ESSnet Smart Surveys

Grant Agreement Number: 899365 - 2019-DE-SmartStat

https://webgate.ec.europa.eu/fpfis/wikis/display/EstatBigData/ESSnet+Smart+Surveys+2020-2021

Workpackage 2
Smart Survey Pilots

Deliverable 2.10: MOTUS documentation

Final version, March 1 2021

Prepared by:

Patrick Lusyne (Statistics Belgium, Belgium)
Joeri Minnen (hbits/VUB, Belgium)
Kelly Sabbe (Statistics Belgium, Belgium)

Workpackage Leader:

Barry Schouten (CBS, Netherlands)
Jg.schouten@cbs.nl
telephone : +31 70 3374905

1. Summary

Deliverable D2.10 relates to the WP2.2 Time Use and deals with the more technical details of the MOTUS application used for time-use research. This deliverable describes fromout the viewpoint of the CSPA and the GSBPM architecture (see also Deliverable 2.7) the MOTUS data collection platform as a modular, reusable and shareable platform. Countries can make use of MOTUS to collect their own national time use data having still the focus on harmonization of the data collection process to arrive to reliable and and comparable statistics at EU level.

2. Introduction

In spring 2019, Eurostat launched a call for an ESSnet on Trusted Smart Statistics. This ESSnet aimed at *"preparing the development of European solutions to satisfy European and national needs related to the adoption of trusted smart surveys for official statistics"* and implied "*an investment for moving from exploratory pilots and individual countries' experiments to ESS-wide capabilities"* (Source: Eurostat call for proposals ESTAT-2019-PA11-B-SmartStat). Eurostat also explicitly underlined that "*the action in the call aimed at deepening the collaboration between the experts involved in the ESSnet and the subject matter experts from different statistical production domains."*

Eurostat promotes a clear vision of smart statistics, and critical to that vision is that[1]:

- Future developments should aim at *"the development of a flexible European platform for trusted smart surveys, implementing a set of common (horizontal) functions and configurable services that can be used to build particular instances of trusted smart surveys for specific application domains and/or target areas;*
- *"Such a platform, to be developed at the European level, can be used independently by an NSI to perform national surveys but can also serve as a basis to launch European transnational surveys;*
- *and it should be modular, evolvable, extensible and agnostic to particular application domains. It should provide ready-to-use solutions for horizontal functions. It should*

[1] ESTAT-2019-PA11-B-SmartStat - ESSnet on "Smart Surveys 2019-2021"
(B4462-2019-SmartStat)

allow each platform user (i.e., any ESS member) to instantiate a specific trusted smart survey by selecting and configuring different modules".

In this document, we present an overview of the conceptual characteristics and technical functionalities of MOTUS, a data collection platform developed by the research group TOR of the Vrije Universiteit Brussels and his Spin-off hbits CV.

The goal work package 2.2 of the ESSnet on trusted Smart Statistics is to evaluate to what degree MOTUS's current and future development stages will allow for implementation in different countries and over various domains. The NSI's teaming up in the work package will participate in several small-scale pilot studies to scrutinize MOTUS' capacities regarding shareability and modularity. The conclusions of these pilots will serve as an input for the ongoing development process that characterizes MOTUS.

In line with deliverable 2.7 of the project, four levels of description are used to describe MOTUS:

- A conceptual level description: an explanation of what MOTUS is and what it does (section 3).

- A Logical level description: an in-depth overview of MOTUS's service capabilities, inputs, and outputs (section 4).

- A Physical level description: a description of the implementation of these service capabilities (section 5).

- A Legal-ethical-policy level description: Legal and security description on the operational level of MOTUS.

This document serves as deliverable 2.10 of the project. This deliverable aims to inform the participating countries in the project on the conceptual and technical capacities of MOTUS in particular. Therefore, considerable weight is given to the sections on the logical and physical level description. The legal-ethical level is outside the scope of this document.

3. Conceptual level

MOTUS' roots are in time use research, and today, the application is mainly used to perform Time Use surveys or related research. However, as the architecture of MOTUS fosters by design shareability and modularity, MOTUS clearly shows potential to outgrow its original habitat

Time Use research has a particular methodology, especially concerning the time diary. The need to have respondents filling in this diary, in addition to households and individual questionnaires, distinguishes it from traditional surveys consisting of only a single data collection tool. By definition almost, the methodology of Time Use research, implies a latent modularity.

Time use research evenly struggles, or struggles even more, from drawbacks that traditional survey research suffers from: high respondent burden resulting in low response rates and selectiveness, quality issues (e.g., fewer activities reported), and mounting costs for data collections.

From the early years of the previous decade, voices advocating technological innovation to address these issues grew strong (e.g. DGINS Wiesbaden Memorandum 2011), and the search was open for technology that could improve respondents' responsiveness, render better integration of other sources of information, and boost data collections' efficiency. MOTUS was developed as a direct reaction to this appeal.

Today MOTUS has grown to be a mature production environment for having Time Use Surveys. In the back-office of MOTUS, the research components (questionnaires, diary, context, communication) are defined, a research flow is designed for automated operations, and data management is organized. In MOTUS, it's possible to have multiple studies running simultaneously, even with the same respondent (for panel research purposes). Conceptually the back-office of MOTUS is defined as 'MOTUSbuilder'.

The MOTUS front-office includes all elements that support respondents in registering their replies and activities. The User Interface (design and inclusion of the necessary functionalities) and the User Experience (the way the user interacts with the application) are critical features to this and characterize an application. Consequently, both are evaluated in this work package (2.2). MOTUS runs in a broad range of web browsers and is available as a mobile application for Android and iOS. Sensors also can be connected to MOTUS. To support synchronization between devices, specific server capacities are implemented. Conceptually the front-office of MOTUS is defined as 'MOTUSresearch.'

MOTUS is well referenced and documented on the web:

To use the web application: www.motusresearch.io

To use the mobile application:

App Store, iOS: https://apps.apple.com/be/app/motus-zap-vub/id956934466

Play Store, Android: https://play.google.com/store/apps/details?id=be.byteworks.motus

Respondents can find information on pages like the following (and similar pages depending on the research they participate in).

https://www.motusresearch.io/en/Home

https://www.everydaylife.eu/how-does-the-mobile-app-work/

https://www.everydaylife.eu/about-this-study/

An application like MOTUS has come a long way from a Paper-and-Pencil data collection mode and surpasses CAWI. However, active participation and registration by the respondent are still crucial. Therefore, to continue making progress on the issues raised above, even smarter solutions are necessary.

There still is some margin for improvement in User Interfaces and User experiences enhancement. Based on the SOURCE TM project's remarks and the first pilot tests in this project, MOTUS UI and UX are redesigned and are now being implemented. An introduction to the new UI/UX can be found here: https://youtu.be/S_BbO-SswHc. The current and new design of MOTUS will be tested on its accessibility following the WCAG-guidelines.

Implementation of passive data collection or registration of information seems another way forward. Active registration implies a straight-away input of the respondent in the front-office application (web/mobile) portal. Passive registration bypasses the front office portal by including other information sources without the respondent's active involvement (except for his consent).

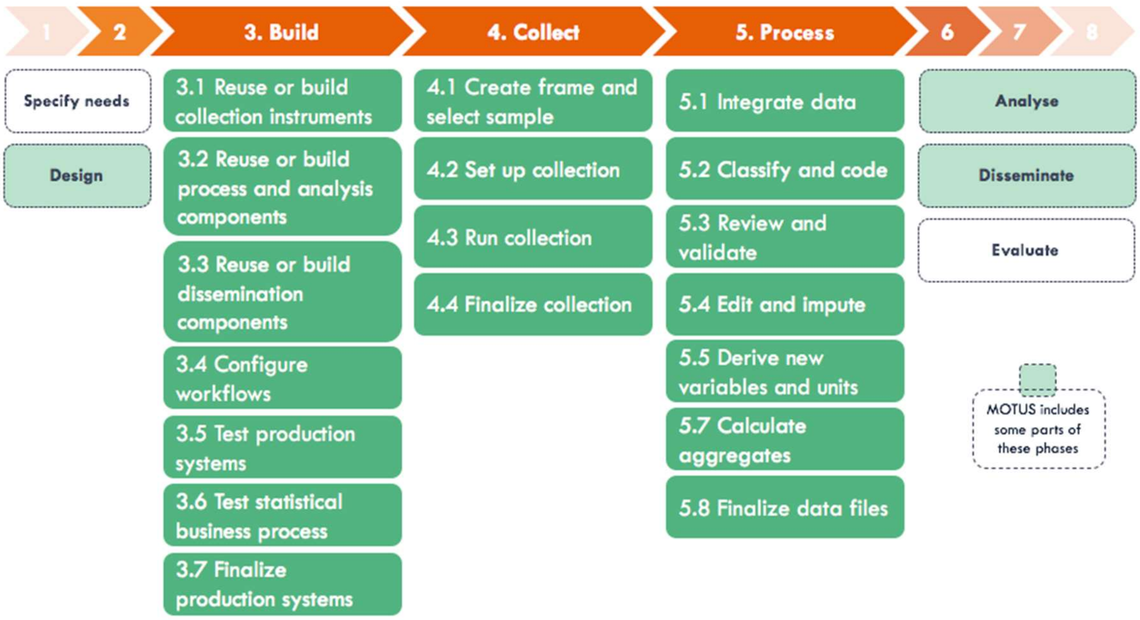
In MOTUS, several micro-services are under development that will render the capacity to collect data passively, processing and contextualizing this information and provide it back to the respondent for validation and optimization of the registration of activities.

4. Logical level

In the last few years, several conceptual frameworks like GSBPM, GSIM and CSPA have grown to maturity, making them very helpful tools to describe MOTUS' technical capacities.

The functional range of MOTUS mainly overspans stages 3 to 5 of the GSPBM; the build, collect, and processing phases of a research cycle (See figure 1 – green boxes). Below we present an overview of each of these phases. The ambition of VUB is to expand that range, and several new developments are under development relating to the design phase (GSPBM 2), the analysis phase (6), and the dissemination phase (7). Some should come available in a Bèta-version during the Essnet and possibly can be integrated into the planned pilot studies.

Figure 1: Mapping MOTUS on GSPBM.



- Phase 3 - Build phase

MOTUS' back-office is the place where a research cycle is set up. The so-called builders are the spine of MOTUS' back-office. Table 1 gives a detailed overview of them.

Table 1: Components and instruments of the MOTUS builder

Number	Builder	Components	Instruments
1	Device builder	Web app Mobile app API	
2	Survey builder	Create surveys Multiple question types Library of surveys	• Individual questionnaire • Household questionnaire • [Country specific questionnaires] • Context questionnaire – non travel • Context questionnaire – travel • [Specific context questionnaire that focuses on (and/or) the temporal, spatial, social, psychological dimensions

3	Diary builder	Create diaries
		Multiple time diary parameters
		Library of ACL
			• Activity Classification List (ACL)
			  ○ Number of levels
			  ○ Coding
			  ○ Selection and/or search functionality
			  ○ Tags
			• Time diary parameters
			  ○ Diary length
			  ○ Grain of precision
			  ○ Cycle
			  ○ Focus
			  ○ Learning period
			• Quality criteria
			  ○ Undefined time
			  ○ Number of activity
			  ○ Number of different activities
			  ○ Rounded hour activities
			  ○ Sleeping time
			  ○ Occurrence of eating, drinking
			  ○ Occurrence of travelling

4	Event builder	ESM Sensors Plugin or Microservice	• Input from geolocation plugin • ESM – Experience Sampling Method
5	Communication builder	E-mail SMS Notification Page	• Page • Email • Text • Notifications
6	Language builder	Default language Additional languages	• Languages • Translation of system buttons • Translation of research components • Translation of communications
7	Research builder	Customize respondent tasks Customize respondent events Customize communication	
8	Invitation builder	E-mail invitation Postal invitation Automatic registration Anonymous registration	
9	Dashboard builder	Response rate Completion rate Infographics	• Progress fieldwork • First results via an interface
10	Data builder	Download .csv R, SAV	• Download database • META-data
11	Quality builder	Quality controls META-data	• Para data/metrics • Status - progress • Logs • Communication overview • Quality determination

In development			
12	Computation builder		• Data availability via RStudio • MOTUS-Library with variables and statistical methods syntaxes
13	Visualization builder		• Data availability via RStudio • MOTUS-Library with visualization techniques • Possible programs are R Markdown and R Shiney

The builders are literally to be perceived as statistical services in the full extent of the CSPA definition of a statistical service:

- They are self-contained: They are building blocks, containers, allowing for a modular built-up of a (GSBPM) research cycle. Except maybe for the most basic ones, none of the builders determine the overall service, being MOTUS as a smart survey tool.

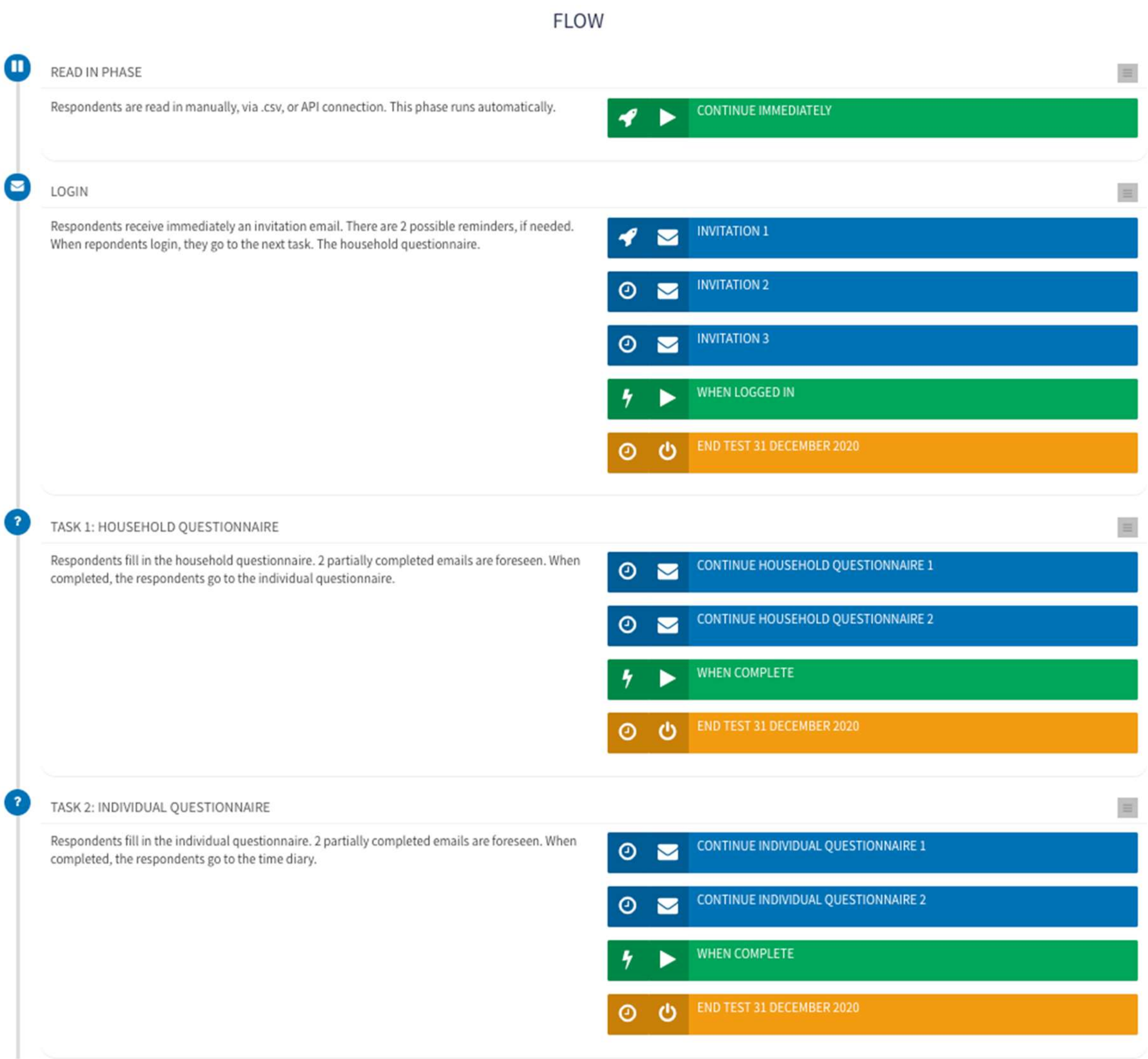
- They are reusable within statistical organizations: In MOTUS, production cycles are seen as a specific constellation of builders and the functionalities they deploy. Completely separated production pipelines, each having particular solutions for functionalities shared by many cycles, don't exist. From this point of view, differences between a LFS-like and a TUS-like data production are only marginal. Both make use of the device and the questionnaire builder, but the latter adds the diary builder. The other way around, LFS could be enriched with modular diary-based data by merely borrowing from TUS.

- They are reusable between statistical organizations: MOTUS as a data collection platform can be shared, either via a native installation, either via the container technology. The MOTUS containerization is part of a new project proposal. Like explained already, MOTUS also can be linked to external services or microservices. These microservices can also be either containerized or hosted by other organizations.

Using builders as the backbone in MOTUS' architecture is entirely in line with the CSPA philosophy and, consequently, vastly enhances reusability and shareability (between research strands and even countries). Therefore, to some degree, setting up a research flow in MOTUS is mainly a matter of recycling existing solutions, not inventing or developing new ones. Each existing component can be tested in development mode via www.motusresearch.io.

A research flow specifies which tasks the respondents have to fulfill to finalize his/her participation successfully. The setup of the research flow is the most critical of the built phase. A research flow combines every task, every communication, every action, and every event within the platform's business logic, implementing the builders' proper functionality. In figure 2 an extract of a particular research flow in the MOTUS back-office .

Figure 2: Extract of MOTUS research flow (back office).



Conceptually in research flows, 'states' like a questionnaire, a time diary, or a communication page can be distinguished from actions like communication (e-mails, notifications, warnings) and events (login, completion, event triggers). Most flows are linear, but event-based flows can be operationalized. Tasks and communications are triggered by specific events based on time or input via sensors and other data streams.

- **Phase 4 - Collect phase**

To launch a data collection, a sample frame must be determined and a sample of respondents drawn. This sample can be read into MOTUS via a .csv-file, by simply copy/paste the sample or automatically via a WordPress API plugin. In this process, it's possible to distinguish between required information (first name, last name, e-mail address, …), and auxiliary information (employment status, family situation, …).

Upon the inclusion of respondents into MOTUS, credentials are created. MOTUS includes parameters to define the username and password. Two-factor authentication may be activated.

All content can be translated in the MOTUS back-office (language builder), and a default language and secondary languages can be set. When a respondent's language preference is known, the respondent receives the research content in his or her language.

Once the research has been defined, also the follow-up to the research itself can be prepared. This includes meta-data, but also the cleaning actions that are being done during and after the data collection. The data collection process can be followed via a dashboard that is linked to an R-package running on the raw data at the server-side.

- **Phase 5 – Process data**

MOTUS outputs data frames through a MariaDB-database on the MOTUS-server. The raw data can be on personal, household, survey, time diary, sensor, or log level. meta-data contains information on the collection elements, the research flow, and the response and quality assessment.

All respondents are linked to a UUID and one or more research UUIDs. Every content in MOTUS related to respondent information, survey questions, or time diary input is linked to a code and a UUID. All externally processed data through MOTUS has a unique UUID, making microservices and external services possible.

During the data collection, MOTUS works with an R-package to examine the raw data. The MOTUS-package checks upon:

- Missing data
- Time diary data are controlled upon different criteria:
    - Length of unreported Time
    - Number of reported activities
    - Number of days reported
    - Check on reports on rounded Time
    - Occasion of eating
    - Occasion of sleeping
    - Disclosure of activity sequences
    - Imputation of sleep

Data can be imputed and flagged. New variables can be derived, and new units can be created. Via R data frames can also be turned into aggregated data. Also, a reorientation of the time-activity data file is possible.

The databases are downloadable from out the MOTUS back-office. Data can be downloaded in .Rdata, .csv or .sav format.

- **GSIM input and output**

GSIM describes the input and output over a GSBPM research cycle. Below an overview of the various statistical products and services (in and out) that MOTUS supports.

GSIM Input:

- Data structure (Content setup):
    - Questionnaires
    - Activity list
    - Context questions
    - Communication
    - Parameters to fieldwork
    - Parameters settings to download

- Dataset (Data file):
    - Administrative validation
    - Communication validation
    - Input validation-questionnaire
    - Input validation-diary
    - Process validation
    - Database validation
    - Security validation
    - Back-up validation
    - Performance validation
- Dashboard (live):
    - Respondent control
    - Researchers control

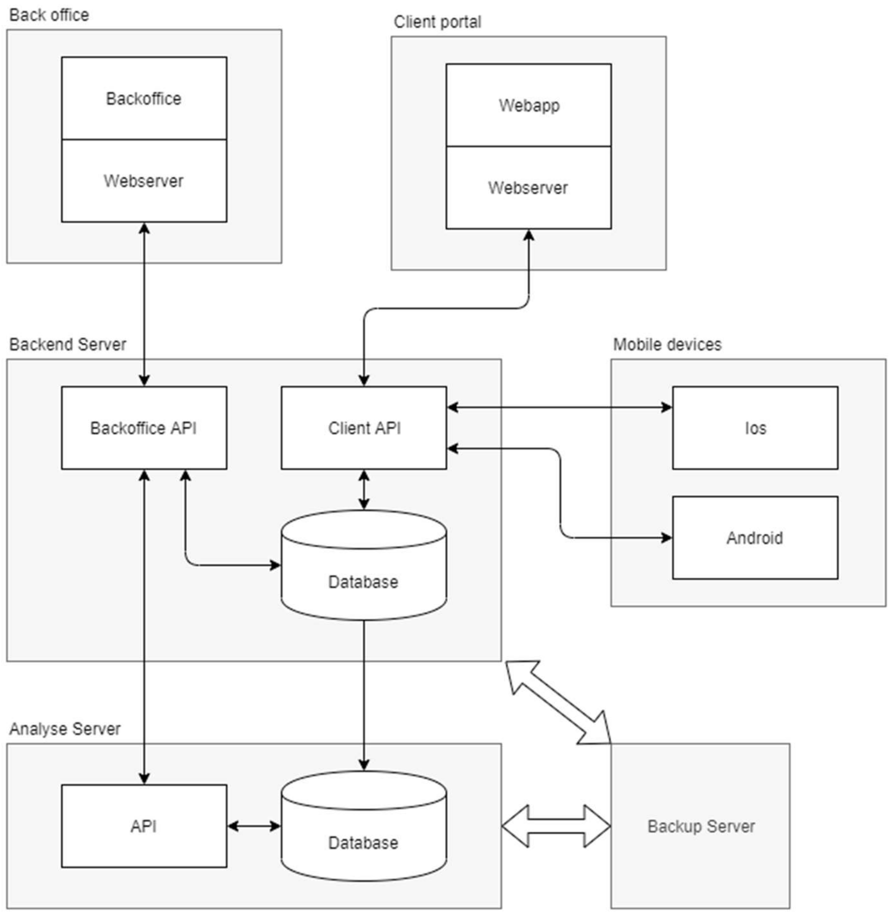
GSIM output:

- Progress report; in various levels of detail
- Response report; in various levels of detail
- Metadata report; in various levels of detail
- Database; in various levels of detail
- API-linked output to other sources and databases

5. Physical level

Figure 2 gives an overview of the core architecture of MOTUS. Below the figure, there's a description of the main components. Subsequently in table 2 a technical description of the front- and back-office is given, both client and server side.

Figure 3: Software architecture MOTUS Core



- Backend server: the backend server stands central in the MOTUS-software platform. It holds the database, the back-office API and the client API.
- Back-office: the back-office serves as the research environment where the researcher sets up a research and the fieldwork can be followed. The back-office runs in a browser.
- Analyse server: the analyse server holds a replicate of the database of the backend server and prepares the reports for the backend server, which at its part can be called by the back-office.

- Back-up server: the back-up server is a replicate for secure storing from the backend server and the analyse server.
- Client portal: the client portal holds the MOTUS-web application and an underlying webserver.
- Mobile devices: the mobile application is available for Android and iOS

There are three API's that arrange the entrance to the components:

- Backoffice API: both ways webserver back-office and analyse server
- Analyse server API: both ways database (to prepare reports) and back-office API to send over reports and other analytics.
- Client API: Receives the input from the web & mobile app and syncs the data on both applications. It could also function as a data harmonization tool.

MOTUS has been set up as a 3-tier architecture:

(1) Client tier with the GUI code
(2) Application tier with the business logic
(3) Data tier with the stored information

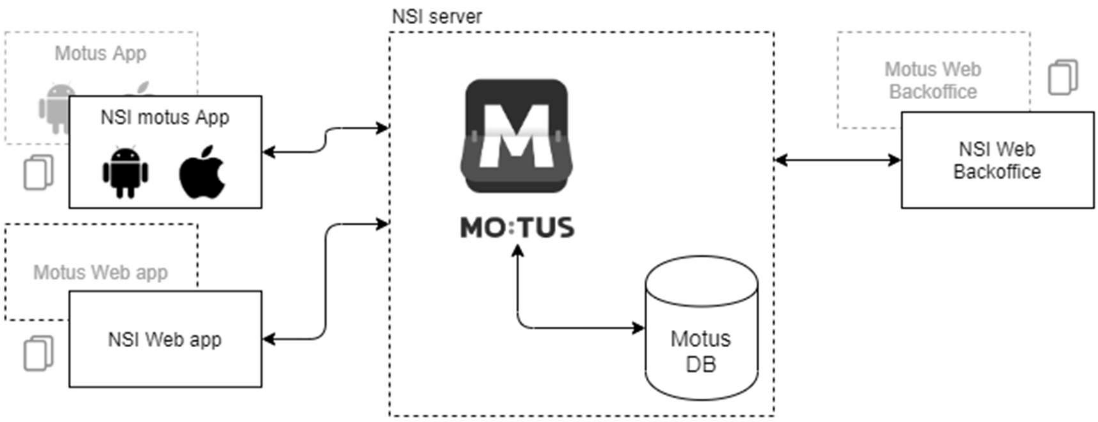
Table 2: Technical description of the front- and back-office.

MOTUS front office		
Respondent front-end web app: https://www.motusresearch.io/		
Respondent client side technology	Programming language	Javascript
	Framework	Angular 5
Respondent server side technology	Programming language	PHP 7.x
	Framework	Koseven
	Database	MariaDB – shared hosting
Respondent mobile front-end technology iOS: https://play.google.com/store/apps/details?id=be.byteworks.motus Android app: https://apps.apple.com/be/app/motus-zap-vub/id956934466		
App technology	Programming language	• Javascript • Objective C • Java
	Framework	• Angular 5 • Ionic 3
Endpoint server technology	Programming language	PHP 7.x
	Framework	Koseven
MOTUS back-office		
Researcher/admin back-office web app		
Client side technology	Programming language	•Javascript
	Framework	• Angular 5 • jQuery
Server side technology:	Programming language	•PHP 7.x
	Framework	Koseven
	Database	MariaDB

	Back-end server	• VPS • Linux: Debian 9 // Updated to CentOS (free version of Redhat) • Ten core CPU • 60GB RAM • Disk storage: 1600GB • Port/bandwidth: 1Gbit/s • DDoS protection
	Analyse server	• VPS • Linux: Debian 9 // Updated to CentOS (free version of Redhat) • Ten core CPU • 60GB RAM • Disk storage: 1600GB • Connection: 1Gbit/s • DDoS protection
	Back-up server	• 2x100GB back-up space • Internal network only

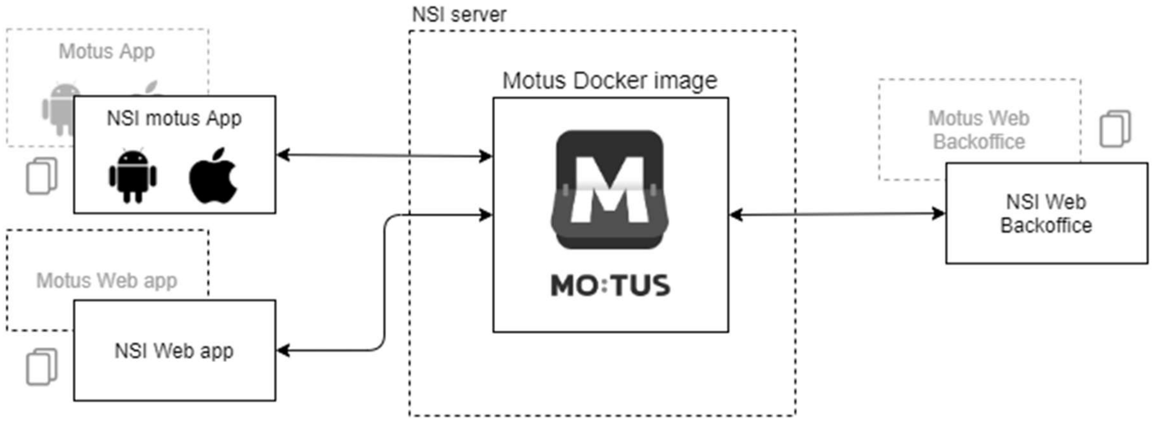
MOTUS' isn't confined to VUB or HBITS; its architecture can be encapsulated into external organizations' data collection architecture, like an NSI. Consequently, today, NSI's can use MOTUS as a service, a data collector, or a native installation.

Figure 4a: MOTUS as a service.



However, in the coming years VUB foresees developing MOTUS as a set of containers:

Figure 4b: MOTUS as a service.



To that end, the MOTUS platform will be transferred from a Koseven/Kohana platform to a Laravel platform, and also the microservices will be written in Laravel.

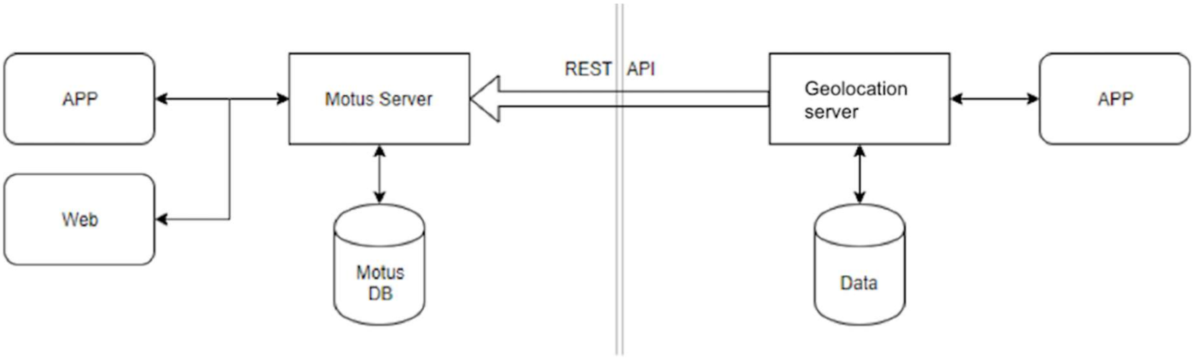
The approach using microservices makes integrating future developments relatively straightforward. It's entirely in line with the reasoning behind CSPA, and it highly increases shareability.

The inclusion of a geolocation microservice is an excellent example of the benefits of using microservices to enhances MOTUS' capacity as a data collection platform. We elaborate this example in some detail in annex.

ANNEX 1: Elaboration of the integration of a geolocation micro-service in MOTUS.

In figure 5 the relation between geolocation environment and the MOTUS core is given.

Figure 5: A simplified architectural of the relation between a geolocation environment and the MOTUS CORE environment.



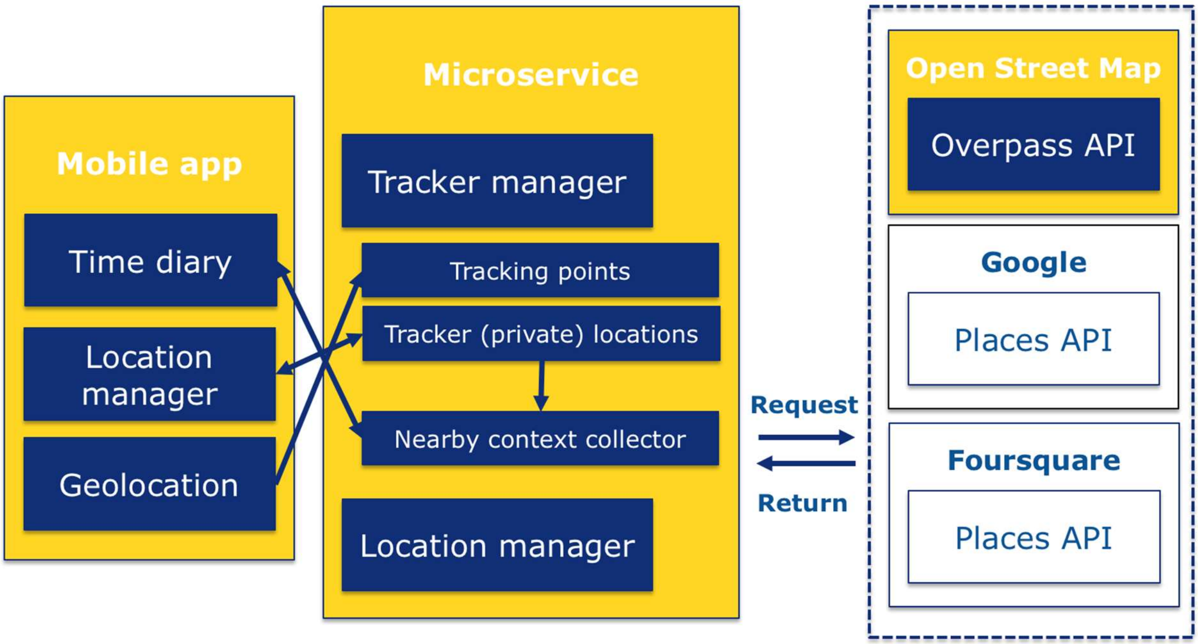
The main asset of the Microservice strategy is that:

- It does not influence the operations of the CORE environment
- The input does not have to be error-free
- The input can undergo various quality controls
- The output can be parsed
- It is flexible to change from technology or updates within a technology

Since there is a clear link between the input-output and the respondent, the information can be visualized to the respondent on his device's screen. At that moment, the respondent can verify/accept/change the information, or additional actions can be run by MOTUS asking for more information (e.g., on the context of the activity). Only when the respondent accepts the data, the tentative information becomes committed data.

Tracking points can be enriched via a nearby context collector connecting to external databases, like Google and Foursquare, but preferable the open-source Open Street Map database.

Figure 6: Enrichment of tracking point by external API's

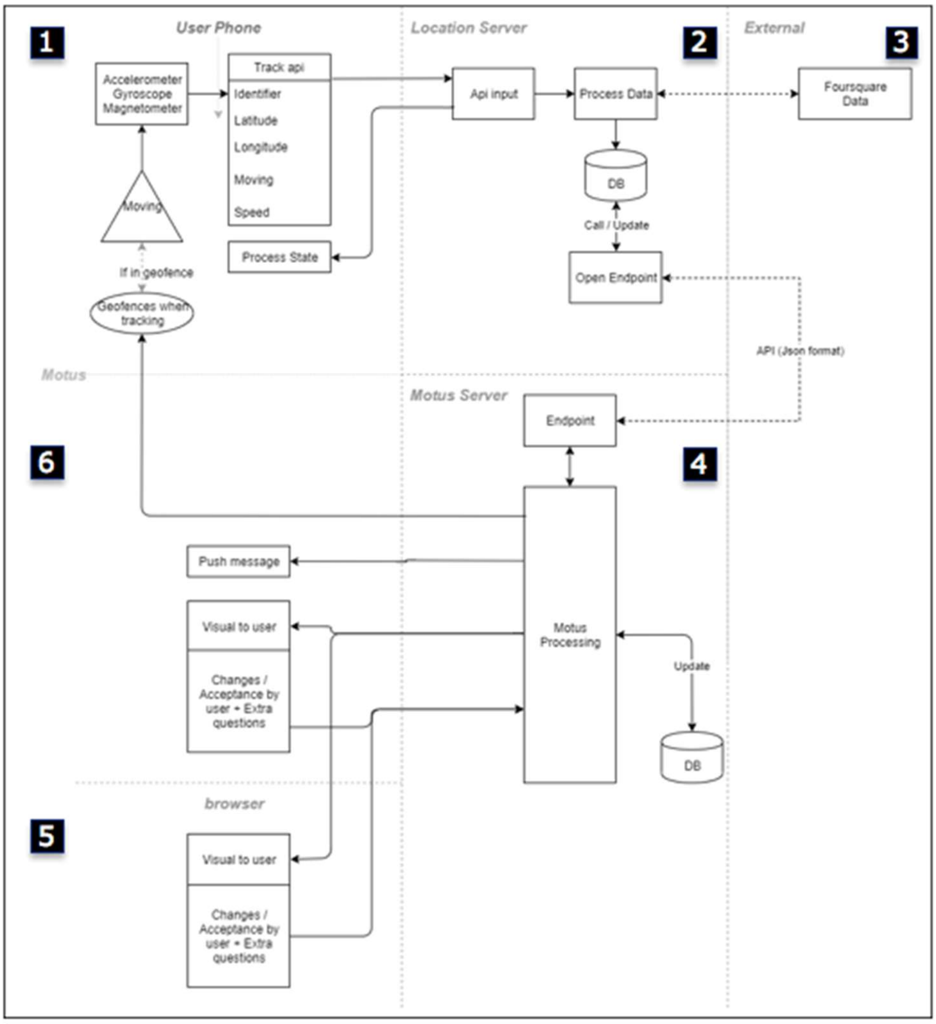


The data flow diagram in the above figure shows the outputs and the inputs of the architecture. Not only the geolocation plugin's functionality as such is presented, but also the interaction (inputs and outputs) with the MOTUS-core and the respondent are shown

The data flow counts in total six steps:

- Tracking geolocation
- Modify signals via an algorithm
- Contextualize geolocation data
- Visualize geolocation data to the respondent
- Accept/change geolocation data by respondent
- Use geolocation data as an event-trigger

Figure 7: data flow geolocation plugin and core.



The raw data collected through the geolocation plugin need interpretation to become insightful. These data are a collection of objects:

- Time stamp
- Latitude
- Longitude
- Accuracy
- Speed
- Mode of transport: still, walking, running, bicycle, vehicle
- Confidence interval

Figure 8 : Modification signals via Algorithm.

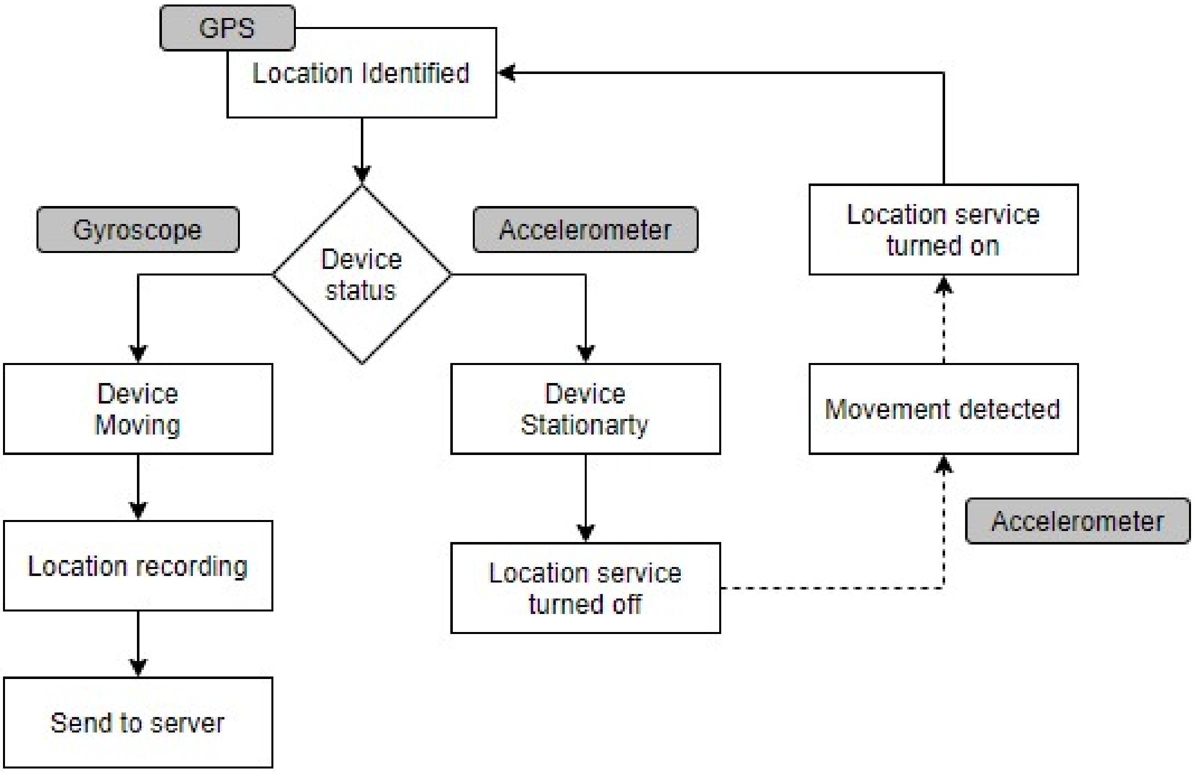
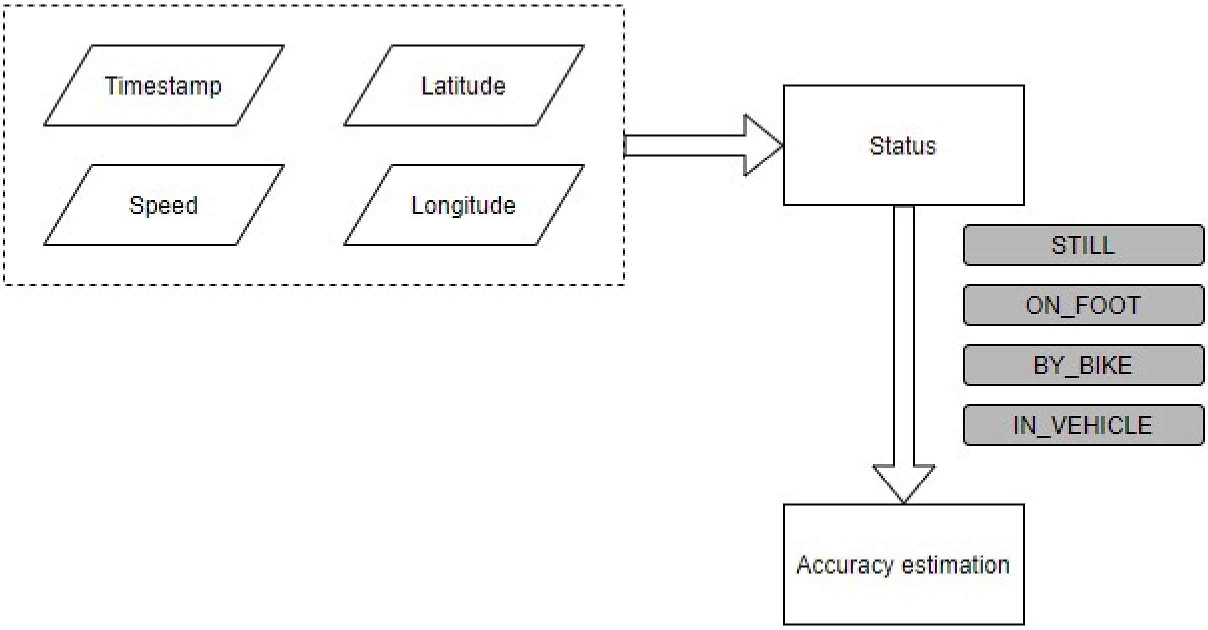


Figure 1: Trip information



The plugin hosts its own (native & flexible) HTTP layer and sends the data to the geolocation server.

Figure 10: Database structure geolocation server