User centric personal data management

Paul Marillonnet

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2021IPPAS011

Thèse de doctorat

La gestion des données personnelles par l'usager au sein des collectivités locales

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom SudParis

École doctorale n°626 Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat: Informatique

Thèse présentée et soutenue à Évry, le 30 novembre 2021, par

PAUL MARILLONNET

Composition du Jury :

Sonia Ben Mokhtar Directrice de recherche, LIRIS Lyon	Présidente
Abdelmadjid Bouabdallah Professeur, Université de Technologie de Compiègne	Rapporteur
Romain Laborde Maître de conférence, Institut de Recherche en Informatique de Toulouse	Rapporteur
Karima Boudaoud Maîtresse de conférence, Polytechnique Nice	Examinatrice
Nicolas Anciaux Directeur de recherche, INRIA	Examinateur
Maryline Laurent Professeure, Télécom SudParis	Directrice de thèse
Mikaël Ates Ingénieur-docteur, Entr'ouvert	Encadrant industriel
Nesrine Kaaniche Maîtresse de conférence, Télécom SudParis	Invitée

This Ph.D. thesis has been funded by, and realized at Entr'ouvert, Paris, France[1] as part of the CIFRE industrial & research program.

[1] https://www.entrouvert.com/

Contents

Acknowledgements

I would like to thank my three Ph.D. thesis supervisors, Maryline Laurent (Ph.D. director), Mikaël Ates and Nesrine Kaaniche for their unfailing support over the past three years. They have provided immense technical and scientific guidance for the accomplishment of the thesis.

I would also like to thank the Entr'ouvert team for their precious advice, their deep technical insight and knowledge of identity and personally identifiable information management, as well as the many reviews they have provided for my work. Their sense of humor has also (often) proven to be very helpful–as well as, more generally, the encouraging and pleasant peer-to-peer work atmosphere they have been enabling. My grieving thoughts go to our late colleague Laurent, who passed away in November 2020, and to his family. Laurent, on top of being a fierce advocate of free software and an inspiring public figure in many French free-software communities, has on a personal basis provided a warm welcome and encouragement of my research work.

I would like to thank my family, my friends and my much-beloved partner-in-crime Akhila, for their continuous support in the many challenges that have arisen in the past three years.

The manuscript front page uses the *Institut Polytechnique de Paris* template, written by Guillaume Brigot and updated by Aurélien Arnoux, the two of whom I would like to thank for that very reason.

Abstract

This Ph.D. thesis addresses the user-centric management of Personally Identifiable Information (PII) within local collectivities. It has been realized as part of a CIFRE program between SAMOVAR lab and Entr'ouvert.

There is a strong need to provide the users of the collectivities' online service with some PII management tools for respecting their privacy when submitting online requests to their collectivities. This need is also coupled with the challenges of free software (including open access to the code, and possibility to evaluate the software's security), which is part of Entr'ouvert's philosophy.

For illustration, a realistic use case is identified for the specific context of territorial collectivities and the public administration (TCPA). It enables to establish a list of useful functional requirements, and a set of users capabilities regarding the management of their own PII.

The first contribution is about a technical comparative survey of academic and industrial solutions. This survey identifies thirteen solutions belonging to four different categories, and evaluates them according to fourteen functional criteria. Eventually, the survey provides per-category synthesis and identifies an optimal solution for our use case.

The second contribution proposes a solution for supporting PII management, which respects the guidelines identified earlier as part of the survey's optimal solution. It also takes into consideration the PII retrieval from third-party sources. The solution, called the PII manager, operates thanks to its four main components: [i] the Source Backend (SB), [ii] the Core Consent Management module (CCM), [iii] the PII Query Interface (PQI) and [iv] the PII Management User Interface (PMUI). A detailed description of each of these four components is given in the manuscript. Additionally, the user-identifier mapping performed by the PQI is identified as a critical part of the solution. It requires security considerations, as failing to verify the consistency of this mapping can enable four types of attacks.

The third contribution proposes an identity-matching solution to counteract the previously identified attacks. Indeed, there is a need to verify the validity of user identity information retrieved across several PII sources. This identity-matching solution requires to identify which components of the architecture is involved in that processing, the workflow across these components to support the full processing, and to perform a security analysis of the workflow that proves its strength against identified attempted attacks.

The fourth contribution is the software validation of the proposed solutions through a proof of concept. The identity-matching solution is implemented thanks to the Django template filters and Entr'ouvert's existing User-Relationship Management (URM) tool. The PII manager is also implemented as a new component to the existing software platform. Eventually, new perspectives are drawn. For instance, this research work could benefit from upcoming protocols such as the Grant

Negotiation & Authorization Protocol (GNAP). Other new perspectives include the integration of the System for Cross-domain Identity Management (SCIM) into the platform and a larger-scale software validation.

French Summary – *Résumé en français*

Cette thèse de doctorat adresse la gestion centrée usager des Données à Caractère Personnel (DCP)[2] au sein des collectivités locales et de l'administration publique. Elle a été réalisée en programme CIFRE entre le laboratoire SAMOVAR et la coopérative Entr'ouvert, éditrice de logiciel libre de gestion de la relation à l'usager, à destination des collectivités.

Il y a un besoin crucial de fournir aux usagers — des services en ligne des collectivités locales et de l'administration publique — des outils pour la gestion de leurs DCP qui soient respectueux de leur vie privée, et utiles lors de la soumission de requêtes en ligne à ces collectivités. Ce besoin est par ailleurs associé aux enjeux du logiciel libre (dont notamment l'accès libre au code source, ainsi la possiblité d'auditer la sécurité du logiciel à partir des sources), qui est le mode de production d'Entr'ouvert ainsi que l'un de ses domaines de spécialisation.

Ainsi, pour adresser ce besoin, un cas d'usage réaliste est identifié pour le contexte spécifique des collectivités territoriales et de l'administration (CLA). Ce cas d'usage permet de dresser une liste d'exigences fonctionnelles utiles, et un ensemble de fonctionnalités offertes à l'usager pour la gestion de ses données.

La première contribution décrit un état de l'art technologique comparatif de différentes solutions sur les plan académique et industriel. Cet état de l'art recense treize solutions, appartenant à quatre catégories différentes, et il en offre une évaluation à l'aide de dix-huit critères fonctionnels. Enfin, cet état de l'art présente une synthèse par catégorie de solution, et identifie une solution jugée optimale au regard de notre cas d'usage territorial.

La seconde contribution propose quant à elle une solution pour la gestion des DCP, respectueuse des prescriptions identifiées en amont lors de l'identification de la solution optimale de l'état de l'art. Cette seconde contribution prend aussi en considération des problématiques telles que notamment la récupération de DCP depuis des sources tierces et une gestion accrue des consentements de l'usager. La solution issue de cette contribution, dénommée gestionnaire de DCP, fonctionne à l'aide de ses trois composantes principales : [i] le backend de source (BS), [ii] l'interface de requête de DCP (*Personally-identifiable information Query Interface* – PQI) et [iii] l'interface utilisateur de gestion des DCP (*Personally-identifiable information Management User Interface* – PMUI). Une description détaillée de chacune des ces trois composantes est fournie dans le manuscrit. En outre, l'appairage d'identifiant utilisateur tel que réalisé par l'IRD est identifié comme étant une partie critique de la solution. Il implique des considérations de sécurité, car échouer à vérifier cet appairage de façon cohérente rend possible quatre types d'attaques différentes.

[2]En anglais dans le reste du manuscrit : *Personally-identifiable information* – PII.

La troisième contribution propose une solution de concordance d'identité pour prévenir les attaques précédemment identifiées. En effet, il est nécessaire de vérifier la validité des informations utilisateurs retrouvées au travers des sources de DCP. Cette solution de concordance d'identité implique d'identifier les composants de l'architecture impliqués dans ce processus, the circuit de traitement de ces composants supportant le processus complet, mais aussi d'établir une analyse de sécurité de ce circuit, démontrant sa robustesse face aux tentatives d'attaques identifiées.

La quatrième contribution est la validation logicielle des solutions proposées en tant que preuve de concept. La solution de concordance d'identité est implémentée à l'aide de filtres de gabarit Django sur la plateforme logicielle de Gestion de Relation de l'Usager (GRU) éditée par Entr'ouvert. Le gestionnaire de DCP est aussi implémenté en tant que nouveau composant de cette plateforme existante. Enfin, de nouvelles perspectives sont dressées. Par exemple, ce travail de recherche pourrait tirer bénéfice de protocoles émergeants tels que Grant Negotiation & Authorization Protocol (GNAP).

Abbreviations & Acronyms

AC	Access Control
ACL	Access-Control List
API	Application Programming Interface
AS	Authorization Server
ASP	Administrative Service Provider
Authn.	Authentication
Authz.	Authorization
C	Client
CCM	Core Consent Management
CRL	Certificate-Revocation List
CRM	Citizen-Relationship Management
CRUD	Create, Read, Update, Delete
FIM	Federated Identity Management
GDPR	General Data Protection Regulation
GNAP	Grant Negotiation & Authorization Protocol
GUI	Graphical User Interface
IDaaS	Identity as a Service
IdM	Identity Manager
IDP	Identity Provider
IETF	Internet Engineering Task Force
IRTF	Internet Research Task Force
JSON	JavaScript Object Notation
KDC	Key-Distribution Center
OIDC	OpenID Connect
PDS	Personal Data Store
PII	Personally Identifiable Information
PMUI	PII Management User Interface
PQI	PII Query Interface
PSP	Private Service Provider
RBAC	Role-Based Access-Control
REST	Representational State Transfer
RFC	Request For Comments
SaaS	Software as a Service
SAML	Security Assertion Markup Language
SB	Source Backend
SCIM	System for Cross-domain Identity Management

SLO	Single Logout
SoC	System on Chip
SP	Service Provider
SSO	Single Sign On
TCPA	Territorial Collectivities and Public Administration
TGS	Ticket-Granting Server
TGT	Ticket-Granting Ticket
TTL	Time To Live
TTS	Token Translation System
UI	User Interface
UMA	User-Managed Access
URM	User-Relationship Management
UUID	Universally Unique Identifier
VPN	Virtual Private Network
WebSSO	Web Single Sign On
XML	eXtensible Markup Language
ZKP	Zero-Knowledge Proofs

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides an introduction of the thesis subject, including the scientific context and the work organization. It also gives the detailed manuscript organization, *i.e.* the summed up content of each chapter.

This thesis has been realized as a collaboration between Entr'ouvert, Paris, France and SAMOVAR lab, Télécom SudParis, Institut Polytechnique de Paris, Évry, France. It has also be founded by the French National Agency for Research and Technology (ANRT) as part of their industrial-academic thesis program (CIFRE).

The subject was proposed in French and submitted to the ANRT (*National Research & Technology Agency*). It is titled "*The management of personal data by the user within local collectivities*"[1].

1.1 Legislative and Economic Context

1.1.1 Thesis Context

The recent events of user privacy infringement and personal data theft (or leakage) on the Web include the Equifax scandal [20], the Cambridge Analytica case [11] and hacking attacks at the expense of many multinational firms [88, 94]. The PII leaked during these privacy infringement events can be bank information or any other kind of information that has a monetary value, either directly or indirectly (*e.g.*, through user impersonation fraud)–*i.e.*, users' addresses, identity documents, phone line documents or miscellaneous contractual documents.

These attacks point out the urgent need to empower users with their personal data governance, by letting them manage their Personally Identifiable Information (PII) along their full lifecycle.

The production, retention, processing and distribution of personal data is a regulated concern, especially since the *Law on digital information and liberties*[2] of 1978 [33], the European directive *95/46/CE* [26] and the European treaty *#108: Convention for the Protection of Individuals with regard to Automatic Processing of Personal Data* [19]. The European regulation

[1] *La gestion des données personnelles par l'usager au sein des collectivités locales.*
[2] *Loi informatique et libertés*

2016/679 of the European Parliament and the Council of April 27th, 2016 on personal data protection (GDPR) has been consolidating the right to user determination regarding their PII since May 25th, 2018 [27].

This strong regulatory context aims at protecting user privacy in a digital environment presenting ever-increasing threats. We notice, for instance, an increasing number of identity thefts. In France, in 2014 this number was already of 14 060 cases, perpetrated by an estimate of 120 000 thieves [22] As a matter of fact, the main reason for identity theft is the personal data available online [82].
The thesis subject includes the security of user-controlled personal-data exchanges in the context of online procedures offered by a local collectivity. The term *collectivity* should be understood by the reader in the French administrative context, *i.e.*, as a subdivision of the state's territory which is granted some partial autonomy by the central government.

The French state's modernization program, enforced both by the French interministerial directions for (i) public transformation and for (ii) digital services and for the State's information and communication system[3] is similar to the *Tell us once* business-oriented program [89]. This modernization program should be introduced for citizens in a relatively near future.

Additionally, the article 90 of the French Digital Services Law from October 7^{th}, 2016 states that the information necessary in order to process a request *can be obtained directly from another administration.* The destination service may get this information directly from the administration, provided that it collected a declaration. That declaration must include the user's authorization for the service to act on the user's behalf.

However, in an effort to increase the user-centric PII determination, it is worth enabling users to manage the exchanges of their information and data produced by or sent to any collectivity, including local collectivities. The target PII includes identity attributes (civil status, electronic mail and postal addresses), fiscal information, information regarding household, social rights, bills, user preferences and authorizations for advertisement purposes. This ability is particularly relevant when exchanging information between information systems within a single local collectivity or between local collectivities and the public administration (*e.g.*, between the city and the metropolis it belongs to; or between the city and the national administration).

In order to enable this data exchange schemes, matching the user's identities among the various applications and information systems is necessary. As a matter of fact, the ministerial decree of July, 4th 2013 [32] regarding online services *bans the creation of a population file or of a unique user identifier by the administrative authority.*

On the contrary, performing identity matching on the user's PII from different locations is allowed by the administration: indeed [32] also states that *in order to avoid asking for the user an information when that information has already been produced when using a data-processor service, and when that information is required for an administrative procedure offered by another data processor*[4]*, the latter can collect said information from the service detaining it, after obtaining the explicit and unequivocal consent from the user.*

Empowering users to self managing their personal data makes the thesis subject adhere to the regulatory context of these collectivities. As previously explained, this context is purposely

[3] DINUM, formerly DINSIC.
[4] *Data processor* is the legal term used to describe any online service that collects, stores or processes users' PII.

restrictive and it involves technologies that enable PII exchanges between information systems without any unique administrative identifier, providing that user consent and control are given.

1.1.2 Specifications Adhering to Legislative and Operational Requirements

The thesis subject tackles the security of user-controlled personal-data exchanges for a distribution when the user is online or offline, more precisely in the context of online procedures offered by a local collectivity. Considering the context imposed at national and European levels, the legislative properties of user-centric personal-data management within local collectivities can be summarized as follows.

- *Global objective addressing the French state's modernization program:* the user conveys personal data, whether it be certified or not. The target system must allow the user to convey personal data to third-parties in order to customize the offered service. This data must be at least partially issued, validated or certified by the service provider's trusted third party.
- *Compliance with the GDPR:* the principles of consent, data minimization, accountability, the rights to be forgotten (Section 3, Article 17), and to opposition (Section 4, Article 21) must be respected. The target system must comply with the regulation and thus enforce *privacy-by-design* principles. Thus the service provider must obtain the user's consent when collecting their personal data; the request for consent must be clearly stated, along with the purposes for which the user data are collected. The provider must not ask for more data than strictly required in order to achieve its processing. The service provider must afterwards be able to prove that the user's consent was fulfilled, and must keep the data and the proofs during a time window specified by the GDPR. The user is able to ask for the deletion or the correction of any subset of their collected data.
- *User's control on the retention and use of their personal data:* the user must be able to keep a digital footprint of the data exchange and must be able to obtain the guarantee that the retention and the use of their collected data consider their consent and the regulation. Ideally, the system allows the user to proceed to these checks, without the need for a monitoring third-party.
- *Compliance with the ministerial decree of July 4^{th}, 2013:* a partitioning of the user identifiers from one service to another must be enforced. The data exchange happening between service providers, including providers maintained by different third parties, at least partially involving the user's digital identity, must not rely on a unique user identifier.
- *User online and offline modes:* the data exchanges happening between providers must be possible even whether the user is online, submitting a request, or offline, when a service needs to access some data required for the processing of the request. For this purpose, obtaining the user's consent prior to said access is necessary.

1.1.3 Absence of User-Centric Solution Meeting the Requirements

There are nowadays mature technologies covering Web Single Sign-On (WebSSO) and identity federation features (SAML2 [68], OpenID Connect [83]) or emerging ones covering user management authorization(UMA [55] and Consent Receipts [47]). SAML2 [68] and ID-WSF [2] have been built on the basis of two distinct roles for the identity provider and the attribute provider– although ID-WSF never encountered the expected success, and SAML is systematically deployed

with an identity provider that also bears the role of attribute provider during the connection.

[35, 70] satisfy most of the identified functional requirements, but do not comply with the aforementioned protocols. User-driven data lifecycle management is indeed still not part of identity management domains for which standard protocols such as [68, 83, 55] have been proposed. Neither do architectural solutions such as personal data banks. Indeed, they do not address informational self-determination throughout the personal-data lifecycle. More recent work of user-centric personal data management [35, 70] or even [61], do not propose a direct solution to consistent and user-driven provisioning/deprovisioning.

Similarly, thorough user-centric solutions addressing access-revocation constraints have not been stated yet. Within all the work considered for the survey, the data consumers definitively and permanently possess the personal data that they were provided with at a given point in time, even when an access revocation occurred in between–see for instance [67, 95, 35, 70, 62].

Eventually, various research work such as [12, 62, 61, 60] highlight the significance of deploying data-protection technologies at a national or European scale, complying with the current legal context. However, the systems ensuring the compliance with the legal constraints of privacy are not present at every data-processing stage in the proposed solutions [70, 62, 61].

For instance, the architecture proposed by [12], despite being a reference in the field of identity management ensuring the user's governance of their personal data, requires a strong trust link between the user and the data consumer. The data consumer agrees to decipher the user's data only under specific conditions–in particular through what is referred to as a *data decryption policy* in [12].

The thesis solution must also rely on components as catalogs, directories and providers' metadata enabling the determination of data sources. A proposition of architecture for the Internet of Things (IoT), relying on the use of a catalog, but addressing the constraints only in a partial manner, is described in [3].

Eventually, applications of blockchains such as cryptocurrencies–see for instance the Bitcoin cryptocurrency [64]–illustrate the possibility to distribute some, usually centralized, features over different actors, in order to strengthen overall trust in the system (fulfilling several properties such as tamperproofing and chain monitoring). The approach will be studied for trusted third parties when data is exchanged between users and the governmental entities.

1.2 Detailed Objectives of the Thesis

Our objective is to study, analyze and specify an architecture responding to the objectives as follows.

Architectural solutions addressing the specifications must deal with the concern of user consent when sharing data for the purpose of an online transaction, as well as the emerging concern of the definition of consent for data disclosure happening while the user is offline. Consent management offers the advantage of reducing the temporal validity of the consents, as well as to ensure their revocation when necessary. While enforcing the property of data unlinkability, the architecture will also have to consider user choices regarding the lifecycle of their data. The "*right to be forgotten*" and the "*right to opposition*", enforcing privacy by design, would then be the pillars of the proposed architecture The architecture's informational control would be given to the resource owners, throughout the informational lifecycle.

The double feature of (i) personal data storage and (ii) the deployment of an intermediation platform between the source entities which store data, and the destination entities which process this data, will also be addressed.

The stake for these proposed solutions will be to switch from a data-consumption scheme, to mechanisms allowing property-validation requests. For this purpose, the proposed solutions will eventually use and extend side-channel initiation mechanisms for data- or property-retrieval challenges, such as presented by [63].

Leveraging pseudonymity and cryptographic certificates, such as formalized in [12] thus could ensure data unlinkability, addressing the constraints of the General Data Protection Regulation (GDPR).

Consequently, the choice of the user whether to exclude profiling features will have to be included in the objectives–although the commercial purpose of profiling, in order to add business value to user data, is *a priori* unjustified in the context of the services provided by territorial collectivities.

Additionally, issues arising with gathering Personally Identifiable Information (PII) from multiple sources, and in particular identity-matching, should be discussed.

The subject is then set in the domains of

- digital-identity federation [12, 17];
- trusted architectures whose access-control relies on cryptographic certificates [5] and attributes [40];
- management of consent and personal-data online submission.

The brief preview offered by the current survey led on the domains of interest highlights the absence of a cryptographic, protocol and architectural answer offering the target functional coverage.

The thesis contributions are presented as follows

1. the definition of a user-centric solution,
2. the combination of existing cryptographic tools and algorithms addressing the identified objectives,
3. the design of protocols and their specification based on standards
4. the validation of different levels of proof: simulation or mathematical proofs, and proof of concept.

1.3 Manuscript Organization & Content Summary

The remainder of the manuscript is organized as follows:

Chapter 2 introduces the problem statement.
First, it proposes a list of the actors involved, and describes our main territorial use case. Second, the functional requirements that apply as part of the use case are also listed Third, this chapter presents the industrial context at Entr'ouvert. Eventually, the security hypotheses that have been selected for this problem are presented

Chapter 3 presents a technological survey that covers both academic and industrial solutions. Those solutions cover some parts of the use case and address some functional requirements. This survey relies on fourteen functional criteria used to evaluate a selection of thirteen solutions. These solutions belong to four categories: (i) personal data store (PDS), (ii) identity manager (IdM), (iii) anonymous certificates and (iv) delegation architectures. A per-solution evaluation is provided, and eventually a coarse-grained per-category synthesis is given. This chapter concludes by identifying missing features for these solutions that an optimal solution would address. The way to address these features are discussed in the following chapters.

Chapter 4 specifies the management of user Personally Identifiable Information (PII) that are provided by third-party sources. It does so by presenting a PII manager and its different component that make this management of PII across sources work. These third-party sources being a legitimate part of the territorial use case, managing the PII they provide is a significant part of the requirements of the architecture. This chapter defines a consent model that enables the support of the relevant critical criteria identified in Chapter 3, *i.e.*, the support of PII sources, the extent of delegation & consent management, and the support of online & offline modes.
It also addresses the interoperability concerns that arise when dealing with PII sources of different types.

For this purpose, three main components of the PII manager must be specified: (i) the source backend, responsible for the protocol logic when interfacing heterogeneous sources; (ii) the PII query interface, providing a single PII endpoint for territorial online services (hence abstracting the aforementioned heterogeneity); (iii) the PII management user interface, addressing the user governance of the PII offered by the PII manager.

Chapter 5 provides a solution to identity-matching concerns that arise when retrieving PII across several sources as described in the previous chapter. The typical sources that require identity-matching within our territorial use case are presented in this chapter, along with the base components participating in the identity-matching process. Eventually, this chapter proposes a security analysis of the matching process, according to an attacker model performing different types of attacks.

Chapter 6 presents the validation of the identity-matching process as specified in Chapter 5. This validation has been performed on the existing URM software suite developed and maintained by Entr'ouvert. Additionally, this chapter presents the proof of concept of the PII manager specified earlier in Chapter 4. It provides implementation considerations and technical choices that were made according to the industrial context as well as Entr'ouvert's existing software ecosystem.

Chapter 7 provides the global conclusion of the manuscript, and also suggests new perspectives of research for user-centric personal data management within territorial collectivities.

Chapter 2

Problem Statement of User-Centric Personal-Data Management within Territorial Collectivities and the Public Administration

This chapter states the main problem of this thesis. It starts by defining the five main actors of the problem, and then proposes a use case that fits the thesis subject. A list of functional requirements, derived from the use case, is then proposed. A brief presentation of the industrial context is given, and the main security hypotheses of the thesis are given.

2.1 System actors

Figure 2.1 depicts the actors and their interactions.

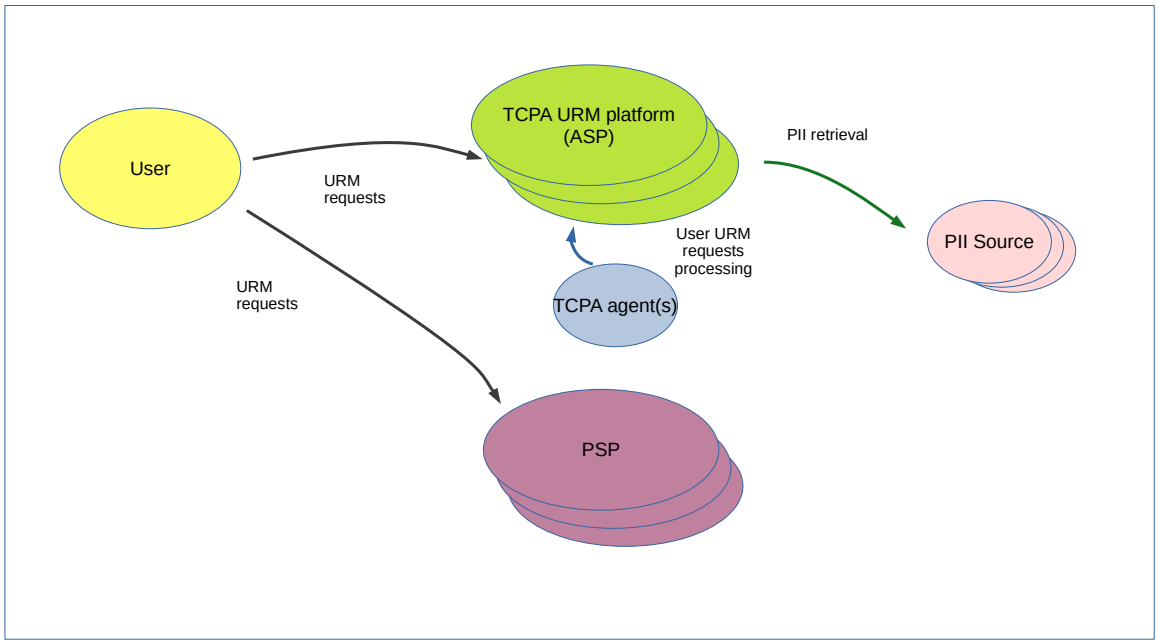


Figure 2.1: Problem statement — use case actors

The use case involves the five following actors:

- **The users of the online TCPA services**: they submit through their web client one or several URM requests to the TCPA online services. The URM requests are tracked through the user's account on the platform. The user is considered as trusted, meaning that it is assumed that this user won't misuse their own PII for malicious motives[1]. Additionally, the user is provided with PII management tools through a User Interface (UI), in order to help them understand the potential risks of their PII misuse.
- **The TCPA URM platform**: it is considered to be an Administrative Service Provider (ASP). It acts as a service provider for the user, and relies on the PII manager service. Services provided by this platform can for instance be an online school restaurant registration service. The TCPA URM platform is considered to be a trusted entity with regard to the user.
- **The human agent(s) of the TCPA URM platform**: they are responsible for the processing and validation of requests. Their role is essential as the processing of URM requests requires fine human appreciation, with deep understanding of the technical, functional and legal stakes of the procedure.
- **The Private Service Providers (PSP)**: they are *semi-honest* entities [72], *i.e.*, they do not try to break the system's technical rules, but instead they try to access any data *technically* available to them for reading even though they were not *functionally* meant to be accessed by them.
- **The data sources**: They may be official, *i.e.*, maintained or acknowledged as such by TCPA, or private, *i.e.*, maintained by a third party service provider. We consider three main PII sources defined as follows:

[1] Although, a case of users colluding together for malicious privilege escalation purposes is detailed in Chapter 5.

– The *FranceConnect*[2] official federated-identity service of the French administration that provides online national citizen identities.

– The *DGFIP*[3] PII source, that provides tax information. This information is necessary for instance with some paying services of the TCPA, where the custom fee is proportional to the users' income, which is retrieved through that DGFIP source

– The *CNAF*[4] PII source, that provides information dealing with various social and children allowances. Indeed some requests, like computing school catering custom fees for the users' children, involves the retrieval of some of this allowance information.

2.2 The Territorial Use Case

2.2.1 The Territorial Services Offered to the User

The main use case considered in this chapter is the registration of the user's children to the school restaurant of their territorial collectivity in France. With regard to our use case, these territorial collectivities are responsible for the children registration to schools and school restaurants that belong to their territory. Such collectivities usually provide an online service for parents to register children and pay the school restaurant fees. As defined in French collectivities, the school restaurant fees depend on the parents' fiscal situation (and in particular their tax reference revenue document) as well as their children's allowance information (in particular their familial quotient value). Obtaining such information enables the collectivities to define custom and fair school restaurant fees.

As a result, when registering their child to the school restaurant, the parents fill an online form that requires to provide the following information:

1. The child's allowance registration number;
2. The postcode of their current main address;
3. The identification number in the French tax system;
4. The last yearly tax receipt.

While most of the online procedures only support the upload scan documents as PII, this process can be enhanced by (a) providing PII sources from the state administration that offer access to these documents and (b) making this source provide the user's PII in more rigorous formats, such as machine-readable formats.

Now referring to the previous list, items 1. and 2. are required to retrieve the user's children's allowance information from the CNAF attribute source, presented in Section 2.1. Similarly, items 3. and 4. are required to retrieve the user's tax information from the DGFIP attribute source, also presented in Section 2.1. All four items are retrieved through the *API Particulier*[5]. Put in place in 2017 by DINUM[6], this API offers an access to the two aforementioned attribute sources through two different endpoints, accessible for TCPA after registration and retrieval of a client-specific token. As a result, the user also needs to retrieve and manage PII **originating from remote PII**

[2] `https://franceconnect.gouv.fr/` (resource in French).
[3] `https://www.impots.gouv.fr/portail/presentation-de-la-dgfip-overview-dgfip` (resource in French).
[4] `https://www.caf.fr/` (resource in French).
[5] `https://particulier.api.gouv.fr/` (resource in French)
[6] `https://www.numerique.gouv.fr/dinum/` (resource in French)

sources: this PII is necessary while using the administrative and private service providers[7]. The sources for this PII may be multiple, but the management features offered by this service remain unchanged.

This PII is used by various TCPA services possibly belonging to various collectivities at different scales–town, department, region, country.

The user, as a citizen within a TCPA, also interacts with the TCPA URM platform available to them, for instance for a passport renewal request, as well as for registering their son in an elementary school.

2.2.2 The Need for Identity Matching

Generally, when completing online procedures, citizens are expected to prove their identity. For this purpose, the registration form enables the user to log in using the *FranceConnect* federated-identity service.

On the contrary, when that identity federation service is not used by users while filling the form, they are instead asked to provide a scanned copy of an official identity document (*e.g.*, their identity card, driving license or passport). In this case, a TCPA (human) agent validates the authenticity of the scanned document.

The *FranceConnect* federation service is a digitization of the users' identity–it therefore replaces the users' identity (paper) documents. The governmental decision of November 8^th, 2018[8] allows the TCPA to use the *FranceConnect* service instead of asking a scanned copy of the users' identity documents. Whenever the TCPA needs to obtain a valid user identity with a high level of trust, *FranceConnect* providers with eIDAS levels higher than one are involved. Regardless of the multiplicity of *FranceConnect* providers, the *FranceConnect* service is responsible for performing identity reconciliation. For instance, when choosing the *DGFIP* provider for the authentication, the identity served might differ from the identity which the *CAF* provider would serve: it is the *FranceConnect* service's responsibility to re-conciliate the user's identity in spite of these variations.

This reconciliation gives the confidence that, regardless of the *FranceConnect* provider used by the user, it is always the same user identity that will be served to the TCPA URM platform. According to the concept of *sector* in the OpenID Connect protocol, two different TCPA URM platforms belong to two different sectors *per se*, and are given two different user identifiers for a same user. This sector separation ensures that two different platforms do not collude in order to illegitimately obtain knowledge about their users.

In both cases the user identity needs to be validated as properly matching with the different PII provided by the sources. Consequently, the subject of the identity given by *FranceConnect* must be verified to be the owner of the PII provided by the DGFIP and CNAF sources, respectively. Failing to do so may allow different subjects to collude, thus obtaining illegitimate advantages from the service. This validation, being in fact an identity-matching procedure, is necessary to the completion of the citizen-relationship management process. This identity matching, either automated or performed manually by the agent, is described later in Section 5.3 for user PII provided by *FranceConnect* and the *API Particulier* endpoints.

[7] *Private* as these service providers are not under the TCPA's authority.
[8] See `https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000037611479`.

2.2.3 User's Documents and PII

In our use case the relevant Personally Identifiable Information (PII) can be a set of official documents such as the user's family register, or even identity documents such as the user's ID card or a driver license. Moreover, this PII can also be raw personal data managed by the TCPA, such as a string describing the user's postal address in a well-known format, or geographical information about the user at a given time. Eventually, the metadata generated and linked to this PII is also considered to be Personally Identifiable Information.

Scanned documents and third-party-issued PII enable the user to provide the TCPA with *validated* data as valuable input for processing their requests. The *FranceConnect* official identity service provides user identity information, complying with the OIDC [83] identification protocol. Implicit grant OAuth 2.0 [36] authorization is also at experimental stage for tax and children allowance information.

Without sources in the environment, the user has to provide scanned documents in order to submit their URM requests. In this case, manual validation tasks are required by the (human) TCPA agent for the completion of the request. Alternatively, when the URM environment includes sources that are able to provide structured PII, the process of completing the user's request can be greatly simplified.

From the user's point of view, said user also needs to define a validity time window for the authorized access to apply on their documents and PII. They can also authorize their relatives to use some of their documents to Administrative or Private Service Providers.

For convenience, the user also needs to define the set of service providers that will later be able to access these two documents without asking them later for proper synchronous (hence blocking) permission (Private Service Providers, PSPs, authorized to access the documents will not wait for the user's permissions, as they already obtained their consent). The private connected services that are supported in our use case can be services providing subscriptions to local sponsored events or local business offers. Although the user registering their children to school catering doesn't require the write access of the TCPA on the user's PII, some URM requests do. Therefore such write access should be supportable.

For this purpose, user-centric architectures that provide consent management aim at allowing the user to have the governance of their PII through their lifecycle, that is, the consent to collection, the usage control over time and the visibility of past collections. These capabilities should happen even when the PII has been provided by third-party sources.

Eventually, in the European Union, functional requirements regarding user data management in TCPA have recently changed as part of the international regulations. Those strategies must address the need to give online users a thorough governance of their Personally Identifiable Information (PII)–whether these PII be, for instance, service-provider transactional data or user profile data[9].

These requirements address the *Tell us once* French interministerial program. In order to comply with the GDPR, the user must be able to restrict consent to a specified set of purposes for their PII. For a better user experience, the user might be proposed a set of purpose categories, *e.g.*, administrative procedures, subscription to sponsored events, registration in local associations.

[9] Additionally, some TCPA still partly rely on scanned documents.

Also, once a service provider is revoked, this provider must not illegitimately access the user's PII anymore.

2.3 The Functional Requirements Implied by the Territorial Use Case

The use case results in the following functional requirements:

- **Purpose-based authorization** definable by the user.
- **Service provider revocation enforcement** by the user.
- **Time-based authorization validity** defined by the user for any given authorization.
- Ability to ensure that the PII provided by the available sources for a given user are **related to the concerned user** and not anyone else. In particular, the solution must prevent the collusion cases illustrated in Figure 2.2.
- **Provability** of a user's identity through the information provided by the available sources.
- **Detectability** of the numerous true positive cases of identity-matching in an autonomous manner.
- **Collectability** of (either true or false) negative cases, as well as ambiguous cases, that need thorough inspection by a TCPA agent.
- **Usage definition**, meaning that the user can define the purposes justifying the PII collection.
- **Consent management**, *i.e.*, the PII manager keeps track of the authorizations given by the user for each piece of PII.
- **Usage monitoring**, meaning that the user is given clear metrics of the PII consumption by any TCPA service. This monitoring facility offers a view of the user's PII usage by the TCPA services.
- **Delegation capabilities** for the PII manager. This PII manager is able to decide whether or not to grant access to the PII, even when the user is not connected to the platform. During that partially-autonomous decision making, the access granting process is asynchronous from the authorizations granted by the user.
- **Interactions with remote PII sources** through the user-relationship platform.
- **PII location abstraction** for the PII manager regardless of the original source of the PII.
- **Protocol standardization** through standard interfaces, *i.e.*, the PII manager can be queried with a common interface relying on standard PII management protocols.
- **Access uniformization**, for the multiple PII data sources which are then accessible in the same way.
- **Authorization protocol interoperability**, with the main identity management protocols–access mechanisms and authorization schemes. These protocols are supported with the heterogeneous remote sources to achieve interoperability.

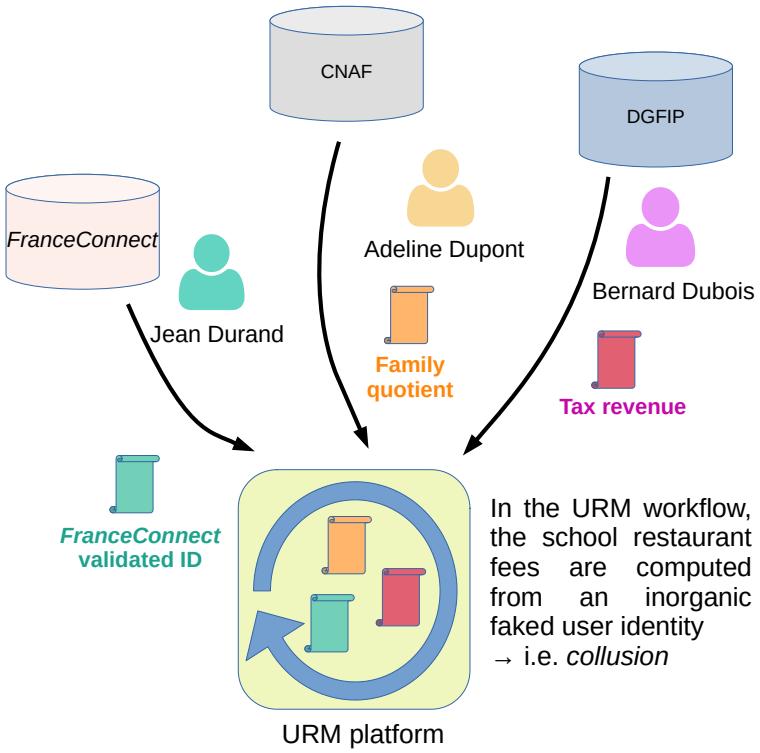


Figure 2.2: An illustrated classic collusion case

2.4 The Industrial Context at Entr'ouvert

2.4.1 Presentation

Entr'ouvert is a software development cooperative specialized in User-Relationship Management (URM) and more particularly in Citizen Relationship Management (CRM)[10]. Its customers are the TCPA that propose online services to their citizens, like the use case described in Section 2.2.

The URM software suite, named Publik, is based on a modular architecture. This software suite relies on the Django web framework[11], itself written in the Python programming language[12].

Entr'ouvert also aims at releasing free software only, and more particularly under the AGPLv3 (Affero General Public Licence in its third version).

Eventually, Entr'ouvert also offers *Software as a Service* and manages the biweekly software releases of Publik.

2.4.2 Functional Modules

A non comprehensive list of Publik's functional modules is its:

[10] A common use of the acronym CRM is also *Customer-Relationship Management*, a commercial concept which involves a completely different set of technologies; the latter will not be covered in this document.
[11] See `https://www.djangoproject.com/`.
[12] See `https://www.python.org/`.

- form and workflow edition software, *w.c.s.*[13],
- identity management, *authentic*[14],
- appointment management, *chrono*[15],
- content management,*combo*[16],
- document management,*fargo*[17] and
- statistics generation,*bijoe*[18].

These functional modules require an authenticated access. The identity management tool, *i.e.* *authentic*, performs Single Sign-On (SSO) and Single Logout (SLO) thanks to the Security Assertion Markup Language (SAML) protocol.

2.4.3 Technical Modules

Publik also has a main technical module *hobo*[19], that holds the functional modules altogether by performing user accounts and roles provisioning, as well as taking part in multi-tenancy management and automates instances deployment. Additionally, the *django-tenant-schemas* software library extends the PostgreSQL schema model for applicative multi-tenancy purposes.

Figure 2.3 provides a schematic description of applicative tenants and database schemas that enable our base architecture.

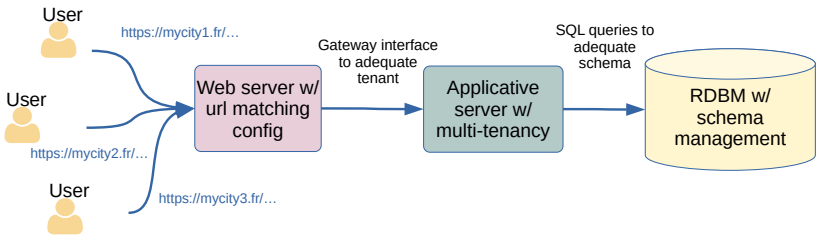


Figure 2.3: Applicative tenants and data schemas as part of our architecture

2.4.4 Main System Administration Capabilities

The servers are administered:

- through ssh (secure shell) [52] access, relying on public-key cryptography;
- by the generation and deployment of Debian[20] packages, visible at `https://deb.entrouvert.org/`.
- thanks to supervision tools, deployed to ensure any upcoming issues threatening high-availability, *e.g.*, memory shortage, lack of disk space, network excessive latency or unavailability.

[13] See `https://dev.entrouvert.org/projects/wcs`.
[14] See `https://dev.entrouvert.org/projects/authentic`.
[15] See `https://dev.entrouvert.org/projects/chrono`.
[16] See `https://dev.entrouvert.org/projects/combo`.
[17] See `https://dev.entrouvert.org/projects/fargo`.
[18] See `https://dev.entrouvert.org/projects/bijoe`.
[19] See `https://dev.entrouvert.org/projects/hobo`.
[20] See the Debian operating system project page: `https://www.debian.org/`.

2.4.5 Software Development Pipeline

The software development pipeline starts with the identification of a new feature or of a software bug. Identifying a bug or a new feature can either be performed by technical referees or project managers at Entr'ouvert. Alternatively–and following the free software principles–any community member can identify a bug or a new feature deemed useful. In any case, specification of the feature or the bug happens on Entr'ouvert ticketing platform, `https://dev.entrouvert.org/`[21].

Then the software development process in itself starts with the deployment of a local instance on the developer's computer. That local instance is deployed thanks to a dedicated Ansible playbook[22], visible at `https://git.entrouvert.org/publik-devinst.git/`.

Integrating a new feature into the existing software starts with its development on a separate git branch[23]. Aside from developing the feature itself, thorough unit testing coverage is expected. A continuous integration platform, checking the absence of software regression, is accessible to the developer at `https://jenkins.entrouvert.org/`.

Having the new code, corresponding to the upcoming feature, integrated into the main code base requires peer review. The peer review process happens on the ticket platform, and may require several rounds of modification to the submitted code changes before having it accepted by the peer reviewer(s). The upstream code base for each Publik module is visible at `https://git.entrouvert.org/`.

Upon integration of the code changes into the main code base, tags may be applied to said code base as milestones of code modification. In this case, dedicated scripts generate a new Debian package, visible at `https://deb.entrouvert.org/`.

Along with the generation of a new Debian package, any upcoming feature is documented in the open-access official documentation, at `https://doc-publik.entrouvert.com/`.

The Debian package, corresponding to a new software version that includes the upcoming feature, is first installed on a test platform. On that test platform, collectivities using the Publik software can try out the new features–potentially detecting new bugs and starting over a new development pipeline cycle. After a week on the test platform, the Debian package is then installed on the production platform. The test- and production-platform upgrade frequency is bimonthly. Keeping track of the versions is actually installed on the platforms is visible at `https://scrutiny.entrouvert.org/`.

2.4.6 Protocol and Data Format Hypotheses

The PII formats used in the PII exchange and retrieval protocols are (mainly) JSON–for OAuth, OIDC, REST– and XML–for SAML. We take the hypothesis that such nomenclatures are used for every given type of PII. These nomenclatures are already in use in TCPA environments. For instance, *date* and *datetime* information rely on ISO 8601 [96] and its use on the Internet, as covered by RFC 3339 [66]. Similarly, the standardization of phone numbers as URIs is covered in RFC 3966 [85].

[21] Which is itself an instance of the Redmine free software project, see `https://www.redmine.org/`.
[22] See `https://docs.ansible.com/ansible/latest/index.html`.
[23] See `https://git-scm.com/`.

2.5 Security Hypotheses

Stating the main problem also comes with the security hypotheses of the solution. These security hypotheses are strongly linked to the technologies used either with the identity-matching procedure or more generally with the citizen-relationship management software environment itself. These hypotheses are listed below:

- Server-side SSL/TLS authentication is used, as for usual Web technologies.
- The identity information is provided by the *FranceConnect* service according to the OIDC identification layer of the OAuth 2.0 protocol.
- The DGFIP and CNAF endpoints are restricted: they are accessible after registration only, which involves per-case validation. The endpoints expose read-only resources according to the Representational state transfer architecture style [30].

Eventually, this thesis assumes that the different acting entities' clocks are loosely synchronized, which is common and considered easy to achieve.
Additionally, it assumes that the TCPA URM platforms do not permit the same identifier to be assigned to several users time after time. This is a loose requirement as most of the identifiers are either reversible or irreversible high-entropy pseudonyms, where reversible pseudonyms rely on symmetrical cryptographic functions, whereas irreversible pseudonyms rely on hash functions. In both cases, provided that the user PII being used as input to the pseudonym function varies, the risk of collision is considered negligible.
More generally, identity management protocols such as SAML and OIDC let the administrator choose their underlying cryptographic primitives, depending on their actual implementation. Thus we consider that the primitives chosen are the ones commonly considered as secure, for instance by the Internet Engineering Task Force (IETF)[24] and the Internet Research Task Force (IRTF)[25].

2.6 Conclusion

In this chapter, we defined the actors, the use case and the functional requirements. We specified the scientific subject in its industrial environment and its subsequent hypotheses.

The clear and thorough definition of the problem is paramount as it bears an impact on all the collectivities–*e.g.*, on a national level, around 36.000 municipalities in France. The problem could also be stated with many other URM procedures. For instance, more than 300 procedures are offered by the Publik production platform.

The next chapter deals with the existing solutions that answer, in an at least partial manner, to the use case along with its requirements, while respecting the main hypotheses.

[24] See `https://www.ietf.org/`.
[25] See `https://irtf.org/`.

Chapter 3

User-Centric Personal Data Management Solutions

This chapter presents a survey of technologies for personal data self-management interfacing with TCPA service providers. These services are fed with significantly sensitive PII (users' name, address, family status, allowances, tax status and so on). Additionally, these services are interacting with other Private Service Providers (PSP): they are indeed used to ensure URM with services offered by third-parties.

In order to achieve this technological survey, this chapter offers a classification of academic and industrial solutions into four categories: Personal Data Store (PDS), Identity Manager (IdM), Anonymous Certificate System and Access Control Delegation Architecture.

Each category, along with its technological approach, is analyzed thanks to fourteen functional criteria that encompass architectural and communication aspects, as well as user data lifecycle consideration.

Eventually, the outcome of this chapter is the clear identification of functional gaps of each solution and, more generally, for each category of solutions. As a result, this chapter establishes the research directions to follow in order to fill these functional gaps.

This chapter is organized as follows. Section 3.1 describes the system model along with the main use case, illustrating PII self-management. Section 3.2 identifies the fourteen differentiating criteria selected for this comparative survey. Section 3.3 introduces the four categories of solutions selected for the survey. The main technical background for each solution is also described. The solutions share the common objective of providing Web users with informational governance tools, but they differ by the scope of supported functional mechanisms. These solutions stand out by the way they deal with PII management, each one offering its own functional mechanisms. Section 3.4 evaluates each of the four categories of solutions according to the fourteen criteria that were identified in Section 3.2. Section 3.5 gives a synthesis of the survey. It identifies the functions that remain unsupported so far. Eventually, Section 3.6 concludes this chapter by defining an optimal solution and stating the research directions that would fill the functional gaps identified in Section 3.5.

3.1 Fundamentals of Personal-Data Management According to our Use Case

3.1.1 Our System Model Selected for the Survey

First of all, our survey relies on the generic architecture of Figure 3.1. This figure includes the relevant actors as presented in Section 2.1 and provides a generic basis for the four specific categories.

In this chapter, we refer to service providers that may be either administrative (ASPs) or private (PSPs). Whenever that distinction between administrative and private service providers is not necessary, the acronym SP (Service Provider) is used.

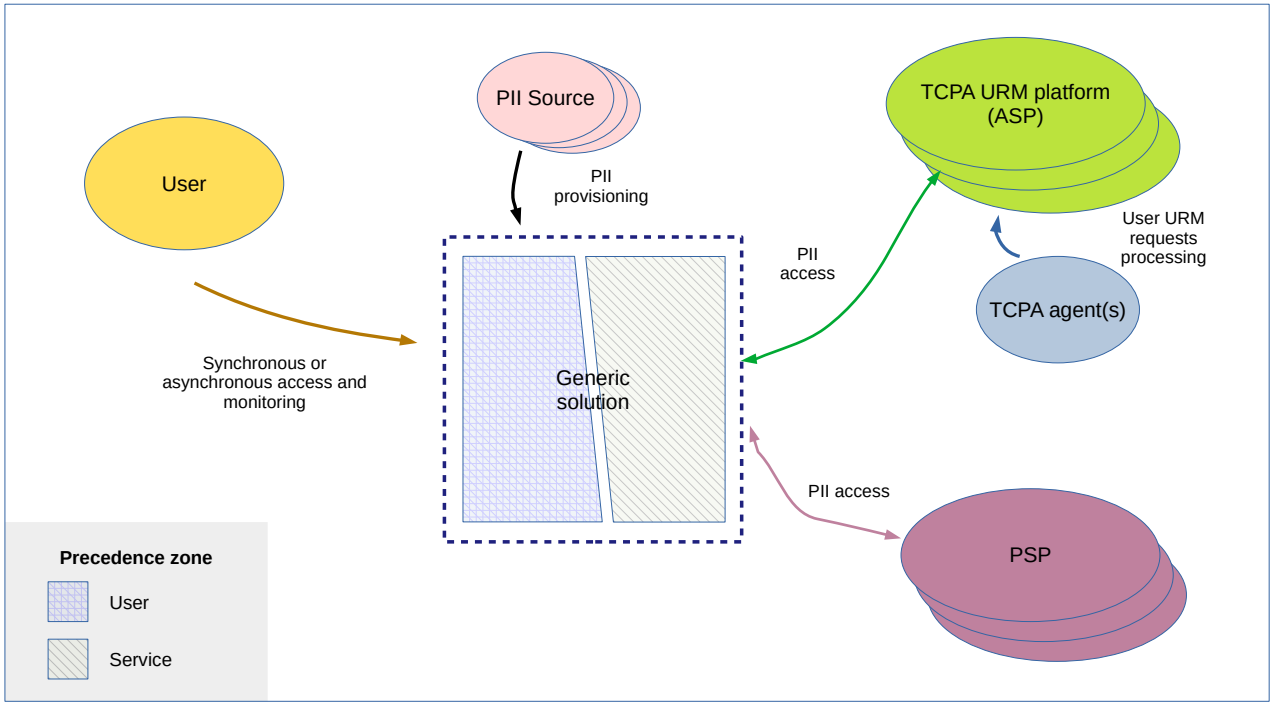


Figure 3.1: Generic architectural layout diagram

3.1.2 Preliminary Definitions

Two core concepts of PII management are defined as follows.

- **Access control** is the enforcement of rules enabling only authorized services to access user data. Several access control models have been presented in the literature. These models vary a lot, for instance relying on properties of the requester, or even on the behavior of this requester on the system over time. [93] provides an accurate description of the most common AC models.
- **PII provisioning** is the process of creating or updating data on the user's PII storage base, while *PII deprovisioning* is the processing of deleting such information. SPs might be empowered with both provisioning and deprovisioning capabilities.

3.2 Selected Criteria for the Territorial Use Case

Tables 3.3 and 3.2 give the list of criteria of interest for TCPA service providers, **among which five, in bold type, are considered as critical** for the territorial collectivity use case (see Section 2.2). This set of criteria is split into three different categories.

Criteria belonging to the first category deal with user consent management on their PII, *i.e.*, the ability for users to manage their PII during their whole lifecycle.

Additionally, criteria identifying properties in the data exchange flows that are prone to enforce PII management in a privacy-compliant manner belong to a second category, named *data exchange flow criteria*.

Eventually, other criteria, dealing with user governance over their PII, are also taken into consideration. By *governance* we mean the user's right to control which data processes are applied to their PII. This governance happens regardless of the user's actual ownership of their own PII.

Table 3.1: Consent management criteria

Criterion
Type of user consent
Type(s) of supported access-control
PII collection purpose definition

Table 3.2: Data exchange flow criteria

Criterion
Type(s) of supported PII
PII validation
Provisioning and deprovisioning management
Re-usability of previously uploaded PII
Minimization management
Support of remote PII sources

Table 3.3: Misc. user governance criteria

Criterion
Privacy usability trade-off
User interface
Service provider revocation
Extent of delegation
History/logging of transfers

Some criteria are *qualitative* rather than *quantitative*, hence the evaluation scale has to be explicitely given for the reader to understand the evaluation according to this criteria. For such criteria, a subsection titled *Evaluation scale* is added in the criterion presentation.

3.2.1 Consent Management Criteria

3.2.1.1 Type of User Consent

3.2.1.1.1 Presentation

Evaluating solutions according to this criterion implies the study of the user consent for these PII exchanges. Thus, in this particular context of potentially sensitive PII collection for later processing, the consent should be perceived as a grant to a service given by the user.

3.2.1.1.2 Reason for criterion selection and criticality

This criterion is tagged as critical as it is a fundamental part of user's privacy guarantees: it supports the margin of decision that the user has upon the transfer of their PII from, or to, SPs. Indeed, according to our territorial use case, in Section 2.2.3, the user's PII may be used by ASPs as well as PSPs, for various purposes, and therefore require proper consent management.

3.2.1.1.3 Underlying technical challenge

The technical challenge for this criterion is the ability for the solution to handle a consistent consent model, suitable for the administrative use case. In other words, the consent model should be suitable for managing PII all along its lifecycle and tackle the subset of services (*e.g.*, social and family-related services, health-related services and civil procedures) that the public administration provides. From a technical point of view, it means that the consent model has to bear (i) *spatial* information, *i.e.*, the extent to which the consent is given and the services allowed to access the user's PII, as well (ii) *temporal* information, *i.e.*, the time at which the consent was given, its expiration date, and possibly version numbers of the end-user agreements for the services being granted authorizations.

3.2.1.1.4 Evaluation scale

This non-quantitative criterion consists in identifying the global user consent process, the PII involved in enforcing user consent, and, depending on the solution, any other element that facilitates the enforcement of the territorial use case.

3.2.1.2 Type(s) of Supported Access Control

3.2.1.2.1 Presentation

This AC criterion evaluates how easy, flexible and robust the solution is for the user to ensure the application of access rules on their PII. These access rules restrict the way their PII is shared with ASPs and, most important, with PSPs (considered less trusted). When correctly defined and applied, AC ensures that only legitimate SPs have access to the user's PII.

The evaluation according to this criterion requires, to some extent, the understanding of the underlying security mechanisms and models. Indeed, these mechanisms and models shape the definable access control policies. These policies can take the form of an access control list and they can bear more flexible options like contextual information, delegation to an access control agent and cascading authorizations.

Generally speaking, AC encompasses both the user-side and the SP-side when applying AC rules. However, as users are considered trusted in our system model–see Section 2.1–, the evaluation

of solutions according to this criterion must focus on the SP access control.

3.2.1.2.2 Reason for criterion selection

This criterion directly addresses the extent of possibilities for the user to define access control over their PII, thus it complies with the territorial use case as seen in Section 2.2. The access control capabilities indeed have a significant outcome on the ability for the user to manage their own PII when interacting with their TCPA service providers.

3.2.1.2.3 Evaluation scale

Evaluating solutions according to this criteria means determining how expressive the access control model is in order to enforce our use case.

3.2.1.3 PII Collection Purpose Definition

3.2.1.3.1 Presentation

This criterion is functionally close to the previous one, described in 3.2.1.2. It performs a focus on the way the access control capabilities help the user set the authorized purposes for which their PII can be collected. It doesn't offer the study of the way access control is applied once PII collection purpose has been defined–that is the object of 3.2.1.2.

For instance, with this criterion, the user may want that a copy of their ID card be limited to administrative purposes only, or, on the contrary, extended to private services. The solution should address this requirement.

Additionally, we consider the agreement–between a user, owning their PII, and an SP requesting PII–to be a contract. The criterion should help evaluate whether parties are able to negotiate the terms of the agreement before the agreement is met. If so, the user and the SP should be able to define, in an interactive way, the conditions of PII collection. This means that the solutions should enable the user to define which PII a service may be able to collect, and for which purpose.

3.2.1.3.2 Reason for criterion selection

The ability for the user to restrict the PII collection to a specific and explicit set of purposes is a legal requirement. Purpose-based authorization is also a convenient access-control abstraction, easier for the user than per-case SP authorization.

3.2.2 Data Exchange Flow Criteria

3.2.2.1 Type(s) of Supported PII

3.2.2.1.1 Presentation

The three following types of PII are considered:

- Structured **documents**, *e.g.*, a digital copy of a family register.
- Raw **data** (usually under the form of non-document data formats, such as XML or JSON, or plain text), *e.g.*, the user's postal address or date of birth.

- **Metadata**, *i.e.*, attribute or value metadata, for instance as specified in NIST internal report 8112 [34] for federated identity systems.

These types of PII lead to support several possible use cases. For instance, the solution may be used as a platform for storing personal and secure storage of documents owned by the user.

With metadata, a higher abstraction level is provided, thus leading to a number of possible use cases including validation based on data content, time-to-live metrics, data type, identity, *etc.*

3.2.2.1.2 Reason for criterion selection

The supported PII types are significant properties as they are of interest for the SP, determining its own usage of PII. They also give a clue of how much governance is left to the user regarding their PII: for instance, a user having access not only to their documents and data, but also to the metadata linked to its collection, has more visibility over the usage of their PII by SPs.

3.2.2.2 PII Validation

3.2.2.2.1 Presentation

Validated data, either it be documents or raw data, might be required by some services. For instance, school restaurant registration, as described in our territorial use case–Section 2.2–may require such validated data.

The objective is to mitigate identity theft by avoiding the use of false or stolen PII. This criterion expresses whether such a validation process is supported by each of the selected solutions. We purposely do not give a precise definition of this validation, as it can encompass several processes, such as the manual validation by a human operator of the TCPA, or even a validation performed by a trusted third-party authority–which isn't the TCPA itself.

3.2.2.2.2 Reason for criterion selection and criticality

This criterion is tagged as critical as it is necessary for collectivities and administrations in order to perform an efficient overall validation of procedures. Indeed, PII validation highly enables partial automation of the procedures, which spares the human agent from repetitive tasks–the latter can in return provide help regarding more complex or non automatable works. Additionally, when the validated PII is structured documents–albeit not enabling automation of the procedures like raw validated PII would do–they may weigh more than non-validated PII in the URM procedure.

3.2.2.2.3 Underlying technical challenge

From an administrative service's point of view, validating PII at different stages of their processing enables different workflows. To some extent, validated PII makes it possible to skip some processing states in the administrative functional workflow. For instance, validated PII may not need a manual verification from administrative agents whereas invalidated PII may. The technical capabilities of the solution must be able to follow the functional scenarios of PII validation. From a technical point of view, this criterion raises the challenges of (i) the origin of the PII (for instance when this PII may come from any of several PII sources) and whether it takes part in determining the degree of validation of this PII; and (ii) whether the solution includes different validation scenarios depending on the lifecycle stage at which the PII is.

3.2.2.3 Provisioning and Deprovisioning Management

3.2.2.3.1 Presentation

This criterion deals with the possibility for PSPs or ASPs to act as a PII source for the user. Such a criterion is particularly relevant in case of administrative or territorial service providers [1]. Such administration (or territorial collectivity) detains sensitive PII describing their users, and their provisioning needs to be managed. This may also be necessary for PSPs.

In order to define and qualify this criterion, we express the following concerns:

- the ability for the ASP or PSP to provision data on behalf of the user.
- if so, the provisioning taking place on the entity directly receiving PII from the user.
- optionally, the precedence of authority (between the user and the SP), regarding the provisioned PII.

This criterion also tackles the management of PII deprovisioning on the solution, *i.e.*, the planned deletion of PII, when reaching the end of its lifecycle. Managed deprovisioning involves compliance with a main concern of data protection, the "*right to be forgotten*", defined by the recent European regulation as the right to erase its own PII.

It means that the user should be able to make some PII be unusable to SPs any longer–the most obvious way to do so being PII deprovisioning on these SPs.

3.2.2.3.2 Reason for criterion selection

Provisioning and deprovisioning capabilities are a direct part of the use case as they enable SPs to perform operations on the user's PII after obtaining their consent to do so.

3.2.2.3.3 Evaluation scale

Provisioning and/or deprovisioning features may be supported in different manners depending on the category of solutions. For instance, the extent of provisioning varies from plain user PII to user accounts. Additionally, provisioning may be supported in a very partial and incomplete way for some solutions. We believe that a brief evaluation of the extent of provisioning and/or deprovisioning features for the selected solutions give a relevant scale of the ability for these solutions to address the territorial use case.

3.2.2.4 Re-usability of Previously Uploaded PII

3.2.2.4.1 Presentation

This criterion means that the user can reuse some PII previously uploaded or provisioned, in order to fulfill another later processing. For instance, after uploading a copy of their family register so as to register their son in primary school, the user may have to use this document again in order to register their daughter in the municipality's day nursery (see Section 2.2). This also avoids that the user loads the same PII twice into the system.

[1] The French *Unique Reglementary act* RU030 defines a set of public online services for which the collection and provisioning of some PII by TCPA is legitimate *de facto*. This text of law acknowledges the importance of such services of public interest, while ensuring that no unnecessary data is collected or provisioned. See `https://www.legifrance.gouv.fr/loda/id/JORFTEXT000027697207/2021-01-19/` (*French resource*).

3.2.2.4.2 Reason for criterion selection and criticality

The *Tell us once* program is a core element of the use case (see Section 2.2.3), as it is directly relevant to the re-usability dimension. This is a critical criterion.

3.2.2.4.3 Underlying technical challenge

The technical challenge of PII re-usability involves determining when the conditions for such re-usability are gathered. These conditions are:

1. Enforcing user consent. In other terms, this challenge implies studying whether the evaluated solutions support PII re-usability without undermining user consent.
2. Applying the legal framework of PII processing, as stated by the GDPR. In other terms, we need to know whether the re-usability is adaptable to the legal framework applied in the territorial collectivity.
3. Applying rules specific to the context of administrative services. In other words, this criterion implies identifying the solutions supporting variations in such rules.

3.2.2.5 Minimization Management

3.2.2.5.1 Presentation

Respecting this criterion guarantees that the ASP and PSP are collecting PII in a minimized way, *i.e.*, only the PII strictly needed for the data processing as declared by the SP. For instance, a PSP only requiring the user's name and postal address should not access the entire content of their ID document.

3.2.2.5.2 Reason for criterion selection

The principle of *finality* of the European GDPR results in the minimization of the PII collected by the ASPs and the PSPs. It is a legal requirement, but is not considered functionally critical for the territorial use case.

3.2.2.6 Support of Remote PII Sources

3.2.2.6.1 Presentation

Users managing their PII may want the solution to abstract the differences appearing while managing remote PII sources. As a result, supporting this criterion ensures that the solution provides such an abstraction regarding the PII location. Whether the PII is locally stored on the solution or remotely accessible on remote sources shouldn't hinder the user in their PII management.

This criterion is tagged as critical as some procedures in our use case require PII that are available on remote sources only. The use case 2.2 involves the use of remote PII sources, for instance the ones supplied by the French administration, such as *FranceConnect* identity providers [2], the DGFiP (central fiscal system) data source and the CNAF (children and family allowances) data source.

[2] See `https://franceconnect.gouv.fr/`.

3.2.2.6.2 Reason for criterion selection and criticality

The support of remote sources is considered critical as it is at the core of the use case relying on several sources maintained by official authorities at national and European levels.

3.2.2.6.3 Underlying technical challenge

The underlying technical challenge involved in the support of remote PII sources is threefold:

1. Consent management must be applied in a consistent manner regardless of PII origin. That is, the consent model enforced must be compatible with all the sources.
2. Sources trust management must be enforced, as some sources may be more reliable or more honest than others. For instance, some PSPs may act in a honest-but-curious manner, gathering information that was not meant for them to collect.
3. Interoperability concerns arise when dealing with several sources, with potential variations in access and authorization protocols.

3.2.3 Misc. User Governance Criteria

3.2.3.1 Privacy Usability Trade-Off

3.2.3.1.1 Presentation

This criterion defines whether privacy enforcement happens at the expense of the solution's usability. Usability and privacy are sometimes viewed as contradictory objectives in identity management, hence the trade-off. However, depending on the nature of the solution, such a trade-off can be adjusted, softened, or even sometimes avoided. For example, some locally-deployed user-centric solutions–for instance on user hardware–may enforce privacy without degrading the usability of these solutions (*e.g.*, due to proximity of the user and the deployment on trusted hardware).

User privacy is enforced towards the SPs. For some selected solutions, user privacy also needs to be enforced towards the solution–when, under some security hypothesis, it cannot be trusted.

This criterion raises several concerns. First, in order to use the solution fully, the users should understand the privacy-enforcement mechanisms and the reason(s) for the trade-off, if any. Second, they should know their privacy-related rights while using the services–and the solution should guide them to the enforcement of these rights. Third, an adjustable privacy-usability trade-off might be of interest, but then remains the question of identifying which actors of the use case would be able to configure or adjust that trade-off.

3.2.3.1.2 Reason for criterion selection

The solutions enforcing the territorial use case are managed by users. Respecting privacy properties must be essential to them, but the enforcement of these properties should not come at the price of a degraded usability. Indeed, a significant trade-off reduces the solution's usability, defeating the use case *de facto*.

3.2.3.2 User Interface

3.2.3.2.1 Presentation

This criterion measures how efficient and user-friendly the user interface is. It involves studying the understandability of the supported functionalities by the users, that is how it helps users understand how their privacy is ensured.

3.2.3.2.2 Reason for criterion selection

The fact that the selected solutions are user-driven of course implies the presence of a user interface. This user interface directly conditions the possibilities of empowerment of the user when dealing with the solutions. Studying the UI for each solution is therefore necessary.

3.2.3.2.3 Evaluation scale

Evaluating solutions according to this criterion consists in an appreciation of the elements of user interface that help users manage their PII. Of course, due to the varying nature of the evaluated solutions, these elements may vary from a category of solutions to another, or may not even be identifiable at all. We still believe that identifying such elements is relevant in order to identify an optimal solution.

3.2.3.3 Service Provider Revocation

3.2.3.3.1 Presentation

The enforcement of PII access revocation for a SP should be implemented. For instance, a user having previously granted consent to a PSP should be able to revoke at any time the possibility to collect their PII. Regarding this example, the granted access could be given to the SP in order to validate the user's registration to a local sports event. Once the sports event is over, the user should be able to revoke the PSP authorized access to the copy of the user's PII. The PII management system should provide the user with a simple way to perform that operation.

This criterion is technical. It excludes the legal means that an authority can exert in order to have user PII be deleted by a PSP.

3.2.3.3.2 Reason for criterion selection

On top of being an obviously-expected feature, the ability to revoke service providers is a legal requirement: according to the GDPR, and more especially its right to oppose and its right for correction, the user must be able to change previously-given consents at any time.

3.2.3.4 Extent of Delegation

3.2.3.4.1 Presentation

The autonomous behavior of the solutions consists in authorizing, on behalf of the user (whether they are online or not), some SPs to perform operations on elements of the user's PII data base.

Therefore, the user may want to define the degree to which the PII management solution should act on their behalf. The user should be able to require the solution to ask for their explicit consent each time an SP wants to access some of their PII. Conversely, the user could decide that

the solution acts autonomously on their behalf once a delegation policy has been set.

This criterion deals with *temporal* autonomy of the selected solutions, *i.e.*, allowing the study of asynchronicity properties of the selected solutions. It addresses the user's control over their PII exchanges, whether the user is "present" or not, when these exchanges happen.

It also deals with these solutions' *spatial* autonomy, *i.e.*, the ability for the selected solutions to act on behalf of the user at different steps of the management of their PII and for different set of services.

3.2.3.4.2 Reason for criterion selection and criticality

Ensuring this delegation means finding the right balance between assistance and consent, *i.e.*, the user should be assisted by the delegation process to offload their decisional tasks, while the user's consent should be fulfilled. Eventually, this criterion is tagged as critical for our use case, as a consistent delegation model is necessary to take care of PII transfers on behalf of the user. As depicted in our use case 2.2, the purposes for ASP or PSP may vary a lot. Instead of defining fine-grained management of their PII, users may prefer to define delegation policies for the system to next adapt their behavior when dealing with ASPs and PSPs.

3.2.3.4.3 Underlying technical challenge

The technical challenge of these modes is providing the user with the ability to define which PII processing should happen when the user is offline. The consent model proposed by the solution should be able to apply the user's choices regarding offline PII processing. From a technical point of view, the *spatial/temporal* distinction defined in Section 3.2.1.1 still applies for the extent of delegation. The delegation operates spatially. As a result, the following questions must be answered when studying the solution according to this criterion:

- Does the solution provide a categorization of services, allowing generic consent to be given by the user?
- Similarly, is PII categorization supported by the solution?
- Does the solution define fine-grained types or scopes of actions for the consent given by the user?
- Does the solution enforce traceability of operations, especially while the user is offline?

3.2.3.5 History/Logging of Transfers

3.2.3.5.1 Presentation

This criterion enforces the "right to be informed" and enables the users to access the history of all executed transfers, with several possible granularities, *e.g.*, an access to the PII metadata only, or to the full PII content.

It raises the following concerns:

- the need for the user to understand which PII have been exchanged, and when it happened.
- the quantity of information revealed by the log or history facility about the content of the PII exchanged.

3.2.3.5.2 Reason for criterion selection

The logging of transfers is also considered as a legal requirement as stated by the GDPR.

3.3 Taxonomy of Academic and Industrial Solutions for User-Centric Personal-Data Management

Thirteen solutions are classified into two families and four different categories, as depicted in Figure 3.2. The first family of solutions *i.e.*, self-managed software implementations, encompasses monolithic software entities that provide users with PII management capabilities.
On the contrary, the second family of solutions *i.e.*, access management architectures, is set at a higher architectural level, and can't be identified as a single software tool. The interactions among several entities of solutions of that latter family enable PII management capabilities for users.

We have retained these two levels (monolithic software implementation, and full architecture) in our survey as the territorial use case does not favour one of the other. Therefore solutions belonging to these two levels benefit from being part of the survey. Of course, software implementations can be part of access management architectures. For instance solutions belonging to the Identity managers category can act as authorization servers in a UMA architecture (twelfth selected solution). However, while in the first family of solutions the standalone software implementation is studied, in the second family the full architecture is studied.

These two families then form a complete partition of PII management solutions. Each of these families is split in two categories for further disambiguation. The four resulting categories are presented in the following paragraphs.

The solutions have been selected for their detailed presentation in published scientific articles, or for their thorough documentation and code available under a free license. This figure describes both academic (BlindIdM, openPDS, Databox, Idemix, U-Prove and INDIGO) and industrial (Mydex, Fargo, Authentic, OpenIDM, Keystone, Keycloak, and UMA) solutions. Industrial solutions are considered as targeting operational ground, and thus can be expected to cover a wider set of properties of interest than academic ones.

As shown in the zoom-in diagrams of Figure3.1, *i.e.*, Figures 3.4 — 3.7 for the four categories of solutions, these solutions–although covering similar functional objectives–have different architectural and functional natures, with different acting entities. These observations serve to establish our arborescent taxonomy, with the two following identified families.

Family self managed platforms includes two categories:

- Personal Data Stores (PDS), for solutions providing a service for PII storage to the user.
- Identity Managers (IdM), which often act as identity providers in the context of federated identity management (FIM) and single sign-on (SSO). They help the user handle their personal accounts and their associated identity information.

Family access control management includes two categories:

- Anonymous certificates, meant for the user to enforce data minimization, *i.e.*, reveal to the SP only the PII strictly necessary for data processing.

- Access control delegation platforms, meant for the user to define fine-grained authorization on their PII.

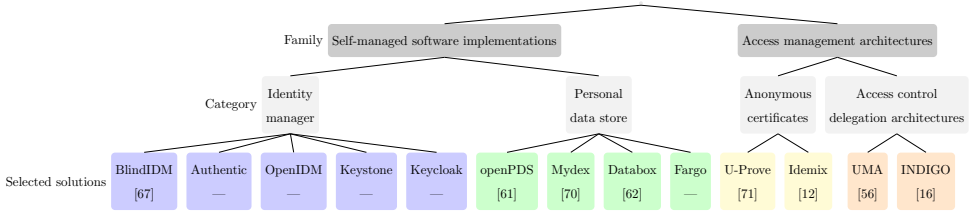


Figure 3.2: Taxonomy of PII management solutions with their categories, partitioned in two main families

3.3.1 Identity Managers (IdM): *(BlindIDM, Authentic, OpenIDM, Keystone, Keycloak)*

Identity managers (IdMs) offer an endpoint for the user-driven management of their digital identities, especially when provided across SPs as shown in Figure 3.4. The cropped arrows and color code of Figure 3.4 refer to the convention and entities in the generic architecture depicted in Figure 3.1. Thus Figure 3.4 should be understood as a zoomed-in version of the generic Figure 3.1.

Depending on the implementation and the configuration, some PII management capabilities are offered to the user while others are left to the administrator only. For example, the user may be able to manage the services requesting their PII, while the administrator may only declare the exact set of services connected to the IdM at any time.

IdMs bear the role of identity providers within a federated identity management [12] (FIM) system, helping the user handle a set of common identities among several registered services. FIM is performed using standardized protocols–either SAML [13] or the OpenIDConnect (OIDC) identification protocol [83] derived from the OAuth 2.0 authorization framework [36]. This framework provides a clear de-correlation of roles when providing access management. These roles are respectively the client, the relying party, the authorization server and the resource server. A system of grant types enables different authorization flows each supporting some specific client types and security hypotheses (*e.g.*, the ability to share secrets, the ability for the client to store its own client secret). Although originally designed as a lightweight protocol–in contrast, for instance, with SAML–, OAuth now specifies richer functionalities such as the dynamic registration of clients [81], the dynamic management of clients [80], protocol interoperability through its assertion framework [14], server metadata [44] and token introspection [77] and revocation [51] endpoints. OIDC also specifies backchannel authentication flows [28], to "compete" with SAML (SOAP-based) artifact-resolution bindings.

The FIM mechanism of interest for this survey is the Single Sign-On (SSO), *i.e.*, the ability for an identity provider to maintain a single authenticated session across several SPs, and its complementary mechanism is Single Logout (SLO).

A basic layout of FIM SSO authentication flow is shown in Figure 3.3, where the client is redirected by the federated service to the identity provider for authentication and for the retrieval of an authorization token. In case of IdMs using exclusively Web technologies, the specified data formats are XML-based or JSON-based, communications are TLS encrypted and the client application is a simple browser.

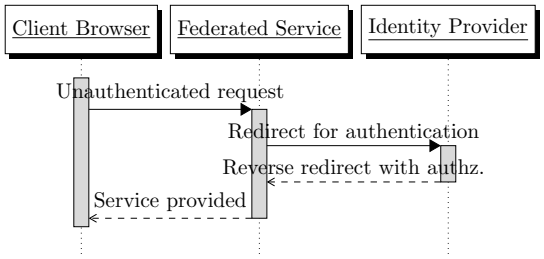


Figure 3.3: FIM authentication sequence diagram

IdMs support a common core of functions, but they have their own management particularities. The basic common core of functions are authentication and user profile management. User profile management involves the ability for the user to set PII profile attributes to be federated over the different SPs registered in the federation. This common core of functions requires that the IdMs support certain set of protocols involving public-key cryptography [23], secret sharing [87] and cryptographic one-way hash functions [84].

3.3.1.1 Selected solutions

This category includes the following solutions: BlindIDM [67], Authentic[3], OpenIDM[4], Keystone[5] and Keycloak[6].

BlindIDM is an IdM whose deployment is meant for untrusted (semi-honest) servers. As a re-encryption proxy–as for instance presented in [8]–, it handles PII without reading its content, thanks to a set of asymmetric re-encryption keys. These re-encryption keys are used to translate a ciphertext encrypted with the user's private key, into a ciphertext encrypted with the SP's private key, with no knowledge of the original cleartext message.

Authentic is an identity provider focused on modularity and extensibility to a wide number of identity management protocols. This modularity is achieved by using the Django Web framework.

OpenIDM, Keycloak and Keystone are three Web identity management servers, presenting similar functional coverage. Keycloak is the main identity management tool maintained by RedHat. Keystone is the identity management layer of the OpenStack cloud-computing framework project. While OpenIDM and Keycloak are thorough generic-purpose identity managers, Keystone is used as an identity-management interface provider among the different services of OpenStack. These three solutions are solving a certain number of well-known concerns of identity management, such as access control, PII provisioning, data reconciliation, password management, and so on. They are designed for a deployment in wide scalable software ecosystems, possibly in federated identity environments.

[3]See `https://dev.entrouvert.org/projects/authentic`.
[4]See `https://backstage.forgerock.com/docs/openidm`.
[5]See `https://docs.openstack.org/keystone/pike/`.
[6]See `https://www.keycloak.org/`.

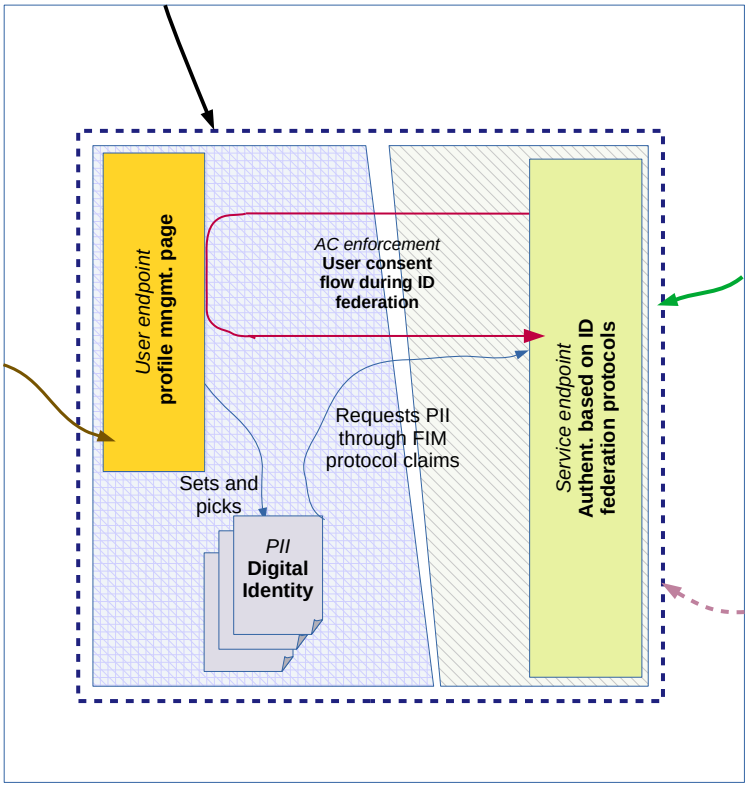


Figure 3.4: IdM architectural layout diagram

3.3.2 Personal Data Stores (PDS): *(openPDS, Mydex, Databox, Fargo)*

PDSs store data on behalf of the user either locally–on user hardware–or not. PDSs are split into a typical set of functional components as depicted in Figure 3.5 The cropped arrows and color code of Figure 3.5 refer to the convention and entities in the generic architecture depicted in Figure 3.1.

The data store is separated from the user front endpoint. The user can access their PII data and documents through the user interface. SPs requests are filtered by an Access Control module which is responsible for granting access, in a partially automated manner (at least), to legitimate providers only. The Access Control module is managed by the user who is typically maintaining a list of legitimate SPs. The user also decides when and how PII can be collected and processed.

3.3.2.1 Selected solutions

The different solutions for this category are: openPDS [61], Mydex [70], Databox [62] and Fargo[7].

openPDS enables the user to adapt the accuracy of the answers provided to the SP. This solution, along with its *SafeAnswers* framework, is designed to receive questions (*i.e.*, algorithms) from the SPs meant to be run locally against the user's PII metadata. Thus no users' raw PII metadata is sent to the SPs. As a result, only the output of the algorithms (considered as the "safe" answers to the questions) is known by the SPs.

[7]See `https://dev.entrouvert.org/projects/fargo`.

Mydex offers classic personal data storage features to the user. It contains a simple access control interface which enables the user to configure their temporal and spatial access control– "temporal" as a time window is defined for the validity of the PII access grant, and "spatial" as a list of registered services is authorized to access the data.

Databox proposes a privacy-preserving interoperable multi-component PDS architecture. The architecture is centralized and defines an *arbiter* responsible for the communications between the components. It is meant to be deployed locally (on user hardware, for instance), with possible access from remote servers.

Fargo acts as a simple document storage server. As a Web application, its modular structure makes it connectable to other components, such as identity providers and data consuming service providers. It is meant for limited straightforward use cases (documents re-usability in user forms) and does not implement complex features.

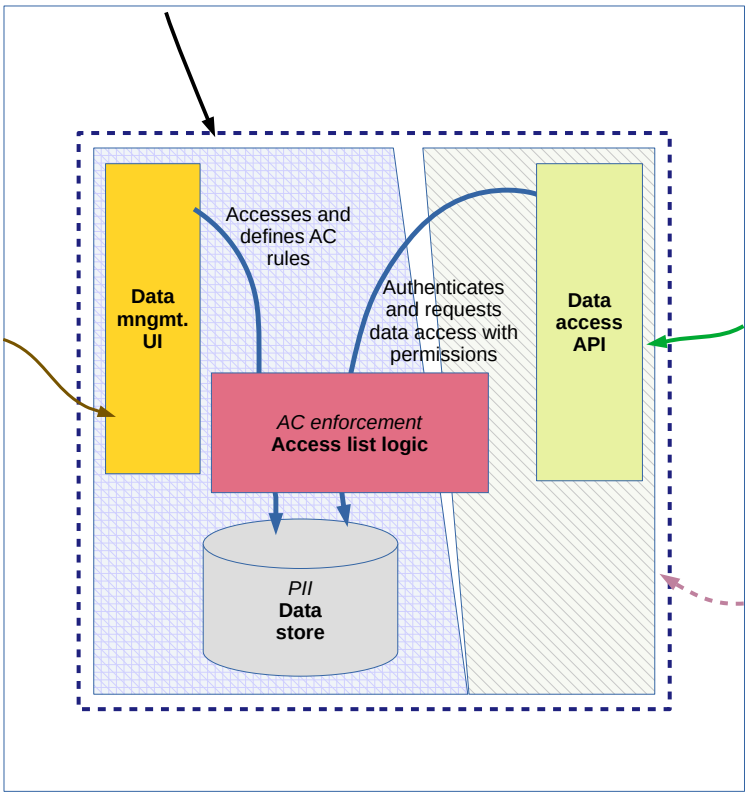


Figure 3.5: PDS architectural layout diagram

3.3.3 Anonymous certificate systems: *(U-Prove, Idemix)*

These solutions rely on a four-party architecture (see Figure 3.6) to support the certification of user attributes (as well as properties derived from these attributes). The cropped arrows and color code of Figure 3.6 refer to the convention and entities in the generic architecture depicted in Figure 3.1.

In these solutions, a Prover (P) owns some PII. (P) obtains anonymous certificates from an Issuer (I). (P) is able to prove the validity of some (properties over) PII to a Verifier (V) which is the PSP in our scenario. Depending on some contractual conditions, a Revocation Referee (RR) may cancel the anonymity and thus re-identify the certificates–*i.e.*, identify the owner of the certificates.

The anonymity property is ensured thanks to the two following principles: the communication unlinkability and the *zero-knowledge* proofs of knowledge [9, 18].

Communication unlinkability refers to the security protocol preventing any entity over the network from possibly determining whether two proofs have been issued for the same user–even the destination services, acting as verifiers. Moreover, the proofs, which can be established to validate some PII or some properties over this PII, are benefiting from the zero-knowledge property: no additional information–apart from the veracity of the proof–can be inferred from the proof verification process.

These solutions are relevant for self-consistent transactions, in which the SP does not need to be bound to a specific user or to any other transactions. For instance, in case of online games, the PSP has to check whether a user is an adult before granting their access, but it does not have to learn about their name, their date of birth, *etc.* As such, only the information strictly needed is collected by PSP. Additional properties can also be ensured, such as non-replay of proofs, which is of interest for specific scenarios (*e.g.*, electronic cash transactions).

3.3.3.1 Selected solutions

The two selected solutions for this category are: Idemix ("*identity mixer*") [12] and U-Prove [71].

They both propose similar features, *i.e.*, a set of anonymous certificates for users to freely use with SPs. These certificates enforce:

- service unlinkability, meaning that a service A is not able to determine that the user is also served by a service B.
- transaction unlinkability, meaning that a service A is not able to determine whether two of its transactions have been issued for the same actual user.

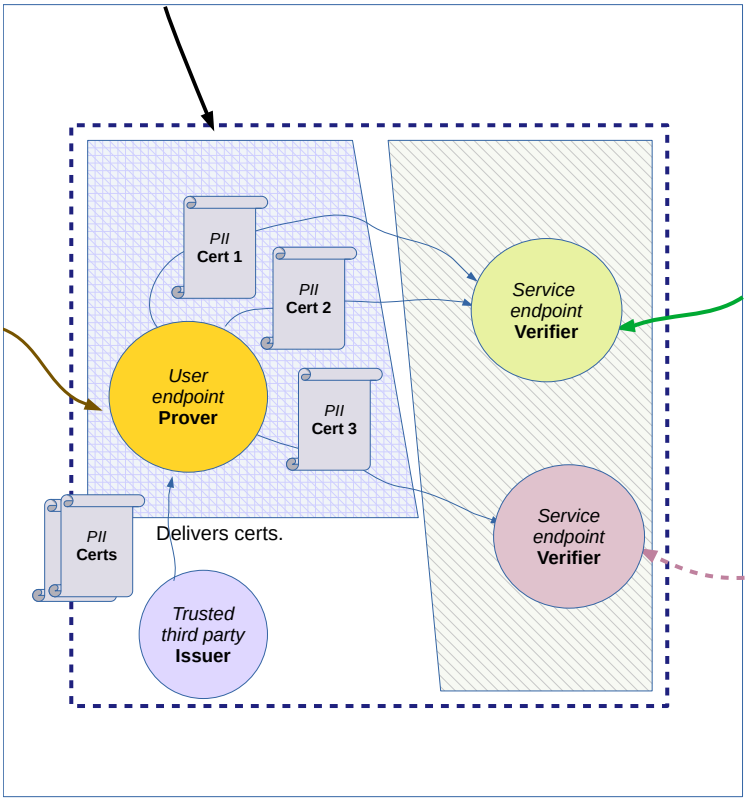


Figure 3.6: Anonymous certificate architectural layout diagram

3.3.4 Access-control delegation architectures: *(User-Managed Access, INDIGO)*

These architectural solutions enable users to delegate access control to a dedicated software agent, acting as an authorization server. This server deals with resource access on one or several resource server(s). The resulting category studied here stands at an upper, more abstract level than the three other categories of solutions: instead of proposing a single software tool, this category specifies the interactions happening between the different software entities ensuring PII management.

This type of solutions puts a strong emphasis over delegation, enabling the users to define how access control should be handled in an autonomous manner.

They enforce role de-correlation over the system by keeping authorization and access control, on the one hand, and data storage, on the other hand, split into separate logical entities. The SPs are then given data access through a standardized interface.

The goal of the de-correlation is to ensure that the various responsibilities are dispatched evenly over the entities. In case of one or more entities acting maliciously, this role de-correlation process also reduces the risk of a failure of the architecture.

Studying this category of solutions is assessing whether:

- a single software solution may not be enough in order to enforce our use case (Section 2.2),
- or an architectural pattern along with a communication protocol and specified interfaces should be chosen.

3.3.4.1 Selected solutions

The two selected solutions for this category are: User-Managed Access [56] and INDIGO [16].

User-Managed Access is an OAuth2 profile meant for access control delegation. It requires a five-party OAuth2 architecture, as depicted in Figure 3.7, in order for the user to enforce access control delegation, and for SPs to consistently access the user's PII. The cropped arrows and color code of Figure 3.7 refer to the convention and entities in the generic architecture depicted in Figure 3.1.

As mentioned above, five entities contribute to this architecture. The user appears as a resource owner (RO). The user is also responsible for their PII management on one (or several) resource servers (RS), and for managing consent to SPs on the authorization server (AS). The SPs appear as requesting parties (RP). The SPs interact with the authorization server, one or several (RS) and indirectly with the resource owner through a client (C).

The PII access happens in three steps:

- (RO) declares the protection upon a resource either before or while (C) attempts any access on it. This configuration uses UMA's *protection API*, and is performed at resource registration;
- (RP) obtains authorization on (AS) through (C). If the user consent has not been obtained before, it may obtained during that step (through the *interactive claims gathering*). If the authorization succeeds, an access token ("Requesting Party Token") is given;
- (RP) uses the access token on (RS), through (C), in order to obtain the resource.

User-Managed Access also specifies an offline decision algorithm based on the client's requested scopes in comparison with the client's previously granted scopes (*e.g.*, when the user was online). Currently being reviewed by the IETF for publication as a *Request for Comments* (RFC), User-Managed Access is meant to enhance the supported OAuth scenarios by bringing further delegation. Although OAuth itself leaves room for delegation, many of its actual delegation technical- and implementation details are *out of scope* of the OAuth 2.0 official specification.

INDIGO, on the other hand, is a cloud computing software suite for authorization and authentication support and targeted at scientific communities. It enables researchers to share documents and data using complex access-control enforcement scenarios. Although meant for collaborative research efforts, it supports numerous features relevant to personal data self-management involving interactions with ASPs and PSPs–mainly thanks to its emphasis on protocol interoperability, access control delegation, and users' official-identities management.

INDIGO's architecture brings solutions for a certain number of identity management problems. For instance, protocol interoperability is ensured thanks to a token translation system. Token translation means that the user can authenticate through different schemes (SAML, OIDC, or plain X.509 certificates). The SPs accessing the user PII then need to interface the solution as OIDC relying parties. The token translation service also enables the non OIDC-compliant SPs to interface with the solution, using other identification protocols.

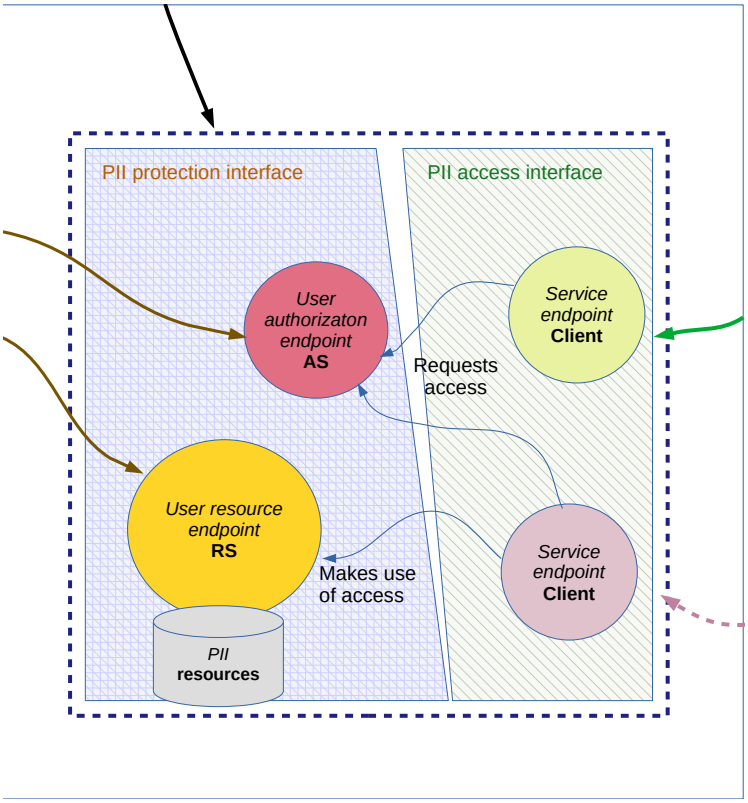


Figure 3.7: AC delegation architecture layout diagram

3.4 Evaluation of the Selected Solutions

This section provides an analysis of the categories of solutions down to individual solutions, with regard to the fourteen criteria identified in Section 3.2. A comparative study with a more significant list of criteria, including criteria that are not strictly related to our territorial use case of Section 2.2, is visible in [57].

A synthetic comparative evaluation is given in Tables 3.4 and 3.5. The five critical criteria identified in Section 3.2 appear in bold font in these two tables. For readability purpose and at no inaccuracy price, several solutions were gathered in the same column, as several approaches are close enough on a functional basis. Thus, one column "Anonymous Certificates" gathers the two anonymous certificate solutions Idemix [12] and U-Prove [71], and column "Federation IdMs" gathers OpenIDM, Keystone and Keycloak solutions.

3.4.1 Type of User Consent

When using IdMs, consent is given by the user through a successful authentication phase. For some identification protocols such as OIDC, the IdM acting as an identity provider shows the user a list of PII pending for their approval before collection by the service provider. This consent, which can be revoked at any time, results in ACL being defined on the IdM regarding the service provider.
According to the European regulation, the consent given by the user is valid provided that the user understands the purposes of the PII collection.
Alternatively, BlindIDM also enforces ACL diffusion. It uses signed cookies ("Macaroons" [7]),

Table 3.4: Comparative evaluation of PII self-management solutions for consent management criteria

Solution / Criterion	BlindIDM	Authentic	Federation IdMs	openPDS SA	Mydex	Databox	Fargo	Anon. Cert.	UMA	INDIGO
References	[67]	—	—	[61]	[70]	[35]	—	[12, 71]	[56]	[16]
Type of user consent	Through login	Through login + ACL on PII attrs by the admin	Through login + ACL on PII attrs by the admin	ACL on *Safe Answers* for SPs by the user	ACL on PII for SPs by the user	ACL, TTL, anon. (enforced by the *arbiter*)	Not supported	ZKPs generated by the user	Protection API consent flow	OAuth2 consent flow
Type(s) of supported access-control	No info. avail.	(RB)AC on the IdP, for each federation	Multiple AC models on the IdP, for each federation	Simple grant/revocation of local PII processing requests	Temporal ACLs on PII or services	Global *Manager* with local delegation to services	No AC implemented	Not applicable	Double interface (protection API and token endpoint)	OAuth2 authorization model
PII collection purpose definition	Not relevant for IdM	Not relevant for IdM	Not relevant for IdM	Not applicable (no PII collection)	No info. avail.	No info. avail.	Not supported	Yes, during contractual agreement	Possible (depends on the client implementation)	Possible (depends on the client implementation)

Table 3.5: Comparative evaluation of PII self-management solutions for data exchange flow criteria

Solution / Criterion	BlindIDM	Authentic	Federation IdMs	openPDS SA	Mydex	Databox	Fargo	Anon. Cert.	UMA	INDIGO
References	[67]	—	—	[61]	[70]	[35]	—	[12, 71]	[56]	[16]
Type(s) of supported PII	Data, attr. and value metadata	Data, attr. and value metadata	Data, attr. and value metadata	Metadata	Data	Data, doc.	Doc.	Data, doc., metadata	Data, doc., metadata (depends on the implementation)	Data, doc., metadata (depends on the implementation)
PII validation	No info. avail.	Yes	No info. avail.	Not supported	No info. avail.	Not supported	Not supported	Yes	No info. avail.	No info. avail.
Prov. & deprov. management	No info. avail.	By the user, or SPs through user management REST API	By the user, or SPs through user management REST API	Deprov. by the user or SPs	By the user or SPs	By the user or SPs	By the user or SPs	By the user (PII is obtained from the issuer). Deprov is less critical (pseudonymity or anonymity, zero-knowledge)	User through RS, or SP after authz obtained from AS	User, or SP after authz obtained from AS
Re-usability of previously uploaded PII	Yes, identity is federated	Yes, identity is federated	Yes, identity is federated	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Minim. management	Minim. towards the IdM itself only	No	No	Yes, no raw metadata transfer	No info. avail.	No info. avail.	Not supported	Yes	Possible, client implementation dependent	Possible, client implementation dependent
Support of remote PII sources	No info. avail.	Limited: support of user account backends	Limited: support of user account backends	Not supported	Not supported	Depends on driver implementation	Not supported	No info. avail., possible in future implementations	Possible, client impl. dependent + federated authz	Possible, client implementation dependent

Table 3.6: Comparative evaluation of PII self-management solutions for misc. user governance criteria

Solution / Criterion	BlindIDM	Authentic	Federation IdMs	openPDS SA	Mydex	Databox	Fargo	Anon. Cert.	UMA	INDIGO
References	[67]	—	—	[61]	[70]	[35]	—	[12, 71]	[56]	[16]
Privacy usability trade-off	IdM can't read PII	None (Web identity federation configuration)	None (Web identity federation configuration)	Computational trade-off (The PII processing has to be local)	User needs to set ACLs	Requires local drivers	Not supported	Potential anonymity revocation	User needs to define delegation	User needs to define delegation
User interface	No info. avail.	Web UI	Web UI mostly. If not, set of APIs for UI frontend.	Through mobile app.	Native client UI	No info. avail.	Web UI	Native cert. management. app.	No info. avail.	Web UI
Service provider revocation	Yes (SAML identity federation)	Yes (SAML or OIDC identity federation)	Yes (SAML or OIDC identity federation)	Yes (ACLs)	Yes (ACLs or TTL on authorization)	Yes (ACLs or TTL on PII)	Not supported	Not applicable (user-triggered atomic transactions)	Yes, by the user, enforced by the UMA AS	Yes, by the user, enforced by the OAuth2 AS
Extent of delegation	Not supported	Not supported	Limited delegation scenarios	Simple PII transfer delegation	Simple PII transfer delegation	Queries on agent acting on behalf of the user	Not supported	Full delegation possible, depends on implementation	Complex delegation possible	Simple PII transfer delegation
History / logging of transfers	No info. avail.	Yes, for admin & users	Yes, for admin at least	Audit log UI avail.	No info. avail.	1 data store dedicated to logging	For the platform admin only	No info. avail.	No info. avail.	No info. avail.

which are suitable for carrying user consent information, however, without information about the possibility for users to trigger the generation of such signed cookies carrying caveats of their choice.

With PDS solutions, the approach is different. As the SPs interacting with openPDS only get to receive the processed output for PII processing algorithms, the consent management for this solution consists in configuring how this local processing should happen. The user is able to define which SP is able to send requests, and this definition stands for user consent.
Alternatively, Mydex's consent management simply consists in maintaining a list of AC rules, defined by the user, for each SP. The user does not have any finer-grained consent mechanisms.
Databox lets the user consent to PII collection under some conditions. For instance, the user may define a TTL function for some given PII, or choose an anonymization algorithm to be ran against the PII before any collection happens. In spite of the rather decentralized architecture of Databox, these functionalities are all performed by the central *arbiter*.
Eventually, the rudimentary consent management capabilities of Fargo reflects its basic PII management features: the user, which manually manages the documents of the PDS through CRUD operations, also manually consents to any single reuse of these documents in an administrative online procedure.

With anonymous certificate systems, the user's consent is the decision to disclose the knowledge of some of their PII to SPs.

Access control delegation solutions are meant to tackle user consent issues through a consent management entity.

In UMA, this entity is the authorization server, whose interactions with the SPs are performed using its access token and authorization endpoints. The access policies are defined on the authorization server at resource registration, using its protection API.
INDIGO also implements an OAuth2 authorization server, but does not specify how the implementation should manage user consent.

3.4.2 Type(s) of Supported Access Control

The administrator of federated IdMs enforces the main configuration for data management, leaving the user little flexibility for adapting preferences to their own needs. On the contrary, on the platform administrator side, such federation IdMs offer rich access-control configuration. For instance, Keycloak combines different access control rules involving roles, user attributes and contextual authorization information. Federated IdMs also cascade access control to service providers as part of the identity federation process[8]. Authentic whose access control model is much simpler than the two other federation IdMs selected for this survey, widely relies on the role-based access control model (RBAC [69]) defined in the Django Web framework.
Each user is registered to a set of roles, and for each role, a set of permissions is granted. At a given time the user's authorizations are deduced from the permissions of whole set of roles. Some RBAC implementations also support role sessions, in which the activation/deactivation of a role for a given user may dynamically change over time. A pillar concept of RBAC is the *role hierarchy* [29], in which junior roles can be derived from senior roles according to the following transitivity rules:

- Permissions defined on junior roles transit up to senior roles.
- Any user who is assigned senior roles also obtain their junior roles.

Each PDS solution supports one or several specific access control mechanisms.
Mydex [70] displays an ACL configuration interface to the user.
Each rule in the list can be limited in time, assigned to a restricted subset of services, and revoked at any time.
openPDS [61] proposes a simple grant and revocation mechanism, ensuring the legitimacy of SPs to send requests.
The access control in Databox [62] happens at two different stages:
• the local drivers offer an interface between local data stores and SPs. Databox requires the deployment of one driver for each connected SP. The access control performed at driver-level is made of lists (ACLs).
• the central *Manager* is the main management entity for the functional ecosystem. It coordinates the access control on a more global scale. This access control being performed by a unique and central authority is a mandatory access control system, as presented in [93]. Access control is enforced at the Manager level through some access tokens. Fargo does not implement any thorough access control.

Access control is out of the scope of the anonymous certificate use cases presented in [71] and [12].
More precisely, the user decides which SP the transfer shall happen with. The user is also able to

[8]OpenIDM delegates the federation logic to another module, OpenAM. For readability purposes, and as the two modules are complementary, only OpenIDM will be mentioned in this survey. For more information regarding, OpenAM see `https://backstage.forgerock.com/docs/am`.

manage their different anonymous certificates, possibly switching between multiple digital identities while interfering with different SPs.

UMA offers a twofold access control through its protection API and its access token endpoint. INDIGO [16] complies with the OAuth2 authorization model. The access control is performed by an authorization server (AS), issuing access tokens and refresh tokens to SPs through the use of dedicated clients. Theses SPs must have previously obtained the user's authorization. The implementation of the dedicated clients should be OAuth2-compliant, and are not covered in the INDIGO presentation article.

3.4.3 PII Collection Purpose Definition

IdMs do not handle the concept of PII collection purpose definition. Instead, it is the role of the SPs to declare the reason for the PII collection.

Thus the user is asked for their consent for a particular purpose. However, for this category, no mechanism enforces that SPs use the collected PII according to the declared purpose(s). As a result, this criterion is not relevant for this category of solutions.

Among the PDS solutions, only openPDS addresses this concern, however indirectly: openPDS makes it possible to avoid any raw PII collection. *SafeAnswers* ensures that the PII processing happens locally. Therefore, rather than enforcing collection purposes definition for SPs, openPDS makes it possible to view the underlying PII processing algorithm. This result information, sent to the SP, is of a lower dimensionality than the original input PII, making it more difficult for an intruder to infer identifiable information from it. For instance, it prevents re-identifying partially-anonymized data through usual approaches of data-linkage from multiple sources [37].

The purpose of the PII certificate issuance is defined when reaching the contractual agreement between the user and the certificate issuer. Anonymous certificate systems require the user to establish a contract with the SPs, so as to decide which proofs of known PII have to be delivered to them. The negotiation is not covered in the solution, instead it is part of the implementation-specific contractual agreement.

Eventually, UMA and INDIGO do not directly consider this criterion, which is left to the client implementation. For legal reasons, the client implementation has to deal with PII collection purpose definition concerns. Negotiation of collection parameters in delegation architectures needs to happen before any SP accesses the user PII. The OAuth2 protocol–used in both the UMA and INDIGO solutions–leaves negotiation concerns to the implementation of the client.
UMA suggests the implementation of an interactive claims gathering process, potentially prone to supporting a user-driven negotiation, however without any further implementation details. Additionally, the UMA data-usage specification document [54] declares that the entities from UMA-compliant solutions are able to define a legal or contractual agreement defining the rights and responsibilities of each party.

3.4.4 Privacy Usability Trade-off

There is no visible trade-off for the user of federation IdMs and Authentic, as both the privacy and usability are enhanced by the federation process. This absence of visible trade-off is made possible by identity federation protocols, responsible for PII transfer agreement directly between SPs and federation IdMs.
However, after the PII collection has happened, no FIM mechanism enforces that the PII is used according to the claimed purpose(s), as declared by the SPs.
Alternatively, BlindIDM [67] supports more restrictive security hypotheses: it enforces PII privacy of the user from the IdM at no usability cost. The re-encryption proxy that BlindIDM manages prevents the IdM from reading the exchanged PII, as any other IdM could do, in a *semi-honest* manner. Once the proxy has been established (more particularly, after the re-encryption keys have been generated), this setting is transparent to the user. However the number of keys managed by an IdM can increases rapidly: n users connecting to m SPs require the IdM to manage $n \times m$ re-encryption keys.

openPDS [61], with its local computation process requires that the PII processing happen locally. The solution therefore has a computational trade-off at ensuring user privacy: the PDS is responsible for the data processing, on behalf of the SPs. There is also an SP-side of the usability trade-off: the SPs must send requests to the PDS using the specific *SafeAnswers* interface[9]. Nonetheless, nothing in the related article asserts any trade-off from the point of view of the user.
Adopting a more conventional approach, Mydex [70] requires that the user defines some access control lists for their PII. It is considered as a usability trade-off from the point of view of the user, as some users may not want to define such ACLs themselves.
Databox [62] relies on one local driver for each of its data store. As far as the article goes, there is however no visible trade-off perceived by the user, and there is no trade-off for the SP either.
Privacy enforcement mechanisms for Fargo are rudimentary, as a result there is no usability trade-off.

With anonymous certificate systems, the minimal disclosure of knowledge is under full control of the user, as explained in Section 3.3.3. There is no user trade-off, as only the user is responsible for the way they manage their different anonymous certificates. Nonetheless, the user should know that anonymity might be revoked.
The SP trade-off is the necessity to implement a zero-knowledge proof verification procedure.

Access control delegation architectures require that users explicitly grant authorizations at first. The authorization grant must happen either before the PII transfer, of during it.

3.4.5 User Interface

For federation IdMs, the FIM authentication process is transparent to the user. The authentication scenario from the user point of view, whose simplified representation is in Figure 3.3, does not change in spite of the separation between SPs and identity providers. The authentication procedure itself is simple, usually relying on a simple login prompt on a Web HTML page. The

[9]This second type of usability trade-off could not be assessed in this survey, as no further information could be found about the *SafeAnswers* module.

resulting SSO-authenticated session is also transparent to the user.
The identity federation process therefore does not impact the user interface, which stays as simple as it would be for a non-federated SP. This is the case for federation IdMs as well as for Authentic. As a result, and as a downside of this seamless interface, experience proved that the user of federated SPs does not always understand the federation process [1]. Additionally, when SPs require the collection of the user's PII, either the IdM acting as an identity provider or the SP must offer a consent collection page to the user.
BlindIDM [67] gives no information about its user interface.
Eventually, Keystone, on the contrary, works as a set of identity management services whose only interfaces are APIs.

PDS solutions provide a limited informational user interface, whose main purpose is for the user to manage their PII, with optional configuration capabilities.
openPDS, through its mobile app, proposes a UI for the configuration of the *SafeAnswers* module. This interface outputs (i) the questions "asked" by the SPs to the PDS, (ii) the answer provided by the PDS, as well as (iii) the PII used for the computation of the answer. The UI also makes it possible for the user to view the number of PII processing requests sent by an SP over a given period of time.
The UI proposed by Mydex allows the users to configure a set of ACLs ensuring that data collection is adequately configured, as shown by the visual elements in [70]. As illustrated by these visual elements, the user is displayed the purpose of the PII collection, for one particular PII type. They can define which of the four CRUD actions the SPs can take for this type of PII. They can define a time-to-live value (TTL) for the PII retention. Eventually, they can also decide whether these SPs are allowed to share this data with third-parties SPs. However, the TTL defined by the user is enforced on the PDS only, and no mechanism is proposed for the enforcement of PII TTL when it has already been collected by SPs.
Fargo proposes a simple list of uploaded documents through its Web UI. The user is able to perform simple operations such as uploading, downloading or deleting document.

The specifications provided for anonymous certificate systems do not cover the user interface. U-Prove [71] provides a native desktop client application. The client providing the UI helps the user communicate with the certificate issuer. It enables the user to understand the contractual agreements met with the issuer, including the anonymity revocation policies. The interface enables the user to derive proofs of knowledge based on these certificates. Eventually, it also helps the user communicate with SPs acting as verifiers.

Neither UMA nor INDIGO specifies any UI. The software implementation has to design and provide a UI regardless of the specification documents for these two solutions.

3.4.6 Service Provider Revocation

Authenticated session management by IdMs is the closest such IdMS are to supporting SP revocation : the user can explicitly log out of their FIM session through Single Log-Out (SLO) services. Additonally, the IdM acts as a SAML identity provider, SPs expose SLO SOAP services for direct communication–*i.e.*, not involving the User Agent. The OIDC identification protocol does not support any such revocation capabilities not requiring any interaction with the user agent. Additional specifications [28] have been designed by the OpenId Foundation in order to fill

that gap.

With PDS solutions, the enforcement of simple ACLs for each SP allows for a revocation process compatible with our use case. Such a revocation process is supported by openPDS.
Mydex also allows the user to define implicit revocations triggered after a given timestamp: the user's authorizations for SPs to access PII can be held valid for a specific time-window only. When that time window is over, the resulting revocation happens automatically.
Databox also ensures implicit revocation, as users can define a TTL value on their PII. Thus, SPs having previously collected the users' PII are supposed to discard them when the TTL reaches zero. However, no technical enforcement of this TTL mechanism is presented in the related article [62].

With anonymous certificate systems, certificate information is self-contained, and potentially meant for single-use only. Hence the transaction involving the proof of knowledge is atomic and user-triggered. As a result there is no SP revocation per se.

Finally, in delegation architectures, the central authorization server manages, on behalf of the user, the authorization grants and decides whether to renew access authorizations to SPs. Optionally, and depending on the implementation, the authorization server may expose an access policy definition UI for the resource owner.

3.4.7 Extent of Delegation

With IdM solutions, the user does not delegate any PII management task. Some secondary modes such as offline token issuance imply, to some extent, user delegation–however these secondary modes do not handle delegation as defined in the selected use case in Section 2.2. Keycloak and OpenIDM also serve as Authorization Servers in the User-Managed Access profile for OAuth2, which is the one more step to enforcing the required level of delegation.
Additionally, federation IdMs as well as Authentic comply with the OpenID Connect specifications when acting as identity providers, and are able to deliver offline tokens to the OIDC relying parties. These tokens may be, under certain conditions (especially their expiration timestamp), stored for later use by the relying parties. Still according to the OIDC specifications, the federations IdMs can deliver refresh tokens to their relying parties, enabling later offline access rights renewal.

The delegation capabilities of openPDS and Mydex are simple data access grants and revocations, for each SP.
Databox implements an agent acting on behalf of the user, after obtaining their access-control preferences regarding each SP.

Neither Idemix or U-Prove mention the extent of consent for delegation. However, the certificate disclosure algorithm presented in [12, Section 15.6.4] (see the `ShowCert` primitive), revealing an anonymous certificate to an SP acting as a verifier, can be handled, depending on the implementation, by a software agent acting on behalf of the user.
Additionally, apart from their ability to present standard (interactive) proofs of knowledge, an anonymous certificate system implementation can support non-interactive proofs, as originally suggested by [73], which are perfectly suitable for handling the offline mode: the non-interactive

proofs can be computed first by the prover, and only then showed to a verifier, in an offline manner.

Delegation architectures are by nature designed to implement authorization delegation. The authorization server is the central delegation entity of the user, it enables the fine-grained consent scenarios that are necessary for our use case to be enforced. It acts on behalf of the user when deciding whether to grant, to deny or to renew access to PII for an SP. Usually, the user can define the access policies that apply at resource registration, on the authorization server. Thus the authorization granted to the parties requesting access to PII on the resource server is not necessarily synchronous–*i.e.* it can be performed while the user is offline. Depending on the authorization server configuration, the client, acting on behalf of an SP, can obtain an access token, usually with a limited time-to-live. A pseudo-algorithm, meant as an implementation guideline and visible in [56, chap 3.3.4], defines the conditions under which the authorization server delivers an access token, and which scopes are granted with it.

3.4.8 History/Logging of Transfers

According to its configuration, the Web server bound to any Web IdM solution can log the transfer metadata from HTTP requests and responses (see the SAML2 [13] and the OpenID Connect [83] protocol specifications for details about metadata format for Web FIM). Additionally, the IdM may log FIM authentication sessions across service providers, for traceability or debugging purposes.

The ability to address this criterion differs from one PDS solution to another.
openPDS offers a log page to the user, giving their global information about requests sent by the SPs. Fine-grained PII logging capabilities as required by our use case have not been presented in the related article [61].
Databox, on the contrary, deploys a specific *data store* whose role is to log any PII transfer. However, the high-level presentation of the solution in the related article does not mention the exact content of the logs.
Fargo, as a Web application, stores HTTP metadata through its underlying Web server (either emitted or received). Its rather basic supported use cases do not require fine-grained logging of PII transfers. Logging information in Fargo is not made available to the user.

A logging facility is necessary to anonymous certificate systems if willing to support a revocation procedure. The anonymous transactions may be logged so as to be retrieved in case of a contractual conflict between the prover and the verifier.

Eventually, neither UMA or INDIGO mention how the logging of transactions should happen. However, the Kantara Initiative, editor of the UMA specifications, has also proposed a consent receipt model [47, 38], which maintains an history of the consents granted by the users.

3.4.9 Type(s) of Supported PII

IdMs handle raw data and metadata linked to a user account. This PII is transferred to SPs during user authentication phases.

Once again, each of the selected PDS solutions adopts a different approach. openPDS is designed to perform PII metadata storage. The use cases given as examples in [61] target biometrics PII metadata. Mydex also stores PII data, while Databox stores both PII data and user documents (see examples given in [35]). Eventually, Fargo stores user documents.

Proofs of knowledge are inherent to anonymous certificate systems, and can support any of the three types of PII discussed in Section 3.2.2.1. These solutions allow the user to prove properties such as documents ownership, metadata validity and PII data authenticity.

At last, in access-control delegation architectures, the set of supported types are implementation-dependent. As a result, neither UMA or INDIGO explicitly mention the list of supported types of PII. Depending on the implementation, all three types can be supported.

3.4.10 PII Validation

The IdM, when acting as an IdP for an SP, is considered to be trusted by the SP. As a result, the SP can decide whether to validate or not raw PII and metadata provided by the IdP during the authentication phase.

The PDS solutions do not address the concern of data validation.
Mydex raises the concern of PII validation (see [70, Chapter 4]), however with no technical mechanism proposed to handle this process.

Anonymous certificate systems are designed for PII validation by a trusted authority (the certificate issuer).

Finally, PII validation concerns do not appear in the delegation architectures' respective presentation articles [56, 16].

3.4.11 Provisioning and Deprovisioning Management

Depending on the implementation, IdM solutions can offer an interface for the SPs to perform CRUD operations on user accounts.
Authentic and the three federation IdMs provide a user management REST [30] API, suitable for performing both provisioning and deprovisioning operations.
They also support backends in order to retrieve PII from remote sources, *e.g.*, user accounts from a remote directory server.

openPDS specifies a service protocol designed for read-only operations. This solution therefore focuses on PII *collection* by services, not on its *provisioning*. The PII is provisioned on the solution beforehand, through APIs. No further technical details about Databox's provisioning capabilities are given in the presentation article.
Mydex supports data provisioning by a set of registered services. The data flows happen bidirectionally, and services are able to collect or to provision data on the PDS. The user's consent is defined using access control lists, regulating access to data for each registered service, for a given time window.
Databox also provides an interface for SPs to provision PII. The access modalities of the interface

is implementation-specific, and is not covered in the Databox presentation article.
Fargo offers provisioning and deprovisioning by ASPs and PSPs. openPDS and Mydex also support PII deprovisioning by SPs.

For anonymous certificate solutions, the user is provisioned with PII certificates, at their own will, after contacting the issuer.
The anonymous certificate category does not support "provisioning", as defined in 3.2.2.3, as it does not enable either the ASPs or the PSPs to write or modify PII on the user-centric solution. The only information revealed is the proof of knowledge of the user's PII.
Additionally, when [a] the certificate issuer can be trusted (which is the normal use case of the solution) and when [b] no anonymity revocation is happening (meaning that there is not any contractual conflict between the different actors), no unnecessary PII may be revealed to the SPs. As a consequence, in case of a normal use of the system by all its actors, this category of solutions enforces the minimal disclosure of PII. Thus it reduces the need for PII deprovisioning.

UMA and INDIGO both rely on the OAuth2 protocol [36]. They are able to support PII provisioning or deprovisioning, as this protocol allows for PII management operations on the resource server.

3.4.12 Re-usability of previously uploaded PII

For IdMs solutions, the user PII stored as a profile account is by nature meant to be reused for later transfers.

The PII data collected by PDS is also meant to be reused. Re-usability is a core feature of this category of solutions.

The re-usability of anonymous certificates is supported by this category of solutions. This category of solutions also supports *disposable* certificates, although not relevant for our use case (see Section 2.2).

Eventually, access control and authorization delegation also ensures PII re-usability in a privacy-compliant manner.

3.4.13 Minimization Management

The IdMs are not able to check whether the PII requested by SPs is limited to what is strictly needed for providing services to user. FIM protocols such as OIDC *claims*, enabling SPs to request particular pieces of PII according to a given *profile*. However, no mechanism enables the IdM to verify that the claimed PII is actually needed for the actual services provided to users.

The PDS solutions implementing authorization protocols such as OAuth may propose minimization features. Depending on the client implementation, the user may be able to view the list of data required by SPs and to deny the authorization grant if the list is not minimized enough.
openPDS performs PII minimization by providing only some *safe answers* to the SPs, as explained

in Section 3.4.3.

Anonymous credentials systems are designed for data minimization support. ASPs and PSPs only know the minimum proofs of knowledge, with the optional ability to re-identify users in case of conflicts.

Eventually, implementation-specific data minimization might be enforced with access-control delegation architecture solutions, at the client-side of the architecture.

3.4.14 Support of Remote PII Sources

Federation IdMs and Authentic support authentication backends: they may be connected to remote sources such as directory servers which as a result can synchronize or duplicate partial or complete user account information.
IdMs can also perform *IdP-proxying*, thus becoming SP regarding a third-party IdP.
These features are however a strict subset of the functional expectations when applying this criterion to our use case.

For PDS solutions, only Databox provides a support of PII sources thanks to its extensible data-flow model. However, this support depends entirely on the implementation on the driver for each remote source.

The support of remote PII sources is not covered in the articles presenting this category of solutions.

UMA supports federated authorization [53], enabling remote OAuth2 resource servers (RS) to interact with a single authorization server. Alternatively, non-RS remote sources can implement a client-side party, obeying to the supported client-authorization protocols of these solutions.

3.5 Synthesis of the Functional Evaluation

This section provides a synthesis of the per-category evaluation presented in Section 3.4.

An analysis of how each category may be suitable for supporting administrative services–determining whether these categories address the five critical criteria for our use case–is also conducted.

Alternatively, a concise interpretation of the synthesis is provided by Table 3.7, directly linking the four categories and their inherent ability to address the critical criteria.

3.5.1 Identity Managers

Identity managers provide PII exchange capabilities for their users. The exchanges happen as part of identity management and identification protocols. The Web IdMs require no additional component installation on the user's system–apart from a standard Web browser.

Table 3.7: Summary of the capabilities of categories of solutions to address the critical criteria

Criterion \ Solution	Identity Managers	Personal Data Stores	Anonymous certificate systems	Delegation architectures
Consent management (category)	Supportable	Supportable	No need for it, minimization of collection	Supported, by nature
Extent of delegation	Significant	Implementation-dependent	Not relevant	Significant, by nature
PII validation	Supportable, implementation-dependent	Supportable, implementation-dependent	Inherently supportable	Supportable, implementation-dependent
Re-usability of previously uploaded PII	Limited, by nature	Inherently supported	Inherently supported	Inherently supported
Support of remote PII sources	Implementation-dependent	Implementation-dependent	Not relevant	Implementation-dependent

However, most of the user-driven PII management features offered are not standardized, and they differ from an IdM solution to another.

Also, authorization protocols support several modes, profiles, grant types and authorization flows, thus leading to many varying implementations. Even implementations of the same protocol can decide to adopt mutually exclusive subsets of these variations. For instance, OpenID Connect specifies various profiles and multiple authorization schemes.

Moreover, these IdMs solutions also rely on their sets of specific security hypotheses. For instance, Authentic is at the heart of the chain of trust. By design, it manages the user's PII and has direct access to it. Many user attributes and metadata, obeying to an extensible user attribute model, are stored in the IdM's database. Although the FIM protocols it uses provide security and privacy properties even in case of untrusted service providers, a security failure within the IdM would result in user privacy threats. On the contrary, BlindIDM supports much more restrictive security hypotheses, preventing the IdM to read the users' cleartext PII. BlindIDM could therefore be deployed as an IdP proxy, and is attractive in the case of an untrusted IdM, which is not part of our assumption of our case.

The evaluation of IdMs according to the five critical criteria reveals that:

- The support of remote PII sources is limited for some solutions of this category.
- The consent management model remains implementation-dependent.
- Simple delegation scenarios are possible only for a subset of solutions from this category. No further delegation model is proposed.
- The support of validated PII is also strictly implementation-dependent.

More generally, IdMs do not provide all the necessary features required for the enforcement of our territorial use case. From the user's point of view, IdMs enable identification and account self-management. The other PII management features expected in this survey are by nature not applicable to IdMs. For instance, interfacing with PII sources would require that these sources always support FIM protocols, which is not the case.

Finally, targeted administrative services do not necessarily imply authentication nor they need to manipulate user accounts. For instance, they may require only atomic PII transfers in order to fulfill a user request–or the service can be used in an anonymous manner. Eventually, mechanisms

such as *tracking codes* enable users to view the processing of their requests, without imposing any account nor explicit identification process–hence without the need for identity management.

3.5.2 Personal Data Stores

Two different approaches in terms of user governance are retained by PDSs.

The first approach is the deployment of an online PII-storage instance common to several users. As a result the storage entity is not a personal instance and is not entirely user-driven. It may be deployed in cloud architectures, as does Mydex, designed specifically for this approach. Such solutions into production usually refer to a business model such as *Software as a Service* (SaaS) [92].

The second approach is a fully user-driven storage instance, as supported in Databox for instance. It enforces user governance thanks to the physical ownership of the user's data. This approach is particularly suitable for deployment on a user device (*e.g.*, a smartphone or even specific SoC hardware). In this scenario, the user has the complete responsibilities regarding the management and the transfer of the PII stored on the PDS. This scenario is relevant for the respect of the user's privacy, provided that the user understands the PII management features offered by the solution.

As a consequence, PDSs offer a PII dashboard for users to manage their PII and optionally their previously given consent to PII collection. The user's consent for data collection by third parties is made on an "all-or-nothing" basis: either users keep their PII private, or they make this PII completely available to a third party. Additionally, these dashboard tools may enable users to visualize which services are registered. They also offer data consumption parameters or metrics. Optionally, they may even help users manage storage capabilities on the PDS. The full PII lifecycle is therefore handled by these solutions.

For a given SP and a given PII processing purpose, the PDS must also obtain the user's consent before transferring any PII.

Eventually, evaluating PDSs according to our critical criteria reveals that:

- Databox indirectly addresses the support of remote PII sources.
- Managing consent and delegation remains simple ; advanced scenario such as negotiation or partially-autonomous decision making are not supported.
- However applicable by this category, no selected solution offers PII validation features.

Hence, for simple PII management scenarios, PDS are suitable for handling services offered by the administration and collectivities–especially when these services require a recurrent collection of reusable PII.

3.5.3 Anonymous Certificates

Both anonymous certificate solutions Idemix and U-Prove implement the same set of PII management features. They provide an elegant way to apply the principle of data minimization, as required by the current legislation in the European Union. This principle is enforced by the minimal disclosure of PII.

These solutions are user-centric by design, but not entirely user-driven: actions on the system such as certificate issuance, anonymity revocation and certificate verification are not led by the user. Indeed, their trust model requires that a certificate issuer and a revocation referee be managed by trusted collectivity entities. This model is suitable to the administration context.

Eventually, the revocation referee may act at the expense of the user. The anonymity-revocation policies, as mentioned in the Idemix presentation article [12], may happen without consent from the user. This can happen in case of certificate misuse by the latter, *e.g.*, when the user does not comply with the contractual clauses defined with the SP.

Anonymous certificate systems provide an answer to the critical criteria identified in Section 3.2, as detailed below:

- Supporting remote PII sources is implementation-dependent, and is not addressed directly in the two articles presenting the selected solutions.
- Delegation is possible, but also entirely depends on the implementation. It is not covered in the two articles presenting this category of solutions.
- PII validation is a core concept of this category of solutions.

As a result, anonymous certificate systems are suitable for some needs of the administrations and territorial collectivities. They would prove to be useful when, instead of a significant PII collection, the services only need the assurance that the user is legitimate, or that they detain the adequate information to use the services. Of course, that information can be considered to be PII altogether, but it drastically complies with the principle of minimization of collected information. For instance, an ASP willing to compute some anonymous statistics regarding its users, without any involved PII but the assurance that the user is legitimate, could deploy an anonymous certificate system to obtain this legitimacy information from its users.

3.5.4 Access Control Delegation Architectures

UMA stands out as it asynchronously delegates PII access control to a dedicated entity (an authorization server) It provides two standardized interfaces on user side and SP side. It decorrelates the access policy definition, performed by the user, and the enforcement of these policies, performed by the authorization server.

INDIGO adopts a different approach. Its token translation system makes it possible to support various authorization protocols. It also enables the user to define a set of authorization rules, meant for a software agent to act on their behalf. Although designed as a collaborative academic research tool, it satisfies several needs of our selected use case.

The critical criteria are addressed by this category of solutions as below:

- The support of remote PII sources is implementation-dependent.
- Delegation and consent management are core features.
- PII validation is not covered in the related literature.

More generally, this category of solutions is suitable for services offered by administrations and collectivities in which repeated authorization decisions on behalf of an offline user have to be made. Eventually, the user may appreciate being able to define a set of authorization policies, which are then enforced by the solution.

3.6 Conclusion: Identifying an Optimal Solution

This chapter surveys the technologies addressing personal data self-management in the context of administrative and territorial public service providers. The resulting comprehensive and comparative study identifies the current limits of these technologies, specifically with regard to our five identified critical criteria. We observe generally that there is no definitive position of the approaches against these five criteria, and that some of them are designed to address at least one of the criteria, but not all five at once. No identified solution can address the use case in its totality, only subsets of it.

In order to meet our use case, we can provide layout for a solution matching as many criteria as possible, thus making this solution *optimal*. In particular, this optimal solution has to enforce the five critical criteria identified in Section 3.2. As per our use case in Section 2.2, deploying an access management ecosystem–*i.e.*, the second family of solutions according to the taxonomy performed in Figure 3.2–is not necessary and can't cover our five critical criteria adequately. In particular, the properties enforced by this category of solutions (i) are at too high an abstraction level and (ii) leave too much to potential implementations of subcomponents, for them to be considered optimal. Moreover, as explained in 3.5.3, anonymous certificate systems are relevant in only a subpart of our use case.

Identity providing may be necessary, but it is not central enough for the use case for an IdM solution to be selected as the optimal solution. Additionally, the needs for the optimal solution to support remote sources and extended delegation schemes disqualify IdMs.
Instead, we identify the need for an *augmented* PDS tool, which would provide the following features:

- An extensible remote-source support model. Indeed, the administration and collectivities already offer a wide variety of PII sources, made available to any of their authenticated users. These sources answer to the public services digitization initiatives among the administrations and collectivities of European countries. For instance, the French collectivities maintain their official APIs[10].

  This feature directly addresses criterion 3.2.2.6. Among PDS solutions, Databox proposes a modular and partially decentralized approach, which can be extended in order to fully support remote PII sources (1). Enhancing this model would make it possible for a PDS to abstract the PII location, *i.e.*, to support it whether it is locally hosted or remotely available.

- A clear interface of PII consumption directly mapped to the user's previously given consents. This feature addresses criterion 3.2.1.1. The support of consent metadata within PDSs has already been widely discussed, which eases the research work in order to implement proper consent metadata within a PDS. The optimal solution could for instance implement the management of consent receipt as defined by the Kantara initiative [47]. An optimal solution supporting such consent receipts would benefit the users as well as service providers, for both functional and legal purposes. Using this consent model, PDS as a monolithic logical entity could bear the roles of resource- and authorization-management. Conformance to authorization management protocols such as OAuth2 is a first step towards this objective.

- The ability to validate PII for simpler user-relationship management processes. This addresses criterion 3.2.2.2. Such PII validation, implemented for instance by adding a simple

[10] See `https://api.gouv.fr/`.

boolean flag value to each element of the PII data model in the solution, would lead to simplifications in several PII management processes.

- The ability to act on behalf of the users for the multifold steps of PII management–from the creation or collection of this PII on the platform, to its use with ASPs and PSPs and eventually its deletion. This addresses criterion 3.2.3.4.
- The ability to ensure PII management when the user is not connected to the platform, which is a direct consequence of the conformance to critical criterion 3.2.3.4. This requirement is associated with the ability to reuse previously uploaded PII–addressed by critical criterion 3.2.2.4.

The support of remote PII sources also brings some more specific problems such as the automated matching of identity attributes retrieved across several sources.

As a result, a *PII manager* acting as a source hub should be chosen, minimizing the cumulated efforts necessary for a research contribution in order to handle all the aforementioned features. Such a PII manager is the main contribution of Chapter 4.

Taking a new perspective, the scientific literature such as [70] and [4] proves that many territorial collectivities are willing to participate in pilot projects regarding new PII management solutions. The administration and collectivities wish to enforce their status of authoritative entities providing digital services, and understand the need to be flawless regarding adequate PII protection for the users of SPs they offer, including by testing innovative solutions. These pilot projects are also a way to raise users' awareness, who in most cases, are not aware enough about their rights to privacy.

Chapter 4

User-Centric Consent Management and PII Retrieval From Third-Party Sources

This chapter presents a PII manager meeting the functional requirements for our use case. It enables the user to manage their PII when interfacing with their TCPA, while supporting PII sources.

4.1 Introduction

This chapter pertains to the field of access control of TCPA online services to their citizens. Users of TCPA are requested to submit some regulated administrative requests, *e.g.*, official document renewal, various allowance requests and registrations to local services. Generating and managing consent receipts enables the management of authorizations of the TCPA URM platforms to access the users' PII. The consent receipt model is the formalization of authorization information within the PII manager.

To benefit from these services, the user must provide Personally Identifiable Information (PII). Following our use case from Section 2.2, users issuing a school catering registration request are asked to fill in an online form including their PII fields, uploading scanned documents and retrieving PII from third-party sources, as explained in Subsection 2.2.3.

However, many shortcomings of user-centric PII management within TCPA have been identified over the previous years [49, Section 1.2.2]. The problem of the re-unification of personal data has been well identified in the literature [6, Chapter 4] and remains relevant. In fact, academic and industrial solutions, either they be (i) personal data stores [61, 70, 35, 24]; (ii) identity managers [67, 25, 31, 90, 74]; (iii) anonymous certificate systems [12, 71, 46]; or (iv) delegation architecture [56, 16], do not address the specific needs of PII management within TCPA, *i.e.*, local and national official entities providing online services to citizens.

In particular, none of these solutions addresses all the critical functional requirements for a standard TCPA's use case, namely: [a] the need to manage the various consent information given

by the user over time, [b] a wide extent of delegation on behalf of the user, [c] the possibility to validate PII and [d] the support of remote PII sources.

Moreover, existing solutions did not discuss three other concerns that arise in the specific context of the TCPA.
First, user consent must be enforced consistently regardless of the PII's actual location on any remote source. This chapter aims at providing a way for the user to define consent to offline PII processing, wherever the PII. The fact that the PII is provided by third-party sources does not hinder the consent management capabilities of the contribution.
Second, the interoperability concerns, that arise when dealing with such an heterogeneous system involving different sources, must be addressed.
Finally, the level of trust granted to remote sources for their role in providing user's PII must be formalized, with the possibility for a PII provided by an untrusted source to be relevant for our TCPA use case, but less relevant than a PII originating from a fully-trusted source. This chapter aims at specifying the levels of trust granted to sources, and their impact on the TCPA use case.

This chapter describes a case-study architecture for TCPA services with the primary concern of enforcing users' *informational governance* – see [58]. The main proposition of this chapter is a *PII manager* which supports the TCPA requirements with regard to PII management.

The PII manager meets the following requirements, identified as part of the previous chapter's synthesis:

1. The definition of a consent model that would enable the support of the relevant critical criteria, *i.e.*, the support of PII sources, the extent of delegation & consent management, and the support of online & offline modes. Indeed, solutions managing a thorough consent model relevant to the territorial use case, such as [47] have a tremendous advantage for enforcing subparts of the use case. They provide better traceability and enable delegation capabilities.

2. The interoperability concerns that arise when dealing with PII sources of different types. For instance, some sources may support plain HTTP Basic authentication [75] while others may implement the OAuth 2.0 authorization framework [36] instead. The implementers of such a *PII manager* may want to support the OAuth 2.0 **assertion** framework as specified in RFC 7521 [14] in order to provide further interoperability, for instance by being able to interface with SAML-based sources [13].

The remainder of this chapter is structured as follows:
Section 4.2 defines the system model, *i.e.*, the actors, the functional requirements, as well as the environment and technical hypotheses of our contribution. Section 4.3 describes the related works, *i.e.*, the academic or industrial solutions that are closely related to the use-case. Section 4.4 introduces the *PII manager* within an existing architecture. Section 4.5 deals with the dynamic discovery of the PII manager by the TCPA URM platforms, and the registration of the platforms on the PII manager. Sections 4.6, 4.7, 4.8 and 4.9 respectively present the PII Query Interface, the Core Consent Management module, the Source Backend and the PII Management User Interface of the PII manager. Eventually, Section 4.10 provides a functional analysis of the *PII manager*, proving its adequacy to the initial use case, before concluding in Section 4.11.

4.2 System Model of PII Sources Management

4.2.1 Environment Hypotheses in the Context of the PII Manager

This subsection extends the hypotheses detailed in Sections 2.4 and 2.5 in the context of the PII manager.

1. The online PII Manager offer

First of all, the main hypothesis is the presence of several PII managers made available to the users. These PII managers are deployed by PII management operators.

2. The user's free choice of one or many PII managers

The users are then able to choose which PII manager(s) they are going to use when interacting with their respective TCPA. There might be an interest for the users to rely on several PII managers instead of only one. The direct advantage of distributing the responsibility for managing the PII over several entities would be higher availability of the service and distributed knowledge about their PII.

3. The dynamic discovery of the users PII managers

We assume that a user's PII managers are dynamically discovered by the URM platform of TCPA at the beginning of the online relationship of the user with the TCPA. This is made possible thanks to the user selecting a PII manager among many on the interface of the TCPA URM platform.

4. A trust model based on a regulated PII manager offer

The PII managers offer is assumed to be regulated. This enables the TCPA URM platforms to establish direct trust with the PII manager, the latter having the critical duties to trustfully select PII sources and validate the retrieved PII. Regulation can be enforced in two different hypothetical ways. First, there might be a regulation for a passlist of PII operators hosting several PII managers. The users would then be asked freely to choose the operator of their PII manager. Second, a PII manager authority, trusted by the TCPA, might organize PII managers based on a hierarchical certification architecture (*i.e.*, a public-key infrastructure).

4.2.2 Technical Hypotheses

Four types of sources are considered and defined as follows:

1. Plain OAuth 2.0 resource servers based on OAuth 2.0 providers or OIDC providers.
2. SAML2 providers [68].
3. Plain read-only REST [30] sources accessible after an HTTP Basic authentication [75].
4. Sources acting as resource servers according to the Kerberos [65] protocol.

4.3 Existing Sources Management Solutions

Table 4.1 gives an overview of existing solutions and provides a comprehensive comparison between them with respect to their support of various functional requirements and other identified

technical considerations. This table also includes the PII manager as this chapter's contribution to meeting the identified functional requirements.

Table 4.1: Related personal data management solutions comparison – excerpt of Tables 3.4 and 3.5 extended with the PII manager contribution

		Related work				*This contribution*
Criterion \ Solution		**INDIGO** architecture [16]	**UMA** [56]	**Databox** architecture [35]	**Fargo** [24]	**PII Manager**
Functional requirements	**Usage definition**	?	●	?	✗	✓
	Consent monitoring	✓	✓	✓	✗	✓
	Usage monitoring	✓	✓	✓	✗	✓
	Delegation capabilities	✓	✓	✓	✗	✓
	PII location abstraction	✓	✗	✓	✗	✓
	Protocol standardization	✓	✓	?	✓	✓
	Access uniformization	✓	✓	✓	✗	✓
	Authz. protocol interoperability	✓	✗	●	✓	✓
Technical considerations	**Identified consent model**	?	✓	?	✗	✓
	Available implementations	✓	✓	✓	✓	✓
	Open specifications	✓	✓	✓	✓	✓

Symbols:
- ✓ → Yes
- ✗ → No
- ● → Depends on implementation
- ? → No information available

4.4 The PII Manager as Part of the TCPA Architecture

This section describes an extension of the generic architecture presented in Section 3.1.1 that includes the PII manager. It therefore aims at filling the functional gaps identified in Section 3.5.

The overall architecture including our PII manager is depicted in Figure 4.2. Subfigure 4.1 illustrates the global architecture involving our PII manager interfacing to the URM systems and the remote sources.

Subfigure 4.2a depicts the interaction between the PII manager and the TCPA URM platforms. This subfigure illustrates the need for a user identifier mapping service, presented in Subsection 4.7.5, as the user already has its own local identifier. For organizational reasons, the TCPA URM platforms maintain their own local-identity manager. The term of "sector border" in this figure denotes the logical separation of identifiers between the TCPA URM platforms.

Subfigure 4.2b shows the interactions between the PII manager and the sources. The drivers, as part of the PII manager's source backend presented in Section 4.8, make it possible to interface with several remote sources. This subfigure illustrates the use of a third-party authorization server (AS)–as part of the OAuth authorization process for OAuth-based sources–for getting the adequate access token for a given resource. Alternatively, sources acting as Kerberos-management resource servers refer to permission tickets that are granted in a two step authorization procedure requiring first to get a ticket granting ticket (TGT) from the key distribution center (KDC) and second to get a permission ticket from the ticket-granting server (TGS).

Eventually, Subfigure 4.2c depicts the user-centric PII management zone, through which the user manages their PII, the authorized sources and their consent to TCPA URM platforms. It corresponds to the direct interactions between the user and the PII Management User Interface of our contribution, presented in Section 4.9.

Therefore the PII manager offers an extension of the OAuth 2.0 authorization protocol. Without implementing a UMA authorization server nor a UMA resource server–some UMA specificities are not required for our use case, *e.g.* interactive claim gathering–, it performs out-of-specification interruptions in the authorization flow in order to support third-party PII sources. Nonetheless, conformance to OAuth 2.0 is maximized in order to increase evolutivity properties of the PII manager, while diminishing implementation complexities–see Chapter 6 for such implementation considerations.

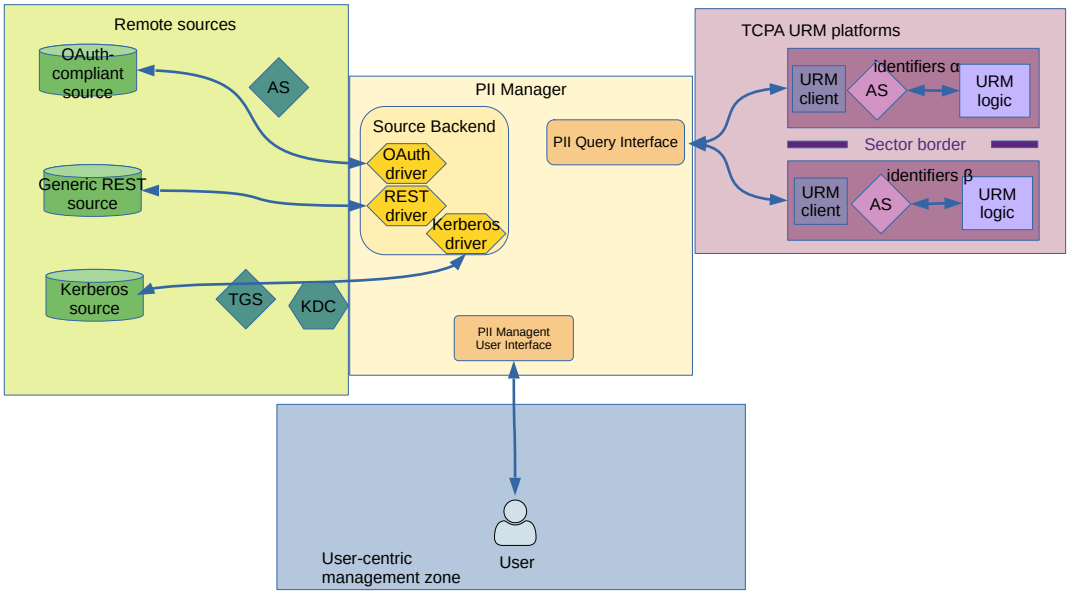
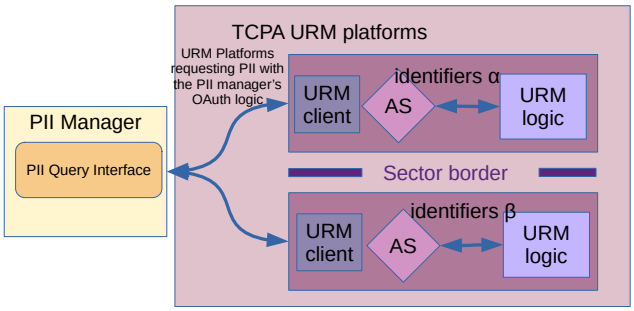
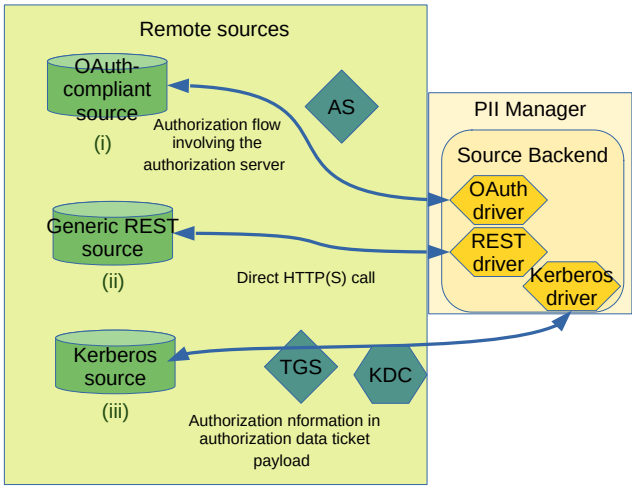


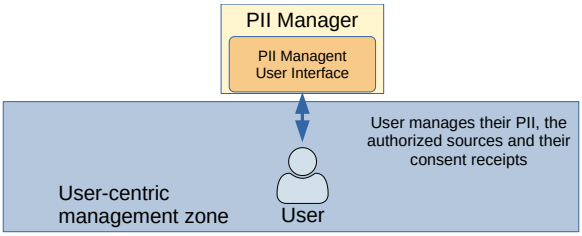
Figure 4.1: General overview of the PII manager



(a) At TCPA URM platform side



(b) At remote sources side



(c) In the user-centric management zone

Figure 4.2: Complementarity in the PII manager's roles regarding our use case entities

We define four main components in the PII manager:

- the PII Query Interface (PQI), presented in Section 4.6, is a means to supporting PII location abstraction and consent management. It proposes a set of endpoints for registration, retrieval and introspection of PII. The standard usage of this interface, and the consent management at this interface level are also described.
- the Core Consent Management (CCM) module, presented in Section 4.7, is responsible for the management and the enforcement of user consent on the PII manager. It does so in spite of the multiple source types and their respective consent models.
- the Source Backend (SB), presented in Section 4.8, enforces the authorization protocol interoperability.

- the PII Management User Interface (PMUI), presented in Section 4.9, manages user consents management. User-definable parameters as part of this interface are also discussed in Section 4.9.2.

We now give an overview of the three main exchanges of the use case:

- The PII Manager discovery;
- The exchanges between the TCPA URM platform and the PII manager;
- The exchanges between the PII manager and the PII sources.

We detail them in the remainder of this section.

The third hypothesis in Section 4.2.1 is the declaration by the users of their PII Managers on URM platform just after they have requested a TCPA service. It means that the registration step is synchronous, as user actions are required. A simple sequence diagram, visible in Figure 4.3, describes the PII manager's discovery by the TCPA URM platforms.

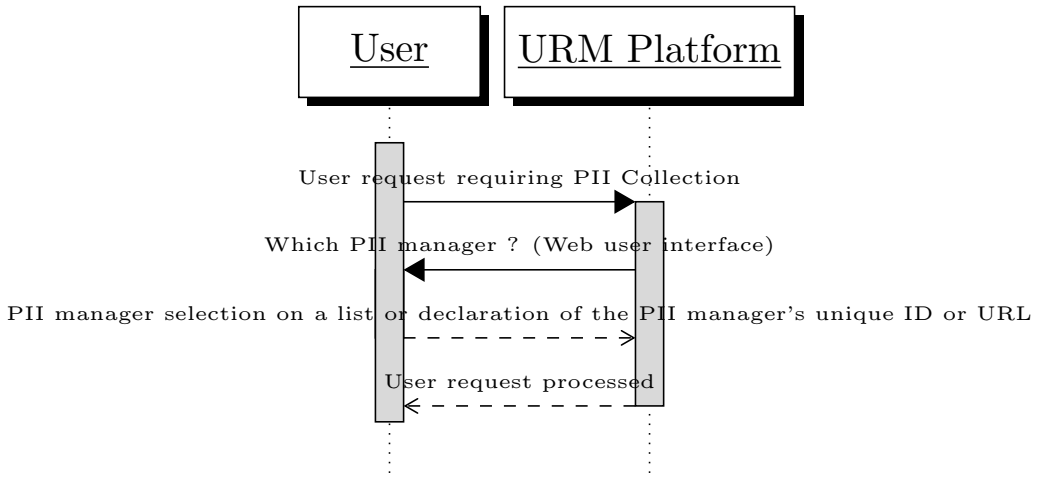


Figure 4.3: Discovery of the PII manager

Figure 4.4 describes the user authentication & consent obtention on the PII manager. Unauthorized PII access requests result in the obtention of a permission ticket. After user authentication & consent obtention, this ticket allows the issuance of an access token with the adequate authorization scopes on the requested resource(s).

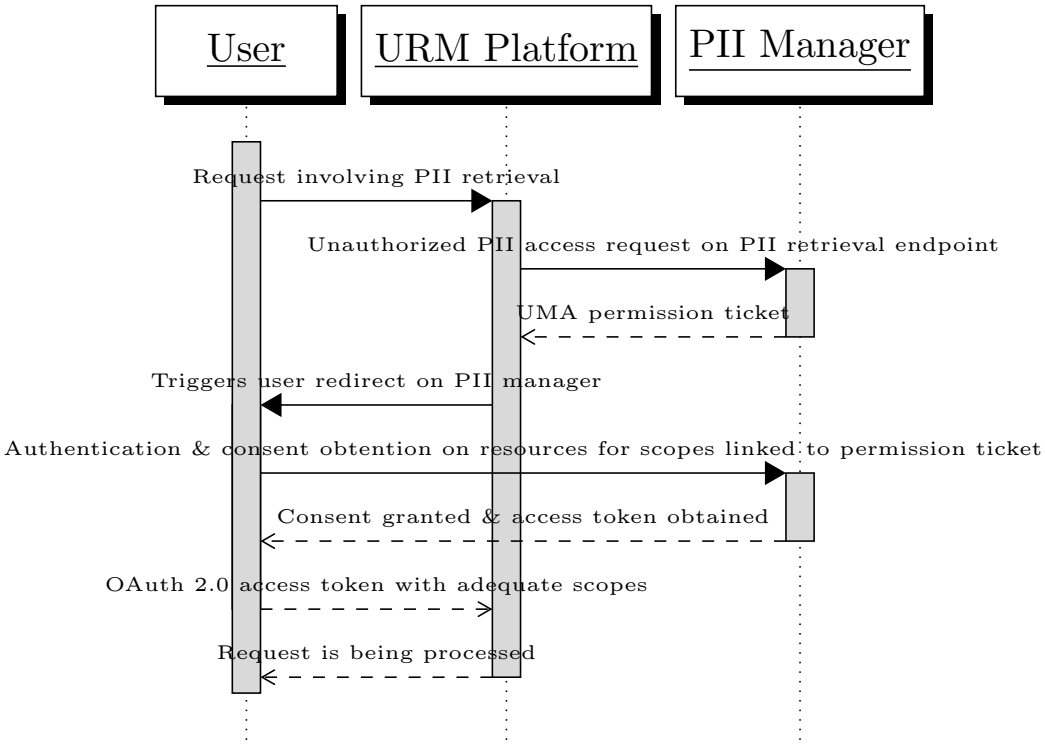


Figure 4.4: User authentication & consent obtention on the PII manager

A sequence diagram for a typical PII collection scenario is provided in Figure 4.5. It shows that the TCPA URM platform interacts directly with our PII manager, regardless of the data sources location. In a two-step process, PII is collected by the TCPA URM platform. First, a request is sent by the TCPA URM platform to our PII manager through the PII retrieval endpoint. Second, the source backend at the PII manager selects the adequate driver for collecting the needed PII from the appropriate remote source. The authorization which is part of the second step, is either synchronous or asynchronous, unbeknownst to the requesting TCPA URM platform.

The PII manager stores authorization information granted by the user. As long as this authorization information still has a valid time-to-live value and the user hasn't revoked the authorization, user-interaction is no longer required.
In the particular case of OAuth 2.0 PII retrieval, our hypotheses are as follows:

- the implicit grant is not supported;
- a first access token has not been obtained;
- the authorization server may or may not issue a refresh token, and if so the refresh token time-to-live value may or may not be sufficient for the PII manager to use. In both cases the diagram below remains relevant.

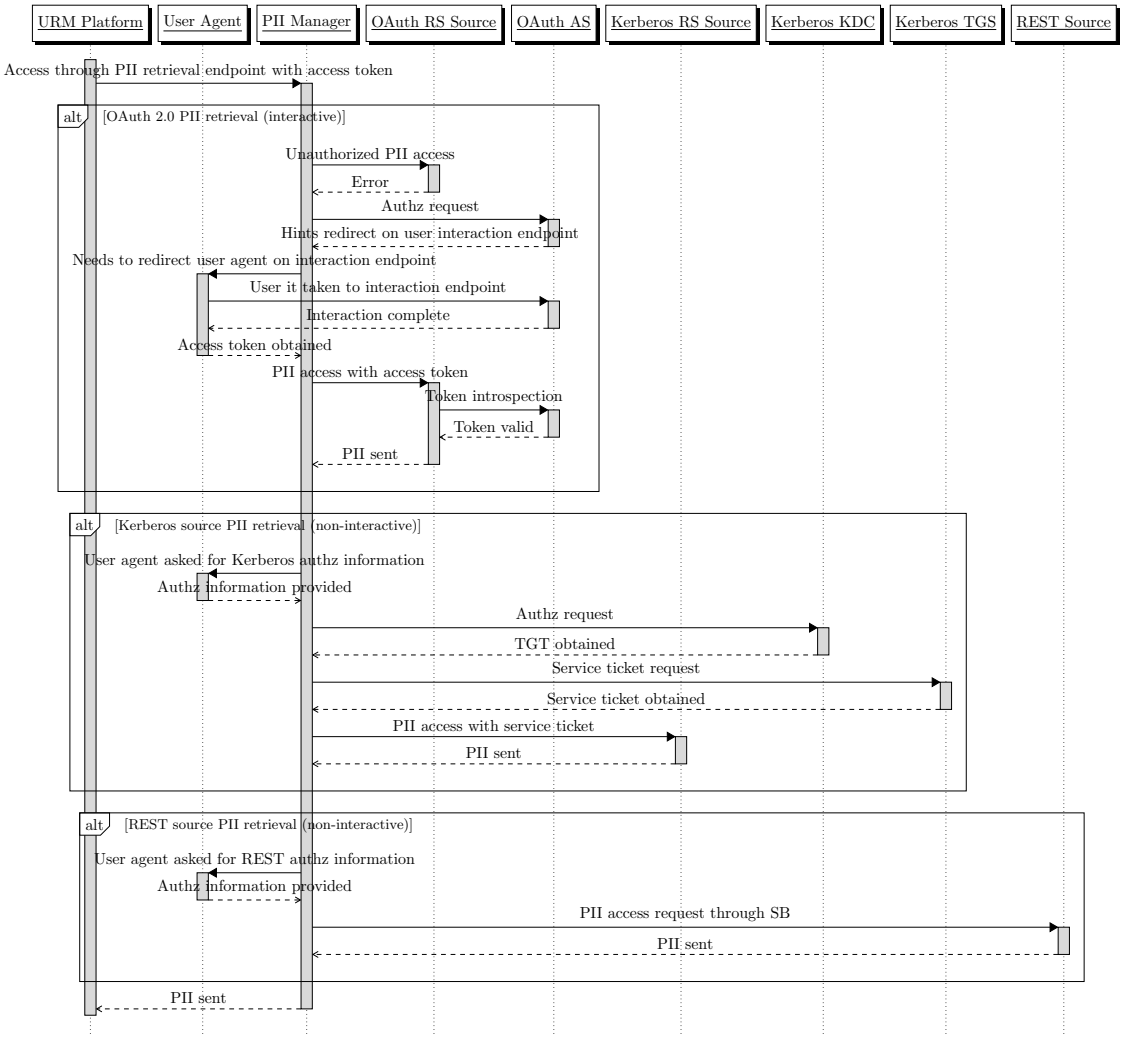


Figure 4.5: PII collection sequence diagram on first source-side user authorization

4.5 Discovery and Registration Processes

4.5.1 PII Manager Discovery by the URM Platform

The discovery of the PII manager is the process that permits the TCPA URM platform to obtain the user PII manager URL. The TCPA URM platform prompts the users with an interface to declare their PII manager–either a list of authorized PII managers, or a text field for the user to declare the PII manager's URL.

From then on, the TCPA URM platform detains the URL of the PII manager of the user. The PII Manager is considered to be trusted as defined in the fourth hypothesis "A trust model based on a regulated PII manager offer" of Subsection 4.2.1.

The TCPA URM platform pursues with the registration phase.

4.5.2 Registration of the URM Platform by the PII Manager

The registration process relies on the OAuth 2.0 dynamic client registration & registration management protocols [81, 80]. The following information is provided by the URM platform while registering to the PII manager:

- *Functional registration information.* It includes terms of the policy, version of the policy terms, the categories of PII that will be collected, and the purpose of the collection. It details all the elements that will be used in consent receipt generation when users decide to give their consent.
- *Technical registration information.* It includes a set of redirection URIs. These URIs will be later used by the PII manager during the PII authorization and PII access process.

In return, the platform is given an identifier (*"client ID"*) and a password (*"client secret"*), necessary for all the future PII access requests. The platform is supposed to securely store these two registration elements, as they are required when issuing PII access requests to the PII manager.

Through the PMUI, at or after registration, the user is also able to define a set of allowed scopes for that TCPA URM platform. This definition eases the authorization flow defined in 4.7 by reducing the set of scopes that require the user's authorization at PII collection time. As explained in Subsection 4.7.3, the use of registration scopes, *i.e.* scopes defined at TCPA URM platform registration time increases the likelihood of the TCPA URM platform being granted the authorization to collect the user PII. Registration scopes mean that the user trusts the TCPA URM platform for PII collection regarding these particular scopes of authorization.

4.6 The PII Query Interface (PQI)

4.6.1 Overview

The PII Query Interface is used by the TCPA URM services to retrieve the user's PII on the PII manager. This section presenting the PQI is organized as follows:

- First, an overview of the PQI endpoints.
- Second, a base usage description of the PQI.
- Third, a presentation of the PII retrieval endpoint.
- Fourth, a presentation of PII metadata introspection endpoint.
- Finally, a description of the PII directory service.

4.6.2 Presentation of the PQI Endpoints

The PQI is used by the TCPA URM platform when issuing requests to the PII manager. The PII manager exposes four endpoints:

- The *PII retrieval* endpoint, complying with standard OAuth 2.0 scenarios for a resource server. It performs the reduction of OAuth scopes, as mentioned in [56, Section 3.3.4]. This PII retrieval endpoint complies with the OAuth 2.0 protocol with the UMA grant. The authorization process that is part of the PQI relies on the generation of consent receipts.

- The *token* endpoint on which the TCPA URM platforms obtains an access token provided that the user's consent has been granted.
- The *PII metadata introspection* endpoint, legitimate for services that cannot, or do not want to, access the actual PII content but instead be provided a set of metadata regarding this piece of PII. Metadata include creation and modification information and user consents granted to the requesting service for that PII.
  User consent management is necessary to ensure that later offline authorization flows can be granted to the service.
- The *PII directory* service, exposing a restricted access to the users' available PII on the PII manager.

4.6.3 Base Usage Description

The PQI, for collecting PII from a PII source is illustrated in Figures 4.6,4.7 (interactive mode) and Figures 4.8,4.9(non interactive mode).

Figures 4.6 and 4.7 illustrate the PII access flow for interactive source types. The interactions with the user happen through the user agent and are based on HTTP redirections. The interactive source types are OAuth 2.0 and SAML.

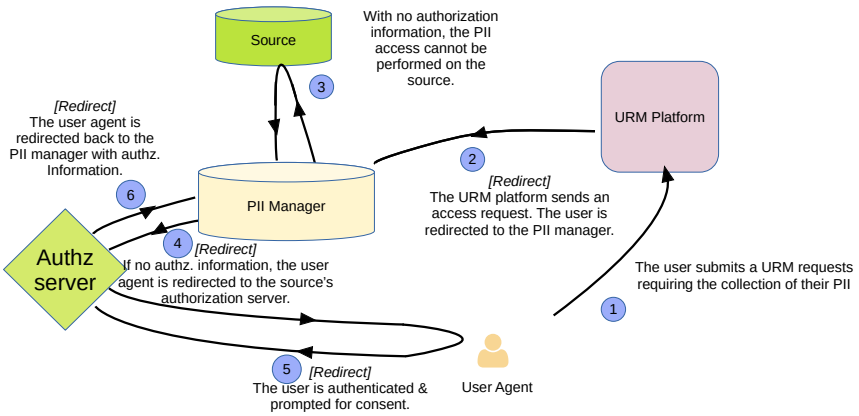


Figure 4.6: Interactive source authz. information retrieval

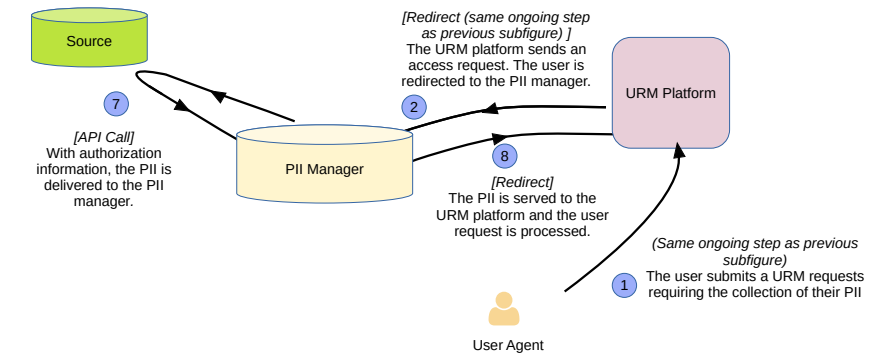


Figure 4.7: Interactive PII collection from the source by the PII manager

Conversely, Figures 4.8 and 4.9 illustrate the PII access flow for strictly non interactive source types. The strictly non interactive PII collection process is based on API calls. These strictly non interactive source types are REST and Kerberos.

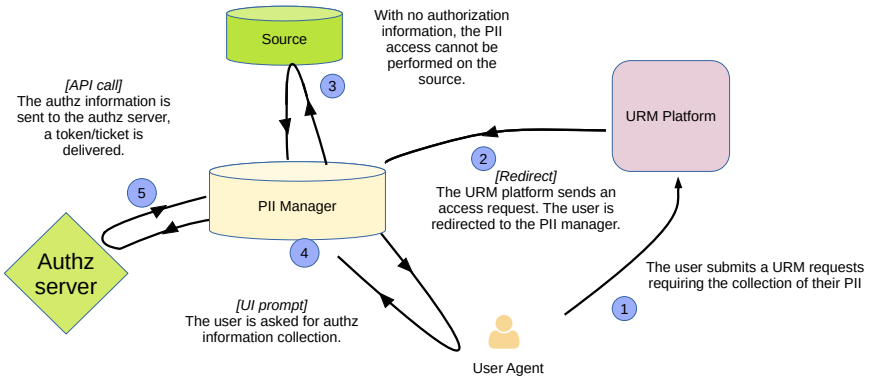


Figure 4.8: Non interactive source authz. information retrieval

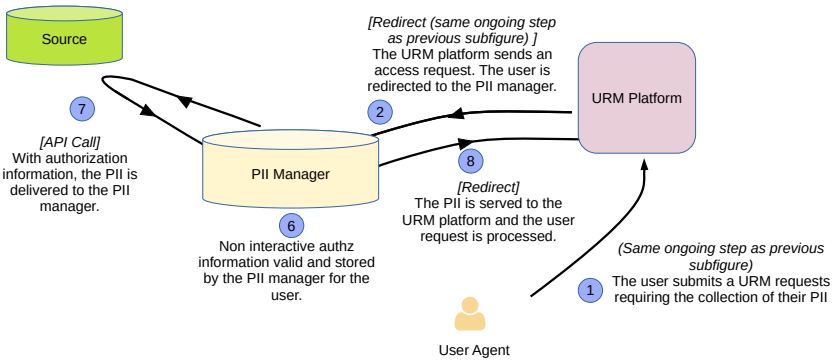


Figure 4.9: Non interactive PII collection from the source by the PII manager

In both cases, the PII retrieval happens as follows:

- Once the PII manager is discovered by the TCPA URM platform, the TCPA URM platform is registered as an OAuth 2.0 client with regard to the PII manager–see Section 4.5.
- The URM platform issues a PII collection request on the PII manager's PII retrieval endpoint.
- If the request contains a valid access token, it succeeds and the PII manager response contains the resource.
- If the request doesn't contain a valid access token, URM platform may not have been granted the user's consent to collect their PII. The necessary steps are as follows:
  - The PII manager, after determining the required scopes for the requested PII collection, replies with a permission ticket. This ticket is meant for the TCPA URM platform to request an access token on the PII manager's token endpoint. The way the PII manager determines the scopes of authorization for the PII collection is explained in Section 4.8.
  - The URM platform issues an access token request on the PII manager's *token* endpoint. This request declares the scopes required for collecting the requested PII.
  - If the PII manager doesn't possess the user authorization information with regard to the actual PII with the designated scopes:
    * The user, through their user agent, is redirected to the PII manager.
    * User authentication happens on the PII manager. This authentication may be a single-sign on, relying on a third-party IdP, and may require a user account creation if the user is not registered yet.

  - If the source type is interactive (OAuth 2.0, SAML2):
    - A HTTP redirect-based flow happens on the PII source, in which the PII manager acts as a client with regard to the source.
    - It is followed by the storage of authorization information for the PII source on the PII manager. It is therefore assumed that the authorization information given by the PII source (access token for OAuth 2.0 sources, credentials to perform SOAP-based backchannel PII collection for SAML sources) has a relevant expiration timestamp.
  - If the source type is not interactive (REST, Kerberos):
    - The user authorization information gathering process happens on the PII manager, it is stored for subsequent uses. It depends on the source type.
    - If the source type is REST with a plain HTTP authentication scheme, the user is asked to input their credentials.
    - If the source type is Kerberos, the PII manager, through the user agent, asks the collection of the Kerberos permission ticket.

    As for interactive source types, it is assumed that this collected authorization information with regard to the PII source has a meaningful expiration timestamp.
  - User identifiers of the source and the PII manager need mapping. The way the mapping is performed is out of scope, though discussed in this chapter's conclusion, in Section 4.7.5.
  - If the user gives their consent, an access token is issued, and a consent receipt is generated as described in Section 4.8.
- The PII manager redirects the user agent back to the TCPA URM platform, along with the previously obtained valid access token.
- The TCPA URM platform renews the PII collection request, this time with an access token.
- The previously stored authorization information on the PII manager is presented to the PII source and the PII is retrieved by the PII manager.
- The PII manager's answer to the TCPA URM platform contains the PII.

4.6.4 The PII Retrieval Endpoint

The basic use of the PII retrieval endpoint has been presented, it is now possible to present its protocolar specificities.

The PII retrieval endpoint is OAuth 2.0 compliant [36, Section 3] in which the PII manager is both the authorization and the resource server with regard to the TCPA URM platforms. It uses the User-Managed Access grant type.

The authorization grant happens as follows:

- The obtention of a permission ticket (UMA extension of the OAuth 2.0 authorization code–[36, Section 4.1]) as part of a first unauthorized request. The unauthorized request of the TCPA URM platform to the PII manager is as follows:

```
GET /resources/?ids=<list of comma-separated resource identifiers> HTTP/1.1
Accept: application/json
Host: provider.mycity.fr
```

Or simply when requesting a single resource:

```
GET /resource/<resource identifiers>/ HTTP/1.1
Accept: application/json
Host: provider.mycity.fr
```

Since the request is unauthorized, the response of the PII manager to the TCPA URM platform contains the permission ticket and is as follows:

```
HTTP/1.1 401 Unauthorized WWW-Authenticate: UMA
ticket="<ticket unique identifier>"
```

The ticket content is just a machine-readable unique identifier to keep track of the TCPA URM platforms authorization requests on the PII manager[1].

- Authn & consent obtention.
  The request of the TCPA URM platform to the PII manager contains the permission ticket and is as follows:

```
POST /token/ HTTP/1.1
Accept: application/json
Host: provider.mycity.fr

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Auma-ticket
    &ticket="<ticket unique identifier>"
```

Alternatively, if the TCPA URM platform not only wants to collect the PII but also to modify it (resulting in modification on the source), it must add the `write` authorization scope. In this case, the TCPA URM platform request to the PII manager is:

```
POST /token/ HTTP/1.1
Accept: application/json
Host: provider.mycity.fr

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Auma-ticket
    &ticket="<ticket unique identifier>"
    &scopes=write
```

The response of the PII manager contains an error with the `need_redirect` error code. The user, through their user agent, is redirected to the PII manager's *authentication & consent obtention* endpoint where they are authenticated and prompted for consent on the requested resource. The HTTP request issued as part of the redirection on the endpoint is as follows (UMA compliant [56, Section 3.3.2]:

```
GET /rqp_claims?client_id=<tcpa urm client identifier>
    &ticket=<ticket unique identifier>
    &claims_redirect_uri=https%3A%2F%2Fprovider.mycity.fr%2Fcallback%2F
    HTTP/1.1
```

[1] It is also useful in a standard UMA architecture where the authorization server and resource server are two distinct entities.

During the consent obtention phase, the ways the scopes of authorization for the resource are determined and the consent receipt is generated are explained in Section 4.8.
Once the user consent has been obtained, the user is redirected to the TCPA URM platform, at the URL declared in `claim_redirect_uri` and with the permission ticket. The PII manager response is then as follows (UMA compliant–[56, Section 3.3.3]):

```
HTTP/1.1 302 Found Location: https://provider.mycity.fr/callback/
    ?ticket=<ticket unique identifier>
```

- The access token[2] upon acceptance of the permission ticket. If the TCPA URM platform request–represented by the permission ticket–is accepted, the response of the PII manager to the TCPA URM platform contains the access token and is as follows (OAuth 2.0 compliant):

```
HTTP/1.1 200 OK
Content-Type: application/json
[...]

{
    "access_token": "<access token value>",
    "token_type": "Bearer"
}
```

- The new PII collection request, on the `/resources/` endpoint. The request of the TCPA URM platform to the PII manager contains the access token and is as follows:

```
GET /resources/<resource identifier> HTTP/1.1
Accept: application/json
Host: provider.mycity.fr
Authorization: Bearer <access token value>
```

The response of the PII manager to the TCPA URM platform contains the requested resource and is as follows:

```
HTTP/1.1 200 OK
Content-Type: application/json
[...]

{
    "resources": [{
        "identifier": "<resource identifier>",
        "value": "<(possibly-serialized) resource value>",
        "type": "<resource type>"
    },{
        "identifier": "<second resource identifier",
        "value": "<(possibly-serialized) second resource value>",
        "type": "<resource type>"
    }]
}
```

[2] *Requesting party token* as per the User-Managed Access terminology.

4.6.5 The PII Metadata Introspection Endpoint

The PII metadata introspection endpoint is OAuth 2.0 compliant. The authorization process is the same as 4.6.4, but the access token is presented on the `/metadata/` endpoint, followed by the resource identifier.

Once having obtained a valid access token, the TCPA URM platform request to the PII manager is as follows:

```
HTTP/1.1 GET /metadata/<resource identifier>/
Content-Type: application/json
Accept: application/json
Host: provider.mycity.fr
Authorization: Bearer <access token value>
```

In return, the PII manager response to the TCPA URM platform is as follows:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "metadata": {
        "created": "...",
        "owner_id": [...],
        "last_modified": "..."
    },
    "err": 0
}
```

4.6.6 The PII Directory Service

Using the PII retrieval and the PII metadata introspection endpoints requires that the TCPA URM platforms know which user PII is available on the PII manager. Exposing a list of the user's available PII is the role of the PII directory service.

The PII directory service is also OAuth-2.0-authorization managed, and requires the obtention of a valid token. Just like the PII metadata introspection endpoint, the TCPA URM platform request to this service offered by the PII manager is as follows:

```
HTTP/1.1 GET /pii-directory/
Content-Type: application/json
Accept: application/json
Host: provider.mycity.fr
Authorization: Bearer <access token value>
```

In return, the PII manager response to the TCPA URM platform is a follows:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "data": [
        {
            "identifier": "abc-resource1",
            "name": "resource1",
            "type": "sometype1",
```

```
            "created": "...",
            "owner_id": [...],
            "last_modified": "..."
        },
        {
            "identifier": "xyz-resource2",
            "name": "resource2",
            "type": "sometype2",
            "created": "...",
            "owner_id": [...],
            "last_modified: "..."
        }
    ],
    "err": 0
}
```

where `owner_id` is the resource owner identifier within the corresponding TCPA URM platform user base as known by the PII manager.

The way the PII directory is provisioned pertains to the Source Backend, and is explained in Section 4.8.5.

4.7 Core Consent Management (CCM)

4.7.1 Presentation

The main objective of consent management is the respect of the user's choices when it comes to considering the TCPA URM platforms' PII queries.

The authorization system of the PII manager hence relies on such consent management. The authorization lifecycle is therefore managed by the user.

In particular, consent management on the architecture implies keeping track of the users' previous choices regarding the collection of their PII by service providers. The user's consents have a limited lifetime–in particular they can be given for a single immediate collection–and they have scope denoting the extent of the granted authorization.

The CCM is responsible for managing consent receipts as defined in [47], keeping track of user-given consent regarding the collection of their PII by some TCPA URM platforms on the PII manager. The information contained in the consent receipts belongs to three categories: (i) receipt transaction fields, (ii) transaction parties fields and (iii) data, collection and use fields. We insist on the fact that using consent receipts for the PII manager does not increase the complexity on the user side and on the TCPA platform side. This data structure is solely used by the PII manager for its inner consent management logic.

All three categories are relevant to enforce consent management as per our consent model, which isi directly derived from the structure of receipts in [47]. In particular, transaction parties fields enable the declaration of TCPA URM platforms and PSPs allowed to collect the user's PII, and whether this collection happens on behalf of entity. Additionally, data, collection and use fields enable the declaration of termination policies.

4.7.2 Comparison of Existing Consent Models

A comparison table of the consent models supported by the PII manager is shown in Table 4.2.

The following criteria are used to make the comparison:

- Terms-of-usage versioning, referring to the ability to record the version number of the terms-of-usage for the PII collection that has obtained the user's consent.
- Direct verifiability, meaning whether the verifiability information requires a request to an authorization server or can be *directly* verified by the resource server.
- Authorization scope, meaning the ability to specify unitary elements defining the authorization request.
- Revocability, meaning the ability to cancel a previously given consent.
- Multi-domain management, which is relevant when, as depicted in Figure 4.2a, several TCPA URM platforms interface with a single PII manager. The management of consent information across domains is therefore a key element.
- Inter-service resource sharing, defining whether the consent information specifies the ability for other services to directly access the PII.

Table 4.2: A comprehensive comparison between different consent models

Consent model / Criterion	**PII manager consent receipt (Kantara)**	**OAuth 2.0 access token**	**Kerberos ticket**	**Standard ACLs**
Terms-of-usage versioning	Yes	No	No	No
Direct verifiability	Yes, by definition of locally managed consent	Yes	No	Depends on model
Authz. scope	Yes	Yes	No	Partially supported
Revocability	Yes, by definition of locally managed consent	Yes, through AS	Yes	Yes
Multi-domain	Yes	Yes	No	Depends on model
Inter-service resource sharing	Yes	No	Yes	Depends on model

4.7.3 Generating and Managing Consent Receipts

The purpose of generating and managing consent receipts is the management of authorizations of the TCPA URM platforms to access the users' PII. With regard to the PII manager, the authorizations directly translate to consent, hence the use of consent receipts. User consent management through consent receipts enables (i) the delegation of access decisions on the PII manager and (ii) the generation of consent receipts for traceability purposes. Properties (i) and (ii) have both undergone specification efforts by the Kantara Initiative. Through the generation and management of consent receipts, the PII manager helps the users manage the lifecycle of authorizations. Indeed, consent information can be a one-time grant, or can have a limited lifetime for usage in several PII collections. A consent is given by the user on a set of PII with an associated set of scopes of authorization. This concept of scopes, component of the OAuth 2.0 (see [36, Section 3.3]), is also part of the consent receipt model of the PII manager. The consent receipt model is therefore the formalization of authorization information within the PII manager's consent management core module. It embodies its access-control list (ACL) model.

The access delegation decision, based on OAuth 2.0 scope-based access requests, is based on [56, Section 3.3.4]. Indeed, UMA specifies the delegation of access decisions. The decision algorithm deals with the way the UMA server should decide on whether to grant access to users. Given a TCPA URM platform C, the input information for this algorithm to be run is registration scopes for C, recently requested scopes for C and (OAuth) *scopes* associated with the resources requested by C.

The algorithm is as follows (compliant with [56]):

- Gather *Reg* the set of registration scopes for C. These scopes have been user-defined at or after TCPA URM platform registration.
- Gather *Req* the scopes that C just requested at the token endpoint.
- Retrieve the requested resources from the permission ticket supplied by the client.
- For each resource i among these retrieved resources.
  - Gather the scopes S_i associated with that resource.
  - Determine $T_i = S_i \cup (Req \cap Reg)$
- For each of these T_i
  - According to the user's current consent receipts, evaluate the authorization status of the scopes set T_i.
  - Any scope having validated the authorization status is transferred to the set of candidate granted scopes U_i.

With this input information, the authorization server evaluates which scopes of authorization can be granted to the client. Therefore, three cases are possible: the authorization server to either grant the authorization with the actual scopes as requested by the client, to restrict the authorization grant to a smaller set of scopes, or to completely deny the authorization request.

The PII verification algorithm is based on the consent model given in [47] and can be formulated as the verification by the PII manager:

- of the issuance date.
- of the expiry date.
- that the terms-of-use version applies.
- that the consent geographical location applies.
- of the scopes according to previously-granted scopes for category of service as follows:
  - Retrieve the set of previously granted scopes as defined in [56, Section 3.3.4].
  - Translation of authorization information into OAuth scopes known to the PII manager. Claims and scopes are checked against the rules defined for this URM client.The two main elements involved in this process are:
    - The translation of the source's authorization information into scopes that are known to the authorization server. This translation step depends on the type of source, as explained in Section 4.8.
    - The comparison of the required scopes with the user-defined preferences.
  - When user defined preferences are insufficient in order to take action, the required scopes are also compared to the scopes previously granted to the URM client. Based on the UMA OAuth 2.0 grant access control process provided in [56], the server can decide to reduce the required scopes to a set of scopes that are appropriate regarding the URM client and the requested resource. If the authorization server chooses to reduce the set of granted scopes–in comparison with the requested scopes–then a reverse translation is necessary: the OAuth scopes known to the PII manager are reverse-translated to the OAuth authorization information model as dealt with by the requesting party. Alternatively, the PII manager acting as an authorization server may reject the URM client's request.

When a new user consent is given, a consent receipt is generated on the PII manager. The relevant fields of the consent receipt are:

- The incremental version of receipt, in case the receipt is later modified by the user;
- The timestamp at which the consent was given;

- The collection method used for obtaining the consent: in our use case the consent is only obtained via the PII manager's PMUI;
- A unique identifier for the receipt;
- The language in which the consent was obtained;
- The identifier of the user having given their consent;
- The TCPA for which the consent was given;
- The identifier of the first TCPA that collects the PII;
- A boolean value indicating whether that TCPA acts on behalf of another one;
- The TCPA URM platform for which the PII is collected;
- The list of purposes for which the PII is collected. This list is provided by the TCPA URM platform at permission ticket obtention time;
- The consent type: in our use case, only explicit consent are granted;
- A link to the consent termination policy of the TCPA URM platform. The TCPA URM platform commits to invalidate the user consent which was granted to it, whenever that termination policy states so;
- If the PII is disclosed to one or several third-parties (TCPA-related PSPs for instance), a list of their names;
- A boolean value indicating whether this consent receipt was granted on sensitive PII: this is true, for instance, for health-related PII.

4.7.4 In Summary: Example of authorization flow

While accessing a REST source, the delegated authorization flow is the following one, as summarized in Figure 4.10:

1. **URM client request**: the URM (OAuth) client asks for accessing to resource `resources/catering_fees.html` associated with the `read write print caption` scopes;
2. **Translation**: the client requested scopes translate to the ability to perform `GET`, `POST` and `PATCH` on the URI by the PII Manager;
3. **Reduction** (as described in Section 4.6.2): the PII manager gets from its own internal authorization information that only the `GET` verb is authorized for that resource and this URM client.
4. **Reverse translation**: the PII manager reverse-translates to the `read` (OAuth) scope.
5. **PII manager response**: the PII manager sends an access token with the reduced `read` scope.

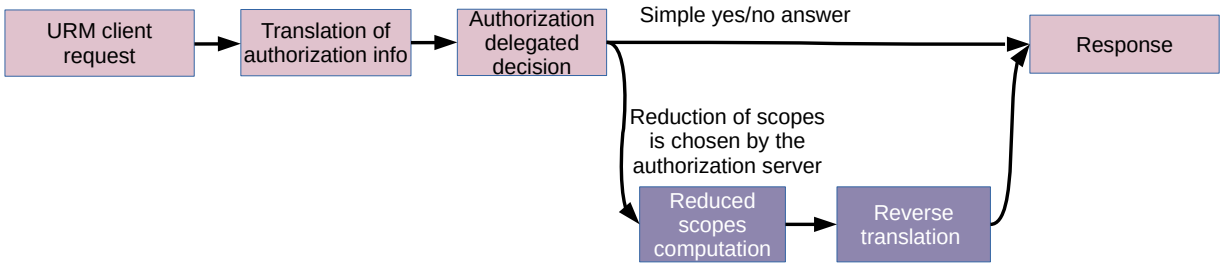


Figure 4.10: PII collection by the URM client — extension of the UMA decision process by the PII manager

4.7.5 Considerations Regarding User-Identity Mapping and Matching

4.7.5.1 Incentives for Identity-Mapping & Matching

In order to manage the users' consent, the PII manager takes responsibility for implementing the user identifier mapping across the sources and the TCPA URM platforms. It is of utmost importance that the collected PII from different sources for one user do actually belong to that user. In order to enforce this user uniqueness, the PII may rely on several possibilities, be it either a unique official identity sources such as *FranceConnect*, or partially-automated matching–see [59]. Thus it is important to understand that the technical necessity of user-identity *mapping* is verified thanks to a *matching* procedure.

4.7.5.2 Performing Identity-Mapping and Verifying with Identity-Matching

The TCPA deploys an identity provider that complies with the identity-federation principles. In particular, it provides sector identifiers according to several service sectors of the TCPA.

When it comes to user authentication on the PII manager, several options are possible:

1. The PII manager can perform standalone authentication, without relying on an external IdP. This is for instance the option chosen for our proof-of-concept implementation, presented in Chapter 6.
2. The PII manager can rely on one external IdP, in which case we can safely assume that a single user gets assigned a single, unchanging identifier during authentication. This IdP may be the TCPA URM platform's.
3. The PII manager can rely on several IdPs, in which case a user, registered to at least two of these IdPs, may end up with several identifiers on the PII manager after authentication, depending on the chosen IdP.

Similarly, the identifiers provided by the PII sources for a same user vary from one source to another.
In both cases (user authentication on the PII manager; identifiers provided by the source), performing and managing user identifier mapping is necessary. The PII manager is thus responsible for managing such user identifier mapping across the services. First, the authorization information from a first TCPA URM platform, say platform α, may present the user information including an identifier u_α. A second platform β only knows the authorization information that contains a user identifier u_β. If the user's consent is applied on platforms α and β, a user-identifier mapping must be performed. The PII manager needs to provide a mapping function m:

$$\begin{aligned} m : & \\ & U_\alpha^* \longrightarrow U_\beta^* \\ & u_\alpha \longmapsto u_\beta \end{aligned} \tag{4.1}$$

where U_α^* and U_β^* are respectively the identifiers sets of platforms α and β. Since those platforms support the OAuth authorization framework, these identifiers can be [i] pseudonyms, [ii] Universally Unique Identifiers (UUIDs) or [iii] human-friendly attributes such as the user's email addresses. Regardless of the platforms' actual identifier policy, these identifiers are considered to be string characters.

Additionally, the PII manager acting as a resource server interacts with several authorization servers that do not belong to the same federated-identity environment. They therefore do not share the same user identifiers.

The identifier is mapped to a pseudonym derived from (a) the source subject identifier and (b) the target sector identifier, in a similar fashion as presented in [83]. The process used to derive these pseudonyms can rely on non-reversible pseudonyms identifier generation as presented in [83, Section 8.1].

The TCPA URM platform then receives a pseudonym with a set of human-readable information that serves as input for the identity matching procedure, as presented in [59, Chapter 5].

4.8 The Source Backend (SB)

4.8.1 Overview

The source backend makes it possible to deal with multiple sources, with a backend for each source type. However, the multiplicity of such authorization protocols makes it hard to identify a unified consent model. The source backend nonetheless enables interoperability through the support of multiple authorization and PII access protocols.

This section also discusses consent management as part of the source backend when interfacing with remote sources. For each type of sources, a study of the authorization information structure is given. Strategies to map authorization information to our consent model are defined.

4.8.2 Preliminary Definition

We now need to define the concept of OAuth authorization scope, which is a recurring term throughout this section. An OAuth scope is an authorization unit, characterizing a resource or an action to be performed on it. In OAuth terms, it is a character string of blankspace separated keywords that constitute the scopes set. The set of scopes is used in the PQI retrieval endpoints to specify the PII requested.

4.8.3 Supported Sources Types

The list of supported sources types is as follows:

- An **OAuth 2.0 (including OIDC) provider** does not require any translation, as it is directly usable by the UMA authorization process used within the TCPA URM platform. The scopes used by the OIDC identification protocol are directly compatible with UMA.
- A **SAML provider** translation mainly relies on the use of the OAuth 2.0 assertion framework [14] and its use with the translation of SAML 2.0 assertions [13]. These specifications provide "out-of-the-box" processes for translating SAML assertions to OAuth 2.0 authorization information. Thus SAML assertions can be used either as OAuth 2.0 client authentication information or authorization grants.
- The **REST sources** only require a static set of scopes predefined by the TCPA URM platform administrator. A direct mapping between HTTP verbs as used by REST sources and standard OAuth scopes used by UMA can be established.
- The resource server operating according to the **Kerberos** authorization protocol needs a valid permission ticket. The ticket scope always concerns access to a resource. Finer scopes of authorizations are not supported by the protocol[3]. The way the PII manager manages the Kerberos session key, and the manager's registration in the principals database maintained by the Kerberos administration server, is out of scope of this chapter.

4.8.4 Source Registration

Just like the PII manager registers the authorized TCPA URM platforms (as OAuth 2.0 clients, as explained in Section 4.5.2), it also offers PII source registration. This registration, although not mandatory, is useful for provisioning the PII directory (Section 4.8.5.

The source registration is performed dynamically thanks to the *source registration* endpoint. A source operator willing to dynamically register its set of sources on the PII manager needs to obtain credential from the PII manager administrator.

[3]Kerberos tickets contain an optional `authorization-data` field that can be used to implement authorization scope support. However it is not covered by the specification–see [65, Section 5.3].

Once the credentials have been obtained, the registration request is as follows:

```
POST /token/ HTTP/1.1
Accept: application/json
Host: pii.somesource.fr

name=somesource_pii_provier
    &client_type=oauth
    &token_endpoint=<encoded uri of the token endpoint>
    &authz_endponit=<encoded uri of the authorization endpoint>
    &directory_provisioning_method=push
```

In return, the PII manager request replies with a source identifier and a confirmation of the registration:

```
HTTP/1.1 201 Created
Content-Type: application/json

{
    "data": {
        "source_identifer": "pii_somesource_fr",
        "push_uri": "https://mypiimanager.fr/pii-directory/"
        "push_authz: "Basic",
        "credentials": "<Basic credentials for push authorization>"
    },
    "err": 0
}
```

4.8.5 PII Directory Provisioning

The PII directory provisioning is necessary to expose a PII directory service to the TCPA URM platforms–see Section 4.6.6.

Two options are possible in order to provision the PII directory. These two options are, respectively:

- The PII manager exposing a push endpoint.
- For supported sources types, at source registration–Section 4.8.4–, the declaration of a PII listing endpoint. That PII listing endpoint makes it possible for the PII manager to interface the source in order to maintain its PII directory. New PII can then be added to the directory, and deleted PII can be removed. While some sources types such as REST support this service natively, other source types entirely rely on the implementation for the support of that service.For instance, the OAuth 2.0 resource server endpoints are out of scope of the OAuth 2.0 specification.

The PII directory stores the following information regarding each PII:

- The owner identifier after performing identifier mapping, see Section 4.7.5.
- The creation timestamp.
- The last modification timestamp.
- The user-friendly name.

Since the authorization information for the PII is subject to changes over time, it is not cached and not included in the PII directory.

Upon PII directory provisioning, the PII manager generates a unique identifier for each PII. This identifier is the identifier exposed by the PII directory service. The PII directory maintained by the PII manager also contains the identifier PII source from which the PII originates. This PII source identifier, determined at source registration (Section 4.8.4), is a technical information for the articulation of the PQI and the SB, and is therefore not exposed by the PII directory service (Section 4.6.6.

Subsections 4.8.5.1 – 4.8.5.4 deal with the translation of authorization & consent information for the PII manager. This translation is non trivial as (i) the supported protocols offer several authorization flows, (ii) the actual implementation of such protocols may cover out-of-specification behavior that still needs to be included in the translation, (iii) such translation is made as invisible to the users and the the TCPA URM platform as possible, and (iv) its modular layout enables the PII manager to evolve and potentially support upcoming protocols.

4.8.5.1 Translation to OAuth (including OIDC) Consent Information

4.8.5.1.1 Structure of authorization information

Access tokens–in their most common JWT form–are structured as follows[4]:

- A header bearing token metadata.
- A (cleartext or encrypted) payload, which contains the core authorization information.
- Optionally, a signature whose validation information is contained in the header.

4.8.5.1.2 Grant type translation

Grant type translation happens when using standard-OAuth input authorization information within the (UMA-OAuth) PII manager. The OAuth supported grant types are the implicit grant and the authorization code grant. The authorization code grant is a two-step grant allowing the URM client to request access tokens, by letting the authorization server know that it was giving the user's authorization. On the other hand, the implicit grant is simpler as no authorization code is involved. The user's consent given to URM client directly results in the authorization server's response that includes an access token.

4.8.5.1.3 Scope translation

Translating scopes is performed in three steps:

1. Identify the scopes of interest for the TCPA URM platform.
2. List all the other scopes that can be part of the translated assertion.
3. For each of these other scopes, provide a mapping to the scopes supported by the TCPA URM platform.

4.8.5.1.4 The OIDC profile

The only supported grant type is the authorization code. OIDC is a simplified version of OAuth, in which the identity provider is also the resource server (in particular, the identity information *is* the requested resource).

4.8.5.1.5 Direct mapping

The direct mapping from the PII manager's consent model to OAuth authorization information happens as follows:

1. Issuer to `iss`.

[4]The structure may vary when the JSON token is encrypted and when it contains unprotected header fields.

2. Start timestamp to `iat`.
3. Expiry timestamp to `exp`.
4. Authz. resource to the resource URI of the authorization process.
5. Authz. scope to the `scopes`.
6. Authz. user identifier to `sub`.

4.8.5.2 Translation to Kerberos Consent Information

4.8.5.2.1 Structure of authorization information

The `authorization` field within a permission ticket bears the authorization information. The use of this field is implementation-dependent; the Kerberos v5 protocol does not specify a structure for this `authorization` field.

4.8.5.2.2 Direct mapping

The direct mapping from the PII manager's consent model to Kerberos' permission ticket information happens as follows:

- Issuer to `cname` and `crealm`.
- Start timestamp to `starttime`.
- Expiry timestamp to `endtime`.
- Authz. resource to authorization data payload of the ticket.
- Authz. user identifier to `principal`.

4.8.5.3 Translation to Plain ACL Consent Information

4.8.5.3.1 Structure of authorization information

According to our hypotheses, plain ACLs' consent information is made of [i] authorization metadata (*e.g.*, temporal and spatial validity metadata) and [ii] core authorization information (*e.g.*, subject and object of authorization).

4.8.5.3.2 Direct mapping

The PII manager's consent model maps to the ACL information. For instance, the following mapping applies:

1. Issuer maps to `client`.
2. Start timestamp maps to `beginson`.
3. Expiry timestamp maps to `endson`.
4. Authz. resource maps to `resources-uris`.
5. Authz. user identifier maps to `subject`.
6. Delegation flag maps to `delegated`.

4.8.5.4 Translation to SAML Assertions

As explained in chapter 5, a SAML assertion is the authorization information exchanged between SAML entities, most commonly between a Service Provider and an Identity Provider. This exchange can happen at Single-Sign On (SSO) time, and optionally through SOAP-based communication backchannel between SPs and IdPs.

4.8.5.4.1 Structure of authorization information

The PII fields of a SAML assertion are:

- The subject of the authorization information (`Subject`):
  - A technical identifier (`NameID`).
  - A set of human-friendly PII about the subject.
  - Validation metadata.
- The authorization:
  - The assertion statement.
  - Additional assertion validation metadata.

When the assertion is signed or encrypted, the public keys and algorithm declarations are available in the server's and service provider's respective metadata [15].

4.8.5.4.2 Direct mapping

The SAML assertion translation specification of RFC 7522 [13], as part of the OAuth 2.0 assertion framework presented in RFC 7521 [14], is used for the SAML driver. RFC 7522 [13] specifies the way SAML assertions can be used as authorization grants or as client authentication information.

Using this framework means that mappings for the elements of a SAML assertion are supported. For instance, the following elements need mapping:

- The user identifier needs to be mapped to the UUID within the TCPA URM platform. This mapping is handled by the identity provider of the TCPA URM platform, which offers a user-identifier resolution service to the PII manager.
- Scopes of authorization need to be translated to the scopes that actually are enforced by the TCPA URM platform. There is no possible comprehensive list for these scopes, as the OAuth-based protocols can extend the standard scope model.

In terms of mapping the content of the SAML assertion to the consent model, the following elements need to be considered:

- User identifier maps to the `NameID`. The NameID format must be one of UUID or email address.
- Start timestamp maps to `NotBefore`.
- Expiry timestamp maps to `NotOnOrAfter`.
- Names, formats and values of standard attributes also need mapping. This mapping depends on the type and format of attribute and is not covered in this document.
- Our consent model also supports extended, admin-definable, attributes that can be mapped in a similar fashion, depending on their respective types and formats.

4.8.6 Considerations Regarding Token Exchange

As specified in RFC 8693 [45], plain OAuth access tokens can also be exchanged for delegation or impersonation purposes. The delegation and impersonation[5] features require that the PII manager be able to perform an additional round of redirection, that enables the retrieval of a security token.

In particular, it requires the ability to:

[5]Impersonation is not used in a negative way in RFC 8693 [45]. It means that the target service does not need to be aware of the delegation that happens between a subject entity and an actor entity.

- Receive an access token and to choose the adequate backend.
- Retrieve the newly-issued security token that bears the correct actor and subject fields.
- Submit the token for consumption to the proper backend.

This RFC would require that the TCPA deploy a Security Token Server [45, Chapter 2], delivering tokens that allow delegation scenarios where the PII manager request sources for PII on behalf of the TCPA URM platform(s). However, the Token Exchange IETF Request for Comments (RFC) is quite recent and the use of such protocol in the industry is not widespread yet.

4.9 The PII Management User Interface (PMUI)

4.9.1 Overview

The PMUI's purpose is to help users decide which TCPA URM platforms should be authorized to collect their PII, including configuring the ability of the PII manager to manage PII even when the user is offline.

User-Managed Access [56, Section 3.3.4] proposes a procedure to ensure these offline properties. The procedure relies on the (OAuth) *scopes* on resources requested by the URM clients.

Additionally, the PMUI reflects the ability of the PII Manager to abstract PII location. As a reminder, we note that the PII abstraction property requires that the PII manager acts first (i) as a requesting party for the registered PII sources and then conversely (ii) as a resource server for the TCPA URM platform. The requesting TCPA URM platform only deals with the PII manager and does not need to know which sources are part of the PII collection process.

The PMUI also allows the user to visualize which PII transfers have happened and with which service providers. When granting PII access to the SPs, the PII manager provides logs' information to the associated users and to the data owners. These logs can be visualized at the convenience of the user, *i.e.*, whenever is suitable for the user. Information obtained at client-registration time is also presented to the user, such as the category of service providers and the purpose of collection.

Eventually, the user must be able to revoke a previously granted access. The PII management user interface thus includes management pages for each previously created access rule.

4.9.2 User-Definable Parameters

We rely on an access-policy definition at resource registration time, enabling the user to optionally define the parameters below:

- The required scopes for the resource. These scopes describe usual operations such as reading, deleting, modifying a resource, accessing a sub-resource. Alternatively, they can also be specific third-party application scopes.
- The time-window of access authorization. When issuing access tokens within the URM system, the PII manager uses these user-defined parameters to adjust the validity time-window of the token.
- The service or the category of service for this authorization rule. The services accessing to the user's PII are sorted according to user-defined categories. Any authorization rule defined by the user is applicable to a category of services only.

When user data are provided by sources acting as SAML or OIDC providers or as OAuth resource servers, this information may already be provided along with the PII payload. Yet it may be overridden by the user, for instance the user can further restrict the time validity of the PII. For plain REST sources however, this information needs to be provided at registration time.

As a result, the user interface adopts this approach, letting the user define the aforementioned parameters when the metadata provided by the source does not provide this information.

The constraint of uniqueness regarding the resource, the scope of action, the time-window and the category of services altogether is enforced. At a particular moment in time, for a given resource, a category of service and a scope of action, at most one authorization rule can apply. The access control system denies all by default.

4.10 Functional Analysis of the Proposed Architecture Including the PII Manager

This section provides an informal analysis of the compliance of the PII manager with the targeted functional requirements described in Section 2.3 and then it provides a description of the functional requirement one after the other. These are the PII architecture components mapped with the requirements.

1. **The PII management capabilities** map to requirement "*usage definition*" (requirement #1).
2. **The PQI and source backend of the PII manager** map to requirements "*consent monitoring*" and "*usage monitoring*" (#2 and #3).
3. **The delegation capabilities** map to requirement "*delegation capabilities*" (#4).
4. **The PQI and the source backend**, again, map to requirement "*PII location abstraction*" (#5).
5. **The unified authorization scheme** maps to requirements "*protocol standardization*" and "*access uniformization*" (#6 and #7).
6. **The support of several types of sources** maps to requirement "*authorization protocol interoperability*" (#8).

4.10.1 Usage Definition *(requirement #1)*

The consent receipt model adopted in our contribution covers usage definition. Indeed, the data fields and the transaction fields that are part of this model make it possible to specify the purpose of PII collection as part of user consent information.

4.10.2 Consent Management and Usage Monitoring *(requirements #2 and #3)*

Consent management is achieved thanks to the use of consent receipts and the mapping of authorization information. Usage monitoring is ensuring by PII manager, acting as a single resource server for the TCPA URM platform(s).

4.10.3 Delegation Capabilities *(requirement #4)*

Section 4.9.2 specifies the necessary delegation capabilities that our PII manager supports in order to comply with the use case. In particular, that section provides a pseudo-algorithm for the reduction of the authorization scopes set, compatible with the UMA delegated authorization process.

4.10.4 PII Location Abstraction *(requirement #5)*

The translation of authorization information defined in Section 4.6 enables the PII manager to provide the TCPA URM platforms with the user's PII regardless of the PII actual location.

4.10.5 Protocol Standardization and Access Uniformization *(requirements #6 and #7)*

The access control rules describe whether the user authorization information for accessing PII can be granted to a URM client, either directly or through inference based on contextual information.

The direct grant is performed through the definition of preferences by the user. Moreover, these preferences are directly linked to a service provider's client.

This model helps ensuring the minimization of PII transfers: the PII manager, when deciding whether to authorize PII access to a given URM client regarding several categories of PII, can quickly verify if one of the categories of PII isn't accessible by that URM client.

4.10.6 Authorization Protocol Interoperability *(requirement #8)*

As described in Section 4.4, respecting our use case involves a strong correlation between the PII management entity and the TCPA URM platform. We now discuss the (a) interoperability property and (b) more specifically the possibility for the PII management entity to interface with other TCPA URM platforms.

In order to ensure this interoperability property, four necessary subproperties are identified: [i] interface standardization, [ii] dynamic registration (or no configuration at all), [iii] authorization protocol(s) standardization and [iv] data exchange format(s) standardization.

[i] means that the PII manager can be used for several TCPA URM platform at a time. This subproperty is ensured by offering a standard REST API. Such an API offers unambiguous data location format, standardized data operation syntax using HTTP verbs and the use of common Web technologies.

[ii] is necessary if that interoperability property is expected to be seamless, *i.e.*, with no configuration whatsoever by any human agent involved. This is achieved by OAuth 2.0 Dynamic Client Registration [81] and its associated management protocol [80]. In order for the PII manager to perform dynamic registration of the TCPA URM platforms, the following information is necessary:

- Endpoints information (support grant types, token authentication methods).
- Redirection URIs.
- Keysets locations.

From the user's point of view, when disclosing a new PII manager to their TCPA URM platform, it is sufficient to simply provide the PII manager URL, or name, for it to be discovered by the TCPA URM platform. This will be followed by the registration process.

In delegated authorization mechanisms such as the UMA grant for the OAuth 2.0 authorization protocol, a URM tool acting as a Requesting Party needs to identify itself (with a prior registration on the Authorization Server) before obtaining the requested authorization data.

[iii] is provided when the PII manager acts as an OAuth 2.0 Resource Server. This PII manager is therefore able to verify the validity of an access token for a given Requesting Party. This validation is performed according to the authorization server's token introspection endpoint [77].

[iv] implies that the PII exchanged is presented in a way that is recognized by both the sender and the receiver. This standardization is enforced by the common use of OAuth-based protocols and the assertion framework that makes it possible to interface with other federated-identity management protocols such as SAML.

4.11 Conclusion

Our PII manager, presented into practical implementation level of details, provides an abstraction solving issues due to multiple sources being considered. These issues include variety of protocols being implemented by the sources, and the resulting variations in the authorization information and in the user consent enforcement. Our approach relies on three main specified components allowing the support of functional requirements identified in our use case, including the support of several sources. Section 6.3 discusses such implementations considerations and provides a proof of concept for the PII manager.

The PII Query Interface (PQI) specifies the way the PII manager interacts with TCPA URM platforms, possibly involving an identifier mapping service.
The Source Backend (SB) specifies the interface with sources obeying to different authorization and PII retrieval protocols. Operating a SB requires a unified consent model, involving an authorization information translation across protocols. This consent model unification step, as performed by the SB, also needs a reverse translation step when the authorization scopes need reduction.
The PII Management User Interface (PMUI) specifies the user-definable parameters that take part in the support of multiple sources and the enforcement of user consent on the PII retrieved across these sources.

Our functional analysis of the architecture demonstrates that the requirements identified in the use case have been correctly addressed.

Adopting such an architecture may have a cost. First, a production-ready implementation of the PII manager is a significant task. Then, sources may vary from the theoretical specifications of the supported protocols. Finally, wide-scale architecture adoption by the TCPA may be a long process.

The next chapter tackles the possible identity matching issues when dealing with several PII sources. This issue is of primary importance as it can lead malicious users to collude and succeed in performing applicative privilege escalation.

Chapter 5

Performing Identity Matching when Interfacing with PII Sources in a TCPA Environment

In order to smoothly counteract users overriding their own privileges [97, 10] derived from their Personally Identifiable Information (PII) in Federated-identity architectures within TCPA platforms, it is now commonly assumed that their declared PII is cross-checked among several sources.

Ensuring that the PII collected from several sources as part of a user's URM request do actually belong to that user–*i.e.*, do match that user–is already performed by TCPA agents. For instance, when the user provides scanned copies of their identity documents, TCPA agents cross check that these several documents bear the same identity.

This identity matching process is however not always well formalized. Formalizing such a process within TCPA URM online services, with digital PII, is the aim of this chapter.

In our use case, and following the contributions of Chapter 4, this formalized identity matching process is carried on by the PII manager: As explained in Section 4.7.5, the PII manager collects information across several PII sources and deals with several TCPA URM platforms.

The reliability of the identity-matching process relies on the quality and the quantity of identity attributes that the sources provide to the TCPA platform. The level of trust that the TCPA platform has on each source also impacts the identity-matching process.

From a functional point of view, a first disambiguation between *validated* PII and *certified* PII can be achieved. Some PII sources are considered as trustworthy, and the PII they provide is therefore considered as *validated*. This validated PII can also be signed by the source, thus adding a property of *certification*. Thus, for qualifying the reliability of a PII, there is a need to distinguish, from an *organizational* point of view, the level of trust that each source is granted, and, from a *technical* point of view, the level of data quality a source is able to provide under a lighter validation or a stronger certification procedure (see for instance the use of such procedure in the Internet public key infrastructure–PKI [98, 42]). Now, from a technical point of view, *Certified* identity information (relying on the use of *certificates*, *i.e.*, an authority-approved public key) can take the form of assertions in the Security Assertion Markup Language [68] (SAML) which are still used in federated-identity architectures. *Validated* identity information are increasingly expanding through service providers using requesting data sources over HTTPS (with server-side authentication only), and the resulting identity information contained in the provider's

applicative response remaining unsigned. In the same vein, information can be either *validated* or *certified* as in the form of JavaScript Object Notation (JSON) which are provided by attributes providers which are mostly application programming interfaces (APIs). As a result, the aforementioned sources mostly provide *validated* identity information instead of *certified* information.

This chapter, which *extends our main use case (Section 2.2)*, presents the necessary measures when performing identity matching in distributed identity architectures. This use case comes from the domain of user-relationship management (URM) within TCPA. For this purpose, this chapter introduces a series of key concepts, involved in defining the identity-matching process itself, as well as formalizing the security analysis given later on in the chapter. The security analysis proves the security suitability of the solution against four types of identified threats.

The remainder of the chapter is as follows. Section 5.1 describes the related work on identity-matching. Section 5.2 describes the PII sources relevant to our use case. Section 5.3 defines the identity matching procedure to follow when combining user data from such sources. Section 5.4 gives the aforementioned security analysis of the identity-matching procedure within the citizen-relationship management environment. Eventually, Section 5.5 gives a brief conclusion and provides some perspectives to this ongoing identity-management research.

A proof of concept of the identity-matching process is provided in Section 6.2.

5.1 Existing Identity-Matching Contributions

Federated-identity architectures and their shortcomings have been widely described in the literature. For instance, [12, Chapters 1–3] provides an analysis on their shortcomings regarding user privacy. However, no academic contributions studying the provision of PII by several sources in federated-identity architectures have been elaborated so far.

Though the management of PII sources, in a privacy-compliant way, for user-centric architecture has been studied at large, for instance in [62] and [61] and the use of PII for TCPA-based purposes has been proposed in [70] and [86], no contributions provide solutions for identity-matching issues that arise when managing such sources.

More generally, the issues linked to identity-matching within federated-identity systems involving personally identifiable-attribute sources have not been proposed yet. Indeed, federated-identity systems are often designed with the assumption that a user is known to only one IdP within the federation. For instance, users of federated-identity systems within the academic world (researchers, students) are known to one IdP of the federation: the IdP deployed by the university, institute or laboratory they belong to.

The lack of academic coverage for this particular subject is notable. This leads us to stating the main issue, by identifying first the use case and second the useful functional requirements.

The following section provides a more thorough description of the aforementioned PII sources.

5.2 The Selected Identity Sources in our Territorial Use Case

5.2.1 *FranceConnect*

FranceConnect is the official identity federation service of the French administration. The identity information it uses comes from the INSEE's[1] RNIPP[2]. It implements the OpenID Connect (OIDC) [83]

[1] *Institut National de la Statistique et des Études Économiques*, *i.e.*, the national institution for statistics and economical studies.

[2] *Registre National d'Identification des Personnes Physiques*, *i.e.*, the national register for identification of French-living individuals–see Section 2.2.

identification layer. Thus *FranceConnect* is a production deployment adopting the OIDC protocol specifications, where OIDC providers are officially registered and have to conform with one of the three authentication levels defined by the eIDAS regulation[3].

As a result, the user identification flow requires the following steps:

1. The online service provider sends an authentication request to the *FranceConnect* service.
2. The user's Web browser is redirected to the *FranceConnect* identity provider selection interface.
3. Upon selecting one of the *FranceConnect* providers, the user authenticates to that provider. The way the user authenticates varies from one provider to another (especially when such providers obey to different eIDAS authentication levels).
4. A reverse redirection back to the service provider is performed, allowing the service provider to obtain an *ID Token*, which characterizes some of the user's identity information, including a local federation identifier for that user. Metadata such as the eIDAS level, varying from 1 *i.e.*, least trusted, to 3 *i.e.*, most trusted, is also sent back to the online service provider.

5.2.2 DGFIP

As explained in Section 2.2, the DGFIP attribute source is a specific endpoint of the *API Particulier*, maintained by the DINUM. It provides various user tax information to a service provider, after registration of the service provider and the obtention of an access token.

In order to call this endpoint to the DGFIP attribute source, the service provider must register to the *API Particulier*. This registration step is necessary prior to any access to the endpoint, and has not been automated yet. This step leads to the obtention of an API key for the newly-registered service provider, necessary for any further call to the endpoint.

Once this pre-required registration step is complete, addressing requests to this endpoint implies providing user information (as query-string arguments). This information, considered to be confidential, is made of (i) the user's identification number in the national tax system and (ii) the reference number of the user's most recent yearly tax receipt.

As a result, the user information returned by the API contains the user tax reference revenue used to determine the school restaurant fees. It also contains human-readable PII, enabling a partial verification of the identity of the user.

5.2.3 CNAF

The CNAF endpoint is also part of *API Particulier*. It provides various children's allowance information regarding the user.

Similarly to the DGFIP source described in Section 5.2.2, calling this endpoint requires the service provider to register to the *API Particulier*, and to provide as query-string arguments (i) the user's allowance identification number and (ii) the user's postcode.

The user information returned by the API contains the user's family quotient value, required to determine the school restaurant fees, as well as human readable PII.

[3]See `https://www.ssi.gouv.fr/entreprise/reglementation/confiance-numerique/le-reglement-eidas/` (resource in French).

5.3 Presentation of Identity Matching Process

5.3.1 Motivations for an Identity-Matching Automated Procedure

5.3.1.1 For an Automated Procedure

Let us consider the validation of PII by a (human) agent of a collectivity. Agents can identify slight variations and compare information presented to them in different formats. In our TCPA use case, the agents, in order to perform a single identity-matching step, are displayed information from the three sources.

Manual validation of user information is a repetitive and time-consuming task for the agents, hence preventing them to perform more meaningful manner such as validating complex procedures or providing citizens with custom case-by-case assistance.

Most importantly, this manual approach does not stand anymore if the number of sources increases in a significant manner: for an information repeated under various forms across n sources, this requires $\frac{(n-1)n}{2}$ validation steps, which rapidly becomes non-viable and incompatible with a manual systematic validation by the agent. Indeed, the underlying complexity is in $O(n^2)$. For five sources, the agent needs to complete ten comparisons; this number rises to forty-five comparisons for ten sources, and so on.

As a result, the need for providing an automated identity-matching procedure is notable. The following Section 5.3.2 presents the identity matching procedure and defines a few key concepts to performing automated identity matching based on data provided by multiple sources, as described in Section 2.2. The reader can refer to the implementation done of the identity-matching process in Section 6.2.

5.3.1.2 For an Advanced Identity-Matching Procedure

Let us consider a first simple approach for which the validation is a straightforward equality testing, *i.e.*, by directly comparing the values returned by each source in order to detect potential mismatches.

A simple approach of straightforward equality testing would require to build, for a given identity attribute I, a *result* vector as follows:

$$result_j = S_j(I), \forall j \in \{1, \ldots, n\} \tag{5.1}$$

where S_j is the j-th available source, $j \in \{1, \ldots, n\}$.

For each identity attribute, the validation process would be as follows, for each I in $\mathcal{I}*$, where $\mathcal{I}*$ is the set of all available PII attributes:

- If I is provided by all the available sources, ensure that all the n elements of the result vector are identical, *i.e.*, $result_1 = \cdots = result_n$.
- More generally, if I is only provided by a subset of all the available sources, ensure that all the elements of the subset $\{n_1, \ldots, n_k\}$ that correspond to valid information given by the sources $S_{n_1}, \ldots, S_{n_k}$ providing I, are identical, *i.e.*, $result_{n_1} = \cdots = result_{n_k}$.

However, adopting this approach will raise false negatives[4], especially when slight variations in the information retrieved across the sources have been noticed. For instance, some sources will strip the accents out of identity information represented as character strings, whereas some others will not.
This approach will also raise false negatives in some cases–for instance when a *data transformation* is required before performing any comparison. Thus a string representation of an address contains a postal

[4]That is theoretically matching identities which are detected as mismatches.

code that can be compared, after extraction, to the postal code provided by other sources of information.

Thus there is a clear need for an advanced identity-matching procedure, relying on cross-checking of asserted PII among different sources. Moreover, determining the cardinality of the set of sources providing the identity attributes is the first step to *information completeness*. The use of that cardinality value as part of the identity matching decision process is presented later in this chapter–see Section 5.3.3.

5.3.2 Presentation of the Identity Matching Procedure

According to the use case defined in Section 2.2, the TCPA URM platform relies on the identity provided by the *FranceConnect* service.

The TCPA URM platform has to ensure that the identity provided by *FranceConnect* matches with the identity attributes contained in the data returned by the CNAF and the DGFIP endpoints.

The *FranceConnect* service returns a "pivot" identity, containing a set of user attributes: (i) a blankspace-delimited list of the user's first and middle names, (ii) the user's family name and (iii) a string representation of the user's birth date.
The CNAF endpoint returns: (i) the user's full postal address, containing their postcode, (ii) a string representation of the user's birth date, (iii) a string representation of the user's full name (*i.e.*, their first and last names).
The DGFIP API returns: (i) the user's current name, (ii) the user's birth name, (iii) a string representation of the user's first and middle names, (iv) a string representation of the user's birth date and (v) a string representation of the user's postal address, containing the postcode.

Using these three sources, our goal is to propose a thorough identity-matching process with simple algorithms. These simple algorithms can be decomposed in three steps:

1. Format unification;
2. Normalization;
3. Distance computation.

However, the concept necessary to present these algorithms is *information completeness*, described in the next section.

Additionally, the remaining of this chapter adopts the following terminology:

- PII *types* are the different types of information provided by the sources, *e.g.* the user's postal address, or the user's birthdate;
- PII *attributes* are the instances of information for a given PII type, *e.g* the user's postal address as provided by source A, or the user's birthdate as provided by source B.

5.3.3 Information Completeness

The following paragraphs define the different degrees of completeness of the information provided by the sources. This concept of *information completeness* is necessary to define the identity-matching process. It denotes the idea that some PII is provided in multiple occurrences, by several sources, whereas other is more sparsely available.

5.3.3.1 Formal Definitions

- **Relation *provides***

A relation from the set of sources to the set of available PII, called *provides*, is defined for a given source $S \in \mathcal{S}$, where $\mathcal{S}$ is the set of all available sources, and a given identity attribute type $t \in \mathcal{T}$, where $\mathcal{T}$ is the set of all available identity attribute types, as follows:

$$S \text{ provides } t \tag{5.2}$$

meaning that an identity attribute of type t is provided by S for any given user of the architecture. This definition implies that the set of available PII provided by a source is the same for any user of the system. This hypothesis does not hold in some corner cases. However these corner cases do not invalidate the identity-matching procedure presented in the chapter. Thus we note $\mathcal{S}_t$ the set within $\mathcal{S}$ of all sources S so that S provides t.

- **Partial Information**

Partial information is defined as the set $\mathcal{T}_p$, respecting the following property:

$$\forall t \in \mathcal{T}_p, |\mathcal{S}_t| < |\mathcal{S}| \tag{5.3}$$

where $\neg$ is the notation used for the *logical negation* operator.

5.3.3.2 Complete Information

Complete information is defined as the logical contrary of partial information, *i.e.*, it is the set $\mathcal{T}_c$ so that:

$$\forall t \in \mathcal{T}_c, |\mathcal{S}_t| = |\mathcal{S}| \tag{5.4}$$

5.3.3.3 Sufficient Partial Information

We need to categorize partial information in a finer way in order to decide of its role in identity matching. The first category is sufficient partial information, and is defined as follows.

According to our TCPA use case, PII (i) first and middle names, (ii) last names and (iii) date of birth, as defined in the end of Section 5.3.2, are the biggest possible set of common information across all the sources. However, potential mismatches on partial information, *i.e.*, information that is shared by a strict subset of all the sources and whose cardinality is at least 2 can also be detected.

Sufficient partial information is defined as the set $\mathcal{T}_s$ so that:

$$\forall t \in \mathcal{T}_s, |\mathcal{S}_t| \geq 2 \tag{5.5}$$

5.3.3.4 Insufficient Partial Information

Some information may be offered by one source only, in which case it cannot be used as input for the identity matching process.

In other terms, insufficient partial information is defined as the set $\mathcal{T}_i$ so that:

$$\forall t \in \mathcal{T}_i, |\mathcal{S}_t| \leq 1 \tag{5.6}$$

5.3.4 Validation Algorithm

5.3.4.1 Format Unification

We still consider three pieces of PII regarding the user, *i.e.* (i) first and middle names, (ii) last names and (iii) date of birth. The comparison as well as the detection mechanisms are based on an ASCII representation of identity information (i), (ii) and (iii). Presenting the information in a similar format

is the first step to performing identity matching: for any given PII type, defining a format-unification procedure is necessary.

As explained in the problem statement (Section 2.2), depending on the sources, the identity information is presented under different formats. That's why a format-unification step is required before performing the comparison between the information available across the sources. For instance, identity information (iii)–*i.e.*, the date of birth of the user–string representation differs between the *FranceConnect* PII source (`YYYY-MM-DD`), the DGFIP data source (`DD/MM/YYYY`) and the CNAF PII source (`DDMMYYYY`).

The DGFIP PII source also provides the user's postcode of main residence, as part of a longer string representing the postal address of the user (*34 Rue des Lilas* ***75001*** *Paris*). Obviously this postcode needs to be extracted if needed for comparison to similar information provided by the CNAF PII source.

The format unification is of course specific to a PII type $t \in \mathcal{T}$ and to a given source Sj, $j \in \{1, \ldots, n\}$. Thus it is defined as a function P_t so that:

$$\begin{aligned} &\forall j \in \{1, \ldots, n\}, \exists P_t : \\ &P_t(S_j(t)) \in \mathcal{C} \end{aligned} \tag{5.7}$$

where $\mathcal{C}$ is the set of comparable elements for PII of type I, meaning that it is a set of elements of a same format along with a comparison operator.

5.3.4.2 Normalization of Unicode Strings

Let us consider the case of a family name that contains non-strictly-Latin characters, *e.g.*, Smïçz, has to undergo this format-unification step. This family name can be present in other sources under the form of a stricter Latin character set, such as Smïcz, or even Smicz.

According to the Unicode specifications [91], a normalization form involving a compatibility decomposition is most appropriate. The *compatibility* decomposition form (NFKD[5]) is favored over the *canonical* form (NFD[6]) so as to handle a subset of Unicode known to be stable. Indeed, the canonical form's ability to preserve the visual appearance of the input characters after normalization is not of interest in our use case, as the normalized information will not be displayed to end users or human agents of the platform but rather be processed against identity matching algorithm instead. Eventually, the relevant normalization process here only requires decomposition, hence the two other existing normalization forms–NFC and NFKC forms both requiring an additional composition step–are not relevant here.

5.3.4.3 Distance Computation

Even over a stable Latin alphabet, small variations appear on names provided by different sources. For instance, cases where the different representations of a user's last name across multiple sources differ only slightly, *e.g.*, Smicz for a source S_1 and Smics for a source S_2, need to be detected.

As a result, a distance computation procedure is described in this section. This procedure can be used in order to detect the aforementioned small variations in PII across sources. This procedure is based on the Levenshtein distance algorithm [50], computing a value between two strings A and B depending on the minimal number of elementary edit operations in order to go from string A to string B.

This distance defines three types of elementary edit operations on a string of characters within a character set Σ: (i) inserting a character, (ii) removing a character and (iii) swapping a character for another one in Σ.

Accordingly, a path $\mathcal{P}(A, B)$ from two strings A and B is a series of elementary edit operations changing string A into string B. A and B belong each to $\Sigma*$, which is the set of all possible strings made

[5]Normalization form by compatibility decomposition.
[6]Normalization form by canonical decomposition.

from the character set Σ along with the empty string λ. The length of this path is the number of edit operation it describes. It is noted $|\mathcal{P}|$.

As a result, the Levenshtein distance $d(A, B)$ between two strings A and B is the length of the shortest path going from string A to string B. The distance operator over $(\Sigma*)^2 \to \mathbb{N}$ is commutative, *i.e.*, $d(A, B) = d(B, A)$.

Further academical work regarding the Levenshtein distance has been published. However, in our specific case, the plain Levenshtein distance algorithm is sufficient. For instance, computing the Levenshtein distance between *Smicz* and *Smics* is straightforward: this distance is 1. Cases where the distance is relatively low are considered. Such cases help to detect variations between PII values across several sources.

The notion of *distance* and its application to our result vectors need to be explained. Each element of a result vector is firstly normalized (Section 5.3.4.2), and the global distance of a result vector is the following matrix representation:
$M(result_i) \in \mathcal{M}_{mm}(\mathbb{R}^+)$ where $i \in \{1, \ldots, n\}$ is the source index.

M is made of elements $m_{ij}, i \in \{1, \ldots, n\}, j \in \{1, \ldots, n\}$ so that each m_{ij} is either the Levenshtein distance $d(result_i, result_j)$ or left empty if $result_i$ or $result_j$ is unavailable (in case of partial information as defined in Section 5.3.3).

5.3.5 Use Case's Specific Information

A per-information validation procedure for the selected types of PII is described here. In order to efficiently detect these mismatches, the most complete information is described first, followed by increasingly more partial sufficient information.

5.3.5.1 Birth Date

The user's birth date is considered as complete for our three sources.

For each source, it is provided in a specific format:

1. *FranceConnect* adopts the OIDC ISO 8601:2004 `YYYY-MM-DD` format [83, Section 5.1].
2. The DGFIP API returns dates according to the `DD/MM/YYYY` format.
3. The CNAF adopts yet another format, returning dates according to the `DDMMYYYY` format.

Sources are considered as trustworthy regarding date consistency, therefore no semantic date validation is performed.

5.3.5.2 String Types

For each PII among our selected information (either complete or sufficient-partial), a result vector whose associated distance matrix is obtained, as explained in Section 5.3.4.3.

5.3.5.2.1 First name(s):

FranceConnect delivers a list of first and middle names. The first name is used as complete information, and the remaining middle names are sufficient partial information as they are retrieved from both the *FranceConnect* service and the DGFIP endpoint.

5.3.5.2.2 Last name:

The last name is provided by all three PII sources and is therefore complete information.

5.3.5.3 Geographical Information

The postal code from the user's main address is sufficient partial information, as it is retrieved from both the DGFIP and the CNAF sources. However, INSEE code of birth place is insufficient partial: is it returned only by the *FranceConnect* service (and only when the user is born in France).

5.3.5.4 Additional Identity Matching Solvability Parameters

Eventually, and before presenting an example of validation results (in Section 5.3.5.5, a couple of definitions of parameters that may be used by the implementers of any such identity-matching solution are given:

1. For an identity matching process involving n different sources, the *lowest degree of conflict unsolvability* for a sufficient partial information of type $t \in \mathcal{T}_s$ is the integer $x \in \{2, \ldots, n-1\}$, where $n = |\mathcal{T}|$ is the number of available sources, such as if at most x elements of the result vector differ, the validation is considered unsolvable–and therefore requires a human agent validation.
2. Similarly, the *shortest distance of conflict unsolvability* for a PII type $t \in \mathcal{T}$ is the integer y such as if two elements of the result vector for t have a relative distance of at most y with each other, the validation is also considered unsolvable without an agent validation.

These parameters should be set according to the quality of the PII information provided by the sources, the number of available sources, the number of complete and partial-sufficient attributes and the degree of partiality of the partial-sufficient attributes.

5.3.5.5 Example of Validation Results

5.3.5.5.1 Matching PII

Following the previously given example, the three sources return respectively the values *Smïçz*, *Smïcz* and *Smicz*.

Consequently, the result vector as defined in Section 5.3.4.3 is (Smïçz, Smïcz, Smicz).

After performing the NFKD-normalization of the retrieved PII, the normalized vector is (smicz, smicz, smicz).

Therefore the Levenshtein distance matrix for this PII vector is

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

This Levenshtein distance matrix obviously describes matching PII.

5.3.5.5.2 Potentially non-matching or ambiguous PII

Non-matching PII, *e.g.*, *Smïçz*, *Smïcz* and *Smics*, is considered in this paragraph.

The result vector with this PII is (Smïçz, Smïcz, Smics).

After performing the NFKD-normalization of the retrieved PII, the normalized vector is (smicz, smicz, smics).

Therefore the distance matrix for this PII vector is

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Depending on the minimum and maximum thresholds, the PII associated with this matrix will either be ambiguous or non-matching.

A visual summary of the identity matching process described in this section is visible in Figure 5.1. This figure illustrates the main steps that take part in the identity-matching process, for a given PII. For a given user, this whole identity-matching process is prone to happen as many times as there are complete or partial-sufficient information attributes available.

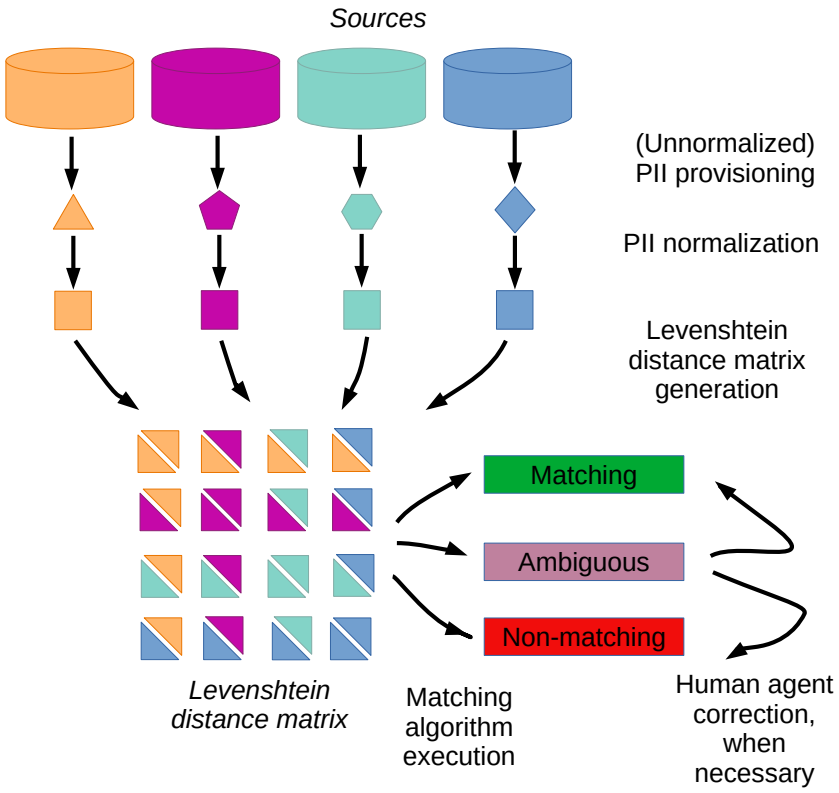


Figure 5.1: Visual summary of the identity matching process for a *complete* PII attribute

5.4 Security Analysis of the Proposed Identity-Matching Solution

5.4.1 Model and Requirements

5.4.1.1 Preliminary Definitions

We define a group of users $\mathcal{G} = \{u_1, \ldots, u_k\}, k > 2$ bringing a set of information $\mathcal{I}^*$ so that:

$$\mathcal{I}^* = \{I_{u_1,1}, \ldots, I_{u_1,v_1}, \ldots, I_{u_k,1}, \ldots, I_{u_k,v_k}\} \tag{5.8}$$

where $\{I_{u_h,j}\},\ j \in \{1, \ldots, v_h\}$ is the set of all information brought by user $u_h,\ h \in \{1, \ldots, k\}$.

Additionally, four functions *type*, *norm*, *source* and *user* are defined, respectively returning the type of PII I –as defined for our TCPA use case in Section 5.3.5–, the normalized value of PII I as defined in Section 5.3.4.2, the source, and the user from which the instance of information I is originated.

In particular, *type* is defined as follows:

$$\begin{aligned} type : \mathcal{I}^* &\longrightarrow \mathcal{T} \\ I &\longmapsto t \end{aligned} \tag{5.9}$$

where $\mathcal{T}$ is the set of all available PII types as defined in Section 5.3.5. Thus t is the actual PII type for information I.

Similarly *norm* is defined as:

$$\begin{aligned} norm : \mathcal{I}^* &\longrightarrow \Sigma^* \\ I &\longmapsto n_I \end{aligned} \tag{5.10}$$

where Σ^* is the set of all possible normalized Unicode strings along with the empty string λ. Thus n_I is the normalized value for PII I.

source is defined as follows:

$$\begin{aligned} source : \mathcal{I}^* &\longrightarrow \mathcal{S} \\ I &\longmapsto S \end{aligned} \tag{5.11}$$

Thus S is the source providing PII I. This means that among all available sources in $\mathcal{S}$, S is the source that provides the particular item of information $I \in \mathcal{I}^*$ –and the fact that I characterizes a given user u is not of interest for this definition.

Eventually, *user* is defined as:

$$\begin{aligned} user : \mathcal{I}^* &\longrightarrow \mathcal{G} \\ I &\longmapsto u \end{aligned} \tag{5.12}$$

Thus u is the user bringing PII I.

5.4.1.2 Attacker Model

Our attacker model considers four different types of possible attacks:

- ***Collusion Attack.***

This attack is a main concern that motivates the identity-matching procedure. In our case, user collusion means that a group of user $\mathcal{G}$ bring the set of information $\mathcal{I}^*$ (as defined in Section 5.4.1.1) such as:

$$\forall I \in \mathcal{I}^*, \exists S \in \mathcal{S}, S \text{ provides } I \tag{5.13}$$

The set of information brought by $\mathcal{G}$ can be used to impersonate a fictive user $u*$ and obtain privilege escalation on the system. In order for the analysis to stay valid, a realistic hypothesis is stated: the group of users $\mathcal{G} = \{u_1, \ldots, u_k\}$ should not be completely homonymous regarding the available sources. As informally stated in Section 2.2, this means that if

$$\begin{aligned} &\forall I, I' in \mathcal{I}*, \\ &(type(I) = type(I') \wedge \\ &source(I) = source(I') \wedge \\ &user(I) \neq user(I')) \\ &\implies norm(I) = norm(I') \end{aligned} \tag{5.14}$$

then the k colluding users are completely homonymous and the identity matching cannot hold. This case is considered extremely unlikely, and stating a hypothesis of non-complete homonymity is reasonable.

Alternatively, this hypothesis can be stated in matrix terms. If for each result vector $result$ of a given PII type $t \in \mathcal{T}$, if we have:

$$\forall i \in \{1, \ldots, n\},\ result_i = \begin{cases} empty \text{ if } \neg S_i \text{ provides } t \\ \alpha \text{ otherwise} \end{cases}$$

where α is a constant value for $result$, then the collusion cannot be prevented.

• ***Identity/Attribute Theft Attack***, on one or several sources.
This means that a user's credentials to one or several PII sources have been stolen by a rogue user.
This means that a rogue user u_r knows a subset of the credentials $\{c_{u_h,1}, \ldots, c_{u_h,n}\}$ of an honest user u_h, used for authentication to sources $S_1, \ldots, S_n$.

• ***Man-in-the-Middle Attack***, tampering data from one or several of the sources.
More formally, if $I = \{i_1, \ldots, i_n\}$ is the set of information retrieved from the remote sources $S_1, \ldots, S_n$, this means that there is a subset $K \subseteq I$ containing tampered data.

• ***Impersonation of Sources.***
This type of attack is similar to the previous one: its direct consequence is the citizen relationship management platform retrieving potentially-erroneous data from the remote sources.

5.4.1.3 Resilience and Security Requirements

Along with the attacker model, a list of security and privacy requirements are defined:

- ***Requirement 1:*** The user should be able to use the TCPA URM platform even in case any of the four aforementioned attacks is launched.
- ***Requirement 2:*** The identity matching should happen even in case of a degraded quality of the PII served by the sources.
- ***Requirement 3:*** Identity mismatches and attempted attacks should be detectable.

5.4.2 Security and Resilience Analysis

5.4.2.1 Security Analysis against the Attacker Model

The resistance of the proposed solution against the attacker model depends on a thorough identity matching across the sources. Therefore this identity-matching process is at the center of the use case, as it prevents–either directly or indirectly–all four types of attacks listed in Section 5.4.1.2:

1. A thorough identity matching process prevents user collusion: The group of colluding users $\mathcal{G} = \{u_1, \ldots, u_k\},\ k > 2$ manages to retrieve information from the complete set of sources $\mathcal{S} = \{S_1, \ldots, S_n\}$. Thus, as explained in Section 5.4.1.2, each colluding member in $\mathcal{G}$ brings some information retrieved from one or several source. The function g is defined as follows:

$$\begin{aligned} g : \mathbb{N} &\longrightarrow \mathbb{N} \\ l &\longmapsto m \end{aligned} \tag{5.15}$$

   meaning that the information retrieved from source S_l comes from a colluding user $u_m \in \mathcal{G}$.
   In this case the distance matrix is made of the elements $m_{i,j},\ i \in \{1, \ldots, n\},\ j \in \{1, \ldots, n\}$ so that:

$$m_{i,j} = \begin{cases} 0 \text{ if } g(i) = g(j) \\ d > 0 \text{ otherwise} \end{cases}$$

where d is the Levenshtein distance between the two elements $result_i$ and $result_j$ of the result vector (see Section 5.3.4.3).

Assumption 1: This statement illustrates and confirms the intuitive assumption that the collusion may be detected if the group of colluding users contains at least two individuals. We assume that these two individuals might be homonymous, but a subset of their PII provided by the remote sources must differ.

Indeed, the distance matrix $M((result_i),\ i \in \{1,\ldots,n\})$ is made of elements $m_{i,j},\ i \in \{1,\ldots,n\},\ j \in \{1,\ldots,n\}$ where $n = |S|$ is the number of available sources, we have:

$$m_{i,j} = \begin{cases} empty \text{ if } result_i \text{ or } result_j \text{ is missing} \\ 0 \text{ otherwise} \end{cases}$$

Conclusion 1: Under the assumption 1, the solution is suited to prevent user collusion: two users or more performing collusion will lead to a non-matching distance matrix.

2. Resistance to user identity theft on the citizen-relationship management platform is also provided.

   An attacker performing a successful identity theft from a given user u on a subset $\mathcal{R} \subset \mathcal{S} = \{S_1,\ldots,S_n\}$ of the remote sources acts in the following manner: For any given source $S_j,\ j \in \{1,\ldots,n\}$, if $S_j \in \mathcal{R}$ then the attacker uses the user's stolen credentials to obtain their information for that source, else the attacker obtains information that does not belong to u. In the best case scenario in which $\mathcal{R} \neq \mathcal{S}$, the attacker is able to perform identity theft on a second user u' on a subset $\mathcal{R}' \subset \mathcal{S}$ with $\mathcal{R} \cup \mathcal{R}' = \mathcal{S}$.

   Assumption 2: For convenience it is also assumed that $\mathcal{R} \cap \mathcal{R}' = \{\}$, without invalidating the current demonstration.

   As a result, the distance matrix M computed as part of the identity matching process contains the elements $m_{i,j}, i \in \{1,\ldots,n\}, j \in \{1,\ldots,n\}$ so that:

$$m_{i,j} = \begin{cases} 0 \text{ if } \{S_i, S_j\} \subset \mathcal{R} \\ 0 \text{ if } \{S_i, S_j\} \subset \mathcal{R}' \\ d > 0 \text{ otherwise} \end{cases}$$

   Note that if $\mathcal{R} = \mathcal{S}$ then the identity theft cannot be detected.

   Assumption 3: As a result, a thorough identity matching should detect an identity theft attempt for any subset of sources $\mathcal{R}$ so that $|\mathcal{R}| < |\mathcal{S}|$.

   Conclusion 2: Under the second and third assumptions, identity theft risks are drastically reduced. An attacker willing to perform identity theft would have to obtain the credentials to all n sources, with $n = 3$ in our use case: the *FranceConnect* credentials, the family allowance private identification number and the national tax system private identification information. As long as the users maintain their credentials carefully, the risk of the attacker performing identity theft on the three sources of our use case seems negligible.

3. Data tampering due to a man-in-the-middle attack is prevented.

   A man-in-the-middle attacker performing a successful data tampering on a subset $\mathcal{R} \subset \mathcal{S} = \{S_1,\ldots,S_n\}$ of the remote sources acts in the following manner: for any given user u, the PII provided by any source belonging to $\mathcal{R}$ will be modified so as to create a fraudulent identity of a (fictive or real) user $u*$. The objective of the attacker is to tamper the data provided by the sources belonging to $\mathcal{R}$ so as to create $u*$ as a consistent identity.

   As a result, the distance matrix M computed as part of the identity matching process contains the elements $m_{i,j}, i \in \{1,\ldots,n\}, j \in \{1,\ldots,n\}$ so that:

$$m_{i,j} = \begin{cases} 0 \text{ if } \{S_i, S_j\} \subset \mathcal{R} \\ d > 0 \text{ otherwise} \end{cases}$$

Indeed, in the general case, the attacker does not know what PII attributes are provided by the sources belonging to $\mathcal{R}' = \mathcal{S} \backslash \mathcal{R}$, where $\backslash$ is the set difference operator. The attacker is therefore unable to tamper PII attributes from sources in $\mathcal{R}$ in a way that would match the ones provided by sources in $\mathcal{R}'$.

Assumption 4: This type of attack cannot be prevented if $\mathcal{R} = \mathcal{S}$, that is $\mathcal{R}' = \{\}$, that is if the attacker is able to tamper data provided by any of all the available sources. In other terms, this attack can only be prevented if $\mathcal{S} \backslash \mathcal{R} \neq \{\}$.

Conclusion 3: Under the fourth assumption, man-in-the-middle attacks leading to users' identity attributes tampering can also be prevented. For instance, [48] provides a security analysis of TLS authentication as used in HTTPS for our Web-based sources. As discussed in that analysis of the TLS protocol, the potential breaches in the implementation of the TLS layer[7] are not specific to our use case. For discussions regarding these potential breaches, see for instance the security considerations of the TLS specification document (Request for Comments, RFC) [76, Section 10], edited by the Internet Engineering Task Force (IETF). The instance of an attacker being able to perform such an attack on all three sources of our use case is therefore considered as negligible.

4. As explained in Section 5.4.1.2, this security requirement also prevents a successful attack led by impersonating one or several sources. Similarly, this type of attack cannot be prevented if the attacker is able to impersonate any of all available sources.

   Conclusion 4: Eventually, the solution is also suited to prevent the impersonation of sources by the attacker. The possibility for an attacker to impersonate any of the available sources depends on the underlying applicative protocol. For that reason, the likelihood is the same the one studied in the previous bullet item, *i.e.*, the ability for an attacker to perform a man-in-the-middle attack on all three sources: in the context of our use case, the impersonation of all three sources by an attacker is considered as negligible.

5.4.2.2 Requirements Enforcement even under the Attack Model

Similarly, the resilience and security requirements identified in Section 5.4.1.3 can be validated as follows:

• ***Enforcement of requirement 1:*** The solution is proved resistant against the four types of attacks defined in Section 5.4.1.2. Additionally, the attacker model does not include direct attacks on the TCPA URM platform. The resistance against such direct attacks are not specific to the application but depend on the Web framework used, *i.e.*, Django–provided that its development guidelines for security and privacy are respected. As a result the first requirement is assured.

• ***Enforcement of requirement 2:*** With our TCPA use case involving three PII sources, the degraded quality of the information provided by these sources can be neglected. Of course, and as described in that section, a higher number of sources would increase the trust in the identity-matching process.

• ***Enforcement of requirement 3:*** Using proper PII normalization and distance matrix generation methods allow for the identification and the prevention of identity mismatches and attempted attacks.

5.5 Conclusion

This increasing number of sources is the result of the also increasing digitization of public services. The current process of human TCPA URM agents performing manual identity matching doesn't bode well with this increasing number of sources. Identity matching across multiple PII sources in a federated-identity environment has therefore become a challenging concern. These sources tend to adopt widely accepted authentication and authorization standards [83, 36]. However, these standards do not offer out-of-the-box solutions for matching the users' digital identities across multiple PII sources, and as a result identity mismatch errors happen.

[7] The most popular implementation of TLS being OpenSSL.

The more sources there are, the higher risk of a human error when the identity matching process is fully manual. Our contribution could result in an assistance tool–possibly a submodule of the PII manager (Chapter 4–in the TCPA agent's backoffice interface, when such identity matching steps are necessary. Such a tool would be adapted to the increasing evolution towards sources providing structured PII, thus deprecating the use of scanned documents in online procedures *de facto*.

Validating the contribution happens on two different planes. First, the security analysis detailed in Section 5.4 makes us confident that the solution is secure against a set of four different attacks, possibly involving several users.
The other validation is the proof-of-concept implementation. Despite not having a significant database of user PII spread over several PII sources, we can provide an implementation and validate it with a restricted set of users. Chapter 6 proposes such a proof-of-concept implementation.

Chapter 6

Proof of Concept & Implementations Considerations

6.1 Introduction

This chapter provides proof-of-concept implementations for two contributions of this manuscript.

First, Section 6.2 validates the ability to implement a simple identity-matching process as specified in Chapter 5. It proves that a concise addition (about a hundred lines of code) to the existing Publik software suite enables the TCPA agents to visualize the result of the identity matching process in the workflows after users have submitted their requests. It provides a short validation algorithm, as described in Section 5.3.4.

Second, Section 6.3 validates the ability to implement the PII manager of Chapter 4 as a utility software tool. Thus it can be included in the Publik software suite, but can also be deployed as part of other URM platforms. The main elements and points of interest of this proof are described, and implementation considerations are given.

The two software proofs are free software and links to their respective sources are given in the following sections.

6.2 Proof of Concept of the Identity Matching Process

6.2.1 Implementation Considerations

This section relies on the Publik URM software suite. Licensed as AGPLv3 (Affero General Public Licence)[1] free software, its sources are available on its project management webpage[2].

This software can either be installed as Debian packages or, for development purposes, directly from sources using an Ansible playbook. For our experimental setup, a development instance of the Publik software suite is installed, using the community documentation[3] of the software installation process.

This TCPA URM platform is made of three types of software entities:

[1] For more information about the AGPL, and its differences with its more famous sibling the General Public License (GPL), see `https://www.gnu.org/licenses/agpl-3.0.html`.

[2] `https://dev.entrouvert.org/`

[3] See `https://doc-publik.entrouvert.com/dev/installation-developpeur/` (resource in French).

- User-oriented software entities, offering URM features such as content management, appointment-making with TCPA agents, or scanned documents depository;
- Similarly, the TCPA URM platform is also made of agent- and administrator-oriented software entities, offering form and workflow design, or collect and expose statistics about the platform usage;
- Technical software entities, necessary for the unity of the TCPA URM platform.

6.2.2 Validation on the URM Platform

This section assumes that a running Publik instance is accessible with administrator privileges in order to set up the identity-matching procedure.

In order to meet our use case, a school restaurant online subscription form and its associated workflow are configured. As explained in the territorial use case–Section 2.2–, the online procedure requires the user to provide their tax and children allowance information. When such information is provided, the workflow performs an identity matching validation.

The key features used in the workflow are:

- WebService calls, in order to retrieve the identity information available at the three *FranceConnect*, DGFIP and CNAF remote sources.
- Evaluation of Django custom template filters, in order to:
  1. perform format unification.
  2. normalize the information retrieved from these sources. This normalization includes splitting the different fields for a consistent information comparison, as well as performing Normalization Form Canonical Decomposition (NFKD) over potentially non-ASCII strings. This normalization form means that the Unicode characters of the strings retrieved from the data sources are translated into a set of characters known to be stable. String types, as mentioned in Section 5.3.4.1 are normalized using the `unicodedata` python module normalization algorithm, set to the NFKD form [21].
  3. compute the distance between elements of a result vector of normalized PII from our three PII sources. One Python software implementation is the `python-levenshtein` software module[4], providing a simple API for computing the edit distance between two strings.
  4. display the identity-matching result information in a human-readable manner.

Adding template filters is performed directly according to the Django template engine.

A simple procedure that generates the result vectors based on the PII retrieved from the multiple sources needs to be implemented. The result vector is built thanks to the following steps:

1. computing the NFKD-normalization on each PII retrieved.
2. sequentially adding all the normalized elements into a (Python) list, representing the result vector.
3. generating the symmetrical distance matrix:

```
@register.filter
def ldistance_matrix(vector):
    matrix = []
    for i, el_i in enumerate(
            vector.split()):
        matrix.append([])
        for j, el_j in enumerate(
```

[4] `https://pypi.org/project/python-Levenshtein/`

```
                vector.split()):
            matrix[i].append(
                ldistance(el_i, el_j))
    return matrix
```

This matrix is of course not meant to be displayed to the user or to the human agent. However, it needs to be stored for later use as input for decision algorithms.

4. applying a threshold function

$$\begin{aligned} & f : \\ & \mathcal{M}_{mm}(\mathbb{R}^+) \longrightarrow \\ & \{\text{"Matching"}, \text{"Non-matching"}, \text{"Ambiguous"}\} \\ & M((result_i),\ i \in \{1, \ldots, n\}) \longmapsto s \end{aligned} \tag{6.1}$$

where $n = |\mathcal{S}|$ is the number of available sources and s is the matching decision.

For instance, a Python implementation of the algorithm described in Section 5.3.4.3 would be:

```
from django import template

@register.filter
def matching_result_strict(matrix):
    min_threshold = 1
    max_threshold = 3

    for i, el_i in enumerate(matrix):
        for j, el_j in enumerate(el_i):
            if el_j > max_threshold:
                return 'Non-matching'
            elif el_j > min_threshold:
                return 'Ambiguous'
    return 'Matching'
```

Figures 6.1 – 6.3 are screenshots from the TCPA backoffice interface offered to the administrator in order to build the URM request.
First, Figure 6.1 is the visual representation of the URM workflow that processes the user's request in the Publik software suite. It is a finite-state machine [39], thus defining a set of states and transitions between states. The actions executing while entering a new state can manipulate the PII sent by the user, interact with the user, compute intermediary data, and so on. Transitions between states can be automatic, *i.e.* depending on an automatically-assessed condition, or manual, *i.e.* performed by the TCPA agent.
Figure 6.2 shows a minimalist user form that has been built up by the administrator in the backoffice interface of the Publik URM platform. The user prompts the user for the information that will be input to the *API Particulier* in order to retrieve the user's fiscal information as well as children's allowance information.
Figure 6.3 shows a basic usage of the Django template language in Publik's form factory, allowing the computation of the successive identity-matching elements defined in Section 5.3.4 (the normalized PII, result vector and distance matrix).
Finally, Figure 6.4 shows a screenshot of the Django-template-computed base components for non-matching PII, while Figure 6.5 shows the screenshot for ambiguous PII.

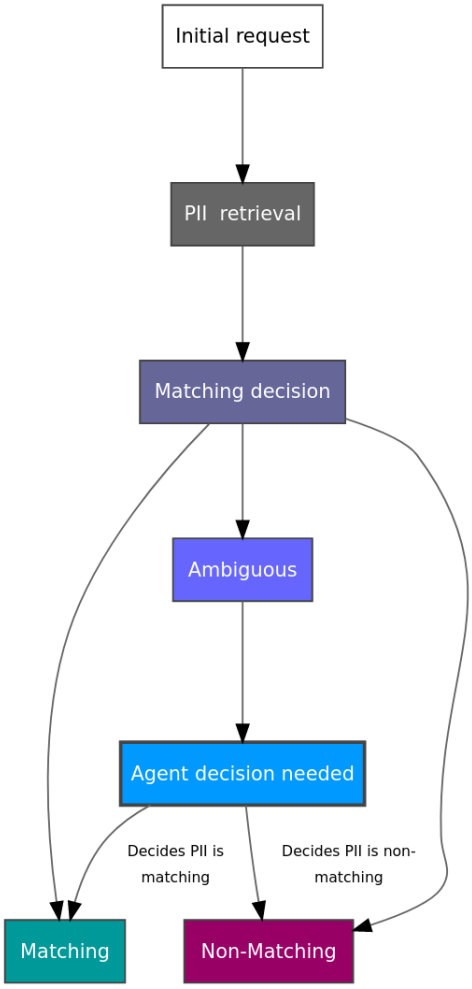


Figure 6.1: Building the identity-matching workflow in the Publik URM platform UI

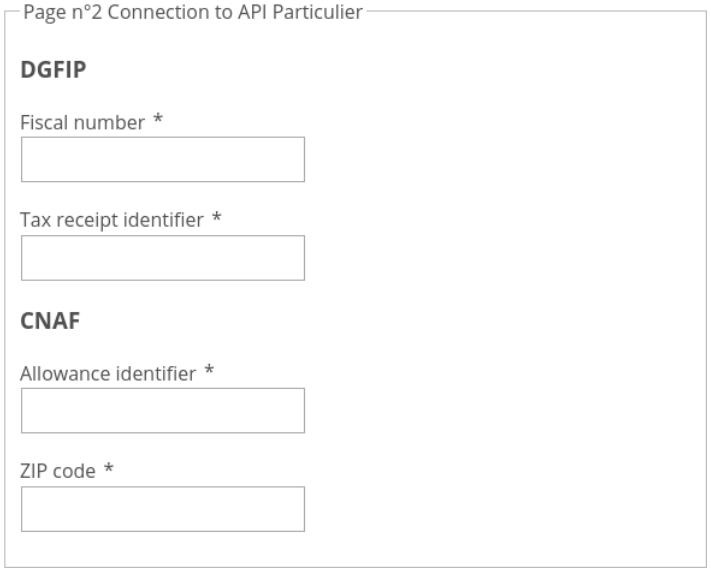


Figure 6.2: Building a corresponding minimalist user form so as to connect to *API Particulier*

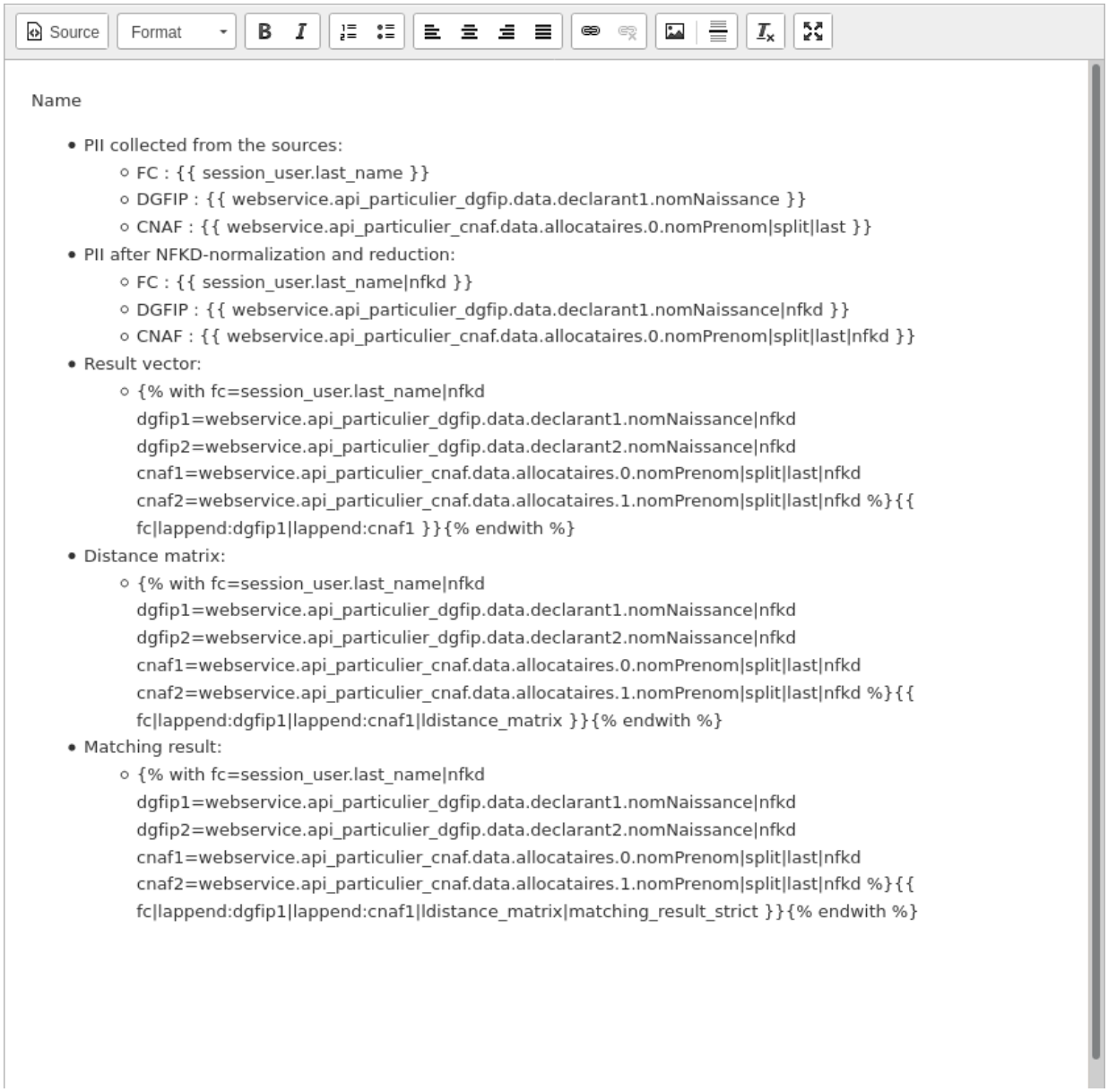


Figure 6.3: Using the Django template language to compute the identity-matching base elements of Section 5.3.4

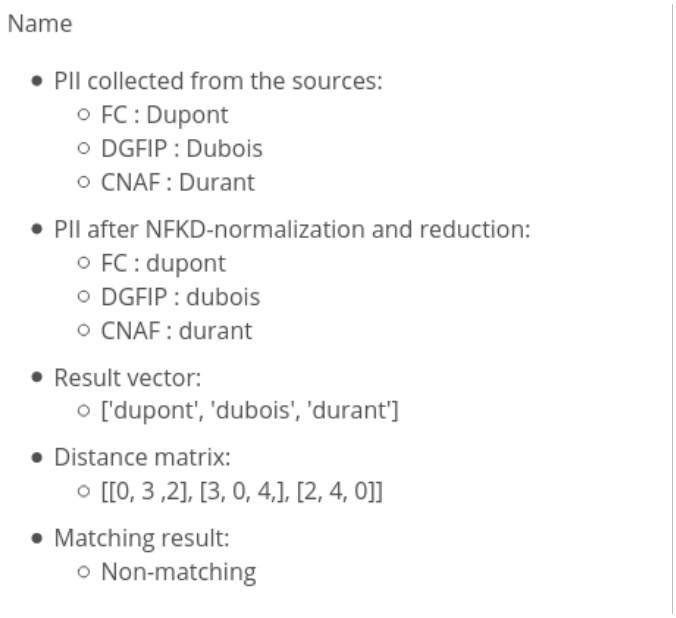


Figure 6.4: Screenshot of Django template identity-matching base elements of Section 5.3.4 for non-matching PII

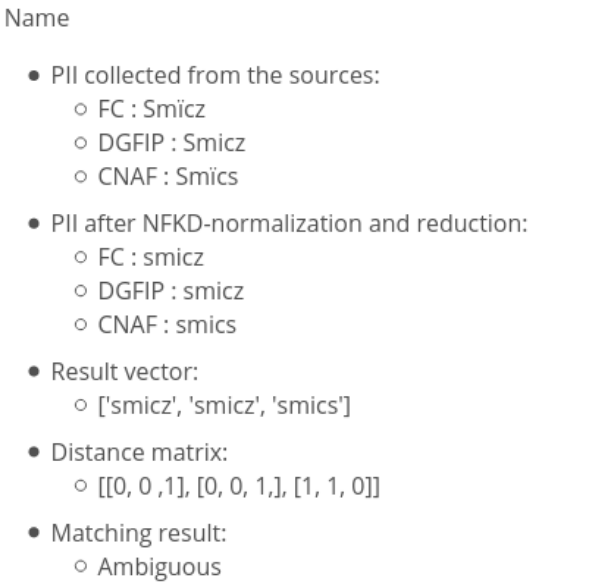


Figure 6.5: Screenshot of the Django template identity-matching base elements of Section 5.3.4 for ambiguous PII

6.3 Implementation Considerations & Proof of Concept of the PII Manager

6.3.1 Proof-of-Concept Implementation

A proof-of-concept implementation is visible at the following public git repository: `https://git.entrouvert.org/pii-manager-poc.git/`. It is licensed as AGPLv3 free software. It provides a proof of concept for the support of OAuth 2.0 sources and REST sources.

It uses the Django web framework, in its second version. The noteworthy parts of the implementation are:

1. The data model implementing, amongst other models, the consent receipt model specified in [47].
2. The view logic, performing OAuth authorization with the scope reduction logic presented in Subsection 4.9.2. The view logic implementation conceals the complexity of gathering PII from several sources from the user's point of view.
3. The source backend, implementing support for OAuth 2.0 sources and REST sources, and putting the base layout for a future support of SAML and Kerberos sources.

Of course, some parts need improvement, such as the PMUI, presented in Section 4.9, which for now only relies on the Django administration user interface ("`/admin/`") for the management of Django data models.

Amongst other features, Django's Object-Relational Mapper allows a direct definition of Consent Receipts as they appear in the specifications [47]. For instance, an excerpt of the code within the PII manager proof-of-concept implementation defining these receipts is:

```
class ConsentReceipt(models.Model):
    # [...]
    '''
    Mandatory consent receipt transaction fields
    '''
    version = models.CharField(
        max_length=31,
        blank=False,
        null=False,
        verbose_name=_('receipt version'))
    jurisdiction = models.CharField(
        max_length=255,
        blank=False,
        null=False,
        verbose_name=_('jurisdiction applying for the receipt'))
    consent_timestamp = models.DateTimeField(
        auto_now_add=True,
        blank=False,
        null=False,
        verbose_name=_('consent timestamp'))
    collection_method = models.CharField(
        max_length=127,
        blank=False,
        null=False,
        verbose_name=_('collection method'))
    receipt_id = models.CharField(
```

```
        max_length=255,
        blank=False,
        null=False,
        default=utils.new_uuid4,
        verbose_name=('uuid4 identifier of the consent receipt'))
    # [...]
```

Finally, although the PII management user interface (PMUI) implementation is incomplete in its current form, a mockup interface of the user authorization-gathering interface on the PII manager would be as depicted in Figure 6.6.

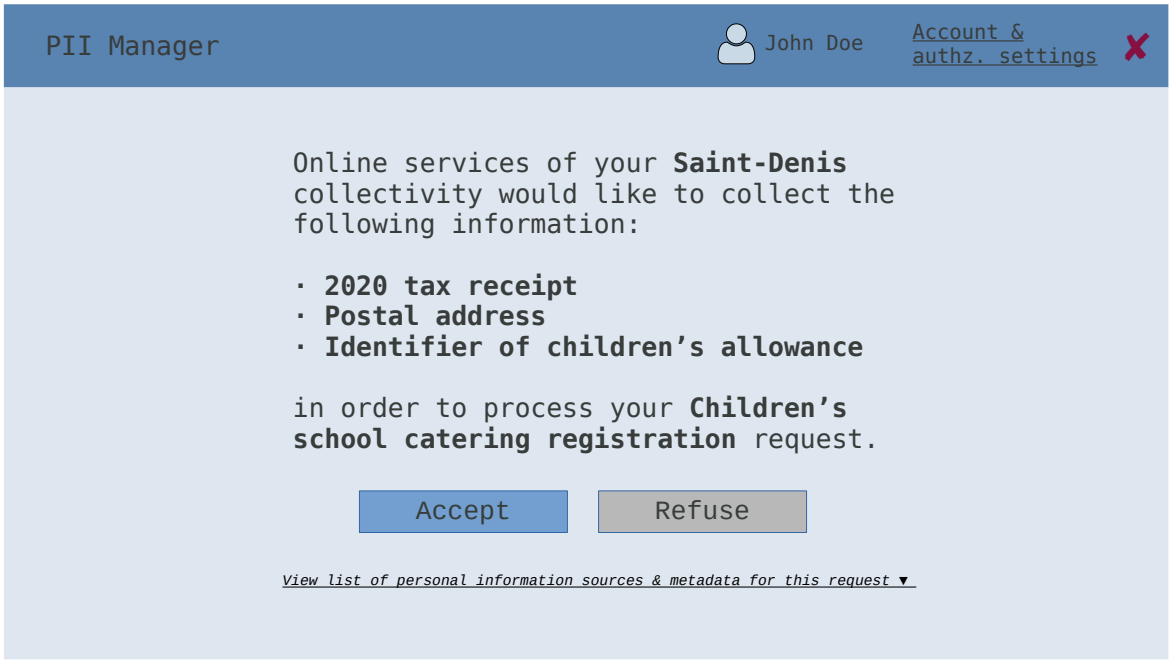


Figure 6.6: User authorization-gathering desktop interface mockup for the PII manager

Similarly, a global consent summary view mockup interface would be as depicted in Figure 6.7.

Figure 6.7: Example of user consent list mockup for the PII manager

6.3.2 Guidelines Regarding the Implementation of the PII Manager

6.3.2.1 Client Type

The (OAuth 2.0) client type depends on the ability of the client to store the secret information that was delivered by the authorization server. For instance, public clients are unable to safely keep a client secret and are therefore excluded from some authorization grant types. Most likely, the PII manager as an OAuth 2.0 client will be able to store its client secret and to operate according to any of the four main authorization grant types defined by the OAuth protocol.

6.3.2.2 Access Token Lifetime

Providing PII abstraction can result in longer PII retrieval time by the PII manager. The lifetime of access token delivered by the PII manager for usage within the TCPA URM platform should take this extra delay into consideration, at implementation level.

The choice to persist refresh tokens must be evaluated according to its privacy-usability tradeoff: persistent refresh tokens are more convenient for the PII Manager. On the contrary, for obvious reasons they make it more difficult to enforce client revocation.

More generally, adequately choosing token lifetime–as well as authorization code expiration timestamp–can help enforce privacy-compliant properties such as forward secrecy: a malicious user obtaining a token will be limited by its lifetime[5].

6.3.2.3 Introspection Endpoint & Token Validation

The choice of performing token validation by the origin authorization server must also be studied carefully while performing the implementation of the PII manager. In some cases, thorough validation

[5]That is in the case of a plain bearer token [43]. When cryptographic tokens such as JSON Web Tokens (JWT) are used, man-in-middle attacks such as the theft of a token can be prevented.

by the origin authorization server may not even be possible, in which case a partial validation by the PII manager acting as a resource server, would be the only possible option.

6.4 Conclusion

This chapter has provided proofs of concept for two contributions: the PII manager of Chapter 4 and the identity-matching procedure of Chapter 5.

These two prototypes can be seen as the first step in the production implementation of each of these contributions. Indeed, although it leaves out performance and other production considerations (packaging, distribution, continuous integration, *etc.*), it proves the feasibility of including such components in an existing URM software suite. Should such a production-ready implementation effort be made, it would most likely be a joint effort with the TCPA, so that production implementations target the real needs of the TCPA.

Chapter 7

Conclusion & Perspectives

7.1 Throwback

7.1.1 Technological Survey

After stating the problem and defining a territorial use case (Chapter 2), performing a technological survey of existing industrial and academic solutions relevant to the use case has helped us identify an optimal solution in Chapter 3. Through an evaluation of thirteen solutions according to fourteen functional criteria, this survey has paved the way for the objectives that a solution needed to meet. Amongst these fourteen criteria, five were identified as critical as they needed to be strictly enforced in order to comply with our use case.
The thirteen solutions were classified in four categories, *i.e.* personal data store (PDS), identity manager (IdM), anonymous certificates and delegation architectures. Performing the evaluation of solutions according to the criteria with a particular attention to the critical ones enabled the identification of an optimal solution for our territorial use case.

7.1.2 Consent & Sources Management

Chapter 4 has provided a solution to handle third-party PII sources. We have proposed a PII manager that would meet the functional requirements of the use case, in particular it lowers the complexity of managing several PII sources from the user's point of view. It provides consent management while supporting sources belonging to any of four types: OAuth 2.0, SAML, Kerberos and REST.
This PII manager bears four subcomponents: the PII Query Interface, the Core Consent Management module, the Source Backend and the PII Management User Interface. The discovery, registration, authorization and PII collections steps performed within the resulting architecture have therefore been specified. The way the solution can be extended in order to support additional sources types has also been proposed in this chapter.

Eventually, we note that an additional application can be identified: at the time of writing, the French government is experimenting the use of *FranceConnect* PII sources and has published technical documentation about such providers. These providers act as resource servers according to the OAuth 2.0 authorization management protocol [36] (complying with the OAuth 2.0 implicit grant type), which is one of the sources types supported by the contribution of Chapter 4

7.1.3 Identity Matching

Chapter 5 has been the opportunity to address a certain number of issues within TCPA services when the user's identity across several sources needs to be cross-checked. Identity matching on the identity conveyed by these sources must be performed in a consistent way. The base concepts in order to perform identity matching on the PII provided by three official sources in production have been presented. A security analysis of the identity-matching process has also been performed.

7.1.4 Software Validation

In Chapter 6, our software proofs of concept and implementations guidelines have proposed directions to follow for whoever would show interest in providing a production-ready implementation of the PII manager. We have also given guidelines on how to perform the direct mapping of authorization information as it turned out to be a necessary part of our contribution.

7.1.5 Ability of the Contributions as a Whole to Answer the Initial Subject

PII management capabilities to the user of the Territorial Collectivities and the Public Administration (TCPA) have been proposed by this thesis. It has also provided processes and tools for the TCPA User-Relationship Management (URM) online services. The identity-matching procedure has been included as a software contribution to the form issuing software tool within Publik (on a separate `git` branch that can be included in the main production source base at any time). Additionally, the PII manager proof-of-concept implementation has been designed as a separate software module that can be integrated in the Publik URM modular software suite.

As stated in the subject, the *Tell us once* program has been a significant concern that was part of each contribution of the thesis. Solutions for the management of various PII types including scanned documents and structured PII have also been proposed. PII lifecycle management has of course been a central concern of the thesis, as well as legal compliance (including the GDPR, recent French ministerial decrees regarding PII exchanges within TCPA, and privacy-related regulations). The study of production PII management workflows within TCPA using the Publik software suite has also helped specifying the contributions.

From a technical point of view, the challenge of leveraging protocols to address the issue of PII sparsity across several sources has also been a major concern for each contribution. More particularly, the issue of compliance with the (current and upcoming) Internet Engineering Task Force (IETF) specifications dealing with privacy and with PII management has been significant.

7.2 Limitations, New Perspectives & Upcoming Challenges

This section describes the current limitations of the contributions, either they be from a theoretical point of view (*e.g.* limitations in the design of the contributions themselves), or from a practical point of view. Additionally, new technical perspectives and functional considerations on the possible wide-scale adoption of such contributions by the TCPA are listed.

7.2.1 Current Limitations

From a practical point of view, the experimental PII management flow proposed by the *FranceConnect* PII sources may be adopted nationally at production level for official online procedures. It will shift the duty of identity-matching from the service providers to these official PII sources. One would therefore wonder whether the identity-matching solution presented in Chapter 5 remains relevant. We state that the proposed automated procedure remains relevant even once the aforementioned experimentation is brought

to production level, as it will need to be ensured by the PII sources themselves, adopting a similar procedure.
Another practical limitation is the impossibility, at the moment, to validate the identity-matching procedure of Chapter 5 against a large set of PII. The PII set supplied by the three official sources for testing purposes is currently small. We hope to see this set extended with the forthcoming introduction of the aforementioned *FranceConnect* PII sources.
These *FranceConnect* PII sources would also benefit from supporting the User-Managed Access grant for OAuth 2.0. For instance, the support of UMA would give more transparency to the user regarding the management of their PII by the TCPA URM platforms interfacing with the *FranceConnect* service.

From an ecosystemic point of view, the implementations variations of operational sources from the theoretical specifications bring complexity to the solution. In theory PII source protocols' implementation variations burden the PII manager's software validation. However, from a more practical point of view, it is more than likely that only a handful of implementations share a significant market share within TCPA. These implementations should be identified by the production-ready implementers and targeted first during the implementation process.
Additionally, a production implementation would require the thorough implementation of the mapping algorithms on authorization information, however challenging that part of the work might be. A production implementation would also require the validation with popular implementations of source clients, regardless of their degree of compliance with IETF standards.

Eventually, from a theoretical point of view, the security analysis of the identity matching process (Section 5.4) would benefit from the study of side-channel attacks[1]. So far the analysis includes four types of attackers whose sole objective remains to hinder the identity matching process for privilege escalation purposes.
On the contrary, a security analysis against privilege escalation that not necessarily involve the identity matching process would be valuable. For instance, a practical analysis against an attack taking advantage of possible inconsistencies in the TCPA URM platform's role-based access control (RBAC) configuration would be meaningful.

7.2.2 Support of the System for Cross-Domain Identity Management

The PII format defined in specifications System for Cross-domain Identity Management (SCIM) [41], although not supported yet by official PII sources in production, would be a solution to the interoperability limitations identified in 7.2.1. Indeed, SCIM proposes a standardized set of interfaces and PII formats for inter-domain PII exchanges–a point which is not covered by many other PII exchange and authorization management protocols.

7.2.3 Contribution to the User-Managed Access Work Group

Additionally, contribution to the Kantara User-Managed Access work group, and their ongoing work on consent and privacy-compliant architectures, would be a logical next step for the research described in this thesis[2].
Directions such as proof of possession for consent information is a relevant ongoing work that would enhance our contributions.

[1] See for instance
`https://www.ssi.gouv.fr/agence/publication/how-to-estimate-the-success-rate-of-higher-order-side-channel-attacks/`
(resource by the French information systems security agency).

[2] The UMA work group mailing list is freely accessible at `https://kantarainitiative.org/mailman/listinfo/wg-uma`.

7.2.4 Compliance with IETF & IRTF Work Groups

Participation in the Privacy Enhancements & Assessments Research Group (PEARG) from the Internet Engineering Task Force (IETF) and the Internet Research Task Force (IRTF) would be an appreciable step[3]. The IETF takes interest in proposing production-ready protocols that most often become *de facto* standards. The IRTF aims at reflections regarding the design of Internet-related technologies in the longer run. Privacy considerations for PII-management related technologies is one of the main objectives of the PEARG within the IETF and the IRTF. Participating in these research group on the topic on user-centric personal data management, and the emerging technologies linked to it, would be beneficial to our contribution.

7.2.5 Support of the Grant Negotiation & Authorization Protocol

New perspectives with the arrival of protocols such as the Grant Negotiation and Authorization Protocol (GNAP) [78] should be taken into considerations. The very recent draft specifications (mid-2021, three days old at the time of writing), cover a lot more than its predecessor OAuth 2.0. In particular, the way the GNAP authorization server interacts with the users is covered by this draft–whereas it was out of scope of OAuth 2.0. Additionally, the endpoints that the GNAP resource server must expose have been specified in the recent draft specification [79][4].

7.2.6 Wide-Scale Validation & Adoption

Finally, a perspective would be to experiment at larger scale the proposed solution within voluntary collectivities, *i.e.* targeting a significant number of users. URM platforms such as Publik are often deployed in medium-to-big size cities, metropolises and departments, and are used by a significant number of citizens at a national level. This will help getting valuable realistic feedback from the field.

We are confident that user-centric PII management solutions such as the ones proposed in this thesis would help mitigate attacks against TCPA users' privacy. User privacy and security must also be thought as a process. They emphasize the necessity to design solutions from the users' point of view. This is this point of view that we have adopted in the thesis. In the current context where a worrying series of ransomware attacks against French collectivities and public entities[5]have recently happened, endangering TCPA users' privacy, adopting this point of view is a necessity. Indeed, providing TCPA users with more governance over their PII mitigates the level of danger of such attacks.
Finally, these solutions should maintain the trust between users and TCPA agents. This relation of trust, either it be through online services or directly at TCPA offices, is a significant part of user-relationship management.

[3]See `https://datatracker.ietf.org/rg/pearg/about/`.

[4]The OAuth 2.0 resource server endpoints were indeed out-of-scope of the specification and purely implementation dependent.

[5]See for instance
`https://www.france24.com/en/europe/20210216-cyber-attacks-hit-two-french-hospitals-in-one-week` (France24) and `https://www.ouest-france.fr/pays-de-la-loire/angers-49000/angers-ce-que-l-on-sait-de-la-cyberattaque-qui-frappe-les-sites-de-la-mairie-7120668` (Ouest-France, *resource in French*).

Bibliography

[1] Ali Alkhalifah and Sulaiman Al Amro. "Understanding the Effect of Privacy Concerns on User Adoption of Identity Management Systems." In: *Journal of Computers* 12.2 (2017), pp. 174 –182.

[2] Liberty Alliance. "Liberty Alliance ID-WSF 2.0 Specifications including Errata v1. 0 Updates." In: *Available at http://www. projectliberty. org/resource_center/specifi cations/liberty_alliance_id_wsf_2_0_specifications_i ncluding_errata_v1_0_updates* (2006).

[3] Yousef Amar, Hamed Haddadi, and Richard Mortier. "Route-based authorization and discovery for personal data." In: 12th EuroSys Conference. Belgrade, Apr. 2017. URL: https://eurodw17.kaust.edu.sa/abstracts/eurodw17-final3.pdf.

[4] Andersdotter et al. "Evaluating Websites and Their Adherence to Data Protection Principles: Tools and Experiences." In: *Privacy and Identity Management. Facing up to Next Steps: 11th IFIP WG 9.2, 9.5, 9.6/11.7, 11.4, 11.6/SIG 9.2.2 International Summer School, Karlstad, Sweden, 8 21-26, 2016, Revised Selected Papers*. Cham: Springer International Publishing, 2016, pp. 39–51. ISBN: 978-3-319-55783-0. DOI: 10.1007/978-3-319-55783-0_4. URL: https://doi.org/10.1007/978-3-319-55783-0_4.

[5] Claudio A. Ardagna et al. "Trust Management." In: *Security, Privacy, and Trust in Modern Data Management*. Ed. by Milan Petković and Willem Jonker. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 103–117. ISBN: 978-3-540-69861-6. DOI: 10.1007/978-3-540-69861-6_8. URL: https://doi.org/10.1007/978-3-540-69861-6_8.

[6] Mikaël Ates et al. *An Identity-Centric Internet: Identity in the Cloud, Identity as a Service and Other Delights*. Tech. rep. IEEE, Aug. 2011, pp. 555–560. URL: https://ieeexplore.ieee.org/document/6045976.

[7] Arnar Birgisson et al. "Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud." In: *Network and Distributed System Security Symposium*. 2014.

[8] Matt Blaze and Martin Strauss. *Atomic Proxy Cryptography*. Tech. rep. Proc. EuroCrypt '97, 1998.

[9] Manuel Blum, Paul Feldman, and Silvio Micali. "Non-interactive Zero-knowledge and Its Applications." In: *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*. STOC '88. Chicago, Illinois, USA: ACM, 1988, pp. 103–112. ISBN: 0-89791-264-0. DOI: 10.1145/62212.62222. URL: http://doi.acm.org/10.1145/62212.62222.

[10] Sven Bugiel et al. "Towards Taming Privilege-Escalation Attacks on Android." In: *NDSS*. Vol. 17. Citeseer. 2012, p. 19.

[11] Carole Cadwalladr and Emma Graham-Harrison. "50 million Facebook profiles harvested for Cambridge Analytica in major data breach." In: *The Guardian* (Mar. 2018). Last checked: June 2021. URL: https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election.

[12] Jan Camenisch and Birgit Pfitzmann. "Federated Identity Management." In: *Security, Privacy, and Trust in Modern Data Management*. Ed. by Milan Petković and Willem Jonker. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 213–238. ISBN: 978-3-540-69861-6. DOI: 10.1007/978-3-540-69861-6_15.

[13] Brian Campbell, Chuck Mortimore, and Michael Jones. *Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants*. Tech. rep. 7522. Internet Engineering Task Force, May 2015. 15 pp. DOI: 10.17487/RFC7522. URL: https://rfc-editor.org/rfc/rfc7522.txt.

[14] Brian Campbell et al. *Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants*. Tech. rep. 7521. Internet Engineering Task Force, May 2015. 20 pp. DOI: 10.17487/RFC7521. URL: https://rfc-editor.org/rfc/rfc7521.txt.

[15] Scott Cantor et al. *Metadata for the OASIS security assertion markup language (SAML) V2.0*. Tech. rep. OASIS, 2005. URL: https://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf.

[16] Andra Ceccanti et al. "The INDIGO-Datacloud Authentication and Authorization Infrastructure." In: *Journal of Physics: Conference Series* 898.10 (2017), p. 102016.

[17] David W. Chadwick. "Federated Identity Management." In: *Foundations of Security Analysis and Design V: FOSAD 2007/2008/2009 Tutorial Lectures*. Ed. by Alessandro Aldini, Gilles Barthe, and Roberto Gorrieri. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 96–120. ISBN: 978-3-642-03829-7. DOI: 10.1007/978-3-642-03829-7_3. URL: https://doi.org/10.1007/978-3-642-03829-7_3.

[18] David Chaum, Jan-Hendrik Evertse, and Jeroen van de Graaf. "An Improved Protocol for Demonstrating Possession of Discrete Logarithms and Some Generalizations." In: *Advances in Cryptology — EUROCRYPT' 87*. Ed. by David Chaum and Wyn L. Price. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 127–141. ISBN: 978-3-540-39118-0.

[19] Council of Europe. *Convention for the Protection of Individuals with regard to Automatic Processing of Personal Data*. 1985. URL: https://www.coe.int/en/web/conventions/full-list/-/conventions/rms/0900001680078b37.

[20] Stacy Cowley and Tara Siegel Bernard. "As Equifax Amassed Ever More Data, Safety Was a Sales Pitch." In: *The New York Times* (Sept. 2017). Last checked: June 2021. URL: https://www.nytimes.com/2017/09/23/business/equifax-data-breach.html.

[21] Mark Davis and Martin Dürst. *Unicode normalization forms*. 2001.

[22] DGPN/DCPJ/PTS. *Éléments de connaissance sur la fraude aux documents et à l'identité en 2014, (Fichier PAFISA)*. 2015. URL: http://inhesj.fr/sites/default/files/fichiers_site/ondrp_ra-2015/fraude_documents_cr.pdf.

[23] Whitfield Diffie and Martin Hellman. "New Directions in Cryptography." In: *Information Theory, IEEE Transactions on* 22 (Dec. 1976), pp. 644 –654. DOI: 10.1109/TIT.1976.1055638.

[24] Entr'ouvert. *Fargo document manager presentation page*. https://dev.entrouvert.org/projects/fargo. Last checked: May 2021 [online].

[25] Entr'ouvert. *The Authentic 2 identity manager presentation page*. https://dev.entrouvert.org/projects/authentic. Last checked: May 2021 [online].

[26] European Parliament. *Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data*. 1995.

[27] European Parliament. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 4 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*. 2016.

[28] Gonzalo Fernandez et al. *OpenID Connect Client Initiated Backchannel Authentication Flow - Core 1.0 draft-03*. Tech. rep. Last checked: June 2021. OpenID, Jan. 2020.

[29] David F. Ferraiolo et al. "Proposed NIST Standard for Role-based Access Control." In: *ACM Trans. Inf. Syst. Secur.* 4.3 (Aug. 2001), pp. 224–274. ISSN: 1094-9224. DOI: 10.1145/501978.501980. URL: http://doi.acm.org/10.1145/501978.501980.

[30] Roy Thomas Fielding. "REST: Architectural Styles and the Design of Network-based Software Architectures." Doctoral dissertation. University of California, Irvine, 2000. URL: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm.

[31] ForgeRock. *OpenIDM (documentation)*. Tech. rep. Last checked: May 2021 [online]. ForgeRock. URL: https://backstage.forgerock.com/docs/openidm.

[32] French National Assembly and Senate. *Arrêté autorisant la mise en œuvre par les collectivités territoriales, les établissements publics de coopération intercommunale, les syndicats mixtes, les établissements publics locaux qui leur sont rattachés ainsi que les groupements d'intérêt public et les sociétés publiques locales dont ils sont membres de traitements automatisés de données à caractère personnel ayant pour objet la mise à disposition des usagers d'un ou de plusieurs téléservices de l'administration électronique*. 2013. URL: https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000027697207&categorieLien=id.

[33] French National Assembly and Senate. *Loi n° 78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux libertés, Version consolidée au 07 décembre 2017*. 2017. URL: https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000000886460.

[34] Paul A. Grassi et al. *Attribute Metadata: A Proposed Schema for Evaluating Federated Attributes*. Jan. 2018.

[35] Hamed Haddadi et al. "Personal Data: Thinking Inside the Box." In: *CoRR* abs/1501.04737 (2015). URL: http://arxiv.org/abs/1501.04737.

[36] Dick Hardt. *The OAuth 2.0 Authorization Framework*. Tech. rep. 6749. Internet Engineering Task Force, Oct. 2012. 76 pp. DOI: 10.17487/RFC6749. URL: https://rfc-editor.org/rfc/rfc6749.txt.

[37] Jane Henriksen-Bulmer and Sheridan Jeary. "Re-identification attacks—A systematic literature review." In: *International Journal of Information Management* 36.6, Part B (2016), pp. 1184 –1192. ISSN: 0268-4012. DOI: https://doi.org/10.1016/j.ijinfomgt.2016.08.002. URL: http://www.sciencedirect.com/science/article/pii/S0268401215301262.

[38] Mary Hodder et al. *Minimum Viable Consent Receipt (MVCR) Specification v.05*. Tech. rep. Last checked: May 2021 [online]. Kantara Initiative, 2014. URL: https://kantarainitiative.org/confluence/display/archive/Minimum+Viable+Consent+Receipt+%28MVCR%29+Specification+v.05.

[39] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. "Introduction to Automata Theory, Languages, and Computation, 2nd Edition." In: *SIGACT News* 32.1 (Mar. 2001), 60–65. ISSN: 0163-5700. DOI: 10.1145/568438.568455.

[40] Vincent C Hu et al. "Guide to attribute based access control (abac) definition and considerations (draft)." In: *NIST special publication* 800.162 (2013).

[41] Phil Hunt et al. *System for Cross-domain Identity Management: Core Schema*. Tech. rep. 7643. Internet Engineering Task Force, Sept. 2015. 104 pp. DOI: 10.17487/RFC7643. URL: https://rfc-editor.org/rfc/rfc7643.txt.

[42] Ray Hunt. "PKI and digital certification infrastructure." In: *Proceedings. Ninth IEEE International Conference on Networks, ICON 2001.* 2001, pp. 234–239. DOI: 10.1109/ICON.2001.962346.

[43] Michael Jones and Dick Hardt. *The OAuth 2.0 Authorization Framework: Bearer Token Usage*. Tech. rep. 6750. Internet Engineering Task Force, Oct. 2012. 18 pp. DOI: 10.17487/RFC6750. URL: https://rfc-editor.org/rfc/rfc6750.txt.

[44] Michael Jones, Nat Sakimura, and John Bradley. *OAuth 2.0 Authorization Server Metadata*. RFC 8414. June 2018. DOI: 10.17487/RFC8414. URL: https://rfc-editor.org/rfc/rfc8414.txt.

[45] Michael Jones et al. *OAuth 2.0 Token Exchange*. Tech. rep. 8693. Internet Engineering Task Force, Jan. 2020. 27 pp. DOI: 10.17487/RFC8693. URL: https://rfc-editor.org/rfc/rfc8693.txt.

[46] Nesrine Kaaniche, Maryline Laurent, and Sana Belguith. "Privacy enhancing technologies for solving the privacy-personalization paradox: Taxonomy and survey." In: *Journal of Network and Computer Applications (JNCA)* 171 (Dec. 2020), 102807:1–102807:85. DOI: 10.1016/j.jnca.2020.102807.

[47] Kantara Consent & Information Sharing Work Group. *Consent Receipt Specification*. Tech. rep. Last checked: May 2021 [online]. Kantara Initiative, 2018. URL: https://kantarainitiative.org/file-downloads/consent-receipt-specification-v1-1-0/.

[48] Hugo Krawczyk, Kenneth G. Paterson, and Hoeteck Wee. "On the Security of the TLS Protocol: A Systematic Analysis." In: *Advances in Cryptology – CRYPTO 2013*. Ed. by Ran Canetti and Juan A. Garay. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 429–448. ISBN: 978-3-642-40041-4.

[49] Maryline Laurent and Samia Bouzefrane. *Digital identity management*. Elsevier, Mar. 2015, p. 272. ISBN: 9781785480041. URL: https://www.elsevier.com/books/digital-identity-management/laurent/978-1-78548-004-1.

[50] Vladimir I. Levenshtein. "Binary Codes Capable of Correcting Deletions, Insertions and Reversals." In: *Soviet Physics Doklady* 10 (Feb. 1966), p. 707.

[51] Torsten Lodderstedt, Stefanie Dronia, and Marius Scurtescu. *OAuth 2.0 Token Revocation*. RFC 7009. Aug. 2013. DOI: 10.17487/RFC7009. URL: https://rfc-editor.org/rfc/rfc7009.txt.

[52] Chris M. Lonvick and Tatu Ylonen. *The Secure Shell (SSH) Protocol Architecture*. Tech. rep. 4251. Internet Engineering Task Force, Jan. 2006. 30 pp. DOI: 10.17487/RFC4251. URL: https://rfc-editor.org/rfc/rfc4251.txt.

[53] Maciej Machulak and Justin Richer. *Federated Authorization for User-Managed Access (UMA) 2.0*. Ed. by Eve Maler. Last checked: March 2019. Jan. 2018. URL: https://docs.kantarainitiative.org/uma/wg/rec-oauth-uma-federated-authz-2.0.html.

[54] Eve Maler. *Controlling Data Usage with User- Managed Access (UMA)*. 2010.

[55] Eve Maler. *UMA Release Notes*. Last checked: April 2019. Mar. 2018. URL: https://kantarainitiative.org/confluence/display/uma/UMA+Release+Notes.

[56] Eve Maler et al. *User-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization*. Internet-Draft draft-maler-oauth-umagrant-00. Work in Progress; Last checked: June 2021. Internet Engineering Task Force, Feb. 2019. 38 pp. URL: https://datatracker.ietf.org/doc/html/draft-maler-oauth-umagrant-00.

[57] Paul Marillonnet, Maryline Laurent, and Mikaël Ates. "Personal Information Self-Management: A Survey of Technologies Supporting Administrative Services." In: *Journal of Computer Science and Technology* 36.3 (2021), pp. 664–692. ISSN: 1860-4749. DOI: 10.1007/s11390-021-9673-z. URL: https://doi.org/10.1007/s11390-021-9673-z.

[58] Paul Marillonnet et al. "An Efficient User-Centric Consent Management Design for Multiservices Platforms." In: *Security and Communication Networks* 2021 (2021), p. 5512075. ISSN: 1939-0114. DOI: 10.1155/2021/5512075. URL: https://doi.org/10.1155/2021/5512075.

[59] Paul Marillonnet. et al. "An Identity-matching Process to Strengthen Trust in Federated-identity Architectures." In: *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications - Volume 3: SECRYPT,* INSTICC. SciTePress, 2020, pp. 142–154. ISBN: 978-989-758-446-6. DOI: 10.5220/0009828401420154.

[60] Massachussets Institute of Technology. *Presentation of the OpenPDS/SafeAnswers project*. 2017. URL: http://openpds.media.mit.edu/.

[61] Yves-Alexandre de Montjoye et al. "openPDS: Protecting the Privacy of Metadata through SafeAnswers." In: *PLOS ONE* 9.7 (July 2014), pp. 1–9. DOI: 10.1371/journal.pone.0098790.

[62] Richard Mortier et al. "Personal Data Management with the Databox: What's Inside the Box?" In: *Proceedings of the 2016 ACM Workshop on Cloud-Assisted Networking*. CAN '16. Irvine, California, USA: ACM, 2016, pp. 49–54. ISBN: 978-1-4503-4673-3. DOI: 10.1145/3010079.3010082. URL: http://doi.acm.org/10.1145/3010079.3010082.

[63] Hassina Nacer et al. "A distributed authentication model for composite Web services." In: *Computers & Security* 70.Supplement C (2017), pp. 144 –178. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2017.05.008. URL: http://www.sciencedirect.com/science/article/pii/S0167404817301153.

[64] Satoshi Nakamoto. *Bitcoin: A peer-to-peer electronic cash system," http://bitcoin.org/bitcoin.pdf*. Last checked: June 2021. 2008.

[65] Clifford Neuman et al. *The Kerberos Network Authentication Service (V5)*. Tech. rep. 4120. Internet Engineering Task Force, July 2005. 138 pp. DOI: 10.17487/RFC4120. URL: https://rfc-editor.org/rfc/rfc4120.txt.

[66] Chris Newman and Graham Klyne. *Date and Time on the Internet: Timestamps*. Tech. rep. 3339. Internet Engineering Task Force, July 2002. 18 pp. DOI: 10.17487/RFC3339. URL: https://rfc-editor.org/rfc/rfc3339.txt.

[67] David Nuñez and Isaac Agudo. "BlindIdM: A privacy-preserving approach for identity management as a service." In: *International Journal of Information Security* 13.2 (2014), pp. 199–215. ISSN: 1615-5270. DOI: 10.1007/s10207-014-0230-4.

[68] Organization for the Advancement of Structured Information Standards. *Security Assertion Markup Language (SAML) v2.0*. Tech. rep. OASIS, 2005. URL: http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html.

[69] Sylvia L. Osborn. "Role-Based Access Control." In: *Security, Privacy, and Trust in Modern Data Management*. Ed. by Milan Petković and Willem Jonker. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 55–70. ISBN: 978-3-540-69861-6. DOI: 10.1007/978-3-540-69861-6_5. URL: https://doi.org/10.1007/978-3-540-69861-6_5.

[70] Eliza Papadopoulou et al. "Enabling Data Subjects to Remain Data Owners." In: *Agent and Multi-Agent Systems: Technologies and Applications: 9th KES International Conference, KES-AMSTA 2015 Sorrento, Italy, 6 2015, Proceedings*. Ed. by Gordan Jezic, Robert J. Howlett, and Lakhmi C. Jain. Cham: Springer International Publishing, 2015, pp. 239–248. ISBN: 978-3-319-19728-9. DOI: 10.1007/978-3-319-19728-9_20.

[71] Christian Paquin. *U-Prove Technology Overview V1.1 (Revision 2)*. Apr. 2013. URL: https://www.microsoft.com/en-us/research/publication/u-prove-technology-overview-v1-1-revision-2/.

[72] AJ Paverd, Andrew Martin, and Ian Brown. "Modelling and automatically analysing privacy properties for honest-but-curious adversaries." In: *Tech. Rep.* (2014).

[73] Charles Rackoff and Daniel R. Simon. "Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack." In: *Advances in Cryptology — CRYPTO '91: Proceedings*. Ed. by Joan Feigenbaum. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 433–444. ISBN: 978-3-540-46766-3. DOI: 10.1007/3-540-46766-1_35. URL: https://doi.org/10.1007/3-540-46766-1_35.

[74] RedHat. *Keycloak IAM documentation*. Tech. rep. Last checked: May 2021 [online]. RedHat.

[75] Julian Reschke. *The 'Basic' HTTP Authentication Scheme*. Tech. rep. 7617. Internet Engineering Task Force, Sept. 2015. 15 pp. DOI: 10.17487/RFC7617. URL: https://rfc-editor.org/rfc/rfc7617.txt.

[76] Eric Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. Aug. 2018. DOI: 10.17487/RFC8446. URL: https://rfc-editor.org/rfc/rfc8446.txt.

[77] Justin Richer. *OAuth 2.0 Token Introspection*. Tech. rep. 7662. Internet Engineering Task Force, Oct. 2015. 17 pp. DOI: 10.17487/RFC7662. URL: https://rfc-editor.org/rfc/rfc7662.txt.

[78] Justin Richer, Aaron Parecki, and Fabien Imbault. *Grant Negotiation and Authorization Protocol*. Internet-Draft draft-ietf-gnap-core-protocol-05. Work in Progress; Last checked: June 2021. Internet Engineering Task Force, Apr. 2021. 132 pp. URL: https://datatracker.ietf.org/doc/html/draft-ietf-gnap-core-protocol-05.

[79] Justin Richer, Aaron Parecki, and Fabien Imbault. *Grant Negotiation and Authorization Protocol Resource Server Connections*. Internet-Draft draft-ietf-gnap-resource-servers-00. Work in Progress. Internet Engineering Task Force, Apr. 2021. 12 pp. URL: https://datatracker.ietf.org/doc/html/draft-ietf-gnap-resource-servers-00.

[80] Justin Richer et al. *OAuth 2.0 Dynamic Client Registration Management Protocol*. Tech. rep. 7592. Internet Engineering Task Force, July 2015. 18 pp. DOI: 10.17487/RFC7592. URL: https://rfc-editor.org/rfc/rfc7592.txt.

[81] Justin Richer et al. *OAuth 2.0 Dynamic Client Registration Protocol*. Tech. rep. 7591. Internet Engineering Task Force, July 2015. 39 pp. DOI: 10.17487/RFC7591. URL: https://rfc-editor.org/rfc/rfc7591.txt.

[82] Philippe Rioux. *Étude : L'usurpation d'identité numérique en France*. 2016. URL: http://www.technomedia.org/2016/03/etude-lusurpation-didentite-numerique.html.

[83] Nat Sakimura et al. *OpenID Connect Core 1.0 incorporating errata set 1*. Tech. rep. Last checked: May 2021 [online]. OpenID, 2014. URL: https://openid.net/specs/openid-connect-core-1_0.html.

[84] Bruce Schneier. *Applied Cryptography (2nd Ed.): Protocols, Algorithms, and Source Code in C*. USA, 1995.

[85] Henning Schulzrinne. *The tel URI for Telephone Numbers*. Tech. rep. 3966. Internet Engineering Task Force, Dec. 2004. 17 pp. DOI: 10.17487/RFC3966. URL: https://rfc-editor.org/rfc/rfc3966.txt.

[86] Nigel Shadbolt. "Midata: towards a personal information revolution." In: *Digital Enlightenment Yearbook* (2013), pp. 202–224.

[87] Adi Shamir. "How to Share a Secret." In: *Commun. ACM* 22.11 (Nov. 1979), pp. 612–613. ISSN: 0001-0782. DOI: 10.1145/359168.359176. URL: http://doi.acm.org/10.1145/359168.359176.

[88] Jonathan Stempel and Jim Finkle. "Yahoo says all three billion accounts hacked in 2013 data theft." In: *Reuters* (Oct. 2017). Last checked: June 2021. URL: https://www.reuters.com/article/us-yahoo-cyber/yahoo-says-all-three-billion-accounts-hacked-in-2013-data-theft-idUSKCN1C82O1.

[89] The French public action modernization Web portal. *« Dites-le nous une fois » : un programme pour simplifier la vie des entreprises*. http://www.modernisation.gouv.fr/les-services-publics-se-simplifient-et-innovent/par-des-simplifications-pour-les-entreprises/dites-le-nous-une-fois-un-programme-pour-simplifier-la-vie-des-entreprises. Last checked: March 2019. 2018.

[90] The OpenStack Foundation. *Keystone, The OpenStack Identity Service (documentation)*. Tech. rep. Last checked: May 2021 [online]. OpenStack. URL: https://docs.openstack.org/keystone/pike/.

[91] The Unicode Consortium. *The Unicode Standard*. Tech. rep. Version 6.0.0. Mountain View, CA: Unicode Consortium, 2011. URL: http://www.unicode.org/versions/Unicode6.0.0/.

[92] Mark Turner, David Budgen, and Pearl Brereton. "Turning software into a service." In: *Computer* 36.10 (2003), pp. 38–44. ISSN: 0018-9162. DOI: 10.1109/MC.2003.1236470.

[93] Sabrina De Capitani di Vimercati, Sara Foresti, and Pierangela Samarati. "Authorization and Access Control." In: *Security, Privacy, and Trust in Modern Data Management*. Ed. by Milan Petković and Willem Jonker. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 39–53. ISBN: 978-3-540-69861-6. DOI: 10.1007/978-3-540-69861-6_4. URL: https://doi.org/10.1007/978-3-540-69861-6_4.

[94] Jane Wakefield. "eBay faces investigations over massive data breach." In: *BBC* (May 2014). Last checked: June 2021. URL: http://www.bbc.com/news/technology-27539799.

[95] Frank Wang et al. "Sieve: Cryptographically Enforced Access Control for User Data in Untrusted Clouds." In: *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation*. NSDI'16. Santa Clara, CA: USENIX Association, 2016, pp. 611–626. ISBN: 978-1-931971-29-4. URL: http://dl.acm.org/citation.cfm?id=2930611.2930651.

[96] Misha Wolf and Charles Wicksteed. *Date and time formats*. Tech. rep. W3C, 1997. URL: https://www.w3.org/TR/NOTE-datetime.

[97] H. Vicky Zhao et al. "Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting." In: *IEEE Transactions on Image Processing* 14.5 (2005), pp. 646–661. DOI: 10.1109/TIP.2005.846035.

[98] Michael Zolotarev et al. *Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols*. RFC 3029. Feb. 2001. DOI: 10.17487/RFC3029. URL: https://rfc-editor.org/rfc/rfc3029.txt.