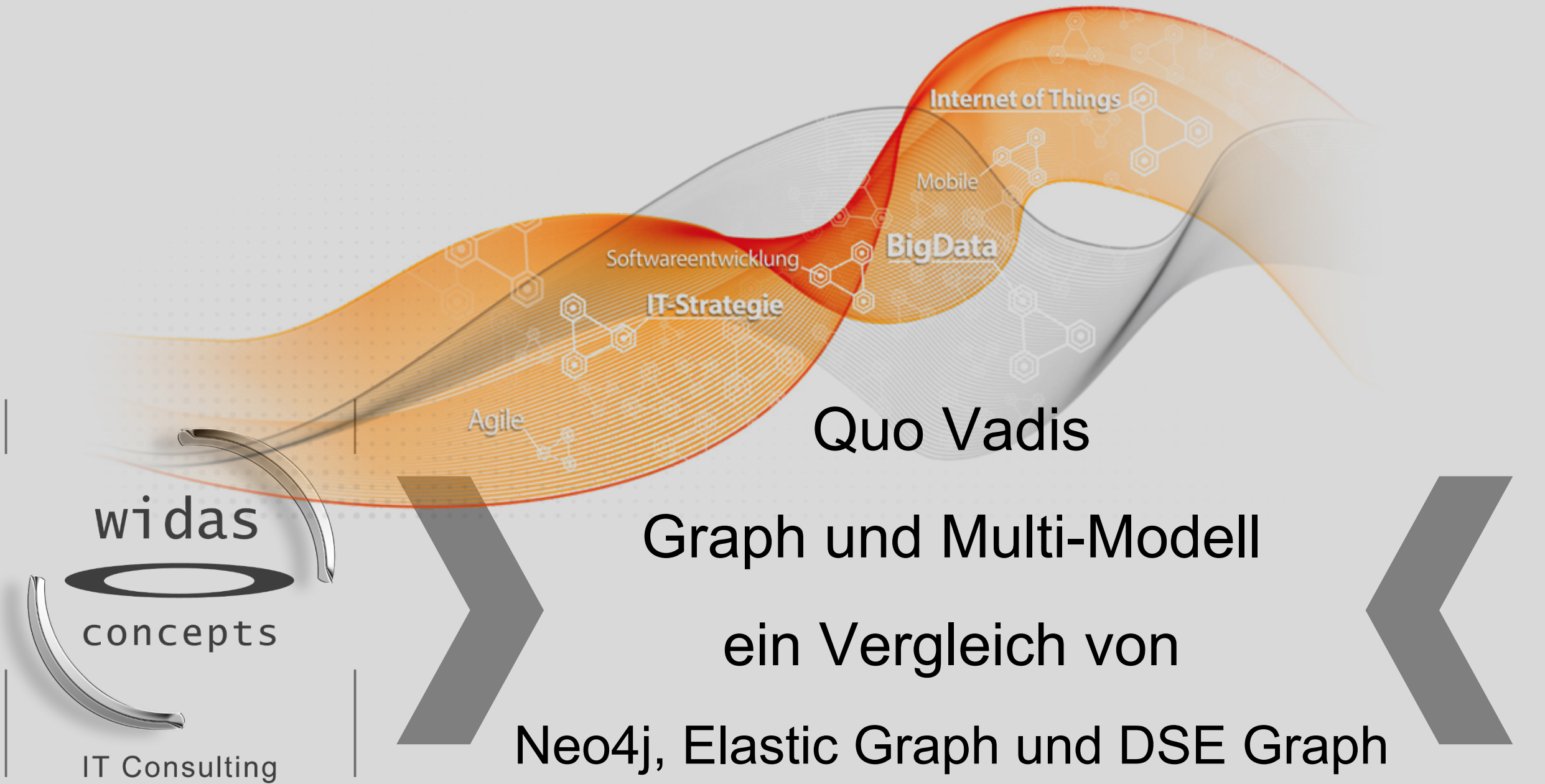




Quo Vadis

Graph und Multi-Modell

ein Vergleich von

Neo4j, Elastic Graph und DSE Graph

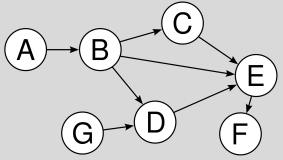
Business Analytics Day – 2017 – 08ter März

Zur Person



Thomas Mann ist Teamleiter und Solution Architect bei WidasConcepts im Bereich Big Data und Data Science. Sein Fokus ist die Konzeption, Architektur und Implementierung von Big Data-Lösungen unter Berücksichtigung moderner Data Science-Aspekte. Aktueller Tätigkeits- und Projektschwerpunkt liegt im Bankensektor bei Projekten im Bereich Data Lake und Betrugserkennung.

Agenda



Graph in a Nutshell



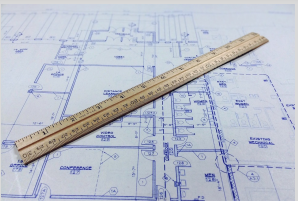
Neo4J



Elastic Graph

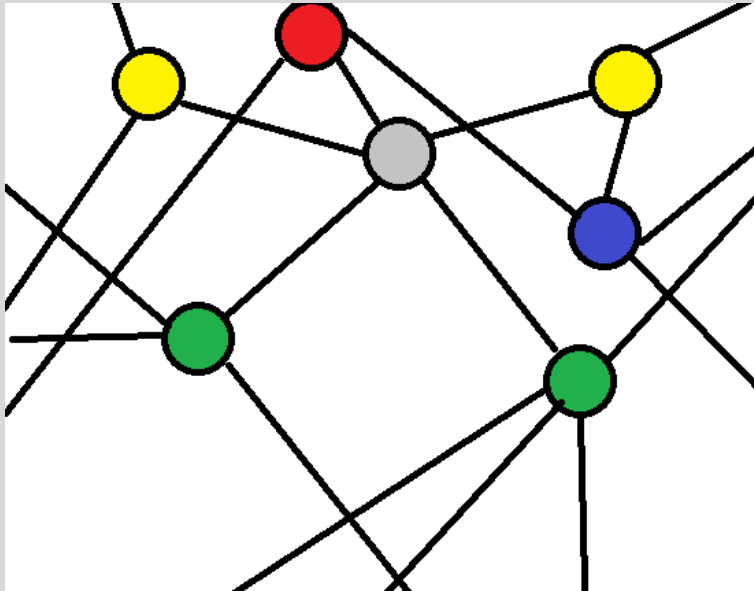


DSE Graph (Cassandra)



Analyse und Vergleich

Graph in a Nutshell



$$G = (V, E)$$

Knoten (Vertex) – Entitäten

Namen, Properties, Typ, Typhierarchie

Kanten (Edge) – Beziehungen

Namen, Properties, Typ, Typhierarchie

S-P-O – Subjekt Prädikat Objekt

Richtung Kante – gerichtet, binär

#Beziehungspartner – binär, tertiär, n

Graph in a Nutshell

Knoten (Vertex) – Entitäten



Kanten (Edge) – Beziehungen



S-P-O - Subjekt Prädikat Objekt

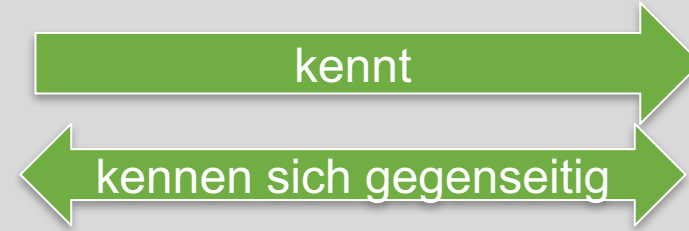


Graph in a Nutshell – Kanten (Vertex)

Richtung

gerichtet

bi-directional

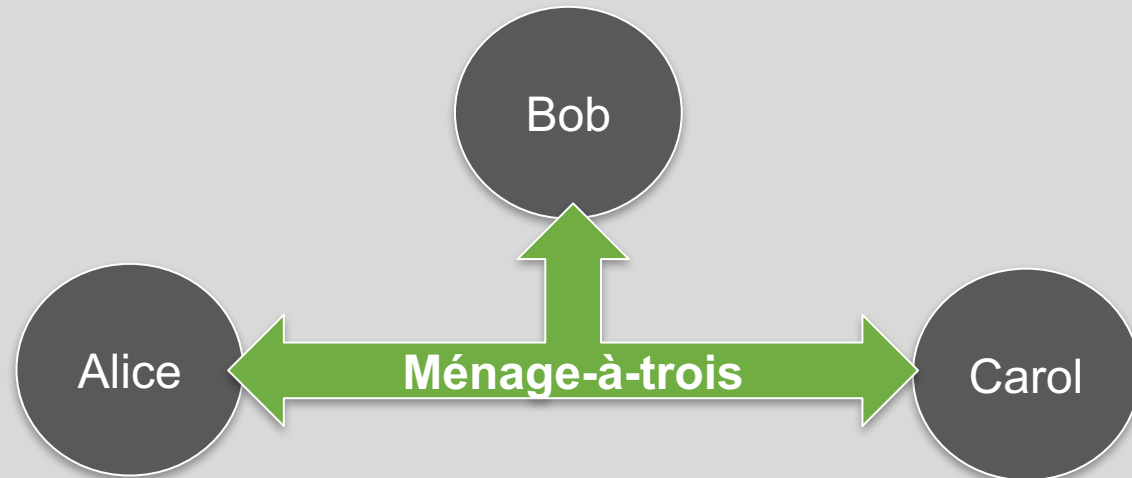


Anzahl Verknüpfungen

binär



tertiär/...



Graph in a Nutshell – Knoten (Vertex)

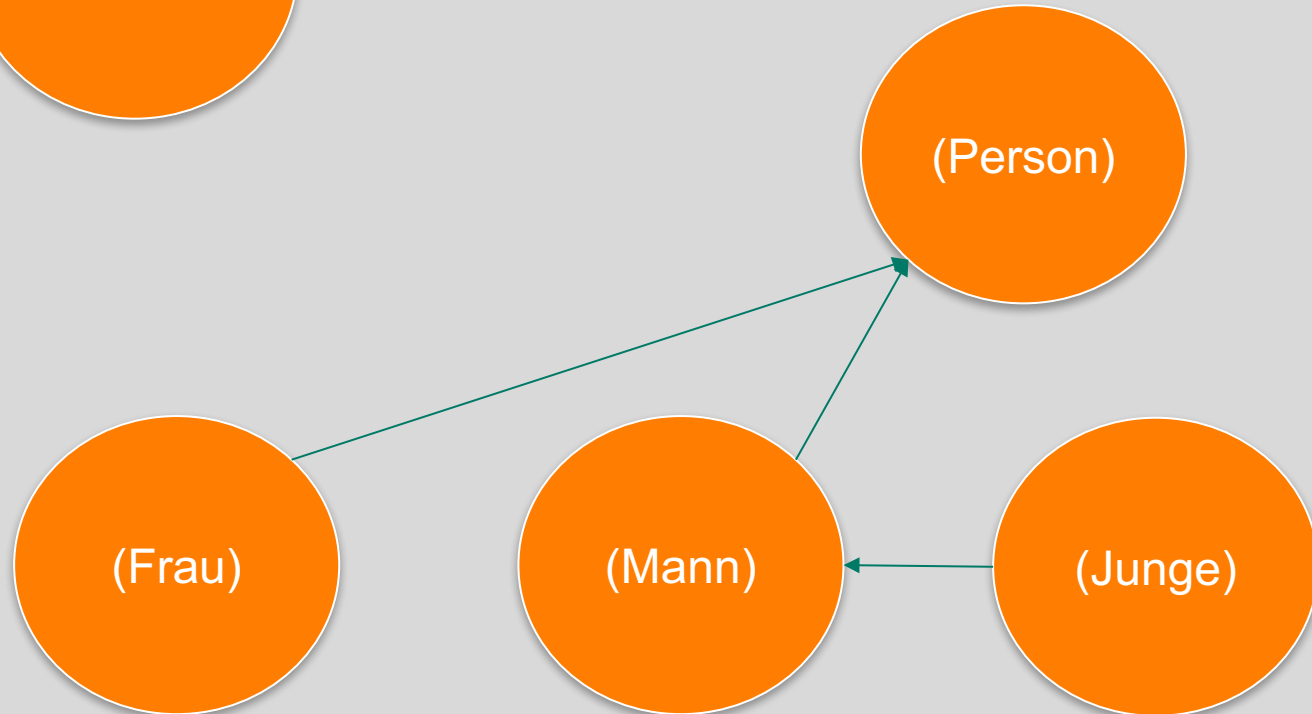
Knoten-Namen (Labels)



Knoten-Typen



Typenvererbung



Graph in a Nutshell – Kanten (Edge)

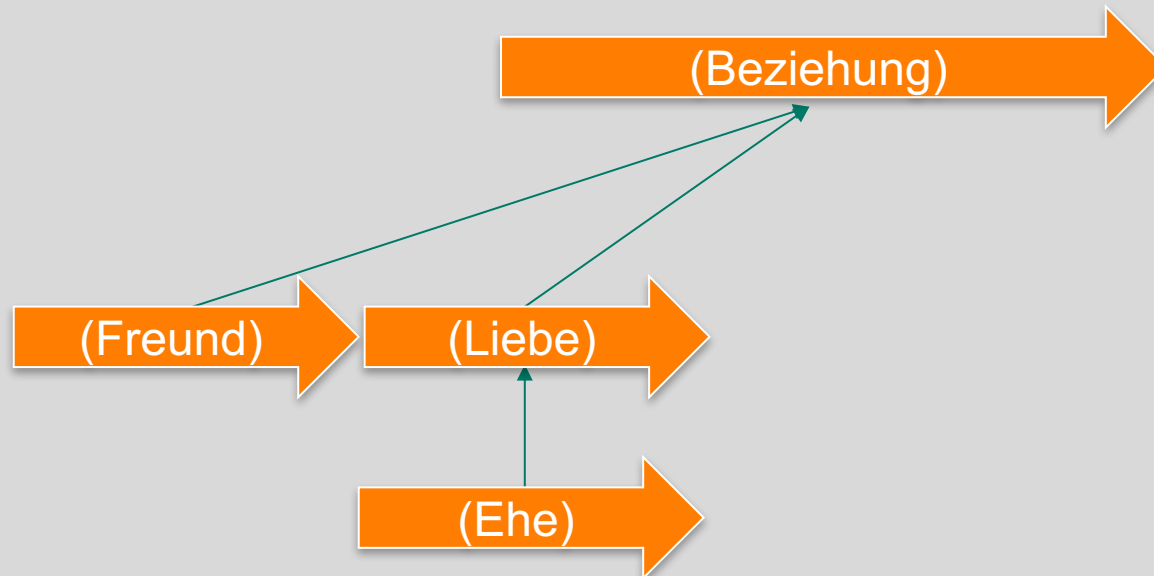
Kanten-Namen (Labels)



Kanten-Typen

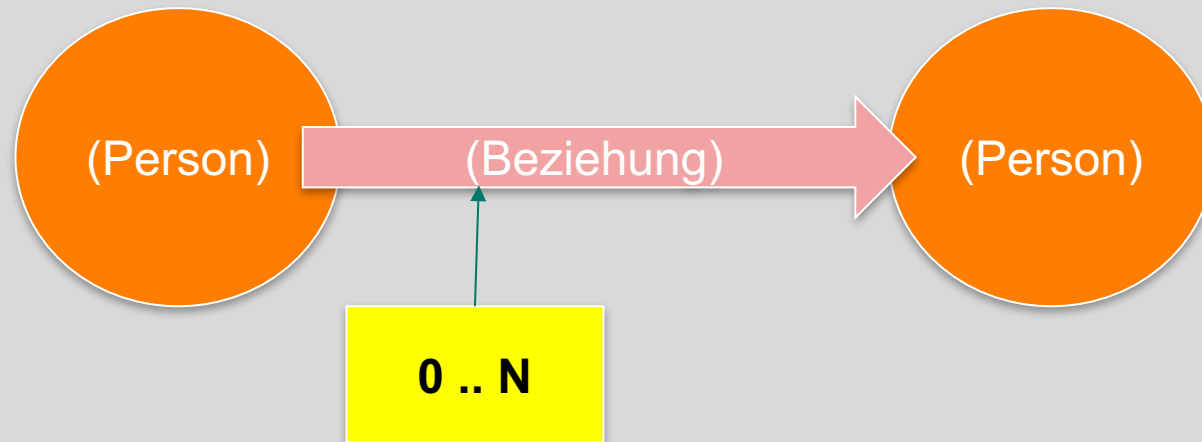
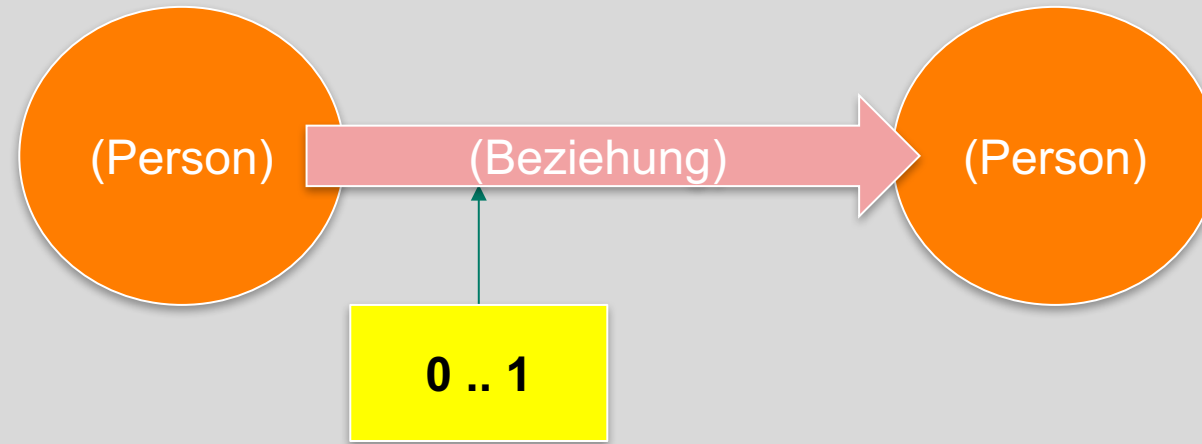


Typenvererbung



Graph in a Nutshell – Kanten (Edge)

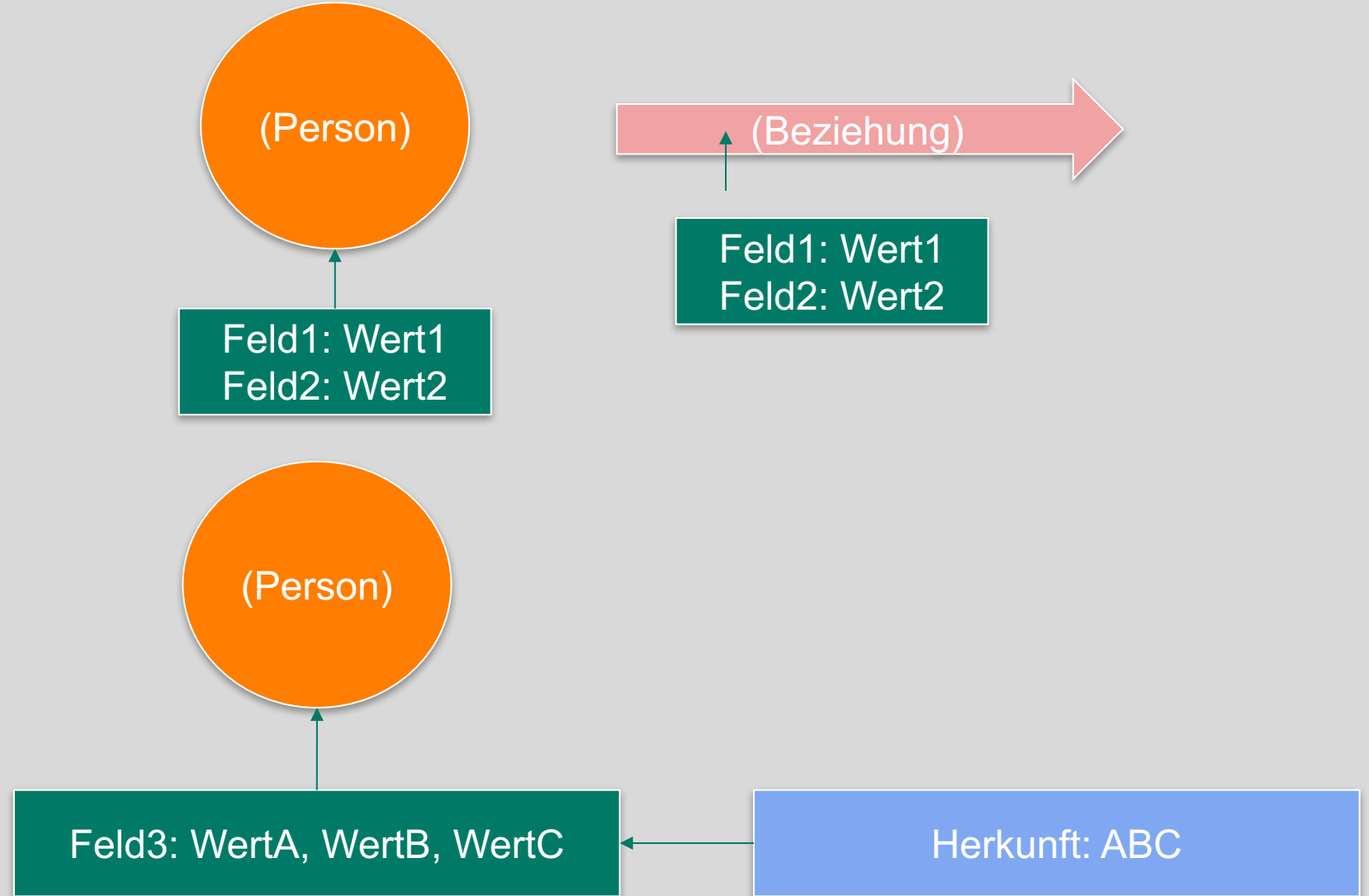
Kardinalität Kanten



Graph in a Nutshell – Eigenschaften (Properties)

Properties
für Knoten und Kanten

Multi-Properties
Meta-Properties



Graph in a Nutshell – Modellmächtigkeit

Ein Graph kann viele klassische Daten-Modelle



Key-Value



Columnar

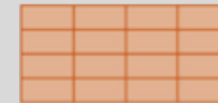


Hierarchisch



Dokumentenorientiert

Relational



abbilden/nachbilden

Graph in a Nutshell – übliches Graph Modell

Labeled Property Graph Data Model

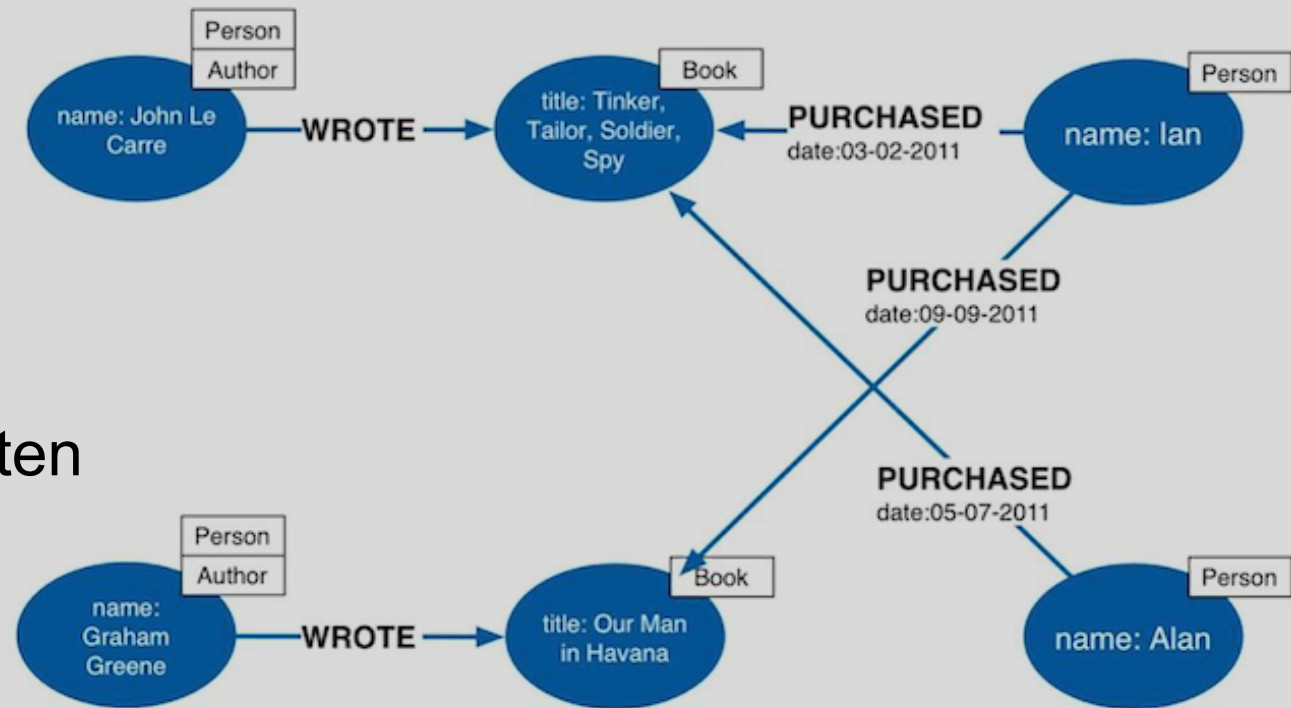
gerichtete Kanten

binäre Kanten

mit einfachen Properties

mit Labels (Namen)

Typen + Vererbung Knoten/Kanten



Graph in a Nutshell – Algorithmen

Breitensuche

Tiefensuche

Pfadlängen

kürzester Pfad

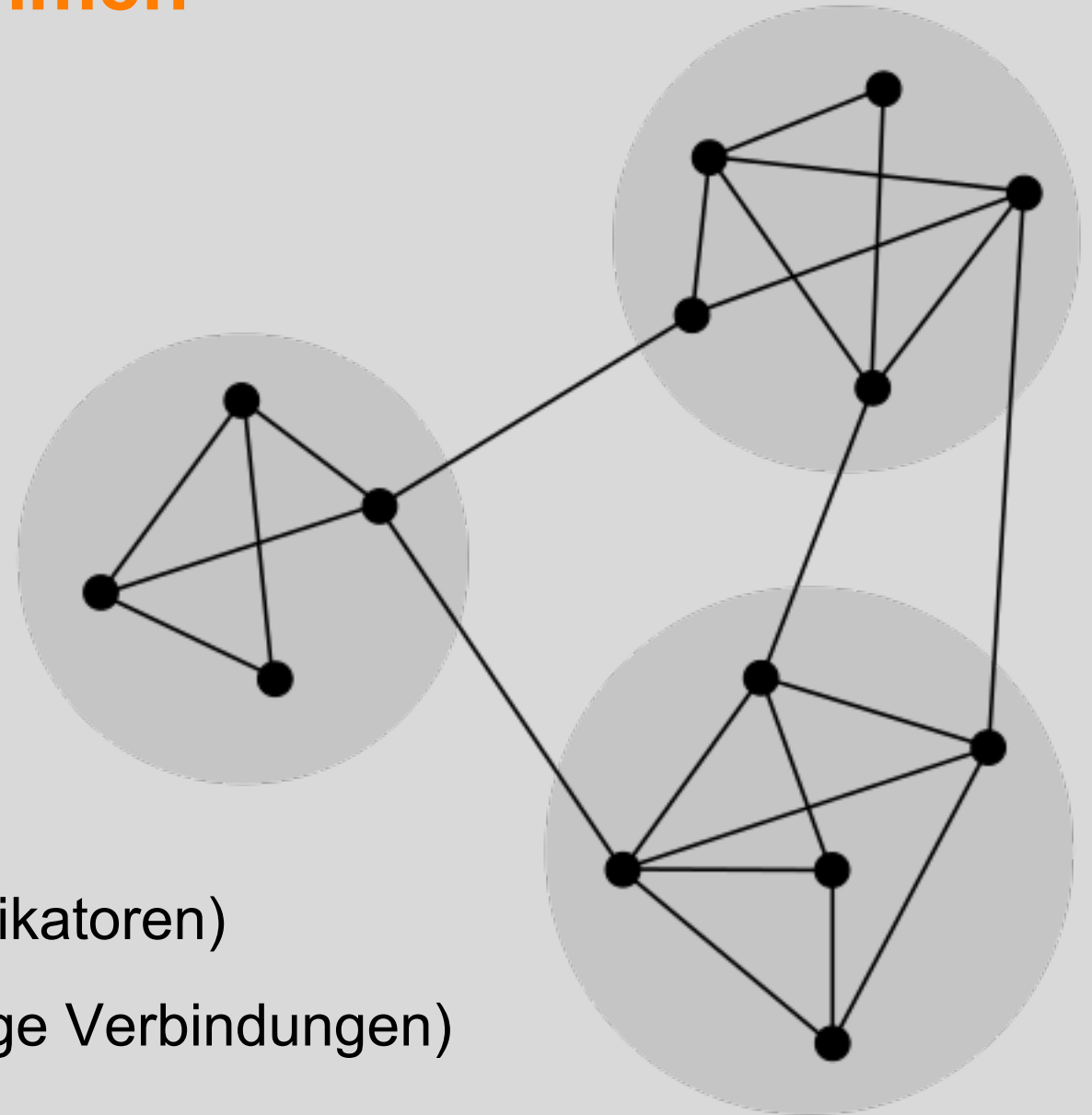
Zyklen-Suche

Community-Suche

Clique-Suche

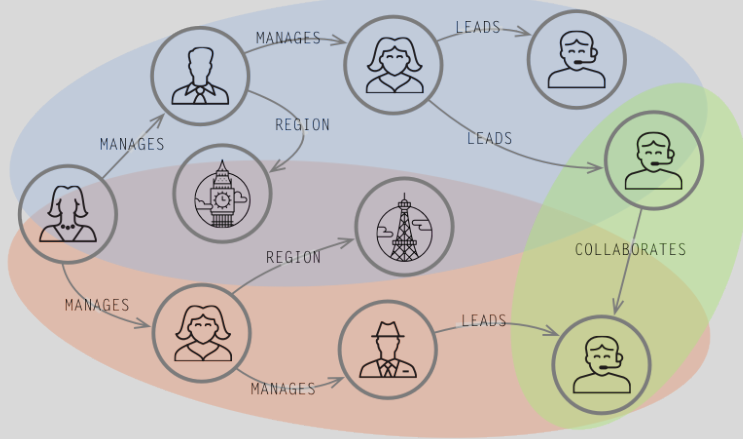
Suche nach hohen Kardinalitäten (Multiplikatoren)

Suche nach “schwächsten” Stellen (einzige Verbindungen)

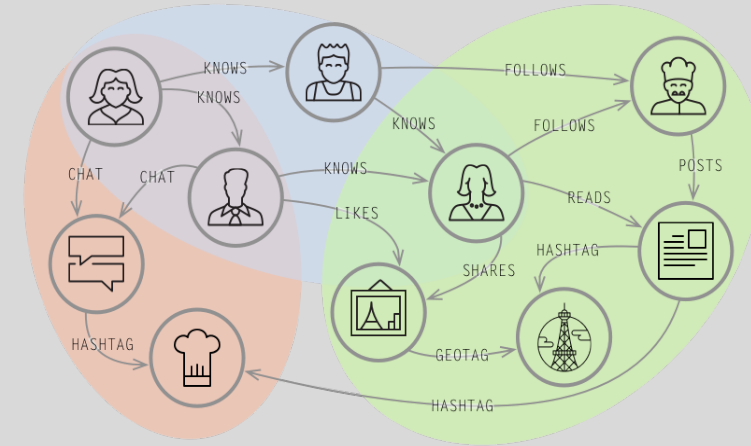


Graph in a Nutshell – Use-Cases

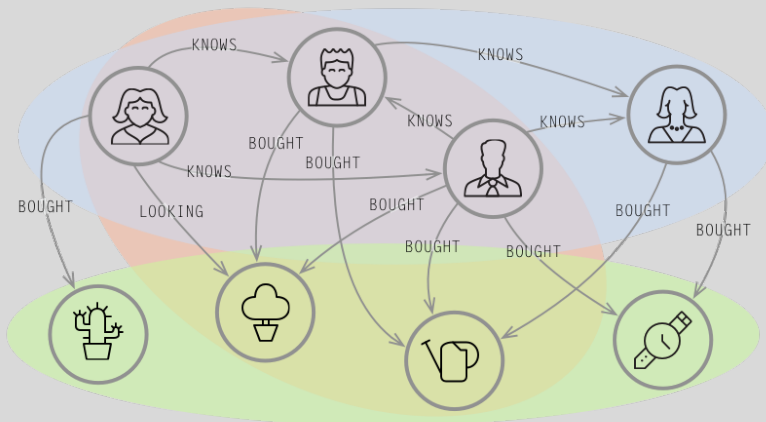
Stammdatenmanagement



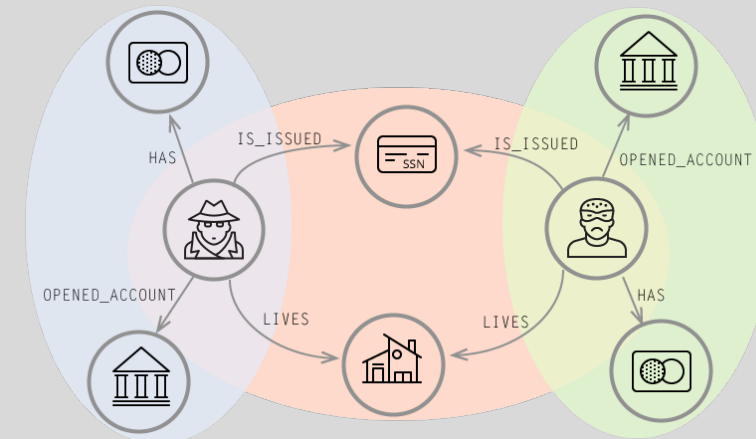
Social Graph



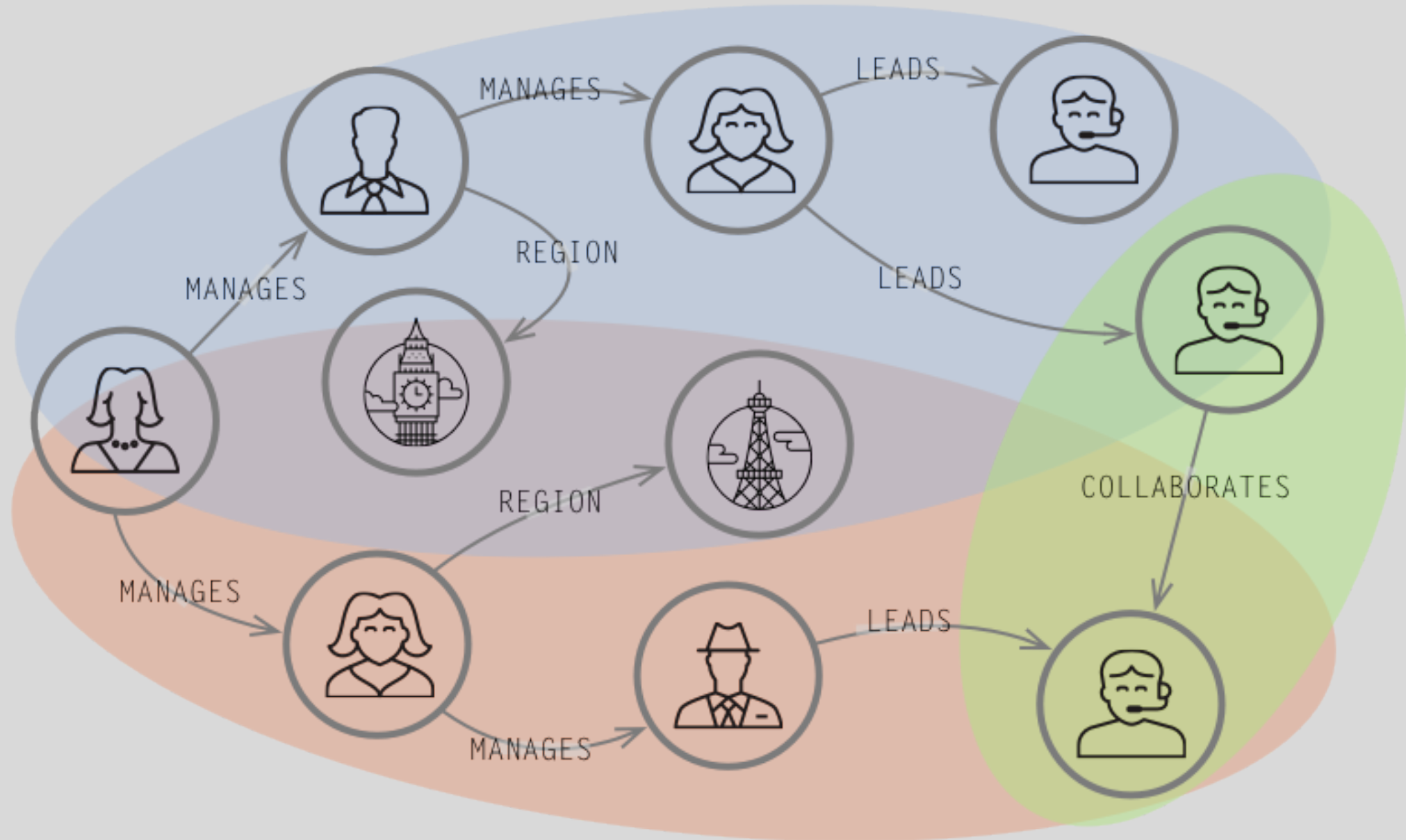
Recommendation Systeme



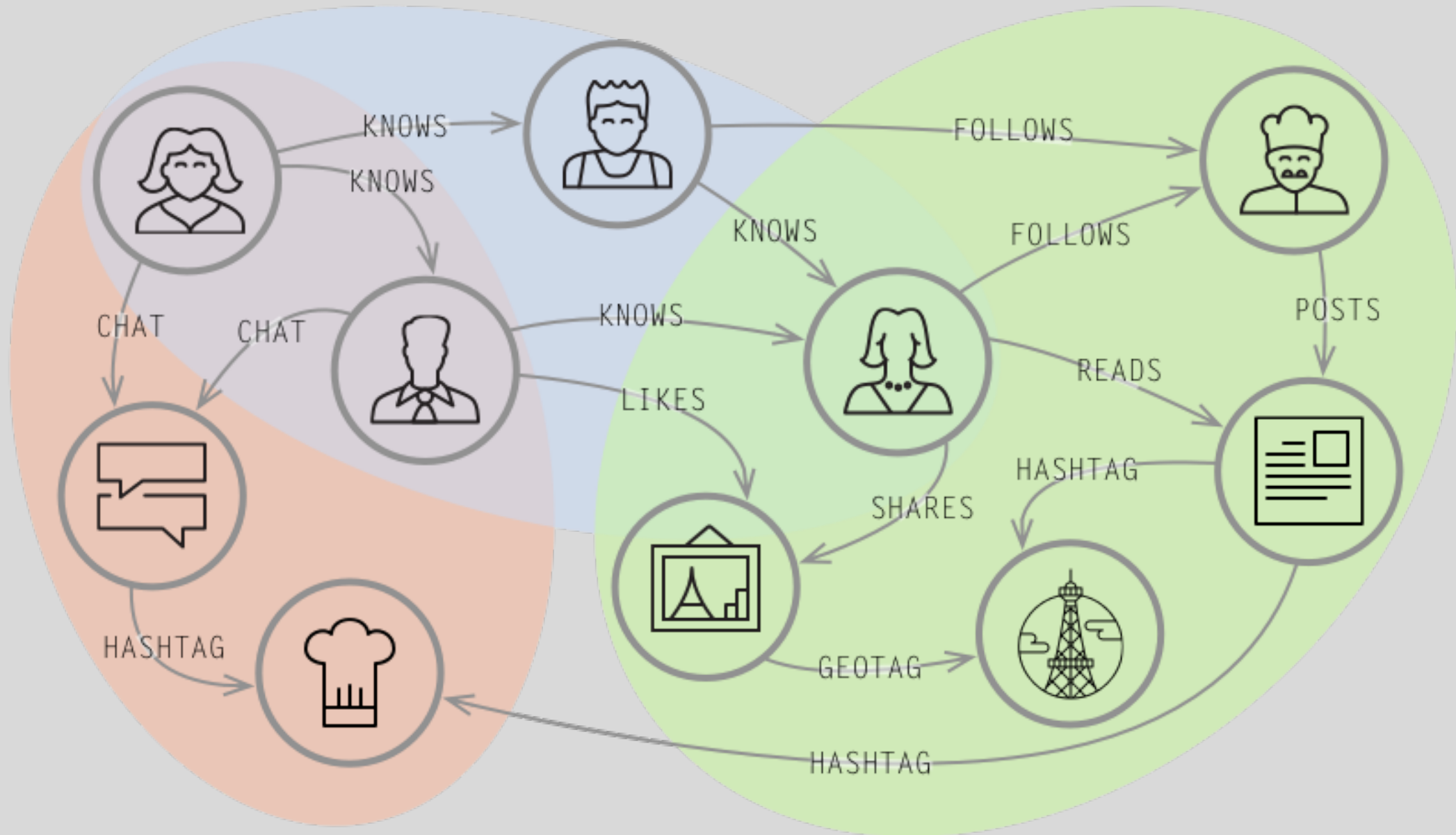
Betrugsfall-Analyse



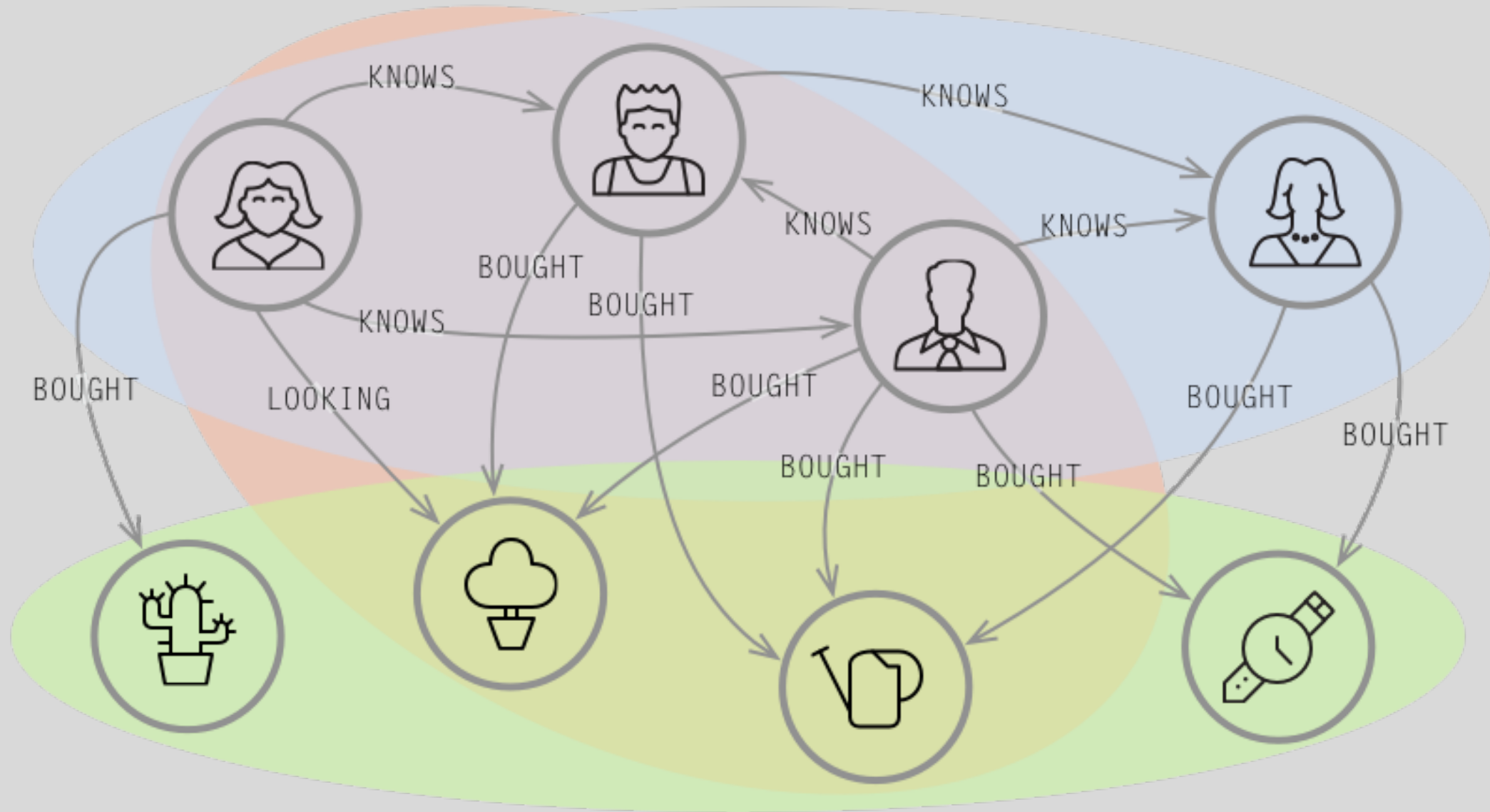
Graph in a Nutshell – Use-Cases – Stammdatenmanagement



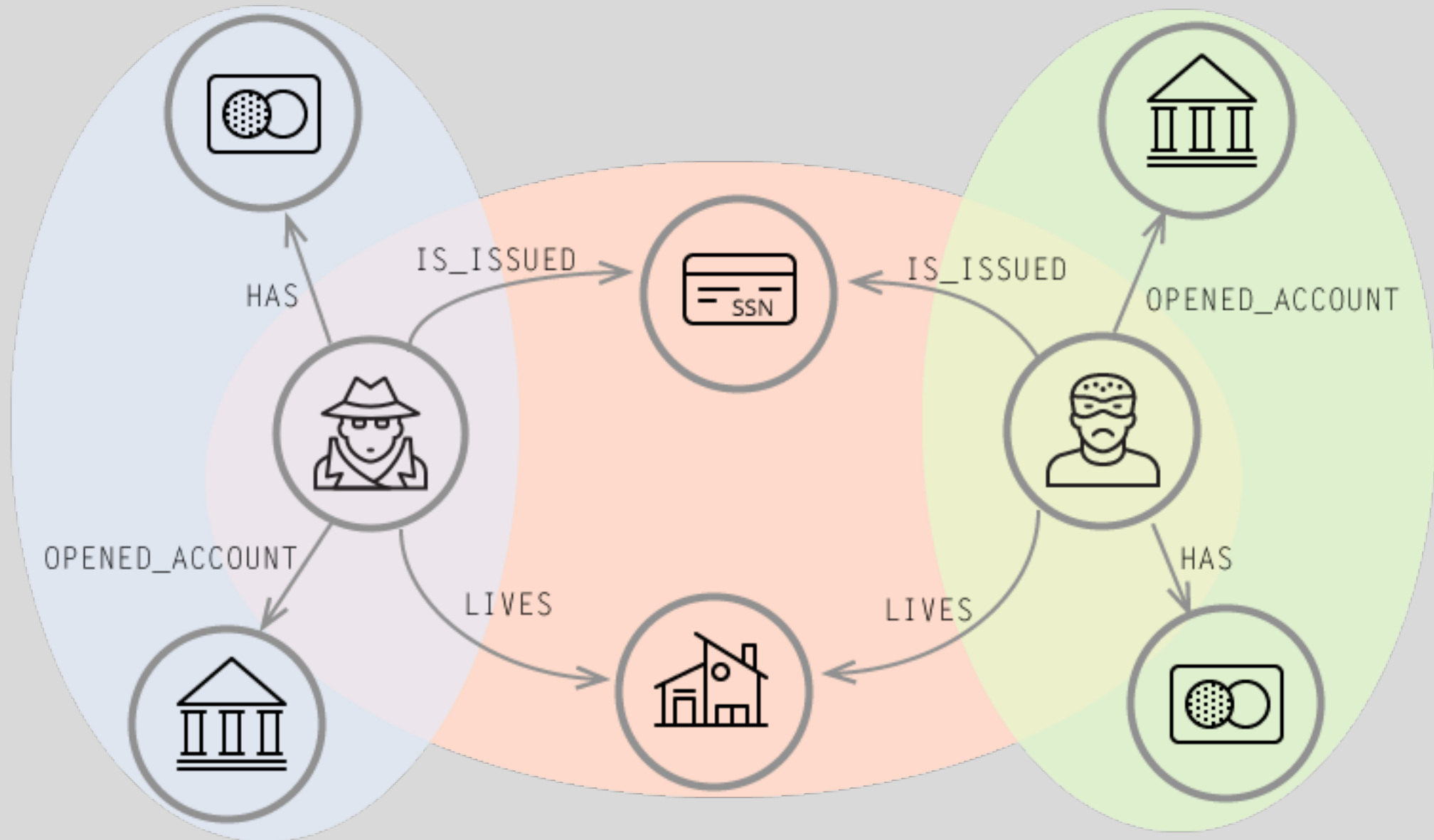
Graph in a Nutshell – Use-Cases – Sociale Graphanalyse



Graph in a Nutshell – Use-Cases – Recommendation



Graph in a Nutshell – Use-Cases – Betrugsfallanalyse



Graph in a Nutshell – Algorithmen – Herausforderungen

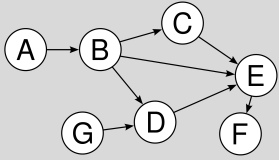
Superconnected Entities

Long-Tail

Adjacency



Agenda



Graph in a Nutshell



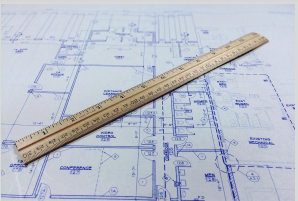
Neo4J



Elastic Graph



DSE Graph (Cassandra)



Analyse und Vergleich

Neo4J – Technologische Grundlagen & Konzepte



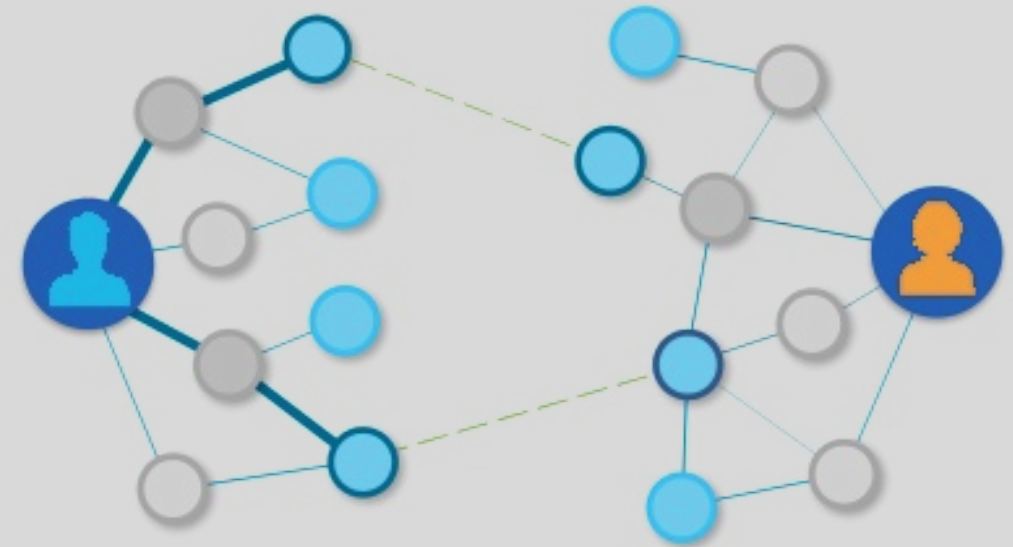
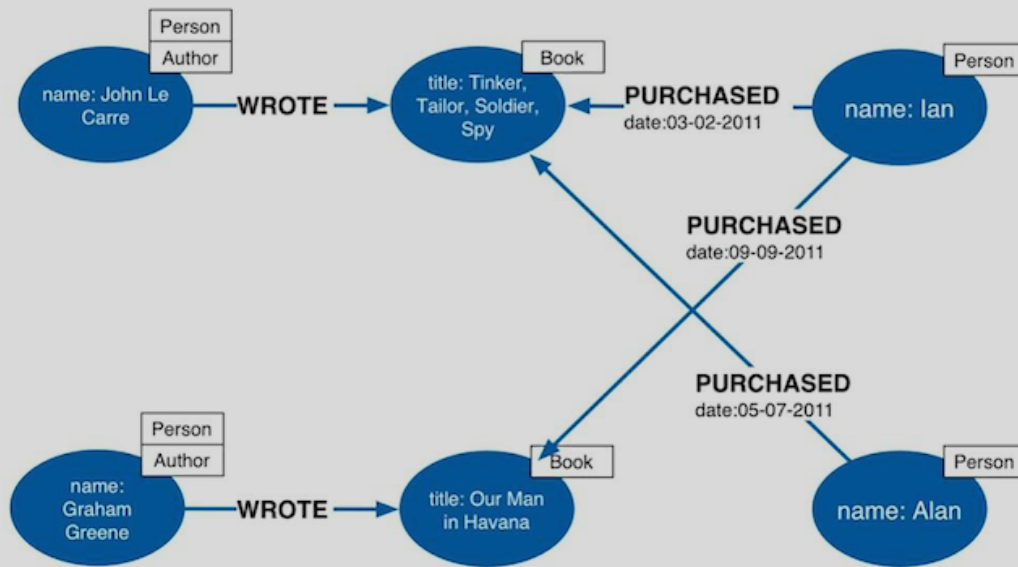
- 1** Labeled Property Graph Model
- 2** Indexfree Adjacency – Storage Model
- 3** ACID-Fähigkeit, Unlimited Graph “Claim”
- 4** Lucene für Suchen, Cypher (SQL) & Gremlin (DSL)
- 5** Master/Slave Sharding, Cache Sharding für Reads
- 6** Triggers, Stored Procedures, Cypher Query Optimizer

Neo4J – Technologische Grundlagen & Konzepte

Labeled Property Graph Model

Indexfree Adjacency – Storage Model

Labeled Property Graph Data Model



Neo4J – Technologische Grundlagen & Konzepte

ACID – Transaktionalität

Verwendung von Lucene

z.B. zum Indizieren von Properties

Natives Graph Modell

Natives Graph Processing

„Unlimited Graph“ Claim

Neo4J – Features

Cypher - Eigene Graphenorientierte Abfrage Sprache – SQL Ähnlich

```
MATCH (actor:Person)-[:ACTED IN]->(movie:Movie)
WHERE movie.title STARTS WITH "T"
RETURN movie.title AS title, collect(actor.name) AS cast
ORDER BY title ASC LIMIT 10;
```

Gremlin Unterstützung (Graph DSL)

```
g.V().out('knows').values('name')
g.V().out('knows').map{it.get().value('name') + ' is the friend name'}
g.V().out('knows').sideEffect(System.out.&println)
g.V().as('person').out('knows').as('friend').select().by{it.value('name').length()}
```

Neo4J – Features

Stored Procedures, Triggers (Java, DB-Built-In)

Query Optimizer für Cypher

Master/Slave Sharding Konzept

ACID-Transaktionalität

Cache Sharding

für Skalierung bei Reads

Domain Specific Sharding

„massive“ Skalierung?

Neo4J on IBM Power8 - 56TB (InMemory)

Scale Up ? Or Scale Out

Quelle: Neo4J Webseite



Neo4J – Beispiel Panama Papers - Process

Vorarbeiten Scannen, OCR, Metadaten

Graph-Modellierung

Knoten, Kanten an Beispieldaten ableiten
Properties und Metadaten ableiten
Dokument-Quellen für Knoten, Kanten
und Properties definieren

Analyse

Analyzer, Rules, Parser definieren
Automatisches Parsing, anreichern der Dokumente, Ableiten der Beziehungen
Graphalgorithmen: Similarities, Transitive Hülle, Mustererkennung
Visualisierungen und manuelle Graphtraversierung mit Algorithmen

Zusammensetzung der veröffentlichten Dateien	
Dateityp	# Dokumente
E-Mails	4.804.618
Datenbankformate	3.047.306
PDFs	2.154.264
Bilder	1.117.026
Textdokumente	329.166
Sonstige	2.242

Neo4J – Beispiel Panama Papers - Process

Vorarbeiten

Acquire documents

Classify documents

Scan / OCR

Extract document metadata

Graphanalyse

Whiteboard domain

Determine entities and their relationships

Determine potential entity and relationship properties

Determine sources for those entities and their properties

Work out analyzers, rules, parsers and named entity recognition for documents

Parse and store document metadata and document and entity relationships

Parse by author, named entities, dates, sources and classification

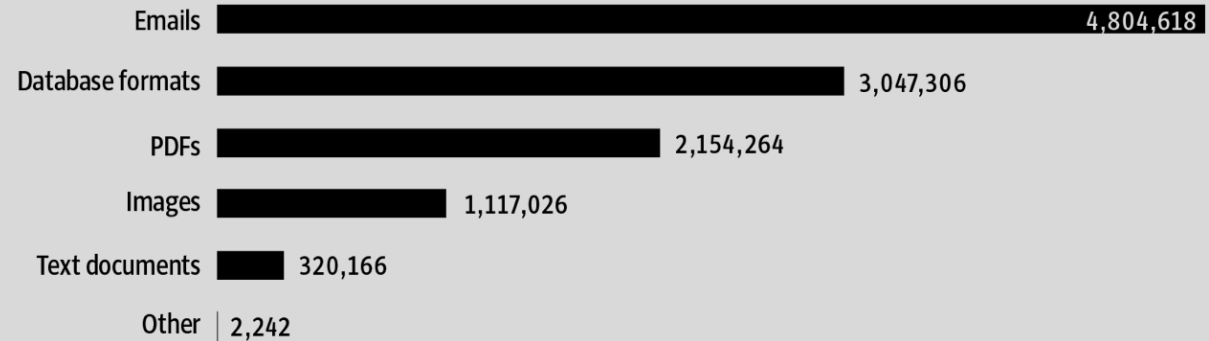
Infer entity relationships

Compute similarities, transitive cover and triangles

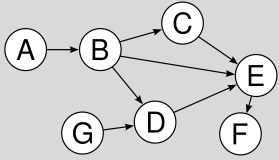
Analyze data using graph queries and visualizations

The structure of the leak

The 11,5 millionen contain the following file types



Agenda



Graph in a Nutshell



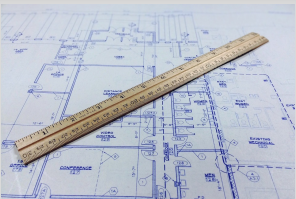
Neo4J



Elastic Graph



DSE Graph (Cassandra)



Analyse und Vergleich

Elastic Graph – Technologische Grundlagen & Konzepte



- 1** Dokumentenorientierte Datenbank JSON
- 2** “alles” ist indiziert, mit Lucene, verteilte Shards
- 3** Kein Graphmodell, indirekte Graphanalyse
- 4** Visueller Graphbrowser
- 5** Beziehungen werden indirekt abgeleitet
- 6** Erweiterung der Query-Syntax um Graphaspekte

Elastic Graph – Technologische Grundlagen & Konzepte

Dokumenten-orientiert JSON

Volltextsuche (Lucene) - “alles“ ist indiziert

Verteilung: Sharding/Replikation der Lucence Indexe

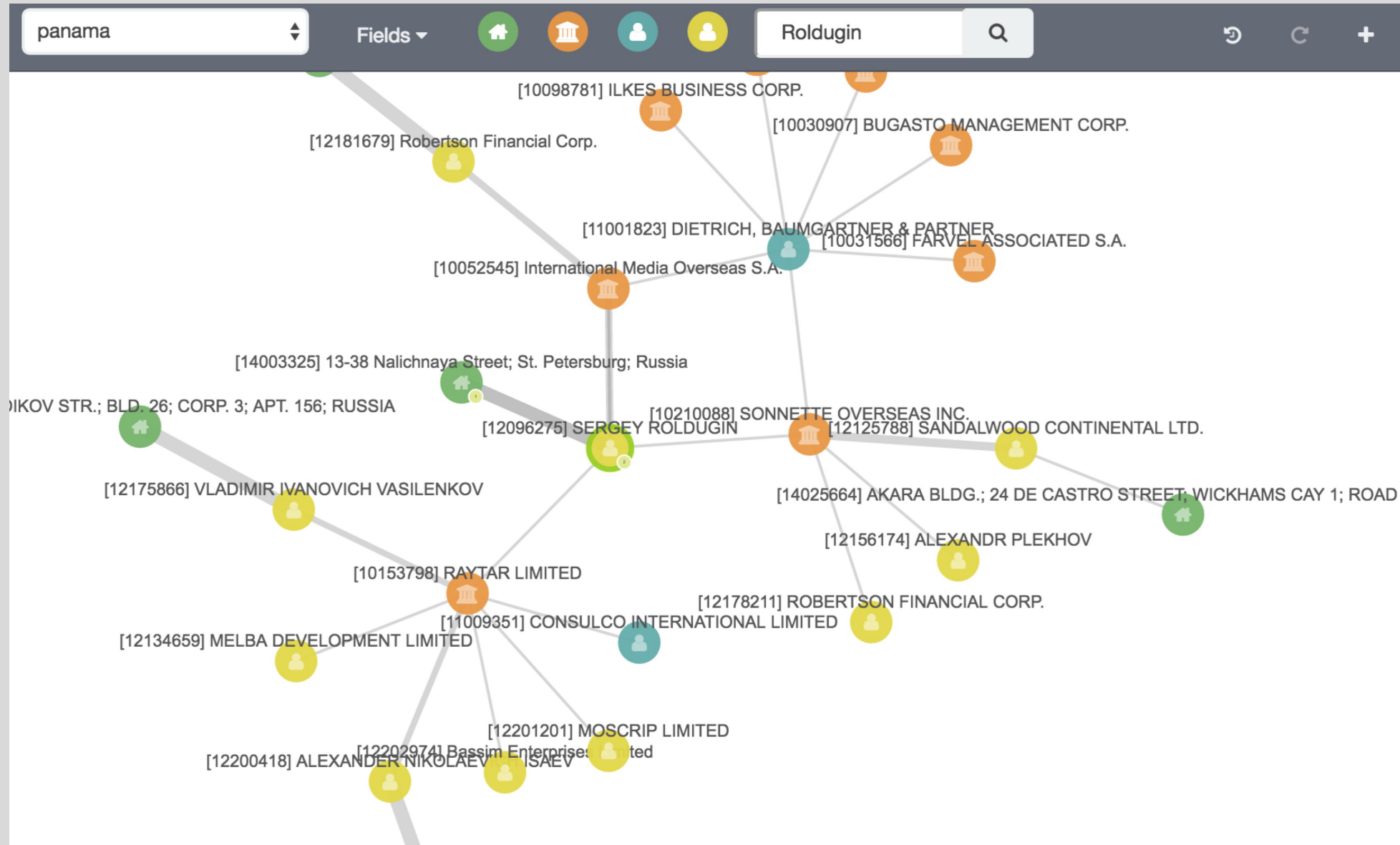
Graph ?

„Graph“-Analyse auf vorhanden Dokumenten

Kanten/Beziehungen werden indirekt aus den Dokumenten abgeleitet

Interaktive Oberfläche

Elastic Graph – GUI – Suche auf einem Index



Elastic Graph – GUI – Knoten gruppieren

[14003325] 13-38 Nalichnaya Street; St. Petersburg; Russia

[10153798] RAYTAR LIMITED

[12180773] SERGEY PAVLOVICH ROLDUGIN

[10210088] SONNETTE OVERSEAS INC

[10052545] International Media Overseas S.A.

Selections

all

none

invert

linked

 [14003325] 13-38 Nalichnaya Street; St. Petersburg; Russia (+1)

 address.id.raw [14003325] 13-38 Nalichnaya Street; St. Petersburg; Russia

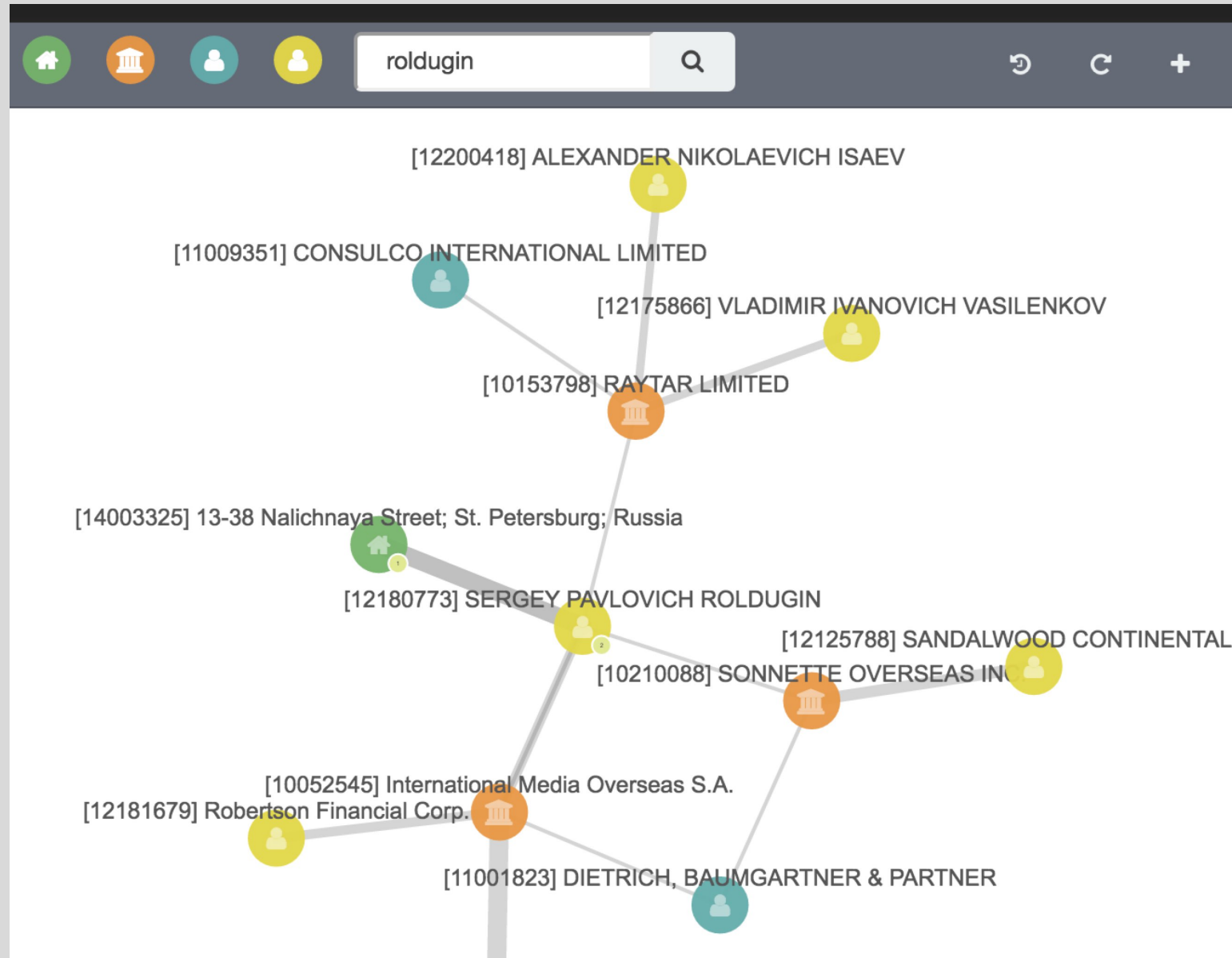
 ungroup

Display label

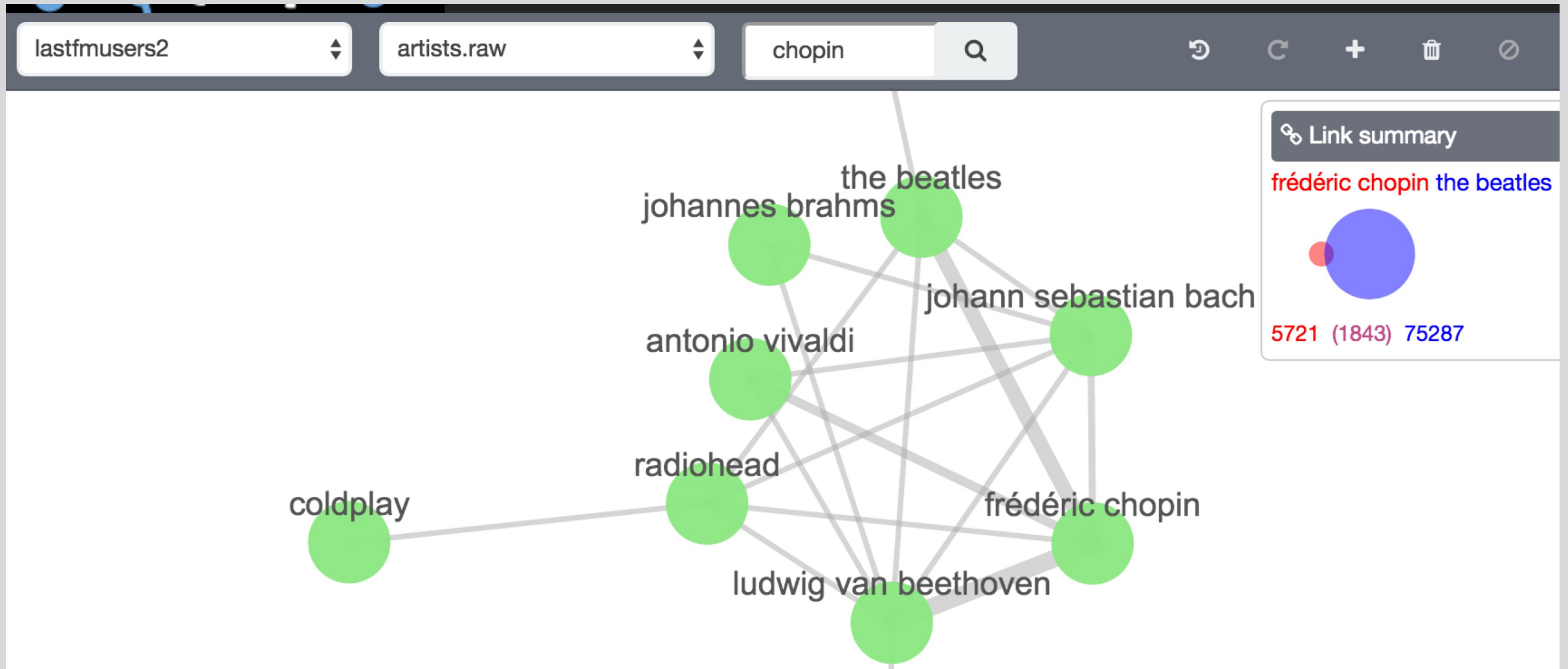
[14003325] 13-38 N

Change the label for this vertex

Elastic Graph – GUI – Raus-Zoomen (Spidering Out)



Elastic Graph – GUI – Relevanz 1



Elastic Graph – Technologische Grundlagen & Konzepte

Was ist eine Beziehung (Kante) ?

Was ist relevant?

Popularität ist es nicht

Anzahl Beziehungen

Sampling

mindest-Anzahl Beziehungen

TF-IDF

Elastic Graph – GUI – Relevanz 2

lastfmusers2

artists.raw

chopin

Q

↶

↷

+

🗑

🔒

🔗

⚙

Last requestBlacklistAdvanced settings

Sample size

2000

Terms are identified from samples of the most relevant documents. Bigger is not necessarily better - can be slower and less relevant.

☒ Significant links

Identify terms that are "significant" rather than simply popular

Certainty

3

The min number of documents that are required as evidence before introducing a related term

Diversity field

[No diversification]

To avoid document samples being dominated by a single voice, pick the field that helps identify the source of bias. *This must be a single-term field or searches will be rejected with an error*

Timeout (ms)

5000

Max time in milliseconds a request can run

^

felix mendelssohn

🔗 Link summary

Elastic Graph – Technologische Grundlagen & Konzepte

Keine eigene Graph-Sprache

Erweiterung der aktuellen Such-API um Graph Aspekte

dynamisches „Graph-Schema“ – Volltextsuche mit Graph-Aspekten

POST clicklogs/_graph/explore

```
{
  "query": {
    "match": {
      "query.raw": "midi"
    }
  },
  "vertices": [
    {
      "field": "product"
    }
  ],
  "connections": {
    "vertices": [
      {
        "field": "query.raw"
      }
    ]
  }
}
```

Elastic Graph – Beispiel Panama Papers - Prozess

Vorarbeiten Scannen, OCR, Metadaten

Graph/Index Erstellung

Alle Daten in Elastic indizieren

explorative Graph-Modellierung (wiederholend)

Explorative Analyse nach Beziehungen

Welche Felder welcher Dokumenttypen stehen in Verbindung

Scriptbasierte Neu-Erstellung neuer „verknüpfter“ Dokumente (in Compound Indexes)

Analyse

Suche nach z.B. Namen

Ausgabe des Verknüpfungsgraphen

Zusammensetzung der veröffentlichten Dateien	
Dateityp	# Dokumente
E-Mails	4.804.618
Datenbankformate	3.047.306
PDFs	2.154.264
Bilder	1.117.026
Textdokumente	329.166
Sonstige	2.242

Elastic Graph – Beispiel Panama Papers

Vorarbeiten

Acquire documents

Classify documents

Scan / OCR

Extract document metadata

Graphanalyse

Drop Documents to Elastic Graph

Modelling: Rinse & Repeat

Identifiy explorative relationship-types

which field in which index combined with other fields using specific comparison

Re-Create new documents from those relationship-types

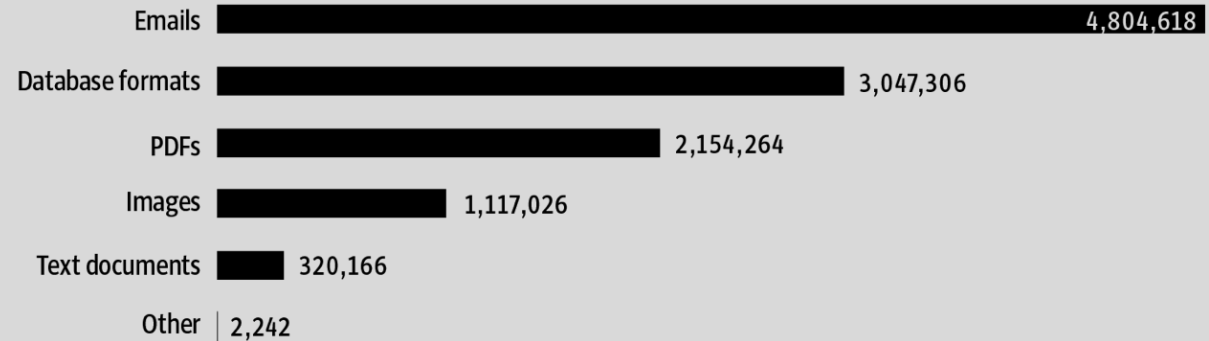
Create compound indexes (new combined fields) in new documents to link entites

Output

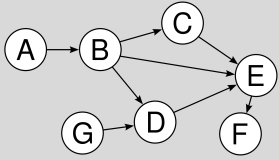
Search graph for specific persons and plot out their subgraph

The structure of the leak

The 11,5 millionen contain the following file types



Agenda



Graph in a Nutshell



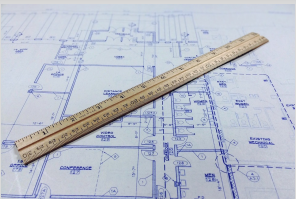
Neo4J



Elastic Graph



DSE Graph (Cassandra)



Analyse und Vergleich

DSE Graph – Technologische Grundlagen & Konzepte



1 Apache Cassandra, KVV/Columnar, Peer2Peer, Hashring

2 Multi-Meta Properties Labeled Graph Modell

3 Dynamische Partitionierung (Adjacency)

4 “Indexe” auf allem Knoten, Kanten, Properties

5 Materialized Views, Secondary Index, Search Index

6 Gremlin als DSL

DSE Graph – Technologische Grundlagen & Konzepte

Apache Cassandra

KKV-Store (Columnar)

Linear skalierend, Peer2Peer, Hashring



DSE Graph – Technologische Grundlagen & Konzepte

Graph?

Multi-Meta Properties Labeled Graph Modell

explizite S-P-O Angaben

dynamische (default) Partitionierung der Knoten-ID (Vertex)

“ähnlich“ Communities, aber kein aktiver Community-Algorithmus

keine Indexfree Adjacency

Custom Partitionierung möglich

DSE Graph – Technologische Grundlagen & Konzepte

ein Index – auf jedes Objekt, jeden Sachverhalt (Hashring Partitionierung)

Knoten (Vertexes)

Kanten (Edges)

Properties, Multi-Properties

Index-Typen:

Materialized View (High Cardinality)

Secondary Index (Low Cardinality)

Search Index (Suchbarkeit, Ähnlichkeitssuche)

DSE Graph – Technologische Grundlagen & Konzepte

Gremlin als Graph DSL

```
g.V().out('knows').values('name')  
g.V().out('knows').map{it.get().value('name') + ' is the friend name'}  
g.V().out('knows').sideEffect(System.out.&println)  
g.V().as('person').out('knows').as('friend').select().by{it.value('name').length()}
```

Schema Erzeugung

Datenablage

Traversierung

DSE Graph – Technologische Grundlagen & Konzepte

graphische Oberfläche (DSE Studio)



DSE Graph – Beispiel Panama Papers

Vorarbeiten

Acquire documents

Classify documents

Scan / OCR

Extract document metadata

Graphanalyse

Whiteboard domain

Determine entities and their relationships

Determine potential entity and relationship properties

Determine sources for those entities and their properties

Think more deep about a graph data model and possible indexes

Index entities, relationships, properties

Work out analyzers, rules, parsers and named entity recognition for documents

Think more deep about a graph data model and possible indexes

Index entities, relationships, properties

Parse and store document metadata and document and entity relationships

Parse by author, named entities, dates, sources and classification

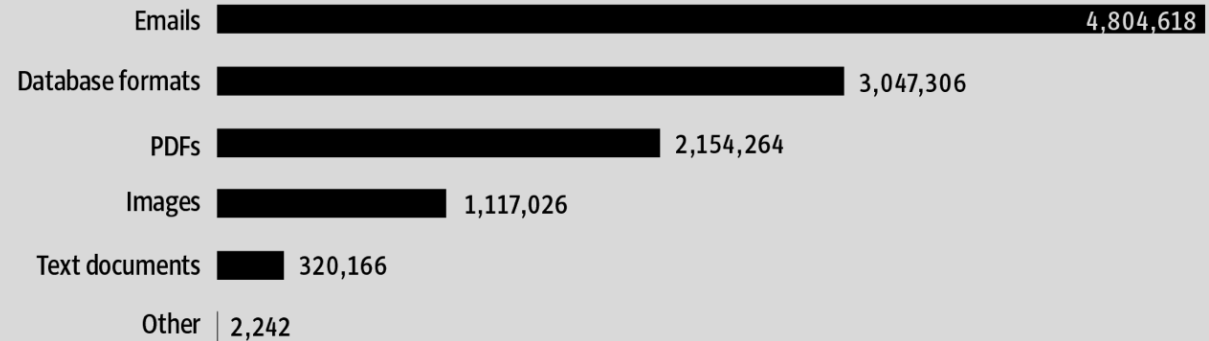
Infer entity relationships

Compute similarities, transitive cover and triangles

Analyze data using graph queries and visualizations

The structure of the leak

The 11,5 millionen contain the following file types



DSE Graph – Beispiel Panama Papers - Process

Vorarbeiten Scannen, OCR, Metadaten

Graph-Modellierung

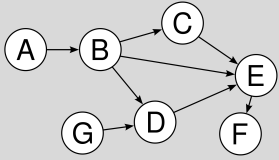
Knoten, Kanten an Beispieldaten ableiten
Properties und Metadaten ableiten
Dokument-Quellen für Knoten, Kanten
und Properties definieren
Customizing Index, Partitionierung
Wiederholende Analyse des Models

Analyse

Analyzer, Rules, Parser definieren
Automatisches Parsing, anreichern der Dokumente, Ableiten der Beziehungen
Graphalgorithmen: Similarities, Transitive Hülle, Mustererkennung
Visualisierungen und manuelle Graphtraversierung mit Algorithmen

Zusammensetzung der veröffentlichten Dateien	
Dateityp	# Dokumente
E-Mails	4.804.618
Datenbankformate	3.047.306
PDFs	2.154.264
Bilder	1.117.026
Textdokumente	329.166
Sonstige	2.242

Agenda



Graph in a Nutshell



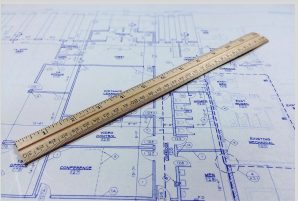
Neo4J



Elastic Graph



DSE Graph (Cassandra)



Analyse und Vergleich

Vergleich – Disclaimer

Verglichen werden NoSQL Datenbanken untereinander (keine RDBMS)

Elastic Graph und DSE Graph sind noch recht neu

Ergebnis: “Es kommt darauf an“ – Worauf genau?

Vergleich – Freiheit

Will man nur „Schemafreiheit“ bei den Properties?

Oder sind selbst die Beziehungen und Entitäten noch unbekannt?



Deklaration

Freiheit

Vergleich – Vorarbeiten in die Modellierung

Wieviel muss man investieren in den Modellierungsaufwand?

Ist das der Preis der Skalierbarkeit?



Modellierungsaufwand (hoch)

geringer

Vergleich – Gremlin (Graph DSL) vs. Cypher

Will man eine explizite Graphsprache

Oder eine SQL artige Abfragesprache

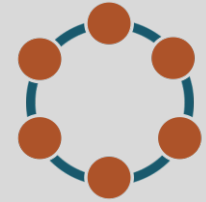


Gremlin

Cypher

Vergleich – Skalierbarkeit

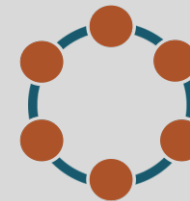
Volume



weniger

mehr

Connectedness



weniger

mehr

Vergleich – Skalierbarkeit

Read



weniger

mehr

Writes



weniger

mehr

Vergleich – Skalierbarkeit

Zugriffsgeschwindigkeiten/Latenzen



Near-Time

Real-Time

Umgehen mit historischen Daten (Aktiv vs. Inaktiv)



weniger berücksichtigt

berücksichtigt

Vergleich – Der Use-Case entscheidet

Neo4J



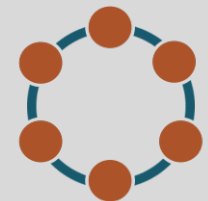
dedizierte explizite Graph Algorithmen auf nicht so vielen Daten

Elastic Graph



explorative Graphanalyse auf anfänglichen nicht “Graph“ modellierten Daten

DSE Graph



Graphalgorithmen auf massiven (verteilten) Daten

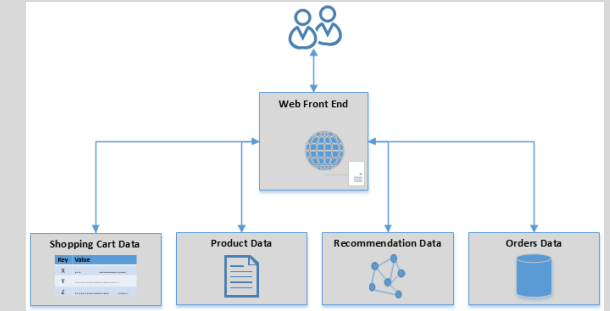
Was will der Kunde? Weiß er überhaupt schon dass er ein Graph Problem hat?

Vergleich – Die Kombination gewinnt

Polyglote Persistenz im Big Data Bereich

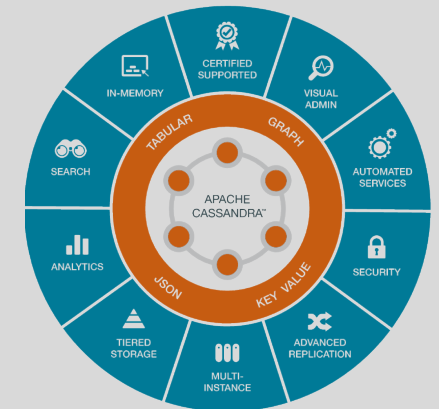
wir „speichern“ bestimmte Daten mehrmals

HDFS, MQ, DB1-NoSQL, DB2-Graph



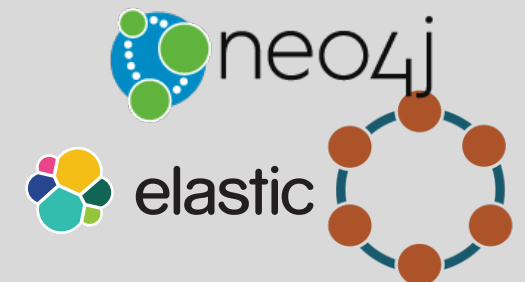
Multi-Modell Datenbanken

Wenn schon polyglote Persistenz wieso nicht mit der selben Technologie?



Operationale vs. Analytische Datenbanken

Welche der Datenbanken kann man tatsächlich operational nutzen, welche wäre ein analytisches Addendum?





HighEnd-Technology requires HighEnd-Competence

WidasConcepts

Thomas Mann, Solution Architect

Mobile: +49 162 259 56 90

Mail: thomas.mann@widas.de

Masterarbeit / Studienarbeit

WidasConcepts GmbH

Maybachstraße 2
71299 Wimsheim

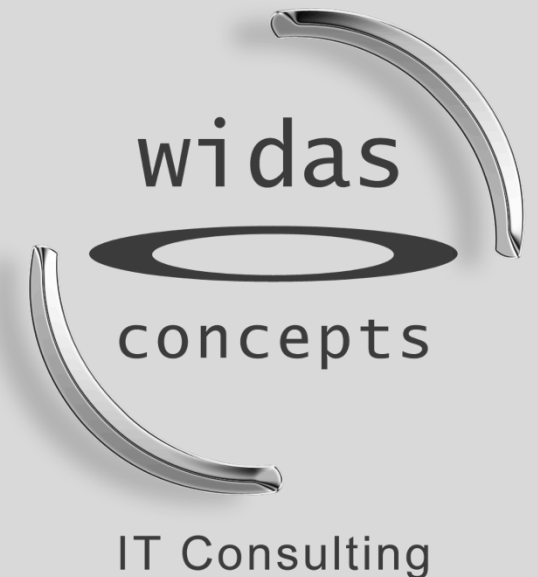
www.widas.de



Big Data Consultant
IoT Consultant
Java Developer

Systemintegrator

Solution Architekt



Dieses Dokument wurde von WidasConcepts erstellt. Die Verteilung, Zitierung und Vervielfältigung – auch auszugsweise zum Zwecke der Weitergabe an Dritte ist nur mit vorheriger schriftlicher Zustimmung von WidasConcepts gestattet.

This presentation was created by WidasConcepts. Distribution, citation, copying - completely or in extracts – for transfer purposes, is only permitted with prior written agreement. These abstracts and graphics were deployed by WidasConcepts within the scope of a presentation. It is no complete documentation of this event.