Privacy Administration in Distributed Service Infrastructure

Nabil Ajam, Nora Cuppens-Boulahia, and Frederic Cuppens

Institut Télécom
Telecom Bretagne, 2 rue de la Chataigneraie Cesson-Sevigne 35576
LUSSI Department
{nabil.ajam,nora.cuppens,frederic.cuppens}@telecom-bretagne.eu

Abstract. In this paper, we propose a framework to administrate privacy policies in distributed service infrastructure. We define new administrative capabilities that model user preferences and specify how data owners can access to them. We investigate a distributed administration of the privacy policy where three different administrative policies can coexist and one can dominate the other. We define the data collector practices, the legal organisation policies, such as emergency service's policies, and the negotiated policy between the data collector and services providers. We finally specify how to manage these three distributed privacy administration policies.

Keywords: Privacy administration, privacy policy model, access control model, legal policy, user preferences, interoperability, SLA.

1 Introduction

Privacy can be defined as the demands from individuals, groups and institutions to determine by themselves when, how and to what extent information about them is to be communicated to others [1]. By personal data we mean any information that can be used to identify directly or indirectly a person, who is the data subject or the owner of the information. Privacy concerns raise more and more scientist's attention, especially in infrastructures of distributed service providers, such as location-based service (LBS) providers. We stress that the data controller, which collects sensitive information, is different from the service provider, which is the requestor that uses this information to offer services.

Access control models provide a scalable solution to specify privacy policies. Related works mainly proposed an extended RBAC model based on the definition of purposes and obligations [6,17,21]. They chose RBAC to integrate privacy policy because it is a widely deployed model in information systems. However, since RBAC is intrinsically not expressive to handle privacy requirements, this leads to many extensions of RBAC that handle different aspects of privacy. By contrast, we argue that the OrBAC model [8] is natively expressive enough to handle dynamic privacy parameters thanks to contexts in order to implement user privacy preferences. In [3], we have focused on the specification of the privacy policy based on the data owner's preferences. In this paper, we propose an administration model of privacy policies.

Administration tasks of a security policy refer to the specification of the privileges and rules that monitor the whole policy. For example in OrBAC policies, administration capabilities are defined through different administrative views that are responsible for specifying who can add or remove new roles or new privileges to the security policy. Usually, only the policy administrator has access to those views, so the administration is centralized. In our case study, we have three different entities that may handle administrative capabilities, namely the data owners, the data collector and the requestors. The data owners are the subscribers that the collected sensitive data refer to. The data collector is the mobile operator that collects the data, stores it and manages the privacy policy. The requestors are the service providers that need that sensitive information to offer their services, such as location-based services (LBS), or legal organisation that needs the information for security purpose or for legal interception. Each of these actors has its specific privacy requirements. We define and show how to manage this distributed administration, which is composed of different requirements defined by different entities.

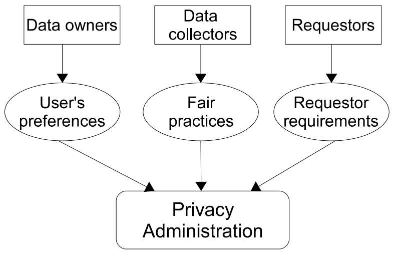

Fig. 1. The distributed administration

This paper is organised as follows. Second section introduces a concrete example to motivate our approach. Third section briefly recalls our privacy-aware approach based on the OrBAC model. Section four is dedicated to our new administration approach of privacy policies. Section five introduces related works. Concluding remarks are presented in section six.

2 Motivating Example

LBS are services that make use of sensitive location information. We can categorize them in two types: the services that use the location information computed and stored within the mobile terminal and the services that need the position collected and managed by the mobile networks. Available positioning techniques depend on the mobile equipment features and the mobile network architecture.

If the location is computed by the mobile equipment itself, a privacy policy can be managed thanks to the P3P framework. However, if the mobile organisation computes and holds these sensitive data, we proposed to use the OrBAC model to define the privacy policy [3].

Let us consider the following example. Suppose that Alice and Bob are mobile subscribers. Alice owns a location-enabled mobile device. The Alice location can be computed locally thanks to its equipment. However, Bob is only located by the mobile network, so the location is stored and managed by the mobile operator. Alice uses the P3P framework to control which third parties may access to her location. For this, she stores her privacy preferences in her user agent, her mobile device in this example, through the APPEL language [20]. And when a third party submits an access request to the user agent, it declares its privacy policy through the P3P language. Next, Alice's user agent will evaluate and compare the privacy preferences with the third party policy and if they are consistent then it discloses the location information to the requestor. Notice that the sensitive information is still under the control of the data owner until the policy check. So, the data owner has full control over her data since she stores the data until the privacy verification.

By contrast, Bob's privacy data protection is slightly different. The mobile operator controls Bob's data until the privacy verification. We need to assume that mobile subscribers trust this mobile operator. An agreement is signed between them to provide the location service. A trustworthy relation exists between them since they already signed an agreement for voice services. But this framework provides an interesting security pledge to the owners. The mobile operator naturally enforces stronger security mechanisms to protect subscriber's data compared to the mobile device. The location data can be easily stolen from the mobile device if it is hijacked, especially when users install unknown applications from Internet. The mobile operator can also prevent data owners from specifying weak privacy policies since user behaviours sometimes do not reflect their needs of privacy protection and they may ignore privacy protection when service are provided to them [2,18]. So, we argue that an intermediary authority like the mobile operator can prevent users from disclosing excessive information. The last point that motivates our approach, using an intermediary entity, is that when Bob's sensitive data must be used for emergency cases or for legal interception. In this case, Alice's privacy policy can prohibit the access to her data even if it is useful for security purposes that can save lives. She manipulates the device's firmware to prohibit access to the location information. By contrast, Bob's preferences can be bypassed by the mobile operator according to a legal procedure such as legal interception (which is a requirement in many countries), since the operator controls such data. Users must trust the operator to allow such privacy exceptions and we assumed that in our case.

In this paper, we are concerned about the privacy policy administration in that context, namely how the mobile operator will integrate data owner's preferences within its fair information practices. The legal procedures and data owner preferences must coexist without conflict generation. Moreover, does the service requestor has the opportunity to negotiate some privacy parameters with the mobile operator?

We opted for three possible administration enforcements due to the specificity of the privacy requirements. Mobile users can subscribe to the location service, so the privacy preferences can be embedded to this agreement or administrated on the fly through the management of the privacy views. Requestors can also specify some privacy requirements. The privacy policy is a trade-off between those requirements and the operator's fair practices. Furthermore, the trustworthy relation is a relevant parameter. It impacts the administration model of privacy policies. In our case study, mobile operators are trusted enough to provide users with confidence on the manner that their privacy policies are managed. So, we can propose such a privacy framework based on these trusted entities.

We identify three cases of privacy administration enforcement. First, the operator organisation will define and enforce its policy that other actors must agree with. Second, the legal organisations, such as emergency services and intelligence departments, can bypass the operator policy and impose their policy without generating policy conflicts since they have prioritised privileges for security purposes. Third case is when the operator looks for a compromise between the different requirements. This case is modeled using the interoperability approach O2O (Organization to Organization) [10] between service provider's organisations and the operator to negotiate the resulting policy. The privacy policy is a deal between the mobile operator and the service provider based on data owner preferences.

3 The Privacy-Aware OrBAC Model

3.1 The OrBAC Model

In the Organization-Based Access Control model (OrBAC) [15], security policies of an organisation *org* are specified at the abstract organisational level through four privileges: permission, prohibition, obligation and dispensation. Instead of directly specifying security policies using concrete subject, action and object entities, these privileges are applied to three abstract entities: roles, activities and views. Moreover, every privilege may depend on some context. For example, $Permission(org, r, a, v, c)$ means that the role r is permitted to perform the activity a on the view v in context c. To derive the concrete security rules, the model introduces three basic built-in predicates:

- *Empower* is a predicate over domains $Org \times S \times R$. If *org* is an organisation, s a subject and r a role, then *Empower(org, s, r)* means that s is assigned to the role r within *org*,
- *Consider* is a predicate over domains $Org \times A \times$ A. If *org* is an organisation, α an action and a an activity, then *Consider(org, α, a)* means that *org* considers that α is implementing the activity a,
- *Use* is a predicate over domains $Org \times O \times V$. If *org* is an organization, o is an object and v is a view, then *Use(org, o, v)* means that *org* uses the object o in the view v.

The correspondent derived concrete privileges are *Is_permitted*, *Is_prohibited*, *Is_obliged* and *Is_dispensed*. They apply to the concrete entities: subjects, actions and objects. *Is_obliged* means that the subject is obliged to perform the action. It has two contexts: activation context and violation context. *Is_dispensed* is the dual of *Is_obliged*.

Contexts are introduced to take into account the dynamic parameters of the security policy, such as the spatial location of subjects. An OrBAC built-in predicate *Hold* is used to specify contexts:

– *Hold* is a predicate over domains $Org \times S \times A \times O \times C$. If *org* is an organization, s is a subject, α is an action, o is an object and c is a context, then *Hold(org, s, α, o, c)* means that context c holds between subject s, action α and object o within *org*.

The OrBAC model defines five types of contexts [8]:

– Spatial context: that depends on the subject position,
– Temporal context: that depends on the time of the subject request,
– User-declared context: that depends on parameters declared by the subject,
– Prerequisite context: that depends on a relation between the subject, the action and the object,
– Provisional context: that depends on the previous actions of the subject.

3.2 OrBAC Administration

The OrBAC model is self-administrated, i.e. the OrBAC model may be used to specify administrative security policies. Initially, the administration model AdOrBAC [9] consists in the definition of roles and the corresponding privileges. AdOrBAC defines two administrative views for that. An assignment of a subject to a role is modelled by an insert of an object in the *role_ assignment* view. Similarly, granting a privilege to a role is modelled by an insert of an object in the *licence* view. The administrator in AdOrBAC specifies which role is permitted to access those administrative views and in which contexts.

Objects belonging to the *role_ assignment* view have two attributes: *assignee* is the subject to which the role is assigned and *assignment* is the role to be assigned. Objects belonging to the *licence* view have four attributes: *grantee* is the subject (or role) to which the licence is granted, *privilege* is the action (or activity) permitted by the licence, *target* is the object (or view) to which the licence grants an access to and *context* is the condition that must be satisfied to use the licence.

Delegation model is based on AdOrBAC and aims to transfer privileges and rights from one role to another [4]. We distinguish between the partial delegation and total delegation. The former delegates some rights whereas the latter delegates roles. So, two more administrative views are defined the *licence_ delegation* view and the *role_ delegation* view. Objects belonging to these views have the same attributes as *licence* and *role_ assignment* objects. But, they have an extra attribute, the *grantor*, which represents the subject who is creating the licence or the role. Inserting objects in these views allows a grantor to respectively delegate permission and role to a grantee. The administrator manages access to the delegation views and to the administrative views. So, a subject may delegate her rights only if some administrator grants her permission to delegate these rights by creating a license delegation (and similarly for the delegation of roles through the role delegation view).

Administrators are also responsible for managing conflict. Conflicts are solved in OrBAC thanks to the prioritised-OrBAC model [11]. The assignment of priorities is still under the control of the administrator. When a conflict is detected by the model and cannot be solved since two opposite privileges, prohibition and permission, are assigned to the same subjects and for the same actions on objects, the administrator has the privileges to specify the precedence of one policy over the other.

3.3 Privacy Contextual Management

In order to specify and manage privacy requirements, we need to model subject's consent over its personal data, accuracy of location objects and purpose for which some access is performed. We proposed in [3] to respectively model the subject's consent as a context, object's hierarchy based on the accuracy of objects, the purpose as a user-declared context and provisional obligation following the access to some sensitive information. Also, we propose to add a current state context and an enhanced spatial context.

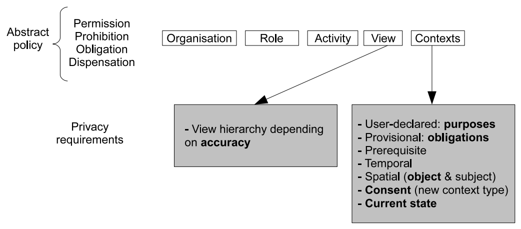
Abstract policy
Permission
Prohibition
Obligation
Dispensation
Organisation
Role
Activity
View
Contexts
Privacy requirements
- View hierarchy depending on **accuracy**
- User-declared: **purposes**
- Provisional: **obligations**
- Prerequisite
- Temporal
- Spatial (**object** & subject)
- **Consent** (new context type)
- **Current state**

Fig. 2. Contextual privacy management in the OrBAC model

The idea behind our proposal is to include the subject's privacy preferences into the contexts of the security policy of the organisation. The result is one policy for the access control and the privacy management.

A new context type, called consent, is used to model if the object owner gives its consent to the subject, who requests the access to that object. Users store their consent preferences in the *consent_preference* view. Each object in this view corresponds to a particular data owner preference and has three attributes: *Requestor*, who is the subject who requests the access to the object, *Target*, which is the requested object, and *NeedConsent*, which is a Boolean parameter

and if its value is *true* so the consent is needed. The user consent context is specified as follows:
Rule$_{consent}$: $\forall org \in Org, \forall s \in S, \forall \alpha \in A, \forall o \in O, \forall v \in V, \forall cp \in O,$
$Hold(org, s, \alpha, o, Consent\_context) \leftarrow Use(org, cp, Consent\_preference) \wedge$
$Requestor(cp, s) \wedge Target(cp, v) \wedge Use(org, o, v) \wedge NeedConsent(cp)$

Then, we suggested that private objects, of each data owner, have different accuracy levels. A private object has four attributes: *data-owner*, *longitude*, *latitude* and *accuracy*. A hierarchy is established between the root view, which contains the collected private data, and sub-views consisting of derived objects based on different accuracies. Those accuracies are defined by the data owner, so she can define different privacy preferences based on the accuracy of the object. The accuracy is specified by the couple *(anonymity level, k)*. *Anonymity level* defines the accuracy of the identity attribute of location information. However, k determines the accuracy of the location attributes, which are the longitude and the latitude. So, the operator can apply *k-anonymity* algorithms ([12], [16] and [14]) to derive the longitude and the latitude of the derived objects. The issue of choosing the optimal algorithm is out of the scope of this work.

We modeled the purpose of the access request by a user-declared context. Each data owner can create purpose objects to specify the purposes for which access to private objects are allowed. The purpose objects are grouped in a *Purpose* view. Each purpose object has two attributes [8]. *Recipient* defines who takes advantage of the declared purpose (a service provider in our case), and *declared_ purpose* associates a purpose value with the declared purpose object. Purpose values range over the purpose value domain PV. On the other hand, the service provider declares the purpose to be provided. So, *user_ declared* is a function over the PV domain. It returns the value of the context entered by the service provider.
Rule$_{purpose}$ $\forall org \in Org, \forall s \in S, \forall \alpha \in A, \forall o \in O, \forall po \in PO, \forall pv \in PV,$
$Hold(org, s, \alpha, o, user\_declared(pv)) \leftarrow Use(org, po, Purpose) \wedge Recipient(po, s)$
$\wedge\ Declared\_purpose(po, pv)$
That is, in organisation *org*, subject s performs action α on object o in the user declared context *user_ declared(pv)*, if there is a purpose object *po* used in view *Purpose* by organisation *org* such that s is the recipient associated with *po* and *pv* is the declared purpose associated with *po*.

Provisional obligations [5] are introduced to oblige subjects to perform some action following its access to the location information. We need obligations to enforce privacy principles of accountability. So, an obligation may be automatically triggered as a counterpart of the access to some private information. The obligation is expressed thanks to two types of contexts *context_ activation* and *context_ violation*:
$Obligation(org, r, a, v, context\_activation, context\_violation)$
meaning that subjects empowered in role r are obliged to perform actions implementing activity a on objects used in view v when $context\_activation$ is activated. This actually corresponds to an organizational obligation. Concrete obligations that apply to subjects, actions and objects are derived when $context\_activation$ is

activated. If subjects actually do not perform the obliged actions on the objects before *context_violation* is activated, then the obligation is violated. See [5] for more details about expression and management of obligations in the OrBAC model.

The *current state context* is used for location privacy policy when the data owner allows service provider to access her location only if she initiated a call or a session data with it. It is necessary to evaluate the current state in this context. The current state indicates if the user has initiated a call or a session to the service provider or not. We assume that *preferred-states* view contains data owner's preferences regarding authorized current states. The data owner specifies its preference by adding new entry to that view. Objects belonging to that view have three attributes: *calling*, *state-type* and *called.*

Physical and logical spatial contexts are relevant features for privacy policy. In addition to using it to locate the subject who asks for an access, spatial contexts are also useful to locate objects. As suggested by [13], we extend the semantic of spatial context to include the possibility to consider the object positions. The predicate *Is_within* determines if a given object or subscriber is within a location area or not. *Is_within* is a predicate over the domains $O \times LA$, where LA is a set of *location areas.* So, *Is_within(o, la)* means that the object o is within the location area la. The spatial context can now be defined using this new predicate.
$\mathbf{Rule}_{SpatialObject}$ $\forall org \in Org, \forall s \in S, \forall \alpha \in A, \forall o \in O, \forall la \in LA$
$Hold(org, s, \alpha, o, position(la)) \leftarrow Is\_within(o, la)$

4 The Privacy Distributed Administration

We propose in this section three administration enforcement approaches of the privacy policies. Before, we shall present the new administrative views related to the management of these privacy policies. Based on privacy principles, the owners would have access to these views to specify their privacy preferences. However, this is actually not the case for the two first approaches. In the first case, the dominant operator policy is deployed and the mobile subscribers delegate all privacy administrative tasks to the operator. This alternative ensures a consistent privacy policy with minimal policy updates. We model this alternative through regular administrative tasks: *role assignment*, *licence definition* and delegation. The characteristic of this alternative is the delegation of the management of the privacy administrative views to the operator. When the mobile subscriber signs an SLA with the operator, it implicitly delegates its rights to the operator. By doing so, the operator can define an optimal privacy policy by enabling or disabling privacy contexts. The operator can offer premium services based on the level of the privacy. This will increase its productivity (for example, it offers a cheaper service to subscribers who accept to be located by advertisers).

The second alternative states that there are prioritised policies. Our objective is to assign a higher priority to the policy defined by legal organisations, for legal interception. This case is useful to enforce security laws that override other privacy requirements. All mobile operators should authorize such conflicts, so we propose a conflict management solution based on the prioritised-OrBAC model [11] to manage conflicts when third parties override user's preferences.

The third alternative proposes an enhanced management of the privacy preferences. Privacy preferences are propagated to the service provider. The established SLA between the mobile subscriber and the operator includes privacy preferences and how the requestors must protect the sensitive data. For example, if the data owner specifies that purpose context must be declared before accessing the location information, this preference must be propagated to the service provider organisation. When another requestor asks for location information from the service provider, the purpose context will be checked. This alternative provides a single privacy policy definition that can be propagated to the service provider. So, it offers the simplest way for data owners to define a universal privacy policy that is enforced by all the interoperable organisations. For this purpose, we use the O2O approach [10]. So, O2O must be supported by the service providers to allow this generalised privacy protection.

4.1 Privacy Administrative Views

Our privacy policy model is mainly based on the definition of contexts. The operator organisation, which enforces privacy preferences of its subscribers, has to administrate the access to those views. In this section, we list the views related to the privacy preferences. Access to these views is controlled by the OrBAC model itself.

Definition *Privacy-administrative views*
Data owners' preferences are implemented in the privacy-aware OrBAC thanks to the views:
- Consent_preference: is responsible for storing if a consent is needed or not,
- Purpose: contains the available purposes that can be declared,
- Preferred-states: contains the preferred states of the connection between the requestor and the data owner at the moment of the access request,
- Spatial_preferences: contains the location areas where data owners can be located.

Logically, data owners have the full privileges over those views when they act in the *owner* role. We define the *management* activity allowing them to add, insert, modify or suppress objects on these privacy administrative views.

$\textbf{Rule}_{preferenceAdministration}$
$Permission(org, owner, management, Consent\_preference, consent\_owner)$
$Permission(org, owner, management, Purpose, subscription)$
$Permission(org, owner, management, Preferred\_states, subscription)$
$Permission(org, owner, management, Spatial\_preferences, default)$
where the *consent_ owner* and the *subscription* contexts are defined as follows:
$\forall org \in Org, \forall s \in S, \forall s' \in S, \forall \alpha \in A, \forall o \in O, \forall position \in O, \forall po \in PO,$
$Hold(org, s, \alpha, o, consent\_owner) \leftarrow use(org, o, Consent\_preference) \wedge Target(o, position) \wedge Data\text{-}owner(position, s)$
$Hold(org, s, \alpha, po, subscription) \leftarrow Use(org, po, Purpose) \wedge Recipient(po, s') \wedge Subscribed(s, s')$

consent_ owner context is triggered if the consent object o belongs to the privacy view *consent_ preference* and it has the attribute *position* as target. This position is owned by s, who is the data owner.

subscription context is triggered when a subject s performs action α on a purpose object *po* and if this purpose object *po* has s' as a recipient and where s' is subscribed to s. This is represented by the application-dependent predicate *Subscribed.*

On the other hand, data owners have the right to modify their locations. Precisely, they can define several accuracies of their locations to define different privileges to service providers depending on the location accuracy. This is modeled as follows:
$Permission(org, owner, modify, location, owning)$
owning is a provisional context defined as:
$\forall org \in Org, \forall s \in S, \forall \alpha \in A, \forall o \in O$
$Hold(org, s, \alpha, o, owning) \leftarrow Data\text{-}owner(o, s) \wedge Consider(org, \alpha, modify) \wedge Empower(org, s, owner)$
It means that in organisation *org* the subject s performs α, which is considered a *modify* activity, on the object o only if s is *owning* the sensitive object. The data owner can modify only the *Accuracy* attribute of the location. Then, the operator will apply the obfuscation algorithms to compute the new *Identity*, *Longitude* and *Latitude* attributes. Note that *Data-owner* is an attribute of the location data.

4.2 First Case: Dominance of the Mobile Operator Policy

When users subscribe to a location service, an agreement is established between them and the operator. The latter, which is the data collector, proposes its fair practices. They include the privacy policy that will be enforced by the mobile operator. In other terms, that policy specifies how privacy contexts are managed by the data controller. If the user accepts this management, she delegates the contextual management of her privacy policy to the operator. The mobile operator organisation defines its access control policy based on its privacy fair practices without the intervention of data owners. The data owner tasks are implicitly delegated to the data controller when the agreement is established between them. To specify this procedure, we define first the delegation privilege given to the owners then we specify the licence delegation. The owners are permitted to delegate their rights to the mobile operator. It is ensured by the next privilege:
$Permission(org, owner, delegate, licence\_delegation, default)$
That is in the organisation *org*, subjects who act in the *owner* role have the right to *delegate* licences in the default context.

The *operator* organisation can be divided into several departments, each of them being responsible for providing one service, such as the location service. Let *administrator_ location* be the administrator role of the location service department. When owners sign an SLA with the operator, the following delegations are performed transparently. For example, the permission derived from the delegation on the *Purpose* view is specified by:

$\textbf{Rule}_{delegationPurpose}$
$Permission(operator, administrator\_location, management, Purpose, default)$ $\leftarrow use(L, licence\_delegation) \wedge grantor(L, data\text{-}owner) \wedge privilege(L, management) \wedge target(L, Purpose) \wedge grantee(L, administrator\_location)$
That is, the *administrator_ location* is permitted to perform the *management* activity on the *Purpose* view if there is a licence L, belonging to the *licence_ delegation*, where the grantor is the *data-owner* role, the target is the *Purpose* view and the grantee is the *administrator_ location.*

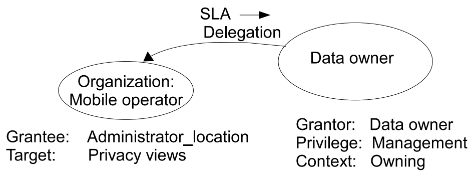

Fig. 3. Privacy policy delegation

Similarly, data owners delegate to the data collector the management of other privacy views, *Consent_ preference*, *Preferred-states* and *Spatial_ preferences*, when they subscribe to the location service of the operator. The operator provides several privacy packages to the users. Each package defines how the user privacy will be managed and which contexts will be activated.

For example, when the user delegates the management of the *Consent_ preference* view to the *administrator_ location*, the data owner will no longer be notified for its consent. The mobile operator can fix consent to *false* by default. However, the management of the *Spatial_ preferences* by the mobile operator is useful since it can reuse its existing location areas without defining new ones according to subscriber preferences (its existing location areas are zone areas used for the mobility management of the voice service).

4.3 Second Case: Prioritised Third Party Policy

Some legal organisations have the right to access sensitive information and to override the operator policy and user preferences. But to be effective, operator organisations have to define how and in which cases that policies are prioritised. Let *legal_ org* be the role of legal organisations. Operator assigns those organisations to that role thanks to the rule:
$\textbf{Rule}_{role\_assignment}$
$Use(mobile\text{-}operator, legal\_org, role\_assignment)$
Data collector shall define the contexts when those organisations are allowed to enforce their policies. Two contexts correspond to this situation: *legal interception* and *emergency.* The former represents the case where there is a legal decision made by a court. The requestor should justify that decision before accessing the sensitive information. The *emergency* state requires an immediate

access to the sensitive information. The proof of emergency can be delayed after the access since the location information can be used for example to save lives or prevent unwanted effects. *Emergency* is a provisional context because it implies an obligation to be fulfilled by the legal organisations after accessing private data. The obligation consists of a proof of the *emergency*, such as voice records of an emergency call. Thanks to the $\textbf{Rule}_{licence}$, the *administrator_ location* adds the *legal* licence:
$Permission(mobile\text{-}operator, legal\_org, read, location, emergency) \leftarrow use(mobile\text{-}operator, legal, licence) \wedge authority(legal, mobile\text{-}operator) \wedge grantee(legal, legal\_org) \wedge privilege(legal, read) \wedge target(legal, location) \wedge context(legal, emergency)$
That is the providers that act in *legal_ org* role, can read the objects belonging to the *location* view in the *emergency*. A similar rule applies to manage the *legal interception* context.

This latter permission can introduce conflicts with the data owner preferences. For example, suppose that *Bob* is a privacy fundamentalist who refuses to let any requestor read its location information. The OrBAC model derives the privilege:
$Prohibition(mobile\text{-}operator, service\text{-}provider, read, Bob\text{-}location, default)$
That is all subjects assigned to the *service-provider* role are prohibited to *read* objects in *Bob-location* view. This will lead to a conflict between the privileges given to the legal organisations and the data owner preferences. We propose to manage such conflicts through the Prioritised OrBAC model [8]. The strategy to solve conflicts in OrBAC is the assignment of priorities to security rules. Privileges with higher priority take precedence over the other security privileges. We first consider a set Π of priority levels associated with a partial order relation $\prec$. The OrBAC model is then enriched by the following predicates. $O - Permission(org, r, a, v, c, p)$ and $O - Prohibition(org, r, a, v, c, p)$ define an organization permission or prohibition respectively and are associated with a priority *p*. The legal organisations have a policy that overrides the data owner preferences and take precedence over the mobile operator policy itself.

Let Π be $\{p1, p2\}$, the set of priority levels. In the prioritised OrBAC, privacy fundamentalist preferences are expressed as follows:
$\forall r \in R, \forall a \in A,$
$O - Prohibition(mobile\text{-}operator, r, a, fundamentalist\text{-}location, default, p1)$
This privilege prohibits any role to access their location information. However, the *legal-organisation* policy has to override such preferences. The following privilege defines their policy:
$\forall a \in A, \forall v \in V$
$O - permission(mobile\text{-}operator, legal\text{-}organisation, a, v, emergency, p2)$
By applying the separation constraints and the potential conflict condition, there is a conflict between previous rules. To prevent such case, priorities will be: $p1 \prec p2$.

The priority assignment is ensured by the administrator. When a legal organisation specifies its policy, the mobile operator shall accept it but it replaces privileges predicates by adding the priority component which is higher than the priorities assigned to data owners.

4.4 Third Case: Policy Negotiation through the O2O Approach

In this section, we will use the interoperability approach to allow the mobile operator to negotiate the access control policy of the location information with service providers based on the privacy preferences of the data owners. We will consider that data owners are the managers of Virtual Private Organizations (VPOs). Each VPO controls the sensitive information of one data owner. As for the first case, the service level agreement (SLA) signed between users and the mobile operator will be used. It guides how the interoperability is controlled.

O2O Basis. O2O is based on virtual organisations. When an organisation aims to cooperate with another organisation, it creates a VPO. Each organisation, that needs to interoperate with other organisations, has to manage the access control and has to administrate its VPO. So, the administration of a VPO is totally decentralized [7].

First, in O2O we mention that there are three kinds of organisations:

- *O_grantor*: it is the organisation that owns the resource. We name its policy the local policy,
- *O_grantee*: it is the organisation that requires resource access,
- *VPO*: it is the organisation that administrates the interoperability policy between two organisations. The VPO controls the privileges of the O_grantee when accessing O_grantor resources.

Always, the organisation that provides the resource is the one that administrates the security policy of the VPO. This represents its authority space. We differentiate between the authority and the managing spaces:

- Authority space: an *O_grantee* organisation is in the authority space of another *O_grantor* organisation if the security policy, enforced by the *O_grantee*, is defined and administrated by the O_grantor,
- Managing space: by default, the *O_grantor* organisation manages its interoperability policy but it can delegate this task to another entity. So the managing space includes the entities that manage the interoperability policies of an organisation.

Assigning subjects to roles, actions to activities and objects to views, must comply with the following constraints to preserve the *O_grantor* control:

- The subjects belong to the *O_grantee* organisation. Thus, the VPO is defined to control the access of subjects, which belong to *O_grantee*, and require resources from *O_grantor*,
- The objects belong to the *O_grantor* organisation,
- The actions are under the control of the organisation that provides the access. This organisation is the *O_grantor*, but *O_grantee* can initiate them [7].

The goal of the VPO is to extend the authority of the grantor organisation to resources that need interoperability with other organisations. So, the security

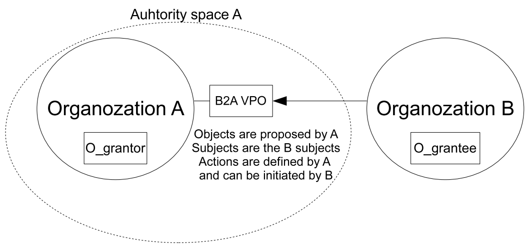

Fig. 4. The O2O approach

policy of the VPO is derived from the local policy of the O_grantor. This policy will manage the access requests coming from other organisations, named *O_grantee*. Since OrBAC enables the definition of hierarchies, the VPOs can be seen as sub-organisations of the *O_grantor*.

Proposed Interoperability Approach for Distributed Privacy Policy Management. From the point of view of the operator, each subscriber can be seen as an organisation, called subscriber organisation. We argue that the mobile operator is the central trusted entity that manages all subscribers' policies because it holds the sensitive information of its subscribers. Each subscriber organisation will create a VPO for each service provider. Service providers are the organisations that require the access to the VPOs of the subscriber organisations.

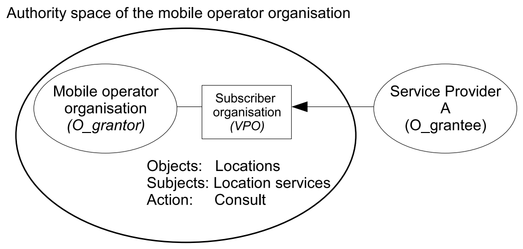
Authority space of the mobile operator organisation
Mobile operator organisation (O_grantor)
Subscriber organisation (VPO)
Service Provider A (O_grantee)
Objects: Locations
Subjects: Location services
Action: Consult

Fig. 5. Management of the subscriber organisation within the operator organisation

According to the O2O approach, the O_grantor is the operator organisation and the O_grantee is the service provider. Each data owner constitutes its own VPO. It defines its privacy policy thanks to the privacy administrative views defined before. The operator also specifies the interoperability policy of that user thanks to the SLA signed with her when she subscribes to the location service. It is a decentralised manner to define the privacy policy. However, the management of the VPOs, belonging to the subscribers of the organisation, is centralized since the operator is the central entity (it is a trusted one).

Policy Propagation and VPO Hierarchy. We stressed the fact that the policies of the VPOs are deduced from the local policy of the *O_ grantor*, which is the mobile operator in our case. This means that the dominant policy, which is the local policy that we specified in the first administration, is the root policy of those VPOs. A VPO encompasses the privacy preferences of one mobile subscriber according to the privacy package fixed by the SLA. When, the service provider, which is the *O_ grantee*, will access the VPO, it has to enforce the policy of this VPO. By this manner, when another requestor connects to the service provider organisation, the privacy policy of the subscriber will be applied to it. So, it is propagated to the *O_ grantee*. The first administration and O2O approach cooperate to spread user preferences.

The definition of a hierarchy of VPOs simplifies the management of the VPOs of the data owner and the specification of the SLA agreement. The data owner will define common privacy preferences within the root VPO, say data_ownertoProvider that is derived from the local policy. The remaining privacy parameters, which depend on the service provider organisation (and are based on the SLA), will be entered to a dedicated VPO, which is a sub-VPO of that data_ownertoProvider. So, the data owner specifies a fine-grained privacy policy for those sub-VPOs.

For example, the data owner can define different accuracies for sensitive data whereas other policy parameters, such as consent requirement and purpose specification, are the same for all service providers. Let Alice be a data owner within the operator organisation. She has three different privacy policies depending on the service provider a, b or c. So, Alice's root VPO has three sub-VPOs allowing the data owner to define different data accuracies depending on service providers.

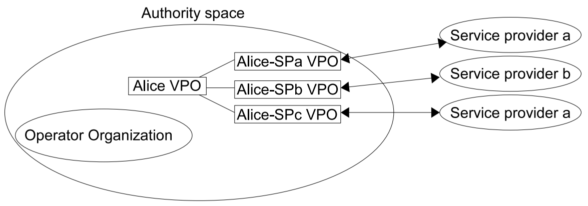

Fig. 6. VPOs hierarchy

This approach simplifies the spreading of the privacy requirements thanks to O2O. By contrast, related works fail to propose a complete framework to propagate user's preferences. In next section, we compare our administration proposal to related works.

5 Related Works

Existing privacy models and languages focused on the definition of the privacy components and how they are expressed but fail to propose a distributed administration model to allow the interoperation between user preferences, data collector practices and legal interception.

The Platform for Privacy Preferences (P3P) [19] is a declarative language that allows web sites specify the fair information practices. The web sites indicate through P3P which personal information is collected and how it will be used. The privacy policy can be attached to some web resources, like web pages and cookies. User agent will compare the P3P privacy policy to client preferences and decides about the access of web sites to client data. P3P defines a standard base data schema, which is a set of data elements that all P3P user agents should understand [19]. P3P defines also a vocabulary for defining privacy practices and statements. It allows essentially web sites to publish their privacy practices in both machine and human readable formats that can be treated by user agents. The ultimate goal of P3P is the automation of the decision in client side through P3P user agent instead of reading privacy policy at every site's visit. It does not provide means to negotiate user preferences and fair information practices. User agent, on behalf of users, uses the P3P Preference Exchange Language (APPEL) [20] to compare user preferences with the fair information practices of web sites. A user can express her preferences in a set of preference-rules, known as ruleset. So, user agent can run the semi-automated or automated decision about the privacy policy from P3P enabled web sites. User agent could be Web browsers, browser plug-ins or proxy servers. The policy evaluation is made by the user agent locally, so the approach does not consider the distributed administration tasks neither a whole policy that includes web site's practices and the user's preferences.

Qui Ni et al. proposed RBAC extensions to incorporate constraints and conditions that are related to privacy. They define a family of privacy-aware RBAC (P-RBAC) models. Privacy policies are expressed thanks to permission assignments. Those permission assignments differ from permissions in RBAC because of the presence of new components: purposes and conditions of the access. A customized language, LC_0, was proposed to allow the definition of conditions. A privacy permission explicitly defines: the intended purposes of the action, under which conditions, and what obligations have to be performed after the access. This work also develops conflict analysis algorithms to detect conflicts among different permission assignments. So, the three main extensions are: purpose component, obligation definition and a dedicated language for conditions. The privacy permission assignment is modelled through privileges, which have the general form: $role \times action \times data \times purpose \times conditions \times obligation$.

In our work, we reason differently about contexts. We explicitly define several types of contexts that belong to different administrative privacy views. We proposed also a distributed administration to take into account operator and requestor requirements. To our best knowledge, these issues were not addressed before.

6 Conclusion

In this paper, we proposed an administration framework of privacy policies. We identified three cases where a privacy policy is administrated differently. The resulting administration is distributed since requirements are issued by different entities. The three alternatives are not mutually exclusive. It means that the data collector can deploy all of them simultaneously. Each of them provides precise functionalities and cooperates to provide all privacy needs.

In this paper, we assumed that owners trust the mobile operator since there is already an agreement between them. We should investigate the case where there is no agreement, and if the data owners can sign an SLA on the fly to define how their privacy preferences will be managed. The specification of the SLA is planned in future work.

References

1. 3rd Generation Partnership Project: Open Service Access; Application Programming Interface (API); Part 3: Framework, 3GPP TS 29.198-3
2. Acquisti, A., Grossklags, J.: Privacy Rationality in Individual Dicision Making. IEEE Security and Privacy 1(1), 26–33 (2005)
3. Ajam, N., Cuppens, N., Cuppens, F.: Contextual Privacy Management in Extended Role based Access Control Model. In: The Proceedings of the DPM workshop, DPM-ESORICS, Saint-Malo, France (September 2009)
4. Ben Ghorbel-Talbi, M.: Decentralized Administration of Security Policies, PhD thesis, Télécom Bretagne (2009)
5. Ben Ghorbel-Talbi, M., Cuppens, F., Cuppens-Boulahia, N., Bouhoula, A.: An Extended Role-Based Access Control Model for Delegating Obligations. In: Fischer-Hübner, S., Lambrinoudakis, C., Pernul, G. (eds.) Trust, Privacy and Security in Digital Business. LNCS, vol. 5695, pp. 127–137. Springer, Heidelberg (2009)
6. Byun, J., Bertino, E., Li, N.: Purpose Based Access Control of Complex Data for Privacy Protection. In: Symposium on Access Control Models and Technologies (SACMAT), Stockholm, Sweden, pp. 102–110 (2005)
7. Coma, C.: Interopérabilité et Cohérence de politiques de sécurité pour les Systèmes Auto-organisant, PhD thesis, Télécom Bretagne (2009)
8. Cuppens, F., Cuppens-Boulahia, N.: Modeling Contextual Security Policies. International Journal of Information Security 7(4), 285–305 (2007)
9. Cuppens, F., Miège, A.: An Administration Model for Or-BAC. International Journal of Computer Systems Science and Engineering 19(3), 151–162 (2004)
10. Cuppens, F., Cuppens-Boulahia, N., Coma, C.: O2O: Virtual Private Organizations to Manage Security Policy interoperability. In: Bagchi, A., Atluri, V. (eds.) ICISS 2006. LNCS, vol. 4332, pp. 101–120. Springer, Heidelberg (2006)
11. Cuppens, F., Cuppens-Boulahia, N., Ben Ghorbel, M.: High Level Conflict Management Strategies in Advanced Access Control Models. Electronics Notes in Theoretical Computer Science, vol. 186, pp. 3–26. Elsevier, V., Amsterdam (2007)
12. Duckham, M., Kulik, L.: Location Privacy and Location-aware Computing. In: Dynamic and Mobile GIS: Investigating Change in Space and Time, pp. 34–51. CRC press, Boca Raton (2006)
13. Gabillon, A., Capolsini, P.: Dynamic Security Rules for Geo Data. In: Garcia-Alfaro, J., Navarro-Arribas, G., Cuppens-Boulahia, N., Roudier, Y. (eds.) Data Privacy Management and Autonomous Spontaneous Security. LNCS, vol. 5939, pp. 136–152. Springer, Heidelberg (2009)

14. Gedik, B., Liu, L.: Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms. IEEE Transactions on Mobile Computing 7(1), 1–18 (2008)
15. Abou El Kalam, A., El Baida, R., Balbiani, P., Benferhat, S., Cuppens, F., Deswarte, Y., Miège, A., Saurel, C., Trouessin, G.: Organization Based Access Control. In: Proceedings of the 4th International Workshop on Policies for Distributed Systems and Networks (Policy 2003), Como, Italy (June 2003)
16. Krumm, J.: A Survey of Computational Location Privacy. Journal: Personal and Ubiquitous Computing 13(6), 391–399 (2008)
17. Ni, Q., Trombetta, A., Bertino, E., Lobo, J.: Privacy-aware Role Based Access Control. In: 12th ACM symposium on Access control models and technologies, Session Privacy management, pp. 41–50 (2007)
18. Spiekermann, S., Grossklags, J., Berendt, B.: E-Privacy in Second Generation E-Commerce: Privacy Preferences versus Actual Behaviour. In: Proceedings of the ACM Conference Electronic Commerce (EC 2001), Florida, USA, pp. 38–47 (October 2001)
19. World Wide Web Consortium (W3C), The Platform for Privacy Preferences 1.0 (P3P) Specification (April 2002)
20. World Wide Web Consortium (W3C), A P3P Preference Exchange Language 1.0 (APPEL), Working draft (April 2002)
21. Yang, N., Barringer, H., Zhang, N.: A Purpose-Based Access Control Model. In: The third International Symposium on Information Assurance and Security, pp. 143–148 (2007)