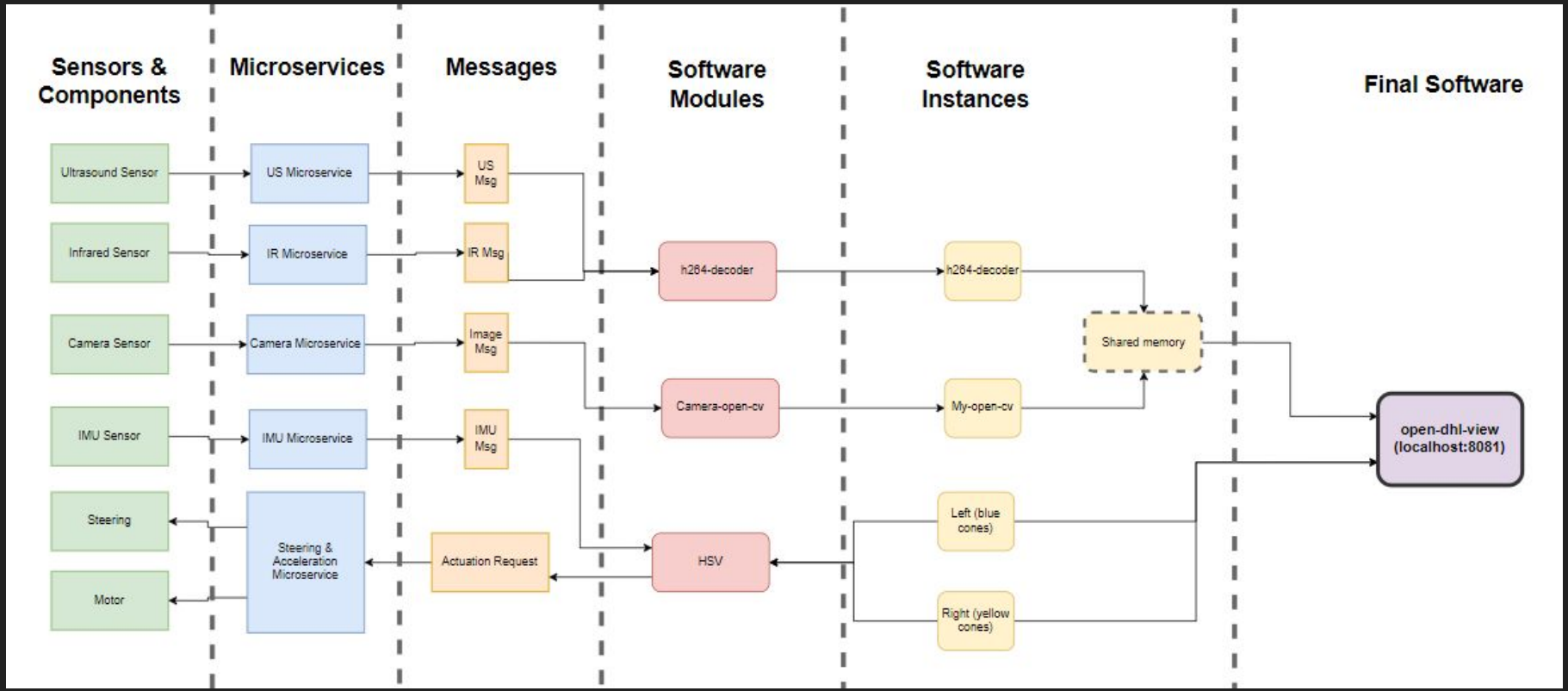
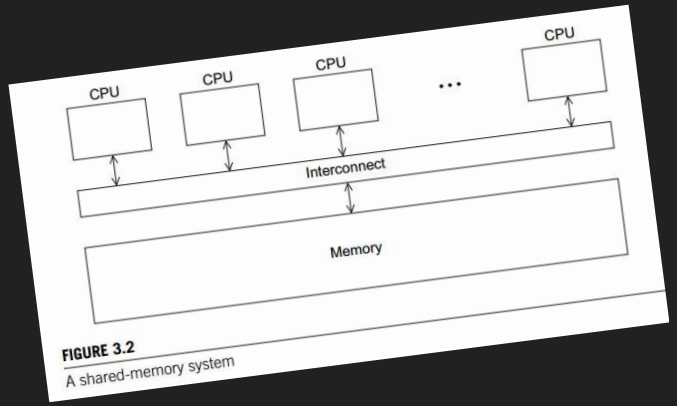
Group 14

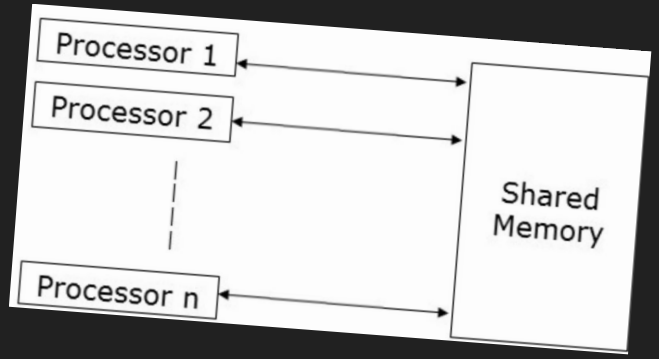
Presentation 1

Microservices Architecture

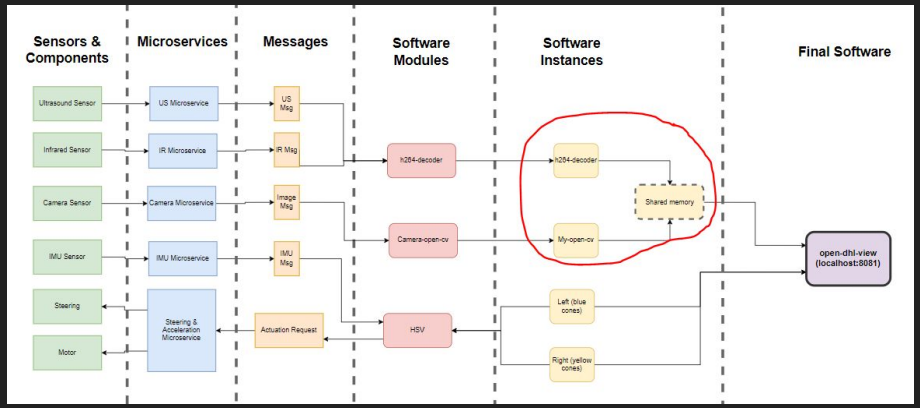


Just like CPUs connect to a memory, processors connect to an external ‘shared memory’. I.e Processes communicate: Inter-process communication

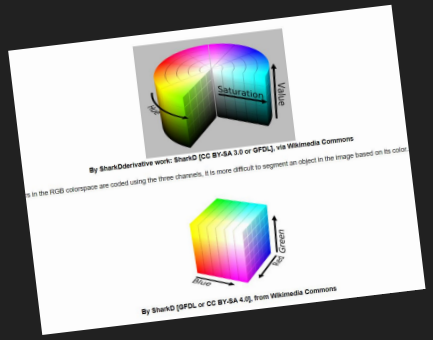


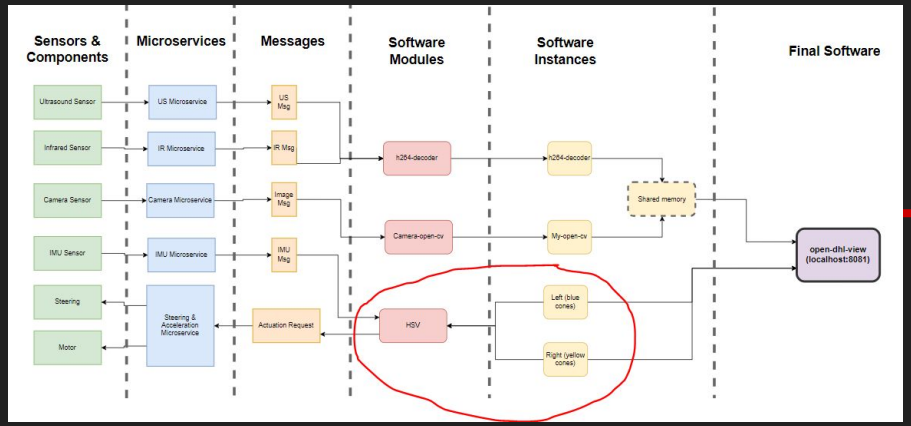


- Attach to a shared memory area created by the H264 microservice

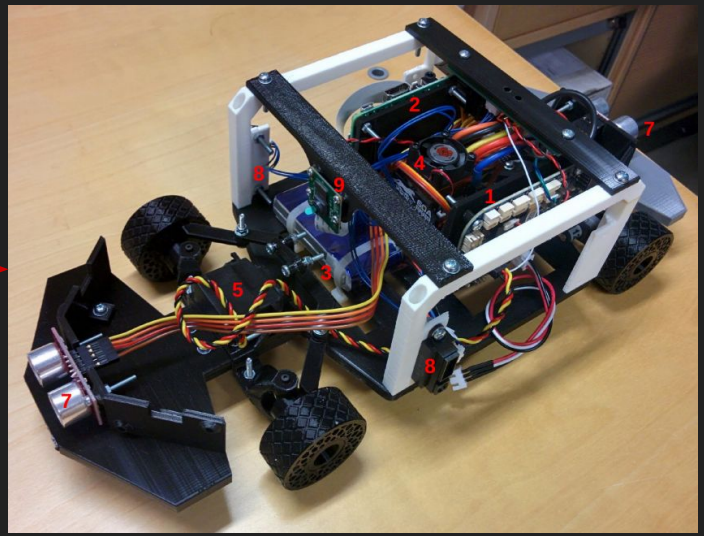


Hsv (hue, saturation, value) microservice

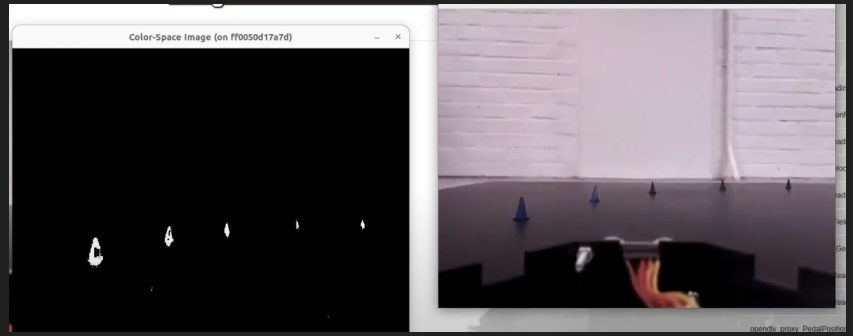


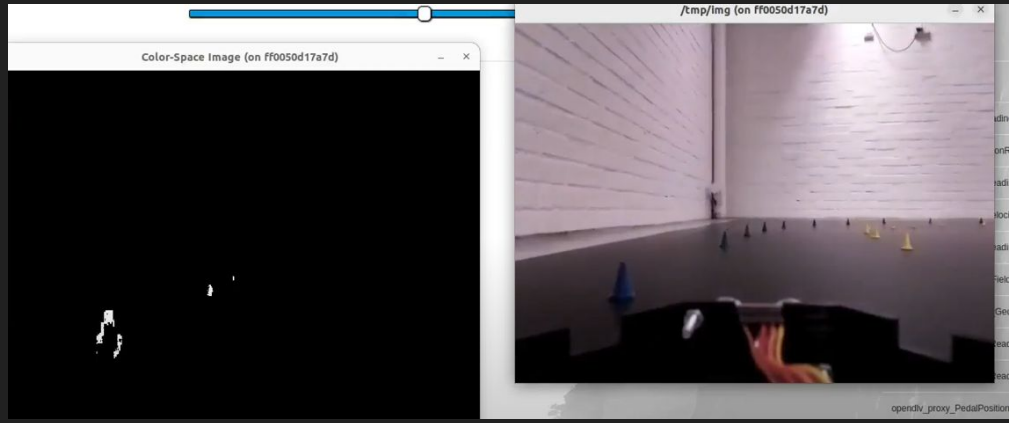


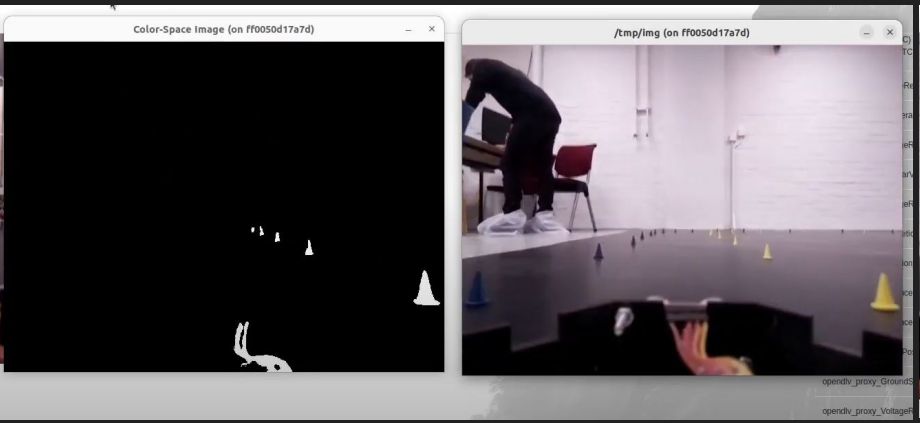
8.

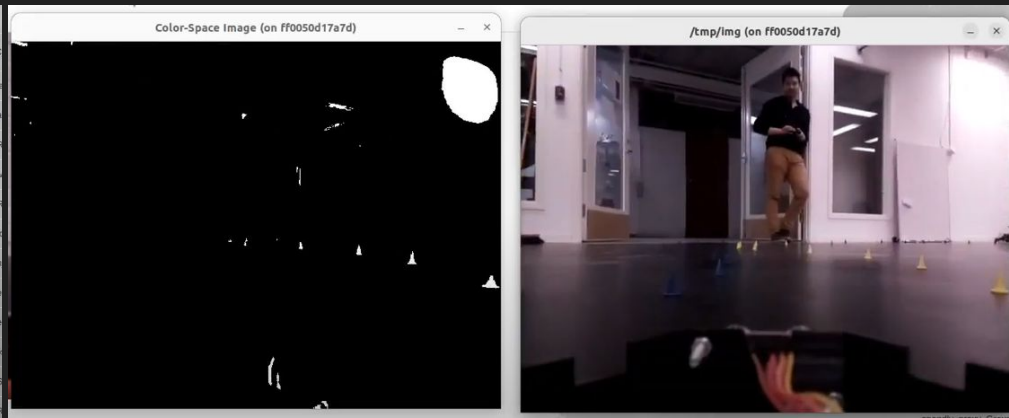


Object detection - hsv microservice

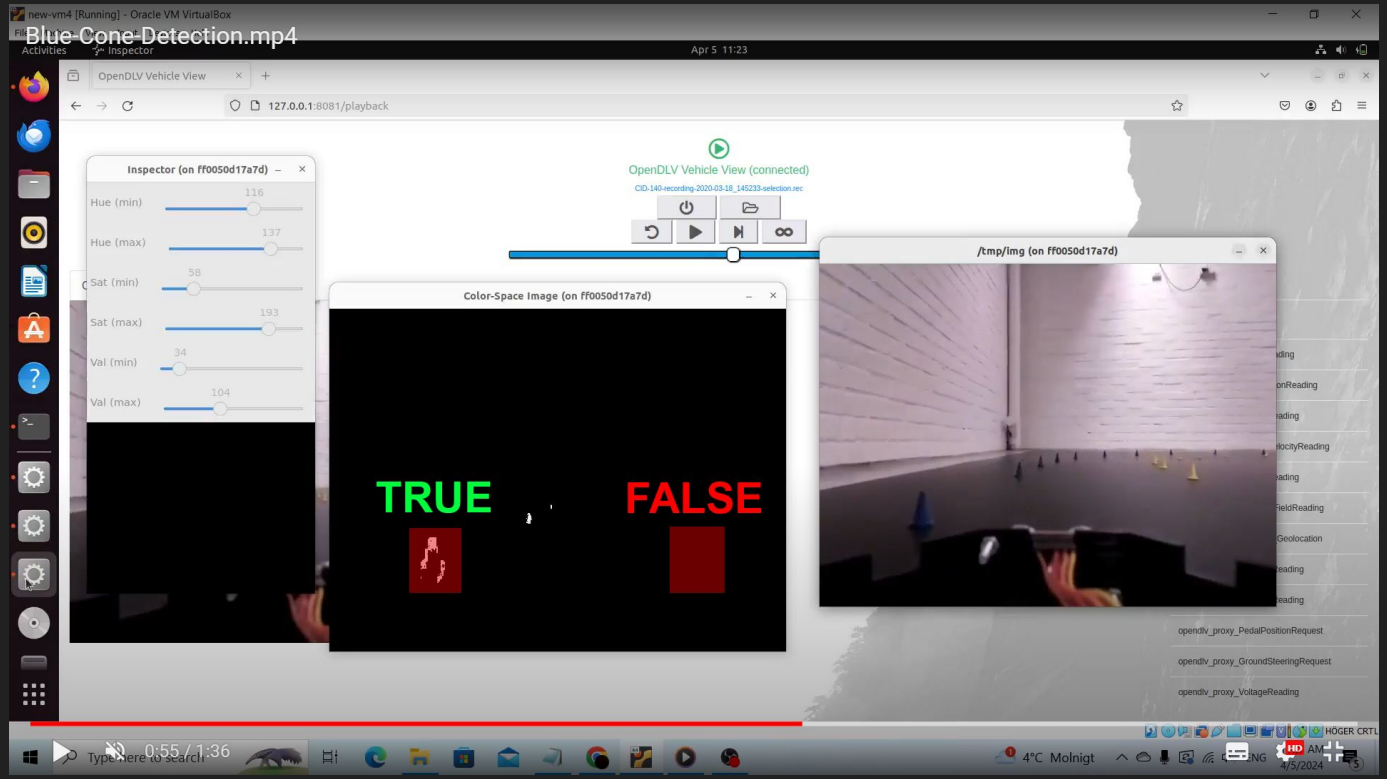






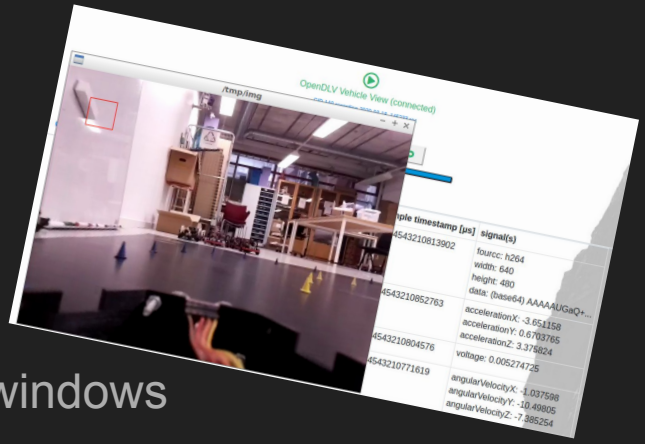


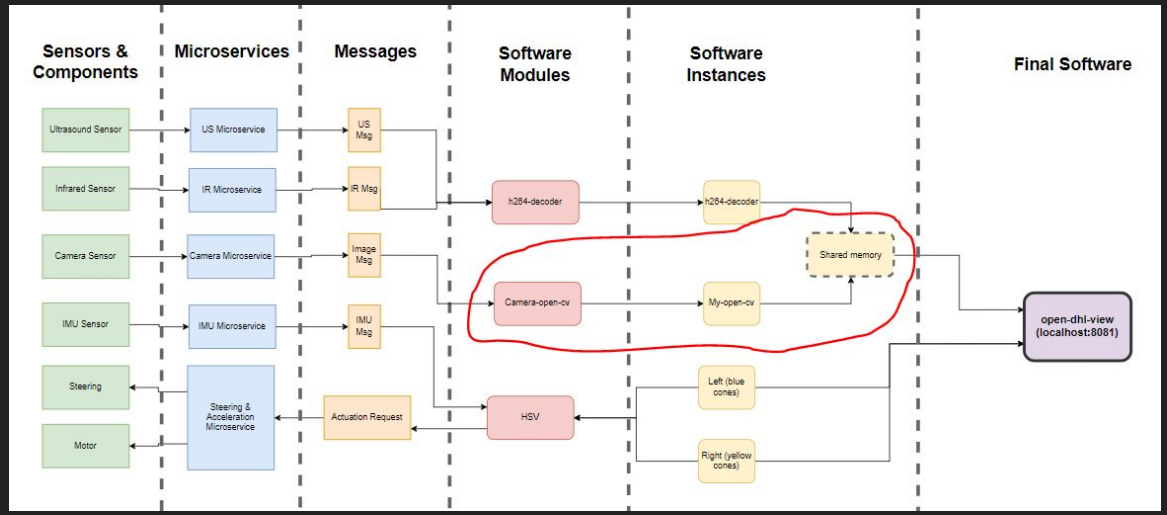
Boxes with predefined static positions on the image:



camera-opencv microservice

- Image processing
- Display UI
- Prints information of interest in the terminal/debug windows





Steering Algorithm

Continuously detect the distance from the centre of the car to the closest cones of each colour. if one cone is closer than the other then we know that we have to turn the car. if there is one cone detected, but another is missing, we know that we must make as sharp a turn as possible. Since there is a lot of background noise, we will take a certain area of the video to detect the cones within.

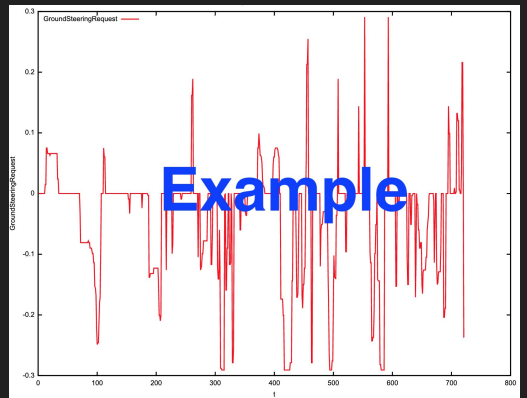
Rationale:

- Hard to detect cones in the back
- Drawing a line for multiple cones would require an advanced math solution beyond our scope
- The second thing → Advanced math solution beyond our scope (Hough Transform)

Testing the algorithm

- Timers to check if blue/yellow cone is continuously detected by the HSV sensors (assume that it's always on a valid road)
- Check difference of steering-wheel angle between the provided program (Christan's) and our program

Correct output:



Note: We will not expect the steering wheel angle to be exactly identical to the provided progam's, since the solutions may differ slightly. Therefore, we will have a const variable "errorMargins" that is the maximal allowed factor to represent the difference between the two programs throughout the entire video

Group 14

End