Ref. Ares(2017)4261 - 02/01/2017

TagItSmart!
Smart Tags driven service platform for enabling ecosystems of connected objects

Grant agreement 688061

Enablers for FC-scanners

Deliverable ID:	**D 3.1**
Deliverable Title:	**Enablers for FC-scanners**
Revision #:	**1.0**
Dissemination Level:	**Public**
Responsible beneficiary:	UPC
Contractual date of delivery:	31.12.2016
Actual submission date:	
Approval:	Srdjan Krco

Start Date of the Project: 1 January 2016 Duration: 36 Months

Contents

Introduction ***3***
1.1.1 Objectives 3
1.1.2 Approach 4

Section 2 - Enablers architecture ***5***

Section 3 - Enablers for FC-scanner ***7***

3.1 Enablers for scanning **7**
3.1.1 2-D Code scanning general description 7
3.1.2 Implementation description 8
3.1.3 Reliability of 2-D code scanning 8
3.1.4 NFC scanning 9
3.1.5 Reliability of NFC scanning 11

3.2 Authentication and IDM enablers **13**
3.2.1 User authentication 13
3.2.2 Authentication methods 14

3.3 Context management enablers **21**
3.3.1 Determination of context using a smartphone 23
3.3.2 Enforcing context-based policies on FC-Scanner application 26

Section 4 - Enablers development environment ***28***

Section 5 - Reference applications ***30***

Section 6 - Conclusions and future work ***31***

Section 7 - References ***32***

Introduction

1.1.1 Objectives

In this document we define how FC[1] – enablers are defined. There are at least three different enablers that have to be implement to make FC - scanning happen. ***Reliable FC-scanning*** is essential for scanning process and ***User authentication*** is needed for secure FC code usage. During the reading procedure it is essential to check user access level. A ***context Management*** enabler has also important role in this project is also because we are using context for many purposes (checking security, giving more information to user etc.). Enablers can be integrated into other applications and that might be very common solution in the future.

In general, an FC-scanner is a device that performs QR (Quick Response) or NFC (Near Field Communications) decoding and can be a mobile device, a head mounted display as well as devices like Raspberry Pi, Intel Edison and standard PC/Mac running software code that makes them FC-scanners. FC-scanners can have functions which work in offline mode, so that they don't require any connection to any server, functions that require interaction with a TagItSmart compatible platform and some functions that connect to external data sources directly. Illustration of this architecture is shown in figure 1.

.

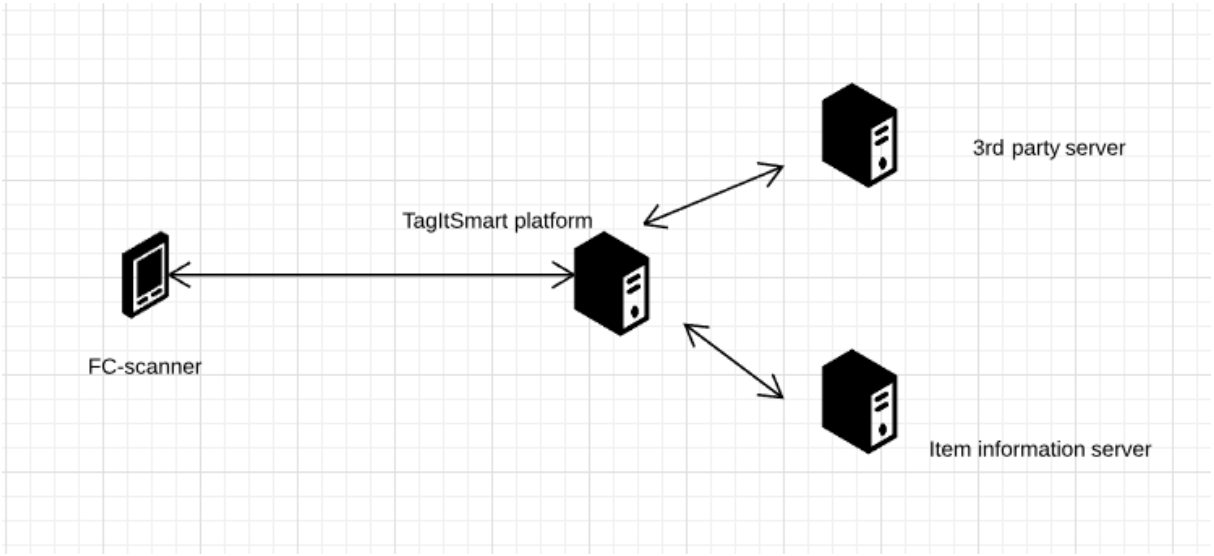

Figure 1: High level architecture of FC-enabler

Enablers for FC-scanner aim to support this architecture. An FC-scanner is able to scan SmartTAG codes (figure 2) and to deliver this scanning information directly to the user or forward it for processing to appropriate platforms.

[1] Please note that terms FC and SmartTag are being interchangeably used throughout this and other TagItSmart documents.

Figure 2: QR code example with embedded data about product, product line and sensor measurement (red)

To meet these objectives this document provides critical information of how enablers should be implemented so that they are able to function as a part of the TagItSmart ecosystem. This document doesn't describe how they should be implemented at low level coding.

This deliverable describes initial definition of a FC scanner. D3.2 due at M18 will ensure secure and reliable usage of the FC-scanner enablers presented in this deliverable document. In particular, D3.2 will investigate and analyse the security related aspects of the FC-scanner enablers (i.e., scanning, authentication and context-aware applications) and propose their corresponding countermeasures in order to ensure their reliable usage. In D3.3 (M24) final implementation of FC-scanners will described. By that time project partners will have already tested FC scanners in lab, controlled environment and field trials (WP 5).

1.1.2 Approach

Implementing FC-scanning in different platform is done by utilizing a multiplatform SDK (Software Development Kit). When a multiplatform SDK is used, production of enablers can be done on many development platforms (Windows, Mac, Linux) and testing can be implemented either on a real device or in the simulator. End product (final application) is usable on all supported platform (iOS, Android, Windows, Linux, Mac, etc.). In this project, the multiplatform SDK considered to be used in development process is Xamarin or some other cross-platform framework.

When creating enablers, it is important to focus on API's, because these interfaces are used to utilize the enablers. For different platforms API's need to stay similar if changes for some platforms need to be made. Also, enablers can be expandable with new features in the future. Enablers are packaged into a TagItSmart library so they can be used from any host application, which support TagItSmart platform.

Section 2 - Enablers architecture

A high-level architecture of the enablers indicating the main components and modules is shown in Figure 3.

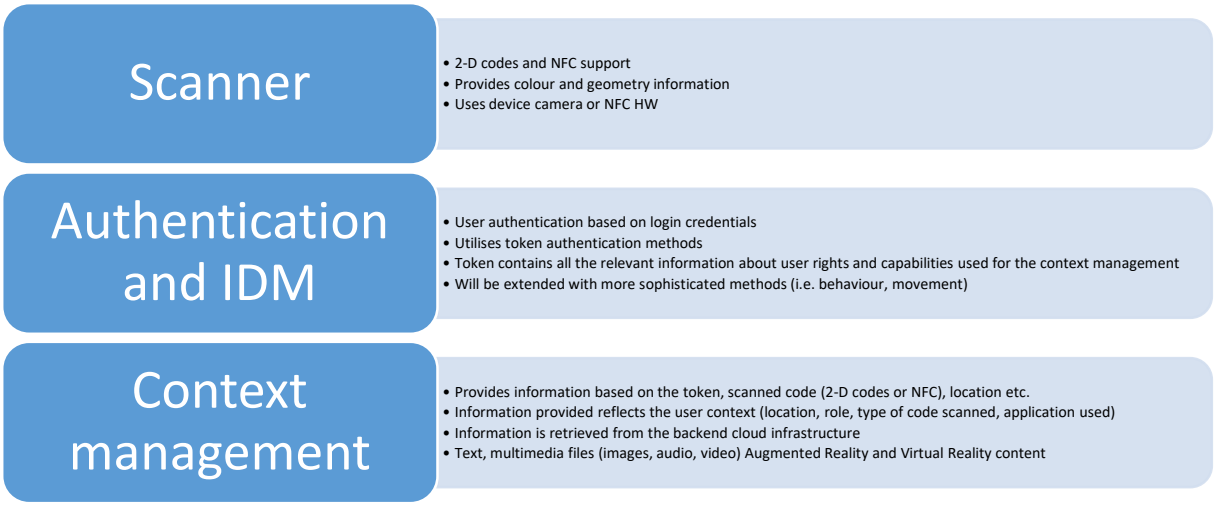

Figure 3: High-level architecture of enablers

The architecture stack consists of several layers providing specific functionality and are described below. The central layer in this architecture is **TagItSmart SDK layer** (figure 4) that consists of three enablers, namely:

- Scanner Enabler
- Authentication and IDM (Identity Manager) Enabler
- Context Management Enabler.

Access to the enablers is achieved through the appropriate **SDK interface** which is primarily used by the **applications** running on different platforms. The SDK is effectively built as a cross-platform library available on different platforms such as Android, IOS and Windows. The SDK will contain a set of reference applications which will demonstrate the usage of all enablers through the associated SDK API. Examples of these applications are provided in Section 5 -

The enabler library relies on the platform services such as file system, cryptography, network, camera, NFC modules and others through the appropriate **Access Modules layer** which is provided separately for each platform or through a cross-platform development IDE (Integrated Development Environment).

Access to the **TagItSmart services** is given through appropriate RESTful interface which is covered by the activities in WP1.

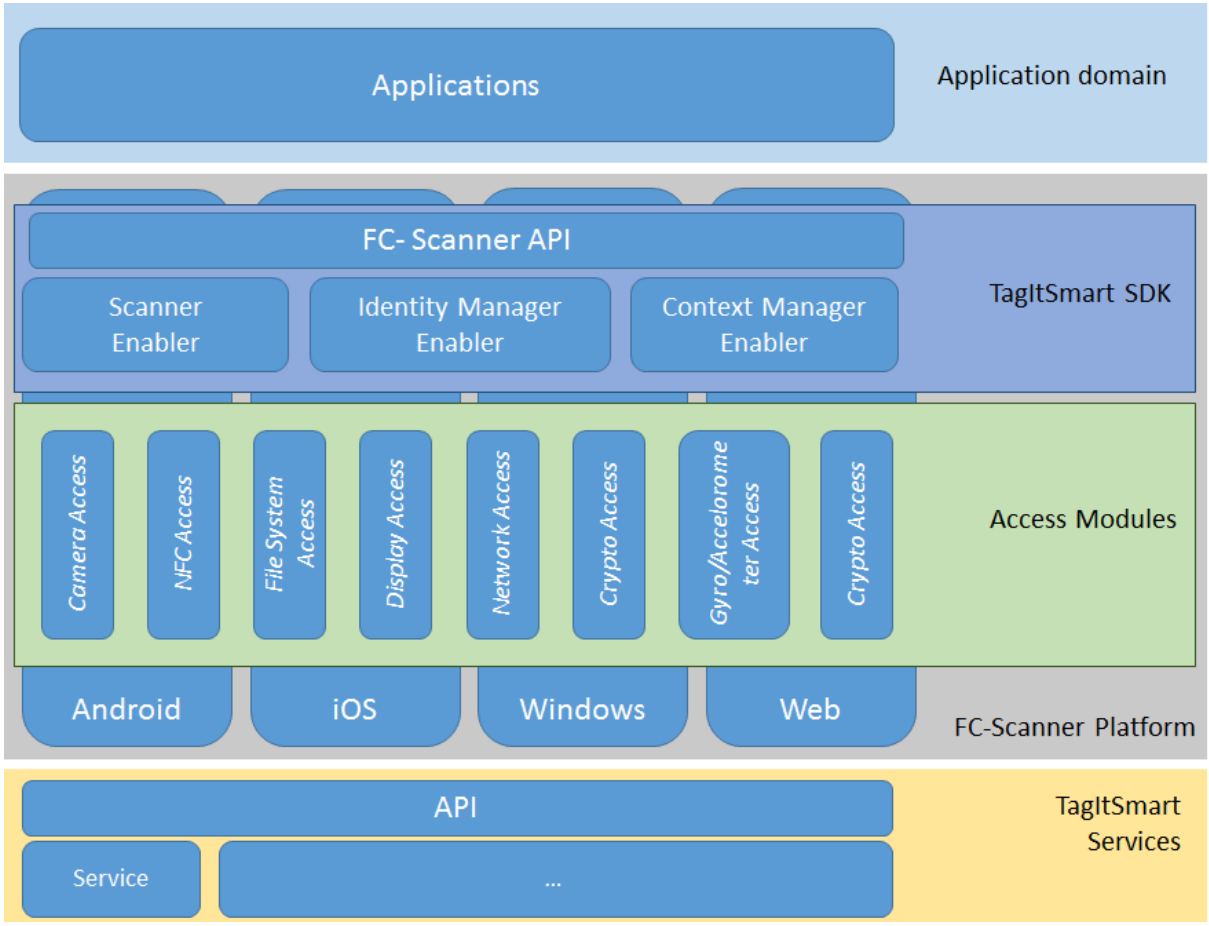

Figure 4: Top-level enabler architecture

Section 3 - Enablers for FC-scanner

Enablers for scanning are modules that make it possible for user to scan SmartTAGs. These enablers are responsible for setting up camera interfaces. Usually setting up camera requires to open a connection with the underlying camera hardware; after this connection has been setup, the application/library can modify camera settings and camera starts to deliver captured frames to the application/library. Depending on the operating system and device, frames are delivered and displayed either coupled or decoupled. If the coupled way is used, frames are displayed on the screen after or before they are used, but a new frame can only be obtained once the old one is processed. In the decoupled way, frames are delivered to screen separately from the processing of frames; this makes live camera feed smoother (higher FPS (Frames per Second)) to the user. Enablers for scanning are responsible for receiving frames from the camera and processing them.

Each operating system and device delivers frames differently, these frames can vary in colour mode used (RGB24, ARG32, YUV, etc.), when cross-platform SDK is used these details are usually hidden from the developer, but in some cases they need to be taken into account. After an image has been received and its image data are decoded they can be sent to the module that is responsible for SmartTAG detection and/or extraction. This process is described later in the document.

Basic features and interfaces are summarized in the table below.

Feature ID	Description
SC1	The scanner is able to decode QR, DM (Data Matrix) and NFC codes
SC2	Application initialises the scanner to use QR, DM or NFC mode
SC3	The scanner uses smartphone camera to decode QR code in the field of view
SC4	Colour information from the QR/DM code is returned to the application
SC5	Geometry information (viewing angle, location of the QR/DM code within the field of view) is returned to the application for each camera frame
SC6	The scanner uses NFC module to read the information from the NFC tag
SC7	Information from the NFC is passed to the application

Interface	
Input	**Output**
Scanner mode (2-D Code/NFC)	QR code information (value, colour, position, plane angle)
Operation (Continuous/single shot)	NFC code information

Table 1. Features and interfaces for enablers

3.1 Enablers for scanning

3.1.1 2-D Code scanning general description

As SmartTAG implementation is still under research it is not possible to know yet how exactly 2-D code scanning would work in the final enabler. However, preliminary steps for scanning are similar as in any other 2-D code scanner:

1. Acquire the image/frame from device or library
2. Locate 2-D code tag from image
3. Accurately detect edges (or corner points) of the 2-D code
4. Detect properties of 2-D code (orientation, rotation, etc.)
5. Extract binary information from 2-D code tag
6. Decode binary information based on appropriate 2-D code standard (QRCode/Datamatrix)
7. Decode sensor information from the 2D-code

After information has been decoded from a Datamatrix tag, the FC-scanner needs to decode SmartTAG sensor from the Datamatrix. The decoding process is described in more details in D4.1. The final result is that the FC-scanner is able to read the state of the sensor inside a SmartTAG and to forward this information either to a TagItSmart platform or to perform actions based on the sensor information.

3.1.2 Implementation description

Enabler for scanning is the component that provides 2-D codes decoding. This component is a C#.NET library based on the open source barcode library ZXing (Zebra Crossing, works with Xamarin.iOS, Xamarin.Android, and Windows Univerzal [27]).

QR code is detected by a 2-dimensional digital image sensor and then is digitally analysed by a programmed processor. The processor locates the three distinct squares at the corners of the QR code image, using a smaller square (or multiple squares) near the fourth corner to normalize the image for size, orientation, and angle of viewing. The small dots throughout the QR code are then converted to binary numbers and validated with an error-correcting algorithm. Error correction capability is to restore data if the code is dirty or damaged.

The data format of the QR codes and the underlying encoding algorithm will enable representation of the item ID together with the sensor type and value. The details of this implementation will be provided in WP2.

The main idea is that data part of the QR code includes static data and dynamic data. The dynamic data part of the QR code is created with functional inks, sensitive (the colour based functional ink will be used) to the change of the parameter it needs to sense and it will change its content with the change of this parameter. The error correction area of the QR code is static and enables reading of the initial state of the sensor.

Because it is likely that some implementations of the dynamic (sensor data) will require decoding processes that are not part of the existing QR/DataMatrix standards, access to the implementation will be needed by developers. Therefore, it is expected that the decoder element of the scanner will be ‘pluggable’ in some sense so that experimental decoding strategies can be implemented. Note, that any changes required to the standards in order to implement dynamic codes, will be submitted by project members to the appropriate standards body as part of project exploitation and dissemination.

3.1.3 Reliability of 2-D code scanning

This section discusses the main factors affecting the 2-D code scanning process. The main reliability problem that can arise is introduced in the data density encoded when compared to traditional 1D bar codes. It is often the case that the QR code is used to pack more data and enable off-line operation of the scanner. Therefore, in order to increase the reliability of the

scanning, less density data should be used and work in the on-line environment which is based on the primary key encoding linking to the back-end infrastructure content. This will also enable the error correction procedure to be more effective. Error correction is an important feature supported by most 2D barcode standards. This procedure involves encoding additional data in the barcode structure which then enables potential reconstruction of the original data in cases when partial damage or defect occurs within the generated barcode. This functionality to compensate for partial damage improves the credibility of the barcodes as a reliable machine-readable format. There are different error corrections settings associated with the most standards with the intention of increasing the readability and subsequently the reliability. However, there is a trade-off when selecting the desired setting of the error correction, as a higher level of error correction reduces the amount of actual information that can be encoded within the 2D code.

End user behaviour has a significant impact on reliability. SmartTags are read in many cases in non-controlled environments, with a number of factors that can decrease reliability:

- Illuminance
- Type of light source (and use of flash)
- Type of camera (ex. different mobile phones)
- Type of lens in camera
- Reading angle.

If we compare the existing 2D codes to the SmartTags, illumination and how colours can be reported by the enabler are more important than before. Functional inks use colour to represent sensor data and detecting it in different lighting conditions can be challenging. Aging of printed ink can also decrease reading reliability (due to the change of the colour). These factors and their influence for reliability of scanning will be reported in D3.3.

Other factors that impact scanning reliability include the type of the printer which creates the physical barcode, the media on which the barcode is printed as well as the quality of the scanner itself.

In order to increase the reliability of the scanner (in this case 2D), it is advisable to utilise the on-line approach with high-level error correction. Furthermore, high-contrast printing should be performed on non-glossy media. Utilising this approach, the reliability will be over 90% as reported in [61].

3.1.4 NFC scanning

Near field communication (NFC) refers to a set of communication protocols that enable two electronic devices, one of which is often a smartphone, to establish communication by bringing them within a few centimetres of each other. NFC-enabled devices, including smartphones and industrial RF readers, can read information stored on inexpensive NFC tags that are often embedded in products, packaging, and labels.

When an NFC tag is brought into close proximity with the NFC reader, the reader's RF energy is harvested by the NFC tag's antenna, which powers the NFC tag's IC (integrated Circuit). In the case of a Thinfilm NFC tag, the tag IC will wirelessly transmit, through the tag antenna, that tag's unique identifier and any sensor information back to the reader device. The tag data are delivered as a stream of bits in a format following one of several publicly available NFC data formats. Once the bit stream is received by the reader, the reader typically takes action from an installed app or via the operating system's built-in NFC service.

APIs in operating systems like Android and Windows allow developers to quickly and easily read NFC tag information inside mobile apps. Those apps can then deliver a rich user experience, using the mobile device's screen, to the consumer or other end user.

Penetration of NFC enabled mobile phones has increased during last two (2) years and it will continue growing. There are different types of NFC tags and not every mobile phone supports all of them. Figure 5 (below) show statistics and forecast how many mobile phones can read NFC tags. It also shows how many mobile phones can read NFC Thinfilm Electronics tags (used in TagItSmart use cases).

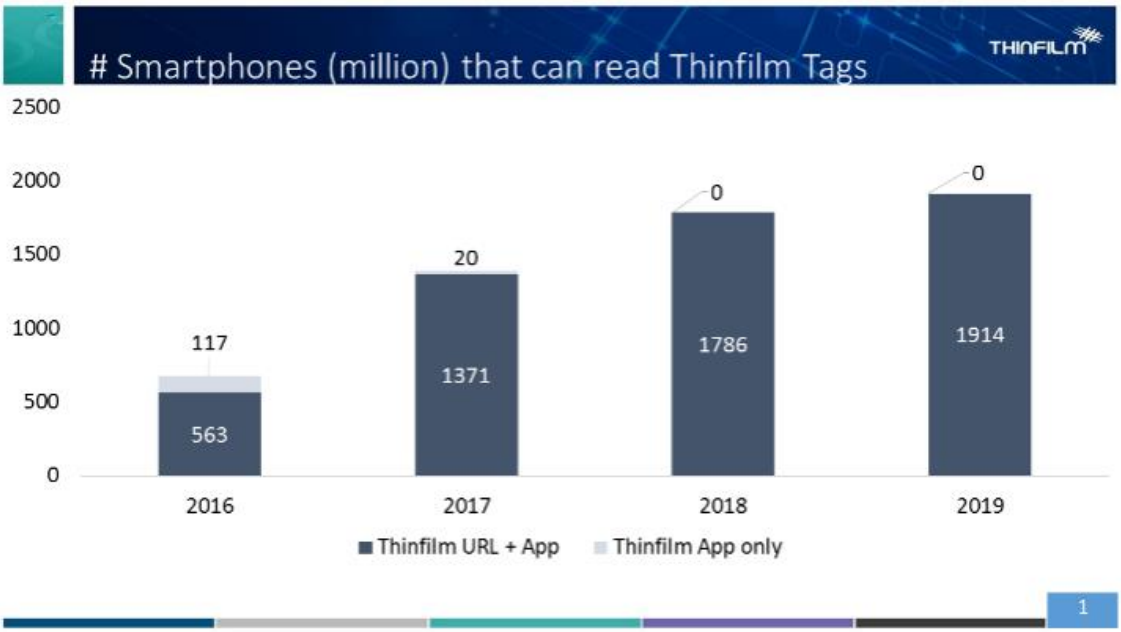

Figure 5: Forecast for the number of smartphones that can read Thinfilm tags

When an FC-scanner is in close proximity to an NFC tag the scanner reads code from the tag by getting a stream of bytes from NFC module and decodes the stream according to NFC Speedtap specification [25]. Reading data and decoding is typically provided by the NFC Service of the underlying software platform such as Android [26]. Decoded NFC Speedtap will contain information about the identity of the tag and data from the sensors attached to the tag.

Smartphones typically read NFC tags at a distance of a few centimeters (up to 4 cm). The reading distance may vary slightly between smartphone models and between tag configurations due to variations in transmit power, antenna design, and other factors. Since NFC is often referred to as a 'tap' or 'touch' technology, the expectation is that a consumer will place their smartphone in a very close proximity to the NFC tag in order to trigger the action. NFC tag read speed varies based on the length of the data transmission and the specific NFC protocol used. TTF (Tag Talks First) protocols such as Thinfilm's NFC Barcode protocol minimize overhead and deliver the tag's full data within 5 milliseconds from the start of the tag read. Low-overhead NFC protocols can verify correct integration of an NFC tag in a product/package while maintaining existing packaging/production line speeds in high-volume applications. Higher-overhead RF protocols, including common RTF (Reader Talks First) protocols, and higher-capacity tag memories can prolong the total NFC transaction

time. One should consider these factors in cases where packaging/production line speed is a priority."

IOS devices contain the same NFC technology as Android based devices, however, Apple has not opened up the technology to third party developers. Now in 2016, Apple is preparing the release of new iPhone models in addition to an updated iOS. Even with this new version in the works, the insiders are speculating that Apple will continue to limit their use of NFC technology to Apple Pay. Therefore, the NFC support on iOS within the TagItSmart SDK will be limited or be virtually nonexistent until the developer access is provided.

Until iOS supports third party NFC applications, focus will be on markets where Android penetration is large such as for instance in Asia, Germany and Spain to name a few.

3.1.5 Reliability of NFC scanning

3.1.5.1 NFC tags on surfaces containing metal

It is not uncommon that packages contain electrically conductive materials such as metals. It could, for example, be blister packs, foil-lined packages or metal caps on bottles. Placing an unshielded RF tag on, or near, a conducting material can have a detrimental effect on the RF tag's read range. It can even make the RF tag unreadable.

The solution to this problem is a ferrite shield, a material with much higher permeability than air, which is placed between the tag antenna and the conducting material. The effect of the ferrite shield is illustrated in Figure 6. Placing the RF tag on, or near, only non-conducting materials has no effect on the tag's read range. However, if the RF tag is placed on an electrically conducting material, such as a metal, the magnetic field generated by the reader's coil antenna will induce the so-called eddy currents in the conductive material, which in turn will generate a magnetic flux that opposes the incident flux. The eddy currents cause the magnetic field to be diverted away from the conductive material and significantly reduces the magnetic flux through the tag's coil antenna. As mentioned, this has a strong negative effect on the tag's read range and can, in the worst case, make the RF tag unreadable. However, the magnetic flux through the tag antenna can be restored by inserting a ferrite shield between the RF tag and the conducting material, giving the RF tag an acceptable read range. This can be explained by that the magnetic field entering the ferrite layer is diverted away from the conducting material, which consequently reduces the eddy currents. Thus, the ferrite shields the conducting material underneath.

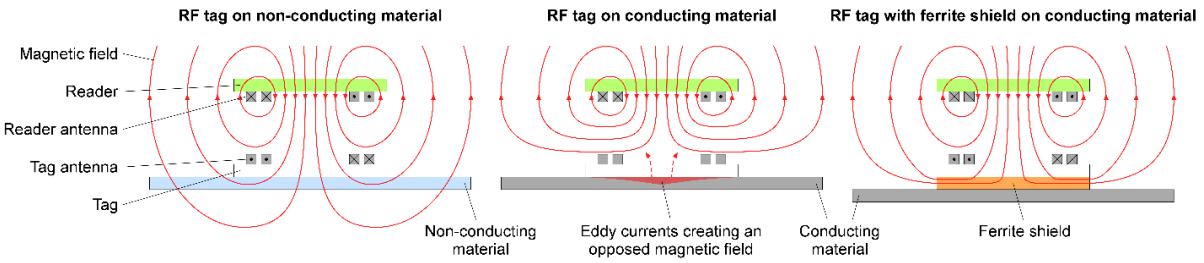

Figure 6: Illustration of the magnetic field distribution when attempting to read an RF tag at a fixed distance for three different scenarios.

For the NFC applications where the NFC tag will be placed closed to metal, a magnetic shield must thus be used. Magnetically shielded ferrite inlays can be sourced through Thinfilm.

3.1.5.2 NFC readability through water

Water and other liquids can alter RF propagation conditions. These changes are dependent on the frequency of the RF signal and are much more severe at higher frequencies. NFC technology can typically be read in the presence of nearby water, whereas UHF RFID technology struggles with severely restricted read range. Figure 7 compares read ranges at typical ISM-band RF frequencies through various types of water. For the best system-level performance, closely evaluation and conducting experiments to find the best solution for each application's unique requirements is needed.

	Low Frequency 125kHz	High Frequency 13.56MHz	Ultra High Frequency 800MHz	Microwaves 2.45GHz
Salt Water 4S/m	71cm	68mm	9mm	5mm
Fresh Water 30μS/cm	26m	2.5m	32.5cm	18.6cm
Fresh Water 2000μS/cm	3.2m	30cm	4cm	2.3cm

Figure7: Readability of RF frequencies through water [49]

3.1.5.3 NFC interference and readability

NFC tags have a low risk of interference. Other ISM band wireless technologies, including UHF RFID, Bluetooth, and WiFi, do not typically interfere with NFC's 13.56MHz operation. Furthermore, NFC's positioning as a 'tap' or 'touch' technology means that the reader and the (typically unpowered) tag are placed in close proximity to each other, thus minimizing interference from external sources.

The operating range of typical NFC tags are from -20 °C to 65 °C. This means that the NFC tags are readable in most reasonable environments where humans have access to smartphones. Furthermore, the storage requirements of NFC labels are typically 15 °C to 25 °C at 20-70 %RH.

3.2 Authentication and IDM enablers

3.2.1 User authentication

FC Readers will have different type of users. There might be a robot (or in general a machine) on a factory floor or an end user in a retail store. User authentication service has to be customized for different purposes and at the same time it should be standardized to match the underlying IoT platform requirements.

In the authentication enabler, our main focus is on the design and development of various robust authentication mechanisms that can be used to securely and simply (i.e., with higher usability) authenticate an honest user. The main aim of the authentication techniques presented in this section is to keep the false negatives (denying access to an honest user) and false positives (allowing access to an imposter) as low as possible. This enabler will not focus on security aspects of the FC-scanner such as *communication security* (i.e., RFID/NFC communication between FC-Scanner and SmartTags, and wireless communication channels between FC-Scanner and TagItSmart platform), *operating system (OS) security* (i.e., security of the underlying OS on which FC-scanner is running and security from the possible nearby malicious applications running alongside of the FC-Scanner in the same OS or device. These security aspects will be investigated and covered in detail in the next deliverable i.e., D3.2 in M18.

The aim of the user authentication is to verify that the user requesting access is the user they are claiming to be. Authentication can be divided into several categories. *Active* and *passive* authentication represent the level of input required from the user. Active requires the user to perform some action that they would not normally do, this includes entering a password or performing a gesture on the phone. Passive authentication uses actions that are already performed, examples of this include gait recognition and how the phone is picked up to answer a phone call. *Knowledge-based* authentication requires that the user knows some secret such as a pin or a password. *Biometric* authentication is based on who you are and characteristics that are considered unique. In addition, some authentication methods require the user to be in possession of an item that uniquely identifies them, such as an ID card or a USB dongle, this often contains a private key as a part of a PKI.

3.2.1.1 Active authentication

The most common form of authentication is active, knowledge-based authentication. Here it is indicated to the user that they are required to authenticate, and the user then performs an action. The sole purpose of the action is to authenticate, and it has no other use. A classic example is entering a username and the corresponding password. A more recent approach is to draw a pattern on the screen of a mobile. A study in 2010 showed that there is low usage of security devices on mobile phones due to low acceptance of the offered authentication methods such as PIN.

Active biometric approaches are those such as fingerprint or facial recognition. Sometimes there are attempts to combine the action with another, more useful action, such as putting the fingerprint reader on the home button of a mobile phone. This makes the approach more like a passive biometric approach.

3.2.1.2 Passive authentication

Passive authentication aims to make authentication more transparent to the user by using actions that are already being performed in the course of a day to authenticate the user.

Examples include gait recognition [4,5], mobility [10], phone pickup [11,12] and touch-based biometrics [13,14,15,16].

Context-aware authentication uses context information such as location and time [17] instead of, or in addition to, passwords. A recent study [18] has shown that mobility is a unique human trait that can be used to uniquely identify individuals. In analysis of mobility data (mobile phone mast connection) from one and half million individuals, it was shown that four spatio-temporal points were sufficient to uniquely identify 95% of individuals.

Passive context-aware authentication uses contextual data such as environmental conditions (lighting, noise level etc.) in order to authenticate users. Witte et al. [19] use an SVM (Support Vector Machine) and contextual features for a context-aware authentication framework.

In addition, many authentication approaches assume that the phone needs to be unlocked, or that it will be used for a secure action. In the future, there might be many more operations where the phone will not be unlocked and used. Examples include, using it as a key to access your home or car, a source of data etc. In these cases, continuous passive authentication is more appropriate. Passive authentication methods are able to continuously and implicitly authenticate the user. This circumvents the problem of point-of-entry authentication where after the initial authentication, the user can perform many tasks without re-authenticating as long as the device does not lock.

3.2.2 Authentication methods

3.2.2.1 Token based authentication

Authentication and Identity Management (IDM) is the component that provides management of user identities, their authentication, authorization, roles and privileges within system. IDM system consists of policies that define whether the user has permissions to use the application based on login credentials and whether it has access to perform some action depending on his application type, role type, location and other factors.

Preliminary features and interfaces for IdM can be found in the table below.

Feature ID	Description
AIDM1	The enabler provides functionality for the user authentication based on login credentials
AIDM2	The enabler is based on token authentication methods
AIDM3	The enabler communicates with the appropriate components on the backend cloud infrastructure to verify the credentials
AIDM4	The enabler returns the appropriate token upon authentication
AIDM5	The token contains all the relevant information about user rights and capabilities
AIDM6	The token has certain validity period
AIDM7	Authentication can be performed using certain phone movement patterns (TBC)

Interface	
Input	**Output**
User credentials (user name/password)	Capability token containing access rights, validity and other parameters

Table 2: Features and Interfaces for Token based Authentication and Identity Management

Authentication and Identity Management (IDM) is the component that provides management of user identities, their authentication, authorization, roles and privileges within the system. An IDM system consists of policies that define whether the user has permissions to use the application based on login credentials and whether it has access to perform some action depending on his application type, role type, location and other factors. IDM functionality for the purpose of project pilots will be developed on top of the ASP.NET Identity system provided by Microsoft which represents the membership system for building ASP.NET Core web applications, including membership, login, and user data. The diagram below shows the components of the ASP.NET Identity system.

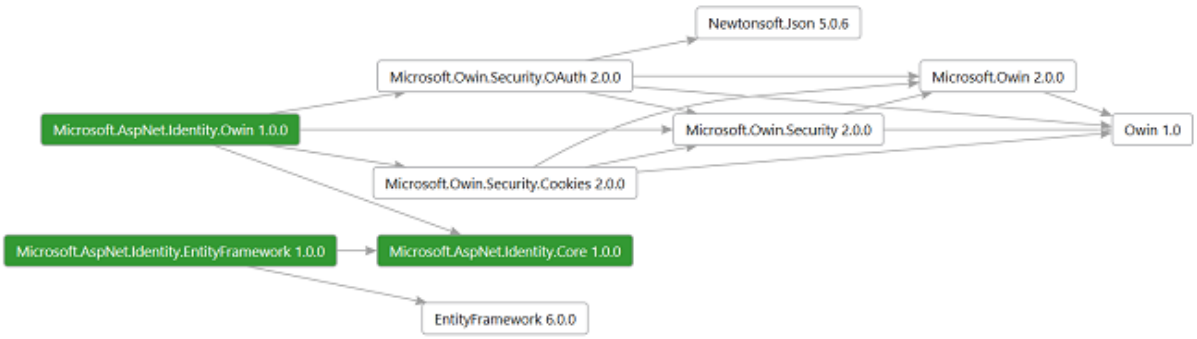

Figure 8: ASP.NET Identity system components

This component will be developed using ASP.NET Web API framework which is a platform for building RESTful applications on the .NET Framework for building HTTP services that reach a broad range of clients, including browsers and mobile devices. It will be exposed as the RESTful web service which supports authentication and will provide secure token-based authentication. This means that instead of sending the hard credentials in every request, the client will send the token to the server to perform authentication and authorization. Only authenticated and authorized users can access the resources. The token sent to the server is self-contained which holds all the user information needed for authentication. The token has a validity period that is configurable and can be adjusted. An anonymous access or access without the appropriate rights will get unauthorized error code when trying access data.

An authentication scheme based on tokens follow these steps:

- The client sends their credentials (username and password) to the server.
- The server authenticates the credentials and generates a token.
- The server stores the previously generated token in some storage along with the user identifier and an expiration date.
- The server sends the generated token to the client.
- In every request, the client sends the token to the server.
- The server, in each request, extracts the token from the incoming request. With the token, the server looks up the user details to perform authentication and authorization.
    - If the token is valid, the server accepts the request.
    - If the token is invalid, the server refuses the request.
- The server can provide an endpoint to refresh expired tokens.

Microsoft SQL Server will be used as a storage provider for ASP.NET Identity although it can easily be replaced with another storage provider. C# will be used as the coding language for developing IDM component. This component can be deployed on the Azure cloud platform or as an on-premises hosted solution.

3.2.2.2 Location-based authentication

It was shown by Edmond Locard in 1930 that 12 points are needed to uniquely identify a fingerprint. Fingerprints are an example of an active approach to authentication. A recent extensive study of human mobility patterns has shown that, like fingerprints, human mobility patterns are highly unique to individuals [17]. The research examined the mobility patterns of one and a half million individuals and found that human mobility traces are highly unique. It was determined that four spatio-temporal points were sufficient to identify 95% of individuals.

Due to the above result, location-based authentication aims to use mobility traces to uniquely identify, and thus authenticate, users. The aim of the authentication module is to authenticate a user based on their current location, and their previous locations.

3.2.2.2.1 Architecture of the location based authentication module

The authentication module will operate in two phases. The first phase aims to learn the user's mobility pattern. The GPS sensors are used to collect location measurements for a period of one week. Once the measurements are collected, a model of the user's normal walking pattern will be constructed. The authentication module then enters the testing phase

where GPS data is provided to the model in order to perform authentication. The testing phase will constantly provide authentication to a user of the smartphone. The high-level architecture of the authentication module is shown in figure 9.

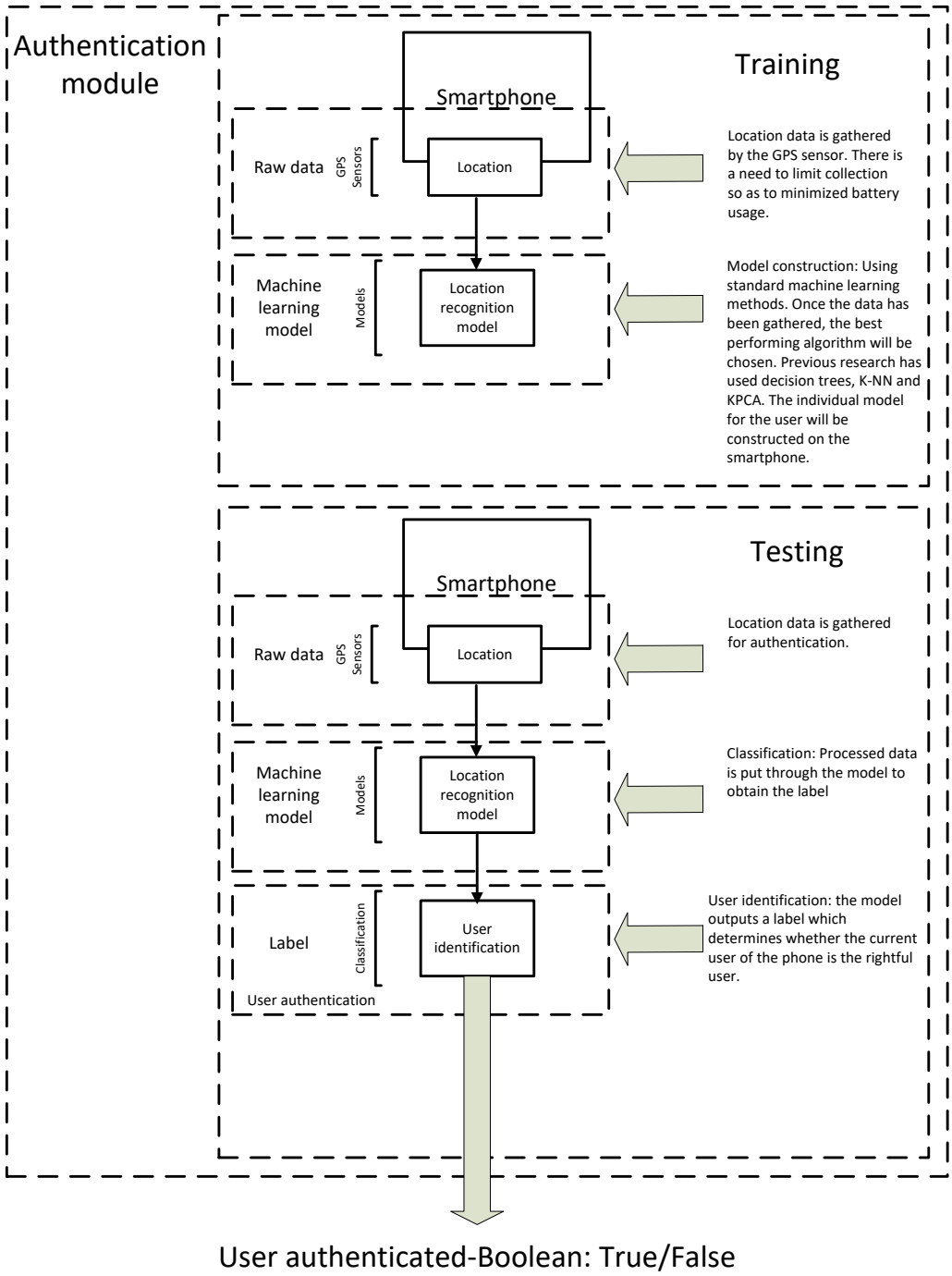

Figure 9: Architecture of the location based authentication module

3.2.2.2.2 Proposed approach

The aim of location-based authentication is to use traces of a user's location through space and time in order to provide authentication. Location will be provided by GPS coordinates

reported by the smartphone at regular intervals. The current location and historic location of a user will be represented in a suitable form for a machine learning method to construct a model of the user's normal mobility pattern. The time element is then removed with the ordered sequence being retained. This transfers the problem into an ordered time-series with the [*latitude, longitude*] tuple representing a point in space. The series of tuples represent movement within space and time through the ordered sequence.

With the time ordered sequence of data in this format, a standard machine learning method will be used to construct a model of the user's location data. Examples of methods used are one-class support vector machine, PCA (Principal Components Analysis) or KPCA. The method used will be chosen to balance two costs, authentication accuracy and efficiency. Efficiency is particularly important as many machine learning methods incur a significant computational and memory cost. There is a requirement to reduce computational cost due to the drain it causes on the battery. There is also a requirement to reduce the memory cost due to the constrained nature of the smart phone.

3.2.2.2.3 Beyond state-of-the-art

Many novel biometrics that are used for authentication suffer from high levels of noise. For example, the performance of gait recognition is decreased due to different clothing, shoes and type of material being walked on. In addition, other novel biometrics suffer from attacks which make it possible for an illegitimate user to be authenticated. An example of this is facial recognition which can be fooled by photographs and masks, see [23] for a review on this problem.

The location-based authentication method is subject to less noise and is less susceptible to an authentication attack. There is less noise in the data as location has a larger granularity than the accelerometer readings used in authentication methods such as gait recognition. In addition, biometrics such as facial recognition suffer from issues caused by varying lighting conditions and camera angles. Location-based recognition will use simple GPS coordinates which will deviate slightly from day to day, however the alterations will be minimal compared to the issues which gait recognition and facial recognition suffer from.

Although it has been shown that mobility acts like a fingerprint for individuals [18], there has been less research done in the area of authentication using location. Lehtonen et al. [20] explores location-based product authentication. The aim is to identify suspicious products from incomplete location information. Hidden markov models (HMM) are used with location-based information in order to identify counterfeit products in the supply chain. An alternative location-based method for a smartphone [21] uses location-based information provided by the WiFi Access Points where authentication questions are generated based on users' locations. A method that incorporates GPS data from a smartphone into an authentication method is detailed by Fridman et al. [22]. The approach uses multiple modalities in order to provide authentication. The modalities examined are; text typed via soft keyboard, apps visited, websites visited, location (GPS or WiFi). Location is specified as latitude and longitude which is provided by the GPS module on the smartphone. Classification is performed using a support vector machine (SVM) with the radial basis function. The SVM produces a score for each pair of coordinates which is then calibrated to probability using Platt scaling. The method is able to achieve an equal error rate (EER) of 0.01 after 30 minutes of user interaction with the smart phone. The performance is due to the fusion of the multiple modalities and the fusion of the decisions.

3.2.2.3 Biometrics for authentication

Biometrics refers to the metrics formed from human characteristics and traits. It is used as a form of user identification and authentication with well-known examples including fingerprints [28], face recognition [50], and iris recognition [51]. With the advent of smartphones with on-board sensors, it has become possible to obtain alternative biometrics by using the sensors to measure the human characteristics. These biometric alternatives can be used to unobtrusively authenticate users by using actions that are carried out naturally, as opposed to more obtrusive authentication such as entering a password or performing a pattern action on a screen. Examples include walking gait [52] [53] [54], smartphone pick-up action [55], keystroke and touchscreen use analysis [56] [57]. An alternative biometric such as walking gait can also be used to authenticate a user while they are not using the smart phone. This will become important in future applications when smart phones become data generators for IoT. It is important to be aware of the identity of the holder of the mobile device even when it is not being used.

Earlier works [46] [47] have suggested touch based biometrics to distinguish different users. The features considered are: keystroke or touching behaviour such as, pressure, area, duration, position, and the reaction of devices such as, acceleration and rotation. These features help the system to model the user well, by considering the user habits, the user actions, and the device reaction. Multi-modal biometrics based techniques have also been considered to improve the authentication mechanism in case some of the modalities fails, hence improving system performance and reliability. The features that are considered in [47] are: slide-unlock features, pickup movements, and voice features while answering a call. In article [48], the authors have considered hand movement, orientation, and grasp (HMOG) to authenticate the user continuously with higher accuracy even when the user is mobile.

3.2.2.3.1 Keystroke-based authentication

Unfortunately, traditional password-based (or PIN- or pattern-based) authentication schemes commonly used on mobile devices have a number of weaknesses that can inadvertently expose the user to security breaches. First, they are susceptible to guessing attacks, with as many as 91% of the passwords found in the top 1000 list [31], a problem exacerbated by the constrained nature of mobile devices that encourages users to select simpler and weaker passwords. Second, they are susceptible to smudge attacks, where attackers infer passwords from the finger smudges left on the touch screen [32]. Finally, they are susceptible to shoulder-surfing attacks [33], where attackers rely on direct observation to steal passwords in a public setting. Recent attacks have also become automated and more sophisticated, with attackers stealing passwords using low-end cameras and fingertip motion analysis through repeated reflections [34].

Interestingly, studies have shown that users are generally favourable to alternative authentication mechanisms [35], which have spurred research on biometric authentication for mobile devices. Several schemes have been proposed in recent years, such as identifying users based on their gaits [36], shake motions [37], phone-to-ear gestures [38], touch gestures [39], or keystroke dynamics [40].

The access control system in mobile devices should ensure that only the legitimate users have the right to access the FC-scanner functionalities. As, smartphones becoming an integral part of our life, earlier authentication mechanisms such as, PIN and fingerprint based authentication are seemingly no longer effective in case of new and more sophisticated attack vector such as shoulder surfing, smudge, video capture, and inside information leakage. Hence, the system demands that the authentication has to be done continuously,

and unobtrusively, i.e., it should not demand to engage the user often to get authenticated. This brings to the design and development of continuous authentication mechanisms.

As the FC-scanner application will be running on smartphones which also keeps other confidential information about its user. The portable nature of mobile devices, and the great exposure to various security attacks [41], calls out for stronger authentication mechanisms than simple password-based identification. Biometric authentication techniques have shown potential in this context. Unfortunately, existing approaches are either excessively prone to forgery or have too low accuracy to foster widespread adoption [42].

Measuring the accuracy of a biometric identification system is a well-studied problem [43]. In the literature, e.g., [44] [45], the Equal Error Rate (EER) is a common accuracy metric. The EER reflects the accuracy of the authentication system when the False Match Rate and the False Nonmatch Rate are equal. It is expressed as a single value between 0 and 1, with lower values representing better accuracy.

3.2.2.3.2 Proposed Approach: keystroke dynamics with multiple sensor modalities

Keystroke dynamics have been widely adopted as biometrics for characterizing users' behaviour. From hardware to software keyboards, from fixed to free-text input, from traditional devices to smart devices, keystroke dynamics has successfully authenticated interactive users by exploiting their intrinsic typing characteristics. In particular, keystroke dynamics relies on timing of key-press and key-release events, associating them in different ways and extracting features such as the key hold time, inter-key-press time, etc.

With the technological advancements in mobile devices, a fairly large number of sensor modalities are available for biometric authentication purposes, such as accelerometer, gyroscope, air pressure, temperature, and many others. In this approach, we specifically focus on movement sensors (i.e., accelerometer and gyroscope). Keystroke dynamics along with multiple sensor modalities augments, keystroke timing features borrowed from traditional keystroke dynamics with the sensor based features derived from real-time data sampled from movement sensors. Building on this rich feature set (i.e., keystroke features and movement sensor data), the keystroke dynamics with multiple sensor modalities techniques will implement different detection algorithms which includes the mean and k-nearest neighbours (kNN) algorithms based on Euclidean (unweighted, weighted, normed, normed weighted) and Manhattan (unweighted, weighted, scaled, scaled weighted) distances.

We will include “keystroke dynamics with multiple sensor modalities”, as a biometric technique to authenticate users typing on mobile devices (e.g., smartphones, tablets etc). The key idea is to characterize the typing behaviour of the user via unique sensor features and rely on standard machine learning techniques to perform user authentication. Keystroke dynamics with multiple sensor modalities based system will combine features from traditional keystroke dynamics techniques along with the features from prior sensor dynamics techniques, thus, leveraging the unique synergies between these two classes of features on modern smartphone devices. Our key intuition is to associate sensor generated data to a sequence of key-press events to improve the accuracy and robustness offered by traditional keystroke dynamics techniques. In particular, the proposed authentication system relies on both,the built-in sensors (e.g., accelerometer, gyroscope) in mobile devices, and the keystrokes timing (e.g., hold time) information. We believe that this rich set of features can

be potentially generalized to different biometric authentication systems, thus motivating our focus in this usage of this technique.

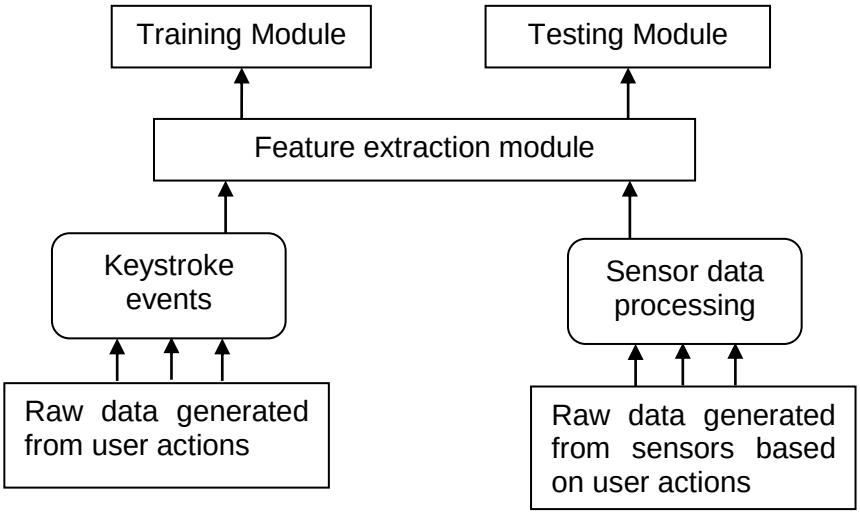

Figure 10: Overview of keystroke-based authentication system

During the authentication session (i.e., either for training or testing purposes), the user is requested to enter a fixed-text password, which is immediately processed by our authentication system for analysis. As the user interacts with the system, it intercepts (and records) all the generated key-press events and periodically samples movement sensor data from the accelerometer and the gyroscope. The high level architecture of the authentication system is shown in Figure 10.

As shown in Figure 10, all the data collected from key-press events and sensors in the form of sampled values are processed by system's feature extraction module, which translates all the previously recorded events into features suitable for our detection algorithms. In particular, the training module processes all the features gathered during a training sessions to build or update, in case of repetitions, a sensor enhanced keystroke dynamics profile is associated to a given user. The detection module, in turn, matches the features gathered during a testing session against the entire known user profiles to authenticate legitimate users (or detect adversaries). In particular, the system works in following four phases: (i) Data collection, (ii) Feature extraction, (iii) Detection, and (iv) Testing.

Along with the advances in biometric authentication (such as keystroke dynamics), however, attacks have also become much more sophisticated and many biometric techniques have ultimately proven inadequate in face of advanced attackers in practice. Therefore, we will investigate and analyse the effectiveness of our sensor enhanced keystroke dynamics technique which is a recent mobile biometric authentication mechanism that combines a particularly rich set of features extracted from keystroke events and sensors. The details and investigative results of various attacks and their countermeasures on different biometric authentication techniques including the keystroke dynamics will be discussed in detail, as part of the TagItSmart 3.2 deliverable which will be produced at M18.

3.3 Context management enablers

This section provides an introduction to the context management structure of the enablers. The method by which context is determined for a smartphone is outlined, along with the

architecture of the components that are used to store the context of smartphones as well as to enforce context based policies.

Context awareness subject have been developed around the systems that ask for complex interaction, where functional adaptation to changing environmental variables are needed for planned service delivery. Several existing application areas for embedded systems have already become more collaborative. For example, a connected car can “talk” to other cars and to the road or traffic management system. Remote patient monitoring has evolved in collaborative monitoring devices and connected healthcare systems, clustered dynamically depending on the monitoring goal. The basic intuition is that providing global context will enable more efficient operation of an embedded system, especially in the case of dynamic environments. The global optimization and control can provide more insights into how the local systems may operate, enabling proactive handling of the local level constraints in a heterogeneous and distributed system – as one composed of tag, scanners and associated IT systems for TagItSmart case. The technological solutions around Cloud computing and Big Data can enhance the development of such systems through opening/extending the processes from the local to the global level and back. However, there are several challenges to be resolved in order to make such a contextualization and control feasible. The key challenges include efficient and distributed processing, quality models, and fulfilling non-functional requirements in a heterogeneous environment, as well as supporting ad-hoc (real-time) changes in the underlying local-global communication. This will also require performance at scale and real-time event processing, optimization and control methods to support various critical constraints such as latency, safety, security, and energy efficiency. Current research and existing solutions only cover parts of these challenges still mostly by vertical integration and at the local level. As per current practice, local control loops are not part of the global optimization, as is the case for our project. This concept is the foundation for **developing complex applications** that sense changes in the environment and utilize collaborative control mechanisms in order to interpret the local situations and react/interact promptly according to global context, consistency, and availability requirements. The concept of extended control-loop could be implemented through the following goals:

- enable global real-time cooperation and control between highly distributed and heterogeneous disparate components (while allowing fast feedback between actuation and sensing) by providing an open, event-driven architecture, protocols and tools for handling fast and secure feedback control loops across multiple platforms and systems of systems, enabled by software-defined solutions.
- Enable adaptation and reconfiguration of multiple configurations in order to handle complex and dynamic situations, as well as emerging properties and functionalities of systems.
- de-verticalisation of platform services and applications by providing an open and interoperable framework and common interfaces between heterogeneous component participants via specialized Cloud repositories, which will lead to time and cost savings for the design, development, and deployment of solutions.

Contextualization will be an important component of the TagItSmart framework that will allow integration and development of sensing, actuation, networking and interface technologies. The context processing mechanisms will provide **flexibility with respect to the analysis**. This means that while interpreting sensed data in one way within one domain, it could be **interpreted differently in another domain**. TagItSmart will **support adaptability within individual instances** based on the context **across managed systems**.

Contextual awareness was consistently studied over the last years in the EC projects such as iCORE, ASCENS, COSMOS or RERUM. Those projects are relevant for TagItSmart since they also refer to Internet of Things and Cyber Physical Systems domains as our project and as such represent a valuable source of inspiration. The context awareness approach aims to support autonomic capabilities at a node or a connected system level using and maintaining a Knowledge repository of a certain systemic instance – in a MAPE-K view. Its role, as is aimed in the TagItSmart too – is to fulfill the role of dynamic memory and support decision processes.

Essentially for IOT, context awareness relies on a few key pillars. The first one is selection of context parameters, relevant for context composition, usually linked to non-functional KPI, constraints between them and functional KPIs. The second one is the instrumentation of those parameters, how those parameters are measured and acquired; here we can refer to sensing and sensed data streaming. The third one is related to the analytics of sensed data, and being a measure of complexity of awareness provided that it covers scenarios from simple detection of thresholds up to predictive or prescriptive analytics. In the iCORE case [58] the Situation Awareness concept that relies on two formalized knowledge models and their associated observers infrastructure is proposed: Real World Knowledge (referring to the acquired perception of out of an iCORE system world) and System Knowledge (referring to the acquired perception of an internal system stasis, relevant for example for load balancing of different service containers). It should be noted that considered observers are doing more than simple signal acquisition, i.e. fusion, analysis and knowledge extraction.
The COSMOS project introduces the concept of Context View [59] that embodies relevant metadata, their interdependence and connects current observed values. Of course, considered subcomponents should be instrumented in order to provision the data needed for a certain time context snapshot generation.

RERUM project [60] defines the Context Manager concept as a function of the middleware. The component is fed by defined monitoring infrastructure receiving normalized streams of data and events for near real time processing. Due to the specifics of the project, the main functional responsibility is to manage privacy and security access policies to IOT services.

Further details of the context management component will be provided in Deliverable 3.3.

3.3.1 Determination of context using a smartphone

Determination of a user's context can be a useful piece of information for the TagItSmart system. The current context can alter the information that is presented upon the scanning of a smart tag. For example, if the current context is determined to be "At supermarket", the tag of a food item can present information concerning the ingredients of the item and health information. This will allow the purchaser to make an informed purchase. However, if the current context is determined to be "At home", the scanning of the QR tag can present information such as cooking instructions and recycling information. It can also report relevant information to the manufacturer such as the date of scanning, to determine how long it was kept. Additionally, the context is further determined with the type of devices used for scanning such as smart phone, Augmented Reality head mounted display such as Microsoft HoloLens, VR (Virtual Reality) head mounted display such as Oculus Rift or other types. Therefore, the information presented to the user will also be dependent on the capability and functionality of the device used.

There are different types of context which are used to determine an action that should occur. The primary types of context are; location, identity, time, activity and the type of device used.

Location can be addressed using GPS and other indoor positioning systems. Identity will be addressed by the authentication module. The aim of the context management enabler is to determine the current activity. This will enable a dynamic scanning of SmartTags and different information being presented to the user depending on the context.

3.3.1.1 Architecture

The smartphone will report the current context to the Context Manager. The Context Manager will store the reported contexts for Smartphone IDs. This information can be queried by other components that require the current context of a specific smartphone as detailed in Fig 11. On the detection of a context, the context is posted to the Context Manager by the Smartphone over the internet. This context is then available on the Context Manager for other applications to query. For example, when the smartphone is used to scan a QR code or a NFC tag, this is transmitted to a backend application. The backend application then queries the Context Manager to determine the current context. The Context Manager returns the current context of the smartphone, and this is used by the backend application to determine the information that should be returned to the smartphone based upon the current context. This process allows the backend application to vary the content returned to the smartphone based on the current context.

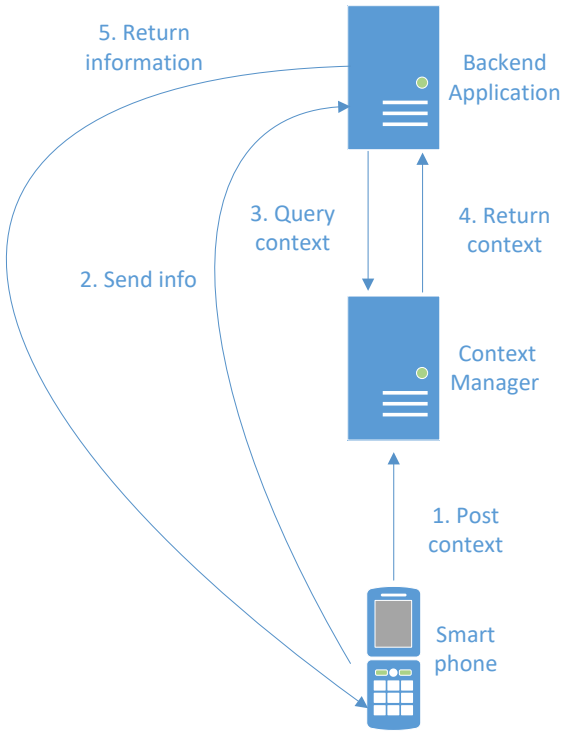

Figure 11: Context Management workflow

3.3.1.2 Determining context using a microphone

Smartphones contain a microphone for transfer of voice across the telephony network. A microphone can be also used by applications, including those used to detect context. Speech recognition is a huge area of research which has made large advancements recently due to the application of deep learning to the area. Speech recognition involves using machine learning methods, such as deep learning, to enable the recognition and translation of speech into text.

An area that has received less attention is the detection of context using a microphone. This involves using machine learning methods to identify the background sounds that are occurring and using this to infer context. For example, the context of “at supermarket checkout” can be identified by the sounds produced by the till as items are being scanned. Popular machine learning methods for speech recognition, such as deep learning have been applied to this area. For example, Lane et al, [23] implement a deep learning method on a smartphone in order to derive ambient sounds in the environment. Another method, [22], aims to determine the type of device that is being used based on the sound that it is emitting. The system uses the MFCCs (Mel Frequency Cepstral Coefficients) and either a Gaussian Mixture Model or a Support Vector Machine in order to classify the sounds.

3.3.1.3 Activity and context recognition

The aim of the Context Management module is to use ambient sounds to determine context. The module will use a microphone to obtain samples of the ambient sound, and then use a machine learning method to label the ambient sound. The types of ambient sounds detected will determine the prediction of a context.

Activity recognition is also an area of research that has received much attention. This is determination of a physical activity that is being performed, for example walking, on train or vacuuming. Context recognition is slightly different. Although the context may involve the activity, there will be additional information. Examples include walking in the countryside, walking a supermarket, walking in an urban setting.

3.3.1.4 Ambient sound detection

The aim is to identify the ambient sound in order to determine context. The process followed will be similar to speech recognition, as these methods have been shown to also work with ambient sound detection. The first step is to obtain sound data using a microphone. This is then segmented into frames. The Mel Frequency Cepstral Coefficients (MFCC) are often used in speech and speaker recognition.

There are two phases that occur during the ambient sound detection. The first is training, where the labelled model data will be used to construct a model of different ambient sounds that are required to be detected. Training will occur offline. This is advantageous as the training phase is usually, depending on the method, the most computationally complex phase. The constructed model will then be placed on the smartphone or other device, where the less computationally complex phase of testing can occur. Testing will sample data from the microphone at regular intervals to determine if the user is in a context that can be detected. If this is the case, the derived context will be transmitted to the Context Managers detailed in Section 3.1.1.1.

3.3.1.5 Possible issues

Using context to extend FC-reading accuracy or make the service better to the end user is a very demanding target. We have listed below the main issues which have to be solved before context can be utilized efficiently.

It is a requirement to perform machine learning on a mobile phone. There are issues for the implementation and issues for resource use; heavy on CPU and consequently the battery. Data collected from the sensors can contain a lot of noise. Accuracy varies between sensors/phone models. This can hinder the performance of any context detection mechanism.

Also, there might be a wide variety of context that needs to be captured. Context detection should not be bound to the current use cases. Implemented methods must be generic. In addition, the enablers should be interoperable with other TagItSmart enablers.

Feature ID	Description
CM1	The enabler shall provide information based on the token, scanned code (QR or NFC), location etc.
CM2	The information provided reflects the user context (location, role, type of code scanned, application used)
CM3	The information is retrieved from the backend cloud infrastructure via appropriate web services
CM4	The information received can be in the form of text, multimedia files (images, audio, video) Augmented Reality and Virtual Reality content

Table 3. Features related to context management

3.3.2 Enforcing context-based policies on FC-Scanner application

One significant challenge in the security of smartphones is to control the behaviour of applications and services (e.g. Wi-Fi, GPS, or Bluetooth). For instance, an application might access a camera of a smartphone to take a live snapshot or it can fetch the GPS data, and send this information to interested parties without the knowledge of the owner of the smartphone. Today, in several smartphone systems the behaviour of the applications is completely under the control of a centralized entity (e.g. once an application is installed, the user cannot control its behaviour). For example, Apple has complete control on the applications installed on iPhone devices. In fact, the only way to install applications onto a (non-rooted) iPhone is by downloading them from the Apple App Store. In turn, in order to appear to be published in the App Store, an application has to pass an Apple vetting procedure.

Hence, there is a need for a system that will help users to enforce user defined policies on individual applications (such as FC-scanner) as well as a system that supports users in complying with the policies specified by authorized third parties (such as TagItSmart Platform). The following examples can be scenarios for which having a practical solution might extend the usability of the FC-Scanner software:

- Alice might want to restrict the use of camera by FC-scanner when she is at home or in her office, not otherwise,

- Alice might lend her smartphone to Bob (a friend), while Alice does not want Bob to be able to use certain applications or to have access to certain data available in her phone (e.g. SMSs).
- For privacy purposes Alice might want to automatically restrict access to some data under certain conditions. For example, a same module of FC-scanner that sends her location to the TagItSmart platform when she is at retail store might be prohibited to send out her location when she is at home.
- When Alice is scanning the SmartTag on a product from her home, the FC-Scanner should not contact the TagItSmart platform as there is no such need, i.e., FC-Scanner might have restricted access to the internet interface of the smartphone.

Therefore, the behaviour of the FC-scanner should be dependent on the contextual information such as location and time. The FC-scanner software should be under the control of its user (i.e., device owner) and authenticated third party (i.e. TagItSmart platform). To this end, we will propose a design for a security system that will allow users and other predefined third parties to define context-aware policies, which can be installed, updated and applied in a fine-grained manner, i.e., on individual application level.

3.3.2.1 Overview

In this work, we will propose an architectural design of a system that can be used for enforcing context-based policies on smartphone applications such as FC-Scanner. The availability of the context such as temperature, time and location is easy to gather, thanks to the hardware advancements in the mobile devices. A context can be defined as a Boolean expression that can take as input the data reported by low level physical sensors (e.g., location, time, temperature, noise, and light). The processing of the collected data will be performed with high level software sensors (e.g., to determine whether the user is at home or retail store, by using data from the GPS), or a particular interaction with authorized third parties. In this system, we assume that the user of smartphone is honest; therefore, he/she either wants to directly set the required policies, or will comply with the policies set by the trusted third party. In particular, this system will incorporate all features at once which includes, definition and identification of fine-grained and dynamic context data, policy enforcement (i.e. make sure that the system is compliant with the behaviour described by the policy) at individual application level as well as in system-wide manner, acceptance of incoming settings at runtime which also includes the verification of the trusted third party sending management messages.

A context-related policy is composed by two different type of policies: (i) an access control policy which is composed of a set access rules, and (ii) an obligation policy which specifies actions corresponding to one or a set of control policies (i.e. start or stop an application; activate or disable a system resource, like the camera or GPS). Initially, the proposed context-based policy enforcement system will act as a security mechanism on top of the standard Android security mechanisms, latter it can be further extended for other mobile OS. As the detailed description of the proposed system is out of the scope of this deliverable, therefore, the detailed overview describing the access control model, the architecture, the components, and the main algorithms of the proposed system will be included in deliverable 3.3.

Section 4 - Enablers development environment

All three enablers shall be developed in such a way that they are completely portable to a number of different platforms such as Android, iOS, Windows and compatible with different hardware like smartphones, tablets and head mounted displays (AR and VR). The enablers are compiled as portable libraries for each platform which then can be included in other application projects and appropriate IDEs using the header files defining the SDK interfaces.

The proposal for the development environment based on Microsoft technology is described in this section which ideally can be used to leverage .NET expertise and seamless cross-platform development. However, other technologies and associated environments will additionally be used, such as Java, Eclipse and other which will be using the SDK libraries and appropriate documented APIs.

The enablers will be developed in the Visual Studio 2015 environment in C# and linked to other third-party native (C/C++) modules if required (e.g. open-source QR code scanner, SLAM module, cryptographic components etc.). For development of the scanner enabler we use Xamarin plug-in for Visual Studio for building native iOS, Android and Windows apps [29, 30].

The Xamarin platform consists of a number of elements that allow develop applications for iOS, Android and Windows:

- **C# language** – Allows to use a familiar syntax and sophisticated features like Generics, Linq and the Parallel Task Library.
- **Mono .NET framework** – Provides a cross-platform implementation of the extensive features in Microsoft's .NET framework.
- **Compiler** – Depending on the platform, produces a native app (eg. iOS) or an integrated .NET application and runtime (eg. Android). The compiler also performs optimization for the mobile deployment such as linking away un-used code.

In addition, because the underlying language is C# with the .NET framework, projects can be structured to share the code that can also be deployed in Windows Phones. The same language, APIs and data structures are used to share an average of 75% of app code across all mobile development platforms.

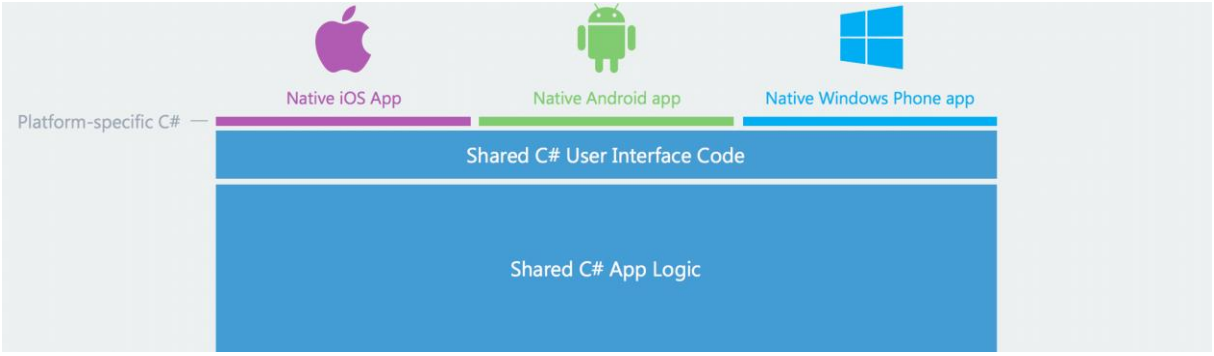

Figure 12: Single shared codebase for Android, iOS, and Windows

Development requirements (for each platform are provided below [30].

- **IOS** - Developing iOS applications requires a Mac computer, running Mac OS X. Visual Studio can also be used to write and deploy iOS applications with Xamarin's iOS Visual Studio plugin. However, a Mac is still needed for build and licensing purposes. Apple's Xcode IDE must be installed to provide the compiler and simulator for testing. To test on a real device and submit applications for distribution must join Apple's Developer Program. Each time an application is submitted or updated, it must be reviewed and approved by Apple before it is made available for customers to download. Code is written with the *Xamarin Studio* or *Visual Studio* IDE and screen layouts can be built programmatically or edited with Xamarin's iOS Designer in either IDE.
- **Android** - Android application development requires the Java and Android SDKs to be installed. These provide the compiler, emulator and other tools required for building, deployment and testing. Java, Google's Android SDK and Xamarin's tools can all be installed and run on the Windows configuration: Windows 7 & above with *Visual Studio 2012, 2013, or 2015.* Xamarin provides a unified installer that will configure system with the pre-requisite Java, Android and Xamarin tools (including a visual designer for screen layouts).

  Applications can be built and tested on a real device without any license from Google, however to distribute the application through a store (such as Google Play, Amazon or Barnes & Noble) a registration fee may be payable to the operator. Google Play will publish the app instantly, while the other stores have an approval process similar to Apple's.
- **Windows** - Windows apps (WinForms, WPF, or UWP) are built with Microsoft's *Visual Studio* IDE. They do not use Xamarin directly. However, C# code can be shared across Windows, iOS and Android.

The complete SDK enabler project can be included in other Visual Studio (Xamarin for Visual studio) application projects and then compiled combined. Alternatively, binaries and appropriate header files can be linked in within other IDEs such as Eclipse or Xcode.

Section 5 - Reference applications

Examples and associated source code of enabler usage will be located on the source control (git) in folder /sample/. The code will provide reference applications which could easily be adapted to other use-cases. Examples of reference application UI for authentication, scanner and context management are shown in figure 13.

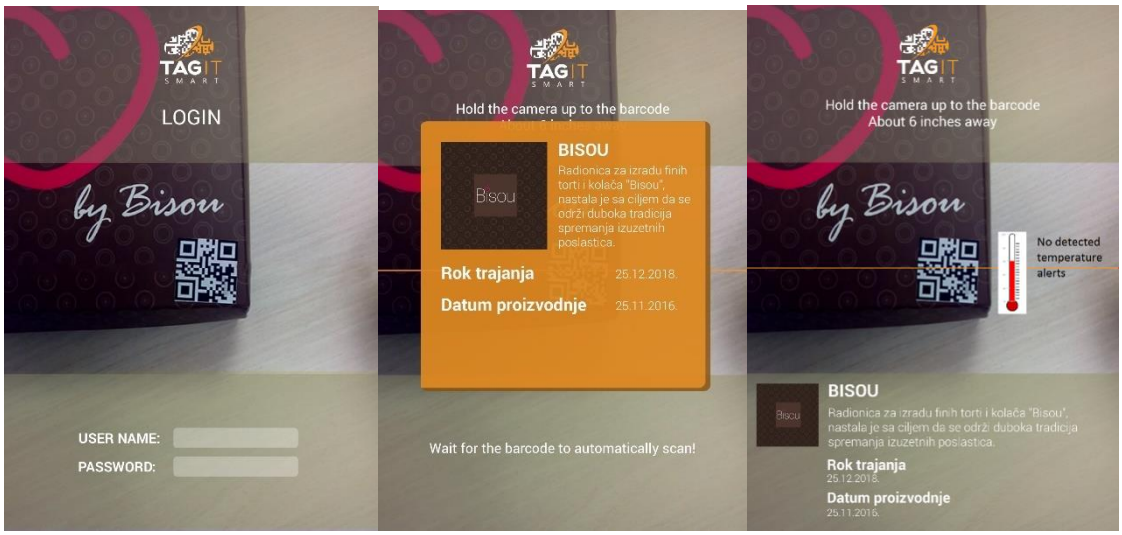

Figure 13: Example of reference application UI

Section 6 - Conclusions and future work

FC-scanner enablers are developed using an iterative approach. We have defined core functions first and implemented enablers for them. The project uses QR and DM codes and NFC tags. This makes using FC codes flexible, but at same time it makes development work for SmartTag harder. Different tags and codes have their own challenges.

At the moment (M12) development process of enablers has been started and modules become usable one-by-one. After each module is tested it can be taken into use by consortium so that WP is able to provide outcomes to other WP's, after testing, other WP provide feedback to WP3 and modules can be then changed.

Important future work applies to scanning enablers, so that project is able to start using SmartTAGs and to provide information extracted from them to TagItSmart platform. Also as this enabler is so crucial, the project is looking forward to deliver first versions of scanning enablers very soon in order to start the outcome-feedback process. Field trials in WP5 have to have properly working enablers as well as Open Call winners starting from the 2nd year of the project.

The FC-scanner can be viewed as an access control system in which once a user obtains access, it could use it to scan the information stored in SmartTags, to communicate with the TagItSmart platform, and other user-oriented services. Security of such access control systems is an important aspect where an adversary can perform various attacks during the authentication process (i.e., exploit vulnerabilities in the authentication enablers) or after the authentication process (i.e., when user is using the system for communicating with the SmartTags and TagItSmart platform, thus exploiting vulnerabilities in the scanning enablers). Due to the aforementioned reasons, a careful security analysis of the various scanning and authentication enablers presented in this deliverable is essential in order to ensure their proper usage. To this end, we will develop our next deliverable (i.e., D3.2) in M18 which includes the investigation/analysis of security aspects of the FC-scanner enablers, and design of the security solutions to countermeasure the identified security problems. In particular, the investigated security applications will prevent compromised OSs from providing fake FCs to the FC-scanner application, as well as, prevent other common attacks, such as OS initiated malicious augmented reality attacks and side-channel attacks), and envision solutions to allow FC-scanners to verify the authenticity or maliciousness of components involved in scanning process, i.e., device sensors and Operating System (OS). Furthermore, the FC-scanner uses various context-aware applications to provide enriched or differentiated services to users and to provide additional security to the system by using contextual security solutions. At the same time, the use of contextual information exposes the users to various privacy risks, thus calls for inclusion of privacy preserving mechanisms. In order to address these challenges, we will analyse and envision various solutions (in deliverable D3.3 in M24) to ensure, the user privacy-preservation, and the correct working of the context-aware management system. Privacy issues will also be addressed from the perspective of consent. With the introduction of the General Data Protection Regulation on May 25, 2018, the extent to which user based contextual information need further protection is significantly more challenging.

D3.3 will also present the final enablers for FC-scanners while the integrated FC-scanner environment will be reported in D3.4 at M30.

Section 7 - References

References of this document are listed below:

[1] V. Saravanan and R. Sindhuja, 'Iris authentication through Gabor filter using DSP processor', in 2013 IEEE Conference on Information Communication Technologies (ICT), 2013, pp. 568–571.
[2] M. Turk, 'Over Twenty Years of Eigenfaces', ACM Trans. Multimedia Comput. Commun. Appl., vol. 9, no. 1s, pp. 45:1–45:5, Oct. 2013.
[3] R. M. Guest, H. He, S. V. Stevenage, and G. J. Neil, 'An assessment of the human performance of iris identification', in 2013 IEEE International Conference on Technologies for Homeland Security (HST), 2013, pp. 623–626.
[4] J. Frank, S. Mannor, and D. Precup, 'Activity and Gait Recognition with Time-Delay Embeddings', in in Proceedings of AAAI, 2010.
[5] S. Sprager and M. B. Juric, 'An efficient HOS-based gait authentication of accelerometer data', IEEE Transactions on Information Forensics and Security, vol. PP, no. 99, pp. 1–1, 2015.
[6] C. Nickel, T. Wirtl, and C. Busch, 'Authentication of Smartphone Users Based on the Way They Walk Using k-NN Algorithm', in 2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012, pp. 16–20.
[7] T. Feng, X. Zhao, and W. Shi, 'Investigating Mobile Device Picking-up motion as a novel biometric modality', in 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS), 2013, pp. 1–6.
[8] T.-Y. Chang, C.-J. Tsai, and J.-H. Lin, 'A graphical-based password keystroke dynamic authentication system for touch screen handheld mobile devices', Journal of Systems and Software, vol. 85, no. 5, pp. 1157–1165, May 2012.
[9] S. P. Banerjee and D. Woodard, 'Biometric Authentication and Identification Using Keystroke Dynamics: A Survey', Journal of Pattern Recognition Research, vol. 7, no. 1, pp. 116–139, 2012.
[10] F. Zhang, A. Kondoro, and S. Muftic, 'Location-Based Authentication and Authorization Using Smart Phones', in 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2012, pp. 1285–1292.
[11] M. Conti, I. Zachia-Zlatea, and B. Crispo, 'Mind How You Answer Me!: Transparently Authenticating the User of a Smartphone when Answering or Placing a Call', in Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, New York, NY, USA, 2011, pp. 249–259.
[12] T. Feng, X. Zhao, and W. Shi, 'Investigating Mobile Device Picking-up motion as a novel biometric modality', in 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS), 2013, pp. 1–6.
[13] A. Jain and V. Kanhangad, 'Exploring orientation and accelerometer sensor data for personal authentication in smartphones using touchscreen gestures', Pattern Recognition Letters, vol. 68, Part 2, pp. 351–360, Dec. 2015.
[14] J.-S. Wu, C.-T. Lin, W.-C. Lin, and T.-E. Wei, 'Smartphone continuous authentication based on keystroke and gesture profiling', in 2015 International Carnahan Conference on Security Technology (ICCST), 2015, pp. 191–197.
[15] C. Bo, L. Zhang, T. Jung, J. Han, X. Y. Li, and Y. Wang, 'Continuous user identification via touch and movement behavioural biometrics', in Performance Computing and Communications Conference (IPCCC), 2014 IEEE International, 2014, pp. 1–8.
[16] M. Qiao, S. Zhang, A. H. Sung, and Q. Liu, 'A Novel Touchscreen-Based Authentication Scheme Using Static and Dynamic Hand Biometrics', in Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual, 2015, vol. 2, pp. 494–503.

[17] F. Zhang, A. Kondoro, and S. Muftic, ‘Location-Based Authentication and Authorization Using Smart Phones’, in 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2012, pp. 1285–1292.
[18] Montjoye, Yves-Alexandre, et al. "Unique in the crowd: The privacy bounds of human mobility." Scientific reports 3 (2013).
[18] H. Witte, C. Rathgeb, and C. Busch, ‘Context-Aware Mobile Biometric Authentication based on Support Vector Machines’, in 2013 Fourth International Conference on Emerging Security Technologies (EST), 2013, pp. 29–32.
[19] Lehtonen, Mikko, Florian Michahelles, and Elgar Fleisch. "Probabilistic approach for location-based authentication." 1st International workshop on security for spontaneous interaction IWSSI. Vol. 2007. 2007.
[20] Albayram, Yusuf, et al. "A location-based authentication system leveraging smartphones." 2014 IEEE 15th International Conference on Mobile Data Management. Vol. 1. IEEE, 2014.
[21] Fridman, Lex, et al. "Active authentication on mobile devices via stylometry, application usage, web browsing, and GPS location." (2015).
[22]Zoha, Ahmed, et al. "Acoustic and device feature fusion for load recognition." Novel Applications of Intelligent Systems. Springer International Publishing, 2016. 287-300.
[23] Lane, Nicholas D., PetkoGeorgiev, and Lorena Qendro. "DeepEar: robust smartphone audio sensing in unconstrained acoustic environments using deep learning." Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, 2015.
[25]http://www.thinfilm.no/wp-content/uploads/2014/01/Thin-Film-Electronics-ASA-Kovio-NFC-Barcode-Data-Format-128bit-NFC-tag.pdf
[26] https://developer.android.com/reference/android/nfc/tech/NfcBarcode.html
[27] https://github.com/Redth/ZXing.Net.Mobile
[28] V. Saravanan and R. Sindhuja, ‘Iris authentication through Gabor filter using DSP processor’, in 2013 IEEE Conference on Information Communication Technologies (ICT), 2013, pp. 568–571.
[29] https://www.visualstudio.com/xamarin/
[30]https://developer.xamarin.com/guides/crossplatform/application_fundamentals/building_cross_platform_applications/
[31] Burnett, M.: 10,000 top passwords, http://xato:net/passwords/more-topworst-passwords/.
[32] Aviv, A.J., Gibson, K., Mossop, E., Blaze, M., Smith, J.M.: Smudge attacks on smartphone touch screens. In: Proc. of the 4th USENIX Conf. on O ensive Technologies. pp. 1{7 (2010).
[33] Tari, F., Ozok, A.A., Holden, S.H.: A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. In: Proc. of the Sec- ondSymp. on Usable Privacy and Security. pp. 56{66 (2006).
[34] Xu, Y., Heinly, J., White, A.M., Monrose, F., Frahm, J.M.: Seeing double: Reconstructing obscured typed input from repeated compromising re ections. In: Proc. of the 2013 ACM Conf. on Computer and Communications Security. pp. 1063{1074 (2013).
[35] Clarke, N.L., Furnell, S.M.: Authentication of users on mobile telephones,A survey of attitudes and practices. Computers & Security 24(7), 519 527 (2005)
[36] Mantyjarvi, J., Lindholm, M., Vildjiounaite, E., Makela, S.M., Ailisto, H.: Identifying users of portable devices from gait pattern with accelerometers. In: Proc. of the Int'l Conf. on Acoustics, Speech, and Signal Processing. pp. 973{976 (2005)
[37] Okumura, F., Kubota, A., Hatori, Y., Matsuo, K., Hashimoto, M., Koike, A.: A study on biometric authentication based on arm sweep action with acceleration sensor. In: Proc. of the Int'l Symp. on Intelligent Signal Processing and Communications. pp. 219{222 (2006)

[38] Conti, M., Zachia-Zlatea, I., Crispo, B.: Mind how you answer me!: Transparently authenticating the user of a smartphone when answering or placing a call. In: Proc. of the Sixth ACM Symp. on Information, Computer and Communications Security. pp. 249{259 (2011)
[39] Frank, M., Biedert, R., Ma, E., Martinovic, I., Song, D.: Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. IEEE Trans. Inf. Forensics and Security 8(1), 136{148 (2013)
[40] Tasi, C.J., Chang, T.Y., Cheng, P.C., Lin, J.H.: Two novel biometric features in keystroke dynamics authentication systems for touch screen devices. Security and Communication Networks (2013)
[41]D. Shukla, R. Kumar, A. Serwadda, and V. V. Phoha. Beware, your hands reveal your secrets! In Proceedings of ACM CCS, 2014.
[42]F. Schaub, R. Deyhle, and M. Weber. Password entry usability and shoulder surfing susceptibility on different smartphone platforms. In Proceedings of ACM MUM, 2012.
[43]J. L. Wayman. Error rate equations for the general biometric system. Robotics & Automation Magazine, 6(1):35{48, 1999.
[44]M. Frank, R. Biedert, E.-D. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. IEEE TIFS, 8(1):136, 2013.
[45]S.-s. Hwang, S. Cho, and S. Park. Keystroke dynamics-based authentication for mobile devices. Computers & Security, 28(1):85{93, 2009.
[46] C. Bo, L. Zhang, T. Jung, J. Han, X. Y. Li and Y. Wang, "Continuous user identification via touch and movement behavioral biometrics," 2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC), Austin, TX, 2014, pp. 1-8.
[47] Buriro, Attaullah, Bruno Crispo, Filippo Del Frari, Jeffrey Klardie, and Konrad Wrona. "Itsme: Multi-modal and unobtrusive behavioural user authentication for smartphones." In International Conference on Passwords, pp. 45-61. Springer International Publishing, 2015.
[48]Sitová, Zdeňka, Jaroslav Šeděnka, Qing Yang, Ge Peng, Gang Zhou, Paolo Gasti, and Kiran S. Balagani. "HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users." IEEE Transactions on Information Forensics and Security 11, no. 5 (2016): 877-892.
[49]http://www.intechopen.com/books/radio-frequency-identification-from-system-to-applications/rfid-under-water-technical-issues-and-application
[50] M. Turk, 'Over Twenty Years of Eigenfaces', ACM Trans. Multimedia Comput. Commun. Appl., vol. 9, no. 1s, pp. 45:1–45:5, Oct. 2013.
[51] R. M. Guest, H. He, S. V. Stevenage, and G. J. Neil, 'An assessment of the human performance of iris identification', in 2013 IEEE International Conference on Technologies for Homeland Security (HST), 2013, pp. 623–626.
[52] J. Frank, S. Mannor, and D. Precup, 'Activity and Gait Recognition with Time-Delay Embeddings', in in Proceedings of AAAI, 2010.
[53] S. Sprager and M. B. Juric, 'An efficient HOS-based gait authentication of accelerometer data', IEEE Transactions on Information Forensics and Security, vol. PP, no. 99, pp. 1–1, 2015.
[54] C. Nickel, T. Wirtl, and C. Busch, 'Authentication of Smartphone Users Based on the Way They Walk Using k-NN Algorithm', in 2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012, pp. 16–20.
[55] T. Feng, X. Zhao, and W. Shi, 'Investigating Mobile Device Picking-up motion as a novel biometric modality', in 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS), 2013, pp. 1–6.

[56] T.-Y. Chang, C.-J. Tsai, and J.-H. Lin, ‘A graphical-based password keystroke dynamic authentication system for touch screen handheld mobile devices’, Journal of Systems and Software, vol. 85, no. 5, pp. 1157–1165, May 2012.
[57] S. P. Banerjee and D. Woodard, ‘Biometric Authentication and Identification Using Keystroke Dynamics: A Survey’, Journal of Pattern Recognition Research, vol. 7, no. 1, pp. 116–139, 2012.
[58] http://www.iot-icore.eu/attachments/article/89/20141031_final_architecture.pdf
[59] http://iot-cosmos.eu/content/d232-conceptual-model-and-reference-architecture-updated
[60] https://bscw.ict-rerum.eu/pub/bscw.cgi/d31979/RERUM%20deliverable%20D2_5.pdf
[61] Amit Grover, Paul Braeckel, Kevin Lindgren, Hal Berghel, and Dennis Cobb, “Parameters Effecting 2D Barcode Scanning Reliability”, Advances in Computers, Vol. 80, Burlington: Academic Press, 2010, pp. 209-235. ISBN: 978-0-12-381025-0, 2010 Elsevier Inc. Academic Press