ESSnet Smart Surveys

Grant Agreement Number: 899365 - 2019-DE-SmartStat

Link to our CROS website

Workpackage 3

Development of a conceptual framework, reference architecture and technical specifications for the European platform for Trusted Smart Surveys

Deliverable 3.3
Report on the Enhanced Framework

Version 1.0, 30-03-2022

Prepared by:
Mauro Bruno, Massimo De Cubellis , Fabrizio De Fausti, Claudia De Vitiis, Francesca Inglese,
Giuseppina Ruocco, Monica Scannapieco (ISTAT, Italy)
Nils Meise (DESTATIS, Germany)
Jacek Maślankowski (GUS, Poland)
Joeri van Etten (CBS, The Netherlands)

Work package Leader:
Claudia De Vitiis (ISTAT, Italy)
devitiis@istat.it; +390647737401

Outline

1. Executive summary

This deliverable focuses on the conceptual framework of the European Platform for Smart Surveys, main goal of the activity of Work-package 3 (WP3). The last part of the project aimed at producing an improved framework revised after the proof-of-concepts and in light of the results provided by the Work-package 2 (WP2).

In deliverable 3.1 a preliminary framework was described, where all the aspects mentioned in the objectives of the WP were addressed at a very high level following a top-down approach. Now, after the end of the second year of activity and considering the results of the Proof-of-Concept, the framework takes a step towards the definition of the platform specifications and requirements.

The overall objective of WP3 is to provide a conceptual framework, reference architecture and technical specifications for a European platform for Trusted Smart Surveys (TSSu), focusing in particular on: (i) the reuse of existing smart survey tools and applications; (ii) appropriate methods for handling missing data, identifying and correcting dubious input data; (i) providing strong guarantees in terms of security, privacy preservation, confidentiality protection and auditability, through a composition of appropriate technological solutions; (iv) exploiting the interaction loop with the respondent and his/her smart device(s).

In the enhanced framework, the general architecture of the TSSu platform is described in terms of business and application layer, aligned with existing frameworks and official statistical standards, such as the GSBPM (Generic Statistical Business Process Model and GSIM (Generic Statistical Information Model). The business layer follows the process and sub-process of GSBPM to identify and standardize the core activities performed by the National Statistical Institutes (NSIs) for smart data collection and processing (What).

The definition of the main process steps fosters the design of the application layer, providing an overview of the main components to be developed for the definition of a common infrastructure (How). These components are the building blocks of the general platform and they are deepened from the different perspectives: Methodology, Privacy and Metadata.

In particular, the methodological aspects of a smart survey deepened in this deliverable are those more tightly related to the collection and exploiting of sensor data and new data sources. Therefore, all the methodological aspects that can be treated analogously to a traditional survey are here out of scope. Moreover, the focus here is mainly on the data collection and data processing phases. The other phases of the survey, in fact, such as the sampling design, the integration of data collected with different modes (if a mixed-mode data collection is adopted) and the estimation phases are not addressed as, at least in this phase, the platform aims to develop and support only the smart features.

It is useful to underline that the platform delineated does not foresee the objective to develop the application tools for the data collection; the tools can be implemented by NSIs autonomously or can be selected from a list of existing tools, such as those reviewed by the Task Force Innovative Tools and Sources. So, the platform could utilize existing tools, fostering their integration in the survey pipeline.

Moreover, the use of a tool and the realization of a survey requires experimentations, through a test phase. The platform can assist the NSIs to project and conduct these tests.

The enhanced framework tries to cover all the objectives of WP3, those addressed explicitly in the proofs-of-concept and those addressed by WP2 in the pilots, concerning mainly the data collection aspects.

In a more practical perspective, the platform outlined by WP3 follows the idea of providing to European NSI statisticians a set of tools for the execution of functionalities to carry out a smart survey, in a modular and incremental approach, starting from what could be implemented in practice. These functions could be deployed locally or centrally. These functionalities should address primarily the strictly "smart" aspects of the survey, such as the sensor data collection, the monitoring system related to data collection and to sensor data quality, machine learning algorithms for processing sensor data, the metadata management for guaranteeing the auditability of the process. These functionalities focus also on privacy and security issues, exploiting privacy protection techniques in the process.

In the discussion with WP2 members, which have a more practical approach than WP3 perspective, a common agreement was that initially the platform could be a repository of software solutions, developed with the aim of being general and domain agnostic, but aware that this condition could limit the possibility to deal with specific issues.

With reference to the *smart survey features* characterizing a smart survey in terms of data sources, considered in WP2 taxonomy, the framework for the platform focuses on those features more connected with the use of a smartphone application for exploiting sensors that are available in the device (giving rise to an "Internal smart survey"), and for the implementation of traditional data collection instruments (questionnaires and diaries). The modularity of the platform allows to extend to other sources of data, such as external data (other sensors nearby and connected with the device) either collected by external sensors or available in existing sources.

The "Enhanced framework at work" aims to provide some examples of realization and to use the components defined in the application layer of the architecture: a couple of user stories are illustrated to give concreteness to the general description of the platform.

The structure of the document is the following: the architecture of the framework is described in Chapter 2; the functionalities of the building blocks are addressed in Chapter 3, where Paragraph 3.1 focuses on methodological aspects, Paragraph 3.2 on the privacy issues and Paragraph 3.3 on the metadata management; Chapter 4 describes the technical requirements of the components; Chapter 5 finally describes the possible functioning of the platform, through the description of some use cases. The Annex, reporting some operational scenarios, complements the description of the enhanced framework.

2. TSSu Architectural Framework[1]

The design of the enhanced framework aims at highlighting the relationships between many different aspects, related to three main dimensions:

- **Architectural**: concerning the design and development of software solutions for smart data collection and processing
- **Methodological**: regarding the available privacy preserving techniques, and the different methods for smart data gathering, validation and processing (data collection strategy, design of the user interface, edit checks and data quality, machine learning models).
- **Technological**: dealing with the design of the technical infrastructure according to the privacy preserving requirements, as well as the interaction between the platform components.

Starting from the alignment with existing frameworks, the architectural PoC has analysed these issues, modelling the main process steps for smart data processing according to GSBPM and GSIM standards, as well as BREAL business functions.

The following analysis is an enhancement of the results of the architectural PoC, based on the insights provided by work package (WP2) pilot surveys. These experiences have highlighted several issues, related to the reduction of the respondent burden in social surveys through smart data acquisition and processing. More in detail, the business layer for a generic TSSu modelled in the architectural PoC has resulted from:

- WP2 description of Pilots
- WP2 deliverables to inform WP3
- Report of the legal-ethical Working Group
- Deliverable 3.1 Report on the Preliminary Framework for methodology and reference frameworks.

An overview of smart data pipeline, from a National Statistical Institute's perspective is essential to assess:

- The impact of smart data sources on GSBPM phases and process steps
- The process steps which can be standardized and executed in a common infrastructure, to identify shareable and reusable application components
- The tasks that can be executed in the TSSu platform or in the respondent's device, as well as the tasks performed in-house by the National Statistical Institutes
- The best strategy to harmonize specific national needs and the peculiarities of the stakeholders engaged in a TSSu process.

2.1 Business Layer

The main goal of modelling the Business layer of a TSSu is to identify and standardize the core activities and behaviors performed by NSI_S for smart data collection and processing (**What**). The adoption of a top-down approach aims at highlighting the interdependence between the key dimensions listed above. In addition, the definition of the main process steps fosters the design of the application layer, to provide an overview of the software solutions (**How**) to develop for a common infrastructure.

The upper part of **Figure 2.1** shows the relationship between the **GSBPM** phases and **BREAL** business functions concerning smart data processing (e.g., Acquisition and Recording, Data Wrangling, Data Representation). While "Metadata management" and "Privacy Preserving techniques and management" are overarching business functions, the sub-processes performed in the different GSBPM phases for smart data have been

[1] Mauro Bruno and Giuseppina Ruocco (ISTAT)

considered as a specialization of the related business functions. As a starting point for the design of the application layer, the following analysis reports, for each GSBPM phase, an overview of the main tasks to execute.

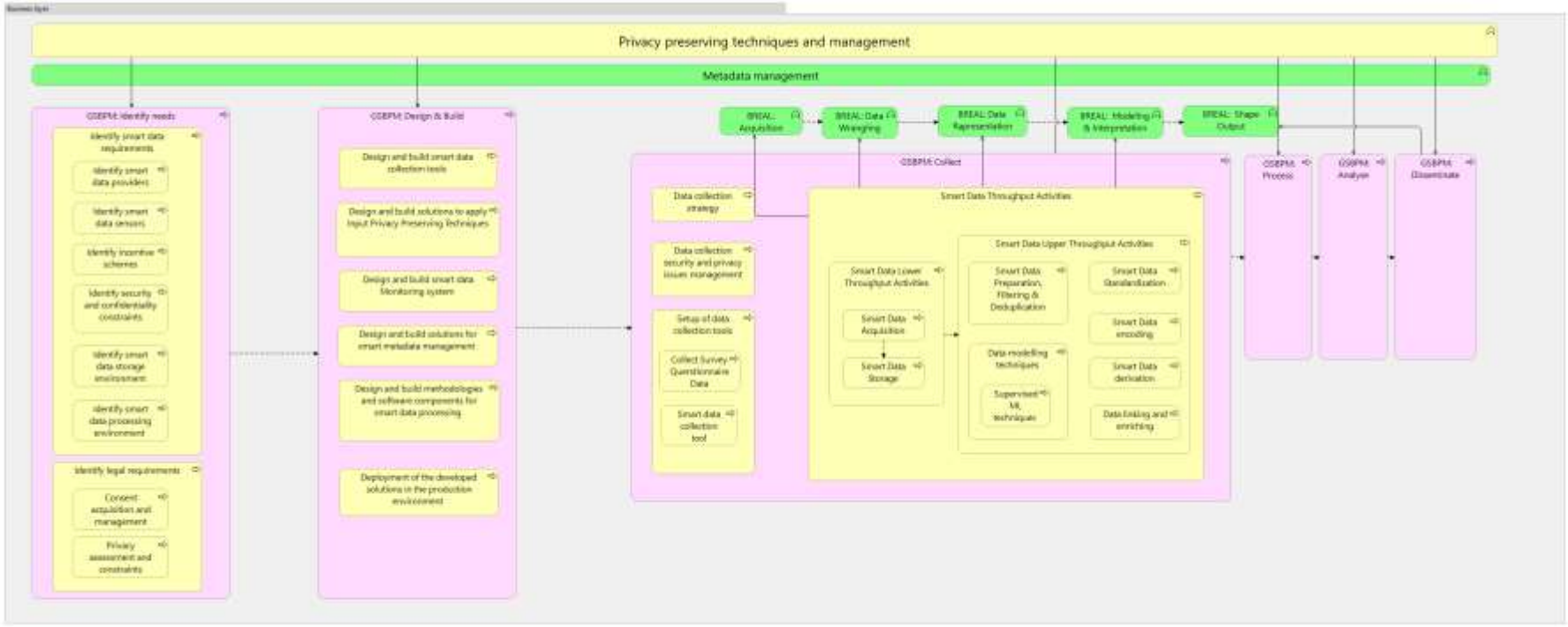
GSBPM phase "Identify needs"

In order to design a **smart data strategy**, in the first GSBPM phase the following elements are specified and grouped in the sub-process **Identify Smart Data requirements**:

- **Smart Data provider**: respondents and/or third parties. The type of data provider has a huge impact on the technical and legal solutions to develop for smart data acquisition. In case of smart data provided by third parties, an initial assessment of data confidentiality may lead to the definition of protocols for smart data acquisition. Following the "push computation out" approach, these protocols may involve the adoption of input privacy preserving techniques, as well as the calculation of quality indicators to assess the quality of acquired data.
- **Smart Data sensors**, such as external sensors, or sensors embedded in mobile devices (e.g., pictures, accelerometer, GPS) generating active or passive data gathering (primary or secondary data collection)
- **Smart Data storage environment**: storage environment for different types of smart data sources (e.g.: relational, NoSQL, Json)
- **Smart Data processing environment**, specifying whether data are processed in-device, in-house, in the TSSu platform, or on third parties' premises. Both, data storage and processing environments will be analysed more in detail in the section describing the operational scenarios.
- **Smart Data incentive schemes**, in order to motivate and reward the respondents.

The sub-process **Identify Legal Requirements** refers to the assessment of the legal implications (consent management and legal/ethical analysis) and privacy issues related to the type of smart data provider, sensor, storage and processing environment.

Figure 2.1: TSSu Business Layer



GSBPM phase "Design & Build"

In the proposed architecture, the GSBPM phases "Design" and "Build", have been collapsed due to their close connection.

The "Design & Build" phase concerns the development or reuse of software components to implement the data collection strategy. In addition, this phase allows to test the implemented solutions and configure workflows, thus facilitating the migration from traditional pipelines to a new process combining smart and traditional data sources. Concerning smart data, the following activities are particularly relevant in this phase:

- **Design and build smart data collection tools:** depending on the type of smart data acquisition active/passive, such phase involves the implementation of the data collection tool (web application, mobile device app, etc.) according to the following specifications **suggested by WP2:**
    1. In-device storage and processing
    2. In-device sensors
    3. Link to external sensor systems
    4. Link to relevant public online data
    5. Private respondent's data donation
    6. Respondent's consent to link private data held by the NSI.

- **Design and build solutions to apply Privacy Preserving Techniques (PPT)**. In order to meet the privacy requirements related to a specific domain, Input privacy techniques should be implemented and tested. Several tests are essential to detail the requirements of the implementation of some Input privacy techniques during smart data acquisition, in terms of infrastructure and data collection tools. The results of the PoCdedicated to this topic (see chapter 2. Application of Privacy Enhancing Technologies to the TSSu platform - Deliverable 3.2 Report on the Proof-of-Concept), as well as the ongoing project promoted by UNECE, provide useful information in terms of software solutions and hardware requirements.
- **Design and build smart data Monitoring system**: the monitoring system provides a set of functionalities that allow to reduce missing and measurement errors and manage survey logistics. Monitoring dashboards should be implemented to monitor respondents' participation and quality of collected data, in order to identify corrective actions. According to WP2 experience, capturing *paradata* and *contextual data* during data acquisition provides auxiliary information for the quality assessment of collected data.
  Further, as explained in WP2 deliverable (Schouten B. et al.), the logistics level should include the following aspects:
    - ***Recruitment****: Many of the ESS surveys have an interviewers-assisted data collection such as doing the starting questionnaire, recruiting and assisting respondents in a diary, motivating respondents during data collection, and/or picking up closing questionnaires.*
    - ***Assistance****: Respondents may be assisted in starting and using smart survey tools passively through a helpdesk and website or actively through interviewers and or technical experts.*
    - ***Human-in the-loop machine learning****: models seldom reach 100% accuracy. Certain population subgroups or certain survey statistics may require manual inspection. In*

ESSnet Smart Surveys where feedback of statistics to respondents is deemed important, such human-in-the-loop processes may even occur during data collection.

- **Design and build solutions for smart metadata management:** to document process steps, data structures, methods and track data transformations due to process execution. In addition, a set of quality indicators and metrics should be produced to assess the results of smart data processing.
- **Design and build software components for smart data processing**: related to the methods for processing sensor data, generally unstructured (e.g., texts, images, signals). Within this context, Machine Learning (ML) techniques are particularly relevant to extract statistical information from a wide range of smart data. ML techniques can be applied in different phases of the TSSu process from data integration, classify and coding, review and validate, to edit and imputation (ESSNet Smart Survey - Deliverable 3.1 Report on the Preliminary Framework - paragraph "6.3 GSBPM Process Phase for Sensor Data and ML Algorithms").
- **Deliver the developed solutions in the production environment**. Starting from the feedbacks provided by WP2 about the challenges faced during the field tests, the TSSu platform should be able to interact with a wide range of production systems (Third parties, NSIs, mobile devices, etc.).

GSBPM phase "Collect"

The design of this GSBPM phase focuses on the combination of the activities performed in traditional surveys, and the tasks directly related to smart data acquisition, for instance, the use of supervised machine learning techniques for online training or for model assessment, as suggested by WP2. Some activities performed at national level and included in this phase, such as the sample selection or arranging an agreement with third parties have not been considered in the design of the platform pipeline. More in detail, the tasks analysed for this phase include:

- **Data collection strategy**, concerning the specification of the contact modes (smart and traditional), reminder initiatives, incentive schemes, logistics management and recruitment materials. The definition of a collection strategy for a generic TSSu should be compliant with national peculiarities.
- **Data collection security and privacy issues management**: a set of activities related to the consent management and the compliance of data collection tools with data security protocols.
- **Data collection tools**: instance of the data collection tools implemented and tested in the build phase.

Smart data throughput activities

In the proposed model, GSBPM "Collect" phase is tightly connected to the following BREAL business functions: *Acquisition and Recording*, *Data Wrangling*, *Data Representation* and *Modelling and Interpretation*. More in detail, the sub-process **Smart Data Throughput Activities** includes the tasks needed to transform collected smart data in statistical data. Throughput activities may vary in each survey, according to the type of sensor, the type of data provider, in-app, or in-house data processing and can be grouped as follows:

- **Lower throughput activities:** concerning mainly smart data acquisition and recording
- **Upper throughput activities**:
  - *Data preparation, reduction, filtering & deduplication*, a set of pre-processing steps to select the relevant information and create input data for the following tasks

- *Data standardisation* to convert data to a target format. This sub-step, as well as the following one is particularly relevant in case of passive data acquisition
- *Data derivation* to transform unstructured information to structured data. While the previous sub-step refers to data format, data derivation refers to the transformation of data content (e.g., creation of new variables), through the application of some rules and/or algorithms
- *Data encoding* to transform categorical data in binary or numeric format
- *Data modelling techniques*, grouping all the methods to extract statistical information from smart data, such as machine learning techniques
- *Data linking and enriching*, to integrate data collected through questionnaires and statistical information derived from smart data.

2.2 Application Layer: main building blocks

The design of the business layer is the first stage to define a preliminary set of **building blocks** needed to run the process steps of a general TSSu. These building blocks will be described in terms of functionalities that could be offered by the TSSu platform and will allow to make some assumptions in relation to the service deployment and sharing.

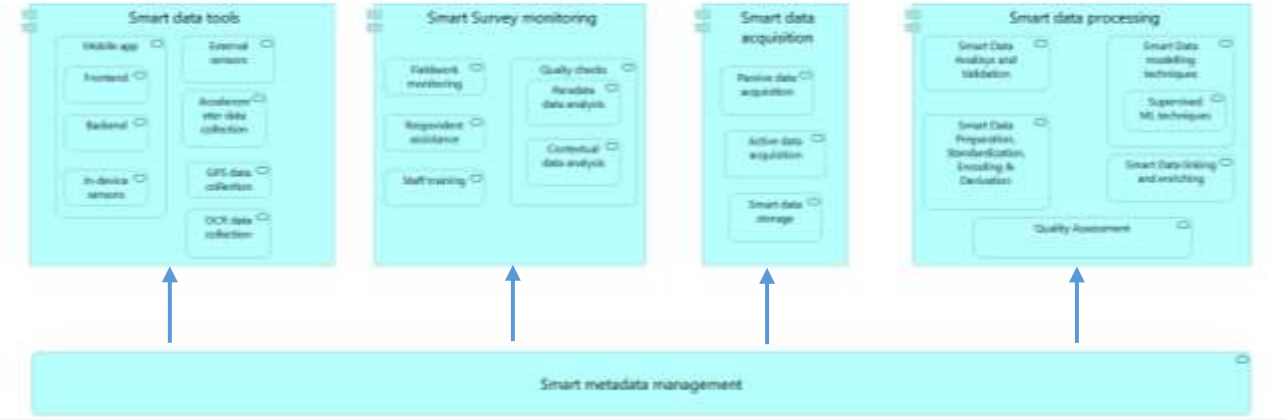
Regarding the quality, security and privacy issues, an analysis of the impact of these dimensions on the building blocks will help to identify the requirements of the technical infrastructure and software solutions. While the application of data security and Input privacy preserving techniques relate mainly to the technical layer, due to the protection of data *by design* across all application components, data quality assessment is a function that can be embedded in each software tool dealing with data processing.

According to the lessons learned in the different WP3 PoCs and in WP2 experiences, the process pipeline of a TSSu results from the combination of several tasks executed by modular components. The design of data collection and smart data processing building blocks should be driven by:

- The type of data source (private or public data)
- The type of data provider (Respondent, Third parties, public authorities)
- The type of sensors used for data gathering (in-device, or external sensors)
- The storage and processing environment (In-app/NSI/TSSu Platform /Third parties).

In order to design a general pipeline for a TSSu and foster the combination of smart and traditional data sources, the following figure reports an overview of the proposed building blocks and the connections with the metadata building block.

Figure 2.2: TSSu Building Blocks



The following table describes the initial set of building blocks resulting from the analysis of GSBPM phases modelled in the business layer. The list of the main components is based on WP2 pilot surveys, to foster the reuse of available smart survey tools and implemented solutions.

Table 2.1: Preliminary list of TSSu platform building blocks

Building Block	Main components	Description
Smart data tools		
- Mobile app - External sensors	- Mobile app frontend (user experience and incentives) - Mobile app backend - In-device sensors - OCR data collection - Accelerometer data collection - GPS data collection	This building block provides a set of functionalities to facilitate the design and the implementation of tools for collecting smart data. Data collection tools can include preliminary validation of entered data through soft and hard checks of plausibility and notifications of missing data. A relevant subset of functionalities offered by a specific service concerns the creation of incentives to foster the respondents' collaboration.
Smart survey monitoring		
Fieldwork monitoring	- Monitoring dashboard	Continuous monitoring of fieldwork activities and preliminary quality checks on collected data (e.g., response rates)

Building Block	Main components	Description
Respondent assistance	- Contact centre - Helpdesk website - Training documentation	Assistance provided to the respondents to facilitate the use of smart tools passively through a helpdesk and website or actively through interviewers and or technical experts
Staff training	- Training documentation - Training dashboard - Training classes	Training for the staff involved in interviewers-assisted data collection (e.g., starting questionnaire, recruiting and assisting respondents in a diary, motivating respondents during data collection)
Quality checks Indicators Paradata	- Soft and hard checks - Indicators to monitor participation, data integrity, etc. - Paradata on sensor, on device, on data transmission, etc.	The main goal of this building block is to monitor smart data acquisition and prevent representation and measurement errors: by enriching the traditional indicators with the paradata analysis to check the fieldwork (technical issues, respondents' behaviour); by adopting rule based checks for the early detection of insufficient data quality delivered by the participants. The results of the quality checks, performed to assess the accuracy of collected data, are stored in the **metadata building block**
Smart data acquisition		
Active data acquisition Passive data acquisition Smart data storage	- Structured or Unstructured data - NoSQL, Relational, Graph - Text files (CSV, XML, JSON)	This building block includes three main components for managing smart data structures and storage. The first one allows gathering in real-time data actively sent by the respondent (**Active acquisition**). The second one realizes data acquisition from smart devices or third parties' platforms (**Passive acquisition**). The third component provides data storage capacities (relational DB, noSQL DB) and optimizes data recording and data access performances.

Building Block	Main components	Description
Smart data processing		
Smart data analysis and validation Smart Data Preparation, Standardization, Encoding & derivation Smart Data modelling techniques Smart Data linking and enriching Quality assessment	- ML for OCR and Natural Language Processing - ML for Sensor data Processing - GPS data Processing	This building block provides a set of methods to: - detect and correct smart data errors, such as outliers and missing data - transform raw smart data (signals, images, texts) in statistical output. The components performing this tasks need to be specialized according to the smart data source and sensor -provide quality indicators to assess the output of each process step and measure the accuracy of ML models. The results of the quality assessment are stored in the **metadata building block**
Smart metadata management		
Smart metadata	- Centralized repository of metadata produced and used by other building blocks (**Metadata Repository**)	This building block realizes the overarching BREAL business function "Metadata management" and allows to acquire and visualize the process metadata used or produced by each building block and stored in the Metadata repository. In addition, within this group, ad-hoc functionalities may allow authorised survey staff to: - check or modify the information reported in each subset of the Metadata repository, and track metadata changes - provide an overview of the metadata captured throughout the different steps for process tracking and auditability

References – Chapter 2

Aracri, Bruno et al.: Task 3.1.3 - Integration in existing architectural framework auditability - Deliverable 3.1 Report on the Preliminary Framework, available from: https://ec.europa.eu/eurostat/cros/content/wp-3-conceptual-framework-european-platform_en

Aracri, Bruno et al.: Deliverable 3.2 Report on the Proof-of-Concept- Final version, 31-01-2022

ESS Enterprise Architecture Reference Framework (EARF), available from: https://ec.europa.eu/eurostat/cros/content/ess-enterprise-architecture-reference-framework_en

Generic Statistical Business Process Model (GSBPM) v. 5.1. January (2019). Available from: https://statswiki.unece.org/display/GSBPM/GSBPM+v5.1

Generic Statistical Information Model (GSIM) v. 1.2 March (2021). Available from: https://statswiki.unece.org/display/gsim/GSIM+v1.2+documents

Scannapieco M., Bogdanovits F., Gallois F.; Fischer, Paulussen R., Quaresma S. et al.: BREAL. Big Data Reference Architecture and Layers. Business layer. ESSnet on Big Data II, Work Package F, Deliverable F1. (2018-2021). Available from: https://ec.europa.eu/eurostat/cros/sites/crosportal/files/WPF_Deliverable_F1_BREAL_Big_Data_REference_Architecture_and_Layers_v.03012020.pdf

Schouten B. et al: " Deliverable 2.8: Functional and technical descriptions of tools - Consumption (WP2.1)" - Version 1.0, 21-12-2021

UNECE project Input Privacy-preserving Techniques 2021. Available from: https://statswiki.unece.org/display/hlgbas/Modernisation+Projects

3. Exploring TSSu Building blocks

The following sections describe the set of building blocks resulting from the analysis of GSBPM phases and modelled in the business layer from the main perspectives mentioned above: methodology, privacy preserving and metadata management.

3.1 Methodology

The methodological aspects of a smart survey deepened in this deliverable are those more tightly related to the collection and exploiting of sensor data. Therefore, all the methodological aspects that can be treated analogously to a traditional survey are not considered here as out of scope. Moreover, the focus is mainly on the data collection and data processing phases. The other phases of the survey, in fact, such as the sampling design, the integration of data collected with different modes (if a mixed-mode data collection is adopted) and the estimation phases are not addressed as, in this phase at least, the platform attempts to develop only the smart features.

The smart survey methodology has been addressed in deliverable 3.1. in chapter 1. The main issues addressed in the report on the preliminary framework are related to: (i) sensor data from a variety of devices that are not standardized in structure, format or availability; (ii) innovative ways of handling sensor data (machine learning algorithms); (iii) sources of error in TSSu and error management; (i) GSBPM phases involved in TSSu; (iv) automation of smart surveys and needed staff profiles. Moreover, a preliminary methodological framework for TSSu was outlined taking GSBPM as a reference statistical model to facilitate connections with the architectural system, in particular with data collection and data processing components of the platform.

In this deliverable the methodological issues are addressed with a more practical view, referring to the building blocks defined in the application layer of the architectural framework and to the topics mainly addressed in the discussions with WP2 members, especially in the Working Group on methodological issues. Data quality and machine learning in particular were of big concern due to the new methods and technologies and they was hands-on problems.

The structure of this chapter on methodology follows the dimensions of the methodological level used by the Work-package 2 in their deliverables, focusing on: Data collection strategy (3.1.1), User interface of frontend (3.1.2), Data quality: framework and building blocks (3.1.3) and Machine Learning potentiality in Smart Surveys (3.1.4).

3.1.1 Data collection strategy[2]

The data collection strategy includes all smart (and non-smart) data collection tools, their design and the logistics involved. In particular, the building blocks *Smart data tools* and *Smart survey monitoring* in terms of *Respondent assistance* and *Staff training*. This concerns the use of contact modes, contact and reminder strategies, incentive strategies, recruitment materials, use of “non-smart” modes. The choices made here have an impact on the other building blocks (see previous part for complete list) in

[2] Nils Meise (Destatis)

regard of required features and needed logistics. The table below presents a brief description of two building blocks related to data collection strategy: smart data tools and smart survey monitoring.

Table 3.1 - Overview of data collection strategy

Building Block		Description
Smart data tools		
Mobile app	Data collection strategy	The mobile app functions as a hub for the data collection strategy. It allows to collect data via the app (e.g. questionnaire), governs access to internal sensors, and is an optional gateway for external sensors and the data delivery/ synchronisation process. Data validation and checks can be a part of the app and involve the respondent via the user interface (see below). Due to its integration into a smart device it also benefits from system services like reminders.
	User Interface	The mobile app presents the user interface to the respondents and is their main interface to the survey and the survey agency. It can be part of the data collection process, if for example questionnaire are a part of the survey (see above). The user interface guides the respondents through the survey and provides meaningful feedback on their progress. Incentive strategies that aim at the phase during the survey utilize the user interface. Either as a hub for external devices that are provided by the survey agency or by substantial changes to the survey expierence with gamification features.
External Sensors	Data collection strategy	External sensors allow data collection beyond the confinement of a survey or a single smart device, if used in conjunction with a mobile app. Alternatively, they could be used without a companion app and provided by a survey agency (e.g. activity trackers) for comparable results. Due to the lessons learned within the work packages, it is not advised to solely rely on sensors readings. Third party apps bundled with external devices can offer the benefit of not having to develop mobile apps. They are also intransparent in regard to data collection, validation and offer no integration of additional incentive strategies.

Building Block		Description
Smart Survey Monitoring		
Respondent assistance	User interface	Respondents need assistance to interact with the mobile app or additional smart devices that might be provided by a survey agency. Following good practices and established standards, during the design of the user interface, help to familiarize respondents with the layout and basic functionality of a mobile app. Smart tools like (gamified) tutorials, in-app knowledge bases and easy options to contact the survey agency via established channels should be part of the user interface design. Futher assistance, for example by interviewers or technical experts, should be easy to schedule via tools within the user interface. In addition, invitations should also include information on how to get on-boarding support. E.g. by including QR codes with links to a web based help-desk and contact option for help by interviewers.
Staff training	Data collection strategy	How staff and in particular interviewers are able to assist during the data collection period, is essential for a smooth survey and data quality. Smart checks and validation can not only alert the respondent to take action, but also the survey agency to get in touch. A good understanding of the data collection strategy and the processes involved is necessary to assist respondents in a helpful way. E.g. the right use or positioning of smart devices has an impact on their sensor readings. Here interviewers can assist to advise best practices for the respondents to increase data quality during the collection process.
	User interface	Interviewers must have hands-on experience with the mobile apps and devices used. Up to date documentation and training material must be accessible by them. Even if basic functionality is highly standardized, each mobile app is unique.

The data collection strategy for smart surveys utilizes the data collection tools in form of a (mobile) app to provide a frontend for questionnaires or reporting and provide services to allow passive and active data collection with sensors of the mobile device and or import of data from external sensors. A data collection strategy for smart surveys must determine, if in-device sensors are sufficient or external sensors/ devices are required to collect the desired data (e.g. wearable devices like activity

trackers), which should still require a companion app to serve as a hub for data entries, reporting features and helpdesk access for the respondent. Furthermore, questionnaires could be integrated in the mobile app. The design goal is to provide a unified survey experience through one standardized frontend.

One of the lessons learned during the field tests in WP2 was that technical support or helpdesk is crucial to assist respondents when interacting with smart devices. Further assistance, passively through a helpdesk website or actively through technical experts, might be required. Such assistance is twofold. On one hand during the recruitment phase, where participation in a survey and motivation are key. On the other hand, by assisting respondents in starting and using smart survey tools. The latter should also be part of the on-boarding process of the mobile app.

Real time validation of smart data with soft and hard checks should be performed on all data collected. Rule based checks require human involvement and their domain knowledge. A smarter data collection strategy includes *contextual data* on the collected data or the data collection process itself. Pilot phases with additional data collection of device information, logs to track user interaction, usage statistics and follow up questionnaires are helpful to identify critical paradata for automated checks. As sensors can fail and respondents may provide false data by accident or on purpose, notifications on missing or inconclusive data could call the respondent or ultimately the survey agency to action. A sound survey monitoring mitigates risks by identifying problems early, in order to adapt the data collection strategy.

3.1.2 User interface of frontend[3]

The most influential design choice in regard to frontend design and incentives is whether or not an incentive is considered crucial *during* the survey phase. Most incentives are given *after* or in some cases *before* the survey starts (c.f.), which means that a reference or hint in the user interface is sufficient. In contrast, *free samples* (e.g. an activity tracker) mean that respondents have to interact with already existing apps to track their progress or tracked information. Alternatively, if the already existing app has a well-defined API for data access, its data could be included in the survey app, which means to adapt the frontend and backend services of a survey app. *Gamification* is an incentive strategy that is focussed on the active survey period and requires the most changes in the frontend design and backend services. Hence, it is an option for longer running surveys and or surveys with high drop-out rates during the survey. Table 3.2 describes different types of incentives and timing.

[3] Nils Meise (Destatis)

Table 3.2 - Incentive Timing

Incentive	before	during	after
monetary incentive	X		X
coupon or discount	X		X
free sample	(X)	X	X
give-aways	X		X
charitable donation			X
Raffle			X
access to survey results			X
gamification		X	

Considerations to implement particular incentives may vary among NSIs. However, there are lessons learned for the frontend and user experience design that should be considered for any survey app. These lessons learned concern the different phases during which respondents engage with the frontend. In terms of gamification we identified three distinct phases for app use: first, *on-boarding*, second, *immersion*, and third *completion*. The most crucial phase for any app-based survey is the on-boarding, which familiarizes the respondent with the frontend user interface and includes the initial consent to access smart device features and to set notifications/ reminders. It is advised to put extra care into testing and pre-testing the design and features of this phase.

The PoC on incentives, which was conducted to understand how gamification could be utilized for smart surveys, showed that gamification performs well as a strategy to convert the hard task of completing a longer running survey into smaller achievable goals. Pursuing these goals defines the immersion and mastering them the completion phase. Both phases need to be represented in the frontend design with distinct features showing the progression of the task ahead and deliver *relevant feedback* for the respondent. Such feedback could be customized for different kinds of respondents as *user types*. An example of such an approach was modelled with the PoC. A broader discussion, which also includes personas as an alternative way of individualizing the survey experience, is discussed in the deliverable on the initial framework. Whatever level of gamification, which includes *game like mechanics and relevant feedback* when engaging with these mechanics, is chosen, it will have an impact on the frontend design of the survey app.

The PoC also raised some methodological and technical issues that need to be addressed during the implementation of gamification. The top methodological issue was the question, if gamification introduces an undesired bias to a survey. Depending on the provided feedback, respondents might change their behaviour during the survey period. Which is something that was reported from WP2 in context of the pilot studies with the air quality sensor and its feedback features. In this example, respondents changed their ventilations patterns due to reported readings that showed “bad” indoor air quality. If we want our surveys to benefit from smart features like self-reporting or gamification,

we have to monitor our surveys closely for changes in behaviour and evaluate the risks. On one hand, if non-reporting and drop-outs are our main concern, we might accept changes in behaviour and mitigate analytical risks with closer monitoring. On the other hand, if un-biased data is crucial or we are not able to employ a close monitoring, we have to accept worse completion and return rates. Given there is no legal obligation to complete the survey at hand.

The technical issues that showed up during the PoC are intertwined with methodological decisions. The use of incentives and adaptive frontend design limits re-usability, if it is closely tailored to the survey at hand. The latter is the preferred implementation to ensure strong mechanics. This issue is mitigated due to the fact that the main principles for implementation of gamification – or in a broader sense other incentive strategies that are linked to the smart survey app – are based on methodological decisions, the discussed trade-offs and design principals for gamified surveys, which derive from design principles for gamified apps. Finally, the main governing decision or options are regulated by the incentive strategies of each NSIs and their need.

On the development side, there is a need for a serious testing to avoid broken mechanics (e.g. shaking a step counter instead of walking to increase the number of steps counted). Regarding the impact of incentives or individualized frontend design on the overall architecture, it is worth to mention that any additional component within this building block produces additional data, metadata and paradata. Consequently, when designing, for example, the monitoring building block, whether or not it is required to monitor or persist the data with components of the smart data acquisition building block and subsequently process with the components of the smart data processing building block.

3.1.3 Data quality: framework and building blocks[4]

A peculiarity of TSSu is undoubtedly the possibility of combining traditional types of data (provided directly by the respondent using a questionnaire) with big/sensor data, provided in active or passive way. The latter in turn can be generated from different sources: internal sensors, external sensors, public online data, personal online data. Ideally, a general data quality (DQ) framework should be declined right from the data source, the type of data and the type of sensor.

For example, structured, unstructured and semi-structured data require different representational DQ metrics based on aspects related to the format of the data. Also, the evaluation of the quality of these data requires the definition and application of different metrics (i.e. DQ assessment of semi-structured and unstructured data requires trustworthiness for the evaluation processes).

In the discussions with WP2, data quality for sensor data played a huge role due to the new methods and technologies, being a hands-on problem. Thus, this topic takes a larger part in the report.

For sensor data passively or actively collected through smartphones/mobile devices, it is necessary to consider characteristics and properties of sensors used, as they play an important role in the outcome of a survey. The quality of sensor measurements can be affected by limitations of the sensor itself, the heterogeneity and fragmentation of devices (i.e., a wide variety of smartphones available models, operating systems (OS) and software versions). iPhones and Android devices usually have the same or very similar embedded sensors, but the way these sensors interact with the operating system (e.g.,

[4] Francesca Inglese and Claudia De Vitiis (ISTAT)

how often measurements are taken with a sensor), and whether and how external apps are allowed to interact with the sensors, differs by OS. It is difficult to standardize in-browser sensor measurement. Different sensor-equipped devices can produce different results raising the issues of comparability. The speed of innovation in sensor measurement poses further threats to comparability of measurement over time.

Different aspects should probably be considered if sensor data derive from third parties, but in this case the flow of the data acquisition would be different and also the quality (checks, assessment) would be carried out in a different perspective.

The quality of sensor data is intrinsically constrained by the characteristics of the sensors. Moreover, device-related error sources, design decisions of the research app, third-party apps, and participants, can interfere with the sensor measurement.

Quality is a fundamental requirement that needs the definition of concepts and metrics, of actions during the data acquisition phase, of validation analysis. Following this perspective, this section focuses on:

- A data quality framework for sensor data
- The realization of the quality framework in:
  - A smart data monitoring system during acquisition – to control survey progress, representation and measurement errors
  - A smart data processing after acquisition – to analyse errors in sensor data.

3.1.3.1 Quality framework for sensor data

Data quality can be defined in terms of compliance to requirements, such as accuracy, completeness, timeliness, consistency, etc. In defining a quality framework for a TSSu, some aspects need to be considered: 1) the intrinsic quality of the sensors used; 2) problems related to the collection phase that can compromise the sensor data quality; 3) the role of respondents in the acquisition of data (active/passive); 4) problems related to security and privacy.

For sensor data, quality can be represented with internal and objective metrics (intrinsic characteristics of sensor data) and with context-based metrics. By following this approach, it is possible to identify two types of DQ estimation:

- DQ assessment, which estimates the quality of the raw data
- DQ evaluation, which estimates the quality of processed data considering context-based metrics.

DQ assessment implies many dimensions: believability (comparison with the correct operating bounds), completeness (missing values), free-of-error (erroneous values), consistency (over time), timeliness (delay), accuracy (deviation from true value) and precision (granularity of readings).

Context-information for DQ evaluation assumes an important role to identify the causes that can affect sensor data quality and to choose the best methodologies to correct errors in the data (pre)processing phase.

Another important aspect on the DQ evaluation is represented by the security and privacy metrics, as data security may influence elements of data correctness. If smartphone users agree to share some

data from their devices, they have to be sure that their other private data is safe (Booth et al. 2019; Immonen et al., 2015).

In the table below, an idea of a quality framework for sensor data is reported with a short description of the main steps.

Table 3.3 - Quality framework for sensor data

Step	Description
Metrics' design	Building the list of data quality metrics (intrinsic and context) and security and privacy metrics.
Metrics' organization structure	Defining metrics organization to facilitate framework modifications and adjustments in order to adopt it in different domains. Metrics must be organized by sensor type in hierarchical/multilevel manner.
Metrics acquisition	Online sources to have information on mobile device for extracting all sensor parameters available and for selecting sensor parameters that affect sensor quality. These parameter's values can be used for the sensors intrinsic quality evaluation.
	Collecting of paradata and contextual data during the ongoing survey to have information about context.
Metrics selection and metrics integration	Defining methods for the integration of DQ metrics (i.e., Data Correctness: consistency, completeness, sensor accuracy).
	Applying Machine learning (ML) techniques for integrating security and privacy metrics (i.e. Data Security: data availability, data integrity, and data confidentiality).
Quality evaluation	Developing evaluation methods by including the security and privacy aspects. DQ components calculated separately and then integrated into the unified overall DQ score.

3.1.3.2 Smart data monitoring and use of contextual data

Collect phase and monitoring indicators in GSBPM terms were described in the Deliverable 3.1 Report on the Preliminary Framework (section 6.2 of task 3.1.1 Smart Survey Methodology).

Here, a smart data quality monitoring system is described focusing on a monitoring dashboards, indicators and paradata. Moreover, an analysis of the acquisition of contextual data, useful for the data quality evaluation, is presented.

For assessing DQ, it is particularly important to have information at the micro level, referred to elementary units. Paradata can offer information on several statistical parameters of the measured

smartphone sensors and insights into their performance, while contextual data can be useful to detect the respondent's behaviour/ability with mobile device, app usage, and the interaction of respondent with sensors.

The problem that arises in a TSSu in the acquisition of information necessary for data quality needs is related to their size and therefore to the application of the data minimization principle (see Article 5 (1) lit c, EU-GDPR) according to which the collection of paradata can be “adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed.”

Monitoring system

In general, a smart data quality monitoring system should be aimed at controlling the data collection process, in order to intervene in real time when problems impacting on the survey progress and on representation and measurement errors occur.

A TSSu monitoring system must be drawn taking into account the differences in the acquired data (passive and active sensors, internal sensors, external sensors, self-report, etc.). This has implications for the way a monitoring system is defined. For example, soft and hard checks should be used on data collected through questionnaires or diaries, or sensors that require the respondent's active participation (sending photos). In these cases, rule based checks or check of scan quality are useful for detecting when participants deliver insufficient data quality and to involve the respondent in the data checking. Notifications of missing data can be used to exhort the respondents to deliver data.

For sensor data passively collected, the situation is different as in many cases the respondents do not know the data they transmit, either they may not provide consent to the acquisition or may have problems due to non-willingness with the device or app. These aspects generate different problems with effects on both representation and measurement errors of sensor data.

The monitoring of incoming data passively (i.e., automatically on the phone, without participant input) and continuously (e.g., every few hours or every few minutes) collected, making it important that any problems with data collection are identified as they occur in real time. Data monitoring tasks include, visualizing participation and attrition rates during the data collection process, and estimating the amount of data collected as the field survey progresses. Such summaries are crucial indications of the application’s performance and participants’ engagement.

Monitoring dashboards

An application monitoring dashboard is useful to detect, understand and resolve issues in the collection phase of a survey (app issues, transmission of data, etc.), and may be inspected on a frequent basis in order to determine whether interventions are needed at overall level. A dashboard is a type of graphical user interface which often provides at-a-glance views of key performance indicators relevant to quality survey objectives. Dashboard system must be designed for monitoring and visualizing data in near–real time. Critical-to-quality metrics are central to the continuous quality improvement approach. Regardless of the metrics identified as key, it can be challenging to simultaneously monitor multiple metrics, especially when monitoring is needed at different levels. Some example interventions for improving data quality include the following: retraining participants in sensor usage, adjusting sensor fit, sending reminders to participants (i.e., smartphone push

notifications) to upload the data. Quickly responding to rectify data quality drops can help preserve the value of the data and minimize data loss.

Monitoring indicators and paradata

In a TSSu, a monitoring system based on indicators and paradata is necessary to control: the various stages in which non-response can occur, as from the consent to participate, to download and install an app (or device), to use the app (whether actively or passively); problems in capturing and transmitting data, if caused by the respondents' behaviour, by technological issues or by the sensor itself.

In the next tables, a monitoring system for sensor data is described based on the input received from WP 2.8. considering indicators to monitor survey progress and a paradata system that should be developed at micro-level.

The two tables below represent examples of what must be monitored during the ongoing smart survey, but it is clear that indicators must be defined on the basis of the research objectives, survey design and data collection strategy.

Table 3.4 - Monitoring system: indicators at macro and micro level

Monitoring type	Indicators
Monitoring participation or drop-out	Indicator at macro-level
	Number of respondents (rate of consent to participate) Number of respondents downloading and installing an app (or device) Number of respondents that consent to transmit sensor data Number of respondents that respect protocol in transmission of data requested Number of respondents that respect protocol in transmission of sensor data Number of participants who drop out of the survey …
Monitoring the quantity of data	Indicator at micro-level
	Data integrity percentage - percentage of data acquired while the app runs

Table 3.5 - Monitoring system: paradata at micro-level

Type of paradata	Type of information
Paradata on app usage	
– for diagnosing issues; – for tracking information on functionalities of the app (e.g. how often did the respondents open the insights page); – to detect insight in technical difficulties in using the survey app related to the device.	Information acquired through logs. Manually reviewing log files and a log management tool to maximize the usefulness of log data that include automated log analysis, real-time log monitoring, log correlation and search, and automated log parsing.
Paradata on device information	
to acquire information - on model, version and on operating system	Information acquired through a browser's user agent string (UA)
Paradata on sensor	
to have control - over what is measured in the app; - on the sensor performance.	Information acquired through: – sensor-related APIs for sensor events - the name of the sensor that triggered the event, the timestamp for the event, the accuracy of the event, and the raw sensor data that triggered the event; – callback methods to monitor raw sensor data - i.e. sensor's accuracy changes
Paradata on data transmission	
to detect no activity signal;	Information acquired through a monitoring system
to detect operational errors - determined by the respondents who may incorrectly initialize the measurements or use the devices wrongly	Information acquired through a monitoring system
Paradata on contact with respondent	
to get information on each contact over time	Information acquired through a monitoring system

Contextual data

In general, contextual data are a set of questions integrated into the survey questionnaire because they are functional to the prediction of the survey construct (see section 2.2 in the Deliverable 3.2 Report on the Proof-of-Concept – task Treatment of sensor data for Smart Surveys through Machine Learning: a Generalized Module "PoC on methodologies"). Nonetheless, contextual questions can be operationalized as a separate questionnaire to control when a respondent registers a specific activity. In the latter case, the methods used to ask contextual questions range from the more traditional (WP2 - Health pilot) ones to those based on experience sampling method (ESM) or beeper-method in which

the questions are presented when a respondent registers a specific activity (Motus -Modular Online Time Use Survey).

The collection of contextual data in a TSSu can be useful to respond to the twofold objective: to understand the degree to which the user interface affects the usage; to understand how the context influence the respondent's behaviour.

Context can be used to characterize users' day-to-day situations that have an influence on their smartphone and app usage, and consequently on data quality. Users' behaviour with smartphone and apps may vary from user to user, according to their contextual information, such as temporal context, work status in workday or holiday, spatial context, emotional state, Wifi status, or device related status etc. (Sarker, 2019).

Such contextual data can be collected passively: the smartphone's sensors can track the user's physical context and the operating system can track the user's interaction with the smartphone and its apps. By processing sensor data, context information can be generated for extracting behaviour patterns or a subject's activity. It is possible to use sensors for specific applications to be able to log more detailed information about application usage, as sensors that query the system state. Changes in the system state can be either triggered by user actions or an indirect result from the usage of the device, for example from the level of battery power.

As context data like detailed location information or app usage statistics is highly sensitive, it is necessary to consider privacy concerns. Another aspect related to the high dimension of contextual data requires the adoption of data reduction methods, to avoid model complexity that may arise over-fitting problem, and consequently decrease the prediction accuracy.

In the next table, an example of information pertaining to usage of apps and contextual information, such as location and proximity, is reported.

Table 3.6 - App logs and contextual data on location and proximity

App logs/contextual data	Description
App logs	App logs consist of the usage events of all applications, including system apps, pre-installed apps and other user downloaded apps. Each time the user accesses an app, the client software captures the event and stores it together with the timestamp (usage frequency).
Location data	The outdoor location information is obtained from the phone built-in GPS. For common visited places in a person's daily life, it is also possible to locate the user based on WiFi access points (APs). The software client periodically scans for WiFi APs and maintains a list of known-location APs (based on the simultaneous availability of GPS).
Bluetooth data	The smartphone scans nearby Bluetooth devices every 1-3 minutes, depending on the state of the client. The number of discovered nearby devices can be used as an approximate measure of the human density and the type of environment, and provides to some degree the social context of the user.

3.1.3.3 Smart data processing: data analysis and validation

Smart data analysis is a part of the data processing building block in which error detection and correction must be done. Measurement (and representation) errors in sensor data were discussed in the Deliverable 3.1 Report on the Preliminary Framework - task 3.1.1 Smart Survey Methodology, section 2: errors in smart surveys.

Aside from issues of selectivity, when inferring from a smartphone-owning participant sample, several factors can potentially influence whether measurements are recorded or valid.

Here, the main sources of measurement errors that can potentially influence sensor quality are summarized with some guidance on which approaches to use. Then a focus on the problems of missing data is reported.

Table 3.7 - Sources of measurement errors and approaches

Factors that impact on sensor data quality	Approach
Differences in intrinsic quality of sensors (device inequivalence)	Algorithm to compensate for the differences Algorithm to combine sensors
Systematic and random measurement errors due to sensor quality and age	Difficult to discern but important for different effects on sensor parameters and data quality
Anomalies in measurements (outlier, noise, missing data, etc.)	Appropriate Strategies/Algorithms based on sensor and error type
Technological and human introduced errors in reporting or data capturing	Efficient system
Participants behaviour (i.e. operating errors, intended or unintended misuse)	Model with paradata and contextual data
Specification errors introduced (in the processing phase) when sensor data are manipulated: - to explore the accuracy and precision of data - to search for patterns - to combine different sensors	Criteria for assessing the quality of gathered data Model with paradata and contextual data to DQ evaluation Appropriate processing strategies (such as aggregation or sampling) Metrics for model validation

Missing in sensor (passive) data

The collection of data from smartphones of different types of sensors can be achieved without success by producing missing data. This missingness can be caused by user behaviour, such as switching off the smartphone or running out of battery, by characteristics of the location (no signal), or by technical problems related to the hardware or OS of the smartphone (e.g., energy-saving modes).

Sensor data can be missing for short periods of time, due to communication loss or technical issues but, also, for longer periods. The entity of missing data may vary due to smartphone batteries running empty, or to a particular sensor, an app, or to the device itself when it is turned off by the participant. Measurement challenges can exasperate the missing data problem, and the collected data will not reflect the true behaviour of an individual. This is the case in which participants install the apps but fail to carry a smartphone everywhere (Bähr et al., 2020).

The strategies of dealing with missing items are very complex because data vary across sensors, depending on the extent and nature of the missingness patterns, etc. Models to disentangle these factors interfering with data quality are needed.

Paradata (i.e. information on the device, its operational state - i.e., battery level, display state -, the state of the mobile network connection), contextual data (i.e spatial, temporal, etc.) and other type of information (socio-demographic) are fundamental for the definition of a complex model for analysing and then controlling the error source in the missingness processes (Bähr et al., 2020).

Bähr et al. (2020) propose a multi-stage model to deal with missing in mobile geolocation sensor data. The approach proposed by the authors can constitute a reference framework for the analysis of the factors that can generate missing in the acquisition of sensor data more generally.

In the next table, some relevant aspects of the strategies to address missing processes and the definition of multi-stage model are reported.

Table 3.8 - Strategies to address missing processes and model

Phases	Objective
General analysis	To detect missingness processes
Definition of Stages (the number of stages depend on the missingness processes)	
1: Device turned off	To detect episodes where the device was turned off. If a participant regularly turns off the device, this systematically affects measurement and inference.
2: No measurement at all	To detect episodes where the research app could not collect any measurements at all from the turned-on device. Energy-saving modes, restrictive permission management by the OS, or even third-party task killer apps could prevent an app from collecting data at the time of a scheduled geolocation measurement.
3. No sensor measurement	To detect episodes where no measurements were collected when the device is turned on and the app is actively collecting data. Technical reasons for failed measurement or users that manually disable detection in the general settings of the smartphone.

Phases	Objective
4. Sensor measurement failed	To detect episodes with anomalies in sensor measurements The device type and OS version determine the quality of the built-in sensors, the computational resources, and the underlying algorithms. User-related factors such as mobility or the placement of the smartphone on the body can also influence the quality of measurement.
Model to disentangle factors that impact on sensor data missingness	
Definition of two separate models in each stage for: - occurrence (identification of sensor measurement gaps) - extent (the length of each identified gap in terms of the number of missing measurements)	To allow for factors to have differential effects on either and to avoid confounding effects of the previous stage.
Definition of Dependent variables	Device turned off–Occurrence, Device turned off–Extent No meas. at all–Occurrence, No meas. at all–Extent No sensor meas.–Occurrence, No sensor meas.– Extent Sensor meas. failed–Occurrence, Sensor meas. failed– Extent
Definition of Auxiliary variables	Participant-level variables (socio-demographic) Measurement-level variables (data permissions, device hardware device state, contextual data, etc.)

3.1.4. Machine Learning potentiality in Smart Surveys[5]

One of the goal of ML techniques is to process the huge volume of data collected through smart devices, thus deriving statistical information from sensor data that are not standardized in structure, format or availability. In addition, ML algorithms allow identifying hidden structures and patterns from the data. They can support different phases of the survey (data acquisition, processing, mixing/fusing sensor data). ML algorithms can be very useful in the statistical production process, for example for the processing of sensor data with the aim of performing the tasks of classification or regression of variables of interest or for the treatment of errors.

[5] Fabrizio De Fausti and Massimo De Cubellis (ISTAT)

This section provides a short description of the previous deliverables. It describes some ML functionality for a TSSu platform, the different type of ML tools, the experiences gained with the ML PoC, considering the input received from WP2 pilots (in particular HBS and HEALTH).

3.1.4.1 Short summary from previous deliverables

The previous deliverables presented:

1. An overview of the machine learning techniques (Del. 3.1 Report on the Preliminary Framework - Task 3.1.1 Smart Survey Methodology – section on ML).
2. A Generalized Machine Learning Component (GMLC) with the aim of making a contribution to the development of a European platform for TSSu (Del. 3.2 Report on the Proof-of-Concept – chapter Treatment of sensor data for Smart Surveys through Machine Learning: a Generalized Module "PoC on methodologies")

Regarding the first deliverable, some issues have been analyzed with respect to application of ML to sensor data in smart surveys:

- differences between three different types of algorithms: supervised, unsupervised and reinforcement learning;
- how there is no direct relationship between the data collected by each type of sensor and the automatic learning algorithm for their processing; the algorithms to be applied to the data collected by the sensors depend on the use cases we want to achieve;
- how the machine learning task and the type of label, defined from the use case, are associated with an appropriate list of ML algorithms;
- aspects related to quality, considering ad-hoc framework for quality assessment (Quality Framework for Statistical Algorithms (QF4SA);
- where and how ML algorithms can be applied in the context of the production process of smart statistics, taking the GSBPM as a reference standard.

The second deliverable focuses on the development of a generalized ML component (GMLC) designed to perform a supervised multi-class classification task, in particular a ML component able to predict human activity or means of transport starting from signal data coming from the accelerometer sensors. The data sources used are: data provided by WP2.3 for the Health pilot, used to perform an accelerometer experiment by the activPAL device; data provided by the University of Trento, which are the result of a smart survey carried out in 2018 within the SmartUnitn(Two) Project (iLog).

Summary of the ML PoC deliverable:

Component requirements:

- Designed in a modular way, in the sense that the component is made of modules or programs implementing well-defined process functions with inputs and outputs
- Insertable in different stages of the production process, depending on how the survey is designed
- Survey-agnostic implementation

Two-step development:

- Starting from data provided by WP2.3 for the Health pilot, we have built a specific ML component for Health able to classify human activity

- Generalization of the component used in the first phase to classify the means of transportation in the SmartUnitn(Two) survey.

Scenarios:

- The GML component can be applied to several contexts: one examples is to train the component on data from users participating in a pilot-survey, with the aim of creating a ML model to embed into an App that will be distributed to carry out the real smart survey; another possible context is to insert the untrained component in the device where it learns from the single user's behavior on how to classify the variable of interest for the survey.

Strengths:

- The GML component allows to process signals from different sensors (accelerometer, gyroscope and thermometer) also jointly
- Use of the machine learning algorithm in several process steps, from classification to regression
- Modular implementation that allows to add other subcomponents to perform new functions.

Weaknesses:

- The GML component does not guarantee the same levels of accuracy as an ad-hoc component;
- Training is expensive in terms of time and computing power and human effort

3.1.4.1 ML – Smart data tools

In relation to the TSSu application layer, the "Smart data tools" building block realizes and implements the functionalities for smart data collection, editing, validation and incentive mechanisms. This paragraph analyzes the use of ML algorithms during smart data collection.

Generalized ML smart data tools

A general platform for all NSIs could provide services for the design and development of ML algorithms to execute several steps of a survey, in particular during data collection. In fact, ML models can be applied to deal with input data provided by smart devices such as images, signals, voice.

As described in the ML PoC report, a ML component has been developed, following agnostic approach, to process the signals from different types of sensors and infer several variables, for example the physical activity or the means of transport of the respondent.

Therefore, the software solution is generalizable in the sense that it can perform its function in several contexts and in different phases of the survey. As shown, this development is possible at the cost of seeking a trade-off with quality.

Deployment of ML Smart data tools: ML in-app vs ML in-house

Machine learning models can reside directly on the devices where they receive the user's data (in-app) or be distributed as services and reside in centralized servers (in-house). In the design of a TSSu, the choice between the two solutions depends on the advantages and disadvantages provided by the two approaches and the requirements of the survey.

In-app training / re-training

The model can be trained or re-trained on the device with data provided directly by the user collected during the data collection

- Given the limited capacity of the devices, it is preferable to install lightweight models that require limited storage and low computing power
- This approach is an advantage to prevent sensitive data from being transferred outside the device
- To avoid problems of autonomy and performance in the use of the device during the day, the training can be performed at a time when the device is at rest.

In-app inference

- During the inference phase, it is preferable to use lightweight ML learning models. In fact, a complex model would need a large storage on the device, and it would be power consuming by decreasing the device's battery life
- The early response of the model allows a better interaction with the respondent, supporting quality controls
- Privacy protection improvement: user's raw data does not leave the device and only the required information that is essential for the investigation is processed and centrally stored in NSI's premises.

When implementing an in-app algorithm, you will need to use dedicated ML libraries for devices such as *TensorFlow-lite*. A platform for TSS could provide a layer to decouple the implementation of the algorithm and the deployment of the code on the device.

In-house ML

In-house ML models do not reside on the device but in servers or cloud architectures, and expose services available via the Web through end-points. Devices send to the servers the necessary data for inference and receive the result through APIs.

Despite in-app ML algorithms, In-house ML models do not necessarily have to be lightweight, due to the use of server's resources, such as memory, or the scalability on cloud architectures. This can have positive effects on inference, accuracy and quality. A TSSu platform could also provide model endpoints.

The maintenance and updates of the algorithm or retraining can be performed in-house without changing and updating users' application.

During the inferential process, the response of the model is bound to the internet connection with the respondent that may vary during a survey. Therefore, an application must be designed for the potential management of data acquisition even without an internet connection, otherwise during data collection, the quality checks based on the response of the ML algorithm, would fail.

The user's raw data for inference must be transmitted over the web and may contain sensitive information such as photos or GPS data. In addition, data such as photos can be very heavy and slow down the response of the server.

Training ML model in a design phase can be beneficial because datasets can be built for training by integrating data from different sources.

Quality check of ML monitoring in-app

ML algorithms do not always achieve the maximum accuracy or the expected levels of accuracy during the data collection phase. This can be due to several factors, such as incorrect use of the device. During the design of a TSSu, it is recommended to plan quality control mechanisms of the data collected by the sensors. These controls can include different actions, depending on the design of the survey and the degree of the interaction with the respondent, both for burden factors and because the iteration with the respondent does not always improve the quality. The possible actions to improve the quality of collected data are:

- (Active) An in-app notification system that can warn the respondent on how to use the device, asking to provide the variable of interest bypassing the inference made by the ML, repeat the data collection (e.g. take a new photo of the receipt), or just notifying the incorrect collection,
- (Passive) sensor data with low quality will be inspected manually during a centralized control and correction phase, at the end of data collection.

In both situations, to trigger the actions, it is important to carry out a quality check of input data, based on the probability computed by the ML algorithm in the inference phase.

3.1.4.2 ML smart data monitoring

The smart data monitoring building block allows to monitor smart data acquisition, enriching the traditional indicators with the analysis of paradata and contextual data to check the fieldwork. It focuses on exploiting the data produced directly by the device, regardless of the app used for the survey. Such checks can be done with unsupervised ML anomaly-detection algorithms on paradata.

The monitoring of the quality of a ML tool is an important aspect of a TSSu and can be managed through the concepts of Human in the loop and Active ML.

Human in the loop

The concept of Human in the loop (HITL) derive from the awareness that ML, despite having demonstrated extraordinary predictive abilities, is not always able to achieve the desired results in a fully autonomous way. To increase the levels of knowledge of the real world, human intervention is required.

HITL can be defined as an approach that places people's knowledge and experience at the centre of automatic learning processes. The interaction’s process between human and machine, as the name suggests (loop) is continuous. There is a virtuous circle in which the machine produces predictive models with increasing accuracy, thanks to the feedback received from humans.

In practice, HITL refers to systems that allow humans to provide feedback to a model where predictions are below a certain level of confidence. The acceptable level of confidence is defined, and human intervention is required below it.

In general, more data is used to train a model, the better its performance will be. In most cases, it takes years to collect the necessary data to produce very accurate models, or if we want to use the available open data, these do not always exactly reflect our needs. To speed up times and to be able to use predictive models immediately, we therefore can foresee a human interaction in our production process.

In designing a platform for TSSu, the HITL concept is very important. As we have seen, ML can be used in different stages of the statistical production process and the levels of "error" that one is willing to accept can be different from case to case.

In the different phases of the process in which ML models are applied, it is necessary to define acceptable levels of accuracy and integrate the HITL approach, to guarantee the desired outcome.

Active and On-line machine learning

The term active learning refers to a ML algorithm that learns through the interaction with the user. In this scenario, the user provides labels to the algorithm to make the learning dataset richer and improve performance. In active learning, the mechanism by which the algorithm requires the user to enter the label is of particular importance. Since the algorithms chooses the examples, the number of examples to learn a concept can often be much lower than the number required in normal supervised learning. In the literature, there are many criteria for selecting the examples to ask the user based on the examples that would change the model the most, or would reduce the generalization error the most, or would minimize the variance of the output.

In a trusted smart survey, active learning can be useful to ask the respondent to label data that can be sent in-house for re-training, or processed on the device. This can be particularly useful if you need to specialize the algorithm for target situations, such as specific behaviors of the user or population (e.g. we can think about the need to make the algorithm more accurate and tailored to the characteristics of a country). Pool-Based Sampling or Stream-Based Selective Sampling mechanisms can trigger the request to the user to assign a label.

While active machine learning aims to search for useful examples, incremental learning and online learning are techniques to re-training an algorithm using updated data from time to time. The scenarios in which algorithms must make predictions in very dynamic situations, and where the data flow that constitutes the dataset is almost continuous and changeable, are particularly suitable for online training are. In a TSSu, such a situation occurs when you want to train algorithms for the automatic classification of products that are subject to continuous updates, as in HBS pilot of WP2.

3.1.4.2 ML smart data processing

This block provides a number of methods for transforming raw sensor data (signals, images, text) into statistical information (data preparation and data modelling). The analysis carried out refers to a phase subsequent to data collection.

Smart data analysis and validation is part of data processing building block that work on two main aspects:

- Processing of sensor data
- Treatment of errors and missing values.

ML technical specifications can be applied in both cases.

Regarding the processing of unstructured data, the production of variables derived from raw data, such as images or text, into variables can be addressed with well-known ML techniques, for example OCR.

As for the treatment of errors and missing value, ML algorithms can be used, as discussed in deliverable 3.1 of WP3, to support the data imputation and validation phases.

Unsupervised machine learning techniques such as PCA or cluster analysis, can identify the types of missing data through the study of paradata and contextual data. The identification of the type of missing data supports the data entry phase also. These operations aims at improving the quality of the data and therefore of the statistical outputs.

3.2 Privacy Preservation for TSSu[6]

To improve upon the work done in sub-task 3.2.3 and facilitate integration of thereof within the enhanced framework, the following chapter will aim to summarize the results and align the terminology with the TSSu architectural framework described in chapter 2 of this document. Specifically, we will discuss how modern privacy preserving technologies apply to the GSBPM phases shown in figure 2.1 and described in 2.2, as well as to the building blocks shown in figure 2.2.

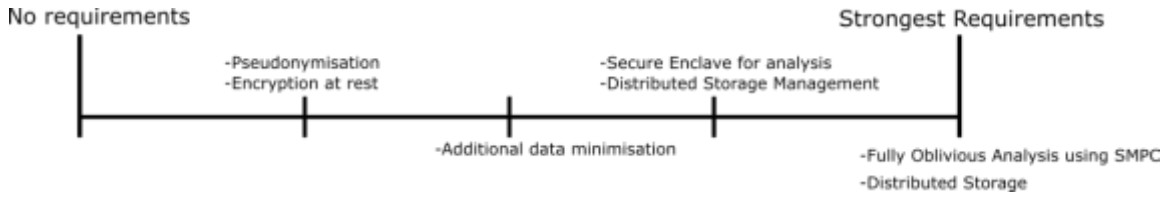
3.2.1. Privacy preservation and GSBPM Phases

In paragraph 2.1 of the "Report on the Proof-of-Concept" (Deliverable 3.2), the relation between privacy preservation and the phases of a (smart) survey were presented. To align these with the official GSBPM phases, we stumble upon the first enhancement of the ideas presented in Deliverable 3.2. Namely, there the first phase was identified as the design phase. However, an important step that should be performed first is to identify the privacy requirements: the "identify needs" phase, discussed below.

During this first phase, an assessment of all requirements should be made. When it comes to the privacy requirements, they produce a list of measures that should be taken that ensure these requirements are met.

Importantly, there should be a balance between the requirements on privacy and the measures taken to satisfy these. That is, strong requirements require strong measures and *vice versa*. To illustrate, below in Figure 3.3, this balance is shown

Figure 3.3. Measures on spectrum of privacy requirements



Application of the privacy preserving measures impact the GSBPM process from start to finish, the aforementioned should be done as soon as possible. Note that during the next GSBPM phases "Design" and "Build", there exists a possibility that one finds that privacy requirements are at odds with the other specifications resulting from one of these phases. This could be due to a lack of (feasible) technological solutions, or because of a fundamental restriction for aligning these requirements.

"Design & Build"

Having identified the privacy requirements, a survey has to be designed accordingly. Implied is the necessity for incorporating the required privacy preserving techniques in the design of the survey. For

[6] Joeri van Etten (CBS)

example, if a survey requires online training of machine learning models on smartphone sensor data, but privacy requirements prohibit collection of thereof, then one should design a federated learning implementation that allows for these models to be trained without the need to collect the raw sensor data.

The design and development of the privacy preserving measures necessary for a given survey can be divided into two parts: (1) the implementation of abstract protocols required by the chosen privacy preserving technique (*e.g.* secret sharing of numbers along with protocols for arithmetic operations) and (2) implementation thereof within the survey

It should be stressed that design of the privacy preserving implementation cannot be done separately from design of the other survey components (collection, analysis, data storage, methodology, *etc*). The reason for this is that design of the privacy preserving implementation can have a profound impact on that of the other components and *vice versa*. Privacy preserving techniques are hence not a building block one can attach to an already designed component. Rather, the survey components should be designed in a way that satisfies the privacy requirements identified in phase 1. Hence, design and development of the survey should be a joint effort of statisticians, cryptographers, software developers and software engineers.

“Collect”

As part of the ‘Smart Data Throughput Activities’, during the “Collect” phase, pre-processing steps are added to prepare for execution of subsequent phases according to privacy requirements identified in phase 1. Without exception, these steps are performed *before* data leaves the site of collection. If privacy requirements are relaxed, pre-processing can be as simple as pseudonymization. When these requirements are strict however, pre-processing can be a complicated series of operations to achieve sufficient data minimization, prepare for oblivious analysis and enable storage without a single point of trust.

“Process”

The amount of additional processing required before the analysis phase can start depends on the amount of pre-processing that has been done on-device. This dependence is related to privacy requirements.

If privacy requirements are strict, data has to be minimized on-device during pre-processing. For efficient privacy preserved computation, the same applies. Hence, application of privacy preserving techniques shifts processing requirements from the “process” phase to the “collect” phase.

“Analysis”

The way in which analysis is performed is highly dependent on the required privacy measures. For example, if we are dealing with relaxed requirements for which pseudonymization might suffice as a sole privacy measure. Application thereof has no impact on the way in which analysis transpires.

When privacy requirements are strict however, and analysis has to be performed obliviously, the entire analysis process needs to be performed without human intervention. In a secure enclave, this

can be a predefined pipeline consisting of virtually any function. Utilizing SMPC, the possible components of this pipeline are more limited, though this technique provides more security.

"Disseminate"

During the dissemination phase, the usual statistical disclosure control considerations apply. That is, it should be ensured that data is sufficiently aggregated so to prevent reidentification of individuals. Also, the risk for other forms of disclosure should be prevented. Note that this is nothing new however, as traditionally, statistical institutes have been dealing with disclosure risks related to dissemination of aggregated statistics.

3.2.2. Implementation of Privacy Enhancement

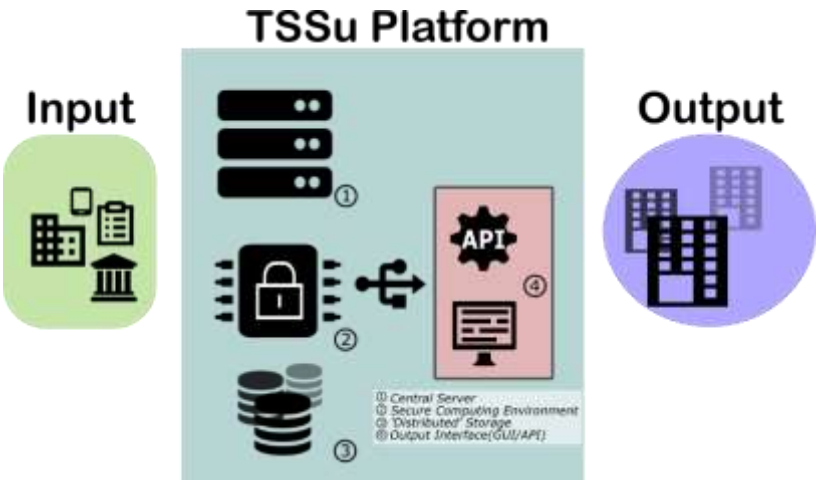
In order to ensure that the requirements identified in the first phase of the GSBPM process can be satisfied, privacy measures need to be taken. For example, traditionally NSI's have used pseudonymization to prevent private respondent data from being accessible to NSI employees. Additionally, for digitally collected data, they are encrypted while being in transit. Of course, such measures will be applicable to the TSSu platform as well. However, to accommodate for more stringent privacy requirements, which might arise from heightened sensitivity of the data collected from smart sensors, additional measures might be necessary. Below we present a list of measures that are expected to be relevant to the TSSu platform.

- **Pseudonymization:** replacement of unique identifiers (*e.g.* a national identification number) by randomized values. This prevents records from being directly identifiable. However, dictionaries relating the randomized identifiers to individuals are kept to retain the possibility of joining datasets on these keys.
- **Data Minimization:** the process of reducing the amount of information present in collected data to that which is necessary for a given purpose. Minimization of data protects privacy through removing the possibility of using it to extract additional sensitive information. This is a concept that is limited in its relevance when it concerns traditional questionnaire based surveys. However, it is extremely relevant when considering smart sensor based surveys. Namely, within sensor data there (likely) exists a great excess of information. For example, consider the air quality data collected in WP2 (add reference to WP2). These data are collected for assessing air quality, however, CO_2 data can also be used to infer when residents go to bed, or leave the house for work. This is private information present in the data other than that for which the data was collected.
- **Oblivious Analysis:** processing and analysis of data without the information contained within becoming accessible. By far the strongest privacy measure, though just as difficult to implement. There exist various techniques that allow for oblivious analysis, for different applications and scenarios. For example, training of ML models can be achieved obliviously using *federated learning,* a distributed machine learning implementation that does away with the need for data to be collected centrally (see 3.2.3.A of Deliverable 3.2 for more information). When it comes to oblivious aggregate statistics on respondent data, this can be implemented using *secure enclaves* or *secret sharing based secure multi-party computation* (see 3.2.3.A of Deliverable 3.2).

In 2.3.4 of 3.2.3 of Deliverable 3.2 a general architecture that allows for all the aforementioned techniques to be applied was presented. This architecture is shown in 2 . It consists of the following components:

- **Input Parties:** these can be individual respondents, NSI's or other institutions. They provide data in various formats, from questionnaires to sensor data. Importantly, for privacy enhancing techniques, part of the processing burden is put upon these parties.
- **TSSu platform:** the set of components that make up the platform:
    - **Central Server:** coordinates collection, processing and analysis while ensuring privacy requirements are met. Also manages access policies.
    - **Secure Computing Environment:** for performing computation according to chosen privacy enhancing measures. For example, for performing Secure Multi-Party Computation, this environment should consist of multiple disjoint servers, owned by different organizations.
    - **Distributed Storage:** for secure storage of input data. 'Distributed' in the sense that responsibility for safeguarding is not taken by a single party, but distributed across multiple. This can be achieved by distributed encryption key management, or actual distribution of data through secret sharing.
    - **Output Interface:** output parties query platform data through this interface. Should support API as well as GUI
- **Output Parties:** NSI's with right to access platform data. These query platform through output interface to subsequently receive results produced by platform. Results obtained are typically aggregated, microdata stays within the platform.

Figure 3.4 Architecture for privacy measure implementation



To learn more about how these components come into play during implementation of the various privacy measures along the GSBPM phases, see 2.3.4 from 3.2.3 of Deliverable 3.2.

3.2.3. Architectural Building Blocks & Privacy

To understand the relation between the aforementioned privacy considerations and measures, and the architectural building blocks described in paragraph 2.2, below we present an assessment of issues that might arise and what measures might be required to deal with these.

Smart Data Tools

- **Mobile app:** For the parts that run on the respondent's device, no privacy issues should be encountered. The same applies for app components that require in-going communication coming from the platform. However, for components that require out-going communication, privacy requirements apply and should be met.
- **External Sensors:** On-device; collection, processing and storage of external sensor data can be performed without privacy restriction. When the data leaves the device, however, it should be considered highly sensitive and all appropriate privacy considerations apply.

Smart Survey Monitoring

- **Fieldwork monitoring:** This component involves access to app usage data. Sensitivity of this data is expected to be less sensitive than actual survey data. Nonetheless, it will have to meet the appropriate privacy requirements.
- **Respondent Assistance:** Basic assistance should not require privacy sensitive information. For more advanced/personalized assistance, app usage data might be required.
- **Quality Checks:** This component is complicated from a privacy (measure) perspective. Paradata required to assess the quality of the actual survey data is, in many cases, expected to be as sensitive as the actual data. Hence, processing and storage should meet the same requirements. Further, analysis required to obtain quality indicators can be complex, so implementation of privacy preserving technologies can be involved. For these reasons, it is advised to treat the quality checks building on the same footing as the actual analysis building block during development of the survey.

Smart Data Acquisition

- **Active Data Acquisition:** As with all data provided by the respondent, privacy requirements apply. For actively provided data, the benefit is that the data collection strategy can be very focused. That is, for example, respondents can be asked very specific questions that provide only the information required for producing required outputs. Hence, implementation of data minimization is practically implied. Also, whenever advanced privacy enhancing technologies are required because of strict privacy requirements, most of the pre-processing burden can be taken away by collecting the right data points.
- **Passive Data Acquisition:** As opposed to actively collected data, passively collected data are highly non-specific, of higher volume and require more advanced pre-processing. From a privacy perspective, this poses several challenges. In contrast to actively collected data, minimization process is not as obvious and already might require advanced processing. For example, extracting information of interest from sensor data often requires machine learning based inference. When it comes to the application of more advanced privacy preserving techniques, similar challenges arise. The amount and complexity of processing required to get from raw sensor data to statistical results is expected to be too high for practical implementation using the appropriate protocols and hence, the aim should be to minimize data locally and perform privacy preserved computation, using chosen technique, on minimized data.
- **Data Storage:** The most basic protective measure when it comes to storage is encryption at rest and should always be applied. However, if the key and storage server are compromised, an attacker will have free access to all data. Hence, if both are managed by the same organization, then this is a single point of failure considering the security of the data.

Protection of the data can be enhanced when storage and key management are taken care of by different organizations. It can be enhanced further by dividing encryption keys into secret shares and distributing across multiple organizations. Finally, maximum protection is achieved when the data itself is split into secret shares and stored in a distributed fashion. Note that, if a multi-party computation protocol based on secret sharing is required for analysis conform privacy requirements, that this is the only appropriate storage solution.

Smart Data Processing

Most considerations regarding privacy as it relates to the smart data processing building block have already been stated. However, given the importance, we will reiterate

- **Error Handling:** For effective processing and aggregate statistics, proper handling of errors and imputation of missing values is of particular importance. In particular, when dealing with strict privacy requirements, access to data is limited after it has left the respondents private space. Hence, at this point, the data should be free from anomalies as no corrections can be made.
- **Pre-processing:** As with error handling, pre-processing is vital for effective processing while conforming to privacy requirements. Pre-processing is required for data minimization as well as to prepare for privacy enhanced computation protocols that greatly increase in feasibility when pre-processing is done well.
- **Aggregate Statistics:** Depending on the privacy requirements, this component can either be relatively simple or very complex. When requirements are relaxed, production of aggregate statistics can transpire without hardly any restrictions using data that has only been pseudonymized data. For strict requirements, we might require aggregation and subsequent analysis to be performed obliviously using secret sharing or secure enclaves.
- **Quality Assessment:** Derivation of quality indicators should be a part of pre-processing. This way, means to derive these are not limited by privacy measures. Indicators can then be used for survey monitoring or to dismiss data that does not meet quality requirements.

3.3. Smart metadata building block[7]

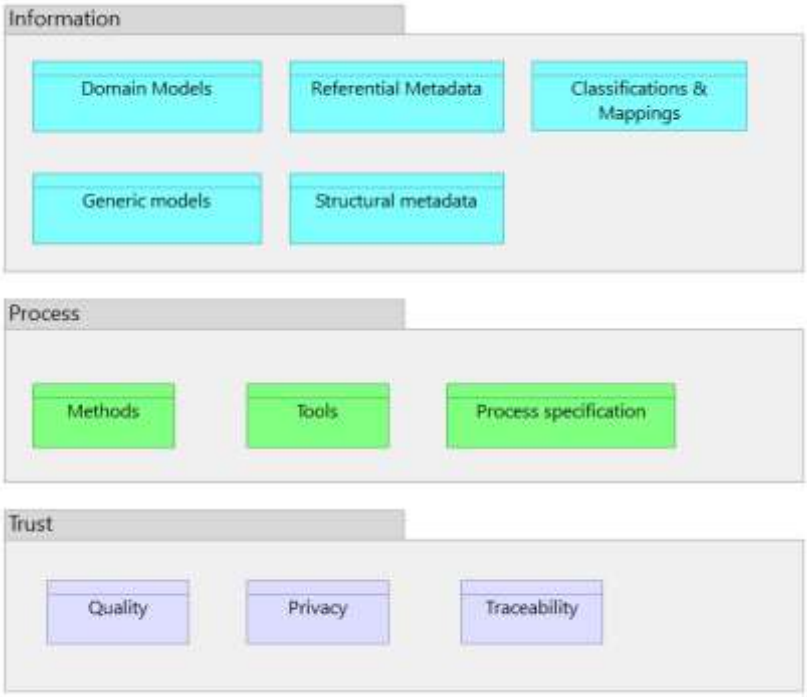
The business layer of the TSSu platform, modelled in Figure 2.1, shows the metadata management as an overarching business function, supporting and documenting the whole statistical process. In order to investigate such a vast subject, in the preliminary framework, the metadata component has been designed to meet the following general requirements:

- **Compliance and alignment with official statistical standards and existing frameworks**, i.e. analysis of the following statistical standards and frameworks:
  - Ontologies (W3C Semantic Sensor Network Ontology, wf4ever Research Object Ontologies)
  - Big Data Reference Architecture and Layers (BREAL)
  - European structural metadata and quality framework
  - Generic Statistical Business Process Model (GSBPM)
  - Generic Statistical Information Model (GSIM)
- **Capture of metadata to describe the statistical process,** including the following dimensions:
  - Collection instruments, variables and codes
  - Statistical methodology used in data processing
  - Process and lineage
  - Quality assessment
- **Process standardization and reuse of application components**
- **Use-case-driven approach**, to test the initial assumptions related to metadata concepts selected for the preliminary analysis of the TSS_U framework.

The main component of the smart metadata building block is the central **Metadata Repository** (MR) shown in Figure 3.5.3. The MR is composed of three main areas: Information, Process and Trust. Such areas are described in the following section.

[7] Mauro Bruno and Giuseppina Ruocco (ISTAT)

Figure 3.5: Metadata repository subsets



3.4.1 Metadata Repository core concepts

The metadata repository contains all the information objects needed to meet the general requirements specified above. These objects have been grouped in three subsets:

- **Information**, the first subset includes the metadata concepts related to:
    - Domain models, describing survey main concepts and objectives
    - Generic models, derived from the reference frameworks (such as, BREAL information model), or providing a common representation of the main concepts describing TSS_U, such as ontologies
    - The description of variables, units, data structures, classifications involved in data processing.

  Some of these objects, such as Referential metadata, are widely documented and can be modelled according to the European Statistical System (ESS) standards for reference metadata reporting[8]. For this reason, most of the elements belonging to this subset have been excluded from the following analysis, to focus on process metadata produced during sensor data processing.
- **Process**: the second area concerns all the tasks executed to produce a statistical output. The main elements belonging to this area provide an inventory of the **methods** and **tools** used for data processing, as well as a description of the main process steps to perform (**Process specification**).

[8] An overview of ESS metadata reporting standards is available at:
https://ec.europa.eu/eurostat/data/metadata/metadata-structure

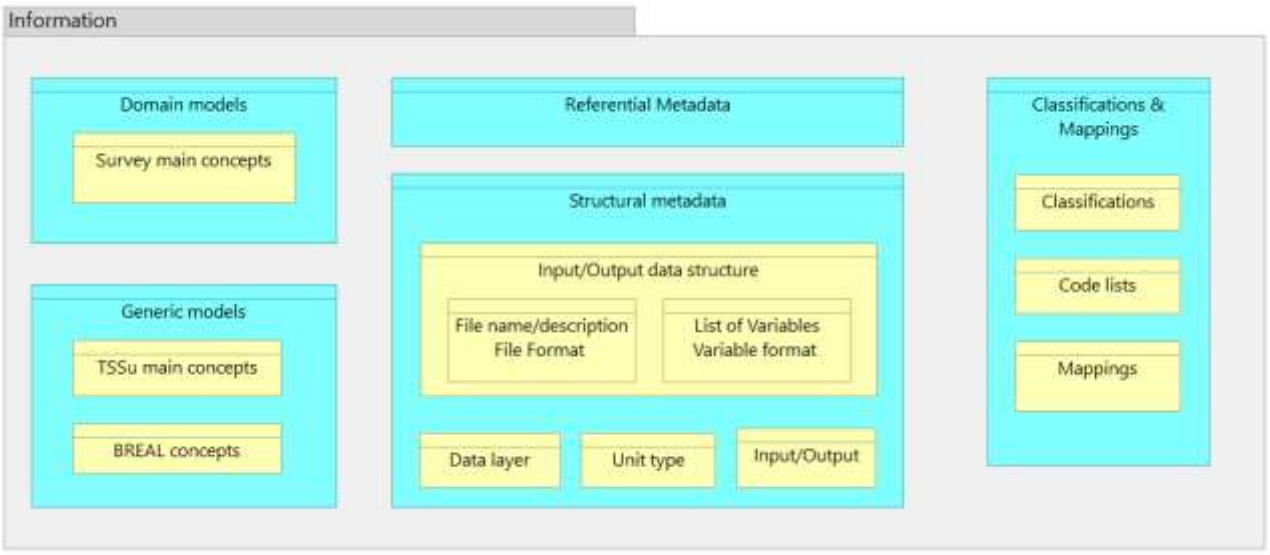
- **Trust**: the third subset includes several types of metadata, such as quality indicators and additional information to manage and monitor **privacy issues**, **data provenance** and **process traceability**. The elements grouped in this area are overarching, i.e. related to each step of the statistical process. In relation to the development of generalized software solutions, while the content of the first group of metadata (Information) is more tied to the survey peculiarities, the elements within the Process and Trust subsets can be easily standardized.

3.4.2 Metadata Repository Proof of Concept

The main goal of the metadata PoC is to validate the MR subsets and define the metadata related to smart data processed through ML models. More in detail, this benchmark has resulted from the analysis of the metadata produced during the activities of Task 3.2, concerning the "Treatment of sensor data for Smart Surveys through Machine Learning Generalized Module." The metadata PoC has confirmed the feasibility of a central repository for the collection and management of the process metadata produced throughout a particular task.

In order to identify the metadata to capture during machine learning execution, the elements of the MR subsets have been further detailed. In addition, some of these elements have been defined on a more granular level, according to GSIM and ontologies concepts. The figures below show the elements modelled in each MR subset. More precisely, within the Information subset, the analysis has concerned primarily the **Structural metadata**, due to its relevance to start any type of data treatment.

Figure 3.1: Main elements of the Information subset



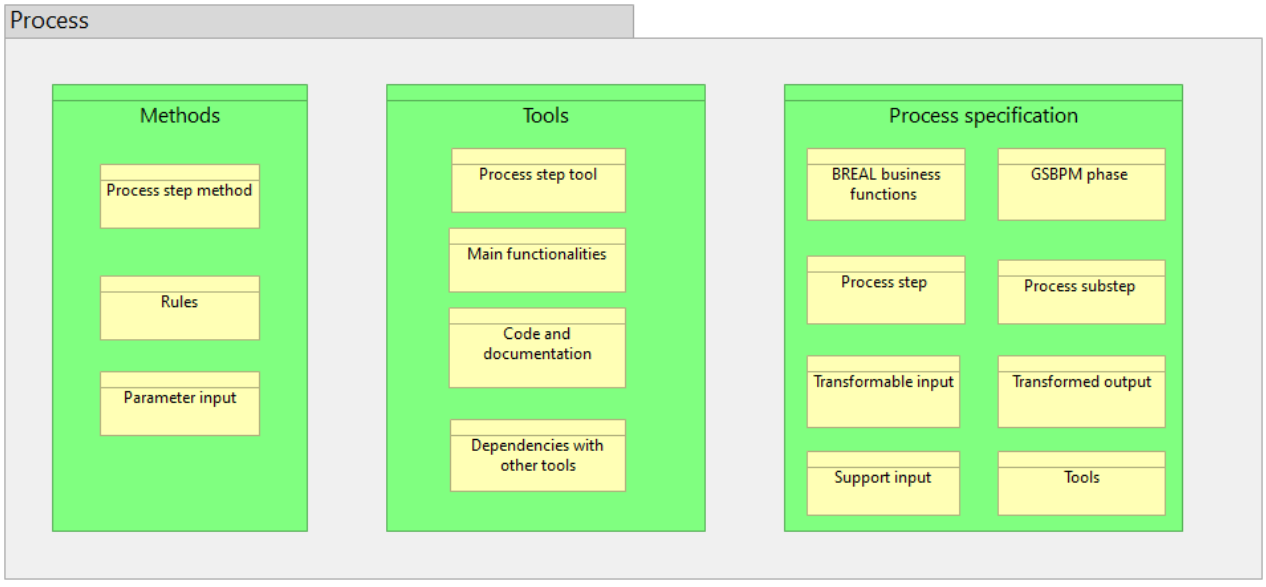
The following table describes the structural metadata belonging to the Information area and captured for the machine learning PoC.

Table 3.9 – Metadata captured for the machine learning PoC

MR Component	MR Subcomponent	Description	Reference standard
Structural metadata	File name / description	Name and/or general description of processed dataset	
	File Format	Standard used to encode and store information	
	List of Variables	Characteristic of a Population to be measured	GSIM
	Variable format	Standard used to represent, read and write variables	
	Data layer	Transformations of smart information objects through the statistical process (Raw, Convergence, Statistical)	BREAL
	Unit type	Class or group of objects of interest, based on a particular characteristic	
	Input/Output	Specification of the main Input and Output related to the Process Step execution	GSIM

The main elements belonging to the Process subset allow to track and document the methods and tools used for data processing (see Figure **2.1**2.1 in Chapter 2.). For each process step, the description of the methods, rules and parameters, as well as the specification of data input and output, allows to fulfil the requirements concerning process tracking and auditability (Figure 3.5).

Figure 3.2: Main elements of the Process subset



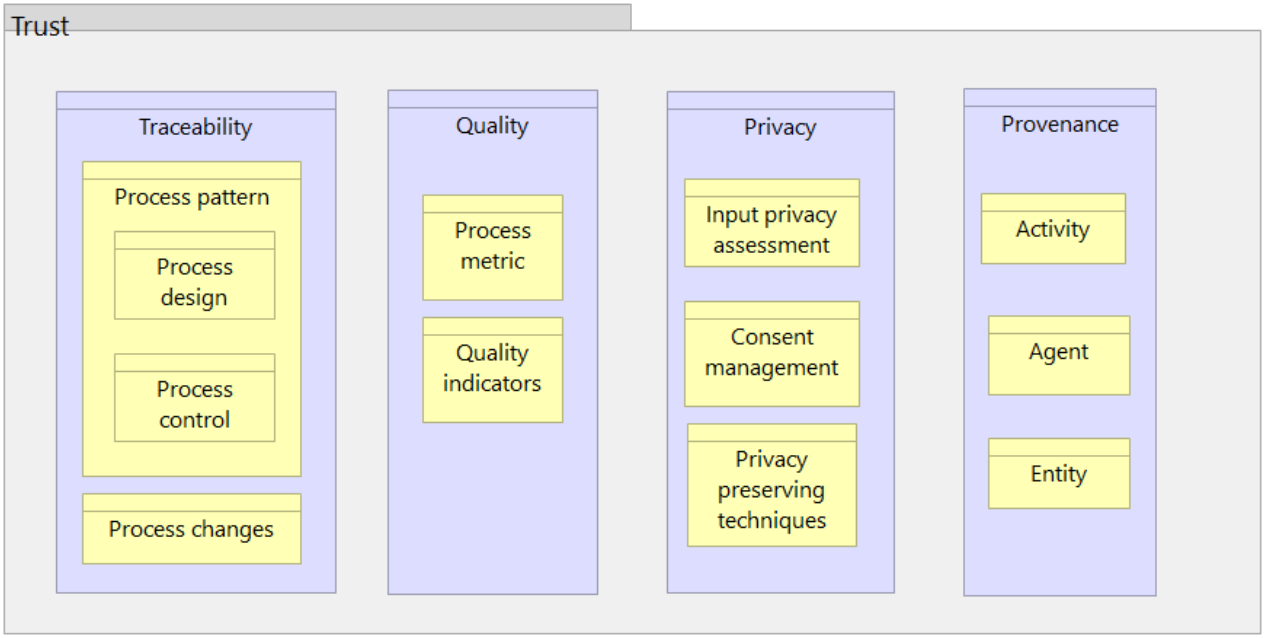
The description of the elements included in the Process subset and captured in the ML Poc (Methods and Process Specification) is reported in the following table.

Table 3.10 – Methods and Process Specification – ML PoC

MR Component	MR Subcomponent	Description	Reference standard
Process specification	BREAL business functions	Set of behaviors concerning the organization of knowledge, resources and skills. BREAL business functions are grouped in two main subsets: Development, Production and Deployment and Support	BREAL
	GSBPM phase	Set of activities describing the main steps of the statistical process. Main GSBPM phases: Specify Needs, Design, Build, Collect, Process, Analyse, Disseminate, Evaluate	GSBPM
	Process substep	Composing element of a process step, executing a specific subset of activities	
	Transformable input	Type of Process Input processed and transformed by a Process Step	GSIM
	Transformed output	Result of the process step execution	GSIM
	Support input	Type of Process Input supporting the Process Step execution, but not changed during data processing	GSIM
	Tools	Tools used for the process execution	
Methods	Process step Method	Description of techniques, statistical methods and algorithms used during step execution	GSIM
	Rules	Mathematical or logical expression to be assessed for defining specific behavior	GSIM
	Parameter Input	Type of Process Input specifying the set-up of a configurable Process Step	GSIM

The elements composing the Trust subset are detailed in the following figure. This group includes several dimensions related to process auditability, quality assessment, adoption of input privacy techniques and the provenance metadata derived from the main classes of PROV Ontology.

Figure 3.3: Main elements of the Trust subset



The subset of metadata, belonging to the Trust area and captured for the ML PoC are listed in the following table.

Table 3.11 – Metadata belonging to the Trust area captured for the ML PoC

MR Component	MR Subcomponent	Description	Reference standard
Traceability	Process design	Design time of a Process Step executed by a Business service	GSIM
	Process control	Specification of possible paths and decision criteria determining the actions to take after the execution of a Process step	GSIM
	Process changes	Inventory of process steps inconsistencies and main changes introduced to improve process execution and/or output	
Quality	Process metric	Type of Process Output containing auxiliary information about a Process step execution to provide a quality assessment of the Transformed Output	GSIM
Privacy	Input Assessment	Main privacy issues related to data acquisition, or a specific Process step (e.g. Data collection)	
	Consent Management	Specification of the process steps concerning the management of privacy consent and any following modification throughput the process	

MR Component	MR Subcomponent	Description	Reference standard
	Input Preserving techniques	Specification of the Privacy Preserving Techniques where applied	
Data provenance and lineage	Activity	Event occurring over a period of time and acting upon or with entities	PROV Ontology
	Agent	Specification of the entity responsible for an activity occurring, the existence of another entity, or for the activity of another agent	
	Entity	Any kind of thing (e.g., physical, digital or conceptual) having some fixed aspects	

3.4.3 Interaction between Smart metadata and the other building blocks

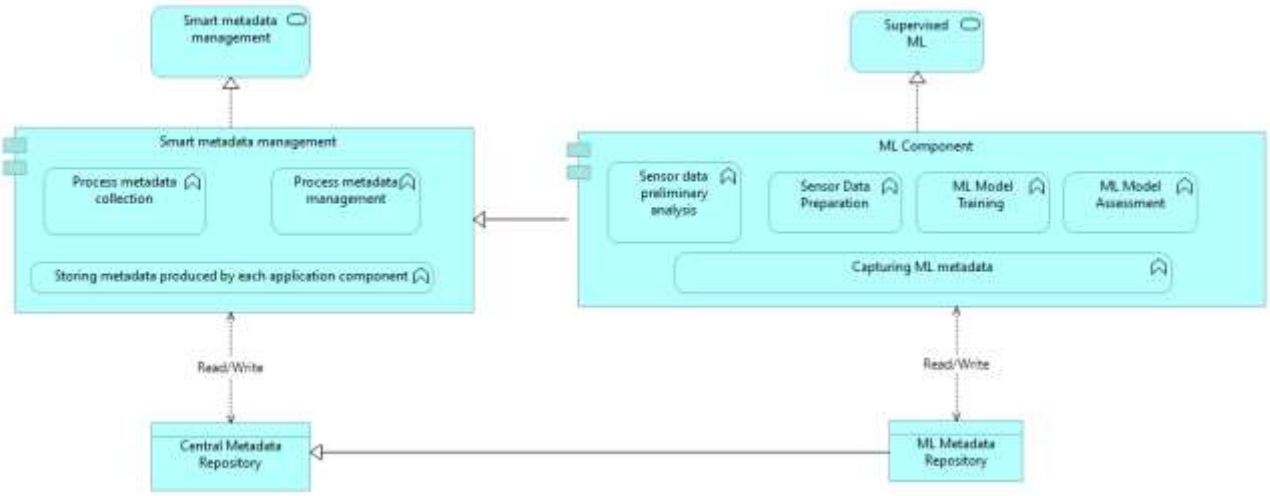
In order to document data transformations and the methods applied in the different process steps, each building block should provide a set of functionalities to:

- **Configure the process step:** read the metadata needed to configure the execution of the step, by specifying the Transformable input and the Transformed output, the methods to apply and the related rules, as well as required parameters (e.g. data structures, classifications, thresholds, model parameters, weights, etc.)
- **Store the metadata produced during process execution**, e.g. through specific quality indicators and metrics.

At the end of the process step execution, or after several iterations of a process step, the metadata produced during data processing can be stored locally, or made accessible to the other building blocks through the central building block for Smart metadata management, i.e. through the **Metadata Repository**. Thus, to run a procedure each building block may retrieve available metadata centrally stored in the MR.

The following figure shows the interaction between the Smart metadata component, described above in the architectural section, and the ML building block implemented in the Machine Learning PoC. The general assumption is that the centralized Smart metadata component should store and manage all the metadata produced during the different stages of ML data processing.

Figure 3.7: Interaction between the Smart metadata and the ML building blocks during the model training



The following table reports an example of the metadata produced or used by each Building block during the execution of the related tasks and stored in the centralized Smart metadata component.

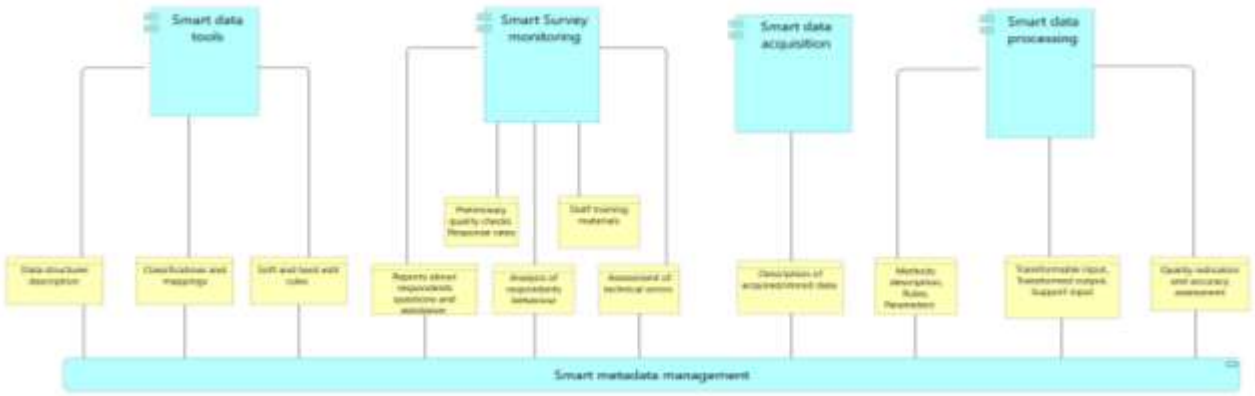
Table 3.12 - Metadata produced or used by each Building block

Building Block components	Metadata produced or used by Building Blocks during process execution	Metadata Repository Subsets
Smart data tools		
- Mobile app frontend (user experience and incentives) - Mobile app backend - In-device sensors - OCR data collection - Accelerometer data collection - GPS data collection	- Data structures description	Information: Structural metadata
	- Classifications and mappings	Information: Classifications & mappings
	- Soft and hard edit rules	Process: Methods
Smart survey monitoring		
- Monitoring dashboard	- Preliminary quality checks on collected data (missing values, detected errors) - Response rates	Trust: Quality

Building Block components	Metadata produced or used by Building Blocks during process execution	Metadata Repository Subsets
- Contact centre - Helpdesk website	- Reports concerning respondents questions and assistance	
- Training documentation - Training dashboard - Training classes	- Materials used for staff training (e.g.: slide, demos, videos)	Process: Tools
- Paradata - Contextual data	- Analysis of errors - Analysis of respondents behaviour - Assessment of data quality	Trust: Quality
Smart data acquisition		
- Structured or Unstructured data - NoSQL, Relational, Graph - Text files (CSV, XML, JSON)	- Description of acquired/stored data (e.g. File type, File path, File name)	Information: Structural metadata
Smart data processing		
- ML for OCR and Natural Language Processing - ML for Sensor data Processing - GPS data Processing	- Methods description, Rules, Parameters	Process: Methods
	- Transformable input, Transformed output, Support input	Process: Process specification
	- Quality indicators and accuracy assessment	Trust: Quality

The relationship between the metadata related to each Building Block and the Smart metadata building block is highlighted in the following ArchiMate model.

Figure 3.4: Relationship between metadata building block and each building block



References – Chapter 3

Bähr S., Haas Georg-C., Keusch F., Kreuter F., and Trappmann M. 2020. Missing Data and Other Measurement Quality Issues in Mobile Geolocation Sensor Data. Social Science Computer Review. DOI: 10.1177/0894439320944118

Booth B. M , Mundnich K. , Feng T., Nadarajan A., Falk T., Villatte J., Ferrara E., Narayanan S. 2019. *Multimodal Human and Environmental Sensing for Longitudinal Behavioral Studies in Naturalistic Settings: Framework for Sensor Selection, Deployment, and Management*. Journal of Medical Internet Research.

Immonen A., Pääkkönen P., and Ovaska E. 2015. Evaluating the Quality of Social Media Data in Big Data Architecture. Digital Object Identifier 10.1109/ACCESS.2015.2490723

Wolfewicz A. Human-in-the-loop in machine learning: What is it and how does it work?, available from: https://levity.ai/blog/human-in-the-loop#:~:text=Humans%20and%20machines%2C%20hand%20in,of%20a%20continuous%20feedback%20loop.

Deliverable 3.1 Aracri, Bruno et al.: Task 3.1.6: Metadata and Process auditability - Report on the Preliminary Framework, available from: https://ec.europa.eu/eurostat/cros/content/wp-3-conceptual-framework-european-platform_en

Deliverable 3.2 - Report on the Proof-of-Concept. Massimo De Cubellis , Fabrizio De Fausti, Claudia De Vitiis, Alessio Guandalini, Francesca Inglese, Mauro Bruno, Giuseppina Ruocco, Raffaella Maria Aracri, Nils Meise, Joeri van Etten, Franck Cotton

ESS Enterprise Architecture Reference Framework (EARF), available from: https://ec.europa.eu/eurostat/cros/content/ess-enterprise-architecture-reference-framework_en

Generic Statistical Business Process Model (GSBPM) v. 5.1. January (2019). Available from: https://statswiki.unece.org/display/GSBPM/GSBPM+v5.1

Generic Statistical Information Model (GSIM) v. 1.2 March (2021). Available from: https://statswiki.unece.org/display/gsim/GSIM+v1.2+documents

Sarker I. H. (2019). Context-aware rule learning from smartphone data: survey, challenges and future directions. Journal of Big Data. https://doi.org/10.1186/s40537-019-0258-4

Scannapieco M., Bogdanovits F., Gallois F.; Fischer, Kostadin G., Paulussen R., Quaresma S. et al.: BREAL. Big Data Reference Architecture and Layers. Application layer and Information layer. ESSnet on Big Data II, Work Package F, Deliverable F2 (2021). Available from: https://ec.europa.eu/eurostat/cros/system/files/wpf_deliverable_f2_breal_big_data_reference_architecture_and_layers_application_layer_and_information_layer_31_03_2021_final.pdf

Scannapieco M., Bogdanovits F., Gallois F.; Fischer, Paulussen R., Quaresma S. et al.: BREAL. Big Data Reference Architecture and Layers. Business layer. ESSnet on Big Data II, Work Package F, Deliverable F1. (2018-2021). Available from: https://ec.europa.eu/eurostat/cros/sites/crosportal/files/WPF_Deliverable_F1_BREAL_Big_Data_REference_Architecture_and_Layers_v.03012020.pdf

4. Technical requirements[9]

4.1. Introduction

The goal of this chapter is to prepare a list of technical requirements for different components used in smart surveys. This deliverable can be used in preparation a suggested framework for smart survey platform. In this document, the aim was to identify the list of technical components identified in previous deliverables of WP3 and find the connections between them. For instance, data storage is a common technical component shared by different aspects of smart surveys, i.e. machine learning, metadata or privacy preserving techniques. More detailed information on different aspects of technical components was shown in subchapter 4.2.

Because of complicated nature of smart surveys, it is quite difficult to suggest one unified set of software which should be used in every smart survey. It means that it is necessary to consider the use a variety of software (e.g. for data storage component both SQL, NoSQL as well as flat files can be used). However, in this chapter it was decided to provide the minimal set of requirements for the smart survey platform, according to the smart surveys applications that were described in previous WP3 deliverables. It considers aspects mostly related to the data storage, data transmission and general overview of possible applications of smart surveys.

Developing a smart survey platform is not a task on building the central repository but also a framework for all the applications sharing this repository. This is the reason why we included in this subchapter information on general requirements and possible data sources presented here as a set of possible sensors.

4.2. Technical components

According to the PoC and documents delivered in previous phases of the project, several technical components have been identified. They can be divided into three general parts, i.e. treatment of data (machine learning and quality evaluation), metadata and privacy preserving. According to the Table 4.1, several components are sharing the same set of technical components.

Table 4.3. General overview of technical components of smart surveys

Machine learning and data treatment	Metadata	Privacy preserving
• Server • Clients • Data sources (training, testing datasets) • Mobile device sensors	• Repository • Mechanisms for accessing this repository • Validation algorithms • Smart survey devices and applications	• Data source • Central server • Secure Computing Environment • Distributed Storage • Output Interface (GUI/API)

Machine learning is usually driven by various algorithms executed on servers. It includes data sources with training and testing datasets as well as mobile device sensors which generate the information

[9] Jacek Maślankowski (GUS)

processed by the algorithms. Therefore, the most important for machine learning is a repository accessible to conduct several iterations of machine learning algorithms.

Privacy preserving concerns different aspects of security components, in particular it includes secure computing environment and how to get to this data.

Metadata part of smart devices is very important from statistical point of view and was defined at a very detailed level. In more detailed form it includes the following components listed in Table 4.2.

Table 4.4. Technical components of smart surveys metadata aspects

General components	Smart services	Different scenarios and services	Infrastructural components
1. Smart data device 2. Smart data source 3. Smart data storage 4. Smart data monitoring systems 5. Smart data linking and enriching 6. Survey questionnaire source 7. Data storage 8. Data processing 9. Data security and confidentiality 10. Data acquisition (passive or active) 11. Metadata repository 12. Data and metadata accessing 13. Data and metadata validation 14. Data preparation, filtering and deduplication 15. Data encoding 16. Data standardization 17. Data modeling techniques	1. Smart metadata management 2. Smart data tools 3. Smart data monitoring system 4. Smart data acquisition 5. Smart data processing 6. Input Privacy Preserving Techniques	1. Autonomous services, developed at local level in NSI's environment, without harmonisation between countries 2. Interoperable services, having a similar service interface across the NSIs, and different back-end implementations 3. Replicated services, if the same application service is delivered to different NSIs 4. Shared services, realized through application services available in the platform and accessed by all NSIs	1. Local platforms implemented and managed by NSIs 2. Local platform of external data providers accessed by NSIs to gather processed data 3. Shared platform, developed by a third party to provide shared statistical services to perform smart data processing

The most important aspect of metadata related to the possibility of creating future smart surveys platform are considering three infrastructural scenarios on how to deploy the solution. For instance, a smart survey platform can be implemented and managed by NSIs. The second and third option is to

use external providers to gather processed data (e.g. data center in the cloud) or third party enterprises to provide shared statistical services to e.g. process data gathered from smart survey applications. Metadata also includes quality indicators, i.e. metrics' design, metrics' organization structure, metrics acquisition, metrics selection and metrics integration and quality evaluation.

All the components presented above, considering metadata, privacy preserving and machine learning have general common references. In other words, there are several components that can be shared by all dimensions of smart surveys. It includes:

- Data storage,
- Data transmission type,
- Server, which can be deployed in the cloud or in the NSIs,
- Client software, i.e. mobile device on which smart survey is executed.

General requirements for this type of components have been provided in parts 3 and 4 of this document. Additional description of potential sensors (e.g., used by machine learning and metadata components) was described in part 4.5.

4.3. General requirements for the smart surveys platform

There are several requirements that must be considered when planning the infrastructure for smart surveys. As mentioned in the introduction to this chapter, it is not possible to identify one unified solution which allows to gather all types of information in one place. The aim is to provide an agile solution that can be scaled and adapted to changing requirements of new implemented smart surveys.

In this sense, the requirements include different groups, of which the most important are those which consider data storage and mobile applications. Mobile applications are mostly based on set of sensors, of which the most important is GPS. Another important aspect is to prepare a set of interfaces that is used as a secure channel between mobile applications and central server, mostly used for data storage. Then the data from data storage can be processed to provide set of tables used for analysis. It includes the most general that is presented in Table 4.3.

Table 4.5. General requirements for Smart Surveys

Id	Name of component	Description of the requirement
1.1.	Data storage	All processed data should be stored in structured format in database or machine-readable files. Raw data can be stored in JSON-like or YAML files. It is recommended to pre-process the data at smart survey device to provide relevant data only.
1.2.	Sensors	Application should gather most of the data from sensors available in devices. More sensors are listed in 4.1.1 part (smart device tools, smart device sensors) of this subchapter.

Id	Name of component	Description of the requirement
1.2.1.	Sensor GPS	Application should locate user based on GPS coordinates.
1.2.2.	Microphone	Application should locate the environment, whether a person is inside or outside the building. This information can be used to validate the data.
…	More listed in subchapter 4.1.1. (smart device tools, smart device sensors)	…
1.3.	API	The application should transmit the data with standards, i.e. pass through JSON, SDMX etc.
1.4.	Devices	The smart survey should use the most common devices, such as mobile phones rather than computers.
1.5.	Platforms	The application should be available to all popular platforms of devices, e.g. Google Android, Apple iOS.
1.6.	Alternative versions	Respondents who are in the sample frame and does not have a device needed by smart surveys should have a possibility to use alternative format, e.g. paperback questionnaire.
2.1.	User interaction	Smart survey (e.g. mobile application) should gather as many data as possible without user interaction.
2.2.	Performance	The application should gather the data with minimum use of mobile phone/device resources, such as memory or processor.
2.3.	Security	The application should be secure, and no sensitive data should be accessible by third parties.

The list shown in Table 3 covers general requirements for the platform that can be used as smart survey platform. As mentioned earlier, because of complicated nature of smart surveys, it is not possible to create one platform that can be shared by all smart surveys' applications. However, it is possible to identify specific technical components that can be shared by various applications. For

example, NoSQL database is very flexible and can be shared by most of smart surveys application. On the other hand, it will be much difficult to maintain than traditional relational database. Because of flexibility of NoSQL database, some data may not be consistent.

4.4. Technical requirements for components

According to Table 2.1 in the first part of Deliverable 3.3, the following technical components can be identified:

- Smart data tools, including mobile app and in-device sensors
- Smart surveys monitoring, including data analysis
- Smart data acquisition, including data storage and acquisition type (active/passive)
- Smart data processing, including analysis, validation and linking

They act also as common points in different architectural scenarios. In this part of the deliverable, three different, i.e. smart data tools, data storage and data transmission was characterized.

4.4.1. Smart Data tools – in device sensors

According to the specification in deliverable 3.2 and 3.1, applications used accelerometer and GPS data for data collection. However, in HBS it is also possible to collect the data with the use of camera to take photos that will be scanned later. In this example, it is necessary to conduct most of data processing at a device and transmit only necessary data to the central data storage server.

The goal of this subchapter is to list in-device sensors that can be used to provide data to smart survey hub. Additionally, some challenges in using various sensors were also listed as a remarks. Some of the data based on sensors can also act as paradata (e.g., battery level, state of network connections – see more in chapter 3.1.3.3).

In general, most of the data can be collected using typical sensors, listed below:

- Microphone,
- GPS (coordinates, altimeter),
- Accelerometer,
- Proximity Sensor,
- Light Sensor,
- Touchscreen sensors,
- Gyroscope,
- Magnetometer,
- Heart Rate Sensor (mostly on wearable devices),
- Pedometer,
- Barometer.

Not all sensors were tested in smart surveys considered by work package 3. However, it was possible to provide a short description on the possible applications of such sensors. Table 4.4 shows the characteristics of the use of different sensors, required devices and expected results.

Table 4.6. Possible sensor usage and the requirements for resources

No.		Smartphones	Tablets	Wearables (smartwatches,	Possible data	Additional methods for processing	Remarks
1.	Microphone	X	X	O	Environment (e.g. forest, inside the building, street)	Machine learning, voice recognition	Battery fast consuming, bandwidth high consumption.
2.	GPS	X	O	O	Coordinates, altimeter	Geotagging (cities by coordinates)	Privacy issues
3.	Accelerometer	X	O	X	Whether a person is driving, walking etc.	Machine learning, pattern recognition	Similar data can be extracted by GPS sensor
4.	Proximity Sensor	X	X	X	Person in selected locations	Machine learning	Similar data can be used with GPS
5.	Light Sensor	O	O	O	Whether a person is outside or inside the building	Machine learning	Must be linked to additional data, e.g. with GPS data, then the precision of identifying the place will be more accurate
6.	Gyroscope	X	O	O	User behaviour – example in [Zahra et al., 2021]	Machine learning	User behaviour can also be extracted from the GPS sensor

No.		Smartphones	Tablets	Wearables (smartwatches,	Possible data	Additional methods for processing	Remarks
7.	Magnetometer	O	O	O	User behaviour – example in [Ligorio et al., 2020]	Machine learning	Can supplement the data gathered from GPS sensor
8.	Heart Rate Sensor	O	-	X	Whether a person is doing some activities, e.g., walking, sitting in the car	Data acquisition, descriptive statistics	Can be treated as a sensitive data, however with link to GPS data it will improve the quality – whether a person is in a car in a traffic jam or walking
9.	Pedometer	X	O	X	Monitoring physical activity, e.g. like in a example [Fortune et al., 2020]	Machine learning	Can be treated as supplementary data do have more accurate data from GPS
10	Barometer	O	O	O	Can only be used to give more accurate results from GPS sensors	Data acquisition, comparing with coordinates	Can supplement GPS data

Legend: X – available, O – optional, - – not available

Table 4.4 shows general assumptions for different sensors. In columns 3-5: (O) - means optional, (X) – means typical, (-) means that in most devices this sensor may not be included. All the devices should at least pre-process the data before delivering to the central data storage. It will not be possible with advanced algorithms, such as machine learning but in many cases the first data processing should be executed in device.

4.4.2. Smart Data acquisition – data storage

Table 4.5 shows the general storage solutions for the smart surveys. However, it is not limited to the ones presented below. In most cases, three different data storage techniques can be used to fulfill most of user requirements.

For example, if the data is based on sensors usually this information can be delivered in JSON, YAML or XML files. Therefore, the suggested repository seems to be NoSQL. However, for further processing the information it is important to migrate this repository into the relational database, which is much efficient in structured data analysis. Table 4.5 shows the prerequisites and requirements for the use of different technical storage solutions.

Table 4.7. Data storage prerequisites and requirements

Starting conditions	**Technical storage solutions**		
	NoSQL	**SQL**	**Text file**
Type of data storage Structured/Unstructured data	NoSQL	SQL	Text files (CSV, XML, JSON)
Possible data processing Database queries / scripts in R/Python etc.	In database, in application (NoSQL queries)	In database, in application (SQL queries)	Any script languages (set of anonymized data files)
Benefits Scalability	More scalable and easy to maintain	Limited scalability	More scalable but not easy to maintain
Security issues Users / groups etc.	Flexible – typical with no user accounts, possibility of creating user accounts	In database authentication, easy to manage	Operating systems security (file permissions), not easy to manage
Type of service deployment Cloud / centralized / cluster of servers	Cloud / centralized / in-database clustering, master-slave	Usually standalone server, can be in a cloud	In a cloud, on a server, cluster of servers
Additional methods ML etc.	Yes, possible aggregations, text mining	Yes, limited data mining	No, should be calculated outside the environment

Planning a data storage is not only to consider the technology which is in line with the smart surveys. It is also important to think about the resources in the NSI. For example, who will use this repository and what are the skills of data scientists who are using these components. As mentioned in earlier parts, even NoSQL is very flexible, it may not be the best environment to make efficient analysis of the data. Most of the datalabs are still using the relational-like databases accessing them by the well-known language which is SQL.

4.4.3. Smart Data acquisition – data transmission

Similar issues may be identified with respect to the data transmission. Even the most expected way – required for smart surveys – would be to have direct transmission, in many cases the way the transmission will be triggered depends on the user devices and their mobile or Internet connection. Three different ways of acquiring the data from users based on different requirements was presented in Table 4.6.

Table 4.8. Data transmission suggested methods

Starting conditions	**Technical transmission solutions**		
	Direct	**Bulk**	**Async**
Type of data transmission	Direct transmission triggered by the activity	Bulk transmission	Transmission at request
Possible data processing	Central processing	Processing on device, sending pre-processed aggregated data	Processing on device
Benefits/obstacles	Less consumption of device resources, more consumption of mobile networks	More secure, less consumption of 3G/LTE/5G data transmission (e.g., transmitted only on WiFi networks), more consumption	More secure, less consumption of 3G/LTE/5G data transmission
Security issues	User is monitored all the time, e.g. when is outside, inside the building – privacy issues may be an obstacle	Not possible to track users in real time	User can send information several days after being registered
Type of service deployment	At centralized platform, thin client on a device	Thick client at device, only aggregated data	Thick client at device, consumption of large memory

		comes to centralized platform	(depending on the sensors used)
Additional methods	Machine learning on the platform	Machine learning on the device	Machine learning on the device

According to table 6, the method of data transmission will be driven by the way the central platform will be implemented. If the recommended method will be to pre-process the data at device, the data transmission should be bulk or asynchronous. All methods presented in table 6 have some advantages and disadvantages. Therefore, it can be necessary to mix them when implementing smart surveys. Description in this chapter includes the most general components. According to the requirements of the smart surveys, relevant technical components include smart data processing and smart data monitoring, described more in subchapter 3.1.3.

4.5. Components of the smart surveys platform

Considering the central platform at Eurostat level, we need to consider different scenarios of data analysis, as described in part 4.4. Typical smart surveys platform should include the following core components:

- data storage, i.e. relational, NoSQL database or flat files,
- data access components, i.e. API or direct access,
- datalab, e.g. for data scientists used for data processing and analysis,
- network security components (e.g., cryptography, secure protocols).

Although several different devices gave a possibility of using them in various type of data analysis, the results received from different smart surveys shows we need additional processing. It was proved in several research papers, showing different sensors used, e.g. [Zahra et al., 2021], [Ligorio et al., 2020]. Some of the requirements may not be possible to fulfill in the smart surveys, e.g., the use of heart sensor data, because of sensitive information. In many cases there is still a possibility of using them by processing on the device and delivering a processed and anonymized information to the data storage.

References – Chapter 4

Zahraa Hashim Kareem, Khairun Nidzam bin Ramli, Rami Qays Malik, Musddak M. Abdul Zahra, Mobile phone user behavior's recognition using gyroscope sensor and ML algorithms, Materials Today: Proceedings, 2021, ISSN 2214-7853, https://doi.org/10.1016/j.matpr.2021.04.639.

G. Ligorio et al., "A Wearable Magnetometer-Free Motion Capture System: Innovative Solutions for Real-World Applications," in IEEE Sensors Journal, vol. 20, no. 15, pp. 8844-8857, 1 Aug.1, 2020, doi: 10.1109/JSEN.2020.2983695.

Fortune J, Norris M, Stennett A, Kilbride C, Lavelle G, Victor C, De Souza L, Hendrie W, Ryan J. Pedometers, the frustrating motivators: a qualitative investigation of users' experiences of the

Yamax SW-200 among people with multiple sclerosis. Disabil Rehabil. 2020 Jun 9:1-7. doi: 10.1080/09638288.2020.1770344.

Documentation from ESSnet on Smart Surveys: 3.1.2 Technical infrastructure. “WP 3 subtask 3.1.2 on Technical infrastructure”.

Documentation of HBS Smart Survey, 2021, available online.

Documentation of MOTUS Smart Survey, 2021, available online.

5. Enhanced framework at work[10]

This section provides an overview of the interaction between the TSSu platform and NSIs infrastructure, to analyse the deployment of the software solutions described in the previous chapter.

The design of the main building blocks for smart data acquisition and processing allows to identify how to deploy the implemented solutions, taking into account the following dimensions:

- Type of sensors: several pipelines for smart data acquisition and processing should be designed, depending on the sensors used for data collection (in-device, external sensors) and the type of information to gather (e.g.: images, accelerometer data, GPS signal)
- Type of data acquisition: application services for smart data gathering must be customized and managed, depending on passive or active data acquisition
- Type of data provider: service deployment may vary also according to the type of data provider, either the survey respondent, or third parties
- Data Processing and Data storage environments: smart data can be stored and processed in several environments, such as In-app, in NSI's premises, in the TSSu Platform, or in infrastructures owned by Third parties. In compliance with best practices, data should be processed at rest i.e., in the environment where data is stored. In case of data acquisition from third parties, or integration between several stakeholders, Input privacy techniques should be used to guarantee privacy preservation.

The following analysis focuses on the deployment of TSSu building blocks, and is based on BREAL operational model, and ESS EARF standard (ESS Enterprise Architecture Reference Framework). According to these reference frameworks, application services can be classified as follows:

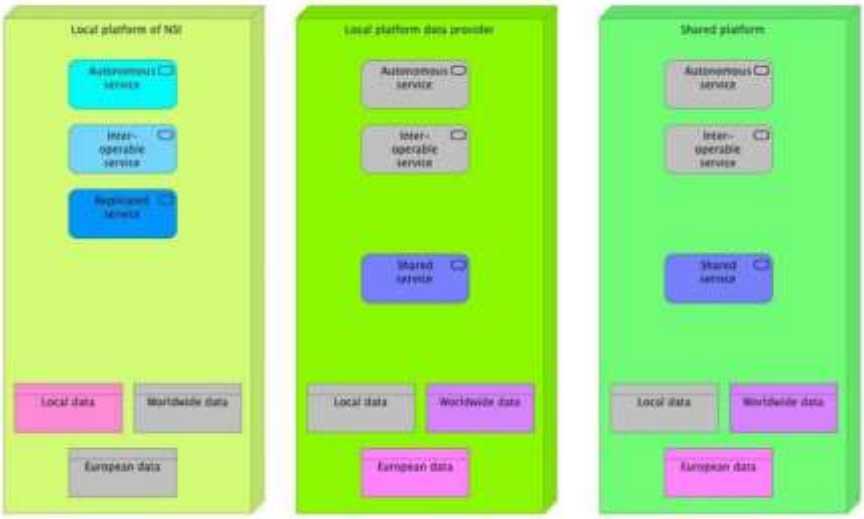
- Autonomous services, developed at local level in NSI's environment, without harmonisation between countries
- Interoperable services, having a similar service interface across the NSIs, and different back-end implementations
- Replicated services, if the same application service is delivered to different NSIs
- Shared services, realized through application services available in the platform and accessed by all NSIs.

Depending on the location and owner, the infrastructures and platforms for data hosting and management can be grouped in:

- Local platforms implemented and managed by NSIs
- Local platform of external data providers accessed by NSIs to gather processed data
- Shared platform, developed by a third party to provide shared statistical services to perform smart data processing.

[10] Mauro Bruno and Giuseppina Ruocco (ISTAT)

Figure 5.1: Operational model proposed by BREAL



Starting from BREAL operational model, the following paragraph describes different use cases, to assess pros and cons related to the deployment of the developed solutions.

The main goal of the user stories described below is to analyse potential operational models, to improve the theoretical analysis and provide examples of service deployment in compliance with the main dimensions explored so far, mainly: **privacy issues**, **quality assessment** and **metadata management**.

A - Active data acquisition through in-device sensors

In the first use case, a National Statistical Institute accesses the TSSu platform to run a smart survey and collect data through a mobile app. According to the data collection strategy, the type of sensors and the smart data source, each NSI may download from the platform a set of software components. For example, the NSI could download and install in-house the following software solutions:

- **Smart data collection tools,** e.g., mobile app frontend, survey questionnaire design (variables, classifications, hard/soft checks, questions sequence and flow, etc.)
- **Mobile app backend,** e.g., a set of rest APIs that allow to store app data in a central repository and manage active data acquisition
- **Monitoring system,** e.g., a web application that offers a set of reports based on the analysis of the data stored in the Mobile app backend. The monitoring dashboards could, for example, monitor respondents' participation and quality of collected data.

These software solutions provided by the TSSu platform, must be configured to meet specific national requirements (e.g., national regulations). More in detail, the tasks performed by the NSI could be the following:

- Implementation of custom reports in the fieldwork monitoring system, i.e., the monitoring system could provide drag and drop functionalities to generate new reports
- Configuration of the mobile app frontend (e.g. set-up of survey questionnaire, translation of materials, specification of national classifications and codelists, etc.).

- Process metadata and quality indicators related to collected data could be stored in the Mobile app backend or in the Monitoring system repository locally installed in NSI's infrastructure.

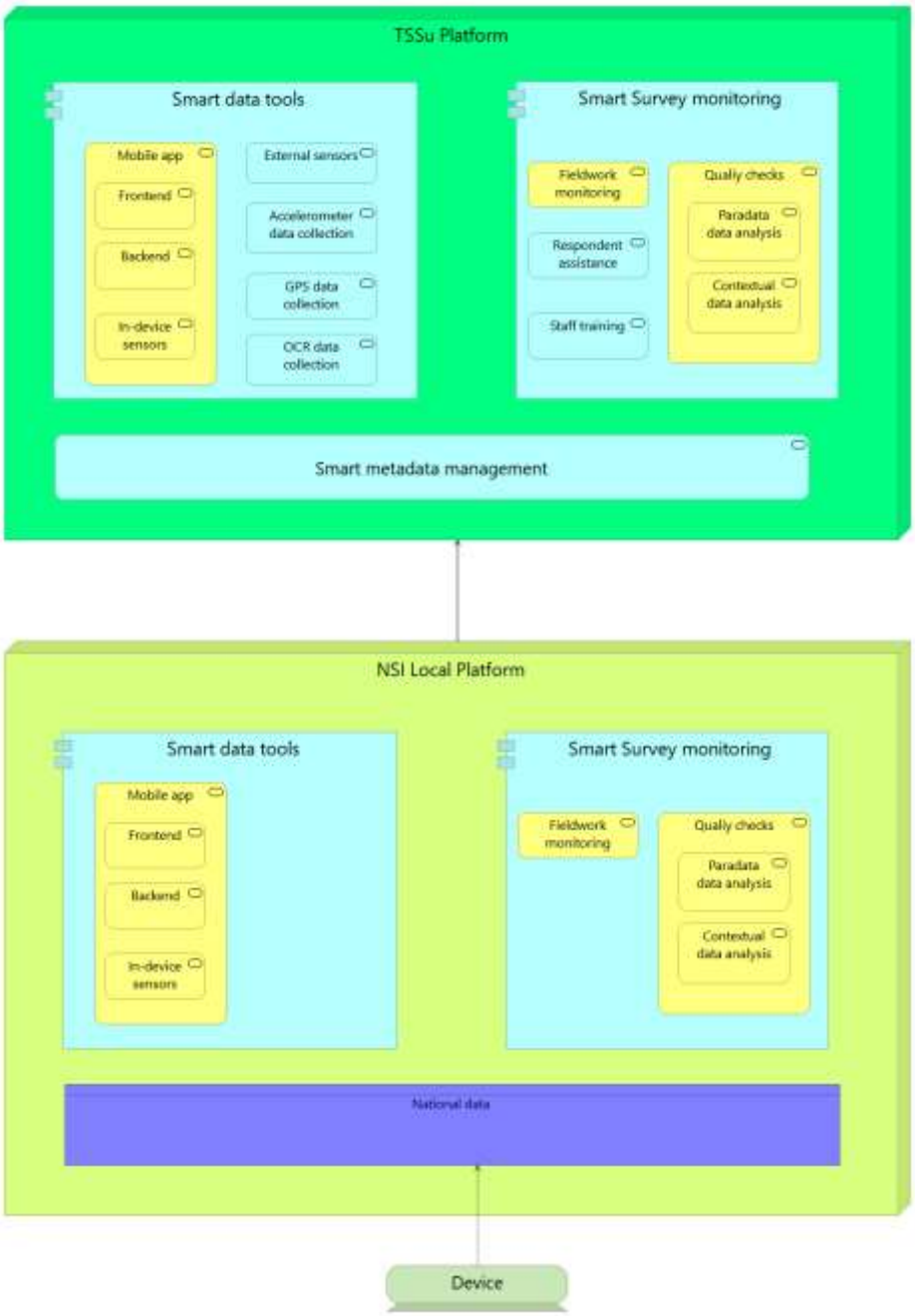
The table below summarizes the starting assumptions of the first user story and provides an overview of the main features of the context.

Table 5.1 – Active data acquisition through in-device sensors

Starting assumptions	Context description
Type of sensors	**In-device**
Type of data acquisition	**Active**
Type of data provider	Respondent
Data Processing	In-app/NSI premises
Data storage	In-app/NSI premises
Service deployment	The platform may offer software components to be installed and configured at a national (e.g., to develop a national version of the App and monitor the data collection)

The interaction between the TSSu platform and the NSI is sketched in the following model:

Figure 5.2: Interaction between the TSSu platform and the NSI - Active data acquisition through in-device sensors



B - ML training models through Input privacy preserving techniques

In the second user story, assuming that smart data are stored in-house, or even during data gathering, NSIs can use the services offered by the TSSu platform to process collected data and train a ML model without sharing data. The TSSu platform may provide shared services and a distributed environment to execute ML training models by applying Input privacy preserving techniques, without sharing national data.

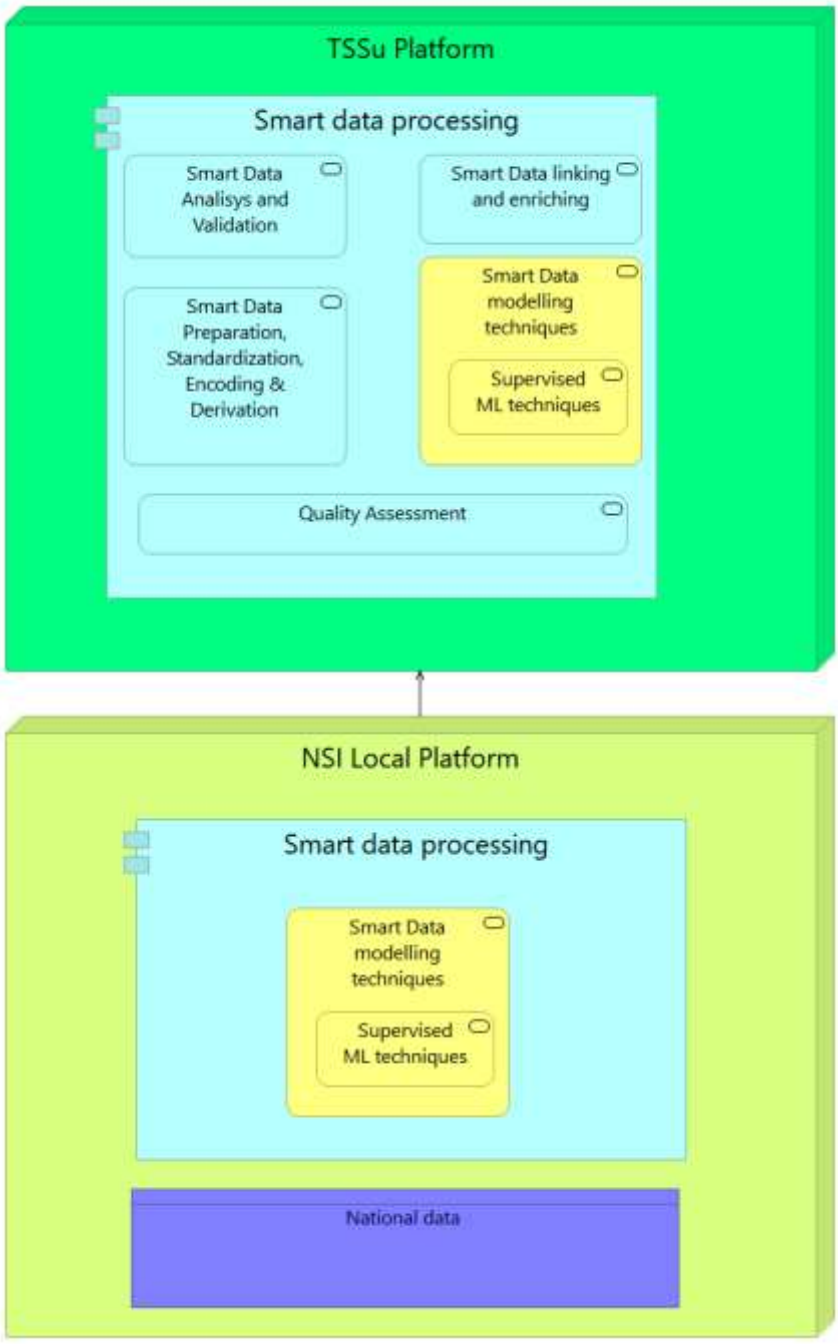
The following table summarizes the starting assumptions of the second user story and provides an overview of the main features of the context.

Table 5.2 – ML training models through Input privacy preserving techniques

Starting assumptions	Context description
Type of sensors	**In-device/External sensors**
Type of data acquisition	**Active/Passive**
Type of data provider	Respondent/Third parties
Data Processing	NSI premises
Data storage	NSI premises
Service deployment	The platform may provide replicated services to be installed and configured at national level to train ML models

The interaction between the TSSu platform and the NSI is sketched in the following model:

Figure 5.3: Interaction between the TSSu platform and the NSI - ML training models through Input privacy preserving techniques

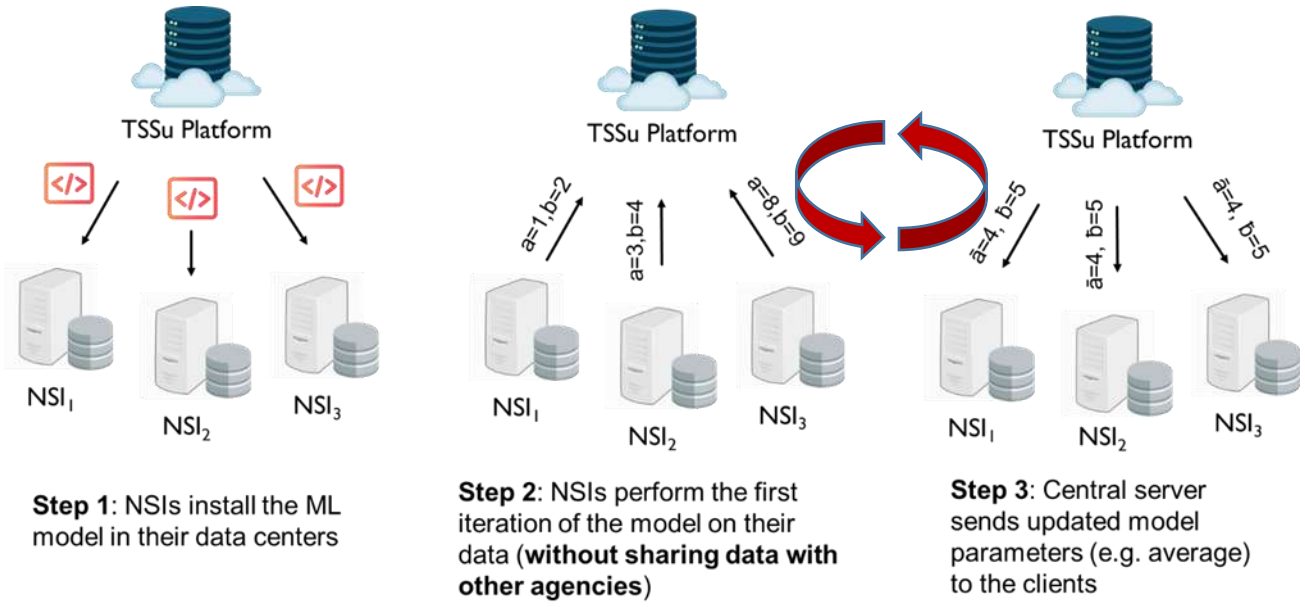


The main steps performed to combine ML training and federated learning can be summarized as follows:

- Step 1: NSIs install the ML model in their data centers through a replicated service available in the TSSu platform
- Step 2: NSIs execute a first iteration of the model without sharing data with other agencies and send the output to the central server
- Step 3: Central server collects ML results from several NSIs and sends the updated model parameters (e.g. average) to the national clients.

This approach, summarised in the following figure, allows to train the ML model on all the datasets available at European level. Each NSI can apply on national data the model trained, thus improving the accuracy of the model.

Figure 5.4: Main steps of ML training through federated learning techniques



References - Chapter 5

ESS Enterprise Architecture Reference Framework (EARF), available from: https://ec.europa.eu/eurostat/cros/content/ess-enterprise-architecture-reference-framework_en

Scannapieco M., Bogdanovits F., Gallois F.; Fischer, Kostadin G., Paulussen R., Quaresma S. et al.: BREAL. Big Data Reference Architecture and Layers. Application layer and Information layer. ESSnet on Big Data II, Work Package F, Deliverable F2 (2021)

Annex: Architectural scenarios[11]

Starting from BREAL operational model, the following scenarios describe the interaction between the TSSu platform and NSIs infrastructure. These solutions result from the combination of the following aspects:

- ✓ Type of data acquisition (Active/Passive data), depending on the sensor used to collect data
- ✓ Type of data provider (Respondent/Third parties)
- ✓ Type of data storage/management (Local (on device)/ In-house (NSI)/Centralized (Platform) data storage)
- ✓ Type of data processing (Local (In-app)/ In-house (NSI)/Centralized (Platform) data processing)
- ✓ Type of service deployment (NSI/Centralized (Platform)).

The proposed architectural scenarios are conceived to highlight pros and cons of each solution, to model a target architecture and specify:

- The services to be implemented, their deployment and the execution environment
- Whether data are stored and processed locally, or on a kind of centralized platform.

Scenario 1: Passive data acquired through shared or replicated services and processed in the TSS_u platform

Type of data acquisition	Passive
Type of data provider	Respondent
Data Processing	Platform (Smart data processing)
Data storage/management environment	NSI/Platform
Service deployment	Shared/Replicated: NSIs may choose whether to run the services offered by the platform locally (replicated services) or on platform infrastructure, depending on available national capabilities and skills

[11] Mauro Bruno and Giuseppina Ruocco (ISTAT)

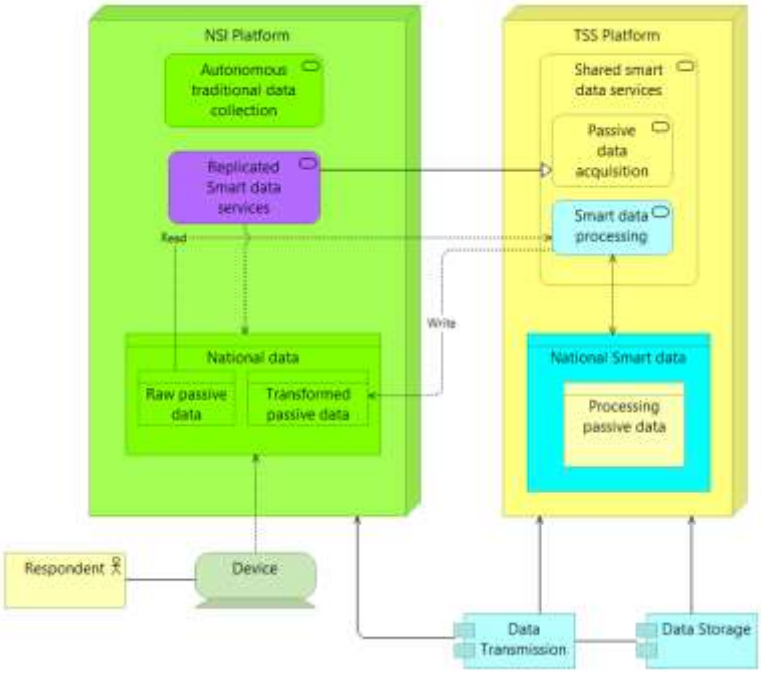


Figure A1. Service deployment in scenario 1

Scenario 2: Passive data acquired through shared or replicated services and processed in the NSI's infrastructure

Type of data acquisition	Passive
Type of data provider	Respondent
Data Processing	NSI
Data storage/management environment	NSI
Service deployment	NSIs may choose to run the services offered by the platform locally (replicated services)

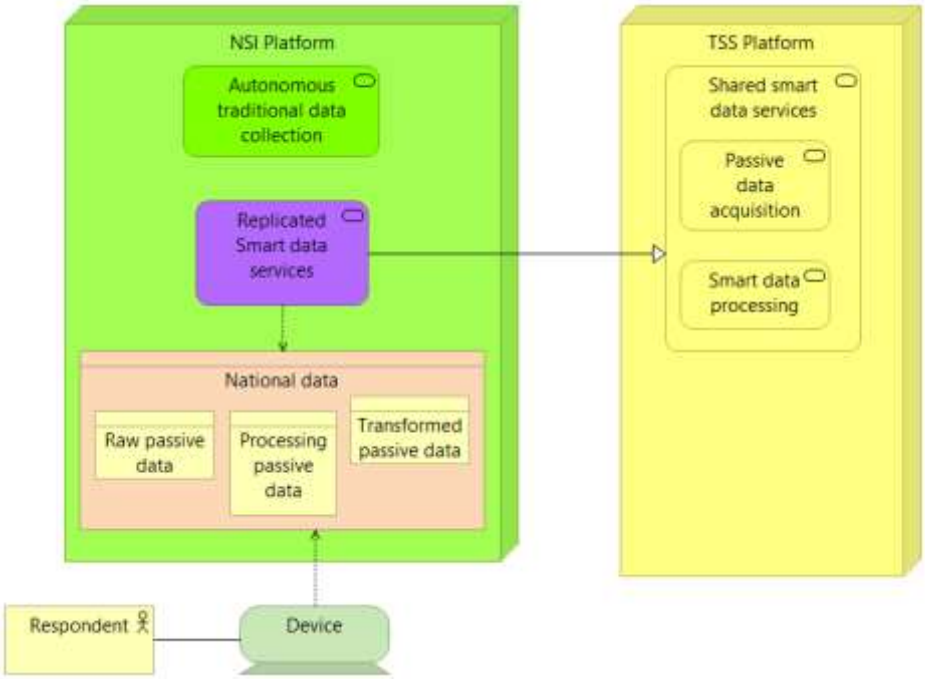


Figure A2. Service deployment in scenario 2

Scenario 3: Active data acquired through interoperable services and processed through shared services in the TSS_u platform

Type of data acquisition	Active
Type of data provider	Respondent
Data Processing	In-app /NSI/Platform
Data storage/management environment	NSI/Platform (Smart data processing)
Service deployment	Shared/Interoperable/ Replicated: The platform may offer: 1) interoperable services to configure a national version of the App to be implemented and monitor the data collection; 2) centralized services to process smart data, stored in national repositories and accessed through Privacy Preserving Techniques. The results are stored in national repositories

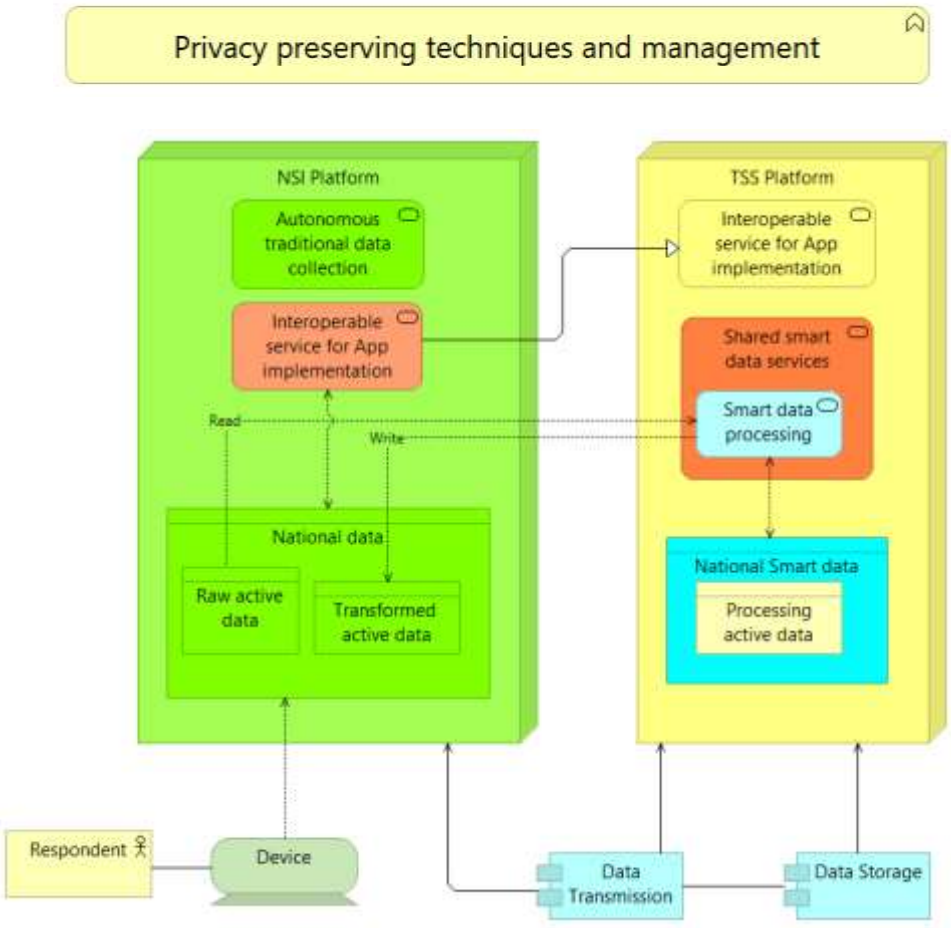


Figure A3. Service deployment in scenario 3

Scenario 4: Active data acquired through interoperable services and processed through shared or replicated services in the NSI's infrastructure

Type of data acquisition	Active
Type of data provider	Respondent
Data Processing	In-app /NSI
Data storage/management environment	NSI
Service deployment	Shared/Interoperable/ Replicated: The platform may offer: 1) interoperable services to configure a national version of the App to be implemented and monitor the data collection; 2) centralized services to process smart data, stored in national repositories and accessed through Privacy Preserving Techniques. The results are stored in national repositories

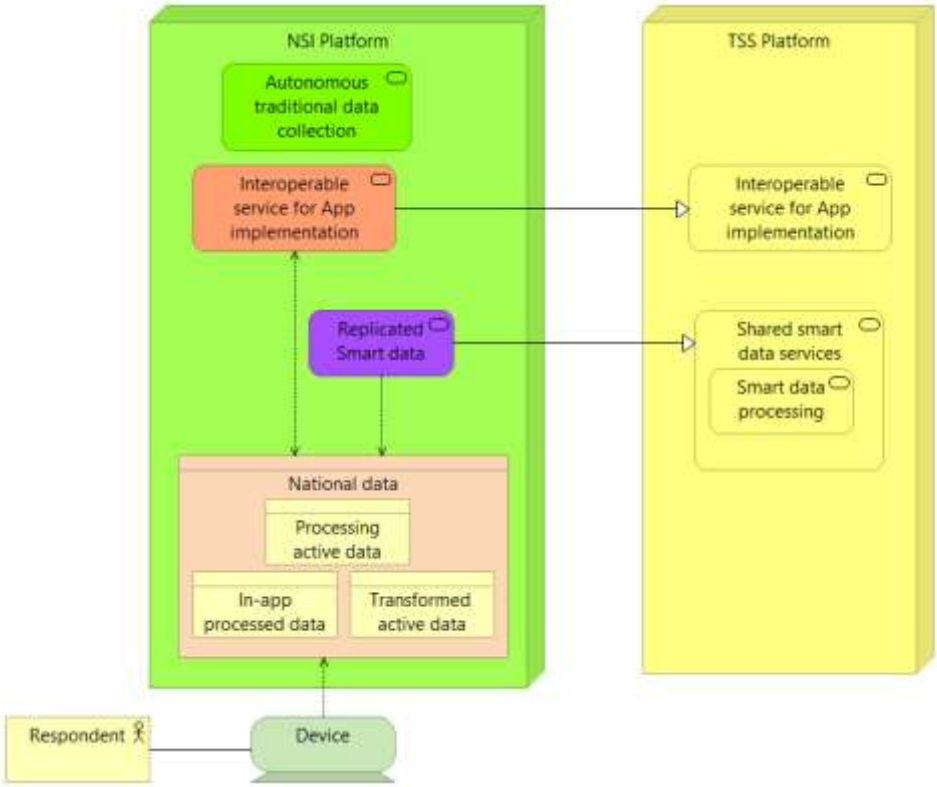


Figure A4. Service deployment in scenario 4

Scenario 5: Passive data provided by Third parties and acquired through replicated services in the TSSu platform. Active data acquired through interoperable services and processed through shared or replicated services in the NSI's infrastructure

Type of data acquisition	Active/Passive
Type of data provider	Respondent /Third parties
Data Processing	In-app/NSI/ Third parties
Data storage/management environment	In-app/NSI/ Third parties
Service deployment	Interoperable/Shared/Replicated: 1) NSIs may access, download and execute locally the services available in the platform to acquire and process third-parties data; 2) interoperable services allow to customize data collection tools and activities

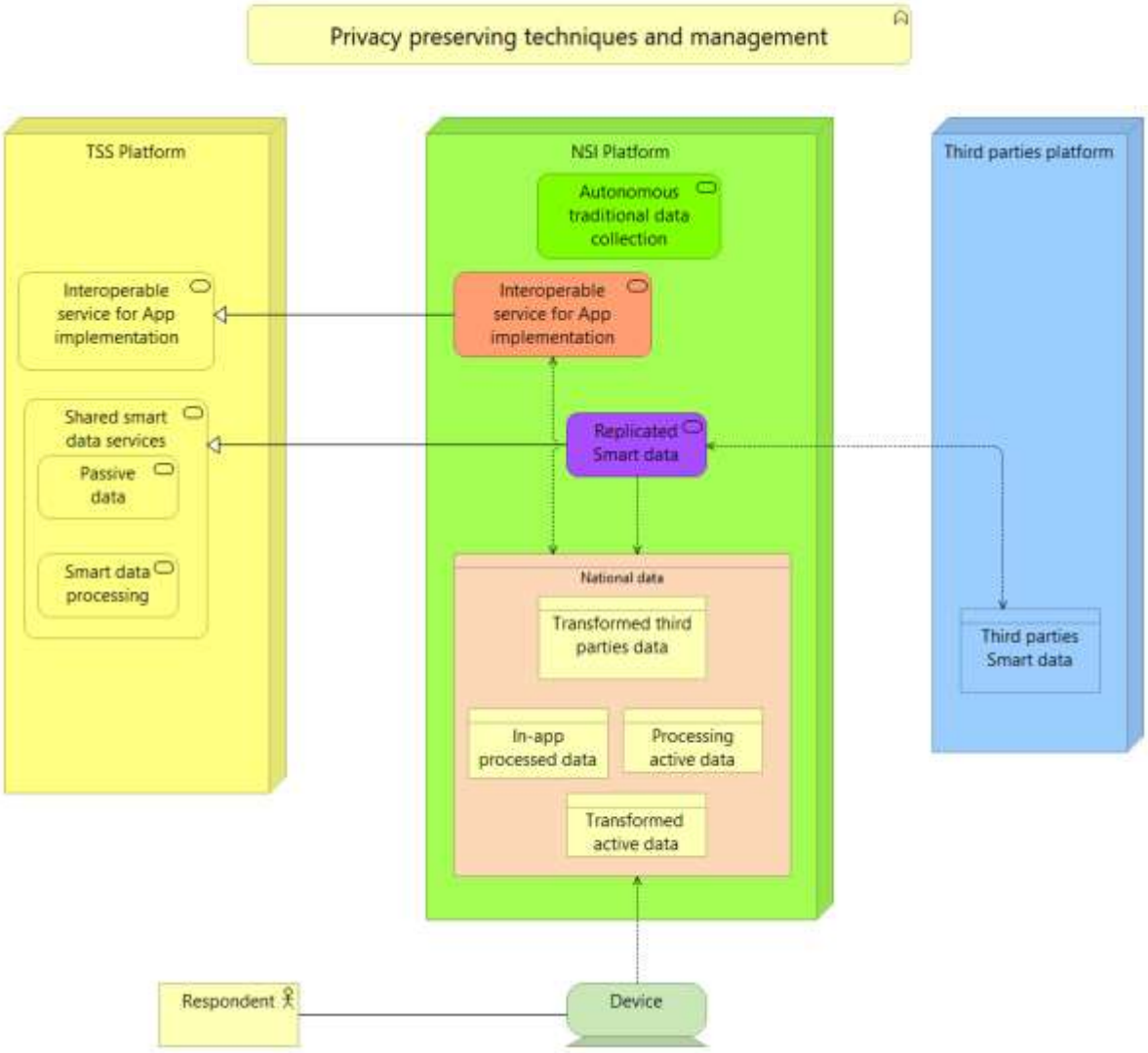


Figure A5. Service deployment in scenario 5