**White Paper –**

Classification: Limited

**Google Cloud Event driven streaming using confluent Kafka**

**What is the Context**

The leading bank in UK wanted to shift from on-premises infrastructure to cloud based infrastructure in order adapt to the customer base scaling exponentially and reduce the operational overhead in on prem infrastructure maintenance.

The programme started in 2019 with a vision to implement streaming applications on Google/hybrid cloud.

**What was the business problem faced by the client:**

The customer has the vision for real time Streaming to make data sources available both on-premises and in GCP, at the same time for:

* Removing latency impacts and cost issues and giving the Bank a ‘360 sync capability’ so it can provide e2e real-time services on-premises, on GCP and a hybrid depending on where the data sources and/or consumers reside.
* Also, the client needed to deepen relationships with its customers (both existing and prospective) and a mechanism to tailor services for their needs on an ‘event-driven’ basis

If this is not addressed, there shall be more:

* Operational overheads in maintaining the infrastructure as it the customer base has grown exponentially in last few years with more acquisitions.
* The problems of scalability and latency involved in providing solution to customers needed to be addressed.
* Customers may consider changing the bank where better services are available. Hence the client wanted to move to more reliable and scalable cloud solution as opposed to current on-prem infrastructure.

**What led to the Solution**

To resolve the problem, we have deployed the confluent Kafka on GCP, which provides the capability to start ingesting feeds from on-prem Kafka cluster to confluent Kafka cluster deployed on GKE using confluent replicator.

The programme has been able to stream (Customer data) and (Transactions) data into the Google cloud environments.

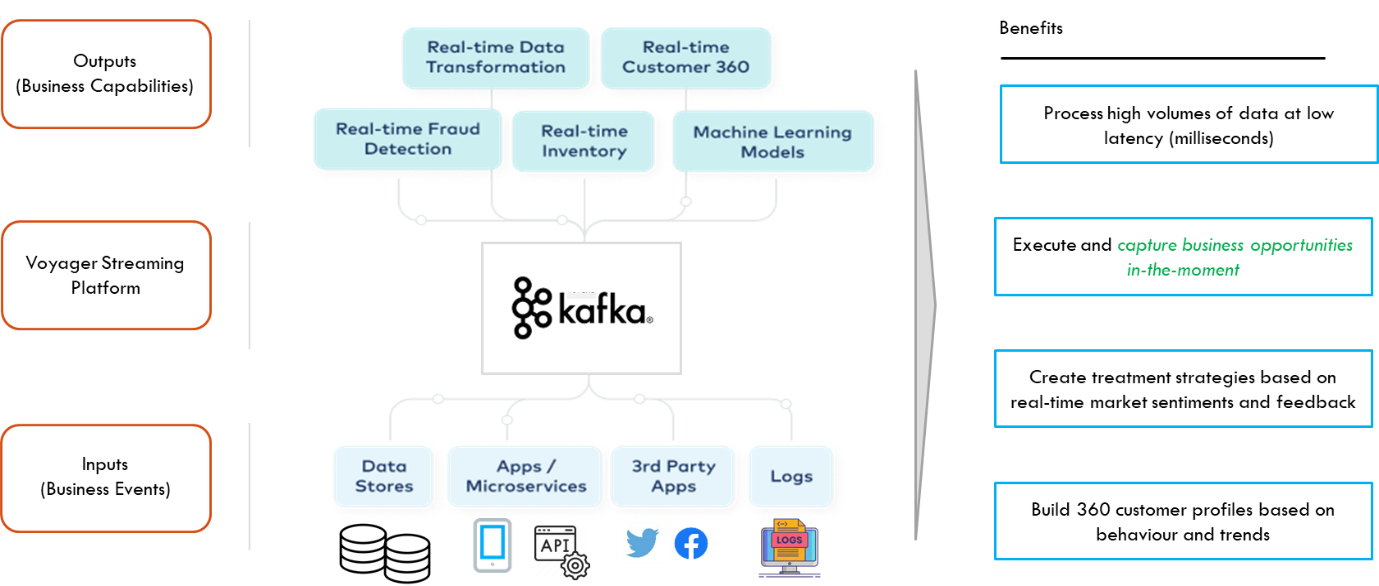
The capability will support several strategic deliveries across the Bank such as:

o   **Transaction Classifications:** Real-time classifications of transactions in <1 second– which allow client to release significant business benefits from personalisation, targeted marketing, analytics at scale etc.

o   **Synchronisation of Customer data and System of records:** Customer changes synchronised to the Customer System of Record (CSOR) within milliseconds.

o   **New Payments Architecture (NPA):**Standardise payment messaging across the globe in line with ISO20022 as we integrate with Form3 (new payments platform)

o   **Various business value propositions –**Removing operational costs and unlocking value by undertake insights at scale, leveraging AI and ML capabilities where appropriate



Kafka Streaming platform

This used **event-based streaming** which is at the core of the Future Bank strategy to enable customer to develop closer relationships with end customers and support their everyday needs by becoming a partner for life

The Programme created the first ‘**shared business service’** pattern on GCP in the Bank – as the first adopters in the Bank

**Technology Stack for the Program:**

As a part of program, we deployed the CFK on Kubernetes cluster on Google cloud platform.

* **Kubernetes**:

Kubernetes is an open-source container orchestration platform that automates many of the manual processes involved in deploying, managing, and scaling containerized applications

* **Confluent for Kubernetes (CFK)**

Confluent for Kubernetes is a cloud-native control plane for deploying and managing Confluent in your private cloud environment. It provides standard and simple interface to customize, deploy, and manage Confluent Platform through declarative API.

CFK runs on Kubernetes, the runtime for private cloud architectures.

Below is the high-level diagram of CFK components deployed on Kubernetes cluster on GCP.

Diagram

Description automatically generated

Main Components of CFK:

* **Control Center:** Control Center helps users manage and monitor Apache Kafka with a friendly dashboard and admin tools We can use control center to monitor brokers, topics, Schema Registry, connectors configurations and to optimize cluster performance.
* **Replicator**: Replicator allows to replicate data from one cluster to another cluster. It also enables automated client to fail over, so producers and consumers automatically fail over to the recovery site upon the failure. So, it provides disaster recovery protection to avoid data loss.
* **Schema Registry**: Kafka does not make data verification. It just takes bytes from input and write them on output. So, if producer sends bad Data, then consumer break. The Schema Registry provides data compatibility. It is a component which rejects bad data.
* **Confluent Connectors:** Confluent Connect is for connecting Kafka with external systems such as databases, file systems etc using so-called Connectors.
* Connectors are ready-to-use components, which helps us to import data from external systems into Kafka topics and export data from Kafka topics into external systems. We can use existing connector implementations for common data sources and sinks.
* **Rest Proxy:** Provides access to Apache Kafka via HTTP protocol. So, we do not need to use native Kafka protocol to produce messages, consume messages.
* **Google cloud platform:**

Google Cloud Platform (GCP) is a suite of cloud computing services provided by Google. It is a public cloud computing platform consisting of a variety of services like compute, storage, networking, application development, Big Data, and more.

**Benefits for the Program**

Below listed are the high-level business benefits achieved out of this transformation idea.

* Highly scalable, robust, and portable **hybrid-cloud** streaming capability
* It will provide resiliency and flexibility from a **Regulator perspective** – client have the capability to fallback to an on-premises streaming service in the event of a pivot away from Public Cloud.
* Reduced cost to delivery for the customer.
* Synchronization of Customer data and System of records with average speed is **0.5s on on-prem.**
* Cloud migration of our on-prem legacy datastores will support **> 4 petabytes (PB)** of data from on-prem to GCP.
* More critically, it will be enabled client to keep pace with the likes of **Atom, Manzo, Stripe** etc.
* It will prove that the client can process more than 250,000 real time transactions per second (tps) in one pipeline which is 365% higher than peak transaction traffic, client’s busiest platform, equivalent to **c1 billion transactions per hour.**