Software Design Document

Rocket View 3000 v0.1.0

May 15, 2017

Copyright (c) 2017 Amanda Murphy, Jeff Patterson, Patrick Overton, Matt Tighe, Yun Cong Chen, Seth Amundsen, Paolo Villnueva, Michael Ohl, Connor Picken

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Contents

I Front-end architecture and design **3**

1 User interface **3**
1.1 Web-app layout . . . 3
1.2 User interaction and controls . . . 3
1.3 Style elements . . . 3
1.4 Real-time data visualization elements . . . 3
1.4.1 L . . . 3
1.4.2 A . . . 3
1.4.3 A . . . 3
1.4.4 A . . . 3
1.4.5 A . . . 3
1.5 Visual design considerations . . . 3

2 Web Server Module **4**
2.1 APRS Middleware . . . 4
2.2 APRS Receiver . . . 4
2.3 Client Transceiver . . . 4
2.4 Telemetry Middleware . . . 4
2.5 Telemetry Receiver . . . 4
2.6 Video Player . . . 4
2.7 Video Rendering . . . 4

3 Earth Frame View (EFV) Module **5**
3.1 EFV display . . . 5
3.1.1 3D map rendering . . . 5
3.1.2 Trajectory path overlay . . . 5
3.1.3 Real-time vehicle trajectory location overlay . . . 5
3.1.4 Vehicle path overlay . . . 5
3.1.5 Recovery Crew Location Indicator Overlay . . . 5
3.1.6 Map angle views . . . 5
3.2 Assets . . . 5
3.2.1 Map files . . . 5
3.3 Configuration files . . . 5

4 Rocket Attitude Module **5**
4.1 Rocket Attitude Display Module . . . 5
4.1.1 Model Storage . . . 6
4.2 IMU Data Calculations . . . 6
4.2.1 Smoothing Calculations . . . 6
4.3 Display Output . . . 6

II Back-end architecture and design **6**

5 Telemetry data module **7**
5.1 Client Service . . . 7
5.2 Data Processing Functions . . . 7
5.3 Error Checking . . . 7
5.4 Display . . . 7

6 Video Processing Module **7**
6.1 Incoming Video Streams . . . 7
6.2 PSAS Server . . . 7
6.3 Video Codec Middleware . . . 7
6.3.1 Codec . . . 7
6.4 Supported Browser . . . 7
6.4.1 Real-Time UI Widgets . . . 8

7 Back-End Server **8**
7.1 APRS Module . . . 8
7.2 Telemetry Module . . . 10
7.3 Video Streaming Module . . . 11

8 APRS Middleware Module **11**
8.1 ARPS Client Submodule . . . 11
8.2 APRS -> GPS Conversion Submodule . . . 11
8.3 GPS Server Submodule . . . 11

Part I
Front-end architecture and design

1 User interface

< Description of angular framework and such >

1.1 Web-app layout

1.2 User interaction and controls

1.3 Style elements

1.4 Real-time data visualization elements

The design of the user interface includes 5 separate elements:

1.4.1 L

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

1.4.2 A

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

1.4.3 A

n 'attitude' element This element is a view of the rocket showing the rocket orientation.

1.4.4 A

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

1.4.5 A

n 'at a glance' element This element is a comprehensive numerical view of the rocket telemetry data.

1.5 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.

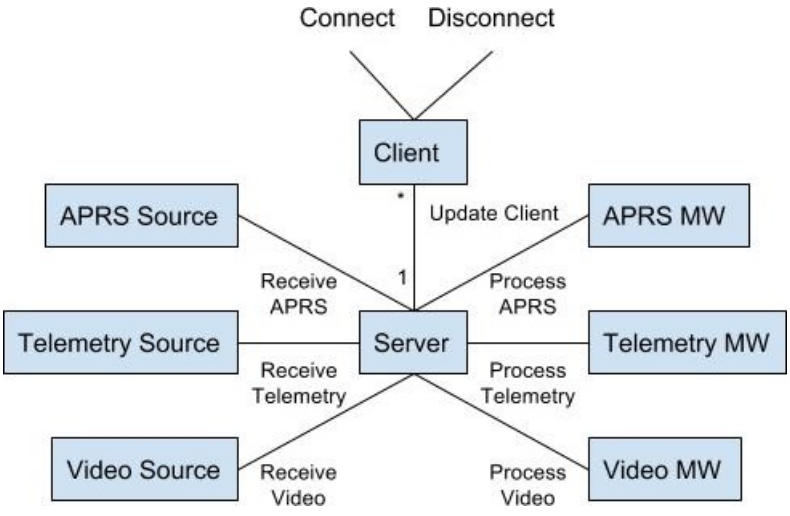

Figure 1: Web Server

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.

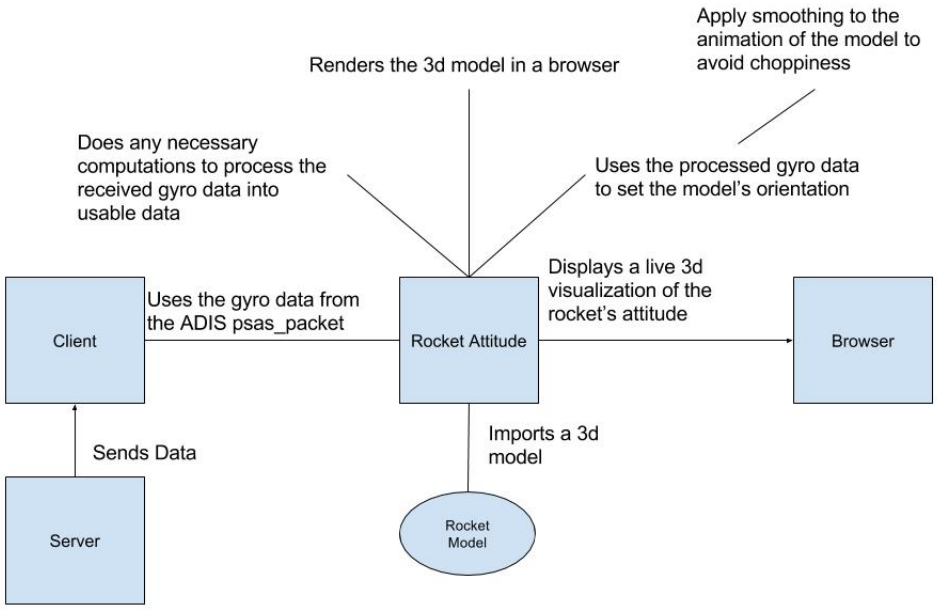

Figure 2: Rocket Attitude

Part II
Back-end architecture and design

5 Telemetry data module

5.1 Client Service

Receives data from the telemetry server middleware module.

5.2 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.

5.3 Error Checking

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

5.4 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.

6 Video Processing Module

6.1 Incoming Video Streams

The system will support or be configurable to support video from multiple input steams from multiple sources

6.2 PSAS Server

The PSAS Server is responsible for serving launch data and tools that are employed during a launch

6.3 Video Codec Middleware

The Video Codec Middleware pipes the uncompressed video stream into the video Codec

6.3.1 Codec

The Codec is responsible for translating the video signal from its raw uncompressed transmission state into a compressed video stream

6.4 Supported Browser

Chromium and Firefox are supported through the HTML5 video tag and the W3C specification Media Source Extensions (MSE) via an open source javascript library.

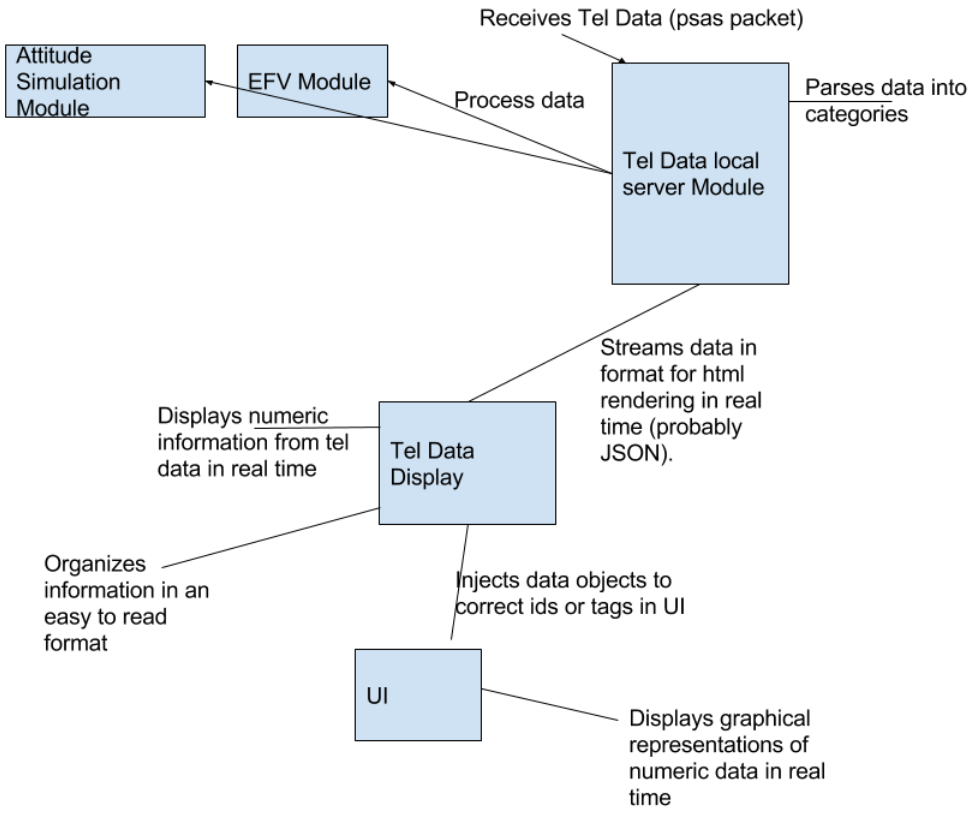

Figure 3: Telemetry Module Architecture Plan

6.4.1 Real-Time UI Widgets

Real-Time UI Widgets provide options that allow the viewer make choose the way in which they view the launch.

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.

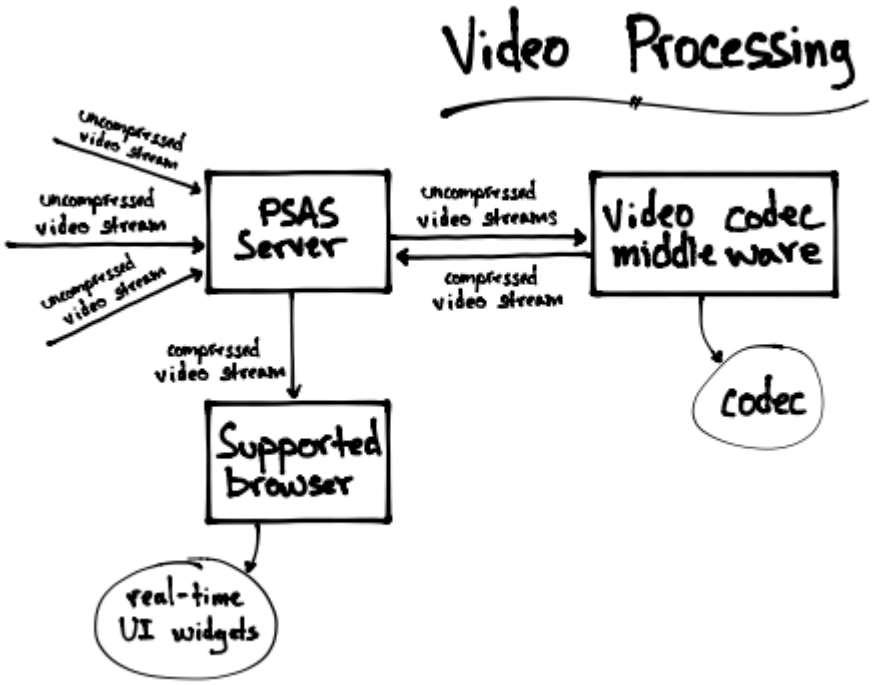

Figure 4: Video Processing

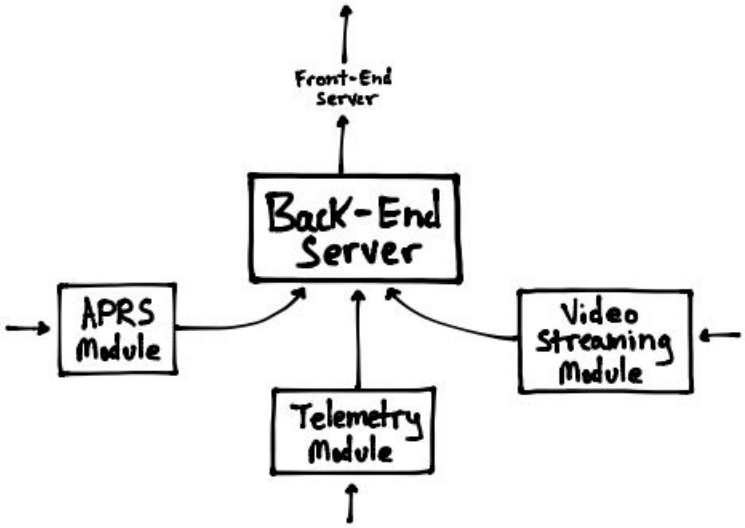

Figure 5: Back-End Server

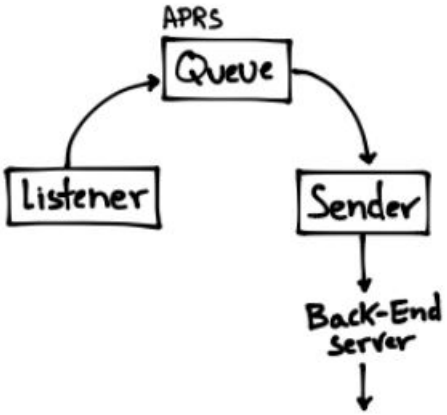

Figure 6: APRS module

7.2 Telemetry Module

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

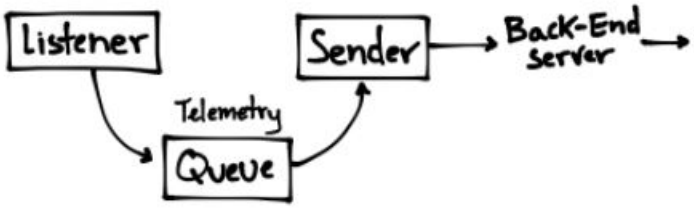

Figure 7: Telemetry Module

7.3 Video Streaming Module

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

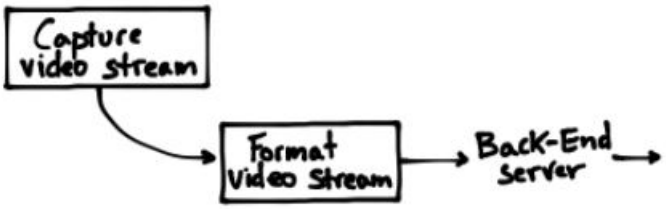

Figure 8: Telemetry Module

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 -> GPS Conversion submodule.

8.2 APRS -> GPS Conversion Submodule

The APRS -> 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.