**Press Release: Real-time Search-as-a-Service**

# Introduction

Real-time Search-as-a-Service (RTS) is a cloud service built for developers, by developers that aims at solving the problem of integration of real-time search with Mobile and Web apps.

We provide our clients SDKs for all platforms, for seamless integration of “as you type” search results. Also, the results are semantically consistent with the query – for example – searching “I have a burst pipe” will return you a list of plumbers.

Please refer to our documentation for a detailed list of features.

To ensure low latency, the calls for search are directly sent from the user’s device (app, website) to our platform.

# Why we built this?

One of the most challenging and time-consuming tasks while developing a web or mobile app is the integration of search in it.

Studies say that a bad search experience causes a business at least 1% loss in sales.

The platform is available as a cloud service, because developers face a tough time integrating small search engines into their apps – for example, many people have tried integrating Apache Lucene into their Android apps.

The results need to be instantaneous and tolerate typos, which are so frequent on a smart phone. Semantic analysis of the query ensures that only the relevant results are returned.

# How is it different from other search engines?

Most of the search engines/providers in the market today offer Document based search. These engines use ranking rules that have been designed for documents. This ranking counts the occurrences of terms of queries inside each document and uses statistics based on term frequency – inverse document frequency (tf-idf).

You want documents that contain many times the query terms you typed to be ranked higher. And this is a statistical formula that will generate a score for each document. This score is a bit cryptic, you can tune the formula, adapt it a little bit. But in the end you get a score that is very difficult to understand. It’s very difficult to tune a working formula for a document search engine. And that works well with full words but it doesn’t work well with prefixes, like when you are not typing the word completely.

**Our focus is not documents, but databases, because developers have databases to query from.**

Because the data is different, we needed to come up with a totally different way of ranking the results, closer to the way people think. For example, if you search for an iPhone on an e-commerce website, you just want to find the word iPhone in the title and not at the end of the description. So the position of the word in the results is much more important than the number of occurrences.

You also don’t want the iPhone 3G, you want the latest iPhone 5S, the most popular one. So the advantage of having this database focus is that we can consider this kind of information, which is very difficult to do with a document search engine.

You also want to take into account other factors such as did I misspell my query? If you are an e-commerce company you may want to get the best sales first in your set of results. So you want to combine all these together and this mix is too complex with a document search engine. So we have developed a different approach that takes all these criteria into account in a very explicit way, everyone can easily understand why you get a certain set of results and why they were ranked that way.