

An Identity-matching Process to Strengthen Trust in Federated-identity Architectures

Paul Marillonnet^{1,2}^a, Mikaël Ates¹, Maryline Laurent²^b and Nesrine Kaaniche³^c

¹*Entr'ouvert, Paris, France*

²*SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, France*

³*Security of Advanced Systems, Department of Computer Science, University of Sheffield, U.K.*

Keywords: Identity Matching, Federated-identity Architecture, Identity Management, Citizen-relationship Management, Trust Enforcement.

Abstract: To smoothly counteract privilege escalation in federated-identity architectures, the cross-checking of asserted Personally Identifiable Information (PII) among different sources is highly recommended and advisable. Identity matching is thus a key component for supporting the automated PII cross-checking process. This paper proposes an efficient identity-matching solution, adapted to a chosen User-Relationship Management (URM) platform, relying on a French Territorial Collectivities and Public Administrations (TCPA) use case.

The originality of the paper is threefold. (1) It presents an original solution to identity-matching issues raised by a concrete use case from the Territorial Collectivities and the Public Administration (TCPA), formalizing concepts such as information completeness, PII normalization and Levenshtein-distance matrix generation. (2) Implementation guidelines are given to deploy the solution on an operational Publik platform. (3) A precise security analysis is provided, relying on an original attacker model.

1 INTRODUCTION

To smoothly counteract users overriding their privileges (Zhao et al., 2005; Bugiel et al., 2012) derived from their Personally Identifiable Information (PII) in Federated-identity architectures, it is now commonly assumed that declared PII are cross checked among several sources. Of course the reliability of the identity-matching process fully relies on the quality of the identity attributes provided by the sources, and the level of trust of the sources.

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 their use in the Internet public key infrastructure (Zolotarev et al., 2001; Hunt, 2001)). *Certified* identity information (i.e. *cer-*

tificates) can take the classic form of assertions in the Security Assertion Markup Language (Organization for the Advancement of Structured Information Standards, 2005) (SAML), which are still used in federated-identity architectures. *Validated* identity information are increasingly expanding through service providers using OpenID Connect (OIDC) (Sakimura et al., 2014) protocol (rather than SAML), 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.

The use case considered in this paper is the French Territorial Collectivities and Public Administration (TCPA), which are encouraged to use such validated sources of identity information, in order to simplify citizen online services, thus moving towards the "Tell

^a  <https://orcid.org/0000-0003-2834-9004>

^b  <https://orcid.org/0000-0002-7256-3721>

^c  <https://orcid.org/0000-0002-1045-6445>

us once” program¹

This paper presents the necessary measures when performing identity matching in distributed identity architectures for a specific use case. This use case comes from the domain of user-relationship management (URM) in territorial collectivities and public administrations (TCPA). This use case requiring specific identity-matching procedures has not been presented in the academic literature so far. In order to do so, this paper 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 article. The security analysis proves the security suitability of the solution against four types of identified threats. Finally, implementation guidelines for audience willing to reproduce the ready-for-production solution on their own are given.

The remainder of the paper is as follows. Section 2 describes the related work. Section 3 defines the selected use case for territorial collectivities and public administration (TCPA), motivating the need for an identity-matching automated process. Section 4 describes the different identity- and personally-identifiable-attribute sources relevant to our use case. Section 5 defines the identity matching procedure to follow when combining user data from such sources. Section 6 gives the aforementioned security analysis of the identity-matching procedure within the citizen-relationship management environment, and Section 7 gives the conclusions. Eventually, Section 7 gives a brief conclusion and provides some perspective to this ongoing identity-management research.

2 RELATED WORK

Federated-identity architectures and their shortcomings have been widely described in the literature. For instance, (Camenisch and Pfitzmann, 2007, Chapters 1, 2 and 3) provide an analysis on their shortcomings regarding user privacy. However, no academic contributions studying the provision of identity and personally-identifiable-attributes by several sources in federated-identity architectures have been elaborated so far.

Additionally, the management of personally-identifiable-attributes sources, in a privacy-compliant way, for user-centric architecture has been studied at large, for instance in (Mortier et al., 2016) and (de Montjoye et al., 2014). Similarly, the use of personally-identifiable attributes for TCPA-based

¹See the dedicated page on the official French *modernisation de l'action publique* website (resource in French).

purposes has been proposed in (Papadopoulou et al., 2015) and (Shadbolt, 2013). However, these four contributions do not 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. 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.

3 PROBLEM STATEMENT

3.1 The TCPA Use Case

The main use case considered in this paper is the enrollment of the user's children to the school restaurant of her territorial collectivity in France. 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. With regard to our use case, these territorial collectivities are responsible for the children enrollment to schools that belong to their territory. Such collectivities usually provide an online service for parents to pay the school restaurant fees.

As a result the different actors of the use case are:

- The parent or the legally-responsible of the child or children. With regards to our use case, the parent is the user of the school restaurant enrollment online service.
- The user-relationship management (URM) online platform, providing the school restaurant enrollment service.
- The *FranceConnect*² official federated-identity service of the French administration.
- The DGFIP³ personally-identifiable-attribute source.
- The CNAF⁴ personally-identifiable-attribute source.

As defined in French collectivities, the school restaurant fees depend on the parents' tax information (and in particular their tax reference revenue document) as

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)

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 enrolling their children to the school restaurant, the parent fills an online form which requires them to provide the following information:

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

Items 1. and 2. are required to retrieve the user’s children’s allowance information thanks to the CNAF attribute source. Similarly, items 3. and 4. are required to retrieve the user’s tax information thanks to the DGFIP attribute source. All four items are retrieved through the *API Particulier*⁵. Put in place in 2017 by DINUM⁶, this API offers an access to the two aforementioned attribute sources through two different endpoints, accessible for TCPAs after registration and obtention of a client-specific token.

When necessary, the identity matching, either automated or performed manually by the agent as described later in Section 5, happens on the PII provided by *FranceConnect* and the *API Particulier* endpoints.

Generally, when completing online procedures, citizens are expected to prove their identity. In order to do so, the enrollment 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.

In any case the user identity needs to be validated as properly matching with the different personally-identifiable attributes provided by the sources. Failing to adequately address this procedure hinders the completion of citizen-relationship management.

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:

⁵<https://particulier.api.gouv.fr/> (resource in French).

⁶<https://www.numerique.gouv.fr/dinum/> (resource in French).

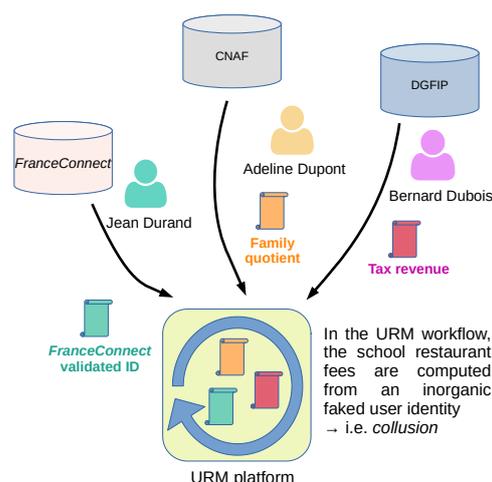


Figure 1: A classic collusion case illustrated.

- 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 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 (Fielding, 2000).

3.2 Functional Requirements

The use case results in the following list of functional requirements, necessary for the TCPA to:

- Prove a user’s identity through the information provided by the available sources.
- Ensure that all the personally-identifiable attributes 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 1.
- Tackle the numerous true positive cases of identity-matching in an autonomous manner.
- Collect (either true or false) negative cases, as well as ambiguous cases, that need thorough inspection by a TCPA agent.

The following section provides a more thorough description of the aforementioned identity- and personally-identifiable-attribute sources.

4 IDENTITY SOURCES

4.1 *FranceConnect*

FranceConnect is the official identity federation service of the French administration. The identity information it uses comes from the INSEE's⁷ RNIPP⁸. It implements the OpenID Connect (OIDC) (Sakimura et al., 2014) identification layer, itself derived from the OAuth 2.0 authorization framework (Hardt, 2012). 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 regulation⁹.

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.

4.2 DGFIP

As explained in Section 3, 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

⁷Institut National de la Statistique et des Études Économiques, i.e. the national institution for statistics and economical studies.

⁸Registre National d'Identification des Personnes Physiques, i.e. the national register for identification of French-living individuals –see Section 3.

⁹<https://www.ssi.gouv.fr/entreprise/reglementation/confiance-numerique/le-reglement-eidas/> (resource in French).

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 prerequisite 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.

4.3 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 4.2, 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 post-code.

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.

5 IDENTITY MATCHING

5.1 Motivations for an Identity-matching Automated Procedure

For an Advanced Identity-matching Procedure.

Let us consider a first naive 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 naive approach of straightforward equality testing would require to build, for a given identity attribute I , a *result* vector as follows:

$$result_j = S_j(I), \forall j \in \{1, \dots, n\} \quad (1)$$

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

For each function attribute, the validation process would be as follows, for each I in I^* , where I^* is the set of all available PII attributes:

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

However, adopting this approach will raise false negatives¹⁰, 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 code that can be compared after extraction, to 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, and using that cardinality value as part of the identity matching decision process is the first step to the concept of *information completeness*, presented later in this article, in Section 5.3.

For an Automated Procedure. Let us consider the validation of PII by a (human) agent of a collectivity. The agent can identify slight variations and compare information presented to him in different formats. In our TCPA use case, the agent, in order to perform a single identity-matching step, is 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 – especially considering their thorough knowledge of their collectivity’s citizen-relationship management procedures.

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

¹⁰That is theoretically matching identities which are detected as mismatches.

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 5.2 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 3.

5.2 Presentation of the Use Case Identity Matching Procedure

According to the use case defined in Section 3, the user-relationship management platform, acting as a service provider to the identity- and personally-identifiable-attribute sources, relies on the identity provided by the *FranceConnect* service.

The user-relationship management platform has to ensure that the identity provided by one of the *FranceConnect* identity providers 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 birthdate.

The CNAF endpoint returns (i) the user’s full postal address, containing their postcode, (ii) a string representation of the user’s birthdate (ii) 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 birthdate and (v) a string representation of the user’s postal address, containing the postcode.

Using all the information provided by these three sources, our goal is to cover most (if not all) identity-matching cases with simple algorithms.

5.3 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.

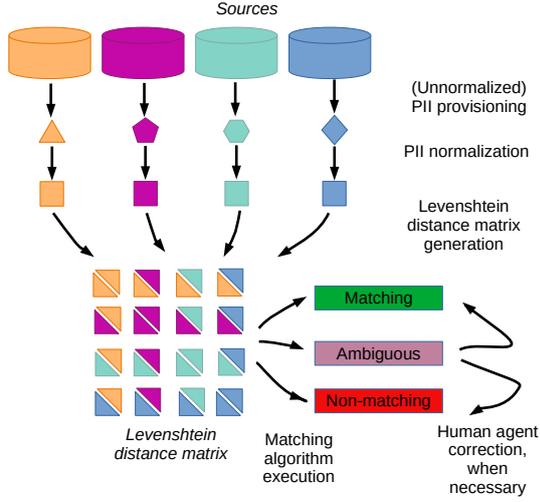


Figure 2: Visual summary of the identity matching process for a *complete* PII attribute.

5.3.1 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:

$$S \text{ provides } t \quad (2)$$

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 article.

• Partial Information.

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

$$\forall t \in \mathcal{T}_p, \exists S \in \mathcal{S}, \neg(S \text{ provides } t) \quad (3)$$

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

5.3.2 Complete Information

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

$$\forall t \in \mathcal{T}_c, \forall S \in \mathcal{S}, S \text{ provides } t \quad (4)$$

5.3.3 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 5.2, 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:

$$\forall t \in \mathcal{T}_s, \exists S, S' \in \mathcal{S}^2, \quad (5)$$

$S \text{ provides } t \text{ and } S' \text{ provides } t$

5.3.4 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:

$$\forall t \in \mathcal{T}_i, \exists S \in \mathcal{S}, \quad (6)$$

$(S \text{ provides } t \wedge (\forall S' \neq S \in \mathcal{S}, \neg S' \text{ provides } t))$

5.4 Validation Algorithm

5.4.1 Format Unification

The comparison as well as the detection mechanisms are based on an ASCII representation of identity information (i), (ii) and (iii) (see Section 5.2). More generally, for any given PII type, defining a format-unification procedure is necessary.

As explained in the problem statement (Section 3), 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 France-Connect data source (YYYY-MM-DD), the DGFIP data source (DD/MM/YYYY) and the CNAF data source (DDMMYYYY).

Similarly, the DGFIP source 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 other sources.

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

$$\forall j \in \{1, \dots, n\}, \exists P_t : P_t(S_j(t)) \in \mathcal{C} \quad (7)$$

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.

5.4.2 Normalization of Unicode Strings

Let us consider the case of a family name that contains non-strictly-Latin characters, *e.g.* *Smicz*, 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 *Smicz*, or even *Smicz*.

According to the Unicode specifications (The Unicode Consortium, 2011), a normalization form involving a compatibility decomposition is most appropriate. The *compatibility* decomposition form (NFKD¹¹) is favored over the *canonical* form (NFD¹²) 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.

5.4.3 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 (Levenshtein, 1966),

¹¹Normalization form by compatibility decomposition.

¹²Normalization form by canonical decomposition.

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 Σ^* , which is the set of all possible strings made 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 $(\Sigma^*)^2 \rightarrow \mathbb{N}$ 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 personally-identifiable-attribute values across several sources. One Python software implementation is the `python-levenshtein` software module¹³, providing a simple API for computing the edit distance between two strings.

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 5.4.2), and the global distance of a result vector is the following matrix representation:

$M(result_i) \in \mathcal{M}_{mm}(\mathbb{R}^+)$ where $i \in \{1, \dots, n\}$ is the source index.

M is made of elements $m_{ij}, i \in \{1, \dots, n\}, j \in \{1, \dots, n\}$ 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).

5.5 Our 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.

¹³<https://pypi.org/project/python-Levenshtein/>

5.5.1 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 (Sakimura et al., 2014, Chapter 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 DDDMMYYYY format.

Sources are considered trustworthy regarding date consistency, therefore no semantic date validation is performed. Django custom template filters are therefore used to perform format unification. This template filter is based on the python standard `datetime` library. When providing the input format, it makes it possible for us to compare the several dates provided by the available data sources.

5.5.2 String Types

String types, as mentioned in Section 5.4.1 are normalized using the `unicodedata` python module normalization algorithm, set to the NFKD form (Davis and Dürst, 2001). 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 5.4.3.

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. Extracting the first name is performed directly thanks Python's string manipulation functions.

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

5.5.3 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: it is returned only by the *FranceConnect* service (and only when the user is born in France).

5.6 Implementation Considerations

This section relies on the Publik URM software suite. Licensed as AGPLv3 free software, its sources are available on its project management webpage¹⁴.

This software can either be installed as Debian packages or directly from sources using an Ansible playbook. For our experimental setup, a development instance of the Publik software suite is installed, using the community documentation¹⁵ of the software installation process.

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

- User-oriented software entities, offering URM features such as content management, appointment-making with TCPA agents, or scanned documents depository.
- Similarly, the 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 URM platform.

5.6.1 Implementation of the Validation Process in 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 Section 3, 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. 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)

¹⁴<https://dev.entrouvert.org/>

¹⁵<https://doc-publik.entrouvert.com/dev/installation-developpeur/> (resource in French).

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.

2. compute the distance between elements of a result vector of normalized PII from our three identity- and personally-identifiable-attribute sources.
3. 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(
            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

$$\begin{aligned}
 f: & \\
 \mathcal{M}_{mm}(\mathbb{R}^+) & \longrightarrow \\
 \{ \text{"Matching"}, \text{"Non-matching"}, \text{"Ambiguous"} \} & \\
 M((result_i), i \in \{1, \dots, n\}) & \longmapsto s
 \end{aligned}
 \tag{8}$$

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

For instance, a Python implementation of the algorithm described in Section 5.4.3.

```
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'
```

5.6.2 Additional Identity Matching Solvability Parameters

Eventually, and before presenting an example of validation results (in Section 5.6.3, 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 $x \in \{2, \dots, n-1\}$, 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 identity- and personally-identifiable-attribute 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.

5.6.3 Example of Validation Results

Matching PII. Following the previously given example, the three sources return respectively the values *Smiçz*, *Smicz* and *Smicz*.

Consequently, the result vector as defined in Section 5.4.3 is (*Smiçz*, *Smicz*, *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.

Potentially Non-matching or Ambiguous PII.

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

The result vector with this PII is (*Smięcz*, *Smicz*, *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 2. 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.

6 SECURITY ANALYSIS

6.1 Model and Requirements

6.1.1 Preliminary Definitions

We define a group of users $\mathcal{G} = \{u_1, \dots, u_k\}$, $k > 2$ bringing a set of information:

$$I^* = \{I_{u_1,1}, \dots, I_{u_1,v_1}, \dots, I_{u_k,1}, \dots, I_{u_k,v_k}\} \quad (9)$$

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

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 5.5 –, the normalized value of PII I as defined in Section 5.4.2, the source, and the user from which the instance of information I is originated.

In particular, *type* is defined as follows:

$$\begin{aligned} type : I^* &\longrightarrow \mathcal{T} \\ I &\longmapsto t \end{aligned} \quad (10)$$

where \mathcal{T} is the set of all available PII types as defined in Section 5.5. Thus t is the actual PII type for

information I .

Similarly *norm* is defined as:

$$\begin{aligned} norm : I^* &\longrightarrow \Sigma^* \\ I &\longmapsto n_I \end{aligned} \quad (11)$$

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:

$$\begin{aligned} source : I^* &\longrightarrow \mathcal{S} \\ I &\longmapsto S \end{aligned} \quad (12)$$

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 $I \in I^*$ – and the fact that I characterizes a given user u is not of interest for this definition.

Eventually, *user* is defined as:

$$\begin{aligned} user : I^* &\longrightarrow \mathcal{G} \\ I &\longmapsto u \end{aligned} \quad (13)$$

Thus u is the user bringing PII I .

6.1.2 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 I^* (as defined in Section 6.1.1) such as:

$$\forall I \in I^*, \exists S \in \mathcal{S}, S \text{ provides } I \quad (14)$$

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 $\mathcal{G} = \{u_1, \dots, u_k\}$ should not be completely homonymous regarding the available sources. As informally stated in Section 3, this means that if

$$\begin{aligned} \forall I, I' \text{ in } I^*, \\ (type(I) = type(I')) \wedge \\ source(I) = source(I') \wedge \\ user(I) \neq user(I') \\ \implies norm(I) = norm(I') \end{aligned} \quad (15)$$

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 matrixial terms. If for each result vector *result* of a given PII type $t \in \mathcal{T}$, if we have:

$$\forall i \in \{1, \dots, n\}, \text{result}_i = \begin{cases} \text{empty} & \text{if } \neg S_i \text{ provides } t \\ \alpha & \text{otherwise} \end{cases}$$

where α is a constant value for *result*, then the collusion is not preventable.

• **Identity/Attribute Theft Attack:** on one or several sources.

This means that a user's credentials to one or several identity- or personally-identifiable-attribute sources have been stolen by a rogue user.

This means that a rogue user u_r knows a subset of the credentials $\{c_{u_h,1}, \dots, c_{u_h,n}\}$ of an honest user u_h , used for authentication to sources S_1, \dots, S_n .

• **Man-in-the-Middle Attack:** tampering data from one or several of the sources.

More formally, if $I = \{i_1, \dots, i_n\}$ is the set of information retrieved from the remote sources S_1, \dots, 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.

6.1.3 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 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 identity- and personally-identifiable-attribute information served by the sources.
- **Requirement 3.** Identity mismatches and attempted attacks should be detectable.

6.2 Security and Resilience Analysis

6.2.1 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 6.1.2:

1. A thorough identity matching process prevents user collusion: The group of colluding users $\mathcal{G} = \{u_1, \dots, u_k\}$, $k > 2$ manages to retrieve information from the complete set of sources $\mathcal{S} = \{S_1, \dots, S_n\}$. Thus, as explained in Section 6.1.2, each colluding member in \mathcal{G} brings some information retrieved from one or several source. The function g is defined as follows:

$$g: \mathbb{N} \longrightarrow \mathbb{N} \\ l \longmapsto m \quad (16)$$

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 $m_{i,j}, i \in \{1, \dots, n\}, j \in \{1, \dots, n\}$ 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 5.4.3).

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 $M((\text{result}_i), i \in \{1, \dots, n\})$ is made of elements $m_{i,j}, i \in \{1, \dots, n\}, j \in \{1, \dots, n\}$ where $n = |\mathcal{S}|$ is the number of available sources, we have:

$$m_{i,j} = \begin{cases} \text{empty} & \text{if } \text{result}_i \text{ or } \text{result}_j \text{ is missing} \\ 0 & \text{otherwise} \end{cases}$$

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 $\mathcal{R} \subset \mathcal{S} = \{S_1, \dots, S_n\}$ of the remote sources acts in the following manner: For any given source $S_j, j \in \{1, \dots, n\}$, 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 $m_{i,j}$, $i \in \{1, \dots, n\}$, $j \in \{1, \dots, n\}$ 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 *France-Connect* 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 neglectable.

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 $\mathcal{R} \subset \mathcal{S} = \{S_1, \dots, S_n\}$ 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 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 $m_{i,j}$, $i \in \{1, \dots, n\}$, $j \in \{1, \dots, n\}$ 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 sources belonging to $\mathcal{R}' = \mathcal{S} \setminus \mathcal{R}$, where \setminus is the set difference operator. The attacker is therefore unable to tamper PII attributes tamper 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 $\mathcal{S} \setminus \mathcal{R} \neq \{\}$.

Conclusion 3. Under the fourth assumption, man-in-the-middle attacks leading to users' identity attributes tampering can also be prevented. For instance, (Krawczyk et al., 2013) 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 layer¹⁶ 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) (Rescorla, 2018, 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 6.1.2, 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.

6.2.2 Requirements Enforcement Even under the Attack Model

Similarly, the resilience and security requirements identified in Section 6.1.3 can be validated as follows:

- **Enforcement of Requirement 1.** The solution is proved resistant against the four types of attacks defined in Section 6.1.2. Additionally, the attacker model does not include direct attacks on the URM platform. The resistance against such direct attacks are not specific to the application but depend on the

¹⁶The most popular implementation of TLS being OpenSSL.

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 identity- and personally-identifiable-attribute 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.

7 CONCLUSION

Identity matching across multiple identity- and personally-identifiable-attribute sources in a federated-identity environment has become a challenging concern as the number of official sources is increasing.

These sources tend to adopt widely accepted authentication (Sakimura et al., 2014) and authorization (Hardt, 2012) standards. However, these standards do not offer out-of-the-box solutions for matching the users' digital identities and personally-identifiable attributes across multiple sources, and as a result identity mismatch errors happen.

The proposed identity matching solution supports efficient automated processing, that requires human assistance for a limited number of corner cases. These corner cases involve interactions with the user for possible identity verification, as well as more subtle and subjective validation

At the time of writing, the French government is experimenting the use of *FranceConnect* data providers and has published technical documentation about such providers. These providers, acting each as a resource server according to the OAuth 2.0 authorization management protocol (Hardt, 2012), must proceed to identity matching between the identity conveyed by the authorization server and their local user base. This experimental identity- and personally-identifiable-attribute flow, if adopted nationally at production level for official online procedures, will shift the duty of identity-matching from the service providers to these official *data providers*. As a consequence, the proposed automated procedure remains relevant once the aforementioned experimentation will be brought to production level, as it will need to be ensured by the data providers themselves.

REFERENCES

- Bugiel, S., Davi, L., Dmitrienko, A., Fischer, T., Sadeghi, A.-R., and Shastri, B. (2012). Towards taming privilege-escalation attacks on android. In *NDSS*, volume 17, page 19. Citeseer.
- Camenisch, J. and Pfitzmann, B. (2007). *Federated Identity Management*, pages 213–238. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Davis, M. and Dürst, M. (2001). Unicode normalization forms.
- de Montjoye, Y.-A., Shmueli, E., Wang, S. S., and Pentland, A. S. (2014). openpds: Protecting the privacy of metadata through safeanswers. *PLOS ONE*, 9(7):1–9.
- Fielding, R. T. (2000). *REST: Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine.
- Hardt, D. (2012). The OAuth 2.0 Authorization Framework. RFC 6749.
- Hunt, R. (2001). Pki and digital certification infrastructure. In *Proceedings. Ninth IEEE International Conference on Networks, ICON 2001.*, pages 234–239.
- Krawczyk, H., Paterson, K. G., and Wee, H. (2013). On the security of the tls protocol: A systematic analysis. In Canetti, R. and Garay, J. A., editors, *Advances in Cryptology – CRYPTO 2013*, pages 429–448, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.
- Mortier, R., Zhao, J., Crowcroft, J., Wang, L., Li, Q., Haddadi, H., Amar, Y., Crabtree, A., Colley, J., Lodge, T., Brown, T., McAuley, D., and Greenhalgh, C. (2016). Personal data management with the databox: What's inside the box? In *Proceedings of the 2016 ACM Workshop on Cloud-Assisted Networking, CAN '16*, pages 49–54, New York, NY, USA. ACM.
- Organization for the Advancement of Structured Information Standards (2005). Security assertion markup language (saml) v2.0.
- Papadopoulou, E., Stobart, A., Taylor, N. K., and Williams, M. H. (2015). *Enabling Data Subjects to Remain Data Owners*, pages 239–248. Springer International Publishing, Cham.
- Rescorla, E. (2018). The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446.
- Sakimura, N., Bradley, J., Jones, M., De Medeiros, B., and Mortimore, C. (2014). Openid connect core 1.0 incorporating errata set 1.
- Shadbolt, N. (2013). Midata: towards a personal information revolution. *Digital Enlightenment Yearbook*, pages 202–224.
- The Unicode Consortium (2011). The Unicode Standard. Technical Report Version 6.0.0, Unicode Consortium, Mountain View, CA.
- Zhao, H. V., Min Wu, Wang, Z. J., and Liu, K. J. R. (2005). Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting. *IEEE Transactions on Image Processing*, 14(5):646–661.
- Zolotarev, M., Sylvester, P., Zuccherato, R., and Adams, D. C. (2001). Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols. RFC 3029.