


Real-time fraud detection on Azure

07/05/2018 • 4 minutes to read • Contributors 

In this article

[Relevant use cases](#)

[Architecture](#)

[Considerations](#)

[Deploy the scenario](#)

[Pricing](#)

[Related resources](#)

This example scenario is relevant to organizations that need to analyze data in real time to detect fraudulent transactions or other anomalous activity.

Potential applications include identifying fraudulent credit card activity or mobile phone calls. Traditional online analytical systems might take hours to transform and analyze the data to identify anomalous activity.

By using fully managed Azure services such as Event Hubs and Stream Analytics, companies can eliminate the need to manage individual servers, while reducing costs and leveraging Microsoft's expertise in cloud-scale data ingestion and real-time analytics. This scenario specifically addresses the detection of fraudulent activity. If you have other needs for data analytics, you should review the list of available [Azure Analytics services](#).

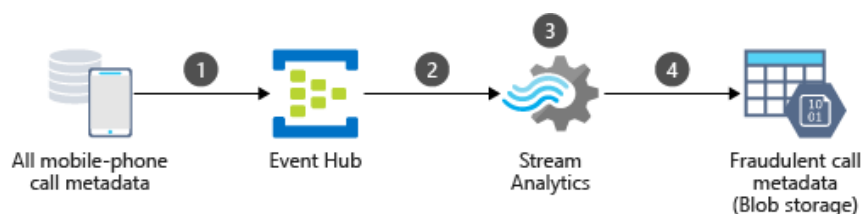
This sample represents one part of a broader data processing architecture and strategy. Other options for this aspect of an overall architecture are discussed later in this article.

Relevant use cases

Other relevant use cases include:

- Detecting fraudulent mobile-phone calls in telecommunications scenarios.
- Identifying fraudulent credit card transactions for banking institutions.
- Identifying fraudulent purchases in retail or e-commerce scenarios.

Architecture



This scenario covers the back-end components of a real-time analytics pipeline. Data flows through the scenario as follows:

1. Mobile phone call metadata is sent from the source system to an Azure Event Hubs instance.
2. A Stream Analytics job is started, which receives data via the event hub source.
3. The Stream Analytics job runs a predefined query to transform the input stream and analyze it based on a fraudulent-transaction algorithm. This query uses a tumbling window to segment the stream into distinct temporal units.

4. The Stream Analytics job writes the transformed stream representing detected fraudulent calls to an output sink in Azure Blob storage.

Components

- [Azure Event Hubs](#) is a real-time streaming platform and event ingestion service, capable of receiving and processing millions of events per second. Event Hubs can process and store events, data, or telemetry produced by distributed software and devices. In this scenario, Event Hubs receives all phone call metadata to be analyzed for fraudulent activity.
- [Azure Stream Analytics](#) is an event-processing engine that can analyze high volumes of data streaming from devices and other data sources. It also supports extracting information from data streams to identify patterns and relationships. These patterns can trigger other downstream actions. In this scenario, Stream Analytics transforms the input stream from Event Hubs to identify fraudulent calls.
- [Blob storage](#) is used in this scenario to store the results of the Stream Analytics job.

Considerations

Alternatives

Many technology choices are available for real-time message ingestion, data storage, stream processing, storage of analytical data, and analytics and reporting. For an overview of these options, their capabilities, and key selection criteria, see [Big data architectures: Real-time processing](#) in the Azure Data Architecture Guide.

Additionally, more complex algorithms for fraud detection can be produced by various machine learning services in Azure. For an overview of these options, see [Technology choices for machine learning](#) in the [Azure Data Architecture Guide](#).

Availability

Azure Monitor provides unified user interfaces for monitoring across various Azure services. For more information, see [Monitoring in Microsoft Azure](#). Event Hubs and Stream Analytics are both integrated with Azure Monitor.

Scalability

The components of this scenario are designed for hyper-scale ingestion and massively parallel real-time analytics. Azure Event Hubs is highly scalable, capable of receiving and processing millions of events per second with low latency. Event Hubs can [automatically scale up](#) the number of throughput units to meet usage needs. Azure Stream Analytics is capable of analyzing high volumes of streaming data from many sources. You can scale up Stream Analytics by increasing the number of [streaming units](#) allocated to execute your streaming job.

For general guidance on designing scalable solutions, see the [scalability checklist](#) in the Azure Architecture Center.

Security

Azure Event Hubs secures data through an [authentication and security model](#) based on a combination of Shared Access Signature (SAS) tokens and event publishers. An event publisher defines a virtual endpoint for an event hub. The publisher can only be used to send messages to an event hub. It is not possible to receive messages from a publisher.

For general guidance on designing secure solutions, see the [Azure Security Documentation](#).

Resiliency

For general guidance on designing resilient solutions, see [Designing reliable Azure applications](#).

Deploy the scenario

To deploy this scenario, you can follow this [step-by-step tutorial](#) demonstrating how to manually deploy each component of the scenario. This tutorial also provides a .NET client application to generate sample phone call metadata and send that data to an event hub instance.

Pricing

To explore the cost of running this scenario, all of the services are pre-configured in the cost calculator. To see how the pricing would change for your particular use case, change the appropriate variables to match your expected data volume.

We have provided three sample cost profiles based on amount of traffic you expect to get:

- **Small:** process one million events through one standard streaming unit per month.
- **Medium:** process 100M events through five standard streaming units per month.
- **Large:** process 999 million events through 20 standard streaming units per month.

Related resources

More complex fraud detection scenarios can benefit from a machine learning model. For scenarios built using Machine Learning Server, see [Fraud detection using Machine Learning Server](#). For other solution templates using Machine Learning Server, see [Data science scenarios and solution templates](#). For an example solution using Azure Data Lake Analytics, see [Using Azure Data Lake and R for Fraud Detection](#).