Protecting Privacy during a Pandemic Outbreak

Karsten Martiny[1], Linda Briesemeister[1], Grit Denker[1], Mark St. John[2] and Ron Moore[2]
[1]*SRI International, 333 Ravenswood Ave, Menlo Park CA 94025, U.S.A.*
[2]*Pacific Science & Engineering, 9180 Brown Deer Rd, San Diego CA 92121, U.S.A.*

Keywords: Privacy, Access Control, Data Sharing, Policies, Reasoning.

Abstract: Respecting privacy is a major challenge when sharing data among enterprises. Because of the high stakes and complexity, enterprises need expressive and clear methods for defining tailored data sharing so that they can share the right information with the right partners with confidence. This paper describes a privacy-oriented declarative policy framework together with an intuitive user interface to manage data sharing policies. It introduces a use case about a pandemic outbreak, in which the system can be used to share relevant information with partners to ensure that help can be coordinated quickly and effectively, while at the same time ensuring that the privacy of individuals remains protected and not sharing overly widely. A set of privacy policies is introduced to describe how the policy system can meet these requirements.

1 INTRODUCTION

Respecting privacy is a major challenge when sharing data among enterprises. Enterprises, including law enforcement, hospitals, financial institutions, and military organizations cooperating on missions or disaster relief share large amounts of data. Much of these data are sensitive and private. Sharing sensitive, private data with the wrong enterprise partners could result in adversarial exploitation and serious harm to the enterprise and subjects of the data and potential civil and criminal liability. On the other hand, withholding the data necessary for a cooperative mission could hamper the execution of the mission, preempt its success, and incur irreparable damage to the joint goals that brought the cooperative partners together.

Additionally, the policies set in place by enterprises, regarding which data to share with whom, are often complex. Enterprises may have thousands of data types and hundreds of partners. Enterprise policies are likely to contain varying scopes and levels, such as broad baseline policies as well as detailed exceptions that tailor data sharing for specific objectives. Further, because enterprises typically have multiple objectives, policies may overlap and interact. Creating and managing enterprise policies, therefore, is likely to be challenging.

Because of the high stakes and complexity, enterprises need expressive and clear methods for defining tailored data sharing so that they can share the right information with the right partners with confidence. Building on an existing privacy-oriented declarative policy framework (Martiny, Elenius, & Denker, 2018) together with an intuitive user interface (St. John, Moore, Martin, Gustafson, Jaramillo, Denker, Martiny, & Briesemeister, 2018, Briesemeister, Gustafson, Denker, Martin, Martiny, Moore, Pavlovic, & St. John, 2019), this paper describes a use case about a pandemic outbreak, in which privacy policies are used to share relevant information with partners to ensure that help can be coordinated quickly and effectively, while at the same time ensuring that the privacy of individuals remains protected and not sharing overly widely. Based on this use case, the paper analyses how a declarative policy framework provides useful capabilities to aid in modeling complex data sharing scenarios through simple policy specifications. The framework focuses on reasoning about data sharing decisions based on a common, implementation-agnostic data model. This allows for an easy integration of various data retrieval operations in existing systems, e.g., SQL queries, MongoDB queries, HTTP Rest calls, etc.

Demonstration videos of using our system for this and other use cases can be found at https://sunflower.csl.sri.com/index.php/projects-sponsors/prisms/.

The remainder of the paper is structured as follows. We conclude this section with a brief discussion of related work. A more comprehensive

Martiny, K., Briesemeister, L., Denker, G., John, M. and Moore, R.
Protecting Privacy during a Pandemic Outbreak.
DOI: 10.5220/0010186603080318
In *Proceedings of the 7th International Conference on Information Systems Security and Privacy (ICISSP 2021)*, pages 308-318
ISBN: 978-989-758-491-6

comparison of our framework with existing approaches can be found in (Martiny et al, 2018). Then, Section 2 describes the key components of the data sharing system relevant for managing policies and making policy-based data sharing decisions. In Section 3 we describe in detail a use case in which the data sharing system can be used to manage sensitive data during a pandemic outbreak, including a characterization of the use case's objectives and explanations how implemented policies achieve these objectives. Afterwards, Section 4 provides an analysis of our experiences and results from developing and implementing this use case. Finally, the paper concludes with Section 5.

A number of machine-readable privacy policy languages exist to protect access to sensitive information. Most notable are Ponder (Damianou, Dulay, Lupu, & Sloman, 2001) EPAL (Ashley, Hafa, Karjoth, Powers, & Schunter, 2003), Rei (Kagal, Finin, & Joshi, 2003), KAoS (Uszok, Bradshaw, & Jeffers, 2004) AIR (Kagal, Hanson, & Weitzner, 2008), SecPAL (Becker, Fournet, & Gordon, 2010) XACML (OASIS XACML Standard, 2013) and PRISMS (Martiny et al, 2018). A common feature of all of these languages is that they provide some means of privacy protection through role-based access control policies. Based on their specification of affected data, these contributions can be categorized into different approaches: In XACML, SecPAL, and Ponder the unique resources targeted by policies need to be explicitly specified, and requested objects need to exactly match a specified policy object in order to trigger a policy decision. If several related resources are to be shared in these systems, dedicated policies have to be specified for each resource, leading to a large number of very similar policies. Other approaches such as EPAL specify policies based on category labels (such as "location data"). This approach significantly reduces the number of required policies to share a set of similar records. However, it only provides very coarse ways of characterizing data, and thus does not allow for fine-grained tailoring of access policies. KAoS, Rei, and AIR on the other hand are expressive enough to represent rich relationships between targeted resources. They achieve their power by essentially exposing a complete logic language to the policy author, who is left to define the precise semantics of each policy from scratch. This makes the process of specifying policies much more challenging and much less accessible to non-experts.

In PRISMS, policy authors use a use case specific data model together with a general, use case independent shareability theory. This allows the policy framework to generalize to a high degree, which in turn enables the author to specify expressive policies in a concise way, capturing their intent through intuitive graphical interfaces without requiring extensive knowledge of the underlying specification formalism. We adopt the PRISMS framework for our pandemic use case.

2 SYSTEM COMPONENTS

The overall information system architecture is depicted in Figure 1. Similar to the architecture of XACML, our high-level architecture is separated into a policy administration point (PAP) where policies are authored, policy decision point (PDP) where data requests are evaluated against policy, and a policy

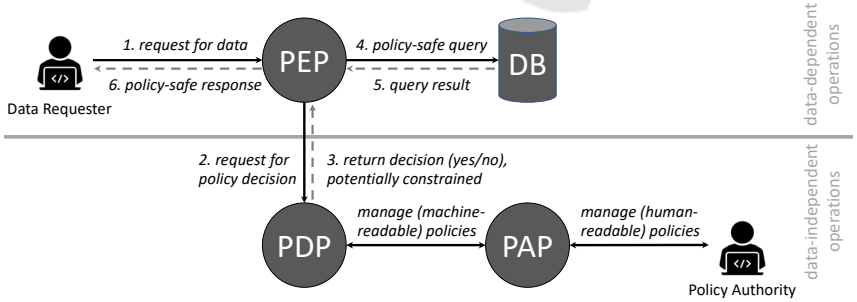

Figure 1: The overall system architecture: 1. Data requesters send requests for data to the Policy Enforcement Point (PEP). 2. Upon receiving a request for specific data, the PEP will ask the Policy Decision Point (PDP) for a decision with respect to the currently defined policies. 3. The PDP will return a result, which consists of an allow or deny decision and optionally additional constraints for this decision. 4. The PEP will use the result from the PDP to rewrite the request into a policy-safe database query and send this query to the actual database. 5. After receiving results from the data base, the PEP potentially applies additional post-processing steps and then 6. returns the final result back to the requester. The Policy Administration Point (PAP) provides a UI that transforms machine-readable policies into human-readable policies that can be managed by the policies' author, here after called the policy authority.

enforcement point (PEP) where those decisions are enforced. This architecture supports the separation of enforcement from decision making, which is particularly relevant for the integration of additional privacy-preserving technologies, as described below.

2.1 The Common Data Model

Our architecture uses a Common Data Model (CDM) as an intermediate representation of the type and organization of data managed by the system. This data model is used as the basis of communication between different components of our architecture. To illustrate the concepts of the CDM, Figure 2 depicts a very simple example: The main building block of the data model are classes that represent certain types of data, for example, Figure 2 contains two classes Community and `Person`. Classes can have *data properties* such as `name` for the `Community` class, and `lastName`, `firstName`, `birthdate`, and `gender` for the `Person` class. These data properties are the representations of information that could actually be shared with requesters.

The CDM also models connecting properties such as the resident property shown in Figure 2. This property expresses that the class `Community` has a property `resident` which points to the class `Person`, i.e., it provides a connection between these two classes. We use the convention of denoting class names with capital letters and property names with lower-case letters to easily distinguish these concepts. We also use a dot notation to denote properties of a certain class, e.g., `Community.name` refers to the `name` property of the `Community` class. This notation can also be used to express paths through the CDM, e.g., `Community.resident.lastName` represents the last names of communities' residents. A lot of times connecting properties have counterparts to express the inverse relatio nship, e.g., in Figure 2 the Person class has a property

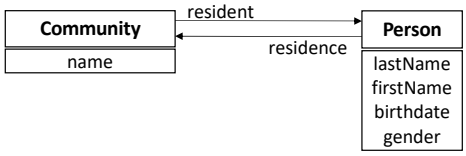

Figure 2: A very simple CDM: The classes are `Community` and `Person`, with data properties *`name`*, and `lastName`, `firstName`, `birthDate`, and *`gender`*, respectively. The classes are connected through *`Community`*'s connecting property `resident`, and its corresponding inverse property `residence` (belonging to the class `Person`).

`residence` pointing to `Community`, i.e., `Person.residence` is the inverse of `Community.resident`. Thus, consequently, `Person.residence.name` represents community names where people reside.

Another feature of the CDM is the ability to express class hierarchies. This feature is not shown here, but it is used in the pandemic use case and shown in Figure 4. This allows the specification of taxonomies of objects, for example, we can have data requesters, and epidemiologists as a more specific subclass of data requesters.

Based on the CDM, *Joined Data Sets (JDS)* can be defined to specify the data addressed by policy specifications. For instance, an (informally represented) JDS `{Community.name, Community.resident.lastName}` specifies that a policy addresses the joined release of community names together with the last names of the communities' residents.

2.2 The Policy Decision Point

The Policy Decision Point (PDP) is the central component for making sharing decisions for incoming requests. It receives access requests from the Policy Enforcement Point (PEP) consisting of information for the *who*, *what*, and *when* of a request, that is, it specifies a requester, the requested data (represented as a JDS), and the time of the request. The PDP then infers a sharing decision for this request in two steps: First, it checks for each of the existing policies whether they match the given request specification and collects the corresponding decisions. Second, if multiple decisions are collected, it evaluates the interplay of different policies by analyzing defined overriding criteria to derive a final sharing decision which is sent back to the PEP.

To determine whether the requested data matches the data specifications of a given policy, the PDP uses a *shareability theory* that models generic relations between different, but related JDSs: For instance, our shareability theory states that if a policy allows sharing of a certain JDS, then it also allows all requests for *subsets* of this JDS. For example, if a policy allows sharing of above JDS combining persons' last names and their community of residence, this also allows a request for just persons' last names as this is a subset of the allowed data specification. Conversely, if a policy specifies that access to a certain JDS is forbidden, this automatically also applies to all supersets of the specified JDS. For example, if a policy denied access to above JDS, it would also deny requests for any

Figure 3: A view of the semantic data model via the CDM explorer tool. The user has navigated to the `Community` class and selected the `name` data property. Furthermore, the user has linked to residents of the community via the `resident` connecting property of `Community`. Residents are persons and the user has chosen two data properties to share: `firstName` and `lastName`. The user has also selected the link to the person's Medical Information to share the data property `diseaseStatus`. In summary, the graphical tool allows the Policy Authority to specify the JDS of `{Community.name, Community.resident.firstName, Community.resident.lastName, Community.resident.medicalInformation.diseaseStatus}`.

superset such as requesting persons' last names, their community of residence, and their medical status. Moreover, the shareability theory also characterizes additional matching relations between the requested and the specified JDSs; (a) if the request is for a subclass of a class that is specified in the policy's JDS, and (b) if the request is for the inverse of a relation that is specified in the policy's JDS. For example, if a policy allows sharing of information about "communities and their residents," it will also permit the equivalent request for "persons and their residence community," because the properties "residents" and "residence" are inverse properties of each other. This quality allows the use of a single, concise specification of a policy's data, while at the same time ensuring that this specification matches a large variety of related requests.

If multiple policies match a given request, the PDP uses a set of defined overriding criteria to determine the final decision. Overriding criteria can be flexibly defined according to the use cases' needs. In the use case that we introduce below, we illustrate two commonly used criteria: *precedence-based overrides* allow one policy authority to specify multiple policies with different priorities, so that in case of conflicts the policy with the highest priority is chosen. Additionally, *hierarchy-based overrides* can be used to model chain of authority relations and give superior policy authorities the ability to override decisions from subordinate authorities.

Next to issuing allow/deny decisions, the PDP is also able to attach constraints to issued decisions. Constraints can be of two different types: *Filter constraints* specify criteria that must be met by returned data, and *action constraints* specify post-processing actions to be performed on the retrieved data before returning it to the requester.

We will illustrate all of these aspects in detail in Sections 3 and 4.

2.3 The Policy Administration Point

The policy administration point (PAP) provides a user interface for the policy authority to create, edit, and review the effects of their data sharing policies. A policy manager component displays a list of policies, including their names, a short description and their precedence level (higher precedence policies override lower precedence policies). The policy manager allows policy authorities to create new policies, edit or delete existing policies, or enable or disable existing policies. Disabled policies remain in the system, but they are not used to compute data sharing decisions.

Creating or editing a policy uses a *four-step software wizard*. In step one, the policy is given a title and a short description. The policy can also be given a timeframe for when it enables and disables. Finally, the policy designates whether to share or deny sharing data. In step two, data requesters for the policy are selected. Data requesters are specified based on attributes. The UI automatically offers choices for all attributes defined in a given use case. In our Pandemic use case, data requesters have attributes for their affiliation (community or nation), and their role (response coordinators, care providers, and epidemiologists). Policy authorities can also define "supergroups" based on specific attributes, such as a set of response coordinators from certain nations responding to a disaster. If a policy references a supergroup whose membership changes, then the policy automatically shares or denies sharing with the updated membership.

In step three of the wizard, the policy authority can select the data shared or denied by the policy. Data may be specified as simply all data available in the system or using semantic properties of the data, such as a JDS from a common data model. We developed the CDM explorer tool to support the selection of

JDSs. In addition to JDSs, policies may also refer to documents, map layers, or other data formats, depending on what data formats are available in the system.

The CDM explorer (see Figure 3) is a graphical tool for navigating through a data model, such as a semantic network. Policy authorities start exploring the semantic network by choosing a root class and then traverse through classes via connecting properties (e.g., `Community.resident` as the first edge). While visiting class nodes, the policy authority can select data properties (e.g., `Person.lastName`) available on each class and thus creates a connected set of paths through the network to define a JDS. In addition to stepping through connecting properties, policy authorities can navigate through class and subclass hierarchies, for example, both `Epidemiologist` and `Response Coordinator` are subclasses of the `DataRequester` class. They can also define constraints on selected data, such as requiring the date of birth to be prior to a certain date or requiring the community of residence to equal a particular community.

Finally, in step four of the wizard, the policy is given a precedence level, and there is a final review of the policy specification.

3 THE PANDEMIC USE CASE

The Pandemic use case was developed to provide a context for researching, engineering, and demonstrating privacy-preserving technologies and user-facing interfaces. The Pandemic use case involves monitoring a disease outbreak in which a number of communities share data about their citizens and their citizens' disease status, governed by policies that a number of policy authorities have defined, and a number of different types of users requesting data relevant to their roles in this use case. Specific data sharing policies are enforced to demonstrate a range of policies and technologies. This use case serves as a good representative of real-world data sharing tasks among multiple nations and organizations that happen on a regular basis. It exhibits privacy challenges on various levels regarding the privacy of individuals' data, as well as inter- and intra-organizational data exchanges.

Within our pandemic scenario, a typhoon has caused extensive damage across a set of countries in the Pacific and relief (food, medicine, water, shelter, etc.) is needed in a number of communities. Adding to the complexity of the situation is the outbreak of a deadly and highly infectious disease that begins working its way through the populace.

The operational objective in this use case is to predict the progression of a major disease outbreak through the impacted communities and to take steps to counter it. This thread introduces the challenge of protecting personally identifying information (PII) within medical records of individuals in the impacted communities while providing access sufficient to enable accurate characterization of the disease and its spread.

3.1 Actors

Actors in this use case belong to one of three different fictional island nations: Cebu, Bohol, and Siquijor. Each nation is comprised of different communities, e.g., Cebu City is one of Cebu's communities. Every nation and every community has its own policy authority, thus enabling data sharing on both the communal and national level. Moreover, this situation induces a hierarchy of authorities, where a (superior) nation policy could potentially override decisions of associated (subordinate) community policies.

Requesters are categorized into three different roles: response coordinators, care providers, and epidemiologists. While care providers need detailed medical information about individuals, response coordinators and epidemiologists need larger-grained data such as counts of infected persons in each community and nation. Requesters are also associated with a specific nation or community. In our use case we model one nation-level response coordinator and one care provider for each of the defined nations, and one community-level response coordinator and care provider, respectively, for each of the defined communities. Moreover, each nation has an epidemiologist to analyze the outbreak and evolution of the pandemic based on SIRD (Susceptible-Infected-Recovered-Dead) data.

3.2 Data

The data relevant for this pandemic use case is depicted in the Common Data Model (CDM) diagram shown in Figure 4. It is focused on residents' personal information (name, birth date, gender) together with their medical information. Moreover, persons' relations to communities and nations are modeled through their resident and citizenship relations, respectively. Every community belongs to one nation, and both communities and nations have policy

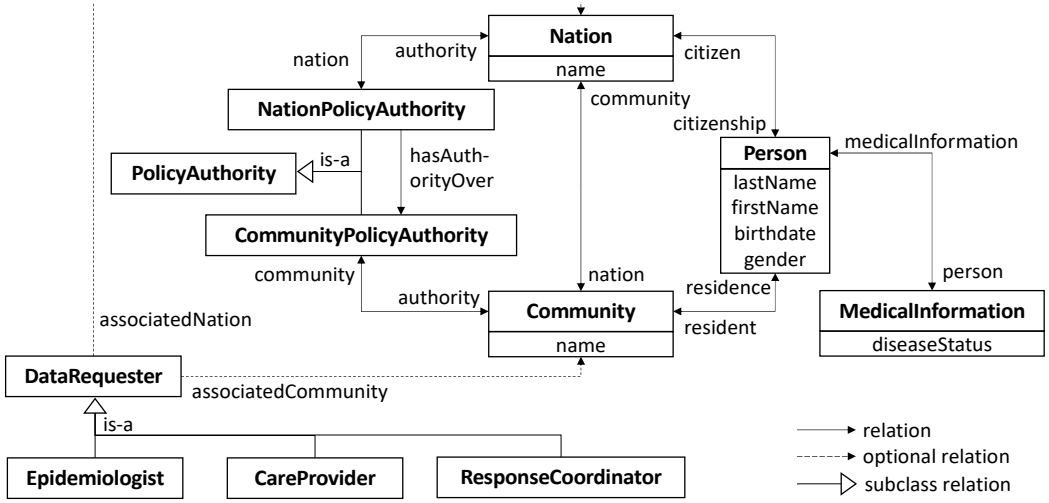

Figure 4: The Common Data Model used in the pandemic use case: Classes names are represented in boldface and start with capital letters; class property names are represented in regular face and start with lowercase letters. Data is centered around nations, their communities, and persons. Persons are citizens of a country, residents of a community, and have medical information. Policy authorities (PAs) are categorized into nation and community PAs, and each nation and each community has a corresponding PA. Nation PAs have authority over their subordinate community PAs. Data requesters are associated either with a nation or with a community and are categorized into epidemiologists, care providers, and response coordinators.

authorities associated with them. One important property of this data model is that there is a transitive residence relation between persons and nations that traverses through community, i.e., if a person is a resident of a community, she is also a resident of the country that the community belongs to. However, the target relation of this transitive residence relation is not necessarily the same nation as the target of the person's citizenship relation. This is a crucial difference for the policies discussed below.

Next to the data being shared, the data model also models data requesters, which can be categorized into epidemiologists, care providers, and response coordinators. Data requesters are either associated with a specific nation or with a specific community, represented by the corresponding properties. Again, for data requesters associated with communities, the associated nation can be determined via a transitive relationship, but no data requester has simultaneously defined properties for associated nation and associated community, thus these relations are marked as optional in Figure 4.

3.3 Policies

A set of data sharing policies was created for this pandemic use case to showcase how data can be shared efficiently with different requesters. The formal policy specifications are machine readable and are used by the policy engine, they are not meant for humans to read. User interactions are performed through the intuitive interface for policy creation, management and review (St. John et al, 2018, Briesemeister et al, 2019). Because of space limitations, we will not provide formal specifications for policies, but rather highlight the main features of our framework by example. Technical details for the formal policy specifications can be found in (Martiny et al, 2018).

For descriptive purposes, the policies of this use case are broken into three vignettes:

Vignette 1. The goal of Vignette 1 is to define policies that share aggregate information (counts of people per disease status: *S* - susceptible, *I* - infected, *R* - recovered, and *D* - deceased). These counts are being shared aggregated at the nation level with *all* response coordinators and aggregated at each community level only with the response coordinators of the corresponding nation. The policies of Vignette 1 make use of various aspects of our powerful policy language and semantics: (a) the ability to define exactly what data to share (or not to share) through JDS path specifications (in particular, in CDMs with many relations between data, there are many subtleties that need to be addressed. The related work cited above does not address these subtleties), and (b) the ability of detailed tailoring with whom data is shared. Vignette 1 contains the following policies:

- **Policy 1: All Nations Allow Sharing of Nation-level Aggregated Disease State Information of their Residents with Response Coordinators.**

This policy specifies that all members of the class `NationPolicyAuthority` allow sharing of data with all members of the class `ResponseCoordinator`, i.e., response coordinators of any level are allowed to see data that is released on a national level. The data addressed by this policy can be informally represented as the joined set of paths (cf. Figure 4) `{NationPolicyAuthority.nation.name, NationPolicyAuthority.nation.community.person.medicalInformation.diseaseStatus}`, i.e., it allows to jointly release the names of nations together with the disease state of its residents.

There are several important aspects to note about this data specification which distinguish the discussed approach from previous work on access control: First, there are different paths through the common data model to connect nations with medical information of persons, namely by traversing through nations' communities and their residents (as this policy does), or alternatively through nations' citizens. The former specification represents all *residents* of a nation, while the latter represents all its *citizens*. It is worth noting that neither of these concepts subsumes the other. In particular, this policy does not allow to share data of a nation's citizens that reside in another country.

Second, even though the specified path traverses *through* communities, it does not allow to share any information *about* communities. Consequently, even though the community class is part of the part specification, information about community associations will not be released to requesters, and thus response coordinators will only get information on the national level.

Lastly, this policy intends to release only aggregate count information about disease states. Usually, this would require the specification of action constraints to make sure that only aggregate-level information is released. However, since the released data only contains categorical information (SIRD), it is not necessary to require additional constraints in this case – the non-aggregated data means that for each of the four categories (SIRD) there will be multiple entries. If one were to count the number of entries per category, one gets the aggregate count. In other words, this policy allows to release a less compact representation of count information (not yet the final count, but lists of entries), but in the end it does not reveal any more information than the final count.

- **Policy 2: All Nations Allow Sharing of Community-level Aggregated Disease State Information of their Residents with Own Response Coordinators.**

The intent of this policy is for each nation to provide its own response coordinator with more fine-grained information about the distribution of disease states. Since this information is mainly required to coordinate aid on an intra-nation level, it is not shared on an inter-national level.

To accomplish this sharing goal, a slightly modified version of the previous policy is used: First, instead of allowing access for all response coordinators, a restriction is added that the data requester's property `associatedNation.name` must be equal to the policy authorities `nation.name` property.

Second, one more shared property is added to the joined data set: `{NationPolicyAuthority.nation.name, NationPolicyAuthority.nation.community.person.medicalInformation.diseaseStatus, NationPolicyAuthority.nation.community.name}`, i.e., the path between nations and persons is still the same as in the previous policy, but this policy additionally allows to share the names of the communities that serve as the intermediate nodes. Consequently, each released disease status can now be associated with a specific community and thus enables the intended community-level aggregation.

Vignette 2. The goal of of Vignette 2 is to define policies that will generally prohibit sharing of personal data, but carve out a specific exception, namely, to share personal data about adults with one's own national response coordinator and one's own community's care provider. To simplify the discussion, this vignette focuses on one particular community (Cebu City) and its associated nation (Cebu). The policies of Vignette 2 make use of various aspects of our powerful policy language and semantics: (a) ability to override policies based on different overriding criteria, (b) shareability theory and superset/subset semantics of decisions to succinctly capture several equivalent data representations, and (c) using constraints to exactly specify what data is made available. Vignette 2 contains the following policies:

- **Policy 3: Cebu City Denies Sharing Personal Data of Residents with Anyone, Precedence = 0.**

Policy 3 is a baseline policy that locks down data sharing for all requesters. By choosing the class `DataRequester` to specify to whom this policy applies, and since that is the root or super class of all data requester classes, this policy applies to all data requesters in the data model. To express that no personal data is shared, the policy is a deny policy that defines `{Community.resident}` as the data that will not be shared. Looking at the CDM, `Community.resident` points to the `Person` class. Even though this is a very succinct description, it does exactly what is intended, namely not sharing any personal data of residents. Because of the shareability theory that underlies our framework, not only the resident property, but also its reverse property `Person.community` is addressed with this policy and, secondly, because of the semantics that for a deny policy, all supersets of the specific data are also denied. Since we deny access to Person in general with this policy, we also deny access to any property of persons, making it impossible to share any meaningful personal data.

- **Policy 4: Cebu City Allows Sharing of Personal Identifiable Information (PII) and Medical Status of *Older* Residents (over 14 Years Old) with *Cebu Nation* Response Coordinators and *Cebu City* Care Providers, Precedence = 1, Specifically the Shared Joined Data Set is Nation, First Name, Last Name, and Medical Status, Plus a Filter Constraint to only Share Persons Whose Date of Birth is Prior to 2006.**

To facilitate effective coordination of a response to the disease, Cebu City decides to make an exception to its default policy of not sharing any personal data: it now creates a policy to share all relevant information (name, gender, birthdate, disease status). Policy 4 has a higher precedence than Policy 3 and overrides it, but note that Policy 4 does not *replace* Policy 3. The overriding mechanism described in Section 2.2 ensures that the policy with the highest priority is selected if there exist multiple policies with opposing decisions from a single policy authority. But there might still be many other records related to residents (not shown in our CDM) and in these cases the newly defined policy to release medical information does not apply and thus the default deny policy (Policy 3) would still ensure that this information is appropriately protected.

As the data shared by Policy 4 contains personal identifiable information (PII), the requesters are further limited to the national response coordinator. Thus, Policy 4 only shares these data with specific requesters rather than all requesters (i.e., the requester that is a member of the `ResponseCoordinator` class *and* has the property `associatedNation(Cebu)` and to the own community care providers, i.e., the requesters that are members of `CareProvider` and have the property `associatedCommunity(CebuCity)`).

Due to the sensitive nature of this data, Cebu City decides to protect the privacy of children even stronger and thus only shares data of persons that are over 14 years old. Our framework supports policies that tailor sharing in sophisticated ways, such as defining constraints to identify exactly which data are shared. This is an example of a filter constraint: the decisions issued by the PDP now have an additional constraint "allow if age > 14."[1] Note that the PDP itself does not have access to the actual data (cf. Figure 1), but attaches this constraint to the decision so that the PEP can filter the retrieved data accordingly before returning results to the requester.

- **Policy 5: Cebu Nation Allows Sharing Medical Status of *All* of Cebu City's Residents with Cebu City Care Providers, Precedence = 0.**

Policy 5 demonstrates another override. Due to the dire situation, Cebu Nation, a national policy authority with higher rank than Cebu City policy authority, has written a policy to share all Cebu City residents' medical status. Even though this policy has low precedence, the higher rank of policy authority prevails. Thus, the effect of this additional policy is that the targeted requesters will now be allowed to receive medical PII of all residents including children.

The policy specification is similar to the previous one, the differences are: (a) this policy is specified by the Cebu nation PA (instead of the Cebu City Community PA), (b) the targeted requesters are limited to Cebu City's care provides (now excluding the national response coordinator), and (c) there are no constraints on the persons' age.

[1] This is simplified for presentation purposed, technically the constrained would be expressed as a constraint on the difference between the current date and the birth date.

Vignette 3. The goal of Vignette 3 is to share not accurate, but differentially private disease state counts at community level. Differentially private counts mean that not exact aggregates are shared, but rather approximate aggregate counts. This provides more privacy while still providing enough benefit to the consumers of such data (e.g., care providers). For example, in order to know how many resources need to be provided at a certain location, approximate counts of sick people are usually sufficient. On the other hand, exact counts (together with auxiliary information) could reveal individual private data (Sweeney, 2000). Thus, being able to integrate privacy-preserving technologies with the policy framework is very attractive to policy authorities as an additional means for fine-grained control and protection of privacy. The policy of Vignette 3 makes use of the ability to integrate our policy framework with privacy-preserving technologies and illustrates how data sharing policies can be written to use those.

- **Policy 6: Share Differentially Private Disease State Community-level Counts with Epidemiologists. Precedence = 1, Specifically the Shared Joined Data Set is Nation, First Name, Last Name, and Medical Status, Plus An Action Constraint To Fuzz up the Counts using Differentially Private Technologies.**

The intention of this policy is to provide helpful information to epidemiologists researching the pandemic outbreak. Disease state information on a community level could be very helpful for epidemiologists across the globe to better understand how the virus spreads and how the infections proceed. However, sharing data globally with epidemiologists significantly opens up the potential audience (and thus significantly lowers the trust the released data will not be exploited). Even if only community-level count aggregates of diseases status are released, this still bears the danger of potentially allowing for the re-identification of specific individuals. To ensure that personal information of individuals is sufficiently protected, the released data should be protected with differential privacy. This is specified through an *action* constraint: Opposed to filtering constraints, these types of constraints don't restrict the records to be released, but instead they prescribe necessary post-processing actions to be applied to the data before it is returned to the requester. In this particular policy, it specifies that differential privacy must be applied to the disease status counts before they are released to epidemiologists. This ensures that epidemiologists receive useful data for research purposes while at the same time protecting the privacy of affected individuals.

The policy demonstrates the incorporation of differential privacy technology into the system. For large data sets, it would be very difficult for anyone to guess the medical status of an individual based on the counts. However, sharing aggregate status counts over much smaller data sets, creates the possibility that requesters could begin to infer the status of individual residents. Similar sensitivities have been demonstrated in the current COVID-19 pandemic. Positive cases are typically reported for whole cities or sometimes whole zip codes, but finer aggregates, or the status of individuals, are rarely if ever reported.

To combat this concern, differential privacy can be used to add enough noise to the shared counts to guarantee that individual status cannot be guessed to a desired low probability. In this way, differential privacy can be used to encourage more data sharing by reducing fears that sensitive data may be guessed from shared statistics.

4 RESULTS

The use case described in the previous section shows how a fairly small set of policies can be used to facilitate sophisticated data sharing that is tailored to multiple simultaneous requirements. This capability is enabled by making use of several key features of the described system:

1. Using Joined Data Sets to concisely specify not only *which* data attributes are shared, but also *how they are connected* provides an easy method to have fine-grained control over exactly what is shared. This is well-illustrated by the policies in Vignette 1: The goal is to share information that connects nation names with medical information about persons. However, without explicitly specifying how these attributes are to be connected, the policies' intention cannot be fully captured: if both properties are shared in isolation, the result is barely useful because the connections between person names and medical status is lost. Allowing any combination of these two attributes on the other hand would result in oversharing, because it would not only allow the (intended) sharing of nations' residents' information, but also the (unintended) sharing of nations' citizens' information. Moreover, Policy 2 shows that the detail level of the data (aggregation on nation-level vs. community-level) can be changed easily by

allowing for additional joined attributes in the shared data.

2. Defining policies based upon a shareability theory allows the separation of general-purpose specifications of data relations (such as subsets, inverse properties, etc.) from situation-specific policies. This approach significantly eases policy specifications and reduces potential errors because the policy authority can focus on the variable parts and does not have to think through "common sense" implications of policy specifications. For instance, the shareability theory specifies that if the sharing of some data set is allowed, then sharing of all corresponding subsets of data is also allowed. For example, if a response coordinator is allowed to see community-aggregated data due to Policy 2, this response coordinator is also allowed to see nation-level aggregated data, even if the nation-aggregated policy were disabled. Moreover, considering inverse properties provides the ability to address multiple different but equivalent requests in a single policy. For instance, if a policy allows sharing of information about "communities and their residents", it will also permit the equivalent request for "persons and their residence community."
3. Addressing requesters in terms of attributes allows for a very flexible and scalable way of tailoring decisions to a wide variety of potential requester types. Essentially, every element in the CDM that can be connected to the requester class can be used to address targeted requesters. For instance, both policies in Vignette 1 specify that members of the *ResponseCoordinator* class are allowed to see data, but the second policy additionally requires that the value of the requester's *associatedNation* property matches the PA's nation. As a result, if response coordinators request community-level aggregated medical data, they will get a positive decision only from their own nation, while a request for nation-level aggregated data will result in positive decisions from all nation PAs.
4. Making use of the overriding feature allows sophisticated interactions between policies. This ability facilitates specifying a succinct set of policies that can yield a variety of different decisions in different circumstances and also allows flexibility in defining exceptions to policies without requiring any changes to existing policies. For instance, Policy 3 in Vignette 2 specifies that no personal data of residents should be shared with anyone. Next, Policy 4 defines that an exception should be made under very specific circumstances, i.e., specific personal medical information of residents over the age of 14 may be shared with the nation's own response coordinators and care providers.

   Furthermore, hierarchy-based overrides show how existing authority hierarchies can be used to achieve centrally coordinated decisions within a chain of authority. This ability is shown by Policy 5, where the Cebu Nation policy defines another policy that removes the restriction from Cebu City's sharing policy for specific requesters without disabling Cebu City's policy. As a result, all three policies in Vignette 2 are still valid and serve requests from different requesters.
5. Two different types of constraints allow policy authorities to further restrict access to and thereby protect the privacy of the accessed data. Filter constraints, such as requiring the date of birth to be prior to some date, enable the release of only subsets of the requested data that meet that constraint. The age restriction used in Policy 4 ensures that the (more sensitive) data of young children is filtered out in corresponding requests. Action constraints on the other hand prescribe additional post-processing actions that need to be performed at the PEP before data is returned to the requester. An example of an action constraint is the requirement in Policy 6 to apply differential privacy before data is released to the requester. A noteworthy property of this constraint mechanism is that neither type of constraint is evaluated at the PDP, but instead, data sharing decisions containing constraints are computed on an abstract level without requiring access to actual data, and then passed to the policy enforcement point. This approach to constraints makes the architecture particularly well-suited to privacy-protecting technologies: first, the PDP can operate completely without access to actual data, which reduces the risk surface of potential privacy violations. Second, and more importantly, this approach enables the integration of privacy-enhanced, encrypted data bases that are able to perform query operations on the encrypted data and only the final result gets decrypted, e.g., the data of children in Vignette 1 or the non-noisy data in Vignette 2 will never be decrypted. That is, a constraint computed at the PDP becomes a part of the encrypted query to the encrypted database, and the constrained data never needs to be unencrypted by any part of the system.

These sophisticated sharing policy features are combined with an easily accessible user interface (St. John et al, 2018, Briesemeister et al, 2019) to make

policy specifications available to lay users. This intuitive user interface allows policy authority users to make use of the entire arsenal of policy features without requiring detailed knowledge about the inner workings of the policy decision point nor any specific technical specification skills.

5 CONCLUSION

Enterprises need to share a wide variety of information with their partners to pursue their objectives. Their dilemma is that these data are often sensitive, and protecting privacy is important. One importance piece for resolving this dilemma is to provide fine-grained specifications of exactly which data to share with which partners so that only needed data is shared. The methods for specifying data sharing need to be expressive both for specifying the data and specifying the requesters who may access the data. Furthermore, the methods need to be clear and easy to use by non-experts so that errors are rare and easy to catch and using the methods does not require specialized training.

Our methods meet both criteria. Here, we described these methods in the context of a pandemic use case. Policy authorities define data sharing policies that specify which persons' medical data, or counts of those data, are shared with different classes of data requesters.

Our methods include a sophisticated JDS specification of which data types to share and what constraints to apply, a shareability theory-based approach to processing requests for subsets, supersets, and inversely specified requests, an expressive role-based specification of data requesters, and a decision process that incorporates both precedence-based and policy authority hierarchy-based overrides. Importantly, this policy decision point requires no access to the data contents in order to makes these policy-based sharing decisions.

Enterprises using these methods may come to share more data and thereby realize more objectives because they can be confident that they can precisely control which data are shared with who and how and which data remain private from all others. This enhanced sharing should be useful in a wide variety of context, and vital in global emergences, such as pandemics, where the appropriate, tailored sharing of sensitive information is crucial.

Distribution Statement "A" (Approved for Public Release, Distribution Unlimited).

REFERENCES

Martiny, K., Elenius, D., Denker, G., 2018. Protecting Privacy with a Declarative Policy Framework. In *12th IEEE International Conference on Semantic Computing (ICSC)*, Laguna Hills, California, USA

St. John, M., Moore, R., Martin, A., Gustafson, W., Jaramillo, M., Denker, G., Martiny, K., Briesemeister, L., 2018. Enterprise-Level Private Data Sharing: Framework and User Interface Concepts. In *9th International Conference on Applied Human Factors and Ergonomics (AHFE)* Orlando, Florida, USA.

Briesemeister, L., Gustafson, W., Denker, G., Martin, A., Martiny, K., Moore, R., Pavlovic, D., and St. John, M, 2019: Policy Creation for Enterprise-level Data Sharing. In *21st International Conference on Human-Computer Interaction (HCI)*, Orlando, Florida, USA.

Myers, K., Ellis, T., Lepoint, T., Moore, R., Archer, D., Denker, G., Lu, S., Magill, S., Ostrovsky, R., 2017. Privacy Technologies for Controlled Information Sharing in Coalition Operations. In *9th International Conference on Knowledge Systems for Coalition Operations*, Los Angeles, California, USA

OASIS XACML Standard, Version 3.0, 2013 http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html, last accessed 2020/06/12

Damianou, N., Dulay, N., Lupu, E., Sloman, M., 2001.The Ponder Specification Language. In *Policy 01: Workshop on Policies for Distributed Systems and Networks*.

Ashley, P., Hafa, S., Karjoth, G., Powers, C., Schunter, M.: Enterprise Policy Authorization Language, http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/, last accessed 2020/06/15

Kagal, L., Finin, T., Joshi, A., 2003. A Policy Language for a pervasive Computing Environment. In *Policy 03: 4th International Workshop on Policies for Distributed Systems and Networks*.

Uszok, A., Bradshaw, J., Jeffers, R., 2004. KAoS: A policy and domain services framework for grid computing and semantic web services. In: *iTrust 2004, Second International Conference on Trust Management*.

Kagal, L., Hanson, C., Weitzner, D., 2008. Using dependency tracking to provide explanations for policy management. In 2008 IEEE Workshop on Policies for Distributed Systems and Networks, pp. 54–61.

Becker, M., Fournet, C., Gordon, A., 2010. SecPAL: Design and semantics of a decentralized authorization language. In: *Journal of Computer Security*, vol. 18/4, pp. 619–665.

Sweeney, L., 2000. Simple Demographics Often Identify People Uniquely. In *Data Privacy Working Paper 3*. Pittsburgh.

St. John, M. F., Pankova, A., Denker, G., Laud, P., Martiny, K. & Pavlovic, D., 2020. Decision Support for Sharing Data Using Differential Privacy. *Manuscript/Under Review*.