**2. Background Information**

The Big Comfy Couch combines different concepts and libraries to achieve its results. These concepts include:

* JavaFX
* Humble Video
* *Akka Actors*
* *UDP/TCP connections*

2.1 JavaFX

JavaFX is a graphics library that allows applications to have a full cross platform user interface. JavaFX API allows the user’s application to manage multiple different interfaces in a single window with their implementation of the stage and scenes. It delivers full modular functionality of forms and interactive windows.

The Big Comfy Couch focuses on three distinct elements in JavaFX to drive its User Interface: Stage, Scene, and Layout. A stage is a window that can render a scene. A scene is a container that can be displayed in a stage. A layout is a tool to segment a scene into structured regions to contain nodes. A Node is an element like text, or a button. The Big Comfy Couch uses three different layouts: BorderPane, VBox, and HBox. An Hbox or Horizontal Box is a layout structure horizontally. Nodes added to an HBox are placed beside each other left to right in the order they are added. A VBox or Vertical Box is the same as an HBox except the nodes are placed vertically, with nodes descending. A BorderPane is a layout that divides the scene into five distinct regions: top, bottom, left, right, center. Each region is logically arranged with the top being above all other regions, and the bottom being below all other regions. Left, right, and center regions are sandwiched between the top and bottom, in their respective order.

2.3 Akka Actors

*Fill in this section*

2.2 Humble Video

Humble Video is a library that is used to Demultiplex, decode, filter, encode and multiplex video and audio in Java. It is built from the FFmpeg project. In the Big Comfy Couch it is used to play the video. Humble first demultiplexes a file to get the different streams. It then looks at each stream until its finds the first video stream. After a video stream has been found Humble Video gets the file ready for decoding by accessing individual packets of the video stream. It takes each packet, decodes them into a picture and comparing the video’s timestamp offset (from the time the video was loaded) to the systems current nano time, produces the correct picture to be rendered.

2.4 UDP/TCP Connnections

*Fill in this section*