# Prodigy

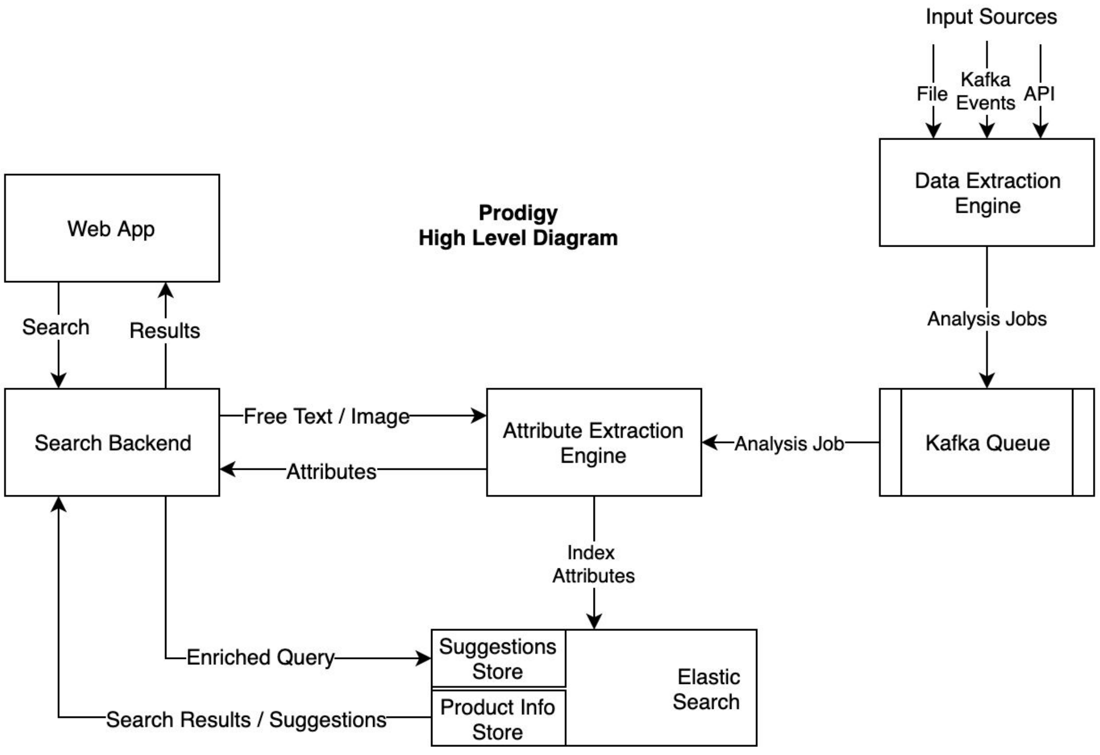
We Understand You

Authors:  Devendranadh, Sai Ravi Teja, Swayam, Venkatesh

## Summary

This document outlines the technical design of product discovery and ontology engine (prodigy). It lists 5 high-level components, lists their basic responsibilities and technologies being used to build them. It also includes below high-level diagram outlining interaction patterns between the components.

## High-Level Diagram



## Abstract

In a standard search application, backend services usually communicate with elastic search to get indexed search results. But, in our approach we have introduced another component call Attribute Extraction Engine, which extracts attributes from the query and enriches the query with this information. This Attribute Extraction Engine also acts as indexer for raw text provided via scrapped jobs.

We have made each component data agnostic i.e. we aim to accept both image and text as search input from users and Attribute Extraction can happen on both images and textual data. After the attributes are extracted Search Backend uses this information to search in elastic search.

Our design is highly scalable at each component making overall system robust.

## Frontend Web Application

> Intro:

User Interface for users to query dumb strings and expect highly intelligent results which would maximise the impact of a product.

> Tech Stack:

- VueJs (A progressive JS framework)

## Search Backend Service

> Intro:

Search Backend Service **(SBS)** glues front end to Tag Attribution Engine **(TAE)** and Data Storage Engine. It orchestrates these services to fulfil a search or suggest request from the frontend. It’s a simple spring boot java application.

> Tech Stack:

- Java Spring Web Framework

## Tag Attribution Engine

> Intro:

The Attribute Extraction Engine **(TAE)** enables the system to churn data into meaningful / structured attributes. This acts as a common component to both Search Backend and Data Extraction Engine, thus centralizing the logic of attribute extraction.

There are several components in the **TAE** service, which extract attributes from each type of data. For textual data we are using NLTK NLP library. For images we are using Google OCR api (or anything similar to that) to extract textual information in the images. Then this textual information can be passed through 'text analyzer' to extract attributes. Apart from text data, **TAE** can also extract attributes from detected objects in the image using Neural Nets. Below is a comprehensive list technologies used in **TAE**.

> Tech Stack:

- Python Flask as the micro web framework for our REST APIs
- NLTK for natural language analysis
- Google OCR for extracting text in images
- Neural Net usage yet to be decided

## Data Extraction Engine (DEE)

> Intro:

The Data Extraction Engine can be thought of **"noise filter"**. This service acts on the multiple input sources, filters each incoming data object and pushes the actual usable data for further processing in the pipeline.

The **DEE** is implemented as a **"filter-chain"** or more popular known as "chain of responsibility". Each node in the chain is a filter looking for a specific corrupt data and neglecting the particular feature of the incoming data object. If the percentage of usable features drops a certain configurable number, the data object is dropped altogether for further processing.

Note that, the rejected data will be diverted to another pipeline which will track of rejected objects for instrumentation purpose. This instrumentation will be used for improvement of the **"Scrapper Service"**.

> Tech Stack:

- Apache Kafka
- Java spring-boot

## Data Storage Engine (DSE)

> Intro:

The Data Storage Engine is the provider layer for search, auto-suggestion etc features.

Since search need not provide exact results but an approximation to the user query, we are using **“elasticsearch”** for our storage layer. Elastic internally uses probabilistic data-structure like bloom-filter which is the best fit for our use case.

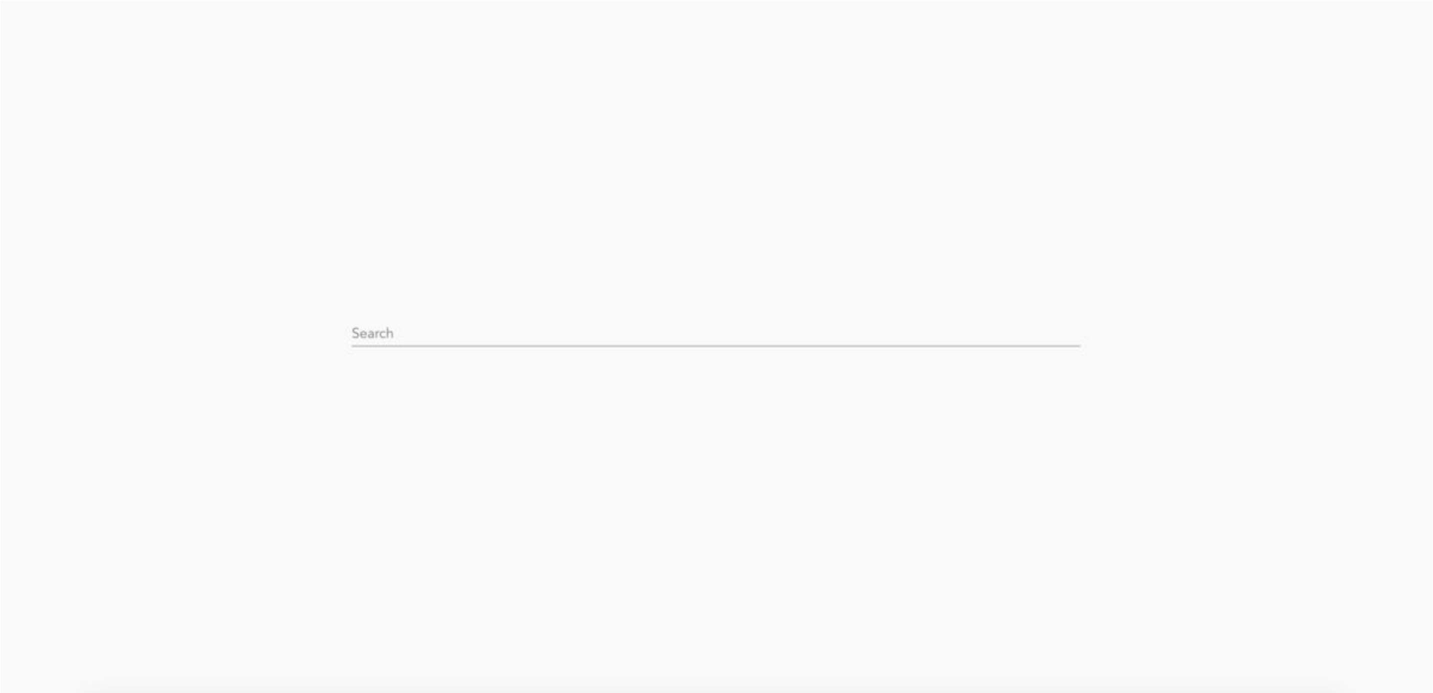
User search will be powered by an elastic store which would be queried against tags attributed by the TAE (Tag Attribution Engine) on the user free text query. Also, since auto-suggestion does not directly result in a user query but instead sets the stage for the query, so it will be powered by another elastic store keeping only the catalogue hierarchy information of all the products. Auto suggestions point to category or subcategory of the products rather than the products themselves. When user query becomes more and more specific, suggestion stores won’t be able to provide suggestions, then we will query the product information store to retrieve the exact product.

## > Tech Stack:

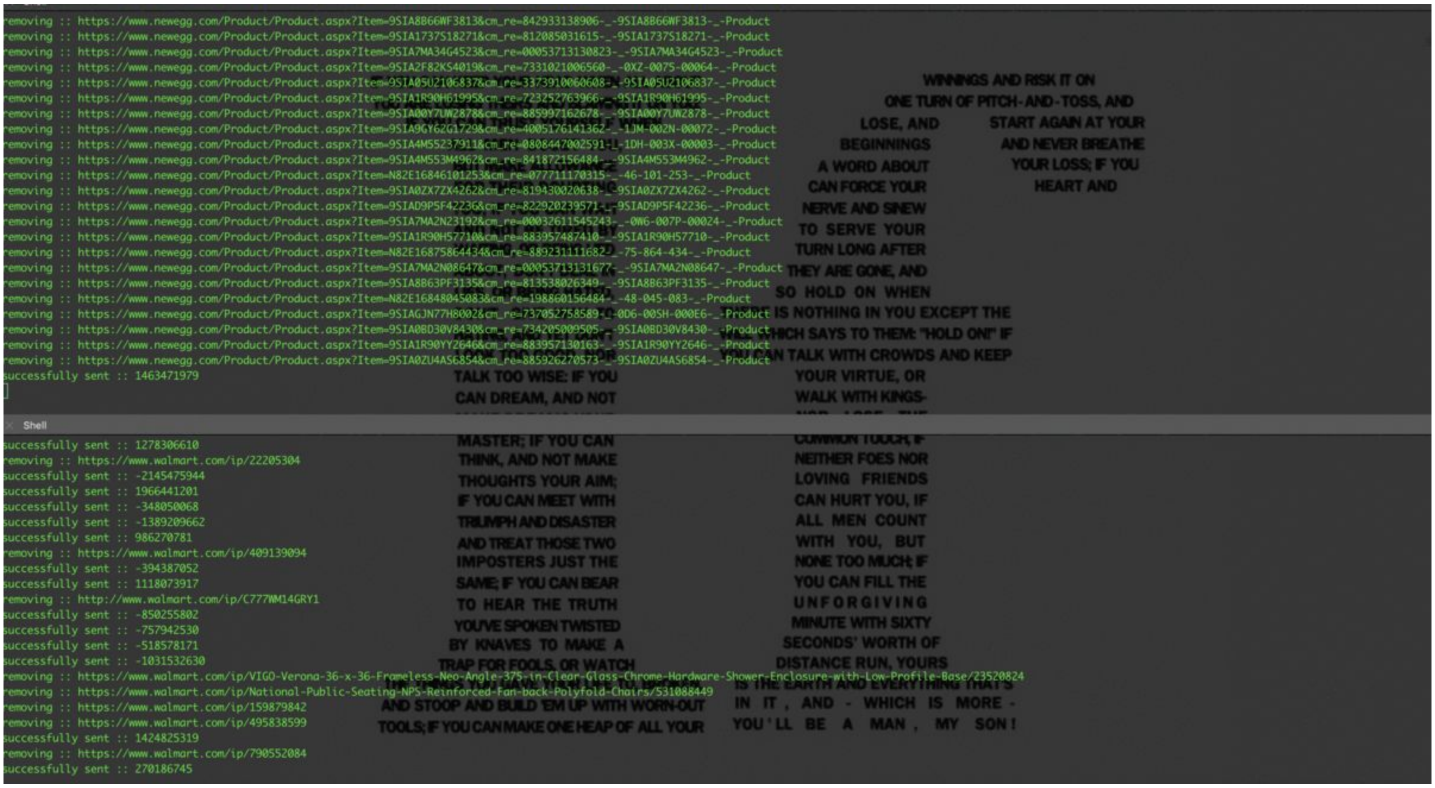
- ElasticSearch

# Images

## 1. Application



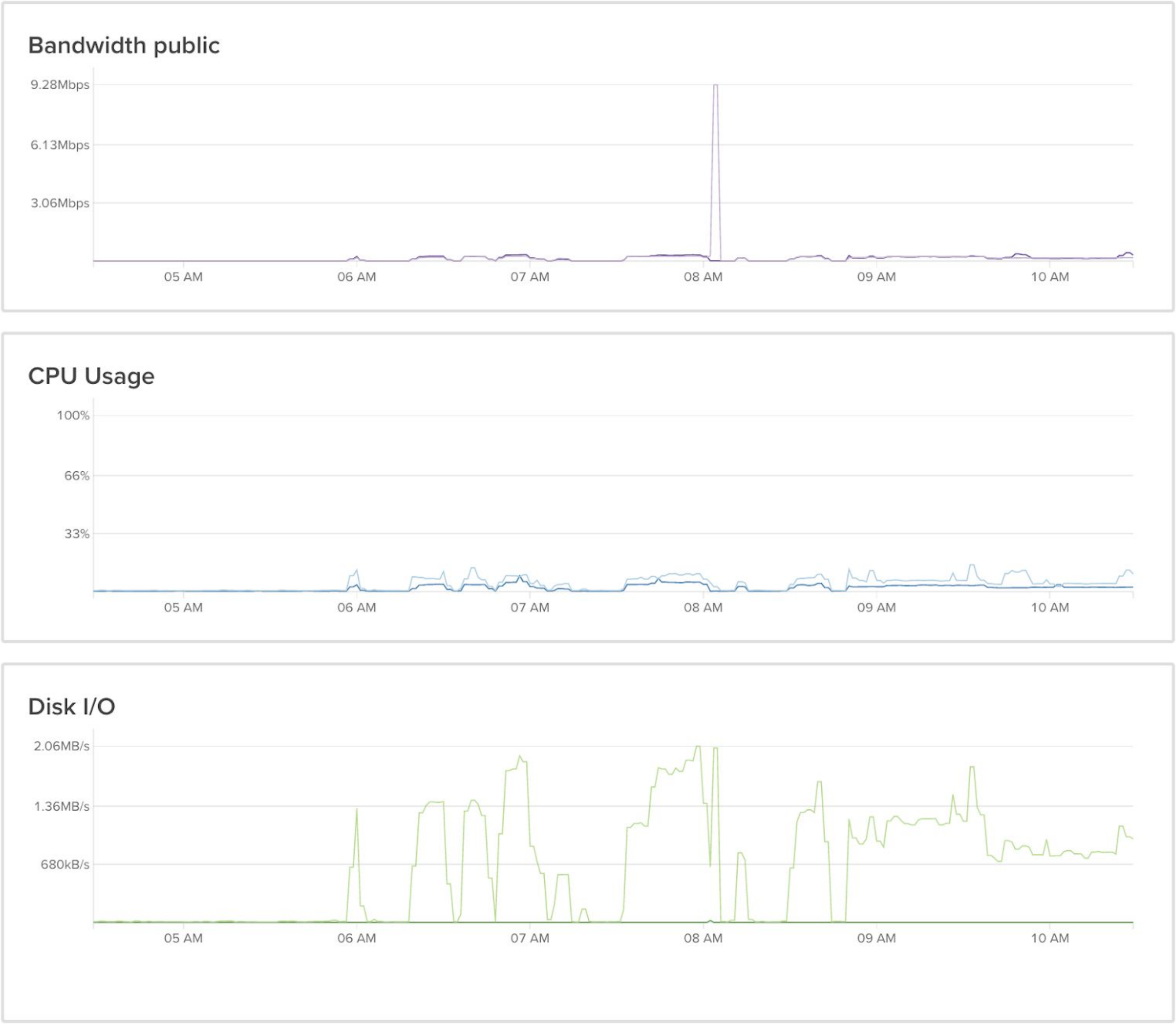
## 2. Data Extractor Engine to Data Attribution Engine syncing…



```
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA8B66WF3813&cm_re=842933138906-_-9SIA8B66WF3813-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA1737S18271&cm_re=812085031615-_-9SIA1737S18271-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA7MA34G4523&cm_re=00053713130823-_-9SIA7MA34G4523-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA2F82KS4019&cm_re=7331021006560-_-0XZ-0075-00064-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA05U2106837&cm_re=3373910060608-_-9SIA05U2106837-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA1R90H61995&cm_re=723252763966-_-9SIA1R90H61995-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA00Y7UW2878&cm_re=885997162678-_-9SIA00Y7UW2878-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA9GY62G1729&cm_re=4005176141362-_-1JM-002N-00072-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA4M55237911&cm_re=0808447002591-_-1DH-003X-00003-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA4M553M4962&cm_re=841872156484-_-9SIA4M553M4962-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=N82E16846101253&cm_re=077711170315-_-46-101-253-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA0ZX7ZX4262&cm_re=819430020638-_-9SIA0ZX7ZX4262-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIAD9P5F42236&cm_re=822920239571-_-9SIAD9P5F42236-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA7MA2N23192&cm_re=00032611545243-_-0W6-007P-00024-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA1R90H57710&cm_re=883957487410-_-9SIA1R90H57710-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=N82E16875864434&cm_re=889231111682-_-75-864-434-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA7MA2N08647&cm_re=00053713131677-_-9SIA7MA2N08647-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA8B63PF3135&cm_re=813538026349-_-9SIA8B63PF3135-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=N82E16848045083&cm_re=198860156484-_-48-045-083-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIAGJN77H8002&cm_re=737052758589-_-0D6-00SH-000E6-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA0BD30V8430&cm_re=734205009505-_-9SIA0BD30V8430-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA1R90YY2646&cm_re=883957130163-_-9SIA1R90YY2646-_-Product
removing :: https://www.newegg.com/Product/Product.aspx?Item=9SIA0ZU4AS6854&cm_re=885926270573-_-9SIA0ZU4AS6854-_-Product
successfully sent :: 1463471979
```

```
successfully sent :: 1278306610
removing :: https://www.walmart.com/ip/22205304
successfully sent :: -2145475944
successfully sent :: 1966441201
successfully sent :: -348050068
successfully sent :: -1389209662
successfully sent :: 986270781
removing :: https://www.walmart.com/ip/409139094
successfully sent :: -394387052
successfully sent :: 1118073917
removing :: http://www.walmart.com/ip/C777WM14GRY1
successfully sent :: -850255802
successfully sent :: -757942530
successfully sent :: -518578171
successfully sent :: -1031532630
removing :: https://www.walmart.com/ip/VIGO-Verona-36-x-36-Frameless-Neo-Angle-375-in-Clear-Glass-Chrome-Hardware-Shower-Enclosure-with-Low-Profile-Base/23520824
removing :: https://www.walmart.com/ip/National-Public-Seating-NPS-Reinforced-Fan-back-Polyfold-Chairs/531088449
removing :: https://www.walmart.com/ip/159879842
removing :: https://www.walmart.com/ip/495838599
successfully sent :: 1424825319
removing :: https://www.walmart.com/ip/790552084
successfully sent :: 270186745
```

# 3. Data Storage Engine (Graphs)


Bandwidth public
9.28Mbps
6.13Mbps
3.06Mbps
05 AM
06 AM
07 AM
08 AM
09 AM
10 AM
CPU Usage
05 AM
06 AM
07 AM
08 AM
09 AM
10 AM
Disk I/O
2.06MB/s
1.36MB/s
680kB/s
05 AM
06 AM
07 AM
08 AM
09 AM
10 AM