Specific Targeted Research Projects (STReP)

SOCIOTAL

Creating a socially aware citizen-centric Internet of Things

FP7 Contract Number: 609112

WP2 – Decentralized governance and trust framework

Deliverable report

Contractual date of delivery:
M24 - August 2015
Actual submission date:
11/09/2015

Deliverable ID:	**D2.3**
Deliverable Title:	**Reputation and Trust Management**
Responsible beneficiary:	1/UNIS
Contributing beneficiaries:	UNIS, CEA, UMU, DNET
Estimated Indicative Person Months:	15

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

Revision: Final
Dissemination Level: Public

Document Information

Document ID: D2.3
Version: V1.0 (Final)
Version Date: 11. September 2015
Authors: **Colin O'Reilly, Michele Nati, Niklas Palaghias, Benoît Denis, Jorge Bernal, Jose Luis Hernandez, Antonio Skarmeta, Nenad Gligoric**
Security: **Confidential**

Approvals

	Name	Organization	Date	Visa
Project Management Team	Klaus Moessner	UNIS		
Internal Reviewer	Jorge Bernal Bernabe	UMU		
Internal Reviewer	Carmen Lopez	UC		

Document history

Revision	Date	Modification	Authors
0.1	15/05/2015	First TOC draft	UNIS
0.3	29/06/2015	Addition of SOTA	DNET,UMU
0.4	29/06/2015	Addition of SOTA	UNIS
0.5	08/07/2015	Addition of SOTA (location-based R&T)	CEA
0.6	23/07/2015	Section 3.1.1 about Trust Model	UMU
0.7	23/07/2015	Section 2.3 and 4.2	UNIS
0.8	23/07/2015	Addition of 3.2, 4.2.2.1 & 5.2 (location-based R&T)	CEA
0.9	29/07/2015	Finalizing Section 4.1	DNET
0.10	07/08/2015	Various updates	All
0.11	14/08/2015	Various updates	All
0.12	21/08/2015	Update of 4.1.1 & 5.2 (location-based R&T)	CEA
0.13	29/08/2014	Updated section 4.2	UNIS
0.14	31/08/2015	All sections completed. Ready for internal review.	All
0.15	04/09/2015	UMU and UC Internal review	UMU, DNET
0.16	07/09/2015	Revision of all location-based sections after review	CEA
1.0	11/09/2015	Final version ready for submission	UNIS

Content

Figures

Executive summary

1.1 Description of the deliverable content and purpose

Work package 2 aims to develop a framework for decentralised governance and trust. Objective O2.3 is reported as the **development of novel mechanisms for the automatic establishment of trust relationships for IoT data providers between citizens and entities, as well as with their respective devices.** The work presented in this deliverable concerns the mechanisms that have been developed in order to automatically establish the trust and reputation of IoT devices which provide data to SocIoTal.

This deliverable reports on the progress of Task 2.2 which aims to provide dynamic trust management between things and users. IoT devices that have been discovered by the methods devised in WP3 need to be managed and assigned levels of trust. Novel methods for the establishment of trust are proposed which use novel concepts such as social relationship in addition to more established concepts such as quality of service. In order for the trust management to occur, it is necessary to design the component which manages the information and interaction with other components in the SocIoTal framework. The Trust Manager is also detailed in this deliverable and where it fits into the SocIoTal structure. The Trust Manager has three roles to perform in SocIoTal. Its first role is to receive information from devices (for example smart phones) that is used to determine trustworthiness. The second role is to take the disparate information and compute a single score which is easily understood and acted upon by other devices. The final role is to provide this score to devices that request it.

In addition to dynamic trust management, it is necessary to have mechanisms that allow the identification and selection of the trustworthy services or data providers in the IoT. This is the aim of Task 2.3. Trust can be determined using historic data and this deliverable presents mechanisms with which to do this. Social interactions are used by the face-to-face enabler where trust and reputation is based on the detected social interaction currently occurring. In addition, anomaly detection methods are used to evaluate reputation and trust. The gait recognition algorithm uses anomaly detection methods in order to model a behavioural pattern (gait) of the *normal* user. This model, built on historic data, is then compared to future data in order to determine whether the data can be trusted. Finally, location can be used to establish the trustworthiness of devices by learning mobility patterns and monitoring for deviations. As part of this task, radiolocation-based mechanisms are used to dynamically determine and refine reputation scores.

The deliverable is organized as follows. **Section 1** details the state-of-the-art related to trust and reputation and the methods by which it is established and reported.

In **Section 2**, the enablers for trust and reputation management are introduced. Three enablers are proposed. The first enabler, location-based reputation, derives trust based on the reputation of the user. The second enabler, user behaviour, uses the walking style of the user in order to determine the current user. This enables the determination of the trust and reputation of the smart phone, as it can be communicated to the Context Manager whether the smartphone is in possession of the *rightful* user or whether an imposter currently has the phone. Finally, the face-to-face enabler uses social relations as a measure of trust.

In addition to the enablers, components are developed in order to manage trust in the SocIoTal framework and these are detailed in **Section 3**. The Trust Manager provides a webservice which devices can query in order to obtain trust and reputation score. It

implements a trust model which contains logic with the aim of determining the reliability and trustworthiness of devices in IoT scenarios where device communication can be disparate and between unknown devices. Through the Context Manager, the Trust Manager is able to access real-time information which is used by the trust model for trust and reputation score generation which is then used to determine the current level of trust.

In **Section 4,** initial evaluations are conducted in order to determine the performance of the proposed solutions. The evaluations indicate that the information provided by the enablers have the reliability and quality for the SocIoTal platform to act upon them. The further integration of the components into the SocIoTal platform will be examined in the user trials of D5.2 as part of the evaluations of Task 5.2. This will provide further feedback on evaluations in real-world scenarios. This is important as it will be performed in real-world environments which are less guided. It will also provide performance metrics not evaluation so far such as user experience, computation and energy consumption and system performance.

Section 5 provides the conclusions. In this section, an overview of the achievements of this deliverable are presented. In addition, there is a discussion of how these achievements contribute to the overall project objectives.

Section 1 - State of the Art

In this section, a survey of state-of-the-art is presented which details the concepts related to the components which are introduced in this deliverable. In addition, this section details how the work in this deliverable goes beyond the state-of-the-art. Sections 1.1 and 1.2 provide a review of the current state-of-the-art related to trust and reputation. In the following section, Section 1.3, the state-of-the-art related to using face-to-face relations as a measure of trust is details. Section 1.4 surveys current research in the area of location-based reputation. Section 1.5 details research in the field of user behaviour.

1.1 Trust and Reputation

There are a lot of definitions about trust management and trust-reputation systems generally. Although each of them has separate ways of functioning, the goals are the same. Trust Management is defined as the activity of collecting, encoding, analyzing and presenting evidence relating to competence, honesty, security or dependability with the purpose of making assessments and decisions regarding trust relationships. [1]

The trust and reputation systems are mainly based on model defining, trust score computing and management of reputation data, respectively providing secure and efficient data recovery [2] and methods for quantification of entity's value that other consumers in the system realize as a measure of trustworthiness. In general, trust can be generated by using trust matrix and reputation vector [3]; by aggregating values for some specific scenarios such as peer-to-peer network [4]; or by implementing rule-based agent that takes input from reputation model [5];

The Trust Manager in this work is a REST component based on a generic rules' model that utilize simplified level of score quantification by assigning weights for each examined rule. Trust Manager is developed as a component that will enable user by using generic model for trust and reputation to add, remove and manage his own set of rules that will be used to quantify a final reputation score for his application. This component utilizes and relies on other SOCIOTAL platform components, more specifically on Context Broker to receive/push the updated version of the entity values which is used for building the reputation score. Subscription enables the Trust Manager to on demand re-compute reputation score only for application that consumes certain context in the quantification process. Other approaches are to run computation as a process in the background or to trigger database updates, but these are rather resource consuming or complex to implement.

1.2 Trust-aware access control

The design of trust and reputation models, as well as its application to access control mechanisms, has been traditionally focused on the deployment of security solutions for distributed systems. Besides, they have been usually tailored to wireless sensor networks scenarios, without considering the inherent requirements and features of IoT scenarios. Usually access control mechanisms does not provide trust mechanisms that take into account the trustworthiness of the involved IoT devices to make access control decision accordingly. In this direction, [6] proposes a trust management model based on QoS parameters such as package delivery ratio and end-to-end packet forwarding ratio. It uses fuzzy sets to manage trust and reputation but provides a set of results based on NS-3 simulations. Authors in [7] propose a context-aware and multi-service trust management system that is intended to fit the inherent requirements of IoT scenarios. This mechanism uses different metrics to assess the trustworthiness of devices with different hardware capabilities, in order to cope with the heterogeneous nature of IoT. Moreover, the use of social properties has been employed in [8] to assess the degree of trustworthiness between IoT devices. The proposed approach is based on the Social IoT (SioT) paradigm [9], in which

nodes can establish autonomously social relationships with each other. They present two models for trust management; in the subjective model, each node calculates the trust of its friends based on their own experience and the opinions of friends, while in the objective model, information is distributed so that all devices can use the same information.

Furthermore, a set of simulations are provided to evaluate the model in the presence of malicious nodes. Following this approach, [10] [11] propose a dynamic trust model which takes into account metrics related to social cooperativeness, community-interest and recommendations. As previous proposals, a set of theoretical results and simulations are provided. The integration of trust models in an access control mechanism for IoT is considered in [12], which proposes a reputation-based RBAC access control model (R2BAC). This model uses only QoS metrics to calculate trust values, which are assigned to roles. Moreover, authors in [13] propose FTBAC, a trust-based access control solution by using the linguistic values of experience, knowledge and recommendation as inputs. Then, these fuzzy trust values are mapped to access privileges. Although scalability and energy consumption are simulated, a real implementation of the solution is not provided.

The previous trust models only consider a common small set of parameters to evaluate the trustworthiness of IoT devices (like QoS or reputation). In contrast, [14] considers most of these parameters altogether, adding even a new dimension of trust regarding security. This security dimension allows considering security evidences, which are inferred from security mechanisms being employed in a specific transaction, as part of the trust model which quantifies the trustworthiness. Furthermore, to cope with the vagueness of information in pervasive environments, the model uses fuzzy logic to infer the trust values. In addition, the trust model in [14], and described in section 3.1.1, has been integrated into a lightweight and flexible access control mechanism based on DCapBAC [15], which makes IoT devices aware of other devices' trust scores and drive their access control decisions accordingly.

1.3 F2F relations as measure of trust

With the evolvement of technology new ideas and mechanisms have been developed. However researchers have identified a major gap that requires tackling, the lack of quantifying a strong term such as trust. A large amount of applications in our daily life is based on the trust relationships of people.

Initially, literature focused on measuring trust relationships among people through static social graph. In this case, an initial process is defined were people are characterised by a certain level of trust. Then, the given trust level of people is kept statically throughout the process and does not evolve based on the context in which people are placed in, such as daily interactions. This approach is based on the assumption that there are certain trust relationships establish that do not change such as being part of family. However, in real-world situations trust is affected by the context and different situations. A static social graph is able to provide contextual information related to trust such as the people included in a trust relationship and a quantifying number i.e. the level of trust. As the trust relationships described by a static social graph do not evolve and are only based on the initial definition this approach abstains from a real-world situation.

From 2002, there is burst evolvement of social networks including Friendster, MySpace, and LinkedIn. In 2004 one of the largest social networks is founded, Facebook. These social networks include nodes, which are the people, and edges, which are their relationships. Users based on their own judgement perform requests to establish a relationship in the social network. When both sides – nodes decide that there is a relationship between them, then an edge is created in the social network. In that sense the users decide based on their

own criteria if there is a relationship between two people. This constitutes a valid approach, as the users only have the knowledge about what type of relationship they have with other people. However, it has been observed that people tend to create edges in the social networks that do not exist [18]. This occurs due to different reasons such as need of popularity. Creating edges in social networks that do not exist limit their correctness and reliability. Thus, utilising social graphs extracted from on-line social networks is not a reliable source. There is a need to evaluate or combine the knowledge of real-world social graphs with on-line social networks to create a more reliable social graph to be able to measure trust relationships among people.

In this work, we focus on measuring trust relationship but quantifying social relations through detecting real-world social interactions. Measuring social relations through real-world social interactions allows the creation of real-world social graphs. Social and trust relationships are highly correlated as indicated in [17]. Thus, understanding social relations is an imperative need. To mine peoples' social relation, we have developed a novel approach that is based on commercial off-the-shelf mobile phones. Our approach does not require any additional hardware or firmware modifications. The F2F enabler detects real-world social interactions by measuring the interpersonal distance of people combined with their relative orientation. The interpersonal distance of people is then mapped to the interaction zones of Hall [16]. These interaction zones provide significant contextual information about the social relation of people. The detected real-world social interactions combined with the social relation of people lead to the extraction of important information to quantify the trust among people. In contrast to static social graphs, our approach is able to evolve with respect to time and incorporate changes in the level of trust based on different real-world situations. Regarding to online social graphs, our approach is able to identify opportunistically the trust relationships among people by creating a real-world social graph based on everyday social interactions of people. As the enabler operates opportunistically, the user does not participate in the sensing and inference process. Hence, the approach is not vulnerable to faulty relationship edges such as on-line social networks that are based fully on users' judgement.

1.4 Location based reputation

1.4.1 ***Reputation-based Secure Localization***

The so-called "reputation" of entities participating into a localization process (i.e., fixed elements of infrastructure or mobiles) can be assessed based on their claimed results and/or on collective verification tasks tracking the space/time consistency of the communicated information. One primarily usage of such reputation scores in the literature is to identify and discard malicious or non-reliable (i.e., faulty) entities, thus enabling even more secure and/or robust localization transactions in the end.
First of all, direct solutions aiming at detecting malicious nodes check the compliance of low-level physical radiolocation metrics, which are measured locally (e.g., radio signal strength), with the (explicit) location information broadcasted by mobile nodes [23]. One step beyond in [19] and [20], a simple decentralized reputation rating approach is proposed for secure mobile targets tracking relying on a wireless sensor network. Fixed sensor nodes endowed with a priori known locations (thus forming the "sensing" infrastructure in charge of localizing the target) broadcast their own (claimed) locations along with reputation tables regarding their 1-hop neighbouring fixed sensors. Decentralized opinion tests (first-hand information) are thus locally performed at each of these fixed nodes, comparing the computed positions based on the claimed neighbours' information with the true local information (i.e., the a priori known location). Accordingly, one can determine if he can believe in a given neighbour and update the reputation of the latter accordingly. But the reputation score can be also updated by performing other local opinion tests based on the published reputation tables received from the 1-hop neighbours (second hand information). In each update process, factors are

used to weigh previous experience (i.e., reputation scores locally computed so far) against current information. Finally, one mobile target can be localized based on a subset of the most reputed sensor nodes, hopefully after discarding malicious sensor nodes. Following quite similar concepts in [21], local opinion tests are also carried out at fixed beaconing anchors based on first-hand location information claimed by other neighbouring fixed beaconing anchors. Then mobile nodes simply compute the reputation of a given beaconing node as the average of the reputations perceived about this anchor by its neighbouring anchors. Finally, the mobile node location is calculated based on a subset of beaconing anchors, after sorting their reputations.
In [22], one computes the Maximum Likelihood Estimator (MLE) of target positions based on a subset of fixed sensors. For each sensor belonging to a subset, the quality of the MLE is evaluated to detect inliers/outliers in the produced data (based on MLE residual error monitoring and a priori assumptions regarding observation noise). At each epoch, avoiding hard decisions (i.e., simply outliers detection), the data quality of a given sensor, depicted by its Instantaneous Reputation (IR), is assignment through an exponential function mapping the assumed MLE residual error onto an interval [0,1]. Then the global reputation of this fixed sensor is computed dynamically in a centralized way, relying on its historical estimation performance. In a Bayesian framework, the update is performed by considering the Dirichlet process (instead of a simple Beta distribution for binary IR), evaluating the posterior probability that an observation is a certain event, and thus representing the expected future behaviour of the node.
An overview of generic applicable reputation computation engines for both centralized and decentralized configurations (i.e., regardless of the underlying localization context), has been provided in [24], including e.g., basic summation/average approaches, Bayesian systems and Fuzzy models.

However, in all the previous approaches cited above, the goal is uniquely expressed in terms of localization robustness against malicious or non-reliable nodes (i.e., beaconing anchors or mobiles). Accordingly, performance is assessed only in terms of attacks detection and/or mostly final localization error, assuming nodes that broadcast out false information into the system (e.g., random with unexpected large noise, uniformly distributed random or systematic -stuck-at-fault- regime). Moreover, most of the proposals rely on highly asymmetric systems assuming a fixed infrastructure.
In 3.1.2, we will adapt and compare some of the previous location-based reputation rating mechanisms not for the purpose of improving localization, but mostly for grating personal scores to the mobile users with respect to a priori spatial goals defined in the community (e.g., typically in the context of reliable spatial crowd sourcing). One step further, we also incorporate the peer-to-peer dimension in establishing reputation and trust to benefit from the cooperation potential authorized in Bubbles of Trust between equipped mobile agents.

*1.4.2 **Radiolocation-based Mobility Learning as Part of Spatial Reputation***

Proposals have been put forward to enable the learning, the prediction and/or the detection of mobility patterns or habits out of observed users' physical behaviours. For instance, some techniques consist in exploiting historic mobility traces of individual users gained from wireless localization systems to feed a mobility model set as a Hidden Markov Model (HMM) [25]. The training data (gathered on-line or off-line) are then used to adjust (a priori or conditional) transition probabilities between "pixels" of an environment map (i.e., probabilities to move from one pixel to the next). This mobility model can be coupled with a tracking filter and/or the user's most likely trajectory is calculated using and extended version of the Viterbi algorithm. In the off-line learning approach, the training phase and construction of the matrix with transition probabilities are realized a priori, before feeding the tracking filter. In the on-line version, live filter outputs are used to update the transition matrix on the wing. Regarding shorter-term mobility, the outputs of tracking filters can also be used to classify and learn specific mobility (e.g. linearly accelerated movement) or activity (e.g. walking/running…)

patterns so as to progressively adapt the underlying filter state models [26] and make decision on the currently experienced mobility regime (e.g. assisting the live decision among parallel filter structures or noise hypotheses…).
Besides, other solutions inspired by fingerprinting aim at determining the probability for a user to be in a particular room out of current radiolocation observables (e.g. radio measurements like Received Signal Strength with respect to a set of Access points), based on e.g., Support Vector Machine learning [27].

Most of the previous schemes were intended mostly for improved localization functionalities on their own, although mobility predictability can be considered as one possible candidate ingredient into the reputation of a mobile device or user.
In 3.1.2, besides instantaneous location reliability and consistency, we also consider medium-/long-term mobility learning to contribute into reputation and trust scores.

1.4.3 Spatial Reputation and Spatial Utility in Participatory Sensing

According to the participatory sensing paradigm, the acquired context can be network-attested by the location and time information, which may be as important to data credibility as the gatherer's identity [28]. From that perspective, verifying location consistency (over space and time) and assessing location-based reputation of the involved agents can play a significant role in applications like spatial crowdsourcing (CS), and positively contribute to relevant decision making and further trusted transactions.
In participatory sensing, the goal is to exploit the mobile users by benefiting from their sensor-equipped devices to collect data. More specifically, one particular class of spatial crowdsourcing is depicted as volunteered geographic information (VGI), which consists in creating geographic information (e.g., events occurring at some particular locations) provided on a voluntary basis by individuals (e.g., [30], [31]). One specificity in comparison with conventional spatial crowdsourcing thus lies in the fact that users freely contribute by randomly producing data on their own, but they are not required to physically go to a particular location in order to generate data with respect to that location. From that perspective, VGI falls into the class of self-incentivized crowdsourcing. On the contrary, in conventional spatial CS, specific tasks are assigned to so-called workers by requesters.
In [29], in a conventional spatial CS context, a Least Location Entropy Priority (LLEP) criterion, integrating past areas occupancy, is proposed to improve the overall tasks assignment by setting higher priority to spatial tasks located in places with lower location entropy (i.e., sparsely covered and unfrequently visited by workers so far). Moreover, each worker is endowed with personal skills and an expertise match is introduced as an assignment of a task to a worker, in which the worker has the required qualification to perform the task. Finally, higher scores are assigned to expertise matches and tasks are assigned following a Maximum Score Assignment (MSA) criterion.

The techniques accounted above have been proposed in the conventional spatial CS context (i.e., controlled by requesters but not on a volunteering basis) with the aim of optimizing the tasks assignment rules, but not as an ingredient contributing to evaluate the future spatial utility of an agent with respect to a priori spatial goals (e.g., targeted uniform sensing coverage in a crowdsourcing) and learnt mobility.
However, these criteria can still be adapted into a VGI context and coupled with location-based reputation aspects, as seen in 3.1.2.3.

1.5 User behaviour as measure of reputation

Reputation scores are an important aspect of devices that perform the action of providing data. If a device is being used by a genuine user or is being operated in a genuine manner, a high reputation score can indicate to the network that the transactions or data originating from the device can be "trusted". On the other hand, if a device is not being used by the

genuine user or is being operated in a malicious manner, a low reputation score can inform the network that the transaction should not be carried out, or that the data provided should be ignored.

To assign reputation scores, the data generated by the devices can be used to distinguish a rightful user from an imposter or malicious usage. For example, a smartphone enables the user to perform transactions or share data from on-board sensors with a network. We define the rightful user as the person who owns the smartphone. An imposter is defined as a person who obtains the smartphone from the rightful owner, by stealing it for example. An imposter may attempt to use the phone to perform a malicious action by trading on the trustworthiness of the rightful user. In order to prevent this, data generated by the smart phone can be used to determine whether the current user of the phone is the rightful user.

Biometrics refers to the metrics formed from human characteristics and traits. It is used as a form of user identification and authentication with well-known examples including fingerprints [35], face recognition [36], and iris recognition [58]. With the advent of smartphones with on-board sensors, it has become possible to obtain alternative biometrics by using the sensors to measure the human characteristics. These biometric alternatives can be used to unobtrusively authenticate users by using actions that are carried out naturally, as opposed to more obtrusive authentication such as entering a password or performing a pattern action on a screen. Examples include walking gait [38][39][40], smartphone pick-up action [55], keystroke and touchscreen use analysis [56][57]. An alternative biometric such as walking gait can also be used to authenticate a user while they are not using the smart phone. This will become important in future applications when smart phones become data generators for IoT. It is important to be aware of the identity of the holder of the mobile device even when it is not being used.

1.5.1 Gait Recognition

Biometrics using the walking gait, known as *gait recognition,* has been recognized as a useful method by which to authenticate users. This is due to the fact that the gait of a person has been shown to be highly specific [42], and it is a common activity. The walking *gait* is a cyclostationary process with a period of around one second. The gaits of different people will depend upon a number of factors including limb length, weight and speed of locomotion. This results in the gaits of different people being very different, enabling its use as an alternative biometric.

Walking is an activity that everyone does and that is dispersed throughout the day. Studies have shown that 700-13000 steps a day is normal for a relatively healthy younger adult, and 6000-8,500 for healthy older adult [37]. Therefore walking is an activity that is done many times each day and is also dispersed throughout the day. This makes it an ideal candidate with which to unobtrusively authenticate the user and provide indications of trust levels. Authentication can be used to determine the trust level for a device. If there is a higher probability that the rightful user is in possession of the mobile device, the trust level is higher.

1.5.1.1 Data for Gait Recognition

In order to perform gait recognition, data produced from the walking pattern is required to be gathered. This leads to several steps which are interpreted in different ways by different researchers. First, the action of walking needs to be measured in some way via *sensors*. The sensors will be located in a device which is then placed at a certain *location* on the person. The sensors will gather data at a predetermined *rate*. After the data has been gathered, some *pre-processing* may take place before it is used.

1.5.1.1.1 Data Gathering from Sensors

The sensors used to gather data can be of two types, *specific sensors* set up to gather data about the gait or a *smart phone*. As smart phones are now ubiquitous, most recent research has focussed on using the sensors embedded in the device in order to gather the data required to perform gait recognition. The aim is to authenticate the current user of the phone. Previous research, for example [51], often used devices specifically created for the gathering of data related to walking.

Another issue is the location of the sensors on the body. Current research relies on the fact that mobile device is in the same position on the body, as different motion will be captured by sensors placed on different parts of the body. Some evaluations only occur with the smart phone placed in one location, e.g. front pocket [36][54]. Other evaluations include a combination of frequently used places (e.g. front pocked, back pocket and handbag). In addition to location, the data generated by the sensors is dependent on the orientation of the sensor. For example, the accelerometer in a smart phone is sensitive to the orientation of the device on the body.

Another parameter of data generation is the frequency at which it is sampled. Modern smart phone are able to sample the accelerometer at rates in excess of 100Hz, with some research using the high data rates [52][53][54]. However, this may produce more samples and information than is needed. Many algorithms use a sample of 25Hz [38][39].

Finally, the type of data that is used is important. Modern smart phones have multiple sensors with which to gather data, these include an accelerometer and a gyroscope. Most research uses the accelerometer, for example [39][36]. However, some work uses the gyroscope in addition to the accelerometer [38].

1.5.1.1.2 Data Pre-processing

The data provided by the sensors is a multivariate stream. If the accelerometer is used, there are three axis of acceleration. Many approaches use the magnitude of the accelerometer, calculated using the Euclidean vector norm, in order to provide an orientation-free implementation. However, this leads to a loss of information and it has been shown that considering the three streams is more accurate. Using the three streams means that the orientation of the device must be constant and research has investigated orientation independence [41].

Another data pre-processing method that is used is the down sampling and smoothing of the data. The android operating system provides an API in which to request the current values for the accelerometer sensor. The rate can be specified, however, this rate is just an indication of the frequency that is required and the OS may not be able to provide the values at these exact intervals. Therefore, smoothing of the data is often used where the data is interpolated to provide samples at the exact required frequency [38][39].

1.5.1.2 Feature Extraction and Model Construction

From the pre-processed data stream, features are required to be extracted that provide adequate information on the walking pattern in order to distinguish between the gaits of different people. The aim is to derive features from the multivariate data stream that can be used with machine learning methods in order to create either multi-class classification problems or models of data.

Currently, research has not determined which features or which machine learning method performs the best. Features extracted from the processed data stream range from simple statistics to more complex transformation of the data. For example, the time ordered data stream is often windowed into sections of one second or less with simple statistics of each window such as mean, standard deviation and maximum value, being used as the features [40]. Features from the frequency domain can also be extracted using methods such as the Fast Fourier Transform (FFT) with commonly used features including FFT coefficients, dc component and entropy [42].

Another approach is to use a phase space [50] to represent the cyclostationary process. Singular spectrum analysis [44] projects the data into a phase space where the data is then used as features in the construction of a model. Principal Component Analysis (PCA) is used to derive a basis of the main components. With this model, methods such as support vector machines [54] or geometric template matching [43][38] can be used to classify the stream.

1.5.1.3 Learning Problem

Once the features have been extracted, machine learning methods can be used to label new data. Research into gait recognition is formulated as one of two different machine learning problems; the choice of the problem to solve depends on the end goal. The first problem aims to determine who, out of a limited set of users, is currently using the device; this is a multi-class classification problem. The second problem is to determine whether the current user is the rightful user; this is a one-class classification problem

Multi-class classification is used to determine who, in a closed set of users, is the current user. In this approach labelled data of the different users is available. To determine the current user, a cross-comparison of the gait patterns is performed based on similarity, and the class with the highest similarity score is chosen. Methods such as k-NN [40] and singular spectrum analysis [38][43][39][54] are used. An alternative two-class classification approach uses supervised Hidden Markov Models (HMM)s [45]. Two models are created, the genuine model and the world model, where the world model has been trained using data from 20 different subjects.

One-class classification aims to identify whether the data originates from the normal user or another user, where the rightful user of the phone is considered to be the normal user, and any other user is considered to be the imposter. The one-class classification problem is a semi-supervised problem where labelled normal data is available, but there are no examples of the anomalous user. The aim is to build a model of normal data and then determine whether the current data matches this model.

An unsupervised learning is proposed by Kwon *et al*. [46] that does not require a labelled training data set. The smartphone was placed in the trouser pocket and the accelerometer is used with a sampling rate 50Hz and a 50% overlap of windows. Noise was eliminated using a simple low-pass filter. There are 24 features extracted from the raw data, 12 from the time domain and 12 from the frequency domain using the FFT. Three different unsupervised learning methods are used; *k*-means, GMM and average-linkage hierarchical agglomerative clustering. When the number of activities is known, the GMM method is most accurate. When the number of activities present in the data set is unknown, there is still a high degree of accuracy through methods that determine the number of activities in the data set.

Nickel *et al*. [40] uses the supervised k-NN algorithm to detect the genuine user. Data from the accelerometer sensor is used, with the phone in one position on a pouch attached on the hip to ensure that the vertical acceleration is measured in the x-axis, the forward-backward acceleration on the y-axis and the z-axis measures sideways acceleration. The sampling rate of the accelerometer is approximately 127Hz. Feature extraction occurs on a window of data

of length of between 3 and 5 seconds with a 50% overlap. The features extracted from this window include mean, min, max standard deviation. The training data set consists of data from the genuine user and imposter users, the algorithm will then determine if a user is a genuine or an imposter. A drawback of the approach is that examples of imposter data are required.

1.5.2 Entire User Behaviour

An alternative approach to determining user identity using specific actions, such as gait, is to use the entire user behaviour. This requires continuous sensing of the required data measurements. Zhu et al. [47] introduced an algorithm called Sensec that calculates the degree (termed sureness) to which it can be said that the phone is being used by the rightful user. If there is doubt as to whether the phone is being used by the rightful owner, additional authentication is required when performing private tasks. The sensors used were the accelerometer, orientation, compass, and gyroscope with a sample rate of 4Hz. To construct features a 2 second window was chosen with a 50% overlap, with various statistical measures used to extract data. In order to construct a model of normal usage, an n-gram Markov model was used. Two evaluations were performed, specific and general usage. The specific usage identified 5 tasks to learn in order to determine whether the rightful owner has the phone. The tasks were; pick up the phone from the desk; unlock the device using the right slide pattern; invoke the email app from the home screen; lock the device by pressing the power button; and put the device back on the desk. In addition, for the general usage scenario the phone was used for 24 hours by users, with the normal model being constructed from this data. The algorithm was shown to detect a change of user in under 5 seconds. However, the accuracy of the algorithm is low with many users reporting a large number of false positives.

A scheme proposed by Zhao et al. [48] targets the sensing of the user identify using the accelerometer, sampled at 25Hz, only. A window size of 10 seconds is used. Features are extracted from the data using the FFT. A user specific model is trained on the extracted features using the Naive Bayes classifier to determine whether the current user is genuine or an imposter. In the online mode, a score is generated to indicate how likely the current user is the genuine user. Performance is measured using AUC and the method has higher accuracy than that of Zhu et al. [47].

1.5.3 Beyond state-of-the-art

In this deliverable, we present an algorithm for performing gait recognition on a commercial off-the-shelf smart phone. The technique that we propose is more accurate, in terms of performance metrics, than another method recently proposed. In addition, the method follows a one-class classification approach where the aim is to determine whether the current user is the rightful user of the device. From this approach, an algorithm is developed in order to identify a difference in walking pattern from the trained model. This approach is contrary to other approaches where the aim is to determine which user from a small set of users currently is in possession of the device.

Section 2 - Enablers for trust and reputation

In this section, three enablers for trust and reputation are detailed. The face-to-face (F2F) enabler determines trust based on the social interaction that is current occurring. The location-based enabler determines trust based on the location of the user/device. Finally, the gait recognition enabler determines trust based on whether the current user of the phone is the rightful user.

2.1 F2F enabler

The F2F enabler provides important contextual information to the SOCIOTAL platform. It is able to infer people's social relations through measuring and quantifying real-world social interactions. The enabler infers by estimating the interpersonal distance of people and mapping it to the interaction zones identified in psychology [16]. Then, the relative orientation of people is computed and combined with estimation of the interaction zones the system infers about ongoing real-world social interaction and the social relation of people. For more detailed analysis of the enabler, the reader is referred to [59] and [60].

The above process is performed on-line at the device. The F2F enabler infers about existing real-world social interactions including the social relation of the people. As mentioned the enabler estimates the interaction zone and the relative orientation of people. However, these types of information are not shared outside the enabler. They are kept locally on the device in order to preserve users' privacy. The information that is generated from the enabler is only shared with the SOCIOTAL platform such as the existence of a real-world social interaction and the social relation of people.

The contextual information provided by the F2F enabler is:

- The detected nearby SOCIOTAL devices that are mapped to users.
- The result of inferring if the nearby users are participating in a real-world social interaction.
- The social relation of nearby users.
- The timestamp of the detected real-world social interaction.
- The location the detected real-world social interaction took place.

SOCIOTAL components such as the F2F enabler generate particular types of information and forward them to the SOCIOTAL platform. This process is performed through a core component i.e. SOCIOTAL Context Broker. This component follows a RESTful communication protocol with each of the enabler, allowing it to exchange information in a structured manner. The shared information is represented through a JSON structure. An example of the information generated by the F2F enabler is provided below:

Call	193.144.201.50:3500/SocIoTal_Context_UC_REST/NGSI10_API/queryContext/

Response JSON:


```
{
   "errorCode": {
      "details": "Count: 1",
      "reasonPhrase": "OK",
      "code": "200"
   },
   "contextResponses": [
      {
         "statusCode": {
            "reasonPhrase": "OK",
            "code": "200"
         },
         "contextElement": {
```

```
            "id": "SocIoTal: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"
         }
      }
   ]
}
```


Including these types of information are crucial for the Trust Manager. They allow the extraction of users' social relation and contextual information about their social interactions. The Trust Manager derives the level of trust and reputation based on the users' social relation and further contextual information.

2.2 Location based

Considering Reputation & Trust (R&T) assessment based on wide-sense location information, the latter shall account for devices' positions (2D coordinates) and connectivity (peer-to-peer ranging with respect to other mobile devices), as well as for users' mobility habits. Therefore, we first suppose that each mobile user under R&T rating is endowed with adequate wireless localization capabilities. The acquired location information, as claimed by the different users (e.g., through periodic broadcasting of estimated location information to both the infrastructure and other mobile users), can be used by a central entity to learn mobility patterns (and possibly detect unexpected patterns) by integrating past estimated trajectories or estimated sequences of visited places.

One step ahead, the instantaneous location information and the learnt mobility patterns can be also exploited to grant each single user/device a R&T score according to the consistency (i.e., over space & time) and accuracy of its currently claimed location, to the predictability of its future locations (depending on the quality of the learning process) or even possibly, to the added value of geo-referenced data with respect to a priori community goals. Regarding this last point, in more specific application contexts like geo-referenced crowd sourcing, each

mobile user is equipped with a location-enabled device that produces also location-dependent & time-stamped data (personal and/or environmental), which is for instance issued at embedded physical sensors (e.g., temperature, light, air quality, etc.). Furthermore, spatial desirability maps can be defined by an operator or collectively, by the community of equipped users. The maps shall reflect the geographical needs for new physical sensing actions, to be spatially distributed in some regions at any given time instant. As an example, if one heating system operator a priori aims at a spatially uniform coverage of the sensed temperature on a given building floor whereas there has been no sensing user visiting the coffee room yet, then the latter room shall be given a higher priority on the map. Each new user who is expected to enter the very room (i.e., in a near future, according to its spatial reputation and/or learnt mobility habits for instance) will thus enjoy a high "spatial utility" with respect to the community needs. Accordingly, spatial utility and reputation may be also tied somehow.

Each ingredient taking part into the location-based R&T rating procedure (i.e., location consistency, predictability and spatial utility/added value) can be evaluated independently or combined, so as to produce an overall location-based score, as it will be seen in 3.1.2.

The radiolocation-based enabler could implement an IoT service suitable for smartphones. The generated information can be also made available through the SocIoTal context manager (or even broker) for simplicity. The required attributes to share the related context information are presented below. First of all, explicit context information related to location acquisition, in the most generic radiolocation scenario, comprise the following data (See e.g., D3.1.2 [34]):

1. $\{\tilde{x}_i(k), \tilde{y}_i(k)\}_{i=1..N_u}$ → Mobile device i's estimated absolute 2D coordinates at time epoch k (among N_u equipped users):
    - o issued at an embedded GPS sensor (in outdoor contexts only) or
    - o computed out of WiFi RSSI-based fingerprinting w.r.t. surrounding Access Points (APs) or
    - o computed out of Zigbee RSSI, Bluetooth-LE RSSI or IR-UWB RT-ToF measurements w.r.t. fixed Wireless Sensor Nodes (WSN) anchors/beacons and/or mobile neighbouring devices, respectively through trilateration or cooperative estimation algorithms.
2. $\{\tilde{d}_{ij}(k)\}_{i=1..N_u, j\in Ne_i(k)}$ → Set of single-link relative range measurements of Device i w.r.t. other neighbouring mobile devices $j \in Ne_i(k)$, where $Ne_i(k)$ stands for the neighborhood of Device i at time epoch k:
    - o Zigbee RSSI-based range measurements or
    - o Bluetooth-LE RSSI-based range measurements or
    - o Impulse Radio - Ultra Wideband (IR-UWB) Round Trip – Time of Flight (RT-ToF) based range measurements.

    Depending on the available ranging technology (possibly different from the positioning technology), peer-to-peer range measurements may be more or less accurate, with an error spanning from a few cm in the most favorable cases (e.g., with

IR-UWB RT-ToF in Line of Sight) up to several meters (e.g., with Zigbee RSSI) (See e.g., D3.1.1 [33]).

3. $\{\tilde{r}_i(k)\}_{i=1..N_u}$→ Optionally, mobile Device i's detected room based on estimated absolute 2D coordinates at time epoch k.

4. $\{t_i(k)\}_{i=1..N_u}$ → Explicit time-stamp associated with the delivery of Device i's estimated absolute 2D coordinates, set of single-link relative range measurements w.r.t. neighbouring nodes $j \in Ne_i(k)$ and optionally, detected room at time epoch k.

Note that in geo-referenced crowd sourcing (e.g., participatory sensing) applications assuming slow dynamics of the sensed physical parameters over time (e.g., sensing temperatures and room occupancy for the sake of driving and controlling the building heating system), reasonable localization refresh rates may be sufficient (i.e., with periods spanning from a few tens of sec up to several minutes), contrarily to conventional real-time tracking and navigation applications (i.e., with periods possibly inferior to 1 sec). In the following, we will herein consider 1 min as a reference period between two consecutive time epochs. Within this period and for each mobile device, it is assumed that one has time to acquire the current 2D location, detect the occupied room, check the 1-hop connectivity and perform cooperative peer-to-peer ranging with reachable neighbours, and finally broadcast its local estimations. This refresh rate is perfectly in line with the room occupancy detection functionalities and coarse mobility learning out of the detected rooms (e.g., learning room transition probabilities between a small number of possible states in a Hidden Markov Model framework, whereas the latter would adversely require more refined space discretization in case of real-time tracking, at the price of much higher complexity [25]).

2.3 User behaviour

Using the gait recognition algorithm, a device is able to infer when it is being carried by the rightful user. Once is has been determined whether the phone is in possession of the rightful user, this information is sent to the SocIoTal platform via the Context Manager. The Trust Manager is then able to use this information in the determination of trust. The process occurs online on the device with the relevant information being communicated to the SocIoTal platform in a similar manner to the F2F enabler using the RESTful communication protocol.

2.3.1 Data for Gait Analysis

In order to perform gait analysis, data from the sensors on the phone are collected. As detailed in Section1.5.1, the accelerometer has been shown to have good performance in gait authentication.

The accelerometer measures the acceleration in each of the three axis of the mobile phone, see Figure 1, in terms of m/s^2. The raw data signals from the accelerometer are illustrated in Figure 2 and Figure 3 where each figure shows the accelerometer sensor readings for one person walking at a normal speed. It has been shown that it is sufficient to use the magnitude of the accelerometer, rather than the 3 individual streams. Equation 1 details the calculation of the magnitude of the accelerometer reading. This is advantageous in that the magnitude is rotation invariant and thus it can be orientated in any way in the pocket, as long as it is in the same position.

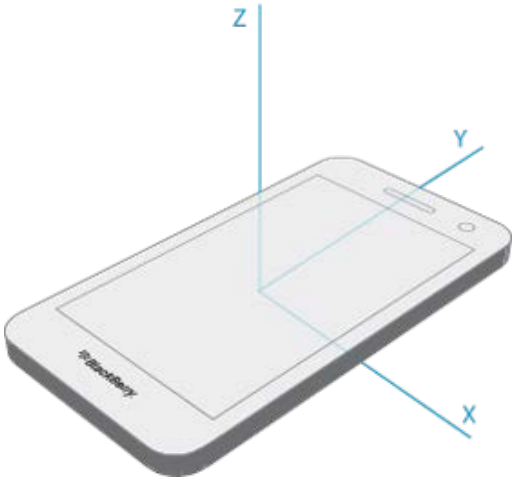

Figure 1: Axis of the accelerometer on a commercial smartphone.

$$\sqrt{x^2 + y^2 + z^2}$$

Equation 1: The magnitude of the accelerometer

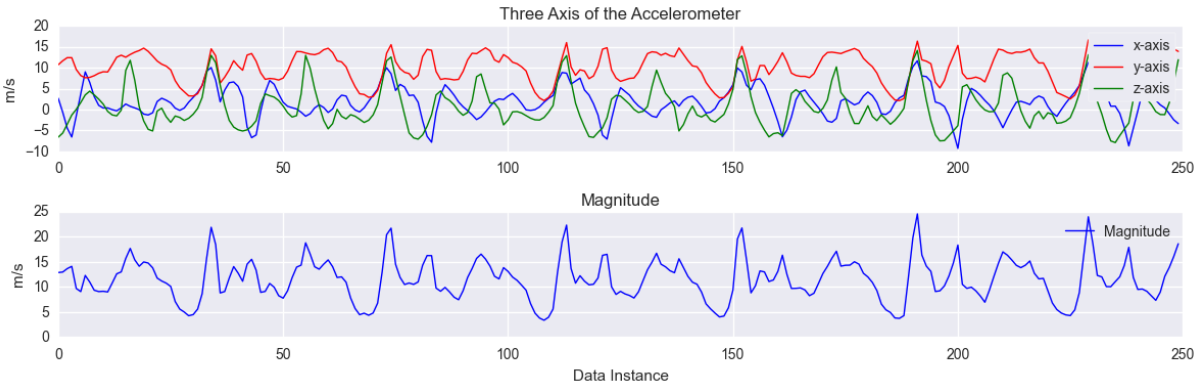

Figure 2: The accelerometer data and magnitude from the accelerometer for user 1 of the McGill data set.

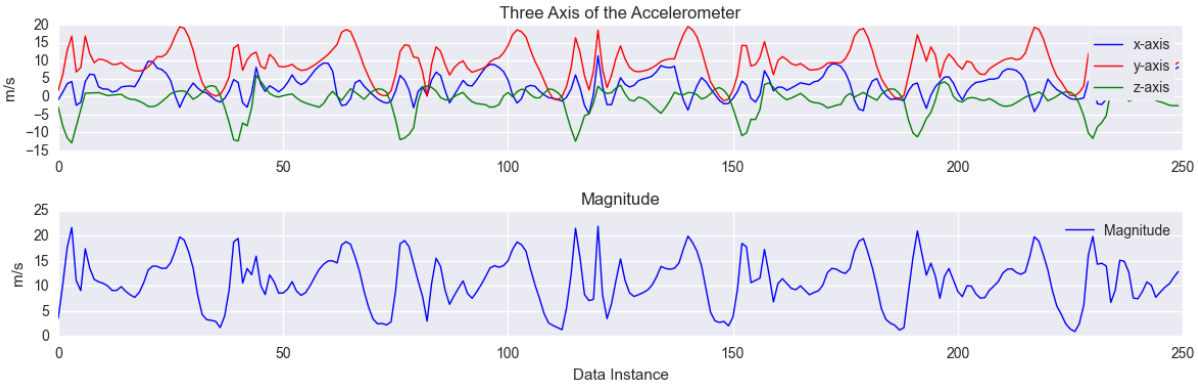

Figure 3: The accelerometer data and magnitude from the accelerometer for user 2 of the McGill data set.

Figure 2 and Figure 3 shows the accelerometer data collected from two subjects. The upper graphs show the three axis of the accelerometer, with the lower axis showing the resulting magnitude vector. When examined with the human eye, patterns can be seen the data

readings. In addition, differences in the patterns can be identified between the users. The aim is to model the patterns and then use this model to determine when this pattern is not found, and thus an imposter has been identified.

2.3.2 Non-linear Signal Modelling

In order to recognize the gait of a user, data is gathered from the smart phone, and from this it is determined when the data differs significantly enough that there is a high probability that the data was generated by a different person.

The main idea of the approach is to perform singular spectrum analysis (SSA) [44] in a non-linear space. SSA is a spectral decomposition method that is used on time series data. The method embeds the time series data $\{X(t): t = 1, \dots, N\}$ into a vector space of dimension M. The non-linear space, which is derived using kernel PCA, is able represent the non-linear dynamics of the system more effectively. Using the non-linear space, new data instances are projected onto the model. The reconstruction error, which represents the error in modelling the data instance in the space, is calculated. A high value for this error means that the data instance was likely created from a different walking pattern.

2.3.2.1 Data Pre-Processing

Data pre-processing is required to ensure that the data received from the sensors are optimal to be used by the algorithm. As the sensors on the smart phone can deliver readings at a variable rate, the data is pre-processed using linear interpolation to ensure the data is at 25Hz. In previous research the magnitude of the accelerometer has been shown to have characteristics distinct enough to differentiate between the gaits of different people, therefore the next step is to calculate the magnitude of the accelerometer, this is done using Equation 1.

2.3.2.2 Form the Trajectory Matrix

The first step is to project the data into a phase space using a trajectory matrix. To perform this, an embedding is created where the original time series is mapped into a series of *lagged vectors* of size L. For example, the first lagged vector is $X_1 = (x_1, \dots, x_{1+L-1})^T$, the second is $X_2 = (x_2, \dots, x_{2+L-1})^T$ etc. In this manner, although the time element has been removed from the data, each feature vector contains a small subset of the entire time series which allows the retaining of the dependency between data measurements in the time series (the lag). The lagged vectors form the columns of the trajectory matrix, Equation 3. This forms the features which represent the dynamics of the univariate time series in the embedded space.

$$\boldsymbol{X} = [X_1 : \dots : X_K] = (x_{ij})_{i,j=1}^{L,K}$$

Equation 2: Formation of the lagged vectors.

$$X_i = (x_i, \dots, x_{i+L-1})^T \ (1 \le i \le K) = \begin{pmatrix} x_1 & \cdots & x_K \\ \vdots & \ddots & \vdots \\ x_L & \cdots & x_N \end{pmatrix}$$

Equation 3: Trajectory matrix

2.3.2.3 Kernel PCA

The second step is to derive a basis in which to represent this space. By using PCA, the principal components can be determined. These are the directions in the space that have the largest variance. By reducing the number of principal components, noise can be reduced in the data. Noise will often have a small variance and thus will not feature in the components with a large variance. The principal components are derived by calculating the eigenvectors of the trajectory matrix, and ordering them according to the eigen value.

A drawback of using PCA is that it is unable to model non-linear concepts as the basis derived is linear, i.e. the principal components are straight lines. The embedding space that the accelerometer data is projected into will contain highly non-linear dynamics that represent the gait of the user. In order to solve this problem, kernel principal component analysis (KPCA) [49] is used in the derivation of the principal components. KPCA is able to represents non-linear dynamics and the kernel principal components are curves in input space, rather than straight lines as is the case for PCA.

KPCA performs PCA in feature space. The feature space $\mathcal{H}$ is related to the input domain by the map. The input vectors are mapped to feature space using $\mathbf{\Phi}: \mathcal{X} \rightarrow \mathcal{H}, \mathbf{x} \mapsto \mathbf{\Phi}(\mathbf{x})$ with the mapped input vectors being $\mathbf{\Phi_x} = [\phi(\mathbf{x_1})\phi(\mathbf{x_2}) \ldots \phi(\mathbf{x_n})]$. The covariance matrix cannot be calculated as it exists only in feature space; however, the eigenvectors of the covariance matrix can be determined using the eigenvectors of the kernel matrix. If

$$\mathbf{K^x} = \mathbf{\Phi_x^\top \Phi_x} = \mathbf{Y \Lambda Y^\top} \text{and } \mathbf{\Sigma^x} = \mathbf{\Phi_x \Phi_x^\top} = \mathbf{U} \frac{\mathbf{1}}{\mathbf{\Lambda}} \mathbf{U^\top} \text{ then } \mathbf{u^p} = \frac{\mathbf{1}}{\sqrt{\lambda^p}} \mathbf{\Phi_x y^p}$$

Equation 4: Eigen decomposition of the kernel matrix

See [49] for details.

$$\boldsymbol{\alpha^p} = \frac{\mathbf{1}}{\sqrt{\boldsymbol{\lambda^p}}} \mathbf{y^p}$$

Equation 5: Calculation of the alphas

Stating $\alpha^{\mathbf{p}} = \frac{\mathbf{1}}{\sqrt{\lambda^{\mathbf{p}}}} \mathbf{y^p}$, the *pth* eigenvector of the covariance matrix can be expressed as $\mathbf{u^p} = \sum_{\mathbf{k=1}}^{\mathbf{n}} \alpha_{\mathbf{i}}^{\mathbf{p}} \phi(\mathbf{x_i})$, or using matrix notation $\mathbf{u^p} = \mathbf{\Phi_x} \alpha^{\mathbf{p}}$. Thus, PCA is able to be performed in kernel space using the feature space mapping $\mathcal{H}$ without every having to perform calculations in the feature space which theoretically is has infinite dimensions.

An assumption has been made that the data has zero mean in feature space. The data cannot be explicitly centred in feature space, as the calculations cannot be performed here. However, this is equivalent to performing the eigen decomposition on the centred kernel matrix [49], Equation 6.

$$K_{centered} = K - 1_M K - K 1_M - K 1_M + 1_M K 1_M$$

Equation 6: Centering the kernel matrix

There are several kernel functions in which to perform the map into feature space. In this work, we have chosen the Gaussian radial basis function (RBF) kernel, Equation 7, due to its ability to map data into a high dimensional non-linear space.

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right)$$

Equation 7: Gaussian radial basis function kernel

The kernel eigenspace (KES) forms the model of the gait of the rightful user. In order to determine whether new data is from the rightful user or the imposter, the data is projected onto the KPCs. Then, a distance metric is used to determine how well the data matches the model. The KPCA and the reconstruction error (RE) has been shown to exhibit good performance in determining whether testing data is drawn from the same distribution as the training data [61], therefore it is used here. The RE is the squared distance between φ(x) and its projection onto the KPCs.

Let $\mathbf{P}$ denote the projection of $\phi(\mathbf{x})$ onto the KPCs, then

$$\begin{aligned} RE(\mathbf{x}) &= \| \overline{\phi}(\mathbf{x}) - \mathbf{P}\overline{\phi}(\mathbf{x}) \|_2^2 \\ &= \overline{\phi}(\mathbf{x}) \cdot \overline{\phi}(\mathbf{x}) - \sum(\overline{\phi}(\mathbf{x}) \cdot \mathbf{u}^{\mathbf{p}})^2 \\ &= \overline{\kappa}(\mathbf{x}, \mathbf{x}) - \mathbf{r}(\mathbf{x})^{\top}\mathbf{A}\mathbf{r}(\mathbf{x}) \end{aligned}$$

where, $\mathbf{r}(\mathbf{x}) = \overline{\mathbf{\Phi}}^{\top}\overline{\phi}(\mathbf{x})$, $\mathbf{A} = \sum \alpha^{\mathbf{p}} \alpha^{\mathbf{p}\top}$ and p is the pth KPC.

Equation 8: Reconstruction error

2.3.3 Implementation

The gait recognition algorithm is implemented as an app for the Android Mobile Operating System, but it is extendable to other mobile platforms as well. The base sensing system makes use of the app that was developed for the face-to-face enabler, which was initially presented in Deliverable 3.1.1 [59]. This app is implemented using a recycling multithreaded approach, where the collection and classification is executed in separate threads retrieved from a thread pool in order reduce the computational burden.

2.3.3.1 Algorithm

The algorithm implementation is detailed in Figure 4. There are two phases; in the training phase the model of the rightful user is constructed, in the security phase ongoing analysis of the accelerometer data is performed to determine who the current user is.

```
User presses Train button;
if Training Phase then
    Pause for 5 seconds;
    Gather 40 seconds of accelerometer data;
    Divide the data into two subsets, Train and Test;
    if Train then
        Create the embedding (Eq. 1 and 2);
        Construct the kernel matrix with the RBF kernel (Eq. 5);
        Centre the kernel matrix (Eq. 6);
        Perform the eigen decomposition on the kernel matrix
        (Eq. 4);
        Calculate the alphas (Eq. 5);
        Retain alphas and training data for the testing phase;
    end
    if Test then
        Calculate reconstruction error (Eq. 8);
        Determine threshold;
    end
end
User presses Security button;
while Security Phase do
    Wait for Activity Recognition (AR) notification;
    if (AR=Walking) then
        if (Current Activity ¬ Walking) ∨ (time elapsed > 60 secs)
        then
            Gather data for 5 seconds;
            Linearly interpolate data to obtain rate of 25Hz;
            Create the embedding (Eq. 2 and 3);
            Project data onto KPC and calculate the reconstruction
            error (Eq. 8);
            Calculate the mean of the reconstruction error;
            Use threshold to determine Rightful User or Imposter;
        else
            Discard data;
        end
    else
        Discard data;
    end
end
```

Figure 4: Algorithm implementation on the Android Smartphone

2.3.3.2 Training

The first phase is training, where accelerometer data for the walking pattern of the rightful user is gathered in order to construct the normal model. As shown in Figure 4, the first phase is to collect the accelerometer data of a normal walking pattern, this is initiated by the user. Once this has been gathered, the data is linearly interpolated to ensure a constant rate of

25Hz. Once complete, the magnitude of each three values is calculated, this will form the signal.

Upon completion of data gathering, 500 data instances (approximately 20 seconds of data) is reserved for training, while the remaining data is used to determine the threshold value for the RE, above which the subject will be considered an imposter.

As detailed in Section 2.3.2, the non-linear algorithm is then applied to the training set in order to construct the model of normal data. The training phase requires the construction of the KES for the training data, as detailed in Section 2.3.2 In order to perform the eigen decomposition of the kernel matrix, the Efficient Java Matrix Library (EJML) is used. This is one of the most computationally complex operations in the algorithm, however, with a training set size of 20 seconds (500 data instances) the eigen vectors are calculated in under twenty seconds on an HTC One with a 1.5GHz dual core processor. In addition, this only needs to be conducted every time training occurs.

Once training is complete, the remaining data is projected onto the model in order to determine the threshold for the RE. The threshold is set at two standard deviations above the mean of the RE.

2.3.3.3 Testing

Once the training phase is complete, the Security mode can be started by the user. The Security phase is also detailed in Figure 4.

In the Security phase, the Android Activity Recognition API [50] is used to determine when walking is occurring. This API is part of the Google Play services APIs and provides a framework with which to recognize the current activity that the user currently in possession of the phone is doing. Activities such as walking, standing still and travelling in a car can be recognized.

The security phase waits for the recognition of the activity of walking. When the current activity changes to walking, 5 seconds of accelerometer data is collected. The test on this data is performed as detailed in Section 2.3.2 to determine if the rightful user is in possession of the phone. In addition to conducting the test when the activity of walking begins, the test is also conducted every 60 seconds, see Figure 4 line 17. This is to prevent too many tests of the walking data being performed, which would lead to redundant information.

In this phase, accelerometer data is gathered in 5 seconds chunks. The model constructed in the training phase is then used to classify each data instance in the 5 seconds of data as with normal or anomaly. This is achieved by calculating the reconstruction error of each data instance, Equation 8. Currently, the mean of the reconstruction error is used as a simple summary statistic for the 5 second chunk of data. More complicated methods are envisioned in the future which may improve performance further. If the mean is greater than the pre-determined threshold, the chunk is considered to have been generated by an imposter. 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.

The testing phase requires the calculation of the reconstruction error for each data instances. This is the second computationally complex operation that is required. On the HTC One smartphone, calculating the RE of 5 seconds of data (125 data instances), takes less than 5 seconds. From this, it can be deduced that the algorithm performs a feasible operation on modern smart phones.

Section 3 - Trust and Reputation Management

In the following section of this deliverable presents work conducted in the area of trust and reputation management. In Section 3.1 the task of building a score from a disparate number of metrics is addressed. In addition, an analysis of how trust and reputation scores can be derived from location is presented. In Section 3.2 the implementation of the Trust Manager is detailed. This component is a webservice with a set of rules for building a reputation score.

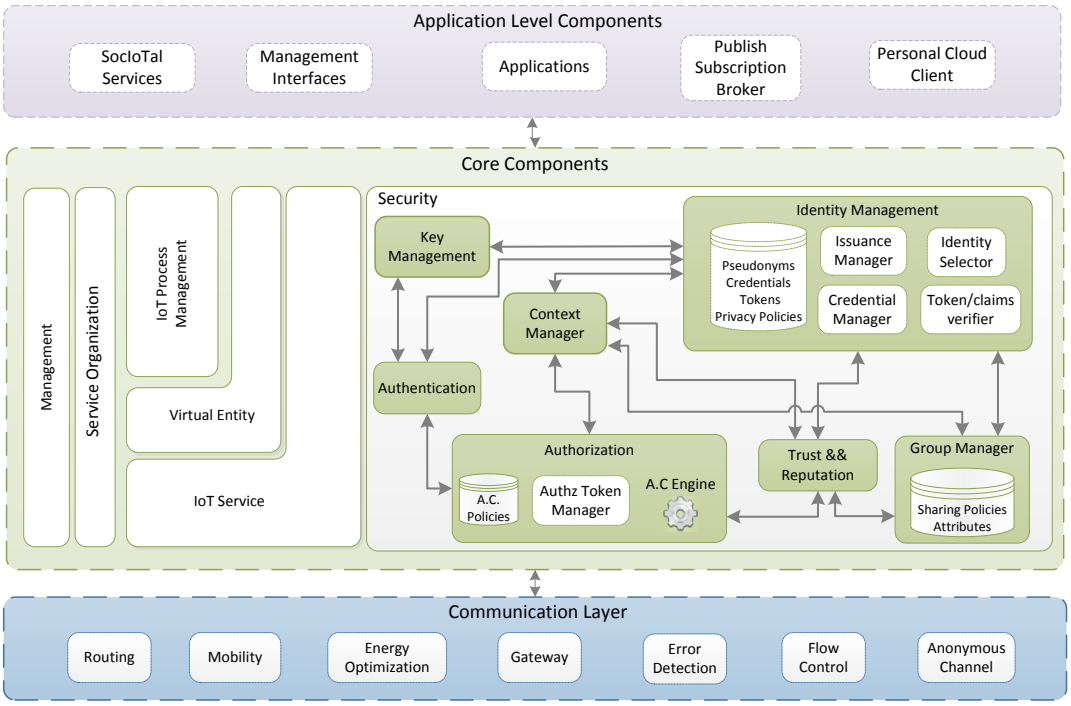

Figure 5 - SocIoTal security framework overview

3.1 Trust-Reputation Model

3.1.1 Multidimensional Trust Model for IoT

The trust model proposed in this subsection, called *TacIoT trust model* [32], aims to improve the reliability and trustworthiness in IoT scenarios where disparate and unknown devices interact each other. The TacIoT trust model follows a multidimensional approach to calculate the overall trustworthiness about an IoT device. It describes the procedure employed to quantify each of the four identified trust dimensions. Then, the dimensions values are aggregated to come up with a final score of trust by means of fuzzy logic.

The trust model considers different properties that could be taken into account in the Internet of Things paradigm. Thus, in addition to traditional considerations such as service feedback and reputation, the trust model takes into account security aspects and social relationships within the peer device. In the end, this approach leads to a more accurate and reliable value of trustworthiness about a given IoT device.

The TacIoT model follows a hierarchical approach in which the different dimensions are split in categories and subcategories, which in turn are composed by measurable properties. This hierarchical approach enables the trust model to be extensible, allowing users to consider and include new properties to the model. Nonetheless, the trust quantification procedure is the same regardless of the amount of properties taken into account. In fact, some of the trust properties explained below could be optional in case the Trust Manager implementing the TacIoT trust model could not have means to obtain evidences about such a property. This would depend on computation capabilities of the device where the Trust Manager is deployed as well as the implementation of the Context Manager component which provides evidences about many of the properties. The Trust Manager is deployed in the same target device when it comes to non-constrained devices like in a smartphones, whereas it can be deployed outside in case of constrained devices. Thus depending on the use case and the specific devices, some of the properties used to quantify trustworthiness that can be considered or not.

Each trust property is measured differently according to its nature. There are basically two kinds of properties, those properties measured as a real number value $x \in [0..1]$ e.g. service availability, and those measured as a booleans, i.e. whether a device features a certain expected security aspect or not. In this last case *x* is a member of the binary set $X = \{0, 1\}$. It should be noticed that property values must be normalized into positive values in range [0,1] using the equation:

$$e = \frac{a - a_{min}}{a_{max-}a_{min}}$$

Equation 9: normalization equation

where a_{min} and a_{max} refers to the minimum and maximum expected values for a given type of property.

The following subsections describe the four main dimensions and the trust properties covered in each of them.

3.1.1.1 Quality of Service dimension

In the TacIoT trust model, this dimension refers to the evaluation of the overall quality of service provided by a device. It is done recapping evidences of previous interactions within the peer device being analyzed. The dimension is, in turn, comprised of four main indicators or properties that help to come up with an accurate overall value of quality of service. All of these properties are real numbers in the [0..1] interval.

The *%Successfull-Interactions* property is the most important property in this dimension since it recaps the percentage of successful interactions over the total amount of previous interactions within the device.

Additionally the *Availability* property indicates the proportion of time that the IoT device is operating. It is measured as the ratio of the total time that the IoT device can be used in a given time interval.

The network *Throughput* when accessing to the IoT service can be also considered to work out the overall quality of service of a given device. The throughput is measured in bits per seconds and is defined as the rate of successful packets delivery over a communication channel. It can be affected by the physical medium or the device processing power.

Similarly, the average network *Delay* within the device measures the overall quality of service. It indicates how long it takes for a bit of data to travel across the network from one device to another.

3.1.1.2 Security dimension

Trust Manager implementing the TacIoT trust model can access the current context by means of the Context Manager component. In this sense, the Context Manager is able to generate dynamically, and in real time, security evidences about the current security mechanisms employed during the actual transaction between two IoT devices. This security information is employed by the Trust Manager in the moment of computing trust. Then, these trust values drive the access control, and therefore, can avoid the transaction before being carried out. The security dimension is set up by the aggregation of different security categories and its properties are explained below.

The *AuthN-AuthZ-System* category refers to the different authorization and authentication mechanisms that are supported by the device in the moment of performing the request. It is in turn split in the mechanisms that are available for different layers, where each mechanism is a boolean property that can be considered by the trust model. In the Network layer, mechanisms such as EAP-IKE, EAP-Coap, EAP-TLS, EAP-SIM, EAP-AKA can be employed in the IoT world. Regarding authorization and authentication mechanisms for IoT at the application layer standards like OAuth, SAML and OpenID can be adopted.

The *Communication-Protocol* security category models evidences about the communication protocols which are used during a transaction between two devices to provide confidentiality and integrity in the communication. The communication protocol employed has a direct impact on the reliability of the peer device. This category is also split in two subcategories, that is, network layer protocols like IPSec, and transport layer protocols like TLS or DTLS.

The *Network-Scope* category evaluates the requester trustworthiness and benevolence from the point of view of its network prefix. Thus, devices in the same local network or with the same network prefix are usually more secure than those devices belonging to global or external networks. Thus, our \trustmodel takes into account, to certain extent (given by the configured weights), the network scope property to evaluate the trustworthiness of the peer device.

The *Device-Intelligent-Capacity* property denotes the amount of risk assumed by the fact of interacting with a powerful device with high computing capabilities. As the device is more capable, it has more probabilities to act dangerously and therefore the risk assumed is higher. The device can identify the peer device and classify it in three levels according to its power and capabilities. Namely, constrained devices (e.g. sensors), common IoT devices (e.g. TVs, smart phones, smart watches) and traditional and powerful machines (e.g. laptops, servers...).

3.1.1.3 Reputation dimension

The TacIoT trust model takes into account recommendations from other devices about a particular device *j*. Let O_j^i be the Opinion about device *j* given by device *i*. The trust model weights each recommender in order to limit their influence according to a recommender's behavior in the past. Thus, the opinions are subject to a credibility process where each reputation evidence coming from a device *i* is subject to credibility factor Cr_i in the interval [0..1], where 1 represents the highest credibility. Therefore, the Reputation property in our trust model is given by $R_j^i = O_j^i * Cr_i$

Additionally, since usually recommender's honesty is not ensured, recommendations are subject to a filtering process to reduce the impact of deceitful recommendations. Recommendations are considered as long as they are similar to an entity's direct experience. The difference between the expectation value obtained based on direct experience and the expectation value obtained by the recommendation is calculated. Then, the recommendation is considered just in the case the difference is less than a predefined threshold.

Each device can store the direct evidences and recommendations provided by other devices to quantify trust of each peer device. Nonetheless, it should be noticed that in the IoT world, constrained devices could not be able to cope with large amount of historical evidences about devices to compute trust. Indeed, these kind of devices can be configured to drop evidences about devices which they do not interact for a long period of time.

Notice that the quantification of the reputation property (like some others properties described in this section) requires a minimum hardware capabilities, which means that it could be not feasible to take into account this property by those Trust Managers deployed in constrained devices. Thus, this trust property is envisaged to be mainly handled by a Trust Managers deployed in non-constrained devices.

3.1.1.4 Social relationship dimension

The TacIoT trust model also considers social parameters as part of the trust quantification for a specific IoT device. These metrics are based on the emerging Social IoT (SioT) paradigm, in which IoT devices are able to establish social relationships with each other. In this case, we consider smart objects can be grouped in different trust bubbles or communities according to the social relationships which are set among them. For example, a *Personal bubble* $B_p$is a set of smart objects that are in contact because they belong to the same owner. Similarly, the smart objects can be connected each other to set up *Family Bubbles* B_f or *Office Bubble* B_o, depending on the kind of the social relationship among them. Other kinds of relationships are *Parental object relationship* and *Co-location object relationship*. The former is defined among similar objects, built in the same period by the same manufacturer. The latter is established among objects that are placed in close locations but without needing to be placed always in the same places, i.e. among objects whose distance in a certain moment is lower than one predefined threshold. Beyond the concept of *Bubble*, a *Community* is a set of smart objects that are in contact because they share a set of common interests.

It usually happens that, as the social relationships between devices get closer, the trust relationship among them are stronger. Our trust model takes into account the kind of social relationship between the evaluated device *j* when assessing the trust of such a device. In this sense, the weights given to a kind of relationship between the devices are different according to the links established among them. For instance, if the device j being analyzed belongs to the device *i* own Personal Bubble B_p, the social relationship will be higher when it only belongs to Family-Bubble B_f. In general, the weights assigned by the trust model to the social relationships are configurable by the user in the interval [0..1] and should satisfy:

$$W_{B_p} > W_{B_f} > W_{B_o} > W_C$$

Equation 10: social relationship weights importance

In addition, in case the devices *i,j* do not belong to the same community or bubble, it can be evaluated the degree of *Interest-In-Common* between the two devices as well as the *Friends-In-Common*. The Interest-In-Common I_j^i trust property can be calculated as the ratio between the interest that both devices share over the total amount of interests of the evaluator device

$I_j^i = \frac{interest(i) \cap interest(j)}{interest(i)}$. Similarly, the Friends-In-Common F_j^i property of the trust model is calculated as the ratio between the number of friends that both devices have in common, and the total amount of friends of the evaluator device $F_j^i = \frac{friends(i) \cap friends(j)}{friends(i)}$.

It should be noticed that to quantify the interests and friends in common the devices should be able to exchange, in a common way, their list of interests (e.g. services and capabilities) as well as the lists of friends.

3.1.1.5 TacIoT model Trust Quantification

Each device is able to provide a set of$Services = \{S_1, \dots, S_z, \dots, S_m\}$, each one providing a certain functionality or resource. The trust model features the *Service-Importance-Factor* $S_{S_z}^j$, which aims to give different degrees of importance to each service provided by the device *j*. Thus, relevance interactions can be given more importance to the overall trust value about a device *j*, whereas irrelevant ones can be discriminated. This factor helps to avoid interactions with malicious nodes that act properly when providing certain minor services, but wrongly when providing the importance ones. The $S_{S_z}^j$ takes values in the range of (0..1], where 1 means highly relevant service and 0 irrelevant service. The Trust Manager holds evidences about each property *p* of the model and weights them by the *Service-Importance-Factor*. Thus, the evidence value *x* about a trust property *p* regarding a device *j*, is given by $x_j^p = v_j^p * S_{S_z}^j$, where v_j^p is the value obtained for the property *p*.

In order to calculate the expected trust value about an IoT device *j* our model recaps the evidences about the different trust properties explained above. Then, when an authorization decision needs to be made, and therefore it is necessary to compute trust, the latest evidence along with the past ones are taken into account in order to have a more reliable value.

In the TacIoT trust model the average rating $\bar{a}$ for each trust property *p* and peer IoT device *j* is calculated with a weighted arithmetic mean. The mean is weighted with the evidences of particular past interactions *n*, ensuring new interactions contribute more than oldest. When the interaction number *n* grows (which means that such interaction is older), its value $x_{j_n}^p$ contributes less to the average mean. The weights follow a forgetting curve $w_n = e^{-n/S}$. *S* is used to customize the exponential function.

$$\bar{a}_j^p = \frac{\sum_{n=1}^{N} x_{j_n}^p * e^{-n/S}}{\sum_{n=1}^{N} e^{-n/S}}$$

Equation 11: average rating equation

An expectation about a trust property is composed of two parameters, the average rating and the certainty. While the *average rating* $\bar{a} \in [0\dots1]$ expresses the average outcome of the past interactions, i.e. the degree of past observations to support the truth of the new evidence, the *certainty* $c \in (0..1]$ indicates the degree that the average rating is representative in the future. The higher the number of obtained evidences the more representative can be considered the calculated security expectation and trust value. Thus, *certainty c* increases according to the number of collected historical evidences. The minimum level of certainty (c=0) holds when there is no previous evidence whatsoever, while the maximum level (c=1) is hold when the total number of interactions *N* reaches the number of

expected number of evidence units. Therefore, the expectation of a certain property *p* for an IoT device *j* is defined as $E_j^p = \vec{a}_j^p * c_j^p$

In order to have an overall value for each dimension, the TacIoT model aggregates the expectations values for each singular property, and in the same category, following a bottom-up approach using a recursive function. The four dimensions are broken-down following a tree structure, having four root nodes corresponding to the four main dimensions and then split in categories until trust properties are reached. Each property and category has a weight assigned that can be customized by the user. Direct children trust properties in the same level are weighted and merged recursively in a single upper value assigned to the parent category, using a weighted mean with different configurable weights for each level. The Trust Manager goes through the tree structure aggregating values in categories until it reaches an overall value for each of the four dimensions.

Then, in order to come up with a final crispy trust value about a given smart object, the four overall values of each dimension defined in TacIoT trust model are evaluated together using a fuzzy approach. Fuzzy logic suits perfectly to deal with the trust quantification since it manages effectively uncertain and aggregated subjective trust values. Fuzziness represents the degree of appropriateness of an element being considered as a member of a group.
Trust quantification based on fuzzy rules suits perfectly in the IoT world since the computation resources required by devices to run the fuzzy system are small. The fuzzy inference uses imprecise linguistic terms, like for instance *high security* or *low reputation*, to specify inference rules, which, when are evaluated in parallel, allow to come up with an overall crisp value of trust.

RULE 1: IF QualityService IS Low THEN trustworthiness IS untrust;
RULE 2: IF Security IS Medium THEN trustworthiness IS trust;
...
RULE n

The output linguistic fuzzy variable *trustworthiness* defines 4 fuzzy values, distrust, untrust, trust and hightrust. The four dimensions are modelled as four fuzzy input variables for the fuzzy system.

The following figure shows the fuzzy grid which represents the fuzzy rules along with its associated weights.

	Low		Medium		High	
	Trust Output	w	Trust Output	w	Trust Output	w
QualityService	Untrust	1	Trust	1	HighTrust	1
Reputation	Distrust	0.8	Untrust	0.8	Trust	0.6
Security	Distrust	1	Trust	0.8	HighTrust	0.8
RelationshipFactor	Untrust	1	Trust	0.8	Trust	0.8

Figure 6 – Fuzzy grid example of TacIoT trust model

The fuzzy inference system is configured to use the Mandami Min implication operator. The membership functions required by the fuzzy system have been chosen bearing in mind the computation limitations which characterizes the Internet of Things scenarios, i.e. choosing lightweight trapezoidal functions to define the input and output variables.

To obtain a final crisp value of trust, a defuzzification method over the resultant fuzzy output is applied. In this sense, TacIoT trust model uses the Center of Gravity (or centroid) method to obtain such a value.

3.1.2 Location-based R&T scores computation

Alternative and/or complementing R&T scoring strategies in charge of the Trust Manager can also be based on the location-dependent information and attributes manipulated by the Context Manager. Note that some of these rating operations could be fully distributed, depending on the embedded capabilities available of the end devices (e.g., while checking the location consistency of neighbors through cooperative ranging) but some others must be centralized (e.g., checking the -predicted- spatial utility of a user based on its learnt mobility patterns against the spatial needs shared by the community of all users), and hence should be ensured by a centralized Trust Manager block. Depending on contextual deployment constraints (e.g. presence of ranging-enabled neighbors or not) and implementation constraints (e.g., sufficient computational resources at the devices), various subsets of the following R&T ingredients can be considered with gradual R&T information richness, and thus, with gradual complexity but also various application potentials. However, regardless of the scoring mechanism it-self (described hereafter in the following subsections) and of the required contextual attributes used as inputs (described in 2.2), the overall data flow can remain similar to that of other schemes which do not rely on localization (i.e., in compliance with 3.2). We provide hereafter generic descriptions for the computation of possible location-based R&T scores.

3.1.2.1 Spatial predictability

For mobility learning purposes, we consider a Hidden Markov Model (HMM) where the *States* $s_i(k) \in \{1,2,..,N_S\}$ correspond to the truly occupied rooms and the successive *Emissions* correspond to the detected rooms $\tilde{r}_i(k)$, based on estimated 2D coordinates $\{\tilde{x}_i(k), \tilde{y}_i(k)\}$ delivered through radiolocation and on the building layout). For the sake of simplicity, we assume here one independent HMM per user, although more complex group mobility learning accounting for inter-personal social habits could be considered as well.

In this HMM representation, the true matrices of state transitions (i.e., room change probability) and emissions (i.e., room detection noise distribution), T_i and E_i, can be estimated "on the wing" (i.e., as function of the time epoch k), given a sequence of emissions $\{\tilde{r}_i(k)\}_{k=1..K_{Learn}}$ (i.e., detected rooms) or a sequence of true occupied states $\{s_i(k)\}_{k=1..K_{Learn}}$ in case of preliminary active learning phase (optional), using for instance a Maximum Likelihood (ML) approach. In the following, we consider this second active option, which is expected to offer better performances, while being still acceptable for some participatory sensing applications (e.g., in a professional environment with natural consent of the mobile agents).

Then, based on the previous mobility learning scheme, for each user $i = 1..N_u$, we consider two indicators:

- A ***Mobility Learning Quality*** indicator (MLQ) in [0,1] accounting for the quality of the learnt transition matrix, which depends on
  - The amount of collected data during the learning phase (i.e., the number of observed epochs)
  - The accuracy of the feeding estimated sequences if no "active learning" is performed (feeding sequences are exact otherwise), which is -in the end- a function of the localization technology accuracy.

In the following, we assume that the learning phase is only based on the data produced by legitimate users (but not supposed to be perturbed by any malicious attacker), although we still assume that an attacker can pretend to be a legitimate user during the generalization phase (i.e., once learning is completed).
As an example, in the following, we simply rely on the global learning RMSE of state transition probability entries (i.e., standard error between true and learnt matrices over their different entries , by design inferior to 1) to derive a common MLQ value as a function of time epochs, averaged over the different users, as follows:

$$MLQ(k) = 1 - \frac{1}{N_u}\sum_{i=1}^{N_u}\left(RMSE_{Trans,i}\,(k)\right)^{\alpha}$$

Equation 12: Global Mobility Learning Quality indicator over users

where α is a parameter to control the dynamics and speed of asymptotic convergence towards 1 as a function of time. Device-dependent quality may be also rated independently as:

$$MLQ_i(k) = 1 - \left(RMSE_{Trans,i}\,(k)\right)^{\alpha}$$

Equation 13: Mobility Learning Quality indicator per user

- A ***Non-Erratic Mobility*** indicator (NEM) in [0,1] accounting for the sparsity of the transition probability matrix, evaluating how the non-zero entries can be concentrated on a few values (i.e., revealing a predictable behavior of the user for the next move, conditioned on the known current room occupancy) in comparison with equally-probable transitions (the latter denoting highly erratic mobility of the user, which can perform a move to any other room). Note that NEM is independent of MLQ but structural/inherent to the user's mobility.
  As an example, for NEM computation in the following, we simply calculate the ratio between non-zero entries in the learnt transition probability matrix over the total number of entries. So this quantity is user-dependent but constant over time after the learning phase (at the very beginning of the learning phase however, the learnt transition probability matrix shall be systematically sparse, due to a limited set of observations $k \ll K_{Learn}$):

$$NEM_i(k) = \left|\left\{\tilde{T}_{i,mn}(k) = 0\right\}_{m=1..N_S,n=1..N_S}\right| / (N_S)^2$$

Equation 14: Non-Erratic Mobility indicator per user

3.1.2.2 Instantaneous location reliability

The accuracy of instantaneous location information (as claimed by a legitimate node) and thus, of room detection, depends on:

- The capabilities of the underlying radiolocation technology;
- The characteristics of the operating physical environment (e.g., obstructed or not, reverberant or not, large-scale or not…);
- The density of infrastructure BS/APs/beacons and related Geometric Dilution of Precision issues.

Besides, for each user, we consider

- An optional ***Device Authenticated Location History*** indicator (DALH) in [0,1] taking larger values if
  - No/rare device impersonations and/or attacks on the localization service have been detected in the past;

- Successful device authentication has been completed, including possibly the verification of location-based pseudonyms as cross-verification overlay (See e.g., D3.1.2 [34]), which do not prevent from device thief but can be advantageously resolved by location space/time consistency verification.

- An ***Instantaneous Cooperative Location Consistency*** indicator in (ICLC) [0,1]
  - Checking at each mobile $i = 1..N_u$ the reliability of each of its neighbours $j \in Ne_i(k)$ on a decentralized basis, by verifying the compatibility between the instantaneous locations claimed by these neighbours $\{\tilde{x}_j(k), \tilde{y}_j(k)\}$, the instantaneous location estimated locally $\{\tilde{x}_i(k), \tilde{y}_i(k)\}$ and peer-to-peer ranging measurements $\{\tilde{d}_{ij}(k)\}_{i=1..N_u, j \in Ne_i(k)}$ with respect to these neighbors.

    We consider a basic approach producing intermediary binary ratings, by computing reputation scores through statistical updating of a Beta probability density function (pdf) where positive hard-decision outcomes represent situations where no outlier has been detected (positive). Note that detected outliers (negative) may result from erroneous measurements (affecting ranging and/or positioning at one or two of the involved legitimate devices) or from the broadcast of erroneous information from a malicious mobile while claiming its own estimated location). The Beta pdf somehow expresses the uncertain probability that future interactions will be also positive. We finally take the expectation $R_{ij}(k)$ of this Beta pdf at each new epoch (function of the relative number of positives and negatives), as follows:

For $k = 1..K$
  For $i = 1..N_u$
    For $j \in Ne_i(k)$

$$\Delta_{ij}(k) = \left| \tilde{d}_{ij}(k) - \sqrt{\left(\tilde{x}_i(k) - \tilde{x}_j(k)\right)^2 + \left(\tilde{y}_i(k) - \tilde{y}_j(k)\right)^2} \right|$$

    For $k > 1$

$$\begin{cases} r_{ij}(k) = r_{ij}(k-1) + 1 \text{ and } s_{ij}(k) = s_{ij}(k-1) & \text{if } \Delta_{ij}(k) \le Th_d \\ r_{ij}(k) = r_{ij}(k-1) \text{ and } s_{ij}(k) = s_{ij}(k-1) + 1 & \text{otherwise} \end{cases}$$

    Else

$$\begin{cases} r_{ij}(k) = 0 \\ s_{ij}(k) = 0 \end{cases}$$

    End

$$R_{ij}(k) = \frac{1 + r_{ij}(k)}{2 + r_{ij}(k) + s_{ij}(k)}$$ (Expectation of the Beta pdf)

    End
  End
End

Equation 15: Neighbours' location reliability per user (decentralized)

  - Centralizing all the intermediary scores to compute an average score $R_i(k)$ per user (based uniquely on the perception of his mobile fellows $j \in Ne_i(k)$), as follows:

For $k = 1..K$
  For $i = 1..N_u$
    For $j \in Ne_i(k)$

$$R_i(k) = \frac{1}{|\{j \in Ne_i(k)\}|} \sum_{j \in Ne_i(k)} R_{ji}(k)$$ (Average of decentralized scores)

$ICLC_i(k) = \frac{R_i(k)}{1-P_{FA}}$ (Final ICLC score)

End

End

End

Equation 16: User's location reliability according to the perception of its neighbours (centralized), and resulting normalized Instantaneous Cooperative Location Consistency indicator per user

Note that we introduce a normalization factor $(1 - P_{FA})$ so that the ICLC indicator falls within the interval [0,1]. This factor can be theoretically computed a priori depending on the detection threshold setting Th_d, based on expected positioning and ranging error regimes (i.e., their standard deviations) while assuming Gaussian centered random errors and legitimate devices on both side of the link for each peer-to-peer intermediary rating.

- A ***Transition Space-Time Consistency*** indicator in (TSTC) [0,1]
  - Verifying the compatibility between the claimed instantaneous sequences and the learnt mobility patterns
  - Detecting forbidden state transitions violating the learnt HMM transition probability matrix (e.g., detecting a non-physical transition from one room to another too distant room within a very short time duration) or very unlikely transitions (anyway with limited impact on scores in case of false alarms) against an arbitrarily low detection threshold (e.g., 5% in the following):

For $k = 1..K$

For $i = 1..N_u$

$\Theta_i(k) = \tilde{T}(\tilde{r}_i(k-1), \tilde{r}_i(k))$ (Estimated probability of the observed room transition)

For $k > 1$

$$\begin{cases} r'_i(k) = r'_i(k-1) + 1 \text{ and } s'_i(k) = s'_i(k-1) & \text{if } \Theta_i(k) \geq Th_t \\ r'_i(k) = r'_i(k-1) \text{ and } s'_i(k) = s'_i(k-1) + 1 & \text{otherwise} \end{cases}$$

Else

$$\begin{cases} r'_i(k) = 0 \\ s'_i(k) = 0 \end{cases}$$

End

$TSTC_i(k) = \frac{1+r'_i(k)}{2+r'_i(k)+s'_i(k)}$ (Expectation of the Beta pdf)

End

End

Equation 17: Transition Space-Time Consistency indicator per user

In 4.1, for our evaluations, we will focus mostly on the two last points for simplicity, assuming no particular authentication/security issues in the past (and thus DALH=1 systematically for all the users and all the epochs).

3.1.2.3 Expected spatial added value

Finally, and mostly for specific geo-referenced crowd sourcing applications, we also consider the expected spatial added value of a user/device with respect to a priori community goals. Thus we define two more indicators, as follows:

- An optional ***Data Rate History*** indicator (DRH) in [0,1], based on

- The available refreshment rate of geo-referenced & time-stamped data produced by the user, which may be independent of the location information rate.
- The observed data traffic so far for each mobile.

- A ***Spatial Utility*** indicator (SU) in [0,1]
  - Evaluating the match between the predicted user's locations and an a priori spatial desirability map (in a reasonably short-term observation period). The spatial desirability for each room r at each epoch k can for instance be determined based on
    - The set of visits already observed to this room over an arbitrary past period (e.g., for the latest 15 minutes, for the last hour…), as revealed by the location entropy $Ent(r,k)$, which measures the diversity of unique visitors to a particular room r:

$$Ent(r,k) = -\sum_{u=1}^{N_u} P(r,u,k).log(P(r,u,k))$$

Equation 18: Current Location Entropy per room

where $P(r,u,k) = \frac{|O(r,u,k)|}{|O'(r,k)|}$ gives the probability that a random draw from the set $O'(r,k)$ of visits to room r over the arbitrary period occurring before the current time epoch k, belongs to the set $O(r,u,k)$ of visits that user u has made to this room r for the same period; $|O'(r,k)|$ is thus the total number of visits to room r over the arbitrary past period. For simplification, in the following we consider as observation period the overall duration since the beginning of the acquisition and up to the current time epoch k.

- A priori goals expressed in terms of spatial coverage for the underlying application. For instance, considering participatory sensing with distributed georeferenced temperature measurements (e.g., with the sake of driving the building heating system), one may want that all the rooms are equally covered by physical sensing, or that all the rooms have been visited at least once, or that the frequency of visits/sensing actions is proportional to the area of the room… In the following, as a particular example, we will simply consider that the room with the lowest location entropy at each epoch (if a few users have visited this room in the past) is the room with the highest spatial desirability $r^*(k)$ (i.e., the one that should be preferably visited in the short-term by any user):

$$r^*(k) = argmin_r\, Ent(r,k)$$

Equation 19: Least-location-entropy selection of priority room(s)

Note that the spatial desirability is extrinsic and independent of the user's spatial reputation but the confrontation of the latter reputation with the needs is the quantity integrated as possible R&T ingredient, as seen below.

- Based on the spatial reputation gained through mobility learning (through the learnt version of the HMM transition matrix T_i and its successive powers $T_i^{K'}$), the spatial utility of each user is computed as the posterior probability that the user will occupy the room with the highest spatial desirability (and thus, the probability that the user fulfils the needs of the community) within a given horizon of time K' after the current time epoch (i.e., in a reasonably near future between k and $k+K'$), given his current room occupancy. In other words, we just evaluate the

powered matrix $T_i^{K\prime}$ at a row index equivalent to the currently detected room $\tilde{r}(k)$ and at a column index corresponding to the room with the highest spatial desirability $r^*(k)$.

The previous location-based R&T scoring techniques will be illustrated and evaluated by means of simulations in Section 4, with a specific application to distributed participatory sensing.

3.2 Trust Manager

Trust Manager (TM) is envisioned as a mixture of a REST webservice and logic with a set of different rules 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 Trust Manager utilize to continuously maintain the updated version of score in respect to last attribute value changes is a Context Manager.

To register and start the reputation process for an application, JSON in predefined structure must be POST-ed to the REST endpoint (1). After the registration, Trust Manager automatically extracts for the first time attributes value from the Context Broker (2), computes (3) and POSTs the final reputation score to the Context Broker (4). Rules are saved in the local database. Trust Manager then subscribes to the attributes' value changes (4). When value of an attribute is changed (5), Context Broker 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).

The figure below shows the Trust Manager interaction with other components and complete process used for building the reputation score.

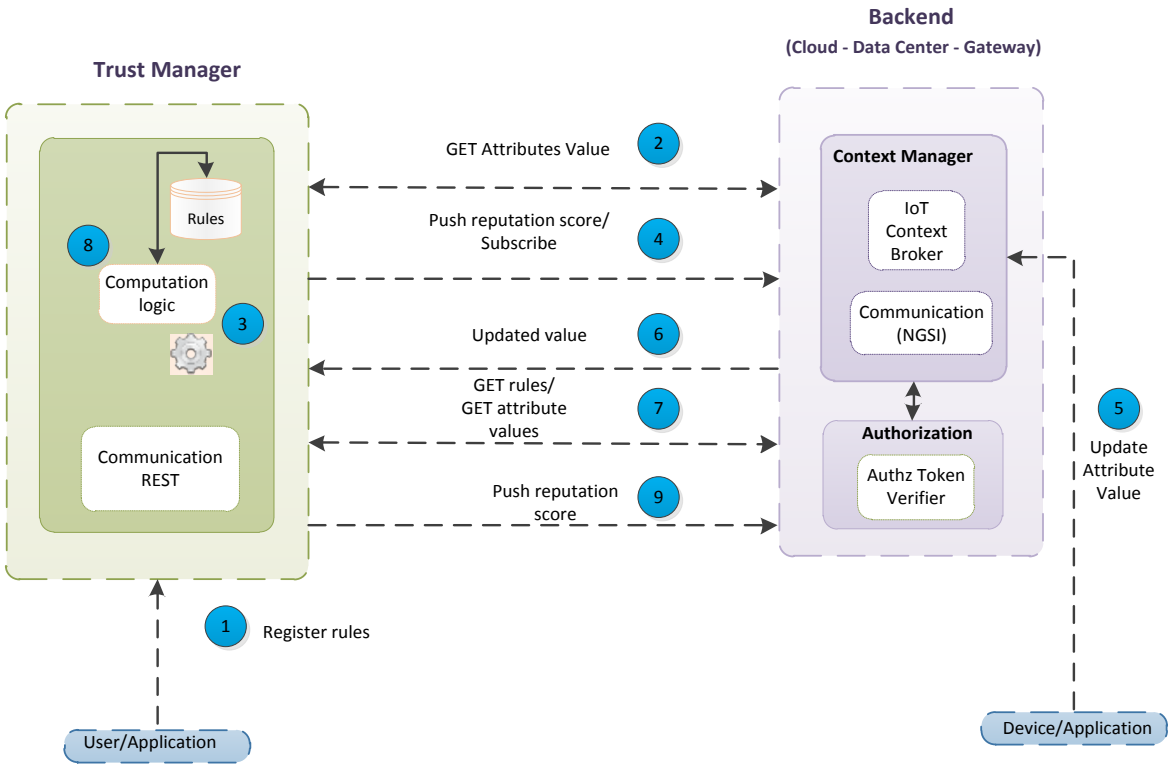

Figure 7 Trust Manager interaction with Context Manager

3.2.1 Computation logic

Each application has one rule table assigned and populated with combination of rules for the given application ID. Trust Manager can compare two attributes value or attribute value against a custom value.

POST(ArrayList<Rule>)
Rule(String application_id, Attribute attribute, String operandKeyword, Attribute compareAttr, float reputationWeight) Attribute(deviceId, attributeName, attributeValue)

The general logic of the Trust manager is based on assigning weights to a condition: if rule is met then computation method returns 5 and if not it returns 1. Final score represents average of all returned values.

Example below shows pseudo rule when two devices must be in the same room and users must be friends in order for reputation score to have value 5:

Rule (application_id, device01->position=room313, IS, device02->position=room313, 5)
Rule (application_id, device01->relation_with_device02=friends, IS, device02->relation_with_device01=friends, 5)
….

TM queryContext for device01's attributes and check if attribute position value is "room 313" for this device and then queryContext for device02 attribute position value and checks if it is as well "room 313". Then TM checks if first device have attribute that indicates that the other device is "friend" with the first device and vice versa. If all conditions are met final reputation score will be 5, and if one or both fails reputation will be 3 or 1 accordingly.

Logic:
1. check if rule is satisfied and if does assign 5
2. If condition is not satisfied default weight value will be 1
3. Mean is calculated from all reputation weights

In case if attribute value should be compared against custom provided value (not extracted from the Context Broker, but given by the user during the rule registration) compareAttribute type must be equal "string".

Rule (application_id, device01->position=room313, IS, attribute=string, 5)

3.2.2 Rule registration

Registration is a process that must be done for each application that consumes final reputation score and this is a first operation that initiates the start of the score quantification. Example of the context for one device that is kept in the Context Manager is presented in the table below. In this example two devices are pushing the same structured JSON with different attribute metadata value for device name.

```
{
   "contextElements": [
      {
         "type": "urn:x-org:sociotal:resource:device",
```

```
        "id": "SocIoTal:UNIS:SmartphoneContext:VirtualSmartphoneContext_002",
        "attributes": [
          {
            "type": "boolean",
            "value": "false",
            "metadatas": [
              {
                "type": "http://sensorml.com/ont/swe/property/pseudonym",
                "value": "SocIoTal:UNIS:SmartphoneContext:VirtualSmartphoneContext_001",
                "name": "DiscoveredDevice"
              },
              {
                "type": "string",
                "value": "PERSONAL",
                "name": "SocialRelation"
              },
              {
                "type": "http://sensorml.com/ont/swe/property/DateTimeStamp",
                "value": "20150603T105032Z",
                "name": "DateTimeStamp"
              },
              {
                "type": "http://sensorml.com/ont/swe/property/Location",
                "value": "20.475431419909, 44.81156132183969",
                "name": "Location"
              }
            ],
            "name": "F2FInteraction"
          }
        ],
        "isPattern": "false"
      }
    ],
    "updateAction": "UPDATE"
}
```


Rules for the previous context presented in JSON format that should initiate trust computation is given in the following table. Provided rules can be interpreted as follows:

1. GET from Context Manager context with id "SocIoTal:UNIS:SmartphoneContext:VirtualSmartphoneContext_002"
2. Find attribute "SocialRelation"
3. Check if attribute "SocialRelation" have value "SOCIAL"

SocIoTal SERVER	89.216.30.230	
POST	/TrustManagerSociotal/rest/api/registerRules	
HEADERS		
	Content-type	application/json
	Accept	application/json
Payload:		


```
[
 {
  "applicationId":"iot_week",
  "attribute":{
   "deviceId":"SocIoTal:UNIS:SmartphoneContext:VirtualSmartphoneContext_002",   1
   "name":"SocialRelation"   2
  },

  "operandKeyword":"IS",   3
  "compareAttribute":{
```

```
      "value":"SOCIAL",
      "type":"String"
    }

  },

  {
    "applicationId":"iot_week",
    "attribute":{
      "deviceId":"SocIoTal:UNIS:SmartphoneContext:VirtualSmartphoneContext_002",
      "name":"SocialRelation"
    },

   "operandKeyword":"IS",
    "compareAttribute":{
      "value":"PERSONAL",
     "type":"String"
    }

  },

  {
    "applicationId":"iot_week",
    "attribute":{
      "deviceId":"SocIoTal:UNIS:SmartphoneContext:VirtualSmartphoneContext_001",
      "name":"SocialRelation"
    },

    "operandKeyword":"IS",
    "compareAttribute":{
      "value":"SOCIAL",
      "type":"String"
    }


  },

  {
    "applicationId":"iot_week",
    "attribute":{
      "deviceId":"SocIoTal:UNIS:SmartphoneContext:VirtualSmartphoneContext_001",
      "name":"SocialRelation"
    },

   "operandKeyword":"IS",
    "compareAttribute":{
      "value":"PERSONAL",
     "type":"String"

    }

  }
]
```


The payload includes a list of rules elements, each one with the following information:

- applicationId: unique ID of the application that starts the reputation computation
- attribute: an attribute with value that is going to be compared
- operandKeyword: keyword used for comparing two attribute, if one attribute value IS, GRATER, LESS or ISNOT equal to another attribute value (or custom provided value).

compareAttribute: value used for comparison. If this is not an attribute then its type should be "type":"String"

3.2.3 Trust Manager Integration with Context Manager

Trust Manager automatically subscribes to attribute value changes to continuously maintain the updated version of score in respect to last attribute in the Context Manager. Trust Manager Endpoint that receives updated value recomputes the reputation score for each application that utilize that attribute for score quantification and pushes it back to the Context Manager.

The score can be extracted with queryContext method by using application Id used during registration as a parameter. The shared information is represented through a JSON structure. An example of the reputation generated by the Trust Manager and extracted with a GET method from the Context Manager is provided below:

Call	193.144.201.50:3500/SocIoTal_Context_UC_REST/NGSI10_API/queryContext/ applicationID

Response JSON:

```
{
    "contextElements": [{
        "id": "applicationID",
        "type": "SocIoTal_Resource:Reputation",
        "isPattern": "false",
        "attributes": [{
            "name": "Reputation",
            "value": "4",
            "type": "float"
        }]
    }],
    "updateAction": "APPEND"
}
```

Section 4 - Evaluation

The following section presents the evaluation of the enablers previously presented in simulated environments and small real-world scenarios. The aim is to assess the performance of the enablers in terms of current state-of-the-art (i.e., gait recognition) and to provide comprehensive illustrations if the domain/topic has not been really addressed yet, to the best of our knowledge (i.e., location-based R&T scoring in participatory sensing). The evaluation of some enablers in more detailed and advanced environments will occur in Work Package 3 deliverables. These will address how the enablers perform and are used in within SocIoTal.

4.1 Face-to-Face Enabler

In this section we provide a brief overview of the evaluation for the F2F enabler. The aim is to introduce the evaluation of the F2F enabler. This evaluation was described and discussed in detail in [59] and [60]. The performance evaluation of the enabler is divided in real-world benchmark of the enabler against an RFID approach and in evaluation of the interpersonal distance estimation technique against the state-of-the-art techniques.

The first scenario [59] includes a real-world evaluation of the enabler among 9 people in an office environment. Participants are placed in a typical office environment to socially interact. Each participant is provided with a smartphone having deployed the F2F enabler. Also, each participant is provided with an active RFID-tag. The results showed that the F2F enabler was able to recognise 81.4% of the on-going social interactions.

The second scenario [60] includes an off-line evaluation of the interpersonal distance estimation technique introduced in the deliverable. The F2F enabler is able to detect the social relationship among people by classifying their interpersonal distance. The evaluation is based on an extensive dataset of 48000 RSSI samples. State-of-the-art techniques were implemented and benchmarked against the proposed interpersonal distance estimation. The proposed interpersonal distance estimation managed to achieve an increase of accuracy of at least 8%.

For a more detailed analysis of the evaluation of the enabler the reader is referred to deliverables [59] and [60].

4.2 Location-Based Reputation

In this section, we provide simulation-based illustrations and evaluations of the proposed location-based R&T rating mechanisms. Some of these mechanisms are expected to be compliant with the radiolocation capabilities foreseen for integration in WP4.

4.2.1 Evaluation Setting

In our nominal scenario, we consider N_u=4 mobile users moving in a typical indoor office environment (e.g., during the working hours) composed of 9 distinct rooms, as follows:

- Room 1: Corridor
- Room 2: Office A
- Room 3: Coffee Room
- Room 4: Office B
- Room 5: Office C
- Room 6: Printers

- Room 7: Toilets
- Room 8: Office D
- Room 9: Lab

At the beginning of a simulation trial, each user is assigned a room number among the available offices (presumably their own office at work).

Considering conventional wireless localization technologies, the users' estimated 2D coordinates are assumed to be affected by additive centred Gaussian random noise terms, whose standard deviations, respectively $\{\sigma_{x,i}\}_{i=1..N_u}$ and $\{\sigma_{y,i}\}_{i=1..N_u}$, mimic faulty users (i.e., reflecting the precision of their positioning technology) or malicious users (i.e., claiming deliberately erroneous estimated positions to alter the system). Peer-to-peer ranging measurements with respect to 1-hop neighbours are also supposed to be affected by additive centred Gaussian random noise terms, whose standard deviation σ_d reflects the quality of the ranging technology, which may differ from the positioning technology (e.g. WiFi and IR-UWB, respectively). Note that the two noise terms affecting the 2D coordinates are i.i.d. for a given user, for simplicity.

The refresh rate of the location information acquisition and its subsequent broadcast (by all the users) is set to 1 min (epoch period) and each trial is simulated for several hours.

At each new epoch, the users are supposed to move from one room to the next (or eventually, to stay in the same room) with a certain a priori transition probability, according to a HMM model (See Figure 5), where the line indexes in the state transition matrix account for the indexes of the currently occupied rooms and columns for the indexes of the visited rooms at next time epoch. In this example, entries clearly reflect the initial room assignment and also how prone one user is to visit other side rooms and stay in the latter (e.g. coffee room, printers' room, toilets…).

In our simulation, we assume full connectivity for cooperative peer-to-peer ranging among the 4 users for simplicity, regardless of possible non-line-of-sight propagation conditions (e.g., in case of different room occupancies).

HMM Room Trans. Matrix T @ User 1

0.2000	0.0600	0.0600	0.0600	0.0600	0.0600	0.0600	0.3800	0.0600
0.2000	0.8000	0	0	0	0	0	0	0
0.2500	0	0.7500	0	0	0	0	0	0
0.2000	0	0	0.8000	0	0	0	0	0
0.2000	0	0	0	0.8000	0	0	0	0
0.3000	0	0	0	0	0.7000	0	0	0
0.2000	0	0	0	0	0	0.8000	0	0
0.0500	0	0	0	0	0	0	0.9500	0
0.5000	0	0	0	0	0	0	0	0.5000

HMM Room Trans. Matrix T @ User 2

0.2000	0.0600	0.0600	0.0600	0.3800	0.0600	0.0600	0.0600	0.0600
0.2000	0.8000	0	0	0	0	0	0	0
0.2500	0	0.7500	0	0	0	0	0	0
0.2000	0	0	0.8000	0	0	0	0	0
0.0500	0	0	0	0.9500	0	0	0	0
0.3000	0	0	0	0	0.7000	0	0	0
0.2000	0	0	0	0	0	0.8000	0	0
0.2000	0	0	0	0	0	0	0.8000	0
0.5000	0	0	0	0	0	0	0	0.5000

HMM Room Trans. Matrix T @ User 3

0.2000	0.3800	0.0600	0.0600	0.0600	0.0600	0.0600	0.0600	0.0600
0.0500	0.9500	0	0	0	0	0	0	0
0.2500	0	0.7500	0	0	0	0	0	0
0.2000	0	0	0.8000	0	0	0	0	0
0.2000	0	0	0	0.8000	0	0	0	0
0.3000	0	0	0	0	0.7000	0	0	0
0.2000	0	0	0	0	0	0.8000	0	0
0.2000	0	0	0	0	0	0	0.8000	0
0.5000	0	0	0	0	0	0	0	0.5000

HMM Room Trans. Matrix T @ User 4

0.2000	0.0600	0.0600	0.3800	0.0600	0.0600	0.0600	0.0600	0.0600
0.2000	0.8000	0	0	0	0	0	0	0
0.2500	0	0.7500	0	0	0	0	0	0
0.0500	0	0	0.9500	0	0	0	0	0
0.2000	0	0	0	0.8000	0	0	0	0
0.3000	0	0	0	0	0.7000	0	0	0
0.2000	0	0	0	0	0	0.8000	0	0
0.2000	0	0	0	0	0	0	0.8000	0
0.5000	0	0	0	0	0	0	0	0.5000

Figure 8: Hidden Markov Model state transition matrix per user accounting for the probability of moving from one room to another (row index: room occupied at current time epoch; column index: room occupied at next time epoch).

For the same example as previously, Figure 6 and Figure 7 show respectively the true and estimated locations successively occupied by the 4 users on the floor layout on the one hand, and the room occupancy as a function of time for approximately 1 hour (still at the refresh rate of 1 min) on the other hand, while assuming $\sigma_{x,i} = \sigma_{y,i} = 1m, \forall\, i = 1..N_u$.

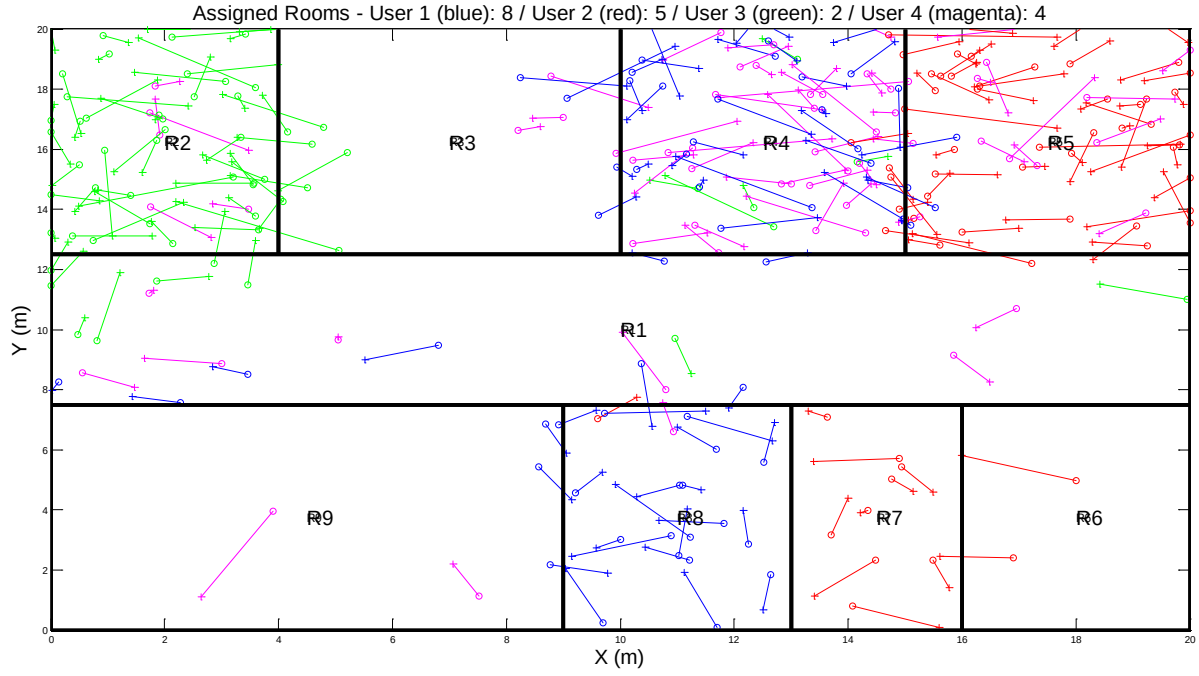

Figure 9: Example of superposition of true occupied (crosses) and claimed/estimated (circles) 2D positions over 1 hour with a refresh period of 1 min for users 1, 2, 3 and 4 (resp. blue, red, green, magenta) in a typical office building, depending on their assigned rooms (8, 5, 2, 4 resp.) and with $\sigma_{x,i} = \sigma_{y,i} = 1m, \forall\, i = 1..N_u$.

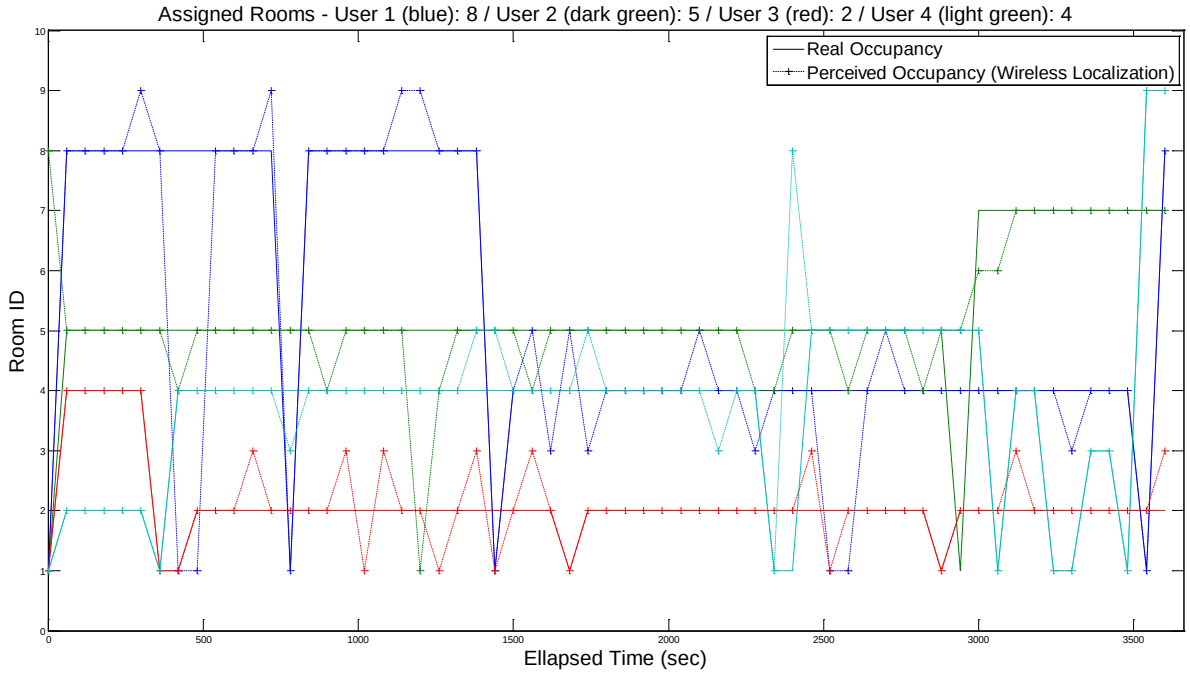

Figure 10: Example of true vs. detected room occupancy over 1 hour with a refresh period of 1 min for users 1, 2, 3 and 4 (resp. blue, dark green, red, light green) in a typical office building, depending on their assigned rooms (8, 5, 2, 4 resp.) and with $\boldsymbol{\sigma_{x,i} = \sigma_{y,i} = 1m, \forall\, i = 1..N_u}$.

4.2.2 Results & Illustrations

For the $N_u = 4$ users, Figure 8, Figure 9 and Figure 10 show the evolution over time of candidate location-based scoring schemes as possible R&T ingredients, namely:

- Instantaneous location reliability indicators
  - Instantaneous Cooperative Location Consistency (ICLC), based on peer-to-peer ranging with 1-hop neighbours and claimed estimated 2D coordinates
  - Transition Space-Time Consistency (TSTC), based on both HMM room transition probability learning and currently detected rooms based on estimated 2D coordinates
- Spatial predictability indicators
  - Mobility Learning Quality (MLQ), based on learnt HMM room transition probability
  - Non-Erratic Mobility (NEM), based on learnt HMM room transition probability

In the evaluated scenarios, we assume 3 standard devices -or equivalently 3 legitimate users- claiming erroneous positions with $\sigma_{x,i} = \sigma_{y,i} = 1m,\ \forall\, i = 1..3$ (Users 1 to 3) and 1 single faulty device -or equivalently 1 malicious user- (User 4), with $\sigma_{x,4} = \sigma_{y,4} = 2m$ (Figure 8), $\sigma_{x,4} = \sigma_{y,4} = 5m$ (Figure 9), and $\sigma_{x,4} = \sigma_{y,4} = 7m$ (Figure 10). Other parameters, α = 1 (in MLQ setting), $Th_t = 0.05$ (in TSTC setting) and $Th_d = 1$ (in ICLC setting, thus leading to $P_{FA} \approx 0.3$), are the same for the three figures. The overall simulated period lasts for 12 hours, including 3 hours of active learning at the beginning.

One first remark is that, despite very close positioning standard deviations between User 4 (2m) and the three other Users (1m), it is still possible to sort out faulty User 4 (Figure 8), especially when benefitting from full cooperation in ICLC to make the reputation scores more representative and solid (than just detecting for impossible room transition). The detection performance would be obviously improved at much larger positioning standard deviations for

User 4 (5m in Figure 9 and 7m in Figure 10). This tends to suggest also that a relevant location-based R&T scoring strategy could take mutual benefits from both, absolute positioning/navigation and peer-to-peer cooperative ranging technologies into one single indicator. Each unitary component being evaluated -by definition- as an indicator in [0,1], one could for instance envision a direct product or a normalized summation thus still lying in the interval [0,1]. Besides, the relatively high and constant NEM indicator confirms the mobility hypotheses retained in the simulated scenario (e.g., assigned offices), meaning that mobility patterns and habits are really specific from one user to the next (See the true room transition matrices at the beginning of the section). Finally, despite large 2D positioning errors at User 4, the quality of mobility learning does not seem to be significantly altered, as revealed by the grouped MLQ curves.

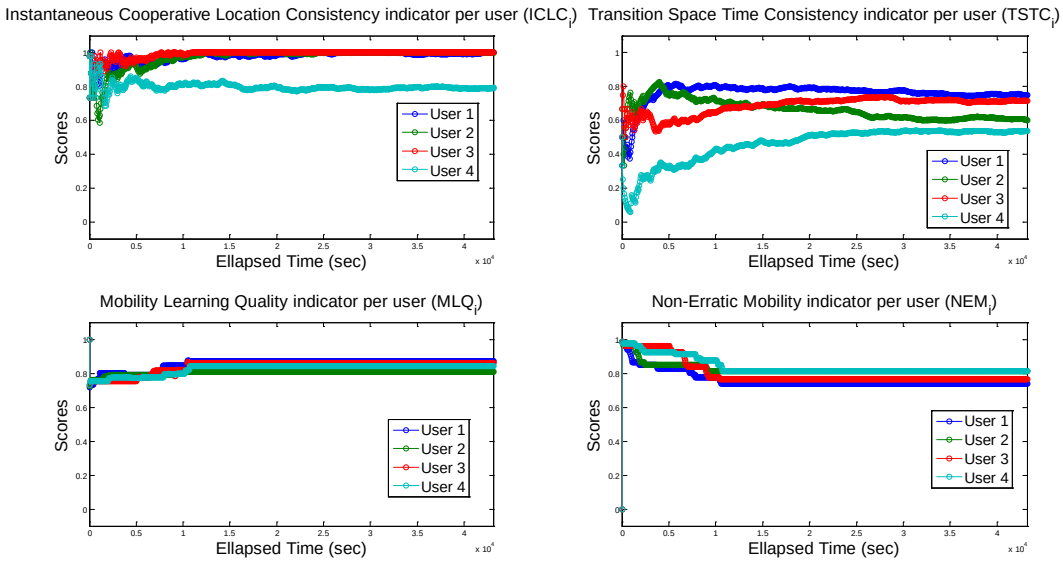

Figure 11: Evolution of candidate location-based R&T scores associated with the 4 users as a function of time, with $\sigma_{x,i} = \sigma_{y,i} = 1m, \forall\, i = 1..3$, $\sigma_{x,4} = \sigma_{y,4} = 2m$ (faulty device or malicious user), $\sigma_d = 1m$, α = 1, $Th_t = 0.05$ and $Th_d = 1$.

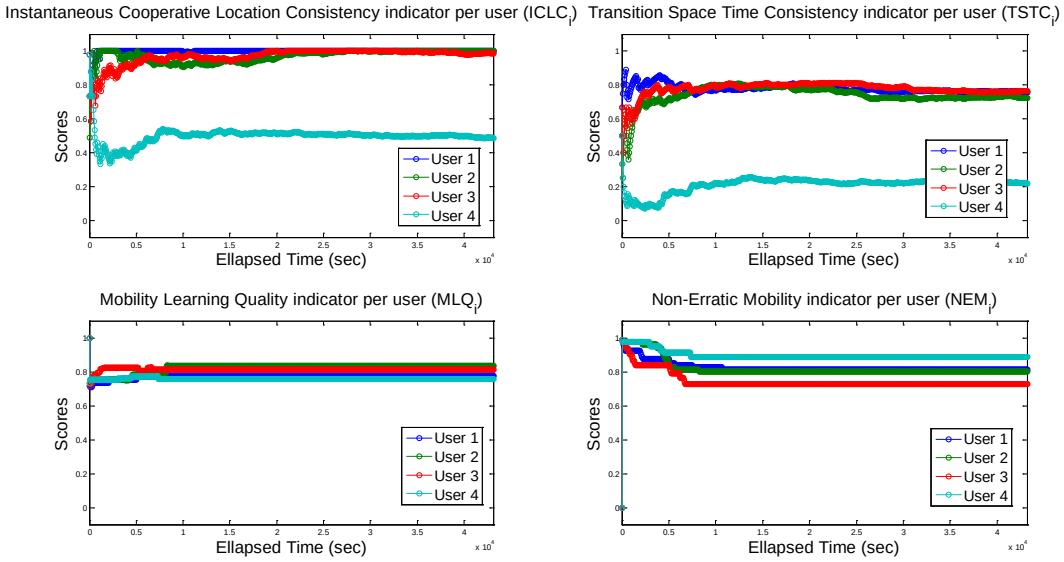

Figure 12: Evolution of candidate location-based R&T scores associated with the 4 users as a function of time, with $\sigma_{x,i} = \sigma_{y,i} = 1m, \forall\, i = 1..3$, $\sigma_{x,4} = \sigma_{y,4} = 5m$ (faulty device or malicious user), $\sigma_d = 1m$, α = 1, $Th_t = 0.05$ and $Th_d = 1$.

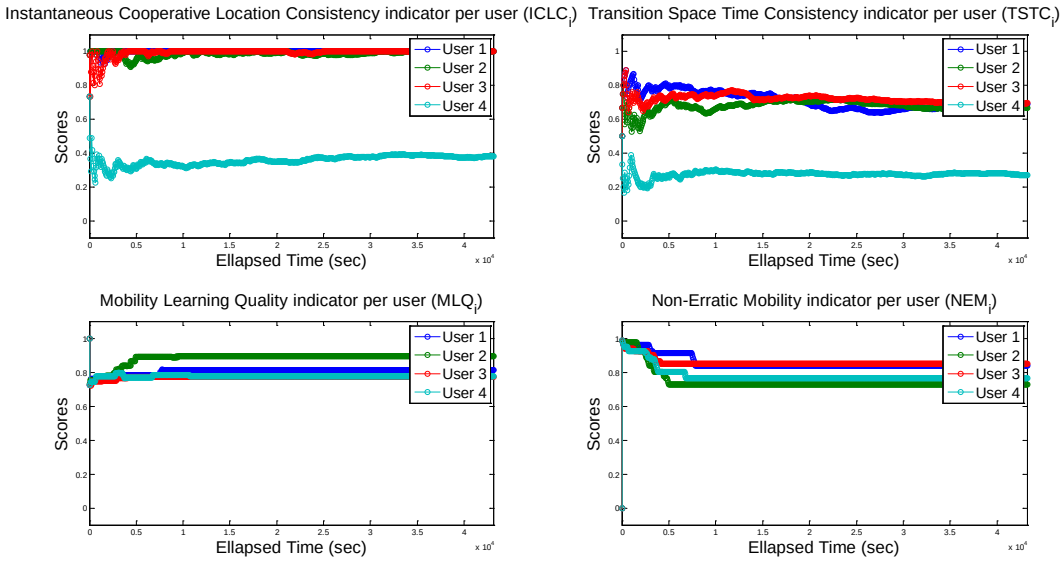

Figure 13: Evolution of candidate location-based R&T scores associated with the 4 users as a function of time, with $\sigma_{x,i} = \sigma_{y,i} = 1m, \forall\, i = 1..3$, $\sigma_{x,4} = \sigma_{y,4} = 7m$ (faulty device or malicious user), $\sigma_d = 1m$, α = 1, $Th_t = 0.05$ and $Th_d = 1$.

Figure 11 and Figure 12 show similar curves, still with $\sigma_{x,i} = \sigma_{y,i} = 1m,\ \forall\, i = 1..3$ (Users 1 to 3) and $\sigma_{x,4} = \sigma_{y,4} = 5m$ (User 4), but this time with $\sigma_d = 0.3m$ and $2m$ respectively, impacting mostly the ICLC level accordingly. All the other parameters are set as previously (i.e., α = 1, $Th_t = 0.05$ and $Th_d = 1$).

At first sight, whatever the considered scenario, the successful detection of unreliable or malicious users looks compatible with the capabilities and levels of accuracy currently available with existing radiolocation technologies, even if the use of cooperation among

users always seems preferable to reinforce the evaluation even when benefitting from relatively poor ranging resolution.

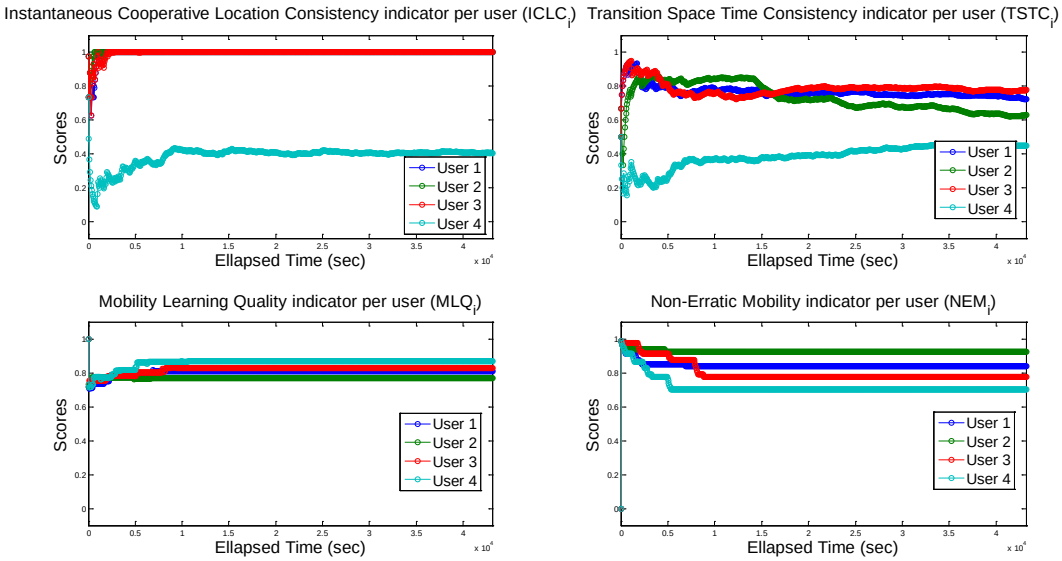

Figure 14: Evolution of candidate location-based R&T scores associated with the 4 users as a function of time, with $\sigma_{x,i} = \sigma_{y,i} = 1m, \forall\, i = 1..3$, $\sigma_{x,4} = \sigma_{y,4} = 5m$ (faulty device or malicious user), $\sigma_d = 0.3m$, α = 1, $Th_t = 0.05$ and $Th_d = 1$.

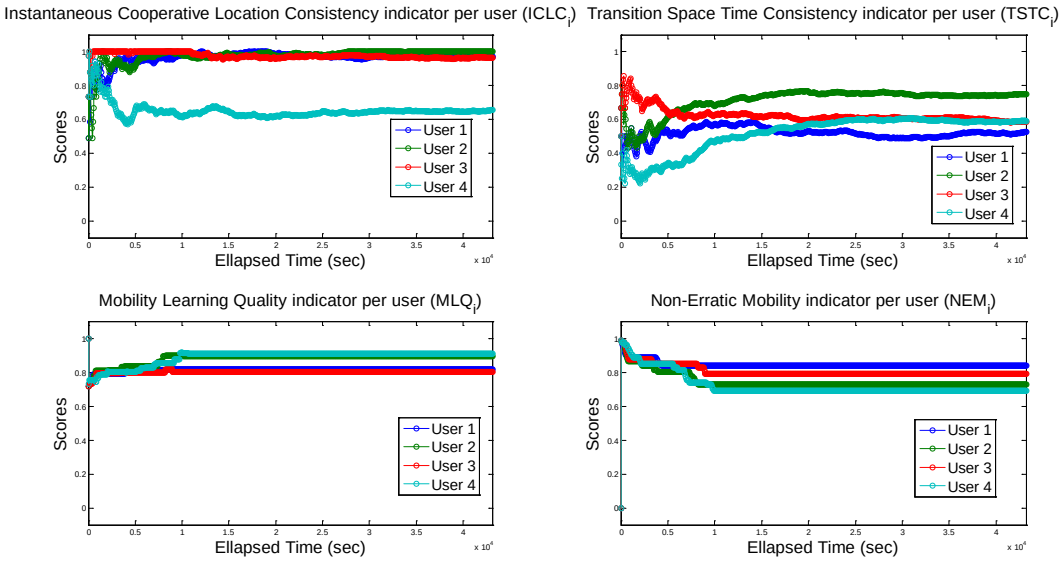

Figure 15: Evolution of candidate location-based R&T scores associated with the 4 users as a function of time, with $\sigma_{x,i} = \sigma_{y,i} = 1m, \forall\, i = 1..3$, $\sigma_{x,4} = \sigma_{y,4} = 5m$ (faulty device or malicious user), $\sigma_d = 2m$, α = 1, $Th_t = 0.05$ and $Th_d = 1$.

Figure 13 shows other illustrating results with $\sigma_{x,i} = \sigma_{y,i} = 1m,\ \forall\, i = 1..3$ (Users 1 to 3) and $\sigma_{x,4} = \sigma_{y,4} = 5m$ (User 4), $\sigma_d = 1m$, α = 1, $Th_t = 0.05$, but with $Th_d = 1.6$ (thus leading to $P_{FA} = 0.1$).

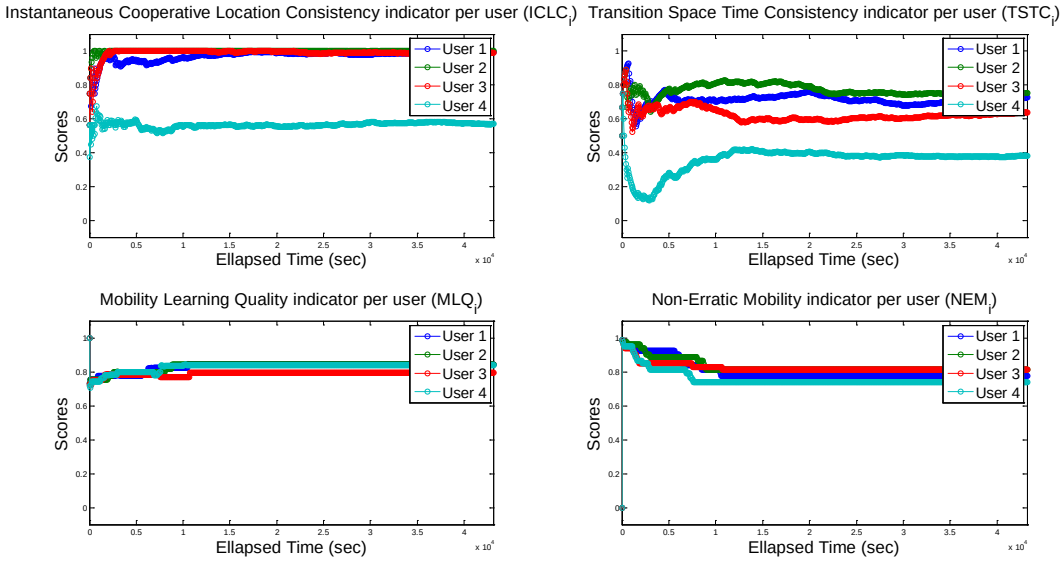

Figure 16: Evolution of candidate location-based R&T scores associated with the 4 users as a function of time, with $\sigma_{x,i} = \sigma_{y,i} = 1m, \forall\, i = 1..3$, $\sigma_{x,4} = \sigma_{y,4} = 5m$ (faulty device or malicious user), $\sigma_d = 1m$, α = 1, $Th_t = 0.05$ and $Th_d = 1.6$ ($P_{FA} = 0.1$).

Regarding ICLC more particularly, Figure 14 shows the evolution of intermediary scores $R_{ij}(k)$ computed in the first decentralized step at the different devices (about their 1-hop neighbors), before centralized averaging into $R_i(k)$ (Figure 15) and finally normalization by $1/(1 - P_{FA})$ to produce $ICLC_i(k)$ (Figure 16). The corresponding parameters are $\sigma_{x,i} = \sigma_{y,i} = 1m,\ \forall\, i = 1..3$ (Users 1 to 3) and $\sigma_{x,4} = \sigma_{y,4} = 5m$ (User 4), $\sigma_d = 1m$ and $Th_d = 1$ (thus leading to $P_{FA} = 0.30$).

One can thus remark that User 4 is rather well identified as faulty/malicious by the 3 other users already in the decentralized step. Another important point is that the behaviors of both a standard/legitimate and faulty/malicious users can be theoretically predicted through analytical derivations and simplified hypotheses such as full-connectivity and collinearity of ranging and 1D positioning errors (Figure 15 & Figure 16). Hence, adaptive threshold setting strategies can be employed to detect too low "R&T" scores and launch alerts (Figure 17) (e.g., Threshold setting 1: put the threshold half-way between the expected "good guy" and the expected "bad guy" at each time-stamp to optimize the decision vs. Threshold setting 2: put a constant threshold sufficiently close to the expected maximum score). The adaptive threshold setting strategy (Setting 1) then rather clearly outperforms the non-adaptive strategy.

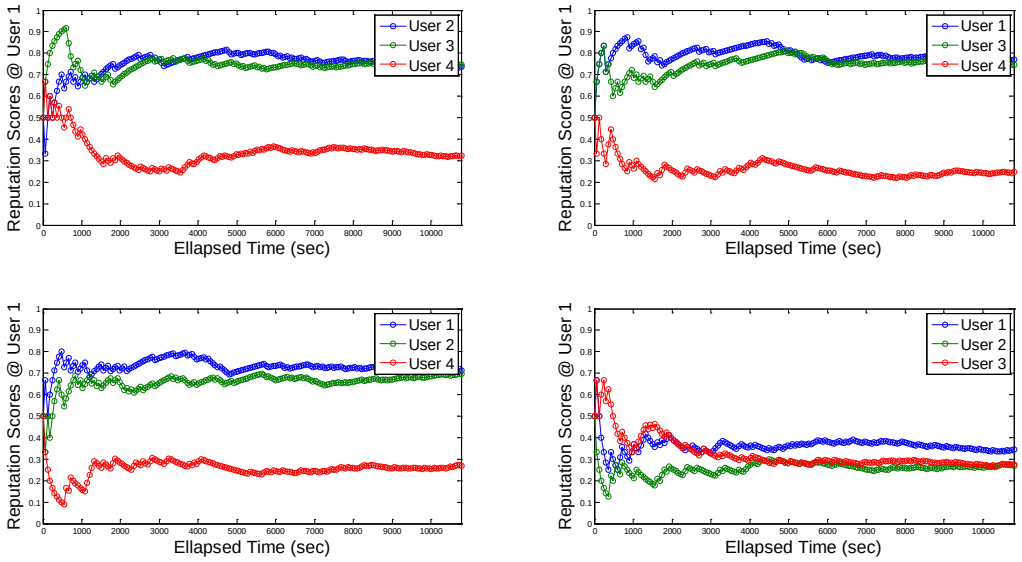

Figure 17: Evolution of ICLC's intermediary $R_{ij}(k)$ scores (decentralized) at the devices (about their 1-hop neighbors), with $\sigma_{x,i} = \sigma_{y,i} = 1m$, $\forall\, i = 1..3$ (Users 1 to 3) and $\sigma_{x,4} = \sigma_{y,4} = 5m$ (User 4), $\sigma_d = 1m$ and $Th_d = 1$ (thus leading to $P_{FA} = 0.30$).

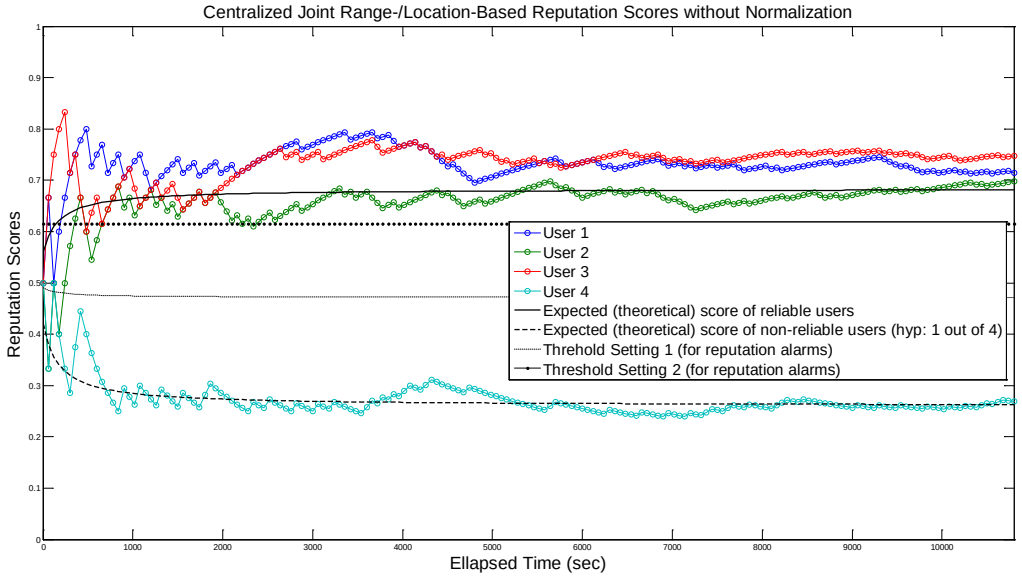

Figure 18: Evolution of ICLC's average $R_i(k)$ scores (centralized) associated with the different users, with $\sigma_{x,i} = \sigma_{y,i} = 1m$, $\forall\, i = 1..3$ (Users 1 to 3) and $\sigma_{x,4} = \sigma_{y,4} = 5m$ (User 4), $\sigma_d = 1m$ and $Th_d = 1$ (thus leading to $P_{FA} = 0.30$).

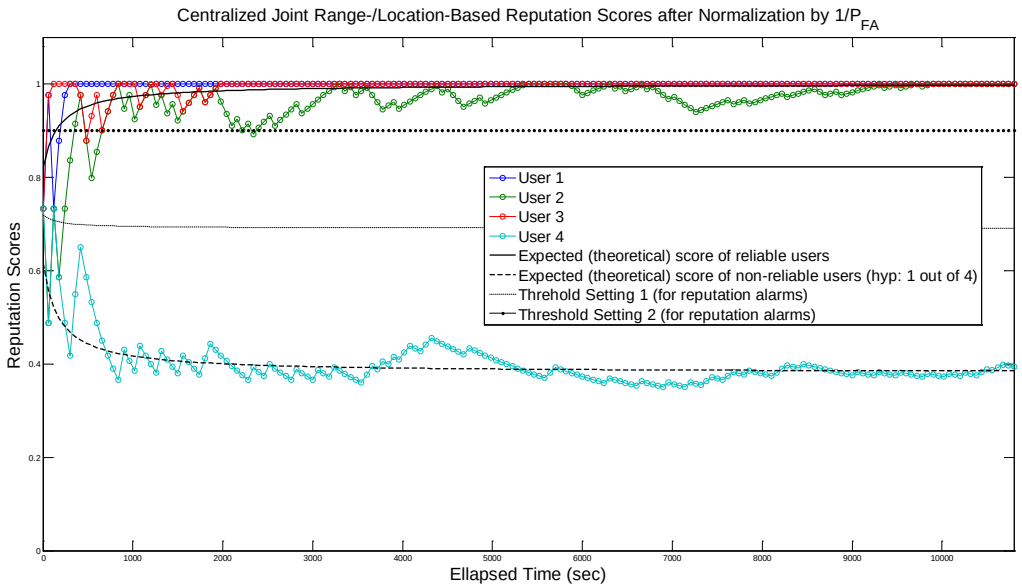

Figure 19: Evolution of average & normalized $ICLC_i(k)$ scores (centralized) associated with the different users, with $\sigma_{x,i} = \sigma_{y,i} = 1m,\ \forall\, i = 1..3$ (Users 1 to 3) and $\sigma_{x,4} = \sigma_{y,4} = 5m$ (User 4), $\sigma_d = 1m$ and $Th_d = 1$ (thus leading to $P_{FA} = 0.30$).

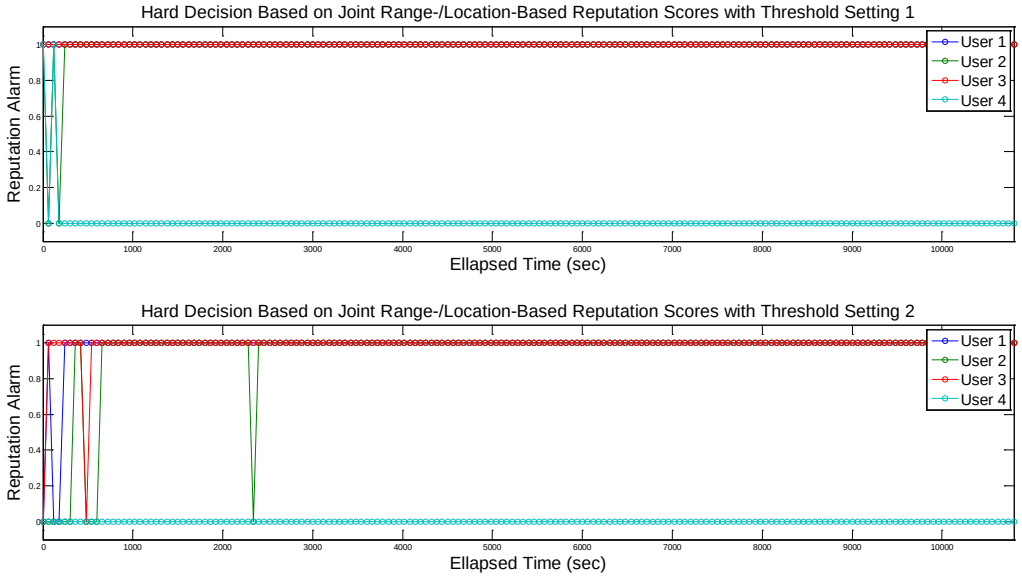

Figure 20: "R&T alert" events based on average & normalized $ICLC_i(k)$ scores and the detection threshold settings of Figure 16.

Now, we aim at evaluating the ability to capture erratic mobility patterns with our location-based R&T scoring strategy, independently of the radiolocation systems accuracy (i.e., assuming $\sigma_{x,i} = \sigma_{y,i} = 1m,\ \forall\, i = 1..4$ and $\sigma_d = 1m$). For this sake, we just modify the HMM state transition matrix of User 4 with equiprobable room changes as shown on Figure 18, while keeping the same refreshment period of 1 min as a basis. This accounts for a user who is prone to change its room occupancy at each epoch without particular sense or, in other words, that one cannot really tell which room would be likely occupied at the next epoch

considering the room occupied at the current epoch. The transition matrices of other users remain unchanged in comparison with previous scenarios.

HMM Room Trans. Matrix T @ User 4

0	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250
0.1250	0	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250
0.1250	0.1250	0	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250
0.1250	0.1250	0.1250	0	0.1250	0.1250	0.1250	0.1250	0.1250
0.1250	0.1250	0.1250	0.1250	0	0.1250	0.1250	0.1250	0.1250
0.1250	0.1250	0.1250	0.1250	0.1250	0	0.1250	0.1250	0.1250
0.1250	0.1250	0.1250	0.1250	0.1250	0.1250	0	0.1250	0.1250
0.1250	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250	0	0.1250
0.1250	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250	0.1250	0

Figure 21: Hidden Markov Model state transition matrix for User 4 with erratic mobility habits (in terms of room changes), reflected by equiprobable matrix entries.

From Figure 19, it can thus be noticed that despite the good quality of the learning process for the 4 users (See MLQ indicators) and even better results for User 4 due to a closer initial guess regarding equiprobable room transitions, the NEM indicator correctly reveals that User 4's behavior is indeed hardly predictable, as expected, and thus hardly exploitable to announce e.g., the spatial utility of the very user in a near future with respect to a priori spatial goals of a community.

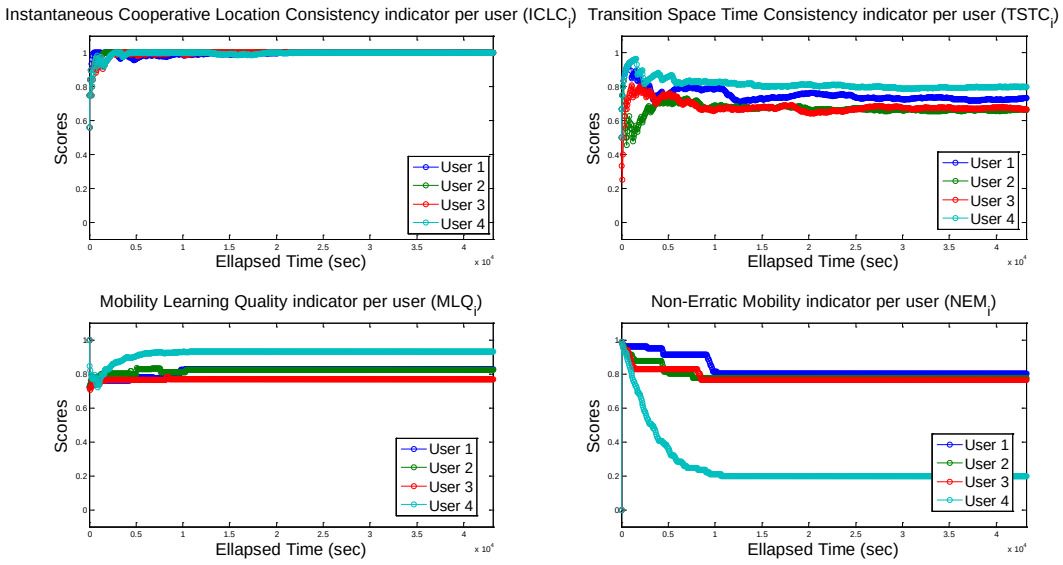

Figure 22: Evolution of candidate location-based R&T scores associated with the 4 users as a function of time, with $\sigma_{x,i} = \sigma_{y,i} = 1m, \forall\, i = 1..4$ and $\sigma_d = 1m$, α = 1, $Th_t = 0.05$ and $Th_d = 1.6$ ($P_{FA} = 0.1$).

Regarding the latter so-called spatial reputation and besides the location consistency or predictability issues illustrated so far, we finally show hereafter how location-based R&T rating mechanisms could be beneficial into the specific context of distributed participator sensing, by indicating how prone each particular user is to meet global spatial needs expressed by the community (based on his learnt mobility habits). One step ahead, note that this part of the R&T scoring can thus be used to trigger or induce new positive behaviors (with respect to the same expressed global goals again) by naturally increasing the user's score in an incentive way.

In the next scenario below, which is simulated over a shorter duration, we stick with the latest initial nominal simulation set-up (i.e., including the same HMM state transition matrix for User 4 as before), but we now show the evolution of the cumulative number of visits as a function of time (Figure 20 and Figure 21) versus the spatial utility SU scores computed per user according to the definition given in 3.1.2.3 (Figure 22). In other words, we represent the expected probability for each user that he will visit at least one of the priority rooms (i.e., the room(s) with the lowest location entropy or equivalently, with the highest spatial desirability) within an arbitrary period of time (i.e., depicted as a horizon of time (HoT)). The evaluation is performed based on the successive powered versions of the learnt HMM room transition matrix (by construction lying in [0, 1]).

In this example, one can first notice that the obtained SU scores are very high at the beginning of the acquisition, since few of the rooms have been visited already (and thus, many of them have the same high spatial desirability) and accordingly, the probability to visit at least one of them is high. Then in steady-state regime, the obtained SU scores are temporarily increased when several rooms exhibit the same minimum number of visits so far, and thus, the probability for any user is higher to visit at least one of those rooms (instead of one). This is due to the fact that we have arbitrarily chosen to assign the highest spatial desirability to the rooms that have been less visited so far, while authorizing that several rooms can share the same minimum number of visits. Another remark is that the SU scores of users exhibiting non-erratic mobility (i.e., with established and repeated mobility patterns, as reflected by sparser state transition matrices), namely Users 1 to 3, can suddenly increase when they visit a room that offers a high probability potential of transition to the required room in the short term (typically in the corridor). If the considered HoT is short, SU is strongly sensitive to the currently occupied room whereas larger HoT values (i.e. larger power exponents of the HMM state transition matrix) lead to smoother variations but the average probability level increases since it is more likely to find a sequence of state transitions that will reach the required room with high spatial desirability, regardless of the initial occupied room. As for User 4, higher SU values correctly captures the fact that his erratic mobility provides more chances to visit the disfavored rooms, and thus, better spatial utility with respect to global community goals.

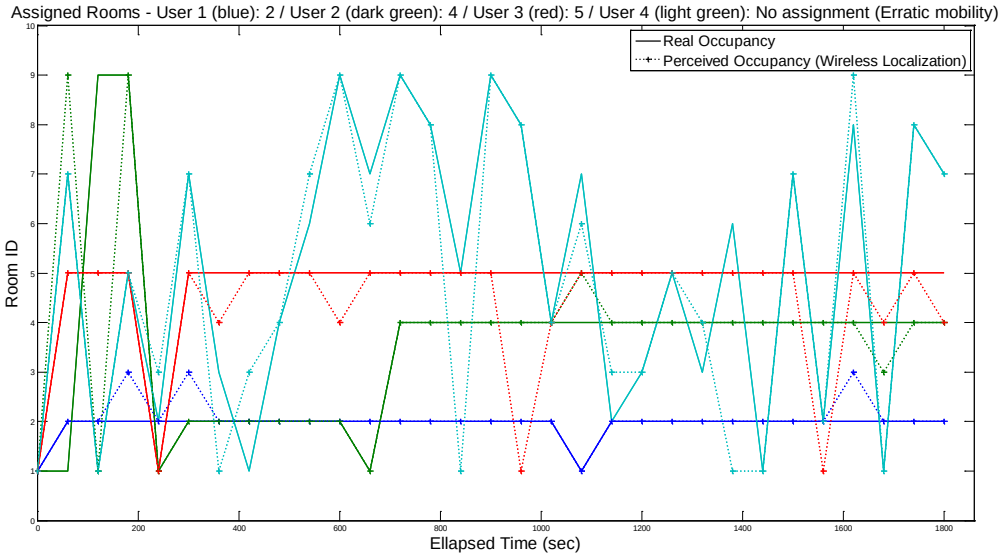

Figure 23: Example of true vs. detected room occupancy over 30 min with a refresh period of 1 min for users 1, 2, 3 and 4 (resp. blue, dark green, red, light green) depending on their assigned rooms (8, 5, 2, no assignment resp.), with $\sigma_{x,i} = \sigma_{y,i} =$

$1m, \forall\, i = 1..N_u$. Contrarily to other users, User 4 exhibits erratic mobility according to the HMM state transition matrix of Figure 18.

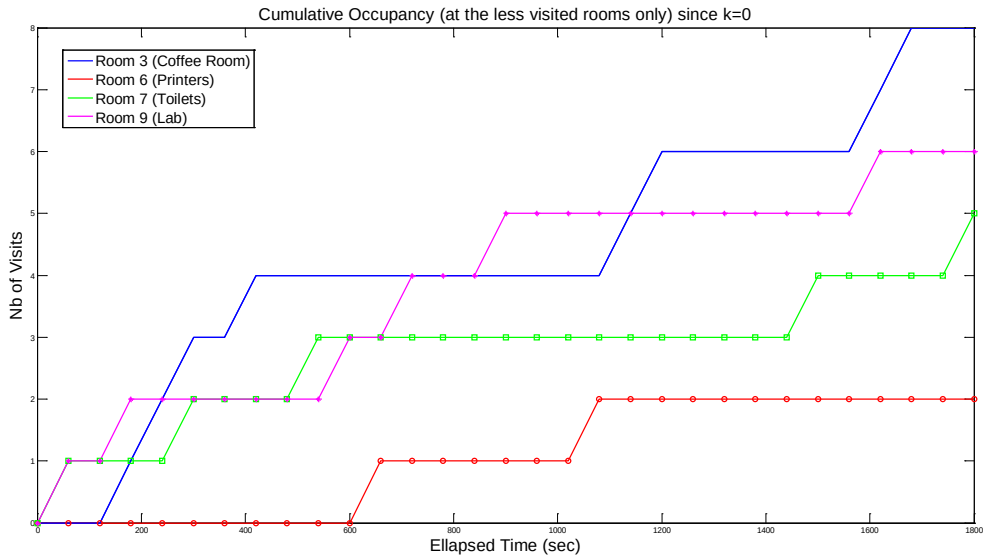

Figure 24: Cumulative perceived occupancy per room since the beginning of acquisition in the scenario of Figure 20, as a function of time, at rooms 3, 6, 7 and 9.

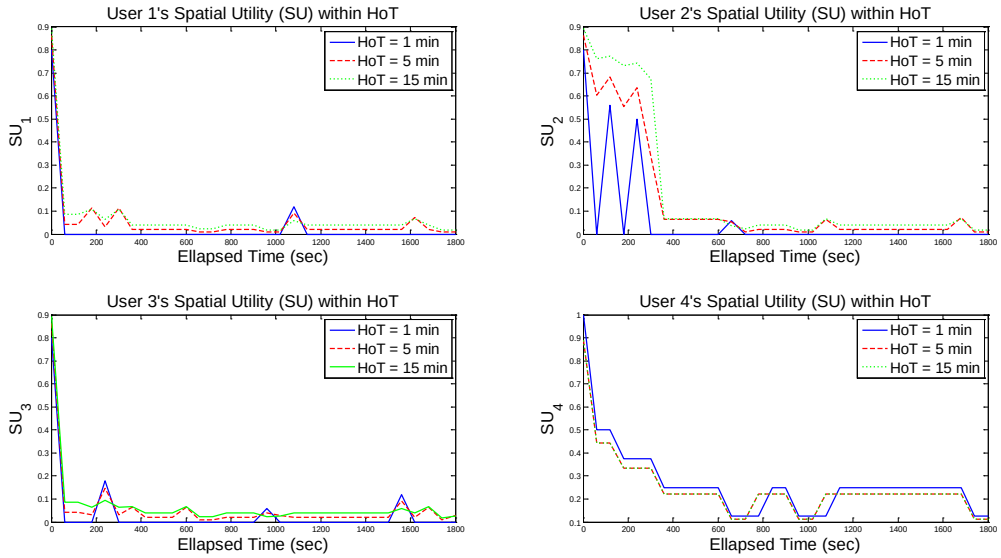

Figure 25: Example of location-based Spatial Utility (SU) per user, calculated based on the spatial reputation. The latter is calculated as the probability to visit within different horizons of time (HoT) the room that has been less visited so far (i.e. up to the current time epoch).

4.2.3 Conclusions

To summarize, the previous simulation-based evaluations have illustrated how location-based R&T scoring mechanisms:

- Can capture and integrate user's habits (including highly repetitive or erratic ones), as part of the "spatial reputation" through mobility learning;

- Contribute to successfully detect faulty/non-reliable devices or malicious nodes claiming wrong/erroneous locations based on conventional radiolocation technologies;
- Benefit from cooperation among users in Bubbles of Trust to reinforce detection, by integrating peer-to-peer ranging measurements;
- Are moderately sensitive to both ranging and positioning accuracy, although practical analytical detection thresholds can be set a priori;
- Are deliberately modular by design, authorizing the incorporation of gradual location-based information depending on devices' capability (e.g. absolute position coordinates, set of range measurements w.r.t. 1-hop neighbours, or a combination of both);
- Look particularly relevant and promising in application contexts requiring the delivery of georeferenced data under non-controlled users' mobility (e.g., participatory sensing).

4.3 User Behaviour

This subsection provides an evaluation of our proposed method for gait recognition. We first present the evaluation methodology, describe the implementation details of the benchmarking techniques and then present the results. In order to measure the performance of the gait recognition algorithm, both a real-world data set and a real-world scenario is used. The main objective of the evaluation is to investigate the performance of the algorithm and compare it with a current state-of-the-art approach.

4.3.1 Evaluations Set-up

For a realistic gait recognition scenario, the owner of the smart phone trains the model using an example of their walking. This is then compared with the walking patterns of the rightful user and other users who are considered imposers. This scenario corresponds to a real-world situation where the aim is detect when the rightful user of the phone is no longer in possession, but rather an imposter has possession of the smart phone.

The proposed scenario creates a semi-supervised one-class classification problem. In this problem, we have access to example data of the rightful user, this is termed normal data. The data is known to be that of the rightful user, and therefore can be labelled as normal data. Using this data, a model of the gait of the rightful user in constructed using the algorithm detailed in Section 3.3.2. In the classification stage, this model is used to label data generated from the accelerometer as either normal (for the rightful user) or anomalous (for the imposter).

4.3.2 Performance Metrics

As stated previously, during classification the data instances are labelled either normal or anomalous. The data associated with the rightful user is labelled negative, and the data associated with the imposter is labelled positive. From this, metrics can be determined based on whether the label matches the ground-truth label. True positives and true negatives occur when data instances are correctly labelled as anomalous and normal respectively. False positives and false negatives occur when data instances are incorrectly labelled as anomalous and normal respectively. Ratios using these metrics offer statistics that summarize the performance on the testing data set. The true positive rate (TPR), false positive rate (FPR), false negative rate (FNR) and true negative rate (TNR) are detailed in Equation 9.

$$TPR = \frac{TP}{P}, FPR = \frac{FP}{N}, FNR = \frac{FN}{P}, TNR = \frac{TN}{N}$$

Equation 20: Performance metrics

To compare schemes, receive operating characteristic (ROC) curves are generated by varying the threshold parameter used to determine the distance. The resulting FPR and TPR were used to form the ROC curve. A summary of the ROC curve is provided by measuring the area under the ROC curve (AUC). An AUC value of 1 represents 100 per cent accuracy, and a value of 0.5 indicates performance equivalent to the random assignment of labels.

4.3.3 Parameter Optimization

In order to determine the optimal parameters, cross-validation is used. A section of 1500 contiguous accelerometer data, which is labelled walking data, was chosen for each user. The first 500 data instances were used for training, and the following 1000 data instances were used for testing. A cross-validation approach was then used to determine the optimal parameters. The parameters that yielded the highest AUC score across all users were chosen to be used in the evaluation. The aim was to determine a set of parameters that could be used for all users. Table 1 shows the optimal parameters determined that were used in all the evaluations.

Table 1: Parameters

Algorithms	Parameters
GeTeM [43]	Embedded Dim = 40, No. components = 22, k-NN = 7, Sequence Length = 50
KPCA	Embedded Dim = 40, No. Components = 140, σ = 12

4.3.4 Real-world data sets

In this section, the algorithm is evaluated offline using a data set gathered from a smart phone, the McGill walking data sets [43]. This data set was chosen as it had been previously used to evaluate gait authentication algorithms in previous research. Additionally, the data set was collected over two different days which enables the evaluation of an algorithm's robustness to change of footwear and clothing. The data set contains data which was collected on two different days from 20 subjects. The data was collected over 15 minutes whilst walking on a variety of surfaces with differing slope levels. The smartphone was located in the subjects' pocket and there was no restriction on the user's clothes. Therefore there is a variation in clothing and footwear over the two days allowing the examination of robustness to change in clothes. The walking segments were extracted as detailed in [43].

The smartphones collected the accelerometer data as quickly as possible, with the mean frequency of collection being 28.57 Hz [43]. As in [43][39], this is resampled using linear interpolation to a fixed frequency of 25 Hz. The magnitude of the accelerometer is then determined and the gravitational constant is subtracted.

For the evaluation, a training set of 500 data contiguous data instances was selected at random from the data set. This was used to train the models. Once again, a 20-fold cross-validation approach was used to determine performance. From the data set, 20 random subsets of 5000 contiguous walking data instances, minus the training data for the normal user, were chosen for testing. Performance metrics were measure for each fold.

The performance for the gait recognition algorithm on the McGill dataset is shown in Figure 23. The performance is compared with GeTeM algorithm [43]. The parameters used in the evaluation are shown in Table 1 and both the mean and the standard deviation of the 20 folds are shown. In Figure 23, the AUC for GTM and KPCA are shown using the Day 1

McGill data set for both training and testing, with the training set being disjoint from the testing set. It can be seen that the algorithm has a higher AUC for 12 out of 20 users from the McGill data set. In general, the KPCA gait recognition algorithm exhibits a performance increase of between across all 20 users of an AUC score of 0.04. This represents an increase in the number of data instances correctly identified as either normal or anomaly.

It can be seen that for some subjects there is a large variation in the standard deviation over the 20 folds. During the gathering of the data, the users walked over a variety of surfaces. If the model was trained on a different surface to that which the test data is generated from, this might results in a slightly different walk pattern. An example of this is walking on a level surface compared to walking up or down a slope. This will cause lower levels of accuracy when compared to evaluations between the same surface.

In Figure 24, performance is shown on the two different days for the McGill data set. The first day is used as the training data, with 500 contiguous data instances selected at random for each fold. The constructed model is used to classify 5000 contiguous walking data instances from the second day. There is a decrease in performance compared to the evaluation on data collected on the same day for both KPCA and GTM. This is to be expected as a change of clothing on the second day can alter the walking style which can cause difficulties in the model classifying the walking data correctly. However, KPCA still outperforms GTM on 13 of the subjects, with an increase in performance across all subjects of an AUC score of 0.06.

ROC curves for selected users are shown in Figure 25 and Figure 26. They show the ROC curve for the selected subject for the testing data set of day 1 and day 2, respectively.

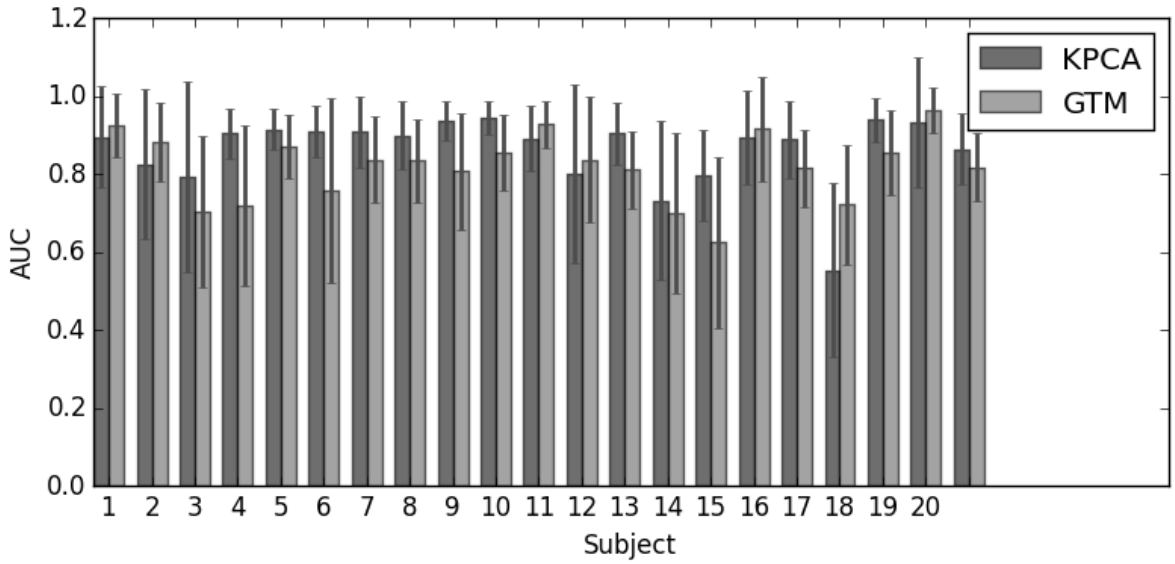

Figure 26 A comparison of the performance of KPCA gait recognition and GTM on the McGill data set using the same day. A 20-fold cross-validation approach is used with the parameters detailed in Table 1.

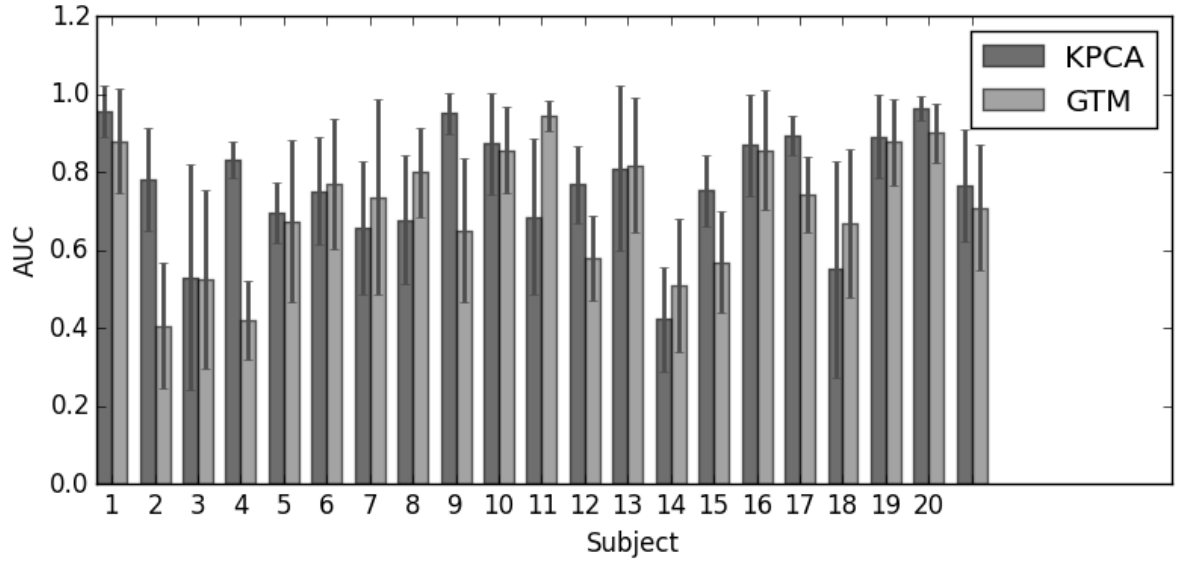

Figure 27 A comparison of the performance of KPCA gait recognition and GTM on the McGill data set using the same day. A 20-fold cross-validation approach is used with the parameters detailed in Table 1.

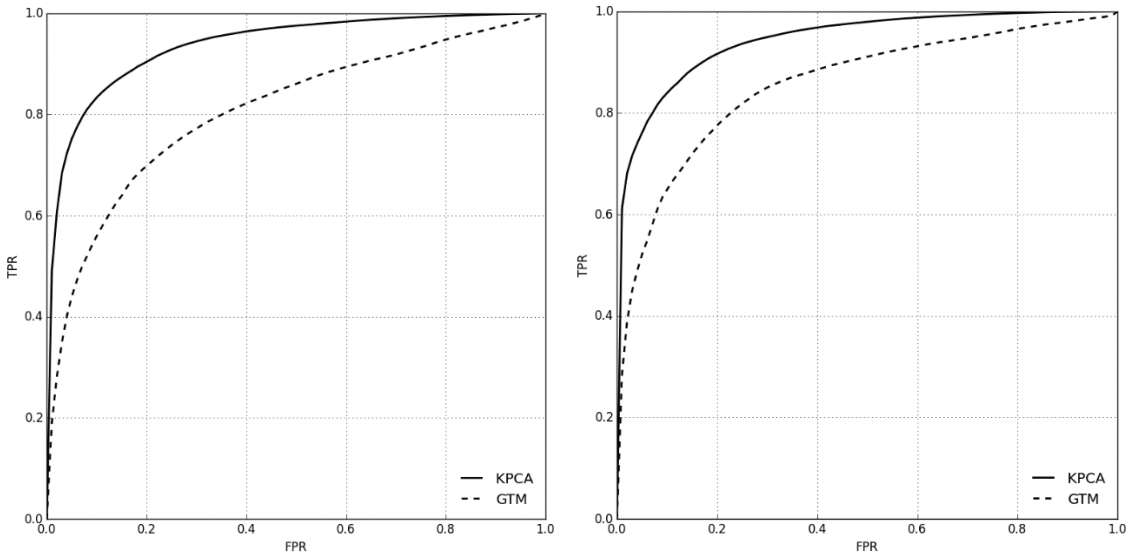

Figure 28: ROC Curves for Day1 vs Day1

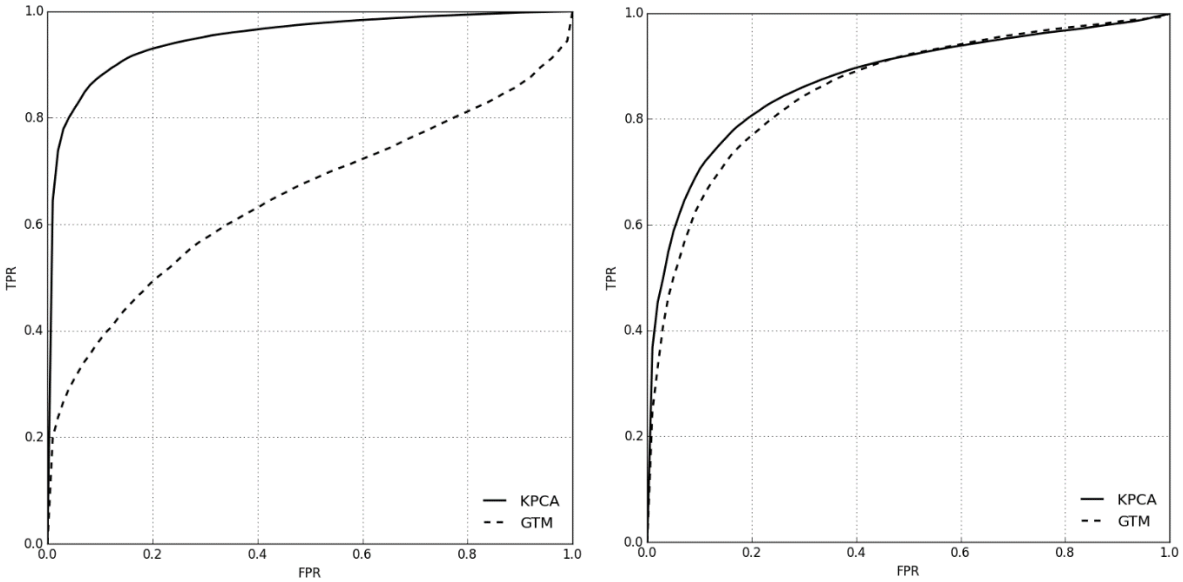

Figure 29: ROC curves for day 1 vs day 2

4.3.5 Real-world scenarios

In this section, the algorithm is evaluated in real-world environment with five users. The group consisted of 5 adults, 2 males and 3 females.

The environment used for testing was an urban one, as it is envisaged that it is in this environment that the phone is most likely to be stolen. The users were required to walk normally around a small town. For the training section, the subjects were instructed to walk in a straight line in order to provide a sample of their walking gait. However, in the testing section there were no restrictions on where and how the subjects walked, this included walking up and down kerbs, and moving around other people walking in the street. During the testing section, 10 periods of data were collected in order to perform recognition ten times over a period of ten minutes. All users placed the mobile phone in their front right pocket.

Figure 27, Figure 28, Figure 29, Figure 30 and Figure 31 show the evaluation of the model with the each of the subjects. Using the model constructed in training, the testing data is projected onto the model and the RE is determined. A smaller RE means that the data is more similar to the data that was used to construct the model. The mean of the RE is used as the summary statistic for the threshold with the threshold being set at two standard deviations above the mean, as detailed in Section 2.3.3.

Comparing the rightful users (black lines) with the imposters (coloured lines) it can be seen that it is possible to distinguish between the walking pattern of the rightful user and the imposter based on the value of the mean of the reconstruction error. In addition, the threshold, which is calculated from a reserved section of the training set, can be used to identify the imposters. For some subjects, such as 5 and 2, there is a clear difference between the rightful user and all the imposters, with all the imposters having a mean reconstruction error above the threshold. For other subjects, there can be occasional confusion where the imposter has a mean reconstruction error below the threshold, but this usually only occurs for a small number of recognition evaluations.

Looking at each subject in more detail, it can be seen for subject 1, Figure 27 that the mean RE of the rightful user is significantly lower than that of two of the imposters, who have a mean RE in the order of 1.0. For the other two imposters, the mean RE is closer to that of the threshold. This means that the walk is considered to be more similar to the rightful user than

the other subjects. One of the imposters has values that always lie above it, meaning that they will always be detected as an imposter. The other has mean RE values that dip below the threshold three times, meaning that for these recognitions they will falsely be identified as the rightful user.

The results for subject 2 are presented in Figure 28. This subject has a lower threshold than the previous subject. All the imposters have mean REs above the threshold, indicating that they will all be correctly identified as being the imposter. With this subject, there are two recognitions of the rightful user that lie above the threshold meaning that, of the ten recognitions, two will be incorrectly identified as the imposter.

Subject 3 is presented in Figure 29. For this subject, there is an imposter whose mean RE constantly lies below the threshold, meaning that they will be incorrectly identified as the rightful user. However, the other three imposters lie above the threshold, ensuring correct identification of these three imposters.

The results for subject 4 are depicted in Figure 30. With this rightful user, the mean REs are closer in value. However, the threshold largely performs its duty in discriminating between the rightful users and the imposters. Once again, there are three occasions when the rightful users is falsely identified as an imposter, but on the other 7 occasions they are correctly identified. All the imposters are correctly identified except for imposter 2 who is incorrectly assigned as the rightful user once.

Subject 5, Figure 31, shows the best performance of the algorithm. The mean RE of the rightful user is significantly lower than that of the imposters. The threshold determined by the section of reserved training data clearly indicates the imposters, with no false negatives. This indicates that the model constructed for the rightful user is able to clearly discriminate between the rightful user and the four imposters.

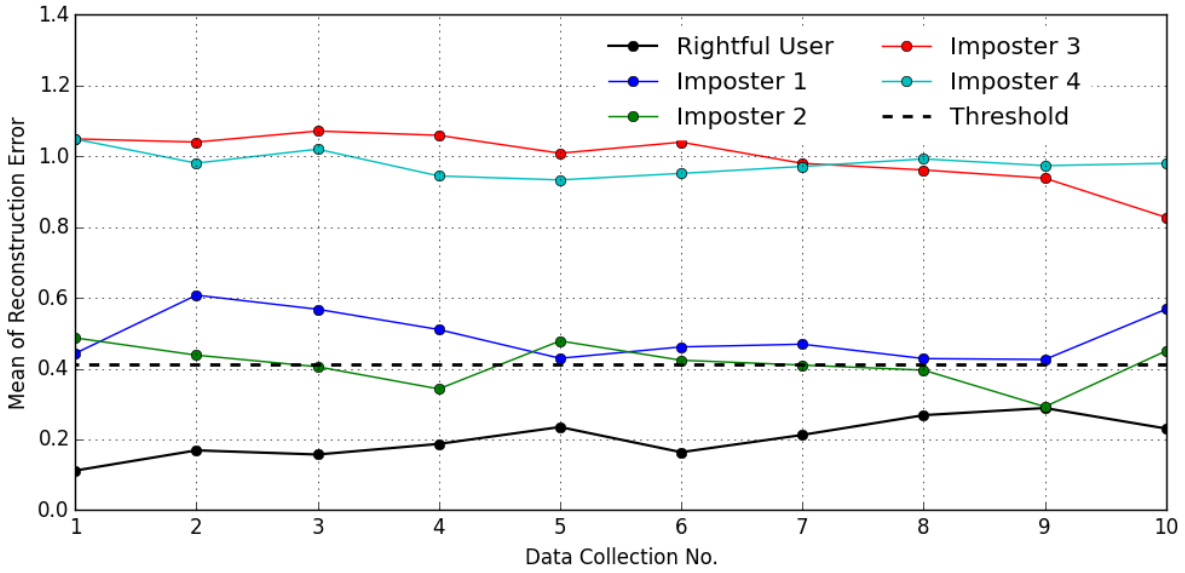

Figure 30: Subject 1 - Ten users recognitions evaluations of the rightful user compared to other subjects.

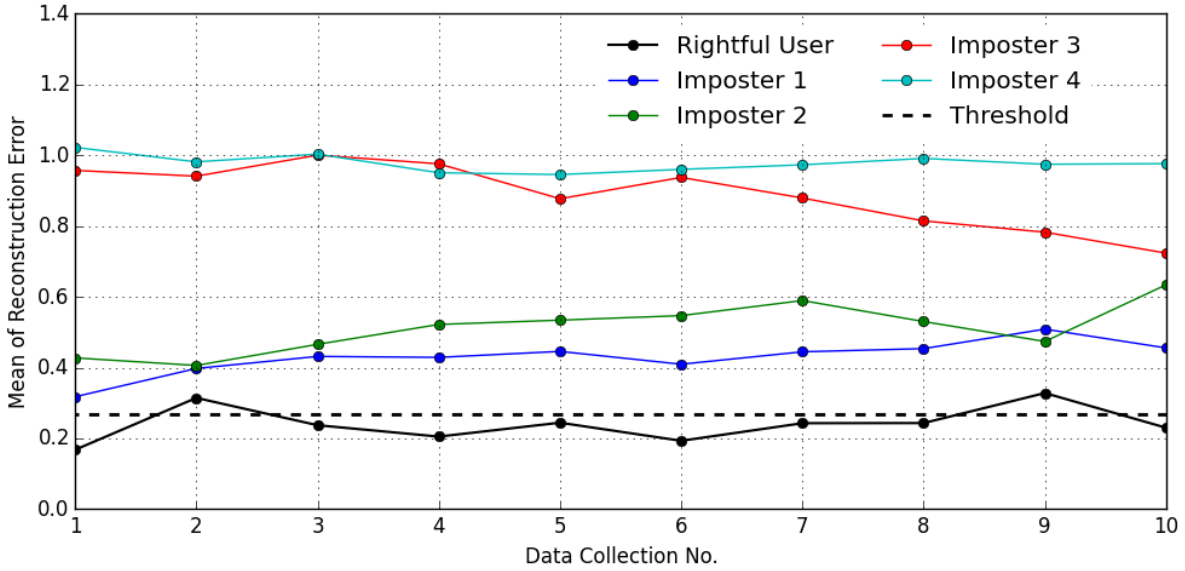

Figure 31: Subject 2 - Ten users recognitions evaluations of the rightful user compared to other subjects.

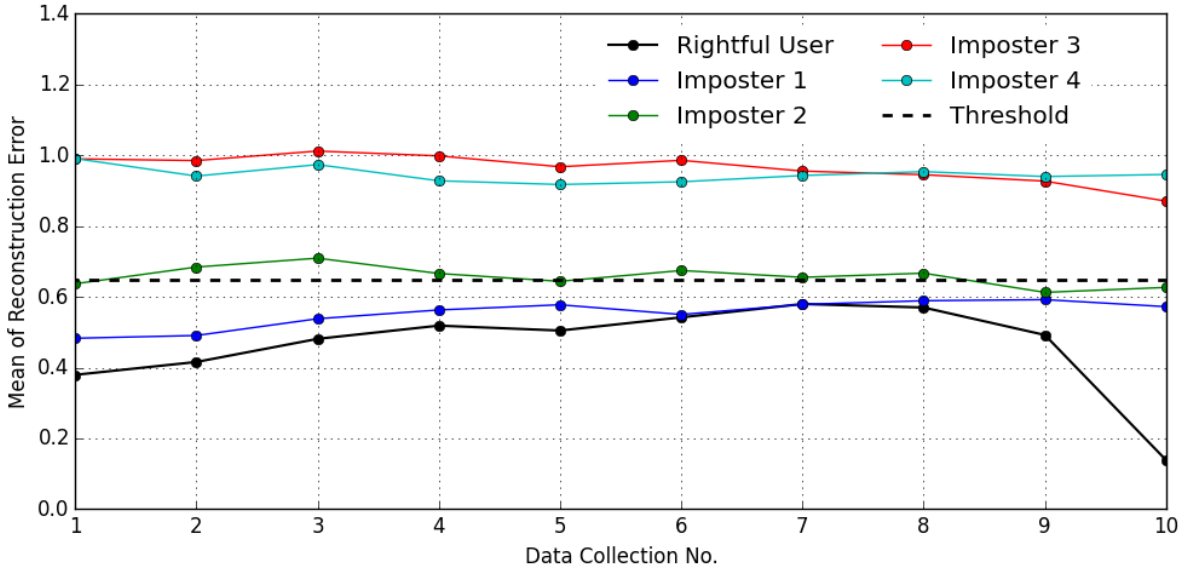

Figure 32: Subject 3 - Ten users recognitions evaluations of the rightful user compared to other subjects.

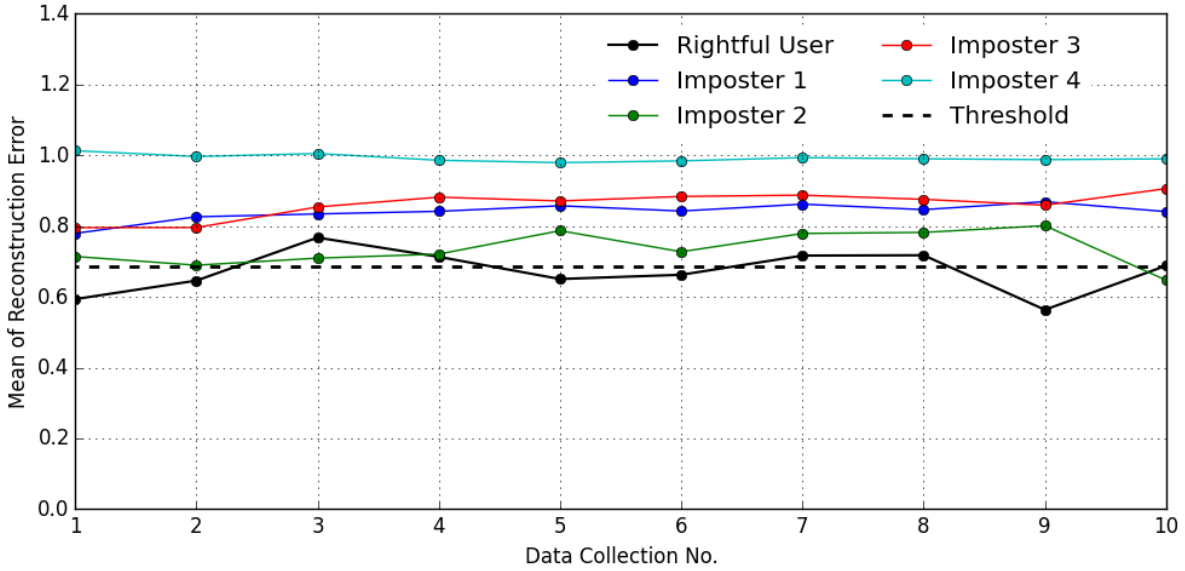

Figure 33: Subject 4 - Ten users recognitions evaluations of the rightful user compared to other subjects.

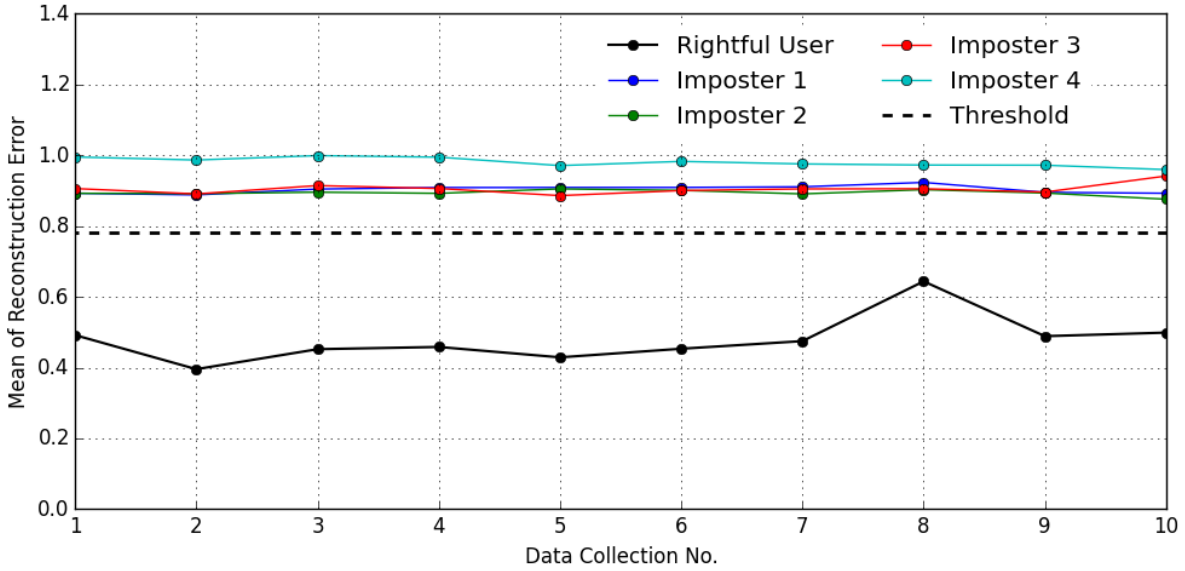

Figure 34: Subject 5 - Ten users recognitions evaluations of the rightful user compared to other subjects.

4.3.6 Conclusions

To summarize this evaluation section, it has been shown that the non-linear signal model is able to construct models of the gait of humans more accurately than a state-of-the-art method. Trials of method with five subjects have shown that this can translate into an algorithm that is able to distinguish between the rightful owner of the phone, and imposters. It has been shown that the algorithm is able to do this more accurately for some subjects than for others, and that in some cases the rightful user is misclassified as the imposter, and vice versa.

Section 5 - Conclusions

This deliverable has presented the research associated with Tasks 2.2 and 2.3. These tasks have examined the trust and reputation management framework which is being devised for the SocIoTal EU project.

This document has detailed three enablers by which trust and reputation can be determined by the SocIoTal framework. Smart phones contain a plethora of data providers, such as on-board sensors and signal generators such as Bluetooth. These data providers have been exploited in order to obtain information which can be used to determine trust and reputation. For the face-to-face enabler app, the Bluetooth signal was used in order to infer social relationships. On-board accelerometer sensor was used by a gait recognition app in order to infer whether the user in current possession of the phone is the rightful user based on their way of walking. In the final app, sensed and predicted location was used in order to determine a trust and reputation score for the possessor of the smart phone.

The evaluation of two of the enablers was presented in Section 4. First of all, the performance of location-based R&T scoring mechanisms has been assessed through realistic simulations in a typical indoor office environment under pedestrian mobility. The provided illustrations have confirmed that the medium-term mobility habits of users could be advantageously integrated as part of their spatial reputation through standard HMM learning (under reasonably low refreshment rates and relatively raw radiolocation precision). In particular, it has been shown that faulty/non-reliable devices or malicious nodes claiming erroneous locations could be successfully detected by checking the validity of their claimed spatial information over both space and time, while considering conventional radiolocation technologies and accuracies (typically, peer-to-peer ranging over Zigbee RSSI or IR-UWB RT-ToF and coarse absolute positioning through WiFi or even GPS in outdoor environments). While authorizing the integration of various kinds of location-based attributes depending on local devices' capability (e.g. absolute position coordinates, detected room occupancy, set of range measurements w.r.t. 1-hop neighbours, or combination of the latter), the proposed schemes can take opportunistically benefits from cooperation in Bubbles of Trust and beyond, various ingredients (regarding spatial predictability, spatial consistency, and spatial utility) can be composed to form the final location-based R&T score depending on the underlying application. As an example, the specific context of participatory sensing has been considered, showing how users' spatial reputation could be confronted to common spatial goals shared by a community (e.g. spatial "desirability maps" according for the number of visits to each room in the past).

The performance of the gait recognition enabler was evaluated against another state-of-the-art method using a data set gathered from 20 subjects walking on two different days. The proposed algorithm was shown to increase performance in terms of the metrics. This will result in an increase in the accuracy of the gait recognition algorithm. In addition, an initial evaluation of the gait recognition algorithm in the proposed scenario of current user identification was conducted. The evaluation showed the ability to correctly identify the rightful user of the smart phone based on their gait.

The evaluation of the face-to-face enabler was summarized in this document, having been presented in previous deliverables.

The information determined by the enablers is used by the trust and management framework. The Trust Manager provides a service for the SocIoTal framework in order to process information obtained from the enablers in order to provide trust and reputation scores to devices that request them. A multi-dimensional trust model is developed in order to

determine the overall trustworthiness of an IoT device. In addition to traditional concepts such as quality of service and reputation, the trust model also considers security and the more novel concept of social relationship. The four dimensions of trust and reputation are aggregated in order to determine more simple concrete values that are easily understood and that devices can act upon in IoT scenarios where disparate and unknown devices interact each other.

This deliverable also details the implementation of the Trust Manager. This component is implemented as a REST webservice that has the logic in order to build reputation scores. The interaction of the Trust Manager with other developed SocIoTal components is detailed, and APIs and example queries are provided. The method by which the Trust Manager obtains information from the Context Manager, which has the information required for trust and reputation score calculation, is also detailed.

The trials, which were detailed in Deliverable 5.1, will further test the components of SocIoTal which were detailed in this document. The evaluations will be extended from controlled environments with a small number of users to less controlled ones with a larger number of users. In addition, along with KPIs indicating the performance of the enabler, alternative metrics will also be used such as user trust and energy consumption. These can be used in order to assess the uptake of the enabler by users. Once performance has been assessed with the trials, evaluation in the service pilots (O5.3) will occur. In this environment, the components will be evaluated extensively by communities having different profiles.

The next step will be to integrate different trust enablers and to provide a common mean for re-employment of their models by utilizing the Trust Manager framework to fuse different reputation quantifications into one reputation score. As a result, the end users of the platform will be able to build their own quantification mechanism which will consume all of the enablers or only required ones and generate reputation for his application or service. The evaluation of the final version of the Trust Manager will be done at first internally, then during the meetups and workshops, and finally in hackaton and pilots evaluation events. Corrective measures will be adequately planned to correct possible issues such and developed additional functionalities.

References

[1] T. Grandison, “Trust Management for Internet Applications”, PhD thesis, Imperial College London, 2003.
[2] BONATTI, Piero, et al. An integration of reputation-based and policy-based trust management. networks, 2007, 2.14: 10.2
[3] de Kerchove, Cristobald, and Paul Van Dooren. "Iterative filtering for a dynamical reputation system." arXiv preprint
[4] Kamvar, Sepandar D., Mario T. Schlosser, and Hector Garcia-Molina. "The eigentrust algorithm for reputation management in p2p networks." Proceedings of the 12th international conference on World Wide Web. ACM, 2003
[5] Paschke, Adrian, Rehab Alnemr, and Christoph Meinel. "The Rule Responder Distributed Reputation Management System for the Semantic Web." RuleML Challenge. 2010.5
[6] Chen, D., Chang, G., Sun, D., Li, J., Jia, J., Wang, X.:Trm-iot: A trust management model based on fuzzy reputation for internet of things. Computer Science and Information Systems 8(4), 1207{1228 (2011)
[7] Ben Saied, Y., Olivereau, A., Zeghlache, D., Laurent, M.: Trust management system design for the internet of things: A context-aware and multi-service approach.Computers & Security 39, 351{365 (2013)
[8] Nitti, M., Girau, R., Atzori, L.: Trustworthiness management in the social internet of things. IEEE Transactions on Knowledge and Data Engineering 26(5), 1 (2013)
[9] Atzori, L., Iera, A., Morabito, G., Nitti, M.: The social internet of things (siot){when social networks meet the internet of things: Concept, architecture and network characterization. Computer Networks 56(16), (2012)
[10] Bao, F., Chen, I.R.: Dynamic trust management for internet of things applications. In: Proceedings of the 2012 international workshop on Self-aware internet of things, pp. 1{6. ACM (2012)
[11] Bao, F., Chen, I.R., Guo, J.: Scalable, adaptive and survivable trust management for community of interest based internet of things systems. In: Autonomous Decentralized Systems (ISADS), 2013 IEEE Eleventh International Symposium on, pp. 1{7. IEEE (2013)
[12] Chen, D., Chang, G., Sun, D., Jia, J., Wang, X.: Modeling access control for cyber-physical systems using reputation. Computers & Electrical Engineering 38(5), 1088 (2012)
[13] Mahalle, P.N., Thakre, P.A., Prasad, N.R., Prasad, R.: A fuzzy approach to trust based access control in internet of things. In: Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), 2013 3rd International Conference on, pp. 1 IEEE (2013)
[14] Bernal Bernabe Jorge.; Ramos, J.L.H.; Gomez, A.S. TACIoT: multidimensional Trust-aware Access Control system for the Internet of Things. Soft Computing 2015
[15] Hernández-Ramos, J.L.; Jara, A.J.; Marín, L.; Gómez, A.F.S. DCapBAC: embedding authorization logic into smart things through ECC optimizations. International Journal of Computer Mathematics 2014, pp. 1–22.DcapBac paper
[16] E. Hall, *The hidden dimension*. New York, NY: Anchor Books, 1969.
[17] Binzel, Christine and Fehr, Dietmar, Social Relationships and Trust (May 1, 2010). DIW Berlin Discussion Paper No. 1007.
[18] Sibona, Christopher, and Steven Walczak. "Unfriending on Facebook: Friend request and online/offline behavior analysis." *System Sciences (HICSS), 2011 44th Hawaii International Conference on*. IEEE, 2011.
[19] A. Srinivasan, J. Wu, and J. Teitelbaum, “Distributed Reputation-based Secure Localization in Sensor Networks”, Journal of Autonomic and Trusted Computing, 2007.

[20] A. Srinivasan, J. Teitelbaum, and J. Wu, "Drbts: Distributed reputation-based beacon trust system," Proc. DASC'06, pp. 277-283, Oct. 2006.
[21] J. He, J. Xu, X. Zhu, Y. Zhang, T. Zhang, and W. Fu, "Reputation-Based Secure Sensor Localization in Wireless Sensor Networks," The Scientific World Journal, Vol. 2014, 2014
[22] X. Wang, L. Ding, and S. Wang, "Trust Evaluation Sensing for Wireless Sensor Networks," IEEE Trans. on Instrumentation and Measurement, Vol. 60, No. 6, June 2011.
[23] M.S. Bouassida, G. Guette, M. Shawky, and B. Ducourthial, « Sybil nodes detection based on received signal strength variations within vanet », International Journal of Network Security, July 2009
[24] A. Jøsang, R. Ismail, C. Boyd, "A survey of trust and reputation systems for online service provision", Decision Support Systems, vol. 43, pp. 618– 644, 2007
[25] T. Laursen, N.B. Pedersen, J.J. Nielsen, and T.K. Madsen, «Hidden markov model based mobility learning for improving indoor tracking of mobile users», in Proc. of WPNC'12, pages 100–104, Dresden, March 2012
[26] S. Bensaid, D. Slock, «Comparison of various approaches for joint Wiener/Kalman filtering and parameter estimation with application to BASS,» in proc. ASILOMAR 2011, pp.2159-2163, 6-9 Nov. 2011
[27] M. Laaraiedh, N. Amiot, B. Uguen, "Refined characterization of RSSI with practical implications for indoor positioning," Proc. WPNC'13, March 2013
[28] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, M.B. Srivastava, Proc ACM WSW'06 at SenSys '06, October 2006
[29] H. To, C. Shahabi, L. Kazemi, "A Server-Assigned Spatial Crowdsourcing Framework," ACM Transactions on Spatial Algorithms and Systems, ACM 2374-0353/2015/02, 2015
[30] http://www.google.com/mapmaker/
[31] http://wikimapia.org/
[32] Jorge Bernal Bernabe, Jose Luis Hernandez Ramos, Antonio F. Skarmeta Gomez. TACIoT: multidimensional trust-aware access control system for the Internet of Things. Soft Computing journal, 2015.
[33] "Device centric enablers for privacy and trust (Intermediary)", Deliverable D3.1.1 of the SocIoTal project, Aug. 2014.
[34] "Device centric enablers for privacy and trust (Final)", Deliverable D3.1.2 of the SocIoTal project, March. 2015.
[35] V. Saravanan and R. Sindhuja, 'Iris authentication through Gabor filter using DSP processor', in 2013 IEEE Conference on Information Communication Technologies (ICT), 2013, pp. 568–571.
[36] M. Turk, 'Over Twenty Years of Eigenfaces', ACM Trans. Multimedia Comput. Commun. Appl., vol. 9, no. 1s, pp. 45:1–45:5, Oct. 2013.
[37] Tudor-Locke, C., & Bassett Jr, D. R. (2004). How many steps/day are enough?. Sports medicine, 34(1), 1-8.
[38] J. Frank, S. Mannor, and D. Precup, 'Activity and Gait Recognition with Time-Delay Embeddings', in in Proceedings of AAAI, 2010.
[39] S. Sprager and M. B. Juric, 'An efficient HOS-based gait authentication of accelerometer data', IEEE Transactions on Information Forensics and Security, vol. PP, no. 99, pp. 1–1, 2015.
[40] C. Nickel, T. Wirtl, and C. Busch, 'Authentication of Smartphone Users Based on the Way They Walk Using k-NN Algorithm', in 2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012, pp. 16–20.
[41] Y. Zhong and Y. Deng, 'Sensor orientation invariant mobile gait biometrics', in 2014 IEEE International Joint Conference on Biometrics (IJCB), 2014, pp. 1–8.

[42] L. Bianchi, D. Angelini, and F. Lacquaniti, 'Individual characteristics of human walking mechanics', Pflügers Archiv European Journal of Physiology, vol. 436, no. 3, pp. 343–356, 1998.
[43] J. Frank, S. Mannor, J. Pineau, and D. Precup, 'Time Series Analysis Using Geometric Template Matching', IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 3, pp. 740–754, Mar. 2013.44
[44] R. Vautard, P. Yiou, and M. Ghil, 'Singular-spectrum analysis: A toolkit for short, noisy chaotic signals', Physica D: Nonlinear Phenomena, vol. 58, no. 1–4, pp. 95–126, Sep. 1992.46
[45] C. Nickel and C. Busch, 'Classifying accelerometer data via hidden Markov models to authenticate people by the way they walk', IEEE Aerospace and Electronic Systems Magazine, vol. 28, no. 10, pp. 29–35, Oct. 2013.48
[46] Y. Kwon, K. Kang, and C. Bae, 'Unsupervised learning for human activity recognition using smartphone sensors', Expert Systems with Applications, vol. 41, no. 14, pp. 6067–6074, Oct. 2014.
[47] J. Zhu, P. Wu, X. Wang, and J. Zhang, 'SenSec: Mobile security through passive sensing', in 2013 International Conference on Computing, Networking and Communications (ICNC), 2013, pp. 1128–1133.
[48] X. Zhao, T. Feng, L. Xu, and W. Shi, 'Mobile user identity sensing using the motion sensor', 2014, vol. 9075, p. 90750L–90750L–7.
[49] B. Schölkopf, A. Smola, and K.-R. Müller, 'Nonlinear Component Analysis as a Kernel Eigenvalue Problem', Neural Computation, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.
[50] 'Making Your App Location-Aware | Android Developers.' [Online]. Available: https://developer.android.com/training/location/index.html. [Accessed: 24-Aug-2015].51
[51] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi, 'The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication', Pattern Recognition, vol. 47, no. 1, pp. 228–237, Jan. 2014.52
[52] F. Juefei-Xu, C. Bhagavatula, A. Jaech, U. Prasad, and M. Savvides, 'Gait-ID on the move: Pace independent human identification using cell phone accelerometer dynamics', 2012, pp. 8–15.
[53] P. Casale, O. Pujol, and P. Radeva, 'Personalization and user verification in wearable systems using biometric walking patterns', Pers Ubiquit Comput, vol. 16, no. 5, pp. 563–580, Jul. 2011.
[54] S. Albert, 'Granular Computing for Gait Recognition through Singular Spectrum Analysis', in Artificial Intelligence Research and Development: Proceedings of the 16th International Conference of the Catalan Association for Artificial Intelligence, 2013, vol. 256, p. 101.
[55] T. Feng, X. Zhao, and W. Shi, 'Investigating Mobile Device Picking-up motion as a novel biometric modality', in 2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS), 2013, pp. 1–6.
[56] T.-Y. Chang, C.-J. Tsai, and J.-H. Lin, 'A graphical-based password keystroke dynamic authentication system for touch screen handheld mobile devices', Journal of Systems and Software, vol. 85, no. 5, pp. 1157–1165, May 2012.
[57] S. P. Banerjee and D. Woodard, 'Biometric Authentication and Identification Using Keystroke Dynamics: A Survey', Journal of Pattern Recognition Research, vol. 7, no. 1, pp. 116–139, 2012.
[58] R. M. Guest, H. He, S. V. Stevenage, and G. J. Neil, 'An assessment of the human performance of iris identification', in 2013 IEEE International Conference on Technologies for Homeland Security (HST), 2013, pp. 623–626.
[59] SocIoTal D3.1.1 Device centric enablers for privacy and trust
[60] SocIoTal D3.1.2 Device centric enablers for privacy and trust.

[61] H. Hoffmann, 'Kernel PCA for novelty detection', Pattern Recognition, vol. 40, no. 3, pp. 863–874, Mar. 2007.