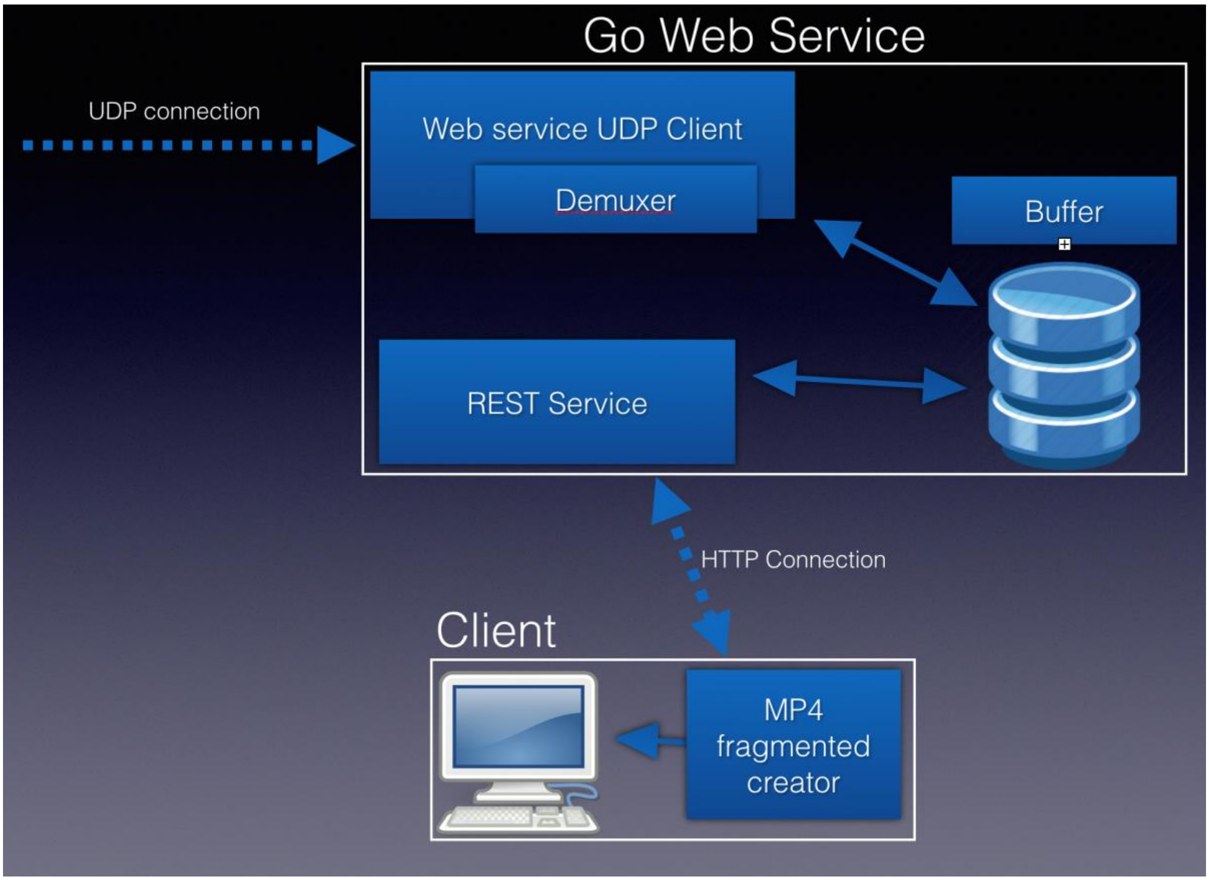
Developer Manual

Overview:



Components:

- Go gateway service:
    - Main thread – listens to UDP socket and transfers received data to 'extractstream' function. There it is demuxed to h.264 frames and stored inside a buffer
    - DispatcherQueueFiller thread – traverses the frames in the main threads buffer and sends them in organized order through a GO buffered channel to the REST service thread.
    - REST service thread – Listens on a http socket for incoming requests. When request is received, sends the next frame (starting with Iframe) to the requestor. If no frame is ready to be sent, sleeps waiting for one on the channel.

- Client code
  - Dispatches HTTP requests to the server to get the frames, stores them in a queue.
  - Process the queue, remixing the frames and generating mp4 fragments which are then pushed into the browser player media source buffer.

Files:

- Mpeg2ts(Go server files):
  - streamer.go - main thread. Handles the http request as well as arranging the output queue.
  - Frame.go - interface for using and creating frames
  - FrameQueue.go – interface for main thread buffer of frames (not output channel buffer)
  - PMTFrame.go – analyzes the PMT frame to identify the PID of the video.
  - StreamInfo.go – constants and definitions for PMTframe.go file
  - Mpeg2TSPacket.go – interface for parsing MPEG2-TS packets
  - MPEG2TSParser.go – collects MPEG2-TS packets to build a single frame
  - Mpeg2TSSource.go – handles UDP connection (as client) and uses the FrameQueue interface to fill the first buffer with frames
- Transmitter (UDP service) – Delivered to us by the supervisor. Not in our scope.
- Client.html – JavaScript Code with html tags. This is the browser interface for the client. Handles the http requests to the go server and remuxes the given frames into mp4 fragments. Pushes the mp4 fragments into the HTML5 video player.