

# **Kafka**

**Kafka and its tools**

**Shivam jaiswal**

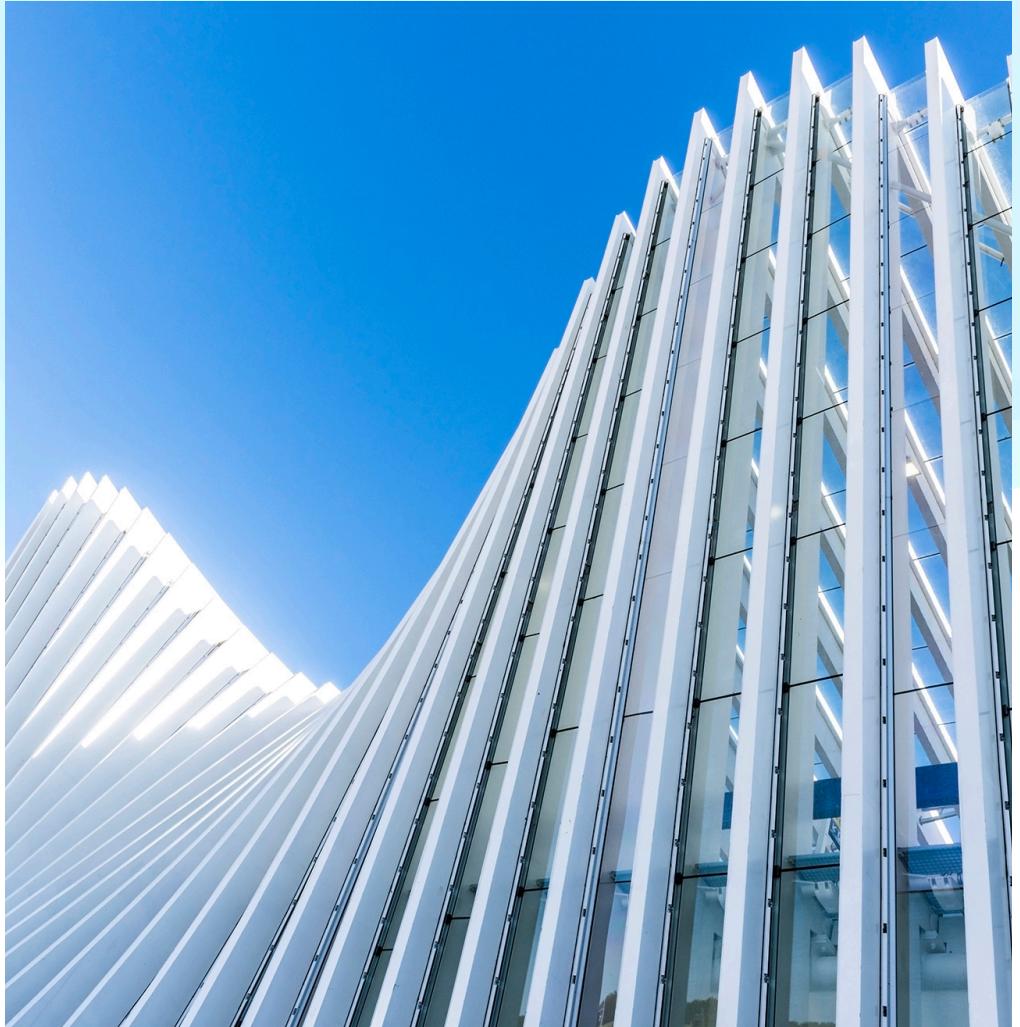
# Topics to Cover

## Overview

- Kafka basics
- Kafka Client API
- Kafka connect
- Kafka streams
- Kafka monitoring
- Schema registry and kSQL

# What is Kafka?

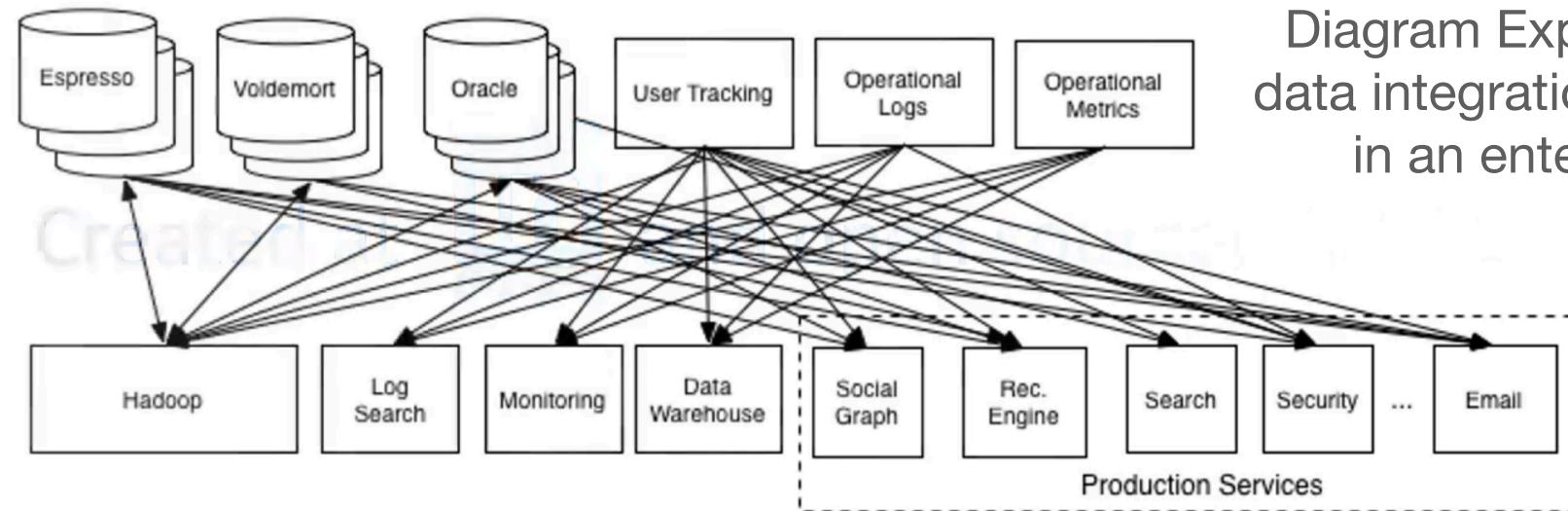
Kafka is one of the most popular **data streaming processing platforms**.



# Kafka History

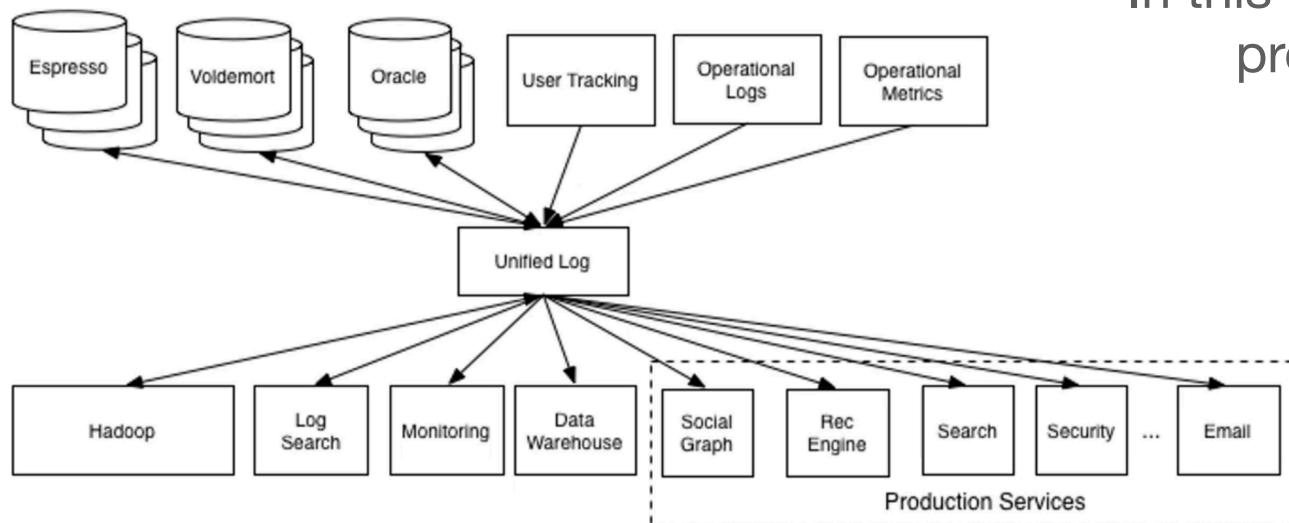
Created at LinkedIn and open sourced in 2011

Kafka was designed to handle the data integration problem.

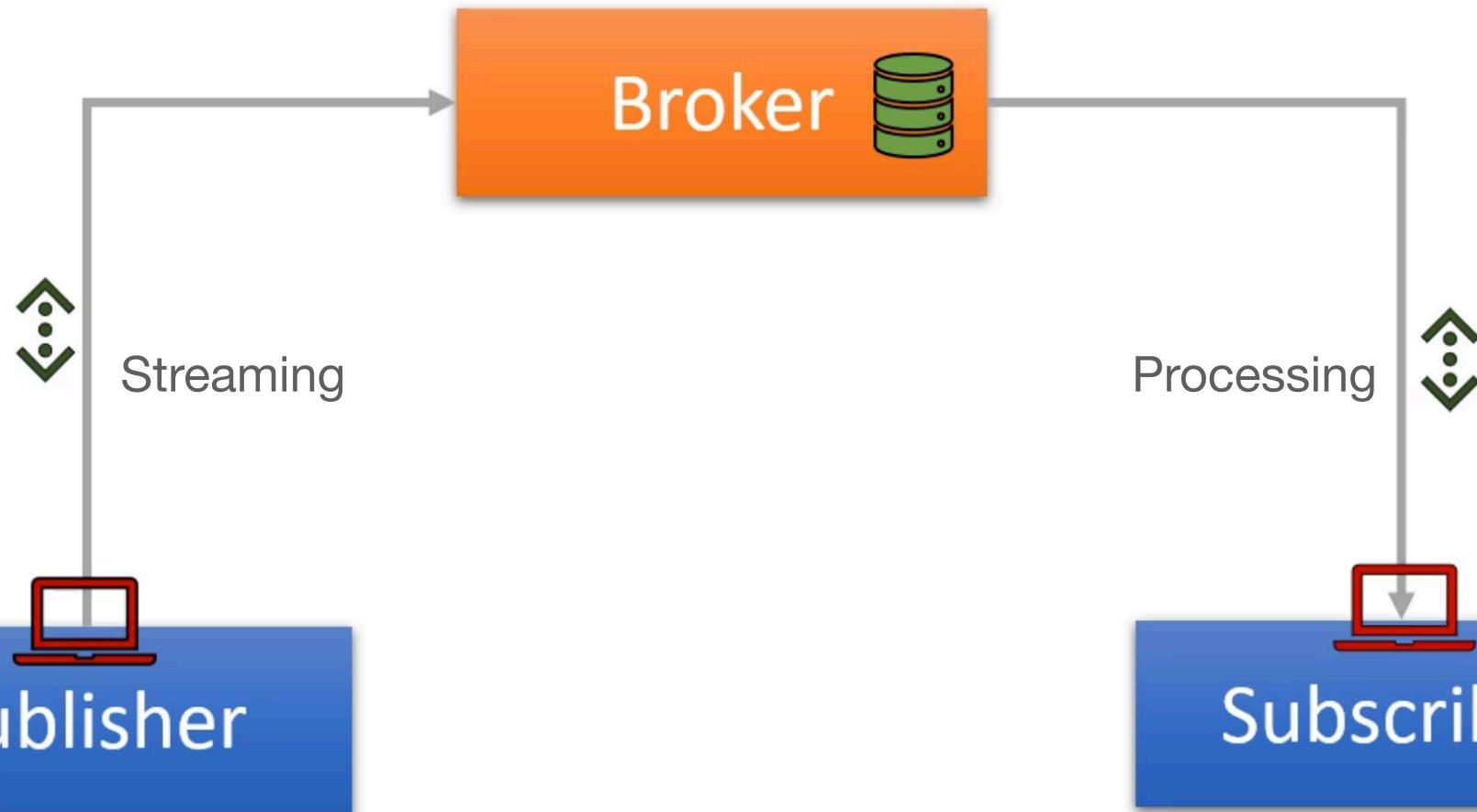


# Kafka History

Kakfa works as middleware  
Data producer and consumers  
Connects to a centralised system  
In this way data integration  
problem resolved.



# Message Broker



Message Producer

Message Consumer

# Kafka Evolution

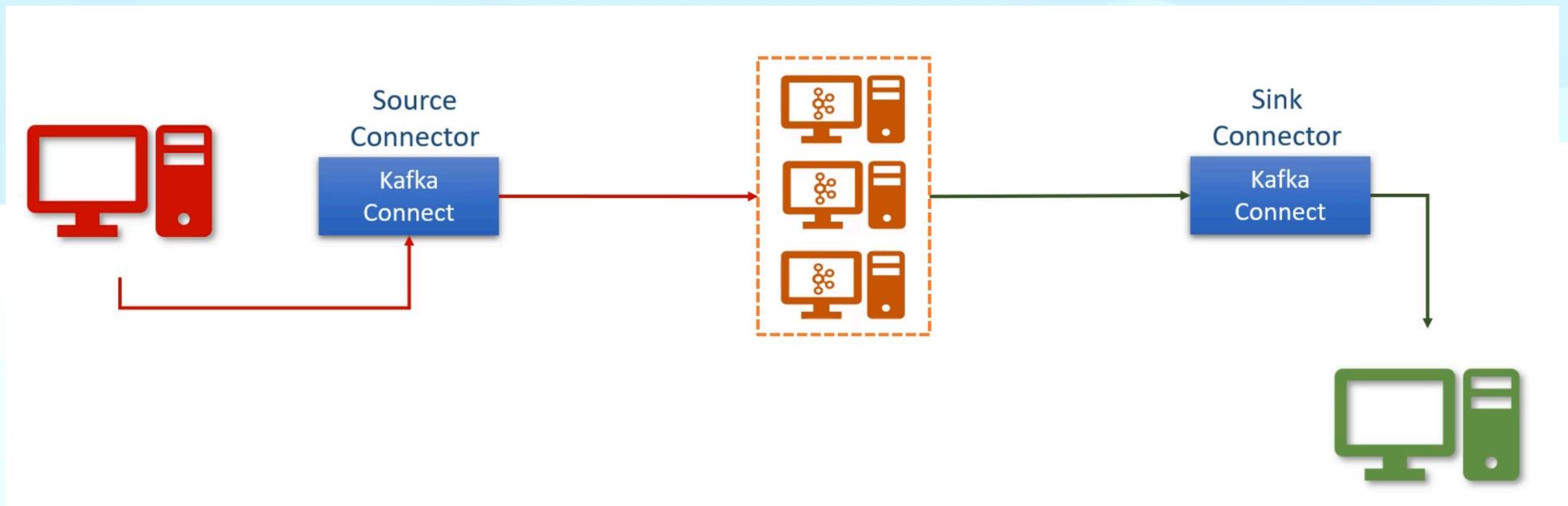
- Kafka Broker : center server system
- Kafka Clients : Producer and consumer api library
- Kafka Connect : It address the initial data-integration problem.
- Kafka Streams : Another library for creating realtime streaming processing applications.
- KSQL : With this Kafka is aiming to be a realtime database

# Kafka Core Concepts

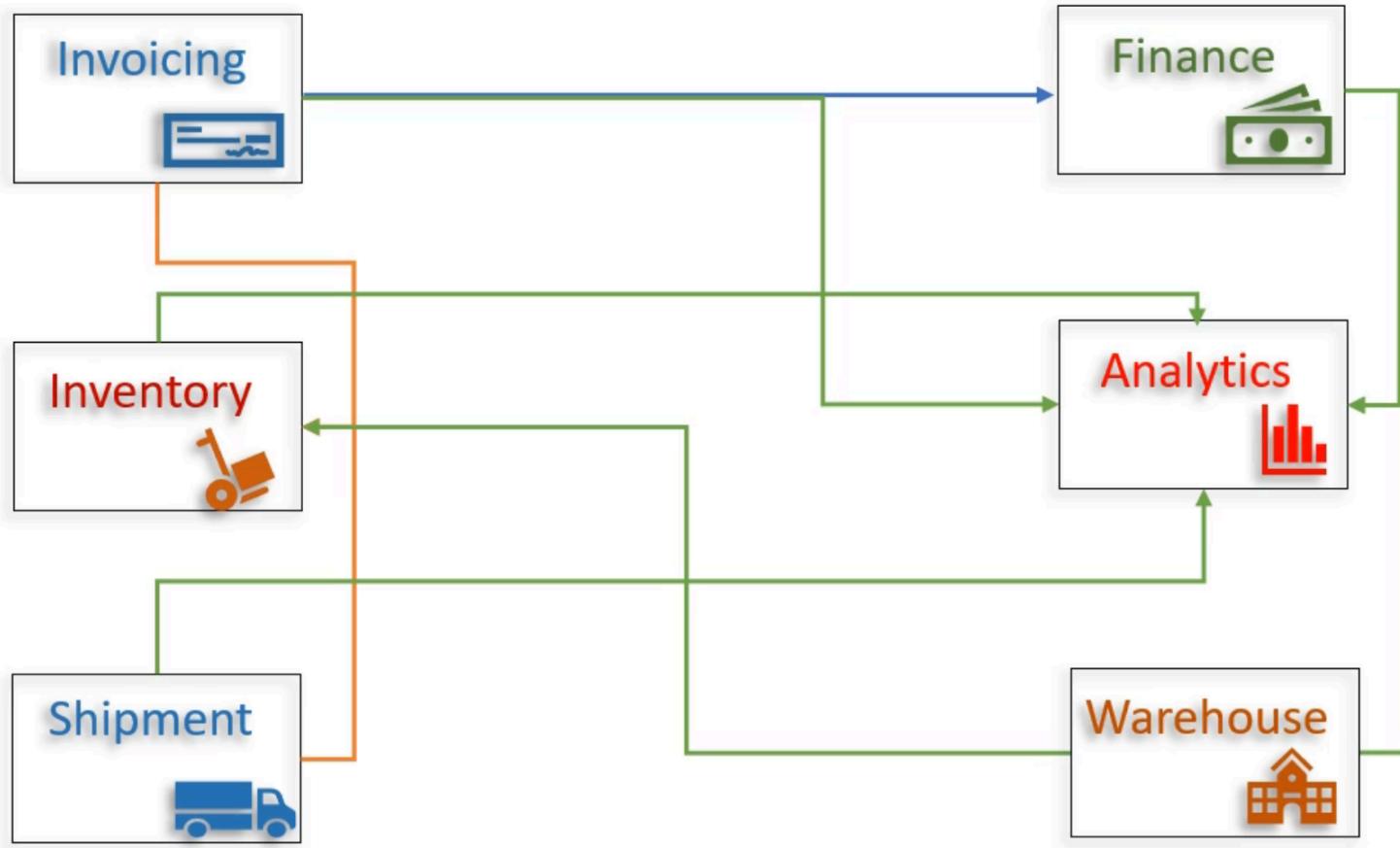
- Producers
- Consumers
- Broker : a server
- Cluster : set of brokers
- Topics : to resolve data segregation problem.
- Partitions : to resolve the storage capacity problems,scalability.
- Offset :
- Consumer Group

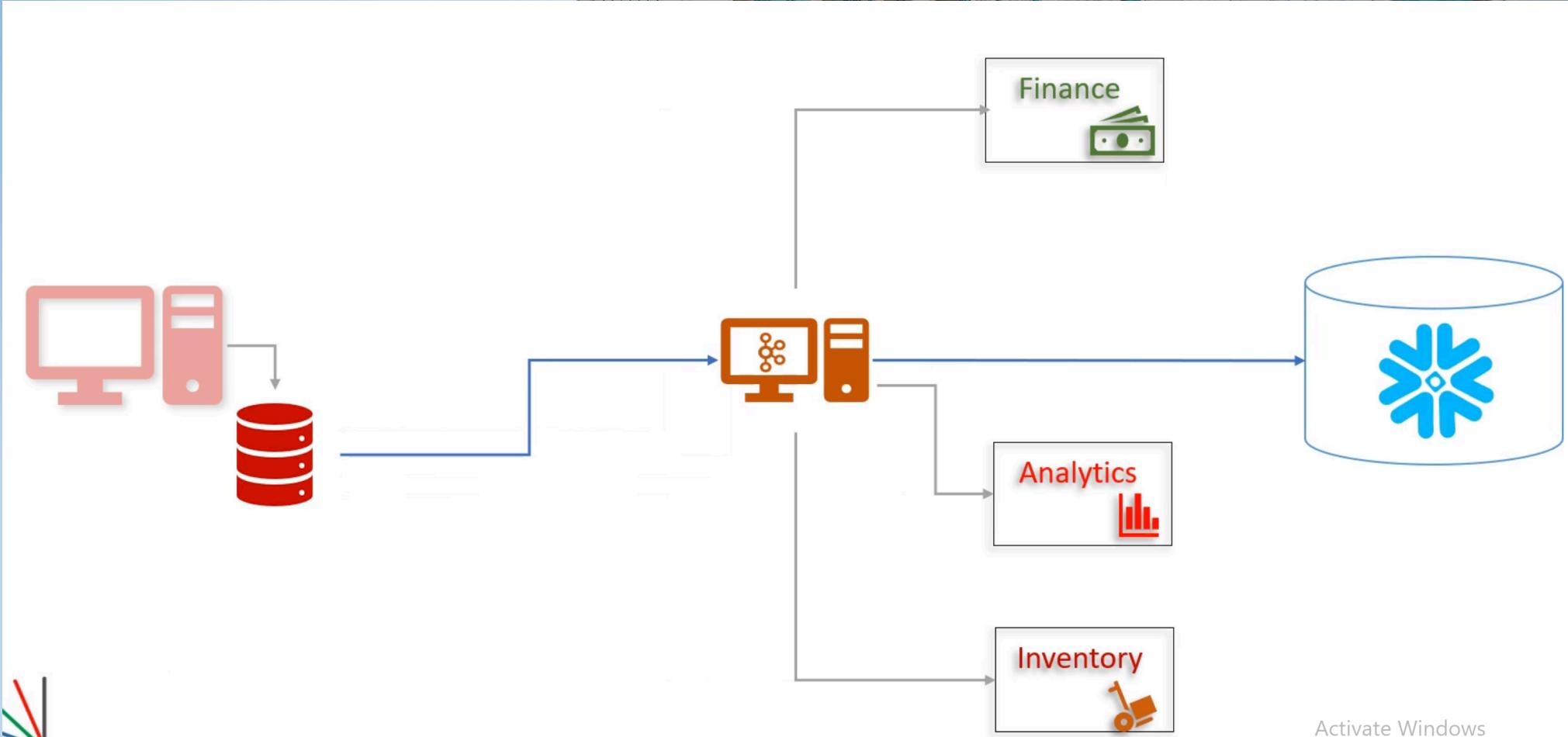
# Kafka Connect Core Concepts

- Kafka connect is component of Kafka , for connecting and moving data between Kafka and external systems.

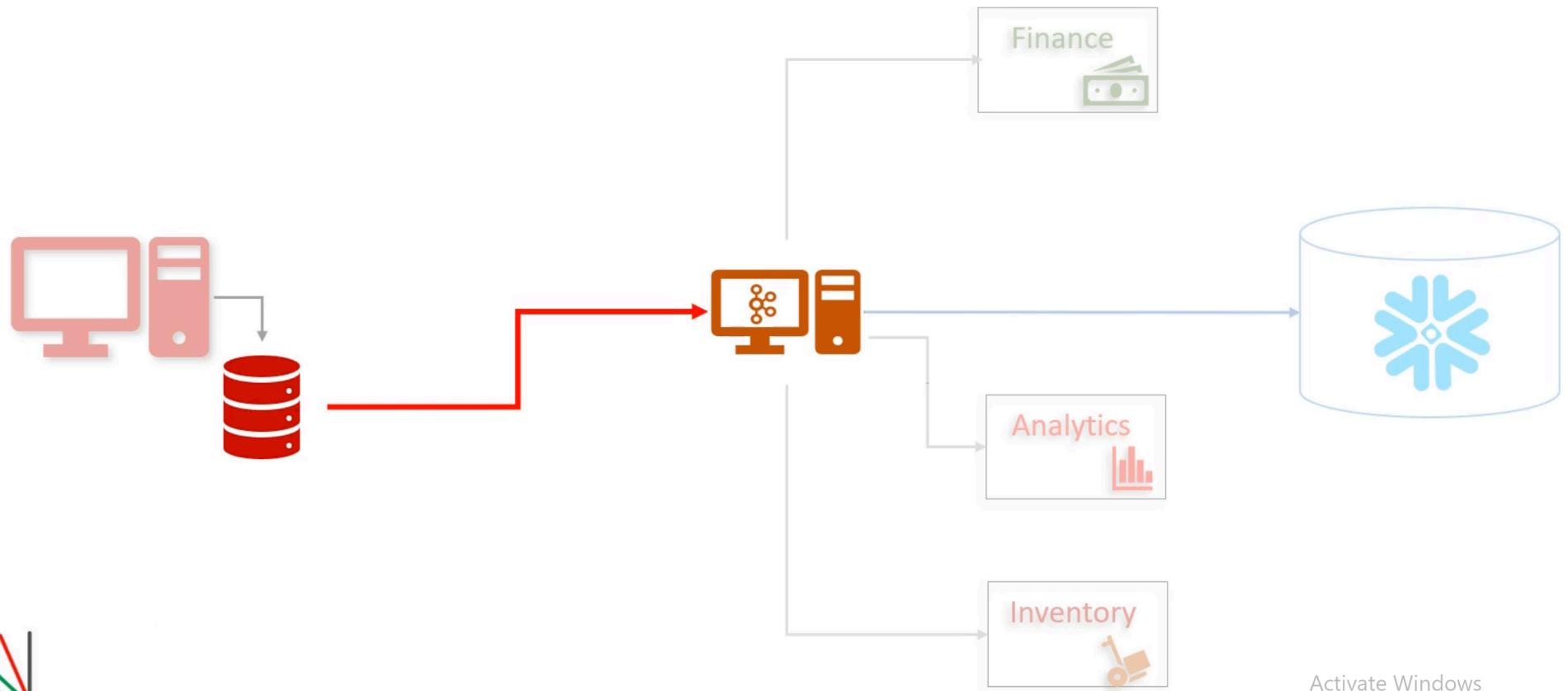


# Data Integration Problem



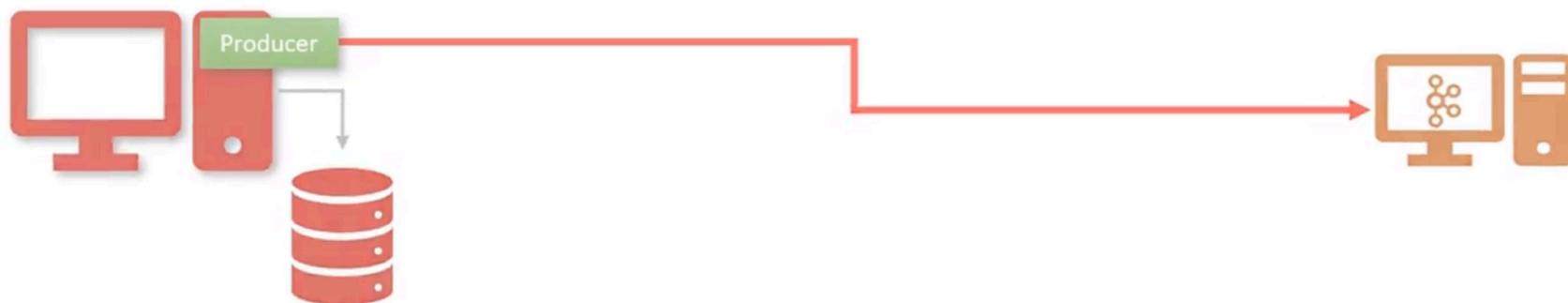


## But how we will bring data from service to Kafka cluster.



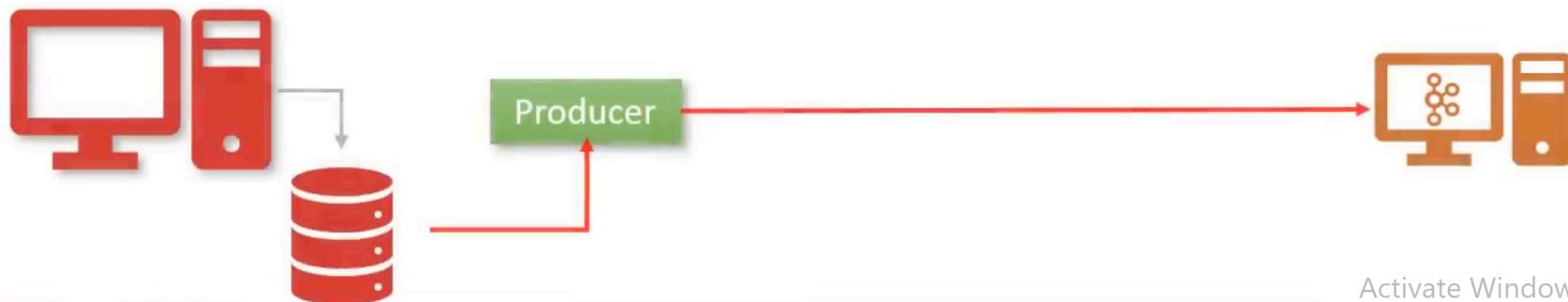
Activate Windows

## By creating a producer?



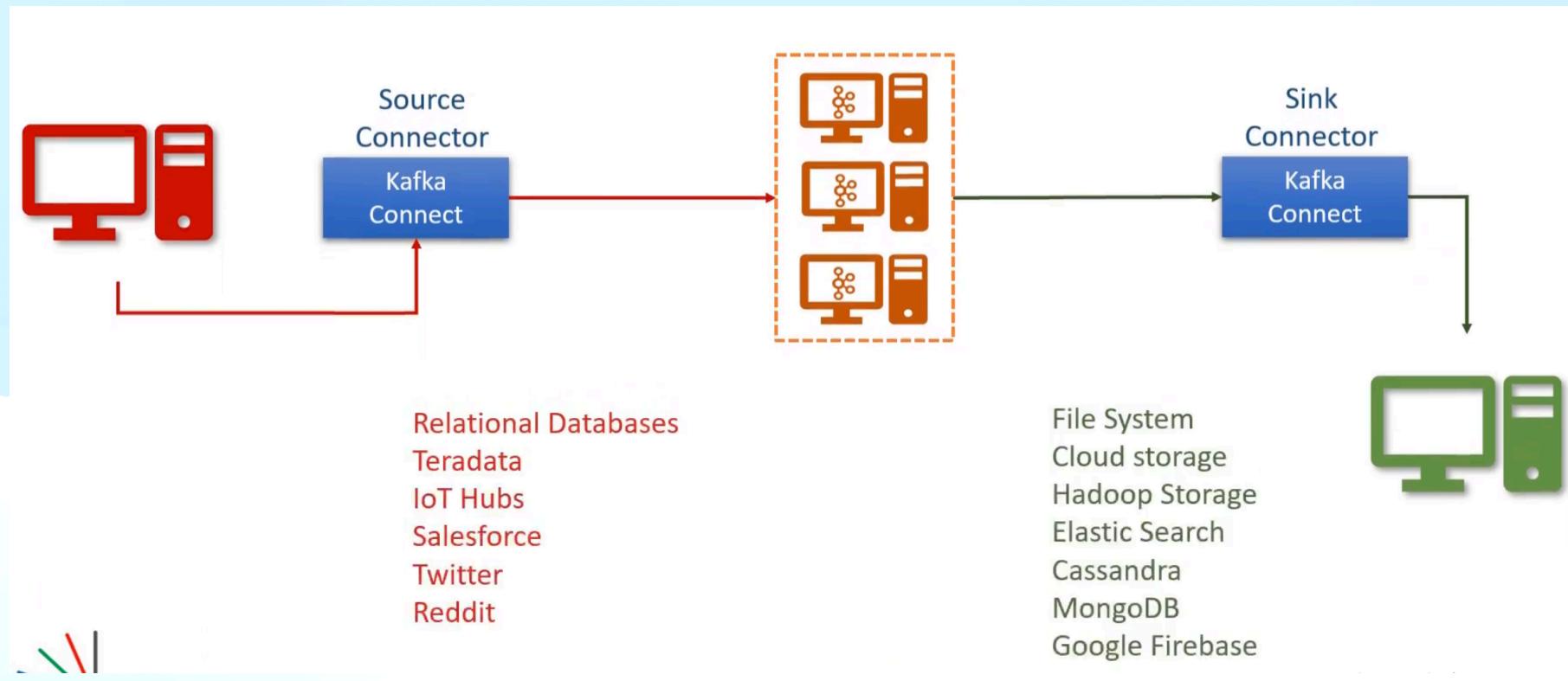
If we have Source code we will embedded our producer code in it.

But what if we don't have ??



Activate Windows

**Then we will be required to create an independent producer. Which will read data from source and send to kafkacluster.**



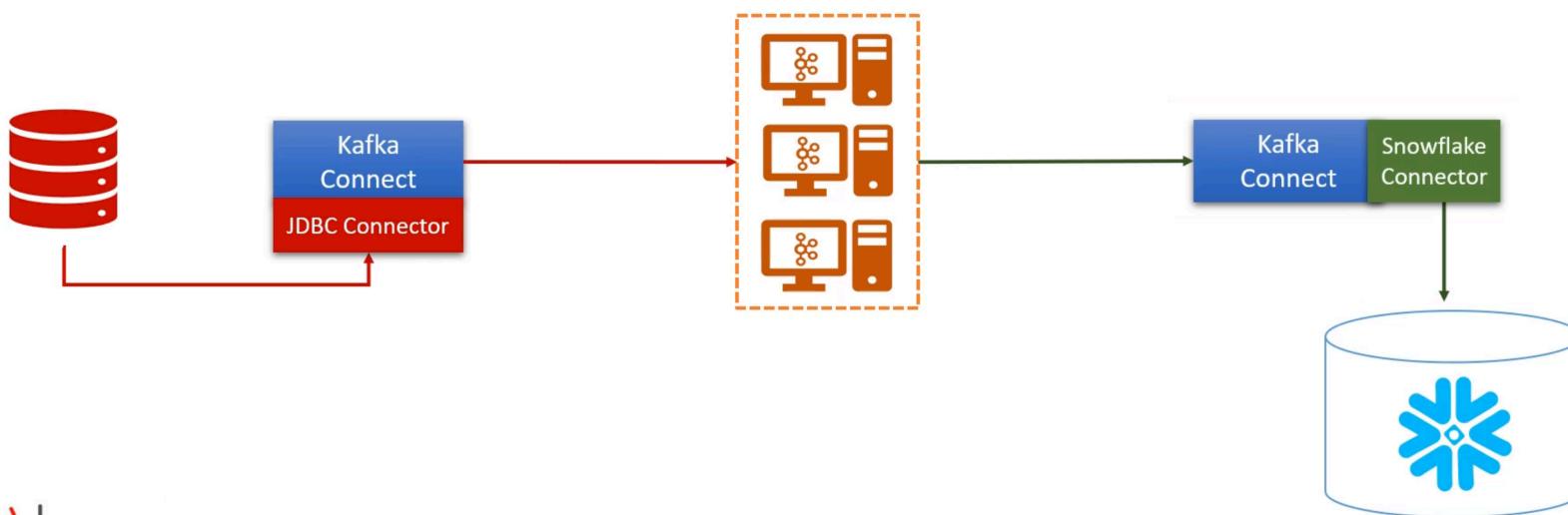
**Rather than creating our own independent producer we can use kafka connect and configure it .**

# We can create our own connector.

## Kafka Connect Framework

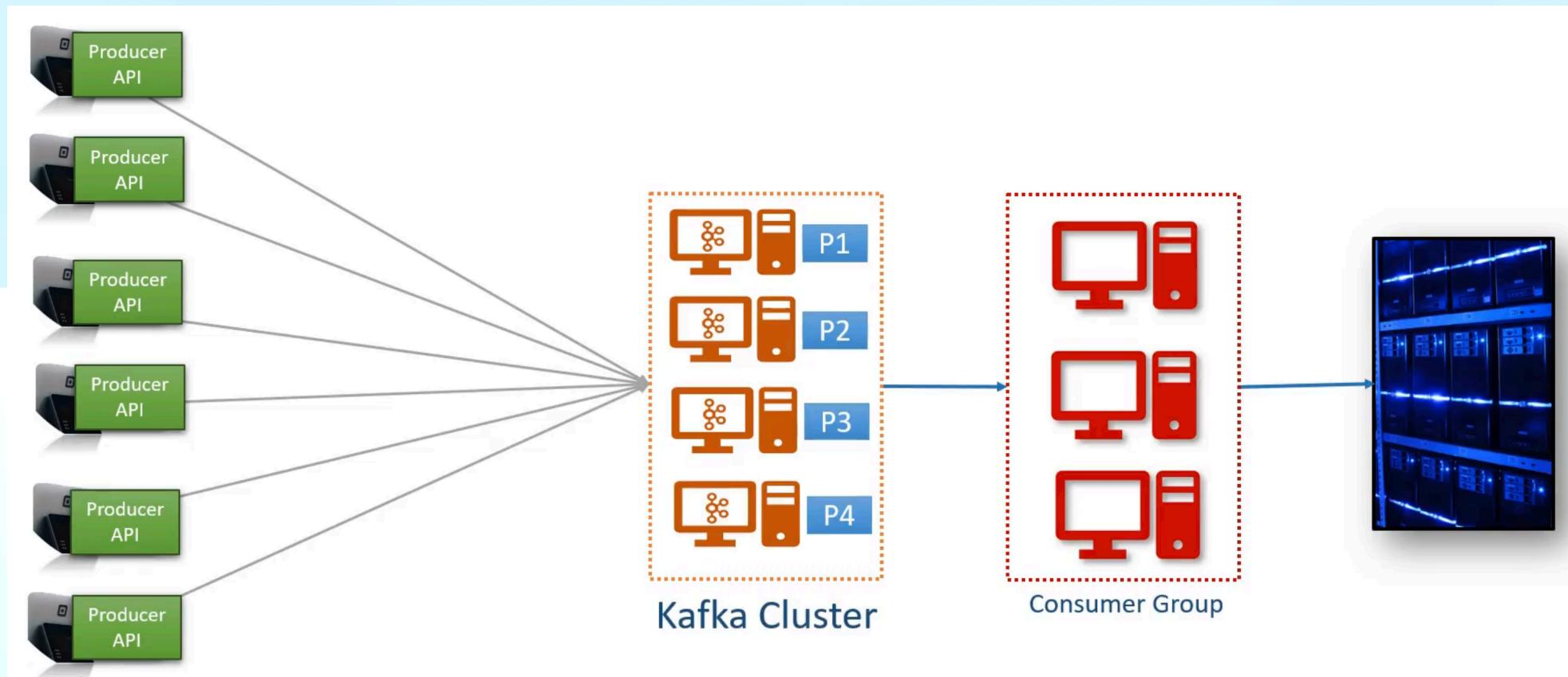
1. Source Connector
  - i. SourceConnector
  - ii. SourceTask
2. Sink Connector
  - i. SinkConnector
  - ii. SinkTask

Connect framework takes care of everything I.e., fault tolerance, error handling ,scalability and all heavy things.  
We only need to implement the Connector and task classes.



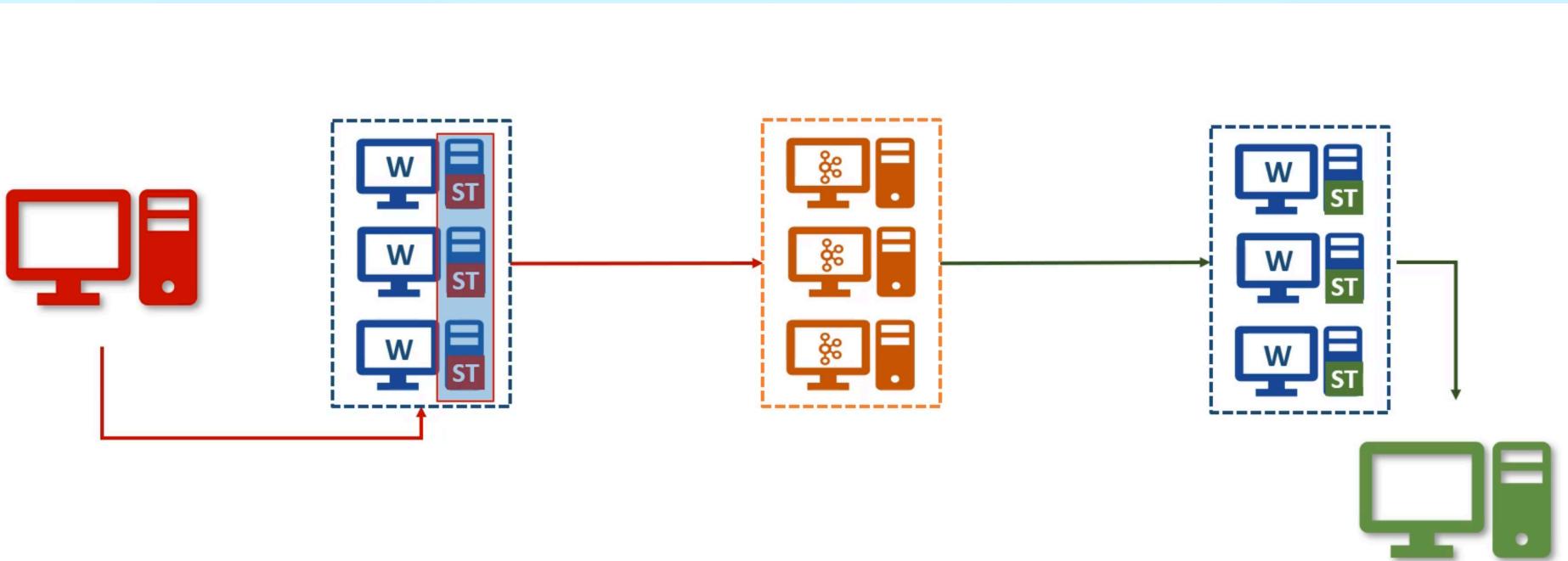
# Scalability

We increases producer api, brokers ,consumers etc 😊.



# Kafka Connect : Scalability

Kafka connect itself is a cluster.

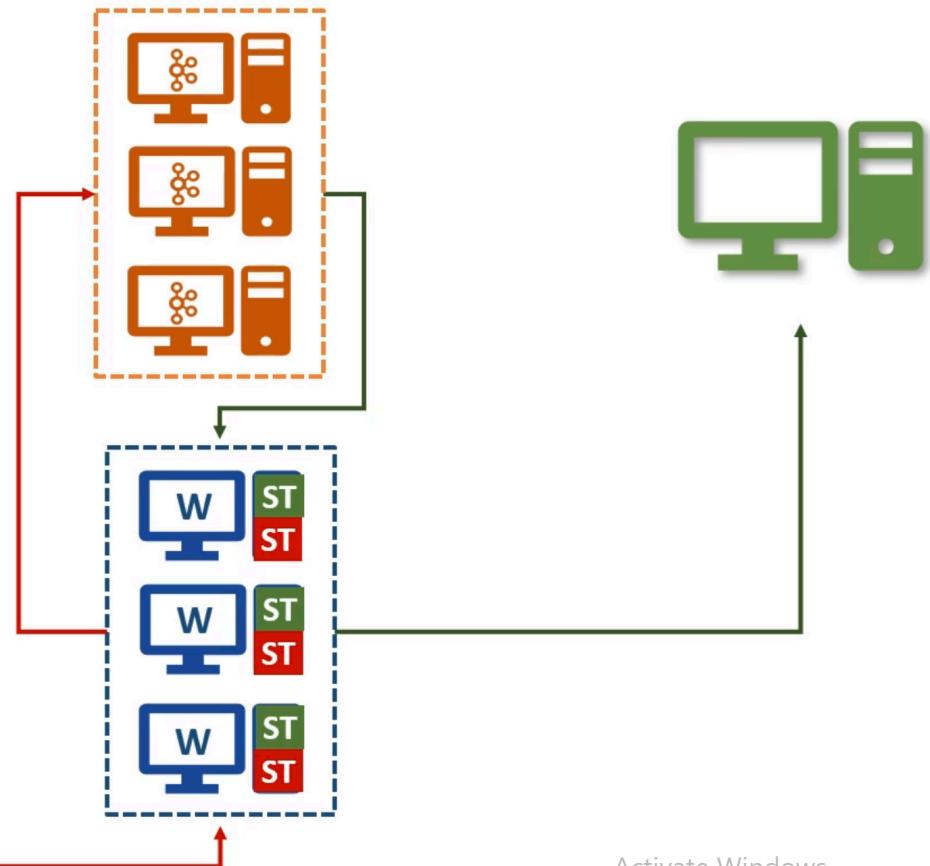
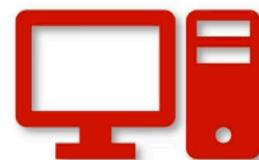


- We can add more connectors in a cluster if we have more available space.  
If the cluster is fully utilised We can dynamically add workers in kafka connect cluster without stopping any existing connectors.

# Kafka Connect – Transformations?

## Single Message Transformations – SMTs

1. Add a new field in your record using static data or metadata
2. Filter or Rename Fields
3. Mask some fields with a Null Value.
4. Change the Record Key
5. Route the record to a different Kafka Topic



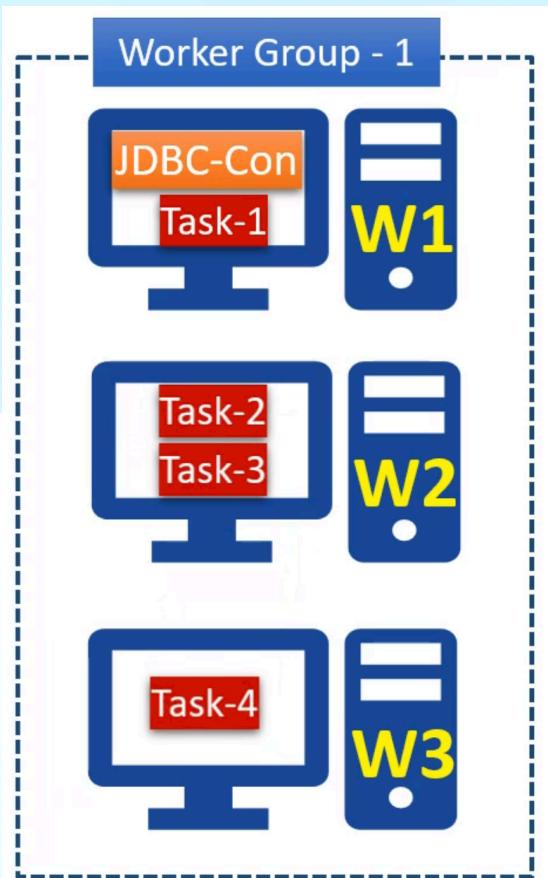
Activate Windows  
GIGANTIC.COM

# Kafka Connect Architecture

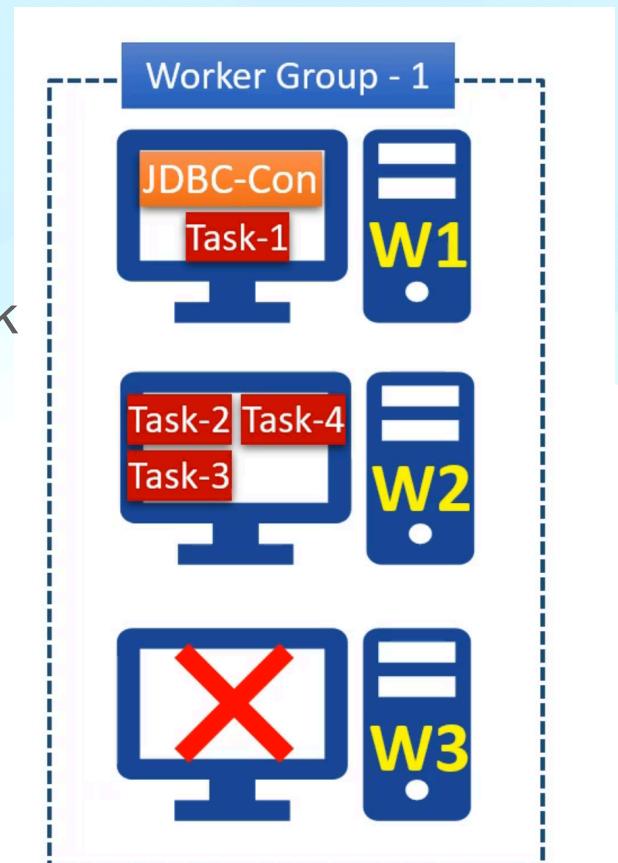
**It has 3 components : workers, connectors and tasks.**

- Workers are responsible for running connector and tasks.
- Worker group is same as consumer group.
- Worker are fault tolerant and self managed.
- Worker gives us reliability, high availability, scalability and load balancing

# Kafka Connect Architecture



If the workers process stops or crashes, other worker take care of the task and as the worker comes back it reassign some task to it.



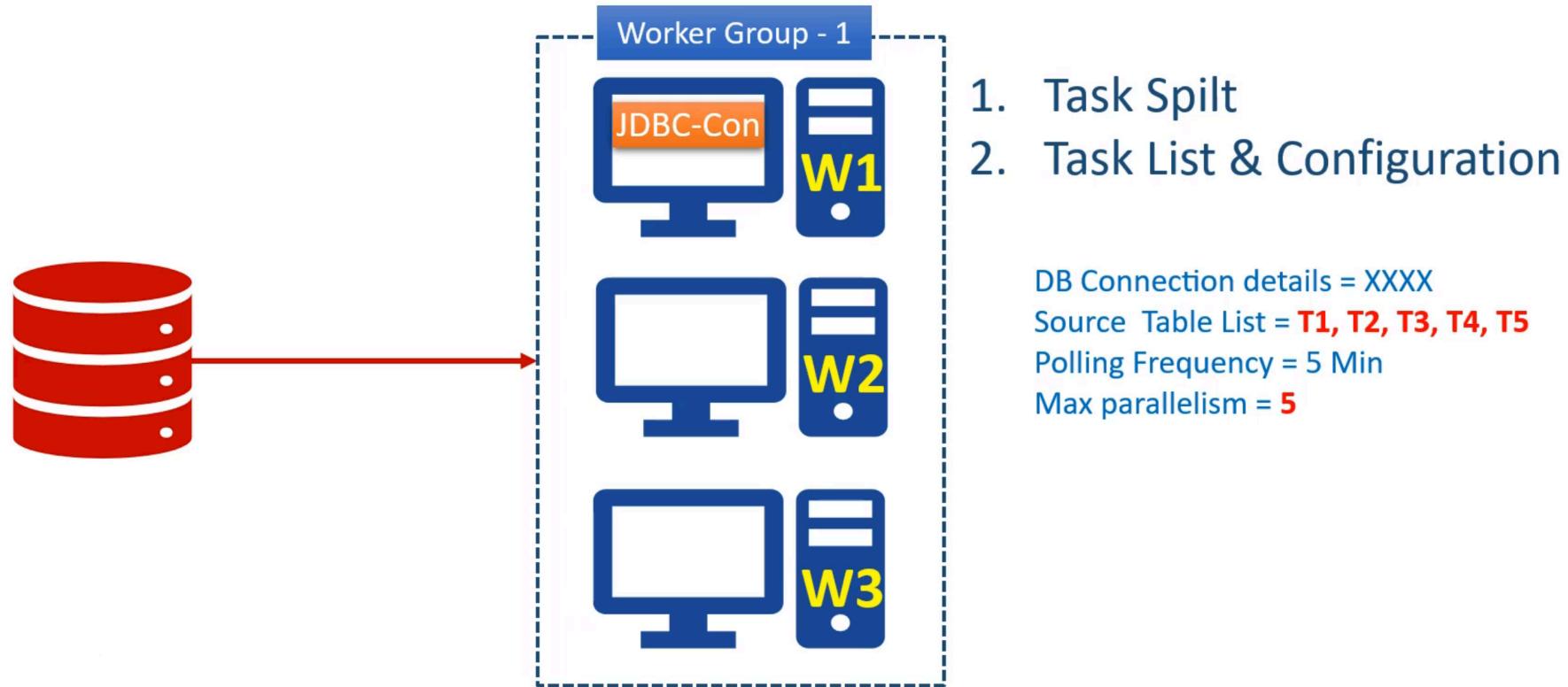
# Kafka Connect Architecture

## Process

- Install the connector in connect cluster.
- Configure the task using connector.Connectors make sure that each task run as independent by providing it required configs.
- After creating task list and configuring it , connectors give task list to workers.Worker distributes tasks among themselves for load balancing.
- Task are responsible for connecting source system, polling data at regular interval, collecting the records and handing over it to the workers.
- Task is only responsible for interacting with external systems , they do not send record to kafka cluster.
- Worker sends and retrieves data from cluster, task interact with worker and external system, not with kafka cluster.

# Kafka Connect Architecture

It has 3 components : workers, connectors and tasks.



# Connector Developer should know

- Reading and writing data to kafka cluster is standard activity , so it is taken care by connect framework(worker).
- Developer needs to take care for the things which are changing for different systems:
  1. How to split the input for parallel processing taken care by connector class.
  2. How to interact with external systems taken care by task class.

# Thank You