Privacy management in connected environments

Karam Bou Chaaya

UNIV PAU & PAYS ADOUR

DOCTORAL THESIS

Privacy Management in Connected Environments

Karam BOU CHAAYA

Advisors:	Pr. Richard CHBEIR	Univ Pau & Pays Adour, France
	Dr. Philippe ARNOULD	Univ Pau & Pays Adour, France
Reviewers:	Pr. Esma AIMEUR	University of Montreal, Canada
	Pr. Allel HADJALI	ISAE-ENSMA, France
Examiners:	Pr. Djamal BENSLIMANE	Claude Bernard University, France
	Dr. Mahmoud BARHAMGI	Claude Bernard University, France
	Pr. Bechara AL BOUNA	Antonine University, Lebanon

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

09/09/2021

I wholeheartedly dedicate this work
to my parents Joumana & Elie
and to my siblings Carmen & Chris

Acknowledgements

"No one who achieves success does so without acknowledging the help of others. The wise and confident acknowledge this help with gratitude."

– Alfred North Whitehead

Foremost, I would like to express my deepest gratitude to my advisors Pr. Richard Chbeir and Dr. Philippe Arnould, and to my collaborators Dr. Mahmoud Barhamgi and Pr. Djamal Benslimane. I want to thank Richard for believing in me, helping me make the most of my abilities, and insisting on elevating my skills and overall approach towards research. His guidance, unwavering support, bright ideas, and immense knowledge made this work possible. My sincere heartfelt appreciation also goes to Philippe. He made things simple, easy, and comforting. This journey would not have been that enjoyable without him. He always motivated me and gave me great advice and constructive comments. My greatest gratitude also goes to Mahmoud for his valuable guidance, profound belief in my abilities, trust, and continuous encouragement during the completion of my Doctorate. My sincere thanks also go to Djamal for his assistance at every stage of my research project and his insightful comments and suggestions. It is a pleasure to have known them, and I look forward to possibly working with them again in the future.

I would also like to express my sincere gratitude to Pr. Esma Aimeur and Pr. Allel Hadjali for the time and effort they devoted to the examination of my dissertation. I also thank them for their valuable and helpful comments, suggestions, and feedback. I am also grateful to the rest of my thesis committee Pr. Djamal Benslimane, Dr. Mahmoud Barhamgi and Pr. Bechara Al Bouna for their presence, valuable questions, insightful comments, and encouragement. I am highly thankful to Bechara for his valuable help to get this opportunity and his encouragements to pursue my career in research.

I am thankful for being part of the LIUPPA laboratory. I specifically appreciate the seminars and scientific gatherings organized by its administration.

I owe my gratitude to the "Communauté d'Agglomération du Pays Basque". Nothing would have been possible without its financial support and interest in promoting science.

I would also like to thank all the technical and administrative staff of the "IUT de Bayonne et du Pays Basque" for the warm and convivial atmosphere and their sympathy. A special thanks to Pr. Philippe ANIORTE for his support and positive vibes. I acknowledge the moral and emotional support provided by Sabri Allani, Elio Mansour, Khouloud Salameh, Nathalie Charbel, Lara Kallab, Wissam Bejjani,

Rita Zougheib, Jocelyn Habib, Fawzi Khattar, Elias Abboud, Houssam Kanso, Reine Abou Sleiman, Rafik Abdallah, and Youssef Fawaz.

I am immensely thankful to my friends in Lebanon, France and abroad for their support and availability in the hard moments. Thank you Elie Abdel Massih, Ralph Abou Zeid, Ahmad Khalil, and Sabine Esber.

I owe my deepest gratitude to Remie Sarkis for sharing the ups and downs of this journey with me, encouraging me to work toward my dreams, picking me up when I struggled, and sharing my happiest moments.

I have to acknowledge the huge support provided by my family, especially Mr. Chaaya Bou Chaaya for his advice and empowering words. Your expertise gave me a lot of confidence in moving forward with the suggestions you outlined.

Lastly, and most importantly, I am eternally indebted to my beloved family, Joumana, Elie, Carmen, and Chris who has never left me alone despite the distance. Thank you Mom and Dad for your unconditional love and care, your consistent support and advice to reach my dreams. Without your drive and support, I might not be the person I am today. Thank you Sister and Brother for being by my side whenever I need. No amount of words will be enough to tell how grateful I am to you. Thank you for everything you have done for me.

Abstract

Recent years have witnessed rapid progress in enabling technologies for data sensing, communication and mining, paving the way for the phenomenal growth of smart connected environments (e.g., smart buildings, cities, factories). These environments are currently providing interesting and useful applications that help users in their everyday tasks (e.g. increasing comfort, reducing energy consumption). However, such applications require to collect, exchange, store, and process large amount of fine-granular data that is often privacy-sensitive for their users (e.g., location, energy-consumption), as its analysis allows data consumers to reveal sensitive information about them, such as their health conditions and preferences.

Consequently, involving users in the management of their privacy is nowadays receiving extensive attention. Nonetheless, various improvements are still required. For instance, how to raise user awareness of the privacy risks involved in their data sharing and/or imposed by their environments. Moreover, how to enable users to assess their situations and make optimal data utility-privacy decisions accordingly.

In this thesis, we focus on six main challenges: (i) representing diverse user contexts with a high semantic expressiveness power; (ii) performing a holistic (all-data-inclusive) context-based privacy risk reasoning; (iii) achieving user-centric privacy management; (iv) making optimal context-based privacy decisions; (v) coping with the inter-context data dependency; and (vi) delivering scalability and efficiency in order to assist the user in a variety of situations.

To address these challenges, we first present an ontology-based data model capable of representing various user contexts with high-level information coverage. Following that, we introduce a context-aware semantic reasoning approach for privacy risk inference that provides a dynamic/contextual overview of risks tailored to the user's expertise. Then, to enable optimal management of data utility-privacy trade-offs, we propose a user-centric multi-objective approach for context-aware privacy management that provides dynamic best data protection strategies to be implemented based on user situations and preferences. Finally, we propose a new stochastic gradient descent solution for privacy-preserving during protection transitions, which gives an additional layer of protection against data inference attacks.

The aforementioned contributions are regrouped in one global generic and extensible framework for context-aware privacy management.

Résumé

Ces dernières années, les technologies de détection, de communication et de gestion de données ont connu de progrès rapides, ouvrant la voie à la croissance phénoménale des environnements connectés intelligents (bâtiments, villes intelligentes). Ces environnements fournissent actuellement des applications intéressantes et utiles qui aident les utilisateurs dans leurs tâches quotidiennes (augmenter le confort, réduire la consommation d'énergie). Cependant, de telles applications nécessitent de collecter, échanger, stocker et traiter une grande quantité de données à granularité fine qui sont souvent sensibles pour leurs utilisateurs (localisation, consommation d'énergie), d'autant plus que leur analyse permet aux consommateurs de données de révéler des informations sensibles (état de santé, préférences des utilisateurs).

Par conséquent, la participation des utilisateurs dans la gestion de leur vie privée fait l'objet d'une grande attention. Néanmoins, diverses améliorations sont encore nécessaires. Par exemple, comment sensibiliser les utilisateurs des risques pour la vie privée liés au partage de leurs données et/ou imposés par leurs environnements. De même, comment permettre aux utilisateurs d'évaluer leur situation et de prendre des décisions optimales concernant l'utilité des données et le respect de la vie privée en conséquence.

Dans cette thèse, nous nous concentrons sur six défis principaux: (i) représenter des contextes diversifiés de l'utilisateur avec une haute puissance d'expressivité sémantique ; (ii) effectuer un raisonnement holistique (toutes données incluses) et contextuel sur les risques en matières de vie privée ; (iii) parvenir à une gestion de vie privée centrée sur l'utilisateur ; (iv) prendre des décisions contextuelles optimales liées à la protection de la vie privée ; (v) gérer la dépendance des données inter-contextuelles ; et (vi) fournir une solution évolutive et efficace afin d'assister l'utilisateur dans diverses situations.

Pour ce faire, nous présentons d'abord un modèle de données basé sur une ontologie capable de représenter divers contextes utilisateur avec une couverture d'informations de haut niveau. Ensuite, nous introduisons une approche de raisonnement sémantique qui fournit un aperçu dynamique/contextuel des risques en matière de vie privée, adapté à l'expertise de l'utilisateur. Ensuite, pour permettre une gestion contextuelle optimale des compromis entre utilité des données et protection de la vie privée, nous proposons une approche multi-objectifs centrée sur l'utilisateur qui fournit dynamiquement les meilleures stratégies de protection de données à mettre en œuvre en fonction des situations et préférences des utilisateurs. Enfin, nous proposons une nouvelle solution de descente de gradient stochastique pour assurer une transition intélligente du niveau de protection des données. Cette solution offre ainsi une couche supplémentaire de protection contre les attaques par inférence de données.

Les contributions susmentionnées sont regroupées dans un framework global générique et extensible pour la gestion contextuelle de la vie privée.

Le manuscrit est organisé comme suit :

Chapitre 1

Introduction

Dans ce chapitre, nous introduisons les facteurs technologiques qui ont contribué à la prolifération des environnements connectés pendant nos jours. Ensuite, nous nous concentrons sur la vie privée des utilisateurs dans le contexte de ces environnements (menaces et défis relatifs à la vie privée, lois et standards mondiales de vie privée). Par la suite, nous présentons le contexte et les objectifs de cette thèse. Nous étudions un scénario qui illustre la motivation de ce travail et les défis émergents. Nous examinons les approches existantes dans la littérature sur la protection contextuelle de la vie privée dans les environnements intelligents. Ensuite, nous présentons notre framework proposé pour la gestion contextuelle de la vie privée dans les environnements connectés (CaPMan), dans lequel chaque module répond à un ensemble de besoins et de défis:

- Premier Module: Gestion de l'information. Ce module est responsable de la gestion des informations contextuelles (acquisition et modélisation des informations) et des préférences de l'utilisateur. Nous nous focalisons dans cette thèse sur la modélisation du contexte et nous proposons un modèle ontologique, notée uCSN, permettant de représenter d'une manière expressive différentes situations de l'utilisateur.
- Deuxième Module: Inférence de risques liés à la vie privée. Ce module comprend un raisonneur de risque, noté CaSPI, en charge de détecter d'une manière dynamique les risques impliqués pour l'utilisateur en fonction de l'évolution de sa situation.
- Troisième Module: Gestion de la vie privée. Ce module est chargé d'assister l'utilisateur dans la gestion de sa vie privée et la protection de ses données avant qu'elles ne soient communiquées aux consommateurs de données. Pour ce faire, ce module comprend un gestionnaire de risque, noté *δ-Risk*, responsable d'analyser les risques détectés et de fournir à l'utilisateur des stratégies de protection de vie privée optimisées à appliquer en fonction de sa situation et préférences. Pour la protection des données, nous nous concentrons dans cette thèse sur les transitions de protection et nous proposons une nouvelle approche de descente de gradient stochastique, notée P-SGD, qui permet de surmonter les vulnérabilités aux attaques par inférence de données.

Finalement, nous répertorions les publications liées à ce rapport avant d'introduire les chapitres suivants.

Chapitre 2

Modélisation du contexte dans des environnements connectés

Dans ce chapitre, nous décrivons un modèle de données basé sur une ontologie. Nous présentons une étude comparative des travaux existants sur la modélisation de l'utilisateur (profile, activité), de l'environnement et du contexte (utilisateur, environnement et d'autres dimensions). Ensuite, nous introduisons notre ontologie pour la modélisation du contexte utilisateur dans les réseaux de capteurs (uCSN), dans laquelle nous enrichissons la représentation du contexte pour prendre en compte divers types de : (i) informations utilisateur/environnement (informations scalaires, multimédias) ; (ii) sources de données (capteur, document) ; (iii) incertitudes (incertitudes liées à l'utilisateur, à l'environnement) ; et (iv) environnements (environnements connectés/non connectés, environnements avec des systèmes et appareils statiques/mobiles). Pour ce faire, nous définissons de nouveaux concepts et propriétés, et nous importons d'autres à partir des ontologies bien connues, à savoir DPV [1], SOSA/SSN [2] et W3C Uncertainty Ontology [3], sans compromettre la possibilité de réutilisation du modèle de données dans différents domaines d'application. Enfin, nous évaluons la performance, la clarté, la cohérence et la précision de l'ontologie proposée.

Chapitre 3

Inférence de risques liés à la vie privée

Dans ce chapitre, nous décrivons le raisonneur de risque que l'on utilise pour déduire les risques en matière de vie privée impliqués dans le contexte de l'utilisateur. Nous passons en revue les travaux existants sur l'inférence des risques avant de détailler l'approche proposée (CaSPI [4]) qui exerce un raisonnement sémantique et contextuel pour l'inférence dynamique des risques. Nous relevons les défis liés à (i) l'augmentation de l'expressivité dans les définitions des risques ; (ii) la mise en œuvre d'un raisonnement holistique prenant en compte différents types de combinaison données/informations contextuelles ; (iii) faire face à la dynamique et à la dépendance contextuelle des risques liés à la vie privée ; (iv) gérer et s'adapter à l'expertise des utilisateurs ; et (v) assurer l'évolutivité et l'efficacité de la solution. Nous validons notre proposition en développant un prototype et nous illustrons son fonctionnement en back-end et front-end. Enfin, nous évaluons ses performances en considérant différents scénarios.

Chapitre 4

Gestion de risques liés à la vie privée

Dans ce chapitre, nous décrivons le gestionnaire de risques qui évalue les valeurs des risques inférés, puis calcule les meilleures stratégies de protection des données adaptées à la situation et aux préférences de l'utilisateur. Nous présentons notre approche multi-objectifs et contextuelle proposée pour la gestion de la vie privée (δ-*Risk* [5]). Nous détaillons le processus suivi depuis les entrées (risques impliqués, préférences de l'utilisateur) jusqu'aux meilleures stratégies fournies à la sortie à l'utilisateur. Nous validons notre proposition en développant un prototype et illustrons son fonctionnement en back-end et en front-end. Enfin, nous évaluons ses performances en considérant différents scénarios et étudions formellement son efficacité dans l'identification de stratégie.

Chapitre 5

Préservation de la vie privée pendant les transitions de protection

Dans ce chapitre, nous nous concentrons à surmonter la vulnérabilité du système aux attaques par inférence de données pendant les transitions de protection des données (par exemple, lorsqu'un changement de stratégie se produit). Nous soulignons les cas qui contribuent aux fuites temporelles de la confidentialité des données lors des transitions de protection. Ensuite, nous introduisons notre solution de descente de gradient stochastique proposée pour la préservation de la vie privée pendant les transitions du niveau de protection des données (P-SGD [6]). La solution proposée est connectée au composant de protection des données du framework, et déclenchée lors des phases de descente de protection pour fournir une couche de protection supplémentaire contre les attaques par inférence de données. Nous détaillons le processus suivi par P-SGD et illustrons son fonctionnement en exécutant le prototype développé. Enfin, nous présentons l'expérimentation et les résultats.

Chapitre 6

Conclusion et Travaux Futurs

Ce chapitre conclut le rapport en récapitulant tous les chapitres susmentionnés et en détaillant les prochaines étapes, extensions futures, et de nouvelles orientations possibles pour la suite de ce travail de recherche.

Contents

List of Figures

List of Tables

List of Abbreviations

ACM	**A**ssociation for **C**omputing **M**achinery
API	**A**pplication **P**rogramming **I**nterface
AWI	**A**pproved **W**ork **I**tem
CalOPPA	**C**alifornia **O**nline **P**rivacy **P**rotection **A**ct
CaPMan	**C**ontext-**a**ware **P**rivacy **M**anagement
CAS	**C**ontext-**A**ware **S**ystems
CASPaaS	**C**ontext-**A**ware **S**ecurity and **P**rivacy as a **S**ervice
CaSPI	**C**ontext-**a**ware **S**emantic reasoning for **P**rivacy risk **I**nference
CCPA	**C**alifornia **C**onsumer **P**rivacy **A**ct
CC/PP	**C**omposite **C**apabilities/**P**reference **P**rofile
CCTV	**C**losed-**C**ircuit **T**elevision
CNIL	**N**ational **C**ommission on **I**nformatics and **L**iberty
CoBrA-Ont	**C**ontext **B**roker **A**rchitecture **O**ntology
CONCON	**C**ontext **O**ntology
COPD	**C**hronic **O**bstructive **P**ulmonary **D**isease
CPS	**C**yber-**P**hysical **S**ystems
CRBAC	**C**ontext-sensitive **R**ole-**b**ased **A**ccess **C**ontrol
DL	**D**escription **L**ogic
DPV	**D**ata **P**rivacy **V**ocabulary
ECA	**E**vent-**C**ondition-**A**ction
EU	**E**uropean **U**nion
FOAF	**F**riend **O**f **A** **F**riend
GDPR	**G**eneral **D**ata **P**rotection **R**egulation
GeoSNs	**G**eo-**S**ocial **N**etworks
GJE	**G**auss-**J**ordan **E**limination
GPS	**G**lobal **P**ositioning **S**ystem
HSSN	**H**ybrid **S**emantic **S**ensor **N**etwork
I-AM	**I**nform-**A**lert-**M**itigate
ICT	**I**nformation and **C**ommunication **T**echnologies
ID	**ID**entifier
IEC	**I**nternational **E**lectrotechnical **C**ommission
IoT	**I**nternet of **T**hings
IoT-O	**I**nternet of **T**hings **O**ntology
ISMS	**I**nformation **S**ecurity **M**anagement **S**ystem
ISO	**I**nternational **O**rganization for **S**tandardization

LGPD	**G**eneral **D**ata **P**rotection **L**aw
MOV	Quicktime **M**ovie
mIO!	Context **O**ntology for **M**obile Environments
NIV	**N**on-**I**nvasive **V**entilation
OSN	**O**nline **S**ocial **N**etworks
OWL	**W**eb **O**ntology **L**anguage
PbD	**P**rivacy **b**y **D**esign
PDP	**P**ersonal **D**ata **P**rotection
PII	**P**ersonally **I**dentifiable **I**nformation
PIMS	**P**rivacy **I**nformation **M**anagement **S**ystem
PIPEDA	**P**ersonal **I**nformation **P**rotection and **E**lectronic **D**ocuments **A**ct
PiVOn	**P**ervasive **i**nformation **V**isualization **O**ntology
PKI	**P**ublic **K**ey **I**nfrastructure
PNG	**P**ortable **N**etwork **G**raphics
POPI	**P**rotection **o**f **P**ersonal **I**nformation Act
P-SGD	**P**rivacy-preserving **S**tochastic **G**radient **D**escent
RDF	**R**esource **D**escription **F**ramework
RuleML	**R**ule **M**arkup **L**anguage
SDN	**S**oftware **D**efined **N**etworking
SGD	**S**tochastic **G**radient **D**escent
SMC	**S**ecure **M**ultiparty **C**omputation
SOSA	**S**ensors **O**bservations **S**amples **A**ctuators
SOUPA	**S**tandard **O**ntology for **U**biquitous and **P**ervasive **A**pplications
SPARQL	**S**imple **P**rotocol **A**nd **R**DF **Q**uery **L**anguage
SPI	**S**ensitive **P**ersonal **I**nformation
SSN	**S**emantic **S**ensor **N**etwork
SWRL	**S**emantic **W**eb **R**ule **L**anguage
uCSN	**u**ser-**C**ontext modeling in **S**ensor **N**etworks
UPO	**U**ser **P**rofile **M**odeling
UPOS	**U**ser **P**rofile **O**ntology with **S**ituation-dependent preferences
UP-PwD	**U**ser **P**rofile **P**eople **w**ith **D**ementia
URI	**U**niform **R**esource **I**dentifier
US	**U**nited **S**tates
W3C	**W**orld **W**ide **W**eb **C**onsortium
XML	e**X**tensible **M**arkup **L**anguage

Chapter 1

Introduction

"He's not our hero. He's a silent guardian, a watchful protector. A dark knight."

– Jonathan Nolan, The Dark Knight

1.1 Connected Environments

Recent years have witnessed great strides in the fields of Ubiquitous Computing (e.g., Internet of Things), Big Data, and Machine Learning that have led to the rapid growth of smart connected environments. These environments are defined as infrastructures that host Cyber-Physical Systems (CPS), such as sensor networks, interconnected using various communication technologies (e.g., Bluetooth, 6LoWPAN). Connected systems are capable of collecting valuable data that can be later mined and processed to provide advanced services for both environments and users. Current CPS-based applications are impacting numerous application domains including smart healthcare (e.g. patient and elderly monitoring), smart buildings/homes (e.g., increasing occupants' comfort, optimizing energy consumption), smart cities (e.g., traffic management, safety and disaster prevention, air quality monitoring), and so forth. Figure 1.1 illustrates examples of connected environments.

The successful proliferation of connected environments has been driven by various technological factors. From the data sensing perspective, recent advances in Sensing Technologies have enabled the development of low-cost, low-power, multifunctional sensor nodes [7], such as cameras, microphones, GPS, environmental sensors (e.g., sensors to measure temperature, humidity), and medical sensors (e.g., sensors to measure heart-rate, blood pressure). These sensors can be embedded on various devices (e.g., mobile phones, smartwatches) or/and deployed in different environments (e.g., cities, buildings). The advanced capabilities of the sensing objects have allowed the collection and transmission of numerous heterogeneous data about users and environments (i.e., data with different types and formats). From the data mining and processing perspective, recent advances in Information and Communication Technologies (ICT), Big Data, and Data Mining techniques have made it easier to deal with challenges related to managing big data volumes, heterogeneous

data, continuous data streams, data pre-processing, and knowledge extraction from data. This has paved the way for better monitoring and prediction of environmental situations, events, and user behaviors and activities, which has improved the capabilities of current applications to make knowledge-driven decisions.

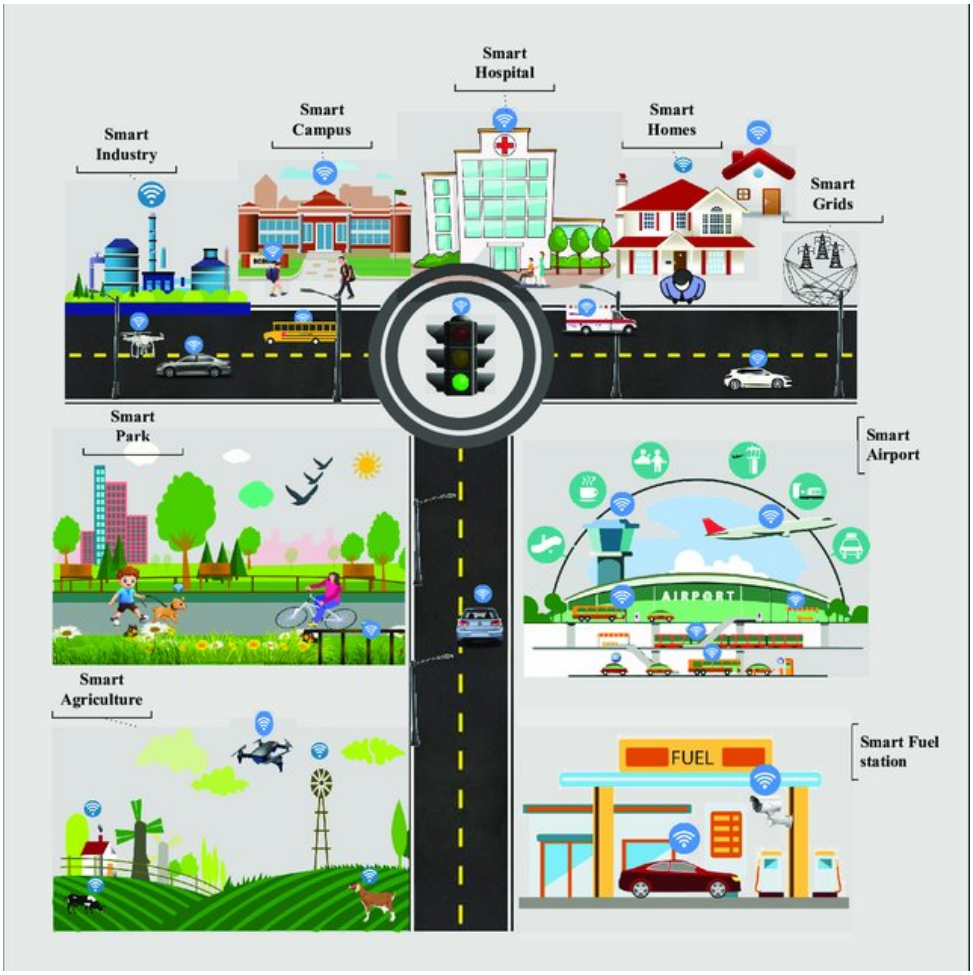

FIGURE 1.1: Examples of Connected Environments

1.2 Privacy in Connected Environments

With the wide and rapid evolution of smart connected environments, the number of applications involving users and the level of interaction with the user are both bound to increase. These applications collect and process large amount of data in order to provide better services that fulfill personal preferences and improve the user experience. However, collected data can be also leveraged to draw detailed profiles of users, including their behaviors, preferences, and activities, which raises significant challenges with regard to the protection of users' privacy.

So far, privacy has been difficult to define and formalize due to its changing perspective over time and across cultures. Privacy was first defined in 1890 as "*the*

right to be left alone" [8]. Then, in the 1960s, it was the rise of electronic data processing that brought into being the notion of information privacy (or data privacy). Westin [9] defined privacy as "*the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others*", which mainly emphasized the control of the data subjects over their data. Following that, Ziegeldorf et al. [10] argued in the 2010s that Westin's definition was too general for the IoT area, and consequently defined IoT privacy as the threefold guarantee including "*(i) awareness of privacy risks imposed by smart things and services surrounding the data subject; (ii) individual control over the collection and processing of personal information by the surrounding smart things; and (iii) awareness and control of subsequent use and dissemination of personal information by those entities to any entity outside the subjects personal control sphere*". After reviewing existing privacy definitions, the scope of this work is best summarized by the privacy perspective of Ziegeldorf [10], which addresses the self-determined management of personal information in the IoT era. This perspective is also compliant with the privacy needs outlined in current privacy laws and standards.

In the following sub-sections, we present the definition of personal information. Then, we outline the most recent threats and challenges to user privacy in connected environments. After that, we provide an overview of the current data protection laws from around the world, followed by the recently released privacy and data protection standards.

1.2.1 Personal Information

The General Data Protection Regulation (GDPR) [11] defined personal information (or personal data) as "*any information relating to an identified or identifiable natural person (data subject)*". The National Institute of Standards and Technology [12] defined personal information as "*any information about an individual maintained by an agency, including (1) any information that can be used to distinguish or trace an individual's identity [...]; and (2) any other information that is linked or linkable to an individual, such as medical, educational, financial, and employment information*". The California Consumer Privacy Act (CCPA) [13] defined personal information as "*any information that identifies, relates to, describes, is reasonably capable of being associated with, or could reasonably be linked, directly or indirectly, with a particular consumer or household*". In light of this, we distinguish between identifiable and sensitive information by introducing two categories of personal information:

- ***Personally Identifiable Information (PII)***: Any information that can be used to distinguish or trace the data subject's identity. For example, the data subject's name, home address, email, phone number, biometrics, pictures, social security number, or domain-related ID like patient ID in e-health.

- ***Sensitive Personal Information (SPI)***: Any other sensitive information that alone does not identify the data subject but it is linked or linkable to him/her. SPI communicates information that is private or likely to harm the data subject if misused or sold to third parties. For example, age, gender, marital status, political/religion beliefs, locations, activities, habits, interests, or other domain-specific information like medical, financial, or social information.

1.2.2 Privacy Threats & Challenges

Advances in sensing technologies and data mining techniques pose several threats and challenges to the privacy of users in connected environments [10], [14], [15]. Recent threats vary from identification, localization and tracking, and profiling to privacy-violating interaction and presentation, device life-cycle transitions, inventory attack, and linkage. We discuss in the following each of the seven threats, before concluding with the currently faced challenges.

(1) *Identification*: consists of linking the identifier (e.g., name, pseudonym) associated with collected sensor data to a specific user. This raises serious privacy concerns for the user since collected data is often privacy-sensitive (e.g., location of individuals, patients' vital signs), implying that when processed, it can reveal a plethora of SPI about the user. As well, the identification threat can also enable other threats like profiling and tracking of users.

(2) *Localization and tracking*: the user's location can be monitored and tracked in time and space through different means, e.g., GPS sensors deployed on user devices (e.g., mobile phone, smart watch), internet traffic, camera recordings. Besides the uneasy feeling of being watched, mining and processing the locations and trajectories of the user can lead to deduce sensitive information about her, such as her performed/daily activities, habits, and health conditions, therefore subjecting her to several privacy breaches.

(3) *Profiling*: refers to the threat of collecting and correlating profiles of data in order to analyze or predict aspects concerning the user, including user's economic situation, health, personal preferences, interests, reliability, behaviour, and movements. Profiling methods are mostly used in e-commerce for personalization (e.g., sending targeted advertisements or newsletters). Consequently, many privacy violations occur when user profiles are collected, processed, correlated, or even sold to third parties interested in exploiting it (e.g., marketing companies).

(4) *Privacy-violating interaction and presentation*: sensor nodes and multi-sensor devices may collect and transmit people's private information through public means in order to present the information (e.g. speakers, video screens) when people interact with the devices (e.g. moving, speaking, touching). This could therefore entail the leakage of private/sensitive information from what is presented to an unwanted audience.

(5) *Device life-cycle transitions*: during their life-cycle, devices (i.e., multi-sensor devices like mobile phones, tablets) can be used, then sold or destroyed. Even though it must destroy all data, some devices often store large amounts of historical data over their life-time that could be sensitive (e.g., personal photos, videos). This entails privacy issues for the user if data was not deleted prior to changing device ownership.

(6) *Inventory attack*: denotes the threat of targeting an object (i.e., sensor or device) by sending various query requests to it and analyzing the related responses. An adversary may use this type of attacks to compile an inventory list of other devices and/or appliances in the environment (e.g., medical devices or smart alarm systems at the user's home). This list can be privacy-sensitive for the user as it could be used for targeted break-ins at private homes/offices, or even may lead to reveal SPI about the user such as her health conditions (it is enough to infer the use of medical devices at home).

(7) *Linkage*: user data could be shared between service providers, or even sold to interested third parties, with or without her knowledge. This raises serious privacy concerns for the user as she may not be aware of how her data is being used, by whom, with what data/information it is linked, or what SPI may be disclosed when combining and processing data bits and pieces.

Privacy remains therefore a major challenge to address in the field of connected environments. This was also supported by the European Union (EU) commission, which identified security and privacy as major IoT research challenges [16]. In particular, the currently faced challenges for IoT privacy are more linked to the goals of reducing the aforementioned threats. These include: how to enable users (data subjects) to understand their privacy situations? how to empower them to control their data sharing and protection appropriately in a way to meet their privacy requirements and business interests? and how to ensure data privacy protection throughout the entire data life-cycle phases, i.e. during data collection, transmission, aggregation, storage, mining, and processing phases?

1.2.3 Worldwide Privacy Legislation

Privacy is considered as a fundamental human right in the United Nations Universal Declaration of Human Rights and Article 8 of the European Convention on Human Rights of 1950 [17]. This right became explicit by the emergence of numerous privacy laws and regulations around the world. Nowadays, there is no general information privacy legislation that covers all areas [18]. Figure 1.2[1] shows the distribution of existing privacy laws among states, countries, and regions. We present in the following a brief overview of these laws.

[1]Source: `https://termly.io/resources/infographics/privacy-laws-around-the-world/`

Currently, more than 130 countries enacted national/regional privacy laws, which define different technical and organizational requirements for the storage and processing of personal data in information systems [19]. As a successor of the Directive 95/46/EC, the European Union adopted the European General Data Protection Regulation (GDPR) [11], which came into force in 2018. Its key changes in terms of the principles, compared with the Directive 95/46/EC, include six aspects: (1) *consent*, the data subject's consent should be graspable, distinguishable and easy to be withdrawn; (2) *breach notification*, the GDPR makes the breach notification mandatory. The notification should be sent within 72 hours after being aware of the breach; (3) *right to access*, the GDPR grants data subjects the right to be informed about data processing and to receive a copy of the handled personal data; (4) *right to be forgotten*, the data is required to be erased when the personal data are no longer necessary in relation to the purposes or the consent is withdrawn; (5) *right to data portability*, the data subject has the right to receive his uploaded data in a machine-readable format and transmit it to other data controllers; and (6) *privacy by design*, the GDPR integrates the privacy by design as a legal requirement, where the controller must implement appropriate technical and organizational measures in order to meet the GDPR requirements and protect the data subjects' rights (the Privacy by Design standard is further detailed in the following section).

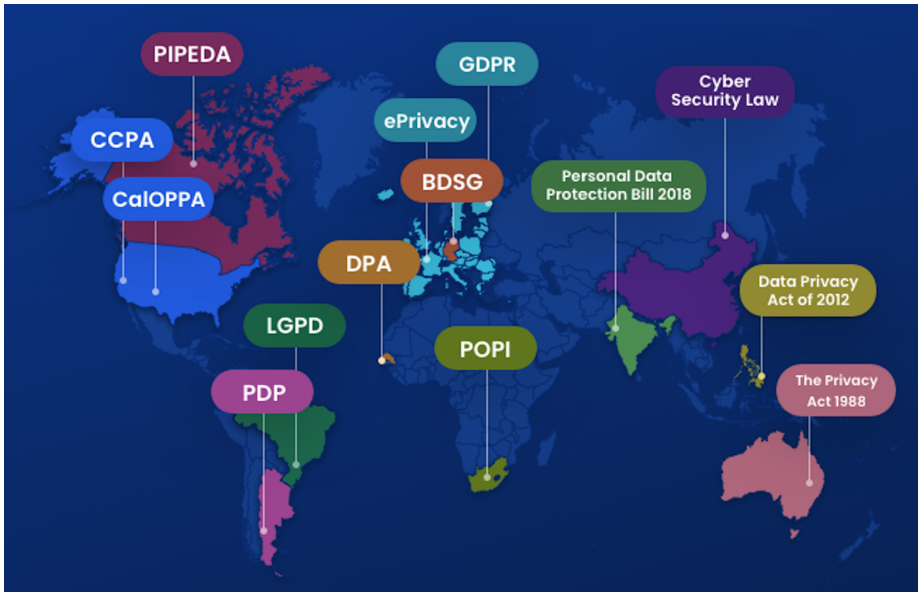

FIGURE 1.2: Privacy Laws Around the World

For the United States (US), some states have its own laws. California adopts the California Consumer Privacy Act (CCPA) and California Online Privacy Protection Act (CalOPPA) laws. The CCPA law came into force in 2020, and became the first GDPR-like law in the country. It boasts three guiding principles: transparency, accountability, and control. It grants data subject rights to access, portability and deletion. The CalOPPA law is the first to require websites to post privacy policies detailing data collection and use, however, it is only applicable to businesses and

online operations with data subjects in California. Nevada adopts the Senate Bill 220 law, which became effective in 2019, and seems very similar to the CCPA but has some significant differences, such as only giving data subjects the right to opt out of having their data sold. There are several other US states in the process of passing comprehensive data protection laws (e.g., Consumer Data Protection Act for Virginia, which will be effective in 2023). Other countries have also adopted their own privacy laws, such as PIPEDA for Canada, LGPD for Brazil, PDP for Argentina, POPI for South Africa, DPA for Senegal, Personal Data Protection Bill 2019 for India, Cyber Security Law for China, Data Privacy Act of 2012 for Philippines, and Privacy Act 1988 for Australia.

1.2.4 International Privacy Standards

The laws principles are usually described with very general and broad terms [20] that makes it hard for companies and organizations to properly integrate them in the system design due to the variety and diversity of existing information technologies. Consequently, several privacy standards were introduced to bridge the gap between legal frameworks and technologies by providing a set of guidelines that translate legal principles into more engineer-friendly principles in order to facilitate the design of privacy capabilities in systems and applications. Figure 1.3 illustrates an overview of the existing privacy standards with their mappings. We provide next a brief description of each standard and highlight the respective principles.

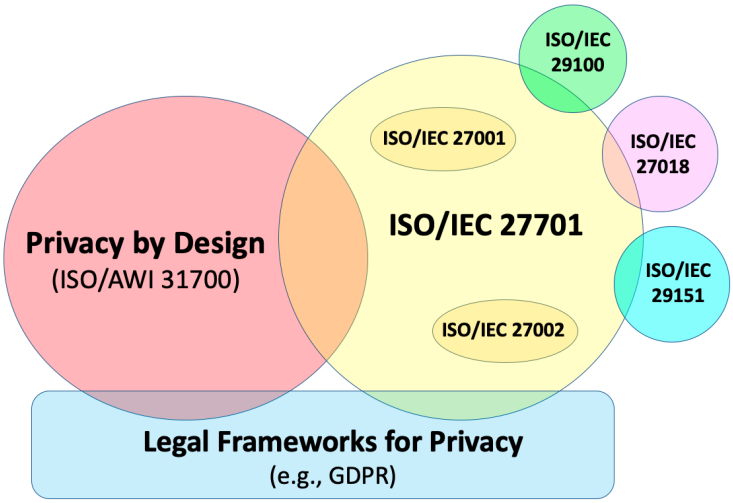

FIGURE 1.3: Privacy Standards

Privacy by Design (PbD) has brought a new vision for privacy protection to cope with the increasing complexity and interconnectedness of information technologies. Instead of reactively addressing privacy breaches after-the-fact, PbD approaches privacy proactively and tends to prevent privacy-invasive events before they happen by making privacy the default setting [21]. In 2010, PbD has been unanimously adopted as an international privacy standard in the 32nd International Conference

of Data Protection and Privacy [22]. Nowadays, PbD is incorporated as a legal requirement in the General Data Protection Regulation (GDPR) [11], and globally recognized as an ISO/AWI standard (ISO/AWI 31700 - under development) [23]. PbD identifies seven foundational principles that should be followed when developing privacy sensitive solutions:

(1) *Proactive not reactive; Preventative not remedial*. The solution should include proactive measures to anticipate and prevent privacy violations, i.e., to prevent privacy risks from occurring.

(2) *Privacy as the Default Setting*. The solution must deliver the maximum degree of privacy and data protection by default, without requiring user intervention.

(3) *Privacy Embedded into Design*. Privacy must be incorporated as an essential component of the solution's core functionality.

(4) *Full Functionality: Positive-Sum, not Zero-Sum*. The solution seeks to accommodate all interests and objectives in a positive-sum (i.e., win-win manner).

(5) *End-to-End Security*. The solution should ensure data protection during the entire life-cycle of data.

(6) *Visibility and Transparency - Keep it Open*. The solution must provide accountability, openness and compliance, which, in turn, improve user satisfaction and trust.

(7) *Respect for User Privacy - Keep it User-Centric*. The solution should empower data subjects (users) to play an active role in the control and management of their data. This can be achieved by ensuring that appropriate notice is given, and by supporting other user-friendly options, such as considering user preferences, delivering human-machine interfaces adaptable to users, and enabling users to make informed privacy decisions.

The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) drew up a new reference standard in 2019, ISO/IEC 27701:2019 [24] for privacy information management. The design goal is to provide guidance for establishing, maintaining, and continually improving a Privacy Information Management System (PIMS). This standard can be used by: (i) data controllers, entities (e.g., person, organization) that, alone or jointly with others, determines the purposes and means of the processing of personal data [11]; and data processors, entities (e.g., person, organization) that processes personal data on behalf of the controller [11]. It has been developed as an extension to ISO/IEC 27001 and ISO/IEC 27002, which respectively provide guidelines for Information Security Management System (ISMS) implementation, and information security controls. The ISO/IEC 27701 also includes mapping to other existing privacy standards and legal frameworks, such as ISO/IEC 29100, ISO/IEC 27018, ISO/IEC 29151, Privacy

by Design, and the GDPR regulation. The ISO/IEC 29100:2011 [25] provides a privacy framework that considers the following privacy safeguarding requirements to protect personal information: (1) consent and choice, (2) purpose legitimacy and specification, (3) Collection limitation, (4) Data minimization, (5) use, retention and disclosure limitation, (6) accuracy and quality, (7) openness, transparency and notice, (8) individual participation and access, (9) accountability, (10) information security, and (11) privacy compliance. The ISO/IEC 27018:2019 and ISO/IEC 29151:2017 standards are both based on ISO/IEC 27002, where the first provides guidance for the protection of personal information in public clouds acting as processors, and the second defines guidelines for personally identifiable information protection.

Therefore, the ISO/IEC 27701 is perceived as a global standard, as stressed by the National Commission on Informatics and Liberty[2] (CNIL-France), that provides several measures/requirements for the processing of personal data/information [24]. These measures can be classified into controller-specific and processor-specific measures. Controller-specific measures include: (i) *privacy notices*, controllers should provide privacy policies to data subjects containing specific information regarding the collection, use, and processing of their data; (ii) *processor agreement requirements*, including data protection, breach notifications, and limiting data processing to the agreed purposes; (iii) *data subjects' rights*, including rights to access, correct, and erase their data, and to restrict the processing of their data among others, (iv) *privacy by design and by default*, controllers must adopt measures that operationalize the principles of privacy by design and by default (e.g., minimization, data de-identification and deletion, data retention). Processor-specific measures include: (1) *processing limitations* to the purpose specified in the contract; (2) *data subjects' rights*; (3) *transfers and disclosures*, processors must inform data subjects in advance of data transfers between jurisdictions or any intended changes thereof; and (4) *subcontractors*, requires processors to only engage subcontractors for processing data subjects' data pursuant to the terms of the contract.

1.3 Thesis Context

Sharing data in exchange for goods and services presents an opportunity for users to improve their quality of life, however, it also exposes them to many privacy risks. In fact, processing and analyzing collected sensor data (e.g., location of individuals, patient's vital signs), which are spatio-temporal in nature [26], can lead to disclosing a wide variety of privacy-sensitive information about users [27], [28], such as health conditions, performed or daily activities, habits, preferences, and so on. This disclosure may be intentional if users are aware of it and have entered into agreements with relevant service providers. However, it can be harmful if the data/information

[2]Source: `https://www.cnil.fr/en/iso-27701-international-standard-addressing-personal-data-protection`

of users is misused by providers, sold to interested third parties without user consent, or stolen by cybercriminals as providers are often victims of cyber-attacks that lead to data breaches.

Consequently, involving users in the control and protection of their privacy is currently receiving extensive attention from both legal and technical perspectives [11], [13], [21], [24], [25]. Nonetheless, existing legal frameworks for data protection (e.g., GDPR [11]) might not necessarily deter data consumers from abusing, intentionally or unintentionally, the data of users. The Facebook-Cambridge Analytica [29] and Exactis [30] scandals are only few examples of a long series of data breach scandals that happened despite the existence of appropriate data protection laws. In addition, privacy laws vary among countries, some providing more protection than others (e.g., GDPR [11] for the European Union, CCPA [13] for the state of California). This increases the difficulty and complexity of managing and preserving the privacy of users, especially when users, service providers, and third parties are located in different countries governed by different data protection laws. Therefore, all these constraints emphasize the need for user-centric technical solutions that maintain the same level of data privacy protection in all countries.

Current approaches of user-centric privacy preserving [31]–[34] mainly rely on preference specification and policy enforcement, where users specify their privacy preferences and accept policies that enforce these preferences. However, they all share two main limitations:

(1) *lack of user awareness*. The user may not be completely aware of the direct and indirect privacy risks involved with the exchange of her data with providers to correctly specify her preferences in the first place. She may simply not know what sensitive information might be revealed from her data when data pieces are analyzed in isolation or combined with each other or/and with other side information acquired from external data sources (e.g., social networks).

(2) *lack of context-based privacy decision making*. The data sharing or protection decisions are often made/accepted by the user in a static way. This means that they remain unchanged regardless of context changes. However, the sensitivity of data may vary from a context to another [28], [35], i.e., new privacy risks may emerge as others may lose their significance. This makes static decisions over-protective in some contexts, causing unnecessary loss of data quality which may downgrade the accuracy of associated services; or under-protective, leading consequently to privacy violations. Therefore, the user must be able to make dynamic adjustments to her privacy decisions to cope with the dynamicity of her context.

1.3.1 Thesis Objectives

The objectives of this thesis are to design suitable solutions that overcome the aforementioned two limitations, and to provide a complete context-aware privacy framework that meets the guidelines of current privacy standards (i.e., Privacy by Design [21] and ISO/IEC 27701 [24]). Specifically, the framework needs to cope with:

- Raising user awareness of the privacy risks associated with their data sharing and/or imposed by their surrounding environments, by providing them with a dynamic/contextual overview of risks tailored to their level of expertise.
- Assisting users in optimizing their data utility-privacy decisions according to their situations, needs and preferences, by providing them with the best data protection strategies that could be implemented in their situations.
- Ensuring appropriate protection of the data collected, according to user decisions, before being transmitted to data consumers.

1.3.2 Motivating Scenario

To illustrate the motivations behind the objectives of this thesis, we investigate a real-life scenario of a user, Alice, who shares data with service providers. This scenario highlights some of the privacy risks that can arise from sharing Alice's data, and underlines the need for dynamic/contextual adaptations of her data protection decisions. Figure 1.4 illustrates the proposed scenario.

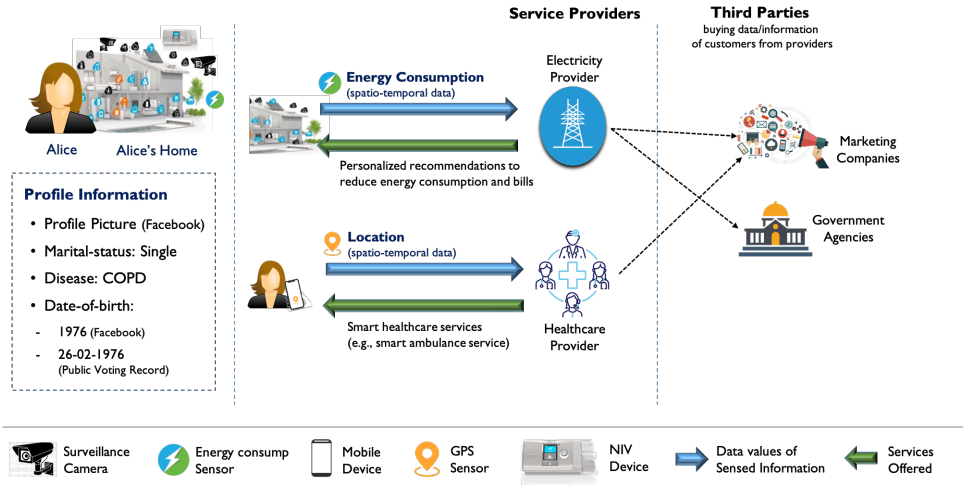

FIGURE 1.4: Motivating Scenario

Assume that Alice is a COPD (Chronic Obstructive Pulmonary Disease) patient. She pursues her medical treatment remotely using a NIV (Non-Invasive Ventilation) device deployed at home. Consider that Alice shares fine-grained data with the following service providers:

- ***Electricity provider***: Alice shares the energy consumption readings of her home through deployed smart energy meters. In exchange, Alice receives personalized recommendations to reduce her energy consumption and bills.
- ***Healthcare provider***: Alice shares her location data through a mobile application to benefit from an emergency care system that offers healthcare services, such as the smart ambulance service that she would use in case of respiratory distress.

The trust relationship between Alice and the providers is not static. It varies due to many factors such as the sensitivity of her situation, or the third parties with whom the provider communicates her data. Assume that both providers have signed contracts with marketing companies and government agencies interested in exploiting the data of their customers (e.g., Alice) for different purposes. For example, marketing companies could be interested in exploiting the energy consumption data to analyze the lifestyle of customers and send them targeted advertisements (e.g., advertisements about appliances that customers own or do not own). Government agencies could be interested in identifying customers involved in wrongdoing (e.g., fraud, crimes).

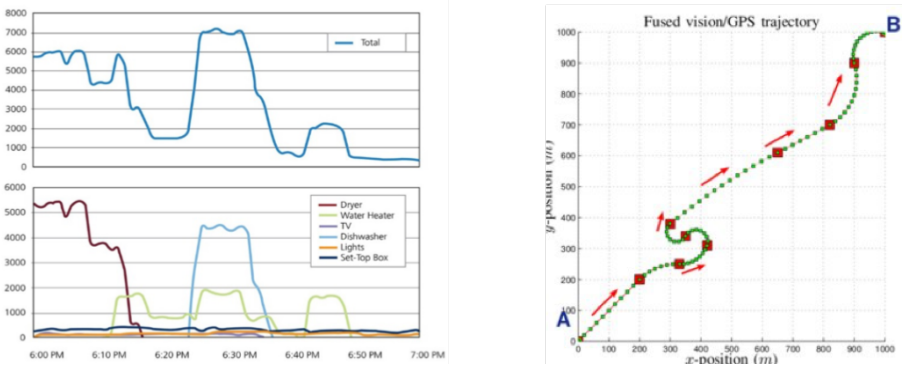

FIGURE 1.5: Energy consumption signature / Location data pattern

Even though Alice is notified, through agreed policies, of consumers who have access to her data, she may not necessarily be aware of the privacy risks involved with this sharing. These risks can be of two types: mono-source and multi-source risks. Mono-source risks arise from sharing data with a single data consumer. For instance, analyzing the energy consumption data (see the signature in Figure 1.5) can entail various mono-source risks for Alice, such as the risks of disclosing her presence/absence hours at home, waking/sleeping cycles, some of her habits and activities (e.g., cooking, TV watching, sports activity using a treadmill) [36]. Moreover, existing works (e.g., [37]) show that consumption signatures can be mined to identify the use of specific appliances (e.g., medical devices). This would reveal the health condition of Alice if the use of her NIV machine was identified. The analysis of location data can also entail significant mono-source risks for Alice such as the

risks of disclosing her habits, behaviors and health conditions by analyzing her trajectory patterns (cf. Figure 1.5). For example, if Alice is located twice per week in a pulmonary rehabilitation center for COPD patients, then she is very likely to be a COPD patient. Multi-source risks are more complex risks that arise when customer data are communicated between consumers (cf. Figure 1.4). For example, assume that Alice has unlawfully certified that she is living alone to be eligible for a welfare program when submitting her application. A marketing company having access to both location and consumption data can infer this fraud (it is enough to identify the use of particular devices, such as microwave and TV, while Alice is outside her home).

All of this highlights the need to inform Alice, in an appropriate manner, of the risks that she accepts to take, with or without her consent, when sharing her data with consumers. This will then enable her to make informed and meaningful privacy decisions. To achieve this, the following needs should be considered:

Need 1. Build a global view of the user's situation (e.g., Alice) by gathering context information about the user and her surrounding environment.

Need 2. Infer the privacy risks involved in the current user context, and maintain continuous monitoring of the risk evolution to cope with context changes.

Need 3. Provide a comprehensible overview of risks, which means an overview adapted to the level of expertise of the user. This enables all users to understand the implicit, direct and indirect implications of sharing their data with consumers.

After alerting Alice of the risks involved in her situation, she may want to adapt her data privacy measures to reduce the risks. Nonetheless, such an adaptation can be difficult for her as it also affects the data utility, and thus the quality of associated services, which might be important to her as well. For instance, assume that the health services are critical for Alice; stop sharing her location data can lead to eliminate the risks of disclosing her habits, behaviors and health conditions, but also to lose the health services received in exchange. This raises consequently the following need:

Need 4. Assist the user in optimizing data utility-privacy decisions in a way to satisfy her privacy requirements and preferences while also maximizing the quality of the main services received in exchange for her data.

However, continuously balancing data privacy can lead sometimes to large gaps in precision between sequential data values, which mainly occurs when sharply decreasing the level of data protection. Such a happening improves the capabilities of an adversary to estimate, with high confidence, the real values of protected data, entailing temporal privacy leakages for the user (e.g., Alice). Accordingly, the following need emerges when considering data utility-privacy optimizations:

Need 5. Measure data dependencies during the protection transition phases and study their impact on data protection in order to avoid large gaps in precision, and consequently ensure full protection of the user's privacy.

1.3.2.1 Scientific Challenges

In order to address the aforementioned needs, one needs to: (1) model an expressive representation of the user's situation; (2) reason on the situational information to infer the risks involved; (3) guide the user to optimize her decisions; and (4) study inter-context dependencies and adapt the levels of data protection accordingly. However, when considering all of the above, several challenges emerge:

Challenge 1. ***Coping with the information heterogeneity and semantics*:** Collected context information can be heterogeneous in types, formats, granularity, origins, and uncertainties. In addition, information pieces are often linked to each others through various (complex) relationships, and can have different characteristics and constraints. The framework must therefore be able to handle information heterogeneity and semantics in order to expressively represent various contexts. This challenge will be further detailed in Chapter 2.

Challenge 2. ***Performing a holistic (all-data-inclusive) privacy risk reasoning*:** As discussed above, collected data pieces can be analyzed in isolation, or combined with each other (e.g., electricity consumption and location data) and/or with other side information acquired from external data sources (e.g., profiles on social networks, public databases). This improves the inference capability of data consumers, thereby increasing the sphere of possible privacy risks. Therefore, the proposed risk inference solution should take into account the different data/information shared by the user or available to data consumers from external data sources, and explores how they combine with each other.

Challenge 3. ***Coping with the dynamicity and context-dependency of privacy risks*:** The sensitivity of data may depend on the context [28], [35]. For example, the sensitivity of Alice's location when she is in the pulmonary rehabilitation center is higher than when she is at home, as in this case location data could be exploited to infer the disease of Alice. That is, as context changes, new privacy risks may emerge, while others may disappear or lose in significance. Therefore, the proposed inference solution should keep track of context changes, analyze their impacts on privacy risks, and maintain an updated risk overview.

Challenge 4. ***Achieving user-centric privacy management*:** Individuals may have different levels of expertise to properly express their needs or preferences, and to interact with the system (e.g., understand risks, make privacy decisions). The proposed solution must therefore be user-friendly, allowing the guided assistance to

be tailored to the user's expertise in order to maintain a good quality of human-machine interactions.

Challenge 5. ***Making optimal context-based privacy decisions*****:** The user-privacy decisions depend on her situation (e.g., risks inferred) and preferences. Therefore, the proposed solution should always be able to provide the user with optimal and adaptive protection strategies to cope with the dynamicity of her context and preferences.

Challenge 6. ***Coping with the inter-context data dependency*****:** The protection assigned to data prior to its release may increase or decrease depending on the context-based user decisions. However, significantly lowering the level of data protection makes subsequent data more precise. Due to data correlations, this may entail the revealing of the real value of previous data that needed more protection, resulting in temporal privacy leakage for the user. The proposed solution should therefore be able to trace data dependencies during contextual transitions, and to appropriately tune the decrease of the data protection level when needed in order to ensure full protection of the user's privacy.

Challenge 7. ***Delivering scalability and efficiency*****:** The solution must be scalable, i.e. handles reasoning over an increasing number of context information, including sensed information (e.g., Location), and risks. It should also be fast to support the user in different contexts, especially since user decisions must sometimes be made in real-time. Finally, it should maintain low computational and storage complexity, which makes it operational on various types of devices, including those with limited resources.

Several other challenges may also arise when considering context-aware privacy-preserving and user-centric privacy, however, we focus in our research work on tackling the aforementioned needs and challenges.

1.4 Related Work

Several frameworks were proposed in the literature to address the challenges of context-aware privacy-preserving and secure context awareness in the fields of pervasive IoT environments (or connected environments). Neisse et al. [38] introduced a context-aware security and privacy framework for smart city applications. This approach defines the context by relying on four parameters: time, location, network, and speed. It provides a context-based security policy management to control access to the data of users based on a set of Event-Condition-Action (ECA) rules. It also provides a privacy-preserving mechanism based on pseudonymization and delayed message delivery. Hence, the access to data could be accepted, denied, modified (using pseudonymization), or delayed. Matos et al. [39] presented an overview of

their context-aware security framework, that aims to provide authentication, authorization, access control, and privacy-preserving in IoT environments. However, the authors only provided a brief description of their framework modules without explaining how privacy is approached in their solution. Sylla et al. presented in [40] a global vision of their context-aware security and privacy as a service (CASPaaS) framework for IoT environments. They briefly discussed the role of each module. The privacy module will be able to continuously analyze the user context and inform her accordingly about the privacy risks involved. However, they have not yet explored any of the framework modules. Other works were proposed for specific IoT domains. Gheisari et al. [41] proposed a context-aware privacy-preserving framework for smart cities using Software Defined Networking. The authors showed that the privacy is preserved by splitting sensitive data and sending split parts via a secure route. The decision made by the SDN controller is based on data sensitivity (that vary depending on the context) and routes credits. Alagar et al. [42] introduced a Context-Sensitive Role-based Access Control (CRBAC) architecture for IoT-based healthcare applications. The approach defines two types of access control: open access, for authenticated clients/medical devices; and closed Access, for non-member clients/devices. CRBAC is user-centric, where the user privacy requirements are included as context-sensitive rules to be enforced whenever patient health information are shared by things.

1.4.1 Comparative Study

In order to compare the referenced works, we define two levels of comparison criteria: the first level consists of the main foundational principles and privacy measures stated by the Privacy-by-Design and ISO/IEC 27701 standards; the second-level comprises specific criteria associated to the needs and challenges defined in Section 1.3.2. The goal here is to assess how well existing works comply with privacy standards and the aforementioned contextual privacy management challenges. We therefore define the following criteria:

- **Criterion 1.** *Proactive & Preventative*: includes proactive measures to prevent privacy violations, i.e., to prevent privacy risks from materializing.
- **Criterion 2.** *Privacy as the Default Setting*: protects the user's privacy by default without requiring user intervention.
- **Criterion 3.** *Full Functionality - Positive-Sum*: seeks to achieve all objectives in a positive-sum (i.e., win-win manner). We focus here on:
  - *Data Utility-Privacy*: optimizes the data utility-privacy trade-off to meet the privacy needs while maximizing the quality of services received in return.
  - *Scalability*: handles increasing (and decreasing) workloads.
- **Criterion 4.** *Data Privacy Protection*: three sub-criteria are considered:

  - *End-to-End Protection*: ensures data protection during the entire data life-cycle.
  - *Context-aware Protection*: provides context-dependent data protection.
  - *Real-time Protection*: offers real-time data protection.

- **Criterion 5.** *Visibility, Transparency, and Openness*: ensures that the data/service exchanges are established in accordance with the stated promises and objectives.

- **Criterion 6.** *User-centric Privacy*: ensures an appropriate involvement of the user in the protection of her privacy (i.e., empowers user-friendly options). We consider four sub-criteria to cover user-centric dimensions:

  - *User Awareness*: raises user awareness about the privacy risks involved in the data sharing and the protection measures that could be taken accordingly through understandable privacy notices.
  - *User-centric Management*: empowers the user to take control and manage her privacy protection.
  - *User-friendly Guidance*: adapts the level of user assistance to her expertise.
  - *User Preferences*: considers the preferences and interests of the user.

Criteria		Neisse et al. [38]	Matos et al. [39]	Sylla et al. [40]	Gheisari et al. [41]	Alagar et al. [42]	**CaPMan** [5]
Proactive & Preventative		YES	YES	YES	YES	YES	**YES**
Privacy as the Default Setting		YES	-	YES	YES	YES	**YES**
Full Functionality	*Data Utility-Privacy*	PARTIAL	-	PARTIAL	YES	PARTIAL	**YES**
	Scalability	YES	YES	YES	YES	-	**YES**
Data Privacy Protection	*End-to-End Protection*	YES	YES	YES	YES	YES	**YES**
	Context-aware Protection	YES	YES	YES	YES	YES	**YES**
	Real-time Protection	YES	YES	YES	YES	YES	**YES**
Visibility, Transparency, and Openness		YES	NO	-	-	-	**YES**
User-centric Privacy	*User Awareness*	NO	NO	YES	NO	NO	**YES**
	User-centric Management	NO	NO	YES	NO	YES	**YES**
	User-friendly Guidance	NO	NO	-	NO	-	**YES**
	User Preferences	NO	NO	YES	NO	YES	**YES**

[1] - means that the referenced work did not approach this aspect.

TABLE 1.1: Review of Context-aware Privacy Frameworks

Discussion. Table 1.1 shows that none of the aforementioned works fully complies with current privacy standards and needs. Most of these works, i.e. [38]–[40], [42], have only presented an overview of their proposed frameworks and briefly explained how they work. The frameworks in [38], [40], [42] have partially addressed the trade-off between data utility and privacy by integrating access control mechanisms. These mechanisms manage only the access rights of providers to user data,

making the data either accessible in a fine-grained version or not accessible at all. This affects the availability of the services for the user when needed, especially if the full quality of service is not required (e.g., the user may wish to share her presence in the city rather than his exact location with the provider in order to get a list of restaurants present in this city rather than just those nearby). In addition, only works in [40] and [42] have been designed with the objective of involving the user in the management of her privacy and considering her preferences when establishing the context-dependent policies. However, they did not consider the fact that individuals may have different levels of expertise, which could impact the quality of individual-system interactions. Finally, some of these works (i.e., [41], [42]) lack re-usability as they were designed to cope with the challenges of domain-specific applications. Consequently, we detail in the following section our proposed generic and re-usable framework [5], which fully meets the aforementioned criteria.

1.5 Proposal: CaPMan Framework

We present here an overview of our proposal for Context-aware Privacy Management in connected environments, denoted CaPMan. The framework addresses the needs and challenges mentioned in Section 1.3.2. Figure 1.6 presents a detailed view of the framework modules and describes the different user-system interactions. The aim of CaPMan is to introduce a user-centric reasoning system capable of keeping the user up-to-date on her evolving privacy situation and assisting her in the management of her privacy protection. In the following, we start by describing the system operation, and then we detail the framework modules.

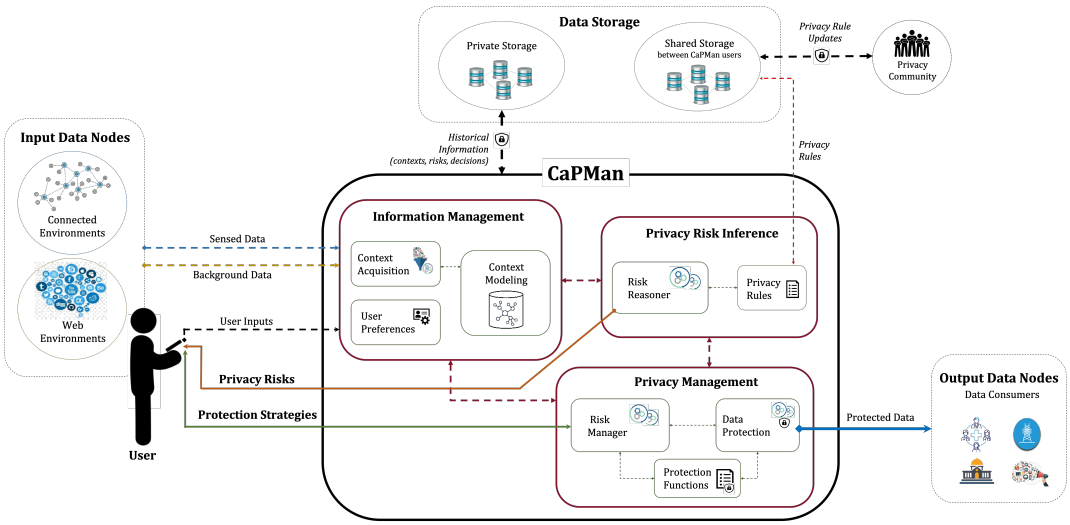

FIGURE 1.6: Overview of CaPMan Framework

The CaPMan system can be embedded on user devices (e.g., computer, mobile phone, tablet) and has two operational modes: passive and active. The passive mode enables the system to be a notifier and a recommender system, where its role consists

of: (i) alerting the user about the direct and indirect privacy risks involved in the sharing of her data; and (ii) recommending data protection strategies to negotiate with relevant data consumers, that lead to meet the user needs and preferences, and to maximize the quality of services received in return. The active mode expands the operations of the passive mode and adds the ability to control and protect user data, based on her decisions, before it is communicated to related data consumers. We consider in our study that all data consumers are not trusted by the user (i.e., service providers and third parties). This is due to the fact that user's data could be misused by consumers, sold to interested third parties without the user consent, or stolen by cybercriminals as data consumers are often victims of cyber-attacks that lead to data breaches. In the active mode, the system is integrated as a middleware between data sources (e.g., sensors, devices) and data consumers. We detail in what follows the system operations and user-system interactions for both modes.

Passive mode. The user specifies her inputs: the list of sensed data that is currently shared with data consumers (e.g., location), and the list of preferences (e.g., privacy requirements, important services). The user preferences are detailed in Chapters 3 and 4. The system, on its side, collects further background data/information describing the user and her surrounding physical environment from other resources (e.g., Web resources). The system models the acquired data/information pieces and the relationships that exist between them to build the overview of the user's situation, and launches the risk reasoner to infer the privacy risks involved in the relevant context. Once the risk inference is completed, the risk manager identifies all possible optimal data protection strategies according to her situation and preferences. Finally, the system provides the user with a risk overview tailored to her expertise, and the list of strategies on which the user can rely when negotiating with data consumers. The levels of user expertise are defined in Chapter 3.

Active mode. Takes into account all passive interactions and operations, and provides additional capabilities for the user to manage her privacy and protect her data. To achieve this, the system has the right to access and control the data values during the collection phase. It therefore operates in this mode as follows. If no risk is inferred, the system continues to generate data values for consumers as received (i.e., without applying additional protection). Otherwise, it alerts the user about the risks she accepts to take, and recommends a list of best protection strategies that could be adopted in this situation. The goal of the strategies here is to improve the user decision making regarding what appropriate amount of protection to associate to the data before communicating it to consumers. Therefore, the system waits for the user to select the strategy to implement and, meanwhile, stops communicating any data to consumers. Upon the user's choice of strategy, the system protects the pending data values and releases a protected version of them to consumers. The system continues to apply the same protection strategy to the received data values until a new context emerges, where the entire reasoning process is relaunched to consider the

changes in the user's situation and their impact on the risk overview and strategies.

The CaPMan system has two types of execution for both passive and active modes depending on user needs: (i) continuous computing, and (ii) on-demand computing, which consists of on-time and scheduled computing. When considering continuous computing, the CaPMan system operates once per context to infer the privacy risks involved and identify the best data protection strategies to be implemented. At this point, when a context change occurs, the system computes the similarity between the current and historical contexts (stored in the user's private storage environment). If a full similarity is detected, the user is given the option of re-applying the actions of the previous similar context (i.e., re-applying the same protection strategy) or launching the global process. This contributes to reducing the computational cost of the system.

In the case of on-demand computing, the system operates only when requested by the user, thereby reducing the use of computational resources. At this stage, if the active mode is enabled, the system continues to protect user's data using the same strategy selected until the system is re-launched. The on-demand computing is not recommended if the user has frequent changes of situation in order to cope with the context-dependency of risks and strategies.

For the default storage management, the system stores locally the context characterizing the present situation of the user only, as well as the associated risks and strategies. This makes the system low-complex in storage, increasing its re-usability on a variety of devices, including those with limited storage resources (cf. Challenge 7). Historical contexts with their associated risks, strategies, and user decisions can be stored in an external storage environment that is private to the user (i.e., the communications between the system and storage environment are secured by the use of appropriate data security mechanisms). Historical information can be used by the system to continuously improve/adjust the default parameters based on user interactions. This will be further detailed in Chapter 4.

1.5.1 Framework Modules

As shown in Figure 1.6, CaPMan is a modular framework comprised of three modules: information management module, privacy risk inference module, and privacy management module. These modules are detailed in what follows.

1.5.1.1 Information Management

Inferring context-aware risks requires first to build up a global view of the user's situation (cf. Need 1). Achieving this requires collecting context information describing the user and her surrounding physical environment. This module is consequently responsible for managing context information (i.e., capturing and modeling

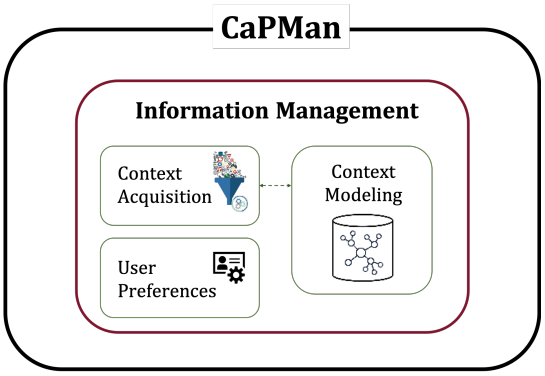

FIGURE 1.7: Information Management Module

information) and user preferences. It comprises the following components (cf. Figure 1.7): (i) *context acquisition*, in charge of capturing data/information about the user and her surrounding environments; (ii) *user preferences*, responsible for managing the preferences of the user; and (iii) *context modeling*, liable for modeling the context information acquired and the relationships that exit among them, which helps in better understanding the user situation. We explored in this thesis the context modeling component, where we proposed a generic and modular ontology for user-context modeling in connected environments, denoted uCSN. The proposal was motivated by the fact that semantic data models allow representation of heterogeneous information with a high expressive power, and maintain flexible data structures which make them able to cope with the dynamicity of the environment. Hence, uCSN introduces a rich vocabulary to represent general information about the user profile, her activity, and the surrounding environment (including smart environment aspects and the mobility of its components). uCSN can be easily aligned with other ontologies, through its pluggable layer, to cover domain-specific knowledge of the user (e.g., medical knowledge) or/and the environment (e.g., knowledge dedicated to smart homes, hospitals, or cities). We further discuss our proposal in Chapter 2.

1.5.1.2 Privacy Risk Inference

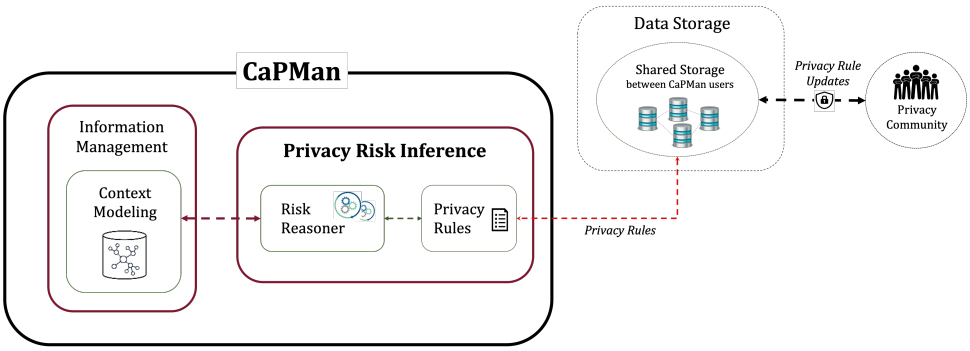

FIGURE 1.8: Privacy Risk Inference Module

This module is responsible for detecting the risks involved in the user context. To achieve this, the module includes two components. First, the *privacy rules* component, which handles the definition/import of privacy rules that specify the risks to be detected by the system. The rules are defined according to the syntax proposed in Chapter 3, and they are used as a reference schema for the reasoning process. This schema is regularly updated by the privacy community that regroups experts belonging to different application domains. This helps in improving the coverage of potential information combinations that entail domain-specific risks, which consequently improve the quality of the risk inference process. The rule updates are imported by the system when relaunching the risk reasoner. It is important to state that the accuracy of the risk inference process depends on the quality of the defined rules. We assume in this study that the privacy rules defined by experts are pre-validated (this validation will be further explored in future work). The second component is the *risk reasoner*, which consists of a semantic rule-based reasoning engine that examines modeled context information and dynamically infers the risks involved. This engine is capable of monitoring the evolution of risk values based on context changes. Chapter 3 details the entire risk inference process. The proposed approach is published [4] in the Future Generation Computer Systems [3] journal:

– *Bou Chaaya, Karam, et al. "Context-aware System for Dynamic Privacy Risk Inference." Future Generation Computer Systems, Elsevier, 2019, 101, pp.1096-1111.*

1.5.1.3 Privacy Management

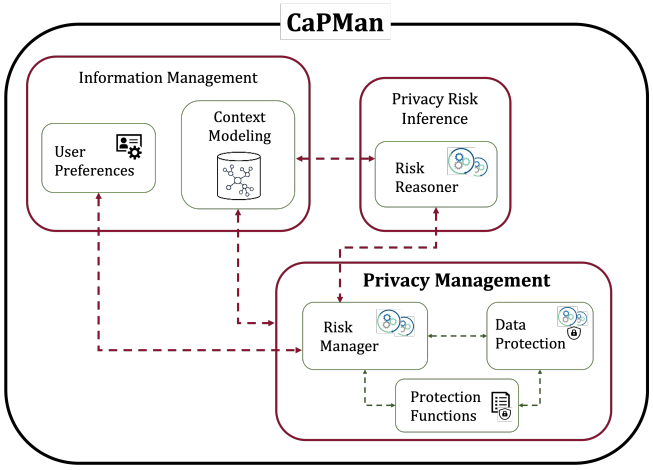

FIGURE 1.9: Privacy Management Module

This module is responsible for assisting the user in the management of her privacy by: (i) assessing and minimizing the risks inferred based on the privacy requirements and interests of the user; (ii) delivering optimized and meaningful strategies;

[3] https://www.sciencedirect.com/journal/future-generation-computer-systems

and (iii) protecting sensor data streams according to the context-based protection strategy selected by the user. In order to do so, the module consists of three components. First, the *risk manager* component, in charge of managing user risks and identifying the best protection strategies to be suggested to the user. Computed strategies are optimal in that they seek to closely satisfy user requirements and preferences while maximizing data utility and minimizing the cost of protection. The risk manager continuously adjusts the strategies provided to cope with the dynamic nature of the user context and preferences. In fact, the user might change progressively her preferences due to the sensitivity of the risks entailed, or the sensitivity of the situation (e.g., private meeting, located in a hospital). The privacy risk management proposal is detailed in Chapter 4, and published [5] in the ACM Transactions on Internet Technology[4] journal:

– *Bou-Chaaya, Karam, et al. "δ-Risk: Toward Context-aware Multi-objective Privacy Management in Connected Environments." ACM Transactions on Internet Technology (TOIT), 2021, 21(2), pp.1-31.*

Second, the *protection functions* component, which includes the list of available protection functions (e.g., random-noise function, generalization function) that the *risk manager* and *data protection* components can rely on during their computing processes. Finally, the *data protection* component, responsible for: (1) selecting the most appropriate protection functions, in terms of compatibility and computational cost, to be executed on sensor data streams to achieve required protection levels (i.e., the protection levels stated in the strategy chosen by the user); and (2) executing selected functions on data pieces in order to communicate protected data to consumers. This component provides therefore context-aware data protection based on user decisions (i.e., when the system operates in active mode). However, ensuring full protection of data requires also to focus on inter-context transitions and their impact on privacy loss. In fact, the protection level assigned to a data stream may increase/decrease from a context to another, making therefore subsequent data values less/more precise at the context transition phase. This raises data leakage problems, especially when protection significantly decreases, which widens the precision gap between prior/subsequent correlated data and makes subsequent data more precise. The large gap in precision improves the capabilities of an adversary when using advanced mining techniques to estimate/infer, with a high confidence, the real values of prior data pieces where protection is critical. This makes consequently the data protection process vulnerable to data inference attacks. To overcome this vulnerability, we proposed a novel stochastic gradient descent approach for privacy-preserving during protection transitions, denoted P-SGD. The goal of this approach is to minimize protection deviation between sequential data values

[4] `https://dl.acm.org/journal/toit`

at the context transition until reaching the targeted protection level (i.e., the protection stated in the newly selected strategy). The gradient descent rate is calculated according to data dependency and protection function dependency (if changed in the new context). The P-SGD approach is detailed in Chapter 5, and published [6] in the proceedings of the 33rd International Conference on Advanced Information Systems Engineering[5] (CAISE'21):

– *Bou-Chaaya, Karam, et al. "P-SGD: A Stochastic Gradient Descent Solution for Privacy-preserving During Protection Transitions." In : International Conference on Advanced Information Systems Engineering. Springer, Cham, 2021. p. 37-53.*

1.6 Report Organization

The remainder of the thesis is organized as follows:

Chapter 2 describes our ontology-based data model that enables the representation of various user situations with high semantic expressiveness power. We review related work on user, environment, and context modeling. Then, we introduce our ontology for user-Context modeling in Sensor Networks (uCSN), which improves the context representation to consider diverse types of: (i) user/environment information (i.e., scalar, multimedia information); (ii) data sources (e.g., sensor, document); (iii) uncertainties (e.g., uncertainties related to the user, the environment); and (iv) environments (i.e., connected/unconnected environments, and environments with static/mobile systems and devices). We do so by defining new concepts and properties, and importing others from well-known ontologies, namely DPV [1], SSN [2], HSSN [43], and W3C Uncertainty Ontology [3]. We keep uCSN generic and reusable in different application domains. Finally, we evaluate the accuracy, clarity, performance, and consistency of the proposal.

Chapter 3 presents the risk reasoner that one uses to infer the privacy risks involved in the user context. We review existing works on privacy risk inference before delving into the proposed context-aware semantic reasoning approach for dynamic risk inference (CaSPI [4]). We address the challenges of (i) increasing expressiveness in risk definitions; (ii) performing a holistic (all-data-inclusive) risk reasoning; (iii) coping with the dynamicity and context-dependency of privacy risks; (iv) dealing with user expertise; and (v) delivering scalability and efficiency. We validate our proposal by developing a prototype, illustrate its functioning from the back-end and front-end, and evaluate its performance by considering different scenarios.

[5] `https://caise21.org/`

Chapter 4 describes the risk manager that evaluates the values of the risks inferred, and then calculates the best data protection strategies that cope with the user's situation and preferences. We present our proposed approach for context-aware multi-objective privacy management (δ-*Risk* [5]). We detail the process followed from the incoming input (e.g., risks involved, user preferences) to the best strategies delivered at the output to the user. We validate our proposal by developing a prototype, illustrate its functioning from the back-end and front-end, evaluate its performance by considering different scenarios, and formally study its effectiveness in strategy identification.

Chapter 5 focuses on overcoming the system's vulnerability to data inference attacks during data protection transitions (e.g., when a context change occurs). We point out the cases that contribute to the temporal data privacy leakages during protection transitions. Then, we introduce our proposed privacy-preserving stochastic gradient descent solution (P-SGD [6]). The proposed solution is connected to the *data protection* component of the framework, and triggered at the protection descent phases to provide an additional layer of protection against data inference attacks. We detail the process followed by P-SGD and illustrate its functioning by executing the developed prototype. Finally, we present the experimentation setup and results.

Chapter 6 concludes the report with a recap of all the aforementioned chapters and discusses in details the next steps and potential future research directions.

Chapter 2

Context Modeling in Connected Environments

"For me context is the key - from that comes the understanding of everything."

– Kenneth Noland

Context-awareness has emerged as a key paradigm for ubiquitous computing and ambient intelligence applications (e.g., IoT-based applications). This paradigm leverages situational information about people and their environments to better improve the quality of machine-to-human communications (e.g., adapt behaviors to people's situation). However, doing so necessitates to represent user situations with a high expressiveness power. Ontology-based data models have been widely adopted as one of the most suitable modeling formats to deal with the heterogeneity of context information.

However, existing ontology-based context models do not fully address the challenges of: (i) covering the representation of domain independent information that describes the main context dimensions, i.e., user, environment, time, and location; (ii) representing diverse data sources from which information can be collected; (iii) representing heterogeneous information in terms of data types and metadata; (iv) representing uncertainty aspects of collected information; and (v) providing a generic model to allow re-usability in various application domains.

In this chapter, we propose uCSN, a generic and modular ontology for user-Context modeling in Sensor Networks. uCSN provides a comprehensive view of the user's situation by introducing new concepts/properties and borrowing others from existing well-known ontologies such as Data Privacy Vocabulary (DPV), Semantic Sensor Network (SSN), Hybrid SSN (HSSN), and Uncertainty Ontology. We evaluate the accuracy, clarity, performance, and consistency of our ontology. The results show that uCSN can be adopted by various context-aware systems, including those requiring high quality of information coverage and/or real-time reasoning (e.g., privacy-preserving systems).

2.1 Introduction

Recent years have witnessed rapid progress in enabling technologies for mobile and ubiquitous computing, ambient intelligence, and machine learning. This allowed the emergence of numerous Context-aware Systems (CAS) in these areas that are capable of perceiving and interpreting changes in people's situation and adapting their operations accordingly. Hence, these systems have paved the way for proactive and intelligent reasoning that helped in minimizing user effort and improving human-computer interactions. Current context-aware applications are impacting various domains, such as healthcare and elderly-care (e.g., [44]–[46]), homes (e.g., [47], [48]), cities (e.g., [49], [50]), military (e.g., [51]), tourism (e.g., [52]), and for different purposes like providing CAS-users with context-driven recommendations, privacy preservation, and so on. Nonetheless, achieving this requires to gather, at any time and any place, as much context information as possible that describes the user (e.g., profile, activity) and her surrounding environment (e.g., environment description, components, characteristics).

The more the information gathered is expressive, the more the CAS is able to understand and interpret the user's situation, which helps in improving the quality of the services/functions delivered in return. However, the system may receive huge amounts of heterogeneous information in terms of data types and formats, originated from different sources (e.g., sensors, Web resources), and describing different context dimensions (i.e., time, location, user, and environment). In addition, it can be uncertain, incomplete, or ambiguous. Information pieces can have different levels of granularity and can be correlated through implicit and explicit relationships. All this makes the modeling of context information a challenging task. Various context modeling techniques exist in the literature [53], including key–value, object oriented, graphical, and ontology-based modeling. However, according to many surveys and studies [53], [54], ontology-based data models have been adopted as one of the most appropriate modeling formats to deal with the heterogeneity of context information. Ontologies allow for information representation with a high semantic expressiveness power, enable comprehensive and complex reasoning over modeled information, and maintain a flexible and extensible data structure.

Several ontologies for context modeling were proposed in the literature [55]–[62], however, they are restrictive due to the following issues: (i) lack of domain-independent information coverage that describes the main context dimensions (i.e., user, environment, time, and location); (ii) lack of considered data and data types (e.g., scalar, multimedia); (iii) lack of considered data sources (e.g., sensors, devices, social network profiles, documents, public databases); (iv) no consideration of information quality aspects (e.g., uncertainty level, nature and type), which is important to limit the impact of context imperfection on the CAS behavior; and (v) lack of genericity/re-usability, most of these approaches (e.g., [58], [59], [62]) are linked to

a specific application domain, i.e. they include domain-specific knowledge, which may increase the semantic complexity or/and computation costs of the data model, and limit its re-usability in other domains.

To address the aforementioned limitations, we present here uCSN, a generic and modular ontology for user-Context modeling in Sensor Networks. uCSN allows the representation of a variety of user situations. It provides a high-level of information coverage of the four context dimensions: time, location, user, and environment. Moreover, it supports the representation of scalar/ multimedia information with their properties, diverse data sources, uncertainty aspects of collected information, and hybrid environments with their static/mobile aspects and components. To achieve this, uCSN introduces new concepts and properties, and imports others from existing well-known ontologies such as: (i) Data Privacy Vocabulary (DPV) [1], to enrich the representation of the user-profile; (ii) Semantic Sensor Network (SSN) [2]/Hybrid SSN (HSSN) [43], to cover the modeling of hybrid connected environments; and (iii) W3C Uncertainty Ontology [3], to represent uncertainty-related aspects. uCSN does not contain domain-specific aspects but can be easily extended and aligned with other ontology models, through its pluggable layer, to cover domain-specific user/environment knowledge (e.g., user-medical data [58], building topology ontology [63]).

The rest of this chapter is organized as follows. Section 2.2 illustrates a scenario that motivates our proposal and identifies the challenges to tackle. Section 2.3 reviews existing ontologies for user, environment, and context modeling. Section 2.4 details the uCSN ontology. Section 2.5 outlines the experimental evaluation of uCSN accuracy, clarity, performance, and consistency. Finally, Section 2.6 presents a summary of the chapter.

2.2 Motivating Scenario

To motivate our proposal, we investigate the following scenario. Consider that Alice uses a context-aware application that provides her with personalized recommendations to protect her privacy. This application requires a high level of information coverage, i.e. it needs to gather all available information that characterize the context of Alice to deliver a good quality of privacy recommendations.

Figure 2.1 illustrates the present situation of Alice. To start with the surrounding environment, Alice is located at her home that hosts various sensors for monitoring purposes like video surveillance cameras and energy-consumption sensors. Alice is a COPD patient who follows her medical treatment from home using a NIV device. In addition, other profile information are also collected by the application from external sources in order to better recommend Alice, such as her marital status and

profile picture (from her Facebook profile), and her date of birth (from her Facebook profile and the public voting database available on the Internet). Moreover, Alice shares the energy-consumption readings of her home, which are sensed by the deployed energy sensors, with an electricity provider, to benefit from personalized recommendations to reduce her energy consumption and bills. She shares also her location data with a healthcare provider, through the GPS sensor embedded on her mobile device, to benefit from a smart ambulance service that she would use in case of respiratory distress. From their side, providers collaborate with third parties interested in exploiting the data of their customers (e.g., Alice) for a variety of purposes, including marketing companies and government agencies.

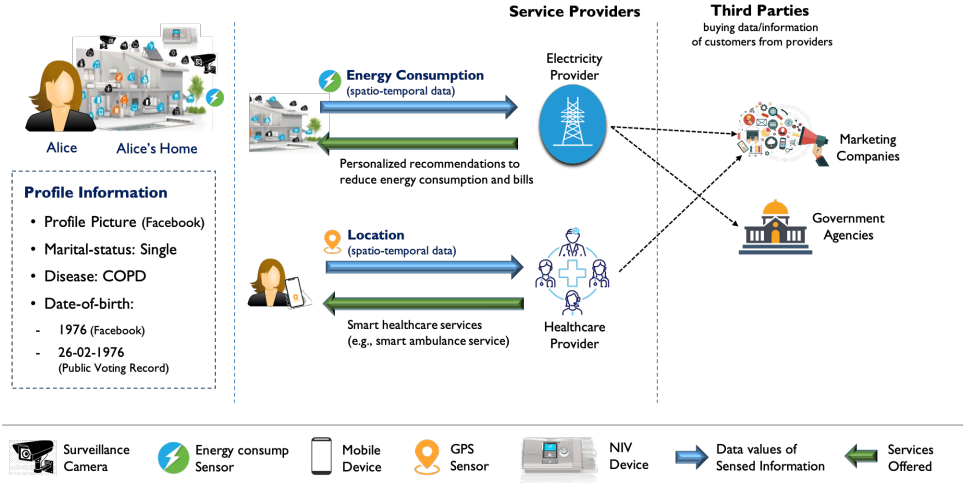

FIGURE 2.1: Running Example

To protect her privacy, the application needs to build the context view by gathering and modeling all these information pieces, which are heterogeneous in types (i.e., scalar and multimedia information), formats (e.g., text, numeric, vector, XML, PNG, MOV), origins (e.g., sensor, social network profile, database, document) and semantics. It also needs to represent the quality of these information that may impact the system behavior. For example, the date of birth information captured from the Facebook profile is incomplete, which might impact the relating recommendations. Moreover, the application may need to track the dynamicity of the environment (e.g., mobile sensor locations and coverage areas, capabilities of devices). It may also get access to sensor data (e.g., video data from surveillance cameras, location data from the GPS sensor) to better monitor the user activity at home.

We considered in this example the case of a privacy-oriented application that requires high-level of information coverage of all context dimensions. Nonetheless, a wide range of context-aware applications exist in many domains [64], where the information coverage requirements vary from one to another depending on their operations. This means they can be interested in tracking user profile, activities, preferences, locations, or/and environment changes. This tracking is achieved by collecting data from deployed/wearable sensors or/and information from other external

resources (e.g., Web resources). Consequently, when modeling context information, several challenges emerge:

- **Challenge 1.** ***Providing high-level information coverage*****:** Context information can characterize different dimensions, i.e., time, location, user, and environment. These information pieces can also have different/complex types of relations among each others, characteristics, and constraints. The data model should therefore provide high-level and expressive information coverage of all context dimensions. This makes it capable of meeting the needs of different applications.

- **Challenge 2.** ***Coping with information diversity*****:** Contextual data/information can be heterogeneous in terms of data types and properties (e.g., scalar and multimedia data). The data model must allow the representation of scalar and multimedia data/information with their properties. This enriches the representation of user contexts, thereby improving the quality of CAS operations.

- **Challenge 3.** ***Coping with source diversity*****:** The data/information can be collected/extracted from a wide variety of data sources, which can be derived from connected environments (e.g., scalar/multimedia sensors, documents) and/or other external resources, such as Web resources (e.g., social network profiles, public databases, emails, documents). Therefore, the data model must support the representation of various data sources with their properties (e.g., origins, data-serialization formats) in order to ensure the traceability of the data/information sources.

- **Challenge 4.** ***Coping with multi-granular information*****:** Context information may have different levels of granularity. For example, the CAS receives two information about the user's location, one indicating that the user is located in the airport (captured from a post on Facebook), and another more precise information indicating the exact location of the user collected from his wearable GPS sensor. Accordingly, the data model must handle the modeling of information with different granularity levels, which enriches the context representation.

- **Challenge 5.** ***Coping with information uncertainty and incompleteness*****:** Contextual data/information is collected, sometimes in real-time, from various/diverse data sources (e.g., sensors, social media platforms). However, the uncertainty level of information may vary due to many factors (i.e., regular and irregular uncertainty [65]) such as node malfunctions/faults or misuse of social media platforms. Furthermore, context information is often incomplete and/or ambiguous [55], [66]. The data model must consequently be able to represent information uncertainties, which can help to minimize negative impacts on the quality/behavior of the CAS.

- **Challenge 6.** ***Coping with environment dynamicity*****:** The surrounding environment of the user is dynamic (i.e., it evolves/changes progressively). This makes

the information modeling process complex, as the data/information received is unpredictable, uncontrollable and unknown in advance.

- **Challenge 7.** ***Delivering re-usability, extensibility and efficiency*****:** The data model must be generic and re-usable in different application domains, i.e., it must not contain domain-specific knowledge. It should also be extensible, so it can be easily adapted to domain-specific particularities. Finally, it must maintain low computational complexity in information retrieval, which makes it also usable by applications where responsiveness and light processing costs are critical.

2.3 Context Background & Related Work

2.3.1 Context Background

The term context has been defined by many researchers. Dey et al. [67] evaluated and stressed the weaknesses of these definitions. They claimed that the definition provided by Schilit et al. [68] is based on examples and cannot be used to identify new context. Further, the definitions provided by Brown [69], Franklin et al. [70], Rodden et al. [71], Hull et al. [72], and Ward et al. [73] used synonyms to refer to context, such as environment and situation. Therefore, these definitions also cannot be used to identify new context. Abowd and Mynatt [74] identified the five W's (Who, What, Where, When, Why) as the minimum information that is necessary to understand context. Schilit et al. [75] and Pascoe [76] have also defined the term context. Dey claimed that these definitions were too specific and cannot be used to identify context in a broader sense, and provided the following definition for context:

"Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

We accept the definition provided by Dey et al. [67] in our research work, because this definition can be used to easily identify whether a provided data element is context or not. Nonetheless, the entity of interest in our work is the user. Therefore, we stress the definition provided by Dey et al. [67] and we consider in our study that the term "context" represents the "group of information" instead of "a single information", and the user-context can therefore be defined as follows:

"A user context is defined as the finite group of information that characterizes the situation of the user in a specific time and space."

The user context is spatio-temporal, and the *user situation* can therefore be regarded as "*the sequence of user contexts in time and space*". We provide in Chapter 3 a

formal definition of *user context* that will be used after (cf. Definition 4). In what follows, we discuss the dimensions used by various researchers to represent a context, and then we highlight the fundamental dimensions considered in our work.

So far, there is no standard to specify what type of information needs to be considered in context modeling [53]. Consequently, the main context dimensions remain neither well defined nor delimited, resulting in a variety of dimensions depending on the perspectives in the field. Authors in [77] considered four dimensions of context information: location, time, identity, and activity. In [78], authors considered the following context dimensions: location, time, identity, and environment. [79] considered user, platform, and environment. [56] considered location, person, activity and computational entity. [55] focused on location, time, user, environment, service and device. [57] considered person, agent, belief-desire-intention, action, policy, time, space and event. In [58], authors were focusing on the medical domain, so they specified eight dimensions for context information, namely location, time, individual, activity, medical, auxiliary, environment, and device.

Our objective here is to provide a generic data model that covers only domain-independent knowledge, i.e. we focus only on domain-independent dimensions. The dimensions outlined in the previous studies characterize four general elements: time, location, user, and environment. From the user perspective, identity, activity, action, belief-desire-intention are properties of the user dimension. Platform, device, service and computational entity can be regarded as properties of the environment dimension. Event is expressed by both time and location dimensions. Therefore, we consider in this study time, location, user, and environment as the four fundamental context dimensions to characterize the user's situation (cf. Figure 2.2).

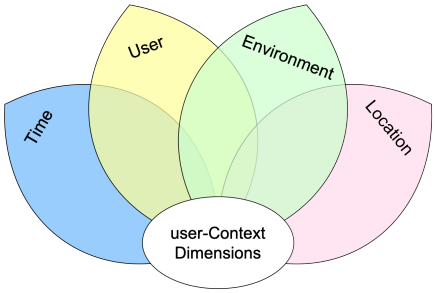

FIGURE 2.2: Fundamental Dimensions of user-Context Information

2.3.2 Related Work

In this section, we study and review existing ontology-based models that represent the user (i.e., user-only models), environment (i.e., environment-only models), and the combination of user and environment with other dimensions (i.e., context models). We compare these works based on the following criteria that are linked to the challenges mentioned in Section 2.2:

- **Criterion 1.** *Re-usability*: The approach is not related to a specific domain and is compatible with different application purposes. In fact, integrating domain specific knowledge can increase the semantic complexity and/or computation costs of the data model. Hence, this criterion indicates if the approach is re-usable in various application domains via $\{YES, NO\}$.

- **Criterion 2.** *Extensibility*: This criterion states $\{YES, NO\}$ if the approach is extensible, i.e., adaptable to domain-specific applications.

- **Criterion 3.** *Information Coverage*: This criterion denotes by $\{YES, NO, PARTIAL\}$ the ability of the approach to cover the representation of domain-independent information related to the different context dimensions:

  (1) Time (i.e., time instants and intervals).

  (2) Location (i.e., location semantics through coordinates, textual descriptions, or/and spatial zones).

  (3) User:
  - Profile, i.e., information describing the user's profile (e.g., identity, preferences, public life, knowledge belief).
  - Activity, i.e., information describing the user's activity (e.g., locations visited, activities performed, data sensed and shared with data consumers in exchange for services).

  (4) Environment:
  - Unconnected environment aspects (e.g., structure, descriptions, devices).
  - Connected environment aspects (e.g., sensors, smart appliances, actuators with their capabilities and properties). In addition, we consider the following two sub-criterion to underline also the ability of the approach to handle sensor diversity/mobility in the environment:
    * *Sensor Diversity*: The environment can host simple sensors or/and more advanced multi-sensor devices that are capable of sensing, processing, communicating, and storing data. In addition, sensors are capable of sensing scalar or/and multimedia data. Hence, this criterion states the ability of the approach to handle sensor diversity in terms of simple sensors/multi-sensor devices and scalar/multimedia sensors.

    * *Sensor Mobility*: The environment can be (i) static, i.e., hosts static sensor nodes or/and multi-sensor devices that do not change locations/coverage areas with time, or (ii) mobile, i.e., hosts at least one mobile sensor/device. Integrating mobile sensing devices in the representation of sensor networks is important to consider mobile data sources from which context information might be collected (e.g., smart phones, drones). However, considering mobility adds challenges related to properly locating and tracking mobile sensors, and updating their coverage

areas when they move. Consequently, this criterion states the ability of the approach to represent sensor mobility.

  - Service (i.e., services provided to the user with its characteristics).

- **Criterion 4.** *Information diversity*: This criterion indicates $\{YES, NO, PARTIAL\}$ if the approach is capable of modeling data/information with different data types.
- **Criterion 5.** *Source diversity*: This criterion indicates $\{YES, NO, PARTIAL\}$ if the approach handles the representation of heterogeneous data sources in terms of origins and types.
- **Criterion 6.** *Information Uncertainty*: This criterion states $\{YES, NO, PARTIAL\}$ if the approach is capable of representing the uncertainty aspects of collected context information.

We do not consider criteria related to information multi-granularity (cf. Challenge 4) and environment dynamicity (cf. Challenge 6) since they are both satisfied when using ontology-based models to which our study is limited.

2.3.3 User Modeling

We present here existing ontologies for user modeling, i.e., ontologies that represent user profile and/or activity information. We start by detailing each work separately (the name of the model is highlighted in bold font). Finally, we evaluate them based on the aforementioned criteria (for criterion 3, we only consider the user information coverage).

A user profile is defined as "*the explicit digital representation of a person's identity. It regroups all personal information describing the characteristics of a person*" [80]. Existing ontologies for user modeling describe the user profile in different manners depending on the usage purpose:

DPV. The Data Privacy Vocabulary (DPV) model [1] is a W3C (World Wide Web Consortium) initiative released in January 2021 (version 2). It introduces classes and properties to describe instances of legally compliant personal data handling according to the EU General Data Protection Regulation (GDPR [11]). This model covers all domain independent profile information and regroups them into different categories such as identifying, demographic, ethnicity, physical characteristics, public life, and preference.

FOAF. The FOAF ontology [81] is one of the most widely used ontologies to model people in the social network field. It specifies a vocabulary that can be used to define, exchange and search for social information that describes people with their social profile characteristics (e.g., first/last name, age, birthday, skypeID, yahooChatID) and their social connections with others.

UPO. In [82], authors proposed an ontological User Profile Modeling for context-aware application personalization within mobile environments (UPO). They introduced concepts and properties to represent the user profile/activity aspects including contact, health, education, capabilities, interests, preferences, and activities.

UPOS. In [83], authors introduced the notion of personalized user profiles and proposed the User Profile Ontology with Situation-Dependent Preferences Support (UPOS). The aim of this ontology was to support the situation-dependent personalization of services within changing environments by splitting the user profile into several profile subsets where each is defined in response to a specific service.

Extended-UPOS. In [84], authors proposed an extension of UPOS ontology for situation-aware social networking. They kept the dynamic aspects of user profiles, and considered the conjunction of context dimensions in order to better identify in real-time the situation of users.

CC/PP. The Composite Capabilities/Preference Profile (CC/PP) model [85] is a W3C initiative that suggests an infrastructure to describe device capabilities and user preferences. CC/PP is developed specifically to facilitate the decision making process of a server on how to customize and transfer web content to the user's device in a suitable format. It can guide to the adaptation of the content delivered to the device according to software terminals, hardware terminals, and applications such as a browser, data types, and protocols.

UP-PwD. In [86], authors proposed a user profile ontology-based approach that provides context-aware personalized services for assisting People with Dementia in mobile environments (UP-PwD). They introduce new classes and properties to represent generic user aspects (e.g., personal information, capabilities, preferences, activities, and locations) as well as other domain-specific aspects such as educational, health, and social information.

2.3.3.1 Comparative Study

Table 2.1 shows that none of the aforementioned works fully considers the entire list of criteria. We discuss in what follows the results of this comparison according to each criterion.

Re-usability. The UPOS [83] and CC/PP models [85] are said to be generic since they do not contain domain-specific knowledge. The rest of data models contain knowledge associated to specific domains (e.g., medical, social, educational), however they can be divided into two categories: (i) domain-independent models; and (ii) domain-driven models. The DPV [1], UPO [82] and UP-PwD [86] ontologies belong to (i), i.e., they integrate domain-specific knowledge through modules/profiles

Criteria		DPV [1]	FOAF [81]	UPO [82]	UPOS [83]	Extended-UPOS [84]	CC/PP [85]	UP-PwD [86]
Re-usability		YES	NO	YES	YES	NO	YES	YES
Extensibility		YES	YES	YES	YES	YES	YES	YES
User Information Coverage	*Profile*	YES	PARTIAL	PARTIAL	PARTIAL	PARTIAL	PARTIAL	PARTIAL
	Activity	PARTIAL	PARTIAL	PARTIAL	PARTIAL	PARTIAL	PARTIAL	PARTIAL
Information Diversity		PARTIAL	PARTIAL	NO	NO	NO	NO	NO
Source Diversity		YES	NO	NO	NO	NO	NO	NO
Information Uncertainty		NO	NO	NO	NO	NO	NO	NO

TABLE 2.1: Comparative Study of Existing User Models

(e.g., health profile, social profile) which makes it easier to exclude the related concepts/properties from the ontology and re-use it in other fields. The FOAF [81] and Extended-UPOS [84] are oriented towards the field of social networks.

Extensibility. Most of the data models (i.e., [1], [82], [83], [85], [86]) are extensible and can be adapted to the particularities of the application domain. The FOAF [81] and Extended-UPOS [84] models can be extended, however, they are limited to their field of interest.

User Information Coverage. All data models (except DPV) partially represent the user profile information and activity information (i.e., performed activities or/and visited locations). The DPV model [1] represents all domain-independent profile information and categorizes them through different classes (i.e., identifying, preferences, demographic, ethnicity, physical characteristic, public life, knowledge belief, and authenticating). However, the coverage of information on user activity remains limited. At present, it only considers user behavioral information.

Information Diversity. DPV [1] and FOAF [81] consider the representation of images along with scalar information, but lack the representation of other multimedia data such as sounds and videos.

Source Diversity. DPV [1] is the only model to support representation of data sources through the `dpv:DataSource` concept.

Information Uncertainty. None of the reviewed models supports the representation of the information uncertainty aspects.

2.3.4 Environment Modeling

We discuss in this section existing ontologies for environment modeling. Environment ontologies vary between those integrating a generic vocabulary to represent different environments and those integrating a domain-specific vocabulary to represent a specific environment in particular (e.g., building, home). In this study, we

only focus on reviewing domain-independent ontologies that cover the representation of many environments. In what follows, we shall follow the same procedure as in the previous subsection 2.3.3. For the information coverage criterion, we only consider here the environment aspects. In addition, criterion 5 (i.e., source diversity) is not considered as the focus here is on how to represent environment descriptions, components (e.g., sensors), and services. Consequently, the only data sources to be considered in such data models are the internal ones, if they exist (i.e., the sensors deployed in the environment).

The surrounding physical environment of the user (e.g., home, mall, street, city) can be a connected environment (i.e., hosts smart cyber-physical systems) or not. Environments might have specific aspects that do not necessarily exist for others. Therefore, we focus here on covering common generic aspects/components across all connected/unconnected environments (e.g., zones, descriptions, components like sensors, devices and appliances). Several ontology-based models exist in the fields of semantic sensor network modeling, IoT/connected environment modeling:

SOSA/SSN. In [2], authors have proposed SOSA/SSN, a joint W3C and OGC (Open Geospatial Consortium) standard, that constitutes the new version of the most foundational ontology for sensors, the Semantic Sensor Network (SSN) Ontology. The main innovation of this SSN new generation has been the introduction of the Sensor, Observation, Sample, and Actuator (SOSA) ontology, which provides a lightweight core for SSN. Together, SOSA/SSN ontologies describe systems of sensors and actuators, observations, platforms, involved procedures, studied features of interest, observed properties, and so forth. SOSA/SSN is a generic and modular ontology that respects the Ontology Design Pattern (ODP), which makes it easier to reuse/extend.

HSSN. In [43], authors have introduced HSSN, an ontology for Hybrid Semantic Sensor Networks. HSSN extends the widely used Semantic Sensor Network ontology (SOSA/SSN) to overcome existing limitations related to sensor diversity, platform diversity and data diversity. HSSN defines new concepts and properties to represent hybrid sensor networks, i.e., networks containing mobile/static sensors, scalar/multimedia properties, and infrastructures/devices as platforms where sensors are deployed.

Other approaches have also extended the SSN ontology. However, they were all contributed before the newly released version of SSN (i.e., SOSA/SSN). Consequently, they tried to deal with the limitations of the old SSN such as the lack of description of essential IoT elements (e.g., objects, actuators, services, etc), services, and so forth. These ontologies are respectively presented in the following.

IoT-O. In [87], authors have introduced IoT-O, a core-domain IoT ontology to represent connected devices networks and their relation with their environment. IoT-O

expands from old SSN with descriptions of sensors, services, units, nodes, things and actuators. It covers the following modules through alignments with existing ontologies: sensing (aligned with the old SSN ontology [88]), acting (aligned with SAN ontology [89]), life-cycle (aligned with Life-cycle ontology [90]), service (aligned with hRest [91], MSM [92], wsmo-lite [93]) and energy (aligned with PowerOnt [94]).

IoT-Lite. In [95], authors have proposed IoT-Lite Ontology, an instantiation of the old SSN ontology. It is a lightweight ontology that represents IoT resources, entities and services. It allows the discovery and interoperability of IoT resources in heterogeneous platforms using a common vocabulary.

IoT Ontology. IoT Ontology [96] is also an expansion of the old SSN. It integrates new concepts such as physical-entity and smart-entity to support semantic expressions for interconnected, aligned and clustered entities.

2.3.4.1 Comparative Study

Criteria			**SOSA/SSN [2]**	**HSSN [43]**	**IoT-O [87]**	**IoT-Lite [95]**	**IoT-Ontology [96]**
Re-usable			YES	YES	YES	YES	YES
Extensible			YES	YES	YES	YES	YES
Environment Information Coverage	*Unconnected Env. Aspects*		YES	YES	YES	YES	YES
	Connected Environment	*Aspects*	YES	YES	YES	YES	YES
		Sensor Diversity	PARTIAL	YES	PARTIAL	PARTIAL	PARTIAL
		Sensor Mobility	NO	YES	NO	NO	NO
	Service		NO	YES	YES	YES	NO
Information Diversity			NO	YES	NO	NO	NO
Information Uncertainty			NO	NO	NO	NO	NO

TABLE 2.2: Comparative Study of Existing Environment Models

As shown in to Table 2.2, the HSSN ontology [43] covers most of the criteria. It lacks only the representation of information uncertainty aspects. We discuss hereafter the comparison of the listed environment models.

All the aforementioned works (i.e., [2], [43], [87], [95], [96]) are generic and extensible. They handle representation of domain-independent aspects of both connected and unconnected environments (e.g., structure, descriptions, sensors, devices, services). Moreover, they all consider representation of simple sensor nodes and multi-sensor devices. However, these models (except HSSN [43]) do not consider the diversity of sensors and information in terms of scalar and multimedia properties, and do not integrate sensor mobility into their representation of sensor networks. Only

HSSN [43], IoT-O [87], and IoT-Lite [95] ontologies represent the services provided to the user with their features (e.g., input/output variables, capabilities). Finally, none of the models tackles the modeling of information uncertainty.

2.3.5 Context Modeling

A broad variety of ontologies exist for context modeling in the fields of connected/smart environments and sensor networks, including ontologies for human behavior/activity recognition. In the following, we start by detailing these works and then we review them according to the aforementioned comparison criteria.

PiVOn. In [55], authors propose the Pervasive Information Visualization Ontology (PiVOn) for context modeling in intelligent environments. They considered the following context dimensions: user, environment, device, and service. Thus, PiVOn is composed of four independent ontologies that represent respectively information related to the considered four dimensions. It integrates the following aspects as properties (or meta-context) of the main dimensions: time, location (e.g., environments where the user is located, GPS coordinates), identity (aligned with FOAF ontology [81] to describe the user profile), and activity. The user context is analyzed from the perspective of the 5 Ws Theory, a journalism principle regarded as basic in information gathering (What, Who, Where, When, Why).

CONCON. In [56], the CONtext ONtology (CONCON) is introduced for modeling context in pervasive computing environments. CONON provides an upper context ontology that captures generic concepts and properties about basic context, and also provides extensibility for adding domain-specific ontologies. Authors consider person, activity (deduced/scheduled activities), surrounding location (i.e., indoor/outdoor space) and computational entity (e.g., device, service) are the fundamental dimensions to define the context.

SOUPA. Authors in [57] propose SOUPA, a Standard Ontology for Ubiquitous and Pervasive Applications. SOUPA consists of two modules. First, the SOUPA-Core that consists of nine ontologies, where together define a generic vocabulary for describing person contact information, beliefs, desires, and intentions of an agent, actions, policies (e.g., rights, obligations), time (i.e., time instants and intervals), space (e.g., geographical regions, geo-spatial coordinates), and events (with their time/space features). The second module is the SOUPA-Extensions, that allows the alignment with other domain-specific ontologies, which justifies its extensibility.

COBRA-ONT. The Context Broker Architecture Ontology (CoBrA-Ont) is an extension of the SOUPA ontology [59]. It focuses on the domain of smart meeting rooms and enriches accordingly the representation of people, places, activities and devices. The main objective of this ontology is to enable knowledge sharing and ontology reasoning within the CoBra (for Context Broker Architecture) infrastructure.

CoDAMoS. In [60], authors propose CoDAMoS, an ontology for context modeling in mobile environments. This ontology has been designed with the aim of solving the challenges of: application adaptation, automatic code generation, code mobility, and generation of device-specific user interfaces. CoDAMoS defines six context dimensions: time, location, user, environment, platform, and service. It provides representation of user preferences/activities/tasks, environmental conditions, device resources (e.g., memory, network, power, and storage) and software (middleware, OS, virtual machine), and the characteristics of the services delivered to the user (e.g., service profile, model, and grounding).

mIO!. In [61], authors propose the mIO! ontology to represent the user context in mobile environments. This accordingly allows to configure, discover, execute, and enhance different services in which the user may be interested. mIO! is a modular ontology, it reuses existing ontologies to enrich some of its eleven core aspects: user (aligned with FOAF [81]), role (i.e., user profiles and preferences; aligned with Reco[1]), environment (aligned with CoDAMoS [60]), location (aligned with SOUPA [57]), time (aligned with W3C Time [97]), service, provider, source, device (e.g., sensors), interface and network.

PalSPOT. In [62], authors introduce an ontology for human activity recognition, denoted PalSPOT. This ontology represent knowledge about user and social activities. It provides also an extensive taxonomy to represent several types of user activities such as personal, physical, professional, and traveling activities. Considering the surrounding environment, PalSPOT is capable of representing their descriptions, as well as the deployed simple sensor nodes. Finally, symbolic locations (e.g., indoor, outdoor) and time granularity are provided.

2.3.5.1 Comparative Study

According to the comparative study presented in Table 2.3, none of the above works fully considers the entire list of criteria. We discuss in the following the results of this comparison according to each criterion.

Re-usability/Extensibility. The COBRA-ONT [59] and PalSPOT [62] ontologies integrate domain-specific knowledge related respectively to the smart meeting rooms and social domains. The rest of works (i.e., [55]–[57], [60], [61]) provide a generic vocabulary to represent the user context. Nonetheless, all aforementioned works are extensible and can be adapted to specific domains.

Information Coverage. All context works satisfy the coverage of information that represent time and location dimensions, as well as the aspects of unconnected environments (e.g., descriptions, environmental conditions). Concerning connected environments properties, the PiVOn ontology [55] handles representation of sensors,

[1] `https://triplydb.com/ctic/reco`

Criteria				PiVOn [55]	CONCON [56]	SOUPA [57]	COBRA-ONT [59]	CoDAMoS [60]	mIO! [61]	PalSPOT [62]
Re-usable				YES	YES	YES	NO	YES	YES	NO
Extensible				YES	YES	YES	YES	YES	YES	YES
Information Coverage	*Time*			YES	YES	YES	YES	YES	YES	YES
	Location			YES	YES	YES	YES	YES	YES	YES
	Environment	*Unconnected Env. Aspects*		YES	YES	YES	YES	YES	YES	YES
		Connected Environment	*Aspects*	YES	NO	NO	NO	NO	PARTIAL	PARTIAL
			Sensor Diversity	PARTIAL	NO	NO	NO	NO	PARTIAL	NO
			Sensor Mobility	PARTIAL	NO	NO	NO	NO	NO	NO
		Service		YES	NO	NO	NO	YES	YES	NO
	User	*Profile*		PARTIAL	PARTIAL	PARTIAL	PARTIAL	PARTIAL	PARTIAL	PARTIAL
		Activity		PARTIAL	PARTIAL	PARTIAL	PARTIAL	PARTIAL	PARTIAL	PARTIAL
Information Diversity				NO	NO	NO	NO	NO	NO	NO
Source Diversity				NO	NO	NO	NO	NO	YES	NO
Information Uncertainty				NO	NO	NO	NO	NO	NO	NO

TABLE 2.3: Comparative Study of Existing Context Models

actuators, dependent/autonomous devices, and multi-sensor devices with their capabilities. The mIO! ontology [61] introduces a specific taxonomy limited to the representation of devices, including simple sensor nodes and multi-sensor devices. The PalSPOT ontology [62] is limited to the representation of simple sensor nodes. As regards sensor mobility, PiVOn [55] is capable of describing only current locations of sensors/devices. However, it does not associate temporal entities to these locations, which denies the ability to track mobile sensors. For the service description, only PiVOn [55], CoDAMoS [60], and mIO! [61] provide classes and properties to describe services with their characteristics (e.g., service profile, model). When considering the coverage of user information, all works reviewed are partially representative. None of them fully covers the representation of domain-independent information that characterize the user's profile and activity.

Information Coverage/Uncertainty & Source Diversity. The compared context models consider only scalar context information. They lack multimedia data/information in their context representation. They also lack representation of uncertainty features of collected context information. As for source diversity, only mIO! [61] considers multi-source modeling through its provided source ontology.

2.3.6 General Discussion

To summarize the previously-detailed studies, none of the existing context models fully answers the list of criteria. They mainly share limitations related to: (i) covering the aspects and properties of connected environments, and the information that characterize the profile and activities of the user; (ii) handling the representation of diverse information (in terms of types/formats) and diverse data sources (in terms of origins/types); and (iii) representing the uncertainty features of collected context information.

When considering the user-only models, we notice that the DPV [1] model presents a rich vocabulary to represent all domain-independent profile information according to the EU GDPR [11]. The generic profile classes can be easily extracted from this model since they are grouped into categories (e.g., demographic, public life, preference), so they can be easily distinguished from domain-specific categories (e.g., medical, financial). As well, when considering environment-only models, we notice that the HSSN ontology [43], which extends the well-known and widely used SOSA/SSN standard ontology [2], is generic and extensible. It introduces classes and properties to represent: (i) structure/component aspects of an environment; (ii) scalar/multimedia sensors and multi-sensor devices; and (iii) services. Moreover, it integrates mobility-related classes to represent mobile sensors and keep track of their locations and coverage areas. Accordingly, we decide to import classes/properties from both DPV and HSSN ontologies to cover the representation of the user's profile and environment.

2.4 uCSN Ontology

In this section, we detail our proposed ontology for user-Context Modeling in Sensor Networks, denoted uCSN. This ontology addresses the challenges mentioned in Section 2.2. It introduces new concepts and properties, and imports others from DPV [1], HSSN [43], SOSA/SSN [2], and W3C Uncertainty Ontology [3], in order to provide a comprehensive view of the user's situation. The following prefixes `dpv:`, `sosa:`, `ssn:`, `hssn:`, `mssn:`, `uo:`, and `time:` refer to DPV [1], SOSA [2], SSN [2], HSSN [43], MSSN [98], Uncertainty [3] and Time [97] ontologies respectively. We start first by presenting an overview of our ontology.

2.4.1 Overview of uCSN

The uCSN ontology is comprised of two main layers as illustrated in Figure 2.3. The core layer (i.e., yellow layer), is composed of the generic core concepts to represent the context dimensions: user (i.e., profile and activity), environment (e.g., descriptions, devices, sensors, services), time, and location (e.g., events). The second layer (i.e., orange layer) is a pluggable layer that allows the alignment with external ontology-based models to represent domain-specific knowledge related to the user (e.g., medical, social) or a particular environment (e.g., home, building, city). Consequently, the core layer ensures the genericity of the uCSN ontology and the pluggable layer justifies its extensibility. Full documentation of the uCSN ontology is available at this link[2]. Also, the ontology files are accessible online[3] for download.

[2] `https://spider.sigappfr.org/uCSNdoc/index-en.html`
[3] `https://spider.sigappfr.org/research-projects/ucsn/` (Ontology Files)

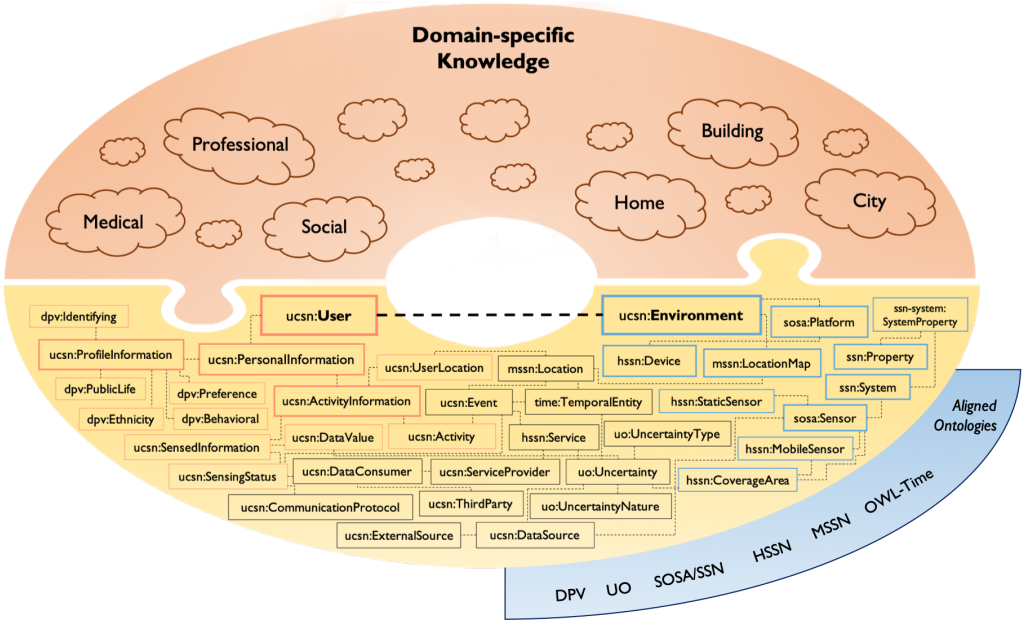

FIGURE 2.3: Overview of the uCSN Ontology

The core layer includes concepts to describe general aspects such as entities, data sources, events, uncertainty features, and so forth (cf. Figure 2.3). It also includes other concepts regrouped into two modules: (i) user module, contains the concepts that characterize the user (i.e., profile and activity); and (ii) environment module, comprises the concepts that describe the environment of the user. In the following, we begin first by defining the entities, and then we explore the user and environment modules with their related general concepts (e.g., data sources, events).

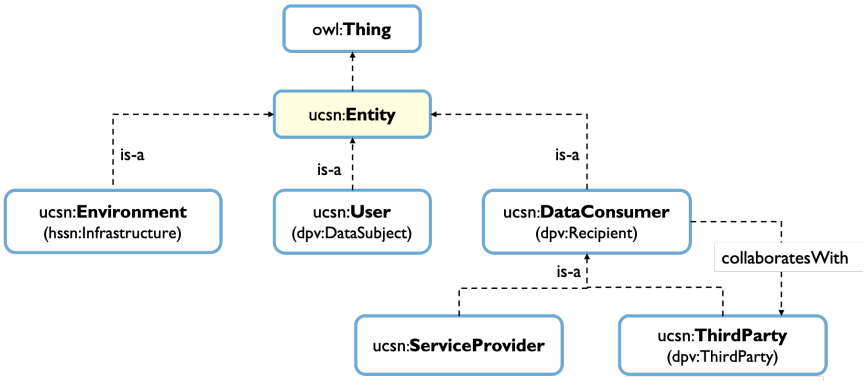

FIGURE 2.4: Entity Representation

Figure 2.4 illustrates the entities considered in uCSN. An entity, expressed by the concept `ucsn:Entity`, can be: (i) the user of interest (or data subject according to DPV [1]), whose data/information is collected, held, shared, or/and processed by other entities (e.g., service providers), represented by the concept `ucsn:User`; (ii) a physical environment, defined by the concept `ucsn:Environment`; or (iii) a data consumer (or recipient according to DPV [1]), denoted by `ucsn:DataConsumer`. The data

consumer is a stakeholder interested in collecting and/or exploiting the information of users. It can be: (1) a `ucsn:ServiceProvider`, first-party responsible for collecting the user's data/information in exchange for services; or (2) a `ucsn:ThirdParty`, external entity interested in buying user's data/information from a principally involved party (i.e., user/service provider) and exploiting it. A `ucsn: DataConsumer` can collaborate with many `ucsn:ThirdParty` (`ucsn:collaboratesWith` property) and exchange customer data for different purposes specified in the relevant agreements.

2.4.2 User Module

The user module contains the user's personal information, expressed by the concept `ucsn:PersonalInformation` (see in Figure 2.5). A `ucsn:PersonalInformation` is spatio-temporal and can be of two types: (i) information that characterizes the user profile, represented by `ucsn:ProfileInformation`, which barely changes over time/space; or (ii) information characterizing the user's activity, represented by `ucsn:ActivityInformation`, which varies over time/space depending on the evolution of the user's situation. The user has many `ucsn:PersonalInformation` as shown in Figure 2.6, each of which is captured from a `ucsn:DataSource` (e.g., sensor, social network profile) and shared with many `ucsn:DataConsumer` (e.g., Facebook).

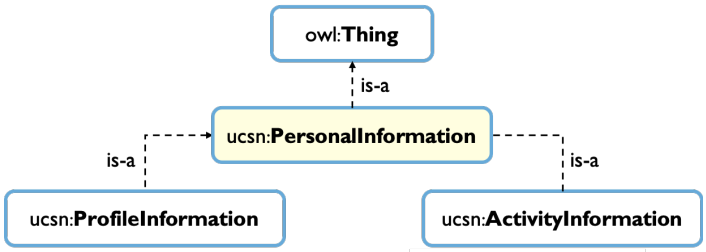

FIGURE 2.5: Personal Information

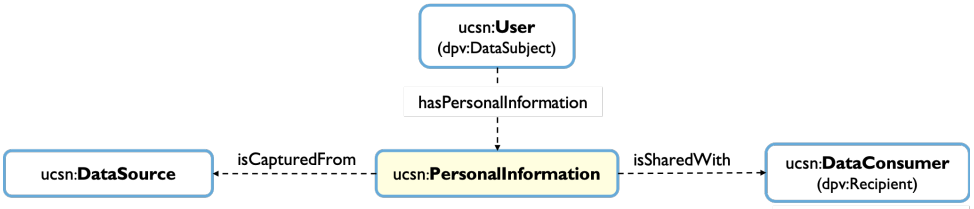

FIGURE 2.6: Personal Information Properties

Source Diversity. Figure 2.7 shows that uCSN is capable of representing different types of data sources, i.e. sensors, devices and other external sources (e.g., social media platforms, databases, documents), through the `sosa:Sensor`, `hssn:Device`, and `ucsn:ExternalSource` concepts (cf. Challenge 3). The data source's origin, URI identifier, and data-serialization format can be also represented through the following properties: `ucsn:origin`, `ucsn:uri-identifier`, and `ucsn:serialization-format`.

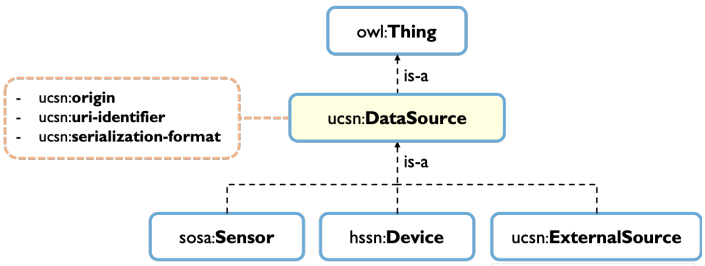

FIGURE 2.7: Source Diversity

In order to enrich the representation of the generic profile information, we import previously-defined concepts from the DPV model [1] and integrate them as sub-concepts of `ucsn:ProfileInformation`. The profile information are consequently grouped into different categories such as `dpv:Identifying`, `dpv:Preference`, and `dpv:PublicLife` (cf. Figure 2.8). The categories are detailed in the next sub-section. Each `ucsn:ProfileInformation` has a specific time and location of capture represented by the `ucsn:hasCaptureTime` and `ucsn:hasCaptureLocation` properties related to the `time:TemporalEntity` and `mssn:Location` concepts respectively.

In addition to profile information, it is also important to monitor the activity of the user. This helps to better understand the current situation of the user and, consequently, to improve the quality of the reasoning process. However, achieving this requires to gather tracking information describing activities performed, locations/-places visited, and user behavior, and to keep track of the information sensed by deployed/wearable sensors. Accordingly, the `ucsn:Activity`, `ucsn:UserLocation`, `dpv:Behavioral`, and `ucsn:SensedInformation` concepts are added as sub-concepts of `ucsn:ActivityInformation` (cf. Figure 2.8). Further details on how to consider the dynamicity of the user's activity are provided in sub-section 2.4.2.2.

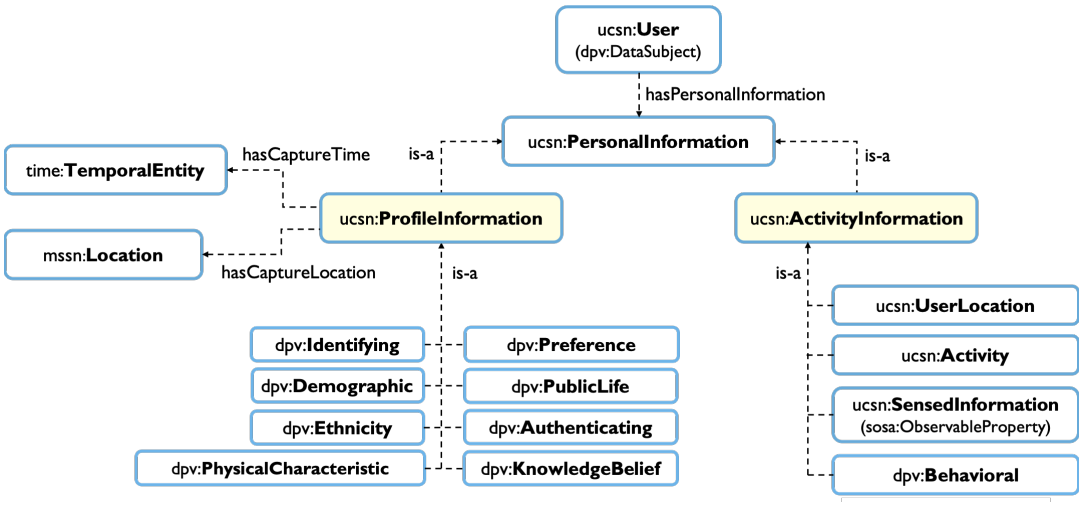

FIGURE 2.8: Profile/Activity Information

2.4.2.1 Profile Information

Eight main categories of generic profile information are imported from the DPV model [1], described by the following concepts: identifying, demographic, ethnicity, physical characteristic, knowledge belief, public life, authenticating and preference. Figures 2.9, 2.10, and 2.11 detail the list of sub-concepts of each category. The `dpv:Identifying` category contains all concepts describing information that can be used to distinguish or trace the user's identity, including `dpv:Biometric`, `dpv:Name`, `dpv:OfficialID` and `dpv:Contact`. The DPV only considers pictures (i.e., `dpv:Picture`) as identifiable multimedia information. Nonetheless, the CAS may receive video/audio recordings that directly identify the user. Therefore, to consider the diversity of information (cf. Challenge 2), a new sub-concept of `dpv:Identifying` is added to characterize the identifiable multimedia information, named `ucsn:MultimediaInfo`. This concept comprises three sub-concepts, namely `ucsn:IdentifiableImage`, `ucsn:IdentifiableAudio`, and `ucsn:IdentifiableVideo` to respectively represent the identifiable images, audios, and videos of the user (see in Figure 2.9).

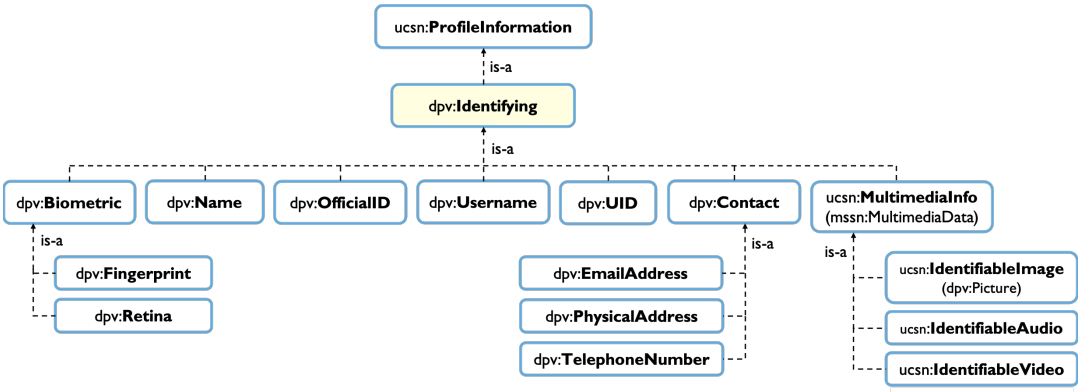

FIGURE 2.9: Profile Information (part-1)

As shown in Figure 2.10, `dpv:Demographic` contains three sub-concepts to represent `dpv:Geographic`, `dpv:IncomeBracket`, and `dpv:PhysicalTrait` information. `dpv:Ethnicity` comprises three sub-concepts to describe `dpv:EthnicOrigin`, `dpv:Race`, and `dpv:Language` information. The `dpv:PhysicalCharacteristic` category consolidates concepts characterizing the physical characteristics of the user, including `dpv:Age`, `dpv:Gender`, `dpv:Height`, and `dpv:Weight`. The `dpv:KnowledgeBelief` category contains three sub-concepts to represent information about the `dpv:Thought`, `dpv:ReligiousBelief`, and `dpv:PhilosophicalBelief` of the user. According to Figure 2.11, `dpv:PublicLife` comprises concepts representing public life information about the user like `dpv:MaritalStatus`, `dpv:PoliticalAffiliation`, `dpv:Character`, and `dpv:SocialStatus`. The `dpv:Authenticating` category regroups authentication information of the user like `dpv:Password`, `dpv:PINCode`, and `dpv:SecretText`. Finally, the `dpv:Preference` category contains concepts characterizing the preferences

of the user like `dpv:Opinion`, `dpv:Interest`, and `dpv:PrivacyPreference`.

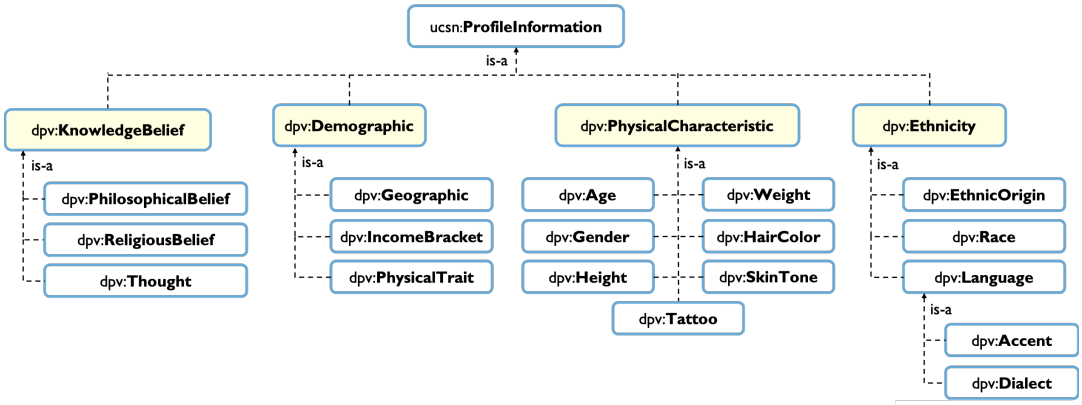

FIGURE 2.10: Profile Information (part-2)

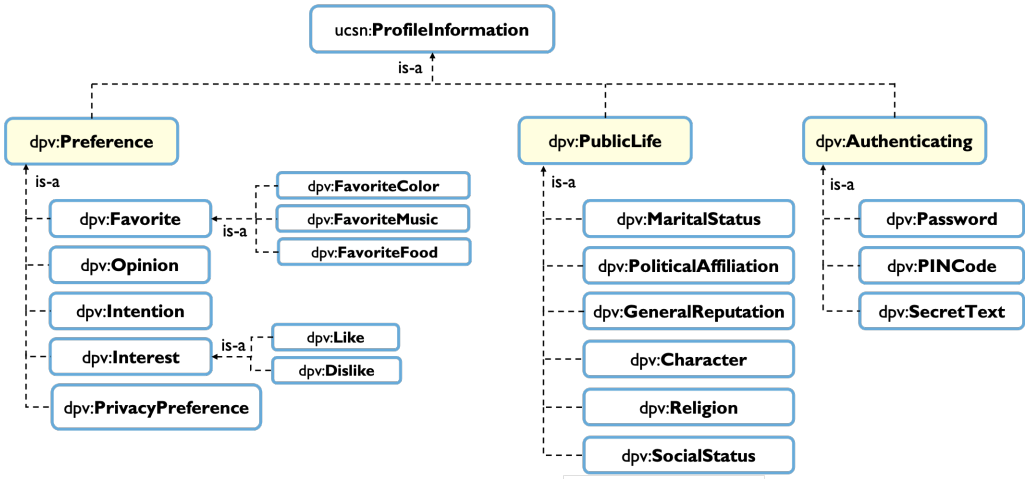

FIGURE 2.11: Profile Information (part-3)

2.4.2.2 Activity Information

Various kinds of information collected can trace the activity of the user in the area of connected/pervasive environments. As shown in Figure 2.8, it can be information describing: (i) the locations or places visited by the user (e.g., the user added a check in post in Paris airport on Facebook), represented by the concept `ucsn:UserLocation`; (ii) the activities performed by the user (e.g., Watching TV or sleeping in the domain of smart homes), expressed by `ucsn:Activity`; and (iii) the evolving user behavior (e.g., attitude/personality changes, calls made/received), modeled through the `dpv:Behavioral` concept. It can also be the information collected from the sensing nodes/devices that exist in the user's environment, represented by the concept `ucsn:SensedInformation`. All these information pieces are regrouped by the super-concept `ucsn:ActivityInformation`. However, the user activity is dynamic, which means it varies over time and space. This necessitates the incorporation of relevant concepts and properties in order to monitor activity changes. Consequently,

for each `ucsn:UserLocation`, a `time:TemporalEntity` is associated to indicate the time instant/interval of this information (cf. Figure 2.12).

A `ucsn:Activity` depends on the domain of interest. For example, the activity can be showering or eating in the smart home domain, as it can be moving hand or using medical equipment in the medical domain. Hence, we do not aim to detail the activity description to keep it re-usable and allow alignments with other activity ontologies depending on the application. However, an activity in general can take place once, as it can be a daily or regular activity. Consequently, the same `ucsn:Activity` can be performed at different times and locations, i.e., an activity can have one or more associated events. An event is a happening that takes place at a particular time (instant or interval) and location [99]. It is represented by the `ucsn:Event` concept (see in Figure 2.13). The `ucsn:hasEventTime` and `ucsn:hasEventLocation` properties are added to map events to their corresponding time instants/intervals and locations. The activity is therefore mapped to its related events through the property `ucsn:isPerformedAt` as illustrated in Figure 2.12.

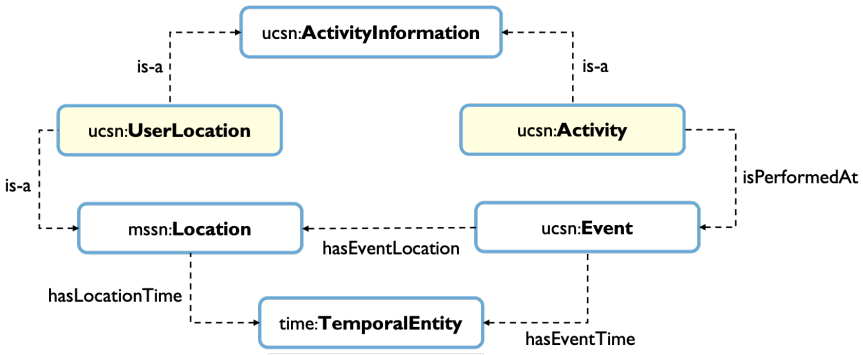

FIGURE 2.12: Location/Activity Information Properties

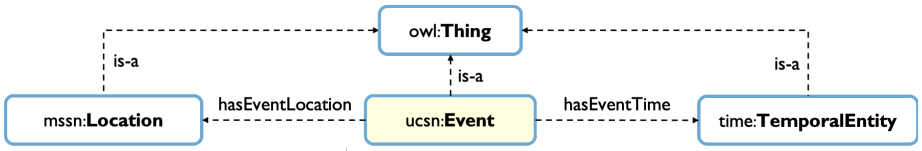

FIGURE 2.13: Event Representation

According to Figure 2.14, a `dpv:Behavioral` comprises several sub-concepts imported from the DPV model to describe different information about user behavior, including `dpv:Attitude`, `dpv:Personality`, `dpv:CallLog`, and `dpv:Demeanor`. Each `dpv:Behavioral` has a specific time and location of capture represented through the `ucsn:hasCaptureTime` and `ucsn:hasCaptureLocation` properties. In addition to behavioral information, a wide range of information can be sensed from the user's environment, represented by the `ucsn:SensedInformation` concept. Indeed, advances in sensor technologies have paved the way for the deployment of various sensors

that are capable of sensing scalar information (e.g., location, temperature, energy-consumption) or multimedia information (e.g., sounds, images, and videos). Collected data values of this information can be very useful, and sometimes mandatory for multiple context-aware applications to monitor and interpret the user's activity. For example, a medical application may need to collect continuous data values from a heart-rate sensor to monitor the user's heart activity. Therefore, in order to cover information diversity, `hssn:ScalarProperty` and `hssn:MultimediaProperty` are added as sub-concepts of `ucsn:SensedInformation` to represent scalar and multimedia sensed information respectively. The `ucsn:SensedInformation` is equivalent to the sensor's `sosa:ObservableProperty` in SOSA/SSN [2].

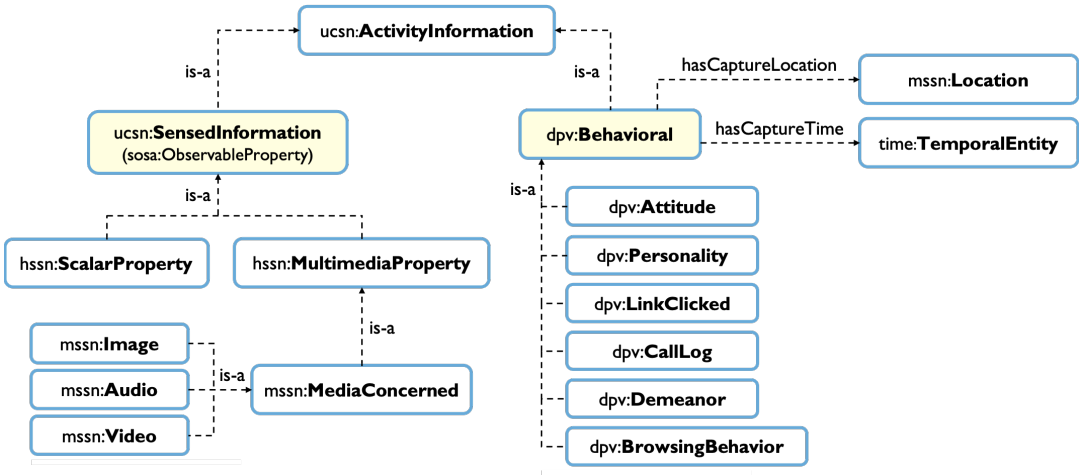

FIGURE 2.14: Sensed/Behavioral Information

Figure 2.15 details the concepts and properties that describe the characteristics of sensed information. A `ucsn:SensedInformation` can describe the user (e.g., location, hear-rate) or her surrounding environment (e.g., energy-consumption of the user's home, room temperature). Hence, the `ucsn:describesEntity` property is added to map the information sensed to its describing `ucsn:Entity`. The `ucsn:Entity` is therefore equivalent to `sosa:FeatureOfInterest` in SOSA/SSN. The same `ucsn:SensedInformation` can be shared with different groups of data consumers using different communication protocols, and its data values can be sensed by different sensors within different sensing events. Thus, the same `ucsn:SensedInformation` can have many sensing statuses, represented by the concept `ucsn:SensingStatus`. Each `ucsn:SensingStatus` indicates a specific set of `ucsn:DataConsumer` with which the information is shared, using a `ucsn:CommunicationProtocol` to communicate collected `ucsn:DataValue` to consumers that are sensed by a `sosa:Sensor` within specific `ucsn:Event` (cf. Figures 2.15 and 2.16). A `ucsn:DataValue` (equivalent to `sosa:Observation`) is spatio-temporal, i.e., it has a time and location of capture described by the `ucsn:hasCaptureTime` and `ucsn:hasCaptureLocation` properties.

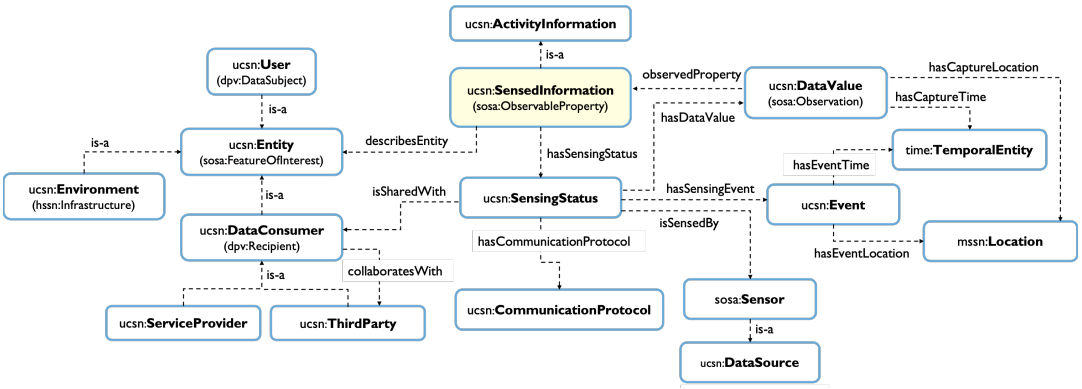

FIGURE 2.15: Characteristics of Sensed Information

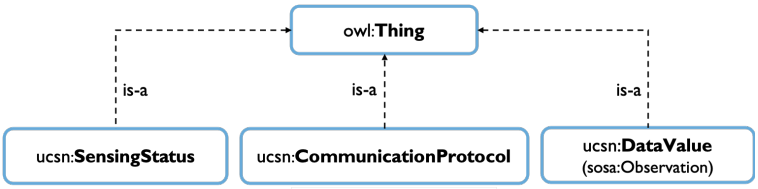

FIGURE 2.16: Sensing Status/Communication Protocol/Data Value

2.4.3 Environment Module

The environment module includes aspects that allow the description of the user's surrounding environment. We aligned this module with the SOSA/SSN [2] and HSSN [43] ontologies that provide a rich vocabulary to represent the environment's structure (e.g., location map), systems deployed (e.g., sensors, actuators), devices, and services. We detail in the following each of these aspects.

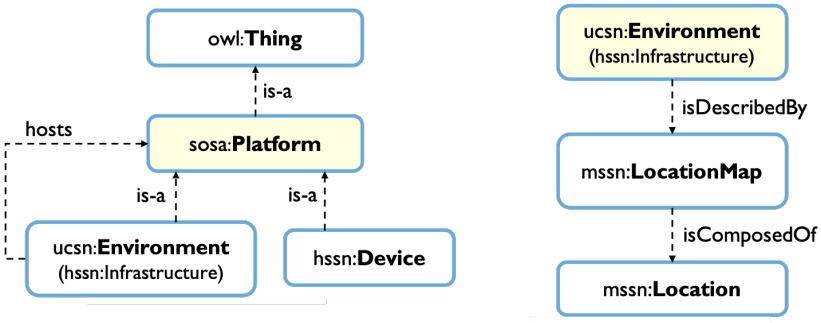

FIGURE 2.17: Environment Representation

In SOSA/SSN, sensors are deployed on platforms. The HSSN model extends this description and add two child concepts of `sosa:Platform` to distinguish between: (i) `hssn:Infrastructure`, a physical environment having locations where sensors could be deployed, which is equivalent to `ucsn:Environment`; and (ii) `hssn:Device`, an electronic equipment where sensors could be embedded (e.g., smart phone, drone). This distinction is illustrated in Figure 2.17. A `ucsn:Environment` can host many `sosa:Platform`, i.e. can host other environments (e.g., cities host buildings, houses host rooms), or devices (e.g., the user's home hosts mobile phones). Regarding the

structure of the environment, Figure 2.17 shows that each `ucsn:Environment` is described by a `mssn:LocationMap` that is composed of a set of `mssn:Location`.

Figure 2.18 details the general representation of devices. A `hssn:Device` has `hssn:Hardware` features related to storage, processing, communication, and power supply, in addition to the ability of embedding sensors via its expansion card. These features are respectively represented by the `hssn:Memory`, `hssn:NetworkInterface`, `hssn:Processor`, `hssn:PowerSupply`, and `hssn:ExpansionCard` concepts. The `hssn:Software` is also considered in the device modeling.

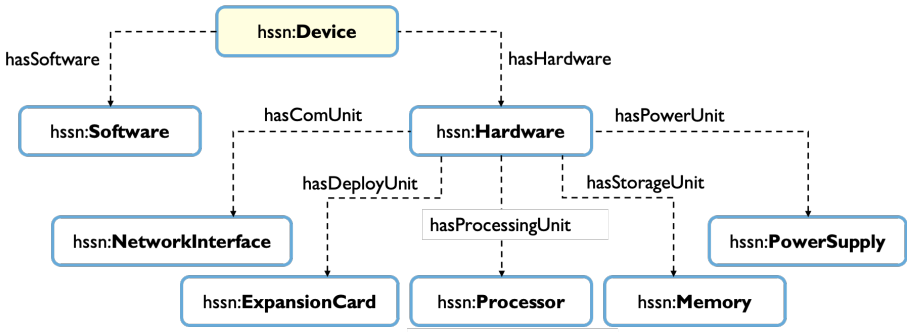

FIGURE 2.18: Device Representation

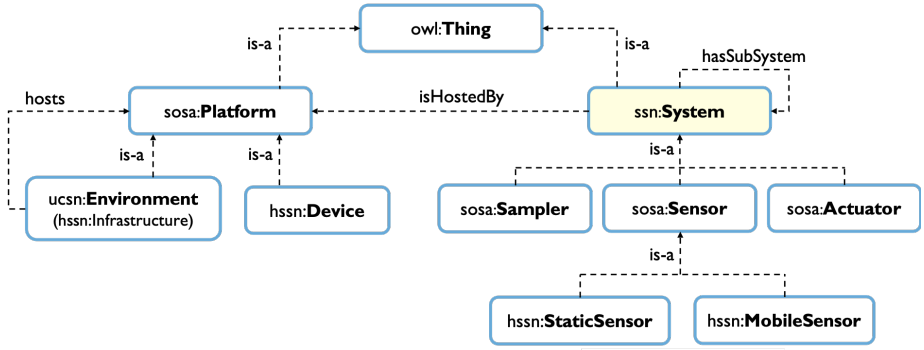

FIGURE 2.19: System Representation

For the system description, the combination of SSN/HSSN ontologies allows the representation of different types of `ssn:System` that could be hosted by platforms (i.e., environments or devices), including samplers, actuators, and sensors (see in Figure 2.19). Sensors vary from `hssn:MobileSensor` that has the ability to move or change location, to hssn:StaticSensor that does not change location in time. As shown in Figure 2.20, each `sosa:Sensor`, mobile or static, observes a specific `ucsn:SensedInformation`, is located in a specific `mssn:Location`, and has a specific `hssn:CoverageArea`, a geographical zone that limits the sensing activity of a sensor (i.e., any happening outside of this zone is not detected by the `sosa:Sensor`). Mobile sensors are capable of continuously changing their locations and coverage areas. To cover the modeling of sensor mobility, the properties `hssn:hasPastLocation` and `hssn:hasPastCoverageArea` are added to map sensors to their previous locations

and coverage areas (cf. Figure 2.21). From their side, locations and coverage areas are mapped to time instants or intervals in order to allow sensor tracking.

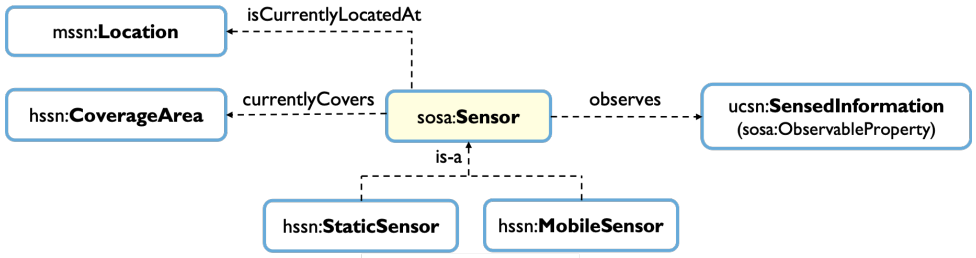

FIGURE 2.20: Sensor Properties

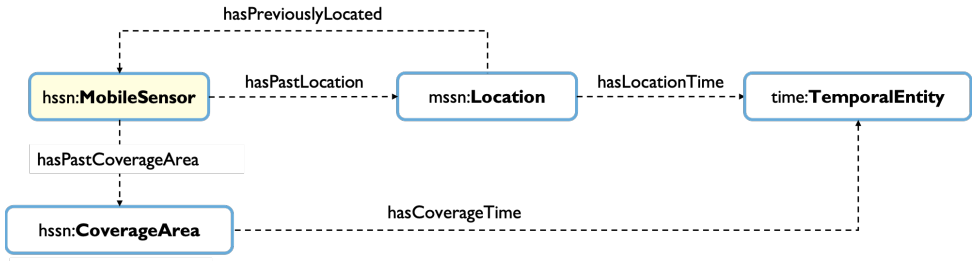

FIGURE 2.21: Sensor Mobility

Numerous services could be provided by many `ucsn:ServiceProvider` to the `ucsn:User` (e.g., personalized recommendations). The `hssn:Service` is provided through a specific `hssn:Device` at many service `ucsn:Event` (cf. Figure 2.22). The HSSN ontology represents only general features of a service to keep it re-usable in different domains. It therefore integrates the following concepts: (i) `hssn:Metadata`, to describe the properties of a `hssn:Service`; (ii) `hssn:Variables`, to represent the set of `hssn:Input` and `hssn:Output` variables of a service (i.e., the set of inputs required for correct service execution and the set of generated results); (iii) `hssn:Interface`, to handle the user/service communications; and (iv) `hssn:Capability`, to describe the functionality of the service.

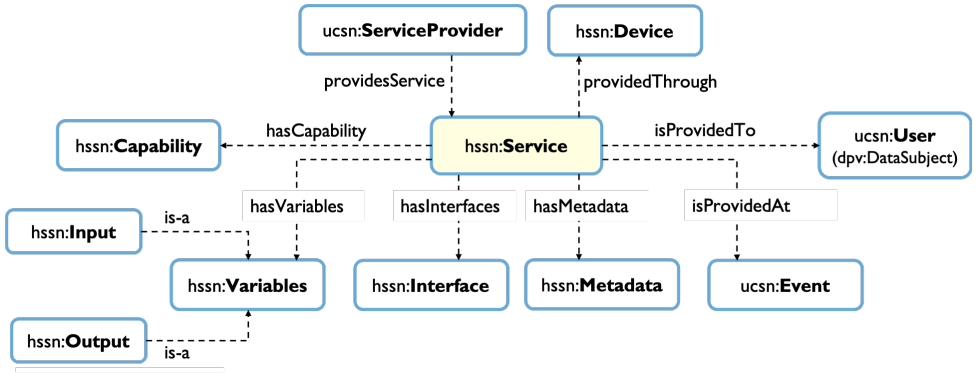

FIGURE 2.22: Service Representation

2.4.4 User/Environment Mediation

We detail here the properties that ensure the interconnection of the user and environment modules. The `ucsn:User` can be located in an `ucsn:Environment`, but also can control the environment, i.e., controls the information sensed from this environment (e.g., in case of her home or office). For example, the user controls the data collected by sensors deployed in her home, such as energy-consumption, temperature, or humidity data. These two relations are respectively represented by the `ucsn:isLocatedInEnv` and `ucsn:controlsEnv` properties. In order to track the presence of the user in the environments, the `ucsn:hasUserPresenceTime` property is added to map the `ucsn:Environment` to the related `time:TemporalEntity`. In addition, a `hssn:Device` can be attached to the `ucsn:User` (e.g., mobile phone, tablet). This is represented by the `ucsn:isAttachedToUser` property. Figure 2.23 illustrates the aforementioned properties.

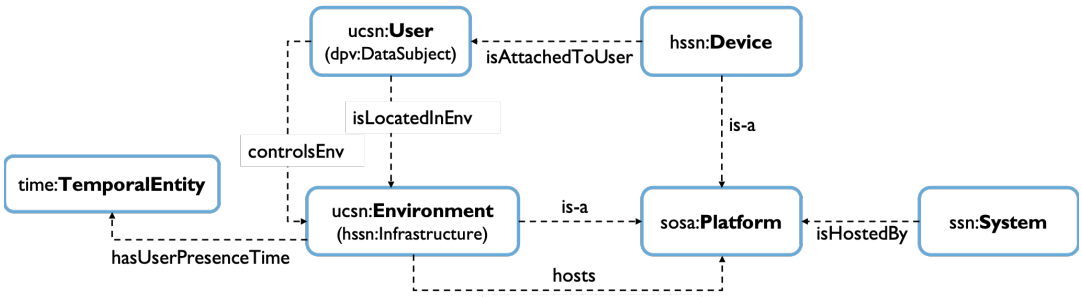

FIGURE 2.23: User/Environment Mediation

2.4.5 Information Uncertainty

The collected context information may be uncertain, incomplete, and/or ambiguous [65], [66]. This affects the quality of corresponding contexts, which can consequently impact the functionality of the CAS and/or the quality of its outputs (e.g., context-aware services). Therefore, we aim to integrate concepts that describe uncertainty aspects of modeled information. For this purpose, three concepts are imported from the W3C Uncertainty ontology [3]: (i) `uo:Uncertainty`, indicates the statement about the uncertainty associated with the information collected; (ii) `uo:UncertaintyNature`, expresses whether the uncertainty is an inherent property of the world or a lack of information; and (iii) `uo:UncertaintyType`, indicates the type of uncertainty (e.g., incompleteness, ambiguity). The properties `uo:nature` and `uo:uncertaintyType` map the `uo:Uncertainty` to its related nature and type. Each `uo:Uncertainty` has an associated value represented by the `uo:uncertaintyValue` property. As shown in Figure 2.24, the `uo:UncertaintyNature` comprises two sub-concepts, `uo:Aleatory` and `uo:Epistemic`, that respectively indicates whether the uncertainty arises from the entities described by the information or from the related data source. In addition, the `uo:UncertaintyType` contains five sub-concepts: (1)

`uo:Ambiguity`, means that the information is not clearly specified; (2) `uo:Empirical`, means that the information about an entity is either satisfied or not for all entities, but it is not known for which entities it is satisfied; (3) `uo:Incompleteness`, means that the information about the entity is incomplete; (4) Inconsistency, means that there is no entity that would satisfy the statement; and (5) Vagueness, means that there is no precise correspondence between the information and the related entities.

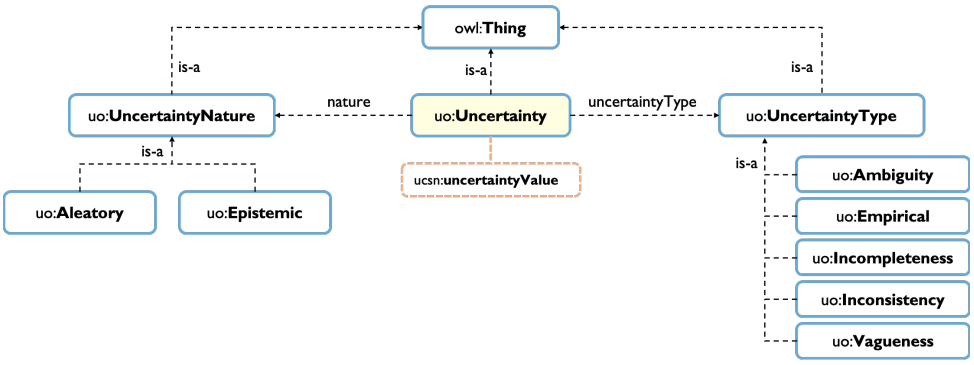

FIGURE 2.24: Uncertainty Representation

Figure 2.25 shows which of the modeled context information is associated with uncertainties. Locations, times, and events might be uncertain. From the user side, collected personal information (i.e., profile and activity information) and sensed data values can be uncertain. From the environment side, location maps, coverage areas, sensing angles (i.e., horizontal and vertical), and service variables can be uncertain.

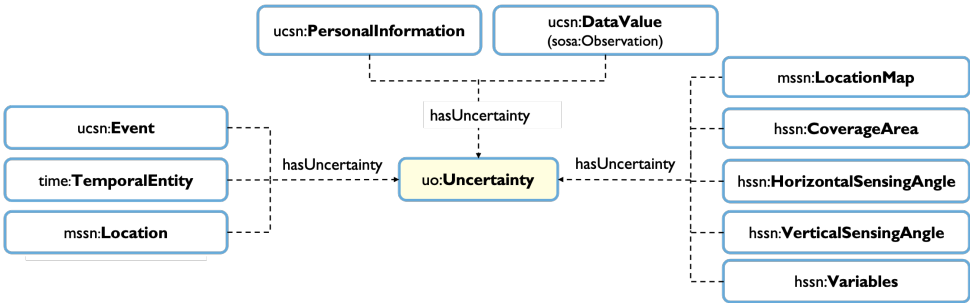

FIGURE 2.25: Uncertainty related to User/Environment Information

2.5 uCSN Experimental Evaluation

In this section, we detail the experimental protocol followed to evaluate both the syntactic and semantic aspects of the uCSN ontology (i.e., the concepts and the semantic inter-concept relations). The objectives of this protocol are:

1. *Accuracy Evaluation*: Checks if the uCSN concepts and properties are capable of answering the challenges mentioned in Section 2.2.

2. *Clarity Evaluation*. Checks if the labels used to describe the newly added concepts and properties are clear and unambiguous to domain stakeholders. The aim is to evaluate the clarity and compatibility of our extensions with respect to the context-awareness domain.

3. *Performance Evaluation*. Measures the impact of the uCSN ontology on performance (i.e., query run time). The aim is to evaluate the feasibility, in terms of performance, of integrating uCSN in context-aware applications.

4. *Consistency Evaluation*. Checks if the added concepts and properties generate inconsistencies (e.g., anti-patterns) within the structure of the ontology. The aim is to evaluate the soundness of the ontology graph.

2.5.1 Accuracy Evaluation

In order to study the accuracy of uCSN, we elaborate a query-based evaluation that highlights the ontology impact towards overcoming the challenges of (i) information coverage, (ii) information diversity, (iii) source diversity, and (iv) information uncertainty. The accuracy evaluation of sensor mobility and diversity is detailed in the HSSN experiments [43]. We start first by detailing the query setup process, then we discuss the obtained results and we compare them according to the expected ones.

2.5.1.1 Query Setup

The aforementioned challenges can be addressed by answering SPARQL queries related to user, environment, and uncertainty information in uCSN. We define in the following the list of queries to execute with respect to each challenge.

Information Coverage. The information coverage queries are divided into user-oriented, environment-oriented and context-oriented queries.

User Information. In order to expressively extract the information characterizing the user from the ontology, we define the following five queries:

- Query 1: Extracts the list of sensed information with their describing entities and sensing statuses (i.e., data consumers with whom information is shared, sensors used to sense related data values, sensing events, and communication protocols used to communicate data to consumers).

- Query 2: Extracts the data values of each sensed information, collected during a specific time interval $[t1; t2]$, with their respective times/locations of capture.

- Query 3: Generates the list of information describing the contextual activity of the user, i.e. user locations, activities performed, behavioral and sensed information with their respective times and locations.

- Query 4: Generates a detailed view of the user profile at a given time instant t. It extracts all profile information characterizing the user at t.
- Query 5: Generates the list of information that express user location semantics, i.e. user locations acquired from external sources (e.g., Facebook), location data values sensed by GPS sensors, and environments where the user is located.

Query 1: Knowing sensed information with their characteristics

```
SELECT distinct ?user ?sensedInfo ?entity ?status ?consumer
                ?sensor ?protocol ?event ?time ?location
    WHERE {
             ?user :hasPersonalInformation ?sensedInfo.
             ?sensedInfo rdf:type ucsn:SensedInformation;
                         :describesEntity ?entity;
                         :hasSensingStatus ?status.
             ?status :isSharedWith ?consumer;
                     :hasCommunicationProtocol ?protocol;
                     :isSensedBy ?sensor;
                     :hasSensingEvent ?event.
             ?event  :hasEventLocation ?location;
                     :hasEventTime ?time.
          }
```

Query 2: Generate the list of data values captured during $[t1; t2]$

```
SELECT distinct ?user ?sensedInfo ?datavalue ?time ?location
    WHERE {
             ?user :hasPersonalInformation ?sensedInfo.
             ?sensedInfo  rdf:type ucsn:SensedInformation;
                          :hasDataValue ?datavalue.
             ?datavalue :hasCaptureTime ?time;
                        :hasCaptureLocation ?location.
             FILTER (?time >= t1  &&  ?time <= t2)
          }
```

Query 3: Knowing the contextual activity of the user

```
SELECT distinct ?user ?activityInfo ?value ?time ?location
    WHERE {
            { ?user :hasPersonalInformation ?value.
              ?value rdf:type ucsn:UserLocation;
                     rdf:type  ?activityInfo;
                     :hasLocationTime ?time. }
            UNION
            { ?user :hasPersonalInformation ?value.
              ?value rdf:type ucsn:Activity;
                     rdf:type  ?activityInfo;
                     :isPerformedAt ?event.
              ?event :hasEventTime ?time;
                     :hasEventLocation ?location.}
            UNION
            { ?user :hasPersonalInformation ?value.
              ?value rdf:type ucsn:SensedInformation;
                     rdf:type  ?activityInfo;
                     :hasSensingStatus ?status.
              ?status :hasSensingEvent ?event.
              ?event :hasEventTime ?time;
                     :hasEventLocation ?location.}
            UNION
            { ?user :hasPersonalInformation ?value.
              ?value rdf:type dpv:Behavioral;
                     rdf:type  ?activityInfo;
                     :hasCaptureTime ?time;
                     :hasCaptureLocation ?location. }
        }
```

Query 4: Generate a view of the user profile at time *t*

```
SELECT distinct ?user ?profileInfo ?value ?time ?location
    WHERE {
            ?user :hasPersonalInformation ?value.
            ?value rdf:type ucsn:ProfileInformation;
                   rdf:type ?profileInfo;
                   :hasCaptureLocation ?location;
                   :hasCaptureTime ?time.
            FILTER (?time <= t)
        }
```

Query 5: Generate the list of information expressing user location semantics

```
SELECT distinct ?user ?location ?time
    WHERE {
            { ?user :hasPersonalInformation ?location.
              ?location rdf:type ucsn:UserLocation;
                        :hasLocationTime ?time. }
            UNION
            { ?user :isLocatedInEnv ?location.
              ?location :hasUserPresenceTime ?time. }
            UNION
            { ?user :hasPersonalInformation ucsn:Location.
              ucsn:Location rdf:type ucsn:SensedInformation;
                            :hasDataValue ?location.
              ?location :hasCaptureTime ?time. }
          }
```

Environment Information. The context-aware application may need to extract an expressive description of the user's surrounding environment, and/or the services provided to her. The following queries answer these needs:

- Query 6: Generates a detailed view of the user surrounding environment, i.e. spatial description, systems (e.g., sensors, actuators) and devices deployed in the environment, as well as the devices attached to the user.

Query 6: Knowing the user's surrounding environment

```
SELECT distinct ?user ?environment ?locationMap ?location
                ?component ?componentType
    WHERE {
            ?user :isLocatedInEnv ?environment.
            ?environment :isDescribedBy ?locationMap.
            ?locationMap :isComposedOf ?location.
            { ?component :isHostedBy ?environment;
                         rdf:type  ssn:System;
                         rdf:type ?componentType. }
            UNION
            { ?component rdf:type hssn:Device.
              ?environment :hosts ?component. }
            UNION
            { ?component rdf:type hssn:Device;
                         :isAttachedToUser ?user. }
          }
```

- Query 7: Generates a detailed view of the services provided to the user, i.e. information on services, related service providers, devices, variables, interfaces, metadata, capability, and associated availability/usage events.

Query 7: Generate a detailed view of the user services

```
SELECT distinct ?user ?service ?provider ?device ?variable
                ?interface ?metadata ?capability ?event
    WHERE {
             ?provider :providesService ?service.
             ?service :isProvidedTo ?user;
                      :isProvidedThrough ?device;
                      :hasVariables ?variable;
                      :hasInterfaces ?interface;
                      :hasMetadata ?metadata;
                      :hasCapability ?capability;
                      :isProvidedAt ?event.
          }
```

Context Information. In order to generate a complete and expressive view of the user's context, we define Query 8 which extracts all of modeled context information with their relationships.

Query 8: Generate the complete view of the user's situation

```
SELECT distinct ?domainType ?domainValue ?relation ?rangeType
                ?rangeValue
    WHERE {
             ?relation rdf:type owl:ObjectProperty.
             ?domainValue ?relation ?rangeValue.
             ?domainValue rdf:type ?domainType.
             ?rangeValue rdf:type ?rangeType.
          }
```

Information Diversity. In order to consider information diversity, on should be able to distinguish scalar and multimedia information. Therefore, Query 9 selects only the multimedia identifiable information of the user, and Query 10 extracts the list of scalar and multimedia sensed information and highlights the type of each one.

Query 9: Knowing the user's multimedia identifiable information

```
SELECT distinct ?user ?mediaType ?mediaValue
    WHERE {
             ?user :hasPersonalInformation ?mediaValue.
             ?mediaValue rdf:type ucsn:MultimediaInfo;
                         rdf:type ?mediaType.
          }
```

Query 10: Knowing the user's scalar and multimedia sensed information

```
SELECT distinct ?user ?sensedType ?sensedValue
   WHERE {
            { ?user :hasPersonalInformation ?sensedValue.
              ?sensedValue rdf:type hssn:ScalarProperty;
                           rdf:type ?sensedType. }
            UNION
            { ?user :hasPersonalInformation ?sensedValue.
              ?sensedValue rdf:type hssn:MultimediaProperty;
                           rdf:type ?sensedType. } }
```

Source Diversity. To highlight the representation of diverse data sources (i.e., sensors, devices, and external sources) with their properties (e.g., origin, data serialisation format), we define Query 11.

Query 11: Generate the list of data sources with their properties

```
SELECT distinct ?source ?origin ?serialisation
    WHERE {
            ?source rdf:type ucsn:DataSource;
                    :origin ?origin;
                    :serialisation-format ?serialisation. }
```

Information Uncertainty. The application may need to extract uncertainty knowledge of modeled information in order to adjust its behavior/outputs accordingly. To do so, we consider three categories to distinguish the uncertainties related to the user, the environment, and the time/location properties. The uncertainty information for the three categories is respectively extracted using Queries 12, 13, and 14.

Query 12: Uncertainties related to the user

```
SELECT distinct ?infoType ?infoValue ?uncertainty ?uValue
                ?uNature ?uType
    WHERE {
            { ?user :hasPersonalInformation ?infoValue.
              ?infoValue rdf:type ?infoType. }
            UNION
            { ?user :hasPersonalInfo ?sensedInfo.
              ?sensedInfo rdf:type ?infoType;
                          :hasDataValue ?infoValue. }
            ?infoValue :hasUncertainty ?uncertainty.
            ?uncertainty :uncertaintyValue ?uValue;
                         :nature ?natureInstance;
                         :uncertaintyType ?typeInstance;
            ?natureInstance rdf:type ?uNature.
            ?typeInstance rdf:type ?uType.  }
```

Query 13: Uncertainties related to the environment

```
SELECT distinct ?infoType ?infoValue ?uncertainty ?uValue
                ?uNature ?uType
    WHERE {
            { ?infoValue rdf:type mssn:LocationMap;
                         rdf:type ?infoClass. }
            UNION
            { ?infoValue rdf:type hssn:CoverageArea;
                         rdf:type ?infoClass. }
            UNION
            { ?infoValue rdf:type hssn:HorizontalSensingAngle;
                         rdf:type ?infoClass. }
            UNION
            { ?infoValue rdf:type hssn:VerticalSensingAngle;
                         rdf:type ?infoClass. }
            UNION
            { ?infoValue rdf:type hssn:Variables;
                         rdf:type ?infoClass. }
            ?infoValue :hasUncertainty ?uncertainty.
            ?uncertainty :uncertaintyValue ?uValue;
                         :nature ?natureInstance;
                         :uncertaintyType ?typeInstance;
            ?natureInstance rdf:type ?uNature.
            ?typeInstance rdf:type ?uType.
          }
```

Query 14: Uncertainties related to Location & Time

```
SELECT distinct ?infoType ?infoValue ?uncertainty ?uValue
                ?uNature ?uType
    WHERE {
            { ?infoValue rdf:type ucsn:Event;
                         rdf:type ?infoClass. }
            UNION
            { ?infoValue rdf:type time:TemporalEntity;
                         rdf:type ?infoClass. }
            UNION
            { ?infoValue rdf:type mssn:Location;
                         rdf:type ?infoClass. }
            ?infoValue :hasUncertainty ?uncertainty.
            ?uncertainty :uncertaintyValue ?uValue;
                         :nature ?natureInstance;
                         :uncertaintyType ?typeInstance;
            ?natureInstance rdf:type ?uNature.
            ?typeInstance rdf:type ?uType.
          }
```

2.5.1.2 Query Run & Discussion

We created a population of individuals and ran the aforementioned queries. Then, we compared the obtained and expected results. We created an environment described by a location map containing 20 locations. This environment hosts 20 actuators, 100 sensors (50 static, 50 mobile, 50 scalar, and 50 multimedia sensors), and 20 devices. Each sensor is located in one location, covers one coverage area, and observes one information (i.e., 50 scalar and 50 multimedia sensed information). Each of the sensed information describes one entity and has one sharing status (for each sensed information: 5 data consumers, 1 sensor, 1 communication protocol, and 1 sensing event associated with 1 temporal entity and 1 location). In addition, each of the sensed information has 50 data values where each has 1 time and location of capture. We created also 10 user locations collected from an external source, 5 activities performed, 2 behavioral information. The user profile is composed of 140 information (20 individuals per sub-concept). Each of the profile and activity information has one data source (140 in total divided into 40 sensors, 40 devices, and 60 external sources). We considered 5 devices attached to the user and 25 services provided to her. We associate an uncertainty value, nature, and type with half of the personal information, data value, time and location individuals. Finally, we consider that 20 of the coverage areas are uncertain (i.e., each of them has an associated uncertainty individual with its corresponding nature and type).

We ran queries 1-14 on the population of individuals, and for each case, the returned results matched exactly the expected ones. Therefore, the query results confirmed that our ontology is able to accurately answer the challenges mentioned in Section 2.2. In addition, the results show that uCSN provides a high-level of generic information coverage to represent the user and the environment. This makes it usable by various context-aware systems in multiple domains, including those requiring high quality of information coverage (e.g., privacy-preserving systems).

2.5.2 Clarity Evaluation

We created an evaluation form[4] to assess the ambiguity of the labels used to describe the uCSN concepts and inter-concept relations (i.e., the object properties). We focus only on evaluating the ambiguity of the newly defined concepts/properties. We sent the form to 50 ontology and sensor network experts, divided into 3 categories as shown in Figure 2.26: 25 computer scientists (i.e., assistant professors, associate professors, full professors, and PhD students), 18 research engineers experts in the fields of semantic web and context-aware computing, and 7 experts in network engineering. From a demographic standpoint (cf. Figures 2.26 and 2.27), the survey

[4] `http://bit.ly/uCSN-clarity-evaluation`

respondents are divided into 31 males and 19 females, and belong to different countries: France (26 respondents), Lebanon (12), United Arab Emirates (6), United States of America (2), Tunisia (2), Germany (1) and United Kingdom (1).

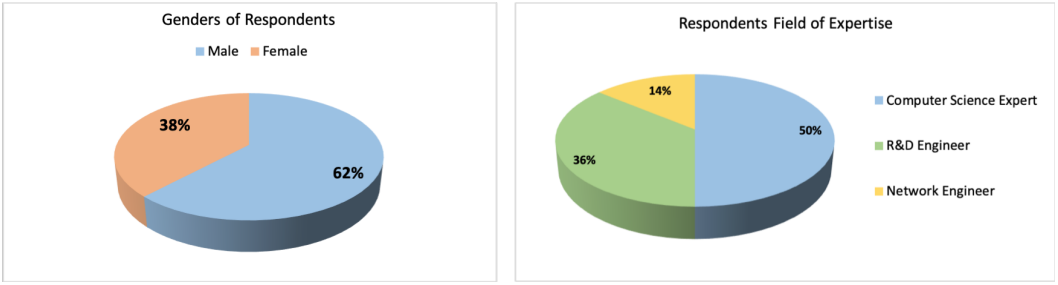

FIGURE 2.26: Respondents Genders/Fields of Expertise

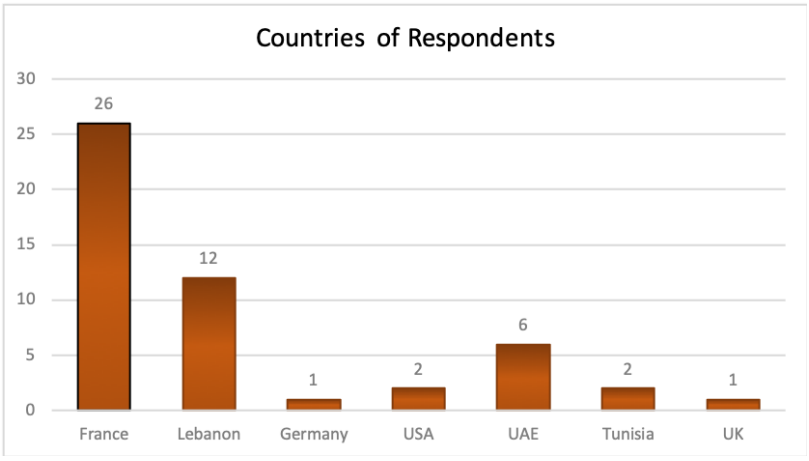

FIGURE 2.27: Countries of Respondents

2.5.2.1 Clarity Results & Discussion

In the form, participants were first asked to guess the correct labels to assign to related concepts according to the described meanings of the concepts. For each concept, a list of 3 to 4 possible choices is provided, where choices are synonyms from several domains. For example, for the user concept, the participant had to choose between the following four labels: person, user, human, and client. Figure 2.28 confirms that the terms used are clear for the multi-domain experts with an average of 90% for guessing correct concept labels. The term "Service Provider" was the most ambiguous, with a 78% correct match percentage, especially for computer scientists where 7 of them (i.e., 28%) chose the term "Supplier".

Next, participants were asked to guess the correct labels of inter-concept relations based on their described meanings. Figure 2.29 shows that labels were correctly matched to their corresponding relations with an average of 88%. Therefore, the terms used to describe the semantic relations are clear for the three categories

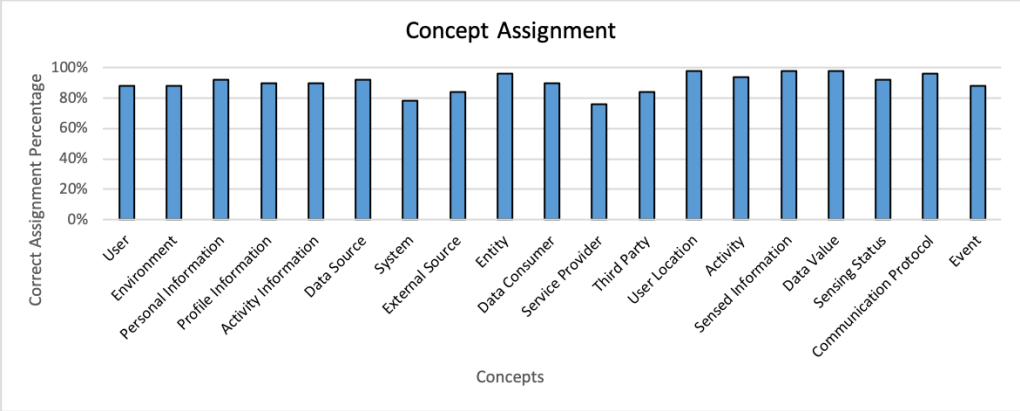

FIGURE 2.28: Concept Evaluation

of experts. The most ambiguous label matching was for relation 16 (i.e., the relation that maps the `ucsn:UserLocation` concept to `mssn:Location`) where 19 of the experts (6 computer scientists and 13 engineers) have considered the opposite relation between the two concepts, i.e., they considered `mssn:Location` as sub-concept of `ucsn:UserLocation`. Knowing that `mssn:Location` is a more general concept that can also be used to describe other location aspects such as the spatial descriptions of the environments and locations of sensors/devices. Consequently, having only a 62% correct match for relation 16 is explained by the fact that the two concepts are syntactically and semantically close, thus linking them without having their descriptions was difficult for the participants. In fact, we only provide the name of the two concepts with a list of possible inter-relations in the relations section of the form.

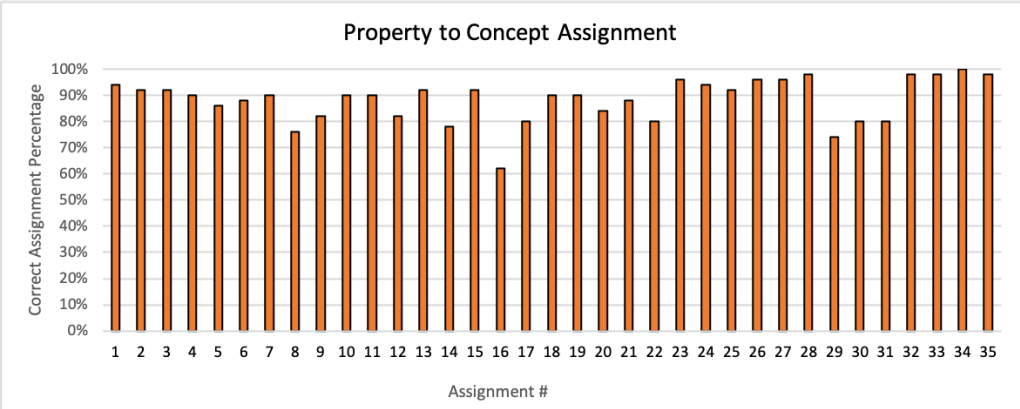

FIGURE 2.29: Property Evaluation

To conclude, the evaluation showed that all concept/relation labels achieve a satisfactory level of clarity based on feedback from multi-cultural stakeholders with different fields of expertise. This reinforces the re-usability of uCSN since it is unambiguous and easily understood.

2.5.3 Performance Evaluation

In order to evaluate the performance of uCSN, we considered several scenarios to study the impact of user, environment and context complexity on performance. Performance tests consist of executing queries related to each scenario (i.e., from the list of previously defined queries) on different population sizes. The performance results take into account the query run-time by running 10 times and calculating the average execution time. The tests were conducted on a machine equipped with an Intel i7 - 2.8 GHz processor, and 16 GB of RAM.

2.5.3.1 User Impact

We studied here the impact of user information on performance. We considered therefore two scenarios, one focusing on user's personal information from a general point of view, and the other targeting the complexity of sensed information in terms of number and associated characteristics.

In the first scenario, we varied the percentage of personal information (divided in half between profile and activity) from 0, 30, 50, 70, to 100% in the population of individuals, while considering the following three population sizes: 100, 1000, and 10 000 context information. Then, we retrieved the list of personal information by combining queries 3 and 4 and measuring the corresponding run-time. In figure 2.30, we noticed that increasing the percentage of personal information in the population of individuals increases the time needed to retrieve it. For example, in the cases of 100 and 1000 context information, the execution time has respectively increased from 10ms (0%) to 30ms (100%), and from 15ms (0%) to 141ms (100%). The progression from 0% to 100% personal information had a quasi-linear impact on query run-time for all three cases (100, 1000, and 10 000 context information). When considering the worst-case scenario of 10 000 personal information describing the user in a single context, the process was able to retrieve it in less than 650ms.

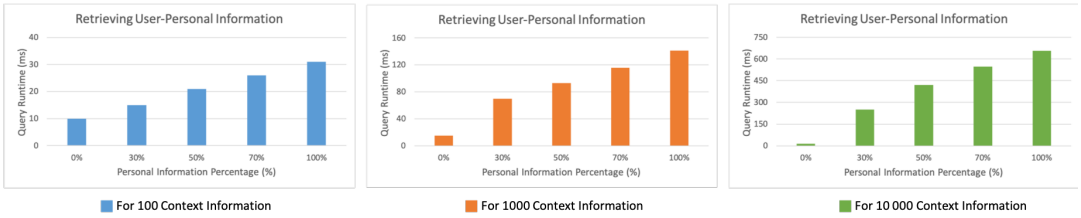

FIGURE 2.30: User Complexity Impact

In the second scenario, we studied the impact of complex sensed information on performance. We considered a two-dimensional complexity, one increasing the number of sensed information from 1, 10, 50, to 100, and the other increasing the number of associated elements per sensed information from 0, 10, 100, to 500. For the associated elements, we considered random partitions between data values, sensing

statuses, data consumers, sensors, and sensing events. We combined and executed queries 1 and 2 for this scenario. Figure 2.31 shows that the process maintained good performance for all complexity cases and the evolution of the query run-time is quasi-linear. When considering the case of 10 sensed information with 100 elements per information (most close to real scenarios), the time required to retrieve it was less than 100ms. The worst case was when having 100 sensed information with 500 elements per information (i.e., 50 000 individuals to retrieve), the process was able to retrieve all these information pieces with an average of 1s.

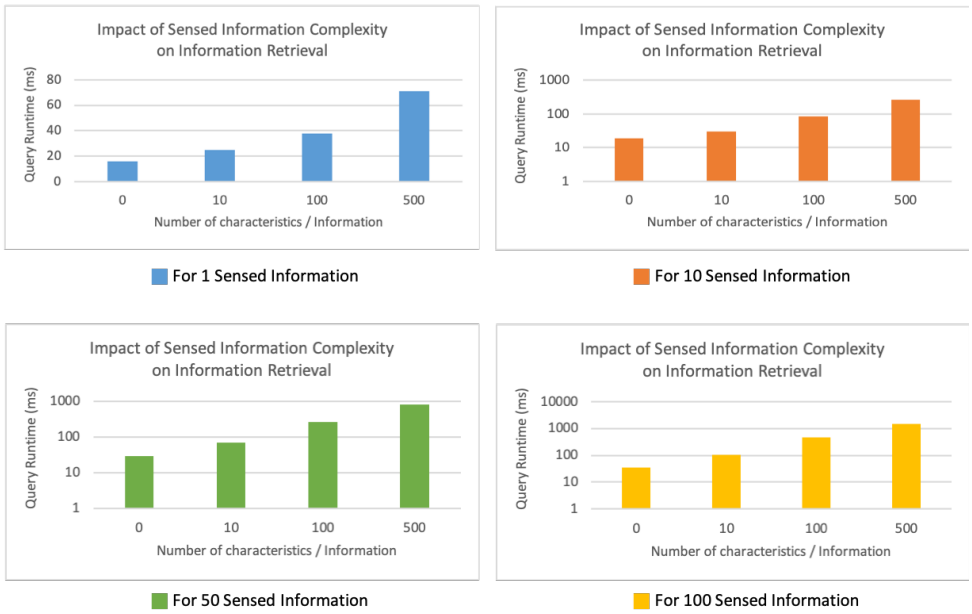

FIGURE 2.31: Impact of Sensed Information Complexity

2.5.3.2 Environment Impact

Here, we checked the impact of having surrounding environments, with various levels of complexity, on performance. To do so, we considered a two-dimensional environment complexity, where the first indicates the number of sub-environments hosted by the environment (e.g., the user's home hosts 5 rooms), and the second states the number of associated individuals per environment/sub-environment. The individuals describe location maps with their associated locations, sensors, actuators, and devices deployed in the environment. We ran query 6 for this scenario and we measured the query run-time for each case. Figure 2.32 shows that increasing the complexity of the environment had a quasi-linear impact on the time required to retrieve the corresponding descriptions and components. Nonetheless, the information retrieval process maintained good performance for all cases, which highlights the ability of the ontology to handle complex environments. If we consider the case of 1 environment with 100 individuals that characterize it (quasi-real scenario), the time required to retrieve the environment characteristics was less than 40ms. The

worst case was that for 100 environments (i.e., 1 environment that contains 99 sub-environments) with 500 describing individuals per environment, the process was able to build a view of all environments in an average of 1s.

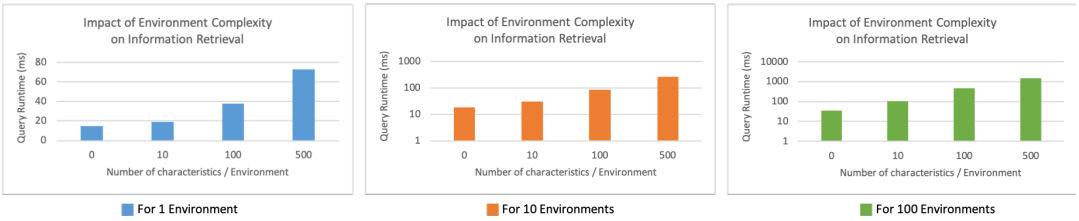

FIGURE 2.32: Environment Complexity Impact

2.5.3.3 Context Impact

Previous tests were conducted by considering controlled scenarios of information partitioning. We aim in this test to create random scenarios and study the impact of various contexts on performance. To do so, we varied the size of the context from 1, 5, 10, 50, 100, 500, 1000, 5000, to 10 000 describing information. Then, we considered for each context size three scenarios of random information partitioning between the user and the environment. We ran query 8 for this test. According to Figure 2.33, the information partitioning per context had no impact on the query run-time. This is due to the fact that all context information are individuals regardless of their semantics, and each modeled individual has at least one associated relation, so the overall number of modeled individuals and relations is closely similar for all three scenarios. Finally, the context size had a quasi-linear impact on the query run-time (cf. Figure 2.33). Nonetheless, the information retrieval process maintained good performance, even for large contexts of 10 000 information, where it was able to retrieve context information within an average time of 523ms.

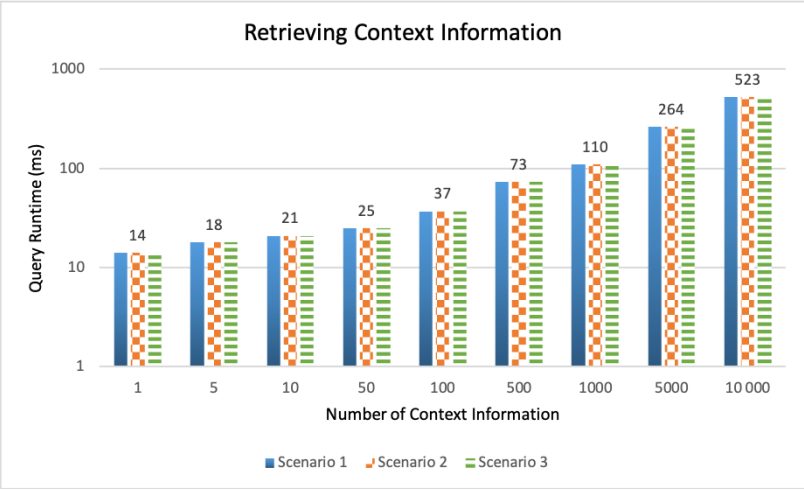

FIGURE 2.33: Context Complexity Impact

Discussion. The performance evaluation showed that adopted uCSN concepts and properties do not heavily impact the query run time, which remains quasi-linear. Moreover, the uCSN ontology can handle various types of contexts, including complex ones from the user or/and the environment perspectives, while maintaining low computational complexity (i.e., in time). This highlights the feasibility of using uCSN for numerous context-aware applications, including those subject to real-time constraints.

2.5.4 Consistency Evaluation

Consistency is defined as a criterion that verifies if the ontology includes or allows any contradictions [100]. The formal and informal descriptions in the ontology must therefore be consistent. In order to evaluate consistency, we adopted the following SPARQL queries that search for anti-patterns in the ontology, a strong indicator of inconsistencies. Query 15 detects concepts with no parent, and Query 16 detects abnormally disjointed concepts in the ontology. Finally, to conclude the inconsistency evaluation, we ran Protege's HermiT 1.3.8.413 reasoner, and found no inconsistencies between the asserted class hierarchy and inferred one.

Query 15: Searching for concepts with no parent

```
SELECT distinct ?a
    WHERE {
              ?a SubClassOf owl:Nothing.
          }
```

Query 16: Searching for abnormally disjointed concepts

```
SELECT distinct ?A ?B1 ?B2 ?C
    WHERE {
              ?B1 subClassOf ?A.
              ?B2 subClassOf ?A.
              ?C subClassOf ?B1.
              ?C disjointWith ?B2.
            }
```

Discussion. The query results show no inconsistencies in the uCSN ontology structure. The only concept subsuming nothing is owl:Nothing (Query 15). Query 16 results indicate that there are no concepts that have abnormal disjoint relations with their relatives. This underlines the soundness of our newly added and imported concepts, and therefore the soundness of the graph structure. This proves critical when considering future alignments between uCSN and other ontologies (e.g., domain-specific user/environment ontologies).

2.6 Summary

Many works adopted ontologies for better semantic representation of user contexts. However these works do not fully address the challenges of information coverage, information diversity, source diversity, and information uncertainty. Moreover, some context models contain domain specific knowledge and are not re-usable for different application purposes. Consequently, we propose in this chapter uCSN, a generic, modular, and extensible ontology for user-Context modeling in Sensor Networks. uCSN provides a high-level coverage of context information by introducing new concepts and properties and importing others from the DPV [1], SOSA/SSN [2], HSSN [43], and W3C Uncertainty [3] ontologies. We implemented uCSN and evaluated its accuracy, clarity, performance, and consistency.

Chapter 3

Privacy Risk Inference

"Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less."
– Marie Curie

With the rapid expansion of smart cyber–physical systems and environments, users are becoming increasingly concerned about their privacy, and asking for more involvement in the control and protection of their data. However, users may not be completely aware of the direct and indirect privacy risks involved with exchanging data with data consumers to properly manage their privacy decisions.

Existing approaches of user privacy risk awareness suffer from several drawbacks, including: (i) no consideration of user contextual knowledge and its impact on data sensitivity, and thus on user privacy; (ii) lack of expressiveness in risk definitions to consider various simple/complex data combinations; (iii) lack of representation and serialization of data that are heterogeneous in terms of types, formats, sources, and semantics, to allow for holistic (all-data-inclusive) risk reasoning; (iv) lack of value-based reasoning; (v) lack of high-level risk detection that encompasses risks of various types and inferences; (vi) lack of an adaptable/user-friendly risk overview; (vii) lack of efficiency, performance-wise, to support the user in various contexts; and (viii) lack of re-usability in different application domains.

To address the aforementioned limitations, we propose in this chapter CaSPI, a context-aware semantic reasoning approach for dynamic privacy risk inference. This approach relies on the use of ontologies and inference rules for contextual knowledge representation and privacy risk definitions with high semantic expressiveness power. The risk inferences are thus achieved by performing rule-based reasoning over modeled context knowledge, which includes sensed data, as well as other background data about the user and her environment, with their relationships. CaSPI is generic and re-usable in different domains. Performance results showed that it provides scalability and computational and storage efficiency, making it able to assist the user in different contexts, including ephemeral ones.

3.1 Introduction

Advances in mobile and ubiquitous computing, such as the Internet of Things (IoT), have reshaped the lives of people over the last few years. Current applications of smart IoT-enabled cyber–physical systems touch almost all aspects of our daily life, including healthcare (e.g., patient and elderly monitoring), entertainment and leisure (e.g., smart entertainment spaces), transportation (e.g., vehicle networks, smart highways), environmental (e.g., energy management, pollution level monitoring), and work (smart manufacturing and work environments).

While such systems promise to ease our lives, they raise major privacy concerns for their users, as the data they collect is often privacy-sensitive, such as location of individuals, patients' vital signs, accelerometer data, and energy consumption data of environments (e.g., building, home). In fact, the processing and analysis of collected sensor data can lead to reveal a wealth of sensitive information about the user, such as her routines and habits, health conditions, political/religious affiliations, preferences, activities performed (general activities like running and driving, or domain-related activities like hand moving in the medical domain), and so forth [101], [102], [103], [104]. This can cause serious harms for the user (e.g., mental, physical, dignity/reputation, financial, and societal harms) if the sensed data or disclosed information were misused by the providers of these systems, or even sold to interested third parties (with/without user consent) and exploited for various purposes.

Many studies (e.g., [105], [106]) showed that users are becoming more and more conscious of their privacy and willing to play an active role in controlling their data. This fact was also backed by the newly released privacy regulations (e.g., GDPR [11], CCPA [13]) and standards (e.g., PbD [21], ISO/IEC 27701 [24]), which call for more involvement of users in the control and protection of their data by enabling them to control what is collected, when, by whom, and for what purposes. Some works (e.g., [31]–[34]) tried to deal with this requirement by enabling users to specify their privacy preferences and accept privacy policies that enforce these preferences. However, the user may not be aware of the privacy risks associated with her data sharing to correctly specify her preferences in the first place. She may simply not know what can be inferred from her data when data bits and pieces are analyzed in isolation, or combined with each other and/or with other contextual data about the user or her surrounding environment, that could be acquired from external sources (e.g., social network profiles, public websites and databases).

To overcome this issue, several approaches [107]–[111] have been proposed to raise user privacy risk awareness in the fields of connected and web environments. However, these approaches share the following limitations: (i) no consideration of user contextual knowledge and its impact on data sensitivity, and thus on the plausible inferences of sensitive information; (ii) lack of expressiveness in risk definitions

to consider different combinations of sensed data together and/or with other contextual data, and their implications for the disclosure of sensitive information; (iii) lack of representation and serialization of heterogeneous data in terms of types, formats, sources, and semantics, to allow for holistic (all-data-inclusive) risk reasoning; (iv) lack of value-based reasoning, the approaches are limited to general reasoning over attributes' names (e.g., location, blood pressure) and do not support reasoning over attributes' data values, constraints, and properties; (v) lack of high-level risk detection that encompasses risks of various types (i.e., direct and indirect risks) and sensitive information inferences (e.g., information describing the user profile and activity); (vi) lack of adapted risk overview to user expertise and interests; (vii) lack of efficiency, performance-wise, to support the user in various contexts; and (viii) lack of genericity and re-usability in different application domains.

Accordingly, we propose in this chapter a context-aware semantic reasoning approach for dynamic privacy risk inference, entitled CaSPI. This approach is capable of providing the user with a complete overview of the direct and indirect privacy risks that she accepts to take. Direct risks are those associated with the sharing of sensed data and that the user can control. Indirect risks are those imposed by the surrounding connected environment and on which the user has no control, such as being under a CCTV surveillance in a monitored area (e.g., airport, mall). To achieve this, CaSPI relies on the use of ontologies and inference rules to respectively represent the user contexts and define the risks to be detected with high semantic expressiveness power. For data representation, an ontology-based data model enables the representation of heterogeneous data (e.g., scalar and multimedia data) that could be acquired from different types of data sources (e.g., sensors, devices, social network profiles, public databases, documents). It also allows to represent the semantics of the relationships between data. For the risk definition, CaSPI features a generic semantic rule syntax for explicitly defining various types of risks, and considering diverse data combinations using basic and advanced operators (e.g., logical, spatio-temporal, semantic operators). For the risk inferences, CaSPI incorporates a semantic reasoner that performs rule-based reasoning over modeled context information in order to infer the privacy risks involved in the current situation of the user. In addition, it monitors the evolution of risks to cope with the dynamicity of the user context. CaSPI is generic and re-usable in numerous application domains. To validate our proposal, we developed a prototype based on Semantic Web tools (e.g., OWL API, SWRL API, Pellet reasoner), and illustrated its functioning from both back-end and front-end perspectives. We also evaluated its performance in different scenarios. Our results showed that CaSPI delivers scalability and efficiency in time and space, which makes it able to assist the user in various contexts, including ephemeral ones (i.e., contexts with short time periods).

The remainder of this chapter is organized as follows. Section 3.2 presents a scenario that motivates our proposal and identifies the challenges to address. Section

3.3 evaluates existing approaches. Section 3.4 details the different modules of CaSPI and provides formal definitions of the key terms. Section 3.5 presents the implementation and experimental protocol. Finally, Section 3.6 summarizes the chapter.

3.2 Motivating Scenario

In order to motivate our proposal, consider the scenario presented in Chapter 1 that illustrates the situation of Alice who shares her energy-consumption and location data respectively with the electricity and healthcare providers. Alice is a COPD patient and follows her medical treatment from home using a NIV device (medical device). Figure 3.1 details the data/service exchange processes and the additional background data known about Alice in her current situation. Background data includes her marital status, profile picture, and date of birth collected from her Facebook profile. The date of birth is also captured in a different format from the public voting records.

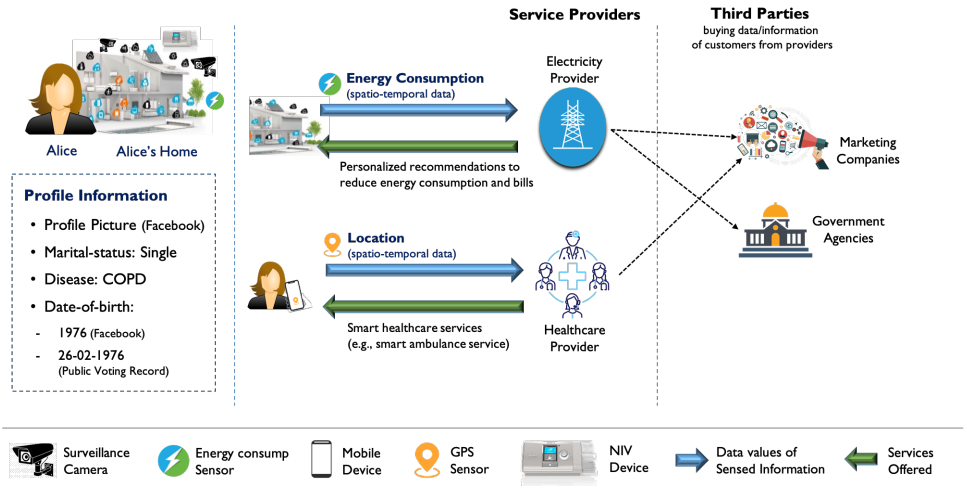

FIGURE 3.1: Running Example

Even though Alice is notified, through agreed policies, of data consumers who have access to her data (i.e., service providers and third parties), she may not necessarily be aware of the privacy risks associated with this sharing. For instance, analyzing the energy consumption signatures of her home (cf. Figure 3.2 can lead to disclose numerous sensitive information about Alice related to her lifestyle and habits [36], including her presence/absence hours at home, waking/sleeping cycles, and activities performed at home with their time duration (e.g., cooking, watching TV, workouts using the treadmill). In addition, existing works (e.g., [37]) showed that consumption signatures can be analyzed to identify the use of specific appliances and devices. This would reveal the disease of Alice if the use of her NIV machine was identified. The analysis of location data patterns (cf. Figure 3.2) can also entail significant risks for Alice such as the risks of disclosing her habits and routines, behaviors, health conditions based on frequent visits to hospitals, political/religious

affiliations, and her identity based on her locations in personal environments (e.g., home, office) [102], [112].

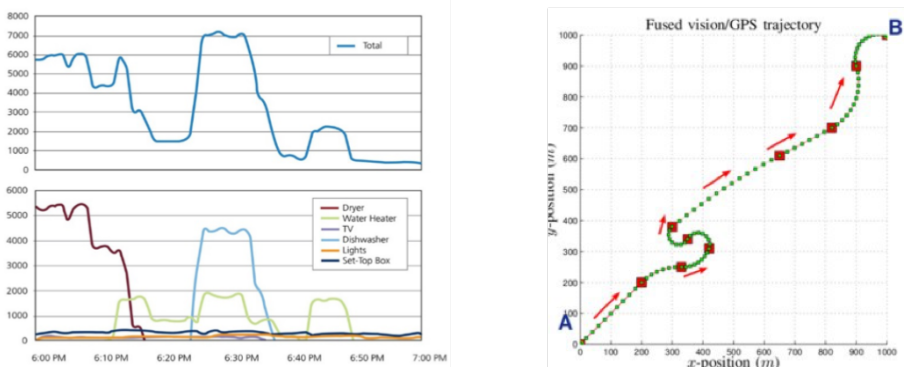

FIGURE 3.2: Energy consumption signature / Location data pattern

As well, data consumers can exchange customer data between each others (cf. Figure 1.4), and collect other background data about them from external data sources, like social network profiles or public databases, to enrich customer profiles with new insights and make it more valuable. However, this can expose users to other more complex privacy risks. For example, assume that Alice has unlawfully certified that she is living alone to be eligible for a welfare program when submitting her application. The parties that have access to both location and consumption data (e.g., marketing company) can infer this fraud (it is enough to identify the use of particular smart appliances, such as microwave, television, or coffee maker, while Alice is outside her home). Also, if Alice stated in her application that she is a teenager, collecting her date of birth from her Facebook profile or from the public voting records leads to infer the fraud.

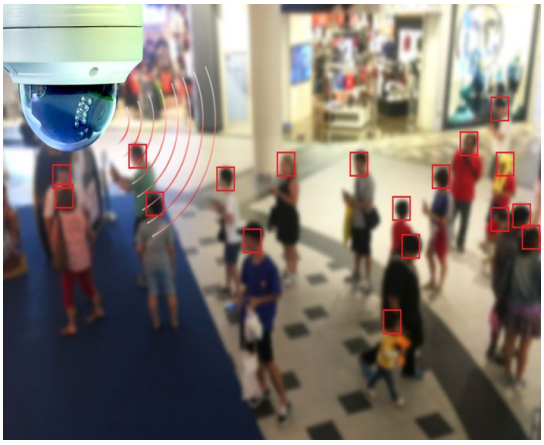

FIGURE 3.3: CCTV surveillance in a smart mall

All the previous risks were direct risks that can be controlled by Alice as they are associated with her sensed data. However, Alice can be also exposed to other risks that are out of her control zone (i.e., indirect risks). For example, as shown in Figure 3.3, being under CCTV surveillance in a monitored area (e.g., malls, streets, airports)

can lead to reveal Alice's presence in the area, her interests, activities, meetings, social relationships, and so on.

All of this highlights the need to inform Alice of the direct and indirect risks that she accepts to take, with or without her consent, in order to enable her to make informed privacy decisions. However, when considering various types of risks, data heterogeneity (e.g., scalar, multimedia) and semantics, user expertise, and the time constraints of user decisions, several challenges emerge. The challenges related to data heterogeneity and semantics were detailed and addressed in the previous chapter (cf. Sections 2.2 and 2.4). We focus here on the challenges related to risk definition, risk detection and monitoring, user expertise, and solution efficiency:

Challenge 1. ***Increasing expressiveness in risk definitions*****:** Contextual data (i.e., sensed and background data) can be processed and/or combined in many simple and complex forms, yielding a variety of sensitive information inferences for the user. The approach should therefore feature a generic syntax that allows the definition of privacy risks with high expressiveness power. This syntax must consider various types of data combinations, including combinations using: (i) basic operators, such as logical and comparison operators; and (ii) advanced operators like spatio-temporal operators that examine the spatio-temporal correlations among data elements, and semantic operators that accurately reflect the semantics of the relationships between them. In addition, it must consider different types of sensitive information that could be revealed about the user, such as information describing the user profile (e.g., age, marital-status, disease) and activity (e.g., habit, behavior).

Challenge 2. ***Performing a holistic (all-data-inclusive) risk reasoning*****:** As discussed above, sensed data can be analyzed in isolation, or combined with each other (e.g., energy-consumption and location data) and/or with other background data acquired from external sources (e.g., social network profiles, public databases, documents). This improves the inference capability of data consumers, thereby enlarging the sphere of privacy risks. Consequently, the proposed solution must be capable of representing and serializing data that is heterogeneous in terms of type (e.g., scalar and multimedia data), metadata, and format. This paves the way for holistic reasoning over contextual data, including attribute-based and value-based reasoning.

Challenge 3. ***Coping with the dynamicity and context-dependency of privacy risks*****:** The sensitivity of data may depend on the user context [28], [35]. For example, the sensitivity of Alice's location when she is in a hospital is higher than at home, as in this case location data could be used to infer the disease of Alice. Consequently, as context changes, new privacy risks may emerge, while others may disappear or lose in significance. Therefore, the proposed solution should keep track of context changes, analyze their impacts on privacy risks, and maintain an updated risk overview.

Challenge 4. ***Dealing with user expertise*****:** People may have different levels of expertise when it comes to specifying their preferences and understanding their privacy risks. The proposed solution should be able to tailor the preference specifications and risk overview to the user's expertise. This ensures good quality of human-machine interactions and allows all users to understand the privacy implications.

Challenge 5. ***Delivering scalability and efficiency*****:** The solution must be scalable, i.e. handles reasoning over an increasing number of contextual data and privacy risks. It should also react fast to support the user in different contexts, especially as user decisions must sometimes be made in real-time. Finally, it should maintain low computational and storage complexity, which makes it operational on various types of devices, including those with limited resources.

Other challenges may also arise when considering privacy risk inferences, however, we focus in our research work on tackling the aforementioned ones.

3.3 Related-work

Several works were proposed in the fields of Connected environments and Web environments to tackle the challenges of user privacy risk awareness.

- Christin et al. [107] proposed a graphical-based warning approach to inform users about the risks of disclosing sensitive information about them in participatory sensing applications. They considered four types of attributes that can be shared by users: location, pictures, audio samples, and acceleration. They defined three levels of granularity (i.e., fine, medium, coarse) for each of these attributes, and associated a list of possible risks to each level. The approach does not include a risk detection mechanism; instead, it discusses some of the risks that might be associated with data sharing. The risks are represented through picture-based warnings, making them easily comprehensible by users. The approach does not cover the risks that may be generated when attributes are combined together and/or with other contextual information. The authors have tested their alerting approach by conducting a user study involving 30 participants.

- Wagner et al. [108] provided a user-centric approach to create privacy-aware user interfaces for mobile eHealth applications. They focused on three classes of applications: fitness trackers, personal well-being applications, and medical applications. The approach features an Inform-Alert-Mitigate (I-AM) cycle that (i) informs users of potential privacy issues for each of these applications regarding privacy policies and permissions; (ii) considers the user's contextual information and alerts her about privacy risks that she exposes herself to; and (iii) provides the user with concrete actions that can mitigate the current risk-exposure. It considers the following attributes that could be collected by apps: identifiable (e.g., name, email), demographic (e.g., date of birth, gender), activity (e.g., location,

time), nutrition, and medical attributes (e.g., heart rate). However, the authors only provided an overview of the I-AM cycle phases without going into detail about the process used to determine the risks involved for the user.

- Alrayes et al. [109] examined the user awareness regarding the privacy risks associated with location data sharing in Geo-Social Networks (GeoSNs). They carried out user studies, in form of online surveys, to gauge users' perceptions to privacy threats as a consequence of recording their location information of GeoSNs. They addressed the following aspects in these studies: (i) the extent of users' awareness of the terms of use they agree to when using these applications; (2) their understanding and attitude to potential privacy implications; and (3) how they may wish to control access to their personal information on these applications. For the privacy implications, the questions focused on evaluating users' awareness of plausible inferences about their private places, activities at different times, their connections to other users, and possible knowledge of this information by the applications. The approach does not feature a risk inference process, it only provides some recommendations for the design of privacy-sensitive GeoSNs.

- Alemany et al. [110] proposed two soft-paternalism mechanisms in the form of nudges that provide information to the user about the privacy risk of publishing information on Social Networks. Privacy awareness refers to the users' knowledge about the potential audience that might see their publication disclosure. The first mechanism shows the profile images of users that are part of the potential audience that may have access to the message with a risk-level alert (e.g., low, high). The second mechanism shows the number of users that are part of the audience that may have access to the message. The two mechanisms were tested with 42 teenagers in an online social network called PESEDIA. The results obtained suggest that the use of soft-paternalism mechanisms could be a suitable option to improve the decision-making process and prevent teenagers from privacy risk publications that could have negative consequences.

- Petkos et al. [111] proposed a scoring framework for raising user awareness in Online Social Networks (OSN) regarding the sensitive information that could be disclosed by OSN operators and other third parties that can access their data. They identified eight categories of sensitive information, including demographics, psychological traits, sexual profiles, political attitudes, religious beliefs, health factors, location, and consumer profile. The data to be shared in OSN includes posted content (text, images), explicitly declared profile data, user network data, sets of likes, and so forth. The approach introduces a privacy scoring mechanism that enriches the sensitive information with several scores, each reflecting a different aspect of information disclosure. The overall privacy score is obtained by multiplying the scores of the following aspects: (i) level of confidence to infer the information; (ii) sensitivity of the disclosed information for the user; and (iii) visibility of the information to other people. Other aspects were also considered to

enrich the scoring model, including (1) the information source (i.e., declared by the user or inferred); (2) the associated data in case of inferred information; and (3) the level of control of the user on the information disclosure. The authors did not conduct experiments to showcase the applicability and effectiveness of their scoring approach. Their ongoing work focuses on developing a risk inference mechanism to complete the computational aspects of their framework.

3.3.1 Comparative Study

In order to compare existing works, we define the following criteria based on the challenges and needs discussed in Section 3.2. The criteria focus mainly on application domains, risk definition, data representation, risk inference, user expertise, and operational features of the approach. These are:

- **Criterion 1.** *Application Domain:* states the domain of application of the proposed approach (e.g., connected environments, social networks).

For the risk definition, we specify criteria related to the scope of attributes considered and the expressiveness of risks:

- **Criterion 2.** *Attribute Scope*. Checks information coverage when defining risks:

  (**2.1**) *Attribute Diversity*: denotes $\{YES, NO\}$ if the approach considers privacy risks related to numerous/diverse attributes of the user.

  (**2.2**) *Context Coverage*: indicates $\{YES, NO\}$ if the approach considers the user's contextual information when defining risks.

- **Criterion 3.** *Expressiveness*. Checks the ability of the approach to provide expressive risk definitions, i.e. to consider different/complex data combinations of various attributes together and/or with other contextual data when defining risks. This helps to broaden the coverage and improve the expressiveness of the risks associated with data sharing. We look at data combinations using basic and advanced operators:

  (**3.1**) *Basic Operators*: states $\{YES, NO\}$ if the approach considers basic operators (e.g., Logical operators) to combine data.

  (**3.2**) *Advanced Operators*: states $\{YES, NO\}$ if the approach supports advanced operators (e.g., semantic, spatial, temporal operators) to combine data.

For the data representation, we specify criteria related to data heterogeneity and data serialisation:

- **Criterion 4.** *Data Heterogeneity*: indicates $\{YES, NO\}$ if the approach supports representation of heterogeneous data in terms of data types and metadata (i.e., scalar and multimedia data).

- **Criterion 5.** *Data Serialisation*: indicates $\{YES, NO\}$ if the approach supports: (i) data extraction from diverse data sources (e.g., sensors, social network profiles, documents), having different serialization formats (e.g., json, rdf, pdf, docx, jpeg, mov); and (ii) data representation in a unified serialization format, allowing consequently the reasoning on heterogeneous data.

For the risk inference, we specify criteria related to the nature and level of control of the risks inferred, and the handling of attribute/value-based risk reasoning:

- **Criterion 6.** *Risk Nature*. Checks the nature of the risks inferred:

  (**6.1**) *Identity Disclosure*: denotes $\{YES, NO\}$ if the approach identifies risks related to the disclosure of the user's identity (i.e., risks of re-identification).

  (**6.2**) *Sensitive Information Disclosure*: states $\{YES, NO, PARTIAL\}$ if the approach identifies risks related to (i) the disclosure of the user-shared sensitive data by unwanted parties, and (ii) the disclosure of other SPI (cf. Section 1.2.1) about the user when processing her shared data.

- **Criterion 7.** *Risk Control*. Checks the ability of the approach to infer direct and indirect risks for the user:

  (**7.1**) *Direct Risks*: denotes $\{YES, NO\}$ if the approach identifies direct risks that can be controlled by the user, such as those associated with her data sharing.

  (**7.2**) *Indirect Risks*: denotes $\{YES, NO\}$ if the approach identifies indirect risks, i.e. risks related to the user but over which the user has no control (e.g., the risks imposed by the surrounding connected environment).

- **Criterion 8.** *Reasoning*. Checks the ability of the approach to handle attribute-based or/and value-based risk reasoning:

  (**8.1**) *Attribute-based*: indicates $\{YES, NO\}$ if the approach performs risk reasoning on attribute names and/or properties.

  (**8.2**) *Value-based*: indicates $\{YES, NO\}$ if the approach performs risk reasoning on data values of attributes (e.g., examines conditions applied to data values and/or properties).

In addition to the aforementioned criteria, we define other criteria related to the risk indicators delivered to the user and the operational features of the approach:

- **Criterion 9.** *User Expertise*: denotes $\{YES, NO\}$ if the risk overview provided by the approach is tailored to the user's level of expertise.

- **Criterion 10.** *Operational Features*:

  (**10.1**) *Automation Degree*: denotes $\{AUTO, SEMI, MANUAL\}$ if the approach is respectively automatic, semi-automatic, or manual.

(**10.2**) *Performance*. We consider two sub-criteria:

(i) *Scalability*: denotes $\{YES, NO\}$ if the approach is capable of performing over an increasing number of information and risks;

(ii) *Efficiency*: states $\{YES, NO\}$ if the approach provides appropriate performance in terms of time behavior, resource utilization, and capacity in various scenarios.

Criteria			Christin et al. [107]	Wagner et al. [108]	Alrayes et al. [109]	Alemany et al. [110]	Petkos et al. [111]	**CaSPI [4]**
Application Domain			Connected Environments	eHealth Technologies	Geo-Social Networks	Social Networks	Social Networks	**Generic**
Risk Definition	*Attribute Scope*	*Attribute Diversity*	YES	YES	NO	YES	YES	**YES**
		Context Coverage	NO	YES	NO	NO	NO	**YES**
	Expressiveness	*Basic Operators*	NO	NO	NO	NO	NO	**YES**
		Advanced Operators	NO	NO	NO	NO	NO	**YES**
Data Representation	*Data Heterogeneity*		YES	NO	NO	NO	NO	**YES**
	Data Serialisation		NO	NO	NO	NO	NO	**YES**
Risk Inference	*Risk Nature*	*Identity Disclosure*	NO	YES	NO	NO	NO	**YES**
		SPI Disclosure	YES	PARTIAL	YES	PARTIAL	YES	**YES**
	Risk Control	*Direct Risks*	YES	YES	YES	YES	YES	**YES**
		Indirect Risks	NO	NO	NO	NO	NO	**YES**
	Reasoning	*Attribute-based*	YES	-	YES	YES	YES	**YES**
		Value-based	NO	-	NO	NO	NO	**YES**
User Expertise			PARTIAL	-	NO	PARTIAL	NO	**YES**
Operational Features	*Automation Degree*		AUTO	-	AUTO	AUTO	AUTO	**AUTO**
	Performance	*Scalability*	-	-	-	-	-	**YES**
		Efficiency	-	-	-	-	-	**YES**

[1] - means that the referenced work did not approach this aspect.

TABLE 3.1: Review of Privacy Risk Inference Approaches

Discussion. All of the proposed approaches contribute to raising user awareness of the privacy risks associated with sharing their attributes. However, they all suffer from the same limitations. They lack re-usability as they focus on inferring risks related to specific user information in particular domains. The majority of them (i.e., [107], [109]–[111]) do not consider the impact of user contexts on plausible inferences of sensitive information, despite the fact that data sensitivity varies across contexts, and thus the sphere of risks (cf. Challenge 4). They mainly focus on identifying the risks associated with processing each attribute separately, without considering the risks involved in combining and processing attributes' data together and/or with other contextual data. Only [107] addresses data heterogeneity from the perspective of considering risks associated with location data, pictures, audio

samples, and acceleration data, without tackling the challenges of data representation. When it comes to risk nature, existing works mainly identify risks related to disclosing user-shared attributes by unwanted audience [108], [110], and revealing some of the user-related SPI when executing inference mechanisms on shared data [107], [109], [111]. They consequently cannot be used to identify the risks of user re-identification through protected data patterns (e.g., anonymized/pseudonymized data) when data is processed and sometimes combined with other side information. Moreover, none of them is capable of identifying the indirect privacy risks for the user, or performing a value-based risk reasoning. When considering the risk indicators delivered to the user, [107] and [110] are capable of delivering simple/user-friendly indicators that could be comprehensible by all users. However, none of the works is capable of tailoring the risk overview based on the user's expertise. Indeed, expert/advanced users may require a more detailed view of their privacy implications than non-savvy/beginner users, who prefer a summary of the main risks. Finally, none of these works includes experiments that evaluate the scalability and efficiency of the proposed solution in various scenarios. Table 3.1 details the comparison between the aforementioned works according to the defined criteria.

3.4 CaSPI Proposal

In this section, we detail our proposed risk inference approach, entitled **CaSPI**, which stands for **C**ontext-**a**ware **S**emantic reasoning approach for dynamic **P**rivacy risk **I**nference. CaSPI addresses all the needs and challenges specified in Section 3.2. We start by describing the approach's functioning. We present after formal definitions of the key terms used in this study, and discuss the user profiles with their characteristics. Then, we detail the proposed rule syntax to define risks and the reasoning algorithm of our approach.

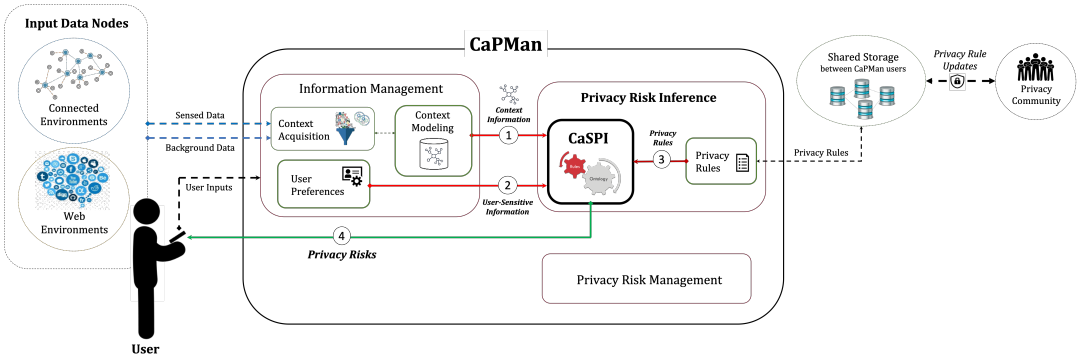

FIGURE 3.4: Overview of the CaSPI proposal

Figure 3.4 illustrates an overview of the solution. CaSPI receives as input: (1) the modeled contextual data providing information about the current user situation, including sensed and background data; (2) the list of user-sensitive information,

which can be personalized by the user depending on her profile; and (3) the list of inference rules expressing the privacy risks to be detected. It launches consequently the rule-based reasoner over modeled data/information in order to infer the direct/indirect risks involved in the current context. The risks inferred are delivered as output (4) through an adaptive user interface according to the specified user profile and preferences. This allows all users, with different levels of expertise, to understand the privacy implications of sharing their data with data consumers (i.e., service providers and third parties) and their presence in the environment (cf. Challenge 5).

When a context change occurs (cf. Definition 4), the system computes the similarity between the current and historical contexts. If a full similarity is detected (cf. Definition 4.1), the user is given the option of re-applying the actions of the previous similar context, or launching the risk reasoner. This contributes to reducing the computational cost of the global CaPMan system.

3.4.1 Context Representation

The quality of the risk inferences depends on the quality of the context coverage (i.e., coverage of data/information that characterizes the user's situation). The approach must consequently enable the collection and representation of various attributes' data describing the user and her surrounding environment, as well as the semantics of the relationships between them, in order to provide a high and expressive context coverage. The data collected is heterogeneous in terms of types and metadata (e.g., scalar and multimedia data), and can be of two categories: (1) *sensed data* by a wearable or deployed sensor in the user's environment, that is shared with data consumers in exchange for services; and (2) *background data*, additional data that describes the user (e.g., profile data, activities) and/or her environment, such as the environment's structure, description, or a hosted component like sensor, appliance, etc. The data can therefore be acquired from various nodes (i.e., data sources) in the user's connected and web environments, including sensors, devices, social network profiles, public databases, and so on. However, data of these sources do not follow a common structure and are serialized in different formats (e.g., json, rdf, xml, pdf, docx, xlsx, png, jpeg, mov, mp4). Consequently, the challenges of data extraction and representation are handled respectively by the *context acquisition* and *context modeling* modules of the CaPMan framework. In what follows, we formally define a *data node*, *physical environment*, *user context*, *context similarity*, *attribute*, *sensed attribute*, and a *background-oriented attribute*.

Let $\boldsymbol{u}$ denotes the ***user of interest***.

Definition 1 (Data Node)**.** Let $\boldsymbol{DN}$ be the set of ***data nodes*** $\{\boldsymbol{dn_1}\ ;\ \ldots\ ;\ \boldsymbol{dn_n}\}$ that could be: (i) sources from which data is collected (e.g., sensors, devices, social networks); (ii) smart systems deployed in the user's environment (e.g., actuators, appliances); or (iii) data consumers with whom the data is shared (e.g., service providers,

third parties). $dn \in DN$ is formalized as follows:

$$dn : \langle\ desc\ ;\ id\ \rangle\ \text{, where:}$$

- $desc$ is the textual description of dn (e.g., GPS-sensor, Healthcare-provider).
- id is the identity of dn, expressed as a uniform resource identifier (URI). ■

Example 1. The sources of the data collected, describing Alice and her home, are the GPS sensor, energy-consump sensor, Facebook profile, and the public database of voting records:

- `sensor-1:⟨ GPS ; http://46.89.1.47:80/ ⟩`
- `sensor-2:⟨ Energy consumption ; http://46.193.0.164:80/ ⟩`
- `socialAccount-1:⟨ Facebook ; https://www.facebook.com/Alice ⟩`
- `database-1:⟨ Voting records ; https://publicvoting/records.html ⟩`

Example 2. The data consumers with whom Alice shares her sensed data are the electricity and healthcare providers:

- `provider-1:⟨ Electricity provider ; http://58.17.37.23:1751/ ⟩`
- `provider-2:⟨ Healthcare provider ; http://64.31.3.12:5051/ ⟩`

Definition 2 (Physical Environment). Let E_u be the set of ***physical environments*** $\{env_1\ ;\ ...\ ;\ env_n\}$ where the user u is/was located. $env \in E_u$ can be of two types: connected (i.e., hosts smart systems) or unconnected environment.

$$\forall\ env\ \in E_u\ ,\ \ env : \langle\ desc\ ;\ sz\ ;\ Sys\ \rangle\ \text{, where:}$$

- $desc$ denotes the textual description of env (e.g., home, office, mall, street)
- sz expresses the *spatial zone* of env (cf. Definition 3)
- $Sys \sqsubseteq DN$ represents the set of systems (*data nodes*) deployed in env (e.g., sensors, actuators). For unconnected environments, $Sys = \varnothing$. ■

Definition 3 (Spatial Zone)**.** A ***spatial zone***, sz, is defined as a geographical surface bounded by a set of distinct locations, where each is expressed by coordinates in space, such that:

$$sz : \langle\ loc_1\ ;\ loc_2\ ;\ ...\ ;\ loc_n\ \rangle\ \text{, where:}$$

- loc is a ***location***, defined as 3-tuple $loc : \langle\ long\ ;\ lat\ ;\ alt\ \rangle$, where $long$, lat, and alt denote respectively the longitude, latitude, and altitude of loc. ■

Example 3. The home of Alice can be represented as follows:

- environment-1:⟨ Home ; zone-1 ; Sys-1 ⟩
- zone-1:⟨ loc-1 ; loc-2 ; loc-3 ; loc-4 ⟩
    - loc-1:⟨ -1.53245 ; 34.0132 ; 200.03 ⟩
    - loc-2:⟨ -1.53310 ; 34.0140 ; 205.14 ⟩
    - loc-3:⟨ -1.51025 ; 34.0581 ; 216.57 ⟩
    - loc-4:⟨ -1.51090 ; 34.0571 ; 218.13 ⟩
- Sys-1={sensor-2 ; device-1}
    - device-1:⟨ NIV medical device ; http://64.17.15.2:5051/ ⟩

Definition 4 (User Context)**.** A ***user context***, $c \in C$, is a spatio-temporal semantic context, defined as the finite group of information characterizing the user's situation within the respective space-time. Specifically, $c \in C$ consists of: (i) the set of attributes' data and properties collected/known about the user, u, and her surrounding environment, $env \in E_u$, within the relevant space-time; and (ii) the set of semantic relations expressing how these attributes are linked within the relevant space-time. $c \in C$ is formalized as follows:

$$c : \langle\ t\ ;\ s\ ;\ A\ ;\ Rel_A\ \rangle \quad \text{, where:}$$

- t denotes the time period of c, defined as 2-tuple $t : \langle\ t_{start}\ ;\ t_{end}\ \rangle$, where t_{start} and t_{end} are two time instants.
- s expresses the *spatial zone* of c (cf. Definition 3).
- $A = \{a_1\ ;\ a_2\ ;\ \dots\ ;\ a_n\}$ is the set of *attributes* characterizing c (cf. Definition 5), where each includes data that is either collected within $\{t; s\}$ or previously collected but still valid within $\{t; s\}$.
- $Rel_A = \{rel_1\ ;\ rel_2\ ;\ \dots\ ;\ rel_m\}$ represents the set of semantic relations between attributes $\{a_1, ..., a_n\} \in A$, such that:

$$\forall\ rel \in Rel_A,\ \ rel \equiv rel(a_i\ ;\ a_j)\ \mid\ \{a_i\ ;\ a_j\} \sqsubseteq A \text{ and } i, j \in [1, n]$$

  - $rel(a_i\ ;\ a_j)$ is a unidirectional relation, with a primitive type of String, that specifies how attribute a_i is linked to attribute a_j.

A context change means a change in the user's situation. It occurs if at least one of the context parameters varies. ■

Definition 4.1 (Context Similarity). Let c_i, c_j be two user contexts, such that $\{c_i; c_j\} \sqsubseteq C$. The similarity between c_i and c_j is determined by computing the similarity between their groups information (i.e., $c.A \sqcup c.Rel_A$):

$$sim_c(c_i, c_j) = sim_c(\ c_i.A \sqcup c_i.Rel_A\ ;\ c_j.A \sqcup c_j.Rel_A\ ) \rightarrow [0; 1]$$

Where:

- $\boldsymbol{sim}$ is a unit similarity function that compares the exact match between attribute descriptions and described entities, as well as the exact match between the sets of semantic relations. $\boldsymbol{sim}$ returns a value ranging from 0 to 1, where 0 means that the two contexts are not similar and 1 means a full similarity.

$$\boldsymbol{sim}(c_i, c_j) = \mathbf{1} \text{ only if:}$$

$$\begin{cases} \forall a_x \in c_i.A,\ \exists!\ a_y \in c_j.A : a_x.desc = a_y.desc \text{ and } a_x.ent = a_y.ent \\ c_i.Rel_A = c_j.Rel_A \end{cases} \blacksquare$$

Definition 5 (Attribute)**.** Let $\boldsymbol{A}$ be the set of $\boldsymbol{attributes}$ $\{a_1\ ;\ a_2\ ;\ \ldots\ ;\ a_n\}$ describing the user $\boldsymbol{u}$ and her physical environments $\sum env \in E_u$. An attribute $\boldsymbol{a} \in \boldsymbol{A}$ is formalized as follows:

$$a : \langle\ desc\ ;\ ent\ ;\ Log\ ;\ access\ \rangle \text{ , where:}$$

- $\boldsymbol{desc}$ denotes the textual description of $\boldsymbol{a}$ (e.g., location data, energy-consump data, user activities, profile images, home appliances).
- $\boldsymbol{ent} \in \{u\} \cup E_u$ denotes the entity related to $\boldsymbol{a}$, which can be the user $\boldsymbol{u}$ or an environment $\boldsymbol{env} \in \boldsymbol{E_u}$.
- $\boldsymbol{Log} = \{\langle\ d\ ;\ M\ \rangle\}$ is the set of spatio-temporal data values of $\boldsymbol{a}$. $\boldsymbol{Log}$ can be viewed as the log file of $\boldsymbol{a}$, where:
  - $\boldsymbol{d}$ denotes the data value, which can be scalar (e.g., location, temperature, age, marital-status) or multimedia (e.g., image, audio, video).
  - $\boldsymbol{M} = \{meta_1\ ;\ \ldots\ ;\ meta_n\}$ is the set of metadata characterizing $\boldsymbol{d}$. For instance, $\boldsymbol{M}$ can include the following metadata:
    - $t_{capture}$, denotes the time of capture of $\boldsymbol{d}$.
    - $l_{capture}$, denotes the location of capture of $\boldsymbol{d}$.
    - $\boldsymbol{source} \in \boldsymbol{DN}$, denotes the data source from which $\boldsymbol{d}$ is captured. $\boldsymbol{source}$ can derive from connected environments (e.g., sensor, device) or web environments (e.g., social media platform, public database).
    - $D_{consumer} \sqsubseteq DN$, represents the set of data consumers with whom $\boldsymbol{d}$ is shared (e.g., service providers, third parties), such that:

$$D_{consumer} = \{\ dc_1\ ;\ dc_2\ ;\ \ldots\ ;\ dc_n\ \} \cup \{\bot\} \text{ , where:}$$

      - $dc_i \in D_{consumer}$ is a *data node* expressing a data consumer.
      - $D_{consumer} = \varnothing$ indicates that data consumers are unknown.
      - $D_{consumer} = \{\bot\}$ denotes that $\boldsymbol{a}$ is a public attribute.

- $access \in \{r\ ;\ r/w\}$ denotes the access rights of the CaPMan system to the data of a, which can be read or read/write. It expresses the level of control of the system over the data of a. ■

Definition 5.1 (Sensed Attribute)**.** Let $SA \sqsubseteq A$ be the set of ***sensed attributes***, i.e. attributes characterizing sensed data by deployed/wearable sensors, and on which the CaPMan system has access to control and manage, such that: $\forall a \in SA : a.access = r/w$. ■

Definition 5.2 (Background-oriented Attribute)**.** Let $BA \sqsubseteq A$ be the set of ***background-oriented attributes***, i.e. attributes characterizing background data about the user and/or her environment, and on which the CaPMan system has read-only access, such that: $\forall a \in BA : a.access = r$. ■

Example 4. Alice has two sensed attributes: her location and the energy-consumption of her home. They can be represented as follows:

- a-1:⟨ Location ; u ; Log-1 ; r/w ⟩
  - Log-1={⟨(-1.53234,34.0180);Meta-1⟩ ; ⟨(-1.53210;34.0132);Meta-2⟩}
  - Meta-1={$t_{capture}$:11:00:00; $l_{capture}$:(-1.53234,34.0180); *source* :sensor-1; $D_{consumer=}${provider-2}}
  - Meta-2={$t_{capture}$:11:03:00; $l_{capture}$:(-1.53210;34.0132); *source* :sensor-1; $D_{consumer=}${provider-2}}
- a-2:⟨ Energy-consump ; environment-1 ; Log-2 ; r/w ⟩
  - Log-2={⟨89;Meta-3⟩ ; ⟨115;Meta-4⟩}
  - Meta-3={$t_{capture}$:21:05:00; $l_{capture}$:(-1.53245;34.0132); *source* :sensor-2; $D_{consumer=}${provider-1}}
  - Meta-4={$t_{capture}$:21:15:00; $l_{capture}$:(-1.53245;34.0132); *source* :sensor-2; $D_{consumer=}${provider-1}}

Example 5. The system collected several background data about Alice, such as her marital status, profile picture, date of birth (two data collected from different sources), and the services provided in exchange for her sensed data. We represent here the 'date of birth' attribute:

- a-3:⟨ Date-of-Birth ; u ; Log-3 ; r ⟩
  - Log-3={⟨(1976);Meta-5⟩ ; ⟨(26-02-1976);Meta-6⟩}
  - Meta-5={$t_{capture}$:09:00:00; *source* :socialAccount-1; $D_{consumer=}\{\perp\}$}
  - Meta-6={$t_{capture}$:17:30:00; *source* :database-1; $D_{consumer=}\{\perp\}$}

The goal of our study is to detect the user-related privacy risks in the context of connected environments. So we use the uCSN[1] ontology proposed in Chapter

[1] https://spider.sigappfr.org/uCSNdoc/index-en.html

2 to represent the user's contextual knowledge, as it provides: (i) high coverage of generic attributes describing the user and her surrounding connected/unconnected environment; (ii) rich description of scalar/multimedia data and metadata, and data sources with their properties; and (iii) extensibility through the pluggable layer to cover domain-specific knowledge, making it re-usable in various application domains (e.g., smart homes, buildings, cities). The use of an ontology-based data model enables the handling of data heterogeneity in terms of types, metadata, formats, and sources. Data is therefore serialized in a unified format (e.g., RDF/XML), allowing holistic (all-data-inclusive) reasoning to detect the risks involved (cf. Challenge 2). In what follows, we represent previous definitions with their constraints using the formal Description Logic (DL) language [100] in order to clarify their integration in the data model in terms of uCSN classes and properties. DL is a popular knowledge representation language that provides logical formalism for ontologies and the Semantic Web. We do not represent here the *user context* (cf. Definition 4) in DL since it consists of all the modeled individuals (i.e., attributes' data) and their relationships in the respective ontology file.

Definition 1: Data Node

(DataSource $\sqcup$ DataConsumer) (*DATA-NODE*)
DataSource $\equiv$ Sensor $\sqcup$ Device $\sqcup$ ExternalSource
Sensor $\equiv$ StaticSensor $\sqcup$ MobileSensor
DataSource $\sqsubseteq$ (description.VALUE) $\sqcap$ (origin.VALUE) $\sqcap$ (uri-identifier.VALUE)
DataSource $\sqsubseteq$ serialization-format.VALUE
DataConsumer $\equiv$ ServiceProvider $\sqcup$ ThirdParty
DataConsumer $\sqsubseteq$ collaboratesWith.ThirdParty
DataConsumer $\sqsubseteq$ (description.VALUE) $\sqcap$ (uri-identifier.VALUE)

Definitions 2 and 3: Physical Environment and Spatial Zone

(Environment $\sqcup$ (Platform $\sqcap$ $\neg$Device)) (*PHYSICAL-ENVIRONMENT*)
Platform $\equiv$ Environment $\sqcup$ Device
Environment $\sqsubseteq$ (description.VALUE) $\sqcap$ (isDescribedBy.SpatialZone)
SpatialZone $\sqsubseteq$ isComposedOf.Location
Environment $\sqsubseteq$ hosts.(Platform $\sqcup$ System)
Device $\sqsubseteq$ (hosts.System) $\sqcap$ (hasSoftware.Software) $\sqcap$ (hasHardware.Hardware)
System $\equiv$ Sensor $\sqcup$ Actuator $\sqcup$ Sampler
System $\sqsubseteq$ (hasSubSystem.System) $\sqcap$ (uri-identifier.VALUE)

The contextual attributes vary from sensed to background-oriented attributes (cf. Definitions 5.1 and 5.2), such that $\boldsymbol{A} = \boldsymbol{SA} \cup \boldsymbol{BA}$. We represent next both categories of attributes with their properties using DL.

Definition 5.1: Sensed Attribute

(SensedInformation) (*SENSED-ATTRIBUTE*)
SensedInformation $\equiv$ ScalarProperty $\sqcup$ MultimediaProperty
MultimediaProperty $\equiv$ Image $\sqcup$ Audio $\sqcup$ Video
SensedInformation $\sqsubseteq$ (describesEntity.(User $\sqcup$ Environment)) $\sqcap$ (access.VALUE)
SensedInformation $\sqsubseteq$ hasSensingStatus.SensingStatus
SensingStatus $\sqsubseteq$ (isSharedWith.DataConsumer) $\sqcap$ (isSensedBy.Sensor)
SensingStatus $\sqsubseteq$ hasCommunicationProtocol.CommunicationProtocol
SensingStatus $\sqsubseteq$ (hasSensingEvent.Event) $\sqcap$ (hasDataValue.DataValue)
Event $\sqsubseteq$ (hasEventTime.TemporalEntity) $\sqcap$ (hasEventLocation.Location)
DataValue $\sqsubseteq$ (hasCaptureTime.TemporalEntity) $\sqcap$ (hasCaptureLocation.Location)
DataValue $\equiv$ ScalarValue $\sqcup$ MultimediaValue

Definition 5.2: Background-oriented Attribute (User-related attribute)

(PersonalInformation $\sqcap$ $\neg$(SensedInformation $\sqsubseteq$ access.R/W)) (*USER-BG-ATTRIBUTE*)
PersonalInformation $\sqsubseteq$ (isCapturedFrom.DataSource) $\sqcap$ (isSharedWith.DataConsumer)
PersonalInformation $\equiv$ ProfileInformation $\sqcup$ ActivityInformation
(Identifying $\sqcup$ PhysicalCharacteristic $\sqcup$ PublicLife $\sqcup$ Demographic $\sqcup$ Ethnicity
$\sqcup$ KnowledgeBelief $\sqcup$ Preference $\sqcup$ Authenticating) $\sqsubseteq$ ProfileInformation
ProfileInformation $\sqsubseteq$ (hasCaptureTime.TemporalEntity) $\sqcap$ (hasCaptureLocation.Location)
(Activity $\sqcup$ UserLocation $\sqcup$ Behavioral $\sqcup$ SensedInformation) $\sqsubseteq$ ActivityInformation
Activity $\sqsubseteq$ isPerformedAt.Event
Event $\sqsubseteq$ (hasEventTime.TemporalEntity) $\sqcap$ (hasEventLocation.Location)
UserLocation $\sqsubseteq$ hasLocationTime.TemporalEntity
Behavioral $\sqsubseteq$ (hasCaptureTime.TemporalEntity) $\sqcap$ (hasCaptureLocation.Location)

Definition 5.2: Background-oriented Attribute (Environment-related attribute)

(Environment $\sqcup$ SpatialZone $\sqcup$ Device $\sqcup$ System $\sqcup$ CoverageArea
$\sqcup$ Property) (*ENV-BG-ATTRIBUTE*)
Environment $\sqsubseteq$ (description.VALUE) $\sqcap$ (isDescribedBy.SpatialZone)
Environment $\sqsubseteq$ hosts.(Platform $\sqcup$ System)
Platform $\equiv$ Environment $\sqcup$ Device
Device $\sqsubseteq$ hosts.System
System $\equiv$ Sensor $\sqcup$ Actuator $\sqcup$ Sampler
Sensor $\sqsubseteq$ (currentlyCovers.CoverageArea)
System $\sqsubseteq$ (hasOperatingRange.OperatingRange) $\sqcap$ (hasSurvivalRange.SurvivalRange)
System $\sqsubseteq$ (hasSystemCapability.SystemCapability) $\sqcap$ (hasSubSystem.System)
OperatingRange $\sqcup$ SurvivalRange $\sqcup$ SystemCapability $\sqsubseteq$ Property

The uCSN ontology considers also the representation of background data describing the characteristics of (i) devices deployed in the environment (e.g., fire/CO detection device deployed at home) or held by the user (e.g., mobile phone); and (ii) services provided to the user in exchange for her sensed data, such as the personalized energy-saving recommendations provided to Alice by the electricity provider (cf. Section 3.2).

Definition 5.2: Background-oriented Attribute (Device-related attribute)

$(\text{Device} \sqcup \text{Software} \sqcup \text{Hardware} \sqcup \text{NetworkInterface} \sqcup \text{Processor} \sqcup \text{ExpansionCard} \sqcup \text{Memory} \sqcup \text{PowerSupply})$ (*DEVICE-BG-ATTRIBUTE*)

$\text{Device} \sqsubseteq (\text{hasSoftware.Software}) \sqcap (\text{hasHardware.Hardware})$

$\text{Hardware} \sqsubseteq (\text{hasComUnit.NetworkInterface}) \sqcap (\text{hasDeployUnit.ExpansionCard})$

$\text{Hardware} \sqsubseteq (\text{hasProcessingUnit.Processor}) \sqcap (\text{hasStorageUnit.Memory})$

$\text{Hardware} \sqsubseteq (\text{hasPowerUnit.PowerSupply})$

Definition 5.2: Background-oriented Attribute (Service-related attribute)

$(\text{Service} \sqcup \text{Capability} \sqcup \text{Metadata} \sqcup \text{Interface} \sqcup \text{Event} \sqcup \text{Variables})$ (*SERVICE-BG-ATTRIBUTE*)

$\text{ServiceProvider} \sqsubseteq \text{providesService.Service}$

$\text{Service} \sqsubseteq (\text{isProvidedTo.User}) \sqcap (\text{providedThrough.Device}) \sqcap (\text{hasCapability.Capability})$

$\text{Service} \sqsubseteq (\text{hasMetadata.Metadata}) \sqcap (\text{hasInterfaces.Interface})$

$\text{Service} \sqsubseteq (\text{isProvidedAt.Event}) \sqcap (\text{hasVariables.Variables})$

$\text{Variables} \equiv \text{Input} \sqcup \text{Output}$

3.4.2 Privacy Risk Definition

Following the modeling of the user's contextual data, the reasoner requires a reference schema that contains the list of inference rules on which it will rely to detect the privacy risks involved for the user. Nonetheless, the rule definition process is challenging due to the variety of risk types (e.g., direct/indirect risks), as well as the diversity of attribute/data combinations that may entail the disclosure of one or more sensitive information (cf. Challenge 1). To address this challenge, we propose next a generic and re-usable privacy rule syntax that allows the definition of various types of risks with high expressiveness power. This is achieved by considering different types of attributes/data combinations, including combinations through: (i) *logical operators* to connect attributes, data, and constraints; (ii) *comparison operators* to assign conditions to data values; (iii) *spatio-temporal operators* to examine the spatio-temporal correlations among data values; and (iv) *semantic operators* that accurately reflects the semantics of attributes/data relations. One can therefore use the proposed syntax to define basic combinations of attributes and data, as well as

more advanced combinations that consider the semantics of the relationships between them, which improves the quality of the risk definitions. In the following, we formally define a *sensitive information*, and a *privacy rule*.

Definition 6 (Sensitive Information)**.** Let $\boldsymbol{SI}$ be the set of ***sensitive information*** $\{\boldsymbol{si_1}\ ;\ \ldots\ ;\ \boldsymbol{si_n}\}$, which expresses sensitive personal information about a user that could be disclosed when combining and/or processing her contextual data. Such a disclosure can cause serious harms for the user if misused (e.g., mental, physical, dignity/reputation, financial, or societal harms [113]). $\boldsymbol{si} \in \boldsymbol{SI}$ can be of two types: profile information (e.g., age, disease), or activity information (e.g., behavior, physical activity). The set $\boldsymbol{SI}$ can thus be formalized as follows:

$$\boldsymbol{SI} = \boldsymbol{Profile} \sqcup \boldsymbol{Activity} \ \textit{, where:}$$

- $\boldsymbol{Profile}$ represents the set of sensitive information characterizing the user profile, which may vary from generic to domain-specific information, such that:

$$\boldsymbol{Profile} = \boldsymbol{Generic} \sqcup \boldsymbol{Medical} \sqcup \boldsymbol{Financial} \sqcup \boldsymbol{Professional} \sqcup \boldsymbol{Social} \sqcup \boldsymbol{Other}$$

  - $\boldsymbol{Generic}$ represents the set of generic profile information

$$\boldsymbol{Generic} = \{\textit{re-identification ; age ; date-of-birth; marital-status ; gender ; height ; weight ; political-affiliation ; sexual-orientation ; physical-trait ; ethnic-origin ; race ; religion ; language ; dialect ; accent ; preference ; interest}\}$$

    - *re-identification* is the happening that occurs when the anonymized sensed data (e.g., location data) is matched with its true owner (i.e., user).

  - $\boldsymbol{Medical}$ represents the set of medical profile information

$$\boldsymbol{Medical} = \{\textit{disease ; allergy ; surgery ; immunization ; blood-type ; drug-test ; mental-health ; genetic}\}$$

  - $\boldsymbol{Financial}$ represents the set of financial profile information

$$\boldsymbol{Financial} = \{\textit{credit-information ; bank-account-information ; transactional-information ; card number ; card type}\}$$

  - $\boldsymbol{Professional}$ represents the set of professional profile information

$$\boldsymbol{Professional} = \{\textit{salary ; job ; certification ; academic-degree}\}$$

- ***Social*** represents the set of social profile information

$$\boldsymbol{Social} = \{\textit{family ; friend ; association ; membership ; meeting}\}$$

- ***Other*** represents the set of other profile information that could be specified by the user or by the privacy community when defining risks.

- ***Activity*** represents the set of sensitive information characterizing the user activity, such that:

$$\boldsymbol{Activity} = \{\textit{behavior ; habit ; performed-activity ; presence-absence ; sleeping-cycle ; fraud}\} \sqcup \boldsymbol{Other}$$

- ***Other*** represents the set of other activity information that could be specified by the user or by the privacy community when defining risks. ■

Definition 7 (Privacy Rule)**.** Let $\boldsymbol{PR}$ be the set of ***privacy rules***, $\{\boldsymbol{pr_1 ; \ldots ; pr_n}\}$, that define the risks to be detected by the reasoner. A privacy rule, $\boldsymbol{pr \in PR}$, is an inference rule, specified by means of an ontology language (e.g., OWL [100]) in the form of an if-then (antecedent-consequent) sentence. It explicitly specifies a sequence of attribute/data elements that, when combined using the stated operators, results in the disclosure of one or more sensitive information. $\boldsymbol{pr \in PR}$ is defined according to the following syntax:

$$\boldsymbol{pr : \varphi(E) \rightarrow SI'} \quad \text{, where:}$$

- $\boldsymbol{\varphi(E) = e_1\ \theta\ e_2\ \theta\ \ldots\ \theta\ e_n}$ represents the sequence of attribute/data elements $\boldsymbol{\{e_1 ; e_2 ; \ldots ; e_n\} \in E}$, combined through operators $\boldsymbol{\theta}$, such that:
  - $\forall \boldsymbol{e \in E}, \quad \boldsymbol{e \in \{class ; individual ; comparisonValue\}}$, where:
    - ***class*** is an ontology class denoting a user, environment, attribute (i.e., sensed or background-oriented attribute), data node, or a metadata related to the previous classes (e.g., time, location).
    - ***individual*** is an ontology individual expressing a data value of a class.
    - ***comparisonValue*** is a comparison value with a primitive data type of Boolean, Decimal, or String, that is used when a condition is assigned to one or more data values.
  - $\boldsymbol{\theta}$ is an operator that combines two or more attribute/data elements. It belongs to one of the following categories:

    $\boldsymbol{\theta \in \{Logical \sqcup Comparison \sqcup Spatial \sqcup Temporal \sqcup Semantic\}}$, where:

    - $\boldsymbol{Logical} = \{\textit{AND ; OR ; NOT}\}$ is the set of logical operators.

  - $\boldsymbol{Comparison} = \{>\ ;\ <\ ;\ >=\ ;\ <=\ ;\ =\ ;\ !=\}$ is the set of comparison operators.
  - $\boldsymbol{Spatial} = \{$*contains ; covers ; crosses ; equals ; above ; below ; closeTo ; disjointWith ; farFrom ; leftOf ; rightOf ; overlaps*$\}$ is the set of spatial operators.
  - $\boldsymbol{Temporal} = \{$*inside ; before ; after ; contains ; disjoint ; during ; equals ; overlaps*$\}$ is the set of temporal operators.
  - $\boldsymbol{Semantic}$ is the set of semantic operators (i.e., ontology relations).

- $\boldsymbol{SI'} \sqsubseteq \boldsymbol{SI}$ represents the set of sensitive information disclosed by $\boldsymbol{\varphi(E)}$. ■

Next, we provide examples of privacy rules, defined using the proposed syntax, and expressing direct and indirect risks.

Example 6. We provide here examples of rules that express direct risks (i.e., risks controlled by the user):

- Rule 1: A user is sharing her location data with a data consumer without any protection. This raises the risk of inferring her habits, behaviors and preferences.

  $\boldsymbol{PR_1}$: (User) *hasPersonalInformation* (SensedInformation = Location)
  AND (Location) *hasSensingStatus* (SensingStatus = Status-1)
  AND (Status-1) *isSharedWith* (DataConsumer)
  AND (Status-1) *isProtected* (= false)
  $\longrightarrow$ {habits ; behaviors ; preferences}

- Rule 2: A user is sharing her location data with a data consumer, the data is anonymized and the user shares publicly her home address on Facebook. This raises the risk of re-identification of the user through protected data.

  $\boldsymbol{PR_2}$: (User) *hasPersonalInformation* (SensedInformation = Location)
  AND (User) *hasPersonalInformation* (HomeAddress)
  AND (Location) *hasSensingStatus* (SensingStatus = Status-1)
  AND (Status-1) *isSharedWith* (DataConsumer)
  AND (Status-1) *isProtected* (= true)
  AND (Status-1) *isProtectedUsing* (ProtectionFunction = Anonymization)
  AND (HomeAddress) *isSharedWith* (DataConsumer = $\perp$)
  $\longrightarrow$ {re-identification}

- Rule 3: A user is sharing her location data with a data consumer without protection, and is located in a medical center dedicated to the treatment of a specific disease. This raises the risk of inferring her disease.

  $\boldsymbol{PR_3}$: (User) *hasPersonalInformation* (SensedInformation = Location)
  AND (Location) *hasSensingStatus* (SensingStatus = Status-1)
  AND (Status-1) *isSharedWith* (DataConsumer)
  AND (Status-1) *isProtected* (= false)

AND (User) *isLocatedIn* (Environment)
AND (Environment) *hasDescription* (= "Medical-Center" *OR* "Hospital")
AND (Environment) *isDedicatedFor* (Disease)
⟶ {disease}

- Rule 4: A user is sharing the energy consumption data of her home with data consumers without protection. This raises the risks of inferring her presence/absence, sleeping cycles, and home activities.

  $\boldsymbol{PR_4}$ **:** (SensedInformation = Energy-consump)
  AND (Energy-consump) *describesEntity* (Environment)
  AND (Environment) *hasDescription* (= "Home")
  AND (Energy-consump) *hasSensingStatus* (SensingStatus = Status-1)
  AND (Status-1) *isSharedWith* (DataConsumer)
  AND (Status-1) *isProtected* (= false)
  ⟶ {presence-absence, sleeping-cycles, activities}

- Rule 5: A user has a chronic disease and follows her medical treatment using a medical device deployed at home. She is sharing the energy consumption data of her home without protection. This raises the risk of inferring her disease.

  $\boldsymbol{PR_5}$ **:** (User) *controlsEnv* (Environment)
  AND (Environment) *hasDescription* (= "Home")
  AND (Environment *hosts* (Device = MedicalDevice))
  AND (SensedInformation = Energy-consump)
  AND (Energy-consump) *describesEntity* (Environment)
  AND (Energy-consump) *hasSensingStatus* (SensingStatus = Status-1)
  AND (Status-1) *isSharedWith* (DataConsumer)
  AND (Status-1) *isProtected* (= false)
  ⟶ {disease}

Example 7. We provide here examples of rules that express indirect risks (i.e., risks uncontrolled by the user):

- Rule 6: A user is located in a public environment (e.g., mall, street) that hosts CCTV cameras. This raises the risks of inferring user presence/absence in the environment, her interests, and her activities in this environment.

  $\boldsymbol{PR_6}$ **:** (User) *isLocatedIn* (Environment)
  AND (User) *NOT*(*controlsEnv* Environment)
  AND (Environment) *hosts* (System = CCTV)
  ⟶ {presence-absence ; interests ; activities}

- Rule 7: The home/office street of the user hosts a CCTV camera, and this camera has a coverage area that contains the spatial zone of the home/office. This raises the risks of inferring the presence/absence of the user at home/office, and her activities in the area covered by the camera.

PR_7 : (User) *controlsEnv* (Environment-1)
AND (Environment-1) *hasDescription* (= "Home" *OR* "Office")
AND (Environment-1) *hasSpatialZone* (SpatialZone)
AND (Environment-2) *NOT*(*hasDescription* (= "Home"))
AND (Environment-2) *hosts* (System = CCTV)
AND (Surveillance-camera) *hasCoverageArea* (CoverageArea)
AND (CoverageArea) *covers* (SpatialZone)
$\longrightarrow$ {presence-absence ; activities}

- Rule 8: The user is located in a public environment that hosts Automatic Number Plate Recognition (ANPR) sensors. This raises the risk of inferring the presence/absence of the user in the environment.

  PR_8 : (Environment) *hosts* (System = ANPR)
  AND (User) *NOT*(*controlsEnv* Environment)
  AND (User) *isLocatedIn* (Environment)
  $\longrightarrow$ {presence-absence}

- Rule 9: The user is located in a medical center or hospital that hosts CCTV cameras. This raises the risks of inferring the presence/absence of the user in the medical environment, her medical information (e.g., disease, surgery, allergy).

  PR_9 : (User) *isLocatedIn* (Environment)
  AND (Environment) *hasDescription* (= "Medical-Center" OR "Hospital")
  AND (Environment) *hosts* (System = CCTV)
  $\longrightarrow$ {presence-absence ; medical-information}

Enhancing the quality of the risk inference process necessitates not only expanding the coverage of direct and indirect risks, but also providing high-quality rule definitions. Indeed, the more we explore application domains, the more we discover combinations of attribute/data elements that lead to disclose sensitive information about a user. In addition, the privacy rules defined must be regularly updated in order to cope with evolution of data sensing and mining technologies. To overcome these problems, the CaPMan system collaborates with a group of privacy experts belonging to various application domains. This collaboration is done using an outsourcing solution that enables the privacy community to define various privacy rules using the proposed syntax, and update existing ones. It also checks the rules validity, and manages the rules conflicts and dependencies. The implementation of this solution and the tackling of associated challenges will be explored further in future work. At this stage, we consider that the expert-defined rules are pre-validated. The rule updates are thus imported regularly by the *privacy rules* component of the CaPMan framework and converted to the chosen semantic rule language (e.g., W3C Semantic Web Rule Language [114]) before being provided as input to the risk reasoner as shown in Figure 3.5.

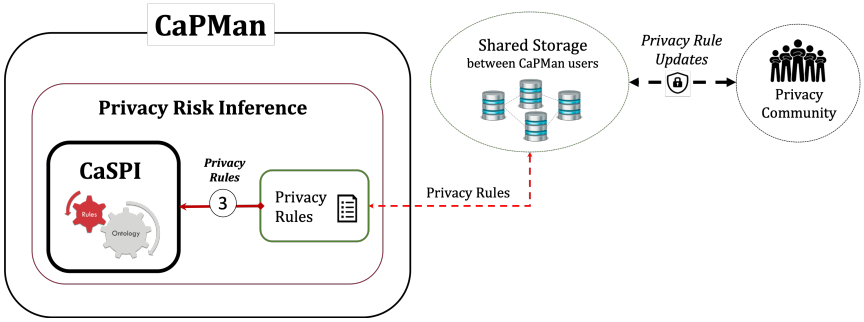

FIGURE 3.5: Privacy Rules Import

The privacy rules defined can be of two categories: (i) domain-independent rules, generic rules that are valid in all application domains; and (ii) domain-specific rules, rules that are only valid in specific domains (e.g., smart homes, vehicles, hospitals, buildings, cities). For example, rules PR_1, PR_2, PR_6, PR_7, and PR_8 are domain-independent rules, PR_3 and PR_9 are specific to the smart hospital domain, and PR_4 and PR_5 are specific to the smart home domain. Experts are therefore able to define rules from both categories. Domain-specific rules rely on the use of domain-specific vocabulary imported from existing ontology-based models (e.g., Vehicle Signal and Attribute Ontology [115], ontology for smart homes [116], building topology ontology [63]). Thus, in addition to the generic rules, the system imports the rules related to the considered domains, and the associated ontologies are plugged into the generic uCSN ontology to ensure interoperability between inference rules and knowledge representation.

3.4.3 User Profiles

Users might have different levels of expertise when it comes to specifying their preferences (e.g., which sensitive information is significant for them), and understanding their privacy risks (cf. Challenge 4). The guided assistance must therefore be tailored to the user's expertise, which helps in improving the quality of user-system interactions. Consequently, we define in the following three user profiles:

- **Beginner**: The user is not familiar with her privacy, which means she does not know how to interpret what is sensitive for her and what is not; nonetheless, she asks for comprehensible descriptions of the risks she accepts to take.
- **Intermediate**: The user understands how to specify her preferences for sensitive information. However, she only requires a detailed overview of the significant risks to her (i.e., the risks associated with significant information inferences).
- **Advanced**: The user is expert in interpreting and analyzing her privacy situation. She can ask for full details about the significant and non-significant risks involved in her situation.

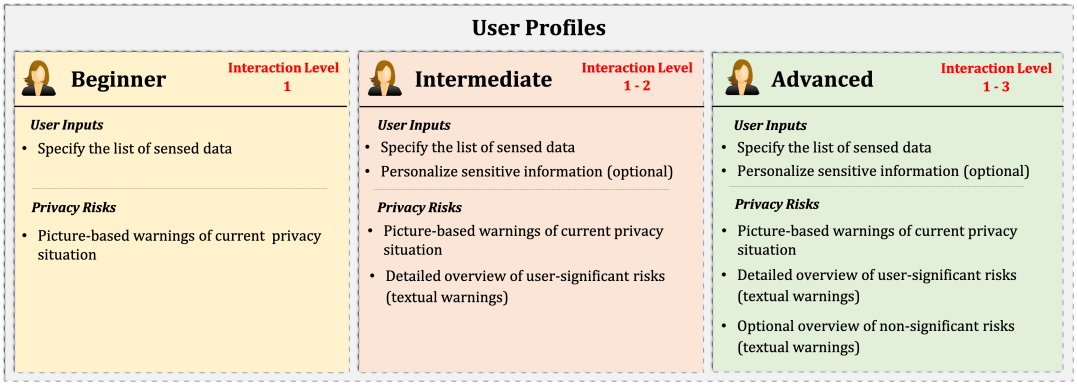

FIGURE 3.6: User Profiles

Figure 3.6 details the defined profiles and their characteristics. The goal here is to limit the level of user-interaction with the system, and the bunch of information provided, according to her profile. The level of user-interaction is expressed by a min-max number in Figure 3.6. For a *beginner*, the system requires only to receive the list of sensed data. It reasons over contextual information while considering all sensitive information as significant inferences for the user. Once done, it summarizes the risks detected and provides the user with comprehensible descriptions of her current privacy situation through picture-based warnings. For an *intermediate*, the system asks for the list of sensed data and allows the personalization of sensitive information. It reasons accordingly over context information to detect only the user-significant risks. Once done, it provides comprehensible picture-based warnings that summarizes the current privacy situation, as well as a detailed risk overview using textual warnings, which includes the risks with their associated sensed data, sensitive information, and values. For an *advanced*, the system provides all *intermediate* options plus an optional detailed overview of non-significant risks.

3.4.4 CaSPI Reasoner

CaSPI employs a semantic reasoner that performs rule-based reasoning over modeled context information, based on the privacy rules imported, in order to infer the risks involved in the relevant user context. The reasoner is launched by default when a context change occurs, allowing continuous monitoring of the risk evolution to cope with the dynamicity of the user context (cf. Challenge 3). In the following, we formally describe the risk inference process and define a *privacy risk*. Then, we detail our proposed algorithm.

The risk inference process consists of executing the $\boldsymbol{riskReasoner}()$ function that takes as input: (i) user-context information, $(\boldsymbol{c.A} \sqcup \boldsymbol{c.Rel}_A)$; (ii) user profile, $\boldsymbol{profile}$; (iii) user preferences related to sensitive information, $\boldsymbol{uSI} \sqsubseteq \boldsymbol{SI}$; and (iv) privacy rules, $\boldsymbol{PR}$. It returns an overview of the risks taken by the user $\boldsymbol{u}$ in $\boldsymbol{c}$, denoted by $\boldsymbol{R_c}$. This can be represented as follows:

$$\boldsymbol{riskReasoner}\big((\boldsymbol{c.A} \sqcup \boldsymbol{c.Rel_A})\ ;\ \boldsymbol{profile}\ ;\ \boldsymbol{uSI}\ ;\ \boldsymbol{PR}\big) \rightarrow \boldsymbol{R_c}$$

Where:

$$\boldsymbol{R_c} = \begin{bmatrix} r_1 & r_2 & \dots & r_n \end{bmatrix} \mid n \in \mathbb{N}$$

Definition 8 (Privacy Risk)**.** A ***privacy risk***, $\boldsymbol{r} \in \boldsymbol{R_c}$, is defined as the risk of disclosing one or more *sensitive information* about the user. Each $\boldsymbol{r} \in \boldsymbol{R_c}$ is associated with one distinct *privacy rule*, $\boldsymbol{pr} \in \boldsymbol{PR}$, that is satisfied in the context $\boldsymbol{c} \in \boldsymbol{C}$. It expresses the probability of achieving the logical consequence of the related $\boldsymbol{pr}$ in $\boldsymbol{c}$ (i.e., $\boldsymbol{pr.SI'}$). $\boldsymbol{r}$ has a probabilistic value ranging from 0 to 1, where 0 indicates that $\boldsymbol{r}$ is negligible, and 1 indicates that the information disclosure is materialized at 100% (i.e., the disclosure of $\boldsymbol{pr.SI'}$). $\boldsymbol{r}$ can be represented as follows:

$$\boldsymbol{r} : \langle\ \boldsymbol{id}\ ;\ \boldsymbol{value}\ ;\ \boldsymbol{SI'}\ ;\ \boldsymbol{SA'}\ \rangle \mid \exists!\ \boldsymbol{pr} \in \boldsymbol{PR}\ :\ \boldsymbol{r} \sim \boldsymbol{pr}$$

where:

- $\boldsymbol{id} \in \mathbb{N}$ denotes the risk identifier.
- $\boldsymbol{value} \in [\mathbf{0;1}]$ denotes the risk value. The quantification of $\boldsymbol{r.value}$ is detailed in Chapter 4.
- $\boldsymbol{SI'} = \boldsymbol{SI'_{pr}}$ is the set of sensitive information associated with $\boldsymbol{r}$.
- $\boldsymbol{SA'} = \{a_1, ..., a_n\} \sqsubseteq (\boldsymbol{E_{pr}} \sqcap \boldsymbol{c.SA})$ denotes the set of sensed attributes associated with $\boldsymbol{r}$ (i.e., stated in the set of elements of $\boldsymbol{pr}$, $\boldsymbol{E_{pr}}$). ■

The number of risks to detect when executing the *riskReasoner*() function depends on the number of privacy rules imported in this iteration, such that: $|\boldsymbol{PR}| = |\boldsymbol{R_c}|$ only if all $\boldsymbol{pr} \in \boldsymbol{PR}$ are satisfied. The privacy risks detected are stored with their properties in the ontology file of the relevant context. They are modeled according to the following classes and properties represented in DL:

Privacy Risk and Sensitive Information

(PrivacyRisk) (*RISK*)
PrivacyRisk $\sqsubseteq$ (risk-identifier.VALUE) $\sqcap$ (hasValue.VALUE)
PrivacyRisk $\sqsubseteq$ (hasInference.SensitiveInformation) $\sqcap$ (hasSensedInfo.SensedInformation)
SensitiveInformation $\equiv$ PersonalInformation $\sqcap$ $\neg$Identifying
SensitiveInformation $\equiv$ (ProfileInformation $\sqcap$ $\neg$Identifying) $\sqcup$ ActivityInformation
(Generic $\sqcup$ Medical $\sqcup$ Financial $\sqcup$ Professional $\sqcup$ Social) $\sqsubseteq$ ProfileInformation
SensitiveInformation $\sqsubseteq$ hasDescription.VALUE

The uCSN ontology is extended in the current application to account for privacy risk modeling. Privacy risks are therefore represented as individuals of the `ucsn:PrivacyRisk` concept. The risk identifiers and values are respectively represented by the `ucsn:risk-identifier` and `ucsn:hasValue` properties. On one hand, each `ucsn:PrivacyRisk` has one or more associated `ucsn:SensitiveInformation` that may vary from `ucsn:ProfileInformation` to `ucsn:ActivityInformation` (cf. Definition 6). `ucsn:ProfileInformation` can be `ucsn:Generic` (e.g., age, marital-status) or domain-specific, such as `ucsn:Medical` (e.g., disease), `ucsn:Financial` (e.g., card number), `ucsn:Professional` (e.g., salary), and `ucsn:Social` (e.g., friends). An additional description can be provided to each `ucsn:SensitiveInformation`, using the `ucsn:hasDescription` property, which helps in better expressing its meaning in the relevant context. On the other hand, a `ucsn:PrivacyRisk` may have one or more associated `ucsn:SensedInformation` (e.g., location, energy-consumption).

3.4.4.1 Reasoning Algorithm

Algorithm 1 presents the algorithm of the *riskReasoner*() function, which takes as input: (i) the ontology file comprising context information, *contextFile*; (ii) the array of privacy rules, PR; (iii) the user profile, $profile$; and (iv) the list of sensitive information with their preference flags for the user, uSI. uSI is a two-dimensional array, where the first column contains the list of sensitive information (i.e., SI), and the second column contains the associated flags expressing user preferences, with a value of 1 if the information is significant for the user, and 0 otherwise. It is important to receive both significant and non-significant information as input, so that the system can provide *advanced* users with an additional insight into non-significant but taken risks (See Figure 3.6). For *beginner/intermediate* users, the preference flags allow filtering the privacy rules so that only those that lead to the disclosure of user-significant information are considered.

The algorithm outputs the array of privacy risks involved in the relevant user context, R_c. This is done following five major steps:

- Step 1 (lines 3-4): It filters the array of rules (PR) before launching the reasoning process, to consider only:
  (a) rules that are significant for the user (i.e., those involving at least one sensitive information of SI that is user-significant).
  (b) rules including only the sensed attributes in the relevant context (i.e., $\forall e \in E_{pr} \sqcap SA : e \in c.SA$).
  (c) rules comprised of only background-oriented attributes and data (i.e., expressing indirect risks).

To do so, it starts by extracting the list of sensed attributes from *contextFile* and stores it in the *sensedAttribute* array (line 3). This is done by calling the function *getSensedAttributes*(). Then, it calls the *filterPrivacyRules*() function that returns the filtered array of rules (line 4).

- Step 2 (lines 5-6): It calls the *createRuleEngine*() function that creates a rule engine instance based on the privacy rules considered, and maps it to the ontology file (line 5). Then, it calls the *createReasoner*() function that creates an ontology reasoner instance and maps it to the ontology file (line 6).
- Step 3 (line 7): It launches the rule-based inference engine to detect the risks involved, by calling the function *infer*().
- Step 4 (lines 8-9): It flushes the changes stored in the buffer (i.e., risks inferences) by calling the function *flush*(), causing the reasoner to append the changes in the ontology instance. Then, the ontology file is updated to save the new inferences by calling the function *saveUpdates*() (line 9).
- Step 5 (line 10): It extracts the privacy risks with their properties (i.e., $r.id$, $r.value$, $r.SI'$, and $r.SA'$) from the ontology file using the *getPrivacyRisks*() function, and stores them in the four-dimensional array R_c, where each row denotes one risk.

Algorithm 1: CaSPI Reasoner

Input: *contextFile*, *PR*[], *uSI*[][], *profile*; // the ontology file containing individuals and relationships expressing the current context, the array of privacy rules, the array of sensitive information with their preference flag, and the user profile;

Output: R_c[][][][]; // the overview of risks, where each row presents the properties of a risk: risk id, risk value, associated sensitive information and sensed data (for direct risks);

1 **Variables:** *sensedAttributes[]*, *ruleEngine*, *reasoner*; // set of sensed attributes, the rule engine variable, and the ontology reasoner variable;

2 **begin**

3 *sensedAttributes*[] ← *getSensedAttributes*(*contextFile*); // returns the set of the currently sensed attributes ;

4 *PR*[] ← *filterPrivacyRules*(*PR*[], *sensedAttributes*[], *uSI*[][], *profile*); // returns the set of filtered rules ;

5 *ruleEngine* ← *createRuleEngine*(*PR*[], *contextFile*); // create the rule engine instance, associate the considered privacy rules, and map it to the ontology file;

6 *reasoner* ← *createReasoner*(*contextFile*); // create the ontology reasoner instance and map it to the ontology file;

7 *ruleEngine* ← *infer*(); // run the inference function to infer the involved privacy risks;

8 *reasoner* ← *flush*(); // flush the reasoner to consider the risk inferences;

9 *contextFile* ← *saveUpdates*(); // save the updates in the ontology file;

10 R_c[][][][] ← *getPrivacyRisks*(*contextFile*); // get the list of privacy risks inferred with their properties: id, value, associated sensitive information and sensed attributes (if exist);

11 **return** R_c[][][][]

The pseudo-codes of the functions called in the reasoning algorithm are detailed in the prototype source code provided in Section 3.5.

3.5 Implementation & Evaluation

In this section, we present the implementation phases of the CaSPI proposal from the back-end and front-end perspectives. Then, we evaluate the performance of the risk reasoner in multiple cases and we formally study its storage complexity.

3.5.1 CaSPI Implementation

In order to validate our proposal, we developed a Java-based prototype of the system using Semantic Web tools, such as OWL API, SWRL API, and Pellet reasoner, and we embedded it on the user's mobile device. As illustrated in Figure 3.7, the prototype collects and models the contextual data of the user (i.e., sensed and background data), as well as user inputs, which vary according to the selected profile (cf. Section 3.4.3). It performs then rule-based reasoning over modeled data, based on imported and filtered rules, and outputs an overview of the risks involved in the relevant context with their characteristics. The produced overview is consequently tailored to the user's profile before being released in order to allow all users to understand their privacy implications. The source code of the CaSPI prototype is available online for download via this link[2].

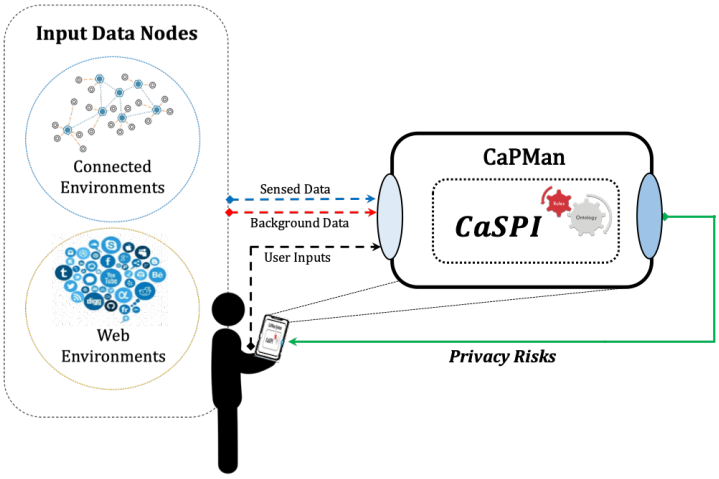

FIGURE 3.7: Implementation of the CaSPI proposal

In what follows, we illustrate how the prototype works in the back-end. Then, we present the front-end mockups of the associated mobile application. It is important to note that this application is currently under development. We only represent here the mockups of the respective user interfaces.

3.5.1.1 Back-end: Java-based Prototype

The goal here is to showcase how the system works and to highlight its ability to track the evolution of risks in response to changes in the user context (cf. Challenge

[2] https://spider.sigappfr.org/research-projects/privacy-oracle/

3). To do so, we consider the context describing Alice's situation in Section 3.2 as the first context, followed by two context changes.

- *Context-1*: Alice is located at home, shares her location data, sensed by GPS sensor of her mobile phone, with a healthcare provider, and shares the energy-consumption data of her home, sensed by a deployed energy sensor, with an electricity provider. Alice has a NIV device deployed at home. Other background data are also known about Alice, such as her date of birth, and marital-status. Figure 3.8 shows how current context information are modeled as uCSN individuals, along with their relationships. We used Protege 5.5.0[3] to illustrate them.
- *Context-2*: Alice continues to share the same data with consumers, but she is now located in a shopping mall that hosts surveillance cameras.
- *Context-3*: Alice leaves the mall two hours later.

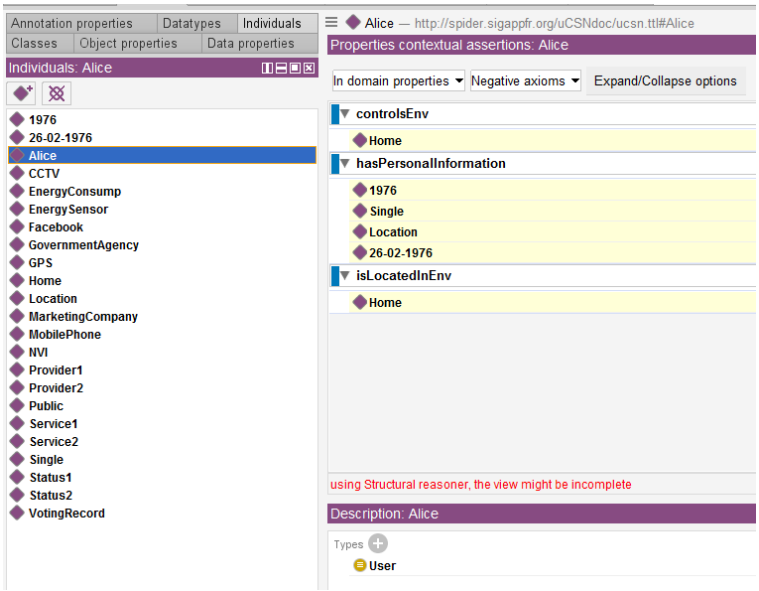

FIGURE 3.8: Context-1 of Alice

Besides, we define the nine privacy rules, provided as examples in Section 3.4.2, using the Semantic Web Rule Language (SWRL) [114]. SWRL is a W3C recommended standard language that combines OWL expressivity with the Rule Markup Language (RuleML) to define rules. It can be roughly considered as the union of Horn-Logic and OWL based on the description logic $\mathcal{SHOIN}$. SWRL allows for interoperability, re-usability, extensibility (through built-ins), and computational scalability [114]. As explained by Fiorentini [117], SWRL provides association rules, that allows to associate new individuals to classes and create properties between individuals. The SWRL-based rule syntax follows the same structure of the one provided in Definition 7. We represent in the following $\boldsymbol{PR_1}$ and $\boldsymbol{PR_4}$ using SWRL. The remaining seven rules are included in the prototype source code.

[3] `https://protege.stanford.edu/`

PR_1 in SWRL syntax

A user is sharing her location data with a data consumer without any protection. This raises the risk of inferring her habits, behaviors and preferences.

```
ucsn:User(?u) ∧ ucsn:hasPersonalInformation(?u, ucsn:LOCATION)
∧ ucsn:SensedInformation(ucsn:LOCATION) ∧ ucsn:isProtected(ucsn:LOCATION, false)
∧ ucsn:hasSensingStatus(ucsn:LOCATION, ?status) ∧ ucsn:DataConsumer(?d)
∧ ucsn:isSharedWith(?status, ?d)
∧ swrlx:createOWLThing(?r, ucsn:HABIT, ucsn:PREFERENCE, ucsn:BEHAVIOR)
⟶
ucsn:PrivacyRisk(?r) ∧ ucsn:ActivityInformation(ucsn:HABIT)
∧ ucsn:Generic(ucsn:PREFERENCE) ∧ ucsn:ActivityInformation(ucsn:BEHAVIOR)
∧ ucsn:hasDescription(ucsn:HABIT, "Habits of the user")
∧ ucsn:hasDescription(ucsn:PREFERENCE, "Preferences of the user")
∧ ucsn:hasDescription(ucsn:BEHAVIOR, "Behaviors of the user")
∧ ucsn:hasValue(?r, 1) ∧ ucsn:hasInference(?r, ucsn:HABIT)
∧ ucsn:hasInference(?r, ucsn:PREFERENCE) ∧ ucsn:hasInference(?r, ucsn:BEHAVIOR)
∧ ucsn:hasSensedInfo(?r, ucsn:LOCATION)
```

PR_4 in SWRL syntax

A user is sharing the energy consumption data of her home with data consumers without protection. This raises the risks of inferring her presence/absence, sleeping cycles, and home activities.

```
ucsn:User(?u) ∧ ucsn:Environment(?env) ∧ ucsn:hasDescription(?env, "Home")
∧ ucsn:controlsEnv(?u, ?env) ∧ ucsn:SensedInformation(ucsn:ENERGY-CONSUMP)
∧ ucsn:describesEntity(ucsn:ENERGY-CONSUMP, ?env)
∧ ucsn:hasSensingStatus(ucsn:ENERGY-CONSUMP, ?status) ∧ ucsn:DataConsumer(?d)
∧ ucsn:isSharedWith(?status, ?d) ∧ ucsn:isProtected(ucsn:ENERGY-CONSUMP, false)
∧ swrlx:createOWLThing(?r, ucsn:PRESENCE-ABSENCE)
∧ swrlx:createOWLThing(ucsn:SLEEPING-CYCLE, ucsn:PERFORMED-ACTIVITY)
⟶
ucsn:PrivacyRisk(?r) ∧ ucsn:ActivityInformation(ucsn:PRESENCE-ABSENCE)
∧ ucsn:ActivityInformation(ucsn:SLEEPING-CYCLE)
∧ ucsn:ActivityInformation(ucsn:PERFORMED-ACTIVITY)
∧ ucsn:hasDescription(ucsn:PRESENCE-ABSENCE, "User's presence/absence at Home")
∧ ucsn:hasDescription(ucsn:SLEEPING-CYCLE, "Sleeping cycles of the user")
∧ ucsn:hasDescription(ucsn:PERFORMED-ACTIVITY, "Home activities of the user")
∧ ucsn:hasValue(?r, 1) ∧ ucsn:hasInference(?r, ucsn:PRESENCE-ABSENCE)
∧ ucsn:hasInference(?r, ucsn:SLEEPING-CYCLE)
∧ ucsn:hasInference(?r, ucsn:PERFORMED-ACTIVITY)
∧ ucsn:hasSensedInfo(?r, ucsn:ENERGY-CONSUMP)
```

The system monitors the user's situation continuously and launches the risk reasoner by default when a change takes place. When launched in *context-1*, only rules PR_1, PR_4, and PR_5 are satisfied, generating consequently three privacy risks. The corresponding risk overview, R_c, is illustrated in Figure 3.9. The *sensitive information* column includes the descriptions of the associated disclosed sensitive information, which are defined in the rules using the `ucsn:hasDescription` property. When the system receives information about the changed environment (i.e., *context-2*), it relaunches the reasoner and updates the risk overview based on the new inferences. Figure 3.10 shows that previous risks are still valid in the new context because the changes had no effect on them, and a new risk is inferred related to the rule PR_6. Once Alice leaves the mall (i.e., *context-3*), the system detects the changes and the inference engine is relaunched. Figure 3.11 shows that the fourth risk detected in *context-2* became negligible and was thus eliminated. Only the first three risks remains valid. The average time of the reasoning process is 1 ms in all three contexts.

May 6, 2021 16:30:26 ms
Context-1: Alice is Located at Home

May 6, 2021 16:30:28 ms
Number of detected privacy risks: **3 Risks**

Risk #	*Sensed Data*	*Sensitive Information*	*Risk Value*
1	Location	- Habits of the user - Preferences of the user - Behaviors of the user	100%
2	EnergyConsump	- User's presence/absence at Home - Sleeping cycles of the user - Home activities of the user	100%
3	EnergyConsump	- User's disease when using the medical device deployed at Home	100%

FIGURE 3.9: Privacy Risk Overview in Context-1

May 6, 2021 18:35:32 ms
Context-2: Alice is Located in a Mall hosting CCTV Cameras

May 6, 2021 18:35:33 ms
Number of detected privacy risks: **4 Risks**

Risk #	*Sensed Data*	*Sensitive Information*	*Risk Value*
1	Location	- Habits of the user - Preferences of the user - Behaviors of the user	100%
2	EnergyConsump	- User's presence/absence at Home - Sleeping cycles of the user - Home activities of the user	100%
3	EnergyConsump	- User's disease when using the medical device deployed at Home	100%
4		- User's presence/absence in the current environment via CCTV cameras - User activities in the current environment via CCTV cameras - User interests via CCTV cameras	100%

FIGURE 3.10: Privacy Risk Overview in Context-2

May 6, 2021 20:40:37 ms
Context-3: Alice leaves the Mall

May 6, 2021 20:40:39 ms
Number of detected privacy risks: **3 Risks**

Risk #	*Sensed Data*	*Sensitive Information*	*Risk Value*
1	Location	- Habits of the user - Preferences of the user - Behaviors of the user	100%
2	EnergyConsump	- User's presence/absence at Home - Sleeping cycles of the user - Home activities of the user	100%
3	EnergyConsump	- User's disease when using the medical device deployed at Home	100%

FIGURE 3.11: Privacy Risk Overview in Context-3

3.5.1.2 Front-end: User Interfaces

The mockups of the mobile application user interfaces were designed using Inkscape graphics editor 1.0[4]. The user starts first by logging in to the application through the login page illustrated in Figure 3.12. The user can create her account automatically by syncing it to her Facebook or Google account, or she can register manually through the application. For each of these scenarios, the user must specify her privacy-aware level of expertise (i.e., Beginner, Intermediate, Expert) that will be assigned to her account (cf. Figure 3.12).

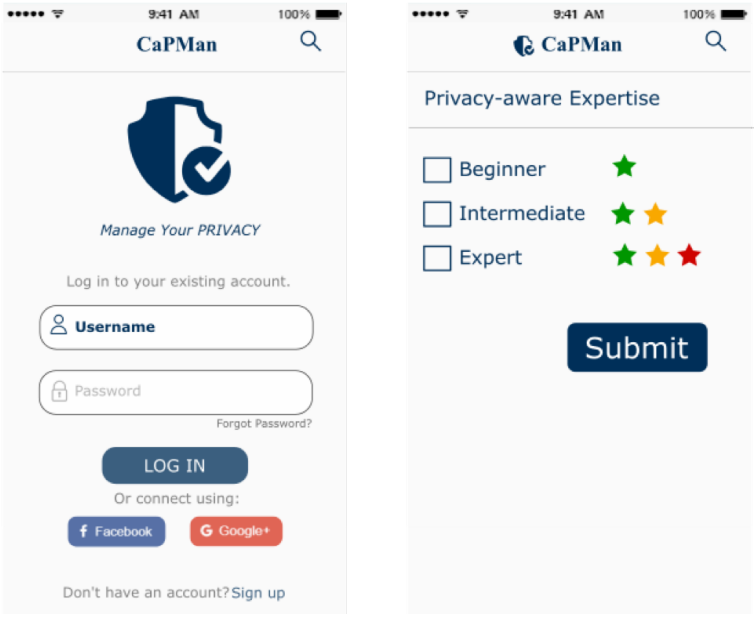

FIGURE 3.12: Login and Profile Specification Interfaces

Once logged in, the user is asked to select her currently sensed data, which may vary from generic user/environment data to domain-specific data (e.g., healthcare data). The user specifies the environment description (e.g., home, office) and selects

[4] https://inkscape.org/

the related sensed data. She can define several environments with the "add" button. The user has the possibility to define new inputs of data and/or domains, which will be automatically taken into account for future specifications. Figure 3.13 illustrates the respective user interface. In the example of Alice, the data selected in her current context are her "Location" and the "Energy consumption" of her home.

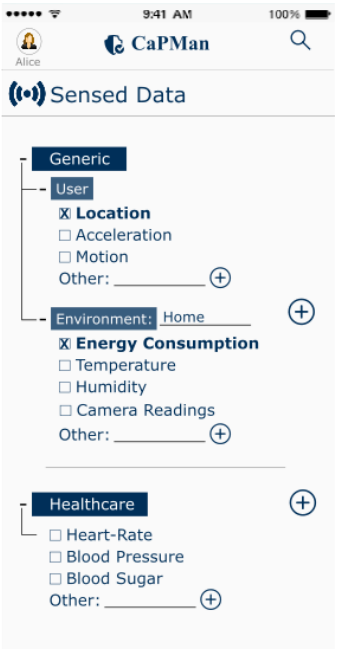

FIGURE 3.13: Sensed Data Selection Interface

FIGURE 3.14: Personalizing Sensitive Information Interface

After specifying her sensed data, only an Intermediate or Advanced user has the option to personalize her sensitive information as shown in Figure 3.14. The personalization can be done by selecting predefined groups of information (e.g., ethnic information, public life information), which could be available in the default settings of the application or customized by the user. As well, the user can manually choose the instances of information that are sensitive to her regardless of their groups. The information instances vary from profile to activity information (cf. Definition 6). On one hand, profile information can be generic (e.g., re-identification, age, gender, marital-status), or domain-specific, such as medical (e.g., disease, blood type, mental health), financial (e.g., bank account information), social (e.g., family, friends, associations), professional (e.g., salary, job), and so on. The user has also the possibility to define new inputs for each of these categories, as well as new domain categories. On the other hand, activity information are related to user habits, behaviors, performed activities, presence/absence, and so on. The user has also the possibility to define new inputs. For Alice, only the "Date of birth", "Age", "Marital Status", "Political Affiliation", and "Preference" are significant from the generic category, as well as the full list of the medical profile and activity information.

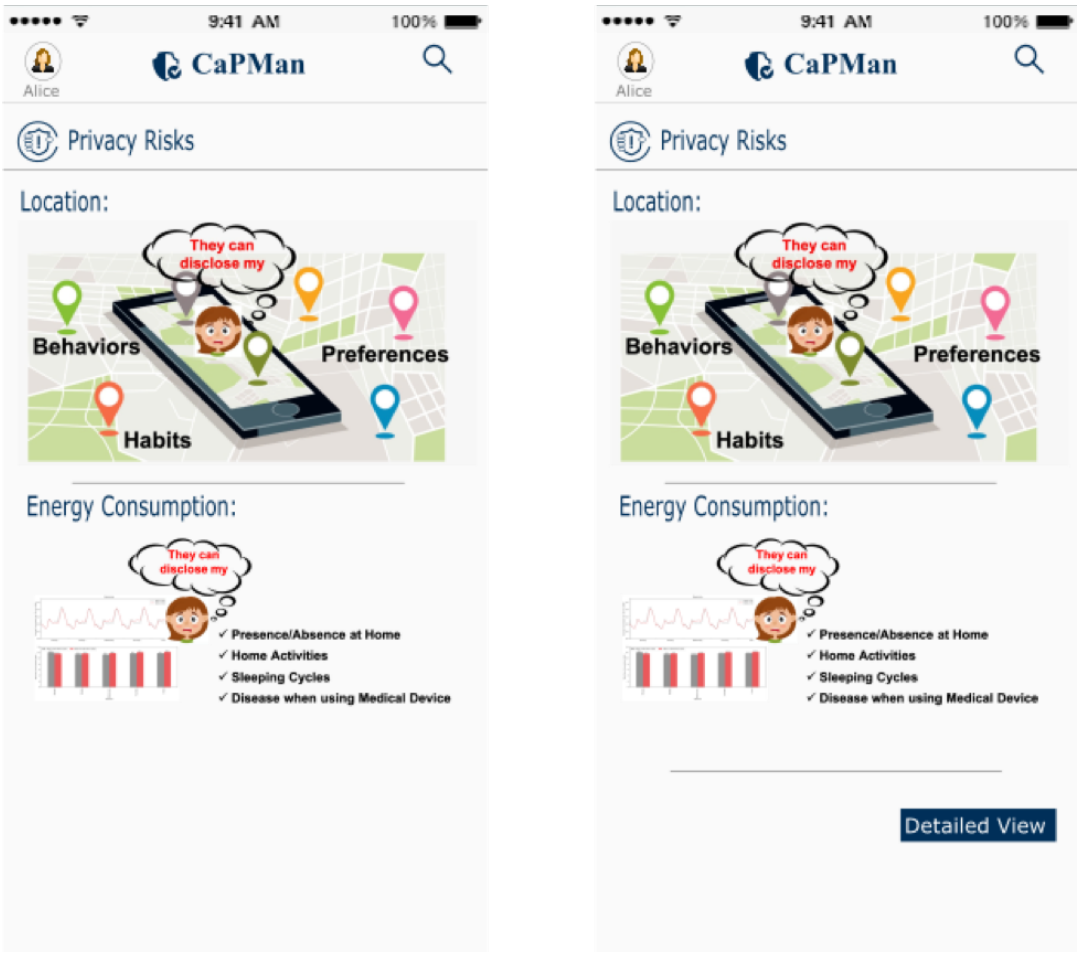

FIGURE 3.15: Privacy Risks Interface: Picture-based Warnings for a Beginner (left) and Intermediate/Advanced (right) user

The CaSPI reasoner is launched subsequently to detect the privacy risks involved in the current user context. Once done, the outputted risk overview, $\boldsymbol{R_c}$, is summarized into comprehensible picture-based warnings associated with each sensed data, and delivered to the user. Only an Intermediate or Advanced user has the possibility to access a detailed view of her risks by clicking on the "Detailed view" button. Figure 3.15 illustrates the picture-based warnings that have been sent to Alice in *context-1*, such that the figure on the left represents the Beginner view, and the one on the right represents the Intermediate/Advanced view depending on her profile.

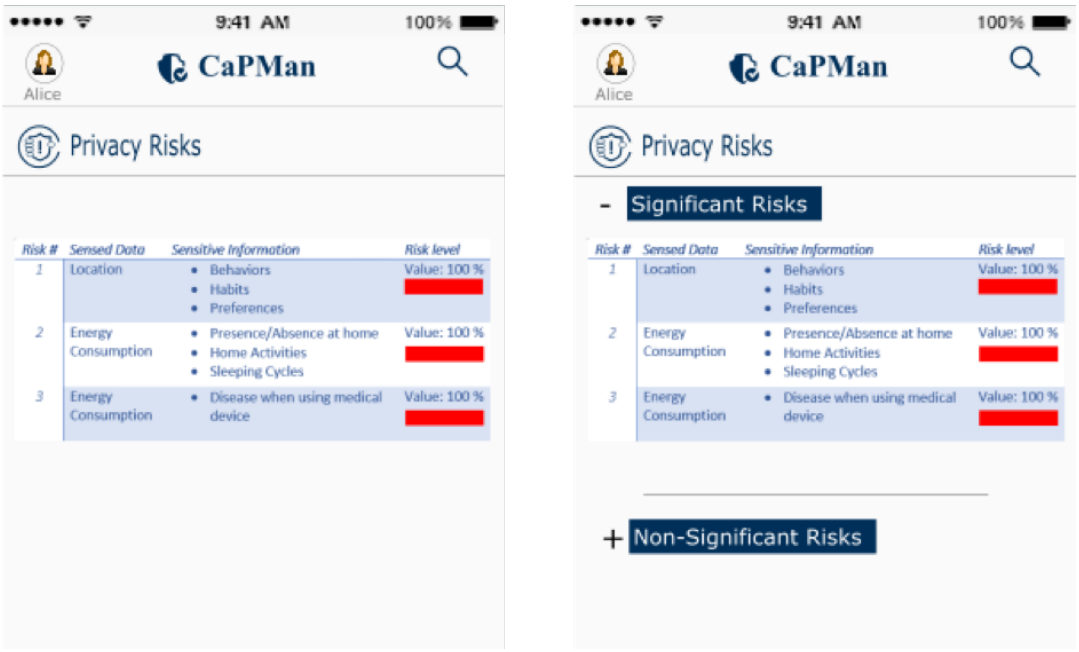

FIGURE 3.16: Privacy Risks Interface: Textual Warnings for an Intermediate (left) and Advanced (right) user

Upon clicking the "Detailed view" button, the user accesses a detailed view of her risks, presented as a table of four columns that details the content of $\boldsymbol{R_c}$ (i.e., risk identifiers, related sensed data, sensitive information, and risk values). As previously stated in Section 3.4.3, the Intermediate view includes only user significant risks, and the Advanced view includes two tables of respectively user significant and non-significant risks. Figure 3.16 illustrates the textual warnings that may be sent to Alice if her profile was Intermediate (the interface on the left) or Advanced (the interface on the right).

3.5.2 Performance Evaluation

The objective here is to evaluate the ability of the approach, performance-wise, to operate in various scenarios, including worst case ones, and to meet the needs of scalability and efficiency (in time and space) outlined in Challenge 5. To achieve this, we start by considering three cases that measure the impact of the following metrics on performance: (1) the number of privacy rules imported by the system for a single

reasoning iteration; (2) the number of risks to be detected in a single iteration; and (3) the size diversity of the user context. Then, we formally study the storage complexity of the proposal. The performance is evaluated based on two criteria: the total execution time and memory usage of one iteration. The tests are conducted on a machine equipped with an Intel i7 2.80 GHz processor and 16 GB of RAM. The chosen execution value for each scenario is an average of 10 sequenced values. We select the peak value of the in-use memory for each scenario when measuring memory usage.

Case 1: We study here the impact of privacy rules on performance by progressively increasing the number of rules imported by the system. We limit the context size to 100 individuals modeled with their relationships, and we consider a mixed list of satisfying/non-satisfying rules in the current context, such that only satisfying rules generate risks to the user. We execute the CaSPI reasoner 6 times, taking into account the following number of rules for each iteration: 1 (satisfying); 10 (5 are satisfying); 50 (25); 100 (50); 500 (100); and 1,000 (100). Figure 3.17 shows that the number of privacy rules has a quasi-linear impact on the total execution time, with an average of 2 s for 10 rules, 3.5 s for 50 rules, 5 s for 100 rules, and up to 27 s for 1000 rules. The evolution is similar for the RAM consumption (see in Figure 3.17), with an average of 150 MB for 10 rules, 170 MB for 50 rules, 210 MB for 100 rules, and up to 278 MB for 1000 rules. This consequently highlights the importance of filtering the list of rules before launching the risk reasoning process.

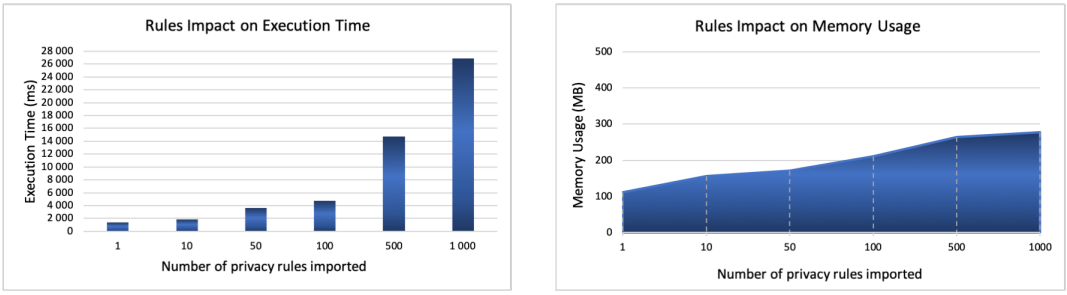

FIGURE 3.17: Privacy Rules Impact

Case 2: We investigate here the impact of the risk number (to be detected) on performance, by increasing the number of satisfying rules in the global pool of rules. We limit the context size to 100 individuals modeled with their relationships, and the pool of rules to 100. We execute the CaSPI reasoner 6 times, taking into account the following number of satisfying rules for each iteration: 1; 10; 30; 50; 70; and 100. Figure 3.18 shows that the amount of risks to be detected has no computation impact on performance. The time and RAM consumption are the same in all scenarios, with an average of respectively 5 s and 210 MB. This is due to the fact that the reasoner scans all rules one by one and generate the relevant inferences. Therefore, only the number of imported rules impacts the performance regardless of their satisfying status.

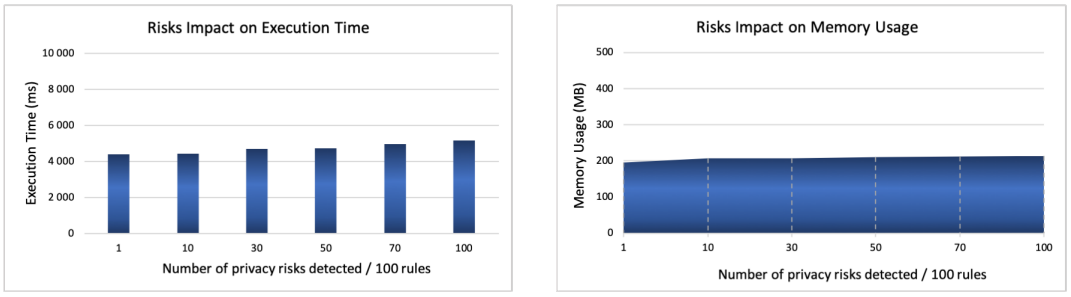

FIGURE 3.18: Privacy Risks Impact

Case 3: We evaluate here the influence of the context size on performance, in terms of number of ontology individuals with their relationships. To do so, we limit the pool of rules imported to 100, including 50 satisfying rules (i.e., 50 risks to be detected for the user). We execute the CaSPI reasoner 8 times, taking into account the following number of individuals for each iteration: 1; 10; 50; 100; 500; 1,000; 5,000; and 10,000. Figure 3.19 shows that the context size has a quasi-constant impact on the total execution time up to 1,000 individuals with an average of 5 s, and then the evolution becomes quasi-linear with an average of 8 s for 5,000 individuals and 12 s for 10,000. The evolution of the RAM usage is quasi-linear (see in Figure 3.19) with an average of 200 MB for 100 individuals and up to 830 MB for 10,000. This consequently underlines the ability of the CaSPI solution to assist the user in a variety of contexts, including ephemeral ones (i.e., contexts with short time periods).

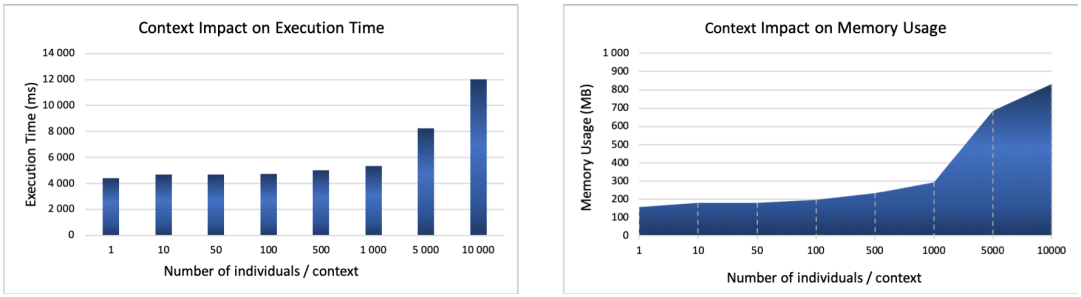

FIGURE 3.19: Context Diversity Impact

THEOREM 1. The CaSPI process maintains low storage complexity.

PROOF. Let i denotes the maximum number of individuals and relationships that express the user context, and p the maximum number of privacy rules imported by the system. The amount of storage space required by the system is increased with the increase of input value, $n = i + p$, resulting in a linear storage complexity of $O(n)$. Therefore, even in the worst case scenario of a large context size (e.g., 10,000 individuals) and a large number of rules (i.e., 1,000 rules), the system maintains a low storage complexity. □

Discussion. The experiments and studies conducted show that CaSPI is scalable, and maintains computational and storage efficiency (cf. Challenge 5). The solution

is capable of operating and assisting the user in different scenarios, including worst case ones. This increases its re-usability for a variety of applications, including those requiring real-time assistance, and allows it to operate on a variety of devices, including those with limited resources.

3.6 Summary

In this chapter, we present a Context-aware Semantic reasoning approach for Privacy risk Inference (CaSPI). The approach is equipped with a semantic rule-based reasoner that is used to infer the risks involved in user contexts. To achieve this, CaSPI relies on the use of ontologies (e.g., uCSN ontology) and inference rules that respectively represent contextual knowledge and define the risks to be detected by the reasoner with high semantic expressiveness power. In order to define the rules, we introduce a generic rule syntax that enables the combination of sensed/background data using basic and advanced operators (i.e., logical, comparison, spatio-temporal, and semantic operators), and considers various types of sensitive inferences (e.g., re-identification, sensitive profile or activity information). CaSPI is generic and reusable in several domains. It is capable of providing the user with a complete and dynamic overview of risks to cope with the dynamicity of her context. The risk overview is tailored to the user's expertise, allowing all users to understand their privacy situations. We developed a prototype to validate our proposal and we illustrated its functioning from both back-end and front-end perspectives. We also evaluated its performance by considering multiple cases. The results show that our approach is scalable and achieves efficiency in terms of computation and storage, even in worst-case scenarios. This increases its re-usability to support the user in different contexts.

Chapter 4

Privacy Risk Management

"What gets measured, gets managed."

– Peter Drucker

In today's highly connected environments (e.g., IoT environments), multiple systems collect, exchange, store, and process large amount of fine-granular data in every aspect of life. Such detailed data improve the delivery of advanced services across a wide range of application domains (e.g., smart homes, cities, e-health). However, the produced data is often privacy-sensitive for their users (e.g., location, blood pressure), and its analysis allows data consumers to deduce sensitive information about users, such as their behaviors, activities, preferences, and so on.

Therefore, users must be able to make appropriate data utility-privacy decisions based on their situations and interests, in order to meet their privacy needs while also maximizing the quality of services received in exchange for their data. However, involving users in the management of such trade-offs is challenging due to the: (i) variety of expertise levels of users to express their needs and preferences; (ii) dynamicity of user contexts and the privacy risks involved; and (iii) complexity of reducing privacy risks to meet user needs without compromising main services. This raises consequently the need for a solution that can assist users in optimizing their data privacy decisions. Nonetheless, such a solution must be adaptive, scalable and fast in order to support the user in various contexts.

To address these challenges, we propose in this chapter ***δ-Risk***, a user-centric multi-objective approach for context-aware privacy management in connected environments. Our approach features a new privacy risk quantification model to dynamically calculate and select the best data protection strategies for the user based on her situation and preferences. Computed strategies are optimal in that they seek to closely satisfy user preferences, while also maximizing data utility and minimizing the cost of protection. We implemented our proposed approach, evaluated its performance in various scenarios, and formally studied its effectiveness. The results show that *δ-Risk* delivers scalability and efficiency (performance-wise). It also provides the user with at least one best strategy per context.

4.1 Introduction

Advances in the fields of ubiquitous computing (e.g., Internet of Things), sensing technologies, and Big Data have allowed the fast evolution of smart connected environments. These environments are equipped with Cyber-Physical Systems (CPS), such as sensor networks, capable of collecting and exchanging data that could be later mined and processed in order to provide advanced services. Current CPS-based applications are impacting numerous application domains including medical (e.g., patient and elderly monitoring), building/housing (e.g., increasing occupants' comfort, optimizing energy consumption), environmental (e.g., monitoring air and water pollution levels), and so on.

Providing smart services requires collecting massive amounts of sensor data, which are spatio-temporal in nature [26], such as individual's location, patient's vital signs, and energy-consumption of user's home. However, collected data are often privacy-sensitive as their analysis exposes associated users to various privacy risks, such as the risks of disclosing their routines and habits, health conditions, behaviors, activities, preferences, and so forth [101], [102], [103], [104]. This can be harmful for users if their data/information is misused by providers, sold to interested third parties, or stolen by cybercriminals as providers are often victims of cyber-attacks that lead to data/information breaches. Therefore, involving users in the control and management of their data privacy is currently receiving tremendous attention from both legal and technical perspectives (e.g., [11], [13], [21], [24]).

Nonetheless, achieving effective user involvement requires improving their privacy decision-making. To do so, we first focused on raising user awareness of the privacy risks they face by proposing CaSPI in the previous chapter, which provides a dynamic context-based risk overview tailored to the user's expertise. This allows all users to understand their implicit, direct and indirect privacy risks, and paves the way to make informed data privacy decisions. However, the interests and privacy needs vary from one user to another, and thus their privacy decisions. For instance, a user may agree to take the risks and share fine-granular data in order to benefit from all the services received in exchange for her data; nonetheless, another user may need to reduce the risks but without compromising the main services for her.

Users might not always know the appropriate data protection measures to apply in their situations. Indeed, over-protective measures limit the utility of shared data to eliminate the risks, but could also downgrade the accuracy of services. Under-protective measures may improve the accuracy of services, but might also lead to privacy breaches. Therefore, optimizing data utility-privacy trade-offs according to user needs, interests and contexts remains a key challenge to tackle. What makes it more challenging is that user-decisions must sometimes be fast (i.e., in real-time), and users may have different levels of expertise to express their needs and preferences. Therefore, the proposed solution needs to: (i) tailor the guided assistance to

the user's expertise; (ii) adapt the optimal data utility-privacy decisions to cope with the dynamic nature of user contexts and preferences; and (iii) provide scalability and computational/storage efficiency, which allows it to assist the user in different contexts, and operate on a variety of devices, including those with limited resources.

To address the aforementioned needs and challenges, we propose in this chapter ***δ-Risk***, a user-centric multi-objective approach for context-aware privacy management in connected environments. Our approach is capable of assisting the user in optimizing her data utility-privacy decisions, by providing dynamic and optimal data protection strategies according to her context and preferences. Each of these strategies intends to minimize the user's risks in a way to meet her interests and privacy needs, while also maximizing data utility and minimizing the cost of data protection. To achieve this, the approach involves a new privacy risk quantification model, that is used to calculate and select the best protection strategies. These strategies are the best combinations of data protection levels in the relevant situation. Each level expresses the amount of protection to add to the data of a specific sensed attribute before being released to data consumers. The assistance provided by our approach is adapted to the selected user profile, which may vary from beginner, intermediate, to advanced. To validate our proposal, we developed a Java-based prototype and illustrated its functioning from both back-end and front-end perspectives. We also evaluated its performance in different scenarios, and formally studied its effectiveness in strategy identification. The results show that *δ-Risk* delivers scalability and efficiency. In addition, it is always capable of: (i) identifying all possible strategies that satisfy the relevant data utility-privacy trade-off; (ii) delivering the best strategies; and (iii) providing at least one best strategy per context.

The rest of the chapter is organized as follows. Section 4.2 illustrates a scenario that motivates our proposal and identifies the challenges to tackle. Section 4.4 details our *δ-Risk* proposal and provides formal definitions of the key terms. Section 4.5 outlines the implementation phases and experimental protocol. Finally, Section 4.6 summarizes the chapter.

4.2 Motivating Scenario

We consider again the scenario describing Alice's situation, represent the associated overview of privacy risks, and consider a variety of interests/requirements for Alice, in order to highlight the need for dynamic adaptation of data utility-privacy decisions based on changes in the user context and preferences.

First, we remind the reader that Alice is a COPD patient and shares fine-granular energy consumption and location data with an electricity and a healthcare provider respectively, as illustrated in Figure 4.1. She receives several services in exchange

for her data, which are respectively the list of personalized recommendations to reduce her energy consumption and bills, and other healthcare services (e.g., smart ambulance service).

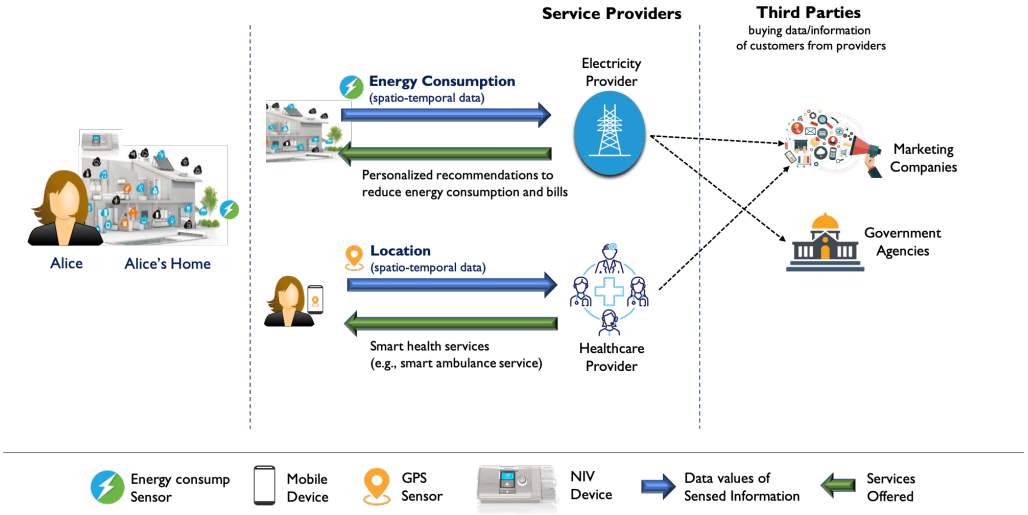

FIGURE 4.1: Alice's Situation

When launching the risk reasoner over modeled context information, Alice receives an overview of the privacy risks involved in her situation as shown in Figure 4.2. Consider that this overview includes three risks, the first is associated with the sharing of location data, and the other two with the sharing of energy-consumption data. All risks are at their highest level (i.e., their values are 100%) as the associated data pieces are shared in their fine-granular version (i.e., without protection).

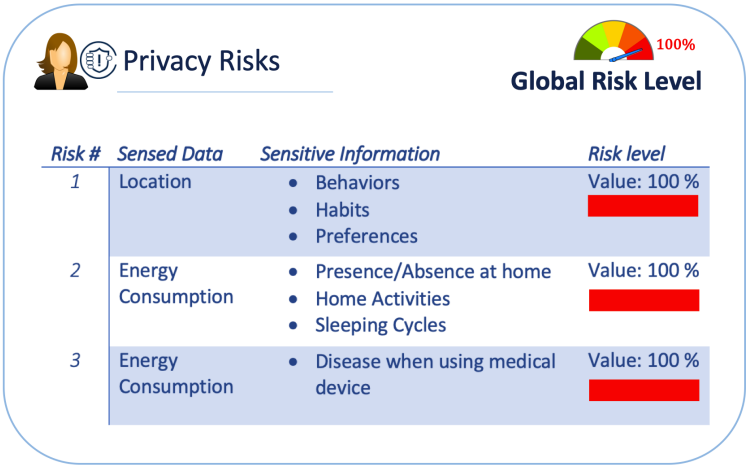

FIGURE 4.2: Alice's Privacy Situation

Once Alice is alerted, she may want to adapt her data privacy measures to reduce the risks. Nonetheless, such an adaptation can be difficult for her as it also affects the data utility, and thus the quality of associated services, which might be important to her as well. For instance, stop sharing her location data can lead to eliminate

risk-1, but also to lose the health services received in exchange. Therefore, assisting the user in optimizing data utility-privacy decisions according to her situation and preferences becomes essential. However, when considering such assistance, the following needs emerge:

Need 1. Coping with data diversity. The user can share a variety of sensed data with data consumers, which could be diverse in terms of attributes (e.g., location, temperature, camera recordings) and types (e.g., scalar and multimedia data). The risk manager should be capable of determining the appropriate levels of protection to assign to the data of diverse attributes when optimizing data privacy strategies.

Need 2. Quantifying privacy risks and the global risk level. The risk manager should be able to measure the impact of data protection on the risk values, and quantify the resulting global risk level for the user. This helps in optimizing the amount of protection to add in order to meet user preferences while also maximizing the quality of associated services.

Need 3. Coping with the diversity of user preferences. The user preferences can be related to three different aspects: data privacy protection, risk level, and service importance. For example, Figure 4.3 describes three cases of preference specification by Alice. In case-1, Alice wants to have a full privacy protection. In case-2, Alice wants also to preserve the full quality of her health services. In case-3, Alice requires also to reduce the global risk level to 50%. Therefore, the risk manager should be capable of adapting the strategies to satisfy all user preferences of different aspects.

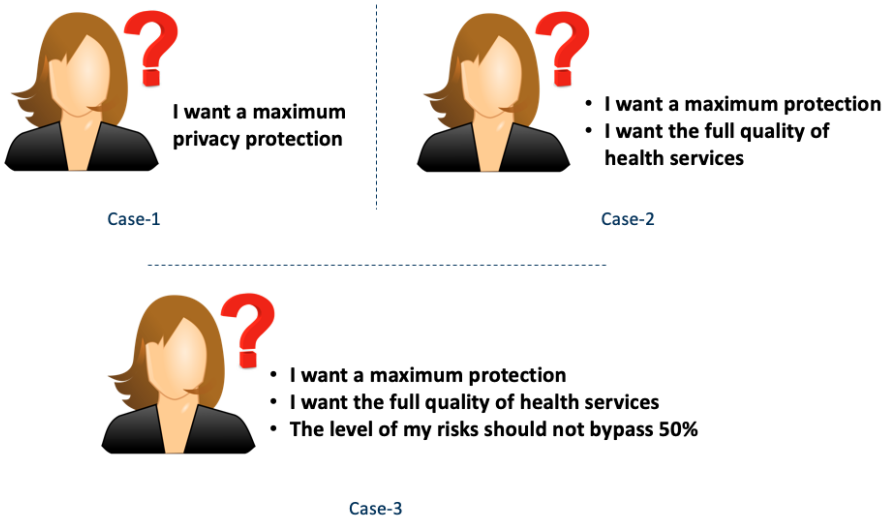

FIGURE 4.3: Privacy Needs and Interests of Alice

Need 4. Coping with protection function diversity and changes. The protection function executed on data pieces may vary from one attribute to another (e.g., random-noise function, generalization function). As well, the function assigned to an attribute may change from one context to another. However, each protection function

has its computational cost that impacts the overall computational cost of the data protection process. Therefore, the risk manager should be able to consider the costs of associated protection functions when optimizing the data protection strategies in order to minimize the global cost of protection.

Need 5. Responding to user-time constraints. User decisions must sometime be fast (i.e., in real-time). The risk manager should consequently be fast when identifying the optimal data protection strategies.

However, when considering the aforementioned needs, the following challenges emerge:

Challenge 1. ***Coping with user expertise:*** People may have different levels of expertise to properly express their preferences and interact with the system. The proposed solution must therefore be user-friendly, allowing the guided assistance to be tailored to the user's expertise in order to maintain good quality of human-machine interactions.

Challenge 2. ***Dealing with the dynamicity and context-dependency of data protection strategies:*** As user context changes, new privacy risks may emerge, while others may become negligible. As well, the user's preferences can change depending on her situation. Therefore, the proposed solution should always be capable of providing adaptive optimal data protection strategies to cope with the dynamicity of the user's context and preferences.

Challenge 3. ***Delivering scalability and efficiency:*** The solution must be scalable, i.e., handles reasoning over an increasing number of sensed attributes and privacy risks. It should also maintain computational and storage efficiency in order to support the user in various contexts, and be operational on different types of devices, including resource-constrained ones.

4.3 Data Privacy Background

Data privacy has received extensive attention over the last decade. Existing functions for data protection vary from data perturbation to data restriction. Figure 4.4 illustrates a proposed classification of data protection functions, based on their perspective of protection. This classification consists of two major categories: data perturbation and data restriction functions. Data perturbation functions focus on modifying original data by either hiding sensitive parts of it leading to user re-identification, or distorting its value by injecting noise [118]. Accordingly, this category regroups anonymization and noise-addition techniques. Anonymization functions (e.g., k-Anonymity [119]), l-Diversity [120], t-Closeness [121], CASTLE [122]) focus on dissociating the link between data and related data owner, and preserving the full utility of the data value. To do so, anonymization operations mask the

owner's identity from the data by removing explicit identifiers, and decreasing the granularity of quasi-identifiers using operations such as generalization and suppression [118], [123]. Noise-addition functions focus on distorting the value of the original data by injecting additive noise. This impacts the utility of the data value, but preserves the link between data and related owner (this link is critical if the owner receives services in exchange for data). Random-noise [124], generalization [125], data swapping [126], and differential privacy [127] are examples of noise-addition functions.

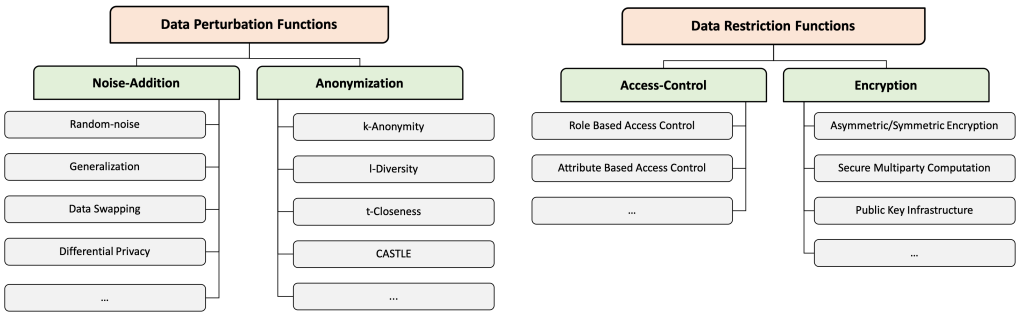

FIGURE 4.4: Classification of Data Protection Functions

Data restriction functions aim at limiting data use by blocking access or encrypting inputs. This is achieved by either applying access-control or encryption operations to data pieces. Therefore, data restriction category includes access-control and encryption techniques. Access-control functions (e.g., [128]–[130]) achieve privacy protection through authorization models and access control policy operations. They focus on limiting access to owner's data by enabling only authorized parties to read and/or manipulate data. Encryption functions (e.g., [128], [131]) vary from (i) secure multiparty computation (SMC) functions, focus on aggregating inputs of distributed entities to produce outputs while preserving the privacy of inputs; (ii) asymmetric/symmetric encryption functions, use encryption keys to protect released data; to (iii) public key infrastructure (PKI) functions, focus on delivering certificates to communicating entities in order to secure the identification process.

4.4 *δ-Risk* Proposal

In order to address the needs and challenges stated in Section 4.2, we propose in the following ***δ-Risk***, a new user-centric multi-objective approach for context-aware privacy risk management in connected environments. δ is a privacy parameter that expresses the risk threshold, i.e., the maximum level of risk that the user accepts to take in her relevant situation. The objective of this approach is to assist the user in optimizing her data utility-privacy decisions, in a way to meet her preferences while also maximizing data utility and minimizing the cost of protection. Accordingly, *δ-Risk* provides a list of best protection strategies from which the user selects one to implement in her relevant situation. In addition to her privacy preferences, the

approach considers also the interests of the user (e.g., which services are important to her), thereby making the strategies provided not only optimal but also meaningful.

Figure 4.5 illustrates an overview of our proposal, including related inputs and outputs. *δ-Risk* receives as input:

(1) The set of *sensed attributes* in the relevant context (cf. Definition 5.1 in Section 3.4.1), $c.SA = \{a_1; a_2; \ldots; a_m\} \mid m \in \mathbb{N}$

(2) The lists of *user preferences*, which vary from privacy to service preferences (detailed in the following subsection)

(3) The *overview of privacy risks* in the relevant context, $R_c = \{\vec{r}; v\}$, such that:

- $\vec{r} = \begin{bmatrix} r_1 & r_2 & \ldots & r_n \end{bmatrix} \mid n \in \mathbb{N}$ is a ***risk vector*** representing the privacy risks involved in c.
- v expresses the ***global risk level*** that the user takes in c. $R_c.v$ is used to interact with the δ value. Its quantification is detailed in Section 4.4.3.1.

The *impact matrix*, W_c, representing the impact value of sensed attributes on the risks inferred (cf. Definition 9)

(4) The list of *protection functions* (cf. Definition 10) selected by the *data protection* module of CaPMan to be executed on data values of sensed attributes.

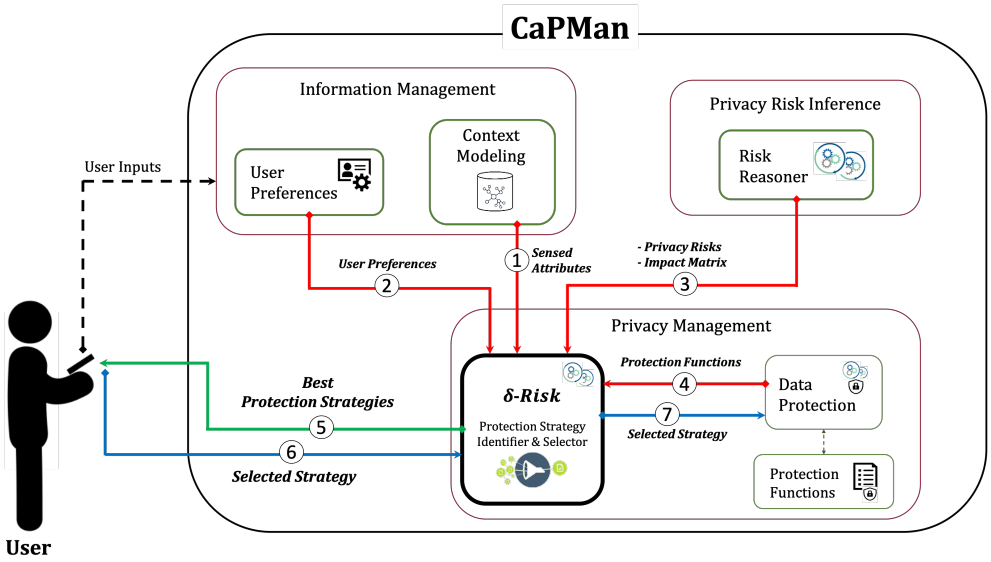

FIGURE 4.5: Overview of the *δ-Risk* proposal

Consequently, *δ-Risk* outputs (5) the list of best data protection strategies that might be adopted in the relevant situation. The user (6) selects one of these strategies to be implemented, which remains valid as long as no changes occur in the entries. Finally, (7) *δ-Risk* transmits the chosen strategy to the *data protection* module

of CaPMan, which is responsible for protecting sensed data values prior to its release to data consumers. The δ-***Risk*** **principle** is defined as follows: the global risk level to maintain in the user context should not bypass the threshold δ. The δ value can be fixed directly by the user (if she wants to limit the maximum level of risk), or automatically computed when executing the risk manager in a way to maximize the user's privacy protection (i.e., δ is fixed at the lowest-possible value that satisfies user preferences in the current situation). In what follows, we formally define an *impact matrix* and a *protection function*.

Definition 9 (Impact Matrix). Let W_c be the ***impact matrix***, expressing the impact status of sensed attributes $\{a_1, a_2, ..., a_m\}$ of $c.SA$ on risks $\{r_1, r_2, ..., r_n\}$ of $R_c.\vec{r}$. W_c is automatically identified by the *risk reasoner* module of CaPMan after performing the risk reasoning process, such that:

$$W_c = \begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1m} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{n1} & \omega_{n2} & \dots & \omega_{nm} \end{bmatrix} \text{, where } \omega_{ij} = \begin{cases} 0 & \text{if } a_j \notin r_i.SA' \\ 1 & \text{if } a_j \in r_i.SA' \end{cases}$$

We remind the reader that according to Definition 8, $r : \langle\ id\ ;\ value\ ;\ SI'\ ;\ SA'\ \rangle$. The impact status ω_{ij} of an attribute a_j on a risk r_i is therefore equal to 1 only if a_j is included in the set of attributes associated with r_i. ■

Example 8. Alice is taking 3 risks in her situation, where the first is associated with her location data and the two others with the energy-consumption data of her home. The risk vector, set of sensed attributes, and impact matrix are consequently represented as follows:

$$R_c.\vec{r} = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} \ ;\ c.SA = \{a_1\ ;\ a_2\} \ ;\ W_c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

where a_1 (i.e., Location) impacts only r_1, and a_2 (i.e., energy-consump) impacts r_2 and r_3.

Definition 10 (Protection Function). A ***protection function***, $f \in PF$, is a protection method that can be executed on data values of an attribute $a \in c.SA$ prior to their release to data consumers. f is a local function stored in the CaPMan system, such that:

$$f : \langle\ name\ ;\ categ\ ;\ Feature\ ;\ Param\ \rangle, \text{ where:}$$

- ***name*** denotes the textual name of f (e.g., generalization, random-noise)
- ***categ*** represents the category to which f belongs, such that:

$$\boldsymbol{categ} \in \{\textit{noise-addition ; anonymization ; access-control ; encryption}\}$$

- $\boldsymbol{Feature}$ is the set of features characterizing $\boldsymbol{f}$, including at least:
  - $\boldsymbol{cost}$, the computational cost of $\boldsymbol{f}$ in terms of processing time and memory overhead
- $\boldsymbol{Param}$ represents the set of input parameters of $\boldsymbol{f}$, including at least:
  - $\boldsymbol{SA' \sqsubseteq SA}$ is the set of attributes to which $\boldsymbol{f}$ is associated
  - $\boldsymbol{P}$ is the set of protection levels to achieve for the data of attributes in $\boldsymbol{SA'}$ ■

4.4.1 User Preferences

As previously stated, user preferences can range from privacy to service preferences. From privacy standpoint, preferences can be related to: (i) the level of risk that the user accepts to maintain; or (ii) the level of data protection for specific attributes (e.g., 80% protection on location data). From service standpoint, the user can specify her preferences regarding the services important to her, allowing the risk manager to maximize the quality of the main services when optimizing the strategies. The preferences can be represented as follows:

- **Privacy Preferences**:
  1. The *risk threshold* δ, with a value between 0 and 1, where 0 indicates that the user $\boldsymbol{u}$ requires maximum protection and does not accept to take any risk, and 1 means that $\boldsymbol{u}$ accepts to take all risks and share fine-granular data to preserve the full quality of services.
     If the user does not specify a value for δ, this means that she wants to maximize her privacy protection while also considering the other preferences. δ should be accordingly fixed at the lowest possible value.
  2. The *data protection levels* enforced for specific sensed attributes. Users can manually enforce specific protection levels to achieve for the data of related attributes prior to its release. Let $\boldsymbol{eP}$ denotes the *set of enforced data protection levels*. $\boldsymbol{eP}$ can also include protection levels extracted from pre-signed agreements with data consumers. It can be represented as follows:

$$\boldsymbol{eP = \{ep_1;\ ep_2;\ \ldots;\ ep_m\}} \quad \text{, where:}$$
$$\boldsymbol{\forall i \in [1;m],\ \ ep_i \in [0;1]\ \ and\ \ \exists!\ a_i \in c.SA\ :\ ep_i \sim a_i}$$

- **Service Preferences**:
  3. The *services important* to the user, expressed by a value of 1 (or 0 if not). Each of the services is associated with one or more sensed attributes. Let $\boldsymbol{S}$ denotes the *set of services* offered to the user in exchange of her data:

$$\boldsymbol{\forall s \in S,\ \ s : \langle\ SA'\ ;\ li\ \rangle} \quad \text{, where:}$$

- $SA' \sqsubseteq c.SA$ represents the set of sensed attributes associated with s
- li denotes the level of importance of s to u. li has a primitive Boolean data type with a value of 1 if s is important and 0 if not

4.4.2 User Profiles

Even though the approach is capable of considering a variety of user preferences, people may have different levels of expertise to interact with the system, such as to correctly express their preferences and/or select one of the best strategies to implement (cf. Challenge 1). In order to overcome this issue, we extend the user profiles defined in Chapter 3 (cf. Section 3.4.3) in order to tailor the assistance provided to the user's expertise. We remind the reader of the three profiles defined:

- **Beginner**: The user is not familiar with her privacy, she does not know how to specify her preferences, interpret her risks and the protection strategies provided.
- **Intermediate**: The user understands how to specify her preferences related to sensitive inferences and services personalization, as well as limiting the maximum level of risk when needed. However, she only requires a detailed overview of the significant risks to her, and a short list of best protection strategy options.
- **Advanced**: The user is expert in specifying her preferences, interpreting and analyzing her privacy situation. She can ask for full details about the significant and non-significant risks involved in her situation, as well as more options of best protection strategies to choose from.

Figure 4.6 details the extended characteristics of the user profiles, which are related to preference specification and protection strategy selection. The goal here is to limit the level of user-interaction with the system, as well as the bunch of information provided, according to her profile. The level of user-interaction is expressed by a min-max number in Figure 4.6. We only discuss in what follows the additional characteristics for each profile. For all profiles, δ is by default free (i.e., not specified), which means that the user wants to maximize privacy protection. However, a *beginner* has the option to manually specify a value of 0 or 1 for δ only during a fixed time period, otherwise the system proceeds with the default δ (i.e., maximize protection). The enforced data protection levels are only extracted from pre-signed agreements with data consumers (if exist). Finally, the system selects automatically one of the best protection strategies without requiring user intervention. An *intermediate* can manually specify a value for δ ranging from 0 to 1, and has the option to personalize important services to her. The enforced data protection levels are only extracted from pre-signed agreements with data consumers (if exist). Finally, an *intermediate* has to select one of the K-best strategies provided by the system during a fixed time period, otherwise the system selects one of these strategies and implements it. For an *advanced*, the system provides all *intermediate* options plus the possibility to enforce specific data protection levels for her sensed attributes.

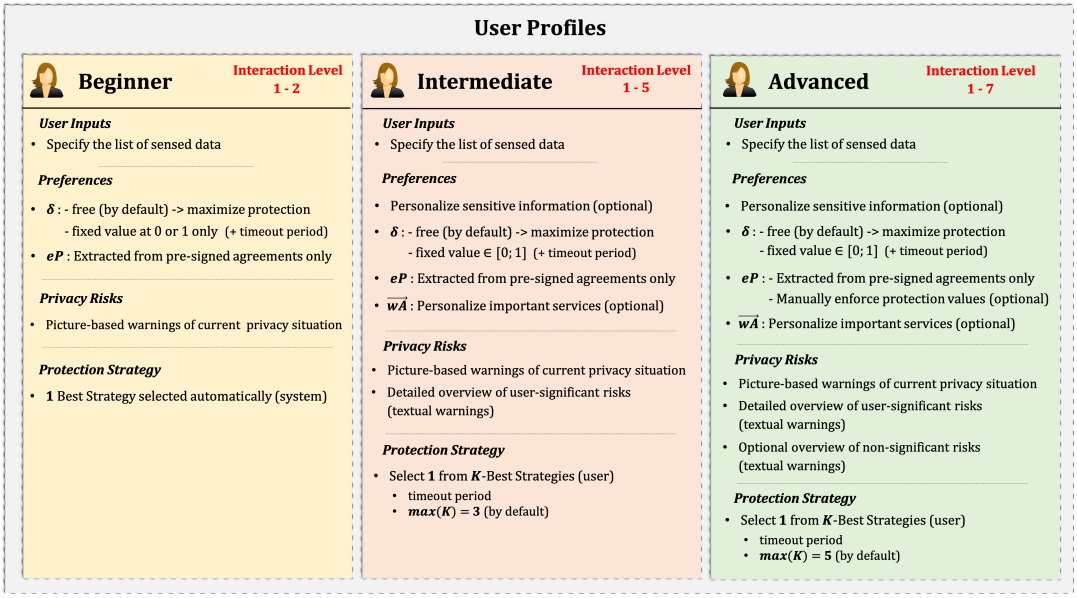

FIGURE 4.6: Extended Characteristics of User Profiles

4.4.3 *δ-Risk* Operations

After collecting user preferences, computing adaptive optimal data protection strategies to cope with the dynamicity of the user's context and preferences becomes a challenging endeavor (cf. Challenge 2). To address this challenge, the *δ-Risk* process consists of two operations: protection strategy identification and best strategy selection. Before detailing these operations, we start by formally defining a *protection strategy*, *data protection level*, and a *best protection strategy*.

Definition 11 (Protection Strategy). A ***protection strategy***, $\vec{p} \in P_c$, is a protection vector composed of an appropriate combination of *data protection levels* $p_1, p_2, ..., p_m$ to be achieved for data of attributes $\{a_1; a_2; \ldots; a_m\}$ of $c.SA$. Appropriate means a combination that meets the privacy preferences of the user u (i.e., δ and eP) while maximizing data utility. $\vec{p} \in P_c$ can be represented as follows:

$$\vec{p} = \begin{bmatrix} p_1 & p_2 & \ldots & p_m \end{bmatrix} \text{ , where:}$$

$$\forall i \in [1;m], \quad p_i \in [0;1] \quad and \quad \exists! \, a_i \in c.SA \; : \; p_i \sim a_i \qquad \blacksquare$$

Definition 12 (Data Protection Level). A ***data protection level***, p, expresses the amount of protection to be achieved for the data values of an attribute $a \in c.SA$. p is probabilistic with a value between 0 and 1, where 0 means that data is shared in fine-granular version (i.e., without any protection), and 1 means that data is not shared (i.e., highest level of protection). A value between 0 and 1 indicates the level of protection that should be reached when executing a *protection function* $f \in PF$ on the data of a. Knowing that the way to achieve p depends on the selected *protection function*. $\blacksquare$

Definition 13 (Best Protection Strategy). A ***best protection strategy***, $\vec{bp} \in BP_c$, is an appropriate strategy $\vec{p} \in P_c$, that also satisfies the service preferences of u (expressed by $\vec{wA}$), and has the lowest cost of protection (i.e., based on the corresponding combination of *protection functions*). These constraints are expressed by the *ranking score* assigned to $\vec{p}$, which is computed as follows:

$$score(\vec{p}) = Rank(\vec{p}, \vec{wA}, cPF) \rightarrow \mathbb{N} \text{ , where:}$$

- $\vec{wA} = \begin{bmatrix} wa_1 & ... & wa_m \end{bmatrix}$ denotes the vector of weights assigned to attributes $\{a_1; \ldots; a_m\}$ of $c.SA$. Each wa_i of $\vec{wA}$ expresses the weight of attribute $a_i \in c.SA$, which is calculated based on the service preferences of the user. wa_i is equal to the number of important *services* from the set $S = \{s_1; \ldots; s_n\}$ to which a_i is associated:

$$\forall\, i \in [1; m],\ \ wa_i = \sum_{k=1}^{n} \alpha_k \ \mid\ \alpha_k = \begin{cases} 0 & \text{if } a_i \notin s_k.SA' \\ s_k.li & \text{if } a_i \in s_k.SA' \end{cases}$$

- cPF represents the set of *costs of the protection functions* selected by the system to be executed on attributes of $c.SA$.

- $Rank()$ expresses the ranking function. It takes as input a *protection strategy* $\vec{p} \in P_c$, the *vector of weights*, $\vec{wA}$, and the *set of costs of selected protection functions*, cPF. It outputs the *ranking score* of $\vec{p}$ that is calculated according to the distance between $\vec{p}$ and $\vec{wA}$, and the costs of the combined *protection functions*. Algorithm 4 details the $Rank()$ function.

Therefore, $\vec{p}$ is said to be one of the *best protection strategies*, $\vec{bp} \in BP_c$, only if it has the highest ranking score:

$$\forall\, \vec{p_i} \in P_c\ :\ \vec{p_i} \models \vec{bp}\ \ \textit{only if}\ \ \forall \vec{p_j} \in P_c,\ score(\vec{p_j}) \leq score(\vec{p_i}) \qquad \blacksquare$$

Figure 4.7 details the δ-*Risk* process. The first operation consists of identifying all possible *protection strategies* (i.e., P_c):

- If no strategies result from this operation, this means that the combination of the privacy preferences (i.e., δ and eP) is inconsistent (cf. Definition 14). In this case, the system asks u to change one of these preferences and assigns a timeout period for this query: (1) if u fails to respond before the timeout expires, the system releases the value of δ, which leads to maximize the user-privacy protection; (2) otherwise, the first operation is re-launched while considering user changes.

- If this operation generates several *protection strategies*, the second operation proceeds with ranking the resulting strategies using the $Rank()$ function, and selecting the K-*best strategies* to be proposed to u.

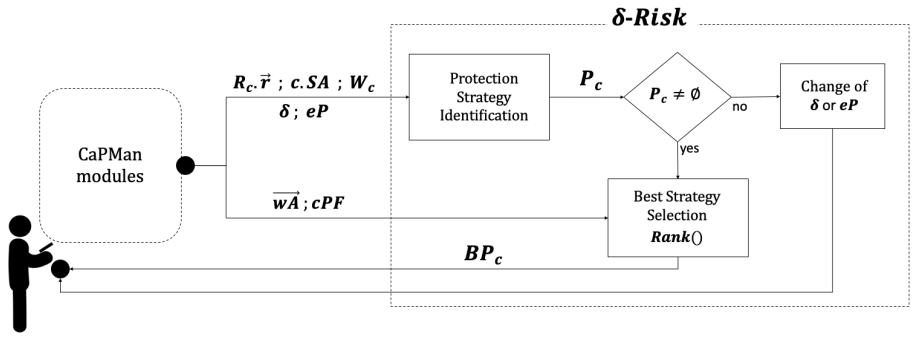

FIGURE 4.7: *δ-Risk* Operations

The *δ-Risk* process is by default executed once per user context unless a full context similarity is detected (cf. Definition 4.1) and the user has chosen to re-apply the actions of the previous similar context. However, within the same context, the user may change her service preferences or the system may select new protection functions, which requires recalculating new best strategies. To handle this while reducing the computational overhead caused by the relaunch of the global process, the system locally stores the *protection strategies* identified by the first operation (i.e., $\boldsymbol{P_c}$) as long as no context change occurs. Therefore, if $\vec{\boldsymbol{wA}}$ or $\boldsymbol{cPF}$ has been changed within the same context, only the second operation is re-executed to select new best strategies that cope with these changes.

Example 9. In order to illustrate the functioning of the process, assume that the first operation generates the following two strategies:

$$P = \begin{bmatrix} \vec{p}_1 \\ \vec{p}_2 \end{bmatrix} = \begin{bmatrix} 0.3 & 0.6 \\ 0.6 & 0.3 \end{bmatrix}$$

Assume also that attributes a_1 and a_2 have the same weight, and the cost of the protection functions associated with a_1 and a_2 are respectively 2 and 1. When executing the Rank() function (detailed in Section 4.4.3.3), the score of $\vec{p}_2$ will be higher than $\vec{p}_1$. $\vec{p}_2$ will be therefore selected as the best strategy, which suggests applying 60% protection on data of a_1 and 30% on data of a_2.

Determining appropriate combinations of data protection levels requires first to quantify privacy risks in order to study the impact of these levels on risk values; then, to quantify the global risk level (i.e., $\boldsymbol{R_c.v}$) in order to ensure that the resulting combinations satisfy the *δ-Risk* principle. Therefore, we begin by formally quantifying a *privacy risk* and the *global risk level*. Then, we detail the two *δ-Risk* operations.

4.4.3.1 Privacy Risk & Global Risk Level Quantification

Privacy risks (i.e., direct risks) have one or more associated sensed attributes. This means that increasing the protection of attributes' data will lead to minimize the risk values. Consequently, the risk vector $\vec{\boldsymbol{r}}$ depends on the protection levels assigned to

sensed attributes, $\vec{p}$, and the impact matrix of attributes on risks, W_c. This can be represented as follows:

$$\vec{r} = \mathcal{F}(W_c\ ;\ \vec{p}) \tag{4.1}$$

Where:

- $\mathcal{F}$ is the risk quantification function, which takes an impact matrix and a protection vector as parameters, and returns the risk vector with the calculated risk values.

$$\begin{bmatrix} r_1.value \\ r_2.value \\ \vdots \\ r_n.value \end{bmatrix} = \mathcal{F}\left(\begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1m} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{n1} & \omega_{n2} & \dots & \omega_{nm} \end{bmatrix} \ ;\ \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix} \right)$$

Before exploring the risk quantification function ($\mathcal{F}$), we define the assumptions to consider:

1. A (direct) privacy risk has at least one impacting shared attribute $a_j \in c.SA$. This means that:
$$\forall\ \vec{w}_i \in W_c,\ \sum_{j=1}^{m} \omega_{ij} \neq 0$$

2. If no protection assigned to attributes impacting r_i, the risk value, $r_i.value$, is equal to 1 (i.e., highest level).

3. If the full protection is assigned to attributes impacting r_i, r_i is negligible (i.e., $r_i.value = 0$).

4. The higher the protection level p_j impacting r_i, the lower the value of r_i.

Attributes may have different values of impact on a single risk, ranging from 0 to 1. For example, two attributes may impact a same privacy risk, however, the impact value could be 40% for the first and 60% for the second attribute. The challenge of determining the impact value of a specific attribute on a risk is addressed in future work. In order to simplify the process, we consider in this study that attributes have the same value of impact on a privacy risk (e.g., if two attributes are impacting the risk, their impact values are equal to 0.5). Accordingly, we proceed with identifying the impact values of attributes on risks according to the *impact matrix* W_c. Let $\widetilde{W_c}$ denotes the ***matrix of impact values***. $\widetilde{W_c}$ is computed as follows:

$$\widetilde{W_c} = \begin{bmatrix} \widetilde{\omega_{11}} & \widetilde{\omega_{12}} & \dots & \widetilde{\omega_{1m}} \\ \widetilde{\omega_{21}} & \widetilde{\omega_{22}} & \dots & \widetilde{\omega_{2m}} \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{\omega_{n1}} & \widetilde{\omega_{n2}} & \dots & \widetilde{\omega_{nm}} \end{bmatrix} \mid \forall i \in [1,n], \forall j \in [1;m], \ \widetilde{\omega_{ij}} = \frac{\omega_{ij}}{\sum_{k=1}^{m} \omega_{ik}} \tag{4.2}$$

Privacy risks are therefore quantified as follows:

$$\vec{r} = \mathcal{F}(W_c \, ; \, \vec{p})$$

$$\vec{r} = 1 - (\widetilde{W_c} \times \vec{p}) \tag{4.3}$$

$$\begin{bmatrix} r_1.value \\ r_2.value \\ \vdots \\ r_n.value \end{bmatrix} = 1 - \left(\begin{bmatrix} \widetilde{\omega_{11}} & \widetilde{\omega_{12}} & \dots & \widetilde{\omega_{1m}} \\ \widetilde{\omega_{21}} & \widetilde{\omega_{22}} & \dots & \widetilde{\omega_{2m}} \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{\omega_{n1}} & \widetilde{\omega_{n2}} & \dots & \widetilde{\omega_{nm}} \end{bmatrix} \times \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix} \right)$$

Example 10. According to Examples 8 and 9, the best strategy delivered to Alice in her situation is $\vec{bp} = \begin{bmatrix} 0.6 & 0.3 \end{bmatrix}$*. Once implemented, the risk values will be minimized to:*

$$\begin{bmatrix} r_1.value \\ r_2.value \\ r_3.value \end{bmatrix} = 1 - \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0.6 \\ 0.3 \end{bmatrix} \right) = \begin{bmatrix} 0.6 \\ 0.3 \\ 0.3 \end{bmatrix}$$

After quantifying the *privacy risks*, we now focus on how to measure the *global risk level* in the user context, $R_c.v$. This level is used to interact with the value of δ in order to determine whether or not it satisfies the *δ-Risk* principle. Accordingly, $R_c.v$ is equal to the maximum value of risk in $R_c.\vec{r}$. This can be formalized as follows:

$$R_c.v = max \begin{bmatrix} r_1.value \\ r_2.value \\ \vdots \\ r_n.value \end{bmatrix} \mid R_c.v \in [0,1] \tag{4.4}$$

4.4.3.2 Protection Strategy Identification

We detail in this section the first *δ-Risk* operation, which consists of identifying the appropriate *protection strategies* (cf. Definition 11) that could be implemented in the relevant user's situation. To achieve this, we start from the *δ-Risk* principle, which

states that the global risk level to maintain (i.e., $R_c.v$) should not bypass the threshold δ. Accordingly:

$$R_c.v \leqslant \delta \tag{4.5}$$

$$\Rightarrow max \begin{bmatrix} r_1.value \\ r_2.value \\ \vdots \\ r_n.value \end{bmatrix} \leqslant \delta$$

$$\Rightarrow \begin{bmatrix} r_1.value \\ r_2.value \\ \vdots \\ r_n.value \end{bmatrix} \leqslant \delta$$

However, maximizing the utility of attributes' data requires assigning the lowest-possible protection levels to these data. These levels are obtained when minimizing risks to the highest acceptable values. Therefore, optimizing the data utility-privacy trade-off necessitates considering only the combinations of data protection levels that satisfy $R_c.\vec{r}.value = \delta$. This results in the following linear system of n equations with m unknowns:

$$\begin{bmatrix} r_1.value \\ r_2.value \\ \vdots \\ r_n.value \end{bmatrix} = \delta$$

$$\Rightarrow 1 - \left(\begin{bmatrix} \widetilde{\omega_{11}} & \widetilde{\omega_{12}} & \dots & \widetilde{\omega_{1m}} \\ \widetilde{\omega_{21}} & \widetilde{\omega_{22}} & \dots & \widetilde{\omega_{2m}} \\ \vdots & \vdots & \dots & \vdots \\ \widetilde{\omega_{n1}} & \widetilde{\omega_{n2}} & \dots & \widetilde{\omega_{nm}} \end{bmatrix} \times \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix} \right) = \delta$$

$$\Rightarrow \begin{cases} \widetilde{\omega_{11}}.p_1 + \widetilde{\omega_{12}}.p_2 + \dots + \widetilde{\omega_{1m}}.p_m = 1 - \delta \\ \widetilde{\omega_{21}}.p_1 + \widetilde{\omega_{22}}.p_2 + \dots + \widetilde{\omega_{2m}}.p_m = 1 - \delta \\ \vdots \quad\quad \vdots \quad\quad \vdots \quad\quad \vdots \\ \widetilde{\omega_{n1}}.p_1 + \widetilde{\omega_{n2}}.p_2 + \dots + \widetilde{\omega_{nm}}.p_m = 1 - \delta \end{cases} \tag{4.6}$$

To solve the resulted system, we use the **Gauss-Jordan Elimination** (GJE) method, an implicit pivoting strategy that performs row operations to convert a matrix into a reduced row echelon form [132]. This method has been widely used in various domains to solve systems of linear equations, such as for traffic control management [133], image change and climate prediction [134], [135], cluster and grid computing [136], [137], and location privacy [138]. Solving the linear system using the GJE method can result in three possible cases: (1) system is inconsistent, resulted when the δ/eP combination is inconsistent, which does not generate any solution; (2) system independent, resulted when attributes are independent, which generates exactly one solution; and (3) system dependent, resulted when attributes are dependent, which generates an infinite number of solutions.

The inconsistency problem presented in case (1) is typically resulted when the system contains at least one equation that includes only *enforced protection levels* (i.e., $\{p_1; \ldots; p_m\} \sqsubseteq eP$). This leads to limiting the options for δ to one possible value, and will therefore entail an inconsistency if the specified δ value by the user/system does not match the acceptable one. Definition 14 discusses this constraint.

Definition 14 (δ/eP Inconsistency). Let $\{p_1; p_2\}$ be the protection levels to be assigned to attributes $\{a_1; a_2\} \sqsubseteq c.SA$. Assume that risk r_i of $\vec{r}$ is impacted only by $\{a_1, a_2\}$. The linear system will therefore include the following equation: $\widetilde{\omega_{11}}.p_1 + \widetilde{\omega_{12}}.p_2 = 1 - \delta$. Accordingly, the δ/eP combination is said to be inconsistent only if:

$$\{p_1; p_2\} \sqsubseteq eP \;\; and \;\; \delta \neq 1 - (\widetilde{\omega_{11}}.p_1 + \widetilde{\omega_{12}}.p_2) \qquad \blacksquare$$

Reasoning Algorithm. Algorithm 2 presents the protection strategy identification algorithm that takes as input the impact matrix, $Wc[][]$, the δ value (specified by the user or left empty), and the array of enforced protection levels, $eP[]$. It outputs the array of identified protection strategies, $Pc[][]$. This is done following one major step that varies according to the δ value. Indeed, the process starts by checking the value of δ, which can be specified by the user (i.e., $\delta \in [0; 1]$) or left empty (i.e., the user asks for maximizing privacy protection).

- Step 1 (lines 3-5): If δ is equal to 0 (line 3), this means that the user does not accept to take any risk and the data protection levels must consequently be at their highest levels. The process calls the *createFullProtStrategy* function that returns the full protection strategy, $\vec{p} = \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}$ (line 5).
- Step 2 (lines 6-8): If δ is equal to 1 (line 6), this means that the user agrees to take all risks and share fine-grained data in order to maintain the full quality of services received in return. The data protection levels should thus be left at their default values. The process calls consequently the *createDefaultStrategy* function that assigns the enforced value to p_j of $\vec{p}$ if $p_j \in eP$, or a value of 0 if not (line 8).

- Step 3 (lines 11-17): If δ was not specified by the user (i.e., equals to NULL), this means that the user wants to maximize her privacy protection while considering other preferences. The process calls the $getLowestPossibleDelta()$ function (cf. Algorithm 2) accordingly, which returns the lowest-possible δ value that can be considered in the relevant situation. This function can return a value greater than 1 (line 14) if the linear system has generated inconsistencies for all possible δ values, which will outputs an empty array of strategies (line 17). Otherwise, the minimized value of $\delta \in [0;1]$ is adopted and the process proceeds accordingly.

- Step 4 (lines 18-34): If $\delta \in]0;1[$ (line 18), which can be specified by the user or identified by the system when calling the $getLowestPossibleDelta()$ function, the process builds the linear system by calling the $buildSystem()$ function and stores the resulted system in the two-dimensional array $System[][]$ (line 20). Then, it solves the system using the GJE method by calling the $solveSystemGJE()$ function (line 21). This function returns a reduced row echelon form stored in $M[][]$. Following that, the process checks for inconsistency by calling the $checkInconsistency()$ function, which returns a Boolean value stored in *inconsistency* (line 22). In fact, the system can generate inconsistencies here only when the user specifies δ.

  - If *inconsistency* equals *False*, this means that the system is consistent. The process checks attribute dependency in $M[][]$ by calling the $checkDependency()$ function, which returns a Boolean value stored in the variable *dependency* (line 24).

    * If *dependency* equals *False* (line 25), this means that attributes are independent, and the system results in one exact solution for each unknown p_j value, leading to create one protection strategy. This procedure is done by calling the $createIndependentStrategy()$ function (line 27).

    * If *dependency* equals *True*, this means that attributes are dependent, and the system has an infinite number of possible solutions. The process calls accordingly the *createDependentStrategies* function (line 20), which starts by identifying existing dependencies among the unknown p_j items. Then, it performs two operations on each dependent p_j item. The first operation prioritizes the attribute of the selected p_j, by assigning a 0 value to p_j, which means that no protection is applied on a_j. The second operation assigns a value of 1 to p_j (i.e., stop sharing a_j), which gives priority to the associated dependent attributes. Next, both operations calculate the remaining p items that are dependent from p_j. This function consequently identifies several appropriate strategies, where each emphasizes at least one dependent attribute.

  - If *inconsistency* equals *True* (i.e., δ/eP combination is inconsistent), the process generates an empty array of strategies.

Algorithm 2: Protection Strategy Identification

Input: $Wc[][]$, δ, $eP[]$; // impact matrix, risk threshold, and the enforced data protection levels;
Output: $Pc[][]$; // the array of protection strategies;

Variables: *System[][], M[][], inconsistency, dependency*;
begin
  if $(\delta == 0)$ **then**
    // user requests the maximum privacy protection;
    $Pc \leftarrow createFullStrategy(1)$; // returns a strategy comprised of p values equal to 1 (i.e., 100% data protection);
  else if $(\delta == 1)$ **then**
    // user accepts to share fine-granular data;
    $Pc \leftarrow createDefaultStrategy(0, eP[])$; // returns a strategy comprised of the default protection levels (i.e., $p \in eP[]$ OR 0) ;
  else
    // δ value is not specified by the user OR $\delta \in]0;1[$;
    if $(\delta == NULL)$ **then**
      // δ is not specified by the user (i.e., maximize protection);
      $\delta \leftarrow getLowestPossibleDelta(Wc[][], eP[])$; // returns the lowest-possible value for δ that meets eP preferences (i.e., it does not generate inconsistencies);
    if $(\delta > 1)$ **then**
      // this can be resulted from the *getLowestPossibleDelta()* function;
      // it means that the linear system has generated inconsistencies for all possible δ values -> the user should change at least one of the enforced protection levels in eP;
      $Pc = EMPTY$; // the resulted set of strategies is therefore empty;
    else
      // $\delta \in]0;1[$ specified by the user or identified by the system (δ minimized);
      $System \leftarrow buildSystem(Wc[][], \delta, eP[])$; // build the linear system;
      $M \leftarrow solveSystemGJE(System)$; // solves the linear system using the GJE method;
      $inconsistency \leftarrow checkInconsistency(M)$; // returns True if δ/eP combination is inconsistent;
      if $(inconsistency == False)$ **then**
        $dependency \leftarrow checkDependency(M[][])$; // returns True if system is dependent;
        if $(dependency == False)$ **then**
          // attributes are independent (one exact solution);
          $Pc \leftarrow createIndependentStrategy(M[][], eP[])$;
        else
          // attributes are dependent (infinite number of solutions) ;
          $Pc \leftarrow createDependentStrategies(M[][], eP[])$;
      else
        // the linear system is inconsistent and the user must change δ OR $p \in eP$;
        // the inconsistency occurs here only if the user has specified a value for $\delta \in]0;1[$;
        $Pc = EMPTY$; // the set of strategies is therefore empty ;
return $Pc[][]$

Algorithm 3 details the *getLowestPossibleDelta()* function that is used to identify the lowest-acceptable value of δ in the relevant user's situation. It receives as input

the impact matrix, $Wc[][]$, and the array of enforced protection levels, $eP[]$. It outputs the minimized value of δ. This is done following two major steps:

- Step 1 (line 3): It sets the value of δ to 0 (start with the lowest value for δ).
- Step 2 (lines 4-13): While the value of δ is lower than or equal to 1, the process proceeds with the following sub-steps:
  - Step 2.1 (line 5): It builds the linear system by calling the $buildSystem()$ function and stores the resulted system in the array $System[][]$.
  - Step 2.2 (line 6): It solves the system using the GJE method by calling the $solveSystemGJE()$, which returns a reduced row echelon form stored in $M[][]$.
  - Step 2.3 (lines 7-13): It checks for inconsistencies in $M[][]$ by calling the function $checkInconsistency()$, which returns a Boolean value stored in $inconsistency$.
    - If $inconsistency$ equals $False$ (line 8), this means that the relevant δ value leads to have a consistent system, making it the lowest-acceptable value in the relevant situation.
    - If $inconsistency$ equals $True$ (line 11), this means that the combination of the relevant δ value with the enforced data protection levels of eP is inconsistent (cf. Definition 14). The process increments accordingly the value of δ by 0.1 and re-enters the while loop (line 13). In fact, we chose to increase the δ value by 0.1 in order to address the trade-off between minimizing δ and reducing the computational cost caused by the total number of iterations required to find the lowest-acceptable value. This satisfies the performance requirements stated in Challenge 3.

Algorithm 3: *getLowestPossibleDelta* Method (cf. Algorithm 2 - Line 13)

```
   Input: Wc[][], eP[]; // impact matrix, and the array of enforced data protection levels;
   Output: δ; // the lowest-possible value for δ;
1  Variables: System[][], M[][], inconsistency;
2  begin
3      δ = 0;// set the δ value to 0 to start with;
4      while (δ <= 1) do
5          System ← buildSystem(Wc[][], δ, eP[]); // build the linear system with the fixed δ;
6          M ← solveSystemGJE(System); // solves the linear system using the GJE method;
7          inconsistency ← checkInconsistency(M); // returns True if the system is inconsistent;
8          if (inconsistency == False) then
9              // this means that the current δ is the lowest-possible value in relation to eP;
10             break;
11         else
12             // the linear system is inconsistent -> increase the value of δ;
13             δ = δ + 0.1; // increment δ by 0.1;
14     return δ
```

We only detail in this chapter the pseudo-code of the main process, including the minimization of δ. However, the pseudo-codes of the remaining functions called in Algorithm 2 are detailed in the prototype source code provided in Section 4.5.

4.4.3.3 Best Strategy Selection

The second operation of *δ-Risk* is performed when the number of strategies resulted from the first operation is greater than 1 (i.e., $|\boldsymbol{P_c}| > \mathbf{1}$). At this point, ranking the strategies and selecting the $\boldsymbol{K}$—best ones to be proposed to the user becomes a need. $\boldsymbol{K}$ expresses the number of best protection strategies, i.e., those with the highest ranking score (cf. Definition 13). However, fixing the maximum value of $\boldsymbol{K}$is challenging as many factors may contribute to the perceived choice overload, such as the number of options, time constraints, and user expertise [139]. Accordingly, we assign the following default values to $\boldsymbol{max(K)}$ based on user profiles: 1 for beginner, 3 for intermediate, and 5 for advanced. Nevertheless, the value of $\boldsymbol{max(K)}$ can be changed manually by the user, and also updated by the system administrator based on user interactions.

The best protection strategies should best meet the privacy and service preferences of the user (cf. Section 4.4.1, while also minimizing the cost of data protection. To achieve this, the current operation ranks the resulting strategies (i.e., $\boldsymbol{P_c}$) based on the user-service preferences, expressed by $\boldsymbol{\vec{wA}}$, and the costs of protection functions selected by the *data protection* module of CaPMan, $\boldsymbol{cPF}$. The ranking process is carried out by the $Rank()$ function based on the following principle: the highest ranking score corresponds to the strategy with the shortest distance to $\boldsymbol{\vec{wA}}$ and the lowest cost of protection.

Reasoning Algorithm. Algorithm 4 presents the $Rank()$ function, which takes as input the array of protection strategies, $Pc[][]$, the array of weights assigned to attributes, $wA[]$, and the array of protection function costs, $cPF[]$. It outputs the array of K—best protection strategies, $BPc[][]$. This is done following three major steps:

- Step 1 (lines 3-14): It identifies the strategies with the shortest distance to $wA[]$:
  - Step 1.1 (line 3): It calls the $sortAndFilter()$ function, which identifies the number of different weight values, sort the array $wA[]$ in a descending sequence, and removes the redundant values. The resulting array is stored in $sortedWA[]$.
  - Step 1.2 (lines 4-10): For each distinct weight value (line 4), the process checks the number of attributes having this weight using the $attributesSimilarWeight()$ function (line 5). In fact, having several attributes with the same weight requires considering strategies that prioritize each of them separately. Hence, for each of these attributes (line 7), the process looks for the strategy that includes the related minimal protection value (line 9), and adds the attribute's weight to the score of the relevant strategy (line 10).

– Step 1.3 (lines 11-14): The process then filters the resulting array of strategies so that only the strategies with the highest score are considered. This will ultimately lead to strategies that include the lowest-possible protection levels assigned to attributes based on their importance to the user. These strategies are said to have the shortest distance to $wA[]$.

- Step 2 (lines 15-18): It calculates the cost of protection of the resulting strategies. The cost of a strategy is equal to the sum of costs of the protection functions associated to the attributes protected by this strategy (i.e., attributes having protection levels higher than 0 in the relevant strategy).

- Step 3 (lines 19-21): It adds the calculated costs to the scores of strategies. Then, only strategies with the highest ranking score are selected and added to the array $BPc[][]$, which will consequently include the best strategies that could be implemented in the current situation (lines 20-21).

Algorithm 4: Best Strategy Selection - $Rank()$ function

```
Input: Pc[][], wA[], cPF[]; // the array of protection strategies, vector of weights, and the
       array of costs of protection functions;
Output: BPc[][]; // the array of best protection strategies;
1  Variables: sortedWA[], A[], minP, Score[][], maxScore, CostPc[][];
2  begin
3      sortedWA ← sortAndFilter(wA[]); // sorts wA[] in a descending sequence and
         removes redundant values;
4      foreach weight ∈ sortedWA do
5          A ← attributesSimilarWeight(wA[], weight);
6          // the array A will include attributes having the same weight weight;
7          foreach a ∈ A do
8              minP ← getMinP(Pc[][], a); // minimal protection level to be assigned to a;
9              Score ← addScore(Pc[][], minP, a, wA[]); // updates the score of strategies
                 having minP;
10         maxScore ← getMaxScore(Score[][]); // returns the maximal score;
11         for k ← 0 to |Score| do
12             if (Score[k][1] != maxScore) then
13                 Pc ← deleteStrategy(k); // keeps only strategies with the highest score;
14     for i ← 0 to |Pc| do
15         for j ← 0 to |Pc[0]| do
16             if (Pc[i][j] != 0) then
17                 CostPc[i][1] = CostPc[i][1] + cPF[j]; // calculate the cost of protection
                     for each strategy;
18     Score ← addCostToScore(Score[][], CostPc[][]); // adds the cost to the strategy score;
19     maxScore ← getMaxScore(Score[][]);
20     BPc ← selectBestStrategies(Pc[][], Score[][], maxScore); // BPc includes only the
         best strategies, i.e., the strategies with the highest score;
21 return BPc[][]
```

4.5 Implementation & Evaluation

In this section, we present the implementation phases of the *δ-Risk* proposal from the back-end and front-end perspectives. Then, we evaluate the performance of the risk reasoner in multiple cases, and we formally study its storage complexity and effectiveness in strategy identification.

4.5.1 *δ-Risk* Implementation

In order to validate our *δ-Risk* solution, we developed a Java-based prototype and embedded it on the user's mobile device. As illustrated in Figure 4.8, the prototype performs continuous reasoning over the user's situation and generates dynamic protection strategies based on user preferences and contexts. The source code of the *δ-Risk* prototype is accessible online for download via this link[1].

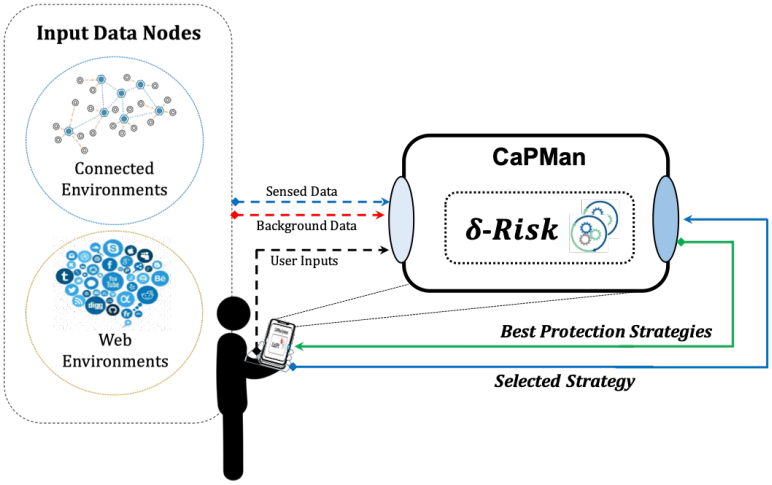

FIGURE 4.8: Implementation of the *δ-Risk* proposal

In what follows, we illustrate the prototype operation in the back-end. Then, we present the front-end mockups of the associated mobile application. It is important to note that this application is currently under development. We only represent here the mockups of the respective user interfaces.

4.5.1.1 Back-end: Java-based Prototype

We consider the privacy situation of Alice described in Section 4.2 to showcase how the system works. Figure 4.9 presents the overview of risks that has been resulted from the execution of the CaSPI reasoner (cf. Chapter 3). Assume that after alerting Alice about these risks, she adjusted her privacy preferences and accepted to take a maximum risk level of 60% in her current situation, as shown in Figure 4.10. We also illustrate in this figure the impact matrix, W_c, of *location* and *energy-consump* attributes on the risks, which is automatically calculated by the risk reasoner. The

[1] `https://spider.sigappfr.org/research-projects/delta-risk/`

δ-Risk process is consequently executed, and generates one best strategy that suggests achieving 40% protection for energy-consumption data and 40% protection for location data (cf. Fig.4.10).

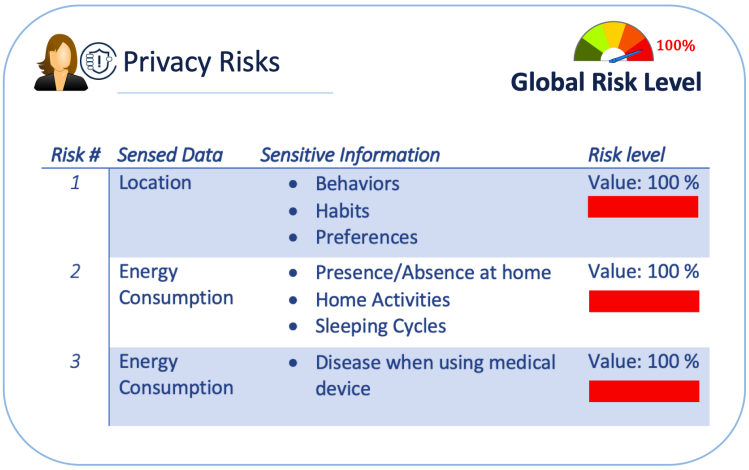

FIGURE 4.9: Alice's Privacy Situation

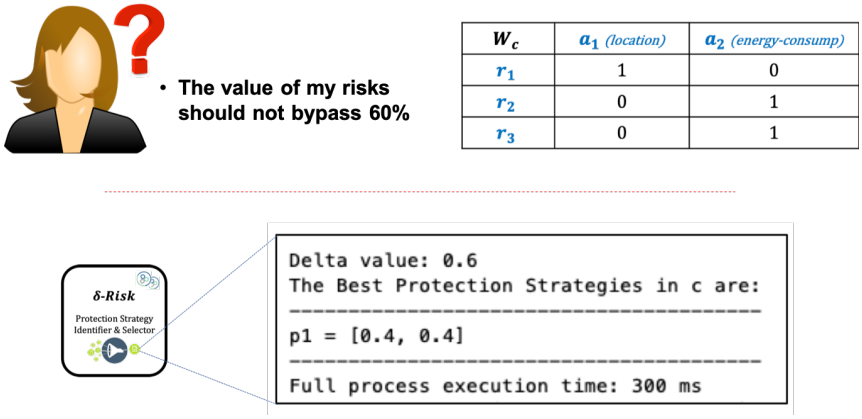

FIGURE 4.10: Alice Preferences, Impact Matrix, and Resulted Strategy

4.5.1.2 Front-end: User Interfaces

In this section, we present the mockups of the mobile application user interfaces related to preference specification, protection strategy selection, and the global privacy situation. The mockups were designed using Inkscape graphics editor 1.0[2]. Figure 4.11 shows the preference specification interface, which varies according to the user's profile. A *beginner* user has only the option to set the value of the risk threshold (δ) to 0 (i.e., maximum protection) or 1 (i.e., maximum quality of services) as previously described in Section 4.4.2. An *intermediate* user can personalize her sensitive information (see in Figure 3.14), services, and enforce a value for δ ranging from 0 to 1. An *expert* user has all *intermediate* properties plus the possibility to

[2] `https://inkscape.org/`

enforce a specific data protection level to a particular sensed attribute. For all user profiles, if no value is specified for δ, the user asks therefore to maximize her privacy protection while also considering the other preferences.

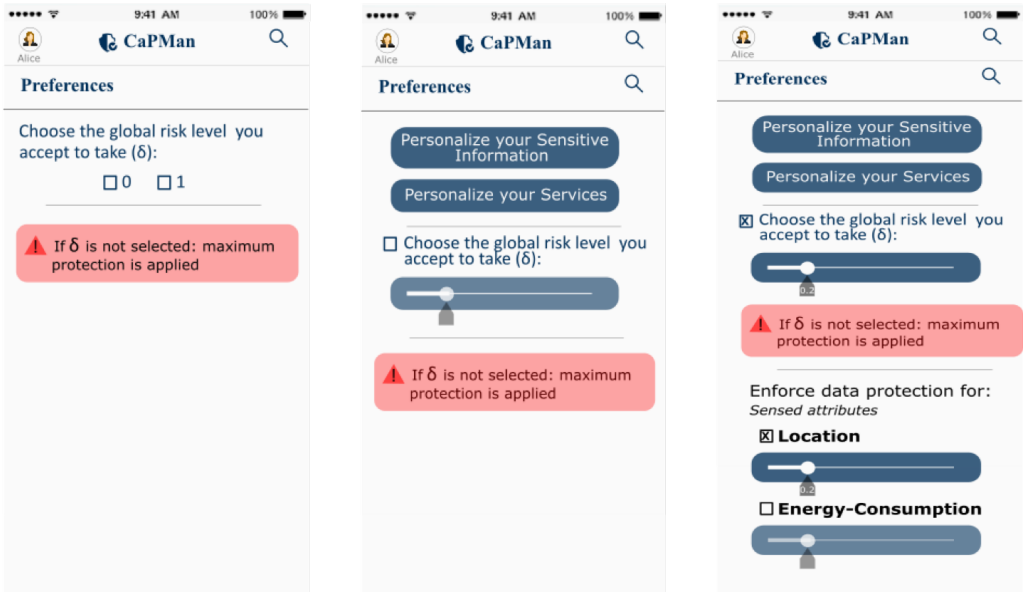

FIGURE 4.11: Preference Specification Interface

Upon the selection of the "personalize your services" button in the preferences interface, the service preferences interface is loaded, and the user has the ability to select which services are important to her. This allows consequently the risk manager to maximize the quality of important services to the user in the strategies provided.

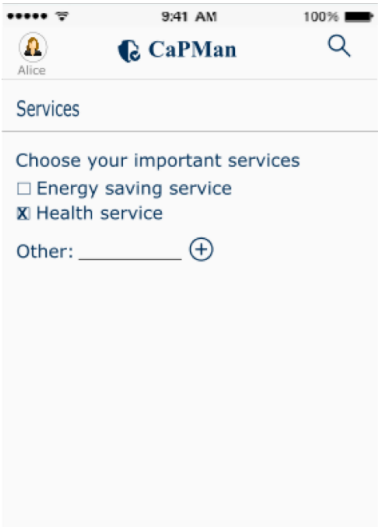

FIGURE 4.12: Service Preference Specification Interface

When a change occurs in the user situation, the risk reasoner is launched to infer the risks involved, and the *δ-Risk* manager is executed subsequently to identify the best protection strategies based on user situation and preferences. Once the process

is complete, *intermediate* and *advanced* users receive a notification to select one of the best data protection strategies that could be implemented in their situations. Figure 4.13 illustrates the strategy selection interface. As previously discussed, a timeout period is assigned to this query, such that if the user fails to respond within this period of time, the system selects randomly one of the strategies. The default maximum number of strategies provided is fixed at 3 for an *intermediate* and 5 for an *advanced* as stated in Section 4.4.3.3. Nonetheless, the user can manually change this variable by sliding the relative cursor.

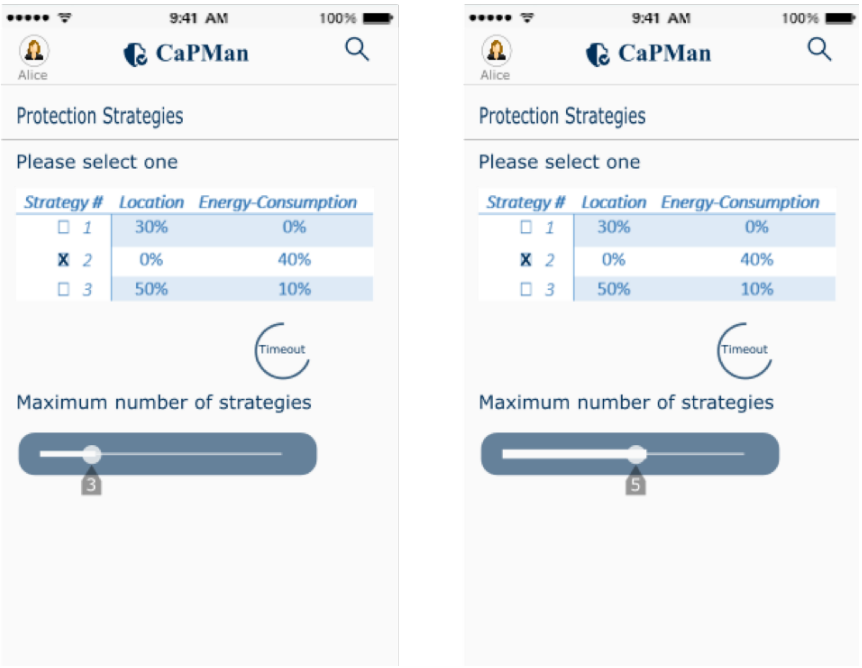

FIGURE 4.13: Protection Strategy Selection Interface

FIGURE 4.14: Global Privacy Situation Interface

The user can access at anytime the summary of her global privacy situation interface illustrated in Figure 4.14, which includes the risk summary (i.e., risk number and global risk level) and the protection strategy selected. She can also change the strategy selected by clicking on the "select your protection strategy" button.

4.5.2 Performance Evaluation

The objective here is to evaluate the ability of the approach, performance-wise, to operate in various scenarios, including worst case ones, and to meet the needs of scalability and efficiency (in time and space) outlined in Challenge 3. To achieve this, we start by considering four cases that measure the impact of the following metrics on performance: (i) the number of privacy risks involved in a single user situation, $|\boldsymbol{R_c}.\vec{\boldsymbol{r}}|$; (ii) the number of sensed attributes in a single user situation, $|\boldsymbol{c}.\boldsymbol{SA}|$; (iii) the level of dependency of sensed attributes in the impact matrix $\boldsymbol{W_c}$; and (iv) the complexity of the strategy ranking process. Then, we formally study the storage complexity of the proposal. The performance is evaluated based on two criteria: the total execution time and memory usage of one iteration. The tests are conducted on a machine equipped with an Intel i7 2.80 GHz processor and 16 GB of RAM. The chosen execution value for each scenario is an average of 10 sequenced values. We select the peak value of the in-use memory for each scenario when measuring memory usage.

Case 1: We study here the impact of privacy risks on performance by progressively increasing the number of risks inferred for the user in her situation. We limit the number of sensed attributes to 4, the level of dependency of attributes in $\boldsymbol{W_c}$ to 4 (i.e., all attributes are dependent), the δ value to 0.6, the vector of weights $\vec{wA} = \begin{bmatrix}1 & 2 & 1 & 2\end{bmatrix}$, and the number of protection functions to 4 with the following costs $cPF = \{1, 3, 1, 1\}$. We execute the *δ-Risk* process 7 times, taking into account the following number of risks for each iteration: 1; 10; 50; 100; 500; 1000; and 2000. Figure 4.15 shows that the number of privacy risks has a quasi-linear impact on the total execution time, with an average of 1 s up to 100 risks, 1.5 s for 500 risks, 2.2 s for 1000 risks, and 3.4 s for 2000 risks. The evolution is similar for the RAM consumption (see in Figure 4.15), with an average of less than 220 MB up to 100 risks, 250 MB for 1000 risks, and 290 MB for 2000 risks. This consequently highlights the importance of using the GJE method to solve the linear system. It is important to note that in practice, the number of risks inferred in a given situation will not practically exceed 100 for the user.

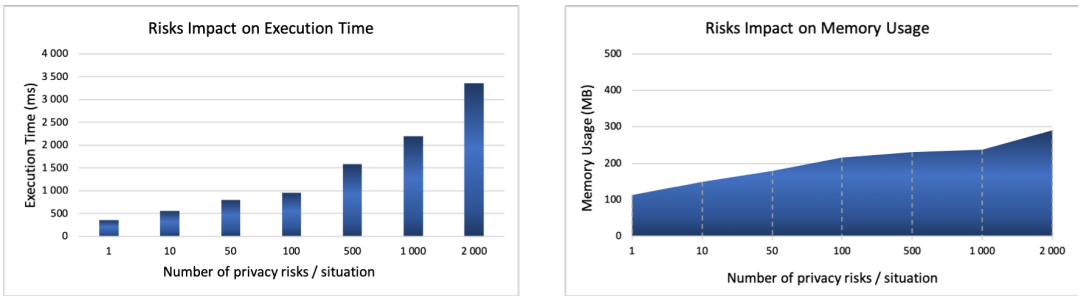

FIGURE 4.15: Case 1: Privacy Risks Impact

Case 2: We investigate here the impact of user-sensed attributes in a single situation on performance. We limit the number of risks to 100, the maximum level of dependency of attributes in W_c to 4 (i.e., each four attributes are dependent from each others), the δ value to 0.6, the vector of weights the vector of weights $\vec{wA} = \begin{bmatrix}1 & 2 & 1 & 2 & 0 & \dots & 0\end{bmatrix}$, and the number of protection functions to 4 with the following costs $cPF = \{1,3,1,1\}$. We execute the *δ-Risk* process twelve times, taking into account the following number of sensed attributes for each iteration: 1; 5; 10; 20; 30; 40; 50; 60; 70; 80; 90; and 100. Figure 4.16 shows that the number of sensed attributes has a quasi-linear impact on the total execution time, with an average of 1 s up to 10 attributes, 2 s up to 50 attributes, and 4 s up to 100 attributes. The evolution is similar for the RAM consumption (see in Figure 4.16), with an average of less than 200 MB up to 10 attributes, 1000 MB up to 40 attributes, and 2000 MB up to 100 attributes. It is important to note that in practice, the number of user-sensed attributes in her situation will not practically exceed 50.

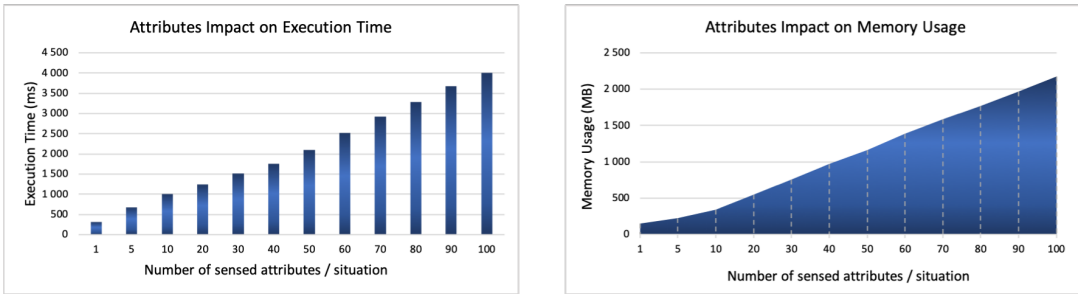

FIGURE 4.16: Case 2: Attributes Impact

Case 3: We evaluate here the influence of the attribute dependency level on performance. To do so, we limit the number of sensed attributes to 50, the vector of weights $\vec{wA} = \begin{bmatrix}1 & 2 & 1 & 2 & 0 & \dots & 0\end{bmatrix}$, the δ value to 0.6, and the number of protection functions to 4 with the following costs $cPF = \{1,3,1,1\}$. We execute the *δ-Risk* process six times, taking into account the following maximum levels of attribute dependency for each iteration: 1 (i.e., attributes are independent); 2; 4; 6; 8; and 10. According to figure 4.17, the number of sensed attributes has a quasi-constant impact on the total execution time, with an average of 1 s up to a maximum dependency level of 6. Then, the impact tends to be exponential with an average

of 9s for a dependency level of 8, and 1227s for 10. The evolution is similar for the RAM consumption (see in Figure 4.17), with an average of less than 1000 MB up to a maximum dependency of 6, and then tends to be exponential, reaching an average of 9500 MB up to a dependency level of 10. However, it is important to note that it is almost impossible to combine more than six *sensed attributes* in order to reveal certain *sensitive information* about the user that could not be revealed otherwise.

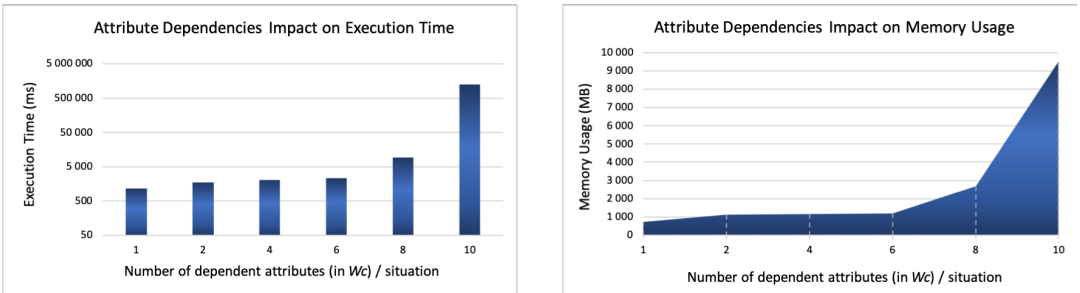

FIGURE 4.17: Case 3: Attribute Dependencies Impact

Case 4: We study here the impact of the strategy-ranking complexity on performance. To do so, we limit the number of risks to 100, the number of attributes to 50, the maximum level of dependency of attributes in W_c to 4 (i.e., each four attributes are dependent from each others), the δ value to 0.6, and the number of protection functions to 5. Then, we only execute the ranking function, $Rank()$, eight times, varying each time the sets of attribute weights, $\vec{wA}$, and costs of protection functions, cPF. As shown in Figure 4.18, the complexity of the strategy ranking process has no impact on the system's performance. The total execution time remains quasi-constant in all scenarios with an average of less than 500 ms. The same for the RAM consumption with an average of 300 MB. This emphasizes consequently the importance of storing the *appropriate protection strategies* resulting from the first *δ-Risk* operation (i.e., P_c) as long as no changes occur in the user's context.

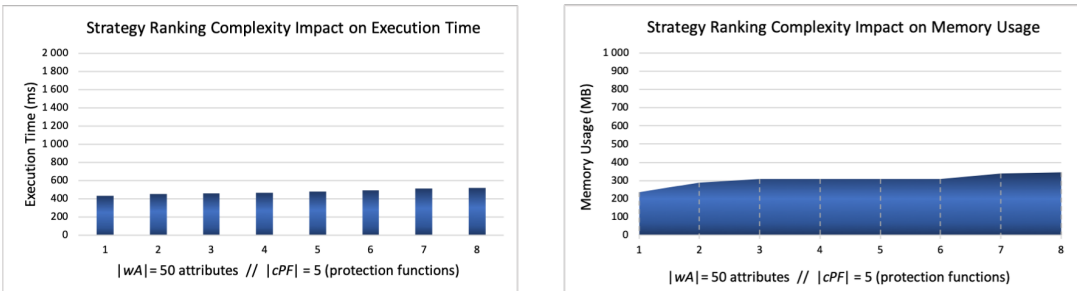

FIGURE 4.18: Case 4: Strategy Ranking Complexity Impact

THEOREM 2. The *δ-Risk* process maintains low storage complexity.

PROOF. The system stores locally the information characterizing only the current user situation. The amount of local storage space required by the system depends

on: (i) the number of individuals/relationships constituting the user context (individuals represents the user's sensed and background-oriented attributes); (ii) the preference specifications by the user; (iii) the number of risks involved in the user's situation; and (iv) the number of appropriate strategies identified in the current situation. This results consequently in a linear storage complexity of $O(n)$. Therefore, even in the worst case scenario of a large context size (e.g., 10,000 individuals/relationships) and a high number of risks (e., 1,000 risks), the system maintains a low storage complexity. □

Discussion. The experiments and studies conducted show that *δ-Risk* is scalable, and maintains computational and storage efficiency (cf. Challenge 3). In fact, in the worst case scenario of 1000 privacy risks, 50 sensed attributes with a maximum dependency level of 6, and 5 different protection functions assigned to these attributes, the solution is able to respond and provide strategies within an average time of 3 s and an average RAM space of 1200 MB. Nonetheless, if we consider a more quasi-real scenario of 20 risks, 5 sensed attributes with a dependency level of 3, and 5 protection functions, the solution responds within an average time of 550 ms and an average RAM space of 180 MB. Therefore, our proposal is capable of operating and assisting the user in different situations. This increases its re-usability for a variety of applications, including those requiring real-time assistance, and enables it to operate on a variety of devices, including those with limited resources.

4.5.3 Effectiveness in Strategy Identification

In this section, we formally study the effectiveness of the *δ-Risk* proposal in identifying always the best data protection strategies for the user according to her situation and preferences.

THEOREM 3. The *δ-Risk* process is always capable of identifying all possible appropriate strategies in the current situation, $\{\vec{p_1}; \vec{p_2}; \ldots; \vec{p_n}\} \sqsubseteq P_c$ (i.e., strategies that meet $R_c.v = \delta$).

PROOF. The proof consists of two cases, namely a simple and a generic case.

SIMPLE CASE. Consider that the user has only one sensed attribute, such that $c.SA = \{a_1\}$. According to Assumption 1 stated in Section 4.4.3.1, all (direct) risks inevitably impact the attribute a_1, which means that W_c is composed of a single vector with values of 1. Consequently, the resulting linear system consists of a single equation $p_1 = 1 - \delta$ (cf. Equation 2.6), generating one protection strategy $\vec{p} = \left[p_1\right] = \left[(1-\delta)\right]$, which will therefore constitute the best strategy to be delivered, $\vec{bp} = \vec{p} = \left[(1-\delta)\right]$.

GENERIC CASE. Consider that the user has m sensed attributes in her context c, $c.SA = \{a_1; \ldots; a_m\}$, and the number of risks inferred is n, $R_c.\vec{r} = \begin{bmatrix} r_1 & \ldots & r_n \end{bmatrix}$. W_c will therefore be a $n \times m$ matrix of {0,1} values expressing the impact of attributes of $c.SA$ on risks of $R_c.\vec{r}$. According to Equation 2.6, this results in a linear system of n equations with m unknowns (i.e., $p_1; p_2; \ldots; p_m$). The process will consequently proceed to identify the appropriate strategies based on the value of δ:

- If $\delta = \mathbf{0}$, this means that the user does not accept to take any risk and the data protection levels must be at their highest levels. All risks must consequently be eliminated, such that $R_c.\vec{r} = \begin{bmatrix} r_1 & \ldots & r_n \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \ldots & \mathbf{0} \end{bmatrix}$, which leads, according to Equation 2.3, to the full protection strategy $\vec{bp} = \vec{p} = \begin{bmatrix} \mathbf{1} & \ldots & \mathbf{1} \end{bmatrix}$.
- If $\delta = \mathbf{1}$, this means that the user agrees to take all risks and share fine-grained data in order to maintain the full quality of services received in return. Consequently, no additional protection is needed, and the data protection levels should be left at their default values. The output will therefore consist of the following strategy:

$$\vec{bp} = \vec{p} = \begin{bmatrix} p_1 & \ldots & p_m \end{bmatrix} \text{, where:}$$

$$\forall j \in [\mathbf{1}; m], \quad p_j = \begin{cases} \mathbf{0} & \text{if } p_j \notin eP \\ val \in]\mathbf{0}; \mathbf{1}] & \text{if } p_j \in eP, \text{ such that } p_j = val \end{cases}$$

- If δ is not specified by the user (i.e., equals to NULL), this means that the user wants to maximize her privacy protection while also considering other privacy preferences (i.e., the enforced data protection levels). Accordingly, the process identifies the lowest-possible value for δ that meet current needs.
  - If the process fails, which occurs when the linear system generates inconsistencies for all considered δ values due to the δ/eP combination (cf. Definition 14), the process outputs in this case an empty array of strategies (i.e., $P_c = \emptyset$). Following that, the user is asked to adjust her privacy preferences, and a time-out period is assigned to this task. If the user fails to respond before the time-out expires, the system sets the value of δ to 0, and the full protection strategy is implemented.
  - Otherwise, the identified value for δ is adopted in the current situation, and the process proceeds accordingly.
- If $\delta \in]\mathbf{0}; \mathbf{1}[$, this means that the user agrees to take risks to preserve as much as possible from the quality of services received in exchange for her data, however, the risk values should not bypass the specified threshold (i.e., δ). Accordingly, the process identifies all possible appropriate strategies that optimize the data utility-privacy trade-off (i.e., strategies that meet $R_c.v = \delta$) using the Gauss Jordan Elimination method to solve the linear system, such that:

$$\left[\begin{array}{cccc|c} \widetilde{\omega_{11}} & \widetilde{\omega_{12}} & \dots & \widetilde{\omega_{1m}} & 1-\delta \\ \widetilde{\omega_{21}} & \widetilde{\omega_{21}} & \dots & \widetilde{\omega_{2m}} & 1-\delta \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \widetilde{\omega_{n1}} & \widetilde{\omega_{n1}} & \dots & \widetilde{\omega_{nm}} & 1-\delta \end{array}\right] \rightarrow M = \left[\begin{array}{cccc|c} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1m} & v_1 \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2m} & v_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \dots & \alpha_{nm} & v_m \end{array}\right]$$

The process results in three possible cases:

1. System is inconsistent, which occurs when the δ/eP combination is inconsistent (cf. Definition 14). At this stage, the user is asked to either release the value of δ (if specified) or one of the impacting $p \in eP$. The δ-*Risk* process is re-launched accordingly with the updated δ/eP, or with $\delta = 0$ in the case when the assigned timeout period for this task expires without user response.

2. Attributes are independent, and the system has a unique solution:

$$M = \left[\begin{array}{cccc|c} 1 & 0 & \dots & 0 & v_1 \\ 0 & 1 & \dots & 0 & v_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & v_m \end{array}\right]$$

   This yields a single appropriate strategy, which will thus constitute the best strategy to deliver: $\vec{bp} = \vec{p} = \left[v_1 \quad v_2 \quad \dots \quad v_m\right]$.

3. Attributes are dependent, and the system has an infinite number of solutions:

$$M = \left[\begin{array}{cccc|c} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1m} & v_1 \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2m} & v_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{n1} & \alpha_{n2} & \dots & \alpha_{nm} & v_m \end{array}\right], \text{ where:}$$

$$\exists\, \vec{\alpha_i} \in M \text{ and } \exists\, j,k \in [1;m] : \alpha_{ij} \times \alpha_{ik} \neq 0$$

   At this point, the process iteratively assigns the lowest protection level (i.e., 0) to each dependent attributes (e.g., a_j, a_k) and calculates the remaining protection levels according to the matrix of dependencies M. After, it repeats the same iterations but with a value of 1, which stands for the highest protection level. When completed, the process identifies several appropriate strategies, $\{\vec{p_1}; \vec{p_2}; \dots; \vec{p_n}\} \sqsubseteq P_c$, where each emphasizes at least one dependent attribute.

Therefore, for all δ values, the process is always capable of calculating all possible appropriate strategies that lead to optimizing the data utility-privacy trade-off in the current user situation. □

THEOREM 4. The ***δ-Risk*** process is capable of always selecting the best data protection strategies to be delivered to the user.

PROOF. After identifying all possible appropriate strategies, the process executes the ranking function, $Rank()$, in order to select only the best ones, which stands for the strategies that best meet user preferences and minimize the cost of protection. The $Rank()$ function ranks the strategies according to the service preferences of the user (i.e., $\vec{\boldsymbol{wA}}$) and the costs of the selected protection functions (i.e., $\boldsymbol{cPF}$). It assigns the highest ranking score to the strategy with the shortest distance to $\vec{\boldsymbol{wA}}$ and the lowest cost of protection. Therefore, for each δ value:

- If $|\boldsymbol{P_c}| = \mathbf{1}$, the identified strategy is automatically chosen as the best one.
- If $|\boldsymbol{P_c}| > \mathbf{1}$, the process ranks the strategies and selects the $\boldsymbol{K}$-best ones to be delivered to the user. □

THEOREM 5. *δ-Risk* provides the user with at least one best data protection strategy per context.

PROOF. As proved in the preceding theorems, the process is capable of providing at least one best strategy in all existing cases. □

4.6 Summary

We present in this chapter our proposed user-centric multi-objective approach for context-aware privacy management in connected environments (*δ-Risk*). This approach features a new privacy risk quantification model to dynamically calculate and select the best data protection strategies for the user based on her preferences and contexts (e.g., involved risks). Computed strategies are optimal in that they seek to closely satisfy user requirements and preferences while maximizing data utility and minimizing the cost of protection. We developed a prototype to validate our proposal and illustrated its functioning from both back-end and front-end perspectives. We also evaluated its performance by considering multiple cases, and formally studied its effectiveness in strategy identification. The results show that *δ-Risk* delivers scalability and efficiency, making it capable of supporting the user in a variety of contexts, including ephemeral ones, and providing her with at least one best strategy per context.

Chapter 5

Privacy-preserving during Protection Transitions

"Inference is always an invasion of the unknown, a leap from the known."

– John Dewey

Advances in privacy-enhancing technologies, such as context-aware and personalized privacy models, have paved the way for successful management of data utility-privacy trade-offs. However, significantly lowering the level of data protection when balancing utility-privacy to meet the individual's needs makes subsequent protected data more precise. This increases the adversary's capability to reveal the real values of the previous correlated data that needed more protection, making existing privacy models vulnerable to inference attacks.

To overcome this problem, we propose in this paper a new stochastic gradient descent solution for privacy-preserving during protection transitions, denoted P-SGD. The goal of this solution is to minimize the precision gap between sequential data values when decreasing data protection by the privacy model. P-SGD intervenes at the protection descent phase and performs an iterative process that measures data dependencies, and gradually reduces protection accordingly until the desired protection level is reached. It considers also possible changes in protection functions and studies their impact on the protection descent rate. We validated our proposal and evaluated its performance. The results show that P-SGD is fast, scalable, and maintains low computational and storage complexity.

5.1 Introduction

The rapid expansion of cyber-physical systems and the technological advances in sensing technologies and data mining techniques have contributed to the tremendous development of smart people-driven applications. These applications tend to reshape the lives of people in many domains by providing them with advanced services (e.g., increasing comfort, monitoring patients and elderlies). Delivering such services requires collecting and processing massive amounts of data (e.g., location data, health data) to discover underlying patterns and trends. However, privacy concerns hinder the wider use of this data especially as data mining and processing may give rise to serious privacy risks for application users, such as disclosing their health conditions, habits, activities, and so forth [101], [103], [104].

Consequently, balancing trade-offs between data utility and privacy protection has been subject to intense study in recent years [140]–[143]. Current context-aware privacy solutions [143]–[145], including our CaPMan proposal, as well as personalized privacy solutions [142], [146], [147] aim to maximize the usefulness of data by optimizing the level of protection according to data sensitivity in the current context and/or user preferences. However, these solutions do not consider the effect of temporal correlations between sequential data values on privacy loss. They assign the appropriate level of protection to the data according to the user's context (e.g., privacy risks involved) and/or preferences.

Nonetheless, continuously balancing data protection levels without considering previous protection patterns may entail temporal privacy leakage. In particular, this leakage occurs when the protection level significantly decreases, which widens the precision gap between prior/subsequent correlated data and makes subsequent data more precise. The large gap in precision improves the capabilities of an adversary, when using advanced mining techniques, to reveal the real values of prior data pieces that required more protection. This makes existing privacy-preserving solutions vulnerable to data inference attacks. A data inference attack is a data mining attack in which adversaries are capable of estimating/inferring real values of protected data with high confidence. One of the possible solutions to overcome this vulnerability is to integrate a gradient descent mechanism at the protection descent phase. This helps to reduce the precision gap between sequential protected data when downshifting the protection level. Gradient descent is a general paradigm that underlies algorithms for solving optimization problems [148]. It has been widely applied to many fields such as location-based applications for predicting moving destination [149], differential privacy [150], and personalized privacy [151]. Nonetheless, to the best of our knowledge, there has not been any work on securing data protection transitions using gradient descent.

The implementation of a gradual descent process for data protection levels is challenging, as the corresponding deviation rate depends on several dynamic factors. First, the temporal correlations between sequential data values, which may vary from sequence to sequence as the data can be generated in regular or irregular time series. Second, the dynamicity of the protection function chosen by the system to be executed on data values. In fact, the system can change the data protection function at the protection transition phases with a view to improving protection, reducing the cost of protection (i.e., computational costs), or due to errors in function operations. However, the protection functions can share similarities in their operations (e.g., generalization and random-noise functions add noise to the real value of data), making it important to consider their dependence and its impact on the protection deviation rate. What makes it more challenging is the need for a fast and low complex solution, which makes it re-usable by various privacy models, including those offering real-time protection, and operational even for resource-constrained devices. Finally, the solution should follow a non-deterministic descent to avoid revealing the deviation rate by adversaries in case of repeated descent patterns.

To address these challenges, this chapter introduces P-SGD, a stochastic gradient descent solution for privacy-preserving during protection transitions. P-SGD empowers existing privacy models against data inference attacks, by minimizing the precision gaps of sequential protected data values during the protection descent phase. It follows an iterative process to identify the appropriate protection level to be assigned to each transitional data until the targeted level is reached. Computed protection levels consider the temporal dependencies between data values and the dependencies between protection functions (in case of change). Our solution is generic (i.e., it handles attributes with different data types and formats), and supports simultaneous reasoning over multiple attributes. We validated our proposal and evaluated its performance. Results show that P-SGD is fast, scalable, and maintains low computational and storage complexity.

The rest of the chapter is organized as follows. Section 5.2 presents the motivating scenario. Section 5.3 details our P-SGD proposal and provides formal definitions of the key terms used. Section 5.4 outlines the experiments and results. Section 5.5 presents an overview existing privacy models to which our proposal can be connected (i.e., context-aware and personalized privacy models). Finally, Section 5.6 summarizes the chapter.

5.2 Motivating Scenario

To motivate our proposal, we consider here a second scenario for Alice. We remind the reader that Alice is a COPD patient, and shares her location data with a healthcare provider to benefit from an emergency care system that offers healthcare

services (e.g., smart ambulance service that she would use in case of respiratory distress). Alice also shares her location data with several other service providers in exchange for their services through applications and social media platforms (e.g., Facebook, Google Maps).

The trust relationship between Alice and the providers may vary greatly due to many factors, such as the privacy risks associated with the sharing of data, the sensitivity of her context (e.g., private meeting), or the third parties with whom her data is communicated. Alice may therefore want to protect her privacy in some situations but without completely losing associated services. To do so, she uses a context-aware privacy-preserving system (e.g., CaPMan) that optimizes the data protection according to her contexts and preferences.

Consider that Alice has a medical appointment at the Belharra-Ramsay center for her COPD treatment. She takes the road from her home to the treatment center. However, locating Alice in the pulmonary rehabilitation center can entail the disclosure of her health condition, which involves privacy concerns for her. Accordingly, assume that the privacy system increases data protection to 80% when Alice arrives at the center, and then shifts the level of protection to 20% when she leaves. The system protects sensed data using a generalization-based protection function. In the following, three cases are considered to highlight the impact of the second protection transition phase (from 80% to 20%) on privacy loss.

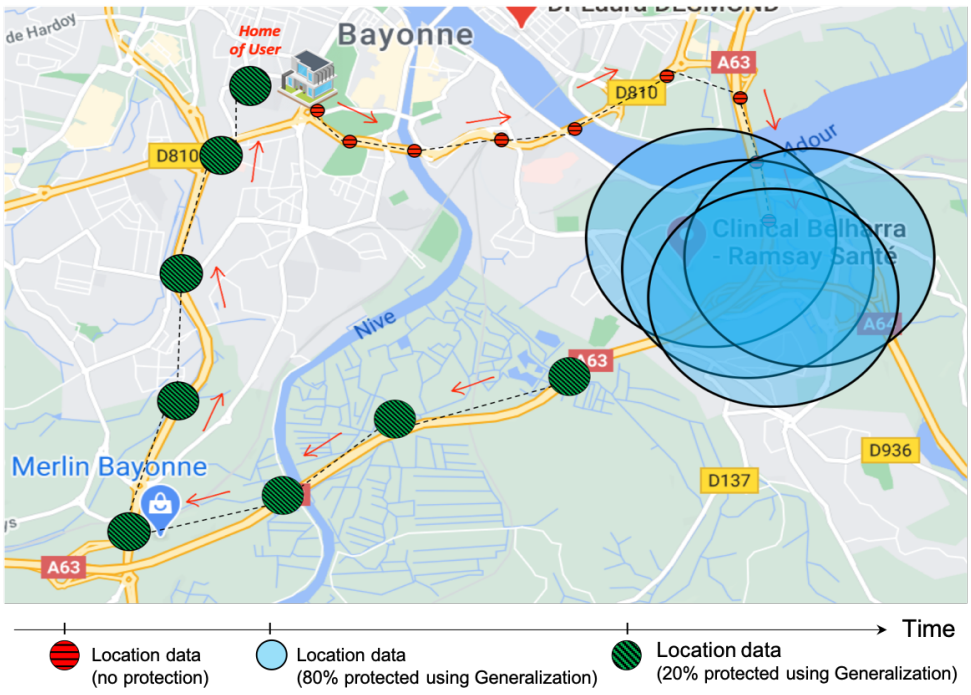

FIGURE 5.1: Case-1

In case-1, represented in Figure 5.1, the system shifts the level of protection to 20% and continues to perform the same protection function on generated data (i.e.,

the generalization function). The location data are generated at a regular time interval. When processing and analyzing protected data values, an adversary can notice a significant gap in the level of precision between transitional/correlated data (see in Figure 5.1). The precision gap limits the range for estimating previous user locations where protection was critical (e.g., Alice's presence in the medical center), which entails privacy problems. This consequently underlines the need for a gradual descent in the protection level in order to overcome vulnerabilities that may arise during protection transitions.

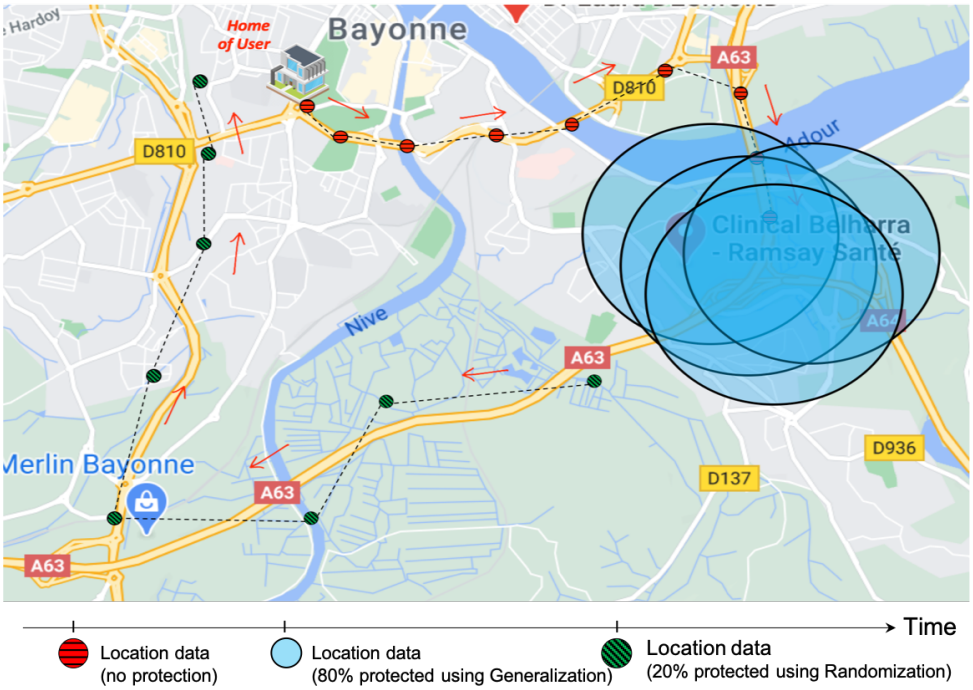

FIGURE 5.2: Case-2

As previously mentioned in Section 5.1, the system can change the protection function to be executed on data at the protection transition phase. In case-2, illustrated in Figure 5.2, the system changes the function when the protection level shifts to 20%, and adopts a randomization-based function that adds random noise to the real location positions. However, the generalization and randomization functions share similarities. They both add noise to the data, which makes them dependent, and the privacy issues related to lowering the protection level persist. This highlights the need to examine dependencies between protection functions and their impact on the protection deviation rate.

In the previous two cases we considered regular time series data. However, data can be also collected in irregular time series, i.e., the data collected follow a temporal sequence, but the measurements may not occur at regular time intervals. For instance, case-3 assumes that after leaving the medical center, the system has stopped sharing (protected) location data only for a specific time interval due to loss of connectivity with the GPS sensor (cf. Figure 5.3). When data sharing started again, the

temporal distance between the last data shared and the current one has already exceeded the temporal granularity of the attribute (i.e., location). The two data pieces are thus independent and the adversary will not be able to link previous and subsequent location patterns. It is thereby important to measure the temporal correlations between sequential data and study its impact on data protection.

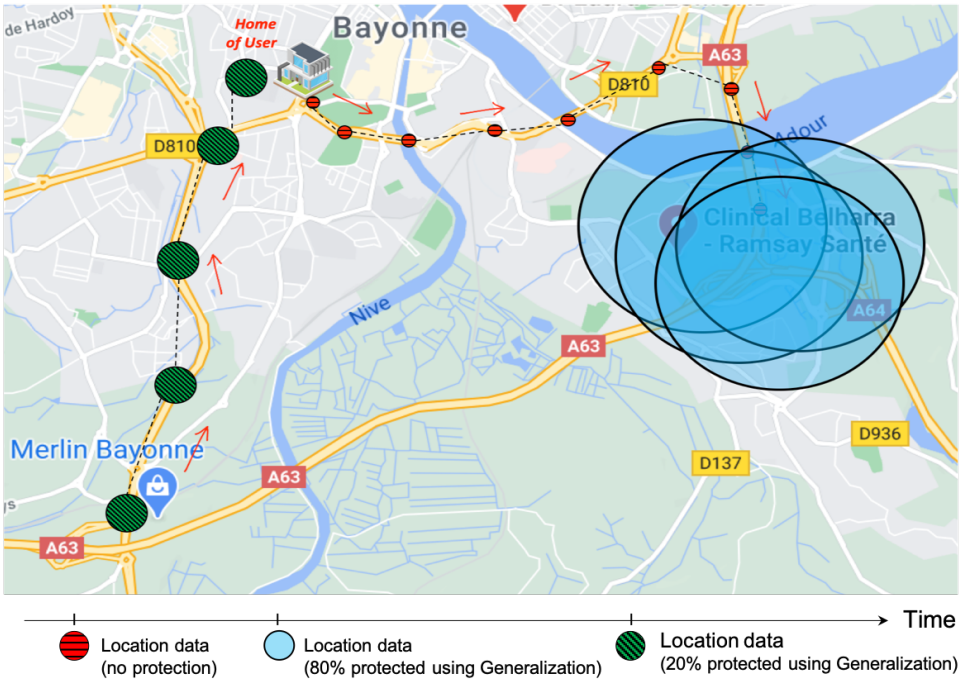

FIGURE 5.3: Case-3

However, designing the gradient descent solution while keeping the aforementioned needs in mind requires addressing the following challenges:

Challenge 1. ***Coping with data dependency*:** How to track and measure the temporal dependencies between sequential data values and study their impact on the protection descent rate?

Challenge 2. ***Coping with protection function dependency*:** How to compute the similarity between transitional protection functions (in case of change) and adjust the downshifting mechanism accordingly?

Challenge 3. ***Providing a non-deterministic solution*:** The data protection level can fluctuate between two same values for several transitions. This may entail the disclosure of the deviation rate by adversaries if the executed process is deterministic (cf. Figure 5.4). The solution should therefore be non-deterministic to overcome the vulnerabilities arising from repeated transition patterns.

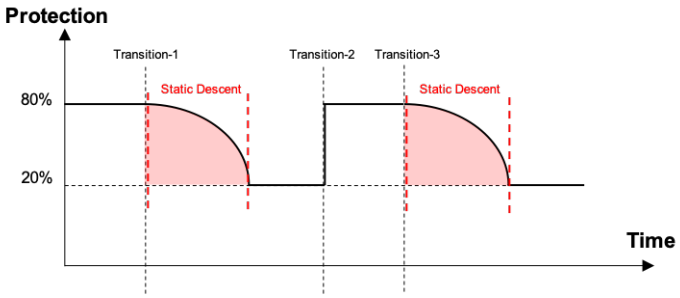

FIGURE 5.4: Repeated Protection Transition Patterns

Challenge 4. ***Delivering scalability and efficiency*:** The solution must be scalable, i.e., handles simultaneous reasoning over an increasing number of attributes. Moreover, it should maintain computational and storage efficiency, which increases its reusability to also include privacy models subject to real-time constraints, and makes it operational on a variety of devices, including those with limited resources.

5.3 P-SGD Proposal

Current context-aware and personalized privacy-preserving models (e.g., CaPMan) enable the variation of data protection levels based on user preferences and/or situations (e.g., privacy risks involved) in order to optimize the balancing of data utility-privacy. However, these models perform direct shifting of the data protection level, which may lead in certain cases to temporal privacy leakage due to data correlations. In particular, the data privacy leakage occurs when significantly decreasing the level of protection, creating a significant gap in the level of precision between previous and subsequent data. This increases the ability of an adversary to reveal the real values of previous correlated data that needed more protection, entailing privacy concerns for the user.

In order to overcome this vulnerability, we propose **P-SGD**, a **P**rivacy-based **S**tochastic **G**radient **D**escent solution for privacy-preserving during protection transitions. Our solution addresses the challenges and needs mentioned in Section 5.2. It operates during protection descent phases to minimize precision gaps between sequential protected data values. To do so, P-SGD features an iterative protection descent process that identifies the appropriate *data protection level* (cf. Definition 12) to be achieved for each data piece prior to its release to data consumers. The process stops when reaching the targeted protection level, i.e., the one specified by the privacy model.

P-SGD supports attribute diversity, i.e., it is capable of operating for data of various *sensed attributes* (e.g., location, energy-consumption, camera recordings) with different data types (e.g., scalar and multimedia data). It also supports *protection*

function diversity. In fact, existing protection functions vary from data anonymization, data perturbation using noise addition, privacy-aware access control to encryption (cf. Section 4.3). Each of these functions achieves differently the desired *data protection level*. This makes therefore our approach generic and compatible with numerous existing privacy models in various application domains. P-SGD can be plugged into the privacy model, as shown in Fig. 5.5, to provide an additional layer of protection against data inference attacks.

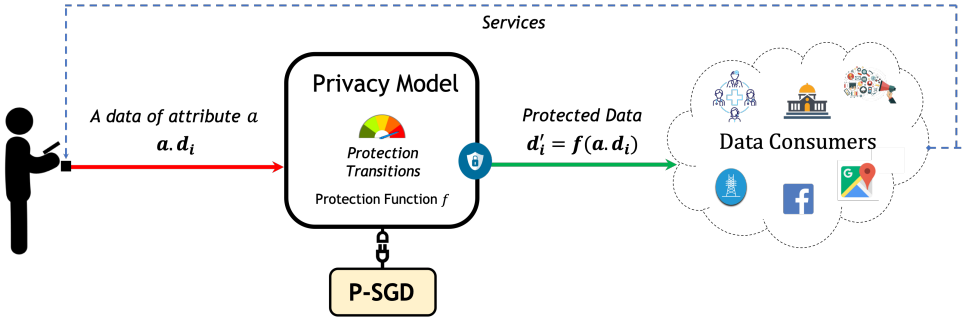

FIGURE 5.5: Integration of P-SGD

Before delving into the process, we would like to remind the reader of some formal definitions that were provided in previous chapters and will be used next. Specifically, the definitions of an *attribute*, *sensed attribute*, *protection function*, and a *data protection level*. However, we extend the *attribute* definition to take into account the standard time periods during which the data of attributes are dependent.

***Definition 5** (Attribute). Let A be the set of $\boldsymbol{attributes}$ $\{a_1 ; a_2 ; \dots ; a_n\}$ describing the user u and her physical environments $\sum env \in E_u$. An attribute $a \in A$ is formalized as follows:

$$a : \langle\, desc ; ent ; Log ; access ; \tau \,\rangle \text{ , where:}$$

- $desc$ denotes the textual description of a (e.g., location data, energy-consump data, user activities, profile images, home appliances).
- $ent \in \{u\} \cup E_u$ denotes the entity related to a, which can be the user u or an environment $env \in E_u$.
- $Log = \{\langle\, d ; M \,\rangle\}$ is the set of spatio-temporal data values of a. Log can be viewed as the log file of a, where:
  - d denotes the data value, which can be scalar (e.g., location, temperature, age, marital-status) or multimedia (e.g., image, audio, video).
  - $M = \{meta_1 ; \dots ; meta_n\}$ is the set of metadata characterizing d. For instance, M can include the following metadata:
    - $t_{capture}$, denotes the time of capture of d.

* $l_{capture}$, denotes the location of capture of d.
* $source \in DN$, denotes the data source from which d is captured. $source$ can derive from connected environments (e.g., sensor, device) or web environments (e.g., social media platform, public database).
* $D_{consumer} \sqsubseteq DN$, represents the set of data consumers with whom d is shared (e.g., service providers, third parties), such that:

$$D_{consumer} = \{\ dc_1\ ;\ dc_2\ ;\ ...\ ;\ dc_n\ \} \cup \{\bot\}\ \text{, where:}$$

  · $dc_i \in D_{consumer}$ is a *data node* expressing a data consumer.
  · $D_{consumer} = \varnothing$ indicates that data consumers are unknown.
  · $D_{consumer} = \{\bot\}$ denotes that a is a public attribute.

- $access \in \{r\ ;\ r/w\}$ denotes the access rights of the CaPMan system to the data of a, which can be read or read/write. It expresses the level of control of the system over the data of a.
- τ denotes the standard time period during which two data values of a are said to be time-dependent. ■

Definition 5.1** (Sensed Attribute)**.** Let $SA \sqsubseteq A$ be the set of ***sensed attributes, i.e., attributes characterizing sensed data by deployed/wearable sensors, and on which the CaPMan system has access to control and manage, such that: $\forall a \in SA :$ $a.access = r/w$. ■

Definition 10** (Protection Function). A ***protection function, $f \in PF$, is a protection method that can be executed on data values of an attribute $a \in c.SA$ prior to their release to data consumers. f is a local function stored in the CaPMan system, such that:

$$f : \langle\ name\ ;\ categ\ ;\ Feature\ ;\ Param\ \rangle\text{, where:}$$

- $name$ denotes the textual name of f (e.g., generalization, random-noise)
- $categ$ represents the category to which f belongs, such that:

$$categ \in \{\textit{noise-addition ; anonymization ; access-control ; encryption}\}$$

- $Feature$ is the set of features characterizing f, including at least:
  - $cost$, the computational cost of f in terms of processing time and memory overhead
- $Param$ represents the set of input parameters of f, including at least:
  - $SA' \sqsubseteq SA$ is the set of attributes to which f is associated
  - P is the set of protection levels to achieve for the data of attributes in SA' ■

Definition 12** (Data Protection Level). A ***data protection level, p, expresses the amount of protection to be achieved for the data values of an attribute $a \in c.SA$. p is probabilistic with a value between 0 and 1, where 0 means that data is shared in fine-granular version (i.e., without any protection), and 1 means that data is not shared (i.e., highest level of protection). A value between 0 and 1 indicates the level of protection that should be reached when executing a *protection function* $f \in PF$ on the data of a. Knowing that the way to achieve p depends on the selected *protection function*.* ■

A stochastic gradient descent (SGD) method is generally defined as an iterative method for optimizing an objective function with suitable smoothness properties [152]. It has been widely adopted mainly for high-dimensional optimization problems as it reduces the computational burden, achieving faster iterations in trade for a lower convergence rate. This agrees with our needs listed in Challenge 4. We detail in what follows our proposed P-SGD method.

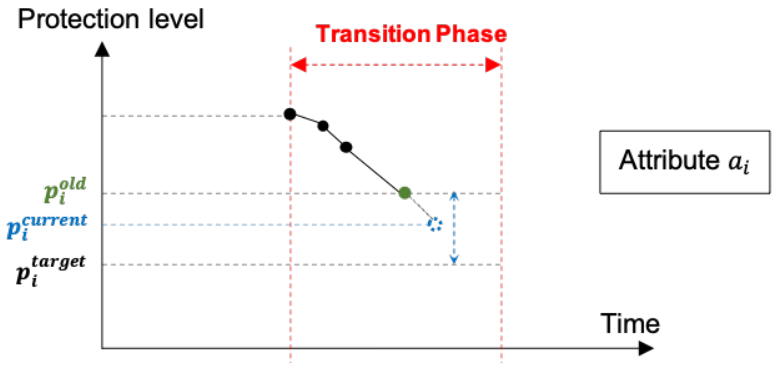

FIGURE 5.6: P-SGD process

According to Figure 5.6, let:

- p_i^{target} refers to the *targeted protection level*, i.e., the next protection level specified by the privacy model for data of attribute $a_i \in SA$. This level indicates the target level that must be reached in order to complete the P-SGD process
- p_i^{old} denotes the protection level of the previous data value of attribute $a_i \in SA$
- $p_i^{current}$ expresses the protection level to be assigned to the current data value of attribute $a_i \in SA$, such that $p_i^{current} \in [p_i^{target}; p_i^{old}]$

The iterative process followed by P-SGD is thus defined by the following formula:

$$p^{current} = p^{old} - \eta \bigtriangledown \quad \text{, where:} \tag{5.1}$$

- η represents the deviation rate of the protection level (the quantification of η is detailed in the following subsection)

- $\bigtriangledown \in [\mathbf{0;1}]$ expresses the random noise added to $\boldsymbol{\eta}$

We consider in this study that attributes are independent. The P-SGD process is therefore performed on the data values of each attribute separately. In order to track and measure the correlations in sequential data and the dependencies between their associated protection functions (cf. Challenges 1 and 2), we define a ***transition matrix***, ***Trans***, that contains only the properties of the last data value (i.e., $\boldsymbol{d_i^{old}}$) of each sensed attribute $\boldsymbol{a_i} \in \boldsymbol{SA}$. We store only the properties of the last data values since the process operates iteratively. This reduces storage overhead and allows for scalability in attribute number (cf. Challenge 4). ***Trans*** denotes therefore the cache, and can be represented as follows:

$$\boldsymbol{Trans} = \begin{bmatrix} \boldsymbol{t_1^{old}} & \boldsymbol{p_1^{old}} & \boldsymbol{f_1^{old}} \\ \boldsymbol{t_2^{old}} & \boldsymbol{p_2^{old}} & \boldsymbol{f_2^{old}} \\ \vdots & \vdots & \vdots \\ \boldsymbol{t_n^{old}} & \boldsymbol{p_n^{old}} & \boldsymbol{f_n^{old}} \end{bmatrix} \tag{5.2}$$

Where:

- $\boldsymbol{t_i^{old}}$ denotes the time of capture of $\boldsymbol{d_i^{old}}$ of attribute $\boldsymbol{a_i}$
- $\boldsymbol{p_i^{old}}$ refers to the protection level of $\boldsymbol{d_i^{old}}$ of attribute $\boldsymbol{a_i}$
- $\boldsymbol{f_i^{old}}$ is the protection function associated to $\boldsymbol{d_i^{old}}$ of attribute $\boldsymbol{a_i}$

5.3.1 Deviation Rate Quantification

The deviation rate, $\boldsymbol{\eta}$, depends on: (1) the temporal dependency of previous and current data values of $\boldsymbol{a_i}$, $\boldsymbol{d_i^{old}}$ and $\boldsymbol{d_i^{current}}$; and (2) the level of dependency of their related protection functions, $\boldsymbol{f_i^{old}}$ and $\boldsymbol{f_i^{current}}$.

Definition 15 (Time Dependency of Data). Let $\boldsymbol{depend_t}$ denotes the temporal dependency score of two data values, $\boldsymbol{d_i^{old}}$ and $\boldsymbol{d_i^{current}}$, of an attribute $\boldsymbol{a_i} \in \boldsymbol{A}$. $\boldsymbol{depend_t}$ has a value between 0 and 1, where 0 means that the data are time-independent, and 1 means that the data are fully dependent (time-wise), which typically occurs only when $\boldsymbol{t_i^{old}}$ and $\boldsymbol{t_i^{current}}$ are similar. The higher the temporal distance between the two data values is, the lower their time dependency is. The two data values are said to be time-dependent only if their temporal distance is less than the *standard time period* of their attribute $\boldsymbol{a_i}$ (i.e., $\boldsymbol{a_i.\tau}$). $\boldsymbol{depend_t}$ is therefore computed as follows:

$$\boldsymbol{depend_t}(\boldsymbol{d_i^{old}}, \boldsymbol{d_i^{current}}) = \begin{cases} \mathbf{1} - \frac{\boldsymbol{t_i^{current}} - \boldsymbol{t_i^{old}}}{\boldsymbol{a_i.\tau}} & \text{if } (\boldsymbol{t_i^{current}} - \boldsymbol{t_i^{old}}) \leqslant \boldsymbol{a_i.\tau} \\ \mathbf{0} & \text{otherwise} \end{cases} \quad \blacksquare$$

Definition 16 (Protection Function Dependency). Let f_i^{old} and $f_i^{current}$ denotes two protection functions. f_i^{old} and $f_i^{current}$ are said to be dependent only if their similarity score is above or equal 0.

$$\boldsymbol{sim}(f_i^{old}, f_i^{current}) \rightarrow [\mathbf{0;1}], \text{ where:}$$

- $\boldsymbol{sim}$ is a unit similarity function that checks the exact matching between the classes and the lists of features of the two protection functions, and returns a value between 0 and 1, such that:

$$sim(f_i^{old}, f_i^{current}) = 1 \text{ only if:}$$

$$f_i^{old}.class = f_i^{current}.class \text{ and } f_i^{old}.Feature = f_i^{current}.Feature \quad \blacksquare$$

The P-SGD process will therefore be executed only if the sequential data values are dependent and their associated protection functions are also dependent (i.e., only if $depend \neq 0$ and $sim \neq 0$). In order to quantify $\boldsymbol{\eta}$, we consider the following principles:

1. The more the temporal distance between previous/current data values increases, the more the time dependency among these data values decreases, and the protection gap between them can be enlarged.
2. The more previous/current protection functions are similar, the more the protection gap should be reduced.

Accordingly, $\boldsymbol{\eta}$ is quantified as follows:

$$\boldsymbol{\eta = c_i \times sim}(f_i^{old}, f_i^{current}) \times \boldsymbol{depend}_t(d_i^{old}, d_i^{current}) \tag{5.3}$$

Where:

- $\boldsymbol{c_i} \in \boldsymbol{C}$ is a system parameter that expresses the maximum deviation value of data protection level for attribute $\boldsymbol{a_i} \in \boldsymbol{A}$. $\boldsymbol{c_i}$ controls therefore the convergence speed of the protection level towards $\boldsymbol{p_i^{target}}$
- $\boldsymbol{sim}(f_i^{old}, f_i^{current})$ is the similarity function that returns a score $\in \mathbf{]0;1]}$
- $\boldsymbol{depend}_t(d_i^{old}, d_i^{current}) \in \mathbf{]0;1]}$ is the temporal dependency score

5.3.2 P-SGD Algorithm

We present here the reasoning algorithm of our P-SGD solution.

Algorithm 5: P-SGD Process

Input: a, c, $t^{current}$, $f^{current}$, p^{target}; // attribute, default deviation value, time of capture and protection function of $d^{current}$, and the targeted protection level;
Output: $p^{current}$; // the protection level to be assigned to $d^{current}$;
1 **Variables:** $Trans[][]$, $depend_t$, $simScore$, $\bigtriangledown$, η; // transition matrix, dependency score of data, similarity score of prot-functions, random noise and deviation rate;
2 **begin**
3 $depend_t = 1 - \frac{t^{current} - Trans[a][0]}{a.\tau}$; // Trans[][0] is the t^{old} column of d^{old} values;
4 $simScore \leftarrow sim(Trans[a][2], f^{current})$; // Trans[][2] is the f^{old} column associated to d^{old} values;
5 **if** $(depend_t \; != \; 0 \;\; \&\& \;\; simScore \; != \; 0)$ **then**
6 // dependent data values and dependent protection functions;
7 $\bigtriangledown \leftarrow randomNumber(0,1)$; // returns a random value between 0 and 1;
8 $\eta = c \times simScore \times depend_t$; // calculate the value of η;
9 $p^{current} = p^{old} - \eta\bigtriangledown$; // calculate the value of $p^{current}$;
10 **if** $(p^{current} \; <= \; p^{target})$ **then**
11 $p^{current} = p^{target}$; // check the validity of the calculated $p^{current}$ value ;
12 **else**
13 $p^{current} = p^{target}$; // data values or/and protection functions are independent;
14 $Trans \leftarrow updateTransMatrix(a, t^{current}, p^{current}, f^{current})$;
15 **return** $p^{current}$

Algorithm 5 presents the algorithm of our P-SGD solution that takes as input the concerned attribute, a, the maximum deviation value of protection, c, the properties of the current data value (i.e., $t^{current}$ and $f^{current}$), and the targeted protection level p^{target}. It outputs the calculated protection level to be assigned to the current data value, $p^{current}$. This is done following four major steps:

- Step 1 (line 3): It computes the dependency score of previous/current data values, d^{old} and $d^{current}$, and stores the result in the $depend_t$ variable.
- Step 2 (line 4): It calculates the similarity score of previous/current protection functions, f^{old} and $f^{current}$, and stores the result in the $simScore$ variable.
- Step 3: It checks the need or not for executing the gradient descent process:
  - Step 3.1 (lines 5-11): If data values are time-dependent and the related protection functions share similarities (line 5), the process is executed:
    * It starts by calculating the amount of the random noise, $\bigtriangledown$, to be appended to η (line 7).
    * It calculates the value of the deviation rate η (line 8) and the value of $p^{current}$ accordingly (line 9).
    * It checks the validity of the calculated value for $p^{current}$. If this value is less than or equal to p^{target} (lines 10), this means that the process has achieved

the protection level specified by the privacy model. Consequently, the value of $p^{current}$ equals the one of p^{target} and the downshifting process ends (line 11). Otherwise, the calculated value for $p^{current}$ is valid.

– Step 3.2 (lines 10-11): If data and/or associated functions are independent, the gradient process is not executed and the protection level is downshifted directly to p^{target}, i.e., the value of $p^{current}$ equals the one of p^{target}.

- Step 4 (line 14): the data properties of the relevant attribute are updated in the transition matrix, $Trans[][]$, and the process is ended.

We only detail in this chapter the pseudo-code of the main P-SGD process. Nonetheless, the pseudo-codes of the aforementioned functions are detailed in the prototype source code provided in Section 5.4.

5.3.3 P-SGD Integration in CaPMan

As previously discussed, the P-SGD proposal is generic and can be connected to various privacy models, including our proposed CaPMan model. We detail in this section the integration of P-SGD in CaPMan. As shown in Figure 5.7, P-SGD is connected to the *data protection* module of CaPMan that is responsible for applying protection on data before being delivered to data consumers. The protection is added based on the *protection strategy* selected in the relevant user's situation.

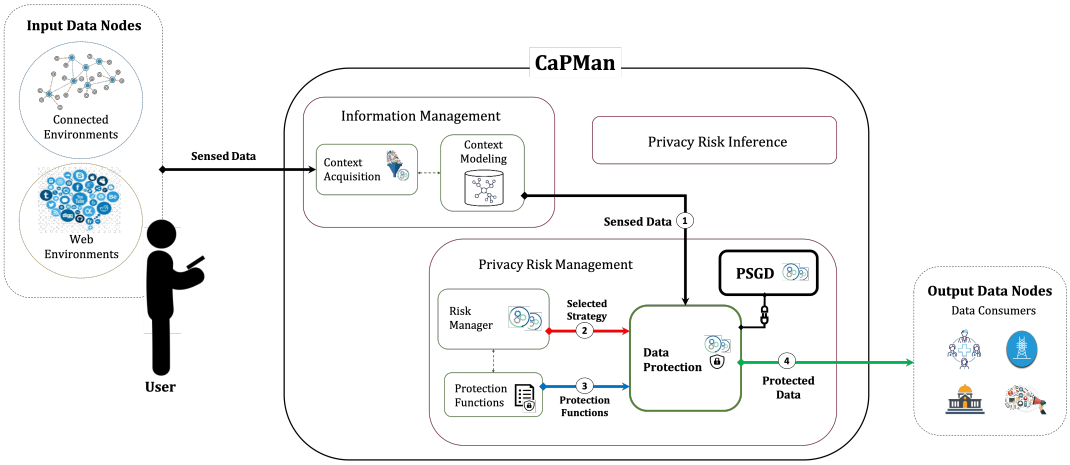

FIGURE 5.7: P-SGD Integration in CaPMan

The *data protection* module receives as input: (1) the data values of sensed attributes; (2) the strategy selected; and (3) the protection functions to be executed on relevant data of attributes. It calls the P-SGD process when the protection level to be assigned to the data of an attribute is decreased, which typically occurs when changing the *protection strategy*. Accordingly, the P-SGD process is iteratively executed for each data value upon its arrival in order to determine the appropriate *protection level*

to achieve for this data prior to its release. Once identified, the *data protection* module applies the corresponding *protection function* to this data in order to achieve the identified *protection level*, and then outputs (4) the protected version of this data to related data consumers.

5.4 Experimental Validation & Evaluation

In order to implement and validate our approach, we developed a Java-based prototype (the source code is available online through this link[1]). We illustrate in the following the prototype operation by considering the scenario of Alice described in Section 5.2. We focus on the second protection transition (i.e., from 80% to 20%), and assume that the protection function remains unchanged. We repeated the descent process three times to emphasize the non-deterministic nature of the solution in the case of repeated transition patterns (cf. Challenge 3). We consider here regular time series data with a data generation time of 1s, and we fix c at 0.5 (i.e., the maximum protection deviation is 50%).

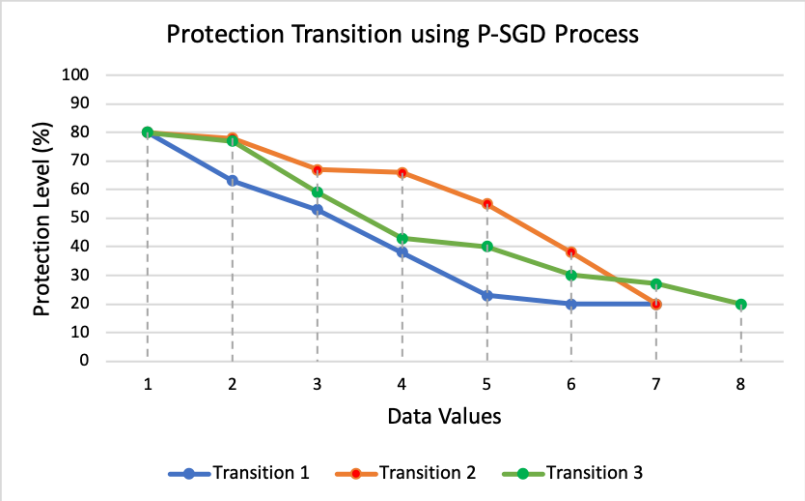

FIGURE 5.8: Securing protection transitions using the P-SGD process

As shown in Figure 5.8, the proposed P-SGD process is able to iteratively and gradually decrease the protection level until reaching the targeted one (i.e., 20%), with an average of 35ms per iteration. The deviation pattern varied between the three similar transition cases, as well as the number of data values required to achieve protection convergence (7 for transitions 1-2 and 8 for transition 3). This is due to the noise value associated with the deviation rate (i.e., $\bigtriangledown$), which varies randomly with each iteration.

[1] `https://spider.sigappfr.org/research-projects/psgd/` (P-SGD Prototype)

5.4.1 Performance Evaluation

The objective here is to evaluate the approach's effectiveness, in terms of performance, to operate in different scenarios. The approach is said to be effective if it meets the needs outlined in Challenge 4: (1) fast; (2) scalable (i.e., supports multi-attribute handling); and (3) low-complex in time and space (i.e., in terms of memory overhead and storage). To do so, we start by considering two cases to study the impact of the following two metrics on performance: (i) the complexity of the protection functions dependency; and (ii) the number of attributes handled simultaneously. Then, we formally study the storage complexity of the proposal. The performance is evaluated based on two criteria: the total execution time of one iteration and the memory overhead. The tests were conducted on a machine equipped with an Intel i7 2.80 GHz processor and 16 GB of RAM. The chosen execution value for each scenario is an average of 10 sequenced values.

Case 1: We consider two dimensions to study the complexity of the functions dependency: the first increases the number of features and the second increases the diversity in features between the two functions. We execute the P-SGD process 13 times, taking into account the following number of features for each iteration: 1, 5, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90 and 100. For each of these scenarios, we consider three sub-scenarios where we vary respectively the percentage of diverse features from 0%, 50% to 100%. As shown in Figure 5.9, the number and diversity of the features have no impact on the function dependency procedure, and thus on performance. This is due to the fact that the procedure verifies only the exact matching of the features' names and values. The process is executed in all scenarios with an average time of 35ms and 10MB of RAM usage.

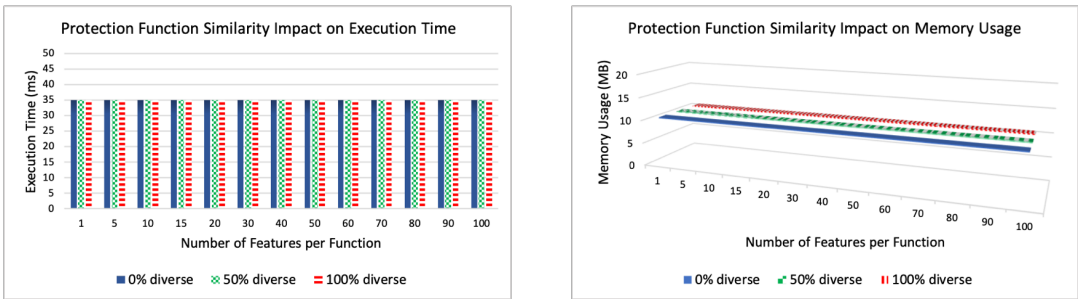

FIGURE 5.9: Protection Function Similarity Impact

Case 2: To study the impact of multi-attribute handling, we incorporate multithreading features in order to perform parallel execution of the process on an increasing number of attributes. We consider the following number for each iteration: 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100. Figure 5.10 shows that increasing the number of attributes has a quasi-linear impact on the total execution time, with an average time of 35ms for 5 attributes and up to 100ms for 100 attributes. The RAM usage remains

constant with an average of 10MB. This highlights the importance of integrating a low-cost transition matrix.

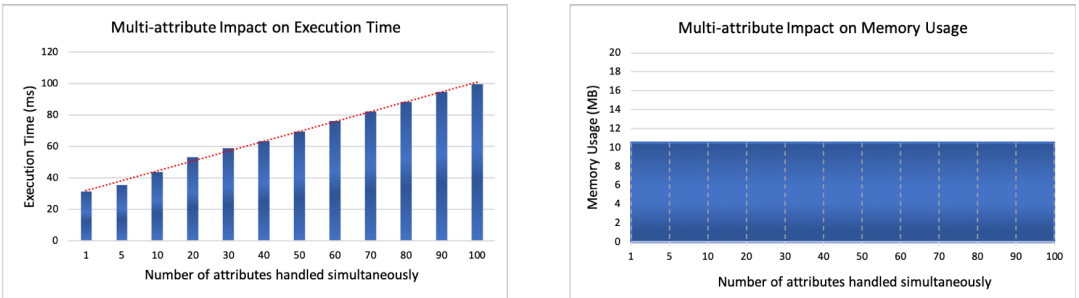

FIGURE 5.10: Multi-attribute Impact

THEOREM 6. The P-SGD process maintains low storage complexity.

PROOF. Let n denotes the maximum number of *attributes* that could be shared by the user with data consumers. As previously mentioned in Section 5.3, the solution stores only the three properties of the last data value for each attribute in $Trans$, and the values of $c_i \in C$, resulting in a linear storage complexity of $O(4n)$. However, the number of *attributes* shared by the user will not practically exceed 100, which makes the storage complexity low. □

Discussion. The experiments conducted show that P-SGD is scalable and efficient in time and space (cf. Challenge 4). The solution is able to maintain effective performance in different scenarios, including worst-case ones. This increases its reusability to also include privacy models that require real-time reasoning, and allows it to operate on a variety of devices, including resource-constrained ones.

5.5 Privacy Models Background

Several approaches have been proposed in the literature to address the challenges of security and privacy in the fields of pervasive Internet of Things (IoT) environments (connected environments). However, to the best of our knowledge, this is the first work to tackle the problem of preserving user privacy against data inference attacks during protection transitions. Therefore, we discuss in this section existing privacy-preserving models to which our P-SGD solution could be connected.

Balancing data utility-privacy has received extensive attention in the last decade. Existing approaches vary from context-aware to personalized privacy-preserving. In our research work, we proposed CaPMan [5], a user-centric context-aware model for privacy management in connected environments that meets current privacy standards (i.e., Privacy by Design and ISO/IEC 27701 standards). Matos et al. [145]

proposed a context-aware security approach, that provides authentication, authorization, access control, and privacy-preserving to fog and edge computing environments. Gheisari et al. [153] introduced a context-aware privacy-preserving approach for IoT-based smart city using Software Defined Networking. Sylla et al. [144] presented a context-aware security and privacy as a service (CASPaaS) architecture to inform the user about the contextual risks involved. Gao et al. [142] proposed a personalized anonymization model for balancing trajectory privacy and data utility. Qiu et el. [146] provided a semantic-aware personalized privacy model that studies user requirements and location's privacy sensitivity to adapt the trajectory construction accordingly. Xiong et al. [147] proposed a personalized privacy protection model based on game theory and data encryption.

5.6 Summary

We introduce in this chapter a new privacy-preserving stochastic gradient descent solution (P-SGD) that can be integrated into numerous existing privacy models in order to provide an additional layer of protection against data inference attacks during protection transitions. P-SGD features an iterative non-deterministic process that gradually decreases the data protection level during the protection descent phases. This allows preserving an appropriate precision gap between sequential protected data values to avoid potential data leakages.

Chapter 6

Conclusion & Future Work

"I am not the product of my circumstances. I am the product of my decisions."

– Stephen Covey

6.1 Report Recap

The study presented in this thesis focuses mainly on privacy risk inference and management in connected environments.

In Chapter 1 we give the reader an insight on connected environments, and why privacy in the context of these environments is considered a topic of interest nowadays. Specifically, we discuss current privacy threats and challenges encountered in these environments, as well as existing international privacy regulations and standards. Then, we focus on our thesis's objectives of raising user awareness of the privacy risks involved in their situations, assisting users in the optimization of their data utility-privacy decisions based on their preferences and situations, and ensuring appropriate protection of the data collected before being transmitted to data consumers. We present a real-life scenario of a user situation and illustrate some of the privacy risks involved in this latter in order to showcase the motivation behind this work and the challenges that lie ahead. Following that, we review existing works of context-aware privacy management in connected environments according to the identified needs and challenges. Then, we present our proposed framework for Context-aware Privacy Management in connected environments (CaPMan) and detail its corresponding modules that answer the objectives and address the set of needs and challenges. Finally, we list the publications related to this report before introducing the following chapters.

In Chapter 2 we present an ontology-based data model for user-Context modeling in Sensor Networks (uCSN) where we improve the context representation to consider diverse types of: (i) user/environment information (i.e., scalar and multimedia information); (ii) data sources (e.g., sensor, device, social network profile,

document); (iii) uncertainties (e.g., uncertainties related to the user and the environment); and (iv) environments (i.e., connected/unconnected environments, and environments with static/mobile systems and devices). We do so by defining new concepts and properties, and importing others from well-known ontologies, namely DPV, SSN, HSSN, and W3C Uncertainty ontologies. The uCSN ontology is generic and re-usable in different application domains. Finally, we evaluate the accuracy of our additions, their clarity, consistency, and the overall impact on performance.

In Chapter 3 we present a Context-aware Semantic reasoning approach for Privacy risk Inference (CaSPI). This approach is equipped with a semantic rule-based reasoner that is used to infer the risks involved in user situations. To achieve this, CaSPI relies on the use of ontologies (e.g., uCSN ontology) and inference rules that respectively represent contextual knowledge and define the risks to be detected by the reasoner with high semantic expressiveness power. CaSPI is generic and re-usable in several domains. It is capable of providing the user with a dynamic overview of risks that copes with the evolution of her situation and is tailored to her expertise. This allows all users to understand their privacy situations, paving the way for them to make informed data privacy decisions. We developed a prototype to validate our proposal, illustrated its operation from both the back-end and front-end, and evaluated its performance in several scenarios.

In Chapter 4 we introduce a user-centric multi-objective approach for context-aware privacy management in connected environments (δ-*Risk*). This approach features a new privacy risk quantification model to dynamically calculate and select the best data protection strategies for the user based on her situation and preferences. Computed strategies are optimal in that they seek to closely satisfy user requirements and preferences, while also maximizing data utility and minimizing the cost of protection. We developed a prototype to validate our proposal and illustrated its functioning from both back-end and front-end perspectives. We also evaluated its performance by considering multiple cases, and formally studied its effectiveness in best strategy identification.

In Chapter 5 we propose a new stochastic gradient descent solution for privacy-preserving during protection transitions (P-SGD). The proposed approach can be connected to numerous existing privacy models, providing an additional layer of protection against data inference attacks during protection transitions. P-SGD features an iterative non-deterministic process that gradually decreases the data protection level during the protection descent phases. It is capable of measuring data dependencies as well as similarity in protection functions, and adapt the descent rate accordingly. This allows preserving an appropriate precision gap between sequential protected data values, avoiding consequently potential data leakages. We developed a prototype to validate our proposal, and we evaluated its performance in multiple scenarios.

6.2 Future Research Directions

Various improvements still need to be considered for this work. We detail future research directions for each contribution separately.

6.2.1 Context Modeling in Connected Environments

Completeness Evaluation. We would like to continue the ongoing evaluation of the completeness of the uCSN ontology through comparisons with situation, user, environment, and mobility taxonomies. This evaluation will potentially help us discover missing concepts or properties that could complement uCSN.

6.2.2 Privacy Risk Inference

Privacy Rules Validity, Dependencies and Conflicts. We aim to address the challenges of verifying the validity of defined privacy rules, as well as the existing dependencies and conflicts between them. For the validity, we would like to consider two validation aspects: (i) testing aspect, which involves evaluating the accuracy of the defined rule in several different scenarios prior to its consideration by the CaSPI reasoner; (ii) human aspect, which involves checking and validating it by a group of privacy experts following the development of the outsourcing solution. In order to manage rule dependencies and conflicts, we would like to proceed with comparing existing rules by measuring the similarities between their related sequences of data elements, as well as the similarities of their associated sets of sensitive information.

Outsourcing Solution for Rule Definitions. We aim to develop and implement the outsourcing solution for the privacy rule definitions with a group of privacy experts in order to provide a high-level risk coverage in various application domains.

Privacy Rules Implementation. Current semantic rule languages (e.g., SWRL [114]) presents some limitations when considering spatial, temporal, and logical operators (e.g., logical disjunction, negation) to define rules. We aim consequently to address these limitations by proposing a new built-in for the extensible SWRL language that enables the use of spatio-temporal and expanded logical operators.

CaSPI Evaluation. Once the development of the mobile application is complete, we would like to extend the evaluation of the CaSPI proposal to also evaluate the time required by users of different profiles to interact with the application (e.g., inputs specification).

Inter-Context Risk Coverage. At this point, the CaSPI proposal reasons over each context separately (i.e., intra-context information reasoning), without considering inter-context patterns and their impact on the privacy situation of the user. For instance, a user located every Tuesday in a sports gym can lead to disclose her regular activity, and also to predict her future time of presence at home on Tuesdays. Such

risks are not currently covered by our proposal. To overcome this limitation, we aim to improve the risk coverage, by allowing for the definition of inter-context privacy rules, and the reasoning over historical contexts to identify the plausible information patterns based on their time and spatial dependencies.

6.2.3 Privacy Risk Management

Privacy Risk Quantification. We would like to improve our risk quantification model to also consider the uncertainty aspects of information elements and their impact on associated risk values.

***δ-Risk* Evaluation.** Once the development of the mobile application is complete, we would like to extend the evaluation of the *δ-Risk* proposal to also evaluate the time costs of user interactions with the application, such as to specify their inputs (e.g., preferences, sensed data), making their privacy decisions by choosing the protection strategy to implement, and so on.

6.2.4 Privacy-preserving during Protection Transitions

Data Dependency. Sensor data are spatio-temporal in nature [26], which means that in addition to their temporal correlations, they also hold spatial correlations that must be considered when measuring data dependency. In addition, the spatial and temporal distances between generated data can vary according to the user's context. For example, distances between location data vary whether the user is driving a vehicle, running, or walking. Therefore, we aim to improve the data dependency measurement by introducing a three-dimensional dependency graph of temporal, spatial, and contextual dimensions.

Time Dependency of Data. We considered in this study that the standard time period during which two data values of an attribute are said to be dependent (i.e., $\boldsymbol{a.\tau}$) is provided as input to the system. As future work, we aim to automate the computing of $\boldsymbol{a.\tau}$, which could be calculated based on several metrics, such as the historical data distribution in time (e.g., regular/irregular time series) and the velocity of data value changes and relative gaps.

Protection Function Similarity. The current similarity measurement of protection functions takes into consideration the exact match between functions' classes and feature lists. We thus aim to improve the similarity measurement to further consider the semantic similarity of their features.

6.2.5 CaPMan Framework

CaPMan Implementation. We are developing the CaPMan mobile application that assists the user in managing her privacy based on her situation. This application consists of the context acquisition, context modeling, risk reasoner, risk manager,

and the data protection engines. The implementation has the highest priority from all future work, since it allows for the testing of our CaPMan framework's accuracy in real-world scenarios, as well as the end to end evaluation of the entire framework operations.

Crowdsourcing Solution for Environment Modeling. We aim to integrate in the mobile application a crowdsourcing solution that allows all CaPMan users for manipulating the maps of their environments, such as adding new components (e.g., sensors, devices, actuators) and reporting updates in their environments (e.g., location change of a camera in the mall). This will improve the quality of information coverage, and enable users to practically exchange information about their environments and contribute to the reinforcement of their privacy protection.

Data Protection. We would like to explore the *data protection* module of CaPMan, and specifically address the challenges of: (i) protection functions selection, which can depend on several metrics, such as the computational cost, vulnerabilities to data inference, and compatibility with attribute type and data format; and (ii) System vulnerability assessment in the face of security threats.

CaPMan Extension. The proposed CaPMan framework is user-centric. As future work, we aim to expand the indexing of the framework to make it entity-centric, where an entity could be a user or an environment (e.g., company, mall).

Bibliography

[1] A. Polleres, B. Esteves, B. Bos, B. Bruegger, E. Kiesling, E. Schlehahn, F. J. Ekaputra, G. P. Krog, H. J. Pandit, J. D. Fernández, M. Lizar, P. Ryan, P. Bonatti, R. G. Hamed, R. Wenning, R. Brennan, and S. Steyskal, "Data Privacy Vocabulary (dpv)," 2021, `https://w3.org/ns/dpv`.

[2] A. Haller, K. Janowicz, S. J. Cox, M. Lefrançois, K. Taylor, D. Le Phuoc, J. Lieberman, R. García-Castro, R. Atkinson, and C. Stadler, "The modular ssn ontology: A joint w3c and ogc standard specifying the semantics of sensors, observations, sampling, and actuation," *Semantic Web*, vol. 10, no. 1, pp. 9–32, 2019.

[3] K. Laskey, K. Laskey, P. Costa, M. Kokar, T. Martin, and T. Lukasiewicz (eds.), "Uncertainty reasoning for the world wide web:w3c incubator group report," *World Wide Web Consortium*, 2005, `http://www.w3.org/2005/Incubator/urw3/XGR-urw3/`.

[4] K. B. Chaaya, M. Barhamgi, R. Chbeir, P. Arnould, and D. Benslimane, "Context-aware system for dynamic privacy risk inference: Application to smart iot environments," *Future Generation Computer Systems*, vol. 101, pp. 1096–1111, 2019.

[5] K. Bou-Chaaya, R. Chbeir, M. N. Alraja, P. Arnould, C. Perera, M. Barhamgi, and D. Benslimane, "δ-Risk: Toward context-aware multi-objective privacy management in connected environments," *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 2, pp. 1–31, 2021.

[6] K. Bou-Chaaya, R. Chbeir, M. Barhamgi, P. Arnould, and D. Benslimane, "P-SGD: A stochastic gradient descent solution for privacy-preserving during protection transitions," in *International Conference on Advanced Information Systems Engineering*, Springer, 2021, pp. 37–53.

[7] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[8] S. D. Warren and L. D. Brandeis, "Right to privacy," *Harv. L. Rev.*, vol. 4, p. 193, 1890.

[9] A. F. Westin, "Privacy and freedom," *Washington and Lee Law Review*, vol. 25, no. 1, p. 166, 1968.

[10] J. H. Ziegeldorf, O. G. Morchon, and K. Wehrle, "Privacy in the internet of things: Threats and challenges," *Security and Communication Networks*, vol. 7, no. 12, pp. 2728–2742, 2014.

[11] N. Vollmer, *Table of contents EU General Data Protection Regulation (EU-GDPR)*, en, 2018.

[12] E. McCallister, T. Grance, and K. Scarfone, "Guide to protecting the confidentiality of personally identifiable information (PII)," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep. NIST SP 800-122, 2010.

[13] State of California Department of Justice, *California Consumer Privacy Act (CCPA)*, en, 2018.

[14] S.-C. Cha, T.-Y. Hsu, Y. Xiang, and K.-H. Yeh, "Privacy enhancing technologies in the internet of things: Perspectives and challenges," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2159–2187, 2018.

[15] M. M. Ogonji, G. Okeyo, and J. M. Wafula, "A survey on privacy and security of internet of things," *Computer Science Review*, vol. 38, p. 100 312, 2020.

[16] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, "Vision and challenges for realising the internet of things," *Cluster of European research projects on the internet of things, European Commision*, vol. 3, no. 3, pp. 34–36, 2010.

[17] S. Pearson, "Taking account of privacy when designing cloud computing services," in *2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, IEEE, 2009, pp. 44–52.

[18] A. Levin and M. J. Nicholson, "Privacy law in the united states, the eu and canada: The allure of the middle ground," *U. Ottawa L. & Tech. J.*, vol. 2, p. 357, 2005.

[19] G. Greenleaf, "Global data privacy laws 2019: 132 national laws & many bills," 2019.

[20] G. D'Acquisto, J. Domingo-Ferrer, P. Kikiras, V. Torra, Y.-A. de Montjoye, and A. Bourka, "Privacy by design in big data: An overview of privacy enhancing technologies in the era of big data analytics," *arXiv preprint arXiv:1512.06000*, 2015.

[21] A. Cavoukian and M. Chibba, "Start with privacy by design in all big data applications," in *Guide to big data applications*, Springer, 2018, pp. 29–48.

[22] A. Cavoukian, "Privacy by design [leading edge]," *IEEE Technology and Society Magazine*, vol. 31, no. 4, pp. 18–19, 2012.

[23] *ISO/AWI 31700 Consumer protection — Privacy by design for consumer goods and services*, en, 2021.

[24] ISO/IEC, "ISO/IEC 27701:2019 Security techniques — Extension to ISO/IEC 27001 and ISO/IEC 27002 for privacy information management — Requirements and guidelines," 2019, https://www.iso.org/standard/71670.html.

[25] ——, "ISO/IEC 29100:2011 Information technology — Security techniques — Privacy framework," 2011, https://www.iso.org/standard/45123.html.

[26] B. George, J. M. Kang, and S. Shekhar, "Spatio-temporal sensor graphs (stsg): A data model for the discovery of spatio-temporal patterns," *Intelligent Data Analysis*, vol. 13, no. 3, pp. 457–475, 2009.

[27] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker, "Inferring personal information from demand-response systems," *IEEE Security & Privacy*, vol. 8, no. 1, pp. 11–20, 2010.

[28] M. Barhamgi, C. Perera, C. Ghedira, and D. Benslimane, "User-centric privacy engineering for the internet of things," *IEEE Cloud Computing*, vol. 5, no. 5, pp. 47–57, 2018.

[29] "Data in the post-gdpr world," *Computer Fraud & Security*, vol. 2018, no. 9, pp. 17 –18, 2018, ISSN: 1361-3723.

[30] "Marketing firm exactis leaks 340 million files containing private data," *Mail Online*, 2018.

[31] C. Castelluccia, M. Cunche, D. Le Metayer, and V. Morel, "Enhancing transparency and consent in the iot," in *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, IEEE, 2018, pp. 116–119.

[32] Y. Verginadis, A. Michalas, P. Gouvas, G. Schiefer, G. Hübsch, and I. Paraskakis, "Paasword: A holistic data privacy and security by design framework for cloud services," *Journal of Grid Computing*, vol. 15, no. 2, pp. 219–234, 2017.

[33] I. D. Addo, S. I. Ahamed, S. S. Yau, and A. Buduru, "A reference architecture for improving security and privacy in internet of things applications," in *2014 IEEE International conference on mobile services*, IEEE, 2014, pp. 108–115.

[34] D. W. Chadwick and K. Fatema, "A privacy preserving authorisation system for the cloud," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1359–1373, 2012.

[35] V. Varadharajan and S. Bansal, "Data security and privacy in the internet of things (iot) environment," in *Connectivity Frameworks for Smart Devices*, Springer, 2016, pp. 261–281.

[36] V. Y. Pillitteri and T. L. Brewer, "Guidelines for smart grid cybersecurity," National Institute of Standards and Technology, Tech. Rep. NISTIR 7628 Revision 1, 2014. DOI: https://doi.org/10.6028/NIST.IR.7628r1.

[37] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker, "Inferring personal information from demand-response systems," *IEEE Security & Privacy*, vol. 8, no. 1, pp. 11–20, 2010.

[38] R. Neisse, G. Steri, G. Baldini, E. Tragos, I. N. Fovino, and M. Botterman, "Dynamic context-aware scalable and trust-based iot security, privacy framework," *Chapter in Internet of Things Applications-From Research and Innovation to Market Deployment, IERC Cluster Book*, 2014.

[39] E. de Matos, R. T. Tiburski, L. A. Amaral, and F. Hessel, "Providing context-aware security for iot environments through context sharing feature," in *17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (TrustCom)*, IEEE, 2018, pp. 1711–1715.

[40] T. Sylla, M. A. Chalouf, F. Krief, and K. Samaké, "Towards a context-aware security and privacy as a service in the internet of things," in *IFIP International Conference on Information Security Theory and Practice*, 2019, pp. 240–252.

[41] M. Gheisari, G. Wang, W. Z. Khan, and C. Fernández-Campusano, "A context-aware privacy-preserving method for iot-based smart city using software defined networking," *Computers & Security*, vol. 87, p. 101 470, 2019.

[42] V. Alagar, A. Alsaig, O. Ormandjiva, and K. Wan, "Context-based security and privacy for healthcare iot," in *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*, IEEE, 2018, pp. 122–128.

[43] E. Mansour, R. Chbeir, and P. Arnould, "Hssn: An ontology for hybrid semantic sensor networks," in *Proceedings of the 23rd International Database Applications & Engineering Symposium*, 2019, pp. 1–10.

[44] A. R. M. Forkan, I. Khalil, A. Ibaida, and Z. Tari, "Bdcam: Big data for context-aware monitoring—a personalized knowledge discovery framework for assisted healthcare," *IEEE transactions on cloud computing*, vol. 5, no. 4, pp. 628–641, 2015.

[45] D.-O. Kang, H.-J. Lee, E.-J. Ko, K. Kang, and J. Lee, "A wearable context aware system for ubiquitous healthcare," in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, 2006, pp. 5192–5195.

[46] H. K. Pung, T. Gu, W. Xue, P. P. Palmes, J. Zhu, W. L. Ng, C. W. Tang, and N. H. Chung, "Context-aware middleware for pervasive elderly homecare," *IEEE Journal on Selected Areas in communications*, vol. 27, no. 4, pp. 510–524, 2009.

[47] M. Alirezaie, J. Renoux, U. Köckemann, A. Kristoffersson, L. Karlsson, E. Blomqvist, N. Tsiftes, T. Voigt, and A. Loutfi, "An ontology-based context-aware system for smart homes: E-care@ home," *Sensors*, vol. 17, no. 7, p. 1586, 2017.

[48] Q. Ni, A. B. García Hernando, and I. Pau de la Cruz, "A context-aware system infrastructure for monitoring activities of daily living in smart home," *Journal of Sensors*, vol. 2016, 2016.

[49] D. Schürholz, S. Kubler, and A. Zaslavsky, "Artificial intelligence-enabled context-aware air quality prediction for smart cities," *Journal of Cleaner Production*, vol. 271, p. 121 941, 2020.

[50] Z. Khan, S. L. Kiani, and K. Soomro, "A framework for cloud-based context-aware information services for citizens in smart cities," *Journal of Cloud Computing*, vol. 3, no. 1, pp. 1–17, 2014.

[51] A. Castiglione, K.-K. R. Choo, M. Nappi, and S. Ricciardi, "Context aware ubiquitous biometrics in edge of military things," *IEEE Cloud Computing*, vol. 4, no. 6, pp. 16–20, 2017.

[52] K. Meehan, T. Lunney, K. Curran, and A. McCaughey, "Context-aware intelligent recommendation system for tourism," in *2013 IEEE international conference on pervasive computing and communications workshops (PERCOM workshops)*, IEEE, 2013, pp. 328–331.

[53] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 414–454, 2013.

[54] T. Strang and C. Linnhoff-Popien, "A context modeling survey," in *Workshop Proceedings*, 2004.

[55] R. Hervás, J. Bravo, and J. Fontecha, "A context model based on ontological languages: A proposal for information visualization.," *J. UCS*, vol. 16, no. 12, pp. 1539–1555, 2010.

[56] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung, "Ontology based context modeling and reasoning using owl," in *IEEE annual conference on pervasive computing and communications workshops, 2004. Proceedings of the second*, Ieee, 2004, pp. 18–22.

[57] H. Chen, T. Finin, and A. Joshi, "The soupa ontology for pervasive computing," in *Ontologies for agents: Theory and experiences*, Springer, 2005, pp. 233–258.

[58] J. Kim and K.-Y. Chung, "Ontology-based healthcare context information model to implement ubiquitous environment," *Multimedia Tools and Applications*, vol. 71, no. 2, pp. 873–888, 2014.

[59] H. Chen, T. Finin, A. Joshi, *et al.*, "An ontology for context-aware pervasive computing environments," in *Workshop on Ontologies and Distributed Systems*, 2003.

[60] D. Preuveneers, J. Van den Bergh, D. Wagelaar, A. Georges, P. Rigole, T. Clerckx, Y. Berbers, K. Coninx, V. Jonckers, and K. De Bosschere, "Towards an extensible context ontology for ambient intelligence," in *European Symposium on Ambient Intelligence*, Springer, 2004, pp. 148–159.

[61] M. Poveda-Villalón, M. C. Suárez-Figueroa, R. García-Castro, and A. Gómez-Pérez, "A context ontology for mobile environments," in *CIAO@ EKAW*, Citeseer, 2010.

[62] D. Riboni and C. Bettini, "Owl 2 modeling and reasoning with complex human activities," *Pervasive and Mobile Computing*, vol. 7, no. 3, pp. 379–395, 2011.

[63] M. Rasmussen, M. Lefrançois, G. Schneider, and P. Pauwels, "Bot: The Building Topology Ontology of the W3C linked building data group," 2021, `https://w3c-lbd-cg.github.io/bot/`.

[64] S. Kulkarni *et al.*, "Context aware recommendation systems: A review of the state of the art techniques," *Computer Science Review*, vol. 37, p. 100 255, 2020.

[65] P. Yang, D. Stankevicius, V. Marozas, Z. Deng, E. Liu, A. Lukosevicius, F. Dong, L. Xu, and G. Min, "Lifelogging data validation model for internet of things enabled personalized healthcare," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 1, pp. 50–64, 2016.

[66] M. Wu and R. Balakrishnan, "Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays," in *Proceedings of the 16th annual ACM symposium on User interface software and technology*, 2003, pp. 193–202.

[67] A. K. Dey, G. D. Abowd, and D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *Human–Computer Interaction*, vol. 16, no. 2-4, pp. 97–166, 2001.

[68] B. N. Schilit and M. M. Theimer, "Disseminating active map information to mobile hosts," *IEEE network*, vol. 8, no. 5, pp. 22–32, 1994.

[69] P. J. Brown, "The stick-e document: A framework for creating context-aware applications," *Electronic Publishing-Chichester-*, vol. 8, pp. 259–272, 1995.

[70] D. Franklin and J. Flaschbart, "All gadget and no representation makes jack a dull environment," in *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, 1998, pp. 155–160.

[71] T. Rodden, K. Cheverst, K Davies, and A. Dix, "Exploiting context in hci design for mobile systems," in *Workshop on human computer interaction with mobile devices*, Citeseer, vol. 12, 1998.

[72] R. Hull, P. Neaves, and J. Bedford-Roberts, "Towards situated computing," in *Digest of papers. first international symposium on wearable computers*, IEEE, 1997, pp. 146–153.

[73] A. Ward, A. Jones, and A. Hopper, "A new location technique for the active office," *IEEE Personal communications*, vol. 4, no. 5, pp. 42–47, 1997.

[74] G. D. Abowd and E. D. Mynatt, "Charting past, present, and future research in ubiquitous computing," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 7, no. 1, pp. 29–58, 2000.

[75] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *1994 First Workshop on Mobile Computing Systems and Applications*, IEEE, 1994, pp. 85–90.

[76] J. Pascoe, "Adding generic contextual capabilities to wearable computers," in *Digest of papers. second international symposium on wearable computers (cat. no. 98ex215)*, IEEE, 1998, pp. 92–99.

[77] A. K. Dey and J. Mankoff, "Designing mediation for context-aware applications," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 12, no. 1, pp. 53–80, 2005.

[78] N. S. Ryan, J. Pascoe, and D. R. Morse, "Enhanced reality fieldwork: The context-aware archaeological assistant," in *Computer applications in archaeology*, Tempus Reparatum, 1998.

[79] D. Thevenin and J. Coutaz, "Plasticity of user interfaces: Framework and research agenda.," in *Interact*, vol. 99, 1999, pp. 110–117.

[80] G. Piao and J. G. Breslin, "Inferring user interests in microblogging social networks: A survey," *User Modeling and User-Adapted Interaction*, vol. 28, no. 3, pp. 277–329, 2018.

[81] D. Brickley and L. Miller, "FOAF Vocabulary Specification," 2014, `http://xmlns.com/foaf/spec/`.

[82] K.-L. Skillen, L. Chen, C. D. Nugent, M. P. Donnelly, W. Burns, and I. Solheim, "Ontological user profile modeling for context-aware application personalization," in *International conference on ubiquitous computing and ambient intelligence*, Springer, 2012, pp. 261–268.

[83] M. Sutterer, O. Droegehorn, and K. David, "Upos: User profile ontology with situation-dependent preferences support," in *First International Conference on Advances in Computer-Human Interaction*, IEEE, 2008, pp. 230–235.

[84] J. Stan, E. Egyed-Zsigmond, A. Joly, and P. Maret, "A user profile ontology for situation-aware social networking," in *3rd Workshop on Artificial Intelligence Techniques for Ambient Intelligence (AITAmI2008)*, 2008.

[85] G. Klyne, "Composite capability/preference profiles (CC/PP): Structure and vocabularies," *W3C working draft*, 2004.

[86] K.-L. Skillen, L. Chen, C. D. Nugent, M. P. Donnelly, and I. Solheim, "A user profile ontology based approach for assisting people with dementia in mobile environments," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, 2012, pp. 6390–6393.

[87] N. Seydoux, K. Drira, N. Hernandez, and T. Monteil, "Iot-o, a core-domain iot ontology to represent connected devices networks," in *European Knowledge Acquisition Workshop*, Springer, 2016, pp. 561–576.

[88] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, *et al.*, "The ssn ontology of the w3c semantic sensor network incubator group," *Journal of Web Semantics*, vol. 17, pp. 25–32, 2012.

[89] L. Spalazzi, G. Taccari, and A. Bernardini, "An internet of things ontology for earthquake emergency evaluation and response," in *2014 International Conference on Collaboration Technologies and Systems (CTS)*, IEEE, 2014, pp. 528–534.

[90] R. Styles and N. Shabir, "Lifecycle Schema," 2008, `https://vocab.org/lifecycle/schema`.

[91] J. Kopeckỳ, K. Gomadam, and T. Vitvar, "Hrests: An html microformat for describing restful web services," in *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, IEEE, vol. 1, 2008, pp. 619–625.

[92] C. Pedrinaci, J. Domingue, *et al.*, "Toward the next wave of services: Linked services for the web of data.," *J. ucs*, vol. 16, no. 13, pp. 1694–1719, 2010.

[93] T. Vitvar, J. Kopeckỳ, J. Viskova, and D. Fensel, "Wsmo-lite annotations for web services," in *European Semantic Web Conference*, Springer, 2008, pp. 674–689.

[94] D. Bonino, F. Corno, and L. De Russis, "Poweront: An ontology-based approach for power consumption estimation in smart homes," in *International Internet of Things Summit*, Springer, 2014, pp. 3–8.

[95] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor, "Iot-lite: A lightweight semantic model for the internet of things and its use with dynamic semantics," *Personal and Ubiquitous Computing*, vol. 21, no. 3, pp. 475–487, 2017.

[96] K. Kotis and A. Katasonov, "An iot-ontology for the representation of interconnected, clustered and aligned smart entities," *Technical report, VTT Technical Research Center, Finland VTT Technical Research Center, Finland*, 2012.

[97] J. R. Hobbs and F. Pan, "W3C Time Ontology in OWL," 2017, `https://www.w3.org/TR/owl-time/`.

[98] C. Angsuchotmetee, R. Chbeir, and Y. Cardinale, "Mssn-onto: An ontology-based approach for flexible event processing in multimedia sensor networks," *Future Generation Computer Systems*, vol. 108, pp. 1140–1158, 2020.

[99] C. C. Aggarwal, *Managing and mining sensor data*. Springer Science & Business Media, 2013.

[100] S. Staab and R. Studer, *Handbook on ontologies*. Springer Science & Business Media, 2010.

[101] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen, "Security and privacy in smart city applications: Challenges and solutions," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 122–129, 2017.

[102] K. Shilton, "Four billion little brothers? privacy, mobile phones, and ubiquitous data collection," *Communications of the ACM*, vol. 52, no. 11, pp. 48–53, 2009.

[103] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker, "Inferring personal information from demand-response systems," *IEEE Security & Privacy*, vol. 8, no. 1, pp. 11–20, 2010.

[104] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[105] J. P. Kolter, *User-centric Privacy: A Usable and Provider-independent Privacy Infrastructure*. BoD–Books on Demand, 2010, vol. 41.

[106] B. P. Knijnenburg, "Simplifying privacy decisions: Towards interactive and adaptive solutions.," in *Decisions@ RecSys*, Citeseer, 2013, pp. 40–41.

[107] D. Christin, M. Michalak, and M. Hollick, "Raising user awareness about privacy threats in participatory sensing applications through graphical warnings," in *Proceedings of International Conference on Advances in Mobile Computing & Multimedia*, 2013, pp. 445–454.

[108] I. Wagner, Y. He, D. Rosenberg, and H. Janicke, "User interface design for privacy awareness in ehealth technologies," in *2016 13th IEEE annual consumer communications & networking conference (CCNC)*, IEEE, 2016, pp. 38–43.

[109] A. I. Abdelmoty and F. Alrayes, "Towards understanding location privacy awareness on geo-social networks," *ISPRS International Journal of Geo-Information*, vol. 6, no. 4, p. 109, 2017.

[110] J. Alemany, E. del Val, J Alberola, and A. García-Fornes, "Enhancing the privacy risk awareness of teenagers in online social networks through soft-paternalism mechanisms," *International Journal of Human-Computer Studies*, vol. 129, pp. 27–40, 2019.

[111] G. Petkos, S. Papadopoulos, and Y. Kompatsiaris, "Pscore: A framework for enhancing privacy awareness in online social networks," in *2015 10th International Conference on Availability, Reliability and Security*, IEEE, 2015, pp. 592–600.

[112] J. Krumm, "Inference attacks on location tracks," in *International Conference on Pervasive Computing*, Springer, 2007, pp. 127–143.

[113] S. J. De and D. Le Métayer, "Priam: A privacy risk analysis methodology," in *Data privacy management and security assurance*, Springer, 2016, pp. 221–229.

[114] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, *et al.*, "Swrl: A semantic web rule language combining owl and ruleml," *W3C Member submission*, vol. 21, no. 79, pp. 1–31, 2004.

[115] B. Klotz, R. Troncy, D. Wilms, and C. Bonnet, "Vsso: The vehicle signal and attribute ontology.," in *SSN@ ISWC*, 2018, pp. 56–63.

[116] M. Alirezaie, J. Renoux, U. Köckemann, A. Kristoffersson, L. Karlsson, E. Blomqvist, N. Tsiftes, T. Voigt, and A. Loutfi, "An ontology-based context-aware system for smart homes: E-care@ home," *Sensors*, vol. 17, no. 7, p. 1586, 2017.

[117] X. Fiorentini, S. Rachuri, H. Suh, J. Lee, and R. D. Sriram, "An analysis of description logic augmented with domain rules for the development of product models," *Journal of computing and information science in engineering*, vol. 10, no. 2, 2010.

[118] B. C. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Computing Surveys (Csur)*, vol. 42, no. 4, pp. 1–53, 2010.

[119] L. Sweeney, "K-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[120] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, 3–es, 2007.

[121] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond k-anonymity and l-diversity," in *2007 IEEE 23rd International Conference on Data Engineering*, IEEE, 2007, pp. 106–115.

[122] J. Cao *et al.*, "Castle: Continuously anonymizing data streams," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, 2010.

[123] C. C. Aggarwal and S. Y. Philip, "A general survey of privacy-preserving data mining models and algorithms," in *Privacy-preserving data mining*, Springer, 2008, pp. 11–52.

[124] M. Z. Islam and L. Brankovic, "Privacy preserving data mining: A noise addition framework using a novel clustering technique," *Knowledge-Based Systems*, 2011.

[125] E. G. Komishani, M. Abadi, and F. Deldar, "Pptd: Preserving personalized privacy in trajectory data publishing by sensitive attribute generalization and trajectory local suppression," *Knowledge-Based Systems*, vol. 94, pp. 43–59, 2016.

[126] M. Sharma, A. Chaudhary, M. Mathuria, and S. Chaudhary, "A review study on the privacy preserving data mining techniques and approaches," *International Journal of Computer Science and Telecommunications*, vol. 4, no. 9, pp. 42–46, 2013.

[127] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*, Springer, 2008, pp. 1–19.

[128] W. Fang, X. Z. Wen, Y. Zheng, and M. Zhou, "A survey of big data security and privacy preserving," *IETE Technical Review*, vol. 34, no. 5, pp. 544–560, 2017.

[129] S. Oulmakhzoune, N. Cuppens-Boulahia, F. Cuppens, S. Morucci, M. Barhamgi, and D. Benslimane, "Privacy query rewriting algorithm instrumented by a privacy-aware access control model," *Annales des Télécommunications*, vol. 69, no. 1-2, pp. 3–19, 2014. DOI: 10.1007/s12243-013-0365-8. [Online]. Available: https://doi.org/10.1007/s12243-013-0365-8.

[130] M. Barhamgi, D. Benslimane, Y. Amghar, N. Cuppens-Boulahia, and F. Cuppens, "Privcomp: A privacy-aware data service composition system," in *Proceedings of the 16th International Conference on Extending Database Technology*, Citeseer, 2013, pp. 757–760.

[131] M. Barhamgi, A. K. Bandara, Y. Yu, K. Belhajjame, and B. Nuseibeh, "Protecting privacy in the cloud: Current practices, future directions," *IEEE Computer*, vol. 49, no. 2, pp. 68–72, 2016. DOI: 10.1109/MC.2016.59. [Online]. Available: https://doi.org/10.1109/MC.2016.59.

[132] A. S. Householder, *The theory of matrices in numerical analysis*. Courier Corporation, 2013.

[133] D Nagarajan, T Tamizhi, M Lathamaheswari, and J Kavikumar, "Traffic control management using gauss jordan method under neutrosophic environment," in *AIP Conference Proceedings*, vol. 2112, 2019.

[134] L. Shang, S. Petiton, and M. Hugues, "A new parallel paradigm for block-based gauss-jordan algorithm," in *2009 Eighth International Conference on Grid and Cooperative Computing*, 2009, pp. 193–200.

[135] L. M. Aouad and S. G. Petiton, "Parallel basic matrix algebra on the grid'5000 large scale distributed platform," in *2006 IEEE International Conference on Cluster Computing*, 2006, pp. 1–8.

[136] L. Shang, Z. Wang, S. G. Petiton, Y. Lou, and Z. Liu, "Large scale computing on component based framework easily adaptive to cluster and grid environments," in *The Third ChinaGrid Annual Conference*, IEEE, 2008, pp. 70–77.

[137] L. M. Aouad, S. G. Petiton, and M. Sato, "Grid and cluster matrix computation with persistent storage and out-of-core programming," in *2005 IEEE International Conference on Cluster Computing*, IEEE, 2005, pp. 1–9.

[138] M. Xue, P. Kalnis, and H. K. Pung, "Location diversity: Enhanced privacy protection in location based services," in *International Symposium on Location- and Context-Awareness*, Springer, 2009, pp. 70–87.

[139] A. Chernev, U. Böckenholt, and J. Goodman, "Choice overload: A conceptual review and meta-analysis," *Journal of Consumer Psychology*, vol. 25, no. 2, pp. 333–358, 2015.

[140] M. Chamikara *et al.*, "An efficient and scalable privacy preserving algorithm for big data and data streams," *Computers & Security*, p. 101 570, 2019.

[141] J. Michael *et al.*, "User-centered and privacy-driven process mining system design for iot," in *International Conference on Advanced Information Systems Engineering (CAiSE)*, Springer, 2019, pp. 194–206.

[142] S. Gao, J. Ma, C. Sun, and X. Li, "Balancing trajectory privacy and data utility using a personalized anonymization model," *Journal of Network and Computer Applications*, vol. 38, pp. 125–134, 2014.

[143] A. Pingley, W. Yu, N. Zhang, X. Fu, and W. Zhao, "Cap: A context-aware privacy protection system for location-based services," in *2009 29th IEEE International Conference on Distributed Computing Systems*, IEEE, 2009, pp. 49–57.

[144] T. Sylla *et al.*, "Towards a context-aware security and privacy as a service in the internet of things," in *IFIP International Conference on Information Security Theory and Practice*, Springer, 2019, pp. 240–252.

[145] E. de Matos *et al.*, "Providing context-aware security for iot environments through context sharing feature," in *TrustCom/BigDataSE*, IEEE, 2018, pp. 1711–1715.

[146] G. Qiu *et al.*, "Mobile semantic-aware trajectory for personalized location privacy preservation," *IEEE Internet of Things Journal*, 2020.

[147] J. Xiong *et al.*, "A personalized privacy protection framework for mobile crowdsensing in iiot," *IEEE Transactions on Industrial Informatics*, pp. 4231–4241, 2019.

[148] S. Han *et al.*, "Privacy-preserving gradient-descent methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, pp. 884–899, 2010.

[149] L. Wang, Z. Yu, B. Guo, T. Ku, and F. Yi, "Moving destination prediction using sparse dataset: A mobility gradient descent approach," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 11, no. 3, pp. 1–33, 2017.

[150] H. Shin, S. Kim, J. Shin, and X. Xiao, "Privacy enhanced matrix factorization for recommendation with local differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1770–1782, 2018.

[151] X. Meng *et al.*, "Towards privacy preserving social recommendation under personalized privacy settings," *World Wide Web*, vol. 22, no. 6, pp. 2853–2881, 2019.

[152] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," *Advances in neural information processing systems*, vol. 20, pp. 161–168, 2007.

[153] M. Gheisari *et al.*, "A context-aware privacy-preserving method for iot-based smart city using software defined networking," *Computers & Security*, p. 101 470, 2019.