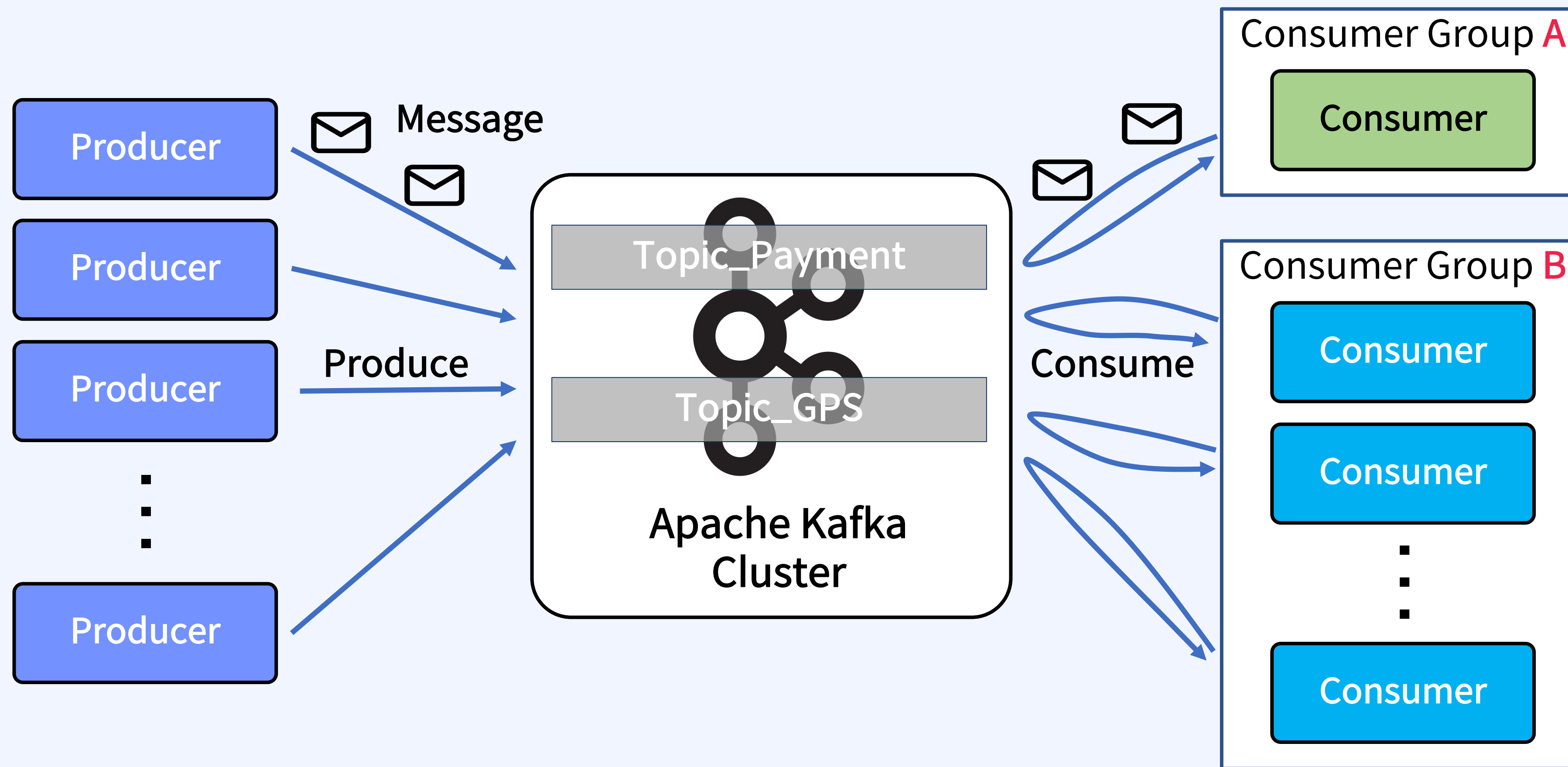


# Apache Kafka 기본 개념 및 이해

4 Producer

## Apache Kafka 주요 요소 Producer, Consumer, Consumer Group

### 4. Producer



## Apache Kafka Clients

### Producer, Consumer, Consumer Group

4.

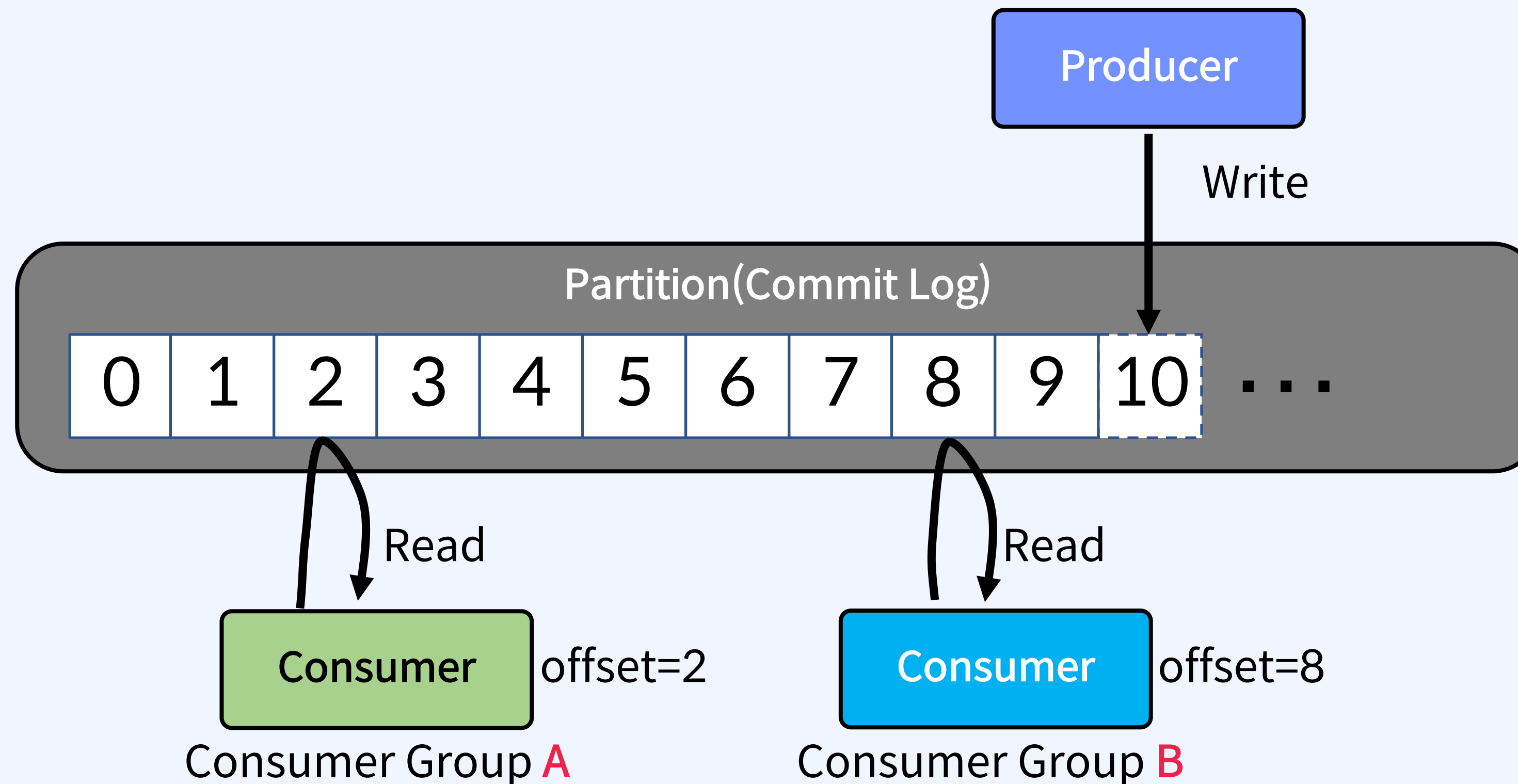
Producer

- **Producer** : 메시지를 생산(Produce)해서 Kafka의 Topic으로 메시지를 보내는 애플리케이션
- **Consumer** : Topic의 메시지를 가져와서 소비(Consume)하는 애플리케이션
- **Consumer Group** : Topic의 메시지를 사용하기 위해 협력하는 Consumer들의 집합
- 하나의 Consumer는 하나의 Consumer Group에 포함되며, Consumer Group내의 Consumer들은 협력하여 Topic의 메시지를 병렬 처리함

## Producer와 Consumer의 분리(Decoupling)

### Producer와 Consumer의 기본 동작 방식

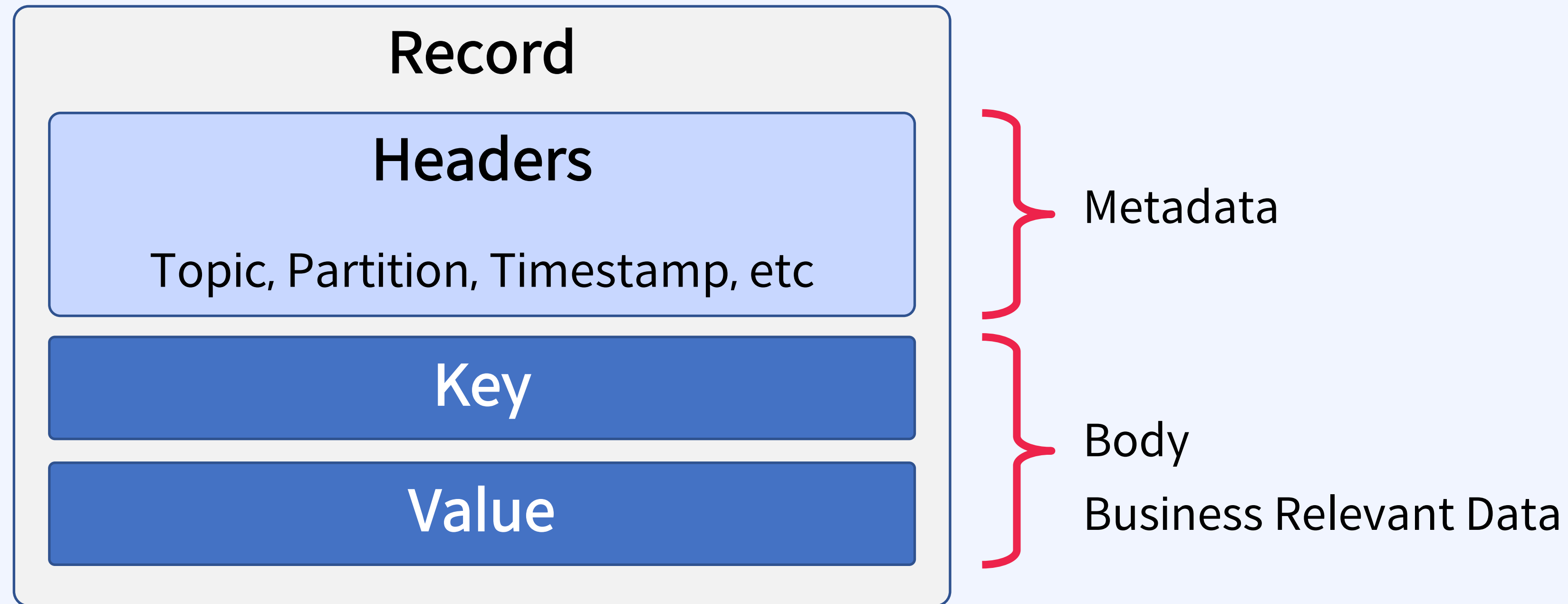
- Producer와 Consumer는 서로 알지 못하며, Producer와 Consumer는 각각 고유의 속도로 Commit Log에 Write 및 Read를 수행
- 다른 Consumer Group에 속한 Consumer들은 서로 관련이 없으며, Commit Log에 있는 Event(Message)를 동시에 다른 위치에서 Read할 수 있음



## Record(Message) 구조 Header, Key, Value

4.  
Producer

Message == Record == Event == Data



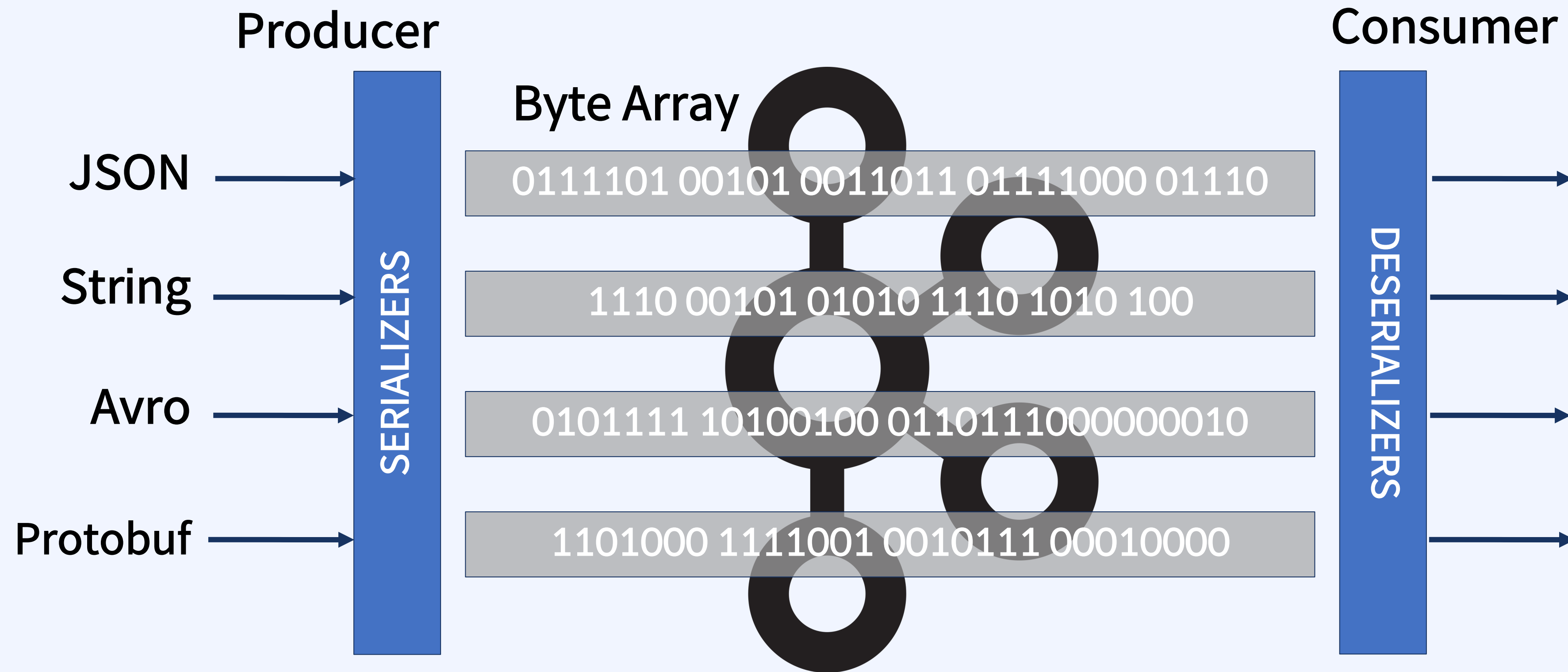
Key와 Value는 Avro, JSON 등 다양한 형태가 가능

강사님 크로마키  
280px \* 400px

## Serializer/Deserializer To/From Byte Array

4.  
Producer

Kafka는 Record(데이터)를 Byte Array로 저장



참고) <https://kafka.apache.org/10/documentation/streams/developer-guide/datatypes.html>

## Producer Sample Code Serializer

4.

Producer

### Key 와 Value 용 Serializer를 각각 설정

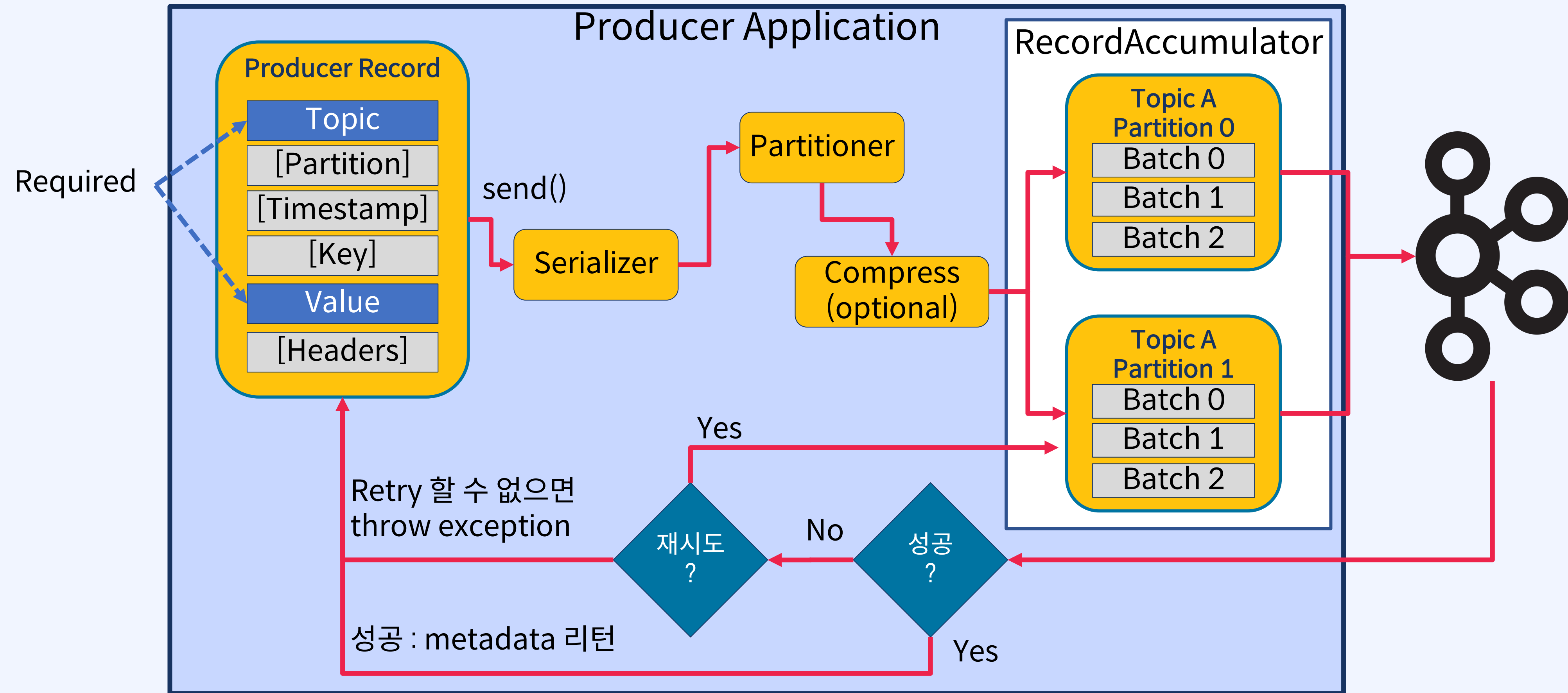
```
private Properties props = new Properties();

props.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, " broker101:9092,broker102:9092 ");
props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, org.apache.kafka.common.serialization.StringSerializer.class);
props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, io.confluent.kafka.serializers.KafkaAvroSerializer.class);

KafkaProducer producer = new KafkaProducer(props);
```

# Producing to Kafka High-Level Architecture

## 4. Producer



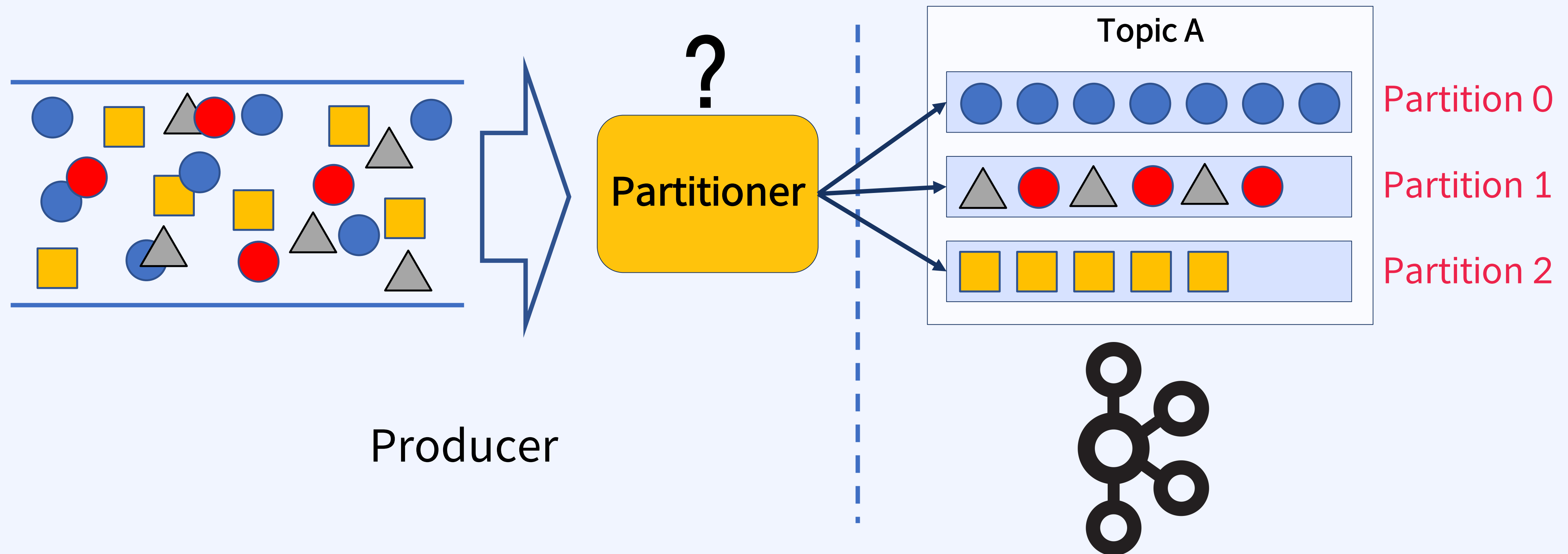


## Partitioner 의 역할

메시지를 Topic의 어떤 Partition으로 보낼지 결정

4.  
Producer

$$\text{Partition} = \text{Hash}(\text{Key}) \% \text{Number of Partitions}$$

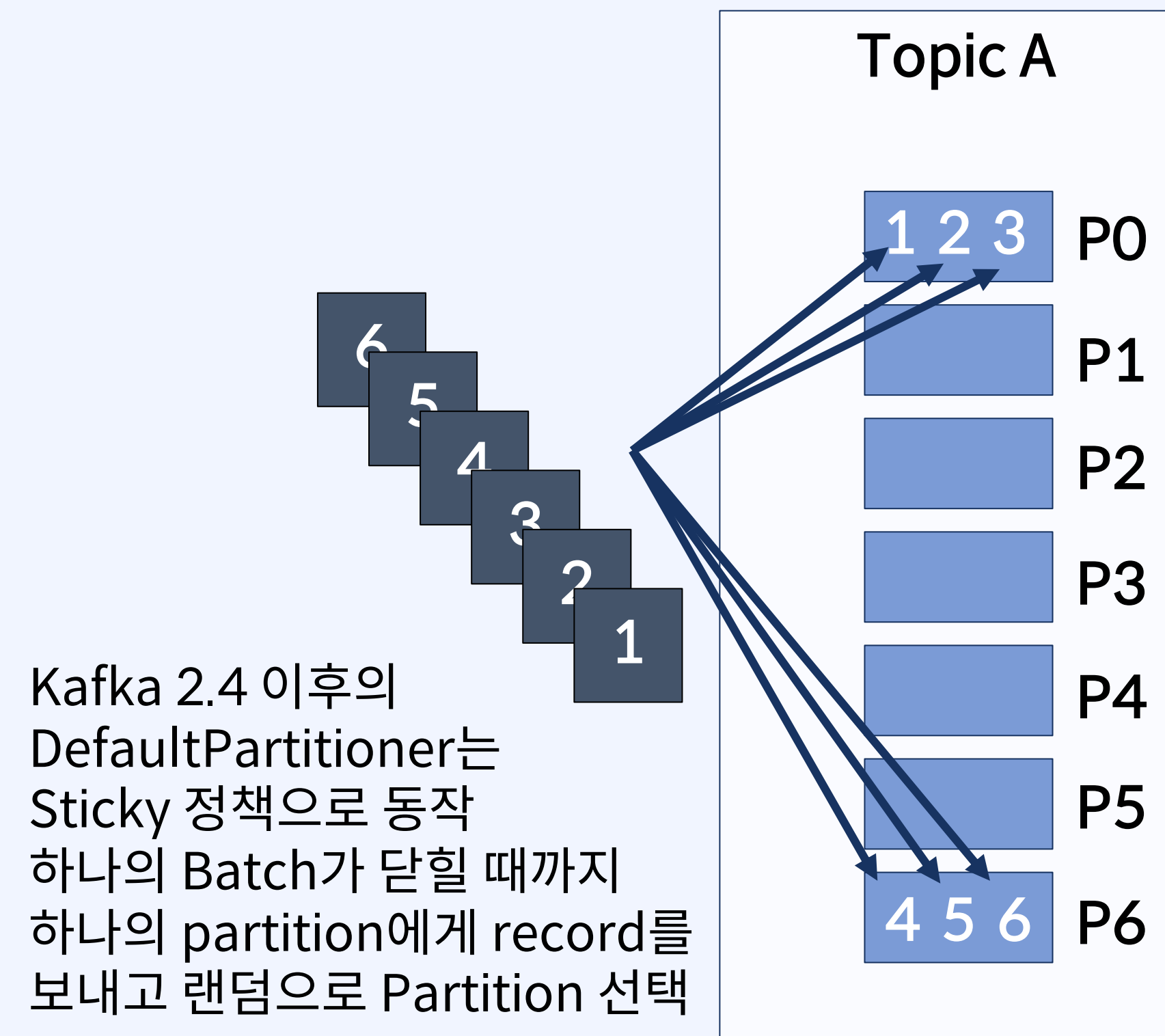
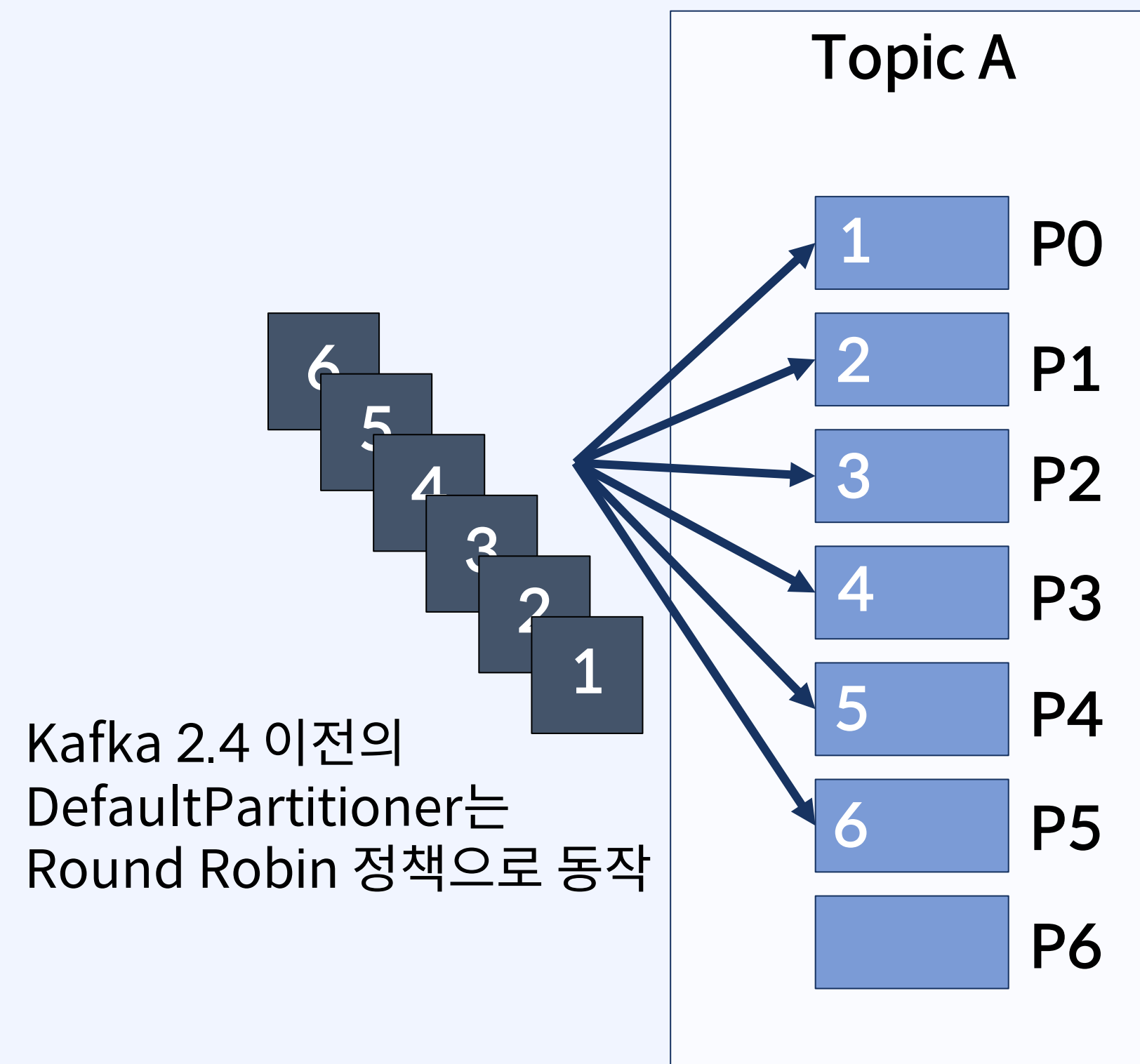


전제 조건 : Key 가 null 이 아닐 때

## Partitioner 의 종류 성능, 작동 방식이 다양함

# Apache Kafka Producer Improvements with the Sticky Partitioner<sup>1)</sup>

전제 조건 : Key 가 null 일 때



Partitioner는  
개발해서  
교체 가능

1) <https://www.confluent.io/blog/apache-kafka-producer-improvements-sticky-partitioner/>

## Summary

### Message 구조, Serializer/Deserializer, Partitioner

- Message == Record == Event == Data
- Message는 Header 와 Key 그리고 Value 로 구성
- Kafka는 Record(데이터)를 Byte Array로 저장
- Producer는 Serializer, Consumer는 Deserializer를 사용
- Producer는 Message의 Key 존재 여부에 따라서 Partitioner를 통한 메시지 처리 방식이 다름