# Different Ways Of Ingesting Data & the Challenges Involved

In this lesson, we will discuss the different ways in which we can ingest the data. Also, we will cover the challenges involved in this process.

**We'll cover the following**

## Different Ways To Ingest Data

There are two primary ways to ingest data, in *Real-time* & in *Batches* which run at regular intervals. Which one to pick of the two entirely depends on the business requirements.

Data Ingestion in real-time is typically preferred in systems reading medical data like a heartbeat, blood pressure via wearable IoT sensors where the time is of critical importance. Also, in systems handling financial data like stock market events etc. These are a few instances where time, lives & money are closely linked & we need information as soon as we can get.

On the contrary, in systems that read trends over time, we can always ingest data in batches. For instance, when estimating the popularity of a sport in a region over a period of time.

Let's talk about some of the challenges which developers have to face when ingesting massive amounts of data. I have added this lesson just to give you a deeper insight into the entire process. In the upcoming lesson, I also talk about the general use-cases of data streaming in the application development domain.

domain.

## Challenges with Data Ingestion #

### Slow Process #

Data ingestion is a slow process. Why? I've brought up this before. When the data is streamed from several different sources into the system, data coming from each & every different source has a different format, different syntax, attached metadata. The data as a whole is heterogeneous. It has to be transformed into a common format like *JSON* or something to be understood well by the analytics system.

The conversion of data is a tedious process. It takes a lot of computing resources & time. Flowing data has to be staged at several stages in the pipeline, processed & then moved ahead.

Also, at each & every stage data has to be authenticated & verified to meet the organization's security standards. With the traditional data cleansing processes, it takes weeks if not months to get useful information on hand. Traditional data ingestion systems like *ETL* ain't that effective anymore.

**Okay!! But you just said data can be ingested in real-time Right? So, how is it slow?**

Two things, I would like to bring up here, *first* the modern data processing tech & frameworks are continually evolving to beat the limitations of the legacy, traditional data processing systems. Real-time data ingestion wasn't even possible with the traditional systems.

*Second*, analytics information obtained from real-time processing is not that accurate & holistic since the analytics continually runs on a limited set of data as it streams as opposed to the batch processing approach which takes into account the entire data set. So, it's basically the more time we spend studying the data the more accurate results we get.

You'll learn more about this when we go through the *Lambda* and the *Kappa* architectures of data processing.

### Complex & Expensive #

The entire data flow process is resource-intensive. A lot of heavy lifting has to be done to prepare the data before being ingested into the system. Also, it isn't a side process, a dedicated team is required to pull off something like that.

Engineering teams often come across scenarios where the tools & frameworks available in the market fail to serve their needs & they have no option other than to write a custom solution from the bare bones.

*Gobblin* is a data ingestion tool by LinkedIn. At one point in time, LinkedIn had 15 data ingestion pipelines running which created several data management challenges. To tackle this problem, LinkedIn wrote Gobblin in-house.

It is a part of the Apache Software Foundation now. This is a good read

The semantics of the data coming from externals sources changes sometimes as they are not always under our control, which then requires a change in the backend data processing code. Today the IoT machines in the industry are continually evolving at a rapid pace.

These are the factors we have to keep in mind when setting up a data processing & analytics system.

## Moving Data Around Is Risky #

When data is moved around it opens up the possibility of a breach. Moving data is vulnerable. It goes through several different staging areas & the engineering teams have to put in additional effort and resources to ensure their system meets the security standards at all times.

These are some of the challenges which developers face when working with streaming data.