In [1]:

```python
import requests
import json
from kafka import KafkaProducer, KafkaConsumer
import time
import datetime

# Define function to query weather API
def query_Weather(lat,lon):
    API_key = 'e103266e4fc3a04046ac9ce7d493bc86'
    base_API_request = 'https://api.openweathermap.org/data/2.5/weather?lat=' + lat + '&lon=' + lon + '&appid=' + API_key
    response = requests.get(base_API_request,headers = None)
    if response.status_code == 200:
        weatherData = json.loads(response.content.decode('utf-8'))
        jdata = json.dumps(weatherData).encode('utf-8')
        return jdata
    else:
        return 'Error: ' + str(response.status_code)

# Define Kafka producer to send data from weather API
weather_producer = KafkaProducer(bootstrap_servers='localhost:9092')

# Define consumer
weather_consumer = KafkaConsumer(
    'weather.minneapolis',
    group_id = 'group01',
    bootstrap_servers=['localhost:9092'],
    auto_offset_reset='earliest',
    enable_auto_commit=True,
    value_deserializer = lambda x: json.loads(x.decode('utf-8')))

# Define function to read messages
def read_message(message_IN):
    data_tmp = message_IN.value
    lat = str(data_tmp['coord']['lat'])
    lon = str(data_tmp['coord']['lon'])
    weather = data_tmp['weather'][0]['main']
    temperature = round(data_tmp['main']['temp']-273.15,2)
    utc_timestamp = data_tmp['dt']
    cst_time = str(datetime.datetime.fromtimestamp(utc_timestamp))
    output = 'Message received (at {time}):\nLatitude: {lat}, Longitude: {lon}\nCurrent Weather: {weather}\nTemperature (deg. C): {temperature}\n'
    str_out = output.format(time = cst_time, lat = lat, lon = lon, weather = weather, temperature = temperature)
    print(str_out)
```

In [10]:

```
# Query weather API every two seconds and produce message
# Latitude and longitude of Minneapolis
lat = '44.98'
lon = '-93.25'

# Initialize input variables
start_time = time.time()
query_time = 600 # in seconds
num_messages = 3
count = 0

while count < num_messages:
    weatherData = query_Weather(lat,lon)
    weather_producer.send('weather.minneapolis', weatherData)
    print('Weather data produced: ' + str(count))
    if count != num_messages-1:
        time.sleep(query_time)
    count += 1
```

```
Weather data produced: 0
Weather data produced: 1
Weather data produced: 2
```

In [ ]:

```
# Read messages
message_count = 1
for message in weather_consumer:
    print('Message number: ' + str(message_count))
    read_message(message)
    message_count += 1
```

```
Message number: 1
Message received (at 2022-03-19 17:56:36):
Latitude: 44.98, Longitude: -93.25
Current Weather: Clouds
Temperature (deg. C): 11.54

Message number: 2
Message received (at 2022-03-19 18:18:28):
Latitude: 44.98, Longitude: -93.25
Current Weather: Clouds
Temperature (deg. C): 11.51

Message number: 3
Message received (at 2022-03-19 18:28:29):
Latitude: 44.98, Longitude: -93.25
Current Weather: Clouds
Temperature (deg. C): 11.38

Message number: 4
Message received (at 2022-03-19 18:01:45):
Latitude: 44.98, Longitude: -93.25
Current Weather: Clear
Temperature (deg. C): 11.58

Message number: 5
Message received (at 2022-03-19 18:01:45):
Latitude: 44.98, Longitude: -93.25
Current Weather: Clear
Temperature (deg. C): 11.58

Message number: 6
Message received (at 2022-03-19 17:56:37):
Latitude: 44.98, Longitude: -93.25
Current Weather: Clouds
Temperature (deg. C): 11.54

Message number: 7
Message received (at 2022-03-19 18:01:45):
Latitude: 44.98, Longitude: -93.25
Current Weather: Clear
Temperature (deg. C): 11.58

Message number: 8
Message received (at 2022-03-19 18:01:45):
Latitude: 44.98, Longitude: -93.25
Current Weather: Clear
Temperature (deg. C): 11.58
```