

# Top 5 des cas d'utilisations

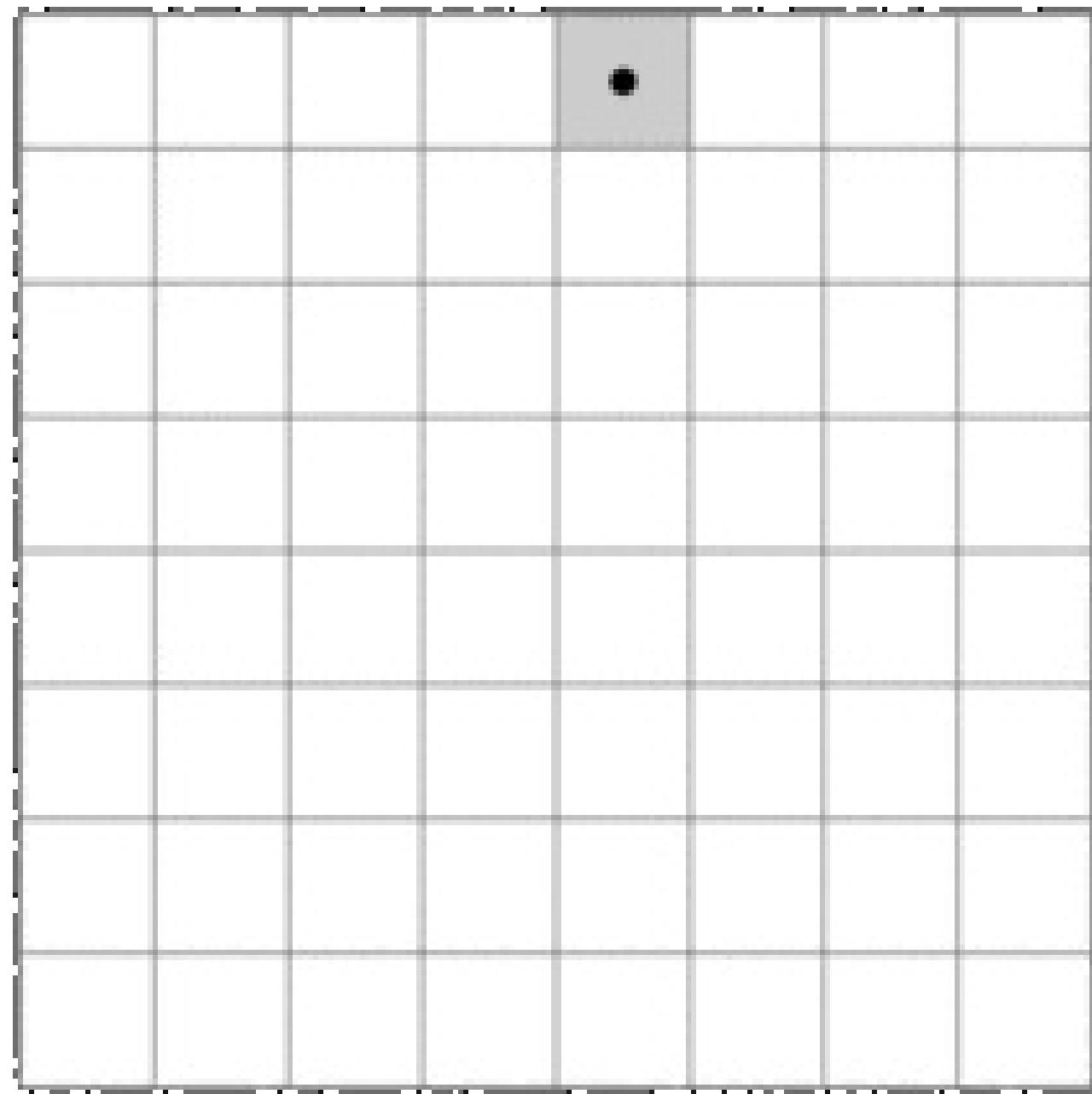
Benoit Simard (@logisima)



# **Un bref historique**

# Vers 800 : Le problème du cavalier

Un cavalier posé sur une case d'un échiquier doit en visiter toutes les cases sans passer deux fois sur la même.



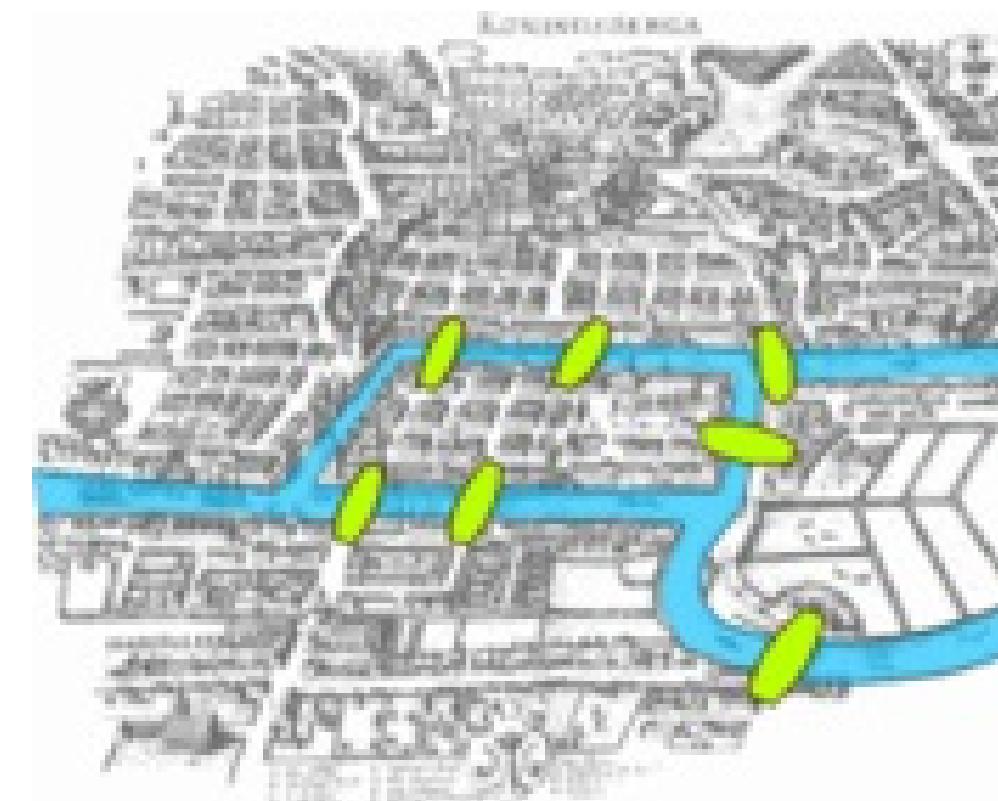
# Leonhard Euler 1707-1783

Mathématicien Suisse à l'origine de la théorie des graphes.

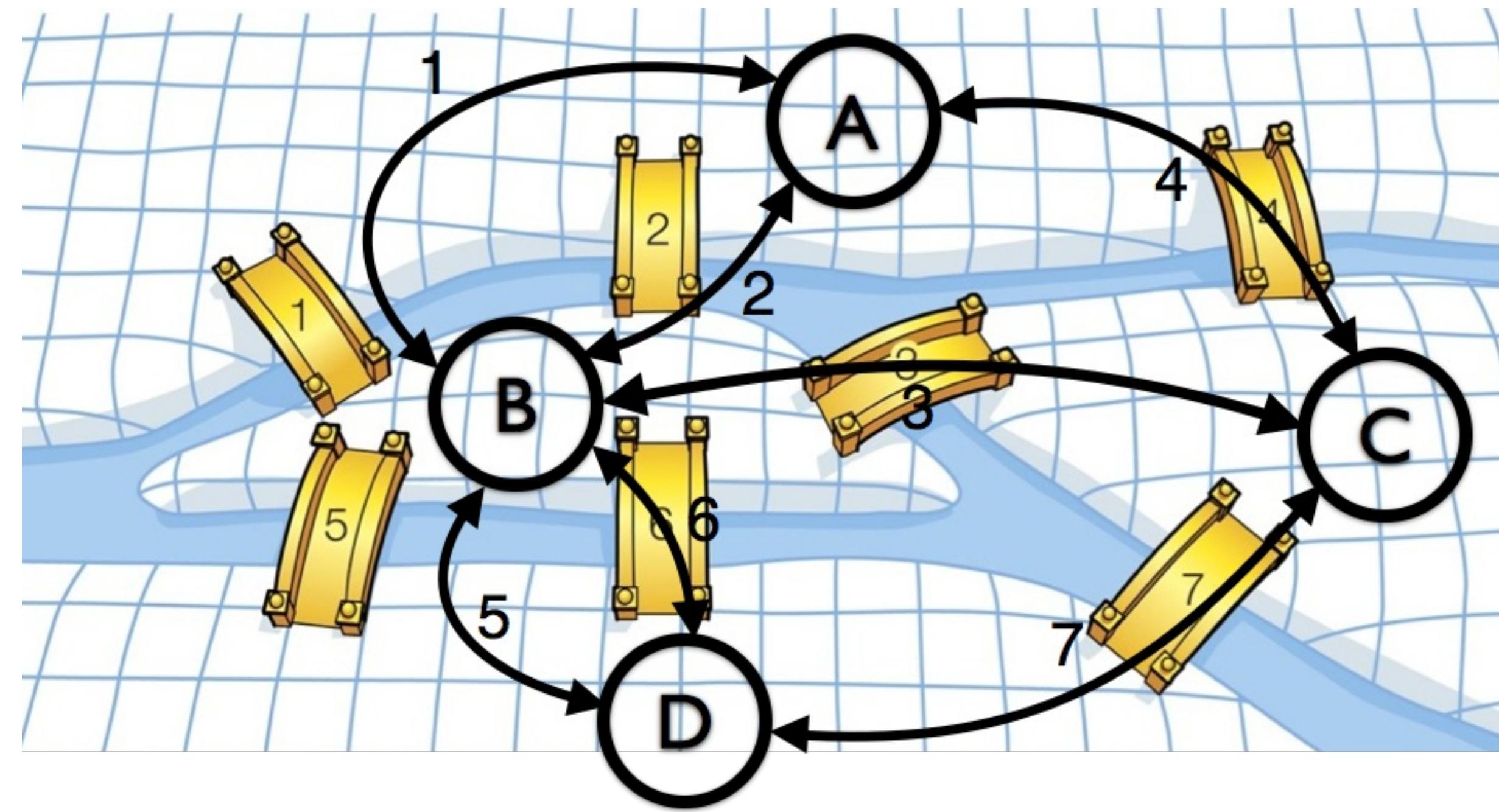


# Les 7 ponts de Königsberg

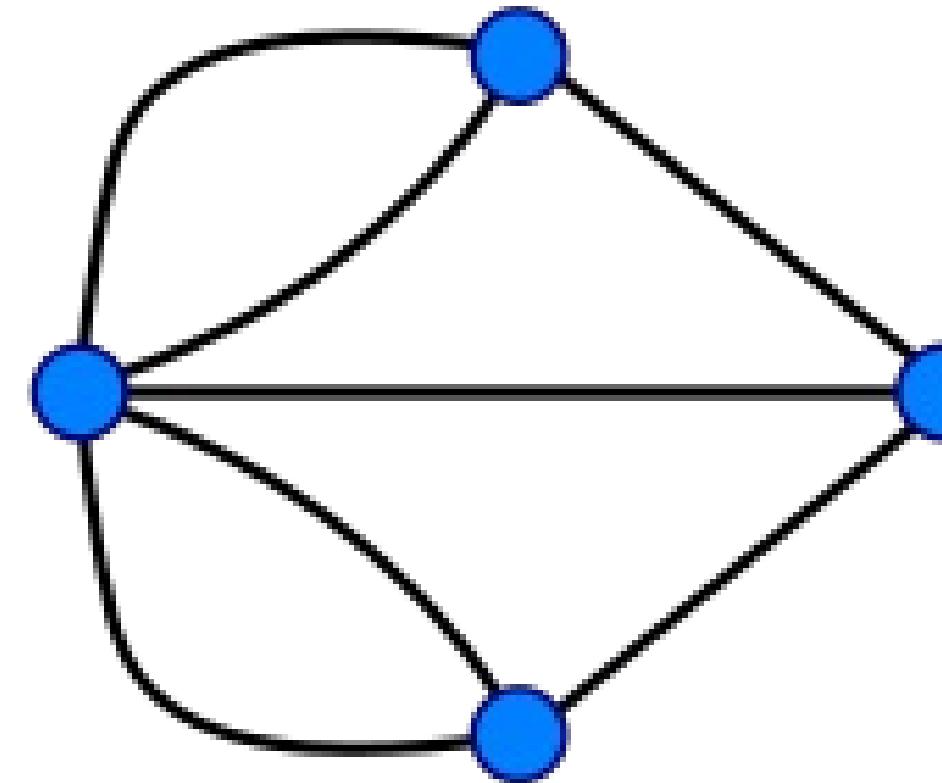
Est-ce qu'il existe une promenade permettant à partir d'un point de départ au choix, de passer une et une seule fois par chaque pont.



# Les 7 ponts de Königsberg



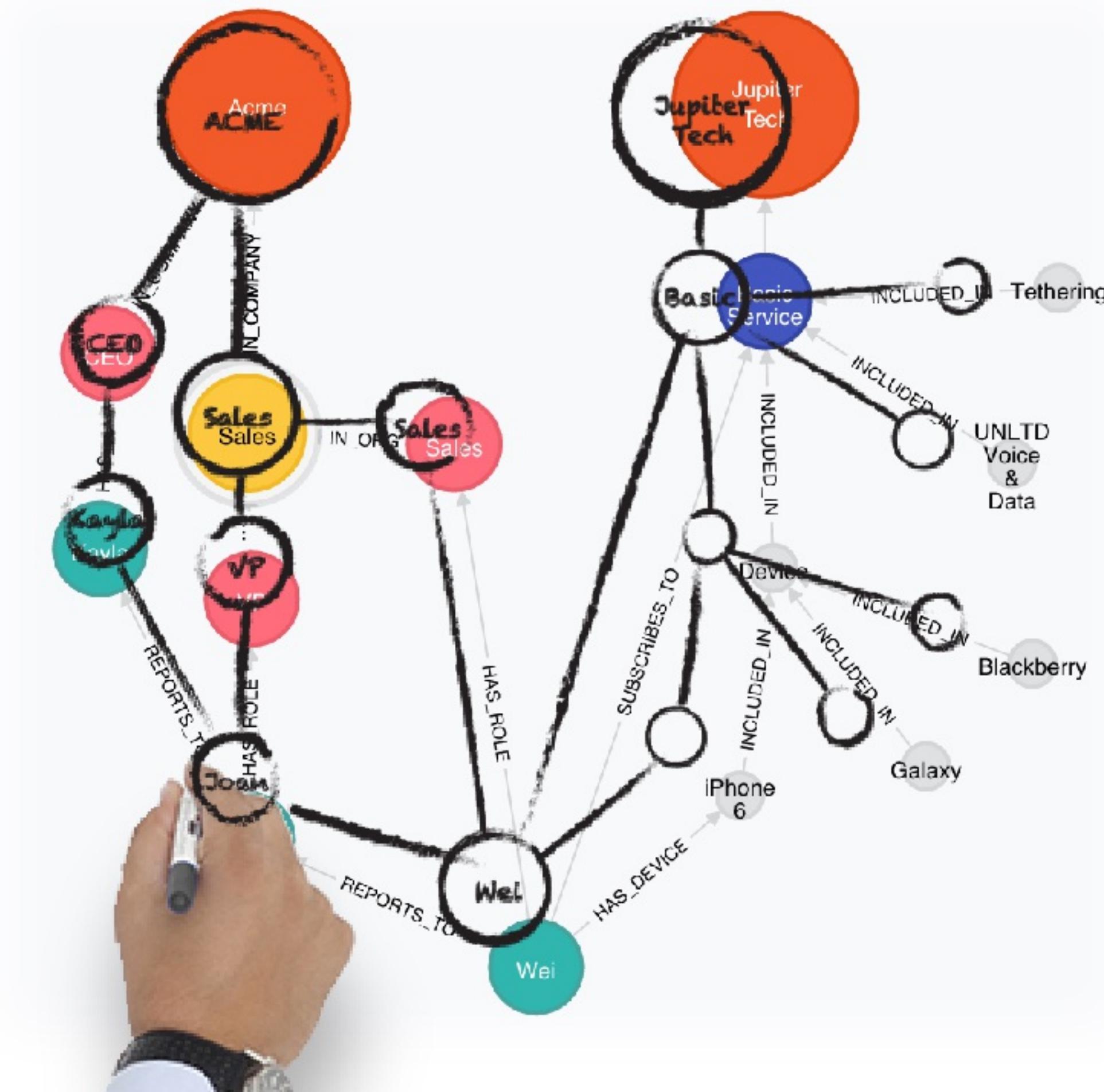
# Les 7 ponts de Königsberg



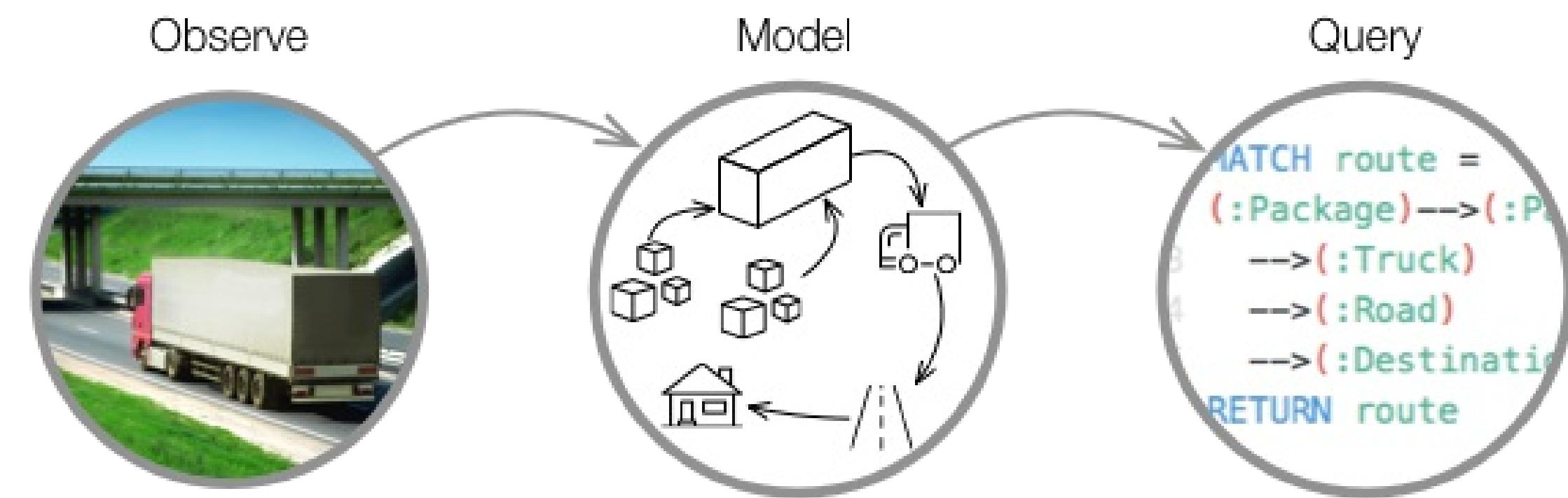
Théorie des graphes et des parcours Eulerien

# Les bases graphe

# La modélisation en graphe



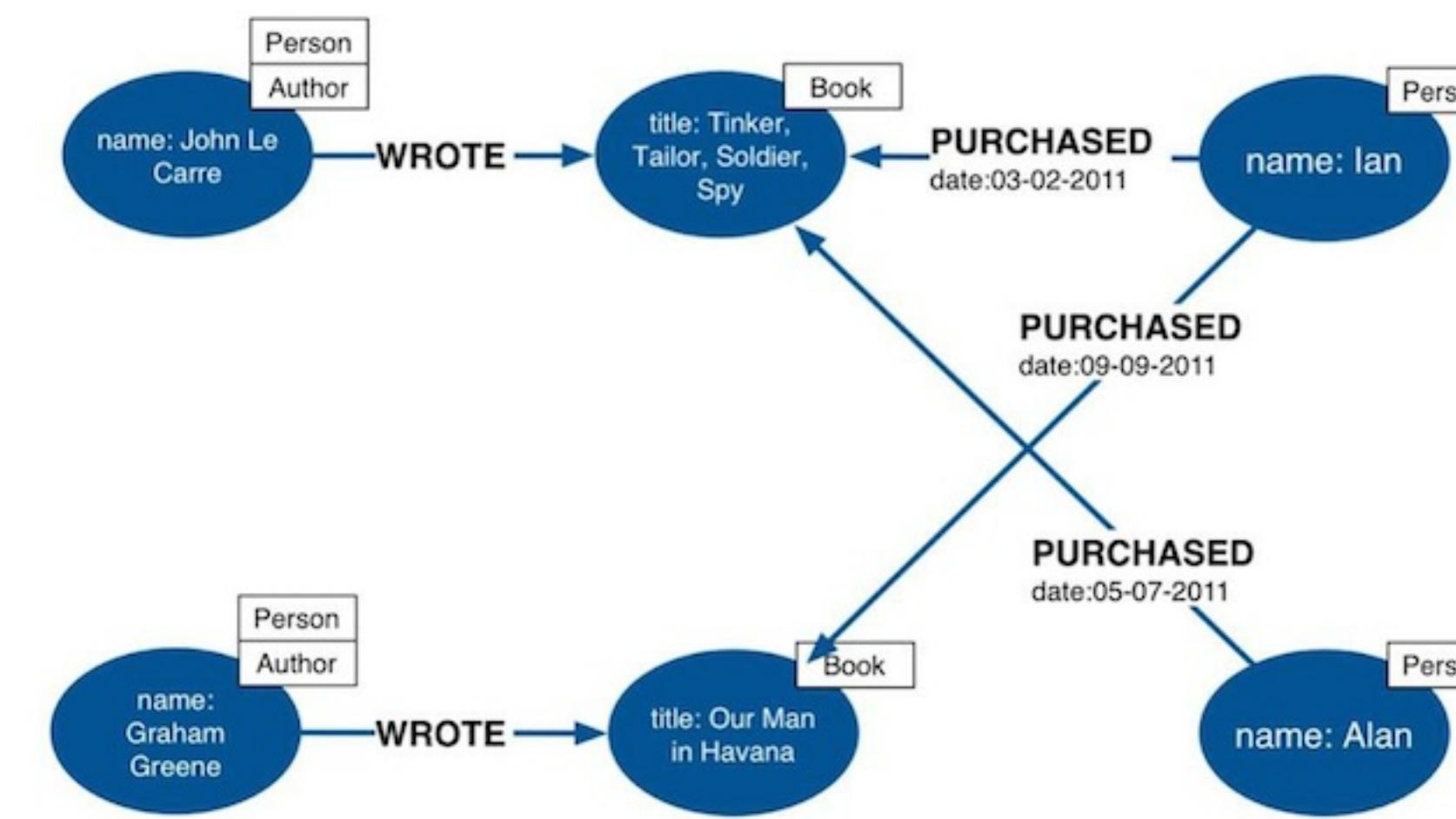
# Une modélisation naturelle



**La modélisation du monde réel et celle de la base sont identiques**

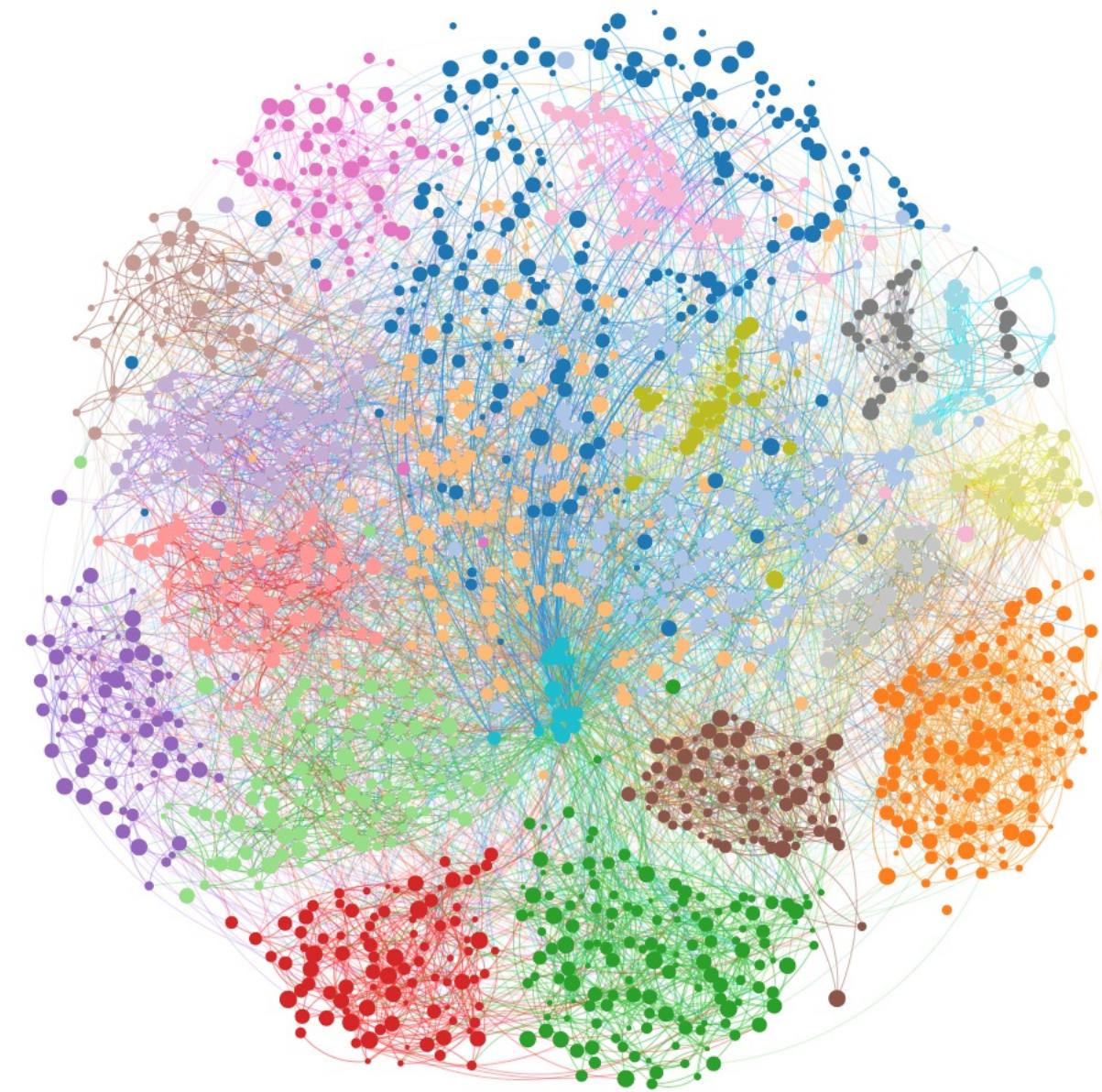
# Un exemple de graphe

- Des noeuds avec des propriétés et des labels
- Des relations typés et orientés avec des propriétés



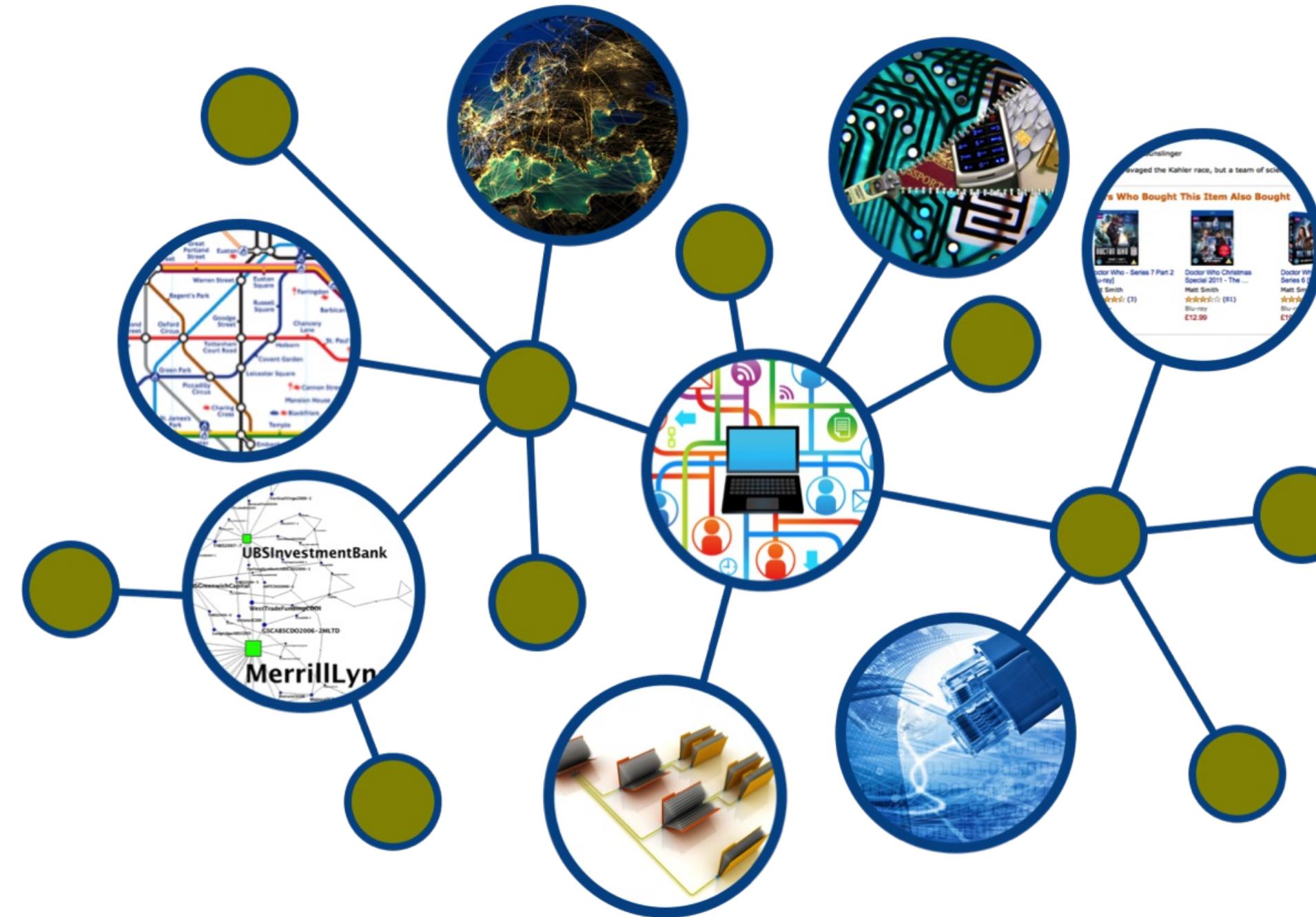
# Linked data

**Les relations existant entre les données sont presques plus importantes que les données elles-mêmes**



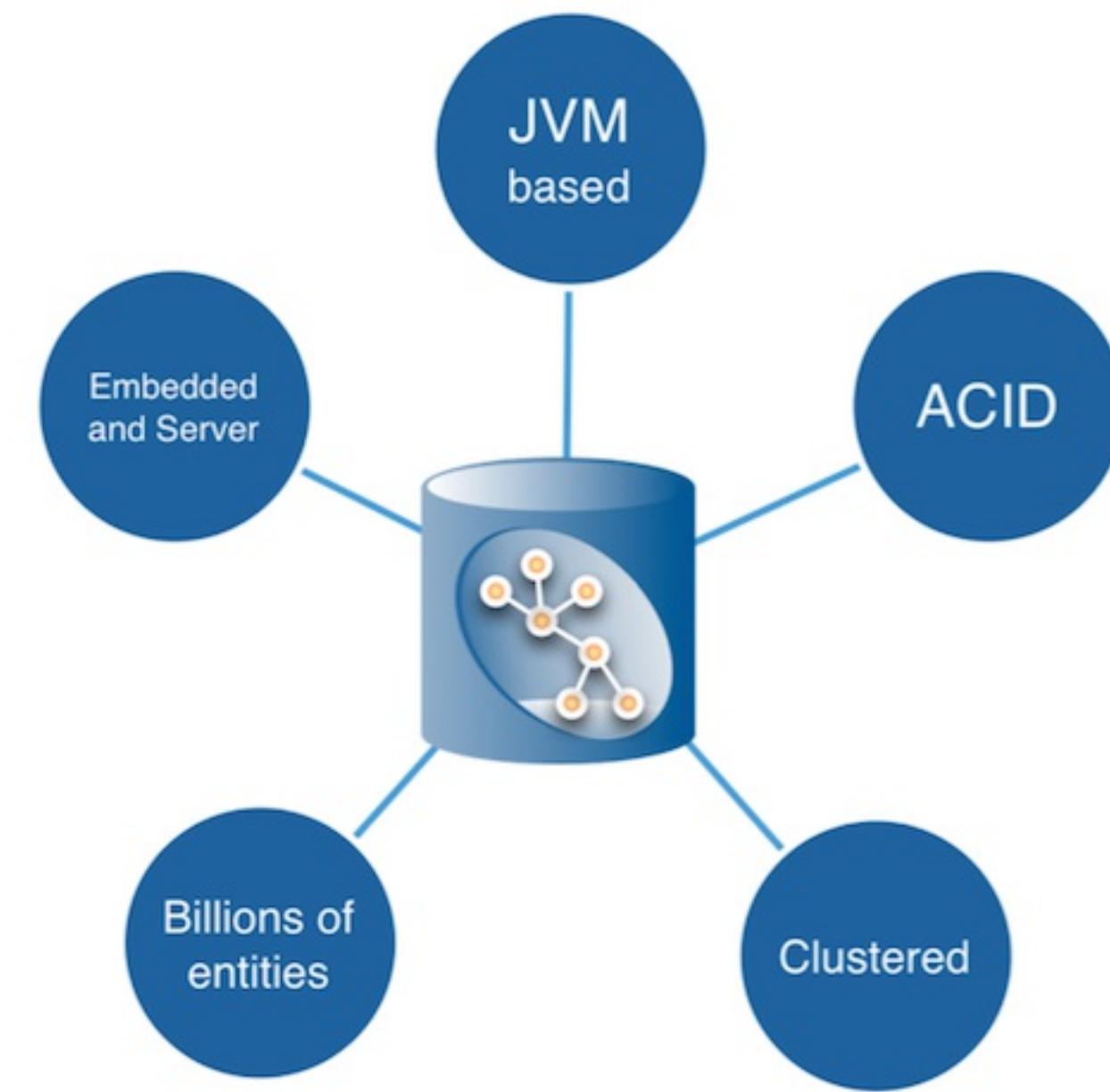
# Compléxité des données

complexité = f(**volumétrie**, semi-structure, **connectivité**)



# Neo4j

Premières lignes de code datent de **2000** et la **version 1.0** est sortie en **2010**.



## Neo4j is a NoSQL Graph Database

- Open source
- Welcoming UI
- Easy data modeling
- Readable queries
- Active community
- High performance
- optional schema

# Moteur de recommandation en temps réel

# La recommandation

Des réseaux sociaux, à la vente au détail, de service ou dans le secteur des médias, la recommandation est partout !

**Objectif :** offrir une recommandation en temps réel, hautement ciblés et contextuelles afin quelle soit pertinentes pour l'utilisateur

LES CLIENTS ONT ÉGALEMENT ACHETÉ...



Suggestions · Actualiser · Tout afficher



linkurious @Linkurious

Suivre



Philip Rathle @prathle

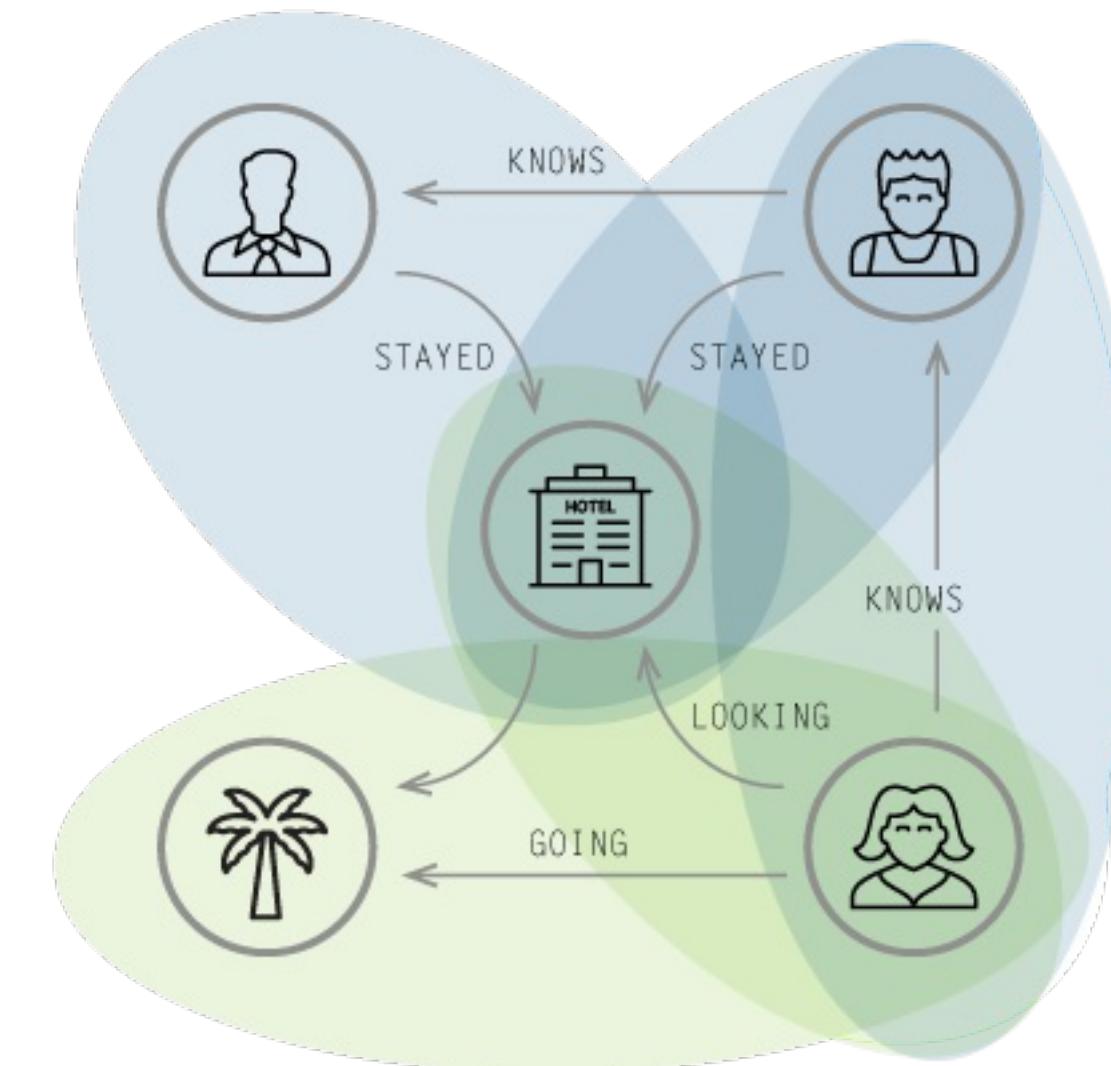
Suivre

Trouver des amis

# Pourquoi du graphe ?

Faire des recommandations efficaces repose sur une base qui comprend les relations existants entre les entités, ainsi que la qualité et la solidité de ces connexions.

**C'est un graphe !**



# Les défis liés à la recommandation

## Traiter de gros volumes de données et de relations pour comprendre le contexte

Les algorithmes standard de recommandations s'appuient sur des données hautement connectées en expansion rapide.

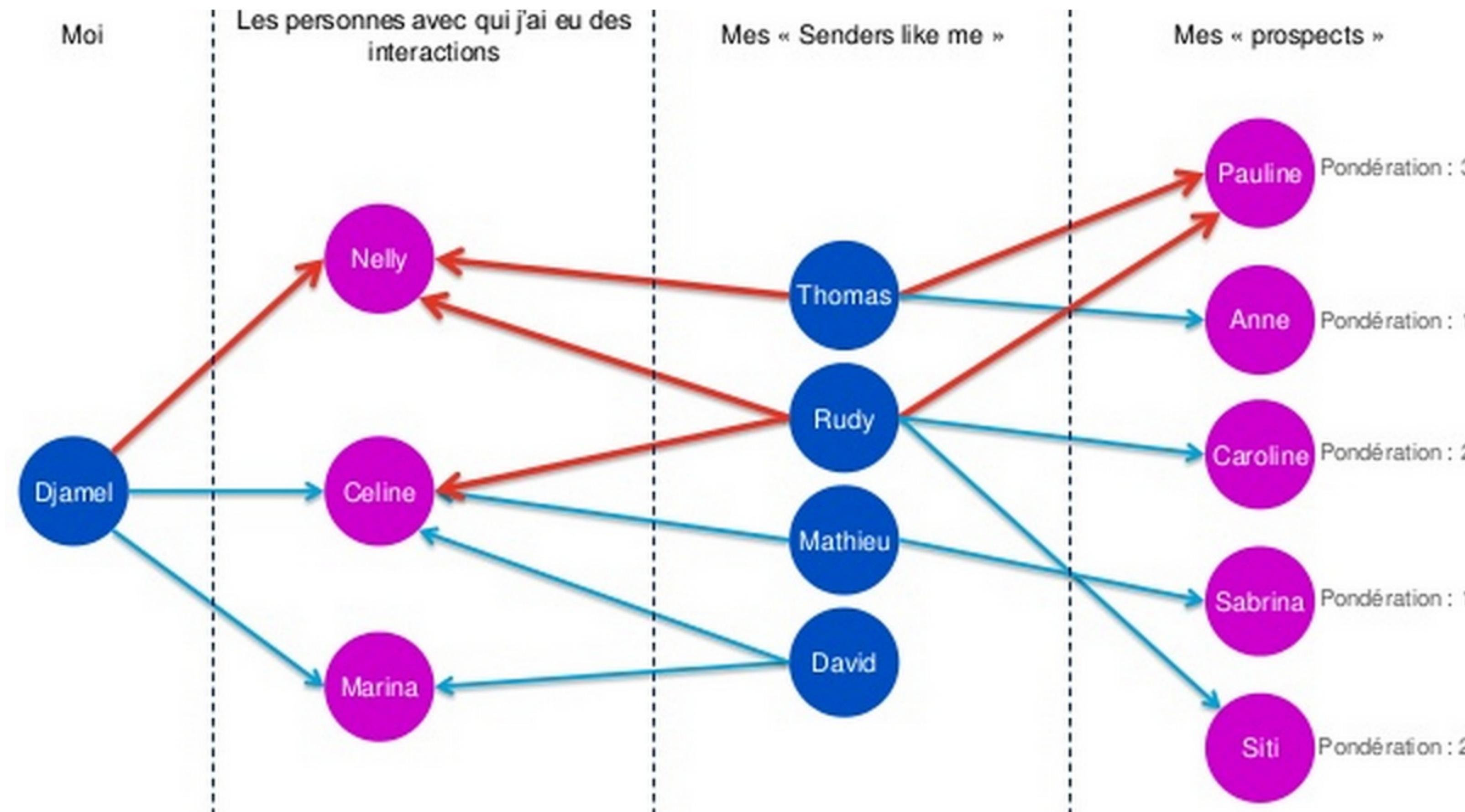
## Offrir des recommandations pertinentes en temps réel, basé sur des faits immédiats

La puissance d'un système de suggestion repose sur sa capacité à faire des recommandations en temps réel à l'aide de faits immédiats.

## Prendre continuellement en compte des données et relations nouvelles

Du fait de la croissance rapide dans la taille et le volume des données, le système doit prendre en compte à la fois les besoins actuels et futurs.

# Etude de cas : Meetic



# Etude de cas : Meetic

## Problème

- Recommandations temps-réel afin de fidéliser les utilisateurs et attirer de nouveaux membres.
- Isoler et traiter les cas des super-dragueurs et super-dragués
- A l'origine réalisé par un traitement batch écrit avec MS SQL Server
- La volumétrie : 10 millions de noeuds (ie. de membres) | 100 millions de relations (ie. d'interactions)

## Solution

- Expérience temps réel pour les utilisateurs de Meetic
- Solution scalable.
- Faible coût de maintenance et architecture fiable

# Autres clients



Fournit des recommandations en temps réel sur leur service de vente en ligne, en utilisant les informations relatives aux préférences des clients.



Le géant du commerce électronique eBay a également utilisé avec succès un moteur de suggestion fonctionnant sur un graphe, en l'occurrence pour une solution sophistiquée d'acheminement de paquets/courrier en temps réel (Ebay now).

# La détection de fraudes

# La détection de fraudes

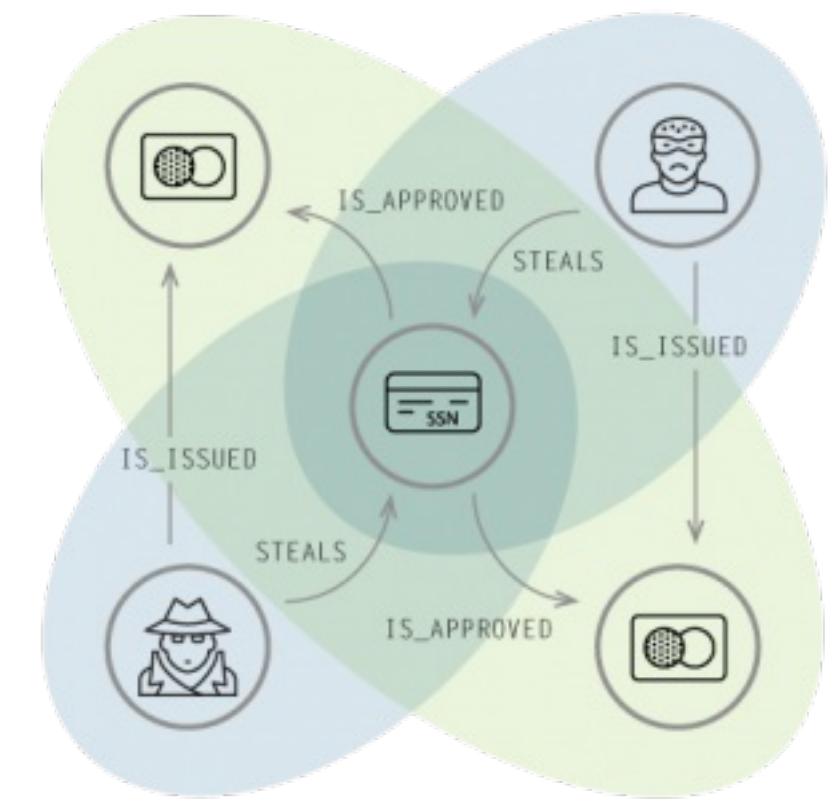
- La fraude coûte des milliards chaque années aux banques et assurances
- Les techniques de fraudes sont de plus en plus sophistiquées
- Il faut constater la fraude en temps réel (ie. avant qu'elle est lieu)



# Pourquoi du graphe ?

Bien qu'aucune mesure de détection de fraudes ne soit parfaite, il est possible d'introduire des améliorations considérables si l'on se penche sur les connexions qui relient les points de données individuels.

**Il faut étudier l'interconnexion des données**



# Les défis liés à la détection de fraudes

## **Effectuer une analyse complexe des liens pour découvrir les schémas de fraude**

Découvrir les réseaux de fraude nécessite d'exécuter une analyse d'une complexité élevée sur les relations existant entre les données ; une difficulté exacerbée par la croissance des réseaux de fraude.

## **Déetecter et empêcher les fraudes en temps réel**

Pour faire barrage à un réseau de fraude, vous devez effectuer une analyse des liens en temps réel au niveau d'un jeu de données interconnectées, à partir de la création d'un faux compte jusqu'à l'exécution d'une transaction frauduleuse.

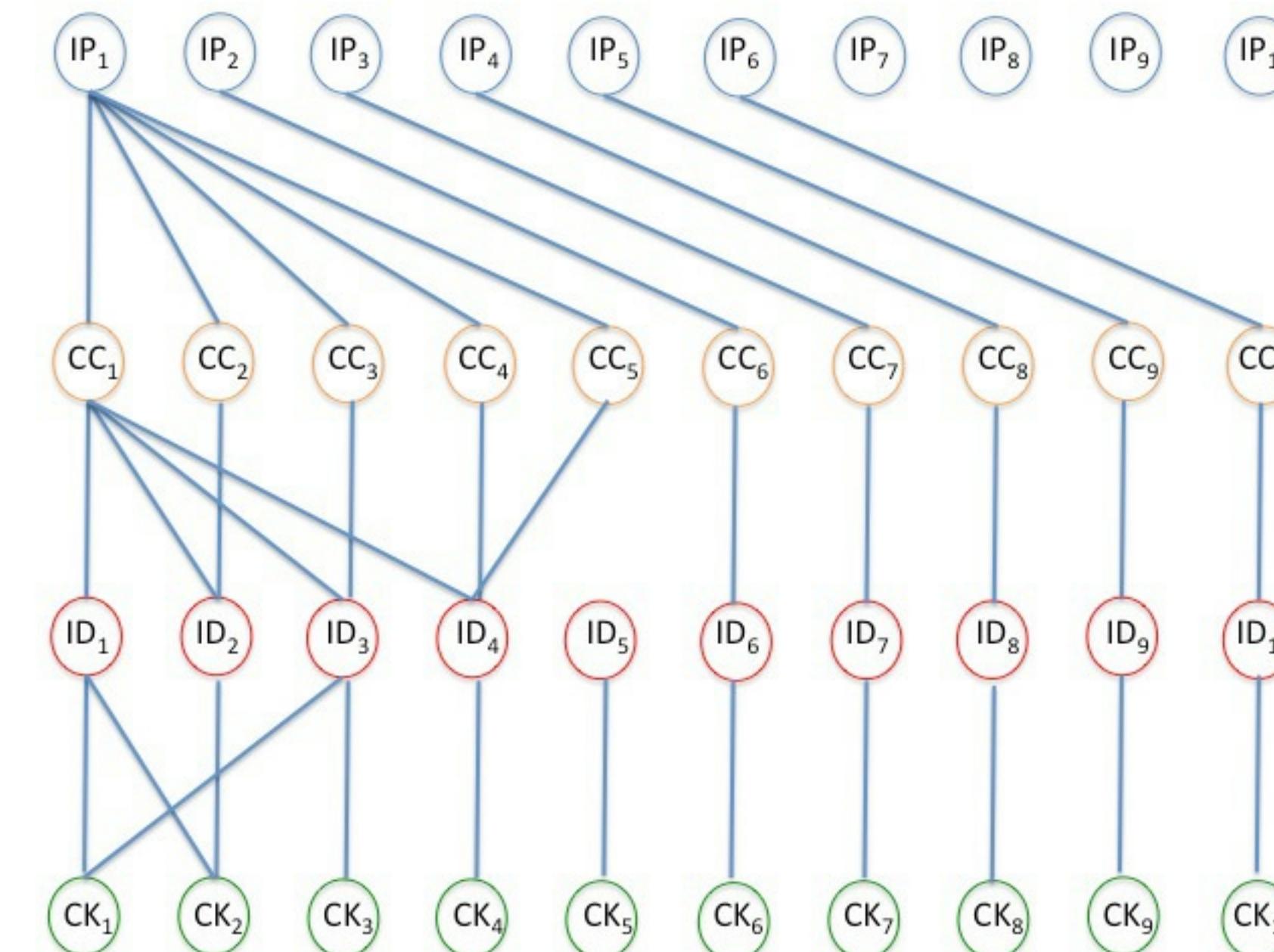
## **Réseaux de fraude en évolution et dynamiques**

Les réseaux de fraude se développent continuellement en forme et en taille et votre application doit détecter ces schémas de fraude dans cet environnement hautement dynamique.

# Etude de cas : Fraude dans le e-commerce

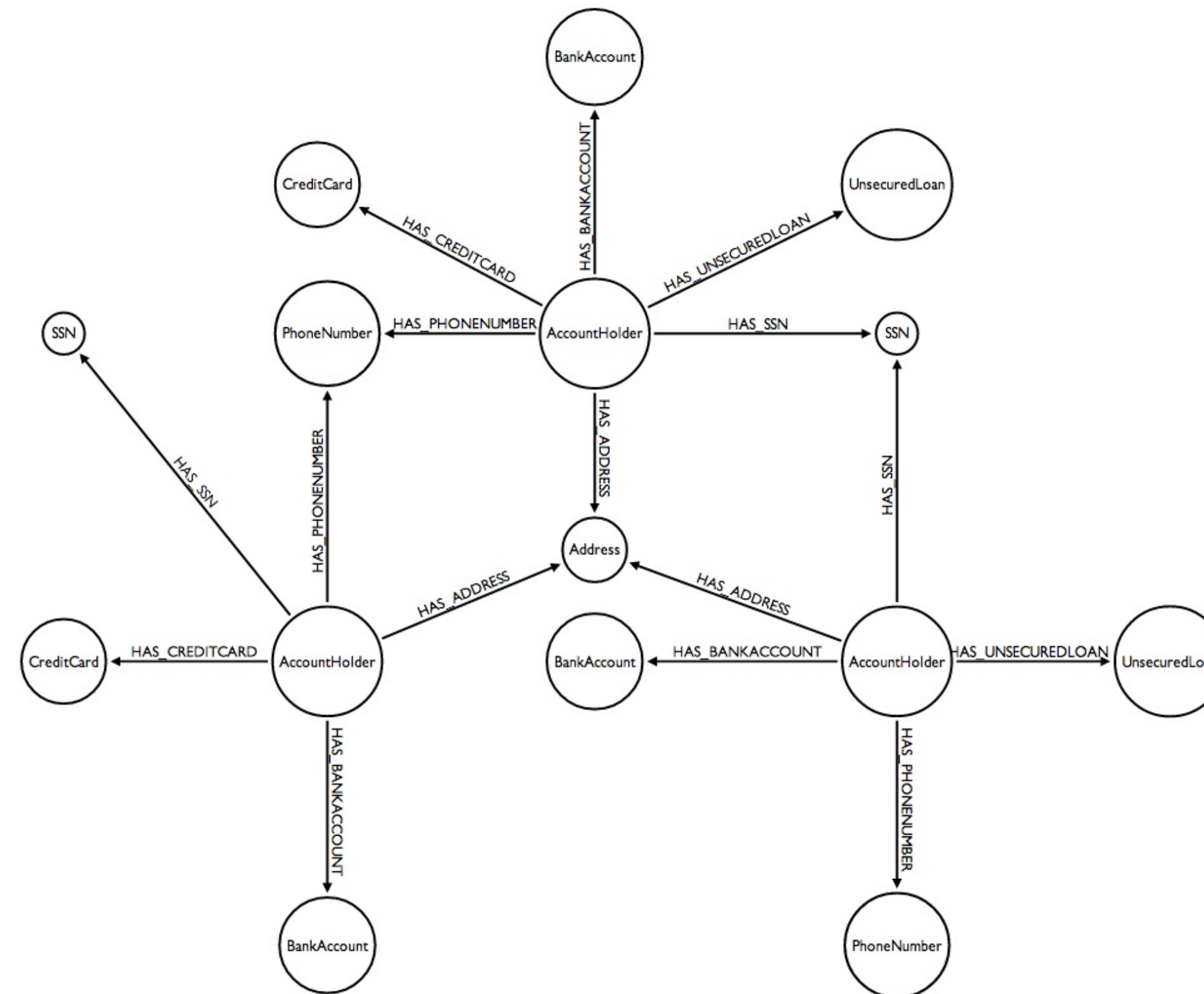
Graphe d'une série de transactions effectuée avec fraude probable se produisant à l'adresse IP1.

( IP : Adresse IP, CC : carte de crédit, ID : Identifiant de l'utilisateur, CK : Adresse de livraison)



# Etude de cas : La détection d'anneau

Constituer un ensemble de fausses identités à partir d'identités volées, dans l'objectif de commettre un acte frauduleux



# **La gestion de données de référence**

# La gestion de données de référence

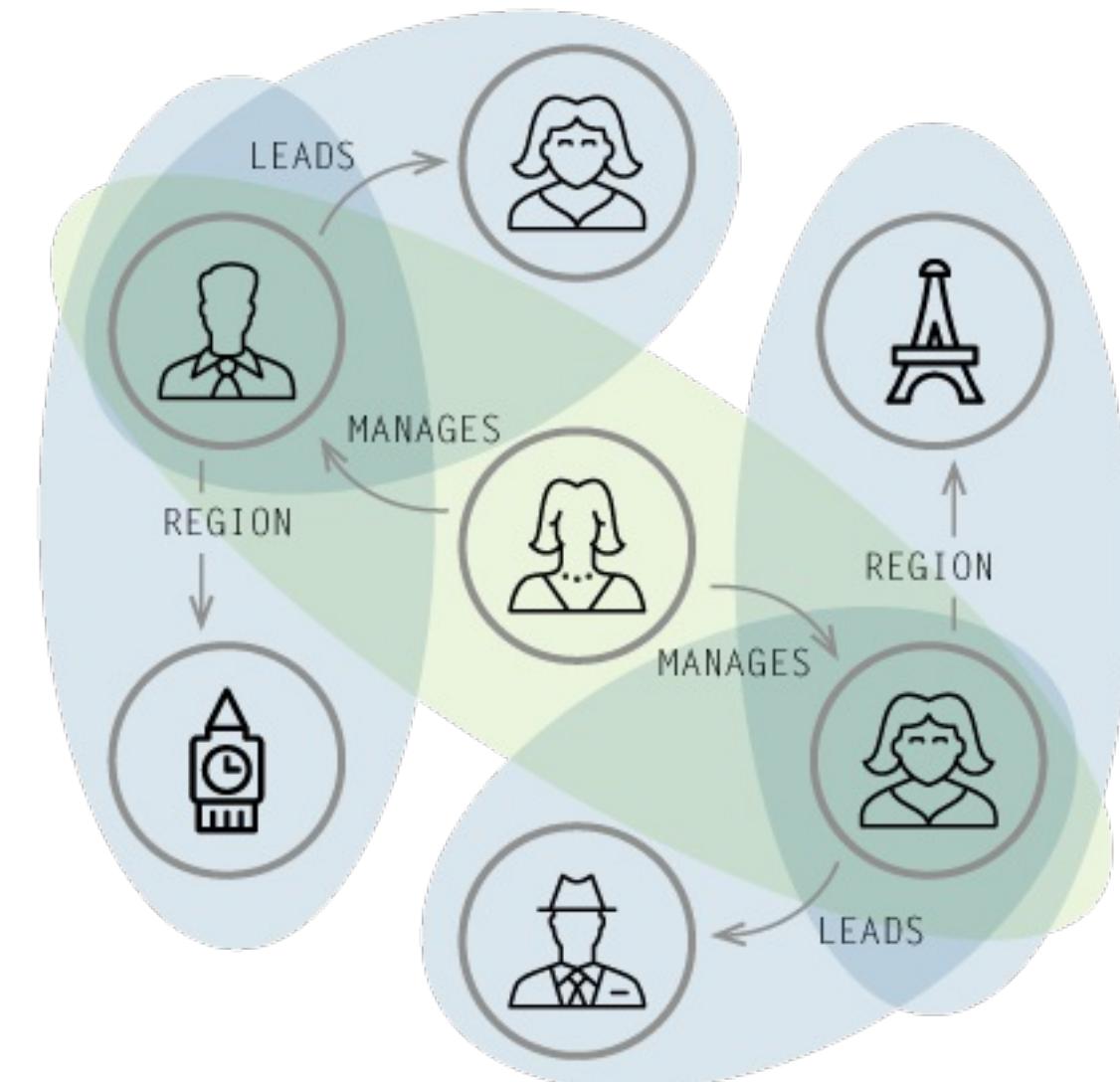
Les données de référence constituent le pilier de votre entreprise et incluent des données telles que les utilisateurs, les clients, les produits, les comptes, les partenaires, les sites, les unités commerciales, ...

Ces données sont :

- stockées à différents emplacements
- avec des chevauchements et des redondances
- dans divers formats
- avec des qualités variables

# Pourquoi du graphe ?

- Les données sont hautement connectées et partagées
- Les systèmes de gestion ne sont pas flexibles
- L'interconnexion des données de références permettent d'obtenir un avantage concurrentiel



# Les défis liés au MDM

## Jeux de données complexes et hiérarchiques

Gérer les hiérarchies descendantes des données de référence avec une base de données relationnelle se traduit par un code complexe lent à exécuter, cher à concevoir et long à maintenir.

## Performances de stockage et de requête en temps réel

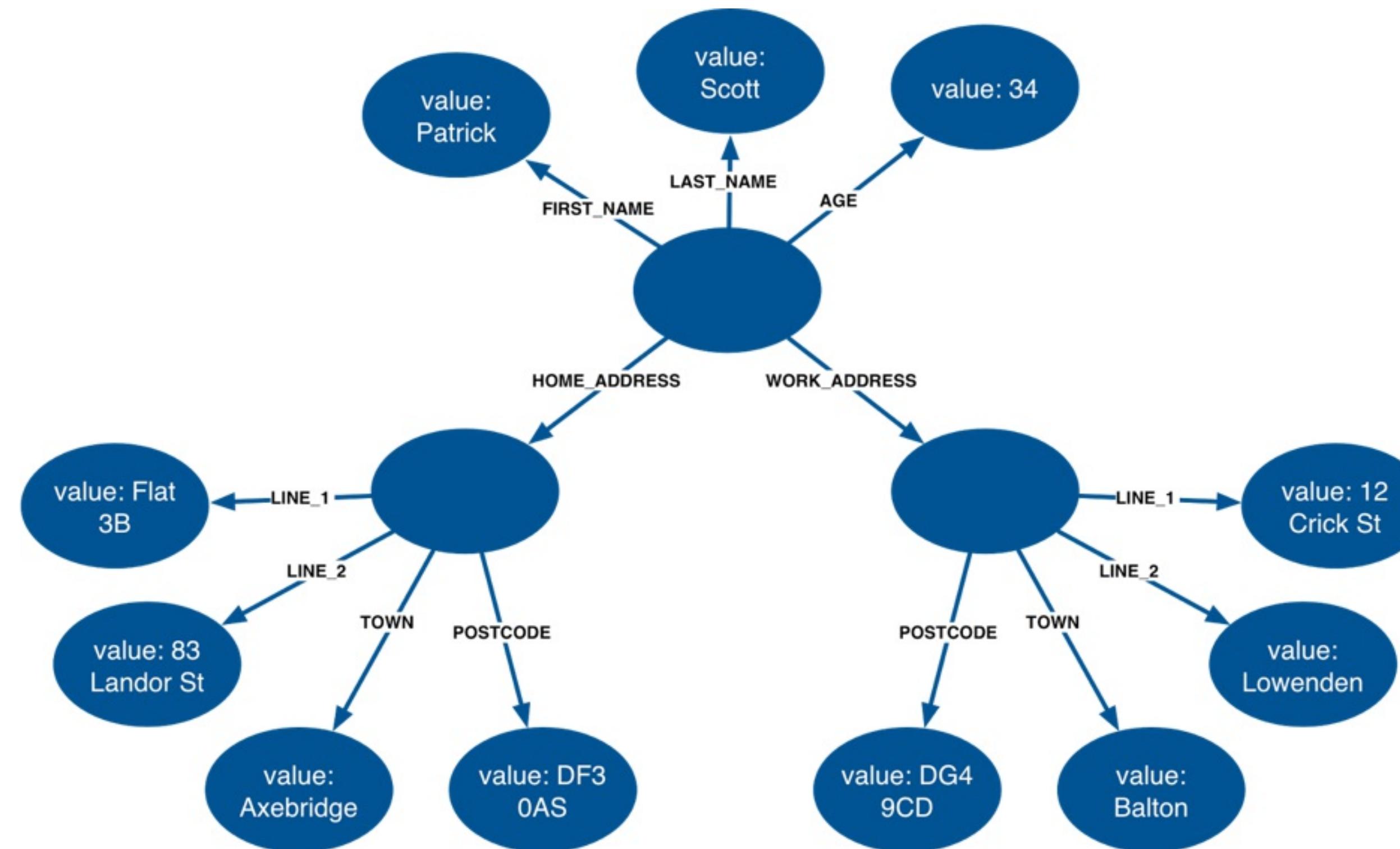
Le magasin de données de référence doit s'intégrer à une multitude d'applications au sein de l'entreprise et fournir des données à ces applications. Fournir des informations en temps réel sur des jeux de données complexes et hautement interconnectés est un véritable challenge.

## Structure dynamique

Les données de référence étant dynamiques par nature, il est plus difficile pour les développeurs de concevoir des systèmes qui prennent en compte leur évolution.

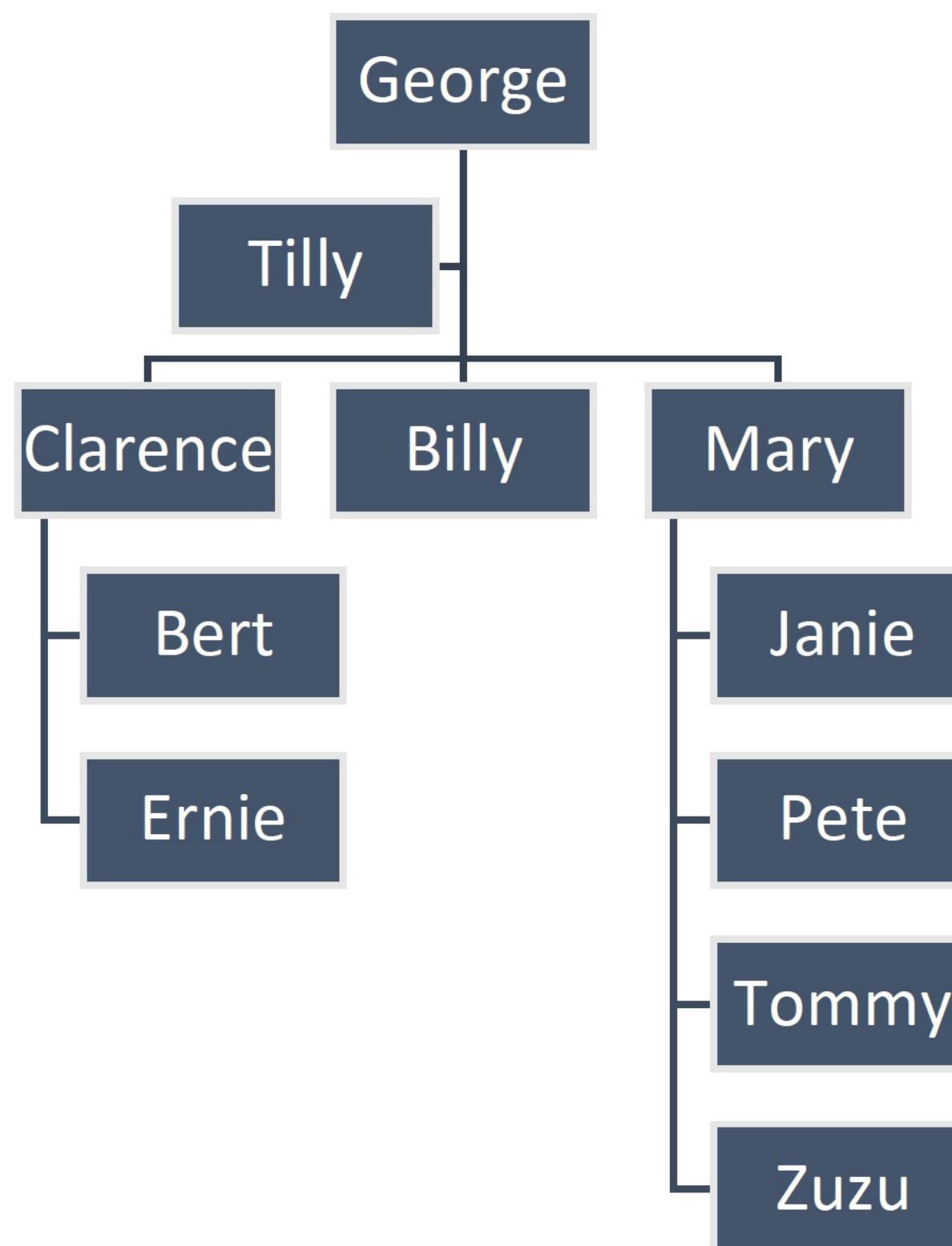
# Etude de cas : Reconciliation des données

Réconciliation des données par degré de similitude



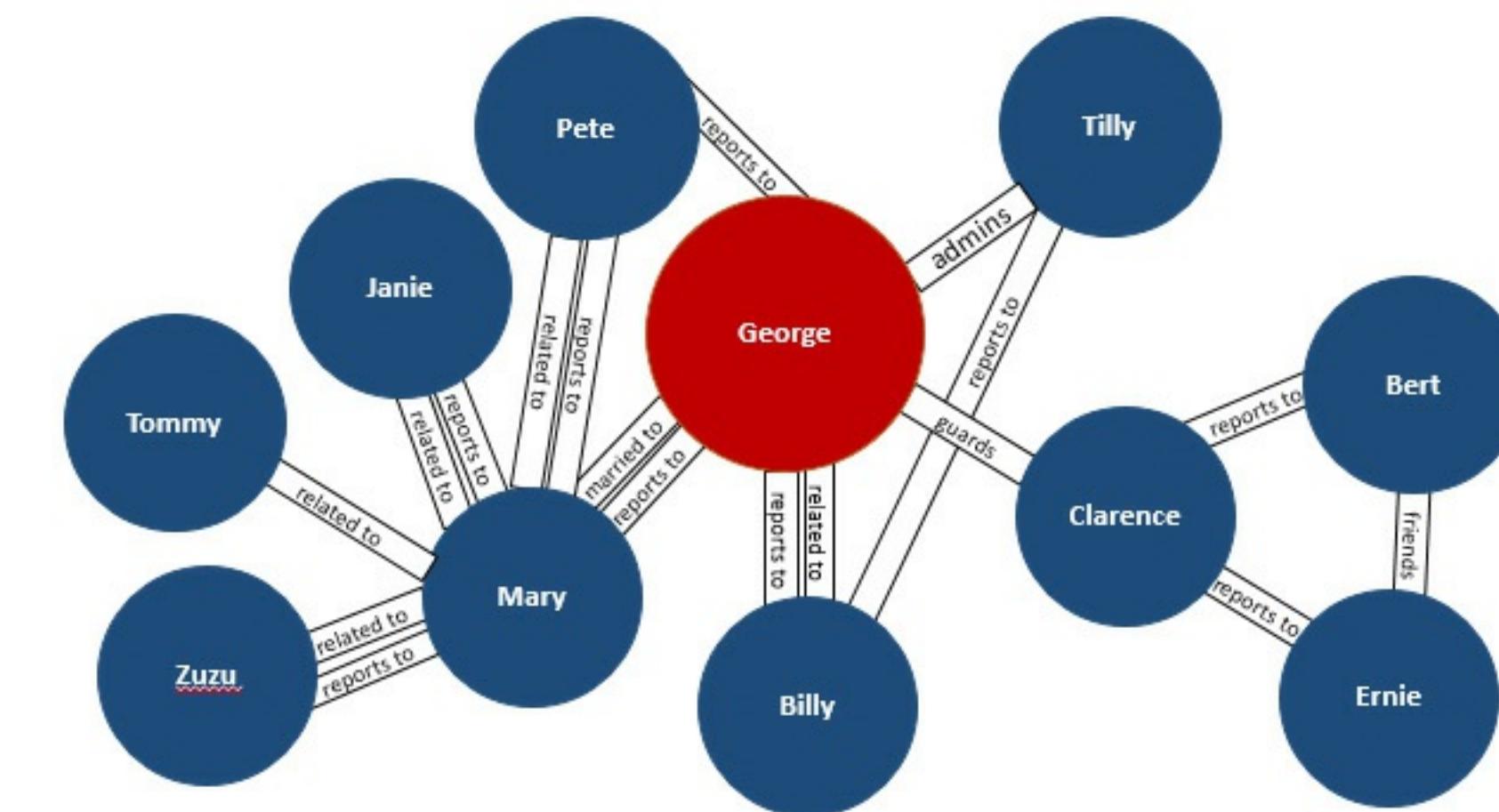
# Etude de cas : La modélisation RH d'une entreprise

Hiérarchie de données de référence illustrant des structures de rapport et de supervision d'employé.



# Etude de cas : La modélisation RH d'une entreprise

Réseau de données de référence détaillant les relations de rapport et de supervision d'employé, ici avec une plus grande complexité du **monde réel**.



# Client



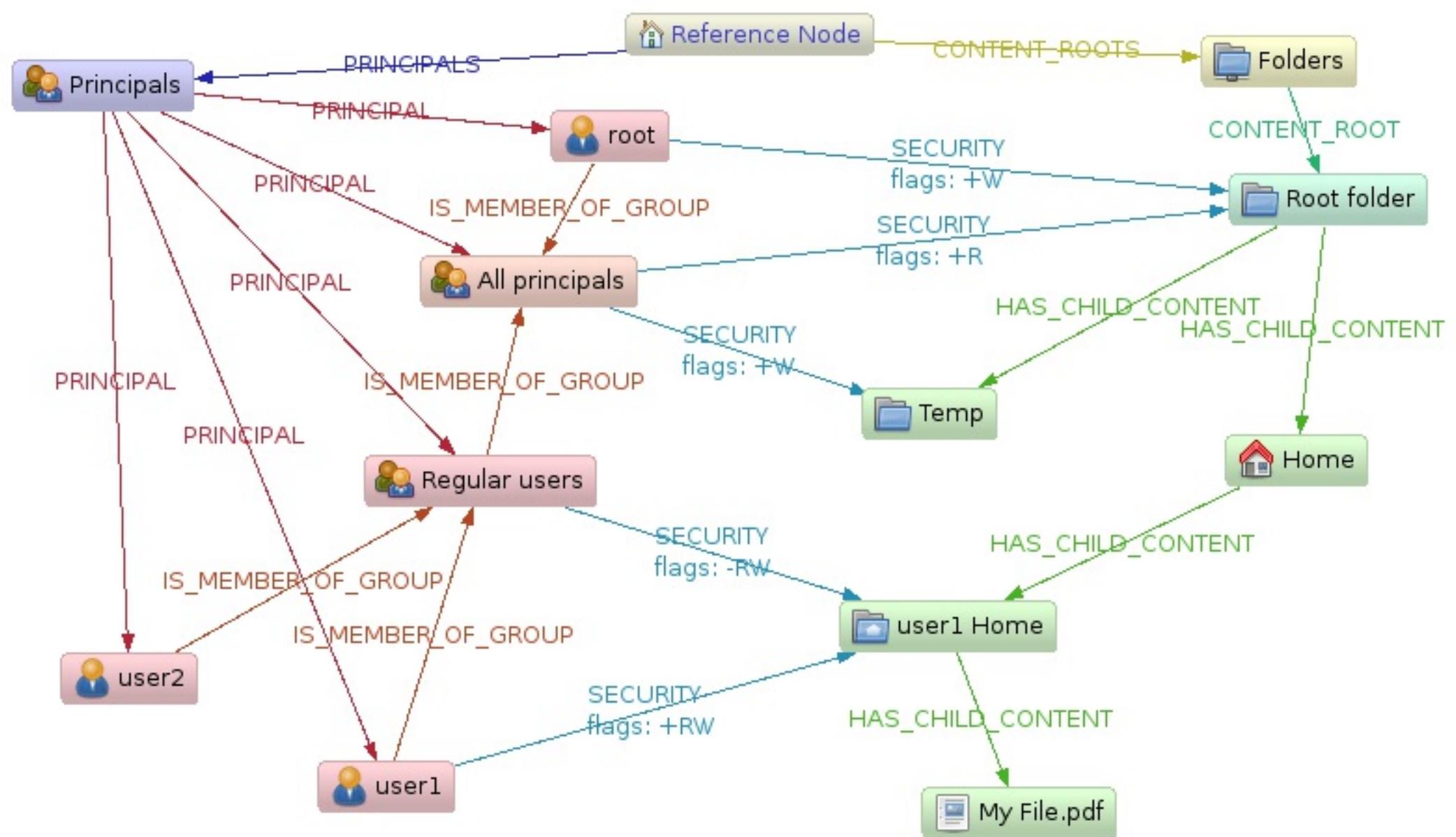
Adidas utilise Neo4j pour créer leur référentiel interne, incluant les données sur les produits, les marchés, les médias sociaux, les actifs numériques, ...



Wine Data System utilise Neo4j pour réconcilier les bases de données des négociants afin de construire un référenciel unique.

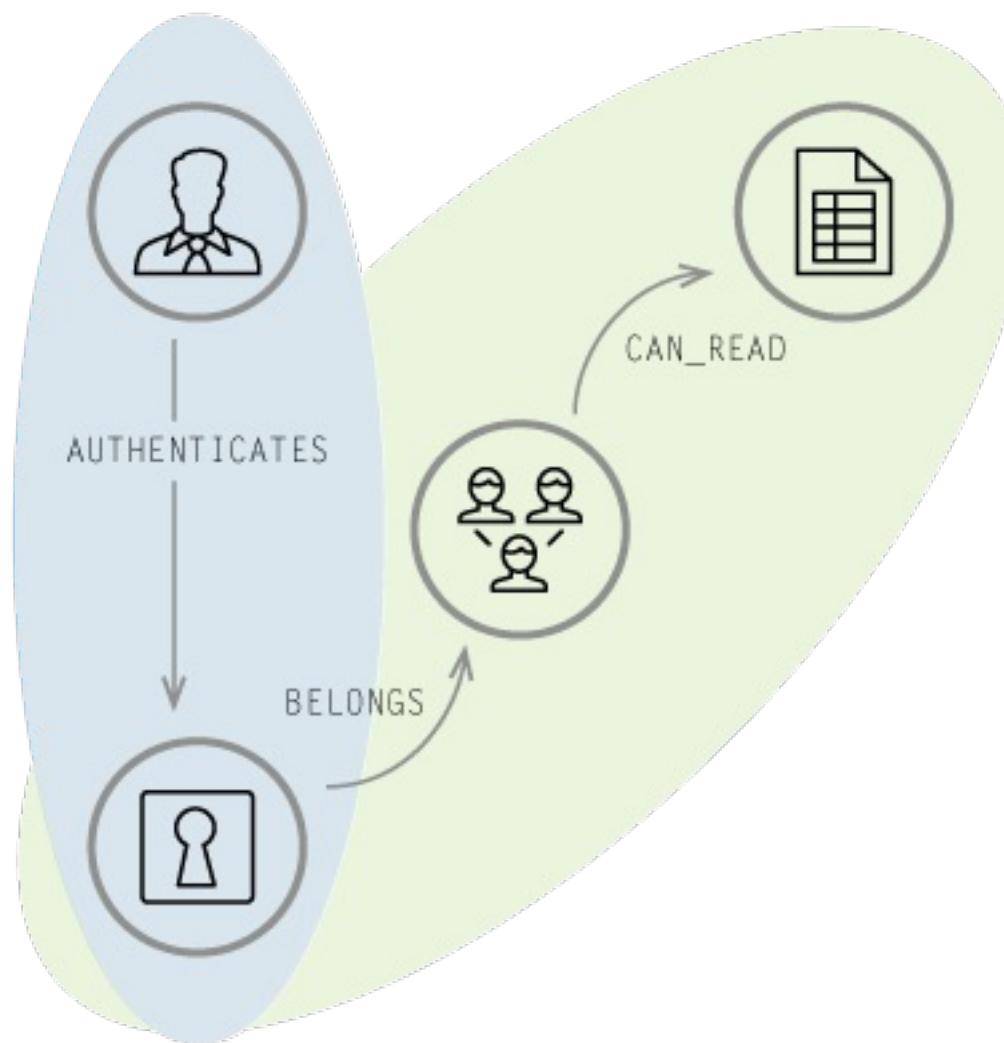
# **La gestion des identités et des accès**

# La gestion des identités et des accès



# Pourquoi du graphe ?

C'est un graphe !



# Les défis liés à la gestion des identités / accès

## Autorisations d'identité et d'accès hautement interconnectées

Pour vérifier une identité précise et ses autorisations d'accès, le système doit traverser un jeu de données hautement interconnectées dont le volume et la complexité augmentent en permanence.

## Productivité et satisfaction du client

À mesure que le volume d'utilisateurs, de produits et d'autorisations augmente, les systèmes traditionnels n'assurent plus de réponses réactives, ce qui se traduit par une frustration chez les utilisateurs.

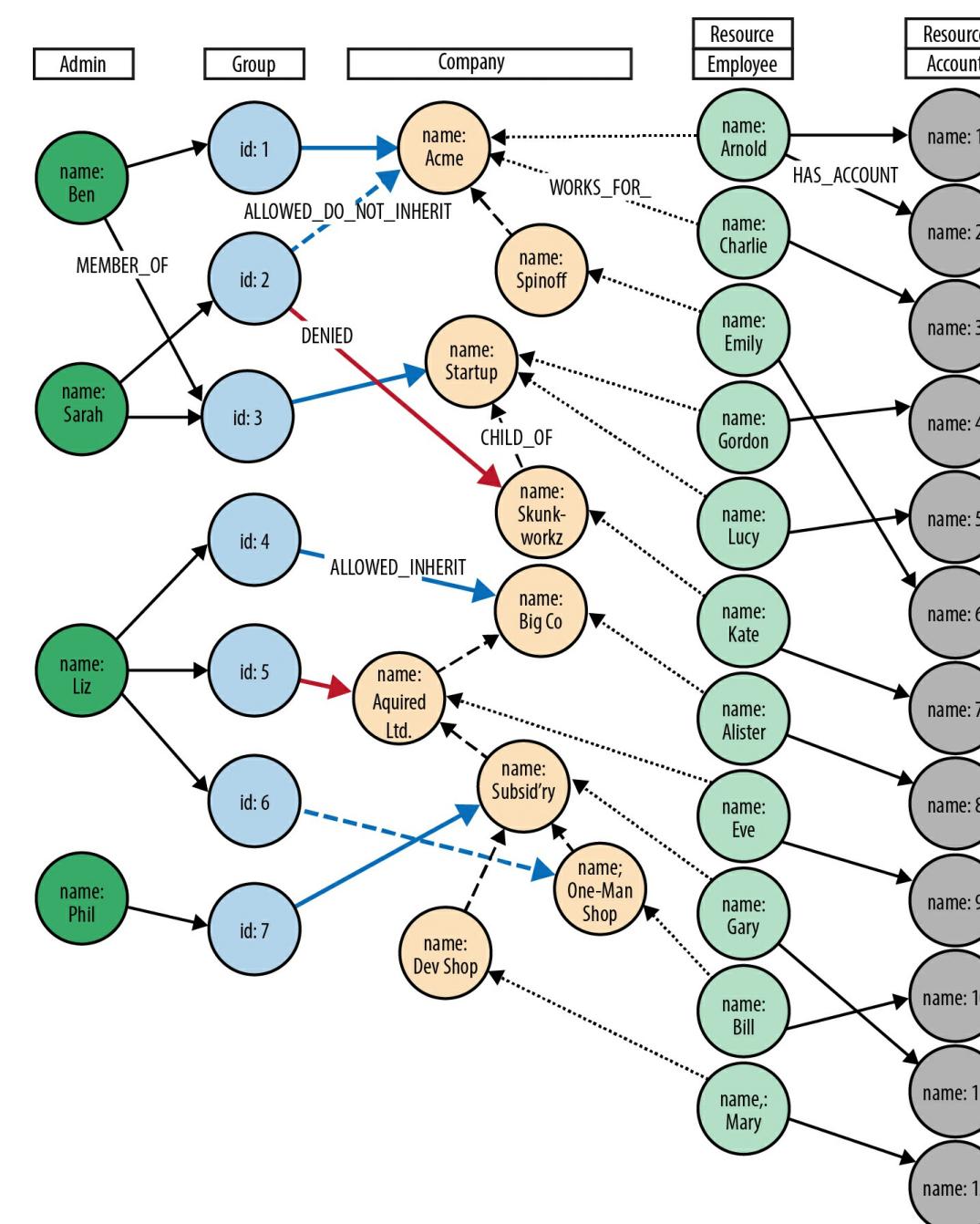
## Structure et environnement dynamiques

Avec l'augmentation rapide de la taille des utilisateurs et de leurs métadonnées associées, votre application doit prendre en compte les besoins de gestion des identités à la fois actuels et futurs.

# Etude de cas : Telenor

Telenor permet à ses clients entreprises d'utiliser leurs comptes pour ajouter ou supprimer des services à leurs employés.

Ainsi il faut garantir que les utilisateurs et les administrateurs ne voient et ne modifient que les parties de l'entreprise et les services qu'ils ont le droit de gérer.



# Etude de cas : Telenor

## Définition du problème

- Besoin d'un contrôle d'accès fiable pour 5 millions de clients, souscriptions et accords.
- Dépendances complexes entre les groupes, sociétés, individus, comptes, produits, souscriptions, services et accords

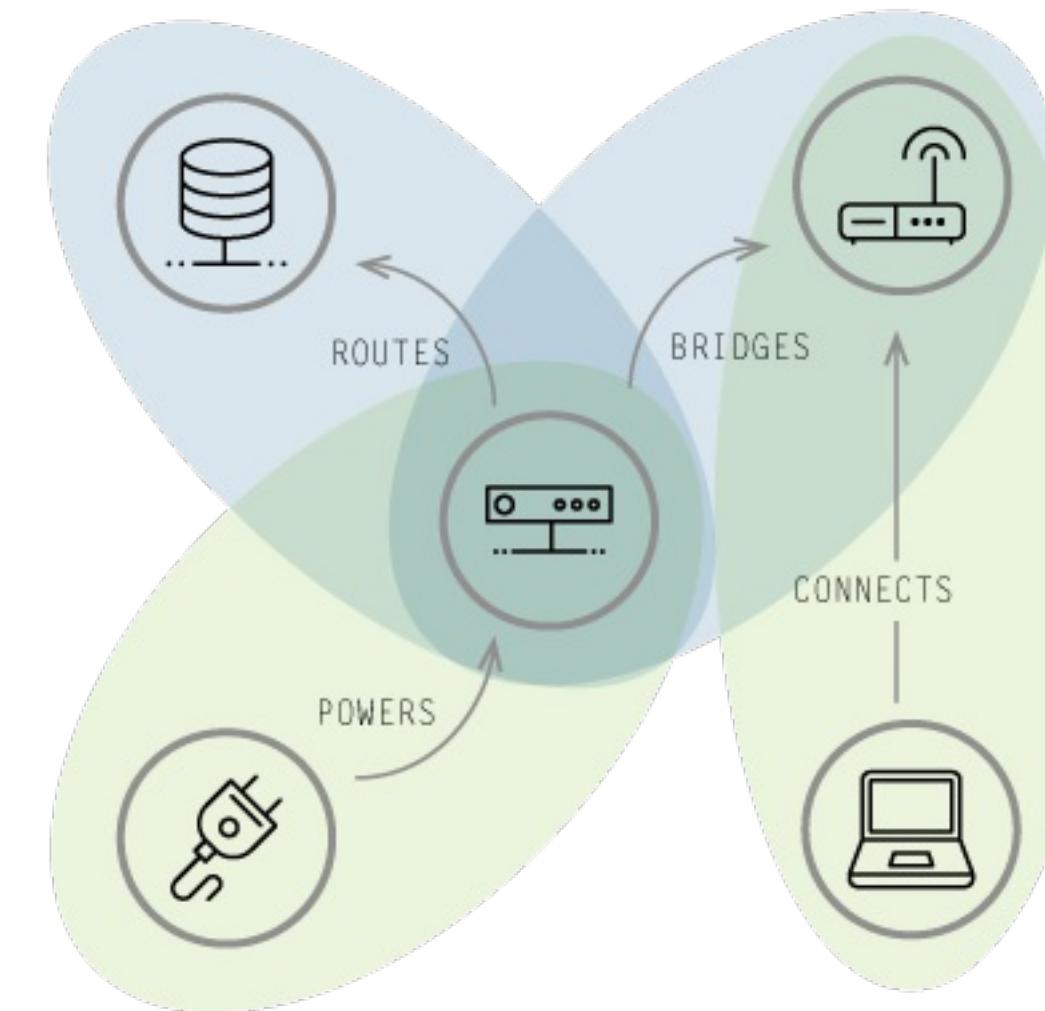
## Bénéfices

- Architecture flexible et dynamique.
- Performance exceptionnelle.
- Bas coût comparé aux autres alternatives.
- Modèle de données extensible supportant l'ajout de nouvelles applications et fonctionnalités.

# Les réseaux informatiques

# Les réseaux informatiques

Un réseau est un graphe !



# Pourquoi du graphe ?

- Modélisation/stockage naturelle
- Meilleur visualisation et apprehension des données
- Permet de réaliser des analyses d'impacts en cas de panne
- A l'inverse, voir les redondances du réseau

# Les défis liés aux réseaux

## Résolution des problèmes sur un réseau

Les interdépendances physiques et humaines sont extrêmement complexes dans un réseau ou un environnement informatique, ce qui rend difficile la résolution des problèmes.

### Analyse d'impact

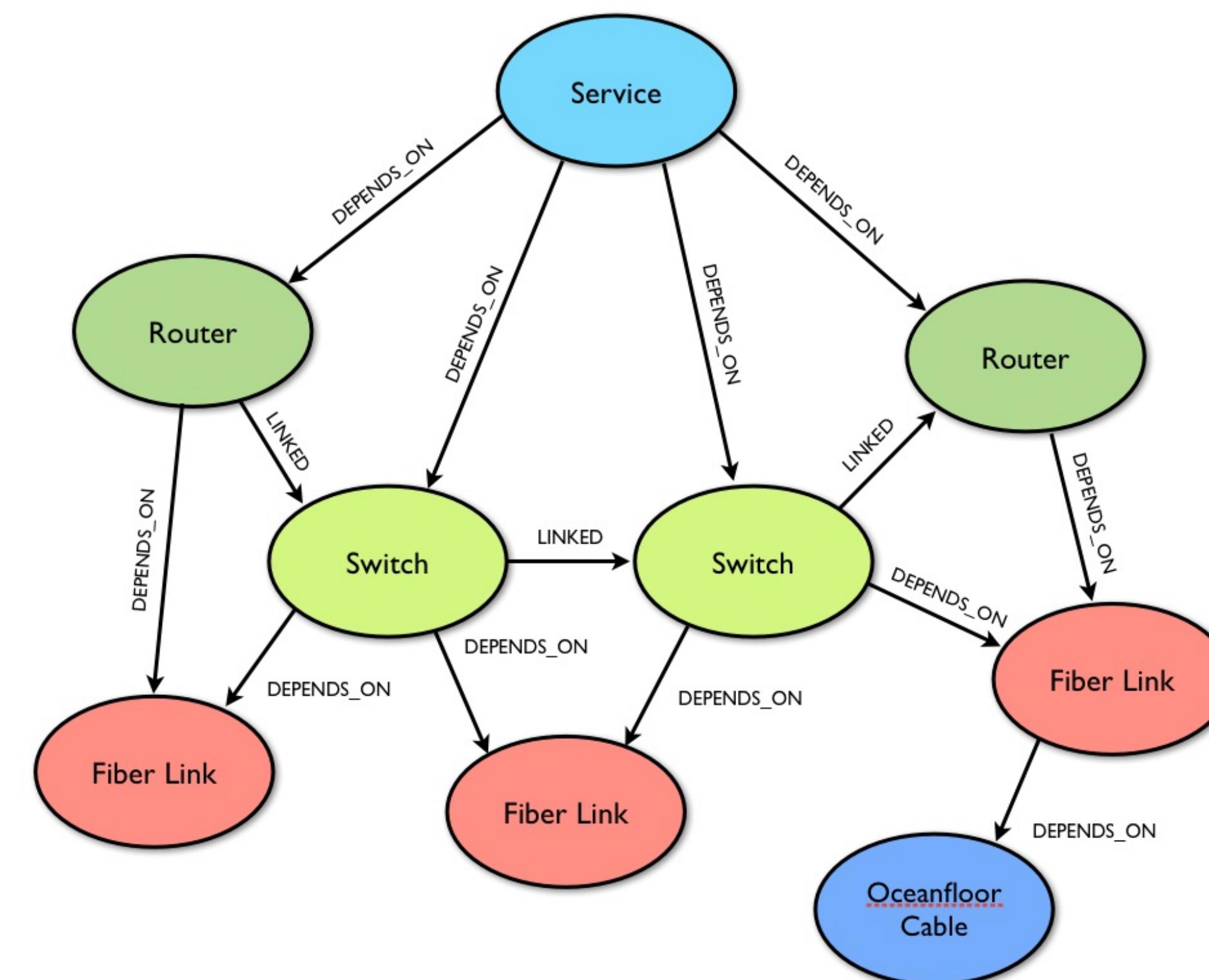
Les relations existant parmi les noeuds d'un réseau ne sont ni purement linéaires ni hiérarchiques et il est donc difficile de déterminer les interdépendances des éléments du réseau les uns envers les autres.

### Nombre croissant de noeuds physiques et virtuels

Du fait de la croissance rapide dans la taille et le volume des données, le système doit prendre en compte à la fois les besoins actuels et futurs.

# Etude de cas : Analyse d'impact chez SFR

# Modélisation de leur réseau physique dans Neo4j.



# Etude de cas : Analyse d'impact chez SFR

## Problème

- Une semaine entière pour planifier une maintenance.
- Analyste d'impact sur de multiples systèmes (>30).
- Changements quotidiens dans l'infrastructure.
- Identifier les clients impactés par une maintenance.
- Identifier les faiblesses du réseau (Deuxième projet).
- Une semaine entière pour connaître le noeud de raccordement d'un nouveau client (Troisième projet).

# Etude de cas : Analyse d'impact chez SFR

## Bénéfice

- Identification temps réel des clients liées à une panne.
- Gestion d'inventaire réseaux très flexible.
- Une seule source de vérité (Neo4j) représentant le réseau entier.
- Système dynamique de chargement (ETL) des données issues de +30 sources variées.
- Mise en application rapide des changements métiers.

# **Graphs are everywhere !**



# GraphConnect Europe

**20% de réduction avec le code NEWS20 !!!**

Mardi 26 Avril, 2016, Queen Elizabeth II Centre, Londres

- 3 Thématiques + Lightning Talks
- GraphClinics: Rencontrez nos experts
- Networking

Formation, Lundi 25 Avril 2016, 9h30-17h30 : Fundamentals, Modelling, Neo4j in Production,  
Creating a Recommendation Engine



# Des questions ?

- **Twitter:** Suivez les comptes @neojFr & @neo4j
- **Google group :** Avec les groupes Neo4jFr & Neo4j
- **Stackoverflow :** avec les tags neo4j & cypher

