Questions:

1. What do you mean by rating rating?
2. I think it is difficult to make a homogenization without having a prioritization of sources. Is it something to define a posteriori?

Definition:

With the structure that is proposed having n providers and that the sources are diverse and asynchronous, what is proposed are:

- we would have **n** grafters for each supplier and type of source so that each source can be processed independently.

Once processed, it will be sent to a queue where a time m will be saved. Several consumers will have to be implemented to send partially homogenized information to a database and the rest will go to a datalake.

To have control to enable or disable functionalities as well as general configurations, it would be necessary to have a configuration system, where it tells us movies are already processed or to be processed (for this, it would be required to put in the created_at and update_at tables in timestamp_z or unitxtimestap) or the confidence of Providers.

Platform provides

By having the classifications calculated in different ways, it would be necessary to homogenize the system or weight the results, thus having a system that is easier to consume and standardize the result. (Standard score “Puntajes Z“)

And it would be the same in the genders, it would be necessary to decide how to homogenize the categorization and for this the company's genders could be used since a movie or series can only have the rights to a provider, if desired it could be saved as an SCD type 2.

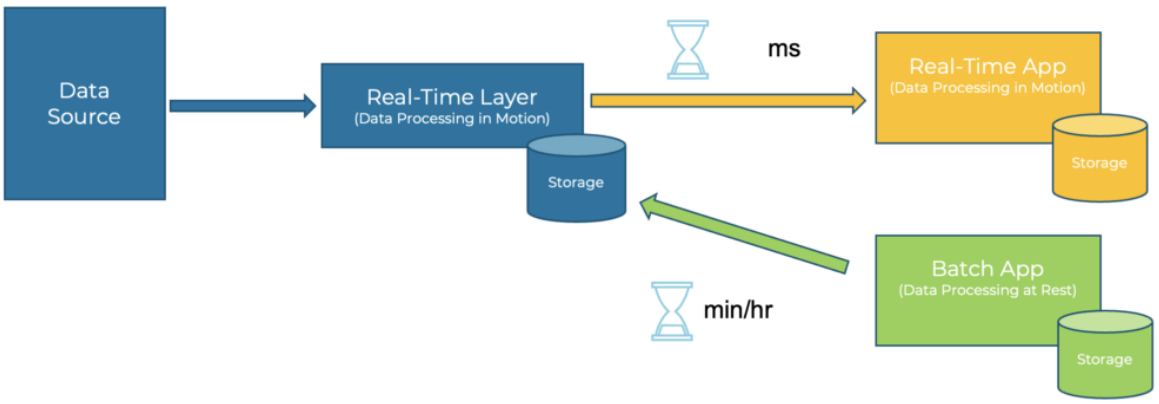
As I mentioned before, the trust of each provider should be saved by default, and the **C-Rating** for standard users.

Api

I would go for a classic approach where it would have a differentiation to know if the user is premium or not. On the database, there would be a Redis or Memcached type cache where the results are saved so as not to overload the database.

We have two different methods one of them for a standard user and other for a premium via key, maybe thinking about the rate limit for a standard user

Backend



Api