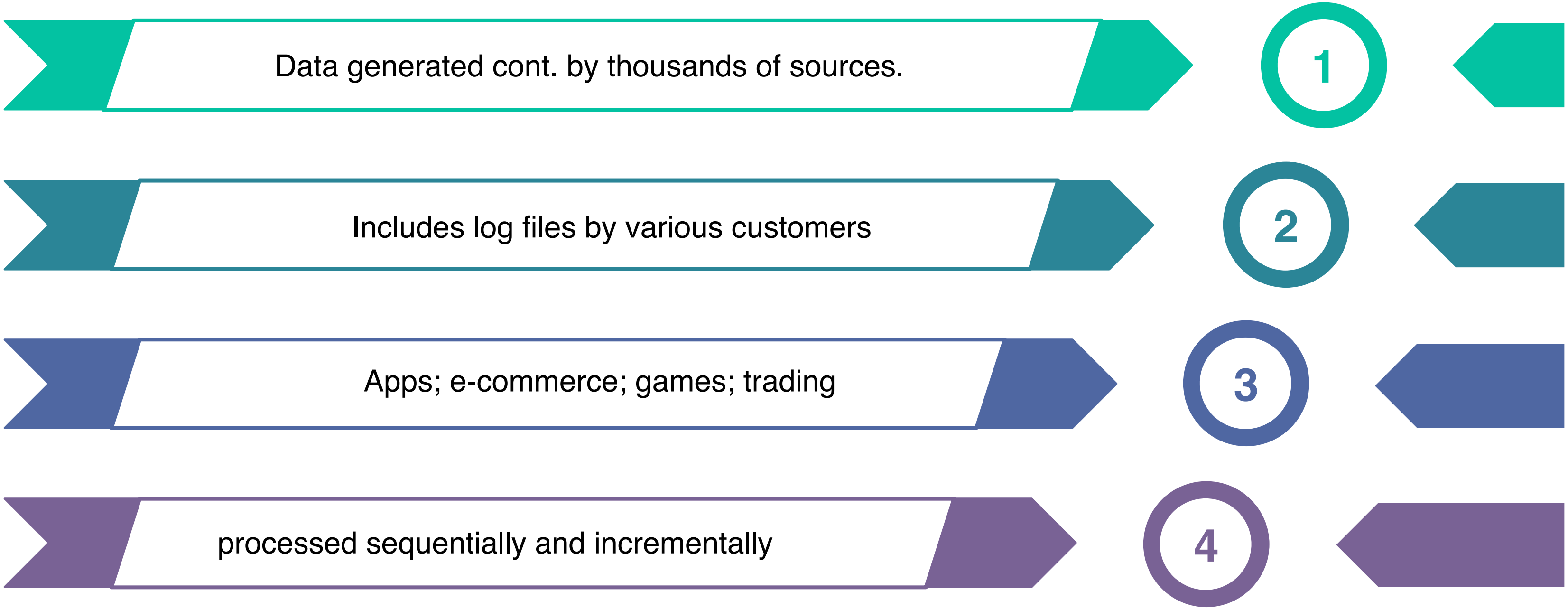
Live Streaming Data

A snapshot of live streaming data;
Its uses and examples.

What is streaming data?



1. Data generated cont. by thousands of sources.
2. Includes log files by various customers
3. Apps; e-commerce; games; trading
4. processed sequentially and incrementally

Analytics

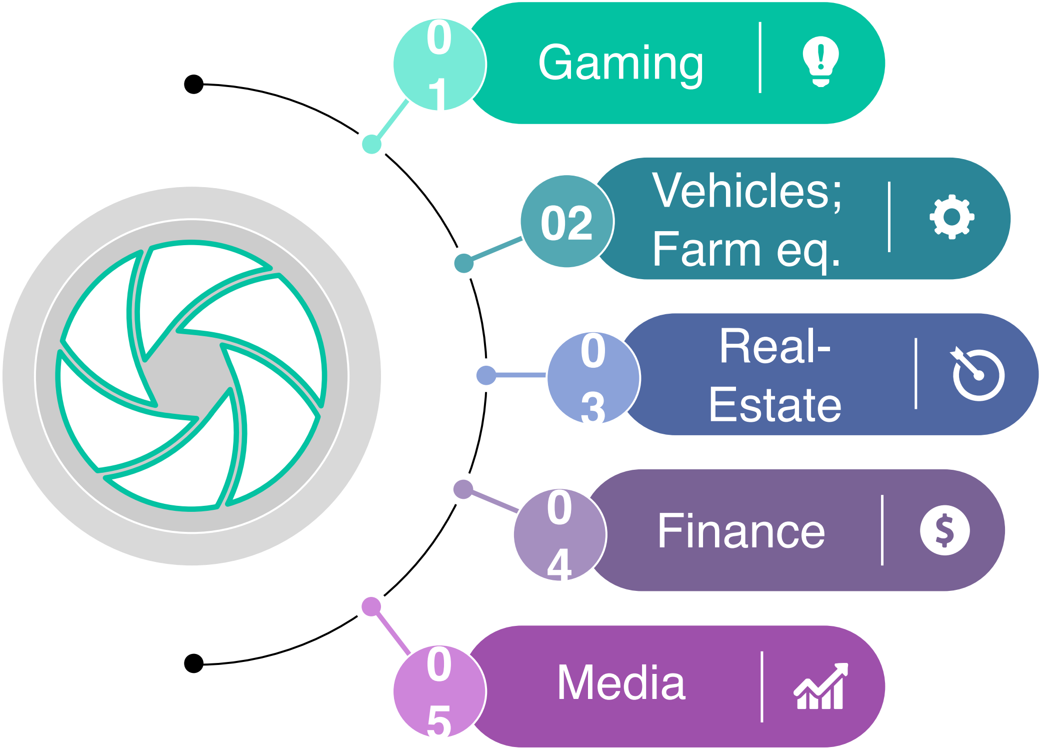
Aggregations

Correlations

Filtering

Sampling

Streaming Data Examples



01 Gaming

Collects streaming data about player-game interaction; analyzes in real-time; offers incentives to engage its players

02 Vehicles; Farm eq.

App monitors eq ,detects defects; then automatically orders spare part

03 Real-Estate

Website tracks subset of data from consumer’s mobile devices and makes real-time recommendations

04 Finance

Tracks changes in market in real-time, computes value at risk, and rebalances portfolio.

05 Media

Streams billions clickstream records from its online properties; enriches data

Benefits

- Streaming data processing is beneficial in most scenarios where new, dynamic data is generated on a continual basis.

- It applies to most of the industry segments and big data use cases.

- Companies generally begin with simple applications such as collecting systems logs and rudimentary processing like rolling min-max computations

- Applications then evolve into more sophisticated near real-time processing.

- Initially, applications may process data streams to produce simple reports, and perform simple actions in response, such as emitting alarms when key measure exceed certain thresholds.

- Eventually, those applications perform more sophisticated forms of data analysis, like applying machine learning algorithms, and extract deeper insights from the data.

- Over time, complex, stream and event processing algorithms, like decaying time windows to find the most recent popular movies are applied, adding more visibility from the insights.

Challenges

- Streaming data processing requires two layers:Storage layer and a processing layer.

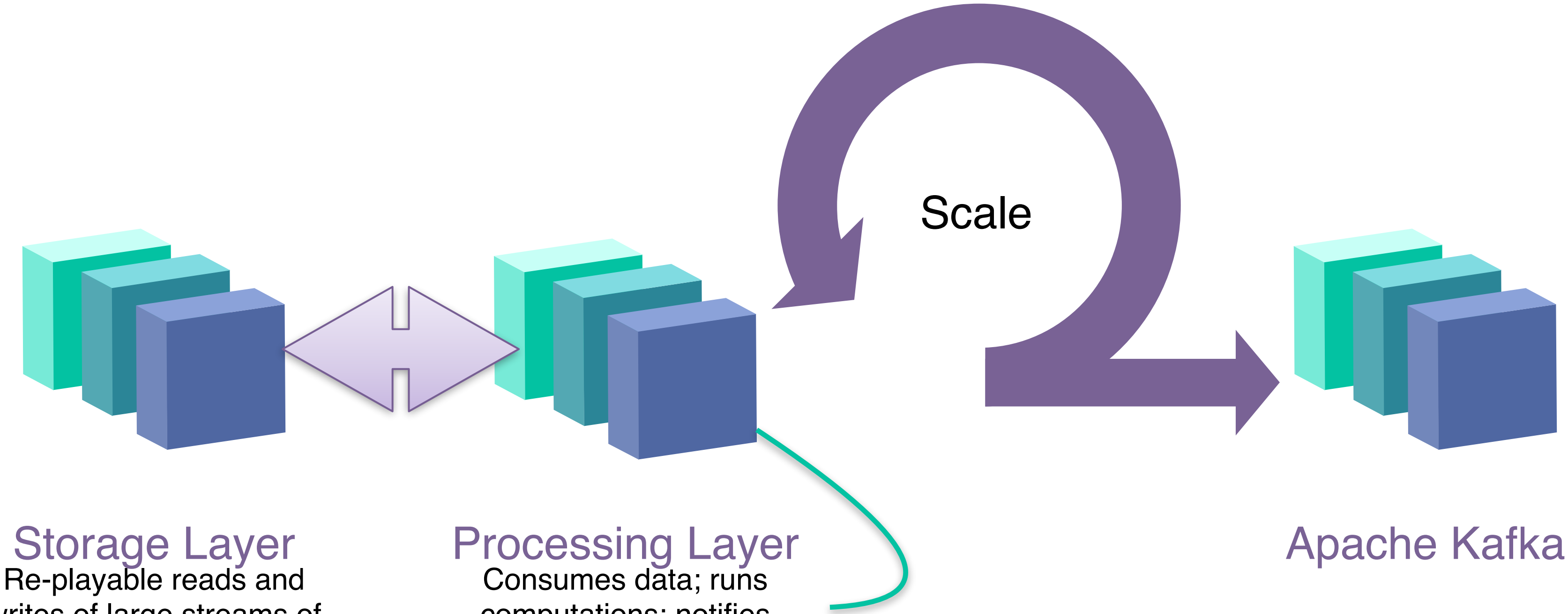
- The storage layer needs to support record ordering and strong consistency to enable fast, inexpensive, and replayable reads and writes of large streams of data.

- The processing layer is responsible for consuming data from the storage layer, running computations on that data and then notifying the storage layer to delete data that is no longer needed.

- You also have to plan for scalability, data durability, and fault tolerance in both the storage and processing layers.

- As a result, many platforms have emerged that provide the infrastructure needed to build streaming data applications including

Challenges in working with Streaming Data



Storage Layer

Re-playable reads and writes of large streams of data.

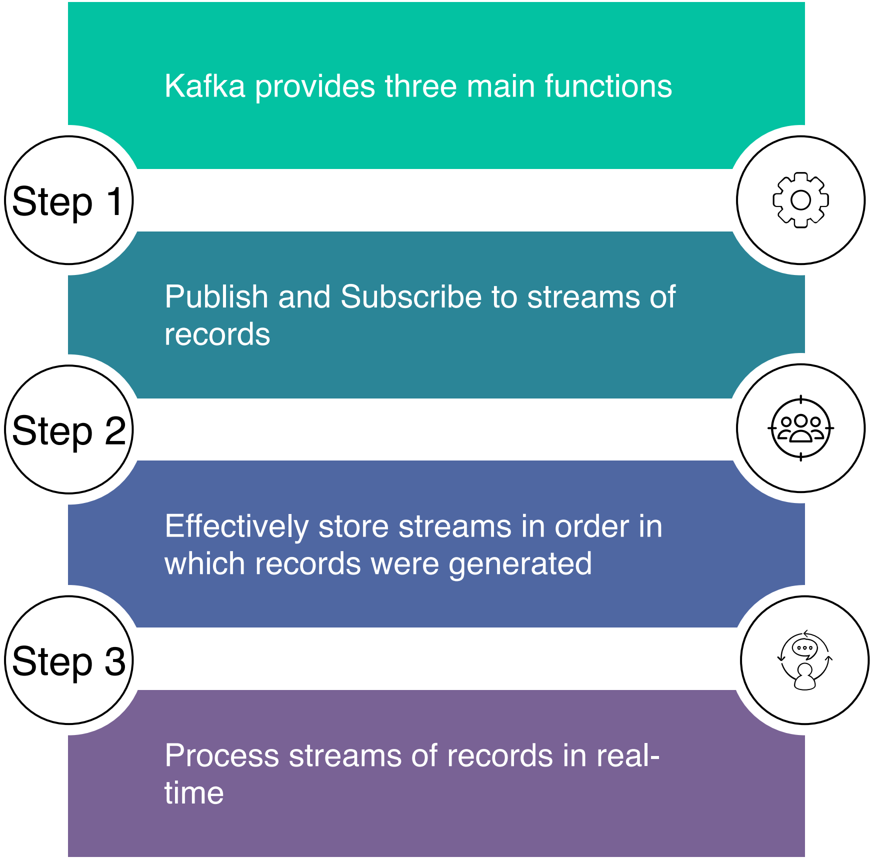
Processing Layer

Consumes data; runs computations; notifies storage to delete data

Apache Kafka

What is Apache Kafka

Apache Kafka is a distributed data store optimized for ingesting and processing streaming data in real-time. Streaming data is data that is continuously generated by thousands of data sources, which typically send the data records in simultaneously. A streaming platform needs to handle this constant influx of data, and process the data sequentially and incrementally.



Kafka provides three main functions

Step 1

Publish and Subscribe to streams of records

Step 2

Effectively store streams in order in which records were generated

Step 3

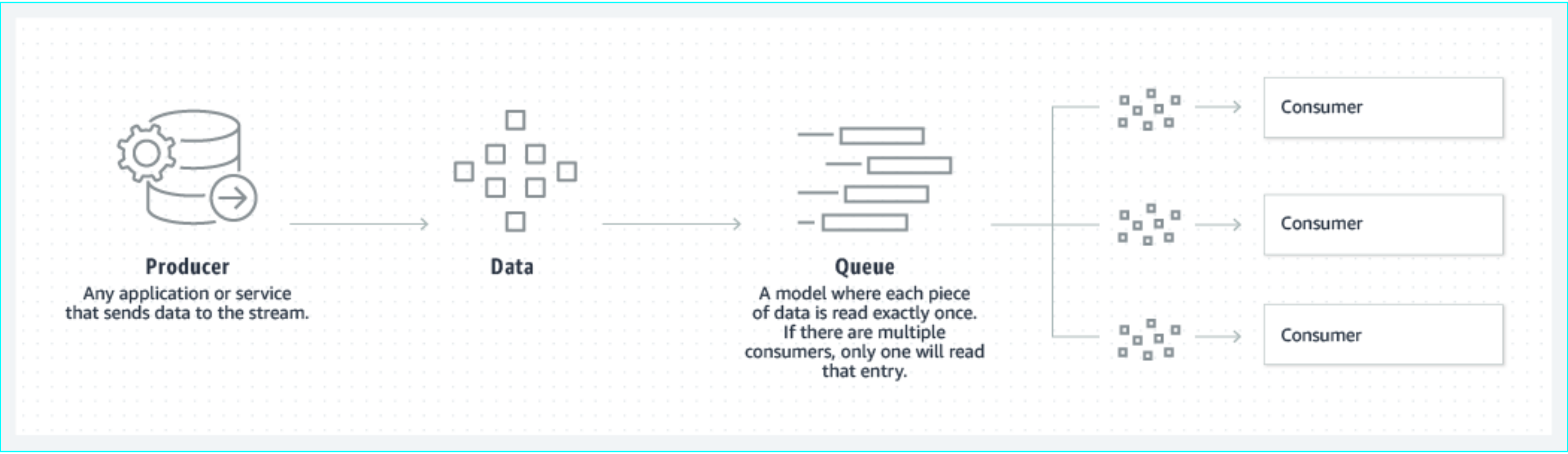
Process streams of records in real-time

Why use Kafka?

Kafka is used to build real-time streaming data pipelines and real-time streaming applications. A data pipeline reliably processes and moves data from one system to another, and a streaming application is an application that consumes streams of data. For example, if you want to create a data pipeline that takes in user activity data to track how people use your website in real-time, Kafka would be used to ingest and store streaming data while serving reads for the applications powering the data pipeline. Kafka is also often used as a message broker solution, which is a platform that processes and mediates communication between two applications.

How Kafka Works

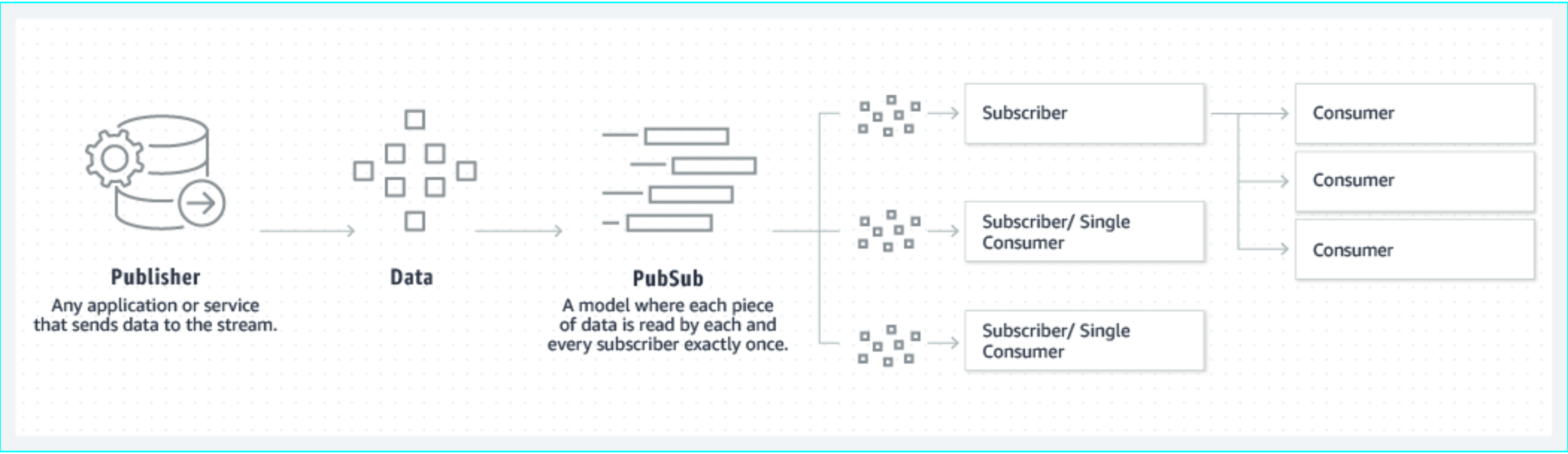
Queuing



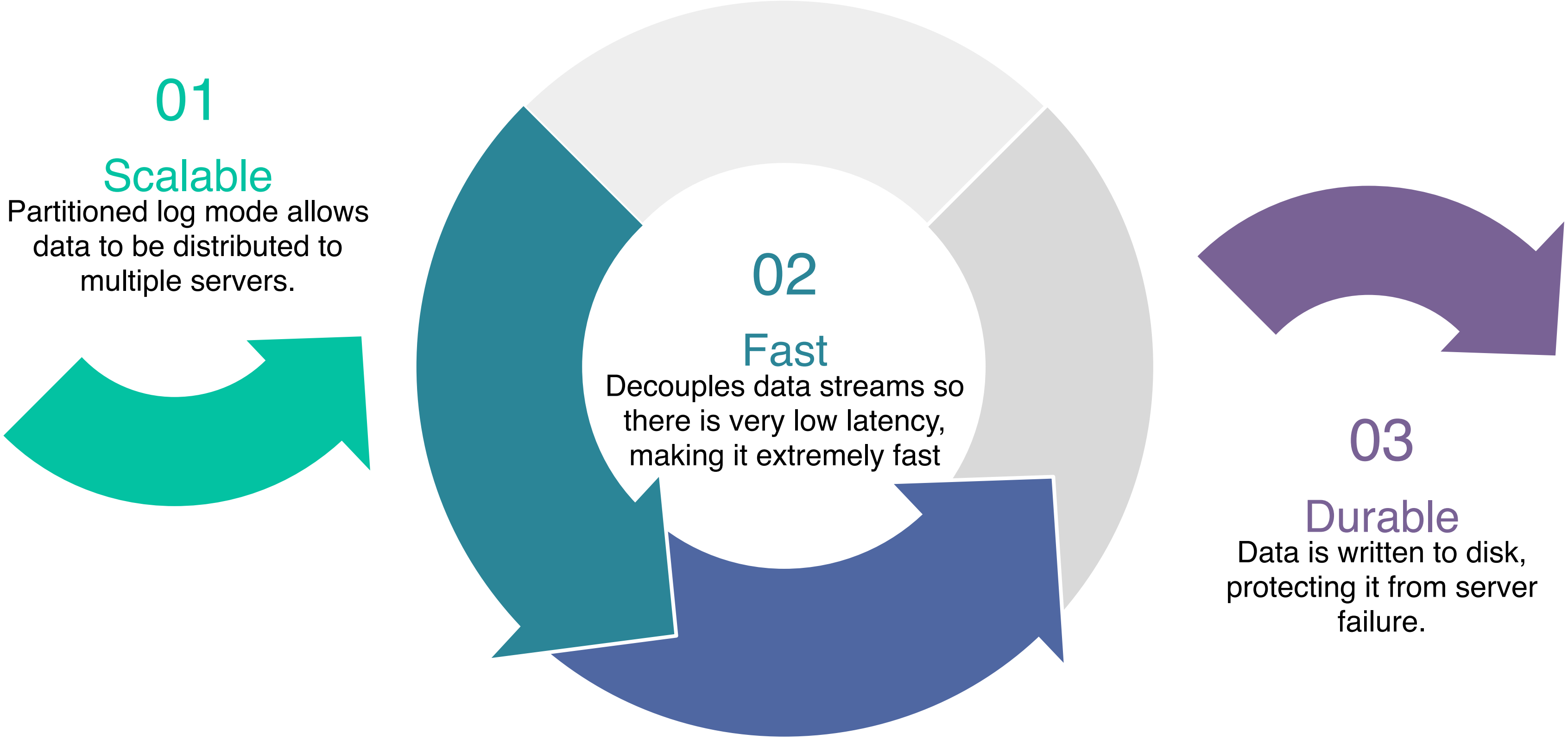
Any application or service that sends data to the stream.

A model where each piece of data is read exactly once. If there are multiple consumers, only one will read that entry.

Publish-Subscribe


Publisher
Any application or service that sends data to the stream.
Data
PubSub
A model where each piece of data is read by each and every subscriber exactly once.
Subscriber
Subscriber/ Single Consumer
Subscriber/ Single Consumer
Consumer
Consumer
Consumer

Benefits of Kafka’s Approach.



01

Scalable

Partitioned log mode allows data to be distributed to multiple servers.

02

Fast

Decouples data streams so there is very low latency, making it extremely fast

03

Durable

Data is written to disk, protecting it from server failure.