

OTERIA CYBER SCHOOL

APPRENTICESHIP THESIS, MASTER'S DEGREE IN CYBERSECURITY

A Method for Sovereign and Resilient Enterprise AI

From threat prevention to value creation

English translation of the French original

Student	Thomas Housset
Company	Advens
Apprenticeship mentor	Tanguy Ekizian, senior consultant, Advens
Thesis supervisor	Paul De Montalivet, Oteria
Apprenticeship period	September 2024 to August 2026
Academic year	2025-2026

Contents

1	Context	1
1.1	Advens	1
1.2	The team and my role	1
2	Introducing the project	2
2.1	Origins	2
2.2	Objectives and intended impact	2
2.3	State of the art	3
2.4	Main stages of the project	4
2.5	My role in the project	4
3	Project description	5
3.1	The framework as a whole	5
3.2	The five phases	6
3.3	Focus 1: mapping the risks of language models	7
3.4	Focus 2: comparing architectures, sovereignty versus performance	8
3.5	Resilience: holding up when the provider lets you down	10
3.6	Focus 3: an operational governance framework	11
3.7	Mapping a use to its regulatory obligations	12
3.8	Putting the framework to the test on real tools	13
4	Putting AI to work: the value at the end of the framework	14
4.1	Why integrate, and what deciding nothing costs	15
4.2	AI in service of defence	15
5	Project management	16
5.1	Methods and tools	16
5.2	Managerial organisation	16
6	Results	17
6.1	Meeting the initial objectives	17
6.2	Unforeseen impacts	18
6.3	The comparison matrix as a tangible result	18
6.4	Early lessons from the complementary study	18
6.5	Points of comparison from CISOs	19
7	Critical analysis	19
7.1	What worked well	19
7.2	What worked less well	19
7.3	What could have been done differently	20
7.4	The field study: a deliberate broadening of scope	20

7.5	Outlook: agentic AI shifts the question to identity	20
7.6	Analysis of my personal contribution	20
8	Personal learning	21
9	Conclusion	21
10	Acknowledgements	22
	References	23
A	Appendices	24
A.1	Glossary of acronyms	24
A.2	The framework: reference document	25
A.3	Full threat model	27
A.4	Technical controls toolkit	28
A.5	Detailed use case sheets	29
A.6	Worked end-to-end example	29
A.7	Maturity grid: scoring criteria	30
A.8	Example maturity grid	30
A.9	Scoring method and detailed campaign scores	31
A.10	Decision tree	31
A.11	Architecture cost model	32
A.12	Reversibility: exit strategy diagram	33
A.13	Mapping the framework to reference standards	34
A.14	AI Act timeline	34
A.15	Sovereignty levels	35
A.16	Data-flow diagram and trust boundaries	35
A.17	Policy as code: a complete example	36
A.18	Structure of the complementary study	37
A.19	Comparison points gathered from CISOs, by sector	37

1 Context

Before turning to the project, I need to say where I carried it out from. My job, my company and my team go a long way towards explaining the shape this work took and the constraints I set myself.

1.1 Advens

Advens is an independent French company whose sole business is cybersecurity. It brings together more than six hundred experts in France and several other European countries, and covers the whole chain, from governance and compliance consulting to offensive security, with incident response and solution integration in between. Its operational core is mySOC, its managed security operations centre, a detection and response platform holding ANSSI's PDIS qualification (security incident detection service provider).

This positioning has a direct effect on how I work. I almost never work on Advens's own information system; I work on those of its clients, with their tools and their level of maturity, which varies a great deal from one engagement to the next. Depending on the subject, the work involves governance and risk, securing cloud environments, or running integration projects. My own path has moved between these areas, and that is what allowed me to approach AI integration from both a compliance angle and a technical one.

1.2 The team and my role

I work as a cybersecurity consultant, on a scope that combines governance, risk and compliance, technical subjects and project management. In practice, I am seconded to clients, most often working autonomously on a specific work package, sometimes within a larger team led on the client side. This autonomy matters for understanding what follows: it is what allowed me to turn a vague need into a concrete deliverable.

A seconded consultant has to switch stance quickly: more advisory with a security manager, more technical with engineers, sometimes within the same day. And from one engagement to the next, organisations vary widely in maturity. One requirement always comes back: what we hand over has to hold up without us once the engagement is over. This constraint directly shaped how I designed the deliverable described later, which is built to be taken over by others.

Over these two years, my engagements covered several areas. I carried out cyber maturity assessments in mergers and acquisitions; secured Azure cloud environments (posture, landing zones, and privileged identity and access management); did secure engineering on infrastructure as code and continuous integration pipelines; conducted NIS2, ISO 27001 and GDPR gap analyses; and ran crisis management exercises for large enterprises.

This mix explains the angle I took. A purely compliance-driven approach would probably have produced an AI policy; a purely technical one, a reference architecture. I tried to hold both together, because that is what the engagements called for: an answer that speaks to the CISO as

much as to the engineer who has to plug the tool in.

One last area: software integration project management. Needs analysis, comparison of solutions, integration, testing, presentation to the client, often all carried out on my own. It is this habit of running an integration from end to end that served as the template for the project described in this thesis.

2 Introducing the project

2.1 Origins

The project grew out of an observation I kept making on engagements. At clients, generative AI was taking hold everywhere but almost never by deliberate decision: an office assistant switched on because it came with the licence, a model plugged into an automation workflow, an AI feature suddenly offered in a tool where personal data had already been stored. Each time, data was flowing to a third-party model without anyone having really decided that it should. And where the IT department blocked AI, users switched to their personal accounts: shadow AI, a blind spot I kept running into throughout the project.

This growth in AI use showed up in my engagements. I was increasingly asked to deal with AI governance and risk, either to scope a new use before it went into production, or to look at what was already running and say whether it was acceptable. These two questions come up constantly, and they do not call for the same answer. The first requires building properly. The second requires auditing what already exists, measuring the gaps and proposing remediation. That is where the project started: from the idea that a consultant should be able to answer both with a single framework.

The client contexts were varied, and I present them here in anonymised form, as I committed to doing. I worked mainly with organisations sensitive to the residency of their data: an operator of essential services subject to NIS2, a large industrial enterprise, public bodies handling personal data. This profile has a bearing on the subject. When part of the data is subject to strict constraints, the question of where it is processed and under which law stops being theoretical.

Finally, there was an element of anticipation: building this framework now meant preparing Advens and its clients for a regulated subject that is only going to grow.

2.2 Objectives and intended impact

The aim was not to produce yet another paper on the dangers of AI. It was to deliver something a consultant or an engineer could open on an engagement and use the same day. A framework that says, for a given use, whether it can go into production, under what conditions, and how to monitor it afterwards. A framework that works just as well for building from scratch as for assessing what already exists.

The intended impact is twofold. For Advens, a common frame of reference for handling clients' AI integrations, and material to present internally to align teams on the subject. For clients, a way

out of the binary choice between banning everything and letting everything through, replacing intuition with a method that can be defended in front of a CISO.

2.3 State of the art

The subject sits at the crossroads of several bodies of work, all of them already substantial. The point is not to recite them but to see what they cover, where they converge, and what they leave for the person making the decision on the ground to settle. The field does not lack frameworks; it lacks a bridge between those frameworks and the concrete decision, and that is where this work comes in.

On the regulatory side, the European Union has adopted the AI Act, which classifies AI systems by level of risk and imposes obligations that increase with that level [19]. It applies in stages, and this is a point I remind clients of. The prohibited practices and the AI literacy requirement have applied since February 2025, the obligations on general-purpose models since August 2025, and those on high-risk systems come next. The latter, initially scheduled for August 2026, have been postponed to December 2027 for the uses listed in Annex III¹. This postponement illustrates what I keep repeating on engagements: we are dealing with a moving trajectory rather than a fixed deadline, and settling on a method early is better than chasing every date. Alongside it, the NIS2 directive widens the range of entities subject to cybersecurity obligations and governs incident notification [17], the DORA regulation does comparable work for the financial sector, and the GDPR remains the reference as soon as personal data enters an AI processing operation. These texts do not speak the same language, and a concrete AI use often falls under several of them at once. That is one of the difficulties the framework had to absorb.

As for risk management methods, several references structure the landscape and already cover a great deal. The NIST AI Risk Management Framework proposes an approach organised by functions: govern, map, measure, manage [13]. The ISO/IEC 42001 standard describes an AI management system following the logic of the management system standards companies already know [8]. ENISA's multilayer framework organises AI security into three levels: cybersecurity foundations, AI-specific aspects, and sector-specific application [3]. On the industry side, Google has published its Secure AI Framework, built around six core elements, from security foundations through to the contextualisation of risks [5], and Gartner has popularised the notion of AI trust, risk and security management, which emphasises model inventory and runtime inspection [4]. All of them provide a solid backbone. Their common limitation is that they stay at a level of abstraction that does not tell an engineer what to do with a specific tool at the moment a decision has to be made.

For the French context, the ANSSI guide on generative AI systems is the reference closest to my own needs [1]. It sets out thirty-five recommendations organised by phase (training, integration and deployment, then production), and it classifies attacks into three main groups: manipulation,

¹Digital Omnibus on AI, political agreement of 6 and 7 May 2026 confirmed by the Member States on 13 May, still provisional at the time of writing. The details of the postponements are given in the appendix (figure 8).

infection and exfiltration. My framework extends this guide: it adopts its phase-based structure and its logic of controls specific to each stage, and it carries on where the guide stops, down to the decision on a specific tool on the market, with an entry point through auditing what already exists. That step into practice is what it contributes.

For mapping threats, two resources stood out in practice. The OWASP Top 10 for Large Language Model Applications lists the most common failures, from prompt injection to leakage through outputs [15]. The MITRE ATLAS knowledge base maps attack techniques targeting AI systems, in the spirit of what ATT&CK did for conventional security [12]. Both sources are concrete, and the project’s threat model is built on them.

That leaves the question of sovereignty, less codified but central for my clients. An AI provider subject to US law exposes the data it processes to extraterritorial access mechanisms, first and foremost the CLOUD Act, supplemented by Section 702 of FISA. Using such a service for personal or strategic data means accepting that risk, even when the advertised hosting is European. The authorities’ recommendations stress data residency and reversibility, but they leave each organisation to draw its own line. On engagements, we are the ones who help the client draw it, and this is one of the places where the framework adds something to the existing references, which tread carefully on this legal ground.

One last strand deserves a mention: the use of AI in defensive operations themselves, from alert triage to analysis support. It is a reminder that AI is also a tool for security, which ties in with the title of this thesis: moving from threat prevention to value creation.

On balance, the state of the art provides solid principles, threat catalogues and governance structures. What it tends to leave aside is support for the concrete decision, on a given tool, with given data, and a client waiting for an answer. That last step is what the framework takes on: carrying principles all the way down to the final call. I give the detailed phase-by-phase mapping to these frameworks in the appendix (table 11).

2.4 Main stages of the project

The project unfolded in four stages. First, observing and collecting the uses actually encountered on engagements. Next, building the framework itself, its method and its tools. Then testing it against concrete use cases, which served to refine it. Finally, shaping it for use on engagements and presenting it internally. A complementary study (an online questionnaire left open) was added in parallel, as an exploratory strand whose limitations I discuss later.

2.5 My role in the project

I carried this project from start to finish. Nobody had commissioned it; I formulated the need myself, based on what I was seeing on engagements. The method, the grids, the threat model and the governance: I assembled and wrote them, drawing on the frameworks cited above. The structuring choices are mine (reasoning by type of use rather than by product, holding compliance

and technical concerns together, providing a mode for existing uses as much as for new ones), and I revisit them in the critical analysis.

3 Project description

This is the core of the thesis. I first describe the framework as a whole, then the three components where my contribution was greatest, and I end by testing them against real cases.

3.1 The framework as a whole

The central deliverable is a method for integrating AI into the enterprise, together with grids and templates that can be used directly, on client engagements as well as internally. It is a deliverable in its own right, designed for Advens and beyond, which this thesis presents and analyses, and whose reference version is reproduced in the appendix. In other words, the thesis tells the story of how this framework was built, and the framework itself accompanies it as an operational document. Figure 1 summarises it at a glance: how an AI use, whether new or already in service, becomes a reasoned decision. I made two design choices that set it apart from a mere policy.

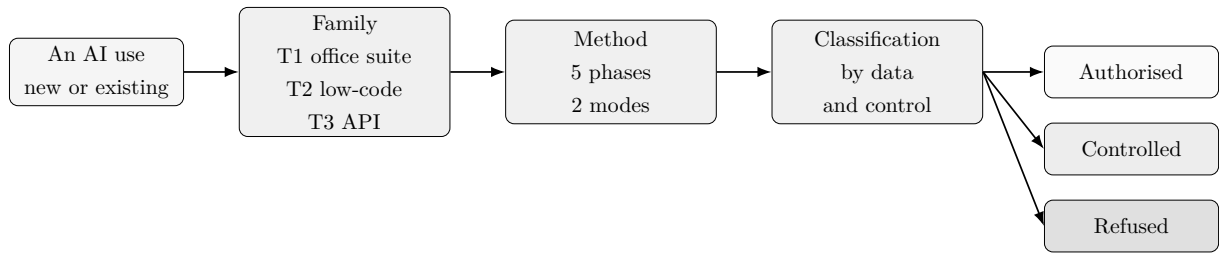


Figure 1. Overview of the framework: from an AI use to a reasoned decision, reassessed whenever the use or the offering changes.

The first choice is to reason by type of use rather than by product. Writing rules for one assistant, then for another, then for the next fashionable tool, means forever chasing the market. I therefore grouped integrations into three families. The first covers AI assistants in the office suite, where the model accesses the user's document scope. The second covers low-code or business tools that embed AI, where data flows between connectors and the tool already holds permissions. The third covers calls to AI APIs in custom integrations, where everything has to be built but you keep full control. A new tool falls into one of these families and inherits its specific concerns, without the framework having to be rewritten. Figure 2 summarises the three families and the issues specific to each.

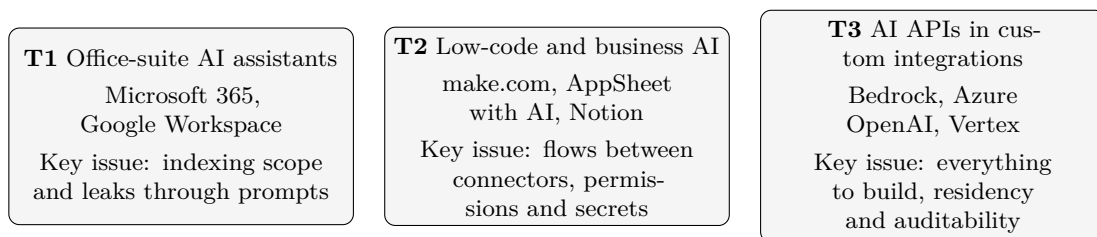


Figure 2. The three AI integration families adopted by the framework.

The second choice is to serve two situations with a single method: integrating a solution from scratch, and assessing then improving a solution already in place. In the field, it is often the second that comes up first, because AI arrives before governance. The method is then worked through as an audit grid, each control point becoming a finding, met or not met, which feeds a maturity grid and a remediation plan. The same tool therefore serves both to build and to correct, which avoids maintaining two separate baselines.

The method runs in five phases, each closed by a checkpoint. A checkpoint that has not been cleared blocks go-live, but not the analysis work. I first detail these five phases, then go deeper into the three that carry most of the technical value and where my contribution was clearest.

3.2 The five phases

Figure 3 gives an overview of the method: its five phases, the checkpoints that separate them, and the iterative back-and-forth between them.

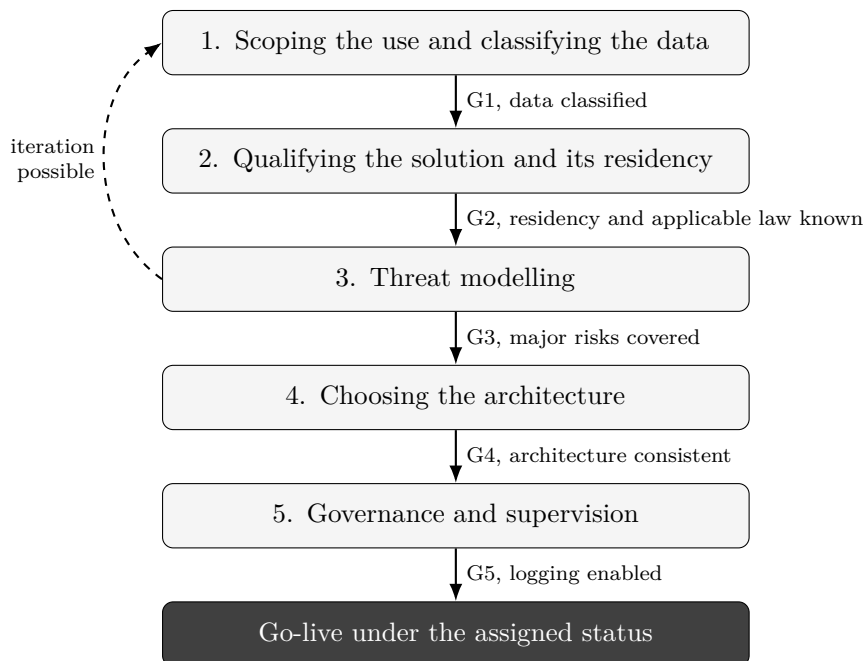


Figure 3. The five-phase method, with its checkpoints and feedback loops.

The first phase scopes the use. It is described in one operational sentence (who does what, on which data, for what gain), and above all the data involved is classified, from the most public to personal or sensitive data. This step seems obvious, but it is not. On engagements, the first difficulty is often that nobody has clearly established which data would go into the tool. The checkpoint is simple: without data classification, there is no moving forward, because the classification drives everything else.

The second phase qualifies the solution and the data residency. It establishes where the data lives and where it transits, which subprocessors are involved, whether there are transfers outside Europe, which law applies to the provider, how the service is accessed, and what the provider does with the data: does it retain it, does it reuse it to train its models? The recurring sticking

point lies here. A tool that offers AI on spaces where business teams have already stored personal data has that data processed by the provider, and if it is hosted in the United States, you are de facto accepting extraterritorial jurisdiction. The checkpoint requires known residency, documented applicable law, and access in the company’s name, not through a personal account.

The third phase models the threat, the fourth chooses the architecture, and the fifth places the use under governance and supervision. These are the three focus areas I develop in detail below, because that is where the framework’s technical value is concentrated and where my contribution was greatest. I add a resilience component that follows directly from the architecture choice, so that continuity is not forgotten once the solution has been selected. Before going into them, a remark on how the phases connect. They are not a one-way administrative pipeline; they loop back. An unfavourable sovereignty qualification can send the use back to scoping to reduce the data scope. A threat model that reveals an uncovered risk can force a different architecture choice. It is this iterative character that distinguishes a useful method from a mere box-ticking exercise.

3.3 Focus 1: mapping the risks of language models

The first building block is a reference threat model, designed to be reused for any use that relies on a language model, and then extended to applications with embedded AI.

I built this model starting from documented failure modes, in particular the OWASP Top 10 for Large Language Model Applications [15], and then mapped them to two grids familiar to security teams: STRIDE for the nature of the violation and MITRE ATLAS for the attack technique [12]. Each risk is then scored on three axes, likelihood, impact and detectability, on a simple three-level scale. The third axis counts as much as the other two. A risk that is hard to detect takes hold unseen, and it is often the most costly in the long run. It is this axis that justifies requiring logging even when the per-incident impact seems low. Table 1 gives a condensed version; the full version, with the ATLAS reference and the control associated with each risk, as well as a prioritisation heatmap, is in the appendix.

Risk	STRIDE	L	I	D
Direct prompt injection	Tampering	+++	++	++
Indirect injection via a source	Tampering	++	+++	+
Guardrail jailbreak	Elev. of Priv.	+++	++	++
Data poisoning	Tampering	+	+++	+
Exfiltration through outputs	Info. Disclosure	++	+++	+
Leak through user prompts	Info. Disclosure	+++	+++	++
Provider dependency	Denial of Service	++	+++	++
Model drift	-	++	++	+
Supervision drift	Repudiation	++	+++	+

Table 1. Reference threat model, condensed version. L likelihood, I impact, D detectability. Scale: the more signs, the higher the value, from + low to +++ high.

The model's value lies less in its list than in the controls I attached to it and in the way it is tailored. For a given use, only the relevant risks are kept, the score is adjusted to the context, and a decision is made: either a control is required, or the risk is accepted, on record.

A concrete example of tailoring. For an AI assistant built into the enterprise office suite, the dominant risk is information disclosure through overexposure, ahead of prompt injection. The assistant relies on the organisation's search index and respects existing permissions, and that is precisely the trap: it surfaces and summarises, all at once, everything that was already poorly partitioned [10]. A permissions error that had lain dormant and unnoticed becomes a generated answer. The threat model rates this risk as high impact and low detectability. The control therefore does not sit at the prompt but upstream, in permissions hygiene, and it can be tooled concretely: an overexposure assessment before deployment, using AI-specific data security posture management; sensitivity labels that make the model's access to encrypted content conditional on an extraction right; and search restricted to a validated scope as a stopgap while permissions are being fixed. The lesson goes beyond any one product: for this type of assistant, the real security issue is the governance of existing access rights, not the model itself. The data-flow diagram with trust boundaries given in the appendix (Figure 9) shows where these controls apply.

The most structural risk in these uses deserves a closer look: indirect injection. As soon as a model reads data to produce its answer (a web page, a document, a ticket), an attacker can slip instructions into it, and the model cannot tell the data from the instruction. Greshake and his co-authors demonstrated this on real applications: integration blurs the boundary between data and instructions, which opens the way to remote attacks with no direct contact with the user [7]. This is why, in the framework, source control and context sanitisation are not optional for a RAG-type use; they are the first-line countermeasure. The 2025 edition of the OWASP Top 10 has in fact added risks arising from RAG and agents: vector and embedding weaknesses, excessive agency (a system acting beyond what is expected of it), system prompt leakage, and unbounded consumption, which turns a use into a denial of service or a runaway bill. The threat model includes these entries, with excessive agency mapped to permissions governance and unbounded consumption to the supervision of quotas and costs.

For applications with embedded AI, I added the risks specific to these environments: plaintext secrets in an automation scenario, overexposure arising from permissions already granted to the tool, an overly broad indexing scope. The rest of the model still applies, which suggests that it remains valid beyond conversational agents alone.

3.4 Focus 2: comparing architectures, sovereignty versus performance

The second building block answers a question every client eventually asks: where should the model run? I formalised three standard configurations and a grid to decide between them, without naming a winner.

The first configuration relies on open-weight models, operated on a sovereign cloud or in-house.

Sovereignty and auditability are strong, at the cost of significant engineering effort and a lower capability ceiling than the very largest models. The second configuration calls a managed model from a major provider in a European region. Capability and latency are at their best and deployment is fast, but the law applicable to the provider needs watching and the risk of dependency is higher. The third configuration is hybrid: it explicitly routes sensitive processing to the sovereign side and the rest to the managed service. You gain capability without exposing what is critical, at the price of more complex routing and governance.

Criterion	Sovereign	EU API	Hybrid
Sovereignty, residency	High	Medium	Adjustable
Control of applicable law	High	Needs monitoring	High on sensitive data
Model capability	Medium	High	High on non-sensitive data
Latency	Variable	Low	Variable
Cost	High	Pay per use	Intermediate
Auditability	High	Medium	High
Data exposure	Low	High	Medium
Engineering effort	High	Low	High

Table 2. Comparison of the three architecture configurations.

Figure 4 places the three configurations on the two axes that matter, control and capability, and shows that none dominates the others across the board. I add, as a reference point, the global API outside the EU, which the framework rules out for sensitive data. Its capability is the same as that of the EU API, since these are the same models, and it sometimes gives earlier access to the very latest versions subject to geographic restrictions, which is what makes it tempting, but it gives up control over the applicable law.

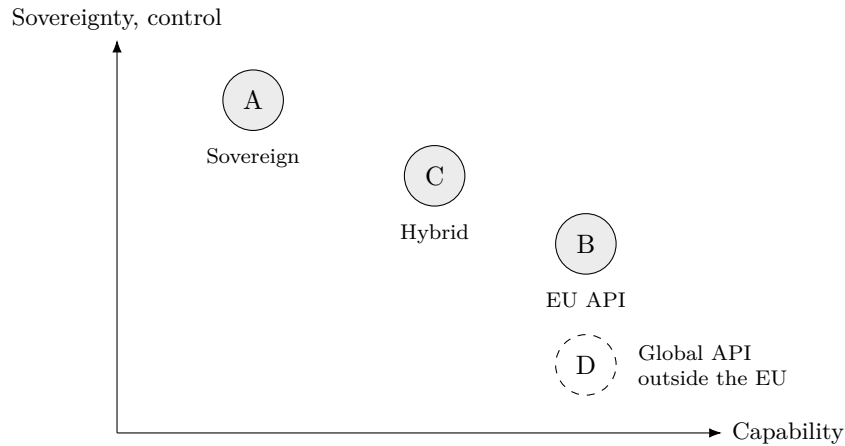


Figure 4. Positioning of the configurations. A sovereign, B managed API in an EU region, C hybrid. D, the global API outside the EU, is shown as a reference point (dashed) and is ruled out for sensitive data.

Table 2 serves to make the trade-off explicit; it does not decide on the client's behalf. A CISO can use it to defend a choice on the basis of criteria rather than preference. For sensitive data with an auditability requirement, a sovereign or hybrid configuration is called for. For a use with no confidentiality stakes where capability comes first, the managed API is justified. The precise

figures (costs and latencies) are measured engagement by engagement; the framework provides the structure of the decision. The orders of magnitude for cost that underlie this trade-off, price per million tokens versus the cost of a sovereign GPU, are given in the appendix (Table 10).

One point the table does not show comes up on every engagement, and it deserves to be stated clearly. The major providers have made significant progress on the contractual front. On Vertex AI, as with competing offerings, customer data is not used to train models by default, residency of data at rest is chosen through regional APIs, and zero data retention can be obtained by eligible customers through an amendment to the data processing agreement [6]. Managed APIs in an EU region, such as Claude on Amazon Bedrock or OpenAI models on Azure, offer comparable guarantees. This is often enough to move a use into controlled status. But it does not remove the underlying risk: a provider subject to US law remains exposed to the CLOUD Act, whatever the hosting region. Residency settles the question of where, not the question of which law applies. The framework forces this distinction to be made, and it is what separates an acceptable use from one to be avoided for truly sensitive data. The three sovereignty levels and their compensating controls, including on-premises pseudonymisation, are detailed in the appendix (Table 12).

An example of an actual trade-off decision. For a support assistant handling customers' personal data, I rule out the global API outside the EU and compare two options: self-hosted sovereign, heavier but with no legal exposure, and hybrid, a managed API in an EU region with on-premises pseudonymisation of identifying fields. At moderate volume, hybrid wins: it delivers the capability of a large model while keeping identifiable data within the controlled perimeter. The same reasoning, applied to strategic research and development data, tips towards sovereign, because there the risk of data exposure cannot be offset.

3.5 Resilience: holding up when the provider lets you down

Sovereignty answers the question of law; resilience answers that of continuity. The title of this thesis pairs them deliberately, because an AI use creates a new dependency whose extent few people grasp. The risk goes beyond a simple outage. A provider can change its prices, tighten its terms, or withdraw the model on which you have built your automations, and my study in fact surfaced this dependency as the most underestimated risk.

The framework addresses this through reversibility, designed in from the architecture phase. For a critical use, I document an exit path: which fallback model, and what effort it takes to switch to it. Migrating from a managed API to a self-hosted open-weight model is anything but instantaneous: the inference pipeline has to be re-tooled, the prompts and guardrails reworked, and the whole validated, which amounts to a few weeks of work and a real engineering cost. Planning this switch in advance, even without triggering it, makes it a real option rather than a stated principle. The diagram of this exit path is in the appendix (Figure 7).

For a client in the financial sector, this reasoning becomes an obligation. The DORA regulation requires a documented and tested exit strategy for critical digital services, as well as an assessment

of concentration risk on a single provider [18]. The exit path defined in the architecture phase is precisely the technical building block of this regulatory exit strategy, which anchors the framework’s resilience in a binding text and not merely in good intentions.

3.6 Focus 3: an operational governance framework

The third building block turns the previous decisions into rules that can be sustained over time. It is built around a classification grid, a policy written in a machine-readable format, and a supervision mechanism.

The classification grid assigns each use one of three statuses: authorised by default, controlled, prohibited. The calibration rule fits in one sentence: move up a level as soon as the data becomes more sensitive or control over residency weakens. A use on public content, with a provider offering EU residency and no reuse of data, is authorised. A use on confidential internal data moves to controlled, with mandatory controls and periodic review. A use that would send sensitive data to a service whose residency is not controlled, or that cannot be logged, remains prohibited for as long as these conditions hold.

In practice, an internal writing assistant working on confidential notes, with a provider offering EU residency and no training on the data, is classified as controlled, with filtering of sensitive data on input, classification of outputs, and logging. The same assistant, additionally connected to the human resources database, which contains personal data, remains the same tool but moves to a different class: the sources must then be restricted or the use must go back through qualification. The grid does not judge the tool; it judges the pairing of use and data, and that is what keeps it stable as tools change.

So that the policy does not stay on paper, I wrote it as policy as code, in a format that can be version-controlled. The excerpt below shows the skeleton for a controlled use.

```
use_case: internal-writing-assistant
data_class: internal_confidential
status: controlled
provider:
  residency: eu
  training_on_data: false
controls:
  input:
    dlp: true
    allowed_prompts: writing_allowlist
  output:
    classification: true
    pii_masking: true
  logging:
```

```

    enabled: true
    content: [timestamp, user, prompt_hash, output_category]
supervision:
    review: quarterly
incident:
    mapped_to: nis2_notification

```

Supervision rests on minimal but real logging: timestamp, user, a hash of the prompt rather than its plaintext content, output category, and the guardrails’ decision. I paired it with a catalogue of incident patterns, each with its response playbook and its link to notification obligations. For an entity subject to NIS2, these obligations follow a precise timeline that the catalogue incorporates: an early warning within twenty-four hours, a detailed notification within seventy-two hours, then a final report within one month of that notification [17]. Linking an AI-specific incident pattern, for example a leak of personal data through a model output, to this timeline is what turns a good intention into an obligation that can actually be met. The loop is then closed: a use is not merely authorised, it is monitored and tied to the client’s legal obligations.

3.7 Mapping a use to its regulatory obligations

A single use often falls under several legal texts at once, and the framework includes a quick step to map it to its obligations, without replacing a legal analysis. For this, I maintain a mapping table, summarised in Table 3, which for each use case ticks the applicable texts and recalls what each one concretely requires.

Text	Applies when	Key operational requirement
GDPR	Personal data is processed	Legal basis, minimisation, informing data subjects, oversight of processors
AI Act	The system falls under a regulated risk level	Transparency, documentation, human oversight depending on the level
NIS2	The organisation is an essential or important entity	Security measures, incident notification within the deadlines
DORA	The organisation is a regulated financial entity	Operational resilience, control of third-party risk

Table 3. Regulatory mapping table for an AI use.

It is a safeguard, not a legal opinion, and it avoids the most common blind spot: treating a use as a purely technical matter when it triggers specific legal obligations. It is also what makes the framework credible in the eyes of a risk management department, because it speaks their language.

One clarification prevents a common misreading. In most integrations, the company is not the provider of the model but its deployer. The deployer’s heavy obligations (human oversight, retention of logs for at least six months, informing employees) apply only if the use falls under the high-risk cases in Annex III of the AI Act, for example recruitment or the management of critical infrastructure [19], whose entry into application has been postponed to the end of 2027.

For the vast majority of productivity assistants, this regime does not apply, and the obligations come down to the transparency requirements of Article 50 and the AI literacy that Article 4 has imposed on everyone since February 2025. The framework incorporates this distinction: it triggers the heavy regime only where it is due, which avoids over-compliance that is as costly as it is pointless.

3.8 Putting the framework to the test on real tools

To check that the framework held up beyond paper, I applied it to the AI uses I encountered on engagements. The approach was the same for each: work through the five phases as an audit, score each phase from zero to three, record the gaps, then define and, where possible, implement corrective controls before scoring again. It is the difference between the two scores that measures the framework’s effect. The uses came mainly from a large enterprise that had already deployed several AI components: Gemini as an enterprise assistant, AppSheet applications using its AI features (predictive models and Gemini-based capabilities), and numerous make.com scenarios in production, some of them with embedded AI. To these were added an internal use of an enterprise AI assistant and the study of a sovereign assistant for a public operator. The organisations cannot be identified. Each score follows the framework’s maturity grid, applied phase by phase on the basis of what was actually in place, and the detailed scoring is given in the appendix.

In total, I assessed a dozen or so uses, which I group into five representative tools, the make.com scenarios alone accounting for about ten of them. I recorded around twenty-five gaps, eight of which were judged high priority because they concerned data residency or the absence of logging. Table 4 summarises the results by representative tool.

Tool	Family	Mat. before	Mat. after	Main risk	Status
Office assistant (internal)	T1	4/15	11/15	Classified content to be excluded	Controlled
Enterprise Gemini	T1	5/15	11/15	Uncontrolled residency and APIs	Controlled
make.com + Vertex	T2	6/15	13/15	APIs outside the tenant, plaintext secrets	Controlled
AppSheet with AI	T2	3/15	9/15	Permissions and data scope	Controlled
Notion AI	T2	4/15	-	Extraterritorial exposure of PII	Refused as is

Table 4. Framework test campaign: maturity scores before and after remediation. Detailed scoring in the appendix.

A few quantified findings are worth drawing out of the table. On make.com, secrets were present in plaintext in several of the ten or so scenarios examined, and some AI calls went out through keys that did not belong to the enterprise tenant, two classic flaws of these automation tools. On Notion, the spaces concerned contained personal data that was processed outside Europe, which led to the campaign’s only refusal, on the AI layer. Several of these situations deserve to be told in detail.

The case most representative of my work is that of the automation scenarios. At a large enterprise, dozens of make.com workflows were already running in production, and some of them called an AI model. As I opened them one by one, two problems kept coming up. First, several scenarios called the AI through keys that were not those of the enterprise tenant, personal or test access left in place, which took the data outside the controlled perimeter. The work consisted of routing everything back through the company’s Gemini APIs via Vertex AI, with residency and logging. Second, secrets were lying around in plaintext in scenario steps; these were fixed through centralised secrets management. Nothing spectacular, but this is exactly the kind of invisible drift the framework is designed to catch.

A second, shorter case comes from an internal use. An enterprise AI assistant is deployed, but the framework prohibits submitting certain content to it, in particular classified material linked to public-sector or highly sensitive clients. Authorising a tool does not mean authorising it for all data: the boundary is drawn on the data, and it is the data that decides what can go into the prompt.

The most telling case for sovereignty is Notion. The same large enterprise was already using it widely, with governance in place and an accepted risk on the tool itself. The real question was therefore not Notion, but what its embedded AI added once grafted onto spaces containing personal data. The qualification revealed a less clear-cut situation than is often assumed. Notion does not reuse data to train its models, and it relies on subprocessors such as Anthropic and OpenAI with zero retention on its enterprise plan [14]. The sticking point lies in the hosting and the subprocessing chain, both located in the United States. Enabling AI on this data amounted to accepting exposure to the CLOUD Act, regardless of any no-training clause. Since the additional risk introduced by the AI layer was not covered, the use was refused as is, pending an alternative with controlled residency. This is a case where the framework’s main purpose is to say no, cleanly and with arguments to back it up.

The last case approaches the problem from the other end: building rather than assessing. For a public operator of essential services, intent on sending nothing outside its perimeter, I studied the feasibility of a fully sovereign internal RAG assistant, built on a SecNumCloud-qualified offering with an open-weight model. The benefit is twofold: the data never leaves a trusted environment, and auditability is complete. The trade-off is effort: inference has to be operated in-house, and RAG reopens the door to indirect injection, where booby-trapped content in an indexed source can hijack the answers, which calls for source control and sanitisation. This case illustrates the sovereign configuration from the architecture comparison in a real-world setting, chosen for control and not for cost. Detailed write-ups of all the cases are given in the appendix.

4 Putting AI to work: the value at the end of the framework

The title of this thesis signals a shift, from threat prevention to value creation. This section steps away from the project account for a moment to embrace that value-based reading, which is the angle of my title and the very purpose of the framework. A security apparatus that served only

to prohibit would miss that shift. Mine follows the opposite logic: secure in order to put AI to work, and turn control into a licence to move forward rather than a brake.

4.1 Why integrate, and what deciding nothing costs

The first value of the framework is that it makes AI use possible. In the field, business teams want AI for very concrete gains: drafting faster, or simply automating what is repetitive. Saying no to this demand does not make it go away. It moves elsewhere. This is the shadow AI mechanism, observed at several clients and confirmed by my study: prohibition pushes users towards their personal accounts. At one client that had blocked AI tools, I saw operational staff retyping excerpts of internal documents by hand into the app on their phones, then pasting the answer back into their workstations, a makeshift but very real workaround that let data leave through a door nobody was watching any more. The cost of deciding nothing is therefore twofold: the productivity gain is lost and the risk is inherited all the same, in a worse form, because it has become invisible. That is what the framework is really about: bringing these clandestine uses back into a supervised perimeter, where they can be seen, framed and defended.

Seen from this angle, a conditional yes creates value in its own right. An assistant used within a framework, with its scope, its guardrails and its logging, frees up the time it promises without opening the door we feared. For a drafting and document summarisation use that I supported, the time needed to produce the first version of a typical deliverable fell by roughly a third, that is, a few hours regained per user per week: a modest gain for one person, but one that counts at the scale of a team. The framework makes this performance defensible instead of setting it against security. That is the message I bring to my engagements.

One caveat: the value is only real if the cost stays under control. An AI use can run up a bill as fast as it delivers gains, and this is indeed one of the risks in the threat model: unbounded consumption. The framework therefore provides simple governance levers that markedly reduce spending: a token budget per use, caching of recurring prompts and batch processing. On top of that comes choosing the cheapest model capable of doing the job, rather than the largest one by reflex. Value depends as much on the gain achieved as on the cost it is allowed to incur.

4.2 AI in service of defence

The other side of the value is AI placed in the service of cybersecurity itself. The uses are maturing quickly: alert triage and correlation, analysis support, documentation reconstruction, and detection based on intent rather than on signatures. For a defence team, the time saving is real: part of the first-level triage is absorbed and the analysis of a case is shortened as a result, which frees analysts for higher-value investigations.

But this side carries a paradox that the framework forces us to face squarely: using a model hosted outside Europe to protect a European information system. The very technology that creates the risk we want to contain is then enlisted as a protection tool. The answer I adopt lies in compensating controls rather than in incantations of sovereignty. On-premises pseudonymisation,

for example, processes sensitive data within the controlled perimeter before any call to the external model, so that the remote service never sees identifying data. One has to stay clear-eyed: this measure reduces exposure without eliminating it, and pseudonymised data remains personal data within the meaning of the GDPR. It is a deliberate trade-off, not a silver bullet, and it is precisely the decision that the framework exists to make explicit rather than sidestep.

5 Project management

5.1 Methods and tools

The project was not commissioned against a formal specification. I ran it as an iterative process, starting from the field. The raw material came from engagements, from situations seen at clients, from questions that kept coming up. I first collected and sorted these situations, then derived the typologies and the method from them, and finally tested the framework on concrete cases, which forced me to correct it several times.

I treated the framework as a product to be maintained, not as a static document. The method and its grids are written in a text format that can be put under version control, and the policies are expressed as readable code, which makes it possible to track changes and to reuse templates from one engagement to the next. This choice comes from my familiarity with infrastructure as code, which I carried over to a governance deliverable, because a framework that is not versioned ages badly and soon contradicts itself. The pace of the subject bore this out: I had to revise elements several times as things moved (an offering that changed, a model that was withdrawn, a regulatory deadline that was pushed back), and versioning made it possible to absorb these revisions without starting over.

The pace was that of the apprenticeship: progress in fits and starts between engagements, with regular check-ins to validate the structural trade-offs. This time constraint had a rather healthy effect, as it forced me to aim for what was useful.

In practice, the framework was built over about eight months, alongside my engagements, and I versioned it like code: the methodological core in structured text, the policies in YAML, the maturity grid in a spreadsheet. Each application to a real case produced feedback that corrected the framework, and a dozen uses I reviewed served as successive test beds. I held to a simple rule: add nothing that had not been used at least once on an engagement, to avoid theoretical inflation. The tools involved are deliberately mundane (a version-controlled repository, a text editor, a spreadsheet), because a framework that needs a dedicated platform to be maintained does not survive the first rushed engagement.

5.2 Managerial organisation

I led the design without a dedicated team, drawing on discussions with my apprenticeship mentor and with practitioners in the field, and on several client contexts. My role therefore meant wearing several hats: the one that defines the need, the one that designs, and the one that presents and defends the result.

Interactions took place on three levels. With clients, on engagements, where I applied and tested the framework on real uses without being in control of everything: the client set the scope and what could or could not be done, and I slotted into the AI-related work package. With Advens, to turn the framework into an internal deliverable that could be shared and presented to the teams. With my academic supervisors, to stay on course as feedback came in. This lightweight organisation favoured speed of decision, and choices were challenged through discussions on engagements, internal presentations, interviews with French and Swiss CISOs from several sectors, and the complementary study.

6 Results

6.1 Meeting the initial objectives

The primary objective was to produce an actionable framework, usable on engagements in both situations: building from scratch and assessing what already exists. On this point, the result is tangible. The method exists, its five phases are backed by tools, and the threat model, the architecture comparison and the governance framework have been written and applied to real cases. The framework did not remain a statement of intent: it has been used to settle concrete situations, and that is the criterion I put first.

As for indicators, I distinguish between two things. I have concrete measurements from the campaign described earlier: maturity scores before and after remediation on real uses. What I do not yet have are adoption figures over time, and that is to be expected, since the framework is recent and is designed to be reused within Advens over the long term. What I can already state is that it has led to concrete classification decisions and remediation plans, and that it has been presented internally to serve as a common foundation. As an order of magnitude, qualifying a use, which used to involve informal exchanges spread over several days, now fits into half a day structured by the method.

These campaign results come from a first round of tests, not from statistical monitoring over time, but they show that the framework produces decisions and corrections, not just findings. The case of the tool whose AI was refused for lack of controlled residency is, in its own way, as telling a result as a successful remediation: it shows that the framework can also say no when it has to.

Furthermore, the framework provides its own measurement instruments, and this is one of the most lasting results. The maturity grid, which scores each phase from zero to three, gives a reproducible indicator for tracking a use over time and comparing several uses with one another. The framework also defines the indicators to monitor as part of supervision: the rate at which leaks are blocked at input, the number of incidents and the volume of use. In other words, the project produces not just a method but also the means of measuring whether it works.

6.2 Unforeseen impacts

Two effects I had not anticipated are worth pointing out. The first is that the assessment mode for existing solutions took on more importance than expected. I thought I would mainly be providing tooling for integrating new solutions, yet it was auditing what was already running, often without governance, that turned out to be the most immediate need. This strengthened the audit dimension of the project and brought it closer to the core of my profession.

The second effect is educational. When presenting the framework, I saw that it served as much to align teams and build their skills as to settle cases. This alignment effect was not one of the initial objectives.

6.3 The comparison matrix as a tangible result

Among the deliverables, the architecture comparison matrix, presented in Table 2, is the most directly reusable result. It turns a recurring question that is often settled by intuition, namely where to run the model, into an explicit trade-off based on criteria. A CISO can use it to justify a choice to senior management, which makes it a concrete result beyond the thesis itself.

6.4 Early lessons from the complementary study

The framework draws on my Advens engagements, a real field of practice but one limited to my own clients. To broaden the perspective, I first conducted a large number of individual interviews, with a range of profiles and with CISOs from several sectors, then opened an online questionnaire to the public to go further. Its consolidated results will be published on research.thomas.how. Here I only report signals from it, not settled results. The first respondents are deliberately varied and unconnected to my engagements, ranging from an independent developer to a data protection consultant and the technical lead of a cybersecurity organisation abroad. The study is ongoing, so I draw no statistics from it, only leads that complement a field of practice that is narrow by nature.

Despite the study’s exploratory nature, a few consistent signals emerge, and they align with the framework’s choices, which is rather encouraging. First, almost everyone built as they went, with no framework available when generative AI became an issue, which confirms that the gap lies less in the risk than in the lack of a method for dealing with it. Second, data comes up as the real control point: one respondent even makes it their first non-negotiable principle, that classification governs use, which is exactly the guiding principle of the framework. Third, prohibition drives workarounds: shadow AI through personal subscriptions is mentioned, and governed enablement is preferred to blocking. Fourth, vendor dependency is considered underestimated, ranking above the more visible technical risks. Fifth, threat-oriented reference frameworks such as the OWASP Top 10 are seen as actionable, whereas governance frameworks are still perceived as theoretical for small organisations, which matches the observation made in the state of the art.

I remain cautious about the reach of these signals, since the sample is still small. But seeing practitioners from very different backgrounds arrive at the same principles as those that emerged

from my engagements is a form of external validation that I did not expect to be so clear-cut.

6.5 Points of comparison from CISOs

Beyond my engagements and the questionnaire, I spoke with French and Swiss CISOs in the energy, finance and industrial sectors about what they already had in place and how they handled the risk. These interviews gave me points of comparison that my own field of practice alone could not provide. Finance stood out as the most advanced on vendor risk and exit strategies, driven by DORA on the French side and by FINMA's expectations on the Swiss side. Energy, as an operator of essential services, started from a more restrictive posture that was more sensitive to sovereignty. Industry, driven more by productivity, was mainly grappling with shadow AI and the protection of its intellectual assets. On the Swiss side, data residency within the country and the nFADP came up as a top-priority requirement, distinct from the European framework but converging with it in substance. These differences confirm that the dial between sovereignty and use is set according to sector and country, which the framework handles through its graduated sovereignty. The anonymised details are given in the appendix (Table 13).

7 Critical analysis

This is the part where I have to be most honest, including with myself. A framework carried by one person, alongside engagements, inevitably has strengths and blind spots, and naming them is better than hiding them.

7.1 What worked well

The decision to reason in terms of types of use rather than products delivered on its promise. Whenever a new tool came up during an engagement, it fitted into one of the three families without my having to rewrite the framework. This is the decision that has held up best over time: it protects the framework from obsolescence.

The dual mode also held up well. Using the same method to integrate a new solution or to audit one already in place avoided spreading the effort thin, and it was the audit of existing solutions that proved most useful. Finally, writing governance as code made the deliverable reusable and versionable, which makes a big difference in practice.

7.2 What worked less well

The first weak point is me. Even after challenging my choices with my apprenticeship supervisor, with professionals, with CISOs from several sectors and countries, and against a variety of client contexts, a framework carried by a single person inherits that person's habits and what they know best. My background combines governance and technical work, but I am probably more comfortable with risk qualification than with the economic side of architectures, and the comparison shows it: it structures the trade-off without claiming to settle it in figures.

The second weak point is that some of the proposed controls remain theoretical until they have been deployed at scale. I validated the method on real cases, but certain elements, in particular

the incident catalogue and its mapping to notification obligations, need to be tested over time to be fully credible.

7.3 What could have been done differently

In hindsight, I could have built on existing frameworks such as ANSSI's or ENISA's earlier and more explicitly, which would have saved me time in structuring the work. Finding the same broad principles in them actually reinforced my confidence in the direction I had taken, and takes nothing away from what I added on top: the concrete application to commercially available tools, the dual audit and integration mode, the quantified sovereignty trade-off and governance as code, which these reference frameworks stop short of providing.

7.4 The field study: a deliberate broadening of scope

The field study is not a weakness of this work but a broadening of scope that I fully stand by. I carried it out to look beyond my own clients, first through individual interviews with practitioners and CISOs from several sectors, then through a questionnaire open to the public to go further. It remains exploratory and is still ongoing, so I draw no general conclusions from it, only points of convergence that shed light on the framework. Presenting it for what it is, a living complement rather than a closed proof, seems fairer to me than making it say more than it can.

7.5 Outlook: agentic AI shifts the question to identity

The framework was designed for assistants and integrations in which a human stays in the loop. The wave now arriving, made up of autonomous agents that chain actions together and call tools through protocols such as MCP, shifts the problem. When an agent decides on its own to invoke a tool based on a natural-language description, the blurring between data and instructions mentioned earlier becomes a permanent attack surface, and the real control point is no longer the prompt but identity: what the agent is allowed to do, and on whose behalf. The signals are already there: the proliferation of non-human identities and of their poorly managed secrets has become a major issue in 2026, and a growing body of research addresses the security of agentic systems [9]. My framework does not exhaust this question, but its five phases remain useful for it, provided that a qualification of the agent's permissions is added from the scoping phase onwards and intent-based access control is added in governance. This is the direction I would take in a future version.

7.6 Analysis of my personal contribution

My contribution lies in the move to practice: turning principles already set out in the reference frameworks into a method that a consultant actually opens and uses, on commercially available tools, in engagements where data is sensitive and sovereignty is a real issue. That is precisely what these reference frameworks stop short of doing. I also know my areas for improvement. First, measuring better over time. Second, testing my choices against other perspectives even earlier. Finally, digging deeper into the economics of architectures, which I have only touched on. I see them as open workstreams.

8 Personal learning

This project taught me things that engagements alone had not yet shown me. The first is that a good security deliverable is only worth something if it is usable. I tended to concentrate on the rigour of an analysis, but I came to understand that a grid people actually open in a meeting is worth more than a perfect report that nobody reads. Designing for use, not for exhaustiveness, is a reflex I will keep.

I also came to appreciate the value of building on what already exists. When I started, I had the somewhat naive urge to build my own method. Reading the frameworks of ANSSI, ENISA or NIST, I saw that much of the path had already been laid out, and that maturity lies as much in knowing how to reuse and adapt as in inventing. It is less gratifying at the time, but sounder in the end, and it is ultimately what I did, building my method on these frameworks rather than trying to redo them.

Professionally, having carried this project from the initial need through to the presentation gave me a concrete sense of my strengths and my limits. I know how to articulate a need that nobody dictated to me and carry it through to delivery. I also know now that challenge has to be sought out earlier, because when you press on alone for too long, you end up locked inside what seems obvious to you.

Finally, the subject itself taught me to hold together two demands that seem opposed: securing and enabling. A beginner's reflex in cybersecurity is to say no. This project convinced me that the real value lies elsewhere, in the ability to authorise under verifiable conditions, to make a use possible rather than block it. That is the most useful lesson for the rest of my career.

9 Conclusion

The project started from a simple mismatch: artificial intelligence was entering companies faster than the governance that should have come with it. The answer I built at Advens does not claim to settle the matter. It offers a method for staying on top of it: a five-phase approach that serves just as well to integrate a solution from scratch as to assess and correct what is already running, equipped with a threat model, a comparison of architectures, a reversibility requirement and a governance framework expressed as code.

If I had to keep one idea, it would be the one in the title. Securing AI only makes sense if it lets you put it to work. Banning it for good does not hold: users route around it, as I have seen on engagements. Authorising everything does not hold either. That leaves a narrow path: open the use while framing it, and make that framing verifiable. This is where sovereignty, control over data and reversibility play their part, that of guardrails that make a use defensible.

I remain measured about the framework's reach. It follows in the wake of more established references, and it still needs to be tested over time. But it exists, and it has already been used to settle real cases. I sketched the next steps in the critical analysis: above all, it remains to be

put to the test in hands other than mine.

One last thing, and it is probably the most important. The subject moves so fast that a static framework would soon be outdated, as these eight months have confirmed. Mine is designed to live and to be kept up to date. It is already in use on engagements, and I maintain it on my research site, research.thomas.how, together with this thesis, the practical template of the framework and the questionnaire, which is still open. My aim is for it to become a shared reference, reused from one engagement to the next at Advens, and for it to keep evolving with uses and regulation. It is in that reuse, as much as in these pages, that its value will be measured.

10 Acknowledgements

I would like to thank Tanguy Ekizian, my apprenticeship mentor at Advens, for his guidance and for the trust that let me lead this project independently. I also thank Paul De Montalivet, my thesis supervisor at Oteria, for his feedback, which helped me scope this work properly.

I am grateful to the Advens teams with whom I discussed this subject, and to all the professionals I met over these months, those who pointed me towards research or technologies, and everyone who took the time to answer my field survey on my research website. Finally, I thank my family and friends for their support throughout these years of apprenticeship.

References

- [1] ANSSI. Recommandations de sécurité pour un système d'IA générative (ANSSI-PA-102). Agence nationale de la sécurité des systèmes d'information, 4 2024.
- [2] Anthropic. Claude API, tarification par million de jetons. anthropic.com, consulté en juin 2026, 2026.
- [3] ENISA. Multilayer Framework for Good Cybersecurity Practices for AI. European Union Agency for Cybersecurity, 6 2023.
- [4] Gartner. AI Trust, Risk and Security Management (AI TRiSM). Gartner, 2024.
- [5] Google. Secure AI Framework (SAIF), a conceptual framework for secure AI systems. Google, 2023.
- [6] Google Cloud. Generative AI, Privacy, and Google Cloud, gouvernance des données Vertex AI. Google Cloud, 2024.
- [7] Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec)*, 2023. arXiv:2302.12173.
- [8] ISO/IEC. ISO/IEC 42001:2023, Information technology, Artificial intelligence, Management system, 2023.
- [9] Sahaya Jestus Lazer, Kshitiz Aryal, Maanak Gupta, and Elisa Bertino. A Survey of Agentic AI and Cybersecurity: Challenges, Opportunities and Use-case Prototypes. arXiv:2601.05293, 1 2026.
- [10] Microsoft. Microsoft 365 Copilot, Restricted SharePoint Search et permissions Microsoft Graph. Microsoft Learn, 2025.
- [11] Mistral AI. Mistral AI, tarification des modèles. mistral.ai, consulté en juin 2026, 2026.
- [12] MITRE. MITRE ATLAS, Adversarial Threat Landscape for Artificial-Intelligence Systems. MITRE Corporation, 2024.
- [13] National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST, 2023.
- [14] Notion. Notion AI security and privacy practices. Notion Help Center, 2025.
- [15] OWASP Foundation. OWASP Top 10 for Large Language Model Applications, édition 2025, 2025.

- [16] Scaleway. Scaleway, tarification des instances GPU H100. scaleway.com, consulté en juin 2026, 2026.
- [17] Union européenne. Directive (UE) 2022/2555 (NIS2) concernant des mesures destinées à assurer un niveau élevé commun de cybersécurité dans l’Union. Journal officiel de l’Union européenne, 2022.
- [18] Union européenne. Règlement (UE) 2022/2554 sur la résilience opérationnelle numérique du secteur financier (DORA). Journal officiel de l’Union européenne, 2022.
- [19] Union européenne. Règlement (UE) 2024/1689 du Parlement européen et du Conseil établissant des règles harmonisées concernant l’intelligence artificielle (AI Act). Journal officiel de l’Union européenne, 2024.

A Appendices

The appendices gather the detailed versions of the deliverables, so as not to weigh down the body of the thesis.

A.1 Glossary of acronyms

Acronym	Meaning
AI	Artificial intelligence
LLM	Large language model
PII	Personally identifiable information (personal data under the GDPR)
GDPR	General Data Protection Regulation
AI Act	European regulation on artificial intelligence
NIS2	European directive on the security of network and information systems, second version
DORA	Regulation on digital operational resilience for the financial sector
RAG	Retrieval-augmented generation
STRIDE	Threat categorisation model (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege)
MITRE	Knowledge base of attack techniques targeting AI systems
ATLAS	
NIST AI RMF	NIST’s AI risk management framework
ENISA	European Union Agency for Cybersecurity
ANSSI	National Cybersecurity Agency of France
CISO	Chief information security officer
DLP	Data loss prevention
API	Application programming interface
ESN	IT services company

Acronym	Meaning
OWASP	Open Worldwide Application Security Project, which publishes the Top 10 risks for LLM-based applications
PDIS	Security incident detection service provider, an ANSSI qualification
SecNumCloud	ANSSI security label for trusted cloud computing
CLOUD Act	US law on extraterritorial access to data (Clarifying Lawful Overseas Use of Data Act)
FISA	US Foreign Intelligence Surveillance Act, including its Section 702
nFADP	New Swiss Federal Act on Data Protection (nLPD in French)
FINMA	Swiss Financial Market Supervisory Authority
MCP	Protocol for connecting tools to agents (Model Context Protocol)

A.2 The framework: reference document

This appendix reproduces the outline of the deliverable as it is used on engagements, independently of the thesis narrative.

Guiding principles. Data drives everything: its class, not the brand of the tool, sets the level of requirements. Sovereignty is a matter of degree, a slider spanning residency, applicable law and reversibility. The enterprise account is the rule, never a personal account. Control is proportionate to risk. Finally, what cannot be logged does not go into production.

Three integration families. T1: office-suite AI assistants, where the stakes are the indexing scope and overexposure through existing permissions. T2: AI embedded in low-code or business tools, where the stakes are data moving between connectors, secrets, and keys outside the tenant. T3: APIs called from custom integrations, where the stakes are end-to-end control, residency and auditability. A new tool inherits the watch points of its family.

The five-phase method. Table 5 lists the phases, with the checkpoint and the deliverable for each.

Phase	Checkpoint	Deliverable
1. Scoping the use	Data classified, owner identified	Use sheet
2. Sovereignty qualification	Residency and law known, enterprise access	Qualification memo
3. Threat modelling	Major risks covered, or accepted and recorded	Threat model for the use
4. Architecture choice	Architecture consistent with the data class	Reasoned architecture decision
5. Governance and supervision	Use classified, policy active, logging in place	Governance sheet and policy

Table 5. The five phases of the framework, their checkpoints and their deliverables.

Who does what. R does the work, A approves, C is consulted, I is informed. The cybersecurity consultant builds the case and makes a proposal, the CISO decides and owns the risk, the business remains responsible for the use and its data, the DPO steps in as soon as personal data is involved, and IT provides the logging tooling.

Phase	Business	Consultant	CISO	DPO	IT
1. Scoping	A	R	I	C	I
2. Sovereignty qualification	I	R	A	C	C
3. Threat modelling	I	R	A	I	C
4. Architecture	I	R	A	I	C
5. Governance and supervision	I	R	A	C	R

Table 6. Allocation of roles by phase.

Classifying uses. Authorised by default: public or non-sensitive internal data, with a provider offering European residency and no reuse of the data. Controlled: confidential data or limited personal data, under mandatory controls and review. Prohibited: sensitive data without control over residency or applicable law, or any use that cannot be logged, for as long as these conditions hold. A use moves up a level as soon as the data becomes more sensitive or control over residency weakens.

Two ways to use it. The framework is worked through either in build mode, before a new use goes live, or in assessment mode, as an audit grid applied to a use already in service, which then feeds the maturity grid and a remediation backlog.

A.3 Full threat model

An extended version of Table 1, supplemented with the risks specific to RAG and agents, giving the MITRE ATLAS reference and the typical control for each risk.

Risk	ATLAS reference	Typical control
Direct prompt injection	LLM Prompt Injection	Input filtering, separation of instructions and data, guardrails
Indirect injection via a source	LLM Prompt Injection (indirect)	Source vetting, signing, context sanitisation
Jailbreaking the guardrails	LLM Jailbreak	Output guardrails, recurring adversarial testing
Data poisoning	Poison Training Data	Control over datasets, provenance
Exfiltration through outputs	LLM Data Leakage	Output classification, masking, review
Leaks through prompts	LLM Data Leakage	Awareness training, data loss prevention, no sensitive data in prompts
Provider dependency	-	Reversibility, hybrid architecture, emergency exit route
Model drift	Erode ML Model Integrity	Quality monitoring, non-regression testing
Supervision drift	-	Logging, periodic review, indicators
Excessive Agency (agent)	LLM Plugin Compromise	Least privilege, approval of sensitive actions, dedicated identity
Vector and Embedding Weaknesses	RAG poisoning	Index access control, per-tenant isolation, source validation
System Prompt Leakage	System Prompt Extraction	No secrets in the system prompt, separate secret storage
Unbounded Consumption	Denial of ML Service	Quotas, rate limits, cost monitoring

Table 7. Reference threat model, full version.

Figure 5 reads the threat model in terms of priority, plotting likelihood against impact. It shows

that most risks cluster on the high-impact side, and that the most critical cell (high likelihood, high impact) holds data leakage through prompts.

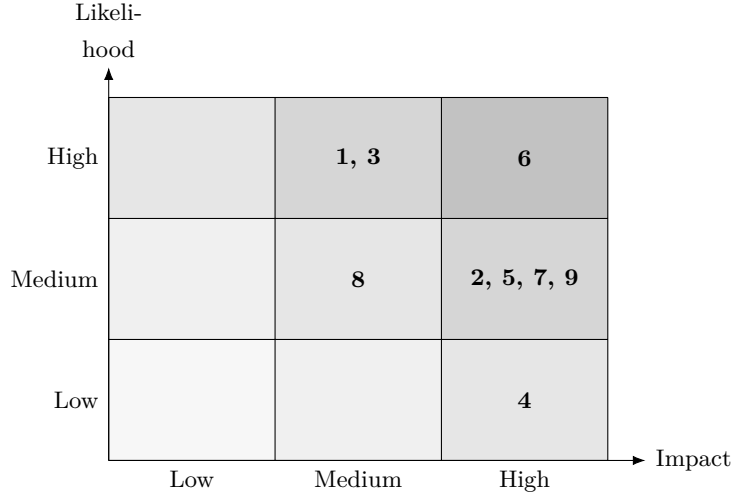


Figure 5. Risk prioritisation heatmap: likelihood against impact. The numbers follow the order of the rows in Table 1.

A.4 Technical controls toolkit

The threat model says what to address; this toolkit says what to address it with. Each dominant risk maps to concrete controls, selected according to the integration family and the data class.

- Direct injection and jailbreak: strict separation of instructions and data, input marking (spotlighting, datamarking), instruction hierarchy, input and output guardrails, recurring adversarial testing.
- Indirect injection and RAG: vetting and sanitising sources before indexing, enforcing permissions within the vector query rather than after the fact, per-tenant index isolation, source signing.
- Leaks and exfiltration: data loss prevention on prompts, output classification and masking, on-premises pseudonymisation before any external call, a ban on sensitive data in clear text in the prompt.
- Overexposure in the office suite: upstream review of permissions and sensitivity labels, restricted search as an interim measure, an overexposure audit before activation.
- Excessive Agency in agents: least privilege, a dedicated non-human identity, intent-based access control, human approval of sensitive actions.
- Unbounded Consumption: quotas, rate limits, a token budget, caching and batch processing, cost alerts.

None of these controls is universal: they are selected according to the risks retained in phase 3.

A.5 Detailed use case sheets

The cases are anonymised; no organisation can be identified.

Case A: low-code automation with built-in AI. Context: a large enterprise, dozens of make.com scenarios in production, some of them calling an AI model. Mode: assessment of what was already running. Findings: AI calls made with keys from outside the enterprise tenant, secrets in clear text in some steps. Remediation: all calls switched to the enterprise Gemini APIs through Vertex AI, with residency and logging, and centralised secrets management. Reclassification: controlled.

Case B: office-suite assistant used internally. Context: an enterprise AI assistant deployed internally. Finding: some content, notably classified material linked to public-sector or sensitive clients, must not go into the tool. Control: restriction based on data class, sensitivity labels, awareness training. Reclassification: controlled.

Case C: AI built into a knowledge management tool. Context: Notion, already governed and approved at a large enterprise, with an AI feature offered on workspaces containing personal data. Issue: the additional risk brought by the AI layer, with hosting and subprocessing in the United States and exposure to the CLOUD Act, despite no training on the data and zero retention. Decision: the AI layer refused as is, pending an alternative with controlled residency.

Case D: sovereign RAG assistant. Context: a public operator of essential services that refuses to let any data leave its perimeter. Mode: build from scratch. Study: an internal RAG assistant on a SecNumCloud-qualified offering with an open-weight model. Key points: data kept within a trusted environment, full auditability, indirect injection controlled through source vetting, at the cost of the engineering and operational effort.

A.6 Worked end-to-end example

To show the method in action, here is case A taken phase by phase, with its checkpoints. Assessment mode: the use already existed.

- Phase 1: scoping. Business automations that summarise and route documents through an AI model, on confidential internal data including some personal data. Checkpoint 1 met once the use and the data class were established.
- Phase 2: qualification. Several scenarios call the AI with keys outside the enterprise tenant, and secrets are lying around in clear text. Checkpoint 2 not met: access is not controlled.
- Phase 3: threat modelling. Dominant risks: exfiltration through uncontrolled keys and leaked secrets, both with low detectability and therefore prioritised. Required controls: centralised secrets and mandatory routing through the enterprise APIs.

- Phase 4: architecture. A managed API in an EU region through the enterprise tenant, with residency and logging, rather than self-hosting, which would be disproportionate for this volume. Checkpoint 4 met.
- Phase 5: governance. Reclassified as controlled, policy as code, logging active, incidents linked to notification obligations. Checkpoint 5 met.

Result: maturity rose from 6 to 13 out of 15 (Table 9), and two P1 gaps were fixed. In build mode, the same grid would have been used to build the automation properly from the start.

A.7 Maturity grid: scoring criteria

Each phase is scored from 0 to 3 on a common scale. 0, absent: nothing is in place. 1, partial: handled informally or incompletely. 2, controlled: in place and documented. 3, instrumented: in place, documented and supervised over time. Table 8 specifies, for each phase, what characterises the controlled level, the threshold at which a use is considered under control. Level 3 adds continuous supervision in every case, with periodic review and indicators tracked over time.

Phase	Controlled level (2) reached when
1. Scoping	The use and the data class are described and validated by a business owner
2. Sovereignty qualification	Residency, applicable law and access mode are documented and compliant
3. Threat modelling	The major risks are identified, each with a control or a recorded acceptance
4. Architecture	The configuration is chosen, justified and consistent with the data class
5. Governance and supervision	Written policy, active logging and a link to incident obligations

Table 8. Criteria for the controlled level, by phase.

A.8 Example maturity grid

The maturity grid (scored from 0 to 3 per phase) applied to case A before remediation.

```

Phase 1 scoping the use           : 2
Phase 2 sovereignty qualification : 1
Phase 3 threat modelling          : 0
Phase 4 architecture             : 2
Phase 5 governance and supervision : 1
Overall score = 6 / 15

```

Reading: a use in production but with no threat modelling and no supervision, a common risk profile. Remediation targeted phases 3 and 5 first.

A.9 Scoring method and detailed campaign scores

The scores in the body of the thesis are not set by hand; they follow from the grid. Each use was scored phase by phase against the criteria in Table 8, based on what was actually observed during the engagement: the configuration of the tools, the review of scenarios and permissions, and discussions with the teams. Each phase is worth 0 to 3, for a total out of 15. Table 9 gives the breakdown before and after remediation for each use, which makes every figure in the body traceable back to its source.

Use	Before (P1..P5)	Total	After (P1..P5)	Total
Office-suite assistant (internal)	1, 1, 0, 1, 1	4	3, 2, 2, 2, 2	11
Enterprise Gemini	2, 1, 0, 1, 1	5	3, 2, 2, 2, 2	11
make.com + Vertex	2, 1, 0, 2, 1	6	3, 3, 2, 2, 3	13
AppSheet with AI	1, 1, 0, 1, 0	3	2, 2, 2, 2, 1	9
Notion AI	2, 1, 0, 1, 0	4	Refused	-

Table 9. Breakdown of scores by phase, before and after remediation. P1 scoping, P2 sovereignty qualification, P3 threat, P4 architecture, P5 governance.

One pattern stands out: phase 3, threat modelling, is almost always at zero before intervention, and phase 5, governance and supervision, is barely under way. This is the typical profile of a use switched on by the business without going through security, and it is exactly what remediation corrects first.

A.10 Decision tree

The tree in Figure 6 sums up the logic of the framework, from the question that can stop everything (how sensitive the data is) through to go-live.

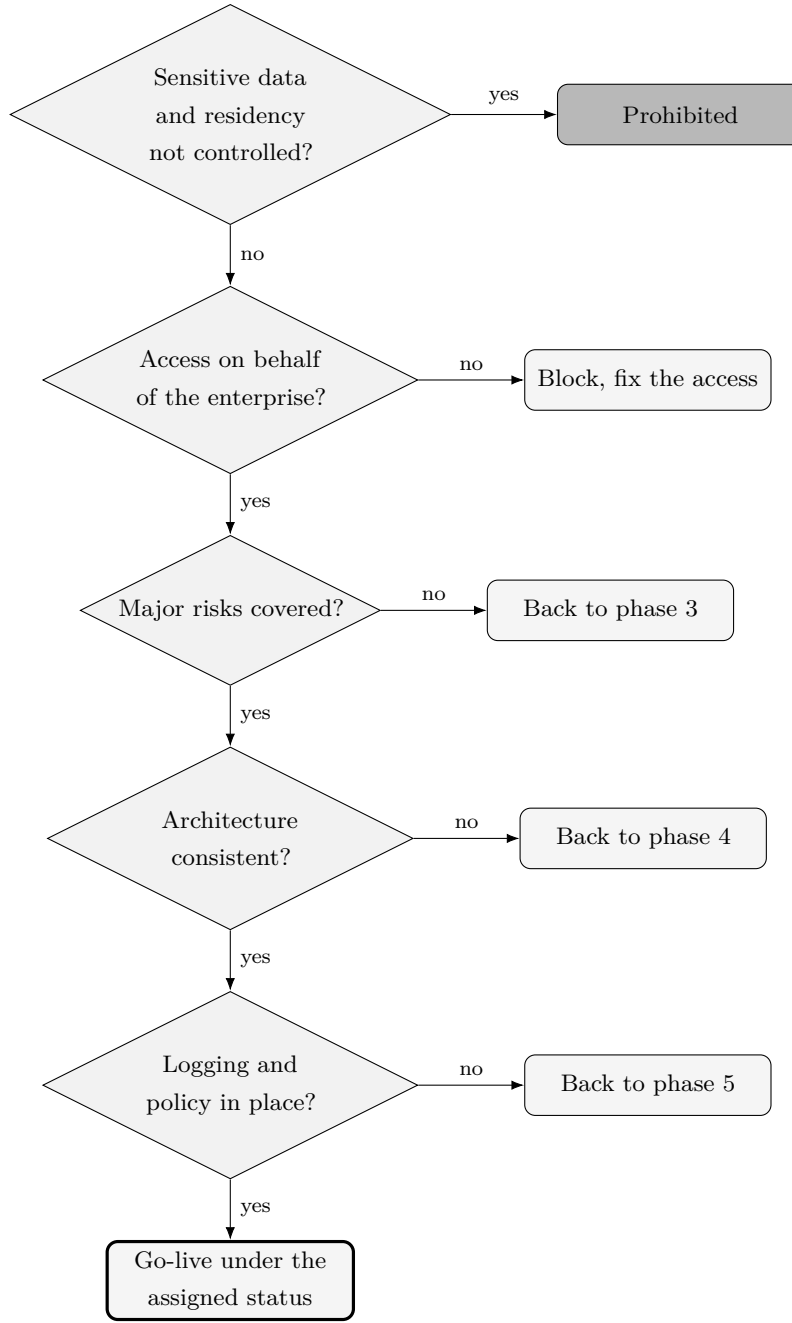


Figure 6. Summary decision tree of the framework.

A.11 Architecture cost model

Orders of magnitude recorded in June 2026, to give substance to the trade-off in Focus 2. API prices are in dollars (US providers) and the GPU cost in euros (European provider), which are of the same order at the current exchange rate. Prices change quickly: they are useful as a structure for reasoning, not as a quote.

Option	Indicative cost	Notes
Managed API, high-end model	approx. \$5 input and \$25 output per million tokens [2]	Batch processing –50%, prompt caching up to –90%
Managed API, European model	approx. \$2 input and \$6 output per million tokens [11]	Easier reversibility towards self-hosting
Self-hosted sovereign	approx. €2.73 per hour per H100 GPU [16]	Fixed cost, no per-token cost, requires engineering

Table 10. Indicative costs of the configurations, June 2026.

The break-even point is less favourable to self-hosting than a naive calculation would suggest, and it has to be set out honestly. A seventy-billion-parameter model served on a node of four H100s with an optimised inference engine reaches around three thousand tokens per second. At full load, that would amount to nearly eight billion tokens a month, but no service runs at full load, and at a realistic utilisation rate of about thirty per cent, the figure drops back to around two to three billion tokens a month. This node costs nearly eight thousand euros a month. On top of that comes redundancy, so as not to depend on a single machine, which doubles the bill, and then the cost of engineering and MLOps (monitoring, updates, hardening), a far from negligible fraction of a full-time equivalent.

Taken together, at these realistic volumes of two to three billion tokens a month, the managed API remains markedly cheaper than the sovereign node once redundancy is added, even before counting engineering. Self-hosting only comes close under very high sustained load, when a node runs continuously near its capacity, which remains rare. The consequence is clear, and it runs counter to a widespread intuition: the sovereign option is chosen for sovereignty, auditability or reversibility, almost never on price alone.

A.12 Reversibility: exit strategy diagram

Figure 7 outlines the exit path for a critical use, from a managed API to a sovereign fallback target, with its triggers and indicative lead time. It is the technical expression of the exit strategy that DORA expects.

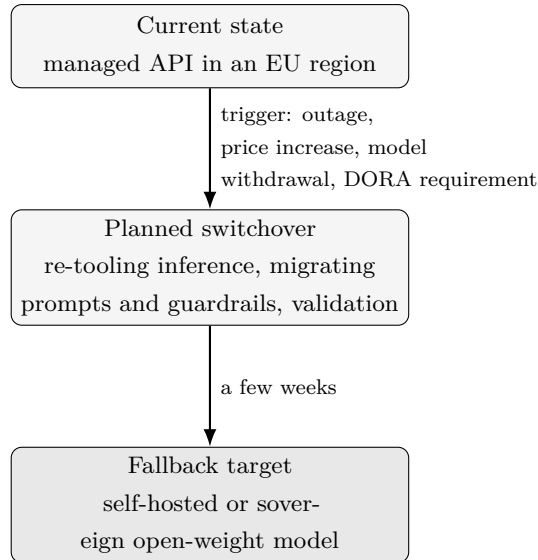


Figure 7. Exit path for a critical use, from the managed API to a sovereign option.

A.13 Mapping the framework to reference standards

The framework does not invent a theory; it operationalises existing standards. Table 11 traces this mapping, phase by phase.

Framework phase	NIST AI RMF	ISO/IEC 42001	ANSSI GenAI	OWASP LLM	AI Act
1. Scoping and classification	Map	Context, objectives	Scoping the use	-	Risk level
2. Sovereignty qualification	Govern, Map	Suppliers, data	Hosting, data	LLM03, LLM02	Residency, transparency
3. Threat modelling	Measure	Risk assessment	Attack scenarios	Full Top 10	Risk management
4. Architecture	Manage	Resources	Deployment	LLM03, LLM10	Robustness
5. Governance, supervision	Govern, Manage	Monitoring, improvement	Production, logs	LLM05, LLM06	Post-market monitoring

Table 11. Mapping of the five phases to the main reference standards.

A.14 AI Act timeline

Figure 8 summarises the phased application of the AI Act, as amended by the Digital Omnibus of May 2026. It helps place a use on the regulatory timeline rather than measuring it against a single deadline. In detail, the Omnibus postpones the obligations for high-risk systems under Annex III from 2 August 2026 to 2 December 2027, those under Annex I to 2 August 2028, and the content marking required by Article 50(2) to 2 December 2026, while adding to Article 5 a ban on nudifiers and child sexual abuse material. The other transparency obligations, including those on deployers, and AI literacy remain applicable on 2 August 2026. The agreement remains provisional, subject to its final adoption and publication in the Official Journal.

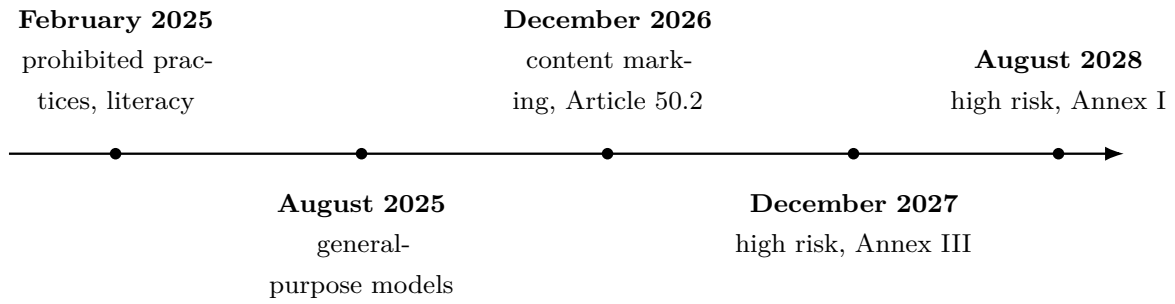


Figure 8. AI Act application timeline, including the postponements introduced by the Digital Omnibus of May 2026.

A.15 Sovereignty levels

Sovereignty is not binary. Table 12 distinguishes three levels, with the compensating controls available at each.

Level	Residency	Applicable law	Compensating controls
Qualified sovereign (SecNumCloud type)	EU	EU	Rarely needed
Hyperscaler in an EU region	EU	Exposed to the CLOUD Act	On-premises pseudonymisation, client-side encryption, customer-managed keys
SaaS or global cloud	Possibly outside the EU	Foreign	Often insufficient for sensitive data

Table 12. Three sovereignty levels and their compensating controls.

A.16 Data-flow diagram and trust boundaries

The diagram in Figure 9 models a typical use: an assistant querying internal sources through a control gateway before calling an external model. The trust boundaries, shown as dashed lines, are where data passes from one domain of control to another, and therefore where controls are concentrated.

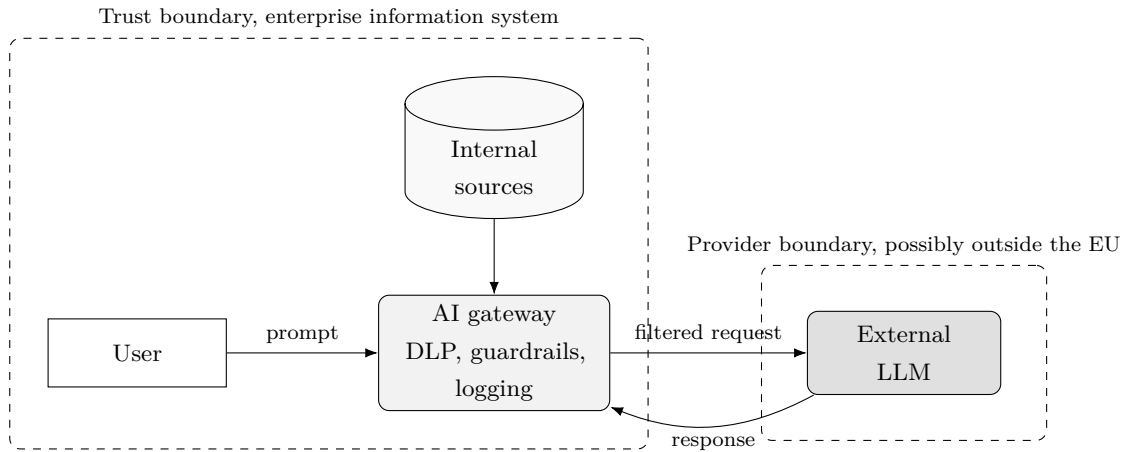


Figure 9. Data flows and trust boundaries for a typical use.

A.17 Policy as code: a complete example

An extended version of the skeleton presented in the body of the thesis, for a controlled use.

```

use_case: customer-support-assistant
data_class: limited_pii
status: controlled
family: T3
provider:
  name: managed-eu-model
  residency: eu
  applicable_law: eu
  training_on_data: false
  retention_days: 0
controls:
  input:
    dlp: true
    on_premises_pseudonymisation: true
    allowed_prompts: support_allowlist
  output:
    classification: true
    pii_masking: true
    human_review: if_sent_externally
  logging:
    enabled: true
    content: [timestamp, user, prompt_hash, output_category, guardrail_decision]
    retention_days: 365
supervision:
  review: quarterly
  kpi: [dlp_block_rate, incidents, volume, reported_hallucination_rate]
incident:
  patterns:
    - pii_leak_via_output
    - confirmed_indirect_injection
    - provider_unavailability

```

```
mapped_to_nis2:
  early_warning_h: 24
  notification_h: 72
  final_report_days: 30
```

A.18 Structure of the complementary study

The online questionnaire, which is still open, is organised into seven blocks that follow the actual path an organisation takes when confronted with AI.

- 1. The trigger point: when AI became a cybersecurity issue, and with what framework.
- 2. Trade-offs and tensions: performance versus security and sovereignty, shadow AI, incidents.
- 3. Architecture and technical choices: open models, the riskiest link, visibility over APIs.
- 4. Governance and regulation: who decides, where things stand on the AI Act, reference frameworks, budget.
- 5. AI in the service of cybersecurity: tools, the paradox of a foreign LLM protecting a European information system.
- 6. Emerging technologies and trajectory: agents, local models, RAG, embedded AI.
- 7. Outlook and closing: three-year scenario, underestimated risk, advice.

The consolidated results will be published at research.thomas.how.

A.19 Comparison points gathered from CISOs, by sector

An anonymised summary of discussions with French and Swiss CISOs from several sectors, covering what they had in place and how they handled AI risk. No organisation can be identified.

Sector	Starting posture	Observed strength	Dominant concern
Finance (FR and CH)	Governance already under way	Provider risk and exit, driven by DORA and FINMA	Confidentiality, traceability
Energy	Restrictive and cautious	Compartmentalisation, sovereignty requirements	Residency, continuity, NIS2
Industry	Adoption driven by use	Speed of deployment	Shadow AI, intellectual property

Table 13. AI risk management practices observed among the CISOs consulted, by sector.

Across all three sectors, one Swiss specificity stands out clearly: data residency within the country and the nFADP treated as prerequisites from the outset, with a strong culture of confidentiality.

Two findings run through these discussions. Data and its residency come up everywhere as the real decision point, which supports the choice of making it the first phase of the framework. And dependency on the provider is seen as underestimated in every sector, which justifies having raised reversibility to the status of an architecture requirement.