Technical application architecture of Pseudalis Community

A Windows service that finds personal data with four detection devices and replaces it by format-preserving encryption

Aldric Pierrain
Cydral Technology, Paris, France
contact@cydral-technology.eu

8 November 2026

Abstract

Pseudalis Community protects the personal data of documents on the Windows computer that holds them, and is distributed under the Boost Software License 1.0. A local service does the work; the Explorer, a tray icon, a web application, a drive and an HTTPS interface give access to it. Its engine finds personal data with four devices: scanners that verify the check digits of identifiers, word tables in five languages, a structural SVM tagger and a small transformer that reads bytes. Each value is replaced by a tag that carries its encryption under FF1 and AES-256 with a short check, or by the encrypted value alone, and the codecs keep layouts, check digits and the intervals between dates. On a synthetic test set in five languages the detector reaches an F1 of 0.945 and reads about 41 000 characters per second on two cores, and on hand-written texts its F1 lies between 0.889 and 0.899. Under the GDPR, the output remains pseudonymized personal data for whoever holds the key.

Keywords: *pseudonymization; personal data detection; format-preserving encryption; FF1; structural SVM; byte-level transformer; GDPR; dlib*

1. Introduction

Pseudalis Community protects the personal data of documents on a Windows computer, on that computer alone [1]. It is written by Aldric Pierrain, copyright Cydral Technology, distributed under the Boost Software License 1.0, and its installer and signed update feed are published on GitHub [2]. Its README presents the protection as a pseudonymization in the sense of the GDPR [1].

The application rests on Dlib-Transformer-extensions [3], a fork of the dlib C++ library [4]. Its engine compiles the module `dlib/privacy` [5], which holds the detector, two learned recognizers, a format-preserving pseudonymizer built on FF1 of NIST SP 800-38G [6, 7] and the readers and writers of Office, PDF and text documents. A coverage matrix maps the detected categories to the GDPR [8, 9], and evaluation reports give the scores of the successive models [10].

The report describes the application, the detection, the pseudonymization, the dataset and the measured performance, then relates the output to the GDPR. It rests on the source code as published on 8 November 2026, on the evaluation reports, and on runs and time measurements made that day on synthetic data, on a virtual machine with two cores. Section 7 is a technical reading of the texts, which leaves the legal qualification to the controller.

2. Application architecture

2.1 Programs and engines

Pseudalis installs one Windows service, three programs of the user session and a web application (Table 1). The service runs as LocalSystem and holds everything that keeps state [1]. Figure 1 shows these components and the two ways in for documents, the HTTPS interface and the drive. The signed update feed (Section 2.5) lies outside the service, as does Albireo, an optional local inference service built on the same library, which can describe pictures (Section 2.6).

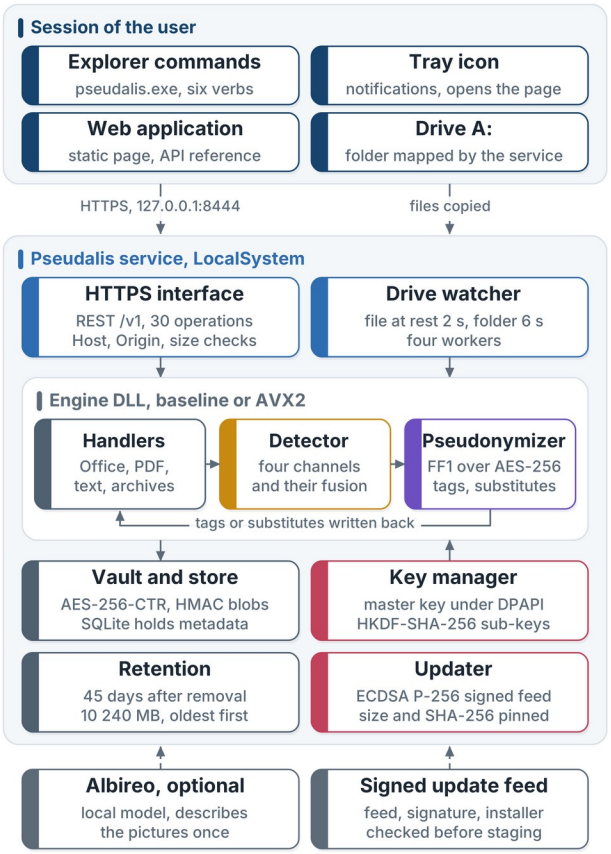


Figure 1. Components of Pseudalis Community: programs of the user session, the LocalSystem service with its engine library, and the optional Albireo service and signed update feed outside it.

File	Role
`pseudalis-service.exe`	LocalSystem service: engine host, drive, vault, database, HTTPS interface, updater
`pseudalis-engine.dll`	engine for every x64 processor
`pseudalis-engine-avx2.dll`	engine compiled for AVX2 and FMA
`pseudalis-tray.exe`	tray icon, started at each sign-in
`pseudalis.exe`	Explorer commands and command line
`pseudalis-setup.exe`	installer, kept as uninstaller
`web\`	web application and interface documentation

Table 1. Files installed by Pseudalis Community and their roles [1].

The two engines come from the same sources and export the same C interface; the trained models are compiled into both. The service picks the AVX2 engine when the processor and Windows support it. Before any code of an engine runs, the service checks that its build stamp matches its own, so a stale or planted library is refused. The programs carry their runtime and import only libraries of Windows, which a check of the build enforces.

2.2 Local HTTPS service

The service answers on 127.0.0.1, port 8444 by default, over TLS. Its certificate comes from a local authority created at installation, limited to localhost and the loopback addresses, whose private key signs once and is never written. Before reading a request, the service refuses a foreign `Host` or `Origin` and an oversized body, so a web page open in the browser cannot drive it; a local process can, which the README accepts as the trust model of a computer [1].

The interface offers 30 operations under `/v1`, described by an OpenAPI document with a Swagger UI page. File, text and name operations need no token, and administration routes need a token that the tray hands to the pages it opens (Table 2).

Route	Token	Use
`POST /v1/anonymize`	no	protect a file
`POST /v1/restore`	no	restore a file
`POST /v1/text/anonymize`	no	protect a text
`POST /v1/names/anonymize`	no	protect a file name
`GET /v1/info`	no	status, key fingerprint
`GET /v1/records`	yes	list the records
`DELETE /v1/records/{id}`	yes	forget a record
`PUT /v1/settings`	yes	change the settings
`PUT /v1/terms`	yes	set the imposed terms
`PUT /v1/columns`	yes	set the imposed columns
`POST /v1/key/export`	yes	export the key

Table 2. Main routes of the interface [1].

2.3 Drive, Explorer commands, tray icon and web application

The drive needs no file-system driver. The service gives a letter, A: by default, to a folder of its data folder, and watches it: a file that has rested for two seconds is protected in place, and a file it cannot process moves to a folder that only its owner reads. The written file lies in clear on the disk until then, a limit that the README states [1].

The Explorer offers six commands, to protect a file or a copy with or without tags and to restore it, on selections of up to 100 files. The tray icon follows the service, shows its notifications and opens the web application, which protects and restores files, folders and texts and, for administrators, manages the records, the key, the settings, the imposed terms and the imposed columns. The engine reads Word, Excel and PowerPoint files, PDF, text files of any extension that decode cleanly, pictures and ZIP or TAR archives, within bounds on the size an archive may expand to.

2.4 Vault, database and keys

The vault keeps the originals, the pictures and their descriptions, each encrypted with AES-256 in CTR mode and authenticated by an HMAC-SHA-256 that the service checks before any decryption [11, 12, 13]. The SQLite database holds only metadata: names, sizes, hashes, places, events, imposed terms and columns and the pairs of substitute and tag, never a content or a value in clear.

At its first start, the service draws a master key of 32 bytes and protects it with DPAPI for the LocalSystem account, in a folder that only SYSTEM and the administrators open. HKDF-SHA-256 [14] derives from it the key of the pseudonymization and the keys of the vault. The export of the key, which serves as the backup of an installation, seals it under a passphrase of at least 12 characters with PBKDF2 over 600 000 iterations, the count that OWASP recommends [15]. Every user of the computer shares its key, its originals and its administration, so people who must not see each other's documents need separate computers [1].

2.5 Signed updates

The service reads the update feed on GitHub five minutes after its start, then every day [2]. It checks an ECDSA P-256 signature over the exact bytes of the feed with a public key compiled into the programs, downloads the installer that the feed names, checks its size and its SHA-256, and runs it at the next start of the service. The private key that signs the feed never enters the repository.

2.6 Protection, restoration, retention and forgetting

Protection detects the personal data of a document, writes the protected file and, when the original is kept, records it: the original goes to the vault and its names and hashes to the database (Figure 2). Values in the content become tags `[[TYPE:body#CCC]]`; file names, entries of archives and document properties take the encrypted value alone, recorded with its tag.

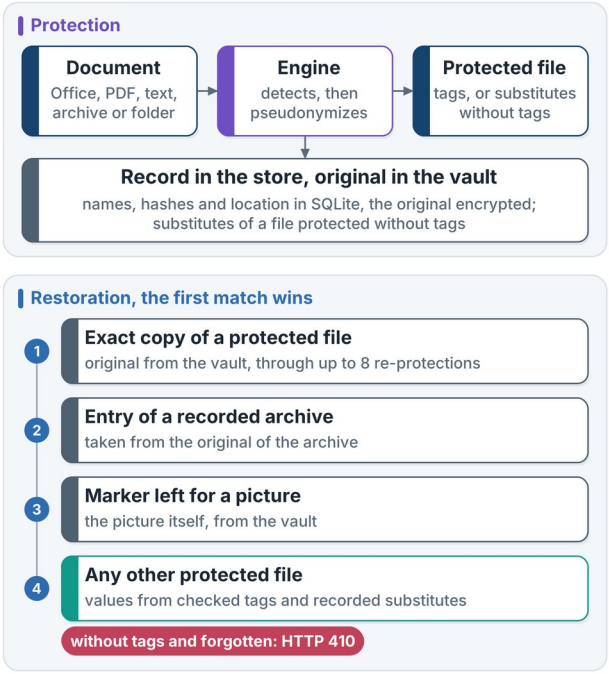


Figure 2. Lifecycle of a document: protection with its record, and the four restoration paths tried in order, with HTTP 410 for a file without tags whose record was forgotten.

Pictures leave the documents. The engine stores each picture in the vault and writes a marker `[Image@<id>]` in its place, or a gray box in a PDF, and it reads no text inside pictures, so a scanned page loses its text [1]. Albireo, another product of the author built on the same Dlib-Transformer-extensions library and running on the same computer, can describe the pictures. When it is installed, Pseudalis asks it for a description of each picture, pseudonymizes the description like any text and writes `[Image@<id> : <description>]` in the place of the picture.

Restoration tries four paths in order. An exact copy of a protected file returns its original from the vault; an entry of a recorded archive returns its original entry; a marker returns its picture; any other file goes through the engine, which restores every tag whose check verifies and the substitutes that the database knows. The originals stay 45 days after the last protected copy leaves the drive, within a space budget of 10 GB whose oldest records leave first. A forgotten file with tags comes back from its tags alone, without its pictures, and a forgotten file without tags is refused with HTTP 410.

2.7 Protection without tags

The values can also be written without their tags, the encrypted value alone, so that the file reads as an ordinary document, for a prompt, test data or a file sent outside [9]. A setting decides for the drive, and each card of the web application and each Explorer command can choose otherwise. Nothing in such a file tells a value from its substitute, so the record keeps the pairs of substitute and tag, and restoration depends on it. An edited copy is restored once the service finds its record again, and a file without record or with a forgotten record is reported as not restorable.

3. Detection of personal data

3.1 Types of data and the two readings of a document

The detector reports 21 types, each named by the keyword of its tag: `PERSON`, `ORG`, `LOCATION`, `ADDRESS`, `EMAIL`, `PHONE`, `IBAN`, `CARD`, `NIR`, `SIRET`, `ID`, `IPV4`, `IPV6`, `MAC`, `URL`, `DATE`, `PLATE`, `GEO`, `TITLE`, `GENDER` and `NATIONALITY`. It reads French, English, Italian, Spanish and German. The two learned recognizers know seven finer labels, such as given name, family name, city or street address, and the structured types come from rules.

Each document is read twice [5]. The first reading learns the names, places and organizations found with strong evidence; the second reports every value, the other mentions of what the first reading learned included. Figure 3 shows the four detection devices and the stages that follow them.

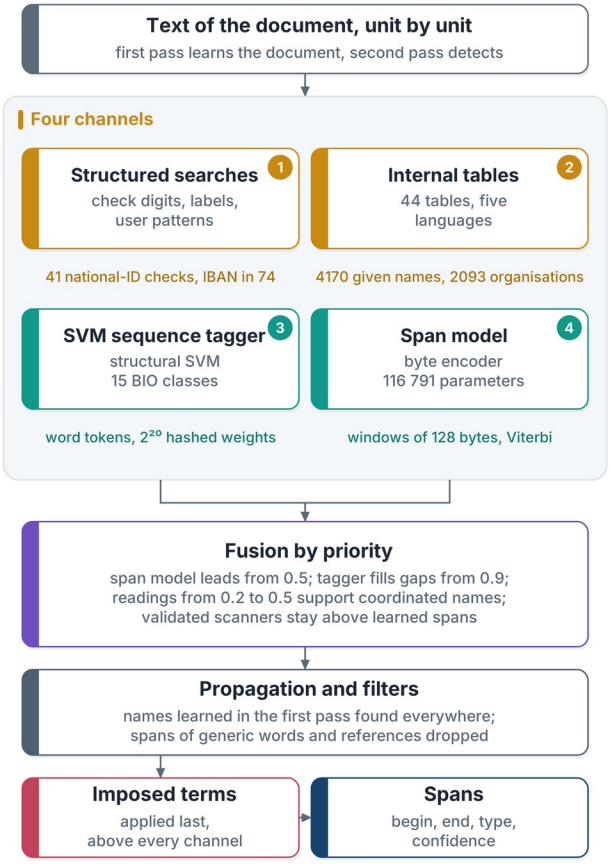


Figure 3. Detection pipeline: two readings of the document, the four detection devices, fusion by priority, document propagation and filters, and the imposed terms applied last, which yield typed spans.

3.2 Device 1: structured searches

The scanners read regular layouts in a single pass and accept a value only when its check digits verify, or after a word that announces it when the number has a weak check or none. An IBAN must have the length of its country and pass mod 97, a card must pass Luhn within the lengths and prefixes of its issuer, and a French social security number must carry a valid month and its key. Further checks cover the national numbers of about forty countries. Dates are read in the five languages, while street names and software versions such as `6.1.24` stay readable.

Field labels type the value that follows them, such as `Nom` before a name, and the structure of a document can force a type, as for a cell under a column headed `E-mail`. An organization adds its own patterns for its references, and the imposed terms, up to 100 names of clients, projects or contracts, are replaced before anything else [1]. In a CSV file with a header, the imposed columns go further: every value below the name of such a column, the names of the clients of an export for instance, takes the substitute of the type of the column without any detection, while the record of the file keeps what restores it [1].

3.3 Device 2: internal tables

The module holds 44 word tables for the five languages (Table 3). The rules turn them into scores: a known given name, a known full name or a civility before a capitalized word make a person more likely, and a label types its value. Laws and standards written in capitals, such as `RGPD`, `NIS2` or `ISO 27001`, are never read as a person, a place or an organization. The tagger reads twelve of the tables as features.

Table	Entries	Role
common words	10 428	words that never name alone
given names	4170	given names of all regions
ambiguous given names	268	given names that are also common words
known organizations	2093	organizations known by name
field labels	838	label to type map
nationalities	756	nationalities
countries	315	countries, left readable
public bodies	548	acronyms and names of public bodies

Table 3. Main internal tables of `dlib/privacy`, with their sizes [5].

3.4 Device 3: the structural SVM tagger

The tagger labels word tokens with 15 classes in the BIO scheme [16, 17]: a class for words outside any value, and a class for the beginning and one for the inside of each of the seven labels. Its features describe each token and its neighbors: folded form, shape, prefixes and suffixes, the flags of the tables and the type announced by a preceding label. They are hashed into about one million weights, whatever the vocabulary. The current tagger was trained afresh on 80 000 examples in about an hour on two cores [10].

3.5 Device 4: the span model

The span model is a small bidirectional transformer encoder [18] that reads the UTF-8 bytes of a text, without tokenizer, and gives each byte one of the same 15 classes. It holds 116 791 parameters in two blocks of attention with rotary positions and feed-forward layers [19, 20, 21, 22] (Figure 4). It reads long texts in windows of 128 bytes, and a constrained decoding turns its probabilities into spans whose ends fall on whole words. It looks both ways, since whether `Rose` is a person depends on the words that follow as much as on those before.

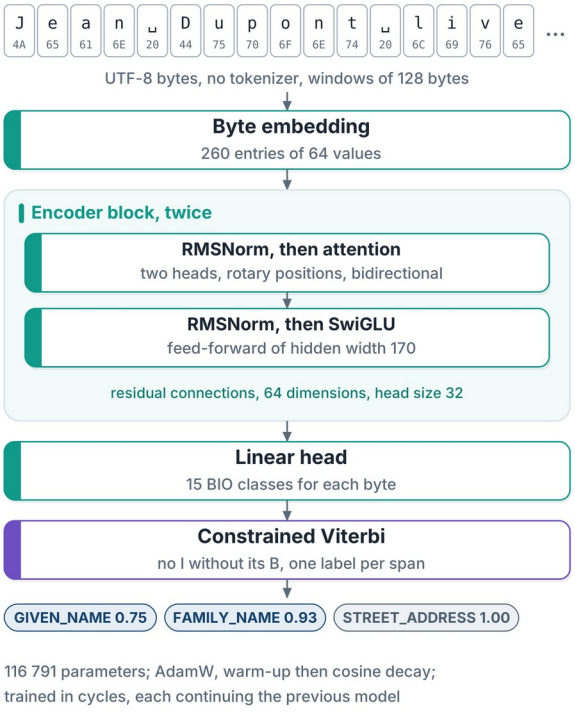


Figure 4. The span model of the engine: byte embedding of 260 entries of 64 values, two encoder blocks, a linear head of 15 BIO classes and a constrained Viterbi decoder, with the outputs of the current model on the sentence "Jean Dupont lives at 12 rue de la Paix".

The model is trained with AdamW [23] on windows of 32 to 128 bytes. The current model continues the previous one on 1 207 499 examples, in 71 minutes on two cores [10]. Both models are compressed into a header of the repository, which Pseudalis compiles into its engines.

3.6 Fusion, propagation and filters

Every candidate carries a priority: the base value of its type (Table 4) plus a bonus of its channel, the largest for a type forced by the document, then for the patterns of the organization, the national identifiers, the user terms and the confident learned spans. Candidates are accepted by priority, then length, then position, as long as they overlap no accepted one. Validated types start at 75 to 100 and names, organizations, places and addresses at 50 to 62, so a learned span never displaces a value whose check digits verify.

Type	Base priority
`EMAIL`	100
`URL`	95
`IBAN, CARD, NIR`	92
`SIRET`	88
`IPV6`	86
`IPV4, MAC`	85
`GEO`	84
`PHONE`	80
`PLATE`	78
`DATE`	75
`GENDER, NATIONALITY`	72
`ID`	70
`ADDRESS`	62
`LOCATION`	60
`TITLE`	58
`ORG`	55
`PERSON`	50

Table 4. Base priorities of the fusion [5].

The span model leads with its spans of confidence 0.5 and more, and the tagger fills the gaps from 0.9. Weaker readings serve as evidence: a capitalized word joined by a conjunction to a person found becomes a person. The names learned in the first reading are then found in their bare mentions, and filters drop the spans made only of generic words, such as `Politique de sauvegarde`, unless a given name, a civility or a label keeps them.

In "Our lawyers Martin and Dupont signed the contract with Veolia Water in 2019.", the detector reports `Martin` as a person, `Dupont` through the conjunction, `Veolia Water` as an organization and `2019` as a date, while the rules alone find only the organization and the date. In "Rose, Violette et Capucine fleurissent au jardin de Mme Claire Martin.", the tagger reads `Capucine` as a given name, and the coverage matrix names such given names used as common nouns as the main remaining false positives [9].

3.7 Document handlers

The same detector and pseudonymizer serve every format. A text keeps its encoding, a table is read cell by cell with its header row typing its columns, and in markup only text and attribute values change. An Office document keeps its layout and its structure: comments, notes, headers, text boxes, formulas, charts and metadata are treated too, and macros and embedded objects go. A PDF is written anew with its pages, its graphics and its tables, each line of text redrawn in place. Encrypted or password-protected files are refused.

4. Format-preserving pseudonymization

4.1 Tags, checks and substitutes

The pseudonymizer replaces each value by a tag `[[KEYWORD:BODY#CHK]]` (Figure 5) [5]. The keyword names the type and selects the codec, the body is the encryption of the value in the form of the value, and the check holds 15 bits of an HMAC-SHA-256 over the keyword and the original value, written as three characters. The same value of the same type always gives the same tag under one key, civilities, sexes and nationalities excepted. The header of the module sums up the design: holding the key "takes the place of the correspondence table other tools write next to the protected document".

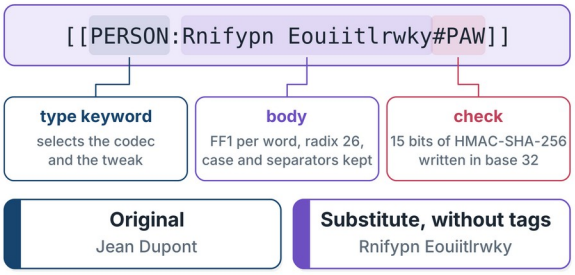


Figure 5. Anatomy of the tag of the name Jean Dupont: the type keyword selects the codec and its tweak, the body encrypts each word with FF1 in radix 26 while keeping case and separators, and the check holds 15 bits of HMAC-SHA-256 in base 32; without tags, the body alone is written as the substitute.

Restoration accepts a tag only when its check matches. A tag altered by one letter stays in place and is counted as rejected, and a random string passes the check about once in 32 768 tries. Without tags, the body alone is written; it carries no check, so restoring it needs the record of its tag.

4.2 FF1 over AES-256

The encryption is FF1 of NIST SP 800-38G [6], written against the draft of its first revision of February 2025 [7]. FF1 is a Feistel network of ten rounds built on AES (Figure 6). It encrypts a string of digits or letters into a string of the same length and the same alphabet, which is how a card number stays a card number. The code enforces the minimum domain of one million values that the draft makes mandatory, and it implements neither FF3 nor FF3-1, which the draft removes. Its tests check the nine NIST samples of FF1, AES, SHA-256, HMAC [24] and HKDF.

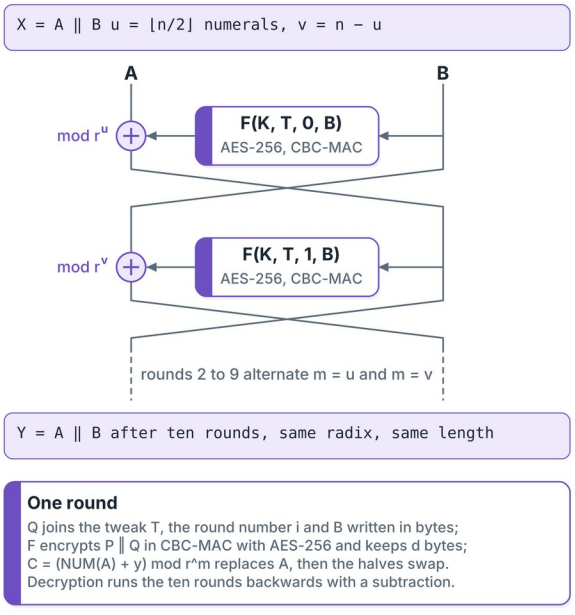


Figure 6. The FF1 Feistel ladder: halves A and B, the round function F built on AES-256 in CBC-MAC mode, the modular addition modulo radix^u or radix^v, and ten rounds that keep the radix and the length.

4.3 Keys and contexts

A master key of 256 bits, drawn at random or derived from a passphrase, is expanded with HKDF into separate keys for words, digits, codes, IP addresses, checks and dates. A context enters this derivation: another context gives unrelated codes, which separates the domains in which values can be linked. Dates move into the past by one secret number of weeks per key and context, between 1 and 260, so their order, their intervals and their weekdays survive.

4.4 Codecs and examples

Table 5 shows the codecs at work under a demonstration key. Names, organizations, places and addresses go through a word codec that keeps case and separators, and encrypts each word into letters whose number follows the length of the word. A person keeps no word of its name, while organizations keep their legal forms, places words such as `Saint` and addresses their street types. Numbers with check digits receive valid new check digits, a card keeps its first six digits, an IBAN its country, and a phone number its country code and first digit. IP addresses use Crypto-PAn [25], which keeps the prefixes that two addresses share, and a MAC address keeps its vendor prefix. A value that its codec refuses, such as a card that fails its check, takes an opaque code that ends with `!`.

Kind	Original	Substitute
person	`Jean Dupont`	`Rnifypn Eouiitlrwky`
organization	`Atelier Dupont et Fils SARL`	`Xkekwmufwmcj Eouiitlrwky et Wldwdst SARL`
place	`Saint-Malo`	`Saint-Ntaahgp`
e-mail	`jean.dupont@example.com`	`rnifypn.eouiitlrwky@pspbshklkspj.com`
phone	`06 12 34 56 78`	`06 07 09 00 95`
card	`4970 1012 3456 7893`	`4970 1065 4873 4584`
IBAN	`GB29 NWBK 6016 1331 9268 19`	`GB45 NWBK 6152 7186 9288 16`
social security number	`1 85 03 75 123 456 41`	`3 88 14 42 078 158 84`
date	`12/03/2019`	`28/08/2018`
date	`3 mars 1985`	`19 août 1984`
IPv4	`192.168.12.45`	`2.176.115.162`
MAC	`00:1A:2B:3C:4D:5E`	`00:1A:2B:38:8D:86`

Table 5. Codecs on invented values under the demonstration key of the module [5].

4.5 The rule applied to each kind of data

Table 6 summarizes the rule applied to each kind of data. Format preservation is syntactic: a protected card or IBAN passes its check, while a protected social security number may carry an implausible month, which shows that a value was replaced. Civilities, sexes and nationalities receive a random element mixed into their encryption and are padded to a common length, so `M.` and `Madame` look alike and two tags of `Madame` differ.

Kind of data	Rule	Kept readable
names, organizations, places, addresses	FF1 radix 26 per word	case, separators, word lengths, legal forms, street types
e-mail, web address	word codec	separators, top-level domain
phone	FF1 radix 10	country code, first national digit
card	FF1 radix 10, Luhn recomputed	first 6 digits
IBAN	FF1 radix 10 and 26, check recomputed	country, short bank codes
NIR	FF1 radix 10, key recomputed	Corsican letters
SIREN, SIRET	FF1 radix 10, Luhn recomputed	NIC
identifier	FF1 radix 10, 36 or 62	separators, length
UUID, MAC	FF1 radix 16	hyphens, vendor prefix
plate, GPS	FF1 radix 10	layout, signs
IPv4, IPv6	Crypto-PAn	shared prefixes
date	secret shift	order, intervals, weekdays
civility, sex, nationality	FF1 with random nonce	length bucket
value refused by its codec	opaque FF1 radix 26	equality

Table 6. Rule applied to each kind of data [5].

5. Synthetic dataset

Annotated personal data cannot be collected without collecting personal data, which the recognizers exist to avoid. The recognizers therefore learn from synthetic texts written by the generator `examples/slm_tools/build_pii_dataset.py` of the repository [26], whose values come from lexicons or are invented, so that no line describes a real person. Its resources, lists of names, places, organizations, templates and sentences in each language, stay outside the repository, and the generator applies further variations that this report leaves aside. Figure 7 shows the chain from these resources to the engine.

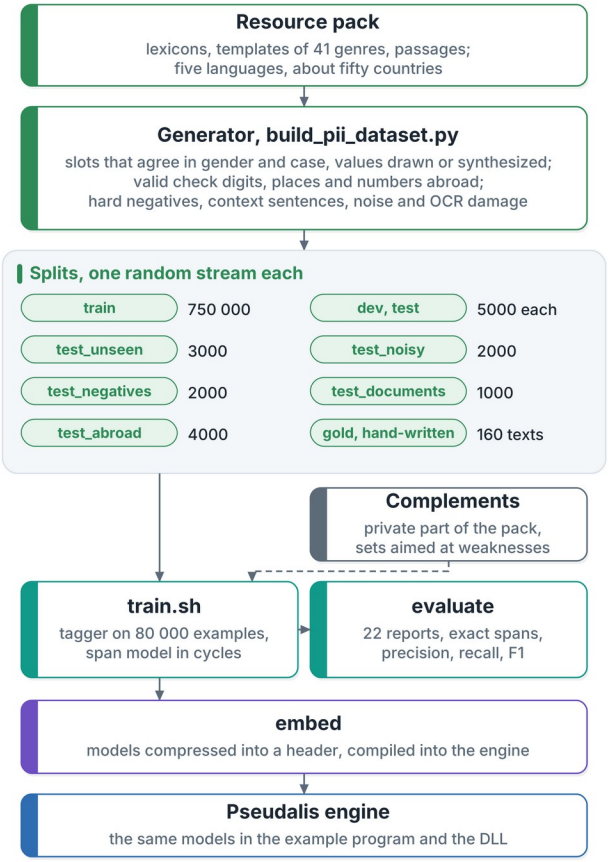


Figure 7. From the resource pack to the engine: the generator, its splits with their sizes, the complementary sets kept outside the repository, training, evaluation, and the embedding of the models into the engine.

The templates cover about forty kinds of documents, from forms, letters and e-mails to medical, banking, legal and technical texts. Every number carries a valid check, and hard negatives place capitalized words that are not personal data, such as laws, holidays, regions and common nouns that look like given names, in every kind of document. Table 7 gives the splits. A share of the lexicons is held out for `test_unseen`, and 160 hand-written texts, 32 per language, serve evaluation only.

Split	Examples	Spans	Content
`train`	750 000	4 623 319	seen values
`test`	5000	32 118	all values
`test_unseen`	3000	19 173	held-out values
`test_noisy`	2000	12 619	six times the noise
`test_negatives`	2000	853	hard negatives
`test_documents`	1000	29 877	3 to 6 templates per text
`test_abroad`	4000	25 180	places abroad only
hand-written texts	160	2299	written by hand

Table 7. Splits of the synthetic dataset and the hand-written texts [10, 26].

6. Performance

6.1 Evaluation protocol

The example program of the module scores in one run the rules alone, each recognizer alone and the whole detector. The references take the type their value is replaced as, given and family names becoming `PERSON` for instance, and a prediction counts only with the right type and exactly the right first and last bytes. The figures below concern the whole detector, every category and exact spans.

6.2 Results of the current models

On `test`, the detector reaches a precision of 0.955, a recall of 0.935 and an F1 of 0.945 [10]. Figure 8 compares each split with the first models, which were trained on French and English only. The languages range from 0.913 for Spanish to 0.970 for French, the hand-written texts from 0.889 to 0.899, and noise costs recall, which stays just under 0.90 on `test_noisy` (Table 8).

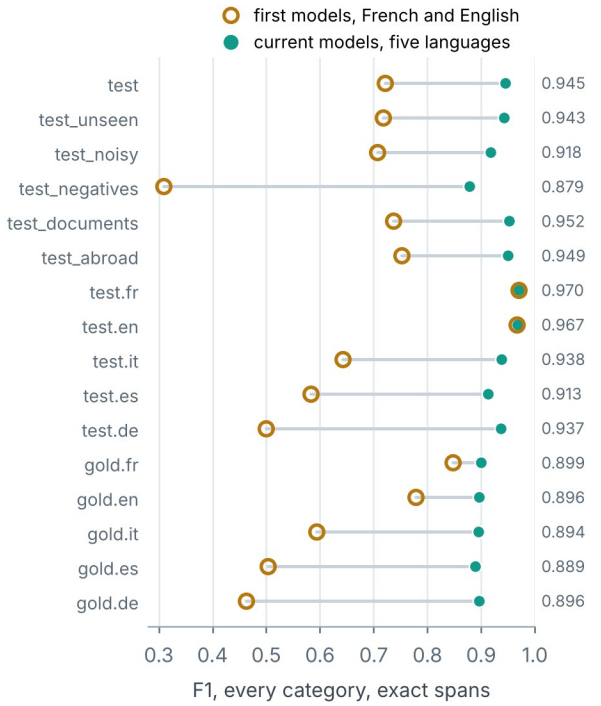


Figure 8. F1 per split, every category and exact spans, for the first models, trained on the French and English corpus, and for the current five-language models.

Split	Precision	Recall	F1
`test`	0.955	0.935	0.945
`test_unseen`	0.951	0.935	0.943
`test_noisy`	0.941	0.897	0.918
`test_negatives`	0.804	0.968	0.879
`test_documents`	0.961	0.943	0.952
`test_abroad`	0.960	0.939	0.949
French	0.975	0.965	0.970
English	0.971	0.964	0.967
Italian	0.950	0.926	0.938
Spanish	0.928	0.898	0.913
German	0.949	0.926	0.937
hand-written, five languages	0.887 to 0.920	0.879 to 0.905	0.889 to 0.899

Table 8. Whole detector with the current models on the public splits [10].

Figure 9 gives the F1 of each category on `test`. Nine categories reach 1.000, among them e-mail addresses, cards, social security numbers and IP addresses, while organizations (0.894), identifiers (0.896) and nationalities (0.904) close the ranking. Persons, organizations, places and addresses account for most of the errors.

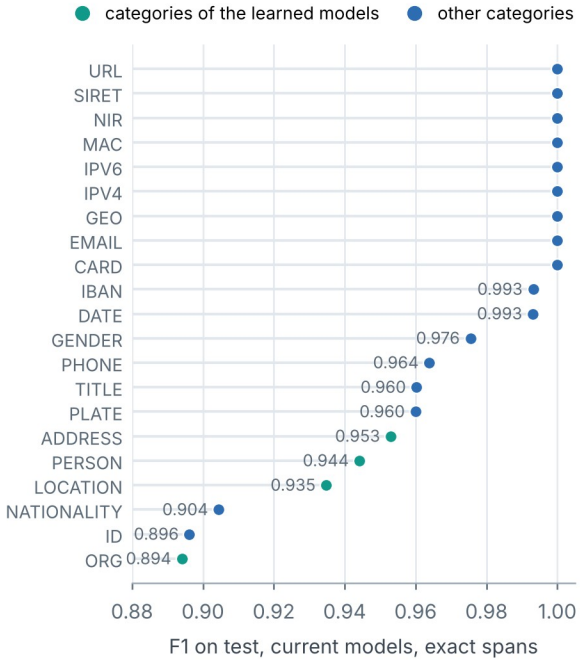


Figure 9. F1 per category on test with the current models, exact spans, for the categories of the learned models and the other categories.

6.3 What each component brings

The rules alone reach an F1 of 0.840, with a high precision and a lower recall, so the learned recognizers mainly add recall (Table 9). The span model alone does much better than the tagger, and it takes about 85% of the detection time.

System	Labels	F1
rules alone	every category	0.840
tagger alone	fine labels	0.749
span model alone	fine labels	0.903
span model alone	learned categories	0.920
whole detector	learned categories	0.935
whole detector	every category	0.945

Table 9. Components alone and combined on `test` [10].

6.4 From the first models to the current ones

The first models reached an F1 of 0.721 on `test` and 0.499 on German, and produced 2797 false positives on the hard negatives [10]. The release that brought the five languages raised `test` to 0.943 and cut those false positives to 256. Later releases added rules for coordinated names, imposed terms and technical documents, and the false positives went on to 201. The current models were retrained without any change of rule and improved most splits slightly.

6.5 Real documents

On a private sample of real documents, the current models reach a precision of 0.872 and a recall of 0.917 [9]. About one personal value in twelve therefore stays in clear in such documents, which is why the README and the coverage matrix recommend reading a protected document before sharing it [1, 9].

6.6 Speed

On two cores, the whole detector reads the synthetic test split, about one million characters, in 23.4 to 23.7 s, about 41 000 characters per second and slightly faster than the previous models, while the rules alone take 1.5 to 1.7 s [10]. Table 10 gives the time of the service for whole documents, request included, on its Linux build. The Word file goes faster than the same text in a plain file, which the runs do not explain.

Document	Size	Time
text, 5000 words, 720 values, `.txt`	27.5 KB	2.5 to 2.6 s
same text, `.docx`	27.5 KB of text	1.4 to 1.5 s
CSV, 1000 rows, 5027 values	95 KB	5.8 to 5.9 s

Table 10. Time of the service for synthetic documents, three runs each.

7. GDPR coverage

7.1 Definitions and the status of the output

Supporting compliance with the GDPR is a stated goal of Pseudalis [1, 9]. This section reads the texts as they stood on 8 November 2026; the qualification of a processing, its legal basis and the need for an impact assessment remain with the controller, under the supervisory authorities and the courts.

The Regulation defines personal data as "any information relating to an identified or identifiable natural person", and pseudonymisation as a processing after which the data "can no longer be attributed to a specific data subject without the use of additional information", kept separately and protected [8]. Recital 26 keeps pseudonymised data within the Regulation and leaves anonymous information outside it. The output of Pseudalis is a pseudonymization: the key restores the data, which remain personal data for whoever holds it [1]. The additional information is the master key, with the vault and the record of the substitutes, and a file written without tags remains pseudonymized for the holder of the key, who can decrypt a substitute or confirm a guess. The example program also offers an irreversible mode under a random key that is never stored; whether its output is anonymous then depends on what the remaining text reveals. Organizations fall outside the Regulation, yet Pseudalis replaces them by default, since the name of an employer, a fund or a court often suffices to identify a person [9].

7.2 Categories covered, partly covered and not covered

Table 11 condenses the coverage matrix [9]. Direct identifiers, contact data, identifiers with a check, online and vehicle identifiers and organizations are covered, and identifiers recognizable only through their label are partly covered. Health details, beliefs or criminal facts written in whole sentences stay readable, since a tag would destroy the meaning of the sentence; the matrix notes that a classifier to flag such sentences for review remains to be built.

Category	State	Means
names, nicknames, logins	covered	rules, labels, propagation, models
addresses, cities, postal codes with city	covered	rules, labels, models
phone, e-mail, dates	covered	scanners, dates shifted
civility, sex, nationality	covered	rules, labels
identifiers with a check (NIR, IBAN, cards, about 40 countries)	covered	validated scanners
IP, MAC, UUID, IMEI, plates, VIN, GPS	covered	validated scanners
organizations, named public bodies	covered	rules, tables, models, imposed terms
identity card, driving license, residence permit	partly	label only
customer, case, contract, employee numbers	partly	label, cue or pattern
postal code alone, region, country	not covered	left readable
age, amounts linked to a person	not covered	left readable
health, beliefs, criminal facts	not covered	left readable
session tokens, API keys	not covered	user patterns only
text inside pictures	not covered	no OCR, picture removed

Table 11. Coverage of the categories of personal data by Pseudalis [9].

7.3 Articles 25 and 32

Article 25 asks for measures "such as pseudonymisation" from the design of a processing, and limits by default the data processed, their storage and their accessibility [8]. Article 32 lists "the pseudonymisation and encryption of personal data", the confidentiality and integrity of systems, the restoration of availability and the regular testing of the measures. Pseudalis provides several of them in its code: processing on the computer behind the loopback interface, FF1 tags under AES-256, an authenticated encrypted vault with limits in time and space and a function to forget, a key protected by Windows and exported only under a passphrase, signed updates, and a command that evaluates the detector on an organization's own samples. The

export of the key is the backup of an installation, so a lost key is a loss of availability.

7.4 EDPB guidance as of November 2026

The EDPB adopted its Guidelines 01/2025 on Pseudonymisation on 16 January 2025 as a version for public consultation [27, 28, 29]. On 8 November 2026 no final version had been published, and in December 2025 the EDPB stated that it was reassessing the work in the light of the case law [27, 30]. The draft holds that pseudonymised data remain personal data, and describes pseudonymisation within a pseudonymisation domain, as a measure under Articles 25 and 32 [28, 29]. For an installation of Pseudalis, that domain is everything outside the computer, since every user of the computer lies inside its circle of trust [1].

The Guidelines 02/2026 on Anonymisation, adopted on 7 July 2026 for consultation until 30 October 2026, remain a draft [31, 32, 33]. They test whether a person can be singled out, linked or inferred, and require a closer analysis, from the perspective of each relevant party, when data fail a test. The equal codes and the shifted dates of one installation bear on linkage, and readable health details and amounts on inference, so the output of the irreversible mode needs a documented assessment before anyone treats it as anonymous.

7.5 What the courts say

Court rulings point the same way. Rulings of the Court of Justice of the European Union have held that an IP address or a vehicle identification number is personal data for whoever has the means to link it to a person, which supports the protection of these identifiers, and that the same pseudonymised data can be personal data for the holder of the key and not for a recipient who has no reasonable means to re-identify the persons [34, 35, 36]. In France, the highest administrative court held in 2026 that pseudonymised health databases keeping ages, sexes, pathologies and a patient identifier remained personal data [37], and the CNIL likewise treats pseudonymised data as personal data [38].

Every protected file therefore remains personal data for the holder of the key. A recipient without the key may hold data that are not personal for it, which depends on what the readable remainder of each document reveals to that recipient. The equal codes of one installation act as persistent pseudonymous identifiers, the configuration that the French ruling held to be personal data when combined with readable health details.

7.6 Limits

The circle of trust is the whole computer: any user of the computer can restore files, read the records and export the key, and a file written to the drive lies in clear until its processing [1]. Detection is probabilistic in the learned categories, with a recall of 0.917 on real documents, and a reference that no word announces needs a pattern of the organization [9]. Values hidden in encoded text stay in clear, and both the README and the coverage matrix recommend reading a protected document before sharing it [1, 9]. Codes keep the case and the length of their words, and grammatical agreement can reveal the sex of a person whose civility was replaced [9]. One known pair of an original date and its shifted form reveals the shift of a whole installation, and substitutes of identifiers with valid checks can coincide with real identifiers of other people. Pseudalis is therefore one technical measure within the compliance of a controller, which keeps the legal qualification, the impact assessment and the review before sharing.

8. Code availability

The implementation is part of the public Dlib-Transformer-extensions repository on GitHub, under the Boost Software License 1.0, at https://github.com/Cydral/Dlib-Transformer-extensions [3]. The source code and README of Pseudalis lie at https://github.com/Cydral/Dlib-Transformer-extensions/tree/main/dlib/dlib/external/pseudalis [1], and the installer with its signed feed at https://github.com/Cydral/Dlib-Transformer-extensions/tree/main/dist/pseudalis [2]. The module `dlib/privacy` lies at https://github.com/Cydral/Dlib-Transformer-extensions/tree/main/dlib/dlib/privacy [5], and the dataset generator at https://github.com/Cydral/Dlib-Transformer-extensions/blob/main/examples/slm_tools/build_pii_dataset.py [26]. The trained models are compiled into the engine from the header `examples/slm_document_anonymizer_ex.h`, while the resources of the dataset and the complementary sets stay outside the repository.

9. Conclusion

Pseudalis Community keeps every secret of reversibility on one computer: the master key, the vault of the originals and the record of the substitutes. With tags, a protected document carries its own way back, under the key. Without tags, the way back lies in the record, so the retention settings decide how long an original stays recoverable, and choosing files without tags for external sharing is also choosing that duration.

The measurements place the remaining work in the learned categories, organizations first, since the validated scanners hold the structured categories at or near an F1 of 1.000. The span model of 116 791 parameters brings most of the recall in these categories, and its capacity is the limit named since the corpus grew to five languages [10]; gains on organizations and on real documents call for a larger model or for more evidence drawn from the whole document. The codecs keep syntax, check digits and equality, which keeps protected documents usable for their recipients and also linkable, so the status of a protected document for a recipient without the key depends on what its text leaves readable, case by case.

References

[1] A. Pierrain. Pseudalis Community, source code and README. Dlib-Transformer-extensions repository on GitHub,

https://github.com/Cydral/Dlib-Transformer-extensions/tree/main/dlib/dlib/external/pseudalis, 2026.

[2] A. Pierrain. Pseudalis Community, installer and signed update feed. Dlib-Transformer-extensions repository on GitHub, https://github.com/Cydral/Dlib-Transformer-extensions/tree/main/dist/pseudalis, 2026.

[3] Dlib-Transformer-extensions project. Dlib-Transformer-extensions, transformer extensions of the dlib C++ library. GitHub repository, https://github.com/Cydral/Dlib-Transformer-extensions, 2026.

[4] D. E. King. Dlib-ml: a machine learning toolkit. Journal of Machine Learning Research, vol. 10, pp. 1755-1758, 2009.

[5] Dlib-Transformer-extensions project. The dlib/privacy module for personal-data detection and pseudonymization. Dlib-Transformer-extensions repository on GitHub, https://github.com/Cydral/Dlib-Transformer-extensions/tree/main/dlib/dlib/privacy, 2026.

[6] M. Dworkin. Recommendation for block cipher modes of operation: methods for format-preserving encryption. NIST Special Publication 800-38G, DOI 10.6028/NIST.SP.800-38G, 2016.

[7] M. Dworkin and N. Mouha. Recommendation for block cipher modes of operation: methods for format-preserving encryption, second public draft. NIST Special Publication 800-38G Rev. 1 (draft), DOI 10.6028/NIST.SP.800-38Gr1.2pd, 2025.

[8] European Parliament and Council of the European Union. Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation). Official Journal of the European Union, L 119, pp. 1-88, CELEX 32016R0679, 2016.

[9] Pseudalis project. Personal data, strict definition and coverage of dlib/privacy (coverage matrix, in French), version of 8 November 2026. Internal document, 2026.

[10] Pseudalis project. Evaluation and training reports of the Pseudalis detectors, from the first models to the current ones, with the README of the dataset pack. Dataset pack kept outside the repository, 2026.

[11] National Institute of Standards and Technology. Advanced Encryption Standard (AES). FIPS 197, updated 2023, DOI 10.6028/NIST.FIPS.197-upd1, 2001.

[12] National Institute of Standards and Technology. The keyed-hash message authentication code (HMAC). FIPS 198-1, DOI 10.6028/NIST.FIPS.198-1, 2008.

[13] National Institute of Standards and Technology. Secure Hash Standard (SHS). FIPS 180-4, DOI 10.6028/NIST.FIPS.180-4, 2015.

[14] H. Krawczyk and P. Eronen. HMAC-based extract-and-expand key derivation function (HKDF). IETF RFC 5869, 2010.

[15] OWASP Foundation. Password storage cheat sheet. OWASP Cheat Sheet Series, consulted on 8 November 2026, 2026.

[16] I. Tsochantaridis, T. Joachims, T. Hofmann and Y. Altun. Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research, vol. 6, pp. 1453-1484, 2005.

[17] L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-based learning. Third Workshop on Very Large Corpora, arXiv:cmp-lg/9505040, 1995.

[18] A. Vaswani, N. Shazeer, N. Parmar et al. Attention is all you need. Advances in Neural Information Processing Systems 30, arXiv:1706.03762, 2017.

[19] B. Zhang and R. Sennrich. Root mean square layer normalization. Advances in Neural Information Processing Systems 32, arXiv:1910.07467, 2019.

[20] J. Ainslie, J. Lee-Thorp, M. de Jong et al. GQA: training generalized multi-query transformer models from multi-head checkpoints. Proceedings of EMNLP 2023, arXiv:2305.13245, 2023.

[21] J. Su, M. Ahmed, Y. Lu et al. RoFormer: enhanced transformer with rotary position embedding. Neurocomputing, vol. 568, 127063, 2024.

[22] N. Shazeer. GLU variants improve transformer. arXiv:2002.05202, 2020.

[23] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. International Conference on Learning Representations, arXiv:1711.05101, 2019.

[24] M. Nystrom. Identifiers and test vectors for HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512. IETF RFC 4231, 2005.

[25] J. Xu, J. Fan, M. H. Ammar and S. B. Moon. Prefix-preserving IP address anonymization: measurement-based security evaluation and a new cryptography-based scheme. 10th IEEE International Conference on Network Protocols (ICNP), Paris, 2002.

[26] Dlib-Transformer-extensions project. Synthetic dataset generator build_pii_dataset.py. Dlib-Transformer-extensions repository on GitHub, https://github.com/Cydral/Dlib-Transformer-extensions/blob/main/examples/slm_tools/build_pii_dataset.py, 2026.

[27] European Data Protection Board. Guidelines 01/2025 on Pseudonymisation, version 1 for public consultation. EDPB, 2025.

[28] European Data Protection Board. EDPB adopts pseudonymisation guidelines and paves the way to improve cooperation with competition authorities. EDPB news, 17 January 2025, 2025.

[29] Hunton Andrews Kurth LLP. EDPB advises on pseudonymisation for GDPR compliance. Privacy Laws and Business, 5 June 2025, 2025.

[30] European Data Protection Board. Discussion paper for the stakeholder event of 12 December 2025 on anonymisation and pseudonymisation. EDPB, December 2025, 2025.

[31] European Data Protection Board. Guidelines 02/2026 on Anonymisation, version 1 for public consultation. EDPB, 2026.

[32] D. Cooper, K. Van Quathem and A. Bertrand. EDPB publishes draft guidelines on anonymisation. Covington and Burling, Inside Privacy, 4 August 2026, 2026.

[33] Sidley Austin LLP. EDPB publishes draft guidelines on anonymisation. Data Matters, 27 July 2026, 2026.

[34] Court of Justice of the European Union. Judgment of 19 October 2016, Patrick Breyer v Bundesrepublik Deutschland. Case C-582/14, 2016.

[35] Court of Justice of the European Union. Judgment of 9 November 2023, Gesamtverband Autoteile-Handel e.V. v Scania CV AB. Case C-319/22, 2023.

[36] Court of Justice of the European Union. Judgment of 4 September 2025, European Data Protection Supervisor v Single Resolution Board. Case C-413/23 P, Official Journal of the European Union C/2025/5551, 2025.

[37] Herbert Smith Freehills Kramer. French Supreme Court rules on pseudonymisation vs anonymisation. HSF Kramer data notes, 2026.

[38] Commission nationale de l'informatique et des libertés. Identifier les données personnelles. CNIL, 27 January 2020, 2020.