David Chappell

STREAMING SCENARIOS USING THE MICROSOFT DATA PLATFORM

A GUIDE FOR IT LEADERS

Sponsored by Microsoft Corporation

Contents

Microsoft's Data Platform: The Big Picture 3

Scenario: Processing Data from Many Devices in the Internet of Things 5

Technology Snapshot: Azure Stream Analytics 5

Technology Snapshot: HDInsight Storm 5

Technology Snapshot: HDInsight Spark Streaming 6

Technology Snapshot: Azure IoT Hub 6

Describing the Scenario 6

Understanding Your Options 7

Scenario: Finding Trends in IoT Data 8

Technology Snapshot: Azure Blobs 8

Describing the Scenario 9

Understanding Your Options 10

Scenario: Detecting Fraud in Real Time 10

Technology Snapshot: Power BI 10

Technology Snapshot: Azure Event Hubs 11

Describing the Scenario 11

Understanding Your Options 12

Scenario: Doing Complex Event Processing On-Premises 12

Technology Snapshot: SQL Server StreamInsight 13

Describing the Scenario 13

Understanding Your Options 14

Conclusion 14

About the Author 14

Microsoft's Data Platform: The Big Picture

The value we get from information technology revolves around data. And the volume, variety, and velocity of that data seem to increase every day. Accordingly, IT vendors offer a broad range of data technologies. Viewed together, these technologies comprise a *data platform*.

A useful way to think about the technologies in a data platform is to divide them into three categories based on the kind of data they work with. Those categories are:

- *Operational data*, such as transactional data used by a banking system, an online retailer, or an ERP application. This data is typically both read and written by applications, commonly in response to user requests. A banking application might read your account balance, for instance, then write a new value to reflect a deposit you make. And while operational data was once almost entirely relational, the increasing volume and variety of data have changed this. Today, working with unstructured operational data can be just as important.

- *Analytical data*, such as the information kept in a data warehouse. This data is typically read-only, and it usually includes historical information extracted over time from other data sources, such as operational databases. Analytical data is commonly used for things such as business intelligence and machine learning, and like operational data, it can be either relational or unstructured.

- *Streaming data*, such as data produced by sensors. The defining characteristic of streaming data is velocity; if the data isn't processed quickly, it can lose a large share of its value. Many streaming scenarios today relate to the Internet of Things (IoT), where the focus is on interacting with data provided by lots of devices. Streaming data is also used in other situations, such as analyzing financial transactions as they happen. In both cases, the challenge is to work effectively with large amounts of data being produced in real time.

The Microsoft data platform provides technologies for all three categories, along with connections among the three. Figure 1 summarizes the platform's offerings in each area.

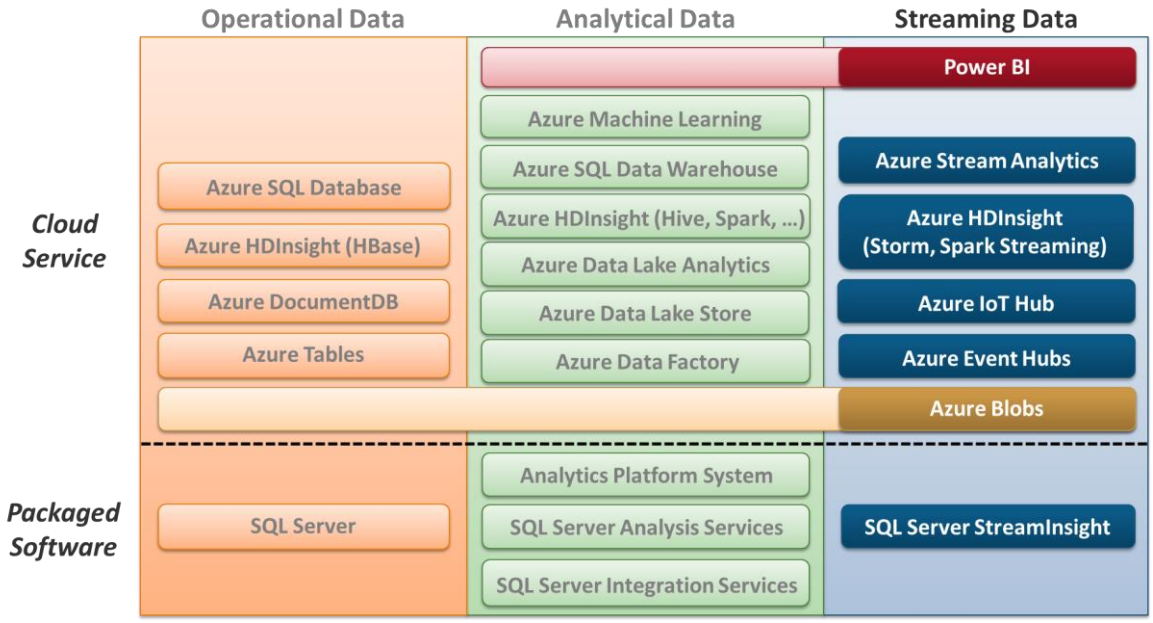

Figure 1: The Microsoft data platform includes cloud services and packaged software for working with operational data, analytical data, and streaming data.

This paper focuses on the rightmost column in the figure, Microsoft's offerings for working with streaming data. (For more on the other two categories, see the companion papers *Operational Data Scenarios Using the Microsoft Data Platform* and *Analytical Data Scenarios Using the Microsoft Data Platform*.) And don't be confused by the diagram: These technologies aren't layered in the sense that each one depends on the others shown below it. Instead, think of each column as a group of technologies for working with data in a particular way. Also, realize that the lines between the columns are quite permeable—the technologies are used in various combinations. For example, the streaming technologies in the right column are often used together with the analytical technologies in the center column.

The clearest way to understand how these offerings can help your organization is to walk through scenarios that use them. And given how important cloud computing is to IT leaders today, those scenarios should largely use the cloud in some way. This paper looks at four streaming data challenges organizations often face, describing how the Microsoft data platform addresses each one. The scenarios are the following:

- Processing data from many devices in the Internet of Things (IoT)
- Finding trends in IoT data
- Detecting fraud in real time
- Doing complex event processing on premises

Along the way, we'll briefly examine each of the streaming data technologies shown in Figure 1. The goal is to provide a big-picture view of how the Microsoft data platform addresses the challenges of working with streaming data.

Scenario: Processing Data from Many Devices in the Internet of Things

The essence of streaming data is high velocity. If this data isn't analyzed immediately after it's created, it loses much of its value. A common example of this is information produced by many small devices. These might be sensors on an oil well, meters monitoring energy use in houses, fitness trackers on people's wrists, toll payment devices in cars, or many other things. What they all have in common is that they generate lots of useful information, with new data being produced constantly. This is the world of IoT.

One option would be to store all of this data as it arrives, then process it later using analytical technologies. In fact, this is often done; batch processing of stored streaming data can be useful. But it's often at least as important to process and respond to this data in real time. Doing batch processing only after data has accumulated is too slow for many useful applications, and so traditional data analysis technologies aren't enough. What's needed are technologies designed to continually process a stream of incoming data.

But where should those technologies run? Given that streaming data often comes from lots of devices that can be located anywhere in the world, there's an obvious answer: the cloud. Handling streaming data in the cloud provides both global accessibility and the ability to scale up and down as needed. This is why most of the Microsoft data platform's technologies for streaming data are part of Microsoft Azure. Those technologies include Azure Stream Analytics, Azure HDInsight Storm, Azure HDInsight Spark Streaming, and Azure IoT Hub.

Technology Snapshot: Azure Stream Analytics

Software that analyzes streaming data needs to do several things. It must handle fast-moving information with very little delay, i.e., with low latency. It must also help its users work with this data in useful ways, hiding as much complexity as possible. These are exactly the goals of Stream Analytics.

One of the most common things we want to do with streaming data is understand what's happening in that stream within specific periods of time. Maybe we want to know how many cars have passed through an automated toll booth in the last three minutes, for example, or how much electricity was used by houses in a particular neighborhood in the last hour. Stream Analytics is designed to make this easy to do. A developer can use the Stream Analytics Query Language, a subset of T-SQL, to issues queries on an incoming stream of data. Each query can specify a window of time to which the query applies, returning a result for just the data that arrives within that window. And once it's started, the query keeps running, sending back results for each window. Rather than querying tables, as in a relational database, Stream Analytics instead allows querying slices of an incoming stream.

Technology Snapshot: HDInsight Storm

HDInsight, Microsoft's cloud implementation of Hadoop and more, includes several different technologies. Among them are MapReduce, Hive, and Pig, all of which are commonly used for analyzing large amounts of analytical data on disk, and HBase, a store for operational data. HDInsight also provides Storm, a technology for working with streaming data.

Storm is similar in some ways to Stream Analytics. Both run in the cloud, and both support applications that process streaming data. In Storm, those applications are created using *spouts* and *bolts*. A spout accepts incoming data streams, while a bolt processes streaming data in some way. An application, called a *topology*, is made up of

spouts and bolts. Storm takes a quite general approach to working with data streams, and so it's useful in a broad range of streaming scenarios.

Technology Snapshot: HDInsight Spark Streaming

Along with traditional Hadoop technologies, HDInsight also provides Spark as a cloud service. Spark is an integrated set of open source technologies that can run on a Hadoop cluster. The Spark family includes options for analyzing large amounts of operational data, doing machine learning, and more. It also includes Spark Streaming, a technology for working with streaming data.

Spark Streaming is similar to Storm in some ways. Like Storm, it's a general-purpose technology for processing streaming data. Unlike Storm, Spark Streaming is implemented as an extension to the basic Spark engine—it's not an add-on technology. This tight connection can make Spark applications faster, since there's less need to move data between components, and easier to create, since everything uses the same core Spark technology. Because of this, Spark Streaming (and Spark in general) are getting more popular by the day.

Given the similarities between Spark Streaming, Storm, and Stream Analytics, which one should you choose? The answer depends on your situation—there's no single right answer. For guidance on making this choice, see *Understanding Your Options* later in this section.

Technology Snapshot: Azure IoT Hub

Stream Analytics, HDInsight Storm, and Spark Streaming all let your organization create software that processes streaming data. But none of them is designed to take in and buffer massive amounts of streaming data, something that's commonly required in IoT scenarios. Without some kind of buffering in front of these stream processing services, data will be lost.

Azure IoT Hub addresses this problem. This cloud service is commonly used in front of Stream Analytics, Storm, or Spark Streaming, providing a place to store incoming data until it's processed. IoT Hub can handle large amounts of incoming data from many devices, and it also provides a way to communicate back to those devices.

Describing the Scenario

A typical IoT scenario starts with devices generating events. Those devices might speak IP natively, which lets them talk directly to cloud services, or they might rely on an intermediary to do this. Whatever option is used, these cloud services see a stream of events coming in from devices. Figure 2 shows how this looks.

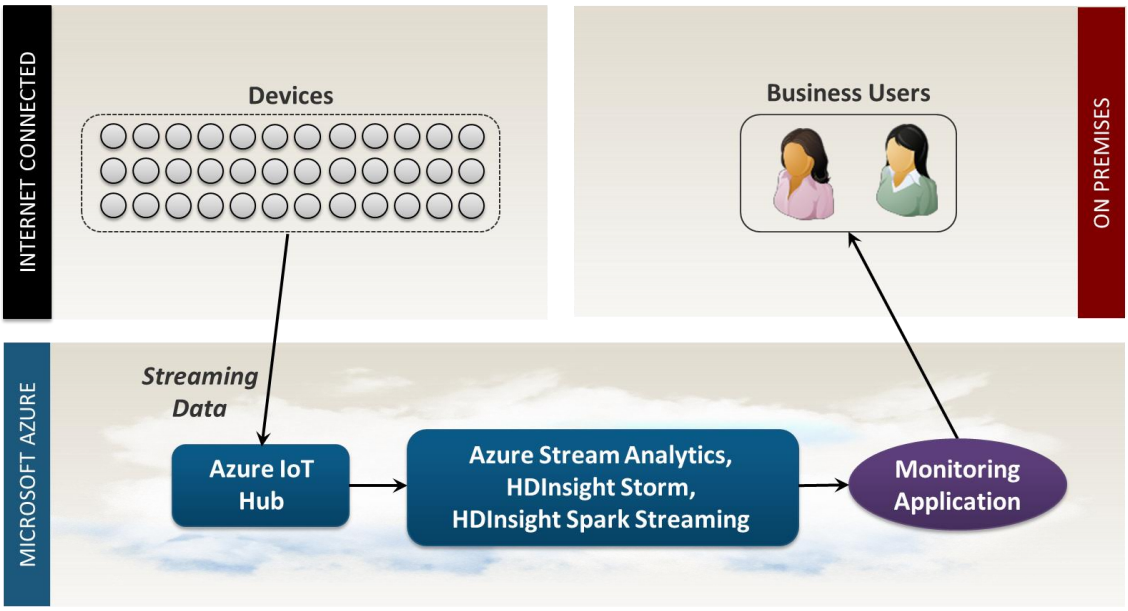

Figure 2: The basic IoT scenario, processing events from many devices, can use Stream Analytics, Storm, or Spark Streaming.

As the figure shows, incoming data is typically received and buffered by IoT Hubs. Any of Azure's stream processing technologies—Stream Analytics, Storm, or Spark—can then be used to examine that data. In this example, the result of that processing is sent to a monitoring application, which provides a custom user interface that gives business users the information they need.

Understanding Your Options

Since a basic IoT scenario can use any of Stream Analytics, Storm, or Spark Streaming, which one should you choose? The choice commonly depends on these factors:

- If your application is doing time-based queries, Stream Analytics is probably a better choice. This cloud service is designed to answer questions like this, and its SQL-based query language will likely be easier for your developers to understand. It's possible to do time-based queries with Storm and Spark Streaming, but since neither one is specifically designed to make this easy to do, your development team will probably need to write more code.

- If your application is doing event-based queries or other kinds of stream processing that go beyond what Stream Analytics is designed to do, either Storm or Spark Streaming is likely to be a better option. They're more customizable, and they let your developers work in more general programming languages rather than just the Stream Analytics Query Language. Both bring a bit more complexity, but Microsoft and the open source community provide a range of software (such as existing Storm spouts and bolts) to make developers' lives easier.

- HDInsight is an ecosystem of related technologies. Using any of them requires you to create an HDInsight cluster on Azure. If you're already using an HDInsight cluster for, say, data analysis with Hive, using Storm is a natural extension. If you're using an HDInsight cluster with Spark to do data analysis, using Spark Streaming

probably makes sense. If you're not using HDInsight for anything else, choosing either Storm or Spark Streaming will require you to spin up and pay for an HDInsight cluster. Stream Analytics, by contrast, is a managed service. You don't need to create your own cluster to use it, which simplifies getting started with streaming applications.

Having lots of options is generally a good thing. It does, however, require you to think a bit more to make the right choice for your situation.

Azure IoT Suite

Addressing real IoT scenarios commonly requires combining multiple technologies from the Microsoft data platform. As just shown, for example, a full solution might require using IoT Hub, Stream Analytics, and more. To make this easier to do, Microsoft provides Azure IoT Suite.

IoT Suite addresses the two main challenges you're likely to face in creating an IoT solution on Azure. First, it provides pre-configured solutions for common scenarios, including remote monitoring and predictive maintenance. Rather than making your developers choose and connect the various Azure components, IoT Suite does much of that for them. Second, IoT Suite provides a single SKU for a variety of IoT-related Azure technologies. Rather than guessing how much a solution will cost, you can know the price up front.

The Internet of Things will likely affect most organizations, including yours. The goal of Azure IoT suite is to provide a broad technology solution for IoT applications, along with a predictable price for running those applications.

Scenario: Finding Trends in IoT Data

Once an organization is able to accept and process IoT data, the next thing it often wants to do is look for atypical trends in that data. For example, an application that processes streaming data from oil wells might want to raise an alert if a well's pressure is steadily increasing, or maybe an application that monitors refrigerators for a storage company needs to let workers know if the average temperature for any of them is falling over the last 30 minutes. In both of these examples, changes like these might signal the need for maintenance or repair.

Finding atypical trends usually isn't hard. All that's required is to monitor values in streaming data, then notice when the trends differ from the norm. But figuring out what that norm looks like can require examining large amounts of historical streaming data. To store that historical data, organizations can use Azure Blobs.

Technology Snapshot: Azure Blobs

The term "blob" is an acronym for Binary Large OBject, and that's exactly what Azure Blobs store: raw binary data, Blob storage is quite scalable—a single blob can hold hundreds of gigabytes of data—and relatively inexpensive at

just a few cents per gigabyte per month. If you need to store large amounts of unstructured data as cheaply as possible, Azure Blobs are hard to beat.

Describing the Scenario

Imagine once again a standard IoT scenario where lots of devices are sending lots of streaming data to the cloud for processing. The code that examines this data is looking for atypical trends, unusual situations it needs to flag. Figure 3 shows how this might look.

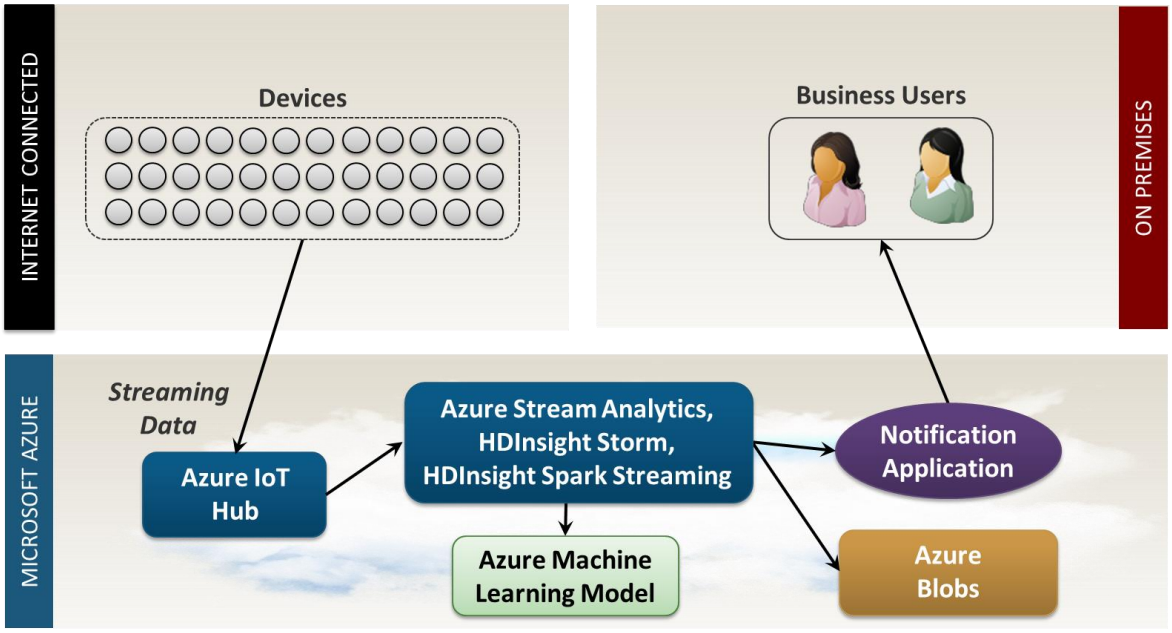

Figure 3: Combining streaming data with machine learning allows detecting trends in streaming data, then acting on these trends.

Much like the previous scenario, the incoming data is first buffered by IoT Hub, then processed by a streaming data technology: Stream Analytics, Storm, or Spark Streaming. This streaming technology does several things with the data it receives. First, it calls into a model created with Azure Machine Learning (ML). As its name suggests, Azure ML is a machine learning technology built on the Azure platform. It can find patterns in large amounts of data, then create a *model* that's able to detect those same patterns in new data. For example, suppose Azure ML was given data about various oil well parameters, including an indication of when each well had problems. It might be able to find patterns in this data that predicted failure, then produce a model that recognizes those patterns.

The streaming technology passes the model current data from a stream. The model can then return the probability that this new data matches a known pattern. If there is a match, such as when an oil well sends data indicating that it's about to fail, the streaming technology can take action. In this example, it notifies business users of the likely problem through a custom application.

Rather than just throw incoming data away, streaming applications commonly save it for later analysis. Accordingly, the streaming technology in this scenario also writes all of the data it receives in this scenario into Blobs. This historical data can then be used in various ways. For example, Azure ML might re-read this data

periodically to create a new model reflecting current patterns in the streaming data. This would let the application keep up to date with new kinds of anomalies.

Understanding Your Options

Figure 3 is simplified a bit to make the main points clearer. In fact, there are other things to think about, including the following:

- Historical data stored in Blobs might be read by Azure ML, as just described. It might also be examined by other analytical technologies in the Microsoft data platform, including Hive, Spark, and a Microsoft-created approach called Azure Data Lake Analytics. All of these are capable of analyzing large amounts of unstructured data in parallel.

- Along with Azure Blobs, the Microsoft data platform includes other options that might be used to store streaming data. For example, Azure Data Lake Store is a cloud offering that implements the Hadoop Distributed File System (HDFS) as a service. Especially for data that will be used for later analysis, Azure Data Lake Store can offer more scale and better performance than Blobs.

- As the figure shows, a streaming technology can send streaming data to several outputs simultaneously. It can even send that data to other Azure services, letting the same stream of data be processed in multiple ways.

Scenario: Detecting Fraud in Real Time

Fraud is a problem in many businesses, and detecting it quickly can be worth a lot of money. In the modern world, where so much fraud takes place on devices, streaming data can help do this.

In some situations, the best approach is to let software respond automatically to fraud. If a mobile phone company can detect fraudulent calls, for example, it can immediately shut down the offending phone. But even in cases like this, it's also important to let people track fraud. This requires giving them a way to see what's happening right now, something that streaming data can provide. To help people work with streaming data and other information through a common user interface, the Microsoft data platform provides Power BI.

Technology Snapshot: Power BI

Power BI is a cloud-based service that lets its users access diverse data from anywhere. It can present up-to-the-minute views from many different sources, then make those views accessible via browsers and mobile devices. Figure 4 shows an example of a Power BI interface.

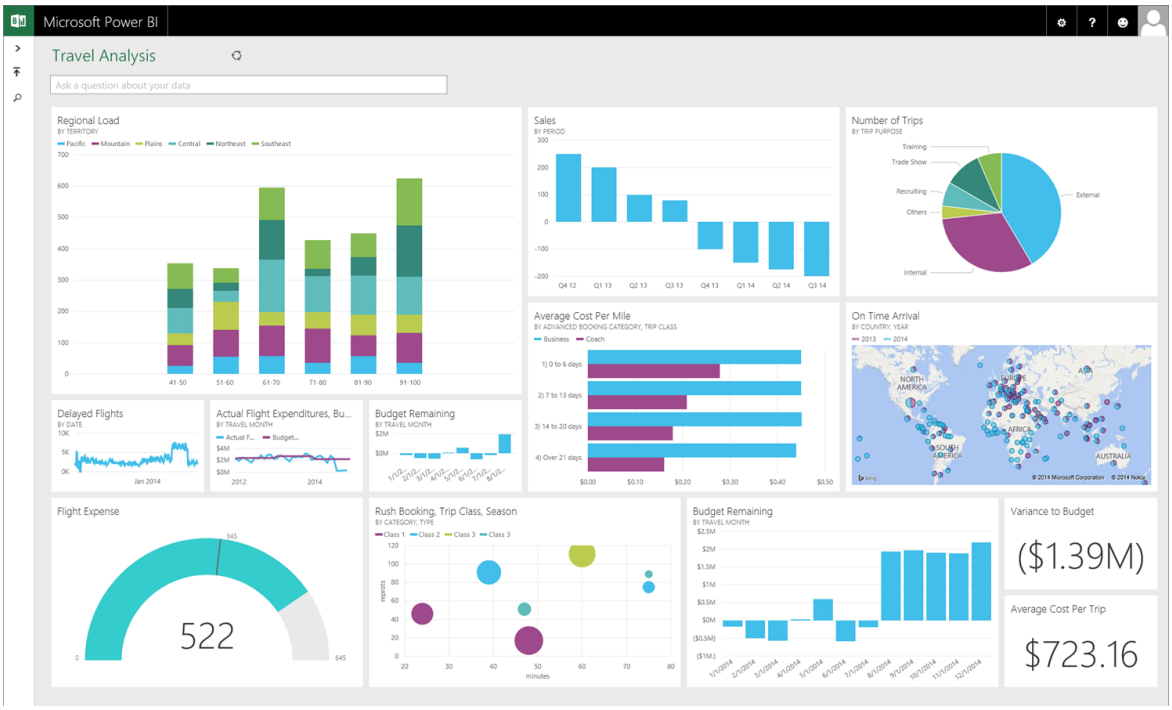

Figure 4: Power BI is a cloud-based service that provides a common user interface to data from many sources, including streaming data.

As this example suggests, Power BI can display information from many different sources in a unified way: cloud applications such as Office 365, data analysis technologies, and others. Power BI also provides a designer that lets business users create reports and dashboards on their own—they don't need to rely on developers to do this. And although Power BI isn't designed just to display streaming data, expect to see more and more Power BI interfaces showing results from real time streams. This kind of data has become too important to ignore.

Technology Snapshot: Azure Event Hubs

As described earlier, working with streaming data commonly requires buffering. Without this, you're likely to lose some incoming events. For IoT scenarios, the best choice for doing this is IoT Hub. For other scenarios, however, Microsoft provides Azure Event Hubs.

Like IoT Hub, the Event Hubs service provides a place to store incoming data until it can be processed. Event Hubs can handle as much as a gigabyte of data per second spread across billions of incoming events, and by default, the buffered data will be stored for 24 hours. Unlike IoT Hub, however, the Event Hubs service doesn't provide a way to talk back to the devices generating that data. This service is also intended to be used in more general scenarios, such as passing streams of data between different Azure stream processing services. It's a little bit simpler than IoT Hub—in fact, IoT Hub is built on top of Event Hubs—but it's also aimed at a broader range of situations.

Describing the Scenario

Think about a mobile phone company with users around the world. Each of its phones contains a subscriber identity module, more often called a *SIM* card. Among other things, the SIM card defines who will get the bill for calls made from this phone.

Suppose criminals find a way to duplicate the information on a phone's SIM card. They could then make calls from another phone using this duplicate SIM, with those calls billed to somebody else. The mobile phone company might use streaming data to detect this kind of fraud in real time, as Figure 5 shows.

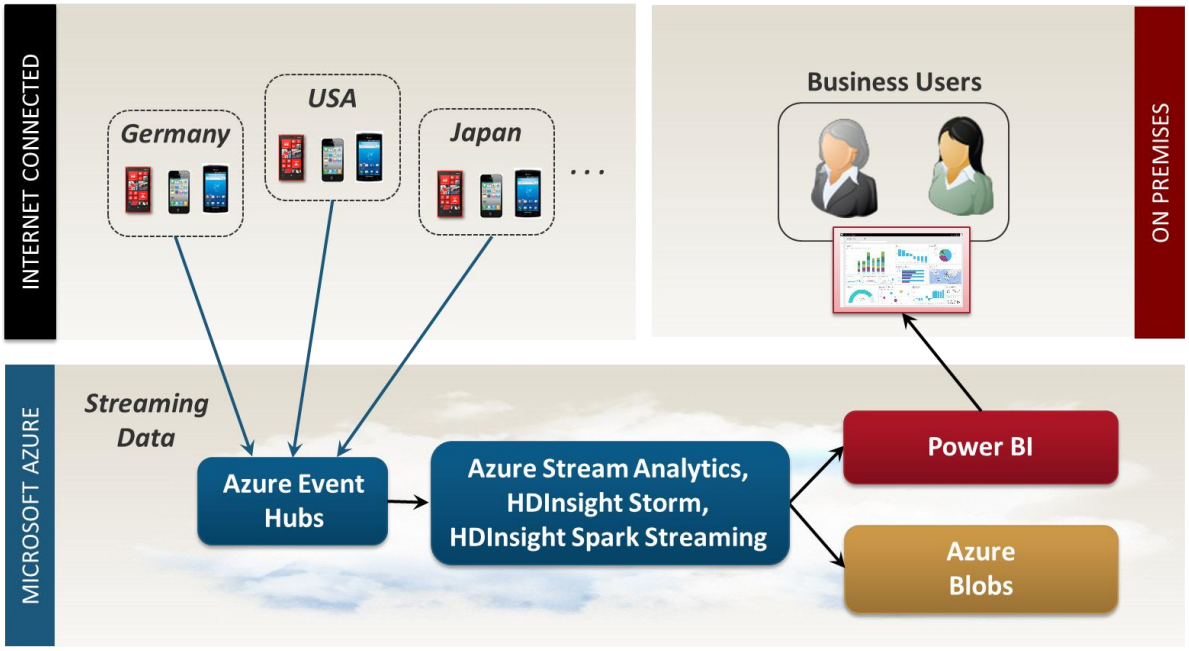

Figure 5: Processing call data in real time can help detect fraud as it's happening.

In this scenario, data about every call made from every phone in every country is sent into Event Hubs. This data includes the location the call was made from, such as the cell tower that handled the outgoing call. A streaming technology—Stream Analytics, Storm, or Spark Streaming—can use this data stream to detect whether the same SIM card is being used in two places within an unreasonably short interval.

For example, the query might look for calls made in Germany and Japan inside the same five-minute window. If this situation appears, the phone company can shut down the phones immediately, preventing the fraud from continuing. Real time information about this type of fraud can also be displayed to business users through Power BI, while the detailed data is written to Azure Blobs for further analysis.

Understanding Your Options

This scenario is much like the one that preceded it, so all of the issues listed there still apply. It's also worth pointing out that the large amount of streaming data collected in this scenario might let the phone company find patterns that indicate other types of fraud. Fraud is an anomaly, so just as in the previous scenario, the firm could use Azure Machine Learning to do this.

Scenario: Doing Complex Event Processing On-Premises

Handling streaming data in the cloud has much to recommend it. Cloud technologies such as Event Hubs and Stream Analytics are broadly accessible and quite scalable. But taking this approach means that the data must leave your organization to be processed. There are cases where this isn't possible.

For example, think about a financial services firm that needs to analyze regulated data streams in real time to make better trading decisions. Depending on where that firm is located, it might be prohibited from sending this data outside its own borders for processing. If there's no Azure datacenter in that country, the firm can't use Azure's streaming technologies. Instead, it must process this streaming data on premises. For situations like this, the Microsoft data platform provides SQL Server StreamInsight.

Technology Snapshot: SQL Server StreamInsight

StreamInsight is in many ways similar to Azure Stream Analytics. Both are designed to process streaming data in real time, and both allow defining relational queries on slices of a stream. But while Stream Analytics is often viewed through an IoT lens, StreamInsight is aimed at scenarios that fall under the heading of *complex event processing (CEP)*.

The concept of CEP isn't all that different from working with IoT data. As with IoT, a CEP application processes incoming streams of events, then identifies and responds to the important things it sees, whatever those might be. CEP is more explicitly business-focused, however, and so StreamInsight supports options such as incorporating key performance indicators (KPIs) directly into the logic of a CEP application. It also lets developers create applications using .NET languages such as C#. While StreamInsight is an older technology than Stream Analytics, it can still be the right choice in some scenarios.

Describing the Scenario

Suppose a financial services firm needs to do event processing on financial data streams produced by several different applications. Perhaps this processing drives the firm's trading algorithms, for example, or is used to make real-time pricing decisions. Figure 6 shows how this might look if it's done on premises using StreamInsight.

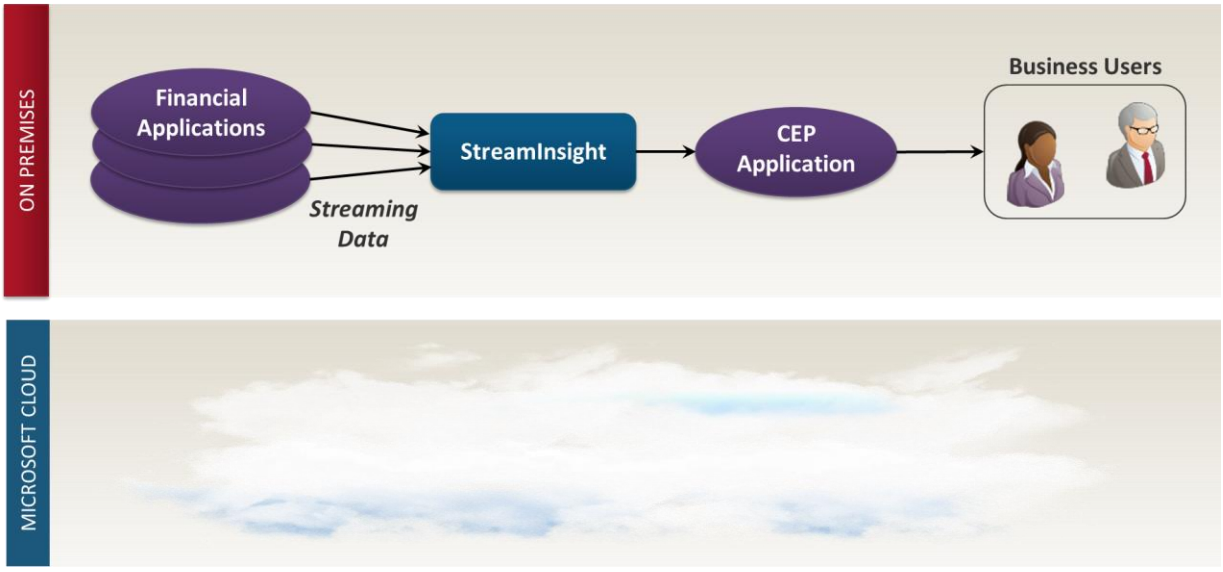

Figure 6: StreamInsight allows processing streaming data within your own datacenter.

In this example, the financial applications communicate with StreamInsight over the firm's internal network, so the streams of regulated data never leave the premises. A CEP application then processes these streams, displaying the result to business users.

Understanding Your Options

StreamInsight is a supported Microsoft product, and it's a reasonable choice for an on-premises CEP scenario today. Still, because StreamInsight is a product rather than a cloud service, it's more work to set up and manage than Stream Analytics. It's also significantly less scalable, which can be a concern in some situations.

In fact, it's clear that Microsoft's focus for working with streaming data today is in the cloud, and so you should expect new features and new technologies to appear here first. Given this, it's worth considering whether you can use Azure Stream Analytics, Storm, or Spark Streaming for a new data streaming project rather than StreamInsight.

Conclusion

A modern data platform must include technologies for working with operational data, analytical data, and streaming data. The first two of these have been with us for decades, while the value of streaming data has become apparent more recently.

To support streaming data, the Microsoft data platform provides several different options, both in the cloud and on premises. The cloud technologies are often used in IoT scenarios, although they're commonly applied to other kinds of streaming problems as well, while the on-premises solution is focused on complex event processing. Whatever problem they're addressing, these streaming technologies can be combined in useful ways with other components of the platform, such as those for machine learning. The goal is to address the streaming data needs of modern organizations in an effective and integrated way.

About the Author

David Chappell is Principal of Chappell & Associates (http://www.davidchappell.com) in San Francisco, California. Through his speaking, writing, and consulting, he helps people around the world understand, use, and make better decisions about new technologies.