POLITECNICO DI TORINO

Master Degree course in Computer Science Software Module

Master Degree Thesis

Automated Software Analysis for Privacy Policy Compliance

Supervisors
prof. Antonio Lioy
prof. Ugo Buy, University of Illinois Chicago

Candidate
Sofia Lucca

Academic Year 2022-2023

† Alla mia nonna Paola

Summary

Privacy of data collected and processed online has become more relevant in the last 20 years. Due to the rise of the Internet, online marketing, and advertisement, consumer data has become a very valuable resource. In recent years laws have been introduced in multiple jurisdictions to regulate the processing of these data. The businesses and websites which use this information must comply with these laws. Currently, the processes for checking and analysing software compliance with policy requirements are still highly human-dependent.

In this thesis, we analyse previous work in automatic software compliance with privacy requirements. We selected PrivGuard, an existing framework, and we conducted extensive experiments with that framework to assess its ability to automate software compliance verification [1].

The goals of this thesis are twofold. First, we select today's two most popular privacy regulations, namely CPRA and GDPR [2, 3] and we write formal specifications of those regulations in PrivGuard. Next, we identify three software benchmarks that elicit, store and manipulate personal consumer information, and we analyse those with PrivGuard's analysis tool, called PrivAnalyzer.

Finally, we assess the effectiveness of PrivGuard and PrivAnalyzer concerning the specification and verification of privacy policies. We also identify ways in which these tools can be improved to increase their effectiveness.

Acknowledgements

Penso che questo giro mi tocchi farla questa parte. Con la triennale ho evitato per non prendermi troppe responsabilità e per potere togliermi un doppio lavoro. Visto che però non posso più posticipare e che questa è l'ultima occasione - e per fortuna direi - a seguire farò del mio meglio per i ringraziamenti.

Ringrazio il mio relatore del Politecnico di Torino, *professore Antonio Lioy*, e il mio relatore dalla University of Illinois Chicago, *professor Ugo Buy*, che mi hanno dato la possibilità per questa ricerca e mi hanno affiancata nella stesura della seguente tesi.

Ringrazio la mia *famiglia* che mi ha supportata in tutti i miei 19 anni di scuola. I *miei genitori* che hanno permesso di vivere numerose esperienze, dall'anno all'estero in Texas, le vacanze studio in Inghilterra, l'università a Torino e il semestre a Chicago. *Matteo e Davide*, i miei fratellini - che forse non possono più essere chiamati tali - che nonostante le urla e le litigate mi hanno sempre sopportata e con cui sono cresciuta. I *miei nonni*, che non hanno sempre capito cosa studiassi ma che mi hanno comunque incoraggiata. Ringrazio specialmente la **mia nonna Paola**, che sfortunatamente non è riuscita a vedermi raggiungere questo traguardo, nonostante ci sia arrivata tanto vicina.

Ringrazio con tutto il cuore *Lore*: la persona che mi è stata accanto, anche a 700 km di distanza, negli ultimi due anni di università, ascoltando ogni mio sclero, pianto, crisi e momento di stress. Grazie anche per i momenti di svago, risate, nonostante so di potere essere difficile ogni tanto. Grazie per avere sempre creduto in me, soprattutto quando neanche io ci credevo, e per vedere le mie capacità. Non penso sarei riuscita a superare molti momenti senza il tuo amore, le tue parole di supporto e i tuoi abbracci.

Grazie a *Anna*, la persona che mi è accanto da più tempo, dopo la mia famiglia. La mia vera anima gemella, che mi ha fatto capire quanto possa essere forte un amicizia nonostante la vita possa avere in serbo cose diverse per ognuno. Grazie per esserci e per eventualmente mollare tutto quando ho bisogno di te.

Grazie a *Margie* la mia compagna di Poli dal giorno uno, ma soprattutto la mia compagna di serata a Torino. Grazie per i ricordi e le esperienze vissute insieme, grazie per i numerosi spritz alle Panche, per le pizze, per gli allenamenti in McFit, per i pranzi insieme nei cortili del Poli, per condividere i momenti di ansia e stress. Grazie soprattutto per avermi fatto vivere una fantastica esperienza universitaria e per essere diventata un'amicizia così importante nella mia vita.

Grazie a tutte le persone che hanno incrociato la mia strada in questi ultimi cinque anni a Torino. Grazie a *Zoltan* che so che mi vuole bene, nonostante io mi crei ansie e "pare" inutili. Grazie a *Marta*, con cui ho condiviso davvero troppo poco tempo insieme, ma che è diventata un'amicizia davvero vicina al mio cuore. Grazie a *Chiara*, che tecnicamente è un amicizia precedente a Torino, ma che ha influenzato la mia esperienza torinese, con le sue grida, so che un po' ti manco come coinquilina.

Grazie a *Bea*, che nonostante i periodi di silenzio è sempre presente e mi dimostra quanto valga la nostra amicizia. Grazie per esserci nonostante la distanza.

Grazie ad *Aly*, la mia ex-compagna di viaggi e vacanze. Grazie per essere stata la mia compagna di vita in tantissime occasioni da quando ci siamo incontrate.

Grazie a *Marta*: l'esempio fisico di amicizia a distanza. La persona che forse ha un rapporto tra tempo passato insieme e tempo di conoscenza e più basso di tutte le mie amicizie. Grazie

per tutte le nostre esperienze passate in Texas e per farmi credere nelle amicizie che non muoiono nonostante tempi e distanza lunghi. Sarai sempre la mia Food Partner.

Grazie allo *Squero* e a tutte le persone che ho conosciuto lavorando lì negli ultimi cinque anni. Non penso che sarei la persona che sono oggi se non anche grazie a questa esperienza. Grazie per avermi insegnato a vivere nel mondo reale e a organizzare la mia vita.

Grazie a tutti coloro che ho conosciuto a Chicago, che hanno reso bellissimi i miei 4 mesi là. Grazie per le feste in palazzina e per le deep dish, che devo ammettere un po' mi mancano.

Spero di essere stata brava e avere ringraziato tutte le persone necessarie, anche se sono sicura che potrei avere scordato qualcuno.

SL

Contents

1 Introduction 9

1.1 Motivation . . . 10

1.2 Thesis structure . . . 10

2 Previous works 12

2.1 Background Information . . . 12

2.1.1 Legalease and Grok . . . 12

2.1.2 Hapi: Hierarchical Access Policy Implementation . . . 14

2.1.3 PrivGuard and PrivAnalyzer . . . 14

2.2 Related Work . . . 15

2.2.1 A-PPL . . . 16

2.2.2 Jeeves . . . 16

2.2.3 P2U: Purpose-to-Use . . . 17

2.2.4 PrivPolicy . . . 17

2.2.5 PrivFramework . . . 18

3 Personal work 19

3.1 Laws translation . . . 19

3.1.1 Overview of the laws . . . 19

3.1.2 Translation . . . 21

3.2 Software systems . . . 31

3.2.1 LibreTaxi . . . 31

3.2.2 Selfmailbot . . . 34

3.2.3 Traccar . . . 37

4 Software manuals 42

4.1 Developer manual . . . 42

4.1.1 CPRA translation . . . 42

4.1.2 GDPR translation . . . 46

4.2 User manual . . . 52

4.2.1 The flags . . . 54

4.2.2 *example_id* values . . . 56

4.2.3 Input files . . . 57

5 Results 59

5.1 LibreTaxi results 59

5.2 Selfmailbot results 65

5.3 Traccar results 68

6 Conclusion 72

Bibliography 74

Chapter 1

Introduction

In the last 20 years, the amount of personal data retrieved from users by websites has increased drastically. This kind of data is quite valuable. Personal data is a great means for companies and businesses to understand their consumers, their interests, and their needs, thereby enabling, for instance, targeted advertisements to consumers.

Targeted advertising is challenging because consumer interests shift quite rapidly. In addition, vendors regularly introduce new products and services, requiring constant updates to their ads.

A typical goal for a business is to reach potential customers to increase sales and earnings. Direct marketing is an effective way to accomplish this goal, based on customer data. With the increase in popularity of the Internet in the 1990s, online advertising started to rise. Companies were able to track the number of clicks on an ad, the amount of purchasing done online, and the number of times a coupon was downloaded. Cookies allowed websites to gather and store all this information on consumer's computers, in a way that could be used as a reference for future marketing initiatives. User data became a golden and precious source of revenue.

In parallel with direct advertising, "free" service providers started rising. Organisations, such as Yahoo and Google, started offering services that seemed free at first sight, as they did not require any payment. No advertisement was present and no upgraded or premium version was available for sale. Evidently, nothing is for free. In those cases, the return to the organisation was in the form of data harvested from their users. Consumer data was systematically collected and sold to marketers and advertisers.

As the market around user data was becoming quite profitable, no applicable regulation was in place, thereby adversely affecting the privacy of consumers. There is a fine line between the proper use of consumer data and the harmful spreading of private information.

An example is what happened in 2012 when Target learned about a teen girl's pregnancy before her father did. A dad from Massachusetts found in the mail a Target mailer full of coupons for maternity clothing, nursery furniture and pictures of infants. After a big complaint to the manager at the local Target, he talked to his daughter and learned about what Target analysis knew already. In fact, the company managed a data-mining process to target its pregnant consumers, by analysing the history of purchases and demographic information Target has collected or bought from other sources. Based on these kinds of data, they were able to understand the buying patterns of someone who was expecting. At this point, the marketing and sending of any form of advertisement was done based on this data.

For the past several years, laws are being introduced to guarantee a more distinct definition of this line. In 2016, the European Union (EU) introduced the *General Data Protection Regulation* (GDPR) [3], which defines limitations on the use of data from users by all kinds of websites. This law took effect in 2018, engendering a big ruckus around it. The GDPR was an evolution of the "Privacy Act" from 1995, which was a significant milestone regarding privacy data and regulation within the EU. A change was needed due to the evolution of the Internet and related technologies, which caused the Privacy Act of 1995 to become outdated. The main points of the GDPR are

associated with the possibility for users to select and define which kind of data a website is allowed to use, while still guaranteeing to receive equal benefits from the website's business.

The GDPR stipulates that a business must clearly spell out what consumer data will be used, the purpose of such use, and the length of time for such use. Another important provision allows users to eventually delete their data, thereby accommodating users who change their minds about the uses of their data after the initial agreement.

The main limitation to consumer choice is the inability to remove or delete data that a website needs to conduct business with a user. We will see in the following chapters any exceptions for the general cases. With this law, the main goal is to regulate a new form of product, that is, consumer personal data, which has become a big source of revenue in this new technology era.

In contrast with the EU, the United States does not have a federal law similar to the GDPR. Consequently, no such regulation can be applied to all states. However, in recent years, some states have developed privacy laws to guarantee privacy protection within their borders. An example that we analysed in the following sections is California, which introduced in 2018 the *California Consumer Privacy Act* (CCPA) as an initial definition of rules and limitations of data usage for users' privacy online. As of today, specifically since January 2023, a reinforced version of this act has been introduced: the *California Privacy Rights Act* (CPRA) [2]. CPRA is similar to GDPR, in terms of rights for consumers and regulations to guarantee the right of privacy, although significant differences between the two acts exist. CPRA was put to a referendum in California during the 2020 election cycle and approved by a majority of voters. It took effect on January 1, 2023.

1.1 Motivation

The basic concept underlying this thesis is summarised in the paper:"Data Capsule: A New Paradigm for Automatic Compliance with Data Privacy Regulations" [4]–explained in Section 2.1.3. The idea is to find an automated (or partially automated) solution for verifying compliance with privacy policies by websites and businesses that process consumers' data. Our goal is to evaluate existing techniques and tools for specifying privacy policies, for the automatic analysis of the resulting specifications.

We chose compliance with the *California Privacy Rights Act* (CPRA) and the *General Data Protection Regulation* (GDPR) as the focus of our investigation. CPRA is a recent and comprehensive legislation. GDPR, on the other hand, is a similar law that was already analysed [4] but can represent a reference point for comparison concerning the results.

Previous work resulted in the creation of frameworks that automate, in part, the process of compliance verification, by comparing the consumer software against the regulations. An example is PrivAnalyzer [5], which we selected for our analyses.

1.2 Thesis structure

Chapter 2 surveys previous work in privacy compliance. First, we will analyse existing technologies that allow analysis of programs with respect to given requirements. Next, we will describe the possible languages available to write the privacy laws into formal languages suitable for analysis.

Chapter 3 is the first part of our work. We summarise CPRA and GDPR, and we explain how key portions of these laws are captured in the selected specification language, namely (Legalease). We then introduce the three main benchmark systems selected for the analysis. Furthermore, we also explain the work that we did to prepare the source code of those systems for analysis.

Chapter 4 is separated into two sections. The first one defines the developer manual for *PrivAnalyzer* for our work, it lists all the selected CPRA and GDPR articles and their translation into Legalease. The following section is the user manual, that explains how to install the system and all the steps to take to use it and obtain the same results.

Chapter 5 analyses the results of each benchmark systems with respect to both GDPR and CPRA. We compare the results returned by the automated tools with the conclusion that we reached by manually examining the source code of the benchmark systems.

Chapter 6 explains in synthesis the work done and the results obtained. We reflect on eventual future applications and works that could be done in this field.

Chapter 2

Previous works

2.1 Background Information

2.1.1 Legalease and Grok

Sen et al. [6] define a combination of the LEGALEASE language with the GROK mapper as a solution for verifying compliance of website systems with privacy regulations. LEGALEASE is a domain-specific language for specifying privacy policies. Care was taken to ensure that LEGALEASE could be used by policy authors and privacy champions lacking expertise in formal languages. As a result, LEGALEASE allows a user to specify restrictions imposed by privacy laws in a natural way, for instance, by supporting a specification structure that mirrors the natural language text in each law.

GROK maps any possible element from different code sources written in a programming language into a datatype usable for LEGALEASE. GROK makes it possible to verify that a software system complies with the requirements imposed by a given privacy policy in real time as the system is being written. Data analysis is mainly performed in such systems written in the existing programming languages Hive [7], Dremel [8] or Scope [9]. These languages are based on the use of tables for data-structure representation. Operations on the data are modelled with an SQL-like language; operation results are also represented as tables with columns and values from input tables. The paper itself focuses on Scope as the data analysis language ([9]). LEGALEASE specifications consist of nested ALLOW and DENY clauses where each clause is an exception to its directly enclosing clause (Figure 2.1)[1].

```
DENY DataType Location
    UseForPurpose Advertising
EXCEPT
    ALLOW DataType Location: Encrypted
    UseForPurpose AbuseDetect
EXCEPT
    DENY DataType Location
    AccessByRole Intern
```

Figure 2.1. Example Legalease privacy clause.

[1]This Legalease clause means: "location cannot be used for advertising. Location can be used if encrypted to detect eventual abuse. Location cannot be accessed by any intern"

Each clause is defined with attributes whose values are defined by a concept lattice. The lattice is used as a bridge between the policy defined on LEGALEASE and the data retrieved and converted by GROK. LEGALEASE and the language tools are not affected by any additions as long as the additions can be represented by adding a new lattice. When talking about a concept lattice we refer to a hierarchical graph. The hierarchy of the nodes in this graph will allow to understand how the clauses in the policies affect the data and their access. LEGALEASE's lattices represent four main attributes:

InStore defines collection and source for collection of data. Meaning that the storage of data can be limited depending on how it was collected.

UseForPurpose limits the possible uses of the data.

AccessByRole defines who can or cannot access the data.

DataType is responsible for the limiting of accessing data-types. For this attribute the history of the policy datatype must be taken in consideration . To do so limited type-state are used in combination with policy data-types, both of which are defined in a lattice. In the end, the datatype is defined as Datatype datatype:type-state with a lattice that depends on the sensitivity level of privacy champions.

The previous attributes are the root of the lattices and will be the starting point for building complete lattices from the use of GROK.

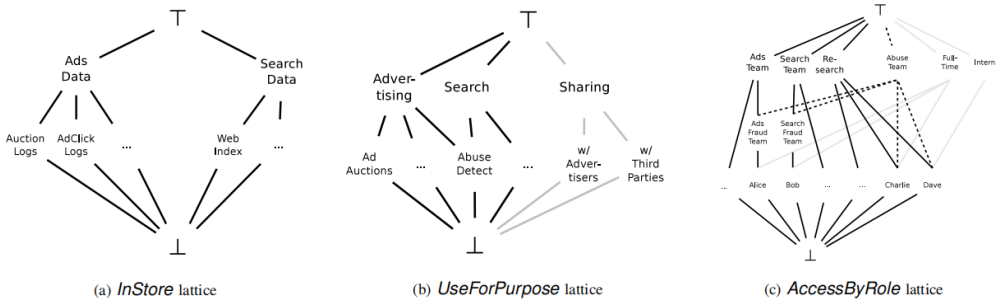

Figure 2.2. Complete lattice from paper

GROK maps the data obtained from different source codes into a datatype language usable for LEGALEASE as elements into the lattice of the various attributes. With a constant and continuous check of privacy compliance and update of code, it is possible to write and change privacy policies while defining the code to retrieve and obtain data.

This option is valuable because code and policy can be updated in parallel with each other. The approach used is bootstrap, causing to limit the privacy of the data of the users. In fact, we obtain the complete lattice and correct hierarchy for the data itself. An initial log, which contains all the jobs, files and users, is analysed to extract the coarse-grained data flow graph. The possible labels for *InStore* and *AccessByRole* are defined and associated with a high confidence score, since they are explicitly tracked from the code. From a given role the corresponding purpose is then obtained, so the labels for *UseForPurpose* are generated with low confidence, since they are obtained heuristically. A second heuristic approach is used to define the labels for the *DataType* attribute. Starting from the source code identifiers, a limited set of data-types are inferred with the use of a set of regular expressions, the all process generates labels with low confidence. Semantic analysis of the programs allows to reduce the nodes, while a static data flow analysis expands the coverage for the *DataType* attributes. Finally, a small set of nodes is identified to define the final hierarchical lattice. The data is never accessed, granting privacy to the user. To obtain all the labels GROK uses the input code with a level of confidence appropriate to the approach. It is possible to check the final lattice manually to eventually increase the confidence score of the labels and allow more precise results.

2.1.2 Hapi: Hierarchical Access Policy Implementation

Unfortunately as stated in [10] the use of Legalease is very limited, also due to the complexity of generating a lattice to move through for defining any kind of policy clause. Hapi (*Hierarchical Access Policy Implementation*) is a Legalease implementation obtained by making some assumptions with respect to the original language. First, it is based on the removal of lattices and the introduction of a simpler abstraction, that is a partially ordered set or poset. A user can access and modify these sets and eventually define a different hierarchy among the elements. Attributes defined in the previous paper are now the topmost elements of the ordered set. To allow an easier understanding and interpretation of the policy clause, instead of relying solely on the indentation, with Hapi each eventual exception and definition is done with the use of brackets and a syntax oriented to the use of explicit blocks. Lastly, to allow and simplify its use, an implementation with Kotlin has been defined supporting the future development of a graphical user interface. Hapi programs will give as result a set of tuples based on the input data used to define the posets, but to obtain this result it is necessary to normalise the program. Normalisation means satisfying two main rules:

- Every ALLOW and DENY clause refers to a single element
- Every program must start with an ALLOW or DENY clause associated to the topmost element in the product poset. When talking about the product poset we refer to the cartesian product of two posets where we consider the combination of all its elements if one proceeds the other.

When talking about poset we define a partially ordered set with a topmost element which is analogous to the attribute defined in the lattice from Legalease and Grok. The definition of them is done, not anymore through the data but through a previous definition.

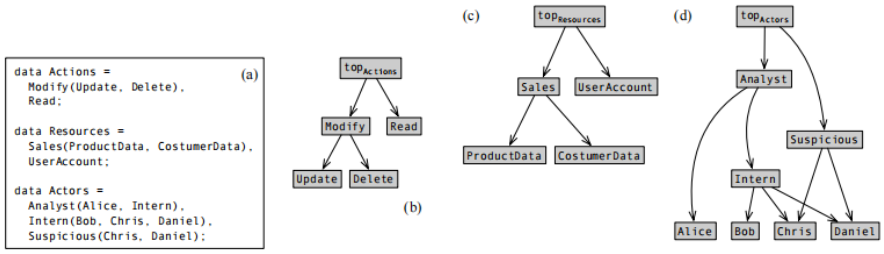

Figure 2.3. Poset declaration and representation.

Hapi is a significant simplification of Legalease, making it an attractive option for analysis. The result obtained by the parser can be analysed and used by a policy checker, making it a good intermediate representation for subsequent checking of correct privacy definitions. The existence of a GUI and e-book boosts the ease of use of this implementation. The main disadvantage is that Hapi requires a normalisation of the policy, causing an increase in the length of even simple privacy policies. The translation is automated and implemented with a function that converts the original specification into a normal form through pattern matching. This requires an input of data done manually. This is a disadvtange because of the high amount of data to be analysed and checked.

2.1.3 PrivGuard and PrivAnalyzer

Wang et al. [1] define a new framework for data analysis: Privguard. Since Legalease is an easily readable policy language, it is the one chosen for this specific research, but PrivGuard allows its application to different machine-readable policy languages. The framework is responsible for checking the correctness of policy for a specific data analysis. First, legal and data protection experts start by defining a base policy: the policy that represents at best the translation of the

privacy regulations defined by the law. Secondly, users, from the base policy, specify their privacy preferences, through an API, starting from the base policy. Lastly, data analysts will define a guarded policy and the program that collects and uses consumer data. Collection and use of data must be done while taking into consideration a guard policy stronger than the base one. This checking is done with the use of a static analyser: PrivAnalyzer, which will check if the base policy is not stricter than the guard policy and if the program meets the privacy spec or requires further modifications. This will happen if PrivAnalyzer's output contains residual policies which model unsatisfied privacy requirements. PrivAnalyzer itself does not access the data because it only analyses the program and policy to access it, so it guarantees privacy of the user information. From a semantic point of view only ALLOW is used in this language and each attribute is connected with AND/OR. The language used to define the policies is an implementation of Legalease. The main attributes defined are as follows:

ROLE who can access and examine the data;

SCHEMA which defines the columns of the data that can be accessed;

PRIVACY which defines how data can be used, for possible privacy-enhancing technologies:

- Deidentification/pseudonomynazation;
- Aggregation;
- k-Anonymity;
- l-diversity;
- t-closeness;
- differential privacy;

FILTER exclude certain items with respect to some conditions;

REDACT to remove partially or completely data from a column;

PURPOSE for the use of the data analysed.

The framework, PrivGuard, works under the protection of cryptographic tools, to protect the data stored, and a trusted execution environment (TEE). In the TEE, the analyser checks for compliance of the guard policy and users' privacy preferences, while the execution engine will access data and execute the program. In the case of residual policies from PrivAnalyzer the results of the program are encrypted and kept in a storage layer until a further analysis will give no residual policies, meaning that the program satisfies the requirements.

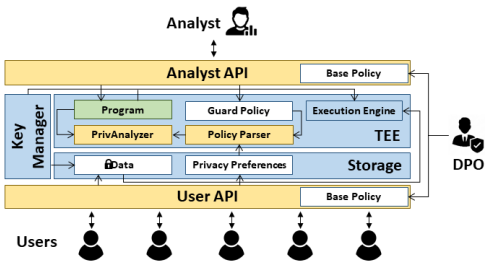

Figure 2.4. Data flow in PrivGuard

PrivGuard is a useful option for our project because it can verify the guard policies that we defined with respect to target programs. Our guard policies are based on two privacy acts, namely CPRA and GDPR. The residual policies returned by the PrivGuard are a starting point to understand whether programs under consideration are complying with privacy legislation. Wang et al. [1] already implemented the code for *PrivAnalyzer* and tested it to guarantee scalability and usability over Python programs.

2.2 Related Work

As stated by each one of the previous papers Legalease is not the only possible language used for privacy definition. Kasem-Madani and Meier [11] define and classify 27 main domain-specific languages, including general ones based on XML. Each language is classified based on:

- Type
- Intention of use
- Scope
- Design and implementation details

2.2.1 A-PPL

Azraoui et al. [12] define A-PPL, shorthand for Accountable PPL. It is a research and implementation of an already existing policy language (PPL) but, as its name states, its main goal is guaranteeing accountability for privacy policy definition. A-PPL is an extension of XAMCL with a new obligation and authorisation syntax. The main difference in A-PPL is the introduction of new features to guarantee all the goals for accountability:

- Reporting and notification
- Data Location Rules
- Ability to audit
- Logging
- Privacy policy
- Access control rules
- Usage Control Rules
- Data retention period

The researchers applied A-PPL in a healthcare cloud system.

A-PPL and this research are easily a starting point for understanding how to implement and define policy constraints. Since the rest of the article is an applied example of the language, facilitating the understanding of the language. The main problem associated with A-PPL is that it is an XAMCL implementation, difficult to understand with respect to Legalease or Hapi.

2.2.2 Jeeves

Yang et al. [13] define and analyse Jeeves. Differently from PrivGuard, Jeeves is not responsible for checking if a website satisfies the specified constraints. This language defines the privacy constraint that the code must follow to use the users' data. This means that a programmer will write a program that will use the data for a specific output. The results obtained from the program will be checked with respect to the privacy constraints defined. In case of positive response from this comparison the output will be displayed. In other words, Jeeves allows to separate the core program functionality from privacy concerns.

This is a language example which can be used after being sure that the policies defined are following the new laws associated user privacy. In fact, it is based on the assumption that the policies submitted for filtering the code are legal and correct. So we believe this can be a possible option for a later application of the privacy we will eventually define. The main problem is that the language, in this case, is not as easily understandable as in some previous cases.

2.2.3 P2U: Purpose-to-Use

Iyilade et al. [14] implement and expand the already existing privacy language Platform for Privacy Preferences (P3P) with P2U. As stated in the paper, P3P is not considered a good option/solution regarding data management and sharing data with third parties. It is a static approach developed when the retrieval and use of user data started around 2003. P2U has the following goals:

change the role of the user which is not anymore a passive data subject but an active subject which creates and develops data to be used by the organisation;

manage purpose change of data since initial collection. In fact, as time goes by data collected can be used in new and different ways that initially weren't considered or available, with respect to the time of collection. The user must be notified and eventually, he must decide how this change in possibilities can be managed.

support negotiation of data User can choose and select the privacy policy for the managing of his data, this means that he doesn't necessarily have to accept the option given by the organisation but he can change depending on the user's preferences. P3P was based on 'all-or-nothing' concept, such that user could either accept or reject the terms of privacy, without allowing possibility to select only the one which he preferred.

This language is in an XML format, so a language not very similar to the human one but requires a knowledge of the domain.

This option is valuable for the definition on managing of sharing data, the main problem is associated with the fact that is mainly limited to the usage of data and sharing with respect to third parties. This is limiting since privacy laws cover a wider set of topics.

2.2.4 PrivPolicy

Wang et al. [4] define an approach to manage privacy rules and compliance for websites. It is associated with the renewal and introduction of new privacy laws such as GDPR. The idea is to manage data as data capsules, such that they can be singularly retrieved and managed when needed. This idea is associated with the possibility of allowing consumers to ask to remove or delete their information. The previous approach would cause to have data possibly saved in multiple copies and be difficult to retrieve. The data capsule manager is the one from PrivGuard, so it will receive a code for analysing the data and policies which will protect the data and the result will be either a set of residual policies that the program doesn't satisfy or an approval of the program that allows the use of the data. To define privacy laws a language similar to the one in PrivGuard has been introduced: PrivPolicy. Each privacy policy is associated with an ALLOW statement associated with a possible set of attributes:

- SCHEMA, to specify the column of the data;
- FILTER, to filter based on the column and a specific integer value;
- ROLE, to specify who can access the data;
- NOTIFICATION_REQUIRED;
- CONSENT_REQUIRED;
- PURPOSE;
- DECLASS, for the type of modification of the data.

The idea for this research was to guarantee principles of data privacy:

- Transparency and auditing;

- Consent;
- Processing Control;
- Data portability;
- Guarantee against re-identification.

This approach has been applied to different current laws: *General Data Protection Regulation* (GDPR), *California Consumer Privacy Act* (CCPA), *Health Insurance Portability and Accountability Act* (HIPAA) and *Family Educational Rights and Privacy Act* (FERPA). All these laws and acts are the main regulations for privacy available at the time of the publication of this research.

This option can be a very valuable and useful starting point to understand how to write privacy limitations about CPRA. This solution still allows checking a program and seeing its compliance with what we will personally write as a privacy policy.

2.2.5 PrivFramework

Khan et al. [15] carry out a research, similar to the previously cited, about PrivGuard and PrivPolicy. With this research, they theoretically introduce PrivFramework, which is a framework with a client-side API that allows for the collection and analysis of privacy policy compliance. The main features of this framework are:

Data owner uploads the data with a formal policy on the use of this data through an application;

Analyst submits an analysis program for the access of the data;

Data Manager receives the program and controls if it satisfies the constraints. After that, he will send back to the analysts either the results or the privacy requirements not yet satisfied.

As stated before, data is collected and analysed through the use of PrivGuard, so in a data capsule format, while the policies are encoded in PrivPolicy.

The positive aspect of this paper is the introduction and definition of an application, with also a client-side aspect. The main problem is the difficulty of accessing it or finding it online.

Chapter 3

Personal work

3.1 Laws translation

This chapter will explain the work we did, starting from the research works previously described. We translated into LEGALEASE two main privacy laws (CPRA and GDPR) with some differences in the attribute values due to differences in the laws and the terms used in them. It is important to state that for the translation we had to discard some clauses. The reason is CPRA and GDPR contain both technical specifications in the use of consumers' data as well as legal information that cannot be analysed by PRIVANALYZER. We discarded the clauses which defined how to legally manage the violations of technical clauses. Another big group of clauses that we could not translate were the ones that defined how the customer must be informed about the use of his data and, also, how to manage communication between the user and website about the management of the data for possible changes. In these cases, PRIVANALYZER cannot be used to check compliance with the laws.

3.1.1 Overview of the laws

The *California Privacy Rights Act* (CPRA) is a California ballot proposition approved during the 2020 November US general election. It is an extension of the *California Consumer Privacy Act* (CCPA) from 2018. CPRA took effect in January 2023; it manages privacy regarding consumer-business relationships. It is composed of 43 articles; we translated 7 of them. From the 7 selected ones, we were able to obtain 11 policy clauses written in LEGALEASE. The selected articles [2] are:

"1798.100. General Duties of Businesses that Collect Personal Information" is composed of some sub-articles, that we divided to manage different purposes. The first part defines what a business can do with the data obtained from the consumer and with his consent. Secondly, the article introduces how those data can be managed by third parties in case a business sell or shares them with the consumers' consent. Lastly, it defines how the business must protect the data when consumers allow the retention of it.

"1798.105. Consumers' Right to Delete Personal Information" defines the right of a consumer to request the deletion of his data from the storage of anyone who had the power to access it.

"1798.106. Consumers' Right to Correct Inaccurate Personal Information" introduces the right for a consumer to request to business a modification of the data retained which is inaccurate.

"1798.110. Consumers' Right to Know What Personal Information is Being Collected. Right to Access Personal Information" allows a consumer to retrieve the information that a business has collected about him (e.g.*Name: John, LastName: Doe,*

Email: john.doe@fake.com). The article has a separate sub-article which defines the right to know just the type of data collected and not the data itself (e.g. *Name, LastName, Email*). We assumed that the retrieval of the data can cover also this requirement, instead of having a separate policy for only the type of data collected.

"1798.120. Consumers' Right to Opt Out of Sale or Sharing of Personal Information" requires that the consumer can change his mind concerning selling or sharing the data after initial consent. We assumed for this article, that this policy could be interpreted as the possibility for a user to revoke consent. This article has a second section which defines the legal minimum age for consent, with respect to sharing or selling of the data.

"1798.121. Consumers' Right to Limit Use and Disclosure of Sensitive Personal Information" allows the consumer to limit the usage of sensitive personal information, which has a higher level of importance, thereby requiring a higher level of protection. Whenever a consumer limit the use a business and eventual third parties are not allowed to use those data.

"1798.145. Exemptions" introduces some special cases when the previous articles are not applied. We selected 2 main sub-articles. First of all personal data can be accessed by a government agency in case of risk and danger of a natural person. We assumed that the access is done when there is a dangerous situation since PrivAnalyzer can not check it. Second data requires a certain level of privacy, that can be met with aggregation or deidentified, to protect the single consumer

The research work done to write the Legalease policies for CPRA covers any main technical aspect of CPRA. Even though the number of final translations is limited with respect to the original set of articles from CPRA. In fact, as mentioned before many were not possible to be translated for PrivGuard.

The *General Data Protection Regulation* (GDPR) regulates privacy online for the countries belonging to the European Union. It was initially published in 2016 but became officially active in 2018. Differently from CPRA, it has been around for longer so it has been analysed and extensive research work has been done on it. It also does not only manage consumer-business relationships, but any kind of relationship regarding privacy over data and information online. It is composed of 99 articles, but we translated only 19 of them. For CPRA the number is higher because for GDPR many articles are specific for one purpose, while for CPRA one generic article contains many specific sub-articles.

"Art. 5 Principles relating to processing of personal data" has three main sub-articles we translated. It defines the cases for which personal data can be accessed: business purpose, scientific research, public interest or historical research. The second part introduces the possibility to change or delete the data in case of lack of accuracy. Lastly, the data must be protected with an appropriate security level. For this last part, we assumed anonymisation as a privacy requirement, since it is a form of privacy introduced with GDPR. Data is partially modified, in a way that the single consumer cannot be identified unless other information is given.

"Art. 6 Lawfulness of processing" is responsible for the limitation over the processing of data. It lists the figures which have the right to process the data, the reasons for which the process can be done and the age limitation of the consumer to which the data belongs (above 16).

"Art. 7 Conditions for consent" is related to Article 6, since it also defines the need for consent from the user in case of processing. "Processing is any operation or set of operations performed on personal data".

"Art. 8 Conditions applicable to child's consent in relation to information society services" expands Article 6 by defining the case of a consumer between the ages of 13 and 16, who requires consent from a guardian.

"Art. 9 Processing of special categories of personal data" introduces personal data that allow one to identify uniquely an individual. They must be used differently and with eventual specific consent by the consumer, so they needed an individual policy clause.

"Art. 10 Processing of personal data relating to criminal convictions and offences" states that criminal data are accessible by law enforcement.

"Art. 14 Information to be provided where personal data have not been obtained from the data subject" states that a consumer has the right to know where the data came from and who manages it, in case data had not been obtained from the consumer himself.

"Art. 15 Right of access by the data subject" allows the consumer to access a copy of the data that has been processed.

"Art. 16 Right to rectification" states that a consumer has the right to modify data stored which is incorrect.

"Art. 17 Right to erasure ("right to be forgotten")" defines the consumer's right to have the data deleted, in case either the consent has been removed or the data is no longer needed for the initial purpose of collection.

"Art. 20 Right to data portability" is similar to Article 15, but it allows a consumer to access all his stored personal data.

"Art. 21 Right to object" allows the consumer to object to the processing of personal data for marketing purposes.

"Art. 25 Data protection by design and by default" states the need for usage, storage, selling and sharing of data with the processing of data that guarantees protection for the user. This must be applied by anyone that uses the data.

"Art. 39 Tasks of the data protection officer" introduces the data protection officer, who is responsible for managing and checking compliance with GDPR.

"Art. 44 General principle for transfers" and **Art. 46 Transfers subject to appropriate safeguards** define the need for protection of personal data that needs to be transferred to a country outside the European Union or an international organisation.

"Art. 49 Derogations for specific situations" is applied in case of exceptions from Article 44 or Article 46. It allows a transfer for only specific purposes and with eventual consent.

"Art. 58 Powers" define the supervisory authority: which is the figure chosen by the country to monitor the application of GDPR. He has the power to access personal data to perform his tasks.

GDPR allowed us to obtain a broader set of policy clauses. The bright side is that it allows to check for the capability of PrivAnalyzer. The negative side is that GDPR is very broad and can be applied to many situations, such that not all the clauses can be checked with the same type of code that will be used for the clauses obtained with CPRA. It is possible to affirm that the translation into Legalease is complete for the technical aspect of GDPR, even after filtering some articles.

3.1.2 Translation

We divided the translation into two parts for the two different laws. Since GDPR and CPRA have some differences with respect to definition and applications.

```
ALLOW ROLE Oncologist
    AND SCHEMA age, condition
    AND PRIVACY DP(1.0,1e-5)
    AND FILTER age > 18
    AND REDACT zip(2:)
    AND PURPOSE PublicInterest
```

Figure 3.1. Example of a policy written in LEGALEASE from [1].

PRIVANALYZER uses as language an implementation of LEGALEASE. Any privacy clause can be defined as a set of attributes, which can be (we will refer to Figure 3.1[1] for any attribute value):

- ROLE specifies the people able to access the specific data (*e.g. Oncologist*).
- SCHEMA defines which column can be accessed (*e.g. age, condition*).
- PRIVACY lays the condition on how the data may be used (*e.g. DP(1.0,1e-5)*). It is the only attribute that doesn't give the freedom for liberally chosen values. PRIVGUARD limits the choice to:
  - De-identification (or pseudonymization);
  - Aggregation
  - k-Anonymity
  - l-diversity
  - t-closeness
  - Differential privacy
- FILTER allows to specify that certain data items must be excluded (*e.g. age >18*).
- REDACT allows to require a partial or complete redaction of information in a column (*e.g. zip(2:)*).
- PURPOSE restricts the reason for which the data may be analysed (*e.g. PublicInterest*).

Each of these attributes can have one or more values, which we will define in the following section. All the naming that we chose for the attribute values were based on the definition given by the laws. Both GDPR and CPRA have a section dedicated to the definitions of the terms used in the articles. LEGALEASE allows total freedom on the attribute values choice, so, except for PRIVACY, all the values we will list are going to be unique for this application. We defined all the values.

ROLE allows us to define the figure responsible for accessing the data. We divided the translation based on the law. First of all for the ROLE values in CPRA:

Business is responsible for the collection of the data and managing it for its purpose;

ServiceProvider accesses, through a contract with the *Business*, data and process it on behalf of the business;

Contractor accesses personal information with a contract with the business. Different from *ServiceProvider*, because it does not work on data on behalf of the business. It is limited to the contract and the business has to check at least once every 12 months for the correctness of the behaviour.

[1]Oncologist is allowed to access data regarding age and condition, for all patients above 18 years of age, under the condition of differential privacy. The zip code for this data must be reduced to the first 2 digits and the all work must be done with the intention of public interest.

ThirdParty receives personal information through selling or sharing from the business. The consumer must give permission to manage this way his data. This value is hard to identify since CPRA defines it for what it is not and not for what it is. Meaning, a *ThirdParty* is not the *Business* with which the consumer directly interacts, *ServiceProvider* of the *Business* or a *Contractor*.

ConsumerID is a unique identifier for the user to access his data. It is necessary to check it such that the access regards only his data and not the personal information of others.

GovernmentAgency has a different level of access to the personal data of consumers.

For GDPR the terminology in the law was different and broader, due to the application of it for more than the business-consumer relationship. Therefore, also the values for ROLE were different:

Controller is similar to *Business* for CPRA. It defines the figure responsible for managing the data for a given purpose;

PublicAuthority is a governmental organisation that carries out tasks in the public interest;

LegalAuthority is the person allowed to access the data in case of legal obligation (*Art. 6* from GDPR);

NonProfit is an organization without any profitable income that the consumer was/is part of;

OfficialAuthority is the figure allowed to access criminal records (*Art. 10*;

DataSubjectID is the single user to access his data. It is not anymore considered a consumer;

Processor who is granted by *Controller* the access of the data solely for the processing of the data itself;

DataProtectionOfficer manages a large amount of data. They are responsible for guarantying security over data collected;

ThirdCountry and InternationalOrg. The law regulates eventual sharing of data with external countries through a specific contract;

SupervisoryAuthority is a public authority chosen by the country to be responsible for monitoring the application of GDPR.

SCHEMA is the attribute that defines what information can be retrieved from the specific policies. CPRA defines the data accessed as "consumer's personal information", such that the corresponding attribute value is PersonalInformation. GDPR uses the term "personal data", prompting us to introduce the attribute value *PersonalData* when referring to policies from GDPR. When writing the initial policies we used these terms to allow a general application. But when we used them on the system for the analysis we substituted them with the actual data that were used by the specific code. In fact, *PersonalData* and *PersonalInformation* are broad and generic definitions for more specific data. Both the laws have articles (*Art. 4.1* for GDPR and *1798.140.v* for CPRA) which define what kind of data can be classified as personal. We used these definitions to identify what information needed to be protected when analysing the source code in the second part of the research. The values that were shared by both laws are, as follows:

Name, LastName identify the consumer;

OnlineID is a unique identifier generated online to distinguish different data owners.

For CPRA *PersonalInformation* can also be:

PostalAddress, Address are the address and location information, depending on the state and area. In the case of California, the postal address is associated with the Zone Improvement Plan (ZIP) code while in Italy it is the Codice Avviamento Postale (CAP);

PersonalProperties defines everything that is owned by the consumer, associated with the knowledge of the business and what the consumer shared with it;

IPAddress, EmailAddress ;

ProductHistory, ServiceHistory, BrowsingHistory History of the consumer of what he purchased or where he has navigated through;

ProfessionalInformation is information regarding the consumer's past and history with respect to work and employment;

EducationInformation is information regarding education which is not publicly available. In fact, some of the information regarding a consumer's education can be accessed publicly by the business without requesting them from the consumer. Some information is analogous to the previously listed, so it requires the consumer's consent to access it;

InferenceInformation Starting from consumers' information and history it is possible to generate a profile that can be used to group the consumers or for marketing purposes;

BiometricInformation are the ones that uniquely identify a person, with respect to psychology, biology and behaviour;

SensitivePersonalInformation are more sensitive and require a higher level of protection.

- **Social security number** (SSN) unique identifier for the American citizen. It is a set of 9 digits, mainly used by American workers to uniquely identify themselves and to manage their work and paycheck;
- **DriverLicenceNumber** is the unique identifier for the number associated with the driver's licence, since it is often used as an ID in the United States;
- **PassportNumber** can be used instead of *SSN* or *DriverLicenseNumber*;
- **GeolocationData** For many websites the requirement to work or to execute a high part of the work is the use of the location of the consumer;
- **GeneticData** is data results from the biological sample of an individual and concerns genetic material, such as DNA, RNA, genes and chromosomes;
- **AccountCredentials** are the consumer's account credentials to access the personal profile;
- **DebitCardNumber, CreditCradNumber** In case of bank accounts or websites requiring payments by the user the information regarding debit or credit cards must be kept safe and the access must be limited;
- **MailContent, TextContent** is the mail exchange that does not have business as the final recipient;
- **HealthHistory, SexualOrientation** is the history regarding the health or sexual orientation of the consumer.

GDPR has some different values. This is due to the differences with the other laws and how other data are managed in Europe with respect to the US. They are:

IDNumber is a unique number for identifying the owner of the data;

LocationData is information regarding the current location of a consumer;

PhysicalInformation are data that define the physical aspect of the data subject;

PhysiologicalInformation is information regarding the physical health of the person;

GeneticInformation is similar to *GeneticData* for CPRA. It is the set of genetic characteristics of a natural person, that had been obtained from analyses of biological samples of the person, such as chromosomes, DNA, or RNA ;

MentalInformation are data regarding the mental health of the subject;

EconomicInformation is the history of the economic side of a person, current and past;

CulturalInformation is information regarding the cultural background of the data subject;

PDIdentifiable is information that allows one to uniquely identify a person. This kind of information can be:

- *RacialInformation* is the data for the racial origin of the person;
- *PoliticalInformation* is information regarding political preferences and interests of the subject;
- *ReligiousInformation* or *PhilosophicalInformation*, both associated with history in beliefs and ideas of the person;
- *TradeUnionMambership*. A trade union is an organisation with members belonging mainly to the workforce.

The POLICY attribute is used to define the level of protection needed for an action with the specified data. Differently from all the other attributes, the values for PRIVACY cannot be chosen with total freedom. PrivAnalyzer limits the choice to:

- *Anonymization*
- *Aggregation*
- *k-anonymity*
- *l-diversity*
- *t-closeness*
- *DP(float, float)*

Based on the definitions we selected three for the Legalease translation.

Anonymization is used for the policies both in GDPR and CPRA. It is a technique introduced by GDPR. It allows the data to be partially modified, in a way that the single user cannot be identified unless additional information is given. It is the only type of privacy used for GDPR policies. From GDPR *Art.25* "Taking into account the state of the art, the cost of implementation and the nature, scope, context and purposes of processing as well as the risks of varying likelihood and severity for rights and freedoms of natural persons posed by the processing, the controller shall, both at the time of the determination of the means for processing and at the time of the processing itself, implement appropriate technical and organisational measures, such as pseudonymisation, which are designed to implement data-protection principles, such as data minimisation, effectively and to integrate the necessary safeguards into the processing to meet the requirements of this Regulation and protect the rights of data subjects." (Figure 3.2)

```
ALLOW ROLE Controller
    AND SCHEMA PersonalData
    AND (PURPOSE Retention OR PURPOSE BusinessPurpose OR PURPOSE Sell OR
        PURPOSE Share OR PURPOSE Processing)
    AND PRIVACY Anonymization
```

Figure 3.2. GDPR: Art.25 translation in Legalease.

In Figure 3.2 the ROLE clause indicates a controller (i.e. an entity managing consumer data), SCHEMA refers to the data for which constraints are being specified; PURPOSE

indicates all the possible situations for which data can be used for this clause: retention (i.e. the collection and storage of data), business purpose (i.e to execute any task for the business), sell or share to any possible third party and processing (i.e any use or alteration of the original data); PRIVACY, as explained before, is how the data must be managed for the purpose.

Aggregation means that the business will combine ad group the consumers based on specific characteristics and use the resulting grouped data for the business purpose. It is required by CPRA in *1798.145.6*: "Collect, use, retain, sell, share, or disclose consumers' personal information that is deidentified or aggregate consumer information" (Figure 3.3).

```
ALLOW ROLE Business
       AND SCHEMA PersonalInformation
       AND (PURPOSE Retention OR PURPOSE BusinessPurpose OR PURPOSE Sell
           OR PURPOSE Share)
       AND (PRIVACY Anonymization OR PRIVACY Aggregation)
```

Figure 3.3. CPRA: art.1798.145.6 translation in LEGALEASE.

Figure 3.3 defines the CPRA version of Figure 3.2, which is from GDPR. It has a ROLE clause which indicates the business (i.e. responsible for the managing of the data), SCHEMA has a different value but still refers to the data for which we are specifying constraints; PURPOSE indicates similar reasoning of use except for processing and lastly, PRIVACY has two possible values, previously explained and listed.

Differential Privacy was assumed as a possible solution. CPRA *1798.100.d.2* states that "A business that collects a consumer's personal information shall implement reasonable security procedures and practices appropriate to the nature of the personal information to protect the personal information from unauthorised or illegal access, destruction, use, modification, or disclosure ". It doesn't specify the type of minimum privacy needed, but previous work [1], which gave some policy examples from GDPR used this type of PRIVACY. The corresponding attribute value is DP(1.0, 1e-5), where the numbers define the privacy budget. This kind of privacy guarantees that the consumer will not be affected by allowing the use of his data for any study or analysis. The goal is to extract useful information about a population without learning about a single individual. The privacy budget defines the level of accuracy of the final result, the smallest it is the lowest is the accuracy and the higher is the noise to protect the data.

FILTER is the attribute that defines eventual conditions to exclude some data. When defining the attribute values for policies of CPRA and GDPR the main ones involved consent. Any action, or PURPOSE - which will be explained in the following section -, requires specific consent from the consumer. Some consents are similar but might have different names, due to the different nomenclatures of the articles. GDPR and CPRA share some similarities with respect to some topics. For example, the articles regarding the age of the consumer or data subject for the usage of his data are very similar. They both allowed to define a set of attribute values shared among the policies. *Art. 8* from GDPR states: "The processing of the personal data of a child shall be lawful where the child is at least 16 years old. Where the child is below the age of 16 years, such processing shall be lawful only if and to the extent that consent is given or authorised by the holder of parental responsibility over the child. Member States may provide by law for a lower age for those purposes provided that such lower age is not below 13 years." *1798.120.c* in CPRA: "A business shall not sell or share the personal information of consumers if the business has actual knowledge that the consumer is less than 16 years of age, unless the consumer, in the case of consumers at least 13 years of age and less than 16 years of age, or the consumer's parent or guardian, in the case of consumers who are less than 13 years of age, has affirmatively authorised the sale or sharing of the consumer's personal information. A business that wilfully disregards

the consumer's age shall be deemed to have had actual knowledge of the consumer's age." We translated the constraint for age and guardian consent as:

```
(FILTER Age <16 AND FILTER Age >=13
AND FILTER GuardianConsent == 'Y')
OR (FILTER Age >=16)
```

Age is an integer which indicates the age of the consumer

GuardianConsent is a flag that can either be 'Y' or 'N' to indicate possible consent by the legal guardian of the underage consumer

Whenever an article states a specific request from the user to change, delete or retrieve the data, the access must be limited to the user's data only and not also others' data. In CPRA a situation like this happens with *art. 1798.110* which states: "A consumer shall have the right to request that a business that collects personal information about the consumer disclose to the consumer the following: the specific pieces of personal information it has collected about that consumer" (Figure 3.4). We assumed to have a possible condition that filters the data regarding only the consumer with a specific unique identifier: FILTER ConsumerID == 'n_consumer'. To generalise we set *n_consumer*. A similar situation happens with GDPR, but in that case *DataSubjectID* and *n_data_subject* are used for the condition: FILTER DataSubjectID == 'n_data_subject'.

```
ALLOW ROLE ConsumerID
    AND SCHEMA PersonalInformation
    AND FILTER RequestDisclosure == 'Y'
    AND FILTER ConsumerID == 'n_consumer'
```

Figure 3.4. CPRA: art. 1798.110 translation in LEGALEASE.

In Figure 3.4 it is indicated that a consumer with a unique identifier (i.e. ROLE ConsumerID) can manage data specified by SCHEMA, on the filtering condition that he previously requested a disclosure of the data (RequestDisclosure == 'Y') and that he access only his data (ConsumerID == 'n_consumer').

We then defined all the consents associated with the various cases and situations. CPRA has:

ConsentUse allows the use of the data and information, it is often associated to *BusinnesPurpose*, to define a generic situation;

ConsentCollection allows to store and to use data for future analysis that must be defined to the consumer;

ConsentRetention allows to store data but not use it for analysis;

ConsentShare and ConsentSell allow to share and sell the data to eventual external ROLE;

RequestDeletion allows to delete data after the consumer requested it explicitly;

RequestInaccurate is the consumer's request to modify the saved data when inaccurate. This modification can be done only if requested and approved by the consumer;

RequestDisclosure is used for the clauses associated with the consumer's request to access his data;

LimitUse is an explicit request of the consumer to limit the use of his data.

GDPR has some similar ones but with different names:

CorrectInformation is used when data is detected as incorrect and it is necessary to modify or delete it;

ConsentProcessing allows processing of the data;

ConsentProcessingIdentifiable allows processing of identifiable data (SCHEMA: *PDIdentifiable)*);

RequestProcessing is used when the data subject requests to obtain the data that has been processed;

RequestRectification allows data subject to request modification of wrong data;

RequestDeletion indicates the data subject's request to delete the data;

RequestDisclosure indicates if the data subject asked to obtain his personal data.

PURPOSE allows to define the reason to access the data. The values had been chosen based on the actions allowed by the different articles. The laws are applied in similar contexts, so the reason behind dome action on the data can be shared among them. This allowed to have some attribute values shared among the two laws:

BusinessPurpose associated with the business itself. It is a very broad definition;

Share and Sell related to the relationship with a third party;

DeletionData and ModificationData used for allowing the business or the controller to access the data and modify it. For CPRA, the consumer can, through a request, require a modification or deletion of his data. The request must be done properly and must be followed by a fast action by the business, as stated in *art. 1798.105* and *art. 1798.106*. GDPR requires accuracy so for *Art. 5.d*: "Personal data shall be: accurate and, where necessary, kept up to date; every reasonable step must be taken to ensure that personal data that are inaccurate, having regard to the purposes for which they are processed, are erased or rectified without delay ("accuracy")" (Figure 3.5[2]). Access to the data for these purposes can be done even without a data subject request in case the data is inaccurate. They are associated with a FILTER for the *ConsumerID* (CPRA) or *DataSubjectID* (GDPR) such that the resulting data accessed are only the ones of the requesting user.

```
ALLOW SCHEMA PersonalData
    AND FILTER CorrectInformation == 'N'
    AND (PURPOSE DeleteData OR PURPOSE ModificationData)
```

Figure 3.5. GDPR: Art. 5.d translation in LEGALEASE.

Retention allows the business to store data from the consumer. This kind of PURPOSE is a more specific subgroup of *BusinessPurpose*.

CPRA in *art. 1798.120.a* states: "A consumer shall have the right, at any time, to direct a business that sells or shares personal information about the consumer to third parties not to sell or share the consumer's personal information. This right may be referred to as the right to opt-out of sale or sharing" (Figure 3.6). This means that other extra purposes are *OptOutSell* and *OptOutShare*. This means that a consumer can modify the preferences of selling and sharing after the initial agreement.

[2]SCHEMA refers to the data that can be either deleted or modified (as stated by PURPOSE), if the data is in any way incorrect (FILTER CorrectInformation == 'N').

```
ALLOW ROLE ConsumerID
    AND ((SCHEMA ConsentSell
    AND PURPOSE OptOutSell
    AND FILTER ConsentSell == 'Y')
    OR (SCHEMA ConsentShare
    PURPOSE OptOutShare
    AND FILTER ConsentShare == 'Y'))
    AND FILTER ConsumerID == 'n_consumer'
```

Figure 3.6. CPRA: art.1798.120.a translation in LEGALEASE.

In Figure 3.6 we refer to the possibility for a consumer to opt out of selling and sharing the data. The clause defines with ROLE the consumer and its unique identifier and checks with FILTER ConsumerID == 'n_consumer', he will only access his data. The rest of the clause can be split into two parts, the first one for selling and the second one for sharing. SCHEMA refers to the consent that must be modified, PURPOSE is one of the two previously defined, and lastly to opt out the current consent must have a value set to 'Y'.

GDPR manages privacy not only regarding businesses and consumers. So it was necessary to introduce a set of different additional purposes to apply to LEGALEASE:

- *PublicInterest*;
- *ScientificResearch*;
- *HistoricalResearch*;
- *Processing* "includes the collection, recording, organisation, structuring, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, restriction, erasure or destruction of personal data";
- *EmploymentPurpose*;
- *SocialSecurity* or *SocialProtection* are associated with exercising and protecting rights for the controller or the data subject;
- *PoliticalPurpose*;
- *PhilosophicalPurpose*;
- *ReligiousPurpose*;
- *TradeUnionPurpose*;
- *LegalPurpose*;
- *MedicalPurpose*;
- *ScientificPurpose*;
- *HistoricalResearchPurpose*;
- *DirectMarketingPurpose*;
- *MonitorProtection*;
- *Transfer* is used when there are transactions with international organisations or third countries;
- *SupervisionPurpose* is used by *SupervisoryAuthority* when need to perform the task of supervision of correct execution of the law[3].

```
ALLOW ROLE SupervisoryAuthority
    SCHEMA PersonalData
    PURPOSE SupervisionPurpose
```

Figure 3.7. GDPR: Art. 58 translation in LEGALEASE.

We never used or applied the attribute REDACT, since there was not any law or article which explicitly required redaction of specific type of data for a specific level of privacy. REDACT allows a policy to require partial or complete removal of information from a column. For example, REDACT IPAddress(3 :) indicates that only the first 3 digits of the IP address can be accessed,

[3] *Art. 58*: "Each supervisory authority shall have all of the following investigative powers: to obtain, from the controller and the processor, access to all personal data and to access all information necessary for the performance of its tasks (Figure 3.7)."

while the rest can't be retrieved. In case a column can't be accessed at all the policy will contain REDACT IPAddress.

The final list of policy clauses written in Legalease, both from CPRA and GDPR, was 41. They were used separately to check compliance with the law, individually. We listed all of them with their respective article in the developer manual (CPRA 4.1 and GDPR 4.1.1).

3.2 Software systems

This chapter will explain in depth the software system chosen for the testing of PrivAnalyzer. For each source code, we will define an overview of it, by describing what it does and why it has been chosen. The following description of the architecture will describe the organisation of the code and the main elements of it. Depending on the code there will be an eventual definition of the work done on it to allow the use of PrivAnalyzer. In fact PrivAnalyzer requires the data to have a specific form. First of all, it's necessary to write a folder for the data, which will contain a text file with the policies and one with the metadata. For metadata, we mean the list of columns that define the database in which the data is stored. Generally, the list of columns contains all the important data, which we previously defined as *PersonalInformation* or *PersonalData* (Figure 3.1.2), and the list of all the consents. We wrote two versions for each file (policy and metadata), one associated with GDPR and one associated with CPRA and we analysed one at a time. All the files analysed with PrivAnalyzer need to have a function, which defines the actions for that file, named run with two arguments (Figure 3.8):

data_folder which was the variable with the directory for the root of PrivAnalyzer source code. It was used when needed to retrieve the data from the database;

****kwargs** is a dictionary containing data. It was used to pass sample data when analysing the functions by themselves, to test them as if it was a realistic application.

```
def run(data_folder, **kwargs)
```

Figure 3.8. Function format for PrivAnalyzer.

This approach was different from what we originally anticipated by reading the previous works [1], causing some extra work to obtain the final results. In fact, it limits what can be analysed. We initially expected an analysis of any Python code, but this requirement restrains the analysis to a specific code format. Any function that needs to be checked individually must be in the previously defined format and also written in a single and unique file. If the analysis of the function with PrivAnalyzer is not mandatory we can keep it with its original name and the original input arguments and it can be called when performing the analysis of other files.

Lastly, we identify all the important and relevant data to write as meta columns for the metafile and explain the choice and translation into Legalease.

3.2.1 LibreTaxi

LibreTaxi [16] is an open-source Uber proof-of-concept that works through Telegram [17]. It works as a bot that allows users to send a message to look for car rides, as passengers, or to offer car rides, as drivers; depending on the format of the post the user specifies his needs. There are two main bots managed by LibreTaxi: the first one allows the user to interact and create a post or change his location stored in the database. For a post creation, the driver and the passenger need to both write a starting point, destination, date and time of the ride that they offer or that they are looking for. A passenger is also required to write the number of people who need to find

a ride; they can give eventually extra information, such as the form of payment they can use. Drivers, on the other hand, will need to always specify the form of payment that they accept, when writing the post. After sending the message every user within a radius of 25 kilometres (15 miles) will be notified, with the use of the second bot [18]. The second bot is a read-only text thread, that allows one to read all the posts, eventually report them or find out who are the new users who joined LibreTaxi. Users are not allowed to communicate with each other through LibreTaxi, if they want to they can have a direct conversation by clicking on the handles of the post. The handle directs them to a new private chat on Telegram.

Libretaxi is composed of around 2000 lines of code divided into 13 folders and a main file "libretaxi.go", from which the main function is called to launch the system. The all code is implemented in Go, so not already testable for PrivAnalyzer. The set of folders are:

bin which contains two files of around 5 lines each, then manages the deployment of the system. They are used and accessed in case of personal deployment of the system, instead of using the already running bot on Telegram

callback contains a Go file with a callback function that manages an action. It receives the context, that allows to obtain information on the user or post, and a string, that gets unmarshalled to know the specific action. An action can be REPORT_POST, which will increase the report count of a user, or SHADOW_BAN, which sets the state of a user as shadowbanned.

config contains the file that manages the configuration initialisation;

context manages the context for the deployment of the system. This variable is retrieved to manage the communication and also eventual access to storage.

db contains the .db file which initialises the tables for posts and users;

locales has six files will the list of default messages to send to a user based on an interaction with the bot. Each file has the same messages written in different languages: English, Spanish, Portuguese, Brazilian Portuguese and Russian. Each message has an identifier, which is the one called when trying to access it.

menu contains five files. Each file implements the functions for the different interactions between the user and the first bot. Each user has a current state variable that indicates which interaction has been performed. "menu.go" will retrieve and check the user state and act depending on it. The possible interactions are the initialisation of the user (*init_menu*), asking for the location (*ask_location_menu*), posting of a new message (*post_menu*) or waiting for the user to write a new post (*feed_menu*).

objects define the struct for the objects of Post and User, explained later in more detail.

rabbit implements the use of RabbitMQ, which is an open-source message broker. It is used to manage the message exchange between the user and the system. It is used to create different channels for the communication and the publishing of the messages.

repository manages the interaction with the database. The file inside of it has a list of functions that access the database to retrieve, modify or delete data.

sender implements specifically the object responsible for the sending of the messages to the system.

util has a file with a function responsible to add before "_ * [ ] ( ) ~ ># + - = — { } . !" an escape character "\\", if they are inside a given string.

validation holds the file implementing the function that checks for the messages sent by the users. The messages must satisfy a format, based on the action, but also not be offensive in any way. In case these conditions are not pleased a message error is returned and sent to the user.

To manage the use with Telegram LibreTaxi uses Telegram Bot API, which is passed and used with Context struct.

The analysis of this program required two main preprocessing steps: translating the code into Python, to allow analysis by PrivAnalyzer, and identifying the information collected and their importance with respect to privacy. The first part is very important, because LibreTaxi is written in Go. We translated every single file and the libraries into a Python version. This translation does not allow the use of LibreTaxi through Python but guarantees to check and analyse all its components with PrivAnalyzer. It was possible to check the compliance for both the main program, so the main file, and the single files which implement only some functions. We analysed both of them to obtain a fuller and more complete result. With this approach, it was possible to analyse the various section of code separately and to identify the possible causes of a negative output of the analysis. We were able to see and find which were the specific functions which used data incorrectly. When we performed the translation, it was necessary to write the code in a way that was usable for PrivAnalyzer. We divided all the functions into separate files, which could be tested individually. The need to have all functions named as run and with specific arguments extended the work on the translation since any time a function was called any argument needed by it was added to the dictionary as extra arguments and then later retrieved and extracted. For example in the original when saving a post the function would receive a struct of the post and any information directly obtained from it (Figure 3.9).

```
func (repo *Repository) SavePost(post *objects.Post){
        ...post.UserId, post.Text, post.Lat, post.Lon,
            post.ReportCnt).Scan(&post.PostId)
}
```

Figure 3.9. LibreTaxi: passing arguments in Go

When we translated it the information needed to be extracted one by one before their use from ***kwargs* (Figure 3.10).

```
def run(data_folder, **kwargs):
    pd = kwargs.get('pandas')
    ConsumerID = kwargs.get('extra_args').get('user_id')
    MessageTxt = kwargs.get('extra_args').get('text')
    lon = kwargs.get('extra_args').get('lon')
    lat = kwargs.get('extra_args').get('lat')
    ReportCnt = kwargs.get('extra_args').get('report_cnt')
```

Figure 3.10. LibreTaxi: passing arguments in Python

We skipped the translation for some folders and respective files such as bin, config, context, rabbit and sender. The object files were used as a reference to then write the meta files for the database in the analyser. We created two meta files, one for posts and one for users, and also their respective sets of policies. Depending on the information needed. The biggest proles encountered were due to the understanding of the new language. Go was a language never implemented before, so it was important to understand how it worked and how to perform a correspondence with Python. It was important to preface and decide from the beginning that the translation would generate a final source code which wouldn't work if deploy. We decided to have a translation that would have the same meaning, with respect to the action it could perform. This is why we discarded the translation of the previously listed files since they were necessary only for the deployment and we could see how no result could come from an analysis with PrivAnalyzer.

With respect to the second part, We learned from the code that LibreTaxi collects and stores in its database both all the posts and data regarding the users. With respect to the user, it collects data from Telegram itself or from the user's interaction with the bot. From Telegram it obtains name, username and location, as a combination of latitude and longitude; whereas in the second case, when creating the database instance it will add:

Unique identifier is given when first interacting with the first bot and allows to access the database and retrieve the information from that given user;

Report Count Each post can be reported by the other users, so it is necessary to count how many times it happens. The report can happen when a post does not follow the guidelines and can be offensive, wrong;

if the user is currently shadow banned, requiring to change how he can interact with the program;

Language Code is the language through which the user interacts with the program. Based on the setting of Telegram, the program can communicate, in English, Spanish, Portuguese, Brazilian Portuguese, and Russian.

Menu ID is the current state of the user. It identifies where the user is at when interacting with LibreTaxi. It allows to know what needs to be displayed or how to manage the user's actions.

Meanwhile, each post has some information that is given by the user when writing it:

- User that sent the message, so both username and unique identifier;
- Date of the post;
- Location of the post, as latitude and longitude;
- Content of the message;
- Language code.

Similarly to the users, every post will also have a unique identifier, to find and retrieve the message in the future, and a report count, based on the number of users who reported this specific post.

Based on GDPR and CPRA, a part of this information must be regulated and checked to guarantee privacy to the users. In Section 3.2.3 we defined, as one of the main *SCHEMA* values, PersonalInformation and PersonalData, which group together a large number of values, due to the many possible applications of the law. In LibreTaxi we were able to narrow this group to: *Username*,*LastName*,*FirstName*, *Longitude*, *Latitude*, *ConsumerID*, which is the name of column given to the unique identifier of the user and lastly *MessageText*, the content of the post sent by the users. Report count, shadow ban and menu id are not considered pieces of information that require the same level of privacy, so they do not need to be regulated and controlled by the privacy clauses. The previously listed information required to be accessed with the specific consents of the users, for this reason, we added to the users' data file a series of columns that defined the users' consents. When writing the meta.txt file, used by PrivAnalyzer, we wrote all the previously listed attributes and also all the consents, that GDPR and CPRA required.

3.2.2 Selfmailbot

Selfmailbot [19] is a source code to implement a bot that allows sending messages, photos and voice messages to a defined mail. Given the code, it is possible to deploy it directly from the machine, but it is already implemented as a Software as a service (*SaaS*) through Telegram (*Reference link:* `selfmailbot.co`). For each user, there is a mail associated with him, which is personally given by him to the bot. When given, Selfmailbot will send a confirmation link to it, as soon as the mail has been confirmed it is possible to send content through forms of emails.

The GitHub repository provided, both the source code and the testing performed over the various functions. We worked and analysed only the source code. It has 3 folders and 8 singular files. The folders are:

email contains the text for the confirmation email

html contains three HTML files for possible feedback when a user clicks on the confirmation link

messages contains a set of text files for all the messages that the user could receive based on the action he performs.

The files are written in Python and used to manage the system, for a total of around 550 lines of code. The main file (*app.py*) initialises the database, which will contain all the users' information, and start the Updater. The Updater is a class from the Telegram package, that allows to constantly listen for eventual tasks to execute. It waits for a user interaction to then perform the associated response. In this file, there are implemented all the main response methods:

start called when there is the first interaction, causes an initial welcome message to be sent in the chat;

resend sends again a confirmation email;

rest_emai removes the current saved email address for the specific user;

confirm_email checks for the email to be confirmed correctly;

send_text_message or send_voice or send_photo send to the user saved email, respectively, a text, a voice message or a photo;

prompt_for_setting_email sends to the Telegram chat a message for the user to send an email address for performing the tasks;

send_confirmation send the confirmation email to the mail address given by the user;

prompt_for_confirm sends on the Telegram chat a message to the user to remind him to confirm the email. It allows the user to choose between resending the confirmation or changing the address.

. To manage the asynchronous executions of the tasks, SELFMAILBOT uses the package Celery. The package is implemented in one of the separate files and gathers the information needed for sending the emails. Emails are sent with the use of the package "pystmark", which sends messages with the use of an API key. The storage for the users' information is controlled with "sqlite".

Differently from LIBRETAXI the code is already written in Python, so the translation was not necessary. It was necessary to just modify the code in a way that was possible to analyse it with PRIVANALYZER. We started by modifying any access to the storage of information. The code must access the information from the data stored in PRIVANALYZER, this allows to control and check the compliance with the privacy clauses. For example, the following snippet shows how a user instance is retrieved (Figure 3.11). We pass with ***kwargs* the arguments from the original function, we then obtain the user_id and retrieve from the database the user that has it. We then returned the instance by discarding all the consent information which are not going to be used by SELFMAILBOT.

After dividing all the functions into singular files, we modified the input of any kind of functions, other than the arguments needed by the original code we added the ones needed by PRIVANALYZER. This is what has been done also for LIBRETAXI (Figure 3.8). PRIVANALYZER needs to receive the directory for the storage of the data and the libraries it implements, so any code that is analysed must receive those kinds of data. For this code, we were able to check and analyse the single files and function; the main code is implemented in a way that does not allow a successful analysis by itself. The modification caused to have a final Python code, that could be analysed, but no possible deployment was possible, similarly to what happened with LIBRETAXI. The main difficulties regarded changing the code in a way that was still performing the action but allowing a correct and full analysis with PRIVANALYZER. The limitations connected to PRIVANALYZER, such as the local metadata for storage and some internally implemented libraries, limited some performance, causing to change the code in a way that still could make sense. Needing to only

```
#original code
def get_user_instance(user: telegram.User, chat_id: int) -> User:
    instance, created = User.get_or_create(
        pk=user.id,
        defaults=dict(
        pk=user.id,
        full_name=user.full_name,
        username=user.username,
        confirmation=str(uuid.uuid4()),
        chat_id=chat_id,
        ),
    )
    return instance

#translation for PrivAnalyzer
def run(data_folder, **kwargs):
    pd = kwargs.get('pandas')
    user_id = kwargs.get('extra_args').get('user_id')

    selfmail_users = pd.read_csv(data_folder + "data.csv")
    selfmail_users = selfmail_users[selfmail_users.ConsumerID == user_id]

    return selfmail_users.drop(
    ['ConsentUse', 'ConsentShare', 'ConsentShare', 'ConsentSell',
        'ConsentCollection', 'GuardianConsent', 'RequestDeletion',
        'RequestDisclosure', 'RequestInaccurate', 'LimitUse', 'Age'], axis=1)
```

Figure 3.11. Selfmailbot: modification user retrieval.

modify the code and not translate and rewrite all of it, was a big advantage and allowed to save some time.

We then needed to identify the information used by the bot to perform any kind of task. Each user is stored in the database as an instance with a set of data that are both retrieved from the context and also given by the user interaction with the bot. Starting from Telegram itself, each user has a username and a full name, retrieved from Telegram. As introduced before, SELFMAILBOT will request and confirm an email before any other possible future actions. Both these data are stored and unique for each user, in fact, no email can be associated with more than one user. To keep track of the confirmed email, there is a flag regarding the confirmation of the email itself, set as default to 'False'. To keep track of the communication there is a counter regarding the number of messages sent by the user in the specific chat. Lastly, to distinguish and identify each instance there is a unique identifier for the user and also an identifier for the chat itself, to allow its retrieval.

To allow the analysis we translated these data into LEGALEASE language:

ConsumerID or DataSubjectID is the unique identifier for the user;

Created is the timestamp for when the new user is stored by SELFMAILBOT

FullName is the name of the user, specified on Telegram;

Username

Email is the currently saved email for the specific user

IsConfirmed is the flag to check the confirmation of the *Email*;

MessageCnt is the number of messages from the user with the given *Email*;

ConfirmationLink is the link sent to the email when it needs to be confirmed;

ChatID is a unique identifier for the chat of the user with the bot. It can be shared among different instances when the user changes his *Email*.

They are not all sensitive information, defined as PersonalInformation or PersonalData (Figure 3.1.2). Only *ConsumerID*, unique identifier of the user, *FullName*, *Username* and *Email* needed to be protected and translated as possible SCHEMA attribute values for the policy. The remaining data of each user instance are not of the same level of sensitiveness and do not need the same regulations for their use and access, but they had been translated into LEGALEASE language since they need to be added into the metafile, but they were not inserted in any of the clauses of the list of policies. As previously done for LIBRETAXI all these attributes are inserted into the meta.txt file with all the consent attributes, to allow comparison by PRIVANALYZER with the policies and the code.

3.2.3 Traccar

TRACCAR [20] is an open source GPS tracking system. It allows to manage and control of devices' location. It is a convenient solution to find devices and always know where they are. It is a software that has two different phone applications, both for iOS and Android. TRACCAR CLIENT can be downloaded on the device that must be tracked. It saves and stores information to identify the specific device. TRACCAR MANAGER allows to find and look up different devices. The manager requires to create an account with an email and a password. After accessing the account users can find any device, which had been previously registered with TRACCAR CLIENT, through the unique identifier.

The work done with TRACCAR is a mixture of the one previously done with LIBRETAXI and SELFMAILBOT. We started by translating the source code, which is divided into front-end and back-end. The back end is written in Java and defines all the main components and elements used to perform the task. It is the one that manages the storage of the information regarding the devices, the various positions and the users. It is around 3200 lines of code divided into 22 folders plus 29 external Java files. Since the code was a lot bigger than with LIBRETAXI we initially translated from Java to Python and then modified the Python code in a way that could be analysed by PRIVANALYZER, like what we did for SELFMAILBOT. The back end is written in a way that could be associated with multiple possible front-ends, causing to have even multiple implementations for the same actions. An example is that for the retrieval of the location the system has multiple location providers:

- Google;
- Mozilla;
- OpenCelliD, which is a database of cell towers identifier, when called a GET the closer ID is obtained;
- Unwired, based on the WiFi access point, the network and the cell tower to which the device is connected.

Out of all the folders, we selected the ones that were important and valuable for our work. Since the implementation is through Java it is object-oriented, having a total of 33 files that implement the various object/model. First of all the *model* folder, which defines all the objects used through the system. It allowed us to define which data were more important than others and needed protection depending on the law, later we will explain in more detail this work. Secondly, we focused on the translation of *storage* folder, which contains a set of files which implement the database and memory management. Figure 3.12 and Figure 3.13 show the translation for the retrieval of a specific set of columns from a database. The original code builds a query to access the database, based on the input information obtained. It starts by checking the columns that

```
public <T> List<T> getObjects(Class<T> clazz, Request request) throws
    StorageException {
    StringBuilder query = new StringBuilder("SELECT␣");
    if (request.getColumns() instanceof Columns.All) {
        query.append('*');
    } else {
        query.append(formatColumns(request.getColumns().getColumns(clazz,
            "set"), c -> c));
    }
    query.append("␣FROM␣").append(getStorageName(clazz));
    query.append(formatCondition(request.getCondition()));
    query.append(formatOrder(request.getOrder()));
    try {
        QueryBuilder builder = QueryBuilder.create(config, dataSource,
            objectMapper, query.toString());
        for (Map.Entry<String, Object> variable :
            getConditionVariables(request.getCondition()).entrySet()) {
        builder.setValue(variable.getKey(), variable.getValue());
        }
        return builder.executeQuery(clazz);
    } catch (SQLException e) {
        throw new StorageException(e);
    }
}
```

Figure 3.12. Traccar back-end in Java: retrieval of selected columns from storage

need to be retrieved, causing an eventual filter if not all of them are selected. Then it will add the class to access, the eventual conditions and ordering (Figure 3.12).

The translation into Python was very straight and similar since all the arguments were passed as extra arguments through the dictionary ***kwargs*. For PrivAnalyzer there was no SQL database associated, so the retrieval didn't need a query creation. We started by selecting the file from which the data needed to be obtained. The only data stored were regarding, *Device*, *User* and *Position*. Filtering and ordering were then performed. For filtering we passed an array containing, in order, the column of reference and the Boolean to define eventual ascending. The selection of the columns was done as last action. The implementation wasn't based on objects so all the data were passed as arrays from which we retrieved the data (Figure 3.12). Traccar can be used with any major SQL database system, allowing easier and shorter work to understand the actions taken by the system.

```
def run(data_folder, **kwargs):
    pd = kwargs.get('pandas')
    clazz = kwargs.get('extra_args').get('clazz')
    columns = kwargs.get('extra_args').get('columns')
    conditions = kwargs.get('extra_args').get('conditions')
    sort_cond = kwargs.get('extra_args').get('sort_cond')
    if str(clazz) == 'Device':
        traccar_data = pd.read_csv(data_folder + "devices/data.csv")
    elif str(clazz) == 'User':
        traccar_data = pd.read_csv(data_folder + "users/data.csv")
    elif str(clazz) == 'Position':
        traccar_data = pd.read_csv(data_folder + "positions/data.csv")
    for cond in conditions:
        traccar_data = traccar_data[cond]
    traccar_data =
        traccar_data.sort_values(by=sort_cond[0],axis=1,ascending=sort_cond[1])
    traccar_data = traccar_data[columns]
    return traccar_data
```

Figure 3.13. TRACCAR back-end in Python: retrieval of selected columns from storage

The user can have an email address associated with his account, allowing eventual communication, so the last mail folder taken into consideration was *mail*, which managed the communication with the user through the email address. All the other folders were taken into account and consideration since the object contained in them were imported for their use in the selected files. They weren't discarded, but we avoided their individual analysis. The main reason for this choice is the fact that, with the translation into Python, it was hard to be loyal to the original performance, but the folders also weren't directly involved with the data that needed protection.

For the front end, there are different source codes, depending on the deployment methods. TRACCAR can be accessed through two applications, either Android ([21], [22]) or iOS ([23], [24]), or through a website [25]; all these cases had different source codes written in the corresponding languages. For Android, the origin code is written in Kotlin, for iOS in Swift, while the web application is written in JavaScript. The mobile applications are two for each language since there are TRACCAR MANAGER and TRACCAR CLIENT. It was necessary to analyse also these options because they were the ones managing eventual permissions and consent requests for some information usage. For example, the android application TRACCAR MANAGER, before accessing the location, requests permission from the user (Figure 3.14).

```
if (ActivityCompat.shouldShowRequestPermissionRationale(activity,
    Manifest.permission.ACCESS_FINE_LOCATION)) {
    AlertDialog.Builder(activity)
        .setMessage(R.string.permission_location_rationale)
        .setNeutralButton(android.R.string.ok) { _: DialogInterface?, _: Int
            ->
            geolocationRequestOrigin = origin
            geolocationCallback = callback
            ActivityCompat.requestPermissions( activity,
                arrayOf(Manifest.permission.ACCESS_FINE_LOCATION),
                REQUEST_PERMISSIONS_LOCATION)
        }
        .show()
    } else {
        geolocationRequestOrigin = origin
        geolocationCallback = callback
        ActivityCompat.requestPermissions(
            activity,
            arrayOf(Manifest.permission.ACCESS_FINE_LOCATION),
            REQUEST_PERMISSIONS_LOCATION
            )
}
```

Figure 3.14. Traccar Manager Android Kotlin: request permission for location

When translating into Python, we transformed it into a function which receives as an argument the consent and changes it from the data stored in the database. Whenever it is needed to access the location the consent will be checked. To do so, we retrieve the data for the devices from the storage, calling *include_columns_analizer* (explained in Figure 3.13). The function receives as extra data the name of the folder (*Device*) and the set of columns to return (*PositionID, ConsentAccess, DeviceID*). After that, we filter to get only the data for the specific current identifier. Lastly, the consent is changed and the instance, as it is, is returned (Figure 3.15).

```
def run(data_folder, **kwargs):

    pd = kwargs.get('pandas')
    deviceid = kwargs.get('extra_args').get('deviceid')
    consent = kwargs.get('extra_args').get('consent')

    extra_args = kwargs.get('extra_args')
    extra_args.__setitem__('clazz','Device')
    extra_args.__setitem__('columns',['PositionID', 'ConsentAccess',
        'DeviceID'])
    kwargs.__setitem__('extra_args',extra_args)
    res = include_columns_analizer.run(data_folder, **kwargs)

    res = res[res.DeviceID = deviceid]
    res.ConsentAccess = consents
    return res
```

Figure 3.15. Traccar Manager Android Python: request permission for location

On average, between the different source codes, the number of lines of code is around 1000 divided into 3 big folders. Between these lines and folders there are the ones containing the information for the display of information, so from an interface point of view. These were discarded since they don't manage the usage of the data. The translation work was analogous to the back-end part: an initial translation to Python and then a modification to allow analysis with PrivAnalyzer. The main problem was associated with Swift since it is a new language that we didn't encounter before, but it wasn't so far off from our experience and knowledge with Kotlin.

At this point, we analysed and defined which were the main information used and stored by Traccar. The system separately stores data with respect to the devices, which must be located, the positions and the users. As stated before, with Traccar Client a user to register a device, without any kind of user registration. Each device is defined by a name, given by the user and an identifier, unique within the system. The code stores any kind of movement or change in the device status. Each device instance will define the status (online, offline or unknown), the timestamp of the last update and the identifier of the position with respect to the timestamp. Whenever a device changes position and moves the new location is stored in the system. Each change is defined by a unique position identifier, the identifier of the device, the time with respect to the server and also with respect to the device, the location and lastly the speed of the device. The location is identified as altitude, longitude, latitude and also as an address. Lastly, a user can access his information with Traccar Manager, by creating an account and writing the unique device identifier. Each user is saved in the system with a unique identifier, the name, the mail, the password in a hashed form, the salt to solve the password, the latitude and the longitude. Extra information is stored, such as preferences regarding language, units for distance and speed and if the user is an administrator or not.

After translating this data into a language that PrivAnalyzer could understand, we confronted them with the privacy limitations required by CPRA and GDPR to identify which of them must be regulated and protected. *PersonalInformation* and *PersonalData* in this context can be defined based on the kind of data:

Devices : *DeviceID*,*PositionID*;

Positions : *PositionID*,*DeviceID*,*Latitude*,*Longitude*,*Altitude*,*Address*;

Users : *ConsumerID* or *DataSubjectID* - depending on the law we are analysing-, *Email*, *HashedPassword*, *Salt*, *Latitude*, *Longitude*

The previously listed names were set as possible SCHEMA values in the policies' file. All the other data, even if stored by the system do not need the same level of protection and control. We still translated their name into a Legalease similar language. So in the end, all the information written in a language comprehensible by PrivAnalyzer were set as columns name in the meta.txt file with also all the consents listed in Section 3.1.2.

Chapter 4

Software manuals

4.1 Developer manual

4.1.1 CPRA translation

"1798.100. General Duties of Businesses that Collect Personal Information"

c. "A business' collection, use, retention, and sharing of a consumer's personal information shall be reasonably necessary and proportionate to achieve the purposes for which the personal information was collected or processed, or for another disclosed purpose that is compatible with the context in which the personal information was collected, and not further processed in a manner that is incompatible with those purposes."

```
ALLOW ROLE Business
    AND SCHEMA PersonalInformation
    AND (((FILTER ConsentUse == 'Y' OR FILTER ConsentCollection
        == 'Y' OR FILTER ConsentRetention == 'Y')
    AND PURPOSE BusinessPurpose)
    OR (FILTER ConsentShare == 'Y'
    AND PURPOSE Share)
    OR (FILTER ConsentSell == 'Y'
    AND PURPOSE Sell))
```

d. "A business that collects a consumer's personal information and that sells that personal information to, or shares it with, a third party or that discloses it to a service provider or contractor for a business purpose shall enter into an agreement with the third party, service provider, or contractor, that:

(a) Obligates the third party, service provider, or contractor to comply with applicable obligations under this title and obligate those persons to provide the same level of privacy protection as is required by this title.

(b) Grants the business rights to take reasonable and appropriate steps to help ensure that the third party, service provider, or contractor uses the personal information transferred in a manner consistent with the business' obligations under this title."

```
ALLOW (ROLE ServiceProvider OR ROLE Contractor OR ROLE ThirdParty)
    AND ((PURPOSE Share
    AND FILTER ConsentShare == 'Y')
    OR (PURPOSE Sell
    AND FILTER ConsentSell == 'Y'))
    AND PURPOSE BusinessPurpose
    AND PRIVACY DP(1.0, 1e-5)
```

e. “A business that collects a consumer’s personal information shall implement reasonable security procedures and practices appropriate to the nature of the personal information to protect the personal information from unauthorised or illegal access, destruction, use, modification, or disclosure in accordance with Section 1798.81.5.”

```
ALLOW ROLE Business
    AND SCHEMA PersonalInformation
    AND PRIVACY DP(1.0, 1e-5)
    AND FILTER ConsentRetention == 'Y'
```

“1798.105. Consumers’ Right to Delete Personal Information”

1. “A business that receives a verifiable consumer request from a consumer to delete the consumer’s personal information pursuant to subdivision (a) of this section shall delete the consumer’s personal information from its records, notify any service providers or contractors to delete the consumer’s personal information from their records, and notify all third parties to whom the business has sold or shared the personal information to delete the consumer’s personal information unless this proves impossible or involves disproportionate effort.

3. A service provider or contractor shall cooperate with the business in responding to a verifiable consumer request, and at the direction of the business, shall delete, or enable the business to delete and shall notify any of its own service providers or contractors to delete personal information about the consumer collected, used, processed, or retained by the service provider or the contractor. The service provider or contractor shall notify any service providers, contractors, or third parties who may have accessed personal information from or through the service provider or contractor, unless the information was accessed at the direction of the business, to delete the consumer’s personal information unless this proves impossible or involves disproportionate effort. A service provider or contractor shall not be required to comply with a deletion request submitted by the consumer directly to the service provider or contractor to the extent that the service provider or contractor has collected, used, processed, or retained the consumer’s personal information in its role as a service provider or contractor to the business.”

```
ALLOW (ROLE Business OR ROLE ServiceProvider OR ROLE Contractor OR ROLE
    ThirdParty)
AND FILTER ConsumerID == 'n_consumer'
AND FILTER RequestDeletion == 'Y'
AND PURPOSE DeletionData
```

“1798.106. Consumers’ Right to Correct Inaccurate Personal Information”

a. “A consumer shall have the right to request a business that maintains inaccurate personal information about the consumer to correct that inaccurate personal information, taking into account the nature of the personal information and the purposes of the processing of the personal information.

b. A business that collects personal information about consumers shall disclose, pursuant to Section 1798.130, the consumer’s right to request correction of inaccurate personal information.

c. A business that receives a verifiable consumer request to correct inaccurate personal information shall use commercially reasonable efforts to correct the inaccurate personal information as directed by the consumer, pursuant to Section 1798.130 and regulations adopted pursuant to paragraph (8) of subdivision (a) of Section 1798.185.”

```
ALLOW ROLE Business
    AND SCHEMA PersonalInformation
    AND FILTER RequestInaccurate == 'Y'
    AND FILTER ConsumerID == 'n_consumer'
    AND PURPOSE ModificationData
```

"1798.110. Consumers' Right to Know What Personal Information is Being Collected. Right to Access Personal Information"

"A consumer shall have the right to request that a business that collects personal information about the consumer disclose to the consumer the following: the specific pieces of personal information it has collected about that consumer."

```
ALLOW ROLE ConsumerID
    AND SCHEMA PersonalInformation
    AND FILTER RequestDisclosure == 'Y'
    AND FILTER ConsumerID == 'n_consumer'
```

"1798.120. Consumers' Right to Opt Out of Sale or Sharing of Personal Information"

a. "A consumer shall have the right, at any time, to direct a business that sells or shares personal information about the consumer to third parties not to sell or share the consumer's personal information. This right may be referred to as the right to opt-out of sale or sharing.

b. A business that sells consumers' personal information to, or shares it with, third parties shall provide notice to consumers, pursuant to subdivision (a) of Section 1798.135, that this information may be sold or shared and that consumers have the *right to opt-out* of the sale or sharing of their personal information."

```
ALLOW ROLE ConsumerID
    AND ((SCHEMA ConsentSell
    AND (PURPOSE OptOutSell
    AND FILTER ConsentSell == 'Y'))
    OR (SCHEMA ConsentShare
    PURPOSE OptOutShare
    AND FILTER ConsentShare == 'Y'))
    AND FILTER ConsumerID == 'n_consumer'
```

c. "Notwithstanding subdivision (a), a business shall not sell or share the personal information of consumers if the business has actual knowledge that the consumer is less than 16 years of age, unless the consumer, in the case of consumers at least 13 years of age and less than 16 years of age, or the consumer's parent or guardian, in the case of consumers who are less than 13 years of age, has affirmatively authorised the sale or sharing of the consumer's personal information. A business that wilfully disregards the consumer's age shall be deemed to have had actual knowledge of the consumer's age.

d. A business that has received direction from a consumer not to sell or share the consumer's personal information or, in the case of a minor consumer's personal information has not received consent to sell or share the minor consumer's personal information, shall be prohibited, pursuant to paragraph (4) of subdivision (c) of Section 1798.135, from selling or sharing the consumer's personal information after its receipt of the consumer's direction, unless the consumer subsequently provides consent, for the sale or sharing of the consumer's personal information."

```
ALLOW ROLE Business
    AND ((FILTER ConsentShare == 'Y'
    AND PURPOSE Share)
    OR (FILTER ConsentSell == 'Y'
    AND PURPOSE Sell))
    AND ((FILTER Age <16 AND FILTER Age >=13
        AND FILTER GuardianConsent == 'Y')
        OR (FILTER Age >=16))
```

"1798.121. Consumers' Right to Limit Use and Disclosure of Sensitive Personal Information"

a. "A consumer shall have the right, at any time, to direct a business that collects sensitive personal information about the consumer to limit its use of the consumer's sensitive personal information to that use which is necessary to perform the services or provide the goods reasonably expected by an average consumer who requests those goods or services, to perform the services set forth in paragraphs (2), (4), (5), and (8) of subdivision (e) of Section 1798.140, and as authorised by regulations adopted pursuant to subparagraph (C) of paragraph (19) of subdivision (a) of Section 1798.185. A business that uses or discloses a consumer's sensitive personal information for purposes other than those specified in this subdivision shall provide notice to consumers, pursuant to subdivision (a) of Section 1798.135, that this information may be used, or disclosed to a service provider or contractor, for additional, specified purposes and that consumers have the right to limit the use or disclosure of their sensitive personal information.

b. A business that has received direction from a consumer not to use or disclose the consumer's sensitive personal information, except as authorised by subdivision (a), shall be prohibited, pursuant to paragraph (4) of subdivision (c) of Section 1798.135, from using or disclosing the consumer's sensitive personal information for any other purpose after its receipt of the consumer's direction unless the consumer subsequently provides consent for the use or disclosure of the consumer's sensitive personal information for additional purposes.

c. A service provider or contractor that assists a business in performing the purposes authorised by subdivision (a) may not use the sensitive personal information after it has received instructions from the business and to the extent it has actual knowledge that the personal information is sensitive personal information for any other purpose. A service provider or contractor is only required to limit its use of sensitive personal information received pursuant to a written contract with the business in response to instructions from the business and only with respect to its relationship with that business.

d. Sensitive personal information that is collected or processed without the purpose of inferring characteristics about a consumer is not subject to this section, as further defined in regulations adopted pursuant to subparagraph (C) of paragraph (19) of subdivision (a) of Section 1798.185, and shall be treated as personal information for purposes of all other sections of this act, including Section 1798.100."

```
ALLOW (ROLE Business OR ROLE ServiceProvider OR ROLE Contractor OR ROLE
    ThirdParty)
    AND SCHEMA SensitivePI
    AND PURPOSE BusinessPurpose
    AND FILTER LimitUse == 'N'
```

"1798.145. Exemptions"

"The obligations imposed on businesses by this title shall not restrict a business' ability to:

4. Cooperate with a government agency request for emergency access to a consumer's personal information if a natural person is at risk or danger of death or serious physical injury provided that:

   A. The request is approved by a high-ranking agency officer for emergency access to a consumer's personal information.

   B. The request is based on the agency's good faith determination that it has a lawful basis to access the information on a non-emergency basis.

   C. The agency agrees to petition a court for an appropriate order within three days and to destroy the information if that order is not granted"

```
ALLOW ROLE GovernamentAgency
    AND SCHEMA PersonalInformation
    AND FILTER ConsumerID == 'n_consumer'
```

6. "Collect, use, retain, sell, share, or disclose consumers' personal information that is deidentified or aggregate consumer information."

```
ALLOW ROLE Business
    AND SCHEMA PersonalInformation
    AND (PURPOSE Retention OR PURPOSE BusinessPurpose OR PURPOSE Sell OR
        PURPOSE Share)
    AND (PRIVACY Anonymization OR PRIVACY Aggregation)
```

4.1.2 GDPR translation

"Art. 5 Principles relating to processing of personal data"

"Personal data shall be:

b. collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes; further processing for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes shall, in accordance with Article 89(1), not be considered to be incompatible with the initial purposes ("purpose limitation");

e. kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed; personal data may be stored for longer periods insofar as the personal data will be processed solely for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes in accordance with Article 89(1) subject to implementation of the appropriate technical and organisational measures required by this Regulation in order to safeguard the rights and freedoms of the data subject ("storage limitation");"

```
ALLOW SCHEMA PersonalData
    AND (PURPOSE BusinessPurpose OR PURPOSE PublicInterest OR PURPOSE
        ScientificResearch OR PURPOSE HistoricalResearch)
```

d. "accurate and, where necessary, kept up to date; every reasonable step must be taken to ensure that personal data that are inaccurate, having regard to the purposes for which they are processed, are erased or rectified without delay ("accuracy");"

```
ALLOW SCHEMA PersonalData
    AND FILTER CorrectInformation == 'N'
    AND (PURPOSE DeleteData OR PURPOSE ModificationData)
```

f. “processed in a manner that ensures appropriate security of the personal data, including protection against unauthorised or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organisational measures (“integrity and confidentiality”).”

```
ALLOW SCHEMA PersonalData
    AND PRIVACY Anonymization
```

“Art. 6 Lawfulness of processing”

“Processing shall be lawful only if and to the extent that at least one of the following applies:

a. the data subject has given consent to the processing of his or her personal data for one or more specific purposes;”

```
ALLOW SCHEMA PersonalData
    AND FILTER ConsentProcessing == 'Y' AND FILTER Age >= 16
    AND PURPOSE Processing
```

b. “processing is necessary for the performance of a contract to which the data subject is party or in order to take steps at the request of the data subject prior to entering into a contract;”

```
ALLOW SCHEMA PersonalData
    AND PURPOSE Processing AND PURPOSE BusinessPurpose
```

c. “processing is necessary for compliance with a legal obligation to which the controller is subject;”

d. “processing is necessary in order to protect the vital interests of the data subject or of another natural person;”

```
ALLOW SCHEMA PersonalData
    AND (ROLE PublicAuthority OR ROLE LegalAuthority)
    AND PURPOSE Processing
```

e. “processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller;”

```
ALLOW SCHEMA PersonalData
    AND PURPOSE Processing AND PURPOSE PublicInterest
    AND ROLE Controller
```

“Art. 7 Conditions for consent”

“Where processing is based on consent, the controller shall be able to demonstrate that the data subject has consented to processing of his or her personal data.”

```
ALLOW SCHEMA PersonalData
    AND FILTER ConsentProcessing == 'Y'
    AND PURPOSE Processing
```

“Art. 8 Conditions applicable to child’s consent in relation to information society services”

1. “Where point (a) of Article 6(1) applies, in relation to the offer of information society services directly to a child, the processing of the personal data of a child shall be lawful where the child is at least 16 years old. Where the child is below the age of 16 years, such processing shall be lawful only if and to the extent that consent is given or authorised by the holder of parental responsibility over the child.

2. Member States may provide by law for a lower age for those purposes provided that such lower age is not below 13 years. The controller shall make reasonable efforts to verify in such cases that consent is given or authorised by the holder of parental responsibility over the child, taking into consideration available technology."

```
ALLOW SCHEMA PersonalData
    AND FILTER ConsentProcessing == 'Y'
    AND FILTER GuardianConsent == 'Y'
    AND FILTER Age < 16 AND FILTER Age >= 13
    AND PURPOSE Processing
```

"Art. 9 Processing of special categories of personal data"

"Processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation is allowed if:

a. the data subject has given explicit consent to the processing of those personal data for one or more specified purposes, except where Union or Member State law provide that the prohibition referred to in paragraph 1 may not be lifted by the data subject;"

```
ALLOW SCHEMA PDIdentifiable
    AND FILTER ConsentProcessingIdentifiable == 'Y'
    AND PURPOSE BusinessPurpose
```

b. "processing is necessary for the purposes of carrying out the obligations and exercising specific rights of the controller or of the data subject in the field of employment and social security and social protection law in so far as it is authorised by Union or Member State law or a collective agreement pursuant to Member State law providing for appropriate safeguards for the fundamental rights and the interests of the data subject;"

```
ALLOW SCHEMA PDIdentifiable
    AND FILTER ConsentProcessingIdentifiable == 'Y'
    AND PURPOSE Processing AND ( PURPOSE EmploymentPurpose OR PURPOSE
        SocialSecurity OR PURPOSE SocialProtection)
    AND ROLE Controller
```

d. "processing is carried out in the course of its legitimate activities with appropriate safeguards by a foundation, association or any other not-for-profit body with a political, philosophical, religious or trade union aim and on condition that the processing relates solely to the members or to former members of the body or to persons who have regular contact with it in connection with its purposes and that the personal data are not disclosed outside that body without the consent of the data subjects;"

```
ALLOW SCHEMA PDIdentifiable
    AND PURPOSE Processing
    AND (PURPOSE PoliticalPurpose OR PURPOSE PhilosophicalPurpose OR
        PURPOSE ReligiousPurpose OR PURPOSE TradeUnionPurpose)
    AND ROLE NonProfit
```

e. "processing relates to personal data which are manifestly made public by the data subject;"

```
ALLOW SCHEMA PublicPersonalData
    AND PURPOSE Processing
```

f. "processing is necessary for the establishment, exercise or defence of legal claims or whenever courts are acting in their judicial capacity;"

```
ALLOW SCHEMA PDIdentifiable
    AND PURPOSE Processing AND PURPOSE LegalPurpose
```

h. "processing is necessary for the purposes of preventive or occupational medicine, for the assessment of the working capacity of the employee, medical diagnosis, the provision of health or social care or treatment or the management of health or social care systems and services on the basis of Union or Member State law or pursuant to contract with a health professional and subject to the conditions and safeguards referred to in paragraph 3;"

```
ALLOW SCHEMA PDIdentifiable
    AND PURPOSE Processing AND PURPOSE MedicalPurpose
```

j "processing is necessary for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes in accordance with Article 89(1) based on Union or Member State law which shall be proportionate to the aim pursued, respect the essence of the right to data protection and provide for suitable and specific measures to safeguard the fundamental rights and the interests of the data subject."

```
ALLOW SCHEMA PDIdentifiable
    AND PURPOSE Processing AND (PURPOSE PublicInterest OR PURPOSE
        ScientificPurpose OR PURPOSE HistoricalResearchPurpose)
```

"Art. 10 Processing of personal data relating to criminal convictions and offences"

"Processing of personal data relating to criminal convictions and offences or related security measures based on Article 6(1) shall be carried out only under the control of official authority or when the processing is authorised by Union or Member State law providing for appropriate safeguards for the rights and freedoms of data subjects. Any comprehensive register of criminal convictions shall be kept only under the control of official authority."

```
ALLOW SCHEMA CriminalData
    AND ROLE OfficialAuthority
```

"Art. 14 Information to be provided where personal data have not been obtained from the data subject"

"Where personal data have not been obtained from the data subject, the controller shall provide the data subject with the following information:

a. the identity and the contact details of the controller and, where applicable, of the controller's representative;

b. the contact details of the data protection officer, where applicable."

```
ALLOW ROLE DataSubjectID
    AND SCHEMA ControllerID, ControllerRepresentative, ContactDataProtection
```

"Art. 15 Right of access by the data subject"

"The controller shall provide a copy of the personal data undergoing processing. For any further copies requested by the data subject, the controller may charge a reasonable fee based on administrative costs. Where the data subject makes the request by electronic means, and unless otherwise requested by the data subject, the information shall be provided in a commonly used electronic form."

```
ALLOW (ROLE Controller OR ROLE Processor)
    AND SCHEMA PersonalData
    AND PURPOSE Processing
    AND FILTER RequestProcessing == 'Y'
```

“Art. 16 Right to rectification”

“The data subject shall have the right to obtain from the controller without undue delay the rectification of inaccurate personal data concerning him or her. Taking into account the purposes of the processing, the data subject shall have the right to have incomplete personal data completed, including by means of providing a supplementary statement.”

```
ALLOW ROLE DataSubjectID
    AND SCHEMA PersonalData
    AND FILTER RequestRectification == 'Y'
    AND FILTER DataSubjectID == 'n_data_subject'
```

“Art. 17 Right to erasure (“right to be forgotten”)”

“The data subject shall have the right to obtain from the controller the erasure of personal data concerning him or her without undue delay and the controller shall have the obligation to erase personal data without undue delay where one of the following grounds applies:

a. the personal data are no longer necessary in relation to the purposes for which they were collected or otherwise processed;

b. the data subject withdraws consent on which the processing is based according to point (a) of Article 6(1), or point (a) of Article 9(2), and where there is no other legal ground for the processing.”

```
ALLOW (ROLE Controller OR ROLE Processor)
    AND FILTER DataSubjectID == 'n_data_subject' AND (FILTER RequestDeletion
        == 'Y' OR FILTER ConsentProcessing == 'N')
    AND PURPOSE DeletionData
```

“Art. 20 Right to data portability”

“The data subject shall have the right to receive the personal data concerning him or her, which he or she has provided to a controller, in a structured, commonly used and machine-readable format and have the right to transmit those data to another controller without hindrance from the controller to which the personal data have been provided.”

```
ALLOW ROLE DataSubjectID
    AND SCHEMA PersonalData
    AND FILTER RequestDisclosure == 'Y' AND FILTER DataSubjectID ==
        'n_data_subject'
```

“Art. 21 Right to object”

“Where the data subject objects to processing for direct marketing purposes, the personal data shall no longer be processed for such purposes.”

```
ALLOW (ROLE Controller OR ROLE Processor)
    AND SCHEMA PersonalData
    AND (PURPOSE Processing AND PURPOSE DirectMarketingPurpose)
    AND FILTER ConsentProcessing == 'Y' AND FILTER ConsentDirectMarketing ==
        'Y'
```

"Art. 25 Data protection by design and by default"

1. "Taking into account the state of the art, the cost of implementation and the nature, scope, context and purposes of processing as well as the risks of varying likelihood and severity for rights and freedoms of natural persons posed by the processing, the controller shall, both at the time of the determination of the means for processing and at the time of the processing itself, implement appropriate technical and organisational measures, such as pseudonymisation, which are designed to implement data-protection principles, such as data minimisation, in an effective manner and to integrate the necessary safeguards into the processing in order to meet the requirements of this Regulation and protect the rights of data subjects."

```
ALLOW ROLE Controller
    AND SCHEMA PersonalData
    AND (PURPOSE Retention OR PURPOSE BusinessPurpose OR PURPOSE Sell
        OR PURPOSE Share OR PURPOSE Processing)
    AND PRIVACY Anonymization
```

2. "The controller shall implement appropriate technical and organisational measures for ensuring that, by default, only personal data which are necessary for each specific purpose of the processing are processed. 2That obligation applies to the amount of personal data collected, the extent of their processing, the period of their storage and their accessibility. 3In particular, such measures shall ensure that by default personal data are not made accessible without the individual's intervention to an indefinite number of natural persons."

```
ALLOW ROLE Processor
    AND SCHEMA PersonalData
    AND PURPOSE Processing
    AND PRIVACY Anonymization
```

"Art. 39 Tasks of the data protection officer"

"The data protection officer shall have at the following task: to monitor compliance with this Regulation, with other Union or Member State data protection provisions and with the policies of the controller or processor in relation to the protection of personal data, including the assignment of responsibilities, awareness-raising and training of staff involved in processing operations, and the related audits."

```
ALLOW ROLE DataProtectionOfficer
    AND SCHEMA PersonalData
    AND PRIVACY Anonymization
    AND PURPOSE MonitorProtection
```

"Art. 44 General principle for transfers"

"Any transfer of personal data which are undergoing processing or are intended for processing after transfer to a third country or to an international organisation shall take place only if, subject to the other provisions of this Regulation, the conditions laid down in this Chapter are complied with by the controller and processor, including for onward transfers of personal data from the third country or an international organisation to another third country or to another international organisation. All provisions in this Chapter shall be applied in order to ensure that the level of protection of natural persons guaranteed by this Regulation is not undermined."

"Art. 46 Transfers subject to appropriate safeguards"

"In the absence of a decision pursuant to Article 45(3), a controller or processor may transfer personal data to a third country or an international organisation only if the controller or processor

has provided appropriate safeguards, and on condition that enforceable data subject rights and effective legal remedies for data subjects are available."

```
ALLOW (ROLE ThirdCountry OR ROLE InternationalOrg)
    AND SCHEMA PersonalData
    AND PURPOSE Transfer
    AND FILTER PRIVACY Anonymization
```

"Art. 49 Derogations for specific situations"

"In the absence of an adequacy decision pursuant to Article 45(3), or of appropriate safeguards pursuant to Article 46, including binding corporate rules, a transfer or a set of transfers of personal data to a third country or an international organisation shall take place only on one of the following conditions:

a. the data subject has explicitly consented to the proposed transfer, after having been informed of the possible risks of such transfers for the data subject due to the absence of an adequacy decision and appropriate safeguards;

b. the transfer is necessary for the performance of a contract between the data subject and the controller or the implementation of pre-contractual measures taken at the data subject's request;

c. the transfer is necessary for the conclusion or performance of a contract concluded in the interest of the data subject between the controller and another natural or legal person;

d. the transfer is necessary for important reasons of public interest;

e. the transfer is necessary for the establishment, exercise or defence of legal claim."

```
ALLOW (ROLE ThirdCountry OR ROLE InternationalOrg)
    AND SCHEMA PersonalData
    AND PURPOSE Transfer
    AND FILTER ConsentTransfer == 'Y'
    #1.b-c
    (OR PURPOSE BusinessPurpose
    #1.d
    OR PURPOSE PublicInterest
    #1.e
    OR PURPOSE LegalPurpose)
```

"Art. 58 Powers"

"Each supervisory authority shall have all of the following investigative powers: to obtain, from the controller and the processor, access to all personal data and to all information necessary for the performance of its tasks."

```
ALLOW ROLE SupervisoryAuthority
    SCHEMA PersonalData
    PURPOSE SupervisionPurpose
```

4.2 User manual

As previously stated PrivAnalyzer is a software system, implemented for a previous research work [1]. Starting from the source code in GitHub [1], we followed the instructions from "READ.ME"

```
sudo apt install python3.6
sudo apt install python3.6-venv
```

Figure 4.1. Prerequisites installation.

for the prerequisites. The analyser has been tested in Ubuntu 16.04 system. It is necessary to install python3.6 and python3.6-venv using the following commands.

The original code doesn't include all the examples and tests we run to obtain the following results. It is necessary to download the source code we modified with all the new information [26], by running:

```
git clone https://github.com/sofialucca/thesisResearch.git
```

Figure 4.2. Download source code.

It is also possible to directly extract everything from the source code contained in the .zip file, instead of downloading from the GitHub repository.

It is then important to enter the root directory of the repository, create and activate a python3.6 virtual environment, install the required python packages and set environment variables by running:

```
chmod u+x ./setup.sh
./setup.sh
```

Figure 4.3. Running setup.

The source code implements three main elements:

1. a policy parser to translate LEGALEASE policy strings into a Python object;
2. a set of functions used for data analysis and their possible privacy effects;
3. a static analyser that checks whether a Python program satisfies a LEGALEASE policy.

To test the policy parser, run

```
python path-to-repo/src/parser/policy_parser.py
```

Figure 4.4. Policy parser code run.

and then it will be required to insert a valid policy string in LEGALEASE (e.g. "ALLOW ROLE Oncologist AND SCHEMA age, condition AND PRIVACY DP(1.0,1e-5) AND FILTER age ¿ 18 AND REDACT zip(2:) AND PURPOSE PublicInterest"). The program will output the corresponding Python object (e.g. "[[role: Oncologist, 'AND', [schema: ['age', 'condition'],

'AND', [privacy: DP (1.0, 1e-05), 'AND', [filter: age [e19, einf], 'AND', [redact: zip(2:None), 'AND', purpose: PublicInterest]]]]]]").

It is possible to also convert a policy into its disjunctive normal form (DNF), so the normalisation of the logical formula into Boolean mathematics (considering the previous example the result would be "ConjunctClause(role: Oncologist, schema: ['age', 'condition'], privacy: DP (1.0, 1e-05), filter: age [e19, einf], redact: zip(2:None), purpose: PublicInterest) "). The user must run the following command and then input the policy string in LEGALEASE format:

```
python path-to-repo/src/parser/policy_tree.py
```

Figure 4.5. Policy tree code run.

The last element, the analyser, is the one used for our research. It access and uses also the other elements cited and described before. The original source code already provided some examples which used and implemented the developed functions and simple policy clauses. On top of those examples, we added the benchmark systems of LIBRETAXI, SELFMAILBOT and TRACCAR, and we introduced a total of 33 new files to analyse. To perform the analysis of a single file the command to be run is:

```
python path-to-repo/src/analyze.py
```

Figure 4.6. Analyser code run.

4.2.1 The flags

The main command for the analyser (Figure 4.6) must be followed by a set of flags to specify the file to analyse and the needed input data. Since the files analysed can represent the translation of functions with specific input arguments, we introduced new flags to pass for the analyses, that would behave as those input arguments. The flags that were used for all the analyses were:

example_id is a mandatory flag that specifies, with an integer the file to be analysed, we will explain later the value it can have.

lat and lon are strings indicating the location in latitude and longitude. It is a flag used by all three software systems.

user_id is used to pass the identifier of the user when needed for information retrieval by any of the systems. We used a default value of *n_consumer*.

LIBRETAXI required to introduce a large number of flags:

post_report is used to analyse the callback function, which is called when an action has been performed on a post. It can have as value *REPORT_POST* or *SHADOW_BAN*.

consent_use is a string indicating the consent given by the user over the use of its data. It has a default value of 'N' and is passed when analysing the saving of the user information, with all the other and following consents.

consent_share is the user's consent for the sharing of his data. It has a default value of 'N'.

consent_sell is the user's consent for the selling of his data. It has a default value of 'N'.

consent_collection indicates the user's consent for the collection of the data. Its default value is 'N'.

consent_retention used to pass the consent of the user for the retention of the data. It is a string with a default value of 'N'.

guardian_consent states the consent by a guardian for the managing of an underage user's personal information. It is managed as the previously listed consents, with a default value of 'N'.

request_deletion indicates if the user requested the deletion of his data. It is set as 'N', in case no value is passed.

request_disclosure is the flag for the request of the user to obtain access to his data, which by default is set to 'N'.

request_inaccurate states if the data currently stored is incorrect and eventually needs to be modified or deleted. It is set as 'N' by default.

limit_use indicates if the user requested the limitation of usage of sensitive personal information, to guarantee a higher level of security. It is set to 'N' as the default value.

age is an integer indicating the age of the user. It has a default value of 16.

menu_id specifies the current state of the user. It is a piece of information used to know what must be displayed on the screen of the user. It is a string that can be: *Menu_Init*, *Menu_AskLocation*, *Menu_Feed* or *Menu_Post*.

username specifies the username of a user. It is a string passed when checking for the saving of users or posts.

first_name and last_name are the name information that usually LibreTaxi retrieves from Telegram.

language_code specifies in a string the language for the communication used by the user.

report_count is an integer with the number of times a user has been reported. It has a default value of 0.

shadow_banned indicates if the user has been shadow banned, by default it is 'N'.

text is the string containing the text sent by the user to the application.

post_id is the analogous for the post of *user_id*. It is used when retrieving information from a post.

locales , similarly to *language_code*, indicate the language set for the communication. It is used to know which file to access for the default messages.

For Selfmailbot it was necessary to introduce flags representing the information that could be sent by the program. Other than the flag *text* previously defined, there are:

attachment is an integer that defines the possible attachment for the email to be sent.

attachment_name is the name of the file that will be attached to the email, so it is a string value.

email is a string with the email address which will receive all the messages.

subject indicates the string to be set as the subject for the email.

link and key are two strings used for the confirmation of the email address. The first one is sent to the user and the second is used to check if the link has been validated.

Lastly Traccar introduced:

clazz used to specify which information we want to access from the database. It has a default value of *Device*, but can also be set as *Position* or *User*

columns is a list of columns that must be included or excluded -depending on the file that needs to be analysed - from the database extraction.

deviceid is a string for the unique identifier of the device, by default for our analysis we set it as *n_default*.

new_val is a string passed when analysing the setter functions and indicates the new value that needs to be set for the specified column.

4.2.2 *example_id* values

As introduced in the previous subsection, the flag *example_id* is mandatory because it indicates the file we want to analyse. It can have different values, which are all integers:

- 0, 4, 5, 6, 7, 23 are the examples from the source code, so not used by our research work;
- from 25 to 40 there are the files for the LibreTaxi analysis:
  - 25 analyses the all program and requires as a flag only *user_id*.
  - 26 for the saving of the user information. It needs to receive all the information about the user with the flags: *user_id*, *menu_id*, *lon*, *lat*, *first_name*, *last_name*, *language_code*, *report_cnt*, *shadow_banned*.
  - 28 analyses the callback function, used when an action is performed on a post. We passed as input flags "–post_report SHADOW_BAN –user_id n_consumer –lon 1213.645 –lat 2345.009 –report_cnt 4".
  - 30 analyses the saving of a post. As flags it receives: *user_id*, *text*, *lon*, *lat*, *report_cnt*.
  - 31 checks the function which manages the finding which retrieves the user in a radius of 25 kilometres (15 miles). Its flags are : *lon*, *lat*.
  - 32 analyses the retrieval of recent posts from a user in the last five minutes. The necessary flag is *user_id*.
  - 33 checks for compliance in the menu which asks for the location. It requires only the *user_id* flag as input.
  - 34 analyse the function that manages the display of the main feed in LibreTaxi. It needs many input flags: *text*, *user_id*, *consent_use*, *consent_share*, *consent_sell*, *consent_collection*, *consent_retention*, *guardian_consent*, *request_deletion*, *request_disclosure*, *request_inaccurate*, *limit_use*, *menu_id*, *username*, *first_name*, *last_name*, *lon*, *lat*, *language_code*, *report_cnt*, *shadow_banned*.
  - 35 checks the initial menu, which welcomes the new user. It is similar to the previous situation and will require all the data of a user: *user_id*, *consent_use*, *consent_share*, *consent_sell*, *consent_collection*, *consent_retention*, *guardian_consent*, *request_deletion*, *request_disclosure*, *request_inaccurate*, *limit_use*, *menu_id*, *username*, *first_name*, *last_name*, *lon*, *lat*, *language_code*, *report_cnt*, *shadow_banned*.
  - 37 checks the initial function called for managing the receiving of a post from the user. It receives *text*,*user_id*, *consent_use*, *consent_share*, *consent_sell*, *consent_collection*, *consent_retention*, *guardian_consent*, *request_deletion*, *request_disclosure*, *request_inaccurate*, *limit_use*, *menu_id*, *username*, *first_name*, *last_name*, *lon*, *lat*, *language_code*, *report_cnt*, *shadow_banned*.

  - 36 is the first function called when the program is started. It manages the switching between the various possible situations (e.g. asking for location, initial menu, feed menu, post menu). It needs the *text* and all the previously listed user flags: *user_id*, *consent_use*, *consent_share*, *consent_sell*, *consent_collection*, *consent_retention*, *guardian_consent*, *request_deletion*, *request_disclosure*, *request_inaccurate*, *limit_use*, *menu_id*, *username*, *first_name*, *last_name*, *lon*, *lat*, *language_code*, *report_cnt*, *shadow_banned*.
  - 38 checks for the finding of a post in the database. It needs simply *post_id*.
  - 39 analyses the finding of a user in the storage. It requires *user_id*.
  - 40 recalls from the database the currently required language for the communication.

- from 50 to 63 there are all the files used for the SELFMAILBOT analysis:
  - 50 checks if the confirmation of the email address has been performed. It needs to receive *user_id* and *key*.
  - 51 is called to resend a confirmation link for the email address. It needs *user_id* and *email*.
  - 52 analyses the resetting of the email address, so it checks the work done over the removal of it from the storage. It only needs *user_id*
  - 53 checks for the sending of the confirmation email; it requires *email*.
  - 54 analyses for the sending of a picture via mail. It needs *text*, *user_id*, *attachment*.
  - 55 checks for the sending of a text through email, given *text*, *user_id*.
  - 57 is an implementation of celery library for the sending of the confirmation mail. It requires *user_id*, *email*.
  - 58 is a celery implementation for the sending of files; it requires *user_id*, *attachment*.
  - 59 implements, starting from the celery library, the sending of a text through email. It needs to have set the flags *user_id*, *text*.
  - 60 and 61 are the files which manage the creation of the text to be inserted in the email when it is needed to be sent. They require *attachment*, *attachment_name*, *email*, *subject*, *text*.
  - 61 retrieves from a database the user given *link*.
  - 62 checks for compliance when trying to retrieve a user given *user_id*.

- between 70 and 76 there are the flag's values for TRACCAR :
  - 70 checks for the database access and retrieval of data based on *clazz* value.
  - 71 analyses the access to the database based on the *clazz* and *columns* that must be discarded.
  - 72 is the opposite situation of 71, where the *clazz* is specified in combination with the list of *columns* to be included.
  - 73 checks for the retrieval of a device's latest position stored, given *deviceid*
  - 74 analyses the getter for the user information, by giving *user_id* and a single value for the *columns*'s flag.
  - 75 checks for compliance in the user setter. It receives *user_id*, a single value *columns* and a *new_val*.
  - 76 analyses the sending of an email to a user, by receiving only the *user_id*.

4.2.3 Input files

In the source code, all the input files are already present. In the directory *path-to-repo/src/examples/data* there is a set of folders for each system analysed. Each folder contains the other two input files used by PRIVANALYZER:

meta.txt which defines the information stored in the database, by listing its columns;

policy.txt lists all the policies in LEGALEASE format.

Each pair of these files represent a database that must be protected. For SELFMAILBOT the database and storage were unique, having only two files in its folder. A different situation is associated with LIBRETAXI, which has two separate tables for posts and users, and TRACCAR, which collects separately the data for devices, positions and users. These two cases required to introduce separate sub-folders for each of the database tables, which then contained their own copy of meta.txt and policy.txt. Their content varies, due to the difference in information they store and the different levels of privacy the data require.

Since our research was based on two different sets of policies, inside each folder there are some extra files(e.g. *policy_cpra*, *policy_gdpr*, *meta_cpra*, *meta_gdpr*), containing the respective content for both cases of the research. When an analysis must be performed the user must copy the text of the relative policy that must be checked. The differences are based on the various definitions explained in subsection 3.1.2. It is important to have both meta and policy file associated with the same set of policies. In case this doesn't happen an error occurs, causing to obtain a different result than what expected.

Chapter 5

Results

In this chapter, I will define and describe the results obtained from the analyses of the previously introduced codes. All the codes I previously mentioned had been analysed twice with respect to two different lists of policies. The first list is the translations of CPRA while the second one contains the translations of GDPR. It is important to state that for all these analysis I used attribute values for ROLE, FILTER, PRIVACY and PURPOSE introduced in section 3.1.2. For SCHEMA I used only a subgroup of *PersonalData* and *PersonalInformation*. Between GDPR and CPRA there are some shared SCHEMA attributes which have different names; an example is the unique online identifier: *ConsumerID* for CPRA and *DataSubjectID* for GDPR.

5.1 LibreTaxi results

LibreTaxi does not satisfy either of the two laws concerning privacy and usage of users' data.

For CPRA we started by checking and analysing the main code and program, which gave back a negative response: unsatisfied policies (Figure 5.1). This type of result means that the code must be changed because it manages incorrectly the data. There are no possible ways to fix it, by adding actions, it is needed to remove some of them.

```
ConjunctClause(UNSAT)
```

Figure 5.1. LibreTaxi for CPRA: unsatisfied residual policies.

At this point, we moved on to the single files and single functions to try and identify what were the main problems and policies that this code did not follow. This might give a better and more clear idea of the reasons behind the previous result. The most common problems were due to a lack of checking for the consents of the user, any action taken upon the data was never associated with a user's consent. Many websites allow users to give consent to the use of data right when this data is given, but LibreTaxi never asks. First of all, location is a piece of information vastly used, it is initially given by the user when signing up, allowing it to be collected and stored.

The problem arises when, after a new post the program needs to notify the users that are nearby and might be interested. Location is retrieved from the database and used for *BusinessPurpose*, but the user should need to allow it by denying the limit use of it ($LimitUse == 'N'$) (Figure 5.2). CPRA in *art. 1798.121* states that: "A consumer shall have the right, at any time, to direct a business that collects sensitive personal information about the consumer to limit its use of the consumer's sensitive personal information to that use which is necessary to perform the services or provide the goods reasonably expected by an average consumer who requests those goods or services".

```
ConjunctClause(role: Business, schema: ['Longitude', 'Latitude'], purpose:
    BusinessPurpose, filter: LimitUse [eN, eN])
```

Figure 5.2. LibreTaxi for CPRA: residual policies for location.

All the personal data belonging to the user (*ConsumerID*, *Location*, *Username*, *LastName* and *FirstName*) are collected when he interacts with the bot for the first time and then eventually updated and retrieved. The law requires two main important points: the program must protect the data from any possible attack or access by someone not authorised and the user must grant consent for any use or manipulation of his data. With respect to protection, it is required a minimum level of privacy for the information that is collected and used (*privacy: DP (1.0, 1e-05)*). As stated in *art. 1798.100.e*: "A business that collects a consumer's personal information shall implement reasonable security procedures and practices appropriate to the nature of the personal information to protect the personal information from unauthorised or illegal access, destruction, use, modification, or disclosure in accordance with Section 1798.81.5". The residual policy obtained (Figure 5.3) shows that this requirement is not met. Other, than a lack of protection, this policy is another example of a lack of checking for user's consent (*ConsentRetention == 'Y'*).

```
ConjunctClause(role: Business, schema: ['ConsumerID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude'], privacy: DP (1.0,
    1e-05), filter: ConsentRetention [eY, eY]),
```

Figure 5.3. LibreTaxi for CPRA: residual policies for personal information retention.

CPRA requires in *art. 1798.145.6*[1] that any manipulation of the user's personal information is done with a level of privacy *Aggregation* or *Anonymization*, but LibreTaxi does not grant it, for either *Retention* or *BusinessPurpose* (Figure 5.4).

Secondly, the user needs to know and give his consent for any possible data usage from any ROLE. Specifically LibreTaxi manages mainly for *BusinessPurpose*: so the residual policies show that for this kind of usage, it is required to check for the consent of use, retention and collection (Figure 5.5). It does not mean, that the user is against the use of his data, but it is not known what the user wants since the program never checks. *Art. 1798.100.c* states: "A business' collection, use, retention, and sharing of a consumer's personal information shall be reasonably necessary and proportionate to achieve the purposes for which the personal information was collected or processed, or for another disclosed purpose that is compatible with the context in which the personal information was collected, and not further processed in a manner that is incompatible with those purposes".

At this point, we moved on to the files that were defining the functions which access and use the data about the posts. The results are very similar to the previously introduced ones, but the small difference was the addition of *MessageText* as personal information. This information has the same level of sensitivity as *Latitude* and *Longitude*, so it is considered sensitive personal information. LibreTaxi does not take into consideration this and the residual policies show a lack of any form of checking also for this information. Figure 5.6 shows the result obtained regarding lack of checking for the *LimitUse* attribute value: it is required to have this set to 'N', as stated by *art. 1798.121.*

Since *MessageText* is personal information, it requires the same levels of protection and checking that we previously stated for *Username*, *FirstName*, *LastName*, *Latitude* and *Longitude*. So

[1] "The obligations imposed on businesses by this title shall not restrict a business's ability to collect, use, retain, sell, share, or disclose consumers' personal information that is deidentified or aggregate consumer information"

```
ConjunctClause(role: Business, schema: ['ConsumerID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude'], purpose: Retention,
    privacy: Anonymization),
ConjunctClause(role: Business, schema: ['ConsumerID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude'], purpose: Retention,
    privacy: Aggregation),
ConjunctClause(role: Business, schema: ['ConsumerID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude'], purpose:
    BusinessPurpose, privacy: Anonymization),
ConjunctClause(role: Business, schema: ['ConsumerID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude'], purpose:
    BusinessPurpose, privacy: Aggregation)
```

Figure 5.4. LibreTaxi for CPRA: residual policies for personal information.

```
ConjunctClause(role: Business, schema: ['ConsumerID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude'], filter: ConsentUse
    [eY, eY], purpose: BusinessPurpose),
ConjunctClause(role: Business, schema: ['ConsumerID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude'], filter:
    ConsentCollection [eY, eY], purpose: BusinessPurpose),
ConjunctClause(role: Business, schema: ['ConsumerID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude'], filter:
    ConsentRetention [eY, eY], purpose: BusinessPurpose)
```

Figure 5.5. LibreTaxi for CPRA: residual policies for consent.

```
ConjunctClause(role: Business, schema: ['Longitude', 'Latitude',
    'MessageText'], purpose: BusinessPurpose, filter: LimitUse [eN, eN])
```

Figure 5.6. LibreTaxi for CPRA: residual policies for all the sensitive personal information.

the residual policies obtained when performing the analysis of those files are analogous to the previous ones but they list an additional attribute for SCHEMA (Figure 5.7).

Any kind of major manipulation of the data, such as modification and deletion, must be requested by the user(Figure 5.8). It is a condition defined by *art. 1798.106*: "A consumer shall have the right to request a business that maintains inaccurate personal information about the consumer to correct that inaccurate personal information, taking into account the nature of the personal information and the purposes of the processing of the personal information.

A business that collects personal information about consumers shall disclose, pursuant to Section 1798.130, the consumer's right to request correction of inaccurate personal information.

A business that receives a verifiable consumer request to correct inaccurate personal information shall use commercially reasonable efforts to correct the inaccurate personal information as directed by the consumer The analyses report how this does not happen since the data is modified without the user's consent, meanwhile the deletion cannot be managed or controlled by the user". In fact, the data stored by LibreTaxi is not even accessible by the user in any possible way, causing an impossibility to allow him to know if the information is wrong and needs modification or if he wants to delete it.

```
ConjunctClause(role: Business, schema: ['ConsumerID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude', 'MessageText'],
    purpose: Retention, privacy: Anonymization),
ConjunctClause(role: Business, schema: ['ConsumerID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude', 'MessageText'],
    purpose: Retention, privacy: Aggregation),
ConjunctClause(role: Business, schema: ['ConsumerID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude', 'MessageText'],
    purpose: BusinessPurpose, privacy: Anonymization),
ConjunctClause(role: Business, schema: ['ConsumerID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude', 'MessageText'],
    purpose: BusinessPurpose, privacy: Aggregation), ConjunctClause(role:
    Business, schema: ['ConsumerID', 'Username', 'FirstName', 'LastName',
    'Longitude', 'Latitude', 'MessageText'], privacy: DP (1.0, 1e-05),
    filter: ConsentRetention [eY, eY])
```

Figure 5.7. LibreTaxi for CPRA: residual policies for all the personal information.

```
ConjunctClause(role: Business, filter: RequestDeletion [eY, eY], purpose:
    DeletionData),
ConjunctClause(role: Business, schema: ['ConsumerID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude', 'MessageText'], filter:
    RequestInaccurate [eY, eY], purpose: ModificationData)
```

Figure 5.8. LibreTaxi for CPRA: residual policies for modification and deletion.

Art. 1798.110 states that "A consumer shall have the right to request that a business that collects personal information about the consumer disclose to the consumer the following: the specific pieces of personal information it has collected about that consumer". User has the right to access with a request his information from the business that is using them, but as shown with the residual policy in Figure 5.9 with this code it does not happen and is not possible.

```
ConjunctClause(role: ConsumerID, schema: ['ConsumerID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude', 'MessageText'], filter:
    RequestDisclosure [eY, eY])
```

Figure 5.9. LibreTaxi for CPRA: residual policy for access to data.

The second part of the analysis of LibreTaxi was performed with a different set of input privacy clauses. These clauses were the translation of GDPR articles. The results and the residual policies obtained were different due to the different specifications and translations of the law into policy clauses. In fact, the residual policies obtained are shorter and not as specific as the previous ones. The first main difference is associated with the analysis of the main code, which in this case gives a set of residual policies and not only a negative result like the one from Figure 5.1.

We are going to split and consider the results separately to allow an easier description of the result from the main code. First of all, with GDPR there is no sensitive personal data, so *Username*, *LastName*, *FirstName*, *Latitude*, *Longitude* and *DataSubjectID* are all personal data managed in the same way. They must be all protected and GDPR requires a specific type of

minimum privacy: de-identification. This concept was introduced with the law itself (*Art. 25.1*[2]) to guarantee that data is manipulated in a way that does not allow to recognise the user to which it belongs to unless other information is given. Unfortunately, LibreTaxi does not satisfy this requirement, as shown by one of the residual policies obtained (Figure 5.10), that requires as form of privacy *Anonymization*.

```
ConjunctClause(role: Controller, schema: ['DataSubjectID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude'], privacy: Anonymization)
```

Figure 5.10. LibreTaxi for GDPR: residual policy for privacy of data.

Data must be accessed and managed with the user's consent, as stated by *Art. 5.d*: "Personal data shall be accurate and, where necessary, kept up to date; every reasonable step must be taken to ensure that personal data that are inaccurate, having regard to the purposes for which they are processed, are erased or rectified without delay ("accuracy")". In case of deletion or modification the data must be incorrect, so requires *CorrectInformation == 'N'*; *PrivAnalyzer* works in a way that allows to know if this information has been checked. When a residual policy is obtained and still has the previous filter listed it means that this information hasn't been checked (Figure 5.11).

```
ConjunctClause(role: Controller, schema: ['DataSubjectID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude'], filter:
    CorrectInformation [eN, eN], purpose: DeleteData),
ConjunctClause(role: Controller, schema: ['DataSubjectID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude'], filter:
    CorrectInformation [eN, eN], purpose: ModificationData)
```

Figure 5.11. LibreTaxi for GDPR: residual policies for modification and deletion.

As discovered from the previous analysis, the user can't access his data, Figure 5.12 confirms this negligence. The first residual policy (*line 1*) is analogous to the CPRA, so the user should be able to obtain his data after a specific request (*RequestRectification == 'Y'*)[3]. The second line defines a policy associated with the possibility for the data subject to obtain the data that has been processed; this requires a separate request, hence a different filter attribute (*RequestProcessing == 'Y'*)[4].

Lastly, the GDPR analysis brought up another problem, related to the processing of the data. *Art. 6.a* states: "Processing shall be lawful only if and to the extent that at least one of the

[2] "Taking into account the state of the art, the cost of implementation and the nature, scope, context and purposes of processing as well as the risks of varying likelihood and severity for rights and freedoms of natural persons posed by the processing, the controller shall, both at the time of the determination of the means for processing and at the time of the processing itself, implement appropriate technical and organisational measures, such as pseudonymisation, which are designed to implement data-protection principles, such as data minimisation, in an effective manner and to integrate the necessary safeguards into the processing in order to meet the requirements of this Regulation and protect the rights of data subjects."

[3] *Art. 16*: "The data subject shall have the right to obtain from the controller without undue delay the rectification of inaccurate personal data concerning him or her. Taking into account the purposes of the processing, the data subject shall have the right to have incomplete personal data completed, including by means of providing a supplementary statement".

[4] *Art. 15*: "The controller shall provide a copy of the personal data undergoing processing. For any further copies requested by the data subject, the controller may charge a reasonable fee based on administrative costs. Where the data subject makes the request by electronic means, and unless otherwise requested by the data subject, the information shall be provided in a commonly used electronic form".

```
1  ConjunctClause(role: DataSubjectID, schema: ['DataSubjectID', 'Username',
       'FirstName', 'LastName', 'Longitude', 'Latitude'], filter:
       RequestRectification [eY, eY]),
2  ConjunctClause(role: DataSubjectID, schema: ['DataSubjectID', 'Username',
       'FirstName', 'LastName', 'Longitude', 'Latitude'], purpose: Processing,
       filter: RequestProcessing [eY, eY])
```

Figure 5.12. LibreTaxi for GDPR: residual policies for access to data.

following applies the data subject has given consent to the processing of his or her data for one or more specific purposes". Other than the consent (*ConsentProcessing == 'Y'*) the age of the user must be above 16 to allow the *Controller* to process this data (Figure 5.13).

```
ConjunctClause(role: Controller, schema: ['DataSubjectID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude'], filter:
    ConsentProcessing [eY, eY], filter: Age [e16, einf], purpose: Processing)
```

Figure 5.13. LibreTaxi for GDPR: residual policies for data process.

We then proceeded with an analysis of the single files and functions. We mainly encountered the same residual policies, except for when we checked the files that directly managed posts and their data. As already explained before, posts' data has also *MessageText* as information, causing to have residual policies with an extra SCHEMA argument. Their meaning is the same as the ones explained before and without this new information (Figure 5.14).

```
ConjunctClause(schema: ['DataSubjectID', 'Username', 'FirstName', 'LastName',
    'Longitude', 'Latitude', 'MessageText'], filter: CorrectInformation [eN,
    eN], purpose: DeleteData),
ConjunctClause(schema: ['DataSubjectID', 'Username', 'FirstName', 'LastName',
    'Longitude', 'Latitude', 'MessageText'], filter: CorrectInformation [eN,
    eN], purpose: ModificationData),
ConjunctClause(schema: ['DataSubjectID', 'Username', 'FirstName', 'LastName',
    'Longitude', 'Latitude', 'MessageText'], privacy: Anonymization),
ConjunctClause(schema: ['DataSubjectID', 'Username', 'FirstName', 'LastName',
    'Longitude', 'Latitude', 'MessageText'], filter: ConsentProcessing [eY,
    eY], filter: Age [e16, einf], purpose: Processing),
ConjunctClause(schema: ['DataSubjectID', 'Username', 'FirstName', 'LastName',
    'Longitude', 'Latitude', 'MessageText'], filter: ConsentProcessing [eY,
    eY], filter: GuardianConsent [eY, eY], filter: Age [eninf, e15], filter:
    Age [e13, einf], purpose: Processing),
ConjunctClause(role: Controller, schema: ['DataSubjectID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude', 'MessageText'],
    purpose: Processing, filter: RequestProcessing [eY, eY]),
ConjunctClause(role: Processor, schema: ['DataSubjectID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude', 'MessageText'],
    purpose: Processing, filter: RequestProcessing [eY, eY]),
ConjunctClause(role: DataSubjectID, schema: ['DataSubjectID', 'Username',
    'FirstName', 'LastName', 'Longitude', 'Latitude', 'MessageText'], filter:
    RequestRectification [eY, eY])
```

Figure 5.14. LibreTaxi for GDPR: residual policies with *MessageText*.

5.2 Selfmailbot results

Selfmailbot is an example of a lack of satisfaction for both laws. Even though we were only able to analyse and check the single files and functions, the results were not positive. The results obtained were very similar to the previous ones obtained with LibreTaxi.

Starting with CPRA, the data used were all *PersonalInformation*, so there was no sensitive information that need extra privacy converge. Except for the email address, all the other data were retrieved without asking the user for their usage. Still, also the email address, even if confirmed and written by the user, the bot never actually requests explicit consent for usage, collection and retention. The user directly interacts with the bot, by giving an email and then sending the information that he wants to share. The user never allows or denies Selfmailbot to manage his data (Figure 5.15). It is still his right to do so, as defined by *art. 1798.100.c*: "A business' collection, use, retention, and sharing of a consumer's personal information shall be reasonably necessary and proportionate to achieve the purposes for which the personal information was collected or processed, or for another disclosed purpose that is compatible with the context in which the personal information was collected, and not further processed in a manner that is incompatible with those purposes".

The information that is stored and defines the users must be accessed and saved with a minimum level of privacy: *Aggregation* or *Anonymization* (*art. 1798.145.6*). This requirement is essential to guarantee eventual protection against attacks and not satisfied access. This code does not protect the data in any way. It simply stores it in a database, without any protection and the same approach is done for eventual access. CPRA has this requirement in case of usage for *BusinessPurpose* and *Retention* (Figure 5.16).

Other than the *Business* itself, access to the data must be allowed and granted for the consumer when he requests it (*art. 1798.110*). CPRA allows access only for the data that belongs to the specific consumer (*ConsumerID == 'n_consumer'*). PrivAnalyzer yields a result that displays

```
ConjunctClause(role: Business, schema: ['ConsumerID', 'FullName', 'Username',
    'Email'], filter: ConsentUse [eY, eY], purpose: BusinessPurpose),
ConjunctClause(role: Business, schema: ['ConsumerID', 'FullName', 'Username',
    'Email'], filter: ConsentCollection [eY, eY], purpose: BusinessPurpose),
ConjunctClause(role: Business, schema: ['ConsumerID', 'FullName', 'Username',
    'Email'], filter: ConsentRetention [eY, eY], purpose: BusinessPurpose),
```

Figure 5.15. Selfmailbot for CPRA: residual policies for retention and usage.

```
ConjunctClause(role: Business, schema: ['ConsumerID', 'FullName', 'Username',
    'Email'], purpose: Retention, privacy: Anonymization),
ConjunctClause(role: Business, schema: ['ConsumerID', 'FullName', 'Username',
    'Email'], purpose: Retention, privacy: Aggregation),
ConjunctClause(role: Business, schema: ['ConsumerID', 'FullName', 'Username',
    'Email'], purpose: BusinessPurpose, privacy: Anonymization),
ConjunctClause(role: Business, schema: ['ConsumerID', 'FullName', 'Username',
    'Email'], purpose: BusinessPurpose, privacy: Aggregation)
```

Figure 5.16. Selfmailbot for CPRA: residual policies for retention and usage privacy.

how this does not happen, causing to have users who are in the dark with respect to the data used by this system (Figure 5.17).

```
ConjunctClause(role: ConsumerID, schema: ['ConsumerID', 'FullName',
    'Username', 'Email'], filter: RequestDisclosure [eY, eY], filter:
    ConsumerID [en_consumer, en_consumer])
```

Figure 5.17. Selfmailbot for CPRA: residual policies for access data.

The access to data for a user is also associated with an eventual check of this information and its correctness (*art. 1798.106*). When information is incorrect user should be able to modify them. This is not possible and does not happen. The only possible element that is modifiable is the email, that can be reset. The resetting is implemented correctly; it is a modification of the data performed after a request from the user. This approach is the one defined by CPRA for *ModificationData*, the problem that PrivAnalyzer encounters are that it is not possible to modify the other information (Figure 5.18).

```
ConjunctClause(role: Business, schema: ['ConsumerID', 'FullName',
    'Username'], purpose: ModificationData)
```

Figure 5.18. Selfmailbot for CPRA: residual policies for data modification.

We moved on with the analysis of Selfmailbot with respect to clauses from GDPR. We obtained a set and list of residual policies, meaning that the code should be modified to comply with and satisfy the law requirements. Also, this analysis enlightens the lack of privacy for the data. As for CPRA, also GDPR requires a minimum level of privacy (*Anonymization*) when these

data are processed, retention and used for a business purpose (*Art. 25.1*). Business purpose is a generic super-category for processing and retention, in fact, these actions are performed to do the work for the business. The residual policies are showing how the data is used without any protection (Figure 5.19).

```
ConjunctClause(role: Controller, schema: ['DataSubjectID', 'FullName',
    'Username', 'Email'], purpose: Retention, privacy: Anonymization),
ConjunctClause(role: Controller, schema: ['DataSubjectID', 'FullName',
    'Username', 'Email'], purpose: BusinessPurpose, privacy: Anonymization),
ConjunctClause(role: Controller, schema: ['DataSubjectID', 'FullName',
    'Username', 'Email'], purpose: Processing, privacy: Anonymization)
```

Figure 5.19. Selfmailbot for GDPR: residual policies for privacy.

GDPR requires that any user consents to the processing of his data, this consent must be given by the user if the user is older than 16 years old or by a guardian if he's between 1 and 16 years old (*Art. 6.a*). SELFMAILBOT does not ask for user consent, so, without any consent, it does not check if that consent is positive (*ConsentProcessing == 'Y'*), but also most importantly does not check for the user's age. Figure 5.20 are the residual policies regarding this issue, where processing is done without checking for the user's approval and for his age.

```
ConjunctClause(schema: ['DataSubjectID', 'FullName', 'Username', 'Email'],
    filter: ConsentProcessing [eY, eY], filter: Age [e16, einf], purpose:
    Processing),
ConjunctClause(schema: ['ConsumerID', 'FullName', 'Username', 'Email'],
    filter: ConsentProcessing [eY, eY], filter: ConsentGuardian [eY, eY],
    filter: Age [eninf, e15], filter: Age [e13, einf], purpose: Processing)
```

Figure 5.20. Selfmailbot for GDPR: residual policies for processing.

As seen before, no user can access data stored regarding himself. He should be able to do after a request, as stated in *Art. 20* (*RequestDisclosure == 'Y'*). For GDPR in *Art. 15* he should also be able to access the data that had been processed if there are any (*RequestProcessing == 'Y'*). No implementation of this process has been written, PRIVANALYZER yield a set of residual policies showing this lack (Figure 5.21).

```
ConjunctClause(role: DataSubjectID, schema: ['DataSubjectID', 'FullName',
    'Username', 'Email'], filter: RequestDisclosure [eY, eY]),
ConjunctClause(role: DataSubjectID, schema: ['ConsumerID', 'FullName',
    'Username', 'Email'], purpose: Processing, filter: RequestProcessing [eY,
    eY])
```

Figure 5.21. Selfmailbot for GDPR: residual policies for processed data.

The previously mention impossibility to access the data does not allow the user to know if the data is incorrect or needs modifications. *Art. 5.d* from GDPR states that *Controller* can modify the data if it is incorrect (*CorrectInformation == 'N'*, so it does not require an explicit request of the user. *Selfmailbot* allows user to modify and remove the email; the results of PRIVANALYZER show the impossibility to modify the other information (Figure 5.22).

```
ConjunctClause(schema: ['DataSubjectID', 'FullName', 'Username', 'Email'],
    purpose: DeleteData),
ConjunctClause(schema: ['DataSubjectID'], purpose: ModificationData),
ConjunctClause(schema: ['FullName'], purpose: ModificationData),
ConjunctClause(schema: ['Username'], purpose: ModificationData)
```

Figure 5.22. Selfmailbot for GDPR: residual policies for modification.

Lastly, other than modifying data it is necessary to manage the possibility to delete the data and information. *Art. 17* defines the *right to be forgotten*: "The data subject shall have the right to obtain from the controller the erasure of personal data concerning him or her without undue delay and the controller shall have the obligation to erase personal data without undue delay where one of the following grounds applies:

a. the personal data are no longer necessary in relation to the purposes for which they were collected or otherwise processed;

b. the data subject withdraws consent on which the processing is based according to point (a) of Article 6(1), or point (a) of Article 9(2), and where there is no other legal ground for the processing".

This means that the deletion is possible in case any of the following situations apply:

- Data is incorrect: *CorrectInformation = 'N'*;
- User request deletion of the data: *RequestDeletion == 'Y'*;
- User withdrew consent for the processing of his data: *ConsentProcessing == 'N'*.

The code never performs any data deletion or even checks for this information, so one of the last residual policies obtained is associated with this missing element (Figure 5.23).

```
ConjunctClause(schema: ['DataSubjectID', 'FullName', 'Username', 'Email'],
    purpose: DeletionData),
ConjunctClause(role: Controller, filter: RequestDeletion [eY, eY], purpose:
    DeletionData),
ConjunctClause(role: Controller, filter: RequestProcessing [eN, eN], purpose:
    DeletionData)
```

Figure 5.23. Selfmailbot for GDPR: residual policies for deletion.

5.3 Traccar results

Traccar has been implemented in a way that follows and satisfies some of the defined laws. It does not satisfy them completely but more than LibreTaxi and Selfmailbot, allowing to see possible differences in the results. The analyses have been done similarly to the previous cases: an analysis of the main code and then one for the various functions and single files. The functions which gave back the most residuals were the setters and getters for the information stored.

We started by considering CPRA policies. When registering a new device, the application asks for permission to access the location. It is the only consent asked by the system. In fact, there

are no requests for the usage of that information. The user consents only access to the device's GPS. Any use, processing or retention done after this consent is not allowed or controlled by the owner(Figure 5.24), differently from what *art. 1798.100.c* requires.

```
ConjunctClause(role: Business, schema: ['DeviceID', 'PositionID'], filter:
    ConsentUse [eY, eY], purpose: BusinessPurpose),
ConjunctClause(role: Business, schema: ['DeviceID', 'PositionID'], filter:
    ConsentCollection [eY, eY], purpose: BusinessPurpose),
ConjunctClause(role: Business, schema: ['DeviceID', 'PositionID'], filter:
    ConsentRetention [eY, eY], purpose: BusinessPurpose),
ConjunctClause(role: Business, schema: ['DeviceID', 'PositionID'], filter:
    ConsentShare [eY, eY], purpose: Share)
```

Figure 5.24. Traccar for CPRA: residual policies for consent for devices.

For each device the user controls eventual status, allowing to limit the use and access to the position. This is a requirement very important since the location is sensitive information that requires a higher level of protection. On the other hand, problems arise when discussing the level of privacy for information storage (*art. 1798.145.6*). In fact, no protection is guaranteed (Figure 5.25). The only security is due to the hashing of the password for the user credential. The password is not stored as it is but is hashed and stored with the *Salt* necessary to decrypt it.

```
ConjunctClause(role: Business, schema: ['ConsumerID', 'Name', 'Email',
    'HashedPassword', 'Salt', 'Latitude', 'Longitude'], purpose: Retention,
    privacy: Anonymization),
ConjunctClause(role: Business, schema: ['ConsumerID', 'Name', 'Email',
    'HashedPassword', 'Salt', 'Latitude', 'Longitude'], purpose: Retention,
    privacy: Aggregation),
ConjunctClause(role: Business, schema: ['PositionID', 'DeviceID', 'Latitude',
    'Longitude', 'Altitude', 'Address'], purpose: Retention, privacy:
    Anonymization),
ConjunctClause(role: Business, schema: ['PositionID', 'DeviceID', 'Latitude',
    'Longitude', 'Altitude', 'Address'], purpose: Retention, privacy:
    Aggregation),
ConjunctClause(role: Business, schema: ['DeviceID', 'PositionID'], purpose:
    Retention, privacy: Anonymization),
ConjunctClause(role: Business, schema: ['DeviceID', 'PositionID'], purpose:
    Retention, privacy: Aggregation)
```

Figure 5.25. Traccar for CPRA: residual policies for privacy retention.

With respect to privacy, the law requires also the previously cited privacy when managing and using the data and information for *BusinessPurpose*. The residual policies obtained with respect to this article were associated with getter and setter functions for this kind of data (Figure 5.26). The code, written in Java, created a class for each of the previous main elements - Device, Position and User.

A big difference of Traccar with respect to the previous analysed system was the user control over the information and data used by the code. In fact, the user has a high control of the information, allowing eventual modifications and deletion and listing of the data collected (*art. 1798.110*). This was a big lack for LibreTaxi and Selfmailbot, which never managed the possible need for the user to know what information where stored (*RequestDisclosure* ==

```
ConjunctClause(role: Business, schema: ['ConsumerID', 'Name', 'Email',
    'HashedPassword', 'Salt', 'Latitude', 'Longitude'], purpose:
    BusinessPurpose, privacy: Anonymization),
ConjunctClause(role: Business, schema: ['ConsumerID', 'Name', 'Email',
    'HashedPassword', 'Salt', 'Latitude', 'Longitude'], purpose:
    BusinessPurpose, privacy: Aggregation),
ConjunctClause(role: Business, schema: ['PositionID', 'DeviceID', 'Latitude',
    'Longitude', 'Altitude', 'Address'], purpose: BusinessPurpose, privacy:
    Anonymization),
ConjunctClause(role: Business, schema: ['PositionID', 'DeviceID', 'Latitude',
    'Longitude', 'Altitude', 'Address'], purpose: BusinessPurpose, privacy:
    Aggregation),
ConjunctClause(role: Business, schema: ['DeviceID', 'PositionID'], purpose:
    BusinessPurpose, privacy: Anonymization),
ConjunctClause(role: Business, schema: ['DeviceID', 'PositionID'], purpose:
    BusinessPurpose, privacy: Aggregation)
```

Figure 5.26. Traccar for CPRA: residual policies for privacy usage.

'Y'). A residual policy was still obtained regarding *HashedPassword* and *Salt*, which are data not accessible by the user of the system (Figure 5.27).

```
ConjunctClause(role: ConsumerID, schema: ['HashedPassword', 'Salt'], filter:
    RequestDisclosure [eY, eY], filter: ConsumerID [en_consumer,
    en_consumer]),
```

Figure 5.27. Traccar for CPRA: residual policies user private information.

The following step was the analysis of the code with GDPR as input for the privacy file. The results we obtained were very similar and allowed to highlight the main lacking points for Traccar. In fact, the residual policy highlights the problem of lack of consents request. Figure 5.28 shows that, other than not checking for consent to process the information (*ConsentProcessing*), there is no checking for the age of the user that gives this information. We were able to notice that Traccar never requires to give an age when signing up. It can be a problem, mainly when the information is about underage users. *Art. 8* states: "The processing of the personal data of a child shall be lawful where the child is at least 16 years old. Where the child is below the age of 16 years, such processing shall be lawful only if and to the extent that consent is given or authorised by the holder of parental responsibility over the child.

Member States may provide by law for a lower age for those purposes provided that such lower age is not below 13 years. The controller shall make reasonable efforts to verify in such cases that consent is given or authorised by the holder of parental responsibility over the child, taking into consideration available technology". The previously cited article defines the importance to check and guarantee consent from the user himself or an eventual guardian (*GuardianConsent*) if they are younger than 13 years old.

Another highlighted issue is the storage of information. As seen in previous examples, GDPR requires a minimum level of privacy *Anonymization* for the use of the data (*Art. 25.1*). This request is not met by Traccar for any of the data stored (Figure 5.29).

As already seen with CPRA, Traccar allows the user to have higher control over managing the information. No residual policy associated with users' requests to access and modify the data has been obtained. Eventual modification and retrieval of the data were possible by the

```
ConjunctClause(role: Controller, schema: ['DataSubjectID', 'Name', 'Email',
    'HashedPassword', 'Salt', 'Latitude', 'Longitude'], filter:
    ConsentProcessing [eY, eY], filter: Age [e16, einf], purpose: Processing),
ConjunctClause(role: Controller, schema: ['DataSubjectID', 'Name', 'Mail',
    'HashedPassword', 'Salt', 'Latitude', 'Longitude'], filter:
    ConsentProcessing [eY, eY], filter: GuardianConsent [eY, eY], filter: Age
    [eninf, e15], filter: Age [e13, einf], purpose: Processing),
ConjunctClause(role: Controller, schema: ['DeviceID', 'PositionID'], filter:
    ConsentProcessing [eY, eY], filter: Age [e16, einf], purpose: Processing),
ConjunctClause(role: Controller, schema: ['DeviceID', 'PositionID'], filter:
    ConsentProcessing [eY, eY], filter: GuardianConsent [eY, eY], filter: Age
    [eninf, e15], filter: Age [e13, einf], purpose: Processing),
ConjunctClause(role: Controller, schema: ['PositionID', 'DeviceID',
    'Latitude', 'Longitude', 'Altitude', 'Address'], filter:
    ConsentProcessing [eY, eY], filter: Age [e16, einf], purpose: Processing),
ConjunctClause(role: Controller, schema: ['PositionID', 'DeviceID',
    'Latitude', 'Longitude', 'Altitude', 'Address'], filter:
    ConsentProcessing [eY, eY], filter: GuardianConsent [eY, eY], filter: Age
    [eninf, e15], filter: Age [e13, einf], purpose: Processing)
```

Figure 5.28. Traccar for GDPR: residual policies for consent usage.

```
ConjunctClause(role: Controller, schema: ['DeviceID', 'PositionID'], privacy:
    Anonymization),
ConjunctClause(role: Controller, schema: ['PositionID', 'DeviceID',
    'Latitude', 'Longitude', 'Altitude', 'Address'], privacy: Anonymization),
ConjunctClause(role: Controller, schema: ['DataSubjectID', 'Name', 'Email',
    'HashedPassword', 'Salt', 'Latitude', 'Longitude'], privacy:
    Anonymization)
```

Figure 5.29. Traccar for GDPR: residual policies for privacy.

user, as requested by *Art. 15* and *Art. 16*. The only limitation is the impossibility to retrieve *Salt*, *HashedPassword* (Figure 5.30). It is easily possible to access and modify the rest of the information.

```
ConjunctClause(role: DataSubjectID, schema: ['HashedPassword', 'Salt'],
    purpose: Processing, filter: RequestProcessing [eY, eY]),
ConjunctClause(role: DataSubjectID, schema: ['HashedPassword', 'Salt'],
    filter: RequestRectification [eY, eY]
```

Figure 5.30. Traccar for GDPR: residual policies user private information.

Chapter 6

Conclusion

The goal of this thesis was to find a possible automated solution for policy compliance verification. Based on previous research and papers, we selected PrivAnalyzer as the best option to manage this problem. It is important to state that the source code of this framework is not already used commercially, so we tested it to see if it was the solution we were looking for. To understand if the result was positive we performed all the automated analysis and then compared the results with a human analysis. By human analysis, we mean verification of the software systems (LibreTaxi, Selfmailbot and Traccar), to see how they worked and if the residual policies were actually correct. The verification was done by selecting CPRA and GDPR as policies that the software system needed to comply with. We translated them into a form that was usable by the analyser. This means that they were re-written into single clauses in a programming language that implemented Legalease. From the laws we translated only the articles which could be checked by PrivAnalyzer; excluding the ones regarding the legal consequences and requirements with respect to the website interface. The choice of software systems for this research was done based on their deployment language and what they were used for. We needed to select options that managed user data, mainly in a way that is not privacy-friendly, this would allow to have residual policies from the analyser. LibreTaxi and Selfmailbot allowed to see some of the worst-case scenarios with a high number of residual policies. Traccar gave a fewer results, permitting us to see a different test case for PrivAnalyzer. Secondly, they needed to be written in Python, since PrivAnalyzer worked only with those kinds of files. Selfmailbot was the only one that satisfied these criteria, for the other source codes we needed to translate them, so they needed to be in a language we could understand. *PrivAnalyzer* requires a specific format for the Python code, so it was necessary to modify also Selfmailbot code to satisfy the requirements.

From the results obtained, we can say that PrivAnalyzer is a valuable solution for automated privacy compliance verification. It reduces the time to check and analyse a system when given a set of privacy clauses. It doesn't require to have actual data to perform verification, so there is no danger to possible consumer data when using it. The language used by it for writing the privacy clauses is very simple and easy to understand, requiring only a small training for the use of the analyser. It also allows a high degree of freedom, so can be used on different systems which have different naming of the variables, by changing only some names. The results themselves, since they are based on the policy clauses are understandable and easy to compare to the specific clause they are referring to.

There are some disadvantages associated with this solution. First of all, it limits the usage to Python codes, so it is possible to analyse only codes written in Python. The number of libraries implemented for the code is limited, so we can encounter some errors when using libraries not implemented by the source code of PrivAnalyzer. For big codes, developed through different files and directories, sometimes it is necessary to perform the analysis on multiple files to understand where the problem specifically occurs. It is not possible to check the correctness of ROLE. The initial research [1] assumed that all the ROLE attributes would be logged in a record as a way to hold people accountable and know who was responsible. Other than the language, the analyser limits the format of the code. It must follow a specific format to undergo checking through PrivAnalyzer, requiring possible extra work for the analyst.

Concerning CPRA and GDPR, they represent only a sample of the privacy laws currently available. CPRA was a more specific solution for our work since it was made specifically for the business-consumer relationship. We believe it needs some improvements, specifically when talking about requirements for the privacy of data. In *art. 1798.100.d* and *art. 1798.100.e* CPRA indicates the need to implement "security procedures and practices appropriate to the nature of the personal information". This definition is vague and not as helpful as the one given by GDPR when introducing pseudonymisation as a minimum form of protection for the data. GDPR, on the other hand, has been around for longer and covers the privacy of a broader set of data, e.g. criminal records, and for more situations, e.g. historical research, scientific research or public interest.

For possible future works, we think it is important to implement and update PrivAnalyzer source code to fill the gaps associated with the libraries. An possible further analysis could be performed with other laws and their translation. Since the beginning of this research work new acts and laws are being introduced around the United States, such as Colorado, Connecticut, Utah and Virginia. During our search for the systems to analyse, we encountered many options written in Go or Python, that manage consumers' data. We initially discarded these codes because they needed a lot of work for their translation and/or used some libraries not yet implemented with PrivAnalyzer. We think there is still some work that could be done with respect to this field, to guarantee the best solution to a problem that is rising very fast. We reached a point where the help of technology is highly necessary, mainly for what concerns technology itself. We believe an automated solution, such as PrivGuard will allow a faster and better execution of a job that is becoming every day more important and valuable.

Bibliography

[1] L. Wang, U. Khan, J. P. Near, Q. Pang, J. Subramanian, N. Somani, P. Gao, A. Low, and D. Song, "Privguard: Privacy regulation compliance made easier", 31st USENIX Security Symposium, Boston (MA,USA), August 10-12, 2022, pp. 3753–3770. https://www.usenix.org/system/files/sec22-wang-lun.pdf

[2] "CPRA:California Privacy Rights Act." https://cpra.gtlaw.com/cpra-full-text/

[3] "GDPR:General Data Protection Regulation." https://gdpr-info.eu/

[4] L. Wang, J. P. Near, N. Somani, P. Gao, A. Low, D. Dao, and D. Song, "Data capsule: A new paradigm for automatic compliance with data privacy regulations", Heterogeneous Data Management, Polystores, and Analytics for Healthcare - VLDB 2019 Workshops, Poly and DMAH, Los Angeles (CA, USA), August 30, 2019, pp. 3–23, DOI 10.1007/978-3-030-33752-0_1

[5] "PrivGuard: Github source code." https://github.com/sunblaze-ucb/privguard-artifact

[6] S. Sen, S. Guha, A. Datta, S. K. Rajamani, J. Y. Tsai, and J. M. Wing, "Bootstrapping privacy compliance in big data systems", 2014 IEEE Symposium on Security and Privacy, Berkeley (CA, USA), May 18-21, 2014, pp. 327–342, DOI 10.1109/SP.2014.28

[7] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, S. Antony, H. Liu, and R. Murthy, "Hive - a petabyte scale data warehouse using hadoop", 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010), Long Beach (CA, USA), March 01-06, 2010, pp. 996–1005, DOI 10.1109/ICDE.2010.5447738

[8] M. Sergey, G. Andrey, L. Jing Jing, R. Geoffrey, S. Shiva, T. Matt, and V. Theo, "Dremel: Interactive analysis of web-scale datasets", 36th International Conference on Very Large Data Bases, Singapore, September 13-17, 2010, pp. 330–339, DOI 10.14778/1920841.1920886

[9] C. Ronnie, J. Bob, L. Per-Åke, R. Bill, S. Darren, W. Simon, and Z. Jingren, "SCOPE: easy and efficient parallel processing of massive data sets", 34th International Conference on Very Large Data Bases, Auckland (New Zeland), August 23-28, 2008, pp. 1265–1276, DOI 10.14778/1454159.1454166

[10] V. Julião, A. Holmquist, F. Lúcio, C. Simões, and F. Pereira, "Hapi: A domain-specific language for the declaration of access policies", SBLP'21: 25th Brazilian Symposium on Programming Languages, Joinville (Brazil), 27 September 2021 - 1 October, 2021, pp. 9–16, DOI 10.1145/3475061.3475084

[11] S. Kasem-Madani and M. Meier, "Security and privacy policy languages: A survey, categorization and gap identification", CoRR, vol. abs/1512.00201, December 2015. https://doi.org/10.48550/arXiv.1512.00201

[12] M. Azraoui, K. Elkhiyaoui, M. Önen, K. Bernsmed, A. Santana de Oliveira, and J. Sendor, "A-PPL: an accountability policy language", Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance, Wroclaw (Poland), September 10-11, 2014, pp. 319–326, DOI 10.1007/978-3-319-17016-9_21

[13] J. Yang, K. Yessenov, and A. Solar-Lezama, "A language for automatically enforcing privacy policies", Proceedings of the 39th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Philadelphia (PA,USA), January 22-28, 2012, pp. 85–96, DOI 10.1145/2103656.2103669

[14] J. Iyilade and J. Vassileva, "P2U: A privacy policy specification language for secondary data sharing and usage", 2014 IEEE Security and Privacy Workshops, San Jose (CA, USA), May 17-18, 2014, pp. 18–22, DOI 10.1109/SPW.2014.12

[15] U. Khan, L. Wang, J. Subramanian, J. P. Near, and D. Song, "Privframework: A system for configurable and automated privacy policy compliance", NeurIPS 2020 Workshop on Dataset Security and Curation, Online, December 11, 2020, DOI 10.48550/arXiv.2012.05291
[16] "Libretaxi: GitHub source code." https://github.com/ro31337/libretaxi
[17] "LibreTaxi: Telegram bot channel." https://t.me/libretaxi_bot
[18] "LibreTaxi: Telegram public chat." https://t.me/libretaxi_all
[19] "Selfmailbot: GitHub source code." https://github.com/f213/selfmailbot
[20] "Traccar: GitHub source code." https://github.com/traccar/traccar
[21] "Traccar manager android app: source code." https://github.com/traccar/traccar-client-android
[22] "Traccar manager android app: source code." https://github.com/traccar/traccar-manager-android
[23] "Traccar client ios app: source code." https://github.com/traccar/traccar-client-ios
[24] "Traccar manager ios app: source code." https://github.com/traccar/traccar-manager-ios
[25] "Traccar web app: source code." https://github.com/traccar/traccar-web
[26] "Privguard: personal source code github." https://github.com/sofialucca/thesisResearch