Eckerson Group | eBook

Getting Started with Knowledge Graphs

An Introduction to Knowledge Graphs and How They Can Help Your Business

JOE HILLEARY

MAY 2022

RESEARCH SPONSORED BY DATA.WORLD

data.world

About the Author

Joe Hilleary is a writer, researcher, and data enthusiast. He believes that we are living through a pivotal moment in the evolution of data technology and is dedicated to helping organizations find the best ways to leverage their information. He has authored numerous articles and reports covering analytics, governance, and data sharing. He has also spoken widely on topics ranging from data catalogs to business intelligence and DataOps. A firm advocate of innovation, he loves tracking the latest developments in the world of data.

About Eckerson Group

Eckerson Group is a global research and consulting firm that helps organizations get more value from data. Our experts think critically, write clearly, and present persuasively about data analytics. They specialize in data strategy, data architecture, self-service analytics, master data management, data governance, and data science. Organizations rely on us to demystify data and analytics and develop business-driven strategies that harness the power of data. Learn what Eckerson Group can do for you!

About This Report

Research for this report comes primarily from numerous briefings with software vendors. This report is sponsored by data.world, who has exclusive permission to syndicate its content.

Table of Contents

Introduction

Lately, everyone seems to be buzzing about knowledge graphs. Once relegated to academics, tech giants, and niche enthusiasts, graph technology is going mainstream. Just this past year, a graph vendor completed the single largest funding round ever for a database of any kind, bringing in more than $325 million dollars. But even as venture capitalists bet big on knowledge graphs, most companies still have no idea what they are.

In contrast to the many dissertations and technical guides on knowledge graphs targeted to scholars and engineers, this eBook aims to serve businesspeople. It provides a thorough introduction to the fundamentals of knowledge graphs with an emphasis on providing data and analytics leaders with sufficient context to make informed decisions for their organizations. Each chapter answers a different question, collectively covering the "what," "why," "how," and "where next" of this technology.

- **Chapter One: Origins and Definitions** provides historical background and an overview of the basic concepts and terminology needed to understand knowledge graphs.
- **Chapter Two: Use Cases** explores many of the business applications for knowledge graphs.
- **Chapter Three: Building a Knowledge Graph** introduces a step-by-step guide to building a knowledge graph from an organizational perspective.
- **Chapter Four: Into the Future** projects future trends as the knowledge graph industry continues to mature over the next few years.

We hope to demonstrate both the power and limitations of this technology and give you the tools to sort hype from reality. Although we can't cover everything, as a reader you should walk away with a better sense of knowledge graphs and their potential.

Chapter 1: Origins and Definitions

This chapter provides historical background and an overview of the basic concepts and terminology needed to understand knowledge graphs.

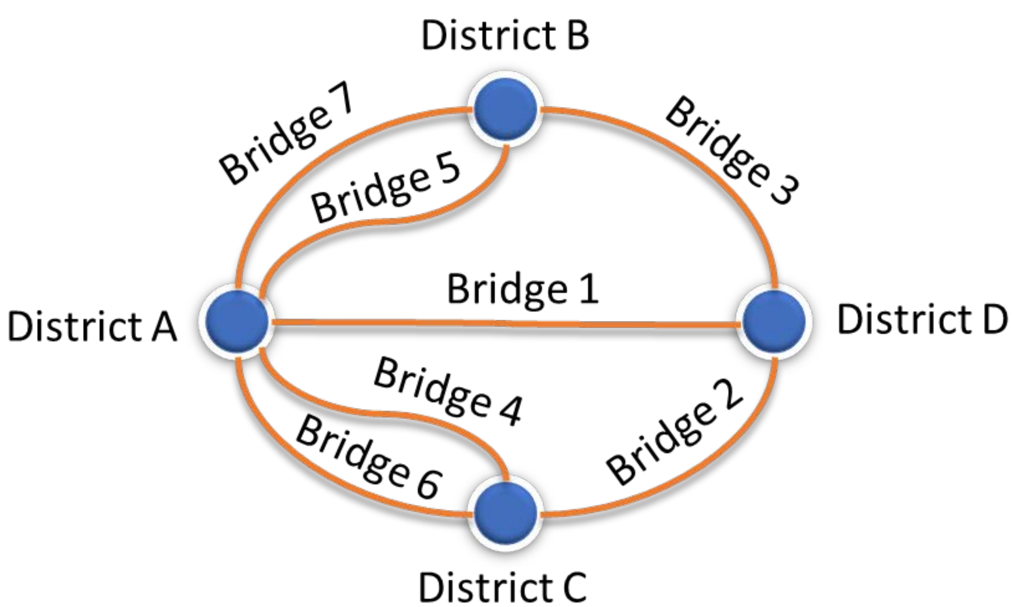
When I say "graph," what do you think of? Something with x and y axes? Maybe a chart? Unless you took a fair amount of college mathematics, the odds are high that the terms vertices and edges don't pop into your brain first. But as graph databases become more widespread and knowledge graphs begin to power business critical use cases like fraud detection, customer 360, market intelligence, content management, drug discovery, and data integration, we'd do well to get a better handle on these concepts and how they work.

In this section I will introduce the basic terminology for understanding the world of graphs and define knowledge graphs—a specific application of this technology. In later chapters, I will highlight key use cases for knowledge graphs, go over best practices for implementing them, and offer predictions about the long-term role of these tools.

The Origins of Knowledge Graphs

The Mathematical Definition of a Graph. Our story starts in Königsberg, Prussia in 1735. Peerless mathematician Leonhard Euler solves a puzzle involving seven bridges that cross the Pregel River and connect the four landmasses, or districts, of the city. As part of his proof that there was no way to walk through the city while crossing every bridge exactly one time, Euler treats each landmass as a vertex, or node, and each bridge as an edge, or link. His approach would become the basis of graph theory (See figure 1.)

Figure 1. Euler's Map of Königsberg.



Since then, at least in the realm of mathematics, a graph has been defined as a set of objects linked by relationships between one or more pairs. And, as it turns out, you can describe a whole lot more than bridges with a graph.

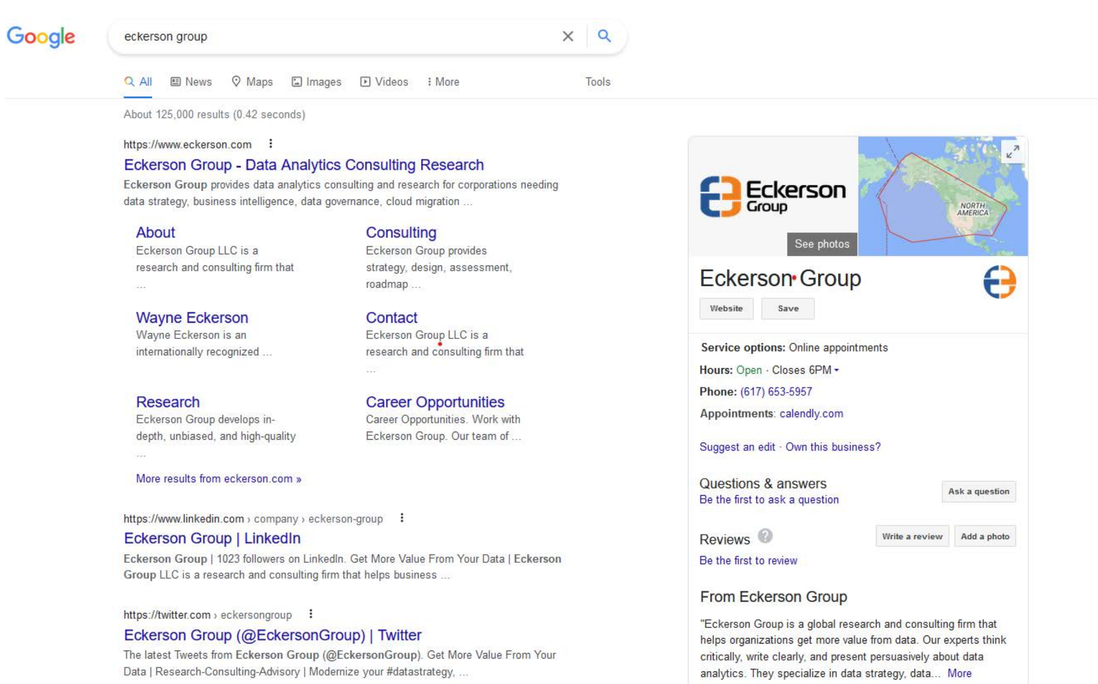
The Modern Evolution of Knowledge Graphs.

Let's fast forward to the late 1990s/early 2000s. The internet has arrived and Tim Berners-Lee's World Wide Web links millions (soon to be billions) of webpages in what is no less than a giant graph. At that scale, keeping track of what all the objects represent is nearly impossible, so Berners-Lee makes a proposal in the May 2001 issue of Scientific American for a new approach—the Semantic Web.

The Semantic Web would use standards to structure metadata about webpages and the links between them, making the knowledge stored in these relationships machine-readable. Unfortunately, the Semantic Web developed a reputation for being too academic and didn't fully catch on at the time. Nevertheless, the idea behind the proposal remained sound, becoming W3C standards (RDF, OWL, SPARQL).

In fact, search engines and social media companies in the mid to late-2000s quickly realized they were dealing with extremely large graphs and in order to make sense of them, they needed a systematic way to map general knowledge about the nodes and their relationships. Google can ultimately take credit for rebranding the Semantic Web and popularizing the term "knowledge graph." In 2012, it used the term to introduce the Google Knowledge Graph, which powers its search engine and provides the information found in the knowledge panels that appear after most Google searches. (See figure 2.)

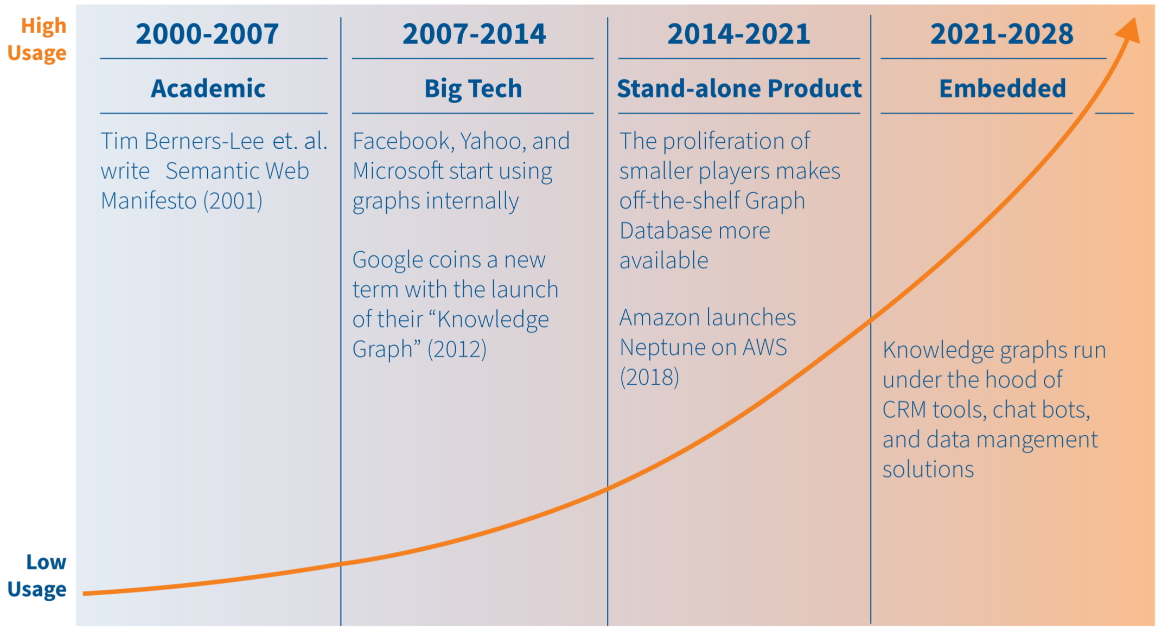
Figure 2. Eckerson Group Google Knowledge Panel.



Google wasn't the only company to start using knowledge graphs, however. In addition to its competitors in search, Microsoft and Yahoo, companies including Facebook, LinkedIn, Amazon, and Airbnb all started to rely on knowledge graphs internally to manage their core offerings. But while large tech companies had the expertise to build their own in-house graph solutions, other industries needed more out-of-the-box platforms to build their knowledge graphs.

In recent years, the number of graph technology vendors has proliferated, lowering the bar for smaller and less technologically savvy organizations in various industries to take advantage of knowledge graphs. New offerings from older vendors, fresh start-ups, and large cloud vendors, such as AWS, have reduced the time to value, increased ease of use, and improved query performance at scale. As we move into the future, I anticipate knowledge graph adoption will continue to grow, becoming a standard solution that runs under the hood of many of the point tools that already exist. In a later chapter, I will dig into this final phase in more detail. The chart in figure 3 below summarizes the trajectory of the adoption.

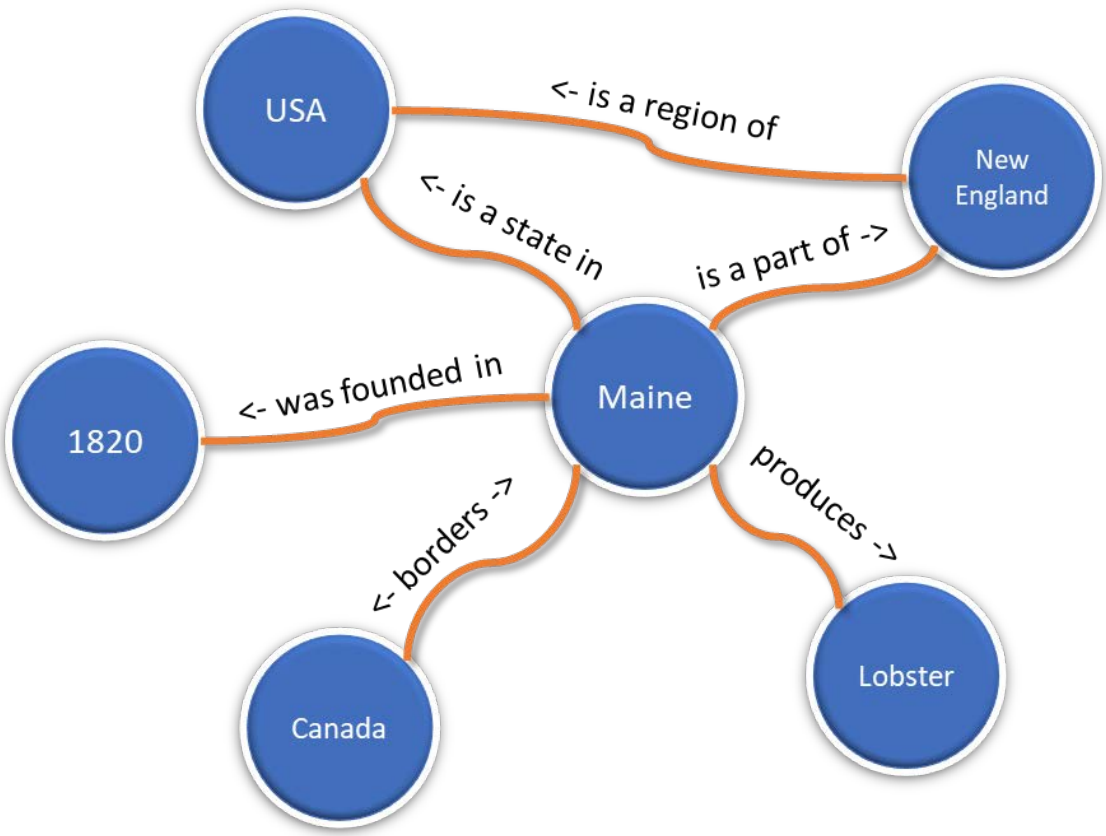
Figure 3. The Adoption of Knowledge Graphs.



What is a Knowledge Graph?

Now that we've covered the origins of knowledge graphs, it's time to dig into their basic attributes. Essentially, a knowledge graph maps the relationships between objects (data) and provides information that helps humans and machines understand what the data actually means. A knowledge graph is data plus metadata (or semantic information) linked in a graph. This concept is easier to understand visually, so I have crafted a small knowledge graph in figure 4:

Figure 4. Knowledge Graph About the State of Maine.



From this graph, I can not only learn about Maine (where I live), but also uncover new information that was not explicitly entered. For example, nowhere does it say that Canada borders New England. However, because I know from the graph that Maine borders Canada and is a part of New England, I can logically infer that Canada must border New England. In practice, most knowledge graphs are exponentially larger than the one I've depicted here, so it makes sense to use software to keep track of them. This also allows machine learning models to ingest the information they contain, allowing them to power a wide range of applications.

Most knowledge graphs reside in a graph database. Although theoretically, one could build a knowledge graph on any type of data store, given the fact that it's a graph, it's best to use a store specifically designed to handle graph data. Graph databases differ from traditional relational databases because they treat relationships between objects as first-class citizens. There are two main types of graph databases:

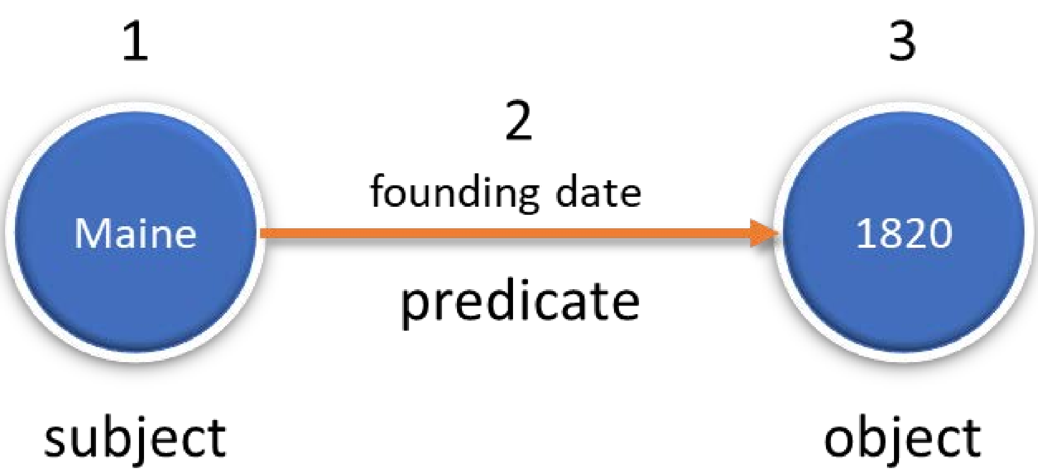
> RDF Triple Stores

> Labelled Property Graphs

At a high-level, RDF Triple Stores lend themselves to metadata management, standards, data interoperability, while Label Property Graphs focus on analytics and fast start up times for developers. This a gross generalization, and, as the approaches evolve, their capabilities are quickly converging.

RDF Triple Stores. Triple stores are the direct descendants of Berners-Lee's Semantic Web initiative. Using the **Resource Description Framework** (RDF) specification created for handling metadata on the web, triple stores break graphs down into sets of three, consisting of two nodes and an edge. (See figure 5.)

Figure 5. A "Triple."



What makes RDF triples special is that each piece of the triple—subject, predicate, and object—are given a Uniform Resource Identifier (URI) that gives it a unique, machine-readable identity. Standard ontologies and taxonomies give these triples structure and make them interoperable with other RDF-based knowledge graphs. This standardization makes it easier to integrate data and to tap into open data sources and third-party data. Triple stores also have a standard query language SPARQL, so engineers only have to learn one language regardless of which database they use.

Labelled Property Graphs. Labelled Property Graphs initially came into being in Sweden as part of an effort to build an enterprise content management system. Unlike RDF, with its focus on standards and interoperability, labelled property graphs, or just property graphs, emphasized storage and query speeds. Property graphs differ from RDF not only in their lack of standardization and schema language but also because they allow both nodes and relationships to have, well, properties. Instead of having a triple for every piece of information, property graphs permit elements of the graph to have internal structure. For instance, in my example, I might represent founding dates as an attribute or property, rather than a linked node. (See figure 6.)

Figure 6. Properties in a Graph.

This approach allows for greater flexibility but requires more building from scratch. Organizations are largely responsible for creating their own ontologies and taxonomies to manage the graph and cannot rely on open-source resources to the same degree. Query languages for property graphs tend to be more approachable for developers than SPARQL, but they lack standardization and essentially every database has its own language.

Functional convergence. Over time, both approaches have appropriated features of the other. RDF-Star, a new standard for RDF graphs permits the addition of metadata to edges, previously something that could only be done in a property graph. At the same time GQL, an initiative to standardize the query language for property graphs, will create greater interoperability between property graph databases giving them some of the advantages that RDF currently enjoys. Ultimately, we are rapidly reaching a point where the distinction between RDF and property graphs will blur as the databases on both sides begin to interface with and convert from one format to the other. In the meantime, either can store a knowledge graph.

In Summary

A knowledge graph is essentially a map of everything an organization knows about a given topic. They can be confined to a particular domain, or, in the case of an enterprise knowledge graph, map everything the company knows about everything. Knowledge graphs combine data with semantic information to provide context, enabling human and machine-intelligible insights. Graph databases, both property and RDF, provide a storage layer for knowledge graphs, providing the means to efficiently query massive graphs.

The knowledge graph industry has grown enormously since the days of the Semantic Web, and knowledge graphs are an increasingly attainable solution for organizations outside the world of big tech. In the next chapter, I will dive more into the various applications of knowledge graphs and the business use cases they enable.

Chapter 2: Use Cases

This chapter explores many of the business applications for knowledge graphs.

I'm an amateur woodworker and, living in a heavily forested area, every so often I'll see a gorgeous piece of live-edge wood for sale on the side of the road. Each time it takes every ounce of my willpower to keep myself from pulling over and purchasing it. However beautiful the wood may be, I know unless I have a particular project in mind, it will just hang around my house taking up space. Technology is the same way. There's no point in having it just for the sake of having it, you have to put it to work.

In the first chapter, I introduce the concept of a knowledge graph. But as nifty as they are, to be useful, a knowledge graph needs a domain—it has to be a graph about something. It could store knowledge about people, organizations, content, machines, even the very data landscape of which the knowledge graph is a part. This chapter will explore some of these key applications to demonstrate the ways you can put a knowledge graph to work.

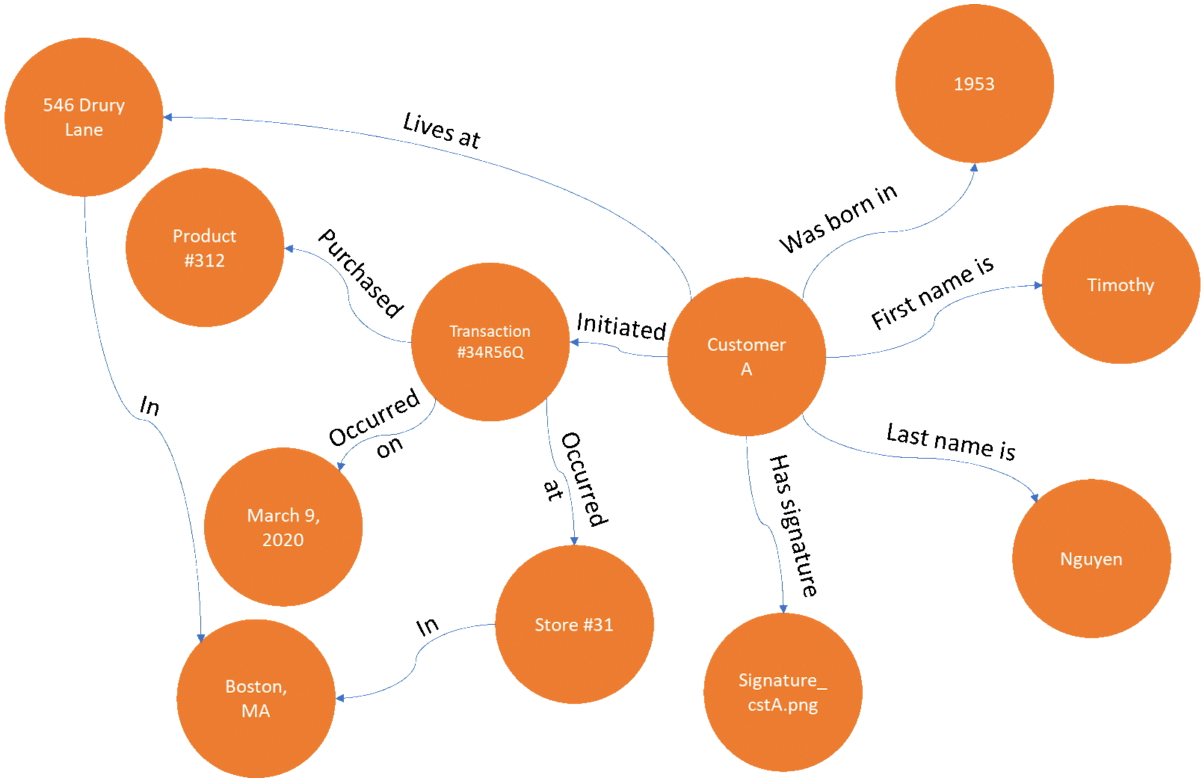
Graphs About People

People are one of the most common subjects for knowledge graphs. After all, people and their actions generate much of the data companies use to run their businesses. A knowledge graph allows organizations to link all of the data about a person together and then connect multiple people together into a network.

Customer Graphs. Businesses cannot exist without customers, which makes understanding them vital to the success of any enterprise. It shouldn't come as a surprise, then, that companies often start their journey into knowledge graphs by trying to map information about their customers.

Consider the challenge of a customer 360 strategy. In order to create a unified view of every customer, the department has to create a unique customer entity and connect it to data drawn from a multitude of business systems. Knowledge graphs provide an intuitive way to link this kind of information. A knowledge graph allows companies to turn each customer into a node in the graph that's connected by relationships to the data about them. Within the graph, not only is the data unified, but it's also connected to its meaning. The relationships themselves express how the person relates to the data. (See figure 7.)

Figure 7. A Sample Customer in a Knowledge Graph.



A knowledge graph like this one can link master data such as name, age, and address to a person just as easily as their transaction history. And what's more, because each node is unique, it can link customers to each other in ways that might not be captured by source systems. In the graph above, perhaps a Customer B also lives at 546 Drury Lane in Boston. Now we know Customer A and B live together even though that specific fact was never recorded.

The deeper understanding provided by a customer knowledge graph also facilitates better customer recommendations. Because knowledge graphs provide their data in a machine-readable format, data scientists can train machine learning (ML) models on them. These models pick up on the relationships between customers to judge their similarity and use that to suggest products with greater accuracy.

Customer knowledge graphs also have application in the realm of fraud detection. By analyzing the relationships between groups of customers, their IP address, transactions, known credit cards, etc., fraud specialists and security algorithms can readily identify suspicious behavior that might be invisible in traditional systems. Perhaps a customer's most recent transaction occurred in a new store. In and of itself this may not be unusual, but if the signature looks different than previous ones—the location of the store is across the globe from the customer's address, and the products purchased are typically bought by customers of an entirely different demographic—something might be up. Because the knowledge graph both stores and connects this information, conclusions become easier to draw.

Employee Graphs. Companies also create person-focused graphs of their employees. As with customer graphs, employee graphs are a way to unify the data about the people critical to an enterprise's success. These knowledge graphs can serve as an org chart on steroids. Not only do they provide information about the company hierarchy and unify human resource data, but they can also help link employees through the projects on which they work, creating a social and productivity map of the entire company. This map can then be used to evaluate performance, improve company culture, and monitor interactions.

Graphs About Organizations

Knowledge graphs can take more than just people as their subjects. Companies frequently create them about other organizations. A robust knowledge graph can serve as a kind of super customer relationship management (CRM) platform. It might model data extracted from a traditional CRM or it might serve as the initial system of record. In either case, it takes the data to the next level by linking it to other data to create context. An organization becomes a single node in the graph linked to other nodes that might consist of contacts, contracts, projects, meetings, emails, or any number of other relevant entities. Once again, the graph structure allows new information to emerge that wouldn't be captured in a table. A company would be able to track its relationship with a particular contact across multiple organizations and see how sales to that contact change over time, because the contact would remain a unique entity linked to an employer by a relationship rather than an attribute.

A robust knowledge graph can serve as a kind of super customer relationship management (CRM) platform.

Company intelligence is another key use case for a knowledge graph of organizations. Financial firms looking for investments, or companies in any vertical trying to track competitors or potential acquisitions can benefit from this type of graph. Like a person-focused graph it facilitates the modelling of social networks, only of companies instead of people. Links in the graph can reflect complex business relationships that help new insights and connections to emerge.

Graphs About Content

Knowledge graphs don't have to store information about external entities, however. They can also facilitate better understanding and use of assets created within a company. Content management is one of the most important use cases for knowledge graphs today. In fact, much of the academic underpinning of knowledge graphs comes from the library science discipline which takes content management as its central calling.

Content management is one of the most important use cases for knowledge graphs.

In a content graph, raw content is stored as a node and metadata about the content is linked to it. This metadata can help create associations between pieces of content improving content search functions, much as customer graphs can improve recommendation systems. The knowledge graph can also store information about how to present the content on different platforms or in different geographic regions. For example, the text of an article might be linked to code that tells the website how to display the text on a phone versus a desktop. It might also link to regional versions of the content, allowing these variations to still be identified with the original piece.

Graphs About Machines

The assets stored in a knowledge graph don't have to be entirely digital either. Particularly in the manufacturing industry, companies create knowledge graphs of their production lines. In this case, the knowledge graph contains nodes representing each part of the machines in use. These nodes connect to the internet of things (IoT) data generated by the sensors on the machines and linked together to form a virtual copy, or digital twin, of the factory floor. Using this virtual model, companies can test new processes, or diagnose problems by running analytics on the graph.

Graphs About Data

Finally, organizations can apply knowledge graphs to the enterprise data landscape itself. On a basic level, this might take the form of a knowledge graph-powered data catalog. For example, data.world is powered by an RDF knowledge graph which enables users to extend, query and visualize the data in the catalog. A data catalog as a graph makes sense because, at its core, a catalog simply links together metadata describing a particular data asset, such as descriptions, locations, statistical information, reviews, access rights, and lineage. This use case is very similar to the content management use case, but the content in question is actually data.

The holy grail for data-focused knowledge graphs, however, is the enterprise data fabric. The data fabric goes one step beyond the data catalog to link information about how to access each data asset into the graph itself. In the case of a data fabric, the knowledge graph is a semantic layer and acts as a kind of data virtualization engine to unify the entire data landscape. Consider a node in the graph that represents a particular table. In addition to information describing the table, the knowledge graph also contains the code to automatically query the source system, retrieve, and format the table. This allows for federated queries that reach through the knowledge graph to access every other data source or business system. A data fabric does more than virtualize the data, though. Because it relies on a knowledge graph, it uses relationships to record how each system understands the data, while allowing the data itself to exist outside of an application silo.

The holy grail for data-focused knowledge graphs, however, is the enterprise data fabric.

Is my use case a good fit for a knowledge graph?

While the examples covered in this article are by no means exhaustive, hopefully, they give you a better sense of what's possible with a knowledge graph. To evaluate other potential applications, ask yourself the following questions:

> Do the entities I want to know about naturally form networks, i.e., is the data "graph-y"?

> Does this use case require linking metadata or additional context to the data itself?

> Are there insights in my data that will only emerge if it's interlinked?

> Are the relationships between the data just as important as the data itself?

If you answered yes to any of these questions, it might be worth looking into implementing a knowledge graph. Fortunately for you, in the next chapter, we will look into the logistics of building a knowledge graph at an organizational level.

Chapter 3: Building a Knowledge Graph

This chapter introduces a step-by-step guide to building a knowledge graph from an organizational perspective.

Consider this section the businessperson's how-to guide to knowledge graphs. I will explain in 10 steps the people, processes, and technology needed to construct an effective knowledge graph. Although these steps form a loose sequence, ultimately building a knowledge graph is an iterative process. You may need to cycle between a few steps multiple times before proceeding to the next. These steps follow a pay-as-you-go methodology to design and build a knowledge graph.

Step 1: Confirm That You Need a Knowledge Graph

Every knowledge graph should start with a business use case. Let the need drive the solution, not the other way around. Knowledge graphs have a lot of applications, but not every use case requires a knowledge graph. At the end of my last article, I proposed four questions that can be used to determine if a particular use case would benefit from a knowledge graph:

1. Do the entities I want to know about naturally form networks, i.e., is the data graph-y?
2. Does this use case require linking metadata or additional context to the data itself?
3. Are there insights in my data that will only emerge if it's interlinked?
4. Are the relationships between the data just as important as the data itself?

If the answer to any of the questions is yes, proceed to the next step.

Step 2: Decide on a Minimum Viable Product

Many knowledge graph initiatives fail because the initial vision is too ambitious. Yes, the ultimate goal might be to link all of the data within your organization within an enterprise knowledge graph, but that's not where to start. Work with end-users to consider what the graph needs to do to solve the business problem you selected and decide how small the graph could be while still accomplishing the objective. Strictly limit the scope of your graph to that minimum viable level and prove its value before trying to expand.

Step 3: Determine What Information Needs to be Modeled

Great, now through conversation with business stakeholders you know what the graph needs to do. The next step is to figure out what information it must contain to accomplish that goal. Consulting with domain experts is critical at this stage. Only they know the kinds of data most relevant to the use case.

Ask what data they look for most often, what features drive their decision making, and what they'll need in the graph to do their jobs. Ultimately, the knowledge graph will need to mirror how they already visualize the information in their domain.

Step 4: Find Out What Work Has Already Been Done

Once you know what data you'll need, see if you can benefit from someone else's work. The knowledge graph community was founded on open-source principles, so public resources often exist to give you a jump start. These might include domain specific graph-compatible data sets, industry standards, or pre-defined ontologies (more on that in step 7). Don't reinvent the wheel if you don't have to. Researching available resources before you get started will save enormous amounts of time.

Step 5: Select People to Build the Graph

At a minimum, building a knowledge graph means filling two key roles. If this is your first knowledge graph, the odds are high no one on your team has specific knowledge graph expertise. Although knowledge graphs have become increasingly mainstream, knowledge of the relevant languages and technologies remains somewhat rare.

The first persona needed for a successful knowledge graph initiative is a designer. They need to be an expert in knowledge management and ontology development. This person will decide how to organize the data in the graph. They will create the theoretical framework over which your data will be laid. People with this expertise often come from the world of library science, which has been thinking about taxonomies, ontologies, and other knowledge management concepts for longer than any other discipline.

The second role is an engineer. This person will be responsible for realizing the designer's vision. They need to be fluent in SPARQL (for RDF graphs) or one of the many property graph languages such as Gremlin, GSQL, or Cypher. They should understand the technical differences between different graph databases and have familiarity with their user interfaces.

These personas may have the titles of Knowledge Scientist or Knowledge Engineers.

Step 6: Choose the Appropriate Tools

With your team assembled, it's time to choose your tech. At minimum you will need:

> **A data store**, ideally a graph database that's already optimized for graph data structures

> **An ontology management tool**, these exist to help designers create and manipulate ontologies

Often dedicated knowledge graph platforms will bundle these capabilities, combining a storage solution with the tools needed to actually construct the graph. You can also take a best-of-breed approach and purchase stand-alone solutions as long as they integrate with one another. Select tools that complement the experience and expertise of your team. If they're already comfortable with the tools you select, you'll realize faster time to value.

Step 7: Design an Ontology

Your knowledge graph designer should work closely with domain experts and business users on this step. An ontology is the basic framework of the graph. It defines how concepts relate to one another. Unless this piece is spot on, the graph will be a pain to use. It should reflect the understanding of end-users so that it is intuitive for those who aren't ontology experts. Once the ontology exists at a theoretical level, the engineer will encode it into the graph.

Step 8: Populate the Graph

When the ontology is ready, the engineer should add the actual data to the graph. Although simple from a conceptual standpoint, this step can be very time and labor-intensive. This is the piece most other how-to guides focus on.

Step 9: Integrate the Graph with Other Systems and Processes

Congratulations! You now have a graph. But a graph in isolation is useless. Remember your end-user is a businessperson, not the knowledge graph engineer. Once the graph exists, the engineer needs to integrate it with business systems, so users can actually take advantage of it. Depending on the use case, the graph should connect to business intelligence tools, customer relationship managers, enterprise resource planning platforms, and/or content management solutions.

Step 10: Check that the Business Need is Met

Does the graph accomplish what it needs to?

1. **If not, determine why not and repeat steps.**

Diagnose what went wrong and try again. This time it won't be from scratch, though. Try to figure out on which step things went wrong, then go back and iterate. The work you've already done should give you a significant boost.

2. **If so, celebrate, then look for other business use cases that could benefit from the graph or extensions of it and repeat steps from the beginning.**

You did it! You built a successful knowledge graph! Since you've proven the value of the approach, it's time to see what other business problems you can solve. Some may only require small tweaks to your existing graph; others could warrant significant expansions. Just make sure you don't get ahead of yourself, and keep it grounded in what the business needs.

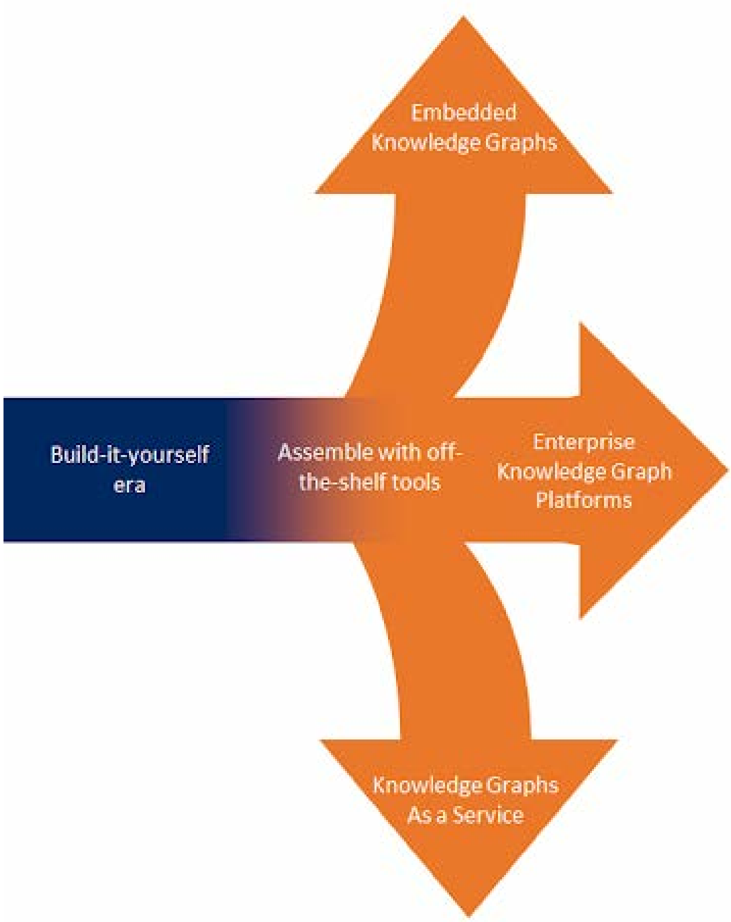
Chapter 4: Into the Future

This chapter projects future trends as the knowledge graph industry continues to mature over the next few years.

We currently stand on the threshold of a new era for knowledge graphs. Once confined to academia and tech giants, the advent of off-the-shelf graph databases in the last decade led to the adoption of knowledge graphs across a wide range of industries. Even with graph databases, however, it still fell to organizations to understand how to build a knowledge graph. If a company possessed a knowledge graph, it was likely that a graph wonk on the tech team had advocated for it and made it work.

Today, vendors continue to take steps to make the benefits of knowledge graphs more accessible. These approaches generally fall into three categories: the enterprise knowledge graph platform, the knowledge graph as a service, and the embedded knowledge graph. (See figure 8.)

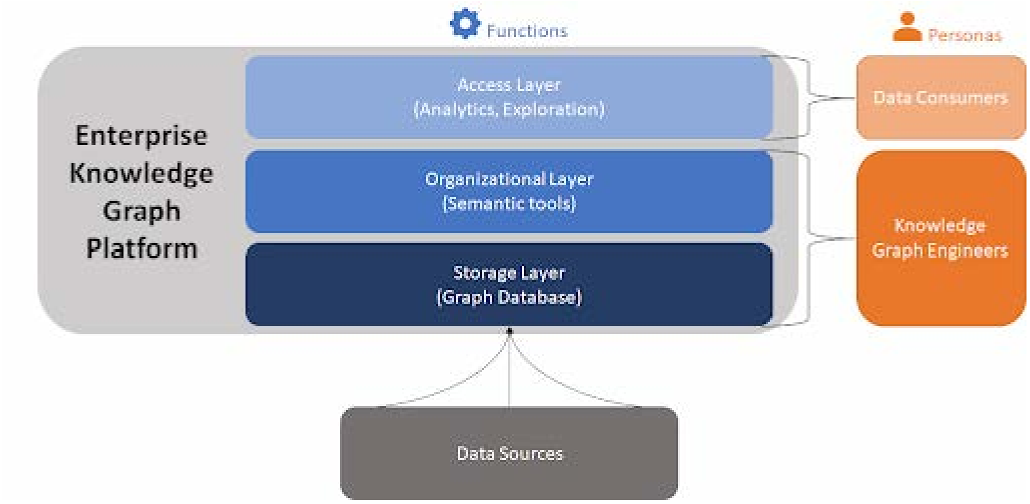
Figure 8. Trajectory of the Knowledge Graph Market.



The Enterprise Knowledge Graph Platform

The enterprise knowledge graph platform is the most direct successor of the previous generations of knowledge graph tools. This category includes offerings from companies such as Stardog, Ontotext, and Neo4J. Though each takes a different technical approach, they all combine a graph database storage layer with features for building, exploring, and analyzing the graph. (See figure 9.)

Figure 9. Enterprise Knowledge Graph Platform.



Putting all the features needed to both build and consume a knowledge graph in a single platform reduces the burden on organizations to manage separate tools for each essential function. In addition, these platforms offer low or no-code consumption environments, so non-technical folks can benefit directly from the knowledge graph. I anticipate that mid-sized enterprises will turn to the platform-based approach to build knowledge graphs since their ease of use makes them a smoother entry point for organizations that have limited engineering resources.

The Knowledge Graph as a Service

Other companies remove the burden of development altogether, simply providing information in the form of a prebuilt, searchable knowledge graph. These knowledge graphs are themselves the product offering. Diffbot is a prominent example of this approach for web data. It provides a massive graph of more than a trillion facts about organizations, products, and news articles sourced from across the web. Rather than constructing their own knowledge graph of these entities, organizations can simply buy access to Diffbot's.

Data platform company Tresata operates a freely available knowledge graph (BADaaS) based on leaks of financial information provided by the International Consortium of Investigative Journalists (ICIJ) and the Organized Crime and Corruption Reporting Project (OCCRP), including the Panama Papers and Paradise Papers. They intend this graph as a public service that enables global citizens to seek out and identify financial crime. Other open knowledge graphs include Wikidata and DBpedia.

In both cases, whether a paid service or free, the labor of constructing the knowledge graph and the data that populates it are provided to the consumer. All a user needs to do is explore or extract the information they desire.

The Embedded Knowledge Graph

The final category is the embedded knowledge graph. In recent years, software vendors have incorporated knowledge graphs into their application engines. Like the knowledge graph as a service, leveraging an embedded knowledge graph requires no expertise in knowledge graph development. In fact, most application users will interact with the knowledge graph without realizing it.

data.world, the data catalog company referenced in an earlier chapter, is a great example of this approach. It provides a consumer-grade search experience for finding and cataloging data assets. This application sits on top of a knowledge graph data.world builds automatically from metadata it gathers from customer data sources, but the average user never even knows it's there. They simply conduct their searches and view and retrieve data through a graphic interface.

Roam Research takes a similar approach to the world of notetaking. Customers use Roam to jot down thoughts, as with any number of other notetaking apps, but Roam captures these notes in a knowledge graph. This underpinning allows users to link their thoughts and record relationships between notes, providing enormous benefit over traditional filing approaches that isolate ideas. But again, users don't need to know the first thing about knowledge graphs to benefit from them.

In addition to these two examples, vendors today use knowledge graphs to power many other types of software applications including supply chain management tools, chatbots, and AI search features. Given the natural synergies we explored in the second chapter, I expect next-gen customer relationship management systems and content management systems will use knowledge graph engines too.

Take-away

As knowledge graphs become mainstream, all three of these approaches will prosper. Although distinct, together they represent a larger trend toward the wider adoption and democratization of knowledge graphs. This growing ubiquity makes it important for business and data leaders to understand the basics of knowledge graphs even if they don't intend to build one.

Next Steps

Hopefully this eBook has served as an effective introduction to knowledge graphs. By now, you should have a grasp of what a knowledge graph is, why you'd use one, how to build your own, and where the industry is likely to evolve. Perhaps you've been persuaded that your organization could benefit from this technology, or perhaps you simply want to deepen your understanding further. Either way, the recommendations below provide additional actions that can help you on your knowledge graph journey.

Recommendations:

- Identify other graph enthusiasts among your colleagues or employees
- Start a working group to identify graph-y problems within your organization
- Take a briefing with a knowledge graph vendor
- Join knowledge graph communities on LinkedIn, Slack, and other social media.
- Attend the Knowledge Graph Conference

About Eckerson Group

Wayne Eckerson, a globally-known author, speaker, and consultant, formed Eckerson Group to help organizations get more value from data and analytics. His goal is to provide organizations with expert guidance during every step of their data and analytics journey.

Eckerson Group helps organizations in three ways:

> **Our thought leaders** publish practical, compelling content that keeps data analytics leaders abreast of the latest trends, techniques, and tools in the field.

> **Our consultants** listen carefully, think deeply, and craft tailored solutions that translate business requirements into compelling strategies and solutions.

> **Our advisors** provide one-on-one coaching and mentoring to data leaders and help software vendors develop go-to-market strategies.

Eckerson Group is a global research and consulting firm that focuses solely on data and analytics. Our experts specialize in data governance, self-service analytics, data architecture, data science, data management, and business intelligence.

Our clients say we are hard-working, insightful, and humble. It all stems from our love of data and our desire to help organizations turn insights into action. We are a family of continuous learners, interpreting the world of data and analytics for you.

Get more value from your data. Put an expert on your side. Learn what Eckerson Group can do for you!

Eckerson Group
RESEARCH CONSULTING ADVISORY
GET • MORE • VALUE • FROM • YOUR • DATA

About the Sponsor

data.world is the enterprise data catalog for the modern data stack. Our cloud-native SaaS (software-as-a-service) platform combines a consumer-grade user experience with a powerful knowledge graph to deliver enhanced data discovery, agile data governance, and actionable insights. data.world is a Certified B Corporation and public benefit corporation and home to the world's largest collaborative open data community with more than 1.3 million members, including 2/3 of the Fortune 500. Our company has 50+ patents and has been named one of Austin's Best Places to Work six years in a row. Follow us on LinkedIn, Twitter, and Facebook, or join us.