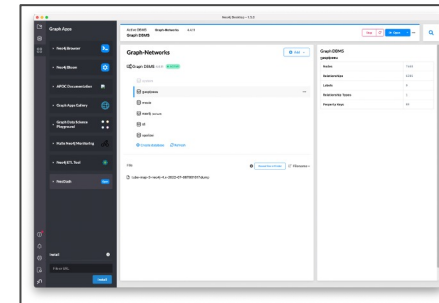


Neo4j
Workspace*

Daten speichern



Neo4j Aura



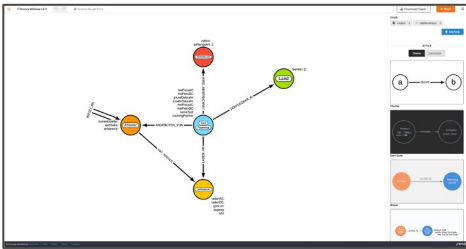
Neo4j Desktop

Daten analysieren

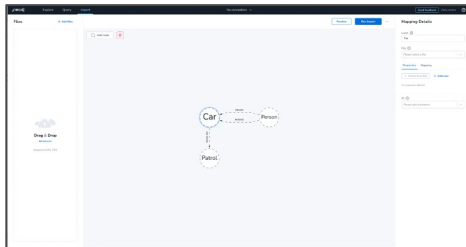
* Roadmap Tool in Preview + initial als SaaS Tool, on-prem später

Tools für den Einstieg

Daten modellieren



[arrows.app](#)

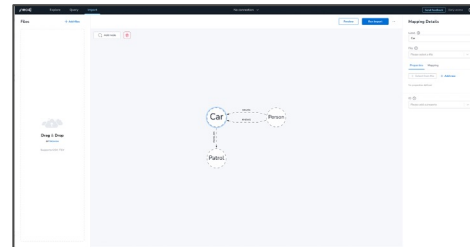


[Neo4j
Workspace*](#)

Daten laden

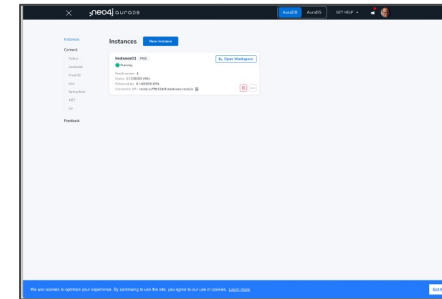
- Cypher CSV Load
- APOC Library
- neo4j admin import
- ETL Tool
- Python API

[Daten laden, etc.](#)

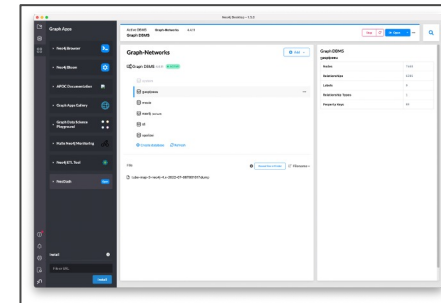


[Neo4j
Workspace*](#)

Daten speichern



[Neo4j Aura](#)

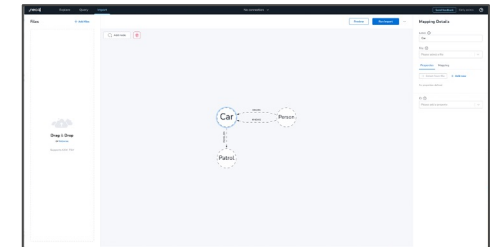


[Neo4j Desktop](#)

Daten analysieren



[Neo4j Bloom](#)



[Neo4j
Workspace*](#)

* Roadmap Tool in Preview + initial als SaaS Tool, on-prem später

DEMO TIME!

NEO4j Workspace
(Preview)

NEO4j Aura (SaaS)

NEO4j Desktop

Demo Daten, Folien, Datenmodelle, etc. auf Github unter:

github.com/luzidl/it-novum-webinar_112022

Vielen Dank für Ihre Aufmerksamkeit



Michael Deuchert

Team Lead Data Analytics

T +49 661 103-365

E michael.deuchert@it-novum.com



Marco De Luca

Principal Solutions Architect

T +49 151 539 44 333

E marco.deluca@neo4j.com



it-novum GmbH Deutschland

Hauptsitz: Edeltzeller Straße 44, 36043 Fulda
Niederlassung: Ruhrallee 9, 44139 Dortmund
Niederlassung: Kaiserswerther Str. 229, 40474 Düsseldorf

it-novum Zweigniederlassung Österreich

Ausstellungsstraße 50 / Zugang C
1020 Wien

it-novum Schweiz GmbH

Seestrasse 97
8800 Thalwil