BehavAuth

Human behaviour based authentication for Internet of Things

Open Call Project for TagItSmart: Grant agreement 688061

WP1: D1.1 Initial system design based on the use cases and requirement analysis

Deliverable ID:	**D 1.1**
Deliverable Title:	**Initial system design based on the use cases and requirement analysis**
Revision #:	**1.0**
Dissemination Level:	**Public**
Responsible beneficiary:	QBE
Contributing beneficiaries:	All
Contractual date of delivery:	31/10/2017
Actual submission date:	10/11/2017

Start Date of the Project: 1 September 2017 Duration: 6 Months

Contents

Section 1 - State-of-the-art ***6***

1.1 SingleID **6**

1.2 Auth0 ..... **7**

1.3 Google Identity platform **8**

1.4 IdentityX platform **9**

1.5 CallSign **11**

1.6 Twillio **11**

1.7 TrustedX **13**

1.8 Entrust **14**

Section 2 - Use cases and requirements ***17***

2.1 Use cases **17**

2.1.1 Digital product 17

2.1.2 Lifecycle management 18

2.1.3 Brand protection 18

2.1.4 Dynamic pricing 19

2.1.5 Home services 20

2.2 Engaging stakeholders **21**

2.2.1 Engaging psychologists 21

2.2.2 Engaging potential early adopters 23

2.2.3 End-users 24

2.2.4 Engaging privacy experts 24

2.3 Risk analysis **25**

2.4 Requirement analysis **28**

Section 3 - Reusability analysis ***29***

3.1 Web application frameworks **29**

3.2 Machine-learning frameworks **31**

3.3 Data collection frameworks **33**

Section 4 - Initial design ***35***

4.1 Overall design of the system **35**

4.2 BehavAuth platform **36**

4.3 BehavAuth mobile app/library **36**

Section 5 - Conclusion ***38***

Section 6 - References ***39***

Document history

Revision	Date	Modification	Authors
0.1	30/10/2017	Draft ready for review	N. Palaghias
0.2	31/10/2017	Internal review	N. Loumis
1.0	31/10/2017	Corrections after internal review	N. Palaghias

Introduction

The main objective of this document is to provide a detailed description of the initial system design based on the use cases and the derived requirements that were extracted from the collaboration with the stakeholders (SMEs) that are interested in the BehavAuth and TagItSmart.

Section 1 provides an introduction to the deliverable about the content described, as well as, an analysis of the state-of-the-art platforms that exist in the current authentication market. The analysis of the state-of-the-art platform has as sole purpose to understand the existing technologies in the area of authentication in order to be able to build upon them and create a new innovative solution for this area. The state-of-the-art analysis will provide an overview of the existing architectures that the state-of-the-art systems used to offer compatibility with third-party platforms, as well as to understand which biometrics and behavioural cues (if any) were used to authenticate users.

In **Section 2**, use case and requirement analysis is performed to derive the functional requirements of the BehavAuth project. The section initiates with the description of the TagItSmart use cases, where the BehavAuth tool is applicable and could be utilised. Following, a requirement analysis is performed by engaging various stakeholders such as psychologists, end-users, potential early adopters of the tool and privacy experts. BehavAuth collaborators, as suggested also in the initial proposal, engaged the various stakeholders from the beginning of the project to shape the requirements of the tool given the needs of the involved stakeholders. Following, a risk analysis is performed to derive the potential risks of the project and the platform. Having collected the uses cases and the requirements of the various stakeholders (including the risks), the functional requirements of the BehavAuth are produced.

Next, **Section 3** a reusability analysis is conducted to understand what existing technologies are available and could be utilised to speed up the development process of the BehavAuth tool. Initially, the state-of-the-art web application development frameworks are analysed, to understand their capabilities and limitations with respect to the BehavAuth project. Then, given the nature of the BehavAuth project, machine-learning is a significant part of the tool, an analysis is performed on the state-of-the-art machine-learning frameworks. Then, for the data collection part of the project, existing solutions are analysed that provide an abstraction in the data collection process including various enabling functionalities for the adaptation, energy efficiency as well as the privacy of the users.

Given the use case and requirement analysis and the reusability analysis detailed in the previous sections, **Section 4** provides an initial design of the BehavAuth tool and the overall architecture of the TagItSmart and BehavAuth ecosystem. Initially, an overview of the overall design of the TagItSmart and BehavAuth system is described, including the workflow. Next, the BehavAuth platform is detailed, including an overall description of the main functionalities of the platform. Following, the BehavAuth mobile app/library that will perform the data collection is described including its functionalities.

Finally, this deliverable ends with the conclusion provided in **Section 5**.

Section 1 - State-of-the-art

This section details the state-of-the-art solutions that offer authentication to third-party platforms. Each solution is being described regarding the functionalities provided and a following paragraph analyses the actual platform indicating its gaps.

1.1 SingleID

SingleID [1] is a platform that allows users to sign-in to third-party platforms through a single id. This is performed by using a mobile app where all the information of the user is stored (See Figure 1). Then the user accesses the third-party platform that is integrated with the SingleID platform. Next to the registration form of the third-party platform, a sign-in button from SingleID is placed and by clicking on this button and inserting a verification code, provided by the mobile app, the user creates automatically an account at the third-party platform. Figure 1 shows the information flow among the different components of the SingleID system.

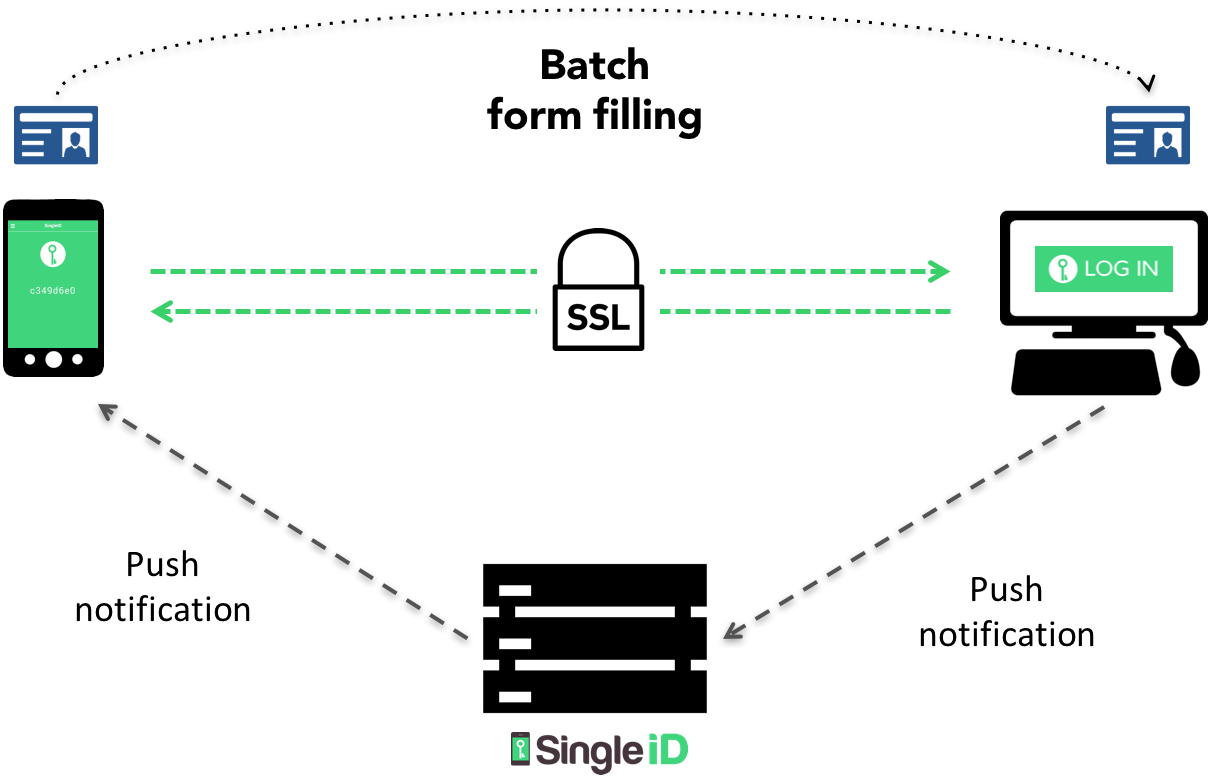

Figure 1. Information flow of the SingleID system.

Gaps. SingleID focuses on providing a single point where users can sign-in on different services. Once the user has authenticated, the data are shared with the third-party platform. It is not fully clear however, how secure the mobile app- which store the credentials and information of the user - is. In case the device is compromised, e.g. the passcode has fallen in

the hands of a malicious user, then the malicious user could authenticate on various platforms. There is no tool that continuously authenticates the user with the mobile device.

1.2 Auth0

Auth0 [2] is a service that abstracts how users authenticate to different platforms and services by providing a unified authentication service that operates as a bridge to multiple platforms and services, as shown in Figure 2. Auth0 provides an authorization API that allows third-party developers to manage the authorization requirements for server-to-server and client-to-server application through the OAuth0 2.0 protocol. Third-party developers do not need to take into consideration about the OAuth 2.0/OpenID Connect specification, or the many other technical aspects of API authorization.

Any application can be connected (having different programming language or application stack) to Auth0 while creating the connection with the actual platform allowing the usage of authentication method such as:

- Custom credentials such as username and password
- Social network based authentication through Google, Facebook, Twitter as well as compatibility with OAuth1/2 and OpenID.
- Enterprise directories such as LDAP, Google Apps, Office365, ADFS, AD, SAML-P, WS-Federation etc.
- Password-less such as Touch ID, one time codes on SMS or emails.

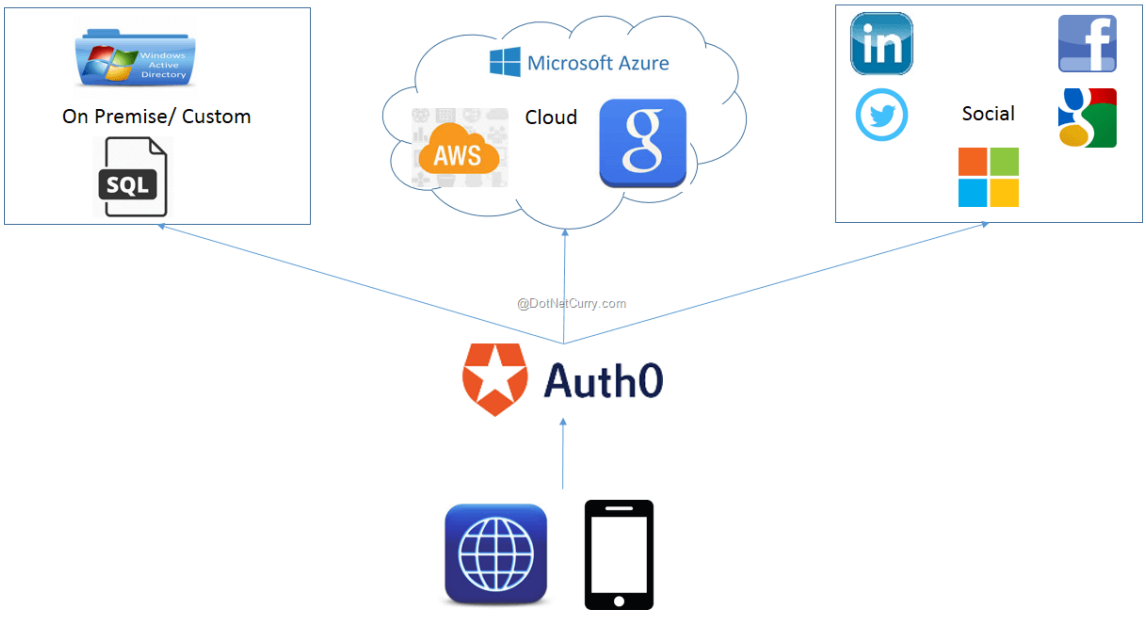

Figure 2. Auth0 information flow

Gaps. Auth0 provides multiple authentication points for users to authenticate to a particular service. The main benefit for third-party platforms and services is the ability to provide multiple authentication mechanisms to the users. However, the authentication mechanisms are mainly based on username/password sets, which leads Auth0 to inherit the native vulnerabilities of

the username/password based mechanism. From the biometrics' point of view, despite that Auth0 is compatible with Touch ID, it does not support any other types of biometrics or behaviour-based authentication.

1.3 Google Identity platform

Google Identity platform [3] consists of a set of tools that enable the users to authenticate to various platforms as shown in Figure 3.

- **Google Sign-In** is a secure authentication system that allows users to sign in to a third-party service with their Google account. It operates also as a gateway to Google's users together with services. Developers are able to provide the users the ability to perform payments with Android Pay, share information with their Google contact, and store a file, or event.

- **Smart Lock for Passwords** is a seamless sign in and sign up into apps service, on Android and websites in Chrome via any account. Through this tool, users get automatically signed-in across devices. This allows the users to reduce the risk of forgetting their authentication credentials which may lead to users' frustration and further to customer churn. Smart Lock constitutes a password storage tool where users do not need to re-enter the username and password throughout all their Chrome and Android devices.

- **FireBase Authentication** is an authentication mechanism for sign-in and on-boarding for users on all their devices. Firebase contains backend services that allow the authentication of users with a Software Development Kit (SDK). It provides the ability to authenticate users through passwords and federated identity provider credentials. User interface (UI) Libraries are also implemented to facilitate the authentication process at third-party apps.

- **SMS Retriever API** is a mechanism that allows users verification through SMS. Verification codes are sent through SMS and can be retrieved automatically by the third-party Android, without the need of the user to copy and paste these codes. It should be noted that permissions for full request of SMSs are not required.

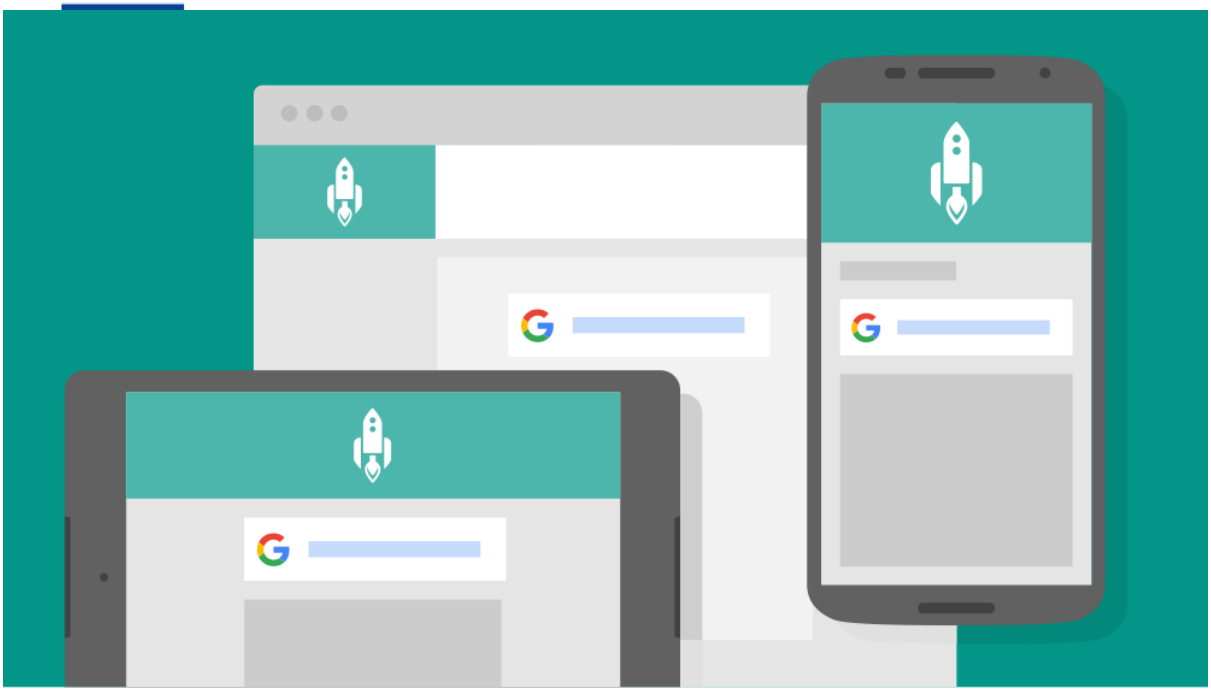

Figure 3. Google Identity platform

Gaps. Google Identity platform provides a set of tools to authenticate mainly using Google accounts through different types of devices. The authentication process is based on the old-fashioned username/password credentials that users need to input to a form. In this way, users are able to authenticate to third-party services using a single identity. A two-factor authentication process is provided by the SMS Retriever API, but apart from that, no biometrics or behavioural pattern recognition process has been implemented. Meaning that, in case the Google credentials have been compromised, malicious users could get access not only to the Google services, but also to all third-party services that have integrated the Google Identity platform. Google implements also some "suspicious" logins detection mechanism from previously unknown devices and alerts users through an email. However, since the malicious user will also have access to the Google email, which implements the same service, it is highly possible that the rightful owner of the account will never get notified.

1.4 IdentityX platform

IdentityX [5] platform constitutes an authentication platform that uses three biometrics retrieved from a smartphone device in order to authenticate a user to a third-party platform. According to the company, the training process has a duration of 30 seconds. The data are sent to an authentication platform that communicates and connects the third-party's mobile app e.g. banking mobile app, and then authenticates the user (See Figure 4). The user then is able to start using the mobile app of the third-party without needing to perform any authentication process through username/password. IdentityX platform does not require from the user to remember any credentials, such as username/password, in order to login to a service. Thus, it discards the users' pain in case they forget their credentials, or they do spelling mistakes when providing the credentials.

The IdentityX platform performs authentication by using the three following biometrics

- Take a selfie, where the system performs face recognition,
- Speak a phrase, where the system performs voice recognition,
- Touch ID, where the system uses the fingerprint biometric to identify the user.

Depending on the availability of the aforementioned biometrics on a particular device, the user decides which biometric to use in order to perform authentication.

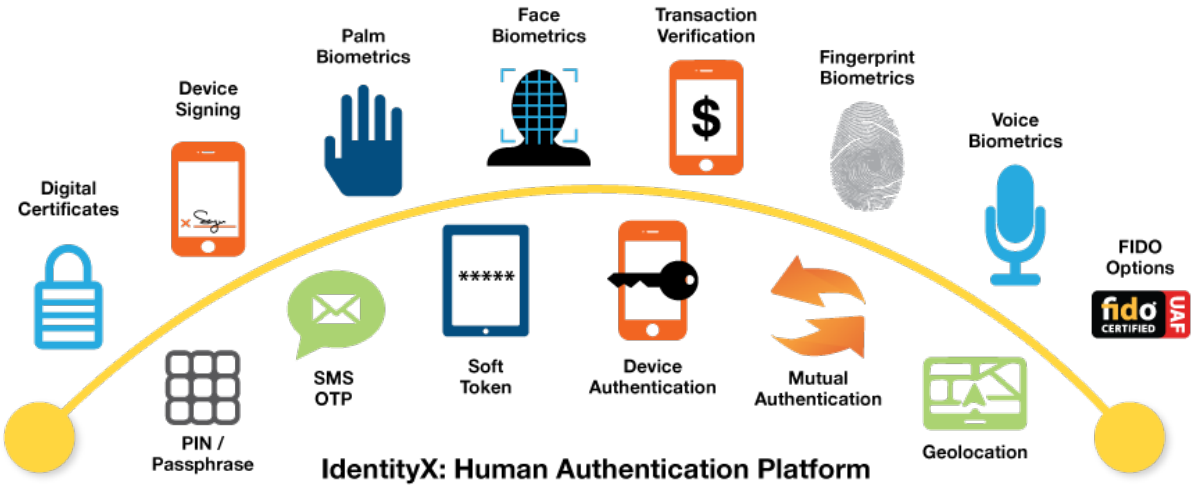

Figure 4. IdentityX components for authentication

Gaps. IdentityX platform provides three biometrics to choose from when trying to sign-in to a particular service. Each time the user is able to choose one of the three biometrics, depending on the convenience and the availability of the biometric, every time they try to sign-in into a particular service or platform. The training process of the platform takes about 30 seconds according to the company where a user takes a selfie, speaks a passphrase and then adds a fingerprint. This means that for each biometric one training sample is only taken. In case the company's recognition algorithms aren't very robust, the authentication process will fail quite often as machine-learning techniques require usually larger number of training samples. For example, for the image recognition, their approach needs to tackle the angle, ambient light and other attributes of the photo. For the speaker recognition, the machine-learning model needs to take into consideration the ambient noise, which can occur in different context. For the fingerprint, the method needs to take into consideration that users may try to use other fingers than the one used for training and in different angles. In overall, the system uses one biometric for authentication and provides the ability to the user to choose which one. Individual biometrics have shown that have their own vulnerabilities and weakness [4], thus there is a need to provide a more advanced mechanism that removes the sources of error for each individual biometric.

1.5 CallSign

CallSign [5] provides a solution called **Intelligence Driven Authentication** (**IDA**) that predicts data breaches and cybercrime and tries to proactively defend the systems against them. The main goal is to secure the identities of employees by preventing data breaches and protecting consumer identities by minimising the impact of the vulnerable username/password based authentication. The system operates by leveraging artificial intelligence to understand users' patters and based on that, it adapts the authentication journeys (See Figure 5). The focus of IDA is to provide security to third-party platforms and their users considering the trade-off of users' experience. The technology is based on deep learning that uses device's location and behaviour analytics to provide a secure authentication mechanism. As shown in the following figure, different biometrics such as face, swipe, voice, and iris recognition are used on the mobile app end-point to authenticate the users.

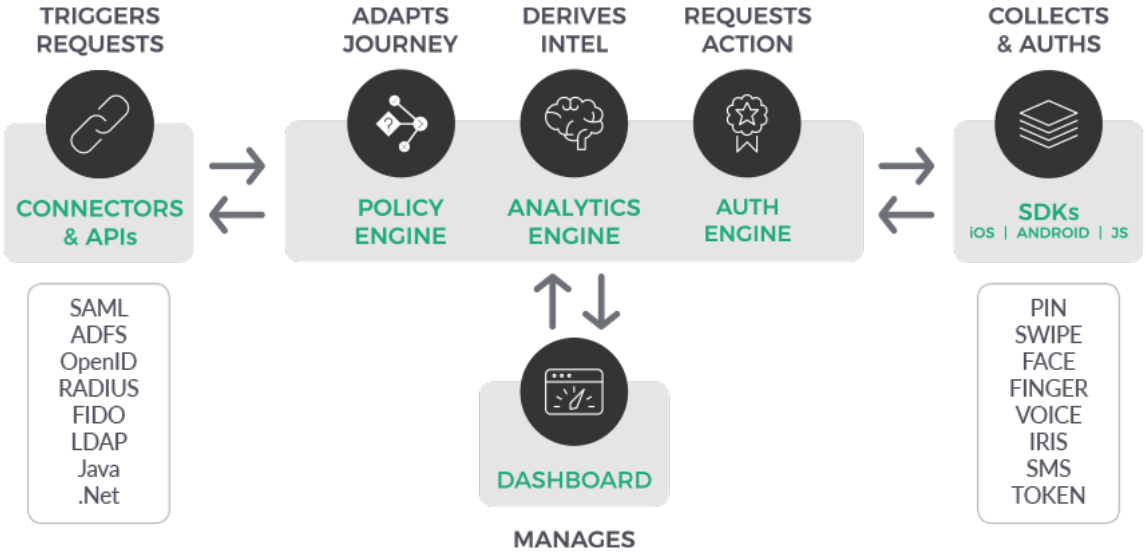

Figure 5. IDA system architecture.

Gaps. CallSign provides a holistic authentication platform that uses biometrics, passcodes and authentication codes to verify the identity of the users. An intelligent mechanism uses various analytics to predict unauthorized access and adapts the authentication journey. It should be noted that this system uses individual biometrics to authenticate users, which minimises the security of the system. Depending on the predefined policy, different authentication method is requested by the end-user, however, multiple authentication biometrics are **not** combined. Also, the platform does not use any social or human behaviour mechanism to continuously authenticate the users, which could improve system's security and user experience.

1.6 Twillio

Twillio [7] provides a tool for third-party platforms to implement two-factor authentication that does not rely only on username/password credentials to login. Two-factor authentication constitutes a second layer of security upon the username/password credentials that are often used at on-line platforms to verify users. The use of other devices in parallel to authenticate a

user at a particular service provides multiple factors off authentication, which strengthens the security of the system. Twillio uses a two-factor authentication, where a user tries to login with username/password and following, a second layer of authentication is added where a verification code is sent to the users' mobile phone. The implementation of the verification sent/validation differs and is explained below.

The two-factor authentication is implemented through the mobile phone:

- In case the user has downloaded the related app:
  - A push notification is sent to the users, informing them about the various login attempts. In this case, users are able to review the login attempts and prevent any unwanted or unauthorized access to the third-party platform (See Figure 6).

  - A time-based one-time password (TOTP) is provided to the users through the mobile app assuming internet connectivity. After a certain time period, the password, which can be used only once, expires.

- In case the user has not downloaded the mobile app, an SMS is sent to the user including the verification code.

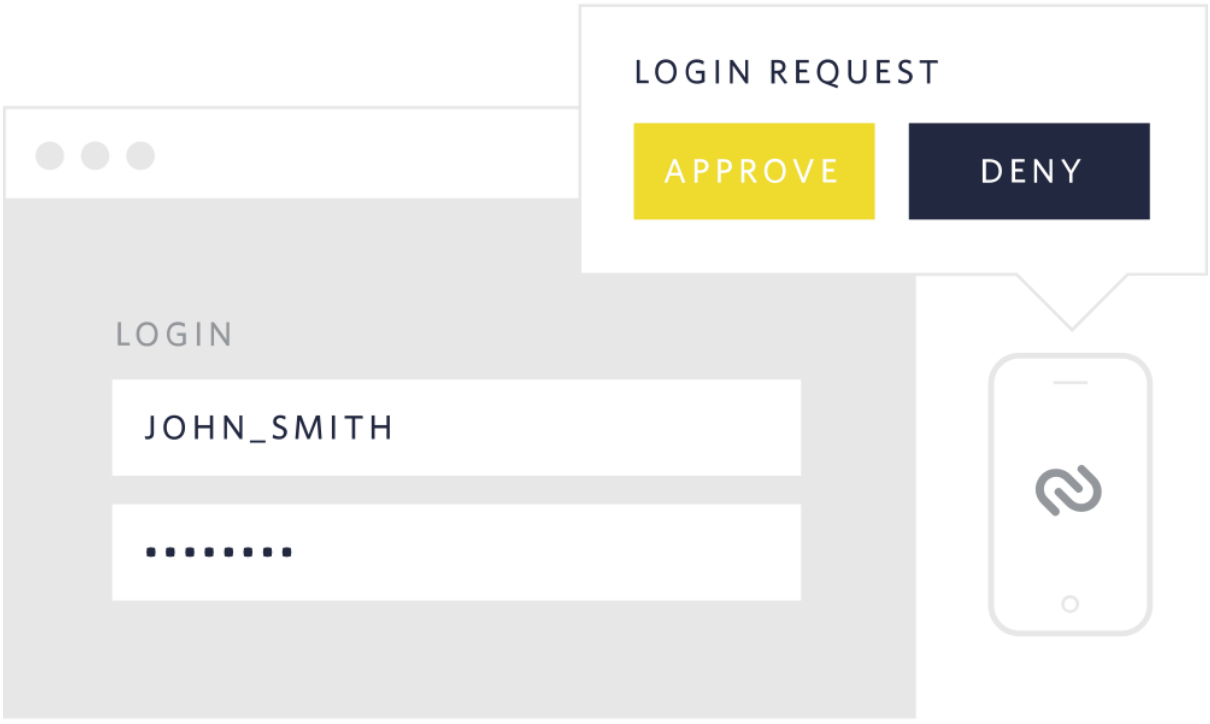

Figure 6. Twillio two-factor authentication through push notifications.

Gaps. Twillio provides a tool for third-party platforms and services to implement two-factor authentication in order to enhance the security of their authentication systems by adding a second layer of identity validation by incorporating a second device to the authentication process. The main weak point of two-factor authentication is that it is prone to spoofing, man-in-the-middle and Trojan attacks [10]. Verification codes are sent either through mobile carriers

or through internet connection, which are both vulnerable and could be hijacked through man-in-the-middle attack. The National Institute of Standards and Technology has withdrawn support for SMS-based two-factor authentication in August 2017, as they indicated the risk of interception or spoofing. In addition, the username/password based authentication that is a core component of this method, have shown its vulnerabilities and disadvantages, which is also inherited by the two-factor authentication.

1.7 TrustedX

TrustedX [7] authentication platforms is an adaptive authentication and federation platform for web and cloud environments. It operates as a middle tool between the user application and the identity services (See Figure 7). The platform supports corporate, social network and government authentication mechanisms. The predefined trust policies for the users of the platform can be modified in order to add another authentication factor for certain situations. The platform uses certain context information, such as keystroke patterns, to improve the user experience of the authentication process. A single sign-on system has been implemented and integrated with a reporting system to easily detect security breaches. Following, some additional key-features of the platform are described.

- **Identity provider**. The platform is able to provide identity management to third-party applications as well as sign-in functionalities with external providers that have already established a databased of users. Also, the system supports login with corporate directories, national eIDs and social networks.

- **Layered security**. Risk analysis is performed depending on the user profile that tries to authenticate using habits and biometrics. An additional authentication step is added when the risk analysis exceeds a certain level of risk.

- **Cloud applications**. Cloud platforms such as Google apps, Salesforce and Office365 are integrated with the platform through standard web and cloud protocols (SAML 2.0 and OAuth 2.0/OpenID)

- **Centralized control and auditing**. The different authentication factors can be modified and configured depending on the user group such as employees, visitors

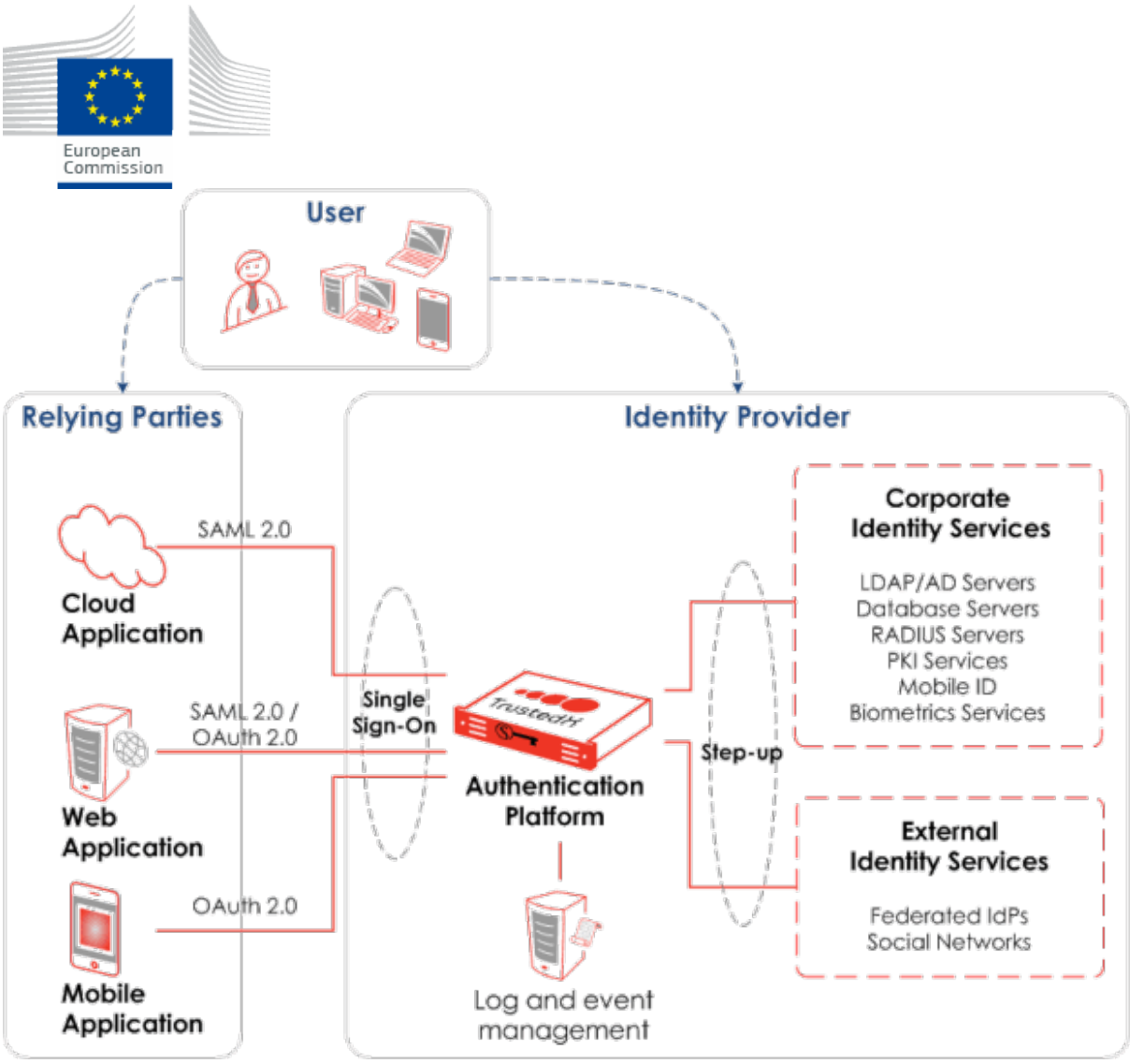

Figure 7. TrustedX system architecture.

Gaps. TrustedX is an adaptive authentication platform that adapts the authentication process based on the risk analysis performed for the levels of trust among the users. An anomaly detection mechanism is also implemented for the different authentication factors that are included in the system. The old-fashioned username-password authentication process (including the social network-based and other corporate sign-on mechanisms) introduce various aforementioned vulnerabilities and weakens the security of the system. In case the credentials are compromised, including the external or social network based, then the authentication platform will also be compromised. Individual biometrics are used to improve the user experience but there is no holistic approach to combine multiple biometrics and learn the behavioural patterns of the users, which would improve the security of the authentication system and the user experience.

1.8 Entrust

Entrust [8] provides a platform for intelligent and adaptive authentication service to enable third-parties to authenticate their users (See Figure 8).

- **Device reputation**. The authentication of the user is done explicitly with fingerprint biometric. The system uses various analytics and learning mechanisms to detect

fraud at the device level. This includes learning the users' transaction patterns to detect fraud, as well as, the existence of proxies, mobile VPNs and TOR networks.

- **Locational and velocity analytics**. The system considers if users have previously authenticated in a certain location, if the user is expected to authenticate in a particular location and if the IP used to connect to the internet is blacklisted.

- **Predictive intelligence**. The system uses deep learning to predict the actions (in terms of transactions) of the users and in that sense, expects these certain actions by the users. In case the detected actions of the users, depending on the group policy, are different than the expected actions, a red flag is raised.

- **Behavioural analytics**. The system uses data from the application usage such as location and time to detect the behaviour of the user with respect to application usage.

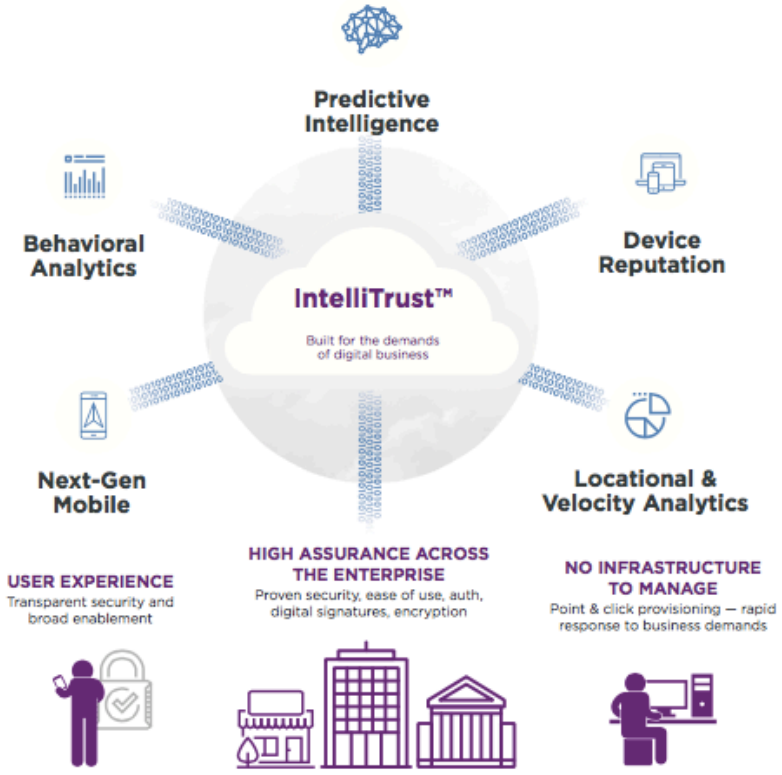

Figure 8. Entrust functionalities diagram.

Gaps. Entrust provides an adaptive authentication platform that tries to secure third-party systems through learning the application usage patterns such as expected time, expected locations, and common transactions. The provided fraud detection mechanism has a significant value in terms of security as it detects proxy networks, TOR networks and mobile VPNs and prevents attacks such as man-in-the-middle attack and spoofing information from the mobile app. This is included in the Device Reputation concept; however, it is limited to the time and location of the user trying to authenticate with the device. Two-factor authentication is also provided by the platform; however, this approach has various vulnerabilities such as

man-in-the-middle attack and Trojan attack that have previously been identified [10]. Further, for the actual authentication of the user, the old-fashioned username/password approach is used as well as a biometric fingerprint. These are quite limiting authentication methods in order to verify the user; both are not secure enough to verify the users.

Section 2 - Use cases and requirements

This section provides a detailed description of the use cases defined by the TagItSmart and BehavAuth project, through the co-creation workshop. The use cases were leveraged in order to derive the appropriate requirements through also the engagement of the various stakeholders.

2.1 Use cases

The BehavAuth collaborator concluded on these use cases based on the engagement with stakeholders including the TagItSmart consortium. The images and information presented below have been taken from TagItSmart website [11] and deliverables.

2.1.1 Digital product

This use case is focused on using fast-moving consumer goods and combining them with the so-called smart-tags and the TagItSmart in order to track the product during the whole lifecycle. In this use case the product is being tracked from factory and the road to the end customer. The smart-tags are placed on the goods and are capable of tracking the environmental conditions in which the goods are exposed when they leave the factory and until they reach the end-customer. The customers are able to scan the smart-tag and identify if the goods have been exposed to certain conditions, for example above the maximum allowed temperature (See Figure 9).

Figure 9. Digital product use case.

BehavAuth contribution. BehavAuth will provide an authentication mechanism that will enable the end customer to authenticate through the TagItSmart mobile app to the TagItSmart platform. In this way, the user will be able to login to the TagItSmart mobile app and then scan the smart-tag that is placed on the products. BehavAuth will provide a secure and user-friendly authentication mechanism that will enable users to securely login to the mobile app and check the environmental conditions in which the product have been exposed during the journey from the factory to the shelf of the store.

2.1.2 Lifecycle management

This use case is focused on the managing the lifecycle of every fast-moving consumer goods or consumer packaged goods (CPG). The goal is to assist and empower companies and citizens in order to recycle their waste. TagItSmart will create a new ecosystem where the users prompted to recycle by providing to the customers information about how to recycle the product including where to recycle the product and the local regulations about the recycling process. Consumers will be able to scan the product and receive information about the recyclability of the product through the TagItSmart mobile app. A more detailed description of the process is described in the Figure 10 below.

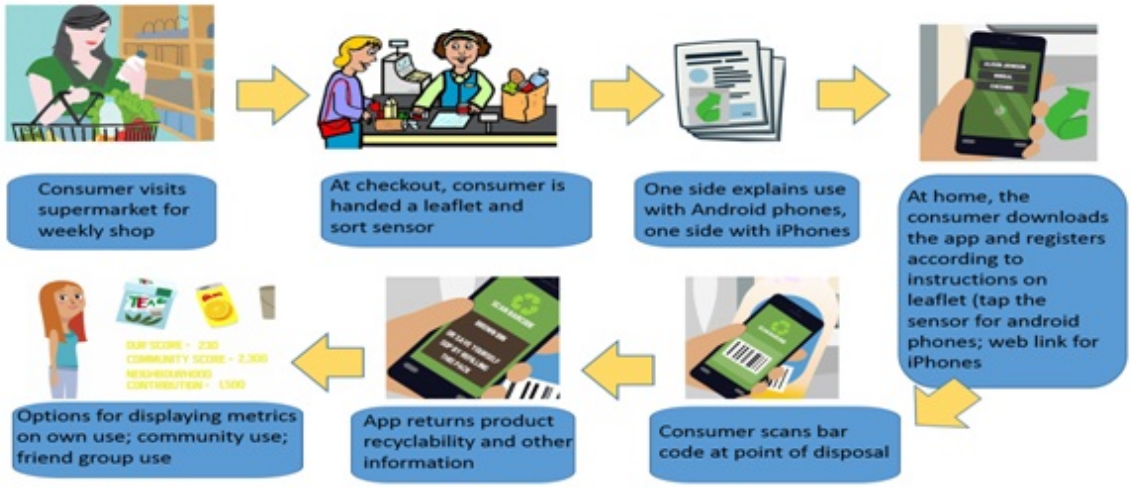

Figure 10. Lifecycle management use case.

BehavAuth contribution. BehavAuth will provide an authentication mechanism that will enable the end customer to authenticate through the TagItSmart mobile app to the TagItSmart platform. In this way, the user will be able to login to the TagItSmart mobile app and then scan the bar code in order to understand the recyclability of the product including additional information. BehavAuth will provide a secure and user-friendly authentication mechanism that will enable users to securely login to the mobile app and check the recyclability of the product as well as to get information on how to recycle the product based on contextual information.

2.1.3 Brand protection

This use case is focused on leveraging the smart-tags and the TagItSmart platform in order to product the brand of the products. The smart-tags initially will monitor the environmental conditions in which the product will be exposed such as ambient light and ambient temperature. Also the smart-tag will constitute a unique identification mechanism for the brand of the product, where the user will be able to scan the smart-tag through their mobile phone and the TagItSmart mobile app will provide information to user about the authenticity of the product. For a more detailed explanation of the process please refer to the Figure 11. This will enable the consumer to differentiate between the official and the fake products.

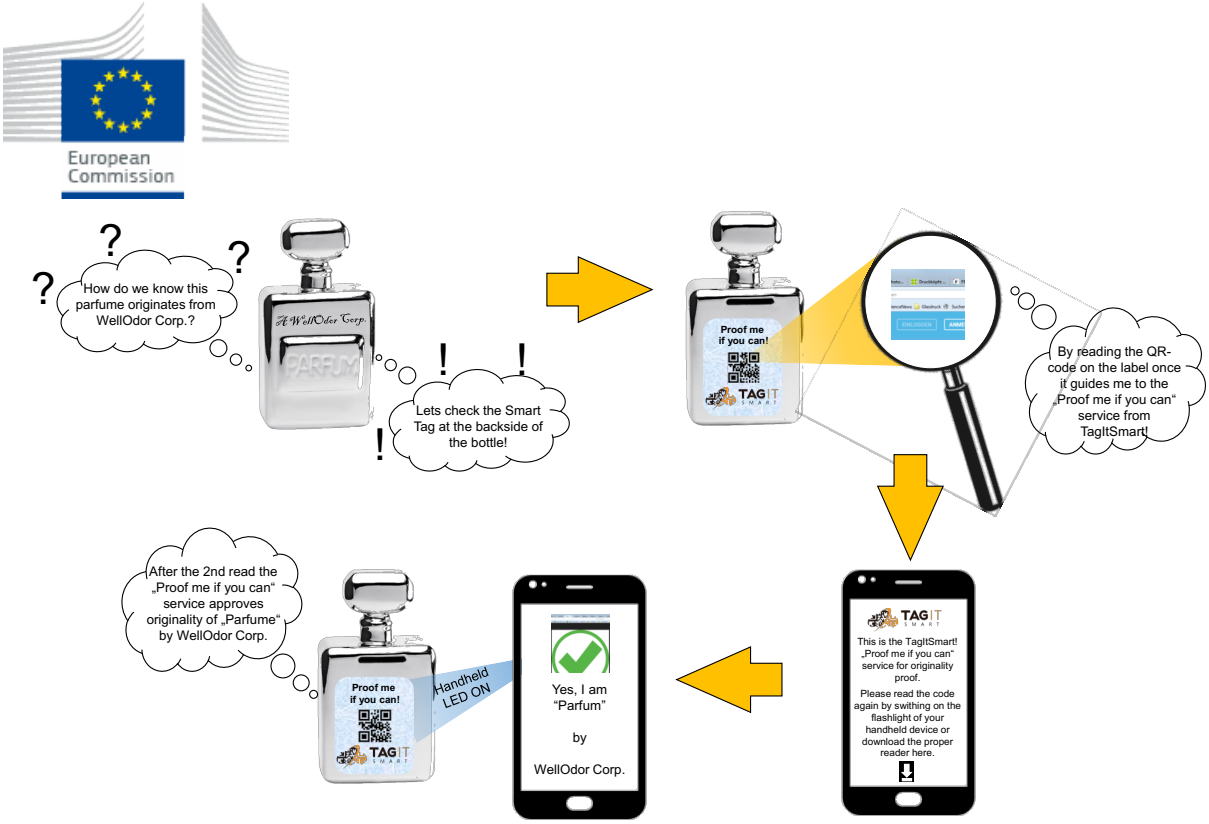

Figure 11. Brand protection use case.

BehavAuth contribution. BehavAuth will provide an authentication mechanism that will enable the end customer to authenticate through the TagItSmart mobile app to the TagItSmart platform. In this way, the user will be able to login to the TagItSmart mobile app and then scan the bar code in order to understand if the product is legitimate or not. BehavAuth will provide a secure and user-friendly authentication mechanism that will enable users to securely login to the mobile app and check the authenticity of the product while reducing counterfeiting.

2.1.4 Dynamic pricing

This use case is focused on providing the ability to the manufacturer to insert product information in the smart-tag and based on that allow the dynamic pricing concept to emerge. Once the product is created, the smart-tag is defined with information such as best before, ingredients, recycling, consumable before etc. This information allows the product change price depending for example on the current date, as particular thresholds of the information are added. Consumers are able to scan the smart-tags and through the mobile app they are able read information about the product such as best before, recycling, consumable before, recipes. The mobile app communicates with the platform and the dynamic pricing calculation provides the update price of the product (See Figure 12).

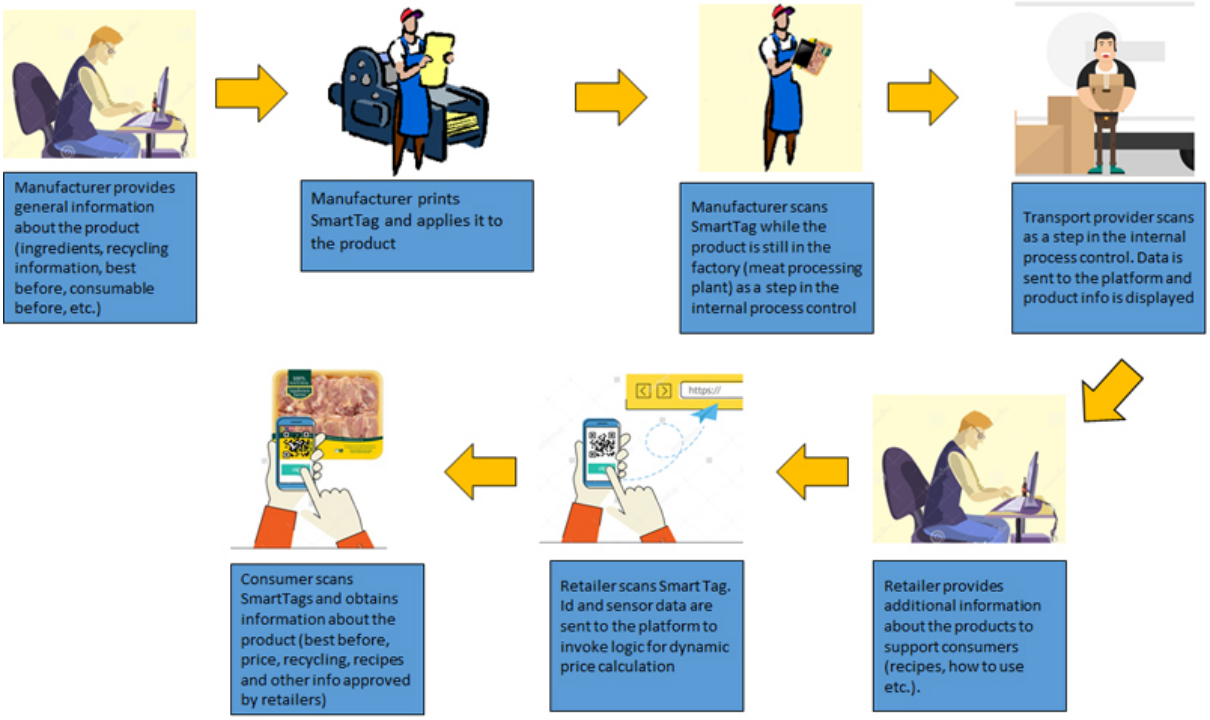

Figure 12. Dynamic pricing use case.

BehavAuth contribution. BehavAuth will provide an authentication mechanism that will enable the end customer to authenticate through the TagItSmart mobile app to the TagItSmart platform. In this way, the user will be able to login to the TagItSmart mobile app and then scan the bar code in order to read the dynamic price of the product. BehavAuth will provide a secure and user-friendly authentication mechanism that will enable users to securely login to the mobile app and check the various information of the product such as best before, consumable before, recycling, recipes as well as to calculate through secure communication with the TagItSmart platform the dynamic price of the product.

2.1.5 Home services

This use case focuses on monitoring, maintaining, repairing and replenishing of the heating equipment inside a household. Various of the TagItSmart services will be tested in real-world households, where the smart-tags will provide the appropriate environmental and contextual information regarding the heating equipment of a household. The monitored information is related to the actual heating equipment product, the environment in which it operates, the type of use performed by the owners of the product as well as additional services from the point-of-view of the product manufacturer. Figure 13 shows the overview of the use cases including the various stakeholders and the way they interact with the TagItSmart platform, the smart-tags and the different heating equipment.

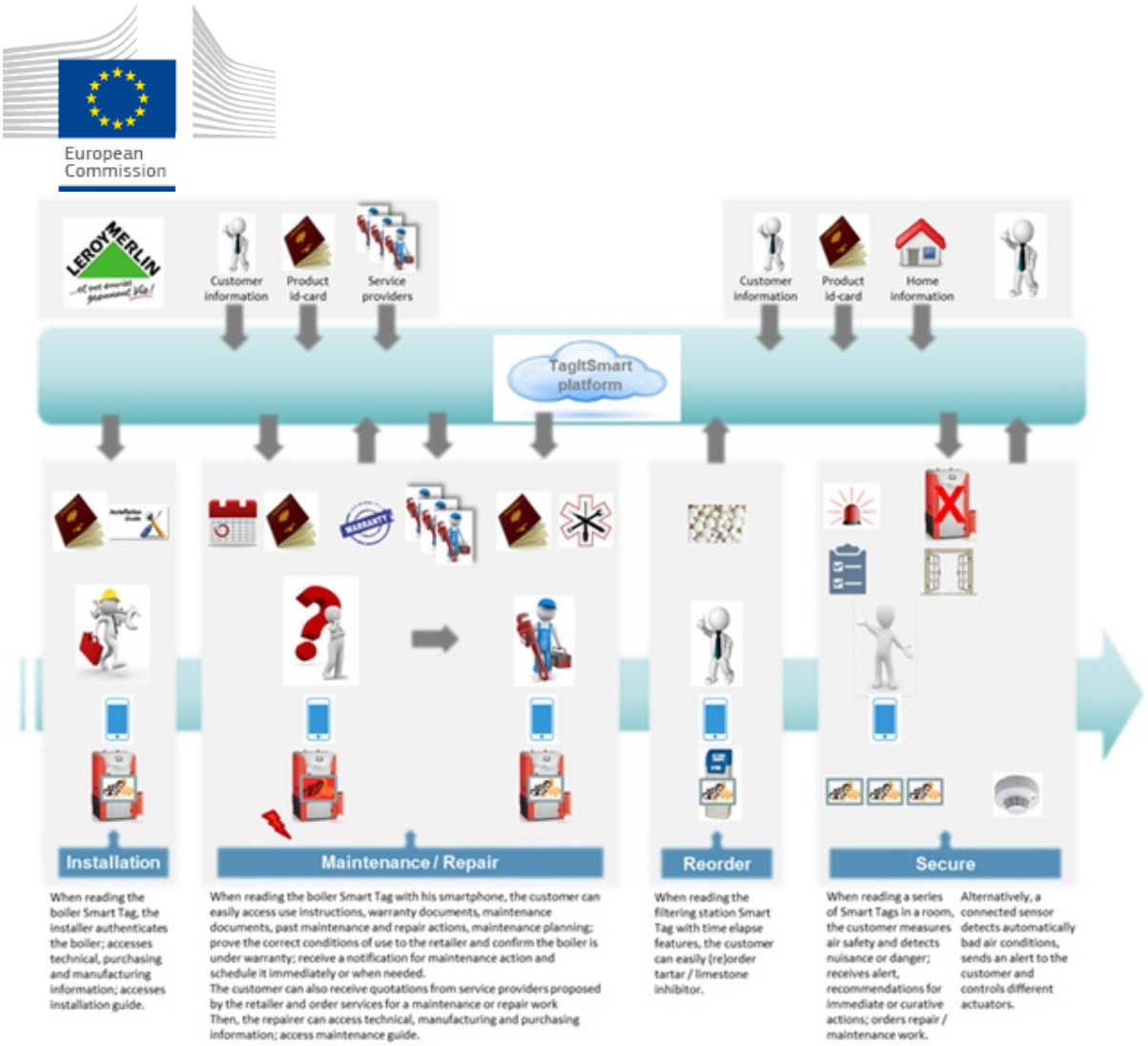

Figure 13. Home services use case.

BehavAuth contribution. BehavAuth will provide an authentication mechanism that will enable the end customer to authenticate through the TagItSmart mobile app to the TagItSmart platform. In this way, the user will be able to login to the TagItSmart mobile app and then scan the bar code in order to read the information from the different appliances for maintenance, monitoring, repair and replenishment.

2.2 Engaging stakeholders

As part of the process of defining the use cases and deriving the requirements of the projects, is the engagement of the related stakeholders. Among them are psychologists, early adopters, end-users and privacy experts that will allow QBE to shape the use cases and the requirements of the project.

2.2.1 Engaging psychologists

BehavAuth is a project that focuses on providing an authentication tool based on behavioural patterns of the users. This means that the authentication process is based on detecting the behavioural patterns of the users and based on them, to verify the identity of the device user. In order to correctly identify the behavioural cues required for the human behaviour recognition process, there is a need to engage psychologists. They will provide the know-how of

understanding human behaviour and the knowledge about which behavioural cues provide the most significant information in order to verify a user.
Among the various psychologists that we contacted as part of the system design and the dissemination of the project, Lisa Talia Moretti, a digital sociologist at University of Cardiff was engaged. Ms Moretti provided us various insights and information from the psychological aspect of the project. Some of the key points and behavioural cues that were mentioned are described below:

- **Average location or location hubs.** For most people, because of static locations of work/school/university and home (even the gym), they tend to frequent the same geographic area. Even for many consultants, who travel more than the average office worker, client work tends to be delivered around set project outcomes and therefore, even though their average location for work may change more than usual, there is a habit/pattern of behaviour that occurs during that timeframe. Indeed, consultants and the freelance community could be identified by their lack of average location stasis – a behavioural trait in itself.

- **Travel to our average location.** For many people, they have a tried and tested route to work and home. So not only is there a pattern that could emerge around an average location, but *how* people travel to those average locations is a pattern too. Does their average location include a train station? They are probably taking the train. Does it involve taking the back streets or main roads? The directions people take to travel to their average locations could become a key pattern to identifying them.

- **Speed and distance within the environment.** How we move around our environment is another behavioural trait worth exploring. For some it's by bicycle, for others by car, train or even, their two feet. This largely determines how fast we move and how many steps on average, we take a day. Therefore, looking at how fast someone moves around their environment in addition to the relationship between speed and distance is important.

- **Relationship with our phones.** Studying how we hold, carry and store our phones could prove to be one of the most unique, small data points we have in identifying an individual. From hand to back pocket, or hand to handbag, in addition to whether you have a preference for taking photos landscape or portrait or indeed, whether you watch your videos landscape or portrait. In addition, there are small habitual movements that some people have developed when their phone is in their hands; spinning it around with two fingers, resting the long-edge on a flat surface while tipping it from side-to-side. In addition, the movement upwards towards an ear – indicating calls – versus, portrait being held still mid-air – indicating texting – would prove to be interesting small data points to capture; although this may not prove to be as useful because of people using headphones (whether tethered or untethered to the phone) to create a hands-free experience.

Not only is it important to measure these unique movements, but to timestamp them as these behaviours may be unique to certain points of an individual's day, e.g. watching videos on the bus on the commute home.

Large moments of inactivity can either indicate sleeping patterns. Again, working out a kind of average and time stamping this – time between inactivity and activity - could prove interesting.

In the current world of the global workforce, a large amount of inactivity could indicate time spent flying – something worth considering when looking at 'inactive' data.

2.2.2 Engaging potential early adopters

As part of the system design based on the use cases and the requirement analysis is the engagement with potential early adopters of the BehavAuth. The early adopters of the platform constitute significant part of the initial design of the BehavAuth tool. This is because BehavAuth is creating a tool that considers stakeholders during the whole duration of the project, to receive feedback and to define the system based on the requirements of the stakeholders including the early adopters.

The first early adopter of the BehavAuth tool is the TagItSmart platform, which will allow QBE to shape the use cases and the requirements based on the TagItSmart platform needs. Following, are the requirements extracted from the engagement with the early adopters of the platform:

- **Non-intrusive**. The use of personal data does not affect only the end-users but also the potential early adopters of the platform. This is because of the new EU regulation about personal data (GDPR [12]). Particular measures need to be taken by the adopters of the platform, to safeguard their organisations (including TagItSmart) in order to protect their users' privacy. TagItSmart consortium pointed out that it is important to utilise less intrusive personal data and to avoid methods such as face and voice recognition.

- **Low cost**. Pricing is always considerable when a third-party platform wants to adopt a new authentication solution. A reduced price in comparison to the solutions that are already in the market could intrigue third-party platforms to change to a more innovative authentication tool. The cost reduction could be also translated in terms of minimisation of security breaches, meaning that a more secure system could save money/reduce the costs for a third-party company without providing a 'cheaper' (in terms of pricing) solution.

- **Secure**. Security is imperative for authentication systems in order to convince third-party systems to discard any legacy authentication tools and to adopt innovative authentication tools such as BehavAuth. In order to achieve this, the BehavAuth tool needs to provide a more or equally secure solution as the state-of-the-art systems.

- **Compatible**. Compatibility is a very important requirement in the design of an authentication tool. State-of-the-art protocols and standards needs to be followed in order for the BehavAuth to be compatible with the majority of the third-party systems that are currently already in the market such as OAuth 2.0/OpenID and SAML.

- **Portability**. Depending on the third-party platform, it may already have a user management component. For that reason, the BehavAuth architecture needs to be portable in order to be able to operate both with platforms that do already have a user management component and with platforms that do not have a user management component and require from BehavAuth to provide that service.

- **Scalability**. The system needs to be scalable in order to be able to serve a large number of users independent of the workload and the number of requests. The

BehavAuth tool will be implemented as an Authentication-as-a-Service on cloud infrastructure in order to enable the scalability of system.

2.2.3 End-users

End-users constitute a significant component of the project's stakeholders as they will be the actual users of the authentication mechanism. They will have direct access to the mobile app that will collect data from the smartphones' sensors and device usage data in order to infer the identity of the users. End-users will directly interact with the app. For that reason, it is imperative need to receive feedback from the end-users regarding the design of the system and in particular the mobile app. End-users need to provide feedback about the BehavAuth platform, in order to help us shape the use cases and the requirements of the platform. At the stage of the initial design of the platform, end-users provided some initial requirements that they consider important in such type of applications.

Following, some key points that were outlined by the end-users are described:

- **Training process**. There is a need to minimise the end-user involvement in the training process of the platform for a particular user. As mentioned the BehavAuth platform will learning the behavioural patterns of the user in order to authenticate them at the TagItSmart platform. Thus, the user involvement needs to be minimal in that process.

- **Privacy**. The data collected by the mobile app may be intrusive regarding the privacy of the user. BehavAuth needs to pay particular attention on the privacy aspect in order to safeguard the personal data of the end-users.

- **User-friendly**. The platform needs to be user-friendly and allow the user authenticate at the TagItSmart platform in an easy, fast and secure way.

- **Security**. The platform must provide a secure solution for the end-users and the TagItSmart platform, which minimises the vulnerabilities and improves the existing state-of-the-art systems. This requires a secure architecture of the overall system but also accurate machine-learning and artificial intelligence models that will allow the accurate and robust recognition of the behavioural patterns.

2.2.4 Engaging privacy experts

BehavAuth focuses on collecting sensor data from Internet of Things devices including smartphones in order to authenticate users to the TagItSmart platform based on behavioural patterns. The collected data that will allow the authentication of the users should be correlated with particular users, which will allow the differentiation among the users and further the identification of the users. Meaning that the data collected to identify the users are personal data. BehavAuth pays particular attention to the privacy and security of the users by recognising the existence of personal data in the authentication procedure. For that reason, from the beginning of the project, has defined some actions in order to preserve the privacy of the users. Among these actions are the engagement of privacy experts during the whole duration of the project who will provide feedback regarding the privacy aspects of the projects. Furthermore, an ethical review will be provided at the end of the project to evaluate the privacy issues that may arise during the project.

As defined in the project plan and pointed out by the consortium of TagItSmart, BehavAuth engaged a privacy expert from Digital Catapult (UK), Dr Michele Nati. Dr Nati is a Lead Technologist for Personal Data and Trust with a great focus on privacy from the point of the end-users. QBE established a collaboration with Dr Nati, in order to receive feedback about the potential privacy issues that may arise during the project. The BehavAuth project was presented to the privacy expert including(together with?) an overview of the initial tool design. Dr Nati provided feedback and considering the new European Commission data privacy protection scheme (GDPR), he helped us shape the initial tool design from the perspective of users' privacy. A few points that where highlighted from the privacy expert which need to be taken into consideration

- **Consent**. For each biometric or data that will be collected for the authentication of the user, there is a need to receive explicit consent from the user. The consent must be requested in clear and plain language. For example, in case the tool uses three biometrics, an explicit consent in plain language must be requested from the user for each one of the biometrics.

- **Breach notification**. In case of a breach, the end user needs to be informed at most in 72 hour from the breach.

- **Right to access**. Users are allowed to request confirmation if personal data is collected, where, and for what purpose. Furthermore, users could ask for the personal data collected, thus BehavAuth shall provide a copy of the personal data, free of charge, in an electronic format.

- **Right to be forgotten**. The entity or organisation that controls the personal data collected from the users, needs to provide the ability to the user to delete any collected personal data, to cease any dissemination of the data and potentially have third parties halt processing of the data.

- **Data portability.** User can ask from BehavAuth to receive the personal data concerning them and have the right to transmit that data to another company.

- **Privacy by design**. BehavAuth should collect and store only personal data that are necessary for the completion of the authentication process (data minimization), and refuse personal data access to people that are not immediately connected to the authentication process.

2.3 Risk analysis

This section provides a description of the risks that are related to the BehavAuth system. The risk analysis that was performed had as sole purpose to assist the design of the system by understanding the use cases and requirements of the system, as well as, the risks that are related to the system. Table 2.1 provides a more detailed explanation of the potential risks that arose from the risk analysis. The first column provides a description of the risk, the second column shows the impact of that risk (High/Medium/Low), the third column present the likelihood of that risk taking place (High/Medium/Low) and the last column provides the way how BehavAuth tries to mitigate that risk.

Table 2.1. Risk analysis

Risk	Impact	Likelihood	Mitigation
Malicious user tries to register to the system, (registering multiple ids)	Medium	Low	Third-party company can only initiate the registration of the user. This risk has been mitigated by the design of the system, where the mobile app itself cannot register a new user at the BehavAuth platform; only the third-party platform is able to perform a new user registration. This assumes the trustworthiness of the third-party company that wants to securely authenticate its users.
Malicious user tries to train the system	High	Low	Third-party company can only initiate the registration of the user. The training process is encapsulated in the registration process of a new user at the BehavAuth system. A new user is authenticated by the third-party platform through credentials, which have generated a token. This token is used to securely transfer the initial training biometric data from the device to the third-party platform and further to the BehavAuth platform. Then, the initialisation method is called by the third-party platform, including the training biometric data. The initialisation method creates and trains the initial models. Thus, the malicious users cannot train the models.
False positive authentication followed by training	High	Low	Multiple biometrics are proposed by the BehavAuth platform which has as sole purpose to increase the authentication rate (meaning reduce to minimum the likelihood) and reduce the rate of false positives. As the correct authentication rate will be very high, the likelihood of faulty accepting an authentication and including the faulty biometric data in the training process, will be very small.
Biometrics fail to authenticate user may not be able to access the service	Medium	Low	Use of secondary username/password credentials to sign in at the service. This constitutes a back-up mechanism in case all biometrics fail to authenticate and the actual user tries still to access the service. A limit of authentication tries will be implemented for the BehavAuth, to avoid attacks such as Brute Force.
Compromise communication channel between third-party platform & BehavAuth. Spoofing of the APP_KEY used	High	Low	An HTTPS communication channel will be established between the third-party and the BehavAuth platforms. In addition, data transmitted between the two end-points will be encrypted to strengthen even more the security among the two systems.

in BehavAuth platform.			
Each biometrics has a certain error of false positives.	Low	High	The use of multiple biometrics is the solution for mitigating this problem. It is a problem that is known in advance. However, the combination of multiple biometrics will allow each approach to compliment the error of other methods, leading to an authentication mechanism with very high accuracy i.e. minimal number of false positives.
Biometrics templates get exposed	High	Low	Encryption of the biometrics templates when created in the BehavAuth platform and stored at the database storage. This will enable the secure storage and retrieval of the biometric templates. The create/read/update/delete methods of the biometric templates are all implemented at the BehavAuth platform, which minimises the exposure of the data to malicious users.
Impersonate at the mobile app level, app sends wrong sensor data	Medium	Medium	An authentication token allows initial secure identification of the mobile app. Then, encryption hides the types of data that are sent to the platform.
Impersonate at user level	Medium	Low	Multiple biometrics constitute the solution for impersonating the biometric at the user level. For example, a malicious user might attempt to trick the voice recognition component, however BehavAuth combines multiple biometrics and behavioural patterns which constitute the impersonation practically impossible.
Reverse engineer the learning mechanism	Medium	Low	The learning mechanism is developed on the cloud and the mobile app implements only data collection process. For that reason, the behavioural pattern learning mechanism is highly unlucky to be exposed to third-parties as the code is preserved at a cloud infrastructure and is not accessible to others except from the BehavAuth developers.
Exposure of personal data in case of data breach	Low	Medium	Anonymous id is only registered. BehavAuth stores only a pseudonymised ID followed by the behavioural personal data. This implements the so-called *unlinkability* of the data, as it is not possible to reach the person for whom the data have been collected, unless there is a link between the pseudonym and the identification data. The identification data are stored at the third-party platform. In that way, the sensor data that are collected are unlikable.
User has requested data and data breach	N/A	Medium	It occurs outside the ecosystem of BehavAuth as from the point BehavAuth has provided the copy of the data to the user in a portable form

occurs at the end-users side.			and a breach occurs at the user's end, then BehavAuth is not related to that

2.4 Requirement analysis

This section describes the requirements derived from the use cases and the engagement of the various stakeholders such as end-users, psychologists, potential early adopters of the platform and privacy experts. Table 2.2 provides a list of the requirements including category, requirement ID (REQ_ID) and description of each requirement.

Table 2.2. Functional requirements.

Category	REQ_ID	Description
Security	F01	The tool must provide a secure communication channel between the cloud and the mobile devices.
	F02	The platform must provide a secure communication channel between the cloud and the TagItSmart platform.
	F03	The platform must ensure the integrity of all the communications.
Privacy	F04	The platform must should store a pseudonym for each registered user.
	F05	Users' data should be processed only when explicit consent has been given by the users.
	F06	Users should have the ability to stop the data collection and processing at any time.
	F07	Only data required for the authentication mechanism should be collected.
	F08	Users should have the ability to delete the collected data and any extracted knowledge.
ID	F10	Only the pseudonym of each user will be stored, followed by the behavioural data.
	F11	The ID of each user will be anonymised.
	F12	The platform should allow the registration and deletion of users.
Authentication	F13	The authentication mechanism should user-friendly.
	F14	The authentication mechanism should allow developers to easily integrate with TagItSmart and other third-party platforms.
	F15	The behavioural patterns should accurately identify and verify the users.

Section 3 - Reusability analysis

This section provides a reusability analysis of existing components and solution, that could be utilised in the design of the BehavAuth system. The analysis is split in three different types of frameworks that could be utilised in the BehavAuth system including web application frameworks, machine-learning frameworks and data collection frameworks.

3.1 Web application frameworks

This subsection provides an overview of the available web app development frameworks. Modern web application development is based on widely-used frameworks that have been developed through state-of-the-art design software design patterns. Common practices have been applied when developing these frameworks, while in parallel refining them and creating a solid base for the development of a new web application. This allows developers to re-use already existing mature technologies/frameworks. This not only avoids re-inventing the wheel (saves time), but also to establishes a solid base for a reliable and robust backend platform. Internally, most the systems are handling basic functionalities that are common in the majority of web applications, such as user management, exposure of services, scalability in terms of number of users and requests. Each of these frameworks have certain predefined capabilities and, in most cases, support similar functionalities either inherently or by adding external software components. A key criterion for the analysed frameworks was the fact of them being released as open-source, in order for BehavAuth to be able to leverage the advantages of the open-source software such as the community support, maturity, and security.

The web application development frameworks are being analysed based on the:

1. *Programming language* in which they are developed.

2. *Security components* that are already built-in the framework.

3. *Open-source license* that they release by, in order to take into account any licensing issues of the BehavAuth.

4. *Scalability* is an important factor in this case, as BehavAuth focuses on serving a large number of users, making scalability an imperative functionality. It is rated as High/Medium/Low, based on the native capability of the framework to scale to a larger number of users.

5. *Machine-learning compatibility* is of great importance, as the authentication service will be based on leveraging machine-learning models in order to validate the identity of the users.

6. *NoSQL support* is imperative as the BehavAuth platform will scale to a large number of users, the ability to store behavioural data in a distributed manner is imperative.

Table 3.1. Web application frameworks.

Frame-work	Programming Language	Security	Open-source license	Scalable	Machine-learning compatibility	NoSQL support
Django [14]	Python	Built-in (XSS, CSRF, SQL injection, clickjacking protection, SSL/HTTPS, Host header validation, session security	BSD	High	High	Yes
Play [15]	Java, Scala	Add-ons	Apache 2.0	High	High	Yes
Spring [16]	Java	Spring security embedded	Apache 2.0	High	High	Yes
Ruby on Rails [17]	Ruby	Built-in (SQL Injection, CSRF, XSS), needs add-ons	MIT	Low/ Medium	Low	Yes
Symfony [18]	PHP	Core, HTTP, CSRF, ACL	MIT	High	Low	Yes
Laravel [19]	PHP	Authentication / encryption	MIT	Medium	Low	Yes
Node [20]	Javascript	Add-ons	MIT	High	Medium	Yes
ASP.NET [21]	.NET	Built-in security, authentication, authorization	.NET Foundation	High	Medium	Yes

Django [14] is one of the most popular web application frameworks following the model-view-template architecture design pattern. It is used by many major companies, including NASA, and its first version was released in 2005. It is safe to say that Django is a quite mature framework. Play Framework [15], which is used by LinkedIn, was first released in 2009, and follows a quite different architecture of native Java application and more similar to Ruby on Rails framework. Spring [16] was firstly released in 2002, constituting one of the mostly used and widely adopted Java-based frameworks. Ruby on Rails [17], which was firstly released in 2005 allowing it to establish a certain maturity period, is widely used for prototype development and has a concise architecture. Symfony [18] was also released in 2005 and was highly inspired by Spring, focusing on creating robust enterprise application with low performance overheads. Laravel [19] is a more recent framework, released in 2011, that follows the model-view-controller architecture pattern and is amongst the most popular PHP frameworks, together with Symfony. Node [20] was presented in 2009, and since then, it has been widely adopted for prototype development. However, concerns arise in the compatibility issues

among the different versions of the packages installed as add-ons to the framework. ASP.NET [21] and is a framework supported by Microsoft for developing enterprise applications for building dynamic websites, web applications and web services.

3.2 Machine-learning frameworks

This subsection provides an overview of the current machine-learning frameworks available. Each of the available machine-learning frameworks provide their own capabilities and design principles. In the literature review of the state-of-the-art machine-learning frameworks, only open-source frameworks were considered, as this constitutes a pre-requisite for the use of the framework through the BehavAuth platform. This will allow the BehavAuth platform to build upon the open source framework in order to develop the authentication platform and in parallel leverage the large advantages provided by the open-source community including support, openness, and security. Regarding the business model, it is important to develop a sustainable business plan that will minimise the costs of the platform. Hence, selecting a free and open-source framework would allow BehavAuth to reduce the costs of using a paid third-party solution. Table 3.2 presents the currently available open-source solutions for developing machine-learning techniques on the cloud, including the:

1. *Available programming language,* the frameworks support certain programming languages that must be compatible with the web application framework, in order to early spot any compatibility issues.

2. *Supported machine-learning techniques,* depending on the machine-learning techniques that are required by the authentication mechanism, a corresponding machine-learning framework should support these requirements.

3. *Open-source release license,* as the machine-learning framework will become part of the BehavAuth platform, any license compatibility issues needs to be identified at the early stage of the requirement analysis.

4. *The primary use of the framework*, depending on the behavioural cues that BehavAuth will focus on, “a more suitable” framework in the analysis of the particular behavioural cues is required.

Table 3.2. Machine-learning frameworks.

Framework	Programming language	Machine-learning techniques	Open-source license	Primary use
Apache Singa [22]	Python, C++	Deep learning, feed-forward models like convolutional neural networks (CNN), energy models like Restricted Boltzmann Machine (RBM), and recurrent neural networks (RNN)	Apache 2.0	Natural language processing, Image recognition

Shogun [23]	C++, Python, Octave, R, Java, Lua and C++	Classification, regression, dimensionality reduction, clustering, incl. wealth of efficient SVM implementations, multiple kernel learning, kernel hypothesis testing, Krylov method	GPL v3	Unified large-scale learning for a broad range of feature types and learning settings
Apache Mahout [24]	Java	Distributed and scalable machine-learning algorithms incl. collaborative filtering, clustering and classification	Apache 2.0	Recommendation engine, clustering, content classification
Apache Spark MLlib [25]	Java, Scala, Python, R	Classification, clustering, collaborative filtering, dimensionality reduction, lower-level optimisation primitives	Apache 2.0	Distributed machine-learning framework
TensorFlow [26]	C++, Python	Graph flows, deep learning, neural networks, dynamically-sized multi-dimensional data	Apache 2.0	Perceptual and language understanding, deep learning
Oryx 2 [27]	Java	Collaborative filtering, classification, regression, clustering	Apache 2.0	App building, end-to-end apps
Accord.NET [28]	.NET	Classification, regression, clustering, distributions, hypothesis tests, kernel methods	GNU Lesser Public License v2.1	Scientific computing, Statistical data processing, pattern recognition, linear algebra

Apache Singa [22] has been developed as a distributed deep-learning platform to enable developers of multimedia platforms, including natural language processing and image recognition. Shogun [23] is a toolbox that includes algorithms and data structures for machine-learning problems, which allows the interconnection with other machine-learning libraries. Apache Mahout [24] focuses on providing distributed and scalable machine-learning algorithms, using Hadoop and MapReduce paradigm. Apache Spark MLlib [25] focuses on providing an easy way to solve practical machine-learning problems in a scalable way, including feature extraction, machine-learning algorithm, pipelines, persistence, and other assistive utilities. TensorFlow [26] is a numerical computation library that uses data flow graphs for developing machine-learning and deep learning algorithms in a distributed way, allowing the deployment on diverse devices including desktop, server and mobile device while reusing the same source code. Oryx 2 [27] is a framework for real-time large-scale machine-learning that allows developers to build applications including collaborative filtering, classification, regression, and clustering. The framework implements lambda architecture and is built on Apache Spark and Apache Kafka. Accord.NET [28] is a framework for developing machine-learning commercial solutions using .NET, focused mainly on audio and video processing applications.

3.3 Data collection frameworks

This subsection focuses principally on sensing frameworks that perform only data collection and some minor pre-processing, allowing the extraction of human behaviour off-line [13]. An application is deployed on the device, which collects data from pre-configured sensors, and then either stores the information on device's internal storage, or performs uploading to a backend server. The retrieved data are merged and forwarded to the next stage in order to extract behavioural cues. Table 3.3 summarises the existing frameworks for data collection by presenting the:

1. *Type of sensor data retrieved,* depending on the data required to be collected, an appropriate data collection framework needs to be selected, that will allow the collection of the required data.

2. Incorporation of a mechanism for *energy efficiency* such as adaptive sensing, as BehavAuth will continuously collect data from the users, and energy efficiency is imperative in order to avoid any side effects at the level of the user experience.

3. Embodiment of a *privacy preserving* approach, as BehavAuth focuses on providing a privacy preserving approach, it is important to leverage any existing mechanisms.

4. The *software license* through which the authors released a particular framework, in order to avoid any incompatibility issues with other components.

The above columns, constitute the criteria for the selection of the data collection framework.

Table 3.3. Data collection frameworks [13].

Framework	Type of sensors				Energy Efficient	Privacy	License
	Inertial	Ambient	Position	Virtual			
MyExperience [29]		X	X	X			BSD
SeeMon [30]	X	X	X	X	X		
OpenDataKit [31]	X	X	X	X			Apache 2.0
SystemSens [32]	X	X	X	X			Custom
Funf [33]	X	X	X	X			GNU GPL
Medusa [34]	X	X	X				Apache 2.0
METIS [35]	X	X	X	X	X		
MSF [36]	X	X	X	X			
Anonysense [37]	X	X	X	X		X	
LiveLab [38]	X	X	X	X		X	
PRISM [39]	X	X	X	X	X		

MyExperience [29] retrieves and logs information from physical sensors and phone usage patterns. SeeMon [30] investigates the context in which the device is and adapts the sensing process based on it. This is achieved by mining local sensors and installed applications in an energy-efficient manner. OpenDataKit [31] is a set of tools for sensing and aggregating data from mobile phones. SystemSens [32] constitutes a data retrieval tool that enables researchers to capture mobile data in large scale experiments. Funf [33] is a mobile data collection platform, developed by MIT Media Labs, characterised by easily-configurable sensing and data-retrieval capabilities. To achieve crowd-sensing and simultaneous coordination of multiple mobile devices, Medusa [34] was designed. METIS [35] is a distributed system that decides, based on the device status and user context, to perform on-device or infrastructure-oriented sensing. MSF [36] is a recent data collection framework that follows multi-pipeline architecture and targets in providing an abstraction regarding the sensing process. To preserve privacy in crowd-sensing applications, [37] introduced Anonysense, which permitted, through a centralised sensing coordination, the collection and storage of anonymous smartphone-sensed data in a collaborative manner. Another approach that focuses on privacy preservation, but also on energy efficiency for long-range duration deployment, is LiveLab [38], by storing inertial, phone usage, positioning and network-based data. Further, in [39] they developed PRISM, which is a platform for dynamic deployment of participatory and opportunistic sensing applications on mobile phones through a push mechanism, while maintaining security through a sandboxed environment.

Section 4 - Initial design

This section provides an initial design of the BehavAuth system by taking into consideration the uses cases and requirements detailed in the sections below. It should be mentioned that stakeholders have been engaged during the use case and requirement definition process, in order to create a system that is tailored to the stakeholders' needs, including end-users and third-party platforms. The section is structured as follows; the first subsection provides an overview of the design of the system followed by a more detailed description of each of the core components of the system.

4.1 Overall design of the system

This section provides an overview of the initial design of the BehavAuth system. BehavAuth focuses on authenticating users for third-party platforms, such as TagItSmart. For the design of the authentication system, state-of-the-art architecture design patterns were considered [40] from IEEE Centre for Secure Design. The authentication process is based on understanding and learning the behavioural patterns of the users by leveraging data extracted from smartphones, such as sensor and device usage data. The data are collected from a smartphone app and are forwarded to the TagItSmart platform. Then the TagItSmart platform forwards the collected data, including a pseudo-ID, to the BehavAuth platform. The BehavAuth platform will, as a first step, create a record of the pseudo-ID and initiate the learning process for the authentication models. Once the behavioural models have learned the patterns of the user, the same path will be followed for the biometric data. However, in this case, the authentication service will be also executed. Initially, it will perform authentication followed by learning, in order to adapt to the contextual changes of the person.

The BehavAuth constitutes of:

- **The platform**, where the processing and the authentication procedure takes place.
- **The mobile app/library**, where the data are being collected in order to send them to the back-end platform.

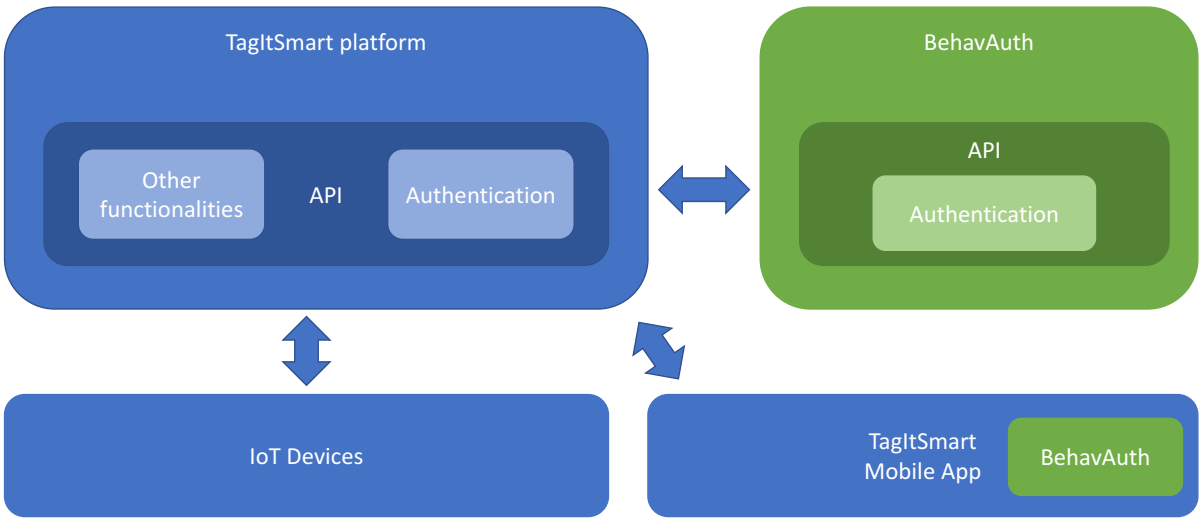

Figure 14. BehavAuth overall architecture.

Figure 14 shows the overall design of the BehavAuth system including the scenario, where TagItSmart is the third-party platform which requires an authentication mechanism based on behavioural patterns. The overall system consists of the TagItSmart and the BehavAuth platforms, the IoT devices that are coordinated by the TagItSmart platform, and the TagItSmart mobile app (which includes a BehavAuth component). The TagItSmart mobile app will integrate the BehavAuth library. The latter will constitute the component which collects the behavioural data from the smartphone, encrypts them and then the TagItSmart, through the secure communication channel it has established, it will propagate the behavioural data to the TagItSmart platform and further to the BehavAuth platform, in order to perform authentication and inform the TagItSmart platform that this is the actual user and not some imposter.

4.2 BehavAuth platform

The first core component of the BehavAuth system is the cloud platform. This component of the system implements the machine-learning and artificial intelligence models using sensor and device usage data from users' smartphones. The models are trained initially and then adapt based on the contextual information in which the user is. To ensure the validity of the user at the initialisation process, the BehavAuth platform communicates only with the TagItSmart platform, which has previously established a secure communication with the mobile app. Authentication tokens are used between the TagItSmart platform and mobile app, to ensure the secure communication and validation among the two components. Similar secure communication is established between the two platforms.

The main functionalities that the BehavAuth platform will provide are:

- **User registration**. Users are registered at the BehavAuth platform through the TagItSmart platform. The latter uses a pseudonym, an application key and the behavioural data to register a user at the BehavAuth platform. During this registration, an initialisation process is performed, the appropriate entities are created, and the training process of the various machine-learning models is initiated.

- **User authentication**. Given that a user has been registered at the BehavAuth platform, an authentication service will be exposed, that will allow users to authenticate to the TagItSmart platform. The various machine-learning models will receive the behavioural data of the user and will perform validation. A response will be sent to the TagItSmart platform indicating the result i.e. if the user is an imposter or not. In case this is the actual user, the biometric data will be fed to the machine-learning models in order to implement the adaptation based on the contextual data.

4.3 BehavAuth mobile app/library

The second core component of the BehavAuth system is the mobile app/library. BehavAuth performs authentication by collecting data from smartphone's sensors, as well as, the device usage in order to identify behavioural patterns which allow the unique authentication of the device owner.

The main functionalities of the BehavAuth library will be:

- **Data collection**. The library will implement the data collection process regarding all the predefined biometrics. It should be noted that the data that will be collected are based on the user's permission. At any point, the user will be able to suspend the data collection process and remove any collected data at both device level and back-end platform.

- **Data encryption**. The library will include a public encryption key that will allow the app to encrypt inside the library all the collected data and then forward them to the TagItSmart app and further to the TagItSmart platform.

Section 5 - Conclusion

This deliverable described the initial design of the BehavAuth tool based on the use cases and the requirement analysis.

In addition to the initial Description of Work (DoW) for this deliverable, this document included a detailed analysis of the existing state-of-the-art authentication systems that are already available in the market. The analysis included a description of the service provided by each authentication platform and in addition a section for each one where the gaps and drawbacks of each solution were presented. These gaps validated the initial research done before the beginning of the project, about the major issues identified in today's authentication systems, as well as, the lack of continuous, secure and user-friendly authentication. Current authentication systems use at most two biometrics that operate as individual authentication mechanisms and are not combined to provide a more secure and robust authentication solution.

Next, as suggested in the DoW, the BehavAuth collaborators defined the use cases and the requirements of the project by engaging the stakeholders of the project such as psychologists, end-users, potential early adopters of the platform, and privacy experts. Each stakeholder showed great enthusiasm and provided great insights about the definition of the use cases and the derivation of the project's requirements. A risk analysis was added upon the DoW, to ensure the project takes into consideration all the potential risks and also provides an appropriate mitigation strategy for each risk. Given the above analysis, the initial functional requirements for the project were defined, which will be evaluated throughout the duration of the project.

Following, the predefined reusability analysis was performed in order to leverage potential existing technologies and avoid reinventing the wheel. The reusability analysis was split into sections for the web application development framework, the machine-learning frameworks and the data collection frameworks. For each section state-of-the-art frameworks were analysed based on certain evaluation criteria that would allow the adoption of the particular framework and the integration with the rest of the components.

Finally, an initial design of the BehavAuth tool architecture was provided, taking into account the TagItSmart platform as an initial use case. BehavAuth collaborators concluded on the particular design based on the requirements derived from existing state-of-the-art authentication systems and the stakeholder engagement. This constitutes an initial design of the BehavAuth architecture and given the use of agile software development methodologies, this will be refined during the development and evaluation process.

Section 6 - References

1. Single ID - http://singleid.com/
2. Auth0 - https://auth0.com/
3. Google Identity platform - https://developers.google.com/identity/
4. M. Faundez-Zanuy, "On the vulnerability of biometric security systems," in *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 6, pp. 3-8, June 2004.
5. Identity X platform - https://www.daon.com/products/identityx-platform
6. CallSign - https://www.callsign.com/
7. Twillio authentication platform - https://www.twilio.com/two-factor-authentication
8. TrustedX authentication platform - https://www.safelayer.com/en/products/trustedx-adaptive-authentication
9. Entrust authentication platform - https://www.entrust.com/news/entrust-versatile-authentication-platform-enables-secure-online-access-for-virtual-documents-customers/
10. Schneier, Bruce. "Two-factor authentication: Too little, too late." *Commun. ACM* 48, no. 4 (2005): 136.
11. TagItSmart EU project – www.tagitsmart.eu
12. General Data Protection Regulation - http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679
13. Niklas Palaghias, Seyed Amir Hoseinitabatabaei, Michele Nati, Alexander Gluhak, and Klaus Moessner. 2016. A Survey on Mobile Social Signal Processing. *ACM Comput. Surv.* 48, 4, Article 57 (March 2016), 52 pages.
14. Django web application framework - https://www.djangoproject.com/
15. Play web application framework - https://www.playframework.com/
16. Spring web application framework - https://spring.io/
17. Ruby on rails web application framework – http://rubyonrails.org/
18. Symfony web application framework - https://symfony.com/
19. Laravel web application framework - https://laravel.com/
20. Node web application framework - https://nodejs.org/en/
21. ASP.NET web application framework - https://www.asp.net/
22. Apache Singa Machine learning framework - http://singa.apache.org/en/index.html
23. Shogun Toolbox Machine learning framework - http://www.shogun-toolbox.org/
24. Apache Mahout Machine learning framework - https://mahout.apache.org/
25. Apache Spark MLlib Machine learning framework - https://spark.apache.org/mllib/
26. Tensorflow Machine learning framework - https://www.tensorflow.org/
27. Oryx 2 Machine learning framework - https://github.com/OryxProject/oryx
28. Accord.NET Machine learning framework - http://accord-framework.net/
29. Jon Froehlich, Mike Y. Chen, Sunny Consolvo, Beverly Harrison, and James A. Landay. 2007. MyExperience: A System for in Situ Tracing and Capturing of User Feedback on Mobile Phones. In *Proceedings of the 5th Interna- tional Conference on Mobile Systems, Applications and Services (MobiSys '07)*. ACM, New York, NY, USA, 57–70. DOI:http://dx.doi.org/10.1145/1247660.1247670
30. Seungwoo Kang, Jinwon Lee, Hyukjae Jang, Hyonik Lee, Youngki Lee, Souneil Park, Taiwoo Park, and Junehwa Song. 2008. SeeMon: Scalable and Energy-efficient Context Monitoring Framework for Sensor-rich Mobile Environments. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (Mo- biSys '08)*. ACM, New York, NY, USA, 267–280. DOI:http://dx.doi.org/10.1145/1378600.1378630
31. Y. Anokwa, C. Hartung, W. Brunette, G. Borriello, and A. Lerer. 2009. Open Source Data Collection in the Developing World. *Computer* 42, 10 (Oct 2009), 97–99. DOI:http://dx.doi.org/10.1109/MC.2009.328

32. Hossein Falaki, Ratul Mahajan, and Deborah Estrin. 2011. SystemSens: A Tool for Monitoring Usage in Smartphone Research Deployments. In *Proceedings of the Sixth International Workshop on MobiArch (MobiArch '11)*. ACM, New York, NY, USA, 25–30. DOI:http://dx.doi.org/10.1145/1999916.1999923
33. Funf. 2011. Open Sensing Framework. (2011). [online] http://www.funf.org.
34. Moo-Ryong Ra, Bin Liu, Tom La Porta, and Ramesh Govindan. 2012. Medusa: A Programming Framework for Crowd-Sensing Applications. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys'12)*.
35. Kiran K. Rachuri, Christos Efstratiou, Ilias Leontiadis, Cecilia Mascolo, and Peter J. Rentfrow. 2013. METIS: Exploring mobile phone sensing offloading for efficiently supporting social sensing applications. *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)* 0 (2013), 85–93. DOI:http://dx.doi.org/10.1109/PerCom.2013.6526718
36. Giuseppe Cardone, Andrea Cirri, Antonio Corradi, Luca Foschini, and Dario Maio. 2009. MSF: An Efficient Mo- bile Phone Sensing Framework. *International Journal of Distributed Sensor Networks* 2013, 538937 (2009), 9. DOI:http://dx.doi.org/10.1155/2013/538937
37. Cory Cornelius, Apu Kapadia, David Kotz, Dan Peebles, Minho Shin, and Nikos Triandopoulos. 2008. Anonysense: Privacy-aware People-centric Sensing. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*. ACM, New York, NY, USA, 211–224. DOI:http://dx.doi.org/10.1145/1378600.1378624
38. Clayton Shepard, Ahmad Rahmati, Chad Tossell, Lin Zhong, and Phillip Kortum. 2011. LiveLab: Measuring Wireless Networks and Smartphone Users in the Field. *SIGMETRICS Perform. Eval. Rev.* 38, 3 (Jan. 2011), 15–20. DOI:http://dx.doi.org/10.1145/1925019.1925023
39. Tathagata Das, Prashanth Mohan, Venkata N. Padmanabhan, Ramachandran Ramjee, and Asankhaya Sharma. 2010. PRISM: Platform for Remote Sensing Using Smartphones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*. ACM, New York, NY, USA, 63–76. DOI:http://dx.doi.org/10.1145/1814433.1814442
40. Best practices for an authentication system by IEEE Center for Secure Design - https://cybersecurity.ieee.org/blog/2016/06/02/design-best-practices-for-an-authentication-system/