Software Design Document

Rocket View 3000 v0.1.0

June 11, 2017

Contents

Part I
Front-end architecture and design

1 User interface

The user interface is modularized by an Angular.js framework

1.1 Web-app layout

Layout is configured using CSS, and angular material directives as described in index.html

1.2 User interaction and controls

User interaction is limited to a few buttons in the navbar section of the app.

1.3 Real-time data visualization elements

The design of the user interface includes 5 separate elements:

1.3.1 Live streaming video of the rocket launch

This element is a video player that displays live video of the rocket launch.

1.3.2 An 'earth frame view' element

This element is a map view of the rocket position in relation to the surface of the earth.

1.3.3 An 'attitude' element

This element is a view of the rocket showing the rocket orientation.

1.3.4 A telemetry data element

This element is a visualization of numerical telemetry data including graphs and charts.

1.3.5 An 'at a glance' element

This element is a comprehensive numerical view of the rocket telemetry data.

1.4 Visual design considerations

The visual design of the user interface facilitates a comprehensive and visually stimulating rocket launch viewing experience that caters to the individual's viewing preferences. The design allows for the expansion of each visual element so that it is the center of the web page in order for the user to be able to clearly view each element. The user is able to switch which element is highlighted at the click of a button so that he/she is able concentrate on the desired element. The design also allows the user to view a data intensive display that encompasses all of the telemetry data being recorded from the rocket during the launch to track the important numerical data important in a rocket launch. As the user changes the visual layout through the navigation controls, the animations are smooth and easy to follow.

2 Web Server Module

2.1 APRS Middleware

The APRS middleware will be responsible for processing the raw APRS data into a client compatible format.

2.2 APRS Receiver

The APRS receiver will be responsible for catching the raw APRS data sent by the APRS source.

2.3 Client Transceiver

The client transceiver will be responsible for receiving/sending data from/to all connected clients.

2.4 Telemetry Middleware

The telemetry middleware will be responsible for processing the raw telemetry data into a client compatible format.

2.5 Telemetry Receiver

The telemetry receiver will be responsible for catching the raw telemetry data sent by the telemetry source.

2.6 Video Player

The web server video player is rendered using the HTML5 defined video tag.

2.7 Video Rendering

The web server renders video through an open source javascript library. This library uses Media Source Extensions (MSE), a W3C specification to provide browser support for Chromium and Firefox as well as the HTML5 video tag.

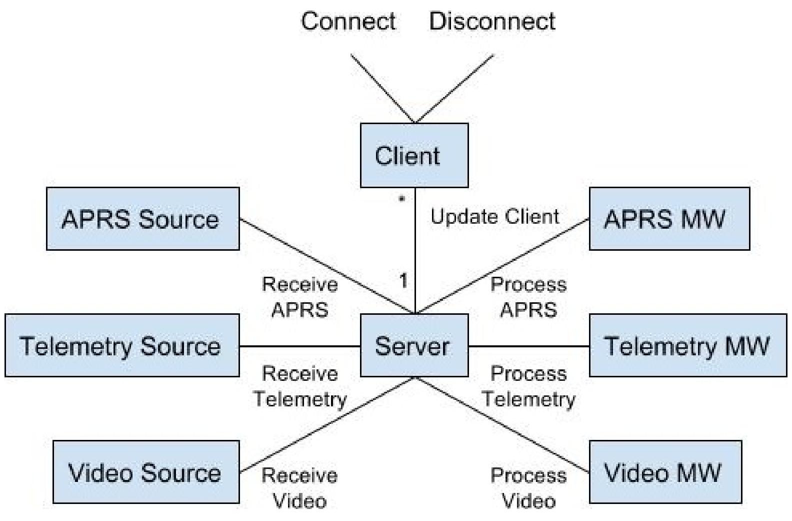

Figure 1: Web Server

3 Earth Frame View (EFV) Module

3.1 EFV display

The Earth Frame display, takes in coordinates, map files, processed trajectory data, telemetry data, and recovery crew GPS data. It then outputs browser compatible code which renders a 3D map with overlay graphical representations of this data in an Earth Frame View.

3.1.1 3D map rendering

The data must be in a form that the map rendering module can easily display.

Map is rendered from local map files using a JavaScript library (TBD) which injects the map into an HTML div element.

3.1.2 Trajectory path overlay

The pre-calculated trajectory is provided in a configuration file by the user. This trajectory object is pulled into the EFV module and rendered as an overlay on the map.

3.1.3 Real-time vehicle trajectory location overlay

This module takes in the coordinates of the vehicle. The purpose for of the GPS data is to indicate the rockets current location on its trajectory.

The current location of the vehicle is represented as a dot in the EFV display.

3.1.4 Vehicle path overlay

As the vehicle moves on its trajectory path a red trail will be rendered to show the trajectory path the vehicle has traveled on. The logs the incoming coordinates of the vehicle. This is to insure that the rendering can display the path the vehicle has taken.

3.1.5 Recovery Crew Location Indicator Overlay

The system takes in the GPS coordinates of the recovery crews. Earth frame view only needs to keep track of the current location of the recovery teams.

3.1.6 Map angle views

3.2 Assets

3.2.1 Map files

Map files consist of collected terrain maps which are indexed by cartographic coordinates.

3.3 Configuration files

Configurable JSON file that stores information about calculated trajectory, network ports, and location.

4 Rocket Attitude Module

4.1 Rocket Attitude Display Module

The Rocket Attitude display takes in IMU data and a 3D model. It renders this 3D model, and then uses IMU data from the rocket to alter the orientation of the model to match that of the live rocket.

4.1.1 Model Storage

There is an offline storage which stores the 3D model, as well as the material that is applied to the model.

4.2 IMU Data Calculations

The system performs any necessary computations on the IMU data to make it usable for the display module.

4.2.1 Smoothing Calculations

The default rotating of the rocket model sometimes causes choppy animation. As a result, the rotating goes through a smoothing process in order to ensure a smooth display.

4.3 Display Output

The module uses a rendered 3D model, along with the processed and smoothed data, to produce a live simulation of the real rocket's attitude during its flight.

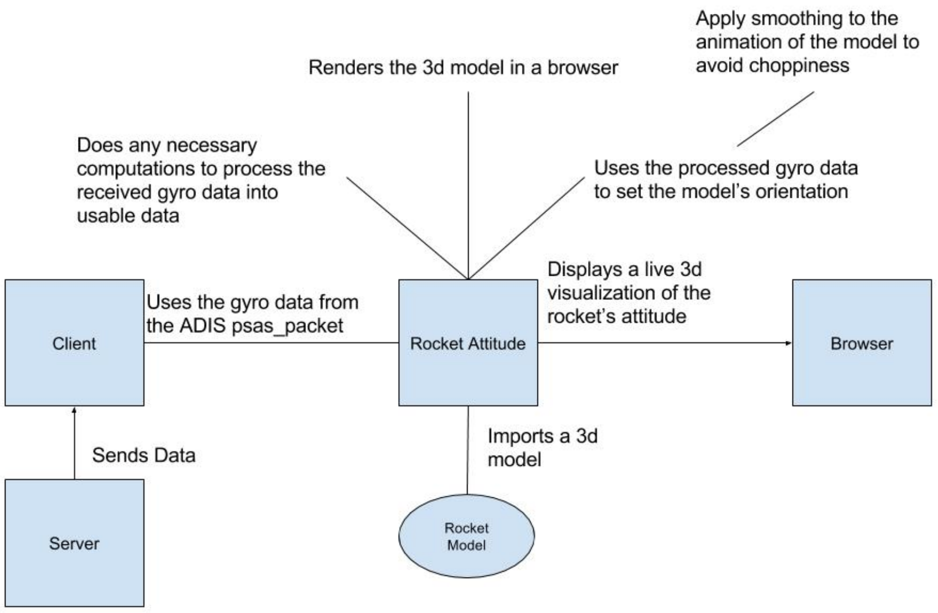
Apply smoothing to the animation of the model to avoid choppiness
Renders the 3d model in a browser
Does any necessary computations to process the received gyro data into usable data
Uses the processed gyro data to set the model's orientation
Displays a live 3d visualization of the rocket's attitude
Client
Uses the gyro data from the ADIS psas_packet
Rocket Attitude
Browser
Sends Data
Imports a 3d model
Server
Rocket Model

Figure 2: Rocket Attitude

5 Telemetry Data Module

5.1 Main View

Receives data through a socket opened to the telemetry server module, and pipes it to relevant directives that control visualization of the data on the main view of the page.

5.2 At A Glance View

Receives data through a socket opened to the telemetry server module to the At A Glance directive, which in turn uses Angular's 'ng-repeat' directive to cycle through the data and serve it to At A Glance view of the page.

Part II
Back-end architecture and design

6 Telemetry data module

6.1 Client Service

Receives data from the telemetry server middleware module.

6.2 Data Processing Functions

Receives data from client service and splits it into relevant, useful categories such as gyroscopic data, acceleration, etc.

Performs functions on data as necessary to make it useful for outside modules such as the EFV module.

6.3 Error Checking

Performs functions on data as necessary to indicate strength and integrity of data and signals.

6.4 Display

Displays pertinent telemetry data to relevant views, in respective HTML ids and tags.

Displays graphical representations of numeric data in real time. Organizes information in an easy to read format.

Responsible for "at-a-glance" view.

7 Back-End Server

The Back-End server instantiates different versions of itself based on command line arguments. Each instance of a Back-End server is be responsible for listening for a specific type of data, APRS data, telemetry data and video data.

7.1 APRS Module

The APRS module catches raw APRS packets sent by the Recovery Crews and sending on to the Front-End server.

7.2 Telemetry Module

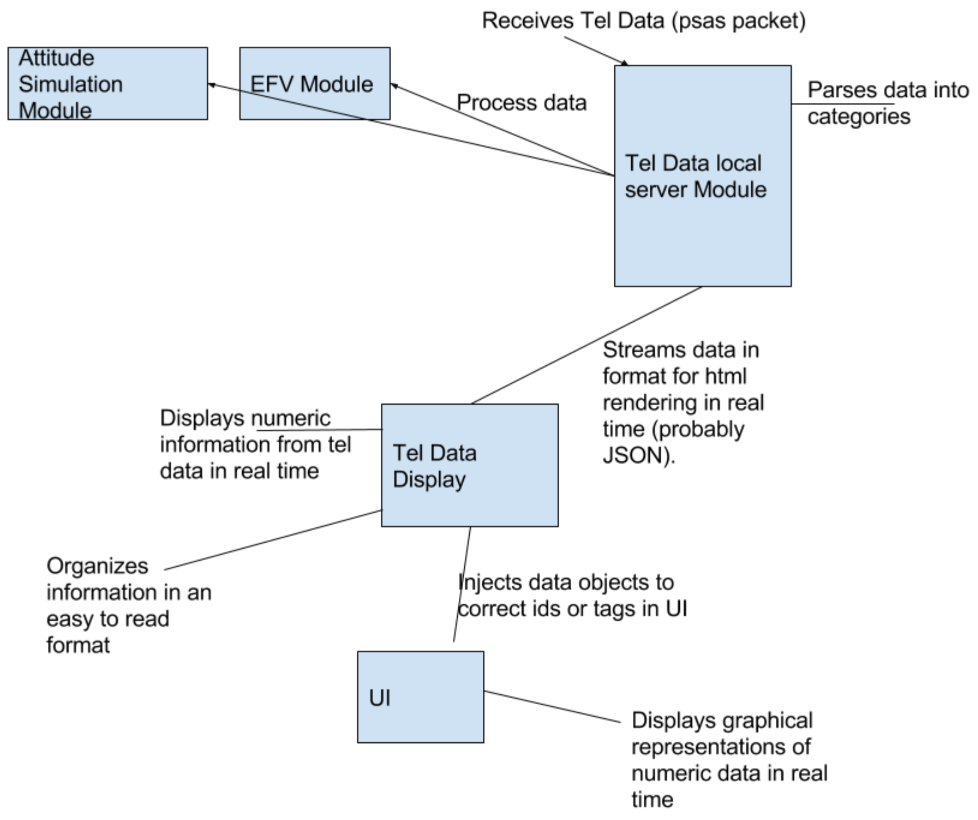
Receives Tel Data (psas packet)
Attitude Simulation Module
EFV Module
Process data
Tel Data local server Module
Parses data into categories
Streams data in format for html rendering in real time (probably JSON).
Displays numeric information from tel data in real time
Tel Data Display
Organizes information in an easy to read format
Injects data objects to correct ids or tags in UI
UI
Displays graphical representations of numeric data in real time

Figure 3: Telemetry Module Architecture Plan

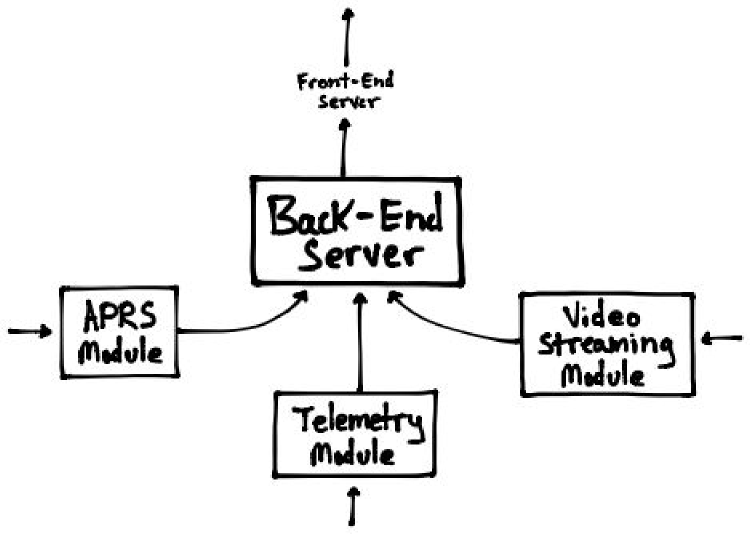

Figure 4: Back-End Server

The Telemetry module catches raw telemetry data from the rocket and sending to the Front-End server.

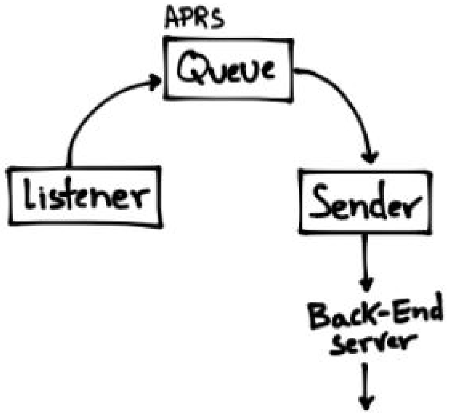

Figure 5: APRS module

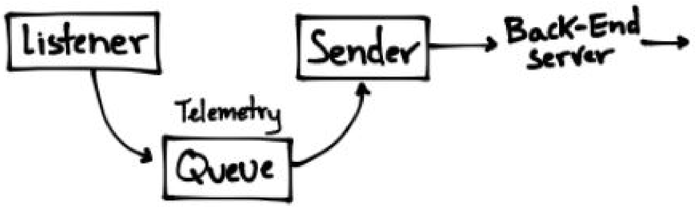

Figure 6: Telemetry Module

7.3 Video Streaming Module

The video module catches the live video streams and sending them to the Front-End server.

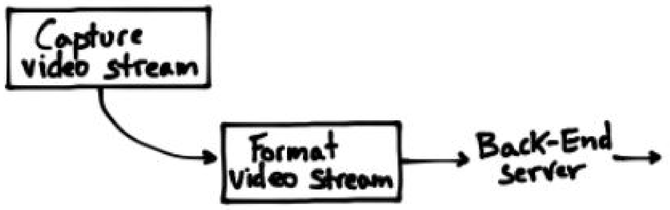

Figure 7: Telemetry Module

8 APRS Middleware Module

8.1 ARPS Client Submodule

The APRS client module will receive APRS data incoming from the local app server module, and pass it to APRS $\rightarrow$ GPS Conversion submodule.

8.2 APRS $\rightarrow$ GPS Conversion Submodule

The APRS $\rightarrow$ GPS conversion submodule will convert incoming APRS data to a more- friendly GPS data format and then pass the GPS data to the GPS server submodule.

8.3 GPS Server Submodule

The GPS server submodule will serve GPS data received from the conversion submodule to the local app server module.