# Refer Engine Provisional Patent

Refer Engine is an online application that integrates application stores of different platforms with social networks and enables app developers to integrate their apps in a powerful way.

## General Description

Refer Engine consists of several customer facing experiences.

### ReferEngine.com

This is the main Refer Engine website which shows the information about applications in different platforms like iOS App Store, Google Play, Windows Store, etc… The user can browse these applications by category/subcategory and sort by different criteria like ratings, number of recommendations, etc… Once logged in to Facebook, the user can also browse the apps that different friends use and sort those by category, etc…The user can also search for applications in different stores or in specific app stores.

Refer Engine will include different modes of browsing apps. The user can sort and view using known methods such as sort by rating, sort by number of likes, etc… Refer Engine also offers different viewing modes: browse 1 app at a time, browse a grid of apps (default), and browse by app screenshots only. Another browsing mode is to sort randomly and show one random app at a time. This helps with discovering apps that may otherwise not come up to the surface.

ReferEngine.com will fully integrate with the social graph of Facebook. Every Store, Category and App page will be a Facebook Graph object that support native Facebook graph actions like sharing, liking and posting on timelines, etc…

ReferEngine will enable app developers to fully control the listings of their apps. Once app developers register and verify their ownership of an application, they can modify all text, add high resolution screenshots and videos and connect their twitter and Facebook feeds on the page. This creates very powerful landing pages for these applications that can be controlled by the developer to a large extent.

### ReferEngine.com/Developer

This is the developer portal where application developers can register their ownership of applications and modify content that shows up within their application’s page on ReferEngine.com. The developers can also get the code packages that they need to install REAPEX (below) within their applications.

The developer can specify what kinds of rewards they wish to offer their customers and how many recommendation points each of those rewards cost. Examples could be:

* 4 game points for every 1 recommendation point
* 10 game points for every 2 recommendation points
* 20 game points for every 3 recommendation points

### ReferEngine In-App Experience (REAPEX)

The ReferEngine In App Experience (REAPEX) is an additional feature that app developers can sign up for and integrate with their apps. Once integrated, REAPEX offers the customers the ability to:

* Sign up as a Refer Engine Verified User of the app
* Submit a Verified Recommendation of the app

The Verified User is a Refer Engine concept for customers to identify themselves as really using this app. If the customer particularly likes this app, they can submit a verified recommendation. Verified recommendations are recommendations that can only happen from within the applications. The developer can optionally choose to offer an incentive or an in-app reward based on the result of the customer’s recommendation. Refer Engine will track these recommendations and calculate what rewards every customer earns. In order to maximize their chances of collecting Recommendation Points, the user can submit multiple recommendations. To minimize spam, the user can only submit one recommendation per app every certain period of time (currently set at one month).

Developers who participate in the In-App experience have additional capabilities such as contacting their customers directly through Refer Engine.

Another task that REAPEX does is to offer the customer the ability to redeem the rewards they earned from their recommendations. Whenever the app launches, REAPEX checks with ReferEngine.com servers and determines whether the user has earned a reward. If the user earned a reward, it pops up and notifies the user of the reward. If the user chooses to claim the reward, REAPEX will call a pre-defined function inside the app to notify the app of the reward to be claimed. The app gives the user the reward (game points, etc…) and calls back into REAPEX that the reward has been claimed successfully. REAPEX will then send this info to ReferEngine.com servers and the handler will deduct the appropriate points from the user’s Recommendation Points.

## Implementation

Refer Engine is implemented as a native cloud application with several cloud service endpoints, background workers and data stores.

### ReferEngine.Cloud

ReferEngine.Cloud is the main cloud service that responds to all requests made to any part of [www.ReferEngine.com](http://www.ReferEngine.com). This includes all three services mentioned above. It includes those components:

#### ReferEngine.Web Web Worker

This is the main worker that all requests made to [www.ReferEngine.com](http://www.ReferEngine.com) are services. It is implemented as an ASP.NET MVC Application.

#### ReferEngine.DataWriter

This is a background worker that the Web Worker sends database updates to. This is used so web requests are not blocked on database writes which could be costly.

### ReferEngine.Workers

This is a cloud service that contains several background workers that are necessary to make Refer Engine work as required.

#### ReferEngine.Workers.WinApps

This is an automated background worker that wakes up once every 24 hours. It starts by pinging the Windows Store Sitemap for the links to all apps ([http://apps.microsoft.com/windows/sitemap/sitemap\_{0}.xml](http://apps.microsoft.com/windows/sitemap/sitemap_%7b0%7d.xml)) where {0} is the page number. All the links to the apps are stored in the database.

Once the worker has collected all the links, it then pings every link one by one and parses the information for every app. This information includes Name, Age Rating, Publisher Name, Description, Website Link, Privacy Policy, Screenshots, Supported Languages and Architectures, etc…

The information for all these apps are stored in the database. If an app link fails to load, we store the number of failures and once it fails to load 5 days in a row, we consider it as a deleted application and delete its information from the database.

Once the process is finished, the worker sleeps for 24 hours then wakes up and repeats.

#### ReferEngine.Workers.iOSApps

This worker works in a very similar way to the REFERENGINE.WORKERS.WINAPPS worker but instead of collecting information about the Windows 8 apps, it collects information about iOS (iPhone and iPad) apps.

#### Additional Collector Works

Addition app collection workers are planned to collect information about applications in the Android (Google Play) store, Windows Phone, Blackberry and other application stores.

#### ReferEnigne.Workers.RewardGenie

This is the worker that scans the latest activity that is happening on Refer Engine and determines which users have earned which rewards from which app developers.

When this worker wakes up, for every app that is signed up with the Refer Engine In App Experience: it scans the database for any new verified users and verified recommendations added since it last operated.

* For every User A who registered as a verified user or submitted a recommendation for App-1
* How many Users B are there who: are a friend of A and who submitted a recommendation of App-1 within the last X weeks? (X is a variable that we are experimenting with and could change)
  + If there is zero Users B
    - User A is here organically.
  + If there is one User B
    - User B earns a recommendation point for App-1
  + If there is more than one Users B
    - User B who submitted a recommendation first will earn the recommendation point
    - All other Users B don’t earn a recommendation point. This is to avoid rewarding more than one recommenders recommendation points for one new user. Note that the recommendation “expires” after X # of weeks and is considered ineffective.