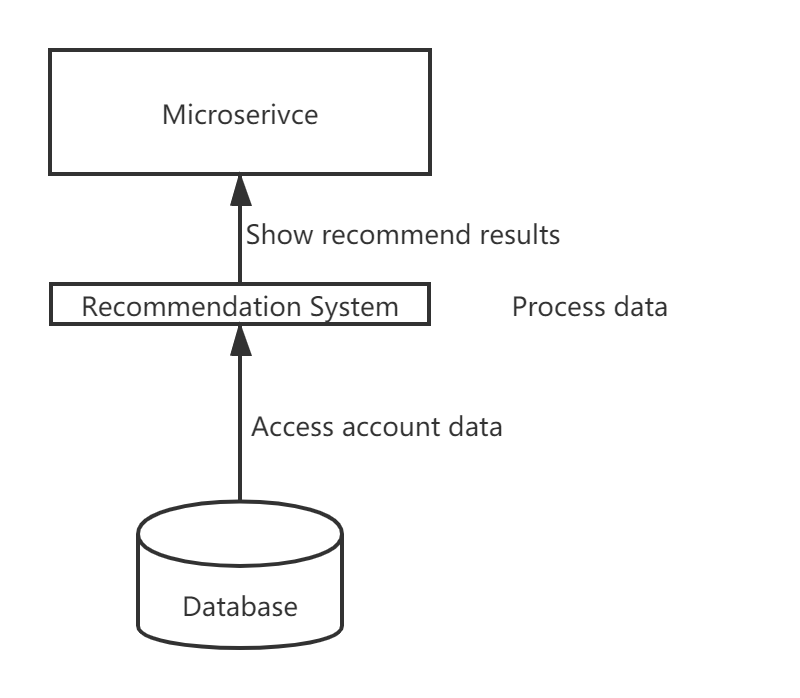
Recommendation System

1. **Structure:**
2. **Strategy:**

There are three different strategies commonly used in Collaborative Filtering. They are user-based, model-based and item-based.

**User-based:**

User-based collaborative filtering needs to find the similarity relationship between users and users online. The computational complexity is higher than that of item-based collaborative filtering. It is often used in large projects.

**Item-based:**

For item-based collaborative filtering, the similarity of the items under consideration will not change for a period, so it can be easily calculated offline, and the accuracy is generally acceptable. Also, most small projects like to use item-based filtering.

Model-based filtering is more complex than these two. So I didn’t take it for consideration.

For our project, I think we can use the item-based strategy. The number of the hobbies is much smaller than the number of users. Also, the recommend results are based on user’s historical behavior, which will make users more convinced. What’s more, our initial data can’t be very large, the item-based strategy is very suitable for our condition.

1. **Process:**

* Since our goal is to recommend similar hobbies for users, we need to set up a matrix for user and hobbies.
* Then, we need to construct the co-occurrence matrix of hobbies.
* Calculate the similarity between hobbies, which means we need to calculate the similarity matrix
* Finally, we can recommend hobbies to users based on their records and histories.

1. **Data:**

We need to read all the user’s information. We need the user’s **hobbies**, the **searching history**, the **browsing history**, the **attended events** and the **comments**.

For the calculation of the similarity matrix, we need to settle down different coefficient for different variables. For example, we can set the degree of interest of each hobby to 1. Then for the searching history, we can set it to 0.25, the browsing history we can set it to 0.5, the attended events we can set it to 0.75. Since we can rate the events, we need to take the ratings into consideration.

1. **Output:**

After the calculation, we need to store the similarity results. Then, when the users click on an event, the service can read the results and choose hobbies to recommend to the users based on the similarity results.