

Specific Targeted Research Projects (STReP)

SOCIOTAL

Creating a socially aware citizen-centric Internet of Things

FP7 Contract Number: 609112



WP4 – Citizen Empowerment

Deliverable report

Contractual date of delivery:
M30 – February 2016
Actual submission date:
14/03/2016

Deliverable ID:	D4.3
Deliverable Title:	Beta release of integrated SocloTal platform
Responsible beneficiary:	DNET
Contributing beneficiaries:	DNET, UNIS, CRS4
Estimated Indicative Person Months:	10

Start Date of the Project: 1 September 2013 Duration: 36 Months

Revision: Final
Dissemination Level: Public

PROPRIETARY RIGHTS STATEMENT

This document contains information, which is proprietary to the SOCIOTAL Consortium. Neither this document nor the information contained herein shall be used, duplicated or communicated by any means to any third party, in whole or in parts, except with prior written consent of the SOCIOTAL consortium.



PROPRIETARY RIGHTS STATEMENT

This document contains information, which is proprietary to the SOCIOTAL Consortium.
Neither this document nor the information contained herein shall be used, duplicated or communicated by any means to any third party, in whole or in parts, except with prior written consent of the SOCIOTAL consortium.

Document Information

Document ID: SOCIOTAL_D4.3v0.9.doc
Version: V0.9/Final
Version Date: 14. March 2016
Authors: Nenad Gligoric, Antonio Pintus, Alberto Serra, Andrea Manchinu, Colin O'Reilly, Niklas Palaghias
Security: **Confidential**

Approvals

	Name	Organization	Date	Visa
<i>Project Management Team</i>	Klaus Moessner	UNIS		
<i>Internal Reviewer</i>	Ignacio Elicegui Maestro	UC		
<i>Internal Reviewer</i>	Jorge Bernal Bernabe	UMU		

Document history

Revision	Date	Modification	Authors
0.1	7/10/2015	TOC, structure and initial assignments	DNET
0.2	8/11/2015	Executive summary and section 1	DNET
0.3	15/12/2015	Section 2 added	CRS4
0.4	10/01/2015	Additional content	DNET
0.5	12/01/2015	Introduction for the section 2	CRS4
0.6	08/02/2016	Addition to deployment architecture in Section 3	UNIS
0.7	10/02/2016	Update to the webEnv in section 2.1.1	CRS4
0.8	19/02/2016	Added integration details for the mobEnv in section 2.1.2	DNET
0.9	14/03/2016	Final versions	UNIS

Content

Section 1 - System Architecture and Technical Solution Overview	7
1.1 Context Manager	8
1.2 Trust Manager	10
1.3 Authentication	11
1.4 Identity Management (IdM)	11
1.5 Authorization	11
1.6 Group Manager	12
1.7 Communities Manager	12
1.8 Enablers	13
Section 2 - User Environment	16
Section 3 - Deployment architecture	31
3.1 Software (list of components)	31
3.2 Software repository and Wiki knowledge base	32
Section 4 - Conclusion	33
Section 5 - References	34

List of Figures

Figure 1: Architecture of the SocloTal Integrated Platform	8
Figure 2: SocloTal Context Manager Architecture	9
Figure 3: The SocloTal User Environment Architecture	17
Figure 4: User Environment integration with Security framework and Context Manager	18
Figure 5: Web UserEnv home page, the dashboard providing an overview of the user's workspace.	20
Figure 6: the user profile page.	21
Figure 7: the list of the Channels created by the logged user.	21
Figure 8: Channel details page	22
Figure 9: The user's registered smartphones list for a user	23
Figure 10: The UI to configure a Channel connection.	23
Figure 11: Integration of the Web UserEnv with other SocloTal components targeted to security: the sequence diagram shows all the involved (integrated) steps from users' sign up to SocloTal resources access.....	25
Figure 12: The Web User Environment login step	26
Figure 13: Interaction between User and Context Manager	27
Figure 14: Interaction between Mobile Environment and Context Manager	27
Figure 15: Interactions between user and Identity Manager	29
Figure 16: Interaction between MobEnv and Context Manager	29
Figure 17: Secure transmission from the Face-to-Face enabler to the Context Manager.....	30

Executive summary

The main objective of this document is to provide an overview of the technical integration of the SocloTal beta platform release, more specifically design description and integration of the developed SocloTal security mechanisms into the users' workspace. The user workspace is based on Mobile and Web User environment integrated with mechanisms and techniques developed in the WP2-3 in order to enable following functionalities:

- Automatic establishment of trust relationships and reputation of by leveraging Trust Manager framework and different enablers
- Privacy framework for privacy-preserving context-sensitive communication to avoid leakage of data that user generates and shares with peers in a particular group and to manage the identity of IoT devices and associated attributes based on encryption techniques allowing the secure communication of data within a different groups.
- A secure communication framework, based on attribute-based encryption, cryptography and key management.

These algorithms and mechanisms are integrated and combined into the working platform that will be evaluated during two pilot evaluation rounds and Hackathon event planned to be held during IoT Week 2016 in Belgrade, Serbia.

The overview of the System Architecture and Technical Solution is provided in **Section 1** with description of the Content Manager and the Security components of the SocloTal Beta platform. **Section 1** includes functionalities of the Trust Manager, information about authentication mechanisms. The Identity Management component description, The SocloTal Authorization systems overview, Group Manager and its responsibilities based on the Ciphertext Policy Attribute Based Encryption. Furthermore, Communities Manager organisation and its implementation is explained in the SocloTal platform, and objects the Communities Manager work with. Moreover, **Section 1** provides details about Enablers describing their importance to the SocloTal beta release, more specifically, Face-to-Face Enabler, and Gait Recognition Enabler.

In addition to the SocloTal beta release, in **Section 2**, the SocloTal User Environment is described and focused on details about Web User Environment, which is web-based environment, and Mobile User Environment, Android application, and visualisation of its interface and integration with the SocloTal security framework.

Information about list of components, repository and wiki page can be found in **Section 3**. **This Section omits the implementation details as they are already described and covered in the SocloTal wiki [1].** Finally, the conclusion of this deliverable is provided in **Section 4**.

The final purpose of this deliverable is to provide an implemented beta solution for SocloTal, including overviews and descriptions of core components and their integration into the platform. In addition, component correlation with each other and with SocloTal Context Manager is explained.

Section 1 - System Architecture and Technical Solution Overview

The SocloTal Integrated Platform includes standardized protocols for secure communications that enables authentication, authorization, confidential data sharing, and identity management through different techniques, i.e. DTLS, Idemix, XACML and CP-ABE cryptographic, modified and adapted to the purposes of SocloTal.

As shown in Figure 1, architecture design of the system, which defines the structure and the behaviour of the SocloTal beta, can be described by three main components:

- Application level components
- Core components
- Communication Layer

Application Level Components includes, SocloTal Services, Identity Selector, Application, Publish Subscription mechanism and Personal Cloud Client.

Context Manager is centralised component of the system (Figure 1), with the core components of the system architecture like security components, Management, Service organisation, Device-centric enablers which are described later in this section (Section 1.8), also IoT service and IoT Business Process Management are components of the Core Layer of the SocloTal beta release along with Security components.

Furthermore, integration of the communication between components of the system is described in the Communication layer. This layer is in charge of the energy optimisation, flow control, error detection. Moreover it provides gateway services and also mobility, routing and anonymous channels.

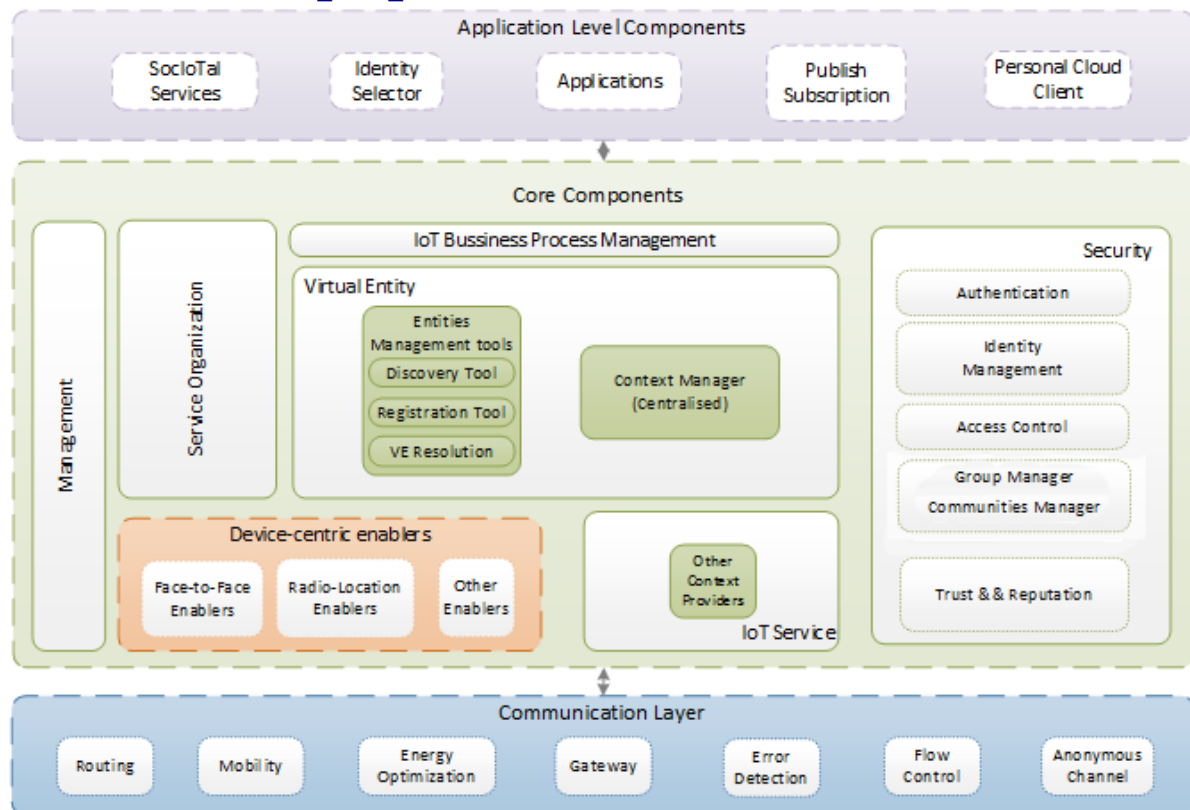


Figure 1: Architecture of the SocioTAL Integrated Platform

1.1 Context Manager

The SocioTAL centralized Context Manager covers two main functionalities of the integrated SocioTAL Platform: it provides a resource directory where all the available (and operative) context entities are listed and stores all context information related to these entities and associated enablers. It also implements NGSi9 and NGSi10 [2] based RESTful APIs although those standards are not fully supported) that manage resources and grant access to context information. Figure 2 presents this Context Manager architecture, including all the sub-components built-in. Every element of this architecture is described in D3.2.1 [3].

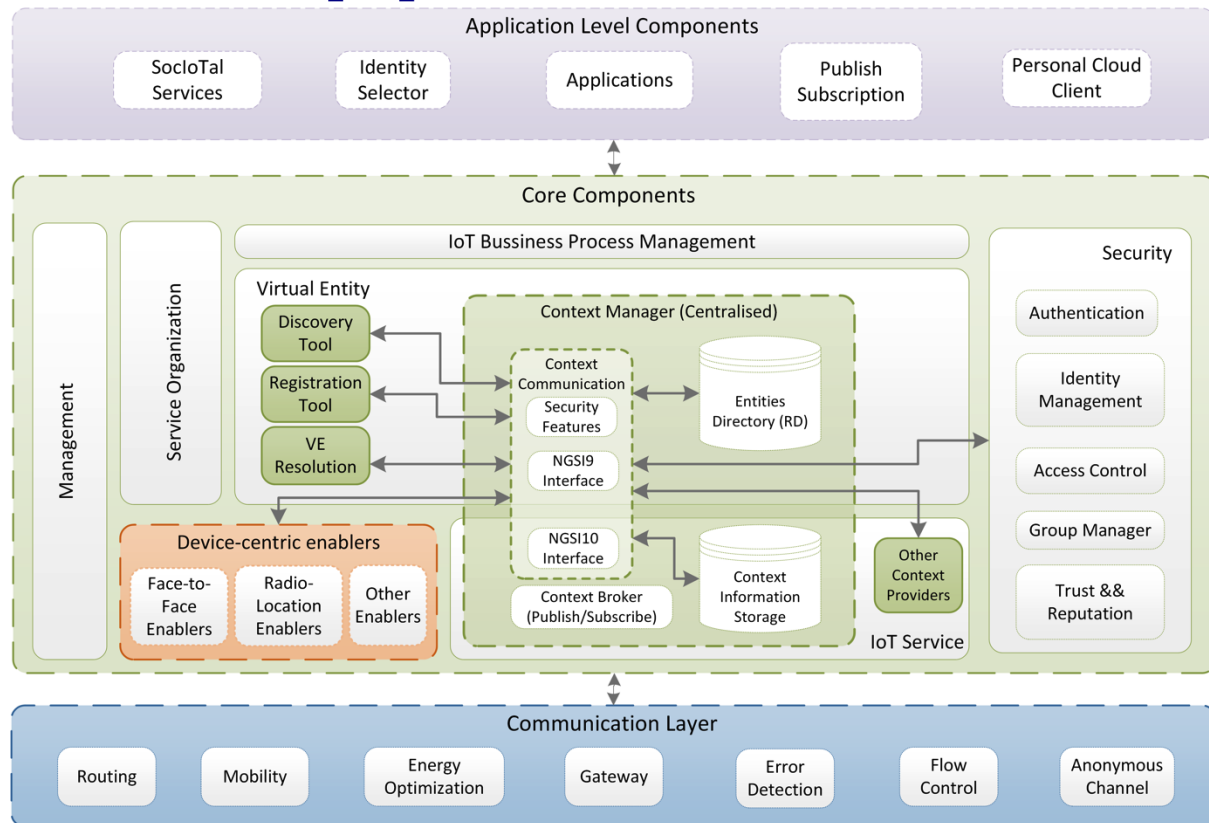


Figure 2: SocloTal Context Manager Architecture

The first implementation of SocloTal Context Manager has been built over selected FIWARE platform, using an instance of its Orion CB [4] as the inner SocloTal Context Broker of its Context Manager. It also supports MongoDB storage to implement SocloTal CM Context Information Storage and Entities' Directory.

Its **NGSI9 based** interface supports registering, updating and Context Entities discovering, including their related attributes and availability. The **NGSI10 based** interface is oriented to register, update and discover Context Information, providing the attributes' values and defining the associated metadata. A set of "extended" methods provide easier ways to create, manage and query context entities and information.

This way, the SocloTal Context Manager acts as the integration point for context information, sharing all the context information uploaded by the registered entities and SocloTal enablers and notifying (by subscriptions) when new interesting data and/or resource is available.

1.1.1 Security

Every request sent to SocloTal Context Manager must contain a **Capability-token** header (as part of all the HTTP required headers). This header will include, in JSON format, the needed data (credentials) to check the identity and policies that allows access to the requested resource. If the requestor identity and/or credentials do not match with the specified resource, the SocloTal Context Manager will return an "Unauthorised" message (Error 401).

An Example of Capability Token is the following:

```
{
  "id": "rt19kt9g31rnr904tssuemn1",
  "ii": 1457359559,
  "is": "capabilitymanager@um.es",
  "su": "6c0a3e29-756d-46c0-b943-7913c5b4d3fd",
  "del": "MIIBIjANBgkqhki...",
  "de": "SocIoTal:CRS4:WeatherStation:CRS4_WEATHER_STATION",
  "si": "EmGJ9KNoBGqbJrqNTwI...",
  "ar": [{
    "ac": "queryContext",
    "re": "*"
  }],
  "nb": 1457359559,
  "na": 1457370559
}
```

by this Capability Token the UserEnv is able to perform a request to the Context Manager to retrieve the entity identified by the *SocIoTal:CRS4:WeatherStation:CRS4_WEATHER_STATION* id. The "su" field identifies the Authorization Token provided by the Identity Manager (IdM, see Section 11). This token is used by the Capability Verifier to allow the user to have access or not to the particular resource in the Context Manager.

The Capability Token JSON fields and the overall access control flows over several parameters such as identification ("id"), time of token creation called issued time ("ii") issuer or subject who issued the token ("is") further information contains information about subject who can access token ("su"), device information linked to the token ("de") and digital signature for the token ("si"). Moreover, access rights are defined in "ar" field, with information about action "ac", information needed from the device ("re") and conditions for the device ("co"), "nb" and "na" defines time frame in which token must be accepted. More information including how to obtain a Capability Token can be found in the D2.2 [5].

1.2 Trust Manager

The Trust Manager is a central component for the quantification of the reputation score by utilizing input from the SocIoTal reputation enablers: the face-to-face (F2F) enabler (Section 1.8.1), and user behaviour (section 1.8.2).

The Trust Manager is a REST based component with logic that consumes a set of different rules as an input for building a reputation score. Generic model for rules enables mapping between provided JSON format and relational database for mining and extraction of rules previously added over a registration API. The crucial component that the Trust Manager utilizes to continuously maintain the updated version of score in respect to last attribute value changes is the Context Manager.

The main interaction of the Trust Manager with other components can be visualised moreover by explaining the process of reputation quantification starting from the device registration as follows: user registers a set of rules for an application, with JSON in predefined structure POST-ed to the Trust Manager REST endpoint (1). After the registration, Trust Manager automatically extracts for the first time attributes value from the Context Manager (2), computes (3) and POSTs the final reputation score to the Context Manager (4). Rules are saved in the local database of the Trust Manager. The Trust Manager then subscribes to the attributes' value changes (4) in the Context Manager. When value of an attribute is changed (5), Context Manager pushes an updated value to the TM

(6). Logic of the Trust Manager then checks for what context this attribute is used to build the reputation by querying the database for the rules (7), recalculates the score (8) and pushes it back to the Context Broker (9). Each call between these main two components is using token generated using security framework.

The Trust Manager is subscribed to all attributes changes used for quantification of the reputation score, keeping the score constantly updated.

1.3 Authentication

SocioTal proposes different authentication mechanisms, according to the component, enabler or application used. Thus, the web user environment uses login-password while other components and apps uses authentication based on Public Key Infrastructure (PKI) through certificates [7]. SocioTal also provides an alternative and more sophisticated way of performing authentication ensuring, at the same time, privacy and minimal disclosure of attributes. Thus, this kind of alternative privacy preserving way of authentication using anonymous credential system is handled in the SocioTal framework by the Identity Management Component (IdM [8] The IdM verifier module is able to verify partial identities derived from an obtained credential to authenticate the user holding such a credential. The SocioTal platform also support login-password authentication by relying on the SocioTal Fiware keyrock client library, which allow interacting with Keyrock to authenticate users registered on it.

1.4 Identity Management

The Identity Management component of the SocioTal security framework is an anonymous credential system that ensures user privacy and minimal disclosure of personal information when accessing IoT services. It is based on Idemix [9] and it is integrated with the Fiware IdM “Keyrock” [10]. The IdM is mainly focused for deployments in smartphones, providing an Android app that allows managing the credential obtained from the IdM Issuer, manage partial identities derived from the credential and use the partial identities to access to IoT services. The IdM has been integrated with the Authorization component to obtain authorization tokens based presenting the partial identity. The IdM is also being evolved in interaction with the Context Manager so it is able to select the proper partial identity to use according to the required context. The SocioTal IdM has been also integrated in the Communities Manager, the Web user environment, the Attribute Authority and the Sociotal Mobile User environment.

1.5 Authorization

The SocioTal Authorization system follows a policy-based approach relying on capability tokens. A subject entity gets a capability token in order to get access data from a target device. This token is usually generated by a Capability Manager entity which takes access control decisions that are embedded into the token [5]. The authorization system uses a lightweight policy engine which gathers the context and trust scores to take authorization decisions accordingly. During this last year it has been developed the Policy Decision Point (PDP), the Policy Administration Point (PAP), the Capability Manager Server and the Android Client. The Authorization system has been integrated with the IdM, together with other SocioTal components, enablers and tools that use the capability tokens, such as the web user environment and the Context Manager. This last one acts as policy enforcement point that verifies the authorization tokens prior allowing access to the resources stored in the Context Manager. In addition to the authorization services (Manager, PDP and PAP), SocioTal has also developed two libraries to deal with authorization. On one hand, the

Capability Client allows performing requests to the Capability Manager to obtain capability tokens, which are used to perform actions over entities that are registered with the Context Manager. On the other hand, the Capability Evaluator library aimed to validate capability tokens prior allowing the access to the target resource.

1.6 Group Manager

The Group Manager component of the SocloTal security framework is based on the use of the Ciphertext Policy Attribute Based Encryption (CP-ABE) [11] cryptographic scheme in order to enable a secure data sharing mechanism with groups of entities (i.e. communities and bubbles of smart objects). The Group Manager module allows the data producer to share information with groups of entities which meet certain combinations of identity attributes.

After the information is encrypted and disseminated by the producer entity, the Group Manager of a target entity (acting as a consumer) will try to decrypt it with CP-ABE keys related to its identity attributes through its CP-ABE engine.

A group manager sharing app for Android has been developed in the scope of the project, which allows sharing information within groups preserving confidentiality. The Group Manager has been integrated with the Context Manager since data communication can be done disseminating the encrypted data through the Broker of the Context Manager. Thus, subscribed target entities in the group can access the shared information and decrypt the data as long as they have the proper CP-ABE keys. The group sharing app allows sharing data between users and the devices belonging to a bubble. The bubble should have been previously defined in the context manager (using the Web User Environment). The bubble includes the devices as well as the identity attributes that will be needed by the entities (users) that can implicitly be part of the bubble. It is achieved by the fact of holding the identity attributes (and therefore the cryptographic CP-ABE keys) that satisfy the policy to decrypt the shared data in the scope of the bubble.

1.7 Communities Manager

SocloTal's Communities Manager [7] component provides a set of centralized tools to create and manage groups of users and resources as well as controlling the access to these groups. Based on FIWARE IdM 3 [10], it will build a centralized Users' directory and a Communities Registry, all of them organized within domains. This way, Communities Manager will work with: Domains, Users, Communities and Tokens

- Domains: a set of classified and isolated users and communities. A domain can, for example, group all members and communities belonging to a city, project or application, differentiating them from other domains elements. For these first versions, domains will be managed by the centralized SocloTal Integrated Platform instance administrator.
- Users: identifies a user entity within the SocloTal Communities framework, keeping the credentials (names, passwords, roles and tokens) needed to be authenticated within the Communities Manager. The same user name can be utilized within different domains.
- Communities: identifies groups of users and resources within the same domain. Unlike the domains, communities within a domain could be created and managed by users belonging to it, fostering users' data sharing
- Tokens: provided by the Communities Manager and requested by a user, this UUID (Universal Unique Identifier) will identify the requestor user (previously registered within the Communities Manager) and the community it belongs to, providing also

extra related information such as the role the requestor has and its validity (as well as its expiring date)

Initially, the Communities Manager will be integrated within the SocloTal Platform through the provided token: called “Community-Token”, which will be added to the selected method request (among the different provided SocloTal components APIs) as a header. The SocloTal component, in turn, will read the Community-Token and check, against the Communities Manager, the requestor’s identity, its role and the community it belongs, besides the validity of the token and then, execute or not, the requested operation.

1.8 Enablers

The enablers in the Integrated Platform provide information about the user obtained from the environment they are in. For example, the face-to-face enabler provides information about the social interactions of the user. The information generated by the enablers are forwarded to the Integrated Platform through the Context Manager. The enablers communicate with the Context Manager using a RESTful communication protocol, allowing an exchange of information in a structured manner. The shared information is presented in a JSON format. The communication protocol is based on OMA, in particular with NGSi9 for context registration and NGSi10 for publishing and subscribing to context. Each enabler creates a context entity in the Context Manager based on the information it provides.

1.8.1 Face-to-Face Enabler

Social interaction is an important component of humans’ social behaviour. In order to understand social interaction, we have developed the Face-to-Face (F2F) enabler. The F2F enabler is focused on off-the-shelf mobile phones and is an opportunistic, collaborative and non-intrusive mobile app that detects the on-going social interactions. The inference process of social interactions is based on users’ interpersonal distance and relative orientation. When users are detected in vicinity, then the enabler estimates their interpersonal distance based on a machine-learning model for Bluetooth Received Signal Strength Indicator and computes the relative orientation of the users based on their facing directions. Users’ facing is estimated independent of the device’s wearing position and based on the users’ walking locomotion the users’ torsos direction is estimated. To exchange the facing direction among the devices, a collaborative sensing technique was developed. Having estimated the users’ interpersonal distance and relative orientation, an overall inference is performed to understand the existence of a social interaction.

The social interaction inference is performed on-line on the device and does not require any additional hardware. Apart from the social interaction inference, the F2F enabler estimates also the social relation of people based on their interpersonal distance.

The F2F enabler determines the social interaction between users of the SocloTal Integrated platform using an app installed on their mobile phone. The face-to-face enabler provides the following information.

- The detected nearby SocloTal devices
- The result of inferring if the nearby users are participating in a real-world interaction.
- The social relation of nearby users
- The time-stamp of detected real-world social interaction.
- The location that the detected real-world social interaction occurred.

This is communicated to the Context Manager as detailed in the previous section. The SocloTal platform includes various enablers and components that communicate through the Context Manager. The Context Manager constitutes the core component of the SocloTal platform. The communication among these components is performed through a RESTful transmission protocol, where all the enablers publish/subscribe their context entities to the Context Manager. The information is modelled through the NGSI-9/10 data model and is represented through the JSON light-weight data-interchange format. Finally, an example of the F2F context entity is described below:

Call	193.144.201.50:3500/SocloTal_Context_UC_REST/NGSI10_API/queryContext/
Response JSON:	<pre> { "contextElement": { "id": "SocloTal:UNIS:SmartphoneContext:VirtualSmartphoneContext_002", "attributes": [{ "name": "F2FInteraction", "value": "false", "type": "boolean", "metadatas": [{ "name": "DiscoveredDevice", "value": "Nick?s MacBook Air", "type": "http://sensorml.com/ont/swe/property/pseudonym" }, { "name": "SocialRelation", "value": "PUBLIC", "type": "string" }, { "name": "DateTimeStamp", "value": "20150317T134409Z", "type": "http://sensorml.com/ont/swe/property/DateTimeStamp" }, { "name": "Location", "value": "-0.58823666, 51.24346692", "type": "http://sensorml.com/ont/swe/property/Location" }] }], "type": "urn:x-org:sociotal:resource:device", "isPattern": "false" } } </pre>

Table 1. Example of F2F context entity

For further information about F2F enabler can be found in the D2.3 [6] and more information about performance of the F2F enabler can be found in the D3.2.1 [3].

1.8.2 Gait Recognition Enabler

It is important to determine the authenticity of data providers in order to allow trust of the data that they generate. One way to determine authenticity is for users to authenticate themselves when they access services on a smartphone. A common way to perform this is to enter a password, use a swipe pattern, or use a fingerprint. A drawback on these approaches is that they only authenticate a user when they are using the smartphone. It is possible that a smartphone is providing data while it is not actually being used by the user, an example of this is providing location information. Alternative methods are required in order to authenticate data when the smartphone is not in use.

Biometrics, such as fingerprints, are a common way to authenticate users. Alternative biometrics which exploits human characteristics and raising trend of the sensor availability in modern smartphones. One such method is gait recognition, where the user is authenticated on their walking pattern. The Gait Recognition enabler aims to determine if the smart phone is currently being used by the authorized user by analysing their walking pattern. A more detailed analysis of the operation of the Gait Recognition enabler can be found in Deliverable 2.3 [6] with further performance analysis provided in Deliverable 5.2 [12].

The Gait Recognition is performed online on the mobile phone and does not require the communication of data to a backend server. Initially, a model is constructed of the user's normal walking pattern. Future walks are compared with this in order to determine whether the same user, or a different user, is in possession of the smartphone. The Gait Recognition enabler was implemented in Android, but the concept is not specific to this operating system. In addition, the algorithm for the Gait Recognition enabler does not require any additional specific hardware, unlike authentication with a fingerprint. The algorithm uses the accelerometer sensor to measure the walking pattern of users. This is a sensor which most modern smartphones have and is used by a variety of applications.

The Gait Recognition enabler communicates with the Context Manager as detailed in the previous section. The result of the evaluation, Rightful User or Imposter is communicated to the Context Manager. This information is then used by the Trust Manager as one of the metrics in the calculation of the trust and reputation score.

Section 2 - User Environment

The specific objectives targeted by the task T4.1 and task T4.3 related to the SocloTal User Environment and integration with WP2 and WP3 outcomes are the following:

O4.1: To define and develop tools that will provide intuitive mechanisms to the user for expressing the way in which they want their (IoT) environment behave;

O4.3, to provide integration of trust/reputation and privacy-preserving communication mechanisms inside the user environment alongside social circles and social media;

The plan for Y3 at M30 was to provide a beta version of the User Environment (Web and Mobile), including the integration with other components and enablers from work packages WP2 and WP3.

Figure 3 shows the updates on the high level architecture of the beta version of the User Environment.

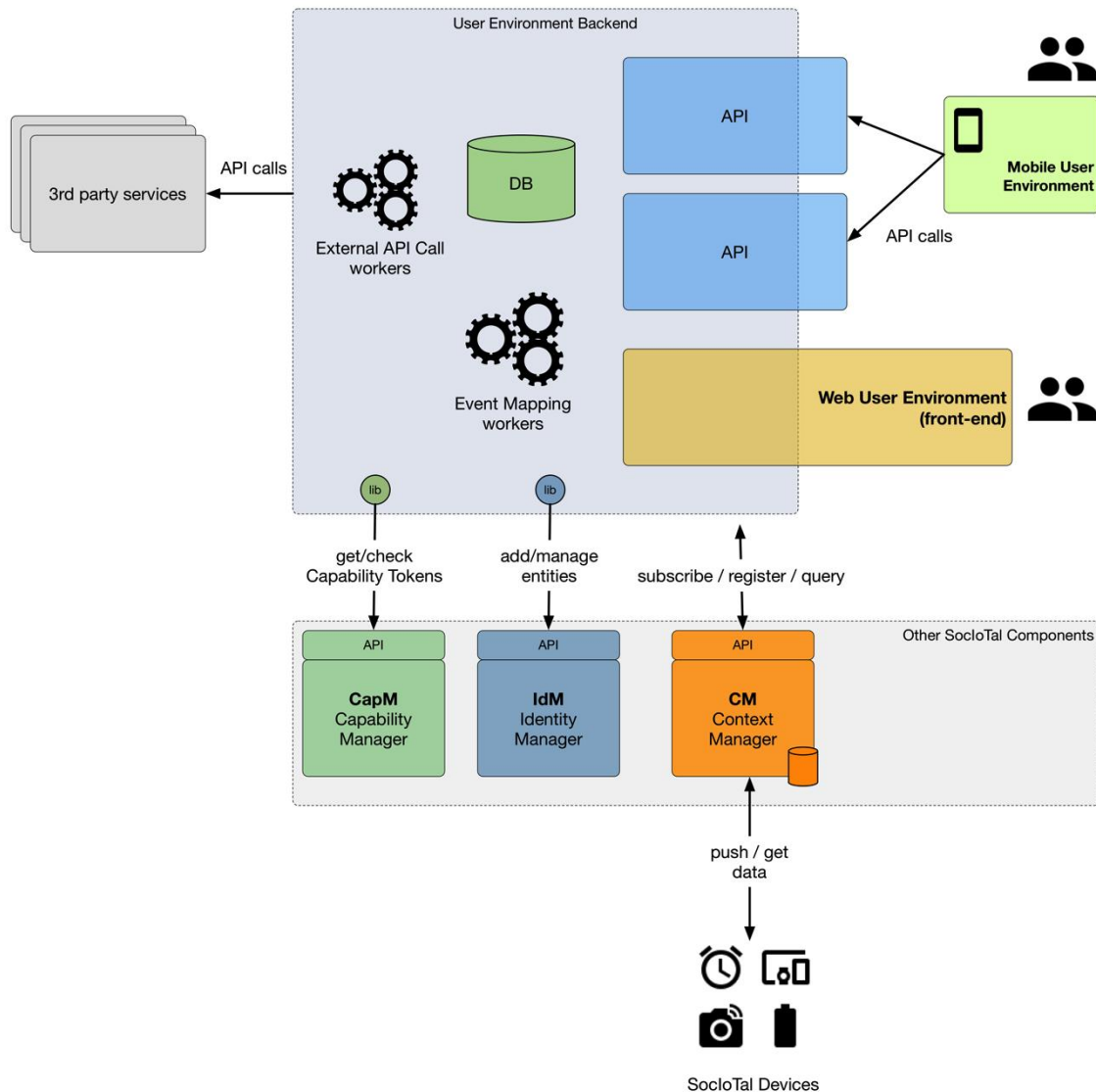


Figure 3: The SocioTal User Environment Architecture

SocioTal user environment is composed of the **Web User Environment (WebEnv)** and **Mobile Environment (MobEnv)**, currently available for Android.

User and Mobile environment are using Capability Manager to obtain capability token used in the service request call. In addition, Mobile environment can access resources from User environment.

Bubble creation/management enables creation of bubbles and communities for authorized users that belong to a same circle of trust. If the user is not authorized, he cannot join the bubble and access the data belonging to the bubble.

When the user wants to access resources registered in the Context Manager he must authenticate his user environment against SocloTal. Identity Manager is used to obtain the credentials which are then employed in the service request call as shown in Figure 4.

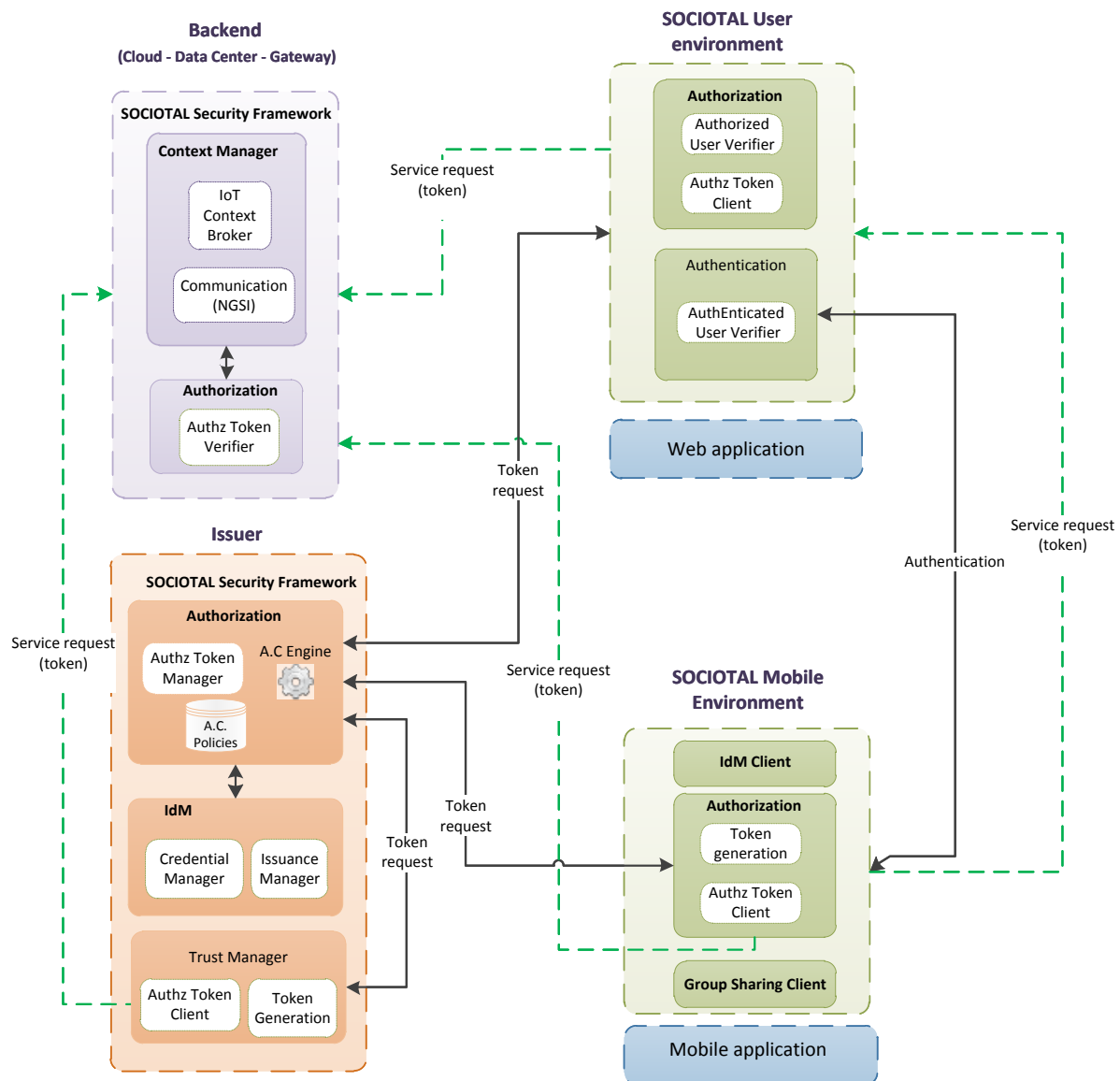


Figure 4: User Environment integration with Security framework and Context Manager

To avoid confusion about the tokens and its purpose, it is important to understand there are two types of token in the SocloTal platform:

1. User token generated by the Web User Environment that is used for authentication from the mobile environment
2. Service token generated by the Security framework and used for authorization when accessing resources in the Context Manager

The integration of the security framework into the user environment is further separately explained in the following sections, presenting the integration of the web and mobile environment with the overall privacy tools and mechanisms.

2.1.1 Web User Environment (WebUserEnv)

As shown in Figure 3, the Web User Environment is composed of two high-level architectural components: the **User Environment Backend** and the **Web User Environment (front-end)**, this section reports both.

User Environment Backend (UserEnv): is the core server-side architectural component of the SocloTal User Environment. It includes and orchestrates all the required back-end services in order to provide:

- Web APIs for developers;
- Message routing from one endpoint to another (e.g., from an API endpoint of a SocloTal Device to a API endpoint of a third party cloud service);
- Integration with the other involved SocloTal components (e.g. those concerning security).

The User Environment introduces and uses two main abstract entities and concepts:

Channel: a logical link to a SocloTal Device or an external third -party service. Basically, a Channel allows getting data from a device or sending data to it. For example, a Channel could be linked to (represents) a SocloTal deployed Weather Station or a Xively data stream. Each category of devices or services to be managed in the UserEnv will have a corresponding Channel.

Connection: a logical link between two Channels in the User Environment. A Connection between a Channel A and a Channel B can be configured, specifying a *trigger* (a condition on data produced by Channel A) and a correlated *action* to perform on Channel B when the trigger is activated. For example, a Connection could be configured to act like this rule: *WHEN temperature on Channel A > 25 DO Send a “temperature alert” to Channel B.*

The User Environment Backend also includes other modules to manage the required features, and they are:

Event Mapping Workers: submodules that collect events from event sources (Channels) and look up actions that must be invoked according to trigger/action rules.

External API calls Workers: submodules responsible to call all the third party services, like Xively Data Streams.

Web User Environment (front-end): is the front-end dashboard, designed for desktop and tablets. The Web front-end communicates with various server side components and provides a user interface in the web browser targeted to end users. This component exposes the following functionalities:

- Tools to create and manage Channels;
- Tools to manage, search and to subscribe to data events triggered by Devices deployed in the SocloTal Context Manager;
- Tools to create and manage connections between Channels and to set trigger/action rules;
- Will support data views and anomaly detection (not included in the Beta version of the software);

The current Beta version of the Web User Environment has been developed, re-designed and improved following the feedback came from users, developers during demo and evaluation sessions performed during several events and meetings.

Feedback and recommendations not only allowed us to fix several bugs but also to greatly improve the User experience, in particular targeted to citizens as final (not experts) users, as WP4, task T4.1 required.

More in details, the beta version of the Web User Environment sports:

- A revamped, modern, responsive and consistent Web UI design, including better information and explanations for final users, feedback and a more user-friendly management of Channels, Devices, Connections and Smartphones;
- Improved API thanks to several bugs fixed;
- The whole Web User Environment now is secured running under HTTPS protocol;
- Noticeable performance improvement, thanks to the re-design of some internal components.

Next set of pictures show some screenshots about the Beta version of the Web User Environment.

Figure 5 shows the user personal dashboard page. The dashboard summarizes the Channels created by the user, the number of available Devices registered in the SocloTal Context Manager and the smartphones the user registered. Using the buttons and the menu, the user can add and manage all the entities in her workspace.

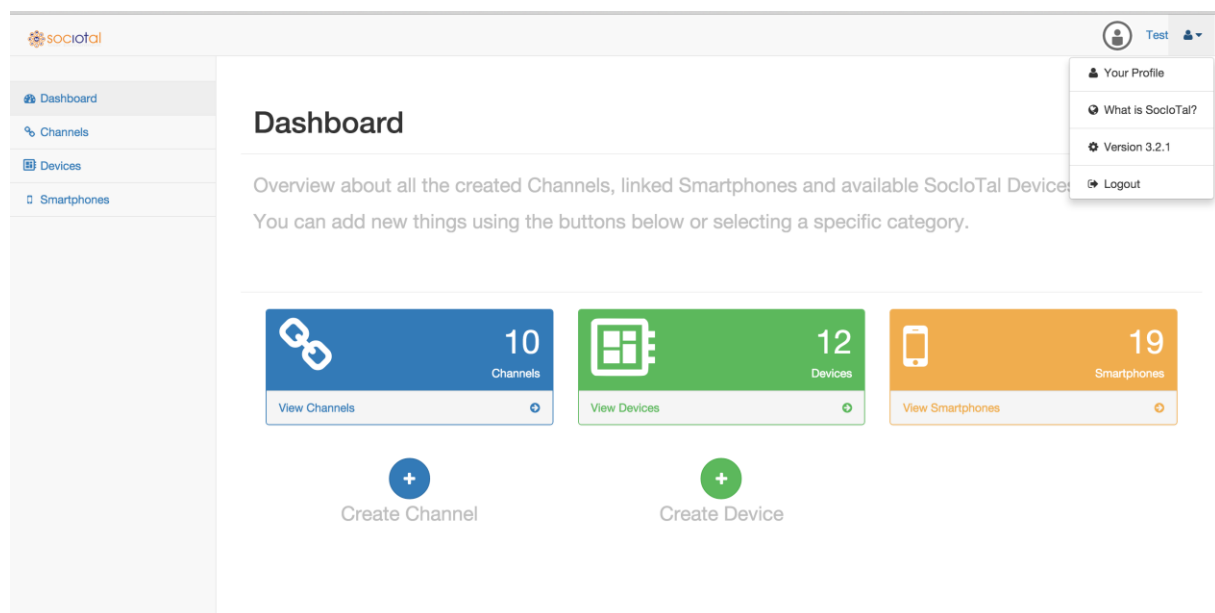


Figure 5: Web UserEnv home page, the dashboard providing an overview of the user's workspace.

Figure 6 shows the user profile page containing basic set of account information and the API Key to be used in API calls in case the user is also a developer. User can also delete its account.

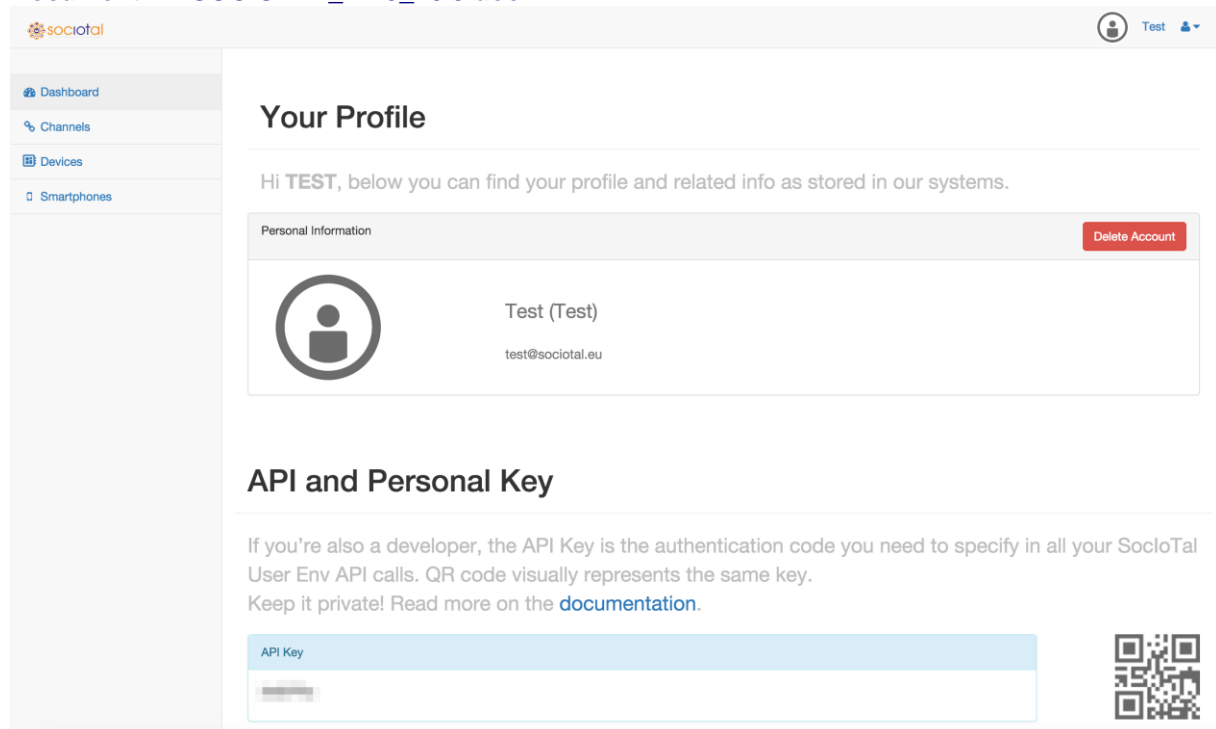


Figure 6: the user profile page.

Figure 7 reports the Channels page. Channels are logical links to Devices registered in the platform. Through Channels user can gather data from Devices or build simple trigger/action based personal applications.

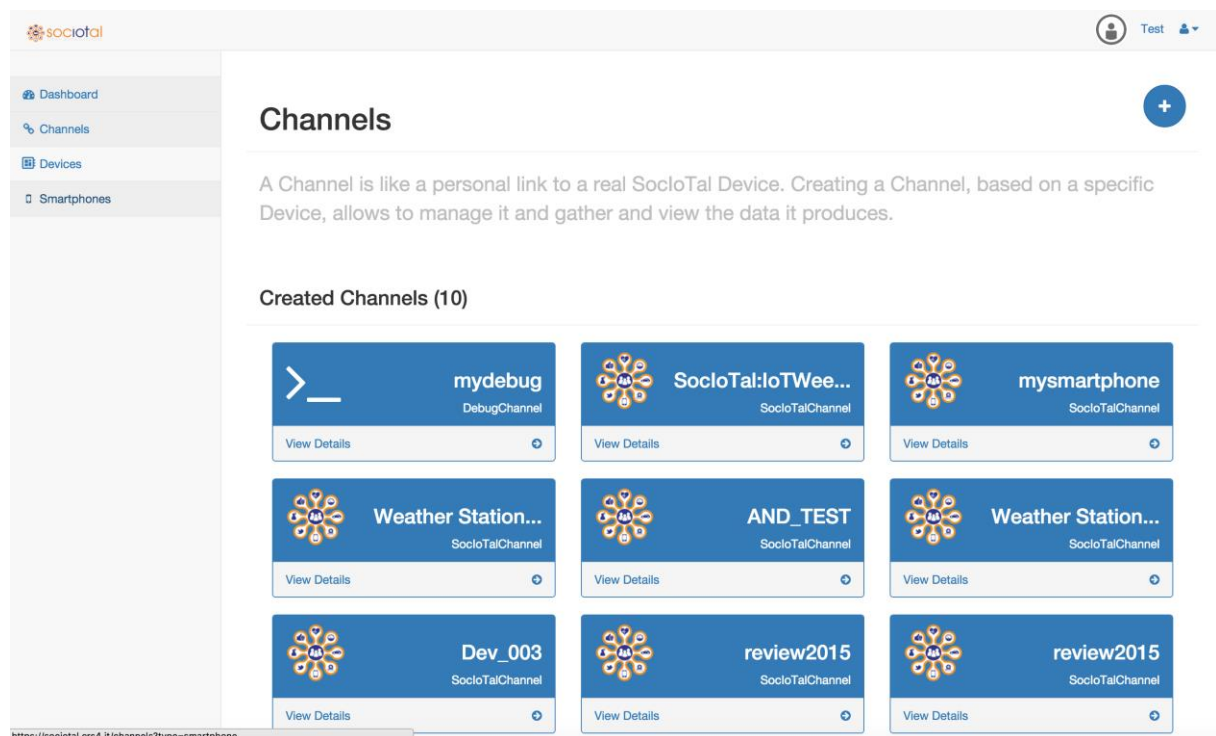
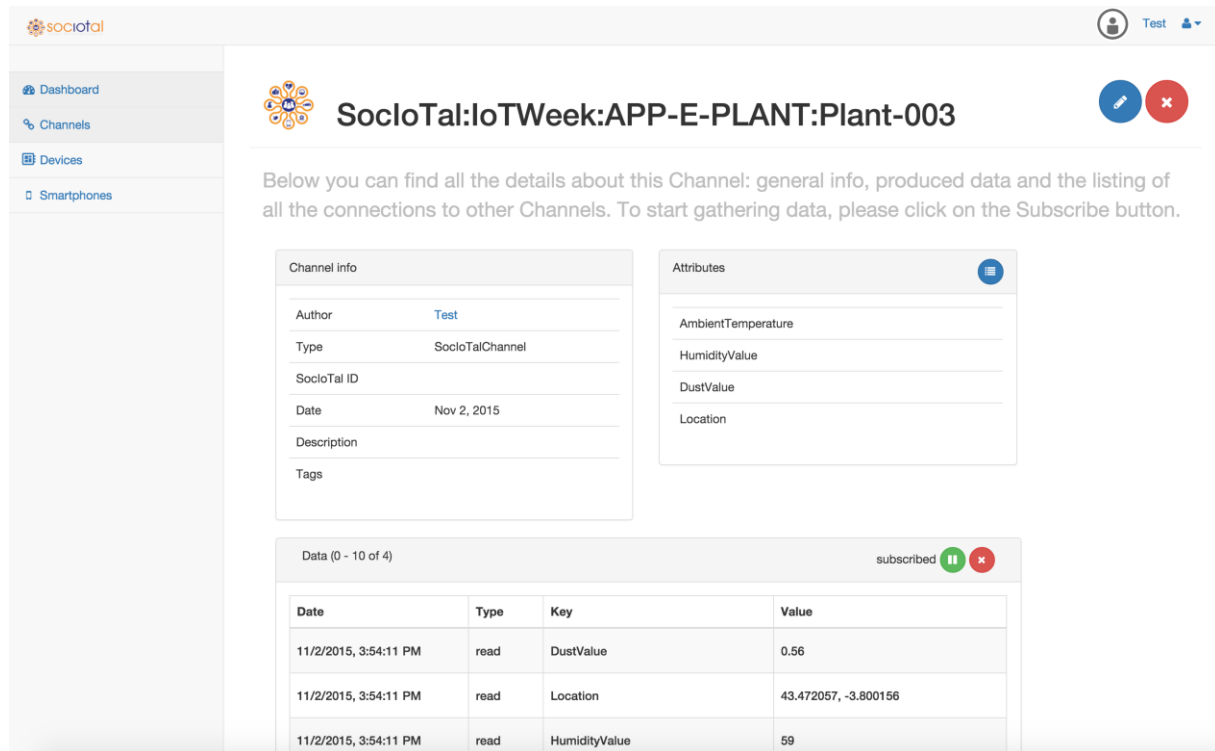


Figure 7: the list of the Channels created by the logged user.



SocioTal:IoTWeek:APP-E-PLANT:Plant-003

Below you can find all the details about this Channel: general info, produced data and the listing of all the connections to other Channels. To start gathering data, please click on the Subscribe button.

Channel info

Author: [Test](#)

Type: SocioTalChannel

SocioTal ID

Date: Nov 2, 2015

Description

Tags



Attributes

AmbientTemperature

HumidityValue

DustValue

Location

Data (0 - 10 of 4) subscribed  

Date	Type	Key	Value
11/2/2015, 3:54:11 PM	read	DustValue	0.56
11/2/2015, 3:54:11 PM	read	Location	43.472057, -3.800156
11/2/2015, 3:54:11 PM	read	HumidityValue	59

Figure 8: Channel details page

Figure 8 reports a page for a particular Channel. The Channel is subscribed to the linked Device in order to gather data, which is shown to the user as it comes from the real Device registered in the SocioTal Context Manager. The Channel page also shows all the details and gathered data and allows managing and editing it. In the Figure 9 is representation of the all user's registered smartphones.

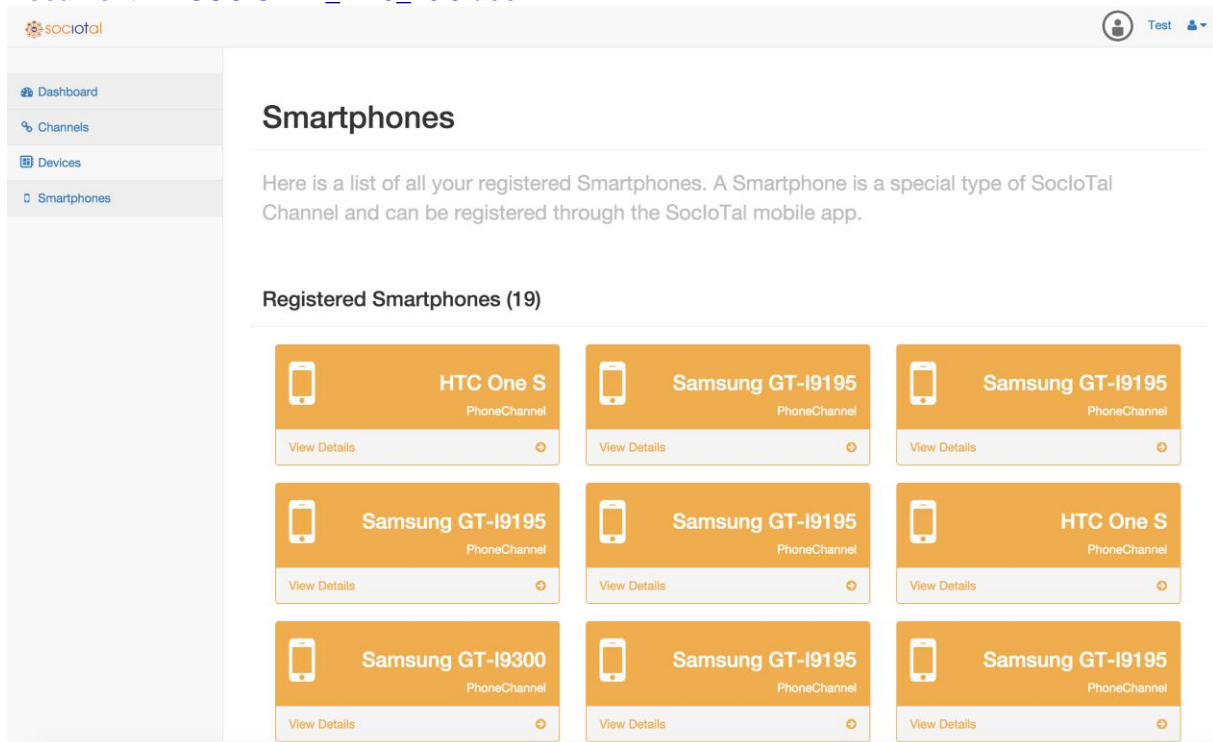


Figure 9: The user's registered smartphones list for a user

Figure 10 shows the user interface that assists the user to create a connection between two Channels. In the reported example a Xively Channel has been connected to a Smartphone. Thus, it will receive a push notification whenever the temperature coming from the Xively Data Stream is greater than 60 degrees.

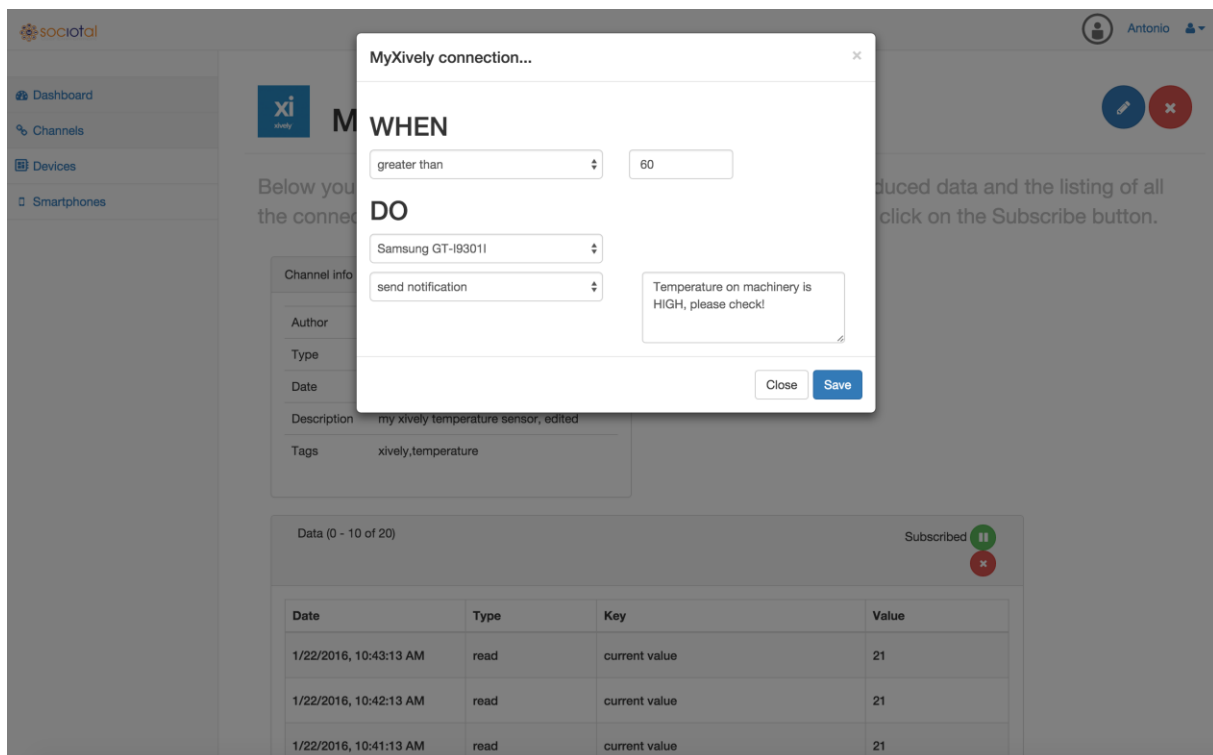


Figure 10: The UI to configure a Channel connection.

2.1.1.1 Interaction with platform (Context Manager)

The Web User Environment is smoothly integrated with the SocloTal Context Manager module (CM) [13]. The UserEnv interacts with the CM through Web API calls. Basically, CM is used to: register new Devices and query existing Devices in the SocloTal platform, subscribe to Devices data. Each API call to CM includes a Capability Token header, which assures the required security and permissions checking in accessing a particular Device by a specific user, as explained in Section 2.1.1.2.

2.1.1.2 Security (auth, authz, idm, https)

Starting from the current beta version of the Web User Environment, all the website and data traffic is under HTTPS protocol. Also the API endpoints work under HTTPS, although they also continue to support HTTP calls for backward-compatibility and to provide an API set for all the limited environments and applications which can't use HTTPS protocol. Nevertheless, using HTTPS is highly recommended, as specified in API documentation [14].

Focusing on security, the Web User Environment is also integrated with the other special SocloTal Components.

Figure 11 shows these components; with them the Web UserEnv communicates and interacts.

Identity Manager (IdM) is used to create and manage users, from sign up to login steps, it allows to authenticate user and to obtain an Authentication Token. The integration with this component is implemented using the SocloTal IdM libraries to authenticate against Keyrock IdM. Capability Manager is used in order to obtain a Capability Token to access SocloTal resources/devices through the Context Manager. Each interaction/API call, after user authentication through IdM, makes use of the Authentication Token, plus a Capability Token. By these tokens the user, through the Web User Environment, can access to Context Manager resources, as reported in the sequence diagram shown in the next figure. All these features are described in the API documentation [15] [16].

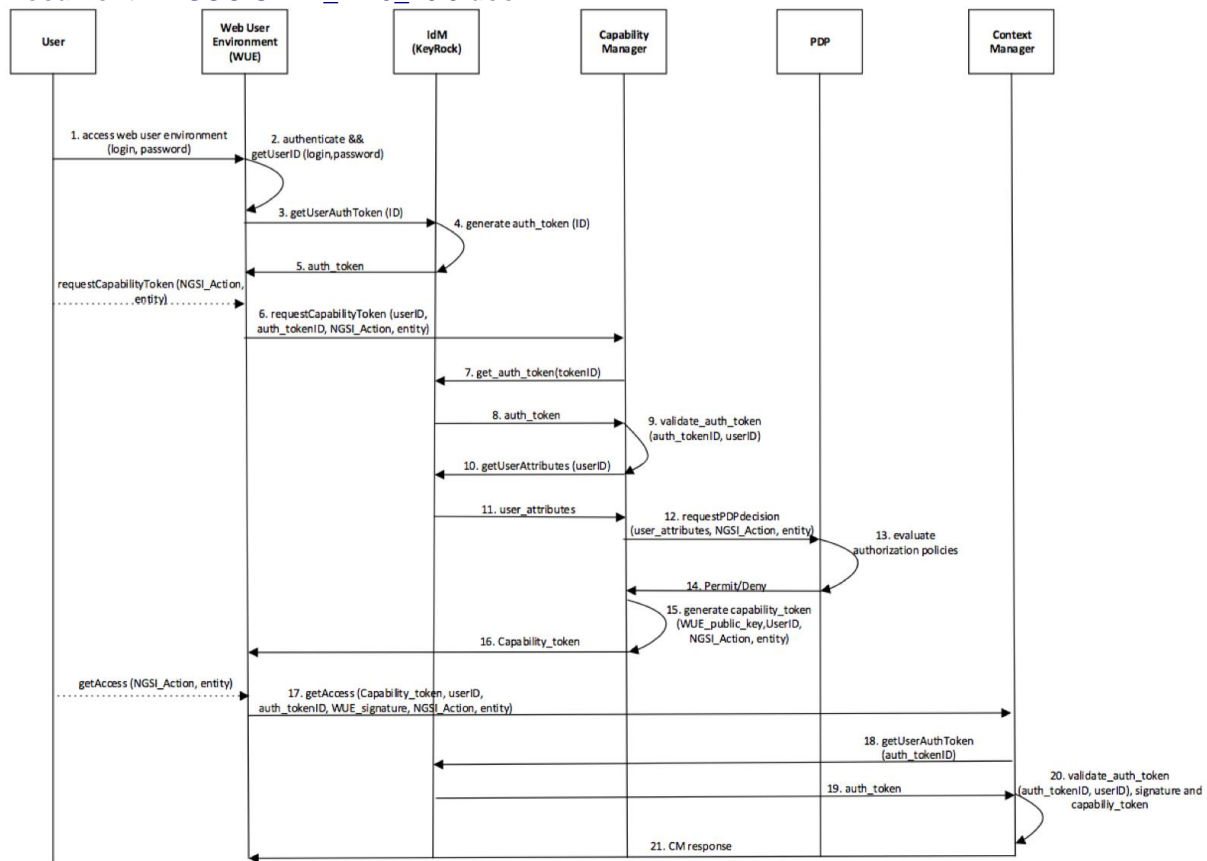


Figure 11: Integration of the Web UserEnv with other SocloTal components targeted to security: the sequence diagram shows all the involved (integrated) steps from users' sign up to SocloTal resources access.

The integration between the Web User Environment and the Identity Manager has been done following the sequence diagram reported in Figure 11.

Once the user does the signup it will be automatically registered both in the SocloTal Identity Manager (IdM, Keyrock-based) instance and in the Web User Environment. Every time the user wants to login to the SocloTal User Environment the following authentication process starts:

- The user fills username and password in the login page of the Web UserEnv (Figure 12: The Web User Environment login step)
- The Web UserEnv authenticates the user against the IdM instance and gets a new authorization token.
- The authorization token is used in order to get a capability token to request access to a resource in the Context Manager.
- To get the Capability Token, username (or subject), the requested resource and the action type for the resource must satisfy the policies stored in the PDP. These policies are verified by the Capability Manager instance and the PDP.
- Once the Capability Token has been obtained, the Web UserEnv can get access to the Context Manager that verifies the token against the IdM.
- If the token is valid, the Web User Environment can get access to requested resources on behalf of the user.

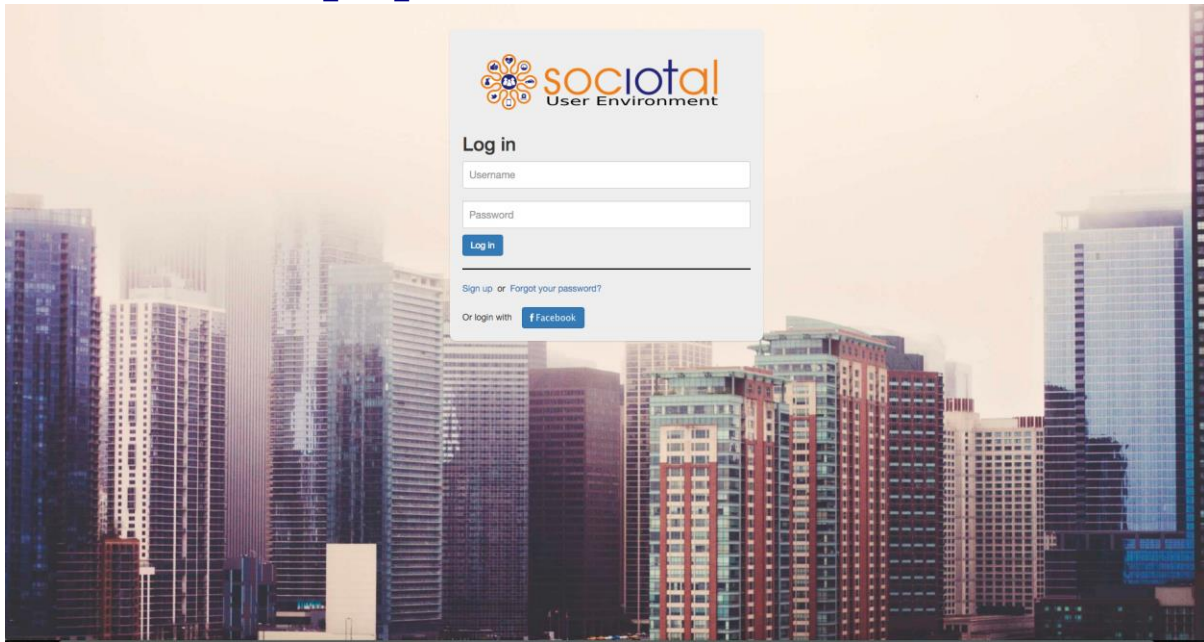


Figure 12: The Web User Environment login step

2.1.2 Mobile user environment

SocioTal User mobEnv is a component of the User Environment that enables users to do the following functionalities:

- Acquisition of device profile via QR code (or to NFC). QR code is the actual ID of the device registered with SocloTal's platform
- Views to manage operations on the mobile: adding device and device deletion
- Push collector: a module that receives Push notification from API-to-API broker.
- Pairing tool: a module that allows the mobile app to be paired with Web User Environment and synchronize operations through APIs.

2.1.2.1 Interaction with the platform

As shown in Figure 3, mobEnv interacts with the rest of SocloTal components using their own interfaces. Therefore, mobEnv supports the following communication with the Context Manager and webEnv (1) Token authorization, (2) Token authentication, (3) Identity management, (4) Resource access, (5) Group sharing, (6) Social interaction detection, (7) Gait recognition (device owner recognition).

2.1.2.2 Authentication

Authentication in the mobEnv is done from the application layer by asking the user to enter the token at the login screen of the application. This token is generated by the webEnv and it is correlated with a created user account.

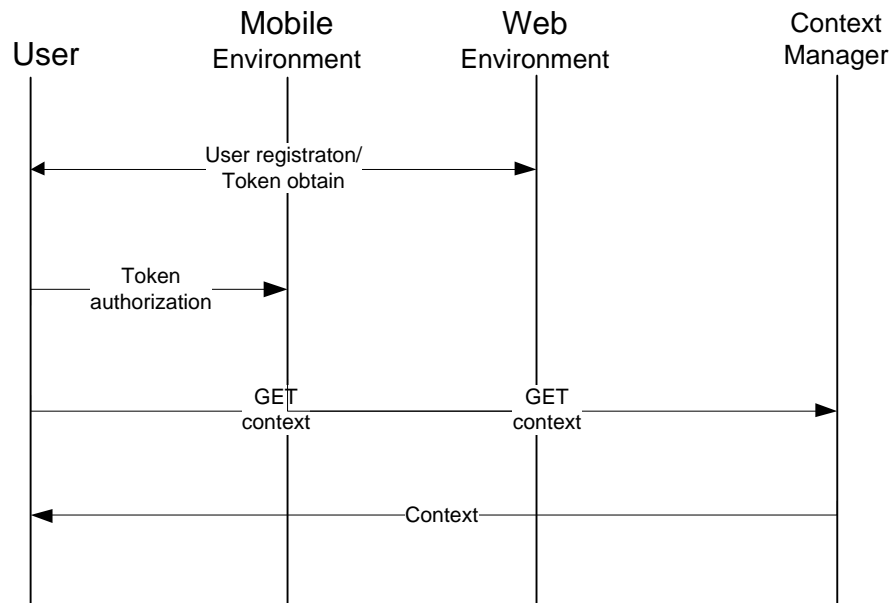


Figure 13: Interaction between User and Context Manager

2.1.2.3 Authorization

Authorization Java library is deployed into the Mobile environment and receives token from the Capability Manager. Token is then sent in the header each time Mobile environment communicates to the Context Manager. Context Manager evaluates the token and authorizes access to resources only if token is valid.

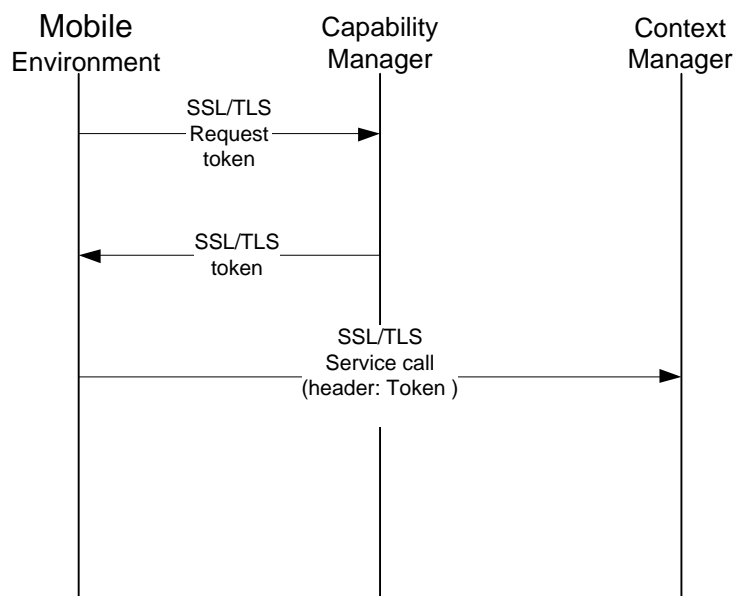


Figure 14: Interaction between Mobile Environment and Context Manager

2.1.2.4 Identity Management

The SocloTal Identity Management (IdM) library is deployed on Android mobEnv and enables privacy-preserving authentication, since users can employ their partial identities to authenticate against the verifier, following the Idemix verification protocol. SocloTal mobEnv is used to obtaining Idemix credentials from the Identity Manager (Issuer server). The issuer server is connected to the Fi-ware Keyrock IdM, so that idemix credential can be obtained according to the user attributes stored in Keyrock. Then validation of the partial identity (idemix proof) is done by the Verifier server which can validate the partial identity derived from the credential. The SocloTal IdM allows also request authorization tokens presenting the partial identities with the attributes that are needed to obtain the authorization token.

In addition to the android app, the The SocloTal Identity Manager is composed of four additional components as defined in the wiki

1. SocloTal-Issuer-Server: It is a web application implemented with Java servlets and XML-RPC which allows generating Idemix credentials for clients. Communications are done by https. The client must be authenticated against the Issuer using a valid certificate. The Issuer also support the verification functionality.
2. SocloTal-Verifier-Server: It is a web application, also implemented with Java servlets and XML-RPC, which is able to validate partial identities presented by the client application.
3. [SocloTal-IdM-Enabled-Capability Manager](#): a web application that allows users to obtain capability tokens using their partial identities. In other words, it allows authenticating and demonstrating their attributes by means of Idemix proofs of having a valid credential issued by the Issuer.
4. [SocloTal IdM KeyRock Client](#): a Java library that provides a basic API for identity management by implementing a client to interact with the FIWARE KeyRock server. To carry out such communication, the SCIM 2.0 and Identity API v3 interfaces provided by this IdM are used.

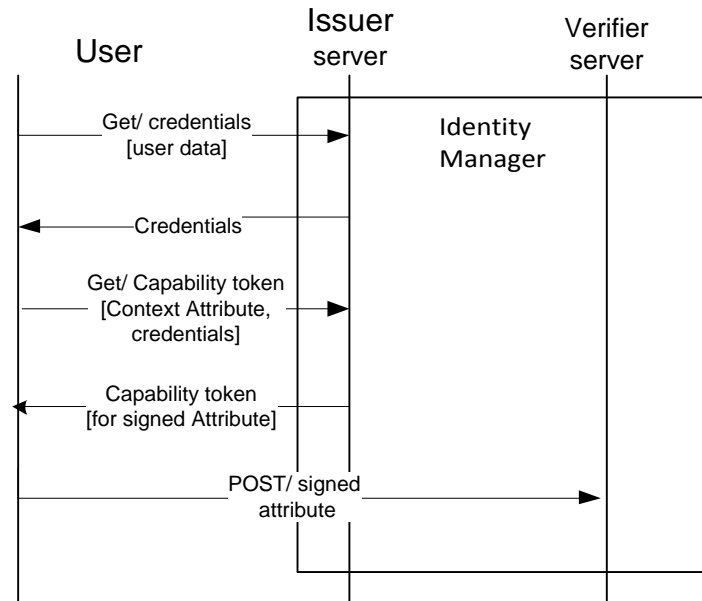


Figure 15: Interactions between user and Identity Manager

2.1.2.5 Group Sharing

Group Sharing provides the low level functionalities to create and manage a bubble of trusted devices according to selected attributes. The sharing key is first obtained from the Attribute authority and saved to the device. This key is then used to encrypt the context which is POSTed to the Context manager by using updateContextEncrypted method [17].

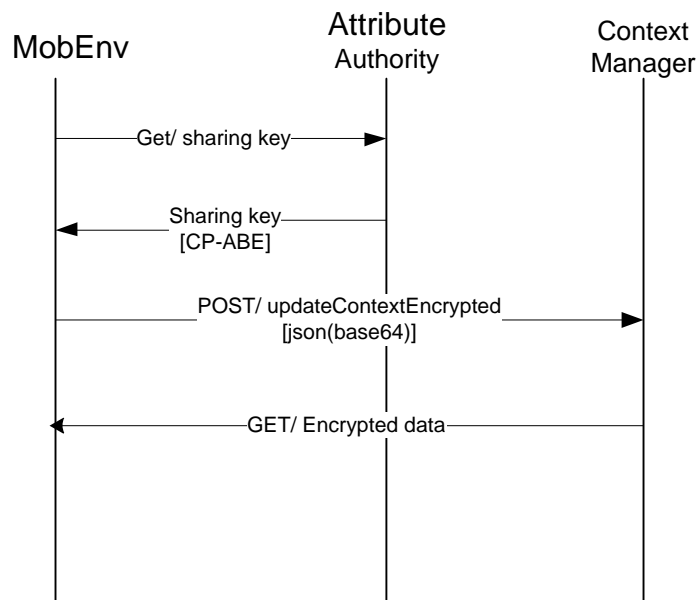


Figure 16: Interaction between MobEnv and Context Manager

2.1.2.6 Face-to-face interaction detection

The Face-to-face (F2F) interaction detection enabler provides the ability to detect social interactions among people based on relative spatial arrangement data. F2F has been integrated with the authorization components, using the capability client library. It is able to obtain capability tokens from the Capability Manager that can be used afterwards, for instance, for accessing to the Context Manager. Sociotal has also implemented a group sharing library. This library allows the F2F-enabler to perform the encryption/decryption operations in order to share the information securely. The F2F enabler considers the interpersonal distance and the relative orientation of the users in order to infer if the participants are taking part in a social interaction. Furthermore, the enabler is able to infer the social relation among people based on their interpersonal distance. The enabler communicates with the SOCIOTAL Context Manager in order to publish the detected social interactions and social relations.

Each device initially retrieves the sharing key from the Attribute Authority and stores it internally in the device. Having retrieved the sharing key, the device is able to post the detected information to the Context Manager in a secure manner. The Web User Environment is able to acquire information that were detected by the F2F enabler through the Context Manager by performing a secure query command [18].

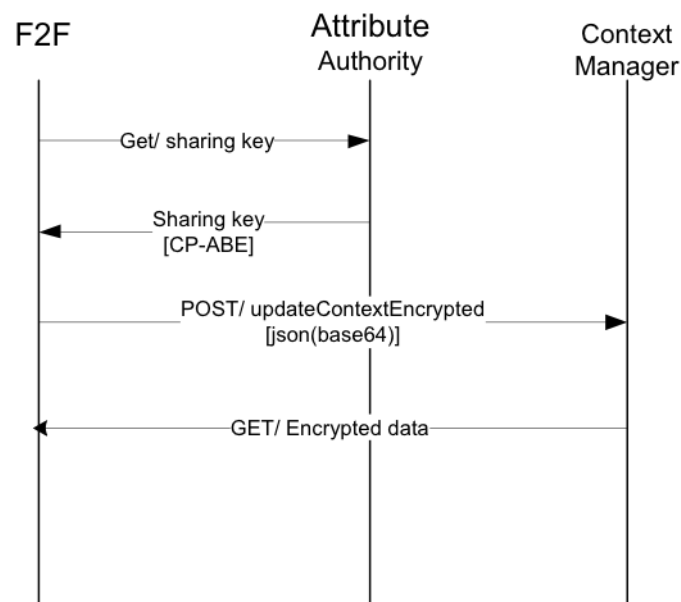


Figure 17: Secure transmission from the Face-to-Face enabler to the Context Manager

2.1.2.7 Gait recognition

The gait recognition app is integrated into the SocloTal platform in the same way as the face-to-face enabler (see Figure 12). The device posts information to the Context Manager in a secure manner. The device obtains the sharing key from the Attribute Authority and this is then stored internally on the device. The key is used to encrypt the information that is sent to the Context Manager. In this way, secure transmission of the result obtained by the gait recognition app is performed. For interacting with the SocloTal Attribute Authority (for key request), SocloTal has implemented a group sharing library. The library also allows to perform the encryption/decryption operations to share the information securely

Section 3 - Deployment architecture

This deliverable details the beta release of the integrated platform. Therefore, in this section, the deployment architecture of the SocloTal project is detailed. The components that form the output of the SocloTal project are made available to developers and end-users through a variety of channels. The aim is to provide high quality components with sufficient documentation so that they can be exploited and further developed by end-users and developers. In addition to separate installations of the components the platform is envisaged to be delivered into a Virtual Machine, with everything needed installed on it.

3.1 Software (list of components)

The SocloTal framework consists of a number of components, a list of which is detailed in Table 2 along with the responsible partner. The following software APIs, components and enablers are identified in the SocloTal architecture, and will be available on the SocloTal's github account [19].

Software component	Responsible	Enabler/Virtual machine requirements
SocloTal Context Manager	UC	JBoss AS 7, Java 1.7, Orion Context Broker (v 0.19 or above)
Fiware Orion Context Broker	UC	Jboss
SocloTal Communities Manager		JBoss AS 7, Java 1.7, KeyRock v4.4.1 (or above)
SocloTal Capability Manager	UMU	Tomcat
Group Sharing app	UMU	Part of mobEnv
KEM Server (Attribute Authority)	UMU	Tomcat
SocloTal Idm Issuer, Idm Verifier	UMU	Tomcat
SocloTal Idm app	UMU	Part of mobEnv
Fiware Keyrock Idm	UMU	Apache
Policy Decision Point	UMU	node.js
Policy Administration Point	UMU	Tomcat
F2F Interaction API	UNIS	Android app

Gait Recognition API	UNIS	Android app
Trust Manager API	DNET	mysql, tomcat
Web User Environment	CRS4	node.js v.0.10.40, mongoDB v2.6 (NGINX, only if https is required)
Mob User Environment	DNET	Android app (includes group sharing app and IdM app)

Table 2: List of Software Components

LIBRARY	PARTNER	LANGUAGE /REQUIREMENTS
Fi-ware IdM keyrock client library	UMU	Java
Capability Verifier library	UMU	Java
Capability client library	UMU	Java
Group Manager client library	UMU	Java

Table 3: List of Software libraries

3.2 Software repository and Wiki knowledge base

In order to disseminate the software developed as part of the SocloTal project, a publically available software repository is used. The beta release of the SocloTal platform is made available on Github [19]. The components listed in Table 2 are those that are currently available.

In addition to the software, detailed documentation is available online in order to aid users and developers with the implementation of the SocloTal integrated framework. The Wiki is created using Github website [19], and contains detailed documentation for the software components that are listed in Table 2. The documentation provides information on the operation of the framework, as well as all the available API endpoints and methods for interacting, extending and working with the framework.

Section 4 - Conclusion

In this deliverable we have presented the Beta release of integrated SocloTal platform. System architecture and technical solution overview are given with detailed descriptions of the components and associated interfaces.

The Web User Environment is a web tool targeted to citizens and end users. Using the Beta release the user can interact with the SocloTal entities in a user-friendly and more secure way. He/she can create her own virtual entities linked to devices previously registered in the Context Manager and new connected devices now can be easily created using ready-to-use templates (weather station, bubble, smartphone, etc...). The Beta version of the Web UserEnv has been totally revamped: new clean design, more intuitive to use and user-friendly, as well as with more information to the end-user.

The integration between the User Environment and the SocloTal security framework, composed of Identity Manager, Capability Manager and the new version of the Context Manager, grants a secure access to all the SocloTal resources: this integration with security framework components works in the background and it does not affect the user experience.

For users with development skills also a set of improved API is included in the Beta release.

The Mobile User Environment is mobile version of the Web Environment targeted to the citizens and non-developers, enabling them to add/remove device to the workspace, reading the values from the added device, etc. The Mobile Environment is integrated with the SocloTal security framework, thus several layers of security are employed for the communication between the mobEnv and the Context Manager, as well as between mobEnv and the webEnv.

The Trust Manager enables the developer to utilize Trust Framework to build a set of rules used to evaluate and quantify different scores as a one reputation score which can be consumed by the end application/service.

The major beta testing is concentrated in upcoming Hackathon, held in Belgrade in June 2016. During the Hackathon event, the SocloTal beta platform release will be tested and evaluated, by enabling participants to use the platform and tools to develop/ make services. The collected feedback will drive the corrective measures that will be applied to the final version of platform before its official release. .

Section 5 - References

- [1] SocloTal github wiki <https://github.com/sociotal/SOCIOTAL/wiki>.
- [2] **NGSI 9/10**, https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/NGSI-9/NGSI-10_information_model
- [3] D3.2.1 Privacy-aware context-sensing device discovery, SocloTal deliverable, FP7 Contract Number: 609112, 2015.
- [4] Orion Context Broker, <http://catalogue.fiware.org/enablers/publishsubscribe-context-broker-orion-context-broker>
- [5] D2.2 Framework specification for privacy and access control, SocloTal deliverable, FP7 Contract Number: 609112, 2015.
- [6] D2.3 Reputation and Trust Management, SocloTal deliverable, FP7 Contract Number: 609112, 2015.
- [7] D3.2.2 Privacy-aware context-sensing information exchange, SocloTal deliverable, FP7 Contract Number: 609112, 2015.
- [8] D2.1 Creating a socially aware citizen-centric Internet of Things, Sociotal deliverable, FP7 Contract Number: 609112.
- [9] Ideentity Mixer, IBM <http://www.zurich.ibm.com/idemix/downloads.html>.
- [10] Fiware IdM “Keyrock”, Fiware catalogue, <http://catalogue.fiware.org/enablers/identity-management-keyrock>.
- [11] Brent Waters, "Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization", Public Key Cryptography – PKC 2011, Vol.6571 of the series Lecture Notes in Computer Science pp 53-70.
- [12] D5.2 SocloTal evaluation, SocloTal deliverable, FP7 Contract Number: 609112, 2015.
- [13] SocloTal Context Manager, Updated version of API Specification, SocloTal deliverable, FP7 Contract Number: 609112, 2015.
- [14] SocloTal User Environment, SocloTal github <https://github.com/sociotal/SOCIOTAL/wiki/User-Environment-API>.
- [15] SocloTal Identity Manager, SocloTal github <https://github.com/sociotal/SOCIOTAL/wiki/SocloTal-Identity-Manager>.
- [16] SocloTal Authorization Manager, SocloTal github , <https://github.com/sociotal/SOCIOTAL/wiki/SocloTal-Authorization-Manager>.
- [17] D1.3.2 Updated version on API specifications, SocloTal deliverable, FP7 Contract Number: 609112, 2015.
- [18] <https://github.com/sociotal/SOCIOTAL/wiki/F2F-Interactions-API#query-f2f-interaction-information>.
- [19] SocloTal github page <https://github.com/sociotal/SOCIOTAL>.

