Subject: 31261 Internetworking Project

Autumn 2026

FINAL REPORT

Privacy-preserving self-hosted de-identification workflow for sensitive legal documents

---

Project supervisor	Dr Yi Zhang
Legal supervisor	Dr Susanne Lloyd-Jones
Technical supervisor	Runsong Jia
Owners	Alex Harb, 24715841, Govemance / Legal Ajay Yuvaraj, 24934301, Governance / Legal
Contributors	Ishaan Pathak, 24963593, Architecture / Evaluation Lori Basmajian, 24563090, Architecture / Evaluation Nicholas Hatzidimitriou, 25471398, Implementation / Development Adrian Martinez, 25494636, Implementation / Development
Project / working title	Privacy-preserving self-hosted de-identification workflow for sensitive legal documents

Table of Contents

Subject: 31261 Internetworking Project

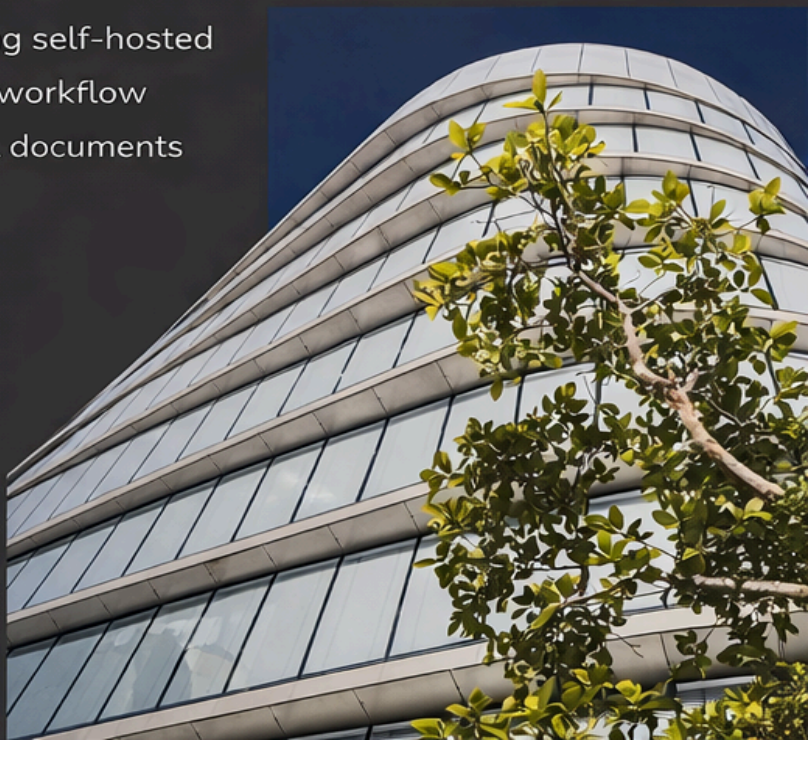
Autumn 2026

GOVERNANCE REPORT

Privacy-preserving self-hosted de-identification workflow for sensitive legal documents

---

Project supervisor	Dr Yi Zhang
Legal supervisor	Dr Susanne Lloyd-Jones
Technical supervisor	Runsong Jia
Owners	Alex Harb, 24715841, Govemance / Legal Ajay Yuvaraj, 24934301, Governance / Legal
Contributors	Ishaan Pathak, 24963593, Architecture / Evaluation Lori Basmajian, 24563090, Architecture / Evaluation Nicholas Hatzidimitriou, 25471398, Implementation / Development Adrian Martinez, 25494636, Implementation / Development
Project / working title	Privacy-preserving self-hosted de-identification workflow for sensitive legal documents



Executive Summary

This report presents a governance-led, self-hosted de-identification workflow for sensitive legal documents in AI-assisted environments, supported by a prototype implementation. Its primary contribution is not the software alone, but a structured method for producing risk-reduced “safe copies” of legal documents for controlled downstream use, together with evidence artefacts that support review, traceability, and accountability.

The project responds to a growing operational problem in legal practice. It is a reality that confidential and privacy-sensitive matter documents are increasingly used in AI-enabled workflows without consistent safeguards for confidentiality, verification, metadata handling, or auditable review. In response, the workflow is designed for deployment in a firm-controlled environment and applies policy-driven transformation, primarily through pseudonymisation rather than blind redaction. The prototype produces two main outputs: a pseudonymised “safe copy” for downstream use, and an evidence pack containing artefacts such as a manifest, audit log, and court- or governance-facing report.

A central premise of the report is that de-identification is a risk-management process, not a binary status. Accordingly, the project does not claim that transformed outputs are automatically anonymous, outside privacy regulation, or free from duties of confidence, privilege, or professional responsibility. The aim is to enable legal documents to be transformed into risk-reduced outputs that lower direct exposure while preserving enough coherence for low-risk downstream tasks, subject to review and controlled use.

The report argues that this problem cannot be solved by tool choice alone. Existing redaction, de-identification, and enterprise AI solutions address parts of the problem, but often do not provide an SME-appropriate, auditable, legal-document workflow. This project therefore combines minimisation, trust-boundary separation, metadata awareness, reviewer sign-off, and structured evidence generation into a single governance model.

The prototype is intentionally limited. It does not claim compliance, certification, or legal advice; it does not guarantee anonymisation; and it does not replace legal judgment, privilege review, source verification, or broader organisational governance. Its value lies in showing how sensitive legal-document handling can be made more controlled, more reviewable, and more defensible in AI-assisted environments.

0. Document control

0.1 Purpose of this document

This document is a governance and methodology draft for a techno-legal prototype that produces “LLM-ready pseudonymised safe copies” of sensitive legal documents, with an audit trail and an exportable court- or governance-facing report. The prototype is intended to assist legal professionals in preparing documents for controlled downstream processing while reducing disclosure risk and improving traceability. This document is written as a student project deliverable for the 31261 Internetworking Project and is intended to support the project’s design, implementation, and evaluation streams. It does not claim organisational compliance, certification, or legal advice.

0.2 What this document is, and is not

This document is a risk-based governance and methodology framework for the designed prototype, informed by and structured with reference to relevant ISO standards, Australian regulatory guidance, and court anonymisation and document-handling practices (International Organization for Standardization, 2024; Office of the Australian Information Commissioner [OAIC], 2018; O'Keefe et al., 2017). It is intended to provide a practical baseline of requirements, assumptions, controls, and expectations for evidence for the build and associated evaluation streams.

This document is not a claim that a deploying law firm is compliant with ISO standards, the Essential Eight, OAIC guidance, or court practice notes merely by following this framework. It is not a full implementation of an organisational information security management system, quality management system, or privacy information management system (Standards Australia, 2022). It is not a guarantee of anonymisation, and it does not assert that any output falls outside “personal information” or equivalent legal concepts in all contexts (International Organization for Standardization, 2024; OAIC, 2018). It does not replace legal, privacy, security, or professional advice, nor does it remove the need for human verification, organisational controls, or context-specific risk assessment in any real deployment (International Organization for Standardization, 2024; O'Keefe et al., 2017).

0.3 Audience and intended use

The audience is any reader seeking to understand how the prototype could be deployed or extended in a real-world legal practice environment. For that reason, the document

explains not only what the prototype is expected to do but also the limits of what it does not claim to do.

The document is intended to be used as a governance baseline for the prototype, a requirements reference for the build stream, an evaluation reference for testing, evidence collection, and reporting, a structured explanation of assumptions, limitations, and residual risks, and a foundation for the project's final report, evidence pack, and related deliverables. It is not intended to operate as a standalone operational policy for a law firm, nor as a substitute for a formal Privacy Impact Assessment, legal advice, security certification process, or production deployment approval (International Organization for Standardization, 2023, 2024).

1. Definitions and glossary

1.1 Key terms and abbreviations

Unless otherwise indicated, the following terms are drawn from or adapted from OAIC de-identification guidance, ISO/IEC 29100, ISO/IEC 20889, and the CSIRO De-Identification Decision-Making Framework; project-specific terms are identified as such (International Organization for Standardization, 2018, 2024; OAIC, 2018; O'Keefe et al., 2017).

- **Anonymisation:** Processing by which PII is irreversibly altered so that the individual can no longer be identified, directly or indirectly, whether by the controller alone or in collaboration with another party (International Organization for Standardization, 2024).

- **Approving authority:** The role authorised to approve higher-risk downstream use, external disclosure, or other actions requiring acceptance of residual governance risk.

- **Artefact:** Digital artefacts (US: artifacts) are collections of related data generated through the regular use of digital devices or systems, whether intentionally or unintentionally (Nguyen, 2023). This may result in a file, record, or output generated by the workflow, such as a safe copy, manifest, audit log, governance-facing report, or integrity file.

- **Audit trail/audit log:** A record of what was processed, by whom, when, and under which settings or policy.

- **Data environment/access environment:** The setting in which data is accessed, used, or disclosed, including the people, other data, governance processes, and infrastructure surrounding that use.

- **Data situation:** The relationship between the data and the environment it moves through, including agency, governance, infrastructure, and other data.

- **De-identification:** A process of removing or altering information to reduce the likelihood that an individual is identifiable, directly or indirectly, in context.

- **Deterministic pseudonymisation:** Consistent replacement of the same entity with the same substitute within a defined scope, such as per document, job, or matter.

- **Direct identifier:** An attribute that can identify a person on its own, such as a name, passport number, Medicare number, or precise address.

- **Disclosure risk:** The risk that data release enables identity disclosure, attribute disclosure, or inference about a person.

- **Evidence pack:** The structured export bundle containing the safe copy and supporting artefacts, such as the manifest, audit log, and governance-facing report, used to support review, traceability, and reconstruction of the processing run.

- **Human verification/reviewer sign-off:** A procedural control requiring a reviewer to assess output quality, residual risk, and suitability for the intended use before reliance-critical use or external disclosure where required.

- **Indirect identifier/quasi-identifier:** An attribute that may not identify a person alone, but can do so when combined with other information, such as age, suburb, job title, or event date.

- **Linkability:** The ability to link records relating to the same individual. Pseudonymisation typically preserves linkability; anonymisation aims to destroy it.

- **Mapping register/substitution register:** A confidential record linking original identifiers to substituted labels or pseudonyms. If retained, it effectively serves as a re-identification key.

- **Metadata:** Data about the document or file, such as properties, comments, tracked changes, embedded text, or hidden fields, which may itself contain identifying information.

- **Metadata stripping:** Removal or transformation of supported forms of document metadata that may reveal identities or hidden content.

- **Motivated intruder:** A plausible person, internal or external, who attempts re-identification using reasonable effort, available information, and ordinary means, without assuming specialist powers or unlawful access.

- **Personal information/PII:** In this report, a shorthand term for information about an identified individual, or an individual who is reasonably identifiable, directly or indirectly. The exact legal formulation varies by instrument and jurisdiction.

- **PII controller:** Entity that determines purposes and means of processing PII; in this project, the deploying law firm.

- **PII principal (data subject):** Natural person to whom the information relates.

- **PII processor:** Entity that processes PII on behalf of the controller and under instruction; in this project, the de-identification service typically operates as a processor-style function within the firm's controlled environment.

- **Pseudonymisation:** Processing that replaces identifiers with a pseudonym while preserving some ability to link records. It does not, by itself, eliminate the risk of re-identification (International Organization for Standardization, 2024).

- **Reasonably identifiable:** A contextual threshold asking whether there is a reasonable likelihood that an individual could be identified, having regard to the nature and amount of information, who holds it, what other information is available, the practicability of identification, and likely motivations to attempt it (OAIC, 2018; O'Keefe et al., 2017).

- **Recipient environment:** The environment into which a safe copy or other output is disclosed, accessed, or used, including its people, systems, governance controls, and available auxiliary information.

- **Re-identification:** The process of identifying an individual again from de-identified or pseudonymised data, whether from the data itself or in combination with other available information.

- **Reliance-critical use:** A use case where output may materially affect advice, filing, rights, obligations, or court-facing work product.

- **Safe copy:** A project-specific term for the pseudonymised, risk-reduced output prepared for controlled downstream use. A safe copy is not assumed to be anonymous or unrestricted and may remain sensitive depending on context.

- **Third party:** A person or entity outside the controller-processor relationship that receives data, creating a separate access or disclosure environment.

- **Trust boundary:** The point at which data moves between trust zones, users, or systems with different access, security, or governance assumptions.

- **Trust zone:** A part of the workflow, such as input, processing, output, or administration, that is treated as having its own access, handling, and control assumptions.

- **Trust-zone separation:** Separation of input, processing, output, and administrative functions so that each is subject to access and handling controls appropriate to its risk.

1.2 Data roles, controller, processor, third party

This project uses controller/processor role language to clarify responsibility for handling personal information within the prototype's intended deployment model. In the default deployment model, the deploying law firm is treated as the PII controller because it determines the purposes and means of processing client and matter data (International Organization for Standardization, 2024). The PII principal, or data subject, is the natural person to whom the information relates, such as a client, witness, staff member, or other third party mentioned in legal documents.

The de-identification prototype is treated as PII processor-style processing within the firm's jurisdiction. In other words, the prototype processes information on behalf of and under the direction of the deploying firm, rather than as an independent decision-maker.

Any entity outside that controller-processor relationship that receives data is treated as a third party for governance purposes. In this project, that may include an external enterprise LLM provider, external eDiscovery platform, external translation service, or any other recipient environment not operating solely under the firm's internal control. Once data is disclosed into that external environment, the risk context changes and separate safeguards may be required. This role mapping is used throughout the report to distinguish processing types and disclosure into separate recipient environments (International Organization for Standardization, 2024).

1.3 Data classes and identifier categories, direct vs indirect identifiers

For the purposes of this project, identifiers are classified into two broad categories: direct and indirect identifiers (International Organization for Standardization, 2018, 2024; O'Keefe et al., 2017).

Direct identifiers are attributes that can identify an individual on their own, such as names, licence numbers, passport numbers, Medicare numbers, or precise residential addresses. These are generally the first targets for transformation or removal.

Indirect identifiers, also described as quasi-identifiers, are attributes that may not identify an individual alone but may do so in combination with other information. Examples include age, suburb, event dates, occupation, workplace, or other relevant details. These

matter because re-identification risk often arises not from one field alone, but from combinations of attributes and surrounding factors.

This project also treats metadata as a separate and important disclosure vector. A document may appear to be redacted or pseudonymised in the visible text while still containing identifying information in tracked changes, comments, document properties, hidden fields, or embedded text layers (International Organization for Standardization, 2024).

1.4 “Safe copy” definition and handling assumption

A safe copy is the project’s term for a pseudonymised, risk-reduced version of sensitive information in a legal document prepared for controlled downstream use. It is intended to preserve sufficient structure and referential integrity to remain useful for tasks such as summarisation, chronology-building, and extraction.

The term is deliberately narrower than “anonymous document.” A safe copy is not assumed to be anonymous, and it is not treated as relieving *Privacy Act* obligations or professional duties, including duties owed to the court (*Legal Profession Uniform Conduct (Barristers) Rules 2015* [NSW]; *Legal Profession Uniform Law Australian Solicitors’ Conduct Rules 2015* [NSW] [The Uniform Law]).

This report also proceeds on the basis that confidentiality risk is not exhausted by privacy regulation alone. Legal documents may contain information received, generated, or held in circumstances giving rise to duties of confidence, including client-confidential, privileged, commercially sensitive, draft, or court-related material.

2. Project overview, problem context, and scope

2.1 What we designed

This project develops a governance-led workflow for producing risk-reduced “safe copies” of sensitive legal documents in a firm-controlled environment, supported by a prototype implementation. The prototype packages existing open-source de-identification components into one possible technical implementation of that workflow, but the project’s primary contribution is the method, control logic, and governance structure rather than the software alone. The prototype is designed primarily for Australian law firms and related legal-service environments, though the workflow may also be applicable to other small-to-medium organisations handling sensitive documents. It accepts uploads through a controlled internal link, pseudonymises the documents through consistent replacement rather than blind redaction, and produces two main outputs: a pseudonymised “safe copy” for controlled downstream use, and an evidence pack containing an audit trail, manifest, and court- or governance-facing report summarising what was changed and under which settings.

The governance model assumes deployment on firm-controlled infrastructure rather than public cloud services. This does not eliminate security, privacy, or compliance duties, but it reduces unnecessary third-party exposure and supports stronger control over data flows, retention, and auditability. The workflow is designed not only to reduce disclosure risk, but also to preserve enough task-relevant structure, role coherence, and factual utility for its intended downstream uses (International Organization for Standardization, 2024; OAIC, 2018; O'Keefe et al., 2017).

2.2 Why it matters: governance

Generative AI is increasingly used for practical legal work, including summarisation, chronology-building, information extraction, and question answering across matter files. These workflows routinely involve documents containing confidential client information, privileged material, and personal information. The governance risk is not simply that AI exists, but that real legal documents may be copied from controlled environments into ad hoc tools and workflows without consistent safeguards for confidentiality, verification, and accountability.

2.3 Why it matters: problem

Official records show that AI use and misuse are already appearing in legal proceedings. The UNSW Centre for the Future of the Legal Profession compiled a multi-jurisdiction dataset of matters in which suspected or confirmed use or misuse of generative AI was recorded in the official record, identifying recurring problems, including incorrect citations, flawed reasoning, and procedurally defective documents (Centre for the Future of the Legal Profession [CFLP], 2025). The CFLP also states that this material is likely only the "tip of the iceberg," because much use is never detected, recorded, or published. Market reporting points in the same direction, suggesting that AI use in Australian legal practice is now widespread (Croft, 2025).

Court responses reinforce that this is an operational issue, not a speculative one. Australian courts have issued guidance and practice controls addressing AI-assisted work, particularly around verification and responsible use (Federal Court of Australia, 2025; Supreme Court of New South Wales, 2025; Supreme Court of Queensland, 2025). Commentary on law firm practice also identifies more routine risk pathways, including third-party data exposure, permissive vendor terms, hidden AI integration in everyday tools, cloud misconfiguration, and non-malicious human error such as uploading confidential material for editing or summarisation (Kolochenko, 2024).

This position has since been reinforced at the federal level. The Federal Court's Use of Generative Artificial Intelligence Practice Note (GPN-AI), issued on 16 April 2026, requires disclosure where GenAI has been used to summarise or analyse information relied on by a witness, or otherwise in a way that may affect admissibility or the Court's use of the evidence. It also requires that disclosure be made in the body of the document and warns that non-compliance may have consequences, including adverse costs orders and issues of professional compliance (Federal Court of Australia, 2026).

2.4 Existing approaches and adoption barriers for SMEs

A range of existing tools and approaches already address parts of the sensitive-document handling problem. These include manual redaction and review, generic data loss prevention and redaction products, de-identification libraries and frameworks, enterprise AI platforms with privacy and security positioning, and local pseudonymisation layers that mask identifiers before text is sent to an external large language model. These approaches are useful and, in many cases, materially reduce exposure compared with sending raw documents directly to a remote model.

However, the existence of such tools does not remove the implementation gap faced by small-to-medium legal practices. First, many available solutions are either

enterprise-oriented or technically fragmented. A firm may need to combine multiple components for text extraction, identifier detection, pseudonymisation, mapping storage, document reconstruction, retention handling, access control, and audit logging. For smaller practices, this creates cost, procurement, and operational barriers. Secondly, many tools are designed for generic PII handling rather than legal-document workflows. Legal documents often contain not only direct identifiers, but also contextual and role-based information, privilege-sensitive content, commercially confidential detail, and procedural facts that may remain identifying even after obvious names and numbers are removed. A tool that masks direct identifiers may therefore reduce risk without producing a document that is safe to use downstream in a controlled and defensible way. These issues are consistent with the contextual, governance-dependent approach to de-identification reflected in OAIC and CSIRO guidance, which emphasises that data risk depends not only on the transformed data itself but also on the release environment, available auxiliary information, and the controls around access and reuse (OAIC, 2018; O'Keefe et al., 2017).

Thirdly, many existing tools do not generate the evidentiary and governance outputs needed for legal practice. A transformed document on its own may not be sufficient. Firms may also need to know what policy was applied, what settings were used, what was retained, whether any mapping or substitution key exists, how long it is stored, and whether processing can later be reconstructed for review. This is especially important where confidentiality, professional duties, and internal governance expectations extend beyond privacy regulation alone. Finally, some available approaches still depend heavily on user judgement at the point of use. This can make adoption uneven, especially in SME environments without dedicated privacy engineering or legal operations teams. In practice, the issue is not that solutions do not exist, but that many do not translate cleanly into a realistic, low-friction, auditable workflow for sensitive legal-document handling in smaller firms. That broader workflow gap is also consistent with legal-sector guidance that emphasises verification, accountability, and responsible AI use rather than mere tool availability (CFLP, 2025). This is also consistent with the OAIC's guidance on the use of commercially available AI products, which states that the Privacy Act applies to uses of AI involving personal information and is intended to help organisations select appropriate products, understand privacy settings and terms, and manage risks when using publicly available chatbots, productivity assistants, and similar tools (Office of the Australian Information Commissioner, 2024)

This project is therefore positioned not as a claim that existing tools are absent, but as a response to the gap between existing technical capabilities and an SME-appropriate, firm-controlled workflow for producing risk-reduced "safe copies" of legal documents together with associated evidence and governance outputs. A key purpose of the workflow is to externalise and structure decisions that smaller practices may otherwise have to make inconsistently or ad hoc, including what should be transformed, what should

be retained, how re-identification risk should be framed, what evidence should be kept, and how downstream use should be bounded (OAIC, 2018; O'Keefe et al., 2017). Representative examples show how existing approaches solve important parts of the problem while still leaving gaps in legal-document workflow, governance, or SME implementability.

Example	What it does well	Why not enough on its own for this project
Presidio	Flexible open-source PII detection/transformation framework	Requires surrounding workflow, storage, audit, review, and legal-document handling design
Philter	Strong PHI/clinical text de-identification	Healthcare-focused, not built around legal-document workflows or evidence outputs
PII Shield	Local pseudonymisation before Claude sees text; restore after analysis	Tool-centric and model-adjacent; not by itself a broader governance and safe-copy workflow
Enterprise AI platforms	Can offer privacy controls and contracts	Often expensive, externally dependent, and may still not provide firm-specific audit/evidence logic

2.5 What outputs are produced: safe copy, evidence pack, audit, report

The prototype produces a structured export bundle rather than a single transformed file. Its primary output is a pseudonymised safe copy prepared for controlled downstream use while preserving enough structure and coherence for tasks such as summarisation, chronology-building, and limited AI-assisted processing. Each successful run also produces an evidence pack containing the supporting artefacts needed for review and defensibility, including a manifest, audit log, and court- or governance-facing report. Together, these artefacts are intended to show what was processed, what was changed, when processing occurred, and under which policy or configuration (International Organization for Standardization, 2024; Supreme Court of New South Wales, 2010).

2.6 Key constraints and non-negotiables

The prototype is intentionally constrained to a self-hosted, firm-controlled deployment model and does not use public cloud processing for source documents. It is designed around several non-negotiable governance requirements:

- pseudonymised outputs may remain personal information depending on context;
- metadata must be handled or its limitations clearly disclosed;
- auditability is mandatory rather than optional;
- and any mapping or substitution register is treated as a confidential re-identification key rather than a general output.

These constraints reflect the project's risk-reduction posture (International Organization for Standardization, 2024; OAIC, 2018; O'Keefe et al., 2017).

2.7 What this does not solve

The prototype does not, by itself, solve the broader governance problem of AI use. It does not guarantee anonymisation, does not automatically remove *Privacy Act* obligations, and does not make a document non-confidential merely because direct identifiers have been transformed. It does not replace professional verification, privilege review, or source checking, nor does it determine whether downstream legal content is accurate, complete, or safe to file or serve. The project also does not attempt full enterprise integration, public cloud deployment, guaranteed OCR for scanned or handwritten documents, or automated legal determinations, such as breach-reporting thresholds (International Organization for Standardization, 2024; OAIC, 2018; O'Keefe et al., 2017).

3. Background and problem context

3.1 Operational reality, GenAI workflows and document handling risk

Generative AI is increasingly used for practical legal tasks, including Q&A across large matter datasets, summarisation, chronology-building, and information extraction. These workflows routinely involve documents containing confidential client information, privileged material, and personal information. The operational risk is not "AI exists." The risk is that real documents are copied from controlled systems into multiple non-enterprise or non-tailored tools, failing to comply with ideal governance requirements and exercise professional duties responsibly.

Courts and professional bodies are consistently reacting to ineffective safeguards around (a) confidentiality, (b) verification, and (c) accountability. In current legal practice, many of the control activities described in this report, such as formal release registers, detailed reliance classification, or fully documented downstream-use records, are unlikely to be implemented consistently in every matter because of time pressure, fragmented tooling, and uneven workflow integration. That practical limitation is part of the project rationale rather than a weakness in the proposal. The purpose of the prototype is to show what a defensible workflow could look like now in a lightweight form, while also establishing a structured basis for future adoption as document management systems, integrated AI tooling, and workflow automation mature.

The governance gap is not merely detecting identifiers and replacing them. It is the absence of a repeatable, defensible workflow that keeps raw documents within controlled trust boundaries, applies consistent transformation rules, and produces evidence of what was processed, under which settings, and by whom.

3.2 Evidence of GenAI issues appearing in proceedings

To avoid speculative claims about overall adoption, this report relies on evidence that AI use or misuse is already appearing in proceedings. The UNSW Centre for the Future of the Legal Profession (CFLP) compiled a dataset of 520 matters across 10 jurisdictions, including 87 Australian matters, limited to instances where suspected or confirmed GenAI use or misuse is recorded or referenced in the official record (CFLP, 2025, p. 6). The CFLP expressly states that the dataset is not a prevalence study and likely represents the "tip of the iceberg" because non-case-law anecdotes are excluded and much of the use is never detected, recorded, or published (CFLP, 2025). That limitation supports a practical inference about governance. Visible cases likely understate actual use in legal workflows.

Market reporting points in the same direction. *Lawyers Weekly*, reporting on Clio's 2025 Legal Trends findings, stated that almost 100 per cent of Australian legal professionals use AI in some capacity and that most firms adopted AI within the last year, with common uses including workflow automation and document creation (Croft, 2025).

Australian court responses reinforce that this is an operational issue, not a speculative one. The Federal Court has issued a Notice to the Profession on the use of artificial intelligence in the Federal Court (Federal Court of Australia, 2025). The Supreme Court of New South Wales has issued Practice Note SC Gen 23 on the use of generative artificial intelligence (Supreme Court of New South Wales, 2025), and Queensland has introduced Practice Direction Number 5 of 2025, which requires verification of authorities in written submissions (Supreme Court of Queensland, 2025). Together, these developments indicate that AI-assisted legal work is already sufficiently common and sufficiently risky to warrant explicit control measures. For this project, the significance is not an abstract concern about AI. It is the reasonable likelihood that legal documents are already moving into AI-enabled workflows without consistent de-identification, metadata handling, verification, or auditability.

3.3 Why a workflow is needed, not just a tool

A key reason for adopting a workflow approach is that AI is easy to use informally. In legal practice, the governance problem is not simply whether identifiers can be detected, but whether matter data is kept within controlled trust boundaries, handled through a repeatable process, and supported by records that allow later review.

The problem is not limited to hallucinated citations or obviously defective filings. It also arises from routine document-handling practices involving AI-enabled tools. The Queensland Law Society guide reflects the same governance concern in practical terms by explicitly requiring firms to "[e]nsure staff are not using AI systems without approval" as an early deployment step, reinforcing that an approved tool register and a clear approval pathway are necessary to prevent unmanaged use (Queensland Law Society, 2024).

More recent practitioner guidance for small and mid-tier firms points in the same direction. A joint AI selection and use checklist released in February 2026 frames AI adoption as a matter of confidentiality, guardrails, vendor-risk assessment, supervision, record keeping, and ongoing review, rather than mere access to a tool. That supports the report's core position that defensible use depends on workflow, approval pathways, and governance controls, not only on whether a system is technically capable (Queensland Law Society et al., 2026)

Commentary from the American Bar Association (ABA) notes that many law firms' AI governance strategies remain nascent or non-existent, and warns that confidential firm

data may be exposed not only through dedicated generative AI tools, but also through the often-overlooked integration of AI into platforms lawyers use every day (Kolochenko, 2024). The ABA identifies practical risk pathways, including vendor terms that permit model training or broader data use, hidden third-party access, cloud misconfiguration, and non-malicious human error, such as uploading confidential documents to online tools for summarisation or editing (Kolochenko, 2024). Recent Australian law reform work points in the same direction. The Victorian Law Reform Commission treats safe AI use in courts and tribunals as requiring more than access to a tool, instead emphasising governance structures, clear responsibilities, AI policy settings, assurance processes, consultation, and public trust (Victorian Law Reform Commission [VLRC], 2025). These examples are directly relevant because they justify the need for a controlled, repeatable workflow for preparing legal documents before downstream processing.

3.4 Why pseudonymisation, not blind redaction

This workflow reduces identification risk; it does not make the content non-confidential. Even after pseudonymisation, a legal document can still contain privileged communications, litigation strategy, admissions, settlement positions, commercially sensitive facts, or other confidential matters that remain protected regardless of whether names and identifiers are removed. De-identification is commonly treated as a context-dependent risk-management practice rather than a guaranteed end state. OAIC guidance makes clear that information may be de-identified in one setting yet still be identifiable in another, and the APPA anonymisation guide similarly emphasises that risk depends not only on the data itself, but also on the context in which it is shared and used (Asia Pacific Privacy Authorities Technology Working Group [APPA Technology Working Group], 2025; OAIC, 2018). Recent LLM-focused deanonymisation research reinforces this caution by showing that contemporary models can extract identity-relevant signals from unstructured text, search auxiliary sources, and reason over candidate matches at scale; in practical terms, pseudonymisation cannot safely rely on obscurity alone. Deterministic pseudonymisation improves coherence and reviewer usability, but it may also preserve recurring patterns that contribute to residual inference or re-identification risk, particularly across repeated disclosures or larger document sets (Garat & Wonsever, 2022). For that reason, this project treats the safe copy as a risk-reduced artefact, not as a claim that the output is anonymous or automatically outside privacy obligations (Garat & Wonsever, 2022; Lermen et al., 2026).

3.5 Court expectations, minimisation, substitution registers, metadata hygiene

Courts and court publication processes generally do not prescribe numerical “redaction accuracy” measures. Instead, they focus on practical handling expectations:

- minimise unnecessary identifiers;
- use redaction or substitution methods that genuinely remove identifying content rather than merely obscuring it visually;
- maintain hygiene around document metadata;
- and control access to any unredacted or identifying reference material.

In other words, the question is not whether a tool was used, but whether the resulting document handling is appropriate and defensible (Federal Court of Australia, 2024; Supreme Court of New South Wales, 2010).

A first recurring expectation is minimisation. Court-facing anonymisation guidance warns that including unique personal identifiers can create a risk of identity theft and that this risk can often be reduced by avoiding, anonymising, or minimising such identifiers wherever possible. The same material gives concrete examples of partial masking or reduced-detail treatment of identifiers, such as obscuring passport or account numbers rather than reproducing them in full (Federal Court of Australia, 2024).

A second recurring pattern is the use of a separate substitution or reference register under controlled access. The NSW Supreme Court transcript anonymisation practice is a useful example. It applies automated substitution to transcripts and produces two related artefacts: an anonymised transcript and a register of substitutions for verification. Crucially, that register may be treated as a confidential exhibit and must be excluded from transcripts made available more broadly. This pattern is directly relevant to the project because it mirrors the distinction between a downstream-safe output and a confidential linkage artefact. It supports the idea that verification and privacy protection do not need to be carried by the same document. A public-facing or broader-circulation version can be pseudonymised, while the identifying truth remains under separate control (Supreme Court of New South Wales, 2010).

A third expectation is metadata hygiene. Court and publication guidance identify that anonymisation is not only about the visible text on the page. Personal information can persist in document properties, tracked changes, comments, hidden fields, and related metadata. Family Court anonymisation guidance explicitly emphasises the need to remove personal metadata from Word documents, and it treats anonymisation as a high-volume operational reality rather than a niche or exceptional task. Metadata handling is therefore not a formatting detail, but a core part of accountable document preparation (Federal Court of Australia, 2024).

3.6 Professional duties framing, confidentiality, verification, accountability

This project's governance posture is shaped not only by privacy regulation and de-identification guidance (OAIC), but also by the professional obligations imposed on legal practitioners under the *Legal Profession Uniform Law* (NSW) and the *Legal Profession Uniform Law Australian Solicitors' Conduct Rules 2015* (*Legal Profession Uniform Law Australian Solicitors' Conduct Rules 2015* (NSW) [ASCR], r 9.1). These obligations are technology-neutral, but GenAI and LLM-based workflows concentrate failure opportunities. Australian professional guidance has increasingly translated these duties into practical expectations as set out below.

3.6.1 Confidentiality and privilege preservation (ASCR r 9.1)

ASCR r 9.1 imposes a baseline duty not to disclose confidential client information acquired during engagement, except where authorised or otherwise permitted. The Victorian Legal Services Board + Commissioner, the Law Society of NSW, and the Legal Practice Board of Western Australia have made clear that lawyers cannot safely enter confidential, sensitive, or privileged client information into public AI chatbots/copilots (such as ChatGPT) or any other public tools. Absent strong contractual controls and a controlled processing environment, entering client-confidential, sensitive, or privileged material into public or uncontrolled AI tools poses a high risk of unauthorised disclosure, a risk that is difficult to reconcile with r 9.1 in practice (Law Society of New South Wales et al., 2024).

3.6.2 Independence and forensic judgment (ASCR rr 4.1.2–4.1.4)

The ASCR requires solicitors to act honestly and courteously, deliver legal services competently and diligently, and maintain integrity and independence (ASCR rr 4.1.2–4.1.4). Across reports, advice, briefings, guidelines, and caselaw, it has been consistently emphasised that LLM outputs may assist with drafting and structuring, but they do not exercise accountable legal judgment. A practitioner remains responsible for the advice, filings, and representations made to clients, courts, and other parties, and for the subsequent care that accompanies the tools used to make those submissions. Accordingly, downstream AI use in this project is framed as "assistance, not authority". This requires human review and task restrictions, including limiting AI use to lower-risk, more easily reviewable tasks, with independent verification required for reliance-critical outputs and auditability maintained throughout. This is also consistent with recent Victorian law reform analysis, which states that courts and tribunals should retain human oversight of decisions made or materially influenced by AI (VLRC, 2025).

3.6.3 Honesty, competence, and diligence

This project does not directly eliminate the risk of hallucination or fabrication in downstream LLMs. Any real deployment should operate alongside an existing risk policy that includes (a) verification obligations for any AI-assisted legal content, (b) restrictions on reliance-critical uses, and (c) traceability requirements sufficient to reconstruct how content was produced. If AI-generated outputs are adopted into work product without verification, this creates a direct pathway to misleading clients, the court, or other parties, which engages the practitioner's duties of honesty, competence, and diligence (ASCR rr 4.1.2 - 4.1.3). If the de-identified output is inaccurate, incomplete, structurally degraded, or insufficiently sanitised, any analysis generated from it may be unreliable or misleading, or may enliven confidentiality and privacy risks. If replacement variables undermine the document's usability or coherence, the resulting output may not support competent review or responsible downstream use.

3.6.4 Conflicts and billing incentives

If AI materially changes how work is produced, while requiring more verification overhead, governance ought to address cost integrity and incentives. ASCR r 12.2 is framed as a conflict rule that requires solicitors not to act where there is a conflict between the solicitor's interests and the client's best interests. Firms should take great care to avoid perverse incentives in which AI use, particularly where it is combined with the workflow proposed in this project, is treated as margin extraction rather than client benefit. Firms ought to ensure that any use of AI does not compromise client interests through reduced verification, diminished quality, or misrepresented effort, particularly given the heightened data-handling and verification risks associated with AI-assisted workflows. Cost fairness and proportionality are primarily grounded in the Uniform Law costs framework (ss 172–173) and should be reflected in internal billing guidance for AI-assisted work, in accordance with the *Legal Profession Uniform Law* (NSW) (the Uniform Law) and the firm's internal policy position. AI-assisted workflows may create incentives to compress review time while maintaining or increasing throughput. In legal practice, that creates a governance risk if efficiency gains are achieved by reducing supervision or verification, rather than by reducing low-value manual handling. At minimum, firms should ensure that any use of AI does not weaken review standards, misstate the basis of work performed, or create billing practices inconsistent with client interests.

3.6.5 Recommended controls derived from professional duties

To operationalise the duties set out by the Uniform Law and ASCR, the following implementation guidelines are recommended for any client information processing,

including the workflow proposed in this project. At a minimum, the following controls are expected:

- "**No raw or identifiable matter documents uploaded to external tools**": raw matter documents stay inside the controlled boundary, within the firm; only a risk-reduced safe copy may be used downstream where approved, within approved destinations only (ASCR r 9.1).
- **Mandatory verification for reliance-critical outputs**: any AI-assisted legal content must be verified against primary sources before filing, advising, or serving (ASCR rr 4.1.2–4.1.3; CFLP, 2025, pp. 3, 14–15).

- **Supervision and review gates**: staff use is supervised, and AI-assisted drafting is reviewed before settlement (ASCR rr 4.1.2–4.1.3; CFLP, 2025, pp. 3, 14–15).

- **Traceability and auditability**: record who processed what, when, and under which configuration; it is advisable to be able to explain the workflow in a court or governance-facing report.

- **Clear permitted-use policy**: define permitted tools, permitted tasks, prohibited tasks, and what information can be used where; make those rules enforceable through role separation and training (ASCR rr 4.1.3–4.1.4).

- **AI-enabled tools register:** The deploying organisation should maintain an approved tools register identifying which AI-enabled tools may be used, for what purposes, under which data-handling restrictions, and subject to what review date or approval status. Tools outside that register should be treated as unapproved for matter use.

Professional bodies have also moved to operationalise expectations for the responsible use of GenAI in legal practice. For example, the Law Society of NSW has published a solicitor-facing guide on responsible AI use, framing GenAI use as permissible only within a solicitor's ethical and professional obligations and warning against unsafe verification patterns. This supports the report's core governance premise. "AI use" does not dilute legal duties but does increase the need for disciplined controls and for workflows that can produce evidence of what was done and under which settings (Law Society of New South Wales, 2026).

4. Governance and methodology

4.1 Governance deliverables

This project is framed in terms of testable quality attributes, including confidentiality, integrity, auditability, privacy risk reduction, workflow reliability for internal staff, and output usability. A pseudonymised output that materially degrades chronology, entity-role relationships, or issue structure may be privacy-protective but operationally unfit for purpose. These attributes are supported by the backend evaluation plan and by the prototype's evidence artefacts, including audit logs, the manifest, and the court- or governance-facing report. In governance terms, the report is intended to produce a clear problem statement and scope, a standards-informed methodology, a risk register and threat model, an audit and reporting structure, and an evidence-based evaluation approach (International Organization for Standardization, 2023, 2024; Standards Australia & Standards New Zealand, 2023).

4.2 Risk management method, AS ISO 31000 approach and criteria

AS ISO 31000 defines risk as the effect of uncertainty on objectives and describes risk management as the systematic application of practices for communication and consultation, context establishment, risk assessment, risk treatment, monitoring and review, and recording and reporting (Standards Australia, 2018). In this project, the objective is to generate LLM-ready safe copies of legal documents while reducing confidentiality, privacy, and governance risk. AS ISO 31000 provides the organising method for the governance stream. It informs the identification of threat assumptions and trust boundaries, the structure of the risk register, the treatment of identified risks, and the documentation of control assumptions, limitations, and evidence of evaluation (Standards Australia, 2018, 2023a; Standards Australia & Standards New Zealand, 2024a).

4.2.1 Leadership, accountability, and user education

Consistent with AS ISO 31000's emphasis on leadership, integration into governance, and assigned roles and responsibilities, the deployment model assumes an accountable risk owner and clear operational roles (Standards Australia, 2018). A designated owner, such as a Knowledge Management leader or a Risk position such as a general counsel, is responsible for approving risk acceptance, maintaining policies, and ensuring users are trained in safe handling procedures. These responsibilities would include maintaining de-identification policies, retention settings, and safe-handling rules such as minimisation and "no raw upload" defaults. User education is treated as a control that reduces the risk of misuse and should be supported by basic evidence, such as training records or

acknowledgement documentation, with periodic refresh and review. The same general governance direction appears in the Victorian Law Reform Commission's recommendations, which emphasise ongoing education and training, including training at induction, when new AI tools are introduced, and on an ongoing basis, together with clearer guidance on how AI risks interact with existing professional obligations (National AI Centre, 2025; National Institute of Standards and Technology [NIST], n.d.; VLRC, 2025).

4.2.2 Communication, implementation, and evaluation

Effective risk management depends not only on assessment and treatment, but also on communication, implementation, and evaluation. In practice, this means that the organisation should adopt a clear approach to communication and consultation, identify who makes which decisions and at what stage, allocate appropriate time and resources, and ensure that risk-management arrangements are understood and followed in day-to-day practice. Evaluation should not be treated as a one-off activity. The effectiveness of the framework should be reviewed against its purpose, implementation plans, and expected behaviours, with updates made where internal practices, technologies, or external obligations change (Standards Australia, 2018, 2023b; Standards Australia & Standards New Zealand, 2024a).

As a practical governance control, the deployment model assumes the firm will first conduct a stocktake of AI tools already in use across teams and departments, then assess the associated risks, and finally provide clear direction on permitted experimentation and applicable data-handling constraints. Allens has recommended explicitly identifying whether and how legal teams are already using AI, noting that many leaders lack visibility into where AI has become embedded in day-to-day tools. This is reinforced by the Victorian Law Reform Commission's recommendation that AI tools used by courts and tribunals be publicly disclosed in an organisational AI inventory and that new AI uses be assessed through an AI assurance framework (Allens, 2023; VLRC, 2025). In project terms, that supports maintaining documented records of tool use, policy versions, and reviewable conditions of deployment. It also emphasises that any experimentation should be bounded by operational controls and vendor assurances regarding data security and confidentiality. This supports the project's broader position that tool selection alone is insufficient; organisations also need training, permissioning, and monitoring to reduce the risk of matter content being routed into uncontrolled environments (Allens, 2023; Kolochenko, 2024; National AI Centre, 2025; NIST, n.d.; VLRC, 2025).

4.2.3 Summary

In practical terms, AS ISO 31000 structures the governance report in five ways. First, it establishes scope through the definition of project scope, deployment assumptions, trust

boundaries, and internal and external data environments. Second, it informs risk identification by using attacker profiles, abuse cases, and disclosure scenarios. Third, it supplements risk analysis and evaluation through the risk register, including likelihood, impact, rationale, and residual risk assessment. Fourth, it supports risk treatment by selecting control requirements for access, retention, metadata handling, auditability, and downstream disclosure conditions. Finally, it informs monitoring, review, recording, and reporting through audit artefacts and manifest records, and highlights the need to re-evaluate the workflow when policies, tool versions, or recipient environments change (Standards Australia, 2018; Standards Australia & Standards New Zealand, 2024a).

This report also applies two AI-specific governance overlays without treating them as separate compliance frameworks. AS ISO/IEC 23894 is used as supplementary guidance alongside AS ISO 31000 to interpret AI-related risk in context, particularly where changing tools, external recipient environments, and stakeholder impacts require ongoing reassessment rather than one-off treatment (Standards Australia, 2023a). AS ISO/IEC 42001 is used at a benchmark level, not as a certification claim, to support the report's treatment of documented roles and responsibilities, operational control, performance evaluation, review, and continual improvement in relation to AI-enabled processing (Standards Australia, 2023b).

4.3 Privacy and de-identification governance basis: Standards and Regulations

4.3.1 ISO/IEC 29100

ISO/IEC 29100 provides the core privacy terminology used in this report. It defines pseudonymisation and explicitly warns that pseudonymised data may still be identifiable where linking information is available to the recipient or controller, supporting that proposition that document-handling risk is not limited to visible text alone (International Organization for Standardization, 2024).

This project also utilises ISO/IEC 29100 role language to clarify responsibility for handling personal information. In the default deployment model, the deploying law firm is treated as the PII controller because it determines the purposes and means of processing. The de-identification workflow service operates as a processor-style function within the firm's boundary and under the firm's instruction. Where applicable, managed service providers administering infrastructure strictly under instruction may also be treated as processors. PII principals include clients, witnesses, staff, and other third parties whose information appears in legal documents (International Organization for Standardization, 2024).

Third-party environments may include external enterprise LLM providers, external eDiscovery providers, external translation services, or any other recipient that does not operate solely on behalf of, and under the direction of, the firm. Once data is disclosed into such an environment, the recipient may remain a processor or may become a separate controller in its own right, depending on the arrangement and applicable legal framework. Consistent with this role logic, the report distinguishes between internal processing within firm-controlled systems and disclosure into external recipient environments, with the latter requiring additional safeguards, contractual controls, and documented conditions (International Organization for Standardization, 2024).

More broadly, ISO/IEC 29100 supports the project's emphasis on data minimisation, disclosure limitation, accountability, and information security as core privacy principles relevant to handling legal documents (International Organization for Standardization, 2024).

4.3.2 ISO/IEC 29134

Although this report is not a formal Privacy Impact Assessment (PIA), the governance approach intentionally follows a PIA-style structure to make privacy risk decisions explicit and auditable. ISO/IEC 29134 describes a PIA as a process for identifying, analysing, evaluating, consulting on, communicating, and planning the treatment of privacy impacts arising from PII processing within a broader risk-management framework (International Organization for Standardization, 2023).

ISO/IEC 29134 compliance would require threshold analysis to determine whether a PIA is necessary, as well as documented preparation, including the assessment team, the plan, and a description of the assessment. It also covers the performance of the assessment itself, including the identification of PII flows, the analysis of privacy implications, the determination of safeguarding requirements, the assessment of privacy risk, the preparation of risk treatment, and proportionate follow-up through reporting, implementation, review, and audit (International Organization for Standardization, 2023).

Where organisations intend to deploy automated de-identification workflows in production environments, a formal PIA or equivalent privacy risk assessment should be undertaken in accordance with recognised standards such as ISO/IEC 29134. Conducting a full PIA is outside the scope of this prototype project. However, the governance controls and documentation structures proposed in this report are designed to support such an assessment in future implementations. Accordingly, this report includes a threshold-style justification for privacy analysis, a documented description of PII flows and privacy risk assessment, and a proportionate risk-treatment and review structure suitable for a student prototype (International Organization for Standardization, 2023).

4.3.3 AS/NZS ISO/IEC 27559

AS/NZS ISO/IEC 27559:2024 reinforces the need for an objective and structured de-identification process (Standards Australia & Standards New Zealand, 2024b). It requires evaluation of the environment in which data will be accessed, shared, or released, and supports threat modelling that considers insider, accidental, and environmental disclosure scenarios. It also emphasises transparency and auditability, including the need to show what processing occurred, by whom, and when. In this project, those expectations are reflected in the trust-boundary model, structured audit logging, manifest design, and the requirement to produce reviewable evidence of what was processed and under which configuration. ISO/IEC 27559 also supports the use of recipient-side safeguards, including clear release conditions, restrictions on re-identification, and evidence-based evaluation of whether the receiving environment provides adequate protection (International Organization for Standardization, 2024; Standards Australia & Standards New Zealand, 2024b).

4.3.4 OAIC de-identification guidance and the APPA anonymisation guide

OAIC guidance provides the practical Australian framing for de-identification used in this report. It describes de-identification as a process involving the removal of identifiers, the removal or alteration of other identifying data, testing, and ongoing management of re-identification risk. The OAIC also emphasises that de-identification is context-dependent: information that may be de-identified in one environment can become identifiable in another. That point is central to this project, which assesses not only the transformed document itself but also the access environment into which it is released or used (OAIC, 2018).

The *Guide to Getting Started with Anonymisation* is a practical resource developed by APPA's Technology Working Group. It is a practical resource for organisations adopting anonymisation practices, reinforcing the same practical position. It emphasises that disclosure risk depends not only on the transformed data, but also on the surrounding governance setting, including who can access the data, what other information is available, and what controls constrain use. It also warns against describing pseudonymised data as anonymised where meaningful re-identification risk remains. For this project, those sources support the report's core position that safe copies should be treated as risk-reduced but still potentially sensitive outputs, especially where linking information, contextual detail, or broader recipient access remains available (APPA Technology Working Group, 2025; OAIC, 2018).

In Australian law, the relevant statutory baseline remains the *Privacy Act 1988* (Cth) where a transformed output is still "personal information", that is, where an individual

remains identified or reasonably identifiable in context. For this workflow, the most relevant practical principles are APP 6, APP 8, APP 10, and APP 11. APP 6 is engaged because downstream use or disclosure of a safe copy remains a use or disclosure question where the output still relates to a reasonably identifiable person. APP 8 may also be engaged where a safe copy is disclosed to an offshore provider or otherwise made available in a cross-border processing environment, affecting the majority of AI services. This triggers a review of whether the appropriate controls have been applied. APP 10 is engaged because transformation quality matters: a pseudonymised output that is inaccurate, misleading, or structurally distorted may create legal and governance risk, even where direct identifiers have been removed. APP 11 is engaged because raw inputs, temporary artefacts, safe copies, logs, manifests, and any mapping register all require reasonable steps to protect them from misuse, interference, loss, and unauthorised access, modification, or disclosure. For that reason, this report uses OAIC de-identification guidance as the practical interpretive layer for applying the Privacy Act threshold, rather than treating de-identification as automatically carrying an output as outside the jurisdiction of the *Privacy Act* (Privacy Act 1988 (Cth), sch 1).

4.3.5 CSIRO De-Identification Decision-Making Framework

The OAIC and CSIRO's Data61 developed the De-identification Decision-Making Framework to guide Australian organisations in reducing privacy risks when sharing or releasing data, developed primarily for data sharing and release contexts rather than document-by-document legal text workflows. When the principles are applied to other types of data, a central takeaway from that framework emerges indicating that "safe data" alone is insufficient. Instead, the CSIRO adopts a Five Safes approach prioritising safe projects, safe people, safe settings, safe data, and safe outputs. In this report it is used as an interpretive governance aid, not as a perfect methodological fit, particularly to emphasise that disclosure risk depends on people, settings, outputs, and surrounding controls rather than on transformation of text alone. In this project, that logic maps directly onto the trust-boundary model: safe settings correspond to controlled processing environments, safe people to role-based access and supervision, and safe outputs to the controlled release of safe copies and evidence artefacts (Australian Bureau of Statistics, 2021; O'Keefe et al., 2017).

For this project, the framework is helpful but must be adapted to the distinctive risk profile of legal documents. In legal workflows, identifiability may arise less from data volume than from factual specificity, strategic context, unusual combinations of attributes, or distinctive descriptions that remain meaningful even after direct identifiers have been replaced. A relevant example from the framework is that replacing names and places does not necessarily remove identifiability. A description such as a 16-year-old widow in remote Tasmania may still be reasonably identifiable in a small community such as

Strahan or Queenstown, particularly where local knowledge or client records are available for cross-reference (O'Keefe et al., 2017).

The CSIRO framework supports the project's core design choice to pair pseudonymisation with governance controls. In that sense, "safety" is treated as a matter of degree rather than a binary state, and the evaluation pack tests not only whether identifiers are detected, but whether the workflow's controls reliably enforce the intended trust boundaries and reduce inappropriate access or leakage (OAIC, 2018; O'Keefe et al., 2017).

4.4 PIA-style structure and why it is used in this report

Although this project is not a formal Privacy Impact Assessment (PIA), the governance structure of this report follows a simplified PIA-style approach. The purpose is to make privacy risks visible and show how they are considered in the design of the prototype. Using a PIA-style structure allows the report to clearly document how personal information flows through the system, what privacy risks exist during processing, and what controls are used to reduce those risks. This approach helps demonstrate that privacy considerations were included during system design, even though the project does not claim formal compliance with any regulatory framework.

4.5 Roles, responsibilities, and PII flows

The system involves several roles responsible for handling personal information, approving use, and maintaining the governance position of the workflow. In the intended deployment model, the organisation operating the system acts as the PII controller because it determines how documents are processed. The de-identification workflow itself operates as a processor-style system within that controlled environment and under those instructions.

For governance purposes, responsibility should be distinguished from accountability. Operational processing may be performed by an authorised user or operator, but accountability for the workflow's policy settings, approved uses, and residual-risk posture should sit with designated internal roles. At minimum, the operating model should identify:

- the policy owner responsible for maintaining the de-identification policy and approving material policy changes;

- the system or platform administrator responsible for controlled configuration, access administration, and operation of the service;

- the reviewer responsible for checking outputs before reliance-critical use or external disclosure where review is required;

- and the approving authority, such as a supervising lawyer, knowledge leader, or other designated risk owner, responsible for approving higher-risk downstream uses, including disclosure to external AI environments where permitted.

Where a downstream external platform is contemplated, accountability for approving that use should not rest solely with the operator who runs the workflow. The approval decision should instead sit with the role authorised to accept residual governance risk for that matter or use case, supported where necessary by legal, privacy, security, or compliance input. Similarly, responsibility for assessing whether an incident requires escalation, including possible regulatory or professional reporting, should be assigned clearly rather than left implicit.

- Personal information moves through the workflow in several stages:

- Document upload into the system

- Text extraction and preprocessing

- Detection of personal identifiers

- Redaction or pseudonymisation of identifiers

- Generation of a safe-copy output and supporting evidence artefacts

Most personal information exists in the ingestion and processing stages. After transformation, the output document is intended to reduce the likelihood that individuals can be identified. That reduction does not remove the need for accountable decision-making about downstream use, disclosure, retention, review, or incident handling.

4.6 Policy governance, versioning, change control, traceability

The behaviour of the de-identification workflow is controlled by a set of configurable rules that determine how identifiers are detected and transformed. Because these rules directly affect how personal information is processed, they are treated as controlled governance artefacts.

To support accountability, each configuration must have a unique policy identifier and version. Every processing run must record the policy name or ID, policy version or hash,

and any relevant runtime configuration parameters (such as confidence thresholds or detection cut-offs). Where multiple lenses or models are available for different domains, the specific model or recogniser used for each job must also be recorded.

Policy changes must be restricted to authorised administrators and managed as controlled changes to prevent accidental or unauthorised modification. All changes must be logged, including what was changed, by whom, and when. This ensures a clear separation between responsibility for maintaining policy configurations and responsibility for operating the workflow under those policies.

Traceability is maintained through audit records capturing timestamps, job identifiers, policy versions, and associated configuration details for each run. The evidence pack must allow outputs to be directly linked back to the exact policy and configuration used, without requiring reconstruction of the system environment.

Material changes to policy settings, tool versions, recogniser or model configurations, supported document types, or downstream recipient environments must be subject to formal change management and clearly defined approval authority. Such changes should trigger re-evaluation of relevant assumptions, risks, and supporting evidence where applicable, ensuring continued alignment with governance, auditability, and accountability requirements.

4.7 Tool selection rationale and non-claims (Presidio)

This project packages an existing open-source de-identification service, Presidio, rather than building a redaction engine from scratch. Presidio runs as a network service exposing an API. It processes incoming text/documents using a policy that defines what sensitive information to identify and how to treat it. It also supports confidence scoring and policy conditions that allow tuning detection thresholds such as ignoring entities under a confidence value (Microsoft, n.d.).

On the governance side, the tool was selected because it supports requirements central to the project's control model: policy-driven detection and treatment rules, pseudonymisation and structured replacement rather than simple visual redaction, reproducible configuration, versionable policy artefacts, and service-style integration into a self-hosted workflow (Microsoft, n.d.). These characteristics support defensibility because each processing run can be linked to a specific policy version and recorded in the audit trail, enabling later review of what was done and under which settings. Presidio is not treated as perfect detection. Accuracy depends on document characteristics and configuration, and both false negatives and false positives remain possible. Accordingly, the governance model requires human verification before reliance-critical use, treats

outputs as sensitive by default, and records limitations such as metadata handling and supported file types in the evidence pack. This report does not claim that Presidio is natively optimised for Australian legal text or that its default recognisers are sufficient without domain-specific tuning. That question is treated as an evaluation and implementation issue rather than as settled by tool selection alone (APPA Technology Working Group, 2025; International Organization for Standardization, 2024; OAIC, 2018; O'Keefe et al., 2017).

5. System overview and trust boundaries

5.1 System description and processing steps, ingest, transform, export

At a high level, the prototype accepts a document through a controlled internal upload point, validates it against the supported scope, applies policy-governed transformation, primarily through pseudonymisation, and generates an export bundle for review and traceability. The workflow is designed to reduce direct identifier exposure before any downstream use while preserving enough readability and referential coherence for the intended low-risk tasks while ensuring access is firmly regulated(International Organization for Standardization, 2024; Supreme Court of New South Wales, 2010).

5.2 Deployment model and boundary conditions, self-hosted assumptions

The governance model assumes deployment on firm-controlled infrastructure rather than on public cloud platforms for confidential and sensitive source documents. The report does not assume that self-hosting is inherently more secure than a mature enterprise cloud environment. Rather, the self-hosted deployment model is used here as a boundary-control choice that reduces ordinary third-party exposure and increases local control over processing, retention, and auditability. That choice also transfers operational security responsibility to the deploying organisation. Where a firm lacks sufficient security maturity, patching discipline, access control, monitoring, or incident capability, self-hosting may increase overall risk rather than reduce it. Where a safe copy is later disclosed to an external platform, such as an enterprise LLM or another downstream service, that use is treated as disclosure into a separate environment that requires its own risk assessment, safeguards, and documented conditions (International Organization for Standardization, 2024; OAIC, 2018; O'Keefe et al., 2017).

5.3 Trust zones, input, processing, outputs, admin/config

For governance purposes, the workflow is understood as operating across four trust zones. The input zone contains uploaded documents and is treated as untrusted because files may contain both confidential information and potentially unsafe content. The processing zone is the controlled environment in which de-identification rules are applied, and temporary processing artefacts may be created. The output zone contains the safe copy and evidence artefacts, which remain sensitive even after pseudonymisation because they may still contain personal information or disclose patterns useful for re-identification.

The admin or configuration zone contains policy settings, retention controls, and privileged access to logs and related artefacts. The purpose of distinguishing these zones is to make clear that different handling, access, and review assumptions apply at each stage of the workflow (International Organization for Standardization, 2024; Standards Australia & Standards New Zealand, 2023).

5.4 Identity and access model assumptions

The governance model assumes role-based access controls (RBAC) with separation between ordinary processing and privileged administration. Given the project scope, authentication may be implemented as local application accounts, a reverse proxy enforcing authentication and optionally MFA, or VPN-only exposure plus basic authentication. Whatever model is adopted must be clearly documented and treated as part of the risk context. Self-hosting is also not treated as “safe by default”; it requires an explicit, reviewable access-control stance and guidelines (Australian Cyber Security Centre, 2023; Standards Australia & Standards New Zealand, 2022, 2023).

5.5 Data situation audit, environments E1 internal, E2 downstream platform, E3 court or disclosure

Consistent with the framework above, de-identification is assessed contextually rather than by transformed text alone. The relevant environment includes not only the document output, but also the other data available in that environment, the actors able to access or combine that data, and the governance and technical controls that constrain use. Accordingly, this workflow evaluates identifiability across three disclosure environments: the internal operating environment (E1), the downstream platform environment (E2), and the court or disclosure environment (E3) (International Organization for Standardization, 2024; OAIC, 2018; O'Keefe et al., 2017).

Environment	Data in scope	Plausible linkage sources	Relevant actors	Governance and control considerations	Primary risk emphasis

E1 Internal	Raw uploads, intermediate artefacts, safe copy, audit logs, manifests, court- or governance-facing report, and any optional substitution or mapping register	Metadata, file naming conventions, document management systems, email, client or matter registers, prior exports, and internal records	Standard users, reviewers, administrators, IT operators, and plausible insider misuse or credential-compromise scenarios	Role-based access control, retention rules, policy approval, reviewer sign-off, prohibition on re-identification attempts, logging, and incident handling	Residual linkage through internal systems, excessive access, and misuse by authorised or compromised users
E2 Downstream platform	Safe-copy excerpts or full safe copies, prompts, outputs, and platform-side logs	Provider telemetry, account data, prior prompts, public information, and internal details reintroduced by users	Approved internal users, platform operators, and relevant sub-processors	Contractual terms, retention and deletion settings, approved use cases, sub-processor controls, and downstream auditability	Loss of contextual control, downstream retention, re-linkage through prompts or outputs, and provider-side visibility
E3 Court or disclosure	Filed documents, transcripts, annexures, public-safe reports, and other disclosure outputs	Court lists, public reporting, social media, public registers, and third-party datasets	Court staff, parties, external recipients, media, and the broader public where disclosure is public or practically accessible	Strong minimisation, metadata hygiene, verification, controlled filing practice, and strict separation of any confidential substitution register from broader-circulation outputs	Public re-identification, irreversible disclosure, and amplified linkage through external datasets

Across these environments, the same transformed document may present materially different identifiability risk depending on the surrounding data situation. In practice, E1 concentrates risk in internal access and linkage through existing organisational systems, E2 shifts risk toward downstream reuse and provider-side handling, and E3 creates the

greatest exposure to external linkage because public or semi-public datasets may substantially reduce the practical protection achieved by text-level transformation alone (Federal Court of Australia, 2024; International Organization for Standardization, 2024; Kolochenko, 2024; Supreme Court of New South Wales, 2010).

6. Threat model and risk register

6.1 Assets and attacker profiles, insider, accidental, external compromise

The principal assets in this workflow vary in sensitivity, but all may pose confidentiality, privacy, or governance risks if exposed or mishandled. The relevant attacker profiles for this project are not limited to malicious outsiders. They also include internal staff with legitimate access, unintended recipients, compromised user accounts, and downstream platform-side exposure. The threat model considers insider misuse, accidental disclosure, credential compromise, and external compromise as distinct but overlapping pathways (International Organization for Standardization, 2024; Standards Australia, 2018, 2023a).

6.2 Threat assumptions and abuse cases

In the intended deployment model, most users of the workflow are internal staff of the firm. The threat model focuses heavily on internal misuse, accidental disclosure, and compromise of legitimate access rather than on anonymous public users. Relevant threat assumptions include authorised misuse by staff, accidental mishandling of sensitive outputs, credential compromise through phishing or password reuse, and downstream disclosure into less controlled environments. These assumptions are consistent with the project's broader risk-based method and with de-identification guidance that emphasises realistic disclosure pathways rather than only direct-identifier removal (International Organization for Standardization, 2024; OAIC, 2018; O'Keefe et al., 2017).

6.3 Priority risks

Based on the trust-boundary model, the expected deployment environment, and the disclosure scenarios described below, the following risks are treated as the project's initial priority risks. This list is informed by ISO/IEC 27559's emphasis on realistic disclosure pathways and by court-adjacent handling expectations for sensitive legal documents (Federal Court of Australia, 2024; International Organization for Standardization, 2024; Supreme Court of New South Wales, 2010).

- Unauthorised access to uploads and outputs.
- Credential compromise.
- Insider misuse, including authorised user abuse.

- Accidental disclosure.
- Uncontrolled retention of raw uploads and temporary files.
- Audit gaps or audit tampering.
- Pseudonymisation failure.
- Inconsistent pseudonyms.
- Metadata leakage.
- Dependency vulnerabilities.
- Resource exhaustion or denial of service.
- Evidence pack sensitivity.
- Prototype misuse risk, including the treatment of pseudonymised output as anonymous.

6.4 Disclosure scenarios and auxiliary information pathways

The governance model evaluates re-identification and disclosure risk using scenario analysis, focusing on how a disclosure might realistically occur, rather than only on whether visible identifiers have been removed (International Organization for Standardization, 2024; OAIC, 2018; O'Keefe et al., 2017).

6.4.1 Core disclosure scenarios

A. Insider misuse

Field	Description
Intruder	Staff member with legitimate access.
Auxiliary information	Matter context, client knowledge, and access to a mapping register if misconfigured.
Attack path	Export of a safe copy together with a mapping register, allowing reconstruction of identities.

Controls	Strict role separation, restricted mapping-register access, audit logging, least-privilege access, training, and sanctions.
Evidence	Role-based access-control test results, access logs, denied-access events, and evidence that mapping-register access is restricted to administrators.

B. Credential compromise (phishing or password reuse)

Field	**Description**
Intruder	External attacker using compromised credentials.
Auxiliary information	Exported outputs and any publicly available information.
Attack path	The attacker authenticates using compromised credentials, downloads exports, and exfiltrates the evidence pack or related outputs.
Controls	VPN or reverse-proxy authentication, MFA where available, rate limiting, monitoring, retention minimisation, and encrypted storage.
Evidence	Unauthorised endpoint tests, login-failure logs, download logs, and deletion-event records.

C. Accidental disclosure (wrong recipient, wrong folder, or mis-share)

Field	**Description**
Intruder	Unintended recipient, not malicious.
Auxiliary information	None required, because the disclosure is direct.
Attack path	A safe copy or audit report is emailed to the wrong recipient, placed in the wrong shared location, or exposed through an incorrect link.
Controls	Treating outputs as sensitive by default, operator checklist, minimum-necessary handling, separation between broader-circulation reports and confidential registers, and retention and access controls.

Evidence	Operator guidance, report templates containing sensitivity labels, and evidence that confidential and broader-circulation outputs are separated.

D. Metadata leakage (hidden identifiers persist)

Field	**Description**
Intruder	Any recipient opening the file, or any person using metadata-extraction tools.
Auxiliary information	Metadata fields, tracked changes, comments, hidden text, and document properties.
Attack path	A recipient extracts metadata or hidden content and recovers names, identifiers, or contextual information not removed from the visible text.
Controls	Metadata stripping for supported formats, explicit limitation warnings where stripping is incomplete, and reviewer checks.
Evidence	Test files containing tracked changes, comments, or document properties, and evidence that metadata is either removed or clearly flagged as a residual risk.

E. Downstream platform retention and secondary exposure (enterprise LLM or third-party system)

Field	**Description**
Intruder	Provider personnel, other account holders, or attackers compromising the downstream platform.
Auxiliary information	Platform logs, prior prompts, retained content, and account metadata and usage context.
Attack path	The downstream platform retains prompts, outputs, or uploaded safe-copy content, and later access or misuse occurs.
Controls	Contractual controls, retention settings, purpose limitation, access restrictions, auditability, and a default rule prohibiting raw-document upload.

Evidence	Documented contractual checklist, platform-configuration evidence where available, and governance-report fields identifying the intended downstream environment.

6.5 Risk treatment plan, controls selected, controls deferred, residual risk

Detailed technical analysis of implementation-specific risks, including upload validation, temporary file handling, dependency risks, and service-level hardening, is provided in the development and backend reports. The present section focuses on governance-significant risk categories, disclosure pathways, and control expectations relevant to legal-document handling.

At the governance level, the selected controls serve two distinct functions. Some are primarily preventive and protective, meaning they are intended to reduce the likelihood or impact of disclosure. Others are primarily detective and accountability-oriented, meaning they are intended to support reconstruction, review, supervision, and incident response after processing has occurred.

Control grouping	Primary function	Example controls	Governance significance
Preventive and protective controls	Reduce likelihood or impact of disclosure	Access separation, metadata handling, mapping-register restriction, retention and deletion rules	Reduce exposure to internal misuse, accidental disclosure, and residual identifiability
Detective and accountability controls	Support reconstruction, review, supervision, and incident response	Audit logging, manifests, governance-facing reports	Support traceability, oversight, and post-event review

Both control groupings are necessary because they address different dimensions of risk. Preventive controls reduce exposure in advance, whereas detective and accountability controls support organisational assurance once processing has occurred. Together, they address the principal governance risk pathways identified in this report: internal misuse, accidental disclosure, residual identifiability, and inability to reconstruct what occurred after processing (International Organization for Standardization, 2024; OAIC, 2018; Standards Australia, 2018).

Some controls are intentionally deferred because they depend on implementation maturity, organisational infrastructure, or future production deployment decisions. These include enterprise identity integration, advanced monitoring and alerting, full production incident response, broader document-management-system integration, and any organisation-wide privacy impact assessment or compliance program. They are outside the scope of what this student prototype can credibly claim to implement and evaluate (International Organization for Standardization, 2023; Standards Australia & Standards New Zealand, 2023).

7. Requirements

7.1 Functional requirements

The system must support a controlled, end-to-end document processing workflow. It must be capable of ingesting supported document formats, initially limited to text-based PDF files, and extracting their contents for processing.

The workflow must apply policy-driven transformations, including redaction and pseudonymisation, and produce a corresponding output in a consistent and usable format. Outputs must be generated as part of a structured bundle, including both the transformed document and associated artefacts.

The system must support repeatable execution, such that the same input processed under the same policy version produces consistent outputs. It must also provide a simple operator interface, such as a command-line interface or lightweight web interface, to submit jobs, monitor progress, and retrieve outputs. The workflow must enforce separation between input, processing, and output stages, consistent with defined trust boundaries.

7.2 Privacy requirements

The system must reduce the likelihood that individuals can be identified from processed outputs, while recognising that de-identification is a risk management process rather than a status.

Direct identifiers must be removed or transformed, and indirect identifiers must be handled in a manner that reduces re-identification risk. The system must support consistent pseudonymisation where required to preserve document structure while limiting exposure of identifiable information.

Processing must occur within a controlled, self-hosted environment, and raw source documents must not be exposed to external systems. Outputs must be treated as sensitive by default, and any further processing must be subject to organisational policy and review.

The system must also support minimisation principles, ensuring that only necessary information is retained in outputs and that unnecessary identifiers are not preserved.

7.3 Security requirements

The system must operate within a controlled environment with clearly defined trust boundaries between input, processing, output, and administrative zones. Access to each zone must be restricted based on role and necessity.

Processing services must follow baseline hardening practices, including non-root execution, minimal exposure of network services, and restricted inter-component communication where feasible. Temporary files generated during processing must be isolated and protected from unauthorised access.

Sensitive artefacts, including any substitution or mapping registers, must be stored separately and accessible only to authorised users. The system must also assume minimal external exposure, with deployment intended for local or controlled network environments.

Security controls must be designed to reduce the risk of unauthorised access, unintended disclosure, and misuse of both inputs and outputs.

7.4 Auditability requirements

The system must generate sufficient audit artefacts to allow each processing run to be reconstructed and reviewed. This includes maintaining a record of what was processed, when processing occurred, who performed actions, and under which policy settings or system configuration.

An append-only event log must be generated for each job, capturing key processing steps, timestamps, and system actions in chronological order. A structured manifest must also be produced, summarising the processing configuration, outputs generated, and relevant metadata associated with the run.

Audit events must include, where applicable:

- upload events (user or role, timestamp, file ID or hash, file type, and size)
- processing start and end events (including success or failure status)
- pseudonymisation or redaction summaries (recorded as counts by category, avoiding raw sensitive values)
- export bundle generation (bundle ID or hash)
- download or access events (who accessed outputs and when)
- deletion events (covering raw inputs, temporary files, and outputs where applicable)

- authentication and authorisation events (login success or failure, and privilege changes)

Audit records must be designed to support traceability without exposing raw identifiers or sensitive data. Where possible, identifiers should be represented through counts, hashes, or abstract references rather than direct values.

All audit artefacts must be preserved and made available for authorised review, supporting internal governance, evaluation, incident investigation, and verification processes. Logs must support a clear “who did what, when” record, aligning with audit trail and transparency expectations. Accountability for system use must be demonstrated through these records, including evidence of system actions, decisions, outputs, and errors, rather than through unsupported assertions alone.

7.5 Metadata handling requirements

The system must account for metadata as a potential source of identifying information. This includes document properties, tracked changes, comments, and embedded data that may not be visible in the main document content.

Where feasible, metadata must be removed, transformed, or normalised as part of the processing workflow. The system must ensure that sensitive information is not retained in hidden fields or auxiliary data structures.

Any limitations in metadata handling must be clearly documented, including cases where metadata cannot be reliably removed or where format constraints apply.

Metadata handling must be treated as an integral part of de-identification rather than a secondary step.

7.6 Retention and deletion requirements

The system must implement controlled retention and deletion practices for all processing artefacts. Raw input files must only be retained for the duration necessary to complete processing and must be securely deleted thereafter.

Temporary processing files must be stored in isolated locations and removed immediately upon completion of the job. The system must ensure that no residual raw data remains in working directories after processing.

Retention of outputs, logs, and manifests must follow defined policies, balancing audit requirements with data minimisation principles. Sensitive artefacts must not be retained longer than necessary for their intended purpose.

Deletion processes must be consistent and verifiable, ensuring that data is not recoverable through normal system access.

7.7 Evidence pack requirements, manifest, logs, report, hashes

Each processing run must produce an evidence pack that consolidates all relevant artefacts into a structured and portable format. This pack must include the processed output (safe copy), a manifest, an event log, and a governance-facing report.

The manifest must summarise key information about the processing run, including inputs, outputs, policy version, and configuration details. The event log must provide a chronological record of processing steps and system actions.

The report must provide a human-readable summary of what was processed and how, including any limitations or residual risks. Where appropriate, hashes or checksums may be used to verify the integrity of files and support reproducibility.

The evidence pack must be generated in a consistent structure, enabling review, validation, and reuse across evaluation and reporting stages.

8. Controls and operational governance

8.1 Access control, roles, least privilege, admin separation

At minimum, the system must distinguish between ordinary processing access and privileged administrative access. Administrative privileges should be limited to governance-sensitive functions such as policy configuration, retention settings, role management, mapping-register access, and audit-log access. Where review is part of the operating model, a reviewer role may also be recognised. The access model should be documented clearly enough to show who may perform privileged actions, who may approve higher-risk downstream use, and how those privileges are controlled and reviewed. Standard processing access should not automatically enable a user to alter policy, include restricted artefacts in an output, or approve external downstream disclosure. This supports least-privilege and administrative-separation principles by ensuring that routine processing access does not automatically grant control over settings or artefacts that materially address confidentiality, traceability, or reviewability (Standards Australia & Standards New Zealand, 2022, 2023).

8.2 Data handling, retention defaults, temp files, deletion events

The workflow must define, in explicit and reviewable terms, where uploaded source documents and temporary processing artefacts are stored during processing. Raw uploads should be retained only for the minimum period necessary to complete processing and generate the output bundle, with deletion after export as the default unless a different retention setting is deliberately configured and justified. The same expectation applies to temporary artefacts created during processing. Their handling and deletion behaviour should be documented clearly enough to support later review, incident reconstruction, and assessment of whether raw identifying material persists beyond the intended processing window (International Organization for Standardization, 2023; Standards Australia & Standards New Zealand, 2023).

8.3 Mapping register handling, confidential register separation

The same level of specificity is required for pseudonym mapping storage. The system should state whether mappings are generated and retained per document, per job, per session, or per matter, and whether any mapping artefact is included in the export bundle at all. If a mapping register or substitution table is created, it must be treated as a sensitive re-identification key rather than as part of the general downstream output. Consistent with the court-facing separation reflected in substitution-register practice, any such register should be stored separately from broader-circulation or “public-safe”

outputs, subject to stricter access and retention controls, and excluded from ordinary disclosure or downstream AI use unless there is a specific and controlled reason to retain it. In governance terms, the existence of a mapping register is itself part of the disclosure-risk context and should be recorded as such (International Organization for Standardization, 2024; Supreme Court of New South Wales, 2010).

8.4 Secure deployment baseline, hardening assumptions, patching, backups

This report does not prescribe the full technical deployment design, which is addressed in the backend and development streams. At the governance level, however, the deployment context must be documented clearly enough to support risk assessment and later review. This includes the hosting model, the assumed network exposure, the responsible party for patching and maintenance, the principle of minimal exposed services, and the existence of backup and recovery arrangements for configuration and evidence artefacts. These matters are included here because they affect confidentiality, integrity, auditability, and defensibility, not because this report claims to provide the implementation design itself.

Accordingly, this report states the control objectives, assumptions, and minimum governance expectations for secure deployment, while the backend and development streams are expected to implement and evidence the relevant technical mechanisms. Where those streams define service hardening, network restrictions, logging implementation, temporary-file handling, or other technical safeguards, they should be read as the technical implementation of the governance requirements stated here.

These documented assumptions should extend to operating procedures, privileged-access handling, protection of logs and evidence artefacts, backup and restoration expectations, and the collection and preservation of evidence if a security event, suspected misuse, or incident occurs. Because the workflow is self-hosted, the governance model requires documented hardening assumptions for the host, container, dependencies, and exposed services. The project does not claim Essential Eight compliance, however, the Essential Eight is used as a practical Australian benchmark for hardening guidance and, more importantly, for evidence framing. In particular, the evaluation strategy prefers controlled demonstrations of control effectiveness over screenshots or policy assertions alone. For this prototype, that means documenting hosting assumptions, network exposure, patching responsibility, backup expectations, and any tested restrictions on access, privilege, or service exposure (Australian Cyber Security Centre, 2023; Standards Australia & Standards New Zealand, 2022, 2023).

8.5 Downstream enterprise LLM use controls, contractual minimums and purpose limits

Contractual controls are not sufficient, by themselves, to eliminate governance risk when an enterprise LLM platform is used. They may reduce some supplier-side risks, such as secondary use, uncontrolled retention, or cross-tenant exposure, but they do not remove legal duties or the fact that matter documents may still contain privileged, confidential, and sensitive personal information. For that reason, de-identification remains the default gate for downstream AI use, even where the downstream platform is enterprise-grade (International Organization for Standardization, 2024; OAIC, 2018). This is also consistent with OAIC guidance on commercially available AI products, which emphasises that the Privacy Act applies where personal information is used with such products and that organisations should assess matters such as the suitability of the product, the provider's terms and privacy settings, handling of prompts and outputs, and whether the product is appropriate for the sensitivity of the information involved (Office of the Australian Information Commissioner, 2024).

Where a safe copy is disclosed to an external enterprise LLM provider, that disclosure should be treated as used in a separate environment requiring additional safeguards. At minimum, contractual controls should address purpose limitation, prohibition on training or secondary use, prohibition on re-identification attempts, retention and deletion requirements, sub-processor restrictions, and auditability. Governance should also require review of material supplier changes, including new subprocessors, changed retention settings, significant service-term changes, or deterioration in the provider's security or privacy posture, consistent with current practitioner guidance on third-party AI selection. Approval for downstream external AI use should therefore remain conditional and reviewable rather than assumed to remain adequate indefinitely (International Organization for Standardization, 2024; Queensland Law Society et al., 2026; VLRC, 2025).

Generic SaaS or processor terms may also be inadequate for AI services because they often fail to address model improvement, embeddings, vector-store persistence, or layered subprocessor chains. Accordingly, governance should require AI-specific terms that restrict reuse of customer inputs, outputs, prompts, and derivatives, flow equivalent restrictions down to subprocessors, and extend deletion obligations to derived representations such as embeddings or vectorised content. Due diligence should also extend to the provider's actual control environment, including available assurance artefacts, encryption in transit and at rest, multi-factor authentication, role-based access control, audit logging, data residency arrangements, subprocessor transparency, and incident-notification processes. Where external enterprise LLM use involves offshore processing or disclosure to foreign subprocessors, APP 8 is also relevant because cross-border disclosure of personal

information may arise even where the originating workflow is locally controlled (Privacy Act 1988 (Cth), sch 1 app 8).

These measures improve defensibility and reduce disclosure risk, but they do not convert pseudonymised output into non-personal information by default. Even where only a safe copy is disclosed, downstream LLM use remains a residual-risk activity rather than a presumptively safe one. Contractual, technical, and organisational safeguards may reduce that risk, but do not eliminate model-assisted re-identification or contextual de-anonymisation risk (International Organization for Standardization, 2023, 2024; Kolochenko, 2024; Pasupuleti, 2024; VLRC, 2025).

8.6 Training and operator guidance controls

Training and operator guidance are treated as governance controls rather than optional support material. Users should be instructed on the distinction between raw source documents, safe copies, and any confidential mapping register; the rule that safe copies remain sensitive by default; the prohibition on uploading raw or identifiable matter documents to external AI tools; the requirement to apply minimum-necessary disclosure in downstream use; and the need for human verification before any reliance-critical use. Guidance should also explain supported file types, known metadata-handling limits, retention and deletion defaults, and when escalation or reviewer sign-off is required.

Training should be role-appropriate and refreshed when policies, tools, or downstream environments change. Basic records of training or acknowledgement should be retained as governance evidence. Operator guidance should also make clear that minimum-necessary disclosure is a practical rule for downstream AI use. In ordinary cases, only the smallest portion of the safe copy reasonably required for the task should be used. Where broader context is genuinely needed, that expansion should be deliberate and justified rather than treated as a default convenience (NIST, n.d.; Standards Australia, 2018; VLRC, 2025).

8.7 Client engagement terms, consent posture, and privilege-preservation controls

Internal self-hosted processing is treated as part of ordinary service delivery rather than as third-party disclosure. By contrast, use of an external AI provider should be treated as disclosure into a separate environment and governed through prior risk review, enforceable provider terms, and client-facing authorisation language where required. Firms should therefore consider reflecting their technology practices in engagement terms, including the use of secure internal systems, restrictions on uploading unredacted

confidential documents to public AI tools, and any additional safeguards applied where external providers are used.

The privilege implications of disclosing even a pseudonymised safe copy to an external AI provider are not yet settled in Australia. The safer governance assumption is therefore neither that privilege is necessarily waived in every case nor that external AI use is simply equivalent to a neutral internal tool. Rather, external provider use should be treated as a legally and operationally distinct disclosure environment that may affect confidentiality, control, and privilege analysis depending on the facts, contractual terms, technical architecture, and extent of disclosure. For that reason, this report treats external AI use as a residual-risk activity requiring specific review rather than as a presumptively safe or routine downstream step.

The privilege consequences of disclosure to an external AI provider may also require attention under s 122 of the *Evidence Act 1995*, which addresses loss of client legal privilege by consent. In practical terms, disclosure of a privileged document to an external enterprise LLM is increasingly characterised as inconsistent with maintaining the confidentiality on which the privilege depends, with relevant US and UK cases concurring. A pseudonymised safe copy may reduce that risk by limiting direct identifier exposure and narrowing the content disclosed, but it should not be treated as automatically preventing waiver. (Evidence Act 1995 (Cth), s 122; Evidence Act 1995 (NSW), s 122).

Particular attention should be given to confidentiality, preservation of legal professional privilege, and whether clients should be given a clear opportunity to decline external processing pathways where appropriate. This report does not attempt to prescribe retainer wording as legal advice or to determine waiver consequences as a matter of law. Its position is that uncertainty in the present Australian setting is itself a reason to require prior risk review before third-party AI processing is used. Where such processing is contemplated, the client-authorisation posture should align with the sensitivity of the material, the surrounding contractual safeguards, and the privilege implications of the relevant environment. In the present Australian position, the more clearly established obligations are those concerning confidentiality, verification, supervision, and responsible handling, as reflected in court guidance and professional materials, rather than a fully settled doctrine on privilege waiver in the AI context (International Organization for Standardization, 2024; VLRC, 2025).

9. Court and governance -facing reporting

9.1 Report template, mandatory sections and disclaimers

Each processing run must generate a governance-facing report as part of the evidence pack. The purpose of this report is to provide a clear, reviewable account of what was processed, how it was handled, and the policy settings under which the system operated.

At a minimum, the report must include a concise processing summary outlining the job ID, execution timestamp, operator (where applicable), and the policy or system version used. It should also clearly state the purpose of the run and the method applied, allowing reviewers to quickly understand the context and intent of the processing activity.

The report must describe the input material at a high level, including file type, document category, and overall volume. A category-level summary of transformations applied must be included, such as redaction, pseudonymisation, and any metadata handling procedures. Where relevant, simple indicators of detection and transformation coverage should be provided to support interpretation and transparency.

In addition, the report must include a clear metadata-handling statement, specifying how embedded or associated metadata was treated during processing. It must also explicitly state that any mapping or substitution registers (e.g., re-identification keys) are confidential and excluded from outputs intended for broader circulation.

A limitations and residual risk statement is required to ensure the output is not misinterpreted as fully anonymised. The report must clearly communicate that outputs are risk-reduced “safe copies” and may still contain identifiable or sensitive information depending on context. It should also emphasise that the workflow does not replace human judgment and that any downstream use remains subject to organisational policy and professional obligations.

Finally, the report must include a reviewer verification and sign-off section confirming that the output has been reviewed and approved prior to use.

9.2 Release and share register fields

A release and share register is a useful governance control for recording downstream use or disclosure of processed outputs. It can provide a traceable record of how data moves

beyond the controlled processing environment and support audit, accountability, and review.

For this prototype, the release and share register is best treated as a recommended or future-state control rather than a baseline assumption for ordinary use in every matter. Its practical value is greatest where outputs are disclosed outside the internal environment, relied on in formal work product, or otherwise subject to heightened review expectations. In those contexts, the register should be capable of recording enough detail to reconstruct what was shared, when, under whose authority, to which recipient environment, and subject to what conditions.

Where a release and share register is used, it should record the relevant job or bundle identifier, the date and time of release, the individual approving the release, the intended recipient, and the purpose of the release. It should also record the classification of the output being shared, such as whether it is a safe copy or a restricted artefact.

A future implementation could allow this record to be generated at user request at the point of export and, where appropriate, stored alongside the relevant evidence pack or filed into a document management system. At minimum, relevant fields would include the job or bundle identifier, policy version, recipient environment, intended purpose, whether metadata handling was applied, whether a mapping register exists, and reviewer identity where reviewer approval was required.

Where applicable, the register should also record confirmation that no raw or identifiable source material has been disclosed, together with any conditions imposed on the recipient, such as restrictions on reuse or re-identification. Where outputs are shared outside the internal environment, the register should also reflect whether the receiving environment meets minimum governance and confidentiality expectations. If maintained, the register should be available for internal review and incident investigation.

9.3 Verification and sign-off process

Processed outputs intended for reliance-critical use or external disclosure must undergo human verification prior to release. This step ensures that outputs align with policy expectations and do not introduce unnecessary residual risk.

Verification involves a reviewer assessing whether identifiers have been appropriately handled, whether the output remains structurally usable, and whether any obvious confidential or identifying information persists. The reviewer must also evaluate whether

the output is suitable for its intended purpose, taking into account both the context of use and the limitations of automated processing.

The depth and rigor of review must be proportionate to the intended use, document sensitivity, output condition, and overall residual risk profile. Higher-risk or externally shared outputs should be subject to more thorough scrutiny, while lower-risk internal uses may justify a lighter-touch review approach.

Reviewer sign-off must not operate as a purely procedural or symbolic approval step. For verification to function as an effective governance control, it must be supported by sufficient protected time, appropriate access to source material where necessary, and the use of a review method suited to the specific use case. This may include targeted sampling, full document review, or comparison against source inputs, depending on the level of assurance required. A nominal sign-off without a realistic opportunity to assess the output should not be treated as a strong control.

Following verification, a formal sign-off must be recorded. This includes the identity of the reviewer, the time of approval, and confirmation that the review has been conducted in accordance with internal policy. Where relevant, any limitations, assumptions, or conditions on use must also be documented.

For higher-risk use cases, such as external disclosure or reliance in formal outputs, additional review or dual approval may be required. This reinforces that the system supports data handling processes, but responsibility for interpretation and use remains with the reviewer and approving authority.

9.4 Incident handling and audit preservation runbook

An incident handling and audit preservation runbook must be defined to support a consistent and defensible response to any data handling issue. Incidents may include unintended disclosure of identifiable information, unauthorised access, misconfiguration of processing rules, or improper use of outputs.

When an incident is identified, the immediate priority is containment. This may involve restricting access to affected outputs or suspending processing activities where necessary. All relevant artefacts, including logs, manifests, and output bundles, must be preserved to ensure that the event can be properly investigated.

A clear record of the incident must then be established, including what occurred, when it occurred, and which systems or users were involved. This should be followed by an assessment of impact and identification of affected data. Relevant stakeholders must be notified in accordance with governance requirements. Once the issue has been

understood, corrective actions should be implemented, along with any changes required to prevent recurrence.

Incident handling should not be treated as a purely technical exercise. Where an event may involve personal information, confidentiality, privilege, or court-related obligations, it should be escalated for legal, privacy, compliance, or supervisory assessment as appropriate. This includes, where relevant, assessment against the Notifiable Data Breaches scheme and any applicable professional, client, or court-facing disclosure obligations. This is also consistent with OAIC guidance on the Notifiable Data Breaches scheme, which states that entities covered by the Privacy Act must notify affected individuals and the OAIC when a data breach involving personal information is likely to result in serious harm. (Office of the Australian Information Commissioner, n.d.-a, n.d.-b). This report does not attempt to determine those obligations conclusively as a matter of law, but it treats the need for that assessment as part of the governance response.

Audit artefacts must remain intact throughout this process. Logs should not be altered or overwritten, and access to incident data should be restricted. This approach ensures that incidents can be reviewed, explained, and, where necessary, reported in a defensible manner (NIST, n.d.; Standards Australia & Standards New Zealand, 2023).

9.5 Internal processing policy and system notice

The governance model assumes a short internal operating policy that converts this report's controls into practical rules for use. At minimum, that policy should cover roles and permissions, policy-change control, mapping-register access, retention defaults, incident escalation, training expectations, and prohibition on unauthorised re-identification attempts.

Where the workflow is exposed through an interface, the system should also present a short user notice stating that outputs are pseudonymised rather than guaranteed anonymous, may remain sensitive, are subject to stated metadata limits, must not be treated as final filing or service documents without source verification, and must not be uploaded to unapproved external tools. The internal operating policy may incorporate the implementation checklist in the appendix as a practical control summary for technical staff and administrators.

10. Evaluation and evidence strategy

10.1 Test plan overview and success criteria

Before interpreting evaluation findings, the project establishes baseline measures for the tasks the prototype is intended to support. The purpose is to avoid treating isolated examples or anecdotal impressions as proof of effectiveness. Evaluation is therefore framed against a denominator that can be observed and compared, including

- processing time,
- output completeness,
- reviewer burden,
- residual risk findings,
- and usability observations during controlled use.

At the prototype stage, these measures are intended to show whether the workflow functions consistently and defensibly for its intended low-risk use cases. The initial test plan uses a controlled set of synthetic legal documents designed to reflect the types of material the workflow is expected to handle within scope, namely text-based PDF inputs. Success criteria include

- whether the document can be uploaded, validated, processed, and exported without uncontrolled failure;
- whether pseudonymised outputs preserve basic readability and referential coherence;
- whether the export bundle is complete and internally consistent.

In particular, evaluation should confirm that the safe copy, manifest, event log, and any integrity artefacts remain aligned to the same job identifier, policy version, and processing sequence. Where repeated runs are performed using the same input and unchanged policy settings, the workflow should produce materially consistent results, subject to any expressly documented sources of variation.

Evaluation should include a downstream task degradation check. This should assess whether pseudonymisation materially impairs the intended downstream use by degrading chronology, role relationships, issue structure, or factual coherence. For example, the project may compare task outputs generated from the source document and the safe copy for selected low-risk tasks, while assessing whether the safe copy remains sufficiently accurate and usable for the intended purpose without preserving unnecessary identifying

detail. (International Organization for Standardization, 2023, 2024; Standards Australia & Standards New Zealand, 2023).

10.2 Functional testing results

Functional testing is intended to show that the workflow behaves in a controlled and predictable manner across both ordinary and misuse-oriented scenarios. At minimum, this includes successful ingestion of valid text-based PDF files, rejection of unsupported or malformed inputs, generation of a structured manifest, and production of an export bundle containing the safe copy, audit log, and integrity information.

In addition to normal-path testing, functional evaluation should include governance-relevant abuse cases. These include

- attempting to access upload or export endpoints without authentication and confirming denial plus log generation;
- using a standard user account to attempt administrative actions and confirming privilege enforcement;
- attempting to modify or delete logs without appropriate privilege;
- submitting malformed or oversized files and confirming that the workflow fails safely, logs the event, and prevents partial uncontrolled processing;
- and confirming that failed validation stops the job before it proceeds further into the pipeline.

For evaluation purposes, the significance of these demonstrations is not merely that the system works, but that it fails in a controlled and reviewable way. A rejected upload, denied access attempt, or blocked admin action only supports governance claims if the outcome is observable and recorded. Where relevant, functional testing should also confirm that outputs are not generated when mandatory processing steps fail, and that partially completed jobs are either rolled back or clearly marked as failed rather than left in an ambiguous state (Standards Australia & Standards New Zealand, 2022, 2023).

10.3 Security-oriented tests and controlled demonstrations

Security-oriented evaluation is framed as controlled demonstrations of whether the workflow's stated trust boundaries and governance assumptions hold under realistic use and misuse conditions. The purpose is not to produce a full penetration-testing report or duplicate the backend stream's technical detail, but to generate evidence that the key controls relied upon by this report are observable in operation.

These demonstrations should include testing the separation between input, processing, output, and administrative functions; confirming that raw sensitive values are not written into ordinary logs; checking that temporary files are handled and removed in accordance with the retention model; and confirming that safe-copy outputs remain separated from any privileged or review-sensitive artefacts such as mapping registers or related linkage material. Where the prototype relies on configuration-based protections, testing should focus on whether those protections are actually enforced in observable system behaviour rather than merely declared in settings or documentation. (Australian Cyber Security Centre, 2023; Standards Australia & Standards New Zealand, 2022, 2023).

10.4 Re-identification and disclosure risk evaluation, motivated intruder style check

As part of the evaluation pack, the project should include a basic re-identification risk check aligned to OAIC's emphasis on assessing reasonable likelihood in context. This requires the report to state the assumed recipient environment, including who can access outputs, what auxiliary information is likely to be available, and what governance controls constrain use. The access model should therefore be documented as part of the de-identification result, rather than treated as external to it.

The motivated-intruder baseline in this project should not assume only manual matching or traditional structured-data techniques. It should assume an attacker capable of using contemporary LLM tooling and ordinary auxiliary information to combine residual textual detail with external sources in an attempt to infer identity. The purpose is to document residual disclosure risk honestly and assess whether the implemented controls remain proportionate to the intended environment. This evaluation should also state whether outputs are being assessed for internal controlled use, downstream enterprise use, or broader disclosure conditions, because the same transformed document may present different disclosure risk across those environments. It must assess whether, in the stated access environment and under the assumed auxiliary information conditions, the remaining disclosure risk is proportionate to the intended use of the safe copy (International Organization for Standardization, 2024; OAIC, 2018; O'Keefe et al., 2017).

10.5 Evidence quality approach, what counts as strong evidence

The evaluation gives greater weight to evidence produced by controlled demonstrations, observable system behaviour, and reviewable artefacts than to screenshots or policy assertions alone. In practical terms, stronger evidence includes

- denied-access logs,
- successful and failed processing records,

- manifest-output consistency checks,
- deletion-event records,
- and reproducible test outputs tied to a known policy version or configuration.

Evidence should also be traceable. That is, the report should allow a reviewer to connect a test result back to the relevant input, processing event, policy version, and output artefact without needing to infer missing steps. This supports later review, comparison, and re-testing.

Evaluation is also not treated as a once-off prototype exercise. Any material change to the workflow, policy set, recogniser or model configuration, supported input types, downstream AI destination, access model, or retention approach should trigger targeted re-evaluation of the affected controls and regeneration of the relevant evidence artefacts. In a real deployment, this would sit within periodic internal review and continual-improvement processes rather than being left to ad hoc discretion (International Organization for Standardization, 2023, 2024; Standards Australia, 2018; Standards Australia & Standards New Zealand, 2023).

10.6 Findings, gaps, residual risks, and planned mitigations

Evaluation findings should distinguish clearly between what the prototype demonstrated successfully, what remains only partially tested, and what remains outside the current implementation scope. At minimum, this section should identify whether the workflow operated reliably across the tested inputs, whether audit and evidence artefacts remained complete and internally consistent, and whether the main governance controls were observable under basic misuse scenarios.

Residual risks should then be stated directly. These may include incomplete metadata handling for some formats, false negatives or false positives in entity detection, residual re-identification risk from contextual detail, dependence on correct configuration and reviewer diligence, and any limitations arising from the prototype's simplified access model or non-enterprise deployment assumptions.

Planned mitigations should focus on the next practical steps, such as expanding test coverage, improving policy tuning, strengthening metadata handling, formalising reviewer procedures, and adding more robust access-control and audit features in later iterations. Where a risk is not mitigated within the prototype, the report should say so plainly and identify whether the issue is deferred to later development, organisational governance, or out-of-scope deployment controls (International Organization for Standardization, 2023, 2024; OAIC, 2018; Standards Australia, 2018).

11. Discussion

11.1 What worked, what did not, trade-offs

The report's strongest outcome is that it reframes de-identification as a governed workflow rather than a one-step redaction task. The prototype combines minimisation, role separation, evidence generation, and reviewer oversight in a way that is more defensible than ad hoc document upload to external AI tools. The structured evidence pack, policy traceability, and explicit treatment of mapping artefacts as restricted re-identification material are particular strengths.

The main trade-off is that stronger privacy protection may reduce downstream utility. Pseudonymisation that is too weak may leave individuals reasonably identifiable, while pseudonymisation that is too aggressive may degrade chronology, role relationships, or factual coherence to the point that the safe copy becomes less useful for downstream review or AI-assisted support tasks. The workflow therefore depends on calibrated settings, documented limitations, and human verification rather than on any assumption of perfect automated transformation.

11.2 Limitations, assumptions, and non-claims

This workflow does not assume that pseudonymisation makes downstream LLM use safe in any absolute sense. Contemporary LLMs may assist re-identification where residual contextual detail and auxiliary information remain available. This project accepts a central tension: the same class of systems that may benefit from receiving a pseudonymised safe copy for low-risk downstream tasks may also increase re-identification capability where residual contextual detail and auxiliary information remain available. The workflow is therefore not justified on the basis that pseudonymisation makes LLM use safe in any absolute sense. Its justification is narrower. It is designed to reduce unnecessary direct identifier exposure relative to raw-document use and to support a more reviewable, minimised, and governed form of downstream processing for selected low-risk tasks. That justification holds only where the remaining utility benefit is proportionate to the residual disclosure risk in the intended recipient environment and where downstream use remains subject to task restriction, reviewer oversight, and separate release judgment.

11.3 Legal and governance implications for real deployment

For real deployment, the significance of recent AI governance standards is not that a small legal workflow must become a full AI management system. Rather, it is that responsibility, review, approval, monitoring, and change governance should be made explicit. In this

report, standards such as AS ISO/IEC 23894 and AS ISO/IEC 42001 are therefore used as governance benchmarks and interpretive aids, not as compliance claims. Ongoing oversight could also be informed by broader AI governance frameworks such as the NIST AI Risk Management Framework, particularly for governing tool use, mapping context-specific risks, measuring control effectiveness, and managing changes over time (NIST, n.d.).

More broadly, this report proceeds against Australia's current risk-based regulatory posture, in which AI governance continues to rely primarily on existing legal frameworks, supplemented by targeted reform and practical guidance rather than a standalone AI statute. In that setting, privacy law, professional obligations, court controls, and internal governance remain the principal sources of constraint for legal practice (Australian Government, 2026; National AI Centre, 2025).

The Privacy Act issue is not resolved merely by applying pseudonymisation. Under Australian privacy law, the key question is whether an individual remains identified or reasonably identifiable in the relevant context. That is not determined by text transformation alone. It depends on the content that remains in the document, the amount and specificity of contextual detail, who can access the output, what other information is available to them, and how practicable re-identification would be in the circumstances. In legal-document workflows, those surrounding factors may be substantial because recipients may already possess matter context, client records, internal document histories, or other auxiliary information.

OAIC de-identification guidance is used in this report not as a substitute for the Privacy Act 1988 (Cth), but as the practical framework for assessing when transformed information may no longer be personal information and when it remains reasonably identifiable in context. The relevant legal threshold remains whether an individual is identified or reasonably identifiable, having regard to the content, the recipient environment, other available information, and the practicability of re-identification. For that reason, the report does not treat pseudonymisation as automatically taking output outside the APP framework. Rather, it treats OAIC guidance as a practical method for applying that framework and for supporting controls aligned to APP 6, APP 10, and APP 11 where the output remains sensitive or reasonably identifiable in context.

In practice, real deployment would also need to account more explicitly for privacy risks associated with commercially available AI products, possible cross-border disclosure under APP 8, and incident-escalation obligations under the Notifiable Data Breaches scheme, in addition to the professional and court-facing controls already discussed (Office of the Australian Information Commissioner, 2024; Office of the Australian Information Commissioner, n.d.-a; Privacy Act 1988 (Cth), sch 1 app 8).

12. Conclusion and next steps

12.1 Summary of contribution

This project develops a governance-led, self-hosted de-identification workflow for handling sensitive legal documents in AI-assisted environments. Its central contribution is not merely the transformation of identifiers, but the design of a structured and repeatable process for preparing documents for controlled downstream use. The workflow combines pseudonymisation, metadata handling, auditability, and governance reporting into a single process intended to reduce disclosure risk while preserving enough structure and coherence for practical tasks such as summarisation, chronology-building, and extraction. In that sense, the project treats de-identification not as a one-step redaction exercise, but as a governed workflow shaped by trust boundaries, review requirements, and release conditions. A key contribution of the project is demonstrating how technical processing can be supported by governance controls. These include evidence packs, verification requirements, and governance-facing reporting outputs that allow a reviewer to understand what was processed, how it was handled, under which settings, and with what remaining limitations. This addresses a gap in current practice, where documents may enter AI-enabled workflows without consistent handling, verification, or traceability.

12.2 Deployment readiness notes

The prototype provides a baseline facilitating controlled deployment in a legal practice environment, particularly under a self-hosted model. However, its use depends on organisational context and supporting controls beyond the prototype itself. Deployment would require alignment with existing systems and processes, including document management, access control, and retention policies. It also requires clear allocation of responsibility for review and approval, as well as appropriate user training to ensure the workflow is applied consistently. The workflow does not eliminate disclosure risk or guarantee anonymisation. Outputs may still contain identifiable, confidential, or otherwise sensitive information depending on context, residual detail, and the recipient environment. The system therefore does not replace professional judgment, legal review, or organisational governance. Its role is narrower. It exists entirely to reduce unnecessary identifier exposure and support a more defensible, reviewable, and controlled form of downstream document use.

12.3 Future work roadmap

Further development should focus on both technical improvement and operational maturity. On the technical side, this includes improving detection accuracy for more

complex or context-dependent identifiers, strengthening handling of metadata and format-specific artefacts, preserving legal intelligibility where citations or case names should remain intact, and extending support across a wider range of document types and layouts.

On the operational side, future work should focus on stronger evaluation and integration. This includes more structured approaches to residual-risk assessment, more rigorous testing of re-identification and disclosure scenarios, clearer event-log and evidence-pack consistency, and closer integration with enterprise systems such as document management, identity infrastructure, and eDiscovery platforms. Additional governance features, including stronger release controls, clearer reviewer workflows, and more mature policy enforcement, would also improve real-world deployability. Overall, the project demonstrates a practical way to combine technical and governance controls in response to current risks in legal document handling. As AI-assisted workflows become more common, approaches that emphasise minimisation, auditability, reviewability, and controlled downstream use are likely to become increasingly important.

13. References

13.1 Standards

International Organization for Standardization. (2018). *Information technology—Security techniques—Privacy enhancing data de-identification terminology and classification of techniques* (ISO/IEC Standard No. 20889:2018). https://www.iso.org/standard/69373.html

International Organization for Standardization. (2023). *Information technology—Security techniques—Guidelines for privacy impact assessment* (ISO/IEC Standard No. 29134:2023). https://www.iso.org/standard/45123.html

International Organization for Standardization. (2024). *Information technology—Security techniques—Privacy framework* (ISO/IEC Standard No. 29100:2024). https://www.iso.org/standard/45124.html

Standards Australia. (2018). *Risk management—Guidelines* (AS ISO 31000:2018). https://www.standards.org.au/standards-catalogue/standard-details?designation=as-iso-31000-2018

Standards Australia. (2022). *Privacy information management systems—Requirements and guidelines in relation to ISO/IEC 27001 and ISO/IEC 27002* (AS ISO/IEC 27701:2022). https://www.standards.org.au/standards-catalogue/standard-details?designation=as-27701-2022

Standards Australia. (2023a). *Information technology—Artificial intelligence—Guidance on risk management* (AS ISO/IEC 23894:2023). https://www.standards.org.au/standards-catalogue/standard-details?designation=as-iso-iec-23894-2023

Standards Australia. (2023b). *Information technology—Artificial intelligence—Management system* (AS ISO/IEC 42001:2023). https://www.standards.org.au/standards-catalogue/standard-details?designation=as-iso-iec-42001-2023

Standards Australia & Standards New Zealand. (2022). *Information security, cybersecurity and privacy protection—Information security controls* (AS/NZS ISO/IEC 27002:2022). https://www.standards.org.au/standards-catalogue/standard-details?designation=as-nzs-iso-iec-27002-2022

Standards Australia & Standards New Zealand. (2023). *Information security, cybersecurity and privacy protection—Information security management systems—Requirements*

(AS/NZS ISO/IEC 27001:2023). https://www.standards.org.au/standards-catalogue/standard-details?designation=as-nzs-iso-iec-27001-2023

Standards Australia & Standards New Zealand. (2024a). *Information security, cybersecurity and privacy protection—Guidance on managing information security risks* (AS/NZS ISO/IEC 27005:2024). https://www.standards.org.au/standards-catalogue/standard-details?designation=as-nzs-iso-iec-27005-2024

Standards Australia & Standards New Zealand. (2024b). *Privacy enhancing data de-identification framework* (AS/NZS ISO/IEC 27559:2024). https://www.standards.org.au/standards-catalogue/standard-details?designation=as-nzs-iso-iec-27559-2024

13.2 Government and institutional guidance

Allens. (2023, July 12). *Generative AI in practice: A guide for in-house legal teams*. https://www.allens.com.au/insights-news/insights/2023/07/generative-ai-in-practice-a-guide-for-in-house-legal-teams/

Asia Pacific Privacy Authorities Technology Working Group. (2025, June). *Guide to getting started with anonymisation*. https://www.pcpd.org.hk/english/resources_centre/publications/files/appa_anonymisation_guide072025.pdf

Australian Bureau of Statistics. (2021, November 8). *Data confidentiality guide*. https://www.abs.gov.au/about/data-services/data-confidentiality-guide

Australian Cyber Security Centre. (2023, November). *Essential Eight maturity model*. https://www.cyber.gov.au/sites/default/files/2023-11/PROTECT%20-%20Essential%20Eight%20Maturity%20Model%20%28November%202023%29.pdf

Australian Government. (2026, April 1). *Australian Government response to the Senate Select Committee on Adopting Artificial Intelligence (AI) report*. Department of Industry, Science and Resources. https://www.industry.gov.au/publications/australian-government-response-senate-select-committee-adopting-artificial-intelligence-ai-report

Law Society of New South Wales. (2026, January). *A solicitor's guide to responsible use of artificial intelligence*. https://www.lawsociety.com.au/sites/default/files/2026-01/20260109%20LS4816_PSU_GuidetoAI_2026-01-08a%20FINALv2.pdf

Law Society of New South Wales, Legal Practice Board of Western Australia, & Victorian Legal Services Board and Commissioner. (2024, December 6). *Statement on the use of artificial intelligence in Australian legal practice*. https://www.lsbc.vic.gov.au/news-updates/news/statement-use-artificial-intelligence-australian-legal-practice

National AI Centre. (2025, October). *Guidance for AI adoption: Implementation practices*. https://www.industry.gov.au/publications/guidance-for-ai-adoption

National Institute of Standards and Technology. (n.d.). *NIST AI RMF Playbook*. AI Risk Management Framework Resource Center. https://airc.nist.gov/airmf-resources/playbook/

Office of the Australian Information Commissioner. (2018). *De-identification and the Privacy Act*. https://www.oaic.gov.au/privacy/privacy-guidance-for-organisations-and-government-agencies/handling-personal-information/de-identification-and-the-privacy-act

Office of the Australian Information Commissioner. (2024). *Guidance on privacy and the use of commercially available AI products*. https://www.oaic.gov.au/privacy/privacy-guidance-for-organisations-and-government-agencies/guidance-on-privacy-and-the-use-of-commercially-available-ai-products

Office of the Australian Information Commissioner. (n.d.-a). *Notifiable data breaches*. https://www.oaic.gov.au/privacy/notifiable-data-breaches

Office of the Australian Information Commissioner. (n.d.-b). *Part 4: Notifiable Data Breach (NDB) Scheme*. https://www.oaic.gov.au/privacy/privacy-guidance-for-organisations-and-government-agencies/preventing-preparing-for-and-responding-to-data-breaches/data-breach-preparation-and-response/part-4-notifiable-data-breach-ndb-scheme

O'Keefe, C. M., Otorepec, S., Elliot, M., Mackey, E., & O'Hara, K. (2017). *The de-identification decision-making framework* (Report No. EP173122). CSIRO. https://doi.org/10.4225/08/59c169433efd4

Queensland Law Society. (2024, September). *AI companion guide: The QLS overview guide to AI for solicitors*. https://www.qls.com.au/content-collections/guides/ai-companion-guide

Queensland Law Society, Law Institute of Victoria, Law Society of South Australia, Law Society of Western Australia, Law Society of the Australian Capital Territory, Law Society Northern Territory, & Law Society of Tasmania. (2026, February). *AI selection and use checklist for legal practitioners*.

https://www.qls.com.au/getmedia/b3bf168b-8374-496a-bfc3-92b0ea0c4296/2026-FINAL-RISK-ASSESSMENT-TOOL-CON-BODS-AI-IN-LEGAL-PRACTICE-CONSULTING-COMMITTEE.pdf

Queensland Law Society, & QLS Ethics and Practice Centre. (2024, October 24). *Guidance statement no. 37: Artificial intelligence in legal practice*. https://www.qls.com.au/practising-law-in-qld/ethics/guidance-statements/no-37-artificial-intelligence-in-legal-practice

Victorian Law Reform Commission. (2025, October). *Artificial intelligence in Victoria's courts and tribunals report*. https://www.lawreform.vic.gov.au/publication/artificial-intelligence-in-victorias-courts-and-tribunals-report/

13.3 Court materials

Federal Court of Australia. (2024). *Guide to the anonymisation of personal and sensitive information*. https://www.fedcourt.gov.au/online-services/preparing-documents-for-the-court/guide-to-the-anonymisation

Federal Court of Australia. (2025, April 29). *Notice to the profession: Artificial intelligence use in the Federal Court of Australia*. https://www.fedcourt.gov.au/law-and-practice/practice-documents/notice-to-profession/29-april-2025

Federal Court of Australia. (2026, April 16). *Use of generative artificial intelligence practice note (GPN-AI)*. https://www.fedcourt.gov.au/law-and-practice/practice-documents/practice-notes/gpn-ai

Supreme Court of New South Wales. (2010, March 15). *Identity theft prevention and anonymisation policy*. https://supremecourt.nsw.gov.au/documents/Practice-and-Procedure/policyidentity_theft_prevention.pdf

Supreme Court of New South Wales. (2025). *Practice note SC Gen 23: Use of generative artificial intelligence*. https://supremecourt.nsw.gov.au/documents/Practice-and-Procedure/Practice-Notes/general/current/PN_SC_Gen_23.pdf

Supreme Court of Queensland. (2025). *Practice direction number 5 of 2025: Accuracy of references in submissions*.

https://www.courts.qld.gov.au/about/news/news233/2025/supreme-court-practice-direction-5-of-2025-accuracy-of-references-in-submission

13.4 Academic and industry sources

Croft, L. (2025, November 4). Report: Australian firms outpacing global counterparts in AI use. *Lawyers Weekly*. https://www.lawyersweekly.com.au/biglaw/43257-report-australian-firms-outpacing-global-counterparts-in-ai-use

Garat, D., & Wonsever, D. (2022). Automatic curation of court documents: Anonymizing personal data. *Information*, 13(1), Article 27. https://doi.org/10.3390/info13010027

Kolochenko, I. (2024, December 9). How to protect your law firm's data in the era of GenAI. *Business Law Today*. https://www.americanbar.org/groups/business_law/resources/business-law-today/2024-december/how-protect-law-firm-data-era-gen-ai/

Lermen, S., Paleka, D., Swanson, J., Aerni, M., Carlini, N., & Tramèr, F. (2026). *Large-scale online deanonymization with LLMs* [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2602.16800

Microsoft. (n.d.). *Learn Presidio. Microsoft Presidio*. Retrieved March 21, 2026, from https://microsoft.github.io/presidio/learn_presidio/

Nguyen, L. D. (2023). A brief overview of digital forensics. In M. M. Houck (Ed.), *Encyclopedia of forensic sciences* (3rd ed., Vol. 1, pp. 19–25). Elsevier. https://doi.org/10.1016/B978-0-12-823677-2.00237-3

Pasupuleti, M. K. (2024). *AI in business law: Legal frameworks for the digital era*. National Education Services. https://doi.org/10.62311/nesx/rb978-81-981662-0-3

Centre for the Future of the Legal Profession. (2025, November). *GenAI, fake law and fallout: A review of the misuse of generative artificial intelligence in legal proceedings* [Report]. https://www.unsw.edu.au/news/2025/11/genai-fake-law-and-fallout-report

13.5 Legislation

Evidence Act 1995 (Cth), s 122

Evidence Act 1995 (NSW), s 122

Legal Profession Uniform Conduct (Barristers) Rules 2015 (NSW).
https://www.legislation.nsw.gov.au/view/html/inforce/current/sl-2015-0243

Legal Profession Uniform Law 2014 (NSW).
https://legislation.nsw.gov.au/view/html/inforce/current/act-2014-16a

Legal Profession Uniform Law Australian Solicitors' Conduct Rules 2015 (NSW).
https://legislation.nsw.gov.au/view/html/inforce/current/sl-2015-0244

Privacy Act 1988 (Cth). https://www.legislation.gov.au/C2004A03712/latest/text

Appendices

A. Standards and guidance mapping tables

A.1 31000 mapping table

ISO 31000 element	Report location	Project artefact / evidence
Scope, context, and criteria	Sections 0, 2, 5, 6.3	Scope statement, deployment assumptions, trust zones, risk criteria
Leadership, accountability, and integration into governance	Sections 4.2.1, 4.5, 8.1, 8.6	Assigned roles, admin controls, training expectations, reviewer responsibilities
Communication and consultation	Sections 4.2.2, 8.6, 9.5	Internal operating policy, training guidance, user notice
Risk identification	Sections 6.1, 6.2, 6.4	Threat model, attacker profiles, disclosure scenarios
Risk analysis and evaluation	Sections 6.3, 6.5, 10.4	Priority risks, residual-risk discussion, motivated-intruder style check
Risk treatment	Sections 7, 8, 9	Requirements, access controls, retention, metadata handling, downstream controls, reporting
Monitoring and review	Sections 4.2.2, 4.6, 9.4, 10, 11.3	Review triggers, incident runbook, re-evaluation after change, findings and planned mitigations
Recording and reporting	Sections 7.4, 7.7, 9.1–9.2, 10.5	Manifest, audit log, governance-facing report, release/share register

A.2 29100 mapping table

ISO/IEC 29100 theme	Report location	Project artefact / evidence
Privacy terminology and concepts	Sections 1.1–1.4, 4.3.1	Definitions and glossary
Controller, processor, third party roles	Sections 1.2, 4.3.1, 4.5	Role mapping and deployment model
Pseudonymisation vs anonymisation	Sections 1.1, 1.4, 3.4, 4.3.1	Safe-copy framing, non-claims, report disclaimers

Metadata as a privacy issue	Sections 1.1, 1.3, 3.5, 7.5	Metadata handling requirements and reporting
Minimisation, disclosure limitation, accountability, security	Sections 2.5, 4.3.1, 7, 8, 9	Controls, access restrictions, audit trail, release conditions

A.3 27559 mapping table

ISO/IEC 27559 theme	Report location	Project artefact / evidence
Structured de-identification process	Sections 4.3.3, 5.1, 7	Workflow design, requirements, evidence pack
Access/share/release environment	Sections 5.5, 8.5, 9.2, 10.4	Environment analysis, release conditions, downstream-use controls
Threat modelling and disclosure scenarios	Sections 6.1–6.5	Assets, attacker profiles, abuse cases, disclosure scenarios
Transparency and auditability	Sections 7.4, 7.7, 9.1, 10.5	Audit log, manifest, governance-facing report
Recipient-side safeguards and restrictions	Sections 8.5, 9.2, 10.4	Contractual controls, release/share register, recipient conditions
Evidence-based review of protection adequacy	Sections 10.2–10.6	Controlled demonstrations, residual-risk findings, mitigation plan

A.4 27001 benchmark-level mapping table

This table identifies where the report reflects ISO/IEC 27001 themes at a benchmark level only and does not imply conformity with an ISMS standard.

ISO/IEC 27001 benchmark area	Report section / project artefact
Scope and context	Sections 0, 2, 5.2, 8.4
Roles, responsibilities, and governance	Sections 4.2.1, 4.5, 5.4, 8.1
Risk assessment and treatment	Sections 4.2, 6, 7, 8

Documented information and traceability	Sections 4.6, 7.4, 7.7, 9.1–9.2; manifest, audit log, release register
Operational control and access separation	Sections 5.3–5.4, 7.3, 8.1–8.4, Appendix G
Performance evaluation, review, and improvement	Sections 9, 10, 11.2–11.3

A.5 29134 mapping table

ISO/IEC 29134 theme	Report location	Project artefact / evidence
Threshold-style privacy justification	Sections 0.2, 4.3.2, 4.4	Purpose/scope and PIA-style justification
Description of PII flows	Sections 4.4, 4.5, 5.1, 5.5	PII-flow description, trust-boundary model
Privacy impact and risk analysis	Sections 6, 10.4	Threat model, disclosure scenarios, residual-risk assessment
Safeguards and treatment planning	Sections 7, 8, 9	Requirements, controls, reviewer gates, downstream safeguards
Reporting, review, and follow-up	Sections 9, 10, 11	Governance report, incident handling, evaluation and planned mitigations

A.6 OAIC/APPA/CSIRO mapping table

Guidance theme	Report location	Project artefact / evidence
De-identification is contextual, not absolute	Sections 1.1, 1.4, 3.4, 4.3.4, 10.4	Safe-copy framing, residual-risk reporting
Re-identification risk depends on environment and available data	Sections 5.5, 6.4, 10.4	Environment analysis, motivated-intruder style check
Safe settings / safe people / safe outputs logic	Sections 4.3.5, 5.3–5.5, 8, 9	Trust zones, access controls, release controls, reviewer sign-off

Ongoing management, not one-off transformation	Sections 4.2.2, 4.6, 9.4, 10.5	Review triggers, change control, re-evaluation

A.7 AI governance benchmark mapping table

AI governance overlay	Report location	Project artefact / evidence
ISO/IEC 23894: AI risk interpreted in context	Sections 4.2.3, 5.5, 6, 10.4, 11.3	Change-triggered review, downstream environment analysis, residual-risk evaluation
ISO/IEC 42001: governance, operational control, performance evaluation, continual improvement	Sections 4.2.3, 4.6, 8, 9, 10, 11.3	Documented roles, controls, reporting, evaluation, review and mitigation

B. Condensed risk register table

This appendix provides a concise governance-facing summary of the project's priority risks and primary treatments. It is not a substitute for any fuller technical risk documentation maintained in the backend or development streams.

Risk ID	Risk	Why it matters here	Primary treatment
R1	Unauthorised access to uploads/outputs	Raw and safe-copy materials remain sensitive	RBAC, authentication, restricted exposure, audit logging
R2	Mapping register exposure	Enables re-identification	Separate storage, admin-only access, exclusion from ordinary outputs
R3	Metadata leakage	Hidden identifiers may persist	Metadata stripping, limitation warnings, reviewer checks
R4	Uncontrolled retention of raw/temp files	Raw identifying material may persist beyond intended window	Defined retention model, deletion events, isolated temp storage
R5	Audit gaps or tampering	Undermines defensibility and incident review	Append-only logging, restricted log access, traceable evidence pack
R6	Downstream platform exposure	Safe copy enters a separate risk environment	No raw upload, contractual controls, release conditions, reviewer approval

C. Minimum audit event requirements

This appendix identifies the minimum governance-facing audit events expected for the prototype. Its purpose is to show the minimum event coverage needed to support traceability, review, and incident reconstruction. It does not replace any fuller backend logging schema or implementation-specific log design.

Event type	Minimum fields	Notes
Upload	timestamp, user, file ID or hash, file type, size, outcome	Do not log raw content
Validation	timestamp, job ID, file ID or hash, outcome, rejection reason where applicable	Supports safe rejection and review of failed inputs
Processing start / end	timestamp, job ID, policy version, outcome	Must link to manifest and processing run
Export generation	timestamp, job ID, bundle ID or hash, outcome	Supports evidence-pack traceability
Download / access	timestamp, user, bundle ID or job ID, outcome	Supports reviewable access trail
Deletion	timestamp, artefact type, job ID, outcome	Raw, temporary, output, or mapping artefact as applicable

D. Example evidence pack structure

This appendix provides a concise illustration of the evidence-pack components referred to throughout the report. It is illustrative only. Filenames, packaging format, and internal directory structure may vary, provided the required artefacts, traceability, and separation controls are preserved.

Component	Purpose	Included by default	Sensitivity
Safe copy	Risk-reduced downstream document	Yes	Sensitive
Manifest	Run metadata, policy version, output inventory, key configuration details	Yes	Sensitive
Event log	Audit trail of processing and access events	Yes	Sensitive
Governance-facing report	Human-readable summary of what was processed, what was changed, and what limitations remain	Yes	Sensitive
Integrity artefacts	Hash or checksum verification for later integrity checking	Optional	Sensitive
Mapping register	Re-identification key linking original identifiers to substitutions	Optional, restricted only	Highly sensitive

Where a mapping register exists, it should be stored separately from ordinary downstream outputs and not bundled by default with broader-circulation artefacts.

E. Operator checklist and templates

This appendix provides lightweight operator-facing and reviewer-facing tools derived from the governance controls set out in this report. Its purpose is to support consistent handling, review, escalation, and controlled downstream use within the prototype's intended operating model. These checklists and templates are illustrative rather than mandatory enterprise workflow documents. They are included to show how the report's governance position could be operationalised in day-to-day use while reinforcing the core controls of the prototype: no raw external upload, safe-copy sensitivity by default, controlled handling of mapping artefacts, metadata awareness, reviewer sign-off, and traceable release where applicable.

E.1 Pre-processing operator checklist

The following checklist is intended to be completed or confirmed before a document is processed through the workflow.

Check	Operator confirmation	Notes / escalation prompt
The input file is within supported prototype scope.	☐	If unsupported, do not proceed. Escalate or reject.
The file appears to be the intended source document and not the wrong matter file, wrong version, or duplicate upload.	☐	Confirm matter context before processing.
Processing is occurring inside the declared controlled environment.	☐	No raw or identifiable matter documents are to be sent to external tools as part of ordinary processing.
The intended purpose of processing is known.	☐	Example: internal summarisation, chronology preparation, low-risk AI-assisted analysis.
The intended downstream environment, if any, is identified.	☐	Record whether the output is for internal use only, downstream enterprise platform use, or another controlled disclosure context.

The correct de-identification policy or profile has been selected.	☐	Confirm policy name/ID and relevant settings before starting the job.
The operator understands that the output will be a risk-reduced safe copy, not an anonymous document.	☐	Safe copies remain sensitive by default.
Known metadata-handling limits for the file type are understood.	☐	If metadata removal is incomplete or unsupported, this must be disclosed later in the report.
Mapping-register handling for this run is understood.	☐	Confirm whether a mapping register is generated, how it is stored, and that it is excluded from ordinary downstream output by default.
The operator is authorised to perform the processing run.	☐	Use must occur under the applicable role/access model.
Retention and deletion defaults for raw uploads and temporary files are understood.	☐	Confirm whether any exception to default deletion is configured and justified.
Any obvious reason to stop or escalate has been considered.	☐	Examples: unsupported scanned input, visible confidentiality concern, wrong downstream use, uncertain policy choice, suspected privileged sensitivity requiring additional review.

E.2 Post-processing and reviewer checklist

This checklist is intended to support human review before reliance-critical use or external disclosure.

Check	Reviewer confirmation	Notes
The job completed successfully and is not marked failed or ambiguous.	☐	If failure occurred, do not treat any output as complete or reliable without review of the failure state.

The expected evidence pack artefacts are present.	☐	At minimum: safe copy, manifest, event log, governance-facing report.
Job identifiers and version information are internally consistent across artefacts.	☐	Check job ID, timestamps, and policy/version references.
The safe copy remains readable and structurally usable for the intended purpose.	☐	Confirm referential coherence and basic task usability.
Direct identifiers appear to have been handled in accordance with policy.	☐	Note any obvious misses or suspicious replacements.
No obvious indirect identifier combination appears to create an immediate and unreasonable disclosure risk in the intended environment.	☐	This is a practical reviewer judgment, not a guarantee.
The metadata-handling statement is present and understandable.	☐	Confirm whether metadata was removed, transformed, or remains limited.
The limitations and residual risk statement is present.	☐	Output must not be treated as fully anonymous.
The governance-facing report clearly states that the safe copy remains sensitive by default.	☐	Check report disclaimer language.
Any mapping register is excluded from the ordinary output bundle unless specifically authorised.	☐	If it exists, confirm restricted handling.
The intended downstream use remains appropriate.	☐	Confirm alignment between output condition, sensitivity, and planned use.
Release or external disclosure, if proposed, is justified and documented.	☐	Complete the release/share register where required.

Reviewer sign-off has been recorded.	☐	Include reviewer identity, time, and any conditions on use.

Additional prompts for higher-risk review

Where the intended use involves external disclosure, filing, advice, or another reliance-critical purpose, the reviewer should also consider:

- whether the safe copy is sufficient for the task without expanding disclosure unnecessarily
- whether further minimisation is possible
- whether a second reviewer or supervisor should approve release
- whether the receiving environment has been assessed and is appropriately controlled
- whether any contractual, policy, or client-authorisation condition must be satisfied first

E.3 Templates

The following templates are intentionally short. They are designed to support consistent documentation rather than replace fuller backend or operational records.

E.3.1 Governance-facing report minimum header template

The following template identifies the minimum header-level fields that should appear in a governance-facing report. It is intended as a structural prompt only. A fuller illustrative example appears in Appendix F.

Field	**Minimum content**
Job ID	Unique processing-run identifier
Date and time	Processing completion timestamp
Operator	User or processing account initiating the run

Input type	Supported file type processed
Policy version	Policy name or ID and version or hash
Processing summary	Short statement of what was processed and what transformation approach was applied
Metadata statement	Whether metadata handling was applied, limited, or unsupported
Mapping-register status	Whether a mapping register exists and that it is restricted from broader-circulation output
Residual-risk statement	Short statement that the output remains a risk-reduced safe copy and may still contain sensitive or identifiable information depending on context
Reviewer sign-off	Reviewer identity, date, and approval status where required

E.3.2 Example release and share register template

Field	**Entry**
Release/share record ID	
Job ID	
Export bundle ID	
Date/time of release	
Approving person	
Releasing person	
Intended recipient	
Recipient environment	Internal / Enterprise platform / Court / Other
Purpose of release	

Output type released	Safe copy / Governance report / Other
Mapping register exists?	Yes / No
Mapping register included in release?	Yes / No
Metadata handling applied?	Yes / No / Limited
Policy version	
Reviewer approval required?	Yes / No
Reviewer identity	
Confirmation that no raw or identifiable source document was disclosed	Yes / No
Conditions imposed on recipient	
Notes / exceptions	

E.3.3 Short user notice template

This workflow produces pseudonymised safe copies for controlled use. Outputs are not guaranteed anonymous and may remain sensitive depending on context. Metadata handling may be limited for some formats. Outputs must not be treated as final filing, service, or advice material without source verification and reviewer approval where required. Raw or identifiable matter documents must not be uploaded to unapproved external tools.

E.4 Suggested use within the project

For prototype purposes, these tools may be used in three ways:

- as operator prompts during controlled demonstrations
- as reviewer prompts during output verification and evaluation
- as practical examples of how the report's governance controls could be operationalised in a real deployment

F. Sample court or governance report output

This appendix provides an illustrative example of the governance-facing report produced as part of the prototype's evidence pack. Its purpose is to show the kind of information the workflow is expected to present in a human-reviewable format, not to prescribe a single mandatory layout or file format. The example is indicative only. In implementation, the exact layout, styling, file naming, and rendering format may vary, provided the required fields, disclaimers, and traceability elements are preserved.

The report is intended to support review, accountability, and defensibility by showing what was processed, under which policy settings, what transformations were applied, what limitations remain, and whether the output was approved for its intended use.

F.1 Example governance-facing report

Governance Report: De-identification Processing Summary

1. Processing summary

- **Job ID:** JOB-2026-04-18-0017
- **Export bundle ID:** BNDL-2026-04-18-0017
- **Processing date/time:** 18 April 2026, 14:42 AEST
- **Operator:** alex.harb
- **Reviewer:** reviewer01
- **Input file type:** Text-based PDF
- **Input file count:** 1
- **Input file size:** 1.8 MB

2. Policy and configuration

- **Policy name/ID:** LEGAL-DEID-DEFAULT
- **Policy version:** v0.9.3
- **Policy hash:** 8c1d7a2f...
- **Relevant runtime settings:** default confidence threshold applied
- **Recogniser/model/profile used:** Presidio default legal profile
- **Metadata handling mode:** supported metadata removal applied for in-scope fields
- **Mapping register generated:** Yes
- **Mapping register included in ordinary output bundle:** No
- **Mapping register storage status:** restricted artefact, separately controlled

3. Input description

The processed input consisted of a text-based legal document within prototype scope. The document was accepted, validated, and processed inside the declared self-hosted controlled environment. No raw source material was disclosed to any external platform as part of this processing run.

4. Transformation summary

The workflow applied policy-driven transformation to reduce direct identifier exposure and preserve basic usability for controlled downstream use.

Illustrative transformation counts

- Names replaced: 14
- Contact details removed/transformed: 6
- Identifier strings masked or removed: 5
- Address/location fields generalised or removed: 4

5. Output artefacts generated

The following artefacts were generated for this job:

- Safe copy
- Manifest
- Event log
- Governance-facing report
- Integrity artefact(s), where applicable

Ordinary output classification: sensitive

6. Metadata handling statement

Metadata handling was performed for supported fields and supported format features within current prototype scope. This processing reduces, but does not guarantee elimination of, metadata-related disclosure risk. Reviewers should not assume that all possible hidden fields, embedded layers, or format-specific metadata have been removed in every case unless explicitly confirmed by the workflow and testing evidence.

7. Limitations and residual risk statement

This output is a risk-reduced safe copy, not a guaranteed anonymous document. The workflow reduces disclosure risk but does not guarantee that the output is no longer

personal information or no longer confidential in all contexts. Residual risk may remain because of:

- contextual detail
- unusual combinations of indirect identifiers
- metadata limitations
- false negatives in automated detection
- the access environment into which the output is later disclosed or used

This report does not replace human judgment, legal review, source verification, privilege review, or organisational policy controls. The suitability of the output for any downstream use remains context-dependent.

F.2 Minimum fields reflected in the sample

The sample above reflects the minimum reporting expectations described in the main report. At minimum, a governance-facing report should include:

- job identifier
- processing time
- operator and, where applicable, reviewer
- policy name/version and relevant configuration reference
- input description
- high-level transformation summary
- metadata-handling statement
- limitations and residual risk statement
- safe-copy sensitivity disclaimer
- mapping-register status
- intended-use / downstream-use statement

- reviewer sign-off or review outcome
- traceability references to manifest, audit log, and related artefacts

F.3 Notes on implementation

This sample is deliberately written in a governance-facing style rather than as a backend schema. In implementation, the report may be rendered as HTML, PDF, or another human-readable format. The exact presentation may differ, but the content should remain sufficiently complete for a reviewer to understand:

- what was processed
- what was changed
- what remains uncertain
- what controls or restrictions still apply
- whether the output was approved for its intended use

G. Governance implementation checklist

This appendix translates the report's governance position into the practical controls, handling rules, and evidence expectations that a reader should take away and implement for the prototype. It is intended to function as the report's operational takeaway. It does not restate every recommendation made elsewhere in the report and does not imply organisational certification, production readiness, or full legal sufficiency in all contexts. Its purpose is to identify the baseline governance measures that should be implemented, enforced, or clearly documented if the prototype is to remain aligned with the report's risk, privacy, auditability, and trust-boundary assumptions.

Area	Implementation expectation	Minimum evidence / observable outcome
Deployment boundary	Raw source documents are processed only inside the declared self-hosted or firm-controlled environment. No raw or identifiable matter documents are sent to external platforms during ordinary processing.	Deployment description, network exposure description, and system behaviour consistent with internal-only raw processing.
Scope control	Only supported file types and supported processing modes within the declared prototype scope are accepted. Unsupported, malformed, corrupt, or oversized files are safely rejected.	Observable rejection behaviour; failed validation recorded in logs or failure records.
Validation stop	Failed validation stops the job before extraction, transformation, or export. Invalid files do not proceed into ordinary processing.	No downstream artefacts for invalid jobs except failure records.
Trust-zone separation	Input, processing, output, and administrative functions are separated in a way visible in system behaviour and deployment structure.	Controlled demonstrations showing separation of ordinary processing from admin-only functions and restricted artefacts.
Authentication	Authentication is required for upload, export, and administrative functions.	Access-denial tests and authentication event records.

Role separation	Ordinary processing access is separated from privileged administrative access. Sensitive functions are restricted to authorised users only. Reviewer access is distinct where review is part of the operating model.	Standard-user attempts at admin actions are denied and logged; reviewer and admin functions are distinguishable.
Approval authority	Higher-risk downstream use, especially disclosure to an external AI or other external recipient environment, requires documented approval by the role authorised to accept residual governance risk.	Reviewer or approving-authority record, approval field, or equivalent traceability supporting later review.
Policy governance	De-identification policies, recogniser settings, thresholds, and related processing rules are treated as controlled governance artefacts.	Policy name/ID, version/hash, and relevant settings recorded in manifest or logs.
Change control	Material policy or configuration changes are restricted to authorised roles and are traceable. Material changes trigger review of affected assumptions and evidence where relevant.	Change records showing who changed what and when; version traceability across runs.
Pseudonymisation behaviour	Policy-driven transformation is applied consistently within the defined scope and preserves basic readability and reference coherence where required.	Test outputs or reviewer evidence showing consistent replacements and usable safe-copy structure.
Safe-copy status	Safe copies are treated as risk-reduced but still sensitive by default. They are not presented as anonymous, unrestricted, or outside privacy, confidentiality, or professional obligations.	Report text, labels, and output handling show that the output may still be sensitive after processing.
Utility preservation	The safe copy preserves sufficient structure, coherence, and referential integrity for the intended low-risk downstream task within the declared scope.	Controlled evaluation or reviewer evidence showing that the safe copy remains usable for the intended task.

Evidence pack	Each successful job produces a structured evidence pack, not just a transformed file.	Safe copy, manifest, event log, and governance-facing report are all present and aligned.
Manifest consistency	The manifest records job ID, timestamps, policy/configuration details, and output inventory consistently.	Internal consistency across manifest, outputs, and logs.
Governance-facing report	Each successful job produces a human-reviewable report stating what was processed, what was changed, metadata handling status, and remaining limitations or risks.	Report includes required fields and supports reviewer understanding and sign-off.
Audit logging	Core events are logged, including upload, validation, processing start/end, export generation, download, deletion, authentication, and privileged actions.	Chronological event records exist without exposing raw sensitive values in routine logs.
Log protection	Audit records are protected from unauthorised modification, deletion, or silent overwrite.	Log-preservation behaviour and restricted log access observable in design or testing.
Raw and temporary data	Raw uploads, extracted text, temporary artefacts are stored in defined locations and removed according to the retention rules.	Deletion behaviour is documented, reviewable, and evidenced through logs or test outcomes.
Retention model	Default retention minimises persistence of raw and temporary identifying material. Any departure from default deletion is deliberate, documented, and justified.	Retention settings or documentation show default deletion posture and any exceptions.
Mapping register	Any mapping register or substitution table is treated as a confidential re-identification key, stored separately, and excluded from ordinary downstream outputs by default.	Access restrictions, separation from ordinary bundle contents, and clear disclosure in report or evidence.

Metadata handling	Metadata is handled for supported formats where feasible, or limitations are clearly disclosed. Visible-text transformation is not treated as sufficient where metadata risks remain.	Test files or evidence showing metadata removal, transformation, or an explicit notice of remaining risk.
Failure behaviour	Mandatory processing failure prevents ordinary successful outputs and results in a controlled, reviewable failed state.	Failed jobs are clearly marked, logged, and do not produce misleading success artefacts.
Security baseline	Services run with minimal privileges, network exposure is minimised, and deployment assumptions for patching, maintenance, and backups are documented.	Deployment notes and observable restrictions consistent with the stated security baseline.
Downstream external use	External AI or other external downstream use is treated as disclosure into a separate recipient environment and remains a residual-risk activity rather than a presumptively safe step.	Report, policy, or release record identifies recipient environment, conditions of use, and reviewer/approving-authority input where required.
Recipient conditions	Where a safe copy is disclosed externally, conditions such as purpose limitation, no raw upload, no re-identification, and handling restrictions are stated where applicable.	Release/share record, governance-facing report, or equivalent documentation reflecting recipient conditions.
Reviewer support	Outputs remain reviewable by a human before use that affects advice, filing, or external disclosure. The report and artefacts support meaningful review, not just formal sign-off.	Safe copy remains readable and coherent for review; reviewer outcome and any conditions on use are recorded.
Incident readiness	The workflow and supporting records support escalation, reconstruction, and review of suspected disclosure, misuse, or misconfiguration events, including cases that may raise legal, privacy, confidentiality, or professional issues.	Incident-related artefacts can be preserved and linked to the relevant job, user, policy version, and affected outputs.

Release traceability	Evidence-pack identifiers and key fields are sufficient to support downstream release/share recording and later reconstruction.	Bundle ID, timestamps, policy version, metadata status, and reviewer/approval-related fields are available for tracing disclosure.
Evaluation evidence	Controls relied on by the report can be demonstrated through observable system behaviour and reviewable artefacts, not just claims.	Denied-access logs, failed-input records, manifest/output consistency checks, and retention/deletion evidence are available.
Non-claims	The prototype is not represented as guaranteeing anonymisation, removing legal duties, or substituting for human review, legal advice, or a full organisational governance program.	Report wording, disclaimers, and user-facing statements remain consistent with the prototype's limited status.

G.1 How this should be read

This checklist is intended to be the report's practical control baseline. It should be read together with the main report, where a fuller explanation is needed, but it is designed to stand on its own as a compact statement of what the prototype is expected to enforce, produce, preserve, and avoid claiming. Where implementation details are supplied in the backend and development streams, those materials should be read as the technical realisation of the governance expectations stated here

Subject: 31261 Internetworking Project
Autumn 2026

BACKEND REPORT

Privacy-preserving self-hosted de-identification workflow for sensitive legal documents

---

Project supervisor	Dr Yi Zhang
Legal supervisor	Dr Susanne Lloyd-Jones
Technical supervisor	Runsong Jia
Owners	Ishaan Pathak, 24963593, Architecture / Evaluation Lori Basmajian, 24563090, Architecture / Evaluation Nicholas Hatzidimitriou, 25471398, Implementation / Development Adrian Martinez, 25494636, Implementation / Development
Contributors	Alex Harb, 24715841, Governance / Legal Ajay Yuvaraj, 24934301, Governance / Legal
Project / working title	Privacy-preserving self-hosted de-identification workflow for sensitive legal documents

Backend Design Report

Executive Summary

This report offers the backend design for the de-identification of sensitive legal documents.The system is designed around a security-first architecture which prioritises strict trust separation, controlled data movement, auditability, and protection against both external and insider threats. The backend workflow divides the system into discrete trust zones instead of handling document de-identification as a straightforward upload and redact process. Every zone has a defined security role, storage boundary, and responsibility. This guarantees that raw personally identifiable information (PII) is never sent through the pipeline unnecessarily and is only handled when absolutely necessary.

In legal and sensitive document scenarios where data sovereignty, secrecy, and governance are key considerations, the backend design provides a private, onsite delivery model. Nevertheless, there are additional cybersecurity concerns associated with this deployment architecture, such as malicious file uploads, corrupted PDFs, service denial efforts, privileged insider usage, unauthorised interference with audit logs, and unintentional release of documents that have not been sufficiently redacted. The system uses a number of defensive design principles to mitigate these hazards. PDF-only ingestion, strict validation gates, SHA-256 hashing, quarantine paths, structured audit logging, isolated storage, least-privilege access, separation of duties, and future support for enhanced security controls like host-based intrusion detection, MFA, encryption at rest, and SIEM-compatible event logging are some of these.

The overall goal of the backend architecture is to provide an auditable, defensible, and traceable document processing pipeline. Sensitive papers are handled through controlled phases, with full documentation of what was processed, denied, redacted, and authorised for distribution. This is its primary security assurance. Because of this, the system can be used as both a technical prototype and a governance aligned procedure for the responsible handling of sensitive data.

1. Introduction

To understand the necessity of such a solution, one needs to understand the context of the issue. Legal documents contain highly sensitive information such as personal names, addresses, contact details,case references, health information and one's financial situation. Information like this is crucial to protect as it can expose individuals to private, legal or reputational harm if information is disclosed incorrectly. Due to this, the de-identification process must provide a secure, controlled and auditable process that handles sensitive information appropriately from the moment a file is uploaded to the moment it is released.

The technological architecture of a privacy de-identification process for sensitive legal documents is described in this backend design report. The goal of the project is to develop a self-hosted system capable of ingesting PDF documents, validating them, extracting text, identifying sensitive information, producing outputs that are pseudonymized, and producing accompanying audit records. By guaranteeing that every step of the workflow is precisely recorded and traceable, the system is built to fulfil both technical processing and governance needs.

Cybersecurity is one of the design's main concerns. The system must presume that all uploaded files are potentially harmful, corrupted, large, encrypted, or purposefully created to take advantage. The system needs to take into account internal threats, such as privileged users trying to access mapping registers, alter audit logs, get around approval procedures, or distribute documents without proper review. Due to this, the backend is designed around trust zone separation. The workflow is divided into six zones: ingestion, validation, processing, redaction, output generation and admin review/release. Each zone has its specific responsibility where it is only given the necessary information to perform its function. The purpose of this is to limit the exposure of raw PII and limit the damage that would occur if the system is compromised. The goals of this report is to outline the backend architecture, justify the key design choices, and show how the system facilitates safe document de-identification. It also describes how hazards like malicious uploads, unauthorised access, and dangerous document release are handled by the design.

1.1 Threat Model & Security Context

The sensitivity of the documents being processed and the fact that the workflow is meant to run in a self-hosted, on-premises environment influence the system's threat model. In this scenario, the system needs to protect itself from both trusted insiders and external attackers. Although the risks posed by these two threat groups vary, they both

have the potential to compromise the de-identification process's confidentiality, integrity, and trustworthiness.

1.2 On-premises deployment threat landscape

The system is an on premises deployment, due to the sensitive nature of the legal documents. They would not be suitable to process through a third-party cloud server. This approach provides safer and stronger data freedom and provides the organisation with direct control over where documents,audit logs and copies are stored. Nonetheless, this comes with its own security challenges. The server, file systems, logs and storage directories may be accessible to administrators and staff.Access controls need to be strong, in order to prevent a privileged user from browsing directories, altering files and deleting file logs. As a consequence, the system cannot rely solely on standard application security or network isolation. Application, filesystem, audit, and governance security must all be enforced.

1.3 External attacker surface: upload boundary, malformed files, resource exhaustion

1.3.1 Upload Boundary:

The upload boundary is the primary external attack surface. Any user with the ability to upload a document may do so with a file that is encrypted, disguised, malformed, or purposefully designed to take advantage of the processing engine. An attacker might, for example, upload a file intended to deplete server RAM, rename a non-PDF file with a.pdf extension, or submit a PDF that causes unexpected behaviour in the text extraction library. As a result, the validator is regarded as the initial security barrier in the system. Before dangerous files are allowed to reach the processing zone, they must be rejected. Given this, the design incorporates strict file size checks, SHA-256 hashing, PDF sub-validation, MIME type validation using magic bytes, and quarantine management. These safeguards reduce the possibility that content under attacker control may get to the redaction engine in a dangerous condition.

1.3.2 Malformed files and Resource Exhaustion:

Since the backend must parse uploaded documents before it can redact them, malformed files pose a significant danger. A seemingly harmless file could be corrupt, unreadable, encrypted, or built in a way that prevents the extraction process from working. The system may create a false safe copy in which there has been no actual redaction if it accepts such files without the necessary validation. Resource exhaustion is another important risk. To use up memory, CPU, storage, or audit log capacity, an attacker may try

to upload a huge number of files or numerous smaller files in a short time. While future limiting the rate and queue restrictions would assist control repeated or automated upload attempts, the file size gate helps lower this risk at the individual file level.

1.4 Insider threat model: privileged access, log tampering, self-approving releases

Considering the system handles sensitive legal data and is expected to work in a setting where certain users may have authorised access to the server or workflow, the insider threat is particularly significant in this project. A system administrator, operator, reviewer, or other privileged user who abuses their access could be considered an insider.

Potential insider threats include:

- Accessing raw documents outside the approved workflow
- Modifying or deleting audit logs
- Approving their own submitted document for release
- Extracting the mapping register to re-identify pseudonymised individuals
- Moving files between zones manually to bypass validation
- Releasing an unsafe or incomplete redacted output

Separation of roles, segregated zone storage, structured audit logs, limited access to mapping registers, and planned protections like encryption at rest, MFA, host-based intrusion detection, and tamper-evident recording are all included in the architecture to mitigate these risks.

1.5 Trust assumptions: what the system trusts, what it doesn't

The uploaded files are never trusted, according to the system's design. Until they are verified, file names, extensions, MIME claims, file content, and document structure are all regarded as under attacker control. Furthermore, the system makes the assumption that raw PII should only be present in the bare minimum of locations needed for processing. Certain internal components are trusted by the system, but only to a certain extent. For instance, the audit system is trusted to document workflow events, the redaction engine is trusted to apply configured entity detection and privacy logic, and the validator is trusted to enforce file acceptance requirements. Zone borders, structured outputs, and access control expectations still place restrictions on these components. Logs must never

include raw document content, which is a key principle. Logs are helpful for traceability, but can provide a second source of data leakage if they contain sensitive data . As a result, rather than recording raw PII, the audit design records metadata like timestamps, job IDs, hashes, event outcomes, and failure codes.

1.6 Why standard application security is insufficient here – PII sensitivity multiplier

For this system, standard web application security measures like input validation, login systems, and basic access control are inadequate. This is due to the fact that the system processes extremely sensitive papers, and even a minor error could reveal private, confidential, or legal information. This indicates that, in comparison to a standard file upload application, the dangers are significantly larger. In a typical system, for instance, a little logging problem might just reveal technical details, but in this instance, it might unintentionally reveal the entire text of a legal document that contains personal information. The system requires enhanced, multi-layered security as a result. This involves splitting the workflow into trust zones, restricting access to only what is required, requiring strict file validation, maintaining thorough audit logs, isolating storage, and managing the review and release of outputs. Not only must documents be processed correctly, but each step must be safe, traceable, and compliant in the event that something goes wrong.

2. System Architecture Overview

The privacy de-identification workflow is carefully designed around the principle of strict trust zone separation, allowing our team to emphasise the security of the whole pipeline. Here, rather than treating the system as a single application that simply receives a file and returns a result, the architecture deliberately breaks the journey into six sequential zones, each with clear, defined responsibilities, its own storage, and enforced boundaries on what data may enter or leave it. The core design principle that we utilised here is simple: no raw PII must ever be passed forward in its original form. Each zone receives only what it needs to do its job, and hands off only what the next zone is permitted to see, allowing the safe transformation and movement of data through the pipeline without exposing sensitive information beyond its required boundary. For example, a file that enters zone 1 as an untrusted upload will never be read directly by zone 5 and so on. The data each file contains will be thoroughly validated, processed, redacted, and audited before any form of output is ever produced. This is not just an implementation detail; rather, it is the foundational guarantee of security and anonymity that the system provides. If any zone were to expose raw sensitive content to a downstream zone through logs, shared memory, or uncontrolled file paths, that guarantee would be broken.

2.1 The Six Zones and Their Responsibilities

The workflow is divided into six zones, each representing a distinct stage in the document lifecycle:

1. **Ingestion Zone (Untrusted Input)**
   Receives uploaded files and treats all input as potentially malicious. No assumptions are made about file integrity or format at this stage.

2. **Validation Zone**
   Applies strict validation rules to ensure only compliant files proceed. Files that fail validation are rejected, logged, and quarantined.

3. **Processing Zone**
   Handles text extraction and prepares the document for de-identification. This is the last stage where raw content is handled directly.

4. **Redaction Zone**
   Identifies and removes or replaces sensitive entities using the redaction engine. Output from this zone is considered sanitised.

5. **Output Generation Zone**
   Produces the safe-copy document and associated artefacts such as audit records and metadata.

6. **Admin Review and Release Zone**
   Introduces human oversight before final release, ensuring correctness and compliance before outputs leave the system.

Each zone operates independently, with controlled interfaces between them. Direct access across non-adjacent zones is not permitted, reinforcing isolation and preventing unintended data exposure.

2.2 Key Design Principle

"*No raw PII crosses zone boundaries*"

This principle dictates all data movement in the system. It can be said to be a strict control on the handling of sensitive information at every step of the process. There should be no unnecessary exposure of raw PII data in any of the zones in which it is used. As the data moves through the process, it is systematically minimised, transformed, or completely removed depending on the purpose of the next step in the process. For instance, identifiable data can be completely removed or replaced through pseudonymization. This process of minimising data through the process ensures that the system never stores any data beyond what is absolutely necessary. This can be said to be in line with the principle of least privilege. Each zone only receives the bare minimum of data that it needs to function, which significantly reduces the possible damage of both external attacks and internal misuse.

Aside from the main process of handling data, the principle is also applied to all other artefacts that support the process of the system. These artefacts include log data, temporary files, and other intermediate data structures. These artefacts are designed in such a way that they never store any raw data. Instead, they store only the bare minimum of metadata, such as file hashes, process outcomes, timestamps, and error codes. This prevents the log from becoming a second source of data exposure, which is a common oversight in most systems. This process of minimising data through the process, as mentioned earlier, reduces the overall risk of the system at every step of the process, thus making the process completely traceable and auditable.

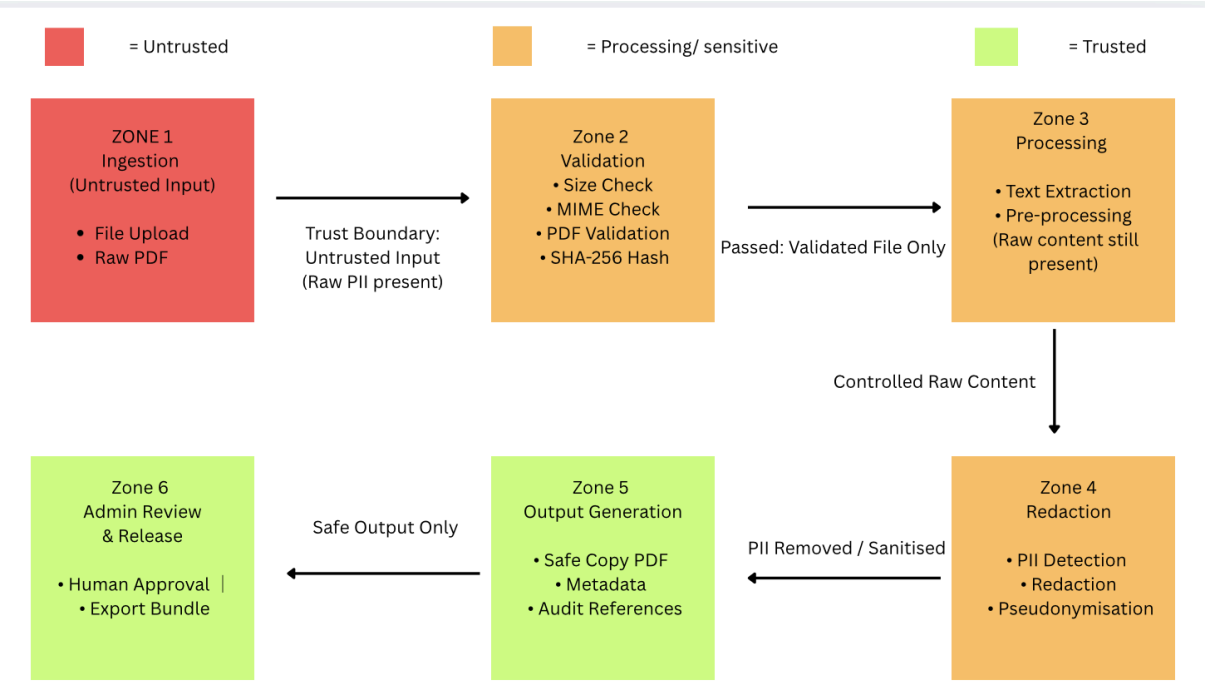


Diagram 1: *Trust Zone Architecture Diagram - The Trust Zone Architecture*

Diagram shows how uploaded legal documents move through six controlled security zones, from untrusted input to validated processing, redaction, and final safe release. The design reduces exposure to raw PII by enforcing trust boundaries and allowing only sanitised content to progress through the workflow.

3. Folder Structure and Storage Separation

3.1 The privacy-deid-workflow/ directory

The privacy-deid-workflow/ folder is designed in such a way as to accurately represent the trust zone structure of the system. Rather than placing all artefacts in one common location, the system categorises them into separate folders according to their function within the workflow and their relative sensitivity.

3.2 Why each zone has its own storage

At each step of the pipeline, isolated storage exists, which ensures strict access to and isolation of data. In particular, this isolation avoids unwanted information exposure and makes sure that every element of the system works exclusively with data it was meant to. For instance, ingestion directory is supposed to store unverified files, whereas processed directory can have unredacted raw texts. They should be inaccessible to those zones which are associated with output and releases. Redacted outputs which belong to the safe-copy directory also cannot be accessed by those zones that process internal artifacts like logs and mapping registers.

The described approach also implies least privilege policy when every zone only accesses what is required. Moreover, isolation of storage reduces the probability of lateral movement. Even in case one directory becomes compromised, others remain inaccessible. Another benefit of isolating storage is that it simplifies auditability. Every directory corresponds to a certain phase of the document life cycle, and it becomes easier to determine the course of events and detect a possible issue with files.

3.3 Governance note: *Safe-copy, audit files, and mapping registers are not treated as equally sensitive separated by design*

However, the sensitivity level associated with the artefacts created within the framework of the system is not equal, which would increase the risk unnecessarily if the data were treated as an equivalent threat level. For this reason, it was necessary to segregate artefacts depending on their security class.

Artefacts contained in the safe-copy directory are already redacted, ready to be distributed to the outside world. They are safer because any personal data that could directly refer to an individual was removed. In contrast, the register of mappings can be considered one of the most dangerous parts of the system since it may contain connections between the real identifier and pseudonymized value. If such information leaks to an untrusted party, then it will be possible to re-identify individuals. Therefore, this part of the system requires the highest level of control and will never be included in the output data. Audit data are more dangerous than safe copies but are less risky than the register of mappings because they do not contain the document itself but other important information about its handling. With these artefacts being separated out, the system will be able to use different control measures for each of them, including:

- Limiting access to highly sensitive artefacts (such as mapping register)
- Permitting access to safe outputs where appropriate
- Ensuring appropriate retention of artefacts based on their function
- Preventing tampering with audit logs

In this way, governance issues can be addressed without exposing sensitive artefacts.

Diagram 2: *Privacy-Workflow*

This diagram presents the secure folder structure used in the privacy-preserving de-identification workflow. Each directory represents a different trust zone based on the sensitivity of the data it contains, separating raw uploads, validation failures, processing data, redacted outputs, audit logs, and final release files. The structure enforces strict handling boundaries to minimise exposure to raw PII and support secure, auditable document processing throughout the system lifecycle.

```
privacy-deid-workflow/
│
├── input/ Untrusted Zone
│   └── uploaded_files/
│       → Raw user uploads
│       → Contains original PDFs with raw PII
│
├── quarantine/ Restricted Zone
```

```
│   └── rejected_files/
│       → Failed validation (malformed/malicious)
│       → Retained for investigation only
│
├── processing/ Sensitive Zone
│   ├── extracted_text/
│   │   → Raw extracted content (PII still present)
│   │   → Must not leave processing boundary
│   │
│   └── temp/
│       → Intermediate working files
│       → Short-lived but sensitive
│
├── redaction/ Controlled Zone
│   ├── safe_copy/  Sanitised Output
│   │   → Redacted documents (no raw PII)
│   │   → Lower sensitivity, ready for output stage
│   │
│   └── mapping_register/ Highly Sensitive
│       → Links original ↔ pseudonymised data
│       → Enables re-identification if exposed
│
├── audit/  Operational Zone
│   ├── validation_logs/
│   └── processing_logs/
│       → Metadata only (hashes, timestamps, results)
│       → No raw PII, but still sensitive
│
```

└── release/ Trusted Zone

└── export_bundle/

→ Final approved outputs

→ Safe for controlled external release

3.4 End-to-End Processing Flow

The flow of processes from start to finish governs how documents get processed in the system, starting from their upload all the way to their release. Every step in the pipeline guarantees trust boundaries, logs audit trails, and restricts data transfer to only the bare minimum. This process ensures that data can be safely minimized and yet still fully tracked throughout.

3.4.1 Step-by-step walkthrough: upload → validation → processing → redaction → output → review → release

The process starts with uploading the file to the system. In that regard, the document is considered untrusted at this moment and might contain malicious content, corrupt data, or raw PII. Then comes the validation step during which the document goes through a series of stringent validations including but not limited to validation of file size, MIME type check by analyzing magic bytes, integrity checks of the PDF, as well as generation of its SHA-256 hash. Any failure triggers immediate stop of the processing, and the file is moved to quarantine. Also, an entry in the structured log is created explaining the rejection cause.

Upon passing all validations, the processing step is executed. That includes extraction and sanitization of the document's text. The processing step is tightly isolated since at this stage we work with raw content. Afterwards, the sanitized content goes through the redaction step where the sensitive entities are identified and subjected to either redaction or pseudonymization. Redaction represents the stage when the raw PII transitions to sanitised one.

Finally, we produce an output bundle. The output contains the safe copy of the document and corresponding metadata. Meanwhile, any sensitive internal artifacts like mapping registers should be left in isolation. The bundle is then subjected to administrative review, where a human operator ensures that the processing process has been carried out effectively. This is an extra safeguard against the release of faulty information. The final result is then released once it has been successfully reviewed. At this stage, only sanitized information is released.

3.5 What is logged at each step and what is passed forward

At each level, the process writes down audit information structured and ensures the data that can be passed into the next step:

Upload Level

- Recorded: file name, timestamp, job identifier
- Passed through: original file

Validation Level

- Recorded: file size, discovered MIME type, SHA-256 digest, success/failure flag, failure reasons
- Passed through: validated file (in case of success)

Processing Level

- Recorded: entering the level, job ID, processing status
- Passed through: extracted text (highly sensitive)

Redaction Level

- Recorded: found entities, events of redacting (metadata only)
- Passed through: sanitized text (without original PII data)

Output Level

- Recorded: creating a package, metadata for output
- Passed through: safe copy of the document

Admin Approval Level

- Recorded: user actions, decision regarding approval
- Passed through: approved package only

Release Level

- Recorded: releasing timestamp, overall status
- Passed through: final output released

Crucially, no un-sanitized document is ever recorded to audit logs, ensuring that they do not become another exposure channel themselves.

3.6 Zone handoff protocol – what data moves and in what form

The transfer of data from one zone to another occurs through a well-defined handover process. It guarantees that:

- Only the essential amount of data is transferred from one zone to another
- Data is first sanitized before being transferred out of the zone it originated from
- There is no direct communication allowed between non-contiguous zones

For instance, only raw files can be accessed in the ingestion and processing zones, whereas only sanitized files can be accessed in other zones, including release zones.

3.6.1 Error and quarantine paths

When an item is rejected due to validation errors, it will not continue moving down the pipeline. On the other hand:

- It will be diverted to the quarantine folder
- A properly structured log message will be written with a proper failure code and description
- The item will not undergo any further processing

This ensures that any suspicious or bad input is blocked from taking up the system's resources or advancing to critical stages of the process. This diagram below shows the step-by-step workflow of how documents are securely handled within the de-identification system. Uploaded PDF files first go through validation checks before moving into text extraction and PII redaction stages. Once sensitive information has been removed, the system generates audit records and prepares a safe output bundle for administrative review and final release. Files that fail validation are redirected to quarantine, helping prevent unsafe or malicious content from entering the main processing pipeline

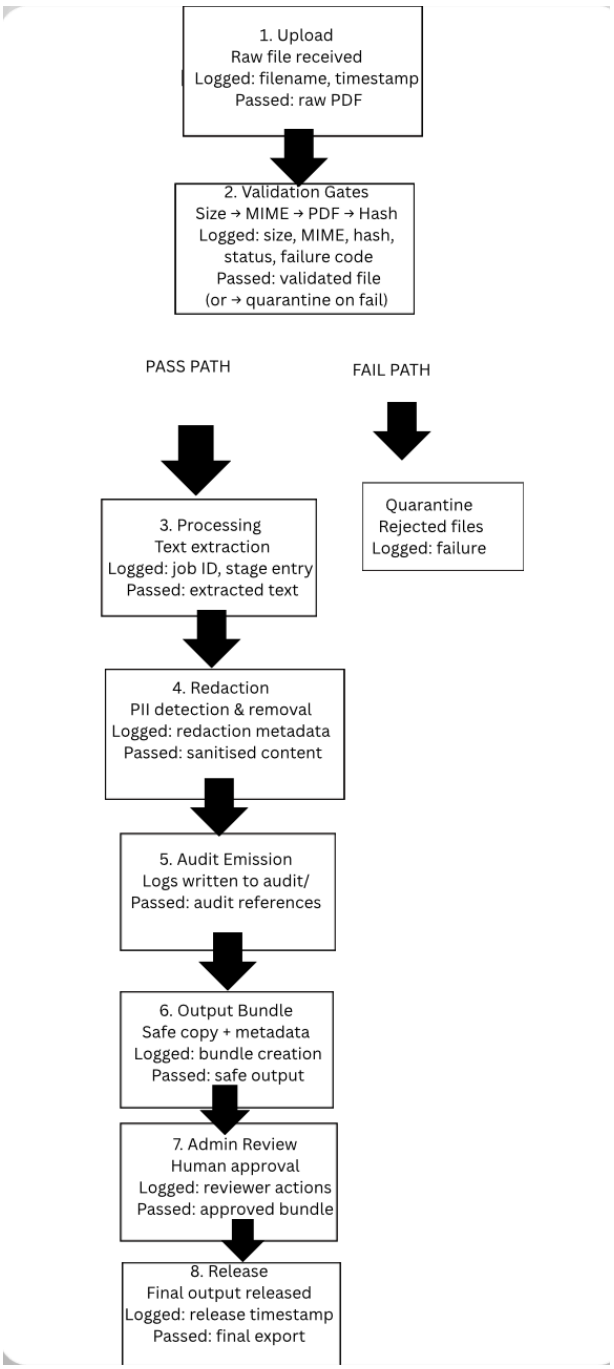


Diagram 3: ***End-to-End Processing Flow***

3.7 Ingestion Validator – Logic Design

3.7.1 Purpose of the validator

The ingestion validator was designed to be the system's first and most critical line of defence against malicious and malformed input. In any given system that accepts file uploads from external or semi-trusted sources, the upload boundary is one of the

highest-risk attack surfaces in the entire architecture. For example, files can be attacker-controlled inputs; here, the name, size, extension, and content can all be theoretically manipulated before submission. Therefore, we have implemented the use of a file validator, where its primary objective would be to ensure that nothing the attacker controls is trusted implicitly. This is not a simple convenience check, but rather a well-defined security boundary. The validator essentially sits at the entry point between the untrusted input zone and the rest of the redaction engine pipeline, and its role is to enforce that only files that meet a precise, explicitly defined specification criteria are ever permitted to proceed. Every file that does not meet that specification is then rejected, logged with a structured record and quarantined. This designer structure follows a foundational principle in defence security, where validation of input occurs at the trust boundary, not inside the trusted zone. Thus, by the time a file reaches the processing engine, it should be structurally impossible for that file to be the wrong type, wrong size, unreadable, or encrypted.

3.7.2 PDF-only decision rationale

During the planning phase of this project, both PDF and DOCX files were considered as supported formats; however, later on, the DOCX support was dropped, and this decision has a security rationale beyond the implementation complexity. DOCX files are essentially ZIP archives; here, magic-byte checking tools such as libmagic frequently return application/zip instead of the correct DOCX MIME type, which means a reliable magic-byte check would require a secondary inspection step. Where we'd have to open the zip archive and verify its internal structure for further processing. This creates a validation path where the system is partially parsing through a potentially attacker-controlled archive content before it has definitively confirmed if the file is legitimate.

ZIP-based formats also tend to carry their own class of vulnerabilities. For example, a zip bomb, which is a small zip archive that expands to an enormous size upon extraction, holds the potential to exhaust memory or disk space even during the validation process itself if the archive is opened natively. Furthermore, path traversal attacks embedded in archive entry names can cause files to be written outside their intended directories if the extraction logic does not sanitise the paths carefully. Given that we are looking at an on-premises deployment of our solution, where the server's file system would be directly accessible to insiders, a successful path traversal attack during the validation process can pose serious risks, potentially allowing a crafted DOCX to write artefacts into directories it should never reach. Restricting the system support to only PDF eliminates this entire class of archive-based attacks from the validation surface. Hence, the decision is recorded here as a deliberate security scoping choice, and not an oversight.

3.7.3 The four gates in detail:

The four gates refer to the validator running four sequential gates in a straight fail-safe order. Here, no gate runs on a file that has already failed at a previous gate; this sequencing in itself is a tactical security decision. This concept is further discussed in the fast-fail section below.

Gate 1 – The File Size Check	This gate runs before any file content is read into the memory. This is done to secure the validator from various attacks, as a native implementation that loads first and checks size second is vulnerable to a straightforward resource exhaustion attack. Where an attacker could submit files large enough to exhaust server memory across concurrent requests, producing a denial of service without needing any special privileges. Two conditions are enforced: *1. Zero-byte files are rejected (FILE_EMPTY), and* *2. Files exceeding 20MB are rejected (FILE_TOO_LARGE).* The 20MB limit is inclusive and applied at a single enforcement point; inconsistently applied size limits are a known bypass vector.
Gate 2 – Magic Bytes Check	File extensions are attacker-controlled and generally trivially changed. At this gate, the magic bytes check uses python-magic to read the actual binary signature embedded in the file's content, ignoring the extension entirely. This helps us defend against file masquerading, which revolves around the idea of renamed executables, web shells, or malformed files crafted to trigger parser vulnerabilities in the redaction engine. In an on-premises deployment, a crafted file that causes the processing engine to execute arbitrary code has direct access to the server and thus can potentially access the broader internal network. One condition is enforced:

	1. Only *application/pdf* is accepted. Failure codes: INVALID_MIME, UNSUPPORTED_FORMAT.
Gate 3 – PDF Sub-Validation	An encrypted PDF tends to pass all previous gates but yields nothing when the text extractor runs; the system appears to succeed while redacting an empty stream. An image-only (scanned) PDF has the same outcome; here, the presence of no text layer meant no redaction occurs, but the output is presented as a processed safe copy. In either case, a document containing sensitive content exists in the pipeline, looking clean when nothing was actually done to it. In a deliberate scenario, both are exploitable by an insider who understands the system's behaviour, and both are rejected here before the processing zone ever sees the file. Failure codes: PDF_ENCRYPTED, PDF_NO_TEXT_LAYER.
Gate 4 – SHA-256 Hashing	This gate is designed in a manner that allows it to always complete and never reject any file. It computes a cryptographic fingerprint of the raw submitted bytes before any transformation from the redaction engine occurs; here, a tamper-evident receipt of exactly what arrived is kept. In an on-premises deployment where a privileged insider could modify files on the file system after submission, this hash is the reference point that exposes any such tampering. The file is read once, and those bytes are reused, while multiple reads introduce a time-of-check to time-of-use (TOCTOU) vulnerability, where the file could be modified between reads, causing the hash to differ from the bytes that were actually inspected at the earlier gates. A single read eliminates this window entirely. The hash is recorded on both a file's pass and fail on the validator check. A rejected file's fingerprint may match known malware signatures and always ensures the audit record is complete.

3.7.4 Fail-fast boundary rationale

When a gate fails, the validator stops and returns an error code immediately, and it does not continue running subsequent gates on a file that has already been rejected. This is a deliberate security posture, where running additional gates on adversarial input would give an attacker incremental feedback about which parts of their submission passed and which did not, effectively telling them how to improve their next attempt. It also produces audit records where multiple gates have partial results, making the authoritative reason for rejection harder to determine, and when looked at through an insider threat scenario, a muddier audit trail is easier to argue about or dismiss. Beyond the security reasoning, there is no legitimate purpose served by continuing to process a file that has already failed a hard boundary check. Also emphasising that every gate that runs unnecessarily is work done in the service of a rejected submission, wasting both time and resources. One file, one failure code, one clear cause, that is the contract that our validator seeks to enforce.

3.7.5 Structured result object schema

Every invocation of the validator returns a structured result object with the following fields, regardless of whether the file passed or failed:

- **status**: "**pass**" or "**fail**". This is the top-level signal consumed by the pipeline controller.
- **Failure_code** → a named code from the defined set if rejected, **null** on pass. Named codes are used deliberately rather than free-text messages as they are queryable, aggregatable, and monitorable. For example, a spike in **INVALID_MIME** rejections over a short window is a detectable signal that someone may be probing the upload boundary.
- **failure_reason** → a human-readable explanation paired with the failure code, intended for operator review and incident investigation.
- **sha256_hash** → always present, always computed over the raw submitted bytes before any transformation.
- **detected_mime** → what libmagic actually reported, not what the file extension claimed. This is the ground truth about the file's type and is recorded as such.
- **file_size_bytes** → the raw byte count of the submitted file.
- **filename** → recorded exactly as submitted, without normalisation or sanitisation in the log. Filenames themselves can carry forensic significance; their path traversal sequences, injection strings, and encoded payloads are sometimes embedded in filenames by attackers targeting systems that parse them unsafely. Preserving the raw value means that evidence is not lost and can be examined to search for patterns later on.

3.7.6 Failure log schema

Each rejection event appended to the validation log would capture the following:

1) timestamp,
2) job_id,
3) original_filename,
4) file_size_bytes,
5) detected_mime,
6) sha256_hash,
7) failure_code, and
8) failure_reason.

The **job_id** field is particularly important, as it ties every log entry back to the job manifest, meaning a single submission can be traced across all audit records from the moment it arrived to wherever it was quarantined. In an incident investigation, this is the thread that connects everything.

Raw file content and any extracted text are never written to the log under any circumstances. This is not just good practice, but in an on-premises deployment where log files may be accessible to a wider set of privileged users than the application itself, a log that contains raw document content becomes a secondary source of sensitive data exposure. The log records what arrived and why it was rejected; however, what the file contained is not the log's concern.

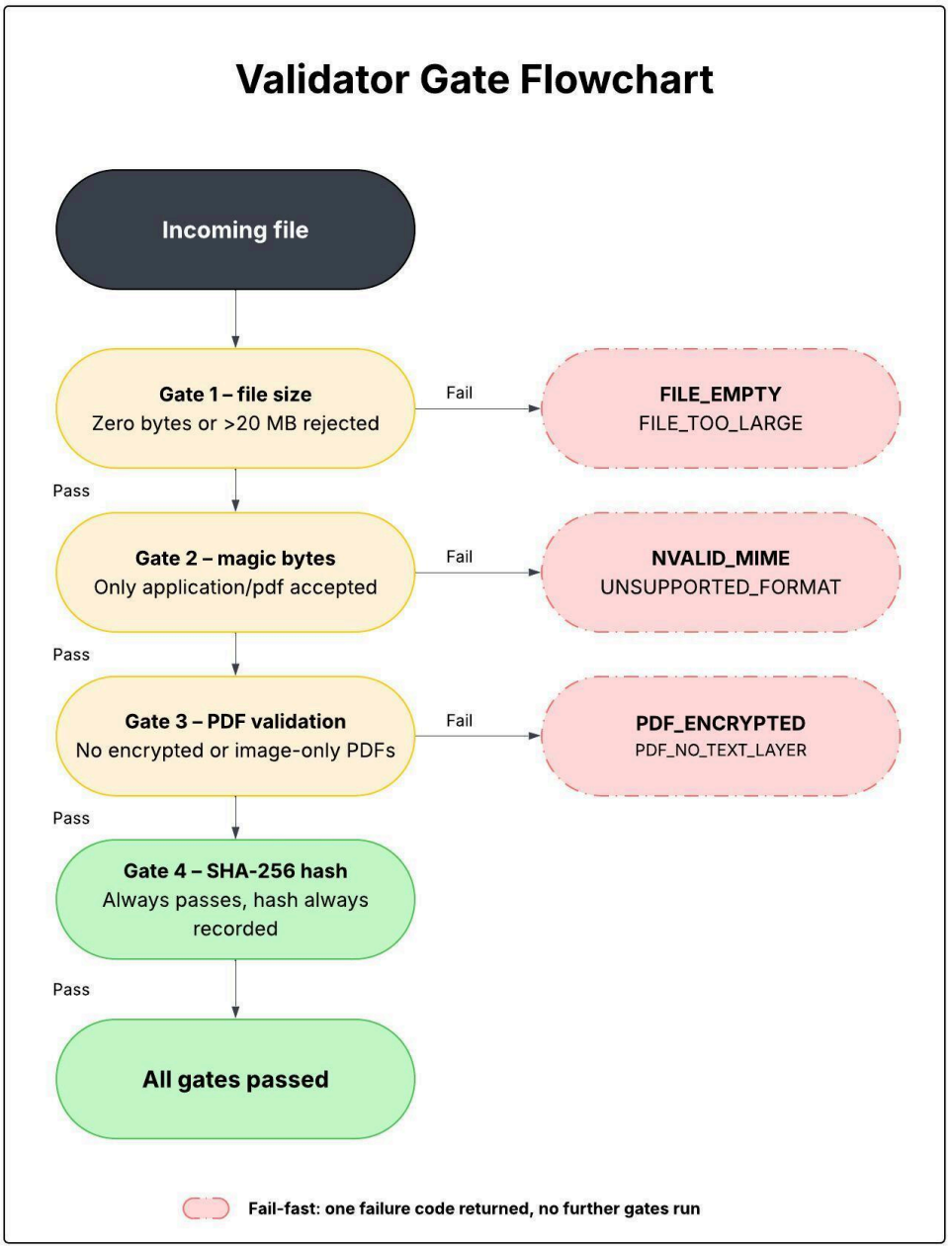


Diagram 4: *Validator Gate Decision Flow*

This flowchart illustrates the sequential validation process used within the system, where each of the validation gates evaluates the processing request against a specific integrity and governance condition. The diagram highlights both pass and fail branches, and the single successful execution path that proceeds through all four validation gates before processing is permitted.

4. Redaction Engine – Presidio

4.1 Why Presidio was selected over alternatives (Philter)

During the technology evaluation phase, two primary redaction engines were considered, these being; Philter and Microsoft Presidio. Philter was initially explored due to its purpose-built nature for PHI/PII redaction; however, it presented a critical compatibility issue when we delved deeper into its functionality. Philter is implemented in Java and lacks a reliable, well-maintained Python port; given that the project stack is Python-first, integrating Philter would have introduced unnecessary cross-language complexities and overhead that is not appropriate within the scope and timeline of this capstone project.

On the other hand, Microsoft Presidio was selected as the redaction engine for several reasons. First, it is natively Python-based, making it a natural fit for the existing codebase with no interoperability overhead. Second, Presidio benefits from extensive official documentation and an active open-source community, reducing onboarding friction and debugging time, allowing our dev team to spend the same time bettering other aspects of the prototypes. Third, MS Presidio ships with built-in pseudonymisation support through integration with the Faker library, enabling the replacement of detected PII entities with realistic synthetic values rather than simply blanking them, allowing us to meet our governance needs, requiring us to keep the redacted document still legible. Finally, Presidio's analyser component is powered by spaCy, a production-grade NLP library explicitly optimised for speed and throughput. This is a particularly important consideration given that batch processing is a planned feature for either Prototype V1 or V2, where performance under load will be a key requirement.

4.2 How It Fits Into the Flask Stack

For our prototypes, we have opted to make the redaction engine integrated into a lightweight Flask web server, selected over alternatives such as Django and FastAPI. Django was ruled out as it is a full-stack framework designed for applications with complex data models, authentication systems, and templating, none of which are core requirements for what is fundamentally a file processing API with a minimal frontend. FastAPI was considered for its performance and modern async capabilities, but the team's greater familiarity with Flask made it the lower-risk choice within the project's time and quality constraints. Within the Flask application, Presidio operates as a processing layer invoked on document text extracted via PyMuPDF. The pipeline flows as follows:

1) A document is uploaded via the Flask API endpoint

2) PyMuPDF extracts the raw text content from the PDF

3) The extracted text is passed to the Presidio Analyser, which identifies PII entities using spaCy-backed NLP models

4) Identified entities are forwarded to the Presidio Anonymiser, which applies the configured redaction or pseudonymisation operator

5) The sanitised output is returned or written to a local file (V1), with SQLite-backed audit logging planned for V2

For Prototype V1, processed files were stored locally to minimise infrastructure complexity. While prototype V2 introduced SQLite as a lightweight embedded database for persisting document batch records and audit logs. SQLite requires no external server process and is bundled with Python's standard library, keeping the application self-contained and locally deployable.

4.3 Custom recogniser extensibility

One of Presidio's key architectural advantages is its support for custom recognisers; here, out of the box, Presidio detects common entity types such as names, email addresses, phone numbers, and national ID formats. However, domain-specific identifiers, such as internal employee IDs, project codes, or organisation-specific reference numbers, may not be captured by the default model. Presidio then allows custom recognisers to be defined using either pattern-based rules (via regex) or NLP-based context clues, and registered with the RecognizerRegistry at runtime. This means the redaction engine can be extended without modifying core library code, making it straightforward to adapt the system to new document types or client requirements in future iterations and full-scale industry deployments.

5. Pseudonymisation & Mapping Register

5.1 What Pseudonymisation Is and Why Does It Matters Here?

When the redaction engine processes a document, it has two fundamental options for handling a detected PII entity; here it can remove it entirely, replacing it with a blank or a generic placeholder, or the engine can replace it with a synthetic value that preserves the document's readability while eliminating the original identifier. The latter approach is commonly known as pseudonymisation, and this is the design choice that our designed system adopts.

The distinction matters for mostly when we are considering result usability. For example, a document in which every person's name has been replaced with '[REDACTED]' is technically de-identified, but it may be structurally unusable for its intended purpose. For example, a legal chronology, an internal review document, or a case summary where entity relationships need to remain legible even if the identities themselves do not. Pseudonymisation in our case addresses this by replacing real identifiers with consistent, realistic synthetic values; all while the document remains coherent and navigable.

In our design, we have opted to use Microsoft Presidio, which implements pseudonymisation through integration with the Faker library, which generates synthetic values that are calibrated to realistic formats. Here, our system is designed to record the mapping between each original value and its synthetic replacement, and the record of this is stored within a mapping register. The design decisions behind both the pseudonymisation logic and the mapping register are documented in the subsections that follow.

5.2 Entity Detection – Scope & Confidence Filtering

The Presidio Analyser Engine scans the text extracted from the PDF and returns a list of all detected PII entities. The current implementation targets four entity types, including, PERSON, ORG, PHONE_NUMBER, and AU_ABN. This scope is deliberately narrow, for, rather than attempting broad general-purpose PII detection across every possible entity category, the system focuses on the entity types most likely to create direct re-identification risk in the document contexts this prototype is designed to process.

Two filtering conditions are first applied to the analyser's output before any entity is acted upon, first, entities with a confidence score below 0.6 are discarded. This threshold reflects a practical calibration decision, where a higher threshold produced false negatives where legitimate PII was not acted upon, effectively still leaving identifiable content in the output. Second, entity matches under three characters in length are

discarded, because very short matches are disproportionately likely to be false positives; fragments of text that match a pattern incidentally rather than representing a genuine identifier. Both these filters apply before the pseudonym map is built, meaning discarded entities are never assigned a replacement and never appear in the mapping register. The implication of this scoped detection design is significant for our system's limitations profile, as addressed in Section 13.5, where PII categories outside these four types (email addresses, physical addresses, dates, and other indirect identifiers), are not detected or acted upon by the current engine configuration. This is a design boundary, not an oversight, and the governance-facing report produced for each run discloses the entity types that were in scope.

5.3 Pseudonym Map

The current system calls the *build_pseudonym_map()* function once per processing job. This function is designed to iterate over every unique PII value detected in the document and assign it a synthetic replacement, producing a dictionary that maps each original identifier to its substitute for that job. The map is built once and applied consistently throughout the document, meaning that the same original value always receives the same synthetic replacement wherever it appears, ensuring that entity relationships remain coherent in the output.

The consistency scope is designed to be bound to per-job, meaning that a person named in ten different places in a single document will be replaced by the same synthetic name in all ten places. However, the same person named across two separate processing jobs will not necessarily receive the same synthetic name in both, as each job produces its own independent pseudonym map. The governance implications of this design are further discussed and highlighted in Section 9.6. Furthermore, these replacement values are generated using the Faker library, depending on each specific entity type.

- PERSON entities receive a synthetic full name via *fake.name()*.
- ORG entities receive a synthetic company name via *fake.company()*. PHONE_NUMBER entities receive a synthetic number via *fake.numerify()* formatted to the twelve-character Australian number format.
- AU_ABN entities receive a synthetic number formatted to the fourteen-character ABN format.

In each of these cases, the synthetic value is designed to be plausible and format-consistent with the original, preserving the document's readability without exposing real identifiers.

5.4 Redaction Mechanics

Once the pseudonym map is built, replacements are applied directly into the PDF structure using PyMuPDF, where for each detected entity, the "add_redact_annot()" function is called on the relevant page, thus annotating the region that is covering the original text with a white fill. Following this, the apply_redactions() function is called, which does two things, it first renders the white fill over the original text visually, then it removes the underlying text from the PDF's internal structure. This is an extremely crucial step, as the original text is not merely obscured by an overlay, it is essentially excised from the document structure entirely so that it cannot be recovered by selecting, copying, or inspecting the PDF's content stream.

Moreover, the synthetic replacement text is then inserted into the now-blank region using the insert_textbox() function. Here,this implementation includes a font-size scaling mechanism where the insertion attempts a font size of 11pt and shrinks it down to 5pt until the replacement text first within the available region. This prevents the replacement values from overflowing their boundaries, or overlapping adjacent content when a synthetic value is longer than the original.

To prevent duplicate annotations on the same region, which could occur if an entity is detected more than once at the same location, a redacted_rects list is used to track all the annotated regions per page. Before any new annotations are placed, the candidate rectangle is checked against this list, and if it's found that it intersects an already-annotated region, the annotation is skipped. This deduplication logic we have used, helps ensure that the output PDF is structurally clean regardless of how many times a given entity is detected at the same location.

5.5 The Mapping Register

After a successful redaction job, the "save_mapping_register()" function writes the pseudonym map to a separate JSON file named <job_id>_mapping.json. This file contains the complete original-to-replacement dictionary that is produced by the redaction engine, with classification labels attached to all the records. A warning statement is also embedded in the file itself, stating that it enables re-identification of individuals in the associated safe copy, and therefore, must be handled with the same access restrictions as the original source document.

This embedded warning is not just a procedural courtesy, rather it is a governance control that allows us to educate people and manage the risk of any data leaks. In a production

environment where the mapping register file may be handled by users who did not originate the processing job, we have designed the file in a way that it communicates the sensitivity of its own contents and the obligations that handling it creates. The register is also scoped per job, where each job produces a unique job ID, that is used to keep track of all the files in this register. Therefore, a single run produces one mapping register, and that file is the authoritative re-identification key for that specific run;'s safe copy; hence, there is no cross-job or cross-matter aggregation in the current implementation of our model.

5.6 Access Control & Route Design

The mapping register is never served via the /download/ route; as the download route serves only the standard evidence pack artifacts (safe copy, manifest, event log and governance-facing report), to operators and reviewers with appropriate access. The mapping register has therefore been structurally excluded from this path at the routing level, not as a configuration option.

Access to the mapping register is restricted to the admin role, and this is enforced via the @admin_required decorator on the /admin/mapping/<job_id> route. An operator or reviewing accessing the system under a non-admin role has no route through which the mapping register may be reachable, as the endpoints do not exists from them at the application layer. Every access to the mapping register via the admin login route is recorded in the audit log as a *MAPPING_REGISTER_ACCESSED* event, capturing the timestamp, job ID, and username of the accessing admin. This provides a complete access history for each register, allowing an incident response team to review when the register was accessed, by whom, and in the context of which job.

Encryption at rest for the mapping register files is identified as a planned control, and the implementation target is Fernet symmetric encryption at the application layer, consistent with the approach described in Section 11.7. This control is not present in the current build and is noted as a known gap between the current prototype state and the intended production security posture.

5.7 Per-job-scope – Governance Implications

The per-job consistency scope of the pseudonym map has a direct governance consequence that operators and reviewers should understand. Within a single process run, entity consistency is guaranteed; our model ensures that the same original identifier always receives the same synthetic replacement throughout the document. While across separate processing runs, even runs involving documents from the same matter, there is

no guaranteed consistency. Therefore, the same person may receive a different synthetic name in two separate documents that have separately been processed as two jobs.

This entails that a reviewer holding the safe copies from multiple jobs associated with the same matter will not be able to identify or infer entity equivalence across those copies without consulting the respective mapping registers. Hence, the presence of two different synthetic names in two different safe copies may not represent the same original individual. Where cross-document entity traceability is operationally required, the mapping registers for both jobs must be consulted under the access controls described in Section 9.6

This specific system design decision accepts a usability trade-off in favour of simplicity and containment, A per-matter shared mapping register would preserve cross-document entity consistency, but would require a persistent, matter-scopped sensitive data store that accumulates over the lifetime of the matter, and must also be governed accordingly. However, our per-job approach produces a smaller, more bounded sensitive artifact whose lifecycle is tied to a single run and whose destruction is straightforward when retention obligations are met.

5.8 Connection to the Evidence Pack

The governance report's Appendix D classifies the mapping register as an optional, restricted-only evidence pack component, rated as the highest sensitivity class in the overall pack structure (even more sensitive than the safe copy, the audit log, or the governance-facing report); and the backend's implications directly operationalises this classification.

The structural exclusion of the mapping register from the /download/ route further implements the governance report's requirement that it not be bundled with broader-circulation artefacts by default. The @admin_required access restriction also implements the "restricted only" classification. While the MAPPING_REGISTER_ACCESSED audit event implements the traceability requirement. The embedded re-identification warning implements the sensitivity disclosure requirements. Therefore, each of the governance classification in Appendix D maps to a specific control in the implementation of our approach, reflecting a coherent and secure system design that is built with purpose.

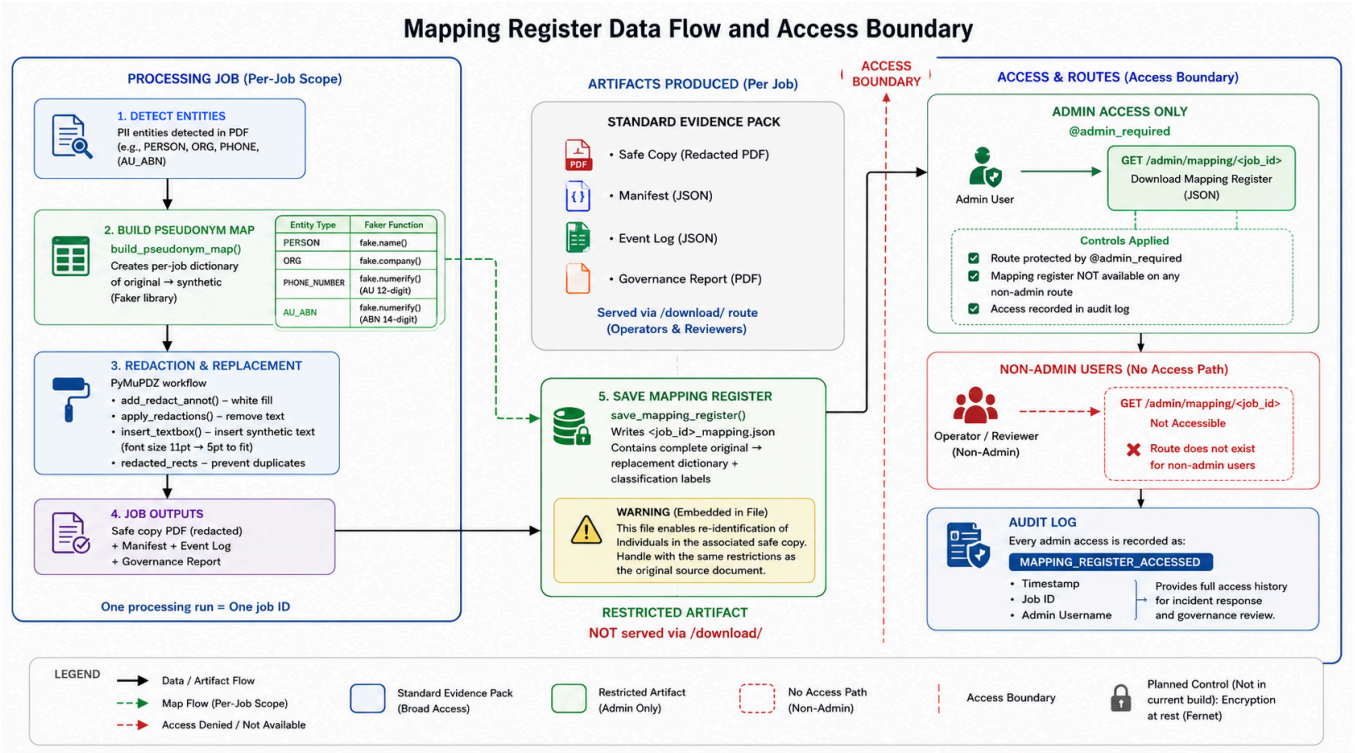


Diagram 5: *Mapping Register Data Flow and Access Boundary*

This diagram illustrates the lifecycle of the mapping register within the redaction system, including pseudonym generation, PDF redaction and replacement, restricted storage of the mapping register, and the access-control boundaries applied to sensitive re-identification data. It also goes on to demonstrate the separation between standard evidence pack artefacts and admin-only mapping register access, alongside the associated audit logging controls.

6. Audit Logging Architecture

6.1 The Design Philosophy

The audit logging architecture servees two distinct but related purposes:

> Compliance – The architecture produces a complete, tamper-evident record of every significant event in the document's lifecycle, from the moment it arrives at the ingestion boundary to the moment the output is released or its job is terminated.
>
> Security – The audit logging system has been designed in a manner that provides a structured, machine readable signal source that exposes any anomalous patterns such as, repeated validation features, blocked self-approvals and mapping register accesses; all which warrant operational attention.

These two purposes are kept architecturally distinct in the logic that they are coded with. The audit log is written to logs/audit.log, this is the record of compliance; while, the SQLite database jobs.db is the operational state record, that tracks job lifecycles and reviewer sign-offs. Together they provide overlapping coverage of the same events from two perspectives; the audit log provides an immutable chronological event stream; the database provides a queryable structured state model, therefore, neither replaces the other.

6.2 Audit Logs – Structure & Content

Every significant that occurs in the system is written to the logs/audit.log as a structured JSON line via the audit_log() function. Here, each entry contains five fields; a timestamp, the job ID associated with the event, the username of the user whose action triggered it, a named event identifier, and a details dictionary containing event-specific metadata. The job ID is the forensic thread that connects every log entry for a given document across its lifecycle. Where a single submission produces log entries across multiple zones, potentially across multiple sessions, and potentially under multiple usernames if an operator submits and an admin reviews. The job ID is the single consistent identifier that allows all of these entries to be assembled into a complete picture of what happened to a given document, in what order and under whose authority. One of the most important things to highlight here is that no raw PII values are ever written to the logs under any circumstances. The details dictionary of the log contains counts, filenames, outcome codes, and failure codes (essentially the metadata about what happened), but never the

content of the document being processed. This is both a security control and a governance requirement as, in an on-premises deployment where log files may be accessible to a broader set of privileged users than the application itself, a log that contains document content becomes a secondary source of sensitive data exposure. Therefore, the logs record what happened; what the document contained is not a concern for log architecture.

6.3 named Event Catalogue

The system logs the following named events, each corresponding to a defined transition or access in the pipeline:

- UPLOAD_RECEIVED – records the arrival of a file at the ingestion boundary.
- VALIDATION_PASSED and VALIDATION_FAILED – record the outcome of the four-gate validator, with VALIDATION_FAILED carrying one of six named failure codes in its details dictionary:
- *FILE_MISSING, FILE_EMPTY, FILE_TOO_LARGE, INVALID_MIME, PDF_ENCRYPTED, or PDF_NO_TEXT_LAYER.*
    - These named codes are queryable and aggregatable, as a spike in INVALID_MIME failures over a short window is a detectable signal that the upload boundary may be under active probing.
- Zone transition events – ZONE_PROCESSING, ZONE_REDACTED, ZONE_OUTPUT_GENERATED, ZONE_AWAITING_REVIEW, and ZONE_RELEASED; record each stage of the document's progression through the pipeline.
- DOWNLOAD and DOWNLOAD_BLOCKED record output access events.
- Authentication events – LOGIN_SUCCESS, LOGIN_FAILED, and LOGOUT, all provide session-level visibility. While ACCESS_DENIED records attempted access to a resource the requesting user's role does not permit.
- SELF_APPROVAL_BLOCKED records the specific case where a user attempts to approve the release of a job they submitted. This event is particularly significant from an insider threat perspective, as its presence in the audit log is evidence of an attempted policy violation, regardless of whether the attempt was deliberate or inadvertent.
- MAPPING_REGISTER_ACCESSED records every access to a mapping register file, as described in Section 9.6.
- PROCESSING_ERROR records unexpected failures during the processing or redaction phases.

6.4 Append-Only Semantics and Access Restriction

The audit log is append-only, meaning that entries cannot be modified or deleted. This was a deliberate security posture, as any log file that can be edited by a privileged user is not a viable audit trail, it essentially becomes a record of what a privileged user chose to preserve. Therefore, the append-only constraint means that even an admin-level user operating within the application will not be able to alter the historical records, as they can only add to it. Furthermore, access to view the audit logs is restricted to the admin role, and is again enforced via the @admin_required on the /logs route, as previously discussed in Section 9.6. Meaning, that operators and reviewers have no application-layer path to the raw logs, this restriction limits the population of users who can observe the audit trail to those whose role requires it, thus, reducing the risk that an operator may use log access to identify patterns in other users' activity or assess their own exposure before an incident review.

6.5 The SQLite Database

The SQLite database jobs.db, provides the structured operational record of the system's activity. It mainly maintains three separate tables including jobs, job_transitions, and reviewer_signoffs.

1. The jobs table holds a record for every processing job that has passed validation and entered the pipeline. It is essentially the authoritative registry of what jobs exist, what state they are currently in, and what artefacts they have produced.

2. The job_transitions table records every zone state change that occurs during a job's lifecycle, along with the username of the user who has triggered the transition. This table is the tamper-evident record of who did what to a given job and when, so in an incident investigation, it provides the precise sequence of actions taken on a document and the identity of each actor at each step. It is also the table that makes the dual-control release control auditable, where the transition from ZONE_AWAITING_REVIEW to ZONE_RELEASED must be triggered by a user other than the one who triggered UPLOAD_RECEIVED, and the job_transitions record makes any violation of this constraint immediately visible.

3. The reviewer_signoffs table records the formal approval event for each released job, here, the identity of the approving admin, the timestamp of the approval, and any conditions placed on the release are recorded. This table is the backend's implementation of the reviewer sign-off requirement, as stated explicitly in the governance report's post-processing checklist (Appendix E.2). Furthermore, a

released job without a corresponding reviewer_signoffs entry represents an anomaly that should not be possible under normal system operation.

6.6 The Job ID as Forensic Thread

The job ID deserves significant emphasis as an architectural choice, not merely as an implementational convenience. For every artefact that is produced by a processing run, this includes the safe copy, the manifest, the mapping register, the log entries and the database records, all carry the same job ID. This means that for any given document, a complete picture of its lifecycle can be reconstructed by querying across all of these artifacts using the job ID as the common key. When it comes to post-incident investigations, this is the property that allows the audit infrastructure to be actionable rather than just a decorative report. Given a job ID, an investigator can retrieve every log entry that is associated with that job from audit.log, every state transition from job.transitions, the approval records from reviewer_signoffs, and when authorised the mapping register from the restricted directory. The full chain of custody for a document, from the bytes that arrived at the upload boundary to the identity of the admin who approved its release, is fully recoverable from this one single identifier.

7. Security integrations within design

The security controls described in this section represent the planned production security posture for the privacy de-identification workflow system. While prototype implementations (versions 1 and 2) focus on establishing the core pipeline and governance artefacts, the controls documented here form the intended security baseline for any production or organisationally deployed version of the system. They are presented at the design level to demonstrate that the architecture has been deliberately planned with production-grade security in mind, and to provide a clear implementation pathway for any future development phases. Each control maps directly to a threat identified in Section 3; where relevant, the relationship between the control and the insider threat model or the external attacker surface is made explicit.

7.1 Threat framing

Before describing the individual controls, it is worth restating the two main threat categories that motivate this section's discussions. The first is the external attacker, who is anyone who is able to control the files submitted into the system. Their primary leverage is at the upload boundary, as they can create files that are designed to exploit the validation layer, exhaust the server’s resources, or trigger other vulnerabilities in the parsing libraries. The validator gates as described in Section 7 are the primary defence against such external attackers here; however, additional controls at the HTTP layer and application level are required to ensure that the validator cannot be overwhelmed or bypassed entirely.

The second, and arguably the most consequential threat within this project's scope, is the insider threat; this entails a privileged user who has legitimate access to the system but may misuse it. This includes the following scenarios: an operator who processes their own sensitive documents and then approves their own release, a system administrator who modifies or deletes any audit log entries, or a user who deliberately extracts the mapping register and uses it to re-identify individuals in a distributed output. Since the system looks to operate within an on-premises private deployment and processes personal and sensitive documents, the insider threat is not just a theoretical concern, but a realistic and structurally significant risk that the architecture must actively look to mitigate.

7.2 Attribute-based Access Control

Standard Role-Based Access Control (RBAC) assigns permission based on a user's role; for example, distinguishing between an operator and a receiver. While RBAC is a meaningful baseline to consider, it is highly insufficient for a system where sensitivity varies not just by user, but by the specific artefacts being accessed, the zone it belongs to, and the actual context in which the access is being requested.

Hence, to negate the use of RBAC and prevent breaches in security, we seek to implement Attribute-Based Access Control (ABAC) instead, where ABAC extends the RBAC model by further evaluating access decisions against a richer set of attributes. IN this system, access to a given artefact would be governed by the combination of the user's role, the zone the artefact belongs to, the sensitivity classification of the artefact (including a safe copy, audit log, or mapping register), and the operational context (for example, whether a job is in an active review state). Under this model, even a user with an elevated role cannot access the mapping register unless their role explicitly includes that attribute and the job state permits it. Similarly, an operator role would grant the ability to submit and process jobs but confer no access to the Admin Review and Release zone's approval controls.

Therefore, in a production deployment, ABAC enforcement would operate at two layers: at the filesystem level, using Linux Access Control Lists (ACLs) to enforce directory-level permissions per zone; and at the application layer within Flask, where route-level access checks evaluate the requesting user's attributes before any artefact is served. This dual-layer enforcement ensures that a bypass at the application layer will not automatically grant filesystem-level access, and vice versa.

7.3 Dual-Control Release and Separation of Duties

One of the most prominent insider threat vectors in a document processing system is drawn from the ability of a single user to both submit a job and authorise its release. Without implementing any structural control that can prevent this, an operator could theoretically print a sensitive document; one that they should not have any access to, or one they are deliberately attempting to exfiltrate, and immediately approve its own output for downstream distribution. This is where our dual-control release addresses this threat by enforcing logic flow that ensures that a user who has submitted a job cannot be the same user who approves its release in Zone 6 (Admin Review and Release). This demonstrates a classic example of separation of duties, whose patterns are well-established in financial controls, cryptographic key ceremonies, and secure document management systems.

In production, this control would be enforced at the application layer; here, the job manifest records the submitting user's identity at ingestion time, and the release

approval endpoint checks that the approving user's identity does not match the submitter. Any attempt to self-approve would be blocked at the API level and recorded as a security event in the security event log (described in Section 10). This creates both a preventive control (blocking the action) and a detective control (recording the attempt), which together ensure the pattern is both structurally prevented and auditable.

7.4 Canary Tokens and Honey Records

Even after implementing access controls, a sufficiently privileged insider may still be able to extract sensitive artifacts, particularly the mapping register. This could be done effectively without triggering any obvious alerts in the system. Here, we seek to implement canary tokens. These tokens are sometimes referred to as honey records or honeyfiles, and are a detection control that is designed to expose attacks revolving around this exact scenario.

The design would involve planting synthetic but plausible PII records within the system, for example, a fictitious employee identifier, a fabricated name-address pair, or a synthetic case reference number that belongs to no real individual and serves no legitimate operational purpose. These records are embedded within documents or within the mapping register itself. And since they are not associated with any real person or matter, there is no legitimate reason for them to ever appear in an external context. If such a token is detected outside the system, such as in a downstream dataset, an external disclosure, or a third-party environment, then its appearance constitutes unambiguous evidence that data exfiltration has occurred.

Beyond detection, canary tokens will also help provide forensic specificity, because each token can be associated with a specific zone, job, or artefact; its appearance in an external context identifies not just that exfiltration occurred but which artefact was the source. This is particularly valuable in an on-premises deployment where multiple privileged users may have access to different zones, and attribution of an incident to a specific access point would otherwise be difficult to establish.

7.5 IDS/IPS Integration – Host-Based Intrusion Detection

Given that our solution is framed within an on-premises deployment context, a Host-Based Intrusion Detection System (HIDS) is more operationally relevant than a network-based IDS. Here, the planned integration would look to use OSSEC or its actively maintained modern fork, Wazuh, deployed on the server that is hosting the pipeline.

Wazuh would be configured to monitor three primary surfaces: first, the zone storage directories themselves. Any file appearing in a zone directory via a path other than the expected application endpoint would indicate that the validator was bypassed entirely, which is a high-severity event. Second, the audit log files, any modification to an existing log entry, deletion of a log file, or unexpected change to file permissions on the audit directory would be flagged as a potential tamper event. Third, process behaviour, which includes any unexpected child processes spawned by the Flask application process (which could indicate that a crafted PDF triggered code execution in the PyMuPDF parsing layer), would trigger an alert.

Wazuh rules would be tuned to the specific directory structure of the pipeline, ensuring that alerts are actionable and specific rather than just generic filesystem noise. In a production deployment, alerts would route to the system administrator and, where configured, to a centralised SIEM platform. Furthermore, on the network perimeter, Suricata is planned as a lightweight Network IPS sitting in front of the Flask API endpoint. Suricata would be configured to detect HTTP-layer anomalies specific to the upload surface: oversized multipart request payloads that exceed the application's declared size limit, malformed Content-Type headers inconsistent with a file upload, and rapid sequential upload attempts from a single source that suggest automated probing of the validator gate behaviour. The combination of Wazuh at the host level and Suricata at the network perimeter provides defence-in-depth coverage across both the application boundary and the filesystem.

7.6 Rate Limiting and Request Throttling

The file size gate in the validator (*Section 7.1*) defends against resource exhaustion at the individual file level; however, it does not protect against an attacker submitting a high volume of smaller files in rapid succession, each individually within the size limit but collectively sufficient to saturate the server's processing capacity or generate excessive audit log volume. In production, Flask-Limiter would be applied to the upload endpoint to enforce a per-session and per-IP request rate cap. The specific thresholds would be tuned to reflect expected legitimate operator behaviour, frequent enough to support normal processing workflows without creating friction, but constrained enough that automated probing at scale is throttled and surfaced as an anomaly in the security event log. For the batch processing capability planned for V2, a job queue depth cap would be introduced to prevent a single operator from monopolising processing resources or generating an unmanageable volume of concurrent audit records. This control also supports the security event log's anomaly detection capability, where a sudden spike in queued jobs from a single user account is a meaningful signal that warrants review.

7.7 Encryption at Rest

The mapping register is the most sensitive artefact that is produced by the system we have designed, as it contains the direct linkage between real identifiers and their synthetic replacements, and its unauthorised disclosure would allow re-identification of individuals in any distributed safe copy. Safe copies, although they have been redacted, remain sensitive by default and may contain residual indirect identifiers, where addressing some of these has been ruled directly out of scope for our project, given the constraints. Therefore, to ensure utmost security, both categories of artifact want encryption at rest in the production/deployment stage of this pipeline. At the application layer, Fernet symmetric encryption (from the Python cryptography library) would help provide a straightforward mechanism for encrypting artefact files before they are written to disk. Fernet guarantees both confidentiality and integrity. Here, an encrypted file that has been tampered with will fail to decrypt, providing an additional tamper-evidence layer on top of the hash-based integrity controls described in Section 7.

At the filesystem level, LUKS (Linux Unified Key Setup) full-disk or partition-level encryption can be optimised to help provide a complementary layer that protects against physical access to the storage medium. This is relevant in an on-premises deployment where the server hardware may not be in a dedicated secure facility. The two layers are not redundant, as Fernet protects individual artefacts at the application level, while LUKS protects the entire storage volume in scenarios where filesystem-level access is obtained without going through the application. Encryption key management is a noted operational dependency for production deployment, where the keys must be stored separately from the encrypted data, with access restricted to the minimum necessary set of roles (following ABAC).

7.8 Multi-Factor Authentication

Any operator or administrator portal that provides access to job management, artefact review, or release approval must enforce multi-factor authentication in a production deployment. Single-factor authentication, even with strong passwords, is insufficient given the sensitivity of the documents and artefacts the system handles, and the consequences of a compromised account gaining access to the mapping register or approval controls. The planned implementation uses TOTP (Time-Based One-Time Password) via the pyotp library, integrated into the Flask login layer. TOTP is well-suited to an on-premises deployment as it requires no external authentication service, operates offline after initial setup, and is compatible with widely available authenticator applications. This approach keeps the authentication infrastructure self-contained, consistent with the system's on-premises design philosophy.

7.9 Application Hardening

Beyond the specific controls described above, the Flask application itself must be hardened against common web application vulnerabilities in a production deployment. Several configuration-level controls are planned as follows:

a. The debug mode must be disabled in any non-development environment. Flask's debug mode exposes full stack traces in HTTP error responses, potentially revealing internal file paths, library versions, and code structure to an external observer, all of which reduce the attacker's reconnaissance burden.

b. A strict Cross-Origin Resource Sharing (CORS) policy must be configured to restrict which origins are permitted to make requests to the API. In an on-premises deployment, the expected origin set is small and known; permissive CORS configuration beyond the known local setup is both unnecessary and a meaningful attack surface expansion.

c. Content Security Policy (CSP) headers must be applied to any frontend components to help mitigate cross-site scripting risks. While the system's frontend is minimal by design, CSP headers are a low-cost, high-value hardening measure that should be applied as a baseline regardless of frontend complexity.

d. The Flask application process must run as a dedicated low-privilege service account with no sudo rights and write access restricted to its designated zone directories only. This limits the blast radius of any successful code execution within the application; a crafted file that triggers arbitrary code execution in the PyMuPDF layer would operate under the constraints of that service account, rather than with the broader privileges of a standard user or administrator.

7.10 Dependency Management and Software Bill of Materials

A production deployment of this system operates within a specific threat context; here, it is expected to process legally sensitive documents and is likely to be deployed in environments with healthcare-adjacent or government-adjacent compliance requirements, all contexts that increasingly expect demonstrable supply chain security practices. All Python dependencies must be pinned to exact versions in *requirements.txt* to ensure reproducibility and to prevent silent dependency upgrades from introducing unvetted changes. A Software Bill of Materials (SBOM) must be generated and maintained, enumerating all libraries and their versions. This is particularly important for PyMuPDF, which has a documented history of Common Vulnerabilities and Exposures (CVEs) related to malformed PDF parsing, the exact attack surface the system is

designed to protect against. A documented patching cadence for all dependencies in the stack (Flask, PyMuPDF, Presidio, spaCy, python-magic) must be established and followed, with priority given to any CVE that affects PDF parsing or file handling.

8. Framework Alignment

The privacy de-identification workflow has been designed with reference to three established cybersecurity and governance frameworks: the NIST Cybersecurity Framework (CSF), the Australian Cyber Security Centre's Essential Eight, and the ISO standards referenced in the governance team's report. This section maps the system's design decisions and security controls to each framework, demonstrating that the architecture reflects recognised best practice and is not an ad hoc collection of implementation choices. Where a control has already been described in detail in Section 11, this section references it rather than restating it. The purpose here is alignment and traceability, not repetition.

8.1 NIST Cybersecurity Framework

The NIST CSF organises cybersecurity activity into five core functions: Identify, Protect, Detect, Respond, and Recover. These functions are not sequential stages but concurrent and mutually reinforcing dimensions of a mature security posture. The following subsections map the system's design to each function.

8.1.1 Identify

The Identify function covers the establishment of an organisational understanding of the system's assets, data flows, and risk context; this essentially provides the foundational knowledge that all other security activity depends on. The six-zone trust architecture described in Section 4 functions as a living asset inventory, where each zone has a defined purpose, a known set of artefacts, and documented sensitivity classifications. The folder structure described in Section 5 makes this asset boundary concrete; here, each directory corresponds to a specific phase of the document lifecycle, and the sensitivity of its contents is explicitly declared. Together, these provide the data flow visibility that NIST CSF's Identify function requires.

The threat model described in Section 3 further constitutes the risk assessment component of this function. By explicitly naming the external attacker and the insider threat as the two primary adversary categories, and by mapping each control to the threat it mitigates, the system is able to demonstrate an active understanding of its risk context rather than a generic application of security measures. For a production/deployment stage, this would be formalised as a Data Flow Diagram (DFD) with the threat annotations, making it a document that maps every data movement in the system against its associated threat surface. The zone architecture

and folder structure diagrams described throughout this report provide the foundations of the DFD.

8.1.2 Protect

The protect function here covers the controls that limit or contain the impact of a potential cybersecurity event. This is the most extensively addressed function in our system's design.

The four-gate ingestion validator (Section 7) is the primary protective control at the upload boundary, enforcing file size limits, magic byte verification, PDF sub-validation, and further cryptographic hashing before any content enters the processing pipeline. The validator essentially aims to ensure that an attacker-controlled input cannot reach the redaction engine in an untrusted state.

Furthermore, the fail-fast principle further ensures that adversarial input receives the minimum possible feedback, reducing the attacker's ability to probe the boundary iteratively. Zone separation (as discussed in Section 4) and isolated storage (Section 5) implement the principle of least privilege at the architectural level. No zone has access to artefacts beyond those required for its specific function, and no data path between non-adjacent zones is ever permitted. This structurally limits the lateral movement within the system; a compromise of one zone does not automatically grant access to another.

ABAC (Section 11.2) and dual-control release (Section 11.3) extend the principle of least privilege to the human layer, where access to sensitive artifacts is gated by user attributes, not just credentials, and the most consequential action in the pipeline, release approval, requires a second authorised party. Further, encryption at rest for the mapping register and safe-copy directories (Section 11.7) provides a Protect-function control against physical or privileged filesystem access, ensuring that artefact confidentiality does not depend solely on application-layer access controls.

Application hardening (Section 11.9), disabled debug mode, strict CORS, CSP headers, and a low-privilege service account for the Flask process; implements Protect-function baseline hardening at the web application layer, reducing the exploitable attack surface of the application itself.

8.1.3 Detect

The detection function covers the capabilities that enable the timely discovery of cybersecurity events. This is the function that is most commonly underdeveloped in

prototype systems, and its inclusion here as a planned production capability is a deliberate design commitment after considering constraints.

The structured fail log system described in Section 7 helps to provide the foundation for anomaly detection at the upload boundary. As failure codes are named and machine-readable rather than free-text, they are aggregatable and monitorable. For example, a spike in INVALID_MIME rejections over a short window of time is a detectable signal that someone is probing the validator. Furthermore, sustained patterns in failure logs such as FILE_TOO_LARGE from a single session would also be consistent with resource exhaustion probing. These signals are only actionable if the log format supports problematic analysis, which the structured failure log schema is specifically designed to enable.

Further, the security event log described in Section 10, distinct from the audit log, captures security-relevant events such as repeated validation failures, blocked self-approval attempts, and also unexpected artefact access patterns. This log is designed to be ingestible by a SIEM platform in a production deployment scenario, helping provide the monitoring infrastructure that this detection function requires.

Through the planned Wazuh/OSSEC HIDS integration (Section 11.5), we are hoping to achieve host-level detection, including directory monitoring for unexpected file writes, audit log tamper detection, and anomalous process behaviour from the Flask application. Suricata at the network perimeter provides HTTP-layer anomaly detection, surfacing upload boundary attacks that the application layer alone may not capture. Also, the use of canary tokens (Section 11.4) is made to function as a passive detection mechanism for exfiltration. Their value is precisely that they require no active monitoring, and their appearance in an external context is itself the alert.

8.1.4 Respond

The respond function covers all the actions taken in response to a detected cyb security attack/event. A system that detects events but has no defined response procedures ends up converting the detection capability into noise rather than action.

For this system, our primary response-function artefact is an operator runbook that translates specific detection signals into defined response actions. While the full runbook is an operational document rather than a design document, our system's architecture pre-positions several response capabilities.

The job quarantine mechanism we have implemented, where files that fail validation are isolated and logged rather than simply discarded, helps to ensure that rejected

submissions are available for forensic review. The SHA-256 hash computed at Gate 4 is the integrity anchor that makes this meaningful, as the quarantined file can be verified against the logged hash to confirm it has not been altered since rejection.

As discussed in prior, for the detection signals to indicate active probing, a spike in INVALID_MIME or FILE_TOO_LARGE failures above a defined threshold within a defined time window, the defined response is to quarantine the source session, alert the system operator, and escalate to the security event log for SIEM ingestion. On top of this, rate limiting via Flask-Limiter (Section 11.6) provides the automated throttling component of this response, reducing the window of opportunity while human review proceeds.

For insider-threat scenarios detected through Wazuh, where unexpected file modifications in a zone directory, or changes to audit log entries are detected, the defined response is immediate escalation to the system administrator, suspension of the affected user's access pending review, and initiation of the forensic investigation process using the job manifest and event log as the primary evidence sources.

8.1.5 Recover

The recovery function covers the capabilities that support the restoration of normal operations after a cybersecurity event and the preservation of the forensic record necessary for post-incident review.

Our system's architecture directly supports recovery through the SHA-256 hash-before-transformation design described in Section 7.4, because a cryptographic fingerprint of every submitted file is computed and logged before any processing occurs, a job can be re-run from its original quarantined file at any point, and the hash provides the integrity anchor confirming that the file being re-processed is identical to what originally arrived. In a scenario where a zone is compromised and its outputs are suspect, this allows clean re-processing from the verified original without relying on any potentially tainted intermediate artefact.

Offline backup of audit logs and the mapping register (Section 11.7) ensures that the forensic record survives even a destructive insider attack against the primary system. The backup is explicitly designed to be inaccessible to the application process itself, where the app user has no write-back access to the backup location, meaning an insider who compromises the application layer cannot destroy the evidence of their own activity.

Dependency pinning and the SBOM (Section 11.10) support recovery from supply chain incidents, for example, when a CVE is disclosed against a dependency, the

pinned version provides the exact reference point needed to assess impact and manage a controlled upgrade to a patched version.

8.2 Essential Eight

The Australian Cyber Security Centre (ACSC) leads the Australian Government's efforts to improve cyber security and prioritises the Essential Eight as a set of mitigation strategies used for securing systems against cyber threats. It is particularly relevant to our design solution given its on-premises deployment context and its likely applicability to government-adjacent or regulated industry environments in Australia. The following maps our system’s design to the Essential Eight controls, which are most applicable to its architecture.

8.2.1 Application Control

Application Control looks to restrict the execution of unauthorised software on a system, ensuring that only explicitly approved applications and processes can run. In the context of this deployment, the primary risk this control addresses is arbitrary code execution that is triggered by a crafted PDF in the PyMuPDF parsing layer; a risk explicitly named in the threat model.

Furthermore, in the production stage, Mandatory Access Control (MAC) systems such as AppArmor profiles or SELinux policies would be configured to restrict the Flask application process to a defined set of permitted operations, such as, reading from the ingestion directory, writing to its designated zone directories, executing its approved Python environment, and nothing else.

Any attempt by the Flask process to spawn an unexpected child process, write to an unauthorised directory, or execute a binary outside its approved set would be blocked at the OS level and flagged by the HIDS. This helps provide a containment layer that limits the blast radius of a successful parser exploit to the constrained environment of the application profile, rather than the full capabilities of the host system.

8.2.2 Patch Applications

A documented and regularly executed patching cadence is required for all libraries in the application stack. The specific libraries of concern are Flask, PyMuPDF, Presidio, spaCy, and python-magic. Of these, PyMuPDF warrants particular attention,as it has a documented history of CVEs related to malformed PDF parsing, which is precisely the attack vector the system processes. A monthly patch review cycle is the minimum

appropriate cadence for this dependency, with critical CVEs addressed outside the regular cycle as they are disclosed. Further, dependency pinning in *requirements.txt* and the maintenance of an SBOM (discussed further in Section 12.4) are the operational prerequisites for a functioning patch process, as without knowing the exact versions in use, assessing the impact of a disclosed CVE is significantly more difficult.

8.2.3 Restricting Administrative Privileges

The Flask application process runs as a dedicated low-privilege service account with no sudo rights and filesystem access restricted to its designated zone directories. Therefore, no operator or standard user account holds administrative privileges on the host system for purposes related to the application's normal operation.

ABAC enforcement (Section 11.2) extends this principle to the application layer, where even users with elevated roles within the application cannot access artefacts or perform actions outside their defined attribute set. The mapping register, in particular, is accessible only to roles that have been explicitly granted that attribute, as this is not a function of general elevated access.

8.2.4 Multi-Factor Authentication

MFA is planned to be enforced on all operator and administrator portals, as described in Section 11.8. The TOTP implementation via pyotp keeps the authentication infrastructure self-contained, consistent with the on-premises deployment model, and avoids dependency on external identity providers.

8.2.5 Regular Backups

Audit logs and the mapping register are backed up to an offline or air-gapped location on a defined schedule, where the backup is write-only from the application's perspective, so the app user can ship logs to the backup location but cannot read from or modify it. This ensures that a destructive insider attack against the primary system cannot simultaneously destroy the backup record. Furthermore, the SHA-256 hash of each backed-up artefact is recorded at the time of backup, providing integrity verification on restoration.

8.2.6 User Application Hardening

Application hardening measures, described in full in Section 11.9, include disabled debug mode, strict CORS policy, Content Security Policy headers, and operation of the Flask process under a low-privilege service account. Collectively, these measures reduce the exploitable attack surface of the web application layer and ensure that common web application vulnerabilities, such as information disclosure via stack traces, cross-origin request forgery, and cross-site scripting, are addressed at the configuration level rather than left to chance.

8.2.7 Attack Surface Reduction – DOCX Exclusion

The decision to restrict the system to PDF inputs only and to explicitly exclude DOCX support is directly aligned with the Essential Eight's broader principle of attack surface reduction. As described in Section 7, DOCX files are ZIP-based archives that introduce a qualitatively different class of attack vectors, including magic-byte ambiguity, zip bombs, and path traversal attacks embedded in archive entry names, all of which would require a materially more complex validation pathway to address safely. By scoping the system to PDF only, this entire class of archive-based attacks is eliminated from the validation surface. This is recorded here as a deliberate, Essential Eight-aligned security scoping decision that helps us standardise the input format, rather than an implementation shortcut.

8.3 ISO Standards Cross-Reference

The governance team's report addresses the ISO standards most relevant to the system's data handling obligations, including the privacy and information security management frameworks applicable to the processing of personally identifiable information. Rather than duplicate that analysis here, this section establishes the connection between the backend's technical design and the governance report's ISO coverage.

The technical controls described in this report, such as zone separation, access control, audit logging, encryption at rest, and the evidence pack structure, are the implementation layer through which the governance requirements expressed in those ISO frameworks are operationalised. Where the governance report states a requirement (for example, that access to sensitive artefacts must be restricted, or that a traceable audit record must be maintained), the corresponding technical control in this report is the mechanism by which that requirement is met in the system's architecture.

We would like to advise any readers requiring the full ISO framework analysis, including the specific standard references and compliance position, to refer to the governance team's report directly. The two documents are designed to be read together, so the

governance report establishes the normative requirements, and this report demonstrates how the technical design satisfies them.

8.4 Dependency Pinning, SBOM, and Deployment Context

The system's likely deployment contexts, environments with government-adjacent or healthcare-adjacent compliance expectations, carry increasing expectations around software supply chain transparency. Both the Australian Government's ISM (Information Security Manual) and healthcare sector security guidance increasingly treat the ability to enumerate and verify software dependencies as a baseline operational requirement, not an optional practice. Hence, in our design, all Python dependencies are to be pinned to exact versions in *requirements.txt* for any production deployment. This ensures reproducibility across environments, prevents silent dependency drift, and provides the precise reference point needed to assess the impact of a disclosed CVE against the specific versions in use.

Furthermore, a Software Bill of Materials (SBOM) enumerating all libraries, their versions, their direct dependencies, and their known vulnerability status at the time of deployment is to be generated and maintained as a living operational document. In the event of a supply chain incident, for example, a compromised package update to a dependency, the SBOM is the document that will help and allow rapid assessment of whether the deployed system is affected. Together, dependency pinning and the SBOM form the supply chain security foundation that supports the patching cadence described in Section 12.2, and that would be required for any formal security assessment or compliance audit of a production deployment.

9. Governance Integration

The governance team's report establishes the normative requirements for how documents processed through this system must be handled, reviewed and released. This section is designed to document how the backend architecture operationalises those requirements; that is, which governance obligations are enforced automatically by the system's design, which are supported by system outputs but remain dependent on human judgement, and which fall entirely outside of our system's technical scope. This distinction is extremely important, as the governance report describes what must happen, while this section is precise about what the system guarantees, what it enables, and what it cannot substitute for.

9.1 The Evidence Pack as the Backend's Governance Output

The governance report's Appendix D defines the evidence pack as the structured set of artefacts that are produced for each processing run. This backend report looks to elaborate on the system that produces this pack. The following maps each required component to the zone and mechanism responsible for generating it.

9.1.1 Safe Copy

The safe copy is produced by the Redaction Zone (Zone 4) and finalised by the Output Generation Zone (Zone 5); it can be seen as the primary downstream artefact. Essentially, the risk-reduced version of the input document with detected PII entities removed or replaced with synthetic values via Presidio's pseudonymisation operator. Critically, and consistent with the governance report's position, the safe copy is treated as sensitive by default from the moment it is written. The backend does not classify it as a public or freely distributable document, as it is the output of a redaction process, not a guarantee of anonymity. This default sensitivity classification is a design decision, not an afterthought, allowing us to effectively harden the solution pipeline.

9.1.2 The Manifest

The manifest is generated by the Output Generation Zone (zone 5) and records the run metadata for the job, including the job ID, the policy version applied, the operator identity, a timestamp, and the inventory of output artefacts produced. The manifest is the document that ties everything together, as it acts as the reference point against which any artefact in the pack can be verified as belonging to a specific run.

9.1.3 Event Log

The event log is produced continuously across all zones from the moment a file enters the ingestion boundary. It records every processing and access event in a structured, append-only form; as described in Section 10, raw document content is never written to the log under any circumstances, as the log records what happened to the document, not what was in it. This is the strict audit trail that the governance report requires to be present in every evidence pack, as it ensures that no third party is able to reverse engineer a redacted document.

9.1.4 Governance Report

The governance-facing report is a human-readable summary generated by the Output Generation Zone (zone 5). It must state what was processed, what entity types were detected and acted upon, what redaction or pseudonymisation operators were applied, what metadata handling was performed and finally, what limitations remain. This is the document a reviewer reads before making a judgment about whether the safe copy is appropriate for its intended downstream use. The backend generates this report automatically from the structured outputs of the processing run, meaning that this report is not manually written, which means its contents are directly traceable to the system's actual behaviour rather than an operator's recollection of it, thus making the system even more robust and ingrained in facts rather than human judgement.

9.1.5 Integrity Artifacts

Integrity artefacts, specifically the SHA-256 hash computed at Gate 4 of the ingestion validator, are recorded in the manifest and are available for later verification. The hash is computed over the raw submitted bytes before any transformation, providing a tamper-evident receipt of exactly what has arrived. A downstream reviewer can then later look to verify that the safe copy was derived from the original file by checking the hash recorded in the manifest against the quarantined original.

9.1.6 Mapping Register

The mapping register, when generated, is stored separately from the ordinary output bundle and is never included in the default downstream distribution. This directly implements the governance report's requirement that the mapping register is to be excluded from broader-circulation artefacts. For a more in-depth look at this artifact

please refer to Section 9 of this report, where its storage location, access controls, and retention handling are better described.

9.2 Pre-Processing Checklist – Backend Enforcement Points

The governance report's Appendix E.1 defines a pre-processing operator checklist; here, several of the checks on that list are enforced automatically by the backend; others are human obligations the system supports but cannot substitute for. For a clear distinction, the following applies:

The backend automatically enforces the rule that the input file must be within the supported prototype scope, meaning it must pass the validator's magic byte check and PDF sub-validation gates, which reject any file that does not meet the system's defined input specification before it reaches the processing engine. An operator cannot accidentally or deliberately process an unsupported file type; the system’s pipeline structurally prevents it. The backend automatically enforces that processing occurs within the declared controlled environment through zone isolation and filesystem-level access controls, and, as a consequence of normal system operation, raw or identifiable documents do not leave the processing boundary.

The backend further records the policy version applied to every run in the manifest, supporting the checklist requirement that the correct de-identification policy has been selected and confirmed. However, the selection of the appropriate policy for a given document and purpose remains an operator decision; here, the system applies what it is instructed to apply, and the manifest records that instruction faithfully.

The backend does not have the capacity to enforce the checks that require contextual human judgment; these include confirming that the file is the intended source document and not the wrong matter file, confirming that the intended downstream environment has been assessed, and confirming that the operator is authorised for the specific processing run under the applicable access model. These checks are the operator's responsibility, and the pre-processing checklist in Appendix E.1 is the governance instrument through which that responsibility is discharged. The system supports this by requiring operator identity to be recorded at submission, but it does not have the ability to verify the operator's intent or contextual knowledge.

9.3 Post-Processing Checklist – System vs Human Judgment

The governance report's Appendix E.2 defines a post-processing and reviewer checklist; again, some items on this checklist are directly supported by system outputs, while others require a human reviewer to exercise independent judgment that the system cannot perform on their behalf. The distinction is as follows:

The system guarantees that the expected evidence pack artefacts are present, which is a job that, once completed successfully, will always produce a safe copy, manifest, event log, and governance-facing report. The manifest records the inventory of produced artefacts, and any discrepancy between the declared inventory and the actual files present is immediately detectable. The system guarantees that job identifiers, timestamps, and policy version references are internally consistent across artefacts, ensuring that it is deliverable; all artefacts in a pack share the same job ID, which is the forensic thread connecting them, so a reviewer can verify consistency without performing any manual cross-referencing.

The system guarantees that the governance-facing report contains a metadata handling statement and a limitations and residual risk statement. These are generated automatically and are always present, ensuring that a reviewer does not need to remember to include them as they are structural outputs of every run. The system does not guarantee that direct identifiers have been correctly handled in every instance, as Presidio's NLP-based entity recognition operates at a confidence threshold, and entities that fall below that threshold may not be detected. The governance-facing report discloses the entity types acted upon, but it cannot disclose what it missed. Furthermore, the reviewer's checklist item, stated as “checking that direct identifiers appear to have been handled in accordance with policy”, is a human judgment task that the system's output supports but cannot replace.

The system does not guarantee that no combination of indirect identifiers creates an unreasonable disclosure risk in the intended environment, as an indirect identifier risk is context-dependent; for the same combination of attributes may be low-risk in one downstream environment and high-risk in another. This assessment requires the reviewer to apply judgment about the specific intended use, a task that the governance report explicitly frames as a practical reviewer judgment, not a system guarantee. Release and external disclosure decisions, where proposed, require documented sign-off from the reviewer/admin. The system records reviewer identity and timestamps access events in the event log, but the release decision itself is a

human governance act; here, the backend supports traceability of that act; it does not perform it.

9.4 Safe Copy Sensitivity

A consistent and explicit position on what the safe copy is and is not is one of the most important things this report can establish, because misunderstanding this point is the most consequential way a downstream user could misuse the system's output. Our designed redaction engine identifies and acts upon PII entities detected by Presidio's analyser within the text layer of the PDF. Here, named entities, contact information, identification numbers, and other recognisable structured PII formats that appear in the text layer and are captured by the configured recogniser set are either removed or replaced with synthetic values; this is what the system essentially does.

The important note here is that the system does not look to strip all metadata from the output PDF. As PDF metadata fields, including author, creator, creation date, and modification history, may persist in the safe copy depending on how PyMuPDF handles the specific file. The governance-facing report contains a metadata handling statement that discloses the system's behaviour for the processed file, and this statement is the mechanism through which that limitation is communicated to the reviewer. The reviewer's checklist explicitly requires them to read and understand this statement before relying on the output, thus introducing heavy human review dependencies.

Furthermore, the system we have designed does not detect PII embedded in images within the PDF; for example, a scanned signature, a photograph of an identity document, or a handwritten note captured as an image are outside the scope of the text-layer redaction engine. Here, the backend has designed Gate 3 of the validator to reject image-only PDFs precisely because a document with no text layer cannot be meaningfully processed, but a hybrid PDF with both a text layer and embedded images will pass the validator, and the image content will not be redacted.

Finally, the backend’s design for the solution does not look to make contextual re-identification risk assessments. It detects and acts on what its configured recognisers identify. Here, any indirect identifiers, such as combinations of attributes that do not individually constitute PII but together may enable re-identification in a specific environment, are not systematically detected. The system's output is a risk-reduced document, not a risk-free one. The governance report's consistent framing of the safe copy as sensitive by default is the correct position, and this is reflected in the backend's design, as the solution is built around that framing from the ground up.

9.5 Limitations Statement

In the interest of being precise about what this system can and cannot claim, the following limitations identified by our team are stated explicitly below.

The designed system does not have the ability to guarantee the complete removal of all personally identifiable information from a processed document. As stated numerous times earlier, Presidio's entity recognition operates at a defined confidence threshold and is trained on general-purpose NLP models. Domain-specific identifiers, novel naming conventions, and entities that appear in formats outside the recogniser's training distribution may not be detected. Here, the use of custom recognisers can extend coverage, but they cannot achieve exhaustive coverage of an open-ended PII space.

The system cannot 100% guarantee that all metadata has been fully removed. PDF metadata handling is dependent on the structure of the input file and the capabilities of PyMuPDF at the point of processing; here, we utilise the governance-facing report to disclose the actual behaviour for each run, which is the mechanism for managing this limitation, not a means of eliminating it.

The system cannot assess re-identification risk in a given downstream environment, as that form of assessment requires knowledge of the receiving environment, the population it operates within, and the auxiliary data available to potential adversaries; none of which the system has access to. Re-identification risk assessment is a human governance responsibility, supported by the reviewer checklist and escalation provisions in Appendix E.2 of the governance report.

The system cannot prevent the misuse of the mapping register by a user with legitimate authorised access to it; however, the integration of measures such as access controls, dual-control release, canary tokens, and audit logging described in Section 11 significantly raises the difficulty and detectability of such misuse, but they do not make it technically impossible. The governance and legal framework within which the system operates is the ultimate constraint on authorised users; here, the backend's role is to make violations difficult, detectable, and traceable, not to eliminate the need for that framework.

10. Design Decisions Register

The following register documents the major architectural and implementation decisions made during the development of the workflow. Each decision was evaluated in terms of security impact, deployment practicality, maintainability, and suitability for handling sensitive legal documents containing personally identifiable information (PII).

Decision	Alternatives	Rationale	Security Implication	Trade- Off
Restricting the system to PDF-only document processing	Supporting DOCX, TXT, image uploads, or mixed file formats	PDF was selected because legal documents are commonly distributed in PDF format and the structure is more predictable during extraction and validation. Limiting scope allowed the team to focus on secure workflow design rather than file compatibility.	Reduces attack surface by eliminating archive-based attacks associated with DOCX and ZIP parsing. Simplifies validation and quarantine logic.	Prevents support for alternative document formats that may exist in real legal workflows.
Flask selected instead of FastAPI or Django	FastAPI, Django	Flask was selected because the prototype required a lightweight backend with minimal framework overhead. The team's prior familiarity with	Smaller framework footprint reduces unnecessary exposed components and simplifies application hardening.	Flask lacks the asynchronous capabilities of FastAPI and has fewer built-in enterprise features than Django.

		Flask also reduced development risk within project time constraints		
Microsoft Presidio selected instead of Philter	Philter, regex-only redaction approaches	Presidio was selected due to its Python-native implementation , extensibility, active maintenance, and integration with spaCy NLP models. It aligned closely with the Python backend architecture.	Supports contextual PII detection rather than relying solely on static pattern matching, improving de-identificatio n coverage.	NLP-based systems may still produce false positives and false negatives. Dependency complexity is higher than simpler regex-only approaches.
SQLite selected instead of an external database server	PostgreSQL, MySQL, MongoDB	SQLite was chosen because the prototype operates locally and does not require distributed multi-user database infrastructure. It simplified deployment and reduced operational overhead.	Removes the need for a network-exp osed database service, reducing attack surface and limiting infrastructure complexity.	SQLite is not ideal for large-scale concurrent workloads or enterprise-scal e deployments.
Fail-fast validation model implemented for uploaded files	Lenient validation or partial recovery processing	The validator immediately rejects files that fail security or structural checks rather	Prevents malformed, encrypted, oversized, or suspicious documents	Strict validation may reject files that could otherwise be partially recoverable or

		than attempting partial processing of potentially unsafe files.	from progressing deeper into the workflow. Reduces parser abuse risk.	manually corrected.
Trust-zone separation architecture used instead of a monolithic workflow	Single-process monolithic pipeline	The workflow was intentionally divided into separate zones so that each stage handles only the minimum information required for its task.	Limits exposure of raw PII, supports least-privilege principles, and reduces the impact of compromise in a single zone.	Introduces additional implementation complexity and stricter inter zone coordination requirements.
Separate storage directories used for different artefact classes	Single shared storage directory	Storage separation mirrors the trust-zone model and isolates artefacts according to their sensitivity level and operational purpose.	Reduces lateral exposure risk and improves containment if one storage location is compromised.	Requires more structured file management and increases operational complexity.
Generation of a “safe copy” instead of modifying the original document	Directly overwriting original documents	Producing a separate de-identified copy preserves the integrity of the original uploaded document for traceability and governance purposes.	Prevents accidental destruction or modification of the original source document and supports audit verification.	Requires additional storage management and introduces the need to securely manage multiple document versions.
Structured	Minimal logging	Structured	Supports	Logging

audit logging implemented across all workflow stages	or console-only logging	audit logs provide traceability, accountability, and forensic visibility across the entire processing lifecycle.	investigation of failures, insider misuse, and validation outcomes while improving governance alignment.	introduces risk if sensitive information is accidentally captured, requiring strict log sanitisation controls.

11. Limitations, Risks & Future Work

11.1 Prototype Limitations

The prototype created provides the core architecture and security concepts that are desired from the legal document de- identification. Despite this, a few limitations remain.

First, a full network-layer deployment architecture is not implemented by the existing system, which functions mainly as a local prototype. The current implementation does not cover security measures like distributed authentication architecture, TLS setup, reverse proxy hardening, or production-grade API gateway safeguards.

Secondly, role-based access control and multi-user authentication are not currently supported by the prototype. Instead of a fully distributed business workflow with numerous reviewers, administrators, and processing operators, the current design relies on a controlled local operator environment. Thirdly, there is currently no link between the system and real-time security monitoring feeds, SIEM infrastructure, or external threat intelligence services. As a result, threat detection and anomaly identification continue to be mainly based on workflow and local.

Furthermore, the redaction engine's recognition set is purposefully limited in scope for prototype development. Organization-specific recognition systems and even greater entity coverage may be necessary in actual-world legal contexts.

11.2 Residual Risks

The system still has residual risks even with the usage of trust-zone separation and tiered security controls. Metadata leakage is one major risk. Sensitive information may still be revealed if embedded PDF metadata, such as author names, programme IDs, timestamps, or document history, is not completely sanitised. Even though document content may be effectively de-identified.The accuracy of NLP detection is another residual risk. Perfect identification of all personally identifiable information is not guaranteed by Presidio and similar NLP-based methods. While false positives may needlessly reject valid text and lower document usability, false negatives may fail to identify sensitive identifiers.Attackers or privileged insiders who obtain direct filesystem access provide a continuous risk. Despite application-layer safeguards, sensitive data may still be exposed if a malicious insider gains access to mapping registers, audit artefacts, or raw storage directories outside of the intended process.

Additional risks include:

- Unsafe logging behaviour
- Configuration mistakes
- Incomplete redaction review
- Incorrect trust-zone handling
- Accidental release of insufficiently sanitised documents

11.3 Future Work - Version 2 Roadmap

For a future V.2 several improvements have been identified.Support for batch document processing would be a significant improvement, allowing several legal documents to pass through the pipeline effectively while maintaining regulated validation and auditability.Additionally, a more structured SQLite-backed implementation of the audit subsystem will enhance query speed, forensic analysis, and long-term audit management.Expanding Presidio recognizers to accommodate more domain-specific things, like:

- Legal references
- Organisation-specific identifiers
- Addresses
- Court matter numbers
- Custom case identifiers

Building on this, further improvements may include:

- Automated metadata stripping
- Stronger quarantine handling
- Improved audit visualisation
- Configurable validation policies
- Enhanced reporting outputs

11.4 Proposed Version 3+ Considerations

Expanded architectural and operational capabilities would be necessary for future enterprise-focused iterations of the system. One planned direction involves a role-based web portal supporting

- Authentication
- Reviewer workflows
- Administrative oversight
- Approval chains
- Attribute-Based Access Control (ABAC)

Integration with Security Information and Event Management (SIEM) platforms would be another significant improvement, allowing for the central monitoring of security events and procedure errors.

Future deployments may also incorporate:

- Air-gapped audit log shipping
- Encrypted artefact storage
- Hardware-backed key management
- Distributed processing nodes
- Secure API integration layers
- Enterprise document management integration

11.5 Statement on the Meaning of “Safe Copy”

It is important to clearly define what the term “safe copy” means within the context of this system.Following the application of validation, redaction, and pseudonymization procedures, the resulting safe copy is a risk-reduced version of the original document. Its goal is to minimise the possibility that downstream handling and distribution will reveal personally identifiable information. Nevertheless, complete anonymity or the complete erasure of all sensitive data are not guaranteed by the presence of a secure copy. Due to their unpredictable nature, NLP-based de-identification algorithms may not be able to

recognise contextual identifiers, indirect references, embedded metadata, or uncommon entity formats. Therefore, rather than being a mathematically guaranteed anonymous document, the term "safe copy" should be understood as a sanitised and low-risk document. The whole security strategy still requires human evaluation, governance oversight, and operational controls.

12. Conclusion

The backend design for a privacy-preserving de-identification workflow created especially for handling sensitive legal documents was presented in this report. The system was designed to protect raw PII. Rather than operating as a straightforward upload-and-redact application, the system was structured around strict trust separation, controlled data movement, auditability, and layered security controls to reduce both external and insider risks.

Each stage of the pipeline contributes to that protection in a different way. The ingestion and validation stages act as the first security boundary by preventing unsafe or malicious files from progressing further into the system. The processing and redaction stages ensure that raw PII is only exposed where absolutely necessary before being removed, replaced, or pseudonymised. Storage separation and trust-zone boundaries help reduce unnecessary exposure between components. While the mapping register introduces controlled re-identification capability under restricted access conditions. Structured audit logging and approval checkpoints provide visibility into how documents move through the workflow and who interacted with them at each stage. Finally, the admin review and release stage adds an important human oversight layer before any output is distributed externally.

This backend report also works alongside the governance report to provide the full picture of the system. While this document focuses on the technical architecture, validation logic, security controls, and processing workflow. The governance report explains how these controls are managed operationally through policies, review procedures, evidence handling, and accountability requirements. The two reports demonstrate how both technical security and governance considerations are integrated into the overall design.

Overall, the system is designed to create a controlled processing environment where document handling can be explained, monitored, and reviewed from start to finish. The proposed workflow aims to provide a document processing pipeline that is traceable, auditable, and defensible when handling sensitive legal information.

Subject: 31261 Internetworking Project

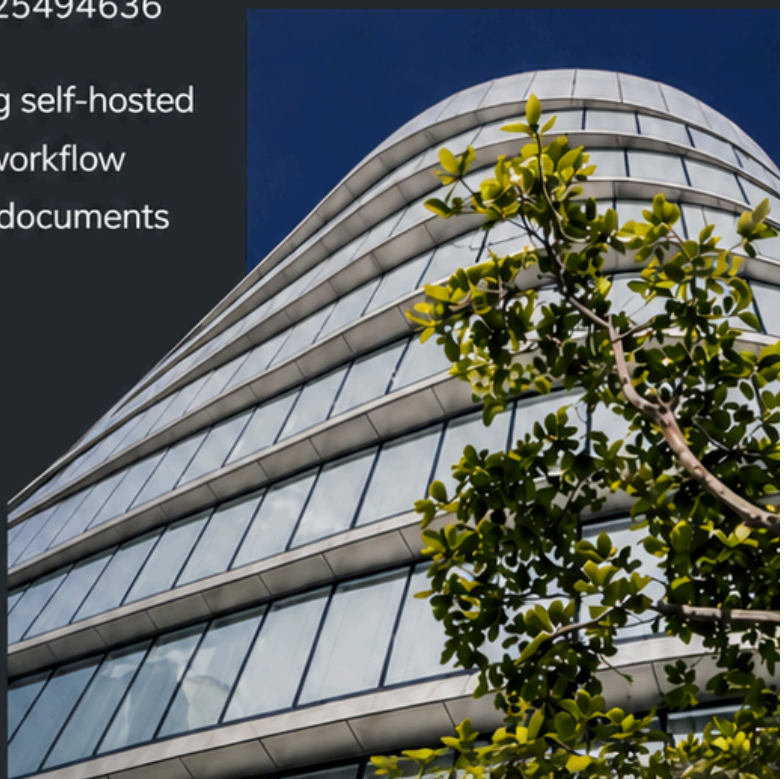
Autumn 2026

DEVELOPMENT REPORT

Privacy-preserving self-hosted de-identification workflow for sensitive legal documents

---

Project supervisor	Dr Yi Zhang
Legal supervisor	Dr Susanne Lloyd-Jones
Technical supervisor	Runsong Jia
Team member(s)	Alexander Harb: 24715841 Ajay Yuvaraj: 24934301 Lori Basmajian: 24563090 Ishaan Pathak: 24963593 Nicholas Hatzidimitriou: 25471398 Adrian Martinez: 25494636
Project / working title	Privacy-preserving self-hosted de-identification workflow for sensitive legal documents



Developer Report

1. Introduction

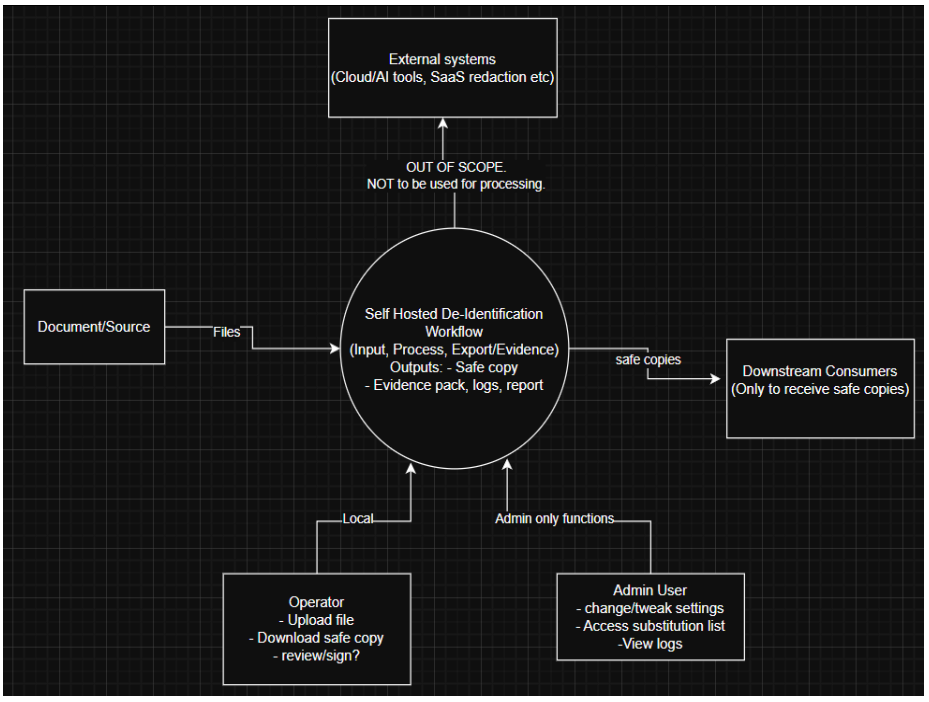
This report aims to summarise the development approach for the project. It will summarise what will be built, what is out of scope and how progress will be demonstrated.

The project will be framed as an internetworking and cybersecurity project rather than a pure software build. The architecture defines local trust zones (Input, Processing, Output), container boundaries, and network exposure assumptions to ensure the workflow is defensible from a cybersecurity standpoint. The core problem is the secure handling of sensitive documents in environments where ad hoc copying, inconsistent redaction, misconfiguration, and “shadow AI” behaviour create confidentiality and governance risk. The preferred outcome is a workflow that can consistently produce a de-identified copy that is safe to use as well as an evidence pack that supports audits and reviews whilst also avoiding revealing PII.

2. Architecture Diagrams

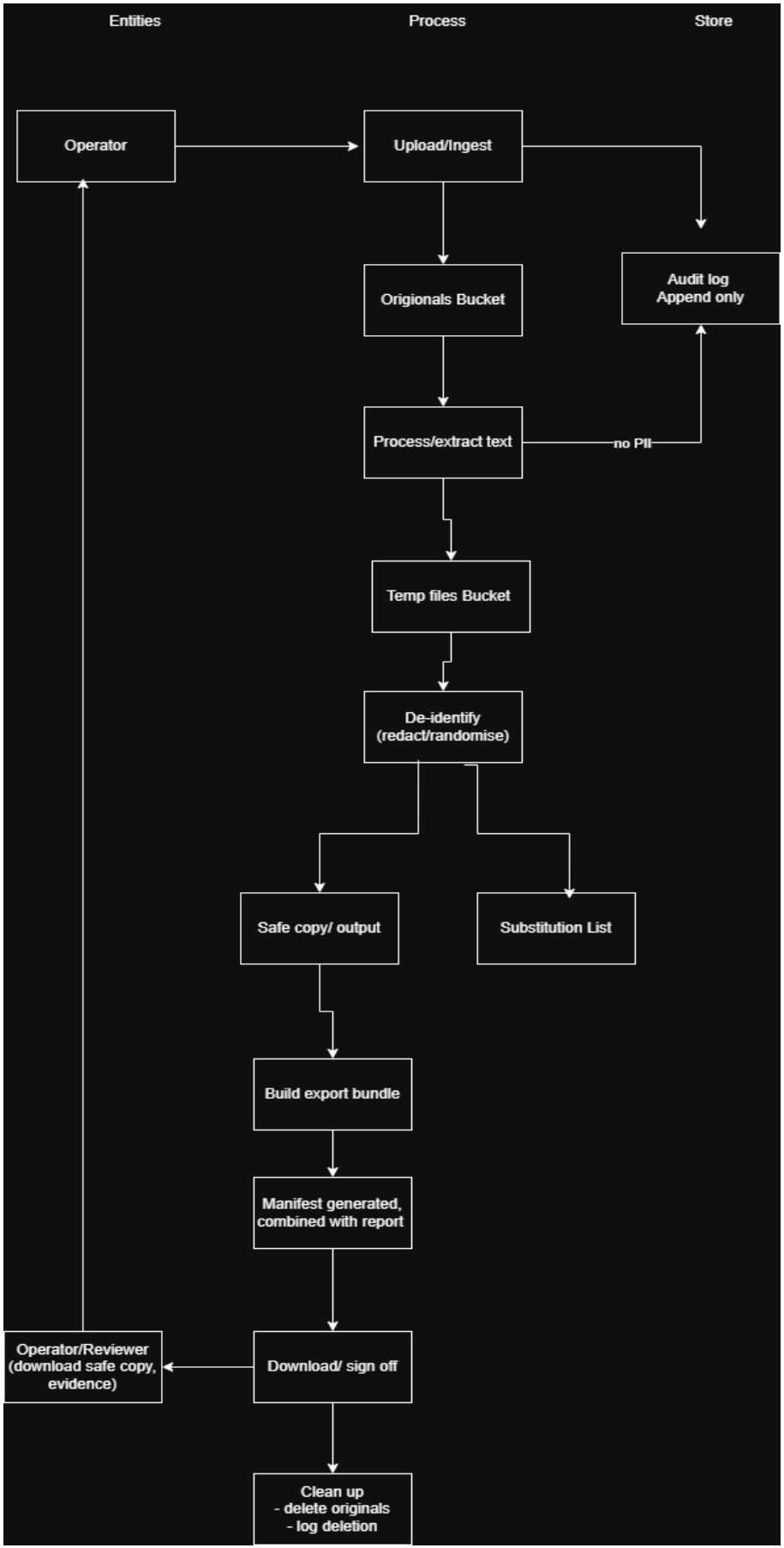
2. 1 Context Diagram

This diagram shows the systems boundaries and the key entities and systems it works with. The diagram clarifies who uploads documents, who has admin functions, what the outputs are and what is out of scope.



2.1 *Data Flow Diagram*

This diagram shows the workflow and steps in separating the sensitive data into buckets, as well as highlighting that PII is not stored whilst producing evidence.



3. Tool choice considerations

Instead of building a new machine learning system ground up, the prototype uses a pre-existing de-identification tool. Because of this choice, development stays manageable within the given timeline. Focus shifts toward strengthening workflow integrity - things like validation limits, secure processing steps, and documented results. Training and assessing models takes less room here. Effort goes where it can matter more.

What makes the last version work is its use of Microsoft Presidio - this tool fits smoothly into Python environments while allowing local deployment. Instead of relying on external services, entity spotting happens through Presidio's AnalyzerEngine, where spaCy handles core language processing tasks behind the scenes. From there, detected data points feed into a methodical redaction process, producing a clean PDF output free of sensitive details. Since traceability mattered for review purposes, each finding gets saved in machine-readable format - a kind of digital paper trail combining reports and logs. This whole chain leaves a permanent mark: every action stored once, never altered, always verifiable.

Although other options like Philter came under review, choosing one built on Java would complicate setup within a Python-dominant environment. Integration hurdles would slow down adjustments aimed at improving detection of entities and shaping output quality. Because of this, Presidio emerged as the better fit for building the working version.

Midway through testing, some types of identifiers - like dates of birth and complete addresses - were set aside on purpose. Because inconsistent results popped up often, alongside recurring matches that skewed counts. Progress here depends on finishing adjustments first. Followed by more rounds of verification work later down the line.

If tighter is what you're after, consider trimming it down further - around five or six lines could work just fine without losing impact. That shorter version still holds its ground, delivering the core idea clearly. Each sentence would carry weight, yet remain straightforward. Length does not always match power; sometimes less becomes more through precision. The message stays sharp, even when compacted. Structure bends, but meaning remains.

4. Prototype

4.1 What the prototype does.

The prototype accepts an input of a document uploaded by the user, applies PII de-identification and returns a safe copy of the document as well as audit evidence suitable for review.

4.2 Prototype implementation summary

The current build produces a “safe copy” output and a supporting evidence pack. The evidence pack is intended to support review by capturing what processing occurred and under what settings, without storing raw PII in logs (counts/hashes/IDs only). The prototype combines multiple techniques to support a defensible de-identification workflow:

Information retrieval (text extraction): PDF content is extracted using PyMuPDF (e.g., text extraction methods).

Natural Language Processing (NLP): spaCy is used as the NLP engine underneath Presidio for tokenisation and named entity recognition.

Pattern recognition / rule-based extraction: Presidio recognisers also apply pattern-based logic (e.g., regex-style recognisers for structured identifiers).

Pseudonymisation / anonymisation pipeline: Microsoft Presidio Analyzer + Anonymizer performs entity detection and replacement, with optional support for synthetic replacement via Faker (where appropriate).

Prototype web layer: a lightweight Flask web interface is used to support upload and download of outputs.

For testing, some identifier categories (e.g., DOB and full addresses) were intentionally excluded while the team investigated false positive/false negative behaviour observed during early testing. In particular, some test cases triggered repeated scanning behaviour that inflated counters into the thousands for a single page. These categories are planned to be reintroduced after tuning and evaluation improvements.

5. Challenges and Limitations

5.1 Formatting and Extracting

The current prototype specifically caters to text-based PDFs. The reliability of PDF extraction largely relies on how the document was produced (embedded text vs scanned images). The design aims to prioritize the human workflow for meeting notes, and as such scanned pdfs/OCR pipelines are out of scope. This is because of the processing time required and the risk of inaccurate information. A major limitation for the real-world coverage, but the prototype remains doable and testable in the semester timeline.

5.2 Metadata/Hidden Data

Metadata is handled as a first-class risk because “visible redaction” does not ensure that sensitive information has been removed. The prototype for PDFs removes the document metadata (author, creation date, tool chain, custom properties) using the PyMuPDF to clear metadata on safe-copy generation. Although it varies by file type, not all. The introduction of later formats (such as DOCX) would entail further work on comments, tracked changes and revision history.

5.3 Accuracy

When refining and fine tuning the program a trade off will be presented where if the tool is too sensitive to recurring patterns it may remove information unnecessarily or if it is not sensitive enough it may not remove information that it should.

6. Implementation details

6.1 Technology stack and dependencies

The current prototype is implemented using a lightweight Flask web application (session-based login and role display). The PDF processing pipeline uses PyMuPDF for text extraction and PDF redaction/annotation, and Microsoft Presidio for PII entity detection. spaCy is used as the underlying NLP engine for entity recognition, and Faker is used to generate replacement values for structured numeric identifiers.

6.2 Application routes

The prototype exposes the following routes:

- /login (GET/POST): authenticates users and assigns a role (admin or user). Failed logins are recorded in the audit log.
- /logout: clears the session and logs the logout event.
- / (GET/POST): main operator page. GET renders the upload UI; POST processes an uploaded PDF and returns output links.
- /download/<filename>: allows authenticated users to download generated artefacts (safe copy, manifest, governance report).
- /logs (admin-only): shows the most recent audit events in the browser (restricted to admin role).

6.3 Validation boundary

Before processing begins, uploads are validated:

- only .pdf files accepted and size is limited (50 MB maximum)
- PDF header must begin with %PDF- (rejects disguised/non-PDF files)
- Validation outcomes are logged as structured audit events.

6.4 Processing pipeline

For each upload, the system generates a job_id (UUID) and performs the following:

1. save the uploaded PDF to a temporary file for processing
2. extract text from each PDF page using PyMuPDF (page.get_text())
3. detect PII entities using Presidio Analyzer (current entity set: PERSON, ORG, PHONE_NUMBER, AU_ABN)
4. build a pseudonym map and generate a safe copy by redacting located entities and inserting replacement text
5. generate evidence artefacts:
    - manifest JSON (*_manifest.json) summarising the run
    - governance report TXT (*_governance_report.txt) with disclaimers, metadata statement, and reviewer sign-off
6. delete the temporary uploaded file at the end of processing

6.5 Replacement strategy (pseudonymisation)

For stability and layout control:

PERSON and ORG are replaced with fixed placeholders (to avoid text overflow in PDFs).

PHONE_NUMBER and AU_ABN are replaced using Faker patterns that preserve the same character format.

Replacement mappings are built per run, so repeated occurrences are replaced consistently within a single job.

7. Testing and evidence

7.1 Test approach (what we tested and why)

Testing focused on demonstrating that the prototype workflow is functionally correct, defensible, and reviewable. Evidence is captured in the form of screenshots, generated artefacts (safe copy, manifest, governance report), and audit log entries showing what occurred, when, and under which role.

7.2 Functional test cases (with evidence to capture)

7.2.1 T1 — Authentication and role behaviour

- Steps: attempt login with invalid credentials; then login with valid user; then login with admin.
- Expected: invalid login shows an error and creates a LOGIN_FAILED audit event; valid login sets session role and creates LOGIN_SUCCESS.
- Evidence: screenshot of login error; screenshot of UI showing role; audit log excerpt showing login success/failure events.

7.2.2 T2 — Access control (admin-only audit logs)

- Steps: as normal user, attempt to access /logs; as admin, access /logs.
- Expected: non-admin access is denied and logged; admin access succeeds and is logged.
- Evidence: screenshot of denial; screenshot of /logs page; audit log entries showing ACCESS_DENIED and LOG_VIEWED.

7.2.3 T3 — Upload validation boundary

- Steps:

  a) upload a non-PDF file → reject

  b) upload a renamed file with .pdf extension but not a PDF → reject

  c) upload an oversized PDF → reject

- Expected: validation fails with a clear reason; failure events logged as VALIDATION_FAILED.
- Evidence: screenshots of UI error messages; audit log excerpt showing failure reason fields.

7.2.4 T4 — Successful processing

- Steps: upload a text-based PDF containing detectable identifiers (person names, organisation, phone number, ABN).
- Expected: processing starts, safe copy is created, and results include job_id, redaction count, processing time, and download links.
- Evidence:
    - screenshot of results shown on the UI (job_id, redaction_count, proc_time)
    - the generated safe copy PDF
    - the generated manifest JSON
    - the generated governance report TXT
    - audit log excerpt showing UPLOAD_RECEIVED, PROCESSING_START, SAFE_COPY_CREATED, and PROCESSING_COMPLETE.

7.2.5 T5 — “No PII detected” handling

- Steps: upload a PDF with no detectable PII.
- Expected: the system returns “No PII detected” and logs a completion outcome.
- Evidence: screenshot of UI message + audit log entry.

7.3 Evidence artefacts to attach in the final submission

For at least one successful run (T4), attach:

- Safe copy PDF (output)
- Audit manifest JSON (evidence pack component)

- Governance report TXT (court/governance-facing summary + disclaimers)
- Audit log excerpt (JSON lines) showing the job lifecycle (start → complete).

7.4 Performance observations

The system records processing time per run and includes it in the manifest and UI result display.

For final reporting, include:

- document page count and size (approx.)
- processing time observed
- any slow cases

7.5 Security-oriented checks

- Confirm uploads are rejected early for invalid input types and size limits (DoS risk reduction).
- Confirm temp upload file is deleted after processing (reduces residual exposure).
- Confirm role-based restrictions (admin-only logs) work as intended

8. Issues encountered and mitigations

8.1 PDF “text vs layout” mismatch (entity detected but not found on page)

Issue. In PDF workflows, the text you extract (page.get_text()) may not map cleanly back to how characters are stored and rendered on the page (spans, line breaks, ligatures, encoding quirks). This can cause a mismatch where Presidio detects an entity in extracted text, but the system cannot reliably locate the exact on-page rectangle(s) to redact/replace.

Where it shows in the implementation. The prototype uses a character-level approach (get_span_chars + find_pii_rects_on_page) to locate detected entity tokens within the PDF’s internal character stream before applying redactions and inserting replacement text.

Mitigation.

- Treat “not located in PDF” as a first-class residual risk and report it (already included in the governance report output).
- Improve matching robustness (e.g., tolerate punctuation/line breaks more aggressively; add tests for split spans).
- Expand testing across multiple PDF generators (Word-exported PDFs vs scanned vs court form PDFs).

8.2 False positives / false negatives and staged entity coverage

8.2.1 Issue. Entity detection will always involve a trade-off:

- False positives reduce usability by over-redacting,
- False negatives increase privacy risk by missing identifiers.

During development, some identifier classes (e.g., DOB/addresses) produced unacceptable behaviour during testing (including runaway matching/inflated counters), so the entity set was scoped down to stabilise the prototype.

Where it shows in the implementation. The current configuration limits detection to PERSON, ORG, PHONE_NUMBER, and AU_ABN.

8.2.2 Mitigation.

- Reintroduce additional entity types only after tuning thresholds and adding regression tests (include “before/after” evidence for missed/over-redaction rates).
- Consider domain-adapted NER for legal text (planned improvement), but avoid overclaiming: it may reduce missed entities, not eliminate them.

8.3 Evidence artefacts currently contain sensitive content (needs minimisation)

8.3.1 Issue.

Governance intent is that evidence artefacts should avoid raw PII wherever possible (prefer counts/hashes/IDs). In the current implementation, the manifest includes per-entity details and can include the detected string as "original" (and the replacement used), which makes the manifest itself sensitive.

8.3.2 Mitigation.

- Move the manifest toward counts-only (e.g., counts per entity type, total redactions, job metadata, input/output hashes) and avoid storing raw matched strings in the evidence pack.
- If a mapping register is required for verification, store it separately with admin-only access (not in the standard evidence pack).

8.4 Retention and deletion gaps for output artefacts

8.4.1 Issue.

Temporary upload files are deleted after processing, but output artefacts (safe copy, manifest, governance report) remain on disk for download and there is no time-based retention policy or deletion event logging for outputs.

8.4.2 Mitigation.

- Implement a retention window for outputs (e.g., delete after N hours) and log deletion events.

- Store outputs in per-job directories to simplify cleanup and reduce accidental mixing across runs.

8.5 Deployment security is partly an assumption (not enforced by code)

8.5.1 Issue.

Some security requirements (e.g., “non-root containers”, “no outbound network calls”, “minimal exposed ports”) are deployment controls rather than application logic. The code enforces authentication and admin-only access to logs, but container hardening and network exposure limits must be documented and applied in deployment.

8.5.2 Mitigation.

- Document the deployment baseline explicitly (local-only binding, reverse proxy/VPN exposure if used, non-root container, restricted egress).

- Treat these as “deployment requirements” in the final report and evaluation evidence.

9. Requirements conformity analysis

Governance requirement	How it's met	Evidence in code/UI
end-to-end workflow — system must ingest supported formats (initially text-based PDFs) and extract contents.	Upload endpoint accepts PDFs only; validation enforces .pdf, max size, and %PDF- header; content extracted with PyMuPDF page.get_text().	ALLOWED_EXTENSIONS={".pdf"} + validate_upload() checks extension/size/header ; import_pdf() extracts text
policy-driven transformations, (redaction pseudonymisation), usable output	Presidio detects PII entities; safe copy is produced by redaction annotations + inserting replacement text (pseudonyms) into the PDF.	detect_pii() uses Presidio analyzer entities ; create_safe_copy() redacts + inserts replacements
outputs generated as part of structured bundle including transformed document artefacts	Prototype generates safe copy PDF + manifest JSON + governance report TXT per job, and exposes download links (currently as separate downloadable artefacts rather than a single ZIP bundle).	Results page provides download links for safe copy/manifest/report ; manifest generated in generate_audit_log() ; governance report generated
repeatable execution (same input, same policy version consistent outputs)	Current build pins a policy_version="v2.0" in the manifest/report and uses fixed placeholders for PERSON/ORG, making outputs more repeatable. Full deterministic behaviour still depends on stable detection results and consistent replacements for all entities.	policy_version: "v2.0" in manifest ; placeholders fixed for PERSON/ORG

simple operator interface to submit jobs, monitor progress, retrieve outputs	Lightweight web UI supports upload + shows job_id + shows counts/time + provides download links.	UI shows job_id, redaction count, processing time, and download links
enforce separation between input, processing, output stages (trust boundaries)	Separation is implemented logically: upload saved to a temporary file for processing, safe output written to new file; processing temp is deleted in finally. Full “zone” isolation is a deployment + directory structure concern (stronger in v2).	temp file created and removed in finally ; safe copy written to new filename safe_{job_id}_...
reduce identifiability; de-identification is risk mgmt not a status	UI and governance report explicitly warn safe copies are “risk-reduced, not anonymous,” and require human review.	UI notice states risk-reduced/not anonymous ; governance report “NOT anonymous” + limitations
direct identifiers removed/transformed; indirect identifiers handled to reduce risk	Current entity scope targets PERSON/ORG/PHONE_NUMBER/AU_ABN; indirect identifiers (addresses/DOB etc.) are not yet enabled due to accuracy issues and will be added later with tuning.	entity list in detect_pii()
consistent pseudonymisation to preserve document structure	Pseudonym map ensures repeated occurrences map to the same replacement within a run; PERSON and ORG use fixed placeholders for consistent readability.	build_pseudonym_map() creates consistent mapping per job ; placeholders defined
processing occurs self-hosted; raw docs not exposed externally	Prototype is a local Flask app; processing is performed locally using local libs (PyMuPDF/Presidio). No	local Flask app + local libs imported

	external API calls appear in the processing flow.	
minimisation (retain only necessary info; avoid preserving unnecessary identifiers)	Current manifest contains per-entity entries including the detected string (original) and positions. That is not minimised; governance intent is “counts/hashes/abstract refs where possible.”	build_audit_data() stores original ; manifest stores redactions list
controlled environment with trust boundaries; access restricted by role/necessity	Authentication required for main route; admin-only route for log viewing; access denials are logged.	login_required + admin_required enforce access ; admin logs link in UI only for admin
baseline hardening (non-root, minimal exposure, restricted inter-component comms) where feasible	These are primarily deployment controls. Code does not enforce container hardening/non-root; must be documented as deployment assumptions (v2 Docker).	Requirement is stated in governance; code does not implement container policies.
temp files isolated, protected	Upload saved to a temp file; it’s removed after processing. Protection via OS temp directory; stronger isolation (per-job directories/permissions) is future improvement.	NamedTemporaryFile(delete=False) then os.remove(tmp.name)
sensitive artefacts (substitution/mapping registers) stored separately, restricted	There is a per-run pseudonym map in memory and included in manifest via replaced_with; it is not stored as a separate protected mapping register with admin-only access.	pseudonym map built and used ; manifest includes replacements

record what processed, when, under which policy settings	Audit log records lifecycle events; manifest includes job_id, timestamp, policy_version, processing time, input/output filenames.	audit_log() writes JSONL with timestamp/user/event/job_id ; manifest fields in generate_audit_log()
append-only event log for each job	Log is append-only JSON lines in logs/audit.log. It includes job_id so events can be associated with a run.	audit_log() appends a JSON object per line
structured manifest summarising config/outputs/metadata	Manifest JSON produced per run; includes output references and policy version; governance report includes metadata statement.	manifest generation ; governance report metadata section
traceability without exposing raw identifiers; prefer counts/hashes	Audit log avoids raw PII (design intent); however the manifest currently contains per-entity original strings. Needs refactor to counts/hashes-only manifest to align with governance requirements.	audit log comment “Raw PII values are never written here” ; manifest stores original entries
remove/normalise metadata where feasible; disclose limitations	Safe copy clears PDF metadata using set_metadata({}). The governance report explicitly states what metadata handling was applied for PDF.	metadata clearing in safe copy ; governance report metadata section
raw inputs only retained as long as needed; temp files removed immediately; no residual raw data	Uploaded PDF saved to temp file and removed in finally after processing; lifecycle events logged.	temp deletion ; processing events logged

outputs/logs/manifests follow defined policies; deletion is verifiable	Uploaded PDFs are saved to a temp file for processing and deleted after the job completes. Audit events are written append-only to logs/audit.log. Output artefacts persist for download; time-based retention and deletion events for outputs are not implemented.	Temp deletion in finally (os.remove(tmp.name)); append-only JSONL audit log; downloads via /download/<filename>.
each run produces safe copy + manifest + event log + governance-facing report	Safe copy PDF + manifest JSON + governance report TXT produced per job; event log exists as logs/audit.log with job_id correlation.	safe copy created ; manifest generated ; governance report generated ; audit log appends events
structured + portable format; include hashes	Safe copy, manifest, and governance report are generated per job and downloadable as separate files. A single ZIP bundle and SHA-256 integrity hashes are not implemented	Manifest and governance reports are written to disk; downloads are served individually via /download/<filename>; no bundle/hash fields or bundle creation step in the current routes.

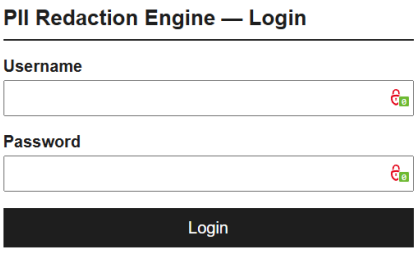
10. User Guide

10.1 Running the prototype (local)

1. Install dependencies:
    a. pip install -r requirements.txt
2. Install the spaCy model used by Presidio:
    a. python -m spacy download en_core_web_lg
3. Start the Flask app:
    a. python app.py
    b. The server runs on port 5000 by default.
4. Open the application in a browser:
    a. http://127.0.0.1:5000/login

10.2 Logging in

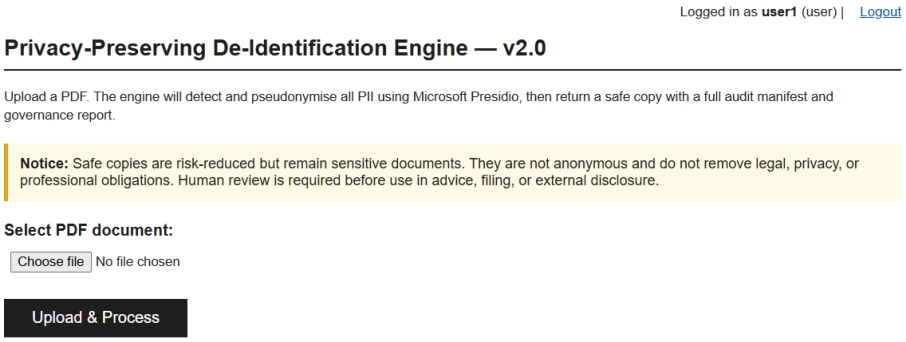
- Use the login page to authenticate.
- The system supports two roles: user and admin. Admins can access audit logs.



10.3 Processing a document

1. After logging in, go to the main page and upload a PDF using the “Choose file” button.

2. Upload and process the document to begin processing.
3. The system will:
    - validate the upload (type/size/PDF header)
    - extract text and detect PII
    - generate a safe copy and evidence artefacts

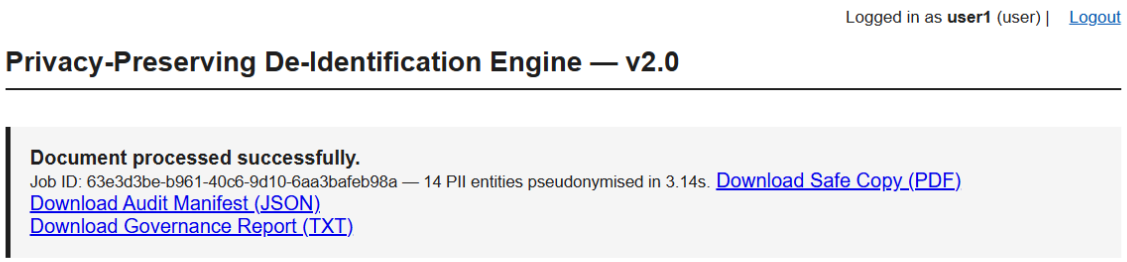


10.4 Downloading outputs

After processing completes, the page provides download links for:

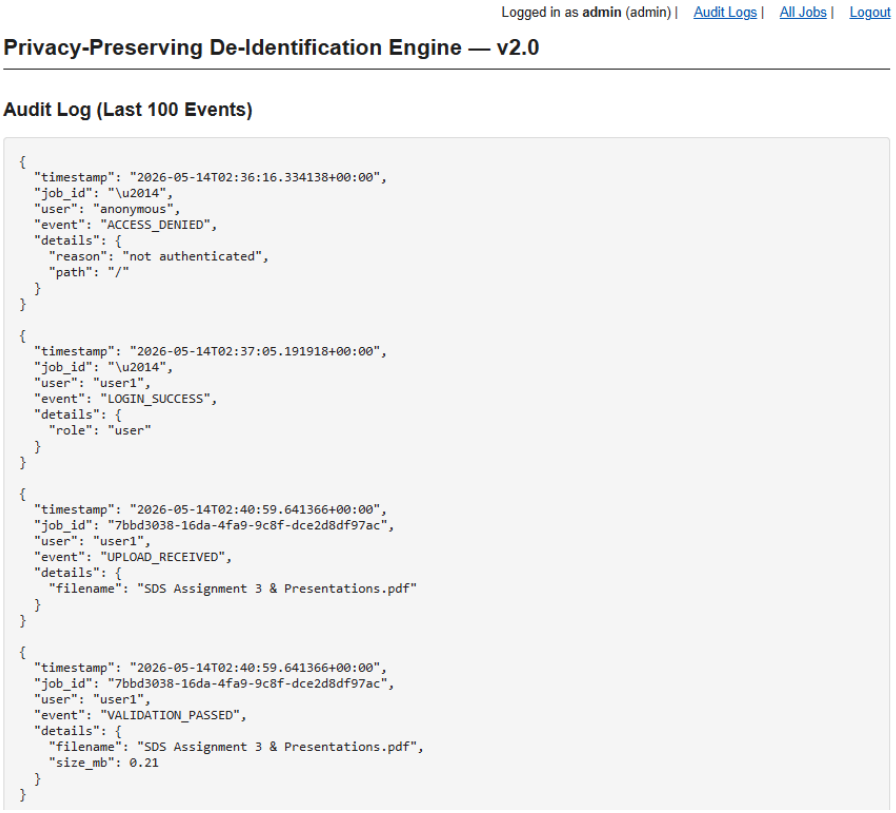
- Safe Copy (PDF)
- Audit Manifest (JSON)
- Governance Report (TXT)

The UI also displays the job_id, redaction count, and processing time.



10.5 Viewing audit logs (admin only)

- Admin users can open /logs to view recent audit events.



10.6 Important usage note

The system treats the safe copy as risk-reduced, not anonymous, and human review remains required before reliance-critical use or external disclosure. This warning is displayed in the UI and reinforced in the governance report.

11. References

Artifex Software. (n.d.). Explore text searching with PyMuPDF. Retrieved April 17, 2026, from https://artifex.com/blog/explore-text-searching-with-pymupdf

DataCamp. (n.d.). Creating synthetic data with Python Faker (tutorial). Retrieved April 17, 2026, from https://www.datacamp.com/tutorial/creating-synthetic-data-with-python-faker-tutorial

DigitalOcean. (n.d.). How to add authentication to your app with Flask-Login. Retrieved April 17, 2026, from https://www.digitalocean.com/community/tutorials/how-to-add-authentication-to-your-app-with-flask-login

Faker. (n.d.). Faker documentation. Retrieved April 17, 2026, from https://faker.readthedocs.io/en/master/

joke2k. (n.d.). Faker (GitHub repository). Retrieved April 17, 2026, from https://github.com/joke2k/faker

manjuraavi. (n.d.). PDF PII redaction tool (GitHub repository). Retrieved April 17, 2026, from https://github.com/manjuraavi/pdf-pii-redaction-tool

Medium. (n.d.). Manipulate text in Python using PyMuPDF. Retrieved April 17, 2026, from https://medium.com/@pymupdf/manipulate-text-in-python-using-pymupdf-8ecda25ef754

Microsoft. (n.d.). Presidio. Retrieved April 17, 2026, from https://microsoft.github.io/presidio/

Microsoft. (n.d.). Presidio sample: PDF annotation (Python). Retrieved April 17, 2026, from https://microsoft.github.io/presidio/samples/python/example_pdf_annotation/

PyMuPDF. (n.d.). PyMuPDF documentation. Retrieved April 17, 2026, from https://pymupdf.readthedocs.io/en/latest/

PyMuPDF. (n.d.). PyMuPDF: App1 documentation. Retrieved April 17, 2026, from https://pymupdf.readthedocs.io/en/latest/app1.html

PyMuPDF. (n.d.). PyMuPDF: Text recipes. Retrieved April 17, 2026, from https://pymupdf.readthedocs.io/en/latest/recipes-text.html

PyMuPDF. (n.d.). Issue #3278 (GitHub issue). Retrieved April 17, 2026, from https://github.com/pymupdf/PyMuPDF/issues/3278

spaCy. (n.d.). EntityRecognizer. Retrieved April 17, 2026, from https://spacy.io/api/entityrecognizer

spaCy. (n.d.). Linguistic features. Retrieved April 17, 2026, from https://spacy.io/usage/linguistic-features

spaCy. (n.d.). spaCy 101. Retrieved April 17, 2026, from https://spacy.io/usage/spacy-101

Stack Overflow. (n.d.). PyMuPDF page.search_for text splits a line break into two completely different... Retrieved April 17, 2026, from https://stackoverflow.com/questions/78753843/pymupdf-page-search-fortext-splits-a-line-break-into-two-completely-different

Towards Data Science. (n.d.). Building a customized PII anonymizer with Microsoft Presidio. Retrieved April 17, 2026, from https://towardsdatascience.com/building-a-customized-pii-anonymizer-with-microsoft-presidio-b5c2ddfe523b/

Flask-Login. (n.d.). Flask-Login documentation. Retrieved April 17, 2026, from https://flask-login.readthedocs.io/en/latest/

12. Conclusion and next steps

12.1 Summary of contribution

This project developed a governance-led, self-hosted workflow for preparing sensitive legal documents for controlled downstream use in AI-assisted environments. The central contribution is not simply that the prototype can transform identifiers. Rather, the project demonstrates how de-identification can be treated as a controlled process supported by trust boundaries, policy-driven transformation, auditability, metadata awareness, reviewer oversight, and evidence generation.

The workflow addresses a practical gap in current legal-document handling. Existing tools may assist with redaction, de-identification, or enterprise AI processing, but they do not always provide a complete, SME-appropriate workflow that explains what was processed, what was changed, under which settings, and with what remaining limitations. This project responds to that gap by combining a pseudonymised safe-copy output with an evidence pack containing a manifest, audit log, and governance-facing report. Together, these artefacts support review, reconstruction, and accountability, rather than relying on a transformed document alone.

The project also establishes a clear governance position. A safe copy is risk-reduced, not anonymous. It may still contain personal information, confidential material, privileged content, or contextual detail that creates residual disclosure risk depending on the recipient environment. For that reason, the workflow does not replace legal judgment, source verification, privilege review, or organisational governance. Its value lies in reducing unnecessary exposure of direct identifiers and making downstream document handling more controlled, reviewable, and defensible.

12.2 Deployment readiness notes

The prototype provides a workable baseline for demonstrating the project's core workflow, particularly for text-based PDF inputs within a self-hosted or firm-controlled environment. It shows how sensitive documents can be ingested, processed, transformed, and exported with supporting evidence artefacts. It also demonstrates how governance requirements can be translated into backend requirements, including audit logs, policy traceability, metadata statements, mapping-register separation, and reviewer sign-off.

However, the prototype should not be treated as production-ready without further development and organisational controls. Real deployment would require stronger

integration with identity and access management, document management systems, retention policies, security monitoring, backup processes, incident handling, and formal approval workflows. It would also require clearer operational ownership, user training, and a privacy or legal risk assessment suited to the organisation's actual environment.

The deployment position is therefore deliberately cautious. The project shows that a controlled workflow is feasible and valuable, but it does not claim organisational compliance, certification, legal advice, or guaranteed anonymisation. The prototype should be understood as a demonstrator and governance baseline, not a complete enterprise system.

12.3 Future work roadmap

Future development should focus on both technical maturity and operational maturity.

On the technical side, further work should improve detection accuracy, expand supported identifier categories, strengthen metadata handling, and extend support beyond text-based PDFs where this can be done safely. Additional work is also needed to improve handling of scanned documents, image-based content, complex layouts, dates of birth, addresses, court references, and other legal-domain identifiers. Any expansion of supported formats should be carefully validated because broader file support increases both technical complexity and security risk.

The evidence model should also be refined. Future versions should strengthen manifest and audit-log minimisation so that evidence artefacts do not unnecessarily record raw sensitive values. Output retention should be governed by clear deletion rules and deletion-event logging. Integrity artefacts, such as hashes or checksums, should be more consistently used to support later verification.

On the operational side, future work should develop stronger reviewer workflows, release/share registers, role-based or attribute-based access controls, dual-approval processes for higher-risk disclosure, and clearer integration with enterprise governance systems. For real-world use, downstream AI disclosure should remain subject to approval, contractual controls, purpose limits, and ongoing review of recipient environments.

Overall, the project demonstrates that safer use of sensitive legal documents in AI-assisted environments requires more than redaction. It requires a governed workflow that combines technical controls with explicit assumptions, review points, evidence artefacts, and honest residual-risk reporting. The prototype and accompanying governance framework provide a practical foundation for that approach.