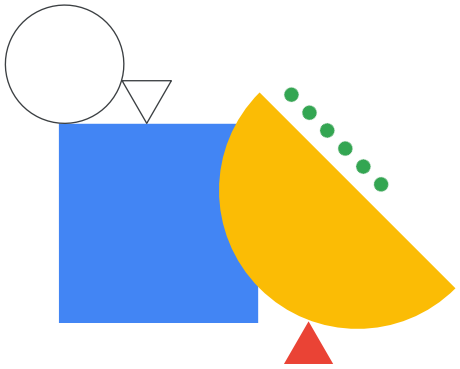
Google Cloud

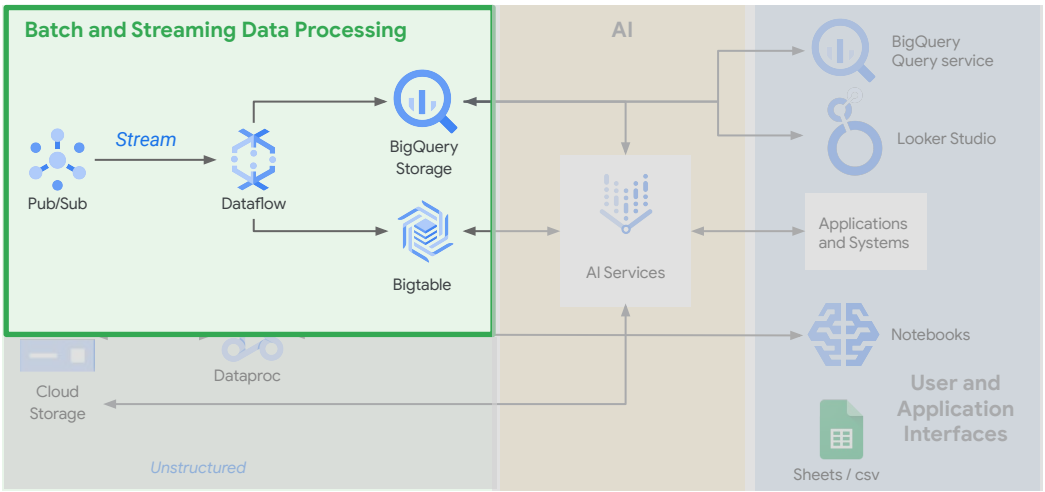
Introduction to Processing Streaming Data



This module discusses what stream processing is, how it fits into a big data architecture, when stream processing makes sense, and also, the challenges associated with streaming data processing.

Proprietary + Confidential

Streaming data processing



Google Cloud

As this module is all about streaming, I'll be discussing that part of the reference architecture.

Data typically comes in through Pub/Sub, then that data goes through aggregation and transformation in Dataflow. Then you use BigQuery or Bigtable depending on whether the objective is to write aggregates or individual records coming in from streaming sources.

Let's look at streaming ideas first. Why do we stream? Streaming enables us to get real-time information in a dashboard or another means to see the state of your organization.

In the case of the New York City Cyber Command, Noam Dorogoyer stated the following: "We have data coming from external vendors, and all this data is ingested through Pub/Sub, and Pub/Sub pushes it through to Dataflow, which can parse or enrich the data,"

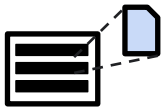
If data comes in late, especially when it comes to cybersecurity, it's no longer valuable, especially during an emergency. So, from a data engineering standpoint, the way we constructed the pipeline is to minimize latency at every single step. If it's maybe a Dataflow job, we designed it so that as many elements as possible are happening in parallel so at no point is there a step that's waiting for a previous one."

The amount of data flowing through the Cyber Command varies each day. Dorogoyer said that on weekdays during peak times, it could be 5 or 6 terabytes. On weekends, that can drop to 2 or 3 terabytes. As the Cyber Command increases visibility across agencies, it will deal with petabytes of data.

Security analysts can access the data in several ways. They run queries in BigQuery or use other tools that will provide visualizations of the data, such as Looker Studio.

Proprietary + Confidential

Streaming is data processing for unbounded data sets



Bounded Data (Batch)

- Finite data set
- Usually complete
- Time of elements is usually disregarded
- Typically at rest
- Held in durable storage

Unbounded Data (Stream)

- Infinite data set
- Never complete
- Time of elements is usually significant
- Typically in motion
- Held in temporary storage

Google Cloud

Streaming is data processing on unbounded data. Bounded data is data at rest. Stream processing is how you deal with unbounded data.

A streaming processing engine provides: low latency, speculative or partial results, the ability to flexibly reason about time, controls for correctness, and the power to perform complex analysis.

Proprietary + Confidential

Stream analytics has many applications

Data integration (10 sec - 10 min)

- Data warehouses become real-time
- Take load off source databases with change data capture (CDC)
- Microservices require databases and caches

Online decisions (100 ms - 10 sec)

- Real-time recommendations
- Fraud detection
- Gaming events
- Finance back office apps

Google Cloud

You can actually use streaming to get real-time data warehouses and then create a dashboard of real-time information. For example, you could see in real-time the positive versus negative tweets about your company's product, use it to detect fraud, use for gaming events, or for finance back office apps, such as stock trading, anything dealing with markets, etc.

Proprietary + Confidential

Big Data challenges

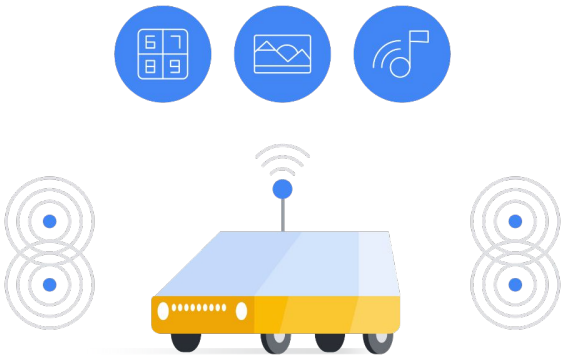
Variety | Volume | Velocity | Veracity

Google Cloud

So, when you look at the challenges associated with streaming applications, you're talking about the 4 V's, variety, volume, velocity, and veracity.

Proprietary + Confidential

1 | Variety



Sensors on roads around the world

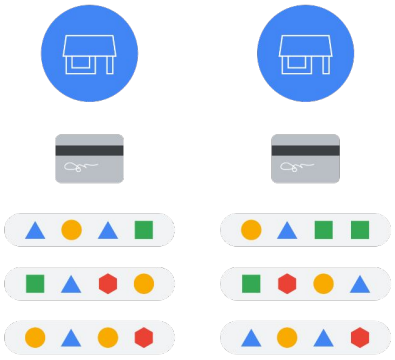
Data could come in from a variety of different sources and in various formats.

Google Cloud

First, data could come in from a **variety** of different sources and in various formats. Imagine hundreds of thousands of sensors for self-driving cars on roads around the world. The data is returned in various formats such as number, image, or even audio.

Proprietary + Confidential

1 | Variety



Point-of-sale data

How do we alert downstream systems of new transactions in an organized way with no duplicates?

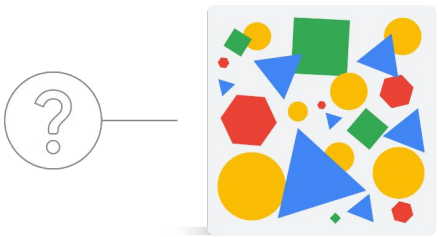
Google Cloud

Now consider point-of-sale data from a thousand different stores. How do we alert our downstream systems of new transactions in an organized way with no duplicates?

Proprietary + Confidential

2 | Volume

Will the pipeline code and infrastructure scale, or will it grind to a halt or even crash?



Volume of data

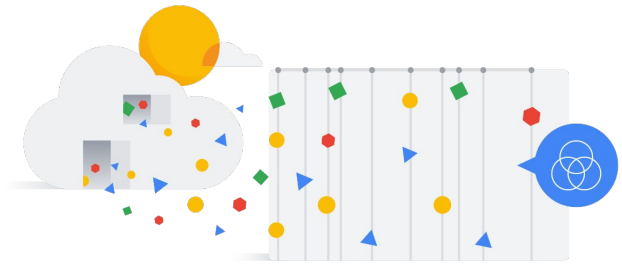
Google Cloud

Next, let's increase the magnitude of the challenge to handle not only an arbitrary variety of input sources, but a **volume** of data that varies from gigabytes to petabytes.

You'll need to know whether your pipeline code and infrastructure can scale with those changes or whether it will grind to a halt or even crash.

Proprietary + Confidential

3 | Velocity



Data often needs to be processed in near-real time, as soon as it reaches the system.

Google Cloud

The third challenge concerns **velocity**. Data often needs to be processed in near-real time, as soon as it reaches the system.

Proprietary + Confidential

3 | Velocity

Need a way to handle data that:

- Arrives late
- Has bad data in the message
- Needs to be transformed

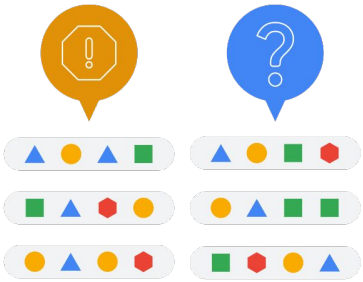
Google Cloud

You'll probably also need a way to handle data that arrives late, has bad data in the message, or needs to be transformed mid-flight before it is streamed into a data warehouse.

Proprietary + Confidential

4 | Veracity

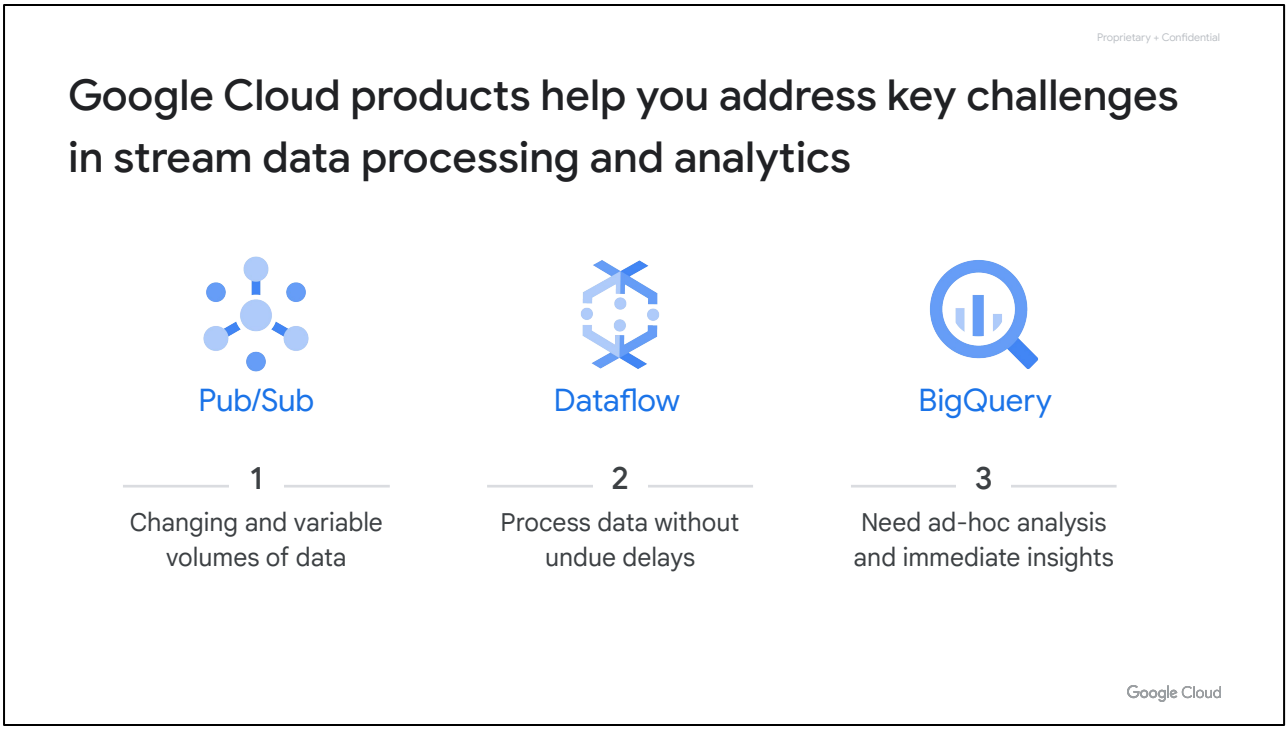
Gathered data might come in with inconsistencies and uncertainties due different data types and sources.



Google Cloud

And the fourth major challenge is **veracity**, which refers to the data quality. Because big data involves a multitude of data dimensions resulting from different data types and sources, there's a possibility that gathered data will come with some inconsistencies and uncertainties.

Challenges like these are common considerations for pipeline developers.

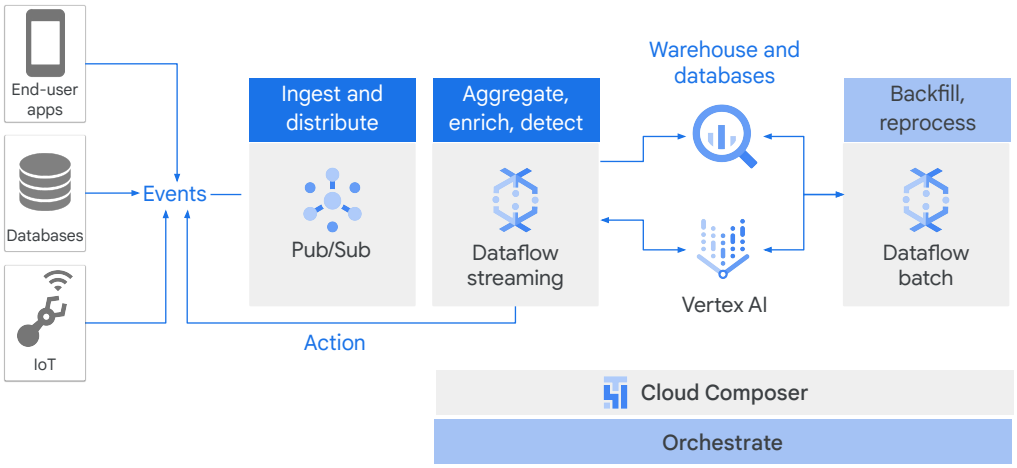


The three products you are going to examine here are:

- **Pub/Sub**, which will allow you to handle changing and variable volumes of data,
- **Dataflow**, which can assist in processing data without undue delays,
- and **BigQuery**, which you will use for your ad-hoc reporting, even on streaming data.

Proprietary + Confidential

Stream analytics includes some common steps



Google Cloud

Let's take a look at the steps that happen.

- First, some sort of data is coming in, possibly from an app, a database, or an Internet of Things, or IoT. These are generating events.
- Then, an action takes place. You are going to ingest those and distribute those with Pub/Sub. This will ensure that the messages are reliable. This will give you buffering. Dataflow, then is what aggregates, enriches, and detects the data.
- Next, you will write into a database of some kind, such as BigQuery or Bigtable, or maybe run things through a Machine Learning model. For example, you might use this streaming data as it is coming in to train a model in Vertex AI.
- Then, finally, Dataflow or Dataproc could be used for batch processing, backfilling, etc.

So, this is a pretty common way to put things together in Google Cloud.