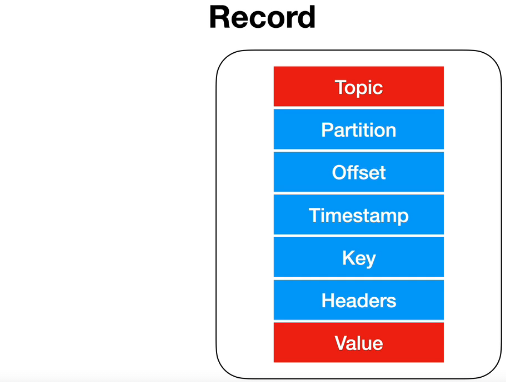
1. **Agenda:**
   1. Walkthrough the Kafka Producer Workflow.
2. In traditional Message System, The Producer API sends the msg to the broker & gets the response back & it is done.  
   But a Kafka Producer does a lot of things behind the scene as you can see in the below snapshot.  
   Diagram

   Description automatically generated
   1. **ProducerRecord**:
      1. To start sending msg to Kafka Broker, we create ProducerRecord on which we can set the following attributes’ values but mainly topic & value (payload).  
         Chart

         Description automatically generated with medium confidence
   2. send():
      1. Once we have the ProducerRecord, we call the send() on **the Producer**.
      2. At this time, Producer will hand over the **ProducerRecord** to the **Serializer**.
   3. **Serializer**:
      1. Producer will hand over the ProducerRecord to a Serializer when Producer.send(ProducerRecord) is called.
      2. **Purpose of Serializer**: To convert Java Object into Bytes Array.  
         Basically key & values from the ProducerRecord.  
         
      3. Kafka has several in-built serializers that work fine with common types in java.  
         But you can create your own serializer to convert java object types into bytes array.
      4. Serializer will look for the **key & value** in the ProducerRecord and converts from Java Types into Bytes Array.
      5. Later on, we will see **Avro** which does it for us out of the box.
      6. Once the serializer hands over the Bytes of array, it hands over the bytes to the Partitioner.
   4. **Partitioner**:
      1. Partitioner receives the bytes of array from the Serializer.
      2. It then checks if the record has partition#.   
         if yes, then the record will go into the partition with that number.  
         If no, it will use the value of key and hashing algo to calculate the partition#.  
         If key also not present, then the partition# will be assigned in round-robin fashion.
   5. **Producer**:
      1. At this point, producer knows into which partition the record will go.
      2. So, it will add the record into appropriate batch (batch is identified based on Topic Name & Partition#).
   6. **Batch**:
      1. A separate thread will pick up the batches and send them to Kafka Broker.
   7. **Kafka Broker:**
      1. If Broker successfully writes the msg into the Kafka, we get a Record Metadata back.  
         If due to any reason, it gets failed, we get failure response back.
      2. At this time Producer can retry multiple times to send the msg again to the Kafka Broker.  
         If it keeps failing, it will give up and throws an Exception.
3. So, it all happens within the Producer.  
   