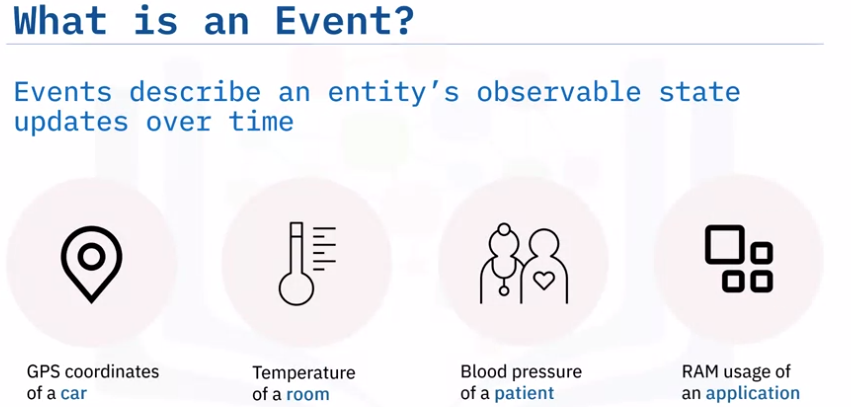
**Apache Kafka**

# **1. Introduction to Distributed Event Streaming Platform Components**

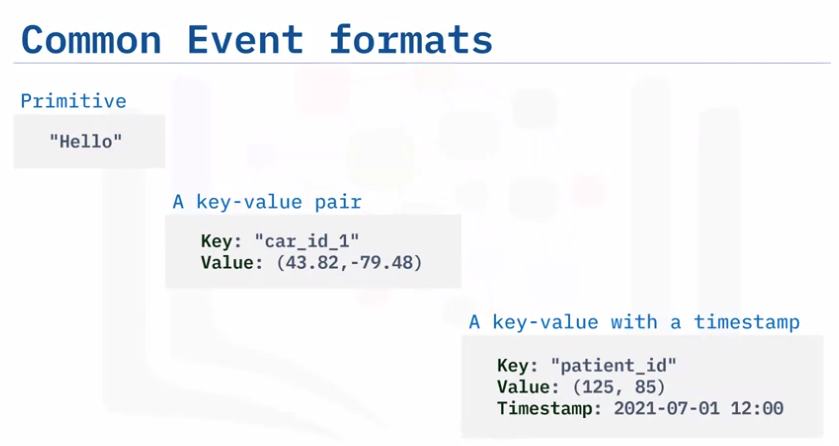
## 1.1. **Event**

Event is type of data which describes the entity’s observable state update over time.



*Common* ***Event*** *formats:*

* ***Primitive*** type: it’s like a plain text, number or date.
* ***Key-value pair*** type.
* ***Key-value pair with timestamp*** type.



## **1.2. Event Streaming vs Event Stream Processing vs Event Processing**

***Event source*** may continuously generate a large event volume at a short time interval or nearly real-time. Those real-time ***events*** need to be properly transported to an event destination such as a file system, another external database, or an application. The continuous event transportation between an ***event source*** and an ***event destination*** is called ***event streaming***.

You might be confusing ***Event Stream Processing*** with the term ***Event Streaming.***

***+ Event streaming*** refers to the process of moving event data from place to place efficiently so other systems can easily access and analyze it. Therefore ***Even streaming*** is a part of the ***Event Streaming Processing.***

***+ Event processing:*** it looks at individual ***events*** one at a time, whereas ***event stream processing*** handles many related ***events*** together. ***Event processing*** is like looking at individual drops of water. ***Event stream processing*** is like putting your finger under a running faucet to see how warm the water is

## **1.3. Event Streaming Platform (ESP)**

### **1.3.1. Problem of old approaches**

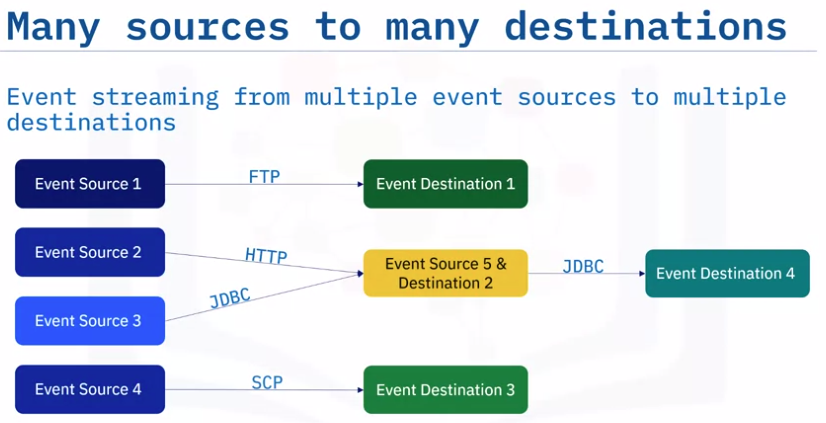
***One source to one destination:***

******

***Many source to many destination (****most of cases****):***

Data transfer pipelines may be based on different communication protocols such as:

* ***FTP***: File Transfer Protocol.
* ***HTTP***: Hypertext Transfer Protocol.
* ***JDBC:*** Java Database Connectivity.
* ***SCP:*** Secure copy.



An ***event destination*** can also be an ***event source simultaneously***. For example, one application could receive an event stream and process it, then transport the processed results as an event stream to another destination. To overcome such a challenge of handling different event sources and destinations,