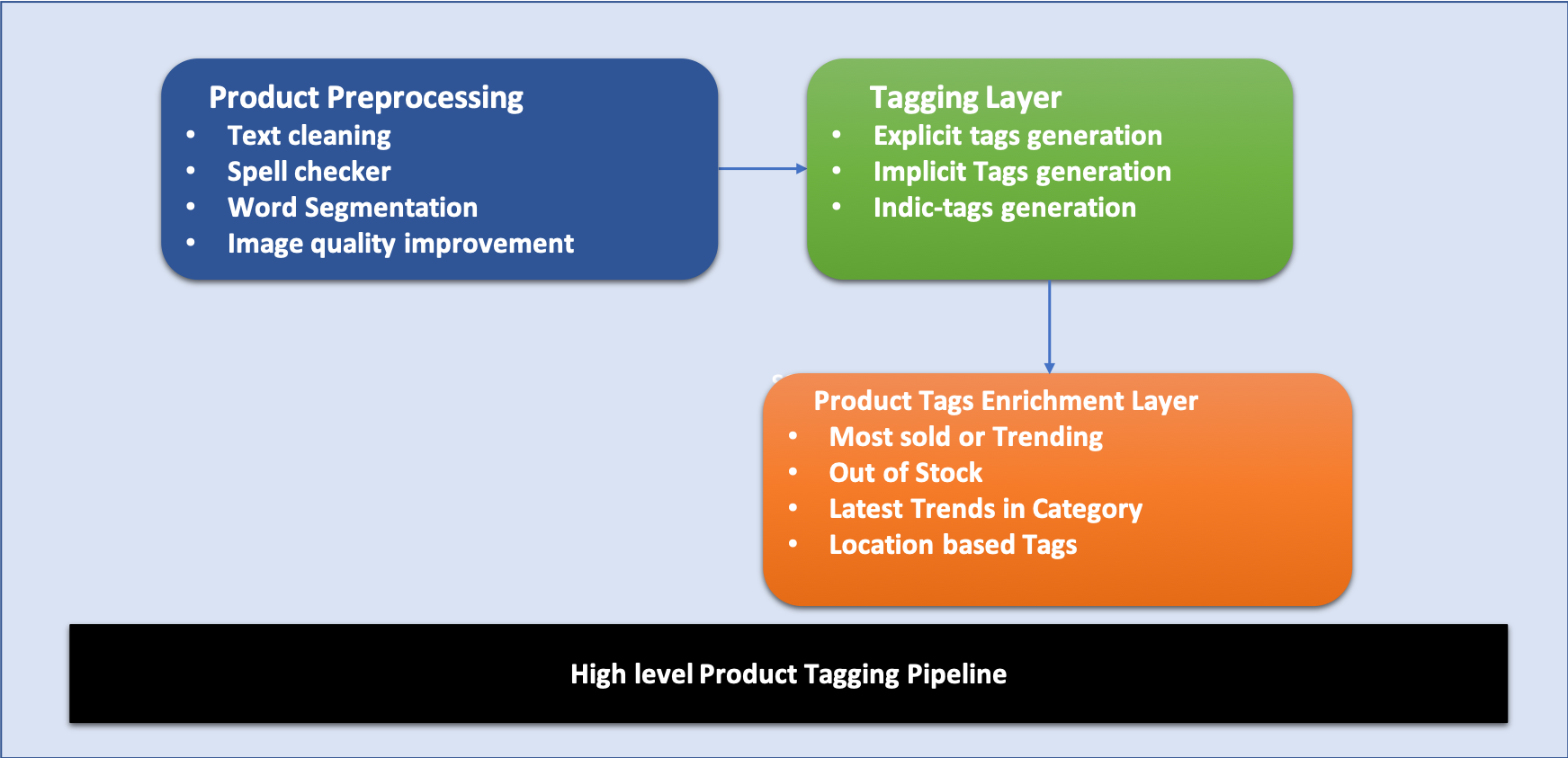
# **Product Tagging Pipeline**



## **Product Preprocessing**

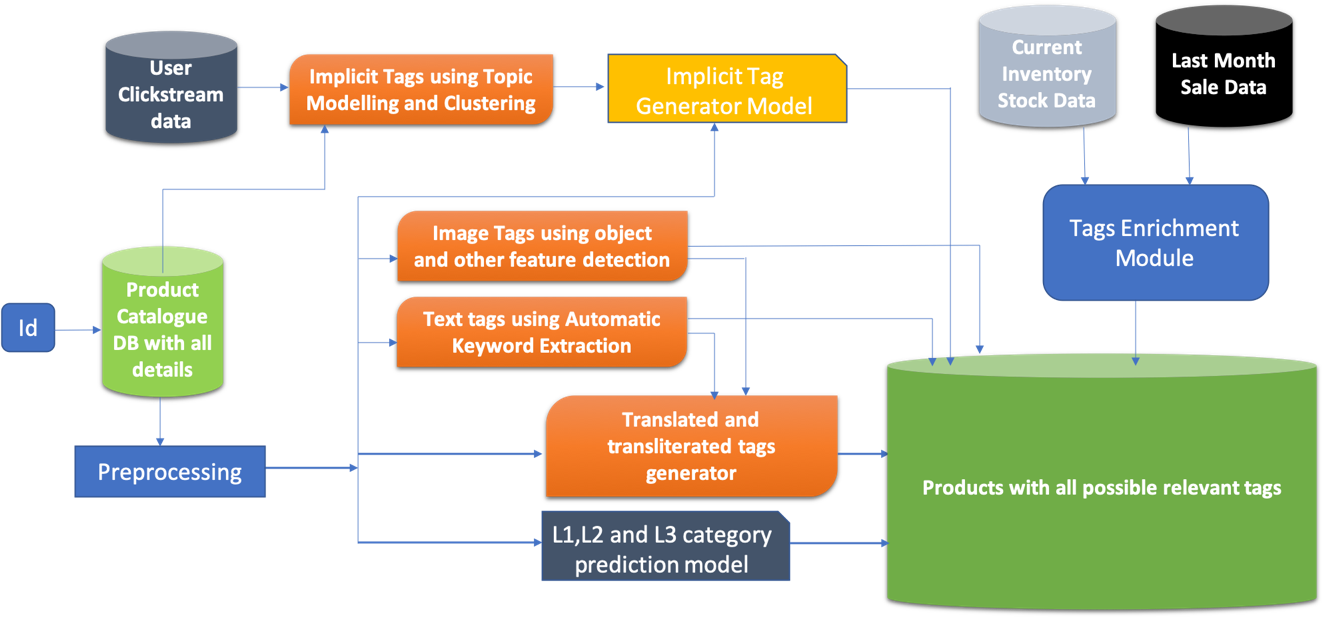
Preprocessing of products can be done by using preprocessing module of query preprocessing service.

* Text Cleaning
* Word Segmentation
* Spell Checker
* Image Quality Improvement

## **Tagging Layer**

* Explicit Tags Generation
  + Textual Tags (Tags extracted from product textual content)
  + Image Tags (Tags extracted from objects and other features in Image)
  + Other features like Brand, Color, Type, Material, Quantity etc.
  + L1, L2 and L3 categories of the product.
* Implicit Tags Generation
  + Thematic or occasional tags.
    - Ex: “**beachwear “or “winter clothes” or “wedding wear” in fashion**
    - **“leafy vegetables” or “root vegetables” in grocery**
* Indic-Tags Generation
  + Translated and Transliterated Tags.

## **Overall Product Tag Generation Flow**



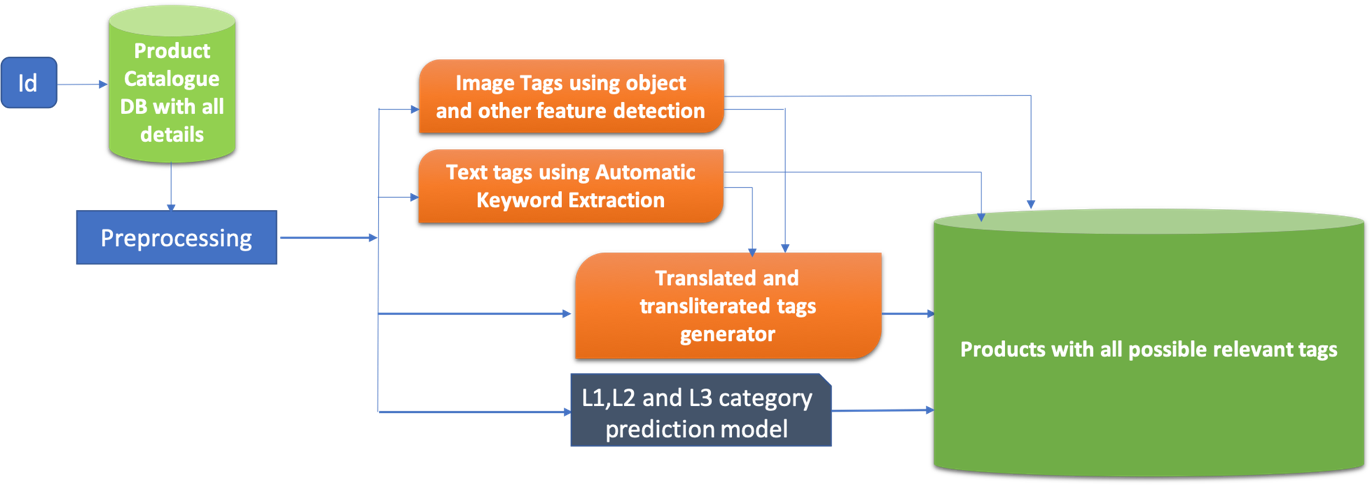
## **Explicit Tag Generator**

For product catalog, this will be a one-time process and hence this service can be run as an offline service or can be exposed as an API to be used for other services.

This service will generate 4 different type of tags as follows:

* Text tags from product description
* Image tags
* Indic-tags (translated and transliterated tags)
* L1, L2 and L3 categories.

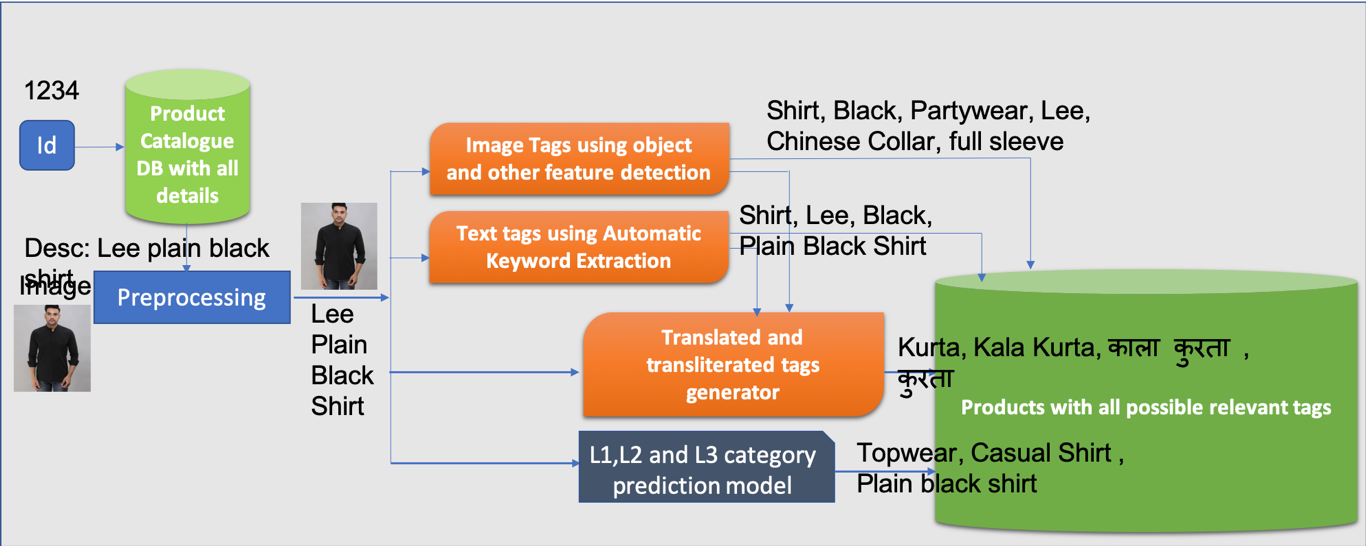
Below is the flow diagram of explicit tag generator.



For generating explicit tags, we need product description and image corresponding to product Id.

* Pass Product text to keyword Extraction service and product image to image detection service.
* Keywords generated from above step will be passed to translate and transliterate the translated keywords to form Indic-aware keywords.
* Current l1, l2 and l3 categories will also be a part of explicit tags.

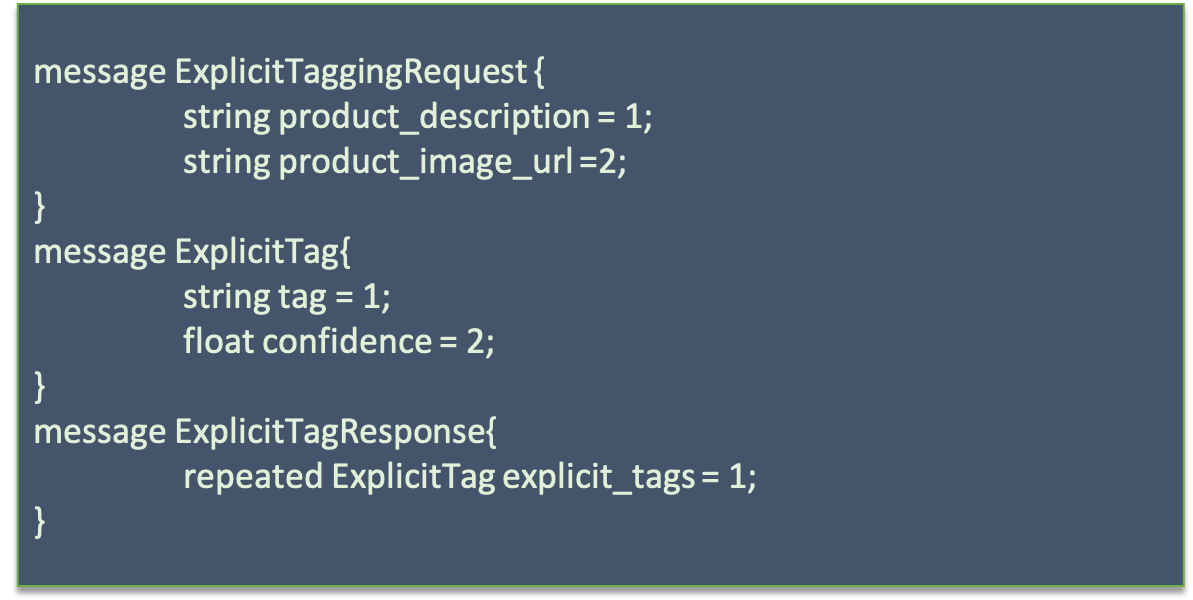
Consider an example,



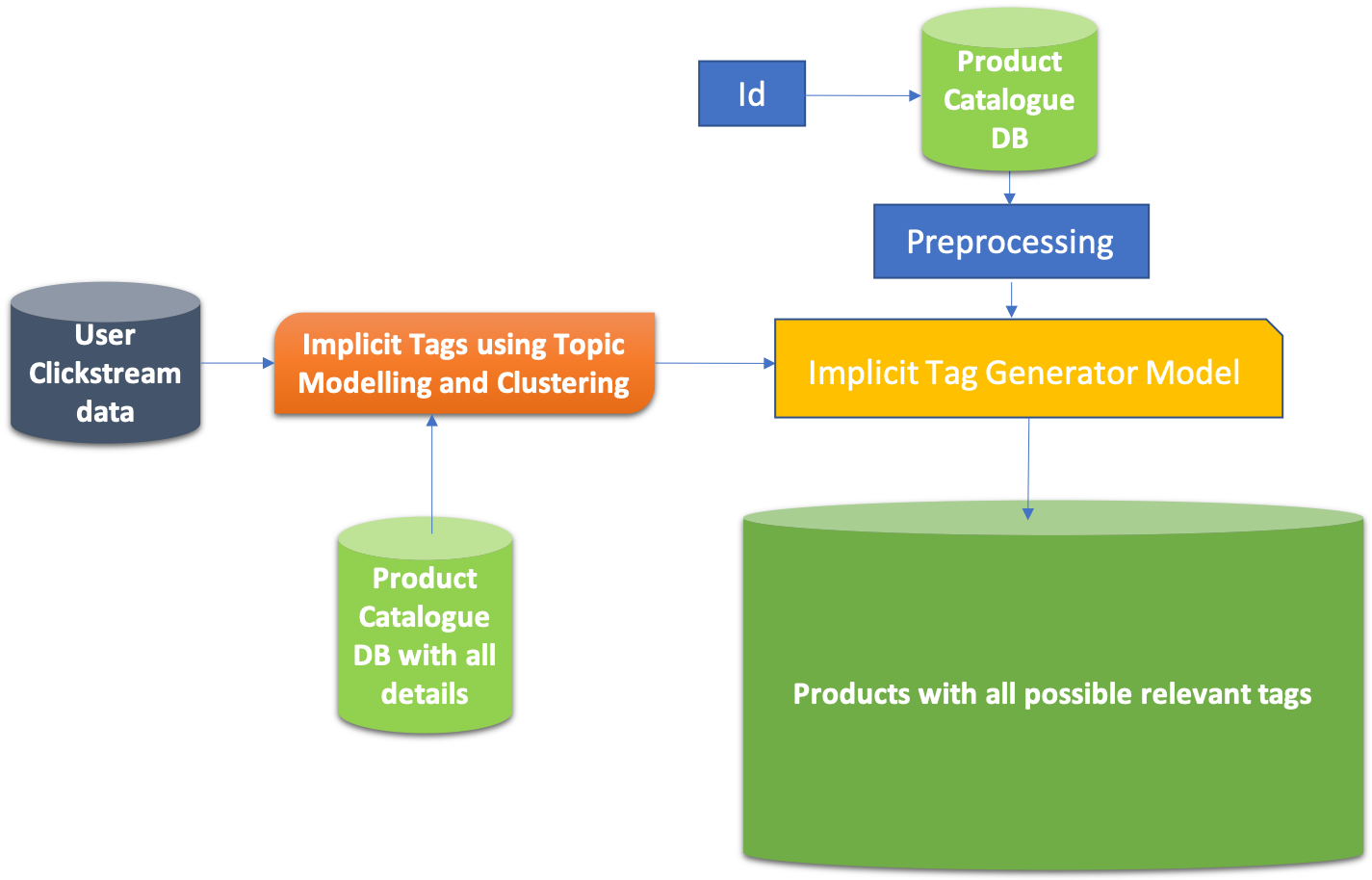
**Service**



**Proto**

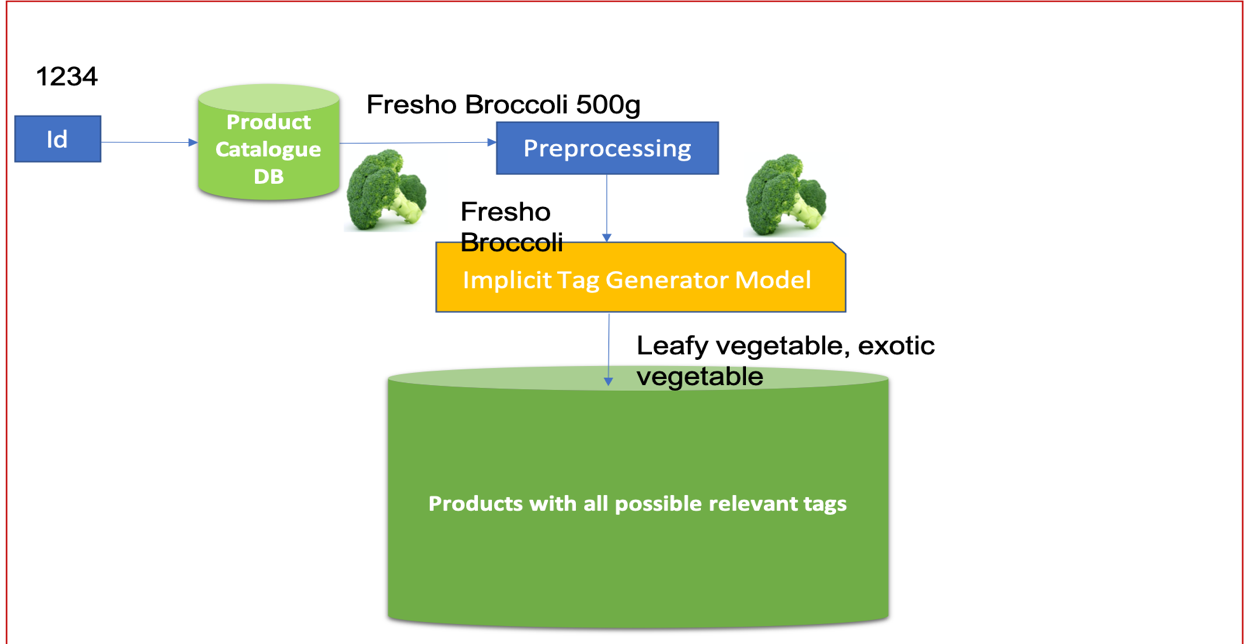


## **Implicit Tags Generator**



* Implicit tags are generated when products are grouped or clustered along different dimension. They can be occasional or thematic in nature.
* Implicit tags generation model can be trained using clustering after computation of embeddings of each product base on the click data of each session.
* Example: Jacket -> Winterwear
* This model can be updated in batch process using weekly or monthly cronjob.

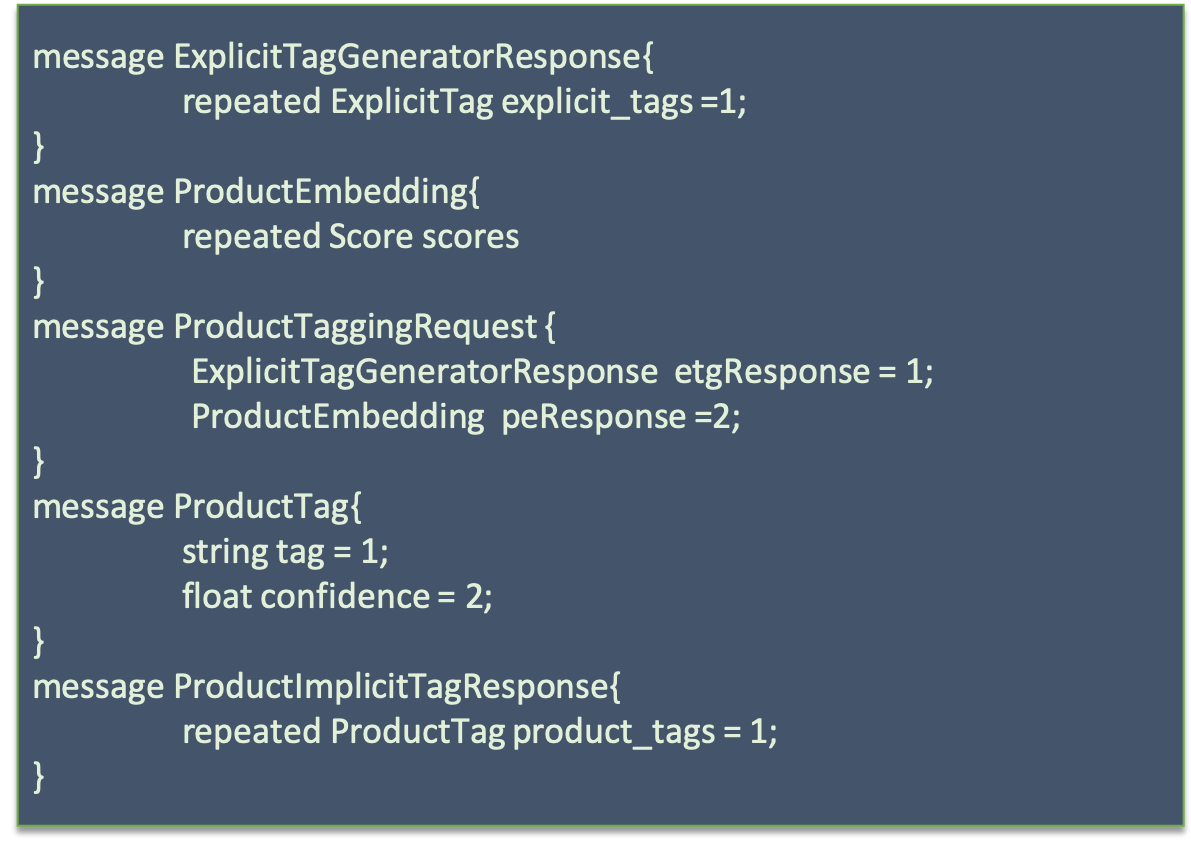
Consider an example,



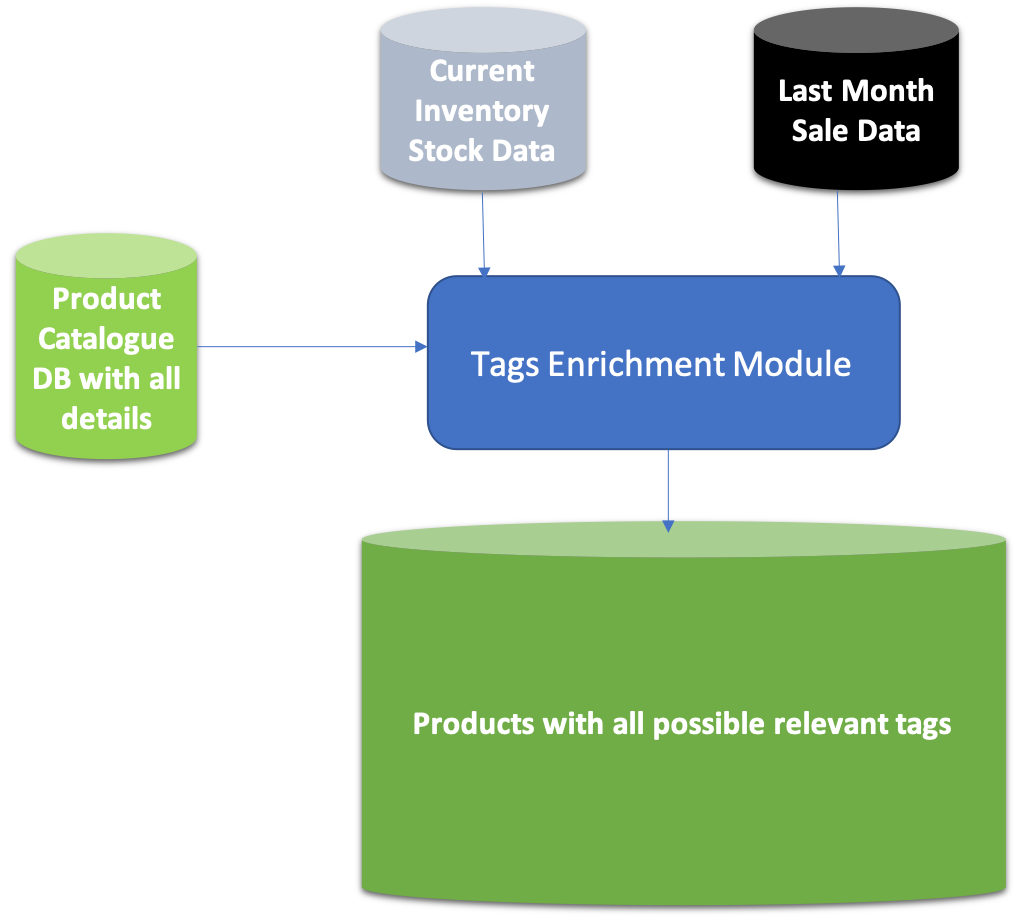
**Service**



**Proto**



## **Enrichment Layer**

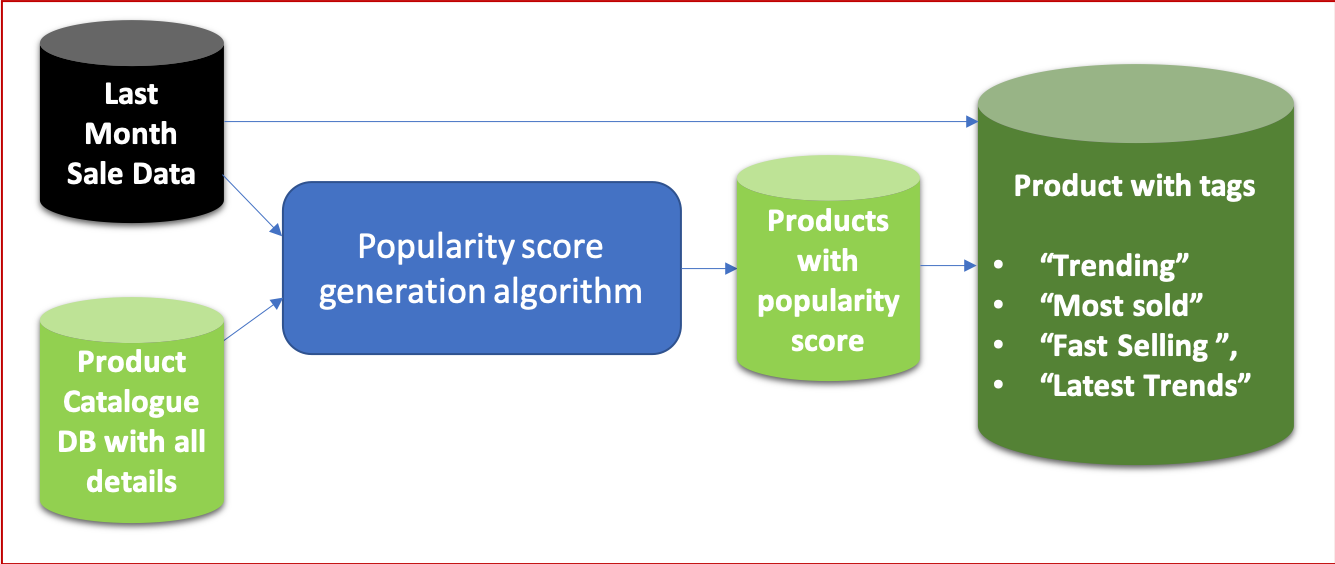


The enrichment layer tags generation will be daily offline process which will be based on the sale data, product catalog and inventory stock.

Broadly these tags can be classified in two categories:

* **Popularity Based**
* **Inventory Based**

# Popularity Based



* These tags will be updated **daily/ twice a week** depending upon the data.
* Based on the sale data, **popularity score** for each product will be calculated and with other sale details, different tags will be assigned to the products.

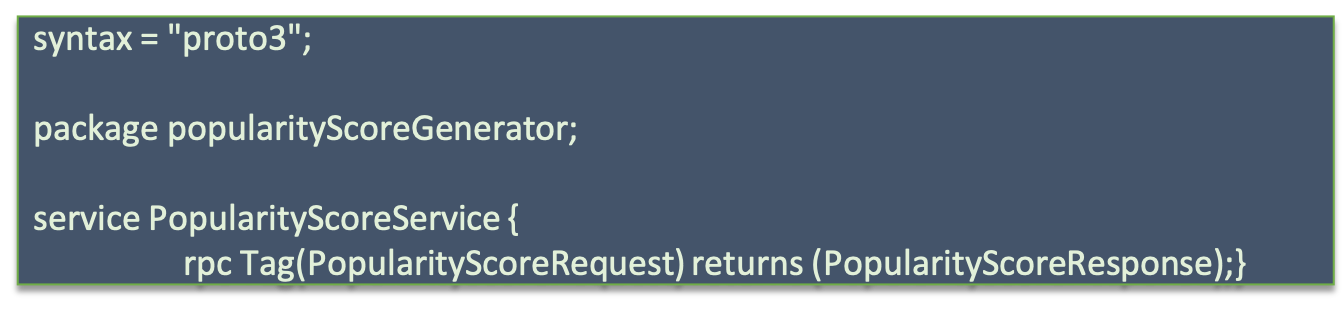
Common tags:

* **Most popular/Trending** products in category/overall. For example, “Plain sports Nike Shoes” sale rate increased exponentially can be marked “Most Popular”.
* **Fast selling** products (can be finalized based on the rate of sale).

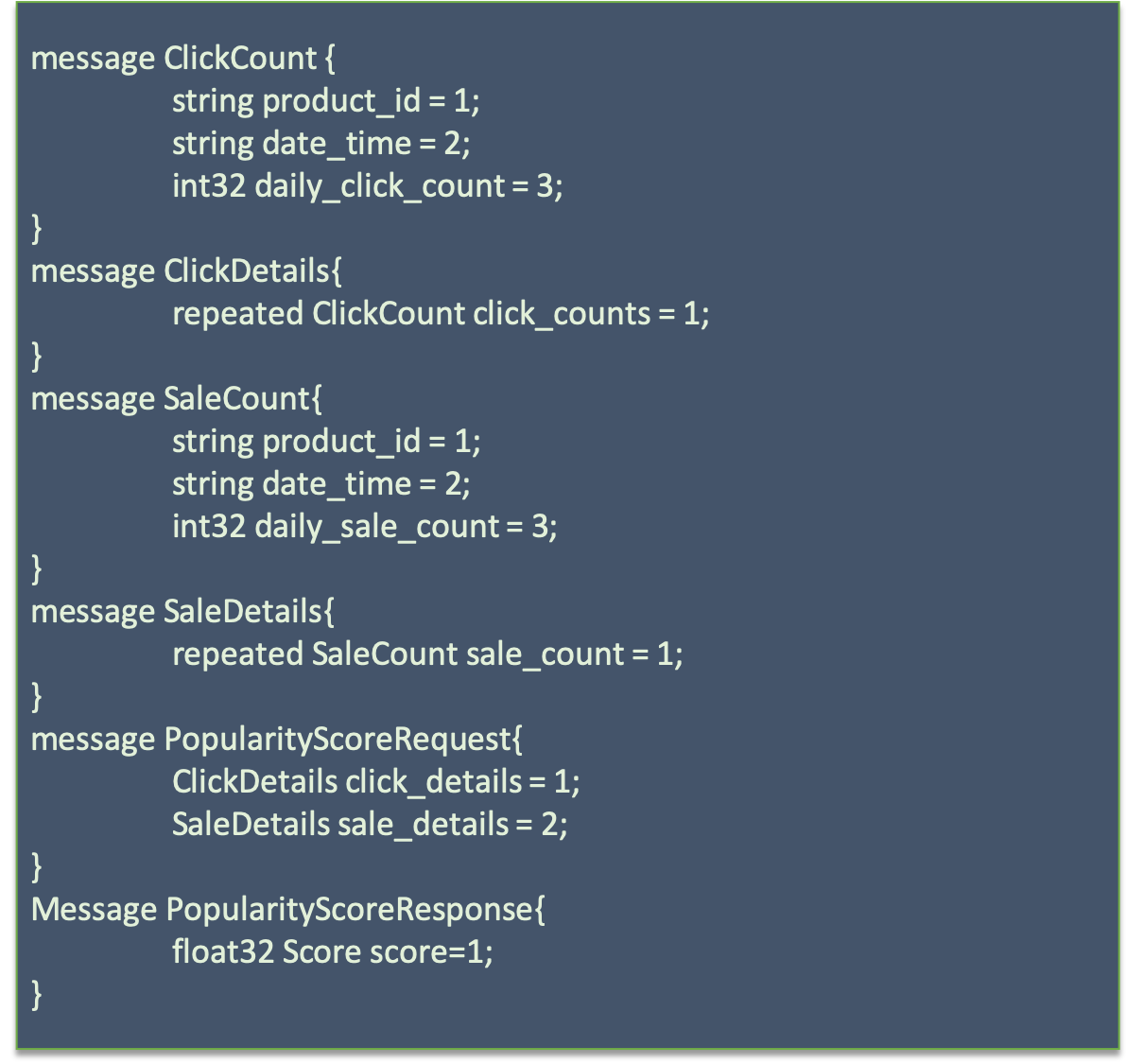
“Ipad pro” whose sale was exponential recently can be marked as “Fast Selling”.

* **Most sold** products in category – Mainly based on the sale volume of products
* **Latest Trends**
* Mainly for fashion e-commerce.
* Will be given to products of some specific types whose CTR/conversion is maximum in last month (can be changed based on sale volatility).

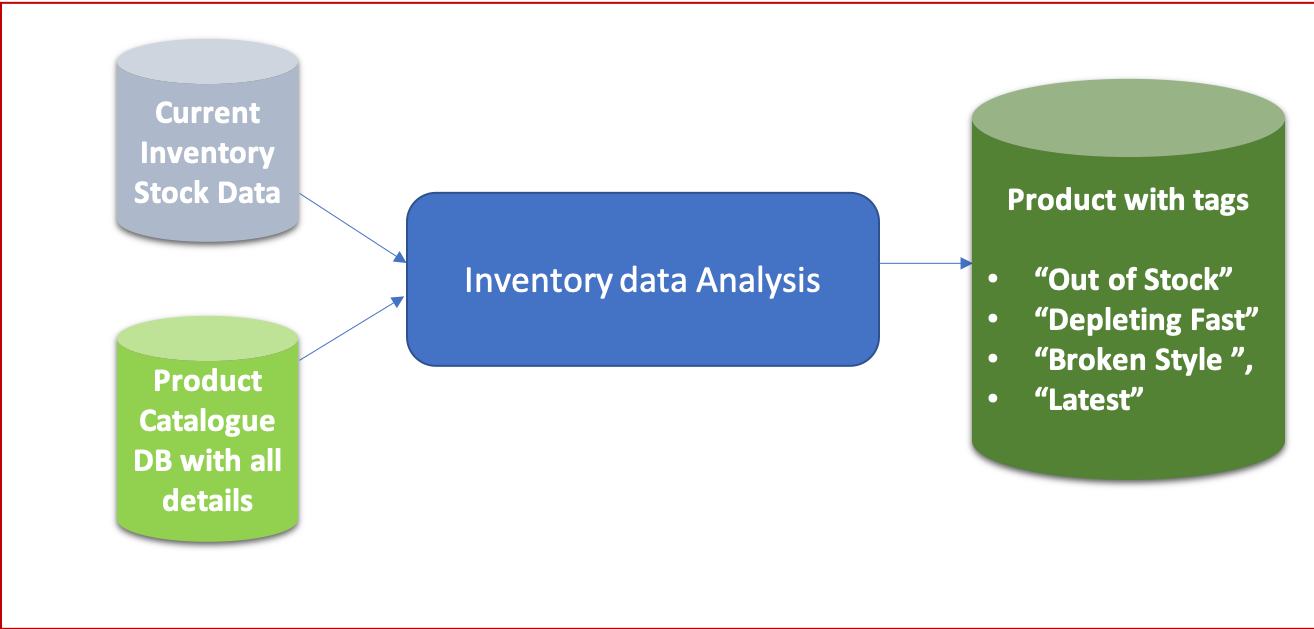
**Service**



**Proto**



# Inventory Based



* **Out of stock**.

All the products having stock count 0 will be tagged as “Out of Stock”.

* **Depleting Fast**

Products whose rate of sale will be more than a threshold and whose stock count will be less than a threshold number can be tagged as “Depleting Fast”.

* **Broken Style (Fashion Ecommerce)**

Products whose stock count is 0 for one or more sizes but is available in some other size can be tagged as “Broken style”.

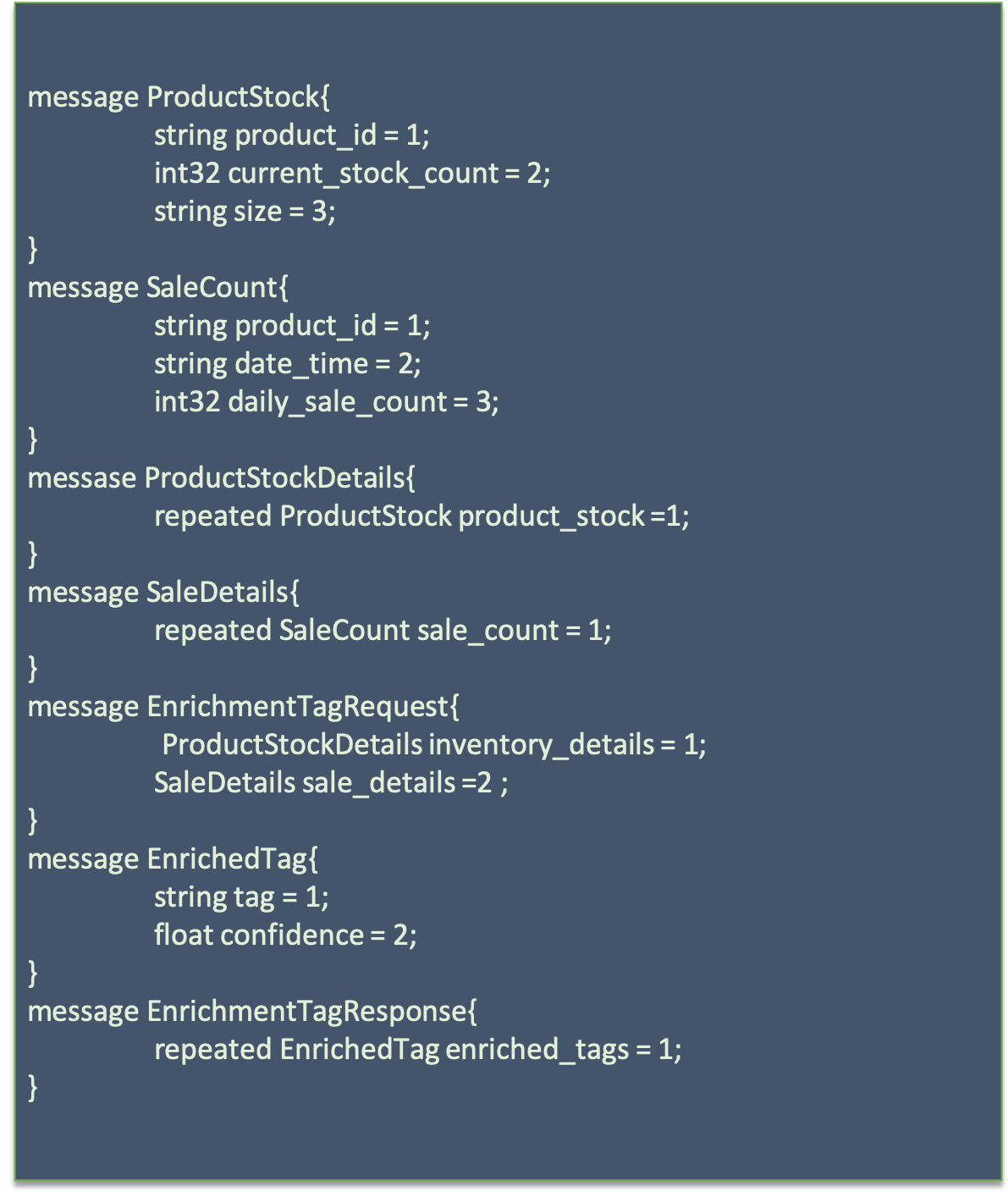
* **Latest products in category**

Most recent in inventory for any category can be marked as “Latest”.

**Service**



**Proto**



Other possible tags can be:

* Enrichment based on other product attributes
  + Ex: For fashion e-commerce, model used for product display.
  + Ex: Brand ambassadors of the product/brand .
* Location data can be used to filter out/rank products based on other user’s trend from the location.
* Attributes level Synonyms- color similarity.

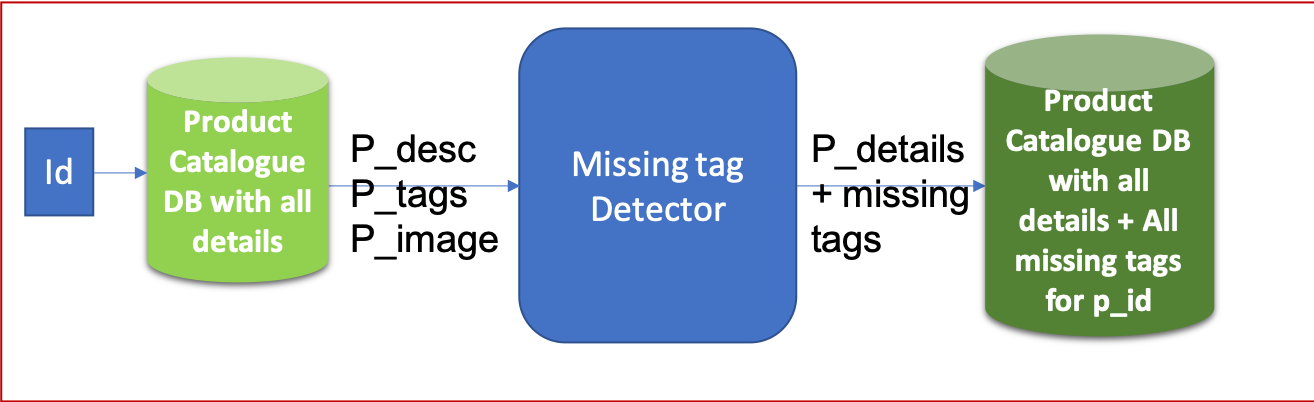
# **Document level Problem Statements**

## Defining Searchable fields and their order

* Identification of Searchable fields and the order in which search should happen for a particular query.
* Mostly searchable fields will be identified based on the type of queries or query analysis and every product attribute.
* From product perspective, all the categories level I.e. L1, L2 and L3 should be searchable as after query category prediction, the product level filtering will happen on the basis of query categories and product categories matching.

## Missing Tags

* Identify any missing tag corresponding to any product id and then add them to make them more searchable.
* For any product, some tags can be missing and they can be any type of tags like thematic tags such as beach wear or any enriched tags like most sold or any Indic tags or can be normal product tags.



## Discrepancy detection in catalog

* Identification of any kind of discrepancy in entire catalog like wrong tag or wrong description or wrong categories (l1, l2 or l3) or maybe wrong image corresponding to a given description for any product.
* For each product, there should be a validation algorithm/process which can check if all the data provided is correct/valid.
* Possible kinds of discrepancies which can occur are wrong description for a text provided corresponding to different images and tags or a wrong image can be there for any text/product description provided.
* Discrepancy can be also in Categories, name or any tags.
* Consider these examples -
  + - “Fruits” category tagged for “Orange Juice”
    - “Shoe image” for a product “Half sleeve shirt”
    - “RedTape brand name” for “Nike shoes”.

## 

## Outlier /Fraud Detection

* It’s similar to what we have already discussed in last point. It’s kind of identification of any outlier or any kind of fraud within a category of products.
* This is discrepancy detection with respect to other products of same group rather than their own description.
* Consider these examples-
  + - Price of a **Nike** shoe is “**Rs 10**”.
    - Discount of **99%** on **Blackberry** formals.

## **Evaluation Metrics for Product Tagging Module:**

### Offline Metrics

* F1 score
* Recall

### Online Metrics

* Click through rate
* Revenue per session.

## **Data Requirements for Product Tagging:**

### **Product Sale Data:**

* This can be used to identify the trending, most popular and latest trends products.

### **Inventory Stock Data:**

This can be used to identify the out of stock items or very less available items.

Daily inventory logs will be better to identify the rate of sale also.

### **Session Data**

* For all the queries searched on all search platforms, we need to track clicks, carts and orders events along with the session\_id, user\_id and event\_id

### Sample data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User\_Id** | **Session\_Id** | **Event\_Id** | **Query** | **ProductId** |
| user\_1 | session\_1 | search | rice | NA |
| user\_1 | session\_1 | click | rice | 87985756 |
| user\_1 | session\_1 | cart | rice | 87985756 |
| user\_1 | session\_1 | order | rice | 87985756 |
| user\_1 | session\_2 | search | sugar | 23423342 |

* **User\_Id** is a unique Id associated with a user
* **Session\_Id** isalsoa uniqueId associated with a session on the website
  + Session Id is changed whenever user fires a new query, this will help in tracking different queries in different sessions
* **Event\_Id** is set of different events (search, click, cart, and order) that defines the user’s action on the site
* **Query** is the search query typed by the user, this will remain same for a unique session\_Id
* **ProductId** is a unique Id associated with the product that was interacted with (clicked, added to cart, ordered) by the user in a particular session\_Id and event\_Id

Session/clickstream data will be useful in generating thematic/implicit tags.