**MovieFlix case study**

MovieFlix is a company that allows us to watch TV shows and Movies on demand.

**Problem:**

MovieFlix business wants the following capabilities:

1. Make sure the user can resume the video where they left it off.
2. Build a user profile in real time.
3. Recommend the next show to the user in the real time.
4. Store all the data in the analytics store.

**Solution using Kafka:**.



MovieFlix decided to implement the above functionalities using Kafka. Kafka was in the middle. Topic show-position was used to tell how far a user has proceeded in a TV show even in a video. So there must be producer for the show\_position. There is video player in customer’s browser while playing the video once in a while it will send data to video position service which is a producer and it will send the data of video’s position to show\_position topic. Video position service will make sure that it is sending correct information to the topic.

There is a Resuming service as consumer which can be basically a database which fetches the position of videos for each user and stores in database. It will only store the last position of the video.

When a person starts a video player , it will call the resuming service to get the last position of the video and start from that position.

For recommendations , there will be a recommendation topic which will store data about the shows which user has watched and how far. It will help to provide recommendations to the users about shows. There is a Recommendation Engine which is nothing but kafka streams that fetches data from show\_positions topic and perform some real time algorithm on the video positions and provide the recommendations in the recommendation topic.

Recommendation service is a consumer which fetches records from recommendation topic and whenever a person goes to the MovieFlix portal , it will provide recommendations for different shows which the user may like.

Also to store all the data from show\_positin and recommendation topic there is a Analytics consumer which is a kafka connect and it stores data in a analytics store in Hadoop.

Show\_position Topic:

This topic can have multiple producers from all over the world

Should be highly distributed if high volume > 30 partitions.

User\_id can be the key as it will help to store data from each user to store in order.

recommendations Topic:

The kafka streams recommendation engine may source data fro the analytical store for historical training

It will be a low volume topic.

User\_id can be the key and the no of partitions will be less.