



# Amazon Neptune徹底紹介

クラウドのための、高速で信頼性の高い  
グラフデータベース

データベース事業開発本部

NoSQL GTM Specialist

浅野 哲

# 自己紹介

## 浅野 哲 (あさの さとし)

所属：

データベース事業開発本部  
NoSQL GTM Specialist

担当製品：

Amazon Neptune  
Amazon Timestream  
Amazon QLDB  
Amazon Managed Blockchain

前職：

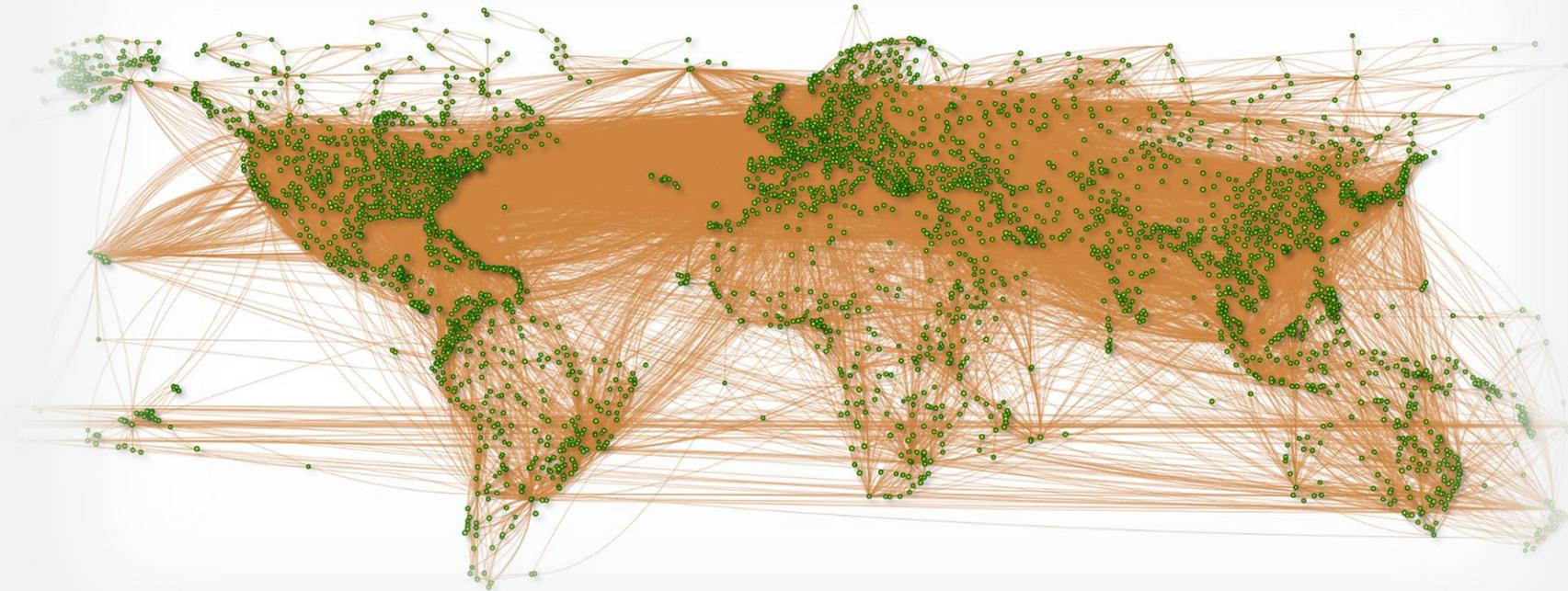
Oracle社において、OpenVMSで稼働するリレーショナルデータベースの開発。



# Agenda

- グラフとは何か?
- なぜ、グラフおよびグラフデータベースなのか?
- Amazon Neptuneの利点
- 事例
- はじめるためには

# グラフとは何か？ : グラフ可視化の例



# グラフデータベースとは何か？

- グラフデータベースは関係を保存およびナビゲートするための専用に構築されたいわゆるpurpose-built データベースです。
- ノード(node, vertex)は実世界の対象物(object)を表します。
- エッジ(Edge)はオブジェクト間の関係を格納します。
- プロパティとラベルはノードとエッジの両方に追加できます。

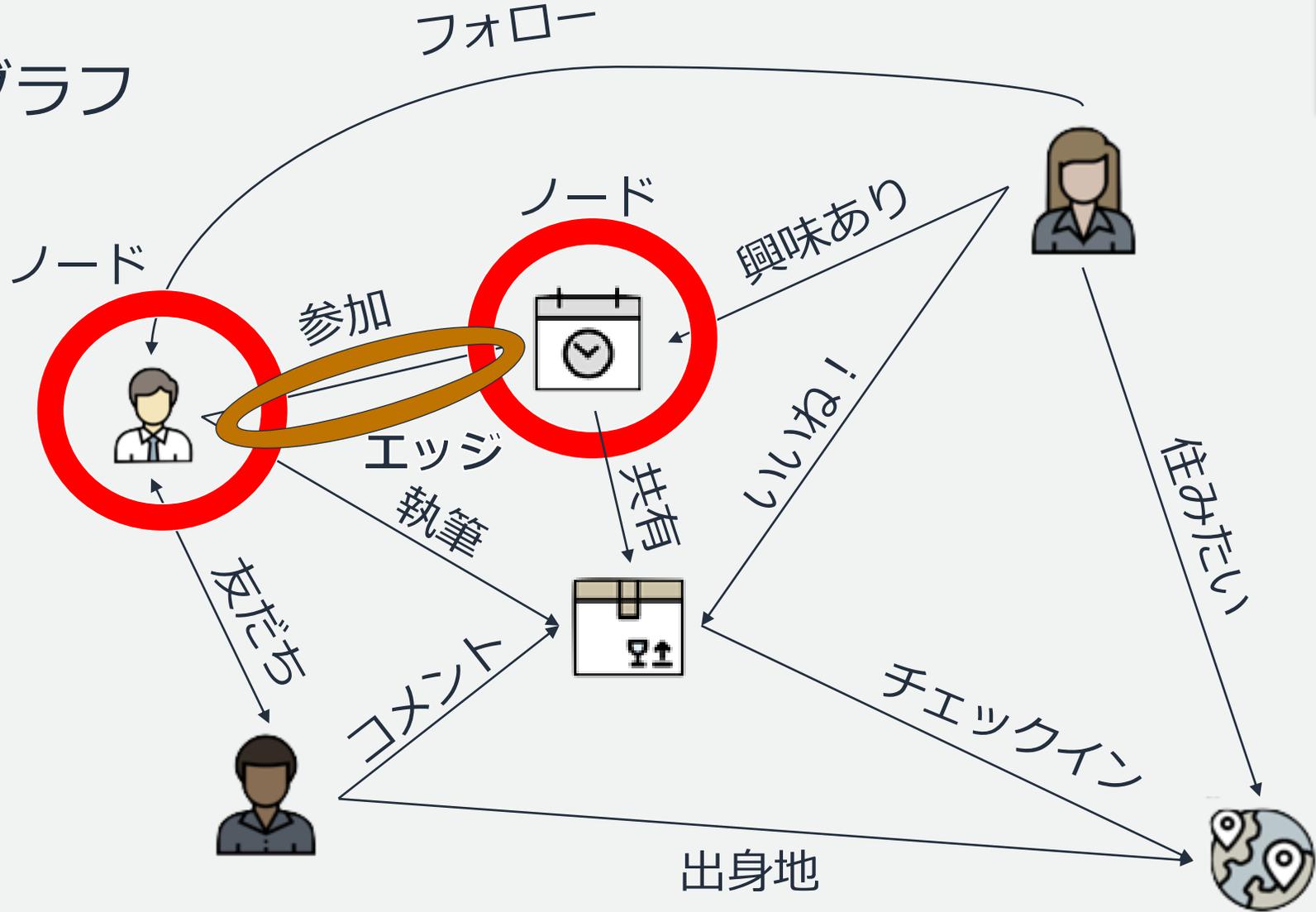


どのような場面で  
グラフデータベースを  
使うべきなのか？

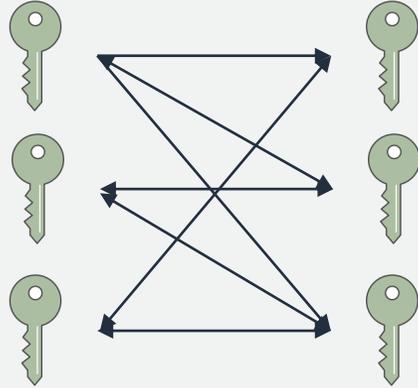


# グラフ構造～ノードとエッジ

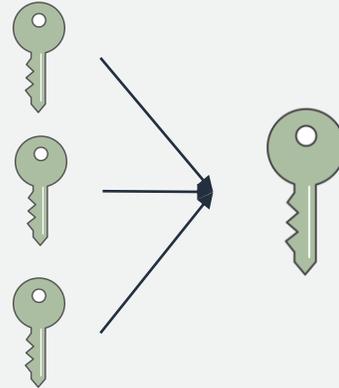
## ソーシャルグラフ



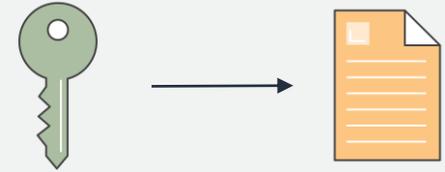
# データモデリングからの視点



• 多対多



多対1



• 1対1

• グラフデータベース



Amazon Neptune

RDBMS

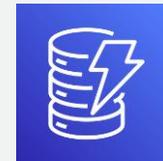


Amazon RDS



Amazon Redshift

• Key-Value ストア

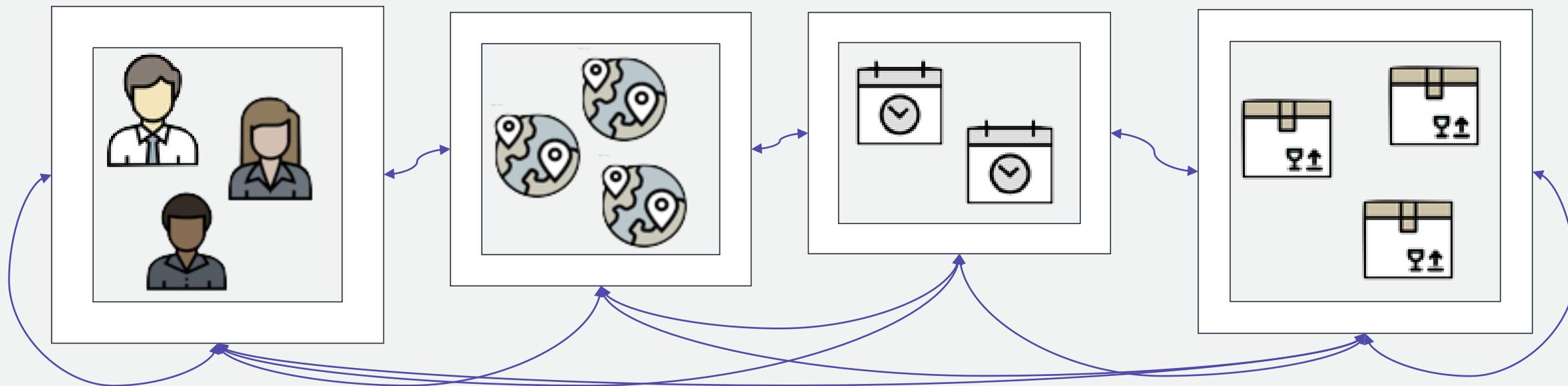


Amazon DynamoDB



Amazon ElastiCache

# ソーシャルグラフでは？



すべての関連が多対多  
It's Graph

# SQLからの視点

- 使用するSQL文に再帰With句やConnect By (Oracle)を使う必要がある

- **with** tab(ID,PID,Lv,Path) as(  
select ID,PID,1  
from Data  
where PID is null  
union all  
select d.ID,d.PID,t.Lv+1  
from tab t, Data d  
where t.ID = d.PID)  
select \* from tab;

→グラフ探索はクエリの複雑性が高くなりやすい

# なぜグラフなのか？



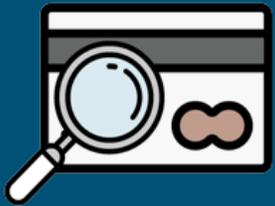
ソーシャル  
ネットワーク



レコメンデーション  
/アイデンティティ



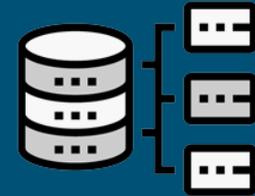
ナレッジグラフ



不正検知



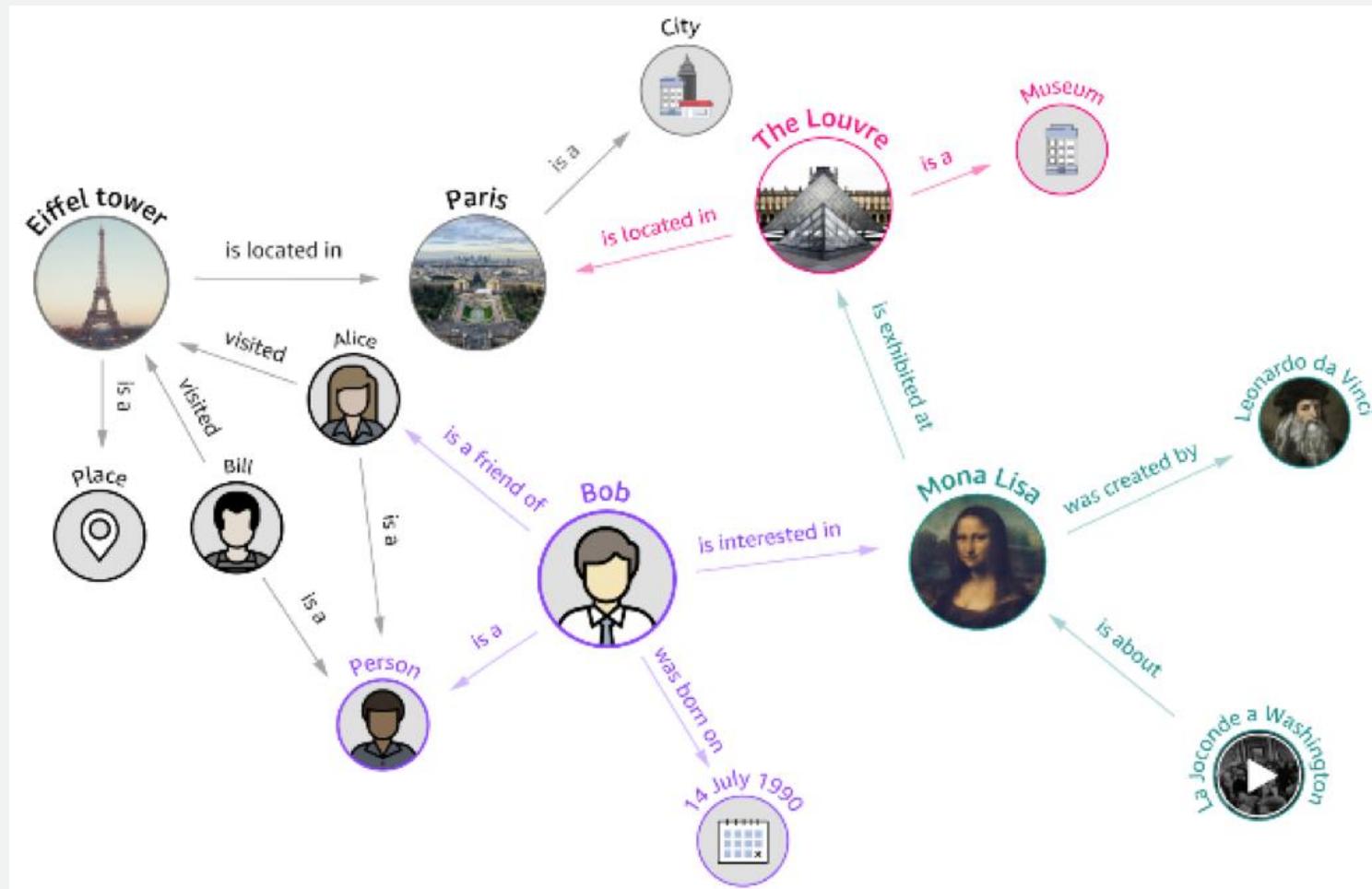
製薬・医療



ネットワーク  
・IT運用

# ナレッジグラフとは何か？

誰が、何を、いつ、どこでを理解



## Benefits

### 1. 異なるデータソースをリンク

異なる、不均質なデータソースをリンクして、隠れた接続を発見します。

### 2. 検索結果の改善

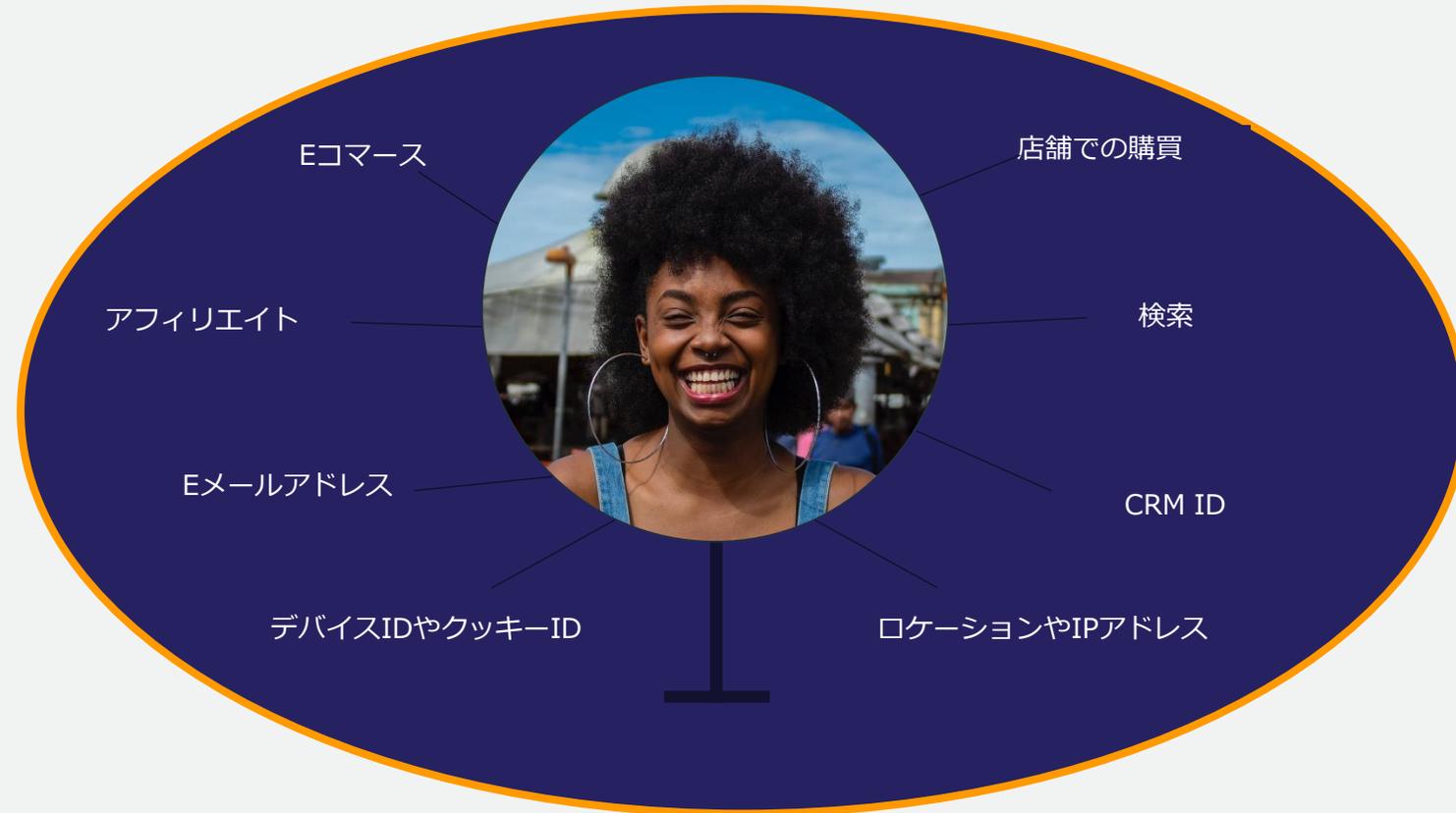
検索の関連性を改善してデータに簡単にアクセスできるようにすることで、生産性を向上させます。

### 3. ML/AIを強化する

関連するコンテンツでコンテキストを提供することで、機械学習モデルの効率と効果を向上させます。

# アイデンティティグラフとは何か？

## Unified 360° View Of The Customer



## Benefits

### 1. デバイスおよびチャネル間での永続ID

関連するすべてのデバイスとIDをリンクする永続的な識別子。一元的なプロフィールの作成、ターゲティング、パーソナライズを可能にします。

### 2. 閲覧者の構築とセグメンテーション

同様の興味、好み、購入の基づいて閲覧者を作成します。

### 3. クロスデバイスの頻度キャッチング

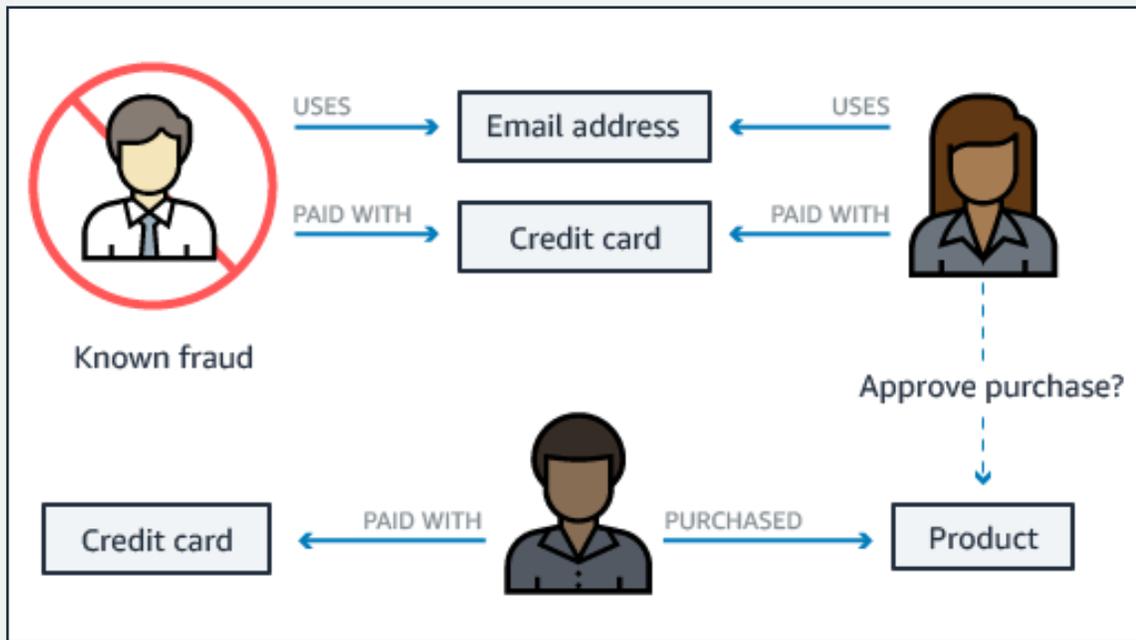
閲覧者向けにデバイスとブラウザ全体の頻度のキャッチングを改善して、カスタマーエクスペリエンスを向上させます。

### 4. カスタマージャーニーの分析

エンドツーエンドの顧客行動を分析し、購入へのパスと属性を追跡します。

# 不正検知グラフとは何か？

## 不正検出



## 利点

### 1. リアルタイムでの不正防止

トランザクションの不正パターンを特定し、発生後ではなく発生時に防止します。

### 2. 独自のインサイト

ユースケースは金融関連だけに留まらず、ゲームや種々の異常検知などに及びます。

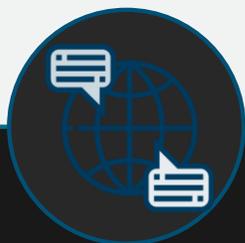
### 3. 変化への対応

接続されたデータの柔軟な性質を使用して、最小限の労力に変化する不正の性質に適応します。不正AI/MLワークフローとの適切な組み合わせ。

# Amazon Neptune

クラウド向けに構築された高速で信頼性の高いグラフデータベース

Open



Apache TinkerPop,  
openCypher と W3C RDF グ  
ラフモデルをサポート

Fast



ミリ秒のレイテンシで数十億の関  
係をクエリー

Reliable



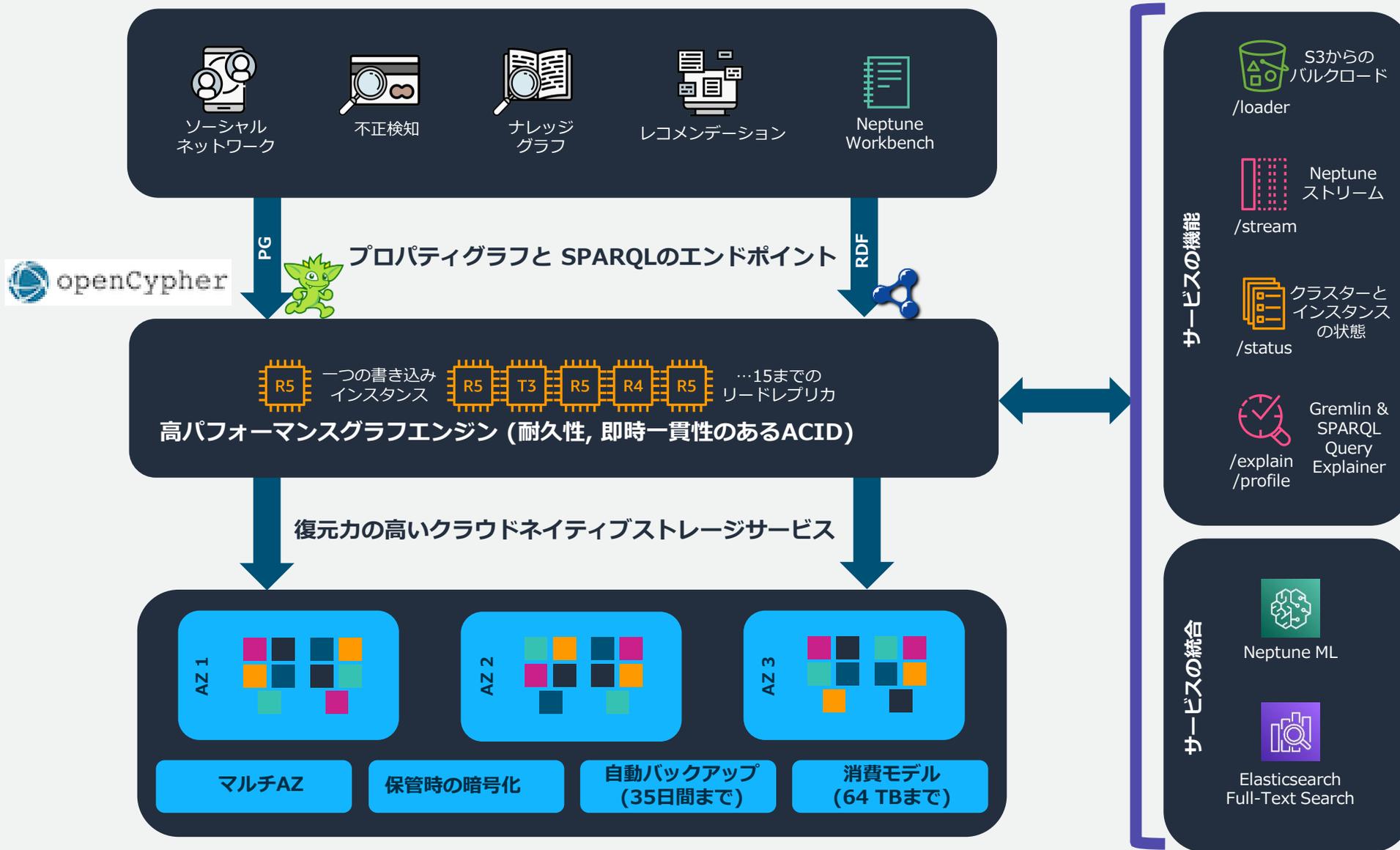
3つのAZに6個のレプリカ

Easy

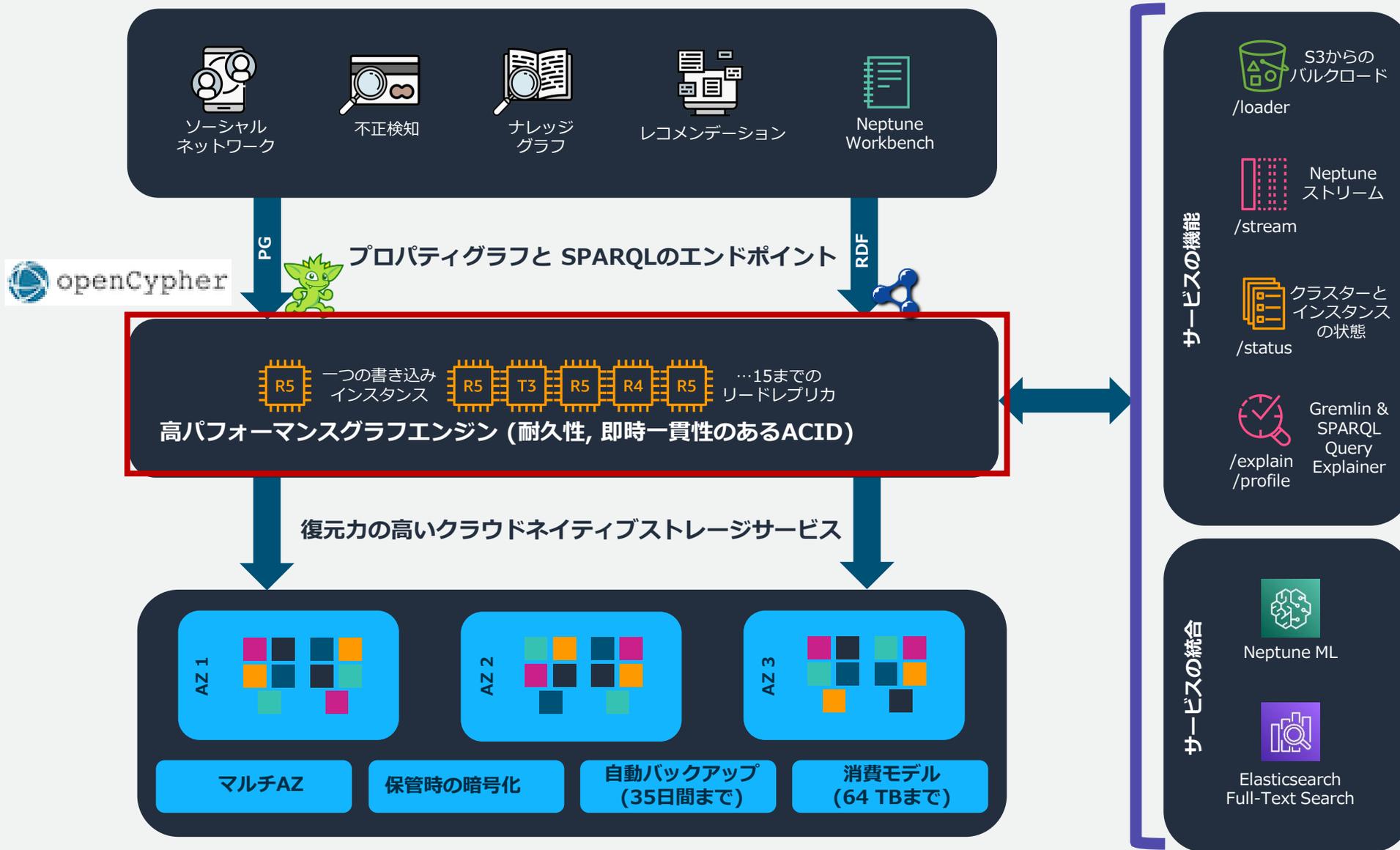


Gremlin と SPARQLを使用し  
て今日六なクエリを簡単に構築

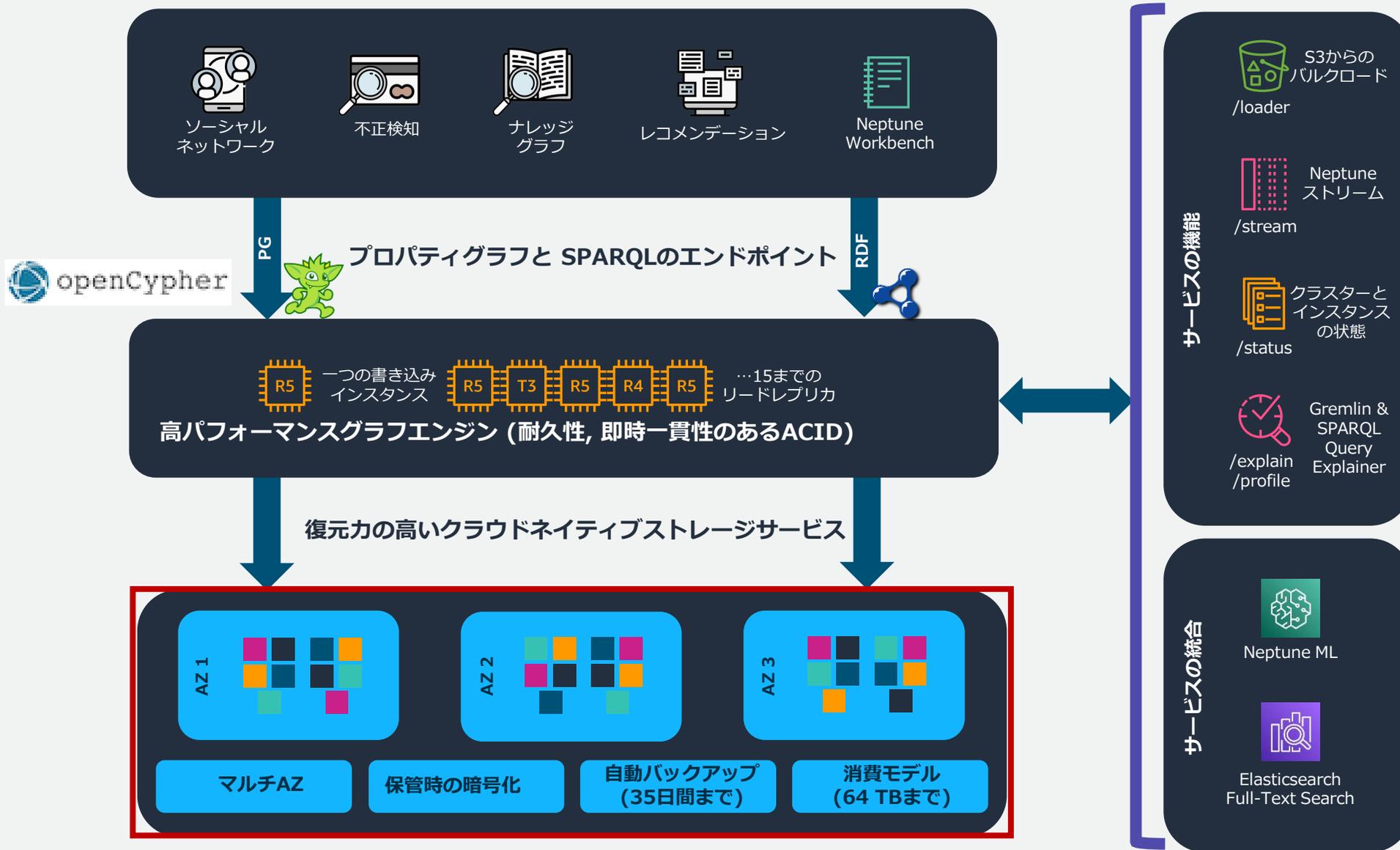
# Amazon Neptune のハイレベルアーキテクチャ



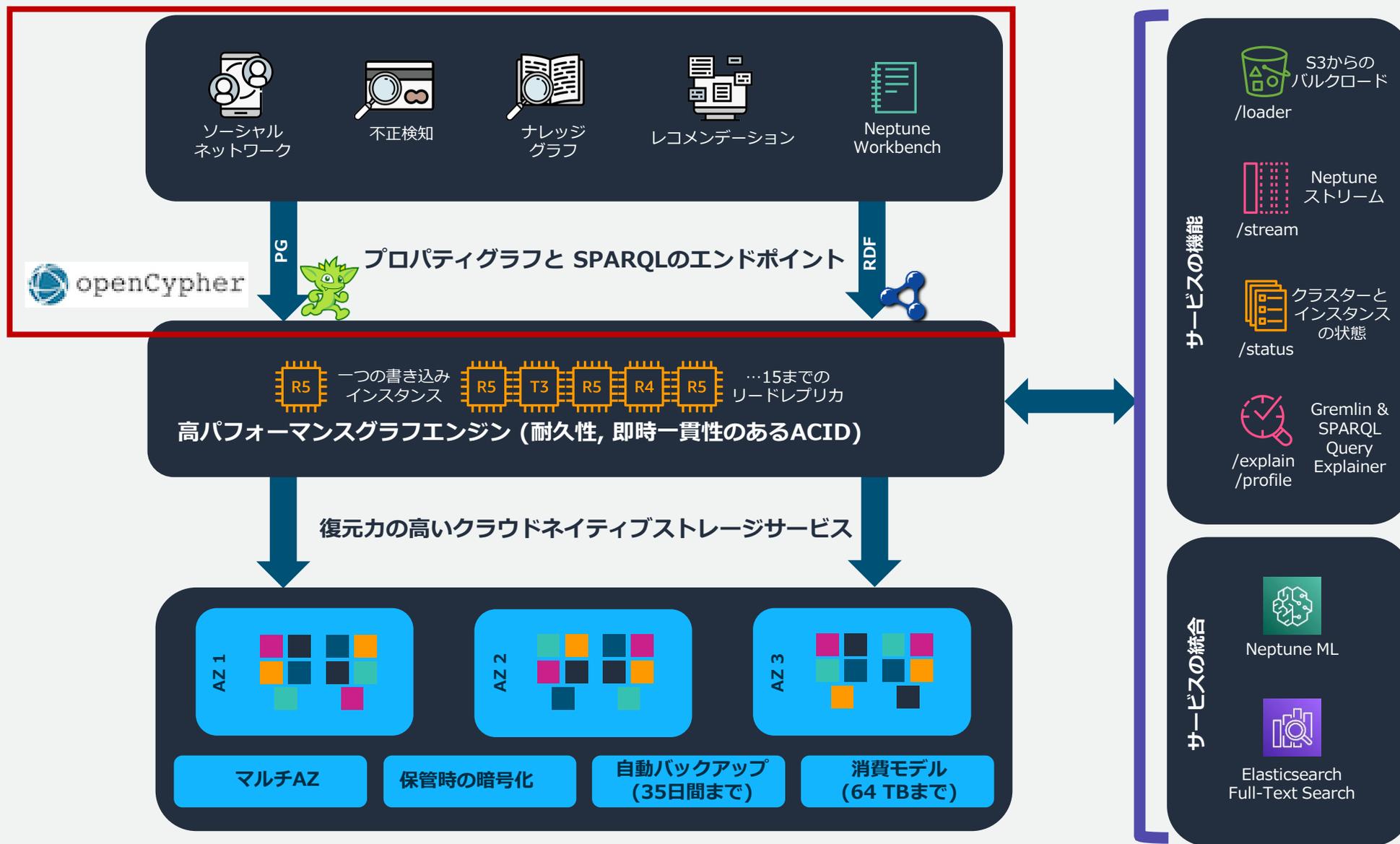
# Amazon Neptune のハイレベルアーキテクチャ



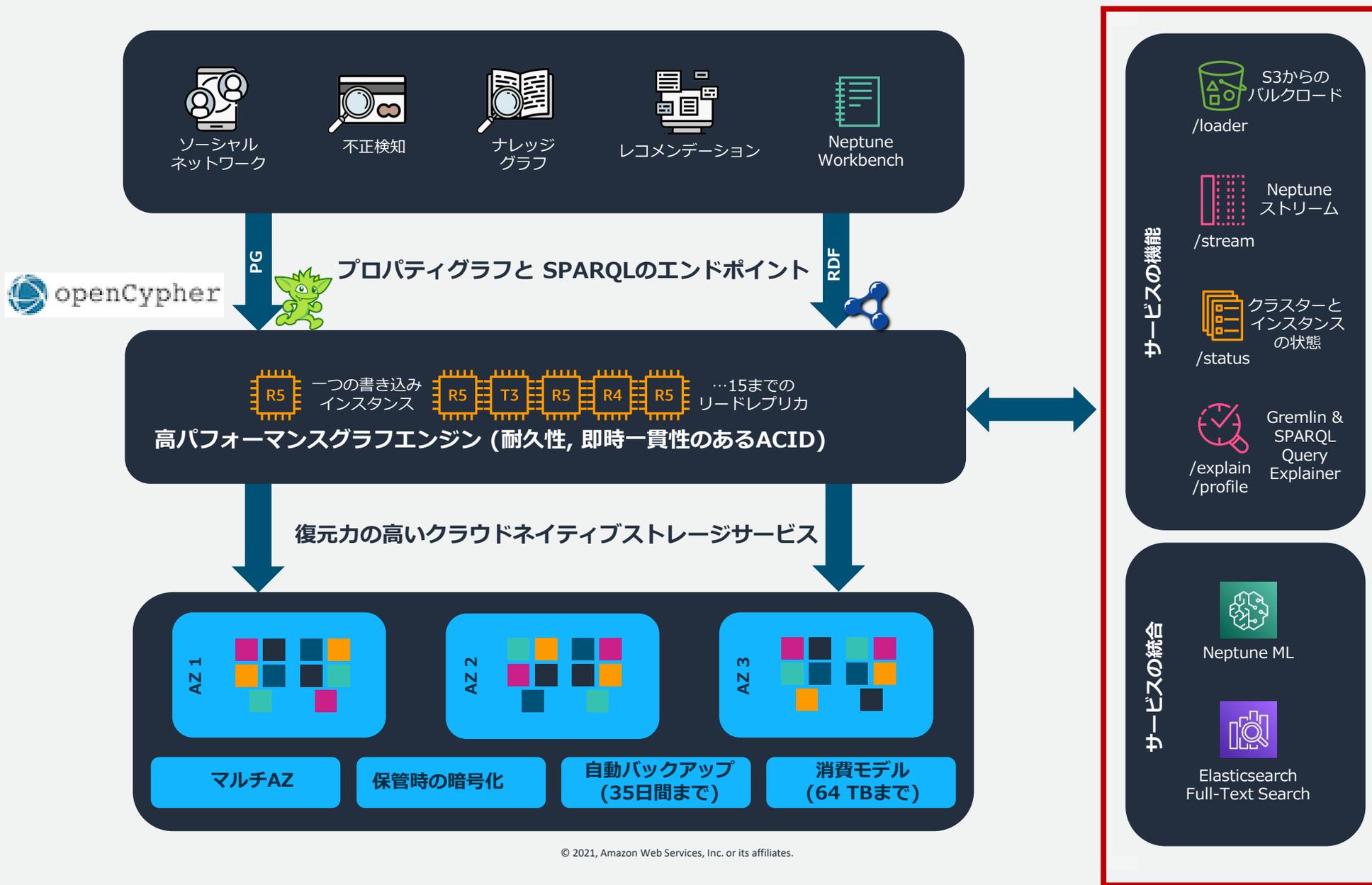
# Amazon Neptune のハイレベルアーキテクチャ



# Amazon Neptune のハイレベルアーキテクチャ



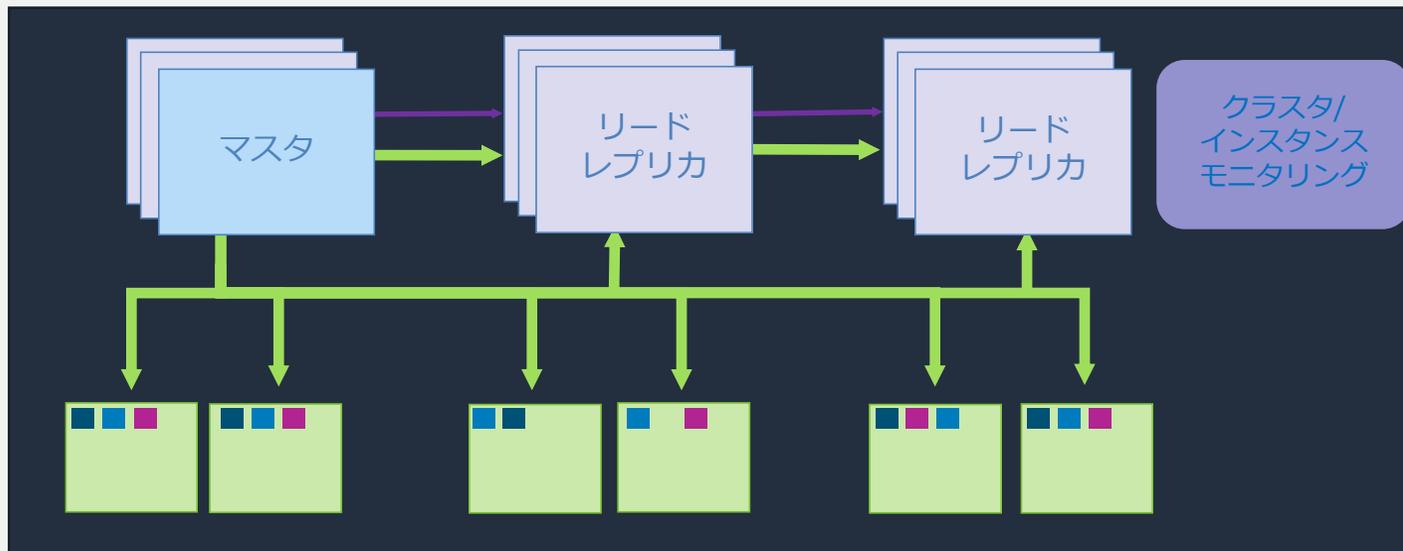
# Amazon Neptune のハイレベルアーキテクチャ



# リードレプリカ

## 可用性

- データベースノードの障害は自動的に検知され、交換される
- データベース処理の障害は自動的に検知され、リソースはリサイクルされる
- リードレプリカは必要に応じて自動的にマスタに昇格する
- どのリードレプリカに対して優先的にフェイルオーバーさせるかを指定できる



## パフォーマンス

- リードレプリカによってアプリケーションの読み込みトラフィックをスケールアウトさせることができる
- 読み込みエンドポイントによってリードレプリカを跨がって負荷が分散される

# クラウドネイティブストレージエンジンの概要

データは3つのアベイラビリティゾーンに跨った6つのレプリカにコピーされる

継続的に堅牢な Amazon S3 へバックアップされる

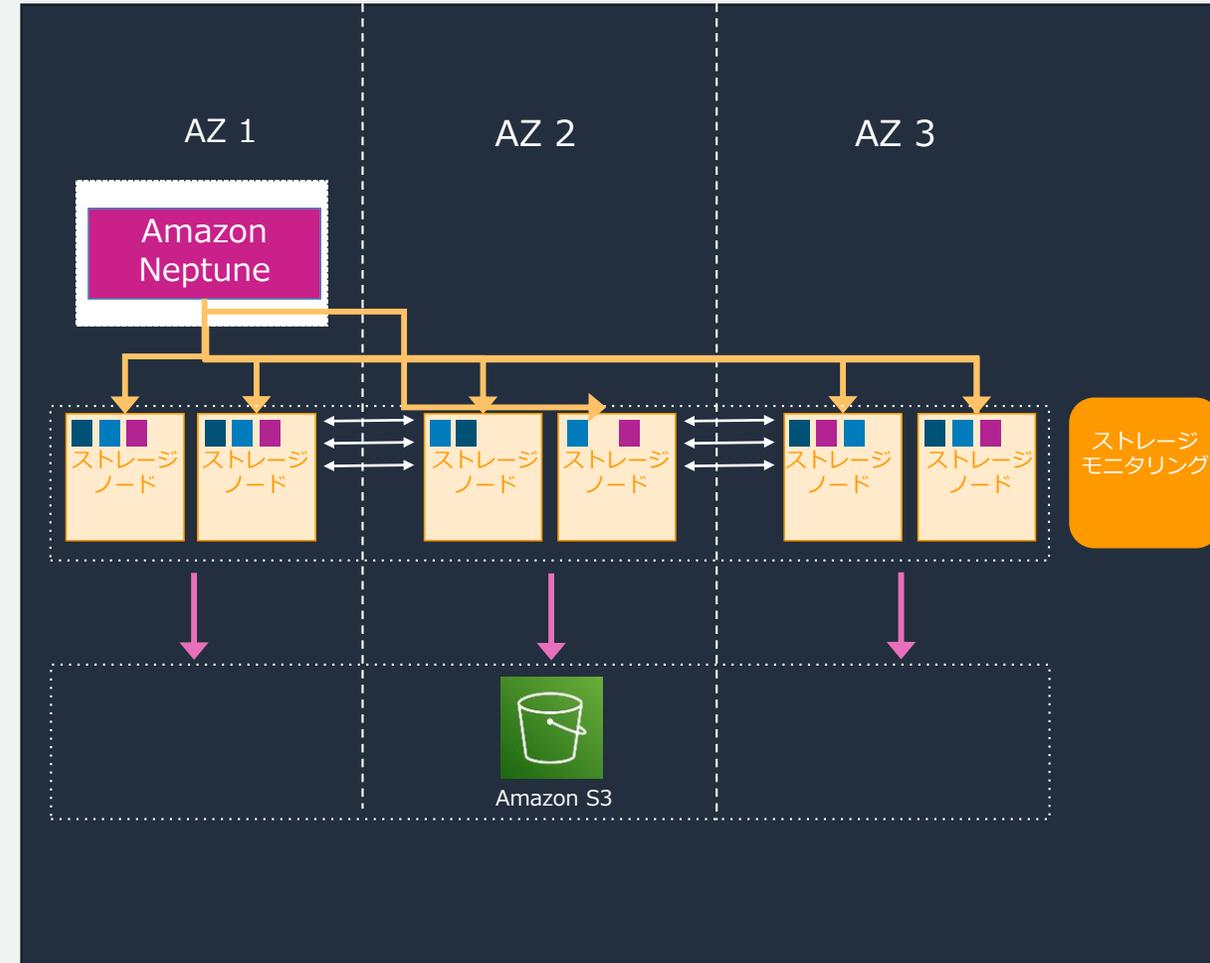
継続的にノードやディスクが修復される

修復やホットスポットのリバランスのために10GBのセグメントユニットで管理されている

読み書きにはレイテンシ耐性を持つクォーラムシステムを使用している

クォーラムのメンバーシップが変更されたとしても書き込みは阻害されない

ストレージは使用に応じて自動的に64TBまで拡張される



# プロパティグラフと**RDF**の サポート



# グラフモデルとオープンソースフレームワーク

## PROPERTY GRAPH

Apache TinkerPop™

Gremlin Traversal Language



openCypher Query Language



## RESOURCE DESCRIPTION FRAMEWORK (RDF)

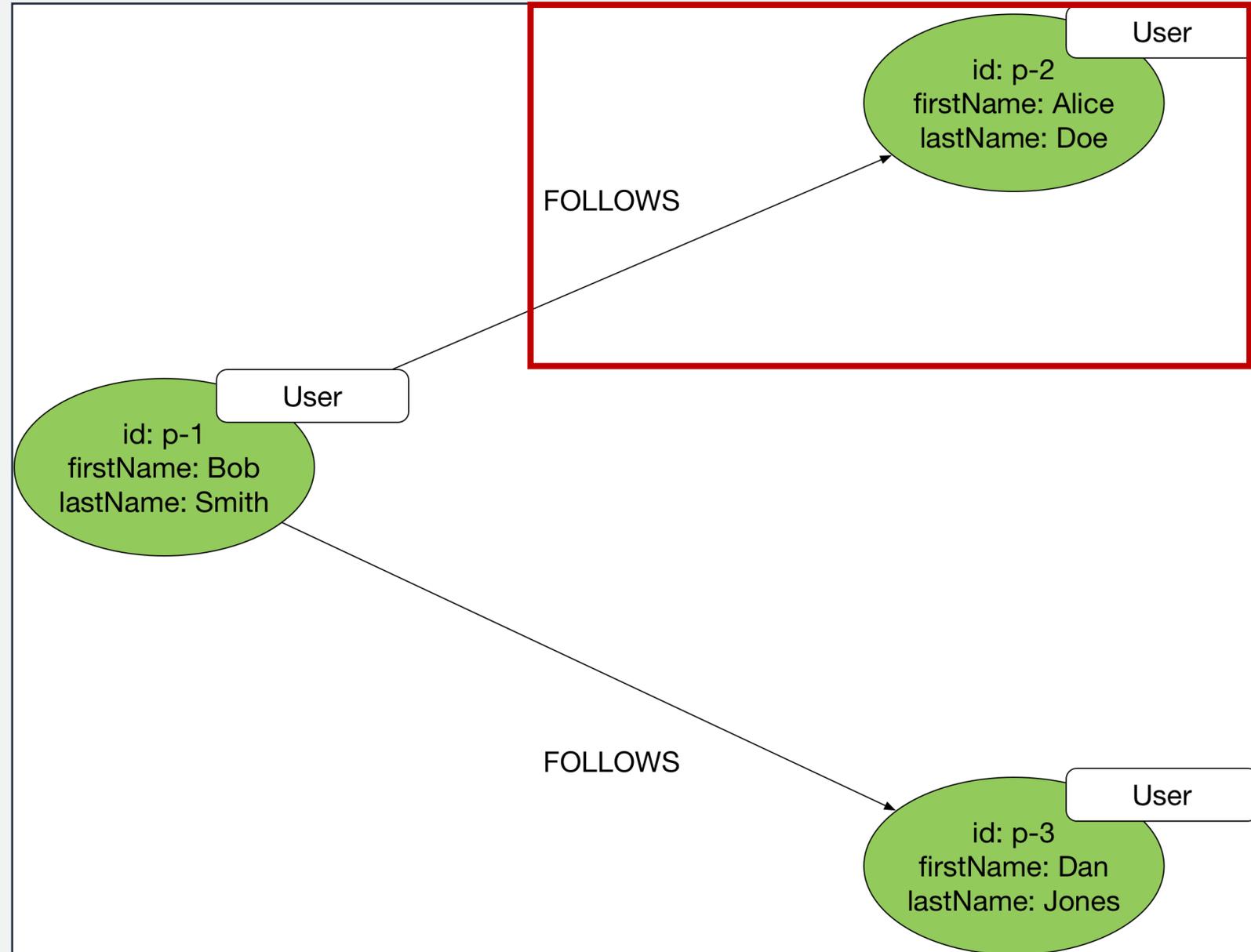
W3C Standard

SPARQL Query Language



# プロパティグラフ

- firstNameがBobを探す。  
`g.V().has('firstName', 'Bob').  
=>['Bob']`
- firstName:BobがFOLLOWSしている人を探す。  
`g.V().has('firstName', 'Bob').out('FOLLOWS')  
=>['Alice', 'Dan']`



# RDF グラフ

@prefix rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>

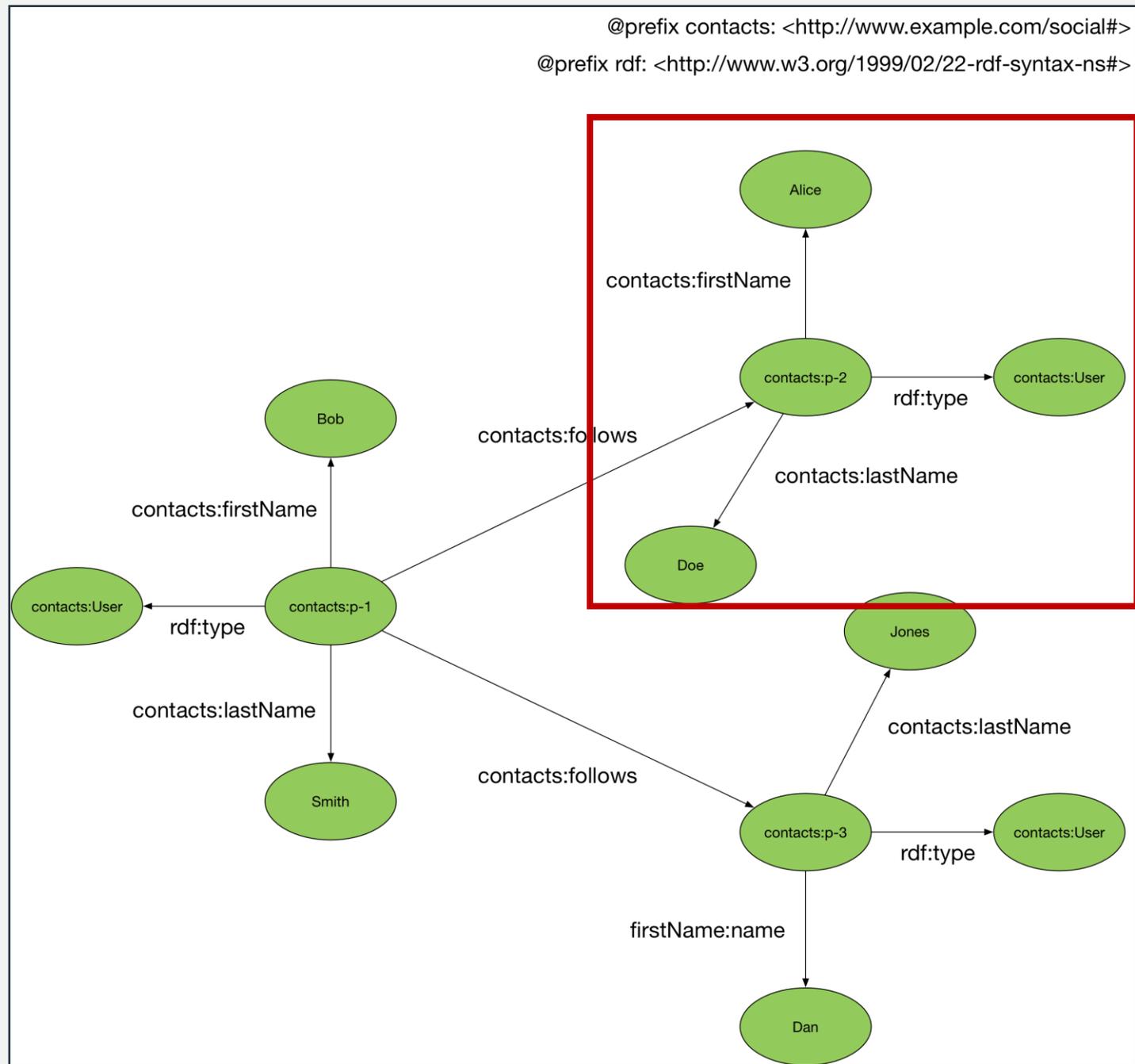
@prefix contacts: <<http://www.example.com/social#>>

contacts:p-2 contacts:firstName "Alice"

contacts:p-2 rdf:type contacts:User.

主語(IRI) 述語(IRI)

目的語(IRI またはリテラル)

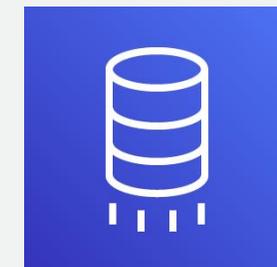


# データの投入

- バルクでデータを投入する方法は2種類
- S3上にCSVファイルを用意し、Neptune Loaderでロード
- AWS Database Migration Serviceを使用して他のDBから移行



Amazon S3



AWS DMS

# S3上のCSVファイルによるデータロード



Amazon S3

## NodeのCSVファイル例

```
~id, name:String, age:Int, ~label  
v1, UserA, 29, person  
v2, UserB, 30, person
```

## EdgeのCSVファイル例

```
~id,~from,~to,~label  
e1,v1,v2,follow  
e2,v2,v1,follow
```

[https://docs.aws.amazon.com/ja\\_jp/neptune/latest/userguide/bulk-load.html](https://docs.aws.amazon.com/ja_jp/neptune/latest/userguide/bulk-load.html)



# S3上のCSVファイルによるデータロード



Amazon S3

## NodeのCSVファイル例

```
~id, name:String, age:Int, ~label  
v1, UserA, 29, person  
v2, UserB, 30, person
```

## EdgeのCSVファイル例

```
~id,~from,~to,~label  
e1,v1,v2, follow  
e2,v2,v1, follow
```

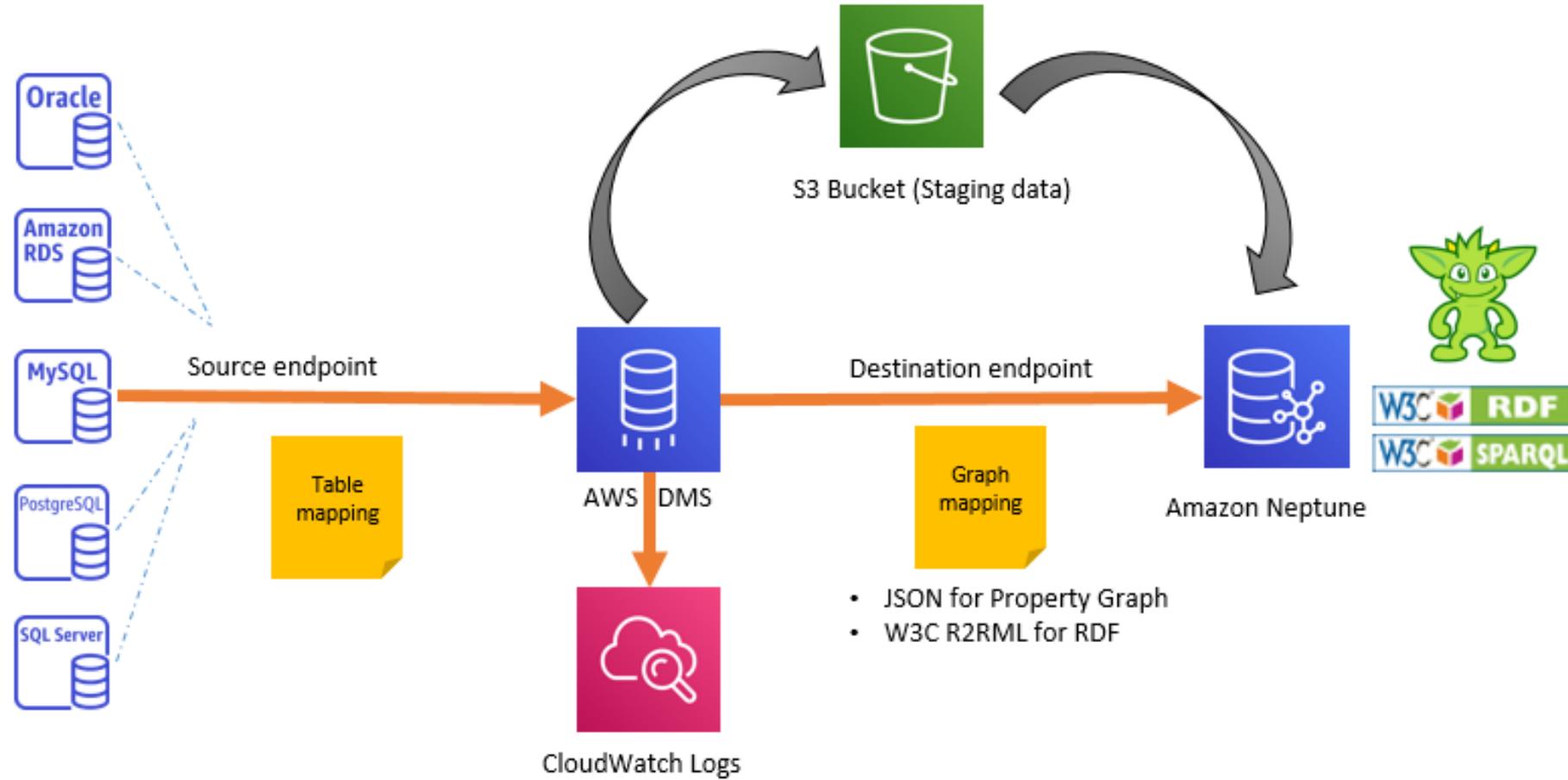
## ロード実行例

```
curl -X POST -H 'Content-Type:  
application/json' https://<neptune-endpoint>:8182/loader -d  
{  
  "source" : "s3://<S3 URI>/",  
  "format" : "csv",  
  "iamRoleArn" : "<IAM Role ARN>",  
  "region" : "<region>",  
  "failOnError" : "FALSE",  
  "parallelism" : "MEDIUM",  
  "updateSingleCardinalityProperties" : "FALSE",  
  "queueRequest" : "TRUE"  
}
```

[https://docs.aws.amazon.com/ja\\_jp/neptune/latest/userguide/bulk-load.html](https://docs.aws.amazon.com/ja_jp/neptune/latest/userguide/bulk-load.html)



# DMSによるリレーショナルデータベースからNeptuneへの移行



# 事例



# 顧客事例



SIEMENS



SAMSUNG

FACTSET



NETFLIX



# ナレッジグラフ: 経済産業省様 – gBizInfo 事例

<https://aws.amazon.com/jp/solutions/case-studies/meti/>

## 1.1. 概要



### ● gBizINFO が取り扱うデータ

#### ➢ 基本情報

#	活動情報	概要	レコード数	トリプル数
1	基本情報	法人番号基本 3 情報	4,764,737	247,259,324

#### ➢ 活動情報

#	活動情報	概要	レコード数	トリプル数
1	届出認定情報	公表している、法人の届出認定に関する情報	78,760	7,209,119
2	表彰情報	表彰した法人の情報	53,325	1,954,583
3	補助金情報	補助金決定に関する法人の情報	319,268	8,781,711
4	調達情報	競争入札・随意契約などによって調達された法人の情報	170,981	5,308,660
5	特許情報	特許データに関する情報 特許・意匠・商標情報を含む	1,092,457	25,338,171
	合計		1,714,791	48,592,244

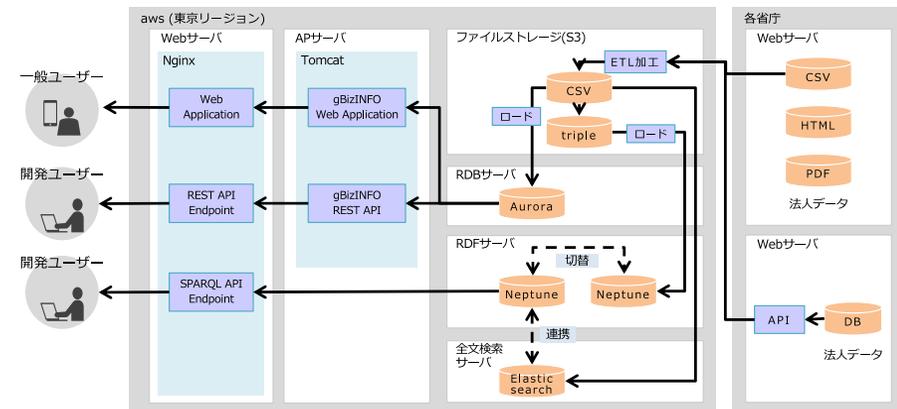
基本情報、活動情報合計 295,851,568トリプル (約3億トリプル) を公開。  
性能計測は上記データ、トリプル数にて検証した。

## 3.2. システム概要



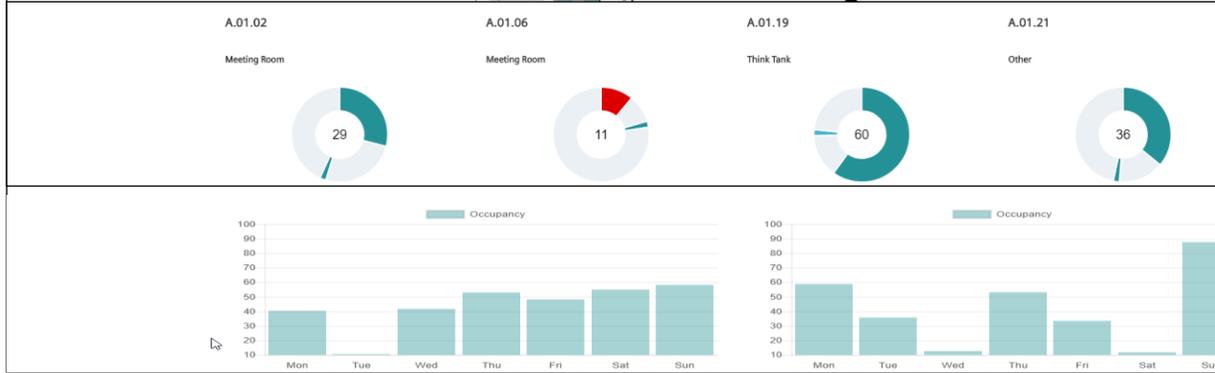
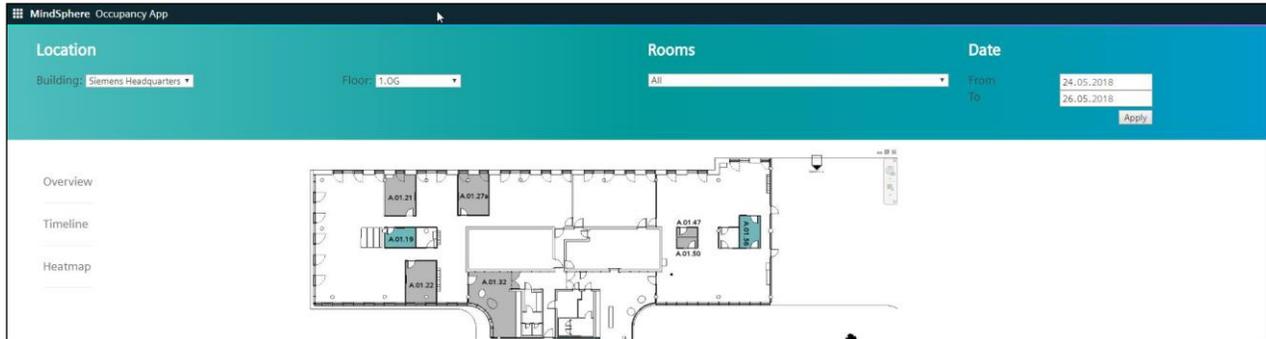
### ● AWS + Neptune の環境概要

- AWSのサービス、Neptuneを利用、かつインスタンスサイズを最適化し、コストを削減する。
- AWS(東京リージョン)に環境構築。RDBにAurora(PostgreSQL)、RDFにNeptuneを使用し、Elasticsearchと連携。
- SPARQLリクエストはWebサーバで受け付け、NeptuneのHTTPSエンドポイントへリバースプロキシ。
- グラフデータベースのスペックは16vCPU、128GB。



# SIEMENS における BIM

## MindSphere



Unrestricted © Siemens AG 2018

Page 17

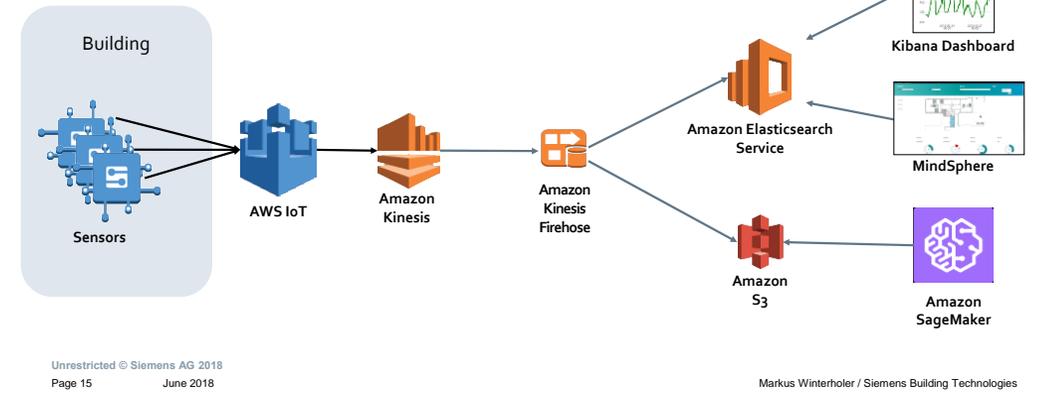
June 2018

Markus Wir



SIEMENS  
Ingenuity for life

## Time Series Data from Sensors IoT Connectivity

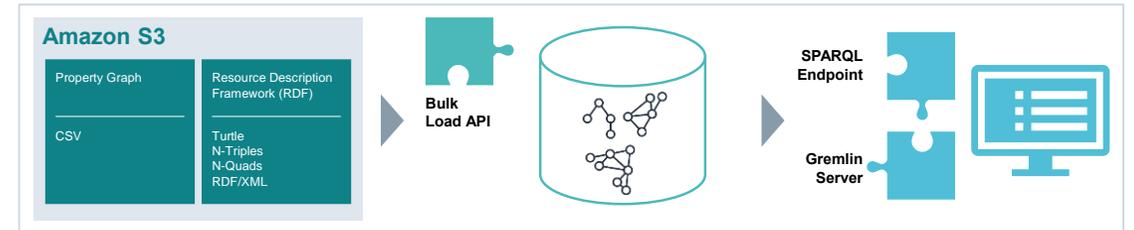


Unrestricted © Siemens AG 2018  
Page 15 June 2018

Markus Winterholer / Siemens Building Technologies

## Amazon Neptune Graph Database maintains the logical data model and links to other data sources

SIEMENS  
Ingenuity for life



Load Property Graph and RDF Data

Store billions of relationships

Fast graph queries

<https://aws.amazon.com/de/neptune>

Unrestricted © Siemens AG 2018

Page 18

June 2018

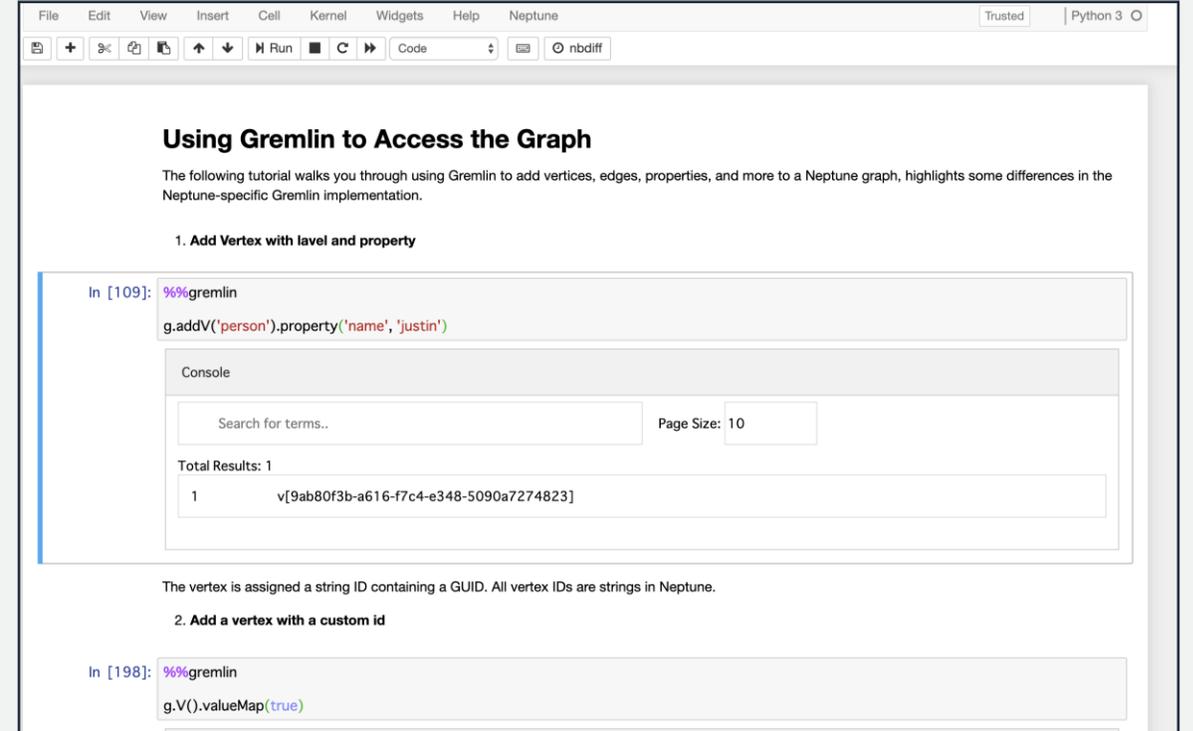
Markus Winterholer / Siemens Building Technologies

はじめるためには



# Neptune Workbench : Jupyter Notebookからクエリ実行

- “%%sparql”や“%%gremlin”を記述したセルでクエリを実行



The screenshot shows a Jupyter Notebook interface within Neptune Workbench. The notebook is titled "Using Gremlin to Access the Graph". The first code cell, labeled "In [109]:", contains the following Gremlin query:

```
%%gremlin
g.addV('person').property('name', 'justin')
```

Below the code cell is a console output window. It features a search bar with the text "Search for terms..", a "Page Size: 10" dropdown, and a "Total Results: 1" indicator. The results are displayed in a table:

Index	Value
1	v[9ab80f3b-a616-f7c4-e348-5090a7274823]

Below the console output, a text block explains: "The vertex is assigned a string ID containing a GUID. All vertex IDs are strings in Neptune." The second code cell, labeled "In [198]:", contains the following Gremlin query:

```
%%gremlin
g.V().valueMap(true)
```

# Neptune Workbench : その他の機能

- 各マジックコマンドでNeptuneの様々な機能をJupyterNotebookから実行
- `%status`
  - Neptuneクラスタの情報を取得して表示
- `%load`
  - S3上のファイルについてURI等を指定し、バルクロードを実行
- `%graph_notebook_config`
  - 使用中のJupyterNotebookの接続情報を表示
- `%query_mode (query|explain|profile)`
  - クエリモードの変更

# [Gremlin] 学習パスの例

- Neptune WorkbenchとGraphexpを用いてクエリを実行しながら学習
- Workbenchのサンプルノートブックを実行してみる
- AWS SamplesのTutorialに沿ってクエリを実行してみる  
<https://github.com/aws-samples/amazon-neptune-samples/tree/master/gremlin/collaborative-filtering>
- TinkerpopのTutorialで学習する  
<https://tinkerpop.apache.org/docs/current/tutorials/getting-started/>
- ※Neptuneにおける実装での相違点に留意しながら実行する  
[https://docs.aws.amazon.com/ja\\_jp/neptune/latest/userguide/access-graph-gremlin-differences.html](https://docs.aws.amazon.com/ja_jp/neptune/latest/userguide/access-graph-gremlin-differences.html)

# [RDF/SPARQL] 学習パスの例

- Workbenchのサンプルノートブックを実行してみる
- 公開されたグローバルなナレッジベースを用いてSPARQLを学習  
例: DBPedia <http://ja.dbpedia.org/sparql>
- 既存のクエリを少しずつ改変しながら学ぶなどが有効。



**Thank you!**