Requirements Analysis Document

BHRP: Biometric Healthcare Research Platform

This document introduces the background, functional, and nonfunctional requirements of the system and system models. Scenarios in which the system will be used and more detailed system models are documented separately on Google Drive.

Revision history

Date	Version	Description
24/10/2017	0.1	First draft presented and discussed at project meeting on 25th of October, 2017 to align clear expectations for the platform.
01/11/2017	1.0	First deliverable of the BHRP project, covering a high-level overview and analysis of system requirements for the platform. This includes a completed analysis object model and use case diagram. However, some aspects are still open for more detailed analysis later in the project (if the need arises), including detailed use case descriptions and additional dynamic models.
12/04/2018	2.0	Updated version as part of the second deliverable (SDD) of the BHRP project to keep the two documents aligned. ● Split the single object model in a model for ‘study protocol’ and ‘study deployment’, including introduction of ‘study’ and ‘consent’ and minor renaming of entity objects. ● Renamed ‘stimulus’ entity to ‘output’ and introduced ‘stimulus’ as a specialization of ‘task’ instead (including both output and measure during the same time interval). ● Renamed ‘user action’ entity to ‘interaction’. ● Added ‘olfactory’ as ‘output’ (before ‘stimulus) specialization. ● Included UI mockups for minimal viable product.

1 Introduction

1.1 Purpose of the system

The rapidly increasing interest within research to monitor and diagnose diseases in people's daily lives ('in-the-wild'), in addition to the *gold standard* laboratory setup ('in-the-lab'), requires an IT infrastructure which can combine recordings of various types of mobile sensors (e.g., *electrocardiography*, *accelerometry*, *electrodermal activity*, …) with self-reporting tools (e.g., questionnaires, *ecological momentary assessments*, ...). Up until recently, infrastructures supporting this have mostly been custom-built for the purpose at hand. However, there is a great deal of similarity between such systems.

The aim of this project is to develop a software infrastructure which can collect and aggregate data collected in-the-wild with data collected in-the-lab and prepare it for state-of-the-art analytic methods such as *machine learning* and *classification algorithms*, thereby empowering researchers to conduct biometric healthcare research more efficiently. In addition to handling *data ingestion* and storage, the platform will allow researchers to define the studies they want to run: configuring the sensors and devices used for data collection, defining when data should be collected, and defining the output (e.g., stimuli) to be presented to participants over the course of the experiment. Based on such a *study protocol*, the platform will enable subsequent deployment and enable researchers to monitor the state of participants over the course of the study and support two-way communication.

1.2 Scope of the system

The scope of the system is limited to the necessary architecture and basic code templates which can be used to implement and run concrete software applications and integrations with specific devices used in a study. Hence, this document focuses on specifying the requirements for the basic platform, including its architecture, runtime (i.e., non-functional) requirements, as well as basic application programming templates, which are to be used by programmers to implement applications using this platform. Even though the specification of the platform is driven by the two use cases from the BHRP project—epilepsy and autism—this document does not specify the requirements for these end-user applications.

The focus lies thus on *providing the common foundation needed to run biometric healthcare research studies*. However, the code templates will include a basic user interface which provides the means to set up and run a study, integrating with the set of devices and components which will be supported as part of the use cases tied to this project. In other words, the system will provide the means to run a study using default components 'out-of-the-box', but

will not provide any end-user functionality to customize this experience; this will require further extension beyond the present platform.

1.3 Objectives and success criteria of the project

The primary objective of the platform is to support the two clinical studies in the BHRP project for which it is developed (see Innovation Fund Denmark grant application): one involving patients with epilepsy and another involving patients with autism spectrum disorder. Studying these two conditions in the lab has been challenging; seizures can be infrequent and a lab environment can induce stress in patients with autism. Therefore, using *multimodal* wearable sensors in the patient's free-living environment shows significant promise. The platform will be tailored specifically to the devices and measures required for these two use cases. A first success criteria is the successful development and deployment in at least one clinical study for each of these two use cases, enabling researchers to perform in-the-wild data collection.

Although the platform is developed with these two use cases in mind, a secondary objective is to create a more generic modular software architecture which can be extended programmatically to support other healthcare related 'in-the-wild' studies. By using the platform in at least one other healthcare project, unrelated to the project for which it is developed, the secondary objective is achieved.

1.4 References

- Innovation Fund Denmark - Grand Solutions, grant application: *"BHRP: Biometric Healthcare Research Platform for research in psychiatric and neurological diseases using sensor technologies."*

2 Current System

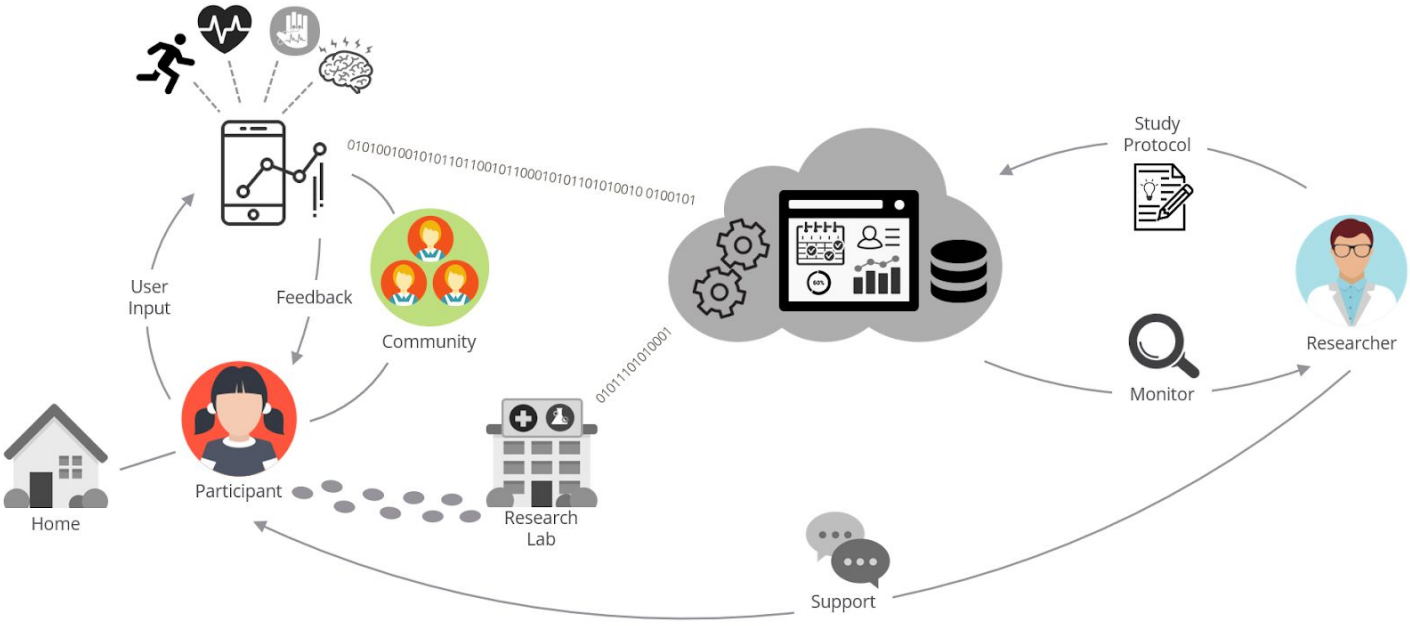
The project lead—iMotions—has created a leading software platform for academic and commercial human behavior research in laboratory settings. With this platform, researchers can easily perform multimodal research, measuring and combining eye tracking, facial expressions, *electrodermal activity (EDA)*, *electroencephalography (EEG)*, *electromyography (EMG)*, and *electrocardiography (ECG)*. iMotions eliminates the need to navigate multiple software tools (one for each individual measure) by providing a single easy-to-use integrative solution. In addition, the software allows specifying stimuli to be presented during a study and the tasks participants should perform.

The current system, however, does not support 'in-the-wild' research. The software only runs on desktop computers and does not integrate with wearable sensors which can collect data over longer periods of time. The platform presented here will extend on the functionality of the current iMotions software by enabling in-the-wild research.

3 Proposed System

3.1 Overview

The overall platform is called the **Biometric Healthcare Research Platform (BHRP)**.



We will often refer to the following subcomponents:

- **Participant user interface (UI)**: The user interface (front end) which participants in research studies interact with, primarily using their mobile devices (e.g., smartphone).
- **Mobile back end**: The infrastructure used on the mobile device to handle data ingestion and pass data to the cloud.
- **Cloud**: The online data store for all study protocols and uploaded measures which can be queried to gain early insights and perform analytics.
- **Lab software**: The software used to run studies in a traditional lab environment.
- **Researcher dashboard**: The user interface used by researchers to define and upload their study protocols to the cloud.

3.2 Functional Requirements

The platform needs to provide support for the following processes and allow extending or customizing them through the implementation of new components which can easily be integrated into the platform without extensive modification (i.e., by extending base components):

- **Define study protocol**: Define the studies to run and how to deploy them: configuring the sensors and devices used for data collection, defining when data should be collected, and defining the output (e.g., stimuli, tasks, ...) to be presented to participants over the course of the experiment. While a study is running, it should be possible to modify parts of the study protocol (e.g., include new tasks) without modifying the original. This supports running pilot studies where the protocol is not yet set in stone.
- **Export study protocol**: Export a description of the study protocol which can easily be incorporated into ethical approval forms or consent forms for participants.
- **Mobile data sampling**: Collect data from three sources: (1) sensors built into mobile devices of study participants, such as *accelerometer*, GPS, and microphone; (2) external sensors like *ECG* and *EDA*; and (3) from monitoring the use of the mobile device itself, such as sampling the number of phone calls, text messages, and applications used.
- **Lab studies**: It should be possible for data collected as part of traditional lab studies (including studies ran using the current iMotions software) to be integrated (be made part of the study protocol and aggregated on the cloud) with the data collected 'in-the-wild'; either automated, or when integration is not possible, by uploading the study results manually. Similarly, besides mobile data sampling, study participants can also run more complicated studies on a stationary device in their home (a 'home lab').
- **Self-reporting**: Collect self-reported data from users of the platform via questionnaires, diary, or *ecological momentary assessments (EMAs).* Examples of self-reported data include daily reporting of mood, stress, sleep duration and quality, exercise type and duration, medication adherence, a seizure diary, etc.
- **Data processing**: Pre-process data on mobile devices of study participants, prior to uploading it to the cloud. E.g., to clean and aggregate data, or to calculate existing biomarkers (e.g., activity recognition of everyday activities like walking, stair climbing, and running, based on *accelerometry*). More advanced data processing should be supported by allowing to query the cloud for ongoing as well as historical study data.
- **Monitor**: Researchers should be able to monitor study progress and incoming data of all participants in order to assess any complications which might arise (e.g., non-compliance, quality of the collected data). Likewise, although not all collected data should be immediately visible to participants (researchers need to be able to specify which *feedback* is provided to participants), they commonly need to be able to see study progress and the status of connected devices.
- **Feedback**: Provide feedback, optionally in the form of data visualizations, to the mobile device of study participants in order to alert, motivate or educate. Such feedback can be triggered based on incoming sampled data. For example, issue a warning a doctor should be contacted in case blood pressure readings are above a certain level.
- **Incorporate patient network**: Data collection and feedback should not be restricted to the devices of patients; it should be possible to include (devices of) the patient network (relatives, caregivers, …) in the study. For example, when a warning is triggered for a patient, also notify relatives.
- **Communication**: Two-way communication between researchers and study participants in order to request ad-hoc information and provide assistance when needed.

- **Privacy and silent mode**: In order to align the expectations of a study with the daily life of participants, it should be possible for participants to configure when they do not want to be interrupted (e.g., during sleep). In addition, they need to be able to temporarily disable data collection (for privacy reasons). However, whether or not (and when) these options are available should be configurable in the study protocol.

3.3 Nonfunctional Requirements

3.3.1 Quality requirements

Usability:
For developers, documentation on how to use the platform should be provided as:

- Documentation of the architecture in a *System Design Document (SDD)*.
- Documentation of individual components: at a minimum, all public classes and non-trivial properties and methods should have a description in code.
- Basic code templates for all extendable components in the platform, including examples on how to use them in the form of concrete existing implementations.

Dependability (reliability and safety)**:**
Fault detection and resolution:

- For all devices connected to the platform, it should be possible to see whether a connection can be established or when the last connection was established.
- For connected devices, when expected data collection (defined as part of the study protocol) does not occur within a configurable amount of time a warning should be made available (as a notification or in an overview).
- When a software component (part of the platform) fails mid-operation (e.g., due to an uncaught exception or mandated shutdown by the OS), an attempt should be made to restart the component and continue normal operation without human intervention. Such failures should be logged.

Considering the sensitive nature of the data processed by the platform:

- Data at rest and in transit between devices should be encrypted.
- Two-factor authentication should be supported for users of the platform.
- All authorization requests should be logged, including rejected requests.
- When within a specified period a predetermined number of consecutive failed login attempts are registered from the same device or with the same user identification, further login attempts must be blocked. Re-enabling the login should be configurable to be time-based (after a timeout) or done manually.
- All access to stored data should be logged, including the user, time, duration (of authenticated session), data accessed, and functionality invoked (e.g., search query). The log should be retained for a configurable duration.

Performance:

- The platform should support the collection, storage, and querying (once it is stored on the cloud) of raw high-frequency sensor data. Data ingestion and storage should thus be designed to scale. For example, a naive implementation will not support EEG data collection at 256 hz per channel over the course of several weeks.
- Data collection and processing on mobile devices should be implemented keeping in mind their limited processing power and battery life. Techniques such as micro-batching and data and computation reuse need to applied where possible and supported for developers using the platform.
- How data is collected (frequency and how often data is uploaded to the cloud) should be configurable and modifiable over the course of an experiment. For example, to reduce power consumption when a mobile device is running low on battery.

3.3.2 Constraints

Implementation:

- The participant user interface and mobile back end needs to run on Android and iOS. However, due to limitations on iOS we can not guarantee that sensors which are not supported by Apple will be fully functional (e.g., data collection in the background).

Interface:

- One option for iOS is to integrate with ResearchKit, created by Apple, which might provide access to operating system functionality not available to us directly. However, the surface area of this framework might differ greatly from the functionality we intend to provide, in which case the mobile back end on iOS might be constrained by ResearchKit.
- A basic integration with the current iMotions lab software needs to be supported. At a minimum, it should be possible to manually upload the result of iMotions lab studies to the cloud so that they are aggregated with the data collected ‘in-the-wild’. Concretely, this means it should be possible to specify the time interval during which the study was conducted. Optionally, this process can be automated from the existing iMotions lab software.

Legal:

Implementations using the platform need to be able to comply with the EU General Data Protection Regulation (GDPR). In addition to the earlier mentioned (mainly dependability) requirements, this implies:

- Data collected by the platform should be possible to export in a non-proprietary format (data portability).

Additional legal platform requirements are based on the data processor agreement made with Region Zealand in Denmark (relevant to the epilepsy use case for which the platform will be used). This includes the danish Executive Order on Security 528 and subsequent amendments. Some relevant requirements (pertaining to technical constraints) listed are:

- It should be possible (in the case of war or similar) to dispose of (destroy) all data processed by the platform.
- Personal data processed on the platform should not be transmitted outside of the EU (unless a specific agreement is put in place).
- The data processed on the platform (without agreement by the data controller) should not be processed by any third-party data processors.
- When in use, it should be impossible for a single developer to push changes into production (i.e., peer code review to prevent a single individual from being able to override security implementations).

Concretely, this means that it should be possible to deploy the platform entirely within the EU on equipment owned, managed, or approved by the data controller and that no data is to be transmitted to components other than the ones described in the *SDD* except upon request by an authorized user of the platform.

In addition, it should be possible to integrate the platform within an ISO27001 Information Security Management System (ISMS). This implies the following documentation is required:

- A list of assets (different components, i.e., server and mobile phones used in a study) which make up the platform.
- Identified vulnerabilities of the assets, documented through a risk-based assessment.

3.4 System Models

3.4.1 Scenarios

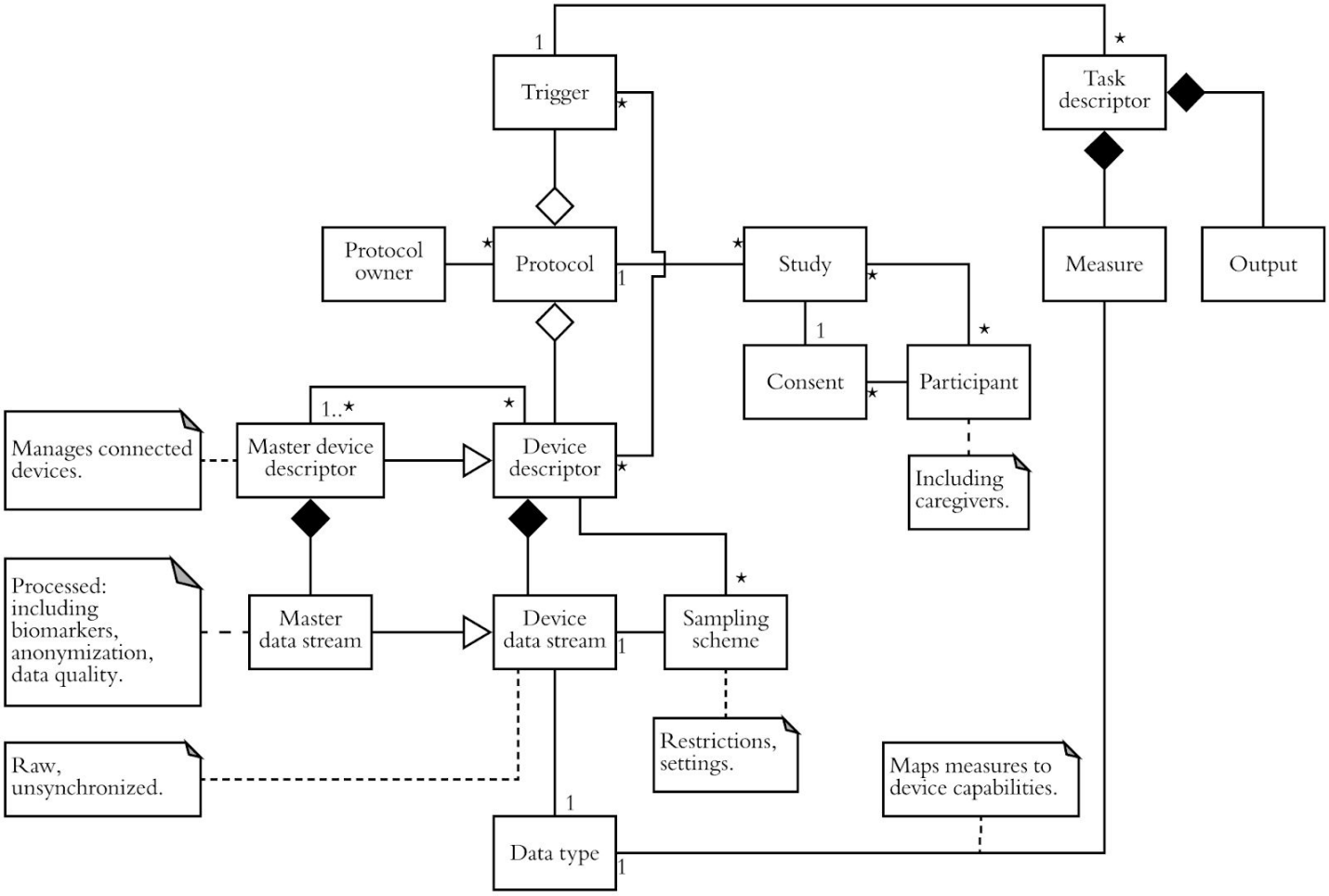
The platform supports *biometric research*. In order to align the research of the clinical partners on this project who will use the platform with the development of the platform, we have identified individual scenarios (clinical studies) which will need to be supported. For each scenario, we have identified the motivating research question, the symptoms relevant to the medical condition being studied, and the related types of data (measures) which can be gathered to determine those symptoms. These scenarios, which act as preliminary study protocols, were used to facilitate requirements elicitation. They will be extended on iteratively over the course of the project, and new scenarios will be included as ideas for new studies come up.

3.4.2 Object Model

The diagrams of the analysis object model are also documented separately.

We identified two main phases which need to be supported when conducting biometric research: (1) setting up a **study protocol** and (2) **study deployment**, or ‘running’ the study. We will model these separately. The following entity objects were identified (some occurring in both models), representing general concepts which can act as the primary extension points for the platform.

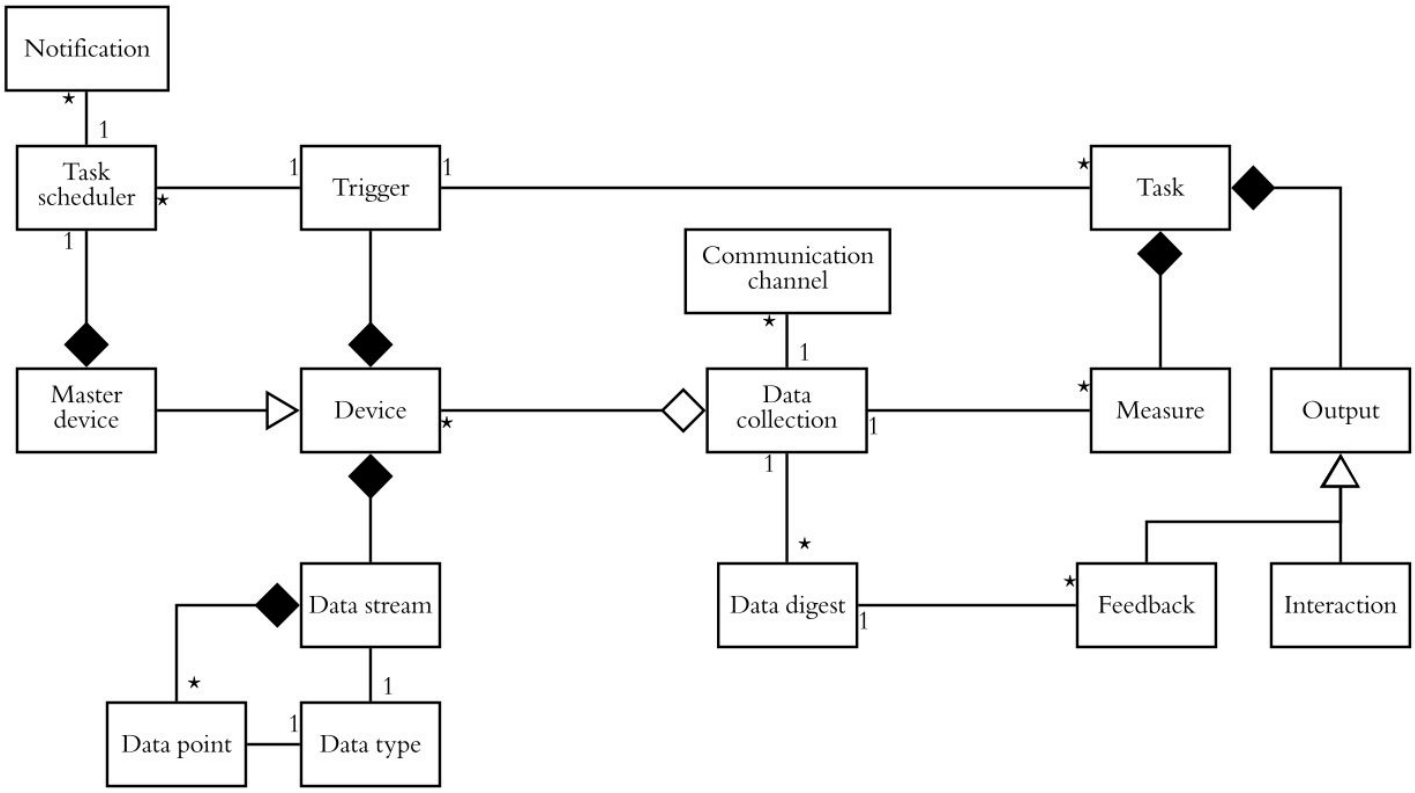
Study protocol:



- **Device descriptor**: Provides a definition of any electronic device, such as a sensor, video camera, desktop computer, or smartphone that collects data which can be incorporated into the platform after it has been processed by a *master device* (potentially itself). Optionally, a device can present *output* and receive user input.
- **Master device descriptor**: Provides a definition of a *device* which aggregates, synchronizes, and uploads incoming data received from one or more connected *devices* (potentially just itself). Typically, a desktop computer, smartphone, or web server.
- **Device data stream**: A stream of data coming from a *device*, unchanged by the platform, encompassing *data points* of one specific *data type* which can be identified in time according to an internal clock of the *device*. For example, an *accelerometry* data stream describes acceleration along an X, Y, and Z axis at certain points in time.
- **Data type**: A description of how to interpret a *data point* collected during study *deployment*: e.g., acceleration along three axes, heart rate, but also survey responses.
- **Master data stream**: A data stream which arises from processing one or more other *device or master data streams* on a *master device* (thus processing is performed by the platform). For example, a moving average or data quality assessment.

- **Sampling scheme**: Specification of sampling settings and constraints for *data streams* available on a *device* (e.g., sampling frequency).
- **Protocol owner**: The researcher who created and owns a *protocol*. This can also be a research group, e.g., defining a *protocol* using a shared account.
- **Protocol**: A description of how a *study* is to be executed, defining the *master device(s)* responsible for aggregating data, and the *trigger(s)* which lead to data collection on said *devices*. For example, a protocol which defines a questionnaire to be sent out to a smartphone every morning, and over the course of the entire experiment expects a continuous heart rate measure.
- **Trigger**: Any condition which triggers *tasks* to be sent to *devices* during *deployment* (one task per device). The condition can either be time-bound, based on *data streams*, initiated by a user of the platform, or a combination of these. For example, a trigger which initiates a questionnaire every morning at 9 AM.
- **Task descriptor**: A collection of requested *measures* and/or *output* to be presented on a *device* over the course of one or more specified time intervals (e.g., only collect data during daytime). Some examples of concrete implementations of tasks: cognitive tests such as response time measurements, IQ tests, measuring arousal while watching a video, but also requesting sensor data collection (without user input) over a specified time interval.
- **Measure**: Time intervals during which *data streams* of a particular *data type* are sampled and stored.
- **Output**: A visual, tactile, auditory, or olifactory output which is presented on a *device* during specified time intervals. For example, an image or video.
- **Study**: An instantiation of a *protocol* (intended for *deployment*) with a set of participants which need to give *consent* in order to participate in the study.
- **Consent**: *Participants* in a *study* need to give consent (to the conditions specified by the *protocol*) prior to starting any data collection.
- **Participant**: Any person that collects data or for who data is collected during the *deployment* of a *study* (e.g., patient, caregiver, doctor).

Study deployment:

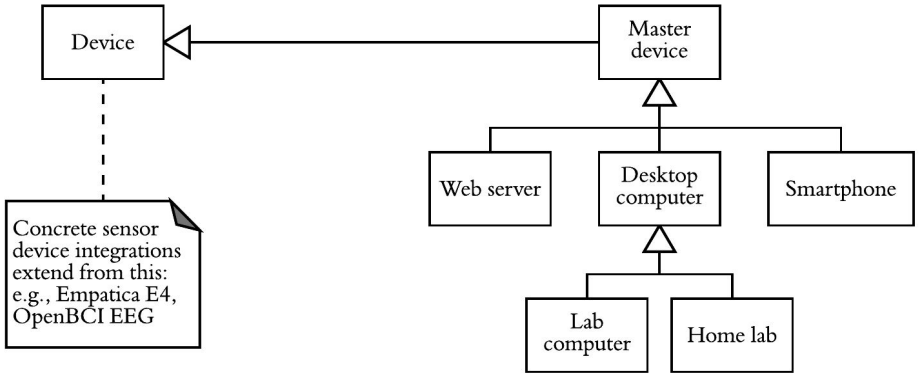


- **Device/master device**: Concrete device (as opposed to descriptors) used during the *deployment* of a *study*.
- **Task**: Concrete task (as opposed to descriptors) executed on a *device/master device* during the *deployment* of a study.
- **Data collection**: Collected data (*task measures)*, while and as the result of running a *study*, optionally processed as *data digests*, which can be accessed by *participants* depending on their access rights. The data collection thus holds the collected data of all participants for a *study*. While a study is running, changes to the study protocol are possible. This will result in a new version of the protocol. Data collected is associated with a specific version of a protocol.
- **Data digest**: Any subset of (possibly processed) *study* data at a given point in time. For example, amount of hours of sleep each night.
- **Task scheduler**: Runs on a *master device* and executes, schedules, or dismisses incoming *tasks* based on relevance, priority, available resources (whether tasks can run in parallel), and participant availability. This decision can also be delegated to participants by showing *notifications*. In order to determine task relevance, communication with other task schedulers is possible (e.g., to determine if a task on a different device has completed). For example, a task scheduler takes care of not initiating a questionnaire while a participant is sleeping.

- **Notification**: Triggered by the *task scheduler* to inform participants of incoming/upcoming *tasks*, allowing to execute, postpone, or (optionally) dismiss tasks. For example, a participant might postpone a questionnaire (task) while in a meeting.
- **Data point**: Data of a particular *data type* pertaining to a particular point in time. For example, one measurement at a specific point in time of accelerometry data.
- **Interaction**: An *output* which in addition requires user input: e.g., a consent form or a survey.
- **Feedback**: Any representation of a *data digest* presented on a *device* while the study is running. For example, a graph showing the amount of hours of sleep per night.
- **Communication channel**: Supports the researchers running the *study* to communicate with the *participants* and vice versa. For example, through instant messaging.

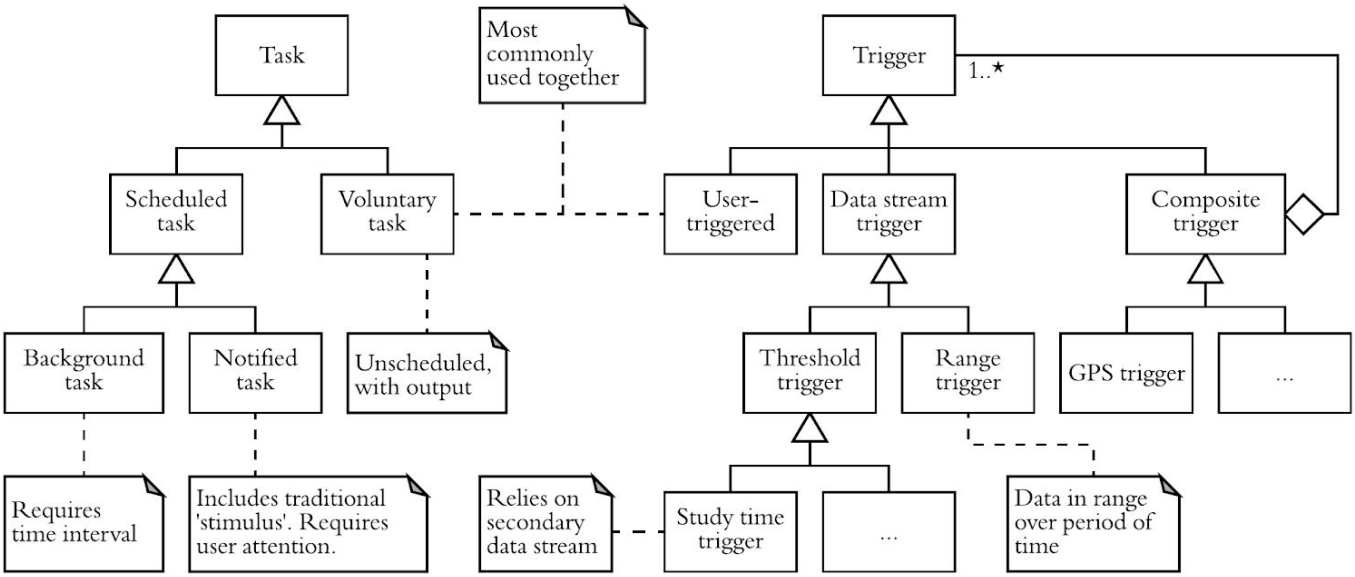
The following diagrams exemplify likely and possible specializations for some of the general entity objects identified above.

3.4.2.1 Device and master device specializations



- **Web server**: A web server can act as a *master device* in case no mobile master devices are used as part of a study. For example, when sensors upload their data directly to the web server (e.g. the default mode in which the Sense accelerometer patch operates).
- **Desktop computer**: A stationary desktop computer can be used to aggregate data coming from connected sensors. Such a desktop computer can be located both in a participant's home and operated by participants (*home lab*), or in a lab environment operated by researchers (*lab computer*).
- **Smartphone**: A smartphone can act as a mobile lab which aggregates data from wearable sensors.

3.4.2.2 Task and trigger specializations



Tasks:

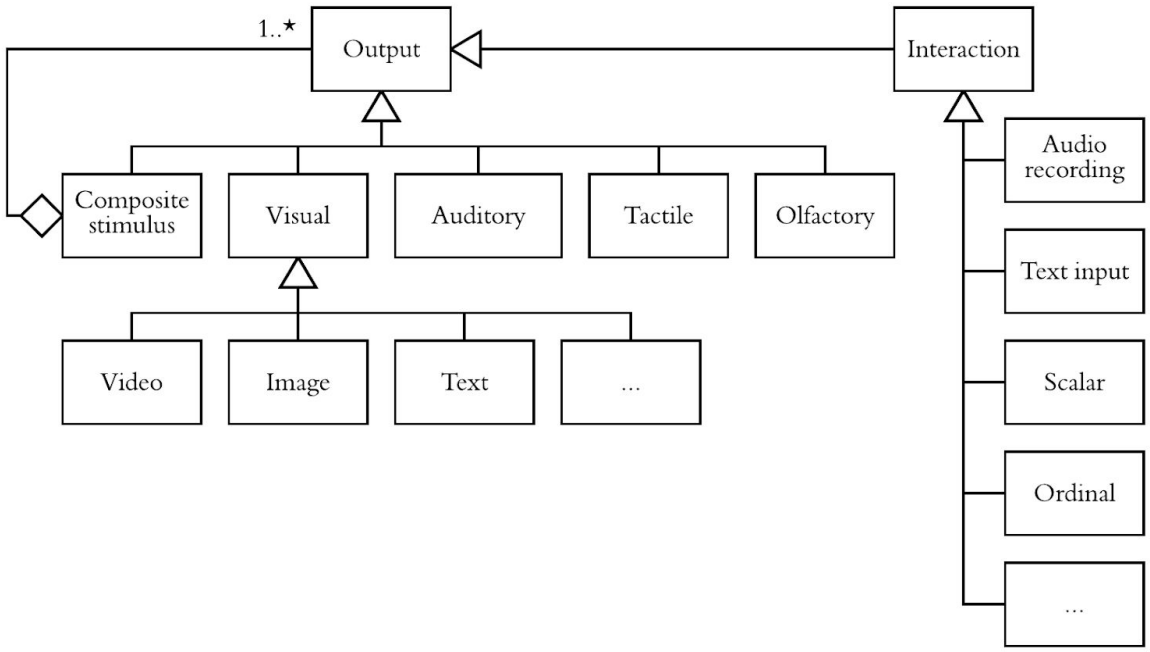
- **Scheduled task**: Tasks which are specified to be executed at or over the course of a certain time interval.
- **Voluntary task**: Tasks which are initiated by the user and therefore do not need elaborate involvement of the *task scheduler* (besides being aware the task is active).
- **Background task**: A task which does not require any output to be presented to the user and can therefore be initiated immediately (at the time interval specified) by the *task scheduler* without notifying the user.
- **Notified task**: A task which includes an output which needs to be presented to the user and therefore requires user attention (e.g., a traditional '*stimulus*' in experimental psychology). Therefore, the *task scheduler* needs to send out a *notification* (or several in case of non-compliance) to the user.
- **Stimulus**: A *notified task* which presents an *output* for a given duration while simultaneously including one or more *measures*.

Triggers:

- **User-triggered**: A trigger which can be initiated by the user, effectively allowing users to open the associated *voluntary task* on demand.
- **Data stream trigger**: A trigger which is initiated based on the processing of a single data stream. The trigger is responsible for decomposing complex data types (e.g., selecting only values for the X component of accelerometry data). Once initiated, a timeout period specifies the period during which the trigger can not be re-initiated.

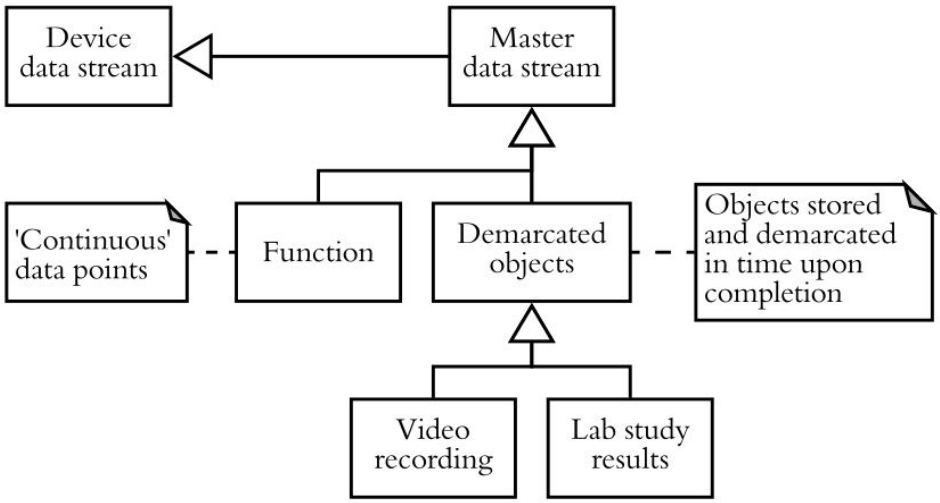
- **Threshold trigger**: A *data stream trigger* which initiates a task (by sending it to one or more *task schedulers*) once a certain value is exceeded.
- **Range trigger**: A *data stream trigger* which is triggered after a certain period of time has passed during which incoming values remain within a specified range.
- **Composite trigger**: Combines a set of triggers by performing logical operations on them. For example, initiate once all included triggers have been initiated.

3.4.2.3 Output and interaction specializations



- **Scalar**: Specify any value within a certain range on a scale.
- **Ordinal**: Select one out of a set of predefined values.

3.4.2.4 Data stream specializations



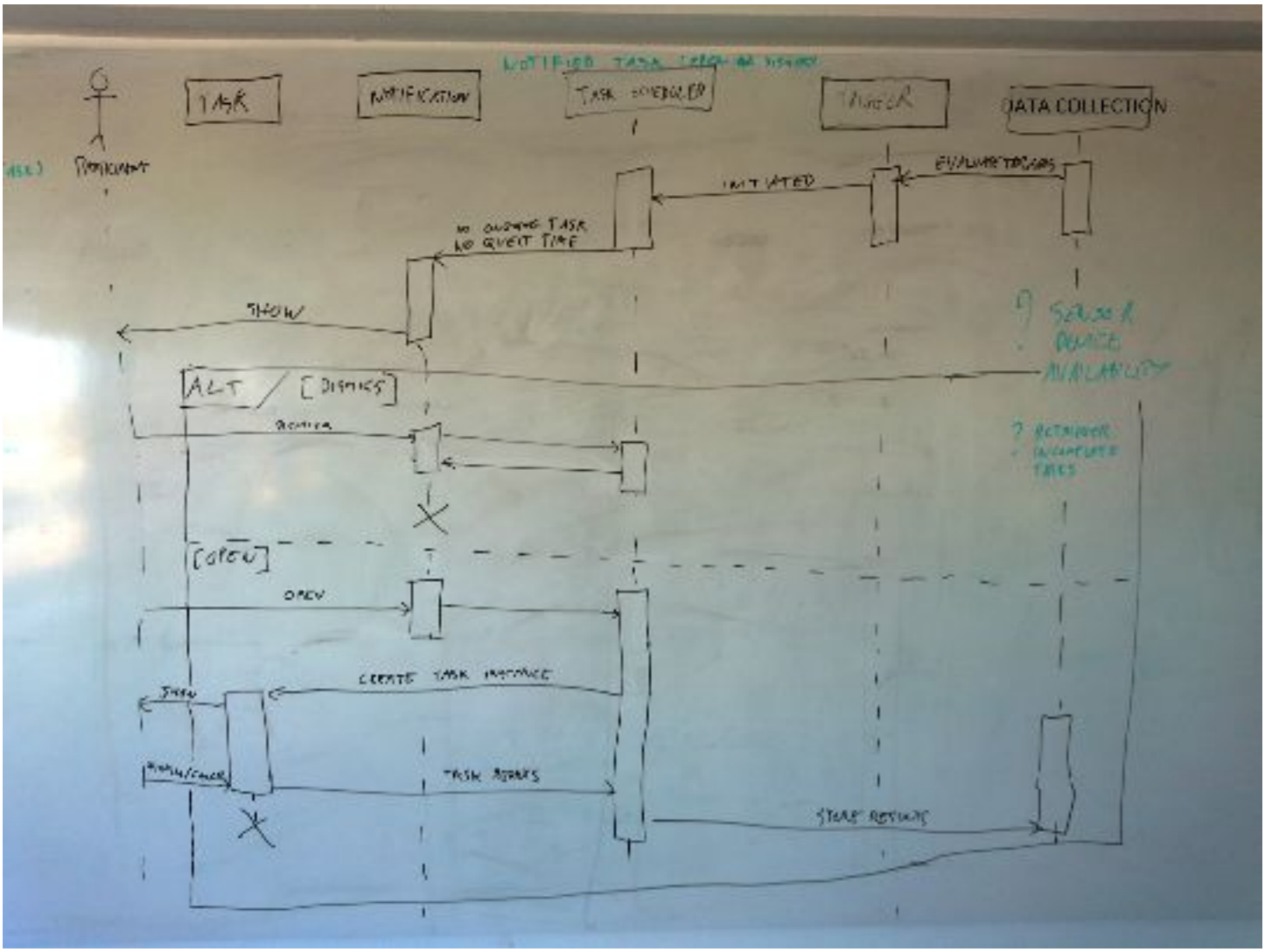
- **Function**: Processes one or more other data streams and produces a single new data stream based on the data in the processed streams. For example, a moving average.
- **Demarcated object**: Data streams which, rather than collecting data continuously, demarcate data collected at an earlier point in time and outline it in time (i.e., when the data was collected and during which time interval). This supports storing ‘blobs’ of data which might not necessarily be accessed in more detail using the platform. For example, an uploaded video recording, or the results of a laboratory study in a proprietary format.

3.4.3 Dynamic models

Although the full analysis object model is provided, only those dynamic models which clarify unclear system behavior are provided.

3.4.3.2 Respond to task notification sequence

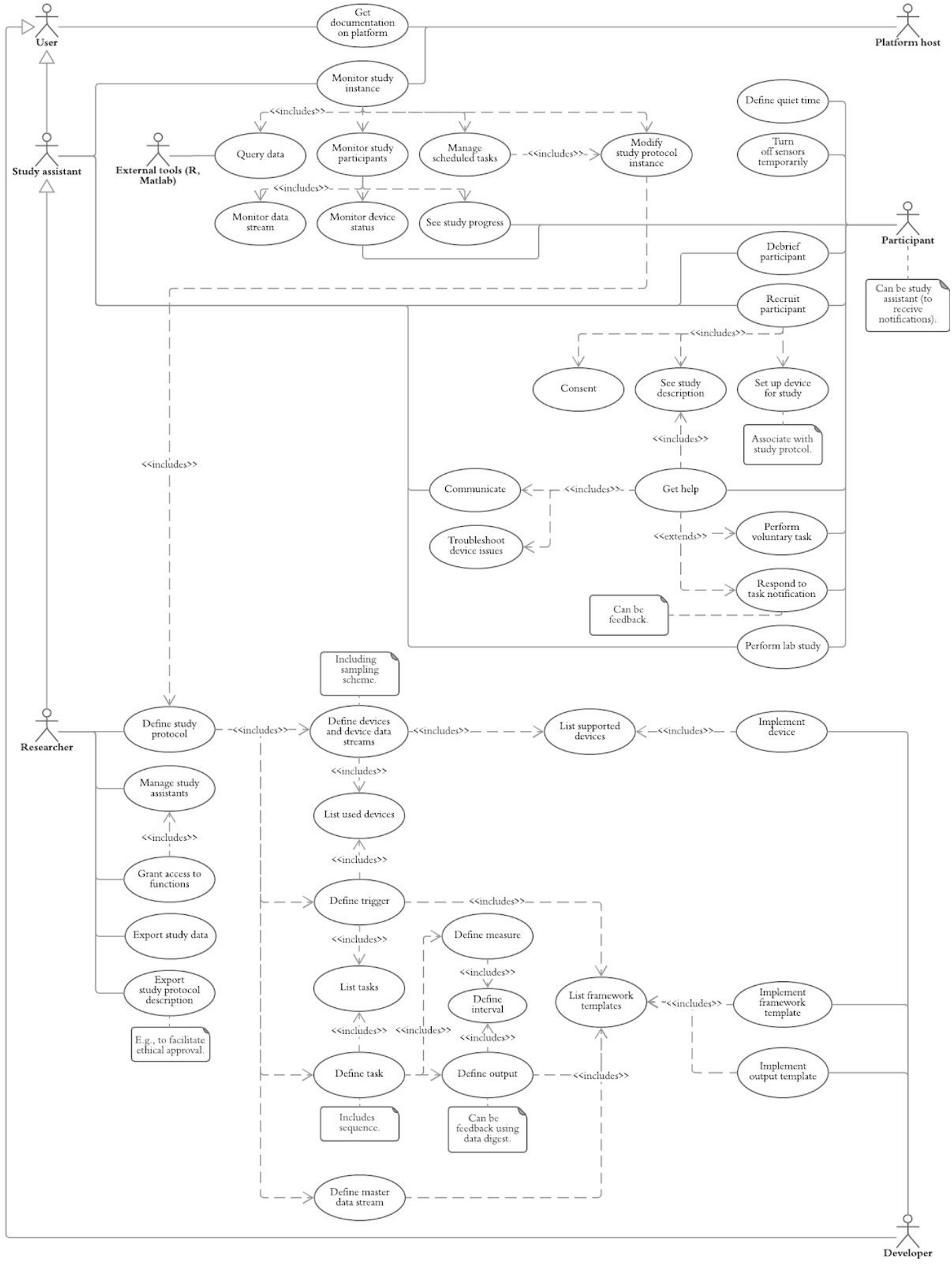
Notified tasks (containing output to be presented to the user) can be dismissed (depending on task priority defined in the trigger) or opened by the participant. The task scheduler ensures a task can be displayed (no other task with output is running).



3.4.4 Use case model

The diagram of the use case model is also documented separately.

The use case diagram depicts all use cases, but detailed descriptions of individual use cases are only included for those which involve complex interactions.


User
Platform host
Get documentation on platform
Monitor study instance
Study assistant
External tools (R, Matlab)
Query data
Monitor study participants
Manage scheduled tasks
Modify study protocol instance
Define quiet time
Turn off sensors temporarily
<<includes>>
Monitor data stream
Monitor device status
See study progress
Participant
Debrief participant
Recruit participant
Can be study assistant (to receive notifications).
Consent
See study description
Set up device for study
Associate with study protcol.
Communicate
Get help
Troubleshoot device issues
<<extends>>
Perform voluntary task
Respond to task notification
Can be feedback.
Perform lab study
Researcher
Define study protocol
Including sampling scheme.
Define devices and device data streams
List supported devices
Implement device
Manage study assistants
Grant access to functions
Export study data
Export study protocol description
E.g., to facilitate ethical approval.
List used devices
Define trigger
Define measure
Define interval
List tasks
Define task
Define output
Includes sequence.
Can be feedback using data digest.
List framework templates
Implement framework template
Implement output template
Define master data stream
Developer

The diagram above contains multiple use cases which apply the word ‘define.’ Generally speaking, the act of defining supports Creating, Updating, Reading, or Deleting (CRUD) within each use case. Limitations in the availability of CRUD operations arise due to various conditional factors such as inherent restrictions, the state of the entity or related entities. For example, neither removing a study protocol which is currently in use (by data collection) nor adding a device to a study protocol when data collection is running would be allowed. A further example would be defining a master data stream before a device data stream has been defined (either by creating, updating, or reading), as the order of operations would not be allowed.

3.4.4.1 Define task

Participating actors	Initiated by *Researcher*
Flow of events	1. The *Researcher* activates the “Define tasks” function from the study protocol overview. 1.1. *BHRP* displays a list of previously defined tasks (‘*List tasks*’ use case) and provides the option to (1) remove existing tasks, (2) select an existing task to edit, (3) create a new task based on an existing task, or (4) create a new (blank) task (containing no measures or output). **If the *Researcher* decides to remove a task:** 1. The *Researcher* selects the task to be removed. 1.1. When the task is used by one or more triggers, *BHRP* warns the task will be removed from the specified triggers. 1.2. *BHRP* provides the option to approve or cancel removing the task. 2. The *Researcher* approves or cancels the action. *2.1.* *BHRP* carries out the desired action and returns to the list of tasks. **If the *Researcher* decides to edit or create a task:** 1. The *Researcher* chooses the task to edit or clone, or selects the option to create a new task. 1.1. *BHRP* shows a form which allows specifying the task name. 1.2. *BHRP* lists the defined measures and output for the selected task including their sequence, i.e., for each the specific time interval when they are to be measured/presented, either at a specific moment in time, or relative to the moment in time when the task is executed by the task scheduler. 2. The *Researcher* fills out the form and can initiate the *‘Define measure’* and *‘Define output’* use case for each measure/output to be added, removed, or modified. 3. The *Researcher* closes the form. 3.1. BHRP stores the changes and returns to the list of tasks.

Entry conditions	• The *Researcher* is in the process of defining a study protocol as part of which tasks need to be defined (‘*Define study protocol*’ use case).
Exit conditions	• When an existing task is removed, the task no longer shows up in the ‘*List tasks*’ use case and is removed from all triggers that use it. • When a task was edited or created, the task from now on shows up in the *‘List tasks’* use case.
Quality requirements	

3.4.5 User Interface

Given that the system described is an *infrastructure*, there will be no one single user interface used to interact with the system. Therefore, instead, the user interface mockups correspond to the frontends used as part of the individual scenarios in which the system is used.

3.4.5.1 Minimal Viable Product (full stack)

The following sketch illustrates a potential user interaction of a researcher (panels 1, 2, and 3 on the left) and research participant (upper-rightmost panel) with the BHRP platform. This represents how a researcher could create a study by defining a *protocol* including one or more *devices* and a *voluntary user task* and accessible *feedback*. Next, the researcher can add a participant to the study by linking their unique devices, and subsequently see an overview of ongoing *data collection*. The participant can use a smartphone-based interface where they can either complete the defined task or view the specified feedback.

Both the researcher and participant interfaces illustrated above have been fleshed out in the form of two interactive mockups created using PowerPoint and Marvel (Marvelapp.com).

- Browser-based Researcher Interface Mockup
- Smartphone-based Participant Interface Mockup

4 Glossary

- **Accelerometry**: Physical activity measured using an accelerometer—a device that measures physical acceleration.
- **Classification algorithm**: An algorithm which identifies to which set of categories (sub-populations) a new observation belongs. Typically used in *machine learning* on the basis of a training set of data containing prior observations.
- **Data ingestion**: the process of obtaining and importing data for immediate use or storage in a database.
- **Ecological Momentary Assessments (EMA)**: A research method which asks participants in a study to stop their ongoing activity at certain times and make notes of their experience in real time.
- **Electrocardiography (ECG)**: The process of recording electrical activity of the heart over a period of time using electrodes placed on the skin from which measures such as heart rate can be inferred.
- **Electrodermal activity (EDA)**: Variation in the electrical skin conductance of humans as measured by placing electrodes on the skin, typically the palms or fingers. This gives an indication of psychological or physiological arousal.
- **Electroencephalography (EEG)**: An electrophysiological monitoring method to record electrical activity (voltage fluctuations) of the brain. It is typically noninvasive, with electrodes placed along the scalp.
- **Electromyography (EMG)**: A technique for recording the electrical activity produced by skeletal muscles.
- **Gold standard (test)**: The most accurate (best performing) test available to researchers under reasonable conditions.
- **Machine learning**: A technique which gives computers the ability to learn (e.g., interpret measured data) without being explicitly programmed.
- **Multimodal**: Combining multiple types (modes) of measurements.
- **Study protocol**: A predefined procedural method in the design and implementation of scientific experiments.
- **System Design Document (SDD)**: Documentation which describes the decomposition of a system, how each individual component maps to hardware and software, who has access to individual components, and how data is stored and flows in between different components.