

第3回 データ未来会議 — 2026.3.18

つながるを 支える技術

海外で利用加速中の「グラフDB」とは何か

株式会社UTI | 細井 公義

SPEAKER



細井 公義

Kimiyoshi Hosoi

株式会社UTI 代表取締役

DSA会員

経歴

東京大学 理学部 (物理学科) 卒

日本生命

モルガンスタンレー

ソシエテジェネラル

2006年 UTI設立 (福岡)

Neo4j — 約10年

Authorized Distributor

導入実績:

電力

通信

自動車

ナレッジマネジメント

IPA 受託実績

行政分野のデータ辞書整備に関する調査

最近の注力:

グラフDB × LLM → GraphRAG

EVERYDAY PROBLEM

代表取締役が引っ越した

山田太郎さんは港区から渋谷区へ引っ越し、住民票を変更した。

📄 住民票テーブル

氏名	住所
山田太郎	港区六本木1-1

📄 法人登記テーブル

法人名	代表者	代表者住所
株式会社X	山田太郎	港区六本木1-1

同じ「住所」が、各テーブルに**コピーとして保持**されている。片方を変えても、もう片方は変わらない。

忘れれば登記懈怠 → 100万円以下の過料

なぜ、2回同じことをしなければならないのか？

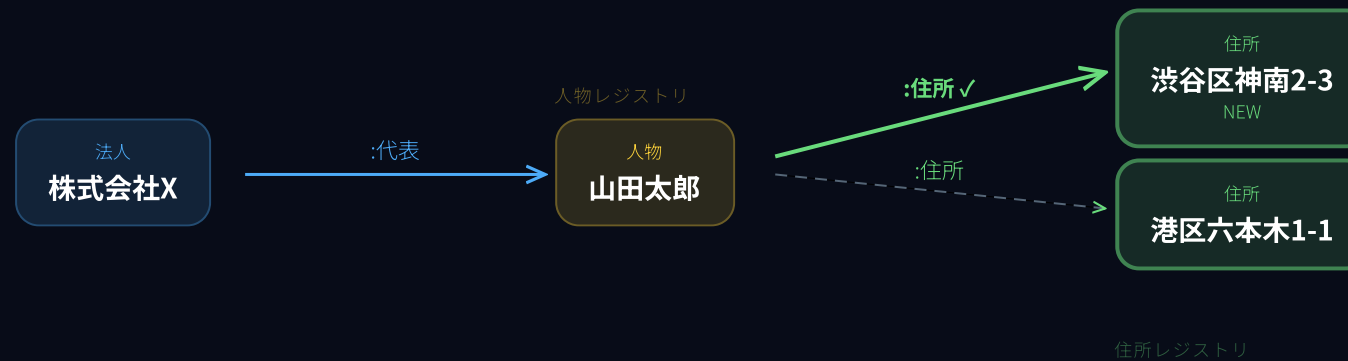
WHAT IF?

グラフ屋の**夢**想

WHAT IF?

もしデータが「参照」で つながっていたら

企業の代表者は人物レジストリへの参照、個人の住所は住所レジストリへの参照

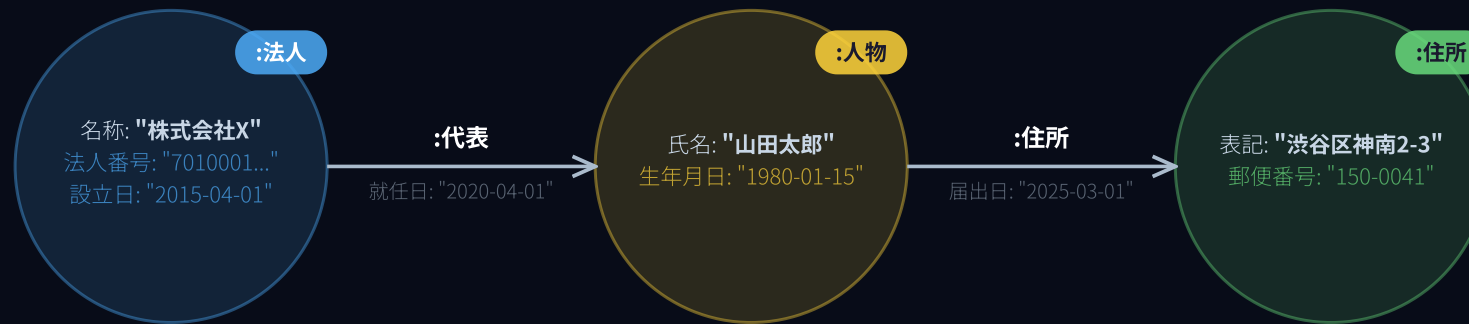


これが、グラフ的な考え方

コピーではなく参照でつなぐ

PROPERTY GRAPH MODEL

プロパティグラフモデル



これがプロパティグラフモデル

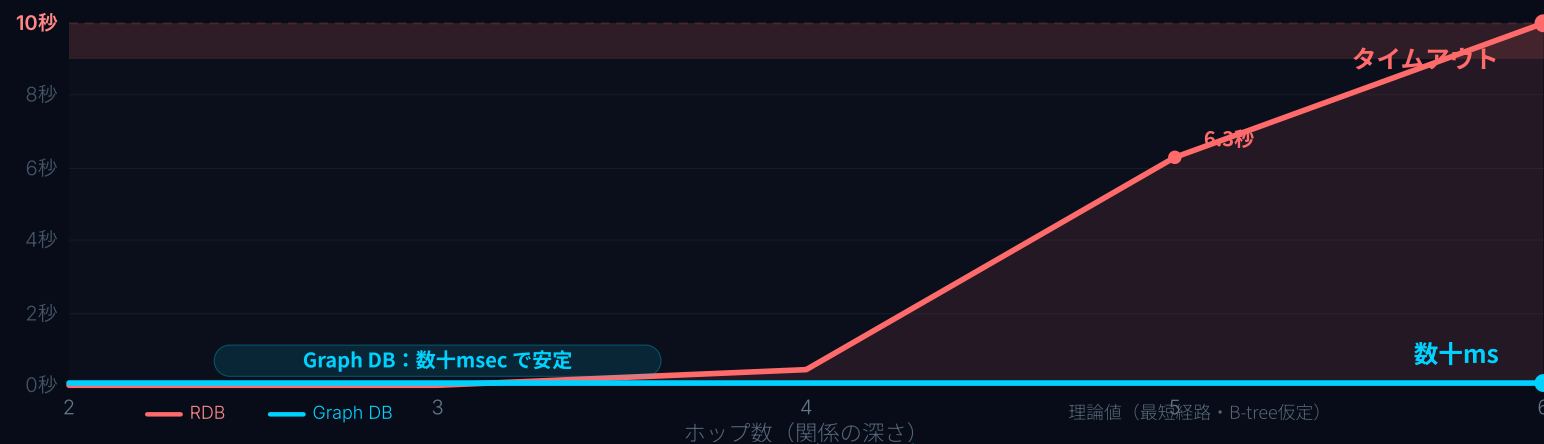
ノード+リレーションシップ+ラベル+プロパティ = グラフDBの基盤

GRAPH DATABASE

グラフデータベース

プロパティグラフをそのまま保存・検索するデータベース

最短経路の検索 — 法人5万 / 個人15万 / 関係300万 (B-tree インデックス仮定・理論値)



RDBだと難しい検索が、グラフDBなら一瞬

最短経路・到達判定・パターンマッチ…関係を辿る検索はグラフDBの得意領域

USE CASES

グラフDBが真価を発揮する領域

不正検知・AML

共有属性や共起パターンで
不正グループを検出

推薦エンジン

「買った人はこれも」＝
多段の嗜好探索

ナレッジグラフ

概念間の関係を辿り
推論・発見を実現

ネットワーク・IT運用

障害時の影響範囲を
リアルタイムに特定

サプライチェーンとBOM

部品→完成品の多段追跡
リコール影響分析

ルート最適化・物流

最短経路・配送計画
リアルタイム再ルーティング

アクセス制御・認可

グループ→ロール→権限の
継承チェーンを探索

データクレンジング

名寄せ・参照整合性
孤立データの検出

共通点：「関係を辿る」ことが本質の問題

多段ホップ・パターン検出・最短経路 — RDBのJOINでは限界がある領域

不正検知・AML

共有属性による不正グループ特定

別人名義の口座が同一住所・同一端末・同一IPを共有 — 個別に見れば正常だが、グラフで繋げると不自然なクラスタが浮かぶ。



保険金詐欺ネットワーク

無関係の複数の請求者が、同一医師・同一修理業者に集中 — 偶然にしては不自然な共起パターンをグラフで検出。



共有属性の発見も共起パターンの検出も、本質は「経路探索」と「パターンマッチ」。個別のレコードでは見えない不自然なつながりを、グラフで俯瞰して初めて発見できる。

推薦エンジン

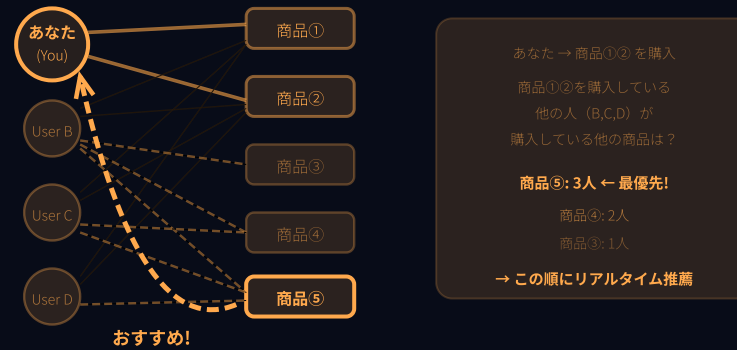
協調フィルタリング

「この商品を買った人はこれも買った」 = ユーザー→商品→ユーザー→商品の2ホップ探索。グラフDBなら購入の瞬間にリアルタイムで推薦を更新できる。



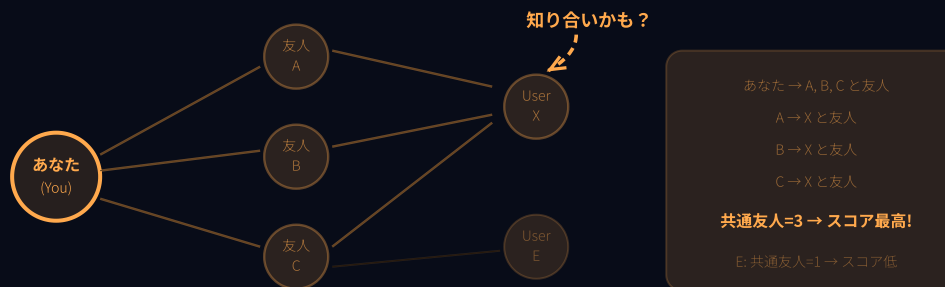
ソーシャル推薦

LinkedIn 「知り合いかも」 = 2ホップの共通コネクション数でランキング。



コンテンツ推薦

ジャンル・タグ・出演者の共有関係を辿り、類似コンテンツを発見。

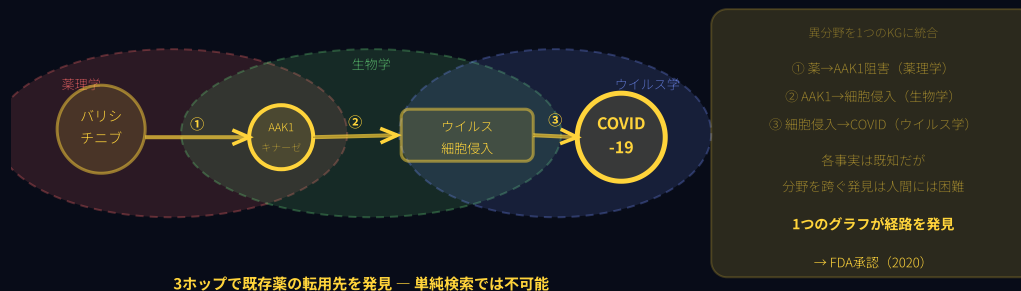


推薦の本質は「似た人が好きなもの」の探索。グラフの隣接関係をそのまま辿れるため、リアルタイム推薦が実現できる。

ナレッジグラフ

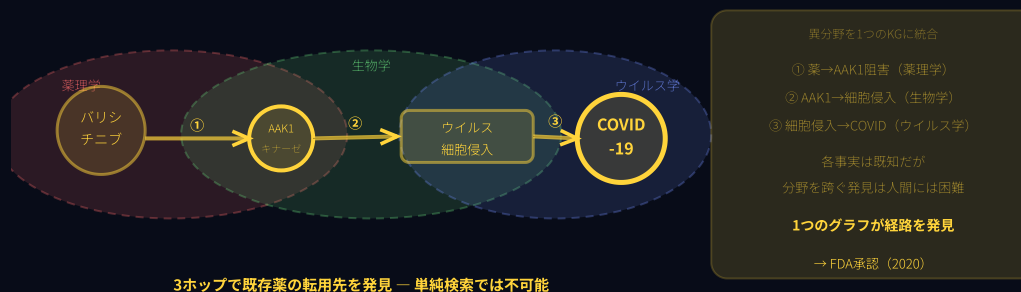
創薬 — 実例

BenevolentAI (2020) — 生物医学KGの多段探索で、関節リウマチ薬パリシチニブのCOVID-19転用を発見。FDA承認。



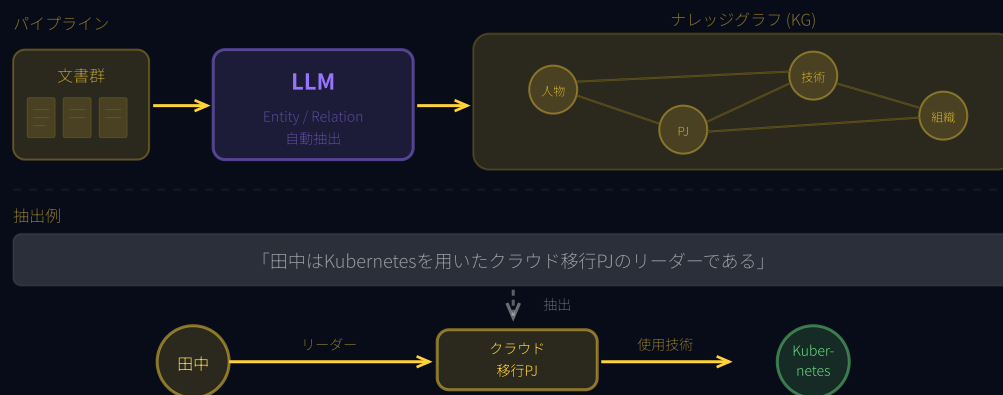
① KG構築

文書群からLLMでEntity (実体) とRelation (関係) を自動抽出し、ナレッジグラフを構築。手動のオントロジー設計なしに、大量文書から知識を構造化できる。



② RAG利用

RAGの本質は、LLMの限られたtoken枠に最も効果的な情報を選び出し、最適なプロンプトを構成する技術。差が出る根本原因は「準備段階」にある。Vector RAGはテキストをトークンサイズに切っただけで構造は失われる。Graph RAGは事前に知識を構造化しているから、関係を辿って必要十分な文脈を集められる。

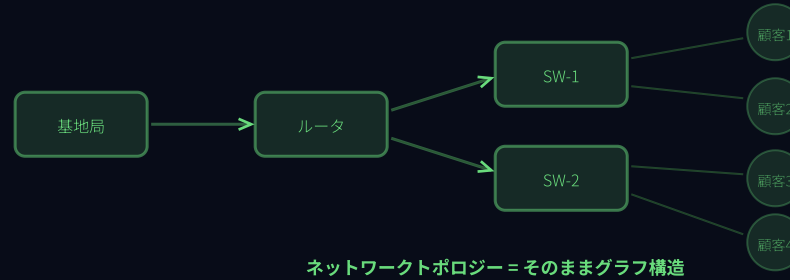


グラフDBは概念間の関係を構造として保持する。この構造があるからこそ、関係を辿って必要十分な文脈を集められる。RAGにおいて「最適なプロンプト」を作れるかどうかは、情報を事前に構造化しているかどうかで決まる。

ネットワーク・IT運用

通信ネットワーク管理

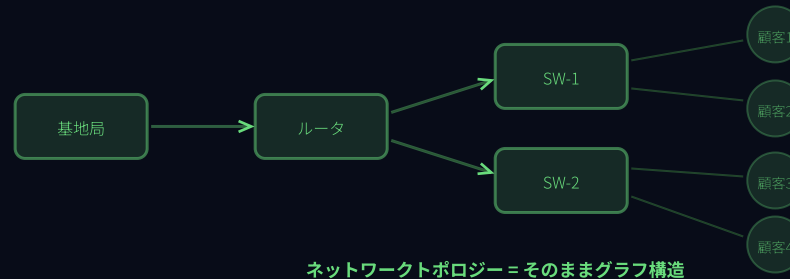
基地局→ルータ→スイッチ→顧客のトポロジーで、障害伝播をリアルタイムにシミュレーション。



ネットワークトポロジー=そのままグラフ構造

サイレント故障の被疑箇所特定

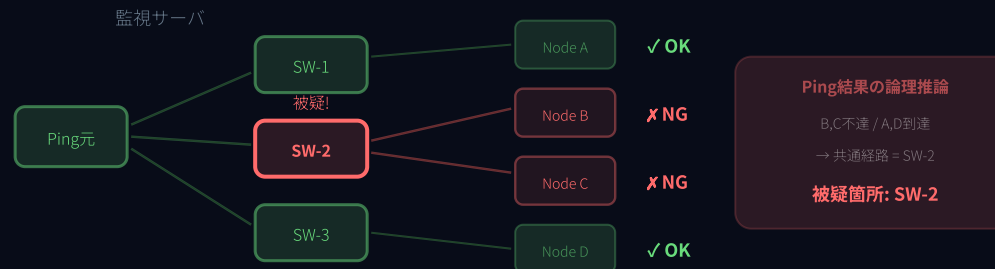
ネットワーク上の複数ノードへPingを打ち、到達・不達パターンからグラフ上の被疑箇所を絞り込む。依存関係の逆方向探索で「どの機器が原因か」を特定。



ネットワークトポロジー=そのままグラフ構造

マイクロサービス依存関係

複数の呼び出しパスが1つのサービスに集中→単一障害点を自動検出。グラフDBなら「全パスが通過するノード」をクエリー一本で特定できる。



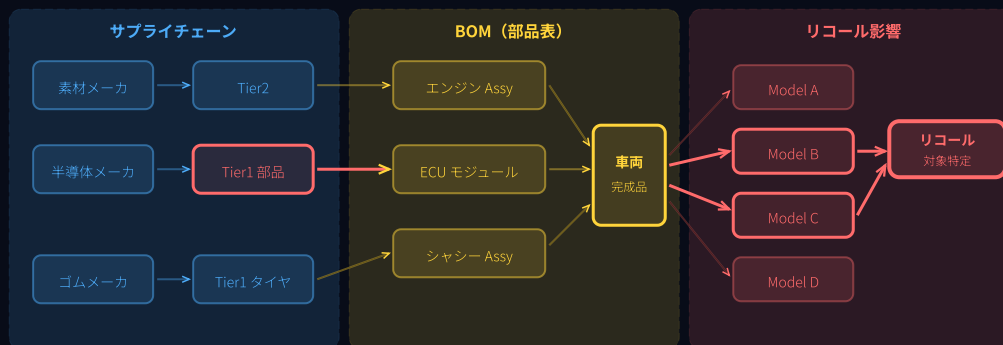
ネットワークトポロジーはそのままグラフ構造。依存関係の逆引き・影響範囲の特定はグラフ探索そのもの。

サプライチェーンとBOM

BOM×サプライチェーン×リコール統合

欧州大手自動車メーカーは12のデータソースを1つのグラフに統合。BOM（車1台≈30,000部品）+数百サプライヤーの供給網+不具合影響分析を一元管理。半導体不足時、計画を3週間→45分に短縮。

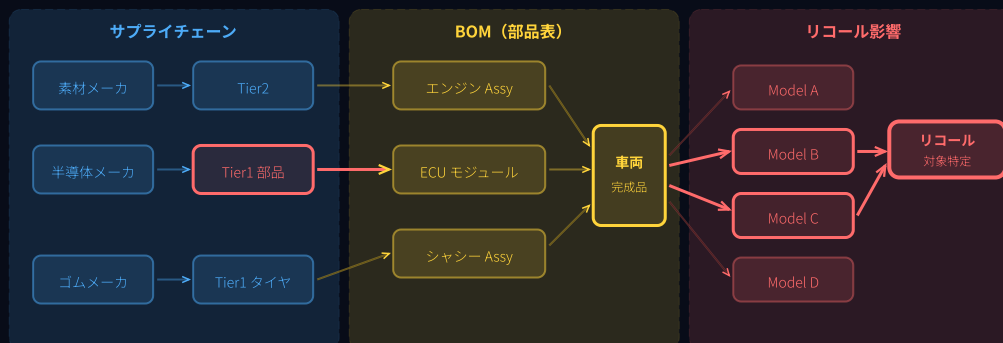
— 3つのレイヤーが1つのグラフで繋がる —



食品トレーサビリティ

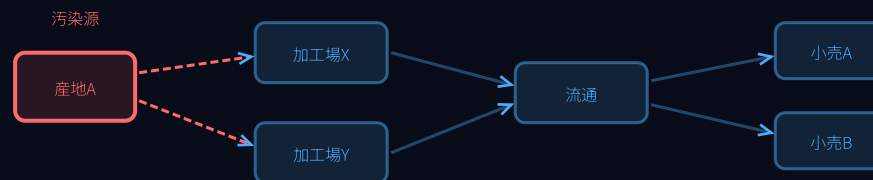
産地→加工場→流通→小売の経路を辿り、汚染源からの影響範囲を特定。

— 3つのレイヤーが1つのグラフで繋がる —



TIER-N 波及分析

Tier3の半導体メーカーが供給停止 → 影響を受ける最終製品は？ 多段の依存関係を逆方向に辿る必要があり、RDBでは再帰CTE×複数テーブルJOINが必要。グラフDBなら可変長パス探索一本。



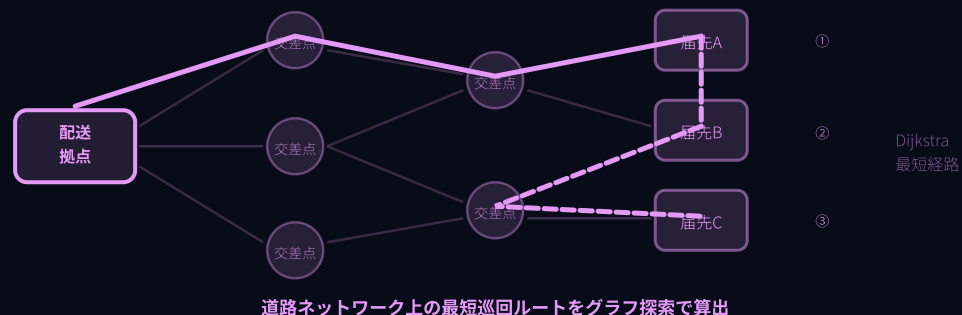
汚染源から全経路を辿り、影響範囲の小売を特定

部品→中間品→完成品の多段依存関係と、「この部品を使っている全製品」の逆引きは、グラフの双方向探索が最適。

ルート最適化・物流

ラストワンマイル配送最適化

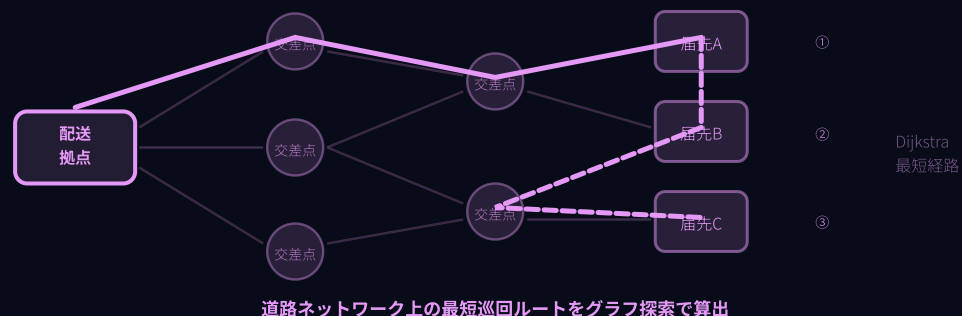
配送拠点から複数届先への最短巡回ルートを算出。道路ネットワークをグラフとして探索し、リアルタイムの渋滞情報で動的に再計算。



道路ネットワーク上の最短巡回ルートをグラフ探索で算出

リアルタイム再ルーティング

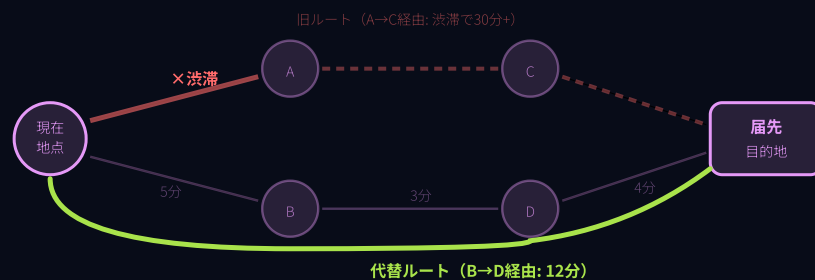
事故・渋滞が発生したらエッジの重み（所要時間）を更新し、影響を受ける配送車に代替ルートを即時提示。



道路ネットワーク上の最短巡回ルートをグラフ探索で算出

マルチモーダル経路探索

乗換必須の区間で、路線×乗換駅の組合せをグラフ探索し所要時間・料金のパレート最適ルートを算出。



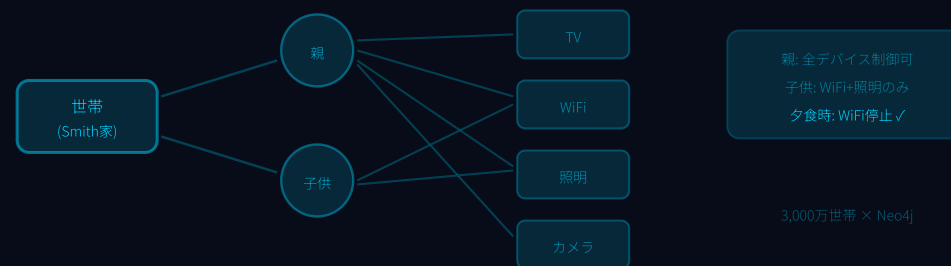
代替ルート (B→D経由: 12分)

道路・鉄道ネットワークはそのままグラフ。Dijkstra/A*等の最短経路アルゴリズムがネイティブに動作し、エッジの重み（距離・時間・料金）を動的に更新して即座に再計算できる。RDBでは再帰CTEの組合せ爆発で現実的な応答時間に収まらない。

アクセス制御・認可

COMCAST XFINITY XFI

3,000万世帯超のスマートホーム基盤。ユーザー→デバイス→場所→家族の関係グラフで、権限制御をリアルタイムに判定。Neo4j採用。

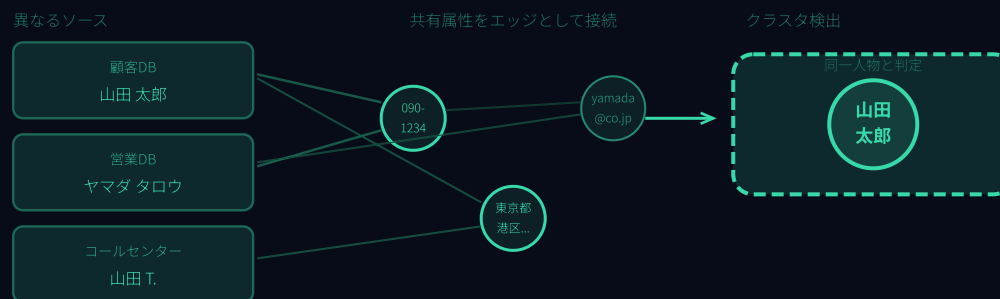


「アクセス可能か？」は本質的に経路探索問題。グループのネストや権限の継承は可変長パスであり、グラフDBの得意領域。

データクレンジング

重複レコード検出

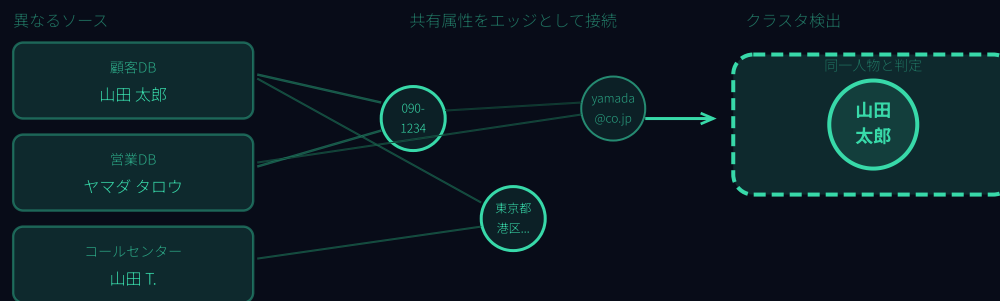
複数システムに散在する「同一エンティティ」を、共有属性（電話番号・住所・氏名の類似）をグラフの辺として接続し、クラスタリングで自動検出。RDBのクロスJOINでは爆発する計算量を、グラフ探索で実用的に処理。



RDBのクロスJOIN → 爆発 | グラフ → 属性ノードを辿るだけ

参照整合性・孤立データ検出

メタデータをグラフ化し、参照先が消えたぶら下がり、投入順序が決まらない循環参照、どこからも参照されない孤立テーブルをグラフ走査で一括検出。移行時の「何を捨てるか」「どの順で入れるか」の判断材料に。



RDBのクロスJOIN → 爆発 | グラフ → 属性ノードを辿るだけ

データクレンジングの核心は「レコード間の関係」の問題 — 名寄せ・重複・参照切れ・孤立。これらは全てグラフ探索で効率的に解ける。

LIVE DEMO

実データで試してみた

e-Gov法令APIから外国人労働に関する法令の全文を取得し Neo4j に投入

```
MATCH (n) RETURN labels(n)[0] AS type, count(n) AS cnt
```

6 法令

2,907 ノード

1,643 関係

※ 当日はライブクエリを実行

KNOWLEDGE GRAPH CONSTRUCTION

法令KGはどう作ったか

STEP 1

構造の取込み

e-Gov法令API からXML取得
法令 → 章 → 条 → 項 の階層を
そのままグラフ構造に変換

15法令 → 6,213ノード / 10,312エッジ
デジタル庁 法令標準XMLスキーマ準拠



STEP 2

関係の抽出

条文テキストから参照関係を検出
「第X条の規定により」 → REFERENCES
法令間の承継 → SUCCEEDED_BY

正規表現 + LLMで条文間リンクを構築
→ Demo 1-4 でライブ探索した基盤



STEP 3

概念の構造化

法律オントロジーで型を定義
LKIF (EU標準) + LegalRuleML (OASIS)
義務 / 許可 / 禁止 を区別して抽出

LLMで抽出するが「何を抽出するか」は
オントロジーが決める (LLM丸投げではない)

しかし、本当に難しいのは…

法律オントロジー (LKIF等) は公開標準 — 誰でも使える
しかし法が規制する**現実世界の条件** (居住環境、報酬水準、保険要件…) を
構造化する**ドメインオントロジー**は、入管法分野では標準が存在しない

付加価値の所在

LLM・DB・検索 → 買う

法律オントロジー → 使う

ドメインオントロジー → 作る

「何をノードにし、何をエッジにするか」の設計には発注者の業務知識が不可欠 — 外注できない知的資産

INDUSTRY TRENDS

グラフDBを取り巻く潮流

GQL 標準化

ISO/IEC 39075:2024 発行

SQLに次ぐ国際標準クエリ言語

Neo4j (Cypher), TigerGraph (GSQL) → GQLへ収束

RDB のグラフ対応

Oracle 23ai — SQL/PGQ でグラフクエリ

SQL Server 2025 — グラフテーブル強化

「グラフは特殊」 → 「RDBの一機能」へ

GraphRAG の台頭

Microsoft GraphRAG (2024 OSS公開)

LLM + KG で構造的な知識検索

Vector RAG の弱点 (構造・関係性) を補完

海外での利用加速

金融: 不正検知 (HSBC, PayPal)

医療: 創薬・臨床 (AstraZeneca)

公共: NASA, 欧州特許庁, ICIJ

日本は導入黎明期 — だからこそ機会

KG × 創薬 — ドラッグリポジショニングの加速

NEC × 中外製薬 (2025): グラフAIでがん併用療法の薬剤組み合わせを予測 **名古屋大**: AlphaFold × 既存薬8,000種で総当たり探索

AI×リポジショニング市場 年成長率 20.5% — BenevolentAI (2020) 以降、KGによる異分野横断発見が創薬の新常識に

「グラフDBはニッチ」 → 「標準技術」へ — 日本でも導入の好機

LESSONS LEARNED

実装・運用のリアル

メモリとサイジング

10MノードでOOM (3GB heap)

→ 本番は最低8GB heap + ページキャッシュ設計

Cartesian Product 爆発

密結合データでCOUNT(*) が243倍

→ DISTINCT / LIMIT / APOCで制御

スキーマ設計が命

ランダムデータ ≠ 本番データ

→ ホットクラスター (密結合) を意識した設計

何でもグラフにしない

集計・帳票 → RDB / DWH

→ 関係探索が本質の問題にのみ適用

人材・スキル

Cypher / GQL を書ける人材は希少

→ RDB経験者の転換は比較的容易 (概念は近い)

オントロジー設計

「何をノード/エッジにするか」が最難関

→ 発注者の業務知識が不可欠 (外注不可)

技術より設計と運用が成否を分ける — RDBと同じ

SUMMARY

つながるを 支える技術

Graph DB

関係を辿る問題に
圧倒的な強さ

GraphRAG

LLM × グラフで
構造的に回答

時間軸

SUPERSEDES で
未来の法令も管理

株式会社UTI | 細井 公義