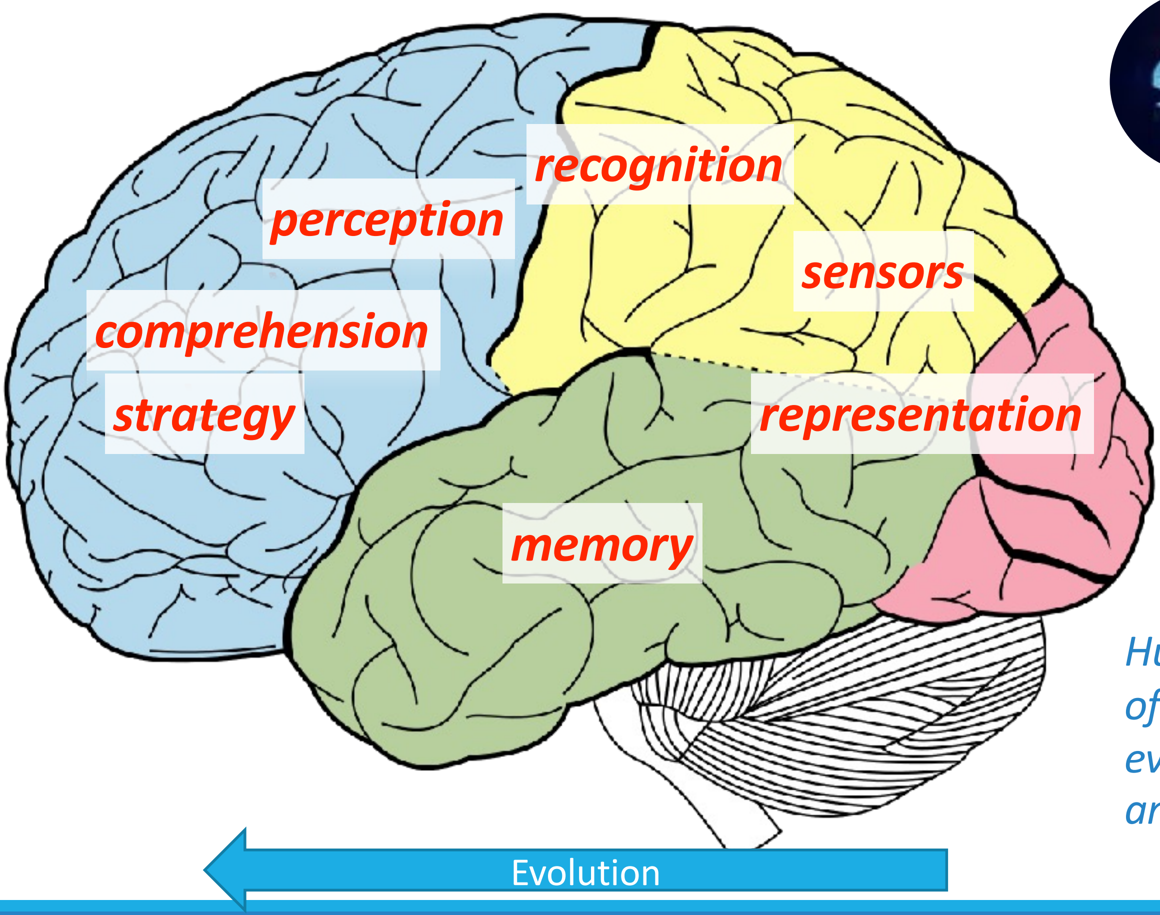
GRAPHEN
An Artificial Intelligence Company

# AI PLATFORM INTRODUCTION

Prof. Ching-Yung Lin
Columbia University

February 16th, 2024

# Ardi Platform




## Ardi AI Platform

Contextual Analysis | Autonomous Learning

**Advanced Enterprise Full-Brain AI Platform to build solutions – Scalability, Stability, and Advanced AI Technologies**

*Human Brain – a graph network of 100B nodes and 700T edges evolved and became smarter and smarter.*

# Ardi's Enterprise- Ready Functions

**Memory**
- Graph Database
- Relational Database

**Perception**
- Graph Analytics
- Feature Engineering

**Learning**
- Machine Learning
- Deep Learning
- Autonomous Model Optimization

**Understanding**
- Natural Language Processing
- Deep Video Understanding

**Strategy**
- Action Strategy Simulation

**Pipeline**
- Production Workflow

**Expression**
- Visualization
- ML Explanations

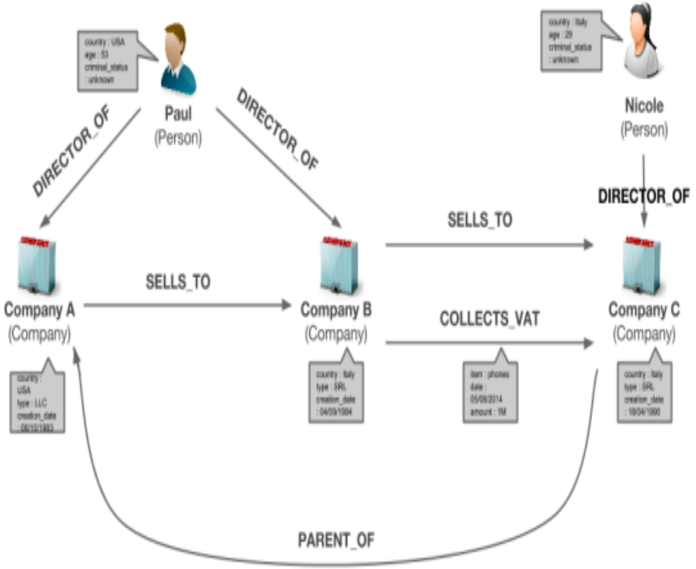
**Reasoning**
- Causality Modeling
- Behavior Prediction

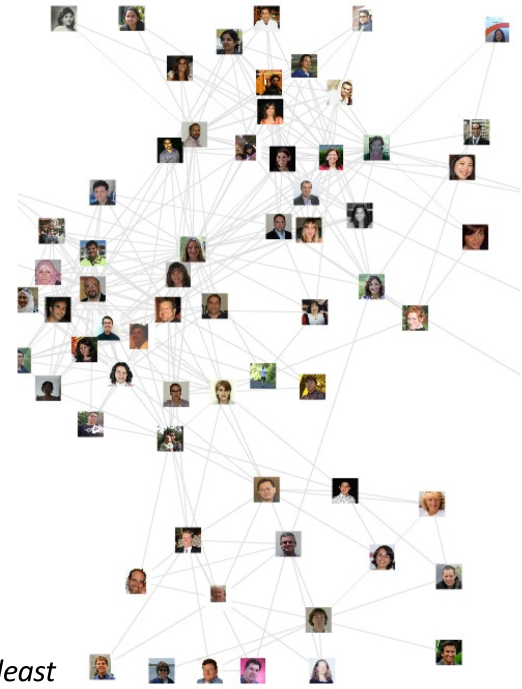
Ardi AI Platform

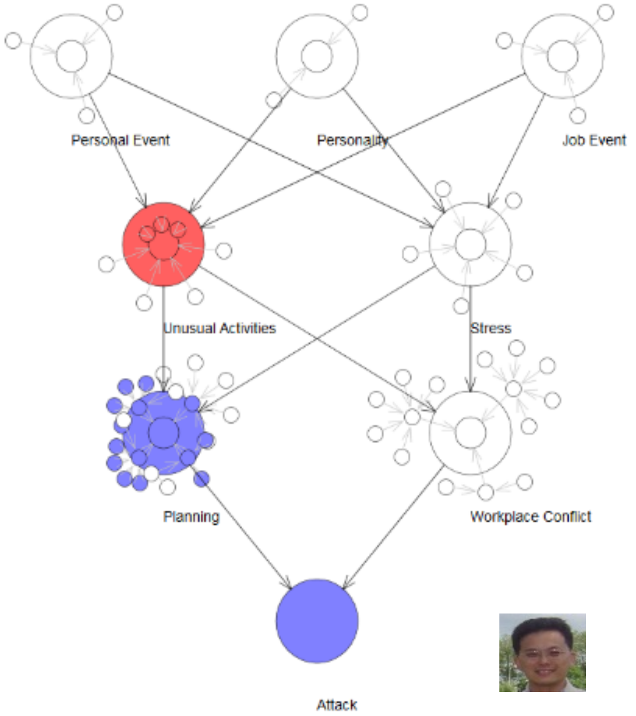
# Three Major Reasons why Graph makes Machine Smarter

GRAPHEN
An Artificial Intelligence Company

| Context | Complex Relationships | Reasoning |
| --- | --- | --- |
| *Attributes and Relationships* | *Topology and Flow* | *Causality and Prediction* |






*Based on 15+ years of AI Security and AI Finance: usually accuracy is at least 2x-3x by context / relation analysis, 10x+ by reasoning / behavior prediction*

# Example: Graphen Financial Industry Platform and Solutions

GRAPHEN
An Artificial Intelligence Company

## Graphen Finance AI Solutions

- Core-Banking Monitoring
- Non-Performing Loans Prediction
- Anti Money Laundering
- Fraud Detection
- Regulation Reasoning & Compliance
- Market Intelligence
- AI Trader

## Graphen AI Financial Industry Platform

### *Multimodal Analysis*

- Behavioral Analysis
- Network Analysis
- Time-Series Analysis
- Flow Analysis

### *Risk Modeling*

- Anomaly Detection
- Risk Assessment
- Bayesian Inference
- Risk Propagation

## Graphen Ardi AI Platform

### *Computation Engine*

- Graph Analytics
- Statistical Computing
- Machine Learning
- Machine Reasoning

### *Visualization Engine*

- Graph Vis
- Statistical Charts
- Machine Learning Vis
- Cognitive Vis

### *Data Engine*

- Graph Database
- Index
- File Storage
- Other Data Storage
- Data Ingestion
- Data Retrieval

# Application Summary

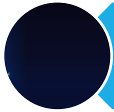
**Products:** AI offerings from the foundational platform to industry solutions

**Main Business Models:** selling valuable industry solutions to business customers or jointly selling solutions with key business partners to users

AI Foundation | Full-Brain Platform

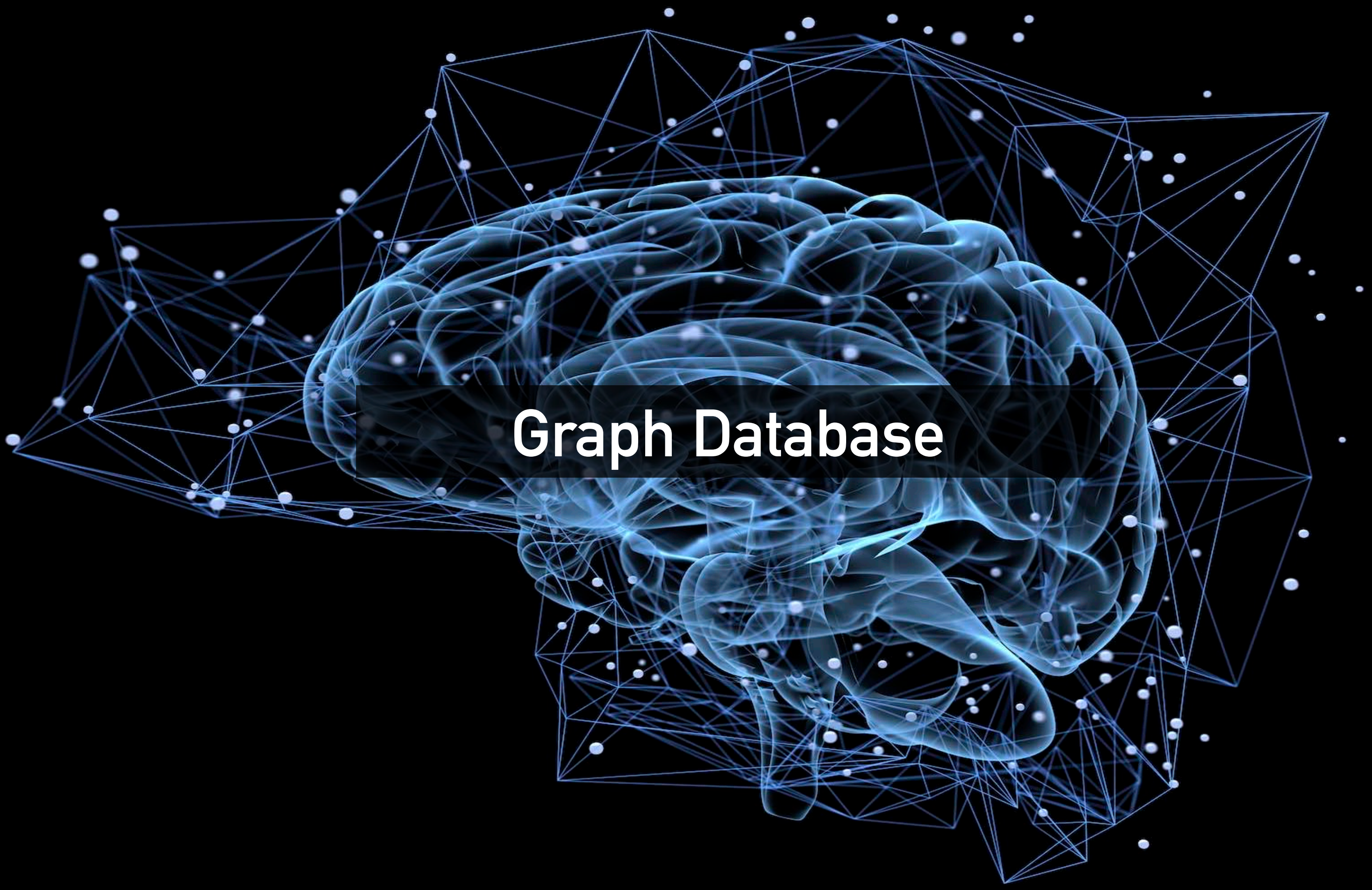
AI Finance | Risk, Fraud, ESG & Intelligence

AI Medicine | Knowledge, Drug & Precision



AI Automobile | Car Doctor

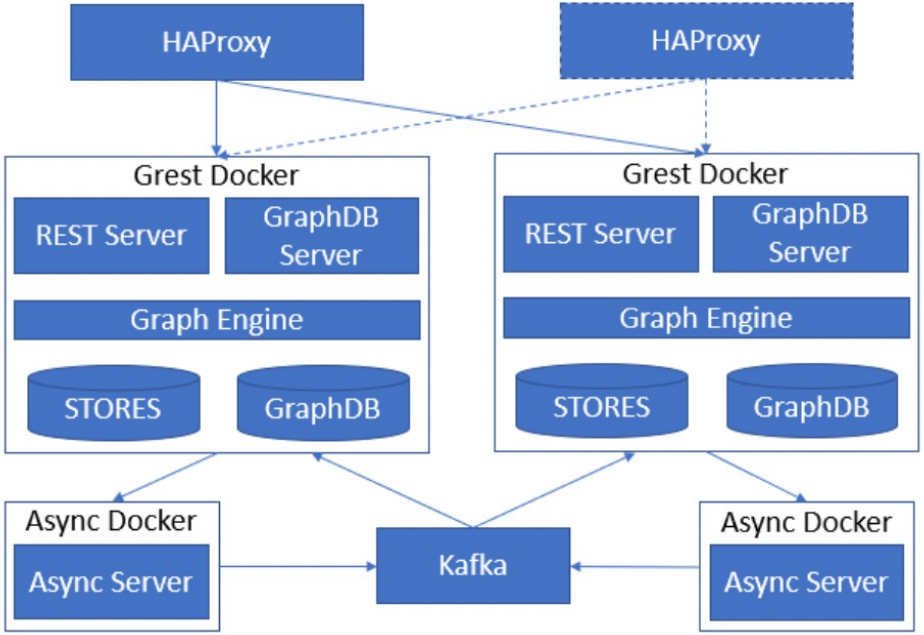
AI Energy | Clean Energy & Smart Grid



# Graph Database

# Ardi Graph Database is a C++-based native DB

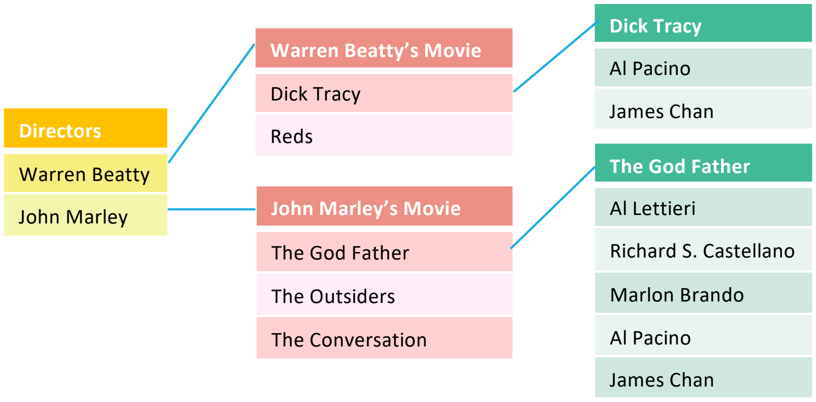
GRAPHEN
An Artificial Intelligence Company



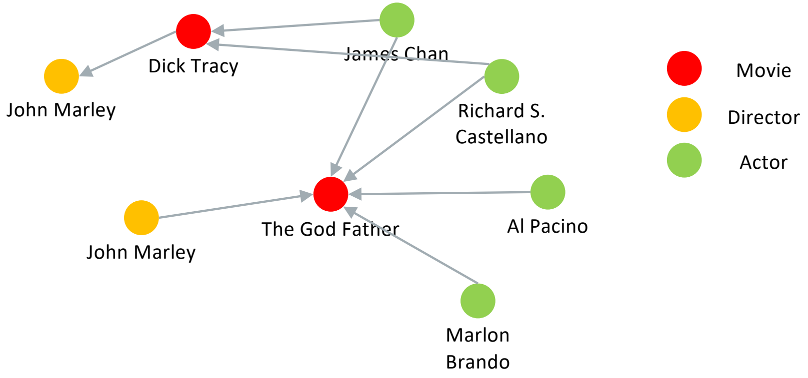

- **The storage layer** contains a high-performance **native graph store based on C++, indexers** to facilitate property-based search, **process store** to keep record of input/output associated with graph analytics and machine learning, and a **storage engine** to manage the graph database, indexes, and data store.
- **The analytics layer** has several computation modules utilizing Ardi platform's graph analytical algorithms and machine learning algorithms.
- **The query layer** includes all the services needed to synchronously or asynchronously load data into the storage layer, invoke graph analytics and machine learning algorithms, and retrieve data and results from running graph analytics and machine learning algorithms.
- **The visualization layer** enables raw graph data or computation results to be retrieved and displayed at the user interface via interactive visualizations.
- **The High Availability proxy servers** guarantees continuous system operations. Communications between layers are achieved via standardized APIs.

# Graph Database vs. Relational Database

- SQL requires expensive join operations to compute neighborhoods of a vertex. Often the number of joins required is proportional with the distance from the source vertex as required by a specific algorithm.




Finding neighbors requires joins operations which may become intractable depending on the size of the database




Finding a neighborhood is logarithmic in the size of the database, thus tractable for any depth required

**=> Ardi Platform supports both proprietary native Graph DB (by C++) and open-source relational database**

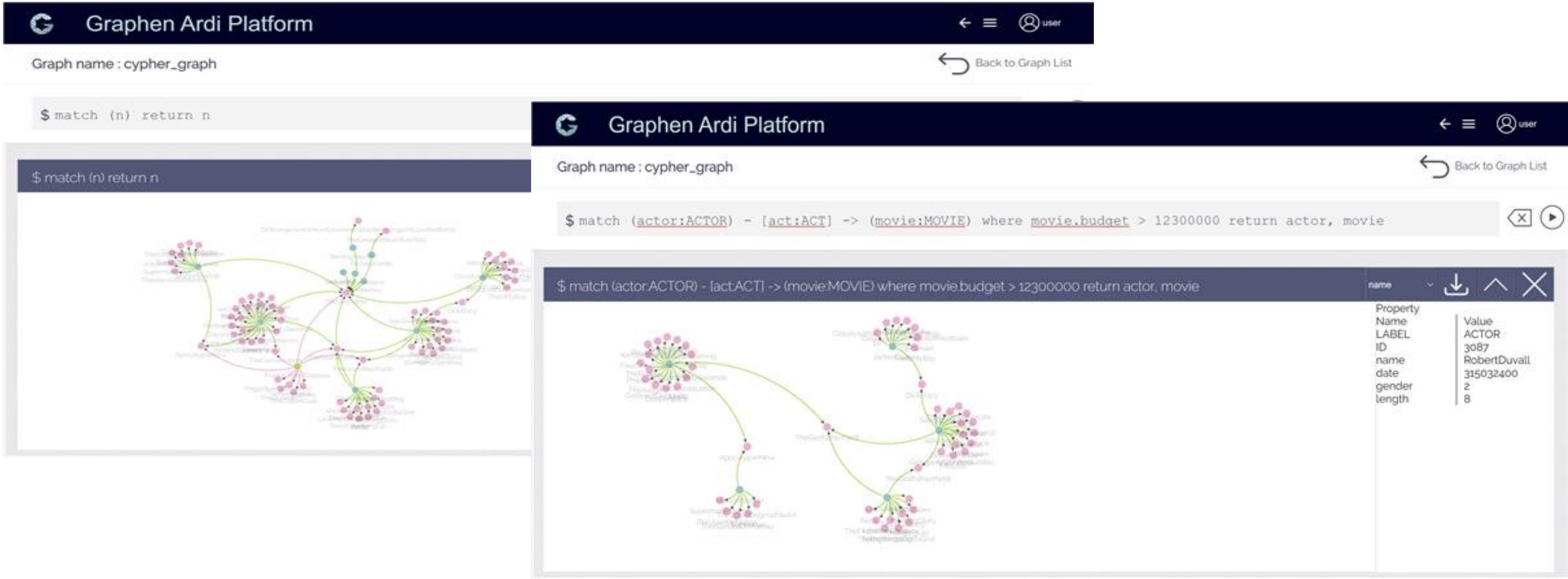
# Graphen Database is Enterprise Production-Ready

Deployed in production in several largest banks in the world (in New York, Honk Kong, Shanghai and Taipei by Dec 2020):

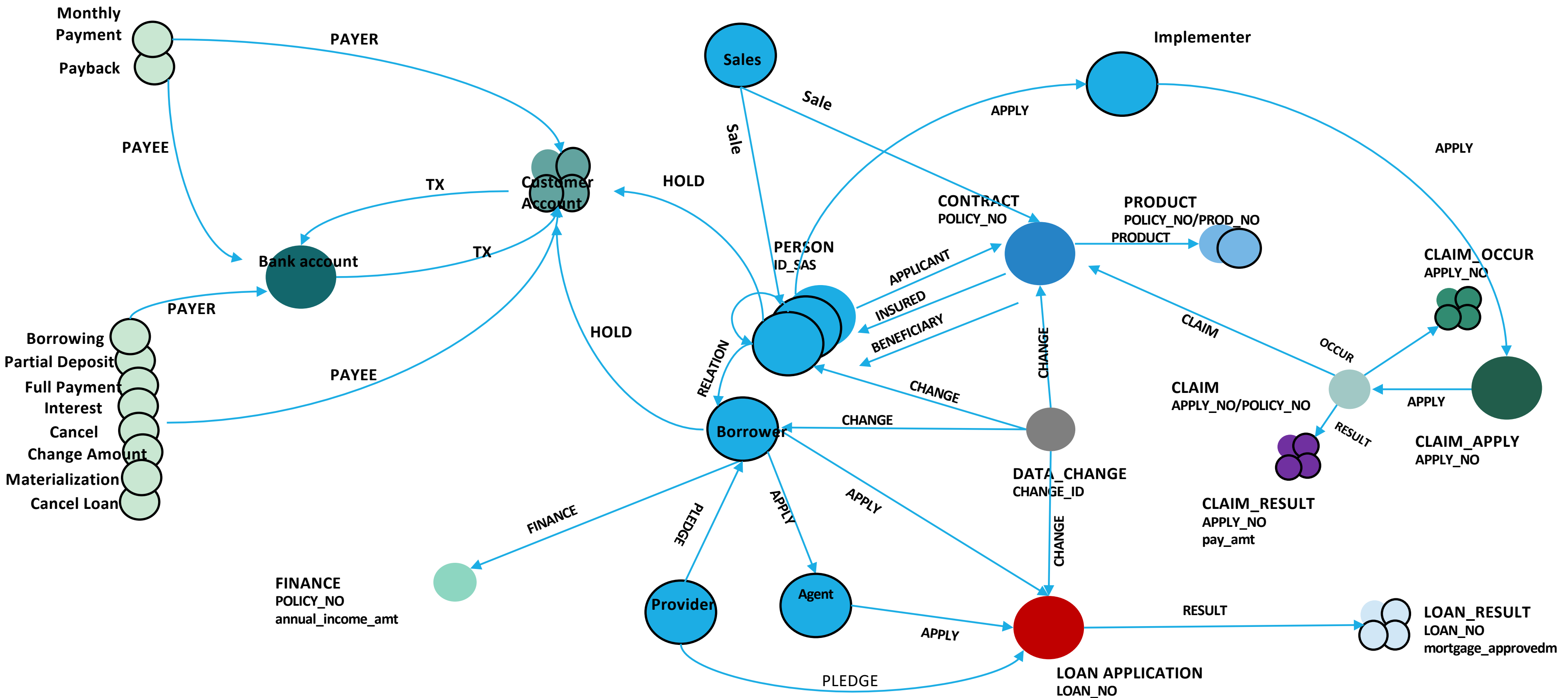
- Terabyte-sized native GraphDB, supports trillion of vertices and edges
- ACID-compliant and distributed Graph database and analytics
- Asynchronous job scheduling (both Autonomous ML and GraphDB)
- Scalable, distributed Analytics, modular and expandable through plugins
- Cluster, Replication and High-Availability with disaster recovery
- Error and event Logging, Monitoring, Backup and Recovery
- Supports both Graph Database and Relational Database
- Supports OpenCypher query language

# Ardi Database's OpenCypher query support

Ardi Database supports OpenCypher query, making it easy to incorporate graph processing capabilities.



# Example: Graphen Graphs for Insurance Fraud



# Ardi Graph Database Comparison

GRAPHEN
An Artificial Intelligence Company

## Performance Experiments on most graph functions common to Ardi, TigerGraph, and Neo4j

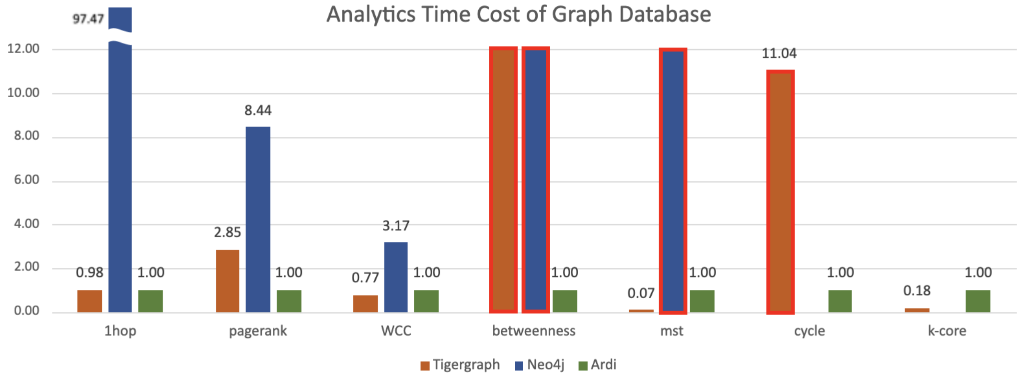
Performance experiment is conducted under following condition

Graph contains
- 2.4M vertices
- 67M edges

Testing Environment
- 32 cores CPU
- 128GB RAM

| | Ardi | Tigergraph | Neo4j | Tigergraph/Ardi | Neo4j/Ardi |
|---|---|---|---|---|---|
| 1hop | 0.0766 | 0.0748 | 7.466 | 0.98 | 97.47 |
| PageRank | 10.829 | 30.838 | 91.384 | 2.85 | 8.44 |
| WCC | 42.004 | 32.401 | 133.134 | 0.77 | 3.17 |
| betweenness | 101.482 | >7200 | >43200 | >70.95 | >425.69 |
| MST | 1163.563 | 76.655 | >43200 | 0.07 | >37.13 |
| cycle | 19.035 | >210.152 | N/A | >11.04 | N/A |
| k-core | 214.512 | 38.814 | N/A | 0.18 | N/A |

- Ardi database outperforms Neo4j, which possess highest market share of graph database in all cases, including 1-hop neighbor, PageRank, Betweenness Weakly-Connected-Component and Minimum-Spanning-Tree. Besides,
- Ardi is competitive with Tigergraph, who claims beating most other graph databases in computation speed. Ardi outperforms TigherGraph in PageRank, betweenness and cycle finding, while being slower in MST and k-core.
- Therefore, we are confident to say that Ardi database is quite competitive among graph databases in the market.




* Neo4j timed out without results after 43000 seconds, and Tigergraph resulted timeout after 7200 second execution on calculating betweenness
** Tigergraph resulted in memory fault after 210 seconds and did not finish the cycle finding yet.
*** Neo4j did not support cycle finding and k-core finding in its Graph Data Science (GDS) library

# Ardi Graph Database Comparison

## Supported Algorithm

| | Ardi | Tigergraph | neo4j |
|---|---|---|---|
| egonet | O | | |
| Centrality | [1] Betweenness<br>[2] Closeness<br>[3] eccentricity<br>[4] peripheral vertex<br>[5] is central vertex | [1] Betweenness<br>[2] Closeness | [1] Betweenness<br>[2] Closeness<br>[3] Harmonic<br>[4] Degree<br>[5] Eigenvector<br>[6] ArticleRank |
| Pagerank | O | O | O |
| K-Core | O | O | |
| (Weakly/ Strongly) Connected Co mponents | O | O<br>(wcc had been removed in version 3) | O |
| Louvain commu nities | O | O | O |
| Triangle counts | O | O | O |
| Cycle detection | O | O | |

| | Ardi | Tigergraph | neo4j |
|---|---|---|---|
| Cliques | O | | |
| Local Clustering Coefficient | O | | O |
| Maximal Independent Set | | O | |
| Label Propagation | | | O |
| K-1 Coloring | | | O |
| Modularity Optimization | | | O |
| Heuristic estimate of graph diameter | | O | |
| Spectral Clustering | O | | |
| Entity Resolution | O | O | O |

# Ardi Graph Database Comparison

## Supported Algorithm

| | Ardi | Tigergraph | neo4j |
|---|---|---|---|
| Single-Source Shortest Paths | O | O | O |
| Yen's K-Shortest Paths | | | O |
| Shortest path between two vertices | O | | O |
| All Pairs Shortest Path | | | O |
| A-star (an improved dijkstra algorithm) | | | O |
| Breadth First Search | O | | O |
| Depth First Search | | | O |
| Random Walk | | | O |
| Minimum Spanning Tree (MST) | O | O | O |
| Minimum Spanning Forest (MSF) | | O | |

| | Ardi | Tigergraph | neo4j |
|---|---|---|---|
| Link prediction | [1] number of common neighbors<br>[2] jaccard similarity<br>[3] jaccard similarity<br>[4] salton index<br>[5] leicht index<br>[6] hub index<br>[7] resource allocation index<br>[8] number of all paths<br>[9] shortest distance<br>[10] katz distance<br>[11] hitting time | | [1] Adamic Adar<br>[2] Common Neighbors<br>[3] Preferential Attachment<br>[4] Resource Allocation<br>[5] Same Community<br>[6] Total Neighbors |
| Similarity ranking | [1] Cosine Similarity | [1] Cosine Similarity<br>[2] Jaccard Similarity | [1] Cosine Similarity<br>[2] Jaccard Similarity<br>[3] Pearson Similarity<br>[4] Overlap Similarity<br>[5] Euclidean Distance |
| k-Nearest Neighbor classification | | O | O |
| Node embedding | | | [1] Node2Vec<br>[2] GraphSAGE<br>[3] Random Projection |