This4That: Tamper-proof Incentive Scheme for Community Sensing Systems

Diogo Miguel Pardal Calado

Thesis to obtain the Master Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Dr. Miguel Filipe Leitão Pardal

Examination Committee

Chairperson:	Prof. Dr. João António Madeiras Pereira
Supervisor:	Prof. Dr. Miguel Filipe Leitão Pardal
Member of the Committee:	Prof. Dr. Hugo Alexandre Tavares Miranda

November 2017

Acknowledgments

I would like to start by thanking to my parents for all the effort invested in my education and for all the support and incentives given to me in order to conclude with success this stage in my life. To my girlfriend Marisa, thank you for the comprehension in all that days and weekend that was needed to work in this dissertation and I was not always available and especially thank you for your support, you know that is essential.

I would like to thank to my advisor Professor Miguel Pardal, for all the advices given during this master and for all the reviews for the documents like papers, project dissertation and this dissertation document. It was great to work with you professor! I would like to thank too, to my master colleague João Ferreira, with who I had the master meetings, all the ideas and advices that we have exchanged. Finally, to my colleagues, Pedro Joaquim, Tiago Rechau, André Tavares, Andriy Zabolotnyy, Tomás Alves e Miguel Santos for doing this journey with me in the last 5 years. I learned a lot with you guys. Thanks!

Agradecimentos

Queria começar por agradecer aos meus pais por todo o esforço que fizeram para que tivesse uma boa educação e por todo o apoio e incentivos que me deram para concluir com sucesso esta etapa da minha vida. À minha namorada Marisa, obrigado pela compreensão em todos aqueles dias e fins de semana em que era necessário trabalhar na dissertação e que por isso nem sempre estava disponível, e acima de tudo obrigado pelo o teu apoio, sabes que é essencial! Queria agredecer, ao meu orientador Professor Miguel Pardal por todos os conselhos dados durante este mestrado e por todas as revisões em documentos como papers, projeto de mestrado e este documento de dissertação. Foi fantástico trabalhar consigo professor! Agradeço também ao meu colega de mestrado João Ferreira, com o qual tinha as reuniões de tese, todas as ideias e conselhos que fomos trocando. Por fim, queria agredecer aos meus colegas de grupo, Pedro Joaquim, Tiago Rechau, André Tavares, Andriy Zabolotnyy, Tomás Alves e Miguel Santos por realizarem esta caminhada comigo durante estes 5 anos. Aprendi imenso convosco. Obrigado!

Abstract

People are increasingly more connected to the Internet through their personal smartphones. Each smartphone has a wide range of sensors that can capture useful data about the world and the users themselves can be asked short questions about what they are seeing. This so-called crowdsensing has the potential to improve the daily lives of people by sharing useful data about our world. However, there are significant concerns when sharing data including privacy and resource consumption that affect the user motivation. In addition, there is a lack of incentives to motivate the users to continue participating.

In this work we propose This4That, a secure incentive scheme for crowdsensing systems. We implemented a prototype of a crowdsensing system with both the server components (back-end), and the mobile application in the Android platform (front-end). The user can participate in the system and observe the tasks results collected by other users in the community. The prototype includes anonymization mechanisms to protect the privacy of users. For incentives, the prototype uses a 'tit-for-tat' approach where the user contributions are rewarded. The incentive ledger is stored using a variant of blockchain technology that brings two benefits: decentralized resource sharing and record integrity protection. A data quality mechanism is also present to exclude outlier answers to protect the sensing community from self-serving users.

This4That empower users to create communities that can manage and sustain themselves, instead of having to rely on central authority and infra-structure. We believe that this approach fosters many innovative use cases, with a great potential for positive social impacts: Such as the monitoring of waiting queues in hospitals opened to the public.

Keywords: data sharing, crowd-sensing, participatory sensing, blockchain, incentive mechanism, privacy, data quality, data integrity

Resumo

Cada vez mais as pessoas estão ligadas à Internet através dos seus smartphones. Cada smartphone tem uma variedade de sensores que conseguem obter dados úteis sobre o mundo e os seus utilizadores podem responder a pequenas perguntas sobre o que estão a ver. Este método, designado por *crowdsensing*, tem um potencial para melhorar o dia-a-dia das pessoas através da partilha de dados úteis sobre o nosso mundo. Contudo, existem preocupações significativas quando se partilha informação, como aspetos de privacidade e o consumo de recursos que afetam a motivação dos utilizadores. Adicionalmente, existe uma falta de incentivos para motivar os utilizadores a continuarem a participar.

Neste trabalho propomos o This4That, um esquema de incentivos seguro para os sistemas de *crowdsensing*. Implementámos um protótipo de sistema de crowdsensing, com ambos os componentes de servidor (*back-end*) e uma aplicação móvel em Android (*front-end*). O utilizador pode participar no sistema e observar os resultados das tarefas realizadas por outros utilizadores na mesma comunidade. Este protótipo inclui mecanismos de anonimização para proteger a privacidade dos utilizadores. Para os incentivos, o protótipo usa uma abordagem toma-lá-dá-cá (*tit-for-tat*), onde as contribuições dos utilizadores são recompensadas. O registo dos incentivos utilizam uma variante da tecnologia *blockchain* que traz dois benefícios: partilha de recursos descentralizados e proteção da integridade dos registos. O mecanismo da qualidade dos dados está também presente para excluir as respostas de *outliers*, para evitar que haja utilizadores que apenas usem a comunidade para proveito próprio.

This4That capacita os utilizadores para criarem comunidades que se possam gerir e sustentar sem depender de uma infra-estrutura ou entidade centralizada. Nós acreditamos que esta abordagem promove diversos casos de uso inovadores com um impacto social positivo, como por exemplo a monitorização das filas de espera de hospitais abertos ao público.

Palavras-Chave: partilha de dados, crowdsensing, participatory sensing, blockchain, mecanismo de incentivos, privacidade, qualidade dos dados, integridade dos dados

Contents

List of Tables

List of Figures

Acronyms

ALE Application Level Events. 9

AMT Amazon Mechanical Turk. 14, 17, 25, 27

API Application Programming Interface. 6, 26, 31, 32, 41, 51

B2B Business-to-Business. 17

B2C Business-to-Consumer. 17

BLE Bluetooth Low Energy. 9

C2B Consumer-to-Business. 17

C2C Consumer-to-Consumer. 17

CA Certificate Authority. 23

DLL Dynamic-link Library. 33

EPC Electronic Product Code. 9, 10

EPCIS EPC Information System. 9, 10

GPS Global Positioning System. 8, 14, 16, 27, 31

GS1 Global Standard One. 9, 10

HIT Human-Intelligence Task. 14

IoT Internet of Things. 3, 9–12

IP Internet Protocol. 31, 41–43

JSON JavaScript Object Notation. 27, 51

LLRP Low Level Reader Protocol. 9

MAC Media Acess Control. 15

MNs Mobile Nodes. 15

P2P Peer-to-Peer. 19, 20, 22, 23

PDP Partnership-Data-Process. 10

PIS Proactive Information Sharing. 10

PoI Point-of-Interest. 21

PoW Proof-of-Work. 28, 29, 38, 39, 42, 46

RA Registration Authority. 15

RFID Radio Frequency Identification. 9

RPC Remote Procedure Call. 51

RS Report Service. 15

SPC Sensing-Processing-Communication. 14

TIS Template-based Information Sharing. 10

TS Task Service. 15

Chapter 1

Introduction

Some years ago, it was not so easy to get the geographic position or count the number of steps while jogging because equipments to make the necessary measurements were not available or were too costly. In the last decade, the smartphone has appeared and made all these tasks practical with just one electronic device. Also, the smartphone is connected to the Internet and this allows sharing the captured data and making it even more useful. Furthermore, the Internet of Things (IoT) [14] will further connect the physical world to the digital world. This extended network of sensing devices can provide people with more awareness of the state of the world in a digital form and help them make better decisions in their life.

Before data collection and sharing can be done by every user, there are *resource consumption* concerns, as the user may worry that too much battery power and network bandwidth may be consumed in the sensing activities. Also, there are *privacy* concerns [8, 23], as the user may refrain from using the system because sharing information from sensors can expose sensitive aspects about her personal life, like where she is and with whom. To study the importance of both concerns we conducted an end-user survey in a universe of 150 persons, where 35.8% of the respondents had ages between 18 to 25 and 39.1% had ages between 41 to 65. The first relevant result of the survey is that crowdsensing [15] is desired by end-users: 80% of the respondents said that they are available to participate, and 76% even answered that they would participate without rewards. This presents a positive outlook as there are many end-users willing to help others, whenever it is possible.

Regarding *resource consumption*, the users stated that it would have to be in the same level as other popular mobile applications, like social networks.

Regarding *privacy*, 55% of the answers said that constant collection of data is a concern, and the majority of the respondents care about the information collected, like GPS positions, and where this information goes. In Figure 1.1, we can see that people are not very willing to share sensitive data like, credit card, personal data or data from the Smartphone sensors. However, over 60% of people are willing to share social media content.

Overall, more than 90% of the respondents will only adhere to the system if their privacy is assured. These answers provide strong indication that the lack of privacy will have a huge impact in the user participation, so privacy mechanisms must always be present in this type of system.

To help overcome both the resource consumption and privacy concerns, there is a need for

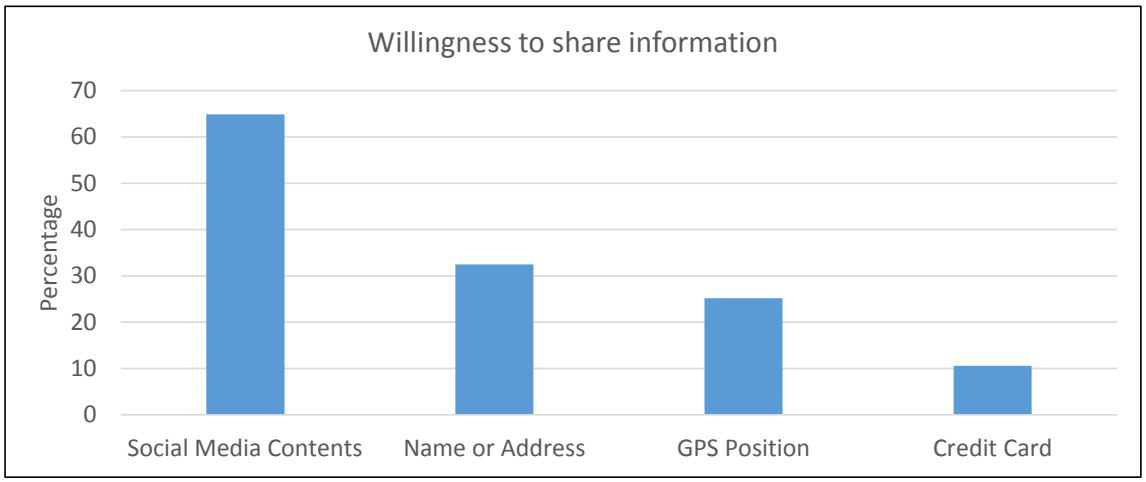

Figure 1.1: Survey answers regarding the willingness to share data.

an *incentive scheme* to motivate users to start using the system, and even more important, to continue using it. Some of the common incentives for data sharing are [2]:

- Direct-exchange: do something and expect something in return;
- Monetary: money in different formats, like coupons, cash or other forms of electronic money;
- Reputation: earned by positive past behavior [10];
- Gamification: points, medals, trophies, or other game rewards [29].

The above list shows that there are alternatives to monetary incentives, and all of them have to be considered, depending on the targeted user communities [22]. In all cases, incentive records need to be trusted by the participants, and the data needs to have good quality [32].

1.1 Goal

In this dissertation we propose This4That, a *secure incentive scheme* for mobile crowdsensing [15] systems where *communities* of people share data among them and create a decentralized and tamper-proof ledger where all the incentive-related information is stored dependably using a Blockchain [24]. Furthermore, we applied some privacy techniques to protect the identity of the user who is reporting the data or is receiving the responses and a quality process to evaluate the data uploaded and give more reward to the users that shared useful information.

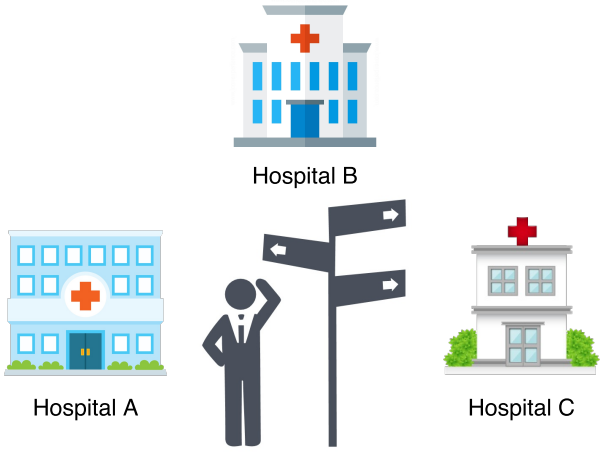

Figure 1.2: Which hospital should the user choose for faster treatment?

1.2 Use Case

The following example demonstrates a situation that can be improved by the use of crowdsensing with This4That.

In the Winter time many people get sick with the flu and usually a trip to medical doctor is needed. The user lives in a city with three hospitals near her house at approximately the same distance. As illustrated in Figure 1.2, *where should the user go for faster treatment?* The user wants to be seen by a doctor as fast as possible, but she has no way of knowing the wait time in each hospital before choosing one and going there. Of course, the hospitals could publish wait time on-line, but this information is sensitive because as the waiting lines are long it can damage the reputation of the institution.

The solution, supported by This4That, is for a community of users to organize itself and share data about the hospital wait times. The user finds a hospital monitoring community that shares information about the hospitals around its home and creates a task to ask to the community: *How long is the line at the hospital?*. Users present at the hospitals receive prompts for checking the state of the line, and answer back. After receiving the results from the community, the user can make a decision and avoid wasting precious time in a hospital that is completely full instead of going to a hospital with less affluence. Overall the general population benefits from a more balanced use of the health services.

Later, when the user is at the hospital, it is her chance to give back to the community: she will be asked about the state of the waiting line. This information will contribute back to the community. This is a 'tit-for-tat' approach – and inspired the name of the system to be This4That – sometimes you help, other times you receive help from others. For the system to work well, all the participations need to be dependably recorded and the people that contribute with good information should be rewarded.

A community sensing system works as represented in Figure 1.3. An user can join a community, starts by creating a task, participates in the tasks created by the other users and wait to be rewarded.

Figure 1.3: Community positive feedback loop.

1.3 Overview

The remainder of the dissertation is organized as follows. In the Background chapter we discuss existing research by presenting other crowdsensing systems and its main features. We also analyze the different business models, what are the incentives used to share information and how the privacy is important. In the Related Work chapter, we talk about different types of incentives that exist and how are they supported by the technology. In the next chapter, the crowdsensing system, defines a basic implementation of a crowdsensing system in order to apply the our secure incentive scheme. The Prototype implementation explains the Application Programming Interface (API) defined to interact in the crowsensing system. The incentive scheme evaluation in shown in Evaluation chapter. Finally, our conclusions and future work prospects are stated in the last chapter.

Chapter 2

Background

This chapter introduces mobile crowd-sensing systems, the main incentives employed to motivate users to participate, privacy-preserving techniques and real use cases that use the crowd to share information and how are they encouraged to participate. We define the user as a person that uses its Internet-connected smartphone to capture and share information.

2.1 Crowdsensing Definitions

There are two approaches to reporting data from our smartphones: *Opportunist Sensing* [13], where the user agrees that her smartphone can be automatically activated on opportunities for data capture and data sharing; and, *Participatory Sensing* [3] where the user has a direct involvement in the data collection activity because the sensing required for the task needs a human observation from the real world. According to Guo [15], the *user participation* can be implicit or explicit by using the opportunistic or participatory sensing, respectively. We argue that both opportunity and user participation are important and should be addressed together. On both cases, the user is consuming smartphone resources.

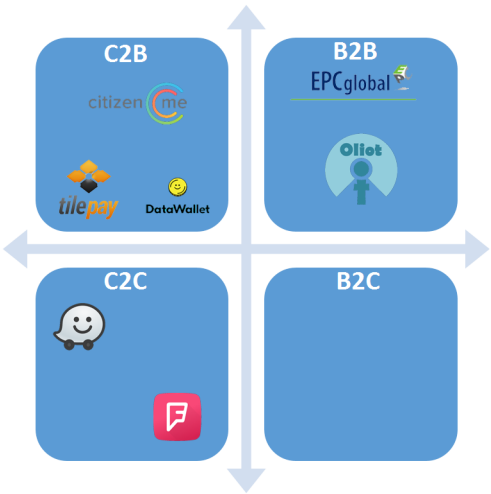

Figure 2.1: Business Models

2.2 Sensing-as-a-service models

In this Section, we will analyze in the real world, the willingness to sharing data with other people, how can the users collaborate with others and what are the constraints that affects the users participation. Therefore we will starting by address the different business models illustrated in the Figure 2.1, by introducing real uses cases, defining who are the stakeholders, how in the different scenarios the users are encouraged to share data and what is the incentives they get.

2.2.1 Consumer to Consumer (C2C)

This business model facilitates the goods or services exchanges directly between users. It can be applied to a community where users will exchange sensor data between each other. One representative solution of this scenario is Waze, the mobile application for sharing traffic information.

Waze is all about contributing to a 'common good', using mobile crowdsensing. This application can connect drivers to each other, creating a community that works to improve everyone's daily driving. The application architecture is represented in Figure 2.2.

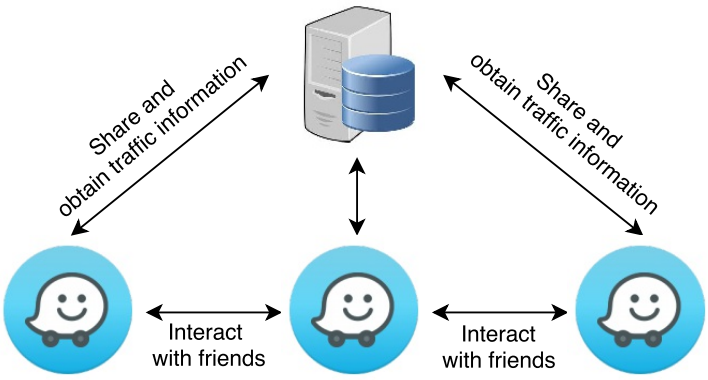

Figure 2.2: Waze Architecture (adapted from [1])

The data sharing is made by using opportunistic sensing [13] to optimize the routes by providing the Global Positioning System (GPS) information and accelerometer. The participatory sensing [13] is used as well, to provide detailed information such as car accidents, road conditions and traffic jams.

But these sensing tasks waste resources from our mobile phones and the users have to be encouraged to keep sharing information. Havlik and Schimak [16] observed some motivations in Waze community:

1. Financial savings due to routes optimization and providing up-to-date on the fuel prices.

2. Situation awareness, due to the event reports by users, a driver can take decisions based on the information observed.

3. Using a reputation schema with 'points' as a virtual currency can motivate users to share events.

[1] https://tools.ietf.org/html/draft-liu-its-scenario-00

Lendák [19] presents a more detailed comparison analysis between applications that use mobile crowd-sensing in order to improve people's daily activities in the urban environment and their motivations. As a result of this analysis, Waze has:

- Mid-range societal impact, by measuring the quantity of active crowd-sensors(users) and the user contribution level.
- Great Economic impact due to traffic optimization, steering drivers more quickly and efficiently to their destination.

To support these benefits, the author identified the most relevant motivation mechanisms for Waze:

- Gamification [29], by using a ranking, where the users earn points by sharing traffic events or by traveling just with the application running on and their game avatar is changing as long as the ranking level is increasing.
- Altruism, to help others to get a fast way to their destination.
- Social networking, to interact with friends and family.

2.2.2 Business-to-Business (B2B)

This business model facilitates goods or services exchange directly between companies. This model can be used in supply chain scenarios where each participant (company) can share product's information with another company. EPCglobal framework [27] has defined standards to identify, capture and share data across the supply chain. They use identification keys to identify each product uniquely and these keys, the Electronic Product Code (EPC) can be captured by reading bar codes or Radio Frequency Identification (RFID) tags. To share the information, a company must implement an EPC Information System (EPCIS) that enables the supply chain participants to share products information about physical movements, status and a overview of the product travel along the supply chain. Each participant has its own EPCIS and publishes the events to the discovery service and this service is responsible to find where is every product.

Project Oliot [4] - Open Language for Internet of Things proposes a standard to extend Global Standard One (GS1)[2] organization code system to IoT data. This project extends the scope of EPCglobal by including not only the barcodes, i.e., in the supply chain use cases, but also collecting IoT sensor data. Oliot has an implementation for EPCIS to share IoT sensor data and a discovery service to get this information. The EPCglobal architecture starts with the Low Level Reader Protocol (LLRP) that reads RFID tags and gives to the next level, the tag information collected. Oliot extends this layer by introducing another Reader Protocols to include another type of connectivity like, ZigBee, 4LoWPAN, Bluetooth Low Energy (BLE) and others which are represented in the Figure 2.3. In the next layer is the Application Level Events (ALE) which provides an API for filtering and grouping RFID data. For the Oliot project, the ALE layer was adapted to include sensor network protocols, as stated above, and to achieve this goal they developed a stream processing feature to filter and collection refined high-level events

[2]GS1 is a global organization which develops the barcode standards

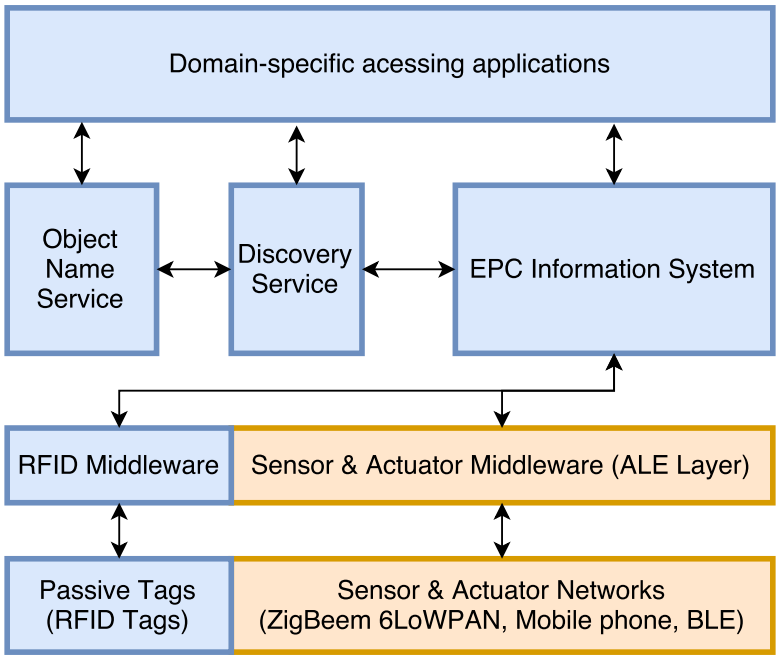

Figure 2.3: Oliot Architecture

that follow the GS1. These events need to be stored in a persistent way and well structured in order to be found in future system queries. Oliot made an adapted EPCIS to store sensed data events and an interface to share these events with other IoT applications. The Oliot project can applied in the HealthCare Service[4], where the patient health status is monitored by an health sensor, the EPCIS stores this information and provide the patient status to other applications.

In this model business-to-business, based on Kürschner et al. [18] research there is a lack of incentives to share products' data in the supply chain. There are no willingness to share the EPC and their EPCIS addresses with other companies. The information sharing between the companies in the supply chain is fundamental, and to support it, the security must be ensured. At the moment, security measures are only access rights and this is not sufficient to enable the participants to use the EPCglobal network because they are interacting with companies they do not know who they are. For these unknown parties they proposed a one-to-one contract, using a pay-per-information pricing model, where an organization share information with others and is rewarded for its sharing.

Du et al. [12], have a different point of view, they propose a Partnership-Data-Process (PDP) perspective to improve the willingness to share information across a supply-chain because the contracts are not enough to reach the success of information sharing, so they included the trust, coordination and interdependence as well. The PDP model focuses on the relationship between partners in the supply-chain which is a major factor to determine the degree of data sharing. In this model, the information was divided into 2 groups, Template-based Information Sharing (TIS) and Proactive Information Sharing (PIS). The first one is information restricted by agreements and the latter is information that when necessary they share with their partners. As the result of their survey, they concluded that only when a supply-chain have a TIS agreement for data sharing they can achieve a high level of sharing by using successful partnership, but in order to achieve a high level of PIS, increasing the partnership is not enough.

2.2.3 Consumer to Business (C2B)

In this business model a consumer produces value and a company consumes the value produced by a consumer. Typically in this model the user is rewarded with money in return for their personal data. Some recent applications like DataWallet, CitizenMe or TilePay follow this model. CitizenMe aims to reward users with money for their data by answering surveys, always with anonymity. DataWallet is C2B data marketplace where the user decides who buys their data and what type of data is shared with an intent of putting an end to the Databrokers. The user provides information of Facebook, Twitter or other social networks and selects which company can access to the data. Whenever a company uses the shared data, the user will be paid.

Specifically in the context of IoT, there is a solution named TilePay , where an IoT owner shares their sensors data by publishing a data profile with the description of sensor data that is available to sell. The company that wants this data profile, pay it using Bitcoins [24] as illustrated in Figure 2.4.

Sharing data from people is not so trivial because the most of the people are not willing to sell their data without having something in return and the assurance that their privacy is preserved.

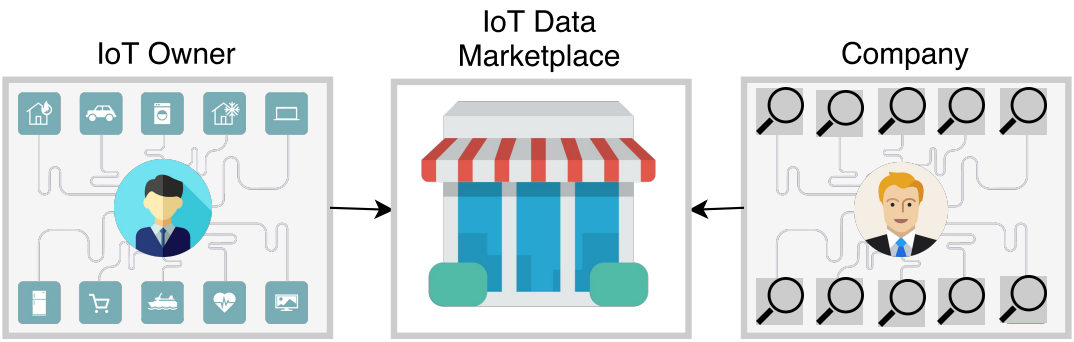

Figure 2.4: TilePay[3]

Benndorf and Hans-theo [1] made a survey about the willingness to sell personal data with money as an incentive. Five of six participants are willing to sell personal data but always with privacy in mind and something in exchange. This study shows there is a general willingness to accept money in exchange of private data. They compared with a non-incentivized survey where they do not receive nothing in return, only a small part, 12% of the people with ages between 18 to 29 years old, are willing to sell their data for commercial uses.

Leon et al. [20] assessed in a non-incentivized environment the factors that affect users willingness to share information with online advertisers. An half of this survey's participants are not willing to share personal information to receiving targeted ads. They found that no one was willing to share sensitive data, like credit card, address, phone number, but they are willing to share less sensitive information like, country, gender, operating system and so on. Data retention policies by online advertisers and the possibility of the collected data be used in other websites (scope of data) are the main factors that affects users willingness to share

[2]Tilepay - http://www.tilepay.org/

personal data with online advertisers.

2.2.4 Business to Consumer (B2C)

In this business model a company produces some product or service and the customer consumes it. The transactions are performed directly between the company and the user. There are no known crowd-sensing uses cases because, perhaps the companies do not want to disclose their information to the others, but there some interesting scenarios that can be formulated in this model. Like a IoT marketplace, where a company publishes some data and sells it for a price. For instance, a hospital publishes the waiting list for a person to be attended. If a certain group of hospital publish this type of data in the same platform may be there are someone interested in buying this information in order to choose the hospital that is more available.

2.3 Crowdsensing Frameworks

Mobile crowd-sensing is a paradigm to collect data from the crowd, different from a participatory sensing system, this type of system uses the the opportunistic sensing and the data from social networks as well. A typical crowd-sensing system has the following work-flow, illustrated in Figure 2.5: one user creates a task and this task is spread to the other users in the crowd by the system. The user receives the task, execute it, and submit the results back to the system, and receive some kind of reward. We will present three different systems, the first is a participatory sensing system and the other two systems are an example of crowdsensing systems that address different concerns.

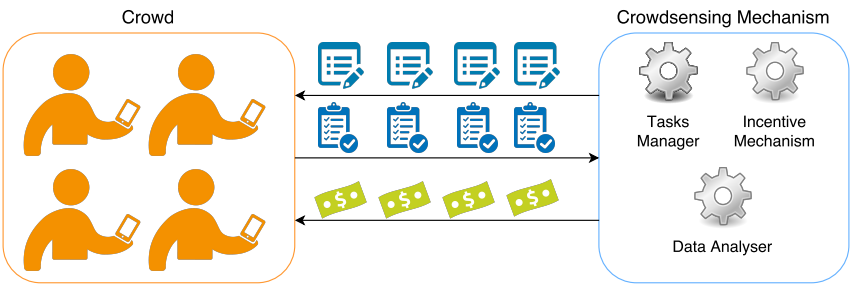

Figure 2.5: Generic crowdsensing Architecture.

2.3.1 Participatory Sensing Framework for Multi-modal Datasets

Wu and Lou framework[30] uses sensor data from mobile phones reported by the users and the information from external applications, like Waze and others crowdsensing applications to enrich the system. This framework is divided in 7 modules as described in figure 2.6:

1. Sensing specifications and use cases: Allows the application developers to specify the data format from the sensors, the type of users, a sampling rate for each sensor and the expected participants that will use the application to forecast the quantity of users in order to adjust the back-end scalability.

2. Crowdsourcing sensing frontend: Provides an user interface for report information.

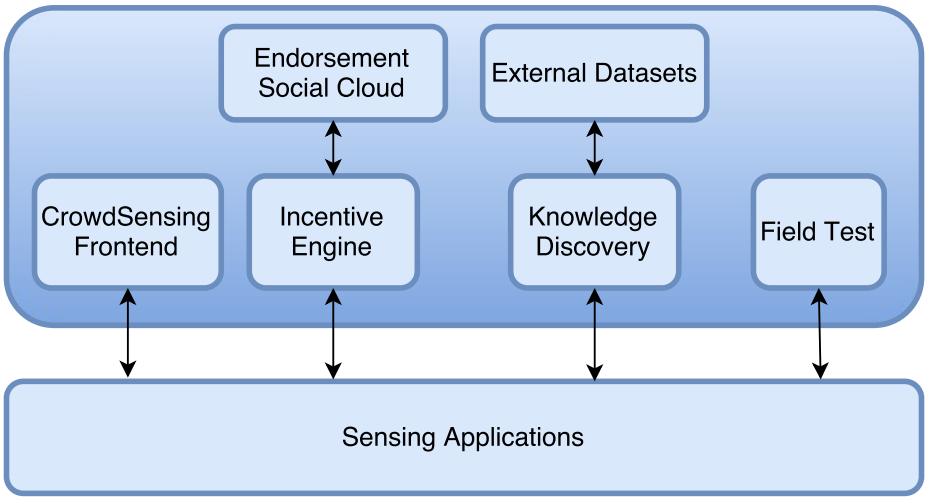

Figure 2.6: Participatory System Architecture

3. External DataSets: This framework was built to read not only from user's information report but also from external data sources in order to use public datasets information regarding traffic information, parking lots availability and other types type of sensed information. This feature will enrich the system functionality by combining the dynamic and static information.

4. Endorsement social cloud: This framework includes an endorsement social network, where a participant indicates that trusts the data contributed by another user, so when an user A shares information, this user can ask to another user B to be endorsed by B and this can accept or reject. This endorsement social network will affect the incentive engine because the user's contribution level will depends on the number of endorsements. It is a kind of reputation system in order to distinct the nodes that provide data with quality.

5. Incentive Engine: The main goal of this engine is to increase the quality and quantity of data rate by rewarding the users with money or other type of non-monetary incentive. In this case the incentive value is calculated based on the user's data quality by taking into account the number of endorsements.

6. Knowledge Discovery: This module provides machine learning techniques to build useful information from the data reported by the users.

7. Field test: This module is an application that integrates the system with the real world to perform pilot studies.

2.3.2 Medusa: A Programming Framework for Crowd-Sensing Applications

Medusa [25] is a programming framework for crowd-sensing applications that provides an high-level abstraction to create tasks. Its goal is to simplify the creation and management of crowd-sensing tasks by implementing a programming language aimed to the people which are not familiar with programming. This language has two main concepts: stages and connectors. The connectors express the work-flow between these steps based on each step's outcome value. The

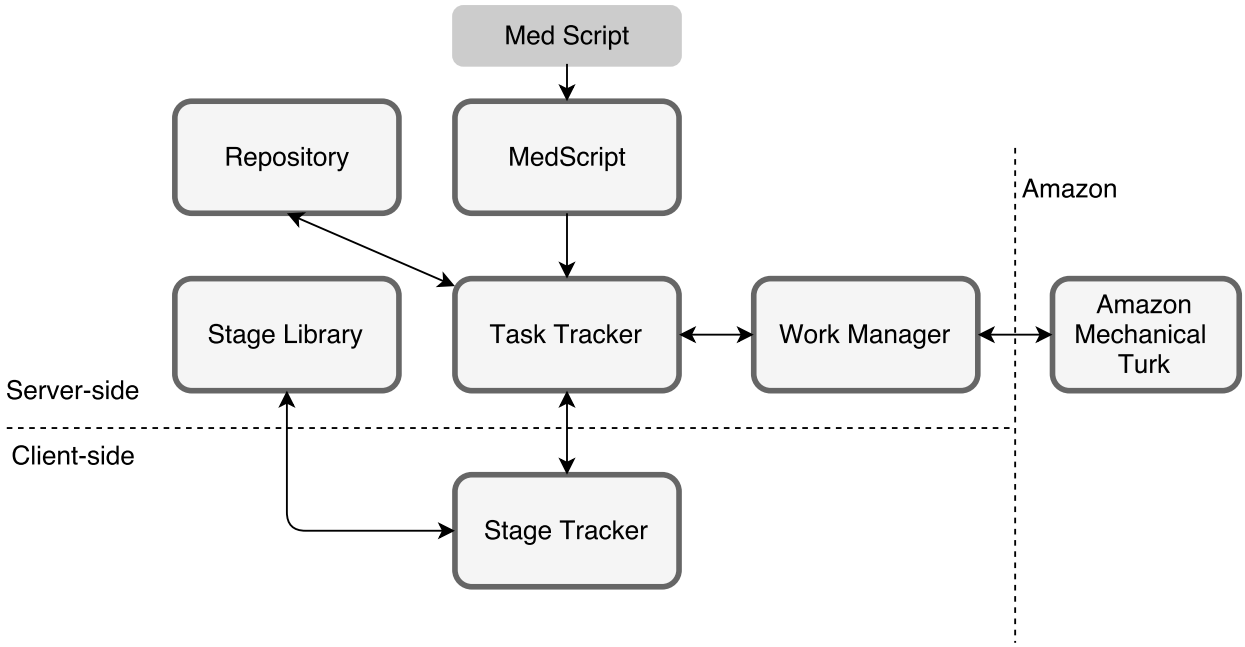

Figure 2.7: Medusa System Architecture

stages are steps belonging to a crowd-sensing task and can be divided in two categories, which are Sensing-Processing-Communication (SPC) or Human-Intelligence Task (HIT). The first are stages that extract data from the sensor and the latter are stages that only require human review. The system architecture is described in the Figure 2.7.

The authors illustrated the system behavior using a video task which consists in making a video of different parts of the world. A user writes the task using the high-level language and submits the program to the *Medscript*. This interpreter parses the program and creates the task on *TaskTracker* and this component will contact the *Work Manager* that uses Amazon Mechanical Turk (AMT) which is a platform that uses crowd-sourcing to perform tasks and reward users with money when they complete a certain task. This platform will recruit people to perform the task by sending a message to the *Stage Tracker* which is running in every phone to receive and to manage the tasks that need to be performed.

After the users accept the task, they will execute a sensing task, which is, in this example, TakeVideo. This task is performed by using the *Stage Library* which provides a set of features like retrieve information from a sensor, detecting faces, get the coordinates from a GPS sensor and so on to facilitate the sensor task, which in this use case, is to access to the mobile phone camera and record a video. Finally the video is sent to *Task Tracker* and the AMT is used again to get another set of users to evaluate the best videos based on the requester requirements and when this step is concluded the requester is notified to get the information that is stored in *Data Repository*.

The most important contributions from Medusa are: the work-flow to create and process the crowd-sensing tasks; the use of AMT payments as incentive to motivate the crowd; and the data quality procedures, resorting again to AMT.

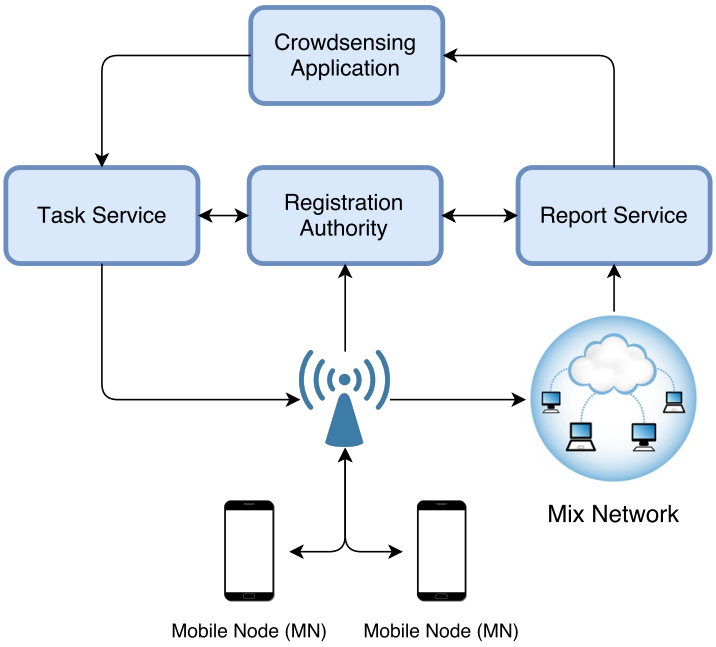

Figure 2.8: Anonysense System Architecture

2.3.3 AnonySense: Privacy-Aware People-Centric Sensing

Ensuring privacy in mobile crowd-sensing tasks is crucial to motivate users to share information. Anonysense [11] is a framework that provides security and privacy in mobile crowd-sensing tasks. Their main goal is to preserve privacy in the entire process, such as in the tasks execution, distribution and in the reports submissions.

The Anonysense's architecture is described in the figure 2.8. This system is composed by Mobile Nodes (MNs) such as mobile phones, one Registration Authority (RA) which is accountable to register nodes who want to participate and issuing certificates to another system components. Task Service (TS) is accountable for receiving tasks from applications and for distributing the tasks to certain MNs that satisfy some preconditions defined in the tasks. They developed an task language different from the one presented in Medusa. This language AnonyTL, is used to specify the task's behavior, a set of acceptance conditions to execute the task, report statements and termination conditions. Report Service (RS) receives the reports from the users and aggregate them to increase privacy and returns these reports to the APP that previously submitted the sensing task. The last component is the Mix Network, a communication channel between the MNs and the RS that provides anonymity to the MNs by unlinking the report to its owner. They propose a rotation of Media Acess Control (MAC) and IP address to decrease the possibility to associate an MAC address to a device. The APP has a certificate issued by RA to communicate with the TS and RS in order assure that is communicating with the right nodes. TS authenticates the MNs as users of the network by using *group signature schema*, without revealing their identity.

The **cryptographic mechanisms** used to support the systems presented above are the following:

- **Group Signatures [7]:** Group signatures are used by members of a group to sign something anonymously. Each group has a pair of keys and has a group manager that is accountable to add, remove users, distribute keys to each user and is the unique person with the capability to reveal the original signer. In that way we the receiver can conclude, the message came from certain group but does not know who really sent it.

- **MIX Communication Channel:** The mix communication channel is used to send messages by hiding the relation between the send and the receiver through a chain of proxy servers. It is useful to protect against traffic analysis. Each message is sent to a proxy server by encrypting the message content with receiver's public key and the proxy server public key. This prevents the proxy server to see the message and increases the privacy without revealing the source machine.

The most important contribution of AnonySense is the use of privacy aspect more concretely the techniques to preserve the users identity when they receive the tasks and when they report the results.

2.4 Privacy in Data Sharing

In this Section we will address the privacy in data sharing, what are the different techniques to protect the users' identity when they are executing sensing tasks and a example of a mobile crowd-sensing framework that provides user anonymity.

In crowd-sensing systems the data collection may affects the users' privacy because there are systems that requires 1glsGPS position or other sensors to collect information and in a privacy perspective this is a problem because some private information, like users' quotidian can be disclosed by tracking their GPS position. There are mechanisms to encourage users to share data without revealing private information. Based on the study made by Christin et al., *Pseudonymity* [8] is a mechanism to protect anonymity and privacy using pseudonyms without expose user's real name. Nevertheless, pseudonymity is not sufficient to protect from user's location disclosure. *Spatial cloaking* [8] can be applied to protect the location privacy. This technique builds groups of k participants where they share a common attribute (*k-anonymity*). For example, the users that live in same country, the location can be replaced by the country name instead of the precise location of each user.

Another way to preserve privacy is *Data aggregation*. This technique is used to distribute data among the neighbors in a network. When an user replies to the querier with the sensed data, this contains the neighbors data and the remaining owner data. It will decrease the probability of matching the sensing data and the user's mobile phone. Shi et al. [26] proposed a distributed scheme called PriSense. Each node in the network chooses randomly n other nodes, to send a piece of sensed data. At the end, each node sends to the aggregation server the total of received pieces of data by the other nodes.

Finally another technique identified is *Data Perturbation* where the data is intentionally perturbed by adding noise. The intention of data perturbation is to still allow determining the community trends without reveal any individual sensed data. The quantity of noise added to data must be chosen carefully because it has to ensure the statistical trend is preserved.

Messaoud et al. [23] proposed a general privacy-preserving model that reduces the relation with private information and conserves the data utility requirements by producing a distorted version of the sensed data. They used this technique of data perturbation to distort the sensed data and an utility function that sets the bounds to the values of any distorted data. This function is used by the querier in order to quantify the collected data. Their challenge was handle the privacy-utility trade-off between the privacy level and the data utility metric defined by the queries.

2.5 Summary

In this chapter we addressed the four different business models: Consumer-to-Consumer (C2C); Business-to-Business (B2B); Consumer-to-Business (C2B); Business-to-Consumer (B2C). This analysis enabled us to know, in real use cases, how is the willingness to share data, how the privacy is so important to enable a good level of data sharing and what are the incentives used and how it can be applied. So, an *incentive scheme* plays an important role in keeping the users motivated and participating in the system and as the participation context is always changing, so different type of incentives must be applied.

After that, we researched about the crowdsensing paradigm by demonstrating different examples of concrete systems like Medusa and Anonysense, and how these systems handle the problems that arise with the crowdsensing like the user privacy, user participation and the data quality. The Medusa showed the work-flow to create and process the crowdsensing tasks by developing a crowdsensing platform integrated with the AMT to manage the incentives and the data quality procedures. From Anonysense, to complement the Medusa system they focused on the privacy aspect more concretely the techniques to preserve the users identity when they receive the tasks and when they report the results.

We also searched for different types of privacy techniques to preserve the user's identity like the usage of pseudonyms, spatial cloaking, data aggregation and data perturbation.

Chapter 3

Related Work

This chapter introduces different *incentive models*, to motivate users to participate and share information in crowdsensing systems. For each type of incentive we also researched about the security mechanisms underlying the systems and how the authors implemented the incentive schemes.

3.1 Incentives for Data Sharing

Sharing information requires resources, like battery, computing, bandwidth and incentive mechanisms must be applied in order to keep the users motivated to share information. There are many incentives for data sharing like direct-exchange, money, coupons and gamification and according to Bhattacherjee [2], the last three incentives are the main ones.

3.1.1 Direct Exchange-as-an-Incentive

The direct exchange method is based on the 'tit-for-tat' approach, meaning when we do something we expect to receive an equivalent action in return. This context can be applied to data sharing like in Peer-to-Peer (P2P) networks [1] that allow users to use the network nodes resources and is expected that each user donate their resources for other users in the network, but not all the users are willing to share their resources. This problem is known as *free-riding* problem and the 'tit-for-tat' approach can solve it by punishing the greedy users.

BitTorrent [9] is a file sharing system in a P2P network that uses *'tit-for-tat'* method to choose the peers to upload the pieces of file. In order to complete the all pieces belonging to a file, as soon as possible, each peer is accountable to maximize its own download rate and to get it, is applied a tit-for-tat approach. When a peer wants a piece of data from another peer, the first has to upload data to the latter in order to receive the desired piece of data.

3.1.2 Electronic Money-as-an-Incentive

The most common incentive to reward users is money or electronic money. The money can be represented in different ways, it can be as a coupon that has monetary value, simply a monetary

[1] Peer-to-Peer network is a distributed architecture composed by nodes which are named peers connected to each other and main goal of this architecture is to distribute the load among the peers without having a centralized overloading node.

value or even a digital currency like Bitcoin.

The use of electronic coupons (e-coupons) is becoming popular in electronic business because it can helps people to save money. There are some important concerns regarding e-coupons, like security and privacy that must be taken into account.

Chang et al. [5] propose an e-coupon system for the mobile commerce, applying simple cryptography techniques such as one-way hash functions and symmetric encryption without performing complex computations and preventing the *double-spending* problem which consists in spending the same coupon twice.

As we are in a digital era, digital currencies have appeared and its use has growing constantly in the last years [2].

Yang and Hector [31] developed the PPay (PeerPay) system which is a pioneering P2P micro-payments systems. The main functionality of this system is transferring coins between peers without the validation of a centralized trusted authority, like a bank for each transaction. This scheme prevents forgery using digital signatures and detects the double spending problem. An user X starts by requesting a certain number of coins to central trusted authority, T. Each coin is signed by T to ensure authenticity. When the user X makes a transaction to another user Y, X transfers the coin and a sequence number identifying this assignment. The user Y is the new holder of the coin and when he wishes to spend this coin, by transferring the coin to another person, he sends a reassignment to the coins' owner in order to renew this coin by generating a new sequence number that must be grater than the previous. At the redemption phase, where the coins are 'cashed' the trust authority T will check if there are equals multiple coins with the same owner and serial numbers, which indicates that double-spending has occurred.

Actually the most virtual currency used is Bitcoin [24], is a decentralized P2P payment system. As in previous system, a peer can perform transactions to another peer. Each transaction contains a cryptographic hash of the previous transaction and the public key of the transaction's receiver, which is the 'destination address' to send the Bitcoins. As stated before, the main goal of Bitcoin system is not depending on a trusted centralized entity like a bank. Every intention of performing a transaction must be announced to all peers in the network in order to start the validation process. This validation process is known as 'mining' where every 10 minutes some peers will validate the announced transactions and add the valid ones to a distributed ledger, named 'blockchain'. The transactions must have another transactions as an input to justify where the money came and if these transactions had already been spent, the double-spending problem is detected right in the moment. Regarding security issues, the anonymity is ensured by using the public keys as an 'pseudonym' for each peer, the authenticity is provided by signing the transaction with the sender's private key. To sum up a comparison between the classic (PPay) and the state-of-art of digital currencies are described in Table 3.1.

[2]Bitcoin Usage Statistics - https://blockchain.info/pt/charts/total-bitcoins

Table 3.1: Comparing PPay and Bitcoin

Criteria / System	PPay	Bitcoin
Non-repudiation	Yes	Yes
Anonymity	No	Yes
Decentralized	Yes	Yes
Double-Spending	Detects	Prevents

The **cryptographic mechanisms** used to support the systems presented above are the following:

- **Hash-function:** One-way hash functions produces always the same output given an input. The great advantage of hash functions is the easiness to compute the output and the impossibility to find an input value with the same output value. This cryptography function is important to build a block-chain because in the chain the next block contains a hash value of the previous block that is explained below. An application of hash functions are *digital signatures* which allow to ensure a document integrity and detects the document forgery or tempering.

- **Symmetric Cryptography:** The symmetric cryptography is composed by a plaintext message, an encryption algorithm that performs substitutions and transformations on the plaintext and a secret-key which is used an input of encryption to cipher the plaintext.

- **Asymmetric Cryptography:** The public-key encryption or asymmetric encryption that uses two different keys, public key and private key, to ensure confidentiality, key distribution and authentication. This algorithm is based on mathematical operations rather than bit operations like in symmetric encryption.

3.1.3 Gamification-as-an-Incentive

Gamification [29] is a technique to motivate users to execute tasks in a non-gaming environment to collect data using game mechanisms. The common schemes are points, badges or ranking. According to the author research [2], this mechanism increases the quality of data by increasing the quantity of data and became a good incentive in crowd-sensing applications. Foursquare[3] and Waze [4] are an practical example of this system where they use gamification to increase the quantity of sensed data, where in Foursquare the user earn badges by sharing information and in Waze the user's earn points by sharing traffic information and its avatar is changing according to the ranking and the users stay motivated to reach the next level. Ueyama et al. [29] proposed a new incentive mechanism using gamification for participatory sensing [13]. They design a system where a user receives a request to perform a sensing task at a specified location, designated as Point-of-Interest (PoI) and receives some reward (points) after complete the sensing task. The different locations (PoI) have a different quantity of reward points and the main goal is to cover the maximum area of PoI. To minimize this problem they applied a gamification approach, using

[3]Foursquare - https://www.foursquare.com
[4]Waze - https://www.waze.com

a ranking and badge scheme in order to maximize the sensing coverage area by encouraging the users to participate in all sensing tasks.

They evaluated their system by introducing two types of requests. The first, are requests with monetary and gamification as incentives, designated as SP request. The second, are requests with only monetary incentive, designated as normal request. In the results, they observed the participation rate of SP requests was higher than the normal request. So they concluded that the gamification motivated users to participate in the tasks with lower points by increasing the participation rate from 53% (without gamification) to 73% (with gamification).

3.1.4 Reputation-as-an-Incentive

Reputation systems [10] are used to incentive users to interact with strangers in a online context. Like in online commerce, Ebay, Amazon and others online shops where they apply a reputation system to provide the customers an opportunity to evaluate the sellers quality and to promote the honesty between the sellers because when they are selling a product, the reputation system will influence the chance to get more clients based on the clients' feedback.

In P2P networks where nodes share resources with other nodes, there is a critical problem in choosing the nodes that deliver a good quality of service. The common approach to minimize this problem is building a system based on *reputation systems*. However there are problems that should be taken into account as stated by Chang et al. [6] where they identified under-participation, where a peer does not care about the reputation system, and malicious peers that submit false recommendations about others.

The authors suggest a solution to this problem, starting by assessing a peer trustworthiness but is not enough to looking for their reputation because it might be influenced. The peer has to find a set of *witnesses*, which are peers that make recommendations about another peer and construct a chain of referrals, based on their credibility calculation model. After that, the peer make a query only to these referrals to obtain the recommendations. Nevertheless, a witness may not reveal the right peers and to avoid false information about other peers, they implemented the mentioned credibility model. This model analyses the witness's recommendations by their history and assesses their performance. This reputation system adopted a *'tit-for-tat'* strategy to maintain the honest peers participation. When a peer i receives a request from another peer j asking for a recommendation about a peer s two scenarios can happen: If the peer j is not participative and is dishonest, the peer i ignores the request from peer j; If peer j is participating and honest, the peer i sends the recommendation regarding the peer s to peer j. It does not prevent bad recommendations but creates a distinction between peers.

These reputation systems can be applied in centralized or distributed way. Liau et al. [21] proposed a distributed reputation system where the reputation information is stored by the each owner. When an user needs a service from some peer make a request to the peer and this peer reply with all the Reputation Certificates (RCert). These certificates contains the reputation made by peers that evaluated this peer, each certificate is digitally signed by the peer that consumed the service (consumer). After the requester evaluate all RCert, if he decides to consume the service, he sends an acknowledgement to the service provider and begins to consume the service. After that the consumer updates the RCert collection by adding their reputation for

the service.

The cryptographic mechanism used to support this system presented above is the **Digital Certificates.** The public key certificate consists in the certificate owner public key, owner ID and a block signed by a trusted authority. The common third party which signs this block is a Certificate Authority (CA)) which is trusted by the entire community. This technique is used to prove the public key ownership because the CA verified the content of the digital certificate is correct and allow us to realize that we are talking with the right person.

3.2 Distributed incentive ledger using Blockchain

The blockchain technology is new way to store and manage data in a distributed way. Nowadays the databases are the standard way to store information but the problem comes when we have to trust in one entity to manage our data. The blockchain appears as an alternative to databases without the need to trust in a centralized authority to manage our data. This technology comes up in the Bitcoin system using a distributed approach (P2P networks), where each participant is unknown and all the participants have to see the transactions in the same way. In order to accomplish this, they implemented a distributed timestamp server that contains a hash of the previous and actual block of transactions forming a blockchain. Each block contains a set of transactions as illustrated in figure 3.1.

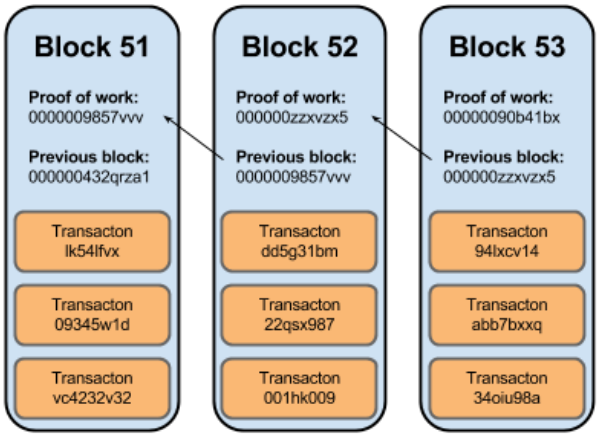

Figure 3.1: Blockchain example (image taken from [5]).

When an user needs to broadcast the intention of performing an transaction to all nodes. The 'miners' (users that are validating blocks) in order to create blocks of transactions have to find a proof-of-work by executing an heavy computational work and after that, they announce the block to the network in order to other nodes can also validate the transactions inside the block. If the transactions are valid, the block will be added to the block-chain forming a proof of the chronological transactions order.

The blockchain are divided into two types: open and private. The Bitcoin system uses a **open blockchain** where there is no control who enters in or out and who they are and every

[5] http://dataconomy.com/2015/10/wtf-is-the-blockchain-a-guide-for-total-beginners/

user can participate in the blockchain.

The private blockchains or **permissioned blockchain** the users who want to participate in the blockchain have to ask for permission and its identity is known at the beginning. This type of blockchain is faster than the open ones because as the entities are known the blockchain owner, there is no need of proof-f-work as happens in the Bitcoin system.

Nowadays block-chain is not only present in Bitcoin system, this technology is being applied in the business context and there are some open source implementations like the HyperLedger Project and MultiChain as a permissioned blockchain that is used in our solution architecture.

The blockchain solution offers a great level of fault tolerance against a centralized database because is built on top of redundancy where every node of the network contains the same transactions that are automatically propagated between them. Besides that, the availability in blockchain is greater than in centralized versions because if the node that is receiving the transactions fails and loses the entire history of transactions made, he can continue receiving transactions when reconnected to the network without have the transaction history consistent. The user will ask to the other users for the previous blocks to start rebuilding the blockchain. In Multichain, as they use the mining diversity, a certain number of nodes must be online in order to build a consistent and valid blockchain, so the mining diversity must be adjusted based on the active users without freezing up the blockchain.

In a centralized system they have a database running in one machine and they use the disaster recovery technique to recover by backups when something goes wrong. In fact the centralized solution offers a great performance level in the transactions registration because it only has to be verified one time and just by one entity, but on the other hand it does not offer a great level of fault tolerance because if the system fails, they have to recover by backups and wait until the system is in a consistent way to receive more transactions.

3.3 Summary

Throughout this section we started by describing the different type of incentives to encourage people to share information that we found in the literature and concrete systems that they are applied. The most relevant and interesting ones are the electronic money, specifically the Bitcoin system that is supported by the blockchain technology and the gamification showed to be a good incentive scheme to use in non-gaming environments.

The blockchain technology showed to be a good decentralized approach to manage information that is protected against tampering. In that way the need of a trusted authority disappears and as the information are replicated along the users and these are responsible to maintain the blockchain, is more difficult to corrupt the data without anyone notice that.

Chapter 4

Crowdsensing System

This4That is a secure incentive scheme for crowdsensing systems. It is based on the crowdsensing systems that are presented in the Background chapter, Medusa and Anonysense, the first comprehends the incentive scheme without any type of privacy and a data quality process to evaluate data and the latter focused on the privacy aspects and does not have an incentive scheme. Our goal was to implement a secure incentive scheme using the previous modules but as the implementation of Medusa is so dependent on the AMT logic and the unavailability of the Anonysense implementation, we had to implement a basic crowdsensing system in order to integrate our incentive scheme.

So, This4That addresses the incentive management, the privacy protection and quality of the data collected, taking some ideas from the existing systems but adding new ones.

The first aspect of our solution, presented in Section 4.1, is the community model which shapes how the users will interact. For the implementation of a crowdsensing system, the task management components, its architecture and the quality process are inspired by Medusa and are described in Section 4.2. The data quality approach, described in Section 4.3.2, uses a modified Z-Score statistic method to reward users in a fair way. As we do not know with high level of confidence what are the real response, we used a statistical method to discover the trending answer and to discover the potential outliers because many times people just want to perform tasks to receive the rewards.

Finally in Section 4.3, we describe our secure incentive scheme to motivate users to keep contributing with information. The incentive scheme is the very important element of crowdsensing systems because the users have costs by using the smartphone to share information and the incentives keep them motivated to continue participating. If the users do not feel rewarded for their actions, they will eventually stop providing information [1]. This incentive scheme removes some participating barriers as the privacy which is described in section 4.3.3 based on the usage of *pseudonyms* and the usage of *blockchain*.

4.1 Community Model

We define *community* as a group of people that have a shared goal and that join together to share information related with the goal. The information can be an answer to a question posed by another member of the community or data collected from the sensors of the Smartphone. We

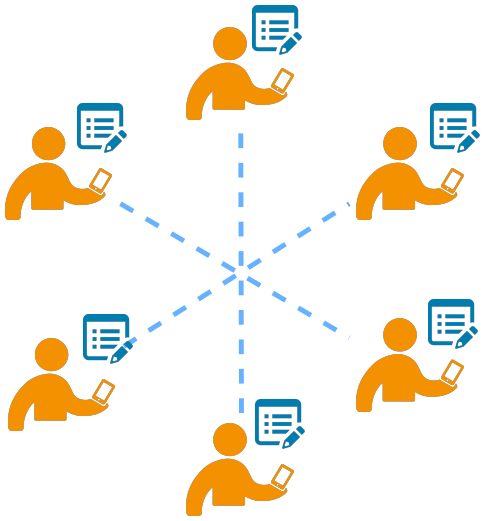

Figure 4.1: Community Model. Each user receives useful information from the community and gives back to it by contributing with sensor readings and answers.

represent this community dynamic in Figure 4.1: sometimes users give to the community, other times they receive from the community.

The vision for these communities is that they appear from the users' needs, are mostly self-organized and do not have "official" support from governmental or commercial entities that at any moment can be shutdown or simply they do not exist to service their needs. For instance, following the wait time in hospital example, in a centralized perspective the users must register in the hospital platform, if it exists and they need to provide a lot of personal information to get the registration complete and in these platforms the data shared with users is totally controlled by the platform owners.

So with this decentralized approach based on the a community, the information is controlled by the users and they are not depending on other platforms. To make this possible, each user must contribute to the sustainability of the community by providing some resources, namely, computational resources that support the incentive ledger. This contribution will be detailed in Section 4.3.

4.2 Architecture

For the architecture of a crowdsensing system we propose a system architecture with six main modules, represented in Figure 4.2. The *Task Creator*, the *Incentive Engine*, and the *Repository* modules are inspired in Medusa to allow the creation of crowd-sensing tasks. The *Incentive Engine* is needed to keep track of user actions and their respective rewards. The *Task Distributor* and the *Report Aggregator* modules are inspired in AnonySense and apply privacy techniques while distributing tasks and collecting reports from users like encryption and decryption. The *API* is an entry-point responsible for receiving the requests from the users and routing them to the destination.

The *Task Creator* is the node that receives the tasks specification and creates this entity in the system and applies rules, if necessary. It accepts two types of tasks:

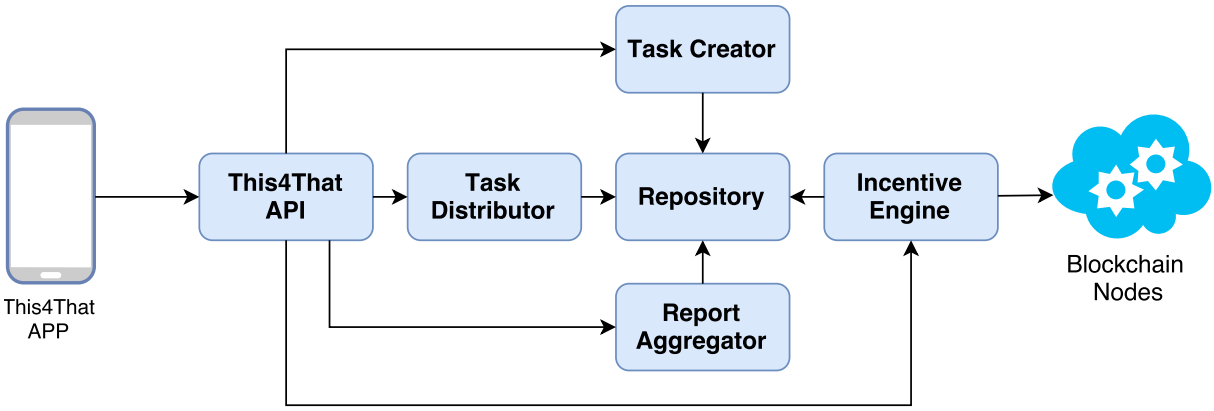

Figure 4.2: Solution Architecture

- Sensing Task - specifies a sensor to be used at a given GPS position (opportunistic sensing);
- Interactive Task - specifies a question and a set of possible answers (participatory sensing).

The task is specified in a JavaScript Object Notation (JSON) format[1]. The task record contains: a name, a topic that refers a set of tasks for a given subject, an expiration date, and, if it is a sensing task, the sensor to be used; or if it is an interactive task, the question and the set of possible answers. The users can subscribe tasks by topic name which is specified along with the task specification and more tasks can be created in the same topic. This provides to the users a way to search tasks. The *Report Aggregator* module collects the reports and can apply rules regarding the reports. The *Repository* will store all the entities like the users, tasks and reports.

4.3 Incentive Scheme

The incentive scheme is the core of our system, because this engine is responsible to make people sharing information by giving incentives to them. As we want a system supported by the community, we want to empower the users to manage themselves by enabling them to manage the incentive transactions. This is in contrast with Medusa [25] that uses AMT to reward the users with money, where all the power of decision to transfer the incentives relies on the provider (Amazon, in this case).

In our approach we developed an incentive scheme that allows different types of incentives, so the implementation is independent on the context. So, incentive schemes like in the Waze application, where the users receive points by sharing information about accidents or traffic jams can be used in our solution.

As this is a community environment, we do not want a central authority or other external entity to check if the incentives transactions are true or if one user is trying to cheat the system, by changing the incentives transactions. This line of thought led us to find an incentive scheme that must be controlled and supported by the community itself without third parties interfering on the process.

[1] JavaScript Object Notation (JSON) defined in `https://tools.ietf.org/html/rfc7159` provides a common syntax for text-based data formats.

Using a distributed scheme to registering the incentives transactions, we discovered the *blockchain* approach is the most suitable technology [24] in terms of a distributed ledger and privacy preserving. It is a public ledger containing transactions distributed for all the participants where the load of the validation process is supported by the users as described in the related work chapter.

Initially our idea was to use a mobile blockchain system because the mobile crowdsensing as the name indicates, the main object to collect and share information is the smartphone, but after making a research on this subject we noticed that, there is no mobile blockchain system implemented at the moment. The resource consumption is the principal constraint due to the execution of the Proof-of-Work (PoW), so we decided to integrate our incentive scheme with the **Multichain** solution which is a variant of the blockchain and needs a computer to running.

4.3.1 Incentive Ledger

This4That was integrated with a permissioned blockchain, called Multichain[2], which allows to build a blockchain and allows to transfer not only money but also *assets*. The assets is something to store in the blockchain that has a name and quantity and it can be used to implement a gamification incentive scheme where the reward is not money, but can be points or trophies. This type of blockchains aims to solve problems like the privacy and the monopoly of mining process to ensure that everyone participates in the mining process and can contribute to the blockchain.

The privacy is ensured by using pseudonyms like in the Bitcoin system, they used the public key cryptography where the public key is the user address and private key is used to sign contents. Besides that, to improve the level of privacy they developed a private blockchain, where blockchain content are only visible by the its users.

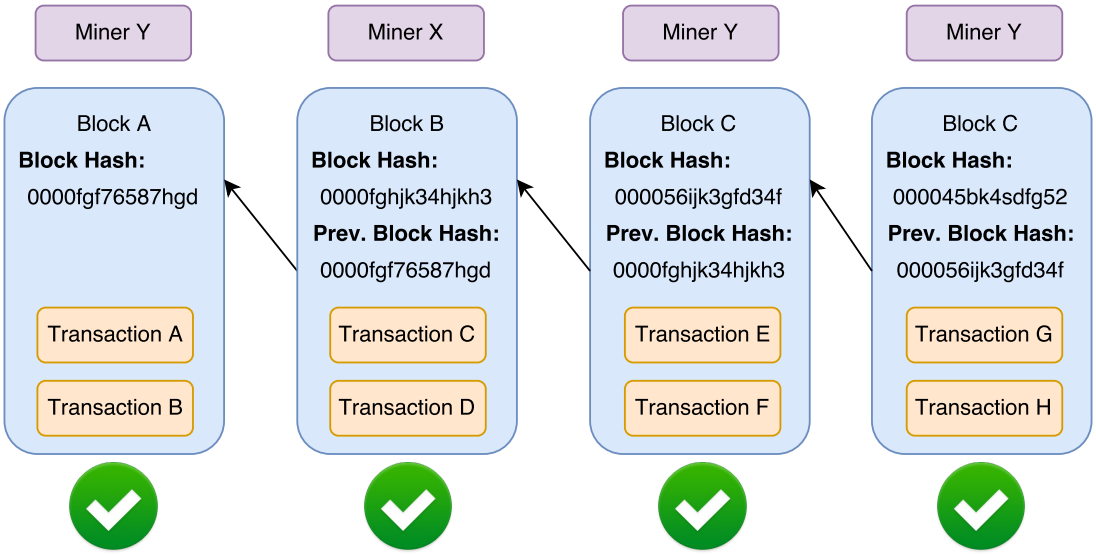

Figure 4.3: Blockchain from Bitcoin system.

In the bitcoin system, as this is an open blockchain and there is no control on its users, the mining process is a race between all participants, the user with more power is the one that

[2]Multichain - `http://www.multichain.com/` - is an open platform to build Blockchains.

generates the next block. Looking at the figure 4.3 the Miner Y is controlling the mining process because he has more power than the orders nodes and the blocks are valid and accepted by the other users.

The validation process of the transaction in Multichain is quietly different, they still use the concept of miners as Bitcoin, which are nodes in the network that provide their resources to validate the transactions and receive something in return by its work, but in Multichain is not the more powerful node who validates the block. There's also the PoW, but as it is a permissioned blockchain, the difficulty level of PoW is low and they used another way to choose who will validate the next block.

$$interval = miners \times mining\_diversity \tag{4.1}$$

In the Multichain, they applied the concept of *mining diversity*, which means that the same miner can only generate a block within a given window. For instance, using the equation 4.1 to determine the interval that an user has to wait to validate the next block. If the mining diversity is 0,5 and the number of miners are 6, the interval to be able to validate the next block is 3. This enforces a rule to create blocks using a rotation between the users to avoid the monopoly of mining blocks and to improve the overall fairness of the blockchain. As we can observe in figure 4.4, in the multichain system, the fourth block is invalid because the Miner Z has mined the previous block and as the interval is 3, the Miner Z has to wait for three blocks to get a chance to mine another one.

In our prototype, we decided in order to preserve the users' identity, if an user wants to join a community he cans, without being asked for permissions to access this blockchain and this will turn the Multichain in a open blockchain. So, every one that enters in a community is a miner and the mining diversity might be not enough to protect the incentive ledger as will be discussed in the evaluation chapter.

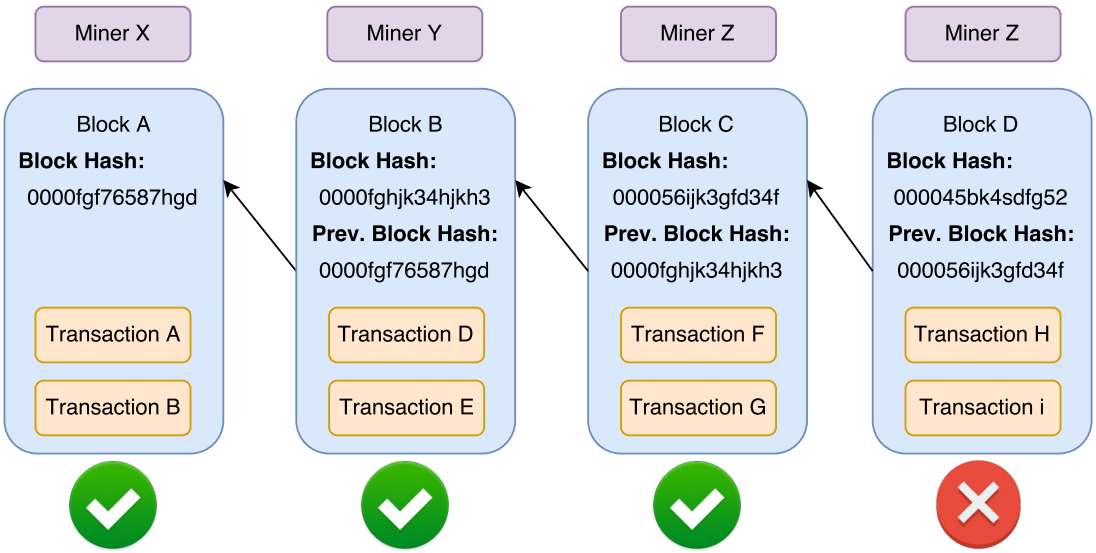

Figure 4.4: Blockchain from Multichain system.

Regarding our system implementation, when the users register themselves in the community they are asked to create a blockchain node in their computer and to create a connection to the

community Multichain node. At this step, an multichain address will be provided, that works as the user identification and the token to establish a connection between the user multichain node and the This4That Multichain Master node. This step will authorize the node to participate in the Multichain network and that way they will contribute with their resources to sustain incentive validation process. Nevertheless, an user can even contribute more to support this process by adding more machines as miners and this contribution is reflected in the reward value.

4.3.2 Data Quality

The quality of data [32] is an important aspect of crowd-sourcing solution because sometimes when there are incentives to be distributed, some users just want to execute the task to receive the incentive and do not care about the quality of the reported information. Submitting wrong information will introduce noise in the data samples and make it difficult to understand among the collective users what is the right information. To mitigate this problem, we used a statistical method to distinguish the *outliers* from the people who really reported the correct information. In our example we do not want use the mean as parameter to calculate the Z-Score of each observation because the outliers will influence a lot this value.

So we used the *modified Z-score* which is more robust and uses the median as parameter to avoid the value oscillations as occur in the normal Z-score. Iglewicz and Hoaglin [17] proposed the *Z-Score modified* to identify outliers. This method is different from the well-known method *Z-Score* which tells how many standard deviations an observation it is from the mean. This calculation will allow the system to infer how much an answer is away from the typical answer.

$$MAD = median(|X_i - median(X)|) \tag{4.2}$$

$$Z_i = \frac{0.6745(X_i - median(X))}{MAD} \tag{4.3}$$

Equation 4.2 refers to the Median Absolute Deviation (MAD) which calculates the central value for each observation deviation.

Equation 4.3 calculates the modified Z-score for each i-value, which will allow us to understand how much this value is deviated from the tendency.

$$k = \left| \frac{Z_i}{Max(Z_i)} \right| \tag{4.4}$$

$$FScore_i = TaskScore - k \times \frac{TaskScore}{2} \ , k \in [0, 1] \tag{4.5}$$

Equations 4.4 and 4.5 are designed specifically for the task reward calculation. Equation 4.4 is the impact that the Z-score answer has compared with the other Z-scores using the median

of i response (Zi) and the maximum of i responses (Max(Zi)), which is a value that goes from 0 to 1. Finally, Equation 4.5 calculates the final score to assign to each answer. This method is expected to penalize the outlier users to encourage them to share more accurate information in the future. We opted to not give an empty reward to the wrong answers because we do not want to discourage the participation. In the evaluation chapter, we will evaluate this method in a concrete scenario and it will be compared with other statistical methods.

4.3.3 Privacy Protection

Providing sensor data without any protection can expose information about the daily lives of people. For instance sharing data from the GPS sensor can reveal the position of person throughout the day which allows an attacker to discover the habits of a person. Also, answering an interactive task about a physical space may indicate who is the user that is in the place. So we adopted a *pseudonyms* mechanism [8] that protects anonymity and privacy by replacing the real names and number of users with different values. In practice when an user registers in the system a Multichain address will be assigned to him and this address works as user identifier and wallet address to receive the incentives. Like in Bitcoin, each user has a key pair, where the private key is only known by the user and is used to sign transactions and the public key with some transformations is the address to transfer the incentives.

But in fact the pseudonyms are not enough to avoid a correlation between the data and the user that reported this data [8] because the Internet Protocol (IP) address that came along with the request is not masked and can reveal the origin. In Anonysense the authors adopted the Mix network to cover the origin IP address. In our approach we also applied a simple Mix network,by using an external service that provides a random proxy to every request when needed. In this way the IP that reaches the platform is always changing even if the user is the same and the smartphone that reported the information cannot be linked to that information.

Nevertheless, as standard techniques of encryption like the use of asymmetric keys will be used and the usage of group signatures like in AnonySense appears to be a good way to authenticate a user in the community without really knowing who is the user that is using the key because the main goal of this technique is to authenticate someone without exposing their identity. This last technique, Group signature - are not implemented in the current prototype and are defined as a future work.

4.4 Prototype Implementation

In order to develop our incentive scheme, we had to develop a basic crowdsensing system with a client and server side that can allows to validate the design of our solution architecture and to enable us to evaluate our incentive scheme.

In order to interact with our platform was necessary to develop an client-server architecture. When a community is created, an instance of a crowdsensing system with the This4That secure incentive scheme is instantiated and the Multichain Master Node is created as well. We deployed the server side which contains the modules presented in Section 4.2 in a web-server that will be part of the platform infra-structure. The web-server contains an REST API, described in

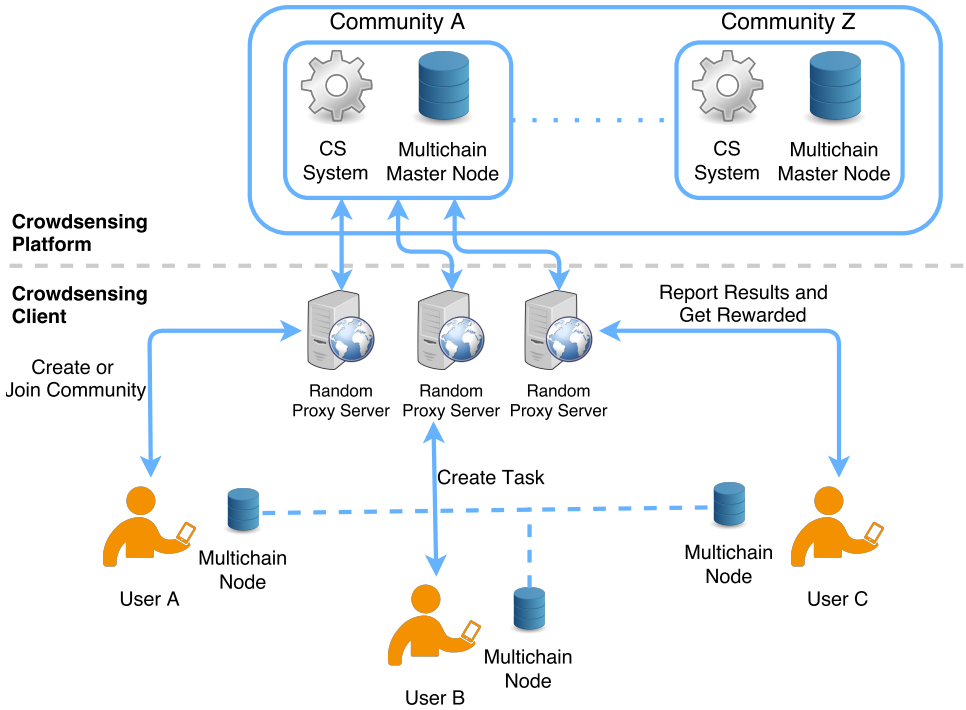

Figure 4.5: Crowdsensing platform running This4That incentive scheme.

the Appendix, to be invoked from the client application in order to communicate and forward each request to the respective module. The client application was developed in Android, but as the back-end API is cross-platform it will further allow the integration with other operating systems. Every user that joins or create a community must has a Multichain Node running in their computer and in order to get access to the community blockchain he must have its Multichain address to the Multichain master node. After this step he is now able to create tasks and participate in other tasks. In Figure 4.5 the interactions between the platform and the users are represented, including: participate or create a community, create a task, report the results and get rewarded.

As described in Section 4.3, the incentive scheme is ready to accept different types of incentives, but as an example we could use the gamification where users receive points for completing tasks and spends points by creating tasks. Each user at registering phase will receive 1000 points, by creating a task will spend 100 points and by answering the community tasks will receive 50 points. At the beginning an asset named "Points" will be created in Multichain master node in order to transfer the points to users Multichain wallet.

4.4.1 Incentive Scheme Implementation

In this section we provide some technical details about the implementation of our incentive scheme. Our back-end server was developed using Microsoft .Net technology and all the code was written in C#. We present a class diagram in figure 4.6 of our incentive scheme

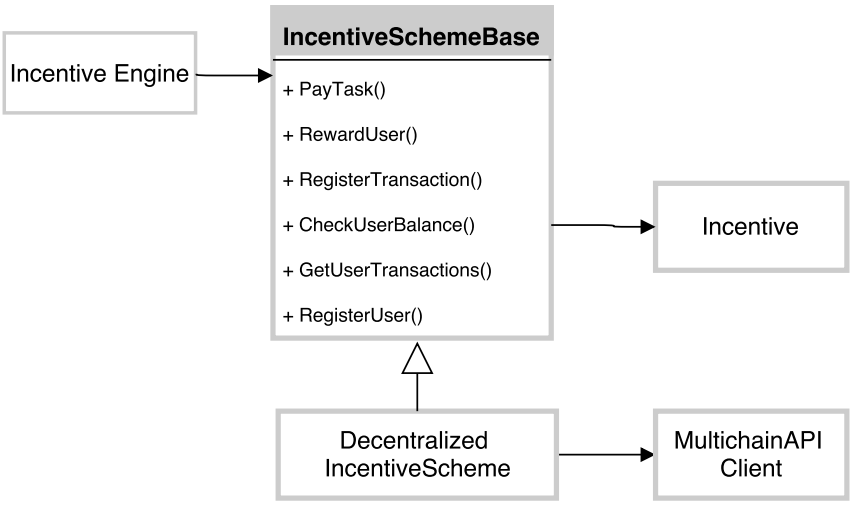

Figure 4.6: Incentive Scheme UML

The incentive engine presented in the crowdsensing system, has defined an incentive scheme and this abstract module provides some methods that each sub class must implement.

Our decentralized incentive scheme uses the a multichain API by including a Dynamic-link Library (DLL)[3] with all the logic and definition of the commands to interact with the multichain. As illustrated in figure 4.6 the incentive scheme allows to have different type of incentives and this can be achieved by extending the Incentive class. As described in figure 4.7, using the Gamification as the incentive, a new class Gamification must be created and must extend the Incentive class. Before initialize the Decentralized incentive scheme, a list of incentive types must be populated with the incentives to create. As shown in this figure the Gamification will use points and badges to reward the users and this list will be used to create the assets on Multichain.

```
public IncentiveEngine(string hostName, int port, string name, string repoURL, string txNodeURL)
        : base(hostName, port, name, "IncentiveEngineLOG")
{
    List<string> incentives = new List<string>();
    Thread checkReportsTh = new Thread(AnalyzeReports);
    try
    {
        //get the repository ref
        ConnectToRepository(repoURL);

        incentives.Add(Gamification.GOLD_BADGE);
        incentives.Add(Gamification.SILVER_BADGE);
        incentives.Add(Gamification.BRONZE_BADGE);
        incentives.Add(Gamification.POINTS);

        this.IncentiveScheme = new DecentralizedIncentiveScheme(this.RepositoryRemote, new Gamification(incentives));

        checkReportsTh.Start();
    }
    catch (Exception ex)
    {
        Log.ErrorFormat(ex.InnerException.Message + " Failed to connect to the Multichain node.");
        Console.WriteLine("[ERROR] - FAILED TO CONNECT THE MULTICHAIN NODE!");
    }
}
```

Figure 4.7: Decentralized Incentive Scheme

[3]https://github.com/PbjCloud/MultiChainLib

Figure 4.8 describes the transaction of incentives between two parties. It starts by checking if the user who is sending the incentives has enough incentives to perform the transaction and the multichain primitive SendAsset is called to transfer the incentive. If the incentive balance is insufficient, the transaction cannot be performed.

```
public override bool RegisterTransaction(string sender, string recipient, IncentiveAssigned incentiveAssigned,
                                         out string transactionId, out bool hasFunds)
{
    transactionId = null;

    hasFunds = false;

    try
    {
        //check if the user can make the transaction
        if (!CheckUserBalance(sender, incentiveAssigned))
        {

            //check if the Manager has  the necessary incentives to distribute among the users
            if (sender.Equals(this.ManagerAddress))
            {
                log.Debug("Manager does not have the necessary incentive quantity. Going to issue more incentives!");
                IssueMoreIncentives(incentiveAssigned);
            }
            else
            {
                hasFunds = false;
                return true;
            }
        }
        hasFunds = true;
        log.DebugFormat("Going to transfer [{0}] from incentive [{1}]", incentiveAssigned.IncentiveQty, incentiveAssigned.IncentiveName);
        var response = this.MultichainClient.SendAssetFromAsync(sender, recipient,
                                                                incentiveAssigned.IncentiveName, incentiveAssigned.IncentiveQty);
        response.Result.AssertOk();
        transactionId = response.Result.Result;
        log.DebugFormat("Transaction Id: [{0}]", transactionId);
        return true;
    }
```

Figure 4.8: Transaction Registration

In figure 4.9 we provide the code for the user registration. Every user has its own pseudonym which is the multichain address. When we call the *GetNewAddressAsync* in the multichain API, it will generate a new address and it will be associated to the user. After this we have to grant permissions to the user in order to start making transactions on blockchain. Finally, we have to transfer to the user's account the initial amount of incentives defined in the Incentive class.

```
public override bool RegisterUser(out string transactionId, out string userAddress, ref string errorMessage)
{
    transactionId = null;
    IncentiveAssigned incentiveAssigned;
    bool hasfunds;
    try
    {
        log.DebugFormat("Goin to register a new user.");
        //get multichain address. this address will be used to deposit and raise rewards.
        var response = this.MultichainClient.GetNewAddressAsync();
        response.Result.AssertOk();
        userAddress = response.Result.Result;
        log.DebugFormat("User Address: [{0}]", userAddress);

        //register user on repository
        this.Repository.RegisterUser(userAddress, this.Incentive);

        //add useraddress to the user nodes list
        this.Repository.AddUserMultichainNode(userAddress, userAddress);

        //give user permissions
        GiveUserPermissions(userAddress);
        incentiveAssigned = Incentive.RegisterUserIncentive();

        if (!RegisterTransaction(this.ManagerAddress, userAddress, incentiveAssigned, out transactionId, out hasfunds))
        {
            errorMessage = String.Format("Cannot Register transaction! From UserManager: [{0}] to User: [{1}]",
                            this.ManagerAddress, userAddress);
            return false;
        }

        return true;
    }
```

Figure 4.9: User Registration

4.5 Summary

In this chapter we present the architecture of our system. We started by presenting the architecture of a crowdsensing system with the implementation of our incentive scheme using the Multichain as the solution to build a incentive scheme supported by the community. We also presented our data quality process to detect the trending answer and reward the users and how the privacy is assured by our incentive scheme. In the last section is presented the prototype implementation of the whole system including the client and server side.

Chapter 5

Evaluation

In this chapter we evaluate our solution based on the following questions:

- Is the blockchain the most suitable model to enable a secure and incentive scheme to community sensing systems?
- Is our incentive scheme capable enough in detecting the trending answer to reward the cooperative users?
- Are the privacy techniques provided by our incentive scheme secure enough to avoid a correlation between the data submitted and user?

5.1 Results

We started by evaluating the time necessary to execute the main activities like creating tasks and reporting results. Our focus was in the evaluation of the decentralized system where the incentives are stored in the blockchain or incentive ledger. The centralized system was the baseline of comparison where the incentive transactions are stored in a centralized database.

5.1.1 Incentive Ledger

We analyzed the time spent in each This4That module but with a special attention on the Incentive Engine. We started by adding a user to the community where the Incentive Engine has to transfer an initial quantity to the user account. These transactions were performed in the centralized version and using the Multichain for the decentralized version. The decentralized approach used a Multichain version with a low level of difficulty to mine a block. By analyzing the results we could observe a difference of the execution time between the Centralized and Decentralized Incentive Engine. With our system configuration, the Centralized takes 34 ms and the Decentralized 63 ms as described in the table 5.1. The API module receives the request, the Repository saves the new user entity and the Incentive Engine transfer the incentive to the user wallet.

Table 5.1: Module Execution Time

Module	Centralized Execution Time (ms)	Decentralized Execution Time (ms)
API	9 ms	9 ms
Repository	13 ms	13 ms
Incentive Engine	34 ms	63 ms

These times includes the operations in the Transaction node for the centralized version and the Multichain Node for the decentralized version. This happens because in the centralized version it just checks if the user can make the transaction and records the incentive transaction with simple operations, but in the decentralized version despite of the process is the same, we have to consider the time in the network to the packets can travels from the Incentive Engine to the Multichain node where the transaction will be stored and distributed to the other community nodes. We also have to consider the internal process of Multichain in sending an incentive (which is known by an asset in Multichain) amount from an address to another. When an asset is transfered to another user it generates an transaction and this transaction as was not generated by a trusted authority like a bank, their authenticity must be checked.

This process in Multichain, as stated in the solution architecture, is different from the *PoW* [24] used in Bitcoin where all the miners try to find the next block by performing a heavy computational work, as we can see in the figure 5.1. This image is relative to the Bitcoin system, where the time to generate a block increases exponentially with the number of zeros required to determine the block hash. In the Bitcoin system they use the PoW to discourage malign nodes to change the blockchain because the cost to change the blocks is too high as shown in figure 5.1.

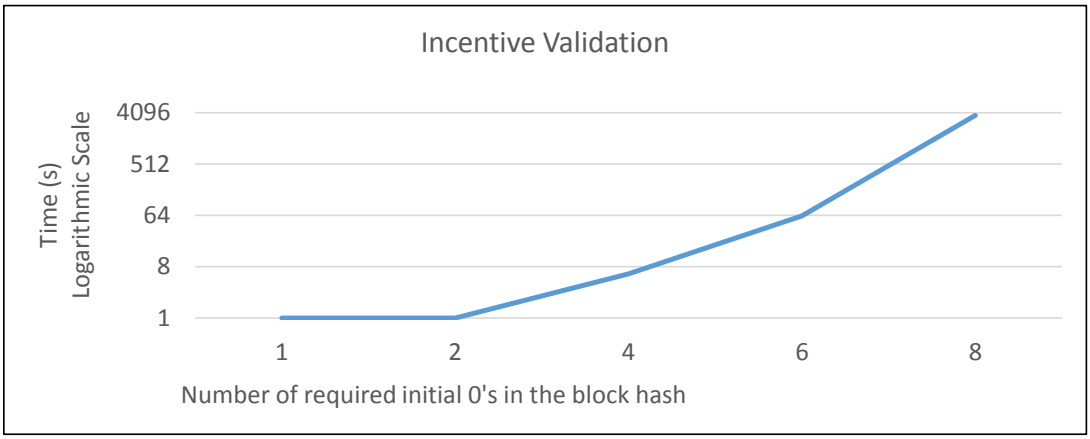

Figure 5.1: Proof-of-Work, Block Generation Time.

For instance to change the content of a block in order to perform a *double spend attack*[1], the further blocks hashes will need to be recalculated. Nevertheless, the double spend can occur by performing a *Attack 51%* or a *Race Attack*. The first is about the proportion of miners that controls the mining process. If an attacker gets 51% of the mining process, he can decides whats

[1]double spend - The same coin is used to pay different transactions

transactions are valid or invalid. The second attack, the Race attack is about to spend the same coin to pay different transactions, but if who is receiving the money do not wait for the transactions confirmations, a double spend can occur.

Comparing the Bitcoin system with our system that uses the Multichain, in the first system the *Attack 51%* can even occurs because every node can be a miner and if there is one powerful node to generating blocks this node will monopolize the process to receive the reward. To avoid this, Multichain by default, used the mining diversity and a lightweight PoW with a low level of difficulty, as stated before in the solution architecture chapter. This mining diversity avoids the monopoly in the validation process but if the same user with a lot of computational power has different Multichain accounts, the mining diversity is not enough, because the mining process is still controlled by the same attacker.

The Multichain provides these two features: the mining diversity and the PoW. These can be adjusted in a configuration file by changing the difficulty of generating blocks and the proportion of miners required to the validation process. So depending on the incentive or the value of the incentive that is stored in the blockchain, the effort to protect the incentives must be adjusted.

A break even point must be achieved in order to define the necessary miners and the level of the PoW. The users must receive their incentives in an acceptable time because if the block that contains transactions takes too long to be generated, the users cannot receive the incentives and spend them until they are valid. On the other hand the difficulty level to mine a block is used to discourage malicious nodes that want to modify the chain in order to change previous transactions and to claim the rewards that were modified because if we change a block, all the next blocks must be generated again and the block hash must be re-calculated. So, as stated before, a break even point must be identified in order to balance the time and the security needed.

5.1.2 Data Quality

In order to evaluate the quality of data reported by the users we used the modified Z-score as explained in section 4.3.2.

We used the hospital use case, presented in Section 1.2 was the evaluation example. Assuming there are people interested to know how is state of the hospitals around its home. An user starts by searching for a community that are talking about hospitals near their home if already there is a community, the user just needs to join the community and create an interactive task to evaluate how is the urgency queue in a hospital nearby her home. She provides 4 options as possible answers to the task: “Empty”; “Few People”; “Lot of People”; “Overcrowded”.

We will compare this approach with other statistical methods to compare and demonstrates why the modified z-score is more appropriated to this scenario.

In Figure 5.2 are trial answers reported by the users to the interactive task about the urgency queue in the hospital. 15 people reported “Empty”, 1 reported “Few People”, 10 reported a “Lot of People” and 20 reported that the hospital was “overcrowded”. Analyzing the data something is strange because there are 16 persons, in the same time window, saying the queue had few people or was empty.

We will start by assessing this data with the Interquartile Range (IQR) [28] method to detect

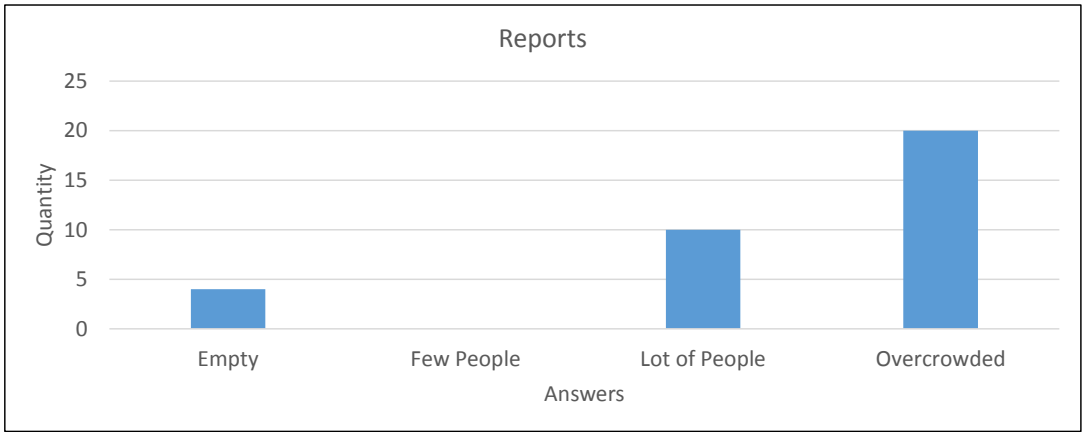

Figure 5.2: Answers

outliers. This method proposes that an outlier must be out of the range:

$$\left[Q_1 - 1.5(Q_3 - Q_1), Q_3 + 1.5(Q_3 - Q_1)\right] \tag{5.1}$$

The Q1 is the first quartile which represents the middle number between the median and the lowest number in our sample, the Q3 represents the middle number between the median and the highest number. Our answers, "empty", "few people", "lot of people" and "overcrowded" are mapped to answer 1, 2, 3 and 4 respectively, the interval to discover outliers based on method defined above is [1.5; 5.5]. This result shows this method is not the most appropriated to evaluate and discover the outliers in our system because the interval of acceptable answers is too large.

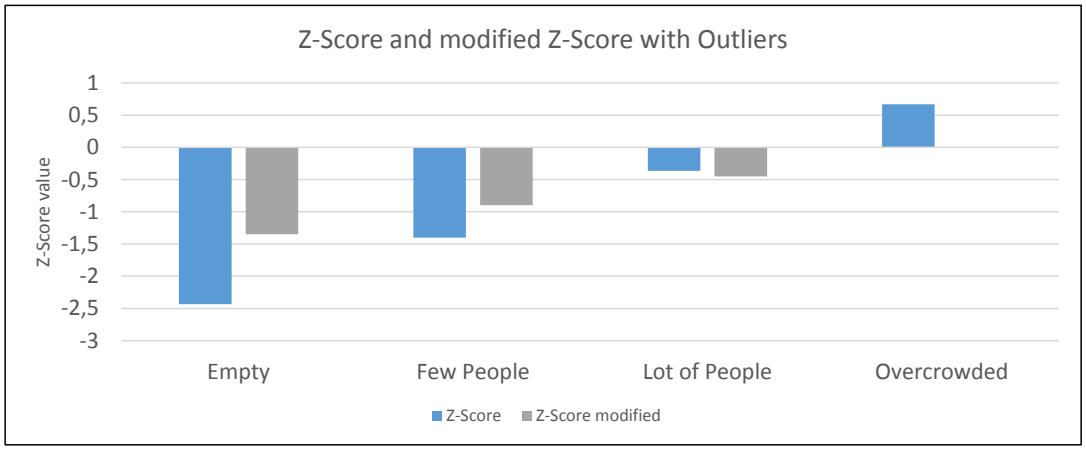

Figure 5.3: Comparison between Z-Score and modified Z-Score

The next method uses the Z-score which tells how many standard deviations an answer is from the mean. In Figure 5.3 the blue bars represents the Z-score value for each answer and as we can see the answer "Lots of People" is the one that is close to the mean. This indicates that answer is probably the right answer to question, but comparing with the modified Z-score this shows that option is not correct and the Overcrowded answer is the right answer. This divergence happens because the Z-score uses the mean as the tendency and the modified Z-score

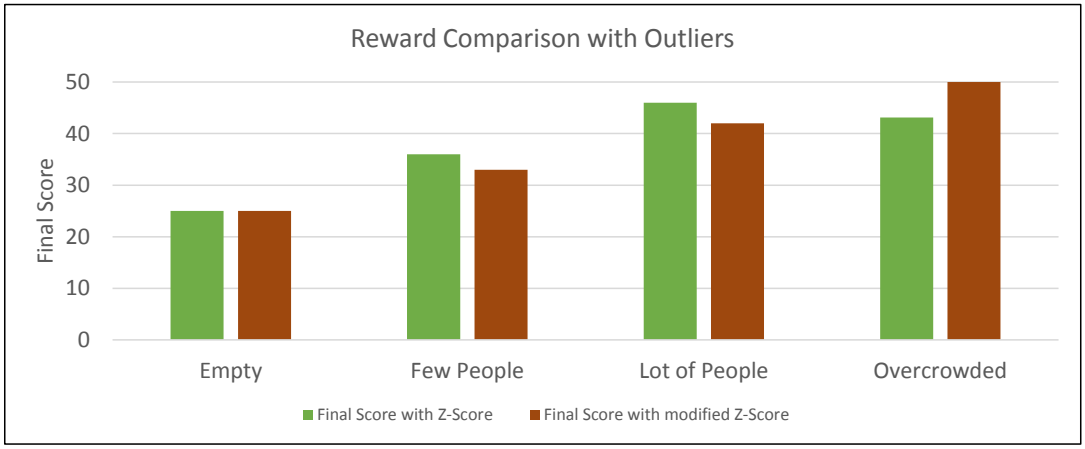

Figure 5.4: Comparing the reward value.

uses the median. The use of the mean in this case can be very influenced by the presence of the outliers and for that reason we chose the modified Z-score to avoid this disturbance.

Analyzing Figure 5.4 that shows the points obtained using the FinalScore function detailed in Section 4.3.2, we can conclude that the users reported "Overcrowded" that seems to be the right answer, using the Z-score method, are not getting the maximum points because of outliers answers but with the modified Z-score which are not influenced by the outliers they can reach the maximum points.

So as the reward for contributing are 50 points, the answer with the Z-score closer to the 0 will receive the maximum points the other answers will receive the points according to the Final Score equation defined in Section 4.3.2.

In this way we can assure a good quality of data because the outliers are discouraged to send wrong responses because they will receive less rewards and they will not be able to participate as many times as they desire.

5.1.3 Privacy Protection

To evaluate the privacy in our incentive scheme we analyzed a possible correlation between the users that are connected to the platform and the transactions performed on Multichain.

We assumed that the crowdsensing platform uses a proxy network to hide the users identity and to do that, we used an API to get random proxies to each request. We deployed the system in our server at Portugal and we analyzed the execution time to each request using different proxy servers around the world and we checked the user IP address that arrives at our server.

Table 5.2: Requests using different proxies

Region / Criteria	Times (s)	Original IP Address	Destination IP Address
Brazil	1.44	188.140.31.217	189.40.191.95
China	2.06	188.140.31.217	183.222.102.106
Ecuador	2.8	188.140.31.217	181.112.228.126
Kenya	1.86	188.140.31.217	197.232.17.83
Russia	4.35	188.140.31.217	212.192.120.42
Ukraine	0.8	188.140.31.217	95.67.57.54

As the server is deployed in the same network, a request without any proxy to calculate the task cost, takes the same time as described in section 5.1.1 to a decentralized version, which is 63 ms because the network time is negligible. Analyzing the results in table 5.2, we can see that the proxy being used impacts the request performance. As long as the distance to the server, the bigger is the time that is takes to complete the request. The advantage of proxies is the IP address masking because the proxy server acts as an intermediary and the IP address that reaches the destination server is the proxy IP address. In our incentive scheme, the transactions are executed in the same way, with or without proxy. So, our incentive scheme is independent on the connection established between the user and the crowdsensing platform because he does not need to know who is really requesting a reward, or pay something, the pseudonym is enough.

Comparing the privacy levels between Multichain and Blockchain used in the Bitcoin system, if the Multichain works as permissioned ledger where some certain users are allowed to enter in the blockchain, only they can observe the transactions that are being processed. But if the Multichain works like the Bitcoin system, using an open blockchain, every transactions are visible to everyone, for those who are inside and outside the blockchain.

5.2 Discussion

In this chapter we started by placing a set of questions to evaluate our incentive scheme. The first question is about the blockchain model as secure ledger to register the incentives. We started by comparing a centralized approach against a decentralized approach, where we concluded that the decentralized approach takes more time to perform the distribution of incentives due to the incentive validation process performed by the miners, but we can also conclude as the blockchain is a distributed database it avoids the single point of failure as it happens in the centralized approach. Although, we are using the Multichain, in this first version, as an open blockchain rather than a permission ledger, our tests revealed that the mining diversity is not enough and the PoW must be adjusted in order to protect the blockchain.

The second question about the capability of our incentive scheme to detect the trending answer to reward the good users. We compared three different statistics methods to detect the outliers, where the modified z-score method revealed the better statistic method because it uses the median value and not the mean as it happens in the standard z-score method because the outliers can influence a lot the mean. Although as we are using statistics methods, our incentive scheme can only obtain the trending answer and we do not know surely if this answer is the

right one answer because if the systems surfers from a massive attack from a group of outliers, their response will be considered the right one.

Regarding the last question about the privacy techniques used to protect the user identity we started by elaborating a survey about privacy in sharing information, we could conclude the privacy is crucial to make people adhering to community sensing systems and our incentive scheme must assure the users privacy as well. In the Multichain, the users' identity is assured by the usage of pseudonyms where the users when registers in the platform he does not need to provide any information about him and gives automatically a Multichain address. When an user is reporting information to the This4That platform, the IP address that reaches the platform can indicates the user who is reporting the data. So we used an proxy API in our mobile application that masks the origin IP address and in that way we avoid a correlation between the data and the user who did it. This will impact on the performance request, but the proxies location can be selected in order to obtain reasonable request time.

Chapter 6

Conclusion

In this dissertation we presented This4That, a incentive scheme for crowdsensing systems to mitigate the problems of user participation, privacy and data quality. We developed an incentive scheme to reward the users when they share useful data with the community. This scheme relies on blockchain technology that does not depend on a central authority and uses the computational resources provided by the community itself to ensure the integrity incentive transactions and the anonymity of users.

The data quality is assured by statistical methods that identify the potential outlier values that answered the crowdsensing task just to receive the reward and not to help others. To evaluate the data quality we compared some statistical methods like, IQR, Z-score and modified Z-score.

Based on our tests the latter better identifies the potential outliers and does not influence the right answers as the z-score influences.

To validate our solution and to evaluate the privacy aspects we did an end-user survey that shows the users privacy is quite important because more than half of the respondents are concerned with the fact of their smartphone is constantly collecting information and approximately 90% will only participate in our system if their privacy is assured.

We also evaluated the response time measurements comparing the centralized approach against the Multichain solution and the bitcoin blockchain. It reveals that the Multichain will take more time to register the incentives but can provide better service availability and fault tolerance. When compared with the bitcoin blockchain, we concluded that the mining diversity is not enough to validate a block and the difficulty of the proof-of-work (PoW) must be adjusted.

Finally our community-as-a-sensor approach was shown to be desired by end-users, as 76% of the respondents answered that they would participate without receiving something in return and 80% said that are available to participate which shows there are many end-users willing to collaborate to help others, whenever it is possible.

6.1 Future Work

This4That maybe improved in several ways. In the context of the mobile application, the graphic interface should be developed beyond the basic interaction to create tasks and reports results.

Regarding the crowdsensing system, we think that an integration with external services like

Facebook or other social networks may allow us to get more accurate information. In this way, we can collect information not only from the users participating in the crowdsensing tasks and we can compare the results provided by the community and the information provided by a external service.

In our solution we still have entities, like users, tasks and results being stored in the server that is supporting the Multichain MasterNode. We think that in a future work, we want to move all these entities to the blockchain and use it not only to store the incentive transactions.

The incentive scheme at the moment uses the blockchain as an open ledger, where everyone is authorized to enter in and this will obligate to use the PoW in order to secure the incentives transactions. Our first idea in the beginning of this dissertation was to use a mobile blockchain, but at the moment there is no such implementation to mobile blockchains due to the smartphone's resources consumption to complete the PoW. So, in order to avoid the users having their own computers contributing to the blockchain as implemented in this first version, we want to go further and use the smartphones to do that. This can be possible, by using the blockchain as permissioned ledger, where there is no PoW to do and use of the smartphone resources is reasonable. But the rules of the community model have to change, because a permissioned ledger works based on a circle of trust, so who creates the community has to have trust in the people in its circle. It can raise privacy concerns because if we want to maintain the user anonymization, the users should be authenticated in a way, maybe using the *group signatures*, without exposing their individual identity. Based on these principles we can build *trusted communities*, which does not require to use a personal computer to support an activity that is mainly done on the smartphone.

Bibliography

[1] V. Benndorf, H.-T. Normann, et al. *The willingness to sell personal data.* Düsseldorf Institute for Competition Economics (DICE), 2014.

[2] D. Bhattacherjee. A comparative study of the incentive mechanisms for mobile crowdsensing. *CSE-E5000 Seminar on Software Systems, Technologies and Security*, 2015.

[3] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. *Center for Embedded Network Sensing*, 2006.

[4] J. Byun and D. Kim. Oliot epcis: New epc information service and challenges towards the internet of things. In *2015 IEEE International Conference on RFID (RFID)*, pages 70–77. IEEE, 2015.

[5] C.-C. Chang, C.-C. Wu, and I.-C. Lin. A secure e-coupon system for mobile users. *International Journal of Computer Science and Network Security*, 6(1):273, 2006.

[6] J. Chang, Z. Pang, W. Xu, H. Wang, and G. Yin. An incentive compatible reputation mechanism for p2p systems. *The Journal of Supercomputing*, 69(3):1382–1409, 2014.

[7] D. Chaum and E. Van Heyst. Group signatures. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 257–265. Springer, 1991.

[8] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick. A survey on privacy in mobile participatory sensing applications. *Journal of Systems and Software*, 84(11):1928–1946, 2011.

[9] B. Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer systems*, volume 6, pages 68–72, 2003.

[10] B. E. Commerce, A. Jøsang, and R. Ismail. The beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference*. Citeseer, 2002.

[11] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos. Anonysense: privacy-aware people-centric sensing. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 211–224. ACM, 2008.

[12] T. C. Du, V. S. Lai, W. Cheung, and X. Cui. Willingness to share information in a supply chain: A partnership-data-process perspective. *Information & Management*, 49(2):89–98, 2012.

[13] R. K. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine*, 49(11):32–39, 2011.

[14] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[15] B. Guo, Z. Yu, X. Zhou, and D. Zhang. From participatory sensing to mobile crowd sensing. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, pages 593–598. IEEE, 2014.

[16] D. Havlik and G. Schimak. State and trends in mobile observation applications. In *iEMSs 2014 Conference*, page 294, 2014.

[17] B. Iglewicz and D. Hoaglin. *How to Detect and Handle Outliers*. ASQC basic references in quality control. ASQC Quality Press, 1993.

[18] C. Kürschner, C. Condea, O. Kasten, and F. Thiesse. Discovery service design in the epcglobal network. In *The Internet of Things*, pages 19–34. Springer, 2008.

[19] I. Lendák. Mobile crowd-sensing in the smart city. In C. Capineri, M. Haklay, H. Huang, V. Antoniou, J. Kettunen, F. Ostermann, and R. Purves, editors, *European Handbook of Crowdsourced Geographic Information*, chapter 26, pages 353–369. Ubiquity Press, London, 2016.

[20] P. G. Leon, B. Ur, Y. Wang, M. Sleeper, R. Balebako, R. Shay, L. Bauer, M. Christodorescu, and L. F. Cranor. What matters to users?: factors that affect users' willingness to share information with online advertisers. In *Proceedings of the ninth symposium on usable privacy and security*, page 7. ACM, 2013.

[21] C. Y. Liau, X. Zhou, S. Bressan, and K.-L. Tan. *Efficient Distributed Reputation Scheme for Peer-to-Peer Systems*, pages 54–63. Springer Berlin Heidelberg, 2003.

[22] T. W. Malone, R. Laubacher, and C. Dellarocas. Harnessing crowds: Mapping the genome of collective intelligence. 2009.

[23] R. B. Messaoud, N. Sghaier, M. A. Moussa, and Y. Ghamri-Doudane. On the privacy-utility tradeoff in participatory sensing systems. In *NCA 2016*, pages 1–8, 2016.

[24] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[25] M.-R. Ra, B. Liu, T. F. La Porta, and R. Govindan. Medusa: A programming framework for crowd-sensing applications. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 337–350. ACM, 2012.

[26] J. Shi, R. Zhang, Y. Liu, and Y. Zhang. Prisense: privacy-preserving data aggregation in people-centric urban sensing systems. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[27] K. Traub, F. Armenio, H. Barthel, P. Dietrich, J. Duker, C. Floerkemeier, J. Garrett, M. Harrison, B. Hogan, J. Mitsugi, J. Preishuber-Pfluegl, O. Ryaboy, S. Sarma, K. Suen, and J. Williams. The EPCglobal architecture framework 1.4, 2010.

[28] J. W. Tukey. *Exploratory Data Analysis*. Behavioral Science: Quantitative Methods. Addison-Wesley, Reading, Mass., 1977.

[29] Y. Ueyama, M. Tamai, Y. Arakawa, and K. Yasumoto. Gamification-based incentive mechanism for participatory sensing. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, pages 98–103. IEEE, 2014.

[30] F.-J. Wu and T. Luo. A generic participatory sensing framework for multi-modal datasets. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6. IEEE, 2014.

[31] B. Yang and H. Garcia-Molina. Ppay: micropayments for peer-to-peer systems. In *Proceedings of the 10th ACM conference on Computer and communications security*, pages 300–310. ACM, 2003.

[32] Y. Zhao and Q. Zhu. Evaluation on crowdsourcing research: Current status and future direction. *Information Systems Frontiers*, 16(3):417–434, 2014.

Appendix A

This4That Application Programming Interface (API)

In this appendix we will present the API that is used in the mobile application to interact with the back-end server.

These endpoints are implemented, but at this moment only supports the creation of interactive tasks because the focus is not implementing a full-feature crowdsensing framework.

We evaluated our solution based on the execution of these endpoints with the Microsoft Unit Testing Framework.

All the endpoints return a errorCode, a errorMessage and an object Response which contains the useful data. All endpoints return data in JSON format.

A.1 POST api/v1/user

This endpoint is used to register a new user in the community, by invoking it the user receives a multichain address that is used to transfer the incentives, as we can see in the example below. The user will receive the initial quantity of incentives to start creating tasks. The Incentive Engine will order to the Multichain master node using JSON Remote Procedure Call (RPC) to invoke the Multichain API.

```
{
  "errorCode": "1",
  "errorMessage": "sucess",
  "response": {
        "multichainAddress": "1RDetFxFBBckqEydJyCbq6vERNsPuTsRgWuXCs"
  }
}
```

A.2 POST api/v1/task/cost

When an user is creating a task, it must be evaluated in order to know how much will cost this type of task. In this case this is an interactive task, which has a question and a set of possible

answers. This method will ask to the Task Creator the cost of the task. At the moment there is no technique applied to calculate the task cost, so the task cost will assume the default value of the incentive.


```
{
    "userId":"1RDetFxFBBckqEydJyCbq6vERNsPuTsRgWuXCs",
    "task":{
        "name":"Queue Lunch",
        "topic":"Tecnico Lisboa Canteen",
        "expDate":"2017-06-04T17:41:00",

        "interactiveTask":{
          "question" : "How is the queue to lunch today at university canteen?",

          "answers" :[
            {
              "answer" : "Empty"
            },
            {
              "answer" : "Few People"
            },
            {
              "answer" : "Lot of People"
            },
            {
              "answer" : "Impossible to lunch"
            }
            ]
        }
    }
}
```


This endpoint will return which incentive will be used to pay the task and its quantity. In this example we applied a gamification incentive scheme, so the task will cost 100 points.


```
{
  "errorCode": 1,
  "errorMessage": "success",
  "response": {
    "incentiveToPay": {
      "name": "POINTS",
      "quantity": 100
    }
  }
}
```

A.3 POST api/v1/task/

In order to submit a task to the platform,a task specification like the previous must be used. This request will generate a task in the system and return its Id and the association transaction in the Multichain. This transaction is generated by transfering 100 points from the platform to the user multichain wallet. In our system, the Task Creator will create the task and the Incentive Engine will order to the Multichain to transfer 100 points from the user to the platform. The body request is the same as described in the endpoint POST api/v1/task/cost.

```
{
  "errorCode": 1,
  "errorMessage": "success",
  "response": {
    "taskId": "628f3633-bb85-4289-8f7c-fef5acd77cab",
    "transactionId": "e87f3791211912ad3b229aaa58774d37e2a2299a6aabadb"
  }
}
```

A.4 GET api/v1/topic/

This endpoint is used to get the list of topics that is talking in the community and based on these topics is possible to obtain the tasks created by everyone. The Task Distributor will provide all the topics.

```
{
  "errorCode": 1,
  "errorMessage": "success",
  "response": [
    "topic": "Tecnico Lisboa Canteen"
  ]
}
```

A.5 POST api/v1/subscribe

This endpoint is used to subscribe a topic in order to receive tasks under this topic. This is the specification to subscribe a topic.

```
{
  "userId" : "1ER2eKm1yAKc2R5hcHKHcPq4UMbjuf6aj7Pdt",
  "topicName" : "Tecnico Lisboa Canteen"
}
```

A.6 GET api/v1/user/{userId}/task

This endpoint is used to obtain all the created and subscribed tasks and its detailed information .The Task Distributor will start to distribute tasks for the user based on the subscribed topics.


```
{
  "errorCode": 1,
  "errorMessage": "success",
  "response": [
    {
      "taskID": "628f3633-bb85-4289-8f7c-fef5acd77cab",
      "name": "Lunch Queue",
      "expirationDate": "2017-06-04T17:41:00",
      "topic": "Tecnico Lisboa Canteen",
      "sensingTask": null,
      "interactiveTask": {
        "question": "How is the queue to lunch today at university canteen?",
        "answers": [
          {
            "answer": "Empty",
            "answerId": "eae63a0a"
          },
          {
            "answer": "Few People",
            "answerId": "fa894e77"
          },
          {
            "answer": "Lot of People",
            "answerId": "3333753b"
          },
          {
            "answer": "Impossible to lunch",
            "answerId": "e45869cc"
          }
        ]
      }
    }
  ]
}
```

A.7 POST api/v1/report/InteractiveTask

This endpoint is used to report the task result to the platform. The Report Aggregator will receive the answer based on the taskId and the answer will be stored in the Repository.

```
{
    "userId" : "1RDetFxFBBckqEydJyCbq6vERNsPuTsRgWuXCs",
    "taskId" : "628f3633-bb85-4289-8f7c-fef5acd77cab",
    "timestamp" : "2017-06-25T17:41:00",
    "result" : {
                    "answerId" : "eae63a0a"
    }
}
```

A.8 GET api/v1/task{taskID}/results

This endpoint is used to check the task results. When the task expires this endpoint will return the number of responses for each answer.

```
{
  "errorCode": 1,
  "errorMessage": "success",
  "response": [
    "interactiveTask": [
        {
        "answerId": "eae63a0a",
        "results": 3
        },
        {
        "answerId": "fa894e77",
        "results": 5
        },
        {
        "answerId": "3333753b",
        "results": 10
        },
        {
        "answerId": "e45869cc",
        "results": 10
        }
    ]
  ]
}
```

A.9 POST api/v1/task/reward

This endpoint is used to check if the task results are already validated by the quality process. When the task expires the results are evaluated and different reward quantities will be assigned to each answer. Based on the answer provided by the user, the Incentive Engine will order to the Multichain node to transfer the incentives to the user. The request has the following body:

```
{
    "userId" : "1RDetFxFBBckqEydJyCbq6vERNsPuTsRgWuXCs",
    "taskId" : "628f3633-bb85-4289-8f7c-fef5acd77cab"
}
```

The following response is an example to this request:

```
{
  "errorCode": 1,
  "errorMessage": "success",
  "response": {
    "reward": {
      "quantity": "50",
      "incentive": "POINTS"
    },
    "txId": "bc551321b2e038c3d4d7ca7a30c03f292e175c60b155c055049ab908"
  }
}
```