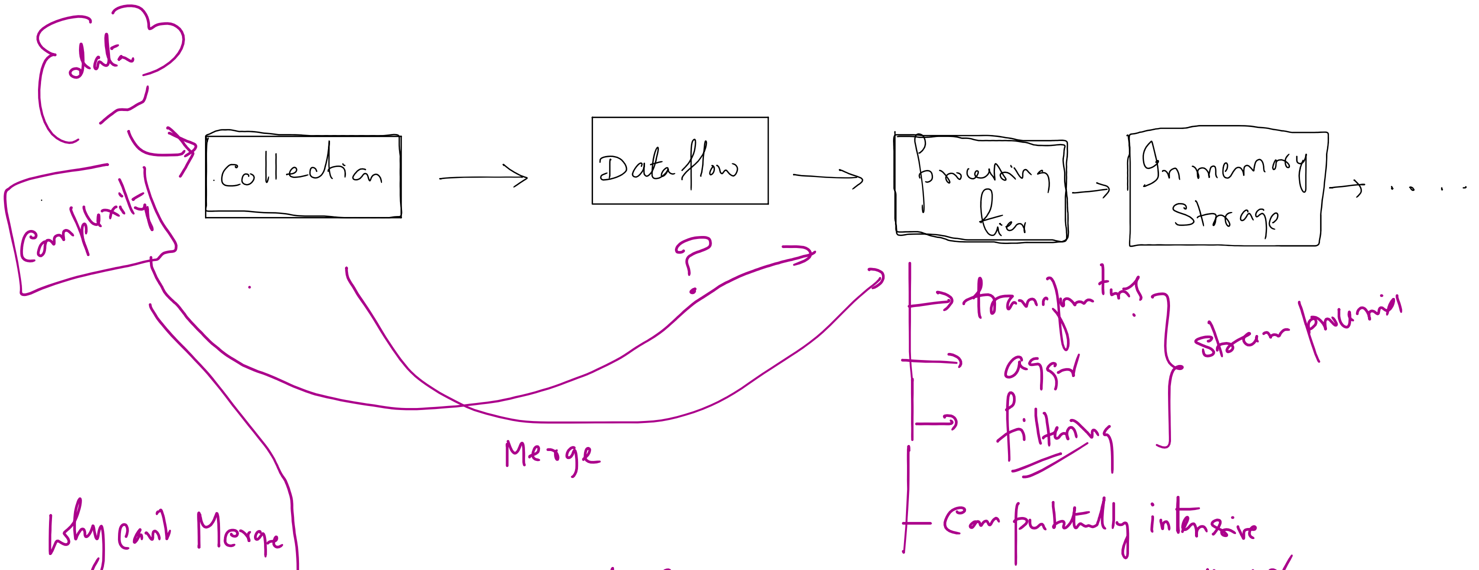
07 September 2025 18:00



Why can't Merge

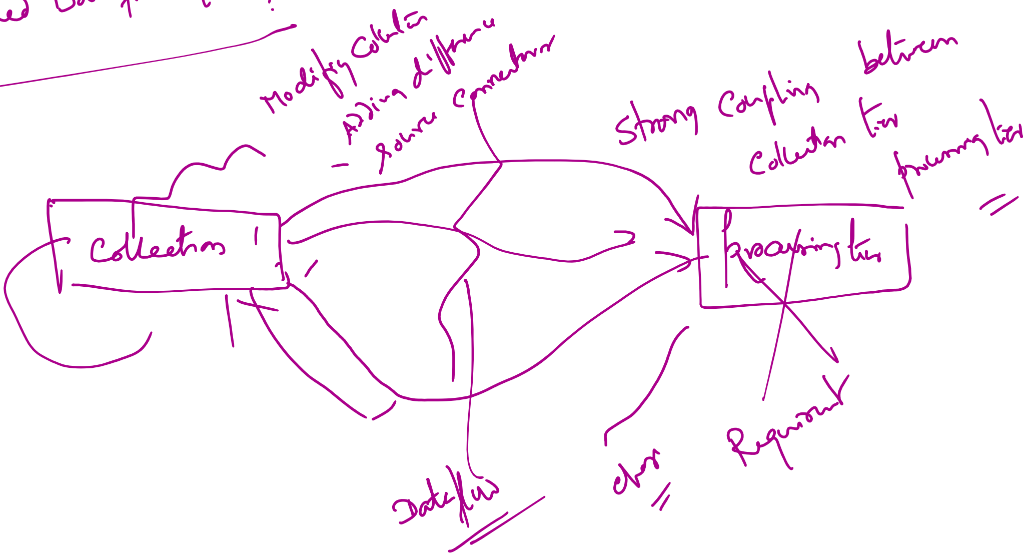
① Different data Sources

② Different Source Connects

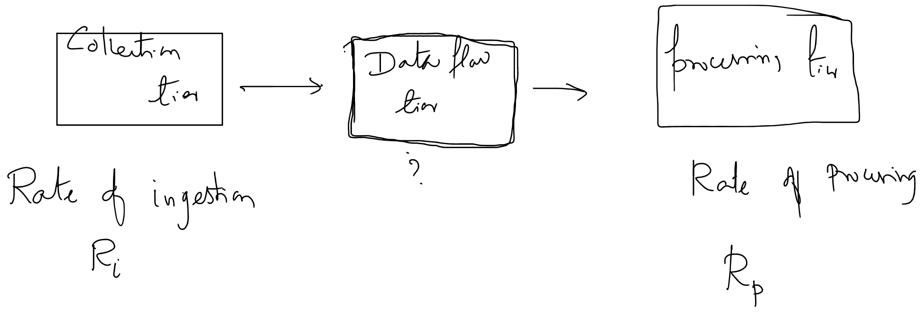
③ Different formats

Collection
– Interaction patterns

Why do we need Dataflow tier ?



② Impedence missmatch



Manage The Throughputs w.r.t Collection

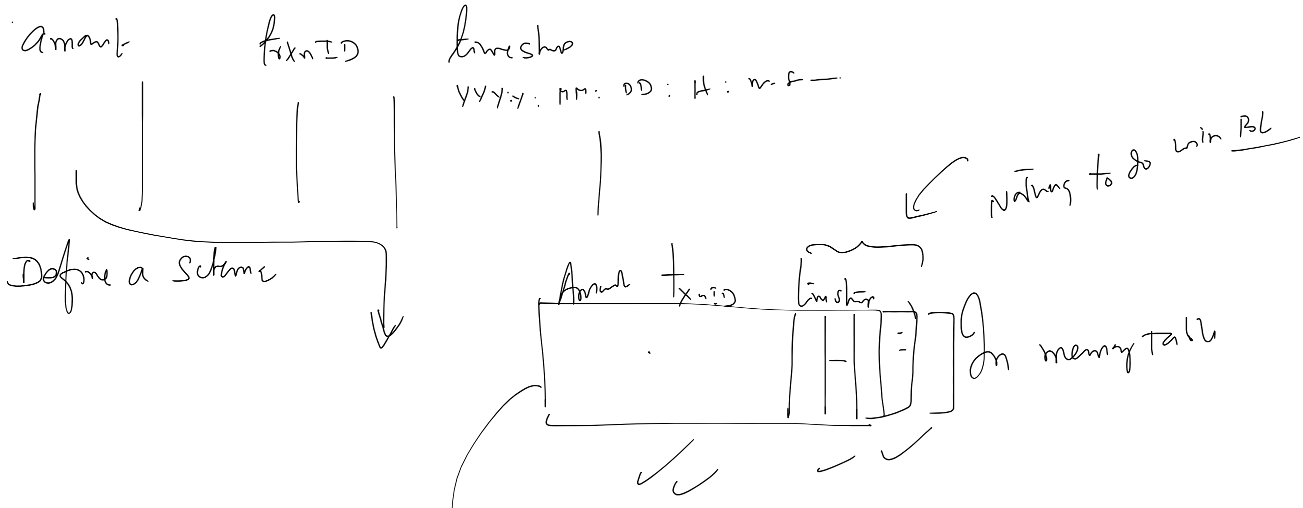
~p

Manage the throughputs w.r.t collection
Processing

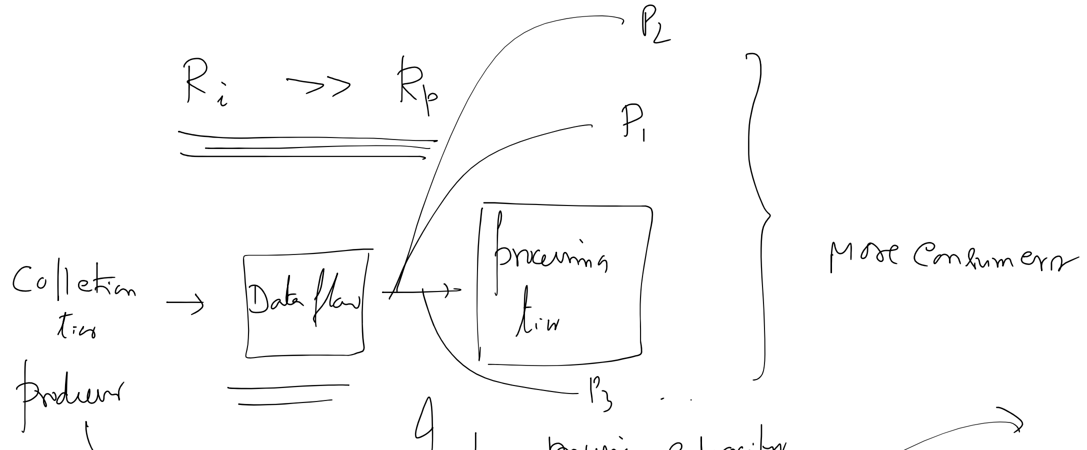
$R_i << R_p$ (rare : monitoring use cases : atmost once - in general)

$R_i \geq R_p$ (Back pressure problem)

$R_i >> R_p$ (most common)



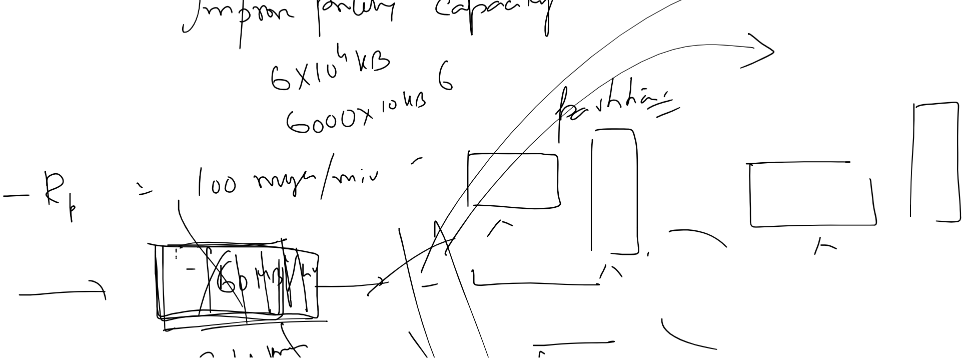
-> date frame ['timestamp'] = dataframe ('timestamp'].
astype ("timestamp")

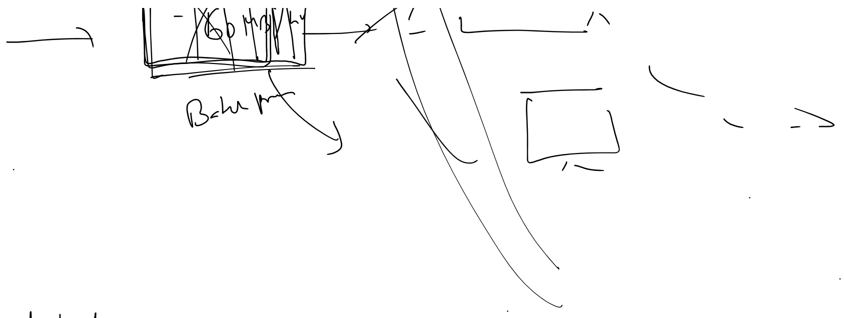


Improve processing capacity

6X10^4 KB
6000X10KB 6

$R_i - R_p$ = 100 msgs/min

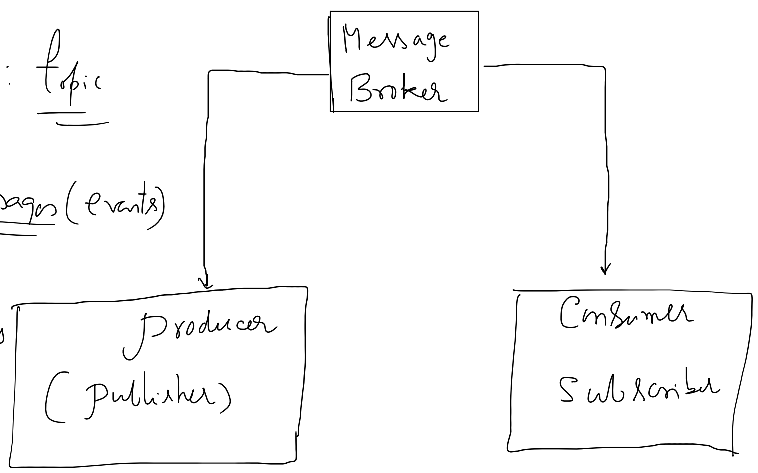




Kafka Architecture

Publisher – Subscriber

Where you are publishing those events: topic

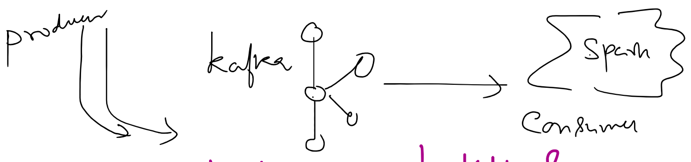


① Producer sends messages (events)

primitive data records

② Broker: responsible for receiving messages from producer and store them in a local storage

③ Consumer: Reads messages from broker and process them



Broker: A Name for Kafka Server

⇒ Broker acts as proxy between producers and consumers

Broker

P1 P2 - - Pm C1 C2 - - Cn

Producer / Consumers can be added on the fly On the fly

A1 (web) A2: Microservice Monitoring Analytics

Fraud / nonfraud volume | spend | vendor

This can be a cluster

'Interested' applns ... the

≃ 2010, 2011

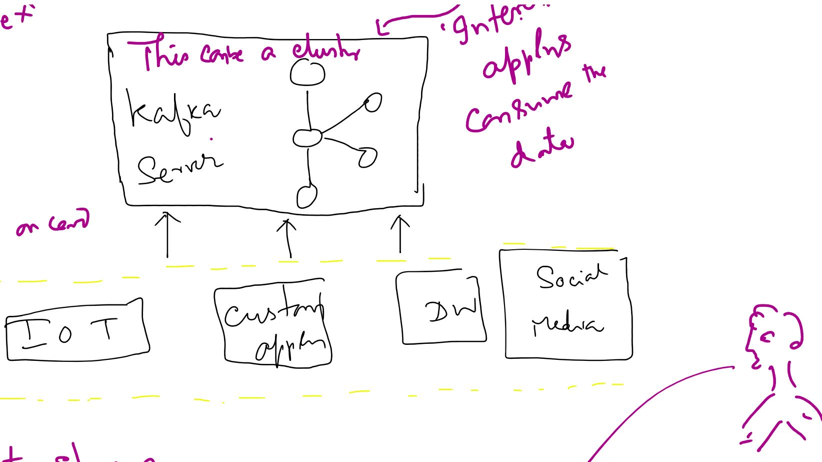
simple. line of text

≃ 2010, 2011

msg: Record of simple line of text from Data file

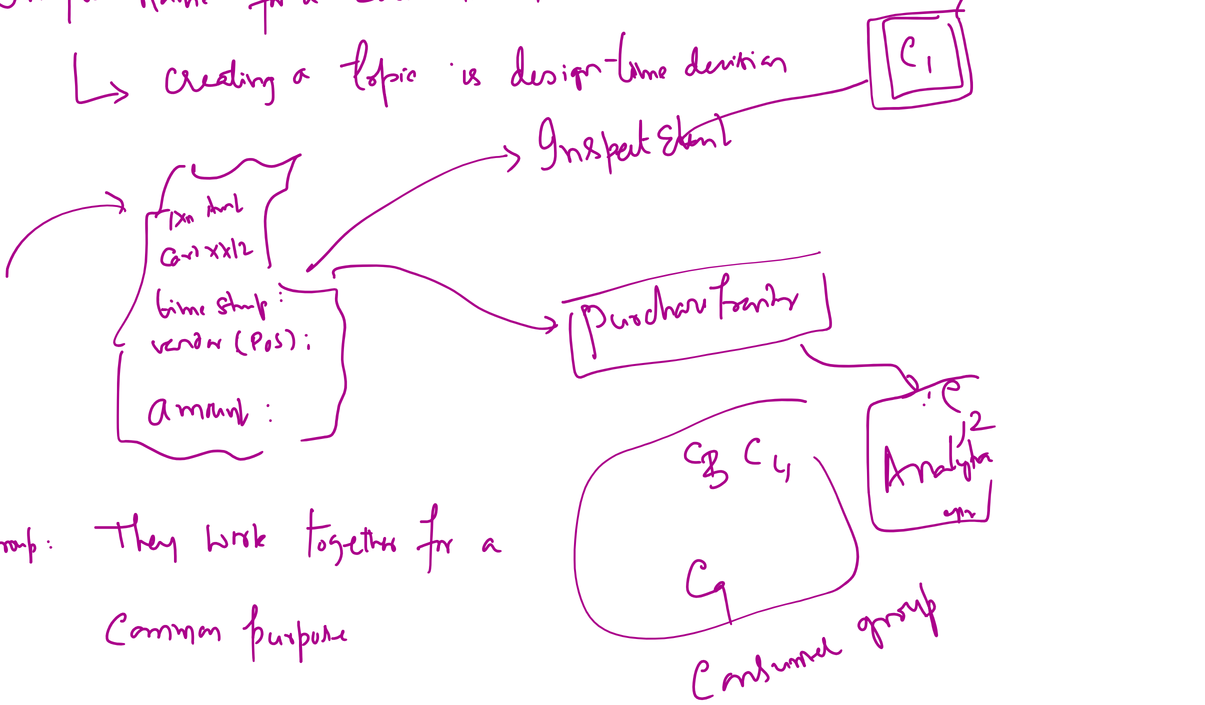
{A line(s) of text}

Members of infrastructure



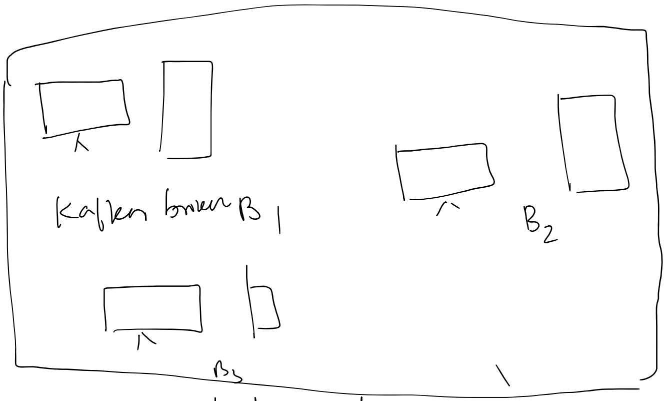
Topic: Unique name for a Data stream

↳ Creating a topic is design-time decision

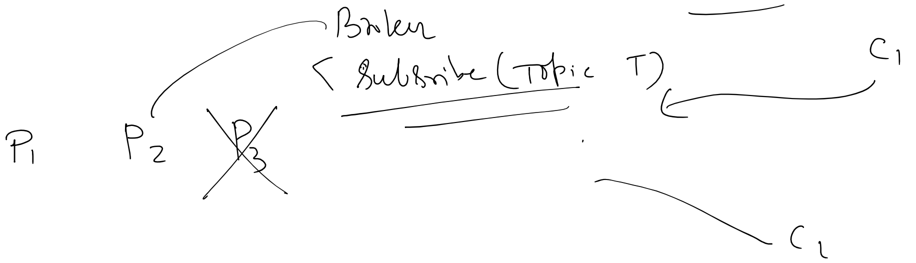


Consumer group: They work together for a common purpose

Kafka cluster



Kafka cluster

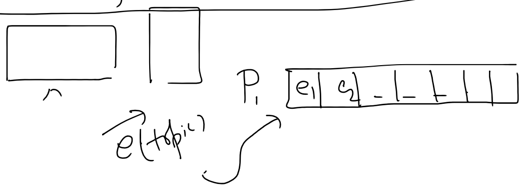


Broker is responsible for storing in local Storage

↳ Broker will be facing storage issues

Break the topic into multiple smaller parts and distribute it over multiple Computers in kafka cluster → partition

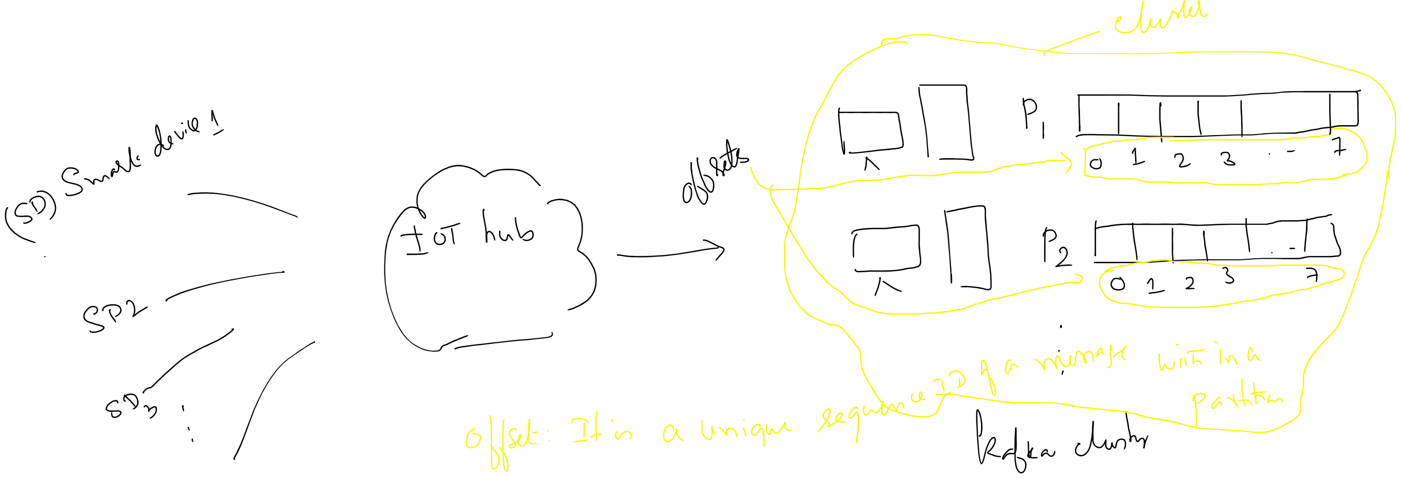
partition : It is smallest unit sitting on a single machine



A partition is small independent portion of a topic

No overlap of events across the partitions

No. of partitions in a topic is a design decision



Offset: It is a unique sequence ID of a message within a partition

⊙ It will be automatically assigned by the broker to every message as it arrives in the partition

⇒ messages are stored in a partition as 'append'

Offset-ID is arrival order number for msg

Topic name ⟶ partition ⟶ partition number ⟶ offset number

— Broker helps (C) —

* partition is used for Scalability

Estimate for number of partitions

Number of partition $(N) = \max\{N_p, N_c\}$

$$N_p = \frac{\text{Total System Throughput}}{\text{Max. Throughput of producer writing msg into single partition}}$$

$$N_p = \frac{T_t}{T_p}$$

$$N_c = \frac{T_t}{T_c} = \frac{\text{Total System Throughput}}{\text{Max Throughput of Consumer reading a msg from a single partition}}$$