



# Project and Architecture Summary



## Table of Contents

1. Project .....	4
1.1. Tag .....	4
1.2. Project Summary .....	4
1.3. Problem .....	4
1.4. The Solution .....	5
1.5. The Business Model .....	5
1.6. Current State and Awards .....	5
2. Architecture .....	5
2.1. Sub-Systems .....	6
2.1.1. Data Collector Sub-System .....	6
2.1.2. Data Collection Controller Sub-System .....	7
2.1.3. Statistics Sub-System .....	9
2.1.4. Enterprise Sub-System .....	10
2.1.5. Web UI / Searcher Subsystem .....	12
2.2. Top Level Deployment .....	13
2.3. Development Environment .....	14

## 1. Project

### 1.1. Tag

**Keywords:** Vertical Search Engine, Information Retrieval, Text Analysis

**Tag Line** Find Everything with Price

### 1.2. Project Summary

Karniyarik.com is a vertical shopping search engine aiming to serve users who are in need of anything that can be bought with money.

The main two important points for Karniyarik.com are:

- **Longtail:** Refers to the statistical property that a larger share of population rests within the tail of a probability distribution than observed under a 'normal' or Gaussian distribution. It describes the niche strategy of selling a large number of unique items in relatively small quantities - usually in addition to selling fewer popular items in large quantities. Karniyarik.com targets to increase findability of long tail.
- **Findability:** Karniyarik.com uses common search engine functionalities and technologies to be able to increase findability. It uses crawlers to collect data, did you mean functionalities and query suggestions to guide user during the search. Karniyarik.com includes advanced query parsing capabilities mainly designed for product descriptions being used in the current local online market.



### 1.3. Problem

Current local “price comparison services” (CSE) are focused on comparable products and product types that are sold frequently, ignoring the long tail. This fact results in a very limited range of product types. Therefore they cannot serve to mothers who are in need of some baby stuff, or women looking for kitchen accessories, or mans looking for automobile parts, children looking for some toys.

Three main properties of CSEs are the main cause of this ignorance: the business model, collecting data only via data feeds, and finally the vision for findability. The business model of CSEs requires every online shop to pay for just to be included in the system. This fact ignores the sites who do not want to pay for just being listed; but has a very different set of products. In the same way because of the technical capabilities of the online shops in Turkey, some sites cannot provide data feeds and they are discarded from system. Finally just having a glance on the CSEs and querying them results in the fact that they have very limited capability of findability. Most of them do not include did you mean mechanisms, query suggestions, powerful free text search and etc. They are just like an aggregated web interface of the XMLs being collected from different sites. There are more than thousands of local online shops and current CSEs include only a very small portion of them (something like 250).

### 1.4. The Solution

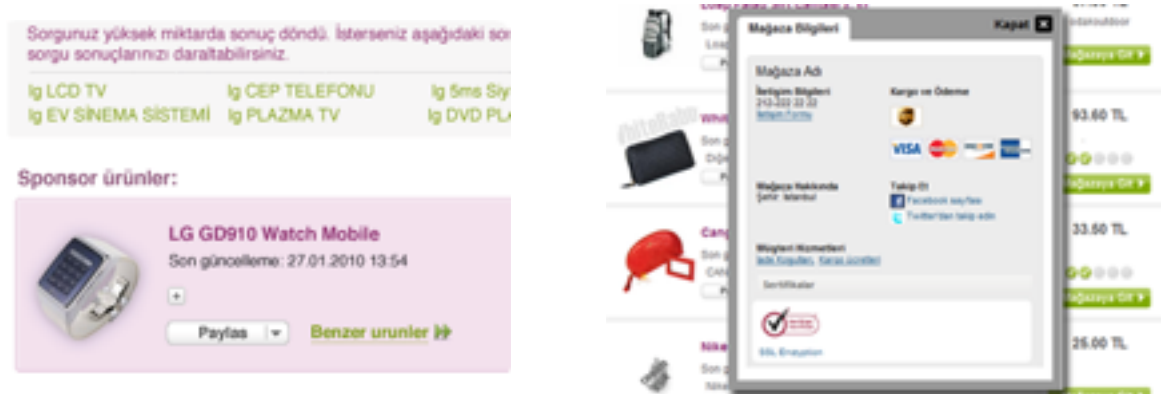
In order to increase findability and product type spectrum Karniyarik.com uses the following features:

- Data collection is accomplished in a hybrid way using both crawlers and data feeds. Sites that can provide data feeds are integrated into the system directly using data feeds. On the other hand sites that cannot provide data feeds are crawled.
- The business model of Karniyarik.com relies on sponsored search, featured merchants and statistics. Therefore in order to be listed in the system online stores do not have to pay. That way the product spectrum is not limited to the ones that can pay for listing. The business model is described in the following section.
- Karniyarik.com welcomes the users with a simple home page similar to general search engines and emphasizes the search engine functionalities. It includes a powerful query analyzer developed for product names and descriptions. The system incorporates auto complete, query suggestion and did you mean mechanisms.

The first two features focuses on Long tail where the last one is mainly for findability.

### 1.5. The Business Model

The business model of Karniyarik.com is based on sponsored search which is a common model on all general search engines in the world. This way the promoted sites/products are listed at the top and differentiated from the ordinary listings. There is no restriction to apply for sponsored search and the mechanism is charged with Cost Per Click (CPC).



Another paid feature of Karniyarik.com is the featured Merchant where merchants can promote their site with a simple box that appears on search result pages. This box contains instant information about the site to the users including contact information, social media links, security certificates etc...This feature is charged according to monthly plans.

### 1.6. Current State and Awards

## 2. Architecture

Karniyarik.com is an information retrieval system that includes many of the common components and functionalities of a search engine. Almost all of the system is developed in Java and sub-systems are designed as web application archives which communicates with the other sub-systems over REST. This section describes the high

level architecture of the system and gives some information about the development environment and 3rd party tools used to implement the system.

## 2.1. Sub-Systems

The current system is composed of five main sub-systems: Data Collector, Data Collector Controller, Statistics, Web UI/Searcher and Enterprise.



Figure 1 Sub-Systems

### 2.1.1. Data Collector Sub-System

Data collectors are responsible for collecting the product information from the configured sites. It includes crawler, parser, ranker, and indexer components which are the main components responsible for collecting information displayed in Karniyarik.com.



Crawlers are Java components that confirms to common crawler policies such as selection, re-visit, politeness and parallelization. It is a robust component that executes 7/24. It has to handle unexpected situations in the machine it is working, the internet connection, the state of the crawled site and etc. Crawler's uses regular expression based configuration in order to narrow the pages to be crawled. Since most of the online stores use the same structure (main pages, category pages and product pages plus informative pages) some common rules based on regular expressions are present in the crawler component.

Karniyarik.com uses a hybrid approach for data collection. While crawlers collect data using automatic robots, data feed component uses XML based data feeds to collect data. In order to be flexible XSLT and XPath is used in converting the site XML formats to Karniyarik.com XML format. This way a site using Data feeds can be easily integrated into the system.

Parsers are responsible for extracting product information from web pages crawled or data feeds collected. Recognizer component uses controlled vocabularies and some common AI algorithms such as Naive Bayes in order to normalize, unify or recognize names entities such as brand, category, gear type, city, color etc. Karniyarik.com can identify the brand information in a product name, as well as the category of a product just by the name of the product. This component is expected to be enhanced so that Karniyarik.com can recognize the models of products and cars; and matching of the same products.

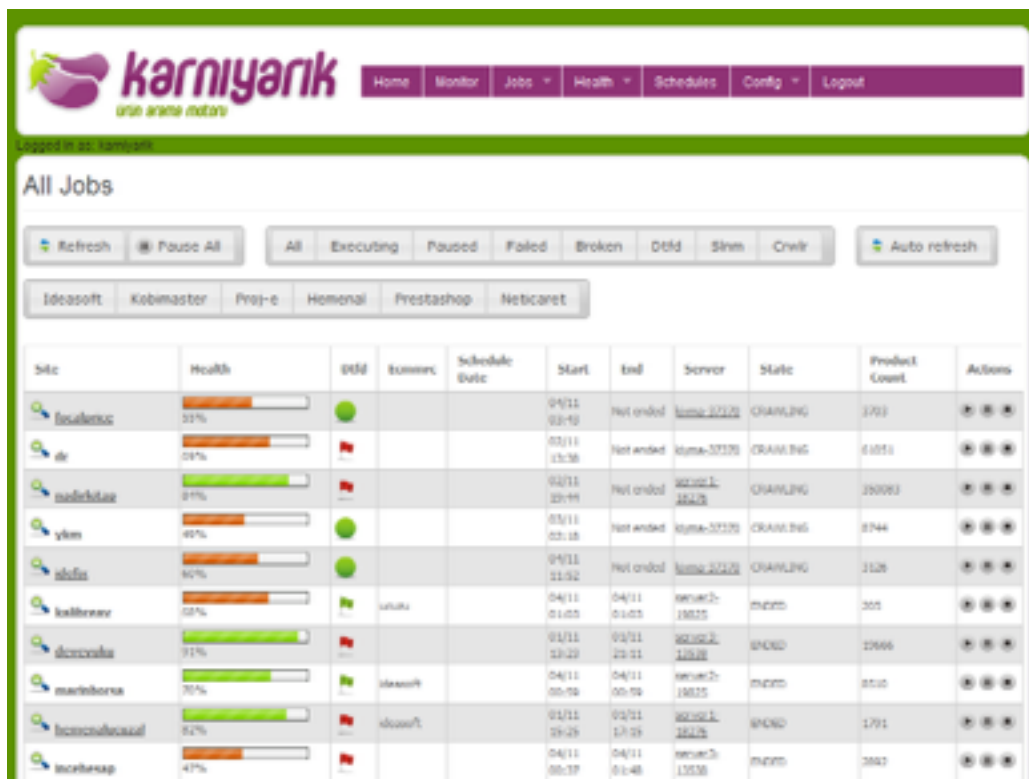
Indexer component uses inverted index algorithm to index the product descriptions. It uses some text and language analyzers specifically developed according to product

descriptions present in the current online stores. The default language is accepted to be Turkish since the system serves to only Turkish market.

Finally, Ranker component is used to rank the products in online stores for search result quality. Karniyarik.com uses page rank to calculate product importance in the crawled sites. It also uses some other metrics to rank the site such as Alexa ranks, site data collection quality (how much time it takes to crawl the site, number of erroneous links) and web usage statistics (site score from the statistics sub-system etc.)

### 2.1.2. Data Collection Controller Sub-System

This sub-system is a monitoring and management extension for Data Collectors. It is a Web Archive that helps the administrators of Karniyarik.com to monitor the state of the crawlers, parsers, indexers and rankers. It presents important statistical and historical information about the data collection process. The interfaces are basic and based on JSP/HTML/CSS/JavaScript. Data collection controller uses Quartz to schedule the collection processes and is integrated to Google Analytics and Alexa.



The screenshot shows the 'All Jobs' page of the Karniyarik Data Collection Controller. The interface includes a navigation bar with links: Home, Monitor, Jobs, Health, Schedules, Config, and Logout. Below the navigation bar, there are tabs for 'All', 'Executing', 'Paused', 'Failed', 'Broken', 'Old', 'Silm', and 'Cnvr'. There is also an 'Auto refresh' button. The main content area displays a table of jobs with columns: Site, Health, PID, Summary, Schedule Date, Start, End, Server, State, Product Count, and Actions. The table lists various e-commerce sites and their crawling status.

Site	Health	PID	Summary	Schedule Date	Start	End	Server	State	Product Count	Actions
basalemus	33%				04/11 02:40	Not ended	home-37129	CRASHING	1703	⌵ ⌵ ⌵
de	59%				03/11 13:38	Not ended	home-37129	CRASHING	61051	⌵ ⌵ ⌵
madefake	21%				04/11 20:49	Not ended	home-37129	CRASHING	29093	⌵ ⌵ ⌵
ylm	40%				03/11 02:18	Not ended	home-37129	CRASHING	8744	⌵ ⌵ ⌵
akofa	60%				04/11 11:52	Not ended	home-37129	CRASHING	3128	⌵ ⌵ ⌵
kollimaz	55%		urulu		04/11 01:05	04/11 01:05	home-37129	CRASHING	305	⌵ ⌵ ⌵
decevala	91%				01/11 12:22	01/11 12:22	home-37129	CRASHING	12666	⌵ ⌵ ⌵
mariborsa	70%		istavak		04/11 00:59	04/11 00:59	home-37129	CRASHING	8110	⌵ ⌵ ⌵
hacretahusul	82%		aksoy		01/11 12:29	01/11 12:29	home-37129	CRASHING	1791	⌵ ⌵ ⌵
aciktesap	41%				04/11 00:37	04/11 00:46	home-37129	CRASHING	3847	⌵ ⌵ ⌵

Figure 2 Site List



Figure 3 Site Detail

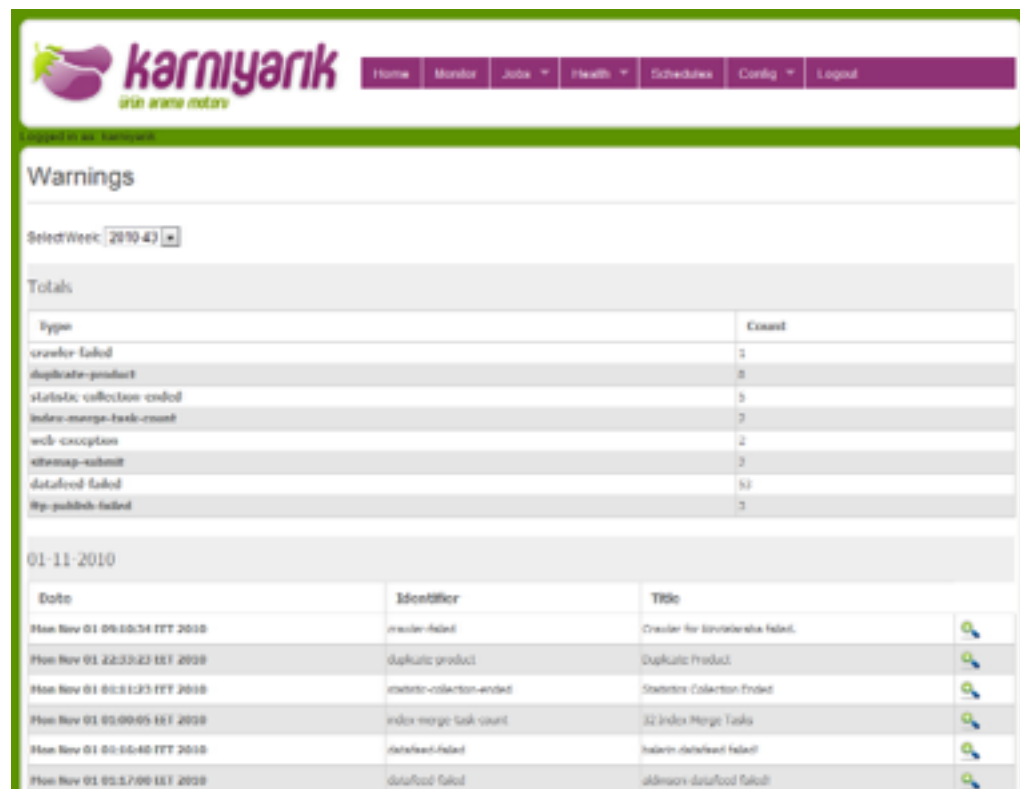


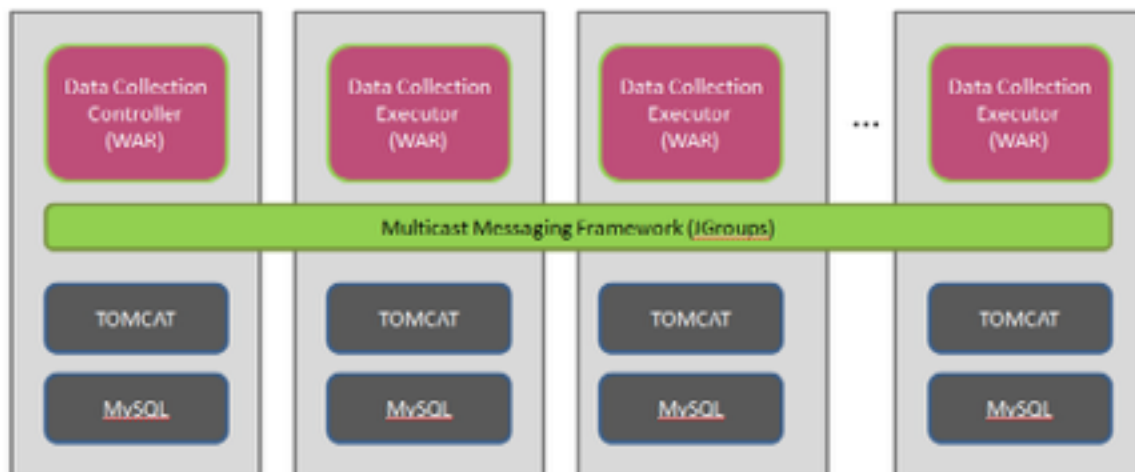
Figure 4 Warnings/Exceptions





Figure 5 Dynamically Recognized Servers/Health

The communication of Data Collector Controller and Data Collectors is accomplished using multicast messaging. JGroups is used in order to handle such a task. In the production system there is one Data Collector Controller and multiple Data Controllers (also shown in Figure 5). The following figure describes the deployment and communication of collectors and the controller.



### 2.1.3. Statistics Sub-System

Statistics subsystem is a web application with no UI and is responsible for collecting statistical information flowing through the whole Karniyarik.com. It provides REST based interfaces and runs on MySQL to collect and present statistical information.

### 2.1.4. Enterprise Sub-System

Enterprise sub-system provides some monitoring and management functionalities to the Online Store representatives. Moreover this system is the main controller for the business model of Karniyarik.com. Actions such as updating online store information, applying for sponsored search or featured Merchant, presenting statistical usage and data collection information of Online stores are handled via this system.

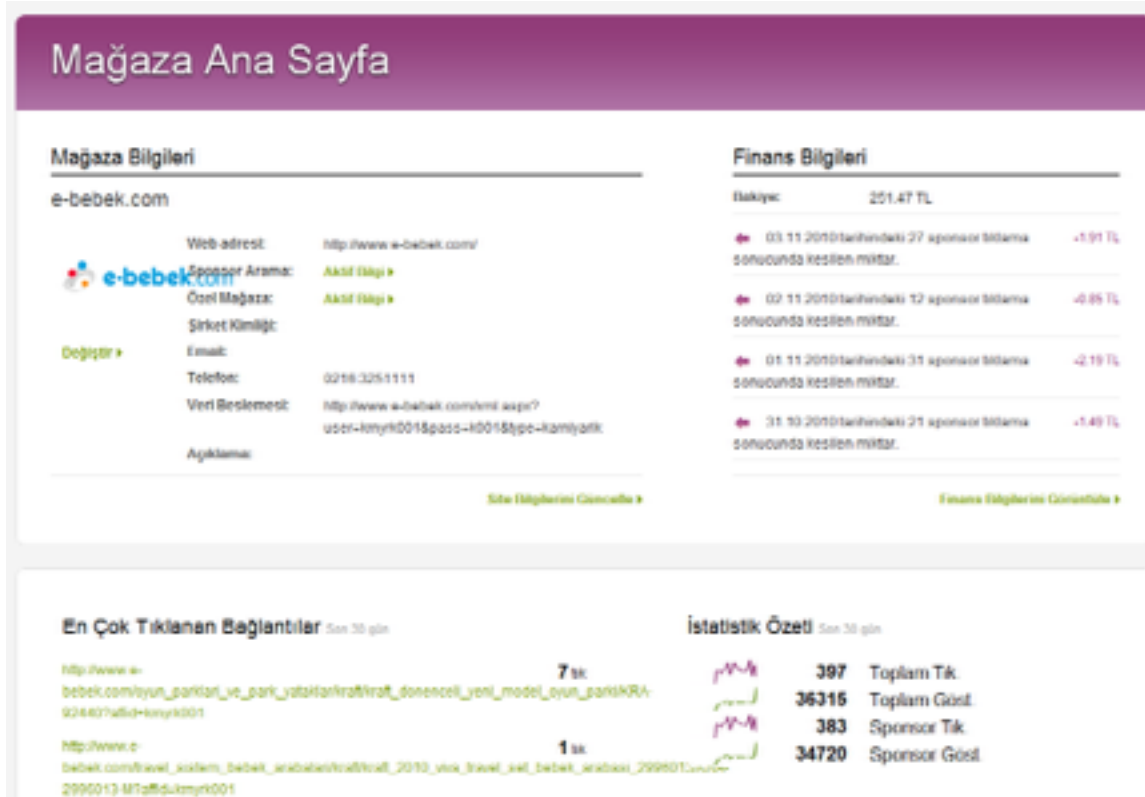


Figure 6 Enterprise - Site Home Page

Unlike the other sub-systems of Karniyarik.com Karniyarik Enterprise is developed using Grails framework. As a final outcome of the build a web archive is generated and deployed on Tomcat server. It mainly contains CRUD operations for both administration and monitoring information. This is the reason for selection of Grails framework for implementation. It uses MySQL as the underlying database.

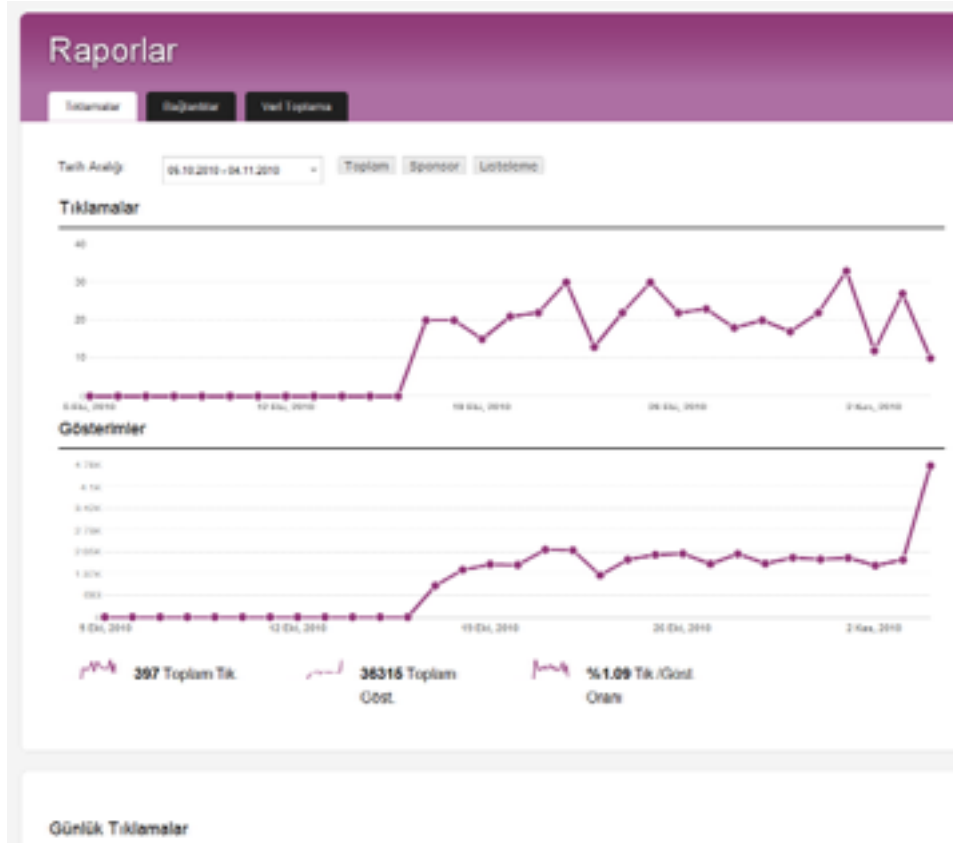


Figure 7 Enterprise - Site Usage Statistics

**Raporlar**

Tıklamalar Gösterimler Veri Toplama

Veri Toplama Geçmişi

Başlangıç Tarihi	Bitiş Tarihi	Durum	Ürün Sayısı
		Durumunda	0
02/11/2010 08:13:12	02/11/2010 08:13:21	Başarılı	4100
01/11/2010 08:31:40	01/11/2010 08:31:49	Başarılı	4100
31/10/2010 08:31:42	31/10/2010 08:31:55	Başarılı	4100
30/10/2010 08:11:18	30/10/2010 08:11:51	Başarısız	0
29/10/2010 08:12:42	29/10/2010 08:12:53	Başarılı	4100
28/10/2010 08:11:38	28/10/2010 08:11:41	Başarısız	0
23/10/2010 08:16:00	23/10/2010 08:16:08	Başarılı	4096
28/10/2010 08:13:43	28/10/2010 08:13:54	Başarılı	4094
26/10/2010 08:31:53	26/10/2010 08:32:02	Başarılı	4100

Figure 8 Enterprise - Site Data Collection Statistics

### 2.1.5. Web UI / Searcher Subsystem

The final sub-system of Karniyarik.com is the part which serves to the users on Internet. It is a very simple web application at the top level containing only JSP/CSS/JavaScript. JSP is selected as the UI framework because this part of the system has to be very fast and contains only a few pages (homepage, search result page and some informative pages). Web UI system use Searcher component to respond user searches. Searcher component borrows many subcomponents from Indexer component of Data Collectors including the text analyzers and inverted index.



Figure 11 Web UI - Home Page

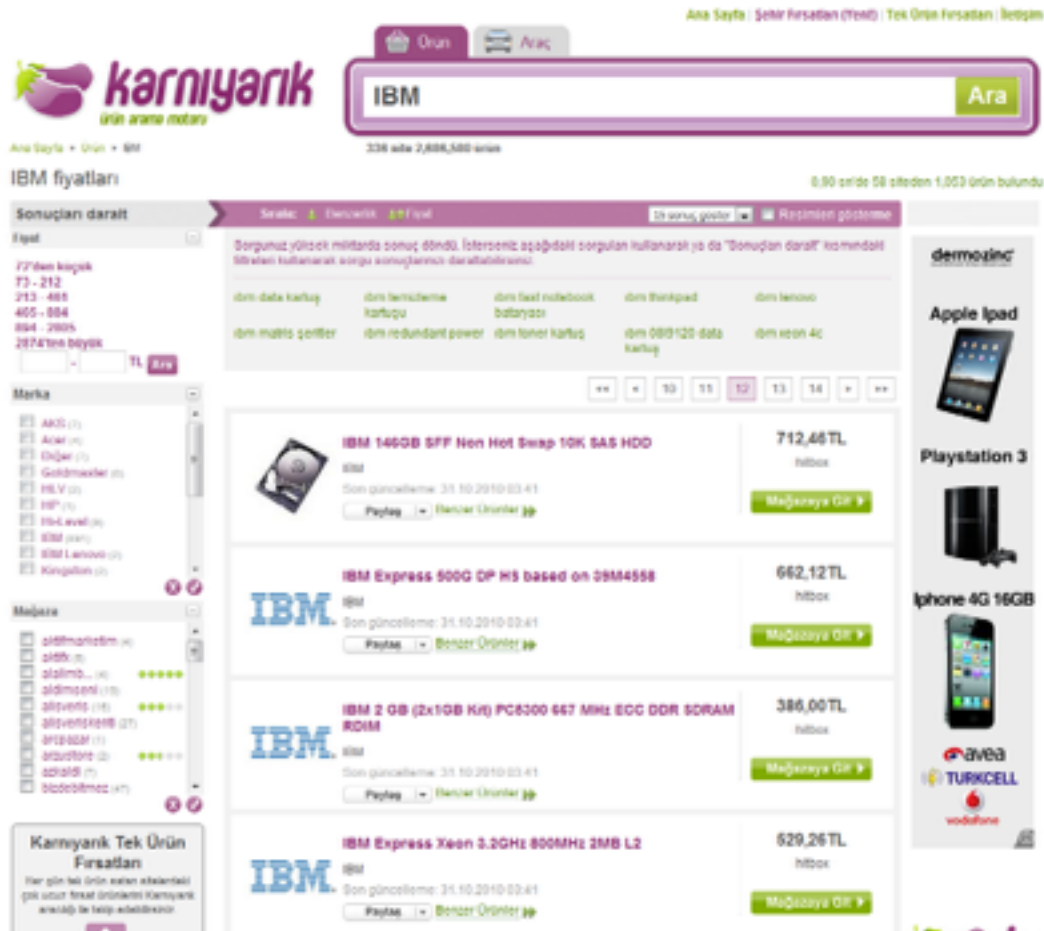


Figure 12 Web UI - Search Result Page

## 2.2. Top Level Deployment

Karniyarik.com uses sub-systems which are bundled as Web Archives (WAR) and uses Multicast Messaging and REST methods for the subsystem communications. Currently 7 machines are used where one or more Tomcat servers are deployed on each machine. MySQL is used as the DBMS and Ubuntu is used as the OS. Apache Load Balancer is used to load balance Web UI and some servlets getting high load (such as image resize servlets). The below diagram presents the current deployment of Karniyarik.com:

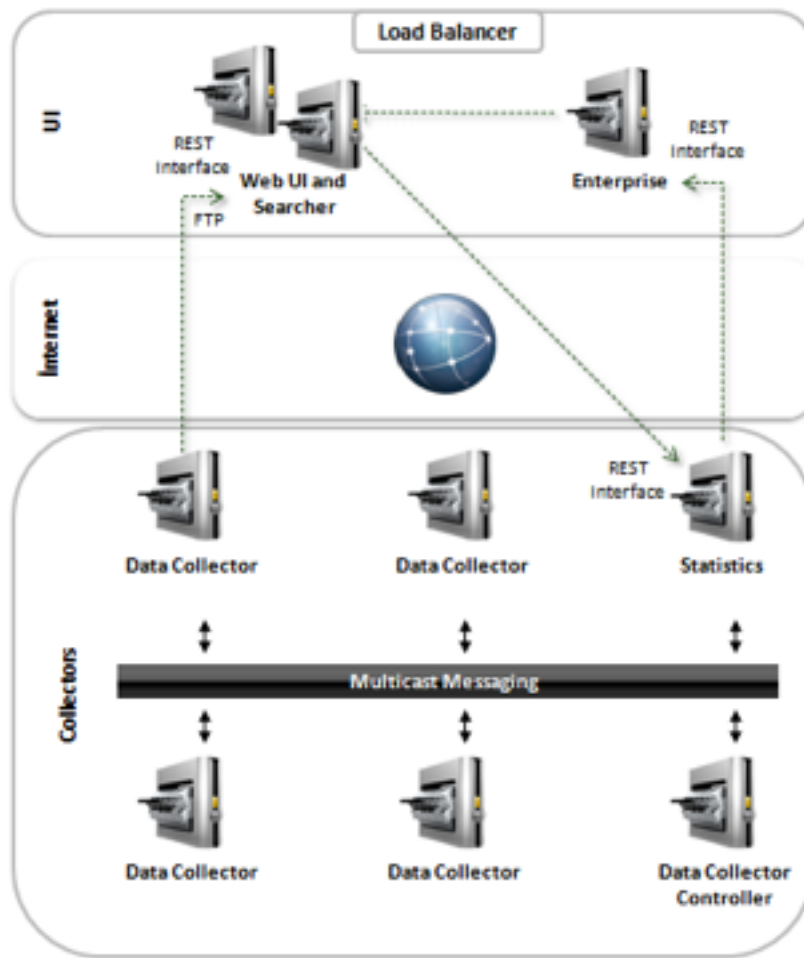


Figure 13 High Level Deployment

### 2.3. Development Environment

The development and deployment tools/software being used in the system are given in the following table.

Name	Version	Description
Java	1.6	Programming Language
Groovy		Programming Language
JEE	1.5	Enterprise Framework
Grails	1.1.1	Enterprise Framework
Eclipse	Galileo	IDE
WTP	3.0	IDE
Maven	2.0.10	Build Framework
Ubuntu Linux 64-Bit	9.0	Production OS
Windows	Vista/7	Development OS
MySQL	5.1	DBMS

Tomcat	6.0.x	Servlet Container
Trac	0.11	Project Management, Bug Tracking tool
Hudson	1.3	Continuous Integration Environment