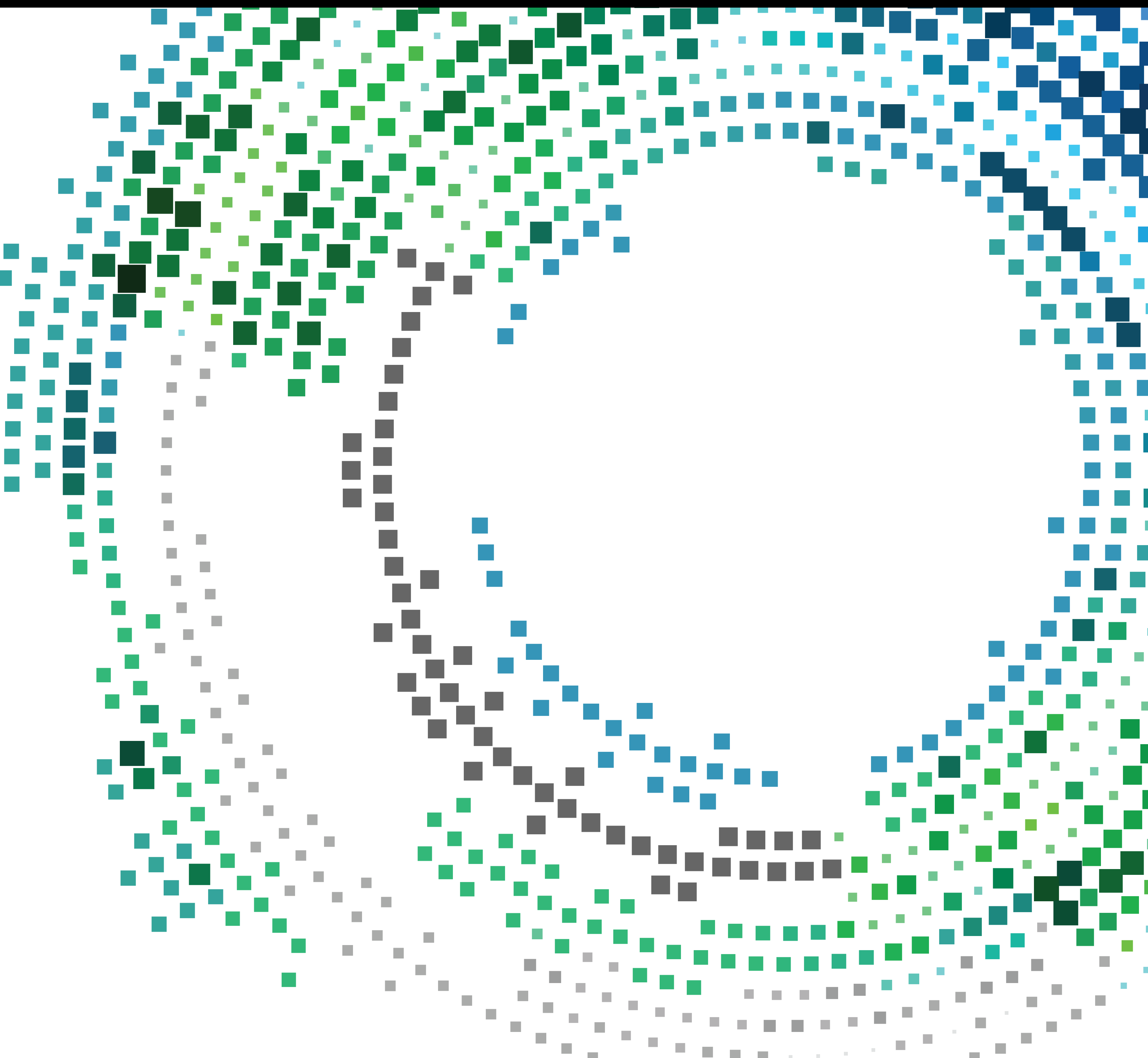
Mobile Information Systems

Collaborative Mobile Applications

Lead Guest Editor: César A. Collazos
Guest Editors: Alejandro Fernández and Patricia Paderewski



Collaborative Mobile Applications

Collaborative Mobile Applications

Lead Guest Editor: César A. Collazos
Guest Editors: Alejandro Fernández and Patricia Paderewski

Contents

Hindawi
Mobile Information Systems
Volume 2021, Article ID 9223096, 13 pages
https://doi.org/10.1155/2021/9223096

Research Article

Cooperation-Driven Virtual Terminal Coalition Formation Games for Task Assignment in Mobile Crowdsensing

Haifei Yu, Shiyong Chen, Xiang Liu, and Yucheng Wu

School of Microelectronics and Communication Engineering, Chongqing University, Chongqing, China

Correspondence should be addressed to Yucheng Wu; wuyucheng@cqu.edu.cn

Received 7 June 2021; Revised 6 July 2021; Accepted 16 August 2021; Published 3 September 2021

Academic Editor: Patricia Paderewski Rodríguez

Mobile crowdsensing (MCS) is a popular way of data collection, which forms the large-scale sensing system by smart mobile terminal users and provides multimodal sensor data. In the sensing scenario, there are various sense resource requirements of tasks released by the platform. One of the most urgent issues in MCS is how to choose corresponding users with appropriate sense resources to accomplish assigned tasks. In this article, cooperating among a host of users to perform sense tasks is considered. Firstly, the cooperation among users to accomplish the sense tasks is described as an overlapping coalition formation game (OCF game). In addition, an initial coalition method of using social networks (SN) is proposed to accelerate the formation of coalition. Finally, the cooperation degree is used to describe the cooperative relationships among users, and virtual terminal coalition formation (VTCF) algorithm is proposed to simplify the process of coalition formation. The simulated results show that the proposed approach effectively improves the system's utility under the constraints of task cost and sense quality.

1. Introduction

With the development of Internet technology and the popularization of smart devices, diverse high-performance sensors have been embedded into smart terminals, which can monitor and collect surrounding data to promote the development of mobile crowdsensing (MCS). Many new applications have been developed based on smart terminal sensing, such as smart healthcare [1], environment sensing [2], traffic condition detection [3], news [4], and even social networks [5]. Generally, these applications can be divided into three types [6]: personal sense, group sense, and community sense [7]. The mobile crowdsensing, in which a certain number of users are encouraged to participate in smart terminal sensing, is applied to ensure the sense quality of applications.

A typical MCS framework consists of task requesters, sense platforms, and mobile users [8–12]. In MCS, the sense platform assigns tasks required by requesters to the users, and then the matched users complete the tasks through mobile terminals. Finally, users upload the sensed results to the platform. Users upload information collected by sensors to the sense platform via wireless communication technologies such as WIFI and cellular network [13]. With the development of 5G [14], SDN [15], D2D [16], and other communication technologies, the communication in MCS has become faster, and the communication performance has also been greatly improved.

While sensing and uploading data, users need to consume their resources, such as battery energy, computing, and communication. Therefore, unless users are rewarded for compensating for the resources they consume, they may not be interested in sensing tasks. If the platform cannot recruit enough users, then sense quality of the task could not be guaranteed, which will affect the service quality provided by the platform. Hence, some works have considered the incentive mechanism to motivate users participating in the sensing task for the MCS system. The mechanism [17] considers the potential contribution of reverse auction and maintaining existing users when participants recruit new users. The optimization goal is to optimize the composition of users in the system while reducing the system cost. By quantifying the reputation of users [18], an incentive mechanism based on local differential privacy is proposed, which includes four aspects: incentive, reputation, data disturbance, and data aggregation. The incentive mechanism can select users who can offer more accurate data and compensate users who have spent privacy costs. Simultaneously, the reputation mechanism quantifies the reputation of users to improve the benefits of users.

It is worth pointing out that, during choosing sense tasks, it is assumed that users will make independent decisions and do not know how their strategies affect the sense performance of each task in most literatures. Therefore, most users are likely to participate in a popular task, while other tasks distributed by the platform cannot recruit enough participants, which leads to uneven allocation of users' resources and a decrease in sensing quality. When the platform releases complex tasks, the user's limited sense resources may affect the completion of the task.

To prevent the problem of uneven resource allocation mentioned above, cooperating among users is considered. Through cooperation and information exchange, users can understand the specific circumstances of their selected tasks, making more intelligent resource allocation choices. Considering the selfishness and reasonability of users, the coalition game theory is considered as an appropriate method to describe users' cooperation relationships [19]. However, most of the literatures using coalition formation game for user cooperation assume that one user can only join one coalition [20], which is not suitable for practical scenario. As each user can participate in multiple sense tasks in the MCS system, the coalitions for different sense tasks may be overlapped.

To tackle the issues discussed above, a novel incentive mechanism that uses social networks to recruit users for the MCS system is described. Users can cooperate through social networks to complete tasks released by the platform. In the process of coalition forming, the concept of cooperation degree inspired by the concept of trust degree [21] is proposed to describe cooperation relationships among users. The main contributions of this paper are summarized as follows:

(1) The concept of cooperation degree in the MCS system is introduced to better describe cooperation relationships among users. When the user completes the task, the corresponding cooperation degree will be updated at once, which estimates the positivity of users taking part in the coalition.

(2) To accelerate the formation of coalitions, a coalition initialization method based on SN according to the characteristics of users' social networks is proposed, which makes coalition formation much faster.

(3) The task allocation problem in MCS is described as an OCF game. To simplify the process of coalition formation, a virtual terminal coalition formation algorithm (VTCF algorithm) is proposed, which can transform an overlapping coalition problem into a nonoverlapping coalition formation one. Moreover, virtual terminals choose different coalitions to join and allocate sense resources to maximize the system utility. Through simulated experiments, the proposed and reported task allocation algorithms are evaluated and compared. Experimental results have verified the effectiveness of the proposed algorithm.

The rest of this paper is organized as follows. First of all, the related works are reviewed in Section 2; and then system models are described in Section 3. In Section 4, an OCF game is used to describe task assignment of MCS. A coalition initialization method based on SN and the VTCF algorithm are proposed to solve the forming of coalition and the utility of platform is evaluated. In Section 5, the proposed task allocation algorithm is simulated and the effectiveness of the proposed algorithm has been verified. In the end, the related works discussed in this paper are concluded in Section 6.

2. Related Works

At present, some researchers have summarized the research work of task assignment [22–26] and cooperation mechanism [27–30] from different perspectives.

Usually, participants will not voluntarily participate in tasks provided by the sense system. Due to the selfishness of users, the appropriate task allocation and cooperation mechanism are very important to improve the performance of the system. The two complementary modes of opportunistic participation and participatory participation into a two-stage hybrid framework called HyTasker were proposed in [22]. In an offline stage, one group of users (called opportunistic users) is selected, and they complete MCS tasks in daily life. In an online stage, the other groups of users (called participatory users) is assigned; they are required to perform tasks that opportunistic users cannot accomplish. A task management framework called Spatial Recruiter to match staffs to perceptual tasks effectively was proposed in [23]. A greedy heuristic method is designed to select and allocate workers. An optimal incentive mechanism for service providers was studied in [24]. The two-stage Stackelberg game method was used to analyze the participation degree of mobile users and optimal incentive mechanisms of service providers by backward induction. In order to motivate the participants, an incentive mechanism was designed by taking the social network effect, which was from the underlying mobile social field, into consideration. An incentive mechanism algorithm of joint coverage and reputation based on Stackelberg game theory was developed in [25]. In the first place, location information and the historical reputation of mobile users were used to select optimal users, meeting the user's requirements for information quality. Secondly, the two-stage Stackelberg game was employed to analyze the sense level of mobile users; in this way, an optimal incentive mechanism of the server center was obtained. Last but not least, the expectation maximization (EM) algorithm is used to evaluate the data quality of tasks, and the historical reputation of each user is updated accordingly. An incentive mechanism was proposed in [26], which first selected some users from online social networks as seeds and then social relationships among users were used to influence selection of potential candidates for task execution. The target was to maximize the number of recruiters and increase coverage based on the initial seed.

A smartphone collaborative sense system to encourage users to participate in sense tasks was considered in [27]. In the case of limited wireless channel resources, the authors therein discussed the incentive mechanism of cooperation and noncooperation and expressed the cooperation of users to complete tasks as overlapping coalitions forming game. Users maximize their own interests by choosing multiple coalitions to join and invest in wireless resources. Compared with noncooperative methods, the system performance will be improved if the cooperative method has been adopted. The cooperation between secondary users (SUs) in cognitive radio to improve sense performance and spectrum efficiency was discussed in [28]. Each SU makes its own cooperative sense decision according to its own business needs. SU's cooperation was described as the formation of overlapping coalitions and proved the stability of the coalition structure. The cooperation strategy between roadside units (RSU) was studied in [29]. Through cooperation, the RSU can coordinate the data types transmitted to the vehicle through the vehicle-to-roadside (V2R) communication link. This solution used the underlying content-sharing V2R communication network increased the diversity of information circulation in the network. Compared with the noncooperative situation, the average rate of return has been improved. A strategy based on a genetic algorithm approach was proposed to achieve optimal collaborative learning groups and the system performance in [30].

3. System Model

In this section, the MCS system model is given, which is composed of a service platform, a wireless base station (BS), and some users carrying smart terminals, as shown in Figure 1. In the system, the sensing platform publicizes N sense tasks, and the set of sense tasks is denoted by $N = \{1, 2, \ldots, N\}$. The set of smartphone users is denoted by $M = \{1, 2, \ldots, M\}$. To simplify the model analysis, it is assumed that a user has only one mobile terminal. The concept of the terminal is equivalent to the user in the subsequent sections. M mobile terminal users form a social network, represented by graph $G = (V, E)$; in a G, the vertex $v \in V$ represents a user on the social network and connects with other users through undirected edges; and the edge $e \in E$ stands for the communication from its source user to its direct destination user. The weight of the edge indicates the degree of cooperation between the mobile users m and n is expressed as $\mathrm{co}(m, n)$. If the value of the cooperation degree is larger, it means that the cooperation relationship is more positive.

It is supposed that a mobile terminal user has at most r_0 types of sense resources and there are l kinds of sense resources in the MCS system; each user $i \in M$ is associated with sense resource $s(i)$, which is a set of sense resources owned by user i, denoted as $s(i) = \{1, 2, \ldots, r_0\}$; for example, the user has three kinds of sense resources: $s(i) = \{1, 2, 3\}$. Sense cost of sense resources $c(i, j)$ represents the cost of user i providing sense resources j. $C(i)$ expresses the total cost of user i

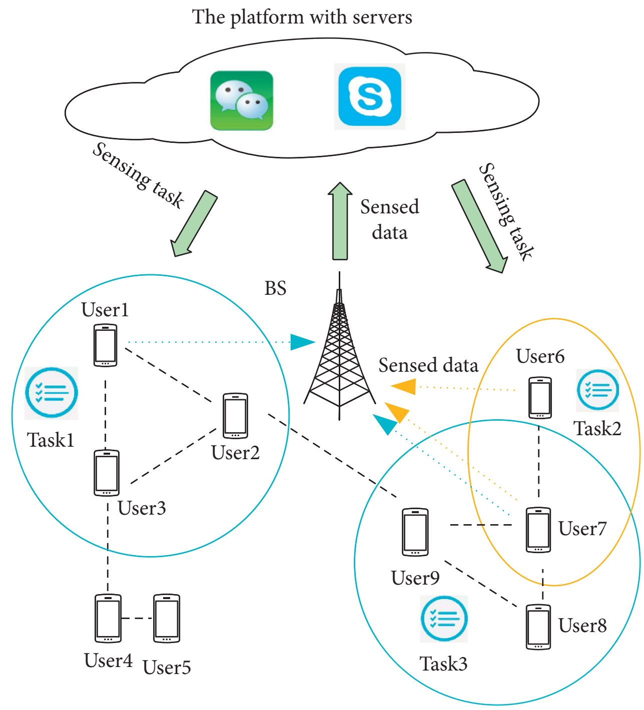


Figure 1: System model of MCS.

providing all sense resources. Neighbor $n(i)$ is the set of social neighbors of user i, where $n(i) = \{b | (i, b) \in E\}$, and user b is the neighbor of user i; for example, the neighbor set of user 1 is $n(1) = \{2, 3\}$ as shown in Figure 1. Sense quality of sense resources $q(i, j)$ expresses sense quality of user i providing sense resource j. The larger value of $q(i, j)$ means stronger sense ability of user i.

Task $k \in N$ is denoted by $\langle R_k, D_k, B_k, U_k, \mathrm{CA}_k \rangle$. R_k expresses a group of sense resources required for task k. D_k represents the quality requirements of sense resources for task k. B_k represents the budget of task k. U_k expresses the revenue of task k, and CA_k expresses the amount of computation required to execute task k. It is supposed that a task always requires multiple users with different sense resources to cooperate with each other. Each user can freely allocate sense resources among these tasks. An $N \times K$ matrix $\mathbf{S}$ represents resource allocation, in which $s_{i,j}^{k} = 1$ denotes that user i provides the j-th resource for task k. Resources of the terminal i are limited as

$$\sum_{j \in s(i)} \sum_{k \in N} s_{i,j}^{k} \leq r_0. \quad (1)$$

The utility function of the certain task, which is a function of terminal sense cost and task budget, is used to evaluate the performance of accomplishing each task. The set of users participating in task k is denoted as M_k. The utility function of task k is given by

$$U_k = \frac{\mathrm{RN}(k)}{|R_k|} \times B_k - \sum_{i \in M_k} C(i), \quad (2)$$

where $C(i) = \sum_{j \in s(i)} c(i, j)$, $\mathrm{RN}(k)$ is the number of sense resources all users actually provided for task k, and

$(\text{RN}(k)/|R_k|)$ is the completion rate of task k. When the user completes the sense task, the cost usually includes consumption of battery energy and computing resources [31]. When the user i performs sense task, the consumption of computing resources and battery energy can be expressed as t_i and e_i. Let $f_i > 0$ denote the computing capability of terminal i in terms of CPU cycles/s. If user i provides sense resource j to execute sense task k, the task completion time is $t_i = (\text{ca}_j/f_i)$, where ca_j represents the amount of computation required to provide sense resource j [32], where $\text{ca}_j = (\text{CA}_k/|R_k|)$.

To calculate the energy consumption of the terminal performing tasks, the energy consumption model widely used $\mathscr{E} = \kappa f^2$ for each computing cycle [33], where κ is the energy coefficient depending on the chip structure and f is the CPU frequency. Therefore, the energy consumption e_i^j of terminal i when providing resource j is calculated as

$$e_i^j = \kappa f_i^2 \text{ca}_j. \tag{3}$$

Therefore, the total cost of user i participating in task k to provide resource j is

$$c(i, j) = \kappa f_i^2 \text{ca}_j + \frac{\text{ca}_j}{f_i}. \tag{4}$$

Considering all tasks of the platform, the utility function of the platform is given by the following formula:

$$\begin{aligned} U_{\text{platform}} &= \sum_{k\in N} U_k \\ &= \sum_{k\in N}\left(\frac{\text{RN}(k)}{|R_k|}\times B_k - \sum_{i\in M_k}\sum_{j\in s(i)} c(i, j)\right). \end{aligned} \tag{5}$$

If users consume their resources without getting any benefits from the sensing platform, they will not be willing to participate in the task. To encourage users to participate in the task and ensure the quality of task completion, an incentive mechanism is designed in which users will be rewarded in the platform according to their contributions to the task. The total revenue of user i can be defined as

$$P_i = \sum_{k\in N} p_{ik} - C(i), \tag{6}$$

where p_{ik} represents the benefit of user i from task k. The total benefit for user i can be described as $\sum_{k\subset N} p_{ik} = \omega \sum_{k\subset N} U_k$, where ω is an incentive factor. Because the benefits of users from different tasks are different due to the resources invested in the task and the utility of task, the resource allocation among tasks is significant to maximize their benefits and improve the platform's performance.

3.1. Centralized Case. Note that users are not willing to participate in tasks without pay. Therefore, the platform needs to reward the contribution of users, which is often called an incentive mechanism. For an existing MCS system, the platform provides an incentive mechanism, while users try to get maximum profits from the platform. The centralized case represented the upper bound of the system sense performance [27], where users are fully scheduled by the platform. In the centralized case, users are forced to participate in the task without any reward. The platform determines how users participate in the task. This example describes the upper bound of the system's sense performance, which does not exist in practice because users are unwilling to participate in tasks without any reward.

Given the constraints in (1), the utility of platform in centralized case is

$$\begin{aligned} U_{\text{platform}} &= \max_{S} \sum_{k\in N} U_k, \\ \text{s.t.} & \sum_{j\in s(i)}\sum_{k\in N} s_{i,j}^k \le r_0. \end{aligned} \tag{7}$$

This is a 0-1 integer nonlinear programming (INLP) problem, which can be solved by optimization algorithm [34].

3.2. Cooperative Approach. The resource allocation problem of the terminals can be modeled as an OCF game. The terminal has the right to determine the resources allocated to each task. In this scenario, it is supposed that a coalition corresponds to a task and the members of the coalition complete relevant sensing task in the OCF game. Moreover, the members of the coalition can allocate sense resources for different tasks. Based on the OCF game model, the overlapping coalition formation algorithm is suitable for multiterminal sense resource allocation. The algorithm can converge to a stable overlapping coalition structure (OCS), improving overall sense performance. The coalition for completing the task can be described as the coalition members and the allocation resources of members.

Definition 1. The sense resource allocation problem of terminal is modeled as OCF game, which is expressed as (M, V), where M represents the set of terminals and V represents the coalition value; the coalition value of coalition C_k is defined as

$$V(C_k) = \frac{\text{RN}(k)}{|R_k|}\times B_k - \sum_{i\in C_k}\sum_{j\in s(i)} c(i, j). \tag{8}$$

To represent the cooperation behavior among terminals, the cooperation degree between the terminals m and n is described as $\text{co}(m, n)$. The cooperation degree of the coalition is represented by $\text{Co}(C_k) = \sum_{m,n\in C_k} \text{co}(m, n)$. Assuming that the cooperation degree among all terminals with cooperation relationships is initialized with the same initial value $(\text{co}^{(0)}(m, n) = c > 0)$, among them, the cooperation degree of coalition member i is expressed as

$$\text{co}(i) = \frac{\text{Co}(C_k)}{|R_k| \sum_{i\in C_k} p_{ik}} g(i) p_{ik}, \tag{9}$$

where $\sum_{i\in C_k} p_{ik}$ is the income of all users participating in task k and $g(i)$ represents the number of sense resources contributed by terminal i to the coalition. The update of cooperation degree among users will be introduced in the third part of Section 4.

In OCF game model, the coalition structure is a set of the total coalitions for accomplishing all tasks and is denoted as $\prod = \{C_1, C_2, \ldots, C_N\}$. Each task corresponds to a coalition. Therefore, the size of the coalition structure is fixed, which is equal to N.

When a coalition completes the task, it can be called an effective coalition. The set of effective coalitions is called an effective coalition structure (ECS). In the process of coalition formation, on the one hand, terminal resource allocation needs to maximize the cooperation degree of coalition. On the other hand, it needs to form the coalition with the most significant coalition value. The sense resource allocation problem of all terminals can be formed as the following optimization problem:

$$U_{\text{platform}} = \max_{S} \sum_{C_k \in \prod} \alpha \mathrm{Co}(C_k) + \beta V(C_k),$$
$$\text{s.t.} \begin{cases} R_k \subseteq \bigcup_{i \in C_k} s(i), \\ q(i,j) \geq d(k,j), \quad \forall i \in C_k, j \in s(i), \\ \sum_{i \in C_k} C(i) \leq B_k, \end{cases} \tag{10}$$

where α, β are scale factors, satisfying $\alpha + \beta = 1; \alpha, \beta \in [0,1]$ and $d(k,j)$ represents the sense quality requirement of sense resource j in coalition C_k. The goal is constrained by three equations: $R_k \subseteq \cup_{i \in C_k} s(i)$ means that the sense resources provided by the terminal must satisfy sense resources required by sense task k, $q(i,j) \geq d(k,j), \forall i \in C_k, j \in s(i)$, means that the quality of the sense resources provided by terminal i is not lower than the quality of sense resources required by the task, and $\sum_{i \in C_k} C(i) \leq B_k$ means that cost of coalition C_k to complete the task does not exceed the budget of the task.

Usually, the coalition pays users a certain reward in some form according to its resource allocation. Hence, a reasonable method of income distribution is very crucial to motivate them to join a coalition. In the OCF game model, the return is a finite vector, expressed by $\mathbf{P} = \{p^1, p^2, \ldots, p^N\}$. p^k is the revenue of coalition C_k. In the proposed OCF game, it is assumed that the total profits of the coalition are affected by the participating degree of coalition members. The coalition will offer rewards to its members. For terminal $i \in C_k$, the revenue of terminal i from coalition C_k is defined as follows:

$$p_{ik} = \omega V(C_k) \frac{\sum_{j \in s(i)} s_{i,j}^k}{\sum_{i \in C_k} \sum_{j \in s(i)} s_{i,j}^k}, \tag{11}$$

where $s_{i,j}^k$ represents the resources provided by terminal i for coalition C_k and $\sum_{j \in s(i)} s_{i,j}^k$ represents total resources provided by terminal i for coalition C_k. With the revenue that terminal i obtained from coalition C_k, the terminal i's total revenue from the platform is

$$P_i = \sum_{C_k \in \prod} \omega V(C_k) \frac{\sum_{j \in s(i)} s_{i,j}^k}{\sum_{i \in C_k} \sum_{j \in s(i)} s_{i,j}^k} - C(i). \tag{12}$$

4. Virtual Terminal Coalition Formation (VTCF) Game

Since one coalition corresponds to one task, one terminal can participate in multiple tasks to make full use of the terminal's resources. Based on distributed social networks, in this section, the coalition forming methods are proposed, where the terminal's sense cost, sense quality, and the degree of cooperation between terminals are considered to be used to comprehensively express preference relationships in the process of dynamic coalition formation.

4.1. Virtual Terminal. To solve the problem of overlapping coalition formation in an uncertain distributed environment, the concept of the virtual terminal is introduced to reduce the complexity in the overlapping coalition formation process, which is inspired by the concept of virtual auctioneer proposed in [35]. In the actual application of the Internet of things, a terminal usually has multiple sense capabilities. For instance, in the application of air quality monitoring, a terminal can collect CO, NO_2, and other gas concentrations at the same time. A terminal with multiple sense capabilities is virtualized into multiple virtual terminals. Each virtual terminal has only one sense capability. For example, terminal i has a sense capability set $s(i) = \{1,2,3\}$, which can be replaced by three virtual terminals $V(i) = \{v_i^1, v_i^2, v_i^3\}$, and each virtual terminal has only one sense capability. In this case, each virtual terminal uses distributed mobility to form a coalition structure. A terminal can participate in multiple tasks in the coalition structure, while a virtual terminal can only participate in one task. Therefore, overlapping coalition formation is simplified as the problem of nonoverlapping coalition formation, if the concept of the virtual terminal is adopted.

In social networks, each terminal can participate in multiple tasks, which leads to the overlapping coalitions. According to the characteristics of the coalition, the virtual terminal coalition form (VTCF) game is described. The terminal can join multiple coalitions at the same time, while the virtual terminal can only join one coalition. A multisense ability terminal is replaced by multiple virtual terminals with single-sense ability. The virtual terminal participates in the allocation of sense resources. The virtual terminal of terminal i is represented by $V(i) = \{v_i^1, v_i^2, \ldots, v_i^{r_0}\}$. When $j \in s(i)$, it indicates that the virtual terminal v_i^j has sense resources. A distributed coalition formation model for terminals in social networks to ensure convergence is built, which adopts three mobile strategies, exiting, joining, and switching, to form a virtual terminal coalition structure.

4.2. VTCF Algorithm Design. In this section, the resource allocation problem is modeled as the formation of the virtual terminal coalition, in which the virtual terminal participates in the resource allocation process. In the overlapping coalition game, the terminal is willing to cooperate to form a coalition for higher return. The virtual terminal can form a nonoverlapped coalition to complete corresponding tasks. However, each terminal has a variety of sensing capabilities, causing different tasks. As a result, it is natural to use overlapping coalitions to form a game to simulate resource allocation of terminals with multiple sense capabilities.

In the OCF game, participants form overlapping coalitions to improve the degree of cooperation and coalition value. To improve the cooperation degree of the system, each cooperative terminal can participate in multiple coalitions by using multiple sense capabilities. In this way, terminals form overlapping coalitions to maximize system utility. The introduction of a single-sense virtual terminal instead of a multisense terminal can effectively reduce the complexity of the overlapping coalition formation process. The coalition formation game for resource allocation is represented by the set of (M, R, U). M is a group of cooperative terminals, and each terminal can be virtualized into multiple virtual terminals. Sense resource owned by all terminals is $R = \{s(1), s(2), \ldots, s(m)\}$. For task allocation, each virtual terminal only participates in one task, and the number of coalitions for each terminal participating is up to $|s(i)|$. In the coalition formation game model, all terminals work together to increase the value of U_{platform}. The game (M, R, U) can be regarded as a VTCF game. The main feature of the game is that each virtual terminal can join a coalition.

As each virtual terminal contributes a kind of sensing resources and can join different coalitions, the set of virtual terminals to form the coalition is described as $C_k = \{v_m^1, \ldots, v_n^p\}$. The supporter of coalition is denoted as $\sup(C_k) = \{v_i^j \in C_k | s_{i,j}^k = 1, i \in M, j \in s(i)\}$. Coalition structure $\prod$ is expressed as $\prod = (C_1, C_2, \ldots, C_{N+1})$. As some terminals prefer to be idle in later tasks, they may not contribute sense resources. C_{N+1} is used to express the virtual terminal coalition that does not participate in any task. The number of coalitions is equal to $N + 1$.

To evaluate the performance of each finished task, the utility function of the virtual terminal is written as

$$U_k = B_k - \sum_{v_i^j \in \sup(C_k)} c(i, j). \tag{13}$$

The total cost of all virtual terminals is expressed as $\sum_{v_i^j \in \sup(C_k)} c(i, j)$. For each coalition structure, the total resource allocation of each terminal i in each coalition does not exceed its own capacity, as shown in the following equation:

$$\sum_{C_k \in \prod} s_{i,j}^k < = s(i), \quad \forall v_i^j \in \sup(C_k), j \in s(i). \tag{14}$$

The cooperation degree obtained by the terminal from different tasks is determined by sensing resources invested in the task and cooperation utility completing the task. Considering coalition value and cooperation degree, the system utility is shown as

$$U_{\text{platform}} = \sum_{C_k \in \prod} (aV(C_k) + \beta \text{Co}(C_k)). \tag{15}$$

It is very vital for the terminal to update a new coalition cooperation degree for maximizing the utility of platform. During the terminal executing the task, the update formula of cooperation degree is given as

$$\text{co}^{(q)}(m, n) = \begin{cases} \text{co}^{(q-1)}(m, n) + \Delta\text{co}^{(q)}(m, n), \\ \text{co}^{(q-1)}(m, n). \end{cases} \tag{16}$$

The variation of cooperation degree between terminals m and n can be expressed as

$$\Delta\text{co}^{(q)}(m, n) = \frac{\text{co}(m)}{\text{degree}(m)} + \frac{\text{co}(n)}{\text{degree}(n)}. \tag{17}$$

$\text{degree}(m)$ is the number of terminals connecting to terminal m in the graph.

From (9), (16), and (17), the update formula of cooperation degree of terminal can be obtained as

$$\text{co}^{(q)}(m, n) = \begin{cases} \text{co}^{(q-1)}(m, n) + \dfrac{\text{Co}(C_k)|g(m)|p_{mk}}{\text{degree}(m)|R_k| \sum_{m \in C_k} p_{mk}} + \dfrac{\text{Co}(C_k)|g(n)|p_{nk}}{\text{degree}(n) \times |R_k| \sum_{n \in C_k} p_{nk}}, & \text{if task complished,} \\ \text{co}^{(q-1)}(m, n), & \text{otherwise.} \end{cases} \tag{18}$$

4.2.1. The Formation of Initial Coalition Structure. Because the terminal is in social network, the theory of graph search can be used to study the generation of initial coalition structure. This section proposes searching for initial coalition members based on SN, as shown in Figure 2. Task initiator a_1 initiates coalition request and negotiates with its directly connected a_2, a_3 firstly to form a coalition. If sense resources contributed by negotiated coalition still cannot satisfy sense resources required by the task, then negotiate with a_2, a_3's acquaintance a_4 to form a coalition. In a large-scale coalition, if the sense cost of existing members is higher than the sense cost of the current member to be negotiated during the negotiation process, the assigned sense resource for the task will be updated, and the process will be repeated until the task requirements are met.

The social network-based coalition initialization method (SN-CI) is described in Algorithm 1. Starting from the task initiator i to search for the virtual terminal set $M(C_k)$ and sense resource set $S(C_k)$ that satisfy sense resource R_k required by task k (Steps 6–12), if the sense

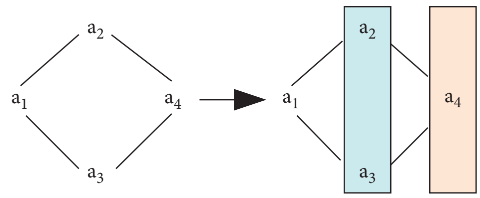


Figure 2: SN-based coalition initialization.

```
(1) Input: G, M, N
(2) Output: coalition structure ∏
(3) ∏ ← ∅
(4) for k ∈ N, i ∈ M do
(5)     M(C_k) ← ∅, S(C_k) ← ∅;
(6)     for j ∈ s(i) do
(7)         if j ∈ R_k and q(i, j) ≥ d(k, j) then
(8)             M(C_k) ← M(C_k) ∪ {v_i^j};
(9)             S(C_k) ← S(C_k) ∪ {j};
(10)            s(i) ← s(i)\{j}.
(11)        end if
(12)    end for
(13)    Q ← n(i);
(14)    while Q ≠ ∅ and |S(C_k)| < |R_k| do
(15)        p ← fst(Q);
(16)        for q ∈ s(p) do
(17)            if ∃j ∈ S(C_k), q == j, q(p, j) ≥ d(k, j) then
(18)                if c(i, j) > c(p, q) then
(19)                    M(C_k) ← M(C_k)\{v_i^j} ∪ {v_p^j};
(20)                    s(p) ← s(p)\{q};
(21)                    s(i) ← s(i) ∪ {j}.
(22)                end if
(23)            else if q ∈ R_k and q(p, q) ≥ d(k, q) then
(24)                M(C_k) ← M(C_k) ∪ {v_p^q};
(25)                S(C_k) ← S(C_k) ∪ {q};
(26)                s(i) ← s(p)\{q}.
(27)            end if
(28)            Q ← Q\{p};
(29)            Q ← append(Q, n(p)).
(30)        end for
(31)    end while
(32)    if |S(C_k)| == |R_k| then
(33)        ∏ ← ∏ ∪ M(C_k).
(34)    end if
(35) end for
(36) return: ∏.
```

Algorithm 1: SN-CI algorithm.

resources of task cannot meet the needs of the task, SN is used to find the neighbors of terminal i to complete the task (Steps 13–31), where $fst(Q)$ means that the first terminal in queue Q is taken out. If sense resource j contributed by terminal i can just be satisfied by neighbor p of terminal i (Step 17), the cost of neighbor p providing resource j is lower than the cost of resource contributed by terminal i. Sense quality requirements are met, and then v_i^j is replaced in the original virtual terminal set $S(C_k)$ with v_p^j (Step 19). Resource j in the resource set $s(p)$ of p can no longer be used in other tasks, so it is deleted from the queue (Step 20). The resource set $s(i)$ that previously removed resource j must readd resource j (Step 21); otherwise, resource q is directly added to $S(C_k)$ (Steps 23–27). append$(Q, n(p))$ means adding p's neighbors to queue Q. The entire social network can be used until all resource requirements of task k are met (Steps 28-29). Finally, a coalition structure containing all virtual terminal coalition sets is obtained.

Different coalition structures may lead to different system utility. To maximize the utility of the system, it is necessary to find the best coalition structure.

4.2.2. Virtual Terminal Joining, Exiting, and Switching. The critical mechanism of coalition formation is to enable virtual terminals to leave or join the coalition according to specified preferences. In particular, each virtual terminal must compare and rank its potential coalitions based on its preferences. Here, the preference order $\succ_i^j$ for any virtual terminal v_i^j is defined below.

Definition 2. (preference order). For any virtual terminal v_i^j, two coalition structures $\prod_p, \prod_q$ for completing tasks are given. For any virtual terminal v_i^j, $\prod_p \succ_i^j \prod_q$ indicates that v_i^j prefers $\prod_p$ to $\prod_q$. Since the preference order depends on the system utility, the preference order for the virtual terminal v_i^j is defined as follows:

$$\prod_p \succ_i^j \prod_q \Leftrightarrow U\left(\prod_p\right) > U\left(\prod_q\right). \tag{19}$$

According to Definition 2, the utility of the coalition structure increases, and the virtual terminal v_i^j will choose this coalition structure. This preference order guarantees a comprehensive increment in the degree of cooperation and platform utility during the formation of the coalition. Using our proposed preference order, the coalition structure can reach stability with a limited number of iterations during its coalition formation.

During the formation of coalition, the virtual terminal decides to join or leave the coalition according to the order of preference. Three-movement rules are defined for possible movement directions of the virtual terminal. Firstly, the virtual terminal exits from the current coalition and enters an idle coalition. Secondly, the virtual terminal leaves an idle coalition and enters any other coalition. Thirdly, the virtual terminal transfers from its existing coalition to a nonidle coalition. Virtual terminals can form different coalition structures with these operations.

Definition 3. (exiting rule). It is assumed that the virtual terminals $v_i^j \in C_m$ and $C_m \in \prod_p$. If $\prod_q \succ_i^j \prod_p$, then v_i^j leaves coalition C_m, and then coalition structure $\prod_p$ becomes the coalition structure $\prod_q$. The new coalition structure is $\prod_q = \left\{\prod_p \backslash \{C_m, C_{t+1}\}\right\} \cup \left\{C_m \backslash \left\{v_i^j\right\}\right\} \cup \left\{C_{t+1} \cup \left\{v_i^j\right\}\right\}$.

According to Definition 3, if system utility of this newly formed coalition structure is better than that of current coalition structure, the virtual terminal v_i^j leaves the current coalition C_m to join the idle coalition C_{t+1}, and $\prod_p$ becomes $\prod_q$. This situation occurs because of virtual terminal v_i^j, which may have a negative coalition effect. Therefore, v_i^j may be more willing to leave coalition C_m to avoid negative benefits. That is, v_i^j is unfriendly, and no other terminal is willing to form a coalition with it.

Definition 4. (joining rule). It is assumed that the virtual terminals $v_i^j \in C_{t+1}$ and $C_n \in \prod_p$. If $\prod_q \succ_i^j \prod_p$ and v_i^j join coalition C_n, the coalition structure $\prod_p$ becomes $\prod_q$, and the new coalition structure is $\prod_q = \left\{\prod_p \backslash \{C_n, C_{t+1}\}\right\} \cup \{C_n \cup \left\{v_i^j\right\}\} \cup \left\{C_{t+1} \backslash \left\{v_i^j\right\}\right\}$.

Based on Definition 4, v_i^j does not initially belong to coalition C_n in the coalition structure $\prod_p$. It is assumed that the virtual terminal v_i^j, which does not participate in the sensing task, leaves the current idle coalition C_{t+1} and joins coalition C_n. The current coalition structure $\prod_p$ becomes a new coalition structure $\prod_q$. If the coalition structure $\prod_q$ is better than $\prod_p$ for virtual terminal v_i^j, according to equation (19), v_i^j joins the coalition and current coalition structure $\prod_p$ is replaced by $\prod_q$. Only when the terminal's actions improve the effectiveness of the coalition structure will the virtual terminal join a new coalition.

Definition 5. (switching rule). It is assumed that the virtual terminal $v_i^j \in C_m$ and $\prod_p$. If $\prod_q \succ_i^j \prod_p$, v_i^j is transferred from coalition C_m to coalition C_n and coalition structure $\prod_p$ becomes $\prod_q$. The new coalition structure is $\prod_q = \left\{\prod_p \backslash \{C_m, C_n\}\right\} \cup \left\{C_m \backslash \left\{v_i^j\right\}\right\} \cup \left\{C_n \cup \left\{v_i^j\right\}\right\}$.

According to Definition 5, when a new coalition structure is better than the current coalition structure, v_i^j is transferred from the coalition where it was to the new coalition. Switching rules balance scale of different coalitions and improve the utilization of resources. During the formation of the coalition structure, one task can be selected by many virtual terminals, and one virtual terminal can select multiple tasks. When the virtual terminal finds that it can improve its system utility by transferring from coalition C_m to C_n, it will execute the switch rules. In this way, C_n can allocate their sense resources to different coalitions autonomously, and the utility of the system will be improved.

Definition 6. If, for any $v_i^j \in M$, there are $v_i^j \in C_m$ and $v_i^j \notin C_n$, where v_i^j will not leave coalition C_m and will not join other coalitions, then the coalition structure $\prod$ of a group of virtual terminals is stable. That is, Nash equilibrium is reached.

According to Definition 6, for a virtual terminal $v_i^j \in C_m$ in a stable coalition structure, it will not leave any coalition and will not join any new coalition. Therefore, all virtual terminals will stay in the current coalition, and no changes will be made.

4.2.3. VTCF Algorithm. To achieve a stable coalition structure, the VTCF algorithm is proposed, as shown in Algorithm 2, which is a distributed algorithm and is executed by each $v_i^j \in M$. In Algorithm 2, $v_i^j \in M$ forms a coalition according to three moving rules. At the beginning of each iteration, $v_i^j \in M$ is randomly selected from feasible coalitions. Assuming that v_i^j leaves current coalition C_m and forms a new coalition structure $\prod_{\text{Quit}}$ (Step 7), judge whether current coalition structure $\prod_{\text{Quit}}$ is a valid coalition structure (Step 8). If so, using v_i^j based on equations (8) and (15), calculate the system utility value of coalition structure $\prod_{\text{Quit}}$ (Step 9). Similarly, the joining rule and the switching rule can generate a potential new coalition structure. The values of coalition structures $\prod_{\text{Join}}$ and $\prod_{\text{Transfer}}$ are calculated by equations (8)) and (15) (Steps 15 and 22), respectively. If the coalition structure produced by the roll-out operation is better than the current coalition structure, the coalition structure is updated (Step 11). Similarly, v_i^j

(1) **Initialization stage:**
(2) Virtual terminal v_i^j forms an initial coalition based on Algorithm 1
(3) **Coalition formation Stage:**
(4) **repeat**
(5) **for** $k \in N$ **do**
(6) v_i^j randomly selects from idle coalition;
(7) $\prod_{\text{Quit}} = \{\prod \backslash \{C_m, C_{t+1}\}\} \cup \{C_m \backslash \{v_i^j\}\} \cup \{C_{t+1} \cup \{v_i^j\}\}$;
(8) **if** $\prod_{\text{Quit}}$ is a ECS **then**
(9) calculates $U(\prod_{\text{Quit}})$;
(10) **if** $\prod_{\text{Quit}} \succ_i^j \prod$ **then**
(11) $\prod = \prod_{\text{Quit}}; U(\prod) = (\prod_{\text{Quit}})$;
(12) **end if**
(13) **end if**
(14) **if** $\prod_{\text{Join}}$ is a ECS **then**
(15) calculates $U(\prod_{\text{Join}})$;
(16) **if** $\prod_{\text{Join}} \succ_i^j \prod$ **then**
(17) $\prod = \prod_{\text{Join}}; U(\Pi) = U(\prod_{\text{Join}})$;
(18) **end if**
(19) **end if**
(20) $\prod_{\text{Transfer}} = \{\prod \backslash \{C_m, C_n\}\} \cup \{C_m \backslash \{v_i^j\}\} \cup \{C_n \cup \{v_i^j\}\}$;
(21) **if** $\prod_{\text{Transfer}}$ is a ECS **then**
(22) calculates $U(\prod_{\text{Transfer}})$;
(23) **if** $\prod_{\text{Transfer}} \succ_i^j \prod$ **then**
(24) $\prod = \prod_{\text{Transfer}}; U(\prod) = U(\prod_{\text{Transfer}})$;
(25) **end if**
(26) **end if**
(27) update cooperation degree according to equation (18);
(28) **end for**
(29) **until** the coalition converges to the final Nash-stable

ALGORITHM 2: VTCF algorithm.

considers joining rules (Step 14) as well as switching rules (Step 21) and updates the system utility value of the coalition structure. When a task is completed, the degree of cooperation is updating between terminals (Step 27) through equation (18). v_i^j repeats the coalition formation process until all v_i^j will not deviate from current coalition or join other new coalitions (Step 29). In other words, the process converges to a stable coalition structure.

Theorem 1. *After a finite number of iterations, the proposed VTCF algorithm converges to a stable coalition structure.*

Proof. Assume that the numbers of terminals M and sense resource types r_0 owned by terminals are limited and there are structural restrictions among terminals. The number of possible coalition structures must be less than 2^{Mr_0}. When the VTCF algorithm is executed, the coalition formation process involves a series of movements of the virtual terminal, which leads to the generation of a series of coalition structures $\{\prod^{(0)}, \prod^{(1)}, \ldots, \prod^{(r)}\}$, where r is the total number of movements of virtual terminal. According to equations (10) and (15), a new coalition structure with higher system utility is formed after each virtual terminal moves. In addition, the number of possible coalition structures is limited. Therefore, r is a finite number. The contradiction proof method is used to prove that if $\prod^{(r)}$ is final coalition structure after last virtual terminal movement, then it must be stable. Assuming that $\prod^{(r)}$ is unstable, according to Definition 6, the existence of $v_i^j \in M$ causes v_i^j to deviate from its current coalition or join a new coalition. Therefore, according to the proposed virtual terminal coalition formation algorithm, v_i^j will join, exit, or transfer operations, and coalition structure $\prod^{(r)}$ will become a new coalition structure. This contradicts the fact that $\prod^{(r)}$ is the final coalition structure. Therefore, $\prod^{(r)}$ is a stable coalition structure. Therefore, after a limited number of iterations, the proposed VTCF algorithm enables v_i^j to achieve a stable coalition structure.

In the coalition formation process of Algorithm 2, each v_i^j seeks to improve value of coalition structure. After each iteration, the movement of v_i^j leads to a new coalition structure $\prod$. Therefore, whenever the coalition structure $\prod$ changes, more excellent system utility is obtained, and v_i^j can adaptively change their cooperation strategy to form a stable coalition structure. □

5. Numerical Simulations

Task allocation using the VTCF algorithm in a social network environment is simulated. The simulated results of the proposed algorithm are compared with coalition forming based on random initialization, SN-CI, and centralized case [27], respectively. In random allocation, users only randomly

Table 1: Simulation parameters.

Parameters	Value
Types of sense resources included in the system: l	10
Terminal contains the types of sense resources: r_0	3
Scale factor: a, β	[0, 1]
Terminal i provides the quality of j resources: $q(i, j)$	[0, 1]
Budget for task k: B_k	[3, 5]
Initial cooperation degree: c	3
Energy coefficient [36]: κ	5×10^{-27}
CPU capability of each terminal: f	1–2.8 (GHz)

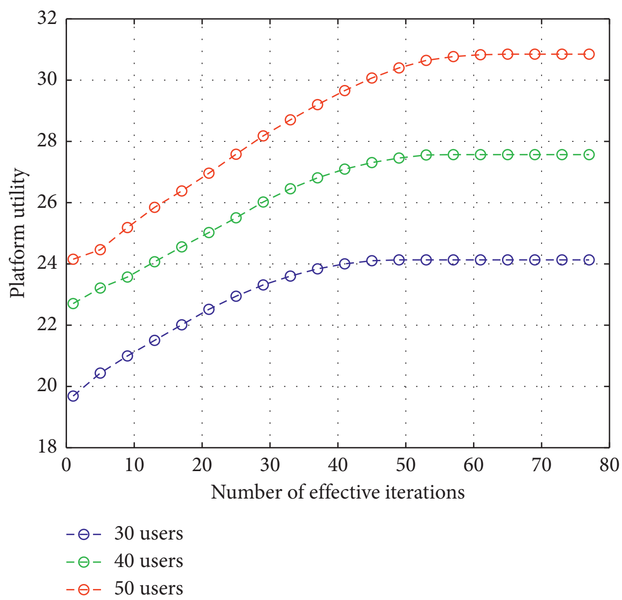


Figure 3: The relationship between total number of effective iterations in the VTCF algorithm and system utility with $N = 10$.

allocate their resources to tasks that they can complete. In the case of SN-CI, the user allocates resources according to the allocation method of Algorithm 1. Finally, the platform utility and the cost of tasks are evaluated in the MCS system. The simulation parameters are shown in Table 1. All curves are generated based on an average of 1000 instances.

In Figure 3, the abscissa is the number of effective iterations, which indicates that a user successfully joins a coalition. It can be seen that the utility of platform becomes stable with the increment of effective iteration numbers based on the VTCF algorithm. Convergence time becomes shorter as the number of users is reduced. The utility value becomes larger when the number of users increases, which indicates that much more users participate in coalition forming. The utility of platform for the VTCF algorithm based on SN and random initialization coalition methods to generate a stable coalition structure are compared in Figure 4. It can be observed that the utility of the platform will increase as the number of iterations increases and tends to be stable. It shows that the initialization coalition method of SN converges faster than random initialization coalition method, and it can accelerate the formation of coalitions.

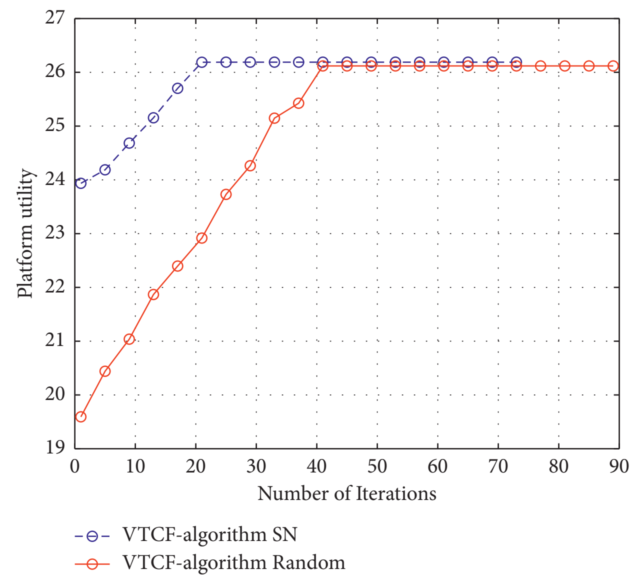


Figure 4: The impact of different coalition initialization algorithms on system utility with $M = 30, N = 10$.

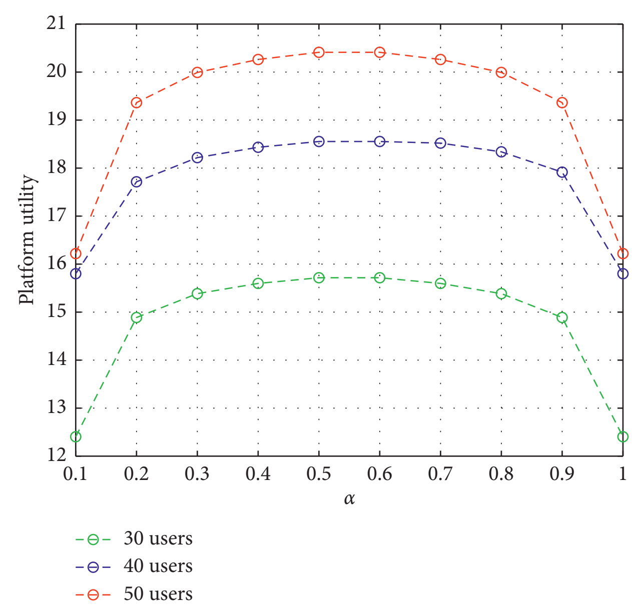


Figure 5: Impact of parameter α on platform utility with $N = 10$.

Figure 5 shows the influence of parameter α on the system utility. The system utility firstly increases with the increment of α and reaches a peak value with $\alpha = 0.4 - 0.6$, which can be regarded as a good combination of cooperation degree and coalition value, and then it decreases as α further increases. Figure 6 shows the effect of the number of tasks on the utility of the platform with $\alpha = 0.5$. By using the VTCF algorithm, the platform utility will increase as the number of tasks increases with $M = 30$. Moreover, the performance of the proposed algorithm is close to that of centralized case and much better than the results of random and SN initialization coalition forming methods.

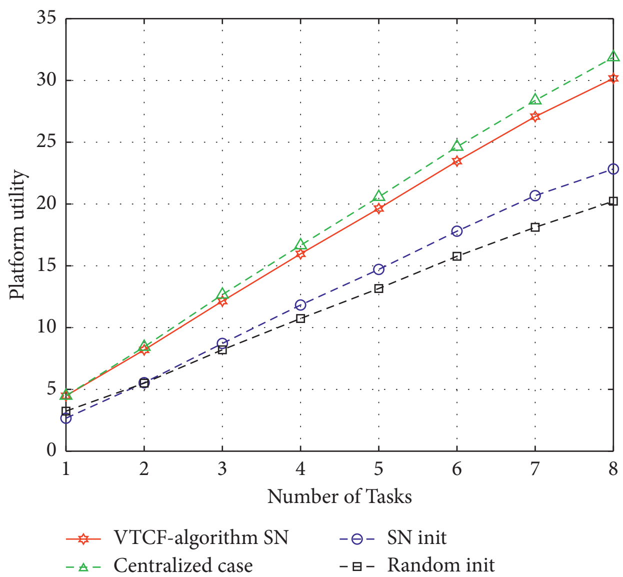


FIGURE 6: Impact of the number of tasks on platform utility with $M = 30$.

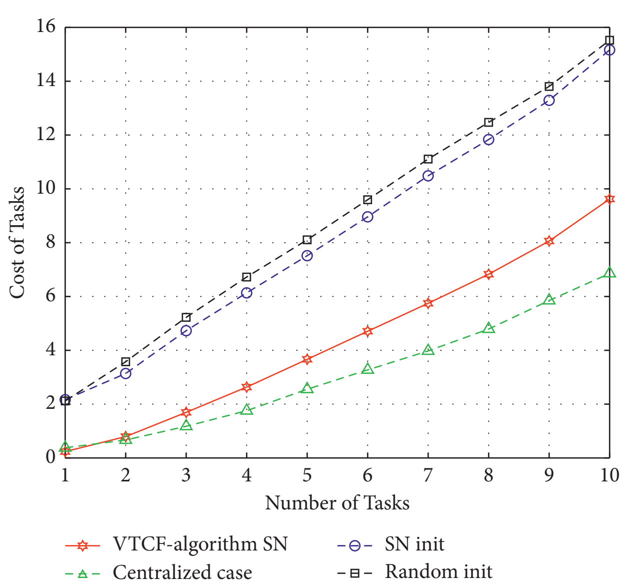


FIGURE 7: Impact of the number of tasks on total cost $M = 30$.

The relationship between the total sense cost and the number of tasks is shown in Figure 7. It can be seen that the whole sense cost of platform will increase with the increment of the number of tasks. At the same time, the cost of tasks by using the VTCF algorithm is close to that of centralized case, which indicates that the proposed algorithm consumes less battery energy and computing time. It is very suitable for the IoT applications. In Figure 8, it can be seen that the utility of the platform increases as the number of users becomes larger. This is explained that much more appropriate users join the coalition to improve the utility of the system as the number of users increases. As the budget of the platform is fixed, the coalition forming in the later period will always

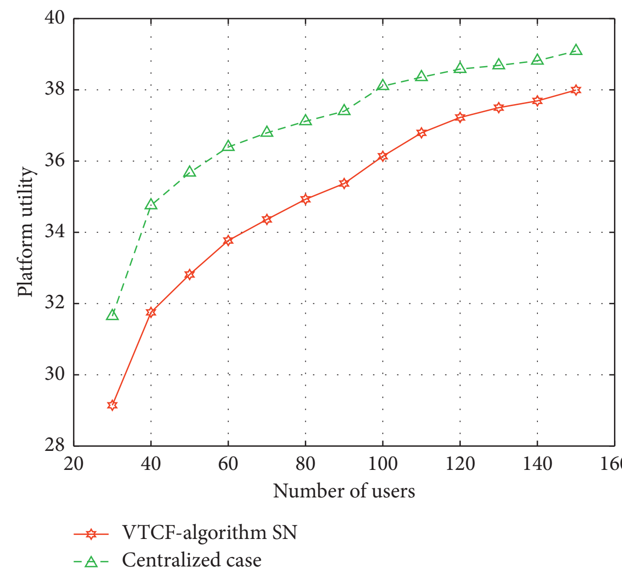


FIGURE 8: Impact of the number of users on platform utility with $N = 10$.

be concentrated among users with high cooperation and low cost; the system utility will enhance slowly as user numbers furtherly increase.

6. Conclusion

In this article, a cooperation method using overlapping coalition formation game is used to describe the assignment of sense tasks in MCS. The SN-CI based on social networks is proposed to accelerate the formation of coalitions. Moreover, the VTCF algorithm by using the concept of virtual terminals and cooperation degree is proposed to simplify the process of coalition formation. The utility of platform is evaluated by using VTCF, SN-CI, and random, centralized case methods, respectively. The simulated results show that SN-CI method is much quicker than random initialization method in original coalition forming. Furtherly, the utility of platform has been evaluated by the proposed VTCF algorithm and compared with that of using random, centralized case methods. Simulated results have verified the performance of VTCF. The concepts and algorithm proposed in this paper can be regarded as a reference in related study of MCS.

Data Availability

The writing material was derived from different journals as provided in the references. A MATLAB tool has been utilized to simulate the concept.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

This research was funded by the National Key Research and Development Program of China under Grant no. 2018YFB2100100.

References

[1] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, "Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system," *Information Sciences*, vol. 479, pp. 567–592, 2019.

[2] P. Dutta, P. M. Aoki, N. Kumar et al., "Common sense: participatory urban sensing using a network of handheld air quality monitors," in *Proceedings of the 7th International Conference on Embedded Networked Sensor Systems (SenSys 2009)*, pp. 349-350, Berkeley, CA, USA, January 2009.

[3] L. Rui, Y. Zhang, P. Zhang, and X. Qiu, "Location-dependent sensing data collection and processing mechanism in vehicular network," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 4, 2018.

[4] T. Aitamurto, "The impact of crowdfunding on journalism," *Journalism Practice*, vol. 5, no. 4, pp. 429–445, 2011.

[5] X. Bao and R. Choudhury, "MoVi: mobile phone based video highlights via collaborative sensing," in *Proceedings of the 8th International Conference on Mobile Systems Applications, and Services (MobiSys 2010)*, pp. 357–370, San Francisco, CA, USA, June 2010.

[6] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[7] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[8] Y. Liu, S. Tang, H.-T. Wu, and X. Zhang, "RTPT: a framework for real-time privacy-preserving truth discovery on crowd-sensed data streams," *Computer Networks*, vol. 148, no. 15, pp. 349–360, 2019.

[9] B. Zhao, S. Tang, X. Liu, and X. Zhang, "PACE: privacy-preserving and quality-aware incentive mechanism for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1924–1939, 2021.

[10] B. Zhao, S. Tang, X. Liu, X. Zhang, and W.-N. Chen, "IronM: privacy-preserving reliability estimation of heterogeneous data for mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5159–5170, 2020.

[11] B. Liu, W. Zhong, J. Xie, and L. Kong, "Deep learning for mobile crowdsourcing techniques, methods, and challenges: a survey," *Mobile Information Systems*, vol. 2021, Article ID 5545470, 11 pages, 2021.

[12] Z. Peng, X. Gui, J. An, R. Gui, and Y. Ji, "TDSRC: a task-distributing system of crowdsourcing based on social relation cognition," *Mobile Information Systems*, vol. 2019, Article ID 7413460, 12 pages, 2019.

[13] D. Jiang, W. Wang, L. Shi, and H. Song, "A compressive sensing-based approach to end-to-end network traffic reconstruction," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 507–519, 2020.

[14] S. Yang, X. Qiu, H. Xie, J. Guan, Y. Liu, and C. Xu, "GDSoC: green dynamic self-optimizing content caching in ICN-based 5G network," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 1, p. 3221, 2017.

[15] F. Wang, D. Jiang, H. Wen, and H. Song, "Adaboost-based security level classification of mobile intelligent terminals," *The Journal of Supercomputing*, vol. 75, no. 11, pp. 7460–7478, 2019.

[16] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[17] G. Ji, Z. Yao, B. Zhang, and C. Li, "A reverse auction-based incentive mechanism for mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8238–8248, 2020.

[18] H. Huang, D. Chen, and Y. Li, "IMLDP: incentive mechanism for mobile crowd-sensing based on local differential privacy," *IEEE Communications Letters*, vol. 25, no. 3, pp. 960–964, 2021.

[19] H. Shi, W. Wang, N. Kwok, and S. Chen, "Game theory for wireless sensor networks: a survey," *Sensors*, vol. 12, no. 7, pp. 9055–9097, 2012.

[20] M. W. Baidas and A. B. MacKenzie, "Altruistic coalition formation in cooperative wireless networks," *IEEE Transactions on Communications*, vol. 61, no. 11, pp. 4678–4689, 2013.

[21] L. Militano, A. Orsino, G. Araniti, M. Nitti, L. Atzori, and A. Iera, "Trust-based and social-aware coalition formation game for multihop data uploading in 5G systems," *Computer Networks*, vol. 111, pp. 141–151, 2016.

[22] J. Wang, F. Wang, Y. Wang, L. Wang, Z. Qiu, and D. Zhang, "Hybrid task allocation in mobile crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 3, pp. 598–611, 2020.

[23] X. Zhang, Z. Yang, Y.-J. Gong, Y. Liu, and S. Tang, "SpatialRecruiter: maximizing sensing coverage in selecting workers for spatial crowdsourcing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5229–5240, 2017.

[24] J. Nie, J. Luo, Z. Xiong, D. Niyato, and P. Wang, "A Stackelberg game approach toward socially-aware incentive mechanisms for mobile crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 724–738, 2018.

[25] J. Zhang, X. Yang, X. Feng, H. Yang, and A. Ren, "A joint constraint incentive mechanism algorithm utilizing coverage and reputation for mobile crowdsensing," *Sensors*, vol. 20, no. 16, p. 4478, 2020.

[26] J. Wang, F. Wang, Y. Wang, D. Zhang, L. Wang, and Z. Qiu, "Social-network-assisted worker recruitment in mobile crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1661–1673, 2019.

[27] B. Di, T. Wang, L. Song, and Z. Han, "Collaborative smartphone sensing using overlapping coalition formation games," *IEEE Transactions on Mobile Computing*, vol. 16, no. 1, pp. 30–43, 2017.

[28] Z. Dai, Z. Wang, and V. W. S. Wong, "An overlapping coalitional game for cooperative spectrum sensing and access in cognitive radio networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 8400–8413, 2016.

[29] W. Saad, Z. Zhu Han, A. Hjorungnes, D. Niyato, and E. Hossain, "Coalition formation games for distributed cooperation among roadside units in vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 48–60, 2011.

[30] O. Revelo Sánchez, C. A. Collazos, and M. A. Redondo, "A strategy based on genetic algorithms for forming optimal collaborative learning groups: an empirical study," *Electronics*, vol. 10, no. 4, p. 463, 2021.

[31] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," *IEEE Transactions on Computers*, vol. 64, no. 8, pp. 2253–2266, 2015.

[32] A. Miettinen and J. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proceedings of the 2nd USENIX Workshop on Hot Topics in Cloud Computing*, pp. 1–7, Boston, MA, USA, June 2010.
[33] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, 2015.
[34] B. Di, T. Wang, L. Song, and Z. Han, "Incentive mechanism for collaborative smartphone sensing using overlapping coalition formation games," in *Proceedings of the 2013 IEEE Global Communications Conference (GLOBECOM)*, pp. 1705–1710, Atlanta, GA, USA, December 2013.
[35] Y. Sun, Q. Wu, J. Wang, Y. Xu, and A. Anpalagan, "VERACITY: overlapping coalition formation-based double auction for heterogeneous demand and spectrum reusability," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 10, pp. 2690–2705, 2016.
[36] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2019.

Hindawi
Mobile Information Systems
Volume 2021, Article ID 2756666, 13 pages
https://doi.org/10.1155/2021/2756666

Research Article

Pushed SOLID: Deploying SOLID in Smartphones

Manuel Jesús-Azabal, Enrique Moguel, Sergio Laso, Juan Manuel Murillo, Jaime Galán-Jiménez, and José García-Alonso

Department of Computer Systems and Telematics Engineering, University of Extremadura, Avda. de la Universidad, S/N. 10003 Cáceres, Spain

Correspondence should be addressed to Manuel Jesús-Azabal; manuel@unex.es

Received 25 May 2021; Revised 15 July 2021; Accepted 6 August 2021; Published 17 August 2021

Academic Editor: Alejandro Fernández

Personal information has become one of the most valuable coins on the Internet. Companies gather a massive amount of data to create rich profiles of their users, trying to understand how they interact with the platform and what are their preferences. However, these profiles do not follow any standard and are usually incomplete in the sense that users provide different subsets of information to distinct platforms. Thus, the quality and quantity of the data vary between applications and tends to inconsistency and duplicity. In this context, the Social Linked Data (SOLID) initiative proposes an alternative to separate the user's information from the platforms which consume it, defining a unique and autonomous source of data. Following this line, this study proposes Pushed SOLID, an architecture that integrates SOLID in the user's smartphone to store and serve their information from a single entity controlled by the users themselves. In this study, we present an implementation of the Pushed SOLID proposal with the aim of experimentally assessing the technical viability of the solution. Satisfactory performance results have been obtained at battery consumption and response time. Furthermore, users have been interviewed about the proposal, and they find the solution attractive and reliable. This solution can improve the way data are stored on the Internet, empowering users to manage their own information and benefiting third party applications with consistent and update profiles.

1. Introduction

Internet applications have become an important part of our daily routines. Every day, thousands of users interact with social networks, sharing new content, consuming posts, and communicating with others. Platforms are concerned on keeping users interested on the services they supply, so that one of the main challenges is providing the appropriate content to the right audience. Successful social webs such as Facebook (https://www.facebook.com), Twitter (https://www.twitter.com), Amazon (https://www.amazon.com), Netflix (https://www.netflix.com), or Spotify (https://www.spotify.com) provide customised services for their users based on their preferences and habits [1]. For this, the storage of personal data becomes critical for applications which benefits from fidelity to enhance the experience and competency [2]. However, these practises are not always favourable both for users and companies. The centralisation and privatisation of the information storage can drive to poor services and risks among other factors.

The privatisation and centralisation of the information is a common practise for applications and companies. Data provide a significant competitive advantage which enables a deeper understanding of the market and users. Services can be improved and adapted to the changing demands of audience which can be analysed and studied to infer these conclusions [3]. Besides, targeted advertisement makes data collection lucrative [4]. However, this privatisation trend drives sites to form information silos, so that data storage is performed independently and autonomously. Hence, the access for other applications is disabled, and users do not enjoy much control over their data. As a consequence, these policies result in potential privacy problems, inconsistencies, duplicity, and insecurity issues.

The lack of any storage standard for information and the unwillingness to share data between platforms have direct consequences on users and companies. Reutilisation of information is not a common mechanism in this kind of application. Users who create an account on a new platform do not find easy by default to port information, friend lists, and interests in new profiles. It is true that many services allow login with Facebook or Google credentials; however, this drives to another point: privacy and dependency. Many of these platforms require accepting terms and conditions which involve targeted publicity, empowering these companies [5]. Therefore, many users place blind confidence in these policies [6], which eventually can result in data leaks [7] and in the use of information for private ends [8–10]. This illusion of control is evident when information and profiles cannot be deleted definitively even when users deactivate their accounts [11]. However, the inconsistencies and duplicity are also significant collateral effects of the privatisation.

Considering most of the applications are reluctant to share their stored information, there is a large set of data duplicated between services [12]. Information about identification, address, contact, and interests are common topics in registration processes. For example, fields such as name, phone, sex, or birthday are required for new profiles in platforms such as LinkedIn (https://www.linkedin.com/signup), Facebook (https://es-es.facebook.com/r.php), or Twitter (https://twitter.com/i/flow/signup). This way, the inputted data are duplicated, resulting in future inconsistencies. A little change in any of these parameters would oblige the user to update the information one by one. As a result, it is common that services recommend content according to past values such as addresses or interests. Likewise, this is typical on multimedia platforms such as music streaming services which recommend content according to previous reproductions. Thus, e.g., if two music streaming platforms are used in unbalanced times, the performance will not be accurate in any of them. Thus, inconsistencies and duplicity affect the user experience, which becomes wasted and results in a bad service for both application and costumer.

As a response for this, there are proposals which attempt to provide alternative models for data storage. Following the basis of data decentralisation, some of the most popular are WebBox [13] or Diaspora [14]; however, Social Linked Data (SOLID) initiative [15] becomes the most relevant among these solutions. SOLID proposes the decentralisation of the information to separate applications and personal data [16]. This way, users store their information in autonomous entities called Personal Online DataStores (PODS). Therefore, applications would adopt a model where they do not store data but request it from PODS. Thus, users are able to control accesses and authorise or deny petitions. Adopting this model provides users and companies an enriched experience based on accuracy, privacy, and control, giving response to information duplication and inconsistencies. This way, PODS becomes useful elements to enhance the experience for users and to improve companies' services.

In the present work, we propose the deployment of SOLID PODS on smartphones. This combination exploits the pervasive and contextual characteristics of these devices to provide a new storage model for personal data. Smartphones are appropriate devices to store data since most of the information involved in the applications processes are obtained from them [17, 18]. They become relevant sources of information, so that they can be easily integrated in the data flows without requiring further changes in applications. Therefore, smartphones are suitable devices to store PODS with the user profile, keeping personal information and relevant data for applications. Thus, external apps request access which the user can authorise or deny. As a result, the user is able to know which applications are consuming information and when, whereas companies benefit from a reliable update and consistent data source.

To present this work, the rest of the study is organized in four main sections. Next, some alternative proposals for data storage are introduced, and the differences with our approach are analysed. Then, our proposal is described in detail, including a reference implementation. Following, an assessment of the prototype is presented to study the acceptation and technical viability of the solution. And finally, last section draws some conclusions about the study.

2. Related Works

As the present work, many approaches have brought solutions for alternative data storage. The management and storage of personal information have elicited multiple works which try to bring transparency and control for the users. In this line, even some big companies manifest concern and corporate responsibility about the data government and propose mechanisms to help users. This is the case of Apple and its tool to enable or disable apps to track the user's activity [19]. This function increases the control over the data and its diffusion. However, the tool does not provide a mechanism to avoid data replication and inconsistencies. For this, there are other works which provide further mechanisms and privacy from external accesses. Some proposals focus on the way the data is shared through a set of entities, while others focus on centralising the storage on just one point. Examples of these can be the HAT project [20], Freenet [21], the DAT Foundation [22], the Threefold Net [23], the ActivityPub [24], the Safe Network [25], or the BBC Databox [26]. However, recently, the SOLID initiative has become one of the most relevant.

Works such as by Paulos [27], Eisenstadt et al. [28], Ramachandran et al. [29], and Mannens et al. [30] take SOLID as a base for implementations which exploit the potential of the tool and develop new functions and mechanisms.

The work developed by Paulos et al. [27] brings a deep usage of SOLID for health purposes. Taking into account the massive amount of health data collected by wearables and training devices, this proposal study an alternative storage architecture based on SOLID. The project investigates about the decentralisation of the records stored by companies such as Fitbit, Apple, or Google into a model based on SOLID deployed in the smartphone. This way, issues such as privacy, consistency, and private management are addressed. However, in contrast with our proposal, the solution stores health data to protect privacy and consistency, while our project provides a tool to centralize information and serve any kind of information to third applications.

In the case of Eisenstadt et al. [28], the work centres on providing a reliable certification method for COVID-19 test results. This way, the project brings an architecture for storing tamper-proof and privacy-preserving certification that allows the identification of people who has submitted to Coronavirus tests. For this purpose, the study makes use of SOLID PODS running on phones, guarantying the privacy of the contained information. Eventhough the project makes use of smartphones to deploy PODS, the purpose centres on providing a validation and certification tool. Thus, the solution does not implement services to store and serve personal information. However, our work offers the mechanism to manage the individual's data as services for third parties.

In the same way, Ramachandran et al. [29] also proposed SOLID as a tool for data verification and confidentiality. Thus, the project connects SOLID PODS and Blockchain to create a decentralized architecture which provides a reliable mechanism for data storage, keeping integrity and privacy.

The project proposed by Mannens et al. [30] presented a model to streamline governmental processes making use of the SOLID PODS. The idea comes up as a response to the large amount of personal information that governments store and how institutions keep multiple copies. Issues such as consistency, access control, and privacy are addressed through the use of PODS, following legal frameworks. Therefore, citizens manage their information storing the data in their private PODS and allowing public institution the access. As a result, citizens get control over their data while responding to store inconsistencies between institutions. However, the solution is not a universal tool for information gathering but centred on bureaucracy and official institutions.

These approaches implement SOLID on smartphones. The solutions equip PODS with new behaviour and possibilities, defining a decentralized architecture based on individual devices. The proposals integrate SOLID for concrete purposes such as certification, validation or, in the case of Paulos [27], storing. However, even though these implementations exploit SOLID, they do not draw a global mechanism for data storage. This way, evaluating the different alternatives, we believe it is interesting to equip smartphones with a real relevance at information management and align their possibilities with SOLID mechanisms. Thus, this proposal places smartphones as the core of data storage to serve a new paradigm where privacy and decentralisation suppose the key element of the information government.

Pushed SOLID provides an information profile which could serve as potential tool for multiple purposes. Projects [17, 31–33] which require a profile to manage, process, and export data can be easily integrated with the architecture. In the same way, solutions [34–36] could find in the present proposal a reliable mechanism for implementation.

In the next section, the details of the proposed work are explained, and the internal mechanisms of the solution are provided.

3. Pushed SOLID

This study proposes an architecture which combines the intimate nature of smartphones and the powerful philosophy of the SOLID initiative. The solution mixes these two lines into a project which situates the user in the centre of the information management. Thus, smartphones become the store of their data, empowering the devices to serve as providers of the information to the external applications which require it. This proposal aims to provide a response to the increasing interest in data governance and presents a solution for inconsistencies and duplicity of information on personal profiles.

3.1. System Definition. The proposed architecture relies on a set of components which collaborate to provide the information to external requests as shown in Figure 1. This scheme explains the complete process to communicate a petition from an external application (ε) to the SOLID PODS (P_{solid}) of the user, executed on its smartphone. This way, three main entities shape the architecture: the SOLID PODS (P_{solid}) and Pusher app (P_{app}) in the smartphone (S) and API Gateway (G) in the server. Additionally, Firebase (F) is used to communicate petitions between the server and the smartphone and the external application (ε) is the one which begins the process. Next, the elements of the architecture are detailed and explained.

The SOLID PODS is the entity in charge of storing the personal information. This element is executed locally in the user's smartphone. It has been designed to be deployed on a server; however, in the current proposal, they have been adapted to be deployed on smartphones. Thus, the PODS stores data locally in the device, independently from any external entity. In the current proof of concept of the solution, users can interact with the platform using the default web interface in the phone. Considering the main goal of this implementation is analysing the viability of the proposal, the interactions can be done using the default front.

Once the SOLID PODS is configured, it is ready to provide information. However, the data store is locally deployed, so it can not be accessed externally, and only the own phone visualises the platform. Nevertheless, we want P_{solid} to provide the requested information to external apps (ε), so that it is required an additional mechanism to perform this. Hence, we turn to the Pusher application (P_{app}) to manage external requests. This application provides communication between the API Gateway (G) and the SOLID Server (P_{solid}) which runs on the smartphone. The API Gateway is required to effectively resolve the petitions and to map the SOLID domain requests with the address of the device. Nevertheless, the adoption of this intermediate layer does not work against the decentralized philosophy since the API Gateway does not store further data than the tuples related to addressing the petitions. In order to perform this communication between the API Gateway and the smartphone, Firebase (F) [37] is used to connect the petitions from G with the device.

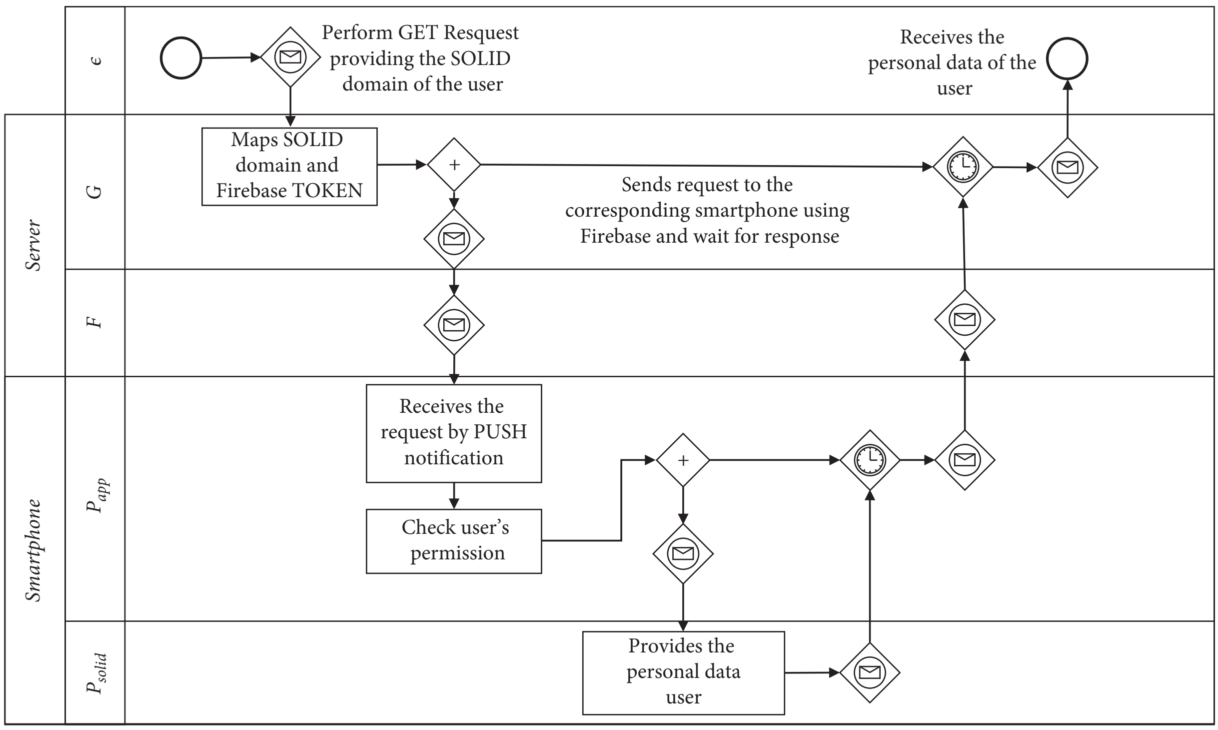


Figure 1: Pushed SOLID architecture.

Firebase is a technology developed by Google which enables a simple push-based communication between entities regardless of the technology used for Internet connectivity (Wi-Fi, 4G, or others). We have considered Firebase as a communication component for its simplicity but can be easily replaced by a similar open-source mechanism like MQTT [38], which would allow information transfer between the layers. Nevertheless, Firebase provides reliable performance and its terms of usage specify how the data stored by the technology are basic information required to operate [39].

Once Firebase communicates the petitions received in the API Gateway, this component communicate the request with the smartphone executing a callback method of Pusher app. Therefore, the method executed in Pusher app asks for the required information to the SOLID PODS, which selects the corresponding data from the profile and returns the values. As a result, Pusher app provides the requested data to the API Gateway. Considering the callback method identifies the requester entity and SOLID PODS counts with access control; security and validation tasks can be easily integrated in the process. As a result, the technical details to replicate the implementation process can be found in the project repository (https://bitbucket.org/spilab/solidsituational.context).

Considering these three components of the system, the way they are coordinated begins from the installation and configuration of the SOLID PODS and the Pusher application. Once these two elements are operative, the Pusher app communicates the addition with the API Gateway, which will be in charge of mapping incoming petitions to the corresponding smartphone. For this, the token ID of Firebase is updated from the Pusher app, in the case it changes. Therefore, the API Gateway keeps updated with the latest value required to operate.

The detailed working of Pushed SOLID is shown in Figure 2, which identifies all steps and collaborations between the different layers. First, the external application (ε) requires a concrete set of information from a user (step 1) (e.g., a new web platform where a user is creating an account). This way, instead of having to input all their personal data, the new member provides its SOLID Domain (2). This variable is the public username of its SOLID PODS and serves as main identification in the architecture. Therefore, the application only have to demand the required data to the API Gateway (G), providing the SOLID domain. The API Gateway is the only entity which truly knows where the information is stored. Thus, this layer uses Firebase technology to easily communicate with smartphones through push notifications. Moreover, the API Gateway maps the SOLID Domain with the Firebase ID and notifies the corresponding smartphone (S) about the petition (3). As a result, the device receives the notification and asks the user for acceptance to allow the external app access to the information (4). This transaction is carried out with the Pusher application (P_{app}) which executes a callback method invoked when the API Gateway receives interactions. Then, the user is asked to allow or deny petitions. This way, the permissions can be configured, defining concrete requester as reliable. This operation can be performed straightly in the SOLID PODS or using the Pusher app. In the case the permission is favourable, the local SOLID PODS (P_{solid}) provides the required information to the Pusher app. Then, the application responds to the API Gateway with the data (5).

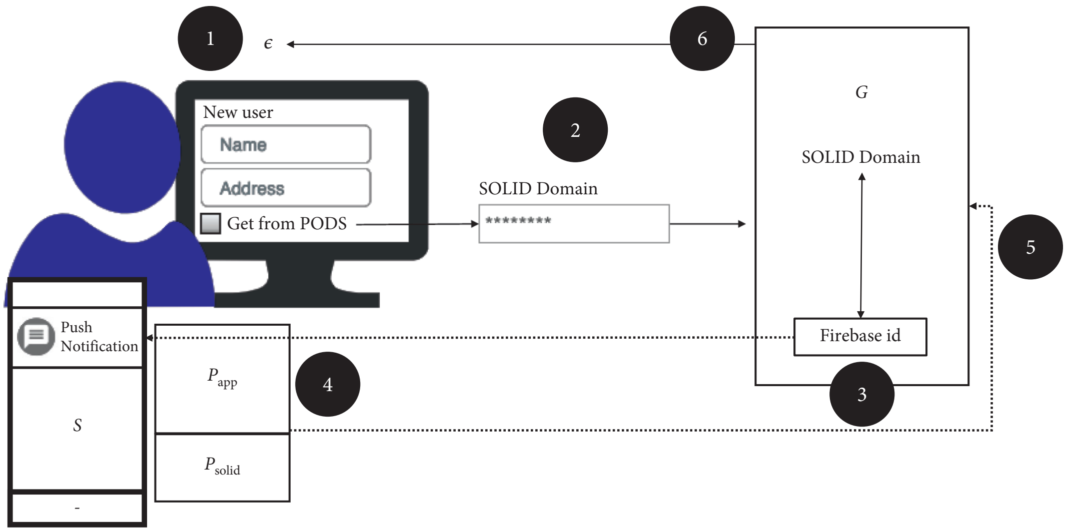


Figure 2: Serving data from a smartphone-based PODS.

Finally, the API Gateway returns the answer to the requesting platform (ε) and the process is completed (6).

Considering this procedure, Pushed SOLID becomes a suitable tool to manage the information both for users and applications. Users are able to keep all their personal information in just one place: their phone, which physically stores the data. Furthermore, it is possible to identify all applications and entities which access the profile. Thus, the user can authorise and revoke petitions at any time. On the other hand, applications benefit from the centralisation of the data in the PODS. This solution guarantees the reliability of the information and that the personal profiles are complete and updated. Besides, inconsistencies and data replication are solved, providing uniformly a common source of data for all applications. As a result, information can be shared between platforms, and it is possible to track the entities which consume our data and the storage becomes standardised. Hence, the way personal information is stored in the Internet could change, bringing a new philosophy based on control and governance for the user.

3.2. Use Case. We have already approached to some of the issues which affect users experience due to the independence of the data between platforms. One of the most recurrent is enjoying a service using multiple platforms, such as music streaming, where users tend to distribute reproductions among several applications [40]. This section compares the ordinary scenario for a user who does not use Pushed SOLID and how the proposal can improve the experience.

For this example, we consider two of the most used music streaming platforms [41]: Amazon Music and Spotify. Both applications count with a recommendation system based on historical reproductions and favourite lists, so the more a user uses a given platform, the more accurate suggestions are got from that platform. However, these applications store the information privately and they do not share it with other services. This way, these politics become inconvenient when users consume more than one streaming platform. Thus, the irregular time spent in each service drives to inaccurate recommendations. Considering this, the proposed use case represents a possible scenario (Figure 3): a user spends 90% of time (t = 90%) listening to music with Spotify ($\varepsilon_{\text{Spotify}}$) and 10% ($t$ = 10%) using Amazon Music ($\varepsilon_{\text{Amazon}}$). As a result, the knowledge about preferences is unbalanced, so that Amazon Music cannot provide accurate recommendations while Spotify lacks on a 10% of the music reproductions. Therefore, this context can highly affect the user experience (ω_1).

As a response for this recurrent situation, Pushed SOLID proposes the unification of the knowledge generated from platforms in the SOLID PODS (ω_2). Thus, the information about historical reproductions and favourite music is stored in a single entity: the SOLID profile (PODS). This way, music streaming platforms read and modify the same music preferences independently from the usage time. As a result, all applications provide recommendations as if they would be used every time. This improvement is reflected on Figure 3 where both Amazon Music and Spotify consume the same information about music preferences.

As a result, Pushed SOLID improves the recommendation performance of the music streaming platforms and provides a better experience, sharing data sources and democratizing the custom experiences possibilities. In the same way, the user is not the only beneficiary since applications will be able to perform more accurate recommendations independently from the time of usage. Thus, Pushed SOLID becomes a very interesting solution for information management, whereas the data governance befalls completely on the user. Considering the proposal uses the smartphone as a data provider, performance and consumption become two main issues for the technical viability. Moreover, next section evaluates the implementation of the architecture and analyses the response of the solution under daily use, checking the technical viability of the solution.

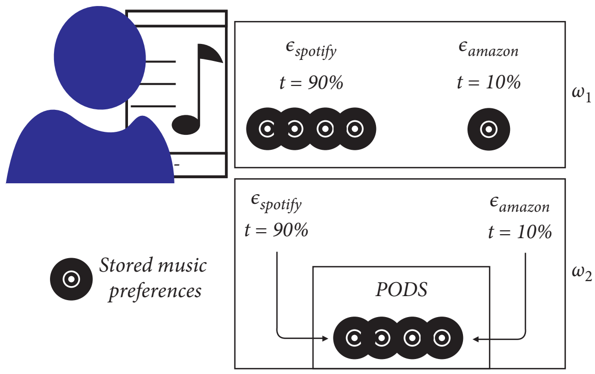


Figure 3: Music streaming with (ω_2) and without (ω_1) situation SOLID.

4. Pushed SOLID Validation

The architecture proposes the use of the smartphone as the centre of all our profiles on the Internet. This means that the device is in charge of managing all incoming data petitions and satisfying them with the corresponding information. Luckily, the technical capabilities of the smartphones have been constantly improving during the last years. However, the high requirements of the proposal suppose a challenge for devices, involving factors such as battery consumption and response time. In addition, it is relevant to consider the possible controversy nature of the project. Taking into account the goal of the platform is managing the large set of personal information distributed along the Internet, user's perception is critical to build a reliable solution. Considering this two points, an experimental evaluation about user's opinion and a system assessments about the performance and technical viability of the proposal has been performed. For this evaluation, the implementation of Pushed SOLID counts with PODS which stores a profile with personal information, historical music played, and main interests (Table 1).

The assessments performed in the experiment followed two dimensions: the perception and usability of the solution and the technical performance of the implementation.

4.1. Usability Tests. The perception and feeling of the users is a critical variable for the proposal. Considering the main goal is managing all personal information, it becomes clearly relevant to study the reaction and opinion from users when they use the prototype. Thus, several steps were considered to perform usability tests. Therefore, a methodology based on user testing [42] was applied to obtain reliable answers. Following this guideline, we were able to elaborate a testing context which allowed us to analyse the perception of the idea and the first opinions after its use. This way, a group of users participated and answered two different surveys.

The first step was to select a group of sixteen users, who are not related with the project, and to introduce them to the purpose of Pushed SOLID. The background of the participants varied, with only six technicians. This way, we try to avoid bias.

Table 1: Personal information stored in a PODS.

Name
Role
Organization name
Address
E-mail
Telephone
Historical music played
Main interests

Then, a first survey about the perception of the idea was done, addressing privacy politics, data centralisation, and opinions, from a neutral point of view. This way, the questions raised clearly the issue, approaching current tendencies and the concept that users have understood about the proposal. The answers were based on ranges: each question (Q) raised a topic and testers had to provide a score (S_p) from $0 \leq S_p \leq 5$, according to the relevance they give to that specific topic. Additionally, a comments box gathered opinions and suggestions. The questions included in the perception survey are given in Table 2. Thus, an average of the obtained results from the survey is shown in Figure 4. Besides, since Q_5 is not a scoring topic but a text parameter, the most recurrent words in the answers are considered. Additionally, a control question was introduced to check if any respondent answered randomly. As a result, all users answered correctly.

Figure 4 shows the obtained scores in the perception survey. The results manifest the interest from users about the proposal and about the phone as the centre of information management. In the same way, testers also showed distrust regarding data management policies in companies (Q_1), especially in those participants who were technicians. For the questions concerning the project concept, users reflected interest for centralisation (Q_2), defining the overall valued dimension. In the same way, testers also think that public would approve and adopt the proposal (Q_3). However, the values define this characteristic as the lowest rated regarding the work. In the case of the main purpose of Pushed SOLID, users think this can become a solution for store issues like replication or inconsistencies (Q_4). Additionally, the answers in Q_5 are analysed considering the recurrence of words. For this, we have considered as common concepts those words which are included in the abstract description of the solution. Therefore, concepts like smartphone (appeared 14 times in responses, considering "phone" as synonym), Internet (13), privacy (11), data (10), and information (6) are ignored. As a result, the words which repeat the most and are not common are centralisation (10), companies (9), security (7), trust (6), change (6), and music (4). Hence, the messages were mainly focused on manifesting the relevance of the idea as a disruptive tool which proposes a great change at data management through centralisation. Besides, the tendency manifests how there is a widespread concern about providing security mechanisms to guarantee a safe storage in the smartphone, defining this as the main core of the project. Also, there were comments which support that companies should take part in the proposal, manifesting the importance

Table 2: User perception survey.

	Q	S_p
Q_1	Do you trust the current politics for storage methods for personal information at big companies?	[0,5]
Q_2	Do you like the idea of keeping all your data in your phone?	[0,5]
Q_3	Do you think current users will embrace this model?	[0,5]
Q_4	Do you think this model solves main management issues? (consistency, duplicity. . .)	[0,5]
Q_5	How would you improve the platform?	[text]

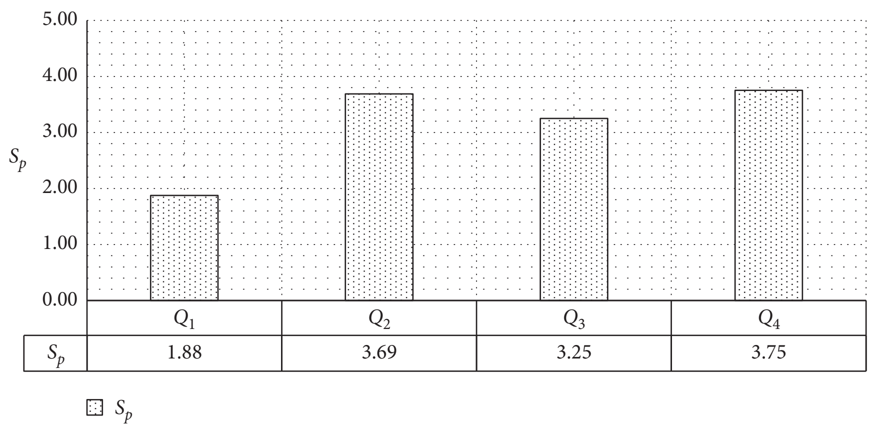


Figure 4: Score (S_p) of the questions (Q_n) about the perception of the project.

of providing a good experience also for enterprises. At last, some contributions were focused on potential applications like music services. Considering the results obtained in the first interaction with users, the survey discloses significant interest in the proposal by the testers.

Once the first approach was finished, testers interacted with the platform. The tasks they performed mainly focused on user experience, so that they created a PODS and inputted information. Besides, they supervise information petitions from external applications. In the same way as the previous step, testers answered a survey about the user experience using the proposal implementation. Hence, testers scored (S_x) from $0 \leq S_x \leq 5$, the satisfaction and user-friendliness of the solution. The questions included in the survey are given in Table 3. As a result, Figure 5 shows the obtained average results.

Figure 5 shows the obtained results from the user experience survey. In average, the scores denote testers have had a successful experience, specially technicians. For Q_5, the average of answers situates the information input with 2.69, around one point below Q_7. The most rated parameter in the survey was Q_6 with 3.69. Hence, these results show a great acceptance for a novel proposal. Testers mainly highlight the simplicity at using the platform, especially for the registration process and configuring the SOLID PODS. Additionally, the suggestions and comments in Q_8 are also interesting. In the same way as previous survey analysis, the opinions in the comments field are addressed with the most recurrent words. Thus, the words which repeat the most were user interface (11 times), easy (10), comfortable (9), battery (8), performance (5), and integration (4). These recurrences manifest some of the most favourable points of the proof of concept. Users talked about the user interface and the way the application is easy to configure. Also, testers place a value on the comfort of checking the information as well as the incoming petitions from external applications. In the same way, performance and battery were two of the most recurrent issues, standing out the importance of providing a good efficiency in the execution. As a result, the implemented solution provides a successful user experience with an easy-to-use interface.

Considering the obtained results in the usability tests, the proposal fulfils the requirements of usability and provides a disruptive solution to manage information. However, it is important to highlight the performance requirements that the prototype meets. For this, the next section assesses the technical behaviour of the solution, evaluating the consumed battery and the response time and drawing a discussion of the results.

4.2. Technical Performance. One of the main dimensions of the project is the technical viability. Considering the architecture proposes the smartphones as the provider for all information required by applications, the solution must guarantee a good performance. This way, parameters such as the energy consumption and response time become critical variables to analyse. In this section, the provided implementation of the Pushed SOLID is assessed in a laboratory context where random petitions are simulated to evaluate battery consumption in the smartphone devices and the time required to obtain a concrete data from the PODS. Next, the setup configuration is explained and the results are discussed.

Table 3: User perception survey.

	Q	S_x
Q_5	Satisfaction at information input in SOLID PODS	[0,5]
Q_6	Simplicity at the registering process	[0,5]
Q_7	Simplicity at specifying SOLID PODS parameters	[0,5]
Q_8	Comments field to suggest changes	[Text]

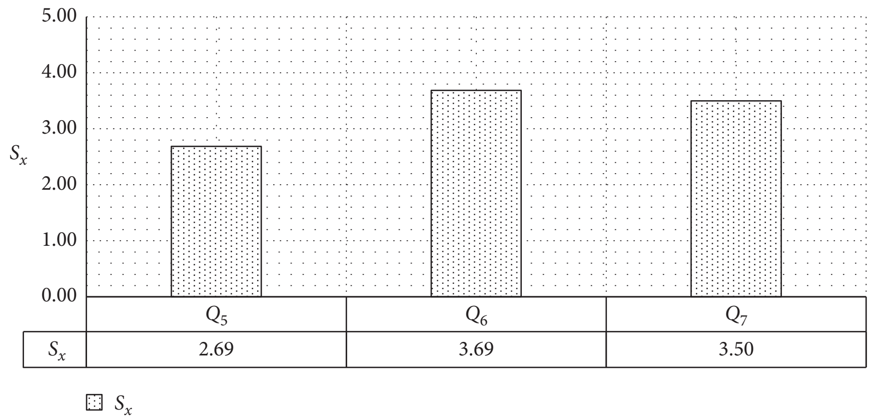


Figure 5: Score (S_x) of the questions (Q_n) about the user experience.

4.2.1. Assessment Setup. In order to evaluate the platform, an assessment setup is built. This scenario is composed by two different contexts: the laboratory and the smartphones. Each of them is in charge of deploying any of the three different entities involved in the architecture: external application which performs petitions (E_{app}), the API Gateway which communicates petitions with smartphones (G), and smartphones which executes the Pusher app (P_{app}) and the SOLID PODS to store the information (P_{solid}). Considering this, an experimental context is built to simulate a heavy charge of petitions to the architecture. Figure 6 represents graphically the process.

The petitions are generated in the laboratory context. It is shaped by two computers, computer $A(C_A)$ and $B(C_B)$ (Table 4). The first one deploys the API Gateway (G), whereas the second one send petitions to it, simulating the activity of an external application (E_{app}). C_B follows Algorithm 1 to request the data to the API Gateway, which communicates with the smartphones. The smartphone context is shaped by two devices: smartphones $A(S_A)$ and $B(S_B)$ (Table 5). Both devices are submitted to a daily use by their owners while executing a SOLID PODS and answering automatically the petitions generated by C_B. As a result, the raised scenario proposes a experimental context where the performance of the platform can be easily tested. Every time C_B performs a petition, the time lapsed between the request and the response is registered. In the same way, both smartphones S_A and S_B monitor the battery consumed by serving the data from the PODS.

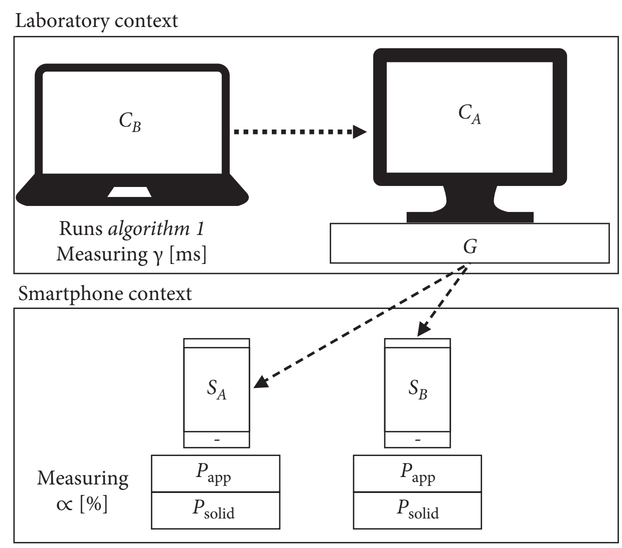


Figure 6: Evaluation setup.

The proposed execution scheme is performed during three days continuously, following the instructions of Algorithm 1. This way, there are two programmed peaks of petitions during the first two days. These events try to check the capability of the platform to provide response to a large set of requests, while analysing the impact on the response time and battery average. The number of petitions performed during an hour in the experiment ranges from zero to twelve. Therefore, the traffic does not follow a clear pattern and changes randomly. This idea corresponds with the goal of providing an irregular and realistic flow of messages. For this, a generated number and a waiting time of five minutes is used. As a result, this behaviour tries to provide a changing number of requests based on the average number of notifications that smartphones use to receive in a day [43]. Once the execution concludes, outcomes are obtained. The next section discusses the results and extracts conclusions.

Table 4: Technical specifications of the computers used in the experiment (C_A and C_B).

Specifications	C_A	C_B
Operative system	Ubuntu 19.10	MacOS Catalina 10.1
RAM memory	8 Gb	16 Gb
Processor	IntelCore i5-3570 (3.4 GHz)	IntelCore i5-7360U (2.3 GHz)

Require: returns false if simulation time is over, $S(\,)$; petition probability, P; petition type, P_t; returns true if it is peak time, P; get the SOLID DOMAIN from smartphone, SOLID (smartphone); generates a number between [0.0, 1.0], R; wait for five minutes, $W(\,)$; returns the type of data to request, $T(\,)$; perform the request of a concrete data to a concrete smartphone, $L(S_n, T)$.

```
do
  P←R
  if P > = 0.5 then
    do
      Pt←T
      L(SOLID(SA), Pt)
      L(SOLID(SB), Pt)
    while P = True
  end if
  W
while S = True
```

Algorithm 1: Pseudocode used to generate random petitions, executed in C_B.

Table 5: Technical specifications of the smartphones used in the experiment (S_A and S_B).

Specifications	S_A	S_B
Model	LG G5 Titan	Xiaomi Mi A2 Lite
Android version	Android 6.0 Marshmallow	Android X
Battery (mAh)	2800	4000
Processor	Snapdragon 820 (2.15 GHz)	Snapdragon 625 (2.0 GHz)

4.2.2. Result Analysis. After three days of executions, the results are obtained. There are two sources of results: computer B (C_B), which has been monitoring the response time, and smartphones A (S_A) and B (S_B), which have monitored the battery level evolution. These two variables are specially relevant to qualify the performance of the platform, since they reflect critical values for the purpose of the solution. This way, the battery consumption and response time are introduced.

On the one hand, the battery consumption (α_s (%) with s as the corresponding smartphone) represents the percentage of battery consumed by the application execution. Smartphones run two main components: the SOLID PODS and the Pusher app. The first one waits for petitions from Pusher app which, at the same time, receives requests from the API Gateway, using Firebase. Therefore, the consumption of energy is an important part of the project. Considering that Internet application would require information to the smartphone, it is important to provide availability in the service. For this, the energy consumed by the architecture must be low in smartphones. In order to obtain realistic results, S_A and S_B have been submitted to daily use by their owner with tasks such as surfing the net, messaging, multimedia, and geopositioning. The experiment is focused on analysing how Pushed SOLID can be embed in the day-to-day routines and address the quality of its integration with the rest of functions. Therefore, we discarded the assessment in a controlled environment, focusing in results obtained from a real context. This way, some inaccuracies are assumed, such as distinguishing between consumptions in WiFi or cellular network. For this task, the android Battery Monitoring Tool [44, 45] has been used. As a result, the obtained values correspond with a realistic context.

On the other hand, the response time (γ (ms)) represents the time elapsed between a petition is done by an external app and the information is received by the entity. This way, this parameter becomes especially relevant since the information must be provided quickly to the multiple demanding applications. Factors such as the CPU usage, the memory available, and the Internet connection can affect the results, but this also enriches the outcomes. Considering the idea is integrating the service with an average use of the smartphone, it is interesting to study the response of the architecture when the device is submitted to a real use. Once the executions are completed, the results are obtained. Thus, Figures 7 and 8 show the outcomes.

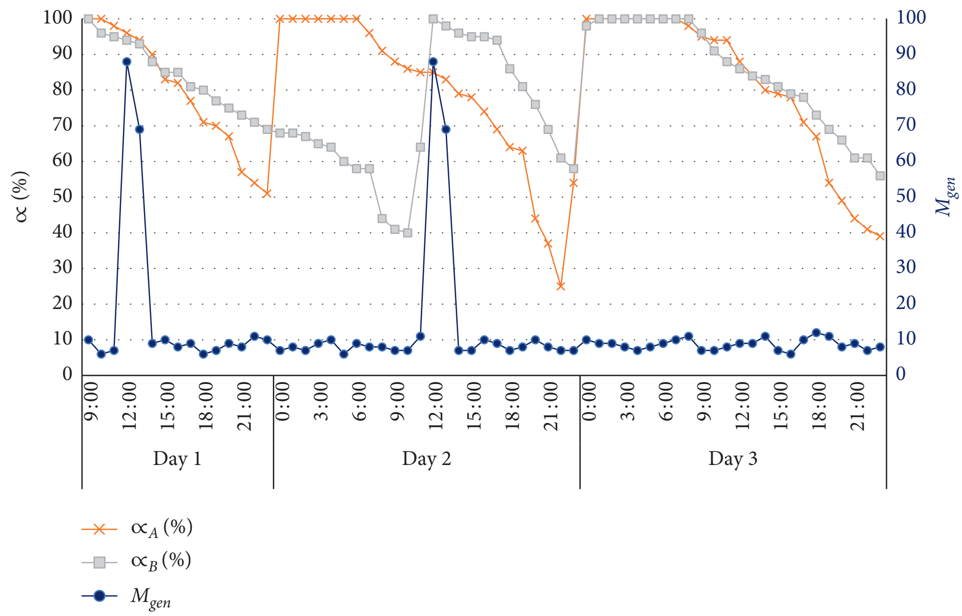


Figure 7: Battery evolution (α) and messages generated (M_{gen}) of smartphones.

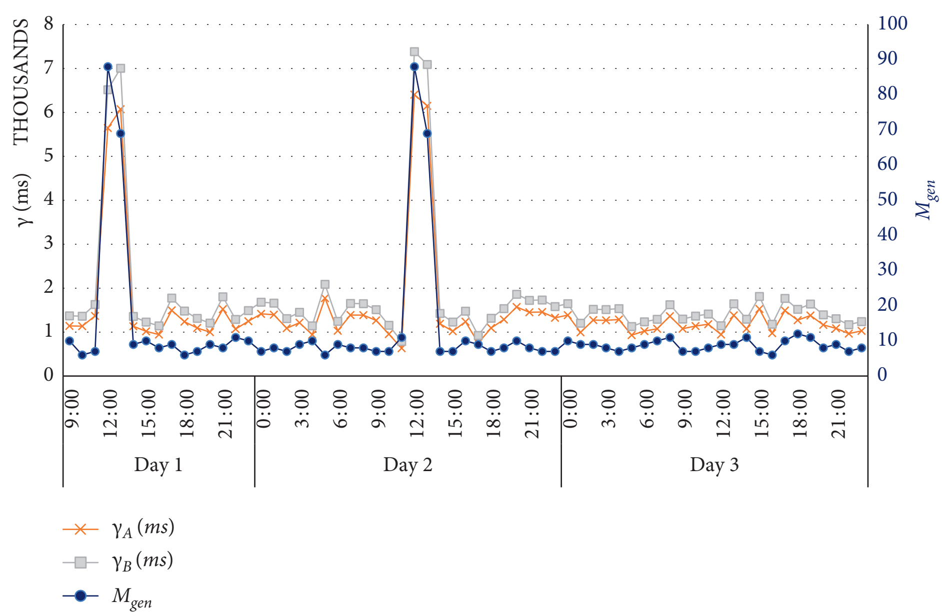


Figure 8: Response time (γ) evolution and messages generated (M_{gen}) in smartphones.

Figure 7 shows the evolution of the battery in S_A and S_B during the three days of executions. The figure represents three different variables as a function of the execution time (T): the battery evolution of both smartphones (α_a and α_b) and the number of generated requests (M_{gen}). As can be seen, there is not a clear connection between M_{gen} and the battery evolution (α_a and α_b). There are two programmed peaks in M_{gen} at $T = 11.00$ on days 1 and 2. These events were preset to find more evident impact on battery. However, as can be seen, the battery is not directly affected by the requests. Therefore, we can deduce that the energy consumption derived from the architecture operation on the smartphones is very low. Additionally, the Android Battery Monitoring Tool indicates consumption below 0% for Pusher app, a report which matches with the battery consumption trajectory.

In the case of γ, Figure 8 shows the recorded times to respond to external applications as a function of the execution time (T). Thus, for this case, we can appreciate a connection between M_{gen} and γ. The average of response time rounds $\gamma = 2152.25$ (ms) with a median number of petitions of $M_{\text{gen}} = 12.87$. The programmed peaks of requests have a direct impact on the time required to solve the petitions. Thus, γ increases from the average value up to more than 7000 (ms), evidencing that the solution is sensitive to a large number of requests. Moreover, in the case of the rest of the petitions, there are some values above one second. The size of the transmitting messages varies, depending if it is a petition or it is a response, between 582 bytes and 612–1024 bytes, respectively. Therefore, it is difficult to identify the size as a bottleneck. Nevertheless, This delay can be introduced by the API Gateway, which performs a waiting time to receive information from the smartphone. As a result, the time required to retrieve information from the smartphone can be optimized. However, considering the main goal of the assessment is analysing the viability of the proposal, the delays can be accepted.

The prototype developed as a proof of the concept of Pushed SOLID provides a valid solution to store our personal data on phones. The approach has been tested to study the technical viability of the project and the results are favourable. Two of the main features the proposal must provide are acceptable energy consumption, which does not affect significantly the battery, and a reasonable response time which guarantee an effective service for external apps. Thus, once the tests have been performed, the results are favourable. On the one hand, the deployment of the SOLID PODS and Pusher app in the smartphones does not have a significant impact on the battery. In fact, the obtained results manifest how low is the energy consumed by the solution. Besides, the petition peaks do not specially affect the battery performance. On the other hand, the response time is sensitive to the simultaneous requests. However, the average time required to provide information is low and satisfies the requirements of the service.

Considering the two dimensions of the assessment, usability and technical performance, we can consider the prototype as a good implementation of the architecture. The implemented solution becomes a reliable method to identify the possibilities of the project and its scope. As summary, the involved testers manifest significant interest for the proposal and the qualitative evaluations show its technical viability. Therefore, the project represents a successful work line with a promising approach to a new data paradigm. In order to define some considerations in the assessment, the next subsection draws the threats to the validation process, identifying some relevant considerations about the results. Furthermore, the last section brings conclusions about the proposal and the advances that the work implements.

4.3. Threats to Validity. In this section, we address the most relevant threats to validity which can condition the performed assessment. For this, two points are approached: tests with users and the smartphone operability.

One of the most relevant stages in the validation of Pushed SOLID is the usability tests. These activities involved a set of research studies to know the concern and opinion of potential users about the proposal. The results manifested a good response from research studies who considered the system as an appropriate alternative option for data storage. However, it must be considered that six of the sixteen users who participated in the experiment were technicians, acquainted with brand new solutions and novel technologies. In the case of other participants, who are ten, they were not technical profiles, so they provide more reliable results for the user testing.

On the other hand, the operability of the smartphone becomes also a threat. The smartphone which executes Pushed SOLID can loose connectivity or its battery can get discharged, so that petitions from external applications can be missed. For this, we rely on the inherent usage patterns of the user and how smartphones use to be always connected and alive. According to studies like [46–48], most users keep their phones with battery during all the day, as well as connected to the Internet. Therefore, we consider the proposal can likely be operative in almost all situations.

5. Conclusion

Personal data have become a very valuable coin for Internet companies. The custom services and advertisement are a relevant issue for enterprises which find publicity as an important source of incomes. As a result, companies keep private their data from other apps, lacking of any standard, and bringing also replication and inconsistencies between platforms. In response to this, the SOLID initiative proposes an alternative storage philosophy which centralises information on uniquely individual entities called PODS. Thus, users keeps all their information while controlling the external accesses to the data. In this framework, we propose Pushed SOLID, an architecture which deploys SOLID PODS on user's smartphone. Thus, the information is stored in the device, which serves as provider of the data to external requests. This way, the user can easily authorise or deny accesses to the PODS.

With the objective of studying the possibilities of the work and the technical viability, a prototype of the proposal was implemented. Thus, a group of testers were interviewed about their opinion of the project, obtaining very good feedback. Users considered the model as a suitable response to the uncontrolled and massive storage of personal data. Besides considering the solution situates the smartphone as the centre of the implementation, it is relevant to guarantee the technical viability. For this, performance assessments were performed to study the impact of the solution on battery consumption and the response time of the requests. As a result, the experimental tests showed a good response of the prototype for both parameters, battery and response time. As a result, the obtained results manifest the relevance of the proposal and the technical viability.

Considering the assessments and evaluations of the proposal, Pushed SOLID can be considered as a valid solution for data storage. The possibility of keeping all personal data on the Internet in just one device enhances the users control over their own information. This way, both users and enterprises benefit from the work. On the one hand, users are empowered with a reliable knowledge about the entities which access to their data, being able to refuse requests while keeping a track on the consumed information. On the other hand, enterprises obtain updated and consistent information while issues like replication are solved. As a result, Pushed SOLID proposes a new paradigm on the Internet which becomes a new way of understanding privacy, while users and enterprises are benefited.

Data Availability

The implementation is available on the repository of the project: https://bitbucket.org/spilab/solidsituational.context. Furthermore, results and obtained data are also included.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the 4IE+ Project (0499-4IE-PLUS-4-E) funded by the Interreg V-A España-Portugal (POCTEP) 2014–2020 program, the project RTI2018-094591-B-I00 (MCI/AEI/FEDER,UE) by the Department of Economy, Science and Digital Agenda of the Government of Extremadura (GR18112 and IB18030), and by the European Regional Development Fund.

References

[1] M. Ruckenstein and J. Granroth, "Algorithms, advertising and the intimacy of surveillance," *Journal of Cultural Economy*, vol. 13, no. 1, pp. 12–24, 2020.

[2] S. C. Matz, J. I. Menges, D. J. Stillwell, and H. A. Schwartz, "Predicting individual-level income from Facebook profiles," *PLoS One*, vol. 14, no. 3, Article ID e0214369, 2019.

[3] A. Esteve, "The business of personal data: Google, Facebook, and privacy issues in the EU and the USA," *International Data Privacy Law*, vol. 7, no. 1, pp. 36–47, 2017.

[4] J. Sheth, "New areas of research in marketing strategy, consumer behavior, and marketing analytics: the future is bright," *Journal of Marketing Theory and Practice*, vol. 29, no. 1, pp. 3–12, 2021.

[5] J. G. Cabañas, Á. Cuevas, A. Arrate, and R. Cuevas, "Does Facebook use sensitive data for advertising purposes?" *Communications of the ACM*, vol. 64, no. 1, pp. 62–69, 2020.

[6] C. Niu, Z. Zheng, F. Wu, S. Tang, X. Gao, and G. Chen, "Unlocking the value of privacy: trading aggregate statistics over private correlated data," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2031–2040, London, UK, August 2018.

[7] D. Lupton, "Thinking with care about personal data profiling: a more-than-human approach," *International Journal of Communication*, vol. 14, no. 19, 2020.

[8] D. Hunter and N. Evans, "Facebook emotional contagion experiment controversy," *Research Ethics*, vol. 12, no. 1, pp. 2-3, 2016.

[9] M. Klenk, "Digital well-being and manipulation online," *Philosophical Studies Series, Ethics of Digital Well-Being*, Springer, Berlin, Germany, 2020.

[10] I. U. Rehman, "Facebook-Cambridge analytica data harvesting: what you need to know," *Library Philosophy and Practice*, vol. 1, no. 1, pp. 1–11, 2019.

[11] J. P. Black, "Facebook and the future of fair housing online," *Oklahoma Law Review*, vol. 72, no. 3, p. 711, 2020.

[12] M. U. Tahir, M. R. Naqvi, S. K. Shahzad, and M. W. Iqbal, "Resolving data de-duplication issues on cloud," in *Proceedings of the 2020 International Conference on Engineering and Emerging Technologies (ICEET)*, pp. 1–5, IEEE, Lahore, Pakistan, February 2020.

[13] M. Van Kleek, D. A. Smith, N. Shadbolt et al., "A decentralized architecture for consolidating personal information ecosystems: the Webbox," 2012.

[14] Diaspora Foundation, 2021, https://diasporafoundation.org/.

[15] The SOLID Foundation, "The solid project," 2021, https://solid.mit.edu.

[16] A. Vlad Sambra, E. Mansour, S. Hawke et al., "Solid: a platform for decentralized social applications based on linked data," MIT CSAIL & Qatar Computing Research Institute, Tech. Rep., 2016.

[17] J. Garcia-Alonso, J. Berrocal, J. M. Murillo, D. Mendes, C. Fonseca, and M. Lopes, "Situational-context for virtually modeling the elderly," in *Proceedings of the International Symposium on Ambient Intelligence*, pp. 298–305, Springer, Toledo, Spain, June 2018.

[18] E. Moguel, J. Berrocal, J. M. Murillo et al., "Enriched elderly virtual profiles by means of a multidimensional integrated assessment platform," *Procedia computer science*, vol. 138, pp. 56–63, 2018.

[19] Apple Inc, "Data tracking by applications," 2021, https://developer.apple.com/app-store/user-privacy-and-data-use/.

[20] I. C. L. Ng, "Can you own your personal data? the hat (hub-of-all-things) data ownership model," 2018.

[21] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: a distributed anonymous information storage and retrieval system," in *Designing Privacy Enhancing Technologies*, pp. 46–66, Springer, Berlin, Germany, 2001.

[22] The DAT Foundation, https://dat.foundation, 2021.

[23] The Threefold Network, https://threefold.io, 2021.

[24] ActivityPub, https://activitypub.rocks/, 2021.

[25] SafeNetwork, https://safenetwork.org/, 2021.

[26] BBC, "BBC box," 2021, https://www.bbc.co.uk/rd/projects/databox.

[27] J. Paulos, *Investigating decentralized management of health and fitness data*, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2020.

[28] M. Eisenstadt, M. Ramachandran, N. Chowdhury, A. Third, and J. Domingue, "COVID-19 antibody test/vaccination certification: there's an app for that," *IEEE Open Journal of Engineering in Medicine and Biology*, vol. 1, pp. 148–155, 2020.

[29] M. Ramachandran, N. Chowdhury, T. Allan, J. Domingue, K. Quick, and M. Bachler, "Towards complete decentralised verification of data with confidentiality: different ways to connect solid pods and blockchain," in *Proceedings of the Companion Proceedings of the Web Conference 2020*, pp. 645–649, Taipei, Taiwan, April 2020.

[30] E. Mannens, R. Verborgh, and T. Berners-Lee, "Streamlining governmental processes by putting citizens in control of their personal data," in *Proceedings of the Electronic Governance and Open Society: Challenges in Eurasia: 6th International Conference, EGOSE 2019*, vol. 1135, p. 346, Springer Nature, St. Petersburg, Russia, November 2019.

[31] M. Jesús-Azabal, J. Rojo, E. Moguel et al., "Voice assistant to remind pharmacologic treatment in elders," in *Proceedings of the International Workshop on Gerontechnology*, pp. 123–133, Springer, Evora, Portugal, October 2019.

[32] E. Moguel, M. Jesus Azabal, D. Flores-Martin, J. Berrocal, J. Garcıa-Alonso, and J. M. Murillo, "Asistente de voz para el recordatorio de tratamiento farmacológico," 2021.

[33] J. Rojo, D. Flores-Martin, J. Garcia-Alonso, J. M. Murillo, and J. Berrocal, "Automating the interactions among iot devices

using neural networks," in *Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 1–6, IEEE, Austin, TX, USA, March 2020.

[34] X. Dong, Y. Guo, F. Li, L. Dong, and A. Khan, "Combination model of heterogeneous data for security measurement," *Journal of Universal Computer Science*, vol. 25, no. 3, pp. 270–281, 2019.

[35] A. Nikiforova, J. Bicevskis, Z. Bicevska, and I. Oditis, "User-oriented approach to data quality evaluation," *Journal of Universal Computer Science*, vol. 26, no. 1, pp. 107–126, 2020.

[36] A. Choi and H. Shin, "Longitudinal healthcare data management platform of healthcare iot devices for personalized services," *Journal of Universal Computer Science*, vol. 24, no. 9, pp. 1153–1169, 2018.

[37] L. Moroney, Moroney, and Anglin, *Definitive Guide to Firebase*, Springer, Berlin, Germany, 2017.

[38] R. A. Light, "Mosquitto: server and client implementation of the MQTT protocol," *The Journal of Open Source Software*, vol. 2, no. 13, p. 265, 2017.

[39] Google Inc, "User agreement firebase," 2021, https://firebase.google.com/support/privacy?hl=es-419.

[40] R. Prey, "Nothing personal: algorithmic individuation on music streaming platforms," *Media, Culture & Society*, vol. 40, no. 7, pp. 1086–1100, 2018.

[41] Statista, "Most used music streaming services," 2021, https://www.statista.com/statistics/798125/most-popular-us-music-streaming-services-ranked-by-audience/.

[42] F. Z. Ghazizadeh and S. Vafadar, "A quantitative evaluation of usability in mobile applications: an empirical study," in *Proceedings of the 2017 International Symposium on Computer Science and Software Engineering Conference (CSSE)*, pp. 1–6, IEEE, Shiraz, Iran, October 2017.

[43] M. Pielot, K. Church, and R. De Oliveira, "An in-situ study of mobile phone notifications," in *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services*, pp. 233–242, Toronto, ON, Canada, September 2014.

[44] S. K. Datta, C. Bonnet, and N. Nikaein, "Android power management: current and future trends," in *Proceedings of the First IEEE Workshop on Enabling Technologies for Smartphone and Internet of Things (ETSIoT)*, pp. 48–53, IEEE, Seoul, South Korea, June 2012.

[45] Google Inc, "Battery monitoring android," 2021, https://developer.android.com/training/monitoring-device-state/battery-monitoring.

[46] D. M. Gezgin, "Understanding patterns for smartphone addiction: age, sleep duration, social network use and fear of missing out," *Cypriot Journal of Educational Sciences*, vol. 13, no. 2, pp. 166–177, 2018.

[47] É. Duke and C. Montag, "Smartphone addiction, daily interruptions and self-reported productivity," *Addictive behaviors reports*, vol. 6, pp. 90–95, 2017.

[48] J. Kuem, S. Ray, P.-F. Hsu, and L. Khansa, "Smartphone addiction and conflict: an incentive-sensitisation perspective of addiction for information systems," *European Journal of Information Systems*, vol. 1, no. 1, pp. 1–22, 2020.