C. Fernandez-Gago and D. Ferraris and R. Roman and J. Lopez, "Trust Interoperability in the Internet of Things ", *Internet of Things*, vol. 26, In press.
NICS Lab. Publications: https://www.nics.uma.es/publications

Trust Interoperability in the Internet of Things

Carmen Fernandez-Gago Davide Ferraris Rodrigo Roman Javier Lopez
mcgago@uma.es, ferraris@uma.es, rroman@uma.es,javierlopez@uma.es

Abstract

The Internet of Things (IoT) is a paradigm where entities or things are interconnected, often in heterogeneous contexts. As the interconnection happens, things establish collaborations with others, sometimes under uncertainty. Although trust can help us overcome this uncertainty, things might not be able to process the information about trust coming from other things: each thing could have its own trust model, which means its own way to understand and measure trust. If new trust relationships are to be established, it would be desirable to have a mechanism of interoperability that allows the things to process the information about the other things in terms of trust. In this paper, we describe an interoperability framework for tackling the trust interoperability issues in IoT, depending on the different types of trust models that might co-exist in the same IoT scenario.

1 Introduction

The Internet of Things (IoT) is based on the interconnection of heterogeneous entities (things) with different connection capabilities and made by different manufacturers (1). At present, this paradigm is being used in real life applications such as smart homes, health, drones or smart parking (2; 3; 4; 5).

The strength of IoT scenarios lies on one of their main features, that is, the interaction among things. This means that things have to work together towards a common goal. For example, in a smart home environment a smart fridge would have to interact with a smart cook and both with a smart watch or smart phone.

Besides the interactions, the heterogeneity of the entities or things involved in these scenarios is another key feature of these environments (6).

In a setting as the IoT, where the interconnection of entities is a fact, the interactions among them have to be a reality in order to make as much profit as possible (7). The advantages of IoT could be jeopardised if users do not perceive its goodness or even worse, if they perceive their use as a menace for their security or privacy (8). One of the security problems that arise in these scenarios is that the interactions have to happen even though in most of the cases the behaviour of the other entity involved in the interaction is uncertain, that is, the entities involved might not know how the others are going to perform when working towards their common goal. It is then when trust management systems could be of help as they overcome the lack of certainty that traditional security mechanisms are not able to solve. Trust management systems ease the decision-making process by helping the entities involved in any interaction to make informed decisions concerning their relationship with others (9). In an IoT setting, interoperability has been always an issue (10). Moreover, if the entities involved are supported by trust mechanisms the most likely will be that each of them have their own trust model, which means their own way to understand and measure trust. Thus, it could be the case when different trust managements systems using different languages and ways to obtain trust have to interact among them. It is then when we need mechanisms that guide the interaction between trust management systems, i.e., we need the trust management systems to be *interoperable*. This way, the trust information that is derived by any entity could be used and understood by another one, if needed.

When achieving interoperability for trust models in IoT, we must distinguish two different levels. On the one hand, the proper interoperability present when different entities have to co-exist, and on the other hand, interoperability between trust management systems as part of the essential services to be included in the services layers of the trust framework that we consider in (11).

The ideal interoperability trust framework should take into account the levels mentioned above. In this paper, we concentrate on the latter. In order to achieve this type of interoperability we must bear different factors in mind. First of all, entities might likely come from different manufacturers. This means that they might use different terminology for referring to the same concept. In this respect, a kind of *semantic interoperability* (not in the sense used for communication but more referred to the context), i.e., using similar terminology to refer to the same concepts related to things or entities could be useful. This might be solved by using appropriate identity management systems for IoT, which is out of the scope

of this paper.

Another important issue when dealing with interoperability is the fact that the entities might use different protocols to communicate. There have been some efforts to establish a common protocol (13) however, the lack of standards for IoT communications is a fact (14). This includes the use of standards that are not generally used in IoT.

In this paper, we present a practical methodology for tackling the trust interoperability issues in IoT, depending on the different types of trust models that might co-exist in the same IoT scenario. Our methodology for dealing with interoperability in IoT scenarios is built on two main pillars: *semantic* and *syntactic* interoperability. We identify the main elements that have to be present in the interoperable-to-be trust models and how the meaning of trust for each of them can be matched into a common one. The context of the applications plays also an important role as the factors that are around a trustor and the trustee define their semantic interoperability.

The structure of the paper is as follows. Section 2 describes the related work and the background. Section 3 introduces the methodology that we have developed for dealing with interoperability issues in trust models for IoT. Section 4 shows an example of the methodology and Section 5 shows the results of the validation of the framework. Finally, we conclude the paper and outline the future work in Section 6.

2 Related Work and Background

Trust in Computer Science takes its meaning from other disciplines such as Psychology, Economics or Sociology (15). There is not a unique definition of trust but it is widely accepted that the main purpose of trust is to assess the decision-making process when entities in a system have to interact (16).

Nevertheless, to the best of our knowledge, the specific problem of interoperability of trust models has not been addressed in the specific case of IoT environments. The problem of trust and interoperability of systems has been addressed from different points of view and for different domains than IoT such as blockchain technologies (17). In (18) the authors consider trust models for cross-domain clouds, that at the end become interoperable through the use of the trust model. Authors in (19) tackle the problem of achieving interoperable trust models by defining a meta-trust model that allows for the composition of trust models.

In fact, interoperability remains a key issue in IoT, not only for trust but also

for other functional requirements. The companies on the IoT business realise that they can make better business if they are not isolated from the others in the market. It is then very important to achieve interoperability among various products. One way functional interoperability can be achieved is by using open source frameworks upon which to build IoT applications (10). In addition, several authors highlight the importance of achieving interoperability from a semantic point of view (20; 21).

As for the problem of managing trust in IoT, there has been various efforts in the literature to address it. In addition, in the last decade, the interest in this topic has been growing. The authors in (22) elaborated an important survey on the existing literature on the field. They made a classification of the main approaches for trust in IoT based on ten properties that they consider important for these systems such as Trust relationship and decision (TRD), Privacy preservation (PP), System security and robustness (SSR), Generality (G) or Identity trust (IT), which they consider the crucial properties to be present in a trust management system for IoT. It is worth to notice that some models they consider in their survery are not proper trust management systems but deal with security in IoT in general. This survey has been an important starting point for the following researches on trust management in the IoT.

More recently, Sharma et. al (25) defined a generic framework for trust in IoT based on quantitative and qualitative parameters. Then, Guo et al., developed a software engineering approach but service-based in (23). Here, the authors introduced different trust-based service management techniques depending on the type of IoT network: whether they are centralized, decentralized or hybrid. A specific solution for a typical scenario of IoT such as e-Health scenarios is introduced in (24) where the authors develop a trust-based decision-making protocol for health IoT devices.

Authors in (26) designed a scalable trust management protocol for IoT that takes into account social relationships and uses properties such as honesty, cooperativeness and community interest in order to evaluate trust. The protocol is distributed and the nodes update trust only for the nodes they are interested in or interact with. The updates are done through direct observations or indirect recommendations. Based on this model, the same authors proposed a dynamic trust management protocol for the IoT to deal with misbehaving nodes or behaviour that may change dynamically (27). Also, in (28; 29; 30) the authors based their approach on social relationships. They considered that the objects in an IoT scenario conform a social network where they establish social trust relationships. They introduced an architecture for the Social Internet of Things (SIoT). Trust is

not explicitly considered but they propose a method for determining trustworthy nodes in this socialised environment. On the other hand, an author that considered trust in the SIoT environment has been Zouzou et al. (31). In their work, the authors proposed a multi-context aware framework for the SIoT. However they do not mention trust interoperability considering it only in a general way.

Other works (32) considered an IoT environment where the 'things' are only wireless sensors. Trust management is used in this case for solving the problem of packet forwarding and, therefore, it does not deal with the heterogeneity that the IoT paradigm targets. The work in (33) proposed a centralised trust management system for the IoT that aims at managing cooperation among nodes with different resource capabilities. The model assigned trust values to cooperating nodes according to different contexts.

To the best of our knowledge no other works consider the case of interoperability for trust models. Even less, the inclusion of interoperability issues in standardisation bodies. Our paper tries to address this issue, which as we have mentioned earlier, is highlighted by some authors as a key issue.

Next, we overview the concepts and theory underneath the framework we are going to present in Section 3.

Trust management models are very context-dependent and are meant to solve a specific purpose for a specific application. Thus, different trust management models may arise in different situations even for a same device, depending on their way to derive trust or the different meaning of the concept of trust for the specific application. The authors in (34) designed a conceptual model for trust that distinguishes two types of trust models: *decision models* and *evaluation models*. The classification refers to the way trust is obtained. Our framework for trust interoperability will use this classification as the basis for its definition. The main features of these two types of models are the following:

- *Decision models*. The origins of trust management lays on these models (35). They make access control decisions by simplifying the two-step authentication and authorisation process into a one-step trust decision. Policy models and negotiation models are decision models. For both of them, the access to resources is based on the fulfillment of certain policies and credentials that specify which ones are required to access them.

- *Evaluation models*. The main purpose of these type of models is to evaluate how reliable an entity is by measuring certain factors that have an influence on trust. These models are often referred to as computational trust, which

has its origin in the work in (36). Reputation models fall into this category. For them, entities evaluate their trust in others based on the opinions on the latter ones. Propagation models are also an example of a type of evaluation models. In the latter, trust information is disseminated along trust chains.

This conceptual model for trust serves as the basis for a development framework that supports the accommodation of heterogeneous trust and reputation models (37) based on the models@run.time paradigm, called trust@run.time. Based on this paradigm of trust@run.time, an effort to make it extensible to IoT has been made in (11) in order to include trust in the development of IoT scenarios in a holistic manner. In this work, the authors introduced a framework for trust@run.time for IoT where one of the key components is the one for trust models interoperability, which is the main topic of this paper.

Including trust management models in the development of IoT entities is not a simple task and it is usually dealt with in an ad-hoc manner, i.e., it is very case-dependent. In order to tackle the problem in a holistic manner, we proposed in (11) a theoretical framework for the inclusion of trust in IoT. This framework consisted of several layers, being one of them the services layer. The interoperability of trust models is one of the most relevant services within this layer and with this paper we want to present how we model this feature. It is out of the scope of this paper quantifying trust. For the work in this paper we assume that these values have been obtained and we have to perform interoperability between them.

3 An Interoperability Framework for Trust

In the framework introduced in (11) for the inclusion of trust management systems in IoT, we identified two main challenges in order to achieve it: dynamicity and interoperability. Dynamicity can be achieved in the proposed framework by applying the trust@run.time paradigm (37). Interoperability, defined by the Merriam-Webster dictionary [1] as "the ability of a system to work with or use the parts or equipment of another system", is especially true and needed in IoT scenarios.

The ideal framework for achieving trust interoperability should be twofold. We assume that interoperability at the level of entities occurs in order to be able to achieve trust models interoperability. N. B. that the interoperability framework that we are defining next is considered for systems in which their entities interact,

[1] https://www.merriam-webster.com/dictionary/interoperability

in the sense that we have introduced earlier. We have however concentrated in IoT scenarios in this paper as it is an obvious environment where interactions happen. In any case, the framework will be easily used in scenarios other than IoT.

Regarding the possible architectures of the IoT environment, we assume that entities can be grouped into clusters. Entities belong to the same cluster if they have established a trust relationship supported by a common trust model. It could occur that an entity in a cluster has to interact with another one in a different cluster. In this case, we need to define a new trust model with all of their elements (actors, roles, engine or compliance checker, etc). It would be desirable to re-use the information available on the entity from the clusters it belongs to in order to use it in the new trust model.

We introduce then an *interoperability trust repository* that is in charge of offering the components needed to develop the new trust model that the entities need in order to establish trust between them. This is shown in Figure 1.

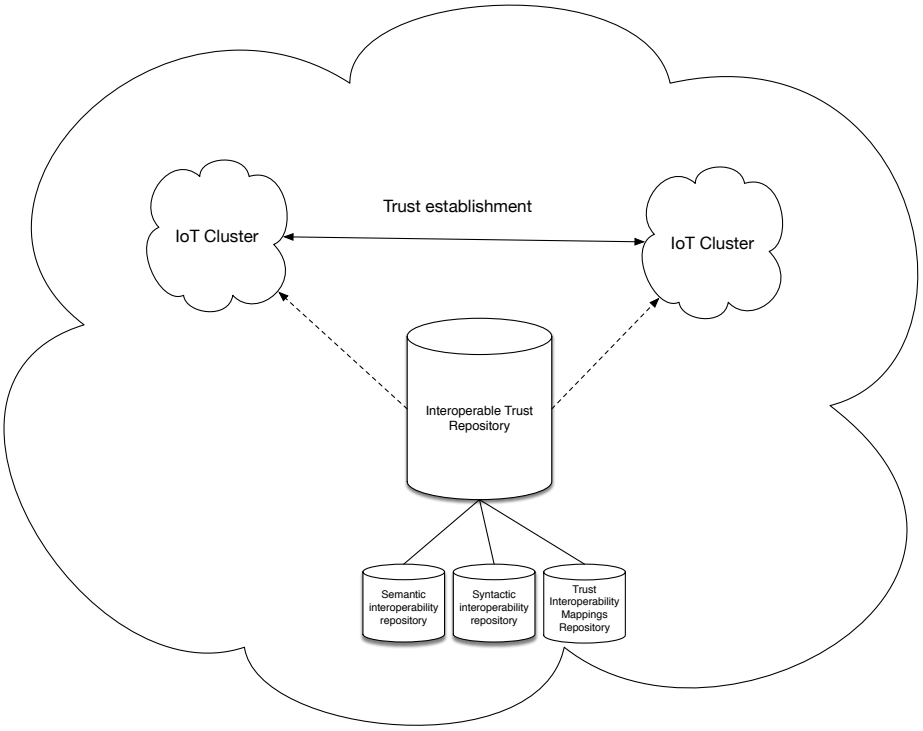

Figure 1: Trust Interoperable Framework

In this figure, we depict a centralised framework that describes how the interactions (as defined above) occur. The dotted lines mean that the information about how the transformation could be done is provided by the repository to the engine and the continuous arrow means that new trust relationships are established, that is, when an interaction is occurring. In case no line is depicted means that these entities are not interacting.

The interoperability trust repository is composed by three repositories itself: syntactic, semantic, and *trust interoperability mappings*.

Precisely, regardless of the type of interoperability needed, interoperability has to take place at two different levels: syntactic and semantic. At the syntactic level, interacting trust models should use the same nomenclature for naming their elements. In an IoT scenario, we identify some common elements for both type of models (34). There are at least two entities that in both cases are called *trustor* (the entity that places trust) on a *trustee* (trusted entity) that establish a *trust relationship*, which is a value in the case of evaluation models or the result of a decision in a decision-based model. Besides these basic and common elements for both trust models, every established trust relationship has a *purpose* that captures the *context* where the trust relationship takes place. Other factors that influence trust are the specific subjective and objective properties of the trustor and trustee (38). Thus, and in order to keep a uniform nomenclature for trust models in IoT, we will adopt these names for the basic concepts needed in IoT introduced above (trustor, trustee, trust relationship, context, etc.).

As for the semantic level, in IoT environments semantic interoperability for IoT refers to the exchange of the outputs of them in a meaningful way for both models. We propose three different interoperability semantic transformations that can happen: evaluation to evaluation trust models, decision to decision trust models and evaluation to decision trust models, and viceversa (i.e., evaluation to decision or decision to evaluation).

In the following sections, we explain how the trust interoperability mapping could be done in each of the cases.

3.1 Decision to Evaluation Trust Models

In order to perform the mapping from decision to evaluation models we have to distinguish two cases: when the decision model is a negotiation model and when it is a policy-based model.

- *Negotiation to Evaluation Trust Model*. In this case the output of the ne-

gotiation model is *trusted* or *not trusted*. Thus, if a trust evaluation model is given by a trust evaluation function, $f : D \to D'$, where D and D' are the sets of real numbers where the values of the evaluation are placed, we could define the following discrete function for the resulting evaluation model,

$$T : \{trusted, not\ trusted\} \to \{a, b\}$$

where $T(trusted) = a$ and $T(not\ trusted) = b$, $a, b \in D'$ and $b \leq a$. The way we choose these values depend on the model we use and they could be chosen as a minimum of the values of D' and a maximum respectively (they could be for instance 0 and 1 if $D' = [0, 1]$).

- *Policy-based to Evaluation Models*. In this case the transformation depends on the type of function that the evaluation model uses (outputs are discrete or continuous values) and how the values for the policy-based model are delivered. We assume anyway that these values are somehow ordered and they form a finite lattice, $\mathscr{P}$. Also, every output of the policy model could be assigned a weight (numerical value), w_i, that depends on the context, that is, the meaning of the output. Thus, for instance, in the case all the outputs have the same importance, these weights could be set up as 1.

  In the case that the evaluation model yields a discrete values in a finite set D the defined interoperability function could be as follows: $f : \mathscr{P} \to D$, where

  1. $f(max(\mathscr{P})) = max(D)$
  2. $f(min(\mathscr{P})) = min(D)$
  3. For those values in $\mathscr{P}$ other than maximum or minimum the assigned value will depend on their weight. Thus, the highest values of w_i for an element in $\mathscr{P}$ would be assigned the predecessor of $max(D)$ and the lowest one the successor of $min(D)$. We could continue this way and assign the next highest or lowest value the next predecessor or next successor respectively.

  For the case where the values of the evaluation model range in a continuous set (interval in $\mathbb{R}$) we divide the interval and as many different subintervals as the number of elements of $\mathscr{P}$, and take a representative value for each of them so that the situation is moved to the one described for the discrete value.

3.2 Evaluation to Decision Trust Models

We will discuss here how the transformation for interoperability can happen for each of the cases of decision models.

- *Evaluation to Policy-based models.* Let us assume an entity in an IoT cluster that uses an evaluation trust model trying to establish a trust relationship with another entity in another cluster using a policy-based trust model.

  Policies for the latter trust models check the conditions under which access to a resource can be granted. These conditions are usually expressed by credentials. A compliance checker checks whether the credentials satisfy the policies. The trust interoperability mapping has to be between numerical values (the output of evaluation models) and either trusted or not trusted, which is the final outcome policy-based decision models. This mapping is very context-dependent, specifically on the credentials that are used in specific use cases. Thus, if a trust evaluation model is given by a trust evaluation function, $f : D \to D'$, where D and D' are the sets of real numbers where the values of the evaluation are placed. Then, we define the *Trust Interoperability Mapping*, T, as $T : D' \to \{trusted, not\ trusted\}$ is a function defined as follows:

  $$T(d) = \begin{cases} trusted & if \quad d \in I \\ not\ trusted & if \quad d \notin I \end{cases}$$

  where $d \in I$ and $I \subseteq D'$ is a subset established as a set of trust that depends on the context of the entity.

  Thus, for example, in the simplest case that the evaluation model results are given as discrete values, 0 and 1, then they could be matched to the output of the decision model. Moreover, in the case where D' is a discrete set of numbers the mapping could be done as follows. Thus, if there is an evaluation function, f, $f : D \to \{0,1\}$, where D is the domain of f, if the obtained trust value is 0, then it can be matched to a *not trusted* decision result of the decision model. If it is 1, then this result could be matched to *trusted*. If the evaluation function yields other values such as for example, those in the set $\{0, 0.5, 1\}$ then, the value 0.5 could be matched to either $trusted$ or *not trusted*, depending on the trusted interval that is defined.

- *Evaluation to Negotiation Models.*

Let us assume an entity, A, in a cluster where an evaluation model is used, needs to establish a trust relationship with another entity, B, in another cluster where a negotiation model is used. Then, the trust information available about A is in terms of the trust values obtained through an evaluation function. B instead requires credentials in order to establish the trust relationship. If the credentials required meet the policies established in B's cluster, then the negotiation is successful and the trust relationship is established. Thus, we propose to define the *Trust Interoperability Mapping* in this case as $T : \mathscr{D} \rightarrow \mathscr{C}$, where $\mathscr{D}$ is the set of all the related elements to A in its context and $\mathscr{C}$ is the set of all the credentials that B requires in order to establish a trust relationship. For establishing the mapping, it is essential to take into account the contexts where the entities are. Thus, for example, if a device, A, is a camera that requires access to certain resources in a smart home scenario and in its cluster A is assessed in a smart home context, then if the first condition that B (another device, for instance a smart tv) requires to be met by A is to be a student, the trust interoperability should include a mechanism that extracts from the trust values of A such information as the output of T. In this case, only the role of A will be enough to verify that the credential that B requires is enough. The definition of T depends very much on the policies defined by B's cluster.

3.3 Evaluation to Evaluation Models

As in the transformation described in Section 3.2, syntactic interoperability should be a 'must' in this transformation.

However, let us assume two entities, A and B in different clusters using trust evaluation models where the results of the evaluations are in D and D' respectively, and $D, D' \subseteq \mathbb{R}$.

The trust values obtained for each entity in any of the sets are related to a specific purpose, p with respect to another entity. The interoperability transformation is defined as (A, B, d', p), where A and B are entities of different IoT clusters, $d' \in D'$ and p is the purpose for which d' has been calculated. N. B. that in the case of the transformations of Section 3.2 the purpose is not explicitly mentioned but it is used when deriving the value of the evaluation model.

The way to obtain d' is very context-dependent but it definitively depends on the purpose, p. Moreover, information about the trust values obtained by A in its cluster is available from all the possible evaluation statements provided about A. Let (A, A', d, p) be an evaluation statement available in the cluster where A

belongs to, and $\mathscr{A} = \{(A, A', d, p)\}$ the set of all statements about A, where A' is in the same cluster, d is an evaluation value and p is a purpose. As we are interested in a new trust relationship between A and B but in different clusters, let (A, B, d', p') be trust relationship that A and B aim to establish. Then, d' is obtained by using the values d available when forming the set $\mathscr{A}$. The key aspect is that the purposes p and p' have to be related somehow, so that the values already available in the repository are meaningful for the new purpose p'.

Let $\preceq$ be an order relationship established between purposes. d' and p' could be obtained as follows:

- d' can be obtained from the values of d.
- If either $p \preceq p'$ or $p' \preceq p$ then we could establish the output of the interoperable model.
- If neither $p \npreceq p'$ nor $p' \npreceq p$ then the interoperability is not possible.

In both cases where it is possible to establish the interoperability relationship, the way of obtaining the new values is context-dependent and we cannot give a general formula to do it. However, in the first case, it might be easier to obtain it as it might even be the same value that it was already in the repository.

3.4 Decision to Decision Models

There are three different types of transformations that can happen in this case. We below describe how this is done. In all the cases we are assuming an entity A in a cluster aims to establish a trust relationship with another entity B in another cluster.

- *Policy to policy models.*

  In policy-based trust models, a compliance checker determines whether the credentials owned by an entity are enough to either trust it or not. Thus, the outputs of these models are usually *trusted* and *not trusted*. Since we are interested in a transformation between two of these models, the trust interoperability mapping in this case could be defined as

  $$f : \{\textit{trusted, not trusted}\} \rightarrow \{\textit{trusted, not trusted}\}$$

  However, as the outputs depend on the policies it makes more sense to establish the mapping between the policies of the different models. Thus, it

is more important for this case to set up the semantic and syntactic interoperability to make sure that the policies, which are checked by a compliance checker can be checked by the other one as well.

- *Policy to negotiation models.*

  In this case, the importance also reverts on the syntactic and semantic interoperability repositories as the policies used in the policy model check whether the credentials owned by an entity satisfy the policies. If $\mathscr{C}$ is the set of credentials that B requires in order to establish a trust relationship with A, then the trust interoperability mapping should be in this case a semantic interoperability mapping between the credentials checked by the policy model and those required by the negotiation model.

- *Negotiation to negotiation models.*

  As in the previous case, the trust interoperability mapping will have to be defined in terms of semantic interoperability. If A aims to establish a trust relationship with B, B requires to A a minimum set of credentials that are released as part of a negotiation strategy. Thus, we have to ensure that the semantic of the credentials are common to A and B. This way B could check for A's credentials and determine whether they are valid or not for establishing a trust relationship with it.

4 Trust Interoperability in Practice

In this section and in the following one, we will explain how the framework introduced in Section 3 works by applying it to a use case scenario. Firstly, we will discuss a use case presenting a scenario where we the system actors perform trust decisions. However, in order to perform such decision interoperability is needed. This interoperability will be achieved as described in Section 3, validating the proposed method. Finally, the outputs will be considered as a common ground in order to perform a final trust decision.

4.1 Application Scenario: Field Service Team

Let us consider the scenario presented in (11) of a *Field Service Team* from a different perspective.

In this scenario, the Dispatching System (DS) of a gas company must choose between two operators working for two different companies in order to fix leaking pipes. We have two operators. Anne is working for Acme Limited Company implementing a trust evaluation model. Bob is working for Beta Corporation which is using a decision model. Luckily, the DS is provided by the Interoperability Framework presented in Figure 1, thus it can perform a trust decision even if the two companies are using different trust models.

The two operators have been provided by their company of different IoT clusters. Let us assume that these clusters can be composed of a maximum of four smart objects carried by the operators: a tablet, a smart car, a laptop and/or a smart watch.

In Figure 2, we can observe the two operators, Anne and Bob, and their IoT clusters. We can also observe that each device has a different trusted value or label depending on whether the used trust model is either a decision or an evaluation trust model.

We can now consider the following specific case.

The DS is implemented by a gas company where a sensor has detected a leaking pipe. The information carried out by the objects in this scenario have to be managed by the DS, which allocates the *Interoperable Trust Repository* introduced in Section 3. The interoperability is done among the objects and then the DS keeps the results in its repository.

The final result will be used to compare the trusted values of both the operators and converted to the trust model used by the gas company in order to perform the final decision. The gas company is implementing an evaluation model based on reputation.

4.2 IoT Clusters Formation

As we have introduced before, the DS must choose between the two operators: Anne and Bob. The clusters of each operator are defined by the objects that are available to that operator at some point in time. The DS should determine whether Anne or Bob are the best operator to attend the critical situation. In order to perform this choice, the DS needs to carry out the transformations that we have proposed in Section 3 that will allow it to derive a unique value for each of the clusters and thus determine whether Anne or Bob are the best ones to attend the emergency.

In the following part, we will show how the Interoperable Trust Repository works in this case. Firstly, we have to analyze the operators' clusters:

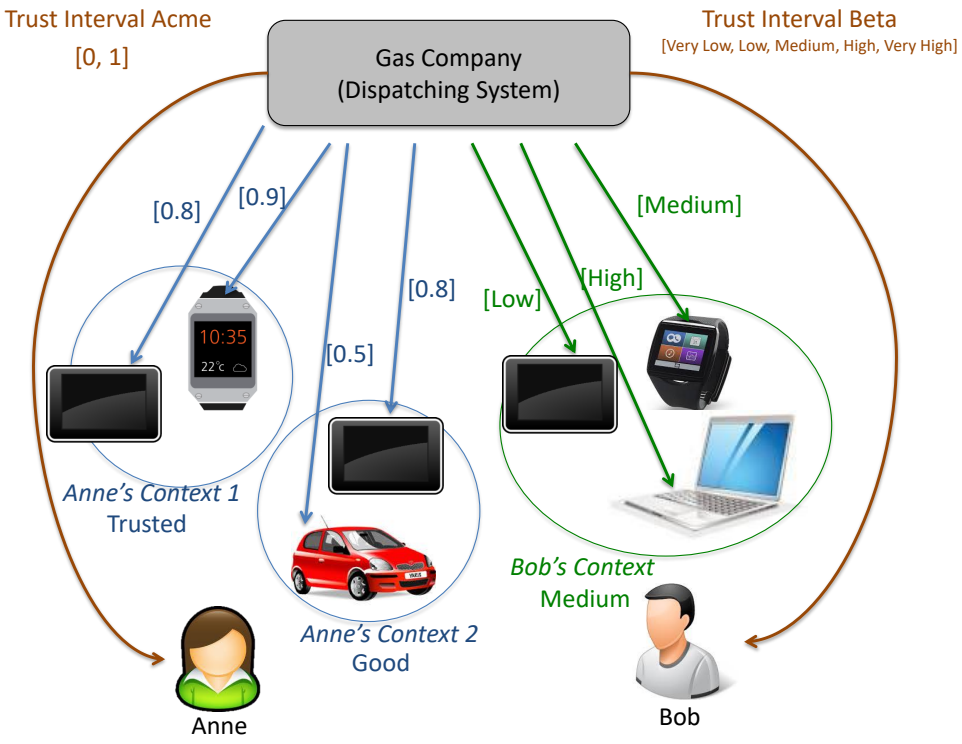

Figure 2: Clusters belonging to the operators

Anne's Cluster 1. The first cluster can be seen on the left of Figure 1. The objects are a smart watch and a tablet. Anne's Company uses evaluation models based on reputation. Thus, as we can observe in Figure 2, for this particular cluster we have two values: $[0.8, 0.9]$. The tablet has a value of $[0.8]$ and the smart watch $[0.9]$. The trust interval used by Anne's Company is $[0.0, 1.0]$. For this reason, we can assume that both devices have high trusted values. Thus, we can preliminary consider the trust value of this cluster as *Trusted*.

Anne's Cluster 2. This second cluster is composed of a smart car and a tablet. Since in both cases the purpose is the same, i.e., to attend the leaking pipe, we can apply the mapping we defined in Section 3. The tablet is the same of cluster 1, so we can consider the same value as before: $[0.8]$. On the other hand, we know that the smart car has been recently repaired due to several malfunctions. This situation has affected its trust value, so even if it has been repaired, its trust value

is now $[0.5]$. We know that the trust interval used by Anne's Company is $[0.0, 1.0]$. Therefore, f we consider the medium value of the smart car and the high value of the tablet, we can conclude that this cluster's trust value will be lower than cluster 1, but we can preliminary consider it as *Good*.

Bob's Cluster. It is composed of three different objects, a tablet, a laptop and a smart watch. Each of them has a trust value related to the decision model of Beta Corporation. The trust model has different levels related to trust:

$$[VeryLow, Low, Medium, High, VeryHigh].$$

For the tablet we can see that it is classified as *Low* because it has been recently patched after it has been hijacked. Then, the laptop is considered with a *High* value because it has always performed well the assigned tasks even if the operating system is not the most recent one. Finally, we can observe that the smart watch has a *Medium* value because it has been recently bought and it has never been used before for a task. Comparing the three values, we can preliminary consider a medium level for Bob's cluster.

After the preliminary consideration of the three clusters and their trust levels, we have to pass to the following part and perform the interoperability calculation in order to be translated into the Gas Company trust model. After this step, they will be finally compared in order to find the most trusted candidate between Anne's and Bob's clusters.

5 Validation

In the previous section, we have presented the three clusters and their trust models. In this section, we will perform the translation of the different trust values in order to have common values for achieving a trust decision.

As we mentioned before, the gas company implements an evaluation model, but the ranges of such model are different with respect to the evaluation model considered in Anne's clusters. Let us assume that the range of the Gas Company's evaluation model is $[1, 10]$. Thus, the DS must choose between the best operator, so it must firstly translate the cluster's values into the evaluation model metric used by the gas company.

5.1 Interoperability

In Section 3, we have presented three different cases related to possible translations between models: evaluation to evaluation, decision to evaluation and decision to decision. In this use case, we will present the first and the second case.

Firstly, we will consider Anne's clusters and the conversion of the Acme Limited Company's evaluation model into the gas company evaluation model.

As they both use trust intervals, we have to adapt the first one in order to fit in the second one. Thus, we will have to translate the values related to the $[0.0, 1.0]$ interval to the interval $[1, 10]$. We can say that if we have $[0.0]$ and $[1.0]$, the translated values will be $[1]$ and $[10]$. For the other values, we have to perform a translation in order to maintain the proportionality.

In cluster 1, we have two smart entities: a smart watch having a trust value of $[0.9]$ and a tablet having $[0.8]$. We have to translate them into the Gas Company range.

As we have explained at the end of Section 3.3, we cannot provide a general formula because it strictly depends on the contexts. Thus, in this case we can proceed as follows. We divide both the ranges in 10 parts. We find that the segments have a value of $[0.1]$ in Anne's context and a value of $[0.9]$ in the gas company context.

For this reason, we can translate Anne's cluster 1 values in $[9.1]$ for the smart watch and $[8.2]$ for the tablet.

We perform the same conversion for Anne's cluster 2. We know that the tablet has the same value $[8.2]$. On the other hand, we have to calculate the new value for the smart car that is $[5.5]$.

Finally, we have to perform a decision to evaluation model translation for Bob's cluster. We are using the simplest case of the theoretical model defined in Section 3.1. We know that the Beta Corporation decision model has five levels:

$$[VeryLow, Low, Medium, High, VeryHigh].$$

Thus, we have to divide the gas company trust values into five sub-intervals that could be as follows (we set on the left-hand side the discrete values of the Beta Corporation and on the right-hand side the possible corresponding values for the gas company):

$$\{Very\ Low\} \mapsto \{[1,2.8[\}$$

$$\{Low\} \mapsto \{[2.8,4.6[\}$$

$$\{Medium\} \mapsto \{[4.6,6.4[\}$$

$$\{High\} \mapsto \{[6.4,8.8[\}$$

$$\{Very\ High\} \mapsto \{[8.8,10]\}$$

We consider the medium value for each interval as they are basically the same, in fact, for example, there is no difference in this situation for a value of [3.2] or [4.4] as they both belong to the second level. However, this is a common problem when migrating from a continuous to a discrete range.

Therefore, if we translate the values from the Beta Corporation decision model to the Gas Company evaluation model we have the following values: [3.7] for the tablet, [7.3] for the laptop and [5.5] for the smart watch. Before, they were [*Low*], [*High*] and [*Medium*].

Now, all the clusters values have been translated by using the interoperability model presented in Section 3 to common values belonging to the gas company evaluation model. Thus, the DS can perform a trust decision according to how its trust management framework works.

Even if the main focus of the paper is on how to translate trust values between different ranges and models, for the sake of completeness, we will show in the following subsection how a trust decision can be performed by a trust management framework.

5.2 Reputation and Tasks

After the translation of the three cluster trust values, the DS can now perform the final trust decision.

Resuming, the trust values according to the gas company are the following.

For Anne's cluster 1, we have values of [9.1] for the smart watch and [8.2] for the tablet. Then, for Anne's cluster 2 we have for the smart car the value of [5.5] and [8.2] for the tablet. Finally, for Bob's cluster we have the following values: [3.7] for the tablet, [7.3] for the laptop and [5.5] for the smart watch.

Now, with respect to the trust management performed by the DS we have to consider the following aspects.

Firstly, an operator reputation is computed depending on the smart devices belonging to the same operator. Let T be a thing, w a weight, N the number of carried things by a candidate c and M the number of required things in order to fulfill a task k. The weights are uniformly distributed values. The candidate reputation for a given task could then be defined as shown in the following formula:

$$R_{c,k} = \frac{\sum_{i=0}^{M} w_i T_i}{\sum_{j=0}^{N} w_j T_j}$$

with the following conditions

$$\begin{cases} 0 < M \leq N \leq 3 \\ \sum_i w_i = 1, w_i = 0 \\ T_i \in \{0,1\} \end{cases}$$

Then, the method selects the candidates according to the criticality for the given task. The higher the task criticality is, the more trusted the candidate should be. If the criticality is low, every candidate is accepted to accomplish the task, then the best one will be selected (see the final part of this section). If the criticality is set to medium, only medium and highly trusted candidates are accepted. And if the criticality is high, only highly trusted candidates are selected for the task. An example of the chosen reputation ranges for high, medium and low reputation are displayed in the following Figure 3.

Reputation / Task criticality	[0, 0.4]	]0.4, 0.8]	]0.8, 1]
Low	Yes	Yes	Yes
Medium	No	Yes	Yes
High	No	No	Yes

Figure 3: Reputation according to Task's Criticality

From the previous selected candidates, the described method chooses the best one according to their reputation and their years of experience. The latter is an extra parameter that is used in the case that the reputation of both candidates is the same or very similar.

Thus, we could define a new metric or score that is calculated by using both the experience and the reputation of the candidate. Let R be the reputation of a candidate c, E the years of experience of a candidate c and C the number of candidates for the task k. If there is no candidate, the task is not performed.

The calculated score S for a task k is computed according to the formula

$$S_{c,k} = \frac{R_c \times E_c}{max(R)_{0<c\leq C} \times max(E)_{0<c\leq C}}$$

5.3 Final Calculation

We have developed a code that computes the previous values calculating the final score of an operator. The code has been developed in Java [2] and we have designed it in order to implement the framework proposed in this paper (i.e., trust interoperability). Such code is supposed to be applied in the DS of the scenario considered in Section 4.

Moreover, in order to apply the previous formulas, we have to calculate a reputation value for each cluster of both the operators, as the output of an evaluation model. We apply the formula presented before. Thus, for Anne's cluster 1 we have a global reputation value of [8.35]. Then, for Anne's cluster 2 we have a global reputation of [6.35]. Finally, for Bob's cluster we have a global reputation of [5.2]. We can now insert the data in the code.

- Dispatching System Trust Management Framework -
***//set of things**: [car,tablet,laptop,smartwatch]*

Operator ID: 1, **Name**: Anne, **Completed tasks**: 5, **Years of experience**: 14, **Global reputation**: 8.65, **Availability**: true
Cluster ID: 1, ***set of things**: [null, 1693, null, 424]*

Operator ID: 1, **Name**: Anne, **Completed tasks**: 5, **Years of experience**: 14, **Global reputation**: 6.85, **Availability**: true
Cluster ID: 2, ***set of things**: [1486, 1693, null, null]*

Operator ID: 2, **Name**: Bob, **Completed tasks**: 5, **Years of experience**: 5, **Global reputation**: 5.5, **Availability**: true
Cluster ID: 3, ***set of things**: [null, 2130, 190, 1794]*

In the previous code, we have to explain some important data. Firstly, we have given to each operator an *ID* and we have inserted a field to collect also the *name* of the operator.

[2] https://www.java.com/en/

Then, we have added a parameter that considers the successfully *completed tasks* in a day by an operator. We could consider a maximum number of tasks performed by an operator each day. Then, there is a parameter related to the *years of experience*. This parameter is used only in the case the final global reputation will be similar for both the operators. We do not consider the exact value of reputation in order to compensate a minimum gradient of error due to the translation among trust models (i.e., in the decision to evaluation model framework we decided to consider the medium value for a particular range, but we could consider the higher or the lower).

Then, there is the value *global reputation* that we discussed earlier. Finally, *availability* is a precondition in order to compute the reputation value of an operator. If it is false, the operator will not be considered.

We can observe that the clusters have an ID and they are composed of a set of things. In the case an IoT device is not present, there is a null value. Otherwise, there will be a code that uniquely identifies a device. We can observe that each user has a unique device. In fact, for example, we can see that Anne's tablet has the code 1693, and Bob's tablet has the code 2130.

Considering the final global reputation value computed by the code, Anne will be the chosen one by the DS. Moreover, according to the value of her clusters, the chosen cluster will be the first one.

6 Conclusion

The amount of objects or things that are interconnected in IoT systems makes sometimes difficult to handle interoperability issues when referring to trust models for IoT. In this paper, we have presented a framework to deal with interoperability of trust issues. The main idea behind the framework is to consider an *Interoperability Trust Repository* that deals with the main interoperability facets of IoT such as semantic or syntactic interoperability issues. Moreover, we have performed the validation of the proposed framework with a use case scenario and we have shown that different trust models in different things can interact obtaining common trust values.

In the future, we intend to apply this framework together with the suggested implementation approach to more real IoT use case scenarios such as smart home scenarios. We will also concentrate on the quantification of trust for the different types of models. It will also be interesting to investigate the use of our framework in other decentralized systems different than IoT where the entities involved in it

have to interact and use different trust models. It would also be of interest to investigate how the framework performs when small devices are clustered into larger virtual IoT nodes. Moreover, in the case where there are multi-trusted domains.

Acknowledgments

This work has been supported by the project PID2022-139268OB-I00, funded by the Spanish Ministry of Science and Innovation, the Research State Agency (10.13039/501100011033) and the European Social Fund Plus. Moreover, we thank Huawei Technologies for their support.

References

[1] Roman, R., Najera, P., and Lopez J., "Securing the internet of things" Computer, vol. 44, no. 9, pp. 51–58, 2011.

[2] Al-Ali A.-R., Zualkernan I. A.,Rashid M.,Gupta R., and AliKarar M., "A smart home energy management system using iot and big data analytics approach" IEEE Transactions on Consumer Electronics, vol. 63, no. 4, pp. 426–434, 2017.

[3] Ferraris D., and Fernandez-Gago C., "Trustapis: a trust requirements elicitation method for iot" International Journal of Information Security, vol. 19, no. 1, pp. 111–127, 2020.

[4] Mukherjee A., De D., Dey N., Crespo R. G., and Herrera-Viedma E., "Disastdrone: A disaster aware consumer internet of drone things system in ultra-low latent 6g network" IEEE Transactions on Consumer Electronics, 2022.

[5] Barhoun, R., and Ed-Daibouni, M. "Trust modeling in a distributed collaborative environment: application to a collaborative healthcare system". International Journal of Information Security, 2023, 1-20.

[6] Bowen, J, and Turner, J. "Interactive System Modelling for the Internet of Things". Proceedings of the ACM on Human-Computer Interaction, 2023, 7.EICS: 1-19.

[7] Chen, J., Farooq, J, and Zhu, Q. "QoS Based Contract Design for Profit Maximization in IoT-Enabled Data Markets". IEEE Internet of Things Journal, 2023.

[8] Shukla, A., Katt, and B., Yamin, M. M. "A quantitative framework for security assurance evaluation and selection of cloud services: a case study". International Journal of Information Security, 2023, 1-30.

[9] Gu L., Wang J., and Sun B., "Trust management mechanism for internet of things" China Communications, vol. 11, no. 2, pp. 148–156, 2014.

[10] Iot's interoperability challenge. [Online]. Available: https://www.networkworld.com/article/3205207/internet-of-things/iots-interoperability-challenge.html, 2017.

[11] Fernandez-Gago C., Moyano F., and Lopez J., "Modelling trust dynamics in the internet of things," Information Sciences, vol. 396, pp. 72–82, 2017.

[12] Ebbers, F., and Friedwald, M. "Responses of the European IoT ecosystem to the GDPR. 2023.

[13] Derhamy H., Eliasson J., and Delsing J., "Iot interoperability—on-demand and low latency transparent multiprotocol translator" IEEE Internet of Things Journal, vol. 4, no. 5, pp. 1754–1763, 2017.

[14] Razzaque M. A., Milojevic-Jevric M., Palade A., and Clarke S., "Middleware for internet of things: A survey" IEEE Internet of Things Journal, vol. 3, no. 1, pp. 70–95, 2016.

[15] Castelfranchi C., and Falcone R., "Trust and control: A dialectic link" Applied Artificial Intelligence, vol. 14, no. 8, pp. 799–823, 2000

[16] Ferraris D., Fernandez-Gago C., and Lopez J., "A trust-by-design framework for the internet of things" in 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS). IEEE, 2018, pp. 1–4.

[17] Patel, D., Anand, H., Chakraborty, S. "CrossTrustchain: Cross-Chain Interoperability using Multivariate Trust Models". In: 2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS). IEEE, 2023. p. 129-134.

[18] Wenjuan L., and Ping, L., "Trust model to enhance security and interoperability of cloud environment". Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings 1. Springer Berlin Heidelberg, 2009. p. 69-79.

[19] Saadi, Rahaman M. A., Issarny V., and Toninelli A., "Composing trust models towards interoperable trust management". IFIP International Conference on Trust Management. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 51-66.

[20] Ganzha M., Paprzycki M., Pawlowski W., Szeja P., and Wasielewska K., "Semantic interoperability in the internet of things: an overview from the inter-iot perspective" Journal of Network and Computer Applications, vol. 81, no. 111-124, 2017.

[21] Ullah F., Habib M. A., Farhan M., Khalid S., Durrani M. Y., and Jabbar S., "Semantic interoperability for big data in heterogeneous iot infrastructure for healtcare" Sustainable Cities and Society, vol. 34, pp. 90–96, 2017.

[22] Yan Z., Zhang P., and Vasilakos A. V.", "A survey on trust management for internet of things" Journal of Network and Computer Applications, vol. 42, no. 120-134, 2014.

[23] Guo J., "Trust-based service management of internet of things systems and its applications" Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2018.

[24] Al-Hamadi H., and Chen I., "Trust-based decision making for health iot systems" IEEE Internet of Things, vol. 4, no. 5, pp. 1408–1419, 2017.

[25] Sharma A., Pilli E. S., Mazumdar A. P., and Govil M. C., "A framework to manage trust in internet of things" in 2016 International Conference on Emerging Trends in Communication Technologies (ETCT), November 2016, pp. 1–5

[26] Bao F., and Chen I.-R., "Trust management for the internet of things and its application to service composition" in IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM). IEEE, 2012, pp. 1–6.

[27] Bao F., and Chen I.-R., “Dynamic trust management for internet of things applications” in International Workshop on Self-aware Internet of Things. ACM, 2012, pp. 1–6.

[28] Atzori L., Iera A., Morabito G., and Nitti M., “The social internet of things (siot) – when social networks meet the internet of things: Concept, architecture and network characterization” Computer Networks, vol. 56, pp. 3594–3608, 2012

[29] Nitti M., Atzori L., and Cvijikj I., “Friendship selection in the social internet of things: Challenges and possible strategies” IEEE Internet of Things, vol. 2, no. 3, pp. 240–247, 2015.

[30] Marche, C., Soma, G. G., and Nitti, M. "A Cognitive Social IoT Approach for Smart Energy Management in a Real Environment". IEEE Transactions on Network and Service Management, 2023.

[31] Zouzou, M. C., Benkhelifa, E., Kholidy, H., and Dyke, D. W. "Multi-Context-aware Trust Management framework in Social Internet of Things (MCTM-SIoT)". In 2023 International Conference on Intelligent Computing, Communication, Networking and Services (ICCNS) (pp. 99-104), IEEE, 2023

[32] Chen D., Chang G., Sun D., Li J., Jia J., and Wang X., “Trm-iot: A trust management model based on fuzzy reputation for internet of things” Computer Science and Information Systems, vol. 8, no. 4, pp. 1207–1228, 2011

[33] Saied Y. B., Olivereau A., and Zeghlache D., “Trust management system design for the internet of things: A context-aware and multi-service approach” Computers & Security, vol. 39, pp. 351–365, 2013

[34] Moyano, F., Fernandez-Gago, C.; Lopez, J. "A conceptual framework for trust models" International Conference on Trust, Privacy and Security in Digital Business. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 93-104.

[35] Blaze M., Feigenbaum J., and Lacy J., “Decentralized Trust Management” in Proceedings of the 1996 IEEE Symposium on Security and Privacy. IEEE Computer Society Press, 1996, pp. 164–173

[36] Marsh S., “Formalising trust as a computational concept” Ph.D. dissertation, University of Stirling, April 1994

[37] Moyano F., Fernandez-Gago C., and Lopez J., “A model-driven approach for engineering trust and reputation into software services” Journal of Network and Computer Applications, vol. 69, pp. 134–151, 2016

[38] Yan Z., and Holtmanns S., “Trust modeling and management: from social trust to digital trust,” Computer Security, Privacy and Politics: Current Issues, Challenges and Solutions. IGI Global, pp. 290–323, 2008.