

WITS UNIVERSITY

SCHOOL OF ELECTRICAL AND INFORMATION  
ENGINEERING

ELEN7046 - SOFTWARE TECHNOLOGIES AND TECHNIQUES

---

# Big Data Visualization using Commodity Hardware and Open Source Software

Individual Report: Twit-Con-Pro Solution

---

*Author:*

Sidwell MOKHEMISA

*Student Number:*

1229756

Shared Project GitHub Repository:


[https://github.com/garethstephenson/ELEN7046\\_Group2\\_2016](https://github.com/garethstephenson/ELEN7046_Group2_2016)

July 3, 2016

## DECLARATION OF ORIGINALITY

I, Sidwell Mokhemisa, student number: 1229756, hereby declare the following:

- I am aware that plagiarism (the use of someone else's work without their permission and/or without acknowledging the original source) is wrong.
- I confirm that ALL the work submitted for assessment for the above course is my own unaided work except where I have explicitly indicated otherwise.
- I have followed the required conventions in referencing the thoughts and ideas of others.
- I understand that the University of the Witwatersrand may take disciplinary action against me if there is a belief that this is not my own unaided work or that I have failed to acknowledge the source of the ideas or words in my writing.

Member Name	Primary Task	Time Spent (Hours)	Discretionary	Member Signature
Sidwell Mokhemisa	Architecture	111	5 %	

## ABSTRACT

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Approach . . . . .	4
1.2	Background . . . . .	4
<b>2</b>	<b>Lifecycle Methodology</b>	<b>4</b>
<b>3</b>	<b>Assumptions and Constraints</b>	<b>6</b>
<b>4</b>	<b>Design Decisions</b>	<b>7</b>
<b>5</b>	<b>Requirements - Use Case Models</b>	<b>7</b>
5.1	View Elections Analytic Data . . . . .	7
5.2	Acquire Twitter Data . . . . .	9
<b>6</b>	<b>Solution Design</b>	<b>12</b>
6.1	High Level Design: Component Architecture . . . . .	12
6.2	Solution Sequence Diagrams . . . . .	14
6.3	Operational Model: Infrastructure Design . . . . .	14
<b>7</b>	<b>Conclusion</b>	<b>16</b>
<b>8</b>	<b>Appendices</b>	<b>17</b>
8.1	Appendix A: Individual Time-Sheet . . . . .	17
8.2	Appendix B: Non-Functional Requirements . . . . .	17
8.3	Appendix C: Lifecycle Methodology . . . . .	17
8.3.1	Inception Phase . . . . .	17
8.3.2	Elaboration . . . . .	17
8.3.3	Construction . . . . .	17
8.3.4	Transition . . . . .	17
8.4	Appendix D: List of Tools and Techniques . . . . .	17

# **1 Introduction**

This document covers a significant amount of deliverables that are key to the delivery of most projects that follow a formalized and structured software development lifecycle.

Deliverables covered in this topic are in the main architectural in nature with greater focus on High Level Design deliverables such as requirements, component model and Sequence Diagram(s) (Software Architecture), and Operational Model/Physical Infrastructure Design.

## **1.1 Approach**

The project shall be delivered by firstly trying to understand the requirements and modeling them into a set of Use-Cases which can then be used to direct the project from analysis to design, development, validation and verification, and benefits tracking once the system is taken into production and starts adding value to its end-users.

## **1.2 Background**

This project was conceptualized to solve for a problem that is one of the key problems for both businesses and academic institutions as we move forward into this new world that is at the cutting edge of innovation and extreme levels of data available.

The scope of the project shall be limited to acquiring data from Twitter (based on topic subscriptions - in this case, US and SA elections data for 2016), taking it through the process of transformation and analysis, and later making it available to users in a meaningful way using some of the cutting edge visualization tools that are relevant to solving the problem at hand.

# **2 Lifecycle Methodology**

The initial assessment of the project based on the project description and high level requirements showed the team that the project to be delivered can be classified

as low-to-medium in terms of the classification scheme for projects based on the following broad categories:

- **Low** - This means the project to be delivered is classified as having low complexity and either small/ small-to-medium in terms of the budget and size of the delivery team.
- **Medium** - This means the project is classified as having medium complexity with medium budget and delivery team-size.
- **High** - This mean that the project is classified as being fairly complex, and requiring a huge budget and workforce for the success of its delivery.

It was for the reason of small-to-medium classification that the project followed a formal, plan-driven approach for the delivery of this project, based on IBM RUP with tailoring in order to make a determination/ up-front decision regarding the artifacts that were to be de-scoped, delivered at a level of detail sufficient to allow for the next phase(s) to continue, or to deliver a comprehensive artifact.

The diagram below depicts the IBM RUP model[1]:

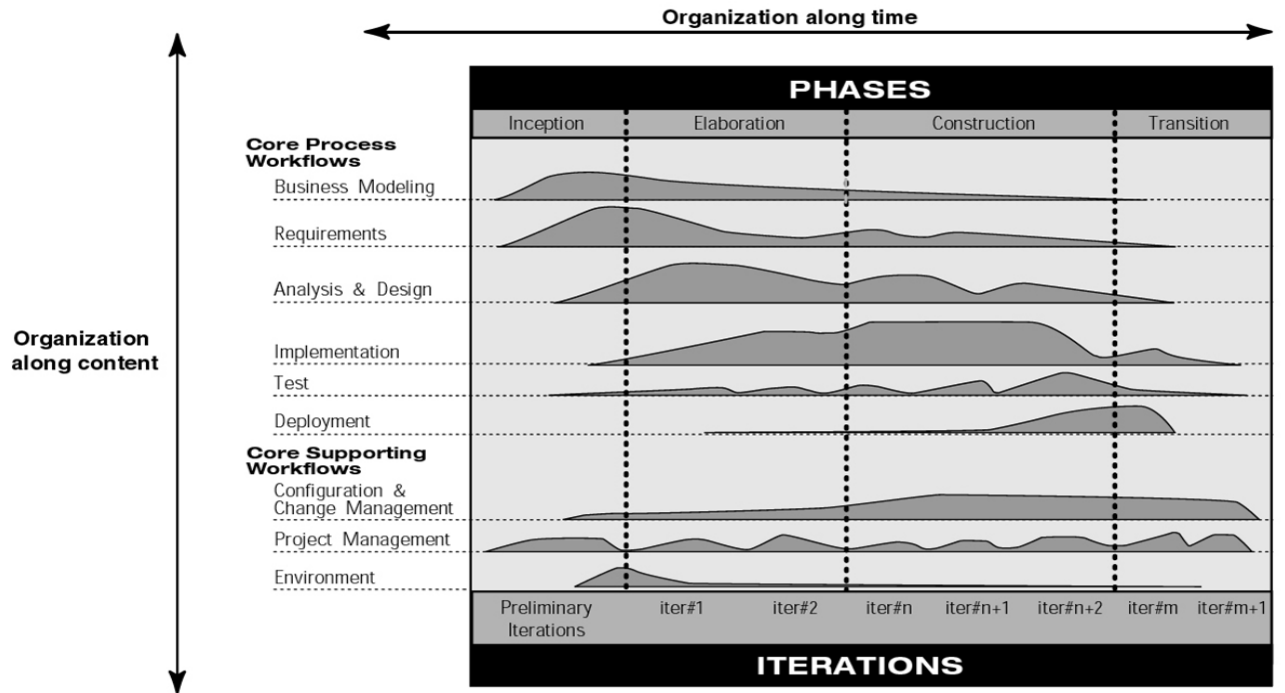


Figure 1: IBM Rational Unified Process (Source: RUP, Best Practices for Software Development Teams)

### 3 Assumptions and Constraints

#### Assumptions:

The key assumptions made for this project are as follows:

- It is assumed that all the resources required to deliver this project are available and will be dedicated to the project from inception to transition.
- It is also assumed that the infrastructure planned for this initiative will be able to cater for the anticipated volumes.

#### Constraints:

The following are the constraints identified during High Level Design:

- Limitation of the source system for the provision of social media and location data for non-commercial purposes.

- Time related constraints for the delivery of a working solution by the end date.

## 4 Design Decisions

The key design decision that was made in this project related to the tailoring of RUP artifacts in order to ensure success in the delivery while pitching the content of the deliverables at a sufficient level of detail to enable down-stream teams to continue with their work.

The other major decision made was that of allowing for development as well as parts of production systems to run independent from each other and at different geographical areas to ensure continuity, and contribute to project service level characteristics of high availability and disaster recovery, albeit, not fully disaster recovery proof.

## 5 Requirements - Use Case Models

The key Use Cases as identified by the project are discussed in this section.

### 5.1 View Elections Analytic Data

This section covers the details around the visualization Use Case. The diagram below depicts the actual Use Case followed by a table that further discusses the Use Case details:



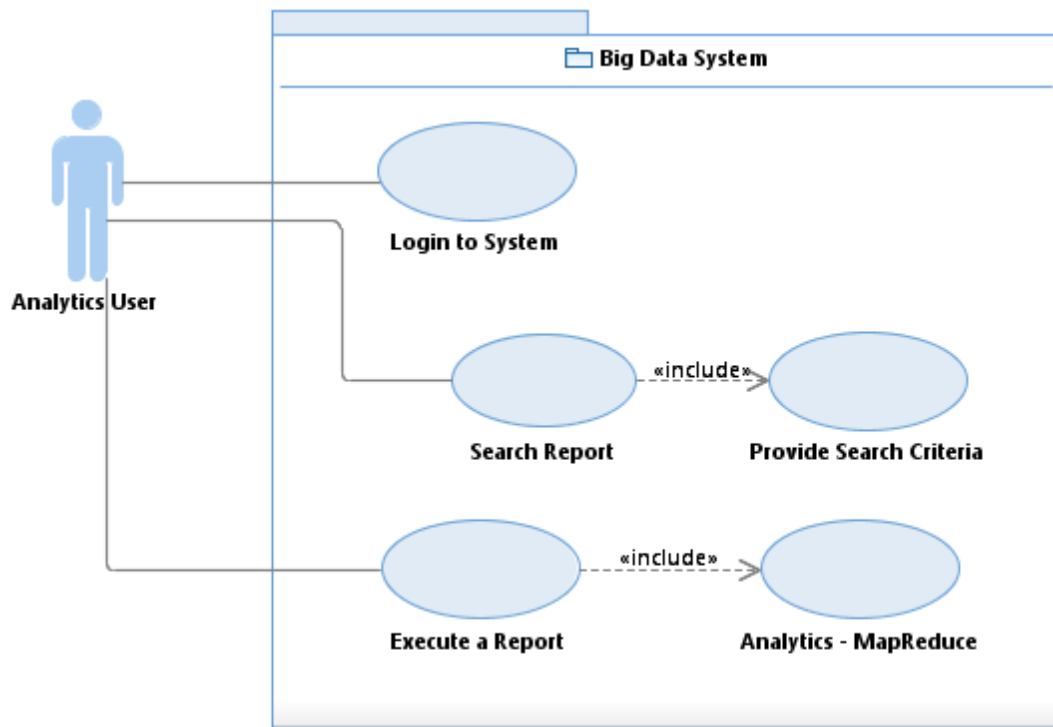


Figure 2: Use Case Diagram - View Twitter Elections Analytics

This table provides additional information to supplement the Use Case diagram.

Use Case ID:	UC01		
Use Case Name:	View Analytics Social Media Report overlaid on map background.		
Created By:	Sidwell	Updated By:	Sidwell
Date Created:	02/05/2016	Date Modified:	07/05/2016

Actor :	Analytics User
Description:	This use case describes how the user will use the system to run analytics based on social media data received from Twitter.
Pre-conditions:	Web browser opened and user logs onto the analytics site.
Post-conditions:	User views requested report overlaid on map background. Drill up/down functionality provided by the application.
Normal Course:	1. Logon to the application. 2. Search report from list of available reports 3. Execute a report of choice
Frequency of Use:	
Alternative Courses:	None
Exceptions:	None
Includes:	1. Provide search criteria or Hashtag(s). 2. System runs report using Map Reduce and parallel processing in order to produce report results.
Special Requirements:	1. Ad-hoc access using most browsers (IE, Chrome, Safari).
Assumptions:	1. User login based on access to computer with browser and not necessarily integration to an LDAP compliant system. 2. Support for mobile apps once developed.
Notes and Issues:	

## 5.2 Acquire Twitter Data

This section covers the details around the data acquisition Use Case. The diagram below depicts the actual Use Case followed by a table that further discusses the Use Case details:

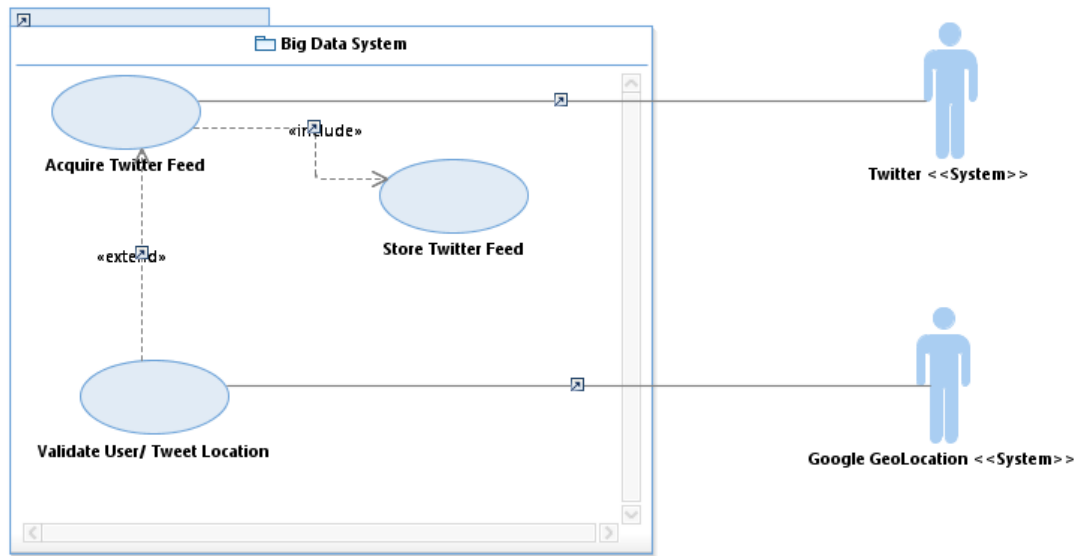


Figure 3: Use Case Diagram - Acquire Twitter Data

This table provides additional information to supplement the Use Case diagram.

Use Case ID:	UC02		
Use Case Name:	Acquire social media feed from twitter to enable big data analytics.		
Created By:	Sidwell	Updated By:	Sidwell
Date Created:	02/05/2016	Date Modified:	07/05/2016

Actor :	Twitter and Google GeoLocation
Description:	This use case describes how data is collected from twitter based on subscribed topics stored in a database for later use in analytics processing.
Pre-conditions:	1. Application logs into twitter with provided credentials and starts streaming all the data that complies with subscribed topic(s). 2. Topics to subscribe to are configured on the system beforehand. 3. For each tweet streamed, the application through its orchestration service attempts to verify location from which tweet was sent, or from profile of user sending twitter using Google GeoLocation Service. 4. Where location could not be verified, the tweet is stored in the database without location information.
Post-conditions:	1. Developed application authenticates and streams data. 2. Streamed data is stored in the database with location information where location could be determined.
Normal Course:	1. Configure election related topics to subscribe to (both US and SA). 2. Allow application to log onto both twitter and Google GeoLocation. 3. Stream tweets through orchestration service while attempting to verify location by validating certain data via Google GeoLocation Service. 4. Store all tweets regardless of location information availability.
Frequency of Use:	
Alternative Courses:	None
Exceptions:	None
Includes:	1. Storing of tweeter feeds in a database.
Special Requirements:	1. Username token provided by twitter. 2. Username token provided by Google GeoLocation Service. 3. Internet access to connect to both services.
Assumptions:	1. Availability of infrastructure resources to harvest more than a million twitter records and store them.
Notes and Issues:	

## 6 Solution Design

### 6.1 High Level Design: Component Architecture

The high level design below depicts the key functions of the Twit-Con-Pro solution with special focus given to components relating to acquisition, processing and visualization of Big Data from Twitter that relates to both the American and South African elections for the year 2016.

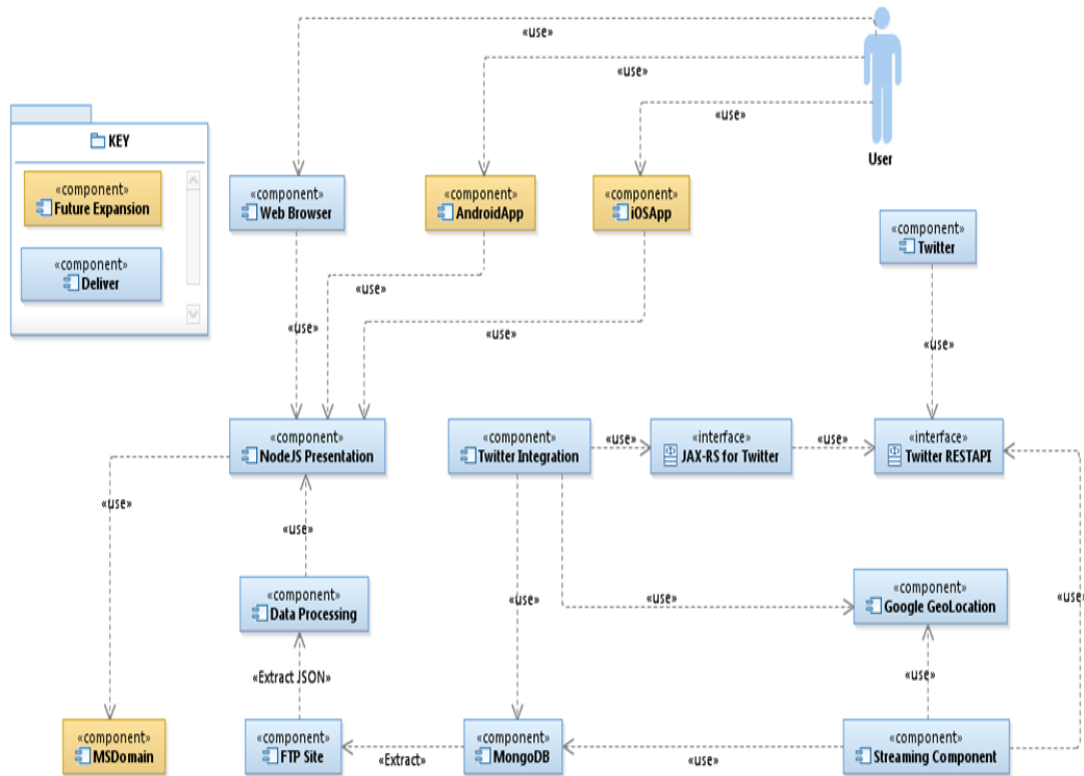


Figure 4: High Level Component Model.

The table below provides a descriptive detail of all the key components that make-up the solution from data acquisition through to visualization.

Component	Key Function of Component
Twitter	This component is the key provider of social media data for analysis and processing based on topics subscribed to. For this project, it was decided that topics subscribed to be centered around top political parties in the US and RSA elections.
TwitterRESTAPI	Twitter exposes a REST API for those interested in its publicly available data to integrate into their systems and acquire the data. Some limitations around how much data can be retrieved at any given time are imposed.
JAX-RS for Twitter	This solution, through this project shall deliver an interface based on JAVA API for XML Representational State service in order to integrate into Twitter for the acquisition of history data based on a date range.
Twitter Integration	This component is responsible for integrating into Twitter, orchestrating the validation and resolution of location information to Geographical Co-ordinates where location details are available as part of Twitter user preferences and persisting all the data on MongoDB.
Streaming Component	The standalone, windows based component shall also provide the same functionality as the Twitter Integration component; however, in this case, data being acquired shall be fresh data, streamed live as it happens.
MongoDB	This open source database was used for storing all the Big Data acquired from Twitter, both history data and online streaming data.
FTP Site	Periodically, a job shall be run to extract the data in the database and transfer it to the in JSON files for further processing on the Raspberry Pi cluster.
Data Processing	This component is dedicated to doing all the number crunching using Apache Spark while applying principles of MapReduce to ensure that all the Big Data is split and shared amongst all the Raspberry Pi cluster worker nodes.
Node.JS Presentation	This component shall read all the processed data and render it in a Web Browser in a manner that tries to communicate to the user the sentiments of potential voters for the upcoming 2016 elections in both the United States and South Africa.
Web Browser	The user interacts with the data using the Web Browser.
User	It is important to note that the user in this instance represent both the end-user who is interested in the elections analytics from Twit-Con-Pro and the admin-user who is responsible for the initial setting up of topics that are to be subscribed to.

## 6.2 Operational Model: Infrastructure Design

As the title of the report suggests, this project is envisaged to leverage the use of low cost, commodity hardware in order to make a case for participating in Big Data projects without having to rely on big budgets and enterprise platforms.

The below diagram depicts the production view of the end-to-end solution. In this diagram, it can be shown that the solution was running at completely separate geographical locations, even though at the heart of it (the number crunching and data analysis) all the nodes participating in the cluster were co-located at one geographical area.

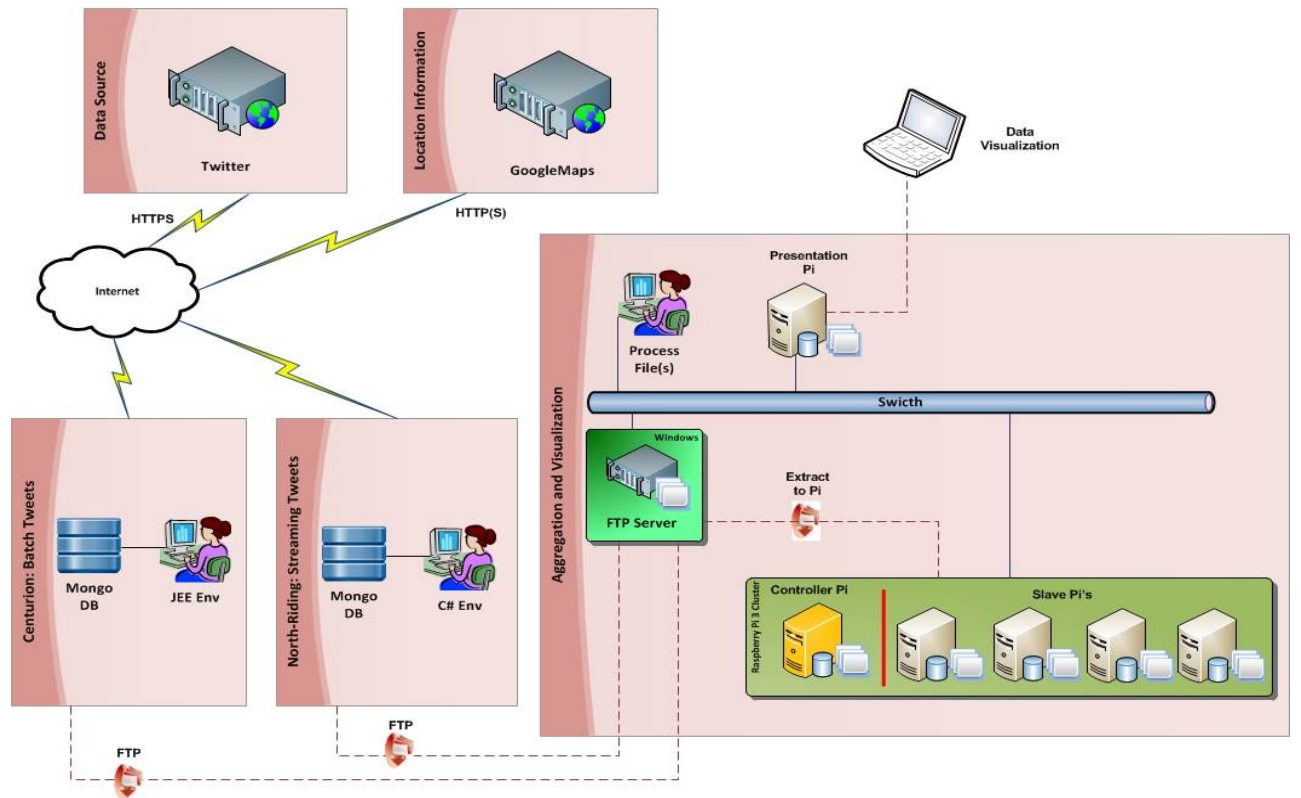


Figure 5: Operational Model: Physical

The breakdown of the solution at an operational level was as follows for the production environment:

- **History Batch Data** - All batch data acquisitions from Twitter will be

based in Centurion. Data acquired from Twitter shall be persisted in a MongoDB locally in that physical geographic area to avoid latency.

- **Streaming Data** - Streaming data shall be processed from the North-Riding location while data persistence is done locally to that site using MongoDB as well.
- **Aggregation and Visualization** - Most of the data "heavy-lifting" and number crunching shall happen using a cluster of Raspberry Pi's which are a very small pocket-sized "servers". The location information has been deliberately left out in this scenario because this part of the solution has been designed with high-mobility in mind.

## 7 Conclusion

All this hardware and software is available to anybody interested in Big Data processing.

The hardware is cheap and the software is free.

The learning curve in the beginning can be quite steep but is ultimately very rewarding in terms of what can be achieved with so little financial investment.



## References

- [1] . S. Madam. From Databases to Big Data. IEEE Computer Society, 2012.
- [2] . V. Kumar, R. Yuvaraj, C. Anusha. Effective Distribution of Large Scale Datasets Clustering Based on MapReduce. 2016.

## 8 Appendices

### 8.1 Appendix A: Individual Time-Sheet

### 8.2 Appendix B: Lifecycle Methodology

#### 8.2.1 Inception Phase

#### 8.2.2 Elaboration

#### 8.2.3 Construction

#### 8.2.4 Transition

### 8.3 Appendix C: Non-Functional Requirements

### 8.4 Appendix D: Solution Sequence Diagrams

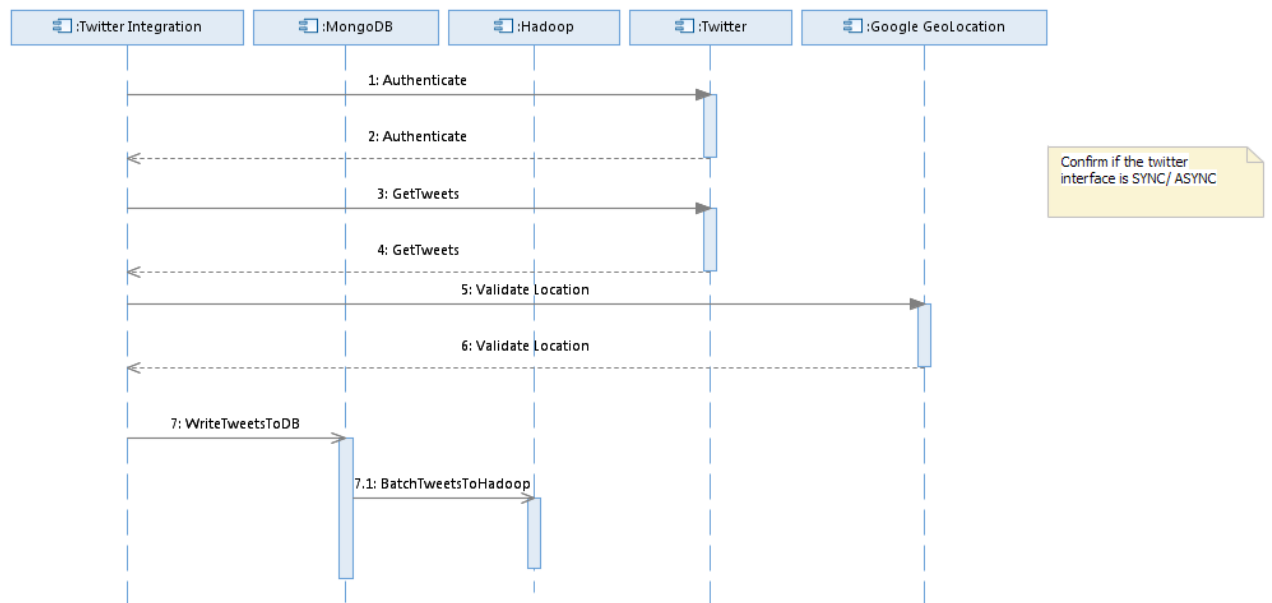


Figure 6: Sequence Diagram: Data Integration - History

### 8.5 Appendix E: List of Tools and Techniques