Similaritipsum

Useful open materials:

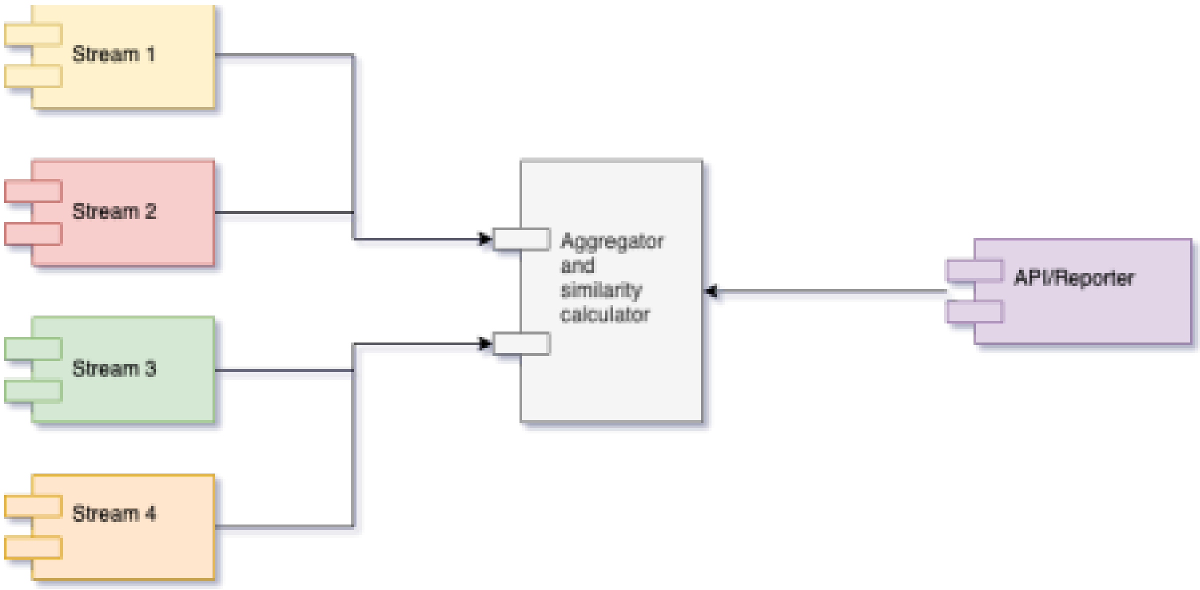
- https://www.lipsum.com/
- https://baconipsum.com/json-api/
- https://pirateipsum.me
- https://www.shopify.com/partners/blog/79940998-15-funny-lorem-ipsum-generators-to-shake-up-your-design-mockups

Task A

Your task is to implement streaming service of lorem ipsum words that will inspect similarity between texts, for instance, the ratio of bacon references to pirate references (aka, words with more than two r’s) or make skewers (map words to skewer format where you get something similar to shish-kebab = *jalapeno* == ***---{[]S[]([]o[]--***) from words and see who has greater skewer average.

For every **n** words from a set of streams, compare them in a free form way (it’s up to you to decide on the similarity algorithm).

Proposed data flow is in following format:



There are no other general requirements, so you can implement this basic data processing, or you can go wild as far as you want.

Requirements

There are no restrictions. You can use as many existing technologies as you find fit and there are no platform requirements.

The point is to showcase your knowledge, but don’t go too deep: write all assumptions in your code (for some specific cases, it is ok to go with the simple path and write that assumption in a code block).

Good places to showcase your knowledge:

- *Good installation documentation*
- *Docker setup*
- *Tests*
- *Good scalability*
- *Clean VCS history*
- *Having VCS*
- *Strict typing in dynamic languages*
- *API documentation*
- *Robustness of input and output parameters*
- *Microservices or some similar approach*
- *CLEAN, layer architecture, DDD*
- *General simplicity*

There is no time limit, so take your time. For any questions, you can contact me in any way possible.