Defining inputs for three microservices: T00

* + **API scraping service**: This tool obtains product information from eBay and sends it to a preprocessor to be placed in a database.
    - Inputs:
      1. **Developer Parameters**: ( type = strings )  
         There will be a few tweakable parameters in the API scraping service to allow for simple and robust data scraping.
      2. **Raw Seller Data**: ( type = tuple(strings) )  
         This data will be returned to the scraping service based on format that the API calls return.
    - Outputs:
      1. **Refined Seller Data**: ( type = (string, string) )  
         List of string pairs describing products. Formatted as (description, item name) pairs.

* + **Database preprocessing service**: This tool obtains lists of product information from API scraper and organizes them for model training/lookup.
    - Input:
      1. **Refined Seller Data**: ( type = (string, string) )  
         List of string pairs describing products. Formatted as (description, item name) pairs.
    - Output:
      1. **Training Files**: ( type = json )  
         json files designed for a model training scheme.
      2. **Database Push Request**: (type = string )  
         Requests to push the json file to a database.

* + **RAG model service**: This tool obtains queries from a user and consults a database before returning a prompt
    - Inputs:
      1. **Image**: ( type = tensor )  
         This is given by the user as their only input. Its processed by an image classifier, and based on that classification a relevant query prompt is generated and sent to the a RAG model.
      2. **Training Data**: ( type = dict(str) )  
         The RAG model receives data from the database based on its prompt and its lookup request. It then trains on this data.

* + Outputs:
    - **Database Lookup**: ( type = string )  
      Requests data from the database to train on based on its prompt.
    - **Description of User Image**: ( type = string )  
      Ideally the description of the users item in the most attractive style to a potential buyer.

API Scraper 
- Code to Prompt eBay for seller data. 
- Output = sends data to database 
Run Parameters 
Developer Selected Parameters 
-> Type: String or Integers 
Seller Data 
- Type: String Pairs (description, label) 
Seller Data 
- Type: String Pairs (descript' 
eBay, 
Input Imaqe 
- Image of item 
reseller wants to 
post. 
n, label) 
Search Code 
— Type: Program 
Item Description Generation 
- Input = Description of 
the users item in the most attractive 
style to a potential buyer. 
- Output = The description to be 
fed to a front end task (that will 
display the generated description). 
— In 
Org 
be 
trait 
Typ 
-01 
trait 
Database Look 
R.A.G Service 
Image Classification 
Generative Output 
R.A.G Description ( 