White paper

Microsoft Azure

Azure SQL Edge: A data engine optimized for IoT workloads on edge devices

Executive summary

The Internet of Things (IoT) is a promising area of application development that, by detecting important patterns in data collected at the network edge, is already yielding significant benefits such as reduced costs and improved safety for many of its adopters. But until now, the potential of IoT applications has been limited because no currently available data platform has included the diverse combination of features that is needed to support IoT applications in an optimal way.

Microsoft Azure SQL Edge delivers the first such data platform. On the one hand, Azure SQL Edge fully supports disconnected applications on small edge devices through a lightweight data engine that features advanced security capabilities and is powerful enough to ingest, store, and perform analytics on locally collected IoT data. On the other hand, Azure SQL Edge is also built to help protect that data and easily move it, if necessary, into the rich ecosystem of Azure products and services that offer additional benefits such as visualizations, longer-term storage, and machine-learning (ML) model training.

Built on the same codebase as Microsoft SQL Server and Azure SQL, Azure SQL Edge also provides the same industry-leading security, the same familiar developer experience, and the same tooling that many teams already know and trust.

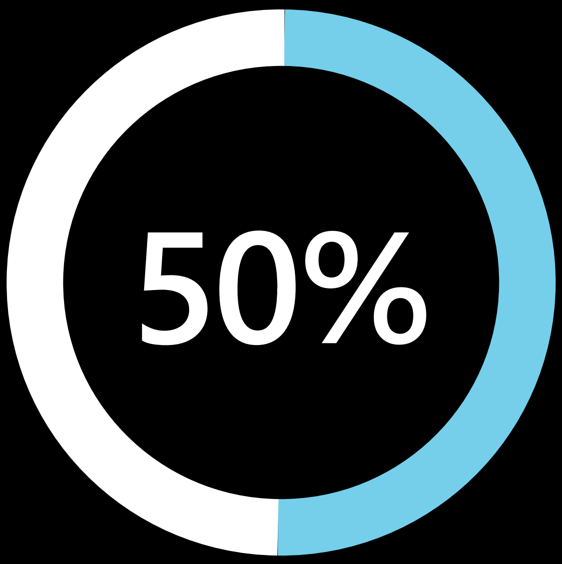
Azure SQL Edge delivers a significant boost to those who are interested in innovating in the IoT space. It provides a data platform that is flexible enough to support the full range of IoT scenarios, powerful enough to support edge compute, and secure enough to help meet the privacy needs of IoT applications—all while remaining fully compatible with a familiar ecosystem of products, tools, and services.

The transformative potential of IoT

By collecting and mining data generated by devices at the network periphery, or *edge*, IoT holds the potential to increase businesses' profits and improve customers' lives. IoT devices can be as diverse as an industrial robot on a factory floor, a humidity sensor on a remote farm, or a camera in a retail shop. If properly captured and analyzed, data generated on these devices can enable IoT applications to bring benefits such as enhanced safety to workers, higher-quality goods for consumers, and cost savings to businesses. In fact, these benefits are potentially so significant that those who don't adopt IoT risk losing their competitive edge.

IoT is already being adopted in numerous industries such as energy, manufacturing, agriculture, consumer electronics, and wearables. And as a prospective area for future application development, the possibilities of IoT are nearly endless. For example, if unusual patterns in pressure-sensor readings on an oil pipeline tend to precede a valve failure, IoT applications could be developed that alert maintenance crews to initiate repairs whenever those same patterns are detected, which can help prevent oil spills and ecological damage. Other applications that detect patterns found in data gathered from soil and air sensors could help spare farmers the expense of using fertilizers, irrigation, and pesticides unless they're truly necessary. And applications that detect patterns gleaned from human body measurements might help prevent health-related calamities such as heart attacks before they occur, enabling patients to get early treatment that could save their lives.

"By 2022, more than 50% of enterprise-generated data will be created and processed outside the data center or cloud."[1]



Challenges in developing IoT solutions

For all the benefits that IoT promises to deliver, however, IoT development teams—and IoT applications in general—have been hampered up to now by limitations in available data platforms. IoT applications, taken as a whole, operate in an unusually wide range of environments, and their corresponding requirements for data processing are equally wide-ranging. But current IoT data platforms do not yet include the diverse combination of features needed to support IoT applications in an optimal and comprehensive way.

Need to support both connected and disconnected scenarios

For example, on the one hand, the first real-world IoT implementations have shown that relying on upstream resources can hinder the success of IoT applications; on the other hand, organizations also report that data collected at the edge tends to get siloed and go unused. As a result, many IoT applications require both edge compute and interoperability with a well-established ecosystem of upstream data services. IoT data platforms until now have not supported this necessary hybrid capability and flexibility, however.

IoT applications need a powerful data engine

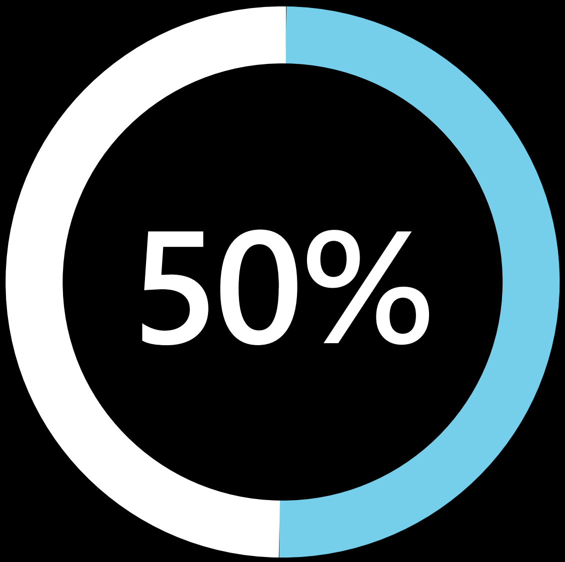
Edge compute is especially needed in IoT applications that require near-real-time analytics (such as in a vehicle, where responses need to be immediate) or when the edge device is totally disconnected from the grid (such as on a farm, where a network connection is often impractical). Other scenarios might require data to be stored and processed at the edge to meet regulatory requirements. But even outside of these cases, sending *all* data to the cloud is unwise because doing so is costly and bandwidth-intensive.

To support the many use cases that require edge compute, IoT applications need a local data engine that is lightweight enough to fit on edge devices, but that also offers the capability to ingest, aggregate, analyze, and move IoT data.

IoT applications need seamless connectivity to a rich ecosystem

At the same time, IoT applications are constrained if they do not connect seamlessly to a mature ecosystem of data services. This ecosystem should be feature-rich and should enable long-term data storage; the retraining of ML models on these larger accumulated datasets; data sharing with other business units and applications; and reporting, visualization, and business intelligence (BI) services.

"By 2022, at least 50% of high-end industrial IoT gateways will offer an optional 5G module."[1]



Security challenges in IoT applications

Another crucial requirement for an IoT data platform is high security. IoT applications often run in locations such as homes, automobiles, or businesses, where privacy and safety are paramount. Therefore, it is essential that users and development teams alike can trust that the database platform used for IoT applications helps provide a high level of security and offers multiple levels of protection.

Challenges of building solutions on unfamiliar platforms

Many existing IoT platforms rely on solution-specific APIs and databases that developers don't know well, or that require specific language skills that developers don't yet have. Any pre-existing applications that work with the broadly embraced standard of T-SQL, for example, will need to be heavily revised to work against these other databases.

Example: Challenges for IoT development in the automotive industry

As an illustration, consider an automotive IoT solution that an independent software vendor (ISV) could assess for potential development. The application would ingest data from dozens of IoT sensors installed throughout the vehicle: on the car battery, brake pads, cameras, and many other parts. Such an IoT application would be especially beneficial if it were able to provide benefits to the driver in both disconnected and connected scenarios.

In a disconnected scenario, medical monitoring could immediately detect when drivers are fatigued or otherwise incapacitated, and then immediately trigger a self-driving feature to slowly guide the car to safety. Predictive maintenance could also help prevent breakdowns before they occur by monitoring the status of car parts and detecting when they need service—even when no internet connection is available.

But when an internet connection is available, the application would ideally enhance these features. Medical monitoring, for example, could be enhanced by sending alerts to appropriate health services when an emergency is detected. And predictive maintenance could automatically notify drivers when needed services—such as an auto shop to replace worn brake pads—are open nearby. Beyond these feature enhancements that are targeted at individual drivers, the ISV can also use upstream connectivity to aggregate data in the cloud and perform analytics on those larger datasets as a way to improve pattern recognition and bring better services to all customers.

However, without a data platform that features both a built-in analytics engine and seamless connectivity to a larger ecosystem, building a solution with all of the desired features would be highly challenging for the ISV. If the capability to perform analytics in near-real-time on data ingested locally were not built-in, the development team would need to build this functionality independently from scratch, increasing development costs and the risks associated with the product.

Furthermore, if the application's data platform couldn't easily connect to a mature and feature-rich ecosystem of data services upstream, the ISV would be limited in its ability to improve services by training ML models on aggregated data. Finally, if the IoT development platform doesn't use a database and query language that is familiar to the ISV's development team, the associated learning curve will further discourage the ISV from attempting to build such an application.

Need for a database platform optimized for IoT

A data platform specially built to meet the wide-ranging, hybrid requirements of IoT applications would remove roadblocks for teams that want to innovate in the IoT space.

To begin with, teams interested in developing IoT solutions need an edge-database platform with hybrid capabilities. Such a platform will give IoT applications the flexibility to operate fully independently when offline, or connect seamlessly to a feature-rich ecosystem when online. More specifically:

- Operating independently when offline means that the IoT data platform needs to be able to ingest and aggregate the time-series data generated by IoT sensors; detect anomalies, filter events, and apply business rules based on live or aggregated data; and take actions locally by executing ML inference models in combination with analytics.

- Connecting seamlessly to a feature-rich ecosystem means that the database platform can be managed through familiar tools with or without the cloud; that its captured data can be normalized for compatibility with upstream tools and services; and that this data can be easily uploaded to the cloud or datacenter for aggregation, further analytics, ML model retraining, data sharing, and reporting.

IoT developers also prefer a database platform based on a familiar ecosystem so that they can easily make use of their existing knowledge and existing applications when developing for IoT. And this database platform needs to be mature, delivering advanced security and administration features such as strong encryption, reporting, data governance, and high availability.

Until now, no data engine has been designed to meet this broad range of IoT application needs.

Introducing Microsoft Azure SQL Edge

Azure SQL Edge is a small-footprint, edge-optimized, ML-capable data engine built on the same codebase as SQL Server and Azure SQL. Thanks to this shared codebase, application developers who have been building solutions for SQL Server and Azure SQL can re-use their code and skills to build edge-specific solutions on Azure SQL Edge that work with or without network connectivity.

Azure SQL Edge answers the need for an IoT-specific data platform by:

- Providing the flexibility to develop solutions that work with or without network connectivity and help enable secure data movement of the local edge data to on-premises datacenters or to Azure.
- Offering support with standard tooling, programming languages, and a query language (T-SQL) that are already familiar to developers and compatible with existing code.
- Enabling artificial intelligence (AI) and analytics at the edge.
- Including native support for ingesting time-series data.
- Delivering industry-leading security and support for regulatory compliance.

Now in public preview, Azure SQL Edge simplifies IoT infrastructure by offering streaming, storage, and analytics all in one platform. It gives teams the power and flexibility to build solutions that run effectively either in a network-connected site, such as on a factory floor, or in a totally disconnected environment, such as on a remote oil rig.

Azure SQL Edge supports a large subset of the T-SQL surface area that developers are familiar with, along with the built-in ML and security capabilities that are available in Microsoft's core SQL database engine. Beyond these compatibilities, all the familiar BI tools such as Microsoft Power BI and Tableau software, along with developer and management tools such as Azure Data Studio, SQL Server Management Studio (SSMS), and Visual Studio, can also all be used with Azure SQL Edge. However, despite the fact that it is built on the same codebase as that of SQL Server and Azure SQL Database, the use cases for these products are different. Azure SQL Edge is optimized for IoT use cases and workloads. SQL Server and SQL Database, in contrast, are built for business-critical data-management solutions and line-of-business applications.

Flexibility for IoT application development

Even with its powerful and rich feature set, the startup memory footprint for Azure SQL Edge is less than 500 MB. This small footprint gives teams the freedom to design and build applications that run on a nearly unlimited range of IoT devices, including on simple battery-powered or solar-powered devices deployed in remote areas with limited connectivity. Azure SQL Edge is a containerized Linux application that runs on either an ARM64-based or an x64-based processor.

In addition, Azure SQL Edge offers platform consistency with SQL Server and SQL Database, allowing developers to build apps once and then deploy them anywhere, whether at the edge, on-premises, or in the cloud.

As shown in Figure 1, Azure SQL Edge is designed to run on the intermediary devices (with at least 1 GB RAM and one core) located between microcontroller units (MCUs), such as sensors at the network edge, and more powerful services in the datacenter or cloud.

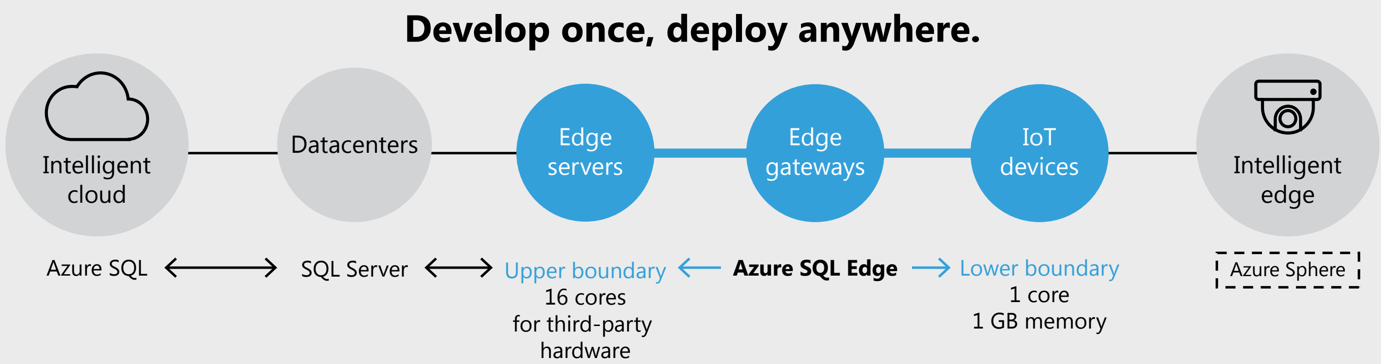


Figure 1. Azure SQL Edge is built to run on a wide range of edge devices located between IoT sensors and the datacenter or cloud

Compatible with a powerful data infrastructure

For those developing or operating IoT applications, no edge data platform that has been previously offered fits seamlessly within a comprehensive data solution spanning the entire spectrum from the network edge to the datacenter and cloud. As a result, the data collected by edge devices tends to get siloed and remain stuck in a proprietary solution, without the option of being fed into complementary cloud services that are part of a business's existing data estate. These complementary services are needed in IoT to perform functions such as running complex analytics, visualizing streamed data, or aggregating data for the long term.

Other edge-platform solutions for IoT are also built from scratch for this single purpose and are not based on a well-established data platform and engine. Consequently, they lack support from a pre-existing ecosystem of developers who could easily contribute powerful service enhancements to the platform because of its compatibility with a query language (such as T-SQL) and set of APIs that are already broadly embraced.

In contrast, Azure SQL Edge prevents data from remaining siloed at the network edge through seamless interoperability with SQL Server and Azure services, delivering consistency in data, tooling, and security across edge, datacenters, and cloud.

Azure SQL Edge offers compatibility with other Microsoft services and applications by providing:

- Synchronization capabilities with Azure services such as SQL Database, SQL Server on an Azure virtual machine (VM), and Azure Cosmos DB.
- A unified management and development experience through graphical user interface (GUI) tools such as Azure Data Studio and SSMS and the command-line interface tools for SQL.
- A security model identical to that of SQL Server and SQL Database, including features such as encryption of data at rest and in motion, role-based or attribute-based access controls, and data masking.
- Integration with Azure IoT Edge and interoperability with other IoT Edge modules to give enhanced functionality for IoT solutions.
- Compatibility with cloud services in Azure that can optionally be used to further explore, visualize, analyze, and store data captured at the edge.

Native data movement

Azure SQL Edge offers built-in capabilities for data movement, allowing native data movement to SQL Server on premises and to services such as SQL Database and Azure Cosmos DB in the cloud. And with these native data-movement capabilities, you have the flexibility to choose data sources (sensors) and targets are best suited to your needs as an application developer or operator.

Time-series and streaming data support

IoT sensors typically transmit time-sequenced measurements, called *time-series data,* as a data stream. There are two aspects of the time-series capabilities offered by Azure SQL Edge:

1. Azure SQL Edge has a built-in streaming engine (built using the same codebase as Azure Stream Analytics) that allows transformation, windowed aggregation, simple anomaly detection, and classification of the incoming stream of data.
2. The time-series storage engine (expected to be available around the general availability [GA] time frame) will allow storage of time-indexed data, which can later be aggregated in the cloud for future analysis.

Figure 2 shows how a streaming data engine with native analytics is one of the core components of Azure SQL Edge, along with a SQL database engine containing built-in ML capabilities. Real-time streamed data ingested by the streaming analytics engine can also be aggregated in the local SQL database. Standard procedures and functions can also be written to process this data as needed.

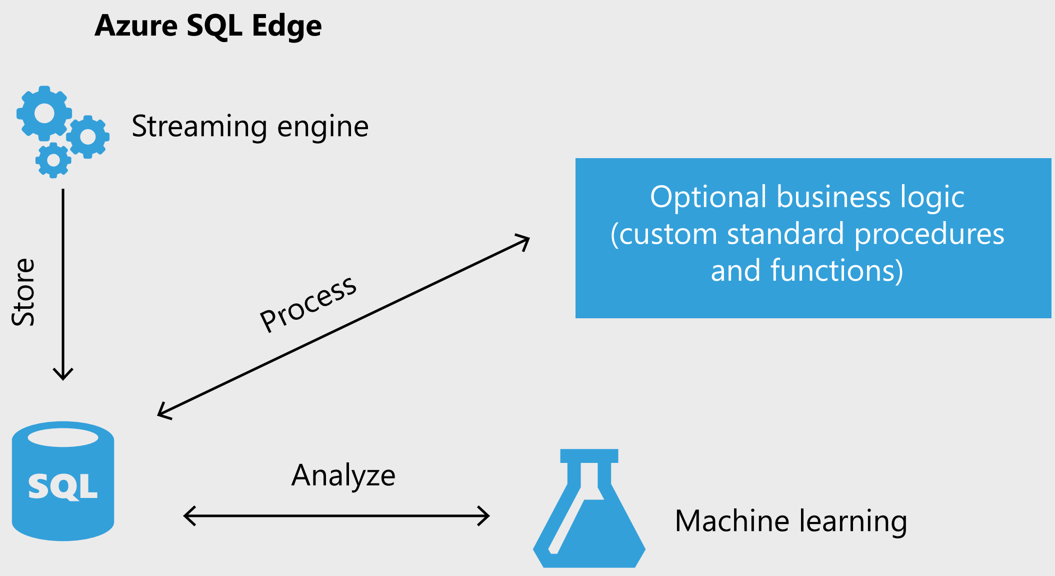


Figure 2. Azure SQL Edge can ingest, store, score, and process streaming analytics data

Once time-series or relational IoT data is stored in the local database on Azure SQL Edge, you can use familiar tools such as Power BI to visualize that data.

Analytics and ML without cloud connectivity

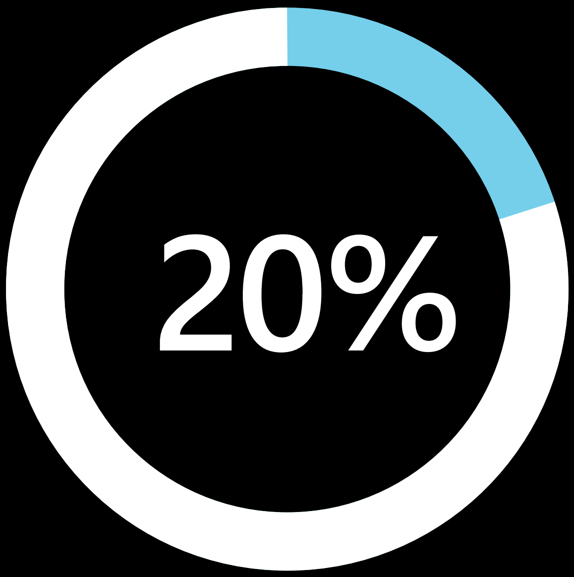
Azure SQL Edge supports analytics and ML even when disconnected from upstream networks, delivering anomaly detection and the capability to score all types of data through in-database ML. ML models can be trained in the cloud via Azure Machine Learning, and you can then deploy those ML models to run on Azure SQL Edge, where you can use them for inferencing and scoring.

Hybrid ML architecture

Azure SQL Edge brings ML capabilities to the edge. Because ML can now be deployed both at the edge and in the cloud, IoT application developers and operators can optimize dataflows for analytics according to their needs.

For example, applications can perform analytics locally when low-latency alerts are required (for automated actions, for example). Another case where you would want to run AI locally is for video analysis—for example, for the purpose of facial recognition. Sending raw video to the cloud would be too costly. For this reason, it's far preferable to run these ML workloads locally on the edge.

"By 2022, 20% of new industrial control systems will include analytics and AI-edge inference capabilities, up from less than 5% in 2019."[1]



On the other hand, you also might want to draw on cloud services for long-term storage or to support other needs, such as applying cloud-based custom business rules for data processing.

Industry-leading security

Security is one of the top concerns for IoT applications. In an increasingly interconnected world, any device that is accessible over the internet is at risk for being hacked, and any data that is stored on or sent either to or from that device is at risk for being intercepted. New regulations in some regions are beginning to enforce security standards on all connected devices, but many existing edge compute solutions still lack important security features.[2] Not only does this lack of security allow data to be read and manipulated, but it also puts regulatory compliance at risk in some scenarios.

SQL Server is the most secure database platform.[3] The same security features of SQL Server Enterprise will be available in Azure SQL Edge to help make sure your data is secure on the device, better ensuring that customers will be kept safe and regulatory compliance will be met.

These built-in security features include:

- **Both role-based access control (RBAC) and attribute-based access control (ABAC):** These two access-control methodologies complement each other by providing a choice of methods through which to control access to resources. RBAC manages access to specific resources based on permissions assigned to the user requesting access, or to the group (or role) to which that user belongs. ABAC manages access control based on rules that refer to the attributes of the user or target data or resource.
- **Data protection with Transparent Data Encryption (TDE) and Always Encrypted capabilities:** TDE enables compliance with many security regulations by encrypting database and log files through an encryption key that itself is encrypted. Always Encrypted provides a separation between those who own the data and those who manage it. It prevents those with administrator privileges to a database from accessing encrypted data that they do not own in that database.
- **Data classification:** This feature helps organizations comply with security regulations by allowing data to be categorized by sensitivity and business impact. Once data is classified, it can be managed in ways that help protect sensitive or important data from theft or loss.

Azure SQL Edge operators and administrators will be able to enforce security policies and manage security updates through a central management portal.

Business-continuity

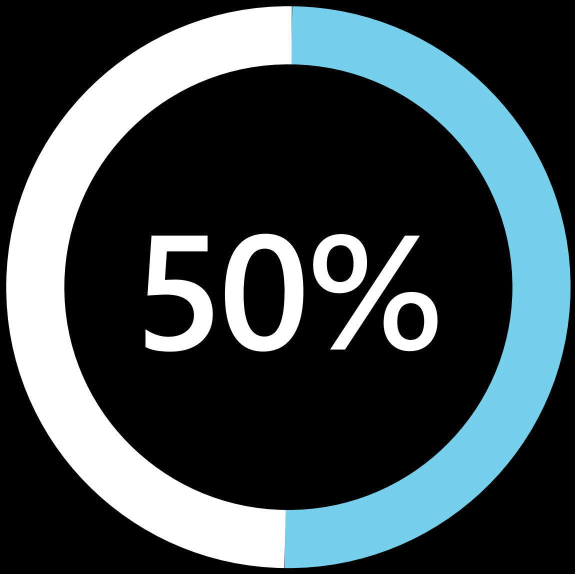
Azure SQL Edge is deployed as a container compatible with Docker. To help ensure availability for Azure SQL Edge deployments, business-continuity options that are compatible with Kubernetes container orchestration are also available.

Simple, scalable edge-device software management through Azure IoT Edge

Azure IoT Edge is a platform that, among other features, provides a way to manage the software installed on edge devices registered in Azure IoT Hub. Through IoT Edge, you can deploy special containers compatible with Docker, called "modules," to edge devices. These modules add functionality to edge devices in the form of applications or microservices.

Azure SQL Edge can be deployed through IoT Edge. This enables customers to manage the deployment of Azure SQL Edge at scale. For example, customers can use IoT Edge to deploy the Azure SQL Edge module to all of their edge devices globally at the same time, or only to a subset of those edge devices as a group based on site location or some other criteria.

"By 2022, at least 50% of greenfield IoT projects will use containers for application life cycle management at the edge."[1]



Easily add functionality to your IoT applications

Because it's an IoT Edge module, Azure SQL Edge delivers benefits to development teams as well. Additional IoT Edge modules can be deployed alongside Azure SQL Edge to the same edge device. These additional modules can easily communicate with Azure SQL Edge and greatly enhance the functionality of IoT applications. For example, the Azure Cognitive Services module can be deployed alongside to provide facial recognition for cameras as part of an IoT application deployed with cash dispensing machines.

Figure 3 shows an edge device with a few modules installed, including Azure SQL Edge, a custom module providing business logic, a Cognitive Services module providing image recognition, and a module providing additional ML capabilities beyond what is already built into the Azure SQL Edge engine.

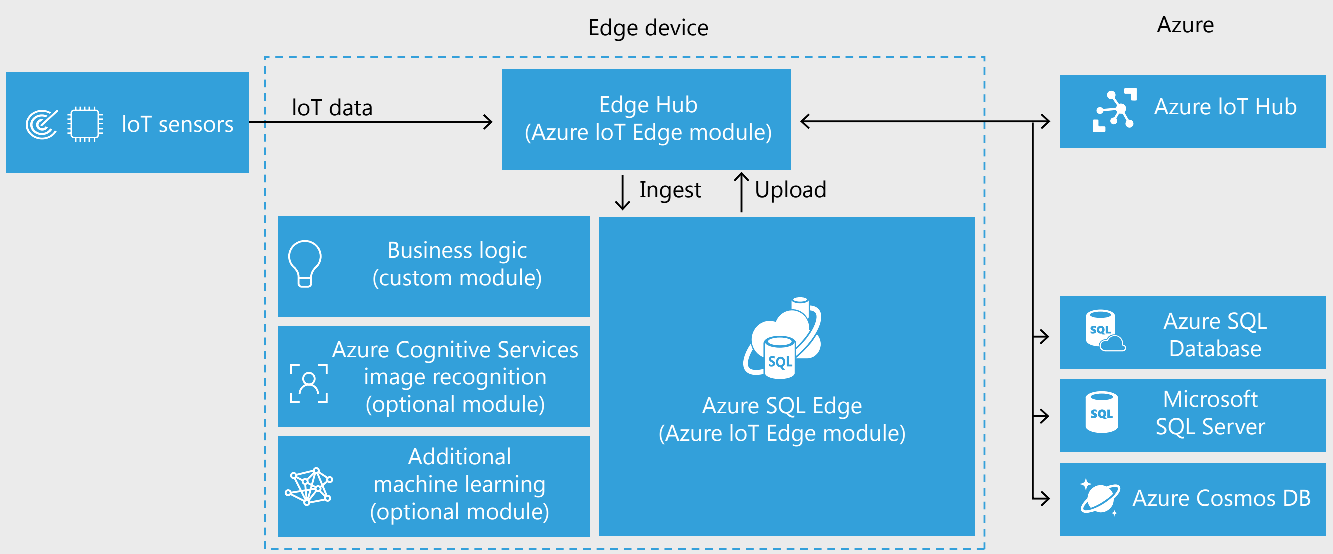


Figure 3. Azure SQL Edge is deployed as an Azure IoT Edge module in this network-connected scenario

Example reference architecture for an end-to-end IoT solution for smart cars

One of the key advantages of Azure SQL Edge for IoT customers is availability of services in Azure that can optionally enhance and complement functionality.

The following example illustrates a real-time data-ingestion and processing pipeline for an IoT application that runs in vehicles. Millions of messages (or events) are generated by the IoT devices and sensors installed throughout the cars. By capturing and analyzing these messages, edge applications built using Azure SQL Edge as the data engine can decipher valuable insights and immediately trigger appropriate actions in a wide range of locations, regardless of whether an internet connection is available. For example, for cars equipped with telematics devices, if applications can capture the IoT messages in real time, the applications could—whether in a connected or disconnected state—monitor the functioning of components, provide safety monitoring for drivers, and plan routes optimized with driver recommendations. Then the applications could rely on Azure services when accessible for long-term storage and further processing.

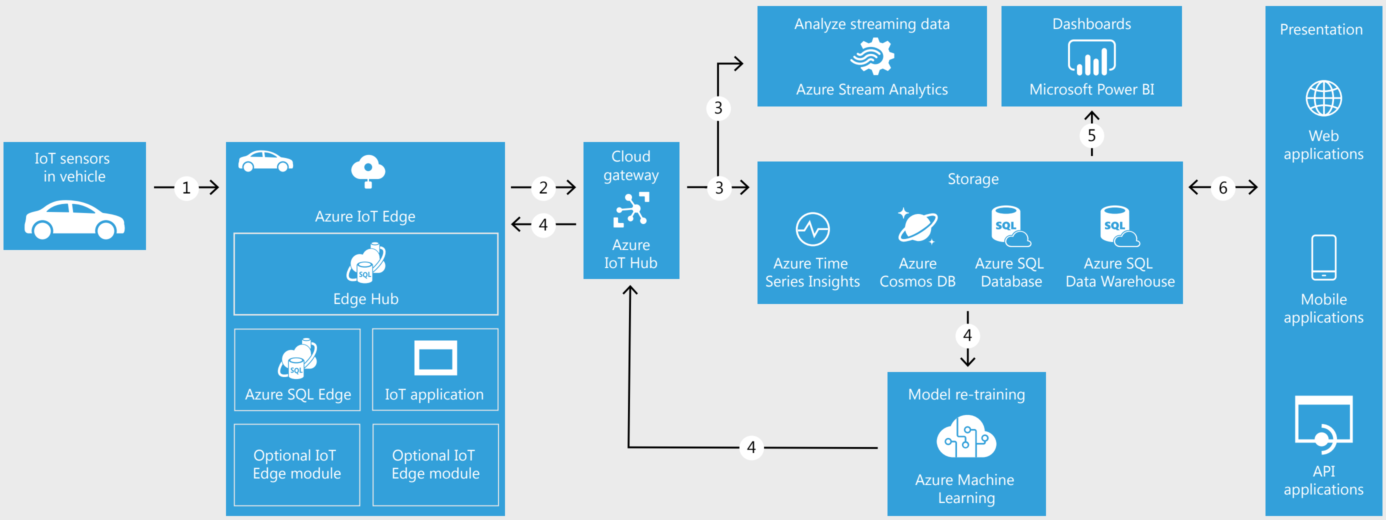


Figure 4. Azure SQL Edge enables a full-featured automotive IoT solution

In this example IoT scenario, data flows through the solution in the following way:

1. Events generated from the IoT data sources are sent to the stream-ingestion layer in IoT Edge. The instance of IoT Edge includes the Azure SQL Edge module, the IoT application module, and other optional modules. This combination of resources enables the application to function fully without access to upstream services.

   Azure SQL Edge provides the application with local storage of time-series and other data, streaming analytics, real-time scoring of data based on ML models, data protection, and real-time visibility to stored data.

2. When an internet connection is available, data is uploaded periodically through IoT Hub. IoT Hub stores streams of data in partitions for a configurable amount of time.

3. Data can be sent from IoT Hub to any of the long-term storage services in Azure, such as Azure Cosmos DB or SQL Database.

   Time-series data that is originally ingested and stored by Azure SQL Edge can also be sent from IoT Hub to the Azure Time Series Insights service in Azure for further aggregation, exploration, and processing. When the application is running in a connected state, streaming data can also be sent to Azure Stream Analytics for further real-time analysis of data for which Azure SQL Edge has detected an anomaly, such as a temperature spike.

4. From storage services, data can be sent to Azure Machine Learning for retraining of models. These updated models can then be sent back to Azure SQL Edge to improve AI performance for the IoT application.

5. From storage services, business analysts can use Power BI to analyze warehoused data.

6. Web, mobile, and other applications can be built on the serving layer as well. For example, you can expose APIs based on the storage data for third-party uses.

Use case 1: Fugro

Fugro is a geo-intelligence provider that helps clients in the energy sector gain vital insights for planned marine projects such as offshore wind farms. To support its clients in early projects phases, Fugro sends a team of oceanographers, meteorologists, and hydrographers to collect environmental data such as water depth, wind, waves, currents, and temperature. This marine geo-data, which can be collected through sensors on Fugro vessels or ocean buoys, is vital to understanding how a structure such as a wind farm might behave in extreme climates.

Until recently, the marine data that Fugro collected offshore was made available via a satellite link to the rest of the Fugro team, which is based at the company office on dry land. This data acquired over satellite was then imported into Microsoft SQL Server on premises. Finally, the Fugro team would run analytics on the collected data to derive insights that would help their clients produce cost-effective, sustainable designs for their energy projects.

However, this initial data-acquisition process led to some problems. The offshore locations where Fugro collects data are often remote, and acquiring this data quickly and consistently was challenging. There's no hardwired broadband over the ocean, and satellite link connections aren't dependable enough to handle large data transfers reliably. These conditions required hundreds of gigabytes in storage capacity in sea buoys, and because analytics could be performed only on land, Fugro was unable to drive real-time insights.

The company set out to fix these problems by moving analytics to the cloud and network edge. First, the Fugro team migrated its on-premises SQL Server environment to Microsoft Azure SQL Database in the cloud. Then, the team devised a new process in which Azure IoT Edge and Azure SQL Edge would be deployed locally at the offshore project site on unmanned vessels. Figure 5 depicts this new architecture.

Thanks to Azure SQL Edge, time-series data is now able to be ingested, queried, stored, and processed locally at the edge, saving the company significant time and costs associated with latency and bandwidth usage.

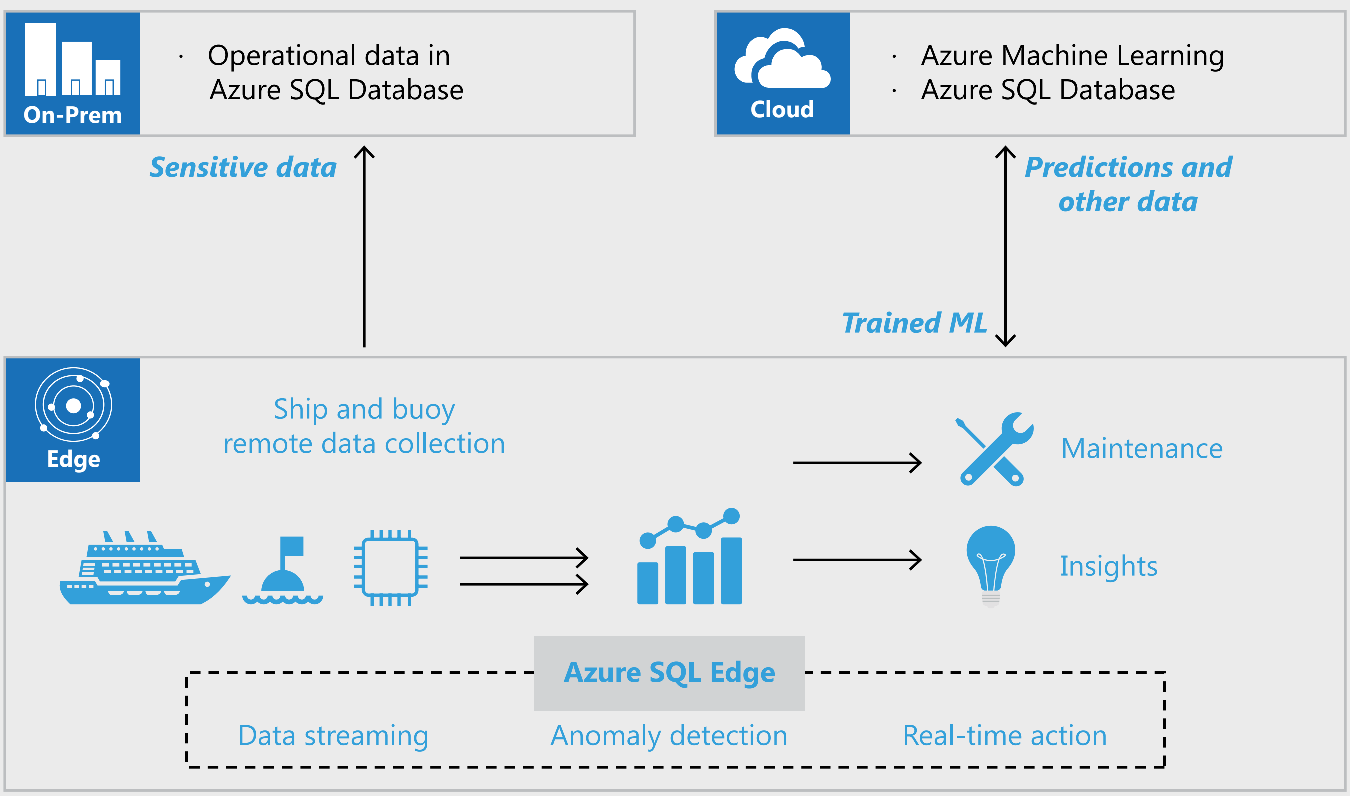


Figure 5. Fugro now uses Azure SQL Edge and Azure SQL Database to support predictive analytics and real-time actions on data collected at sea

Fugro’s data-collecting marine devices now also spend less time communicating with the cloud via satellite because they are able—in real time—to store and act on data collected at sea. And these assets still operate reliably when disconnected from the network. The latest state of each asset is automatically synchronized after it’s reconnected to ensure seamless operability, so satellite outages don’t slow down data delivery or analysis as much. Using Azure SQL Edge and IoT Edge as the foundation of its new solution, Fugro can conduct real-time scoring for improved analysis at the edge, with in-built AI and ML capabilities.

Another key advantage of the solution is that IoT Edge allows Fugro to remotely deploy Azure SQL Edge and gain SQL Server capabilities on a vessel almost instantly. With remote deployment, the company no longer needs to bring vessels back into port to update software, making management much simpler and more cost-effective

Finally, the company no longer has to build a solution from scratch for each instance. Fugro developers’ skills in C# translated easily to Azure. Developers can deploy system updates to all their vessel and cloud environments—wherever they are—using only one set of code.

The company is deploying the solution across nearly 30 oceangoing fleet vessels, effectively connecting edge devices to its cloud architecture so that data collected by vessel sensors flows into IoT Edge and then to Azure SQL Edge in near real time. The solution is so efficient that Fugro has reduced the time needed to process and deliver monthly data reports from offshore locations to client dashboards from two weeks to eight minutes.

With the help of IoT Edge and Azure SQL Edge, Fugro has improved the management, efficiency, and resilience of its solution infrastructure.

Use case 2: ZEISS

ZEISS is a legend in the world of optics, and it has long been revered by many scientists, photographers, and others for producing lenses of exceptional quality. Founded in 1846, the company today also develops custom optoelectronics solutions for research institutions, manufacturing firms, and other businesses.

Like many companies in the twenty-first century, ZEISS has recently begun a period of digital transformation. The goal of this transformation process is to improve the efficiency, precision, and accuracy of ZEISS manufacturing with the help of IoT and analytics. IoT sensors have been deployed throughout the production lines as part of this effort.

In designing its smart factory, ZEISS was keen to avoid a key limitation of many existing IoT solutions: manufacturers for years have been using IoT sensors to register readings of equipment for temperature, vibrational patterns, and other telemetry data; but traditionally, this captured data needed to be uploaded to the cloud for analysis. Uploading data to the cloud increases latency, leading to a slow response time and a delay in identifying manufacturing problems.

ZEISS determined that, to avoid these pitfalls and meet its efficiency goals, its IoT solution needed to incorporate edge computing, which reduces latency by performing analytics at or near the data source. However, there's also a complication in implementing such a solution: edge analytics require a data engine that is both powerful enough to perform analytics and lightweight enough to run on the types of small computing devices that could be deployed throughout the company's production lines.

For this role of edge data engine, ZEISS embraced Microsoft Azure SQL Edge. Azure SQL Edge currently stores IoT data and supports analytics on ZEISS production lines for eyeglass lenses, mechanical parts, and spectroscopic solutions.

Beyond reducing latency and allowing analytics to be performed at the network edge, another advantage of Azure SQL Edge is that it can synchronize the data collected through sensors to an upstream storage target in Microsoft SQL Server or Azure. This synchronization capability allows the collected IoT data to be consolidated, which then enables employees to connect to central dashboards that reveal data insights that are useful to their roles.

The result is a comprehensive data-collection and analytics solution that bridges the distance from edge to cloud. "Collecting this data helps everyone at every level of the factory," says Jochen Scheuerer, Head of Connected Smart Factory at ZEISS. "The decision-maker, the worker, the line manager—everyone is able to make decisions earlier, quicker, and more accurately."

"Before edge computing, by the time cloud analysis noted a problem, we had lost response time and sometimes wasted product," says Scheuerer. "With Azure SQL Edge, we reduce both our reaction time and the number of cycles needed."

As a pioneer in scientific optics, ZEISS focuses on driving innovation that will keep the company defining the future of optics and precision mechanics. "We have to act quickly to stay ahead of the market and our competitors," Scheuerer says. "Azure SQL Edge helps ZEISS become faster, delivering products that support our digitization projects, which in turn helps our customers win more business."

Use case 3: Microsoft Azure SQL Edge in agriculture (viticulture)

Winemakers contend with a dizzying array of threats in the vineyard that put their livelihoods at risk every season. Winemaking is costly and labor-intensive, and for winemakers to have any hope of making a profit, it's essential for them to produce a sizable yield of healthy, ripe grapes at harvest time.

But achieving this goal is often a struggle. Any given growing season might be too hot, too cold, too wet, or too dry, conditions that can make the vintage expensive and labor-intensive, all while leading to poor fruit quality and lower sales, not to mention a damaged reputation. Vintage-destroying diseases such as downy mildew are also often a lurking threat in the vineyard, and by the time the winemaker spots such a disease on the vines, it's often too late to save a good portion of the crop. Then there is the added threat of common equipment failures, such as irrigation leaks, that waste water and can damage fruit quality. Handling all of these challenges requires near-constant vigilance over many acres of vineyards, and painstaking vine care that can be expensive and back-breaking.

An IoT application for viticulture, built on the platform of Microsoft Azure SQL Edge, can significantly reduce risks for winemakers and help their businesses thrive. Many vineyards are located where an internet connection is unreliable, and this fact would normally make such a solution unfeasible. However, Azure SQL Edge supports local analytics, allowing IoT applications to provide enormous benefits to winemakers regardless of the strength of internet connectivity in their vineyards. Even without a reliable internet connection onsite, this type of solution can help lower costs, improve decision making, and raise fruit quality while allowing winemakers to manage crops remotely and spend less time walking through hundreds of acres of vineyards to monitor their crops. Instead, they can use the IoT solution to gain an overall view of the health of their vineyards in an instant and spend their time more productively.

As part of this solution, many IoT sensors and cameras are installed throughout the winemaker's vineyard at various sites above ground, below ground, and on aerial drones. These IoT sensors measure different factors including soil moisture, leaf moisture, grape size, humidity, sun exposure, wind speed, and signs of irrigation leaks. Collecting this data through sensors spares the winemaking staff many hours of traversing the vineyard rows to record similar information manually.

The telemetry data from these sensors is sent over a low-power wide-area network (LPWAN) to an IoT gateway device running Azure SQL Edge and the IoT viticulture application. On the IoT gateway, the viticulture application ingests the data with the help of the streaming analytics engine built into Azure SQL Edge. The streaming analytics engine is used to detect simple anomalies such as temperature spikes in real time, after which it aggregates the collected data in the local database.

Even when no internet connection is available on the IoT gateway, AI and/or ML is used on the data stored locally on the device to detect vine risk factors and inform precision farming, which can help reduce labor and material costs. For example, AI/ML can determine where in the vineyard there might be high risk of disease or infestation, which can improve wine consumer safety and reduce winemaker costs by enabling a limited use of pesticides and fungicides only where needed. AI/ML can also help determine exactly where irrigation might be necessary and optionally trigger targeted, automated irrigation without any need for an internet connection, again saving resources and reducing the winemaker's workload. Photos of the vines taken by drones and sent to the device can be analyzed through AI, and the solution can make recommendations about where pruning might be necessary, whether grapes in certain locations are too big, if the crop load is too heavy, and where sun exposure is suboptimal. The winemaker can connect to the IoT gateway without an internet connection via a mobile or web application over a wireless network and can view these recommended actions or browse visualizations of collected historical measurements.

When an internet connection is available, some or all of the collected telemetry data is synchronized to Azure Time Series Insights. Once the data is available online, vineyard staff can view recommendations and data visualizations from anywhere, freeing them from having to spend so much time in the vineyard. They can then also use the app to manually control irrigation if desired, based on AI-driven recommendations.

The ISV for the IoT viticulture application aggregates the vineyard measurements from all of its customers into larger datasets in Azure. These datasets are then used to train better and better predictive models for use in vineyards with similar weather patterns. These updated models are periodically pushed down to Azure SQL Edge running on the IoT gateways in each customer's vineyard, where the application can use them to further lower risks, reduce toil, save on costs, and make even better recommendations for the winemakers in the future.

Overall, the vineyard IoT application is a flexible edge-to-cloud solution powered by Azure SQL Edge. Regardless of whether an internet connection is available in the vineyard, Azure SQL Edge enables the application to use AI to relieve many of the difficulties of viticulture for winemakers, all the while helping to improve both profitability and product quality through precision, data-driven farming.

Use case 4: Microsoft Azure SQL Edge can help improve worker safety

Workers often face dangerous conditions on the job in sectors such as manufacturing and mining, and the high risk of being injured at work looms as an everyday threat for many men and women around the world. In fact, as many as 321,000 people are killed in accidents at the workplace each year, and many more are non-fatally hurt.[4]

Accidents on the job affect businesses primarily by taking a toll on workers physically, financially, and emotionally. When injuries are serious, they can deprive families of their only source of income. Beyond these potentially terrible human costs, safety incidents can also cost businesses millions of US dollars in lawsuits, medical costs, and fines.

IoT applications for worksite safety, developed for specific industries and built on Microsoft Azure SQL Edge, can help make jobs much safer for workers and improve the quality of their lives as a result. Such solutions can reduce the many costs, financial and other, associated with worker injury, and they can help improve worker morale by alleviating the fear of potential accidents.

An IoT application for worker safety includes sensors, an edge device, and a back-end cloud component. The IoT sensors are installed on workers in their helmets, coats, and wristbands, where they monitor status information such as breathing rate, skin temperature, toxic gas exposure, heart rate, posture, and body or head motion. Sensors are also installed inside equipment and throughout the general surroundings, measuring factors such as temperature, moisture, and pressure.

Telemetry data from sensors is sent through Wi-Fi to the application on a local device running Azure SQL Edge. The IoT application then uses trained ML models to score the data and detect when equipment is about to fail (through readings such as excessive temperature or pressure). Similarly, ML models can read sensor data to detect troublesome signs such as interrupted or fast breathing, excessive skin temperature, a faulty heart rate, or drooping eyelids.

Every second matters when workers are in jeopardy. Because the ML models run locally in Azure SQL Edge, equipment can automatically be shut down as soon as dangerous patterns in data are detected, without the delay that would normally be required to communicate with upstream servers in the datacenter or cloud. This quick response can help protect workers before any accident occurs. To improve detection of potential dangers, collected data can be sent to the cloud where ML models can be retrained.

Overall, worker-safety applications built on the platform of Azure SQL Edge can help reduce accident rates, increase worker productivity, and reduce costs for business. Because Azure SQL Edge supports local AI on small-footprint devices, near-instantaneous actions can be taken to prevent injury, improve employee morale, and raise operational confidence.

Simplify your edge infrastructure and IoT application development

Azure SQL Edge simplifies edge architecture and IoT application development for customers by supplying a small-footprint data engine that runs on a choice of platforms and delivers consistency in data, tooling, and security from edge to cloud. Powerful enough to run offline, but with the capability to connect seamlessly to a broader network of services, Azure SQL Edge includes built-in AI capabilities, support for time-series data, and native data synchronization with SQL Server and Azure services such as SQL Database.

With Azure SQL Edge, customers have a small but powerful data engine that meets the challenges of IoT, which empowers businesses to use and analyze data collected at the network edge to save costs, drive innovation, improve safety, and gain competitive advantage.

To get started with Azure SQL Edge preview, visit aka.ms/sqledge.

Links to more info

Azure SQL Edge product information (including frequently asked questions [FAQs]):
http://aka.ms/sqledge

Azure SQL Edge documentation: **https://docs.microsoft.com/en-us/azure/sql-database-edge/**

Deploy Azure SQL Edge through the Azure portal:
https://docs.microsoft.com/en-us/azure/sql-database-edge/deploy-portal

[1] Gartner. "Market Guide for Edge Computing Solutions for Industrial IoT." Santhosh Rao. 23 September 2019.

[2] For example, California SB-327, which took effect in January 2020, requires all "connected devices" to have a "reasonable security feature." For more information, see https://leginfo.legislature.ca.gov/faces/billNavClient.xhtml?bill_id=201720180SB327.

[3] National Institute of Standards and Technology Comprehensive Vulnerability Database.

[4] International Labour Organization. "ILO calls for urgent global action to fight occupational diseases." April 2013. Copyright © 2013 International Labour Organization. www.ilo.org/global/about-the-ilo/newsroom/news/WCMS_211627/lang--en/index.htm