

## Smart systems lab 4 and 5 report

An overview of the kafka system can be seen in figure 1. Below, the kafka system with producers and consumers will be explained.

The flask\_app will act as producer and send the exercise data alone with time stamp as "json-topic" to kafka.

We then use consumer 1 to read the "json-topic", and use the coordinates of certain points of the body to calculate the angle. In our case, we choose to track the left shoulder and left elbow. We calculate the angle between the left arm and the body, and then send the angle with a timestamp back to kafka. An example of the output of consumer 1 can be seen in figure 2.

Consumer 2 will read the "angle-topic", using angle and the time stamp to calculate the angular speed, and we use a simple expert system to check whether the exercises have been executed correctly and generate quality factor ratios. There are 3 different exercises. An example of the output of consumer 2 can be seen in figure 3.

### *Exercise 1: checkExerciseLength()*

This exercise will check whether the arm has been raised within a predefined interval. In state 0, we start the time, and check whether the angle is raised to 90 degrees. If this is the case, in state 1 the final time for raising the arm will be calculated. If the final time is within the predefined interval, a positive quality score will be given. If it is not the case, a negative quality score will be given. The quality scores will be combined to a ratio, by appending them as strings in the following format: positiveQuality1:negativeQuality1.

### *Exercise 2: checkNumberOfExercises()*

This exercise will check whether the exercise has been executed x number of times within a predefined interval. In state 0, we will start the time. In state 1, we check if the arm has been raised to 90 degrees, and increase a counter if this is the case. Then we go to state 3, where we check whether the counter has reached the desired value. If so, we stop the time, and calculate how long it took for the exercise to be executed. If the exercises have been done within predefined time, positive quality is increased.

### *Exercise 3: checkSpeedUpmovement()*

This exercise will check whether the arm is moved upwards (from 90 to 180 degrees), with a good angular speed. First, we check whether the angle of 90 degrees has been reached. If so, we check whether the angular speed is high enough. If so, the positive quality will be increased.

Consumer 2 then sends the quality factor ratios to kafka for other consumers to use.

Consumer 3 reads the "quality-topic", it calculates the ratio of good exercise vs bad exercise in a certain time interval, saves this in a local database, and also sends this to kafka. An example of the output of consumer 3 can be seen in figure 4.

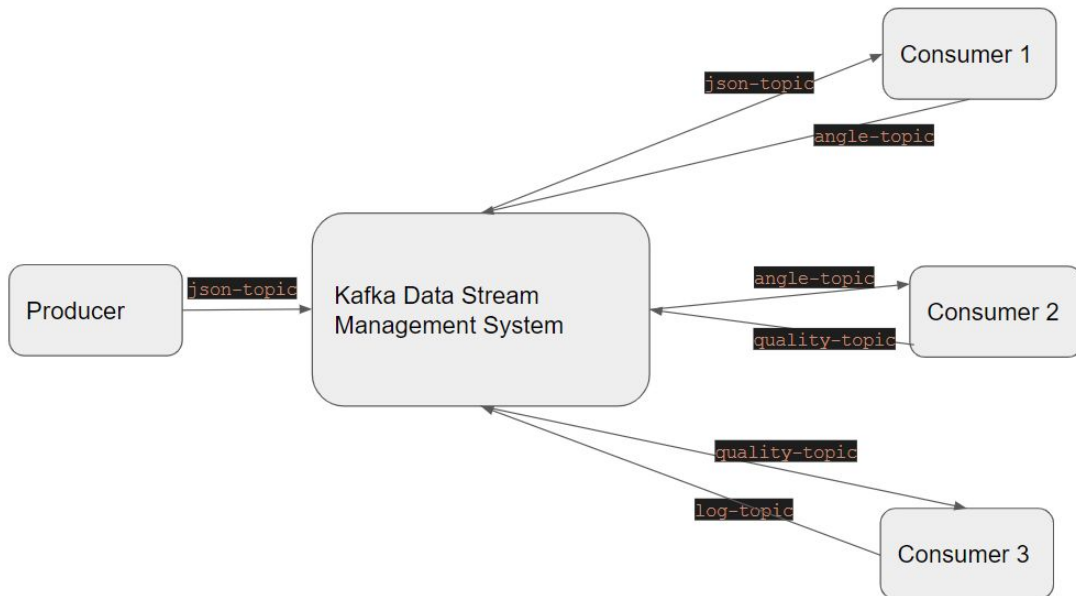


Figure 1: Overview of kafka data management system

Anaconda Prompt (Anaconda3) - "C:\Users\Samir\Documents\smart-systems-lab-4-and-5-group-3-mei-tao-samir-kenny-master\consumer1\_left\_arm\_angle.py"

```

left arm angle 126.18444343010057 degrees
timestamp 1590782581729

left arm angle 62.9332929879862 degrees
timestamp 1590782582029
  
```

Figure 2: Output of consumer 1

Anaconda Prompt (Anaconda3) - "C:\Users\Samir\Documents\smart-systems-lab-4-and-5-group-3-mei-tao-samir-kenny-master\consumer2\_left\_arm\_angular\_speed.py"

```

angle 53.33791470156255
neg2: 0
pos2: 6
neg1: 15
pos1: 5
neg3: 198
pos3: 24
  
```

Figure 3: Output of consumer 2

Anaconda Prompt (Anaconda3) - "C:\Users\Samir\Documents\smart-systems-lab-4-and-5-group-3-mei-tao-samir-kenny-master\consumer3+calculate\_quality\_all\_exercises.py"

```

1590782575698
time to store data
2020-05-29 22:02:55.944229 ex1 quality ratio 1: 4:13 ex2 quality ratio 2: 5:0 ex3 quality ratio 3: 0:23
data is logged!
1590782578499
time to store data
2020-05-29 22:02:58.731870 ex1 quality ratio 1: 5:13 ex2 quality ratio 2: 6:0 ex3 quality ratio 3: 0:23
  
```

Figure 4: Output of consumer 3