Real-time data collection framework in virtual reality environment

SUBARCTIC MONKEYS

OUR TEAM

MUIZ

Sensors

SAKIB

Sensors

MARI

Data & VR

BRI

Data & VR

VIRTUAL REALITY (VR)

VR provides **collaboration** personalised environments where to experience co-presence. **Context**, **physical** and **physiological** data from a variety of sources are required to improve the experience. Therefore, the need of a **VR framework** that brings together all the different sources.

CHALLENGE

Create a solution which **collects and integrates data from several sources** for virtual reality **collaboration** and **co-presence** applications with a particular focus on medical training. **Collect sample data** and make standardized data cleanup for analysis.

VR SCENE

An VR scene involving physical activity is used to trigger physiological sensors

SENSORS

EEG, pulse, respiration and galvanic skin response sensors track physiological data of the user

DATABASE

A free cloud database is used to store the data: MongoDB Atlas

REQUIREMENTS

REALTIME DATA COLLECTION

VIRTUAL REALITY ENVIRONMENT

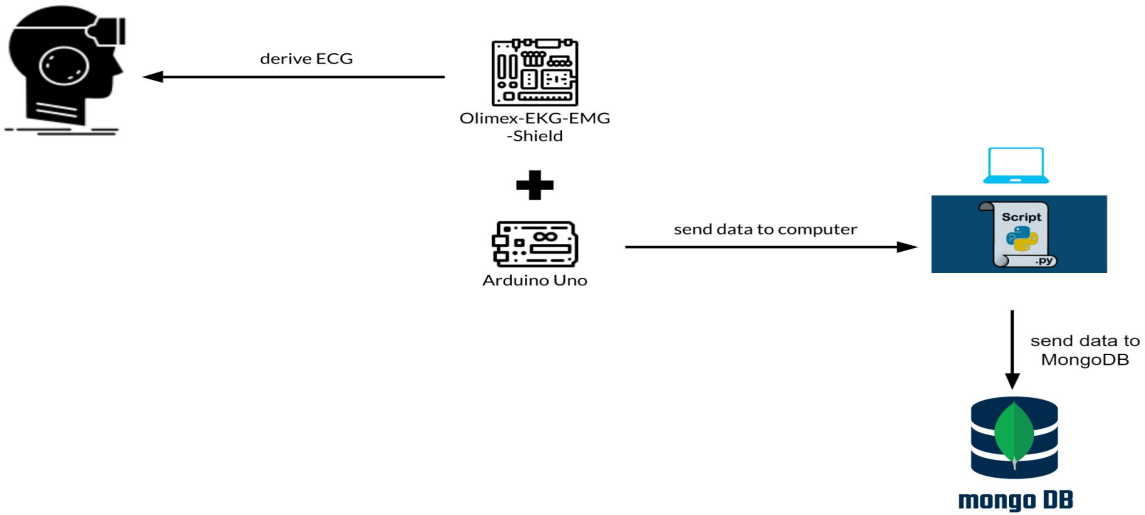
CONTEXT, PHYSICAL AND PHYSIOLOGICAL DATA

MULTIPLE SOURCES

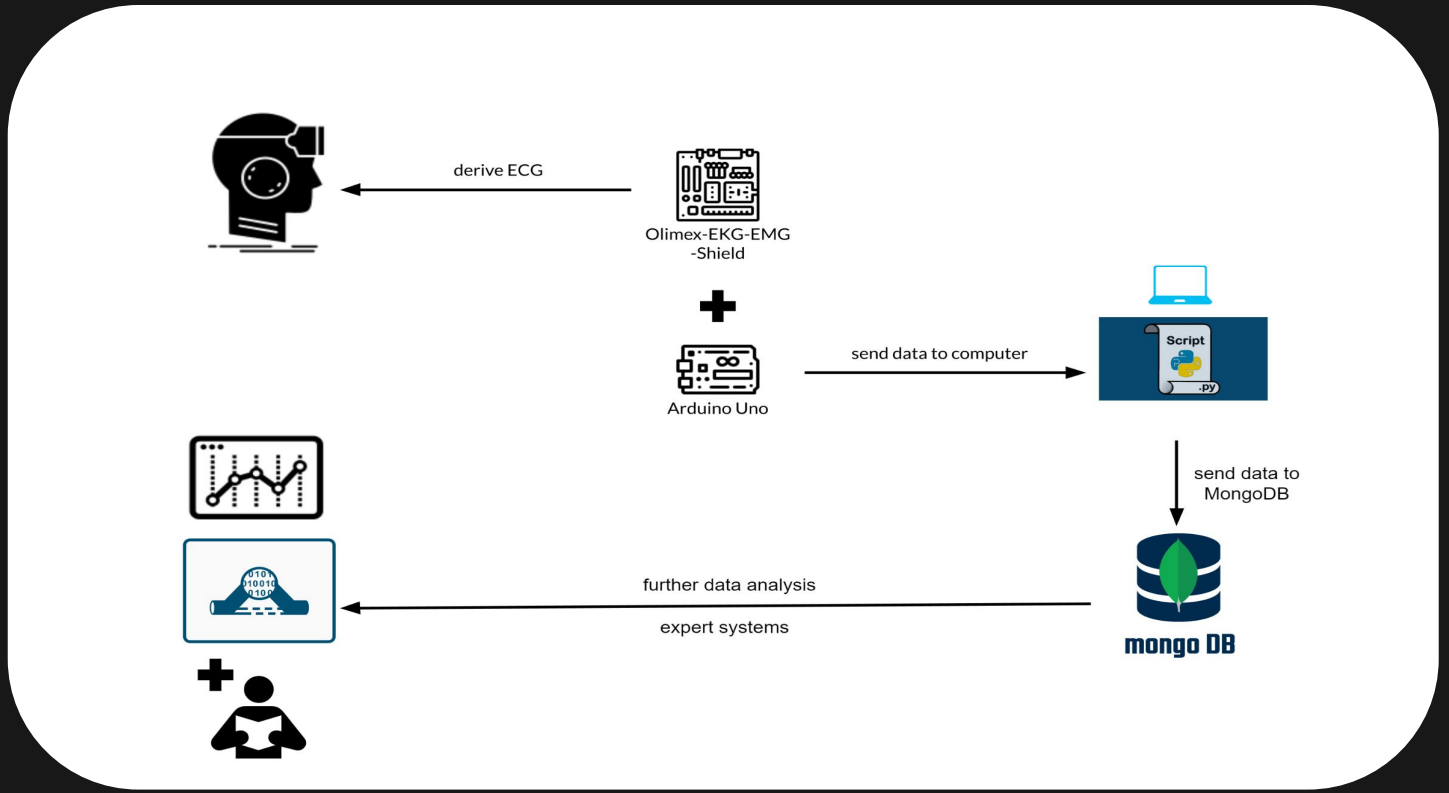
STANDARDIZED DATA

OPEN SOURCE FRAMEWORK

SYSTEM ARCHITECTURE



POSSIBLE EXTENSIONS



COLLECTED DATA

EYE TRACKING
HoloLens 2

- Timestamp (msec offset)
- X, Y, Z (gaze)
- TargetX, TargetY, TargetZ (stimulus)

HEART RATE
Olimex

- Timestamp
- Value of the ECG/EKG

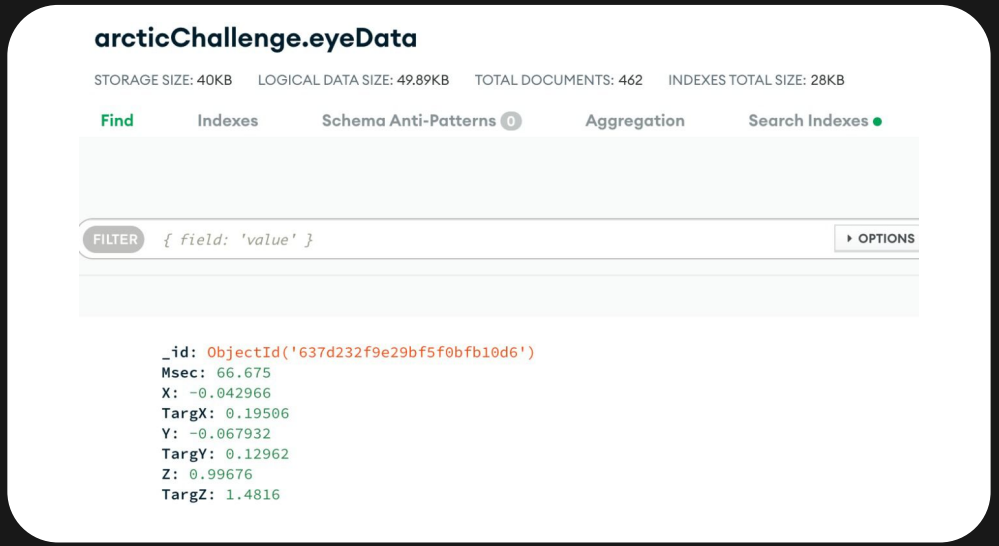
EYEDATA

heartData

Data collected from the sensor

eyeData

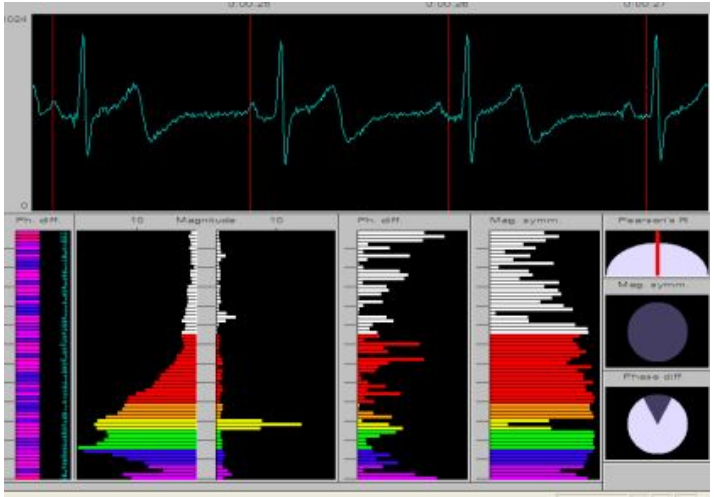
Data collected from the VR



HEARTDATA

heartData
Data collected from the sensor

eyeData
Data collected from the VR



FRAMEWORK

Simple

Standardized

Lightweight

subarctic.py

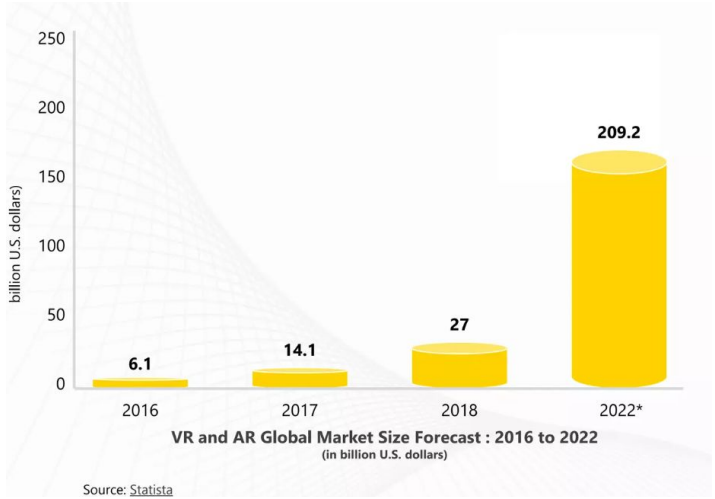
The user has to create a MongoDB account, create a database, and retrieve the connection URI. With the **subarctic** python module it is possible to:

- **Create a collection**
- **Delete a collection**
- From a .csv file, **insert data into a collection**
- **Watch for changes** in database
- **Write data from Arduino** to database

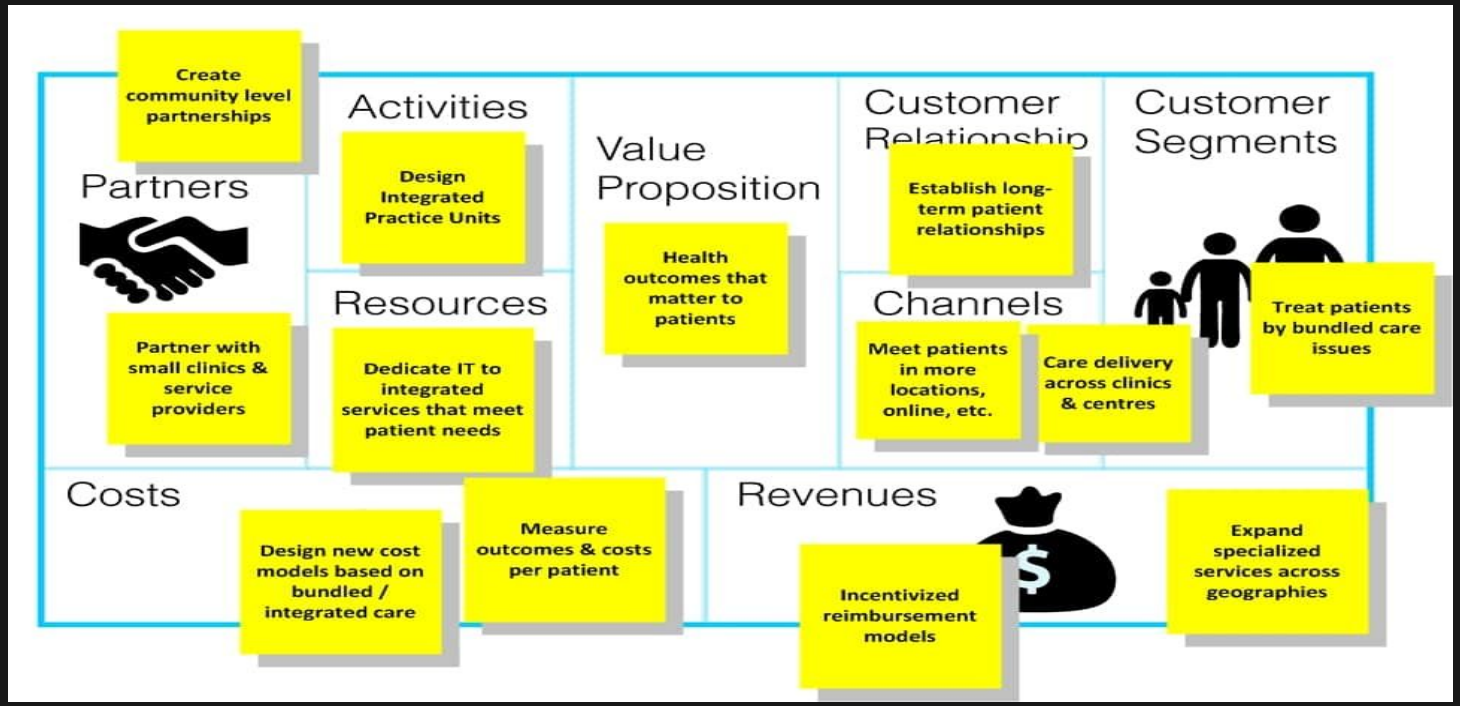
The user only needs to **import subarctic** into his/her script without dealing with low-level pymongo library.

BUSINESS POTENTIAL

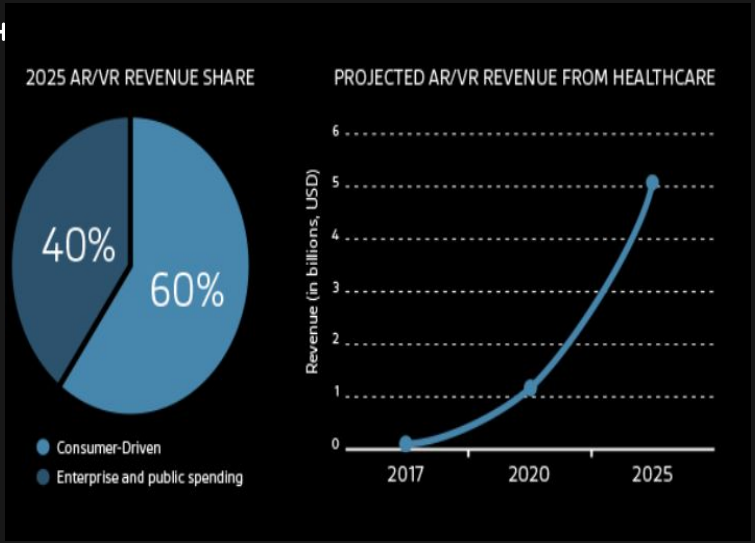
1. Medical training
2. Treatment of patients
3. Pain management
4. Physical therapy
5. Rehabilitation
6. Health education
7. Fitness

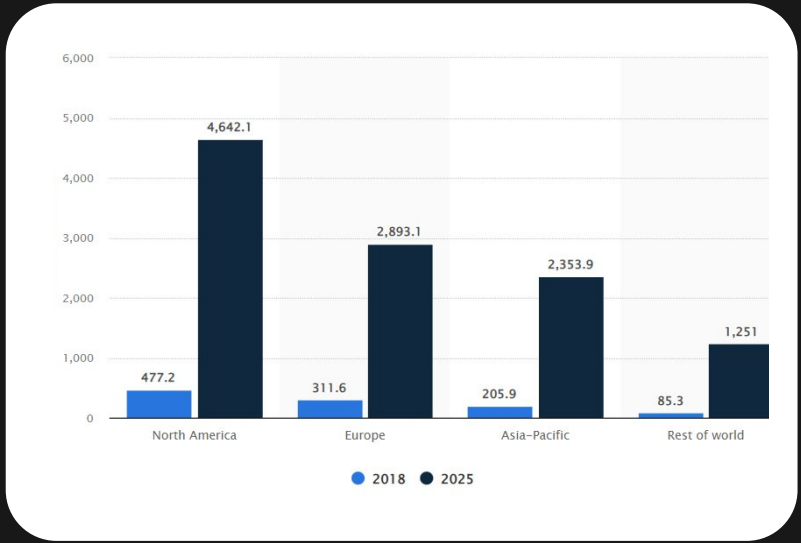


BUSINESS CASE



PROJECTED HEALTHCARE GROWTH IN VR





FUTURE DEVELOPMENT OF THE PROJECT

1

INCLUDE MORE SENSORS

To retrieve indoor contextual data, physical and physiological data

2

DATA ANALYSIS

Analyse the data to improve the co-presence in the environment

3

VR SCENE

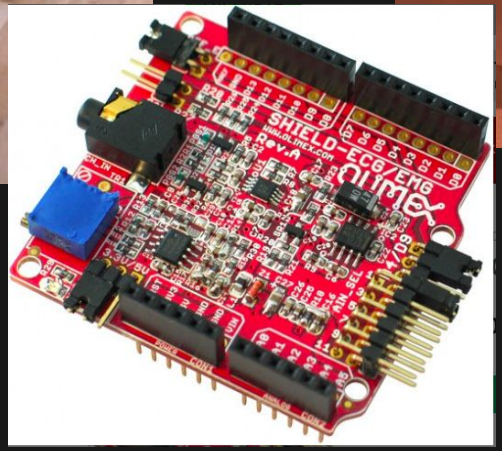
Define a scene that triggers sensors in a supervised medical environment

4

DEPLOYMENT

Integrate components of the system into preventative healthcare plan

DEMO



DEMO #1

DEMO #2