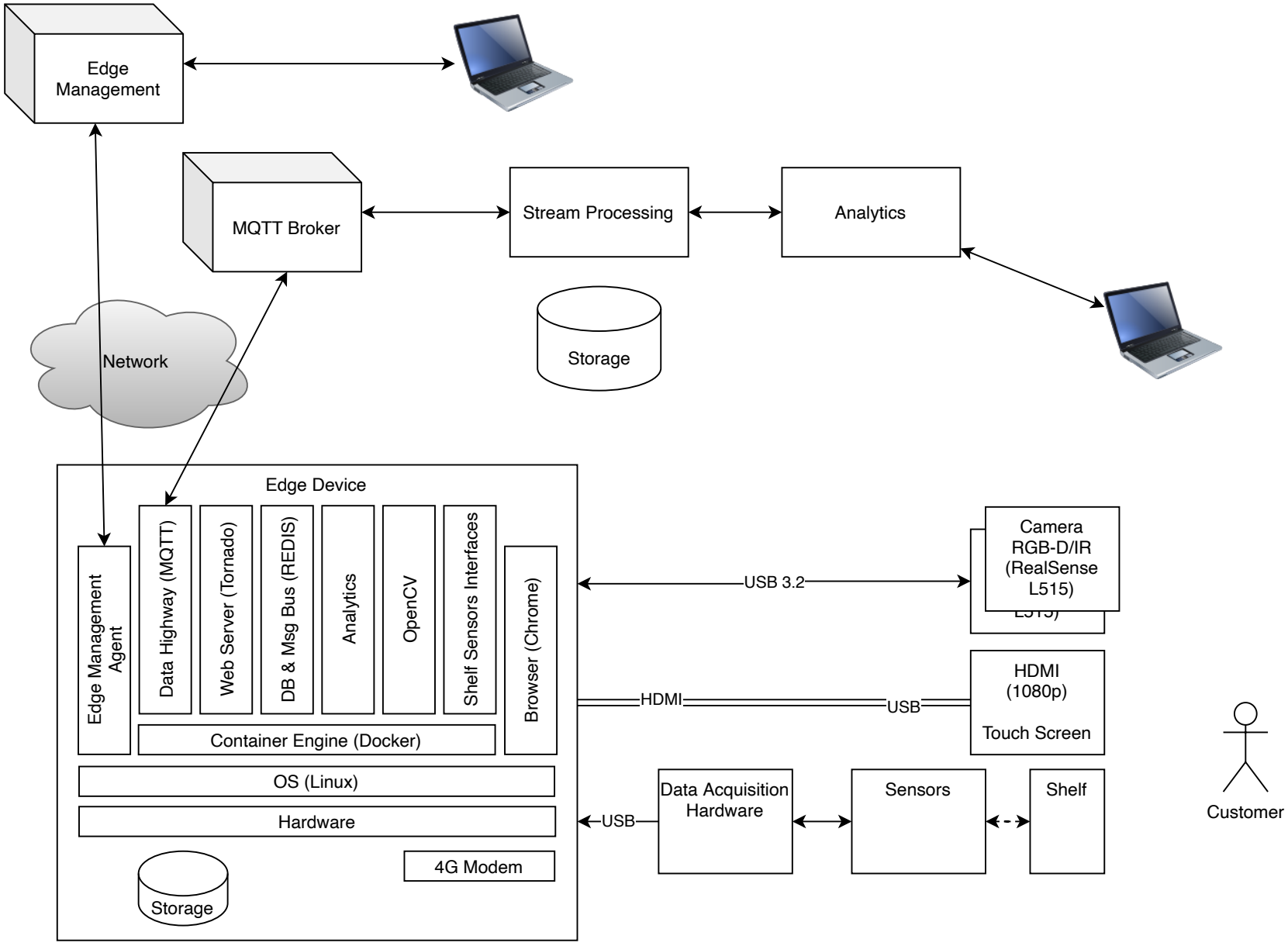
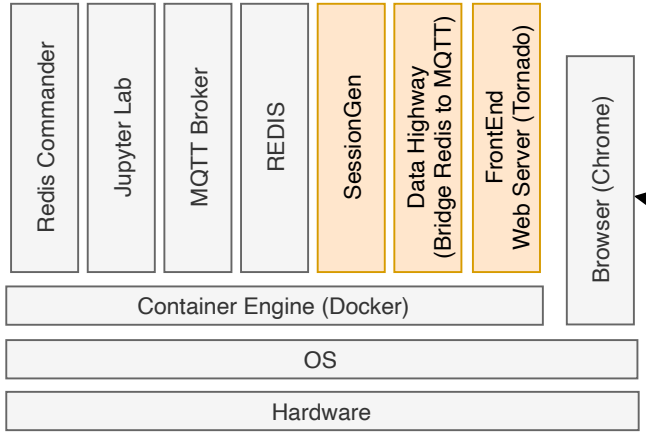
System Overview (expected in field)

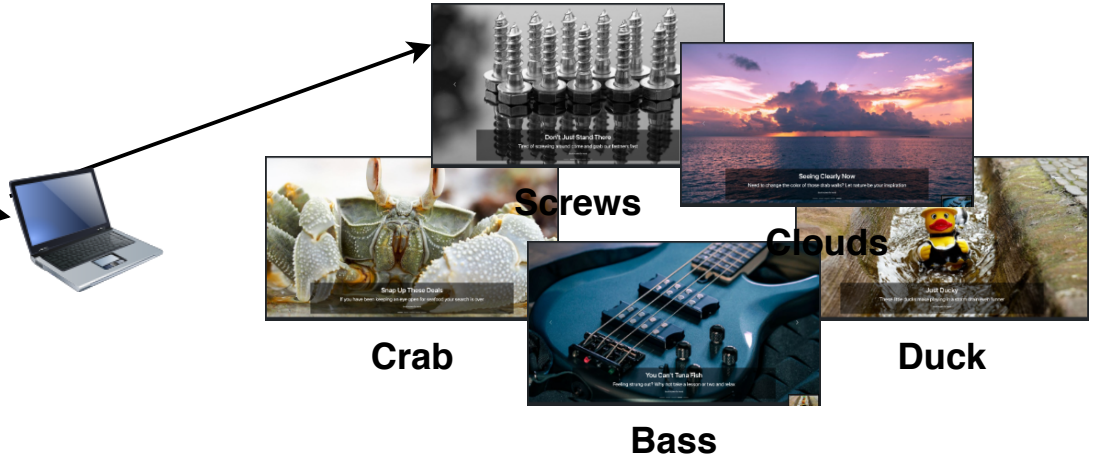


As Built Overview

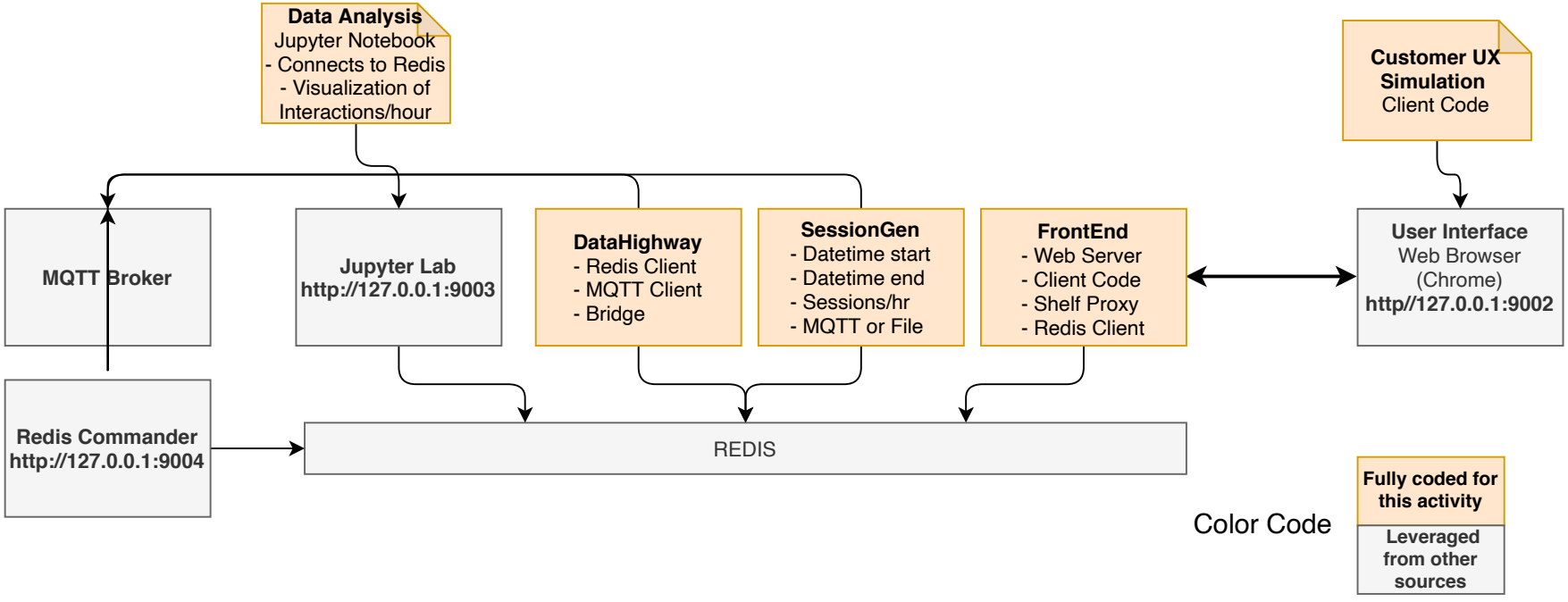
System Components



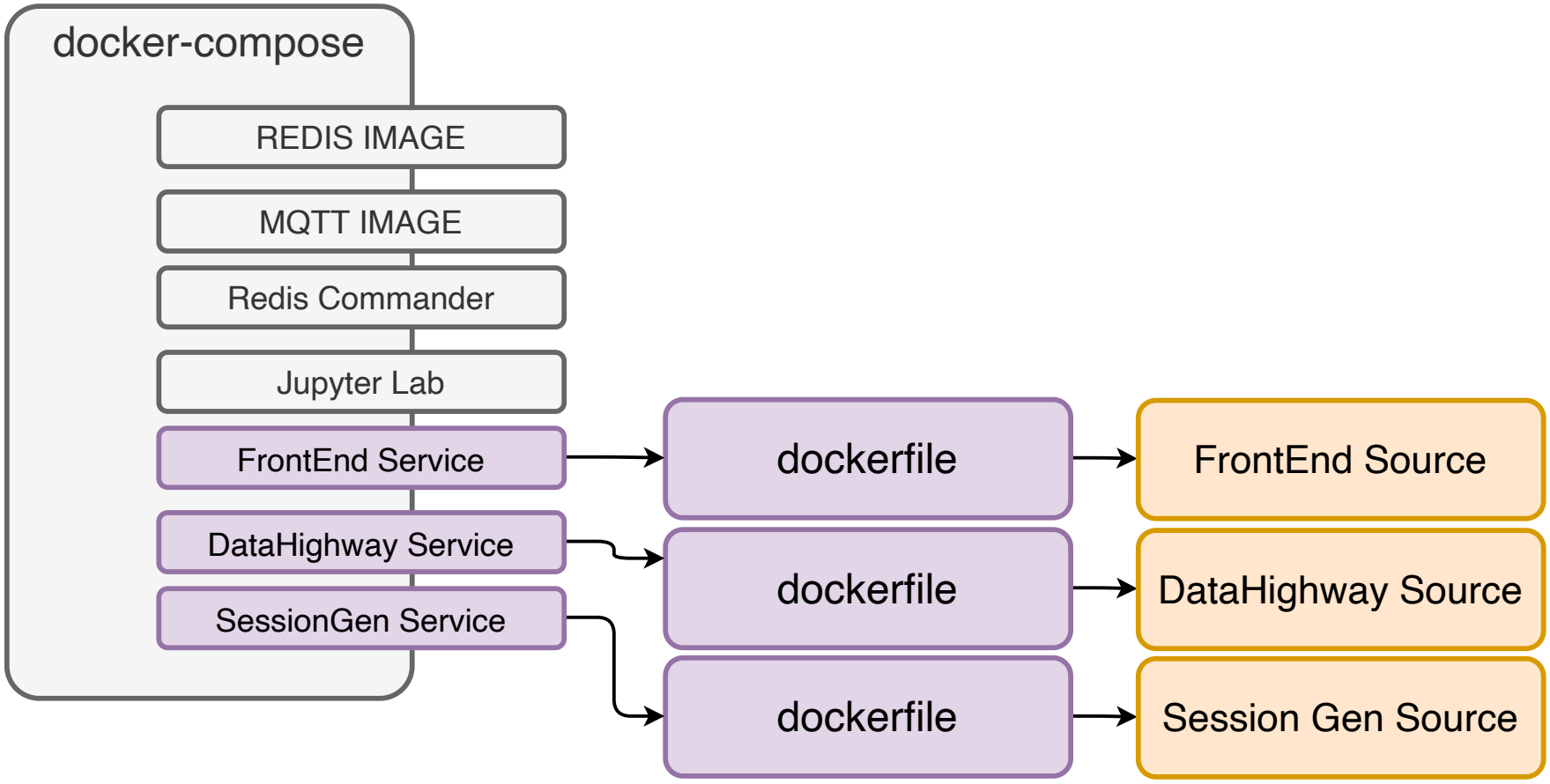
User Interface



Component Interconnection



Build Overview

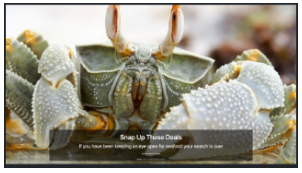


To Build Call
$ docker-compose up

Python Base Image

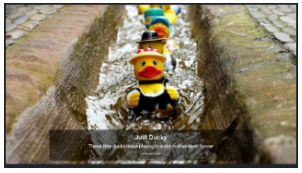
User Interface

5 Products



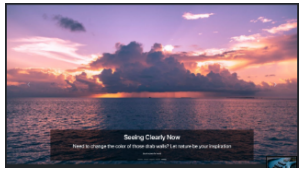
Crab

Screws



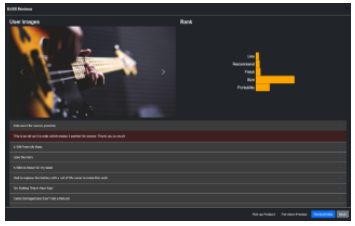
Duck

Bass

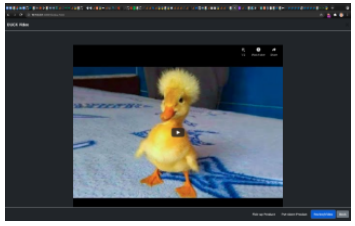


Clouds

From can navigate to each item and select to look at product specific reviews or videos

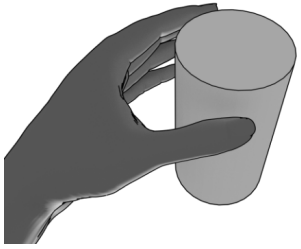


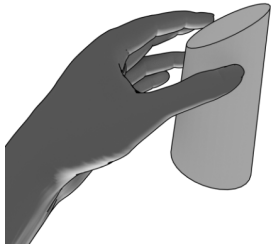
Reviews



Video

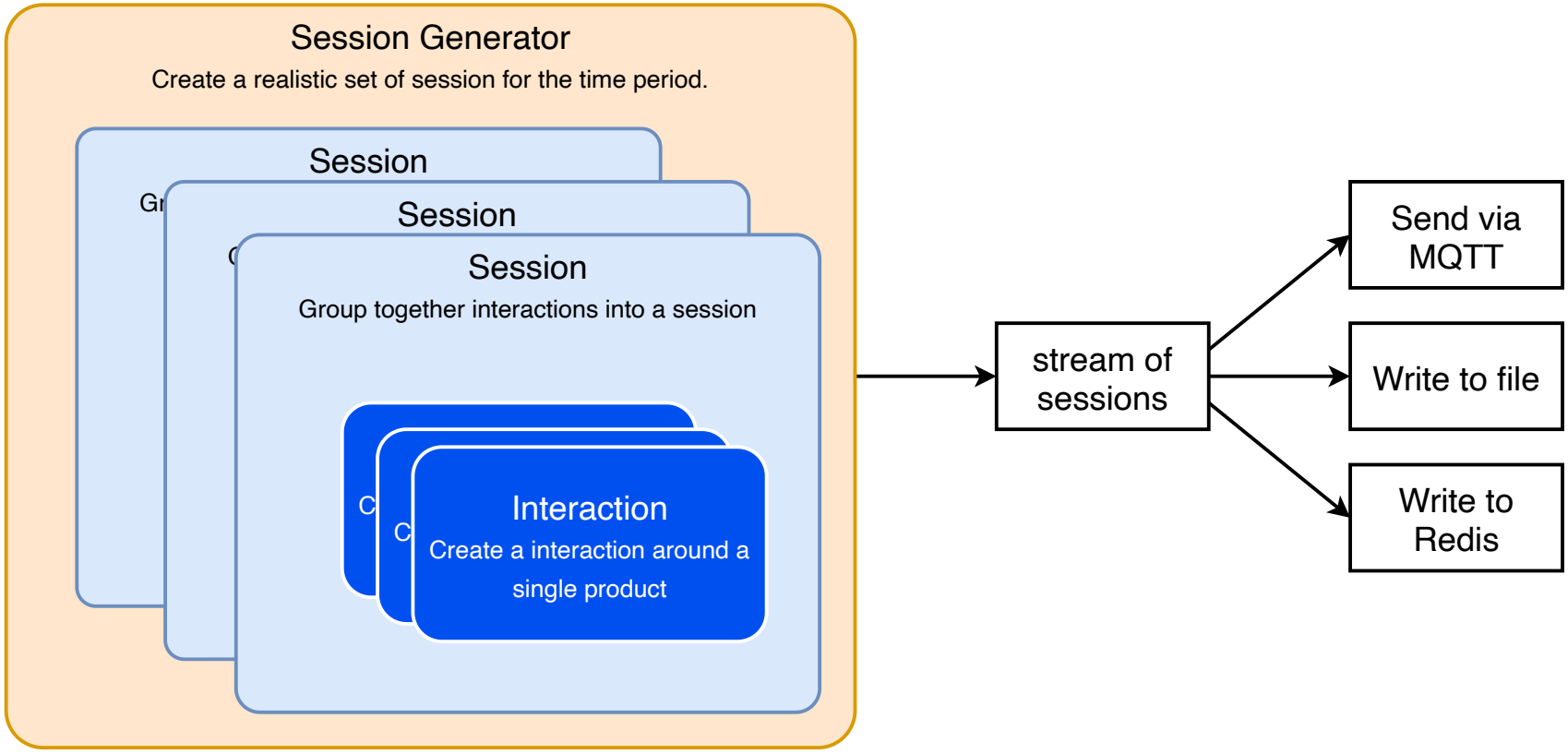
You can simulate picking up or putting down a product by pushing a button



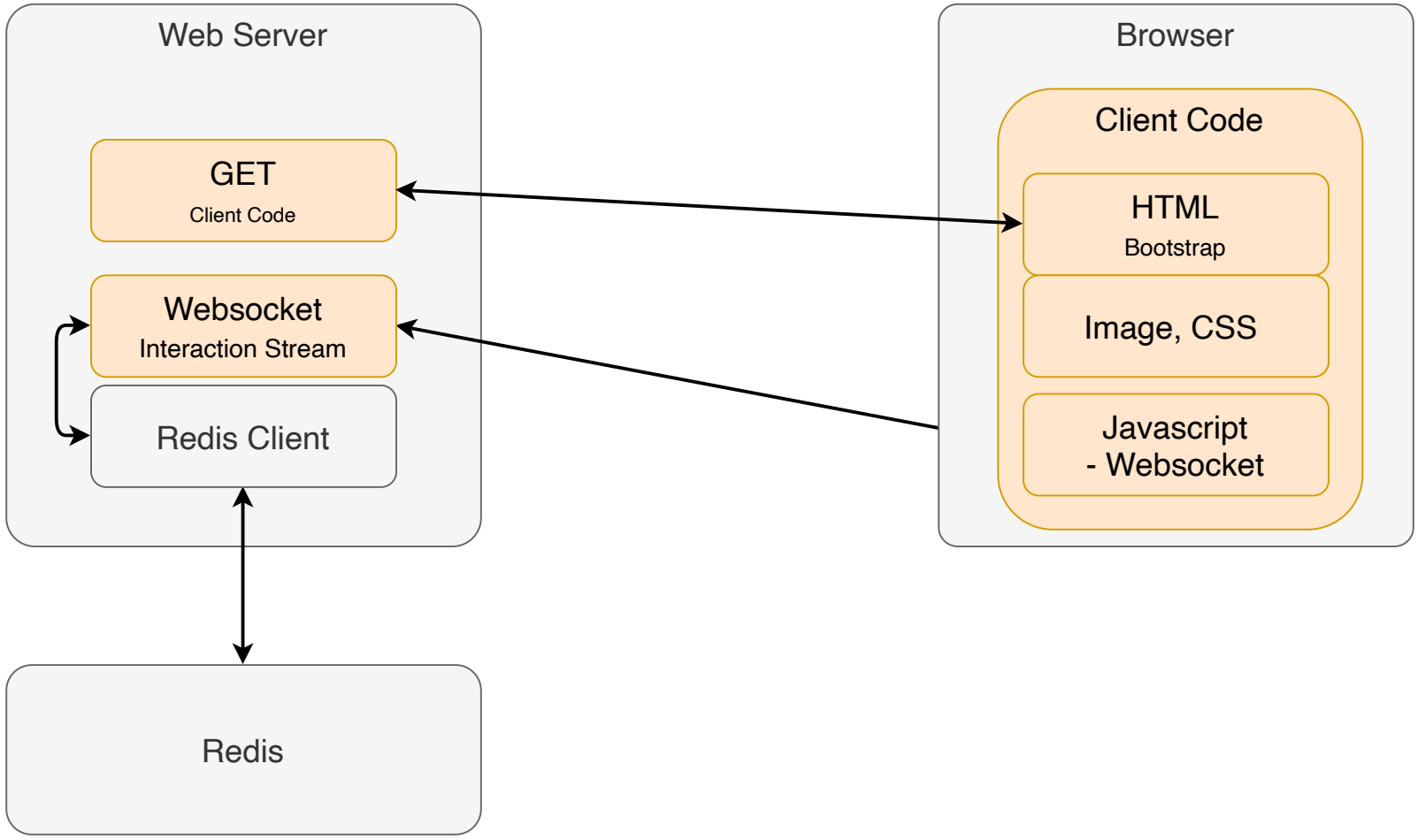


There is motion and sound as well

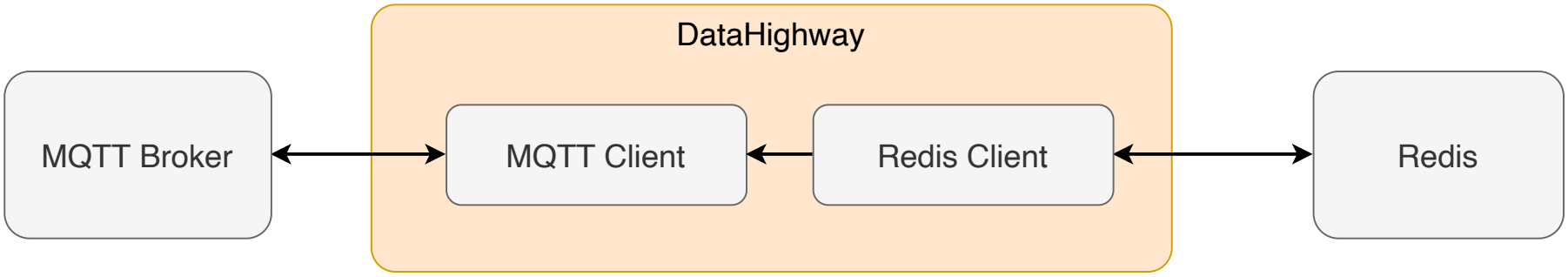
Session Generator Overview



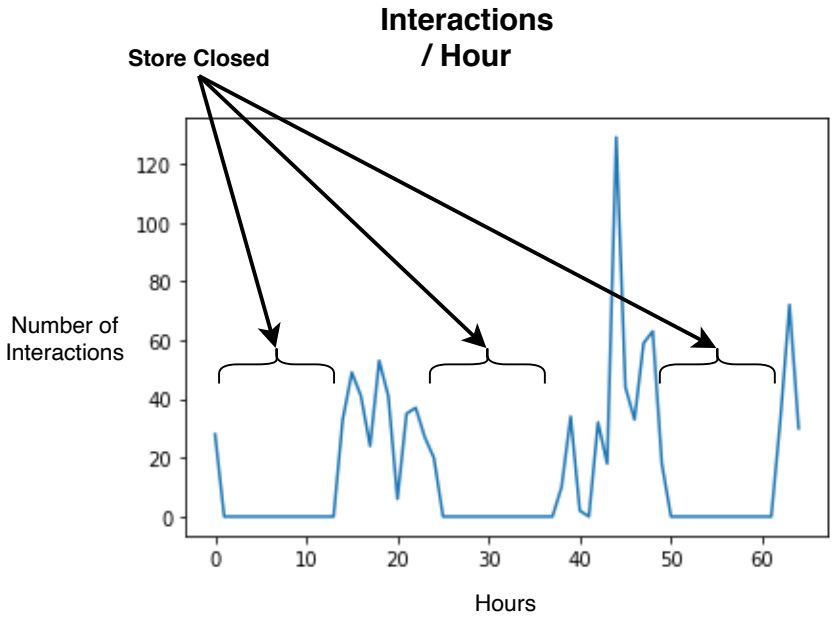
FrontEnd Overview



DataHighway Overview



Data Analytics



Full Notebook can be found in directory ./DataAnalysis or in the Jupyter Lab after the docker-compose up

Plot_Hr_Rate

June 14, 2021

[2]:
```
import redis
import os
import matplotlib.pyplot as plt

# Open Redis
REDIS_IP_ADDRESS = os.getenv("REDIS_IP_ADDRESS", "localhost")
REDIS_PORT = int(os.getenv("REDIS_PORT", "6379"))
# Open connection to redis here and store the client as a property of this␣
 ↪object

redis_client = redis.Redis(host=REDIS_IP_ADDRESS, port=REDIS_PORT, db=0)
```

[23]:
```
# Set the key
key = 'sin_ts'
#key = 'web_ts'
```

[27]:
```
# Get Range (zpopmax and min are destructive they pull the data out of the set)
max = redis_client.zpopmax(key)
#print(max)

min = redis_client.zpopmin(key)
#print(min)
# Put them back
max_member = max[0][0]
max_score  = max[0][1]
redis_client.zadd(key, {max_member: max_score})

min_member = min[0][0]
min_score  = min[0][1]
redis_client.zadd(key, {min_member: min_score})

print("Min",min_score,"Max",max_score)
# Calculate the time interval
dif_score = max_score - min_score
# In milliseconds
print("Milliseconds", dif_score)
# Seconds
```