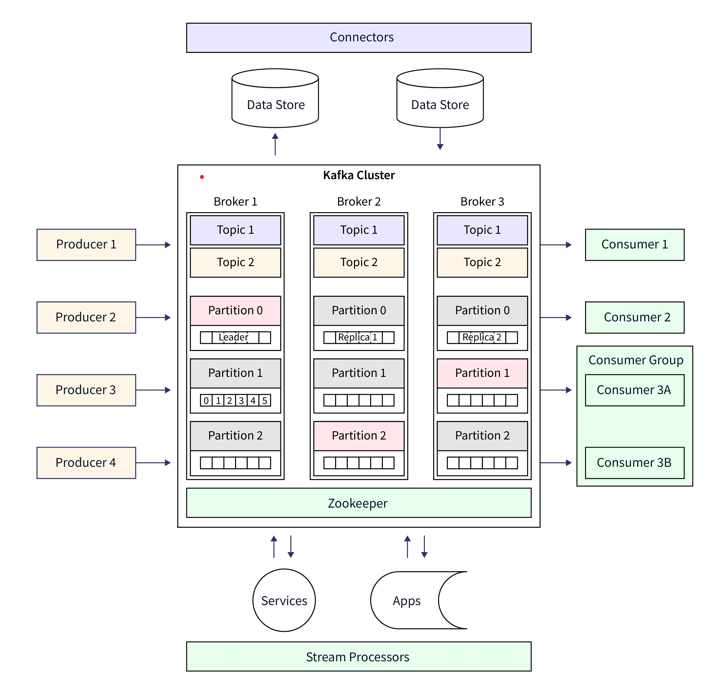
**Kafka -** Reliability / Scalability / Durability / Performance

Use cases - Metrics / Log Aggregation / Stream processing



**Topic** - Divides into partitions (specified while configuring a Topic)

**Broker** – (Kafka server) manages the topics message storage. For load balancing – multiple brokers. **Replication factor** cannot be greater that total number of available brokers.

**Zookeeper** – Brokers, Producer, Consumers are managed and coordinated by zookeeper. Keep track of each cluster’s status, including where broker located, what topics and partitions present and how each component configured

**Producer** – Publish data. For partitioning the data and selecting the appropriate broker for each partition. Compress and serialize the data, authenticate with cluster, and receive acks for messages

**Consumers** – Fetching and processing data. Can be part of consumer group. Scale horizontally and distribute the workload across multiple instances.

**Leader** – Responsible for all reads and writes for given partition

**Follower** – If a leader fails, one of follower will become next reader.

**Consumer Group** – group of consumers.

**Ensure Message Order** –

1. Single partition
   1. Throughput constraint
   2. Reduced parallelism
2. External sequencing with Time Window Buffering

Each message with a global sequence number.

Producer – manage global sequence by a shared resource that accessible across all producer processes (db sequence or distributed counter)

Consumer – grou pmessage into time windows and process sequentially

ISR – In Sync Replicas