# Table of Contents *TEAM PYTHIA*

Requirements October 6, 2016

Team Name: Pythia Version = Rough Draft

Team Members:

| Name |
| --- |
| Clement Cole |
| Christopher Roche |
| Elijah Adedapo |
| Enrique Torres |

# A. PROJECT DRIVERS

## A1. Purpose of the Project

### A1a. Background of Project Effort

The document details the functionality required for the design of a trading system using Field Programmable Gate Arrays (FPGAs).

Financial traders buy and sell financial data instruments (shares in companies) for their respective clients. As a result, they will anticipate to buy stocks at lower prices and sell them when the stock is worth significantly higher than the current price.

In order to accomplish such task, financial traders will use several quantitative methods to evaluate current and historic values for the respective stock's parameters and historical data streams to help predict the future values of such parameters.

The type of data streams for stock values is of stochastic in nature. In order to correctly predict the daily average highs and lows, certain algorithms will have to be implemented on a software based system (Von Neumann based architectures) or hardware based system (reconfigurable logic circuit architectures).

In order to latencies accounted for in software based systems that impedes the process of buying and selling of stocks by stock brokers, the group will design a project implementing certain algorithms for predicting stock prices (daily average high and lows) using both FPGA components and software based software based systems.

Hence create a product that produces low latency prediction of daily average highs and lows of a specified stock that will assist stock brokers and or special interest groups in forecasting future daily highs and lows for the specified stock.

### A1b. High Level Goals of Project

**FPGA trading Infrastructures provides:**

- Reduce latencies between computations as compared to other computer based architectures.
- Provide high computational speeds compared to other processes.
- Predict stock prices (Daily Average High and Daily Average Lows).
- Use a web server or repository to feed the specified stock parameters to enable further computations in real time.
- Identify and implement a stock prediction algorithm on both FPGA and software based system.
- As a test measure for proof of concept, measure the performance in terms of speed for both FPGA and software based systems.
- The development of this project will enable stock traders to save time and money when investing in certain commodities for their specified clients.

# A2. The Stakeholders

## A2a. Executive Sponsor

**Dr. Robin Pottathuparambil**

## A2b. Customer

**Dr. Robin Pottathuparambil**

## A2c. Team Members

**Clement Cole** – **Engineer/Programmer**
**Christopher Roche** – **Engineer/Programmer**
**Elijah Adedapo** – - **Engineer/Programmer**
**Enrique Torres** – **Engineer/Programmer**

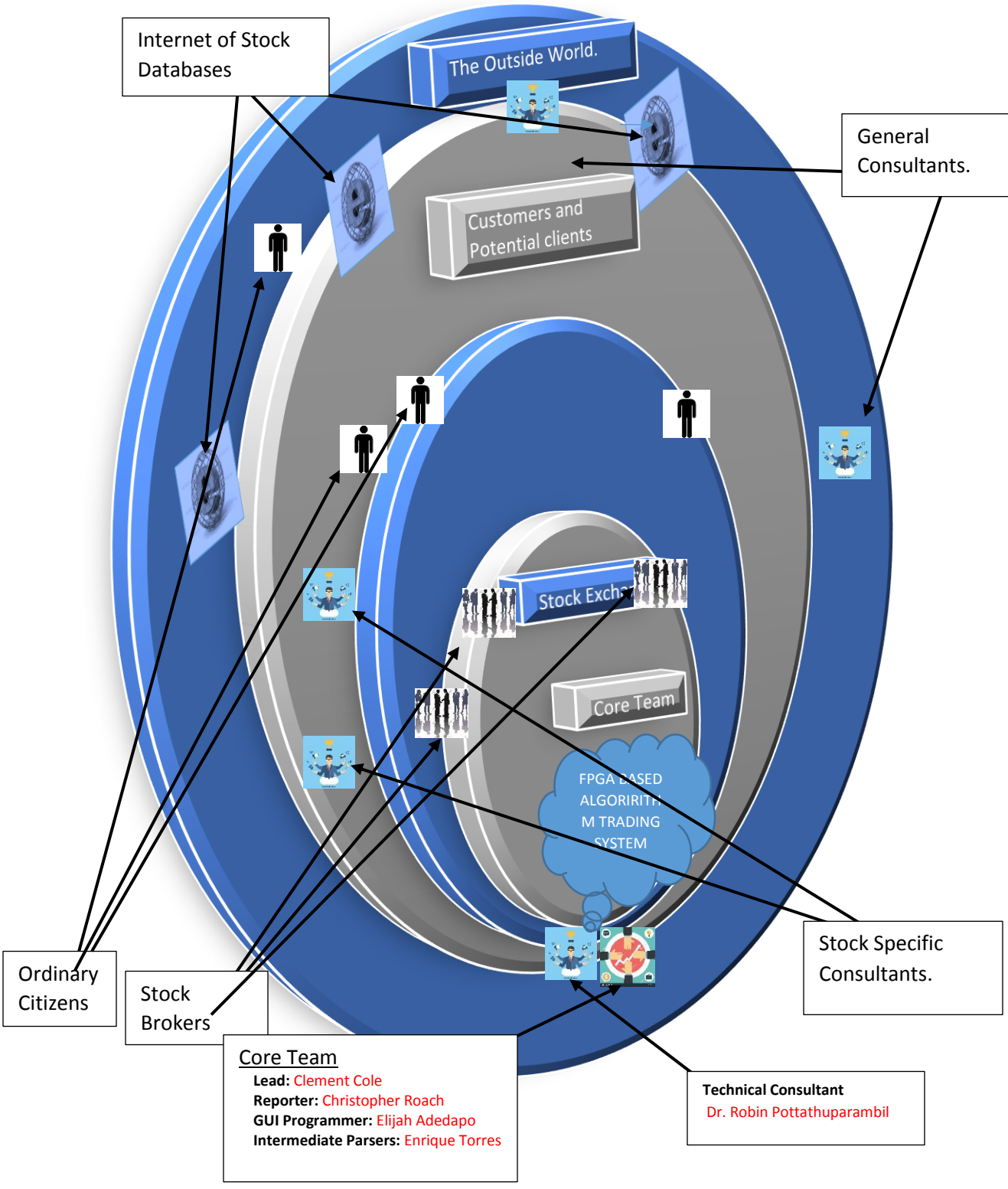
## A2d. Typical Technical (Hands-On) Actors and Roles

| Actors | Roles |
|---|---|
| **Arbitragers and Swing Traders** | Identifies and executes simultaneous sale and purchase of stocks to profit from a difference in prices. |
| **Floor Traders** | An exchange member who executes transactions from the floor of the exchange for their own account. |
| **Options Trader** | Actors who buy and sell options in the capital market. |
| **Agent/Security Lenders** | Agents or group of commissioned agents responsible for lending exchange-traded funds for transactional purposes by other traders or clients. |
| **Investment Dealers** | May act as a broker or dealer depending on the customer's request of whether to buy or sell the stock of a commodity. |

## A2f. Example of Stock Brokers

- **Yahoo stock brokers**
- **Facebook stock brokers**
- **Twitter stock brokers**
- **Apple stock brokers**
- **Coke stock brokers**
- **Tesla stock brokers**
- **Google stock brokers**
- **Microsoft stock brokers**
- **Xilinx stock brokers**
- **Intel stock brokers**
- **AMD stock brokers**

## A2g. Stakeholders Map and Relationship



## A2h. Technical Consultants to Project

**Dr. Robin Pottathuparambil**

# B. Constraints

## B1. Solutions Constraints

### B1a. Network Media

Constraints:

The product will use the current internet provider service for the University Of North Texas to retrieve data from the internet using its respective internet service provider.

Justification:

This above premise is due to the easy access to several mediums (wired/Wi-Fi) University's network via student user name and password.

Fit criterion:

The product shall be compatible to any medium for internet access (Wi-Fi / wireless).

### B1b. Hardware Board (FPGA)

- Constraints:

  The general purpose computer will generate random numbers for the FPGA unit due to the complexity in design for generating random numbers in FPGA.

  Justification:

  Most stock prediction algorithms require generation of random numbers.

  Fit criterion:

  This constraint will cause the FPGA to run at approximately the same speed as the general purpose computer in terms of generating random numbers. Although the speed of the FPGA will be reduced from a paralleled version, it will provide a reference point for measurement in comparison between FPGA unit and general purpose computer.

- Constraints:

  Configuration of Hardware (FPGA Board) will be in VHDL instead of verilog.

  Justification:

  VHDL is the language most members of the project team are familiar with.

  Fit Criterion:

  This will provide flexibility and compatibility in sub component designs by individual group members during the development phase of the project.

### B1c. Operating Systems for Software Based Systems

- Constraint:

  The group will utilize windows or Linux based operating systems.

  Justification:

  Both hardware and software system development aspects can be done on Linux or Windows architectures.

  Fit Criterion:

  Most users/customers in stock traders use analytic tools that are compatible with Linux and Windows based architectures.

- Constraint:
  Yahoo API written in python programming language will be used to scrape the data parameters or terminals from the internet.
  Justification:
  This is because the Yahoo API is the most reliable and tested API by the group and confirmed to work efficiently.
  Fit Criterion:
  A wrapper class written in C++ or any other system based programming language can be used for the yahoo api written in python.

- Constraint:
  The group will utilize windows or Linux based operating systems.
  Justification:
  Both hardware and software system development aspects can be done on Linux or Windows architectures.
  Fit Criterion:
  Most users/customers in stock traders use analytic tools that are compatible with Linux and Windows based architectures.

- Constraint:
  The group will utilize windows or Linux based operating systems.
  Justification:
  Both hardware and software system development aspects can be done on Linux or Windows architectures.
  Fit Criterion:
  Most users/customers in stock traders use analytic tools that are compatible with Linux and Windows based architectures.

- Constraint:
  The group will utilize github for version control.
  Justification:
  Github is standard and free. Also the group can utilize the private feature on the webapplication which is free for student.
  Fit Criterion:
  This will allow the group to work on the project remotely while still contributing significantly to the entire project.

- Constraint:
  The group will utilize windows or Linux based operating systems.
  Justification:
  Both hardware and software system development aspects can be done on Linux or Windows architectures.
  Fit Criterion:
  Most users/customers in stock traders use analytic tools that are compatible with Linux and Windows based architectures.

- Constraint
  The group will use standard text editor and compilers for the software development portion of the project.
  Justification:

Software development commonly in C++ has free compilers example GNU which is widely available.

Fit Criterion:

This standard will enable comparability in execution of the software implementation of the chosen algorithm to execute on any system that has GNU compiler collection built in.

- Constraint

  The group will utilize windows or Linux based operating systems.

  Justification:

  Both hardware and software system development aspects can be done on Linux or Windows architectures.

  Fit Criterion:

  Most users/customers in stock traders use analytic tools that are compatible with Linux and Windows based architectures.

## B2. Naming Conventions and Terminology

**B2a. Field programmable gate arrays (FPGA):**

An integrated circuit designed to be configured by hardware designer.

**B2b. High Frequency Trading (HFT):**

A type of algorithmic trading characterized by high speeds, high turnover rates, and high order-to-trad ratios that leverages high-frequency financial data and electronic trading tools.

**B2c. Stochastic events:**

An event or system that is unpredictable individually by collectively implies a trend due to random variables.

**B2d. GNU:**

A compiler in linux, windows and unix base operating systems for programming kernel based programming languages such as C++.

## B3. Relevant Facts, Assumptions and Dependencies

**B3a. Unable to Access Real Time Data on Physical layer**

- Due to inability for our internal system to access data from stock market in the physical layer, the assumption is that part of our historical data will serve as our real time data stream.

**B3b. Structure of Data Source**

- Data source will be in CSV (comma separated values) format for parsers to the designated units (FPGA and Software Based Systems).
- Data source for User Interface will be handled by the SQL lite API.

**B3c. Types of Possible Algorithms to Implement**

- Genetic Algorithm
- Mean Reversion Algorithm
- Ornstein-Uhlenbeck Process

**B3d. Programming Paradigm**

- For heavy computations, the group will implement a more functional programming paradigm.
- For Software development in the User Interface Blocks, applications will be developed in a more object based programming paradigm.
- Programs and codes developed will be heavily documented for reusability and code update.

**B3e. External Interface Requirements**

- User Interface. ***See section C***
- Hardware Interface. ***See section C***
- Software Interface. ***See section C***
- Communication Interface**. See section C**

**B3f. Hardware Configuration Dependencies**

- Implementations of algorithm for FPGA will be on the Xilinx Virtex-7 board.
- Implementations of algorithm on software based system will be on a 64 bit Windows 7 or higher platform.
- Xilinx Corelib tools will be used extensively to generate pre-compiled codes for mathematical function units embedded in respective algorithm.
- Network server application SQL lite will be installed on a Windows 7 or higher platform.

**B3g. Software Dependencies**

- Raw data translation to FPGA design will be handled by an intermediate parser.
- Raw data translation to User Interface will be handled by an intermediate parser.
- Communication between yahoo data streams and components will be handled by the Yahoo Finance API.
- Repository of Project Development will be handled by a private repository on https://github.com.
- Analysis plugins will be handled by system a generic programming language.

# C. Functional Requirements

C1. The system will have a **User Interface** for **FPGA** unit to select each stocks forecasted lows and high to be displayed.

C2. The system will have a **User Interface** for the **Software Based computer** unit to select each stocks forecasted lows and high to be displayed.

C3. The system will have **data source** for the **FPGA** unit from external web scraping tool to acquire stock data. (From F1 to F.2a)

C4. The system will have internal **data source formatter** for input into the **FPGA** unit.

C5. The system will have internal **data source formatter** for input from **FPGA** unit into the **FPGA designated server**. F.3a to F.4a

C6. The system will have external data output formatter from **FPGA** designated server into the **User Interface**. F.3a to F.4a

C7. The system will have internal **data source formatter** for input into the **Software Based unit**.

C8. The system will have internal **data source formatter** for input from Software Based unit into **Software Based** unit designated server.

C9. The system will have external data formatter from **Software Based** unit designated server into the **User Interface**.

C10. **User** should be able to control the variety of stock output data to be display in User Interface for **Software Based unit**.

C11. **User** should be able to control the variety of stock output data to display in User Interface for software FPGA unit.

C12**. Software based unit output** will be displayed in Software based unit User Interface.

C13. FPGA output will be displayed in **FPGA based user interface**.

C14. **FPGA** must predict the following daily highs and lows of the provided stocks in section **A2F**.

C15. **Software based Architecture** must predict the following daily highs and lows of the provided stocks in section **A2F**.

C16. In order to predict the following daily highs and lows, both **FPGA and Software based systems** will implement the **designated algorithm chosen** to predict each stock's future low and high prices.

# D. Non-Functional Requirements

D1. The system shall have a User interface Display to show stock prediction report via graphic user interface for FPGA unit.

D2. The system shall have a User interface Display to show stock prediction report via graphic user interface for Software based system.

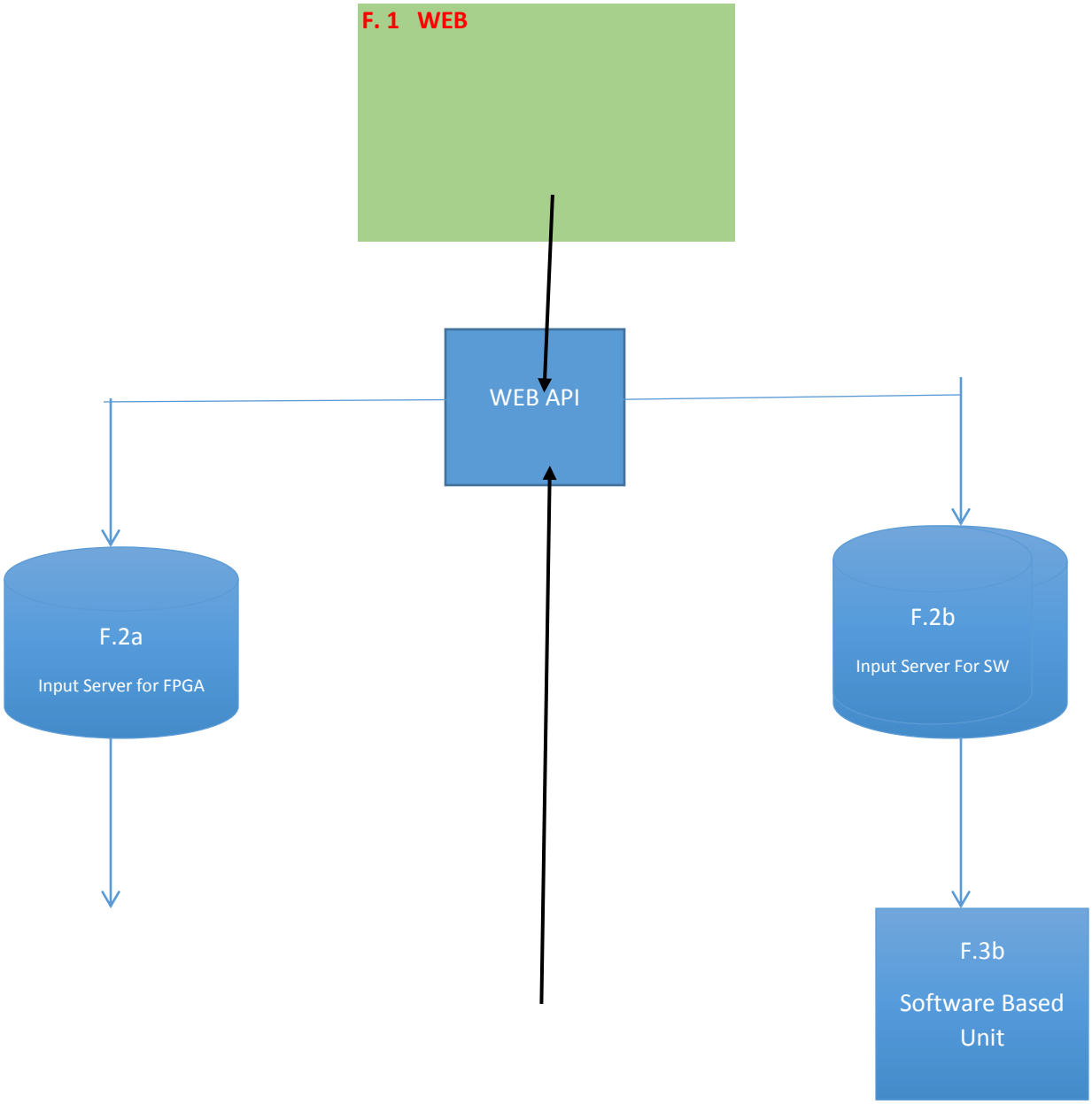
D3. The system must be able to display performance of both hardware and software implementations of algorithm in terms of speed and latency for both FPGA unit and Software based unit.

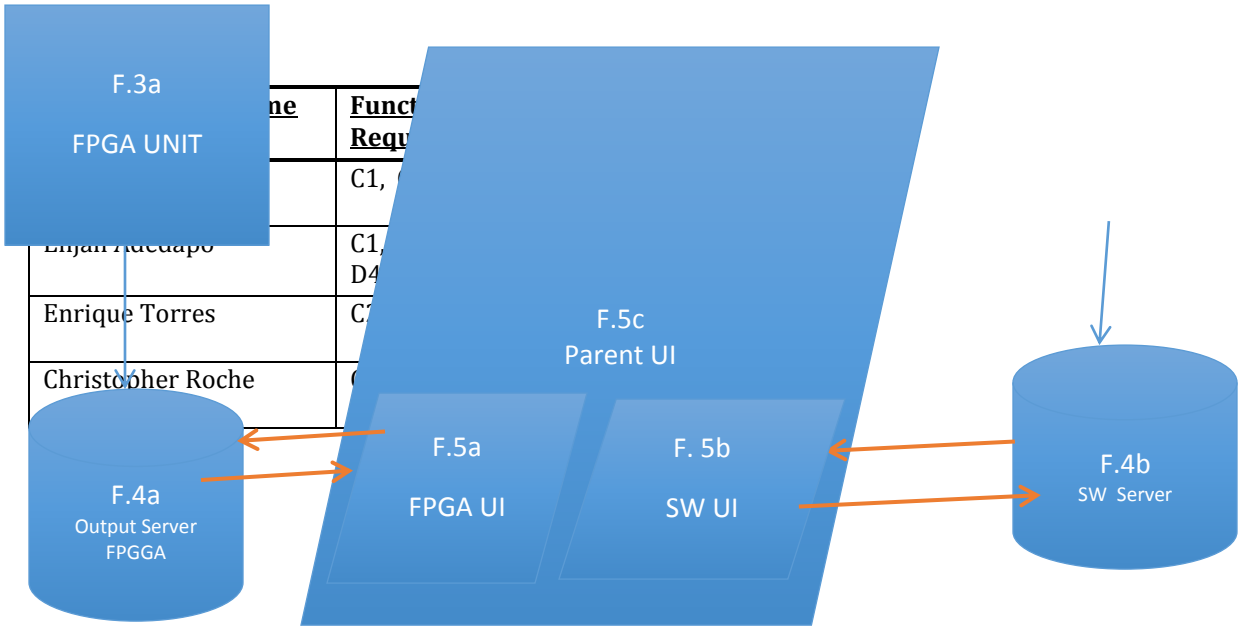
D4. The entire system shall be small enough to fit provided enclosure in the stock trading ground (pit).

D5. FPGA unit must run 99.9% of the time within a typical 24 hour period.

D6. Software based architecture must be running 99.9% of the time within a typical 24 hour period.

# E. Overview of Design






# F. DIVISION OF LABOR

# F. DOCUMENT HISTORY LOG

| Arthurs | Section Modified | Date | Document Version Number |
|---|---|---|---|
| **Clement Cole** | **A – F** | **October 05, 2016** | **0.1** |
| **Clement Cole, Christopher Roach, Elijah Adedapo, Enrique Torres** | **A - F** | **October 06, 2016** | **1.0** |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |