


# D2.1 - TRUMAN USE-CASE DEFINITIONS

| | |
|---|---|
| Work Package | WP2 - Use Case Definition and Elicitation of Requirements |
| Lead Author | Savino Dambra (GEN) |
| Contributing Author(s) | Savino Dambra (GEN), Ivano Bison (UniTn), Vilma Scarpino (DOXA), Ege Bilaloğlu (YKT), Adamantios Ntanis (PDN), Thodoris Vlioras (PDN), Spyridon Kontaxis (PDN), Efstathios Kontogiannis (PDN) |
| Due date | 31.03.2026 |
| Date | 31.03.2026 |
| Version | 1.0 |
| Dissemination Level | PU (Public) |



Funded by
the European Union

**The research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation Programme, through the TRUMAN project, under Grant Agreement No. 101214000. The content and results of this deliverable reflect the view of the consortium only.**

# Revision History

| Revision | Date | Editor | Notes |
|---|---|---|---|
| 0.1 | 20.11.2025 | Savino Dambra (GEN) | First complete version of the document is created. |
| 0.2 | 15.12.2025 | Savino Dambra (GEN) | Input for UC2 |
| 0.3 | 07.01.2026 | Ivano Bison (UniTn), Vilma Scarpino (DOXA) | Input for UC1 |
| 0.4 | 30.01.2026 | Ege Bilaloğlu (YKT) | Input for UC3 |
| 0.5 | 02.02.2026 | Adamantios Ntanis (PDN), Thodoris Vlioras (PDN), Spyridon Kontaxis (PDN), Efstathios Kontogiannis (PDN) | Input for UC4 |
| 0.6 | 19.02.2026 | Andrea Bontempelli (UniTn) | First review |
| 0.7 | 23.02.2026 | Savino Dambra (GEN) | Fix first review comments |
| 0.8 | 25.02.2026 | Melek Önen (EURC) | Second Review |
| 0.9 | | Savino Dambra (GEN), Ivano Bison (UniTn), Vilma Scarpino (DOXA), Ege Bilaloğlu (YKT), Adamantios Ntanis (PDN), Thodoris Vlioras (PDN), Spyridon Kontaxis (PDN), Efstathios Kontogiannis (PDN) | Fix second review comments |
| 0.95 | 23.03.2026 | Melek Önen (EURC) | Third review |
| 0.96 | 26.03.2026 | Melek Önen (EURC) | Quality check |
| 1.0 | 31.03.2026 | Savino Dambra (GEN) | Final version. |

## Table of Contents

Executive Summary ..... 6

Glossary of Terms ..... 7

1 Introduction ..... 9

1.1 Purpose of the document ..... 9

1.2 Outline ..... 11

2 UC1 – Households' daily lives and lifestyles ..... 12

2.1 Context and Motivation ..... 12

2.2 Actors and Roles ..... 14

2.2.1 Field supervisors as Mode Trainer ..... 14
2.2.2 Web panel managers as Model Executors ..... 14
2.2.3 Survey participants (the sample) as Data Owners ..... 14
2.2.4 Smartphone and DTraces app as Data Owners ..... 14
2.2.5 Customers: The Researcher as End-Users ..... 15
2.3 Data Description ..... 16
2.4 AI tools and techniques ..... 20
2.5 Requirements ..... 23
2.6 Trustworthiness and Fairness Challenges ..... 23
3 UC2 – In-browser visual-based phishing detection ..... 26
3.1 Context and Motivation ..... 26
3.2 Actors and Roles ..... 26
3.2.1 Data Owners ..... 26
3.2.2 Data Collectors ..... 27
3.2.3 Model Trainers ..... 27
3.2.4 Model Executors ..... 27
3.2.5 End Users ..... 27
3.3 Data Description ..... 27
3.4 AI Tools and Techniques ..... 28
3.5 Requirements ..... 28
3.6 Trustworthiness Challenges ..... 30
4 UC3 – Detecting Branch Staff Abuse with Federated Learning ..... 31
4.1 Context and Motivation ..... 31
4.2 Actors and Roles ..... 31
4.2.1 Data Owners (Data Producers) ..... 31
4.2.2 Data Collectors ..... 32
4.2.3 Model Trainers ..... 32
4.2.4 Model Executors ..... 32
4.2.5 End Users (Model Consumers) ..... 32
4.3 Data Description ..... 32
4.4 AI Tools and Techniques ..... 33

4.5 Requirements ..... 35
4.6 Trustworthiness Challenges ..... 37
5 UC4 - LLM-powered chatbot for Parkinson's disease ..... 39
5.1 Context and Motivation ..... 39
5.2 Use Case Scenario ..... 43
5.3 Actors and Roles ..... 47
5.3.1 Data Owners ..... 47
5.3.2 Data Collectors ..... 48
5.3.3 Model Trainers ..... 48
5.3.4 Model Executors ..... 48
5.3.5 End-Users ..... 49
5.3.6 Supporting Roles ..... 50
5.3.7 Role Interactions ..... 50
5.4 Data Description ..... 51
5.5 AI Tools and Techniques ..... 52
5.6 Requirements ..... 53
5.7 Trustworthiness Challenges ..... 54
6 Conclusion ..... 56
7 Bibliography ..... 57

## List of Tables

Table 1 - UC1 Actors' role mapping table ..... 14
Table 2 - Requirements for UC1 ..... 22
Table 3 - Requirements for UC2 ..... 27
Table 4 - Requirements for UC3 ..... 34
Table 5 - PDMonitor device components ..... 39
Table 6 - Requirements for UC4 ..... 53

## List of Figures

Figure 1 - Human-in-the-Loop AI architecture for data quality supervision ..... 15
Figure 2 - UC1: Households' daily lives and lifestyles ..... 16
Figure 3 - Examples of Time Diary ..... 18
Figure 4 - UC2 Business Process Management (BPM) ..... 28
Figure 5 - Branch-Level Transaction Graph Representation ..... 32
Figure 6 - Federated GNN Architecture with Local Branch Training ..... 33

Figure 7 - Secure Update Sharing and Federated Aggregation Process ..... 33
Figure 8 - Global Model Redistribution and Local Fraud Scoring Pipeline ..... 34
Figure 9 - The PDMonitor ecosystem. ..... 38
Figure 10 - A selection of screens from the patient's mobile application: (1) the app's welcome screen, (2) the PD diary interface, and (3) the symptom recording functionality, illustrated here with tremor as an example. ..... 40
Figure 11 - The components of the PDMonitor (TRUMAN) system ..... 42
Figure 12 - A sequence diagram detailing the outlined events of one interaction cycle between a user and the AI Assistant. ..... 43
Figure 13 – Visual description of the two main PDN use case scenarios. One for the clinicians and one for the patients. ..... 44
Figure 14 - User journeys of patients, caregivers, physicians, and nurses interacting with the AI assistant; each line represents a complete interaction sequence over time between the user and the AI assistant, without including system responses. ..... 46
Figure 15 - Interactions between users and the components of the PDMonitor system. Note, not all interactions are bidirectional. In the diagram, Mobile App (Pat) refers to the patient's mobile app, while Mobile App (Phys) refers to the physician's mobile app. ..... 48
Figure 16 - Overview of the AI assistant data pipeline leveraging Retrieval-Augmented Generation (RAG). . 51

# Executive Summary

This deliverable provides the specification of the use cases defined in TRUMAN, which are central to demonstrating and validating the project's objectives on trustworthy, robust, and fair AI technologies. It describes the real-world scenarios in which TRUMAN solutions will be applied, identifies the actors involved, details the nature of the data to be processed, and highlights the specific challenges to be addressed in terms of robustness, privacy, and fairness.

TRUMAN is structured around four representative use cases that cover diverse domains and phases of the AI lifecycle:

- **UC1**: Households' daily lives and lifestyles – improving social and lifestyle research through knowledge graphs and continual learning.
- **UC2**: In-browser visual-based phishing detection – deploying robust, distributed phishing detectors directly into users' browsers.
- **UC3**: Detecting fraudulent transactions in banks – applying graph neural networks and federated learning to identify insider fraud across distributed financial data.
- **UC4**: LLM-powered chatbot for Parkinson's disease – ensuring reliability, fairness, and privacy in a conversational health assistant for patients and caregivers.

The deliverable describes each use case according to a common structure: context and motivation, the actors and their roles, the data and AI tools involved, the workflow, the trustworthiness challenges, and the expected outcomes. By doing so, this document ensures that the TRUMAN technologies are guided by realistic requirements and that their evaluation takes place in scenarios of high societal and industrial relevance.

# Glossary of Terms

| | |
|---|---|
| DOXA | DOXA SPA |
| EMA | Ecological Momentary Assessment |
| EURC | EURECOM |
| GDPR | General Data Protection Regulation |
| GEN | Norton LifeLock France |
| GNN | Graph Neural Network |
| KG | Knowledge Graph |
| KVKK | Kişisel Verilerin Korunması Kanunu |
| ILFI | Indagine Longitudinale sulle Famiglie Italiane |
| ILHP | Italian Longitudinal Household Panel |
| LLM | Large Language Model |
| MCP | Model Context Protocol |
| PDN | PD Neurotechnology LTD |
| RAG | Retrieval-Augmented Generation |
| STKG | Spatiotemporal Knowledge Graph |
| TKG | Temporal Knowledge Graph |
| ToC | Table of Contents |
| UC | Use Case |
| VFS-AI | Virtual Field Supervisor – Artificial Intelligence |
| WP | Work Package |
| YKT | Yapı Kredi Teknoloji Anonim Şirketi |

# 1 Introduction

## 1.1 Purpose of the document

This document reports the results of the work carried out for the execution of Task T2.1 ("Use case definition"). The aim of this work is to identify the use cases that will serve as the foundation for validating TRUMAN's research outputs. These use cases provide the concrete settings in which the project's technologies will be applied and demonstrated, ensuring that scientific developments remain anchored in realistic requirements and scenarios of societal and industrial relevance.

The TRUMAN project ("Trustworthy Human-Centric Artificial Intelligence") aims to design, develop, and validate AI solutions that are robust against adversarial threats, respect privacy, guarantee fairness, and remain transparent and explainable to human users. A central part of this validation process is the definition and analysis of representative use cases, which provide the real-world challenges through which TRUMAN's objectives will be addressed.

These use cases were chosen to capture the diversity of challenges that arise across different phases of the AI lifecycle: data collection, preprocessing, training, and execution. Each use case deals with different AI tools and highlights specific risks and requirements with respect to robustness, security, privacy, fairness, and usability. Together, they ensure that the project's research outputs are tested under realistic and heterogeneous conditions, while also demonstrating their relevance to high-impact domains in society and industry.

The four use cases defined in TRUMAN cover distinct domains, actors, and data types:

**UC1 – Households' daily lives and lifestyles (DOXA).** This use case addresses the limitations of traditional survey-based methods in social science research by leveraging IoT devices, mobile applications, and time diaries to capture household behavior "in the wild." TRUMAN technologies such as knowledge graphs and continual learning will be applied to improve the quality, robustness, and fairness of lifestyle data, while safeguarding privacy and ensuring transparency for participants and researchers.

**UC2 – In-browser visual-based phishing detection (GEN).** Phishing remains one of the most prevalent forms of cybercrime, exploiting users through deceptive websites. This use case develops a phishing detection system embedded directly in the user's browser, powered by deep learning and federated learning. It highlights challenges of robustness against adversarial cloaking, fairness in detection across user groups, and explainability through clear, actionable alerts for end-users.

**UC3 – Detecting fraudulent transactions in banks (YKT).** Financial institutions face the risk of insider fraud, where employees exploit their access to commit illicit transactions. This use case applies graph neural networks and federated learning to distributed banking data from multiple branches. It raises challenges of robustness against manipulated transaction data, fairness in fraud detection, and explainability of results for compliance officers.

**UC4 – LLM-powered chatbot for Parkinson's disease (PDN).** The PDMonitor ecosystem provides digital support to Parkinson's patients. This use case integrates a large language model (LLM)-based chatbot to assist patients with symptom logging, medication tracking, and disease management. It emphasizes robustness against prompt manipulation, fairness in treatment recommendations, and strong privacy protections for sensitive health data, while ensuring transparency and user trust in a medical context.

Each use case is tagged with an acronym in the form UCx. These acronyms will be consistently used throughout the upcoming deliverables to refer to the identified use cases. By covering different domains and types of AI processes (e.g., knowledge graphs, deep learning, graph neural networks, large language models), the four use cases provide representative scenarios for the AI methods developed in **WP3** and expose a wide range of trustworthiness requirements, including robustness, privacy, fairness, and explainability, as addressed in **WP4** and **WP5**. This diversity ensures that TRUMAN's research will address heterogeneous needs and provide solutions that can generalize across sectors.

To analyze the TRUMAN use cases from a trustworthiness perspective, this deliverable adopts a role-based view of the AI lifecycle. In line with the project proposal, five main roles are considered: the **data owner** who provides or generates the data used by the system; the **data collector** who gathers, organizes, and prepares these data for the next processing steps; the **model trainer** who uses the collected data to train or update the AI model; the **model executor**, who runs the trained model on new inputs and produces results; and the **end-user** who receives, interprets, or acts upon the model's output. Depending on the use case, one actor may take multiple roles, or different actors may cover different stages of the lifecycle. This role-based analysis is necessary because trustworthiness challenges, including privacy, security, fairness, and explainability, emerge differently depending on who controls the data, who trains the model, who executes it, and who is affected by its outputs. For this reason, each use case in this deliverable will also map its actors onto these roles, so as to support the identification of trustworthiness challenges and provide a basis for the work in the following technical work packages.

The definition of the identified use cases builds on discussions and exchanges with the end users and stakeholders involved in the domains of social science, cybersecurity, finance, and healthcare. The descriptions illustrate how TRUMAN systems are supposed to behave in these contexts and underline the concrete challenges that current services face in terms of robustness, explainability, fairness, and privacy. In several cases, the use cases stem from existing service deployments that are limited in their current capacity to guarantee trustworthiness and are therefore extended through TRUMAN technologies to achieve a higher level of reliability and user confidence.

Operatively, for each use case, this document presents the use case definition — including the involved actors, the nature of the data, the processing steps, and the underlying AI tools — together with the related trustworthiness challenges. This will serve as input for the technical work

packages, ensuring that the methods and tools developed in TRUMAN are guided by real-world needs and can be validated against practical scenarios.

## 1.2 Outline

This deliverable is structured as follows:

Section 2 presents UC1, focusing on households' daily lives and lifestyles, detailing the context, data collection approach, actors, AI techniques (with emphasis on knowledge graphs and continual learning), and related trustworthiness challenges.

Section 3 describes UC2 on in-browser visual-based phishing detection, outlining the system architecture, data flows, AI tools, requirements, and challenges related to robustness, privacy, and explainability.

Section 4 introduces UC3, addressing insider fraud detection in banking through federated graph learning, including actors, data representation, model architecture, and governance-related trustworthiness issues.

Section 5 details UC4, an LLM-powered chatbot for Parkinson's disease, covering the clinical context, user roles, multimodal data, AI techniques, requirements, and ethical and regulatory challenges.

Each use case follows a common structure to ensure consistency and comparability across domains, supporting subsequent technical development and evaluation activities within the TRUMAN project.

# 2 UC1 – Households' daily lives and lifestyles

## 2.1 Context and Motivation

Market and scientific research on people's lives have mostly been based on the administration of questionnaires. This seriously limits the possibility of capturing the complex process of relationships, division of duties, roles, and behavior between the subjects in time and social and physical space. Conducting market and scientific research on people's lives 'observing in nature' means focusing on understanding real-world behavior and interactions without the influence of controlled environments or self-reported data.

By observing "in the wild," researchers focus on naturalistic observation and can provide more authentic insights into how household members live, make decisions, and engage with their social and physical environment. In a nutshell, these data can be used as input to develop new models, including AI models, covering a wide range of social topics, e.g. to help improve the lives of households and individuals, to inform public stakeholders to develop public policies and private companies to develop new business projects.

The data used in this case are produced as a new and methodologically innovative data collection. The Indagine Longitudinale sulle Famiglie Italiane (ILFI) / Italian Longitudinal Household Panel (ILHP) was created with the aim of studying the everyday social interactions of members of domestic households. The objective is to investigate the interactions among participants and between them and the physical space in which they live and interact.

These aspects will make it possible to gain detailed insights useful for outlining the behaviors, habits, and attitudes of participants during their daily activities over time. The study will also contribute to improving knowledge of the living, care, and social conditions of individual household members, as well as the effects of these interactions on each participant's daily life and on their decision-making and life trajectories.

This new way of looking at households can allow us to cover a wide range of topics, e.g., Shopping Behaviors; Daily Routines and Time Management; Meal Preparation and Eating Habits; Leisure and Recreation Activities; Health and Wellness Routines; Social Interactions and Family Dynamics; Parenting and Childcare Practices. In addition to observing how the family acts as a unit, it is also possible to observe how everyone in the family deals with the social and physical space in which they live, for example in their daily activities at school, at work, in their leisure time, with their friends, and so on.

At the same time, smart cities have become the new hip. One of the main problems is collecting data from citizens and their life characteristics, e.g. how they move, work, spend their free time, etc. Smart cities can significantly benefit from household citizen science data collection by harnessing the power of data collected directly from residents. This approach enables cities to gain deeper insights into urban life, improve services, and create more responsive and sustainable environments. Household citizen science data collection offers smart cities a valuable resource for improving urban living, optimizing resource management, enhancing public health,

and fostering a more engaged and resilient community. By integrating household-level data into their strategies, smart cities can become more adaptive, sustainable, and inclusive.

This data can be used as input to develop new models of artificial intelligence, e.g. to help the lives of families. Informing public stakeholders to develop public policies. Up to integration into a digital twin of the city to improve the smart city model.

The use case aims to explore some of the key issues driving this project, related to low data quality that negatively impacts both standard statistical analysis and artificial intelligence models. Although in the methodological literature underlying the social sciences these issues are widely known and codified, with the advent of new survey technologies, such as the one proposed here, new problems emerge that impact the entire **data collection process**. Here, the main issues concern the quality of the data and the presence of anomalies and distortions (**noisy, missing and outliers**) in the collected data, generated either by the participants (e.g. memory errors, typing errors, etc.) or by the technology used (e.g. app operation, sensor errors, OS, etc.).

The UC aims to develop three different AI models.

Data quality is the first requirement for a good AI model and standard data analysis. Due to the complexity of data collection, the risk of missing or incorrect data is high. The first goal is to develop a **virtual field supervisor** (VFS-AI) that tries to detect outliers and correct them.

To reduce the burden on the participants, the time diary will only be filled in during certain observation periods. This means that we will have missing Time Diary data by choice. The aim is to develop AI models that infer and predict from the sensors the information that can fill in the missing data in the time diary.

The final goal is to develop artificial intelligence models that can predict certain habits and routines related to mobility and household members.

**VFS-AI is expected to have an impact on the following areas:**

- **Scientific:** (a) Establishing a new methodological standard for computational social sciences (CSS) based on smartphones; (b) Reducing systematic bias linked to device heterogeneity. (c) Enhancing reliability of longitudinal behavioral inference. (d) Introducing quality-aware modeling frameworks.
- **Technical:** (a) Bridging CSS with MLOps and knowledge of graph engineering. (b) Providing reusable architecture for large-scale sensing studies. (c) Contributing to explainable AI in research operations.
- **Societal:** (a) Increasing trustworthiness and transparency of digital behavioral research. (b) Supporting ethical and responsible use of sensing technologies. (c) Promoting high-quality evidence for policy-relevant insights.

## 2.2 Actors and Roles

The project involves various stakeholders depending on the different steps of the entire data collection process and use, from sample maintenance to supervision of streaming data collection, right up to the researchers who will use this data. Each of them plays a distinct role in ensuring the quality of the data collected. The system operates as a human-in-the-loop AI pipeline, where data generation is distributed across participants and devices; data collection is hybrid (automated + supervised), and model training is continuously updated through validated human feedback. This architecture ensures high data quality, robustness, and fairness, while enabling scalable and adaptive data collection in real-world environments.

### 2.2.1 Field supervisors as Mode Trainer

Field supervisors are the main users of the system and act as human-in-the-loop. They receive daily reports on the data flow collected by the app from survey participants. They review the report using a dashboard on participants' behavior. They highlight suspicious cases and decide whether a case represents a problem. They interact with web panel managers to decide on the most appropriate actions to take. Their decisions generate new labels that are used to improve the model over time.

### 2.2.2 Web panel managers as Model Executors

Web panel supervisors act as humans-in-the-loop, serving as the primary interface with survey participants. Their role is crucial in ensuring sustained engagement and reducing attrition over time. Based on the reports generated by field supervisors, they can engage directly with the flagged participants to better understand the nature of the issues encountered. They can then assess and verify the reported cases, ensuring that the information provided is accurate and actionable. Their decisions also generate new labeled data, which are subsequently used to iteratively improve the model's performance over time.

### 2.2.3 Survey participants (the sample) as Data Owners

They are the producers of the data. Through the app, they generate and share data on their activities. Their data is analyzed by the system to identify potential problems, but they do not interact directly with the AI system. They interact with Web panel managers and can report any problems they encounter during data collection. However, in a certain sense, they can help the system. Through the answers provided in the time diary, the participants produce labels and information which, combined with the sensors, improves the model over time to detect errors, but also routines and habits. This information can be exploited to improve the data in phases where only the sensors are detected.

### 2.2.4 Smartphone and DTraces app as Data Owners

They are also data producers. Through on-board sensors, the network, and the mobile connection, they produce, collect, and store data on user activities. This data is input into VFS-AI

but does not interact directly with the artificial intelligence system. It interacts with the user through the time diary.

### 2.2.5 Customers: The Researcher as End-Users

Researchers are data users, but they do not interact directly with the system. They indirectly benefit from improved data quality, greater reliability of the data collected, and better-quality research products. Although they do not interact with the system, their feedback is crucial in detecting additional data quality issues that were not detected in previous stages.

*Table 1 - UC1 Actors' role mapping table*

| AI Role | UC actor & tools | Core Function | Key Risks | Mitigation Mechanisms |
|---|---|---|---|---|
| **Data Owner** | Survey participants; Smartphones & DTraces app | Generate behavioral and sensor data | Missing data (MNAR), measurement bias, privacy risks | Incentives, onboarding, transparency, sensor validation |
| **Data Collector** | DTraces infrastructure; Field supervisors; Web panel managers | Collect, structure, monitor, and validate data | Data drift, inconsistent reporting, participation bias | Real-time monitoring, anomaly detection, participant support |
| **Model Trainer** | Field supervisors; Web panel managers; Participants (via labels) | Provide validated labels and ground truth for model improvement | Label noise, subjective validation, uneven labelling | Standardized protocols, cross-validation, feedback loops |
| **Model Executor** | Field supervisors; VFS-AI system | Detect anomalies, classify patterns, generate alerts | Model bias, overfitting, error propagation | Continuous retraining, fairness checks, performance monitoring |
| **End-User (Operational)** | Field supervisors; Web panel managers | Interpret outputs and take corrective actions | Misinterpretation of alerts, decision inconsistency | Dashboard design, training, decision guidelines |
| **End-User (Analytical)** | Researchers | Use processed data for analysis | Hidden biases, data quality issues not detected upstream | Metadata transparency, quality indicators, feedback to system |

Additionally, Figure 1 illustrates a multi-layer pipeline integrating data generation, collection, AI-based processing, and human validation. Behavioral and sensor data are generated by survey participants and smartphones (DTraces app), then ingested and structured within the data collection infrastructure. Field supervisors and web panel managers monitor data streams, detect anomalies, and support participants, producing validated and enriched data. The VFS-AI model processes these inputs to identify patterns, assess data quality, and generate alerts. These outputs are reviewed by human actors in a decision layer, whose feedback produces labeled data that continuously improves the model. The system ultimately delivers high-quality datasets to

researchers, while embedding cross-cutting principles of privacy, fairness, and accountability. This figure is generated using ChatGPT.

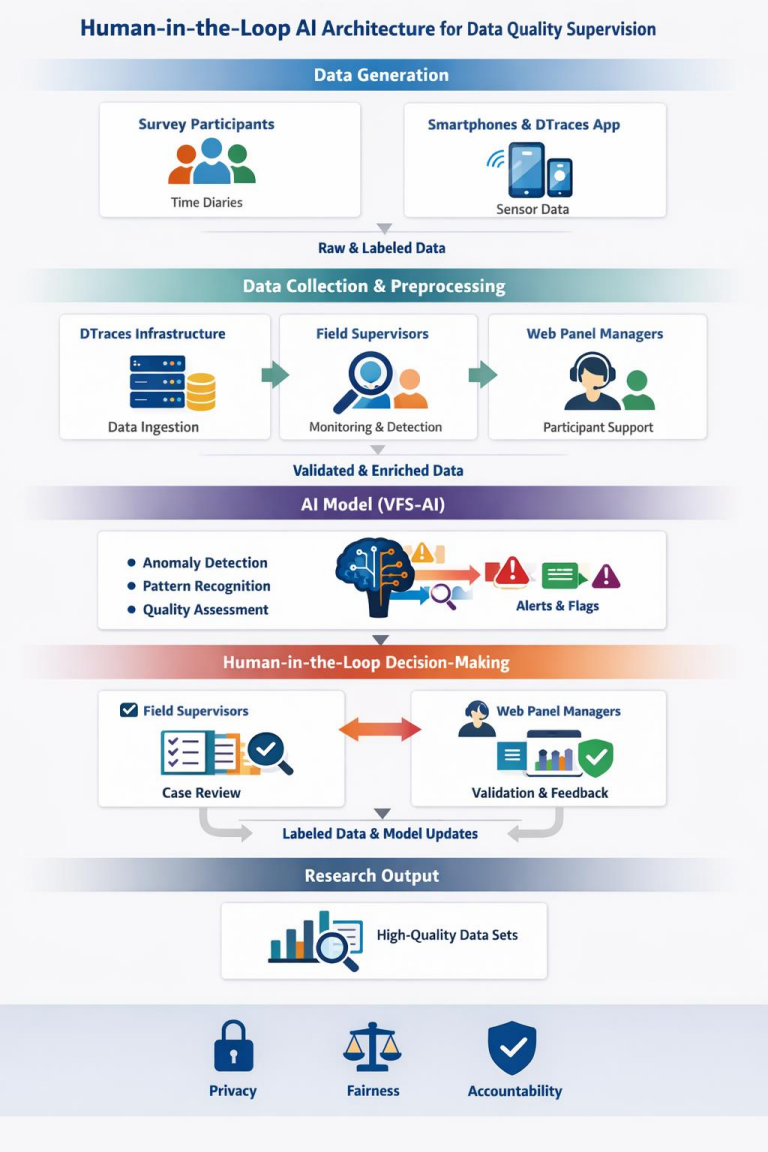



*Figure 1 - Human-in-the-Loop AI architecture for data quality supervision*

## 2.3 Data Description

To realize this use case, data will be collected from 400 sampled Italian households through a specific app called DTraces developed by UniTn. All adult members (18+) agree to install this application on their smartphone and use it for 6 months. The estimated total number of adult participants is about 1,000 individuals.

Participants will be recruited from the DOXA Panel—one of Italy's oldest (since 2008) and largest web panels, with over 140,000 members—ensuring strong control over sample characteristics. A dedicated platform, *Indagine Longitudinale sulle Famiglie Italiane (ILFI) / Italian Longitudinal Household Panel (ILHP)*, accessible via Doxa credentials, will support recruitment and participation. The platform will provide project information, FAQs, a recruitment interface, and a participant area with details on participation, incentives, and withdrawal. It will also manage registration, incentives, household linkages, and user support.

A sample of current panel members with existing household profiles will be invited by email. Those who accept joining the ILHP initiative can invite other adult household members via a personal registration link. Invited members to register and complete a profiling survey, with all data undergoing consistency checks. Participants will receive only ILHP-related surveys, administered via Doxa's CAWI platform.

Data collection will be carried out using an EMA (Ecological Momentary Assessment) methodology implemented in the DTraces app, toensure a constant flow of information between researchers and participants in compliance with ethical principles and the GDPR.

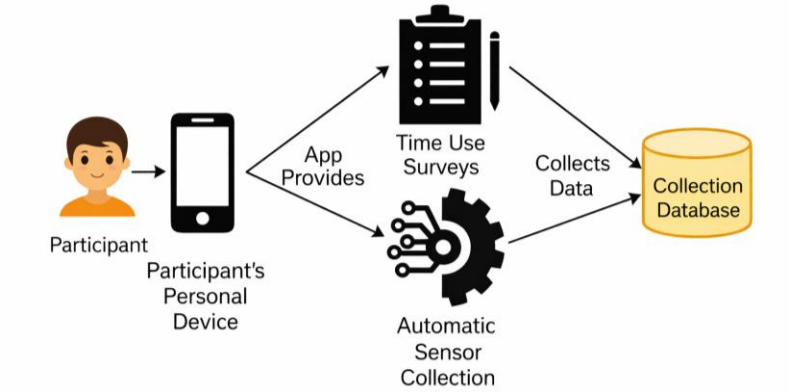
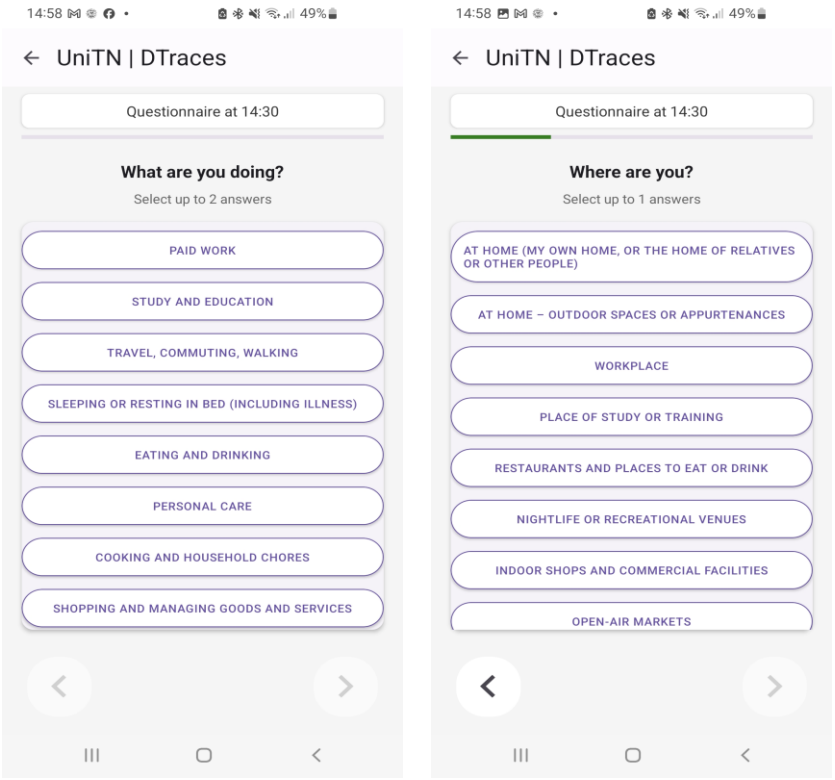



*Figure 2 - UC1: Households' daily lives and lifestyles*

In order to ensure and increase trust and awareness, and to guarantee maximum respect for participants' privacy, we will operate on two distinct levels. On the one hand, we will work closely with the ethics committee and privacy committee of the University of Trento and Doxa to ensure compliance with ethical standards and the GDPR and to provide participants with adequate information on privacy. On the other hand, a new application called DTraces (Figure 2) is being developed, which will be installed on participants' smartphones. This application is designed to support EMA methodologies, but above all it allows participants full control over the data provided, enabling them to turn sensor data sharing on and off at any time. The application is being developed by the Department of Sociology and Social Research (DSRS) at the University of Trento (UNITN). DTraces supports both continuous passive data collection through smartphone sensors (such as GPS, Bluetooth, device usage, etc.) and interactive data collection based on user-reported information (e.g., time diaries administered at regular intervals or structured

questionnaires). It runs on both operating systems (iOS and Android) and can be downloaded from the Play Store and the App Store. The installation process is smooth and user-friendly.

DTraces relies on the infrastructure of the University of Trento (UNITN). The app clearly informs participants about the study and the types of data being collected; each sensor can be individually authorized and activated or deactivated by the user. The overall study procedure, including study design, ethical and privacy approval, data collection, processing, storage, and data sharing, follows the Ecological Momentary Assessment (EMA) scientific methodology [1], as subsequently refined by UNITN (DSRS), and builds on Doxa's long-standing experience in survey data collection. The data catalogue, maintained jointly by Doxa and DSRS, provides the platform supporting all phases of the study. It includes a detailed technical data management plan that outlines the required security measures and technical procedures at each step of the workflow.



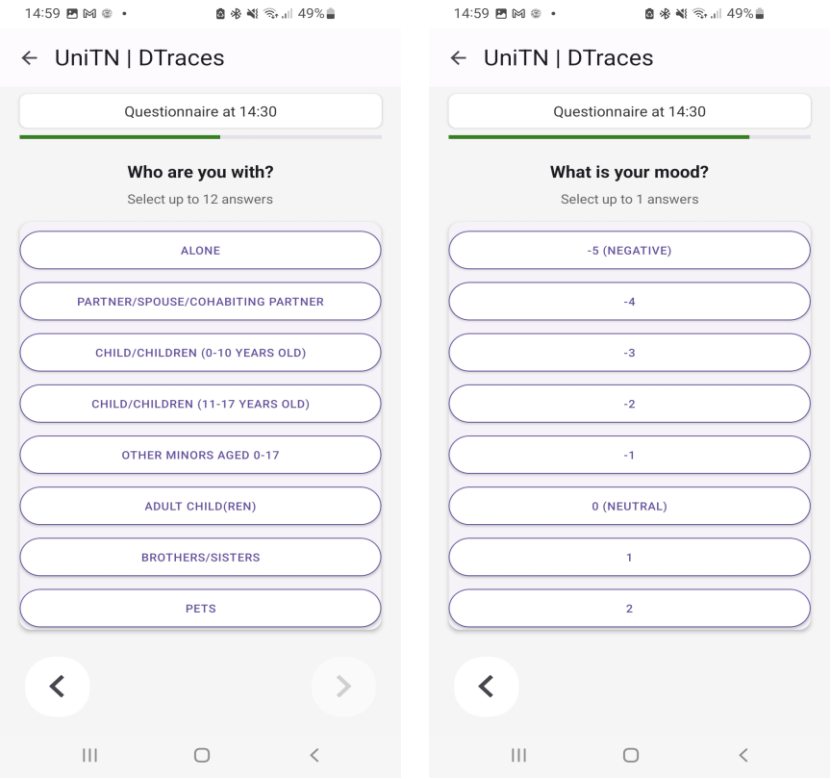



*Figure 3 - Examples of Time Diary*

Data will be collected through a time diary, sensor streaming data, and a web questionnaire. To reduce the burden, the time diary will be administered on average two weeks every two months, following the same research protocol used in the past. Meanwhile, smartphone sensor data will be collected continuously throughout the observation period.

Through the online questionnaire, we collect static information such as: **Sociodemographic** characteristics (e.g. sex, gender, age, education, occupation, municipality, household composition); **Technology available**: (e.g. devices available, wearable devices, internet use, social media); **Technological profile** (Affinity for Technology Interaction (ATI) Scale, Data Donation Profile); **Psycho-social profile** (Big Five Personality Traits, Pure Procrastination Scale, Integrated Locus of Control Scale)

Through the DTraces notification (Figure 3) we collect daily data such as: **What are you doing?** (e.g. Paid work (on-site or remote work), Study and education (school/university study and structured courses),Travel and commuting (walking, public/private transport) (administer questions A1_1 and A1_2), Sleeping or resting in bed (including illness), Eating and drinking (breakfast, lunch, dinner, snacks, aperitifs), Personal care (hygiene, dressing, body care), Cooking and household chores (cleaning, cooking, light maintenance), etc.); **Where are you?** (e.g. At home (my own home, or the home of relatives or other people), At home – outdoor spaces or appurtenances (garden, garage, courtyard), Workplace (office, company premises, workplace canteen), Place of study or training (school, university, nursery, library, cafeteria), etc.); **Who are**

**you with?** (e.g. Alone, Partner / spouse / cohabiting partner, Child/children (0-10 years old), etc.);
**How is your mood?** (scale from (1) sad to (10) happy)
Through the DTraces smartphone sensors, we collect streaming data such as location (GPS), WiFi, Bluetooth, battery, proximity, screen status, headset, ring mode, airplane mode, App usage

Finally, during data collection and once data collection is complete, another critical aspect concerns data quality and the presence of anomalies and distortions (noise, missing data and outliers) in the collected data, generated by participants (e.g. memory errors, typing errors, etc.) or by the technology used (e.g. app operation, sensor errors, operating system, etc.). These aspects will be addressed by VFS-AI applications.

## 2.4 AI tools and techniques

The collected data will serve as a basis for developing standard statistical analyses and AI models, with the aim of improving the quality, reliability, validity, and robustness of the data collected. In addition, the results derived from empirical evidence on families' habits and routines, their lifestyles, and behaviors will be studied to raise awareness among public and private stakeholders about the importance of this type of survey.

In collaboration with WP3, the project aims to develop several AI solutions. In particular, it will address how to generate Knowledge Graphs (KGs), Temporal Knowledge Graphs (TKGs), and Spatiotemporal Knowledge Graphs (STKGs) to data from social surveys and IoT devices. Continuous learning based on incremental training will also be used to continuously monitor, evaluate, and generalize the model, in order to improve the quality of the collected data. The goal is to develop machine learning models for anomaly detection (Anomaly Detection Models), missing data imputation (Missing Data Imputation Models), bias detection and correction (Bias Detection and Correction Models), and outlier detection (Outlier Detection Models).

We are also interested in developing AI models capable of inferring and predicting—based on sensor data—useful information to fill gaps in time-use diaries and in assessing the potential of synthetic data.

In this context, Knowledge Graphs (KGs) provide a powerful methodological layer that significantly improves consistency, precision, and validity, without increasing respondent burden. Knowledge graphs (KGs) address these issues because they: represent structured yet flexible knowledge; preserve semantic coherence over time; enable adaptive personalization; integrate subjective self-reports with observational data. Knowledge Graphs (KGs), and their temporal (TKG) and spatiotemporal (STKG) extensions, provide a powerful framework to address these challenges by acting as a semantic integration and validation layer between self-reported diary data, traditional social surveys, and IoT-based observations.

Temporal Knowledge Graphs extend this structure by explicitly modeling time. Each diary event is associated with time points or intervals and linked to other events through temporal relations such as "before," "after," or "during." This temporal structure enables the detection of common

EMA errors, including overlapping activities, implausible durations, and broken daily sequences. Instead of relying solely on post-hoc statistical corrections, a TKG allows researchers to identify and correct temporal inconsistencies using semantic and logical rules that reflect realistic daily routines. This substantially improves the internal coherence of longitudinal time-diary data.

Spatiotemporal Knowledge Graphs further enhance data quality by integrating space and movement into the model. Locations reported in EMA diaries can be linked to spatial entities such as homes, workplaces, transport routes, or indoor zones, while IoT data, such as GPS traces, Bluetooth proximity, or smart-home sensors, provide continuous spatial evidence. By jointly reasoning over space and time, STKGs enable powerful plausibility checks. For example, they can identify sudden location jumps that are incompatible with known mobility constraints or flag discrepancies between reported locations and sensor-based geofencing. This reduces both random errors and systematic misreporting, thereby improving precision without increasing the number of diary questions.

Moreover, data errors remain pervasive, especially in longitudinal designs. These errors include inconsistent responses, implausible combinations of answers, temporal overlaps, missing entries, and systematic misclassification due to respondent fatigue or satisficing. Knowledge Graphs (KGs) offer a principled way to detect, prevent, and correct these errors by introducing semantic structure and contextual reasoning into the data collection and validation process.

The first class of errors in EMA time diaries concerns logical inconsistencies. Respondents may select combinations of answers that are syntactically valid but semantically implausible, such as reporting “sleeping” in a public transport setting or “having a meeting” while indicating that they are alone. Traditional validation rules treat variables independently and often detect these problems only during post-collection data cleaning. In contrast, a Knowledge Graph encodes semantic compatibility rules between activities, locations, social contexts, and times of day. When an implausible combination is entered, the system can trigger soft validation, such as a confirmation request or a context-aware suggestion. This approach prevents many errors at the moment of data entry, without interrupting the natural flow of EMA reporting.

A second major source of error lies in temporal inconsistencies. EMA time diaries frequently contain overlapping activities, implausible durations, or broken sequences that do not reflect realistic daily routines. Knowledge Graphs explicitly represent temporal relations between activities—such as “before,” “after,” or “during”—as well as typical durations and common transitions. When a reported sequence violates these expectations, the KG can flag the anomaly, suggest a correction, or infer a likely missing transition. This allows researchers to address temporal errors using semantic logic rather than purely statistical smoothing, resulting in more coherent and interpretable activity sequences.

Misclassification errors are another pervasive issue, even with close-ended questions. Under time pressure, respondents often select the most convenient option rather than the most accurate one, a behavior known as satisficing. Knowledge Graphs mitigate this problem by enabling contextual reinterpretation of responses. Because each answer option is embedded in a network of related concepts, its meaning can be refined using surrounding context, such as location, time of day, or

previous activities. For example, a generic response like “using smartphone” can be reinterpreted as work-related communication, social interaction, or leisure activity depending on its position in the graph. Importantly, this correction happens at the interpretation level, preserving the original response while reducing classification error.

Missing data represents a further challenge in EMA studies. Non-responses may occur because participants miss prompts, ignore notifications, or temporarily disengage. Knowledge Graphs support semantically informed handling of missing data by leveraging contextual continuity and known routines. If surrounding entries strongly suggest a likely activity, the KG can infer a plausible value while explicitly marking it as inferred rather than observed. This approach improves data completeness without obscuring uncertainty, and it allows researchers to distinguish true behavioral gaps from reporting artifacts.

In longitudinal EMA designs, errors also accumulate through interpretation drift. Over time, participants may subtly change how they understand and use response categories, even when those categories remain formally unchanged. Knowledge Graphs help control this process by anchoring each close-ended option to a stable semantic definition that persists across waves. Shifts in response patterns can then be interpreted as genuine behavioral change rather than measurement instability. This distinction is critical for maintaining longitudinal data integrity and for drawing valid inferences about change over time.

In other words, Knowledge Graphs transform error management in EMA longitudinal time diaries from a reactive, post-hoc cleaning task into a proactive, explainable, and context-aware process. By embedding semantic rules, relational constraints, and temporal logic into the data collection and interpretation framework, KGs reduce logical inconsistencies, temporal errors, misclassification, missingness, and longitudinal drift. The result is higher-quality EMA data that remain reliable, interpretable, and analytically robust, even in the complex conditions of naturalistic observation.

From a measurement perspective, this integrated KG/TKG/STKG framework improves construct validity by making theoretical assumptions explicit, ecological validity by grounding reports in real-world spatiotemporal context, and longitudinal validity by stabilizing meaning over time. Rather than treating data cleaning as a purely technical, post-collection task, knowledge graphs embed data quality control directly into the measurement process.

In sum, using Knowledge Graphs and their temporal and spatiotemporal extensions transforms EMA longitudinal time diaries into semantically rich, context-aware measurement systems. By integrating social surveys and data from sensors within a unified graph-based model, researchers can systematically reduce data errors and achieve higher levels of consistency, precision, and validity—while preserving the low burden and scalability that make EMA methods so valuable in naturalistic research settings.

## 2.5 Requirements

*Table 2 - Requirements for UC1*

| Reference | Requirement | Description |
|---|---|---|
| **R2.1** | VFS-AI: The first goal is to develop a virtual field supervisor that tries to detect outliers and correct them through Intelligent Imputation. | Here, the main issues concern the quality of the data and the presence of anomalies and distortions (**noisy, missing and outliers**) in the collected data, generated either by the participants (e.g. memory errors, typing errors, etc.) or by the technology used (e.g. app operation, sensor errors, OS, etc.).<br><br>Uses the TKG to suggest corrections for erroneous data based on historical probability (e.g., "User was likely at home") |
| **R2.2** | Develop AI models that infer and predict from the sensors the information that can fill in the missing data in the time diary | To reduce the burden on the participants, the time diary will only be filled in during certain observation periods. This means that we will have missing data by choice. Also, here uses the TKG to suggest corrections for missing data based on historical probability (e.g., "User was likely at home") |
| **R2.3** | Dashboard Alerts for monitoring data collection and reporting of outliers. | The large number of participants and the enormous amount of data collected daily require tools that support supervisors in the field. Visualizes the "health" of the cohort. Signals critical drops in compliance. |
| **R2.4** | Multi-dimensional Data Quality Score (DQS) | Data quality is not as a residual cleaning task, but an observable and modelable construct. Dimensions include: (a) Completeness; (b) Freshness/latency; (c) Cross-modal coherence; (d) Temporal stability; (e) Device-context robustness |

## 2.6 Trustworthiness and Fairness Challenges

UC1 raises a number of critical challenges that span the entire data collection pipeline, from participant engagement to data processing and analytical use. These challenges can be grouped into two main areas.

The first concerns the involvement of participants in the data collection process. This implies the active cooperation of households in continuously providing behavioral and sensor data, which in turn depends on building and maintaining trust in the overall data collection infrastructure. Traditionally, this has been addressed by strengthening privacy guarantees and adherence to ethical principles, which can increase participation rates and reduce panel attrition and data drift. However, in the context of the increasing datafication of society, this approach is no longer sufficient.

More specifically, privacy challenges in this context are multi-layered. They include:

i. **informational privacy risks**, related to the continuous collection of high-resolution behavioral and location data;
ii. **inferential privacy risks**, where seemingly innocuous sensor data can be combined to infer sensitive attributes (e.g., routines, social relations, health conditions);
iii. **collective privacy concerns**, since data about one individual may indirectly reveal information about other household members; and
iv. **control and transparency issues**, as participants may not fully understand how their data are processed, reused, or integrated into analytical models.

Addressing these challenges requires moving beyond compliance-based privacy (e.g., consent forms) toward **ongoing transparency, explainability, and user agency**, including feedback mechanisms, dynamic consent, and clear communication of risks and benefits.

Closely related to trust is the issue of **data quality**, which represents a fundamental challenge for the reliability and fairness of the system. If the data collected are erroneous, incomplete, or systematically biased, the downstream quality of services and analyses will deteriorate. For instance, missing data that are not random (e.g., due to differential participation across socio-demographic groups) may lead to distorted representations of behavior. This, in turn, can generate **fairness issues**, as certain groups may be underrepresented, misrepresented, or systematically disadvantaged in the models and decisions derived from the data. Ensuring high data quality is therefore not only a technical requirement but also an ethical imperative.

The second area concerns the intrinsic specificity of survey-based data collection. While each implementation must be adapted to the specific context, there is a set of core methodological steps—well established in the scientific literature—that must be rigorously defined and “fixed” in order to ensure robustness, comparability, and data quality. These include:

- **Sampling and recruitment design**: ensuring representativeness and minimizing selection bias, including strategies to reach hard-to-engage populations;
- **Onboarding and training procedures**: providing clear instructions and support to participants to reduce user-related errors and heterogeneity in data generation;
- **Instrumentation and calibration**: standardizing the functioning of sensors and devices to reduce measurement error across contexts and devices;
- **Data validation and real-time monitoring**: implementing automated checks to detect anomalies, missingness patterns, and irregular behaviors during data collection;
- **Error correction and imputation strategies**: applying statistically grounded methods (e.g., MCAR/MAR/MNAR-aware approaches) to handle missing or inconsistent data;
- **Bias detection and mitigation**: systematically assessing whether data quality and model performance differ across groups, and correcting for such disparities;
- **Feedback loops and adaptive design**: using metadata and process data to continuously improve the data collection protocol. The key challenge, therefore, is not only to design a context-sensitive data collection system, but also to establish a **standardized and**

**transparent methodological backbone** that ensures data quality, comparability, and fairness across implementations.

# 3 UC2 – In-browser visual-based phishing detection

## 3.1 Context and Motivation

Phishing remains one of the most widespread and persistent forms of cybercrime, targeting users through deceptive webpages designed to mimic legitimate services. These attacks continue to evolve in sophistication, exploiting visual cues, logos, and layouts that are difficult for both humans and automated systems to distinguish from genuine sites. The consequences are significant — from financial loss and credential theft to a growing erosion of trust in digital platforms and online transactions.

Existing countermeasures, such as URL blocklists, heuristic filters, or centralized reputation systems, struggle to keep pace with the dynamic nature of phishing. Attackers employ cloaking techniques and domain churn to evade detection, often showing benign content to automated crawlers while exposing malicious versions to real users. Moreover, most current detection approaches depend on centralized infrastructures and global data aggregation, which introduce latency, privacy risks, and single points of failure. They also offer limited transparency to end-users, providing little insight into why a particular site was flagged as dangerous.

UC2 addresses these limitations through the development of an in-browser visual-based phishing detection system, capable of performing analysis and classification directly within the user's environment. By combining LLM-based inference and federated learning, the system enables local decision-making while supporting collaborative model training across distributed participants, human-in-the-loop for model improvement and error remediation. This architecture enhances both explainability and privacy, as the users are provided with enhanced explanations on the AI decision making and no raw browsing data is transferred externally. It also creates a practical setting to validate TRUMAN's objectives on trustworthiness AI through explainability and usability.

## 3.2 Actors and Roles

The phishing-detection use case involves several stakeholders who contribute to the design, operation, and continuous improvement of the system. Each actor plays a distinct role in ensuring that data are collected securely, models are trained effectively, and detection results are reliable, transparent, and actionable.

### 3.2.1 Data Owners

The primary data owners in this use case are the end users, whose web browsing activities generate the data used by the phishing detection system. These data include visual characteristics of webpages as well as URL-derived features observed during normal navigation. The data are produced locally within the user's browsing environment and reflect real-world interaction patterns with web content. In this sense, end users act as the sources of the behavioral and contextual information that support phishing detection, with data remaining under their control throughout the system's evolution.

### 3.2.2 Data Collectors

The end user's browser acts as the data collector. During web navigation, the browser extracts relevant features, such as visual elements of webpages and URL-based indicators and prepares them for analysis. This collection process is performed locally, ensuring that sensitive information remains within the user's environment. As the system evolves, data collection continues to operate at the browser level while supporting collaborative learning approaches that do not require centralized aggregation of raw data.

### 3.2.3 Model Trainers

AI engineers and security analysts act as model trainers. In the initial stage, model training is performed within GEN's infrastructure, where data preprocessing pipelines are defined, detection models are developed, and performance is evaluated and improved. Security analysts contribute by validating user feedback, analyzing reported cases, distinguishing false positives from real phishing attempts, and refining detection thresholds and rules.

As the system evolves, training progressively shifts toward a distributed setting in which end-user environments participate in the learning process. This enables continuous model improvement through collaborative mechanisms while preserving privacy and reducing reliance on centralized data collection.

### 3.2.4 Model Executors

The end user's browser acts as the model executor. The phishing detection model is deployed locally and operates in real time during web browsing. It analyzes webpage features and URL characteristics to identify potentially malicious content and generates immediate feedback for the user. This local execution remains consistent as the system evolves, ensuring low latency, scalability, and strong privacy guarantees.

### 3.2.5 End Users

End users are the primary consumers of the system. They interact directly with the phishing detector integrated into their browsers and receive real-time alerts when suspicious webpages are encountered. Users can confirm or dismiss alerts, and their feedback provides valuable signals for improving the system. Through this interaction, end users act as humans-in-the-loop, contributing to the continuous refinement, robustness, and adaptability of the detection process.

## 3.3 Data Description

The data used in this use case consists of screenshots of rendered webpages collected during users' normal browsing activity. In the initial phase, a large and diverse ground-truth dataset of both legitimate and phishing webpages is assembled within GEN's infrastructure, where security analysts curate, validate, and use it to train a robust detection model. In this phase, continual learning takes place under the analysts' supervision, allowing the model to evolve based on user reports and confirmed phishing instances.

Once the detector demonstrates sufficient robustness, it becomes suitable for distributed training through federated learning. In this operational stage, the screenshots processed within users' browsers contribute indirectly to the model's continual improvement, as local updates are aggregated without exposing any raw visual data. This two-stage data process—centralized training followed by decentralized refinement—ensures both high detection accuracy and strong privacy protection, while enabling the model to adapt dynamically to emerging phishing tactics.

The phishing detection system combines several AI techniques to analyze webpage screenshots, identify brand impersonation, and evolve through distributed learning.

## 3.4 AI Tools and Techniques

At its core, the detector employs deep learning models for visual similarity and multimodal analysis, capable of jointly interpreting visual and textual elements on rendered webpages. This includes the integration of large language models (LLMs) used in a multimodal fashion to associate textual content (such as brand names or interface labels) with visual features (such as logos, layout, or style). This combined reasoning strengthens the model's ability to detect subtle brand impersonation attempts that would evade purely visual or textual detectors.

During the initial development phase, model training and evaluation take place within GEN's infrastructure, where analysts validate model performance and ensure robustness against diverse attack types. The resulting model is deployed in the browser as a local inference engine, providing real-time classification of visited webpages.

As the system matures, it transitions to federated and continual learning, allowing the model to improve based on user-reported errors and local models updates without transferring any raw data. In this distributed phase, model updates are aggregated securely to enhance performance against emerging phishing strategies while maintaining privacy guarantees.

Finally, the system integrates an explainability layer that communicates phishing alerts in a transparent and understandable way, highlighting visual or textual cues that triggered the warning. This component strengthens user trust and aligns the use case with TRUMAN's human-centric AI principles.

## 3.5 Requirements

*Table 3 - Requirements for UC2*

| Reference | Requirement | Description |
|---|---|---|
| **R2.1** | LPWD – Local Phishing website detector | Module that runs inference on users' device |
| **R2.1.1** | Local training infrastructure | Module that runs training on the users' device |
| **R2.1.2** | User feedback and false positive reporting | Module that runs on users' device used for false positive reporting |
| **R2.2** | CPWD – Centralized Phishing website detector | Module that runs inference in the company cloud |
| **R2.2.1** | Backend training infrastructure | Module that runs training in the company cloud |

| | | |
|---|---|---|
| **R2.2.2** | Human in the loop for continual learning | Module that allows the analysts to improve decision making with feedback gathered from the users |
| **R2.3** | Backend datasets and telemetry | GEN's backend solutions and datasets to bootstrap the training process, ensure data quality, and support continual learning |

As depicted in Figure 4 and summarized in Table 3, the phishing detection use case aims to design and implement an in-browser phishing detection (PWD) system capable of identifying brand impersonation through the visual analysis of webpages and complementary URL-based features. The detector must operate locally within the user's browser, without relying on external suspicious feeds or automated scanners, and deliver real-time alerts when a potential phishing page is encountered. The system must support a two-phase architecture: an initial stage in which data preprocessing and model training are performed within GEN's infrastructure, followed by a distributed phase where collective training and model execution take place directly on end-user devices. The resulting solution should ensure high detection accuracy, robustness against adversarial evasion techniques, and equitable performance across heterogeneous users and browsing environments. Moreover, it must incorporate explainability mechanisms to provide users with clear and interpretable warnings, enabling informed decisions and fostering trust in the system's operation.

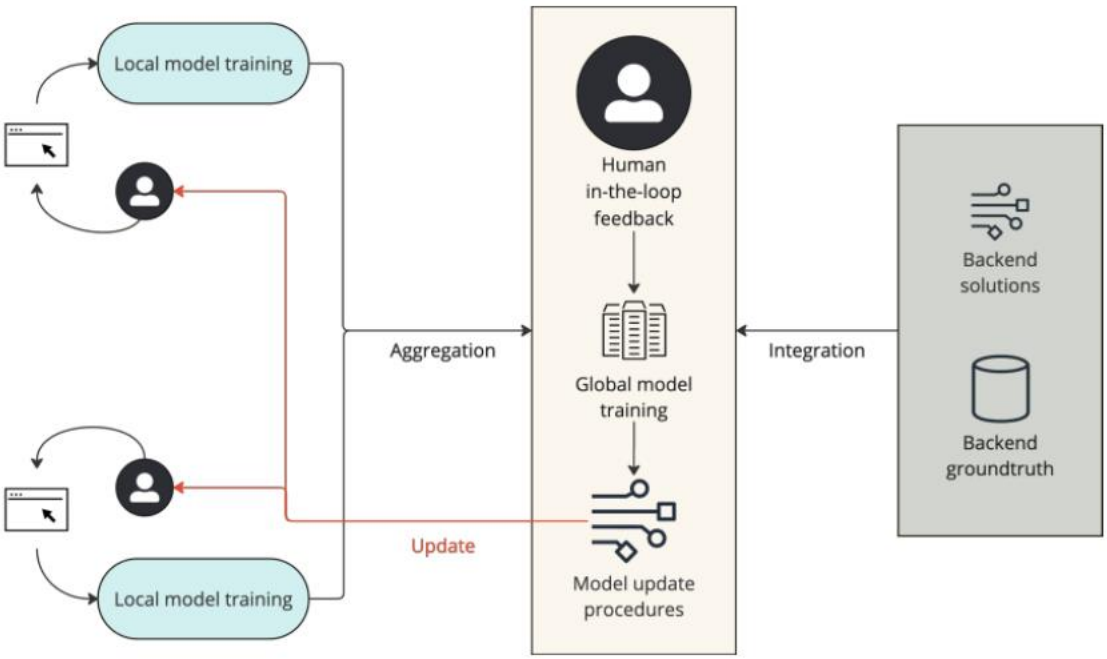



*Figure 4 - UC2 Business Process Management (BPM)*

## 3.6 Trustworthiness Challenges

The phishing detection use case exposes a range of challenges that directly reflect TRUMAN's research objectives about AI trustworthiness through explainability and usability. These challenges emerge both during the design of the system and throughout its operation in real-world conditions.

A first challenge concerns data quality and consistency. Since the detector relies on screenshots collected from heterogeneous user environments, ensuring that these images are representative, properly labelled, and free of bias is essential. Maintaining a high-quality ground truth during the early centralized training phase is equally critical, as it sets the foundation for trustworthy performance once the system transitions to distributed learning.

The second challenge relates to robustness to adversarial attacks. The detector must resist adversarial manipulation, including visual obfuscation, and adversarial examples intentionally crafted to deceive deep and multimodal models. Continual learning and careful model supervision by analysts are necessary to preserve robustness over time and prevent the system from drifting towards less reliable decision boundaries.

Explainability also represents a key requirement. The detector – i.e., its underlying model - must communicate its decisions transparently, providing clear and understandable warnings that indicate the visual or textual cues that led to a classification. This supports informed user responses and fosters trust in the technology.

Finally, the system must guarantee data protection and privacy across all phases. During federated learning, only model parameters and aggregated statistics can be exchanged; no screenshots or identifiable content may leave the user's environment. This constraint must be enforced through secure aggregation protocols and privacy-preserving mechanisms integrated into the learning pipeline.

# 4 UC3 – Detecting Branch Staff Abuse with Federated Learning

## 4.1 Context and Motivation

Financial institutions process large volumes of customer transactions through geographically distributed branch networks, supported by complex internal systems and operational workflows. While most existing fraud detection solutions focus on external threats, an important and less visible risk arises from internal misuse, where branch staff may perform financial or account-related actions on behalf of customers without their knowledge or consent. Such situations can lead to financial losses, regulatory exposure, and long-term erosion of customer trust in the institution.

Traditional approaches to detecting internal misuse often rely on centralized data aggregation and rule-based monitoring, which face limitations in scalability, adaptability, and data protection. Centralized architectures introduce privacy concerns, operational bottlenecks, and limited transparency for investigators who must interpret why specific transactions are flagged. Moreover, behavioral patterns associated with misuse frequently emerge through relationships and sequences of actions rather than isolated transactions, making purely tabular analysis insufficient.

This use case addresses these challenges through a graph-based federated learning architecture designed to operate across distributed branch environments. Each branch models its own operational data locally, capturing relationships between customers, employees, accounts, and transactions as graph structures. Instead of sharing raw data, branches contribute anonymized model updates to a central aggregation process. This design enables collective learning across the institution while preserving data locality, supporting explainability for human investigators, and creating a practical setting to validate TRUMAN's objectives on trustworthiness, privacy, and human-centric AI.

## 4.2 Actors and Roles

The use case involves several stakeholders who contribute to the operation, governance, and oversight of the system, each playing a distinct role in ensuring that detection results are reliable, secure, and ethically grounded.

### 4.2.1 Data Owners (Data Producers)

The primary data owners in this use case are the **customers and branch employees**, whose operational activities generate the data used by the system. These data include financial transactions (e.g., transfers, withdrawals, overdraft operations) as well as non-financial operational records (e.g., account updates or customer profile modifications). The data are produced during routine banking operations and stored within the bank's internal systems. In this sense, both customers and branch employees act as the sources of the behavioral and transactional data that form the basis of the graph-based modeling approach.

### 4.2.2 Data Collectors

The **bank's internal IT infrastructure** acts as the data collector. Transactional and operational records generated through banking systems are collected and structured for analytical use. During this stage, relevant features such as transaction attributes, customer metadata, and employee identifiers are extracted and organized into a graph representation that links customers, employees, and transactions. Data preprocessing, anonymization, and preparation for model training are performed within the bank's secure environment.

### 4.2.3 Model Trainers

The **AI / IT engineers** who are considered as bank employees working for the same company, act as the model trainers. They design and implement the federated graph learning framework, construct the graph structures representing relationships between entities, and train the machine learning models. Their responsibilities include managing the training pipeline, configuring the federated learning process, performing model updates, and ensuring the system operates securely and efficiently within the bank's infrastructure.

### 4.2.4 Model Executors

The **AI system deployed within the bank's infrastructure** acts as the model executor. Once the model has been trained, it is applied to new transaction data to generate predictions. The system processes incoming operational data, analyzes patterns within the graph structure, and assigns a fraud classification outcome (fraud or non-fraud) to transactions. This execution occurs automatically within the bank's internal environment.

### 4.2.5 End Users (Model Consumers)

The **fraud investigators (internal auditors)** act as the end users of the system. They receive reports containing transactions flagged by the model as suspicious and review the relevant case details. Based on their investigation, they confirm or dismiss the alerts. Their decisions provide feedback that can later be incorporated into the training process to improve the model. Through this interaction, fraud investigators serve as humans-in-the-loop within the operational workflow.

## 4.3 Data Description

The data used in this use case consists of structured, transaction-level and operational records generated during normal banking activities at the branch level. These records are processed locally and represented in graph form to capture relationships between entities and sequences of actions.

In the initial phase, historical transaction logs are used to establish baseline behavioral patterns and validate the modeling approach. These data support the development of local branch models and allow fraud investigators to confirm the relevance and quality of detected patterns.

In the operational phase, each branch continuously processes new data streams locally and updates its model accordingly. Model parameters are then shared through a federated learning framework and aggregated centrally, without transferring any raw or identifiable data.

- **Financial data** includes transaction amount, transaction type (e.g., money transfer, withdrawal, credit payment, account opening), and timestamp.
- **Customer-related features** include age, education level, and income level.
- **Branch and employee metadata** include the employee role linked to the transaction and the branch identifier.

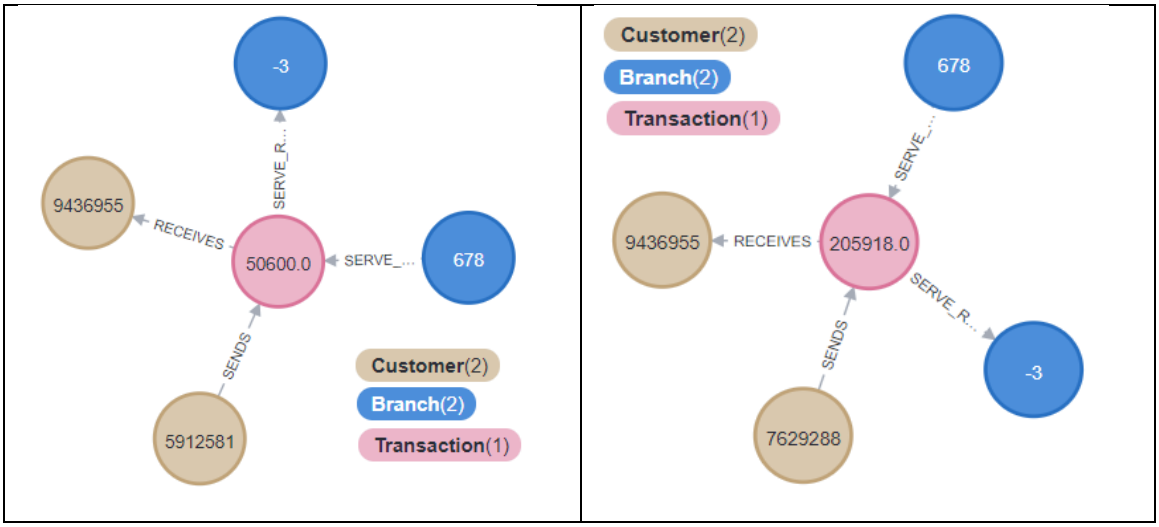



*Figure 5 - Branch-Level Transaction Graph Representation*

These inputs are represented in a graph structure as shown in Figure 5, where nodes correspond to customers, branches, and transactions, and edges represent their relationships. The model output consists of a suspicion score of 0 or 1 for each transaction, indicating whether the transaction appears to be fraudulent or not. This score is used by fraud investigators to validate cases and enrich the labeling process rather than to trigger automated enforcement.

## 4.4 AI Tools and Techniques

The use case is built around a federated, graph-based learning architecture designed to detect insider fraud while ensuring that sensitive banking data remains local to each branch.

At the core of the system are Graph Neural Networks (GNNs), which model the relationships between customers, employees, and transactions as branch-level graphs. In these graphs, nodes represent customers, branches, and individual transactions, while edges capture their operational relationships, such as an employee authorizing a transaction for a customer. Each node and edge

is associated with transaction-level and customer-related features, including transaction amount, transaction type, timestamp, and basic customer attributes such as age, education level, and income level. Models such as GraphSAGE or Node2Vec are used to learn low-dimensional embeddings that capture structural and behavioral patterns within each branch's transaction network.

- The learning process follows a federated learning paradigm. A global GNN architecture is first defined at a central master node and initialized with a shared set of weights. This global model is then distributed to all branches, where each branch trains a local GNN on its own transaction graph. Training is primarily unsupervised or self-supervised, as confirmed fraud labels are limited, allowing the model to learn general patterns of normal and abnormal behavior from the structure of the data itself.

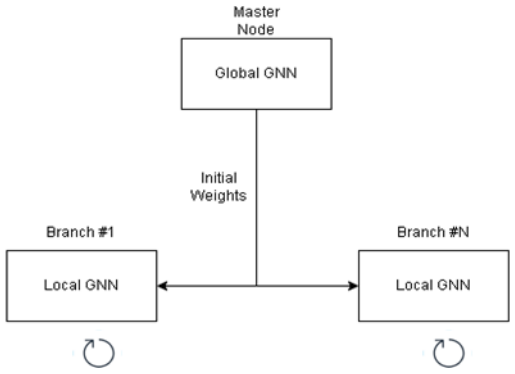



*Figure 6 - Federated GNN Architecture with Local Branch Training*

- Once local training is completed, branches do not transmit any raw transaction or customer data. Instead, they send either model weight updates or learned embeddings back to the master node. The master node applies Federated Averaging (FedAvg) to aggregate these updates and produce a new global model that reflects knowledge learned across all participating branches, while preserving data locality and privacy.

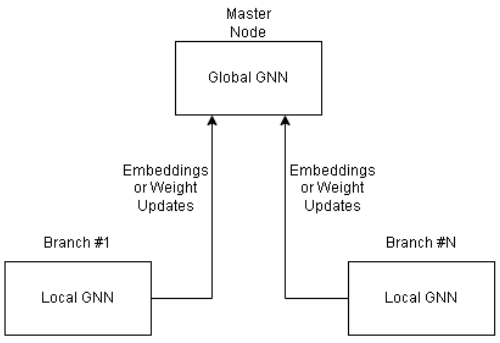



*Figure 7 - Secure Update Sharing and Federated Aggregation Process*

On top of the aggregated embeddings, a fraud classification layer is trained at the master node using a small set of confirmed fraud labels combined with label propagation techniques. This enables the system to spread limited labels across similar nodes in the embedding space and improve detection performance in a semi-supervised manner. The classifier can be implemented using Logistic Regression, XGBoost, or a simple Multi-Layer Perceptron (MLP), depending on the desired balance between interpretability and predictive performance. The resulting global GNN and fraud classifier are then redistributed to all branches, where they are used locally to score new transactions. Each transaction is assigned a binary suspicion label (0 or 1), indicating whether it is classified as legitimate or fraudulent. These scores are compiled into alert reports that are shared with fraud investigators for review and validation.

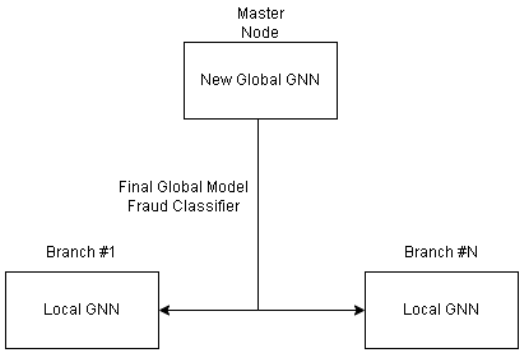



*Figure 8 - Global Model Redistribution and Local Fraud Scoring Pipeline*

- For graph construction, querying, and analysis, tools such as Neo4j and NetworkX are used to support flexible experimentation with graph structures and relationship modeling during development and evaluation.

This combination of graph learning, federated aggregation, and human-in-the-loop validation ensures that the system remains privacy-preserving, adaptable to changing fraud patterns, and aligned with TRUMAN's principles of trustworthy and transparent AI.

## 4.5 Requirements

*Table 4 - Requirements for UC3*

| Reference | Requirement | Description |
|---|---|---|
| **R1** | Branch-Level Fraud Detection Module | Module that performs local inference on branch transaction data and generates suspicion scores for individual transactions. |
| **R1.1** | Local Graph Training Infrastructure | Module that trains a Graph Neural Network (GNN) on each branch's local transaction graph using unsupervised or semi-supervised learning. |
| **R1.2** | Secure Update Sharing Mechanism | Module that transmits embeddings or model weight updates from branches to the master |

| | | |
|---|---|---|
| | | node without exposing raw transaction or personal data. |
| **R2** | Master Aggregation and Global Model Module | Module that aggregates branch updates using federated learning techniques and produces a global GNN model. |
| **R2.1** | Federated Aggregation Engine | Component that applies Federated Averaging (FedAvg) or robust aggregation methods to combine branch-level updates into a stable global model. |
| **R2.2** | Fraud Classifier Training Module | Module that trains a transaction-level classifier on aggregated embeddings using limited fraud labels and label propagation techniques. |
| **R3** | Alert Generation and Reporting Module | Module that produces structured fraud alerts and summary reports and distributes them to relevant business units via existing communication channels (e.g. e-mail). |
| **R3.1** | Human-in-the-Loop Feedback Channel | Mechanism that allows fraud investigators to confirm or dismiss alerts and feed validated labels back into the training pipeline. |
| **R4** | Privacy and Security Enforcement Layer | Module that ensures raw data remains at the branch, enforces access controls, and supports privacy-preserving techniques such as secure aggregation and differential privacy. |
| **R5** | System Monitoring and Audit Module | Module that enables AI/IT engineers to monitor system performance, logs, and model behavior to ensure reliability and compliance. |

As summarized in Table 4, the insider fraud detection procedure aims to design and implement a federated, graph-based AI system capable of identifying suspicious branch-level transactions while preserving the privacy of both customers and employees. The system operates across a distributed architecture in which each branch maintains and processes its own transaction data locally, represented as a graph structure linking customers, employees, and transactions.

The solution must support a multi-stage workflow. In the first stage, a global GNN model is initialized at the master node and distributed to branches. Each branch then performs local training on its own transaction graph, learning embeddings that capture patterns of customer behavior, employee involvement, and transaction characteristics. Only model parameters or embeddings are shared back with the master node, ensuring that no raw personal or financial data leaves the branch environment.

At the master node, the system must aggregate updates from all branches using federated learning techniques, such as Federated Averaging (FedAvg), to produce a stable and robust global model. On top of this global representation, a fraud classifier must be trained using a combination of limited confirmed fraud labels and label propagation methods to improve detection performance in low-label environments.

The resulting model must be redistributed to branches to enable continuous local scoring of transactions. When suspicious cases are identified, the system must generate structured alerts and periodic reports that are delivered to relevant business units through existing reporting and communication workflows, rather than a dedicated user interface.

The use case further requires the integration of a human-in-the-loop process, in which fraud investigators review flagged cases, validate outcomes, and provide feedback that is incorporated into subsequent training cycles. AI/IT engineers must be able to oversee system operation through logs and performance metrics, ensuring technical stability, compliance, and robustness over time.

Finally, the system must enforce strong privacy, security, and governance requirements, including role-based access control, secure transmission of model updates, and alignment with applicable data protection and ethical standards. This ensures that the system remains trustworthy, legally compliant, and suitable for deployment in sensitive financial and regulatory environments.

## 4.6 Trustworthiness Challenges

This use case raises several challenges that directly relate to TRUMAN's objectives on trustworthiness, robustness, and human-centric AI.

A primary challenge is privacy and confidentiality, particularly compliance with GDPR and KVKK.[1] Sensitive customer and employee data must never leave the branch environment, even during collaborative learning. For example, transaction histories, customer demographics, and employee–transaction relationships are processed locally at each branch. Only model updates or embeddings are transmitted to the master node. This design supports data minimization and purpose limitation principles under GDPR and KVKK, and requires privacy-enhancing technologies (PETs) such as secure aggregation and strict role-based access controls to ensure that no personal or branch-level information is disclosed from the exchanged updates.

A second challenge is robustness in a distributed setting. Branches operate under different conditions - for example, some handle high volumes of cash withdrawals, while others focus more on account openings or money transfers. These non-IID data distributions can cause the global model to learn patterns that reflect dominant branches rather than the full network. In addition, incorrect or biased feedback from a single branch, such as repeatedly labeling suspicious transactions as legitimate, could weaken the global model. Continuous monitoring of aggregated updates and branch-level performance is therefore required to maintain stable and reliable detection quality.

Fairness and bias represent another concern. Differences in customer demographics or branch operations can lead to uneven error rates. For instance, a branch serving mostly elderly

[1] **KVKK (Kişisel Verilerin Korunması Kanunu)** is the Turkish Personal Data Protection Law (Law No. 6698), which regulates the processing and protection of personal data in Türkiye. It is broadly aligned with the principles of the EU General Data Protection Regulation (GDPR). https://www.kvkk.gov.tr/en

customers or low-income groups may experience higher false positives if the model associates certain transaction patterns with elevated risk. Regular evaluation across branches and customer segments is necessary to ensure that no group is systematically disadvantaged.

Finally, explainability and human oversight are critical for both operational trust and regulatory accountability. Fraud investigators must be able to understand why a transaction was flagged - for example, whether it was driven by an unusually high transaction amount, repeated actions by the same employee across multiple customer accounts, or an abnormal transaction sequence over time. Clear explanations support informed decision-making and help demonstrate compliance with transparency obligations under GDPR and KVKK.

# 5 UC4 - LLM-powered chatbot for Parkinson's disease

## 5.1 Context and Motivation

AI applications, especially Large Language Models (LLMs), are becoming part of everyday life, and adoption continues to accelerate. LLMs now appear across products and services, from search engines to customer support and decision support systems. Increasingly, these systems are used for higher risk tasks. In healthcare, the conversational nature of LLMs makes them well suited for applications such as patient empowerment and disease management. Although these applications raise privacy and safety risks, those risks are often moderate relative to the potential patient benefits.

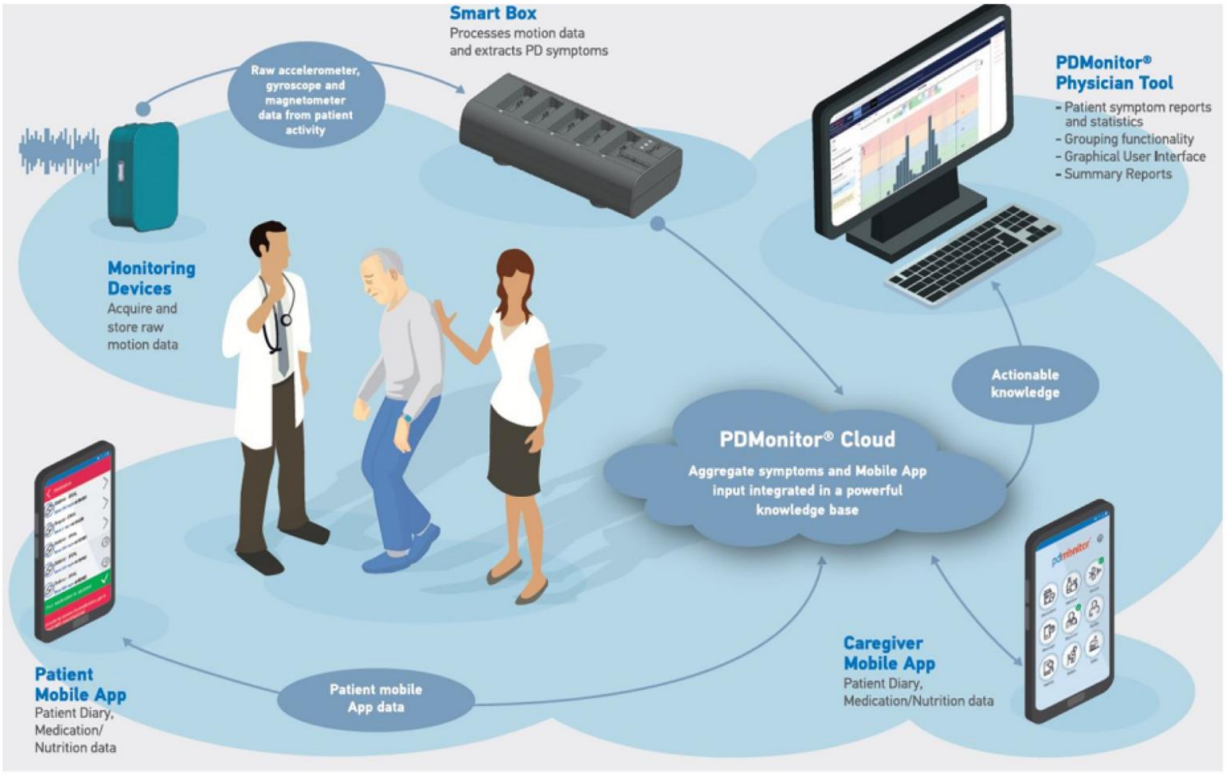
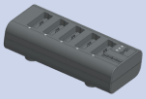
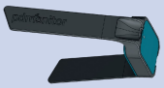



*Figure 9 - The PDMonitor ecosystem.*

PD Neurotechnology Ltd is the manufacturer of PDMonitor, a wearable medical device designed to help monitor Parkinson's disease symptoms during everyday activities and provide objective summaries for clinicians. The PDMonitor comprises: (1) the SmartBox (SB), which is used to collect, process, and upload data to the cloud; (2) five (5) sensing monitoring devices (MDs), consisting of wearable sensors used to collect movement data; and (3) accessories for attaching the MDs to the patient's waist, shanks, and wrists. You can see the PDMonitor ecosystem in Figure 9 and the PDMonitor device components in Table 5.

*Table 5 - PDMonitor device components*

| Component | Description |
|---|---|
|  | The SmartBox (SM), which is used to collect, process, and upload data to the cloud. |
|  | The PDMonitor monitoring devices (MDs), a set of five (5) wearable sensors used to collect movement data. |
|  | Clipframe, used to attach an MD to a patient's belt. |
|  | Strapframe, used to attach an MD to a patient's shank (leg). |
|  | Wristband, used to attach an MD to a patient's wrist. |

PDMonitor is supported by a broader ecosystem of hardware and software, including a mobile application for patients and caregivers (see Figure 10), as well as a mobile application and a web-based dashboard for physicians (Physician Tool). Together, these components help collect, review, and share nutrition, medication and symptom and activity data, supporting day-to-day disease management and more informed clinical decisions.

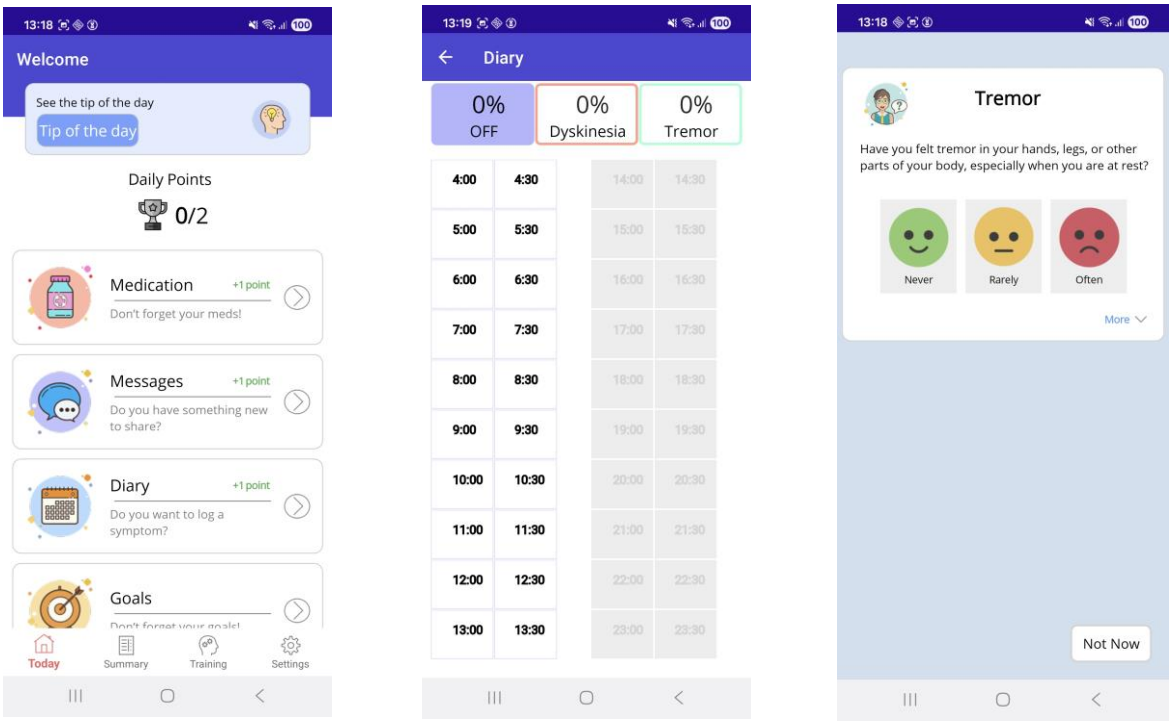



*Figure 10 - A selection of screens from the patient's mobile application: (1) the app's welcome screen, (2) the PD diary interface, and (3) the symptom recording functionality, illustrated here with tremor as an example.*

Effective management of Parkinson's disease requires close collaboration between patients and physicians to monitor symptoms and adjust treatment over time. Because symptom severity can fluctuate throughout the day and may be influenced by medication, meals, and daily activities, both patients and clinicians rely on accurate, real-world information to guide care. Capturing these daily patterns is essential for patients to communicate their experiences clearly and for physicians to make informed therapeutic decisions.

A common clinical practice in Parkinson's care is the use of a PD diary, in which patients systematically record their symptoms, along with detailed information about medication intake, the type and timing of meals, and other relevant daily events, commonly recorded in a paper-based diary or, alternatively, captured through digital means such as the patient's mobile application (see Figure 10). A PD diary helps capture the day-to-day variability of PD, which can be difficult to summarize during occasional clinic visits. With PDMonitor, these entries add patient-reported context to the device's measurements and help clinicians interpret fluctuations against real-world routines. However, PD diaries can be cumbersome, as they must be completed multiple times throughout the day. Some patients also face practical physical challenges, including dexterity limitations that make writing difficult.

In parallel, physicians must manage the volume of information generated by both PD diaries and PDMonitor conducted through the web-based dashboard or, alternatively, via the physician's mobile application), which can make care more complex and time-intensive. Reviewing pages of

patient-reported entries alongside continuous device data requires careful attention, especially when trying to identify meaningful patterns or changes in symptoms. Within the limited time of routine consultations, clinicians must quickly interpret this information and translate it into practical treatment decisions. As the amount of data grows, organizing and making sense of it in a clear and efficient way becomes an increasing challenge.

Recognizing these practical challenges, PDN is developing an LLM-powered virtual assistant, the PDNeurobot, which can:

1. **Patient support:** converse with patients and caregivers to capture diary entries through natural dialogue and support patient empowerment, customer-support guidance (e.g., best practices and troubleshooting), and disease self-management.
2. **Clinical summary:** give physicians a fast overview of patient-generated data by summarizing key updates and enabling follow-up questions about specific patients.

Currently, the LLM assistant is under active development and remains in an alpha stage; it is not integrated into PDN's commercial product lineup and is not offered as a commercial service but is being developed and evaluated within the context of this case study. **The non-commercial PDMonitor system is going to be indicated as PDMonitor (TRUMAN) system.**

At the same time, alongside these benefits, PDN and the TRUMAN consortium recognize the risks of integrating an LLM-powered assistant, the **PDNeurobot**, into a medical-device ecosystem, including generating, processing, and storing sensitive data and granting the assistant access to it. This recognition motivated the design and implementation of a use case to investigate the robustness of the system, particularly with respect to fairness, reliability, security, and privacy risks associated with the use of LLM-based assistants.

## 5.2 Use Case Scenario

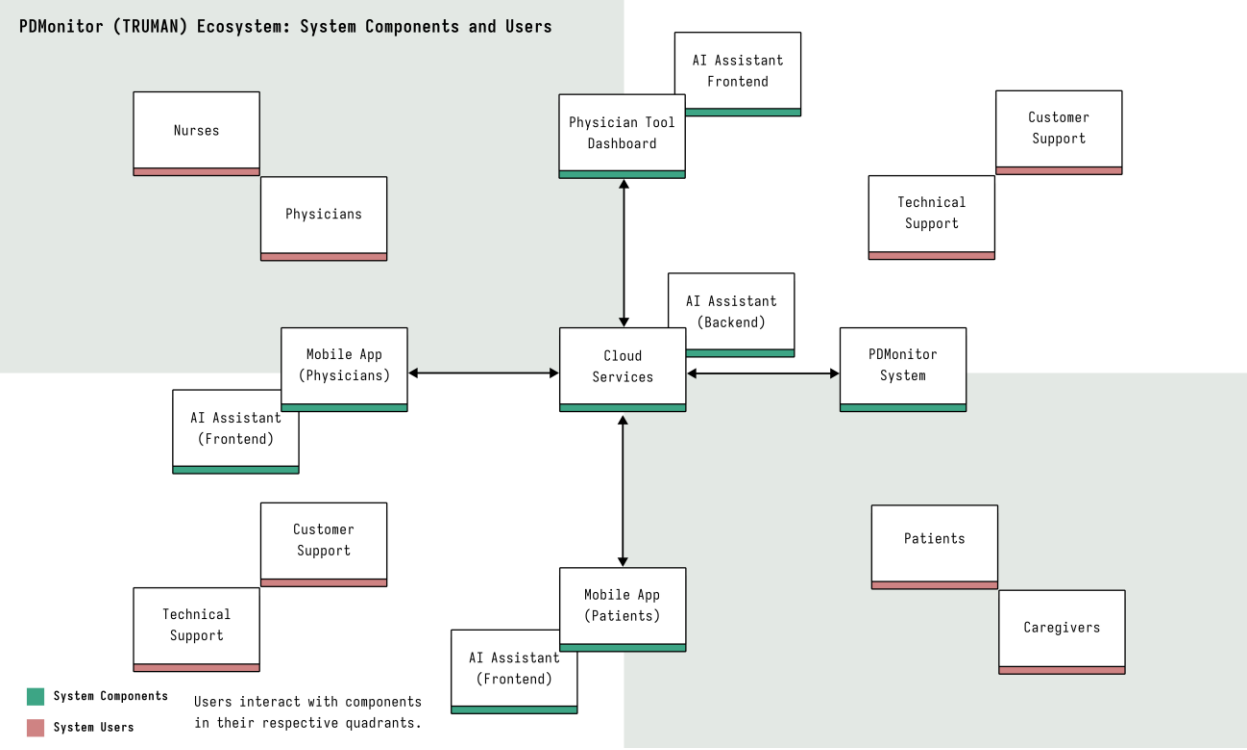



*Figure 11 - The components of the PDMonitor (TRUMAN) system*

During the implementation of use case 4 (UC4), the LLM-powered assistant will be integrated into the PDMonitor ecosystem (see Figure 11) to support user empowerment, user engagement, and disease management. The case study will examine user interactions with the AI assistant (see Figure 12), which will incorporate multimodal data to enable personalization and better address queries related to Parkinson's disease management and individual patients. In this context, multimodal data refers to the integration of patient-reported information (e.g., symptoms, medication timing and dosage, meal type and timing), wearable device measurements from PDMonitor, and contextual inputs collected through structured digital interactions. By combining these data streams, the AI assistant can provide more tailored feedback, highlight relevant patterns, and respond to questions with patient-specific insights rather than generic recommendations.

The use case will simulate routine use of the PDMonitor ecosystem, with the AI assistant integrated into the existing workflow. Users will engage with the system as they would in everyday care: patients and caregivers will record symptoms and contextual information, while clinicians review device-generated summaries and reports. Within this loop, the AI assistant will support diary completion, prompt for missing details, summarize trends, and contributes to improved communication between patients, caregivers and physicians through enhanced data and insights. This setup ensures a coherent and realistic representation of how the ecosystem operates in

practice, highlighting the assistant's role as an embedded support tool rather than a standalone component.

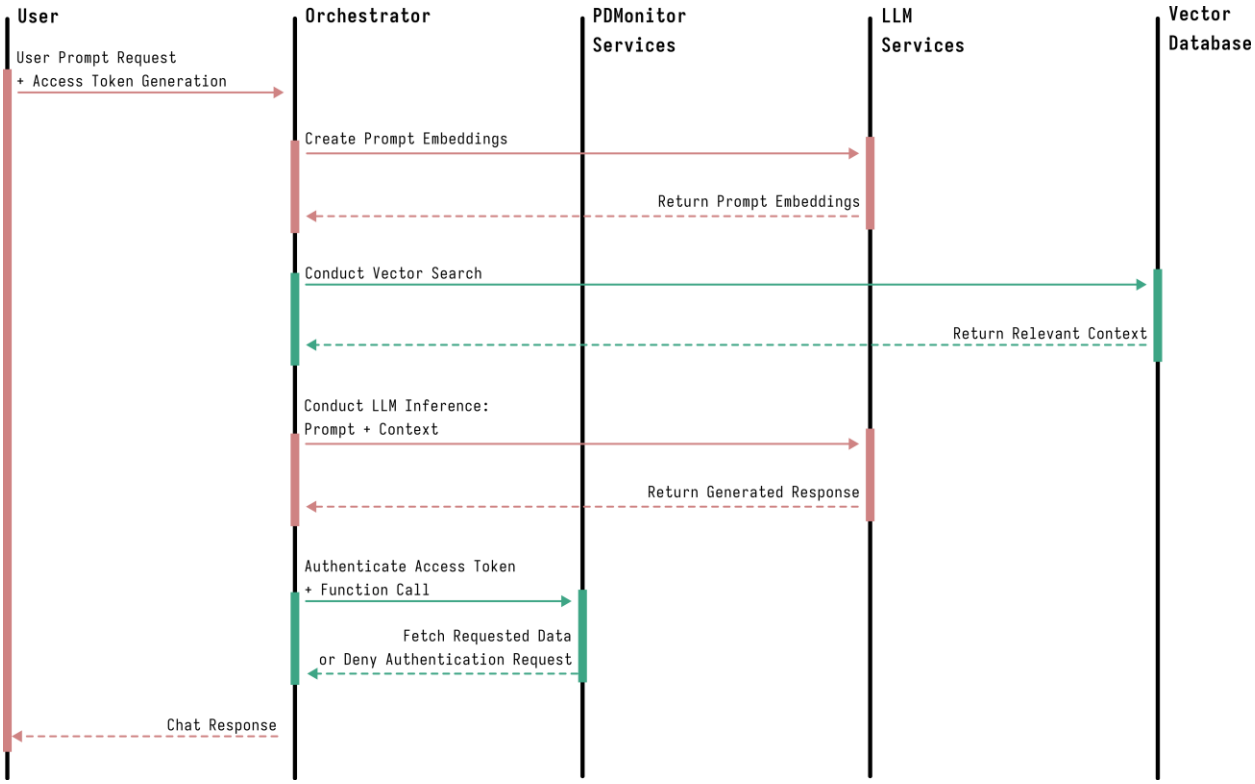



*Figure 12 - A sequence diagram detailing the outlined events of one interaction cycle between a user and the AI Assistant.*

The use case will mainly revolve around the following two scenarios (see Figure 13):

**Patients - AI Assistant:** In this scenario, patients or caregivers (caregivers can also be users of this scenario) interact with the AI assistant to support disease management, patient support and empowerment. Users can provide health-related information such as medications, nutrition, and symptoms, and receive personalized guidance, reminders, and educational insights. The assistant helps users better understand their condition, supports adherence to treatment plans, and offers actionable recommendations to improve daily health management. In this scenario, the AI Assistant will be tailored based on a custom, patient-focused, instruction set, whereby it is instructed to operate as a patient's companion, offering balanced well-being recommendations while providing explicit disclaimers that it does not offer medical advice or substitute for a physician.

**Clinicians - AI Assistant:** In this scenario, clinicians interact with the AI assistant to enhance patient management, clinical decision-making, and workflow efficiency. The assistant supports the review and analysis of patient data (e.g., medical history, medications, symptoms), provides evidence-based insights, assists with treatment planning, and enables ongoing monitoring of

patient progress. It may also help identify potential risks, streamline documentation, and ensure alignment with current clinical guidelines to improve the quality of care. In this scenario, the AI Assistant will be tailored based on a custom, physician-focused instruction set, whereby it is instructed to operate as a physician's assistant, providing balanced summaries of patient symptoms and facilitating easy access to relevant medical guidelines, while including explicit disclaimers that its outputs should not be used without the physician's expert judgment and appropriate consideration of the patient's specific clinical context.

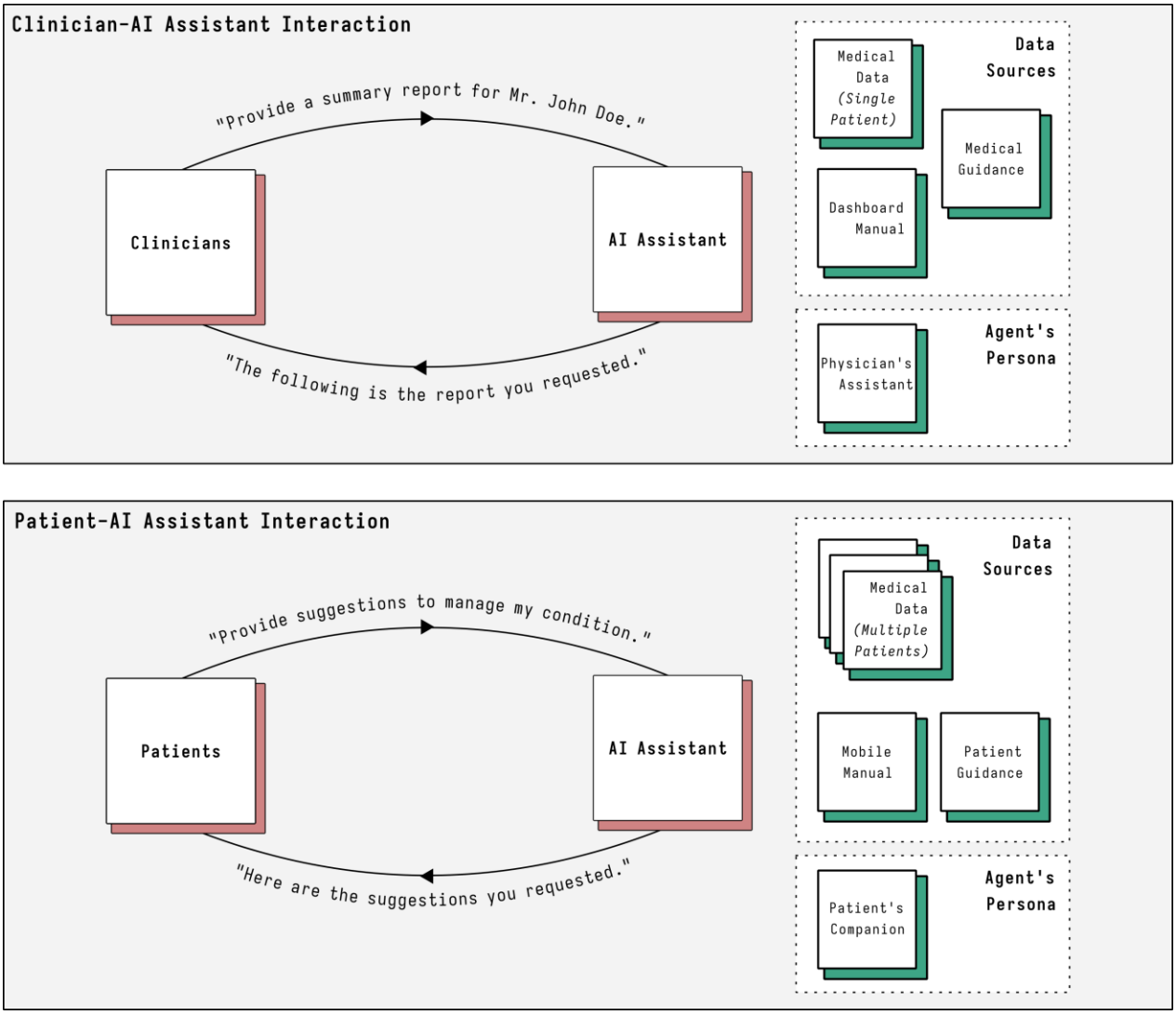



*Figure 13 – Visual description of the two main PDN use case scenarios. One for the clinicians and one for the patients.*

Given the conversational nature of LLMs, interactions between patients, caregivers, physicians, and the AI assistant can vary significantly and may be largely unstructured. Nevertheless, the intended use can be broadly categorized into a set of common themes that explicitly support both patient-facing and clinician-facing perspectives, as outlined below:

1. **Nutrition:** Support related to dietary habits, including guidance on nutrition, meal planning, and dietary restrictions. For patients and caregivers, this enables tracking and improvement of daily habits, while for physicians, it provides structured insight into lifestyle factors through nutrition data (e.g., food intake, nutritional content, and timing) that may influence disease progression and treatment outcomes.
2. **Medication:** Assistance with medication management, including information on prescribed drugs, dosage, frequency, administration methods, and adherence tracking. Patients and caregivers benefit from reminders and clarification, while physicians can review medication data (e.g., adherence patterns and timestamps) to assess treatment effectiveness and adjust prescriptions accordingly.
3. **Symptoms:** Reporting, monitoring, and preliminary interpretation of symptoms. Patients and caregivers can log and describe symptoms in free-text or structured formats, while physicians can access aggregated medical data, including clinical observations and disease progression, to support diagnosis, monitoring, and clinical decision-making.
4. **Advice:** Provision of general health-related guidance based on user input and available knowledge. For patients, this includes accessible explanations and recommendations for self-management, while for physicians, it can support clinical reasoning through the integration of external knowledge sources, expert input, and patient-specific context.
5. **Records:** Access to, summarization of, and interaction with patient-related information. Patients and caregivers can review and contribute to their own data, while physicians can access a comprehensive view that includes medical history, cognitive assessments, activity data (e.g., mobility patterns), and interaction logs, supporting a more holistic understanding of patient status.
6. **Support**: Ongoing assistance and system interaction, including reminders, troubleshooting, and guidance. Patients and caregivers receive continuous support in device use and care routines, while physicians can benefit from structured outputs, summaries, and insights derived from customer support data, interaction logs, and system-generated information to support care coordination and oversight.

Figure 14 illustrates various user journeys describing the interactions between patients or caregivers, as well as physicians, and the AI Assistant. In this figure, each row represents a complete interaction sequence over time between the user and the AI assistant, outlining the progression of exchanges, without including system responses.

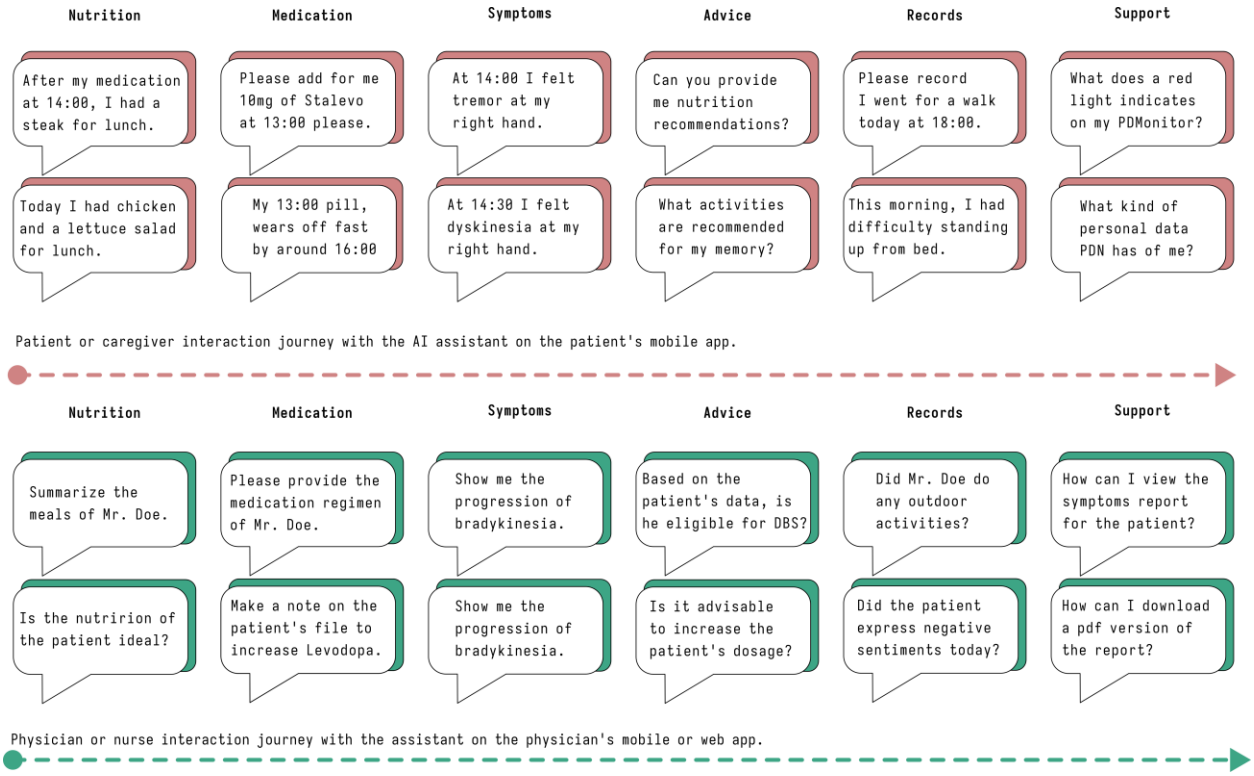



*Figure 14 - User journeys of patients, caregivers, physicians, and nurses interacting with the AI assistant; each line represents a complete interaction sequence over time between the user and the AI assistant, without including system responses.*

## 5.3 Actors and Roles

PDMonitor and its associated products and services form an ecosystem in which users with varying levels of technical skill and domain knowledge interact across multiple environments. These users range from older adults with limited digital literacy to caregivers and clinicians who use such tools routinely.

### 5.3.1 Data Owners

1. **Patients:** Patients are the primary data owners of all personal and health-related data generated through their use of the system, including wearable device data, self-reported information, and interactions with the mobile application and chatbot.
2. **Caregivers:** Caregivers are data owners of any personal data they provide about themselves (e.g., identification and contact details). They are not owners of patient data but may contribute data on behalf of patients, subject to patient consent.
3. **Clinicians:** Clinicians are data owners of their own personal and professional data (e.g., name, contact details, professional credentials) provided to the system. They do not own patient data but are granted authorized access to it for clinical purposes.

### 5.3.2 Data Collectors

1. **Research Engineers:** Research Engineers collect and process data generated by the PDNeurobot system, including data from wearable devices, the mobile application, and chatbot interactions. This is done for development, testing, validation, and improvement of system functionalities and algorithms. Their responsibilities also include data cleaning, curation, annotation, and general preparation of datasets for use in analytical pipelines and machine learning models, typically using anonymized or pseudonymized data.
2. **Technical Support:** Technical Support personnel collect data necessary to diagnose, troubleshoot, and resolve technical issues related to the PDNeurobot device, mobile application, and platform. This may include system logs, error reports, and limited user-related information. They may also perform basic data cleaning and organization to ensure data quality for research purposes and model execution.
3. **Customer Support:** Customer Support staff collect and manage user-provided information during support interactions, such as account details, contact information, and issue descriptions. They may also contribute to data organization, categorization, and preliminary cleaning to support reporting, system improvements, and preparation of data for downstream analytical or model-related use.

### 5.3.3 Model Trainers

1. **Research Engineers:** Research Engineers are responsible for training, validating, and optimizing models using data collected through the PDNeurobot system. Their activities include dataset preparation (cleaning, curation, annotation), feature engineering, model selection, training, evaluation, and performance monitoring. They typically work with anonymized or pseudonymized data and ensure that models are scientifically robust and aligned with research objectives.
2. **Software Engineers:** Software Engineers support the model training process by developing and maintaining the infrastructure, pipelines, and tools required for data processing, model training, and deployment. They ensure scalability, reproducibility, and integration of trained models into the PDNeurobot system, while also contributing to data preprocessing workflows and maintaining data quality standards required for reliable model performance.

### 5.3.4 Model Executors

1. **Research Engineers:** Research Engineers define, configure, and execute the data pipelines that support model operation, including preprocessing steps, inference workflows, and retrieval components such as Retrieval-Augmented Generation (RAG). They ensure that models are run on new data queries in a controlled manner and that retrieved knowledge is appropriately incorporated to refine, contextualize, or adjust model outputs before final outcomes are produced.
2. **Software Engineers:** Software Engineers implement, maintain, and operate the production pipelines through which models are executed on incoming data and user queries. This includes orchestration of inference services, integration of RAG-based

knowledge sources, and management of the technical components that influence or enrich model outputs. Their role is to ensure that the end-to-end execution pipeline reliably produces the final system outcomes.

### 5.3.5 End-Users

1. **Patients:** Patients are the primary end users of the PDNeurobot system. They wear the device, interact with the mobile application and chatbot, provide consent, and input disease-related information through diary entries and system interactions.
2. **Caregivers:** Caregivers support patients in using the system, including device setup, daily operation, and data entry. They may act as secondary users, particularly when patients have limited dexterity or digital skills.
3. **Clinicians:** Clinicians (e.g., physicians, nurses) use the system to access summarized outputs and longitudinal patient data. They interpret this information within a clinical context to support monitoring, follow-up, and care decision-making.

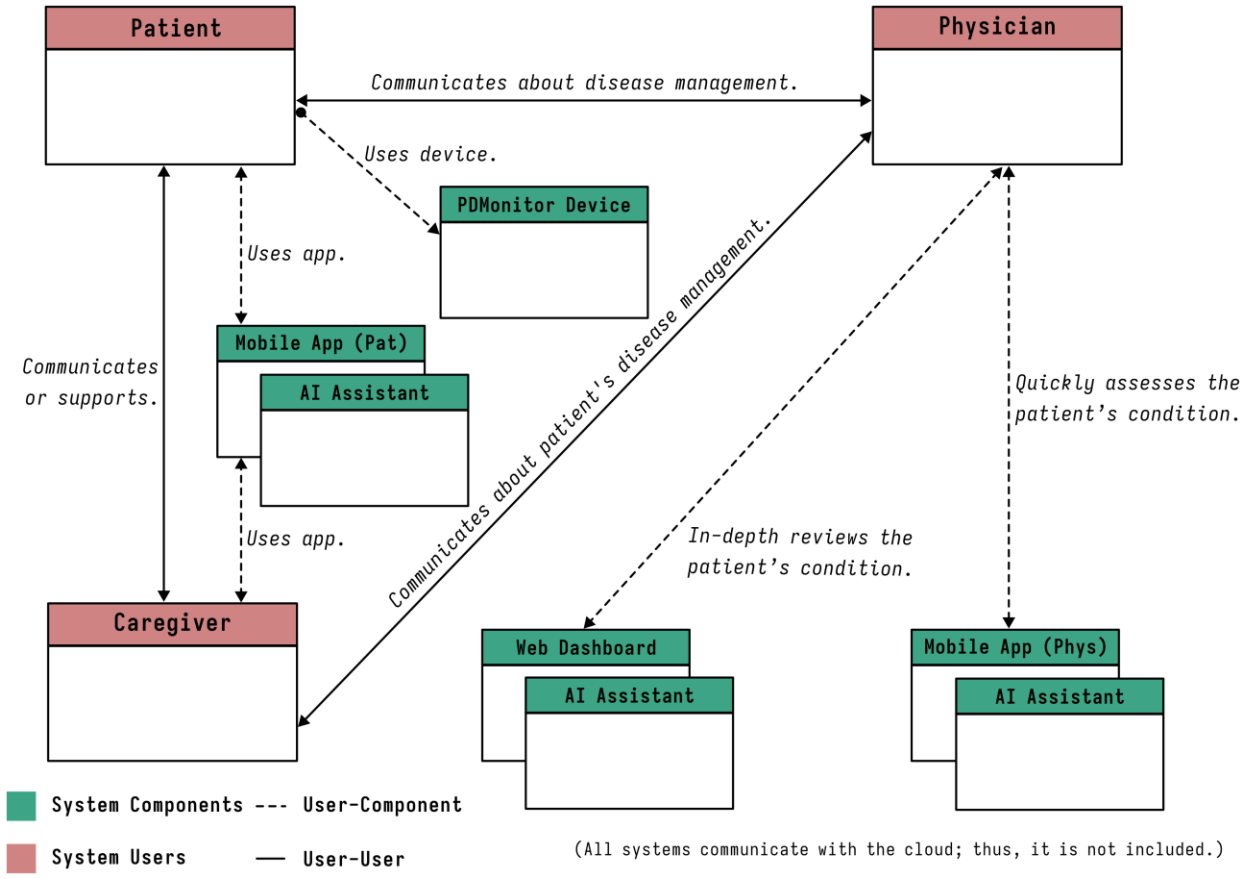



*Figure 15 - Interactions between users and the components of the PDMonitor system. Note, not all interactions are bidirectional. In the diagram, Mobile App (Pat) refers to the patient's mobile app, while Mobile App (Phys) refers to the physician's mobile app.*

### 5.3.6 Supporting Roles

The PDMonitor ecosystem includes additional user groups who are not intended to interact with the chatbot as regular users. They may engage only when needed, but they still play an important role in shaping the overall user experience. These groups include:

1. **Customer support:** Customer Support agents handle user questions and issues, guide users through common tasks, and escalate technical or clinical concerns to the appropriate teams.
2. **Technical support:** Technical Support agents manage accounts, roles, and access permissions; maintain integrations and configuration; and enforce security controls across the ecosystem.
3. **Clinical staff:** Clinical staff assist with onboarding, troubleshooting, and workflow coordination, and help ensure data is captured and routed to the right place.

These roles matter because PDNeurobot can be improved using insights gathered across the ecosystem, particularly from customer support and clinical staff. For example, customer support agents interact regularly with patients and may identify recurring issues, usability barriers, and context about patient needs; these insights can be incorporated into the assistant's supporting knowledge and guidance to make it more useful and safer in real-world use. During the implementation of the use case, not all these user groups may be directly involved. However, listing them provides a clearer picture of the ecosystem, its complexity, and the interactions that can shape how the system is used in practice.

### 5.3.7 Role Interactions

The PDMonitor ecosystem involves recurring interactions among roles and components (see Figure 15) that generate data, inform assistant development, and provide domain-specific knowledge that guides the assistant's outputs:

1. **Interaction 1** (patient interacting with the AI assistant): Patients report their symptoms, medication timing and dosage, meal type and timing, and other relevant daily context, while PDNeurobot prompts for missing details and structures the inputs to support accurate and complete diary completion (see Figure 13 and Figure 15).
2. **Interaction 2** (caregiver interacting with the AI assistant): Caregivers assist with device setup and reporting; PDNeurobot adapts to caregiver-mediated input when patients need help (see Figure 13).
3. **Interaction 3** (patient interacting with the clinician): Patients and caregivers share updates during consultations, and clinicians ask follow-up questions to clarify symptoms, medication effects, and daily context (see Figure 15).
4. **Interaction 4** (caregiver interacting with the clinician): Caregivers provide observations about the patient's daily functioning, behavioral changes, medication adherence, and symptom fluctuations (see Figure 15).

5. **Interaction 5** (clinician interacting with the AI assistant): PDNeurobot summarizes patient-, caregiver-, and device-reported information, and clinicians can ask follow-up questions to explore specific updates (see Figure 13 and Figure 15).
6. **Interaction 6** (patient interacting with customer support): Patients contact customer support for onboarding and troubleshooting, and customer support provides guidance while capturing recurring issues and user needs.
7. **Interaction 7** (caregiver interacting with customer support): Caregivers contact customer support for assistance with setup, account management, and troubleshooting on behalf of the patient. Customer support provides guidance and clarifies caregiver-specific needs to improve the overall support experience.
8. **Interaction 8** (customer support interacting with technical support): Customer support escalates issues and share feedback, and technical support address them through configuration, access control, and integration changes.

## 5.4 Data Description

Detailed information regarding the use case data, including data sources and data collection practices, can be found in the deliverable "D1.2 - Data Management Plan". In summary, this use case involves the collection, generation, and processing of the following multimodal data:

1. **Medical Data:** Information related to a patient's physical health, including reported symptoms, clinical observations, diagnostic findings, disease progression notes, and relevant medical history that supports ongoing care and monitoring.
2. **Medication Data:** Records of medication use, including prescribed drugs, dosages, frequency of intake, administration method, and timestamps to track adherence and support treatment evaluation.
3. **Nutrition Data:** Logs of dietary habits and meal patterns, including food types, portion sizes, nutritional content, and dates and times. These data support assessment of lifestyle factors that may influence overall health and medication effectiveness.
4. **Activity Data:** Information describing a patient's daily activity patterns, including classifications such as Lying, Resting, and Active, as detected by the PDMonitor device. These data support assessment of mobility, behavioral trends, and overall functional status.
5. **Customer Support Data:** Content from the customer support knowledge base and interactions between patients or caregivers and PD Neurotechnology customer support representatives. This may include manuals, best practices, troubleshooting guidance, and insights into PDMonitor use captured during support interactions. Any personally identifiable information (PII) present in source materials will be removed before use.
6. **Personal Data:** Identifying information about patients, such as name, age, contact details, demographic characteristics, and other information needed for identification and delivery of personalized services.
7. **Cognitive Data:** Information related to cognitive functioning and psychological well-being, including cognitive assessment results, mental health evaluations, mood tracking, and behavioral indicators relevant to emotional status.

8. **LLM Prompt Input:** Free-text conversation transcripts and interaction logs between patients, caregivers, or physicians and the chatbot, including user queries, system responses, and relevant context. As free-text data, it may contain any information provided by the user, including personally identifiable information (PII) and financial data.
9. **Mobile App Input:** Free-text entries submitted by patients or caregivers through the mobile application. This may include personal observations, notes, or information not captured by other data categories. These entries may contain identifying information and are used to complement other collected information for ongoing care and monitoring.
10. **External Knowledge:** Information from publicly available, reputable medical and scientific sources, insights from consulted experts, and internal knowledge held by PD Neurotechnology, such as expertise from customer support agents. This supports the chatbot via retrieval-augmented generation (RAG) or custom prompt design.

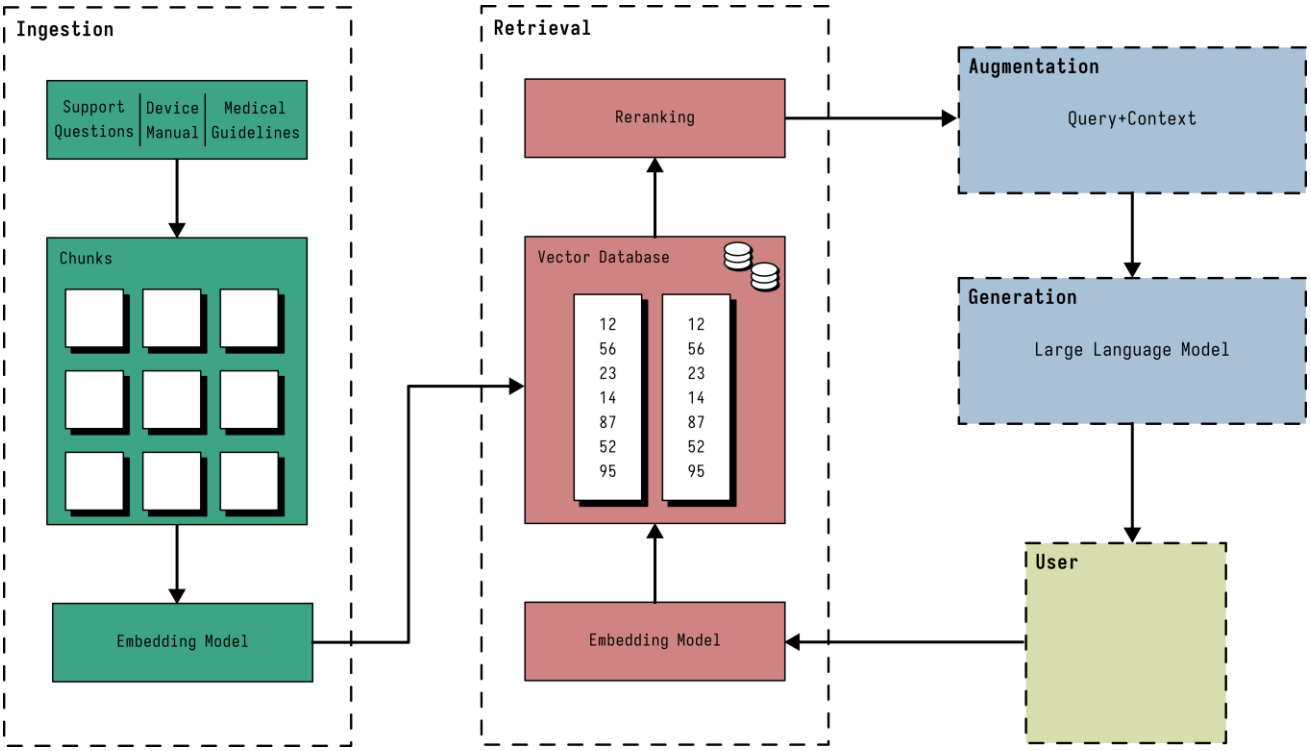



*Figure 16 - Overview of the AI assistant data pipeline leveraging Retrieval-Augmented Generation (RAG).*

## 5.5 AI Tools and Techniques

The LLM-powered assistant currently uses OpenAI GPT models via the OpenAI API, enabling access to models that are regularly updated and provide a broad knowledge base and strong reasoning capabilities. In the context of this use case, an online model provides a practical testbed for the TRUMAN solutions, as its outputs and evaluation results are relevant to many real-world deployments in which such services are increasingly integrated.

Given the fast-paced development of LLMs, the specific model and even the model provider may change over time, and open-source alternatives may be considered. Accordingly, the assistant is designed to remain flexible with respect to the underlying model. Given that the case study targets a future product, a robust solution is required; thus, a balance must be struck between a capable, fast, and resource-efficient approach and security and privacy requirements, in line with the applicable service-level agreement.

Beyond selecting an appropriate LLM backend, a second key aspect of the assistant architecture is how it connects to relevant knowledge sources. Currently, two options are being explored:

1. **Retrieval-augmented generation (RAG):** RAG is an architecture that combines information retrieval with generative LLMs to incorporate external sources into outputs. It supports more relevant, domain-specific responses by drawing on organizational data, scholarly literature, and specialized datasets without additional model training. This option is currently used in the system.
2. **Model Context Protocol (MCP):** MCP (Model Context Protocol) is an open-source standard for connecting AI applications to external systems. It enables access to data sources (e.g., local files, databases), tools (e.g., search engines, calculators), and workflows (e.g., specialized prompts) so the assistant can retrieve relevant information and execute tasks. This option will be evaluated in the future.

Given the rapid pace of development in this area, the architecture will remain adaptable so that alternative knowledge-integration mechanisms can be evaluated and adopted as needed without substantial redesign. An overview of the current data pipeline can be seen in Figure 16.

PDN, through the use case, provides AI-enabled products and services that will integrate and evaluate new TRUMAN techniques for trustworthy AI. The TRUMAN-specific AI tools and techniques supporting this objective will be integrated into the PDNeurobot as modular components. The selection of these modules will be determined after eliciting technical and AI-related requirements through collaboration with TRUMAN technical partners. The PDN system architecture will be studied and evaluated to identify which elements can be mapped to knowledge graphs, continual learning approaches, and large language models. This process falls within the scope of Deliverable D2.2, where its outcomes will be presented.

## 5.6 Requirements

Once operational, the PDNeurobot must be trustworthy to ensure it is lawful, ethical, and robust. The requirements apply across the AI system life cycle and to all stakeholders: developers who research, design, and/or build AI systems; deployers (public or private organizations) that use AI systems in business processes or to provide products and services; end-users who interact with the system directly or indirectly; and the broader society affected by AI systems. These requirements are grounded in the “Ethics Guidelines for Trustworthy AI” by the High-Level Expert Group on Artificial Intelligence and are presented in Table 6.

*Table 6 - Requirements for UC4*

| Reference | Requirement | Description |
|---|---|---|
| R4.1 | Human Oversight and Control | Module that enforces human-in-the-loop operation by requiring appropriate validation of high-impact outputs by clinicians and/or end users, preserving user autonomy, and preventing system outputs from being presented as clinical decisions or definitive determinations. |
| R4.2 | Robustness and Safety Mechanisms | Module that ensures system reliability through input validation, detection of prompt injection and adversarial inputs, and implementation of safe degradation strategies when inputs are incomplete, inconsistent, or out of scope. |
| R4.3 | Data Privacy and Access Control | Module that handles secure processing and storage of patient data, implements role-based access control (RBAC), enforces consent constraints, and ensures compliance with applicable data protection regulations (e.g., GDPR). |
| R4.4 | Transparency and Explainability | Module that provides clear disclosure of system capabilities and limitations, distinguishes between generated and retrieved content, and supports explainability features such as provenance tracking and confidence scoring. |
| R4.5 | Bias Detection and Mitigation | Module that monitors model performance across demographic and clinical subgroups, detects potential biases, and applies mitigation strategies to prevent systematic disadvantage or exclusion. |
| R4.6 | Auditability and Traceability | Module that maintains audit logs, model versioning, and decision traceability to ensure that system outputs are attributable and accountability requirements are enforceable. |

## 5.7 Trustworthiness Challenges

The PDN use case places an LLM-based assistant in a sensitive health context, interacting with patients, caregivers, and clinicians while drawing on external knowledge sources. These scenarios features give rise to the trustworthiness challenges below, which must be addressed for responsible deployment.

1. **Uphold Human Autonomy:** Ensure that the LLM assistant supports user agency and clinical oversight, avoids undue influence, and is not treated as a decision-maker, particularly for vulnerable users. This requirement arises from the LLM's role in summarizing and accessing clinical data that may inform clinical decisions, as well as providing disease management recommendations to patients. It is essential to ensure that users remain in full control of decisions, are consistently asked to provide explicit consent, and are provided with access to the sources underlying the LLM's outputs so they can maintain meaningful oversight.

2. **Prevent Harm:** risks such as hallucinated content, unsafe or misleading health advice, privacy leakage, and security exploits (e.g., prompt injection) through safeguards proportionate to potential harm. This challenge stems from the system's use of external knowledge sources, free-text inputs, and high-stakes health guidance. In the context of LLMs, this challenge mainly stems from:
    - **Adversarial attacks** (e.g., hallucinations, data poisoning, prompt injection): These can lead the system to generate incorrect, misleading, or manipulated outputs, especially when inputs are crafted to exploit model behavior or weaknesses. In healthcare settings, this increases the risk of unsafe or inappropriate guidance.
    - **Privacy leakage** (e.g., unintended exposure or extraction of personal data, including PII): The model may inadvertently reveal sensitive information from inputs or training data, particularly when handling free-text or user-provided content. This poses risks to patient confidentiality and data protection.
    - **Unauthorized access** (e.g., exploitation of system vulnerabilities, weak authentication, or improper permission controls): These can allow malicious actors to access restricted data, manipulate system behavior, or interfere with service integrity. In healthcare settings, this poses significant risks to patient safety, data security, and trust in the system.
3. **Ensure Fairness:** Prevent the assistant from favoring or disadvantaging any group by monitoring for biased outputs and unequal performance across demographics or use contexts. This challenge stems from heterogeneous user populations and variable data coverage across patient groups.
4. **Promote Explainability:** Make the assistant's capabilities and limits explicit, maintain traceability of retrieved sources and prompts, and provide explanations that support appropriate user expectations and accountability. This challenge stems from reliance on retrieval, dynamic prompts, and multi-source inputs in clinical workflows.

The trustworthiness challenges relevant to the PDN use case reflect the ethical principles for AI systems set out in the "Ethics Guidelines for Trustworthy AI" [2] by the High-Level Expert Group on Artificial Intelligence. These principles are rooted in fundamental rights and must be respected to ensure AI systems are developed, deployed, and used in a trustworthy manner.

Overall, the use case will investigate the robustness of the system, particularly with respect to fairness, reliability, security, and privacy risks associated with the use of LLM-based assistants. It will examine whether the assistant provides consistent and unbiased responses across different users and contexts, maintains accuracy when integrating multimodal clinical data, and safeguards sensitive health information throughout interactions. In addition, the study will consider potential vulnerabilities, such as inappropriate data exposure, over-reliance on automated outputs, or unintended biases in generated recommendations, to systematically identify potential risks gain a deeper understanding of their implications and inform the design and implementation of appropriate mitigation strategies when deploying LLM-based assistants within a clinical ecosystem.

# 6 Conclusion

This deliverable has defined the four TRUMAN use cases that will guide the design, implementation, and validation of the project's trustworthy AI solutions across social science research (UC1), cybersecurity (UC2), financial fraud detection (UC3), and digital health (UC4). By grounding the project in concrete and high-impact scenarios, the document ensures that technical developments remain aligned with real-world operational and societal needs.

The use cases collectively span different phases of the AI lifecycle - from data collection and preprocessing to distributed training, deployment, and human interaction — and involve heterogeneous data types, architectures, and stakeholder ecosystems. This diversity provides a robust validation environment for TRUMAN's methods, enabling the systematic assessment of robustness, privacy, fairness, and explainability across domains.

Importantly, the deliverable translates trustworthiness principles into concrete requirements and challenges within each context, creating a clear link between real-world constraints and technical research objectives. By adopting a common structure across all use cases, it ensures comparability, traceability, and coherence for subsequent work packages.

# 7 Bibliography

[1] S. Shiffman, A. A. Stone and M. R. Hufford, "Ecological momentary assessment," *Annu. Rev. Clin. Psychol.,* vol. 4, no. 1, pp. 1-32, 2008.

[2] "Ethics guidelines for trustworthy AI," 08 04 2019. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai.

[3] F. Keusch, B. Struminskaya, C. Antoun, M. P. Couper and F. Kreuter, "Willingness to participate in passive mobile data collection," *Public Opinion Quarterly,* vol. 83, no. 1, pp. 210-235, 2019.

[4] S. Aminikhanghahi, M. Schmitter-Edgecombe and D. J. Cook, "Context-aware delivery of ecological momentary assessment," *IEEE journal of biomedical and health informatics,* vol. 24, no. 4, p. 1206–1214, 2019.