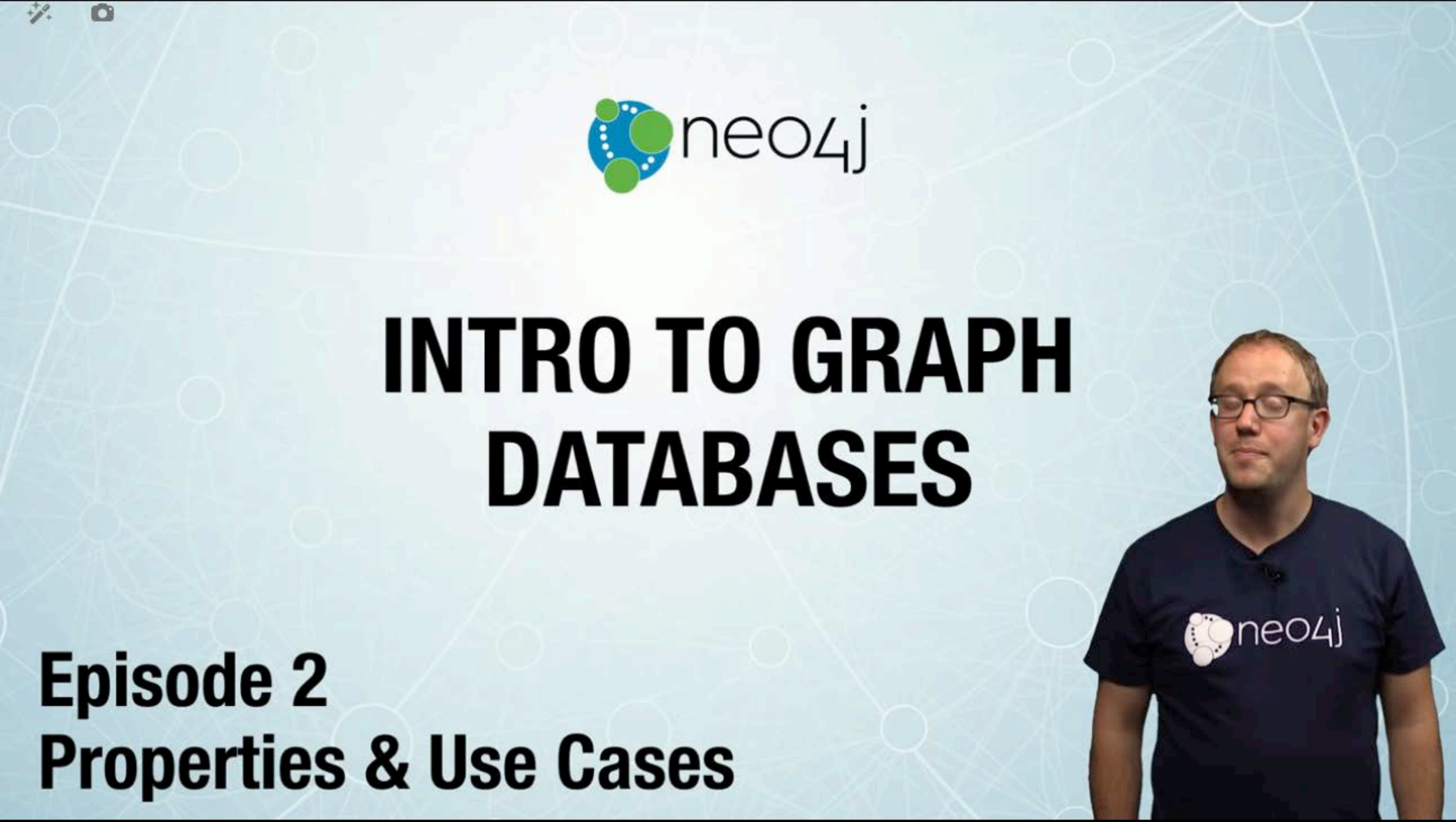

neo4j
INTRO TO GRAPH DATABASES
neo4j
Episode 2
Properties & Use Cases

# Properties of Graph Databases

# Intuitivness

# Speed

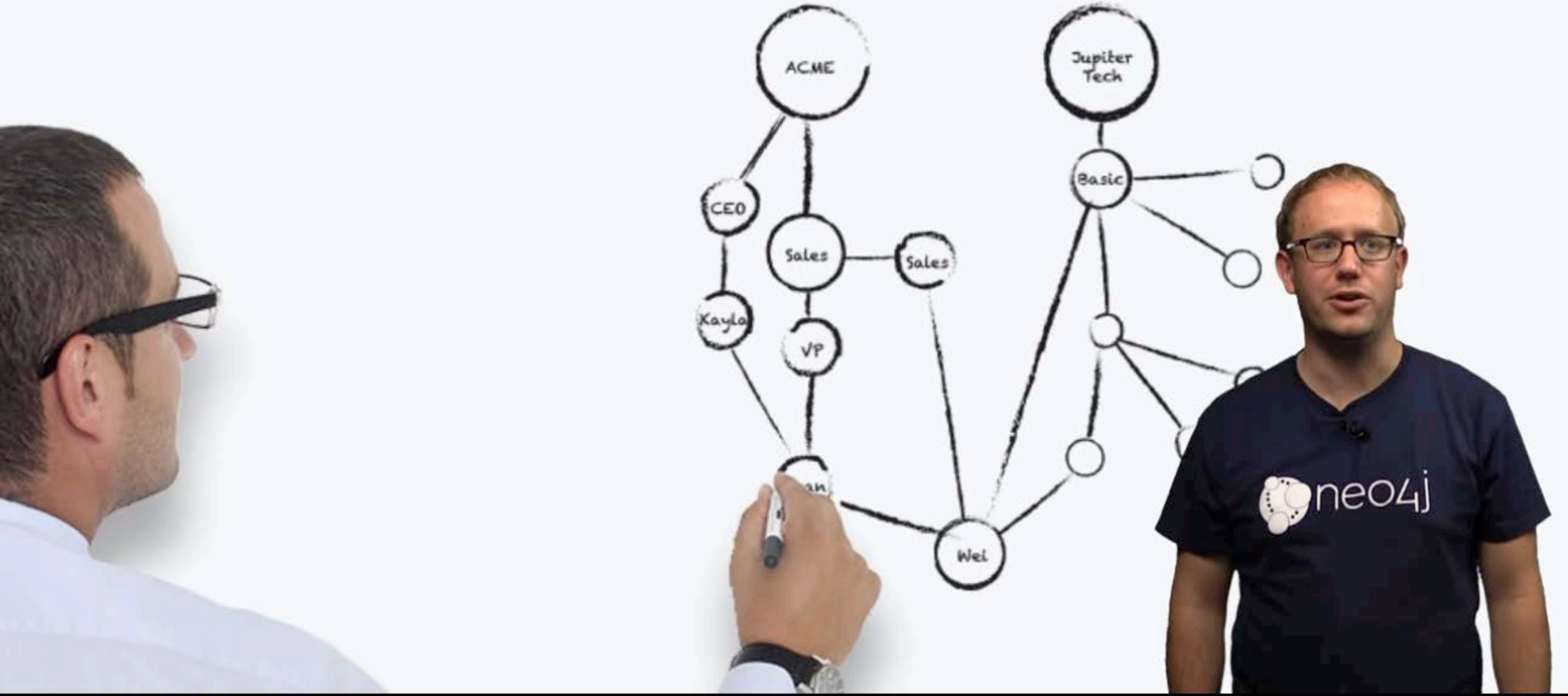
# Agility

# Intuitiveness

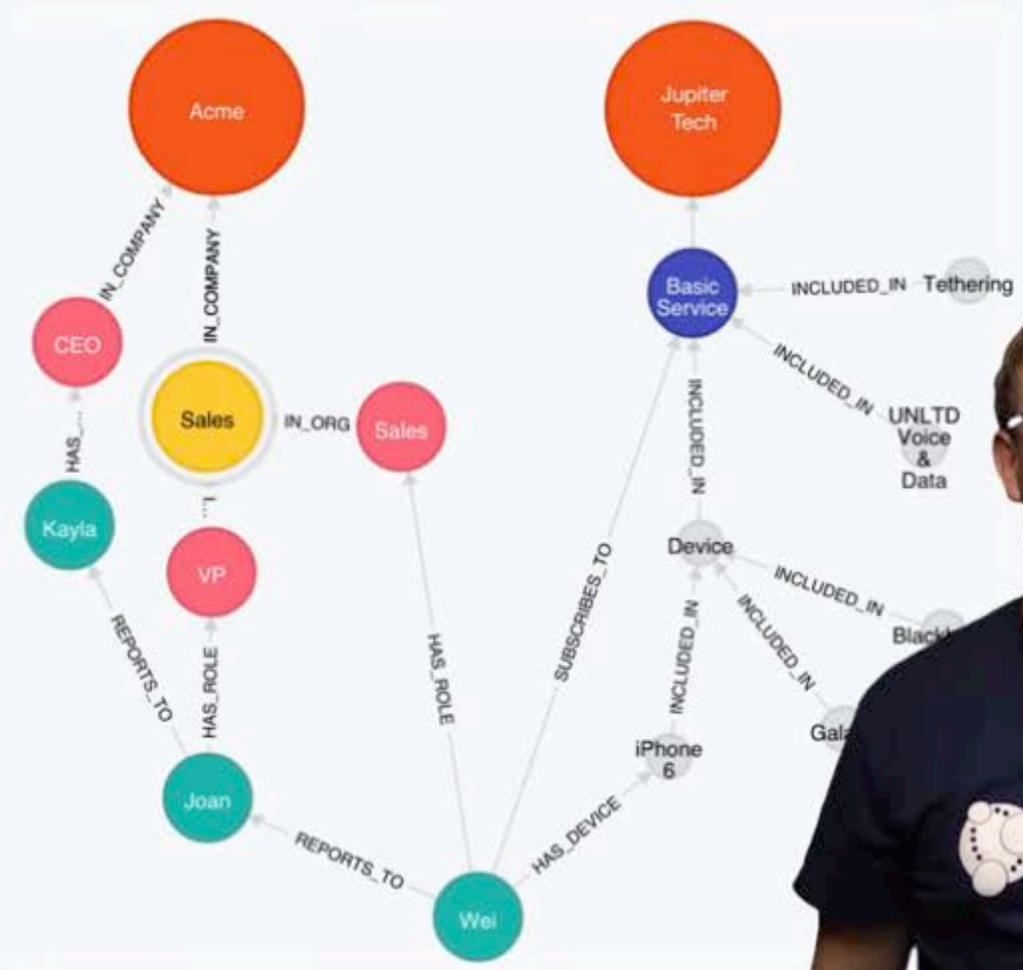
Speed

Agility

# Intuitiveness



# Intuitiveness




Intuitivness

**Speed**

Agility

# Speed

ebay

"We found Neo4j to be literally **thousands of times faster** than our prior MySQL solution, with queries that require **10-100 times less code**. Today, Neo4j provides eBay with functionality that was previously impossible."

- Volker Pacher, Senior Developer

**"Minutes to milliseconds" performance**

*Queries up to 1000x faster than RDBMS or other NoSQL*

Intuitivness

Speed

**Agility**

# Agility =

# A Naturally Adaptive Model

# Agility =

## A Naturally Adaptive Model

## +

## A Query Language Desig
## for Connectedness

# Cypher

## Typical Complex SQL Join

```
 (SELECT T.directReportees AS directReportees, sum(T.count) AS count
FROM (
SELECT manager.pid AS directReportees, 0 AS count
  FROM person_reportee manager
  WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
UNION
  SELECT manager.pid AS directReportees, count(manager.directly_manages) AS count
FROM person_reportee manager
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT manager.pid AS directReportees, count(reportee.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee reportee
ON manager.directly_manages = reportee.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT manager.pid AS directReportees, count(L2Reportees.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
) AS T
GROUP BY directReportees)
UNION
(SELECT T.directReportees AS directReportees, sum(T.count) AS count
FROM (
SELECT manager.directly_manages AS directReportees, 0 AS count
FROM person_reportee manager
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
UNION
SELECT reportee.pid AS directReportees, count(reportee.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee reportee
ON manager.directly_manages = reportee.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT depth1Reportees.pid AS directReportees,
count(depth2Reportees.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
) AS T
GROUP BY directReportees)
UNION
(SELECT T.directReportees AS directReportees, sum(T.count) AS count
  FROM(
  SELECT reportee.directly_manages AS directReportees, 0 AS count
FROM person_reportee manager
JOIN person_reportee reportee
ON manager.directly_manages = reportee.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT L2Reportees.pid AS directReportees, count(L2Reportees.directly_manages)
AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
) AS T
GROUP BY directReportees)
UNION
(SELECT L2Reportees.directly_manages AS directReportees, 0 AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
)
```

# Cypher

## Typical Complex SQL Join

```
 (SELECT T.directReportees AS directReportees, sum(T.count) AS count
FROM (
SELECT manager.pid AS directReportees, 0 AS count
  FROM person_reportee manager
  WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
UNION
  SELECT manager.pid AS directReportees, count(manager.directly_manages) AS count
FROM person_reportee manager
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT manager.pid AS directReportees, count(reportee.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee reportee
ON manager.directly_manages = reportee.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT manager.pid AS directReportees, count(L2Reportees.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
) AS T
GROUP BY directReportees)
UNION
(SELECT T.directReportees AS directReportees, sum(T.count) AS count
FROM (
SELECT manager.directly_manages AS directReportees, 0 AS count
FROM person_reportee manager
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
UNION
SELECT reportee.pid AS directReportees, count(reportee.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee reportee
ON manager.directly_manages = reportee.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT depth1Reportees.pid AS directReportees,
count(depth2Reportees.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
) AS T
GROUP BY directReportees)
UNION
(SELECT T.directReportees AS directReportees, sum(T.count) AS count
  FROM(
  SELECT reportee.directly_manages AS directReportees, 0 AS count
FROM person_reportee manager
JOIN person_reportee reportee
ON manager.directly_manages = reportee.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT L2Reportees.pid AS directReportees, count(L2Reportees.directly_manages)
AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
) AS T
GROUP BY directReportees)
UNION
(SELECT L2Reportees.directly_manages AS directReportees, 0 AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
)
```

## The Same Query using Cypher

```
MATCH (boss)-[:MANAGES*0..3]->(sub),
      (sub)-[:MANAGES*1..3]->(report)
WHERE boss.name = "John Doe"
RETURN sub.name AS Subordinate,
  count(report) AS Total
```

# Cypher

## Typical Complex SQL Join

```
 (SELECT T.directReportees AS directReportees, sum(T.count) AS count
FROM (
SELECT manager.pid AS directReportees, 0 AS count
  FROM person_reportee manager
  WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
UNION
  SELECT manager.pid AS directReportees, count(manager.directly_manages) AS count
FROM person_reportee manager
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT manager.pid AS directReportees, count(reportee.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee reportee
ON manager.directly_manages = reportee.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT manager.pid AS directReportees, count(L2Reportees.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
) AS T
GROUP BY directReportees)
UNION
(SELECT T.directReportees AS directReportees, sum(T.count) AS count
FROM (
SELECT manager.directly_manages AS directReportees, 0 AS count
FROM person_reportee manager
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
UNION
SELECT reportee.pid AS directReportees, count(reportee.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee reportee
ON manager.directly_manages = reportee.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT depth1Reportees.pid AS directReportees,
count(depth2Reportees.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
) AS T
GROUP BY directReportees)
UNION
(SELECT T.directReportees AS directReportees, sum(T.count) AS count
  FROM(
  SELECT reportee.directly_manages AS directReportees, 0 AS count
FROM person_reportee manager
JOIN person_reportee reportee
ON manager.directly_manages = reportee.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT L2Reportees.pid AS directReportees, count(L2Reportees.directly_manages)
AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
) AS T
GROUP BY directReportees)
UNION
(SELECT L2Reportees.directly_manages AS directReportees, 0 AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
)
```

## The Same Query using Cypher

```
MATCH (boss)-[:MANAGES*0..3]->(sub),
      (sub)-[:MANAGES*1..3]->(report)
WHERE boss.name = "John Doe"
RETURN sub.name AS Subordinate,
  count(report) AS Total
```

## Project Impact

Less time writing

Less time de

Code that's

# Use Cases

# NEO4j USE CASES

Real Time Recommendations

Master Data Management

Fraud Detection

Graph Based Search

Network & IT-Operations

Identity & Access Management

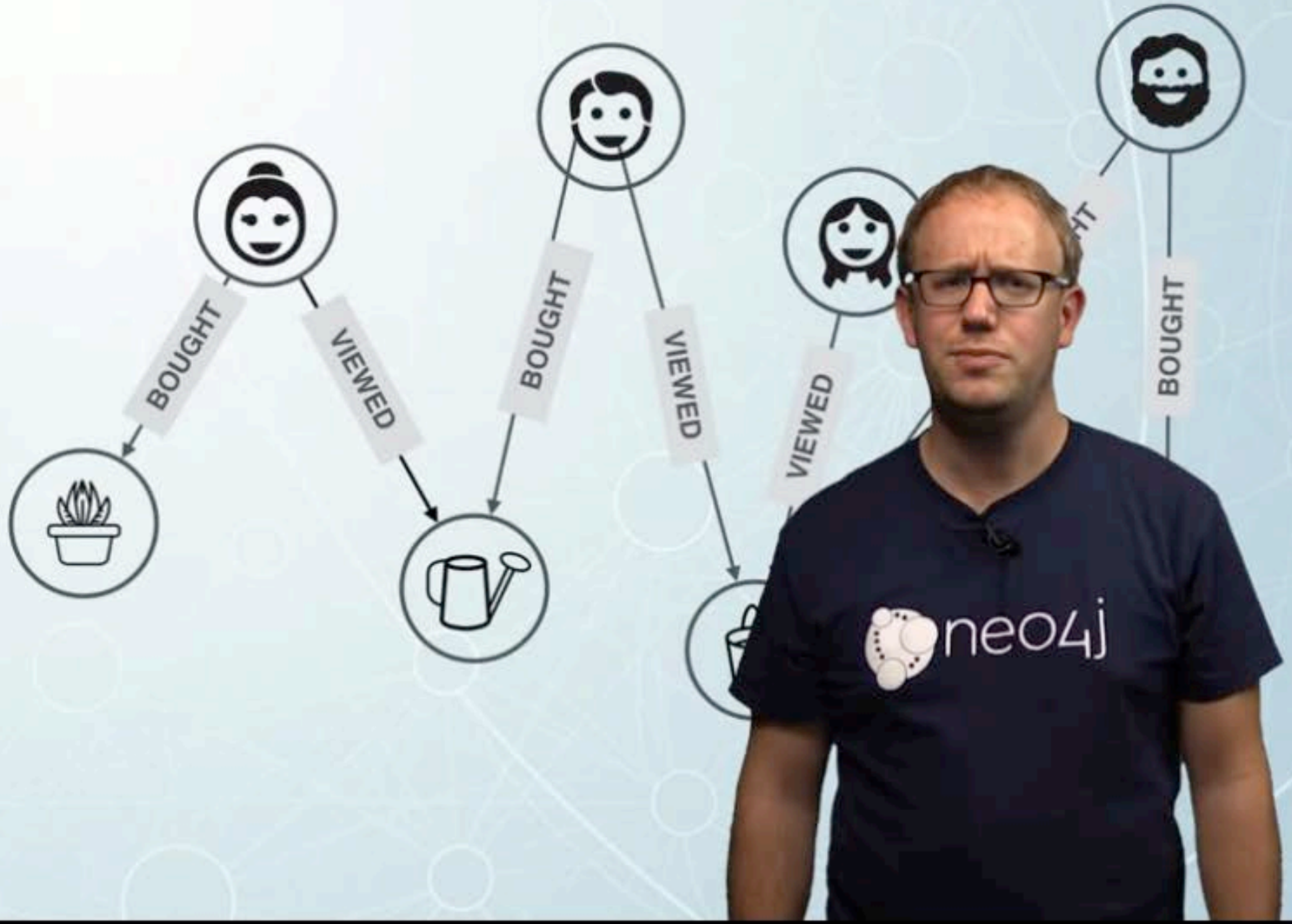
neo4j

# NEO4j USE CASES

- Real Time Recommendations
- Master Data Management
- Fraud Detection
- Graph Based Search
- Network & IT-Operations
- Identity & Access Management

neo4j

**GRAPH THINKING:**
Real Time Recommendations

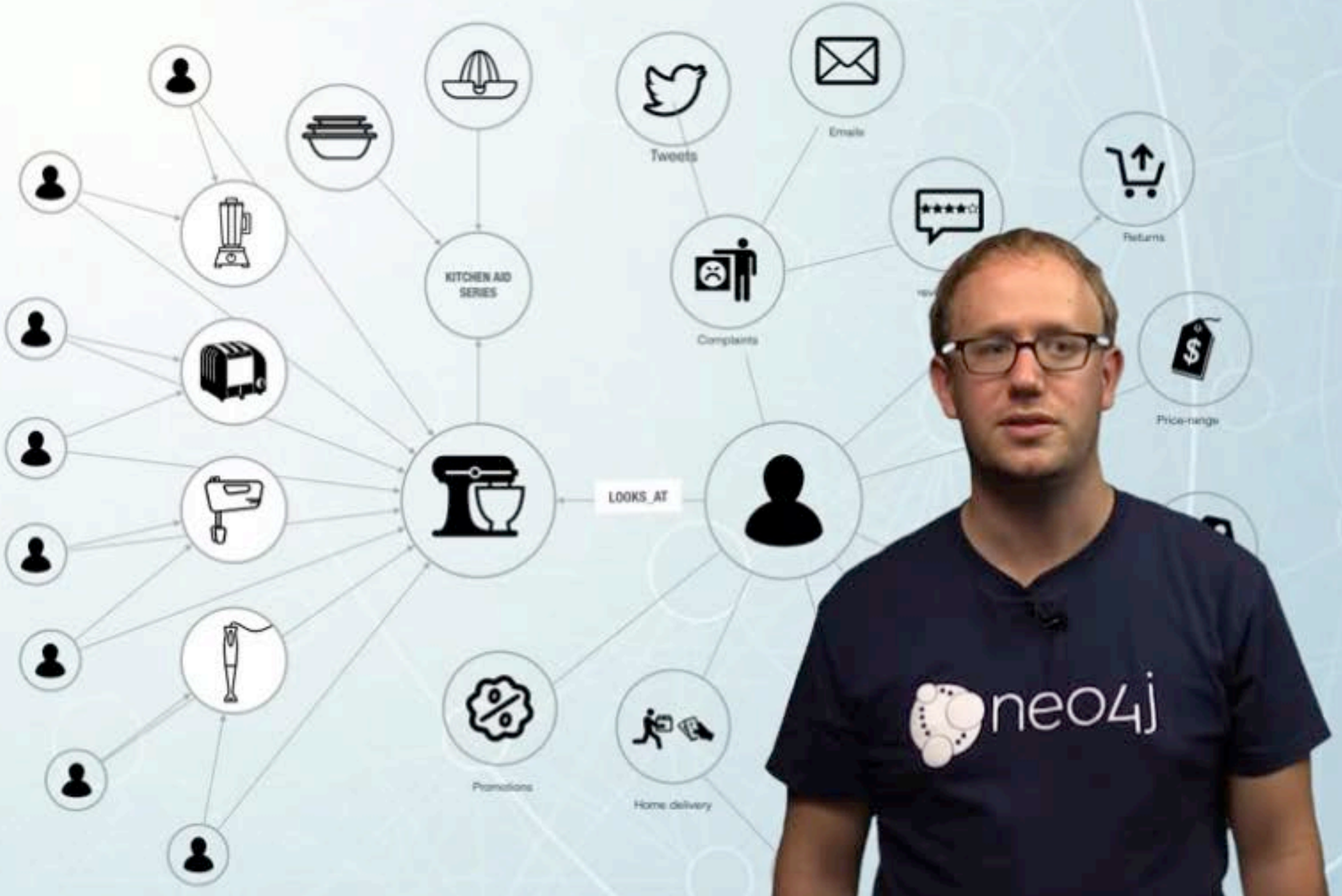


# NEO4j USE CASES

- Real Time Recommendations
- Master Data Management
- Fraud Detection
- Graph Based Search
- Network & IT-Operations
- Identity & Access Management

neo4j

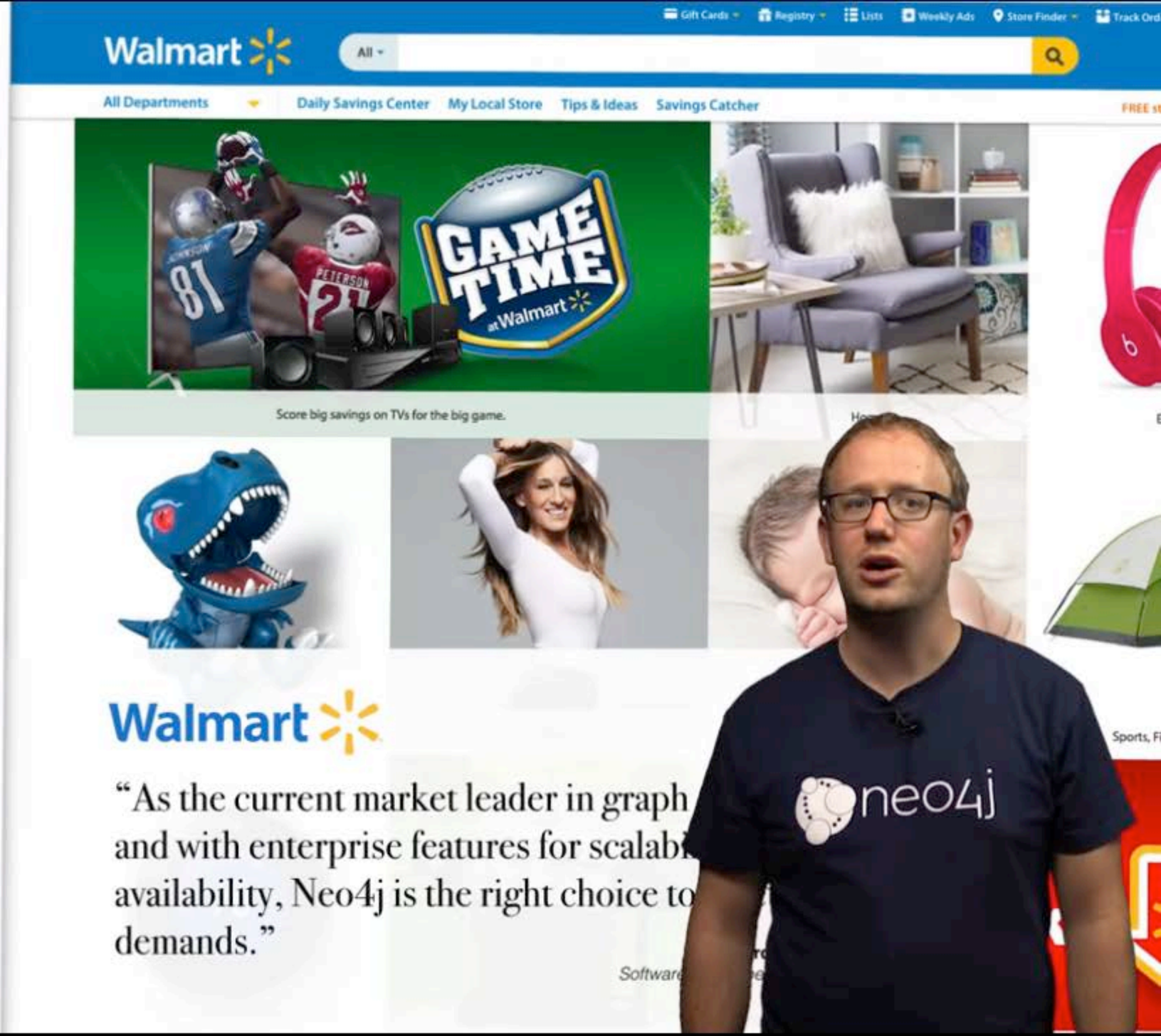
## GRAPH THINKING:
## Real Time Recommendations



# NEO4j USE CASES

- Real Time Recommendations
- Master Data Management
- Fraud Detection
- Graph Based Search
- Network & IT-Operations
- Identity & Access Management

neo4j

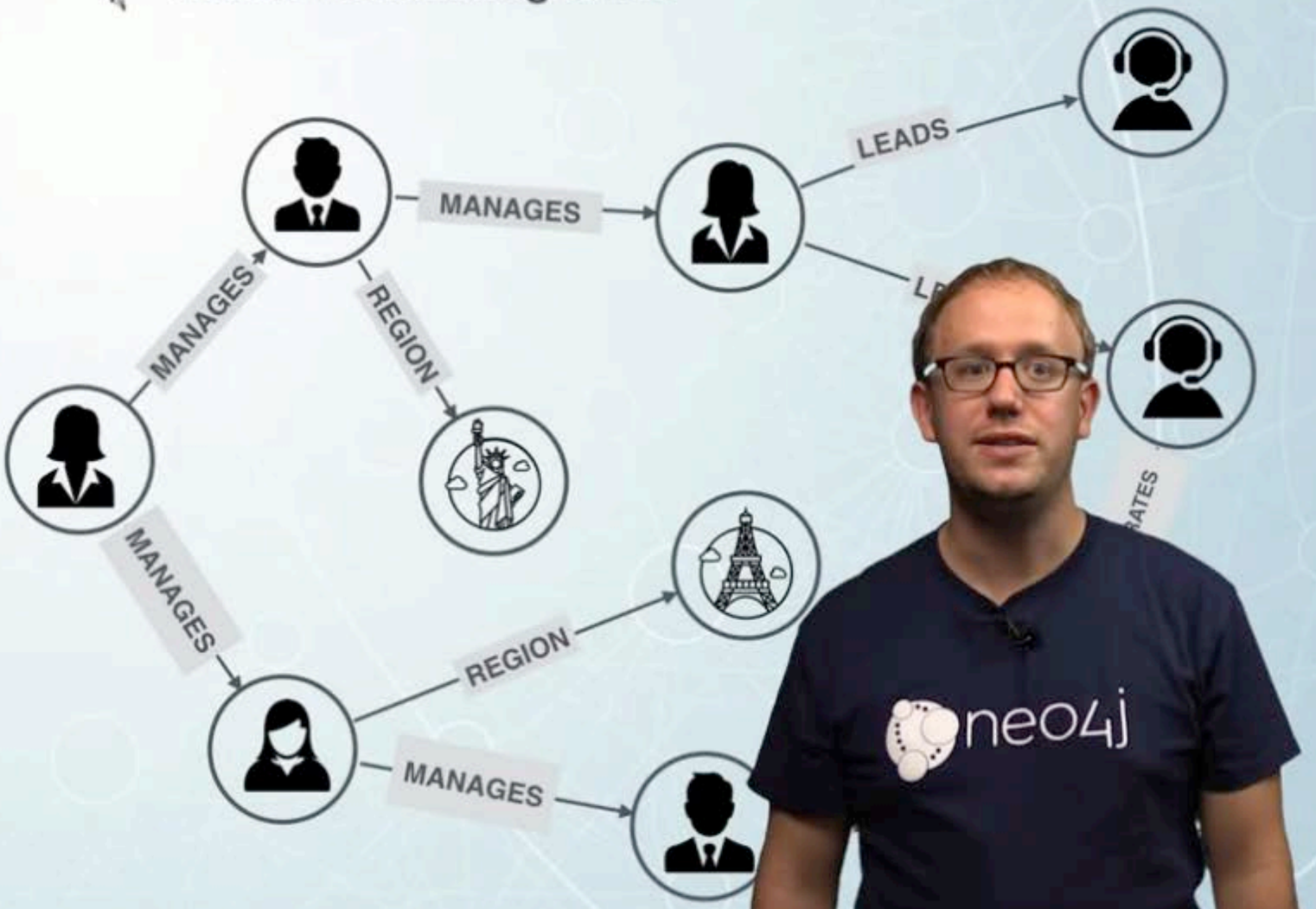


# NEO4j USE CASES

- Real Time Recommendations
- Master Data Management
- Fraud Detection
- Graph Based Search
- Network & IT-Operations
- Identity & Access Management

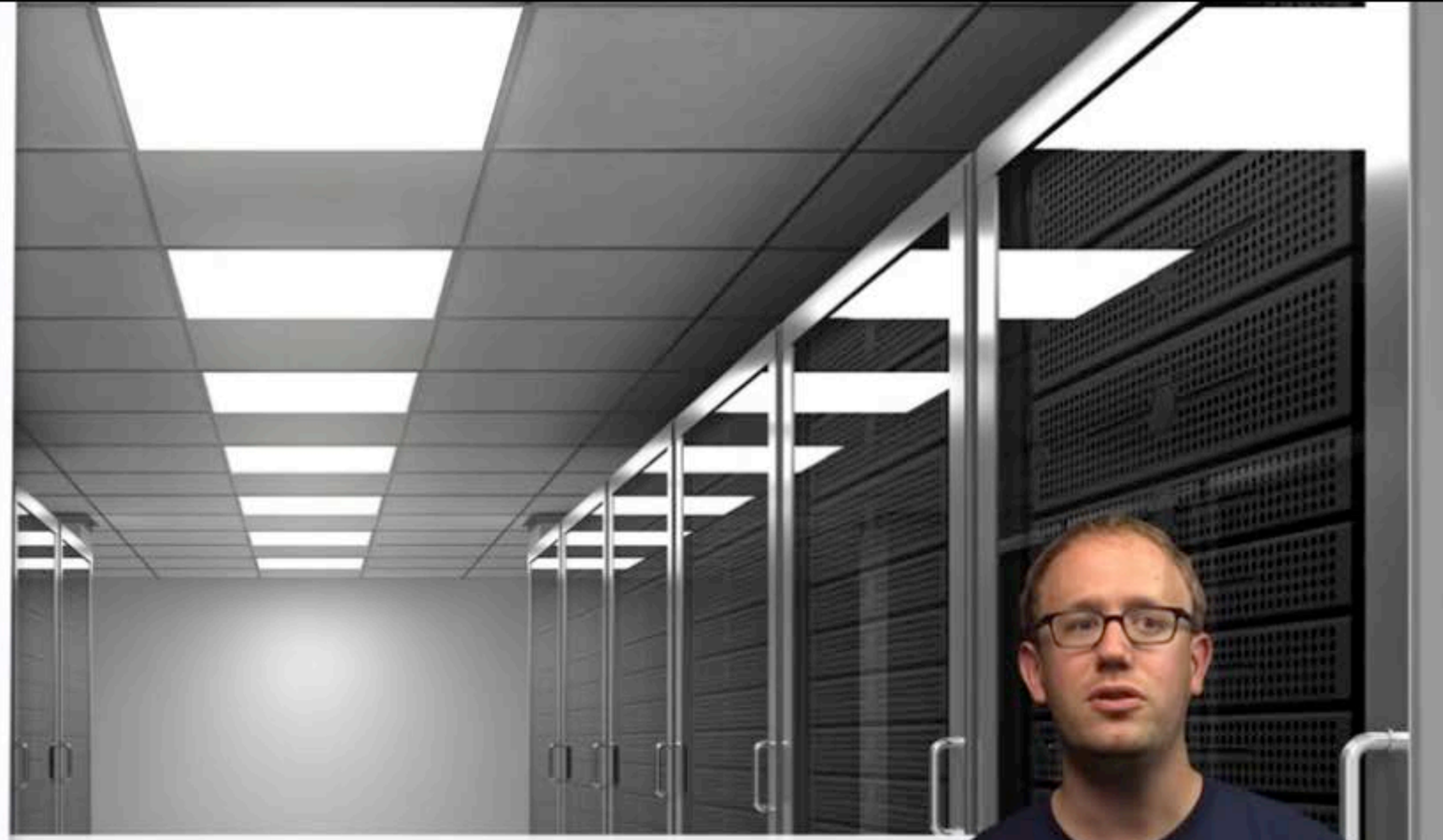
neo4j

**GRAPH THINKING:**
Master Data Management



# NEO4j USE CASES

- Real Time Recommendations
- Master Data Management
- Fraud Detection
- Graph Based Search
- Network & IT-Operations
- Identity & Access Management



CISCO

Neo4j is the heart of Cisco HMP: used for g
and single source of truth and a one-stop sl
of Cisco's hierarchies.


neo4j

# NEO4j USE CASES

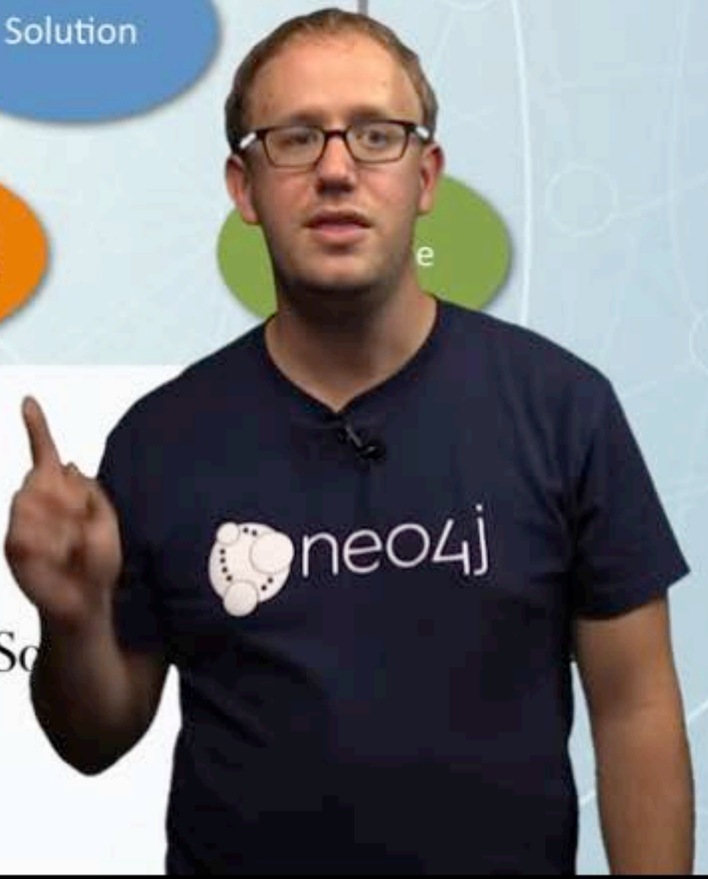
- Real Time Recommendations
- Master Data Management
- Fraud Detection
- Graph Based Search
- Network & IT-Operations
- Identity & Access Management

**GRAPH THINKING:**

## Master Data Management


CISCO

Neo4j is the heart of Cisco's Helpdesk So




neo4j

# NEO4j USE CASES

Real Time Recommendations

Master Data Management

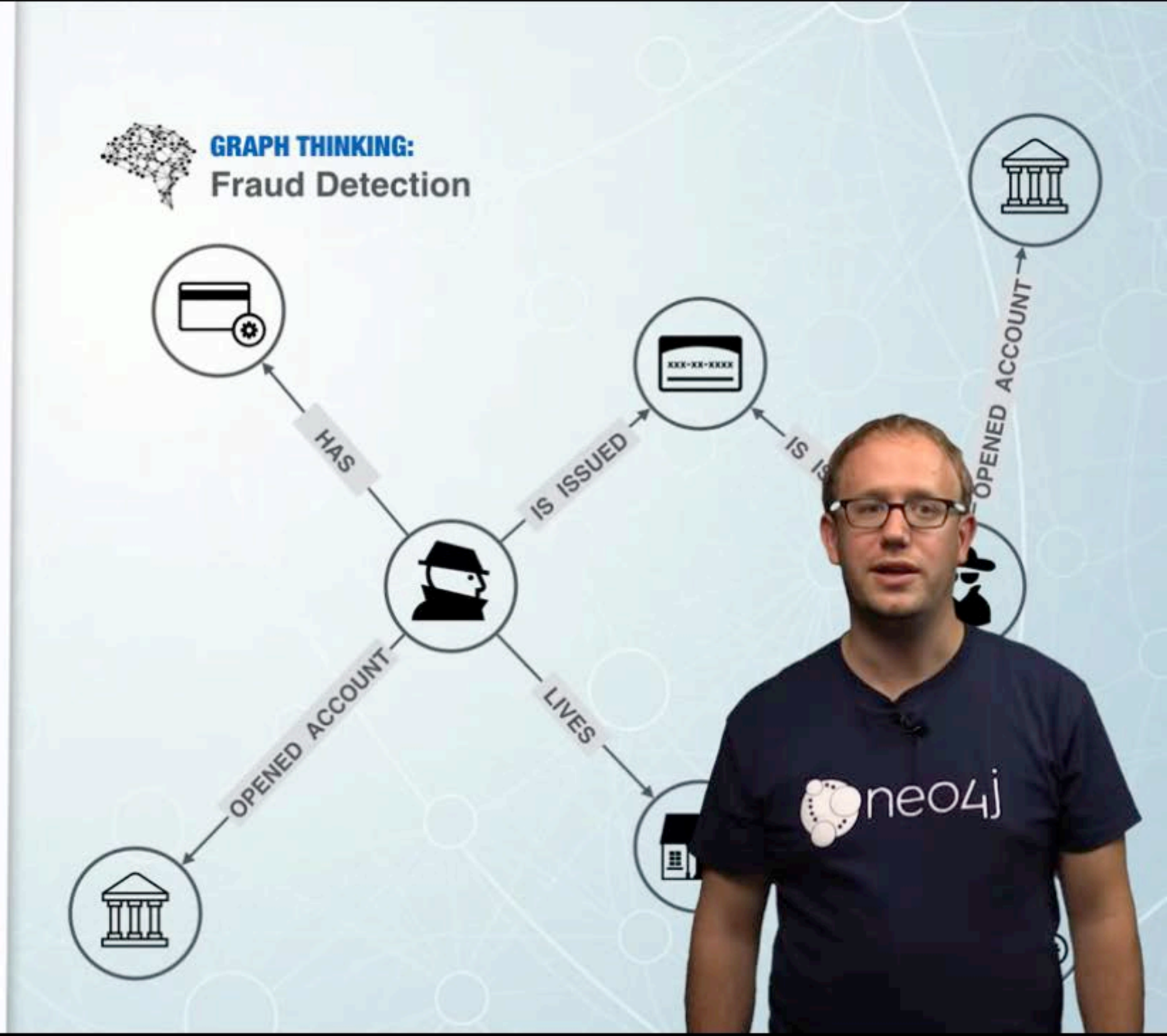
Fraud Detection

Graph Based Search

Network & IT-Operations

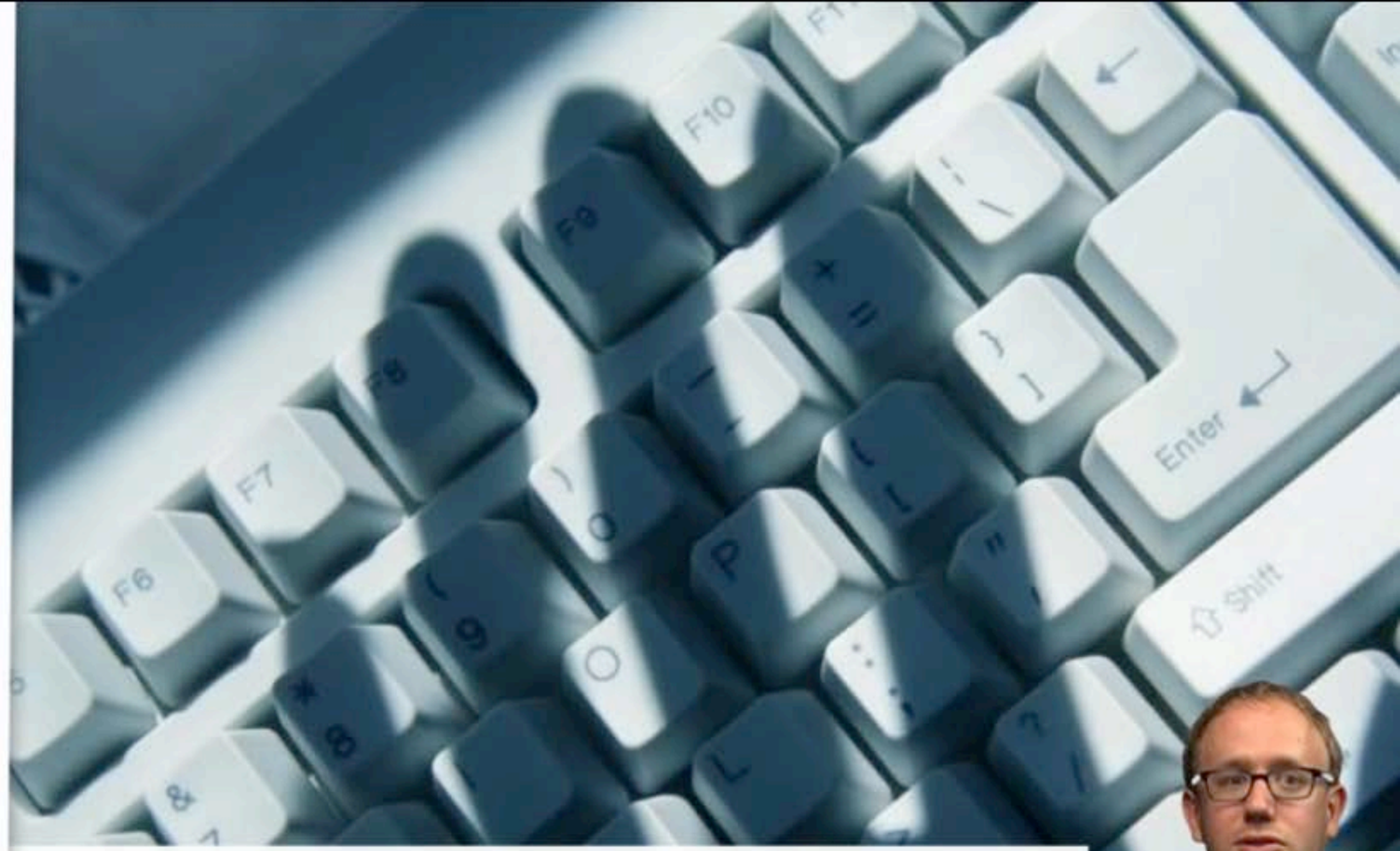
Identity & Access Management

neo4j



# NEO4j USE CASES

- Real Time Recommendations
- Master Data Management
- Fraud Detection
- Graph Based Search
- Network & IT-Operations
- Identity & Access Management



"Graph databases offer new methods of uncovering fraud rings and other sophisticated scams with a high-level of accuracy, and are capable of stopping advanced fraud scenarios in real-time."

**Gorka Sadowski**
*Cyber Security Expert*


neo4j

# NEO4j USE CASES

Real Time Recommendations

Master Data Management

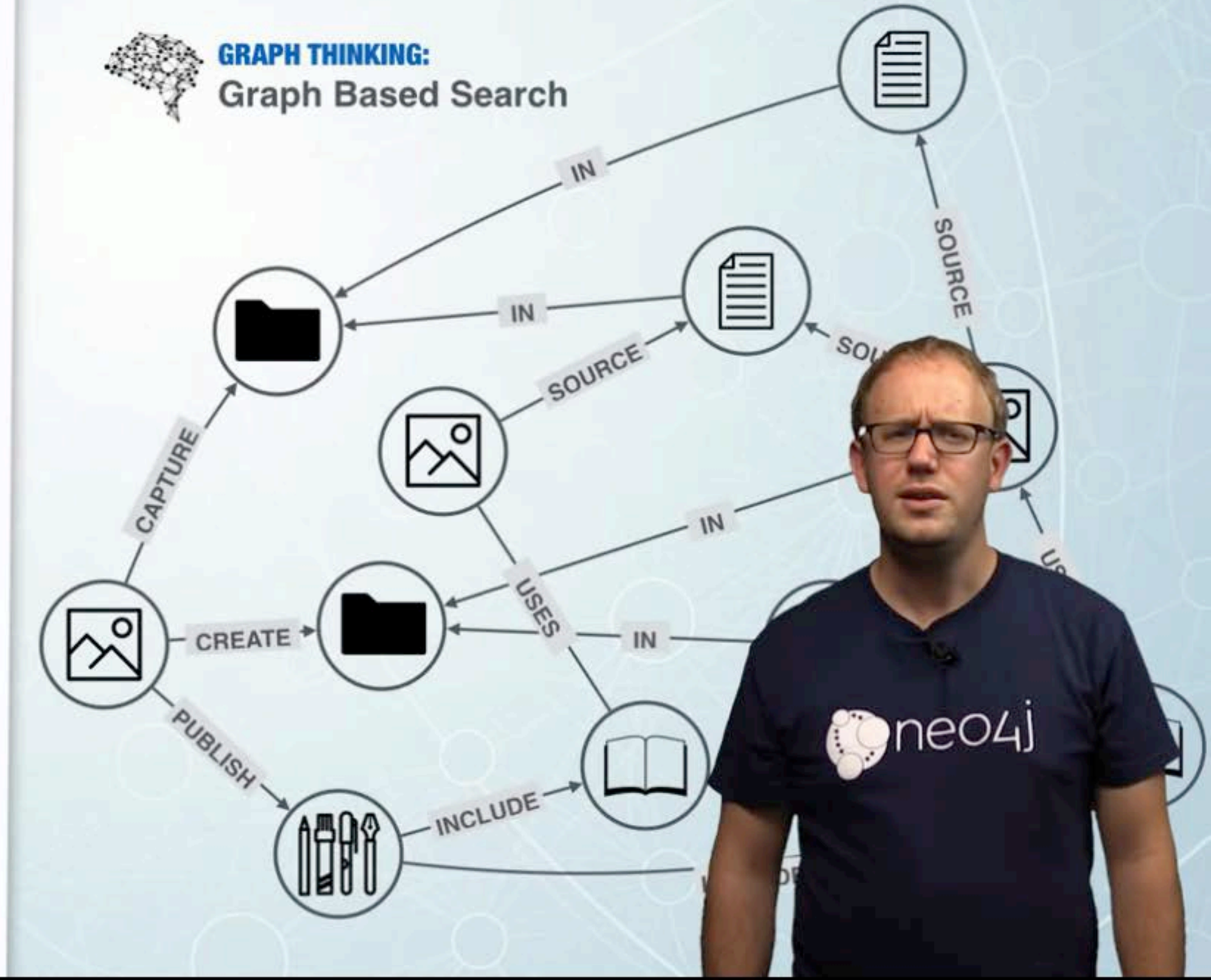
Fraud Detection

Graph Based Search

Network & IT-Operations

Identity & Access Management

neo4j



# NEO4j USE CASES

- Real Time Recommendations
- Master Data Management
- Fraud Detection
- Graph Based Search
- Network & IT-Operations
- Identity & Access Management


**Lufthansa**

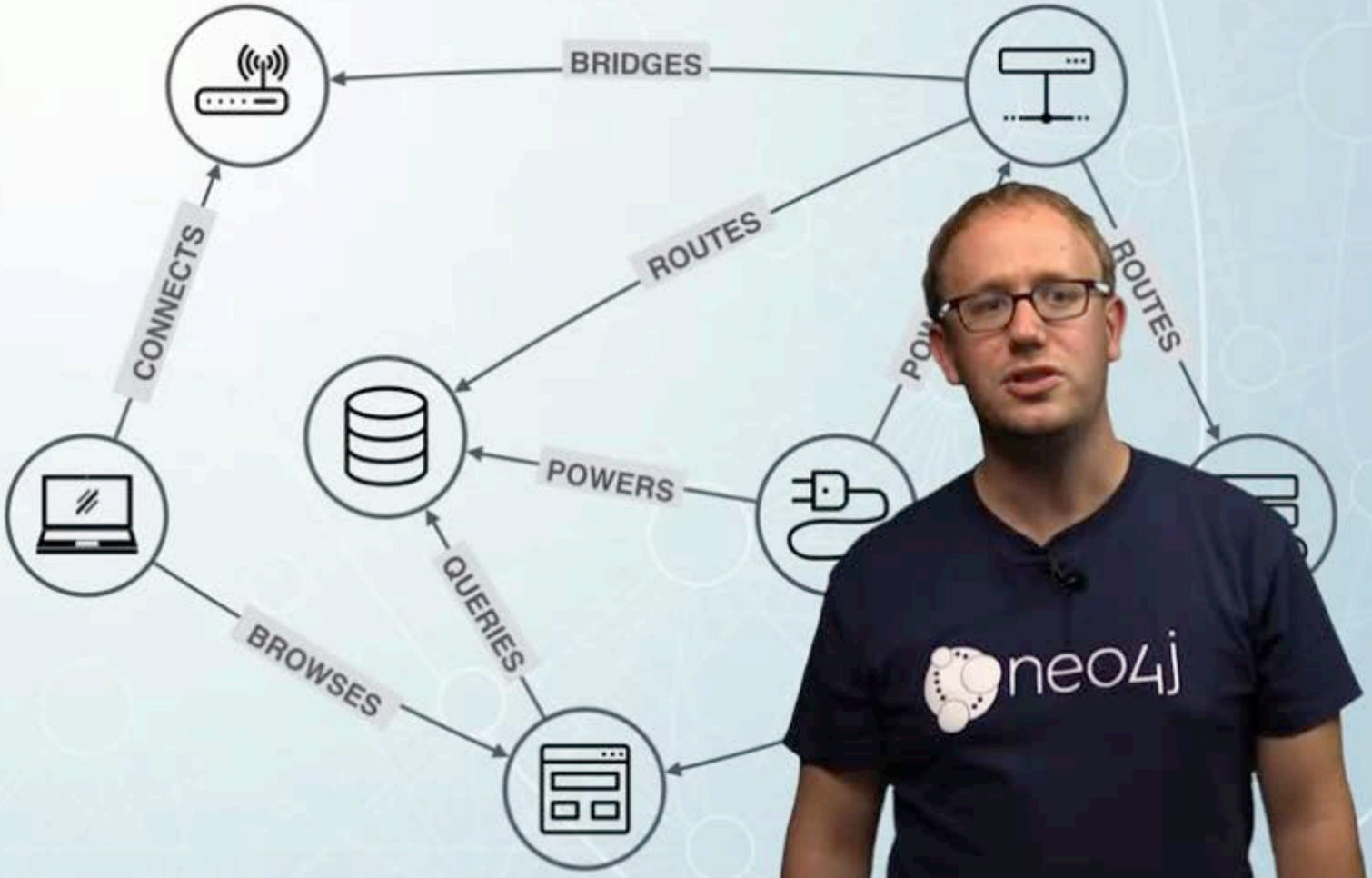
Uses Neo4j to manage the digital assets inside of i
generation in-flight entertainment system.

# NEO4j USE CASES

- Real Time Recommendations
- Master Data Management
- Fraud Detection
- Graph Based Search
- Network & IT-Operations
- Identity & Access Management

neo4j

**GRAPH THINKING:**
Network & IT-Operations



# NEO4j USE CASES

- Real Time Recommendations
- Master Data Management
- Fraud Detection
- Graph Based Search
- Network & IT-Operations
- Identity & Access Management

neo4j

LendingClub / macgyver

Watch 9 Star

Code Issues 38 Pull requests 1 Wiki Pulse Graphs

DevOps Multi-Tool

905 commits 18 branches 6 releases 4 contribu

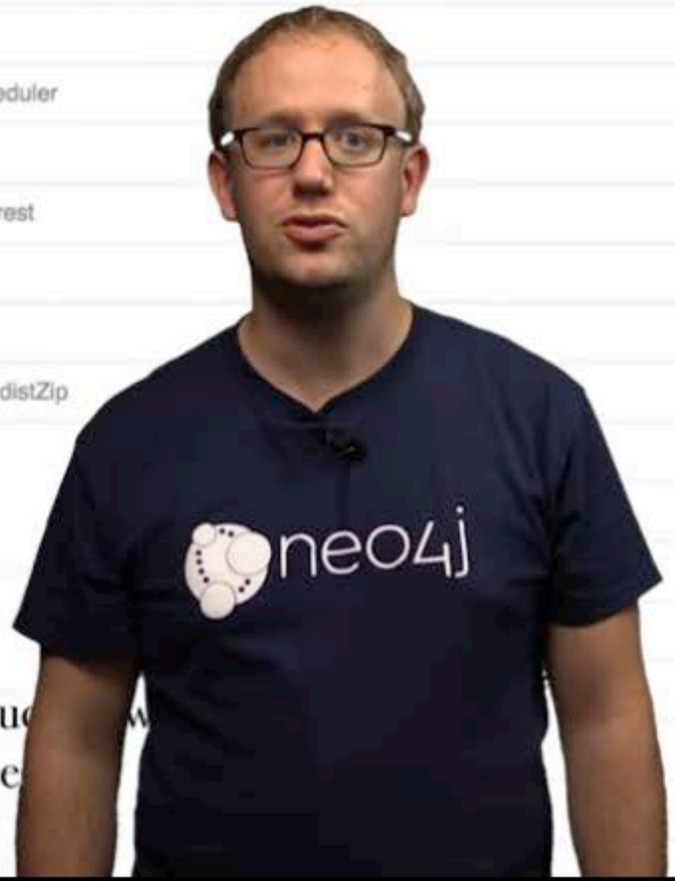
Branch: master New pull request New file Upload files Find file HTTPS https://github.com/Lendin

lf6was9 Merge pull request #278 from LendingClub/feature/manual-schedule ... Latest commit da60dd4

| | |
|---|---|
| config | #74 add license headers |
| gradle/wrapper | gradle 2.10 |
| macgyver-core | fix #277 major refactor of scheduler |
| macgyver-plugin-artifactory | license format |
| macgyver-plugin-atlassian | upgrade neorx, okhttp and okrest |
| macgyver-plugin-ci-jenkins | license headers |
| macgyver-plugin-ci | organize imports |
| macgyver-plugin-clojure | exclude jars from distTar and distZip |

## Lending Club

Manages their DevOps micro services infrastru
MacGyver, an open-sourced tool they develope

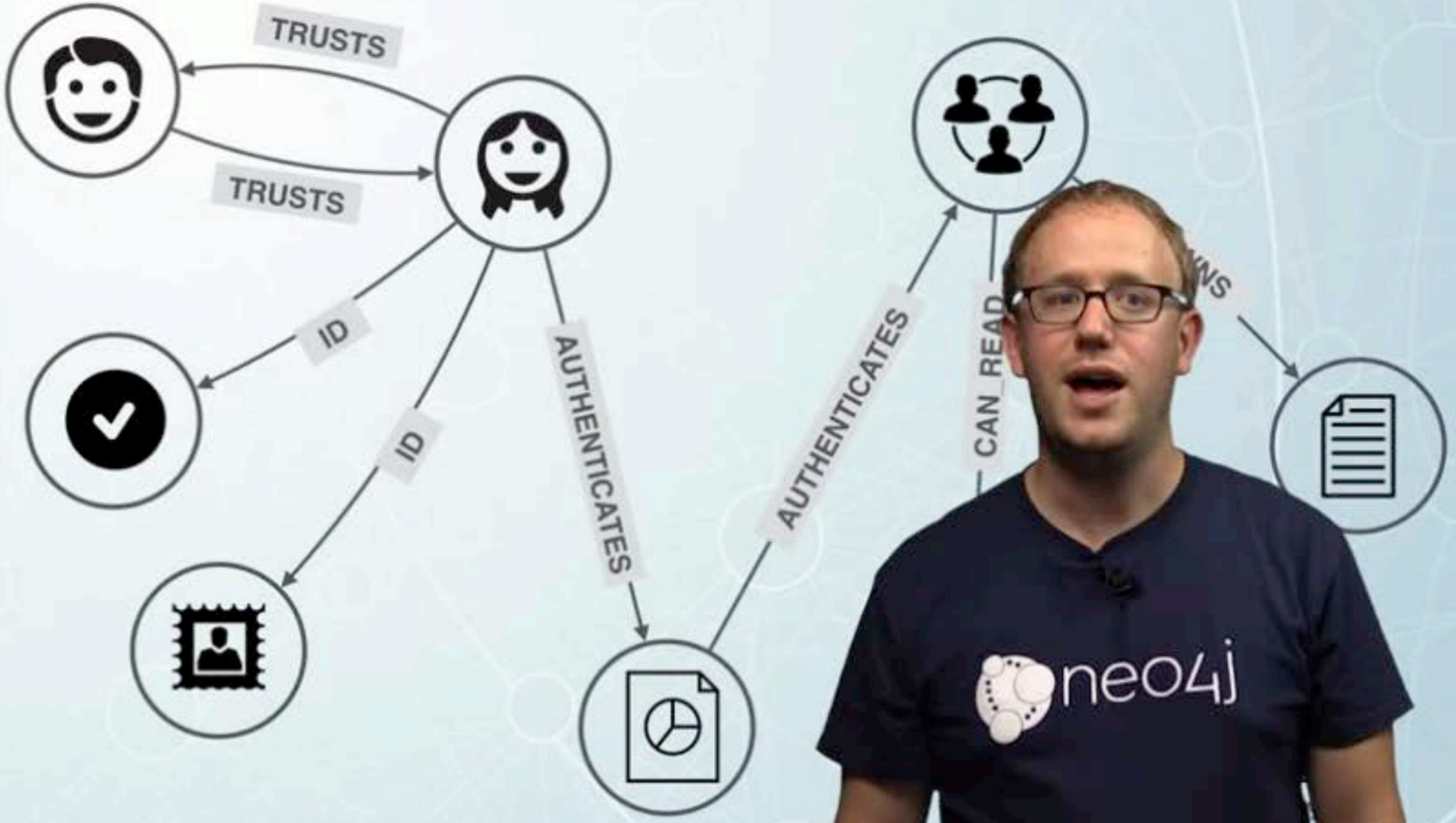


# NEO4j USE CASES

- Real Time Recommendations
- Master Data Management
- Fraud Detection
- Graph Based Search
- Network & IT-Operations
- Identity & Access Management

neo4j

**GRAPH THINKING:**
Identity And Access Management



# NEO4j USE CASES

Real Time Recommendations

Master Data Management

Fraud Detection

Graph Based Search

k & IT-Operations

Id & Access Management

# Neo4j Adoption by Selected Verticals

neo4j

| SOFTWARE | FINANCIAL SERVICES | RETAIL | M… B… | SOCIAL NETWORKS | TELECOM | HEALTHCARE |
|---|---|---|---|---|---|---|
| EQUILAR | UBS | Walmart | | LinkedIn | CISCO | RALLY |
| POLYVORE | First Data. | ebay | | Medium | SFR | Acuraspan |
| TRACEONE | die Bayerische | NORDSTROM | | SNAP | telenor | ZEPHYR |
| 53 | ADP | B&H | | ACADEMIA | | NaviNet |
| entropy | WESTERN UNION | | | | | compugen |
| ALERTLOGIC | veda | | | | CenturyLink | doximity |
| NetApp | bpifrance | | | | rancard | CHRONOTRACK |
| Zenoss | NOMURA | | | | 3 | |
| VIRTUAL | mallow street | | | | hp | |
| pitney bowes | Ria | | | | | |
| SERENA | noble group | | | | | |
| TOMTOM | | | | | | |


# THANK YOU!

neo4j

**Ryan Boyd**
*@ryguyrg*