i·PROGNOSIS

PROJECT
i-PROGNOSIS: Intelligent Parkinson early detection guiding novel supportive interventions

GRANT AGREEMENT No.
690494

D3.1 - Data acquisition and protection version G

CONTRACTUAL SUBMISSION DATE
January 2017

ACTUAL SUBMISSION DATE
January 2017

DELIVERABLE VERSION
4.0 (Final)

MAIN AUTHOR(S)
Fotis Karayiannis (MICROSOFT)
Dimitris Papadimitriou (MICROSOFT)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 690494.

GRANT AGREEMENT No.	690494
PROJECT ACRONYM	i-PROGNOSIS
PROJECT FULL TITLE	Intelligent Parkinson early detection guiding novel supportive interventions
Type Of Action	Research & Innovation Action (RIA)
Topic	H2020-PHC-21-2015 - Advancing active and healthy ageing with ICT: Early risk detection and intervention
Start Of Project	1 February 2016
Duration	48 months
Project URL	www.i-prognosis.eu
EU Project Officer	Ramón Sanmartín Sola

DELIVERABLE TITLE	Data acquisition and protection version G
DELIVERABLE No.	D3.1
Deliverable Version	4.0
Deliverable Filename	i-PROGNOSIS-690494_D3.1.docx
Nature Of Deliverable	R (Report)
Dissemination Level	PU (Public)
Number Of Pages	21
Work Package	WP3 - Behavioural Info Capturing and Machine Learning for Early PD Symptoms Detection
Partner Responsible	MICROSOFT
Author(s)	Fotis Karayiannis (MICROSOFT), Dimitris Papadimitriou (MICROSOFT), Vasileios Charisis (AUTH), Stelios Hadjidimitriou (AUTH), Dimitris Iakovakis (AUTH), Konstantinos Kyritsis (AUTH), Nikos Grammalidis (CERTH), Michael Stadtschnitzer (FRAUNHOFER)
Editor	Fotis Karayiannis (MICROSOFT)

ABSTRACT

This deliverable presents the data acquisition and safety procedures regarding the first phase of data collection, namely GData. The deliverable is composed in two main parts, the data acquisition (capturing) and the security-protection mechanisms. The deliverable corresponds to the work of Task 3.1 and is strongly related to Tasks T2.2 and T1.4.

KEYWORDS

Anonymisation; Cloud storage; Data acquisition/ protection; GData; Mobile application; Security

SIGNATURES

WRITTEN BY	RESPONSIBILITY - COMPANY	DATE
Fotis Karayiannis	Main Author 1 - MICROSOFT	1/22/2017
Dimitris Papadimitriou	Main author 2 - MICROSOFT	1/22/2017
REVIEWED BY		
Hugo Plácido da Silva	Internal Reviewer 1 - PLUX	1/29/2017
Anastasios Delopoulos	Internal Reviewer 2 - AUTH	1/26/2017
APPROVED BY		
Leontios Hadjileontiadis	Project Coordinator - AUTH	1/31/2017

TABLE OF CONTENTS

LIST OF MAIN ABBREVIATIONS

DoA	Description of Action
EC	European Commission
EU	European Union
GData	Generalised data
GDPR	General Data Protection Regulation
JSON	JavaScript Object Notation
ICT	Information and Communication Technologies
IMU	Inertial measurement unit
PD	Parkinson's disease
SData	Specialised data
UI	User interface

1 EXECUTIVE SUMMARY

i-PROGNOSIS involves a two-stage process in order to assess the early detection of Parkinson's Disease (PD). In the first stage, data is collected via the mobile application that a user has installed on his/her smartphone, namely the GData collection phase. The data collected will constitute the basis for the decision on whether or not the user should proceed to the second stage of PD detection. During the first stage, personal data are thus collected, stored and transmitted and the data protection/safety mechanisms are of key importance.

In this context, the scope of this deliverable is to present on one hand the data acquisition (capturing) approach and on the other hand the data protection/safety mechanisms at this first stage of data collection (GData). The two points highlighted above correspond to the work of Task 3.1 (WP3), entitled "Data Acquisition and Protection". The deliverable explains the data acquisition, transmission and storage means at the GData phase, and the mechanisms that guarantee the protection and safety of data at all stages and in all devices, namely on local (mobile phone) or central level (Cloud). This includes anonymisation, secure transmission and storage of data, i.e., the user ID and the actual data, at separate places via secure methods, so that collected data cannot be traced back to a single individual.

Section 3 describes the overall architecture of the GData mobile application along with the coordination of data acquisition, and then dives into the details of the acquisition for each capturing component of the GData stage. For each component, a table has been inserted including the overall description and its implementation, along with their inputs and outputs, the corresponding data exchange formats for each of them and a JavaScript Object Notation (JSON) payload example.

Section 4 presents the mechanisms that will guarantee data protection and safety, at the different phases, namely at the capturing phase, at the local storage phase (on the mobile phone), at the transmission phase (from the phone to the Cloud or other servers) and the central storage phase (in the Cloud or other server). This includes the electronic participant information sheet and the informed consent form in the beginning, along with the anonymisation of the data, by removing all personal information and providing only a user identifier which is stored in a separate location from the captured data. Data protection on the mobile phone and the local SQLite database, along with secure transmission and storage on the Cloud, which in the i-PROGNOSIS case will be the Microsoft Azure solution, are also provided, following required regulations and with appropriate security mechanisms and protocols in place, as described in detail in the corresponding sections.

It is clear that across all phases, the appropriate safeguards have been implemented to guarantee data anonymisation, protection and safety. In conclusion, an appropriate and well-orchestrated plan has been developed for both data acquisition and protection at all phases of the first stage of data collection.

2 INTRODUCTION

2.1 RELEVANT PROJECT ASPECT

The main objective of i-PROGNOSIS is the development of an ICT-based behavioural analysis approach for capturing, as early as possible, the PD symptoms appearance, and the application of ICT-based interventions countering identified risks. To achieve its objectives, the project needs to reach out to more than 5000 elder individuals within the duration of the project, in order to unobtrusively sense large scale behavioural data from them, acquired from their natural use of mobile devices (e.g., smartphone). To this end, as data are of personal nature, the data protection/safety mechanisms are of key importance.

The scope of this deliverable is thus to present on one hand the data acquisition (capturing) approach and on the other hand the data protection/safety mechanisms at the first stage of data collection, namely the GData collection phase. The two aforementioned points correspond to the first part of the work of WP3 Task T3.1, entitled "Data Acquisition and Protection".

Besides this task, there are also other tasks which are very relevant, namely Task 2.2 - *Design of the data collection protocol* and Task 1.4 - *Ethical and Safety Management*, and their corresponding deliverables D2.3 - *First version of system specification* and D1.2 - *First version of ethics and safety manual*.

The ultimate goal for this deliverable is to explain the data acquisition, transmission and storage mechanisms at the GData phase, and guarantee the protection and safety of data at all phases (within the first stage) and across devices, namely on local (mobile phone) or central level (Cloud).

2.2 PURPOSE AND STRUCTURE OF DOCUMENT

The target audiences of this deliverable are both internal and external to the project. Internal, as it is necessary to define a plan and coordinate with other tasks and deliverables, and external, referring mainly to the people interested in understanding the data protection and safety measures at all phases of production, transmission and storage at the GData phase. Such people may include actual users of the i-PROGNOSIS application or people connected to them, as well as third parties who may want or need to verify compliance with data protection and ethical regulations. Other external people or the general public may also have similar interest.

The present deliverable is structured as follows:

- Section 1 is the Executive Summary.
- Section 2 provides an introduction with the overall scope of the document, its target audience and structure.
- Section 3 presents the data acquisition approach and a break down of the data capturing modules.
- Section 4 reports on the data protection/safety mechanisms.
- Section 5 summarizes the main conclusions from the deliverable.

3 DATA ACQUISITION

This section provides two main parts, a first part with the overall architecture and coordination of data acquisition and then a second part which dives into the details of the acquisition for each capturing component of the GData stage.

3.1 THE GDATA MOBILE APPLICATION FRAMEWORK

The main vehicle for data acquisition at the first stage of data collection (GData) is the mobile phone and the corresponding i-PROGNOSIS mobile application backend (**FIGURE 1**). The mobile application, once installed, requires the consent from the user to authorise the capturing, storage and processing of the data. The user is *a priori* informed about the goal of the study and how the data will be used. In addition, there is also information on how the user can opt-out and the option of deleting her/his so far collected or not, after withdrawal. The application runs several services in order to capture appropriate data needed for early detection of PD research set within the context of the i-PROGNOSIS project. These services retrieve data from different capturing sensors of and data available on the mobile phone, such as the microphone (for voice quality evaluation), the inertial measurement units (IMU) (for capturing movement and understanding phone handling), a custom keyboard (for capturing keystroke dynamics), location sensors (for capturing the location patterns of the user) and stored photos and messages (for facial and emotional content extraction).

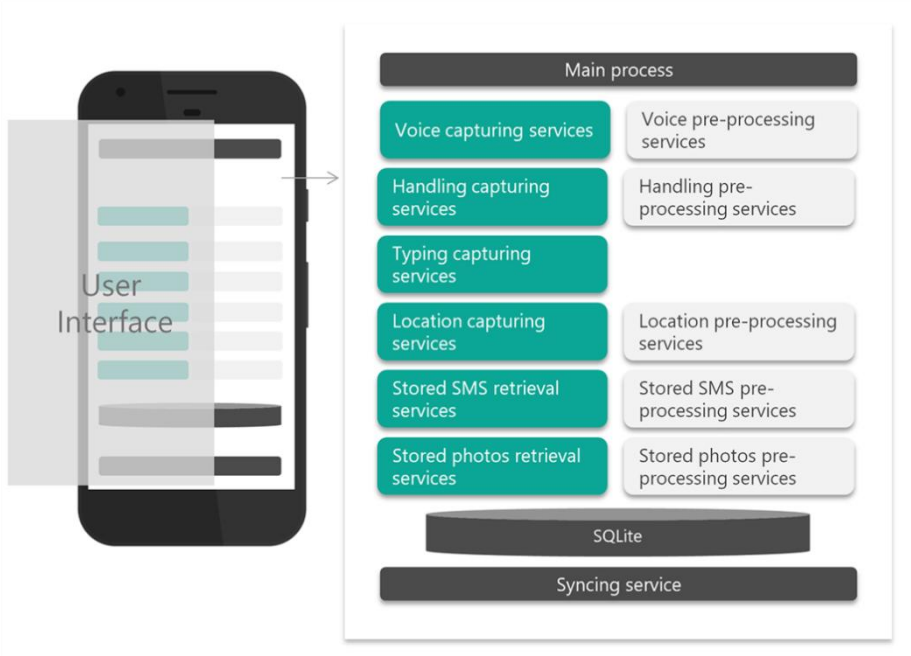

FIGURE 1 The GData mobile application break-down.

All capturing services run as independent background services and each one automatically collects and writes data in separate tables of a local database stored on the mobile phone (specifically an SQLite database, located in the application dedicated storage). The data sources are also pre-processed inside the mobile phone as needed, also taking into account privacy concerns. As an example of pre-

processing, voice (audio) files are pre-processed in such a way that features extracted and transmitted to the Cloud cannot affect the user's privacy and cannot be traced back to her/him. Cloud refers to the storage / databases, data processing and exchange that will take place via third-party data centres. In i-PROGNOSIS, the Microsoft Azure Cloud platform will be used. The next step is that the data is sent to the Cloud for further processing at specific synchronization cycles (when the phone is not being used by the user, is charging and is connected to Wi-Fi). Further processing (main processing) takes place on the Cloud after collection and storage of depersonalised data by the smartphone application.

The Cloud can then also push notifications which the phone is able to receive and properly handle/interpret, in case the user has agreed during the consent process to be contacted by medical partners of i-PROGNOSIS for a study follow up. The details of data anonymisation and de-identification of the actual capturing data, as well as the transmission and storage safeguards on the Cloud are given in Section 4.

Main Process	
Architecture Component:	**`M01 Mobile application`**
Description / Implementation	
The main process of the application (namely `MainActivity`), is responsible for the marshalling of the capturing services. After the user provides informed consent and the application launches for the first time, all the capturing services are enabled and run as independent background services. The user can deactivate/activate the capturing services according to his will.	

SQLite Database	
Architecture Component:	**`M04 Mobile application data storage`**
Description / Implementation	
All capturing services run as independent background services and each of them automatically collects and writes data in separate tables of a local database stored in the mobile phone (specifically an SQLite database, stored in the application's dedicated storage). The creation and initialisation of the database (which is of type .db3[1]) -and for each of the tables that it includes, i.e. one table per capturing service- takes place the first time the application launches. Specifically, this occurs inside the `OnCreate()`[2] method that resides in the `Main Activity` of the application. Each input/output to the database is possible only via a connection to it (the database). Concerning the connection, a synchronous approach is followed. More specifically, a connection is	

[1] DB3 file http://www.openthefile.net/extension/db3

[2] Android activity lifecycle
https://developer.android.com/guide/components/activities/activity-lifecycle.html

provided via a specialised class, which acts as the sole connection provider. The class is marked as sealed so that it cannot be inherited for security purposes. The Singleton pattern[3] is being utilised along with a thread safe lock, thus ensuring that the connection is unique and not more than one connection can exist at any given time. When two (or more) capturing services attempt to write to the database, the architecture of the connection followed (which is described above) prevents them to do it simultaneously. Each service has to finish the writing to the database, before the connection can be used by another service. In order to achieve the maximum possible stability of the database, multiple tables are used (one table per service).

Input / Output			
Input:	Captured raw data as objects following this model structure: > ID (int) >CapturingServiceName (string) > Payload (string)	Output:	Raw captured data exported as objects with the aforementioned fields

It must be noted that the present deliverable focuses on the raw data capturing components. Thus, the pre-processing components and the syncing service (**FIGURE 1**) will be presented, once finalised, along the main processing components in the upcoming deliverable D3.3 - *First version of GData analysis modules* (Month 22).

3.2 DATA CAPTURING SERVICES

This section dives into the details of the acquisition for each capturing component service for the GData stage as initially described in D2.3 - *First version of system specification*. The background services that each capturing component comprises, as well as specifications of the input and output of each component, are presented.

Voice capturing services	
Architecture Component:	**`C01 Voice capturing`**
Description / Implementation	
The C01 Voice capturing component is responsible for the capturing of the microphone channel during incoming and outgoing calls. The component is implemented as a background service that is started by a broadcast receiver[4] listening when the boot of the operating system is completed. Part of the voice capturing background service is the voice capturing broadcast receiver that is listening to the messages of the operating system if a new outgoing call is performed or an incoming call is answered. The operation of the voice capturing broadcast receiver is depicted in **FIGURE 2**. If an incoming call is answered or a new outgoing call is performed the media recorder starts the recording of (only!)	

[3] Singleton pattern http://www.oodesign.com/singleton-pattern.html
[4] Android broadcasts https://developer.android.com/guide/components/broadcasts.html

the microphone channel in format: .3gpp, AMR-WB codec, 16 kHz sampling rate. Also a timer is started to stop the recording after *n* seconds (*n*=75). The recording is either stopped if one of the conversational partners ends the call or by the timer after *n* seconds; whatever comes first. The timer is implemented to limit the length of the recordings and the used memory space on the smartphone. Finally, each successful recording is registered in the tables of the SQL database by entering the temporary filename as a JSON string. The recorded audio files are removed from the smartphone after the audio features are extracted from the voice pre-processing component P01.

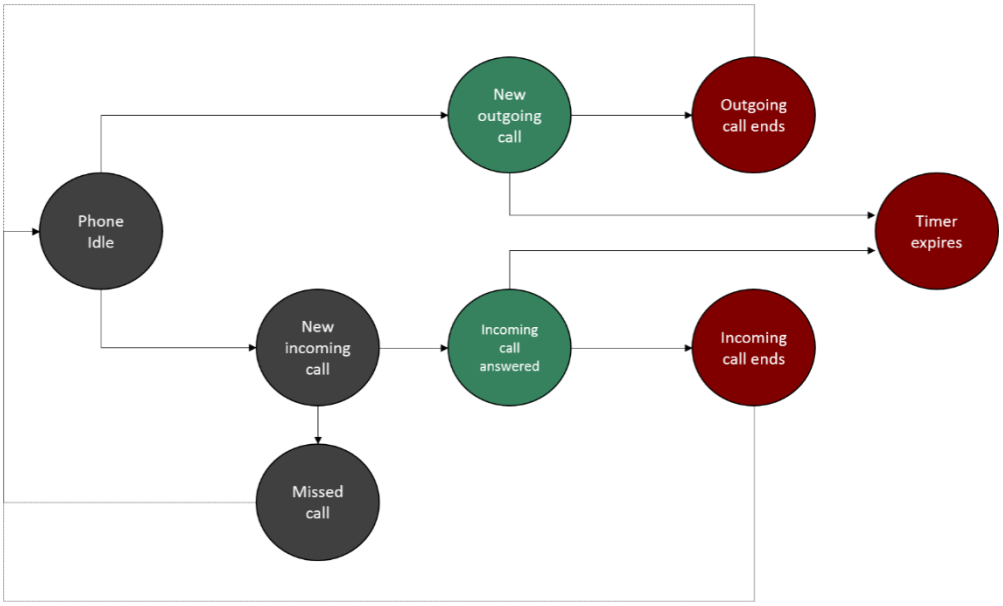

FIGURE 2 Operation of the voice capturing broadcast receiver state machine architecture. The gray coloured states represent idle or non-emitting states (i.e. states that do not trigger any additional events), green states indicate start emitting states (i.e. states that start the capturing procedure) and red states indicate stop emitting states (i.e. states that stop the capturing procedure).

Input / Output	
Input:	Incoming and outgoing phone calls, microphone
Output:	Raw captured audio data, Entry in SQL database
Data exchange:	Input: System broadcasts, microphone channel Output: 3gpp audio files (microphone channel recordings), Entry in SQL database

JSON payload example (*if applicable*):	{
	"AudioFileName":"tmp/20170119-094233.3gpp"
	}

Handling capturing services	
Architecture Component:	**C02 Handling capturing**

Description / Implementation
The aim of this component is to unobtrusively capture user generated IMU data, during the user's daily interaction with the smartphone. More specifically, the term unobtrusive refers to the capturing process being triggered by specific events such as answering or initiating a voice call, or engaging in a typing session, without requiring the user to manually start or stop the capturing procedure. Collectively, the captured IMU data include: 1) Triaxial acceleration values `[x, y, z]`, measured in `m/s^2`, along with the respective timestamp in ns for each acceleration triad; 2) Triaxial gyroscope values `[x, y, z]`, measured in `rad/s` along with the respective timestamp in ns for each gyroscope triad; 3) Triaxial magnetometer values [x, y, z], measured in `uT` (micro-Tesla) along with the respective timestamp in ns for each magnetometer triad; 4) The origin of the generated data, i.e. the name of the event that triggered the capturing procedure; e.g. "`Keyboard`" or "`Call`". Furthermore, due to the fact that smartphones belonging in different price ranges may or may not contain specific sensors such as the magnetometer, the component is able to automatically detect the available sensors in the user's device and use those in the capturing procedure. The entire logic of the specific component in encapsulated in a single background service, namely "IMU Trigger Service". The aforementioned service is initially started upon the first execution of the i-PROGNOSIS application, and then scheduled to restart after during the device's start up (i.e. after a potential reboot). Furthermore, the service is also able to automatically start itself, in case the Android Operating System is forced to stop long-running services in order to free resources. In order for the "IMU Trigger Service" to be able to start and stop the capturing process, the aforementioned service maintains two different event broadcast receivers, specifically designed to monitor the device's status. More specifically, the first broadcast receiver, namely "Call Receiver" is responsible for monitoring events related to voice calls. "Call Receiver" operates in a state-machine fashion meaning that is able to continuously track changes during a voice call. Starting at an idle state, the complete operation of the "Call Receiver" can be seen in **FIGURE 3**. The second broadcast receiver, namely "Keyboard Receiver" is responsible for monitoring any typing related activities initiated by the user. The operation of the "Keyboard Receiver", as depicted in **FIGURE 3**, follows the same state-machine principle of the "Call Receiver" but in a more linear sense, since the software keyboard can only be visible or hidden. As it can be seen in **FIGURE 3**, when either the call or keyboard state machine reaches a green coloured state, it registers a handler (component that allows listening to the sensors in a non-blocking fashion) for all available IMU sensors and begins caching IMU generated data in the device's memory. Consecutively, when either state machine reaches a red-coloured state, it detaches the handler responsible for listening to the sensors and as a result the capturing procedure is stopped. The cached data are then serialized into a JSON format and an identifier relevant to the user's performed action that initiated the collection procedure is also attached. The

JSON formatted data are then forwarded for storage in the raw IMU data table of the SQLite database. The efficiency of storing IMU data in a JSON format will be extensively tested during the Pilot Data Collection Period. In case the JSON approach is found suboptimal, the alternative of storing the IMU payload as a binary file, and attaching the path of the file in the JSON payload field will be adopted (similar to C01. voice capturing service).

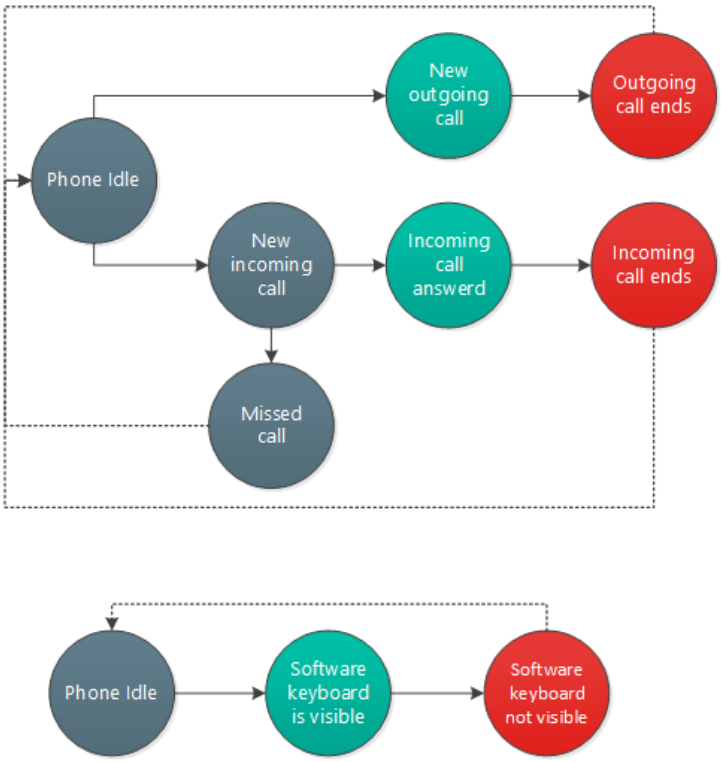

FIGURE 3 Operation of the call broadcast receiver (top) and keyboard receiver (bottom) state machine architectures Gray coloured states represent idle or non-emitting states (i.e. states that do not trigger any additional events), green states indicate start emitting states (i.e. states that start the capturing procedure) and red states indicate stop emitting states (i.e. states that stop the capturing procedure).

Input / Output	
Input	Android Operating System broadcasted events, broadcasted events from the typing capturing component
Output:	JSON string including: 1) three floating point arrays for each one of the accelerometer axes, 2) one long array for the accelerometer timestamps, 3) three floating point arrays for each of the gyroscope axes, 4) one long array for the gyroscope timestamps, 5) three floating point arrays for each one of the magnetometer axes, 6) one long array for the magnetometer timestamps and 7) a string indicating the user action that triggered the collection process
Data exchange:	Input: Sensors broadcasting Output: Entry in the SQLite database
JSON payload	{

example	"Origin":"Call", "AccelX":[0.97503662109375, 0.9464111328125, 0.9312896728515625, 0.937713623046875], "AccelY":[3.13818359375,, 3.08258056640625, 3.0545196533203125, 3.047149658203125], "AccelZ":[9.128753662109375, 8.97393798828125, 8.8331298828125, 8.6864013671875], "AccelT":[6655670915167, 6655675950567, 6655680985968, 6655686021368], "GyroX":[0.153717041015625, 0.17041015625, 0.1781768798828125, 0.174041748046875], "GyroY":[-0.1217193603515625, -0.13836669921875, -0.1451416015625, -0.14093017578125], "GyroZ":[-0.089935302734375, -0.091888427734375, -0.0911407470703125, -0.089202880859375], "GyroT":[6655952897589, 6655957932989, 6655962968389, 6655968003790], "MagnetX":[69.16351318359375, 69.41680908203125, 69.41680908203125, 69.63348388671875], "MagnetY":[-163.48419189453125, -163.63677978515625, -163.38653564453125, -163.604736328125], "MagnetZ":[-55.413818359375, -55.4718017578125, -53.77655029296875, -54.0924072265625], "MagnetT":[6656053605596, 6656073747198, 6656093888800, 6656114030401] }

Typing capturing services	
Architecture Component:	**C03 Typing capturing**
Description / Implementation	

The mission of the component is to capture keystrokes-related data during typing sessions of the user on a touch screen keyboard (see D2.3). The data include: 1) touch events (action down: key is pressed, action up: key is released) of keystrokes, 2) the pressure applied for each keystroke, 3) if a keystroke corresponds to a deliberate long press, 4) the number of "Delete" keypresses and 5) typing session metadata (time window of the session and whether or not sound/vibration key feedback was active).

The component comprises two services (**FIGURE 4**). The main service is a Software Input Method service, i.e., a software keyboard. As the Android Operating System does not allow the recording of the aforementioned data on existing keyboards, a new fully-functional custom keyboard (see also D5.2) was developed that captures the required data in the background, unobtrusively. As soon as a typing session ends, the keyboard service passes a JSON string including the payload, i.e., the captured data and metadata, via an intent, to a second service, i.e., an Intent service, which proceeds with the logging of the data entry in the application SQLite database. A typing session is defined from the moment the keyboard is shown/drawn to the moment it is hidden or redrawn. For typing sessions where no keystrokes occur, the

component does not log data in the database. The two-service approach was adopted in order to disburden the main keyboard service of the database logging, ensuring in this way its proper and responsive operation. It must be noted that the user must activate and use the i-PROGNOSIS keyboard as the default input method for data to be captured.

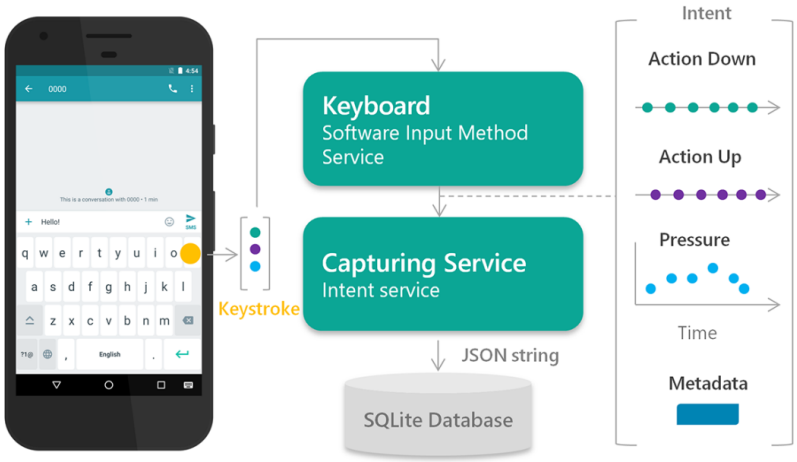

FIGURE 4 The process of capturing and storing keystroke-related data during a typing session by using the i-PROGNOSIS keyboard

Input / Output	
Input:	System: 1) Touch events, 2) Pressure of touch events, 3) Long press events, 4) System time
Output:	JSON string including: 1) Long array of touch events timestamps (Down time, Up time) [ms], 2) Float array of touch events normalised pressure values [0-1], 3) Integer array of "is long press" flags [0 is not long press - 1 is long press], 4) Integer number of "Delete" keypresses, 5) Boolean flag for key sound feedback [false sound off - true sound on], 6) Boolean flag for key vibration feedback [false vibration off - true vibration on], 7) Time window of each recording [date-time]
Data exchange:	Input: System input Output: Log entry in SQLite database
JSON payload example	{ "TimeUnit":"ms", "PressureUnit:"normalised", "DownTime":[101992620,101993841,101994497,101998248], "UpTime":[101992702,101993956,101994613,101999138], "PressureValue":[0.4,0.587500036,0.675,0.6375], "IsLongPress":[0,0,0,1], "NumDels":1, "IsSoundOn":false, "IsVibrationOn":true, "StartDateTime":"2017-01-05T16:37:11.203011+2:00",

	"StopDateTime":"2017-01-05T16:37:17.721027+2:00" }

Location capturing services	
Architecture Component:	**C04 Location capturing**
Description / Implementation	
This component aims to unobtrusively capture the user's location throughout the day, by tracking the smartphone global position via GPS, GSM and Wi-Fi. More specifically, the data collected by the aforementioned component includes 1) a longitude, latitude, altitude tuple and 2) the respective timestamp for the given tuple. The entire logic of the location capturing component is encapsulated in a single background service, namely "Location Trigger Service". Similar to the "IMU Trigger Service", the location capturing component initially starts upon the first execution of the i-PROGNOSIS detection application, and is scheduled to restart each time the user's device reboots. Furthermore, the location trigger service is able to automatically restart itself, in case the Android OS decides to terminate long-running processes in order to free resources. More specifically, the location trigger service registers a handler (module able to manipulate incoming data in a non-blocking fashion) in the Google Play Services location listener, and then sends periodic location requests during a given time window. The collected location data of the aforementioned window are temporarily cached in the device's memory. In order for the temporarily cached data to be stored in the SQLite database, the location trigger service registers a broadcast receiver to the time tick event that is raised every minute by the Android OS. When the number of elapsed minutes passed is equal or exceeds the given time window, location data collection is halted, the cached location data are serialized in a JSON format and forwarded for storage in the location-specific table of the SQLite database. After the new entry is successfully created, the location trigger service reset the embedded tick-timer and registers a handler to the Google Play Services location listener, effectively repeating the same process as explained above. The reason behind using Google Play Services instead of requesting the location directly from the device's global positioning system (GPS) is heavily correlated with the "accuracy versus battery life" trade-off, meaning that Google Play Services will provide high-accuracy location data in a manner that is non-suppressive for battery life (i.e. unobtrusive). The previous notion is widely accepted by the Android development community, and as a result of using Google Play Services as a location provider, this is the proposed method for adding location awareness in an application	
Input / Output	
Input:	Android OS broadcasted events, fused location from GPS, GSM, Wi-Fi
Output:	JSON string including: 1) one double array containing the longitude in degrees, 2) one double array containing the latitude

	in degrees, 3) one double array containing the altitude in meters above the WGS 84 reference ellipsoid, and finally 4) one long array of the timestamp of each [longitude, latitude, altitude] triad, in ms.
Data exchange:	Input: None Output: Entry in SQLite database

JSON payload example	{
	"Longitude":[40.6400630, 40.6400625, 40.6400629, 40.6400629],
	"Latitude":[22.9444193,, 22.9444182, 22.9444200, 22.9444191],
	"Altitude":[27.093, 27.080, 27.073, 27.091],
	"Timestamp":[6655670915167, 6655675950567, 6655680985968, 6655686021368],
	}

Stored SMS retrieval services	
Architecture Component:	**C05 Stored SMS retrieval**
Description / Implementation	
This component has one service responsible for parsing the stored SMS messages on the user's smartphone on a predefined schedule (e.g., during the last 24 hours). The body of all text messages that have been sent by the user during the period of interest is registered in the SMSRawData table of the SQLite database. This service runs with a period equal or larger than 24 hours, so that no SMS message can be captured twice.	
Input / Output	
Input:	SMS Messages sent by the user during a specific interval (period of interest)
Output:	Entries (body text of each SMS message) in SQL database
Data exchange:	Input: Read data from local storage Output: Write SMS body as an entry in the SQLite database (Table "SMSRawData") on a temporary storage directory

JSON payload example	{
	"SMSbody":"This is a sample SMS message sent"
	}

Stored photos retrieval services	
Architecture Component:	**C06 Stored photos retrieval**

Description / Implementation	
This component is responsible for obtaining the path of the photo, captured each time the front camera is used, and store it as an entry to the local SQLite database.	
Input / Output	
Input:	Path (in local storage) of photo taken by the front camera
Output:	Entry (path of the photo) in SQL database
Data exchange:	Input: Photo captured each time the frontal camera is used Output: Path to the photo taken, as an entry in an SQLite database "RawData" (Table "PhotoRawData") on a temporary storage directory

JSON payload example:	{
	"PhotoPath":"/sdcard/Photoxxxx.jpg"
	}

4 DATA PROTECTION

Data collection via the i-PROGNOSIS mobile application involves personal data from human beings. This is why special attention is given on the compliance of the data collection processes with ethical regulations and guidelines on research involving human beings, as well as on the safety of the sensor devices involved and the protection of personal and health data. Deliverable D1.2 entitled "First version of ethics and safety manual" provided the general reference guide for the i-PROGNOSIS investigators by reporting on the international, European and National ethical regulations, device safety standards and certifications, as well as accepted data management procedures.

In this deliverable, we go one step further, examining in more detail, the different phases of data acquisition, local storage in the phone, transmission to secure servers and the Cloud and storage in the Cloud, along with the appropriate anonymisation safeguards and separation of the user identification handle (user ID) and the actual data.

In more detail, the following phases are analysed below:

- Download and installation of the i-PROGNOSIS mobile application, which requires user consent (e-consent) on data collection and processing
- Data anonymisation, including separation of the user ID from the actual collected data
- Local data protection (in the local database of the mobile phone)
- Secure data transmission from the mobile phone to the Cloud or other secure servers.
- Data protection in the Cloud (Microsoft Azure)

4.1 DATA ANONYMISATION

Data protection compliance begins after the user downloads the i-PROGNOSIS mobile application. As soon as the application is launched for the first time, the user is informed about the data collection study and s/he is asked for an electronic consent, allowing the appropriate handling and processing of the data. The e-Consent form (in the form of typing a name or a signature that will be drawn on the smartphone) has been agreed by the medical partners of the project, and further feedback will be received after submitting the related approval application to the national ethical committees of the three countries, i.e., the UK, Germany and Greece. Each participant providing informed consent will be assigned a coded identification number (pseudo-anonymisation). The collected data will only have this coded ID number and will not contain any other information that may identify the participant. The storage location of the ID and the actual data will be separate. Only the principal investigator of the medical partner responsible will have access to the corresponding consent form stored in the separate database, while the developers and all other technical stuff will have access to the anonymised data. In addition, an administrator will have access to the IDs database, but he can access it only after approval from the principal investigator. Although the current landscape of data protection is not fully clear with the new General Data Protection Regulation (GDPR)[5] and leave some space for interpretation, in particular regarding the exceptions for research and health data[6] (which is exactly relevant for i-PROGNOSIS), the aforementioned constitute a solid first step. If needed, and as new codes of conduct become available for such purposes[2], the document will be updated accordingly.

With the adopted approach, the identity of each participant is kept confidential and is not shared amongst consortium partners or any other third party.

4.2 LOCAL DATA PROTECTION

The capturing devices and services first store their data in a local database in the mobile phone (specifically an SQLite database, stored inside the application's dedicated storage and thus utilized only by the application).

Two separate databases are going to be used, in order to avoid conflict and ensure the privacy of the data. The latest version of SQLite database is going to be used (both databases will be of type .db3). The raw captured data are stored in a database, in separate tables according to the capturing service.

In order to anonymize and prepare the data (stored in the database containing the raw captured data) for transmission to the Cloud, the data are pre-processed. This may include appropriate encoding for sensitive data, such as the voice, so that the conversational context is removed and the data cannot be traced back to the user.

[5] http://www.eugdpr.org – GDPR will come into effect in the spring of 2018, replacing the Data Protection Directive 95/46/EC and imposing new obligations on organisations processing personal data of European Union residents

[6] "We must urgently clarify data-sharing rules" http://www.nature.com/news/we-must-urgently-clarify-data-sharing-rules-1.21350#auth-1

After the pre-processing takes place, the (pre-processed) data, now stripped of any potential personal information, are stored in a separate database, following the same principle concerning the tables (one table per service).

The data are then uploaded in the Cloud via the syncing service which runs daily, when the user does not use his/her phone for a long period of time (e.g. when he goes to sleep), and only after the user plugs the phone in the power and has an established Wi-Fi connection. The data stored inside the mobile phone will only have the user ID and the different capturing services will be using separate tables in the local database.

4.3 DATA TRANSMISSION & CLOUD STORAGE

As identified above, the captured data will be sent to the Cloud for further processing. In i-PROGNOSIS, the Microsoft Azure Cloud platform will be used, which is considered one of the most compliant to EU and national regulations (see also D1.2 - *First version of ethics and safety manual*). i-PROGNOSIS will be using Cloud resources residing inside the EU (Ireland and Netherlands). Geo-redundancy in two EU locations and disaster recovery is inherently provided by Azure.

Azure Storage provides all necessary security capabilities. Data can be secured in transit between the mobile application and Azure by using Client-Side Encryption[7], HTTPS, or SMB 3.0. The Azure Storage Client Library for .NET Nuget package supports encrypting data within client applications before uploading to Azure Storage, and decrypting data while downloading to the client. The library also supports integration with Azure Key Vault for storage account key management. Data can be set to be automatically encrypted when written to Azure Storage using Storage Service Encryption (SSE)[8]. Azure "SSE for data at rest" thus contributes in protecting and safeguarding data, so as to meet organisational security and compliance commitments. With this feature, Azure Storage automatically encrypts the data prior to persisting to storage and decrypts prior to retrieval. The encryption, decryption, and key management are totally transparent to users. Operating systems and data disks used by virtual machines can be set to be encrypted using Azure Disk Encryption[9]. The storage account itself can be secured using Role-Based Access Control and Azure Active Directory. Delegated access to the data objects in Azure Storage can be granted using Shared Access Signatures[10]. Thus, all necessary technologies are available to guarantee secure transmission and storage. More information regarding Azure compliance has been included in D1.2.

[7] Client-Side Encryption: https://docs.microsoft.com/en-us/azure/storage/storage-client-side-encryption

[8] Storage Service Encryption (SSE): https://docs.microsoft.com/en-us/azure/storage/storage-service-encryption

[9] Azure Disk Encryption: https://docs.microsoft.com/en-us/azure/security/azure-security-disk-encryption

[10] Shared Access Signatures: https://docs.microsoft.com/en-us/azure/storage/storage-dotnet-shared-access-signature-part-1

5 CONCLUDING REMARKS

This deliverable has described the overall architecture of data acquisition along with the details of the acquisition for each capturing component of the GData collection stage. For each component, all necessary details are described regarding the implementation of the different components, with appropriate text descriptions, tables and figures, as agreed within the Consortium and as already being implemented.

The mechanisms that guarantee data protection and safety at the capturing, local storage (on the mobile phone), transmission (from the phone to the Cloud or other servers) and the central storage phases (in the Cloud or other server) are also outlined. This encompasses the informed obtaining of electronic consent from users, the anonymisation of the data removing all personal information and providing only a user identifier, and the separation of storage between the ID and the actual captured data. Data protection inside the mobile phone and its SQLite database, along with secure transmission and storage in the Cloud, which in our case will be the Microsoft Azure solution, are also provided, following required regulations and with appropriate security mechanisms and protocols, as described in more detail in the corresponding sections.

It is clear that across all phases, the appropriate safeguards have been implemented to guarantee data anonymisation, protection and safety. In conclusion, an appropriate and well-orchestrated plan has been developed for both data acquisition and protection at all phases of the first stage of data collection.