CV-Priv: Towards a Context Model for Privacy Policy Creation for Connected Vehicles

Yunxuan Li
University of Stuttgart
Stuttgart, Germany
yunxuan.li@ipvs.uni-stuttgart.de

Pascal Hirmer
University of Stuttgart
Stuttgart, Germany
pascal.hirmer@ipvs.uni-stuttgart.de

Christoph Stach
University of Stuttgart
Stuttgart, Germany
christoph.stach@ipvs.uni-stuttgart.de

***Abstract*—Connected vehicles are becoming progressively capable of collecting, processing, and sharing data, which leads to a growing concern about privacy in the automotive domain. However, research has shown that although users are highly concerned about their privacy, they usually find it difficult to configure privacy settings. This is because the privacy context, which represents the privacy circumstance a driver faces during the privacy policy creation, is highly complex. To create custom privacy policies, drivers must consider the privacy context information, such as what service is requesting data from which vehicle sensor, or what privacy countermeasures are available for vehicles and satisfy certain privacy properties. This easily leads to information and choice overhead. Therefore, we propose the novel ontology-based privacy context model, *CV-Priv*, for the modeling of such privacy context information for creating custom privacy policies in the automotive domain. In this paper, we analyze the design requirements for a privacy context model based on challenges drivers might face during the privacy policy creation phase. We also demonstrate how *CV-Priv* can be utilized by context-aware systems to help drivers transform their fuzzy privacy requirements into sound privacy policies.**

***Index Terms*—Context Modeling, Ontology, Privacy Policy, Privacy-Preserving, Connected Vehicle**

I. Introduction

In the era of big data, almost every digital device is producing myriads of data every day. Connected Vehicles (CVs), which are equipped with modern applications and are capable of accessing the internet as well as interacting with other smart devices, can collect and produce more than 20 GB of data within just one hour [1]. These data contain not only sensitive information about the underlying CV but also drivers' personal data, such as their home locations or driving behaviors. Thus, as Parkinson *et al.* [2] state in their research, although it is not clear what personal data will be generated by CVs, it is clear that all possible measures must be taken to preserve privacy.

Despite the desire of drivers to protect their sensitive data, their general demand is to continue utilizing as many functions provided by services as possible. This poses the so-called "privacy paradox" [3], i.e., although people claim that they are concerned about their privacy, they still share a lot of private information. To overcome this paradox, i.e., to help drivers protect their privacy without decreasing service functionality, we proposed a situation-aware privacy-preserving framework for CVs (SAPP4CV) in our previous work [4]. With SAPP4CV, drivers can create custom Privacy Policies (PPs) for individual data types, services, and situations. The context of a CV also plays an important role in the PP since the sensibility of the same CV data may change as the situation changes [5], which could lead to a change in the driver's privacy requirements. For each PP, SAPP4CV also allows drivers to determine particular Privacy Enhancing Technologies (PETs) [6] that can generate distorted or anonymized data which should still ensure the service functionality.

However, as Bahirat *et al.* [7] argue in their study, although users are highly concerned about their privacy, they usually find it difficult to configure PPs. In general, drivers only have fuzzy privacy requirements for services in their CVs. For instance, most drivers have the privacy requirement to protect their home location from a certain application. However, to configure a custom PP, they might not know what relevant data need to be protected or which PET is suitable for this purpose. This is because the privacy context for creating custom PPs in the automotive domain is highly complex.

Similar to the vehicle context which describes the situation a CV is in, the privacy context represents the privacy circumstance a driver is facing. This includes information, such as what service is requesting which vehicle data, what PETs are available for CVs, and the desired privacy goal a driver wants to achieve. Without a sufficient understanding of the relevant privacy context, it is challenging for drivers to create PPs on their own. Hence, the design of the context model for such privacy context is essential to help drivers transform their fuzzy privacy requirements into formal and sound custom PPs.

In this paper, we propose the novel ontology-based privacy context model *CV-Priv* that describes the necessary privacy context information for creating custom PPs. *CV-Priv* captures concepts from the automotive and privacy domains as well as their interrelationships, which supports the discovery of fitting configurations for components of a PP (e.g., the appropriate PET) from drivers' fuzzy privacy requirements. This helps drivers to create custom PPs without considering complex privacy context information.

The rest of the paper is structured as follows: In Sect. II we state the challenges a driver might face during the privacy policy creation and then outline the requirements for a privacy context model in Sect. III. In Sect. IV, we present the *CV-Priv* Ontology and demonstrate its usage. In Sect. V, we review the related work and we summarize the paper in Sect. VI.

II. Problem Statement

In this section, we outline the challenges during the privacy policy creation phase for CVs based on a motivating scenario.

Motivating Scenario: Consider the driver Alex, whose privacy goal is to protect his or her home location from the application `NaviApp` while continuing to utilize the navigation service provided by it. In order to provide its service to Alex, `NaviApp` requests navigation data (e.g., current location, speed, and heading sensed by GPS receiver) from the vehicle's `Navigation` component. In terms that the navigation data is not available (e.g., the vehicle is in the tunnel), `NaviApp` also requests speed data (sensed by the gearbox) from the vehicle's `Transmission` component to estimate the vehicle's current location. To create the PP, Alex only has a rough idea. As the location data cannot be related to Alex's home location when the vehicle is away from home, Alex wants to create a PP only in the situation where s/he is close to home (`NearHome`). This allows Alex to utilize the full navigation service of `NaviApp` when the CV is not `NearHome`. When `NearHome`, Alex is willing to sacrifice some service quality of `NaviApp` with regard to accuracy to protect the location data. However, to transform this idea into a concrete and sound PP, Alex faces the following challenges:

- **C_1: Express privacy goal precisely:** Although privacy is a widely used term, Solove [8] argues that "nobody can articulate what it means". In this example, Alex's privacy goal of "protecting home location" can be either interpreted as keeping it confidential (hiding the data content [9]) or anonymous (hiding the link between the information and the identity [9]). To create a concrete PP, Alex must declare the meaning of "protect" precisely.
- **C_2: Identify relevant vehicle data:** While the privacy goal appears to be only related to CV's current location, it is not sufficient to merely protect the location data. It is clear that the current location can be retrieved from the location data. However, the current location can also be inferred through the vehicle's speed and heading data as well as time information. Thus, all vehicle data, from which the location can be derived, must also be protected.
- **C_3: Understand technical and privacy features of PETs:** To select PETs that satisfy their privacy goals and still ensure certain service functionality, drivers must have an understanding of PET's technical specifications, such as the feasible data type that a PET can operate on, and their privacy features, such as the data quality of the generated data and the privacy assurance of a PET.
- **C_4: Understand technical requirements of services:** Similar to $\mathbf{C_3}$, drivers ought to understand technical requirements from the service side, such as the necessary vehicle data and the minimum data quality required for the functionality. Furthermore, drivers should be aware of the exact data sources (e.g., GPS receiver), from which a service requests source vehicle data. This allows the selected PET to be only applied to the data stream of affected data sources.

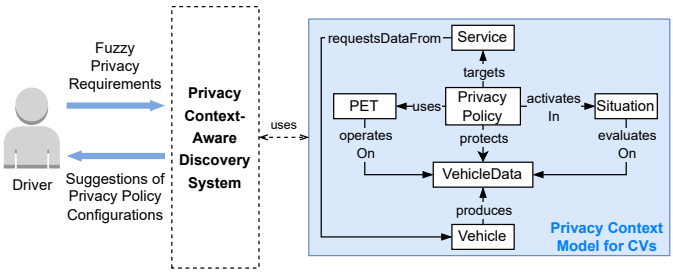

Figure 1: Privacy Policy Creation with Context Model

While the fuzzy privacy requirements of a driver may seem simple, creating custom PPs can be very complicated. Because of these four challenges, it is rather easy for drivers to create flawed PPs where sensitive information is still disclosed as relevant data types are not protected, or the desired service functionality is undermined as the selected PETs are improper.

To assist drivers with the creation of formal and sound PPs, we argue for a privacy context-aware discovery system (depicted in the dashed box in Fig. 1). This kind of discovery systems should have the ability to take drivers' fuzzy privacy requirements as input, and reply to them with suggestions for suitable configurations of their PPs.

III. Requirements for the Privacy Context Model

To enable such a discovery system, necessary concepts from the privacy and automotive domains as well as their relationships for creating custom PPs must be modeled in a machine-readable way. Thus, we see a need for a privacy context model for CVs focusing on the creation of custom PPs. This context model can then be utilized as the back-end or knowledge base for the discovery system. Note that the goal of this paper is to create the privacy context model, which we call *CV-Priv*, that supports the discovery system but not to develop the discovery system itself.

The basic layout of the privacy context model for CVs is illustrated in Fig. 1 (the blue box). In general, a custom PP allows its user to select desired PETs for specific data types and services. Thus, concepts of *Driver*, *PrivacyPolicy*, *PET*, *VehicleData*, and *Service* must be modeled. As introduced in Sect. I, the situation is essential for PPs, hence, *Situation* must also be modeled. Moreover, we consider *Vehicle* as another core concept, since all vehicle data originate from CVs. In the following, we detail the requirements for each identified concept considering the four challenges discussed in Sect. II:

- *Driver*: models the user of CVs (depicted as the user icon on the left side in Fig. 1). In order to distinguish between PPs created by different drivers for the same CV (e.g., car-sharing), the relationship between *Driver* and *PrivacyPolicy* as well as *Vehicle* must be modeled.
- *PrivacyPolicy*: models the custom PP for CVs. Each PP should allow drivers to choose the *PET* it uses, the *Service* it targets, the *VehicleData* it protects, and the *Situation* in which it activates. Considering $\mathbf{C_1}$, it should also empower drivers to declare their privacy goals precisely.

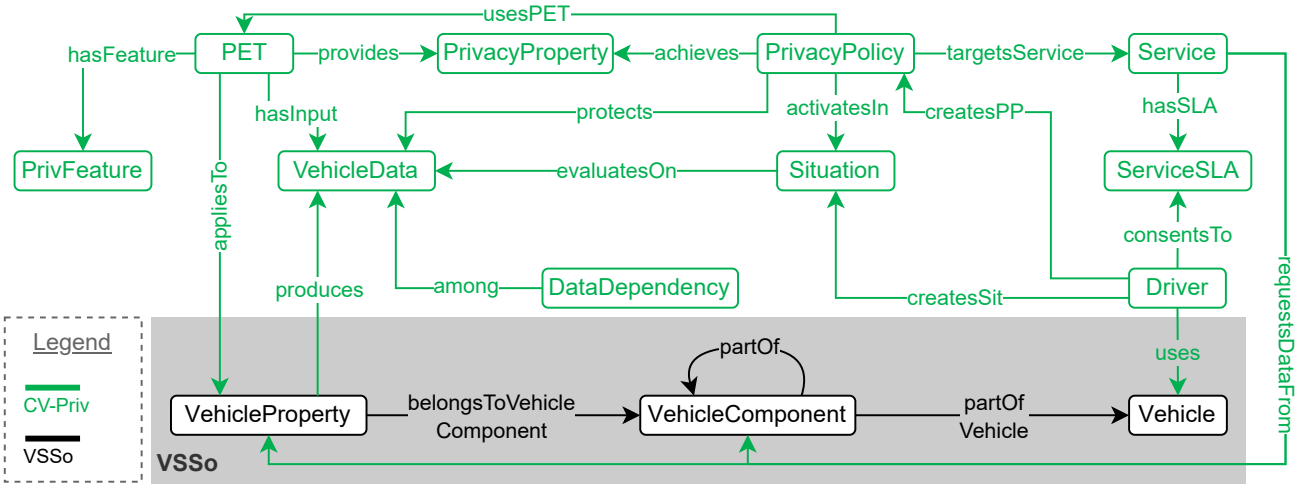

Figure 2: Structure of *CV-Priv* Ontology

- *Vehicle*: models the architecture of a vehicle. Different sensors in a vehicle as well as their signals and semantics should also be modeled.
- *VehicleData*: models different types of vehicle data and their semantics. Each vehicle data type should be semantically unique. As a complex system, it is common for a CV to have multiple sensors located in different vehicle components that measure the same phenomenon. For example, in the motivating scenario, both the GPS receiver (using coordinates) and gearbox (using rotation speed) can measure the speed of the vehicle. This semantic equivalence of signals from different sensors that measure the same phenomenon (e.g., speed) must be modeled by *VehicleData* so that queries, such as *"which sensors/signals measure the speed of the vehicle?"* can be answered. Moreover, it enables drivers to define PPs for different data types rather than specific vehicle signals. As explained in $\mathbf{C_2}$, many vehicle data types represent the physical context of a CV, thus, a strong dependency exists among them. *VehicleData* should also model this data dependency to enable the retrieval of information, such as *"which data types relate to the data type X?"*.
- *Situation:* models concrete situations a driver created for the *PrivacyPolicy*. This concept should also allow situation-aware systems, such as SAPP4CV, to evaluate defined situations based on the vehicle context.
- *Service:* models available services in a CV. As described in $\mathbf{C_4}$, the technical requirements of a service must be modeled. Other service metadata, such as the service description and availability, can also be modeled.
- *PET:* models available PETs in the automotive domain. As described in $\mathbf{C_3}$, both technical specifications and privacy features of PETs should be modeled. The privacy features of PETs must be modeled in a technical way so that a mapping between *PET* and *Service* can be established. Moreover, the privacy features must be modeled in a non-technical way that is straightforward for drivers to understand and that can match the privacy goal of drivers. The doubled modeling of the privacy feature should support queries, such as *"which PETs can ensure the functionality of service X and satisfy driver's privacy goal Y?"*. This enables drivers to find appropriate PETs without considering their technical details.

IV. CV-Priv Ontology

In this paper, we propose the novel ontology-based Privacy Policy Context Model for Connected Vehicles (*CV-Priv*) following the requirements outlined in Sect. III. To develop *CV-Priv*, we decide to use an ontology-based approach, as ontologies can formally describe concepts and their interrelationships in a data structure that is utilizable by computers [10]. Furthermore, ontologies support context reasoning and information retrieval through SPARQL queries, which allow *CV-Priv* to be utilized by context-aware systems to help drivers determine the proper configurations of their PPs.

A. CV-Priv *Ontology*

As depicted in Fig. 2, we reuse the knowledge from the Vehicle Signal Specification Ontology (*VSSo*) [11] for the modeling of vehicle-related concepts. The root element of *VSSo* is the *Vehicle* itself. From there, different parts of a vehicle are modeled as *VehicleComponent* in a tree structure and they serve as a sorting element for the leaf nodes. Each leaf node is a *VehicleProperty* that contains the semantic information of vehicle signals, which change in greater frequency (e.g., speed, location, etc.), and of attributes, which are more static (e.g., fuel type). For a comprehensive understanding of *VSSo*, we refer interested readers to the work of Klotz *et al.* [12].

CV-Priv can be seen as an extension of *VSSo* with extra concepts expressing privacy terms and technical details of services as well as PETs for creating custom PPs in the automotive domain. In Fig. 2, the green text describes classes and object properties from the domain *CV-Priv* whereas the black text represents the domain *VSSo*. In the following, we detail the definition of *CV-Priv* classes depicted in Fig. 2.

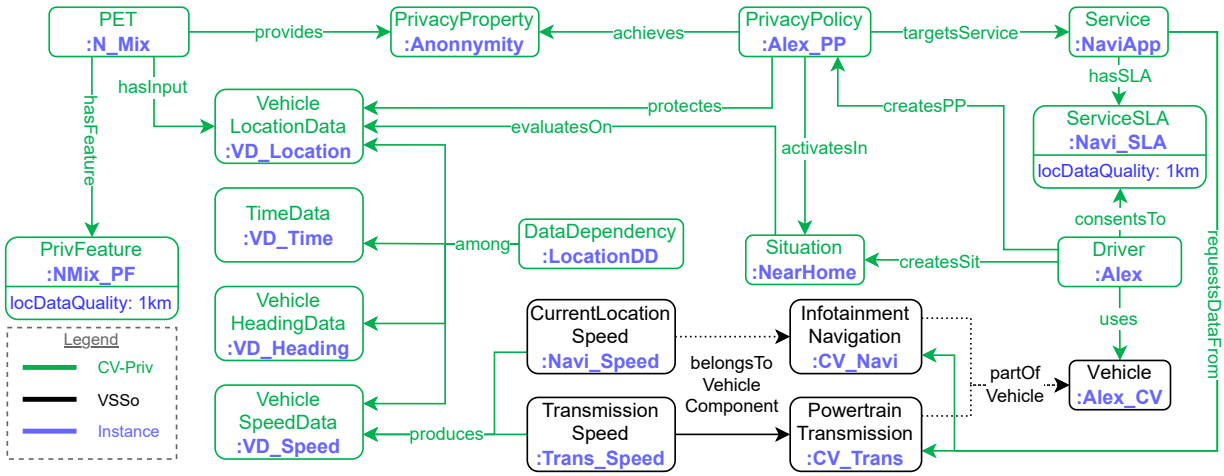

Figure 3: Excerpt of Motivating Scenario Instance

1) cv-priv:Driver: represents the user of a CV and the creator of a PP. In order to utilize applications installed in a CV, drivers must consent to the data requests as well as other terms and conditions stated in the applications' Service Level Aggrement (SLA). Furthermore, a driver can create multiple PPs for the CV s/he *uses*. Drivers can also create a specific *Situation* in which a PP should be activated.

2) cv-priv:PrivacyPolicy: represents custom PPs. As explained in Sect. III, this class allows drivers to select vehicle data types, situations, services, and preferred PETs for a PP. The creator of a PP is specified through the link *createsPP*. It is a generic and extendable class that allows the association with additional aspects for PPs, such as the access purpose.

3) cv-priv:PrivacyProperty: describes diverse properties of privacy. To assist drivers in describing their privacy goals precisely and to model the privacy features of PETs in a non-technical way, we utilize the privacy taxonomy [8], which categorizes different properties of privacy in a comprehensive and concrete manner. In *CV-Priv*, we use privacy properties summarized by Deng *et al.* [9] as a reference.

4) cv-priv:Situation: allows drivers to create particular situations for custom PPs. It is a generic class that allows the integration of other ontologies for a formal definition of a situation. It is also linked to *VehicleData* via *evaluatesOn*, which allows the evaluation of a *Situation* by situation-aware systems, such as SAPP4CV, using concrete values of *VehicleData* when the CV is in motion.

5) cv-priv:VehicleData and cv-priv:DataDependency: describes different and unique data types a CV can produce and the data dependency *among* them, respectively. Each unique vehicle data type is defined as a *rdfs:subClassOf* of *VehicleData* and is associated with at least one *vsso:VehicleProperty* that *produces* this kind of data. Thus, signals from different sensors that measure the same phenomenon (e.g., speed) are linked to a single vehicle data type (e.g., *VehicleSpeedData*).

6) cv-priv:Service and cv-priv:ServiceSLA: represents services or applications in a CV and their SLA, respectively. Both classes are generic and extensible classes for domain-specific ontology. Each *Service* is linked to at least one *vsso:VehicleComponent* or *vsso:VehicleProperty* from which it requests the source data of a vehicle. Service metadata as mentioned in Sect. III are modeled as attributes of *Service* and other technical requirements from a *Service* as described in $\mathbf{C_4}$ are modeled as attributes of *ServiceSLA* (omitted in Fig. 2).

7) cv-priv:PET and cv-priv:PrivFeature: The class *PET* represents different Privacy Enhancing Technologies for CVs. It is a generic concept and is extensible to ontologies from its own domain. The link to *vsso:VehicleProperty* allows drivers to apply a *PET* to concrete source data streams of vehicle signals. In *CV-Priv*, the feasible vehicle data types that a *PET* can operate on is modeled through the link *hasInput* to *VehicleData*. Other technical specifications of a *PET*, such as its parameters, are modeled as attributes of *PET*. The class *PrivFeature* is the technical modeling of the privacy feature a *PET* has. The possible privacy features of PETs, as described in $\mathbf{C_3}$, are modeled as attributes of *PrivFeature* (omitted in Fig. 2). Moreover, to enable the mapping between *Service* and *PET*, *ServiceSLA* should share the same attribute set as *PrivFeature* (example can be found in Fig. 3).

B. *Usage of* CV-Priv

To demonstrate the usage and expressiveness of *CV-Priv*, we model the motivating scenario using *CV-Priv*. An excerpt of the instance for the scenario is depicted in Fig. 3. Analogously, the black text represents *VSSo*, the green text represents *CV-Priv*, and the blue text indicates the instance of the classes. In the following, we first explain the instance more detail and then demonstrate the usage of *CV-Priv*.

For the vehicle structure, the navigation (`CV_Navi`) and transmission component (`CV_Trans`) which are relevant to the motivating scenario are modeled. As explained in Sect. III, both vehicle components have sensors that measure signals (`Navi_Speed` and `Trans_Speed`) which represent the current speed of the vehicle. This semantic equivalence is modeled by mapping the two signals to the same vehicle data type `VD_Speed` through *produces*. Note that several

VSSo object properties are shown in the dotted line, as some intermediate vehicle components are omitted in Fig. 3.

For the *VehicleData*, we demonstrate four different vehicle data types that build a data dependency `LocationDD`. This data dependency describes the fact that knowing the source data stream of any three data types within this dependency can lead to the inference of the value for the remaining data type. To avoid visual clutter, the source signals for these vehicle data types (except `VD_Speed`) are omitted in Fig. 3.

For the *Service*, we model the service `NaviApp` from the motivating scenario with one service-side requirement `locDataQuality` as the attribute of its SLA (`Navi_SLA`). This requirement denotes that to ensure its service quality, `NaviApp` requires a minimum data quality of location data that is in the 1km range of the exact vehicle location.

For the *PET*, we model a location obfuscation method `N_Mix` as introduced by Wightman *et al.* [13], which operates on location data (`VD_Location`), as an example. The privacy feature of `N_Mix` is modeled in a technical way with *PrivFeature* (`NMix_PF`), and in a non-technical way through the *PrivacyProperty* (`Anonymity`). In `NMix_PF`, the attribute `locDataQuality` represents the data quality of location data generated by `N_Mix`, which is also in a 1km range.

Finally, for the *PrivacyPolicy*, Alex first creates a situation `NearHome`, which describes whether the CV is in a given area around Alex's home. Then, Alex creates a custom PP (`Alex_PP`) and specifies it to protect the vehicle location data `VD_Location` from the service `NaviApp` in the situation `NearHome`. Furthermore, Alex selects the *PrivacyProperty* that `Alex_PP` should achieve to be `Anonymity`.

In the following, we demonstrate how *CV-Priv* can be used to assist Alex in improving the configurations of `Alex_PP` and finding suitable PETs with three SPARQL queries.

As mentioned in $\mathbf{C_2}$, it is not sufficient to merely protect location data if drivers want to hide their current location. To identify the relevant vehicle data, from which the location can be derived, the following SPARQL query Q_1 can be used. As depicted in Listing 1, Q_1 is used to find all relevant *VehicleData* that are in the same data dependency as `VD_Location`.

Listing 1: Q_1 - *which VehicleData relates to VehicleData X?*

```
SELECT ?relevantVD
WHERE {
    ?dataDependency cv-priv:among "VD_Location";
                    cv-priv:among ?relevantVD }
-----------------------------------------------------
RESULT: "VD_Speed", "VD_Heading", "VD_Time"
```

To determine if the relevant *VehicleData* needs to be protected, each returned vehicle data type should be further examined whether it is requested by the target service `NaviApp`.

Listing 2: Q_2 - *Does Service X request VehicleData Y?*

```
ASK {
    ?vehicleProperty cv-priv:produces "VD_Speed";
              vsso:belongsToVehicleComponent ?vc.
    "NaviApp" cv-priv:requestesDataFrom ?vc }
-----------------------------------------------------
RESULT: true
```

In Listing 2, we present a sample query Q_2 for `VD_Speed`. The results from Q_2 for each data type should then be processed or integrated. If `NaviApp` requests all *VehicleData* from the same *DataDependency*, Alex should be notified and advised to protect all relevant *VehicleData* in `Alex_PP`.

To retrieve appropriate PETs for `Alex_PP` that have input data type `VD_Location`, satisfy Alex's privacy goal `Anonymity`, and ensure the functionality of service `NaviApp`, query Q_3 depicted in Listing 3 can be used. For Q_3, we assume that the value (in meter) of the requirement `locDataQuality` from `NaviApp` is already queried.

Listing 3: Q_3 - *which PETs provide PrivacyProperty X, have input VehicleData Y, and ensure functionality of Service Z?*

```
SELECT ?pet
WHERE {
    ?pet cv-priv:hasInput "VD_Location";
         cv-priv:provides "Anonymity";
         cv-priv:hasFeature ?privFeature.
    ?privFeature cv-priv:hasAttr "locDataQuality".
    "locDataQuality" cv-priv:hasValue ?petDQVal.
    FILTER (?petDQVal < 1000) }
-----------------------------------------------------
RESULT: "N_Mix"
```

The result from Q_3 contains only PETs that satisfy Alex's fuzzy privacy requirement. This reduces the choice overhead for drivers regarding the selection of PETs. If multiple PETs are returned, Alex can choose a preferred PET based on the introduction provided by its developer.

C. Discussion

In this section, we assess whether *CV-Priv* is a suitable privacy context model for creating custom PPs in the automotive domain based on challenges summarized in Sect. II and requirements outlined in Sect. III.

CV-Priv is developed with the purpose to model the privacy context of creating custom PPs for CVs so that it can help drivers find sound and fitting configurations for PPs with less effort. Accordingly, *CV-Priv* support drivers to express their fuzzy privacy requirements more precisely ($\mathbf{C_1}$) with *PrivacyProperty*. Furthermore, *CV-Priv* introduces a data layer *VehicleData*, which abstracts from vehicle signals in *VSSo* and models the semantic equivalence of them. Thereby, drivers are empowered to create PPs for different data types without having to specify from which vehicle sensors data originate. Moreover, with *DataDependency*, *CV-Priv* support the discovery of all vehicle data types ($\mathbf{C_2}$), from which a certain vehicle property (e.g., location) can be inferred. By creating a technical mapping between *PET* and *Service* through a mutual attribute set as well as a non-technical mapping between *PET* and drivers' privacy goal through *PrivacyProperty*, *CV-Priv* enables drivers to discover suitable PETs ($\mathbf{C_3}$, $\mathbf{C_4}$) with minimum considerations of technical details and privacy context.

In *CV-Priv*, experts from different domains are required to model concrete privacy context, such as which vehicle data types belong to the same *DataDependency* or which data sources are requested by a certain service. In this paper, we modeled *N_Mix* as an example PET for the location data type.

For other data types, such as voice and image data, the state-of-the-art PETs, as described by Stach *et al.* [14], can be modeled.

With an instance of the motivating scenario, we demonstrate that *CV-Priv* enables the determination of whether there are other relevant vehicle data types a driver should protect regarding a given target service. The usage also presents that *CV-Priv* can be used to filter appropriate PETs for a given privacy goal of the driver and the target service successfully. Note that this filtering function can also be used contrariwise for drivers to discover suitable services that match their privacy preferences. For instance, with a given PET and the desired service functionality (e.g., as an attribute of *Service*), *CV-Priv* can be used to filter possible services that provide the functionality and that can work with the data quality delivered by the selected PET.

V. Related Work

As privacy is gaining importance in recent years, there has been an increasing amount of literature on ontologies for data privacy in different domains. Feld and Müller [15] proposed an automotive ontology for user-centered intelligent applications. The ontology can be roughly separated into two parts where the user model captures user-sensitive aspects and the vehicle model describes the general structure of vehicles and their physical context. It also introduces privacy specification as a meta concept in user preference. However, due to their limited focus on privacy, detailed privacy terms are not modeled.

Yankson [16] proposed a Privacy Integrated Context Ontology (PICO) for autonomous vehicles that ensures the functionality of advanced driver assistance systems whiles maintaining user privacy. PICO models the vehicle context with the integration of privacy elements in the vehicle's onboard computer and its subclasses. With reasoning, PICO maintains users' privacy policy concerning context knowledge. As the author takes the privacy policy as a given information, PICO does not consider the privacy context for creating these privacy policies.

Arruda and Bulcão-Neto [17] developed a lightweight ontology IoT-Priv defining the privacy layer upon IoT basic concepts. It captures essential privacy terms, such as privacy policy, data provider, and recipient. In IoT-Priv, the privacy policy is detailed into general policy and storage policy which allow data providers to specify concrete anonymization and cryptography techniques, respectively.

Gharib *et al.* [5] proposed another ontology COPri for a fine-granular definition of domain-independent privacy requirement. COPri also captures fundamental privacy terms, such as personal information, privacy goal, and privacy mechanism. In addition, it models privacy requirements as seven refined privacy categories. However, no attention was paid in both works regarding how to help a user comes to the decision of a comprehensive privacy policy or privacy requirement.

VI. Summary and Outlook

While the fuzzy privacy requirements of a driver may seem simple, creating formal and sound PPs requires drivers to consider complex privacy context information. With simple mistakes, the created PPs can be inadequate, which results in unwanted disclosure of personal data. In this paper, we point out four challenges drivers may face when creating custom PPs for their CVs and how they can be supported by the privacy context model *CV-Priv*. To develop *CV-Priv*, we reuse knowledge from *VSSo*. With a concrete usage of *CV-Priv*, we demonstrate how it can be utilized to assist drivers in finding fitting configurations for components of a PP.

As further work, we plan to implement a discovery system based on *CV-Priv* with a GUI to evaluate the interaction with drivers. We also plan to explore the possibility of modeling privacy context for concrete vehicles, services, and PETs automatically, for example, from their documentation.

References

[1] R. Coppola and M. Morisio, "Connected Car: Technologies, Issues, Future Trends," *ACM Computing Surveys*, vol. 49, no. 3, 46:1–46:36, 2016.

[2] S. Parkinson *et al.*, "Cyber Threats Facing Autonomous and Connected Vehicles: Future Challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 2898–2915, 2017.

[3] P. A. Norberg, D. R. Horne, and D. A. Horne, "The Privacy Paradox: Personal Information Disclosure Intentions versus Behaviors," *Journal of Consumer Affairs*, vol. 41, no. 1, pp. 100–126, 2007.

[4] Y. Li *et al.*, "Ensuring Situation-Aware Privacy for Connected Vehicles," in *Proceedings of the IoT '22*, 2023, pp. 135–138.

[5] M. Gharib, P. Giorgini, and J. Mylopoulos, "An Ontology for Privacy Requirements via a Systematic Literature Review," *Journal on Data Semantics*, vol. 9, no. 4, pp. 123–149, 2020.

[6] G. W. van Blarkom, J. J. Borking, and J. G. E. Olk, Eds., *Handbook of Privacy and Privacy-Enhancing Technologies: The Case of Intelligent Software Agents*. The Hague, The Netherlands: College Bescherming Persoonsgegevens, 2003, ISBN: 90-74087-33-7.

[7] P. Bahirat *et al.*, "A Data-Driven Approach to Developing IoT Privacy-Setting Interfaces," in *Proceedings of the IUI '18*, 2018, pp. 165–176.

[8] D. J. Solove, "A Taxonomy of Privacy," *University of Pennsylvania Law Review*, vol. 154, no. 3, pp. 447–564, 2006.

[9] M. Deng *et al.*, "A privacy threat analysis framework: Supporting the elicitation and fulfillment of privacy requirements," *Requirements Engineering*, vol. 16, pp. 3–32, 2011.

[10] T. Strang and C. Linnhoff-Popien, "A Context Modeling Survey," in *Proceedings of the UbiComp '04*, 2004, pp. 34–41.

[11] B. Klotz, R. Troncy, and D. Wilms, "VSSo: Vehicle Signal Specification Ontology," W3C, W3C Working Draft, Mar. 2022. [Online]. Available: https://www.w3.org/TR/2022/WD-vsso-20220303/.

[12] B. Klotz *et al.*, "VSSo - A vehicle signal and attribute ontology," in *Proceedings of the SSN '18*, 2018, pp. 56–63.

[13] P. Wightman *et al.*, "Evaluation of Location Obfuscation techniques for privacy in location based information systems," in *Proceedings of the LatinCOM '11*, 2011, pp. 1–6.

[14] C. Stach *et al.*, "Protecting Sensitive Data in the Information Age: State of the Art and Future Prospects," *Future Internet*, vol. 14, no. 11, 302:1–302:43, 2022.

[15] M. Feld and C. Müller, "The automotive ontology: Managing knowledge inside the vehicle and sharing it between cars," in *Proceedings of the AutomotiveUI '11*, 2011, pp. 79–86.

[16] B. Yankson, "Autonomous Vehicle Security Through Privacy Integrated Context Ontology (PICO)," in *Proceedings of the SMC '20*, 2020, pp. 4372–4378.

[17] M. F. Arruda and R. F. Bulcão-Neto, "Toward a lightweight ontology for privacy protection in IoT," in *Proceedings of the SAC '19*, 2019, pp. 880–888.