Article

S2NetM: A Semantic Social Network of Things Middleware for Developing Smart and Collaborative IoT-Based Solutions

Antonios Pliatsios *, Dimitrios Lymperis and Christos Goumopoulos *

Information and Communication Systems Engineering Department, University of the Aegean, GR83200 Samos, Greece; dlyberis@gmail.com

* Correspondence: apliatsios@aegean.gr (A.P.); goumop@aegean.gr (C.G.)

Abstract: The Social Internet of Things (SIoT) paradigm combines the benefits of social networks with IoT networks to create more collaborative and efficient systems, offering enhanced scalability, better navigability, flexibility, and dynamic decision making. However, SIoT also presents challenges related to dynamic friendship selection, privacy and security, interoperability, and standardization. To fully unlock the potential of SIoT, it is crucial to establish semantic interoperability between the various entities, applications, and networks that comprise the system. This paper introduces the Semantic Social Network of Things Middleware (S2NetM), which leverages social relationships to enhance semantic interoperability in SIoT systems. The S2NetM employs semantic reasoning and alignment techniques to facilitate the creation of dynamic, context-aware social networks of things that can collaboratively work together and enable new opportunities for IoT-based solutions. The main contributions of this paper are the specification of the S2NetM and the associated ontology, as well as the discussion of a case study demonstrating the effectiveness of the proposed solution.

Keywords: social relationships; ontology; middleware; social network of things; semantic reasoning; interoperability

check for updates

Citation: Pliatsios, A.; Lymperis, D.; Goumopoulos, C. S2NetM: A Semantic Social Network of Things Middleware for Developing Smart and Collaborative IoT-Based Solutions. *Future Internet* **2023**, *15*, 207. https://doi.org/10.3390/fi15060207

Academic Editor: Claude Chaudet

Received: 9 May 2023
Revised: 27 May 2023
Accepted: 1 June 2023
Published: 6 June 2023

1. Introduction

The Internet of Things (IoT) has transformed the way we interact with the world, bringing greater connectivity and intelligence to devices and systems, especially in the context of smart cities [1]. However, traditional IoT networks have limitations in terms of intelligence, context-awareness, and interoperability, which impede their ability to create value for businesses and individuals [2,3]. This is where the Social Internet of Things (SIoT) paradigm comes in; it combines the best of social networks with IoT networks to create a more collaborative and efficient system [4,5].

One of the main benefits of SIoT is its ability to facilitate collaboration and cooperation between devices and applications [6,7]. This results in more streamlined and effective processes, as devices and applications can work together to achieve common goals [5]. Additionally, SIoT offers enhanced scalability, allowing systems to handle large amounts of data and devices, which can be used to create complex, distributed networks [8]. In addition, the adoption and utilization of the SIoT paradigm presents several other benefits. One of these advantages is better navigability in a dynamic network of billions of devices, enabling faster access times among devices [9]. SIoT takes advantage of an enhanced connection to find an optimal navigable route between components, and each node has knowledge of the surrounding nodes and the capability to choose friends to navigate the global system [10].

Another advantage is flexibility, as SIoT is customized to social network behavior to enable friends to search and connect with each other [11]. The node's connection searches faster by accepting the friends' structure, and each node has the necessary information about its friends, neighboring nodes, and the respective concerned friends. This feature eliminates the need for centralized traditional systems, and there is no limitation on the number of nodes that can be connected in an SIoT. Additionally, SIoT provides decision-making models based on the shared level of trustworthiness between friends [12]. This situation improves SIoT reliability because the node interacts exclusively with trustworthy friends, leading to dynamic discovery of services, higher data access, and real-time decision making in more effective and efficient ways [13].

On the other hand, the SIoT presents several challenges that need to be addressed to fully exploit the potential of this paradigm [14]. One of the main challenges is the dynamic nature of social interactions among IoT devices. Relationship selection mechanisms need to be adaptive and capable of adjusting to changes in the social context [10,15]. Static approaches may lead to suboptimal or even ineffective social interactions. Thus, there is a need for dynamic friendship selection mechanisms, such as dynamic ontology-based reasoning, to enable efficient and effective social interactions among IoT devices [9,16].

Another challenge is related to privacy and security in SIoT environments [17,18]. IoT devices may collect sensitive data about users and their social interactions, which could be used for malicious purposes. Therefore, there is a need for robust privacy and security mechanisms that can protect users' data and ensure the trustworthiness of social interactions among IoT devices [19]. One possible solution is to incorporate privacy and security policies into the ontology used to model social interactions [20]. Such policies could govern the access and use of users' data and enable fine-grained control over social interactions.

Moreover, semantic interoperability and standardization are key challenges in SIoT [21–24] environments. The heterogeneity of IoT devices and the lack of standard protocols and interfaces may hinder social interactions among devices from different vendors or platforms. Thus, there is a need for interoperable and standardized solutions that enable seamless social interactions among IoT devices. Ontologies can play a crucial role in this regard by providing a common semantic framework for modeling social interactions [25–29] among IoT devices. Standardization efforts, such as the development of common APIs and protocols, can also facilitate interoperability among IoT devices from different vendors and platforms [30].

Furthermore, the emergence of novel social relationships that surpass traditional human-to-human interactions presents a significant challenge [13]. In the SIoT landscape, interactions involving humans, devices, and hybrid social entities have grown increasingly complex. To fully unlock the potential of the SIoT paradigm, there is a pressing need to develop robust mechanisms that can effectively manage and harmonize these interactions [31]. Failing to address this challenge not only hampers the seamless integration of these entities, but also restricts the realization of the SIoT paradigm's complete potential.

Additionally, efficient service discovery mechanisms are essential to enable seamless integration and resource allocation in heterogeneous SIoT environments [32]. The ability to discover and access services across different entities and applications within the SIoT system is crucial for its smooth operation and to maximize potential benefits. Developing a generic solution adaptable to multiple domains is an imperative challenge to overcome [8,11,33]. A scalable and interoperable solution that can be readily applied to diverse domains within the SIoT ecosystem promotes efficiency, interoperability, and reusability. Finally, rigorous evaluations using representative use case scenarios [13,34] are indispensable to validate the effectiveness and practicality of SIoT solutions.

Our proposed middleware, the Semantic Social Network of Things Middleware (S2NeTM), is specifically designed to tackle critical issues in the SIoT domain. With its innovative features and functionalities, S2NeTM offers comprehensive solutions to overcome obstacles in areas such as semantic interoperability, where S2NeTM leverages advanced ontology and reasoning mechanisms to establish a standardized communication framework and data format. This ensures seamless data exchange and meaningful interaction among diverse IoT devices within social contexts. Additionally, S2NeTM incorporates dynamic relationship selection, trustworthiness management, and efficient service discovery, effectively addressing the evolving needs of SIoT environments. Furthermore, through evaluation using a real use case scenario, our middleware demonstrates its effectiveness, practicality, and real-world applicability.

The main contributions of this paper are as follows:

1. **The specification of the S2NeTM**, integrating a variety of components, including *Context Management*, driving data analysis and context-aware services; *Owner Control*, ensuring access security, privacy, and device ownership; *User Profiling*, fostering personalization; *Service Discovery*, promoting seamless connectivity; *Trustworthiness Management*, ensuring system reliability; *Friendship Selection* and *Relationship Management*, enabling efficient interactions through social connections; and *Semantic Engines*, facilitating semantic data interpretation. Collectively, these components form the foundation of advanced SIoT applications.
2. **The development of an ontology**, providing a standardized vocabulary for describing the relationships between IoT devices, which thereby facilitates the semantic annotation and reasoning of heterogeneous data within the proposed S2NeTM.
3. **An evaluation with a real-world use case**, the "Green Route", demonstrating the effectiveness of the proposed S2NeTM within a smart city environment. The Green Route provides a personalized, eco-friendly route planning service to users. The evaluation demonstrates that S2NeTM enhances the quality of social IoT services, provides personalized recommendations, and significantly improves the user experience.

In order to accomplish our goals, we explore the following research questions concerning the impact of the S2NeTM approach on enhancing semantic interoperability, improving social IoT service quality, and addressing limitations and challenges:

RQ1. How do the S2NeTM components enable social relationships among IoT devices, and facilitate communication and data sharing within a network?

RQ2. What are the benefits of using the S2NeTM in IoT environments, and how does it compare to other middleware solutions in terms of challenges addressed and limitations?

RQ3. How can the S2NeTM be customized and adapted to support specific use cases and applications in different domains?

The remainder of this paper is organized as follows. Section 2 provides a review of the related work in ontology-based social IoT solutions. Section 3 describes the proposed S2NeTM in detail, including its general context and components. It also discusses the S2NeTM ontology, which is used in the semantic annotation and reasoning modules of the proposed solution. Section 4 describes the Green Route use case and demonstrates how the S2NeTM is applied in this use case. Section 5 conveys the results of this study, centered on the research questions that directed the inquiry, and discusses prospective paths for advancing future research in this field. Finally, Section 6 concludes the paper.

2. Related Work

In the context of SIoT, semantics play a crucial role in enabling the interoperability and intelligent decision-making capabilities of devices and systems. Various approaches have been proposed to incorporate semantics into SIoT, with the aim of addressing the challenges of heterogeneity, scalability, and complexity that arise in such environments. These approaches include ontology-based modeling, semantic web technologies, and machine learning algorithms that leverage semantic data. In this section, we review the existing literature on SIoT approaches that utilize semantics, with a focus on their strengths and limitations.

In a related work, the authors propose an architecture and framework for the SIoT called Socialite [35]. They discuss the concept of merging IoT with social networks and propose new relationships for the SIoT. The authors also present use cases for the SIoT and demonstrate how Socialite allows for sharing of information between human users and devices. They introduce semantic models for users, devices, locations, and their relationships, which enable interactions based on rules for relationship management, automation, context generation, social gamification, and common goal management. The authors implemented and demonstrated their concepts by integrating various devices with different functionalities into a single testbed, explicitly representing the relationships between users and devices, and using open-source software.

In another piece of research, a new framework aimed at achieving semantic friendship selection in SIoT networks was introduced [36]. The approach involves the interconnection of smart social objects using a semantic ontology model, and interaction through social network platforms. The main strength of the approach lies in its ability to deliver a satisfying user experience and demonstrate high levels of correctness. However, a significant weakness of the proposed framework is its exponential increase in latency and memory usage when the number of smart objects is increased.

The integration of SIoT and cognitive IoT has been also proposed through smart software agents and semantic web technologies [37]. This work introduced two ontologies to provide semantic meaning to the relationships between smart objects, allowing for complex decision making regarding goal management. The study concluded that the semantic classification of physical object relationships can lead to appropriate deductive reasoning. The use of scalable decision-making and machine learning algorithms can further enhance relationship management (RM). The advantage of this work is that it provides RM capabilities through a cognitive middleware and a set of semantic regulations. However, the lack of real-world scenario evaluations for the proposed techniques is a limitation.

Another related work proposed a socially enabled IoT architecture for industrial applications [38]. The authors introduce an ontology for industrial IoT and discuss the benefits of semantic reasoning and real-time annotation in the proposed architecture. The proposed architecture is accompanied by a practical implementation proposal and an ontology called SIoIT-Ont, which provides semantic meaning to the proposed strategies. The study demonstrated the usefulness of architecture for enabling social interactions, facilitating collaboration and knowledge sharing among industrial IoT devices.

Another study proposed a semantic approach to the SIoT that included Web services and REST principles, aiming to enhance its functionality [39]. This approach also allows for the evolution of social networks into service creation environments, where users can create their own services based on their Web services, devices, and context information. Additionally, the proposed solution enables Web services to be connected to devices, thereby integrating the Web with the IoT. The contribution of SIoT to the field is the provision of a comprehensive and semantic approach that can enrich human life by enabling personalized and context-aware services.

In a related work, researchers examined how the SIoT could potentially revolutionize technology by incorporating a social perspective into existing IoT applications [40]. The paper proposed an architecture for SIoT and outlined a semantic structure to improve comprehension of the concepts and services. The author's objective is to develop a semantic-oriented platform architecture for SIoT and explore its integration into various domains, including industrial manufacturing and agriculture. The goal is to create a more comprehensive understanding of SIoT and establish its potential applications in different fields.

A hybrid recommender system for IoT applications that combines ontology and collaborative filtering techniques to recommend services to users was presented in a relevant work [41]. The proposed approach extends social relationships between users and objects to find alternative sources of missing data. An ontology is used to describe relationships between objects and users, while collaborative filtering is used to predict user preferences. The experimental results showed that the proposed algorithm outperformed existing algorithms in terms of accuracy. Future work includes incorporating spatiotemporal criteria to improve accuracy, and applying privacy preservation techniques to consider the trust and privacy of users.

Finally, a research study proposed a grey wolf algorithm-based user object affiliation mechanism for smarter SIoT [42]. This approach combines object predilection and object sociality to rank objects in SIoT. The proposed approach is a combination of preference, semantic, and context-based approaches, and it has five different models. The paper also discusses a machine learning strategy using the maximum ranked neighborhood approach, which focuses on the delivery of desired services. The proposed model achieved high performance in terms of proper metrics after conducting extensive trials on two real-time SIoT datasets. However, the model lacks a privacy management strategy and only considers an undirected graph, which can be extended in future work. The proposed model can also be used in other scopes of IoT with higher accuracy and certainty.

Table 1 presents an overview of the explored studies focusing on the utilization of semantic technologies for SIoT. It highlights the strengths of these studies, in terms of the challenges they address, as well as their existing limitations.

Table 1. Related work in semantic-based SIoT.

Study	Ref.	Year	Semantic Technologies	Strengths	Limitations
Socialite	[35]	2015	ontology-based modeling	semantic interoperability; new relationships	lack of mechanisms for relationship management, security trust and privacy, and service discovery;
Virtual objects in SIoT	[36]	2018	ontology; semantic web; virtual objects	semantic interoperability	lack of mechanisms for relationship management, security trust and privacy, and service discovery; absence of practical implementation
Cognitive friendship in SIoT	[37]	2017	ontology-based modeling	semantic interoperability; relationship management	limited security trust and privacy; limited evaluation; lack of scalability testing
SIoT for industrial applications	[38]	2019	ontology-based modeling; semantic reasoning	semantic interoperability; relationship management; security trust and privacy; service discovery	implementation and evaluation limited to specific industrial application; lack of scalability testing
Semantic service creation platform for SIoT	[39]	2014	ontology-based modeling	semantic interoperability	lack of relationship management mechanisms; limited security trust and privacy; lack of scalability testing; absence of practical implementation
Semantic-based platform architecture for SIoT	[40]	2019	ontology-based modeling; collaborative filtering	semantic interoperability; service discovery	lack of relationship management mechanisms; limited security trust and privacy; lack of scalability testing; absence of practical implementation
Hybrid recommender system	[41]	2022	semantic web technologies; machine learning algorithms	semantic interoperability; service discovery; evaluation with real use case scenario	lack of relationship management mechanisms; lack of privacy management; lack of scalability testing
Object recommendation-based friendship selection	[42]	2021	semantic web technologies; knowledge graphs	relationship management; dynamic relationship selection; service discovery	lack of semantic interoperability mechanisms; limited evaluation, lack of practical implementation; lack of scalability testing
S2NetM	this work		ontology-based modeling; semantic reasoning; ontology alignment	semantic interoperability; new social relationships; relationship management; dynamic relationship selection; service discovery; security trust and privacy; evaluation with real use case scenario	lack of scalability testing

In conclusion, the reviewed literature on SIoT approaches that utilize semantics highlights the significant progress made in addressing several of the challenges within this field. The use of semantics has enabled SIoT to achieve a higher level of intelligence and sociality, allowing for more efficient and effective interactions between objects and users. However, despite the promising results, there are still some areas that require further research and development. For instance, dynamic friendship selection is a critical area that needs attention to achieve more accurate and reliable recommendations in SIoT networks.

Additionally, privacy management and security issues remain a challenge that needs to be addressed.

S2NetM stands out as a proposed solution that addresses some of the limitations of related works. S2NetM utilizes ontology-based modeling and semantic reasoning techniques, which improve interoperability between heterogeneous systems. One of the strengths of S2NetM is its ability to handle social relationships between users and devices through ontology-based modeling. In addition, S2NetM addresses the limitations of other studies by incorporating privacy preservation techniques and spatiotemporal criteria in its proposed architecture. This can improve the accuracy of recommendations and personalized services provided to users. The S2NetM design, with components distributed across different nodes, enables faster processing and improved scalability. Furthermore, S2NetM has the potential to integrate with edge computing, which can address the limitations of centralized systems by reducing network latency and improving response time.

3. The Semantic Social Network of Things Middleware (S2NeTM)

In this section, we present the proposed S2NeTM that implements the concept of the sociality of things and provides an overview of its architecture. This middleware acts as a bridge between various entities/things in an IoT environment, allowing them to communicate and share information more effectively. We will also discuss the various social relationships that can exist between entities/things in an IoT environment, and how they can be represented using the S2NeTM-associated ontology.

3.1. Foundational Concepts

3.1.1. Sociality of Things

The concept of socialization and collaboration among things is an essential component of the IoT and ubiquitous computing vision [43]. This concept is based on the idea that things can possess a form of social awareness and interact with each other to achieve common goals. To enable such interactions, the development of social networks for things is necessary [44]. These networks allow for communication, collaboration, and knowledge sharing among objects, making the IoT ecosystem more intelligent, efficient, and user-friendly. With socialization and collaboration, things can work together to create complex systems that can solve real-world problems [45].

One of the benefits of socialization and collaboration among things is that it enables them to learn from each other [46,47]. When objects are connected in a social network, they can share knowledge and experiences, which can lead to more efficient and effective decision making [48]. For instance, if an object has previously encountered a problem that another object is facing, it can provide the necessary information to solve the problem quickly. Furthermore, through collaboration, objects can work together to solve problems that they could not otherwise solve individually. This allows for the creation of more sophisticated and intelligent systems that can adapt to changing environmental conditions.

Another advantage of socialization and collaboration among things is that it allows for the creation of more personalized and context-aware systems [32,49,50]. By leveraging social networks, things can better understand their environment and the needs of users. This enables them to provide more relevant and personalized services, such as adjusting the temperature in a room based on the presence of users. Through collaboration, objects can work together to provide more comprehensive services that cater to the needs of users. In this way, socialization and collaboration are critical to creating more intelligent and user-friendly IoT systems that enhance users' quality of life [51].

3.1.2. Social Relationships

Atzori et al. [6,52] proposed various social relationships that can be created among IoT objects, some of which we adopt in the context of S2NeTM. Some of these relationships are static and can usually be determined in advance. Other relationships are dynamic and can be created when the conditions of the relationship are met. These relationships are as follows:

- *Parental Social Relationship*: This is created between homogeneous objects of the same type that were constructed within a close time period by the same manufacturer. This is a static relationship, i.e., created at the beginning, when the node is installed in the SIoT network. It is not implemented in the S2NeTM-associated ontology.
- *Co-Location Social Relationship*: This is created even between heterogeneous objects that are in the same location at a specific time, such as sensors in the same smart city or smart home. In some cases, relationships of this kind are developed between heterogeneous objects that find it difficult to collaborate for a common process. It is a dynamic relationship that changes depending on the location of the objects.
- *Co-Work Social Relationship*: This is created between objects that interact in a common task or process. This is a dynamic relationship that can change over time, depending on the objects involved in the process.
- *Co-Owner Social Relationship*: This is created among heterogeneous objects of the same user (e.g., mobile phones, smart home sensors, etc.). This is a static social relationship. It is not implemented in the S2NeTM-associated ontology.

Dynamic relationships are the ones that interest us, and for this purpose, some static relationships are not considered in the S2NeTM-associated ontology. Additionally, three new social relationships are proposed as follows:

- *Users' Relationship*: This is created between two different users of an application at a specific moment in time. It is a dynamic social relationship, as it changes over time.
- *FOAF Relationship*: This is created between entities that have common friends (i.e., a friend of a friend, or FOAF) and can work towards a common goal. It is a dynamic social relationship, as it changes over time.
- *Co-Semantics Relationship*: This is created between entities with common semantics. It is a dynamic social relationship, as data modeling may change over time (e.g., with an ontology-merging method).

Table 2 summarizes the social relationships managed by the S2NeTM:

Table 2. Social relationships in the S2NeTM.

Social Relationship	Description	Static/Dynamic
Co-Location	Develops between entities that are in proximity at a specific time.	Dynamic
Co-Work	Develops between entities that work together to meet a user's need at a specific point in time.	Dynamic
Co-Semantics	Develops between entities that have common semantics and can work for some common goal.	Dynamic
Users	Develops between two different users of an application at a specific time.	Dynamic
Friend of a Friend (FOAF)	Develops between entities who have mutual friends and can work towards some common goal.	Dynamic

3.2. S2NeTM Architecture

3.2.1. General Context

The SIoT is a new paradigm that enables the integration of social relationships into the IoT ecosystem, creating a more personalized and human-centric environment. The S2NeTM is a solution designed to address the challenges of the SIoT paradigm, such as interoperability and efficient collaboration, by leveraging semantic technologies, dynamic relationship selection, and trustworthiness management.

The S2NeTM is a distributed system, wherein each component is assigned a specific role to provide the necessary functionalities that are crucial for smart applications. The collaborative nature of the components allows them to work in tandem with each other to create a seamless and smooth experience for the end-users. The distributed design of the S2NeTM architecture allows each component to be located on different nodes within the system. Figure 1 offers a high-level perspective of the S2NeTM architecture and its operational environment. S2NeTM serves as a distributed software layer positioned between the *Data Collection Platform* and the *Applications*, facilitating efficient processing and improved scalability.

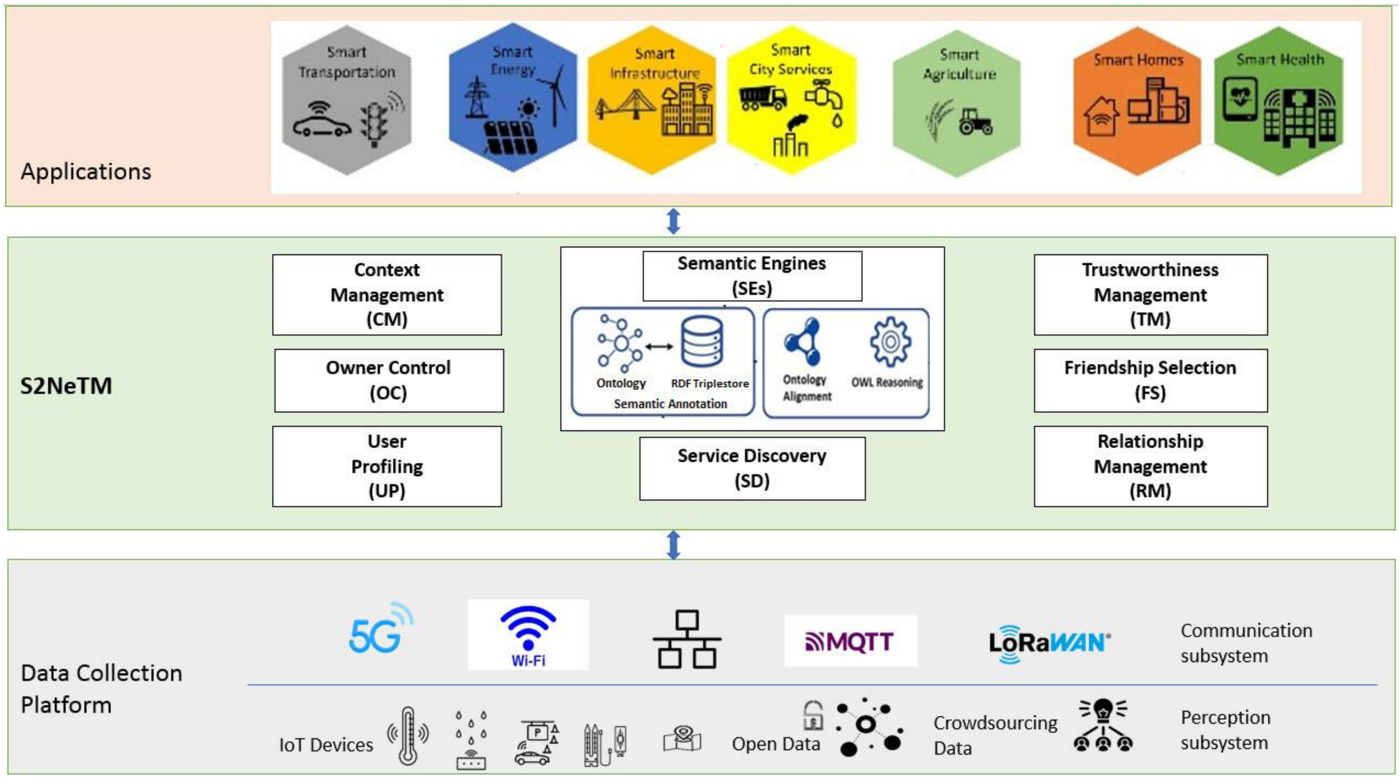


Figure 1. S2NeTM architecture and operational environment.

The *Data Collection Platform* is a crucial part of the operational environment, comprising the *Perception and Communication subsystems*. It involves deploying IoT devices, sensors, and actuators in the physical environment, integrating data from open sources, and crowdsourcing. With a focus on establishing reliable connectivity through diverse communication protocols, it enables reliable, secure, and efficient data transfer for subsequent processing and decision making in the S2NetM.

The S2NeTM provides the necessary components for implementing an SIoT solution, including context management, owner control, profiling, service discovery, trustworthiness management, friendship selection, relationship management, and semantic engines. The semantic engines use an ontology based on OWL2 (Web Ontology Language) to represent information, which allows for the fusion and interpretation of data based on social relationships.

The data are stored in an RDF triple store as a knowledge base, using Neo4j (https://neo4j.com/, accessed on 8 May 2023) graph database management system, and semantic interpretation is based on reasoning techniques and query execution to extract more relevant data about the users and social groups of the objects/entities. Additionally, the S2NeTM includes a semantic annotation module to facilitate the annotation of IoT data with semantic metadata. By leveraging the social relationships between users and devices, the S2NeTM enables the creation of more personalized and context-aware IoT applications. The S2NeTM implements the foundational concepts examined previously in the paper to create social networks among IoT devices and their users, wherein users can interact and collaborate with devices and other users in a more intelligent and personalized way. This is achieved through the use of semantic technologies such as ontologies, which enable the devices to understand and interpret the context and meaning of the data they exchange with other devices and users.

The applications leverage the underlying components of the S2NeTM, serving as the interface through which the collected and processed data are utilized to deliver a wide range of smart services. These applications play a critical role in harnessing the potential of the SIoT paradigm and generating valuable outcomes within smarter ecosystems.

3.2.2. S2NeTM Components

The necessary components for the implementation of the foundational concepts of the S2NeTM are discussed in the following section.

Context Management (CM)

CM is a crucial component of the S2NeTM that collects, processes, and analyzes data from IoT devices and other sources to provide context-aware services. It manages data about users, devices, and their environment, such as location, temperature, humidity, and activity. The CM component works in conjunction with the semantic engines to create a unified view of an SIoT system that includes both the physical and social aspects. It uses this view to interpret and understand the context of the data generated by IoT devices and other sources. By doing so, it can identify patterns and relationships among different pieces of data, enabling the S2NeTM to provide more intelligent and personalized services to the users.

The CM component also facilitates the creation of context-aware applications by providing a mechanism to deliver relevant information to users based on their current context. For example, if a user is in a certain location, the CM component can provide information about nearby events or places of interest. In this way, CM enhances the user experience and enables the S2NeTM to deliver more useful and relevant services.

Moreover, CM enables the S2NeTM to make decisions and take actions based on the current context of the system. For instance, if a user is approaching a smart home, the CM component can trigger the smart home to turn on the lights and adjust the temperature based on the user’s preferences and the current environmental conditions. This functionality is essential for creating an intelligent and responsive system that can adapt to the needs and preferences of the users in real time.

Owner Control (OC)

The OC component is responsible for managing the access control of an SIoT system. It ensures that only authorized users can access and control the IoT devices and their data. This component provides a layer of security and privacy protection for the users.

In an SIoT system, each IoT device has an owner, who has the right to control and manage the device. The OC component manages the access rights of the owners, allowing them to decide who can access their devices and data. This component also allows the owners to revoke access rights at any time.

To implement OC, the S2NeTM uses a decentralized access control mechanism. Each device owner has a set of access control policies that define who can access their devices and data. These policies are stored in the S2NeTM knowledge base and can be updated by the device owners at any time.

The OC component also provides a mechanism for managing device ownership. When a device is added to an SIoT system, the OC component verifies the ownership of the device and assigns the device to the correct owner. If the ownership of a device changes, the OC component updates the ownership information in the system. Thus, the OC component is critical for managing the access control and ownership of the IoT devices and their data. It provides a layer of security and privacy protection for the users, allowing them to control who can access their devices and data. The decentralized access control mechanism and device ownership management features ensure that an SIoT system is secure and trustworthy.

User Profiling (UP)

The UP component is responsible for creating and maintaining user profiles based on their preferences, behavior, and historical interactions with an SIoT system. It uses the data collected from the CM component to build a comprehensive profile of each user. The profile information can include user preferences, past behaviors, interests, and social connections. The user profiles are stored in the knowledge base, allowing for efficient retrieval and utilization of the profile information in various smart applications.

The UP component utilizes machine learning and data-mining techniques to analyze the collected data and extract useful information. This information can be used to make personalized recommendations and to provide more relevant and targeted services to the users. For example, based on a user's past behavior and preferences, the system can suggest a list of recommended devices or services that the user might be interested in.

The UP component also plays a crucial role in managing the privacy and security of an SIoT system. It ensures that users' data are protected and only accessed by authorized parties. It also allows users to control the type of information they share with an SIoT system, and the level of privacy they desire.

In conclusion, the UP component is an essential part of the S2NeTM architecture, as it enables the system to provide more personalized and efficient services to the users while maintaining their privacy and security.

Service Discovery (SD)

The SD component in the S2NeTM is responsible for discovering and locating available services within the SIoT network. This component enables devices and users to discover the services offered by other devices and users in the SIoT network, and to advertise their own services. The SD component allows the S2NeTM to provide automated discovery services, making it easier for devices and users to connect and collaborate.

The SD component uses a variety of protocols and techniques to enable service discovery. These include the simple service discovery protocol (SSDP), the service location protocol (SLP), and the domain name system-service discovery (DNS-SD) protocol. These protocols provide a standard mechanism for service discovery in the SIoT network, ensuring that services can be discovered regardless of the device or platform being used.

In addition to protocol-based service discovery, the S2NeTM also supports semantic-based service discovery. This approach uses the semantic descriptions of services to enable more precise and context-aware service discovery. The service descriptions are based on the S2NeTM ontology, which defines the concepts and relationships related to SIoT services.

The service discovery component also supports dynamic service composition, which enables the automatic composition of services to create new and more complex services. The component uses semantic reasoning techniques to identify compatible services and create new service compositions based on the user's requirements and preferences. This approach enables the S2NeTM to provide more tailored and personalized services to users, based on their specific needs and preferences.

Therefore, the SD component is a critical component of the S2NeTM, enabling the automated discovery, composition, and delivery of services in the SIoT network. It allows devices and users to seamlessly connect and collaborate, providing a more intelligent and interconnected system.

Trustworthiness Management (TM)

TM is a central component of the S2NeTM, which ensures the reliability and security of the system. This component is responsible for managing the trust relationships between devices and users. It assesses the trustworthiness of the devices and users based on their behavior, past interactions, and reputation. This information is used to make informed decisions about the reliability of the devices and users and to take appropriate actions in case of any anomalies or security threats.

The trustworthiness management component uses a combination of trust models and trust evaluation techniques to assess the trustworthiness of devices and users. It applies trust metrics such as reputation, credibility, and reliability to evaluate the behavior of devices and users. It also uses algorithms to detect and mitigate security threats, such as malicious attacks or data breaches.

The TM component ensures that only trusted devices and users are allowed to access an SIoT system and its services. It also monitors the behavior of devices and users to detect any suspicious activity, and takes appropriate measures to prevent security breaches.

Moreover, the TM component provides mechanisms for users to report any issues or incidents related to an SIoT system. It ensures that the reported incidents are appropriately addressed and resolved in a timely and efficient manner, thus increasing the overall trust and reliability of the system.

Friendship Selection (FS)

The FS component is responsible for selecting and recommending potential friends for the devices or users based on their services, interests, location, and other social attributes. This component uses the data collected from the devices and the users to identify potential friends and their interests. Once potential friends are identified, the component uses the relationship management component to manage the social relationships between users and devices.

Relationship Management (RM)

The RM component is responsible for managing the social relationships between users and devices. This component uses social network analysis techniques to analyze the data collected from the devices and the users, and identifies the relationships between them. These relationships can take various forms, such as *co-location*, *co-work*, *co-semantics*, *Users* and *FOAF*. By leveraging these relationships, the RM component can provide personalized services to the users and devices, resulting in more intelligent and efficient interactions. The FS and RM components play a crucial role in creating and managing these relationships, thereby enhancing the overall functionality of an SIoT system.

Semantic Engines (SEs)

SEs represent another vital component of the S2NeTM architecture, responsible for interpreting and processing the semantic data generated by the IoT devices and users. In particular, SEs provide support for reasoning and inference mechanisms, which are essential for making intelligent decisions and deriving new knowledge from the existing data. One of the key aspects of the S2NeTM is the development of a new interoperable ontology, which is used to represent the semantic data generated by IoT devices and users. The S2NeTM ontology is based on OWL2 and extends existing ontologies, such as SSN, IoT-Lite, and M3-Lite. The ontology includes new concepts and relationships that are specific to the social aspect of an SIoT system.

Another important aspect of SEs is the semantic annotation module. This module is responsible for enriching the data generated by the IoT devices with semantic metadata, making it easier for other devices and systems to understand and process. The module uses algorithmic techniques to extract semantic information from unstructured data. The extracted metadata are then mapped to the ontology and stored in the Neo4j knowledge graph (i.e., a type of knowledge base that represents knowledge in the form of interconnected entities and relationships), allowing for easy access and retrieval of relevant information. The semantic annotation module is an essential part of the S2NeTM, as it enables the creation of a more intelligent and interconnected system by providing a common understanding of the data exchanged between devices and users.

Ontology alignment is a mechanism included in SEs, which enables the integration and merging of different ontologies. This is necessary when data from different sources need to be combined and interpreted. Ontology alignment is achieved by mapping the concepts and relationships in different ontologies, and identifying their similarities and differences. The following general steps of the alignment process are used in S2NetM for determining the relationship between two or more ontologies (Figure 2):

1. The two ontologies are given as input.
2. Ontology alignment techniques, such as lexical, word matching and semantic similarity methods, are used in conjunction with methods such as weighted averaging to compute the similarity of classes, object properties, and data properties in the input ontologies.
3. Having obtained the similarity value between the entities, those entities with a value greater than a predefined threshold (e.g., threshold = 0.5) are selected. If the similarity value of the matching is equal to or greater than the selected threshold, then the entities of the input ontologies are considered related; otherwise, the entities are considered unrelated.
4. Finally, the ontology alignment process is completed by refining and adjusting the mappings between the entities of the two ontologies.

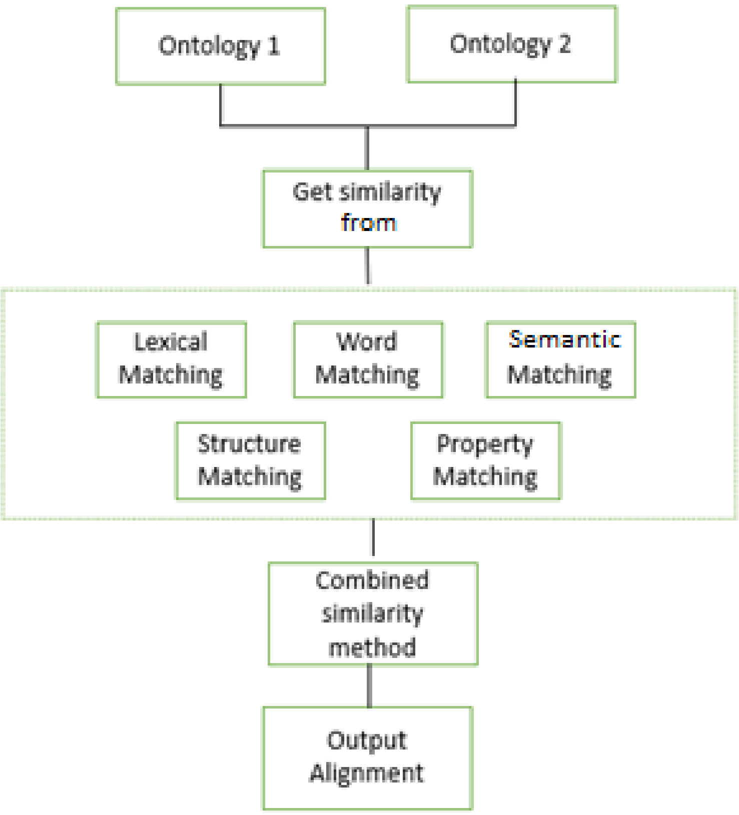


Figure 2. Ontology alignment process in S2NetM.

Reasoning is another critical mechanism provided by the SEs component, which allows an SIoT system to derive new knowledge from the existing data. Reasoning involves applying logical rules and inference mechanisms to the data to infer new relationships and concepts. For example, if an SIoT system knows that two users are friends and that they both use a particular IoT device, it can infer that they are likely to use the device together.

To support these functions, the SEs component requires a semantic repository to store and manage the ontology and data in the S2NeTM. In this case, a triple store knowledge graph, in particular the Neo4j, is used to store semantic data. Neo4j is a graph database that allows for efficient and flexible querying of the data, making it an ideal choice for a semantic repository. The SEs also provide support for querying and retrieving data from the S2NeTM using SPARQL or CYPHER, which are semantic query languages for RDF data. These queries can be used to extract data based on specific criteria, such as device type, location, user preferences or social relationships.

3.3. S2NeTM Ontology

The S2NeTM ontology is a lightweight and interoperable ontology that represents the relationships between heterogeneous entities in IoT environments. The scope of the ontology includes IoT devices, open data entities, and crowdsensing entities, along with their social relationships. The ontology is designed to facilitate the creation of more personalized and context-aware IoT applications by leveraging the social relationships between users and devices.

The S2NeTM ontology is composed of several main classes, including *Entity*, *User*, *Service*, *Service Property*, *Location* and *Platform*. The *Entity* class represents the different types of entities in the IoT environment, such as devices and open data entities. The *User* class represents the human users of the system, while the *Service* class represents the various services offered by the system. The *Service Property* class represents the properties of the services, such as their quality and reliability. Finally, the *Platform* class represents the hardware and software infrastructure of the system.

To represent the relationships between entities, the ontology uses the *hasRelationshipWith* object property. This property has several sub-properties, including *co-work*, *co-location*, and *co-semantics*. The *co-work* sub-property represents entities that work together towards a common goal, while the *co-location* sub-property represents entities that are physically located in the same place. The *co-semantics* sub-property represents entities that share a similar meaning or context.

To ensure interoperability with other ontologies, the S2NeTM ontology reuses several widely adopted ontologies, such as friend of a friend (FOAF), semantic sensor network (SSN), and IoT-lite. FOAF is used to represent the social relationships between users, while SSN is used to represent sensors and their observations. IoT-lite is used to represent the properties of IoT devices and the relationships between devices.

Figure 3 provides a visual representation of the structure of the S2NeTM ontology. This diagram shows the different classes and object properties of the ontology, as well as how they relate to each other. By using the S2NeTM ontology, developers can create more personalized and context-aware IoT applications that leverage the social relationships between users and devices.

As shown in Figure 4, the S2NeTM ontology defines an object property called *need* to represent the relationship between a *User* and a *Service* within the smart environment. This property is crucial in capturing the dependencies between users and services, allowing for better management and optimization of the smart environment. The ontology also introduces new classes and properties to represent sensors that are associated with entities in the smart environment.

The *Sensor* class from the SSN ontology is reused to represent sensors in S2NeTM, providing a standardized way of representing sensor data and enabling interoperability between different sensor networks. The *need* property is used to represent the requirement of a user for a particular service. For example, a user might require a temperature control service to operate efficiently in a smart home. By representing this requirement as a relationship within S2NeTM, it becomes possible to identify which services are essential for each user, and to manage them more effectively. This can lead to improved energy efficiency, cost savings, and a more personalized user experience.

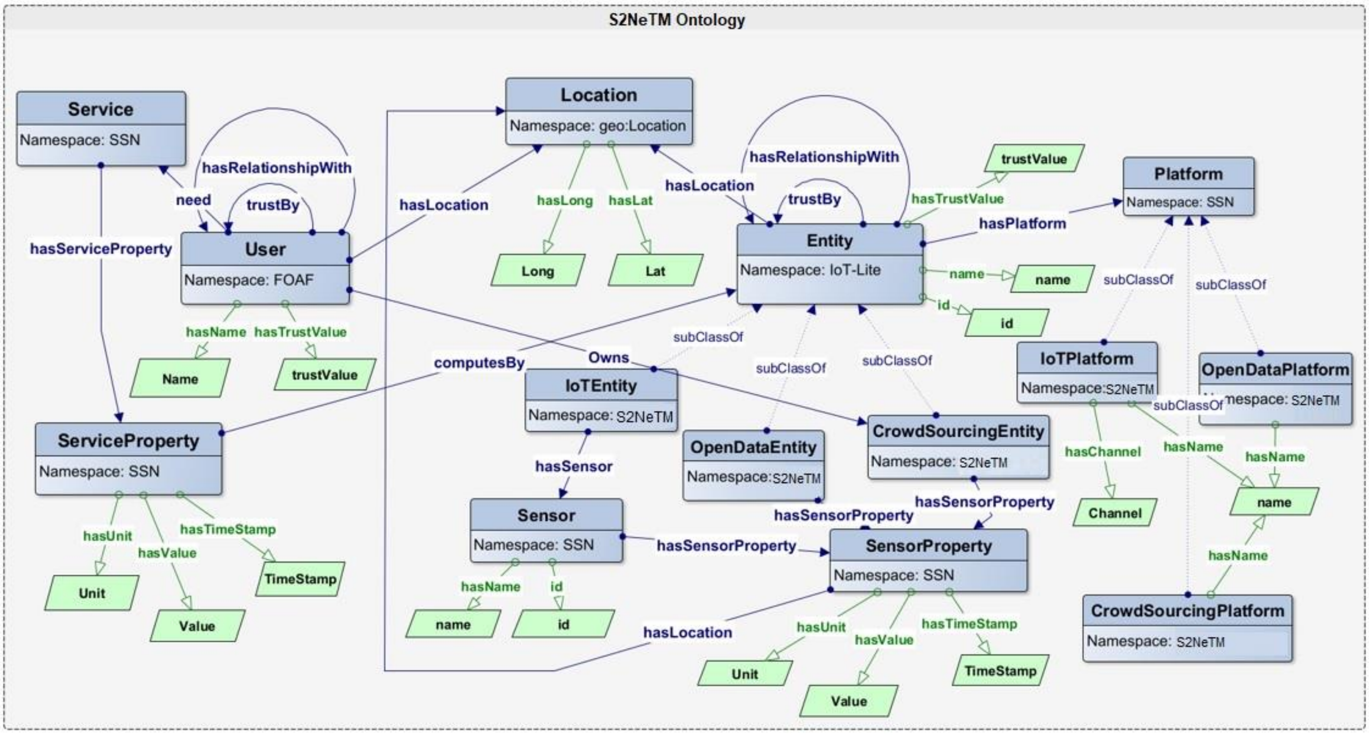


Figure 3. S2NeTM ontology.

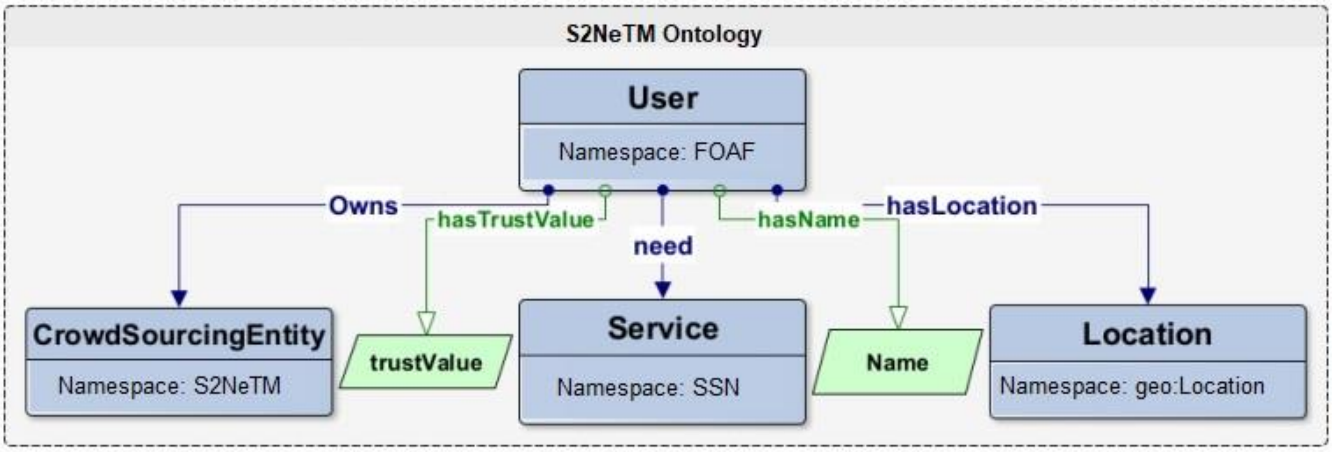


Figure 4. Representation of the user within the S2NeTM ontology.

Entities within the S2NeTM can establish social relationships with one another, which are represented using the *hasRelationshipWith* object property. In addition to the *User* relationships provided by the FOAF ontology, S2NeTM also defines its own types of social relationships. These types of social relationships include *co-work*, *co-location*, and *co-semantics*, which are also considered sub-properties of *hasRelationshipWith*. By using these properties, the S2NeTM ontology can provide a more detailed representation of social relationships between entities within an IoT environment. The ontology diagram in Figure 5 illustrates the various object properties that can be used to represent social relationships between entities.

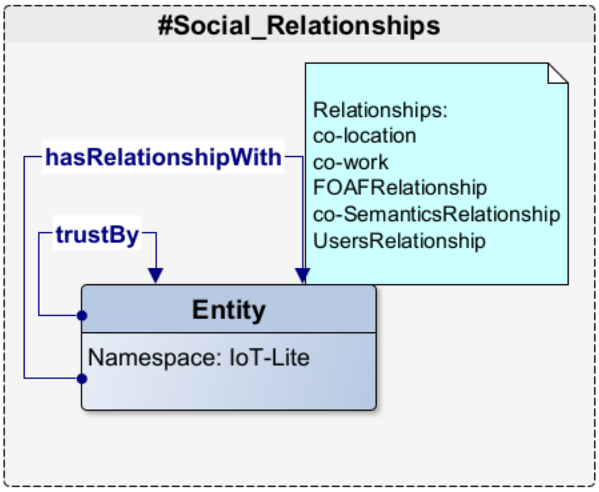


Figure 5. Representation of social relationships in the S2NeTM ontology.

The *trustBy* object property, as illustrated in Figure 5, can be used to represent the trust relationship between two entities in a smart environment. For example, consider a scenario wherein a smart parking system is deployed in a city. The system consists of multiple components, including sensors, cameras, and a backend server. The sensors are responsible for detecting the availability of parking spots, while the cameras are used to monitor the parking lot. The backend server processes the data from the sensors and cameras and provides real-time parking information to drivers. In this scenario, we can define the "*trustBy*" object property to represent the trust relationship between the components of the system. For instance, the sensors can trust the cameras to provide accurate parking data, and the backend server can trust the sensors and cameras to provide reliable information. We can represent this trust relationship using the "*trustBy*" object property, where the domain is the entity that trusts, and the range is the entity that is trusted. For example, we can define "Sensor1 *trustBy* Camera1" to indicate that Sensor1 trusts Camera1 to provide accurate parking data. Similarly, we can define "BackendServer *trustBy* Sensor1" to indicate that the BackendServer trusts Sensor1 to provide reliable information. By representing trust relationships between entities, we can ensure that IoT systems are secure and privacy-preserving.

The S2NeTM ontology provides a comprehensive model for the integration of semantic information derived from heterogeneous entities and objects within an IoT ecosystem. The ontology's main purpose is to represent entities in IoT environments, including devices, open data entities, and crowdsourcing entities, along with their social relationships. By using the semantic description of IoT data, open data, and crowdsourcing data, S2NeTM enables the creation of more personalized and context-aware IoT applications. Additionally, the S2NeTM ontology is designed to be interoperable, which means that it reuses other interoperable ontologies such as FOAF, SSN, and IoT-lite. This interoperability ensures that S2NeTM can be used in different domains, such as smart cities, smart homes, and smart health. Ultimately, the S2NeTM ontology plays a critical role in the advancement of IoT-based solutions by providing a comprehensive model for integrating and representing heterogeneous entities and objects, thereby enabling better communication and collaboration, and achieving common goals.

4. Use Case Evaluation

Evaluation of the S2NeTM in a use case scenario is crucial to identify its practical feasibility and effectiveness in achieving its intended goals. The Green Route use case scenario provides an evaluation of an S2NeTM that aims to provide a sustainable commuting solution by enabling users to find the greenest and most efficient route to their destination. The evaluation considers the effectiveness of S2NeTM in providing context-aware and personalized recommendations for routes that meet the user's sustainability criteria while ensuring an efficient travel time. This section will provide a detailed overview of the Green Route use case, and assess the efficacy of S2NeTM in fulfilling its objectives.

4.1. Green Route: A Use Case Scenario to Find a Sustainable Path

The Green Route is a use case that aims to find the most environmentally friendly route for a citizen named Maria to walk from her house to the port in a smart city. The scenario involves multiple components of the S2NeTM architecture, including *Context Management (CM)*, *Semantic Engines (SEs)*, *Trustworthiness Management (TM)*, *Friendship Selection (FS)*, and *Relationship Management (RM)*. The entities involved in the scenario include IoT environmental devices with sensors for temperature, humidity, and air quality, a sensor web service for accessing open environmental data, and a mobile crowdsourcing application for citizen participation.

The *Data Collection Platform* is responsible for connecting the IoT devices and collecting data from them. In the Green Route scenario, environmental sensors are placed throughout the city, and their data are collected by a sensor web service. The data are then made available for access by other entities through APIs. For example, the air quality sensors can detect pollution levels in different areas of the city and send this information to the *CM* component of the S2NeTM, which then determines the *co-location* relationships between the devices. The *co-location* relationship indicates that the sensors are physically close to each other and can be used to find a green route that avoids areas with high pollution levels.

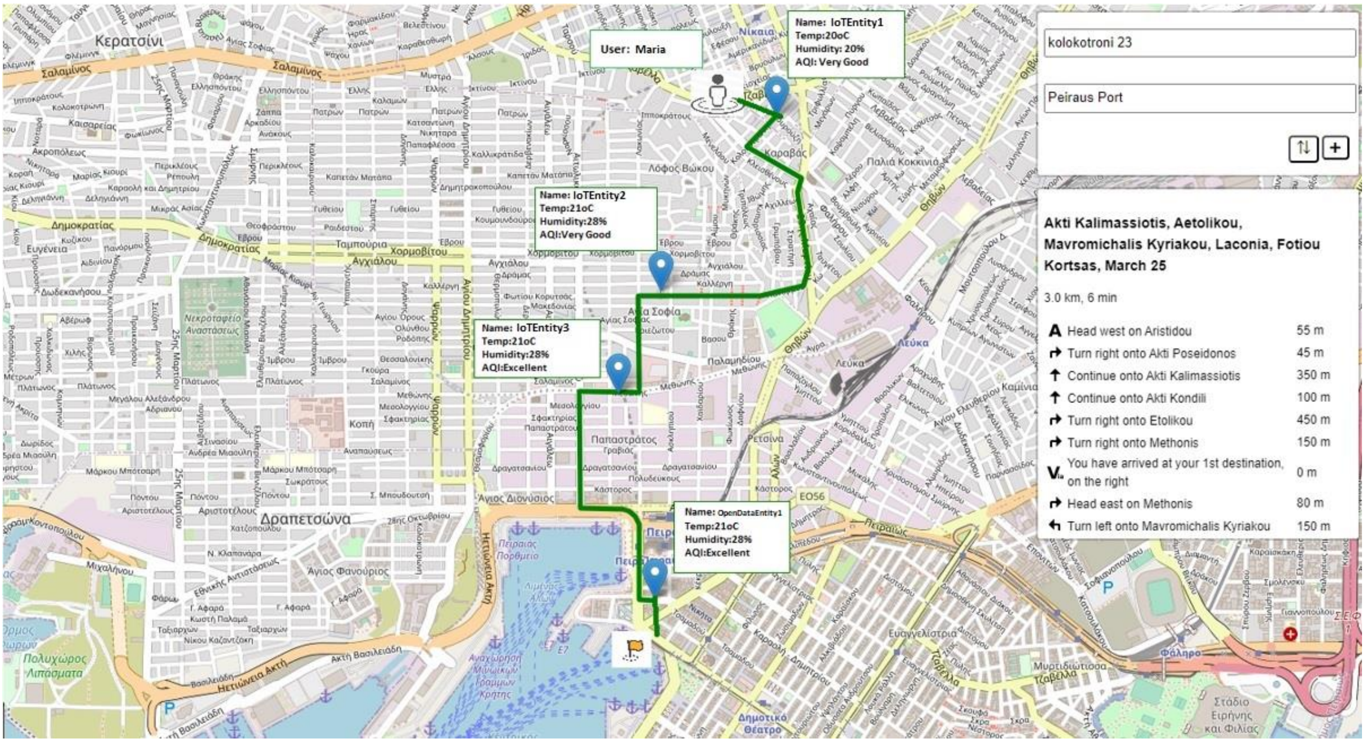
The *SEs* of the S2NeTM provide a common ontology that facilitates interoperability and alignment between the heterogeneous entities. The ontology alignment and reasoning mechanisms in the *SEs* component can dynamically find new relationships among entities based on their shared knowledge. In the Green Route scenario, the *Semantic Annotation* module allows entities to annotate their data with semantic tags that enable the alignment and reasoning mechanisms to identify and utilize new relationships. For instance, the environmental sensors and the sensor web service share the same environmental data ontology, which enables them to establish a *co-semantics* relationship. This relationship means that they can collaborate more effectively in finding the greenest route for Maria.

The *TM* component of the S2NeTM approach evaluates the trustworthiness of the entities based on their behavior, history, and feedback from other entities. In the Green Route scenario, the trustworthiness of the sensor web service is evaluated based on its data quality and availability. The trustworthiness of the mobile crowdsourcing application is evaluated based on the accuracy and usefulness of the route suggestions provided by its users.

The *FS* and *RM* components of the S2NeTM facilitate the social relationships among entities. The *co-location* relationship is established among entities that are physically close to each other, such as Maria and the environmental sensors near her house. The *co-work* relationship is established among entities that collaborate to achieve a common goal, such as Maria and the mobile crowdsourcing application users who share their route suggestions. The *co-semantics* relationship is established among entities that share similar semantics, such as the environmental sensors and the sensor web service that share the same environmental data ontology. These relationships enable the entities to collaborate more effectively and contribute to finding the greenest route for Maria.

Using these social relationships and semantic annotations, the entities collaborate to find the greenest route for Maria. The environmental sensors provide real-time data on the temperature, humidity, and air quality in the area, and the sensor web service provides historical and open data on environmental conditions in the city. Maria uses the mobile crowdsourcing application to share her preferences, such as avoiding busy roads and congested areas, and the application users provide their route suggestions. The S2NeTM aligns and assesses the data from the various entities to find the most environmentally friendly route for Maria to commute to the port.

Figure 6 displays an end user application designed for the Green Route use case scenario. The figure illustrates the interface of the application, which allows users to input their starting location and desired destination, and then generates an optimized route based on various factors such as traffic, air quality, and noise. The application also provides real-time updates on the user's progress and environmental impact.



User: Maria

Name: IoTEntity1
Temp:20oC
Humidity: 20%
AQI: Very Good

Name: IoTEntity2
Temp:21oC
Humidity:28%
AQI:Very Good

Name: IoTEntity3
Temp:21oC
Humidity:28%
AQI:Excellent

Name: OpenDataEntity1
Temp:21oC
Humidity:28%
AQI:Excellent

kolokotroni 23

Peiraus Port

⇅ +

Akti Kalimassiotis, Aetolikou, Mavromichalis Kyriakou, Laconia, Fotiou Kortsas, March 25

3.0 km, 6 min

A Head west on Aristidou 55 m
↱ Turn right onto Akti Poseidonos 45 m
↑ Continue onto Akti Kalimassiotis 350 m
↑ Continue onto Akti Kondili 100 m
↱ Turn right onto Etolikou 450 m
↱ Turn right onto Methonis 150 m
V You have arrived at your 1st destination, on the right 0 m
↱ Head east on Methonis 80 m
↰ Turn left onto Mavromichalis Kyriakou 150 m

Figure 6. End-user application for Green Route.

4.2. Building the Scenario RDF Knowledge Graph

The RDF knowledge graph (KG) is designed to identify an environmentally friendly route using environmental sensor data from IoT entities and open data entities. Each entity in the scenario, such as *IoTEntity*, has sensors that measure values for temperature, humidity, and air quality. These entities have relationships with other entities, which are represented using properties such as *hasCoSemanticsRelationshipWith* and *hasCoworkRelationshipWith*. These relationships are used to determine the most environmentally friendly route to take.

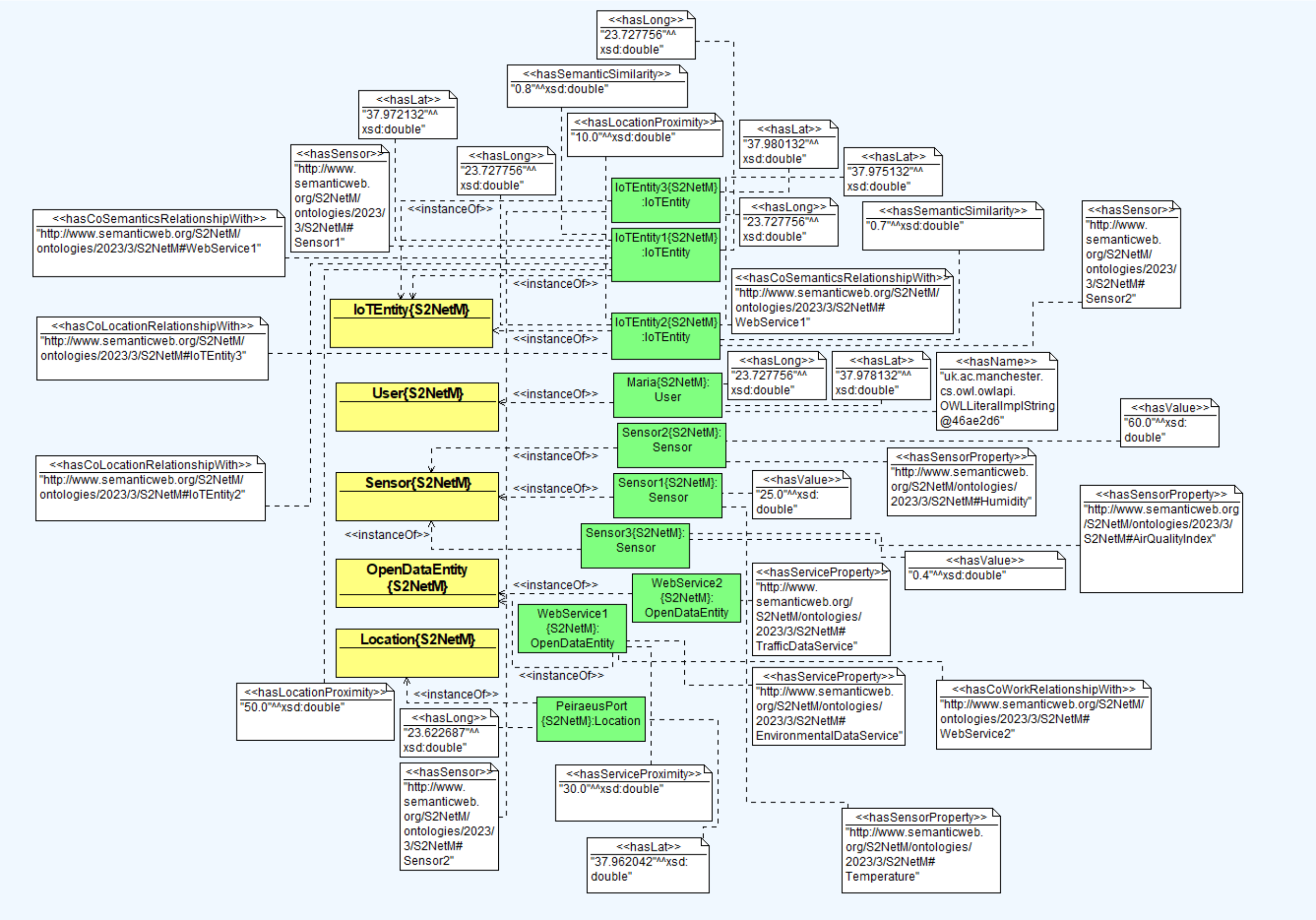
In the RDF KG for this scenario, there are three IoT entities with different types of measurements, and two open data entities that provide different types of data. There are also three types of relationships between these entities, including *co-location*, *co-work* and *co-semantics* relationships with different proximity and similarity values.

Specifically, we have the following:

- *Co-location* relationships between *IoTEntity1* and *IoTEntity2*, and between *IoTEntity2* and *IoTEntity3*, with different proximity values;
- *Co-semantics* relationships between *IoTEntity1* and *OpenDataEntity1*, and between *IoTEntity2* and *OpenDataEntity1*, with different similarity values;
- A *Co-work* relationship between *OpenDataEntity1* and *OpenDataEntity2*, with a proximity value.

The *GreenRoute* resource in the KG is of type *s2netm:Route*, and has properties describing its start and end locations, as well as whether it is optimal or not. The graph also includes information about the green route passing through three IoT entities (*IoTEntity1*, *IoTEntity2*, *IoTEntity3*), as well as two open data entities (*OpenDataEntity1*, *OpenDataEntity2*). Appendix A provides the RDF code for the Green Route scenario, utilizing the S2NeTM Ontology.

Figure 7 presents a visual representation of the RDF KG for the Green Route use case scenario, which uses the S2NeTM ontology. The use of color coding and clear labeling of the classes, instances, and properties helps to make the relationships between the entities in the scenario easily understandable. The yellow boxes in the figure represent the classes defined in the ontology, such as *User*, *Location*, *Sensor*, *IoTEntity*, *OpenDataEntity* and others. The green boxes represent the instances of these classes, such as Maria (an instance of *User*) and PiraeusPort (an instance of *Location*). The white boxes represent the instances of properties, such as *hasName*, *hasLat*, and *hasLong*. The visualization in Figure 7 provides a clear and concise representation of the relationships between the different entities in the scenario RDF KG. For example, it shows that Maria is an instance of the *User* class, and that she has a name (*hasName*) and a location (*hasLat* and *hasLong*). It also shows that the PiraeusPort of is an instance of the *Location* class, and that it has a location with latitude and longitude values.



<<hasLong>> "23.727756"^^ xsd:double"
<<hasSemanticSimilarity>> "0.8"^^xsd:double"
<<hasLat>> "37.972132"^^ xsd:double"
<<hasLocationProximity>> "10.0"^^xsd:double"
<<hasLat>> "37.980132"^^ xsd:double"
<<hasLat>> "37.975132"^^ xsd:double"
<<hasSensor>> "http://www.semanticweb.org/S2NetM/ontologies/2023/3/S2NetM#Sensor1"
<<hasLong>> "23.727756"^^ xsd:double"
<<instanceOf>>
IoTEntity3{S2NetM}:IoTEntity
IoTEntity1{S2NetM}:IoTEntity
<<hasLong>> "23.727756"^^ xsd:double"
<<hasSemanticSimilarity>> "0.7"^^xsd:double"
<<hasSensor>> "http://www.semanticweb.org/S2NetM/ontologies/2023/3/S2NetM#Sensor2"
<<hasCoSemanticsRelationshipWith>> "http://www.semanticweb.org/S2NetM/ontologies/2023/3/S2NetM#WebService1"
<<instanceOf>>
<<hasCoSemanticsRelationshipWith>> "http://www.semanticweb.org/S2NetM/ontologies/2023/3/S2NetM#WebService1"
IoTEntity{S2NetM}
IoTEntity2{S2NetM}:IoTEntity
<<instanceOf>>
<<hasCoLocationRelationshipWith>> "http://www.semanticweb.org/S2NetM/ontologies/2023/3/S2NetM#IoTEntity3"
<<hasLong>> "23.727756"^^ xsd:double"
<<hasLat>> "37.978132"^^ xsd:double"
<<hasName>> "uk.ac.manchester.cs.owl.owlapi.OWLLiteralImplString@46ae2d6"
User{S2NetM}
<<instanceOf>>
Maria{S2NetM}: User
<<hasValue>> "60.0"^^xsd: double"
Sensor2{S2NetM}: Sensor
<<instanceOf>>
<<hasSensorProperty>> "http://www.semanticweb.org/S2NetM/ontologies/2023/3/S2NetM#Humidity"
<<hasCoLocationRelationshipWith>> "http://www.semanticweb.org/S2NetM/ontologies/2023/3/S2NetM#IoTEntity2"
Sensor{S2NetM}
<<instanceOf>>
Sensor1{S2NetM}: Sensor
<<hasValue>> "25.0"^^xsd: double"
<<hasSensorProperty>> "http://www.semanticweb.org/S2NetM/ontologies/2023/3/S2NetM#AirQualityIndex"
<<instanceOf>>
Sensor3{S2NetM}: Sensor
<<hasValue>> "0.4"^^xsd:double"
OpenDataEntity {S2NetM}
<<instanceOf>>
WebService2 {S2NetM}: OpenDataEntity
<<hasServiceProperty>> "http://www.semanticweb.org/S2NetM/ontologies/2023/3/S2NetM#TrafficDataService"
WebService1 {S2NetM}: OpenDataEntity
Location{S2NetM}
<<instanceOf>>
<<hasServiceProperty>> "http://www.semanticweb.org/S2NetM/ontologies/2023/3/S2NetM#EnvironmentalDataService"
<<hasCoWorkRelationshipWith>> "http://www.semanticweb.org/S2NetM/ontologies/2023/3/S2NetM#WebService2"
<<hasLocationProximity>> "50.0"^^xsd:double"
<<instanceOf>>
<<hasLong>> "23.622687"^^ xsd:double"
PiraeusPort {S2NetM}:Location
<<hasSensor>> "http://www.semanticweb.org/S2NetM/ontologies/2023/3/S2NetM#Sensor2"
<<hasServiceProximity>> "30.0"^^xsd:double"
<<hasLat>> "37.962042"^^xsd: double"
<<hasSensorProperty>> "http://www.semanticweb.org/S2NetM/ontologies/2023/3/S2NetM#Temperature"

Figure 7. RDF knowledge graph for Green Route scenario.

Furthermore, Figure 7, shows the relationships between the instances of the different classes. For example, it shows that *IoTEntity1* is an instance of *IoTEntity*, and has a sensor (*hasSensor*) named *Sensor1* that has a sensor type (*hasSensorType*) called *TemperatureSensor*. It also shows that *IoTEntity1* has a *co-location* relationship with *IoTEntity2*, and that *IoTEntity2* has a *co-location* relationship with *IoTEntity3* with proximity values (*hasLocationProximity*) of 50 and 10, respectively.

4.3. Semantic Reasoning with CYPHER Queries

To analyze semantic relationships between entities within the Green Route scenario, we employ a dual approach consisting of ontology alignment and Cypher queries on the RDF KG. Cypher is a query language specifically designed for querying graph databases, allowing for the retrieval and manipulation of data represented as a graph, using a concise and expressive syntax. The analysis focuses on three distinct types of social relationships: *co-location*, *co-work*, and *co-semantics.*

4.3.1. Co-Location Social Relationships

To identify *co-location* relationships between entities, we use the *geofromtext* function in Cypher to calculate the distance between entities locations. We then use a threshold value to determine whether the sensors are close enough to be considered co-located. Here, is an example of the Cypher query used to identify *co-location* relationships:

```
MATCH (e1: Entity), (s2: Entity)

WHERE e1<>e2 AND distance(point({latitude: s1.hasLocationLat, longitude:
e1.hasLocationLong}), point({latitude: e2.hasLocationLat, longitude: 2.hasLocationLong})) < 500

CREATE (e1)-[:CO_LOCATED]->(e2)
```

This query matches all pairs of *Entity* nodes that are not identical and have a distance less than 500 m between their respective locations. The query then creates a CO_LOCATED relationship between the two sensors.

4.3.2. Co-Work Social Relationships

To identify *co-work* relationships between sensors, we use the *computeBy* property of the *Entity* nodes. If two entities compute the same *Service*, they are considered to be co-working. Here, is an example of the Cypher query used to identify *co-work* relationships:

```
MATCH (s1: Service)-[:COMPUTES_BY]->(e1:Entity)

MATCH (s2: Service)-[:COMPUTES_BY]->(e2:Entity)

WHERE e1 <> e2 AND s1=s2

CREATE (e1)-[:CO_WORK]->(e2)
```

This query matches pairs of *Service* nodes (s1, s2) that *computesBy Entity* nodes (e1, e2) that have the same *Service* value. The query then creates a CO_WORK relationship between the two entities.

The identified social relationships can be leveraged to develop a more comprehensive understanding of the environment in which the Green Route scenario operates. This understanding can be used to optimize various aspects of the scenario, such as identifying the most suitable sensors for measuring environmental parameters, refining the criteria for selecting the most appropriate web services for processing sensor data, and enhancing the overall accuracy of route recommendations. Moreover, the insights gained from this analysis can be utilized to develop more advanced decision-making frameworks for managing similar scenarios in other contexts. Overall, the use of Cypher queries for semantic reasoning offers a powerful approach to extract valuable insights from complex RDF data, and to help decision makers make more informed choices in complex and dynamic environments.

4.3.3. Co-Semantics Social Relationships

To identify *co-semantics* relationships between sensors, we use the *hasSemanticSimilarity* property of the *Sensor* nodes. If two sensors have a high enough *hasSemanticSimilarity* value, they are considered to be *co-semantics*. Herein, we present an example of the Cypher query used to identify *co-semantics* relationships:

```
MATCH (s1: Sensor), (s2: Sensor)

WHERE s1 <> s2 AND s1.hasOntology <> s2.hasOntology AND s1.hasSemanticSimilarity > 0.8
AND s2.hasSemanticSimilarity > 0.8

CREATE (s1)-[:CO_SEMANTICS]->(s2)
```

This query matches pairs of *Sensor* nodes (s1, s2) that are not identical and have different *hasOntology* values, but have a *hasSemanticSimilarity* value greater than 0.8. The query then creates a CO_SEMANTICS relationship between the two sensors. However, in the case that the sensor nodes have different ontologies, an ontology alignment process is required, as explained in the next section.

4.4. Ontology Alignment

The S2NeTM is designed to facilitate communication between heterogeneous IoT devices that use different ontologies to represent their data. One important aspect of this middleware is the ontology alignment method used to identify *co-semantic* relationships between entities from different ontologies. In this section, the ontology alignment method used in the context of the Green Route scenario is described.

For a pollution-free walking route suggestion, multiple entities collaborate to provide Maria with the necessary data. Two of these entities, *IoTEntity1* and *IoTEntity2*, have the potential to work together as sensors to provide real-time environmental and traffic information, despite using different ontologies to represent their data. *IoTEntity1* measures air quality and other environmental parameters, using the *S2NeTM ontology*. Meanwhile, *IoTEntity2* is mounted on a lamp post and provides real-time traffic information that can affect the level of pollution on the route, using the *SSOR ontology* [29]. By utilizing the ontology alignment, the S2NeTM enables these two sensors to exchange data with each other.

In particular, S2NeTM facilitates the communication between the two entities by applying the general ontology alignment process outlined in the following steps.

1. ***Input ontologies identification***: The alignment method identifies the appropriate ontology for each sensor device. In this case, one sensor uses the *S2NeTM ontology*, while the other sensor uses the *SSOR ontology*.
2. ***Entity identification***: The alignment method identifies the relevant entities in each ontology that represent the data from the sensors. For example, the "Location" entity in the *S2NeTM ontology* represents the data in the one sensor, while the "GeoLocation" entity in the *SSOR ontology* represents the other sensor's data.
3. ***Entity matching***: The alignment method matches the entities from different ontologies that have similar semantics. For instance, the "Entity" class from the *S2NeTM ontology* matches the "Object" class from the *SSOR ontology*.
4. ***Similarity calculation***: The alignment method calculates the similarity between the matched entities, using various similarity measures such as structural, lexical, and semantic similarity. The combined similarity score is calculated by taking the average of the individual similarity scores. Table 3 gives an example of the similarity calculation between different entities from the *S2NeTM* and *SSOR* ontologies.
5. ***Co-semantics relationship***: If the combined similarity score between two entities reaches or exceeds the predefined threshold of 0.8, the S2NeTM approves the co-semantics relationship between the entities. In this example, the overall combined

similarity score is 0.84; therefore, *IoTEntity1* is considered to have a *co-semantics* relationship with *IoTEntity2*. Once the *co-semantics* relationship has been established, the S2NeTM maps the data from each sensor to the common ontology that both sensors understand, allowing them to exchange data with each other, despite using different ontologies to represent their data.

Table 3. Example of similarity calculation between two IoT entities in Green Route scenario.

IoTEntity 1 (S2NetM Ontology)	IoTEntity 2 (SSOR Ontology)	Structural Similarity	Lexical Similarity	Semantic Similarity	Combined Similarity	Approved Co-Semantics
Entity	Object	0.5	0.5	0.92	0.65	NO
User	User	1.0	1.0	1.0	1.0	YES
Service	Service	1.0	1.0	1.0	1.0	YES
Location	GeoLocation	0.67	0.8	0.92	0.8	YES
hasRelationshipWith	hasFriend	0.5	0.5	0.75	0.58	NO
hasLocation	isLocated	0.7	0.71	0.88	0.83	YES
owns	Owns	0.9	0.9	0.83	0.87	YES
timestamp	timestamp	1.0	1.0	1.0	1.0	YES

4.5. Performance Evaluation

To evaluate the effectiveness and efficiency of the S2NeTM, we conducted a performance evaluation, considering various performance metrics and using the Green Route use case scenario. These metrics include request processing delay, setup computation time, and memory usage. The experiments were performed on a dedicated Ubuntu server with an AMD EPYC 7543 32-Core Processor and 16 GB of RAM. Additionally, we integrated the Neo4j graph database into our setup, to facilitate efficient storage and retrieval of data within the S2NetM. Leveraging Neo4j's graph database capabilities, we effectively modeled and queried the interconnected entities, services, and relationships in the SIoT environment. The reported times represent the averages obtained from twenty repeated tests.

The request processing delay metric measures the time required by the system to process and generate a green route recommendation for a specific user, such as Maria. Our experimental results revealed that the S2NeTM achieved a processing delay of 500 milliseconds (0.5 s), demonstrating its capability to deliver timely recommendations.

The setup computation time metric represents the time required to configure the graph infrastructure, considering multiple factors. Our analysis revealed an average setup time of approximately 3.1 s, indicating the system's ability to handle intricate calculations in advance, enabling timely recommendations upon request. Specifically, the setup time can be divided into two primary components: the processing time for establishing social relationships between entities, and the time taken to populate the graph nodes with data, such as air quality data. Through our experiments, we observed that establishing social relationships, including *co-location*, *co-work*, and *co-semantic* relationships, accounted for an average of 2 s. This step involves analyzing the underlying data and identifying pertinent connections based on the S2NeTM ontology.

Subsequently, the task of populating the graph nodes with data, taking into account environmental factors, yielded an average duration of 1.1 s. This step entails evaluating various criteria, including distance, air quality, and eco-friendly transportation options, to facilitate the identification of the most suitable green route for the user. The overall computation time of 3.1 s demonstrates the system's required overhead for handling periodic computations as cached information for generating green route recommendations. This overhead is incurred during the setup stage and subsequently amortized for real-time requests, thereby improving the overall user experience. It is worth noting that the specific computation times mentioned above are based on our experimental setup and may vary depending on the available computational resources.

Memory usage is an important aspect of system efficiency. We monitored the memory consumption during the green route computation and recommendation process. Our findings revealed that the system operated within a reasonable memory footprint, with an average memory usage of approximately 100 megabytes (MB), demonstrating efficient memory management. Table 4 summarizes the performance evaluation results.

Table 4. Performance metrics of the Green Route application.

Performance Metric	**Result**
Setup Time	3.1 secs
Processing Delay	0.5 secs
Memory Usage	100 MB

Despite being a preliminary performance evaluation, this study highlights the effectiveness and efficiency of S2NetM in facilitating applications such as the Green Route recommendation system, enabling timely responses while considering important factors including setup time, processing delay and memory usage.

5. Discussion

To address the RQ1, we need to examine how the S2NeTM architecture enables social relationships among IoT devices, and facilitates communication and data sharing within a network. The S2NeTM is composed of several components, which are responsible for managing and coordinating the interactions between IoT devices. One of the key components is the *SEs*, which is responsible for providing a semantic layer to IoT devices. This semantic layer is based on the S2NeTM ontology, and enables devices to understand each other's data and metadata. The ontology includes concepts and relationships that enable the representation of social relationships among IoT devices, such as *co-location*, *co-work*, *co-semantics*, and *user* relationships.

In addition, the S2NeTM includes mechanisms for discovering and communicating with other devices in the network. This includes protocols for device discovery, device registration, and message routing. By using these mechanisms, devices can establish and maintain social relationships with each other, which facilitates communication and data sharing within the network. Furthermore, the S2NeTM provides a semantic repository, which is used to store and manage the data and metadata associated with IoT devices. In this case, we use Neo4j as our semantic repository, which enables efficient querying and retrieval of data based on the S2NeTM ontology. The ontology alignment and reasoning mechanisms enable the integration of data from different sources, and the inference of new relationships based on existing data.

To address RQ2, we need to examine the benefits of using the S2NeTM in IoT environments and how it compares to other solutions. The benefits of using the S2NeTM in IoT environments are explicit. Firstly, it enables the development of more advanced and sophisticated applications that require a social network of things. For example, in smart home environments, the S2NeTM can be used to create a social network of appliances, which can work together to optimize energy consumption, automate tasks, and provide personalized services to users. Secondly, the S2NeTM can support large-scale IoT deployments, making it suitable for a wide range of applications in various domains. This scalability is achieved using distributed architectures and cloud-based technologies that allow the S2NeTM to handle a high volume of data and requests. Additionally, the use of a graph database such as Neo4j as the semantic repository ensures that the S2NeTM can handle complex relationships and queries efficiently.

Traditional middleware solutions often rely on a client–server architecture, which can be limited in terms of scalability and performance. In contrast, the S2NeTM uses a distributed architecture that allows for greater scalability and performance. Additionally, the S2NeTM provides a semantic layer that enables devices to understand each other's data and metadata, which can lead to more efficient communication and data sharing.

Finally, the S2NeTM is designed to be user-friendly and easy to use, even for non-experts in semantic technologies. The use of standardized ontologies and reasoning mechanisms ensures that the infrastructure can be easily customized and adapted to different use cases and applications.

To answer RQ3, we can discuss how the S2NeTM can be customized and adapted to support specific use cases and applications in different domains. The S2NeTM ontology provides a generic framework for describing devices, their capabilities, and their social relationships, which can be extended and adapted to support specific use cases in different domains. For example, in the smart home domain, the S2NeTM can be customized to support the social relationships between appliances, sensors, and other devices in the home. This customization can enable more efficient energy consumption, automation of tasks, and personalized services for users. In the healthcare domain, the S2NeTM can be adapted to support the social relationships between medical devices, patients, and healthcare providers. This customization can improve patient outcomes, reduce costs, and enhance the overall quality of care. In the transportation domain, the S2NeTM can be customized to support the social relationships between vehicles, traffic infrastructure, and other devices in the network. This customization can enable more efficient traffic management, reduce congestion, and improve safety on the roads.

Furthermore, the S2NeTM can be adapted to support industry 4.0 applications, which involves the integration of physical and digital systems in manufacturing and industrial processes. The S2NeTM can support the social relationships between machines, sensors, and other devices in the production line, enabling more efficient production processes and reducing downtime. In each of these domains, the S2NeTM can be customized to support specific use cases and applications, leveraging the power of social relationships between devices to enable more efficient and effective IoT deployments.

To facilitate customization and adaptation, the S2NeTM provides a modular architecture that allows for the addition and removal of components as needed. For example, different SEs can be used to support different ontologies and reasoning mechanisms, depending on the specific use case and domain. The semantic repository can also be customized to support different databases and data structures, such as the use of Neo4j for graph-based data storage.

6. Conclusions

SIoT is a paradigm that combines the best of social networks with IoT networks to create a more collaborative and efficient system. SIoT offers several benefits, such as enhanced collaboration, scalability, navigability, flexibility, and decision-making models based on trustworthiness. However, SIoT also presents several challenges that need to be addressed, including dynamic friendship selection mechanisms, privacy and security, and interoperability and standardization. To fully exploit the potential of SIoT, it is crucial to establish semantic interoperability between the various entities, applications, and networks that comprise the system. The proposed S2NeTM addresses this need by leveraging semantic technologies to establish and enhance social relationships among heterogeneous entities in the SIoT network. The S2NeTM can provide effective and efficient semantic interoperability solutions for the rapidly growing SIoT ecosystem, leading to more efficient and effective processes and advanced applications and services that provide increased value to end-users.

Our future work will be centered on harnessing the full potential of SIoT by prioritizing the interoperability with other middleware solutions and advancing the development of new ontologies and reasoning mechanisms. Additionally, we will delve into the exploration of machine learning and AI techniques' integration, while also focusing on enhancing ontology alignment mechanisms. These efforts aim to facilitate the emergence of more sophisticated and intelligent SIoT applications across a wide range of domains. Another future direction for the S2NeTM is to integrate edge computing to enable real-time decision making and reduce latency. By moving some processing and analysis to the edge, we may enhance the performance and efficiency of IoT applications. Finally, in future experiments, we plan to focus on scalability testing. This will involve testing the system's capabilities in handling a high volume of data and user requests, ensuring that it can continue to provide timely and reliable recommendations.

Author Contributions: Conceptualization, A.P. and C.G.; methodology, C.G.; software, A.P. and D.L.; validation, A.P., D.L. and C.G.; formal analysis, D.L.; investigation, A.P.; resources, A.P., D.L. and C.G.; data curation, D.L.; writing—original draft preparation, A.P.; writing—review and editing, A.P., D.L. and C.G.; visualization, A.P.; supervision, C.G.; project administration, C.G.; funding acquisition, C.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the "Research e-Infrastructure [e-Aegean R&D Network] Action 1.2 e-Aegean Geospatial data services" project, Code Number MIS 5046494, which is implemented within the framework of the "Regional Excellence" Action of the Operational Program "Competitiveness, Entrepreneurship and Innovation". The action was cofunded by the Europe-an Regional Development Fund (ERDF) and the Greek State [Partnership and Cooperation Agreement 2014–2020].

Data Availability Statement: Not applicable.

Acknowledgments: We express our gratitude to our colleagues who have been involved in the "Research e-Infrastructure [e-Aegean R&D Network] Action 1.2 e-Aegean Geospatial data services" project for their valuable support throughout the research and development activities presented in this study.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Appendix A provides the code in RDF format for the Green Route scenario described in Section 4. This code is presented in a clear and organized manner, using the S2NeTM ontology to define the classes, instances, and properties used in the scenario. The RDF format allows for easy exchange of information and interoperability between different systems, making it a valuable tool for building and sharing knowledge representations. With the code provided in Appendix A, researchers and practitioners can replicate and extend the scenario, and use it as a reference for building their own RDF-based systems and applications.

```
@prefix S2NetM: <http://www.semanticweb.org/S2NetM/ontologies/2023/3/S2NetM#>
(accessed on 8 May 2023).
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> (accessed on 8 May 2023).

S2NetM:Maria a S2NetM:User;
  S2NetM:hasName “Maria”^^xsd:string;
  S2NetM:hasLat “37.978132”^^xsd:double;
  S2NetM:hasLong “23.727756”^^xsd:double.

S2NetM:PeiraeusPort a S2NetM:Location;
  S2NetM:hasLat “37.962042”^^xsd:double;
  S2NetM:hasLong “23.622687”^^xsd:double.

S2NetM:IoTEntity1 a S2NetM:IoTEntity;
  S2NetM:hasSensor S2NetM:Sensor1;
  S2NetM:hasLat “37.972132”^^xsd:double;
  S2NetM:hasLong “23.727756”^^xsd:double.
```

```
S2NetM:IoTEntity2 a S2NetM:IoTEntity;
  S2NetM:hasSensor S2NetM:Sensor2;
  S2NetM:hasLat “37.975132”^^xsd:double;
  S2NetM:hasLong “23.727756”^^xsd:double.

S2NetM:IoTEntity3 a S2NetM:IoTEntity;
  S2NetM:hasSensor S2NetM:Sensor2;
  S2NetM:hasLat “37.980132”^^xsd:double;
  S2NetM:hasLong “23.727756”^^xsd:double.

S2NetM:Sensor1 a S2NetM:Sensor;
  S2NetM:hasSensorProperty S2NetM:Temperature.

S2NetM:Sensor2 a S2NetM:Sensor;
  S2NetM:hasSensorProperty S2NetM:Humidity.

S2NetM:Sensor3 a S2NetM:Sensor;
  S2NetM:hasSensorProperty S2NetM:AirQualityIndex.

S2NetM:WebService1 a S2NetM:OpenDataEntity;
  S2NetM:hasServiceProperty S2NetM:EnvironmentalDataService.

S2NetM:WebService2 a S2NetM:OpenDataEntity;
  S2NetM:hasServiceProperty S2NetM:TrafficDataService.

S2NetM:Sensor1 S2NetM:hasValue “25”^^xsd:double.
  S2NetM:Sensor2 S2NetM:hasValue “60”^^xsd:double.
  S2NetM:Sensor3 S2NetM:hasValue “0.4”^^xsd:double.

S2NetM:IoTEntity1 S2NetM:hasCoLocationRelationshipWith S2NetM:IoTEntity2;
  S2NetM:hasLocationProximity “50”^^xsd:double.

S2NetM:IoTEntity2 S2NetM:hasCoLocationRelationshipWith S2NetM:IoTEntity3;
  S2NetM:hasLocationProximity “10”^^xsd:double.

S2NetM:IoTEntity1 S2NetM:hasCoSemanticsRelationshipWith S2NetM:WebService1;
  S2NetM:hasSemanticSimilarity “0.8”^^xsd:double.

S2NetM:IoTEntity2 S2NetM:hasCoSemanticsRelationshipWith S2NetM:WebService1;
  S2NetM:hasSemanticSimilarity “0.7”^^xsd:double.

S2NetM:WebService1 S2NetM:hasCoWorkRelationshipWith S2NetM:WebService2;
  S2NetM:hasServiceProximity “30”^^xsd:double.
```

References

1. Qian, Y.; Wu, D.; Bao, W.; Lorenz, P. The internet of things for smart cities: Technologies and applications. *IEEE Netw.* **2019**, *33*, 4–5. [CrossRef]
2. Noura, M.; Atiquzzaman, M.; Gaedke, M. Interoperability in internet of things: Taxonomies and open challenges. *Mob. Netw. Appl.* **2019**, *24*, 796–809. [CrossRef]
3. Afzal, B.; Umair, M.; Shah, G.A.; Ahmed, E. Enabling IoT platforms for social IoT applications: Vision, feature mapping, and challenges. *Future Gener. Comput. Syst.* **2019**, *92*, 718–731. [CrossRef]
4. Roopa, M.S.; Pattar, S.; Buyya, R.; Venugopal, K.R.; Iyengar, S.S.; Patnaik, L.M. Social Internet of Things (SIoT): Foundations, thrust areas, systematic review and future directions. *Comput. Commun.* **2019**, *139*, 32–57. [CrossRef]
5. Amin, F.; Majeed, A.; Mateen, A.; Abbasi, R.; Hwang, S.O. A systematic survey on the recent advancements in the Social Internet of Things. *IEEE Access* **2022**, *10*, 63867–63884. [CrossRef]
6. Atzori, L.; Iera, A.; Morabito, G. Siot: Giving a social structure to the internet of things. *IEEE Commun. Lett.* **2011**, *15*, 1193–1195. [CrossRef]
7. Zannou, A.; Boulaalam, A.; Nfaoui, E.H. Siot: A new strategy to improve the network lifetime with an efficient search process. *Future Internet* **2020**, *13*, 4. [CrossRef]

8. Shahab, S.; Agarwal, P.; Mufti, T.; Obaid, A.J. SIoT (social internet of things): A review. *ICT Anal. Appl.* **2022**, *314*, 289–297. [CrossRef]
9. Marche, C.; Atzori, L.; Iera, A.; Militano, L.; Nitti, M. Navigability in social networks of objects: The importance of friendship type and nodes' distance. In Proceedings of the 2017 IEEE Globecom Workshops (GC Wkshps), Singapore, 4–8 December 2017; pp. 1–6. [CrossRef]
10. Farhadi, B.; Rahmani, A.M.; Asghari, P.; Hosseinzadeh, M. Friendship selection and management in social Internet of Things: A systematic review. *Comput. Netw.* **2021**, *201*, 108568. [CrossRef]
11. Vahdat-Nejad, H.; Mazhar-Farimani, Z.; Tavakolifar, A. Social Internet of Things and new generation computing—A survey. In *Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications: Emerging Technologies for Connected and Smart Social Objects*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 139–149. [CrossRef]
12. Saura, J.R.; Ribeiro-Soriano, D.; Palacios-Marqués, D. Setting privacy "by default" in social IoT: Theorizing the challenges and directions in Big Data Research. *Big Data Res.* **2021**, *25*, 100245. [CrossRef]
13. Chahal, R.K.; Kumar, N.; Batra, S. Trust management in social Internet of Things: A taxonomy, open issues, and challenges. *Comput. Commun.* **2020**, *150*, 13–46. [CrossRef]
14. Khan, W.Z.; Aalsalem, M.Y.; Khan, M.K.; Arshad, Q. When social objects collaborate: Concepts, processing elements, attacks and challenges. *Comput. Electr. Eng.* **2017**, *58*, 397–411. [CrossRef]
15. Nitti, M.; Atzori, L.; Cvijikj, I.P. Friendship selection in the social internet of things: Challenges and possible strategies. *IEEE Internet Things J.* **2014**, *2*, 240–247. [CrossRef]
16. Roopa, M.S.; Buyya, R.; Venugopal, K.R.; Iyengar, S.S.; Patnaik, L.M. DRCM: Dynamic relationship creation and management in social internet of things. *Int. J. Intell. Internet Things Comput.* **2021**, *1*, 200–229. [CrossRef]
17. Um, T.W.; Lee, E.; Lee, G.M.; Yoon, Y. Design and implementation of a trust information management platform for social internet of things environments. *Sensors* **2019**, *19*, 4707. [CrossRef]
18. Omolara, A.E.; Alabdulatif, A.; Abiodun, O.I.; Alawida, M.; Alabdulatif, A.; Arshad, H. The internet of things security: A survey encompassing unexplored areas and new insights. *Comput. Secur.* **2022**, *112*, 102494. [CrossRef]
19. Farahbakhsh, B.; Fanian, A.; Manshaei, M.H. TGSM: Towards trustworthy group-based service management for social IoT. *Internet Things* **2021**, *13*, 100312. [CrossRef]
20. Gharib, M.; Giorgini, P.; Mylopoulos, J. An ontology for privacy requirements via a systematic literature review. *J. Data Semant.* **2020**, *9*, 123–149. [CrossRef]
21. Pliatsios, A.; Goumopoulos, C.; Kotis, K. A review on iot frameworks supporting multi-level interoperability—The semantic social network of things framework. *Int. J. Adv. Internet Technol.* **2020**, *13*, 46–64.
22. Pliatsios, A.; Goumopoulos, C.; Kotis, K. Interoperability in IoT: A Vital Key Factor to Create the Social Network of Things. In Proceedings of the Thirteenth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies UBICOMM, Porto, Portugal, 22–26 September 2019; pp. 63–69.
23. Ali, S.; Kibria, M.G.; Jarwar, M.A.; Lee, H.K.; Chong, I. A model of socially connected web objects for IoT applications. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 6309509. [CrossRef]
24. Rahman, H.; Hussain, M.I. A comprehensive survey on semantic interoperability for Internet of Things: State-of-the-art and research challenges. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*, e3902. [CrossRef]
25. Malekshahi Rad, M.; Rahmani, A.M.; Sahafi, A.; Nasih Qader, N. Social Internet of Things: Vision, challenges, and trends. *Hum. Cent. Comput. Inf. Sci.* **2020**, *10*, 1–40. [CrossRef]
26. Palo, H.K. Semantic IoT: The key to realizing IoT value. In *Semantic IoT: Theory and Applications: Interoperability, Provenance and Beyond*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 81–102. [CrossRef]
27. Lee, I. The Internet of Things for enterprises: An ecosystem, architecture, and IoT service business model. *Internet Things* **2019**, *7*, 100078. [CrossRef]
28. Novo, O.; Francesco, M.D. Semantic interoperability in the IoT: Extending the web of things architecture. *ACM Trans. Internet Things* **2020**, *1*, 1–25. [CrossRef]
29. Pliatsios, A.; Kotis, K.; Goumopoulos, C. A Systematic Review on Semantic Interoperability in the IoE-enabled Smart Cities. *Internet Things* **2023**, *22*, 100754. [CrossRef]
30. Amara, F.Z.; Hemam, M.; Djezzar, M.; Maimour, M. Semantic web technologies for internet of things semantic interoperability. In *Advances in Information, Communication and Cybersecurity: Proceedings of ICI2C'21*; Springer International Publishing: Cham, Switzerland, 2022; pp. 133–143. [CrossRef]
31. Dhelim, S.; Ning, H.; Farha, F.; Chen, L.; Atzori, L.; Daneshmand, M. IoT-enabled social relationships meet artificial social intelligence. *IEEE Internet Things J.* **2021**, *8*, 17817–17828. [CrossRef]
32. Hamrouni, A.; Khanfor, A.; Ghazzai, H.; Massoud, Y. Context-Aware Service Discovery: Graph Techniques for IoT Network Learning and Socially Connected Objects. *IEEE Access* **2022**, *10*, 107330–107345. [CrossRef]
33. De, S.; Zhou, Y.; Larizgoitia Abad, I.; Moessner, K. Cyber–physical–social frameworks for urban big data systems: A survey. *Appl. Sci.* **2017**, *7*, 1017. [CrossRef]
34. Mendhurwar, S.; Mishra, R. Integration of social and IoT technologies: Architectural framework for digital transformation and cyber security challenges. *Enterp. Inf. Syst.* **2021**, *15*, 565–584. [CrossRef]

35. Kim, J.E.; Maron, A.; Mosse, D. Socialite: A flexible framework for social internet of things. In Proceedings of the 2015 16th IEEE International Conference on Mobile Data Management, Pittsburgh, PA, USA, 15–18 June 2015; Volume 1, pp. 94–103. [CrossRef]
36. Shamszaman, Z.U.; Ali, M.I. Toward a smart society through semantic virtual-object enabled real-time management framework in the social Internet of Things. *IEEE Internet Things J.* **2017**, *5*, 2572–2579. [CrossRef]
37. Kasnesis, P.; Patrikakis, C.Z.; Kogias, D.; Toumanidis, L.; Venieris, I.S. Cognitive friendship and goal management for the social IoT. *Comput. Electr. Eng.* **2017**, *58*, 412–428. [CrossRef]
38. Gulati, N.; Kaur, P.D. Towards socially enabled internet of industrial things: Architecture, semantic model and relationship management. *Ad. Hoc. Netw.* **2019**, *91*, 101869. [CrossRef]
39. Beltran, V.; Ortiz, A.M.; Hussein, D.; Crespi, N. A semantic service creation platform for social IoT. In Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, Republic of Korea, 6–8 March 2014; pp. 283–286. [CrossRef]
40. Gulati, N.; Kaur, P.D. When things become friends: A semantic perspective on the Social Internet of Things. In *Smart Innovations in Communication and Computational Sciences: Proceedings of ICSICCS 2017, Volume 2*; Springer: Singapore, 2019; pp. 149–159. [CrossRef]
41. Bouazza, H.; Said, B.; Laallam, F.Z. A hybrid IoT services recommender system using social IoT. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 5633–5645. [CrossRef]
42. Rajendran, S.; Jebakumar, R. Object Recommendation based Friendship Selection (ORFS) for navigating smarter social objects in SIoT. *Microprocess. Microsyst.* **2021**, *80*, 103358. [CrossRef]
43. Perera, C.; Zaslavsky, A.; Christen, P.; Georgakopoulos, D. Context aware computing for the internet of things: A survey. *IEEE Commun. Surv. Tutor.* **2013**, *16*, 414–454. [CrossRef]
44. Sezer, O.B.; Dogdu, E.; Ozbayoglu, A.M. Context-aware computing, learning, and big data in internet of things: A survey. *IEEE Internet Things J.* **2017**, *5*, 1–27. [CrossRef]
45. Kilani, R.; Zouinkhi, A.; Bajic, E.; Abdelkrim, M.N. Socialization of Smart Communicative Objects in Industrial Internet of Things. *IFAC-PapersOnLine* **2022**, *55*, 1924–1929. [CrossRef]
46. Du, Q.; Song, H.; Zhu, X. Social-feature enabled communications among devices toward the smart IoT community. *IEEE Commun. Mag.* **2018**, *57*, 130–137. [CrossRef]
47. Corpino, S.; Mirri, S.; Sole, M.; Giusto, D.; Pau, G.; Girau, R. On implementing socialization algorithms on Virtual Objects in the Social IoT. In Proceedings of the 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2022; pp. 307–312. [CrossRef]
48. Atzori, L.; Iera, A.; Morabito, G. Making things socialize in the Internet—Does it help our lives? In Proceedings of the ITU Kaleidoscope 2011: The Fully Networked Human?-Innovations for Future Networks and Services (K-2011), Cape Town, South Africa, 12–14 December 2011; pp. 1–8.
49. Yang, Y.; Xu, J.; Xu, Z.; Zhou, P.; Qiu, T. Quantile context-aware social IoT service big data recommendation with D2D communication. *IEEE Internet Things J.* **2020**, *7*, 5533–5548. [CrossRef]
50. Almagrabi, A.O.; Al-Otaibi, Y.D. A survey of context-aware messaging-addressing for sustainable internet of things (IoT). *Sustainability* **2020**, *12*, 4105. [CrossRef]
51. Khelloufi, A.; Ning, H.; Dhelim, S.; Qiu, T.; Ma, J.; Huang, R.; Atzori, L. A social-relationships-based service recommendation system for SIoT devices. *IEEE Internet Things J.* **2020**, *8*, 1859–1870. [CrossRef]
52. Atzori, L.; Iera, A.; Morabito, G.; Nitti, M. The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Comput. Netw.* **2012**, *56*, 3594–3608. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.