Never Drop.
Never Block.

Drop and Block are not good paradigms for machine data pipelines.

When it comes to IT observability and monitoring, machine data pipelines form the nerves if your observability platform is the nerve center.

They create channels through which machine data flows from the sources that generate them to the analytics and monitoring platforms that act on them.

Multiple things often challenge machine data pipelines. They need to deal with complexities at the data source and target destinations while managing their scale and reliability. A machine data pipeline inadequately sized for storage and elasticity leads to one critical and irreparable failure - data loss.

Never accept data loss.

Never block

Machine data sources often witness sudden surges in data generation for a variety of reasons. A technical error or an error in system code, applications or services seeing a sudden spike in usage, a large number of repeated authentication failures triggered by a malicious attacker could all generate a ton of additional logs. A machine data pipeline needs to be elastic to support these scenarios. Blocking or clogging data sources because the pipeline lacks adequate storage or cannot scale up elastically leads to data loss.

Most data sources do not have infinite storage. When data is blocked and begins to pile up at the source, the sources start rotating pending data to prevent running out of disk space. This data rotation can lead to a loss of critical data that could hold information about an application failure, an in-progress security attack, or a performance issue. Blocking the sender should never be an acceptable feature or architecture for data pipeline platforms unless in extreme circumstances where the data center itself goes offline.

Never drop

When forwarding data to target systems, it is quite common to witness network partitions or services going offline at the target, causing the machine data pipeline to have data coming in but no place to send the data out. The emerging trend, and unfortunate reality, among DIY data pipelines and pipeline management software is dropping data when target systems go down. Enterprises that rely on such pipelines to ship critical machine data related to production system performance, security, and compliance should never accept data drop as a pipeline management norm. Dropped data is recoverable if your pipelines are architected for it. If not, data recovery will require professional assistance and costs time and resources that a business could be using elsewhere.

There are two ways to prevent data blocks and drops - infinite storage and an elastic data pipeline architecture.

Infinite storage

A solid way to address pipeline source and destination mismatch is to have an infinite storage layer. Limitless storage has been ineffective due to limitations of the traditional disk-based storage design for on-prem and cloud systems. However, LogFlow's InstaStore breaks free from these limitations. Being built primarily on object storage and not treating object storage as a tiered overflow device, InstaStore provides instant elasticity for storage requirements with ZeroStorageTax. Businesses no longer need to expand storage, throttle senders or scramble to handle higher throughput needs. InstaStore is infinite and permanently embedded within the LogFlow system.

Elastic architecture

An elastic design ensures that data pipelines can handle data sources sending more-than-usual amounts of data without manual intervention. Ignoring elasticity while building data pipelines will lead to data backlogs at the source and data loss when the backlog continuously builds up.

Never block, never drop with InstaStore

We built InstaStore to handle the challenges faced by enterprises in high-volume environments. LogFlow writes 100% of all ingested data into InstaStore before forwarding it. InstaStore provides an infinite storage layer by abstracting storage as an API and building on top of any object-store.

Build your data pipelines from day 0 with infinite storage that can be an endless store for throughput mismatches on either the source or the target. You can instantly replay any data in InstaStore to a target on-demand and in real-time. Never block or never drop data with InstaStore.

LogFlow's Kubernetes-backed elastic architecture

Apica's LogFlow is built on Kubernetes and works with Cluster Autoscaling and Horizontal Pod Autoscaling, providing instant throughput on-demand in high-volume data environments.

Let's summarize.
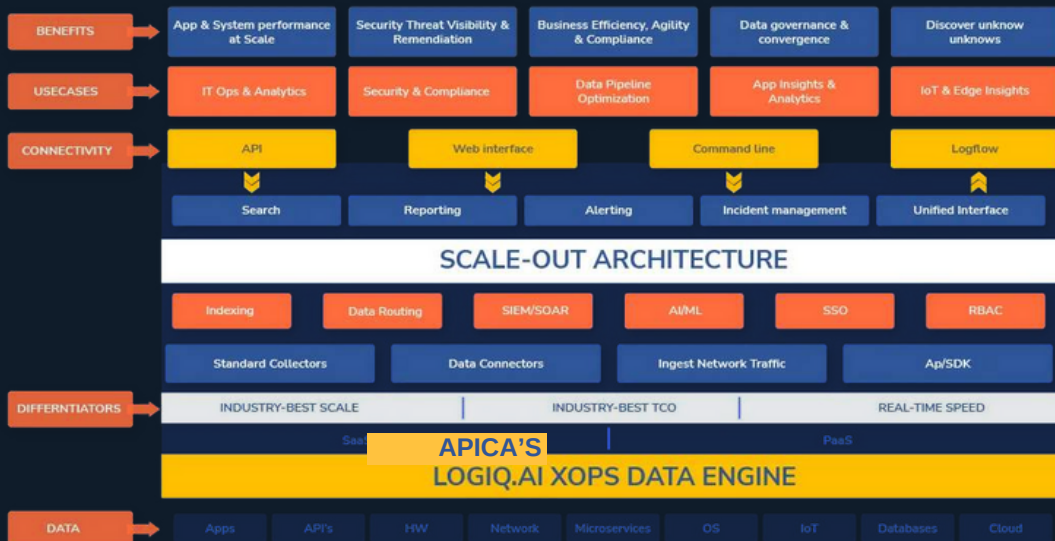
If you'd like to know more about how Apica's LogFlow and InstaStore can accelerate business transformation through unmatched data pipeline control and total observability, do reach out.

www.apica.io