WEKIT Experience Capturing API

Task 3.3

presented by Roland Klemke, Open Universiteit
roland.klemke@ou.nl

T3.3 Experience capturing API

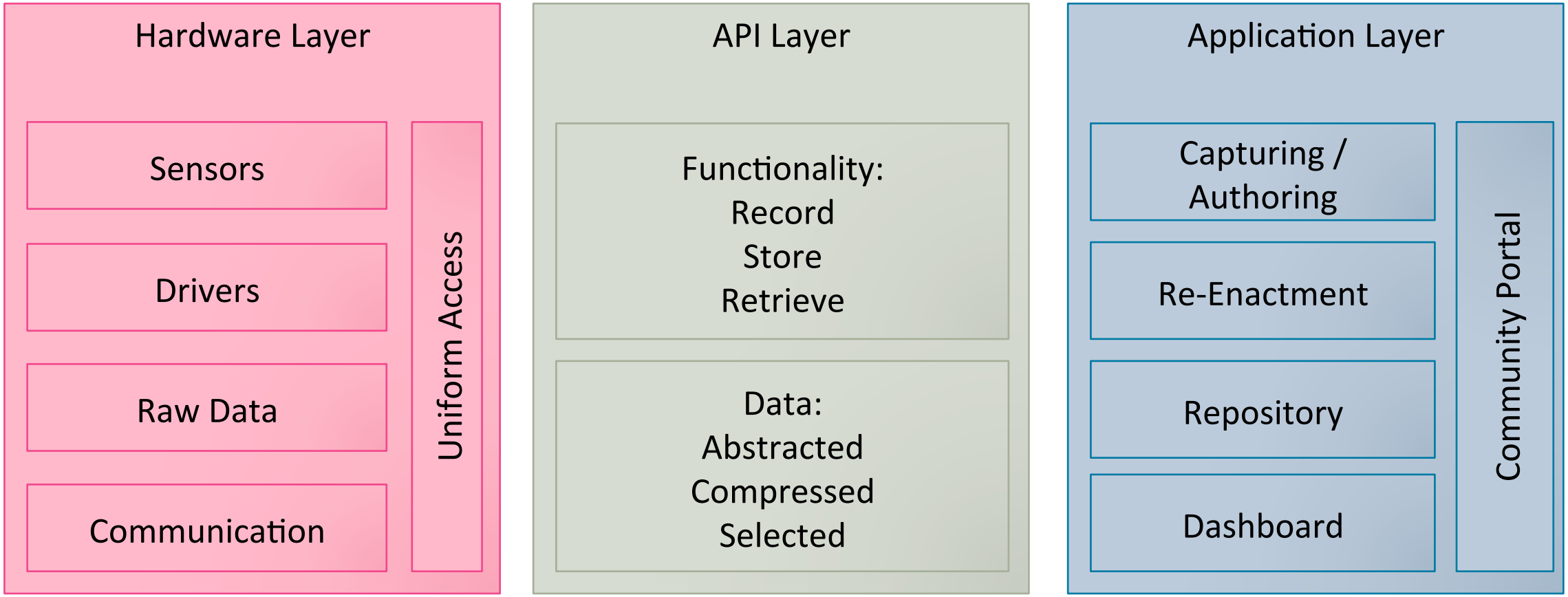
UiT	OUNL	MP
8	8,5	10

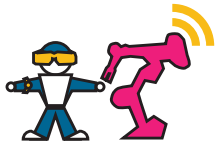
Offers an abstracted sensor fusion API to connect to the various sensor values using high level methods for connecting, retrieving and visualizing (e.g., in a dashboard) the captured raw data as needed e.g. by T2.3

Lead participant OUNL, contributors: MP, UiT

D3.3 Software prototype with sensor fusion API specification and usage description. A prototype and a short report (M12/M27)

General Architecture





Sensors to be considered

Motion

- Position
- Acceleration
- Orientation
- Fokus (Eye-Tracking)

Camera and Video

- Resolution, Rate, Orientation, Fokus
- First person, Third person
- Overview, Detail view
- Object recognition

Audio

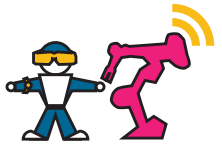
- Environmental audio
- Communication audio

Biosensors

- Heart rate
- Blood pressure
- Skin conductivity

Contextual/environmental sensors

- Temperature
- Light conditions



Requirements 1

Data formats
- Input formats from sensors
- combined output formats

Measuring frequency
- different reaction times of different sensors
- different change rates of measured values

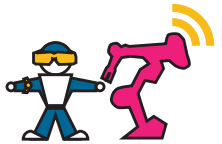
Accuracy and redundancy
- alternative capturing methods for same feature (e.g. leap motion vs. kinect vs. myo)

Amount of data
- full data stream
- compression

Abstraction
- Posture & Gesture
- Object recognition
- Mental state / health state

Architecture
- Permanently online in all scenarios?
- Sync offline/online?

Hardware constraints
- Wearable devices also for capturing
- Limited connectivity/storage capacity



Requirements 2

Interactivity

- Setup and initialization of capturing
- Interactive enabling of specific features (e.g. high resolution camera on/off, audio recording on/off, varying sensor combinations)
- Interactive tagging, setting of marks
- Hands free operation vs. recording assistance (2nd person)

Functionality

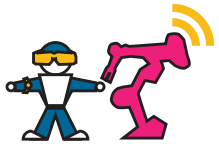
- Record
- Store
- Retrieve

Output

- data delivered to visualization (T4.1) and information adaptation (T4.2)

Environmental requirements

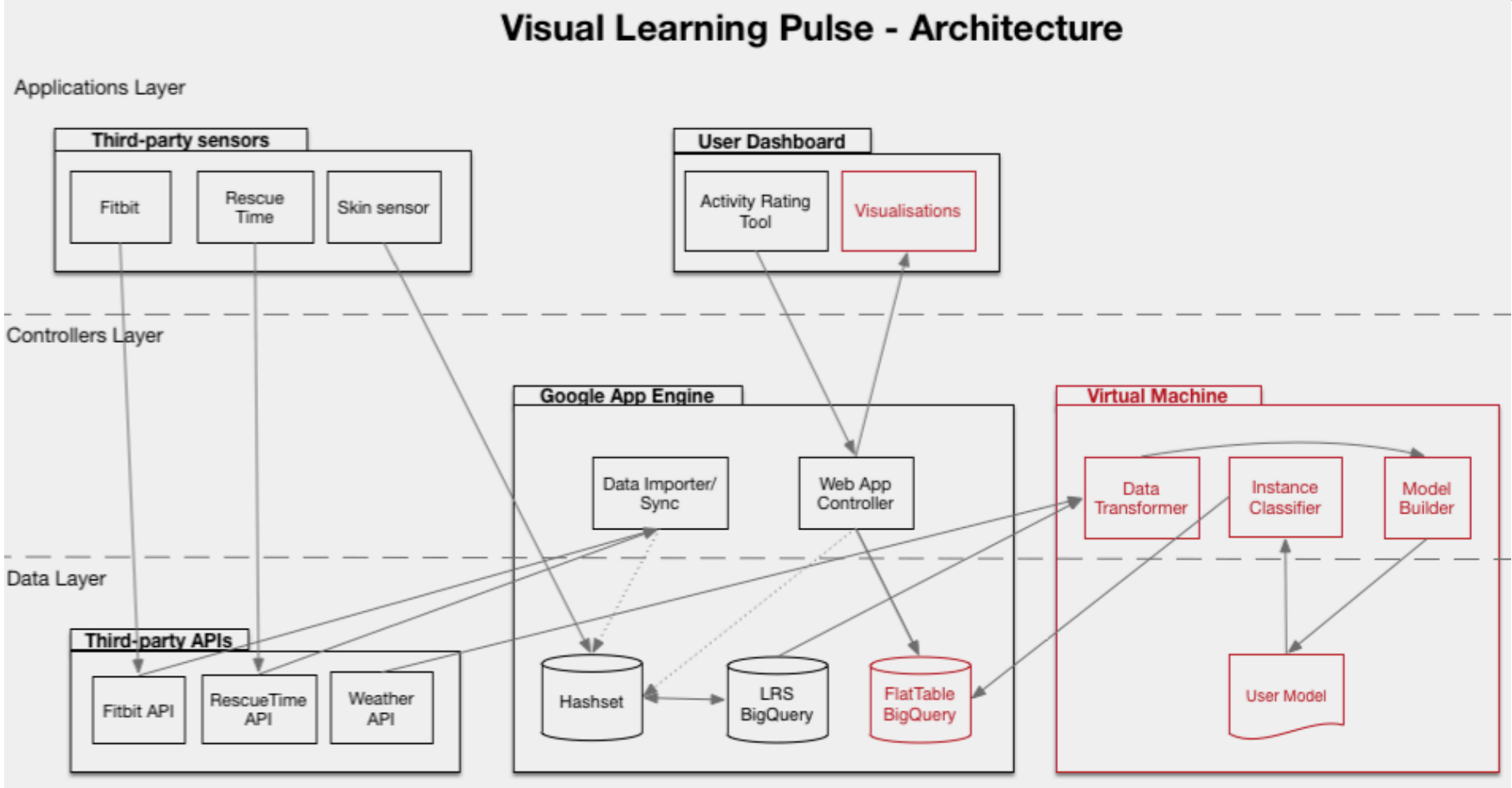
- Noise
- Light
- Temperature
- Humidity
- Cleanliness
- Safety



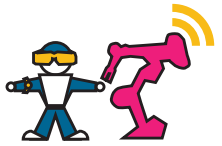
Data

Duration\Device	Kinect	Myo Armband	Leap Motion (2 hands)	1 file (Kinect, Myo, Leap (2 hands))
1 Frame	7 KB	802 Bytes	2.5 KB	11 KB
1 Frame (Zip)	1.5 KB	535 Bytes	851 Bytes	2.4 KB
60 Frames	343 KB	4 KB	32 KB	379 KB
60 Frames (Zip)	41 KB	1.2 KB	5 KB	47 KB
3600 Frames	20 MB	205 KB	1.8 MB	22 MB
3600 Frames (Zip)	2.4 MB	37 KB	222 KB	2.6 MB

Possible Solutions 1:
XAPI used e.g. in Visual Learning Pulse



Possible Solutions 1: XAPI used e.g. in Visual Learning Pulse



Pro:

- established and used in other projects
- explicitly for tracking across devices

Contra:

- not designed for real time sensor data and high frequency tracking events
- rather designed for individual tracking events (large scale)
- WEKIT experience capturing API is comparably low-level
- XAPI would rather help to track success in a single learning task

EXPERIENCE API™

xAPI

Interoperable

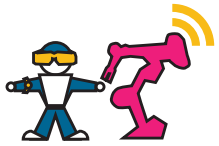
NoSQL approach

Subject Verb Object triples

Semantic rails for learning

```
{
  "actor": {
    "mbox": "mailto:jason.haag.ctr@adlnet.gov",
    "name": "Jason",
    "objectType": "Agent"
  },
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/skipped",
    "display": {
      "en-US": "skipped"
    }
  },
  "object": {
    "id": "http://adlnet.gov/expapi/activities/video",
    "definition": {
      "name": {
        "en-US": "Misconduct in Clinical Trials"
      },
      "description": {
        "en-US": "video on misconduct in clinical trials"
      }
```

Possible Solutions 2: Commercial frameworks



Some commercial IoT frameworks available

- https://temboo.com (IoT software stack),
- https://thingspeak.com (Open Data platform for IoT),
- https://www.tempoiq.com (IoT application framework),
- https://xively.com (IoT product management platform),
- http://www.sense-iot.com (sensor data storage and management framework)

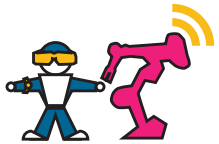
Pro:

- Coverage of many sensors
- Scalability
- Professional support

Contra:

- designed for specific products / services in mind
- license models vary
- not always enough control over code/API/ functionality

Possible Solutions 3: Open source frameworks



Some low level sensor fusion frameworks available

- http://bsautner.github.io/com.nimbits/ (open source framework for time & geo-stamp data),
- https://github.com/ethz-asl/ethzasl_msf (Time delay compensated single and multi sensor fusion framework)
- https://github.com/memsindustrygroup/Open-Source-Sensor-Fusion (Open-Source-Sensor-Fusion)

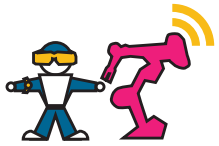
Pro:

- Coverage of many sensors
- Full control after adoption

Contra:

- usually very low level
- application layers missing
- development/maintenance perspective unclear

Workshop: Experience capturing: extracting requirements



Goals

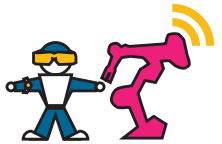
- additional hardware and software requirements for experience capturing
- possible hard-/software constraints and conflicts (e.g. eyetracker won‘t work with AR-Glasses)
- technical setup and constraints (e.g. environmental constraints such as noise, light, temperature, humidity, cleanliness, saftey)
- two different technical set-ups for capturing and re-enactment (devices, sensors, visualisation, interactivity)
- capturing API as a middle layer: requirements for API
- How to best support re-enactment use cases

Orga

- 1 hour, small groups
- Each group presents results in two minutes presentation
- Results collected on GDrive

Tasks

- Reflect framework requirements + use cases collected so far
- Add new requirements



Q & A

Presented by Roland Klemke
roland.klemke@ou.nl

Disclaimer
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 687669. http://wekit.eu/