Azure Data Lake connects supply chain data for advanced analytics

The power of data—data that shows patterns and predicts what’s ahead—helps to fuel digital transformation in the Microsoft supply chain. One building block of digital transformation is a modern data and analytics platform. Specifically, it’s a platform that allows you to gather, store, and process data of all sizes and types, from any data source, and that enables you to access it seamlessly to gain valuable insights.

In the end-to-end supply chain process, the right insights make it easier to identify trends and risks that help you ship on time, provide a quality product, save costs, and optimize inventory. The digital transformation journey is one that Microsoft has already begun, as discussed in Bringing digital transformation to the supply chain with Azure IoT Suite.

The supply chain engineering team at Microsoft provides solutions for core supply-chain functions related to hardware and devices like Microsoft Surface, Xbox, and HoloLens. Examples of these functions include sourcing of materials, planning, manufacturing, delivering products to consumers, and managing the customer care process.

These supply chain activities are highly interconnected and interdependent, yet we’ve often operated in silos—with data in disparate systems and supply chain teams making isolated decisions. While a common result of organic growth over time, this approach can make it difficult to connect the pieces and optimize the whole. This, in turn, makes it even more challenging to apply advanced analytics and machine learning for deeper insights.

An early step in this transformation was to bring this disparate supply chain data into one location—a data lake—rather than managing multiple data warehouses or manually co-locating data on an ad hoc basis. While a data lake doesn’t solve all problems, and may create a few new ones, it allows more immediate focus on value-added activities and innovation with data.

Solving business problems with Azure

To tackle these challenges and bring our data together, we moved from a traditional relational database-centric solution to a big data platform with Azure Data Lake Store as its foundation. Data Lake Store, the Microsoft hyperscale repository for big data analytic workloads, is essentially Hadoop for the cloud made simple. Although SQL Server continues to play an important role in the presentation of data, it’s just no longer a central storage component. This move to Data Lake Store has:

- Enabled us to say “yes” and take on more business requests, by giving us flexibility, scale, and speed. For example, we don’t have to worry about whether our infrastructure can accommodate ever-increasing amounts of data that we need to store and process.
- Enabled application-independent compute engines to be provisioned over a shared data foundation.
- Allowed us to quickly onboard new data assets—what used to take weeks now takes days.
- Provided simpler support for large-scale encryption at rest and in transit. We have out-of-the-box support for enterprise security and compliance, including Azure Active Directory authentication, and folder-level authorization via role-based and POSIX-style access control lists (ACLs).

Consolidating our data is key

Someday the industry may realize the dream of data federation, but today there’s big value in centralizing data and then connecting that data to our supply chain processes. In a blog post from 2016, Josh Klahr—VP of Product Management at AtScale—proposed six architectural principles that influenced our solution. The decision to implement our data lake in Data Lake Store meant we could realize the benefits of three of these principles, right out of the box.
Treat data as a shared asset. The co-location and scalability of Data Lake Store enables us to use the lake for a wide range of use cases and diverse target audiences.

Ensure security and access controls. Data Lake Store includes encryption and auditing, in addition to role- and POSIX-based controls at the folder level.

Eliminate data copies and movement. Data Lake Store can store and enable analysis of all our data in a single layer. While we still move small subsets of data to a database or reporting tool, we can meet many of our use cases by simply layering our compute engine in Spark over Data Lake Store. With this layering, we enable SQL-like experiences with Hive or any other supported capabilities in Spark, like machine learning, all with zero data movement.

**Our key business scenarios**

While there are obvious business benefits to adopting a big data platform, this was largely about technology modernization in the short-term with a focus on preserving business continuity. Even so, we defined some key scenarios that would help us prioritize what to go after on the new platform:

- Use data to identify why a device failed, predict manufacturing yield rates (the number of manufactured products that pass inspection the first time), and predict or even prevent future failures.
- Find the best prices for parts from suppliers and predict supplier prices at a component level for future buying optimization.
- Create accurate demand forecasting for hardware and devices for consumers, online stores, and frequent orders.

**Engineering and business: better together**

We help the supply chain by providing intelligence and data to make our planning and daily operations run more smoothly—and to optimize costs. Manufacturers, for example, need real-time—or earlier—information when defects occur. They need data about what’s failing and why. Day-to-day operations involve gathering, storing, processing, and visualizing data by:

- Gathering and storing the data in Data Lake Store.
- Applying a computing engine to process the data by using Spark.
- Visualizing the data with tools like Microsoft Power BI dashboards and reports.
- Creating predictive and prescriptive models.

**Our end-to-end team process**

From start to finish, our team process works as follows:

- We collaborate with our business partners to determine what data is needed to move the business forward.
- The platform team defines the architecture, designs solutions, performs data engineering, and automates and supports the solution from code to consumption.
- Our engineering team also builds denormalized information assets including Kimball-style dimension and fact tables housed in Azure SQL Data Warehouse and OLAP cubes. We’ve also built a robust data quality engine by using all Azure components, which serves as the centerpiece of our live-site process.
- We partner with the data science team who uses the platform to assist the business with analysis, and to gain deep insights through econometric modeling and machine learning.
- Our business partners build and maintain most of the presentation layer, including Power BI dashboards and SQL Server Reporting Services reports.
Using the intelligent cloud

To achieve our business goals, we needed a modern architecture that could serve as a foundation for digital transformation in the supply chain. In addition to the primary storage layer in Data Lake Store, we use a broad array of building blocks in Azure, with the key components being:

- **Azure Data Lake Store** as the primary means of storing our data.
- **Azure Data Factory** for the pipelines we use to ingest data from various sources.
- **Apache Spark for HDInsight** for transforming and querying data stored in the lake.
- **Azure SQL Data Warehouse**, for curated data assets, supporting read-only queries.

Figure 1 highlights the architecture that surrounds Data Lake Store for our end-to-end solution.

![Figure 1. Our technical architecture](image)

**Technical and design considerations**

If you’re planning to implement a similar solution for your supply chain data, here are a few points to consider:

- To co-locate data in a data lake effectively, determine how you’ll design and organize it. Data Lake Store has a file system like the one on your personal computer. For example, you wouldn’t store all your documents in the C: root directory. It’s similar with a data lake—everything is a file, with other options that can be tailored to a given requirement.

The folder structure in Data Lake Store reflects the levels of governance and ownership. For example, business partners might own everything that’s stored in a particular location and have authority over it. Data in another location might be managed by the engineering team and could have source-control mechanisms in place. Careful design choices in the folder structure will make governance much easier. The data in high-level folder paths can be user-owned, where individuals and teams can set their own policies locally. Or, the engineering teams can manage it.
Figure 2 shows an example of our approach. The arrow from bottom to top demonstrates increasing levels of data curation.

**Figure 2. Data governance and ownership**

Data assets are categorized as raw, gold, or cooking:

- **Raw.** Data assets are onboarded (full loads, delta enabled, timeseries) one by one as they come from the source, with no changes.
- **Gold.** Largely mirrored structure of raw data with assertions about data quality, de-duplication, and merging. Data is also exposed as Hive tables and available for consumption by individuals and tools.
- **Cooking.** Contains reporting assets that are usually the result of an extraction, transformation, and load (ETL) process. This includes all our dimensions, facts, and other information assets that are moved to Azure SQL Data Warehouse.

- With hardware, you have dedicated servers for environments like development, testing, and production. Often, you need data in your test environment that’s similar in size and quality to production data. When this happens, you must physically move it from production to testing and back. With a data lake, your folder structure can be your environment, with roots for testing, development, and production. To move data from one environment to another, point your source to the desired location. Rather than moving data between environments, you’re just reading the data using a different path.

**How do we get there from here?**

Figure 3 shows the process we followed to move from a data-warehouse-centric platform to our modern platform.

**Figure 3. Our approach for moving to a data lake**
Our approach consists of four phases:

- **Phase 1.** This stage is the starting point for most modernization projects with a relational data warehouse in a central role.
- **Phase 2.** This stage mostly focuses on adding the data lake and ingesting data sources in parallel. Optionally, we added Azure SQL Data Warehouse and copied processed dimensions and facts from our legacy data warehouse.
- **Phase 3.** This stage represents the bulk of the project effort. In most data platforms, the core ETL processing is the long pole, and moving that around is a lot of effort and risk. The dotted lines in the graphic denote processes being incrementally turned off in the legacy flow once migrated. This repeats until all are moved. Also, you’ll want to run these processes in parallel for some time for regression testing.
- **Phase 4.** This stage involves turning off the legacy data warehouse, with full processing occurring through the data lake.

**Implementation challenges**

A few of the challenges we’ve faced, but have overcome, include:

- **Initial misalignment on which technology to use.** As you might expect in a company as large as Microsoft, there are many options, and very good internal ones, that aren’t available to the public. We worked with our business partners to refine our options to two candidates, and public Azure was the clear choice.
- **Aggressive timeline.** For a process that might typically be a multi-year effort, our core implementation was done in less than one year. We have about 150 terabytes of raw data over many data sources, and lots of dependencies. We were able to complete this transition rapidly because of the power of Azure and a determined engineering team.
- **Simple, performant consumption layer.** Using the Spark execution engine over Data Lake Store solved many of our problems, especially when paired with Jupyter Notebooks. One of our core architectural principals was to minimize data movement after we had moved to the data lake. Microsoft has exciting developments related to Azure SQL Data Warehouse, Azure Analysis Services, and possibly some other surprises. There is more work to do in achieving parity with traditional approaches for simple and performant consumption of data in this crucial last mile.

**Recommendations based on our lessons learned**

Some of the lessons we’ve learned thus far include:

- **The more you can invest in learning new technologies, the better.** Most Microsoft customers who have worked with a relational database are comfortable writing T-SQL, and the same was true for us. But there is a learning curve, for example, to pivot from SQL Server to Hive (HQL), and even more so for our engineers writing complex code in languages like Scala.
- **Understand the various Azure roadmaps.** The various components in Azure have different levels of maturity. Take time to understand how the various components in Azure work together, what gaps if any there may be for your specific requirements, proceed in an Agile fashion, and plan or the future.
- **Data is power, but it can exist in different forms.** Have a clear strategy for master data management, so that you can provide the most accurate information and a single data source. Proactively plan for and manage data quality. We decided to build our own custom tooling for big data quality management.
Benefits we’ve seen

So far, we’ve seen the following benefits:

- **A home for all data.** We no longer have to be concerned that the next project will be the tipping point for running out of SAN space or hitting a hidden performance limit.
- **Less latency than our previous BI system.** We have reduced onboarding time-to-value across the board and, on some critical datasets, we’ve reduced the time that it takes to load information from days to hours.
- **Improved data consistency.** We built a data dictionary that spans the supply chain. The data dictionary keeps track of standardized KPI definitions so that teams have a common taxonomy.

Next steps

We’re still in the early stages of digitally transforming our supply chain, but we’re already seeing the benefit of having centralized data, a more flexible data processing model that scales as needed, and connected processes. Our roadmap includes using what we’ve already done as a foundation and layering other capabilities including:

- **Expand our use of predictive and prescriptive analytics.** For example, our data science team is currently building on this foundation to deliver a dynamic regression model for better prediction of sales through demand. Also, we’re planning a component “recommender system” to optimize part selection during design and sourcing.
- **Get new insights into the customer life cycle.** For example, how does the supply chain contribute to customers’ and partners’ perception of doing business with “One Microsoft”? How do we measure the lifetime value of a customer?
- **Continue to modernize our compute architecture.** We’ve applied Azure AD ACLs against folders in Azure Data Lake. We’ll continue moving toward a seamless governed experience as users move between the various query and visualization tools. Additionally, we’re in the process of delivering capabilities that build upon other foundational work in our “hot path” for data, including near real-time statistical process control for better yield management and smart services.

For more information

Microsoft IT Showcase

[Microsoft IT Showcase](microsoft.com/itshowcase)

The Microsoft Modern Data Warehouse

A Modern Data Warehouse Architecture

© 2017 Microsoft Corporation. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.