Policy-based Usage Control for A Trustworthy Data Sharing Platform in Smart Cities

Quyet H. Cao[a,*], Madhusudan Giyyarpuram[a], Reza Farahbakhsh[b], Noel Crespi[b]

[a] *Orange Labs, France*
[b] *Institut Mines-Telecom, Telecom SudParis, CNRS Lab UMR5157, France*

Abstract

Although data is a key part in smart cities, traditionally there has been no systematic effort to enable the sharing of data in a trustworthy manner among applications or services. In order to promote sharing of data, mechanisms need to be put into place to provide the different actors - data producers, data consumers, etc. means to control and visualize how their data or requests are being processed and used. In this paper we deal with a key issue involved in trust which is usage control, i.e., how data is used once access to it has been granted. We propose a Data Usage Control Model (*DUPO*) to capture the diversity of obligations and constraints that data providers impose on the use of their data. Based on the *DUPO* model and semantic technologies, we propose a trustworthy data sharing platform which enhances transparency and traceability of data usage in smart cities. Lastly a proof-of-concept is developed to evaluate our solution and results show that the performance of the added trust does not impact negatively on the system.

Keywords: Internet of Things, Smart Cities, Privacy, Trust Model, Data Sharing, Data Usage Control and Policy, Data Protection.

[*] Corresponding author. Tel.:+33 4 38 42 81 32
Email addresses: quyet.caohuu@orange.com (Quyet H. Cao), giyyarpuram.madhusudan@orange.com (Madhusudan Giyyarpuram), reza.farahbakhsh@it-sudparis.eu (Reza Farahbakhsh), noel.crespi@it-sudparis.eu (Noel Crespi)

1. Introduction

In smart cities, the Information and Communication Technologies (ICT) are generally integrated into traditional services of our city to improve quality while reducing costs of these services [1]. Nowadays, the new communication paradigm goes beyond traditional inter-personal interactions, as it involves interactions between devices under the umbrella of the Internet of Things (IoT)[2, 3] technologies which are among the main vehicles for realizing this vision. IoT data can be collected from huge amount of interactions across a large number of devices, and in the near future, large scale IoT applications in smart cities will become a reality. It could enhance a city's innovation capacity as well as provide significant socioeconomic value for the cities[4]. In deploying such applications, the participation of citizens and other players in both data collection and in the emergence of new services is needed.

Currently, applications for smart cities are mostly developed in a vertical manner, with no sharing of data or resources between different players [5]. Many of these vertical applications would benefit from using information sources of different origins to enhance their own services. The landscape consists of a diversity of actors, both public and private, who provide a large variety of services. These applications include energy management for public buildings, waste management, public lighting, mobility management, intelligent parking solutions and a whole range of new services that are being conceived for smart cities [6]. The actors involved in these applications tend to vary with the specific domain, as each comes with its own ecosystem. However we can identify several broad categories of actors: institutional actors (such as districts, municipalities), equipment manufacturers, network operators, infrastructure providers and service providers. With the development of the IoT, the range of actors involved will be enlarged to include micro companies, value-added service providers (such as aggregations, compositions and mashups) and end users. The need for a horizontal platform, which federates information from these disparate sources is particularly important. This intermediation platform, for actors with different and sometimes contradictory requirements brings its own set of challenges. For this horizontal approach to succeed, the platform needs to ensure that the business interests of the different participants are fully honored.

The main requirement to have a successful IoT data sharing in this context of an intermediation platform is that participants contribute and share their data. One example is when people are able to share their data related to different events by leveraging the sensing capabilities of their smartphones. This crowd-sensing is a recent trend [7] and may soon outperform traditional data collection methods such as using pre-installed sensors. However, crowd-sensing may involve privacy issues for device owners. For example, some of the data collected by smartphones may contain sensitive information such as the location of the owners. In addition, the data in smart cities may come from a variety of sources and potentially undergo several transformations, such as aggregation and composition, before reaching their final destination. The IoT data may also be shared for common usage through linked data sets such as Linked Open Data [8]. Therefore, to achieve trustworthy data sharing in smart cities, the shared platform should be able to: (*i*) establish the trust between different players to share their data, (*ii*) solve a potential conflict of interest between actors, (*iii*) achieve competitive advantages, and (*iv*) hide or abstract some information with usage control.

Trust has many facets, but one critical element in the IoT is the ability for each participant to exercise control on how their data is going to be used. Although this is an important research topic, but still it has not yet been treated in a proper manner in the context of smart cities. Thus, this study aims to deal with this key issue of trust and control for the intermediation platform by development of a policy-based usage control approach. In particular, the main contributions of our study are three-fold:

(*i*) First, we propose a Data Usage Control Model (*DUPO*) to capture the diversity of obligations and constraints that data owners impose on the use of data. It takes into account of the major data usage requirements such as spatio-temporal granularity, abstraction/masking of certain information, conditions depending upon the class of actor/purpose, and the monetization of data. The conceptual model, the formal theory based on Defeasible Logic (DL), and illustrative scenario are presented;

(*ii*) Base on *DUPO* and semantic technologies we define the framework which enhances data usage transparency and traceability in the context of an intermediation platform for smart cities. It includes core components for data usage control in perspectives of data providers, data consumers, and IoT intermediation platform. We also illustrate procedures for trustworthy data sharing in the platform.

(*iii*) Finally, a proof-of-concept is developed, its implementation choices and a visualization tool prototype which help users to control and monitor easily how their data is shared. We then do a preliminary performance and comparison analysis for the proposed solution.

2. Scenario and Requirements

To illustrate better the current issues of trust and control for data sharing cases in smart cities, we first present a general motivating scenario with a use case for intelligent parking, and then raise some research questions that will be addressed through this study.

2.1. Smart Cities Data Sharing Scenario

Figure 1 shows our overall smart cities motivating scenario. Various sensors are deployed for sensing data in cities by service providers or citizens. We have different applications or services which may share their data or resources. Examples of such services include intelligent parking solutions, waste management, public lighting, air quality monitoring, and crowd-participatory sensing applications. A shared platform, which may be provided by an operator, will be used by the diverse applications. In this platform, a data usage control module is needed to deal with issues of trust and control. This module allows data providers to exercise some control over the generated data by their sensors and

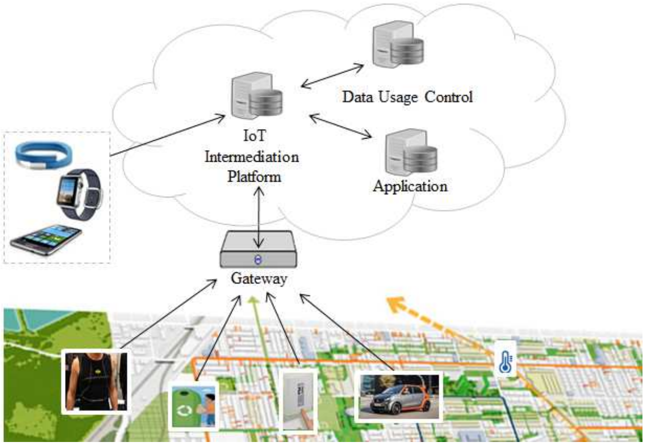


Figure 1: An overall schema of a Data Sharing Scenario in Smart Cities.

ensure that the policies put in place by the data producers are respected by data consumers.

We use a context of an Intelligent Parking Application (IPA) to demonstrate our motivation. This application has three main use cases: (*i*) monitoring data parking places, (*ii*) unexpected usage by data consumers, and (*iii*) observations of data usage. The generated sensor data is used not only by this application but also by other applications. Data owners therefore must define data usage policies to control the usage of their data. We have different data consumers such as municipal authorities, application developers and commercial operators. They can request to access data at the granularity and scope that the data owners has specified. We present examples of the data usage policies in our scenario as follows:

1) The data owner (the company that deploys and is the owner of the parking sensors) will have full access to all the details generated by all the individual parking sensors.

2) The data owner is willing to make available the average occupancy of parking places per street on an hourly basis to municipal authorities.

3) However, the data owner will only offer commercial service providers statistical data, only per zone and only on a weekly basis.

4) The monetization of data is allowed, based on subscription type or on a user's role, for example.

2.2. Requirements reflected as our Research Questions

The main requirement in our scenario is about data usage control: How data is used after access to it has been granted? It is related to two main research questions. 1) How do data owners define their data usage policies? and 2) How do the platform ensure that these policies are enforced correctly?

In particular, the first one will focus on following aspects: (*i*) What are the main criteria to define the policies? (*ii*) How do we deal with potential conflict between dependent policies? and (*iii*) How do data owners exercise some control over the usage of their data?. For the second one, we must deal with: (*i*) How do the platform process the data consumers' request and offer an explanation when the request is refused? (*ii*) How does the platform trace data usage? and (*iii*) How do data owners customize their policies and explore the consequences of certain changes?

3. Data Usage Control Model

This section introduces the proposed model for data usage control, namely *DUPO*. We first introduce its conceptual model, formal theory, and then an illustrative scenario is presented.

3.1. Conceptual Model

Figure 2 presents the conceptual view of *DUPO*. As it shows, we can define a data usage policy which will be attached to a set of data items. The policy is created by using modal operators (including: Obligation, Prohibition, and Permission) and data usage conditions including: (*i*) class of actors, (*ii*) granularity (Spatiality, Temporality, and Aggregation), (*iii*) class of purposes, and (*iv*) monetization constraints. Its naming, life cycle, and priority are also managed. Next, we focus on the aspects of data usage transparency and traceability and explain in detail the concepts behind *DUPO*.

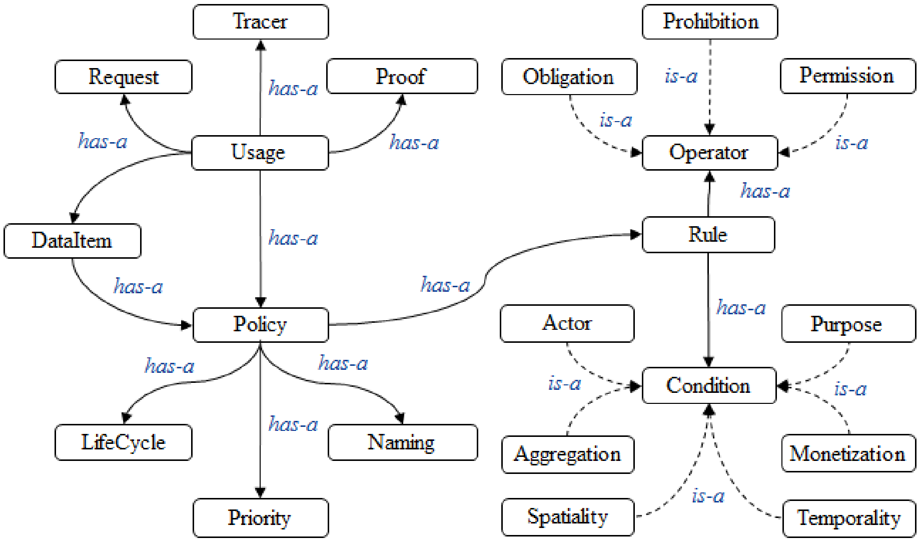


Figure 2: Conceptual view of the DUPO

3.1.1. Data Items

A *Data Item* is an individual of the *Context Element* based on the Information Model [9] standard specification which is used in the European Project FI-WARE [10] for Context Management. An *Entity Element* is a container used to exchange information about an entity. It contains the following information: (i) an entity ID including the name and the type, (ii) a list of the entity attributes, (iii) (optionally) the name of an attribute domain that logically groups together a set of entity's attributes, and (iv) (optionally) a list of metadata that applies to all the attribute values of the given domain. We formally define a *Data Item* by using XML DTD, as mentioned in Listing 1.

```
<!DOCTYPE DUPO[
<!ELEMENT DataItem(EntityElement)>
<!ELEMENT EntityElement(EntityID, AttributeDomainName?,
    EntityAttributeList, DomainMetadata?)>
<!ELEMENT EntityID(Id, Type)>
<!ELEMENT EntityAttributeList(EntityAttribute*)>
<!ELEMENT EntityAttribute(Name, Type, EntitytValue,
    EntityMetadata+)>
<!ELEMENT DomainMetadata(EntityMetadata*)>
<!ELEMENT EntityMetadata(Name , Type , Value)>
...
]>
```

Listing 1: XML DTD Definition of Data Item.

3.1.2. Conditions

The Condition contains (optionally) the following expressions: (*i*) Spatio-Temporal Granularities, (*ii*) Aggregation Granularities, (*iii*) Conditions by Actors, (*iv*) Conditions by Purposes, and (*v*) Conditions of Monetization. We formally define conditions by using XML DTD, as shown in Listing 2.

```
<!DOCTYPE DUPO[
<!ELEMENT Condition(Temporality?, Spatiality?, Aggregation?,
    Actor?, Purpose?, Monetization?)>
<!ELEMENT Spatiality(SpatialScope*)>
<!ELEMENT Temporality(TemporalScope*)>
<!ELEMENT Aggregation(AggregateScope*)>
<!ELEMENT Actor(ActorScope*)>
<!ELEMENT Purpose(PurposeScope*)>
<!ELEMENT Monetization(MonetizationScope*)>
<!ELEMENT TemporalScope(Secondly?, Minutly?, Hourly?, Daily?,
    Weekly?, Monthly?, Yearly?, Any?)>
<!ELEMENT SpatialScope(Street?, Zone?, Any?)>
<!ELEMENT ActorScope(DataOwner?, MulnicipalAuthority?,
    ComercicalServiceProvider?)>
<!ELEMENT AggregateScope(Detail?, Average?, Statistic?, Any?)>
<!ELEMENT PurposeScope(CommercialUse?, Any?)>
<!ELEMENT MonetizationScope(Level?, Any?)>
...
]>
```

Listing 2: XML DTD Definition of Condition.

3.1.3. Operators

An Operator contains (optionally) model operators: (*i*) Obligation (*ii*) Prohibition, and (*iii*) Permission. The formal definitions are created using XML

DTD as presented in Listing 3.

```
<!DOCTYPE DUPO[
<!ELEMENT Operator(Obligation?, Prohibition?, Permission?)>
...
]>
```

Listing 3: XML DTD Definition of Operator.

3.1.4. Policies

A *Policy* has its name, lifecycle, priority, and a collection of rules which is created by defining the *Operator* on the individual *Condition.* Listing 4 formally defines the policy using XML DTD.

```
<!DOCTYPE DUPO[
<!ELEMENT Policy(Name, LifeCycle, Priority?, Rule*)>
<!ELEMENT Name(URI?)>
<!ELEMENT LifeCycle(Duration?, Datetime?)>
<!ELEMENT Rule(Operator?, Condition?)>
...
]>
```

Listing 4: XML DTD Definition of Policy.

3.1.5. Usage

An *Usage* is created by a consumer's request, related policies, and response data. The data could be a proof justification, a tracked data usage, or a list of returned data items. This component has a purpose for transparency and traceability of the data usage. We formally define the Usage using XML DTD in Listing 5.

```
<!DOCTYPE DUPO[
<!ELEMENT Usage(Request, Policy*, Data*)>
<!ELEMENT Request(Rule?)>
<!ELEMENT Data(Tracker?, Proof?, DataItem*)>
...
]>
```

Listing 5: XML DTD Definition of Usage.

3.2. Formal Theory

Formal theory of the *DUPO* is based on the general concept of DL, which is a non-monotonic formalism that deals with incomplete and conflicting information, originally proposed by Nute [11]. In particular, we build on earlier works extending DL with modal and deontic operators, as presented in Governatori [12][13] and Antoniou [14][15]. Deontic logic is concerned with concepts of obligations, permissions and prohibitions, allowing such relationships to be captured with each entity. There are some proposed formalisms for dealing with reasoning, handling and solving the normative conflicts that arise between rules and exceptions. However, DL is one of the best solutions which can manage all these aspects in an efficient and computationally tractable way [13]. Moreover, DL offers enhanced representational capabilities and low computational complexity [16]. According to [12], when DL is enriched with modal deontic operators, the complexity does not increase in most cases.

We define the DUPO theory and its proof as follows.

3.2.1. DUPO Theory

Let $PROP$ be a set of propositional atom. A set of literals $Lit = PROP \cup \{\neg p | p \in PROP\}$. Let $MOD = \{O, P, F\}$ be the set of basic deontic modalities (Obligation, Permission, and Forbiddance/Prohibition). A set of modal literals $ModLit = \{[X]l, \neg[X]l | l \in Lit, X \in MOD\}$.

Let *Lbl* be a set of arbitrary labels. *R* is a set of base and deontic rules. A base rule is expressed as $r : A(r) \hookrightarrow C(r)$, while a deontic rule is $r : A(r) \hookrightarrow_X C(r)$, where (*i*) A unique label $r \in Lbl$, (*ii*) The antecedent (or body) $A(r) = a_1, ..., a_n$, $a_i \in Lit \cup ModLit, 1 \leq i \leq n$; (*iii*) An arrow $\hookrightarrow \in \{\rightarrow, \Rightarrow, \rightsquigarrow\}$, denotes the type of rules: strict rules, defeasible rules and defeaters, respectively, (*iv*) $X \in MOD$, and (*v*) The consequent (or head) $C(r) = b, b \in Lit$.

The different rules have the following meaning. Strict rules can never be defeated, while defeasible rules can be defeated by contrary evidence. Defeater rules are only used to prevent certain conclusions.

Definition 1. *A theory DUPO* $= (F^{DUPO}, R^{DUPO}, >)$*, where* $i)F^{DUPO} \subseteq Lit \cup ModLit$ *is a finite set of facts,* $ii)R^{DUPO} \subseteq R$ *is a finite set of rules and* $iii)$ $>$ *is a superiority relation for priorities among the non-strict rules in* R^{DUPO}*.*

3.2.2. Theory Proof

A conclusion derived from *DUPO* is a tagged literal and it is classified as follows: $+\Delta q$ means that literal q is definitely provable in $DUPO$; $-\Delta q$ means that literal q is definitely rejected in $DUPO$; $+\partial q$ means that literal q is defeasibly provable in $DUPO$; and $-\partial q$ means that literal q is defeasibly rejected in $DUPO$.

A proof $P = (P(1), ..., P(n))$ in D is a finite sequence of tagged literals of type $+\Delta q$, $-\Delta q$, $+\partial q$ and $-\partial q$.

We denote the set of all strict rules in R by R_s, R_{sd} for the set of strict and defeasible rules, and $R[q]$ for the set of rules whose head is q. $P[1..i]$ denotes the initial part of the sequence of length i. The proof conditions for the conclusions are formally defined as follows [14][15]

$+\Delta : If\ P(i+1) = +\Delta q\ then\ either$

$\quad (1)\ q \in F\ or$

$\quad (2)\ \exists r \in R_s[q] \forall a \in A(r) : +\Delta a \in P[1..i].$

$-\Delta : If\ P(i+1) = -\Delta q\ then$

$\quad (1)\ q \notin F\ and$

$\quad (2)\ \forall r \in R_s[q] \exists a \in A(r) : -\Delta a \in P[1..i].$

$+\partial :\ If\ P(i+1) = +\partial q\ then\ either$

$\quad (1){+}\Delta q \in P[1..i]\ or$

$\quad (2)(2.1) \exists r \in R_{sd}[q] \forall a \in A(r) : +\partial a \in P[1..i]\ and$

$\qquad (2.2)\text{-}\Delta \neg q \in P[1..i]\ and$

$\qquad (2.3) \forall s \in R[\neg q]\ either$

$\qquad\quad (2.3.1) \exists a \in A(s) : -\partial a \in P[1..i]\ or$

$\qquad\quad (2.3.2) \exists t \in R_{sd}[q]\ such\ that$

$$\forall a \in A(t) : +\partial a \in P[1..i] \ and \ t > s.$$

$$-\partial : \ If \ P(i+1) = -\partial q \ then$$

$$(1) {-\Delta q} \in P[1..i] \ and$$

$$(2)(2.1) \forall r \in R_{sd}[q] \exists a \in A(r) : -\partial a \in P[1..i] \ or$$

$$(2.2) {+\Delta \neg q} \in P[1..i] \ or$$

$$(2.3) \exists s \in R[\neg q] \ such \ that$$

$$(2.3.1) \forall a \in A(s) : +\partial a \in P[1..i] \ and$$

$$(2.3.2) \forall t \in R_{sd}[q] \ either$$

$$\exists a \in A(t) : -\partial a \in P[1..i] \ or \ t \not> s.$$

Theory proof is used as an efficient method for reasoning consumer's requests in the DUPO.

3.3. Illustrative Scenario

To explain more the DUPO, we consider an example that a commercial service provider requests all the parking data details of a street on an hourly basis. We already have a data usage policy that states commercial service providers are only permitted access to statistical data over a zone, and that on a weekly basis in Section 2.1. Thus, this consumer's request is refused with a proof justification. Otherwise, the related data items will be returned and data usage is tracked. This example basically covers the usage control requirements and related concepts in the DUPO:

$$\begin{aligned}
Actor &= (CommercialServiceOperator),\\
Aggregation &= (Detail, StatisticalData),\\
Spatiality &= (StreetLevel, ZoneLevel),\\
Temporality &= (Hourly, Weekly),\\
Operator &= (Obligation, Prohibition, Permission),\\
Proof &= (ProofJustification),\\
Track &= (TrackedDataUsage).
\end{aligned}$$

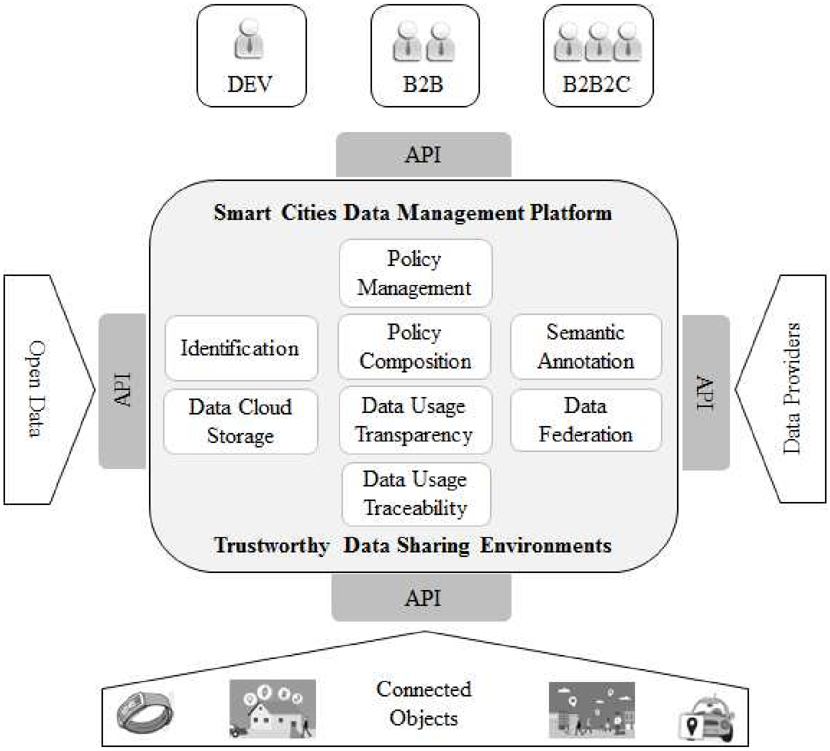


Figure 3: Overall Platform for Smart Cities Data Management

In the next part, we will use this illustrative scenario to explain more how *DUPO* works on the proposed platform.

4. Trustworthy Data Sharing Platform

This section presents a novel intermediation platform for smart cities data management. We aim to provide a trustworthy data sharing environment by enhancement of data usage transparency and traceability.

4.1. Overall Platform

Figure 3 shows an overall overview of the proposed platform. As shown in the figure, we have four groups that are connected to this platform as follow: 1) the connected objects, which can be special sensors or users' mobile phones.

2) the data providers of historical records, additional data sets, etc., 3) public data sources, which are open, e.g. calendars, directories, etc., and 4) the array of business applications and developers accessing this platform, all using the shared data.

These are an ecosystem of developers that wish to exploit the data for commercial services or they can be government agencies charged with providing improved citizens services. Developers are able to ascertain data availability and the conditions of data usage, so they can quickly and reliably assess the feasibility of their intended development.

The platform is built on the principles of our system architecture [17] for smart cites. However, we focus only on the platform layer of the architecture and propose the platform as a service (PaaS). Other aspects of the architecture, such as the infrastructure layer (IaaS) and the application layer (SaaS) in the cloud computing paradigm, are out of the scope in this study and is a potential future direction. This platform is centralized computing and it includes main components and procedures that are developed based on *DUPO* concepts and semantic technologies. In fact, we have added the core components APIs (Application Programming Interface) to allow the transparency and traceability of data usage, and support collaboration between the participants and interoperability of the services in the platform. The platform thus deals with issues of trust and control, and achieves competitive advantages to attract partners sharing their data using the open standard APIs.

4.2. Data Usage Control Components

Figure 4 introduces data usage control components and relationships between them in three perspectives: data providers, data consumers, and the intermediation platform.

4.2.1. Data Providers

The data providers are able to publish their data to the intermediation platform. They are also provided with an editor which they can define the

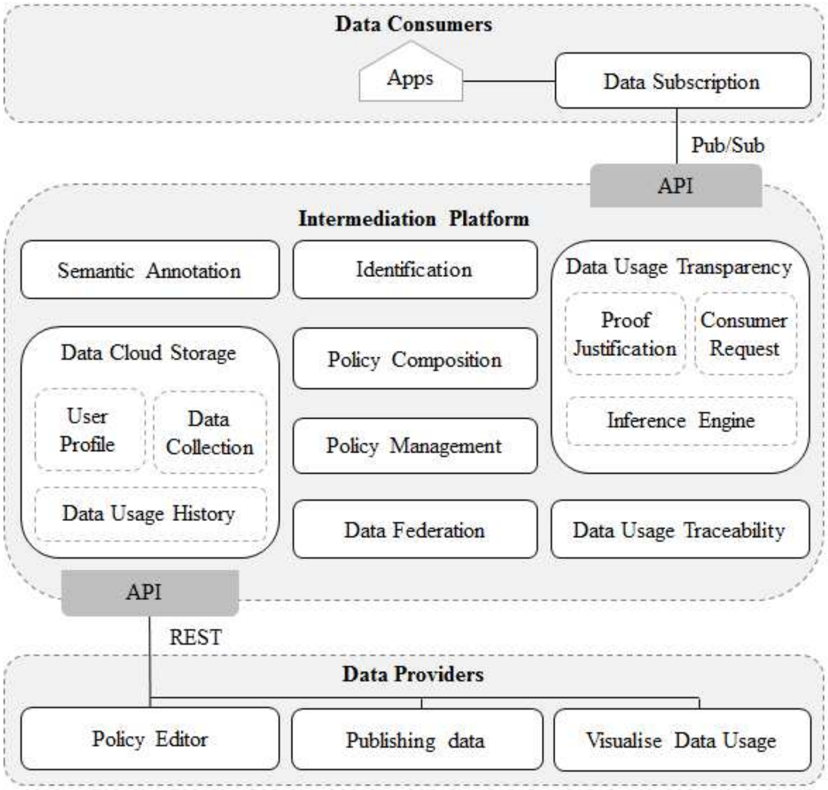


Figure 4: Data Usage Control Components

policy to exercise control on how the data is going to be used.

We develop REST APIs which cover all needed functionalities for the data providers. The APIs are based on a subset of the principles of REpresentational State Transfer (REST) [18], and are used by the data providers to manage their data items, policy/rules, and data usage history in the intermediation platform.

4.2.2. Data Consumers

Data consumers are allowed to request the data from the intermediation platform. They can visualize not only the responded data, but also the proof justification for trusting the results. Moreover, the stated obligations imposed by the data providers are reassured.

Pub/Sub APIs are developed to cover all the needed functionalities for the data consumers. The APIs are based on the principles of Publish/Subscribe which is a highly-decoupled distribution model, where (generally) publishers produce information irrespective of consumers [19]. In particular, the data consumers are provided the APIs for data subscriptions, and proof justification from the intermediation platform.

4.2.3. Intermediation Platform

The platform aims to provide a trustworthy data sharing by enhancement of data usage transparency and traceability. We have several functionalities to ensure this goal as follow: (*i*) Identification of users' profile with reliable authentication, (*ii*) Policy Management for managing the defined policies for data usage, (*iii*) Policy Composition for defining the data usage policies and importing them at the platform level, (*iv*) Transparency for the fair processing of consumers requests, proof justification, and inference engine, (*v*) Traceability for tracing data usage history. It has other components that support (*vi*) Semantic Annotation, (*vii*) Data Cloud Storage for managing user profiles, data collection, and data usage history, and (*viii*) Data Federation for computing consumers' data response.

4.3. Trustworthy Data Sharing Procedures

Figure 5 presents the trustworthy data sharing procedures, which shows the sequence of steps between data provider and consumer. Next we present the detail of the procedures with an illustrative scenario in the following parts:

4.3.1. Identification

As the first step, granting access to the platform is required. In the steps (1) and (6) of the figure 5, the data providers and consumers must create their accounts in the platform. After they are authenticated in steps (2) and (7), they have a secure access to the platform and use the APIs provided. These accounts are stored as user profiles on the Data Cloud Storage.

Mapping to the defined concepts of *DUPO*, the user profiles are facts about *actors*. In our scenario, we have known facts about commercial service operators

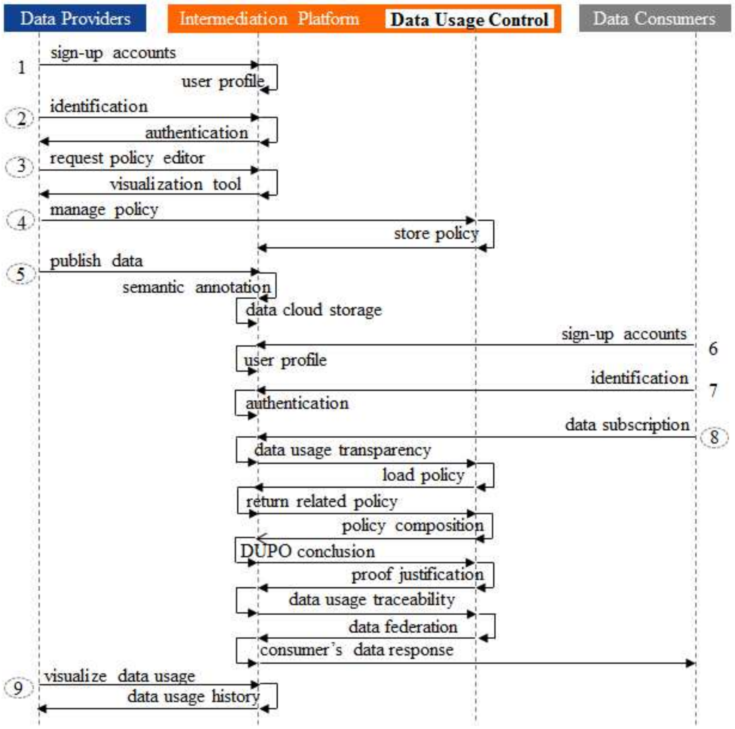


Figure 5: Trustworthy Data Sharing Procedures

(CO), data owners (DO), and municipal authorities (MA), which are presented as follows:

$$F^{DUPO} = \{CO(X), DO(X), MA(X)\}$$

4.3.2. Policy Management

In order to manage the policy in the platform in step (4), we provide a visualization tool and the authenticated data providers need to request it in step (3).

Mapping to the concepts of $DUPO$, we illustrate all of the data usage policies in the scenario as are defined in (R^{DUPO}). In particular, the DO has a full access permission to all the details. This policy is represented with the use of defeasible rules, as follows:

$$\begin{aligned}
&r_{1,d} : DO(X) \Rightarrow_P TemporalScope(X, any),\\
&r_{2,d} : DO(X) \Rightarrow_P SpatialScope(X, any),\\
&r_{3,d} : DO(X) \Rightarrow_P AggregateScope(X, any),\\
&r_{4,d} : DO(X) \Rightarrow_P PurposeScope(X, any)
\end{aligned}$$

The MA has permission to access the available average occupancy of parking places (*average*) per street on an hourly basis. This policy is represented with the use of defeasible rules, as follows:

$$\begin{aligned}
&r_{1,m} : MA(X) \Rightarrow_P SpatialScope(X, street),\\
&r_{2,m} : MA(X) \Rightarrow_F \neg SpatialScope(X, street),\\
&r_{3,m} : MA(X) \Rightarrow_P TemporalScope(X, hourly),\\
&r_{4,m} : MA(X) \Rightarrow_F \neg TemporalScope(X, hourly),\\
&r_{5,m} : MA(X) \Rightarrow_P AggregateScope(X, average),\\
&r_{6,m} : MA(X) \Rightarrow_F \neg AggregateScope(X, average)
\end{aligned}$$

For CO, the consideration is that only statistical data will be available over a zone and on a weekly basis. This policy is represented with the use of defeasible rules, as follows:

$$\begin{aligned}
&r_{1,c} : CO(X) \Rightarrow_P SpatialScope(X, zone),\\
&r_{2,c} : CO(X) \Rightarrow_F \neg SpatialScope(X, zone),\\
&r_{3,c} : CO(X) \Rightarrow_P TemporalScope(X, weekly),\\
&r_{4,c} : CO(X) \Rightarrow_F \neg TemporalScope(X, weekly),\\
&r_{5,c} : CO(X) \Rightarrow_P AggregateScope(X, statistic),\\
&r_{6,c} : CO(X) \Rightarrow_F \neg AggregateScope(X, statistic)
\end{aligned}$$

4.3.3. Publishing Data

The platform supports collection and securing storage of IoT data. In fact, data providers use REST APIs to publish their data in step (5) and the collected data will be stored in the Data Cloud Storage.

Mapping to concepts of *DUPO*, we present an example of data item using Context Element XML format in Listing 6. This data item contains the current state (*line* 9) of the parking sensor (*line* 3) in location (*line* 14) at timestamp (*line* 21).

```
 1 <contextElement>
 2   <entityId type="ParkingSensor" >
 3     <id>ps1</id>
 4   </entityId>
 5   <contextAttributeList>
 6     <contextAttribute>
 7       <name>currentState</name>
 8       <type>integer</type>
 9       <contextValue>1</contextValue>
10     </contextAttribute>
11     <contextAttribute>
12       <name>location</name>
13       <type>string</type>
14       <contextValue>parkingspace1</contextValue>
15     </contextAttribute>
16   </contextAttributeList>
17   <domainMetadata>
18     <contextMetadata>
19       <name>timestamp</name>
20       <type>dateTime</type>
21       <value>2016-02-16T15:23:17.234+0200</value>
22     </contextMetadata>
23   </domainMetadata>
24 </contextElement>
```

Listing 6: Example of Data Item in XML format.

4.3.4. Data Subscription

For the data consumers, they could subscribe to data usage in step (8). We implement the data usage transparency and traceability in the platform based on processing the consumer's request. Toward this end, the Data Usage Transparency component will load the related policies, perform a policy composition, deal with policy conflicts, and do policy enforcement based on defeasible reasoning to obtain the *DUPO* conclusions. In the case that the conclusion is defeasible provable, the Data Federation component will compute to return related data items. The data is filtered or aggregated following the request conditions and the rules extracted from the policy to return the results to the consumers. Every transaction of data usage will be stored as a new data items and later reported to the data owners. The Data Usage Traceability component ensures the traceability of the data usage. In other case, we provide proof justification to the consumer.

For mapping to the *DUPO*, we define the consumer's request in our scenario as a defeasible rule:

$$\begin{aligned} r : CO(X), [P]SpatialScope(X, street), \\ [P]TemporalScope(X, hourly), \\ [P]AggregateScope(X, detail) \\ \Rightarrow_O ConsumerRequest(X) \end{aligned}$$

This consumer's request is processed in the *DUPO* and the conclusions are $-\Delta[O]ConsumerRequest(X)$, and $-\partial[O]ConsumerRequest$. Which means that $ConsumerRequest$ is defeasible rejected in *DUPO*, so the request is refused. We then apply [20] to provide a proof justification to the consumer.

4.3.5. Visualize Data Usage

The data providers could visualize their data usage in step (9), customize their policies, and explore the consequences of certain changes.

5. Implementation and Evaluation

This section presents the proof-of-concept, its implementation choices, a visualization tool prototype, and evaluate the performance of the proposed solution by means of some experiments conducted using the prototype.

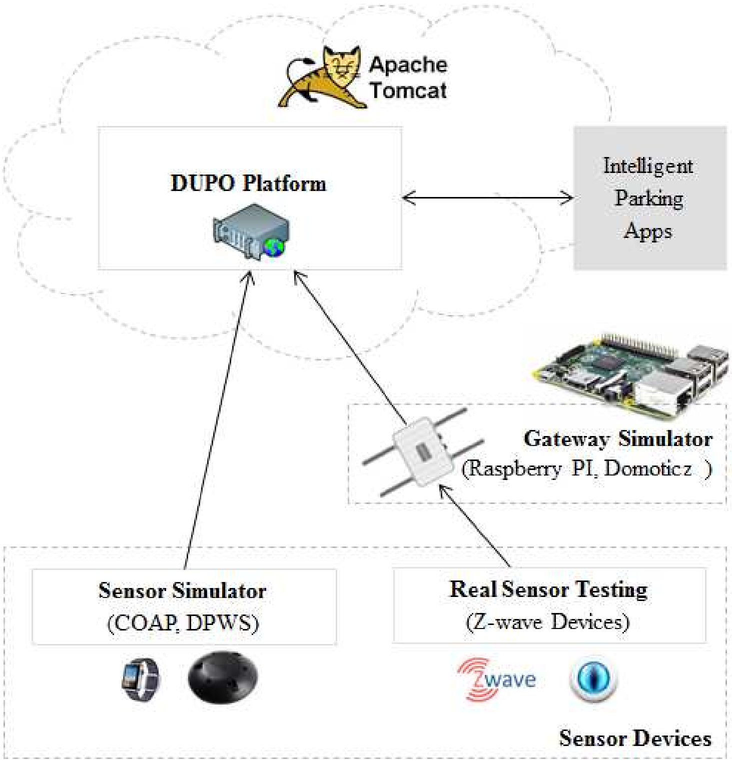


Figure 6: Overview of Proof-of-Concept

5.1. Overall Proof-of-Concept

We define an overall implemented system for the proof-of-concept in Figure 6. The *DUPO* platform is developed to receive data from the sensors and process data subscription from the intelligent parking application (IPA). We used Apache Tomcat[1] as a web applications server to deploy our *DUPO* platform. The IPA is a RESTful service developed using Restlet[2], a framework for developing REST web services. The service requests the relevant data from the *DUPO* platform using the Pub/Sub APIs provided.

Sensor devices are simulated by using DPWS Simulator[3], and CoAP Simu-

[1] http://tomcat.apache.org/

[2] http://restlet.com/

[3] https://github.com/sonhan/dpwsim

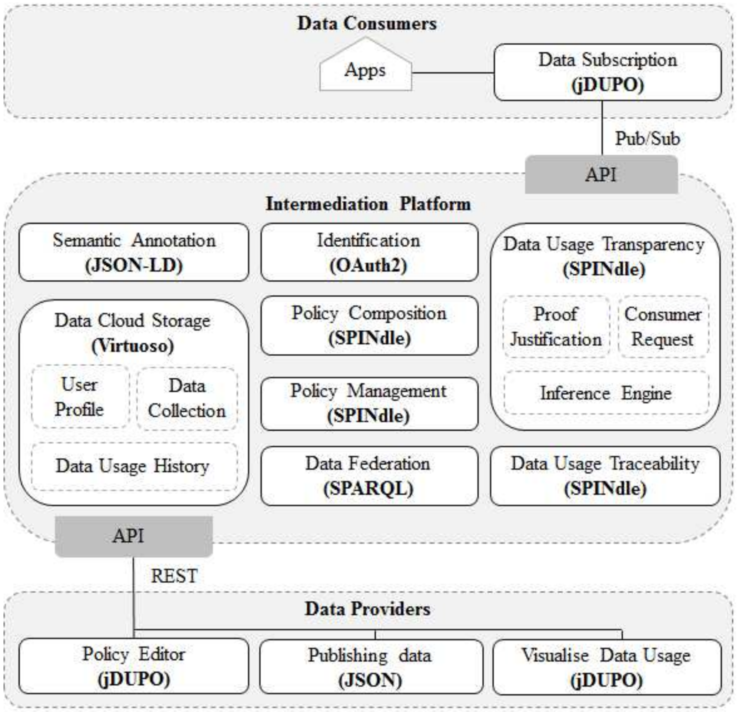


Figure 7: Implementation Choices of the Proof-of-concept

lator[4]. We also use the raspberry PI[5] to run the z-ware/ethernet gateway. All real Z-wave sensor devices emit z-wave messages that are caught by the gateway. This data can be processed locally by the raspberry. The simulated sensors and the gateway use the REST APIs provided to forward the data to our platform.

5.2. Implementation Choices

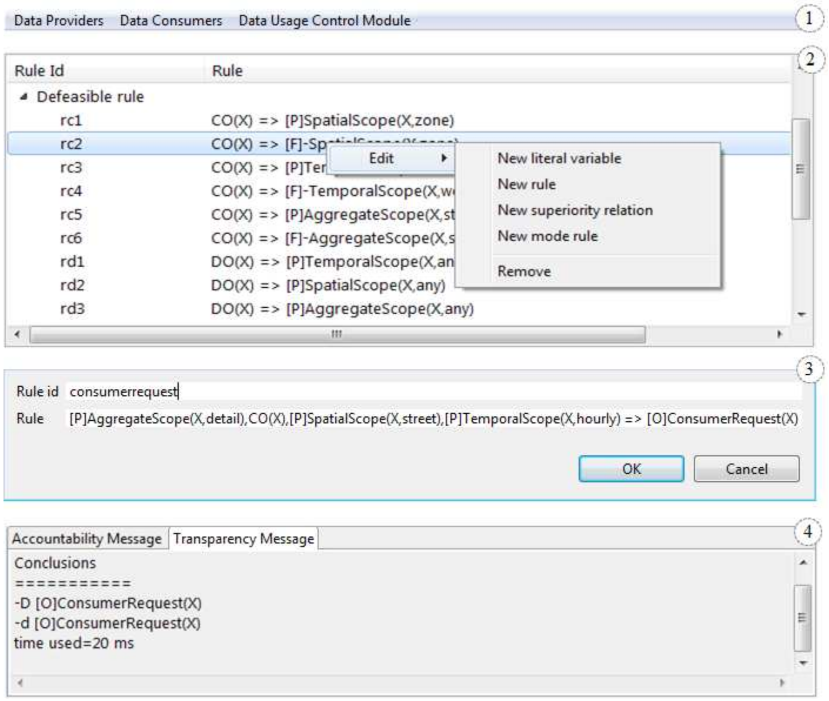
Figure 7 explains more about implementation choices for the proof-of-concept. We proposed essential technologies that are used to develop prototypes for the platform APIs.

We used Apache Jena Framework[6], an open source Java Framework for developing the functionalities of Data Annotation. In fact, the platform received

[4] https://github.com/caohuuquyet/jhess/tree/master/jUCP

[5] http://www.materiel.net/barebone/raspberry-pi-type-b-106574.html

[6] https://jena.apache.org/



Rule Id	Rule
Defeasible rule	
rc1	CO(X) => [P]SpatialScope(X,zone)
rc2	CO(X) => [F]-Sp
rc3	CO(X) => [P]Ter
rc4	CO(X) => [F]-TemporalScope(X,w
rc5	CO(X) => [P]AggregateScope(X,st
rc6	CO(X) => [F]-AggregateScope(X,s
rd1	DO(X) => [P]TemporalScope(X,an
rd2	DO(X) => [P]SpatialScope(X,any)
rd3	DO(X) => [P]AggregateScope(X,any)

Rule id: consumerrequest

Rule: [P]AggregateScope(X,detail),CO(X),[P]SpatialScope(X,street),[P]TemporalScope(X,hourly) => [O]ConsumerRequest(X)

```
Conclusions
===========
-D [O]ConsumerRequest(X)
-d [O]ConsumerRequest(X)
time used=20 ms
```

Figure 8: The main interface of the implemented prototype of our visualization tool namely *jDUPO* including (1) Main menu, (2) Policy editor, (3) Consumer's request, (4) Transparency and Traceability

the raw data from the sensors or the gateway, we aim to convert it to linked data[8]. A specific syntax called JSON-LD[7] is used to serialize Linked Data with the motivation to reduce the size of RDF documents compared to the size yielded by XML serialization. The linked data is stored in the Data Cloud Storage which use Virtuoso[8]. We also processed SPARQL query to implement the component of Data Federation.

We built on SPINdle[21] for functionalities of Data Usage Transparency and Traceability, Policy Composition, and Policy Management. It is a logic reasoner that can compute efficiently the consequences of *DUPO* theories [21].

[7] https://www.w3.org/TR/json-ld/

[8] http://virtuoso.openlinksw.com/

The Identification component is used granting access to the platform. In the prototype, we use OAuth[9] for this purpose.

In the next, we discuss about the jDUPO prototype that is used to edit policy, create consumer' request, and visualize data usage.

5.3. jDUPO Prototype

We developed a prototype version of our visualization tool namely *jDUPO* which aims to help users and data owners to customize their policies in a way that allows them to explore the consequences of each change and monitor how the data is going to be used after sharing it. We implemented an initial policy editor, including its functionalities for data usage control, and a short demo illustrating the use case scenarios. Figure 8 shows a snapshot of a *jDUPO* interface which shows some of the prototype functionalities.

5.3.1. Policy Editor

In this prototype, we use SPINdle syntax to define facts, rules, and rule priorities for *DUPO*. For example, Listing 7 shows the data usage policies for the *CO* in SPINdle syntax. End users, however, could also use *jDUPO* to edit their policies.

```
# Facts
>> CO(X)
# Defeasible rules
r1c: CO(X) =>[P] SpatialScope(X,zone)
r2c: CO(X) =>[F] -SpatialScope(X,zone)
r3c: CO(X) =>[P] TemporalScope(X,weekly)
r4c: CO(X) =>[F] -TemporalScope(X,weekly)
r5c: CO(X) =>[P] AggregateScope(X,statistic)
r6c: CO(X) =>[F] -AggregateScope(X,statistic)
...
```

Listing 7: Data Usage Policies in SPINdle syntax.

[9] Auth: http://oauth.net/2/

5.3.2. Consumer's Request

We are able to use *jDUPO* to create a consumer's request as well. Listing 8 shows an example of a consumer's request in the SPINdle syntax.

```
# Consumer‘s request
r: CO(X),[P]SpatialScope(X,street),[P]TemporalScope(X,hourly),[P]AggregateScope(X,detail) =>[O] ConsumerRequest(X)
```

Listing 8: Consumers' Request in SPINdle syntax

5.3.3. Transparency and Traceability

By using *jDUPO*, we are able to process the transparency and traceability of data usage. Listing 9 shows the conclusions of the consumer's request with an inference logger built on top of the SPINdle Reasoner.

```
# Conclusions
==================
-D [O]ConsumerRequest(X)
-d [O]ConsumerRequest(X)
...

=== Inference Logger ===
Rule_00000
+-- [DEFEASIBLE] Discarded :- [-d [O]ConsumerRequest(X)]
...
```

Listing 9: SPINdle-based Conclusions and Inference Logger.

5.4. Performance Analysis

In order to measure the performance of our solution, we conduct some experiments by using *jDUPO* and considering the intelligent parking use case. We run the prototype on a HP Elite Book 850 G3 computer with an Intel Core-i5-6300 2.4 GHz processor, 8 GB of RAM, and a 64-bit Windows 7 Enterprise operating system.

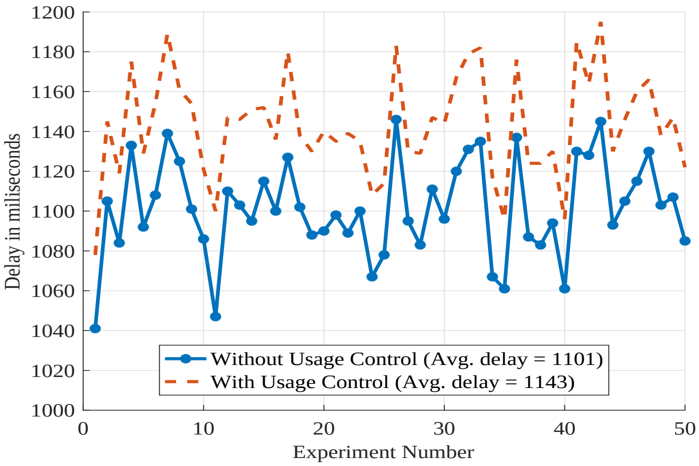


Figure 9: End-to-End Delay (E2ED).

We also use the parking dataset in the European Project CityPulse [22]. It includes a total of 8 parking lots providing information over a period of 6 months (55.264 data points in total) in the city of Aarhus.

In the following experiments, the query was that of a municipal authority asking for the average occupancy of parking places per street on an hourly basis. The performance was assessed in terms of the following metrics: End-to-End Delay (E2ED), Trust Computation Time (TCT), Impact on the Computational Time (ICT), and Memory Usage (IMU). E2ED is the time delay which takes to process the consumer request and get the data response. TCT is time used only for processing usage control. By increasing the number of rules, ICT and IMU were studied in terms of impact on computational time and memory usage.

The first experiment aims to compare performance result of the E2ED with and without usage control. The request is processed and repeated 50 times and the result of this experiment is shown in Figure 9. The highest value of E2ED with usage control is 1195ms, while the lowest one is 1078ms (in average 1143ms). In the other case, without usage control, the highest value of E2ED is 1146ms, and the lowest one is 1041ms (in average 1101ms). As it can be seen, on the average the overhead of usage control in the first experiment is about

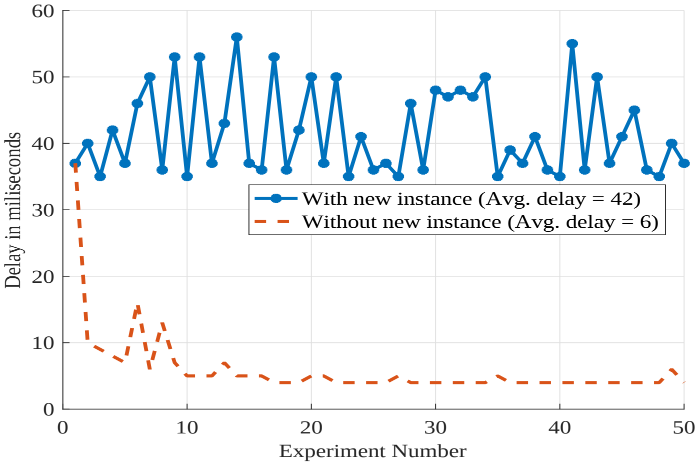


Figure 10: Trust Computation Time (TCT). [Average values are mentioned in legends]

3.8%.

The second experiment aims to evaluate actual value of TCT with and without new instance cases. In the first case, we restarted *jDUPO* to create new instance for each request processing. In the second case, we used the same instance for subsequent consumer request. Based on that, we compared the TCT in delay milliseconds after 50 repetitions. Figure 10 shows the results of the experiment with and without new instance respectively. The highest value of TCT with new instance is 56ms, while the lowest one is 35ms (in average 42ms). The highest value of TCT without new instance is 37ms, and the lowest is around 4ms (in average 6ms).

The third experiment aims to evaluate the impact on the computation time (ICT) and the impact on the memory usage (IMU), we compare the time and memory usage which is needed for trust computation as number of rules increases from 1000 to 10000. Toward this end, 25 cases consisting of 10 runs of each were performed. The result of ICT consumed is shown in Figure 13. It shows that the computational time taken increased linearly ($y = 0.08x$, $R^2 = 0.99$) with increasing number of rules. In the case of IMU, Figure 14 shows the impact result on the memory usage. It demonstrate that the memory also increased linearly with the increase of number of rules ($y = 0.02x$,

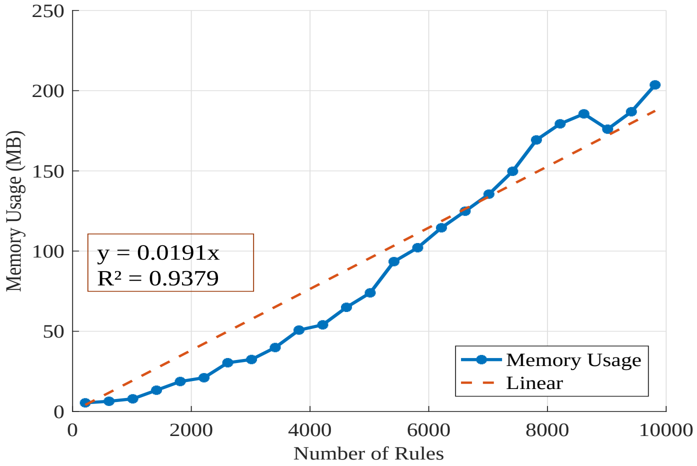


Figure 11: Impact on the Computational Time (ICT).

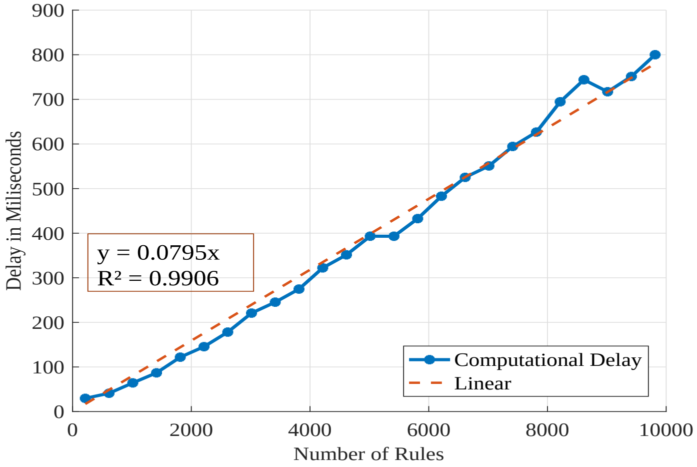


Figure 12: Impact on the Memory Usage (IMU).

$R^2 = 0.94$).

In conclusion the performance evaluation shows that the overhead of usage control stays reasonably in the range of 3.8%with new instance creation and about 0.5% without new instance. Also the growth in overhead of usage control stays linear even in very complex cases with thousands of rules.

6. Related work

This section introduces the relevant studies on trust and control enhancing technologies and explain the gaps beyond these studies. It is categorized in five separate approaches including privacy preservation, data licensing, access control, usage control mechanisms, and trust computation. We then do a comparison of the related frameworks with our proposal.

6.1. Privacy Preservation

Privacy is a major issue when it comes to data sharing. The main challenge is to provide techniques allowing data publishers to publish data in such a way that she does not breach the privacy of the data subjects still retains sufficient utility for the data recipients [23]. According to [24], the designed technologies to enhance privacy can be classified into two main categories: (*i*) Technologies for avoiding or reducing as much as possible the disclosure of personal data, hence enforcing the data minimisation principle; and (*ii*) Technologies for enforcing the rights of the subject if personal data is disclosed or processed. Although privacy is not our main focus but what does go along with data usage control is the notion of abstraction level that the data producer wishes to provide. These abstractions could be studied to provide mechanisms that can be used by a privacy module. However, there is still no specific data usage control model to express the constraints and obligations on the use of IoT data among participants. Moreover, information accountability is complementary to privacy [25]. Shifting to accountability as the basis for considering information sharing and disclosure is more tractable than abstract notions of privacy [26]. We consider that this is also essential in our context, but there are still lack of mechanisms to allow for automated data usage control and traceability of data usage.

6.2. Data Licensing

Data licensing is an active research domain which enables self-description of data consisting in licensing terms. A licensing vocabulary example is introduced by [27]. The licensing terms aim to specify the admitted use and re-use of the data by third parties. Authors in [13] propose a framework to build a composite license starting from the single licensing terms associated to heterogeneous data. However, the existing solutions have not yet been focused on improving the data usage transparency and traceability to address for IoT smart cities use cases. Thus, we aim to allow data owners to express and to ensure that their obligations on data usage have been respected.

6.3. Access Control

Access Control is a key issue to enable a secure and trustworthy data sharing as it regulates who can access protected information or services. Many mechanisms have already been specified to control the access toward software systems [28]. While the security aspects in access control have been dealt extensively, issues to address transparency and traceability of data usage are still subjects of research. Also, access control cannot deal with situations where information is published on purpose but should still have restricted usages [29]. The data-purpose algebra by [30] mentioned the modeling of usage restrictions of data and the transformation of the restrictions when data is processed. In their approach, a data item is associated with its content, the agent who produced it, the set of purposes for which usage is allowed as well as a set of categories. Depending on the performed process on a data item, a function is defined that transforms the allowed usages. However, a mechanism is needed to response the general idea of modeling the constraints and obligations about data usage requirements defined by data owners. It also has to extend to treat the issues of data usage transparency and traceability in IoT.

6.4. Usage Control Mechanisms

Usage control goes further than access control by regulating usage of information after initial access was granted [31]. UCON model is a theoretical foundation for usage control and initially propose by [32] with a purpose of being addressed to emerging digital environments. Usage control may deal with policies and mechanisms to ensure that consumers fulfill the obligations and conditions that data owners desires to impose on its utilization [33, 34]. The main focus of our study are on issues that we consider have not been treated in IoT. In fact, what does go along with usage control is the notion of the levels of abstraction that the producer wishes to provide, for instance mean data over a day and over a geographical zone rather than individual elements from each sensor and for each time period. The main technical challenges are to express the obligations and conditions in usage control policies and to ensure the transparency and traceability of the policy enforcement. Actors also need to have an easily interpretable tool to demonstrate in a clear fashion the reasoning behind.

6.5. Trust Computation

A comprehensive summary on trust has been investigated in [35, 36]. According to [37], several approaches have been proposed to compute trustworthiness based on direct information (direct trust). In this regard, transactions between trustors and trustees are established; and during these transactions, several credentials are generated for evaluating trust value. Others have measured trust based on third-party opinions (indirect trust) by accumulating feedback after interactions. Following this, a reputation value is then calculated by using heuristic algorithms and used to indicate trust. Regarding the trust aspect, we believe that control over the usage of data by other actors is critical in building trust, but further work is needed to work out concrete solution for trustworthy data sharing in the IoT.

6.6. Framework Comparison

In this last part, we aim to compare the general characteristic of the proposed framework in this study with other similar approaches. As it is mentioned earlier, we aim to tackle the issues of trust and control in the context of IoT smart cities use cases. In particular, we use the concept of usage control by [31] as a starting point to develop the data usage control model that enables the expression and definition of obligations on data usage. It should be noted that usage control policies apply to an entire group of devices - for instance a particular class of sensors in a given geographical area and deployed by a specific actor. In particular spatial and temporal constraints that a data provider imposes on the usage of the data apply to the data generated by this group. We believe that is a novelty of our framework which has not been addressed by prior works. In addition the model not only decides whether to provide access to the data, but also provides an explanation for the decision.

To understand better the position of our framework in compare to other solutions, a comparative analysis of our proposed model *DUPO* with respect to others is provided in Table 1. In a relevant study, Speiser *et al.* [38] specified the conceptual policy model to deal with this issue of abstraction of information, but this model does not respond to the obligations defined by the actors for their data. In the context of a social network, Pato *et al.* [26] proposed the solution which encourage responsible use of information by combining clearly expressed usage policies with systems for detecting misuse. However it does not address the issues in an IoT smart city use cases. In another study, Governatori *et al.* [13] focus on the data licensing using the composite license from the single licenses. Our trust model is policy-based usage control approach. We develop the formal theory and its proof based on DL, the data usage policies and each consumer requests are expressed as in regular DL rules. We also apply semantic technologies to IoT Data aggregation and interpretation. Lastly it worth to mention again that our contribution applies to a group of devices and in particular the constraints and obligation used in the policies apply to an

Table 1: Comparative study previous approaches to our proposal based on different features.

	Speiser, et al. [38]	**Pato, et al. [26]**	**Governatori, et al. [13]**	**DUPO**
Domain	Smart Grid	Web, Social Network	Web of Data	Smart Cities
Use cases Scenario	Energy Consumption	Health Insurance	Composite License	Intelligent Parking
Requirement	Usage Perspectives	Usage Restrictions	Set of Licenses	Data Usage Obligations
Policy Model	Yes	Yes	No	Yes
Policy Representation	RDF/N3 Syntax	AIR Language	Deontic Logic Semantics	Defeasible Rules
Deal with Rule conflict	No	No	Yes	Yes
Policy Composition	No	No	Yes	Yes
Trust Model	Abstraction Information	Information Accountability	Data Licensing	Policy-based Usage Control
Proposed Platform	No	Yes	No	DUPO Platform as a Service
Visualization Tool	No	Yes	No	SPINdle-based jDUPO
Evaluation	Policy Matching	No	No	With and without Usage Control

aggregation of devices in spatial and temporal domains which is an novel part in *DUPO*. For IoT domain, we believe that this dimension is needed as millions of devices are involved and the appropriate level for usage control policies needs to be provide for higher level abstractions and not be restricted to individual device level. Considering all said so far, to the best of our knowledge, the ideas presented in this study are novel and different from earlier efforts in the IoT domain.

7. Conclusion

Sharing data across multiple entities can be highly rewarding in terms of insights and usability but trust is the key point when stakeholders share data. One important aspect of building trust is for the data owner to be able to exercise control over the usage of the data by other actors. In this paper, we concentrate on this issue namely Usage Control which have not been adequately addressed in the context of an intermediation platform for smart cities.

We proposed a model for policy-based data usage control (namely *DUPO*) with its conceptual model, formal theory, and illustrative scenario. This model responded to the diversity of obligations or data usage requirements that data owners impose on the use of their data. It also focused on the non-monotonic formalism which aims to handle the normative conflicts between rules, rules with deontic consequents, and exceptions, illustrated the logical reasoning applied when the policies are enforced in a computationally tractable way. The illustrative made use of a smart city scenario aims to explain the model concepts. A trustworthy data sharing platform as service is then defined. It allowed transparency and traceability of data usage with the core components based on *DUPO* and Semantic technologies. We also presented in detail the main procedures for the trustworthy data sharing in aspects of data owners, consumers, and an intermediation platform. Moreover, a proof-of-concept is developed and a visualization tool is provided to help users easily control and monitor how their data is shared. Finally, we investigated the performance of the system with the initial assumption about trust and control to compare the performance results with and without those assumptions. All experiments are presented along with the results and more importantly it showed that the performance of the added trust does not impact negatively on the system.

As future work of this study, several aspects that are not covered here can be considered. The first important future direction can be employment of our trust computing framework. Toward that end, we aim to enhance the efficient query answering and performance on real-time responses in production systems. The improvement can be on the reasoning mechanism for more complex use cases and for supporting real-time processing, discussion regarding scalability and distribution. In addition, we would like to develop open standard APIs which have the ability to attract partners to share data on the platform, manage metadata along their usage and their value, and deliver the right data to partners and handle semantics variability. Another important idea as future work is to involve end-users in the evaluation of the proposed visualization tool in order to ensure their usability.

References

[1] A. M. Townsend, Smart cities: Big data, civic hackers, and the quest for a new utopia, WW Norton & Company, 2013.

[2] L. Atzori, A. Iera, G. Morabito, The Internet of Things: A Survey, Computer Networks 54 (15) (2010) 2787–2805.

[3] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of things (iot): A vision, architectural elements, and future directions, Future Generation Computer Systems 29 (7) (2013) 1645–1660.

[4] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, Internet of things for smart cities, Internet of Things Journal, IEEE 1 (1) (2014) 22–32.

[5] L. Sanchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis, et al., Smartsantander: Iot experimentation over a smart city testbed, Computer Networks 61 (2014) 217–238.

[6] R. Khatoun, S. Zeadally, Smart Cities: Concepts, Architectures, Research Opportunities, Commun. ACM 59 (8) (2016) 46–57.

[7] D. Christin, Privacy in mobile participatory sensing: current trends and future challenges, Journal of Systems and Software 116 (2016) 57–68.

[8] T. Berners-Lee, Linked Data, in: International Journal on Semantic Web and Information Systems, Vol. 4, W3C, 2006.

[9] OMA, NGSI Context Management, Tech. rep., http://goo.gl/mv6qFZ (2010).

[10] FIWARE, FI-WARE Platform, http://forge.fiware.eu/plugins/mediawiki/wiki/fiware/index.php (2016).

[11] D. Nute, Handbook of Logic in Artificial Intelligence and Logic Programming (Vol. 3), Oxford University Press, Inc., New York, NY, USA, 1994, Ch. Defeasible Logic, pp. 353–395.

[12] G. Governatori, A. Rotolo, BIO logical agents: Norms, beliefs, intentions in defeasible logic, Autonomous Agents and Multi-Agent Systems 17 (1) (2008) 36–69.

[13] G. Governatori, A. Rotolo, S. Villata, F. Gandon, One License to Compose Them All, in: The Semantic Web–ISWC 2013, Springer, 2013, pp. 151–166.

[14] G. Antoniou, D. Billington, G. Governatori, M. J. Maher, Representation Results for Defeasible Logic, ACM Trans. Comput. Logic 2 (2) (2001) 255–287.

[15] G. Antoniou, N. Dimaresis, G. Governatori, A modal and deontic defeasible reasoning system for modelling policies and multi-agent systems, Expert Systems with Applications 36 (2) (2009) 4125–4134.

[16] E. Kontopoulos, N. Bassiliades, G. Antoniou, Deploying defeasible logic rule bases for the semantic web, Data & Knowledge Engineering 66 (1) (2008) 116 – 146.

[17] Q. H. Cao, I. Khan, R. Farahbakhsh, G. Madhusudan, G. Lee, N. Crespi, A Trust Model for Data Sharing in Smart Cities, in: IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 2016.

[18] R. T. Fielding, Architectural styles and the design of network-based software architectures, Ph.D. thesis, University of California, Irvine (2000).

[19] J. Bacon, D. M. Eyers, J. Singh, P. R. Pietzuch, Access control in publish/-subscribe systems, in: Proceedings of the second international conference on Distributed event-based systems, ACM, 2008, pp. 23–34.

[20] E. Kontopoulos, N. Bassiliades, G. Antoniou, Visualizing Semantic Web proofs of defeasible logic in the DR-DEVICE system, Knowledge-Based Systems 24 (3) (2011) 406–419.

[21] H.-P. Lam, G. Governatori, The making of SPINdle, in: Rule Interchange and Applications, Springer, 2009, pp. 315–322.

[22] CityPulse, The parking dataset in the European Project CityPulse, Tech. rep., http://iot.ee.surrey.ac.uk:8080/datasets.html#parking (2014).

[23] B. Fung, K. Wang, R. Chen, P. S. Yu, Privacy-preserving data publishing: A survey of recent developments, ACM Computing Surveys (CSUR) 42 (4) (2010) 14.

[24] D. Le Métayer, Whom to trust? using technology to enforce privacy, in: Enforcing Privacy, Springer, 2016, pp. 395–437.

[25] A. Kung, Pears: privacy enhancing architectures, in: Annual Privacy Forum, Springer, 2014, pp. 18–29.

[26] J. Pato, S. Paradesi, I. Jacobi, F. Shih, S. Wang, Aintno: Demonstration of Information Accountability on the Web, in: Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on, IEEE, 2011, pp. 1072–1080.

[27] A. Rotolo, S. Villata, F. Gandon, A deontic logic semantics for licenses composition in the web of data, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law, ACM, 2013, pp. 111–120.

[28] M. Sookhak, F. R. Yu, M. K. Khan, Y. Xiang, R. Buyya, Attribute-based data access control in mobile cloud computing: Taxonomy and open issues, Future Generation Computer Systems.

[29] S. Speiser, A. Harth, Data-centric privacy policies for smart grids, in: 2012 AAAI Workshop— Semantic Cities. The AAAI Press, Palo Alto, California, 2012, pp. 31–36.

[30] C. Hanson, L. Kagal, T. Berners-Lee, G. J. Sussman, D. Weitzner, Data-purpose algebra: Modeling data usage policies, in: Policies for Distributed Systems and Networks, 2007. POLICY'07. Eighth IEEE International Workshop on, IEEE, 2007, pp. 173–177.

[31] A. Pretschner, T. Walter, Negotiation of usage control policies-simply the best?, in: Availability, Reliability and Security, 2008. ARES 08. Third International Conference on, IEEE, 2008, pp. 1135–1136.

[32] J. Park, R. Sandhu, Towards usage control models: beyond traditional access control, in: Proceedings of the seventh ACM symposium on Access control models and technologies, ACM, 2002, pp. 57–64.

[33] A. Lazouski, F. Martinelli, P. Mori, Usage control in computer security: A survey, Computer Science Review 4 (2) (2010) 81–99.

[34] E. Carniani, D. DArenzo, A. Lazouski, F. Martinelli, P. Mori, Usage control on cloud systems, Future Generation Computer Systems 63 (2016) 37–55.

[35] S. Sicari, A. Rizzardi, L. A. Grieco, A. Coen-Porisini, Security, privacy and trust in internet of things: The road ahead, Computer Networks 76 (2015) 146–164.

[36] Z. Yan, P. Zhang, A. V. Vasilakos, A Survey on Trust Management for Internet of Things, Journal of Network and Computer Applications 42 (2014) 120 – 134.

[37] T.-W. Um, G. M. Lee, J. K. Choi, Strengthening trust in the future social-cyber-physical infrastructure: an itu-t perspective, IEEE Communications Magazine 54 (9) (2016) 36–42.

[38] S. Speiser, A. Wagner, O. Raabe, A. Harth, Web technologies and privacy policies for the smart grid, in: Policies for Distributed Systems and Networks (POLICY), 2011 IEEE International Symposium on, IEEE, 2011, pp. 121–124.