Graphs in Government: Fulfilling Your Mission with Neo4j

Jason Zagalsky

Executive Summary

At the heart of every enterprise challenge is an explosion in data size and complexity. As the world becomes increasingly connected, so does the data that represents it. The connections in data are as valuable as the data itself. It's increasingly complex, painstaking, slow and expensive to manage and understand the relationships across billions of data connections. Legacy systems (i.e. relational databases) can't keep up. A fundamentally new approach is needed. Graph creates a more intuitive and connected view of relationships, unlocking deeper understanding and context, enabling organizations to quickly and easily uncover hidden relationships and patterns across billions of data connections. Government agencies can gain a deeper understanding of their data using Graph Analytics and Graph Data Science. Neo4j's full graph stack with native graph architecture and global community of data professionals empower data-driven organizations to build better enterprise apps, faster.

This paper explains why Neo4j is an attractive choice for the storage and analysis of large, complex and highly-connected data.

Section 2

Graphs Are Everywhere

Everywhere you look, you'll find problems whose solutions involve connecting data and traversing data relationships, often across different applications or repositories, to answer questions that span processes and departments.

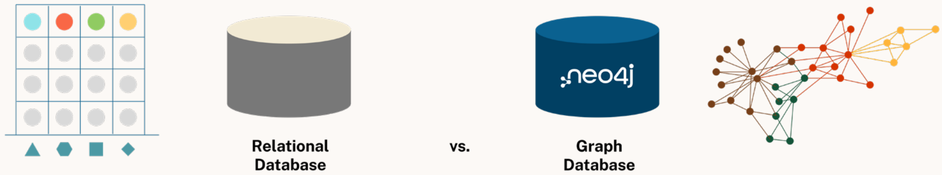
Modeling data as a graph is as natural as drawing connections on a whiteboard. Storing data in a graph offers benefits at scale, for everything from the massive graph used by the U.S. Army for managing strategic assets to recalling NASA's lessons learned over the past 50 years.

Graphs are versatile and dynamic. The use cases for graphs in government are endless. Graphs are the key to solving the challenges you face in fulfilling your mission.

Uncovering the relationships between data locked in various repositories requires a graph database platform that's flexible, scalable and powerful. A graph database platform reveals data connectedness to achieve your agency's mission-critical objectives – and so much more. Using real-world government use cases, this white paper explains how graphs solve a broad range of complex problems that can't be solved in any other way.

The Power of a Graph Database Platform

To understand the power of a graph database, first consider its collection-oriented predecessor, a traditional relational database. Relational databases are good for well-understood, often aggregated, data structures that don't change frequently – known problems involving minimally connected or discrete data. Increasingly, however, government agencies and organizations are faced with problems where the data topology is dynamic and difficult to predict, and relationships among the data contribute meaning, context and value. These connection-oriented scenarios necessitate a graph database. A graph database enables you to discover connections among data, and do so much faster than joining tables within a traditional relational database or even using another NoSQL database such as MongoDB or Elasticsearch.



Good For:

- Well-understood data structures that don't change too frequently
- Known problems involving discrete parts of the data, or minimal connectivity

Good For:

- Dynamic systems where the data topology is difficult to predict
- Dynamic requirements that evolve with the business
- Problems where the relationships in data contribute meaning and value

Figure 1. Relational Database compared with Graph Database

Neo4j is a highly scalable, native graph database that stores and manages data relationships as first-class entities. This means the database maintains knowledge of the relationships, as opposed to a relational database (RDBMS), which instantiates relationships using table JOINs based on a shared key or index.

A native graph database like Neo4j offers index-free adjacency: data is inherently connected with no foreign keys required.

The relationships are stored right with the data object, and connected nodes physically point to each other.

The graph data model is easy to understand, as it reflects how data naturally exists – as objects and the relationships between those objects. It's a model that you naturally sketch on a whiteboard when talking about data, with data elements (nodes or vertices) and the relationships (or edges) between them.

Each node represents an entity, and each relationship represents how two nodes are associated. Property attributes (and indexes) can be attached to both nodes and relationships as well.

By assembling the simple abstractions of nodes and relationships into connected structures, Neo4j allows you to build sophisticated, flexible models that map closely to a problem domain.

As a graph database platform, Neo4j enables government agencies and organizations to do the following:

Perform Deep, Complex Queries

Governments today are challenged with solving complex problems. With the vast amount of data they have pouring in, the answers exist somewhere – but only if you can make sense of the growing volume, variety, and interrelationships of data in disparate sources.

Data becomes more useful once its connectedness is established. Connected data is the representation, usage and persistence of relationships between data elements. Neo4j makes it possible to query relationships across disparate data sources, regardless of the type of data or originating database.

Connected data is the representation, usage and persistence of relationships between data elements.

Neo4j connects multiple layers of data – across processes, people, networks and things. Once you've connected layers, you gain intelligence downstream and provide a connected view of the data to analytic and operational applications. You also obtain context that allows you to more deeply or better refine the pieces of information you're collecting. The better your understanding of data connections, the better your downstream insights will be.

Neo4j empowers government agencies and organizations to iterate and expand on current datasets, gaining momentum to execute on bigger and better ideas, and find deeper contextual meaning in the data. Using graph technology, you can increase the number of hops (the levels of connections) between data without a corresponding increase in compute cost. As a result, you gain higher degrees of context not easily achieved by JOINing three or four tables together in an RDBMS.

Neo4j's architecture enables these deep, complex queries. The enterprise-grade, native graph database is built from the ground up to traverse data connections at depth, in real time and at scale.

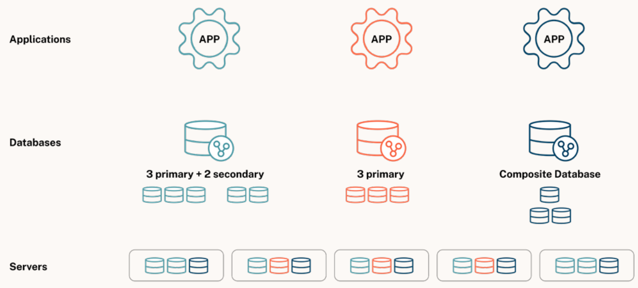
Unlimited Scalability

As an enterprise scale data management platform, Neo4j contains scalability features that enable your application to scale easily and operate more securely and efficiently.

Along with vertical scaling by adding memory or additional CPU cores, Neo4j can scale horizontally to 100's of server instances to accommodate large graphs or to handle high concurrency workloads

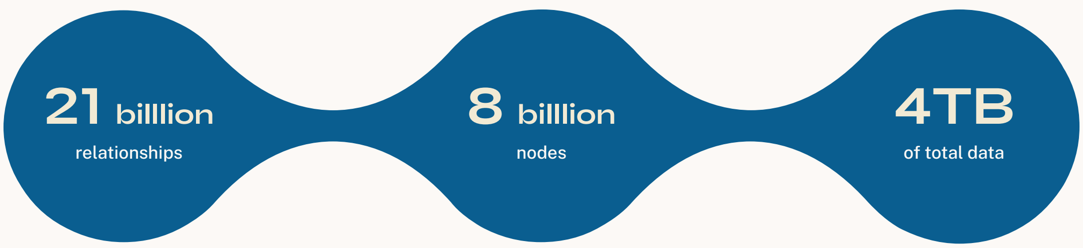
Multi-Cluster Composite Databases enable the unification of data silos and also accommodate partitioning (sharding) on very large graphs and queries across multiple shards.

Neo4j's Autonomous Clustering makes it easier to run and administer fault tolerant and highly available Neo4j clusters and to add and remove instances from the cluster to accommodate changes in workload volume. Autonomous Clustering will rebalance and redistribute copies of the data in the event that one of the cluster instances fails maintaining the desired number of high-availability instances.



Neo4j at scale at the US Army

Database stats



Reduce Infrastructure Costs

Your government agency runs on a lean budget. Any opportunity to reduce infrastructure spending frees up resources to focus on the core mission. A graph database does just that.

It delivers deep, complex queries with less hardware, which means reduced costs. Due to the nature of index-free adjacency and compression achieved with

Neo4j's native graph storage format, the standard, highly available Neo4j installation is 3-5 servers, versus an RDBMS with a graph layer, which requires about 50 servers for the same scale. With this efficiency, Neo4j also requires fewer licenses, further reducing database costs. Neo4j offers deployment flexibility, with servers on premises or in the cloud.

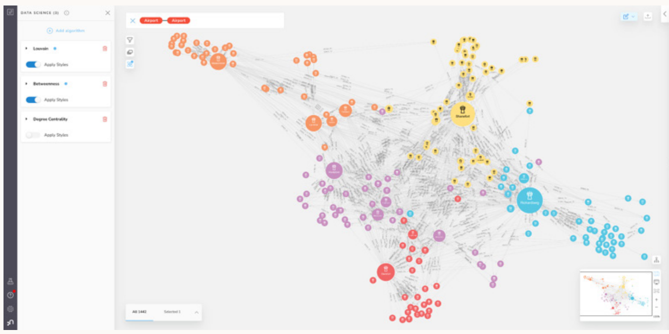
Gain a Deeper Understanding of Data with Graph Data Science

Structuring data as a graph makes it possible to explore billions of connected data points in seconds and identify hidden relationships that can help improve predictions. Graphs are powerful when it comes to computing a shortest path, clustering like entities, identifying influencers, outliers, performing identity resolution, or predicting a missing link. Neo4j provides a library of 65+ pre-tested and ready to use graph algorithms for both supervised and unsupervised machine learning, along with a platform for testing and deploying ML models, and visualizations that help your teams answer questions like what's important, what's unusual, and what's next.

Neo4j's Graph Data Science algorithms run internally in the database so that you don't have to move data from the graph into another platform to perform AI or Machine Learning.

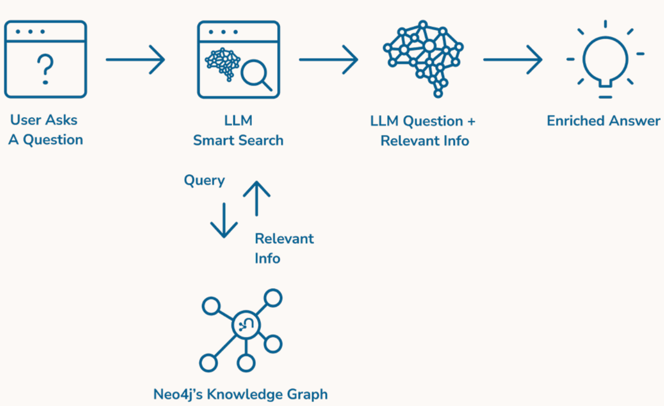
Visualize Data Connections

Neo4j's Bloom graph data visualization illuminates your data and its connections. Bloom allows you to see, explore, enrich, analyze, and share your graph data without having to write a line of code. With Bloom you can visualize the patterns in your data using features such as advanced filtering, coordinate (spatial) layouts, and built in graph data science algorithms.



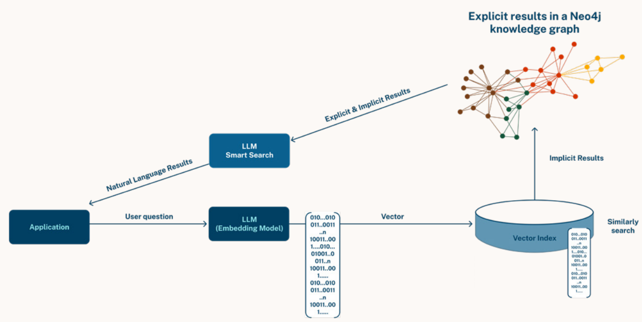
Unleash the Power of Knowledge Graphs for Generative AI

Generative AI can be a game changer for human creativity and productivity. But one challenge is that Large Language Models (LLMs) are trained on open source data that can be dated and inaccurate. When asked to attribute an answer to a source, an LLM comes up dry. Knowledge Graphs are a popular way to ground LLM responses with industry specific information that is relevant, up to date, and not necessarily available on the open internet.



It is possible to improve the accuracy, relevance, completeness, explainability and traceability of GenAI enterprise specific knowledge. This process is known as RAG or Retrieval Augmented Generation.

Using a Neo4j Knowledge Graph for RAG enables you to ground LLM responses in validated facts. Neo4j makes it easy to add to and update information in the graph, indexes, and RAG sources using our many supported data connectors and a developer friendly schema that never requires redesign. Graph structure makes finding factual answers to multi-hop questions possible through traversals that follow a breadcrumb trail of connections.



Neo4j is the ONLY knowledge graph that has native vector search. Knowledge graph queries provide explicit accuracy from your data, while vector search offers implicit responses using semantic meaning.

Using a Knowledge Graph provides context to GenAI query responses, integrates with your enterprise data and associated data governance requirements, and provides a seamless way to leverage the power of GenAI with your data.

Maximize Value from Existing Resources

A rip-and-replace approach is a non-starter for most government technology projects. By connecting data across diverse existing data stores, Neo4j leverages the value of all your existing systems. And when it's time to replace aging applications, government contractors and agencies find that Neo4j is a cost-effective agile foundation for new initiatives.

Deliver Immediate Answers at Scale

Government agencies and organizations must store massive amounts of data and need answers fast.

Neo4j delivers a 1,000x performance advantage over relational and other NoSQL databases hosting graph engines, reducing response times from minutes to milliseconds for queries of graphs containing billions of connections.

Neo4j traverses any level of data in real-time due to its native graph architecture. RDBMS and other NoSQL databases typically see a significant performance degradation when traversing data beyond three levels of depth.

Meet Security Demand

Neo4j fulfills the stringent security demands of government customers. In addition to meeting Federal Information Security Modernization Act (FISMA) requirements, Neo4j's advanced security architecture supports data encryption at-rest, in-transit and intra-cluster, LDAP and Kerberos integration and Single Sign-on (SSO). With Neo4j's built-in schema & role-based access control (RBAC) you can define access policies by role or user and add constraints on nodes, labels, relationships, or properties – you can even wall off specific parts of the graph or limit traversal depth.

Neo4j is ISO 27001 Certified and AICPA SOC2 Type 2 compliant.

Neo4j is approved to run in a classified environment by many Department of Defense and Intelligence Community agencies. Authority To Operate (ATO) has been granted for several applications that are built on Neo4j running on classified networks. Many civilian agencies have Neo4j approved to run on their networks and cloud environments as well. Neo4j can easily be deployed from the AWS, Google and MS Azure cloud marketplaces, including their GovCloud (FedRamp) marketplaces.

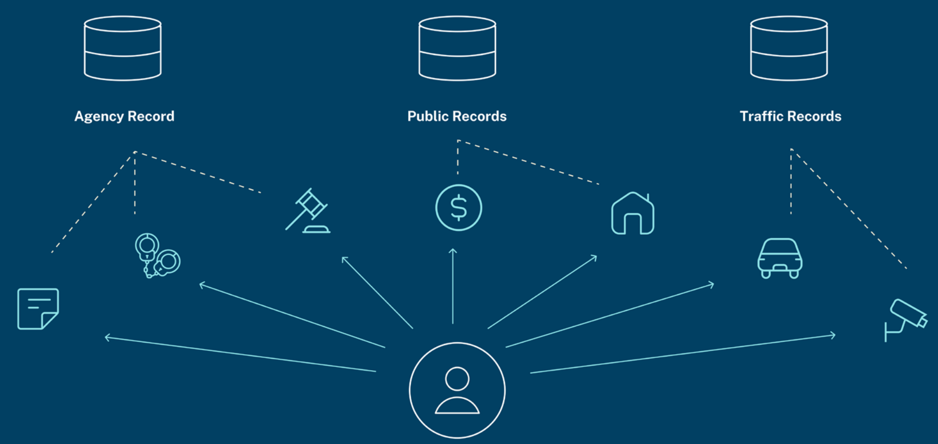
Connected Data in Criminal Investigations

Criminal investigations highlight the value of connected data, because connections in data point to potential suspects in a case. A suspect often appears in several different databases. Connecting that data is key for investigators

to find out all they can about a suspect through phone records, financial transactions, fingerprints, DNA, court records, associates and more.

Separate data silos of people, objects, locations and events (POLE) aren't useful if you're doing a criminal investigation or trying to stop a terrorist attack. You need the relationships that span those data silos and contextualize the activities and associations among suspects.

The idea hinges on who knows who. If Person X has come to attention of the authorities for whatever reason, who else in Person X's network might be of interest? This complexity is hard to capture and explore through conventional database technologies like RDBMS. Graph database platforms excel at analyzing connected data.



A suspect often appears in several different databases.

Bystander investigated due to deep connection found

Graph Databases in Action

Mission objectives vary widely from agency to agency, and even within agencies. Some missions are internal-facing; they help the agency run more efficiently. Others are external-facing and directly impact constituents. The beauty of a graph database is that it supports many diverse applications. Agencies should imagine what they want to do and what data they need to accomplish extraordinary goals, because it's probably possible.

Here are several real-world case studies to get you thinking about how a graph database helps your agency meet its mission objectives. Don't let these examples limit your imagination, but rather use them to imagine other possible use cases – remember, **graphs are everywhere!**

Key Benefits	Minutes-to-milliseconds performance, scalability, high availability, data integrity, model flexibility, built-in AL/ML algorithms, developer productivity and hardware efficiency.
Key Features	Native graph storage and query processing, ACID compliance, causal consistency, enterprise-grade security, autonomous clustering, composite databases (sharding), change data capture, browser based Desktop IDE, drivers for popular languages and platforms, GraphQL, connectors for Spark, Databricks, Kafka, Snowflake, Big Query and other data sources, ODBC/JDBC connectors for Business Intelligence, 80+ built-in algorithms for AI/ML, vector indexing, Bloom graph visualization.
Deployment Options	On-premise, public or private cloud, GovCloud, hardened containers or Neo4j Aura managed service.

U.S. Army

Supply Chain, Bill of Materials and Maintenance Cost Management

As the largest branch of America's Armed Forces, the U.S. Army supports more than one million soldiers and about 200,000 civilian staff. Each of these staff members relies on multiple pieces of equipment, from helicopters and armored vehicles to small arms and radios, to complete their missions.

With maintenance, operation and support costs of equipment representing as much as 80% of total lifecycle costs, it's imperative that the Department of Defense (DoD) track and analyze equipment maintenance costs.

Maintenance and support for this equipment necessitates the procurement of millions of spare parts per year. Prior to Neo4j, the Army used an aging mainframe-based system to track parts orders and their connections to equipment systems, components and subcomponents.

However, an increasing volume of available data and changing historical data sources made data management increasingly difficult, resulting in unpredictable maintenance costs. It was obvious that a system with more flexibility would offer greater performance and the ability to add new dimensions for more insights and richer analysis.

The Army recognized the need to modernize its core tracking system. Working with CALIBRE, a leading provider of management consulting and information technology solutions, the U.S. Army employed Neo4j as a major part of its solution for providing greater visibility into the total costs of owning a system.

With Neo4j, the Army has a much more flexible and robust view of the parts requirements and costs of these parts across systems, components and subcomponents. It's also much easier now to rapidly store, explore and visualize a wealth of logistics and cost data.

"The scale of the information Neo4j handles is vast," said Preston Hendrickson, who leads CALIBRE's technical team for the U.S. Army's Operating and Support Management Information System. "For example, just one of the tanks we track includes ~10 million parts records, creating more than 15 million possible relationships among the components for our cost allocation algorithms to work through.

> *The scale of the information Neo4j handles is vast. For example, just one of the tanks we track includes ~10 million parts records, creating more than 15 million possible relationships among the components for our cost allocation algorithms to work through.*
>
> – Preston Hendrickson, leader of CALIBRE's technical team for the U.S. Army's Operating and Support Management Information System

"The flexibility and speed at which we can now add in new data sources or make changes to the structures of current data in Neo4j has been a real game changer for our IT team.

"Neo4j also saves our analytics team huge amounts of time. The graph now serves as an analytics platform that is capable of housing everything they need together in one place."

"This is giving us visibility into more detailed connections within our data that were previously much harder to find or perhaps sometimes even overlooked. Analysts can now look for answers to their questions and perform 'what-if' scenarios immediately without having to load data from multiple sources and in some cases reload a mainframe for repeat computation."

Army leaders wanted to rapidly query connected data to:

Forecast	Forecast the need for replacement parts considering their theater location and climate
Calculation	Calculate mean time to failure rates given the context aboveBloom graph visualization.
Comparison and Analysis	Perform multidimensional cost comparison and trend analysis
Information	Inform the Army's logistics and budget requirements processes
"what-if" Questions	Answer vital "what-if" questions such as the cost of deploying certain forces and supporting equipment to a new war zone

MITRE

Fighting and Tracing Cybersecurity Threats

In their efforts to stop cyber attacks, analysts track large amounts of detailed information, such as network and endpoint vulnerabilities, firewall configurations and intrusion detection events. The solutions used to analyze this data typically track data points. But to be successful, analysts need to understand how those data points are related.

To address these challenges, researchers at MITRE Corporation, a U.S. federally funded, not-for-profit company, used Neo4j to develop CyGraph, a tool for cyberwarfare analytics, visualization and knowledge management.

CyGraph brings together isolated data and events into an ongoing overall picture for decision support and situational awareness. It prioritizes exposed vulnerabilities, mapped to potential threats, in the context of mission-critical assets. It also correlates intrusion alerts to known vulnerability paths and suggests the best course of action for responding to attacks. For post-attack forensics, CyGraph shows vulnerable paths that warrant deeper inspection.

The model schema in the CyGraph architecture is free to evolve with the available data sources and desired analytics. The data model is based on a flexible property-graph

formulation implemented in Neo4j. REST web services provide interfaces in CyGraph for data ingestion, analytics and graph visualization.

The Neo4j native graph pattern-matching language supports a library of domain-specific queries as well as flexible ad

hoc queries. CyGraph then provides a variety of clients for specialized analytic and visual capabilities, including graph dynamics, layering, grouping, filtering and hierarchical views.

"Graph queries make it possible to focus our analysis on the relevant portions of attack graphs, allowing us to pinpoint vulnerabilities and target responses," said Steven Noel, a cybersecurity researcher at MITRE.

The use of Neo4j at the MITRE provides insight into the mission impact of cyber activities. Graph layers (network infrastructure, cyber defense posture, mission dependencies and so on) define subsets of the overall model space with connections within and across each layer. Analysts also gain visibility into operations for global situational awareness.

> *Graph queries make it possible to focus our analysis on the relevant portions of attack graphs, allowing us to pinpoint vulnerabilities and target responses.*
>
> –Steven Noel, Cybersecurity Researcher, MITRE

MITRE

Lockheed Martin Space

Product 360: Design and Supply Chain Traceability

Some products are so complex and expansive that they are better described as a community of parts. In cases like these, the product is only as good as the sum of its parts, and

the parts are only as good as the processes, suppliers and engineering designs they are based on.

The challenge, as Lockheed Martin Space (LMS) found, is that without visibility into the entire product ecosystem – think

of it as Product 360 – there's no way to easily gauge the downstream impact of a single variance. LMS faces a huge challenge in analyzing data from numerous systems, from engineering to ERP to procurement.

LMS wanted to integrate all of its processes and data across the entire life cycle of its highly complex products. To do so, it had to determine how to connect data across disparate data storage systems. As the LMS team sketched out the problem, the diagram with circles and lines looked like a graph.

"We went searching for graph databases, and we found Neo4j," said Ann Grubbs, Chief Data Engineer at Lockheed Martin Space.

LMS is connecting all of its data silos by storing the relationships between the data and those systems in the graph database. Users will be able to submit queries to an application, which will traverse the graph database. Neo4j guides the application to the appropriate legacy system to drill down. The data remains in the current system, with relationships stored in Neo4j.

In the past, answering a question across systems could take weeks. Interfaces between a few key systems were built, but those interfaces were expensive and not scalable for an organization with thousands of datasets.

By connecting all their data, they get immediate answers to critical questions. "We have tons of telemetry data coming in and artificial intelligence analyzing it," said Grubbs.

"If we see a problem emerging in a particular area, we need to know everything there is to know about that immediately," she explained. "Who can we call? What happened in test?

What did the engineering look like? We need a quick picture of everything in order to respond to that. We can't wait two weeks to find out why a part is failing."

The graph database enables LMS to perform an impact analysis to determine the downstream result of an issue or change anywhere in the product lifecycle. For example, if there's a delay in engineering, how will that impact the overall schedule for the product launch?

The graph database also drives efficiencies for troubleshooting.

"In the past we've had someone manually identify the root cause of a failure," Grubbs explained. "They'd identify all the things that could have influenced that part and caused it to fail. Is it engineering? Procurement? Supplier? Is it a vendor issue or a manufacturing defect? The idea is to let the graph do those traversals and find variance in the process and report it back instantaneously versus a human taking weeks to do it manually."

Similarly, Lockheed Martin identifies potential process improvements.

"We use computer-aided design (CAD) systems, and there's a complexity rating to CAD models," said Grubbs. "Using Neo4j, we figure out if it's really worth spending the time and money to get to the next rating. Will making a design more complex improve quality or not?"

But this is just the tip of the iceberg.

"We have far more data than humans alone could ever understand or manage," said Grubbs. "With Neo4j, we've been able to put it in a perspective that makes sense to anyone all the way up to the CEO. There's a lot of interest from the business. They have all kinds of business cases lined up, ready to go."

> *If we see a problem emerging in a particular area, we need to know everything there is to know about that immediately. Who can we call? What happened in test? What did the engineering look like? We need a quick picture of everything in order to respond to that. We can't wait two weeks to find out why a part is failing.*
>
> –Ann Grubbs, Chief Data Engineer, Lockheed Martin Space

NASA

Knowledge Graph of Historical Lessons Learned

NASA has been collecting project data since the late 1950s. Locked in that data is knowledge that holds incredible value to help cut down on project time, enable engineers to identify trends that can prevent disasters and incorporate lessons learned into new projects.

But accessing that information is a challenge due to silos between departments and within individual groups, products and programs. NASA needed to break down those silos, which is exactly what it achieved using Neo4j.

NASA's Lessons Learned Database is part of the organization's knowledge management strategy for how it collects, stores and shares information. Engineers use this

database to learn about past projects, including any mistakes or successes and what actions were taken.

"We started adding lessons in about 1990, and they went up and down until around 2003, when we had a shuttle disaster that resulted because of a thermal tile malfunction," explained David Meza, Chief Knowledge Architect at NASA. "If we had had this information beforehand and understood the trends better, we might have been able to prevent the disaster from taking place."

Previously, the database was made up of less than 1% of the organization's 20 million documents. With a total of 80,000 employees, the volume, variety and velocity of data was taxing the system. NASA needed a better way for end-users to access this information.

Meza started looking at how to take the documents and convert them into graphs. Because there was a lot of metadata associated within the lessons, Meza correlated the topics based on their self-assigned categories. He could see each lesson with its topic as well as correlations between topics, so he could also see how topics correlated to one another.

This allows users to look at trends, which can potentially help NASA engineers prevent disasters like the aforementioned shuttle disaster.

Meza developed a simple graph model to showcase the data to end users. Engineers quickly perform searches and pull the information they need. They also jump from one part of a system or subsystem to another and see the connections between the subsystems. Similarly, project managers use the system to look at information pertaining to various subsystems handled by disparate team members and pull it all together to understand the entire system.

> *The Lessons Learned database has saved us at least a year and over $2M in research and development towards our Mission to Mars planning*

–David Meza, Chief Knowledge Architect, NASA

Meza developed a simple graph model to showcase the data to end users. Engineers quickly perform searches and pull the information they need. They also jump from one part of a system or subsystem to another and see the connections between the subsystems. Similarly, project managers use the system to look at information pertaining to various subsystems handled by disparate team members and pull it all together to understand the entire system.

“When you start looking at what kind of documents you have and how you’re able to transform those into actionable

knowledge for your end users, you improve your decision making,” said Meza. “Of course, you also leverage lessons from the past, because we tend to make the same mistakes over and over.”

The Lessons Learned database has already generated significant value.

“This has saved us at least a year and over $2M in research and development towards our Mission to Mars planning,” said Meza.

Going forward, Meza and his team plan to provide users with the ability to input lessons directly into the database. They’ll also run a text analysis to find text reuse or similarity that allows them to identify documents that are similar topically, but different enough that they might not be caught.

“We’re constantly looking at how to redo our Lessons Learned Database,” Meza said. “One of our problems is that we don’t read the database when we’re having issues. But part of knowledge management is the ability to take that know-what into know-how for the end user, and transmit that knowledge to the next generation.”

Follow the money: Governments around the world have collected more than $700 million in fines and back taxes using connected data from the Panama Papers, stored in Neo4j.

Thinking about your own connected data possibilities:

Analysis and Decision making	Can I use data relationships (either defined or hidden) to improve my analysis and decision making?
Leverage	Am I missing opportunities to leverage data relationships to support my mission?
Key mission needs	Will connecting data across systems or silos allow me to answer key mission needs?
Efficiency	Can I be more efficient with a flexible data model that can answer questions much faster?

Conclusion

Now that you've seen the innovative ways government agencies are using graph databases to fulfill their missions, it's important to remember that these case studies are not all-encompassing.

Graph databases are as versatile as the government agencies that use them.

Your use case may differ from those illustrated here, but the impact will be just as empowering. At Neo4j, we help our customers realize the power of graph technology to solve the most challenging and obscure problems, and we'd love to help you consider ways a graph database helps you meet your mission objectives.

Questions about Neo4j?

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