Peekin Technical Solution Document

for

Course# DS-610

Big Data Analytics

Muhammed Khan

Version Number: 0.0

Version Date: Dec 13, 2015

Document History

Document Details:

|  |  |
| --- | --- |
| Service Request #: |  |
| Project Title | Project Name |
| Professor |  |
| Author |  |
| Date Prepared: |  |
| Completion Date: |  |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Revision Date | Summary of Changes | Changes marked |
| (#) | (-) | (Describe change) | (N) |

Peer Review

This document requires following approvals.

|  |  |
| --- | --- |
| Name | Title |
| (name) | (title) |

Distribution

This document has been distributed to

|  |  |
| --- | --- |
| Name | Title |
| (name) | (title) |

Contents

1. Introduction 4

1.1 Project Name Overview 4

1.2 Project Name Scope 4

2. Architecture Overview 5

2.1 IT System Level 5

2.1.1 Diagram 5

2.1.2 Key Concepts 5

2.1.3 Key Components 5

3. Architectural Decisions 6

3.1 <Decision 1> Subject 6

4. Statistical Model and Method 7

4.1 Technology 1 7

4.1.1 Statistical Method 7

5. Solution Components 8

5.1.1 Hardware 8

5.1.2 Software 8

5.1.3 Network 8

5.1.4 Monitoring 8

5.1.5 Backup and Recovery 8

5.1.6 Disaster Recovery 9

6. Viability Assessment 10

6.1 Functional Requirements 10

6.2 Non-Functional Requirements 10

6.3 Risks 11

6.4 Assumptions 12

6.5 Dependencies 12

# Introduction

## Propensity to Buy Model Analysis Tool Overview

**Short Value Proposition**

Location based Social Network and online community that provide users with immediate(vicinity and ML) user generated and geo tagged (map, vendors, groupons, etc products) content, offering localized rewards, feedback, and venue suggestions.

**Customer**  **segments**

smartphone users (ultimate market)

business­­­ and venue owners(new vouchers/ultimate company FINANCIAL sponsors)

developers (software using API, data from searches or check-ins)

Marketers(location based advertising)

**Long: Improves Response, Increases Revenue (Incremental), Reduces Cost and helps in targeted marketing**

Propensity models make true predictions about a customer’s future behavior. With propensity models one can truly anticipate a customers’ future behavior. If used in the other way (selecting the customers who has the least propensity to buy based on the probability scores), it can also act as a churn model (identifying top churners for the business). Without propensity models, it becomes difficult to understand as to with which particular product each particular customer/prospective customer needs to be targeted with. With no such predictive model/prior information in place, the targeting would be random which would lead to pitching in of products to the wrong customer who has a low propensity to buy it. Offers/coupons/discounts may be provided to customers who were anyways willing to buy the product resulting in higher costs and zero incrementally.

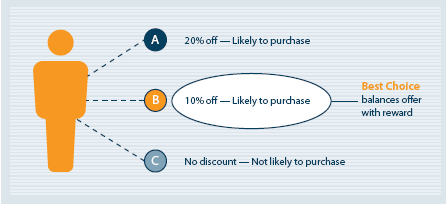
The propensity to buy model tool tells you which customers are ready to make their purchase: so one can find who to target. Moreover, once you know who is ready and who is not helps you provide the right aggression in your offer. Those that are likely to buy won’t need high discounts (This will help to stop cannibalizing margins) while customers who are not likely to buy may need a more aggressive offer, thereby bringing incremental revenue. This can also be used to make location/gender based targeting marketing based on a customer’s past visits to similar places and his behavior at those places.

## Propensity to Buy Model Analysis Tool Scope

Propensity to buy is recorded as a binary variable, 0 or 1, with 1 indicating that a shopper chose a product.

The purpose of this kind of analysis is actionable insights. We would be building here a propensity to buy model which would be useful for an e-commerce firm to target its prospective customers with the right product at the right time. This will not only help in reduced marketing costs and improved response rates but also lead to incremental revenues for the business. In our case, we will be building a propensity to buy model for the products of our business. Similar propensity models can be built for tech products, kids’ products, sports products and so on.

Following can be different scenarios which are encountered in a typical ecommerce scenario and it is important to build a model to come up with the optimal choice.

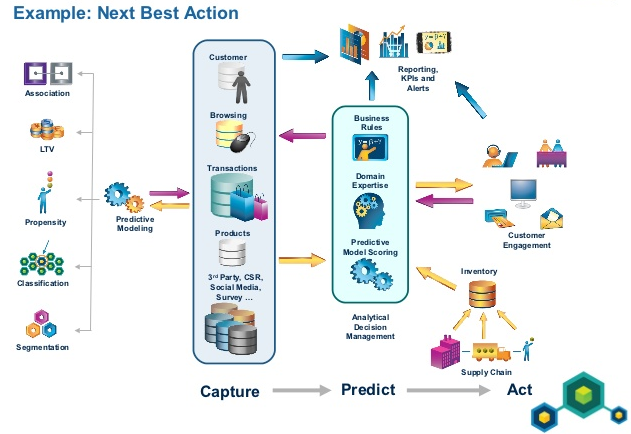


**Typical Propensity to buy model building methodology for taking best actions:**

Techniques: Propensity, Classification, Segmentation, LTV (life time value), Association Mining

Types of Data: Transactional, Demographic, Behavioral, Social Media, Product, Browsing, Survey etc.

Model Building and Validation: Logistic Regression, Business Rules set up, Model Scoring, Out of Time and Out of Sample Validation



Source: <http://image.slidesharecdn.com/leverageyourcustomerdatatopredictyourcustomersactions-copenhagen22nov2012-121122081440-phpapp01/95/leverage-your-customer-data-to-predict-your-customers-actions-colin-linsky-18-638.jpg?cb=1354085200>

# Architecture Overview

## IT System Level

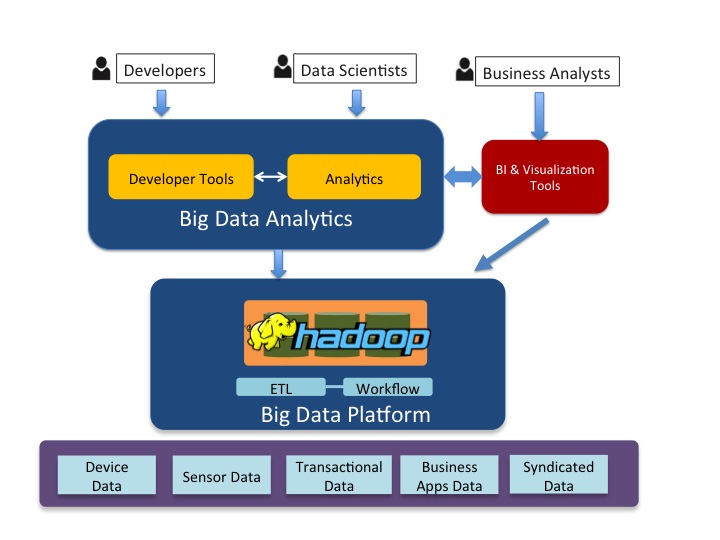
Transactional, Demographic, Browsing (billions of records), Social Media and Survey data will be uploaded to the tool, which will reside on the Amazon Web Services cloud.

The attributes of the Feedback data will include Feedback ratings based on experience of product booking on the website, ratings on shipping time, quality of product etc. The attributes of the Transactional data will include a customer’s transactional history details, average transactions over a specific time period, ticket size, recent check in, frequency and monetary value of transactions. The attributes of the Psychographic data include customer’s interest (hobbies) data for targeted marketing. The attributes of the Demographic data include information on the customer’s age, education, income level, gender. The attributes of the Browsing data include information on the customer’s visits to the website, source of visit (marketing channel touched), funnel metrics etc.

The data will be cleansed, parsed and uploaded to a database for analysis. Using Revoltution R(Underlying C and FORTRAN), Scala the multivariate statistical learning methods will be used to classify the customers into high propensity to buy customers and others. We choose Scala over Python for the big data analytics and text mining unlike interpreted languages where byte code is assembled line by line, through an interpreter, Scala is compiled on runtime. Scala also has MLLib machine learning library, and access to all Java libraries as well (ie ESRI for geographic data). Then predictive analytics will be employed to further classify the number of repeat purchases a high propensity purchase customer will perform.

Diagrams:

Logical Design



Browsing Data

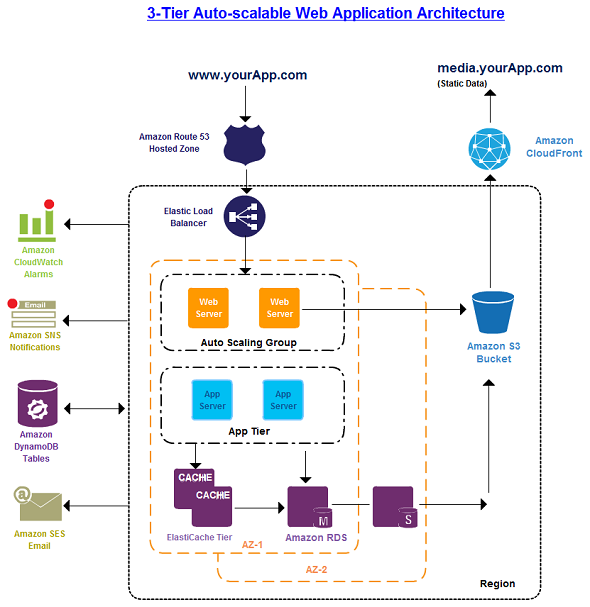
Survey Data

Email Metrics Data

Transactional Data

Demographic Data

Physical Design

******

www.grouponApp.com

www.grouponApp.com

Original Source: <http://www.conceptdraw.com/samples/resource/images/solutions/finance-&-accounting-charts/NETWORK-DIAGRAM-AWS-Architecture-Diagrams-3-Tier-Auto-scalable-Web-Application-Architecture.png>

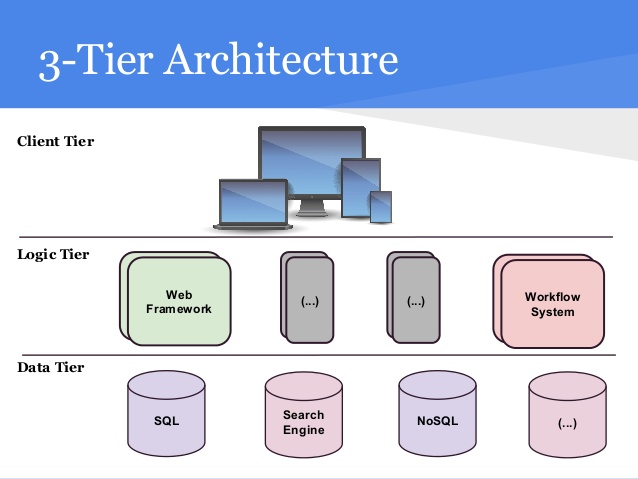
**2.1.2 Key Concepts**

The following key concepts apply to the tool:

* The separation of functions into the Data Tier, the Business Logic Tier and the Presentation Tier.
* 3-Tier Architectural model
* The browser based delivery mechanisms that the architecture will support includes smartphone, tablet and workstation (laptop and desktop).
* There is no hardware that will be deployed as all services are on cloud using AWS.

During an application's life cycle, the three-tier approach provides benefits such as reusability, flexibility, manageability, maintainability, and scalability.

**2.1.3 Key Components**

******

The application uses a 3-tiered approach in moving Transactional, Demographic, Browsing (billions of records), Social Media and Survey data through the tool to the client presentation layer. Below the various technologies are mapped to a tier level

**Layer 1 Technologies (Presentation Layer)**

This layer presents data to the user and optionally permits data manipulation and data entry. The two main types of user interface for this layer are the traditional application and the Web-based application. Web-based applications now often contain most of the data manipulation features that traditional applications use.

* EC2 instance
* Tableau
* Amazon CloudFront
* Elastic Load Balancer

**Layer 2 Technologies (Business Logic)**

The middle tier is where developers can solve mission-critical business problems and achieve major productivity advantages. The components that make up this layer can exist on a server machine, to assist in resource sharing. These components can be used to enforce business rules, such as business algorithms and data rules, which are designed to keep the data structures consistent within either specific or multiple databases. Because these middle-tier components are not tied to a specific client, they can be used by all applications and can be moved to different locations, as response time and other rules require.

* EC2 instance(both front end and running ultimately running SPARK in memory)
* Amazon Web Services
* Elastic Load Balancer
* Business Logic

1. R/R-studio (package 1, package 2, …)
2. SPARK
3. Scala

**Layer 3 Technologies (Data Layer)**

This is the actual DBMS access layer. It can be accessed through the business services layer and on occasion by the user services layer. This layer consists of data access components (rather than raw DBMS connections) to aid in resource sharing and to allow clients to be configured without installing the DBMS libraries and ODBC drivers on each client. Redshift will be the major contributor here, allowing for scalable growth by simply creating new clusters, while allowing for high value employees to be better utilized. Redshift also frees up Peekin from the licensing fees associated with more proprietary data warehousing solutions. Pinterest which is a company with one of the highest valuations per employee, at a possible IPO of $30 billion, has one database administrator managing their entire analytics database infrastructure.

* Hadoop/MapR for ad hoc Business Analytics

1. –Hive/Hbase
2. MySQL

* Amazon Redshift
* Amazon S3 Scalable Storage

1. Buckets (4)

# Statistical Model and Method

## Overview

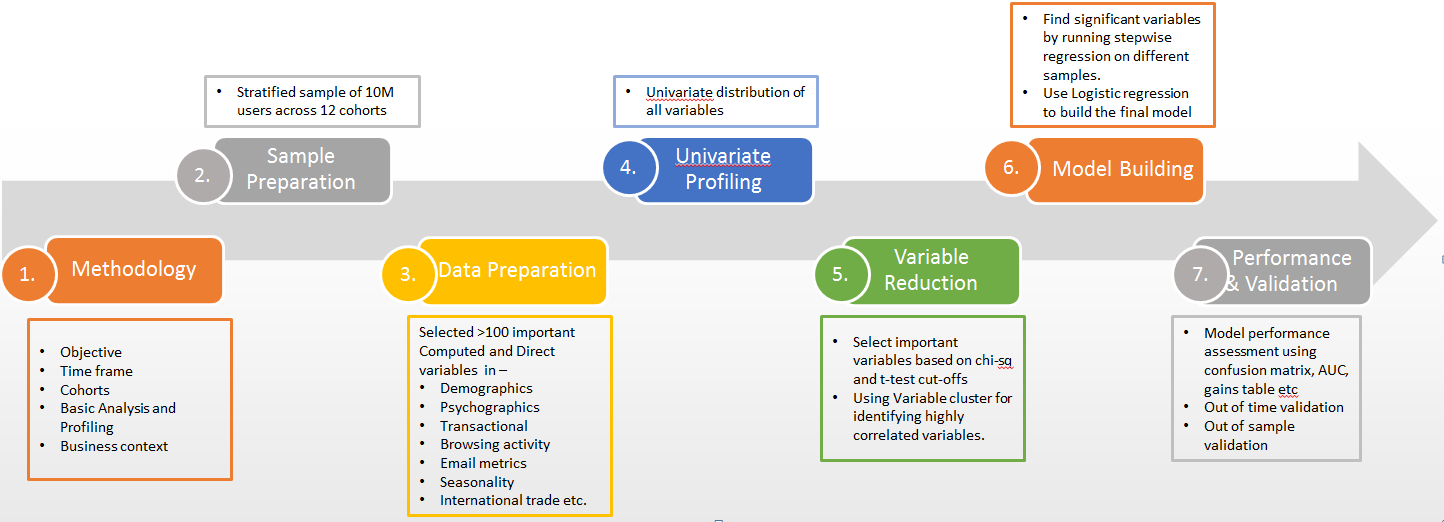
Transactional data, Gender, Psychographic data, Customer Demographics, Browsing Data, Product Data, Coupons/Discounts data, Feedback data, Social Media data, Survey Data etc. will be uploaded to the Redshift database.

The attributes of the Feedback data will include Feedback ratings based on experience of product booking on the website (ie restaurant rating), quality of product, quality of service etc. The attributes of the transactional data will then be included in customer’s transactional history details, average transactions over a specific time period, ticket size, last time checked into the geographic location, user frequency and monetary value of transactions.

The attributes of the Psychographic data include customer’s interest (hobbies) data for targeted marketing. The attributes of the Demographic data include information on the customer’s age, education, income level, gender. The attributes of the Browsing data include information on the customer’s visits to the website, source of visit (marketing channel touched), funnel metrics etc.

Objective: Customers who will make a purchase of product in the when served with an ad from an advertiser. This will form the dependent variable/output.

**Steps involved in Model building-**

****

### Analysis Attributes

1. Feedback Score
2. Type of Customer (Promoter/Detractor of the firm)
3. Average Transactions
4. Average Revenue
5. Transaction in M0 (most recent month)
6. Savings (by transacting on the firm’s products)
7. Average Product Rating
8. Gender
9. Age of customer
10. Feedback comments (Sentiment-Positive/Negative)
11. Income Level
12. Indicator of Interest
13. Marketing Channel Touch
14. #Channels Touched
15. Ads and Banner level data
16. Mobile related data
17. Coupons/Discounts usage
18. Transaction Month Indicator
19. Proportion of Weekend/Weekday Transaction
20. Seasonality

### Sample and Data Preparation

We would use a stratified sample of 10M users across different cohorts (this will include all the active customers in the last 2 years). We will have a total of 12 cohorts with a rolling window. This will help to ensure our model holds good irrespective of the month they are ran in. Predictive models are built on samples and then tested whether they hold well (perform well) when tested out of time or out of sample or for the entire population. Initial data set will include all possible Standalone, Derived and Interaction variables.

### Data Profiling and Variable Reduction

This will be performed for all variables as part of the exploratory data analysis. The appropriate statistic depends on the level of measurement. Depending on the type of variable (nominal/ordinal/interval-type), a frequency table or central tendency/range or arithmetic mean and standard deviation would be appropriate.

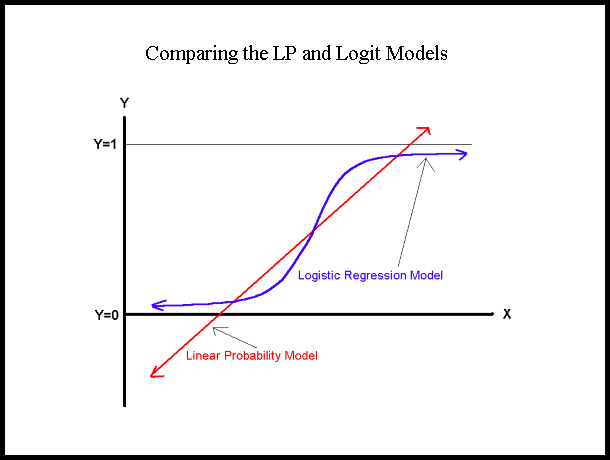
Bi-variate analysis will be performed with respect to click which is the dependent variable. This will help to determine the empirical relationship between the dependent variable (Click) and each of the independent variables. Bi-variate analysis would be performed in the form of a percentage table or scatter plot graphs.

Variable Reduction would be performed using ANOVA functions in Scala’s MLlib which performs variable clustering in order to identify highly correlated variables. 1 variable would be selected from each cluster which would best describe the cluster characteristics. Apart from this, important variables can be selected based on chi-square and t-test cut-offs.

### Logistic Regression for model building

We use logistic regression to estimate the probability that the customer will make a purchase of the specific product over the next 3 months. The end result will be a probability value and a cut-off would be needed to identify purchase vs no purchase.Since our dependent variable for this particular research topic is discrete (and not continuous) we would use the binary logistic regression. In other words, the logistic regression model is simply a non-linear transformation of the linear regression.

**Logistic Regression:**

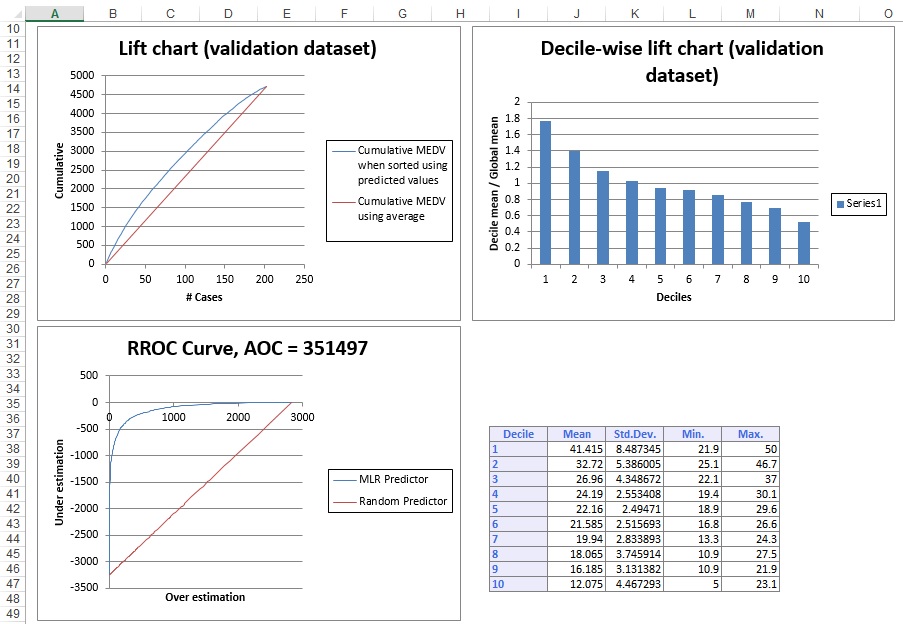
.

### Performance And Validation

Once the model to identify whether a purchase would be made or not has been built, the next step is to test and validate the model using a different time period (Out of Time Validation) and find whether the model behaves correctly (in terms of identifying higher purchase propensity buyers) or starts to deteriorate.

Various Model Performance parameters like Accuracy, Hit-Rate, Precision, Lift Chart, Gains chart, ROC, specificity and sensitivity etc. Confusion matrix will help to find the number of true positives, true negatives, false positives and false negatives for our model.

**Model Performance Charts:-**



### Count Regression Model for Repeat Purchases

This would be a separate predictive model to predict repeat purchases for each customer who has a high propensity to buy. This will be an extension to our purchase propensity model where we predict whether a customer will make a purchase or not and then as a next step we predict the number of repeat purchase the customer would make. For this purpose, various Count Regression Models like negative binomial, poison regression, OLS linear regression would be used to predict repeat purchases for each customer accurately.

### Text Mining Analytics

This will form a separate analysis to understand what a specific customer feels about the product and whether he has a positive/negative sentiment about it. Text mining algorithm like LDA, NMF, clustering etc. would be performed using Scala on these feedback comments. This would first help to identify the top trending topics which the customers are talking about. Unigram and Bigrams analysis would help to identify themes with each of the topics.

Sentiment Analyzer tool using Scala would help to understand if there is a positive or negative sentiment associated with each customer comment. This data can be aggregated at theme/topic level to know if sentiment is more positive or negative for each theme/topic.

Right Business decisions can be made using the above predictive model and analysis by taking into account conversions, investments, budgets and other important things.

# Solution Components

### Hardware

This is cloud based. Describe the cloud and how it will be employed in your solution; include how you will connect to the cloud (e.g., HTTP/HTTPS, Virtual Private Network, Direct Connect). This is a good opportunity to discuss any security concerns (i.e., Risks) and how you will mitigate.

All of the security for the cloud will be managed at the top level by AWS’ Identity and Access Management. This will allow for a centralized console where all access identity, keys, and roles are configured

Peekin streams hundreds of millions of application logs each day. The company relies on analytics to report on its daily usage, evaluate new offerings, and perform long-term trend analysis—and with millions of new check-ins each day, the workload is only growing.

But the database system Peekin used for analytics comes with high annual licensing costs, and required the company to spend staff time keeping the system running. Free company from licensing fees and free up important and scarce staff

### Software

#### Solution Specific Software

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Manufacturer** | **Title** | **Procurement** | **Ownership** | **Installation** | **Support** |
| Amazon | Cloudwatch, can access through Amazon SDK on clusters | Free | Company | automatic | Amazon |
| Amazon | Redshift | 1TB/$1000 per year | Company | User | Amazon |
| Amazon | EC2 instances for Web interface, Spark, data processing (Revolution R , Spark binaries, R scripts that push relevant data) | Cheap and scalable | Company | User | Amazon |
| Amazon | S3 | Cheap and scalable | Company | User | Amazon |
| Amazon | IAM | Free | Company | User | Amazon |

### Network

A network interface controller (NIC, also known as a network interface card, network adapter, LAN adapter, and by similar terms) is a computer hardware component that connects a computer to a computer network. A cloud instance will typically have several Network Interfaces numbered EN0, EN1, etc. They can be used for a dedicated purpose, such as backups (EN1 below) or management (RSA below).

1. EN0: Public Access

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Discription** | **IP Address** | **Subnet Mask** | **Masking Bits** | **VLAN** |
| EN0 |  |  |  |  |
|  |  |  |  |  |

1. EN1: Backup and Recovery Interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Discription** | **IP Address** | **Subnet Mask** | **Masking Bits** | **VLAN** |
| Backup |  |  |  |  |

1. EN2: Backup and Recovery Interface
2. EN3: Backup and Recovery Interface
3. RSA: Remote Supervisor Adapter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Discription** | **IP Address** | **Subnet Mask** | **Masking Bits** | **VLAN** |
|  |  |  |  |  |

### Monitoring

AWS Monitoring options: <http://aws.amazon.com/cloudwatch/>

AWS Monitoring may help predict possible faulty clusters that may result in outage time. Furthermore Cloudwatch also allows for *predictive* redundancy and efficient CPU utilization, by allowing the administrators to administer status checks and set alarms for failed hardware. This is important as most of the hardware we are working on with from the S3 cluster, and data warehouses are NOT high end RAID, SSDs etc. It is commodity hardware with a relatively high failure rate. AWS Monitoring

### Backup and Recovery

AWS Backup and Recovery options: <http://aws.amazon.com/backup-recovery/>

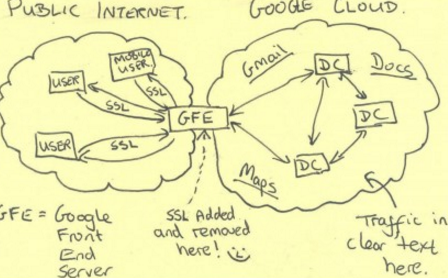
Redundancy is built into not only all of Amazon’s products, but many times over. IE for a Redshift account of 1TB, a redundancy of 3TB, and a 1TB S3 alottment, in addition to existing S3 storage you may have.

Front End Servers and Datawarehousing of production will be backed up nightly for tables/data that may have changed.

Processing heavy distributed clusters will not be backed up. The data will be in memory and is not worth backing up let alone attempting to recover on data that is used to make decisions so quickly. Only the machine learning algorithm and relevant data from the logs need to be backed up.

Considering the app use will most likely be during Friday and Saturday, the peak of nightlife. Predictive algorithms that are used over the long term, as opposed to live backups, should be backed up AT LEAST weekly

SSL VPC allowing for data encryption within and between servers from government snooping as well. Offers protection and time against legal oversight in tricky situations. This can be important in countries with less transparent reasons for accessing data (ie in 2010 China shutdown Foursquare for good after people starting checking into Tianamen Square as a form of protest). :



**Glenn Greenwald**

### Disaster Recovery

As is standard with most Hadoop nodes, three different backup nodes dynamically for web servers and front end user queries.

Datawarehousing will have redundancy backups at major analytical milestones (table creation, updating) and is dependent on the type of analysis going on. Viability Assessment

## Assumptions

|  |  |  |  |
| --- | --- | --- | --- |
| **Assumption ID** | **Finding / Assumption Description** | **Confidence Level (H/M/L)** | **Impact (H/M/L)** |
| A01 | Geo-location data available (ie Google doesn’t cut them off) | H | H |
| A02 | AWS Services Will Be Secure and Segmented | H | H |

**Note:** record any ASSUMPTION with a LOW CONFIDENCE and HIGH IMPACT as a RISK.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Risks **Risk ID** | **Finding / Risk Description** | **Probability (H/M/L)** | **Effort / Cost** | **Impact (H/M/L)** | **Contingency / Mitigation Recommendation** |
| R01 | Data will remain confidential and anonymized (issue of trust and possibly legal oversight ) | H | high | H | Governments/advertisers |
| R02 | AWS IAM Management | H | low | H | Keep keys secure in a centralized and with different privileges |

Note: the following definitions are suggested.  Of course, the exact definition is not as important as agreement by the team on a definition.

Probability:

* High - the risk identified is probably going to occur, or is already occurring.
* Medium - the risk identified is about as likely to happen as not.
* Low - the risk identified is unlikely but still worth considering.

Impact:

* High - resolution is likely to require difficult decisions, probably above the level of the project manager, which are likely to affect the schedule, budget or functional completeness of the project.
* Medium - special management attention is required, but it should be possible to contain the risk within the project plan.
* Low - normal management attention should be sufficient to resolve the issue.

## Functional Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk ID** | **Finding / Risk Description** | **Probability (H/M/L)** | **Impact (H/M/L)** | **Contingency / Mitigation Recommendation** |
| FR01 | **The system shall access SPARK for system availability** | H | H | Spin up another instance immediately |
| FR02 | **The system shall employ redundant servers** | H | H | Spin up another instance immediately |
| FR03 | **The system shall log all transactions and maintain the logs for later analysis** | H | L | Spin up another instance immediately |

Note: In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions.

Broadly, functional requirements define what a system is supposed to *do* and non-functional requirements define how a system is supposed to *be*. Functional requirements are usually in the form of "system shall do <requirement>", an individual action of part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, and process and control functional model. In contrast, non-functional requirements are in the form of "system shall be <requirement>", an overall property of the system as a whole or of a particular aspect and not a specific function. The systems' overall properties commonly mark the difference between whether the development project has succeeded or failed.

Probability:

* High - the risk identified is probably going to occur, or is already occurring.
* Medium - the risk identified is about as likely to happen as not.
* Low - the risk identified is unlikely but still worth considering.

Impact:

* High - resolution is likely to require difficult decisions, probably above the level of the project manager, which are likely to affect the schedule, budget or functional completeness of the project.
* Medium - special management attention is required, but it should be possible to contain the risk within the project plan.
* Low - normal management attention should be sufficient to resolve the issue.

## Non-Functional Requirements

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Risk ID** | **Finding / Risk Description** | **Probability (H/M/L)** | **Effort / Cost** | **Impact (H/M/L)** | **Contingency / Mitigation Recommendation** | **Person Responsible** | **Review Date** |
| NFR01 | **The system shall be accessible 24/7** | H | M | H | Spin up a new cluster/load backed up image | Database Admin/Network Admin |  |
| NFR02 | **The system shall be highly available** | H | MH | H | Amazon’s Elastic Balancing |  |  |

Note: In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions.

Broadly, functional requirements define what a system is supposed to *do* and non-functional requirements define how a system is supposed to *be*. Functional requirements are usually in the form of "system shall do <requirement>", an individual action of part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, and process and control functional model. In contrast, non-functional requirements are in the form of "system shall be <requirement>", an overall property of the system as a whole or of a particular aspect and not a specific function. The systems' overall properties commonly mark the difference between whether the development project has succeeded or failed.

Note: the following definitions are suggested.  Of course, the exact definition is not as important as agreement by the team on a definition.

Probability:

* High - the risk identified is probably going to occur, or is already occurring.
* Medium - the risk identified is about as likely to happen as not.
* Low - the risk identified is unlikely but still worth considering.

Impact:

* High - resolution is likely to require difficult decisions, probably above the level of the project manager, which are likely to affect the schedule, budget or functional completeness of the project.
* Medium - special management attention is required, but it should be possible to contain the risk within the project plan.
* Low - normal management attention should be sufficient to resolve the issue.

## Dependencies

|  |  |  |  |
| --- | --- | --- | --- |
| **Dependency ID** | **Finding / Dependency Description** | **Effect on Plan** | **Associated Risk ID** |
| D01 | AWS infrastructure and cloud service | The product cannot be deployed unless | R01 |
| D02 | Geo-data not available | No way to collect data or run analytics. Risk is multiplied when you rely on another company like Google or Facebook for access to that data ie in earlier stage startup. | R02 |

Notes: Record any external DEPENDENCIES (deliverables to be provided to the Project from areas outside the control of the Project).

* Describe the EFFECT on PLAN in the form 'x' cannot start/complete until 'y' delivered.
* If the DEPENDENCY is required by a specific DATE, record it.
* Name the OWNER responsible for the delivery of an external DEPENDENCY.