



HELPING THE WORLD MAKE SENSE OF DATA

NEO4J INSIDE: A GUIDE TO NEO4J AS AN EMBEDDED DATABASE

Table of Contents

Executive Summary	1
--------------------------	----------

Part 1: Understanding Embedding and Graph Databases

<i>What Is an Embedded Database?</i>	2
<i>The Modern Context: OEM and SaaS Basics</i>	2
<i>What Is a Graph Database?</i>	2
<i>The Rapid Growth of the Graph Database Market</i>	2
<i>Factors Fueling Graph Adoption</i>	3

Part 2: When Does Embedding a Graph DB Make Sense?

<i>The Big Picture of How Graph Databases Create Value</i>	4
<i>Unlocking the Power of Connected Data</i>	4
<i>Graph Ubiquity</i>	5
<i>Tactical Benefits of Embedding a Graph Database</i>	6
<i>Modeling: Communication and Collaboration</i>	6
<i>Modeling: Flexibility</i>	6
<i>Modeling: Agility in Adapting the Data Model</i>	6
<i>Modeling: Insights from Connected Data</i>	6
<i>Performance: Scalability</i>	6
<i>Performance: Computing Efficiency</i>	7
<i>Performance: Developer Productivity</i>	7

Part 3: The Case for Neo4j – What to Look for in an Embedded Graph Database

<i>Seven Considerations for an Embedded Graph Database</i>	8
<i>Why Neo4j Is the Market-Leading Graph Database</i>	10
<i>Advantages of Neo4j over Other Databases</i>	11

Part 4: Key Use Cases – Understanding the Fit of Graph Databases

<i>Fraud Detection and Investigative Analysis</i>	12
<i>Network and IT Operations</i>	12
<i>Cybersecurity: React Faster and Prevent Attacks</i>	13
<i>Using Graphs to Analyze the Origin of an Attack and Block Potential New Mails</i>	13
<i>Retail and Real-time Recommendations</i>	13
<i>Next-Generation Identity and Access Management</i>	13
<i>Augmented Master Data Management</i>	13
<i>Next Steps</i>	14

Executive Summary

This paper explains why Neo4j is an attractive choice for an embedded graph database.

Part 1: Understanding Embedding and Graph Databases

Part 2: When Does Embedding a Graph Database Make Sense?

Part 3: The Case for Neo4j: What to Look for in an Embedded Graph Database

Part 4: Key Use Cases: Understanding the Fit of Graph Databases

Part 1: Understanding Embedding and Graph Databases

What Is an Embedded Database?

An embedded database is a database used inside another company's application, providing added value and functionality.

The embedded database enhances the functionality of the "host" application, usually without the end user realizing they are engaging with the embedded database.

The Modern Context: OEM and SaaS Basics

As SaaS applications become more dominant, the speed, ease, and flexibility of the embedded approach, i.e., adopting the OEM (original equipment manufacturer) model, becomes an even more compelling strategy for product developers to consider.

The practice, referred to as OEM, is rooted in the hardware market, where physical components are embedded inside a physical product that is shipped to the end user. The OEM is the company that is creating the product. Component suppliers provide their devices to the OEM to help create the product. But over time OEM has become a verb that means to embed.

As the idea has gained popularity in many contexts, the concept of OEM has evolved to be far more general and is used in many realms, not just hardware. One of the fastest growing categories of OEM is embedding software inside someone else's software.

A cloud-based distribution and hosting model bypasses the OEM concept of "shipping" something physical. OEM in the software realm historically referred to the case where a company incorporates a software component into their installed application, usually as source code or a compiled library. The organization then distributed the application to end users as an entire package, generally as a set of binaries. In the modern SaaS model, the embedded software may also run as part of the cloud infrastructure hosting the application.

The OEM model enables organizations to minimize their investment in product development while shortening the product life cycle. By embedding someone else's technology in a product, an organization can enhance its product without straying from its core competency.

Using an embedded graph database enables an organization to simplify the code in their product and deliver powerful features that wouldn't be possible without a graph database.

What Is a Graph Database?

A [graph database](#) is a database designed to explicitly capture and store the relationships between data. It is intended to hold data without constricting it to a predefined model. A graph database stores connections alongside the data in the model, showing how each individual entity connects with or is related to others.

A property graph database like Neo4j also stores properties, that is associated data, for nodes and relationships.

The Rapid Growth of the Graph Database Market

The interest in graph technology is rising rapidly. In parallel with the adoption of artificial intelligence (AI) and machine learning (ML), the data science community is turning to graph analytics to uncover hidden insights. Improving ML feature engineering and providing more explainable AI are just some of the uses driven by graph technology.

Graph databases are the fastest growing database category. The graph database market and the world's understanding of the value of relationships have both grown significantly over the last few years. Graph databases form the foundation of many modern data and analytics capabilities, with more organizations adopting graph databases as the basis for their applications.

Gartner identified graph technology and techniques as one of the [top 10 data and analytics technology trends](#) to help organizations respond to the challenges and opportunities resulting from change and uncertainty in 2021 and beyond.

“By 2025, graph technologies will be used in 80% of data and analytics innovations, up from 10% in 2021, facilitating rapid decision-making across the enterprise.” Gartner, "Top Trends in Data and Analytics for 2021," 16 February, 2021.

Factors Fueling Graph Adoption

Graph technology is quickly emerging as a powerful tool to help the data and analytics community discover relationships, trends, and patterns across disparate and diverse combinations of data. Used in conjunction with graph algorithms and in-graph machine learning techniques, data scientists can find insights across deep relationships with mind-boggling speed.

Graph database growth is spiking with market maturity and is creating an appetite for demand. The adoption of graph technology is accelerating due to several factors.

Speed. Critical business questions that used to take weeks to research can now be answered in minutes. Graph databases enable organizations to solve complex business problems that require contextual awareness and understanding of connections across organizations, people, and transactions faster.

AI maturity. [Advanced data and analytics and AI capabilities](#) are fueling graph database adoption. Graph capabilities underpin advanced data and analytics, ML models, and

explainable AI. The use of graph techniques increases in line with AI maturity.

Ease of use. No-code and low-code tools enable visual exploration and interaction with a graph and allow critical insights to be found.

Cost. Scalability and lower cost processing through cloud-based services are accelerating the adoption of graph databases and graph analytics.

Standardization. [Knowledge graphs](#) expose metadata and business rules, enabling data scientists to quickly identify and use the data they need while preserving context and representing all forms of data in a standard queryable format.

Complete trend, starting with January 2013

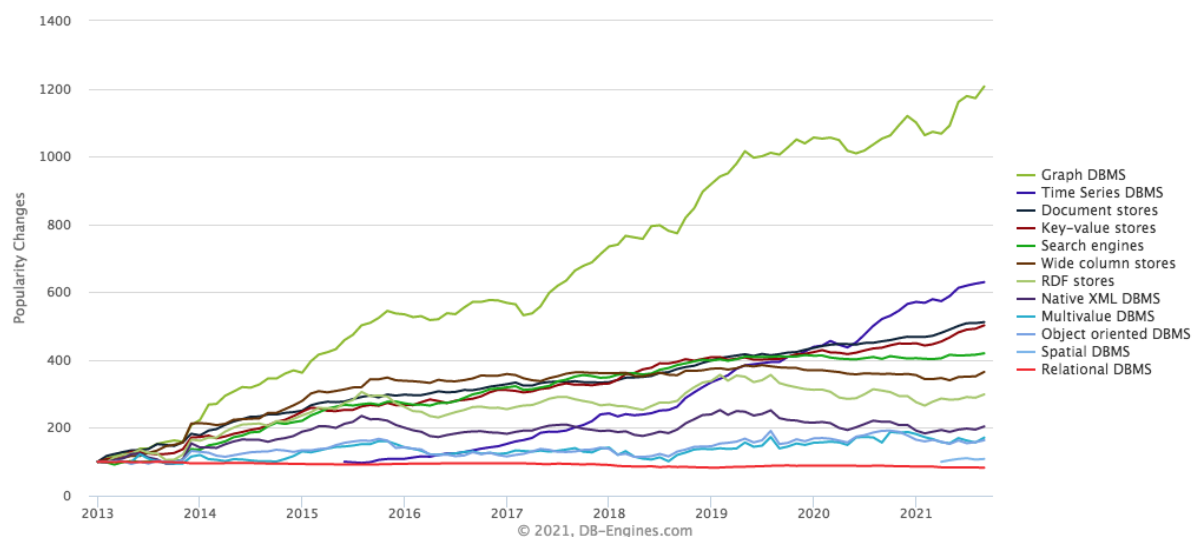


Figure 1. Graph Is the Fastest Growing Database Category (Source: DB-engines.com)

Part 2: When Does Embedding a Graph DB Make Sense?

For all of the reasons just stated, graph databases are a foundational part of a huge number of applications. It is natural that ISVs and other product developers would seek to embed this capability into their products.

When examining the question of embedding, it is crucial to understand when graphs will make a transformational difference in your application.

Graph database use cases and scenarios run the gamut. Common use cases include human capital management; semantic search and data lineage; cybersecurity and fraud prevention; supply chain visibility; and monitoring IT infrastructure. We cover several important use cases in Part 4. By studying these you can understand some of the ways a graph database can fit into your application.

The Big Picture of How Graph Databases Create Value

The high-level story about why graph databases are so attractive can be told quickly by examining two insights:

- Graphs unlock the power of your data.
- Graphs are everywhere, and growing fast.

Unlocking the Power of Connected Data

Since Neo4j introduced its first open source database in 2007, it has been used in hundreds of different use cases as a catalyst for making applications more effective.

Graph databases open a new, richer avenue of understanding and insight for applications by capitalizing on the inherent relationships in the whole dataset. In contrast, more traditional structured data models, which strip much of this context, often hit a plateau as shown in Figure 2. Without a graph database, growth in helping users in new ways had stalled.

With the increased data context that a graph database provides, new questions can be asked and answered, new types of analytics can be created, and new patterns can be revealed. Analysis of the connections in a graph database starts telling you more about everything.

As a result, the power of the applications, and the effectiveness of the use cases supported, starts to grow in new ways.

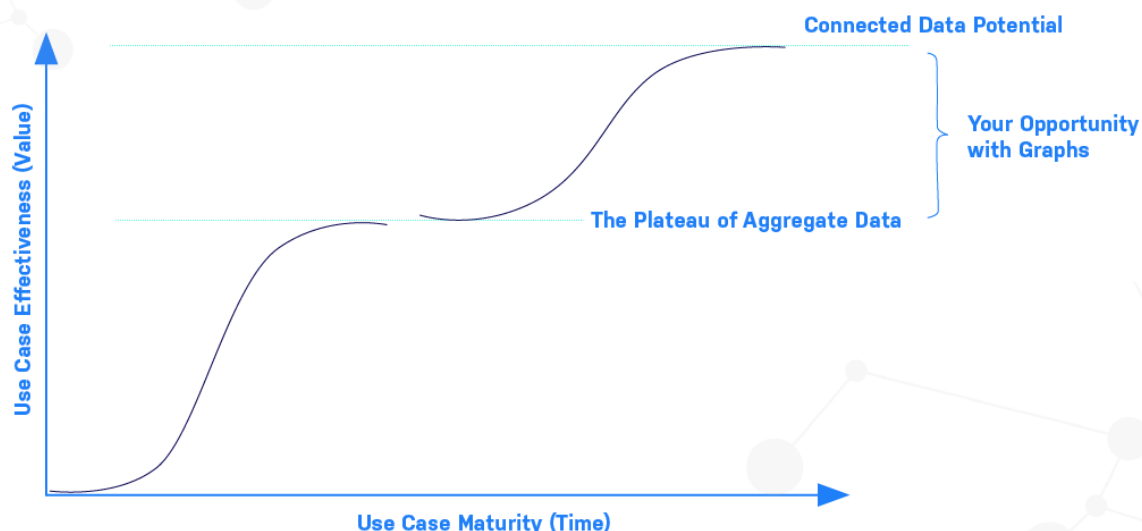


Figure 2. Connected Data Elevates Use Case Effectiveness

Graph Ubiquity

Graphs also dramatically expand the amount of data that can be captured, organized, and analyzed. We have seen applications become more powerful by making better use of the data they have, but also incorporating vast new quantities of data in the following three categories as shown in Figure 3.

Graphs of things. Graphs of things help organizations to build a complete picture of their customers, products, services, and so on. Graphs of things are a great way to start to explore how graphs can add value. For example, you can build a better profile and a deeper understanding of your customers as they connect across multiple touchpoints and lines of business.

The largest Neo4j databases have billions of nodes and connections, representing all the things relevant to a business, whether they are people, products, devices, sensors, or whatever. Most organizations will have nodes and connections ranging from a few thousand on up when they get started. Without a graph, many organizations can't connect their customers across all touchpoints, products, purchases, interactions, and engagements.

Gaining a comprehensive view across these disparate data points will help an organization use these connections to gain a better understanding of who their customers are and how to market to them more effectively.

A telecom might use the same principle, but rather than knitting together the connections between customers and products, they connect all of the things associated with the network. More laterally, an IT department might create a map showing all the devices and apps across different networks to improve an organization's cybersecurity posture.

Graphs of transactions. The next step is a graph of transactions. A graph of transactions shows all the transactions for each customer, with all the products and services a person buys over time. It could be a payment network that connects people sending money to each other. The graph of transactions is a few orders of magnitude larger than a graph of things.

Graphs of activity and behavior. At the largest scale, a business can use graph technology to capture activity and behavior. In this instance, any atom of data in itself is minuscule. The advantage is being able to analyze masses of data to glean conclusions.

Companies haven't mastered the ability to fit millions, or billions of data connections into an affordable repository that supports powerful and speedy queries and analytics. Embedding Neo4j into a product extends these capabilities to any organization using standard machinery.

You can explore different examples of typical use cases in Part 4 of the paper and also on the [use cases section](#) of Neo4j.com.

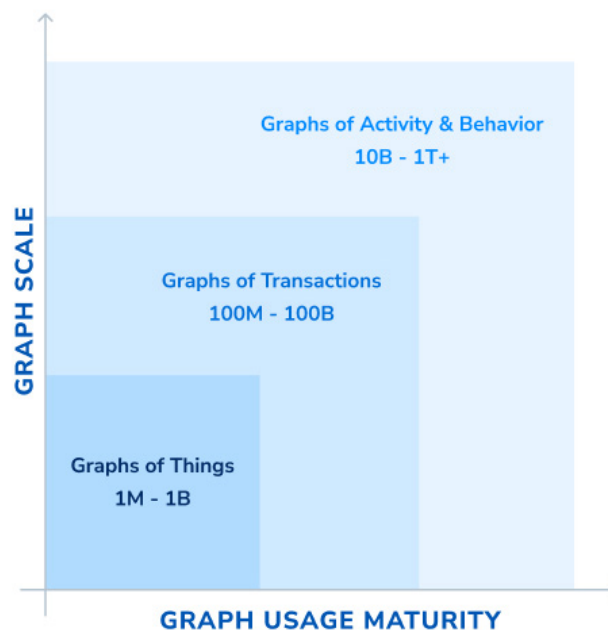


Figure 3. Graph Scaling with Neo4j

Tactical Benefits of Embedding a Graph Database

The high-level benefits just discussed have their origin in the capabilities of graph databases to do more work and make programmers and analysts more productive. We call these the tactical benefits of a graph database -- the numerous advantages of graph databases for performing specific tasks when creating applications and analyzing data.

We can divide these tactical benefits into two categories: modeling and performance. On the modeling side, Neo4j enhances a solution by providing greater flexibility, agility, and ease of connecting disparate sources. With respect to performance, Neo4j reduces queries from days and minutes to milliseconds; it also minimizes the computing footprint needed to run queries while improving developer productivity.

Modeling: Communication and Collaboration

Neo4j provides benefits that both a technologist and a businessperson will understand and appreciate. An organization typically has two very different models in operation. The logical and conceptual side of the house consists of the business and development staff. Database administrators (DBAs), developers, and IT staff fall into the physical side of the house. The root cause of the divide between the conceptual, logical model in which the business and data analysts operate and the physical side of the house, where the DBAs, developers, and IT people operate, is the disconnect between how the data is used by the business and how it is stored in the database.

In a graph database, the logical, conceptual data model is nearly identical to the physical data model used to store the data. As a result, the usual disconnects between the business and IT become less pronounced.

A graph data model aligns conversations between the business and IT: you can show stakeholders how it works.

Modeling: Flexibility

A malleable data model that is easily changed and accommodates sparse or missing data solves myriad development challenges. The inflexibility of the traditional RDBMS model -- rigid, schema-driven, and tabular -- requires predefining a schema for the data. But, in a fast-moving, dynamic real-world application, you need to start working with the data to find out its characteristics, ways to enrich it, and additional feeds that might be incorporated. Using a predefined relational model puts you at a disadvantage because it hinders agility and fast iteration.

With a graph database, instead of defining a schema up front, you create and evolve the data model. You apply a schema when and where desired to ensure that the data stored is not incomplete or inconsistent.

Modeling: Agility in Adapting the Data Model

For people who have been trained to solve problems using tables, thinking in graphs can feel a bit alien. Yet after a very short time working with a graph database, organizations quickly realize that graphs are a natural approach to working with data, and often find many valuable uses for them. Graphs are more like how human beings think and do things.

Using a relational database, a developer builds an application as version 1. Soon after, a flurry of new business requirements come in. These requirements might spring from creativity, responding to a competitor, or reacting to a market condition, whether large or small. Organizations need the flexibility to adapt quickly.

Neo4j's flexible data model enables rapid evolution without disruption. As a result, projects can deliver remarkable outcomes in less time and with less risk.

Having a flexible model allows an organization to add data easily. A new data source such as a cloud data warehouse adds one or more nodes to the graph. It is simple to add properties, nodes, and relationships to the graph data model.

Because of this agility, a graph database like Neo4j thrives on change, making it ideal for high-risk data migration scenarios like shifting to the cloud as well as iterating next generation applications with input from business and IT stakeholders.

Modeling: Insights from Connected Data

Another modeling advantage lies in the new opportunities an organization gains simply by using connected data. The connections between data can be the foundation for new levels of insight about what the data really means. Graph algorithms and AI and ML techniques are able to capture and harvest these insights. The additional context delivered by a graph database, coupled with graph algorithms and AI/ML, provides a very powerful way to harvest insights. Databases based on non-graph architectures lack the additional context needed and can leave organizations at a disadvantage.

Performance: Scalability

Graph databases shine when the relationships between data provide value and information. Graph databases store relationships, adding a new dimension of information that is always available. And while the number of customers grows linearly, the number of transactions per customer and the

number of interactions with a customer may grow exponentially.

This large and growing volume of data can provide valuable insights for recommending products, enhancing customer service, and providing timely offers, all in real-time at the point-of-sale. But within the world of relational databases, deriving these insights is bogged down by rejoining data, complex queries involving multiple hops, inflexible data structures, and analytics processing that can often take minutes.

Neo4j enables users to reduce these types of queries from minutes to milliseconds, dramatically shrinking the time it takes to run high-value queries, as shown in Figure 4. As a result, organizations can take advantage of new forms of value creation across an increasingly complex data landscape.

The more data you have and the more connected it is, the deeper you go past one hop to two or three or more. Organizations can process queries thousands of times faster than even the most robust technologies. The speed equates to much more than efficiency; it completely opens up new possibilities for the business.

As an example, [Marriott International's commerce site processes](#) more than 100 million pricing quotes a day. In the past, every one of those pricing quotes consisted of tens or hundreds of queries.

When Marriott moved their complex pricing workload to Neo4j, they were able to process the same queries a thousand times faster using one-tenth of the computing footprint.

Neo4j also demonstrated its scalability by creating a graph with a trillion relationships, as described in [this video](#).

Performance: Computing Efficiency

Neo4j often requires 10x less compute footprint than other databases when carrying out graph workloads. This includes relational, NoSQL, and RDF databases as well as other graph databases built atop one of those other technologies. As a result, organizations can shave their bottom-line costs, reducing their total cost of ownership (TCO).

Performance: Developer Productivity

Neo4j significantly improves developer productivity and efficiency levels, by leveraging the most widely used graph database query language, as well as providing an extensive array of tools, language drivers, frameworks, and integrations. Coding using the Cypher query language often requires 10x less code than SQL. Cypher originated with Neo4j, but is an open standard, used by dozens of database and tooling vendors.

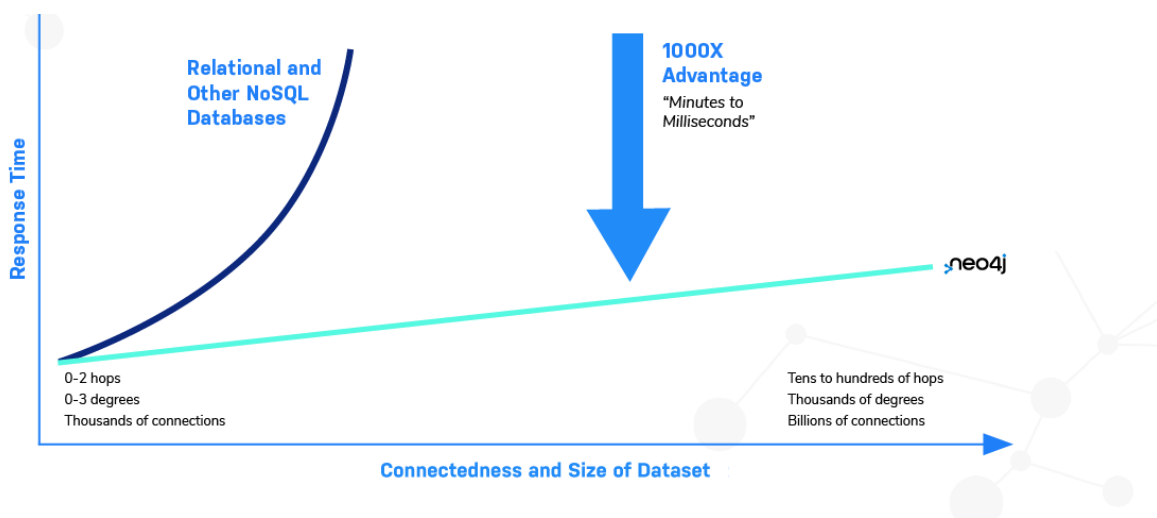


Figure 4. Minutes-to-Milliseconds Performance Advantage

Part 3: The Case for Neo4j – What to Look for in an Embedded Graph Database

In our discussion so far we've covered the power of running your product on a graph database. We've mentioned some specific features of Neo4j that will help. In this part, we make the case for why Neo4j is a safe and mighty choice for an embedded graph database.

Seven Considerations for an Embedded Graph Database

Embeddable	<p>Neo4j is easy to embed in your application. It offers a broad set of drivers for popular programming languages. It can be deployed in a wide range of topologies including multi-tenant SaaS, inside popular containers, and on-premise.</p>
Efficient and Scalable	<p>Neo4j is exceptionally efficient at processing connected data, typically requiring a tenth of the compute footprint compared to relational databases or other NoSQL databases. Neo4j can lower the total cost of ownership because it is easy to embed, has a small footprint, and developers can use Neo4j with their programming language and framework of choice. Neo4j uses Cypher, a powerful open graph query language (also the most popular), which can turn a four-page SQL query into four lines.</p> <p>Neo4j is the only graph database that has a full multi-tenant model that supports multiple active databases. No other embedded graph database enables organizations to achieve physical isolation of data per tenant in a way that makes business-to-business (B2B) SaaS possible. Using Neo4j, it is possible to have hundreds of databases running side by side, each completely physically independent.</p>
Extensible	<p>Neo4j has a thriving global community, providing access to a large ecosystem of extensions and examples, as well as a global talent pool. Neo4j boasts an open library of user-defined procedures and functions and enjoys a rich ecosystem of complementary connectors and solutions that can be integrated for visualization, analytics, business intelligence, and more.</p> <p>Neo4j's user-defined procedures and algorithms enhance its built-in extensibility.</p> <p>Neo4j's flexible data model allows organizations to add data easily and extend the model without schema redesign or database downtime. As a result, organizations can respond faster to user feedback, spinning up new requirements and responding to changing market conditions with little disruption.</p>
Enterprise-Grade	<p>Neo4j is a proven technology used in more enterprise deployments than any other graph database. Its capabilities offer organizations performance at scale, security, fault tolerance, and high availability. It uses a Raft-based architecture that supports scale-up and scale out, across one or multiple data centers.</p> <p>Neo4j runs rich graph queries in milliseconds that would take relational database models minutes or hours. This allows you to respond to events within an application in real time, rather than after the fact. Its superior query performance, multi-database, flexible clustering architecture, and sharding model enable organizations to run millions of queries in a fraction of the time using less computing power.</p> <p>Neo4j protects data and information, providing fine-grained access control for robust management of users, roles, and data access. Neo4j offers organizations the freedom to choose different deployment options, including on-premises, self-hosted, managed, and DBaaS.</p> <p>Neo4j also has a robust and comprehensive global support offering, enabling customers to get questions answered and problems resolved, whatever they are doing and wherever they are located.</p>

<p>Complete Graph Data Platform</p>	<p>Neo4j is the recognized leader in graph data platforms. According to the Forrester Wave for Graph Data Platforms, "Graph is the fastest way to connect data, especially when dealing with complex or large volumes of disparate data. Without graph, organizations have to rely on developers to write complex code that can take considerable time and effort. In some cases, it becomes impractical due to the complexity of data. Graph data platform is a new and emerging market that allows organizations to think differently and create new, intelligence-based business opportunities that would otherwise be difficult to develop and support." (Source: The Forrester Wave™: Graph Data Platforms, Q4 2020)</p> <p>Neo4j's database is surrounded by a rich set of tools and connectors, offering developers, analysts, and business users all that they need to be productive. Neo4j truly is the fastest path to graph.</p> <p>Neo4j also offers the industry's most powerful graph data science solution. Using graph data science, a set of analytic techniques that allows for the exploration of relationships between entities of interest such as organizations, people, and transactions, the data and analytics community can identify unknown relationships in data; they can also analyze data challenging to traditional analytics tools.</p>
<p>Low Risk and Proven Solution</p>	<p>Neo4j is an open-source native graph database that provides an ACID-compliant transactional backend for your applications.</p> <p>Neo4j's success in the embedded market is rooted in its pedigree in the marketplace. Development began in 2003, and the Neo4j graph database has been publicly available since 2007. Neo4j delivers on its commitment to surrounding the product with enterprise-quality integration, tools, education, and support underpinned by a robust partner ecosystem.</p> <p>Neo4j offers a complete graph data platform that includes Neo4j Bloom for visual exploration, Neo4j ETL and Apache Hop integration for data connectivity, and Neo4j Graph Data Science for advanced analytics and AI/ML pipeline integration. Easy to use development tools include Neo4j Desktop and Neo4j Browser along with multiple APIs and drivers to support a broad range of programming and query languages.</p> <p>There is a vast community of talent worldwide experienced with Neo4j. More than 50,000 profiles on LinkedIn reference Neo4j expertise. Neo4j is the most popular graph database by far according to rankings at DB-engines.com.</p>
<p>Solution Value and Differentiation</p>	<p>Neo4j gives organizations the ability to capitalize on new opportunities. Embedding Neo4j opens up the possibilities to stimulate thinking to differentiate your solution further. Neo4j's enterprise license offers security, performance, and reliability unmatched in the industry.</p> <p>Neo4j's graph query language, Cypher, is simple to learn and can operate across Neo4j, Apache Spark, and Gremlin-based products using open-source toolkits including Cypher on Apache Spark (CASP) and Cypher for Gremlin.</p> <p>The Neo4j Graph Data Science Library provides fast, scalable, and parallelized versions of graph algorithms exposed through a simple API. The library contains a growing collection of more than 50 graph algorithms ranging from pathfinding, centrality, community detection, and more that are well-suited for data science problems like link prediction and weighted and unweighted similarity.</p>

Why Neo4j Is the Market-Leading Graph Database

It is no accident that Neo4j dominates the graph database market. Here are five characteristics of the database and the company that keep us ahead of the competition and bolsters a community of over 200,000 developers and data scientists, with tens of thousands of graph projects developed on the platform.

Maturity	As the creator of the native graph database, Neo4j has worked on thousands of enterprise production projects, spanning from real-time fraud detection to cancer research, and from global flight-scheduling to supply chain optimization, in contexts where high-performance, scalable, high-integrity databases are necessary.
Mastery	<p>Neo4j pioneered the property graph and the standards around graph query languages with openCypher. Neo4j is the leader in creating the languages to allow users to query and mine the insights inherent in connected data.</p> <p>As the graph data platform market leader, Neo4j powers more than 2,000 startups, 800 enterprises, 75% of the Fortune 500, and 200K+ developers, all while delivering the definitive foundation for connected data at unlimited scale and developer velocity.</p>
Platform	Neo4j is more than a graph database. Neo4j provides a full platform, with visualization tools, development environment, data science library, native machine learning capabilities, language APIs, and data integrations that include interfacing with external AI/ML pipelines.
Ecosystem	Neo4j is a company focused on enabling developers and data scientists to get more from their endeavors. With its roots as an open-source company, Neo4j relies heavily on its 200,000-member community and its customers to identify the top integrations and partnerships to make its products shine in any environment. To learn more, visit our partner community page .
Advocacy	Neo4j has been downloaded over 100 million times, making it the most adopted graph database by developers and enterprises, enabling developer velocity to build modern, intelligent applications at scale.

Advantages of Neo4j over Other Databases

Speed and agility	The principal reasons why people like graph databases are the speed and agility graph technology brings to understanding data connections.
Schema-optional and nimble	Domain models form the basis of the logic behind applications and how they are built. Domain models depict how applications work, and they look like a graph. The advantage is graphs are lightweight and nimble; they only store what's needed. Neo4j provides a flexible, non-rigid schema that can be adapted and evolved as business needs change, without rebuilding the database.
More agile development	Neo4j offers an inherently more agile development process. Suppose you have an app that requires additional features and functionality based on user feedback. In that case, you can just add those new features and functionalities onto the graph without having to redesign or rebuild the schema, which would be necessary in a relational database.
More conducive to iterative development	A product can be represented by lots of different graphs or mini graphs to enable greater flexibility. There is no need to create a monolithic, rigid model with every area of functionality mapped out in advance. Size is not an issue either; Neo4j can handle many billions of nodes, and into the trillions of relationships .
Designed for developers	Neo4j offers a tremendous number of developer tools, APIs, and support to ensure your embedded project is a success, including drivers for all the most popular programming languages. As a developer-friendly database, it is designed for embedding into other products. Neo4j's first customers used it internally in their applications and as a backend. It's used more often as an embedded database than a standalone database.

Part 4: Key Use Cases – Understanding the Fit of Graph Databases

Graph databases create value in many different ways. By examining the use cases where graph databases have become extremely popular, you can better understand if your use case will be a good fit.

Fraud Detection and Investigative Analysis

Graph technology allows organizations to view a single person's information, and more critically, start connecting that individual to a variety of common elements. For example, it's easy to spot when multiple people are connected to the same email, address, phone number, or bank account; these patterns can indicate fraud when combined with other insight.

Specific examples include [fraud detection and investigative analysis](#). Fraud detection monitors business events and customer data such as new accounts, loan applications, and credit card transactions for current patterns and emergent or new patterns. Investigative analysis tracks bad actors by analyzing phone calls, emails, financial transactions, and suspects in specific locations as well as "skips" or hops in pattern behaviors that indicate something is potentially amiss..

Graph databases also help identify networks of people that are a little less obvious by enabling better visibility into coordinated activity. Detecting financial fraud is vital and can be hindered and delayed by the sheer volume of documents. In the Panama Papers case, a vast amount of information was gleaned by putting the source files into a graph database to reveal shell corporations used for tax evasion.

In isolation, some of these activities appear natural and normal. It's not until you see everything connected that a circular money flow pattern becomes evident. It becomes easier to detect layering, where the same synthetic ID or behavior is noticeable.

For instance, if the same password hash or PIN are all generated from a single mobile device resulting in multiple bank cards getting opened up, that's a pattern that becomes visible through disparate data connections. In isolation, nothing looks out of order. But, when looked at collectively, it reveals that this person now has a 100,000 dollars-worth of credit line because they've opened up 20 credit cards. If all this activity gets generated using the same cell phone or the same laptop, it is likely that a single person is responsible.

Network and IT Operations

Many organizations use graph technology to visualize connections across networks or collections of devices. Visualization can help detect network threats across an

expanding cyber-physical estate. Having a graph-based visualization enables the organization to see that network or group of devices easily.

Neo4j functionality is also used on the backend to do calculations, running algorithms behind the scenes. For example, an organization might use Neo4j to calculate the shortest path between point A and point B in the supply chain. They don't want the graph visualization up front; they need a graph behind the scenes to do the calculations in a performant way.

In other instances, organizations create interactive visualizations of their network to enable somebody to click on a network device, expanding that view to find out what threats exist or where potential threats might emerge. They can also use similar graph visualizations to assess the knock-on impact of a device going offline quickly; how will the downstream be affected if a product goes offline for maintenance, for example.

An industry spin on a similar concept is used in the telecommunications and utility sectors. Telco/Utility System Resource Management analyzes relationships among system components to identify which resources are overloaded, reallocate traffic to reduce risk, and reconfigure the topology to improve operations.

Cybersecurity: React Faster and Prevent Attacks

[Cybersecurity](#) is a vitally important global issue with substantial consequences to our individual and collective safety, as well as human and economic prosperity. The COVID-19 pandemic exponentially changed the cybersecurity landscape. People and businesses are at greater risk of cyber-intrusion today, requiring advanced data solutions to analyze complex connections at scale.

Neo4j is an excellent match for connecting this data – a key technology enabler for collection, analysis, and research. Graph technologies offer new ways to tackle security data and use it to prevent attacks or react faster.

Using Graphs to Analyze the Origin of an Attack and Block Potential New Mails

A zero-day exploit is a previously undiscovered security flaw in a software. Between the moment it is discovered and until the software is patched by those who use it, hackers can use the flaw to compromise systems. The flaw can be used in a phishing attack where a criminal masquerades as a trustworthy entity to obtain sensitive information.

In 2014, an Internet Explorer zero-day exploit ([CVE-2014-1776](#)) became public. Following the announcement of the IE security flaw, a group of hackers sent mails to victims who were asked to login into a website where their identification information was captured.

In this instance, a Neo4j graph was used to analyze the origin of an attack and block potential new mails. After the initial emails were sent, domains used in the phishing attack were collected and stored in Neo4j. Open source information was then added, and graph analysis was conducted to identify potential links to other domains. Based on this analysis, potential threats were identified before they became active.

Retail and Real-Time Recommendations

Real-time recommendations can increase revenue, optimize margins, and improve customer experiences. Retailers use graph databases to power [real-time recommendation](#) engines. Native graph databases, like Neo4j, are ideally suited as the engine underpinning recommendations for retail and commerce sites because they create real-time data connections across disparate data sources, including contextual information.

Graph technology enables the recommendation system to build a profile of a consumer's behavior, preferences, purchase history, inventory, and search terms in real time. As a result, the system can generate a more relevant set of results specific to the user's query.

With graph technology, searches are refined against inventory to represent connections inside data sources based on shopper intent. As a result, the system swiftly builds up a picture of the customer on the fly, working with that portrait as the primary way relevant hits are generated for the consumer.

Next-Generation Identity and Access Management

Managing multiple changing roles, groups, products, and authorizations is an increasingly complex task for enterprise organizations. Identity and access management (IAM) solutions store information about parties (e.g., administrators, business

units, end users) and resources (e.g., files, shares, network devices, products, agreements), along with the rules governing access to those resources. IAM solutions apply these rules to determine who can access or manipulate a resource.

Traditionally, IAM is implemented either by using [directory services or building a custom solution](#) inside an application's backend. Using relational databases for IAM can result in performance, speed, and response issues.

As the number of users grows, custom solutions that use non-graph databases to store identity and access data become slow and unresponsive. Queries take too long because hierarchical directory structures can't cope with the complex dependency structures found in multi-party distributed supply chains.

Graph technology can be used create next-generation IAM solutions. Neo4j enables companies to seamlessly track all identity and access authorizations and inheritances with substantial depth and real-time results. Neo4j provides an interconnected view of all these relationships and dramatically improves organizational insight and control.

Some of the top IAM solutions are built on Neo4j, and many OEMs use Neo4j to build IAM functionality into their applications because it is such a natural fit. Organizations using Neo4j for IAM have significantly reduced query time. Graph database access lookups over large, complex structures execute in milliseconds, not minutes or hours. Learn more about IAM and Neo4j [here](#).

Augmented Master Data Management

Augmented [master data management](#) (MDM) is a common use case for graph technology. Organizations are leveraging graph-powered MDM to unify and connect multiple sources of master data including customer, product, supplier, and logistics information.

Connecting disparate data sources and silos uncovers opportunities and markets. A holistic view of data eases compliance with regulations such as GDPR while enabling data governance.

Neo4j has several successful OEM organizations that offer MDM offerings using graph as a natural extension of their MDM solution. MDM and data quality tools aim to ensure the correct data is available and that the information is clean. Once clean data is available, you can apply a graph on top of it to benefit from looking at the connectivity of everything across those data sources.

Using Neo4j, users can create a 360° view of the master data made available in real time to all operational applications.

That 360° picture can be created by managing all master data inside a single repository or creating a shared metadata repository to support activities ranging from e-commerce to customer support.

Graph databases don't replace traditional relational master data management platforms; they enhance them (that's why it's called augmented MDM). Combining flexible graph technology enables organizations to explore and manage an unlimited number of embedded connections and relationships within master data.

Next Steps

For organizations seeking an embedded solution to enhance their products, Neo4j offers unrivalled leadership and advantages. As an embedded database, Neo4j enables users to take advantage of Neo4j's native integration with other databases, cloud, and compatibility with an extensive Neo4j ecosystem. Neo4j enhancements accelerate developer productivity, providing greater scalability, agility, security, and reliability.

Neo4j is built, designed, and tested to be enterprise-class and ready to be consumed as a fully embedded database. Neo4j is ranked the #1 graph database by DB-Engines, recently claiming a spot on the all-time Top #20 list in the DBMS category. Neo4j has been recognized by the industry-leading, independent market intelligence organization Tech Breakthrough as the winner of the 2021 Data Breakthrough Award for "[Graph DBS Solution of the Year](#)."

Forrester published 2021 Total Economic Impact study of Neo4j, which showed 60% accelerated time to value as average development time shrunk from 12 months to four, and an [ROI of over 400%](#) over three years.

To learn more, contact the Neo4j partner team:
partners@neo4j.com.

Neo4j is the world's leading graph data platform. We help organizations – including Comcast, ICJ, NASA, UBS, and Volvo Cars – capture the rich context of the real world that exists in their data to solve challenges of any size and scale. Our customers transform their industries by curbing financial fraud and cybercrime, optimizing global networks, accelerating breakthrough research, and providing better recommendations. Neo4j delivers real-time transaction processing, advanced AI/ML, intuitive data visualization, and more. Find us at neo4j.com and follow us at [@Neo4j](https://twitter.com/Neo4j).

Questions about Neo4j?

Contact us around the globe:
info@neo4j.com
neo4j.com/contact-us