Software Design Document

Rocket View 3000 v0.1.0

May 10, 2017

Contents

Part I
UI Design

TBD

1 Html 5 and JavaScript

1.1 Templated views

1.2 Wireframe Layouts

1.3 User interaction and controls

1.4 Style elements

1.5 Real-time data visualization elements

1.6 Visual design considerations

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 Middleware

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

2.7 Video Receiver

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

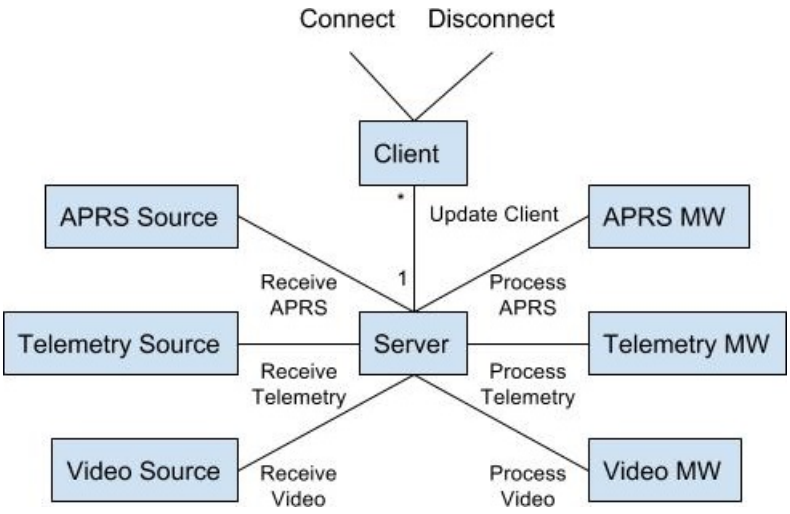

Figure 1: Web Server

3 APRS Middleware Module

3.1 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.

3.2 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.

3.3 GPS Server Submodule

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

4 Earth Frame View (EFV) Module

4.1 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.

4.1.1 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.

4.1.2 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.

4.1.3 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.

4.1.4 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.

4.1.5 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.

4.1.6 Map angle views

4.2 Assets

4.2.1 Map files

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

4.3 Configuration files

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

5 Rocket Attitude Module

5.1 Rocket Attitude Display Module

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

5.1.1 Model Storage

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

5.2 Gyro Data Calculations

The system will need to perform any necessary computations on the received gyro data to make it usable for the display module.

5.2.1 Smoothing Calculations

It is very likely that simply setting the models orientation to the processed data will cause choppiness. As a result, a smoothing process should be applied between the processed data and current rocket orientation to ensure a smooth display.

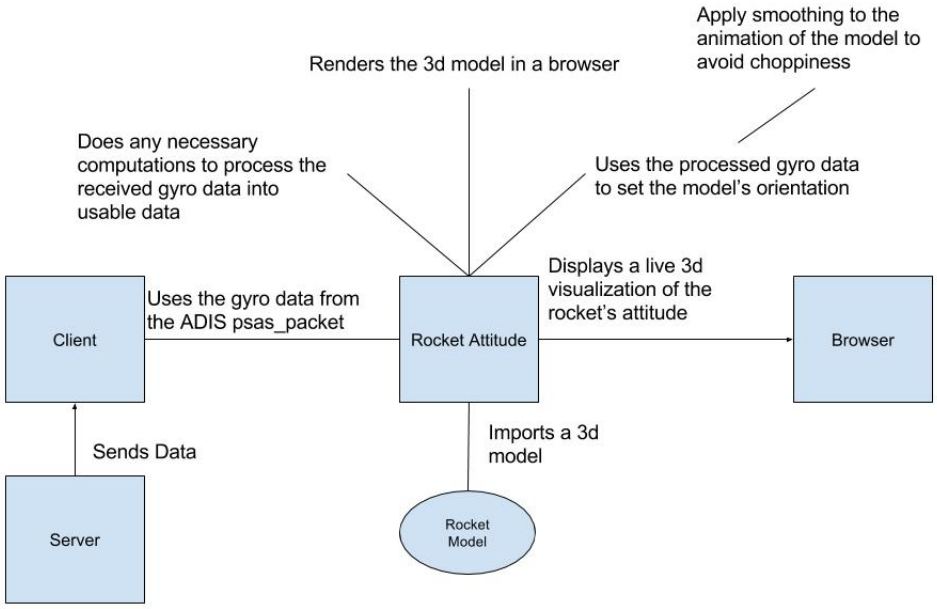

Figure 2: Rocket Attitude

5.3 Display Output

The module will use the rendered 3D model, along with the processed and the smoothed data, to produce a live simulation of the launched rocket's attitude.

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.

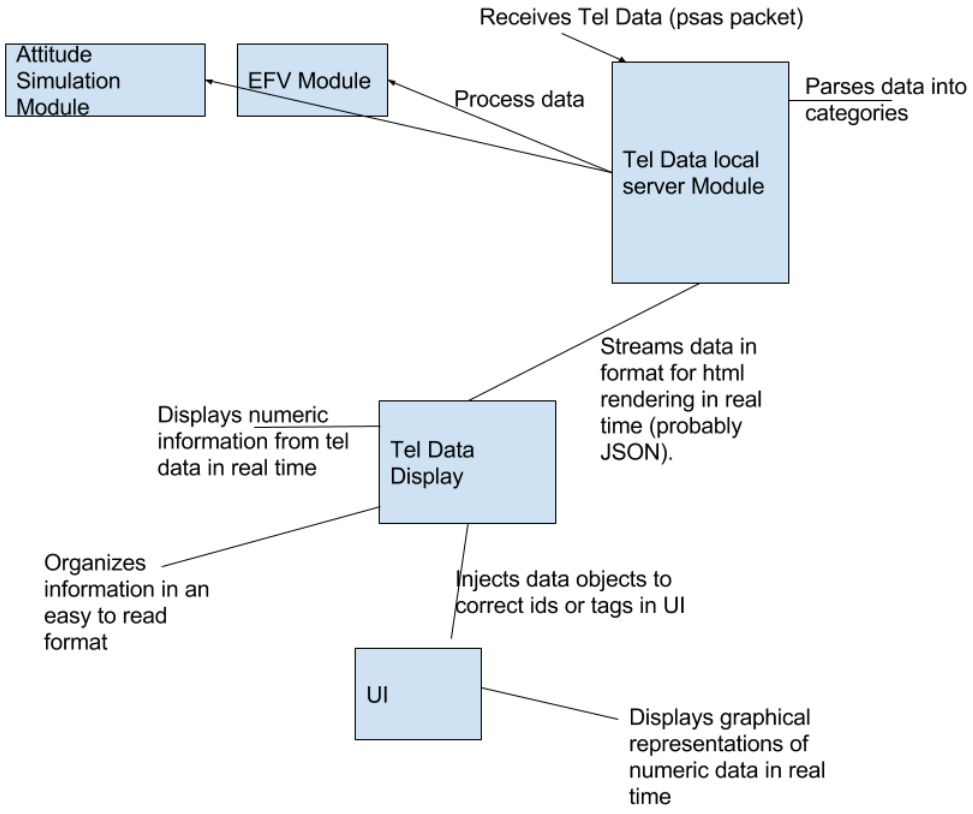

Figure 3: Telemetry Module Architecture Plan

7 Video Processing Module

7.1 Incoming Video Streams

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

7.2 PSAS Server

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

7.3 Video Codec Middleware

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

7.3.1 Codec

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

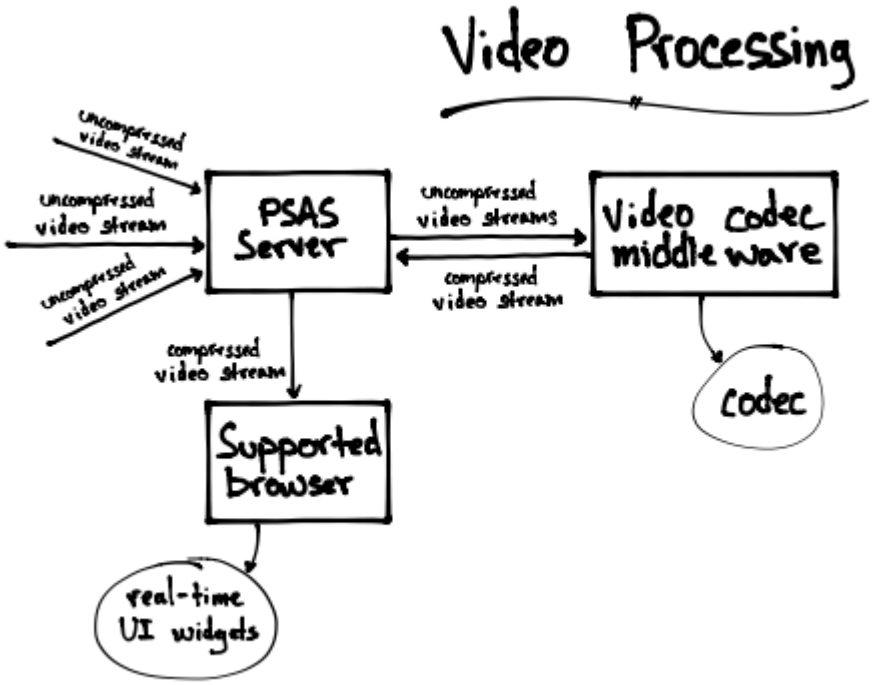

Figure 4: Video Processing

7.4 Supported Browser

A processed video stream is accessible via a web UI

7.4.1 Real-Time UI Widgets

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